ROBOT-AIDED ENDOSCOPE CONTROL UNDER LAPAROSCOPIC SURGERY CONSTRAINTS USING DUAL QUATERNIONS

MURILO MARQUES MARINHO

MASTER THESIS IN AUTOMATION AND ELECTRONICS ENGINEERING

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

MASTER THESIS IN AUTOMATION AND ELECTRONICS ENGINEERING

ROBOT-AIDED ENDOSCOPE CONTROL UNDER LAPAROSCOPIC SURGERY CONSTRAINTS USING DUAL QUATERNIONS

MURILO MARQUES MARINHO

Supervisor: Prof. Antônio Padilha Lanari Bó
Co-supervisors: Prof. Bruno Vilhena Adorno
Dr. Mariana Costa Bernardes

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre Engenheiro em Sistemas Eletrônicos e Automação

Banca Examinadora

Prof. Antônio Padilha Lanari Bó, ENE/UnB _____
Orientador

Prof. João Luiz Azevedo de Carvalho, ENE/UnB _____
Examinador interno

Prof. Glauco Augusto de Paula Caurin, EESC/USP _____
Examinador externo

FICHA CATALOGRÁFICA

MURILO, MARQUES MARINHO
Controle de robô para auxílio em cirurgia laparoscópica usando quatérnios duais [Distrito Federal] 2014.

viv, 78p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado — Universidade de Brasília. Faculdade de Tecnologia.)

1. Manipuladores 2.Cirurgia laparoscópica
3. Quatérnios duais

I. ENE/FT/UnB II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

MARINHO, M. M., (2014). Controle de robô para auxílio em cirurgia laparoscópica usando quatérnios duais. Dissertação de Mestrado em Engenharia de Sistemas Eletrônicos e Automação, Publicação PGEA.DM-560/2014, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 78p.

CESSÃO DE DIREITOS

AUTOR: Murilo Marques Marinho

TÍTULO : Controle de robô para auxílio em cirurgia laparoscópica usando quatérnios duais.

GRAU: Mestre engenheiro ANO: 2014

Murilo Marques Marinho

SMPW Quadra 16 Conjunto 04 Lote 11 Casa H - Park Way.

71741-604 Brasília – DF – Brasil.

This thesis is dedicated to my parents, to whom I am eternally grateful for everything; and to my bother, for being a real genius and enlighten me simply by being at my side.

Dedico esta dissertação aos meus pais, aos quais serei eternamente grato por tudo e ao meu irmão, por ser um verdadeiro gênio e me engrandecer com sua companhia.

MURILO MARQUES MARINHO

Acknowledgments

Com uma orientação oligárquica como a minha, posso dizer com tranquilidade que durante este trabalho estive apoiado sobre o ombro de gigantes.

Agradeço ao Antônio, não só pela presença e suporte, mas também pela capacidade de me desafiar e melhorar nossa pesquisa. Obrigado por confiar no meu trabalho, saber avaliá-lo criticamente e me direcionar pelos melhores caminhos.

Agradeço à Mariana, pelo apoio desde o fim da minha graduação. Foi um prazer poder trabalhar conjuntamente durante todo o mestrado, nos mais diversos projetos. Aprendi muito com você e agradeço pelas tantas vezes em que você "pôs a mão na massa" para me auxiliar.

Agradeço ao Bruno, por todo o suporte dado durante esta empreitada. Mesmo distante, agradeço por ter estado sempre interessado em meu desenvolvimento e disponível para me auxiliar sempre que necessário.

Obrigado também a todos meus amigos do laboratório LARA, pela amizade e suporte durante esses dois anos.

Especialmente, durante o período das matérias, o que seria do meu estado mental sem meus companheiros de Age? Thiago, André, Tadashi, George e Cortes; obrigado não só pelos diversos momentos lúdicos, mas por todas as conversas e dicas que sem dúvida melhoraram esta pesquisa.

Obrigado também a todos os que estiveram comigo nos breves momentos em que não estive no laboratório! Agradeço aos meus queridos amigos Emílio (meu grande amigo empreendedor, sempre comigo em todas as horas), Guilherme, (a chata da) Lilah e a Ana. Pela companhia em diversos momentos, também não posso deixar de agradecer à Marcella e à Marina. Obrigado!

Por fim e sem dúvida não menos importante, agradeço á Patrícia por todo o carinho, companhia e paciência durante esses dois anos do meu mestrado. Além de linda, você é demais.

MURILO MARQUES MARINHO

ABSTRACT

ROBOT-AIDED ENDOSCOPE CONTROL UNDER LAPAROSCOPIC SURGERY CONSTRAINTS USING DUAL QUATERNIONS

Author: Murilo Marques Marinho

Supervisor: Antônio Padilha Lanari Bó

Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos e Automação

Brasília, april of 2014

This work is divided in two complementary contributions concerning the use of serial-link robotic manipulators in a laparoscopic surgery setting. At first, known techniques for singularity robustness and redundancy exploitation are adapted to the use of unit dual quaternions, which have some advantages over homogenous transformation matrices concerning compactness, while not having singularities natural to minimal representations. The performance of the adapted techniques is evaluated in a simple simulated task.

Using those techniques, we can control a manipulator robot to aid in laparoscopic procedures. As opposed to specialized surgical robots, a serial robot might be used for different procedures, lowering the involved costs. In such scenario, the safety on the pivoting point must be assured by software. This work presents a novel control strategy for controlling laparoscopic tools attached to robotic manipulators that makes use of a programmable RCM. The tool movement references are generated intuitively by the surgeon. The method is evaluated in a simulated surgical environment and presented satisfactory results, in terms of pivoting point generation error and tool positioning error.

Keywords: laparoscopy, dual quaternions, robotic manipulators.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Numbers

κ	Condition number
α, β	Damping factors used in the damped inverse of matrices
σ	Singular values

Vectors

$\vec{\theta}$	Manipulator joints absolute position vector	[rad] or [m]

Matrices

$\mathbf{J}_w$	Whitney's jacobian
$\mathbf{J}$	Dual quaternion jacobian
$\mathbf{N}$	Frame invariant dual quaternion jacobian
$\mathbf{K}$	Positive definite real gain matrix

Matrix inversions

$\mathbf{A}^{\dagger}$	The Moore-Penrose inverse of $\mathbf{A}$
$\mathbf{A}^{+}$	The least-squares damped inverse of $\mathbf{A}$
$\mathbf{A}^{-1}$	The inverse of a square full-rank matrix $\mathbf{A}$
$\mathbf{A}^{inv}$	Any of the inverses above, whichever is applicable

Quaternions

$\mathbf{r}$	Rotation quaternion
$\mathbf{t}$	Translation quaternion

Dual quaternions

$\underline{\mathbf{x}}_d$, $\underline{\mathbf{x}}_d(k)$	Desired pose
$\underline{\mathbf{x}}(t)$, $\underline{\mathbf{x}}(k)$	Current pose
$\underline{\mathbf{x}}_{\text{rcm}}$	Remote center of motion pose

Acronyms

MIS	Minimally invasive surgery
RCM	Remote center of motion
HTM	Homogenous transformation matrices
FKM	Forward kinematic model
DH	Denavit-Hartenberg
SVD	Singular value decomposition

Chapter 1

Introduction

*"*Scientia pro bono humani generis.*"*
–Science for the good of mankind

Minimally invasive surgery (MIS) is becoming the new standard for interventions in the human body, whereas open surgery becomes the backup plan [2, 3]. By reviewing the concepts behind both techniques, we can clearly see why.

Open surgery was the standard procedure when anesthetics and antiseptics became widespread, about 100 years ago. Even for diagnostic purposes, exploratory laparotomy[1] was the common tool to diagnose many forms of diseases. The length of a laparotomy incision in the abdominal wall varies accordingly to the procedure (5cm–20cm), having to be long enough to accommodate the surgeon's hands and provide good operative vision.

Although allowing the surgeon the best dexterity possible, those incisions became dreaded to both patients and hospitals. From the patients' point of view, large incisions result in long recovery times to form large scars, with considerable risk of post-operative infection. Moreover, in the case of exploratory laparotomies, those risks can go unrewarded with no more information about the disease being treated. From the public health system economical view, longer hospitalization times often correspond to higher costs.

Several pioneers contributed during the last 100 years to the development of modern *laparoscopy*, which is the MIS counterpart of laparotomy. Instead of a single long incision, laparoscopy uses one or more small (2mm–12mm) incisions. The surgeon uses a long endoscope and other tools inserted through those incisions to diagnose and treat diseases. After the laparoscopic techniques mature in medical practice, they usually provide less surgical trauma, smaller hospitalization times, faster return to daily-life activities and better aesthetics in relation to their corresponding open surgery techniques. As disadvantages, longer procedure times and steeper learning curves are usually reported [2, 3, 4].

For illustrative purposes, we can compare laparoscopy and open surgery in three different procedures with respect to intraoperative and post-operative times, as shown in Table 1.1 and

[1] A laparotomy is a form of open surgery involving a large incision through the abdominal wall to gain access into the abdominal cavity.

	Mean operative times [min]		Mean hospitalization times [day]	
Procedure	Open surgery	Laparoscopy	Open surgery	Laparoscopy
RN	110	115	7.9	3.8
RCC	172.2	217.3	10	6.3
CE	133	112	3.3	3.2

Table 1.1: The mean intraoperative and postoperative times for 3 procedures comparing open surgery and laparoscopy. (RN = retroperitoneal nephrectomy, RCC = resection for colorectal cancer, CE = cholecystectomy in the elderly)

discussed as follows. In the case of retroperitoneal nephrectomy[2] (RN) [5], laparoscopy provided better results concerning hospitalization times, while operative times are comparable between both methods. Another study [4] compares both techniques in resections for colorectal cancer[3] (RCC). Its results show that open surgery required less intraoperative time, while a shorter hospital stay favored laparoscopy. In the last study [6], which is about cholecystectomy[4] in the elderly (CE), laparoscopy had a faster intraoperative time and both techniques demonstrated comparable hospitalization times.

Although laparoscopy is now seen as the standard technique in many procedures, it needed to overcome skepticism that protected the classical open surgery methods from changing [2]. Moreover, as any great advancement in science, the progresses on laparoscopy were deeply dependent on the evolution of other technologies. Therefore, we begin by briefly reviewing the historic context of laparoscopy in the next section. Afterward, the surgical method is thoughtfully explained in section 1.2.

1.1 Historic contextualization

The earlier concepts and methods encircling the modern laparoscopy are usually accredited to Georg Kelling and Hans Christian Jacobaeus [2, 7, 8].

In 1901, Kelling was researching solutions for gastrointestinal bleeding into the abdominal cavity. By that time, the only available method for diagnosis and treatment was the laparotomy, which could worsen the patient's condition. Also, in an economic standpoint, Kelling wanted to shorten his patients' hospital stay. He then hypothesized that injecting air inside the abdominal cavity[5] could by itself stop such bleeding. Although most medical instrumentation was still fairly rudimentary in 1901, Kelling carried out numerous experiments on live dogs to see the effects

[2]Retroperitoneal nephrectomy is a treatment for both benign and malignant conditions related to renal and adrenal lesions. It consists of partial or full removal of a the patient's kidney.

[3]Colorectal cancer resection is the surgical removal of a colorectal tumor. It is the only treatment with curative intent for colorectal cancer [4].

[4]Cholecystectomy is the surgical removal of the gallbladder in the treatment of cholecystitis which affects 23.7% of people with 70 years or older [6].

[5]Insufflating the abdominal cavity with air is called pneumoperitoneum in medical literature.

of insufflating air inside the thorax. He inserted a cystoscope[6] through a trocar in the unopened abdominal cavity to observe the effects of the air pressure on the abdominal organs [7]. Despite the fact that he was not able to stop bleeding with this technique, it comprised most of the procedure known today as laparoscopy [2, 9]. In the years after, he devoted much of his work to refining the instructions of his technique. Those instructions concern patient preparation, contraindications and placement methods for the trocar [2].

Also in 1901, Jacobaeus performed a similar procedure on humans and coined the term laparoscopy[7]. In contrast to Kelling's method, he inserted the cytoscope without inflating the abdominal cavity with air [9]. He was able to perform those procedures as his patients had ailments that caused an accumulation of fluid in the abdominal cavity[8], emulating the effects of air insuflation. Although his technique differed in this aspect with its modern counterpart, Jacobaeus is mostly responsible for the popularization of laparoscopy. By 1912 he had performed about 97 laparoscopies to treat many different forms of pathologies [2]. Around that time, several reports were published worldwide showing that the technique had spread. Reports came from various locations of Europe such as France, Italy and Denmark; as well as from Brazil [9].

Laparoscopy was still only used as a diagnostic tool. In the following years, laparoscopy flourished as a form of interventional medicine, with advancements in electricity that led to the creation of the incandescent light. After the research stagnation during the second world war, laparoscopy ironically suffered a period of oblivion due to shortcomings in the usage of electronics in sterilization and cauterization. Also, there were many causalities related to inaccurate insufflation equipments. For those reasons, laparoscopy was even banned on Germany from 1956-1964 [2].

For some researchers, the question was not whether laparoscopy was a good means of surgery, but whether it could become safe and predictable enough. In the hands of many scientists, such as Kurt Semm, Kalr Storz, Alvin Siegler, and Melving Cohen, technology came in aid to fix the existing problems. After the creation of automatic insuflators, cold lights, and safer cauterization tools, the predictability of the procedures grew enormously. Then, the most recent revolution was the video-laparoscopy in about 1985. It gave surgeons the ability to view and record the patient's viscera[9] with increasing resolution and smaller equipment throughout the years [2].

With those advancements, laparoscopy finally became a safe form of surgery and is already deeply inserted in today's medical paradigm. Its benefits include more precise visualization of tissues, less postoperative pain, shorter hospital length and increased aesthetics. Given those, laparoscopy–whenever feasible–became the first means of surgery, whereas open-surgery rests as a backup if there are complications [2, 3, 10].

The number and complexity of procedures that can be done laparoscopically is indeed enormous. In fact, in 2004 several physicians wrote a manual [10] describing 63 different laparoscopic procedures to treat and diagnose many forms of diseases in the abdomen and esophagus alone. As

[6]The cytoscope is a rudimentary form of the endoscope. The surgeon had to look directly along the shaft direction and had fairly limited view.

[7]Actually, he used the term laparothorascoscopy but it evolved to laparoscopy.

[8]Accumulation of fluid in the abdominal cavity is called ascites in medical literature.

[9]Viscera are the internal organs in the main cavities of the body, especially those in the abdomen.

the amount of possible procedures grew, laparoscopy became a term applied generically to many different areas in the body. In order to better state our problem, we henceforth limit the scope of laparoscopy to only those procedures performed in the abdomen.

1.2 Problem formulation

The modern version of the technique envisioned by Kelling and Jacobaeus is throughly described in the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) manual for basic laparoscopy [3].

After all equipment has been checked and the initial setup of the operating room is made, the patient is laid down face up on the operating table. Then, the steps for the access to the patient's abdomen are shown in Figure **??** and can be summarized on [3, p. 61-71]: (1) insert a Veress needle[10] in the patient's abdomen for initial insufflation with CO2; (2) enter the patient's abdominal skin using a trocar; (3) insert a scope in that trocar to be able to see inside the cavity; (4) insert other trocars under direct scope visualization as necessary for the other surgical tools; (5) explore the patient's organs.

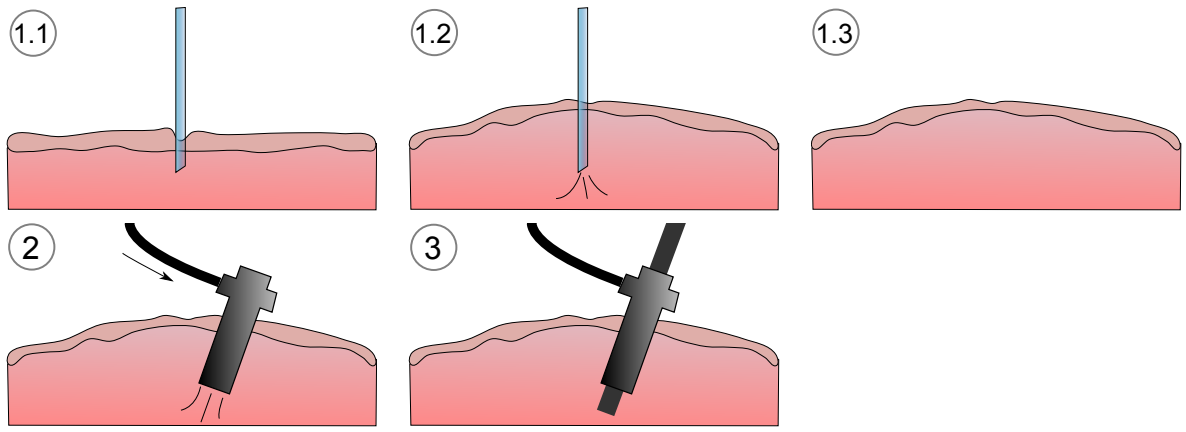

Figure 1.1: Abdomen access. (1.1) Veress needle insertion. (1.2) Veress needle insuflation. (1.3) Veress needle removal. (2) Trocar insertion and further insuflation. (3) Endoscope insertion through the trocar.

Diagnostic laparoscopy is an example of a procedure that has those starting steps. The incisions are very small (the size of each port may vary from 2 mm to 12 mm in comparison with a single 10 cm to 20 cm incision for open surgery). It may be noninvasive to the point of being done in a patient with local anesthesia only, while the patient is awake. In some cases, they may even be asked to direct the surgeon to the point of maximal pain, or even to make slight body movements to help visualization [10, p.481-489].

Most types of laparoscopy report shorter postoperative hospitalization [10], reduced complications following the procedure[11] and lower transfusion rate. However, even with all the ad-

[10]The Veress needle is a hollow needle used for the initial insuflation to minimize chances of complications.
[11]Complications following the procedure are called morbidity in medical literature.

vancements made in laparoscopy, there are still some drawbacks to the surgeons. Some of those techniques are very difficult as stated by the physicians themselves, and should only be performed by skilled laparoscopy surgeons. Moreover, it is commonly reported that laparoscopic procedures have a steeper learning curve in relation to their open surgery analogues.

The difficulty in performing laparoscopic tasks comes from the fact that the incision in the patient's skin restricts the motion of the scope or tool. To simplify the analysis, the intersection of the trocar axis with the skin incision can be approximated by a pivoting point, called the *trocar point*. Therefore, this means that (1) all movements outside the patient are mirrored inside the patient's cavity (Figure 1.2 left); and (2) all movements of the tool are restricted to only four degrees of freedom (Figure 1.2 right). Moreover, (3) the surgeon can only interact with organs and inner structures by using tools, thus losing dexterity and tactile feedback. Furthermore, (4) an assistant surgeon may be required, whose only task is to move the endoscope during the whole procedure.

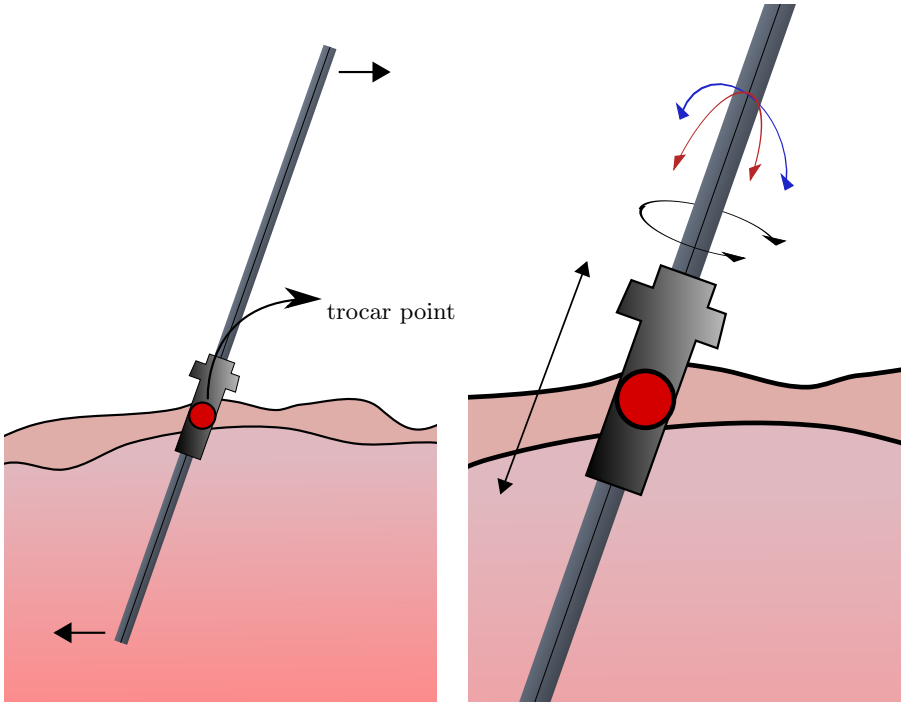

Figure 1.2: Laparoscopic issues. Mirrored movements (*left*) and laparoscopy constraints (*right*).

In order to remove from the surgeon some of the physical and mental strains of compensating for all those laparoscopic technique difficulties, we may use robotic manipulators. Since their conception, robotic manipulators have been used to extend the capabilities of men and to replace men at tedious jobs [11].

1.2.1 Robotic surgery

The idea to tackle laparoscopy difficulties using robotics is not new. For instance, the SAGES manual [3] has a chapter dedicated to the use of robots in laparoscopic surgery. On the view of the surgeons themselves, the introduction of robotics in MIS came as no surprise, given the increased

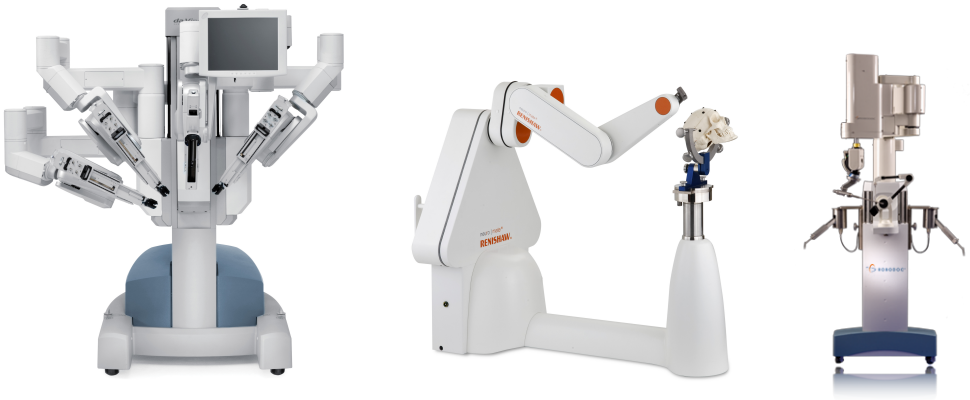

Figure 1.3: Robots currently used in medical practice. (*left*) daVinci (source: Intuitive Surgical). (*center*) NeuroMate (source: Renishaw). (*right*) ROBODOC (source: curexo).

precision and improved quality associated with industrial robots [3, p. 191]. This provided the motivation for *robotic surgery*.

Some robotic systems are already aiding in medical practice worldwide, as shown in Figure 1.3. For instance, one cannot mention robotic surgery without citing the daVinci surgical system. Being currently the only FDA[12] approved robot for laparoscopic surgery, it is available into worldwide market with over 1700 devices introduced by 2010. There are also the NeuroMate robot used in stereostatic surgery and the ROBODOC used in orthopedic surgery. Moreover, there are other systems directed to different purposes, such as camera positioners instruments and so on [3].

Robotic devices used in laparoscopy need to generate a pivoting point to coincide with the trocar point which may either be generated through mechanical constraints of specialized systems (such as the daVinci) or through software using general purpose robots. Specialized systems are precise and reliable, but have the disadvantage of being high cost and have restricted use to the specific application they have been developed for. General purpose robots are intended to be adaptable to various existing and future medical procedures, like the newly proposed MIRO system from DLR[13] [12]. It has advantages of pivot flexibility, increased maneuverability and overall versatility. Moreover, by using general purpose in spite of specialized systems, costs should be diluted among the several different procedures they are able to perform.

In the laparoscopy scenario, such robots may aid in two relatively distinct settings: moving instruments or moving the camera [13]. As the tools have a direct contact with the patients tissues, instrument motion is most concerned with higher accuracy and repeatability as seen in industrial robotics. Safer procedures are expected with robots, by allowing the surgeon hand movements to have tremor filtering and scaling. Camera motion is more associated with endurance and coordination. In laparoscopic procedures, the main surgeon usually handle tools with both hands while an assistant surgeon moves and holds the camera in place, which requires coordination between them. Using a robotic system controlled by the main surgeon to hold and move the endoscope

[12]FDA is the United States Food and Drug Administration office.
[13]DLR is the aeronautics and space research centre of the Federal Republic of Germany.

allows less time to be wasted in communication between surgeons [14]. Also, the assistant surgeon may focus on another task or be free to carry out another procedure in a different operating room.

Hence, our purpose in this work is to develop a framework for the general purpose serial-link manipulator for reliable control of an endoscopic camera under trocar constraints for which the controlling surgeon should only provide intuitive commands. This work is inserted in the CLARA project, which aims the development of national medical instruments and technology.

1.3 Results

Concerning kinematic control of serial-link manipulators, two major areas of research are singularity robustness and redundancy exploitation. In this work, classic singularity robustness techniques are adapted to the unit dual quaternion kinematic control formulation. Beyond adapting classic redundancy exploitation techniques to the dual quaternion framework, a numerical method using redundancy in singularity evasion is proposed. The adapted controllers are evaluated in simulations. All techniques herein are applicable in any serial-link manipulator geometry.

Finally, by integrating the aforementioned techniques, a laparoscopy kinematic controller for the serial-link manipulator is devised. It was designed so that the endoscope motion has only safe displacements in relation to the trocar point, while the desired endoscope positions are provided by user commands through a haptic interface. Different evaluations were performed in order to validate the method: a basic simulated scenario performance evaluation of a single camera motion, a comparison among two different manipulator geometries and finally a system usability experiment, in which 20 individuals used a physical haptic interface to command a simulated endoscope. Moreover, in relation to prior techniques proposed in the literature, our controller provides safer tool positioning considering the laparoscopic constraints.

1.4 Manuscript overview

At first, the reader is introduced to kinematic control in chapter 2. It begins with the mathematical and historic backgrounds in section 2.1, also introducing redundancy and workspace singularities. After, the adaptation of the kinematic control formulation to the unit dual quaternion case is shown section 2.2. In sections 2.3 and 2.4, known techniques for singularity robustness and redundancy exploitation are adapted for the dual quaternion case, which is the first contribution of this work. The numerical method for singularity evasion is shown in section 2.4.1, which is a second contribution.

After the reader is familiar with kinematic control, chapter 3 begins with the state of the art in robotic laparoscopy in section 3.1. The novel laparoscopy controller for the generic serial-link manipulator is shown in section 3.2, being the third and last contribution of this work. The basic functionally of the system is evaluated in section 3.3 by moving the endoscope to a fixed position with increasing values for N. In the second evaluation, shown in section 3.4, two distinct

manipulator geometries are assessed in moving the endoscope in a conical helix path. The final evaluation is shown in section 3.5, in which we assess the user interaction with the system.

Whenever necessary, notations and operations related to dual quaternions can be seen on section I.1.

1.4.1 Contributions index

1. Classic singularity robustness techniques are adapted to the unit dual quaternion framework in section 2.3;
2. classic redundancy exploitation techniques are adapted to the unit dual quaternion framework in section 2.4;
3. a numerical method for computation of gradients and its use in singularity evasion is shown in section 2.4.1;
4. the novel controller for serial-link manipulators in laparoscopic procedures using dual quaternion constrained interpolation is shown in section 3.2.

Chapter 2

Kinematic Control

"Simplicity is the ultimate sophistication."
–Leonardo da Vinci

There are several techniques for controlling serial-link manipulator robots. As those used in medical settings should operate slowly by default, kinematic control is applicable and, being conceptually simple, will be our chosen technique in this work. Before further examining the robot laparoscopy problem, this chapter focus on describing the challenges and characteristics of kinematic control.

Therefore, the first section in this chapter contains a summarized historic evolution that brought kinematic control to the current state. As much as possible, I try to omit mathematical details that would break the reading pace of the first reader.

In section 2.2, the reader is then introduced to the unit dual quaternion kinematic control formulation, along with a brief reminder of how rigid motions are described using unit dual quaternions. After, in section 2.2.3, the frame invariant error is reviewed.

After the basic formulation in unit dual quaternion is shown, sections 2.3 and 2.4 contain the adaptation of regular kinematic resolutions for singularity robustness and redudancy resolution to the unit dual quaternion case. These adaptations are a contribution of this work.

2.1 Historical background & problem formulation

Before 1969, the available controlling schemes of manipulators were very limited. Indeed, two major ways of controlling were master-slave control and rate control. The first is the control method in which a model of the robot (master) is moved and the remote manipulator (slave) copies the movement. The last concerns controlling separately each of the manipulators joints with a set of switches or joystick inputs. The implementation simplicity of such control methods made them popular, but they required a fairly high skill of the operator which hindered any attempt of achieving complex tasks in acceptable time frames.

In 1969 Whitney proposed a work [15] in the theme of medical robotics, related to prosthetic

arm control. It contains the base of kinematic control schemes, named *resolved motion rate control*. He noticed that attempting to perform complex tasks selecting how the robot joints should move– in what he termed *arm coordinates*–was a counter-intuitive scheme. Indeed, it seems irrelevant in most of our daily tasks any reference to arm coordinates. You would simply think to move your hand up to grab a cup in a top shelf, move it left to reach the tap and so forth. In an evolutionary aid for keeping our sanity, no direct reference to how any of our joints by themselves should be configured needs to come to mind. Defining a task in moving a tool up, down, left, right and so on–which Whitney named *world coordinates*–would be more natural for humans in both control and task planning.

Thus it would be a reasonable wish to create a framework in which manipulators can be controlled to perform tasks more intuitively. Back to our daily life example, we can see that in order to move your hand up to grab a cup, many, if not all, joints in your arm must move simultaneously at different time-varying rates. What we want is to apply this concept in a mathematical formula. *Resolved motion* thus means that the motion of the various joints are resolved and combined to generate motions of the tool along world coordinates.

To begin the mathematical formulation of the resolved motion rate control, we first group the pose, or other relevant world coordinate variables of the manipulator tool in a vector $\vec{x}$ of size m. After, the position of some, or all manipulator joints are grouped in $\vec{\theta}$ of size n. We first begin defining our objective, which is

$$\vec{\theta} = g(\vec{x}); \tag{2.1}$$

that means: how the robot joints $\vec{\theta}$ should be configured in order to obtain the reference $\vec{x}$. To begin our search for g, we start by relating some vector valued function with information we already have

$$\vec{x} = f(\vec{\theta}). \tag{2.2}$$

Using our example $\vec{x}$ and $\vec{\theta}$, f is the manipulator *forward kinematics model* (FKM). The FKM can be found for any serial link robotic manipulator using its *Denavit-Hartenberg* (DH) parameters [1, 16]. So, a natural solution for g would be to try finding f^{-1}. However, the inversion of f is seldom straightforward. Furthermore, if $\vec{\theta}$ has higher order than $\vec{x}$ there could[1] be infinite solutions for this problem.

Proposing another approach, Whitney differentiated (2.2) to find

$$\dot{\vec{x}} = \frac{\partial f(\vec{\theta})}{\partial \vec{\theta}} \frac{\partial \vec{\theta}}{\partial t}, \tag{2.3}$$

in which

$$\mathbf{J}_w(\vec{\theta}) \triangleq \frac{\partial f(\vec{\theta})}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial f_1(\vec{\theta})}{\partial \theta_1} & \cdots & \frac{\partial f_1(\vec{\theta})}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\vec{\theta})}{\partial \theta_1} & \cdots & \frac{\partial f_n(\vec{\theta})}{\partial \theta_n} \end{bmatrix}. \tag{2.4}$$

We then combine (2.3) and (2.4) to find

[1]There is the possibility of loss of dexterity in some *singular* configurations that reduce the system to one or no solutions. This is explained in section (2.6).

$$\dot{\vec{x}} = \mathbf{J}_w(\vec{\theta})\dot{\vec{\theta}}, \tag{2.5}$$

which is a nice linear relation. Also, $\mathbf{J}_w$ in this work is defined as the analytical Jacobian. If $m = n$ and $\mathbf{J}_w(\theta)$ is full rank, the backbone of kinematic control is obtained

$$\dot{\vec{\theta}} = \mathbf{J}_w^{-1}(\vec{\theta})\dot{\vec{x}}. \tag{2.6}$$

Whitney had a very intuitive insight on the meaning of (2.6). If we had a group of switches to control a manipulator, we could associate each switch with an entry in $\dot{\vec{x}}$ and use (2.6) to feed the output $\dot{\vec{\theta}}$ to the robot manipulator as motor speeds. By doing this, we have a way to control the tool configuration directly in world coordinates.

Such model has issues natural to linear systems. If $m > n$, $\mathbf{J}_w^{-1}(\vec{\theta})$ is not defined, and there could be infinite solutions to (2.6). A manipulator is called redundant whenever this happens, that is, whenever $\vec{\theta}$ has a higher order than $\vec{x}$. In the pose control case for example, a robot is redundant if it has more than 6 degrees of freedom. Whitney noticed that we may tackle this issue by adding an optimality criterion in the inversion, so that the optimal solution is found within the infinite set. An example of this is to use the Moore-Penrose pseudoinverse, which minimizes the norm of the arm velocities. There are also other very interesting solutions that exploit the redundancy in multitasking, adding a secondary objective for joint limit avoidance, singularity avoidance and so on. Further on this topic can be seen in section 2.4.

Another issue when using (2.6) is the existence of workspace singularities. A robot configuration $\vec{\theta}$ is singular if its corresponding $\mathbf{J}_w(\theta)$ is rank-deficient. The problem is not in the singular position itself[2], but in its neighborhood. Pseudo-inverses have ill-conditioning issues in close vicinity of singularities, in which small world coordinate reference velocities may require unfeasible velocities in the arm coordinates in (2.6). A geometric insight can help visualize some common singular configurations (e.g, when the manipulator is stretched to its limit), but other singular configurations are not so intuitive. A broader description on the issues and solutions related to singularities can be seen in section 2.3.

For now, we leave aside both redundancy and workspace singularities—consider that $\mathbf{J}_w(\vec{\theta})$ always has a suitable inverse given by a $\mathbf{J}_w^{inv}(\vec{\theta})$. With (2.6), we can find the velocity of the arm motors in order to have a given velocity of the tool in world coordinates; but yet we did not find (2.1), that is, the arm coordinates for a desired configuration of the tool in space.

Since we now have (2.6) to calculate the joint velocities for a desired tool velocity, we could devise the following algorithm: given the initial tool position $x(0)$, a final tool position x_d, and a timeframe t_f; we find a reference tool velocity trajectory $\dot{\vec{x}}(t)$. As a simple example, consider moving the tool from the initial position $(0, 0, 0)$ to $(1, 0, 0)$ within 5 seconds, which results in a $\dot{\vec{x}}(t) = (0.2, 0, 0)$. If $\dot{\vec{x}}(t)$ is followed, it is expected that the robot would reach x_d after t_f has passed. Now, let us consider that the goal is to control the joint velocities of a manipulator in a computational system with a sampling period T, such that t_f/T is an integer. The result of such

[2]The pseudoinverse implemented in this work uses the singular value decomposition of the matrix, which guarantees that a pseudoinverse is always defined. More details are shown in section 2.3.

algorithm would be evaluating

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}(\vec{\theta}(t))\dot{\vec{x}}$$

t_f/T times. The actual displacement of the joints can be found by integration

$$\begin{aligned}\int_0^{t_f} \dot{\vec{\theta}}(t)dt &= \int_0^{t_f} \mathbf{J}_w^{inv}(\vec{\theta}(t))\dot{\vec{x}}(t)dt,\\ \vec{\theta}(t) &= \int_0^{t_f} \mathbf{J}_w^{inv}(\vec{\theta}(t))\dot{\vec{x}}(t)dt + \vec{\theta}(0).\end{aligned} \tag{2.7}$$

Considering the tool velocity was constant during the sampling period, we apply Euler integration in (2.7),

$$\vec{\theta}(k) = \sum_{l=0}^{t_f/T} \mathbf{J}_w^{inv}(\vec{\theta}(l))\dot{\vec{x}}(l)T + \vec{\theta}(0).$$

Considering now an additive perturbation vector $\vec{b}(k)$,

$$\begin{aligned}\vec{\theta}(k) &= \sum_{l=0}^{t_f/T} [\mathbf{J}_w^{inv}(\vec{\theta}(l))\dot{\vec{x}}(l)T + \vec{b}(l)] + \vec{\theta}(0)\\ &= \sum_{l=0}^{t_f/T} \mathbf{J}_w^{inv}(\vec{\theta}(l))\dot{\vec{x}}(l)T + \sum_{l=0}^{t_f/T} \vec{b}(l) + \vec{\theta}(0),\end{aligned} \tag{2.8}$$

in which $\vec{b}(k)$ groups the effects of the variations of the tool velocity during the sampling interval. This variation has many sources, such as external disturbances, the variations of $\mathbf{J}_w(\vec{\theta})$ as $\vec{\theta}$ changes in (2.6), and so on[3]. By inspecting (2.8), it becomes clear that such open-loop algorithm has no guarantee of convergence: instead of reaching x_d, the tool would most certainly end somewhere else.

The keyword in the last sentence motivates the use of closed-loop schemes. For the sake of cleaner notations, we henceforth use $\mathbf{J}_w$ in place of $\mathbf{J}_w(\theta)$ with no loss of meaning. We restart with (2.6) using our suitable inverse $\mathbf{J}_w^{inv}$

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}\dot{\vec{x}}, \tag{2.9}$$

and set $\dot{\vec{x}}$ as a variable velocity of the tool in the direction of our desired $\vec{x}_d$

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}[\vec{x}_d - \vec{x}(t)]. \tag{2.10}$$

This means that we want the tool to move with a variable velocity in the direction $[\vec{x}_d - \vec{x}(t)]$—from where it is to where we want it to go—, going exponentially slower the closer it is in relation to the goal. Finally, we add an arbitrary gain $\mathbf{K}$ in (2.10) so we can control the rate of convergence

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}\mathbf{K}[\vec{x}_d - \vec{x}(t)]. \tag{2.11}$$

[3]In [1], the author states this algorithm would not work given the effects of *numerical drift*.

It can be shown that (2.11) converges for $\mathbf{K} > 0$ [1]. Altering (2.11) to the discrete case, we can find the final form of the algorithm when we can either control velocity of the arm joints

$$\dot{\vec{\theta}}(k) = \mathbf{J}_w^{inv}\mathbf{K}[\vec{x}_d - \vec{x}(k)] \tag{2.12}$$

or their position

$$\begin{aligned}
\frac{\theta(k+1) - \theta(k)}{T} =& \mathbf{J}_w^{inv}\mathbf{K}[\vec{x}_d - \vec{x}(k)] \\
\theta(k+1) - \theta(k) =& \mathbf{J}_w^{inv}T\mathbf{K}[\vec{x}_d - \vec{x}(k)] \\
\theta(k+1) =& \theta(k) + \mathbf{J}_w^{inv}T\mathbf{K}[\vec{x}_d - \vec{x}(k)] \\
\theta(k+1) =& \theta(k) + \mathbf{J}_w^{inv}\tilde{\mathbf{K}}[\vec{x}_d - \vec{x}(k)].
\end{aligned} \tag{2.13}$$

Notice that we now have a $\tilde{\mathbf{K}} = T\mathbf{K}$, directly influenced by T. This gives some insight of the effect of T in the stability of the control loop. Consider a constant $\tilde{\mathbf{K}}$ such that (2.13) is stable. As T defines the amount of time the system will take to correct itself, a bigger T should be compensated by a smaller $\mathbf{K}$ in order to keep stability.

The result (2.13) still has an issue. The instantaneous nature of (2.6) gives no guarantee to what happens inside sampling intervals. As a rule of thumb, joint velocities and T should be small. Gladly, it is the case of the laparoscopy framework proposed in this work.

As we can find both the Jacobian and $\vec{x}(k)$ for the general manipulator case (see [1, 17, 18] using homogenous transformations or [16] using dual quaternions), (2.13) is a very appealing iterative solution for the manipulator inverse kinematics and consequently is the current basic standard for kinematic control. We can further tinker with (2.13) to add singularity robustness and multiobjective tracking, as will be shown in sections 2.3 and 2.4, respectively. First, it is important to motivate and introduce the use of dual quaternions in kinematic manipulator control.

2.2 The unit dual quaternion formulation

After the solution (2.13) is found, there is still some ground to be covered until it can be used as a control strategy.

One basic question is which mathematical object should be used to represent $\vec{x}$ and its derivatives. That is, how to describe rotations and translations of rigid bodies. As the problem we are trying to solve is the description of the tool pose, we consider only representations that can unify rotation and translation. Henceforth, some knowledge about dual quaternion algebra is required. If necessary, notation and operator definitions can be seen on section I.1.

2.2.1 Homogenous transformation matrices $\times$ dual quaternions

A *homogenous transformation matrix* (HTM) $\in \mathbb{R}^{4\times4}$ has 12 free parameters used to describe both rotation and translation simultaneously. It is the commonly used tool in the mathematical development of many robotics textbooks [1, 17, 18]. The HTM is a compact notation that contains

the description of a rotation given by a rotation matrix $\in \mathbb{R}^{3\times3}$ and the vector representation of a translation $\in \mathbb{R}^{3\times1}$. One problem related to the use of HTMs is that they cannot be directly in the control equation (2.13). That is, even though we can obtain the FKM of a manipulator as a HTM, we have to convert it to another minimal representation to obtain $[\vec{x}_d - \vec{x}(k)]$.

U*nit quaternions* are a four-parameter rotation representation, constrained to unit norm. They also carry a unique algebra, being able to represent all rotations with lesser parameters than a rotation matrix and without representation singularities that hinder the use of minimal rotation representations. We then show how a unit quaternion is able to represent a arbitrary rotation.

Fact 2.1. Unit quaternion rotation. *The rotation of an angle ϕ around an axis $\mathbf{n} = n_x\hat{\imath}+n_y\hat{\jmath}+n_z\hat{k}$ can be described by the unit quaternion $\mathbf{r}$ as*

$$\mathbf{r} = \cos(\frac{\phi}{2}) + \sin(\frac{\phi}{2})\mathbf{n}.$$

Sequential rotations $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \ldots$ are described by sequential unit quaternion multiplications $\mathbf{r}_1.\mathbf{r}_2.\mathbf{r}_3 \ldots$.

The inverse rotation is given by the quaternion conjugate.

Fact 2.2. Unit quaternion inverse rotation. *The inverse of the rotation given by a unit dual quaternion $\mathbf{r} = \cos(\frac{\phi}{2}) + \sin(\frac{\phi}{2})\mathbf{n}$ is described by its conjugate*

$$\mathbf{r}^* = \cos(\frac{\phi}{2}) - \sin(\frac{\phi}{2})\mathbf{n}.$$

Unit dual quaternions represent rotations (with the nice properties of the unit quaternions) and translations simultaneously, using 8 parameters, as follows.

Fact 2.3. Unit dual quaternion rigid body motion representation. *The rotation $\mathbf{r}$ and the translation $\mathbf{t} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k}$ can be described by the unit dual quaternion $\underline{\mathbf{x}}$ as*

$$\underline{\mathbf{x}} = \mathbf{r} + \frac{1}{2}\epsilon\mathbf{t}\mathbf{r}$$

in which $\epsilon^2 = 0$, but $\epsilon \neq 0$. Similarly to unit quaternions rotations, sequential rigid body motions $\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \underline{\mathbf{x}}_3, \ldots$ are described by sequential unit dual quaternion multiplications $\underline{\mathbf{x}}_1.\underline{\mathbf{x}}_2.\underline{\mathbf{x}}_3 \ldots$.

The inverse of operation of a unit dual quaternion is given by its conjugate in a straightforward operation defined as follows.

Fact 2.4. Unit dual quaternion conjugate. *The inverse motion given by a unit dual quaternion $\underline{\mathbf{x}}$ is given by*

$$\underline{\mathbf{x}}^* = \mathbf{r}^* + \frac{1}{2}\epsilon\mathbf{r}^*\mathbf{t}^*.$$

2.2.2 Unit dual quaternion kinematic control formulation

In order obtain the unit dual quaternion analogue of Whitney's formulation, we substitute $\dot{\vec{x}}$ in (2.9) for its dual quaternion representation, that is

$$\dot{\vec{\theta}} = \mathbf{J}^{inv}\mathrm{vec}(\dot{\underline{\mathbf{x}}}), \tag{2.14}$$

in which $\dot{\underline{\mathbf{x}}}$ is the generalized velocity, i.e. the dual quaternion representation of $\dot{\vec{x}}$. Taking into account the change to the dual quaternion representation, we alter $\mathbf{J}_w$ to $\mathbf{J} \in \mathbb{R}^{8\times n}$ to correctly describe the differential relation between arm coordinates and world coordinates. Note that $\mathbf{J}$ is the dual quaternion analytical Jacobian, which can be found algebraically [16]. Throughout this work it is very important to always remember that although $\mathbf{J} \in \mathbb{R}^{8\times n}$, the maximum rank of the dual quaternion analytical Jacobian $\mathbf{J}$ is 6, as the dual quaternions have eight terms but two constraints.

Following the same steps to obtain (2.14), the unit dual quaternion analogue of (2.11) is given by

$$\dot{\vec{\theta}} = \mathbf{J}^{inv}\mathbf{K}\text{vec}[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(t)]; \tag{2.15}$$

where $\mathbf{K}$ is a positive definite matrix, $\underline{\mathbf{x}}_d$ is the desired pose, and $\underline{\mathbf{x}}(t)$ is the dual quaternion FKM of the manipulator at time t. The stability of (2.15) was proven in Pham's *et al* work in 2010 [19]. Moreover, we can see that the dual quaternion FKM can be directly used in the control equation.

As the control law (2.14) has a clean notation, it will be used throughout the remaining sections whenever an example for a kinematic function is necessary. By no means the techniques shown henceforth are limited to pose control. There are many other kinematic functions which are easily interchangeable. For instance, we could control only the tool translation using the translation Jacobian $\mathbf{J}_p$ [16, p. 77] and a translation velocity $\dot{\mathbf{t}}$ such that

$$\text{vec}(\dot{\mathbf{t}}) = \mathbf{J}_p\dot{\vec{\theta}} \implies \dot{\vec{\theta}} = \mathbf{J}_p^{inv}\text{vec}(\dot{\mathbf{t}}), \tag{2.16}$$

or even only the tool orientation by using the orientation Jacobian $\mathbf{J}_o$, which is the four upper rows of $\mathbf{J}$ [16, p. 77], and an orientation velocity $\dot{\mathbf{o}}$ such that

$$\text{vec}(\dot{\mathbf{o}}) = \mathbf{J}_o\dot{\vec{\theta}} \implies \dot{\vec{\theta}} = \mathbf{J}_o^{inv}\text{vec}(\dot{\mathbf{o}}). \tag{2.17}$$

Concerning computational implementation, we can use the unit dual quaternions in the discrete cases (2.12) and (2.13). In the case that we can control joint velocities, we find

$$\dot{\vec{\theta}}(k) = \mathbf{J}^{inv}\mathbf{K}\text{vec}[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)], \tag{2.18}$$

in which $\underline{\mathbf{x}}_d$ is the desired pose, i.e. the minimal representation of $\vec{x}_d$. Also, $\underline{\mathbf{x}}(k)$ is the current pose, i.e. the dual quaternion representation of $\vec{x}(k)$. In the case we can control joint positions, we find

$$\vec{\theta}(k+1) = \theta(k) + \mathbf{J}^{inv}\tilde{\mathbf{K}}\text{vec}[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)]. \tag{2.19}$$

The formulations using the error $[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)]$ are the result of Pham's *et al* work [19]. An error formulation using the conjugate operation instead of a subtraction is shown in the next section.

2.2.3 Error metrics invariant with respect to coordinate changes

We first recall Whitney's formulation (2.12) given by

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}[\vec{x}_d - \vec{x}(t)].$$

Remember that one interpretation of (2.12) is that we send velocities in the direction $[\vec{x}_d - \vec{x}(t)]$. This is a vector that points from where the tool is to where we want it to go.

The dual quaternion formulation derived from (2.12) is given by

$$\dot{\vec{\theta}} = \mathbf{J}^{inv}\text{vec}[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(t)]. \tag{2.20}$$

Although the same interpretation can be made to some extent, (2.20) does not have the same meaning of (2.12) referring to rigid motions.

That parallel is not right because the reverse motion operation in unit dual quaternion space is not given by a subtraction, but by the multiplication by the conjugate. That is, instead of $\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(t)$, we use

$$\underline{\mathbf{x}}_e = \underline{\mathbf{x}}^*(t)\underline{\mathbf{x}}_d.$$

If we notice that when $\underline{\mathbf{x}}(t) = \underline{\mathbf{x}}_d$, then $\underline{\mathbf{x}}_e = 1$, the frame invariant error metrics $\underline{\mathbf{e}}$ is selected as

$$\underline{\mathbf{e}} = 1 - \underline{\mathbf{x}}_e \tag{2.21}$$

so that when $\underline{\mathbf{x}}(t) \to \underline{\mathbf{x}}_d$, we have $\underline{\mathbf{e}} \to 0$.

By rewriting (2.16) to use the error metrics (2.21), we obtain one result of Figueredo's *et al* work in 2013 [20],

$$\dot{\vec{\theta}} = \mathbf{N}^{inv}\mathbf{K}\text{vec}[1 - \underline{\mathbf{x}}^*(t)\underline{\mathbf{x}}_d],$$

in which $\mathbf{N} \triangleq \bar{\mathbf{H}}(\underline{\mathbf{x}}_d)\mathbf{C}_8\mathbf{J}$. It can be shown that (2.21) is invariant with respect to coordinate changes [20].

It is important to highlight that this error can be used in all formulations described in the following sections for both singularity and redundancy resolutions.

2.3 Workspace singularity resolutions

In this section some of the most used strategies for singularity robustness are reviewed. All techniques are then modified to use the unit dual quaternion formulation, which are the contribution of this section.

Let us start by better understanding what workspace singularities are and the problems they cause. A very handy tool to help our visualization is the *singular value decomposition* (SVD).

Definition 2.5. *Singular value decomposition* [21, p. 412]. For every $\mathbf{A} \in \mathbb{R}^{m\times n}$ of rank r, there are matrices $\mathbf{U} \in \mathbb{R}^{m\times m}$, with $\mathbf{U}\mathbf{U}^T = \mathbf{I}$; $\mathbf{V} \in \mathbb{R}^{n\times n}$, with $\mathbf{V}\mathbf{V}^T = \mathbf{I}$; and a diagonal matrix $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$ such that

$$\mathbf{A} = \mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. The scalars σ_i are the *singular values* of $\mathbf{A}$. The matrix $\mathbf{U}$ is formed by the output singular column vectors $\vec{u_i}$ and $\mathbf{V}$ by the input singular column vectors $\vec{v_i}$.

With the SVD defined, let us remember how to obtain the Moore-Penrose pseudoinverse [21, p. 423].

Definition 2.6. *Moore-Penrose inverse.* Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with a singular value decomposition as in Definition 2.5., its Moore-Penrose inverse is given by $\mathbf{A}^\dagger \in \mathbb{R}^{n \times m}$ such that

$$\mathbf{A}^\dagger = \mathbf{V} \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^T.$$

This inverse has an interesting property. When $m \leq n$, using $\mathbf{A}^\dagger$ gives us the solution with minimal Euclidean norm.

With those definitions, we can now more intuitively understand singularities. Using the Moore-Penrose inverse in (2.14) and remembering that $\mathbf{J} \in \mathbb{R}^{8 \times n}$, we obtain

$$\begin{aligned} \dot{\vec{\theta}} &= \mathbf{J}^\dagger \mathrm{vec}(\dot{\underline{\mathbf{x}}}) \\ &= \mathbf{V} \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^T \mathrm{vec}(\dot{\underline{\mathbf{x}}}) \\ &= \begin{bmatrix} \vec{v}_1 & \cdots & \vec{v}_n \end{bmatrix} \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{u}_1^T \\ \vdots \\ \vec{u}_8^T \end{bmatrix} \mathrm{vec}(\dot{\underline{\mathbf{x}}}) \\ &= \sum_{i=1}^{r} \frac{(\vec{u}_i^T \mathrm{vec}(\dot{\underline{\mathbf{x}}}))}{\sigma_i} \vec{v}_i. \end{aligned} \tag{2.22}$$

As $\mathbf{J}$ is a continuous function of $\vec{\theta}$ and so are its singular values, when the system tends to a singular configuration one or more singular values tend to zero [22]. This effect causes $\dot{\vec{\theta}}$ in (2.22) to tend to infinity, even when $\mathrm{vec}(\dot{\underline{\mathbf{x}}})$ is small. Such increase in joint velocity is undesirable, being dangerous for the robot and any person or object that it might be interacting with.

It is very important to notice that singularities are not exclusive to the resolved motion proposed by Whitney, but it is inherent of the mapping from arm coordinates to world coordinates. In addition, singularities cannot be restricted to isolated regions of the workspace [23].

Even though singularities received some attention from Whitney himself in a later compilation of works [24], it seems the initial singularity avoidance resolution was not correctly considered. In 1983, Klein & Huang [25] concluded from [24] that, as singular configurations can be characterized by high joint velocities, using the Moore-Penrose inverse—remember that this pseudoinverse provides the minimum norm solution—would naturally avoid them. In the following year, Baillieul *et al* refuted Klein & Huang's claim, showing that the local nature of the Moore-Penrose pseudoinverse optimization can generate trajectories arbitrarily close to a singularity [26].

By 1986, Wampler [22] gave kinematic control a push by finding a more efficient computation of $\mathbf{J}$ for the redundant manipulator. This computation is still used in recent textbooks [1, 18, 17]. Not only that, but he also proposed the use of damped least-squares—also known as Levenberg-Marquardt stabilization—to partially solve the singularity problem. We then recall its definition.

Definition 2.7. *Damped least-squares pseudoinverse.* Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a damping parameter $\alpha > 0 \in \mathbb{R}$, the damped least-squares pseudoinverse of $\mathbf{A}$ is given by

$$\mathbf{A}^{+} = \mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T} + \alpha^{2}\mathbf{I})^{-1}.$$

As $(\mathbf{A}\mathbf{A}^{T} + \alpha^{2}\mathbf{I})$ is always a positive definite matrix, it is therefore invertible.

By using the damped least-squares in (2.14), we obtain

$$\begin{aligned} \dot{\vec{\theta}} &= \mathbf{J}^{+}\mathrm{vec}(\underline{\dot{\mathbf{x}}}) \\ &= \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T} + \alpha^{2}\mathbf{I})^{-1}\mathrm{vec}(\underline{\dot{\mathbf{x}}}) \\ &= \sum_{i=1}^{r} \frac{\sigma_i(\vec{u}_i^{T}\,\mathrm{vec}(\underline{\dot{\mathbf{x}}}))}{\sigma_i^{2} + \alpha^{2}}\vec{v}_i. \end{aligned} \tag{2.23}$$

We can then see that instead of increasing indefinitely, the joint velocities now tend to zero as the robot tend to a singular configuration. Not only that, but it can be shown that the increase in joint velocities norm caused by the proximity to a singularity is bounded by $1/(2\alpha)$ [23].

Although resorting to the damped least-squares pseudoinverse protects the manipulator from the dangerous increase in joint velocities caused by singularities, $\dot{\vec{\theta}}$ in (2.19) is no longer the exact solution to $\underline{\dot{\mathbf{x}}}$. This is clear because we are obtaining the solution to a modified $\mathbf{J}$. In most human-robot interaction scenarios however, this may be a negligible side effect.

Even so, it seems unreasonable to always use the damping factor in the inversion. When the manipulator is far from singular configurations, there is no point in obtaining the damped solution. If we can obtain some measure of the closeness to a singularity, we can then decide when such damping should occur. The most accurate measure of proximity to singularities is the same as the measurement of ill conditioning, the condition number [21]. Therefore, we recall its definition from [21, p. 414].

Definition 2.8. *Condition number.* Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ of rank r with singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$; its condition number is given by

$$\kappa = \frac{\sigma_1}{\sigma_r}.$$

Back to the kinematic control problem, the largest the value of $\kappa \geq 1$ for $\mathbf{J}$, the closer the manipulator is to a singularity. To avoid dealing with large numbers, the reciprocal of κ is normally used. We can then define a threshold $\lambda > 0 \in \mathbb{R}$ for $1/\kappa$, such that the damped inverse is only used when $1/\kappa \leq \lambda$. Moreover, the damping factor could begin small and increase to a maximum α_{max} when the manipulator is at the singularity.

This strategy can be improved even further. When one singular value is small, the system is only ill conditioned in one direction, given by the output singular vector related to that singular value. Exploiting this property, we can apply numerical filtering proposed in 1988 by Maciejewski [23] to the damped inverse so that it only damps velocities in the unfeasible direction. Instead

of using the condition number, Maciejewski opted for the minimum singular value for its similar performance [27] and cheaper computation. Applying all this to our system (2.14), we get

$$\dot{\vec{\theta}} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \alpha^2 \vec{u}_6 \vec{u}_6^T)^{-1}\text{vec}(\underline{\dot{\mathbf{x}}}),$$

$$\alpha^2 = \begin{cases} 0 & \sigma_6 \geq \lambda \\ (1 - (\frac{\sigma_6}{\lambda})^2)\alpha_{max}^2 & \sigma_6 < \lambda \end{cases}. \tag{2.24}$$

When at least two singular values are small, we can apply an isotropic damping $\beta > 0 \in \mathbb{R}$ to damp all directions. The final version of this controller is the result of Chiaverini's 1997 work [28, p. 401], given by

$$\dot{\vec{\theta}} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \beta^2\mathbf{I} + \alpha^2 \vec{u}_6 \vec{u}_6^T)^{-1}\text{vec}(\underline{\dot{\mathbf{x}}}), \tag{2.25}$$

with α^2 as in (2.24).

All the discussed solutions are applicable to one human-robot interaction scenario or another, depending on the safety restrictions and the computational power available. There are settings (e.g. medical applications), however, in which neither high joint velocities near singular configurations given by the Moore-Penrose inverse nor reconstruction errors given by the damped inverse are acceptable. In those cases, the most reasonable solution is to use a redundant manipulator and exploit its extra degrees of freedom to evade kinematic singularities.

2.4 Kinematic redundancy resolutions

In this section we review the most common strategies for redundancy resolution. These schemes are modified and applied using the unit dual quaternion framework.

Redundancy is easier to visualize than workspace singularities. Consider a linear system given by

$$\mathbf{J}\dot{\vec{\theta}} = \text{vec}(\underline{\dot{\mathbf{x}}}), \tag{2.26}$$

with $\mathbf{J} \in \mathbb{R}^{m \times n}$. If we have $m \leq n$ and $\text{rank}(\mathbf{J}) = m$, there may be infinite solutions for a given $\underline{\dot{\mathbf{x}}}$. In pose control terms, the manipulator is considered redundant if it has more than 6 degrees of freedom.

This issue was tackled by Whitney himself in his pioneering work in 1969 [15]. He proposed the use of a pseudoinverse to obtain the optimal solution in a least-squares sense such as in (2.22).

Although his solution is interesting, redundancy can have other applications. For instance, there are tasks more easily described as two objectives. Examples would be moving our hand while preventing our shoulder from hitting a surface (controlling two points simultaneously in the kinematic chain); or trying to grab a cup while avoiding hitting our hand on an obstacle.

For the wide range of possible tasks describable by multiple objectives, redundancy exploitation became the theme of many works in the 1980's. One technique proposed by Sciavicco & Siciliano in 1988 is the augmented Jacobian algorithm [29]. We first define a vector $\vec{c}(\theta)$ describing the

secondary objective only in terms of $\vec{\theta}$. Then, we observe the relationship between the secondary objective derivative $\dot{\vec{c}}(\theta)$ and the joint variables

$$\frac{\partial \vec{c}(\theta)}{\partial \vec{\theta}} \dot{\vec{\theta}} = \mathbf{J}_c \dot{\vec{\theta}} = \dot{\vec{c}}(\theta). \tag{2.27}$$

If we are able to find the secondary objective gradient $\mathbf{J}_c$ we can write the augmented Jacobian $\mathbf{J}_{aug}$ as

$$\mathbf{J}_{aug} \dot{\vec{\theta}} = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_c \end{bmatrix} \dot{\vec{\theta}} = \begin{bmatrix} \text{vec}(\dot{\underline{\mathbf{x}}}) \\ \dot{\vec{c}}(\theta) \end{bmatrix}.$$

This formulation allows having a secondary objective with the same priority as the first. However, when any row in $\mathbf{J}_c$ is linearly dependent from those of $\mathbf{J}$ we face algorithmic singularities. Those singularities are the mathematical manifestation of simultaneously trying to reach two conflicting objectives with the same priority. Of course there is no problem whenever we can find a linearly independent secondary objective, but that is no easy task [28].

Instead of giving both objectives the same priority, we can delegate different levels for each one. Liegeois [30] discussed how to do so in his work in in 1977. The idea is to project the secondary objective in the nullspace of the Jacobian of the primary objective. One such projector is

$$\mathbf{P} = (\mathbf{I} - \mathbf{J}^{inv}\mathbf{J}), \tag{2.28}$$

in which $\mathbf{J}^{inv}$ is one of the inverses discussed in section 2.3. To maintain pose control as the first priority and (2.27) as a secondary objective, our control law becomes

$$\dot{\vec{\theta}} = \mathbf{J}^{inv}\text{vec}(\dot{\underline{\mathbf{x}}}) + \mathbf{P}\mathbf{J}_c^{inv}\dot{\vec{c}}(\theta). \tag{2.29}$$

The use of $\mathbf{P}$ causes the system to use motions that do not disturb the primary objective to perform the secondary one. This is very interesting and there is a myriad of available secondary objective options, such as joint limit avoidance, obstacle avoidance and singularity avoidance. Some examples are shown in [1, p. 126].

Making an example of our own, consider controlling the current tool translation $\mathbf{t}$ as the primary objective, and the tool orientation $\mathbf{o}$ as the second one. Such strategy is useful when using a manipulator with less than six degrees of freedom. The resulting control law is

$$\dot{\vec{\theta}} = \mathbf{J}_p^{inv}\mathbf{K}_p\text{vec}(\mathbf{t}_d - \mathbf{t}) + (\mathbf{I} - \mathbf{J}_p^{inv}\mathbf{J}_p)\mathbf{J}_o^{inv}\mathbf{K}_o\text{vec}(\mathbf{o}_d - \mathbf{o}), \tag{2.30}$$

in which $\mathbf{t}_d$ and $\mathbf{o}_d$ are the desired translation and orientation, respectively. Moreover, $\mathbf{K}_p$ and $\mathbf{K}_o$ are the gain matrices for each objective.

Although there are many interesting applications for redundancy, we focus now on singularity avoidance. This will be useful when discussing medical applications.

There are some approaches related to evading singular configurations known *a priori*. That is, if we know that a certain configuration $\vec{\theta}_s$ is singular, we may write $\vec{c}(\theta)$ as a configuration evasion function. For example, Chiaverini *et al* in 1990 [31] described the kinematic singularity set of a seven joint manipulator and derived its configuration evasion function. Such solution is

very interesting when those configurations are known, because $\vec{c}(\theta)$ becomes a simple and cheap function to compute.

Related to this approach, a recent work by Bohigas *et al* in 2013 [32] found how to numerically compute the kinematic singularities of any given nonreduntant manipulator. For those, it is an interesting choice to obtain such singular configurations and use a cheap function to evade them. However, the extension of this framework for redundant manipulators is still a work in progress.

Despite the advantage of having a smaller computational cost, it is rather restrictive to have to find a function for each manipulator geometry. A more generic approach may be more interesting.

We can now introduce the very well known work by Yoshikawa in 1985 [33], where he introduced the concept of manipulability measure $w \in \mathbb{R}^+$ given by

$$w \triangleq \sqrt{\det[\mathbf{J}_w \mathbf{J}_w^T]}. \tag{2.31}$$

If we perform some basic operations

$$\begin{aligned} w &= \sqrt{\det[\mathbf{J}_w \mathbf{J}_w^T]} \\ &= \sqrt{\det[\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T]} \\ &= \sqrt{\det[\mathbf{U}\mathbf{D}\mathbf{D}\mathbf{U}^T]}, \end{aligned}$$

as the determinant is invariant to similarity transformations (simple corollary of the distributive property of the determinant [21, p. 467]) we have

$$\begin{aligned} w &= \sqrt{\det[\mathbf{D}\mathbf{D}]} \\ &= \sqrt{\sigma_1^2.\sigma_2^2 \dots \sigma_r^2} \\ \therefore w &= \sigma_1.\sigma_2 \dots \sigma_r. \end{aligned}$$

We can see that w is the product of the singular values of $\mathbf{J}_w$ or zero when $\text{rank}(\mathbf{J}_w) < r$.

Yoshikawa stated that w is equal to the volume of the *manipulability ellipsoid* and that it could be a good means for the analysis, design and control of robot manipulators. This measure was exceptionally attractive because the determinant in some arm geometries can be calculated explicitly as a function of joint angles and the gradient is easily obtained from it [27]. Yoshikawa's dexterity measure received a lot of visibility in manipulator research, in works related to singularity avoidance [34, 35, 36], manipulator design and workspace optimization. Moreover, there is a reserved space for manipulability discussion in most robotics textbooks [1, 17, 18].

Note that all formulations described in this section and the next are local optimizations and, therefore, are limited. Even so, they perform well in the applications described in this work. Although global optimization is an interesting subject, it will not be discussed.

In the next section, some issues involving w are discussed. Also, a numeric method for computing the gradient of another dexterity measure is exposed, which is a contribution of this work.

2.4.1 Numerical calculation of secondary objective gradients

Although w is the product of the singular values, the determinant is not a good measure of closeness to singularities [21, p. 466]. This issue was observed in 1987 by Klein & Blaho [27], who also noted that the only reliable measure of closeness to singularities is the condition number or even the minimum singular value. For instance, it is noticeable that for large variations of the condition number, w may stay constant. However, the condition number cannot be expressed analytically as a function of joint angles [27]. It seems that mostly for this reason this measure did not become so widespread.

Note that for the unit dual quaternion Jacobian $\mathbf{J}$, the manipulability as defined in (2.31) is not applicable. As the maximum rank of $\mathbf{J}$ is 6 and $\mathbf{J}\mathbf{J}^T \in \mathbb{R}^{8\times 8}$, $\det[\mathbf{J}\mathbf{J}^T] = 0 \ \forall \ \vec{\theta}$. For such measure to be useful, the unit dual quaternion manipulability measure w_{dq} is defined as

$$w_{dq} \triangleq \sigma_1.\sigma_2.\ldots.\sigma_6.$$

However, as the singular values should be found for w_{dq} instead of the determinant as in w, the analytical solution is lost. Hence, the condition number is simpler and seems more suitable for the unit dual quaternion case.

In order to use the condition number in the task-priority formulation in a way similar to (2.29), we first need to find its gradient in relation to θ. Although numerically finding the condition number is easy, as mentioned before its gradient is not. We then focus on numerically obtaining the derivative of the condition number of $\mathbf{J}$ in relation to each θ_i, for $i = 1, \ldots, n$. For this purpose, we obtain the singular value decomposition of $\mathbf{J}$ as

$$\mathbf{J} = \mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^T,$$

in which $\mathbf{U} \in \mathbb{R}^{8\times 8}$ such that $\mathbf{U}\mathbf{U}^T = \mathbf{I}^{8\times 8}$; $\mathbf{V} \in \mathbb{R}^{n\times n}$ such that $\mathbf{V}\mathbf{V}^T = \mathbf{I}^{n\times n}$; and $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_6)$ in which $\sigma_1, \sigma_2, \ldots, \sigma_6$ are the singular values of $\mathbf{J}$.

The condition number of $\mathbf{J}$, $\kappa(\mathbf{J}) \geq 1$, is then defined as

$$\kappa(\mathbf{J}) = \frac{\sigma_1}{\sigma_6}$$

and measures the degree of distortion of the manipulability ellipsoid. For a cleaner notation, consider $\kappa(\mathbf{J}) = \kappa(\theta)$ with no loss in meaning, since $\mathbf{J}$ is a function of θ. The interesting property of the condition number is that the bigger the value of $\kappa(\theta)$, the closer $\mathbf{J}$ is to a singularity[4]. Also, when $\kappa(\theta) = 1$, $\mathbf{J}$ has the best possible theoretical conditioning (it is an orthogonal matrix).

With those definitions, our objective becomes the numerical computation of the partial derivatives

$$\frac{\partial(\kappa(\theta))}{\partial(\theta_i)},$$

[4]To avoid problems with the discontinuity when $\sigma_6 = 0$, $1/\kappa(\theta)$ is usually preferred over $\kappa(\theta)$. In any case, the discussion that follows applies to both cases.

for each θ_i. First, we highlight that in a physical manipulator, each and every joint θ_i is driven by a motor with position sensing. Independently of how this measurement is done, it has a precision given by λ_i. That is, any reading only guarantees that the joint is positioned within an interval $\theta_i \pm \lambda_i$.

Then, we define some selector vectors $s_i \in \mathbb{R}^{n\times 1}$ such that

$$\begin{aligned} s_1 &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}^T, \\ s_2 &= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T, \\ &\vdots \\ s_n &= \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \end{bmatrix}^T. \end{aligned}$$

Consider any real number $\delta_i \leq \lambda_i$. To compute the partial derivatives we can use finite differences given by

$$\frac{\partial(\kappa(\theta))}{\partial(\theta_i)} \approx \frac{\kappa(\theta + s_i.\delta_i) - \kappa(\theta)}{\delta_i}. \tag{2.32}$$

This requires the computation of $\kappa(.)$ $n+1$ times. By using (2.32) we are obtaining the expected value of $\partial(\kappa(\theta))/\partial(\theta_i)$ in the interval $[\theta_i, \theta_i + \delta_i]$, considering that any value inside the interval is equally likely to be the real joint position. Using this gradient, singularity evasion can be projected in the nullspace of the first task for singularity avoidance on any serial link manipulator.

2.5 Kinematic controllers evaluation: simulations

In order to give the reader an illustrative example of the controllers functionality, a simulated experiment was devised using a software framework composed of ROS[5], OpenRAVE[6], and DQ_robotics[7].

In this example, the computational implementations of four controllers described in this chapter are evaluated in a pose control task, with respect to singularity robustness and convergence time. The first controller has no singularity robustness by directly using the Moore-Penrose inverse (2.22), the second controller uses the damped inverse (2.23), the third uses the damped numerical filtered inverse (2.25), and the last is the task-priority controller using damped inverses (2.30).

The selected manipulator was a simulated version of the 7 degrees-of-freedom Schunk LWA3. The task, to be performed by all controllers, was to move from an initial singular configuration

$$\theta = \begin{bmatrix} 0 & \frac{\pi}{2} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

[5]ROS is the Robot Operating System which is a collection of software tools and libraries to aid in robotics related software development. Source: http://www.ros.org/

[6]OpenRAVE provides a simulated environment for testing robotics related applications. Source: http://openrave.org/

[7]DQ_robotics is a software library in which kinematic control algorithms related to dual quaternions are implemented. Source: http://sourceforge.net/projects/dqrobotics/

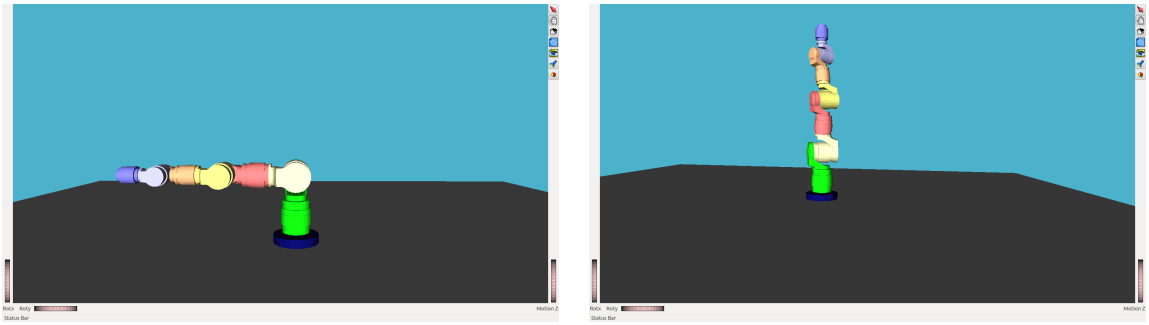

Figure 2.1: Manipulator configurations during the simulations: initial configuration (*left*) and final configuration (*right*).

to the final singular end effector pose

$$\underline{\mathbf{p}}_d = \hat{k} - \varepsilon \frac{1.305}{2}.$$

Figure 2.1 shows the initial and final configurations of the robotic manipulator. It is important to highlight that the initial and final configurations were chosen to be inside singular regions.

The controllers are considered to have reached the target pose when the selected error measurements are within a 10^{-2} tolerance. In the pose controller with damped pseudoinverse, in the one with Moore-Penrose pseudoinverse, and in the one with numerical filtering, this tolerance is

$$\left\| \text{vec}\left(\underline{\mathbf{p}}_{eff} - \underline{\mathbf{p}}_d\right) \right\| < 10^{-2}$$

and, in the priority controller, this tolerance is

$$\left\| \text{vec}\left(\mathbf{t}_{eff} - \mathbf{t}_d\right) \right\| < 10^{-2}.$$

During the simulations, the sample rate used in the simulation was $100Hz$. Also, the scalar gains for the controllers based on (2.23), (2.25), and the controller with Moore-Penrose pseudoinverse were selected as $K = 0.35$, and the gains for the task-priority controller were chosen as $K_p = 0.25$ and $K_o = 0.05$.

The damping factors were $\lambda^2 = 0.1$ in the pose controller with damped pseudoinverse; $\lambda^2 = 0.1$, $\beta^2 = 0.01$ and $\mu = 0.1$ in the numerical filtered controller, and $\lambda^2_{translation} = 0.15$, $\lambda^2_{rotation} = 0.4$ in the task-priority controller with translation priority. In conjunction with the gains, the damping factors were chosen as to make the upper-bound for the joint velocities equal to $0.5rad/s$, considering that the maximum angular velocity of any joint of the Schunk robot[8] is approximately 0.8 rad/s. Figure 2.2 presents the performances of each controller.

2.5.1 Results & discussion

The results of the controller with damped pseudoinverse and of the numerical filtered controller are hard to distinguish. As the damping is approximately the same for both controllers, the

[8] http://www.schunk.com/schunk_files/attachments/PRL_gesamt_EN.pdf

effectiveness in pose control is slightly noticeable from the difference in the convergence time. While the numerical filtered controller converged in 53.51 s, the controller with damped pseudoinverse converged in 53.54 s. This small difference can be explained by the fact that the numerical filtered controller gives a more accurate inverse far from singular configurations.

On the other hand, the task-priority controller with translation priority had a different behavior. As we chose smaller gains for this controller in order to obey the constraint of joint velocities (i.e., $< 0.5\,rad/s$), it converged slower than the damped and the numerical filtered controllers (approximately 80 s). In addition, the prioritized scheme was also robust to the singular configurations, suggesting that the partitioning of the Jacobian matrix into smaller matrices improved the numerical conditioning of the whole system. However, aside from the empirical results we did not perform any theoretical study to support this claim. As a consequence, in future works we intend to investigate the supposedly robustness of the prioritized scheme with dual quaternions from a theoretical standpoint.

The last controller, which makes use of a classic Moore-Penrose pseudoinverse, shows the negative effects of the ill-conditioning near singular configurations. Due to the singularity at the initial pose, some joints reached velocities of 28175.7 rad/s in the first iteration, which is completely unfeasible in practice. This caused an impulse response in Figure 2.2 at 0 s. The manipulator was only able to converge because dynamical aspects were not taken into consideration in the simulation and because we used low gains (for larger gains this usually leads to complete instability). In this way, the manipulator was able to recover itself once outside the singular configuration. However, in practice these extremely large velocities can cause damage to the robot.

2.6 Conclusion

This chapter started with the introduction of Whitney's resolved motion rate control. After rewriting his framework using unit dual quaternions, more advanced techniques for singularity robustness and redundancy exploitation were also adapted.

Now that all relevant works and concepts concerning singularity robustness and redundancy resolutions have been discussed, they will be introduced in the medical robotics context in the following chapter.

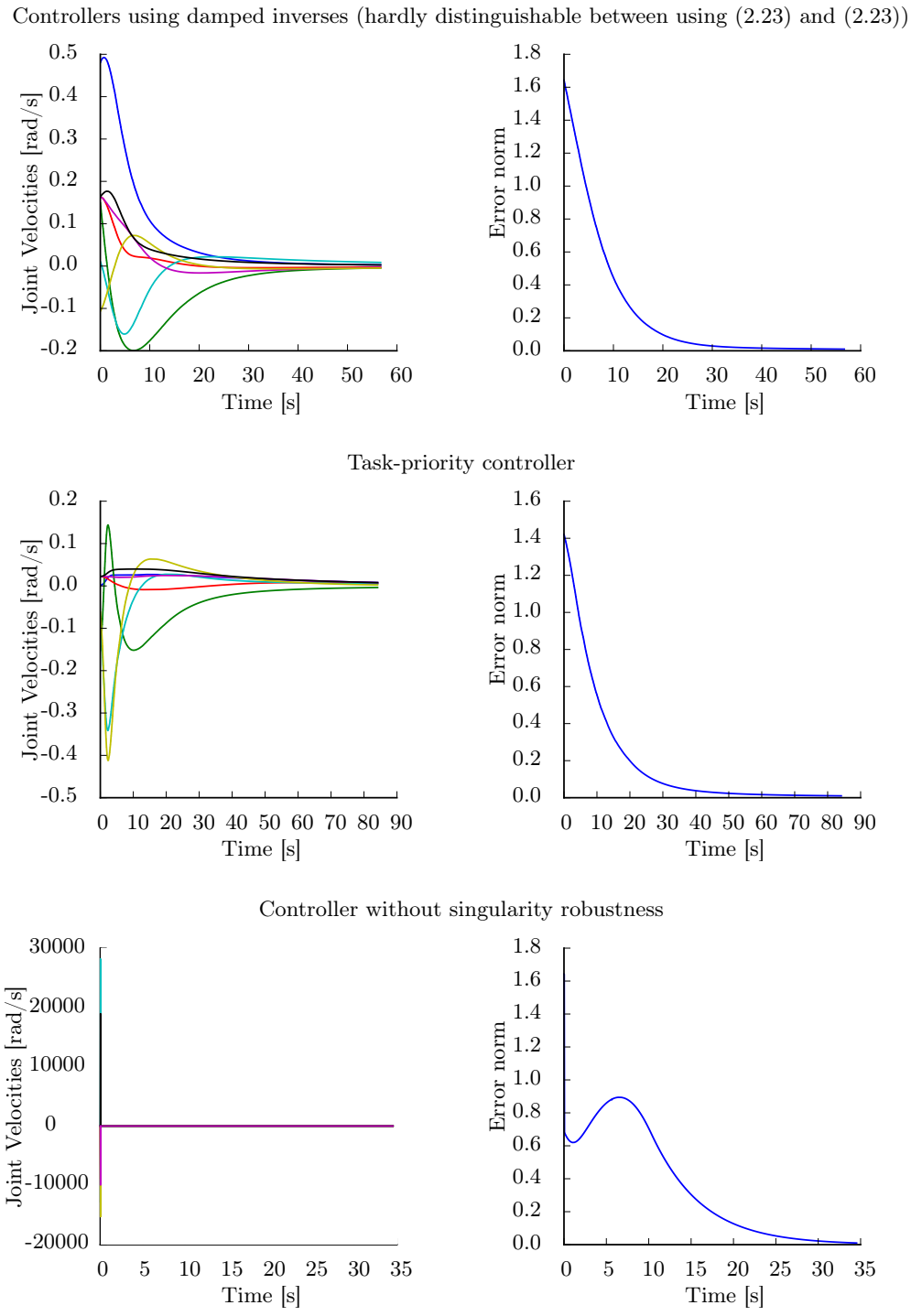

Figure 2.2: Comparison between the performances of the controllers. Note that in the controller without singularity robustness (*bottom left)*, the robot joint velocities reach hazardous values.

Chapter 3

Robotic Laparoscopy

"As to diseases, make a habit of two things– to help, or at least to do no harm."
–Hippocrates, in Epidemics, 1:11

Now that kinematic control of manipulators was reviewed, we begin by exploring the state of the art in robotic laparoscopy in section 3.1. Afterward, in section 3.2, a novel method for laparoscopic control for any redundant manipulator is shown, which is the main contribution of this work. Then, in section 3.5, the controller is used in a simulated endoscope positioning task, so it can be evaluated in terms of variables such as trocar point safety and operational intuitivity.

3.1 State of the art

Works related to laparoscopic surgery are mostly concerned with safely maneuvering a tool while considering the constraints imposed by the *trocar point*. Robotic devices need to generate a virtual pivoting point, called the *remote center of motion* (RCM). During the surgical procedure, it is necessary that the trocar point, on the patient, coincides with the RCM, on the device.

In the literature, the possible forms of RCM generation using robots are commonly divided into three distinct techniques that use: (1) robots with special mechanical devices, (2) robots with passive wrists, or (3) fully actuated robots.

3.1.1 Robots with special mechanical devices

The first form of RCM generation is also the most common. It comprises robots with special mechanical structures that inherently generate the remote pivoting point for the laparoscopic tool. For instance, a simplified mechanical structure with those characteristics is shown in Figure 3.1 (left).

By positioning the RCM of the structure to coincide with the trocar point, the patient is mechanically protected and the accuracy of the restrained motion of the tool is assured. Robots

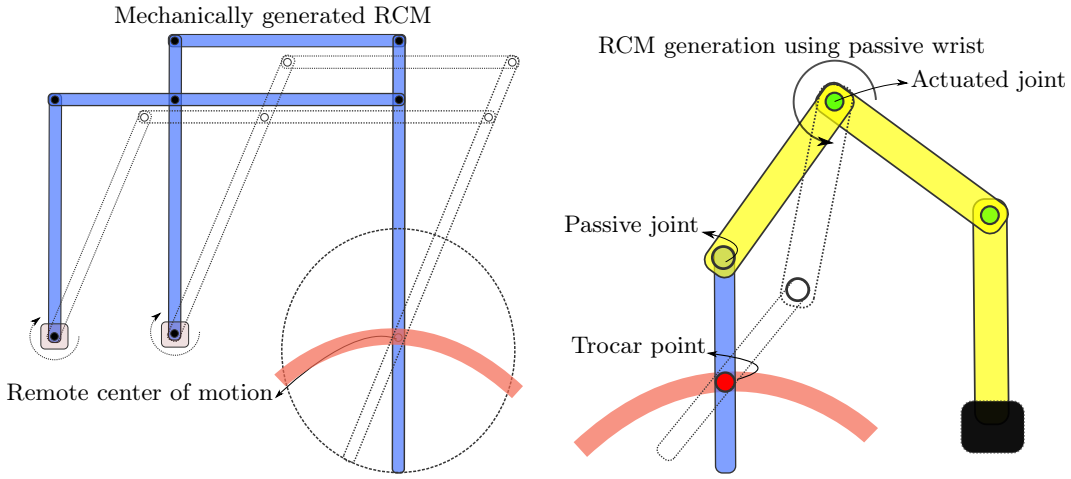

Figure 3.1: Example device for mechanical RCM generation (*left*). Example robot with a passive last joint generating the RCM (*right*).

with those characteristics focus on a dedicated surgical technique or treatment, like the daVinci from Intuitive Surgical [37]. They are precise and reliable, but have the disadvantage of being high cost and have restricted use to the specific application they have been developed for. Works using this technique [38, 37] mostly focus on the mechanical design of the specialized structures; and the controller designs are simple and case dependent, as the laparoscopic constraints are solved mechanically.

3.1.2 Robots with passive wrists

The second technique for RCM generation uses passive wrists, i.e. some of the last robot joints are encoded but comply to any external force applied on them. By using such special wrists to hold a laparoscopic tool, the RCM is generated by the small reaction forces exerted from the trocar point, passively providing safety for the patient. A simplified robot with a passive wrist is shown in Figure 3.1 (right).

Systems that use this technique require an external device to constantly estimate which point of the tool is in contact with the trocar point, so that the passive joints can be correctly positioned using the actuated joints. Therefore, any imprecisions on the estimation of the trocar point position decrease the accuracy of the tool positioning. Moreover, as the physical trocar point is not an ideal pivoting point, more inaccuracies are added in this design [39]. For those reasons, such devices are mostly restricted to camera control tasks [13, 39], in which positioning inaccuracies only result in the bad centering of the camera image.

Given their challenges, there has been much effort into designing and developing control techniques to perform laparoscopic movements with passive wrist mechanisms.

For instance, Funda *et al* [39] used a ceiling-mounted robot with a passive 2 degree-of-freedom wrist in a laparoscopy scenario. Their controller used a constrained cartesian control based on an extended Jacobian technique, which may be extensible to fully actuated manipulators. However,

the different motions expected in an endoscope (translation, zoom and rotation) are atomized in three different controllers, with no clear description of how they could be used simultaneously.

Munoz *et al.* [13] proposed a cartesian controller with an adaptative proportional-integral control law. This controller moves the endoscope while compensating for uncertainties in the trocar point estimation, but the formulation is restricted to a specific manipulator geometry.

Ortmaier and Hirzinger [40] designed a cartesian controller for the AESOP, while also estimating the trocar positioning. Both techniques may be applicable to other passive wrist robots, but not to the fully actuated manipulator. Many other works, especially in the design of such systems are reported in [41].

3.1.3 Fully actuated robots

The last RCM generation technique uses fully actuated robots, which are intended to be adaptable to various existing and future medical procedures, like the newly proposed MIRO system from DLR[1] [12]. In such platforms, the RCM is programmed according to the intended surgical procedure and obtained under coordinated control of multiple joints. Such motions can be achieved under a large variety of high degree-of-freedom robots. It has advantages of pivot flexibility, increased maneuverability and overall versatility. Moreover, as all the robot joints are actuated, its precision is not reliant on the estimations of the trocar point position.

Despite those advantages, this approach is traditionally less favored by clinicians because of its reliance on software means to maintain the required RCM constraint. Generating the RCM by software is seen as a possibly fatal source of error and every control algorithm in this category must prove itself safe enough. The works in this category mostly focus on safe RCM generation strategies and how to deal with possible workspace singularities.

Concerning dexterity, many works try optimizing the robot workspace with metrics such as manipulability (described in section 2.3 on page 16) aiming the safest possible singularity wise work volume.

For instance, the use of an industrial manipulator for laparoscopic surgery has been evaluated in [42]. In this work, they proposed a manipulability metric to choose a proper operational area in the robot task space. Another research group [43] has also proposed optimization techniques to find the best position for the RCM with respect to the robot base.

Many research groups have proposed control methods for fully actuated robots used on laparoscopic scenarios; either platform dependent approaches or more general methods applicable to redundant manipulators.

Regarding platform specific approaches, Mayer *et al* [44] proposed a joint controller method that uses analytical inverse kinematics specific to their manipulator geometry. More recently, another platform-dependent alternative has been proposed for a parallel manipulator in [45].

Concerning works applicable to any manipulator geometry, Michelin *et al.* [46] proposed a

[1] DLR is the aeronautics and space research centre of the Federal Republic of Germany.

dynamic task-posture scheme for torque-controlled robots which was evaluated in a 3 degrees-of-freedom simulated planar robot. This scheme controls the tool positioning as a first objective while the RCM constraint is treated as a secondary one, by using a nullspace projection scheme similar to (2.29). As mentioned in section 2.4 on page 19, while such control law tries reaching the secondary objective, it has no guarantee of convergence contrary Michelin *et al* claims. Therefore, the RCM generation should not[2] be a secondary objective in (2.29) and related formulations.

Another general control method has been proposed by Azimian *et al* [47], which is based on a Jacobian task priority method for redundant robots. This strategy assures the RCM constraint as a first objective, while the tool positioning is considered a secondary one. Such controller was evaluated in a simulated task using a 6 degrees-of-freedom robot. However, as the tool positioning was considered a secondary objective, it showed considerable deviation from the desired trajectory. Such approach shows a comparable flaw to Michelin *et al* [46] approach, where a vital objective was considered secondary. Moreover, only the positioning of the tool-tip is controlled, which means that its orientation is left as a free variable in the inverse kinematics minimization. Thus, not only the tool-tip has tracking errors but it also has arbitrary rotations around its axis. Although this last issue can be dealt with by inserting tool orientation in the control loop, the tool-tip pose will still be only treated as a secondary objective and, as such, will not have any guarantee of convergence.

3.2 Novel endoscope control using redundant serial link manipulators[3]

In our setup, a manipulator was acquired in order to further research RCM generation using fully actuated robots. Our interest is not to design a controller only for this manipulator, but one that may be applied to any serial link manipulator geometry. Such formulation would ease the comparison between different manipulator geometries.

From all the related work reviewed in the last section, the ones from Azimian *et al* [47] and Michelin *et al.* [46] come closer to our needs; however, we need to find a solution that does not share their drawbacks. Recapitulating, the former treats tool-tip control as the first objective, while the RCM maintenance is considered a secondary goal; which results in arbitrarily high RCM errors. The latter treats the RCM constraint as the first objective, while the tool-tip control is the second; which results in arbitrarily high deviations from the desired tool trajectory.

Therefore, we must find a strategy in which there is a reasonable trade-off between tool-tip positioning and RCM error. This motivates the development of another controller, which is the main contribution of this work.

A last issue commonly neglected by other works is how the commands to move an endoscope are

[2] In the case that the secondary objective is orthogonal to the first, such strategy should work. However, in this case the nullspace projection would be unnecessary.

[3] This section requires knowledge about basic dual quaternion algebra. If necessary, notation and operator definitions can be seen on appendix I.1.

sent to the controller by using an user interface (e.g. voice, haptic device, joystick etc). This subject is often left unnoticed, but surgeons usually have strong expectations about system simplicity and performance, while presenting low tolerance for interfaces that impede their work.

Therefore, for camera control we need a framework that can receive simple (1) up-down, (2) right-left, (3) around-itself rotation and (4) in-out commands; which are enough to control any angled endoscope. From those, the framework should perform the reference tool tip displacements, while dealing with the RCM restriction and mirrored movements internally.

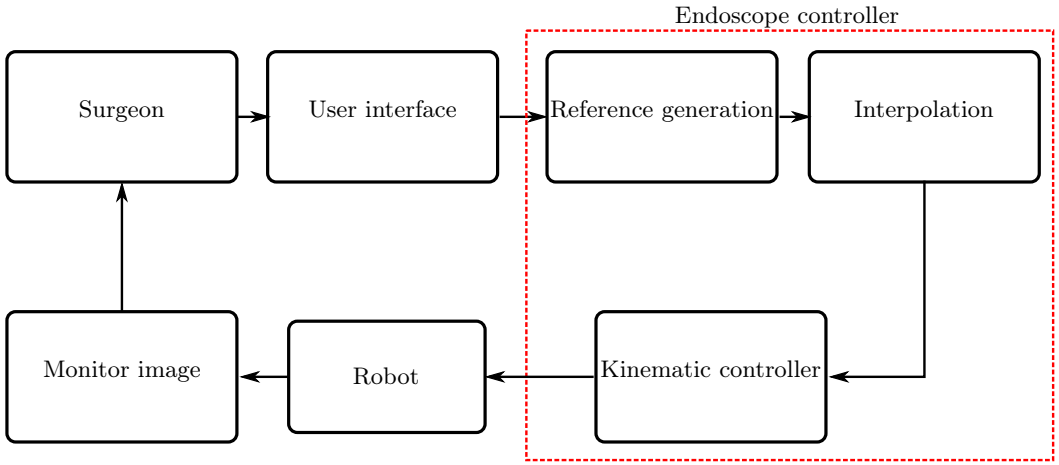

Figure 3.2: Laparoscopy control system overview. In this section the elements inside the red rectangle will be described.

We now go back to our laparoscopy scenario as described in section 1.2. At the beginning of the robot-aided procedure, the patient is laid face-up in the operating table. After their abdomen have been insuflated, the trocar is placed. The robot is already positioned near the patient. With the robot in compliant mode, the surgeon manually guides the endoscope and inserts it through the trocar.

The surgeon has a monitor to see the camera images directly from the patient's abdominal cavity and an user interface to give endoscope movement commands.

A diagram showing the principal elements of the operation room is shown in Figure 3.2. For a better description of the devised endoscope controlling system, it was divided into three elements: reference generation, interpolation and the kinematic controller.

3.2.1 Reference generation

We first recall that a manipulator robot is described by the frame transformations between its links, summarized in its DH parameters. Now that the robot has an endoscope attached on its last link, this should be accounted as a new link in the robot kinematic chain.

As we are not restricting our development to a specific robot configuration, we cannot specify an unique last frame transformation. Instead, we define the subsequent convention: immediately after the initial setup, the translation of the end effector must be given by the point on the endoscope

that has the same translation as the trocar. Its z-axis is given by the endoscope shaft pointing inside the patient, its y-axis in the direction of the endoscope camera "up" direction, and its x-axis selected to finish the right hand reference frame.

By following this convention, the end effector will initially have the same translation as the trocar, which is where the RCM will be generated. We then define the RCM pose as $\underline{\mathbf{x}}_{\text{rcm}} \triangleq \underline{\mathbf{x}}(0)$.

As the last link transformation is dependent on the trocar point translation, a reasonable question is how the relative positioning of the endoscope and the trocar point can be found. One approach is to mark one known point along the endoscope shaft, so that point should coincide with the trocar point when manually placing the endoscope. The inherent inaccuracy of manual placement can be countered by the flexibility of the patient's skin. An external device can be used to grant more precision in the point coordinates if necessary. With a visual marker placed in the trocar point and another along the endoscope shaft, the coinciding point can be found with as good precision as the tracking device allows. Note that, in contrast with passive wrist devices, only one measure is needed and there is no need for *continuous* estimation of the trocar point.

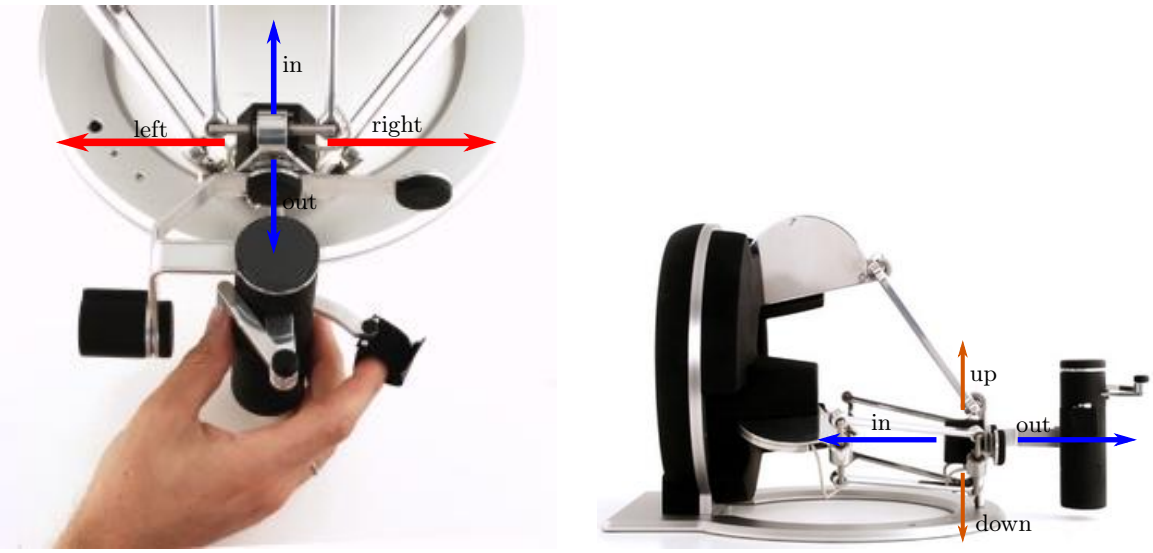

Figure 3.3: Omega 7 views. Top view (*left*). Side view (*right*). (source: ForceDimension)

With $\underline{\mathbf{x}}_{\text{rcm}}$ following the given convention, it is easy to translate the desired camera movements into end effector motions. Considering any user interface, at each instant k the $u_\alpha(k)$ (up-down), $u_\beta(k)$ (right-left), $u_\gamma(k)$ (around-itself), and $u_z(k)$ (in-out) commands are obtained each from an independent degree of freedom in the interface. For instance, suppose the surgeon uses an Omega 7 haptic device to control the endoscope in relation to the images shown in the monitor. In Figure 3.3, it is shown which hand movements the surgeon should perform.

The u functions provide any necessary filtering of the raw signal sent by the device; such as bias, scaling, and so on. It is paramount that $u(0) = 0$ for all inputs.

1. To perform an up-down camera movement, we rotate the end effector around its x-axis, with the quaternion $\mathbf{r}_x(k) = \cos(u_\alpha(k)/2) + \hat{\imath}\sin(u_\alpha(k)/2)$ (positive $u_\alpha(k)$ moves "down").

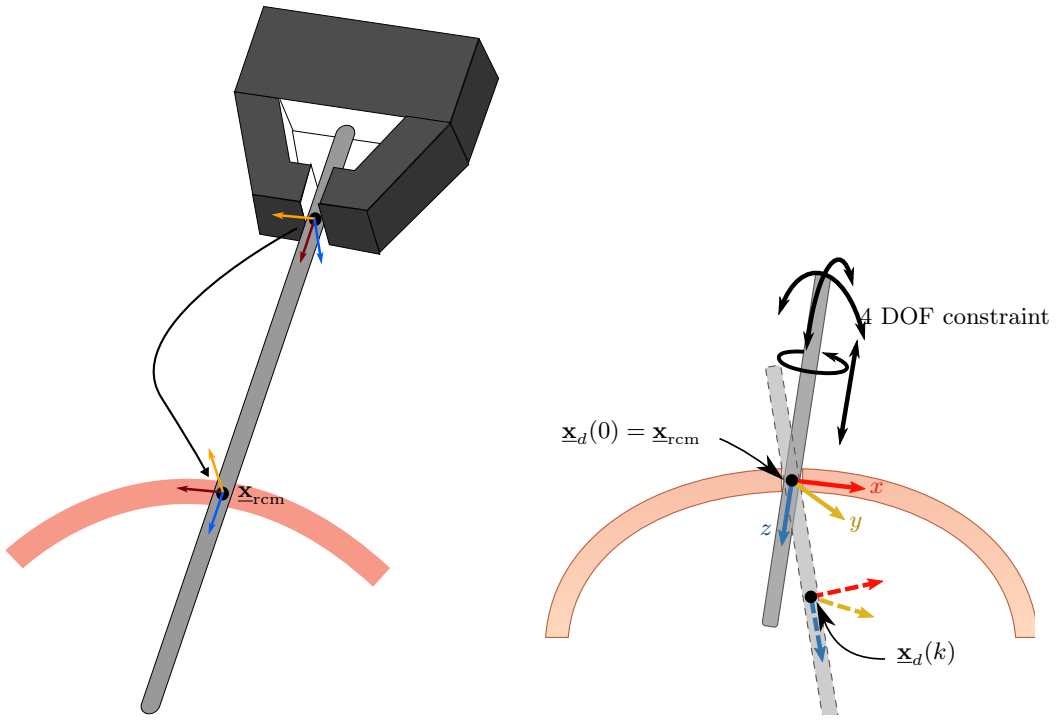

Figure 3.4: Kinematic chain last frame transformation (*left*). End effector reference motion (*right*).

2. The right-left camera motion is made by the rotation of end effector around its y-axis, using $\mathbf{r}_y(k) = \cos(u_\beta(k)/2) + \hat{\jmath}\sin(u_\beta(k)/2)$ (positive $u_\beta(k)$ moves "left").

3. For angled endoscopes, the around-itself camera motion is obtained by rotating the end effector round its z-axis, using $\mathbf{r}_z(k) = \cos(u_\gamma(k)/2) + \hat{k}\sin(u_\gamma(k)/2)$ (positive $u_\gamma(k)$ moves clock-wise).

4. Finally, the in-out camera motion is given by a translation along the end effector z-axis, using $\underline{\mathbf{t}}_z(k) = 1 + (1/2)\epsilon\hat{k}u_z(k)$ (positive $u_z(k)$ moves "in").

Therefore, in any given procedure time, the desired end effector pose is given by

$$\underline{\mathbf{x}}_d(k) = \underline{\mathbf{x}}_{\text{rcm}}\mathbf{r}_x(k)\mathbf{r}_y(k)\mathbf{r}_z(k)\underline{\mathbf{t}}_z(k). \tag{3.1}$$

As we selected $\underline{\mathbf{x}}_{\text{rcm}}$ coinciding with the trocar point and restricted the movements to only the four degrees of freedom available in a pivoting point, any reference generated this way also meets the trocar constraint.

3.2.2 Kinematic control scheme

With reference generation solved, we now need to use them in the kinematic controller. For the sake of simplicity, we restrict the discussion to a manipulator whose joint positions can be controlled. However, as seen in section 2.2, it is very simple to find the joint velocity controlling equations. We start with the closed loop dual quaternion pose controller with invariant error

metrics (section 2.2.3)

$$\vec{\theta}(k+1) = \vec{\theta}(k) + \mathbf{K}\mathbf{N}^{inv} \operatorname{vec}(1 - \underline{\mathbf{x}}^*(k)\underline{\mathbf{x}}_d(k)). \tag{3.2}$$

Recalling: $\underline{\mathbf{x}}(k)$ is the current pose of the end effector given by the FKM. $\mathbf{K}$ is an arbitrary positive definite gain matrix. $\mathbf{N}$ is the modified Jacobian to comply with the transformation invariant error metrics and $\underline{\mathbf{x}}_d(k)$ is given by (3.1).

By using (3.2), we need to select a suitable inversion for $\mathbf{N}$. As shown in section 2.3, we may either choose a damped inverse or the Moore-Penrose inverse. While the damped inverse protects the system from large velocities near task-space singularities, it results in end effector tracking errors. The Moore-Penrose inverse allows precise tracking but may be dangerous near singular configurations. In fact, those two provide trade-offs in which neither options are acceptable. In our case, we need to use the Moore-Penrose inverse when possible, so that the end effector tracking is precise. When near singular configurations, the procedure should stop and demand manual reconfiguration.

We can optimize the manipulator workspace to reduce procedure halting as much as possible with the related works described in section 3.1. Adding another layer to that protection, the robot can try[4] evading singularities at run-time. As shown in section 2.4, this can be done by projecting a singularity evasion function $s(\theta)$ in (3.2) nullspace which results in

$$\vec{\theta}(k+1) = \vec{\theta}(k) + \mathbf{K}\mathbf{N}^{\dagger} \operatorname{vec}(1 - \underline{\mathbf{x}}^*(k)\underline{\mathbf{x}}_d(k)) + \mathbf{P}_N\mathbf{K}_s\mathbf{J}_s^{\dagger}(\vec{s}(\theta) - \vec{s_d}). \tag{3.3}$$

Recapitulating: $\mathbf{P}_N = (\mathbf{I} - \mathbf{N}^{\dagger}\mathbf{N})$ is the nullspace projector. $\mathbf{K}_s$ is an arbitrary positive definite gain matrix for the secondary objective. $\vec{s}(\theta)$ is a singularity evasion function, $\vec{s_d}$ is the desired value for $\vec{s}(\theta)$, and $\mathbf{J}_s = \partial\vec{s}(\theta)/\partial\vec{\theta}$. Whenever necessary, note that $e(k) = \| \operatorname{vec}(1 - \underline{\mathbf{x}}^*(k)\underline{\mathbf{x}}_d(k)) \|$.

A last issue when using (3.3) is related with the motion the end effector performs between the current end effector pose and the reference. Although the closed loop scheme converges, the path is defined by the inversion of the Jacobian and the error definition. For instance, the Moore-Penrose inverse of $\mathbf{N}$ results in a joint path with the least-squares velocity with no concern with the RCM constraints. This problem grows when the reference is farther from the current pose.

The only guarantee we have by using (3.3) in a singularity free path with a suitable $\mathbf{K}$ is exponential convergence. That is, for a constant reference, we have $e(k+1) < e(k)$. Considering this, we can avoid problems of having the current pose $\underline{\mathbf{x}}(k)$ far from the desired pose $\underline{\mathbf{x}}_d(k)$ by adding $N \in \mathbb{N} - \{0\}$ intermediary references that also keep the RCM constraint. An illustrative comparison of using such interpolation or not is shown in Figure 3.5 where one can see how hazardous it could be to ignore such issue.

3.2.3 Dual quaternion constrained interpolation

In order to generate N intermediary RCM constrained poses between $\underline{\mathbf{x}}(k)$ and $\underline{\mathbf{x}}_d(k)$, we must first find their descriptions in relation to the RCM.

[4]Nullspace projection allows "trying" only. The false claim of convergence in this case is far too common in the literature.

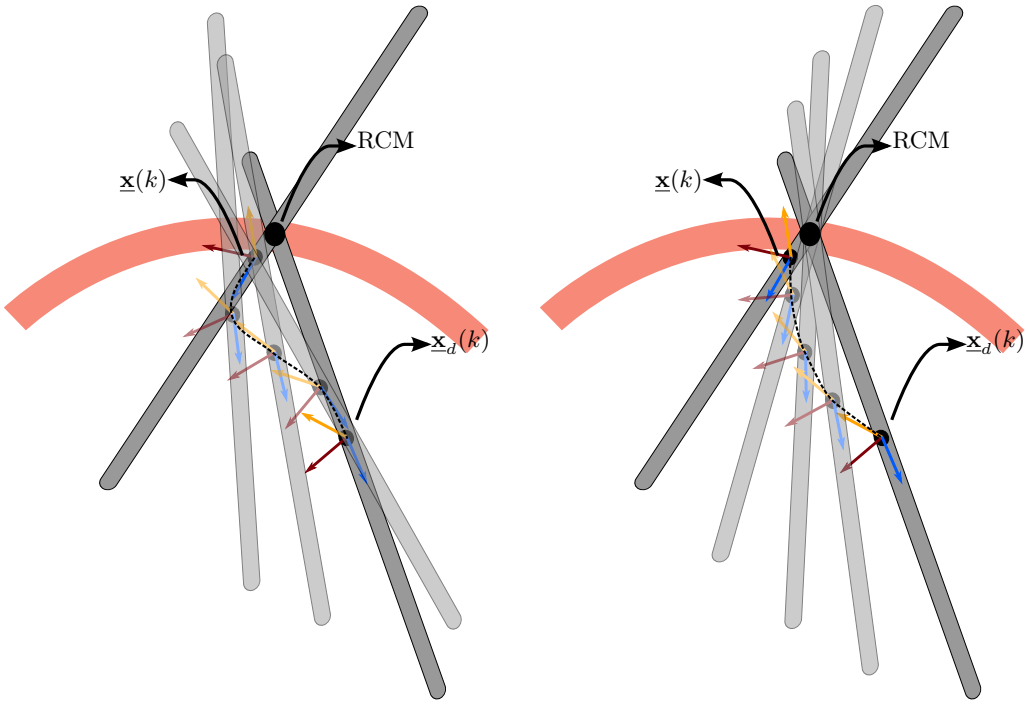

Figure 3.5: Non-interpolated path, where the intermediary poses do not keep the RCM constraint (*left*) and interpolated path with intermediary RCM constrained poses (*right*).

Consider then that the interpolation is made in an instance l, in which we store $\underline{\mathbf{x}}(l)$, the end effector pose at instant l, and $\underline{\mathbf{x}}_d(l)$, the desired end effector pose at instant l. They will be, respectively, the beginning and ending poses in our interpolation. Then, we need to obtain the constrained description of $\underline{\mathbf{x}}(l)$ as if it was given by (3.1). That is, the rotation $\mathbf{r}^l_{\text{rcm}}$ and the constrained translation $\mathbf{t}^l_{\text{rcm}}$ so that $\underline{\mathbf{x}}_{\text{rcm}}\mathbf{r}^l_{\text{rcm}}\underline{\mathbf{t}}^l_{\text{rcm}}$ equals $\underline{\mathbf{x}}(l)$, as shown in Figure 3.6. For that purpose, we note that

$$\underline{\mathbf{x}}(l) = \underline{\mathbf{x}}_{\text{rcm}}\underline{\mathbf{x}}^l_{\text{rcm}}, \tag{3.4}$$

for some $\underline{\mathbf{x}}^l_{\text{rcm}}$ that describes the motion from $\underline{\mathbf{x}}_{\text{rcm}}$ to $\underline{\mathbf{x}}(l)$. From (3.4), we can obtain $\mathbf{r}^l_{\text{rcm}}$ by noticing

$$\begin{aligned} \underline{\mathbf{x}}^l_{\text{rcm}} =& \underline{\mathbf{x}}^*_{\text{rcm}}\underline{\mathbf{x}}(l) \\ \implies \mathcal{P}(\underline{\mathbf{x}}^l_{\text{rcm}}) =& \mathcal{P}(\underline{\mathbf{x}}^*_{\text{rcm}}\underline{\mathbf{x}}(l)) \\ \therefore \mathbf{r}^l_{\text{rcm}} =& \mathcal{P}(\underline{\mathbf{x}}^*_{\text{rcm}}\underline{\mathbf{x}}(l)). \end{aligned} \tag{3.5}$$

With (3.5), we can obtain the translation from the RCM to $\underline{\mathbf{x}}(l)$ as given by

$$\mathbf{t}^l_{\text{rcm}} = \text{translation}((\underline{\mathbf{x}}^l_{\text{rcm}}\mathbf{r}^l_{\text{rcm}})^*\underline{\mathbf{x}}(l)).$$

Due to the RCM constraint, $\mathbf{t}^l_{\text{rcm}}$ can only be a translation in the z-axis. However, numerical and kinematic inaccuracies may cause it to be some

$$\mathbf{t}^l_{\text{rcm}} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k},$$

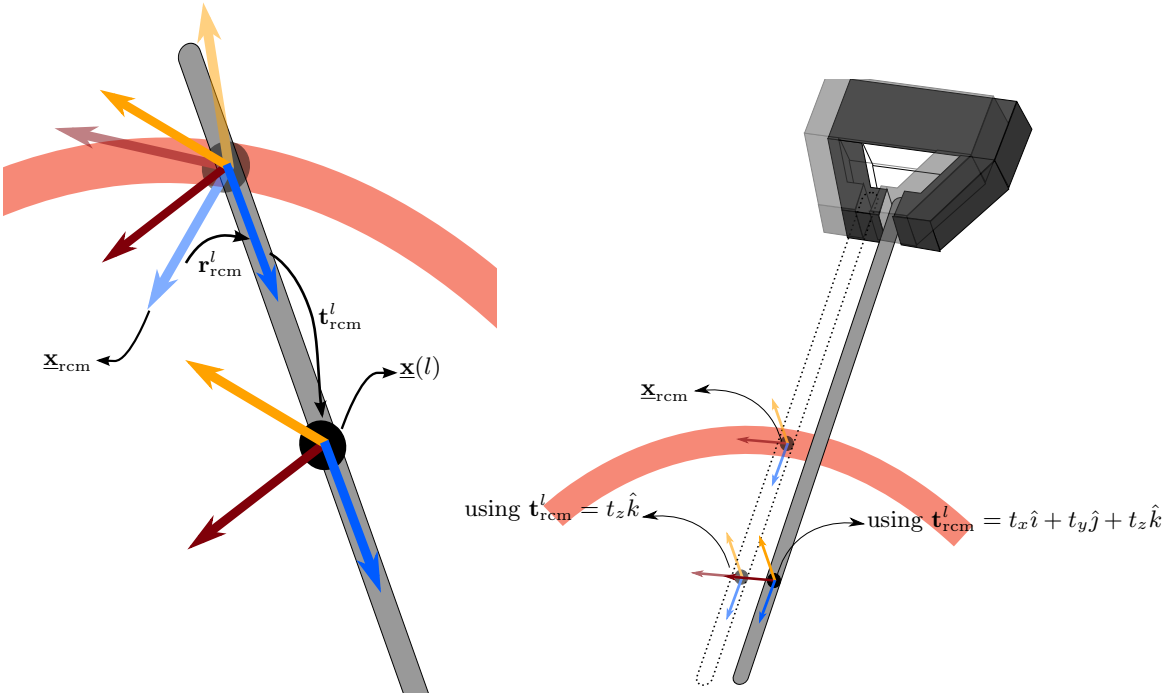

Figure 3.6: The end effector description as seem from the RCM, considering the laparoscopy constraints (*left*). Noisy translation adjustment visualization (*right*).

with $t_x \neq 0$ and $t_y \neq 0$. To correctly interpolate points between two poses that maintain the RCM constraint, we need the starting and ending poses to also keep the constraint. So, instead of using $\underline{\mathbf{x}}(l)$ as-is, we force $t_x = t_y = 0$ to obtain

$$\mathbf{t}^l_{\text{rcm}} = t_z\hat{k} \implies \underline{\mathbf{t}}^l_{\text{rcm}} = (1 + \frac{1}{2}\epsilon t_z\hat{k}).$$

We then define $\underline{\mathbf{x}}'(l) = \underline{\mathbf{x}}_{\text{rcm}}\mathbf{r}^l_{\text{rcm}}\underline{\mathbf{t}}^l_{\text{rcm}}$, which it is $\underline{\mathbf{x}}(l)$ shifted in space so that its z-axis coincides with the RCM point, as shown in Figure 3.6.

In the case of $\underline{\mathbf{x}}_d(l)$, we can use (3.1) to see that

$$\mathbf{r}^d_{\text{rcm}} = \mathbf{r}_x(l)\mathbf{r}_y(l)\mathbf{r}_z(l) \quad \text{and} \quad \mathbf{t}^d_{\text{rcm}} = \text{translation}(\underline{\mathbf{t}}_z(l)).$$

Now that the descriptions the rotations and constrained translation of $\underline{\mathbf{x}}'(l)$ and $\underline{\mathbf{x}}_d(l)$ were found in relation to the RCM, we then find the intermediary points between them.

To better understand the interpolation between $\underline{\mathbf{x}}'(l)$ and $\underline{\mathbf{x}}_d(l)$, we will show the calculation of the rotational and translational parts separately. To find the incremental rotation, we begin by decomposing the relative rotation between initial and final rotations into $N+1$ equal partial rotations $\mathbf{r}_{\text{inc}}$. By accumulating those $N+1$ rotations we go from the current rotation $\mathbf{r}^l_{\text{rcm}}$ to $\mathbf{r}^d_{\text{rcm}}$, that is

$$\mathbf{r}^l_{\text{rcm}}(\mathbf{r}_{inc})^{N+1} = \mathbf{r}^d_{\text{rcm}} \implies (\mathbf{r}_{inc})^{N+1} = (\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}},$$

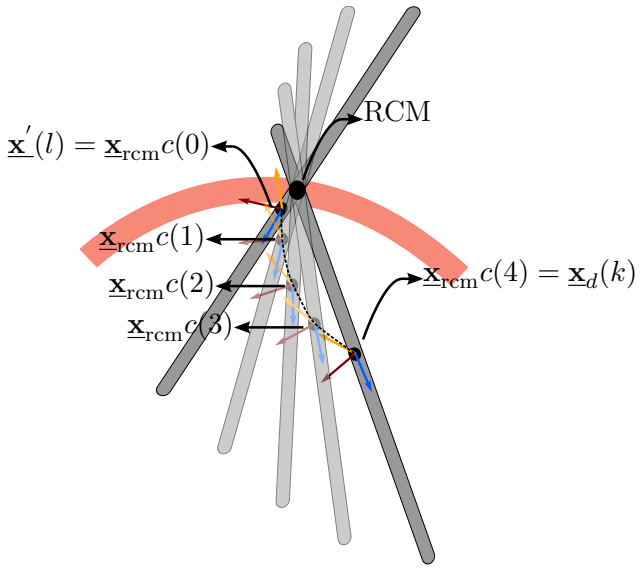

Figure 3.7: Example interpolated path when $N = 3$.

then we can use the log operator to obtain $\mathbf{r}_{inc}$

$$\begin{aligned}(N+1)\log(\mathbf{r}_{inc}) &= \log((\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}}) \\ \implies \log(\mathbf{r}_{inc}) &= \frac{1}{N+1}\log((\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}}) \\ \therefore \mathbf{r}_{inc} &= \exp(\frac{1}{N+1}\log((\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}})).\end{aligned} \tag{3.6}$$

And the incremental translation in the z-axis is simply given by

$$\mathbf{t}_{\text{inc}} = \left(\frac{1}{N+1}\right)(\mathbf{t}^d_{\text{rcm}} - \mathbf{t}^l_{\text{rcm}}). \tag{3.7}$$

Therefore, at each interpolation step $m \in \mathbb{N}$ in the interval $[1, N+1]$, we compose both (3.6) and (3.7) to obtain

$$c(i) = \mathbf{r}^l_{\text{rcm}}\mathbf{r}^{\{m\}}_{\text{inc}}(1 + \frac{1}{2}\epsilon(\mathbf{t}^l_{\text{rcm}} + \mathbf{t}^{\{m\}}_{\text{inc}})), \tag{3.8}$$

so that the interpolated path is given by

$$\underline{\mathbf{x}}'(l) = \underline{\mathbf{x}}_{\text{rcm}}c(0) \to \underline{\mathbf{x}}_{\text{rcm}}c(1) \to \underline{\mathbf{x}}_{\text{rcm}}c(2) \to \cdots \to \underline{\mathbf{x}}_{\text{rcm}}c(N+1) = \underline{\mathbf{x}}_d(l).$$

as shown in Figure 3.7. With this, the intermediary points have been found.

Instead of feeding the kinematic controller (3.3) directly with $\underline{\mathbf{x}}_d(k)$, we first find the interpolated path whenever $e(k) > e_a$, in which $e_a \in \mathbb{R}^+$ is an error upper bound. Then, we send $\underline{\mathbf{x}}_{\text{rcm}}c(1)$ to (3.3) and wait for convergence, then $\underline{\mathbf{x}}_{\text{rcm}}c(2)$ is sent and so on until $\underline{\mathbf{x}}_{\text{rcm}}c(N+1) = \underline{\mathbf{x}}_d(l)$. In order to visualize the effects of N, the reader is advised to read the example shown in section 3.3 on the following page.

With this, the novel camera controller for laparoscopy applications for the general serial link robot is formally explained. The input signals are simple camera motion commands, while the RCM

and tool-tip positioning have the same priority. As much as possible, the controller reconfigures the robot in its nullspace to avoid singular configurations.

In the following sections, the proposed controller is evaluated in three simulated experiments. The basic functionally of the system is evaluated in section 3.3 by moving the endoscope to a fixed position with increasing values for N. In the second evaluation, shown in section 3.4, two distinct manipulator geometries are assessed in moving the endoscope in a conical helix path. The final evaluation is shown in section 3.5, in which we assess the interaction of untrained users with the system in a simulated endoscope control task.

3.3 System evaluation: basic example

In order to fully comprehend the devised laparoscopy controller and see the effects of the dual quaternion interpolation, let us use it in an example. Consider we have a Schunk LWA3 robot with DH parameters as described in Table 3.1.

	θ [rad]	d [m]	a [m]	α [rad]
1	0.0	0.3	0.0	$-\frac{\pi}{2}$
2	0.0	0.0	0.0	$+\frac{\pi}{2}$
3	0.0	0.328	0.0	$-\frac{\pi}{2}$
4	0.0	0.0	0.0	$+\frac{\pi}{2}$
5	0.0	0.2765	0.0	$-\frac{\pi}{2}$
6	0.0	0.0	0.0	$+\frac{\pi}{2}$
7	0.0	0.1793	0.0	0.0

Table 3.1: Standard DH parameters [1, p. 61-65] of the Schunk LWA3 manipulator.

We then attach an endoscope at the end of the robot so that its shaft coincides with the robot z-axis. Suppose that the endoscope has a mark 20 cm from the gripping point. That mark should initially coincide with the trocar point. Also, we add a rotation around the z-axis to correctly align the end effector y-axis with the camera up direction. For those purposes, a last transformation is added on the robot to put $\underline{\mathbf{x}}_{\text{rcm}}$ in the correct place in the kinematic chain as shown in Table 3.2. This transformation is summarized in Figure 3.8 on the next page.

	θ [rad]	d [m]	a [m]	α [rad]
8	$+\frac{\pi}{2}$	0.2	0.0	0.0

Table 3.2: Last frame transformation, which acts as a fixed joint.

In this example, the robot initial posture when inserted in the patient is given by $\vec{\theta}(0) = \begin{bmatrix} 0 & 0.75 & 0 & 0.75 & 0 & 1.5 & 0 \end{bmatrix}^T$. The task will be to move the endoscope from the initial pose $\underline{\mathbf{x}}(0) = \underline{\mathbf{x}}_{\text{rcm}}$ to a desired pose given by

$$\underline{\mathbf{x}}_d = \underline{\mathbf{x}}_{\text{rcm}} \mathbf{r}_x \mathbf{r}_y \underline{\mathbf{t}}_z,$$

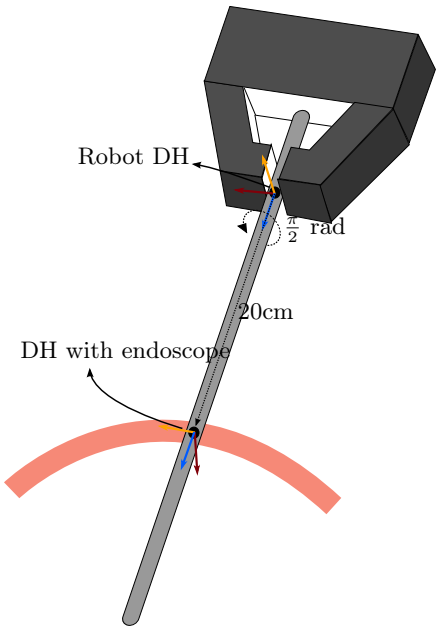

Figure 3.8: Visualization of the last transformation added in the kinematic chain.

in which

$$\mathbf{r}_x = \cos\left(\frac{0.0872}{2}\right) + \hat{\imath}\sin\left(\frac{0.0872}{2}\right),$$
$$\mathbf{r}_y = \cos\left(\frac{0.61}{2}\right) + \hat{\jmath}\sin\left(\frac{0.61}{2}\right), \text{ and}$$
$$\underline{\mathbf{t}}_z = 1 + \frac{1}{2}\epsilon\hat{k}0.05.$$

This motion is a rotation of 5° around the x-axis, a rotation 35° around the y-axis and a translation of 5 cm along the z-axis. For the purposes of this example, the desired motion is "far" enough from the initial pose. It can be reached in a singularity free path with the kinematic controller

$$\vec{\theta}(k+1) = \vec{\theta}(k) + \mathbf{K}\mathbf{N}^{\dagger}\,\mathrm{vec}(1 - \underline{\mathbf{x}}^{*}(k)\underline{\mathbf{x}}_d(k)), \tag{3.9}$$

with $\mathbf{K} = 0.3$. This controller has no singularity avoidance as it was not necessary. Note that $e(k) = \| \mathrm{vec}(1 - \underline{\mathbf{x}}^{*}(k)\underline{\mathbf{x}}_d(k)) \|$ is the dual quaternion error.

The effects of interpolating with $N \in \{0, 5, 10, 20, 50, 100\}$ points can be seen on Figure 3.9 on page 41 and on Figure 3.10 on page 42. A step in the interpolation is reached whenever $e(k) < 0.001$.

3.3.1 Results & discussion

With the chosen value for $\mathbf{K}$ in (3.9), we can clearly see the exponential convergence of $e(k)$. As N increases, the maximum value for $e(k)$ lowers from 0.35 when $N = 0$ to 4.2×10^{-3} when $N = 100$. Given that the intermediary poses also keep the RCM constraint, we can see the maximum RCM error falling from 7.8 mm when $N = 0$ to 1.23×10^{-3} mm when $N = 100$.

The side-effect of increasing N in the interpolation is seen on the required iteration axis, that raises from 18 with $N = 0$ to 510 with $N = 100$. This basically means that it will take more time for (3.9) to reach the desired pose.

In a real scenario with a manipulator that has physical motors and encoders, it may be unnecessary to seek RCM errors under 1.23×10^{-3} mm. Naturally, we do not require such precision from a surgeon. As far as the safety of the patient is concerned, even by selecting $N = 5$ which mildly increases the necessary iterations we already obtain RCM errors under 0.2 mm. It is of course a safe supposition that the skin can handle such errors. Even so, interpolation cannot be abandoned altogether as with $N = 0$ the RCM error is already close to 1 cm and may be worse for a further $\underline{\mathbf{x}}_d$.

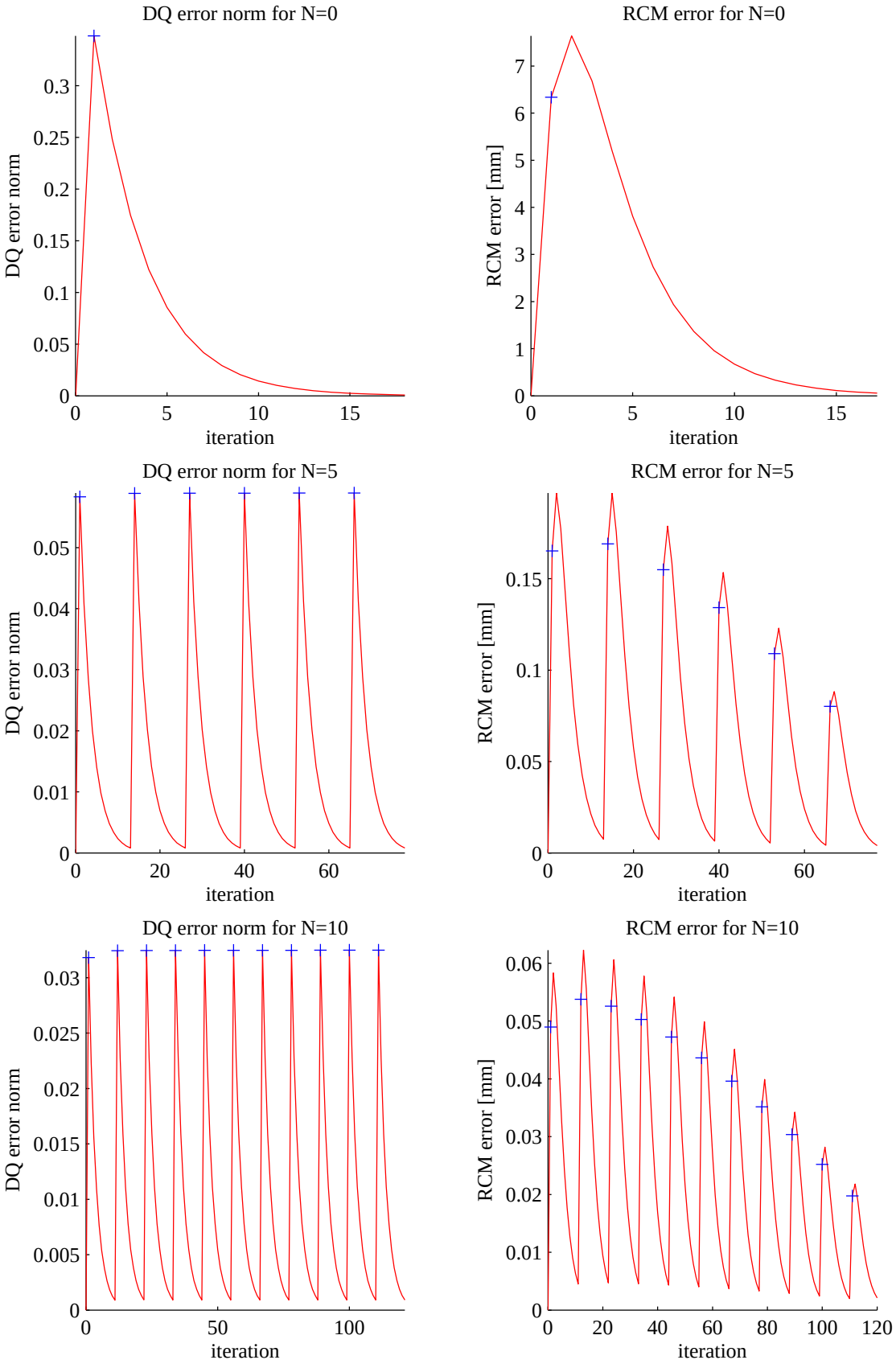

Figure 3.9: Dual quaternion (*left*) and RCM errors (*right*) when $N \in \{0, 5, 10\}$. The blue crosses show the instant when a new step is sent to the controller.

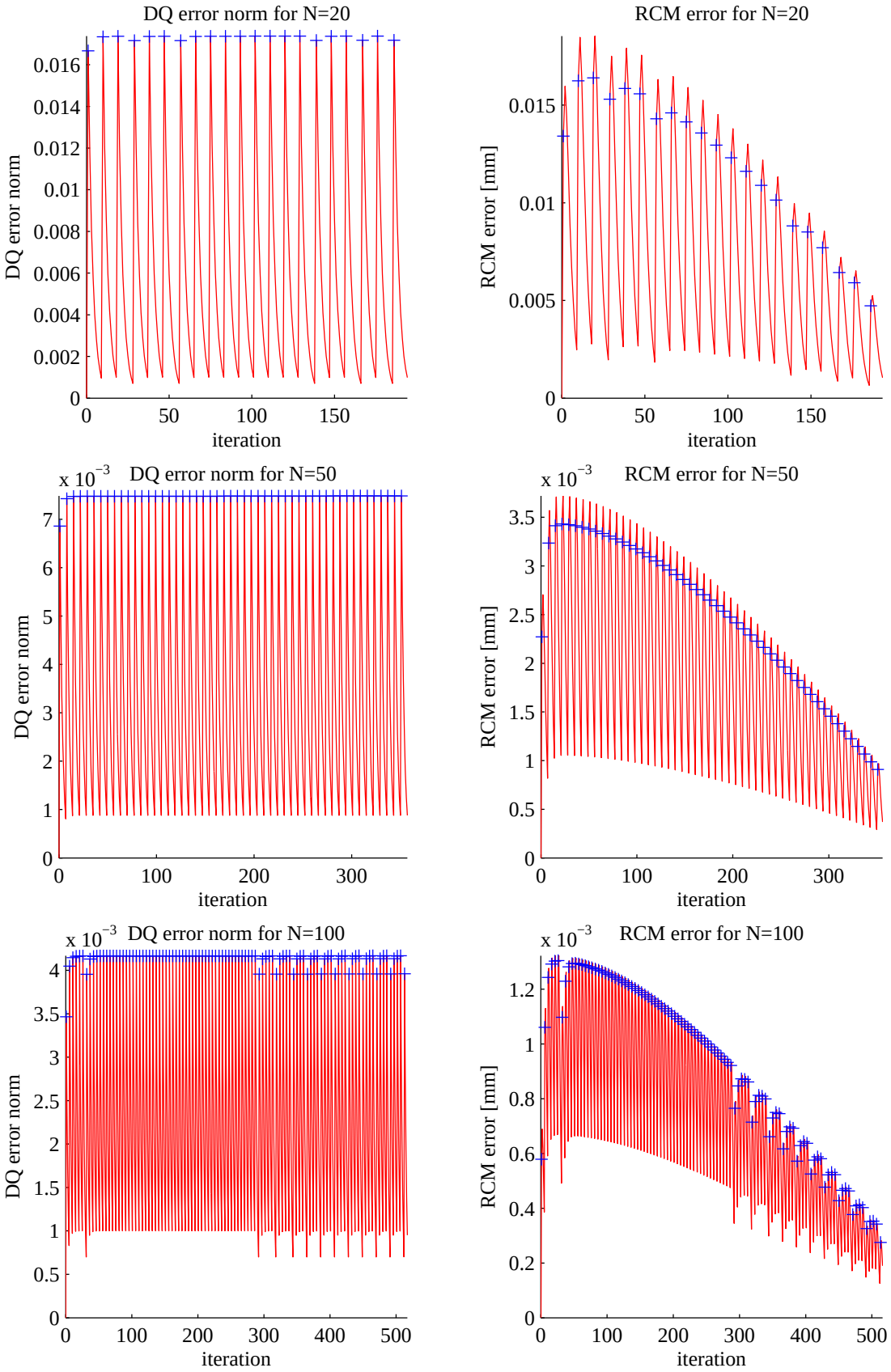

Figure 3.10: Dual quaternion (*left*) and RCM errors (*right*) when $N \in \{20, 50, 100\}$. The blue crosses show the instant when a new step is sent to the controller.

3.4 System evaluation: manipulator geometry flexibility

To validate the control strategy and verify its flexibility, we performed computational simulations with two different commercial robots: a Schunk manipulator and a Meka anthropomorphic arm. The Schunk LWA3 is as described in the last section. The Meka A2 arm, on its turn, is a 7DOF compliant manipulator which is part of the Mekabot Humanoid robot. It is manufactured by Meka Robotics LLC and is a human safe product, intended to be used in cooperative robotics.

In the simulations, a laparoscopic tool was considered to be attached to the end effector of the robots. A predefined conical helix path is given as reference to the tool tip position and the same control scheme described in this chapter is used to control both manipulators, changing only the robot DH parameters accordingly. To emulate the effects of the interpolation, the references in the conical helix path were generated to be close to each other.

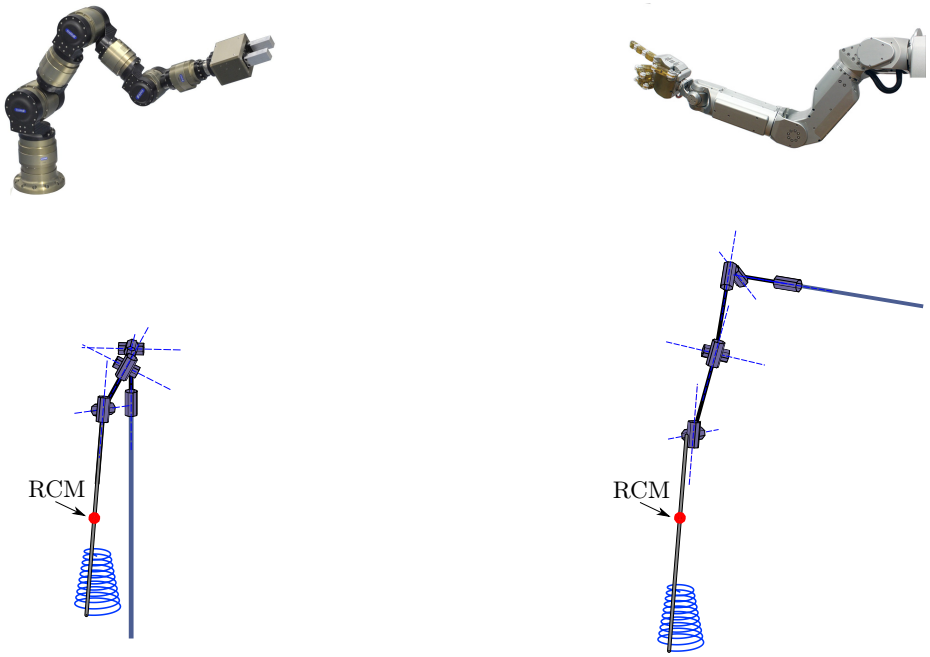

Figure 3.11: Simulations with a conical helix trajectory tracking.

Figure 3.11 shows the tool tip trajectory for both robots. Even though the two robotic platforms have very distinct design and kinematic models, they both behaved as expected and our proposed controller was able to track the desired path while keeping the RCM pivot constraint as shown in Figure 3.12.

The first plot shows the tool tip position error when following the desired trajectory, while the second plot shows the deviation of the tool from the initial RCM point. The obtained error values are negligible during the whole procedure, especially those for the RCM point, which are smaller than 0.05 mm.

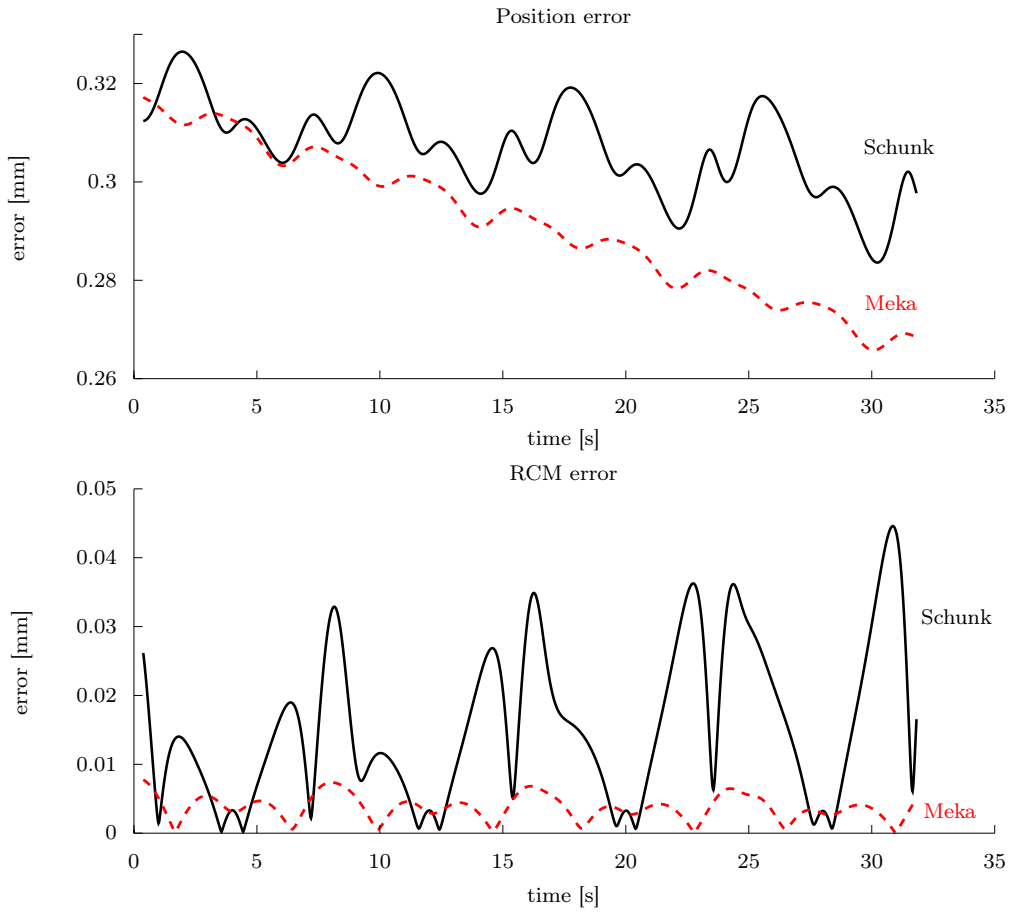

Figure 3.12: Errors for both Schunk (*solid*) and Meka (*dashed*).

3.5 System evaluation: user interaction

As the first form of evaluation, we devised a simulated endoscope controlling task using the OpenRAVE environment. The OpenRAVE was used to represent the robot in 3D space and simulate the endoscopic camera images. This simulation evaluated both subjective and objective parameters of our system. The subjective parameters were related to intuitiveness and ease of use; while the objective parameters were completion time, RCM error and end effector tracking error. Also, in current medical practice, simulations are helping surgeons to be trained in less time as their training is not limited to patients or cadavers [3].

Given the natural unavailability of medically trained personnel for such trials, we performed experiments with 20 subjects with no prior medical training. We assume that medically untrained individuals will have an overall worse performance, as they have no prior knowledge in performing any form of surgery. In our simulation, we chose a 0º endoscope because it is simpler to comprehend.

The selected robot was a simulated schunk LWA3. By the time this experiment was done, the real robot was available for physical experiments on our laboratory, but those plans changed when our compliant robot arrived. Although the schunk LWA3 is not human-safe, it has seven degrees of freedom and is small enough to be used in an operating room; therefore it has the necessary

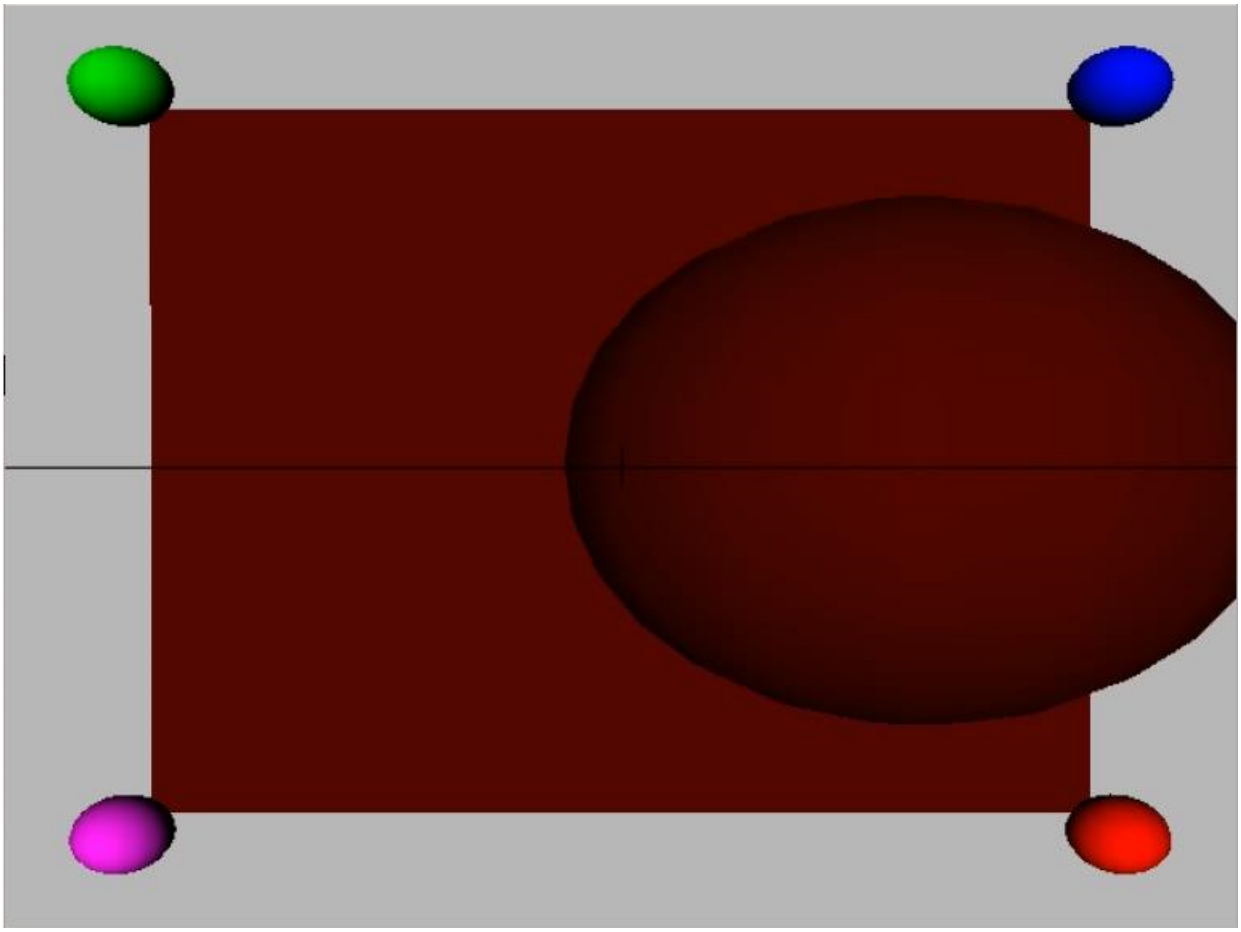

Figure 3.13: The initial camera view. The user should move the camera to obtain closer view of the small colored spheres, while avoiding the largest one.

characteristics to evaluate our laparoscopy camera controlling framework. Using this robot, the constants of the kinematic controller (3.3) were selected as $\mathbf{K} = 60.0$, $\mathbf{K}_c = 7$, and $\vec{s_d} = 1$; to allow both stability and good responsiveness. The modified DH parameters that take the endoscope into account are the same used in the example in section 3.3 on page 38. The objective singularity-evasion function $\vec{s}(\theta)$ was selected as in [31]. Note that those values for the gain matrices are greatly dependent on this ideal implementation. Their values should always be carefully tuned for physical implementations.

In order to generate the user inputs for (3.1), the subjects used the translational degrees of freedom of an Omega 7 haptic interface to move the camera. The deviations around the initial haptic interface translation were transformed into the $u_\alpha(k)$ (up-down), $u_\beta(k)$ (left-right), and $u_z(k)$ (in-out) commands. As the device has force feedback only on the translational degrees of freedom, this choice allowed us to add a viscosity parameter in the hand movement, both reducing hand tremor and helping slowing the users' movements.

The user controls the interface while seated and facing the monitor that shows the endoscopic camera images as in Figure 3.15. The initial camera view for the task is shown in Figure 3.13, with colored spheres of same radius positioned in the corners of a 5×5 cm square. The user

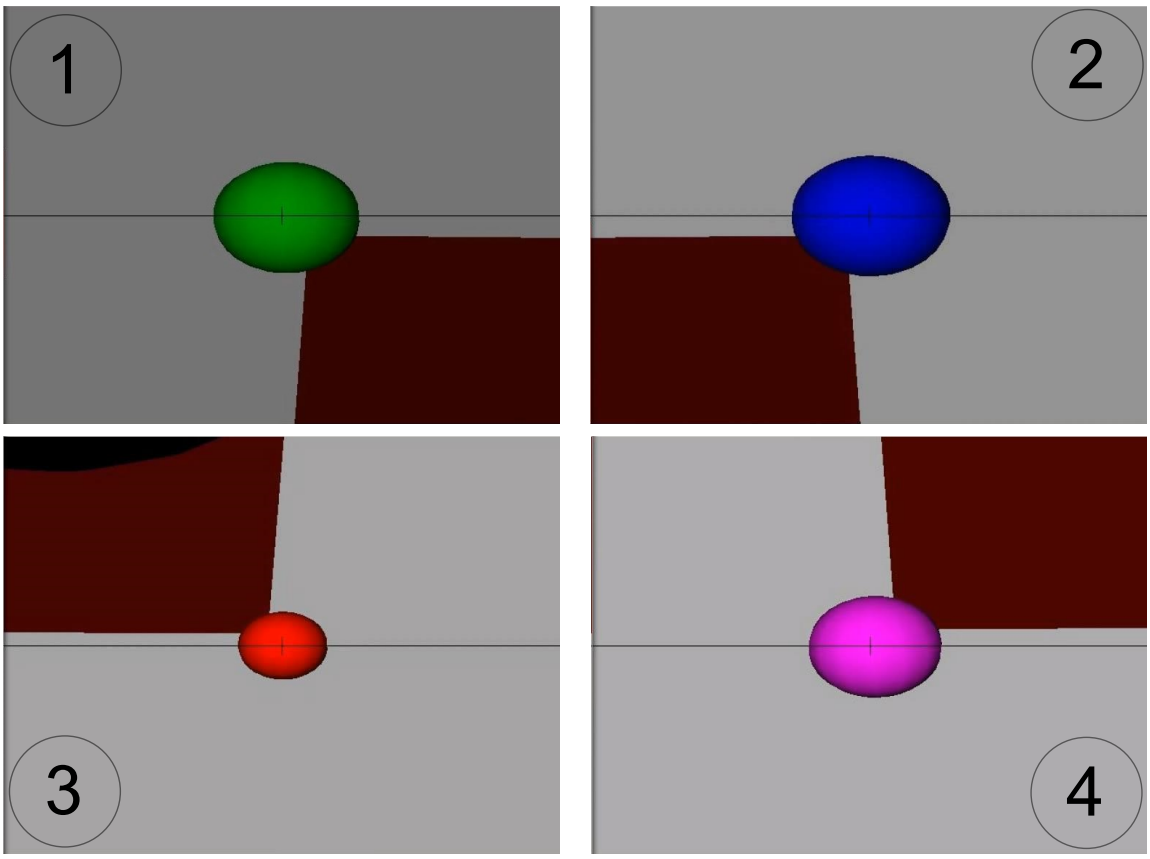

Figure 3.14: The user should obtain an image as similar as possible to the ones shown above (the numbers are only here to show the correct order of targets). When the user succeeded, the current target sphere turned gray and the user should proceed to the next target.

should move around the environment to find the position at which the camera would show the same images in the order of Figure 3.14; in the shortest time possible, while avoiding collision with the big obstacle sphere positioned between objectives 2 and 3. During the task design, each of the target images was coupled with the end effector translation that caused that image to appear on the endoscope camera. Whenever the user positioned the end effector within a 1 mm tolerance from that expected translation, the objective was considered reached and the user had to move to the next target. Such small tolerance required a full comprehension and control from the user, for a more reliable evaluation of the control system.

The experimental evaluation process was threefold. First, before each user started the experiment, they were given a form (shown in appendix II.1) in which they were asked to quantify their previous knowledge concerning laparoscopic procedures, surgical simulators and haptic interfaces in a $1 \rightarrow 4$ scale. This initial conversation was only to find the users with less confidence in their hand-eye coordination prior to the experiment. The less confident received two minutes to interact with the experimental setup, while the others received only a short verbal explanation of the task.

Secondly, during the execution of the task data was stored regarding the distance to the obstacle sphere, the RCM error, and the instant each target was reached for all 20 trials. As our evaluation evolved, in the 19^{th} and 20^{th} trials the end effector tracking error was also measured.

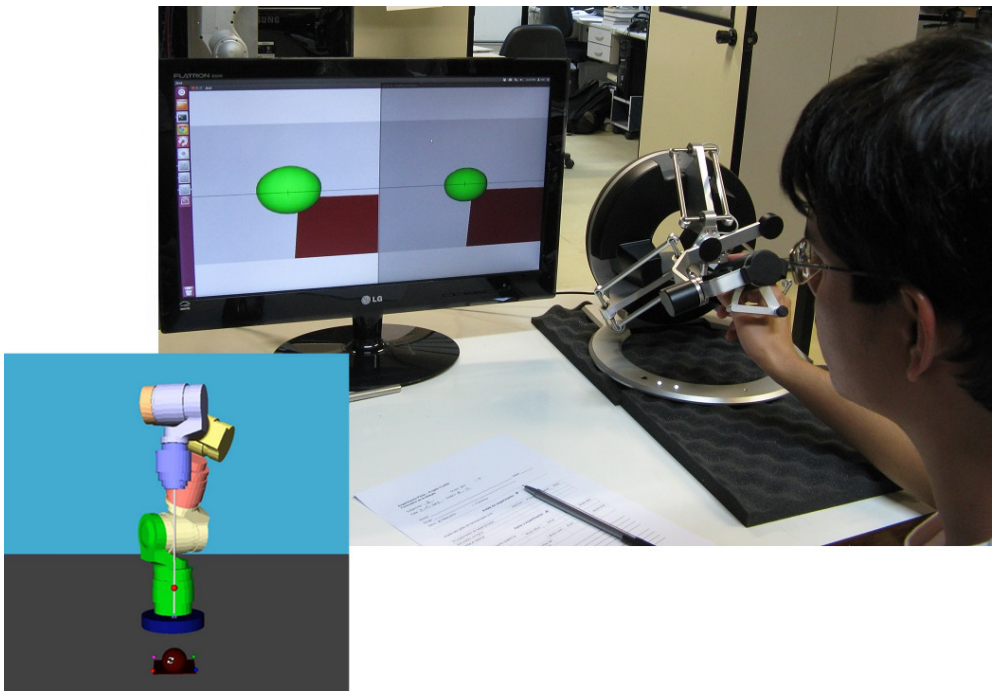

Figure 3.15: Subject 18 performing the experiment. His screen is split into current camera image (*left*) and current objective (*right*). In a separate screen, the evaluator could observe the robot (*bottom left*) for any unexpected motions.

All information was later analyzed for performance evaluation purposes.

Finally, at the end of the exercise we asked each subject to evaluate its ease (1 = very difficult, 2 = difficult, 3 = easy, 4 = very easy) and how well they considered their hand movements were translated into camera movements (1 = very bad, 2 = bad, 3 = good, 4 = very good) so that we could qualify the user interaction with the system.

3.5.1 Results & discussion

Even though all users had no prior knowledge in handling an endoscope, they were able to successfully perform the task proposed within a reasonable amount of time, with average of 68.03 seconds and standard deviation of 30.58 seconds. Moreover, they completely avoided the obstacle. Group-wise, the trainees completed the task within 84 seconds in average, while the others took 52 seconds. All durations correspond to the time they took between obtaining the first target image and the last, to guarantee they fully understood the task during the timing period.

With respect to controller performance and considering all 20 subjects, the variables that quantify its performance are shown in Table 3.3. The end effector positioning error was no bigger than 4.82 mm with a RMS value of 0.3398 ± 0.0163 mm in the two trials considered. Figure 3.16 shows the tool-tip trajectory in one experiment, while the corresponding RCM and end effector tracking errors are shown in Figure 3.16. Even tough the users were encouraged to move the endoscope as fast as possible, both RCM and end effector tracking errors were negligible.

	mean ± standard deviation	min	max
RCM RMS error [mm]	0.00710 ± 0.00706	–	0.37
Minimal distance to obstacle [mm]	12.52 ± 5.03	–	–
Task completion time [s]	68.03 ± 30.58	18.26	149.05

Table 3.3: Performance results for all 20 subjects that participated in the experiments for controlling the virtual endoscope.

Regarding their interaction with the system, the users evaluated how well the system performed their intended movements with an average of 3.68 points (between good and very good). The most common complaint was the lack of depth perception caused by the flat image of the endoscope, which is a limitation of the simulation and not of the controller. As a result of the system ease of use, the users gave an average of 3.11 points (between easy and very easy) in the task ease scale, even with the small camera position tolerance and the lack of depth perception they themselves observed.

The subjective evaluation scores of each user along with their experiment duration is available in Table II.2 on page 62. Also, the data from all users concerning RCM error is shown in Figure II.2 on page 64 and in Figure II.3 on page 65. Finally, the trajectory of the endoscope tip for all experiments is shown in Figure II.4 on page 66 and in Figure II.5 on page 67.

Given that 20 users with no prior medical experience were able to perform a complex task in under 2.5 minutes and that both RCM and end effector tracking errors behaved nicely during all trials considered, we believe that the overall results are promising and, therefore, motivate trials using a real robot. The experimental setup for the physical experiments is shown in section II.1, in the appendix. However, results of systematic experiments are not yet available.

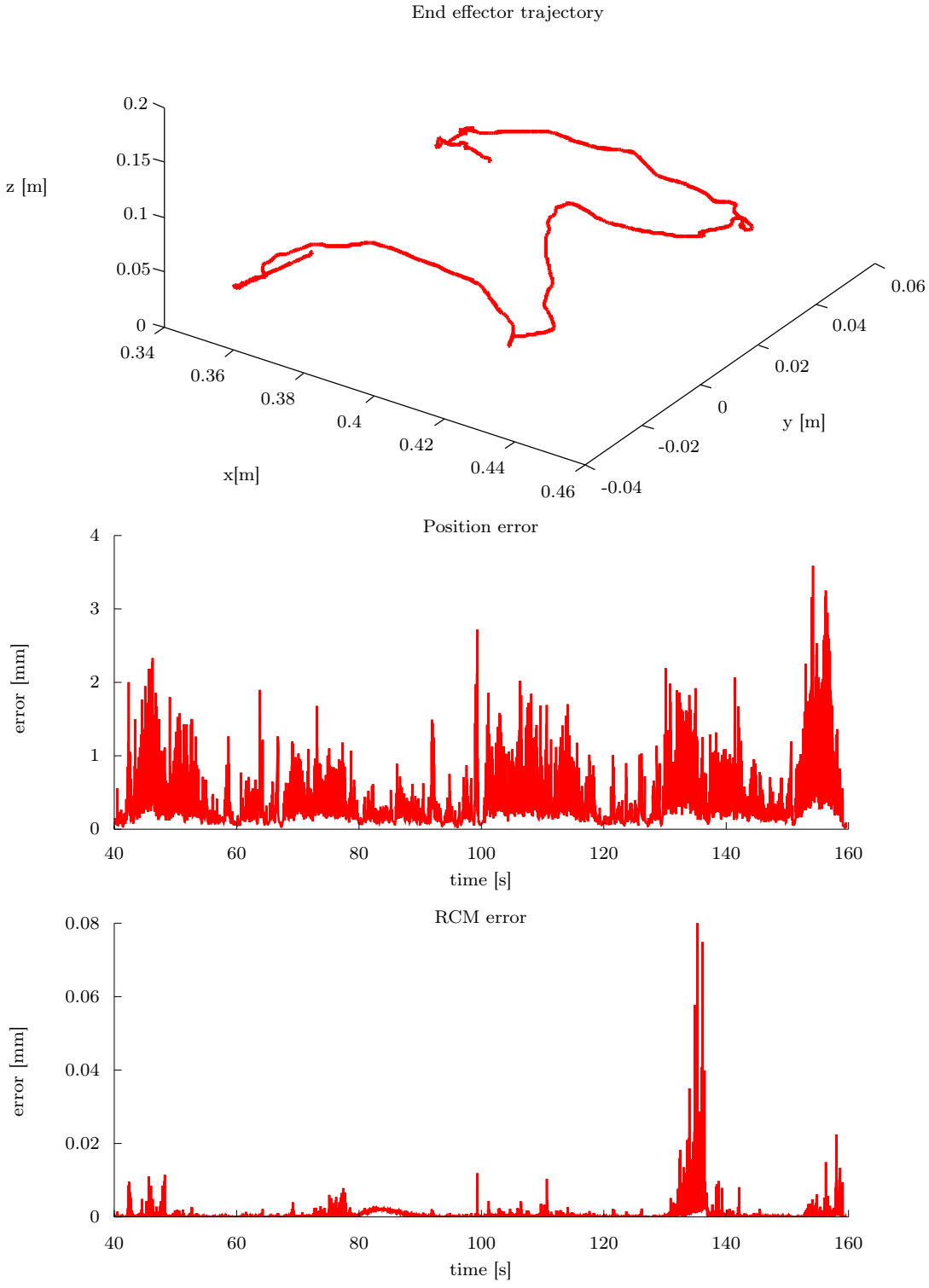

Figure 3.16: Trajectory of the tool-tip and error parameters during one of the trials.

Chapter 4

Conclusion

"Gigantum humeris insidentes."
–*Standing on the shoulders of giants.*

Motivated by the overall difficulty of manually performing laparoscopic surgeries, this work studied how serial-link manipulators could aid surgeons in controlling an endoscope under laparoscopic contraints. Robots in this setting must be controlled to generate a remote center of motion to protect the surgical entry point from dangerous strain. With regard to manipulator control, classic techniques concerning singularity robustness and redundancy exploitation were adapted to the dual quaternion framework in chapter 2. Those contributions aim to further enhance usability of the dual quaternion manipulator control formulation by proving solutions analogous to classic techniques. In the end chapter 2, the performance of four controllers in a tool pose control task were presented. With this, the required mathematical background to fully comprehend the application of such techniques in an endoscope control task was presented.

After briefly presenting the robot-aided laparoscopy problem, chapter 3 begun with the state of the art in robotic laparoscopy in which several works were reviewed. As presented therein, existing techniques that use serial-link manipulators in laparoscopic settings have considerable drawbacks. Motivated by those, a novel laparoscopy controller for any serial-link manipulator was shown in section 3.2, which uses constrained dual quaternion interpolation in order to provide an acceptable trade-off between tool positioning error and RCM error. After the effects of different interpolation numbers and their qualitative effects were shown in a simple simulated task in section 3.3, the flexibility of the proposed controller was evaluated using two different manipulator geometries in section 3.4. Finally, qualitative and quantitative behaviors of the system under user generated inputs were assessed in a simulated endoscope control task in section 3.5.

Concerning the objectives of this work, the major result is the endoscope controlling framework. Although the system was only evaluated using simulated robots, the devised control method was tested with inputs generated by users using a real haptic interface. In this last experiment, it was shown that the controlling scheme provided both safe RCM generation and considerably low tool-tip tracking errors, even with the inherent variability expected of user interaction.

4.1 Future work

There are still some issues that require attention before the developed technique can be safely applied in medical settings. For instance, even though the overall results concerning the laparoscopic controller are encouraging, a mathematical relation between the amount of intermediary points in the interpolation and the maximum RCM error must be found. This would allow a systematic selection of the interpolation size. As of this writing, there are only partial results available.

In an simultaneous ongoing work, the adaptation of the proposed laparoscopic controller to tool control is on its testing stage as shown in section II.1. A simulated user interaction task has not yet been made due to the limitations related to 3D depth visualization in the selected robotic simulator environment.

In order to not be limited to simulations, the proposed laparoscopic controller has been adapted to a physical serial-link manipulator robot and is currently under testing. If such transition shows results equivalent to the simulated experiments, user interaction will be evaluated in a physical setup in tasks in which the user controls both camera and tools. After the basic challenges related to physical implementations and user interaction are solved, the system will then be tested by medically trained personnel with varying levels of expertise in laparoscopic surgery, as to more thoroughly evaluate the intuitivity and overall usability of the system to the target group.

In a long time-frame and after several evaluations are made, it is expected that this framework could aid in real medical setting. At first, we can aim an integrated operation room with several robots controlling both tools and cameras in laparoscopic scenarious. Moreover, by combining frameworks that can safely aid surgeons in other types of surgeries and procedures, it seems reasonable to conjecture that patients will be provided with safer and less invasive surgeries, while allowing surgeons to work under reduced mental and physical strain.

BIBLIOGRAPHY

[1] SICILIANO, B. et al. *Robotics Modelling, Planning and Control*. London: Springer, 2008.

[2] NEZHAT, C. *Nezhat's History of Endoscopy: A Historical Analysis of Endoscopy's Ascension Since Antiquity*. Endo-Press, 2011. ISBN 9783897569164. Disponível em: <http://laparoscopy.blogs.com/endoscopyhistory/>.

[3] SOPER, N.; SCOTT-CONNER, C. *The SAGES Manual: Volume 1 Basic Laparoscopy and Endoscopy*. Springer, 2012. (The SAGES Manual). ISBN 9781461423447. Disponível em: <http://books.google.com.br/books?id=OX0JFZ5DOycC>.

[4] VENDRAMINI, D. a. L. et al. Ressecções colorretais laparoscópicas e laparotômicas no câncer colorretal. *ABCD. Arquivos Brasileiros de Cirurgia Digestiva (São Paulo)*. ISSN 0102-6720.

[5] DOUBLET, J. D. et al. Retroperitoneal nephrectomy: Comparison of laparoscopy with open surgery. *World Journal of Surgery*, Springer-Verlag, v. 20, n. 6, p. 713–716, 1996. ISSN 0364-2313. Disponível em: <http://dx.doi.org/10.1007/s002689900109>.

[6] RêGO, R. E. C. et al. Tratamento cirúgico da litíase vesicular no idoso: análise dos resultados imediatos da colecistectomia por via aberta e videolaparoscópica. *Revista da Associação Médica Brasileira*. ISSN 0104-4230.

[7] LITYNSKI, G. Laparoscopy-the early attempts: spotlighting Georg Kelling and Hans Christian Jacobaeus. *JSLS*, v. 1, n. 1, p. 83–85, 1997. Disponível em: <http://publikationen.ub.uni-frankfurt.de/files/26738/jsls-1-1-83.pdf>.

[8] SCHOLLMEYER, T. et al. Georg kelling (1866-1945): the root of modern day minimal invasive surgery. a forgotten legend? *Archives of Gynecology and Obstetrics*, Springer-Verlag, v. 276, n. 5, p. 505–509, 2007. ISSN 0932-0067. Disponível em: <http://dx.doi.org/10.1007/s00404-007-0372-y>.

[9] LAU, W.; LEOW, C.; LI, A. History of endoscopic and laparoscopic surgery. *World Journal of Surgery*, Springer-Verlag, v. 21, n. 4, p. 444–453, 1997. ISSN 0364-2313. Disponível em: <http://dx.doi.org/10.1007/PL00012268>.

[10] MACFADYEN, B. *Laparoscopic Surgery of the Abdomen*. Springer, 2003. ISBN 9780387984681. Disponível em: <http://books.google.com.br/books?id=2QLPN5t4s4sC>.

[11] PIEPER, D. L. *The kinematics of manipulators under computer control*. Dissertação (Mestrado) — Stanford University, 1968.

[12] HAGN, U. et al. The dlr miro: a versatile lightweight robot for surgical applications. *Industrial Robot: An International Journal*, Emerald Group Publishing Limited, v. 35, n. 4, p. 324–336, 2008.

[13] MUÑOZ, V. F. et al. Pivoting motion control for a laparoscopic assistant robot and human clinical trials. *Advanced Robotics*, Taylor & Francis, v. 19, n. 6, p. 694–712, 2005.

[14] KASALICKỲ, M. et al. Aesop 3000–computer-assisted surgery, personal experience. *Rozhledy v chirurgii: mesicnik Ceskoslovenske chirurgicke spolecnosti*, v. 81, n. 7, p. 346–349, 2002.

[15] WHITNEY, D. E. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, v. 10, p. 47–53, 1969.

[16] ADORNO, B. V. *Two-arm manipulation: from manipulators to enhanced human-robot collaboration [Contribution à la manipulation à deux bras : des manipulateurs à la collaboration homme-robot]*. Tese (Doutorado) — Université Montpellier 2, 2011.

[17] DOMBRE, E.; KHALIL, W. *Robot Manipulators Modelling, Performance Analysis and Control*. London: ISTE, 2007.

[18] SPONG, M. W.; HUTCHINGSON, S.; VIDYASAGAR, M. *Robot Modeling and Control*. New York: John Wiley & Sons.

[19] PHAM, H.-L. et al. Position and orientation control of robot manipulators using dual quaternion feedback. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: Taiwan, 2010.

[20] FIGUEREDO, L. et al. Robust kinematic control of manipulator robots using dual quaternion representation. In: *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2013.

[21] MEYER, C. D. *Matrix analysis and applied linear algebra*. Philadelphia: siam, 2000.

[22] WAMPLER, C. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems Man and Cybernetics*, v. 16, p. 93–101, 1986.

[23] MACIEJEWSKI, A. A.; KLEIN, C. A. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, v. 5, p. 527–552, 1988.

[24] WHITNEY, D. E. The mathematics of coordinated control of prosthetic arms and manipulators. In: *Proceedings of the Eight annual Conference on Manual Control*. Ann arbor: USA, 1972. p. 207–220.

[25] KLEIN, C. A.; HUANG, C. H. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13, p. 245–250, 1983.

[26] BAILLIEUL, J.; BROCKETT, J. H. . R. Programming and control of kinematically redundant manipulators. In: *Proceedings of 23rd Conference on Decision and Control*. Las Vegas: USA, 1984.

[27] KLEIN, C. A.; BLAHO, B. E. Dexterity measures for the design and control of kinematically redundant manipulators. *The international journal of robotics research*, v. 6, p. 72–83, 1987.

[28] CHIAVERINI, S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, v. 13, p. 398–410, 1997.

[29] SCIAVICCO, L.; SCILIANO, B. A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal of Robotics and Automation*, v. 4, p. 403–410, 1988.

[30] LIEGEOIS, A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems Man and Cybernetics*, SMC-7, p. 868–871, 1977.

[31] CHIAVERINI, S.; SCILIANO, B.; EGELAND, O. Kinematic analysis and singularity avoidance for a seven-joint manipulator. In: *Proccedings of the 1990 American Control Conference*. San Diego: USA, 1990.

[32] BOHIGAS, O. et al. A general method for the numerical computation of manipulator singularity sets. *IEEE Transactions on Robotics*, 2013.

[33] YOSHIKAWA, T. Manipulabillty and redundancy control of robotic mechanisms. In: *Proceedings of the 1985 IEEE International Conference on Robotics & Automation*. St. Louis: USA, 1985.

[34] MARANI, G. et al. A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators. In: *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*. Washington: USA, 2002.

[35] KIM, J. et al. A general singularity avoidance framework for robot manipulators: task reconstruction method. In: *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*. New Orleans: USA, 2004.

[36] STEVENSON, R.; SHIRINZADEH, B.; ALICI, G. Singularity avoidance and aspect maintenance in redundant manipulators. In: *Proceedings of the seventh International Conference on Control, Automation, Robotics and Vision*. Singapore: Republic of Singapore, 2002. p. 857–862.

[37] GUTHART, G. S.; SALISBURY, J. K. In: . [S.l.: s.n.].

[38] FUNDA, J. et al. Constrained cartesian motion control for teleoperated surgical robots. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 12, n. 3, p. 453–465, 1996.

[39] FUNDA, J. et al. Control and evaluation of a 7-axis surgical robot for laparoscopy. In: IEEE. *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on.* [S.l.], 1995. v. 2, p. 1477–1484.

[40] ORTMAIER, T.; HIRZINGER, G. Cartesian control issues for minimally invasive robot surgery. In: *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on.* [S.l.: s.n.], 2000. v. 1, p. 565–571 vol.1.

[41] KUO, C.-h.; DAI, J. S. Robotics for Minimally Invasive Surgery: A Historical Review from the Perspective of Kinematics. In: *International Symposium in the History of Machines and Mechanisms.* [S.l.: s.n.], 2009. p. 337–354.

[42] KOWALCZYK, W.; LAWNICZAK, M. Study on opportunity to use industrial manipulator in surgical-laparoscopic application. In: IEEE. *Robot Motion and Control, 2005. RoMoCo'05. Proceedings of the Fifth International Workshop on.* [S.l.], 2005. p. 63–68.

[43] LOCKE, R. C.; PATEL, R. V. Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery. In: IEEE. *Robotics and Automation, 2007 IEEE International Conference on.* [S.l.], 2007. p. 1900–1905.

[44] MAYER, H.; NAGY, I.; KNOLL, A. Kinematics and Modelling of a System for Robotic Surgery. In: *On Advances in Robot Kinematics.* [S.l.]: Springer, 2004. p. 10.

[45] DALVAND, M. M.; SHIRINZADEH, B. Remote Centre-Of-Motion Control Algorithms of 6-RRCRR Parallel Robot Assisted Surgery System (PRAMiSS). In: . [S.l.: s.n.], 2012. p. 3401–3406.

[46] MICHELIN, M.; POIGNET, P.; DOMBRE, E. Dynamic task/posture decoupling for minimally invasive surgery motions: simulation results. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.* [S.l.: s.n.], 2004. v. 4, p. 3625–3630 vol.4.

[47] AZIMIAN, H.; PATEL, R. V.; NAISH, M. D. On constrained manipulation in robotics-assisted minimally invasive surgery. In: IEEE. *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on.* [S.l.], 2010. p. 650–655.

APPENDIX

I. NOTATIONS

"Such is the advantage of a well-constructed language that its simplified notation often becomes the source of profound theories."
–Pierre-Simon Laplace

I.1 Quaternions & Dual Quaternions

The dual quaternions are the basic building blocks of the kinematic control theory implemented in this work. Therefore, the notation used to describe quaternions and dual quaternions as established by Adorno [16, p. 21-25] is used in this work.

We begin by defining $\hat{\imath}$, $\hat{\jmath}$ and $\hat{k}$ as the three imaginary components of a quaternion such that

$$\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = -1 \text{ and}$$
$$\hat{\imath}\hat{\jmath}\hat{k} = -1$$

The general quaternion $\mathbf{h} \in \mathbb{H}$ was introduced by Hamilton and is given by

$$\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}, \quad h_i \in \mathbb{R},\, i = 1, \ldots, 4,$$

in which the scalar part of $\mathbf{h}$ is given by $\mathrm{Re}(\mathbf{h}) \triangleq h_1$.

We can then define its conjugate as

Definition I.1. *Quaternion conjugate.* The conjugate of a quaternion $\mathbf{h}$ is given by

$$\mathbf{h}^* \triangleq h_1 - h_2\hat{\imath} - h_3\hat{\jmath} - h_4\hat{k}.$$

The addition/subtraction and multiplication operations of quaternions are now defined in the two next definitions.

Definition I.2. *Quaternion sum/subtraction.* The quaternion sum/subtraction of two quaternions $\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$ and $\mathbf{h}' = h_1' + h_2'\hat{\imath} + h_3'\hat{\jmath} + h_4'\hat{k}$ is

$$\mathbf{h} \pm \mathbf{h}' = h_1 \pm h_1' + \hat{\imath}(h_2 \pm h_2') + \hat{\jmath}(h_3 \pm h_3') + \hat{k}(h_4 \pm h_4').$$

Definition I.3. *Quaternion multiplication.* The quaternion multiplication of $\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$ and $\mathbf{h}' = h_1' + h_2'\hat{\imath} + h_3'\hat{\jmath} + h_4'\hat{k}$ is

$$\begin{aligned}
\mathbf{h}\mathbf{h}' =& (h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k})(h_1' + h_2'\hat{\imath} + h_3'\hat{\jmath} + h_4'\hat{k}) \\
=& (h_1h_1' - h_2h_2' - h_3h_3' - h_4h_4') + \\
& \hat{\imath}(h_1h_2' + h_2h_1' + h_3h_4' - h_4h_3') + \\
& \hat{\jmath}(h_1h_3' - h_2h_4' + h_3h_1' + h_4h_2') + \\
& \hat{k}(h_1h_4' + h_2h_3' - h_3h_2' + h_4h_1').
\end{aligned}$$

Whenever a quaternion is used in any operation with a vector or matrix, its vector form must be obtained with the following operator.

Definition I.4. *Quaternion vector form.* Given a quaternion $\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$, it can be mapped into a vector using the vec operator $\text{vec} : \mathbb{H} \to \mathbb{R}^4$; that is,

$$\text{vec}\,\mathbf{h} \triangleq \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \end{bmatrix}^T.$$

This operation can also be performed using Hamilton operators.

Definition I.5. *Hamilton operators.* Let $\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$ and $\mathbf{h}' = h_1' + h_2'\hat{\imath} + h_3'\hat{\jmath} + h_4'\hat{k}$, then

$$\overset{+}{\mathbf{H}}(\underline{\mathbf{h}}) = \begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & -h_4 & h_3 \\ h_3 & h_4 & h_1 & -h_2 \\ h_4 & -h_3 & h_2 & h_3 \end{bmatrix}, \qquad \overset{-}{\mathbf{H}}(\underline{\mathbf{h}}') = \begin{bmatrix} h_1' & -h_2' & -h_3' & -h_4' \\ h_2' & h_1' & h_4' & -h_3' \\ h_3' & -h_4' & h_1' & h_2' \\ h_4' & h_3' & -h_2' & h_1' \end{bmatrix},$$

so that

$$\text{vec}(\mathbf{h}\mathbf{h}') = \overset{+}{\mathbf{H}}(\mathbf{h})\text{vec}(\mathbf{h}') = \overset{-}{\mathbf{H}}(\mathbf{h}')\text{vec}(\mathbf{h}).$$

With the conjugate operator and the quaternion multiplication, we can now introduce the quaternion norm.

Definition I.6. *Quaternion norm.* The norm of a quaternion $\mathbf{h}$ is

$$\| \mathbf{h} \| = \sqrt{\mathbf{h}\mathbf{h}^*} = \sqrt{\mathbf{h}^*\mathbf{h}}.$$

Definition I.7. The dual number algebra is based on the element ε introduced by Clifford. It has the following property

$$\epsilon = 0$$
$$\text{but } \epsilon^2 \neq 0.$$

The dual quaternion $\underline{\mathbf{h}} \in \mathcal{H}$ is a dual number whose elements are quaternions, that is

$$\underline{\mathbf{h}} = \mathbf{h}_1 + \epsilon\mathbf{h}_2.$$

In which the real part of $\underline{\mathbf{h}}$ is

$$\text{Re}(\underline{\mathbf{h}}) \triangleq \text{Re}(\mathcal{P}(\underline{\mathbf{h}})) + \varepsilon\text{Re}(\mathcal{D}(\underline{\mathbf{h}})).$$

The primary and dual parts of the dual quaternion $\underline{\mathbf{h}}$ are quaternions. They can be extracted using the operators $\mathcal{P}(\underline{\mathbf{h}})$ and $\mathcal{D}(\underline{\mathbf{h}})$, respectively. Hence,

$$\mathcal{P}(\underline{\mathbf{h}}) \triangleq \mathbf{h_1}$$
$$\text{and } \mathcal{D}(\underline{\mathbf{h}}) \triangleq \mathbf{h}_2.$$

The dual quaternion sum/subtraction and multiplication are defined bellow.

Definition I.8. *Dual quaternion sum/subtraction.* Let $\underline{\mathbf{h}}, \underline{\mathbf{h}}^{'} \in \mathcal{H}$; the dual quaternion sum/subtraction between them is

$$\underline{\mathbf{h}} \pm \underline{\mathbf{h}}^{'} = \mathcal{P}(\underline{\mathbf{h}}) \pm \mathcal{P}(\underline{\mathbf{h}}^{'}) + \varepsilon(\mathcal{D}(\underline{\mathbf{h}}) \pm \mathcal{D}(\underline{\mathbf{h}}^{'})).$$

Dual quaternion multiplication. Let $\underline{\mathbf{h}}, \underline{\mathbf{h}}^{'} \in \mathcal{H}$; the dual quaternion multiplication between them is

$$\underline{\mathbf{h}}\underline{\mathbf{h}}^{'} = \mathcal{P}(\underline{\mathbf{h}})\mathcal{P}(\underline{\mathbf{h}}^{'}) + \varepsilon(\mathcal{P}(\underline{\mathbf{h}})\mathcal{D}(\underline{\mathbf{h}}^{'}) + \mathcal{D}(\underline{\mathbf{h}})\mathcal{P}(\underline{\mathbf{h}}^{'})).$$

This operation can also be performed using Hamilton operators.

Definition I.9. *Hamilton operators.* Let $\underline{\mathbf{h}}, \underline{\mathbf{h}}^{'} \in \mathcal{H}$, then

$$\overset{+}{\mathbf{H}}(\underline{\mathbf{h}}) = \begin{bmatrix} \overset{+}{\mathbf{H}}(\mathcal{P}(\underline{\mathbf{h}})) & 0 \\ \overset{+}{\mathbf{H}}(\mathcal{D}(\underline{\mathbf{h}})) & \overset{+}{\mathbf{H}}(\mathcal{P}(\underline{\mathbf{h}})) \end{bmatrix}, \qquad \overset{-}{\mathbf{H}}(\underline{\mathbf{h}}^{'}) = \begin{bmatrix} \overset{-}{\mathbf{H}}(\mathcal{P}(\underline{\mathbf{h}}^{'})) & 0 \\ \overset{-}{\mathbf{H}}(\mathcal{D}(\underline{\mathbf{h}}^{'})) & \overset{-}{\mathbf{H}}(\mathcal{P}(\underline{\mathbf{h}}^{'})) \end{bmatrix},$$

so that

$$\mathrm{vec}(\underline{\mathbf{h}}\underline{\mathbf{h}}^{'}) = \overset{+}{\mathbf{H}}(\underline{\mathbf{h}})\mathrm{vec}(\underline{\mathbf{h}}^{'}) = \overset{-}{\mathbf{H}}(\underline{\mathbf{h}}^{'})\mathrm{vec}(\underline{\mathbf{h}}).$$

Definition I.10. *Dual quaternion conjugate.* The conjugate of the dual quaternion $\underline{\mathbf{h}}$ is

$$\underline{\mathbf{h}}^{*} \triangleq \mathcal{P}(\underline{\mathbf{h}})^{*} + \varepsilon\mathcal{D}(\underline{\mathbf{h}})^{*}.$$

Definition I.11. *Dual quaternion vector form.* Given a dual quaternion $\underline{\mathbf{h}}$, it can be mapped into a vector using the vec operator $\mathrm{vec} : \mathcal{H} \to \mathbb{R}^8$; that is,

$$\mathrm{vec}\,\underline{\mathbf{h}} = \begin{bmatrix} \mathrm{vec}(\mathcal{P}(\underline{\mathbf{h}})) \\ \mathrm{vec}(\mathcal{D}(\underline{\mathbf{h}})) \end{bmatrix}.$$

This operation has the following property

$$\mathrm{vec}\underline{\mathbf{h}}^{*} = \mathbf{C}_8\mathrm{vec}\underline{\mathbf{h}}, \qquad \text{with} \qquad \mathbf{C}_8 = diag(1, -1, -1, -1, 1, -1, -1, -1).$$

Definition I.12. *Dual quaternion norm.* The norm of the dual quaternion $\underline{\mathbf{h}}$ is

$$\| \underline{\mathbf{h}} \| = \sqrt{\underline{\mathbf{h}}\underline{\mathbf{h}}^{*}} = \sqrt{\underline{\mathbf{h}}^{*}\underline{\mathbf{h}}}.$$

Dual quaternions with unit norm are called unit dual quaternions. They will receive the most focus on this work as they are used in the representation of rigid motions.

Definition I.13. *Unit dual quaternion logarithm.* Let $\underline{\mathbf{h}} = \mathbf{r} + \varepsilon\frac{1}{2}\underline{\mathbf{t}\mathbf{r}}$ be a unit dual quaternion with $\mathbf{r} = \cos(\frac{\phi}{2}) + \sin(\frac{\phi}{2})\mathbf{n}$, $\mathbf{n} = n_x\hat{\imath} + n_y\hat{\jmath} + \hat{k}n_z$ and $\mathbf{t} = p_x\hat{\imath} + p_y\hat{\jmath} + p_z\hat{k}$. The logarithm of $\underline{\mathbf{h}}$ is

$$\log \underline{\mathbf{h}} \triangleq \frac{\phi\mathbf{n}}{2} + \frac{\mathbf{t}}{2}.$$

An important property of the unit dual quaternion logarithm is $\log \underline{\mathbf{h}} \in \mathcal{H}$ and $\mathrm{Re}(\underline{\mathbf{h}}) = 0$.

Definition I.14. *Unit dual quaternion exponential.* Let $\underline{\mathbf{g}} \in \mathcal{H}$ with $\mathrm{Re}(\underline{\mathbf{g}}) = 0$. The exponential of $\underline{\mathbf{g}}$ is

$$\exp \underline{\mathbf{g}} \triangleq \mathcal{P}(\exp \underline{\mathbf{g}}) + \epsilon \mathcal{D}(\underline{\mathbf{g}}) \mathcal{P}(\exp \underline{\mathbf{g}})$$

$$\mathcal{P}(\exp \underline{\mathbf{g}}) \triangleq \begin{cases} \cos \parallel \mathcal{P}(\underline{\mathbf{g}}) \parallel + \frac{\sin \| \mathcal{P}(\underline{\mathbf{g}}) \|}{\| \mathcal{P}(\underline{\mathbf{g}}) \|} \mathcal{P}(\underline{\mathbf{g}}) & \text{if } \parallel \mathcal{P}(\underline{\mathbf{g}}) \parallel \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

Definition I.15. With the definitions I.13 and I.14, the dual quaternion $\underline{\mathbf{h}}$ raised to the r-th power is

$$\underline{\mathbf{h}}^{\{r\}} \triangleq \exp(r \log \underline{\mathbf{h}}).$$

II. SIMULATION DATA

II.1 Data obtained from user interaction

The form described in section 3.5 is shown in Figure II.1 on page 63. The scale for each field is shown on Table II.1. The Table II.2 on the following page shows the data of each of the 20 forms filled, along how long the user took to finish the task as a performance measure. For illustrative reasons, the RCM error for all experiments are shown in Figure II.2 on page 64 and in Figure II.3 on page 65. Finally, the trajectory of the endoscope tip in all experiments is shown in Figure II.4 on page 66 and in Figure II.5 on page 67.

Laparoscopy	
1	no knowledge
2	knows what it means
3	knows someone who has been through one
4	have performed it in someone else
(Surgical) simulators	
1	no knowledge
2	knows what it means
3	have used it once
4	have used it many times
(haptic) interface	
1	no knowledge
2	knows what it means
3	have used it once
4	have used it many times
(Task) ease	
1	very difficult
2	difficult
3	easy
4	very easy
(Control) intuitivity	
1	very bad
2	bad
3	good
4	very good

Table II.1: Scale for each field that should be filled in the forms handled during the simulated trials.

			Before experiment			After experiment	
Group	Subject	Duration [s]	Laparoscopy	Simulators	Interface	Ease	Intuitivity
A	13	77.8118	1	1	1	4	4
A	14	66.2711	1	1	1	3	4
A	6	44.799	3	3	4	3	4
A	8	64.2854	2	3	3	3	4
A	17	18.2691	3	3	1	4	3
A	18	31.7364	1	1	1	3	3
A	19	30.8229	3	2	4	4	4
A	20	83.9377	2	3	2	2	4
A	1	56.5414	3	2	3	3	3
A	2	48.5659	1	1	3	3	3
B	15	47.7155	2	3	3	3	4
B	16	75.3558	1	1	1	2	4
B	3	114.5695	2	1	1	3	3
B	4	102.8864	1	1	1	3	4
B	5	59.4011	1	1	2	3	4
B	7	92.4051	1	1	1	3	4
B	9	59.2723	3	2	1	4	3
B	10	65.0624	3	3	1	3	4
B	11	71.9864	1	1	1	3	4
B	12	149.0546	1	1	1	2	2

Table II.2: Simulation data acquired from the form with the corresponding duration. The users are sorted in the groups A = untrained, B = trained.

Experimento Piloto – Projeto CLARA
Formulário de Avaliação

Subject No: _____ Grupo: ( ) A ( ) B

Data: ___/___/_____ Horário: ____:____

Nome: _____ Idade: _____

Email: _____

Sexo: ( ) Masculino ( ) Feminino

Antes do experimento

Avalie seu grau de familiaridade com:

	Nenhum	Pouco	Razoável	Muito
Procedimento de laparoscopia				
Simulador cirúrgico				
Interface háptica				

Após o experimento

Avalie o experimento quanto a:

	Muito difícil	Difícil	Fácil	Muito fácil
Realização da tarefa				

	Muito ruim	Ruim	Boa	Muito boa
Correspondência entre os movimentos da interface e os da imagem no simulador				

Observações:

_____

_____

_____

_____

Sugestões:

_____

_____

_____

_____

Figure II.1: The form (in Portuguese) handed for each user.

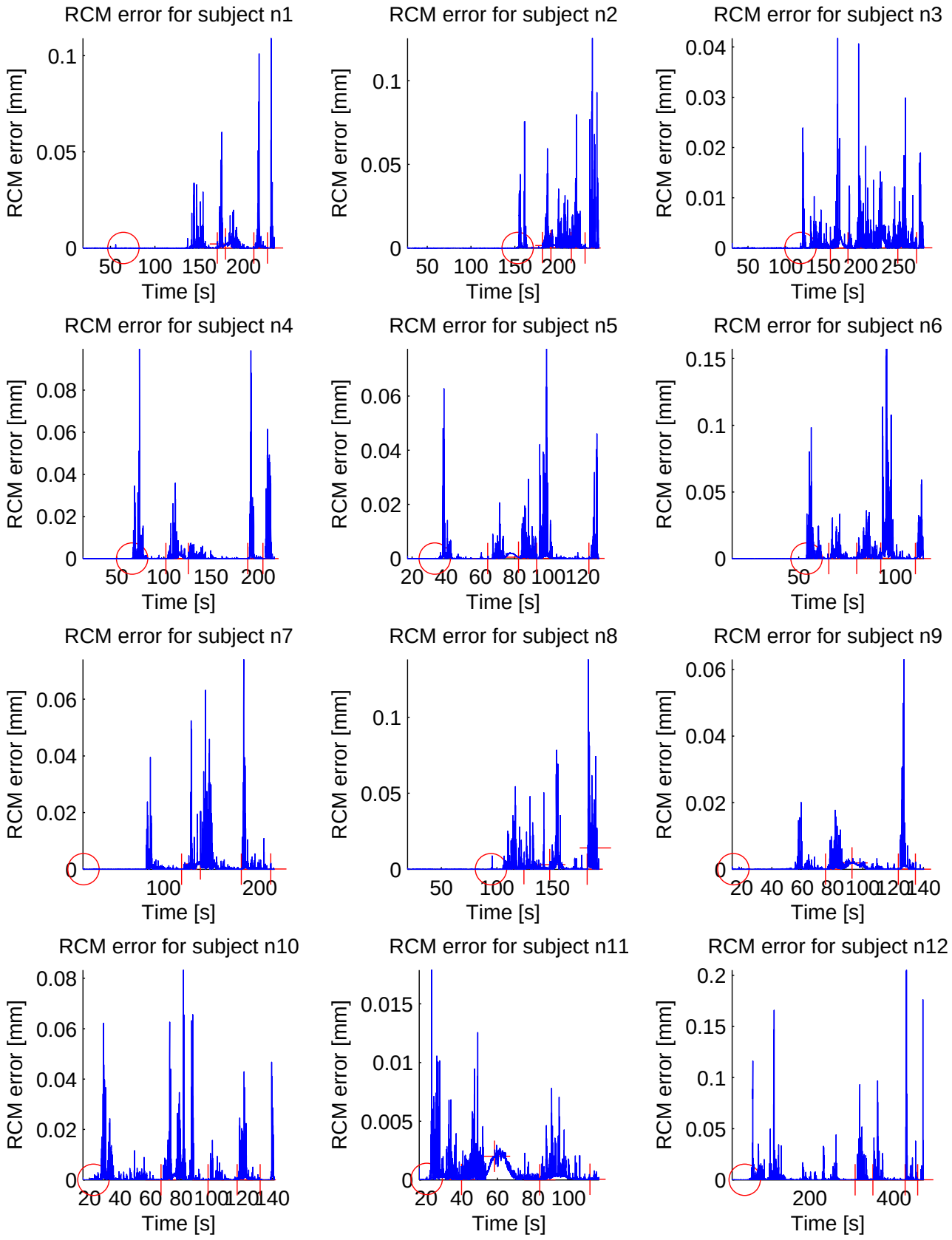

Figure II.2: The RCM error norm under interaction of subjects 1 to 12. The red circle marks the experiment starting point, while the red crosses indicate where each objective was correctly reached.

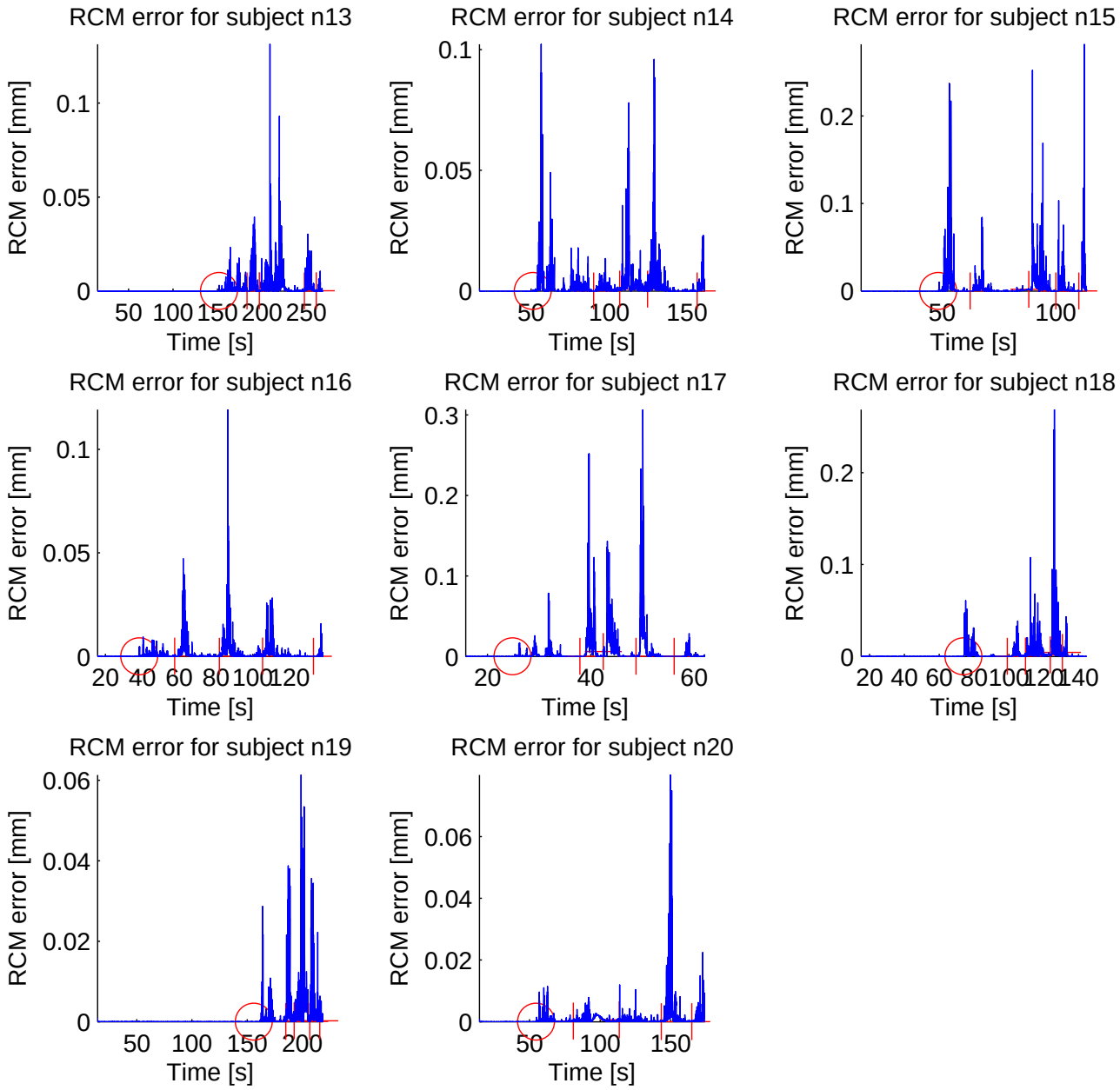

Figure II.3: The RCM error norm under interaction of subjects 13 to 20. The red circle marks the experiment starting point, while the red crosses indicate where each objective was correctly reached.

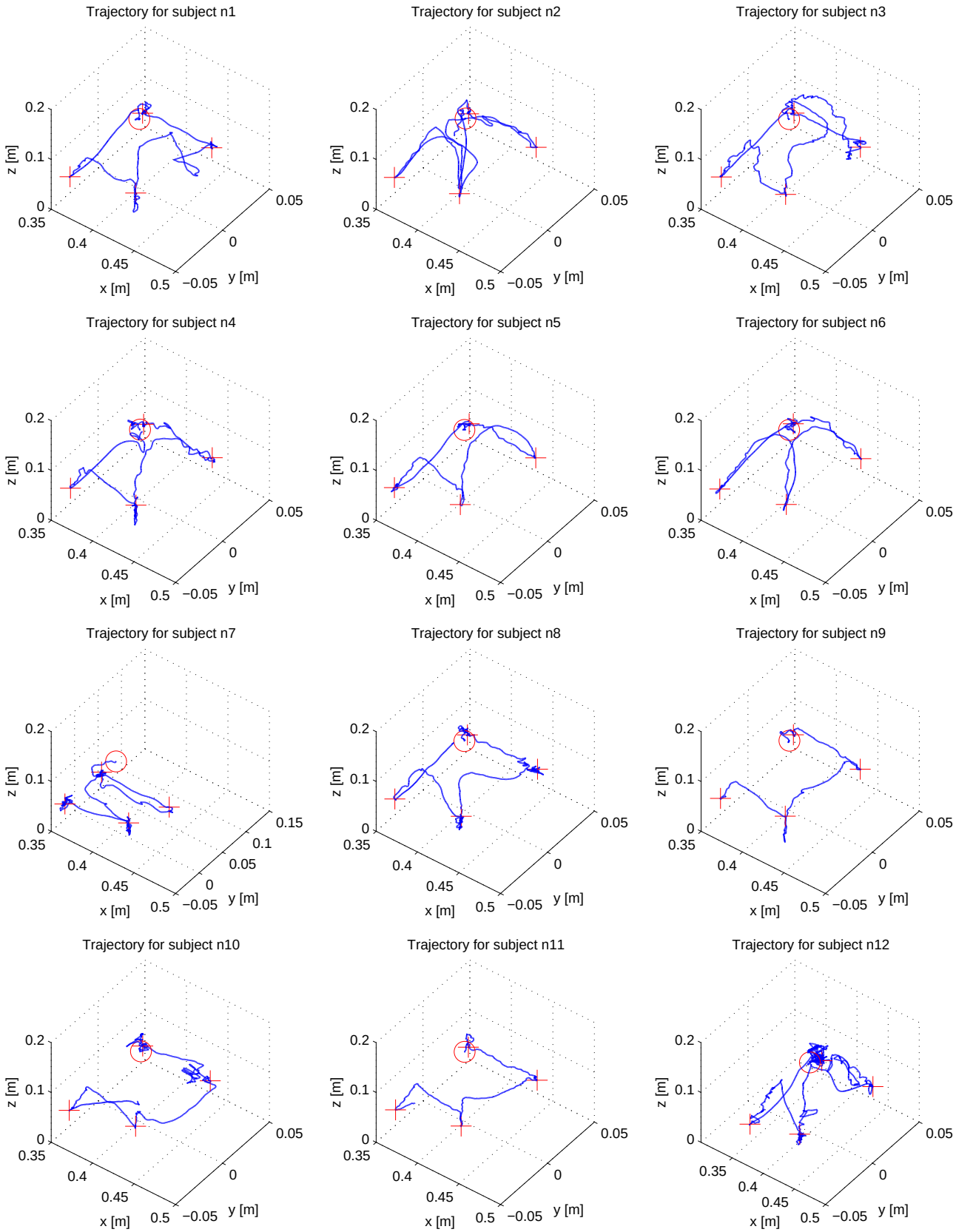

Figure II.4: The trajectory of the endoscope under interaction of subjects 1 to 12. The red circle marks the experiment starting point, while the red crosses indicate where each objective was correctly reached.

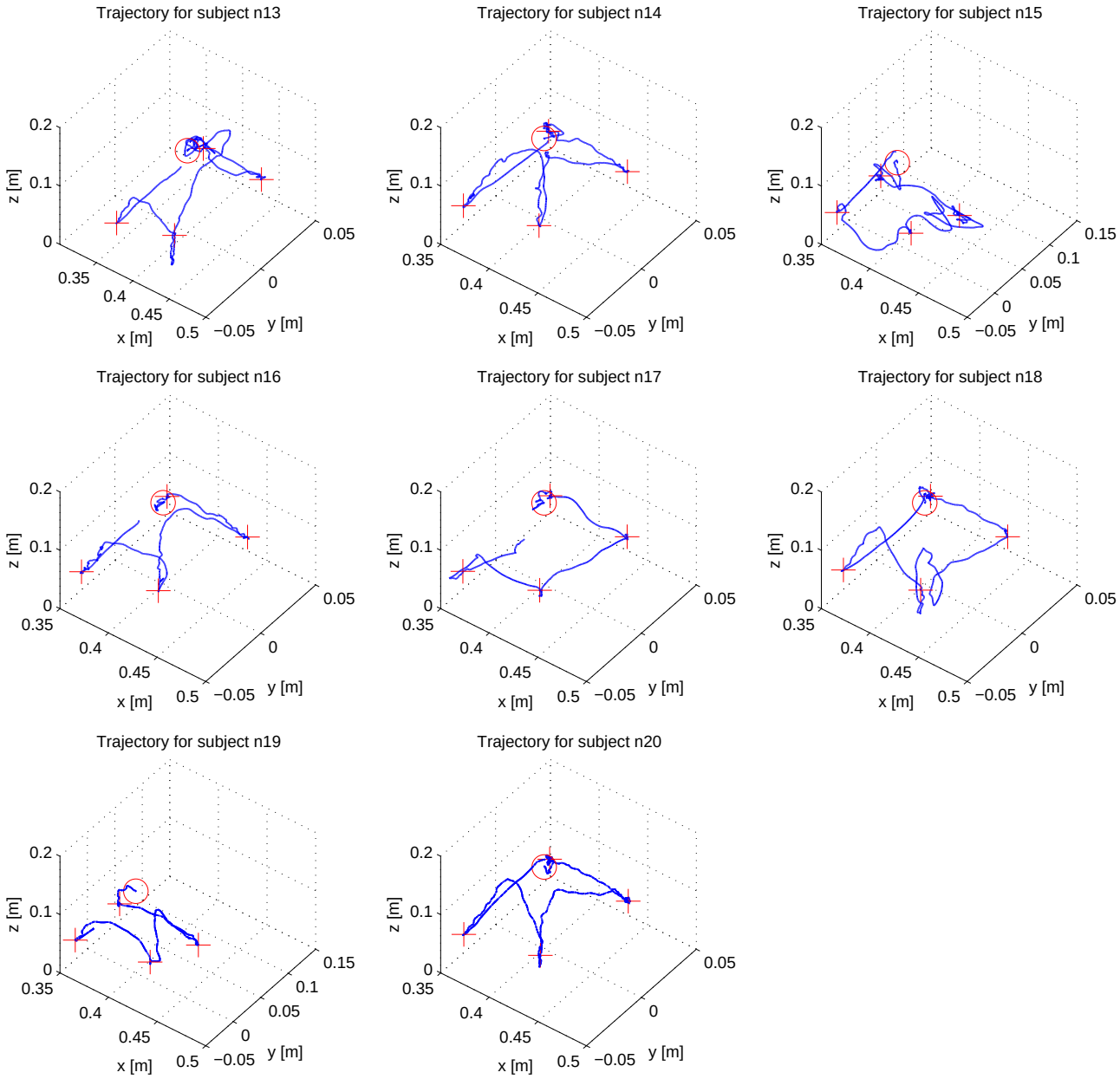

Figure II.5: The trajectory of the endoscope under interaction of subjects 13 to 20. The red circle marks the experiment starting point, while the red crosses indicate where each objective was correctly reached.

II. PHYSICAL EXPERIMENTS

II.1 Experimental setup

Aiming a physical evaluation of the proposed laparoscopic controller, an experimental environment was developed. The full system overview can be seen on Figure II.1. By interacting with the haptic interface, the user sends the desired motion commands to the computer. Those commands are then sent to the computational implementation of the laparoscopy controller, which returns a new set of joint positions that are sent to the robot. After a sampling period, the current manipulator joint positions are sent back to the computer, which simultaneously receives the pose of the visual markers from the visual tracking device. Moreover, the computer periodically receives endoscopic images which are presented in the monitor for the endoscope operator.

Concerning hardware, only the acrylic abdomen was created specifically for this project. Software-wise, all communication between different devices were made using ROS. For this purpose, drivers for the visual tracking device and haptic device were developed. In relation to the manipulator, computational kinematic controller implementations were added to the DQ_robotics library which was also ported to ROS. Concurently, all relevant data is store in MATLAB compatible files using custom software.

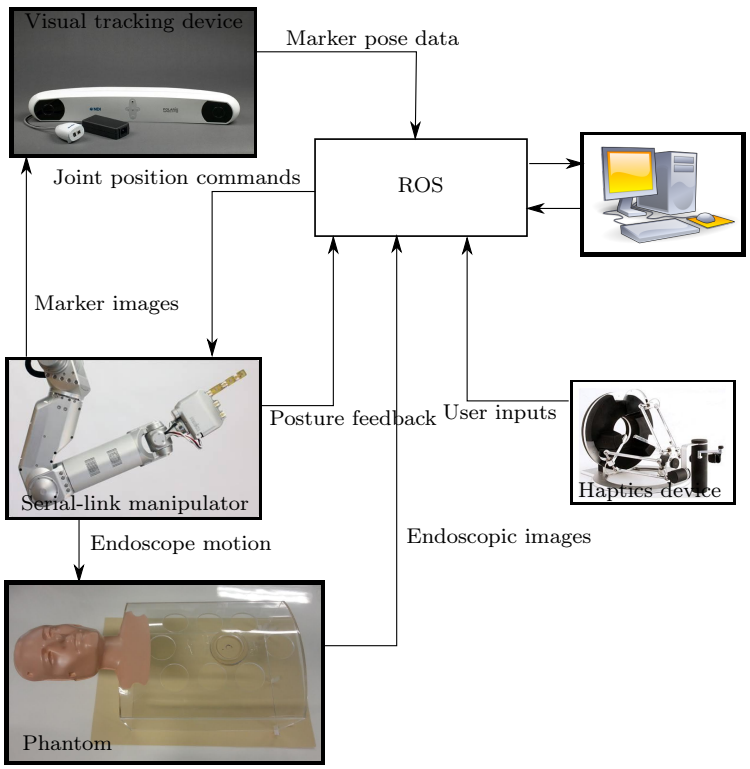

Figure II.1: Physical experimental setup.