GUIDAGE ROBOTISÉ DES AIGUILLES FLEXIBLES POUR DES PROCÉDURES PERCUTANÉES

MARIANA COSTA BERNARDES

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

GUIDAGE ROBOTISÉ DES AIGUILLES FLEXIBLES POUR DES PROCÉDURES PERCUTANÉES

MARIANA COSTA BERNARDES

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.

APROVADA POR:

Geovany A. Borges, ENE/UnB
Orientador

Philippe Poignet, LIRMM/UM2
Orientador

Jocelyne Troccaz, CNRS/TIMC
Examinador Externo

Guillaume Morel, ISIR/UPMC
Examinador Externo

Chafiaa Hammitouche, ITI/Telecom Brest
Examinador Externo

Nabil Zemiti, LIRMM/UM2
Examinador Interno

MONTPELLIER, 19 DE DEZEMBRO DE 2012.

FICHA CATALOGRÁFICA

BERNARDES, MARIANA COSTA Guidage Robotisé des Aiguilles Flexibles pour des Procédures Percutanées [Distrito Federal] 2012. xi, 127p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2012). TESE DE DOUTORADO – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica 1. guiagem de agulha   2. planejamento de trajetória 3. robótica cirúrgica   4. agulhas flexíveis I. ENE/FT/UnB   II. Título (série)

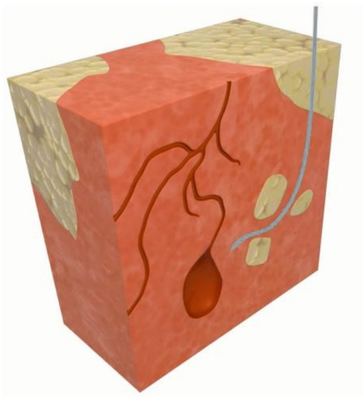
REFERÊNCIA BIBLIOGRÁFICA

BERNARDES, M.C. (2012). Guidage Robotisé des Aiguilles Flexibles pour des Procédures Percutanées, TESE DE DOUTORADO em Engenharia Elétrica, Publicação PPGEA.TD.068/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 127p.

____
Mariana Costa Bernardes
Departamento de Eng. Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

ROBOT-ASSISTED STEERING OF FLEXIBLE NEEDLES FOR PERCUTANEOUS PROCEDURES

MARIANA COSTA BERNARDES



Guidage Robotisé des Aiguilles Flexibles pour des Procédures Percutanées

Guiagem Robotizada de Agulhas Flexíveis para Procedimentos Percutâneos

December 19th, 2012

Mariana Costa Bernardes: *Robot-Assisted Steering of Flexible Needles for Percutaneous Procedures*, (Guidage Robotisé des Aiguilles Flexibles pour des Procédures Percutanées, Guiagem Robotizada de Agulhas Flexíveis para Procedimentos Percutâneos), December 19^{th}, 2012

To those who most inspired it and will never read it,
Tiago, Antonio, Virgínia, and Fernanda.

ABSTRACT

This thesis proposes a robot-assisted approach for automatic steering of flexible beveled needles in percutaneous procedures. The method uses duty-cycled rotation of the needle to perform insertion with arcs of adjustable curvature, and combines closed-loop imaging feedback with an intraoperative motion replanning strategy to compensate for system uncertainties and disturbances. Differently from previous approaches, the closed-loop replanning strategy is suitable for dynamic scenes that present changes of obstacles and target positions. Indeed, we implemented the proposed system using a robotic manipulator, and the results obtained from *in vitro* tests confirmed the viability of our method. To the best of our knowledge, such results are original, specifically in what concerns the use of an intraoperative fast replanning strategy combined with needle duty-cycling and the use of a commercially available manipulator arm.

RÉSUMÉ

Les travaux de cette thèse proposent une nouvelle approche pour le guidage assisté par robots d'aiguilles flexibles pour des procédures percutanées. La méthode est basée sur l'utilisation d'une rotation de l'aiguille avec un rapport cyclique variable pour réaliser une insertion avec des arcs de rayons de courbure différents. Elle combine un retour visuel avec une stratégie de planification adaptative pour compenser les incertitudes du système et les perturbations. Par rapport aux approches présentées précédemment dans la littérature, la stratégie de planification en boucle fermée est adaptée à des scènes dynamiques qui présentent des changements de position des obstacles et de la cible. Cette approche a été implémentée sur un système robotique et les résultats obtenus in vitro confirment tout l'intérêt de cette technique.

RESUMO

Esta tese propõe uma estratégia de guiagem robotizada de agulhas flexíveis com ponta chanfrada em procedimentos percutâneos. O método usa *duty-cycle* de rotação para realizar inserções com arco de curvatura ajustável e combina realimentação de imagem em malha fechada com uma estratégia intraoperatória de replanejamento de movimento para compensar incertezas e distúrbios no sistema. Diferentemente de estratégias anteriores, o replanejamento em malha fechada é adequado a cenários dinâmicos em que há mudanças na posição do alvo e obstáculos. De fato, o sistema proposto foi implementado utilizando um robô manipulador, e os resultados obtidos de testes *in vitro* confirmaram a viabilidade do método. Até onde se sabe, tais resultados são originais, especialmente no que diz respeito ao uso de planejamento rápido intraoperatório combinado com *duty-cycle* da agulha e o uso de um braço manipulador disponível comercialmente.

*"Blessed are those who give without remembering.
And blessed are those who take without forgetting."*

— Bernard Meltzer

ACKNOWLEDGMENTS

One of the joys of finishing a thesis is to look over the past and remember all the people who have supported me along this long process. The completion of this Ph.D. would not have been possible without the aid of countless friends from both Brazil and France, who have greatly influenced me during this journey. Now, it is finally the opportunity to thank them.

First, I would like to express my gratitude to my two advisors, Prof. Philippe Poignet and Prof. Geovany Araújo Borges, who are not only mentors but dear friends. I could not have asked for better role models on both academic and personal levels. I would also like to thank my examiners, Jocelyne Troccaz and Guillaume Morel, who provided encouraging and constructive feedback. It is no easy task reviewing a thesis, and I am grateful for their thoughtful and detailed comments.

Furthermore, I am thankful to the reviewers Chafiaa Hammitouche and Nabil Zemiti, for their insightful comments both in my work and in this thesis. I sincerely acknowledge Prof. Etienne Dombre for the very helpful remarks and encouragement he gave me whenever he was able to attend my presentations. A special thanks goes to Philippe Fraisse, Christine Azevedo-Coste, and Mitsuhiro Hayashibe, whose presence in my daily life served as inspiration and as example of the kind of scientist that I wish to be.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) through the Colégio Doutoral Franco-Brasileiro Program (CDFB), and the scholarship program Demanda Social. Thanks also to the ANR ContInt USComp project, through which I had the necessary structure to develop my research activities, and its project manager, Alexandre Krupa, who has been extremely supportive during our joint laboratory activities. In addition, I would like to express my gratitude to Michaela Müller and Euroflex GmbH, for their high quality nitinol samples.

It is also a pleasure to pay tribute to Murilo Marinho and André Augusto Geraldes, the *needle team* . Their enthusiasm, intensity, and willingness to repeat experimental trials over and over again were a great source of motivation. My thanks to Marcella Cortat, who experimented and shared with me countless phantom recipes. I would also acknowledge Éderson Dorileo for his invaluable contribution with the image processing module, and for our many motivating discussions that I hope may result in a future cooperation.

The researchers and students of LIRMM have contributed immensely to my work and I would like to thank them for welcoming me as a friend and helping to develop the ideas in this thesis. I am especially grateful for the support of Pawel Maciejasz, Divine Maalouf, Alonso Sanchez, Chao Liu, Johann Lamaury, Alejandro González de Alba, Minh-Quyen Le, Kanty Rabenorosoa, and Nicolas Carlési. Other past students members that I have had the pleasure to work with or alongside of are Alfredo Toriz, Andreaa Ancuta, Lotfi Chikh, Giulia Toncelli, Szabolcs Fodor, Qin Zhang, and Rafael Sobral; and the numerous summer and master students who have come through the laboratory. To the staff at LIRMM, in particular Nicole Gleizes, Nicolas Serrurier, and Laetitia Megual, I am grateful for their solicitude and efficiency every time an administrative demand was necessary.

My time in France was also made enjoyable in large part due to the many people that became a part of my life. I am grateful for Tamara Chomont's company in our memorable trips, and for the pleasant moments spent at LIRMM's cafeteria with the folks from Microelectronics: Gabriel Marchesan, Luís Vitório Cargnini, João Azevedo, Raphael Brum, Carolina Metzler, Guilherme Botorin, Bruno Paiva, Rafael Garibotti, Leonardo Zordan, and Luciano Ost (see, I did include your name in a publication!). Also, I could not forget the many *soirées* at Montpellier in the presence of Rogério Richa, Daniella Pingret, Natalia Nicole, Fernando and Giselle Fontainha, Flávia Calmon, Guilherme Sartori, Pedro Heitor Barros and Mariana Ferreira.

I am also in debt with my colleagues from Brazil, who have helped make my learning an enjoyable and stimulating experience. The LARA laboratory has always been a source of friendships as well as good advice and collaboration. My special thanks go to Luís *Postinho* Figueredo, Henrique Menegaz, *Alberto* Roberto Baptista, Eduardo Alves, Glauco Scandaroli, Pedro π Santana, Claudia Ochoa, David Fiorillo, George Brindeiro, Felipe *C. Beckmann* Brandão, Bruno Amui, Pedro *Fashion* Dória, Rafael Cortes, and Thiago Rocha, who have stuck with me through the highs and lows of my academic life. In addition, I could always count on Pedro and Renata Moreira, Bruno Adorno, Polyana Pereira, Antônio Bó, and Jovana Jovic, who gave me a home away from home. You have been like a second family, bearing the brunt of the frustrations, and sharing the joy of the successes; thank you for just being there for me.

Lastly, I wish to acknowledge my family, which has put up with my constant absences during the past years. Without their unconditional support, this journey would not have been possible. In special, I thank my parents, Antonio and Virgínia, who instilled within me a love of creative pursuits and science, all of which finds a place in this thesis. My sister, Fernanda, has also been the best of friends and her incentive during the final stages of this Ph.D. is so appreciated. And most of all, I wish to express my heartfelt gratitude for my loving, supportive, and patient husband Tiago, who encouraged me to accept the co-tutelle even if it meant that we would be apart for two years with an ocean between us (thank you, *mon amour*! All this sacrifice has finally paid off).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CT Computed Tomography

DOF Degree of Freedom

FDA Food and Drug Administration

FEM Finite Element Method

FKM Forward Kinematic Model

IHP Infinite Horizon Programming

LARA Laboratório de Automação e Robótica

LIRMM Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

MDP Markov Decision Process

MPC Model Predictive Control

PEIT Percutaneous Ethanol Injection Therapy

SMA Shape Memory Alloy

RFA Radio-Frequency Ablation

RFID Radio-Frequency Identification

ROI Region of Interest

RPP Randomized Path Planner

PRM Probabilistic Roadmap

RRT Rapidly-exploring Random Trees

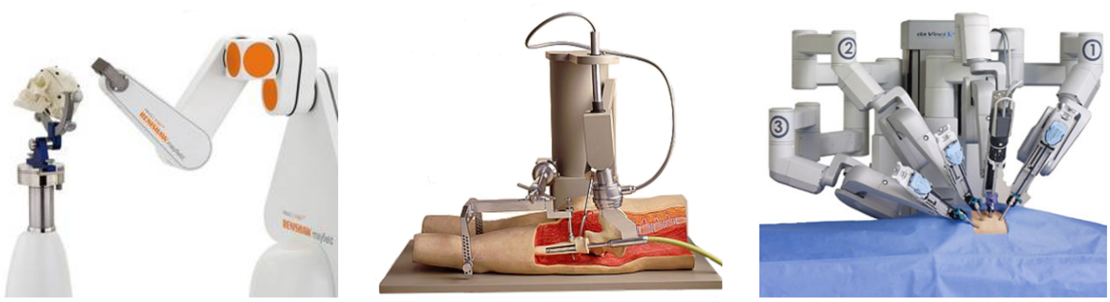
US Ultrasound

INTRODUCTION

In 1985, a PUMA 560 robot was used to place a needle for brain biopsy using Computed Tomography (CT) guidance in the first reported robot-assisted surgical procedure in history (Kwoh et al., 1988). Two years later, Benabid et al. (1987) experimented with an early precursor to the robot marketed as NeuroMate (Fig. I.1a), which was the first neurorobotic device to be approved by the Food and Drug Administration (FDA), as well as the first to be commercially available. Preoperative imaging helped the surgeon to plan the procedure, and a passive robotic arm was able to perform limited tasks in over 1000 procedures (McBeth et al., 2004).

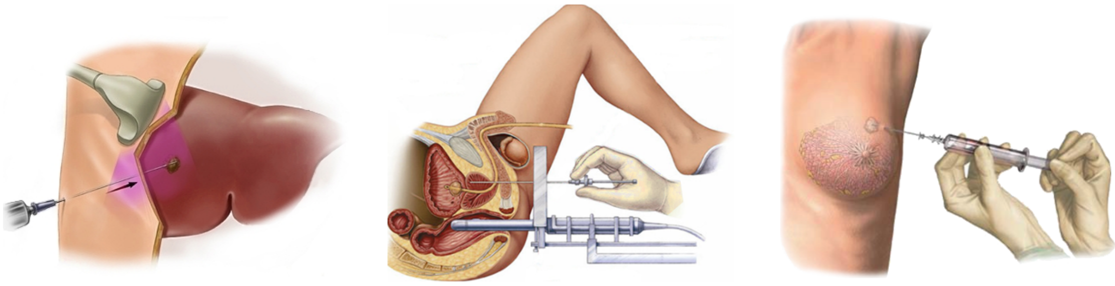
In 1991, Integrated Surgical Systems introduced the ROBODOC (Fig. I.1b) to precise core out the femur in hip replacement surgery. Further development of robotic systems was carried out by Intuitive Surgical with the introduction of the da Vinci (Fig. I.1c). The da Vinci System is FDA approved for a variety of surgical procedures including surgery for prostate cancer, hysterectomy and mistral valve repair, and is used in more than 1,785 hospitals worldwide. Almost three decades after the first initial efforts, medical robotics has become an expanding field of research and surgical robots are being adopted in a wide variety of medical interventions, especially in minimally invasive procedures.

A minimally invasive procedure is any procedure that is less invasive than open surgery used for the same purpose. It is carried out by entering the body through the skin or through a body cavity or anatomical opening, but with the smallest damage possible to these structures. When compared to the equivalent invasive procedures, minimally invasive interventions are significantly becoming the preferred approach since they offer outstanding advantages like less pain, smaller scars and faster recovery time. They also reduce the risk of post operative infection and other complications such as adhesion, intra-operative blood loss, and tissue trauma. Due to these advantages, surgeons are attempting



(a) (b) (c)

Figure I.1: Robot-assisted minimally invasive procedures: *(a)* neurosurgery with Neuromate robot (courtesy of Renishaw plc); *(b)* hip replacement with ROBODOC robot (courtesy of Curexo Technology Corp.); *(c)* laparoscopic surgery with da Vinci robot (courtesy of Intuitive Surgical Inc.).



(a) (b) (c)

Figure I.2: Common percutaneous procedures: *(a)* PEIT treatment for liver cancer (courtesy of Johns Hopkins University); *(b)* prostate brachytherapy (courtesy of the Brachytherapy Advisory Group); *(c)* breast biopsy (courtesy of ADAM Ebix Inc.).

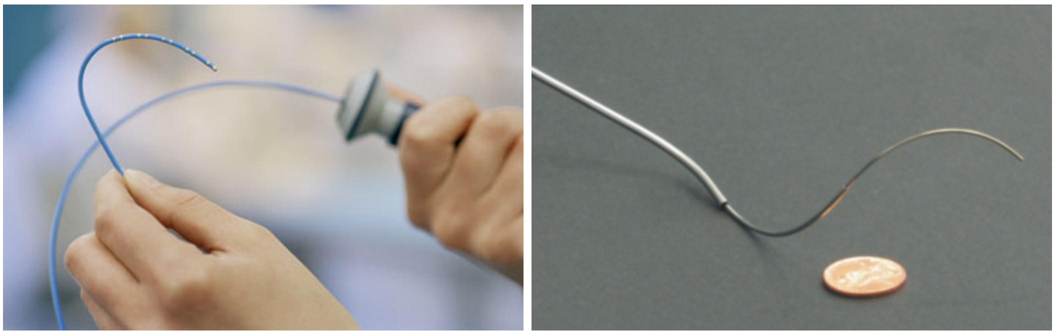
to perform more medical interventions as minimally invasive procedures (Fichtinger et al., 2008).

Percutaneous procedures are considered to be one of the simplest and most minimally invasive medical procedures. More generally, "percutaneous", from its Latin roots, means "by way of the skin" (Collins English Dictionary, 2012) and this type of clinical intervention provides access to inner organs or other tissue by puncturing the skin with thin tubular devices like needles, catheters, tissue ablation probes and etc. The benefit of a percutaneous access is in the ease of introducing devices into the patient without the use of cuts, which can be painful and in some cases can bleed out or become infected. A percutaneous access requires only a very small hole through the skin, which seals easily, and heals very quickly compared to a surgical cut down.

Many medical interventions and diagnosis make use of percutaneous access which already comprise a substantial fraction of minimally invasive procedures. For instance, percutaneous therapy has become a major cancer treatment method (Fig. I.2). Percutaneous Ethanol Injection Therapy (PEIT) and Radio-Frequency Ablation (RFA) are currently performed for liver cancer. In these treatments, a needle is inserted into the cancerous tissue and the tumors are ablated either by the injection of ethanol or by the heat generated with high frequency currents. Brachytherapy is a form of radiotherapy commonly used as an effective treatment for cervical, prostate, breast, and skin cancer and can also be used to treat tumors in many other body sites. In this therapy, a number of small radioactive seeds are permanently implanted inside or next to the area requiring treatment by the use of needles. Other common examples of percutaneous procedures include drainage of fluids, vascular interventions, anesthesia, blood sampling, neurosurgery, biopsy, and etc.

Percutaneous diagnoses and local therapies normally depend on precise positioning of the medical instrument for effectiveness (Abolhassani et al., 2007). Brachytherapy seeds, for instance, must be correctly placed to assure preplanned optimal dosage. Complications can arise in biopsy, in which malignancies may not be properly detected due to miss positioning of the needle tip. In anesthesia, improper needle placement can create traumatic effects and even tiny errors may be fatal to the patient in neurosurgery.

Such complications are usually due to poor technique and incorrect trajectory (Glozman & Shoham, 2007). Several factors, including errors in insertion location, deflection of the instrument, and tissue deformation may contribute to final misplacement, what requires



(a) (b)

Figure I.3: Steerable medical devices: *(a)* steerable diagnostic catheter (courtesy of Biotronik SE & Co.KG); *(b)* concentric tube steerable needle (courtesy of Vanderbilt University).

excellent 3D spatial reasoning and extensive experience from the physician to compensate for their effects and manually correct the trajectory (Webster III et al., 2006a).

Real-time visual feedback has been proved to significantly enhance human precision in percutaneous procedures (Gerovich et al., 2004), but it has been observed that a clinician has limited control over the needle path once it is inserted into the tissue. As a consequence, it is very common that an insertion procedure results in a different trajectory from that defined in preoperative planning. When significant errors occur, the usual solution involves retraction and reinsertion of the device, what causes extra injury to the patient and goes against the idea of minimal tissue damage. Thus, many studies have explored ways to improve the accuracy of percutaneous procedures using medical imaging associated with robotics.

The execution of percutaneous procedures in the presence of obstacles is another important clinical problem to be considered. Sometimes, targets are located in regions of difficult access which cannot be reached by rigid medical devices without causing excessive, injurious pressure on tissue. This issue is critical in cases where the insertion path is obstructed by vital organs, bones, nerves or vessels, since such devices cannot curve around anatomical structures. One possible solution is the use of flexible steerable instruments, which have potential to be clinically used in a number of minimally invasive procedures like vascular and cardiac surgery(Fu et al., 2009), cochlear implants (Zhang et al., 2010), and sinuses and skull base surgery (Webster III & Romano, 2009; Torres & Alterovitz, 2011).

In general, one can split steerable medical devices into two main categories like illustrated in Fig. I.3—steerable catheters and steerable needles. Catheters are typically inserted into either fluid or open space inside the body so their tips can be manipulated with minimal resistance. On the contrary, needles are typically used to target lesions in soft tissue for biopsy, ablation or drug delivery. While both have steering capabilities, catheters pass through channels within the body, and therefore are designed to steer in free space or fluid-filled conduits. Steerable needles, on the other hand, are designed to cut and maneuver through tissue.

This thesis is the result of a co-tutelle agreement between the Laboratório de Automação e Robótica (LARA), within a project in robot-assisted guidance of medical instruments for cancer therapies, and the Laboratoire d'Informatique, de Robotique et de Microélec-

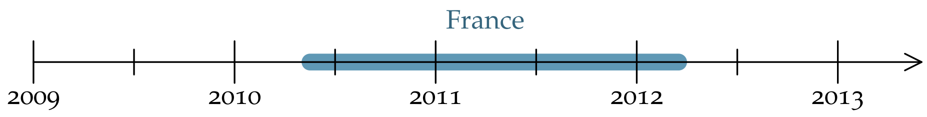


Figure I.4: Co-tutelle timeline.

tronique de Montpellier (LIRMM), in the context of the USComp project whose general objective is to provide methodological solutions allowing real-time compensation of living tissue motion under 2D Ultrasound (US) images. The period of PhD was divided between both countries according to the timeline from Fig. I.4, being the time in France reserved to the development and implementation of the proposed algorithms, as well as the realization of simulations and *in vitro* experiments.

According to the requirements imposed by the two projects, we focused on the problem of steerable needles. Endowing needles with the ability to steer inside the tissue can significantly improve the effectiveness of already existing medical interventions and also expand the applicability of needle-based techniques with the creation of new medical procedures that allow access to targets located deep into soft tissue and previously considered out of reach with straight-line trajectories. More specifically, we consider the use of a robot-assisted system to perform automatic needle steering in closed-loop with image feedback.

THESIS CONTRIBUTIONS

The main contributions of this thesis can be highlighted in three parts:

1. A path planning algorithm, the Arc-RRT, was proposed to obtain feasible trajectories that take the needle from the puncture point to a target located deep into soft tissue while avoiding obstacles that represent anatomical structures. The planner uses explicit geometry to produce movement sequences that respect the needle kinematic model, and is compatible with both pre- and intra-operative use due to its high success rate and fast calculation. It can be used for the general three-dimensional needle insertion and for two-dimensional procedures, which are a special case where the needle path must stay in a constant plane during the whole procedure. The 2D restriction is desirable if the medical imaging modality that is being used provides only planar information, like an ultrasound equipment.

2. A replanning algorithm that uses image feedback from the current position of the target, obstacles and needle tip was adapted from the Arc-RRT. This resulted in a closed-loop strategy to control the needle trajectory and compensate intraoperatively for uncertainties like tissue deformation, model approximations, and changes in the target and obstacles locations. As a result, the needle is able to reach the target with satisfactory precision and to avoid the obstacles even under presence of disturbances.

3. We proposed a system for automatic needle steering using a robotic manipulator arm, whose gripper was adapted to receive the needle base and a telescopic tube that prevents needle buckling. A steering controller has been developed to compute the manipulator joints movements that will result in the planned needle trajectory. As a

result, we obtained a system that integrates the planning and replanning strategies to a robotic system for 2D needle steering with image tracking of the needle tip.

ORGANIZATION OF THE THESIS

The thesis is organized into six chapters, summarized as follows:

Chapter 1 presents some of the most recent developments in needle steering. Also, the main steering techniques are enumerated with their respective benefits and drawbacks, providing the background and motivation for the approach developed in this thesis.

Chapter 2 introduces the kinematic model for steerable needles and compares the different insertion techniques for needle steering.

Chapter 3 proposes a path planning method for steerable needles in both 2D and 3D tasks. It also presents a replanning strategy to correct the needle trajectory intraoperatively from image feedback information.

Chapter 4 proposes a steering control to perform the needle insertion with a robotic manipulator arm with six DOF. It coordinates the rotation and insertion needle movements to obtain the desired path curvature and consequently, the planned trajectory.

Chapter 5 presents the complete system for automatic needle steering and the experimental results obtained from *in vitro* tests with the proposed platform.

Chapter 6 consists of the concluding remarks and the discussion of some open problems in needle steering research.

1 STATE OF THE ART

Conventional straight needles are widely used in surgical procedures, but they present a major drawback—they cannot perform complex curved trajectories to access difficult targets without causing excessive injurious pressure on tissue. Needle steering is a recent field of study that proposes the use of different techniques to guide needles once they are inserted inside the tissue in order to reach targets inaccessible by a straight-line trajectory while avoiding obstacles.

Robot-assisted needle steering may not only improve already existing procedures, but also allow the development of novel medical techniques that might profit of increased accessibility of targets and more dexterous control of the needle path. There are still several open problems that need to be addressed before needle steering systems are commercially available, but recent research has resulted in great advances that show their potential clinical use in needle-based interventions like transperineal prostate brachytherapy (Reed et al., 2011), renal biopsy and nephrolithotomy (Wood et al., 2010a), breast biopsy (Vancamberg et al., 2010), and neurosurgery (Engh et al., 2006b).

Recently, several methods for subcutaneous needle control have arisen, each one being more or less suitable for an specific application, depending upon a compromise between safety, maximum insertion length inside the body, and needle steerability, which is defined as the needle capability to bend and perform curved paths inside tissue. However, if desired, these methods can be combined in order to achieve higher steerability at multiple insertion depths while minimizing tissue damage (Reed et al., 2011).

This chapter offers a general review on the most common needle steering methods and presents the state of the art in steerable needle technologies. For further information on the subject, Abolhassani et al. (2007) provided a survey on the fundamentals of robotic needle insertion in soft tissue while Cowan et al. (2011) introduced a summary of recent research being conducted in the area of steerable needles.

1.1 STEERING APPROACHES AND DEVICES

Steering methods can be roughly divided into active and passive. In active steering, the needle has moving elements that can be actuated inside the tissue to steer the needle. On the contrary, in passive approaches, the needle is manipulated from outside the tissue in order to change its path within it.

Passive steering depends on the interaction forces between needle and tissue which can be complex to determinate, especially when the needle has to penetrate various tissues with heterogeneous properties. In cases like RFA procedures, where the needle has to pass through the derm, the fat, the diaphragm, liver parenchyma and other layers until reaching the target, active needles seem like a promising alternative (Li et al., 2009).

The first design concept for an active needle was presented by Yan et al. (2007), and was based in the deposition of piezoelectric materials along the needle shaft to work as the actuator. However, simulation results indicate that the obtained tip deflection was not

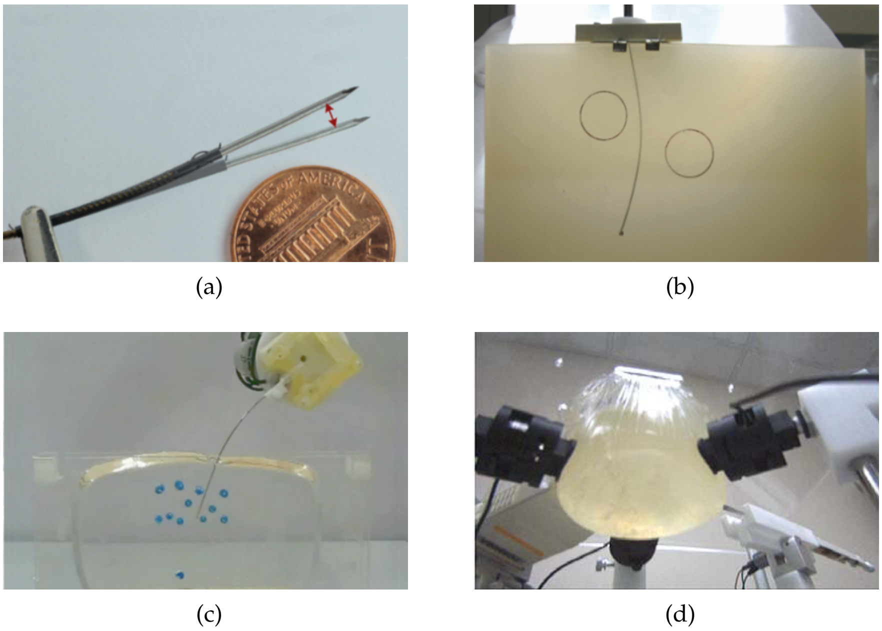


Figure 1.1: Needle steering approaches: *(a)* active needle (courtesy of Stanford University); *(b)* beveled needles; *(c)* base manipulation (courtesy of Guide-X Ltd.); *(d)* tissue manipulation (courtesy of Vanderbilt University).

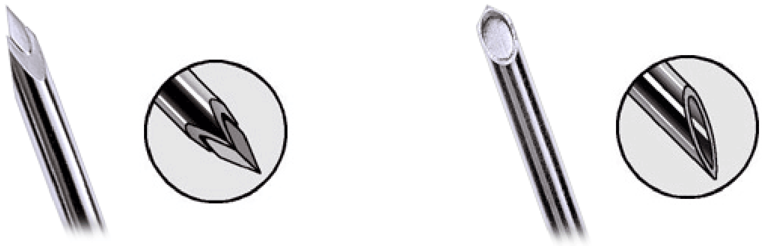
sufficient to provoke significant orientation change in the needle path. More recently, Ryu et al. (2012) presented a prototype for active needle with a flexible joint near the tip that is actuated by a distributed optical heating method combined with Shape Memory Alloy (SMA) (see Fig. 1.1a). Nevertheless, many improvements still need to be incorporated to the current design until a final product can be proposed for clinical trials.

The main disadvantage of using active steering is that it involves the use of sophisticated and complicated mechatronic parts that bring new risks when inserted into the body. A small electrical or mechanical failure could cause catastrophic results for the patient. The development of devices that are both compact and safe enough for performing active steering is a great engineering challenge.

Alternatively, in passive needle steering approaches (see Fig. 1.1b-d), all the electromechanical mechanisms remain outside the patient, enabling the use of thinner needles, larger and lower cost actuators, and a clearer path to clinical application. Passive needle bending has been achieved with two different approaches—one may use the needle to manipulate the tissue or use the tissue to manipulate the needle.

In the first approach, the needle is stiff relative to the tissue and steering is an effect of significant tissue deformation (Glozman & Shoham, 2007). In opposition, the second approach uses very flexible needles relative to the tissue and a pre-bent or asymmetric bevel tip so that bending is achieved by needle deflection, without large displacement of tissue (Engh et al., 2006a). The method of steering via deformable tissue seems to have a large steering capability at shallow depths, and this ability degrades as depth increases. On the other side, using flexible needles may generate less steering at shallow depths, but their steering capability does not degrade with depth (Webster III et al., 2006a).

In the next subsections, we present some passive strategies in more detail.



(a) (b)

Figure 1.2: Common biopsy needles and corresponding tip geometry: *(a)* franseen needle (symmetric) ; *(b)* chiba needle (asymmetric).

1.1.1 *Beveled needles*

Conventional needles can be classified according to its tip shape as symmetric or asymmetric as shown in Fig. 1.2. It is a known effect that when a needle with asymmetric tip is inserted into tissue, the shape of the tip and its interaction with the medium creates an imbalance in the lateral forces, resulting in larger bending if compared to needles with symmetric tips (Okamura et al., 2004).

As a consequence, when inserted into tissue, an asymmetric needle deflects resulting in a slightly curved path that deviates from a straight line trajectory. While such bending is reduced in clinical practice by making the needle shaft as stiff as possible, tissue inhomogeneity and the asymmetric tip can still cause clinically significant placement error (Webster III et al., 2006a). Physicians frequently try to minimize this bending effect by manually spinning the needle during insertion in a drilling-like motion. This reduces the needle friction and cutting forces, and consequently, its bending.

In contrast, one may intentionally use this bending reaction to his advantage by employing very thin and flexible beveled needles that enhance and magnify the needle deflection effect, allowing curved trajectories that could be used to avoid sensitive or impenetrable areas inaccessible with the conventional technique. During the insertion, the lateral forces cause the beveled needle to bend in the direction of its sharpened tip and follow a circular arc of approximately constant curvature. If the needle is flexible relative to the tissue, the rest of the needle shaft will follow the same path as the tip.

Webster III et al. (2006a) showed that the kinematic model of this type of needle can be approximated by that of a nonholonomic bicycle vehicle with constant steering angle. The needle can follow paths in any plane using only two degrees of freedom—insertion and rotation along its shaft. The direction of the needle motion is controlled by rotating the shaft at its base. Since the needle shaft is surrounded and held in place by the tissue, the base rotation is transmitted to the tip, although some lag can be observed due to torsional stiffness (Reed et al., 2009).

The radius of curvature achieved by the needle depends on the combined geometrical and mechanical properties of the needle and tissue. Different sets of needle-tissue will result in different path curvatures. The addition of a enlarged tip (Engh et al., 2006b) or a bent (Wedlick & Okamura, 2009; Majewicz et al., 2010) near the needle tip have already been used to force bigger curvatures thanks to the larger asymmetry at the tip when compared to a beveled tip alone (see Fig. 1.3a-b).

(a) (b) (c)

Figure 1.3: Strategies for enhancing beveled needle steerability: *(a)* enlarged tip (courtesy of Carnegie Mellon University) ; *(b)* pre-bent tip; *(c)* concentric tubes (courtesy of Vanderbilt University).

It is also possible to dynamically change the path curvature during the procedure by "duty-cycling" the needle spinning velocity (Minhas et al., 2007), by changing the offset of the bevel (Ko & Rodriguez y Baena, 2012), or by inserting and retracting a second concentric pre-bent needle to vary the tip asymmetric surface (Okazawa et al., 2005; Walsh et al., 2011; Paul et al., 2012).

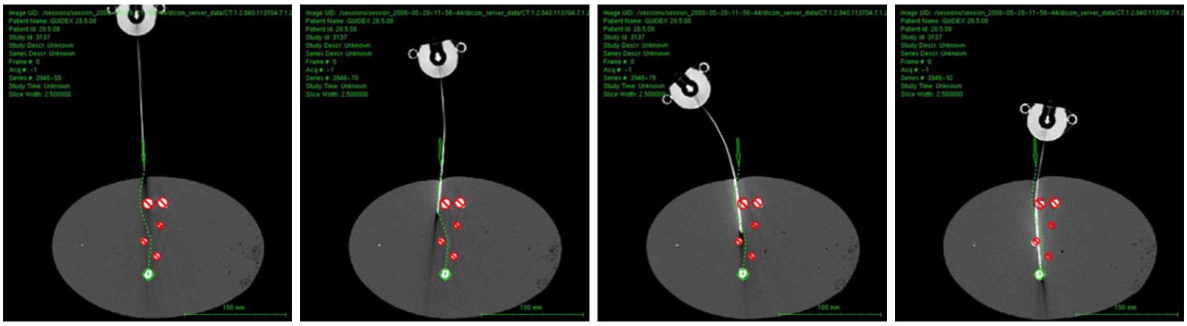
This concentric system can be generalized to include a fixed number of concentric pre-curved tubes that interact with each other to change the device shape. Every tube can be individually rotated or inserted, resulting in a large set of possible tip configurations which do not depend on needle-tissue interaction. This so-called concentric tube robots are needle-like devices (see Fig. 1.3c) that can be controlled to trace curved paths through open air or through tissue as proposed in some recent works (Sears & Dupont, 2006; Webster III et al., 2006b).

Planning and executing needle insertion procedures for flexible beveled needles is a difficult problem. For a human operator, steering a flexible needle by manually actuating at its base is challenging and would require extensive training and experience. This problem can be better handled by the use of a robot-assisted system that combines image guidance and path planning software to compute and perform needle motion.

1.1.2 *Base manipulation*

Another possible method for needle steering involves manipulating the base of the needle perpendicular to the insertion direction (DiMaio & Salcudean, 2003a; Glozman & Shoham, 2007, 2004). The perpendicular motion causes the entire needle shaft to move inside the tissue like a beam inside a compliant fulcrum (Reed et al., 2011). The insertion point acts as the fulcrum, and once the needle is inserted sufficiently far inside the tissue, the base motion causes the needle tip to move roughly in the opposite direction (see Fig. 1.4).

Base manipulation can achieve large changes in the needle path near the tissue surface, but the effect decreases as the needle is inserted deeper since more tissue can resist the lateral force and the moment arm increases. As a consequence, to generate the same change in the tip path throughout the insertion, the force at the base must increase, and the needle is likely to slice through the tissue if too much force is exerted. Since beveled needles are approximately depth independent, base manipulation and beveled needles could be combined for providing additional control over the needle during all the insertion.



(a) (b) (c) (d)

Figure 1.4: Sequence of CT images during a needle insertion with base manipulation steering (courtesy of Guide-X Ltd.).

1.1.3 *Tissue manipulation*

Instead of manipulating the needle to reach a target and avoid obstacles, it is possible to manipulate the tissue in order to move the targets into the needle path or to push obstacles away. Such strategy is often used by physicians which perform it manually, especially in breast biopsy. Inspired by this, recent works have shown that a robotic system can be used to perform image-guided tissue manipulation using blunt-end effectors to achieve the same results in both experiments (Mallapragada et al., 2008; Mallapragada & Sarkar, 2009) and more complex scenarios in simulation (Torabi et al., 2009; Smolen & Patriciu, 2009).

Even though it may be challenging to develop practical mechanisms for deep subsurface targets, robot-assisted tissue manipulation could be used combined with the previously described steering techniques to improve procedure accuracy and target accessibility.

1.2 Robot-assisted needle steering

As previously discussed in the Introduction chapter, robot-assisted needle insertion has been an area of active research in recent years to overcome some of the shortcomings of manual needle insertion. Experiments carried out by different groups have shown that a robotic system can insert needles with consistent precision and enables needle steering to obtain complex trajectories inside the body. A robotic device can also be integrated with medical imaging to combine preoperative models and intraoperative images, cutting down procedure times and resulting in an image-guided system that provides timely and accurate feedback about the patient and the intervention.

The basic process for robot-assisted needle insertion involves the use of patient images to identify targets and obstacles to be avoided within the body, planning the needle trajectory, inserting the needle with the robotic device, verifying the current needle placement, and iteratively adjusting the needle motion if necessary. Thus, a successful robotically controlled needle steering system is normally comprised of a combination of physical systems and computational algorithms (Cowan et al., 2011) as depicted in Fig. 1.5. The computational modules of an image-guided robotic system can be roughly divided in

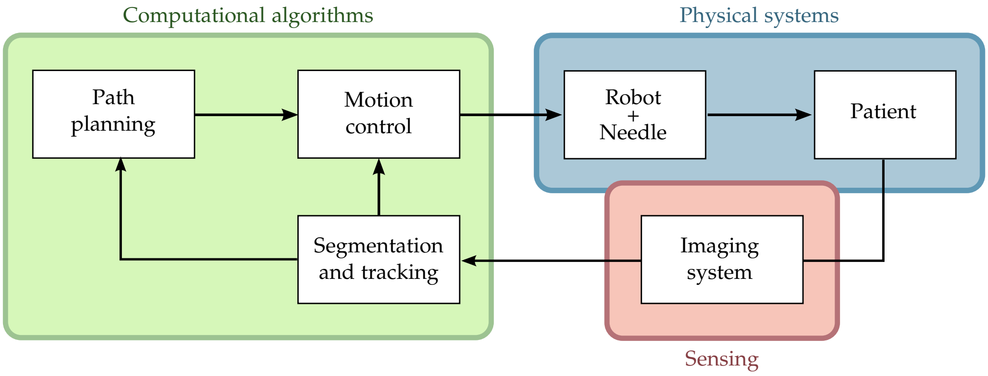


Figure 1.5: Physical systems (blue) and computational algorithms (green) that integrate an image-guided robotically assisted needle steering system. (Adapted from Cowan et al., 2011).

path planning, motion control, and imaging processing. Each one is described in detail as follows.

1.2.1 *Needle steering motion planning*

In percutaneous therapy, it is essential to correctly place the medical device in order to have an effective treatment. For this reason, preoperative planning is normally the first step of a needle steering procedure. For steerable needles, this planning is often beyond the capabilities of human intuition due to the complex kinematics and the effects of tissue deformation, tissue inhomogeneities, and other causes of motion uncertainty. To allow the full potential of needle steering, automatic methods have been developed to help clinicians plan paths and needle motion.

Planning can be used purely preoperatively to generate a plan which will be followed during the procedure by the robot or the physician; or intraoperatively by updating the plan online based on intraoperative images or sensor feedback.

1.2.1.1 *Deformable tissue planning*

The insertion of a needle into soft tissue causes interaction forces between them that result in deformation. Computer simulations that model such forces can be used in preoperative planning to estimate the deformation behavior and optimize paths for needle insertion procedures.

Alterovitz et al. (2005a) used a 2D mesh combined with numerical optimization to compute soft tissue deformations and find a locally optimal path for beveled needle insertions. Based in a similar mesh approach, Chentanez et al. (2009) proposed a 3D needle-tissue simulator aiming at surgical planning for a wide variety of needles. This strategy has also been used by Vancamberg et al. (2010, 2011) to minimize the final error of a Rapidly-exploring Random Trees (RRT) solution in a breast biopsy application.

Such simulations model the interaction between needle and tissue using the Finite Element Method (FEM)—a well known mathematical method for modeling deformations and motions in solids and fluids based on continuum mechanics. However, the accuracy of FEM-based strategies depends a lot on the quality of the mesh simulation and how accurately it represents the real tissue. Also, obtaining a computationally efficient algorithm for FEM meshing is quite challenging. Normally, planners that use finite element methods need several minutes to compute a feasible solution, being more suitable for preoperative planning.

If we specifically consider the base manipulation technique, tissue deformation is essential to obtain needle steerability and should be carefully considered in the planning and control. DiMaio & Salcudean (2003a) introduced a FEM simulation to estimate deformations for tissue and flexible symmetric-tip needles. This method was combined with a numerically obtained manipulation Jacobian and a potential-field-based path planner for planning needle base motion (DiMaio & Salcudean, 2005b). Because of its long computation time, the method was designed for offline planning only. Glozman & Shoham (2004) accelerated it by approximating the tissue with a spring model to compute local, instead of global, deformations and thus, enabling fast online planning.

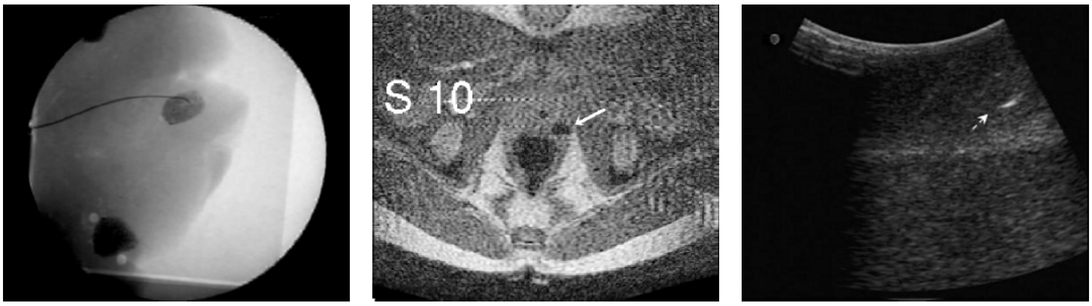
1.2.1.2 *Kinematic planning*

In the beveled needle insertion technique, one may consider that no significant tissue deformation occurs, and only the needle deforms according to a known kinematic model. Many path planning methods based on the kinematic model of beveled needles have already been proposed in the literature. The first work was that of Park et al. (2005), who introduced a diffusion based approach that considers obstacle-free 3D environments. Later, Duindam et al. (2008a) proposed a 3D method for needle path planning which objective is represented numerically as the minimization of a cost function that jointly expresses various objectives: deviation of the final needle tip position from the goal location, required control effort, path length and cost associated with penetration of obstacles.

A different solution to the 3D motion planning problem is presented by Duindam et al. (2008b, 2010), where instead of optimizing a cost function, they used explicit geometric inverse kinematics for 2D and 3D needle motion planning. Xu et al. (2008) were the first to apply RRT-based methods to steerable needle planning and more recently, Lobaton et al. (2011) presented a sampling-based method for planning trajectories with multiple goals. However, none of these methods deal with motion uncertainty caused by modeling approximations, tissue deformation and other interaction forces that may cause the needle to greatly deviate from its planned path.

To tackle the uncertainty issue, Alterovitz et al. (2005a; 2007) considered uncertainty in needle motion by formulating the planning problem as a Markov Decision Process (MDP) using a discretization of the state space, and a Stochastic Roadmap, respectively. In a more recent work (2008), they presented an approach similar to that of (2005a), but adapted to image-guided procedures. Unlike the previous method, in this work they maximize the probability of reaching the target based on parameters that can be extracted from medical imaging without requiring user-specific cost parameters that may be difficult to determinate.

All the presented MDP-based methods have the form of a stochastic shortest path problem and were solved using Infinite Horizon Programming (IHP), which generates a lookup table that allows for instantaneous image-guided control for the steerable needle in a static environment. However, if we consider a dynamic environment that presents tissue and anatomical structures displacement due to patient motion or breathing, the use of pre-computed paths may not be appropriate.



(a) (b) (c)

Figure 1.6: Most common imaging modalities for needle detection: *(a)* fluoroscopic image showing a needle steered into *ex vivo* liver (courtesy of Johns Hopkins University); *(b)* MRI image with arrow pointing to a needle artifact (courtesy of Harvard University) ; *(c)* ultrasound image with arrow pointing to a needle tip (courtesy of Technion).

1.2.2 *Automatic needle tracking in medical images*

Automated detection and tracking of the needle make it possible to provide enhanced navigational cues to the clinician or to employ image-based servoing to perform certain tasks of the procedure using medical robotics. Accurate and up-to-date knowledge of the needle position during the insertion is essential to guide the needle and to confirm if the needle tip is at the expected location. Many different medical imaging modalities can be combined with robot-assisted systems for enhancing the performance of automatic needle insertion as discussed below.

Fluoroscopy and CT

Metal needles have high density and tend to be visible in X-ray images such as CT and fluoroscopy. Such images, can be used to localize and reconstruct a needle in 3D when combined with some prior knowledge about the shaft curve (Heibel et al., 2010). A difficulty in using fluoroscopy is that the device must be precisely calibrated, including the relative pose of the fluoroscopy images (Cowan et al., 2011). In CT imaging there is a trade-off between image quality, frame rate and X-ray dose. Modern CT scanners provide short acquisition time with reasonably low radiation dose which is convenient for intermittent observation of the needle and are also able to produce multiple slices for 3D reconstruction.

However, an universal problem of both X-ray based modalities is the exposition to radiation. To avoid risks to the human operator, the image acquisition is manually triggered which is a time consuming process and subject to errors. Also, to protect the patient, the X-ray dose should be the minimum possible.

X-ray imaging is commonly used for rigid needle insertion in robot-assisted spine procedures (Corral et al., 2004), and biopsies (Kettenbach et al., 2005). This suggests the viability of using this imaging modality also for automatic needle steering. In fact, Glozman & Shoham (2007) have already developed a robotic system for needle base manipulation under real-time fluoroscopic guidance, while other researchers have also considered fluoroscopy for closed-loop steering of beveled needles (Reed et al., 2011). Nevertheless, until now, most of the reported works on robot-assisted needle steering have used this imaging modality only for confirming the final needle placement (Li et al., 2009; Ding et al., 2008).

MRI

The advantage of MRI over X-ray imaging is the absence of harmful radiation. However, there it requires a compromise between resolution and acquisition rate—intraoperative imaging tends to use much lower resolution than diagnostic images, and the acquisition is usually not real-time. Another disadvantage of MRI, is that metal needles create a large signal void in the image, requiring cumbersome artifact localization (Song et al., 2011) or the use of MRI-compatible needles and robotic devices which do not interfere in the magnetic field (Su et al., 2011; Park & Elayaperumal, 2010).

Robotic assistance has already been investigated for automatic insertion of rigid needles under MRI guidance in some medical applications such as transperineal intra-prostatic needle placement (DiMaio et al., 2004, 2006), breast biopsy (Fischer & Kutter, 2004), and transrectal prostate biopsy (Krieger et al., 2005). Its use for robot-assisted needle steering can also be considered.

Ultrasound

The main drawback of ultrasound is a lower spatial resolution when compared to CT and MRI, and a tendency of US images to be noisy due to reflections, reverberations, shadows, air pockets, and biological speckle, which makes needle localization challenging. It also causes some degree of tissue deformation and dislocation as the transducer makes contact with the tissue scanned.

However, ultrasound imaging has a number of advantages—it is relatively inexpensive and compact, does not involve any ionizing radiation, and does not impose significant materials constraints on the needle and robot design. It also provides real-time information related to tissue properties, target displacement and tool position. Thanks to that, many studies have been conducted regarding needle localization methods in both 2D (Okazawa et al., 2006) and 3D (Ren et al., 2011) ultrasound images, automatic detection of anatomic structures for anesthesia (Tran & Rohling, 2010), and studies on needle-tissue interaction from ultrasound data (Dehghan et al., 2007). Also, it is one of the preferred imaging modalities for robot-assisted needle insertion, with systems developed for conventional insertions with rigid needles (Bassan et al., 2007; Hong et al., 2004) and for needle steering with base manipulation (Neubach & Shoham, 2010).

1.2.3 *Image-guided control of steerable needles*

The first robot-assisted systems for needle steering were dedicated devices addressed to operating in open-loop from a sequence of previously assigned insertion and rotation movements (Webster III et al., 2005). Their goal was mainly to evaluate the bending effects of the interaction between needle and tissue. Online update of rotation and insertion velocity references was first introduced by Romano et al. (2007). They proposed a teleoperation system where the control inputs were manually provided by an operator with a haptic device.

At the best of our knowledge, the first steering system to integrate planning and control for automatic closed-loop correction of the needle trajectory was that of Glozman & Shoham (2007), which uses the needle base manipulation strategy combined with fluoroscopic image feedback. In this work, they proposed an inverse kinematics approach to calculate the desired motion of the needle base that would make the tip follow a desired trajectory. A similar approach was later used by Neubach & Shoham (2010) with ultrasound guidance.

For the case of beveled needle steering, Kallem & Cowan (2007, 2009) proposed a low-level image-based asymptotic controller to stabilize the needle to a desired 2D plane. Their non-linear controller only actuates a subset of the needle Degree of Freedom (DOF)'s and has been designed to enforce the plane constraint of 2D insertions while working in parallel with other higher-level controllers responsible for needle navigation. To deal with the rotation lag between the base and the needle tip caused by torsional stiffness, Reed et al. (2009) have developed a torsion compensator that estimates and controls beveled needles using a mechanics-based model of the rotational dynamics of the needle interacting with the tissue during insertion.

Finally, Reed et al. (2008, 2011) presented a functional steering system that integrates patient-specific 2D pre- and intraoperative planning (Alterovitz & Goldberg, 2007) together with a planar controller (Kallem & Cowan, 2007) and torsion compensation (Reed et al., 2009). However, their system requires a dedicated two-DOF device, and the use of intraoperative planning is only suitable for static workspaces since it relies on a roadmap constructed preoperatively.

Some recent works proposed the use of trajectory tracking to follow precalculated paths; for instance, Wood et al. (2010a,b) used a dedicated two-DOF device combined with image feedback to perform trajectory tracking. Ko & Rodriguez y Baena (2012) have proposed another controller for trajectory tracking based on Model Predictive Control (MPC). In both cases, since the controllers do not perform trajectory update, changes in the workspace are not taken into account during the needle insertion.

An alternative to trajectory tracking controllers is the use of intraoperative path planners in order to perform online trajectory update from image feedback information. Hauser et al. (2009) were the first to use this kind of control-loop policy for needle steering. However, only obstacle free environments were considered.

1.3 CONCLUSIONS

This chapter presented different techniques to achieve needle steering in percutaneous procedures and discussed the advantages of using passive over active steering in terms of safety and design simplification. In the sequence, the most relevant methods for passive steering were introduced, each one with its own benefits and drawbacks, mainly related to the relationship between steerability and depth. Even though all of the presented methods may be combined to enable high steerability at multiple insertion depths, in this thesis we have chosen to focus on the problem of inserting beveled steerable needles due to their relative depth independence that results in more dexterous control at deeper insertions when compared to the base and tissue manipulation methods. For simplification purposes, from now on, every time a steerable needle is mentioned, unless explicitly stated otherwise, we refer to a beveled steerable needle.

Next, we discussed the use of image-guided robot-assisted systems and presented the state of the art in steerable needle technologies, together with the main challenges of this kind of procedure. Patient motion and physiological changes between preoperative planning and treatment phases are known causes of inaccuracy in percutaneous therapies. Also, the presence of uncertainties due to tissue deformation, tissue inhomogeneity, positioning errors and other modeling approximations often cause the needle to deviate from the original plan. Motivated by this, we propose a closed-loop strategy for needle steering that considers dynamic workspaces and disturbances in the expected needle motion.

The rest of this thesis presents the modules that compose the developed robotic system. More specifically, Chapters 3 and 4 describe the motion planning and control modules, respectively, while Chapter 5 presents the integrated system and main obtained results. Chapter 6 brings the thesis concluding remarks, and describes possible paths for future research in needle steering by highlighting open research challenges. The next chapter provides the theoretical background in kinematic modeling of steerable needles and presents the equations used to predict the behavior of beveled needles when inserted into soft tissue. One should read it carefully in order to become acquainted with the notation and mathematical representations used in the rest of the thesis.

2 NEEDLE MODELING

In Chapter 1, we examined the main needle steering approaches and how the interaction forces between the needle and the tissue affect the needle trajectory and the position of anatomical structures inside soft tissue. We also presented the concept of robot-assisted needle steering systems and their main computational modules. For robotically steered needles, an analytical model that predicts tissue deformation and needle deflection is desirable for optimization of the system design, path planning, and real-time control.

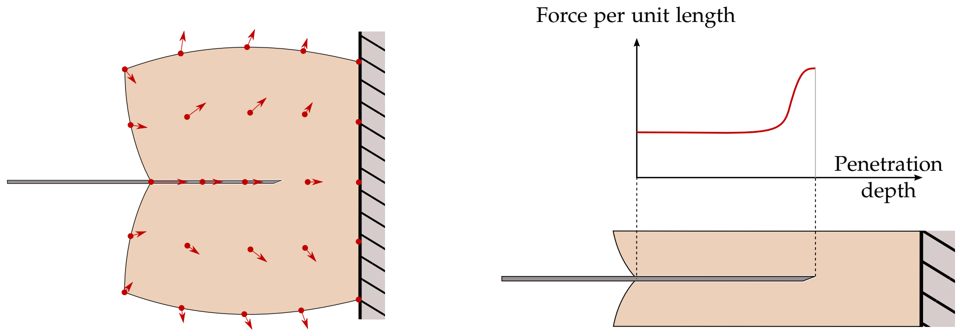
In order to model the effects of needle-tissue interaction, the relation between the locally applied force and motion of both tissue and needle have to be measured. However, there is no way to measure the local force directly since force sensors can only detect the force at their attached point. Also, during insertion, the needle punctures and passes through different tissue layers and membranes, varying interaction forces in the tip during each stage. Forces also change with needle depth and angle of insertion and for the same type of tissue, they even vary due to patient specific characteristics. The modeling of such effects is especially complex because of the inhomogeneous, nonlinear, anisotropic, elastic and viscous behavior of soft tissue (Abolhassani et al., 2007) and it has been a subject of much research.

A general survey on tool and tissue interaction models which describes both physics- and non-physics based interaction models is provided by Misra et al. (2008a). However, most of the presented studies are not specific to needle interaction and all of them focus on the description of tissue deformation, disregarding the interaction effects on the tool itself. In this chapter, we present some results from the literature that are more specific to needle insertion modeling. Also, we introduce the nonholonomic kinematic model adopted in this thesis and present some of the possible insertion techniques that have been developed for the steering of beveled needles in 2D and 3D.

2.1 NEEDLE INSERTION FORCES

It is important to understand the forces that act during the insertion of a needle into soft tissue because they can help to identify and model different tissue types and provide feedback for robot-assisted systems. However, when measuring needle insertion forces, only the resultant force acting at the proximal end of the needle is available, while in fact penetration forces are distributed along the entire length of the needle axis, resulting from physical phenomena such as cutting, elastic deformation and friction. The measured needle insertion force is actually the integration of such force distribution along the needle shaft.

DiMaio & Salcudean (2002, 2003a, 2005a) performed pioneering work in needle insertion modeling and simulation. They proposed a methodology for estimating the force distribution that occurs along the needle shaft by exploring the relationship between needle forces and 2D tissue deformation. It was based on a linear elastostatic material model, discretized using the FEM to derive contact force information that is not directly mea-



(a) A force is applied at the boundary of a deformable body and the resultant tissue deformation is observed.

(b) Needle force distribution for a 1mm/s insertion: constant along the needle shaft and higher at the tip because of tissue cutting.

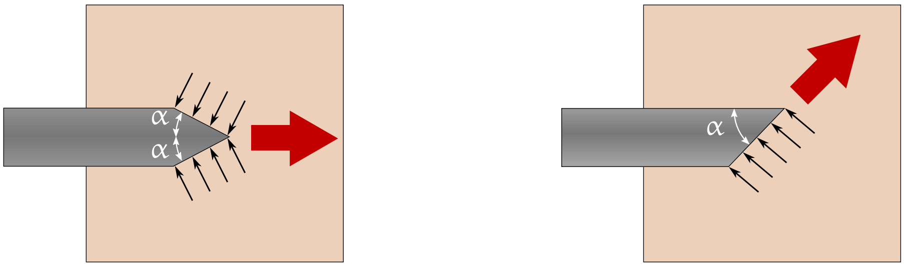
Figure 2.1: DiMaio and Salcudean's modeling of needle insertion is based in: *(a)* the observation of 2D tissue deformation, and *(b)* correspondent force distribution along the needle shaft. (Adapted from DiMaio & Salcudean, 2005a).

surable. The obtained force distribution indicated the existence of two forces: an axial friction force between the needle and the tissue, which is uniform along the shaft, and a force peak at the needle tip, which results from the cutting of the tissue as illustrated in Fig. 2.1b.

The work done by Simone & Okamura (2002) modeled such forces from experimental *ex vivo* studies conducted on bovine liver. They divided the needle insertion process in pre- and post- puncture of the organ membrane. During pre-puncture, we have the presence of stiffness forces due to the elastic properties of the organ, which were modeled as a nonlinear spring with elastic constant obtained by curve fitting the experimental data. As the insertion proceeds, the force value rises steadily until a sharp drop indicates the rupture of the liver membrane. During post-puncture, friction and cutting forces are added to the pre-puncture stiffness force. A modified Karnopp friction model which includes both the static and dynamic friction coefficients was used to model the friction during needle insertion. Finally, the cutting forces were obtained by subtracting the puncture and friction force from the total measured force.

Maurin et al. (2004) studied the forces involved during *in vivo* percutaneous procedures into liver and kidney of anesthetized pigs. Their experimental data was fitted to Simone and Okamura's model and to a second-order polynomial model taken from Maurel (1999), with low errors for both models. Their results also confirmed the increase in insertion forces during membrane punctures. Kataoka et al. (2002) investigated needle deflections with the insertion of a needle with triangular pyramid tip into a canine prostate while measuring the insertion force. By using a needle consisting of an outer and inner part, they were able to separately measure shaft forces and tip forces during needle insertion.

Instead of modeling the interaction forces, Alterovitz et al. (2003) used a 2D linear elastic model discretized with a finite element mesh to simulate the prostate tissue during a brachytherapy procedure and analyze the sensitivity of tip positioning errors to needle, tissue, and trajectory parameters. The effects of changes in the insertion depth, needle sharpness, friction, velocity, and tissue mechanical properties were evaluated but only tissue deformation was considered in their simulations, while needle deflection was ignored. This 2D model was extended to 3D by Nienhuys & van Der Stappen (2004) using



(a) Symmetric needle tip

(b) Asymmetric (beveled) needle tip.

Figure 2.2: Interaction forces at the needle tip when inserted into homogeneous soft tissue. The black arrows indicate the tissue compression reaction forces while the red arrow is the resultant direction of tissue cutting. The asymmetry of the bevel produces a resultant transverse load which causes the needle to naturally bend when inserted into soft tissue. (Adapted from Misra et al., 2010).

an element subdivision approach to ameliorate the effects of added computational complexity. Their work considered homogeneous models and rigid needles, neglecting again the effects of needle deflection.

2.2 NEEDLE DEFORMATION

During the insertion of a needle, the tissue around the tip deforms due to compression, and it imposes a reaction force contrary to this compression as shown in Fig. 2.2. In the case of a perfectly symmetric-tip needle being inserted into homogeneous tissue, such forces are equally distributed in all directions and the cut of tissue occurs in the insertion direction. However, when there is an imbalance in the forces distribution due to changes in the mechanical properties of the material into which the needle is inserted, or because of an asymmetric geometry of the tip, the tissue cut happens at an offset angle which depends on the tip geometry, needle flexibility and tissue properties.

In recent works, a lot of effort has been put in the modeling of needle deformation. Some studies propose methods for estimating the tip motion based on fundamental mechanical and geometrical properties of the needle and tissue, while other approaches use empirical observations of each needle and tissue combination in order to fit model parameters. Both techniques are discussed in more detail as follows.

2.2.1 *Mechanics-based models*

Several groups have developed mechanics-based models to represent needle deformation. Such models aim to relate the needle tip trajectory to the material and geometric properties of the tissue and needle. Kataoka et al. (2001) proposed a model for force-deflection of a beveled needle during insertion. The amount of needle deflection during insertion was described as a function of the needle length inside and outside the tissue, the needle diameter, the needle Young's module and moment of inertia, and the force per length. However, in this model, the mechanical properties of soft tissue, tissue deformation and the bevel angle of the needle are not considered.

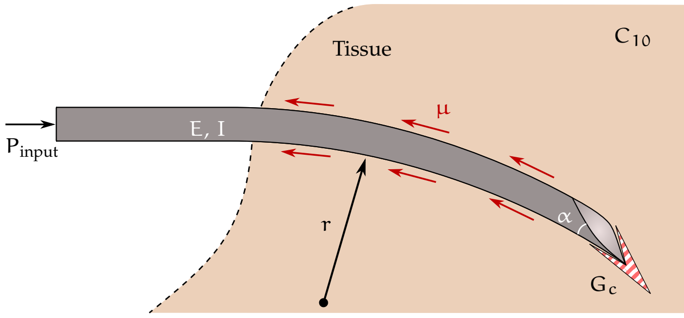


Figure 2.3: Schematic of a beveled needle interacting with soft elastic medium. The model incorporates the tip bevel angle (α), the needle's Young's modulus (E), and second moment of inertia (I); the tissue's nonlinear hyperelastic material property (C_{10}), rupture toughness (G_c), and coefficient of friction (μ); and the input displacement from the robot (P_{input}). (Adapted from Misra et al., 2009).

Okamura et al. (2004) analyzed the effects of needle diameter and tip type on needle deflection. Tests were performed on *ex vivo* bovine liver and silicone rubber phantoms and confirmed that the size and shape of the needle play an important role in determining both the forces of the needle insertion and the amount of needle deformation. Their work concluded that smaller needle diameters lead to less resistance force but more needle bending, and the use of beveled needles also results in more bending when compared to cone and triangular tips.

The effects of insertion velocity and tip bevel angle were analyzed by Webster III et al. (2005), who performed needle insertions into a relatively stiff phantom. Their results showed that decreasing the bevel angle increases the amount of needle deflection, but the bevel angle has little impact on the amount of axial force. They also found that the velocity of needle insertion in a stiff phantom had no significant effect on the amount of needle deflection, while it did change the amount of the axial force.

A study on the effect of needle shape and tissue material on tip forces and needle deflection has been presented by Misra et al. (2008b). They performed experimental studies on phantom and real tissues in order to obtain their elasticity and toughness parameters. These tissue properties were incorporated in a finite element simulation to show the relationship between needle bevel angle and the forces generated at the tip. The interaction of the needle tip deforming and rupturing tissue has been modeled with both contact and cohesive zone models. In general, it has been observed that tip forces were sensitive to the rupture toughness and that smaller bevel angles resulted in larger axial and transverse tip forces. Also, for most applications in which the needle would be steered through soft tissue, large variations in tissue elasticity are not expected.

In a more recent work, Misra et al. (2009) developed an energy-based formulation incorporating tissue and needle parameters (see Fig. 2.3) guided by microscopic and macroscopic observations during *in vitro* needle insertions. Their approach discretizes the needle length into segments and minimizes the total energy and work done in the system. As result, they obtained a 2D mechanics-based model that predicts the deflection and radius of curvature of a beveled needle inserted through a soft elastic medium. Simulations with the proposed model follow similar trends (deflection and radius of curvature) to those observed in experimental data. However, the extension of the model to three-dimensions

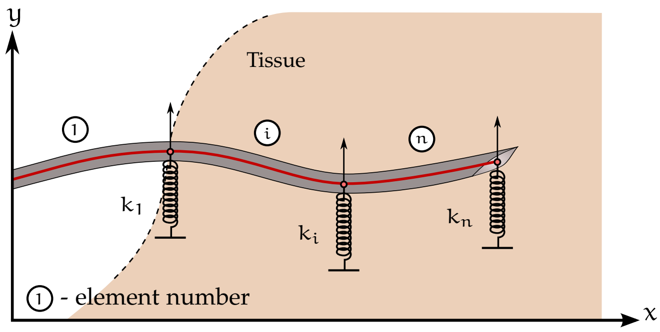


Figure 2.4: System model as a flexible beam subject to a number of virtual springs. (Adapted from Glozman & Shoham, 2007).

and the inclusion of an energy term to represent friction dissipation still has to be investigated in future studies.

2.2.2 *Phenomenological models*

Instead of using mechanical analysis, some models are based in phenomenological observation of needle and tissue interactions, that is, they try to fit data obtained with experiments to parametric models.

Considering the case of base manipulation—when the needle is stiff relative to the tissue—two models that relate the motions at the needle base to motions at the tip have already been proposed. In Glozman & Shoham (2007), the needle kinematic model is derived by modeling the soft tissue as springs with stiffness coefficients that vary along the length of the needle. In this approach, the needle is approximated by a linear beam subjected to point forces applied by the deformed tissue and modeled as virtual springs (see Fig. 2.4). Images are used to track the needle and detect its shape. The beam is split into many small elements and its displacements are used to calculate spring forces using finite-elements theory. From virtual spring forces applied to each element, they obtain the kinematic model of the needle. The resultant inverse kinematics is then used to determine the needle trajectory from motions applied at its base.

The second model from DiMaio & Salcudean (2003b, 2005b) involves numerically calculating the tissue Jacobian which is the matrix that relates the derivatives of needle base and needle tip configurations, as depicted in Fig. 2.5. Using this relation, the tip velocities can be determinated from the velocity of the needle base, and consequently, one may obtain the current needle tip position.

For the case of flexible beveled needles inserted in relatively hard tissues, Webster III et al. (2006a) have developed a kinematic model for needle steering. In contrast to the previously presented approaches (Glozman & Shoham, 2007; DiMaio & Salcudean, 2003b, 2005b), in this case there is no significant tissue displacement during needle insertion. Thus, they consider that only the needle deforms, and its deflection is caused by asymmetric forces applied on the beveled tip. As the needle is inserted into the tissue, the tissue imposes a reaction force on the bevel that deflects the needle tip.

Although speed rotation was included in the nonholonomic model proposed in (Webster III et al., 2006a), the experiments were performed with zero rotational speed. The work of Minhas et al. (2007) extended the previous kinematic model to include the ef-

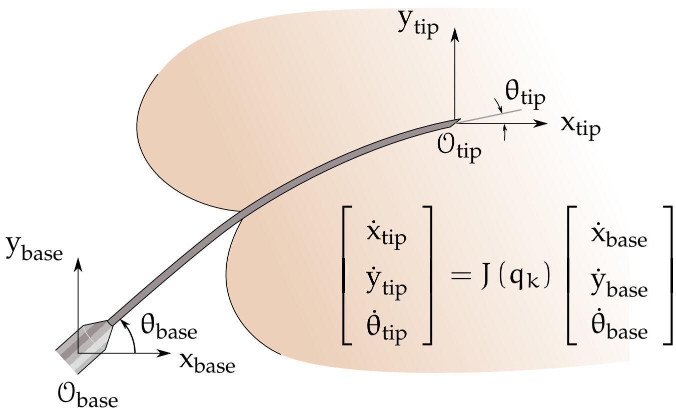


Figure 2.5: Motion of the needle tip with respect to the motion applied to the needle base. The tissue Jacobian matrix $\mathbf{J}_q$ depends on the current needle tip configuration given by $q = [\ x_{tip} \quad y_{tip} \quad \theta_{tip}\ ]^T$. (Adapted from DiMaio & Salcudean, 2005b).

fect of a duty-cycled rotation applied around the needle shaft, initially proposed by Engh et al. (2006a,b). The spinning is used during needle insertion in order to vary the needle path curvature—when the needle is inserted with constant spinning at a rate relatively larger than the insertion velocity, straight trajectories can be achieved. By alternating spinning and non-spinning periods, different curvature values can be achieved with the same needle-tissue combination.

It is still an open question whether mechanics-based, non-physics-based, or some combination of these is the best method for modeling needle-tissue interaction in real time. The development of models that can appropriately and accurately describe needle insertions whilst being computationally efficient remains as a research challenge.

Although mechanics-based models are capable of providing a fundamental understanding of needle-tissue interaction mechanics, they normally require prior knowledge of many geometrical and mechanical properties of the needle and tissue materials, measurement of applied torques and forces, and sometimes, even the numerical computation of finite element meshes which can be quite demanding.

On the other side, phenomenological models were primarily design to enable real time use. Even though accurate physics is not deemed a priority in such models, the results presented in the literature have demonstrated that they are able to capture needle-tissue behavior with sufficient accuracy to be used in planning and control. Thus, for modeling the flexible beveled needle behavior, we adopted the kinematic model developed by Webster III et al. (2006a) and its duty-cycled variation proposed by Minhas et al. (2007). For convenience, we adapted the mathematical formulation, which was originally presented in Lie algebra, to Clifford algebra. The resultant model is described and presented as follows.

2.3 BEVELED NEEDLE KINEMATIC MODEL

Webster's kinematic model for beveled needles was based on the standard unicycle model . Ignoring balancing concerns, an unicycle has two action variables: the rider can set the pedaling speed, and the wheel orientation with respect to the xy-plane. The wheel

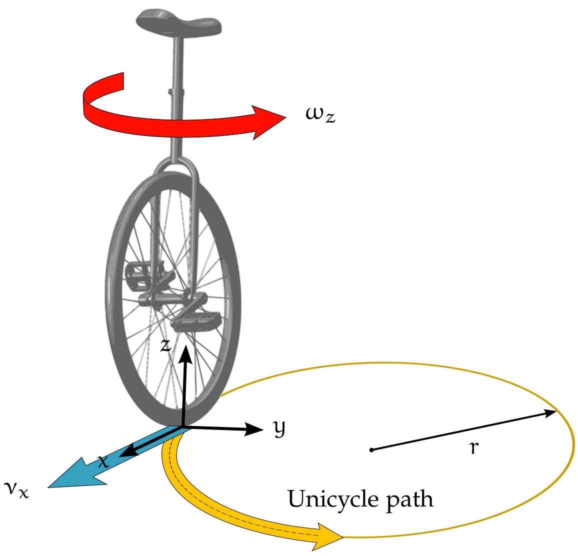


Figure 2.6: The standard unicycle kinematic model.

of a unicycle cannot slide sideways and, consequently, its mobility is restricted by the condition of pure rolling. Systems with this kind of movement restriction are known as nonholonomic, and they are characterized by constraint equations involving the time derivatives of the system configuration variables. Consequently, the standard nonholonomic model for a unicycle has a single no-slip constraint $\nu_y = 0$, and two inputs ν_x and ω_z, which are the unicycle speed and the angular velocity of the unicycle orientation, respectively. However, if we consider an unicycle with fixed front wheel, the angular velocity ω_z is no longer variable, and the standard model can be modified to include a ratio of linear velocity to angular velocity $\omega_z = \nu_x \kappa$. This causes the unicycle to follow a circular path with constant radius $r = 1/\kappa$, as shown in Fig. 2.6.

Steerable beveled needles are controlled by two degrees of freedom: insertion distance and rotation angle about the needle axis. Insertion pushes the needle deeper into the tissue, while rotation reorients the bevel direction. When pushed forward, the needle bends in the direction of its beveled tip, following an arc of approximately constant radius $r = 1/\kappa_{max}$, as shown in Fig. 2.7. If simultaneous rotation and insertion velocities are combined, the needle moves in the three-dimensional space.

This behavior resembles the unicycle with constant front wheel angle, with the difference that unicycle steering occurs in the xy-plane, while needle steering occurs in 3D space. The extension of the unicycle kinematic model from 2D to 3D results in the addition of new nonholonomic constraints: $\omega_y = \nu_y = \nu_z = 0$ and $\omega_z = \nu_x \kappa_{max}$. Thus, the system has two control inputs ν_x and ω_x, that are respectively the needle's insertion and rotation velocities along its shaft, and are referred simply as ν and ω.

It has been shown that if the needle is significantly more flexible than the tissue, its shaft follows the trajectory of the tip almost exactly (Webster III et al., 2005). Consequently, for an accurate representation of the entire needle shape, is it sufficient to describe the tip motion. The configuration $\underline{q}$ of the needle tip can be described in 3D by a rigid transformation from $\mathcal{O}_{world}$ to $\mathcal{O}_{tip}$, where $\mathcal{O}_{world}$ and $\mathcal{O}_{tip}$ are the reference and tip coordinate systems, respectively.

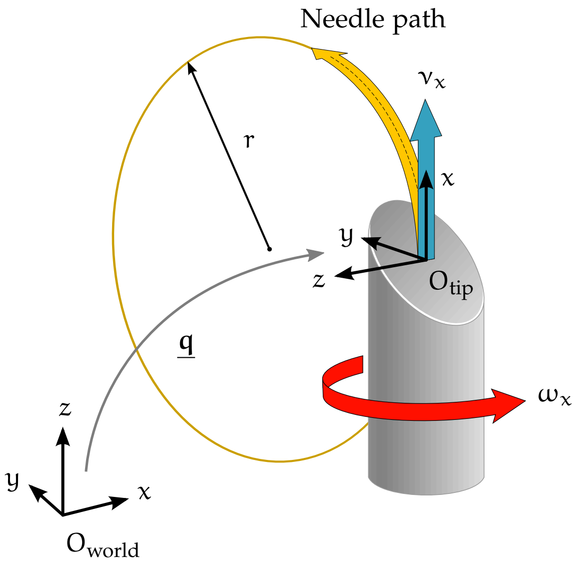


Figure 2.7: The beveled needle kinematic model with its system coordinates and inputs. (Adapted from Duindam et al., 2008b).

In our work, we adopted the dual quaternion notation, which is more compact. Similarly to homogeneous transformation matrices, when using dual quaternions, a single mathematical object describes the complete rigid motion, but instead of using a 4×4 matrix, dual quaternions simultaneously describe positions and orientations with an eight-parameter vector. Hence, in this thesis, a position is represented in the set of quaternions $\mathbb{H}$ by a quaternion $\mathbf{p} = x\hat{\imath} + y\hat{\jmath} + z\hat{k}$ whereas a configuration—that is, a set of position and orientation—is represented in the set of dual quaternions $\mathcal{H}$ by a dual quaternion $\underline{\mathbf{q}} = \mathbf{r} + \varepsilon\frac{1}{2}\mathbf{p}\mathbf{r}$, with $\mathbf{r}$ and $\mathbf{p}$ being respectively, the rotation and translation quaternions, and ε being Clifford's dual unit (see Appendix A for more details).

A sequence of rigid motions can be represented by a sequence of dual quaternion multiplications. As a consequence, we can obtain a discrete implementation of the needle kinematic model as

$$\underline{\mathbf{q}}_{k+1} = \underline{\mathbf{q}}_k \, \underline{\mathbf{q}}_\delta, \tag{2.1}$$

where $\underline{\mathbf{q}}_\delta$ represents the incremental movement during a period T_δ and is given by

$$\underline{\mathbf{q}}_\delta = \mathbf{r}_\delta + \varepsilon\frac{1}{2}\mathbf{p}_\delta\mathbf{r}_\delta, \tag{2.2}$$

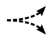
with

$$\begin{aligned} \mathbf{r}_\delta &= \cos\left(\frac{\phi_\delta}{2}\right) + \frac{\sin\left(\frac{\phi_\delta}{2}\right)}{\sqrt{\omega^2 + \nu^2\kappa_{max}^2}}(\omega\hat{\imath} + \nu\kappa_{max}\hat{k}), \\ \mathbf{p}_\delta &= \nu T_\delta \hat{\imath}, \quad \phi_\delta = \sqrt{\omega^2 + \nu^2\kappa_{max}^2}\,T_\delta, \end{aligned} \tag{2.3}$$

and being $\hat{\imath}$ and $\hat{k}$ quaternionic units that represent the x and z axis, respectively.

One should observe that we restricted ν to positive values, since the described tip behavior is only valid when the needle moves forwards, cutting tissue. During retraction, the needle motion is biased to follow the path already cut into tissue (Alterovitz et al., 2006,

Table 2.1: Possible needle trajectories and correspondent inputs.

	Motion	**Input** ν	ω
	Helix	$\nu > 0$	$\omega \neq 0$
	Arc	$\nu > 0$	$\omega = 0$
	Direction change	$\nu = 0$	$\omega \neq 0$
	Straight line	$\nu > 0$	$\|\omega\| \gg \nu$

2008; Webster III & Jones, 2010), and the presented model is no longer accurate. Also, retractions and re-insertions may result in unnecessary tissue damage (van den Berg et al., 2010), and hence, should be avoided in single target applications. The angular velocity ω, on the contrary, may be positive or negative depending on the direction we want the needle to follow.

2.4 Beveled needle insertion techniques

As previously discussed, a beveled needle moves in 3D according to the kinematic model given in (2.1), and the current tip configuration depends on the applied inputs ν and ω. Depending on the combination of translation and rotation velocities employed, different paths can be achieved, as shown in Table 2.1 and discussed as follows.

2.4.1 *Helical path*

When constant rotation and insertion velocities are applied simultaneously, the needle follows an helical path in the 3D space (Hauser et al., 2009; Duindam et al., 2008a). The pitch, axis and radius of the resultant helix depend not only on the natural radius of curvature r, but also on the insertion velocity ν and rotation velocity ω.

The natural curvature is related to the geometrical and mechanical properties of the needle and tissue. So, for a given needle-tissue set, r is constant. However, the control inputs ν and ω can be chosen in order to obtain a desired helical path. By combining different piecewise helical paths, a huge variety of trajectories can be obtained.

Fig. 2.8 illustrates how different helices are obtained for the same needle-tissue set, by applying different control inputs ω and ν. The presented lines were obtained by simulating a 1 m length needle insertion with the discrete model given in (2.1). From the simulated trajectories, we can see that faster rotations and slower insertions result in tighter helices.

In practice, for steering the needle, the insertion velocity ν is kept constant so that ω variations are used to change the direction and shape of the helix, and thus, control the needle trajectory, as illustrated in Fig. 2.9a. One drawback in this strategy is that it is not suitable for 2D steering since the combination of simultaneous ν and ω takes the needle to a three-dimensional path.

2.4.2 *Stop-and-turn*

Instead, in the stop-and-turn insertion technique (see Fig. 2.9c), rotation and insertion are performed one at a time—that is, $\omega = 0$ when the needle is being pushed forward, and $v = 0$ for changing the bevel orientation—the resultant trajectory is a concatenation of 3D arcs with constant radius (Duindam et al., 2008b). As a consequence, the needle can only reach points that belong to arcs of constant curvature. This not only restricts the needle possible trajectories, but also may result in a slow planning, normally not suitable for intraoperative use.

If ω is applied so that the bevel angle only performs complete $\pm 180°$ turns, the needle is kept in a constant plane and, as a consequence, we have a technique for 2D needle steering (Alterovitz et al., 2005b), which results in trajectories that are a concatenation of 2D arcs with fixed radius.

2.4.3 *Duty-cycling*

In the case of simultaneous insertion and rotation with ω being relatively larger than v, we have that the 3D helix tends to a straight line (see Fig. 2.8f). The duty-cycle technique (Engh et al., 2006a) explores such idea to achieve different curvature values. As illustrated in Fig. 2.9b, this technique combines small periods T_{ins} of pure insertion—whose resultant motion is an arc with the maximum natural curvature κ_{max}—with small periods T_{rot} of simultaneous insertion and rotation using $\omega \gg v$—which results in a straight trajectory with zero curvature. If the chosen discretization is small enough, the resultant path will be an arc whose curvature can range from the natural curvature κ_{max} to a pure straight trajectory, depending on applied duty-cycle signal.

The duty-cycle DC is defined as the ratio of the rotation period to the cycle period T_{DC}:

$$DC = \frac{T_{rot}}{T_{DC}} = \frac{T_{rot}}{T_{rot} + T_{ins}}, \tag{2.4}$$

where $T_{DC} = T_{rot} + T_{ins}$. There is a linear relationship between curvature and duty-cycle (Minhas et al., 2007), so any path curvature in the range $[0, \kappa_{max}]$ can be obtained by a proper choice of DC:

$$\kappa = \kappa_{max}(1 - DC), \tag{2.5}$$

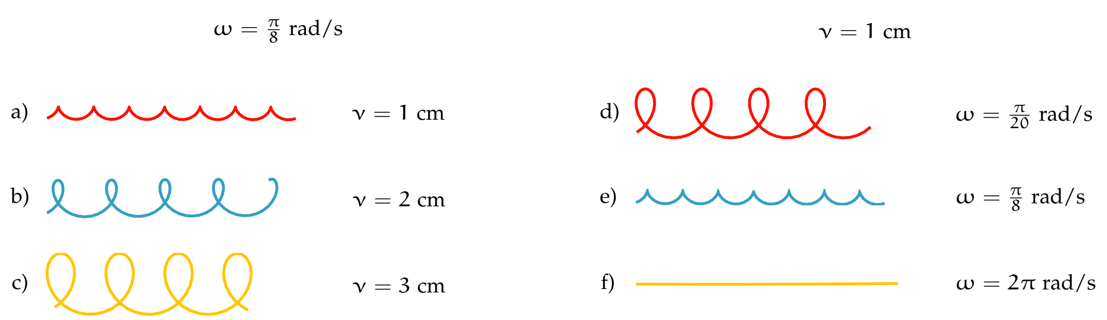
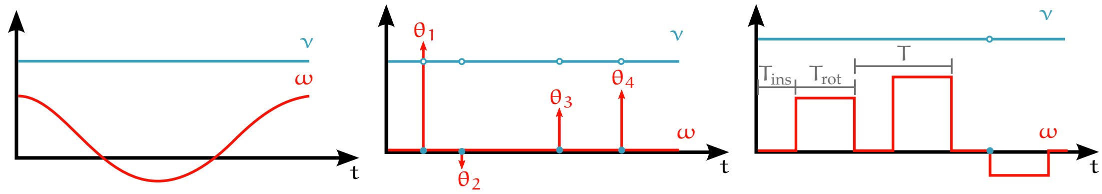


Figure 2.8: Helical paths for different choices of v and ω when inserting a beveled needle with $r = 5$ cm. In practice, for controlling the needle, v is kept constant while changes in ω control the direction and shape of the path.



(a) Helical path (b) Stop-and-turn (c) Duty-cycling

Figure 2.9: Needle insertion techniques. *(a)* Both v and ω are different from zero all the time and the resultant path is an helix. *(b)* Alternates constant v with instants where it is switched to zero in order to perform reorientation of the tip so that arcs with constant curvature are obtained. *(c)* Alternates periods with and without rotation while v is kept constant to obtain arcs with variable curvature.

where κ is the effective curvature and κ_{max} is the needle natural curvature, when no rotation is applied.

A discrete implementation of the duty-cycle needle kinematic model can be obtained by adapting (2.1) to include a combination of both movements:

$$\underline{\mathbf{q}}_{k+1} = \underline{\mathbf{q}}_k \, \underline{\mathbf{q}}_{rot} \, \underline{\mathbf{q}}_{ins}, \tag{2.6}$$

where $\underline{\mathbf{q}}_{rot}$ represents the movement during simultaneous rotation and insertion, and $\underline{\mathbf{q}}_{ins}$ is the movement of the insertion-only period.

Consequently, we have

$$\underline{\mathbf{q}}_{rot} = \mathbf{r}_{rot} + \varepsilon\frac{1}{2}\mathbf{p}_{rot}\mathbf{r}_{rot} \quad \text{and} \quad \underline{\mathbf{q}}_{ins} = \mathbf{r}_{ins} + \varepsilon\frac{1}{2}\mathbf{p}_{ins}\mathbf{r}_{ins}, \tag{2.7}$$

where

$$\begin{aligned} \mathbf{r}_{rot} &= \cos\left(\frac{\phi_{rot}}{2}\right) + \frac{\sin\left(\frac{\phi_{rot}}{2}\right)}{\sqrt{\omega^2 + \kappa^2 v^2}}(\omega\hat{\imath} + \kappa v \hat{k}) \\ \mathbf{p}_{rot} &= v T_{rot}\hat{\imath}, \quad \phi_{rot} = \sqrt{\omega^2 + \kappa^2 v^2}\, T_{rot}, \\ \\ \mathbf{r}_{ins} &= \cos\left(\frac{\phi_{ins}}{2}\right) + \sin\left(\frac{\phi_{ins}}{2}\right)\hat{k} \\ \mathbf{p}_{ins} &= v T_{ins}\hat{\imath}, \quad \phi_{ins} = v\kappa_{max} T_{ins}. \end{aligned} \tag{2.8}$$

During the simultaneous rotation and insertion period, if ω is applied so that the bevel angle only performs complete $\pm 180°$ turns, the needle is kept in a constant plane, providing a variation of the duty-cycle technique for 2D needle steering which provides trajectories that are a concatenation of 2D arcs with variable curvatures.

2.5 CONCLUSIONS

This chapter introduced the literature latest results in modeling the needle-tissue behavior during a puncture event. More specifically, a short review on the main approaches for modeling beveled needle interaction with soft tissue has been presented. In the sequence, we justified the choice of using a kinematic model instead of a mechanics-based model in the context of the proposed robot-assisted system. The nonholonomic kinematic model used in the rest of this thesis was then presented in detail, as well as the possible insertion techniques derived from it.

In the next chapter, we propose the planning module for our robot-assisted needle steering system. From all presented insertion techniques, we based the planning algorithm in the duty-cycle one because it is suitable for 2D needle steering, in accordance with our system requirements. When considering duty-cycling to insert the needle, we also remove the constant turning radius constraint present in the stop-and-turn technique, and replace it by a lower-bound constraint. Hence, the needle can now reach points that belong to arcs of any curvature ranging from 0 to κ_{max}. This increases the amount of path possibilities, and consequently, allows faster planning.

3 BEVELED NEEDLE PATH PLANNING

Path planning consists in defining a continuous sequence of configurations to achieve a desired task while respecting certain constraints. It has applications in many different areas such as robotics, manufacturing design, computer animation, computational biology, logistic, computer-aided design and others (Chang & Li, 1995; Pettré et al., 2003; Song & Amato, 2001; Siméon et al., 2001; Van Geem et al., 1999). Fig. 3.1 illustrates a planning example where the desired task is to pull two bars apart.

As previously discussed in Chapter 1, during a percutaneous procedures, it is essential to correctly place the needle device in order to have an effective treatment. This is critical in many medical applications such as drug delivery and biopsies, brachytherapies, and tumor ablations, during which needles must reach specific locations inside the body with great accuracy while avoiding anatomical structures that should not be damaged. For this reason, preoperative planning is normally the first step of a needle insertion procedure.

For steerable needles, this planning is often beyond the capabilities of human intuition due to the complex kinematics and the effects of tissue deformation, tissue inhomogeneities, and other causes of motion uncertainty which have been discussed in more detail in Chapter 2. To allow the full potential of needle steering, algorithms have been developed to automatic plan needle motion. Planning can be used purely preoperatively to generate a trajectory which will be followed during the hole procedure; or intraoperatively to update the planned trajectory online based on intraoperative images or sensor feedback. The later is the approach which is developed in this thesis.

In this chapter, we present the path planning algorithms used in our robot-assisted needle steering system. First, we introduce some general concepts regarding path planning and its development over the years. Following these introductory ideas, the chapter presents the state of the art in steerable needle path planning and one of this manuscript's main contributions, the Arc-RRT path planner. This algorithm provides high success rates and small computation periods thanks to the use of explicit geometry to obtain feasible trajectories for 2D or 3D beveled needle insertions. In the sequence, we introduce the use of an intraoperative replanning strategy that uses closed-loop feedback of the needle current configuration to compensate for system uncertainties.

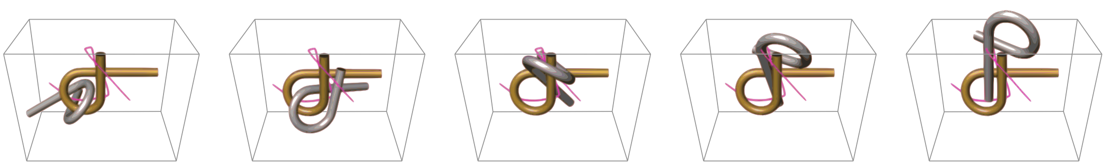


Figure 3.1: The "Alpha Puzzle" is a well-known difficult problem used as benchmark for motion planning algorithms. The puzzle goal is to pull the two bars apart (simulation sequence by James Kuffner and extracted from LaValle, 2006).

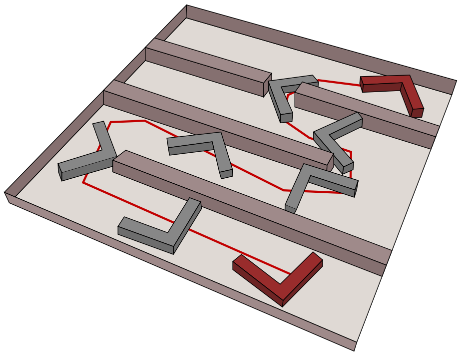


Figure 3.2: The basic motion planning problem. The maze represents the workspace with its walls being the obstacles. The L-shaped objects in red represent the start and goal configurations, while the gray ones are intermediate configurations obtained from the path planning solution.

3.1 PATH PLANNING BASIC CONCEPTS

The first works in path planning appeared in the late 60's during the early stages of the development of computer-controlled robots. However, active algorithmic development only started in the 80's with the notion of configuration space (Lozano-Perez, 1983). During these two decades, a very large number of techniques have been proposed. The book of Latombe (1991) provides an excellent overview of the progress on motion planning until the early 90's. Posterior theoretical advances and modern algorithms have been compiled by LaValle (2006).

The "basic path planning problem" involves computing a collision-free path between an initial and a final configuration in a static environment with known obstacles (see Fig. 3.2). In this case, the constraints on the solution path arise from the geometry of both the obstacles and the robot. Variations of the basic problem statement are numerous and so are its possible applications. Moving obstacles, uncertainties in sensing, and imprecise control add further levels of difficulty to the task.

The configuration of a robot is normally described by a number of variables. For example, in the case of a mobile robot, these variables are its position and orientation while for a manipulator arm, they are the positions of the different joints of the robot. For the case of a beveled needle, these variables are given by its tip configuration $\underline{\mathbf{q}}$, as defined in Chapter 2. The basic path planning problem is typically solved in the configuration space $\mathcal{C}$, in which each configuration of the robot is mapped as a point. The free configuration space $\mathcal{F}$ is the subset of $\mathcal{C}$ at which the robot does not intersect any obstacle. The robot can move from an initial to a goal configuration without intersecting an obstacle if the two configurations lie in the same connected component of $\mathcal{F}$. In this context, planning a collision-free path consists in answering connectivity and other topological questions in $\mathcal{F}$.

We can split path planning into two main classes: *holonomic* and *nonholonomic*. When all the robot's degrees of freedom can be changed independently, we talk about *holonomic path planning*. In this case, the existence of a collision-free path is characterized by the existence of a connected component in the free space $\mathcal{F}$. As a result, path planning consists in building the free configuration space $\mathcal{F}$, and in finding a path in its connected components.

Within the 80's, this problem has been addressed by creating a variety of heuristics and approximate methods, like decomposing $\mathcal{F}$ into simple cells and searching for connections between them. Other pioneer works showed how to describe the free configuration space as being a semi-algebraic set so that the connectivity of $\mathcal{F}$ could be described in a combinatorial way (Schwartz & Sharir, 1983). From there, the road towards methods based on Computational Real Algebraic Geometry was open. It provided various exact methods for specific robot systems, taking into account practical constraints like environment changes.

Exact planners are also called complete planners. Completeness is observed if in finite time, the algorithm either produces a solution or correctly reports that there is none. However, even when used to solve the basic path planning problem statement, complete planners have proved to be computationally intractable, that is, their computational costs increase exponentially with the number of degrees of freedom involved (LaValle, 2006). Moving obstacles, uncertainties in sensing, and imprecise control add further levels of difficulty. For many practical problems, the use of such planners is prohibitive. Consequently, the completeness requirement is often dropped, leading to the design of algorithms that satisfy weaker forms of completeness.

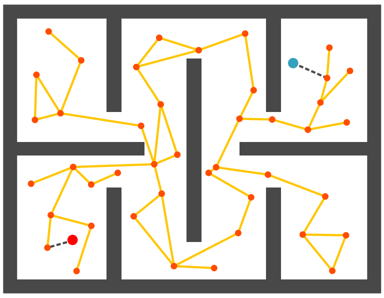
In the 90's, a new instance of the path planning problem has been considered: planning in the presence of kinematic constraints, that is, when the degrees of freedom of the robot are not independent. In such case, we talk about *nonholomic path planning*. For this class of problems, not only the admissible robot configurations are constrained by the obstacles and by the robot geometry, but also the directions of motion are subject to constraints so that any path in $\mathcal{F}$ does not necessarily correspond to a feasible one. This is basically why the purely geometric techniques developed for holonomic systems do not apply directly to nonholonomic ones. Consequently, nonholonomic planning turns out to be much more difficult than holonomic planning.

While complete algorithms are very time-consuming, it is possible to address complicated problems like nonholonomic planning by using alternative methods that relax the completeness constraint for the benefit of practical efficiency. In recent years, a number of path planning algorithms have been successfully used to solve challenging problems. Examples include the Randomized Path Planner (RPP) (Barraquand & Latombe, 1991), Probabilistic Roadmap (PRM) (Kavraki et al., 1996), and RRT (LaValle, 1998). Each of these methods can be seen as belonging to a field called sampling-based motion planning.

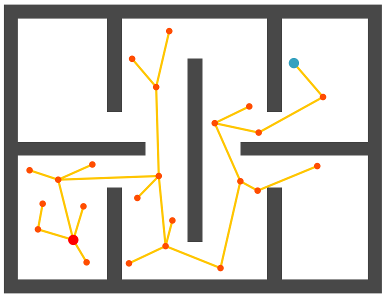
Sampling-based planners can successfully handle a large diversity of problems. Their success can be explained by the fact that no explicit representation of $\mathcal{F}$ is required. Instead of exhaustively exploring all possibilities of the configuration space, they search for collision-free paths in a smaller subset of randomly sampled configurations. Information on the configuration space is acquired by generating samples and edges to connect them, which are stored in a suitable data structure.

Following this paradigm, many different algorithmic techniques have been proposed, and some of them are now widely accepted as part of the standard literature in the field. These planners satisfy a weaker form of completeness called probabilistic completeness. A planner is said to be probabilistically complete if the probability of solving it converges to 1 as the number of samples goes to infinity. Hence, the problem solution is guaranteed provided that the algorithm is executed for a sufficient amount of time.

Sampling-based approaches can be grouped in two categories: those using sampling techniques for constructing a roadmap in $\mathcal{F}$, and those using sampling combined with



(a) Roadmap-based solution with PRM algorithm.



(b) Incremental solution with RRT algorithm.

Figure 3.3: Sampling-based planners: *(a)* for answering a query, it suffices to connect the start and end configurations to the pre-built roadmap; *(b)* for each query, the path is built from the start configuration and grown incrementally until it can connect the goal. The red and blue circles are the start and end configurations, respectively.

incremental search methods for finding a particular path (see Fig. 3.3). The choice mainly depends on the application. Roadmap methods like the PRM are more suitable for multiple-query applications, when several path planning queries involving the same robot in a static environment must be solved. In this type of algorithm, computing time is spent in a preprocessing phase when the roadmap is built, and after that, planning queries can be solved in real-time. On the contrary, incremental methods like the RRT do not have a preprocessing phase, and as a result, they are generally faster. These algorithms focus on solving a particular problem, and hence, are more appropriate for single-query applications. If another query is made, the previously processed information may be unsuitable, making it necessary to run the hole algorithm again.

RRTs are search trees grown from an initial state and expanded incrementally towards unexplored regions of the space, which is why they are called "rapidly-exploring". Like most heuristic search algorithms, the RRT sacrifices path optimality and completeness in exchange for practical efficiency in searching non-convex high dimensional spaces that may include both restrictions from the obstacles, and differential constraints (nonholonomic or kinodynamic). However, the paths computed have been observed to be relatively short, and rarely suffer from loops or self-intersections. In addition, the algorithm has been shown to be probabilistically complete so that the probability of failing to find a solution path, if one exists, decreases exponentially with planning time. Other advantages of RRT methods include simplicity of implementation and tuning, uniform coverage of any non-convex space, and good balance between explorability and local-minima avoidance.

3.2 PATH PLANNING FOR NEEDLE STEERING

As previously discussed in Chapter 2, a steerable needle can be described as a kinematic system with nonholonomic constraints. As a consequence, path planning is a complex task and its difficulty increases as we consider the presence of uncertainties due to errors in tip positioning, needle modeling, tissue inhomogeneity and deformation.

Many path planning methods for steerable needles have already been proposed in the literature. The first work was that of Park et al. (2005), who introduced a diffusion based approach that considers obstacle-free 3D environments. Later, Duindam *et al.* Duindam et al. (2010) used explicit geometric inverse kinematics for 2D and 3D needle motion planning. Xu et al. (2008) were the first to apply RRT-based methods to steerable needle planning and more recently, Lobaton et al. (2011) presented a sampling-based method for planning trajectories with multiple goals. However, none of these methods deal with motion uncertainty caused by modeling approximations, tissue deformation and other interaction forces that may cause the needle to greatly deviate from its planned path. To overcome this, three different approaches can be used.

The first idea, is to perform a trajectory tracking control as proposed by Wood et al. (2010b). Still, it does not take into account changes in the environment. Also, because of the limited needle steerability, this type of solution has to deal with problems related to the saturation of the control inputs, which may result in the needle not being able to correct the trajectory during an insertion length compatible with that of a percutaneous procedure. In such cases, the needle can miss the target completely while risking undesired collisions with obstacles.

Another possibility, is to model such uncertainties and consider their effects during the preoperative path planning. Alterovitz et al. (2005a) used a finite element mesh to compute soft tissue deformations combined to numerical optimization to find a locally optimal initial configuration and insertion distance. A finite element method has also been used by Vancamberg et al. (2010, 2011) to minimize the final error of a RRT solution in a breast biopsy application, whereas Patil et al. (2011) used FEM meshes combined with a sampling-based algorithm to plan in highly deformable environments. But the efficiency of these strategies depends a lot on the quality of the mesh simulation and how accurately it represents the real tissue.

Instead of simulating a tissue mesh, Alterovitz et al. (2008); Alterovitz & Goldberg (2007) considered uncertainty in needle motion by formulating the planning problem as a Markov Decision Process, using a discretization of the state space and a Stochastic Roadmap, respectively. Both these approaches generate a lookup table using dynamic programming that allows for instantaneous image-guided control for the steerable needle in a static environment. However, if we consider a dynamic environment that presents tissue and anatomical structures displacement due to patient motion or breathing, the use of precomputed paths may not be appropriate. van den Berg et al. (2010) used an LQG-based approach for planning and control of beveled needle insertions. Their technique minimizes the *a priori* probability of obstacles intersection and deals with noisy sensor measurements, but target and obstacles displacement due to patient motion are not considered.

The third alternative is to apply a single-query method that is fast enough to be used intraoperatively in order to replan the trajectory from imaging feedback information. Hauser et al. (2009) were the first to propose a control-loop policy for needle steering in deformable tissue, but again only obstacle-free environments were considered. Patil & Alterovitz (2010) proposed a fast RRT-based algorithm with obstacle avoidance but, although it presented potential for intraoperative use, its performance in closed-loop was not evaluated. In a recent work (Bernardes et al., 2011), we have also proposed a fast RRT-based planning method that was combined with an intraoperative replanning strategy for systematically correcting the needle trajectory.

In this chapter, we propose an application-specific path planning algorithm with input sampling strategy—the Arc-RRT path planner. It employs the duty-cycle technique to achieve arcs with adjustable curvature, and hence, obtain improved performance and efficiency. The proposed algorithm uses explicit geometry to obtain feasible trajectories that respect the needle nonholonomic constraints. As a result, we have obtained a planner with high success rate and fast calculation, compatible with intraoperative use for both 2D and 3D needle insertions.

We also developed an intraoperative replanning strategy that uses closed-loop feedback of the needle and workspace current configurations. Its objective is to make the needle insertion more robust not only to system uncertainties such as misplacement of the needle entry point, errors in parameter identification and modeling approximations, but also to changes in the obstacles and target positions caused by patient motion and physiological changes during the procedure.

3.3 THE ARC-RRT ALGORITHM

The objective of the Arc-RRT path planner is to find a sequence of duty-cycle signals parametrized in insertion length that is capable of taking the needle from its entry configuration to the target while respecting the nonholonomic constraints and avoiding the obstacles. This sequence must be obtained in a computational speed compatible with intraoperative use, so that the planned path can be adjusted to changes in the workspace and unexpected behavior of the needle.

Considering the approximation of the needle movement to that of a 3D unicycle model (Webster III et al., 2006a), the desired needle path is a combination of circular arcs. The final extremity of each arc should correspond to the next arc's initial extremity, not only in position but also in orientation, so we have C^1 continuity. If we consider percutaneous applications like some types of biopsy, tumor ablation, and anesthesia, the needle orientation at the target may not be a strong problem requirement. Thus, it may be relaxed and used as an extra degree of freedom to obtain such orientation continuity.

We define the insertion environment to be a bounded workspace and the locations of the targets and obstacles are known and defined by the surgeon. The needle is considered to be in the initial configuration q_{init} and must reach a final position p_{goal} with a sequence of concatenated arcs. Our method is based in the classic RRT approach (LaValle & Kuffner, 1999)—-a tree $\mathcal{T}$ is constructed with its root in q_{init} and continuously expanded until the goal point p_{goal} can be connected to it.

In a previous work (Bernardes et al., 2011), we proposed a first algorithm, the Arc-RRT with point sampling, which is depicted in Algorithm 1. It expands the tree by randomly sampling collision-free points and using them to grow branches and explore the workspace. In this strategy, for each point p_{rand} randomly sampled from the workspace, we define a set $\mathcal{Q}_{reachable}$ of all nodes in $\mathcal{T}$ from which p_{rand} can be reached. A node is considered reachable if the arc that connects it to p_{rand} does not intersect any obstacles and if its curvature respects the needle maximum curvature value κ_{max}. Then, the nearest node from $\mathcal{Q}_{reachable}$, namely q_{near}, is added to $\mathcal{T}$. The process is repeated until the target can be connected to the tree or until a maximum number of generated samples is reached.

Algorithm 1 Arc-RRT with point sampling

ARC_RRT (q_{init}, p_{goal})

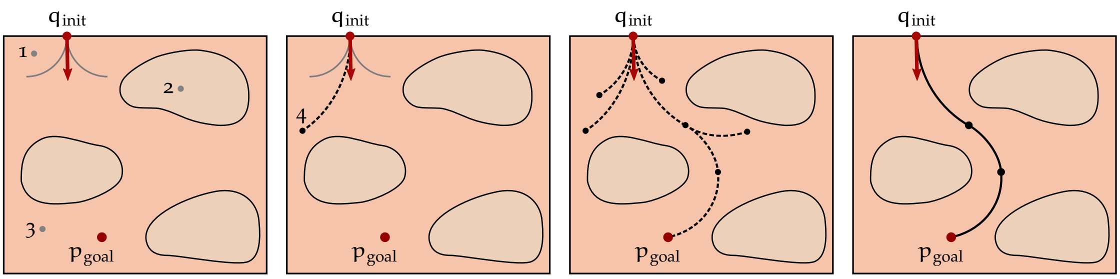
$\mathcal{T} \leftarrow$ INIT_TREE(q_{init})
while $\mathcal{T} \cap p_{goal} = \emptyset$ **do**
  $p_{rand} \leftarrow$ RANDOM_POINT()
  $q_{new} \leftarrow$ RRT_CONNECT_NEAREST($\mathcal{T}, p_{rand}$)
  $q_{goal} \leftarrow$ RRT_CONNECT($\mathcal{T}, q_{new}, p_{goal}$)
end while
$\mathcal{P} \leftarrow$ SEARCH_GRAPH($\mathcal{T}, q_{init}, q_{goal}$)
return $\mathcal{P}$

RRT_CONNECT_NEAREST ($\mathcal{T}, p$)

for all $q_i \in \mathcal{T}$ **do**
  $\mathcal{A}_i \leftarrow$ GET_ARC(q_i, p)
  if COLLISION_FREE($\mathcal{A}_i$) **and** $\mathcal{A}_i.\kappa < \kappa_{max}$ **then**
    add $\mathcal{A}_i$ to $\mathcal{Q}_{reachable}$
  end if
end for
if $\mathcal{Q}_{reachable} \neq \emptyset$ **then**
  $\mathcal{A}_{near} \leftarrow$ GET_NEAREST($p, \mathcal{Q}_{reachable}$)
  $\mathcal{T}$.add_vertex($\mathcal{A}_{near}.q_B$)
  $\mathcal{T}$.add_edge($\mathcal{A}.q_A, \mathcal{A}.q_B, \mathcal{A}.\kappa$)
  return $\mathcal{A}.q_B$
end if
return NULL

RRT_CONNECT ($\mathcal{T}, q, p$)

if $q \neq$ NULL **then**
  $\mathcal{A} \leftarrow$ GET_ARC(q, p)
  if COLLISION_FREE($\mathcal{A}$) **and** $\mathcal{A}.\kappa < \kappa_{max}$ **then**
    $\mathcal{T}$.add_vertex($\mathcal{A}.q_B$)
    $\mathcal{T}$.add_edge($\mathcal{A}.q_A, \mathcal{A}.q_B$)
    return $\mathcal{A}.q_B$
  end if
end if
return NULL



(a) (b) (c) (d)

Figure 3.4: Sequence for finding a solution path using the Arc-RRT algorithm with point sampling. The workspace is limited by the rectangle and has three irregular shaped obstacles. q_{init} and p_{goal} are given in red. The dashed black arcs are tree branches, while the solid gray arcs represent the κ_{max} boundary. *(a)* Samples 1, 2 and 3 are randomly selected one at a time and successively rejected for being unreachable. *(b)* Sample 4 is reachable, and hence, it is added to the tree. *(c)* The procedure is repeated until the goal position can be connected to the tree. *(d)* Then, a graph search is used to return the solution.

Algorithm 2 Arc-RRT with input sampling

```
ARC_RRT (q_init, p_goal)
𝒯 ← INIT_TREE(q_init)
while 𝒯 ∩ p_goal = ∅ do
    p_rand ← RANDOM_POINT()
    q_new ← RRT_EXTEND(𝒯, p_rand)
    q_goal ← RRT_CONNECT(𝒯, q_new, p_goal)
end while
𝒫 ← SEARCH_GRAPH(𝒯, q_init, q_goal)
return 𝒫

RRT_EXTEND (𝒯, p)
q_near ← NEAREST_NEIGHBOR(𝒯, p)
u_rand ← RANDOM_INPUT()
𝒜 ← APPLY_INPUT(q_near, u_rand)
if COLLISION_FREE(𝒜) then
    𝒯.add_vertex(𝒜.q_B)
    𝒯.add_edge(𝒜.q_A, 𝒜.q_B)
    return 𝒜.q_B
end if
return NULL

RRT_CONNECT (𝒯, q, p)
if q ≠ NULL then
    𝒜 ← GET_ARC(q, p)
    if COLLISION_FREE(𝒜) and 𝒜.κ < κ_max then
        𝒯.add_vertex(𝒜.q_B)
        𝒯.add_edge(𝒜.q_A, 𝒜.q_B)
        return 𝒜.q_B
    end if
end if
return NULL
```

Fig. 3.4 illustrates the process of building a tree using the Arc-RRT with point sampling. The extension of our previous method (initially developed for 2D steering) to a 3D environment incidentally resulted in an algorithm that resembles the one presented by Patil & Alterovitz (2010).

However, after further testing the point sampling algorithm, we have noticed that it was not fast enough, especially in difficult cases with reduced needle steerability (see Subsection 3.6.2.1 for statistical analysis). In more challenging situations, we have obtained low success rates and long processing times which are not desirable for intraoperative use. In order to improve Algorithm 1, we have proposed some modifications which are depicted in the pseudo-code from Algorithm 2. The initialization of the algorithm is the same as the previous method, with the construction of a tree $\mathcal{T}$ rooted in q_{init}. Then, a random point p_{rand} is also sampled from the workspace, but instead of trying to connect it to the tree, we select its nearest neighbor q_{near} to be extended by applying a sampled input signal u_{rand}. The procedure is repeated until the target p_{goal} can be connected to the tree. Fig. 3.5 illustrates the process of building a tree using the Arc-RRT algorithm with input sampling.

The Arc-RRT algorithm with input sampling—from now on referred to as Arc-RRT—can be used for both 2D and 3D planning. In the 2D planning, which is a special case of the 3D one, the needle workspace is restricted to a constant plane during the whole procedure. This is desirable if the available imaging modality provides only planar information, like an ultrasound equipment. The main differences between the two cases

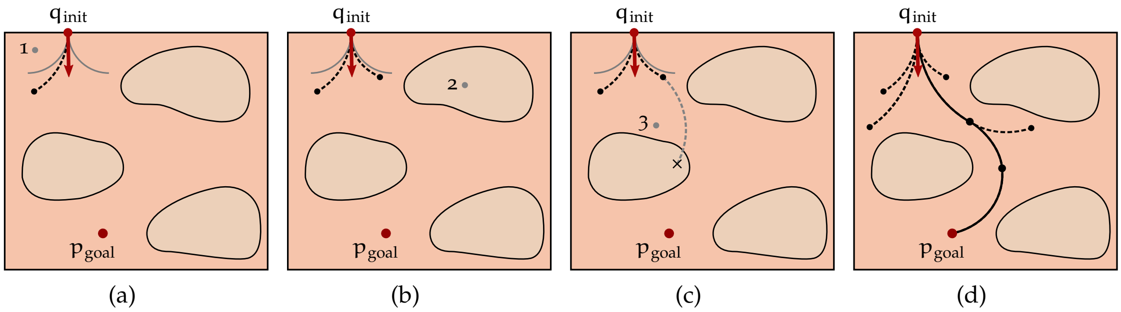


Figure 3.5: Sequence for finding a solution path using the Arc-RRT algorithm with input sampling. The workspace is limited by the rectangle and has three irregular shaped obstacles. q_{init} and p_{goal} are given in red. The dashed black arcs are tree branches, while the solid gray arcs represent the κ_{max} boundary. *(a)* Sample 1 is randomly selected and its closest tree node is expanded by applying a random input. *(b)* The same applies for sample 2. *(c)* The expansion of the closest node to sample 3 results in a collision and hence, this new branch is rejected. *(d)* The procedure is repeated until the goal position can be connected to the tree. The solution path is returned by a graph search.

are the representation of the configuration space, and the GET_ARC and APPLY_INPUT functions, as discussed in the next subsections.

3.3.1 *2D planning*

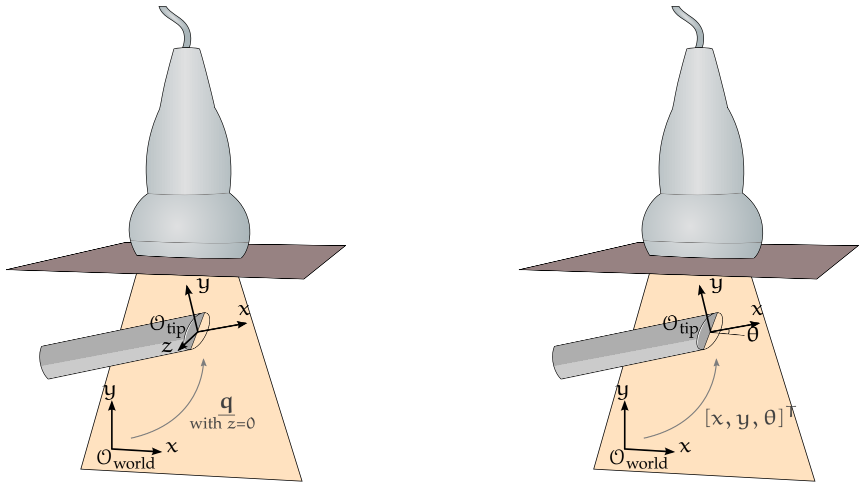
Although the kinematic model presented in Chapter 2 defines the needle tip configuration $\underline{\mathbf{q}}$ in a 3D workspace, to perform a 2D insertion we want the needle to move only in a two-dimensional plane, normally defined by the medical image. One possibility, is to keep the 3D dual quaternion representation $\underline{\mathbf{q}}$ by choosing the needle's xy-plane to be coincident with the desired plane, and by restricting the algorithm sampling to positions inside the xy-plane—that is, $\mathbf{p}_{rand} = x\hat{\imath} + y\hat{\jmath} + 0\hat{k}$, as depicted in Fig. 3.6. As a consequence, the 2D case corresponds exactly to the 3D case, but with all positions defined with z component equal to zero.

Alternatively, we may choose a different configuration space to represent the problem, simplifying the calculations even more. Thus, for planning purposes we let a 2D configuration to be represented in the three-dimensional manifold $\mathbb{R}^2 \times \mathbb{S}$ by a vector $q = [x, y, \theta]^T$ with $\theta = (-\pi, \pi]$, while a position is represented in the set $\mathbb{R}^2$by a vector $p = [x, y]^T$. Hence, we have the needle's initial configuration given as $q_{init} = [x_{init}, y_{init}, \theta_{init}]^T$ and the target position as $p_{goal} = [x_{goal}, y_{goal}]^T$.

In the Arc-RRT planner, we use explicit geometry to describe the needle motion as a concatenation of circular arcs that are used to grow the tree branches. This expansion of the tree is based on two local planning functions described as follows.

2D Function GET_ARC

The GET_ARC function gives the sequence of movements that will take the needle tip from a configuration $q_A = [x_A, y_A, \theta_A]^T$ to a position $p_B = [x_B, y_B]^T$ in the 2D workspace. For this, we insert the needle using the duty-cycle technique, which allows the needle to



(a) 2D configuration in the set of dual quaternions $\mathcal{H}$.

(b) 2D configuration in the three-dimensional manifold $\mathbb{R}^2 \times \mathbb{S}$.

Figure 3.6: Needle tip configuration in a 2D insertion. The needle xy-plane is aligned with the 2D desired plane. In the case of an insertion under ultrasound guidance, the desired plane corresponds to the image reference frame.

follow arcs with different curvatures in its xy-plane. It is possible to geometrically define the arc curvature κ, and the needle final orientation θ_B.

First, we calculate the signed bearing angle φ:

$$\varphi = \arctan\left(\frac{y_B - y_A}{x_B - x_A}\right) - \theta_A. \tag{3.1}$$

From Fig. 3.7, we can see that $\varphi + \phi = \frac{\pi}{2}$ and $\alpha + 2\phi = \pi$, hence $\alpha = 2\varphi$. Using the Law of Sines, we find that the arc curvature is given by

$$\kappa = \frac{1}{r} = \frac{2\sin(\varphi)}{\sqrt{(y_B - y_A)^2 + (x_B - x_A)^2}}. \tag{3.2}$$

The orientation at p_B is

$$\theta_B = 2\varphi + \theta_A. \tag{3.3}$$

Thus, using the relationship between curvature and duty-cycle signal given in 2.5, we obtain the sequence of rotations and insertions that should be applied to connect q_A and p_B.

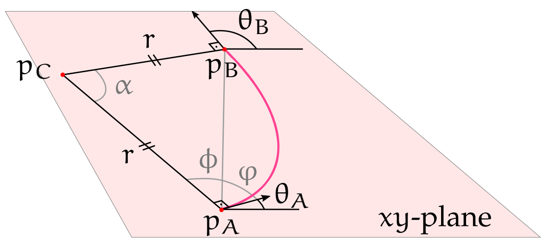


Figure 3.7: 2D version of the GET_ARC function connecting configuration q_A and position p_B.

2D Function APPLY_INPUT

Instead of connecting the configuration q_A to a given point, one might want to expand it by applying a given input signal $u_{rand} = (\alpha_{rand}, \kappa_{rand})$. In this case, once arc parameters α and κ are already known, they are used to determine a corresponding final configuration q_B.

The final orientation θ_B is given by (3.3).

The position of the arc center is given by

$$p_C = \frac{1}{\kappa}\begin{bmatrix} -\sin(\theta_A) \\ \cos(\theta_B) \end{bmatrix} + p_A, \tag{3.4}$$

and consequently, the final position is

$$p_B = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}(p_A - p_C) + p_C. \tag{3.5}$$

3.3.2 *3D planning*

For the three-dimensional planning, we define the insertion environment to be a bounded 3D workspace. The tip configuration is given by its dual quaternion representation $\underline{\mathbf{q}}$, while positions are represented by a quaternion $\mathbf{p} = x\hat{\imath} + y\hat{\jmath} + z\hat{k}$ with real part equal to zero. Thus, we have the needle's initial configuration given as $q_{init} = \underline{\mathbf{q}}_{init}$ and the target position as $p_{goal} = \mathbf{p}_{goal}$. The GET_ARC and APPLY_INPUT functions from the 2D case are adapted to the three-dimensional case as follows.

3D Function GET_ARC

The 3D version of the GET_ARC function gives the sequence of movements that connects a configuration $\underline{\mathbf{q}}_A$ to a position $\mathbf{p}_B$ in the 3D workspace. In the 3D case, we continue to use the duty-cycle technique, but before making an insertion, we must assure that $\mathbf{p}_B$ lies in the needle's xy-plane. When it is not the case, the needle must be previously rotated by an angle γ along its shaft until the *xy*-plane is aligned with $\mathbf{p}_B$. This rotation results in a new tip configuration $\underline{\mathbf{q}}_{A'}$, which can be connected to $\mathbf{p}_B$ by an arc which lies in the new xy'-plane, as shown in Fig. 3.8.

Thus, the GET_ARC function is defined by two steps presented as follows.

- Step 1: Align xy-plane *with* $\mathbf{p}_B$

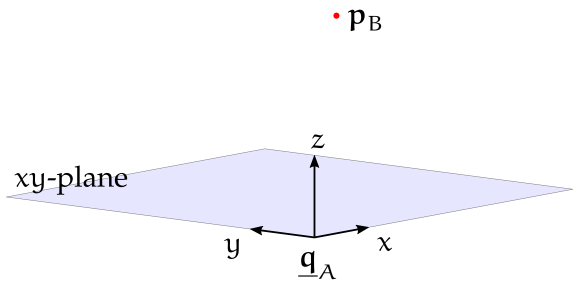
To obtain the intermediate configuration $\underline{\mathbf{q}}_{A'}$, we must calculate the angle γ . First, we write $\mathbf{p}_B$ in $\underline{\mathbf{q}}_A$ frame as

$$\mathbf{p}_B^A = \mathbf{r}_A^*(\mathbf{p}_B - \mathbf{p}_A)\mathbf{r}_A$$

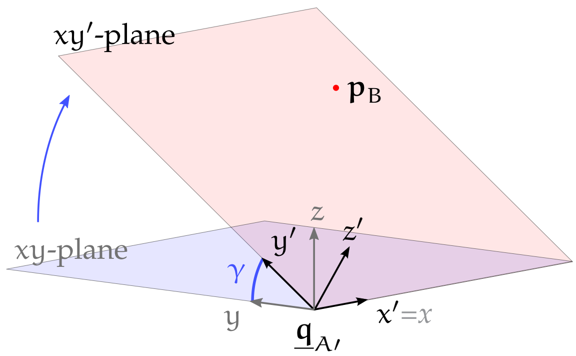
where $\mathbf{r}_A$ and $\mathbf{p}_A$ satisfy $\underline{\mathbf{q}}_A = \mathbf{r}_A + \varepsilon\frac{1}{2}\mathbf{p}_A\mathbf{r}_A$.

From the point coordinates $\mathbf{p}_B^A = (a\hat{\imath} + b\hat{\jmath} + c\hat{k})$, the angle γ between planes xy and xy' is given by

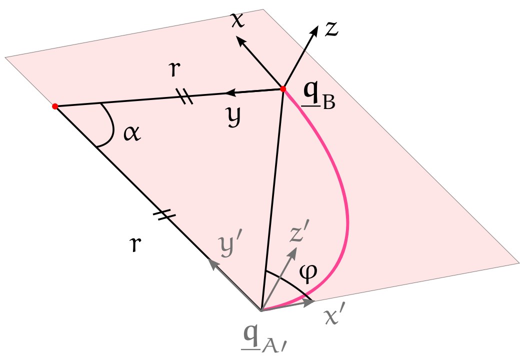
$$\gamma = \arctan\left(\frac{c}{b}\right). \tag{3.6}$$



(a) Current configuration $\underline{\mathbf{q}}_A$ and next position $\mathbf{p}_B$ to be connected.



(b) Step 1: align xy-plane with $\mathbf{p}_B$.



(c) Step 2: connect intermediate configuration $\underline{\mathbf{q}}_{A'}$ and $\mathbf{p}_B$ with a 3D arc.

Figure 3.8: 3D version of the GET_ARC function connecting configuration $\underline{\mathbf{q}}_A$ and position $\mathbf{p}_B$.

Note that angles are represented in the interval $(-\pi, \pi]$.

The intermediate configuration $\underline{\mathbf{q}}_{A'}$ is obtained by rotating $\underline{\mathbf{q}}_A$ of an angle γ around the x-axis

$$\underline{\mathbf{q}}_{A'} = \underline{\mathbf{q}}_A \underline{\mathbf{q}}_\gamma \tag{3.7}$$

with $\underline{\mathbf{q}}_\gamma = \cos(\frac{\gamma}{2}) + \sin(\frac{\gamma}{2})\hat{\imath}$.

Note that if the position $\mathbf{p}_B$ is already aligned with the xy-plane from $\underline{\mathbf{q}}_A$, then the rotation angle γ will be always zero and the resultant intermediate configuration $\underline{\mathbf{q}}_{A'} = \underline{\mathbf{q}}_A$. Thus, the GET_ARC function will have only Step 2.

- Step 2: Connect $\underline{\mathbf{q}}_{A'}$and $\mathbf{p}_B$

From a given start configuration $\underline{\mathbf{q}}_{A'}$ and a final position $\mathbf{p}_B$ it is possible to uniquely define an arc that connects them with a few trigonometric calculations.

The final point $\mathbf{p}_B$ written in the frame of $\underline{\mathbf{q}}_{A'}$ is given by $\mathbf{p}_B^{A'} = \mathbf{r}_{A'}^*(\mathbf{p}_B - \mathbf{p}_{A'})\mathbf{r}_{A'}$.

Being $\mathbf{p}_B^{A'} = (a'\hat{\imath} + b'\hat{\jmath})$, the signed bearing angle φ of the arc that connects $\underline{\mathbf{q}}_{A'}$ to $\mathbf{p}_B^{A'}$ is given by

$$\varphi = \arctan(\frac{b\prime}{a\prime}). \tag{3.8}$$

From Fig. 3.8c, we can see that $\alpha = 2\varphi$. Using the Law of Sines and the relationship between the arc and the bearing angle, we find that the arc curvature is given by

$$\kappa = \frac{1}{r} = \frac{2\sin(\varphi)}{\sqrt{a\prime^2 + b\prime^2}}. \tag{3.9}$$

The end configuration $\underline{\mathbf{q}}_B$ is given by a sequence of dual quaternion multiplications

$$\underline{\mathbf{q}}_B = \underline{\mathbf{q}}_{A'} \underline{\mathbf{q}}_\alpha, \tag{3.10}$$

where $\underline{\mathbf{q}}_\alpha = \mathbf{r}_\alpha + \varepsilon\frac{1}{2}\mathbf{p}_B^{A'}\mathbf{r}_\alpha$ with $\mathbf{r}_\alpha = \cos(\frac{\alpha}{2}) + \sin(\frac{\alpha}{2})\hat{k}$.

3D Function APPLY_INPUT

For the 3D case, the input signal is augmented to $u_{rand} = (\gamma_{rand}, \alpha_{rand}, \kappa_{rand})$. From the desired arc parameters γ,α and κ, we can easily obtain the corresponding final configuration $\underline{\mathbf{q}}_B$.

First, we apply γ to find $\underline{\mathbf{q}}_{A'}$ as given in (3.7).

Then, from the curvature κ, we can determine the position of the arc center as $\mathbf{p}_C = \mathbf{r}_{A\prime}\mathbf{p}_C^{A'}\mathbf{r}_{A'}^* + \mathbf{p}_{A'}$ where $\mathbf{p}_C^{A'} = \frac{1}{\kappa}\hat{\jmath}$ is the position of the center with respect to the frame of $\underline{\mathbf{q}}_{A'}$.

At last, the final point position is given by

$$\mathbf{p}_B = \mathbf{r}_{A\prime}\mathbf{p}_B^c\mathbf{r}_{A'}^* + \mathbf{p}_C \tag{3.11}$$

$\mathbf{p}_B^c = r\sin(\alpha)\hat{\imath} - r\cos(\alpha)\hat{\jmath}$ being its position with respect to the arc center. At last, the final configuration $\underline{\mathbf{q}}_B$ is

$$\underline{\mathbf{q}}_B = \mathbf{r}_B + \varepsilon\frac{1}{2}\mathbf{r}_B\mathbf{q}_B \tag{3.12}$$

with $\mathbf{r}_B = \mathbf{r}_{A\prime}\mathbf{r}_\alpha$ and $\mathbf{r}_\alpha = \cos(\frac{\alpha}{2}) + \sin(\frac{\alpha}{2})\hat{k}$.

Algorithm 3 Intraoperative replanning algorithm

REPLANNING($q_{tip\_now}, p_{goal\_now}, \mathcal{P}$)

$\mathcal{P}_{now} \leftarrow$ UPDATE_PATH_EXTREMITIES($q_{tip\_now}, p_{goal\_now}, \mathcal{P}$)
for all $\mathcal{A}_i \in \mathcal{P}_{now}$ **do**
    $\mathcal{A}_i \leftarrow$ GET_ARC($\mathcal{A}_i.q_A, \mathcal{A}_i.p_B$)
    if !COLLISION_FREE($\mathcal{A}_i$) **or** $\mathcal{A}_i.\kappa > \kappa_{max}$ **then**
        return ARC_RRT($q_{tip\_now}, p_{goal\_now}$)
    end if
end for
return $\mathcal{P}_{now}$

3.4 INTRAOPERATIVE REPLANNING

Generally speaking, the output of an open-loop controller is determined based on a priori knowledge about the motion task and the environment, which may have been previously acquired by sensors. Closed-loop control is instead computed online and based on continuous feedback of sensor data.

Essentially, we regard planning and open-loop control as synonyms, as opposed to closed-loop control. However, the borderline between open-loop and closed-loop solutions may not be so sharp. In fact, we may use repeated open-loop phases, replanned at higher rates using new sensor data to gather current information on the task. In the limit, continuous sensing and replanning leads to a feedback solution.

As firstly mentioned in Chapter 1, in this thesis we propose the use of imaging feedback to perform intraoperative insertion replanning, and thus, compensate for disturbances and uncertainties in closed-loop. The previous section described how the Arc-RRT algorithm provides a sequence of arcs that take the needle from its entry point to the goal while avoiding the obstacles. However, tissue deformation and inhomogeneity, imprecision of the unicycle model, torsional stiffness and uncertainties in the initial configuration may deviate the needle from the planned trajectory, leading to a possible collision with an important organ or misplacement of the needle tip at the end of the insertion task.

In this section, we propose a replanning algorithm which should be executed at each insertion cycle to adjust the planned path. It uses information about the current needle tip and workspace configurations to systematically correct the planned trajectory until the needle tip is sufficiently close to the target. We assume that current information about the workspace is provided by an imaging system and the current needle tip configuration is also known.

This feedback information is used in closed-loop to update the path by considering the needle's current configuration q_{tip_now} as the new starting position for the path and the target's current position p_{goal_now} as the new goal. Then, the *GET_ARC* function is run to adjust all arcs from $\mathcal{P}$, recalculating the new curvatures and final orientations. If a collision is detected in the updated arcs, or if the new curvature does not respect the maximum limit, the complete Arc-RRT planner is run again to find a new feasible trajectory. This replanning algorithm is presented in more detail in Algorithm 3.

3.5 ENTRY POINT PLANNER

In needle steering, the performance of the planner is highly dependent on the needle steerability and how cluttered the environment is. Even for a constant workspace configuration, as we raise the minimum radius of curvature, the harder it gets to find feasible paths, until the limit where there is no possible solution for a given combination of initial configuration and target position.

A possible strategy to overcome such limitation is to relax the initial configuration requirement. In this case, instead of having a defined q_{init}, we define an angle interval and a region of interest from where the insertion can be performed (one should keep in mind possible restrictions of a specific clinical application when defining such limits). A preoperative planning procedure is then performed to find a feasible entry configuration within the specified possibilities.

Here, we propose a preoperative variation of the Arc-RRT algorithm—the Entry Point Planner. It searches for a feasible initial configuration by discretizing the entry angle interval and the region of interest to build a set of candidates for q_{init}. For each candidate, the Arc-RRT algorithm is run and if it successfully returns a path $\mathcal{P}$, a cost function $\mathcal{C}(\mathcal{P})$ is calculated. The candidate with lowest cost path is then selected as the initial configuration to be used in the procedure.

For evaluating each feasible candidate, we have chosen a cost function that takes into consideration two factors conflicting with each other (Mittal & Deb, 2007): the total path length and the collision risk. The objective of having a short insertion clashes with the desire of going as far as possible from obstacles. The closer from obstacles the needle passes, the higher are the chances of collision. Hence, we define the two functions

$$f_{risk} = \sum_{i=1}^{N} \left(\frac{d_{safe}}{d_{i,min}} \right)^2 \text{and} \quad f_{length} = \sum_{i=1}^{K} \left| \frac{2\varphi_i}{\kappa_i} \right|, \tag{3.13}$$

where N is the number of points used to discretize the trajectory, $d_{i,min}$ is the minimum distance from point i to the workspace obstacles, d_{safe} is the minimum safety distance the needle should respect from obstacles, K is the number of arcs in the trajectory, φ_i is the bearing angle from the i-th arc, and κ_i is its correspondent curvature.

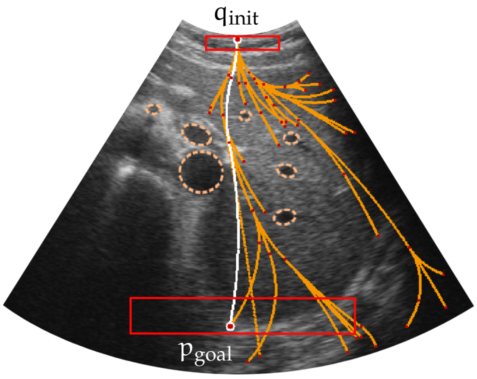
The final cost function is a weighted sum of both functions, with β being a weighting factor, as given by (3.14):

$$\mathcal{C} = \beta f_{risk} + (1 - \beta) f_{length} \tag{3.14}$$

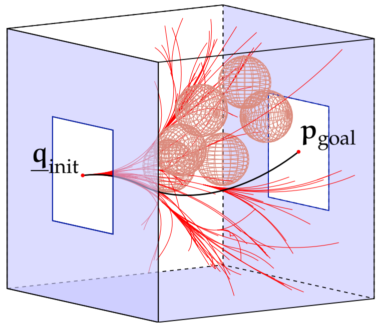
To make the algorithm faster, we can make a coarse initial discretization of the entry region and after we have found the lowest cost path, we perform a finer discretization around the chosen configuration to check for a better solution. This is especially useful for the cases where the given region of interest is very large.

3.6 RESULTS AND DISCUSSION

Some computational simulations have been performed to evaluate the proposed algorithms in two different scenarios, both based on possible clinical situations and depicted in Fig. 3.9. For the planar case, considering the context of this thesis in the ANR USComp



(a) 2D scenario for the simulations based on an abdominal ultrasound image. The dashed ellipses are obstacles that correspond to anatomical structures distinguishable in the image, the rectangles are the regions of interest, the orange lines compose the tree constructed by an execution of the Arc-RRT and the while line is the path found.



(b) 3D scenario for the simulations based on Xu et al. (2008). The spheres represent obstacles, the rectangles are the regions of interest, the red lines compose the tree constructed by an execution of the Arc-RRT and the black line is the path found.

Figure 3.9: Simulation scenarios for the Arc-RRT evaluations.

project, we extracted the workspace from an ultrasound image with some distinguishable anatomical structures considered as obstacles (Fig. 3.9a). For the three-dimensional case, the tests were conducted in a 3D simulated environment with obstacles regarding a typical scene for prostate needle insertion (Fig. 3.9b), where the spherical obstacles represent sensitive or impenetrable areas around the prostate, such as the urethra, the penile bulb, and the pubic arch. All simulations were run in a PC with Intel Core 2 Duo 3.00 GHz, 3.2 GB memory and Ubuntu 10.04 operating system.

This section compares the performance of Algorithms 1 and 2 with respect to computation time and success rate. It also analyzes the influence of the maximum curvature parameter in path planning efficiency, confirming the efficiency of the Entry Point Planner in difficult scenarios. Moreover, we investigate the possibility of using extra CPU time to improve the quality of the paths by exploring the scenario with simultaneous trees. To conclude, we simulate some needle insertion procedures under disturbances to test the replanning strategy and evaluate its capacity to compensate for system uncertainties intraoperatively.

3.6.1 *2D planning evaluation*

We tested the Arc-RRT planner for needle steering in the 2D environment with obstacles depicted in Fig. 3.9a. The workspace is based on an abdominal ultrasound image obtained with a General Electric Diasonics Synergy equipment and is defined to be the area covered by the ultrasound transducer and the obstacles are seven round structures distinguishable in the image. For the kinematic model of the needle, we specified 6.01 cm for the needle radius of curvature. To validate the 2D planner, three different sets of simulations are presented.

Table 3.1: Compared expansion velocity for 2D version of Algorithms 1 and 2.

	Algorithm 1 point sampling	Algorithm 2 input sampling
Number of trials	1000	1000
Number of tree nodes	100	100
Number of samples	435.33 ± 94.22	99.05 ± 0.22
CPU time (ms)	126.61 ± 139.05	0.30 ± 0.05

Table 3.2: Compared performance for 2D version of Algorithms 1 and 2.

	Algorithm 1 point sampling	Algorithm 2 input sampling
Number of trials	1000	1000
Number of successes	891	997
Number of tree nodes	5±6	15 ± 27
CPU time (ms)	3.20 ± 3.49	0.51 ± 0.67
Path length (cm)	13.51 ± 2.97	11.94 ± 0.79

3.6.1.1 *Comparison of algorithms with point and input sampling*

The first set of simulations compares the performance of the Arc-RRT with point and input sampling with respect to computation time, success rate and insertion length. We assumed a needle with radius of curvature of 6 cm.

SIMULATION 1 First, we compared the computational time needed by each algorithm to build a tree with the same amount of nodes. The initial configuration was set to the position $q_{init} = [330, 105, \frac{\pi}{2}]$, and for each algorithm, a tree was created and expanded until it reached 100 nodes. Table 3.1 presents, for both algorithms, the average computational time and the average number of samples generated in 1000 trials. We can see that the input sampling strategy expands the tree much faster than the point sampling one. The main cause of such difference is because for each iteration, the point sampling algorithm has to check the reachability of all nodes in the tree, while the input sampling method performs the check for only one node.

SIMULATION 2 Secondly, we compared the ability of each algorithm to find a solution for the same planning situation. In this test, the maximum number of samples for constructing the RRT was set to 200 and, for each trial, an initial configuration and a goal point were randomly sampled with a uniform distribution from the regions of interest. Each set of q_{init} and p_{goal} was tested in both algorithms. Table 3.2 presents the results for 1000 trials, indicating a better performance of Algorithm 2 (input sampling), which got higher rate of success and lower processing time when compared to Algorithm 1 (point sampling). From the data, we observe that the input sampling strategy is capable of finding a solution with fewer samples when compared to point sampling.

In resume, by changing from point to input sampling, we obtained a new path planner with higher success rate, and more than 6 times faster than the previous one, without increase in the average insertion length. This indicates that the use of the new algorithm is highly advantageous to an intraoperative planner.

Even though the Arc-RRT convergence is not assured in finite time, RRT-based algorithms are proved to be probabilistic complete, meaning that if we give it enough time to search for the solution and if the solution exists, it will be found (Kuffner & LaValle, 2000). In practice, what we observe is that the Arc-RRT converges much before the next insertion cycle due to its high success rate and fast execution when compared to the insertion procedure speed.

Also, the solution path is not the shortest possible since we do not use any expensive numerical optimization in favor of intraoperative use. Nevertheless, the general quality of the solutions converges towards the optimum as we increase the number of executions of the Arc-RRT toward infinity. Consequently, if we profit from the time left between cycles to accumulate executions, the quality of the path will get closer to optimal. This is discussed in more details in the next set of simulations.

3.6.1.2 *Use of simultaneous trees with the Arc-RRT*

The second set of simulations evaluates the quality of the paths generated. For this, we defined the average path length as our quality criterion. The initial configuration and the target point are the same for all tests in Simulation 2 and were arbitrarily chosen in the image as $q_{init} = [323, 92, \frac{\pi}{2}]$ and $p_{goal} = [315, 412]$ given in pixels and radians. The maximum number of nodes allowed for the RRT is 2500.

After 10000 trials, we obtained 100% of success. However, the average path length was 159.1 mm (32% higher than the smallest value), with a standard deviation of 28.10 mm. The smallest path from all trials was 120.15 mm, while the longest one was 410.19 mm (see results for 1 RRT in Fig. 3.10).

To overcome such high variance in the path length, we proposed using extra CPU time to assure a better quality path. The idea then is to build more trees to explore the workspace independently, and to pick the solution with the shortest path. The extra time used to compute the path is not harmful to the overall performance of our system since the dynamic of the needle insertion is very slow when compared to the RRT exploration which is in the range of a few milliseconds.

Fig. 3.10 summarizes the results for tests with 1, 10, 20 and 50 simultaneous trees. For all cases we had the same 100% success rate, but very distinct results for the average path length. The results show that despite the increase in CPU time, the addition of more RRT's to explore the workspace might be justified by the improvement in the overall path and smaller variance of results. However, one should keep in mind that the use of simultaneous RRT's involves a trade-off between computational time and path quality that should be compatible with the application requirements.

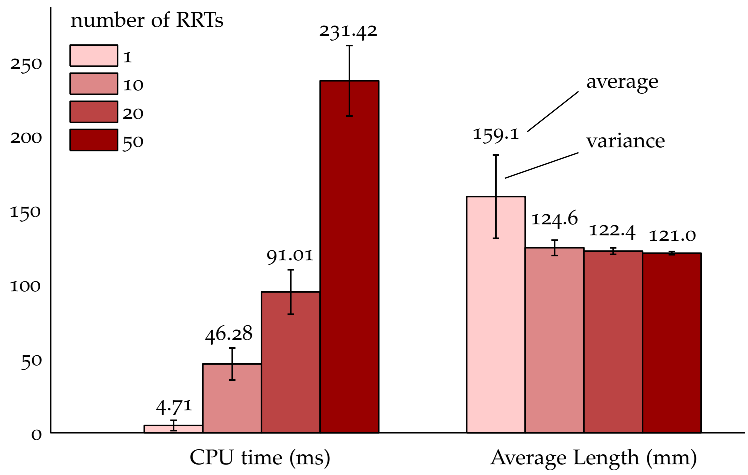


Figure 3.10: Comparison of average path length and CPU time for different number of simultaneous trees in Simulation 2.

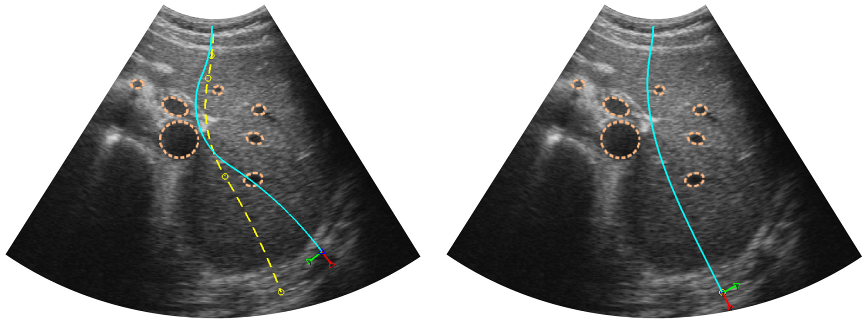
3.6.1.3 *Replanning evaluation*

The third set of simulations evaluates the effects of uncertainties in the needle trajectory and if the replanning strategy is able to compensate them. For this, the needle tip was simulated using the discrete model from (2.6). When we consider the presence of system uncertainties and perturbations such as positioning errors, tissue deformation and inhomogeneity, the result can greatly deviate from the expected one. In simulation, these effects were modeled as white noises added to the initial and measured tip configurations (zero mean, and standard deviation of 1 mm to the tip position and 0.01 rad to the orientation), and an induced perturbation in the actual natural curvature, which was set to 30% bigger than the expected value. The needle tip was simulated using the discrete model from (2.1), with natural radius of curvature of 6 cm. The rotational velocity was set to 2 Hz, and hence, $T_{rot} = 0.5$ s. The insertion distance at each cycle was defined to be $\triangle s = 1$ mm. The workspace is the same used in previous simulations.

We used the Arc-RRT with input sampling to obtain a feasible path and the correspondent sequence of duty-cycles parametrized in path length was directly calculated from the arc's curvatures and (2.5). The sequence of inputs was then applied to the simulated needle and it resulted in the trajectory depicted in Fig. 3.11a. It can be observed that the needle failed to avoid the obstacles and finished the insertion with 33.41 mm of error.

Then, instead of applying the control inputs in open-loop, the replanning strategy was used to systematically correct the trajectory along the insertion procedure. Fig. 3.11b illustrates how the online update of the path was able to compensate for the uncertainties, with a final error of only 0.20 mm.

The path update calculation is extremely fast since it uses geometric relations to satisfy the system nonholonomic constraints. We performed timing experiments on a 3GHz PC and the average execution time for executing the function GET_ARC was 0.038 ms in 10000 trials, with standard deviation of 0.033 ms. Combined with the good performance of the Arc-RRT algorithm and considering that needle insertion procedures normally occur at



(a) Final error = 33.41 mm (b) Final error = 0.20 mm

Figure 3.11: Simulated trajectories of the needle tip in simulations *(a)* under disturbances, and *(b)* with adaptive replanning for noise compensation. Planned path (dashed) and simulated needle trajectory (solid).

small insertion and rotation velocities due to safety restrictions, the replanning strategy can be easily used intraoperatively.

3.6.2 *3D planning evaluation*

The scenario for 3D planning is depicted in Fig. (3.9b) and consists of a cubical region with coordinates in centimeters $(-5, 5) \times (-5, 5) \times (-5, 5)$ and six unit-radius spheres, centered at $[-1, 0, 0]$, $[3.5, 0, 1.5]$, $[2.5, 0, 2.9]$, $[5, 0, 2]$, $[0.5, 1.4, 0.3]$ and $[0.5, -1.4, 0.3]$ to represent obstacles around the prostate such as the pubic arch, the urethra and the penile bulb Xu et al. (2008). The 4x4 cm blue squares are centered around the start and goal positions and represent regions of interest for possible entry configurations and targets. To evaluate the proposed planner in a 3D case, four different simulations are presented.

3.6.2.1 *Comparison of algorithms with point and input sampling*

Similarly to previous 2D simulations, we compared the performance of the Arc-RRT with point and input sampling in a 3D situation. We specified a needle with 4 cm of minimum radius of curvature.

SIMULATION 1 First, we compared the computational time needed by each algorithm to build a tree with the same amount of nodes. The initial configuration was set to the position $(x, y, z) = (-5, 0, 0)$ and aligned with the world frame (i.e, $\underline{\mathbf{q}}_{\text{init}} = 1 - \varepsilon 2.5\hat{\imath}$), and for each algorithm, a tree was created and expanded until it reached 500 nodes. This process was repeated for 1000 trials and the obtained average computation time in milliseconds is 648.58 ± 93.45 for Algorithm 1 (point sampling), and 159.63 ± 90.75 for Algorithm 2 (input sampling). We can see that the input sampling strategy expands the tree much faster than the point sampling one. The main cause of such difference is because for each iteration, the point sampling algorithm has to check the reachability of all nodes in the tree, while the input sampling method performs the check for only one node.

Table 3.3: Compared performance for 3D version of Algorithms 1 and 2.

	Algorithm 1 point sampling	Algorithm 2 input sampling
Number of trials	1000	1000
Number of successes	768	951
Number of samples	224±198	79 ± 132
CPU time (ms)	33.19 ± 39.46	3.46 ± 7.59
Path length (cm)	10.55 ± 3.92	10.07 ± 1.64

Table 3.4: Performance for 3D version of Algorithm 2 (input sampling) with different κ_{max}.

	$r = 4$	$r = 5$	$r = 6$
Number of trials	1000	1000	1000
Number of successes	1000	982	953
Number of samples	220± 99	459± 209	627 ± 329
CPU time (ms)	6.14 ± 23.40	19.83 ± 66.33	35.40 ± 89.99

SIMULATION 2 Secondly, we compared the ability of each algorithm to find a solution for the same planning situation. In this test, the maximum number of samples for constructing the RRT was set to 1000 and, for each trial, an initial configuration and a goal point were randomly picked with an uniform distribution from the regions of interest. Each set of $\underline{\mathbf{q}}_{init}$ and $\mathbf{p}_{goal}$ was tested in both algorithms. Table 3.3 presents the results for 1000 trials, indicating a better performance of Algorithm 2 (input sampling), which had a higher rate of success and lower processing time when compared to Algorithm 1 (point sampling). From the data, we observe that the input sampling strategy is capable of finding a solution with fewer samples when compared to point sampling.

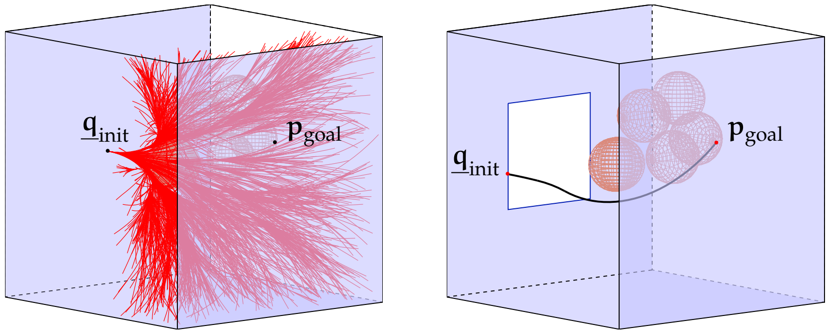
In resume, by changing from point to input sampling, we obtained a new path planner with higher success rate, and more than 9 times faster than the previous one, without increase in the average insertion length. This indicates that the use of the new algorithm is highly advantageous to an intraoperative planner.

3.6.2.2 *Performance of the Arc-RRT with input sampling*

The second simulation set evaluates the influence of the needle steerability in the performance of the Arc-RRT with input sampling (Algorithm 2). For each trial, an initial configuration and a target point are randomly picked from an uniform distribution. The maximum number of nodes for the RRT is 2500. For the kinematic model of the needle, we specified three different minimum radii of curvature (4, 5, and 6 cm). Table 3.4 presents the obtained results for 1000 trials. Feasible paths were found in most of the trials, with a success rate higher than 95% in the worst case. However, we can notice that the performance of the planner is highly dependent on the needle steerability. As we raise the minimum radius of curvature, the harder it gets to find feasible paths, until the

Table 3.5: Performance for the 3D version of the Entry Point Planner.

	$r = 6$
Number of trials	100
Number of successes	100
Number of samples	17798 ± 1390
CPU time (s)	13.17 ± 1.66



(a) Tree with 5000 nodes. (b) Solution from Entry Point Planner.

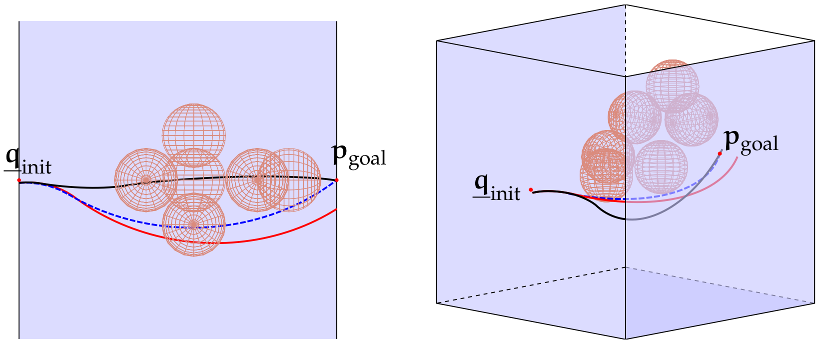
Figure 3.12: Scenario of Simulation C with a difficult combination of $\underline{\mathbf{q}}_{init}$ and $\mathbf{p}_{goal}$ and lower needle steerability.

limit where there is no possible solution for a given combination of start configuration and target position.

3.6.2.3 *Performance of the Entry Point Planner*

An especially difficult combination for the given workspace is the initial configuration at position $\mathbf{p}_{init} = -5\hat{\imath}$ and $\mathbf{r}_{init} = 1$ (which corresponds to $\underline{\mathbf{q}}_{init} = 1 - \varepsilon 2.5\hat{\imath}$), and the goal position at $\mathbf{p}_{goal} = -5\hat{\imath}$, both depicted in Fig. 3.12. In this situation, the target is very close to one of the obstacles and the obstacle is in the middle of the way between the entry point and the target zone, making it difficult to find a feasible path. For this task, no solution could be found after 1000 trials with $r = 6$ cm, which is the lowest steerability from the three cases evaluated in the previous simulation. Fig. 3.12 shows one of the exploration trees generated in the trials, with more than 5000 nodes and no feasible solution found.

Then, we used the Entry Point Planner to search the region of interest for a new feasible start configuration. Table 3.5 presents the simulation results. In total, 100 trials were performed and all of them were able to find a path that connects to $\mathbf{p}_{goal}$, confirming the interest in using the Entry Point Planner for finding a better start configuration whenever the planning requirements may be relaxed. The average CPU time in 100 trials was 13.7 s, which is much bigger than the computational time of the Arc-RRT intraoperative planner, but completely compatible with preoperative use. In fact, even some minutes would be acceptable to assure a good entry point choice for the insertion procedure.



(a) Top view (b) Isometric view

Figure 3.13: Performance of the Intraoperative Replanning. In dashed lines, the initial planned path; in red the trajectory obtained in open-loop and in black the trajectory with replanning.

3.6.2.4 *Replanning evaluation*

The last simulation evaluates the adaptive replanning strategy and its capability of driving the needle to the desired target even with the presence of disturbances and model uncertainties such as positioning errors, tissue deformation and inhomogeneity. In simulation, these effects were modeled as white noises added to the measured tip configuration, and to the applied needle rotation. Perturbations were also induced in the tip initial position and orientation (zero mean, and standard deviation of 1 mm to the tip position and 0.01 rad to the orientation), and to the actual natural curvature which was 25% bigger than the expected value. The needle tip was simulated using the discrete model from (2.1), with natural radius of curvature of 4 cm. The rotational velocity was set to 2 Hz (i.e., $T_{rot} = 0.5$ s). The insertion distance at each cycle was defined to be $\triangle s = 1$ mm.

Fig. 3.13 shows an example of the replanning in action. The initial planned trajectory is shown in dashed blue and it would lead the needle tip to the desired target if it were not for the disturbances. Without the replanning action, the effects of such disturbances cause the needle to follow the trajectory in red, with 8.3 mm final error to the target location. Instead, if the replanning algorithm is used, as the needle insertion advances, the planned trajectory is updated in closed-loop. The constant online replanning results in the path depicted in solid black, with a final error of only 0.06 mm, that is, more than 138 times smaller than the open loop insertion.

3.7 CONCLUSION

3.7.1 *Summary*

Path planning is an important task for autonomous systems and it has been an area of intensive research in the past decades. Its use is specially important in medical applications, when patient-specific information can be used to plan the procedure and allow greater precision, and less risk to the patient. However, planning for nonholonomic systems is a challenging task, specially for an underactuated one such as a steerable needle. In this case, complete planners tend to be computationally inefficient and impractical. A good alternative is to relax the completeness requirement and use sampling-based methods that present good practical results and are proved to be probabilistic complete.

In addition, needle steering procedures must deal with uncertainties like tissue deformation and inhomogeneity, torsional stiffness, imprecision in the kinematic model and other disturbances. To successfully perform the needle insertion procedure, such effects must be compensated. One attractive strategy is to use the path planning in closed-loop so that feedback information of the workspace and of the needle tip configuration is used to adapt the planned path, and consequently, forces its convergence to the desired target while avoiding the obstacles current positions. To be able to accomplish this, we need a path planning method that is fast enough to be used intraoperatively and that takes into account the changes the scene may have suffered due to tissue deformation, for instance, or even patient motion.

This chapter presented a brief description of the fundamental path planning problem and some key concepts such as the configuration space, nonholonomic planning, and completeness. It also introduces the main techniques developed for the area in the past decades and motivates the use of sampling-based planning for many practical problems in robotics, and more specifically, the use of RRT in path planning for steerable needles.

Next, the Arc-RRT planner is proposed. It is an application-specific planning algorithm based on the classic RRT that uses explicit geometry to obtain feasible trajectories that respect the needle nonholonomic constraints. Considering the approximation of the needle movement to that of a 3D unicycle model (Webster III et al., 2006a), the desired needle path is a combination of circular arcs of different curvature radius. The duty-cycling insertion technique is then used to produce insertions with arcs of adjustable curvature.

In the sequence, a closed-loop strategy for needle steering is developed. It combines the Arc-RRT with feedback information to constantly update the needle inputs and adjust the trajectory. This replanning strategy made the insertion procedure more robust to system uncertainties. The simulation results showed that even under perturbations, the needle was able to reach the target with satisfactory precision while an open-loop strategy would fail to avoid the obstacles and to reach the desired position.

Finally, the chapter presents a variation of the Arc-RRT algorithm to be used as a preoperative planning if the starting configuration requirements may be relaxed. Its objective is to find a feasible starting configuration for cases where the needle presents low steerability and/or the target is in a region of difficult access.

3.7.2 *Contributions*

The main contribution of this chapter is the proposal of a closed-loop replanning strategy to be applied during the insertion procedure to compensate for system uncertainties, like tissue deformation and inhomogeneity, positioning errors due to physiological changes and patient motion, and other modeling approximations.

Furthermore, we developed an application-specific planning algorithm for steerable needles, the Arc-RRT planner, that explores the duty-cycling technique for insertion. The proposed Arc-RRT planner has shown to have a good success rate while being fast enough to be used in an intraoperative system. We also presented a variation to use it as a preoperative planner for calculation of feasible entry configurations for challenging workspaces and low steerability needles.

Related publications:

The results presented in this chapter have appeared previously in the following publications:

Bernardes, M. C., Adorno, B. V., Zemiti, N., Poignet, P. & Borges, G. A., "Path planning for steerable needles using duty-cycling," in *Proceedings of the International Congress and Exhibition on Computer-Assisted Radiology and Surgery*, CARS'11, 2011, pp. 293–294.

Bernardes, M. C., Adorno, B. V., Poignet, P., Zemiti, N. & Borges, G. A, "Adaptive path planning for steerable needles using duty-cycling," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'11, 2011, pp. 2545– 2550.

Bernardes, M. C., Adorno, B. V., Poignet, P., Zemiti, N. & Borges, G. A, "Path planning for steerable needles using duty-cycled spinning," in *Proceedings of the International Skills Conference*, 2011, pp. 44–47.

4 BEVELED NEEDLE MOTION CONTROL

The motion control is a fundamental part in any robotic system since it determines the sequence of positions, velocities or torques developed in order to guarantee the execution of a commanded movement. Generally speaking, a motion controller generates the desired motion in a closed-loop control architecture. It calculates a control signal which is transformed by a drive or amplifier into a higher power electrical current or voltage. This electrical signal is then applied to an actuator like an electric motor or a hydraulic pump, resulting in movement of the robotic mechanism. One or more feedback sensors such as optical encoders, Hall effect devices or torque transducers return the current actuator status to the motion controller, closing the control loop.

When modeling the needle kinematics in Chapter 2, we described how the insertion and rotation movements along the needle shaft are related to the tip configuration. We implicitly assumed that we could command arbitrary insertion and rotation inputs, and that such inputs would be faithfully executed by the robotic steering system that actuates the needle. In this chapter we look more closely at how to execute a given input on the needle with the use of a robotic device.

First, we examine two different techniques for robotic beveled needle actuation and their correspondent mechanical structures. Next, we motivate and introduce the use of a telescopic device with a robotic manipulator for performing needle steering. In the sequence, we present the implemented control module used for applying the duty-cycle reference in the robot, and thus, steering the needle accordingly. Last, the chapter presents some simulation and experimental results that validated the proposed motion controller.

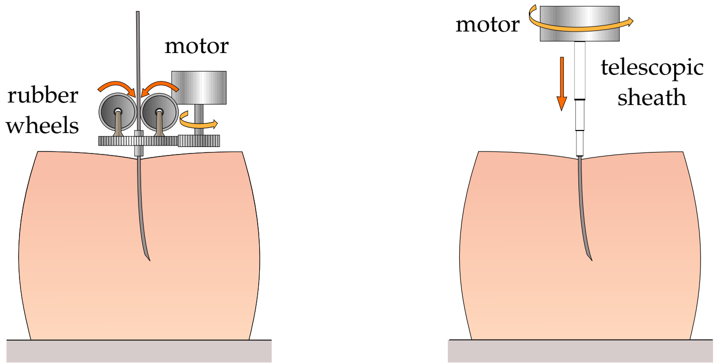
4.1 BEVELED NEEDLE ACTUATION

As discussed in Chapter 2, to correctly steer a beveled flexible needle, one must control its insertion and rotation velocity inputs. Humans are not able to manually insert the needle with a precise velocity and orientation angle. Also, during a freehand insertion, lateral forces or torque about the needle axis may be may accidentally applied, changing the intended actuation.

Hence, to perform beveled needle steering, it is necessary to use a robotic needle driving system. Several techniques can be employed for robotically driving a needle and its choice, as well as the way it is implemented, may have significant influence on the resultant performance.

4.1.1 *Friction Drive and Telescopic Support devices*

So far, two different mechanical designs have been proposed to enable inserting the needle while concurrently controlling its rotation. They differ mainly by the actuation point in the needle and are both presented as follows.



(a) Friction Drive (b) Telescopic Support

Figure 4.1: Devices for flexible needle steering: *(a)* the needle is actuated close to the entry point in the tissue; *(b)* the actuation occurs at the needle base.

The first device for beveled needle steering was proposed by Webster III et al. (2006a), and it was inspired in the same principle as that used in the 'PAKY' (Percutaneous Access of the Kidney) robot (Stoianovici et al., 1997). Webster's *Friction Drive* device actuates near the tissue entry point and consists of two parts: a friction-transmission subassembly responsible for the insertion movement, and a rotation one, which generates spin about the needle axis (see Fig. 4.1a).

The insertion subassembly drives the needle by grasping it with two opposing rubber wheels actuated by a motor-driven worm gear. Rotation of the needle about its shaft is achieved by rotating the insertion subassembly as a unit. Since the needle is firmly grasped by the wheels, rotating the subassembly causes the needle to rotate as well. A needle guide is used to fix the orientation of the needle base with respect to the rubber wheels, reducing unwanted needle rotation as the drive wheels turn. To prevent needle buckling, the needle passes through a small hole which was drilled through the aluminum rod that supports the insertion subassembly. The rod extends to the surface of the tissue into which the needle is inserted.

Besides its simplicity and compact design, the main advantage of this device is that it actuates the needle from close to the tissue entry point. This allows for a better movement transmission from the actuators to the needle tip. However, after preliminary experiments some drawbacks in this design have been identified. It presented slippage in the insertion degree of freedom caused by the properties of stiffness and friction of the tissue. As the needle was inserted further, the slippage effect increased because of the the augmentation of the contact area with the tissue. Another negative effect was caused by uneven contact forces of the needle with the rubber wheels. As a consequence, unwanted needle spinning might occur during the insertion, and the needle tip movement could not be completely determined. Also, measurements of needle force and torque which can be useful for control purposes are difficult to be obtained. The combination of such negative effects mitigate the potential benefits of an actuation close to the tissue entry point.

An alternative solution was presented by Webster III et al. (2005), where a design based in a *Telescopic Support* device was proposed (Fig. 4.1b).

In this case, the needle is directly attached to the motor controlling the spin about the needle shaft, while the rotation subassembly is moved toward the tissue by a linear stage.

The method is named after a telescopic support sheath through which the needle passes during the insertion and is invariably used to prevent the needle from buckling. The last and smallest section of the telescopic support is inserted into a block of rigid material with a small drilled hole, through which the needle passes after leaving the telescope and prior to entering the tissue. This allows the needle to puncture the tissue at a normal angle with its surface.

Since the needle is firmly attached to the mechanism by its base, there is no more relative motion between the needle and the steering device, and hence, the slippage issue no longer exists. The change for a telescoping also allowed more precise spinning and easy attachment of a force sensor. For those reasons, this solution has become the standard mechanism for flexible beveled needle steering and has been largely used in studies in the area (Engh et al., 2006a; Romano et al., 2007; Wood et al., 2010a; Majewicz et al., 2010; Reed et al., 2011). Nevertheless, the actuation in the needle base adds other problems to the design. Despite the use of the telescoping support, small buckling still happens inside the tube, causing uncertainty in the insertion depth. Also, the actuation at the needle base introduces additional uncertainties due to incomplete movement transmission from the needle base to the tip.

4.1.2 *Manipulator with telescopic support*

Due to the previously discussed advantages, we decided to base our system on the telescopic approach. However, instead of using a dedicated device, our idea is to adapt the needle steering system to a robotic manipulator arm. At early research stages, the use of a manipulator to drive the needle might be a good option if such equipment is already available, since it only requires the addition of the telescopic support. This structure can be easily adjusted to the manipulator, resulting in a fast implementation of the system with low costs, provided that the robotic manipulator is previously available. As a side effect, the manipulator has more degrees of freedom and this redundancy adds in complexity to the control.

Fig. 4.2 shows the telescopic support connected to the end-effector of a robotic manipulator. The robot end-effector firmly grasps the needle with a drill chuck positioned at its extremity. The needle passes through a telescopic tube which is attached to the end-effector by two metallic braces. A plastic cast with the internal profile of both the end-effector and the tube is used to assure concentricity. At the end of the telescopic tube, a bearing assures that contact with the phantom surface will not prevent the device from rotating freely. Finally, a metallic u-shaped clamp is used to clasp the telescopic structure to the phantom support which is placed below the robot.

Because both needle base and end-effector are firmly attached by the drill chuck, we consider that the movement of one is entirely transmitted to the other. Hence, when moving the robot end-effector with ν and ω velocities, so will the needle base. If we disregard torsional stiffness effects and consider that the telescopic tube will completely prevent needle buckling, one may assume that the movement of the needle base will correspond to that of its tip.

In the rest of the chapter, we propose and validate the methods used to control the manipulator end-effector, and consequently, the movement of the needle tip.

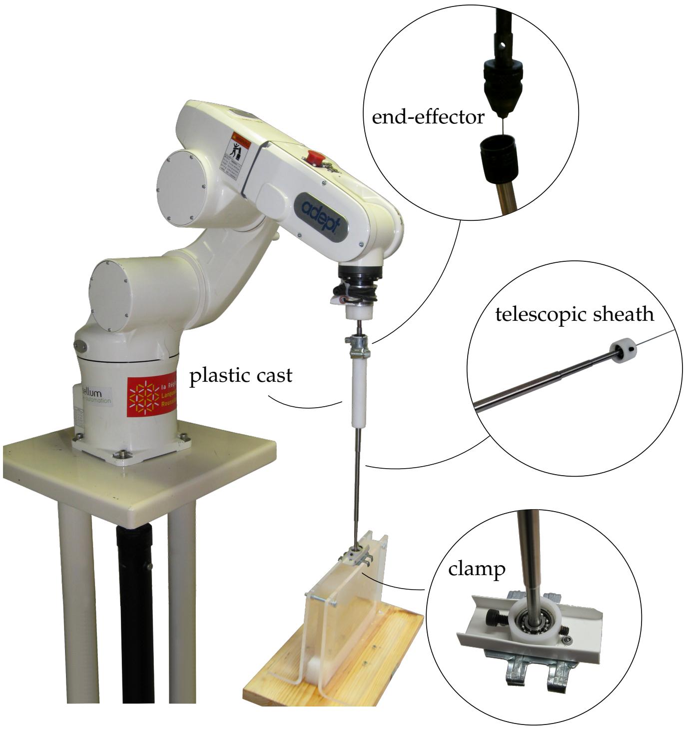


Figure 4.2: Telescopic needle steering mechanism adapted to a robotic manipulator end-effector.

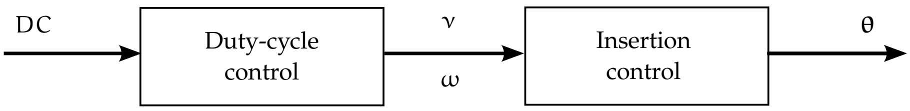


Figure 4.3: Schematic of the steering module. The duty-cycle control receives the desired DC reference and returns the respective ν and ω inputs, which are used by the insertion control to determine the robot joint positions that should be applied.

4.2 STEERING MODULE

In order to perform the motion control, we propose a steering module which is responsible for applying the desired duty-cycle in the needle by actuating the manipulator to which it is attached. As depicted in Fig. 4.3, it consists mainly of two parts: the *duty-cycle control*, which coordinates the duty-cycle by choosing appropriate reference values for ω and ν, and the *insertion control*, which computes the manipulator joints positions θ that will result in the insertion movement toward the tissue. Both parts are described in detail in this section.

4.2.1 *Duty-cycle control*

As defined in Chapter 2, the duty-cycle insertion technique combines small periods T_{ins} of pure insertion—whose resultant motion is an arc with the maximum natural curvature κ_{max}—with small periods T_{rot} of simultaneous rotation—which results in a straight trajectory if $\omega \gg \nu$. Changing the proportions between T_{ins} and T_{rot} results in a arc trajectory with any curvature in the range of κ_{max} and a straight line. To obtain a 2D insertion,

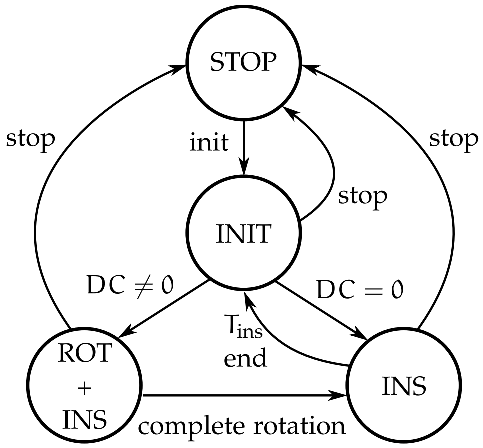


Figure 4.4: Duty-cycle control state-machine.

the needle must lie in a constant plane during the procedure. Hence, while rotating, the needle should spin 180°or 360° turns before entering the pure insertion phase.

The role of the duty-cycle control is to coordinate such rotation and insertion movements by choosing appropriate periods and reference values for the velocities v and ω. The reference for the rotation velocity oscillates between $\omega = 0$ during T_{ins}, or $\omega = \omega_{ref}$ during T_{rot}. By choosing a fixed value for ω_{ref}, the rotation period T_{rot} is also constant since we want the needle to perform a complete turn in order to keep its insertion plane. The period of pure insertion T_{ins} is obtained from (2.4). If we move a fixed insertion distance $\triangle s$ at each cycle, the insertion velocity is variable and given by

$$v = \frac{\triangle s}{T_{DC}}, \tag{4.1}$$

where $T_{DC} = T_{rot} + T_{ins}$.

The duty-cycle control was implemented as a state machine, as presented in Fig. 4.4, and it has the following states:

STOP*:* when there is no duty-cycle signal to be applied;

INIT*:* initializes a cycle by orienting the direction of the tip according to the desired arc and chooses between pure insertion or simultaneous rotation and insertion. Negative DC values indicate negative arc curvature;

ROT + INS*:* applies simultaneous rotation and insertion. This state is kept until a complete turn is performed;

INS*:* applies pure insertion. This state is kept until the expected insertion period ends.

For the duration of a cycle, the DC reference is checked only at the INIT state, when it is used to define T_{ins} and v. For the rest of the cycle, DC is kept constant, until a new INIT state starts and the desired DC value is observed again.

The DC signal can be either positive or negative, with its sign indicating the direction of the arc curvature. If the sign changes when compared to the previous arc direction, the needle orientation is inverted at the INIT state by performing a 180° rotation without simultaneous insertion.

The duty-cycle control checks the last joint angle before entering the INS state to ensure that a complete turn is performed in each cycle. Thus, considering that the rotation of the needle base is completely transmitted to the tip, the needle is expected to stay in a constant 2D plane.

At the end of each cycle, the machine enters the STOP state. At this point, the duty-cycle control checks for the most recent DC reference to be applied in the next cycle. This state is kept whenever there is an invalid signal (i.e., $\|DC\| > 100\%$), which corresponds to an emergency stop required by the operator at some point of the insertion.

4.2.2 *Insertion control*

The insertion control determines the manipulator joints positions $\boldsymbol{\theta}$ that will result in the needle insertion movement with the given velocity inputs defined by the duty-cycle control. As previously shown in Fig. 4.2, in our implementation of the manipulator telescopic system, the needle is inserted into a tissue placed right bellow the robot's end-effector. Thus, the needle insertion motion consists in performing a purely vertical movement toward the tissue, while occasionally spinning around the end-effector joint.

To control the manipulator movements, we decided for a kinematic approach. The kinematic control of robotic manipulators has been widely studied and is a topic of most robotics textbooks (Spong et al., 2006; Siciliano et al., 2009; Dombre & Khalil, 2007). Kinematic controllers do not take into account the nonlinear and coupling effects of the robot dynamic model, but they are conceptually simple and can be easily implemented in position-actuated robots. This technique may be considered a good choice for the needle steering system since large accelerations are not required and the velocities involved in the needle insertion are very low compared to the robot dynamics.

Kinematic control methods can be classified in *joint space* and *task space* methods, depending on the choice of variables used to represent the control problem. In joint space control, they consist in the variables representing the configuration of the articulated manipulator $\boldsymbol{\theta} = [\begin{array}{ccc} \theta_1 & \dots & \theta_N \end{array}]^T \in \mathbb{R}^N$, with N being the number of DOF. In task space control, they are given by the variables describing an assigned motion task in the spatial representation of the end-effector.

The manipulator is naturally described and actuated in the joint space. This generally makes joint space controllers simpler, since each joint can be considered separately. However, the control task is usually defined in the task space, making this approach more frequently used to control the configuration of a manipulator end-effector.

4.2.2.1 *Controlling* ω

The robot used for inserting the needle (see Fig. 4.5) is a commercially available manipulator arm with six torque-controlled joints $\boldsymbol{\theta} = [\begin{array}{ccc} \theta_1 & \dots & \theta_6 \end{array}]^T$ which can be directly actuated by an embedded joint position controller provided by the manufacturer.

The proposed insertion controller adopts a decoupled strategy to actuate the rotation and insertion movements independently. For this, the manipulator is divided into two parts: the first part $\boldsymbol{\theta}_v = [\begin{array}{ccc} \theta_1 & \dots & \theta_5 \end{array}]^T$, composed of joints 1 to 5, is responsible for

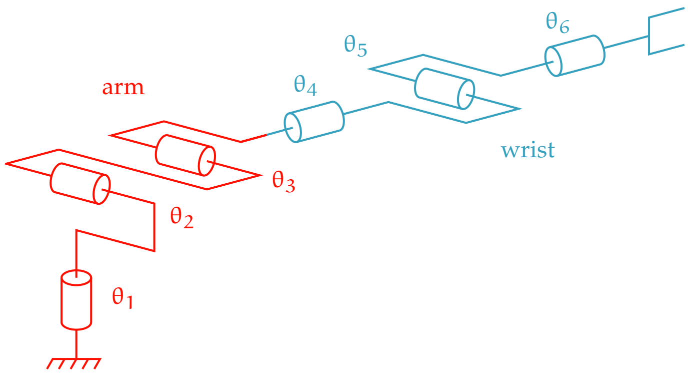


Figure 4.5: Architecture of the Adept Viper s650 robot. This six DOF manipulator consists of an anthropomorphic arm combined with a spherical wrist with three intersecting axes.

making the vertical movement that will result in the desired ν; while the second part $\boldsymbol{\theta}_\omega = \theta_6$ consists in the 6^{th} and last joint, which is responsible for rotating the end-effector with the desired velocity ω.

Since the manipulator supports high-level position-actuated control laws, the rotation control of the last joint is straightforward and consists in integrating the desired ω with the control cycle period, resulting in the discrete joint space control law

$$\theta_6(k) = \theta_6(k-1) + \omega T_c. \tag{4.2}$$

4.2.2.2 *Controlling ν and stabilizing the entry point*

The robot's five remaining DOF are used to control the translational motion of the insertion. In the case of our experimental implementation, it corresponds to moving the end-effector down toward the tissue, while keeping the insertion angle and the entry point constant so that we have a purely vertical movement.

A kinematic task space approach was chosen for implementing the controller. To represent the end-effector configuration, we adopted the set of dual quaternions $\mathcal{H}$ which can be used to simultaneously describe positions and orientations in a single vector. Hence, the configuration $\underline{\mathbf{q}}_E$ of the robot's end-effector can be described in 3D by a rigid transformation from the robot base frame $\mathcal{O}_B$ to its end-effector frame $\mathcal{O}_E$ (see Fig. 4.6). This rigid motion is represented in dual quaternions as

$$\underline{\mathbf{q}}_E = \mathbf{r}_E + \varepsilon \frac{1}{2} \mathbf{p}_E \mathbf{r}_E, \tag{4.3}$$

being $\mathbf{r}_E$ and $\mathbf{p}_E$ the end-effector rotation and translation quaternions, respectively (see Appendix A).

The choice of the dual quaternion representation justifies by the fact that it is a compact mathematical tool that presents many advantages in the point of view of feedback control. The definition of both position and orientation in a common vector simplifies the design of the controller and avoids the computation of complex transformations to obtain the orientation error. Since this formulation relies on the use of quaternions, it is also non-singular, unlike other representations based on sequence of angle rotations (e.g., Euler angles, Roll-Pitch-Yaw). More details on the use of the dual quaternion representation for manipulator kinematics is presented in Appendix B.

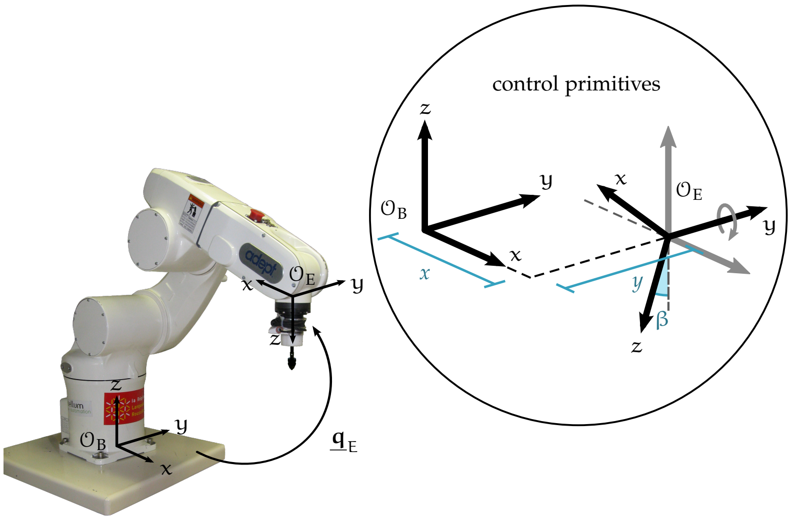


Figure 4.6: Robot's end-effector configuration $\underline{\mathbf{q}}_E$, and control primitives used for stabilization of the entry point during the insertion procedure (shown in blue). Note that for the task space kinematic control, the 5^{th} joint is considered as the robot's end-effector, since the 6^{th} joint rotation is directly controlled at the joint level.

A classic task space control law, proved to be asymptotically stable in the absence of representation singularities (Siciliano et al., 2009), is based in the inversion of the analytical Jacobian mapping. Its adaptation to the task space with dual quaternions is proved to be stable (Pham et al., 2010), and can be written as:

$$\dot{\theta} = \mathbf{J}^{\dagger}_{\underline{q}_E} \left(\operatorname{vec} \underline{\dot{\mathbf{q}}}_{E_{des}} + \mathbf{G}\mathbf{e} \right), \tag{4.4}$$

where $\mathbf{J}^{\dagger}_{\underline{q}_E}$ is the pseudo-inverse of the dual quaternion Jacobian, $\mathbf{G}$ is a positive definite gain matrix, and

$$\mathbf{e} = \operatorname{vec} \left(\underline{\mathbf{q}}_{E_{des}} - \underline{\mathbf{q}}_E \right), \tag{4.5}$$

with $\underline{\mathbf{q}}_{E_{des}}$ and $\underline{\mathbf{q}}_E$ being the desired and measured configurations of the end-effector represented in dual quaternions.

The problem can be reduced to a set-point control by setting $\operatorname{vec} \underline{\dot{\mathbf{q}}}_{E_{des}}$ to zero. Thus, considering a discrete controller, (4.4) becomes

$$\frac{\theta(k) - \theta(k-1)}{T_c} = \mathbf{J}^{\dagger}_{\underline{q}_E}(k)\mathbf{G}\boldsymbol{e}, \tag{4.6}$$

where T_c is the control cycle period. If both sampling time and gain matrix are made constant, then (4.6) reduces to

$$\theta(k) = \theta(k-1) + \mathbf{J}^{\dagger}_{\underline{q}_E}(k)\mathbf{K}\boldsymbol{e}, \tag{4.7}$$

with $\mathbf{K} = {}^{G}/_{T_c}$.

Sometimes, however, instead of using a control error in terms of the end-effector configuration, the task can be more easily defined with different primitives (Adorno et al., 2010). If we consider a generic control primitive $\mathbf{x} = [\, \mathbf{x}_1 \;\; \ldots \;\; \mathbf{x}_M \,]^T$ that defines an specific task, the relation between the joint space vector $\boldsymbol{\theta}$ and $\mathbf{x}$ can be considered as a direct kinematics equation

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}), \tag{4.8}$$

where $\mathbf{f}$ is a continuous nonlinear vector function. Accordingly, it is useful to consider the mapping $\dot{\mathbf{x}} = \mathbf{J_x}\dot{\boldsymbol{\theta}}$, that can be obtained by differentiating (4.8), with $\mathbf{J_x}$ being a $M \times N$ matrix referred as the *task Jacobian*.

Typically, one has $N \geqslant M$ so that the joints can provide at least the number of degrees of freedom required for the task. If $N > M$ strictly, the manipulator is kinematically redundant for that given task.

The general control law becomes

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_{\mathbf{x}}^{\dagger}\mathbf{K}(\mathbf{x}_{\text{des}} - \mathbf{x}), \tag{4.9}$$

where $\mathbf{J}_{\mathbf{x}}^{\dagger}$ is the generalized pseudo-inverse corresponding to the task Jacobian (which varies according to the control primitives), $\mathbf{K}$ is a positive definite gain matrix, and $\mathbf{x}_{\text{des}}$ and $\mathbf{x}$ are the desired and measured primitives, respectively. Under the assumption that the manipulator is kinematically redundant, the solution given by (4.9) provides the least squares solution to the given task, i.e., it minimizes $\|\mathbf{x}_{\text{des}} - \mathbf{x}\|$.

This control law can also be discretized, resulting in an equation similar to (4.7). The advantage of using this Jacobian-based control law is that it can be applied to a stacked vector containing different control primitives, so that all tasks are satisfied at the same time. Also, for a kinematically redundant manipulator, the inverse kinematics problem admits an infinite number of solutions. Hence, the availability of degrees of freedom in excess with respect to those strictly needed to execute a given task can be used to improve the value of performance criteria during the motion.

In our specific case, the controlled variables are represented by the vector of primitives $\mathbf{x} = \begin{bmatrix} \beta & x & y & \nu \end{bmatrix}^T$, where β is the angle between the z-axes of $\mathcal{O}_E$ and $\mathcal{O}_B$ when rotating around the y-axis, x and y are the end-effector's position coordinates, and ν is the insertion velocity given in the z-axis direction of $\mathcal{O}_B$ (see Fig. 4.6). To assure a vertical motion, we want β to be always zero, and x and y to correspond to their initial values.

If we define $\boldsymbol{\theta}_\nu(k) = [\ \theta_1(k) \ \ \ldots \ \ \theta_5(k)\ ]^T$ as the robot's first five joints positions in time step $k = 0, 1, 2, \ldots$, the control input is given by the discrete PID controller:

$$\boldsymbol{\theta}_\nu(k) = \boldsymbol{\theta}_\nu(k-1) + \mathbf{J}_{\mathbf{x}}^{\dagger}(k)\left[\mathbf{K}_p\mathbf{e}(k) + \mathbf{K}_i\sum_{i=1}^{k}\mathbf{e}(i) + \mathbf{K}_d(\mathbf{e}(k) - \mathbf{e}(k-1))\right], \tag{4.10}$$

where $\mathbf{K}_p, \mathbf{K}_i$ and $\mathbf{K}_d$ are the PID gains, and $\mathbf{e}(k) = \mathbf{x}_{\text{des}}(k) - \mathbf{x}(k)$, in which $\mathbf{x}_{\text{des}}(k)$ and $\mathbf{x}(k)$ are the desired and measured values of the control primitives, respectively.

$\mathbf{J}_{\mathbf{x}}^{\dagger}(k)$ is the pseudo-inverse of the Jacobian that satisfies $\dot{\mathbf{x}}(k) = \mathbf{J_x}(k)\dot{\boldsymbol{\theta}}_\nu(k)$ and can be divided in two parts:

$$\mathbf{J_x} = \begin{bmatrix} \mathbf{J}_\beta \\ \mathbf{J}_{\mathbf{p}_E} \end{bmatrix}, \tag{4.11}$$

being $\mathbf{J}_\beta$ related to the angle β, and $\mathbf{J}_{\mathbf{p}_E}$ related to the end-effector position.

Let the current end-effector's configuration be given by $\underline{q}_E(k)$. The angle β is given by

$$\beta = \arcsin(2q_1q_3 - 2q_4q_2), \tag{4.12}$$

with q_i representing the i-th component of $\mathcal{P}\left(\underline{q}_E\right)$, which is the primary part of $\underline{q}_E$. Derivating (4.12), we obtain

$$\dot{\beta} = \mathbf{J}_\beta \dot{\theta}_v, \tag{4.13}$$

with $\mathbf{J}_\beta$ given by

$$\mathbf{J}_\beta = \frac{2\begin{bmatrix} q_3 & -q_4 & q_1 & -q_2 \end{bmatrix}}{\sqrt{1-4(q_1q_3+q_4q_2)^2}} \mathbf{J}_{\mathcal{P}(\underline{q}_E)}. \tag{4.14}$$

The Jacobian $\mathbf{J}_{\mathcal{P}(\underline{q}_E)}$ is defined as the primary part of the analytical Jacobian $\mathbf{J}_{\underline{q}_E}$, as shown in Appendix (B). Note that $\mathbf{J}_\beta$ has a singularity for $\beta = \pm\frac{\pi}{2}$. However, it does not represent a problem for our application since this only happens when the end-effector is positioned horizontally.

The $\mathbf{J}_{\mathbf{p}_E}$ is the Jacobian that satisfies $\operatorname{vec}\dot{\mathbf{p}}_E = \mathbf{J}_{\mathbf{p}_E}\dot{\theta}_v$, where $\mathbf{p}_E$ is the translation quaternion related to $\underline{q}_E$.

Recall from 4.3 that $\mathbf{p}_E = 2\,\mathcal{D}\left(\underline{q}_E\right)\mathcal{P}\left(\underline{q}_E^*\right)$; thus

$$\dot{\mathbf{p}}_E = 2\,\mathcal{D}\left(\dot{\underline{q}}_E\right)\mathcal{P}\left(\underline{q}_E^*\right) + 2\,\mathcal{D}\left(\underline{q}_E\right)\mathcal{P}\left(\dot{\underline{q}}_E^*\right). \tag{4.15}$$

Changing to the vector notation and using the Hamilton operators

$$\begin{aligned}
\operatorname{vec}\dot{\mathbf{p}}_E &= 2\overset{-}{\mathbf{H}}\left(\mathcal{P}\left(\underline{q}_E^*\right)\right)\operatorname{vec}\left(\mathcal{D}\left(\dot{\underline{q}}_E\right)\right) + 2\overset{+}{\mathbf{H}}\left(\mathcal{D}\left(\underline{q}_E\right)\right)\operatorname{vec}\mathcal{P}\left(\dot{\underline{q}}_E^*\right)\\
&= \left(2\overset{-}{\mathbf{H}}\left(\mathcal{P}\left(\underline{q}_E^*\right)\right)\mathbf{J}_{\mathcal{D}(\underline{q}_E)} + 2\overset{+}{\mathbf{H}}\left(\mathcal{D}\left(\underline{q}_E\right)\right)\mathbf{J}^*_{\mathcal{P}(\underline{q}_E)}\right)\dot{\theta}_v,\\
\mathbf{J}_{\mathbf{p}_E} &= 2\overset{-}{\mathbf{H}}\left(\mathcal{P}\left(\underline{q}_E^*\right)\right)\mathbf{J}_{\mathcal{D}(\underline{q}_E)} + 2\overset{+}{\mathbf{H}}\left(\mathcal{D}\left(\underline{q}_E\right)\right)\mathbf{J}^*_{\mathcal{P}(\underline{q}_E)}.
\end{aligned} \tag{4.16}$$

The matrices $\mathbf{J}_{\mathcal{P}(\underline{q}_E)}$ and $\mathbf{J}_{\mathcal{D}(\underline{q}_E)}$ are the primary and dual parts of the dual quaternion analytic Jacobian

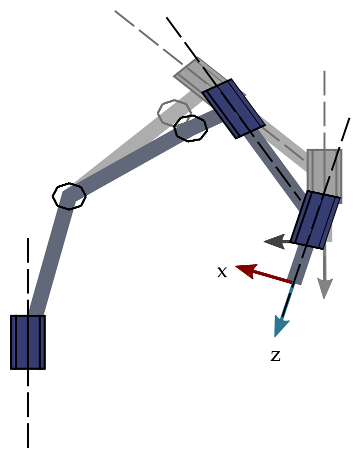
$$\mathbf{J}_{\underline{q}_E} = \begin{bmatrix} \mathbf{J}_{\mathcal{P}(\underline{q}_E)} \\ \mathbf{J}_{\mathcal{D}(\underline{q}_E)} \end{bmatrix}, \tag{4.17}$$

with $\mathbf{J}_{\underline{q}_E}$ obtained from the robot's Denavit-Hartenberg standard convention as demonstrated in Appendix B. Note that in this specific case, only the first five joints of the manipulator robot are considered, and hence, $\mathbf{J}_{\underline{q}_E}$ is a 8×5 matrix.

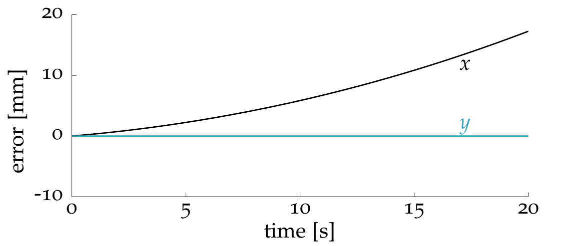
4.3 RESULTS AND DISCUSSION

4.3.1 *Simulation results*

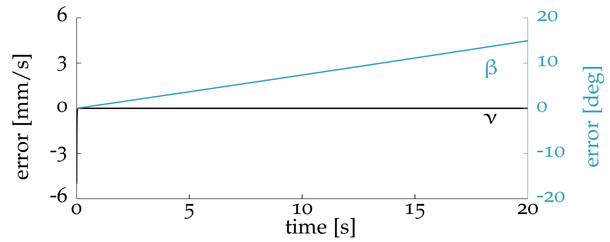
To illustrate how the stacked Jacobian control works, we performed some simulations in Matlab using both the Robotics Toolbox (Corke, 1996) and the DQ Robotics Toolbox (Adorno, 2012). In such simulations, the robot kinematic model is equivalent to that of the Adept Viper s650 (whose D-H parameters are given in Table B.1 from Appendix B). In our control strategy, only the first five robot joints are actuated by the stacked Jacobian controller, while the last joint is directly controlled at the joint level using a different



(a) Manipulator initial (gray), and final (blue) configurations. Note that y-axis points toward the paper.



(b) End-effector position in x (black), and y (blue) directions.



(c) End-effector insertion velocity v (black), and orientation angle β (blue).

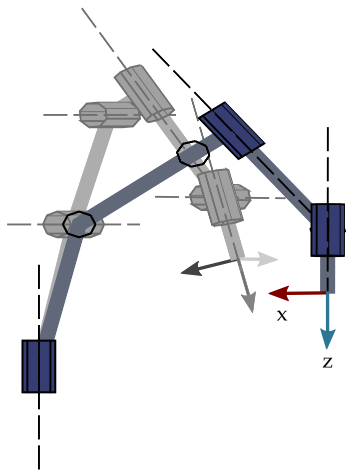
Figure 4.7: Results from simulation controlling the insertion velocity without stabilization of the entry point: *(a)* resultant manipulator movement, *(b)* errors in the entry point position, and *(c)* in the insertion angle and velocity.

control law. Hence, even if the 6^{th} joint is included in the following illustrations, it was not being controlled during the simulations, with its position being constant and given by $\theta_6(k) = 0$, with $k = 0, 1, 2, \ldots$. For convenience's sake, the manipulator is then considered as a five DOF robot during the simulations discussion, and the 5^{th} joint is referred as the robot's end-effector.

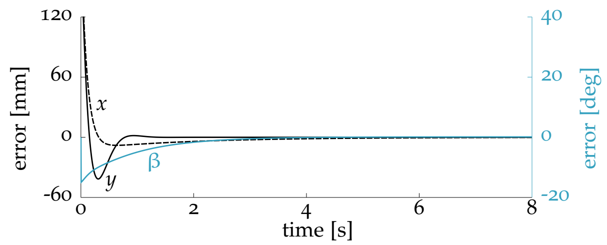
In all simulations, the robot's initial joints positions are $\theta_v(0) = [\ 0 \quad -\frac{\pi}{4} \quad \pi \quad 0 \quad \frac{\pi}{4}\ ]^T$. As a result, the end-effector is initially positioned at $x = 53.82$ cm, $y = 0$ cm, and $z = 38.10$ cm with respect to the base, and its z-axis is vertically aligned. Thus, the insertion entry point is given by the initial x and y values, and $\beta = 0°$, while the desired insertion velocity was chosen to be $v = 5$ mm/s.

In the first simulation, we only controlled the insertion velocity v, so that the vector of primitives is reduced to $\mathbf{x} = v$, and the task Jacobian $\mathbf{J_x}$ is given by the third row of $\mathbf{J}_{\mathbf{p}_E}$, which is related to the end-effector z-axis direction. An insertion of 10 cm length was simulated with the described controller using PID gains set to $\mathbf{K}_p = 0.01$, $\mathbf{K}_i = 100$ and $\mathbf{K}_d = 0$. The resultant manipulator motion is shown in Fig. 4.7. We can see from the graphics that although the velocity errors were approximately zero, the end-effector angle and position errors increased as the end-effector moved down. Consequently, when performing a needle insertion with such motion controller, the end-effector would induce lateral forces around the entry point, probably causing the needle to slice through the tissue.

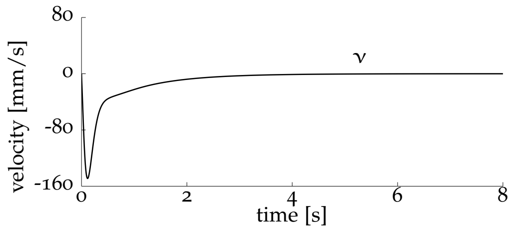
Then, in a second simulation, we evaluated the stabilization of the end-effector around a given entry point. This is accomplished by choosing $\mathbf{x} = [\ \beta \quad x \quad y\ ]^T$ as control primitives. In this case, the task Jacobian $\mathbf{J_x}$ is obtained by stacking $\mathbf{J}_\beta$ given by (4.14), and



(a) Manipulator initial (gray), and final (blue) configurations. Note that y-axis points toward the paper in the final configuration.



(b) End-effector position in x (black), and y (blue) directions.



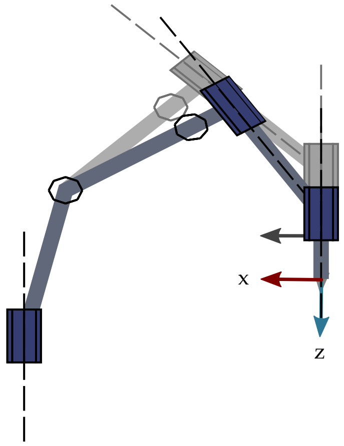
(c) End-effector insertion velocity ν (black), and orientation angle β (blue).

Figure 4.8: Results from simulation of stabilization of the entry point: *(a)* resultant manipulator movement, *(b)* errors in the entry point position, and *(c)* in the insertion angle and velocity.

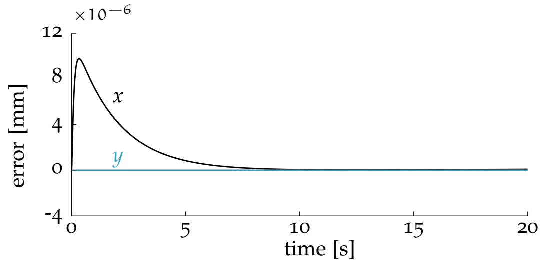
the first two rows of $\mathbf{J}_{\mathbf{p}_E}$, which are related to the end-effector x and y directions. For this simulation, the PID gains were set to 3×3 diagonal matrices $\mathbf{K}_p = \text{diag}(\begin{matrix} 1 & 10 & 10 \end{matrix})$, $\mathbf{K}_i = \text{diag}(\begin{matrix} 0.1 & 5 & 50 \end{matrix})$ and $\mathbf{K}_d = \text{diag}(\begin{matrix} 0 & 0 & 0 \end{matrix})$. The simulation starts with the end-effector initial configuration being disturbed in 2 cm for both x and y coordinates, and in $15°$ for β (see illustration in Fig. 4.8a).

The graphic from Fig. 4.8b shows the evolution of the entry point errors after 8 s. We can see that they stabilize to the desired values with control errors being approximately zero after 4 s. However, the insertion velocity ν is not constant, presenting a pick value at the beginning of the simulation, when the control errors were high, and progressively reducing to zero as the end-effector approached the desired entry configuration. Nevertheless, the analytical Jacobian $\mathbf{J}_\mathbf{x}$ is a 3×5 matrix with full-rank. Consequently, the stabilization task leaves the robot with redundant degrees of freedom that can be used to control ν.

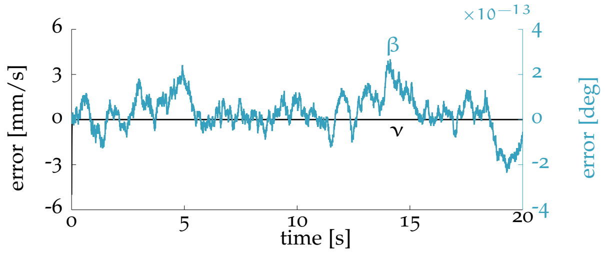
Finally, we repeated the same insertion procedure from the first simulation, but this time combining both insertion velocity and entry point stabilization controllers. Consequently, the vector of control primitives becomes $\mathbf{x} = \begin{bmatrix} \beta & x & y & \nu \end{bmatrix}^T$, and the task Jacobian is given by (4.11). The PID gains were set to 4×4 diagonal matrices $\mathbf{K}_p = \text{diag}(\begin{matrix} 1 & 10 & 10 & 0.01 \end{matrix})$, $\mathbf{K}_i = \text{diag}(\begin{matrix} 0.1 & 5 & 5 & 100 \end{matrix})$, and $\mathbf{K}_d = \text{diag}(\begin{matrix} 0 & 0 & 0 & 0 \end{matrix})$. As seen in Fig. 4.9, for the duration of a 10 cm insertion, the proposed kinematic controller was able to control the desired velocity $\nu = 5$ mm/s, while using the robot's redundant DOF to stabilize the entry point position and orientation according to our implementation requirements, which presumes a purely vertical movement toward the tissue placed below the end-effector.



(a) Manipulator initial (gray), and final (blue) configurations. Note that y-axis points toward the paper.



(b) End-effector position in x (black), and y (blue) directions. Note that the scale is at $\times 10^{-6}$.



(c) End-effector insertion velocity ν (black), and orientation angle β (blue). Note that the scale corresponding to β is at $\times 10^{-13}$.

Figure 4.9: Results from simulation controlling the insertion velocity with stabilization of the entry point: *(a)* resultant manipulator movement, *(b)* errors in the entry point position, and *(c)* in the insertion angle and velocity.

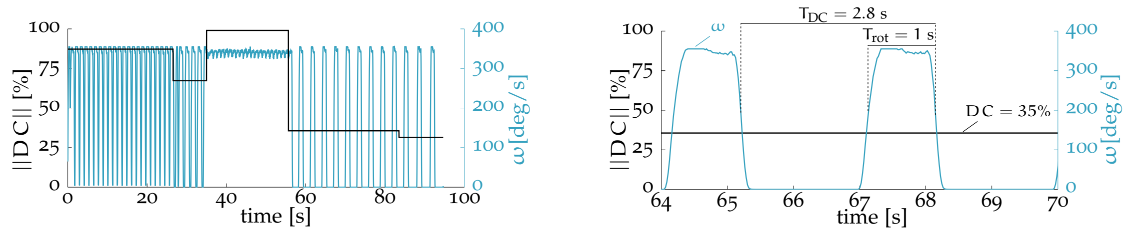
4.3.2 *Experimental results*

The complete steering module was implemented and tested in a real manipulator with telescopic support as described in Section 4.1. Both duty-cycle control and insertion control were run in a PC with real-time Linux RTAI at control period $T_c = 1$ ms, and connected by firewire communication to the manipulator robot. We used an Adept Viper s650 manipulator arm which is a commercially available robot with six torque-controlled joints that can be directly actuated by an embedded position joint controller. Consequently, it supports the use of high-level position-actuated control laws, as proposed in this chapter.

Considering the intended application, the Adept Viper s650 has a reasonable payload-to-weight-ratio (5/28 Kg) and is equipped with high resolution absolute encoders, providing good repeatability (± 0.020 mm). According to the velocity limitations of the robot 6^{th} joint, we chose $\omega_{ref} = 360°/s$ for performing the needle duty-cycled rotation. Consequently, each insertion cycle lasts at least 1 s, corresponding to the fixed period T_{rot} of simultaneous rotation and insertion. Also, for safety purposes, we made $\triangle s = 2$ mm, limiting the insertion velocity to $\nu_{max} = 2$ mm/s.

The experiment consisted in providing five different DC references to the control module and in verifying if the proposed control architecture resulted in the desired insertion motion. Fig. 4.10 shows the 6^{th} link rotation velocity ω, which was coordinated by the state machine from the duty-cycle control. In this figure, one can observe that ω values alternated between $\omega = 0$ and $\omega = 360°/s$, as defined for this experiment.

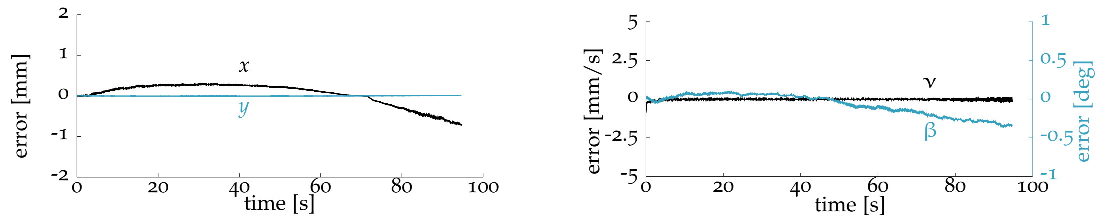
Also, the period T_{rot} is kept constant and equal to 1 s (see Fig. 4.10b), while the insertion period T_{ins} varies according to the DC reference. As illustrated in Fig. 4.10a, larger duty-



(a) Duty-cycled rotation velocity ω corresponding to five different DC references.

(b) Zoom showing the obtained duty-cycle period for a reference of DC = 35%.

Figure 4.10: Duty-cycle control results: DC reference (black) and correspondent ω output (blue).



(a) Errors in the end-effector position in x (black), and y (blue) directions.

(b) Errors in the end-effector insertion velocity ν (black), and orientation angle β (blue).

Figure 4.11: Insertion control results.

cycle references correspond to shorter periods T_{ins} when no rotation is applied, until the limit where the rotation is continuous for DC = 100%, as described in (2.4).

The insertion control also presented good performance, with position errors of less then 1 mm, orientation error lower then 0.5 degree, and velocity error in the order of 0.2 mm/s during an insertion with 14.14 cm length (see Fig. 4.11).

4.4 CONCLUSION

4.4.1 *Summary*

For performing automatic steering of flexible beveled needles, one may actuate the needle in two different ways: from the tissue entry point, or from the needle base. The actuation from the entry point would be the ideal situation due to its proximity to the needle tip, which results in a better input transmission. However, the mechanisms designed for such strategy present a series of drawbacks such as slippage and unwanted spinning of the needle, invalidating its potential advantages.

Needle base actuation is currently the standard choice in literature for beveled flexible needles. Hence, we based our robot-assisted system in this approach by adapting a telescopic support device to the end-effector of a robotic manipulator. Although a with 2 DOF dedicated device might be more appropriate for future clinical trials due to its simplicity and portability, at early research stages the use of a manipulator might be a good option if such equipment is available, requiring only the addition of a telescopic support to the already existing actuation structure. As a result, one can obtain a fast implementation of the steering system with low costs. On the other hand, the manipulator is kinematically redundant, adding complexity to the motion control.

To actuate the robotic arm, we proposed a steering module consisting of two parts: the duty-cycle control, which is a state-machine that coordinates the duty-cycled rotation by choosing appropriate reference values for ω and ν; and the insertion control, responsible for defining the manipulator joint positions that will result in the desired ω and ν. For the insertion control, we adopted a decoupled strategy in which the first five DOF of the robot are used to control the vertical insertion, while the sixth and last DOF is used to control the rotation independently.

The motion controller was evaluated both in simulation and with experimental trials conducted with a commercially available robotic manipulator.

4.4.2 *Contributions*

The main contribution of this chapter is the proposal of a testbed for robot-assisted needle steering using a commercially available robotic manipulator combined with a telescopic-based device. Because it requires only the addition of a telescopic support to the robot's end-effector, it might be a good option for having a fast implementation of the steering system during early stages of the research.

In addition, we have developed a motion controller capable of performing a vertical duty-cycled insertion movement, while dealing with the kinematic redundancy of the manipulator. Even though we did not test non-vertical movements, we believe that the presented approach based in stacked task vectors could be used to produce insertions at other orientation angles if one choses appropriate control primitives and control references to describe the desired motion. Despite it not being important at the current development stage, it may be necessary to test other insertion configurations if the experimental platform with manipulator is considered for future research with animal experiments.

Related publication:

The results presented in this chapter have appeared previously in the following publication:

> **Bernardes, M. C.**, Adorno, B. V., Poignet, P. & Borges, G. A., "Semi-automatic needle steering system with robotic manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, ICRA'12, 2012, pp. 1595–1600.

5 ROBOT-ASSISTED BEVELED NEEDLE STEERING

The basic process for robot-assisted needle steering involves the identification of the insertion target and obstacles to be avoided within the body, planning the needle trajectory, inserting the needle with the robotic device, verifying the current needle placement, and iteratively adjusting the needle motion when necessary. Generally speaking, the system proposed in this thesis consist in a combination of physical systems, and computational algorithms that plan and control the robotic actuation in closed-loop.

While Chapters 3 and 4 introduced the path planning and motion control, respectively, in this chapter these two parts are combined together to obtain an automatic system for beveled needle steering.

First, we describe the complete architecture and how the planning and control parts interact to control the needle insertion in closed-loop. In the sequence, we present the experimental setup used to implement the proposed system, and the results obtained from *in vitro* insertions into transparent tissue phantom and standard camera imaging.

5.1 CLOSED-LOOP STEERING SYSTEM

The complete robot-assisted needle steering system is depicted in Fig. 5.1, and can be separated in two main computational modules—the *adaptive motion planning* and the *steering control*. The adaptive motion planner consists basically in the path planning algorithms presented in Chapter 3, and is responsible for calculating and updating the needle tip desired trajectory. The steering control strategy was presented in Chapter 4, and it is responsible for defining the manipulator arm movements in order to apply a sequence of duty-cycle inputs that will drive the needle through the planned trajectory.

These modules interact with a physical system, composed by the manipulator robot, the needle attached to its end-effector, and the tissue being punctured. Sensors extract information on the physical system and provided it as feedback to the planning so that

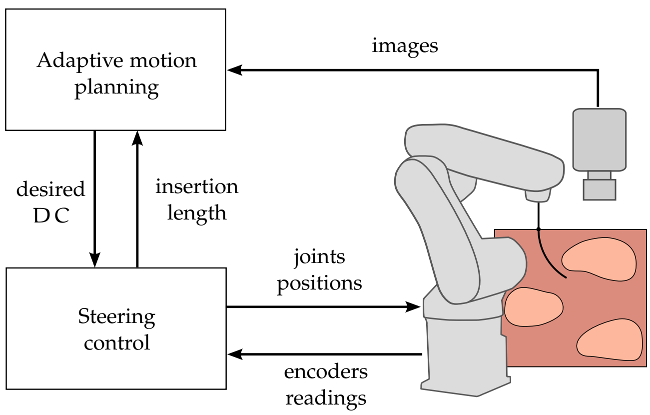


Figure 5.1: Complete robot-assisted needle steering system.

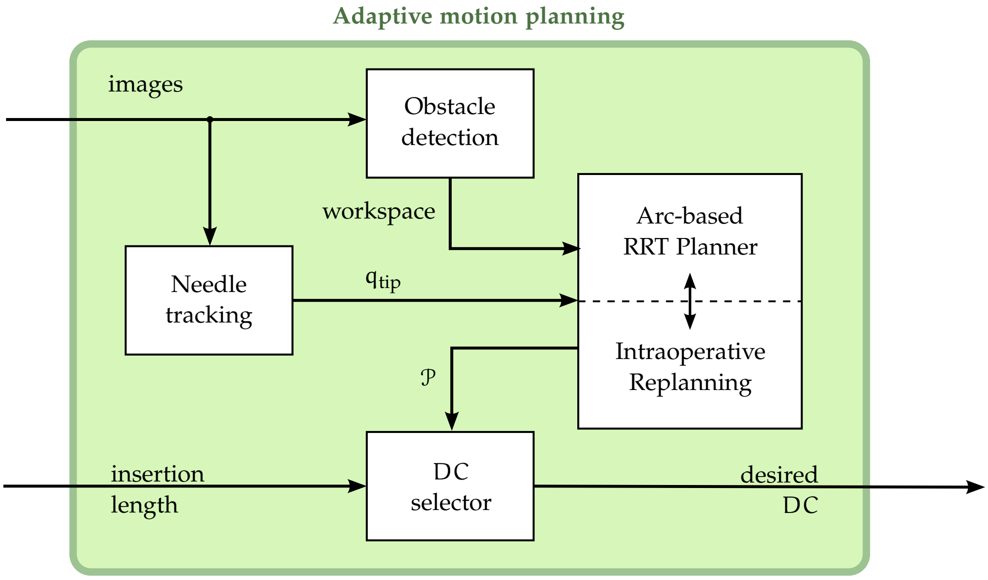


Figure 5.2: Module of the adaptive motion planning.

it can adjust the desired needle trajectory accordingly to the observed behavior. In this thesis, we propose an image-guided system where the feedback information is extracted from images. Consequently, the integration of all parts operate as a self-regulating system that works in a closed-loop.

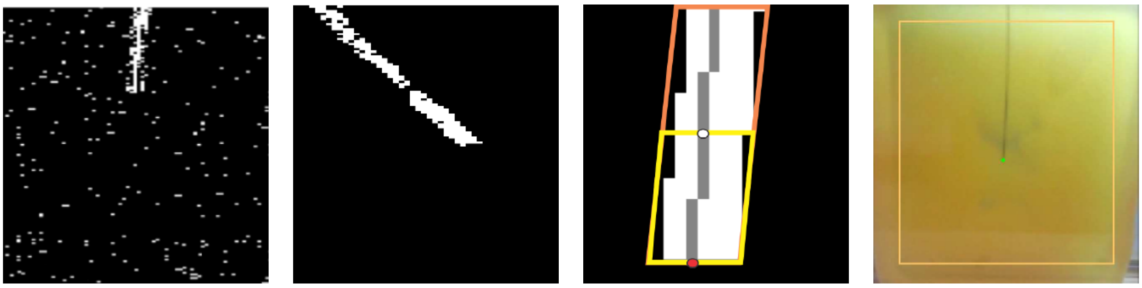
5.1.1 *Adaptive motion planning*

The adaptive motion planning module (see Fig. 5.2) calculates a trajectory for the needle tip using the 2D version of the Arc-RRT algorithm presented in Chapter 3. Recall that the objective of the 2D Arc-RRT planner is to find a combination of circular arcs capable of taking the needle from its initial configuration $q_{init} = [x_{init}, y_{init}, \varphi_{init}]^T$ to a final position $p_{goal} = [x_{goal}, y_{goal}]^T$ while avoiding obstacles present in the workspace and respecting the needle's nonholonomic constraints. The workspace limits and obstacles, the initial configuration, and the goal position are manually defined during the preoperative initialization through a user interface which we developed for the system.

During the insertion procedure, the obtained path $\mathcal{P}$ is constantly updated by the intraoperative replanning until the needle tip is sufficiently close to the target. The intraoperative replanning uses the current obstacles and target positions, and the current needle tip configuration provided by the tracking module to adjust the path $\mathcal{P}$ when possible or to recalculate a new one if a collision is detected in the updated arcs, or if the new curvature does not respect the maximum limit κ_{max}.

5.1.1.1 *Needle tracking*

The needle tracking is responsible for obtaining the current tip configuration q_{tip} to be used by the intraoperative replanning. At each insertion cycle, q_{tip} is updated by the tracking module and the intraoperative replanning algorithm is run to update the desired needle path $\mathcal{P}$. In our testbed implementation, the needle tracking consists in an image processing module that extracts the needle tip from a video sequence. The images are



(a) Pre-processing (b) Blob extraction (c) Tip location (d) Tracked needle tip

Figure 5.3: Progress of the needle tracking algorithm proposed in (Dorileo, 2011) (courtesy of LIRMM). Its processing time depends on factors like the ROI size, number of detected blobs and computational power. The images above correspond to an implementation in a Intel Neon 1.86 GHz processor, which presented an average period of 278 ms.

provided by a standard CMOS camera which is placed in front of a transparent tissue phantom.

Tissue phantoms have been used extensively in medical robotics research since they offer a controlled environment for repeatable measurements, and do not require the previous approval of experimental protocols by a committee. Moreover, they can be relatively transparent allowing to monitor the progress of experiments without the need of an expensive medical imaging system, being more appropriate to validate and refine the methodology at early development stages.

The used tracking algorithm was developed by Dorileo (2011). It first extracts the needle shaft from the camera image, and then, identifies the tip location to obtain its 2D configuration with respect to the image reference frame. The algorithm basically consists of three steps roughly described as follows and presented in detail in (Dorileo, 2011).

PRE-PROCESSING: The pre-processing phase uses filters to prepare the image to the image processing steps. More specifically, a Gaussian smooth filter and a Laplace filter are used to reduce noises and detect borders, respectively. Then, the workspace limit is defined as the Region of Interest (ROI). The use of a ROI restricts the processing window, and consequently, saves computational time. Finally, the pre-processing phase converts the image to binary using a minimal threshold to preserve the maximum of information from the original image. Fig. 5.3a shows an example of pre-processed image.

SHAFT LOCATION: The first step of the processing phase is the shaft location, where the algorithm recognizes the needle shaft in the image. The shaft segmentation is obtained with two geometrical filters that select blobs, which are groups of connected pixels candidates to the object of interest (see Fig. 5.3b). The first blob filter elects the most relevant blobs in the image. Then, a second filter selects the biggest blob as the main one, and all other blobs are added to the main if they are close enough (at a given distance threshold), or rejected otherwise. At the end, only a single blob is left, which is considered to represent the needle shaft. Then, a curve fitting function is used to estimate the needle shaft from the remaining blob pixels.

TIP LOCATION: The needle tip is considered to be the lowest pixel in the blob (the highest one corresponds to the needle entry point in the tissue). Hence, two parallelograms are used to define a region in the neighborhood of the curve extremity (see Fig. 5.3c). The final tip position corresponds to the median of the external parallelogram border (red point), while the tip orientation is approximated by the slope of the line segment formed by the tip and the median of the internal parallelogram border (white point). The size of the parallelogram is an algorithm parameter to be set according to the image's m/pixels ratio.

The C++ implementation of this algorithm in a PC with Intel Xeon 1.86 GHz, 2.73 GB memory, presented an average period of 278 ms, although the processing times vary a lot depending on the quality of images and chosen parameters.

5.1.1.2 *Obstacle detection*

The obstacle detection is responsible for updating the insertion workspace during the procedure. It detects the target and obstacles current positions from the provided images, and returns the workspace free-space used by the path planning algorithm.

In our implementation, it consists of a user interface through which the operator defines the insertion task at the beginning of each insertion by manually selecting the workspace boundaries, the obstacles, the desired target and the initial needle tip configuration, as described in more detail in Section 5.2.2.

Hence, for the *in vitro* experiments presented in this chapter, the position of obstacles and target is considered to be constant during the whole procedure, although the adaptive motion planning algorithms are compatible with dynamic workspaces, which are expected during *in vivo* trials due to tissue swelling, respiratory motion, and other physiological effects.

5.1.1.3 *Duty-cycle selection*

The output of the Arc-RRT algorithm is the planned path $\mathcal{P} = [\ \mathcal{A}_1 \ \ \dots \ \ \mathcal{A}_n\ ]$, given by the concatenation of n arcs with each 2D arc $\mathcal{A}$ given as

$$\mathcal{A} = \begin{bmatrix} q_A^T & q_B^T & \kappa \end{bmatrix}^T, \tag{5.1}$$

with κ being its curvature, and $q_A = [x_A, y_A, \theta_A]^T$ and $q_B = [x_B, y_B, \theta_B]^T$ its start and end configurations, respectively.

The duty-cycle selector receives the planned path $\mathcal{P}$ and calculates the correspondent duty-cycle sequence $DC(s)$ parametrized in insertion length that will take the needle from its insertion point to the goal while following the desired path. The duty-cycle sequence is recomputed every time $\mathcal{P}$ is updated by the intraoperative replanning, and it is obtained from the accumulated arcs lengths and from the linear relation between duty-cycle and arc curvature:

$$\begin{aligned} DC(s) &= 1 - \frac{\kappa_i}{\kappa_{max}}, \quad l_{i-1} \leqslant s < l_i, \quad i = 1, \dots, n \\ l_i &= \begin{cases} 0 & \text{if } i = 0 \\ \frac{\theta_{B_i} - \theta_{A_i}}{\kappa_i} & \text{otherwise,} \end{cases} \end{aligned} \tag{5.2}$$

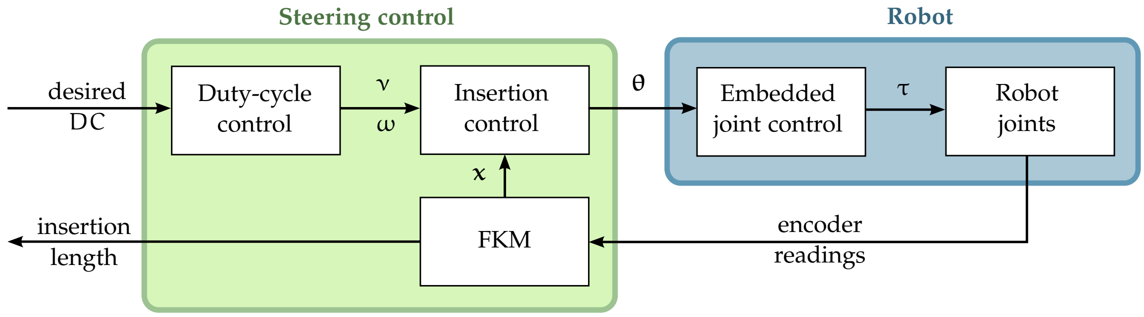


Figure 5.4: Module of the steering control.

with i being the number of a given arc, l_i being the arc length, and κ_i, θ_{B_i},and θ_{A_i} its parameters.

Considering that no buckling occurs during the needle insertion, and that the robot manipulator is moving in a straight line trajectory, the current needle insertion length s corresponds to the manipulator total vertical displacement. According to the current insertion length, the desired DC reference is then selected from $DC(s)$ and sent to the steering control module.

5.1.2 *Steering control*

The steering control actuates the robot using the control-laws presented in detail in Chapter 4. It receives the desired DC signal from the adaptive motion planning and applies it in the needle by actuating the manipulator to which it is attached.

As depicted in Fig. 4.3, the duty-cycle control receives the DC reference, and coordinates the needle rotation and insertion movements by choosing appropriate reference values for the velocities v and ω. Following, the insertion control uses a task control strategy to compute the manipulator joints positions θ that will result in the given v and ω values. Its output is received by the manipulator robot, which has an embedded joint position controller that actuates directly at the torque-controlled joints.

The current end-effector configuration is obtained from the joints encoders readings by the Forward Kinematic Model (FKM). This feedback information is employed to calculate the task vector x used in the insertion control-law, and the current insertion length, which is given by a comparison between the initial and current end-effector positions, and is used in the adaptive motion planning.

5.2 EXPERIMENTAL SETUP

To implement the complete closed-loop system, the adaptive motion planning and steering control modules were divided into two different computers which communicate through a point-to-point TCP connection (see Fig. 5.5). The adaptive motion planning executes in a PC equipped with a high-speed camera DALSA 1M75 (FujiPhoto Lens 1:14/25 and 512×512 video resolution). The camera was placed in front of the robot, and with the closest possible focus distance to the tissue phantom. At this distance, the visual tracking was able to provide the current needle tip position with approximately 1 mm resolution.

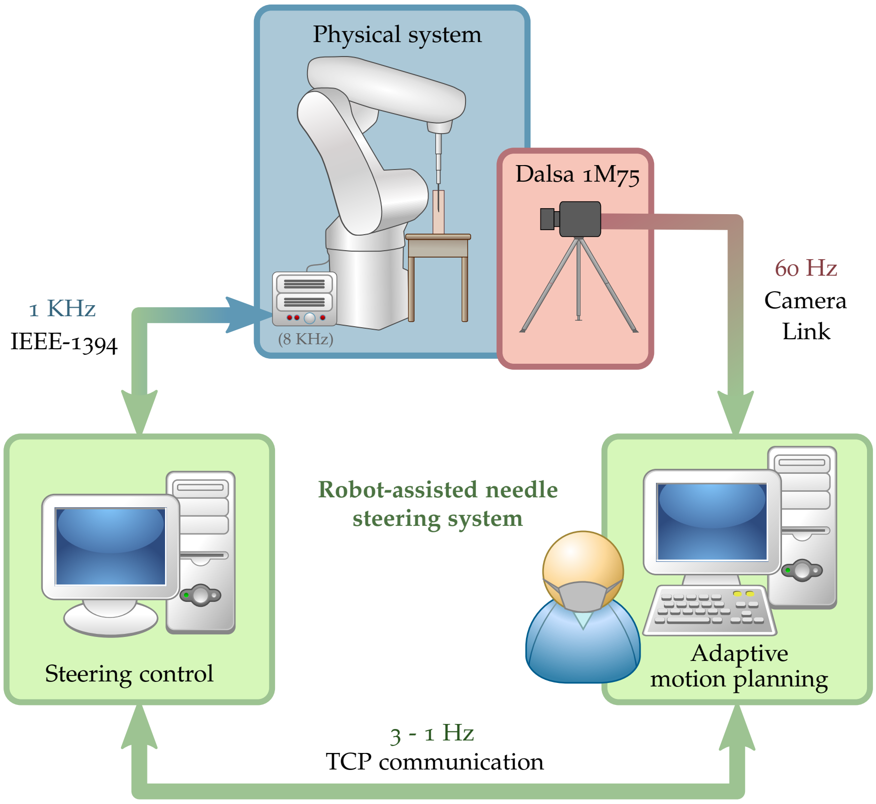


Figure 5.5: Schematic of the experimental setup built to validate the proposed robot-assisted steering system.

The steering control module is run in a PC with real-time Linux RTAI and connected by firewire communication to an Adept Viper s650 manipulator robot. The Viper robot is equipped with an Adept MotionBlok60-L module, which actuates at 8 KHz directly at the joint motors, providing embedded joint position control. The steering control, in its turn, reads the robot encoders and provides a new joint position reference at 1 KHz. The adaptive motion planning was not implemented in a real-time environment, but extensive tests showed that the DC reference was updated with frequencies between 3 Hz and 1 Hz, with this variance being due mainly to the computational time required by the needle tracking algorithm to process the images.

According to the velocity limitations of the robot 6^{th} joint, we chose $\omega_{ref} = 360°/s$ for performing the needle duty-cycled rotation, which results in insertion cycles of at least 1 s. Also, for safety purposes, we made $\triangle s = 2$ mm, limiting the insertion velocity to $v_{max} = 2$ mm/s. The state machine from the duty-cycle control checks the DC reference only at the beginning of a cycle and keeps it constant until the next one. Hence, the DC values received from the adaptive motion planning are stored in a buffer by the steering control. When a new cycle begins, only the most recent DC reference is used.

The tissue phantom consists of a molded block of water- or plastic-based material which lies upon a vertical acrylic support mounted below the robot arm. The molded block is made as thin as possible to assure the maximum visibility of the needle through it. We have tested two different tissue phantoms: one made of plastisol, which is a suspension of PVC particles in plasticizer (Plastileurre Rigide, Bricoleurre SARL), and a water-based phantom composed of agarose (a plant-based jellifying) and 10% glycerin.

In turn, the needles are prototypes made of nitinol wire. Nitinol was chosen for its superelastic properties: the needles can curve in the tissue with a fairly small radius of curvature without plastically deforming. Two prototypes with hand-grounded beveled tips have been used: the larger needle was 0.8 mm in diameter with a 16° bevel angle, while the smaller needle was 0.508 mm in diameter with a 20° bevel angle.

To conclude the setup, a series of calibrations and identification procedures must be fulfilled in order to obtain the relationships between the image frame, the robot base and the 2D needle insertion plane. Also, before each insertion experiment, an initialization step is performed to define the workspace limits, obstacles, initial configuration and final target. Such procedures are described as follows.

5.2.1 *Calibration and identification procedure*

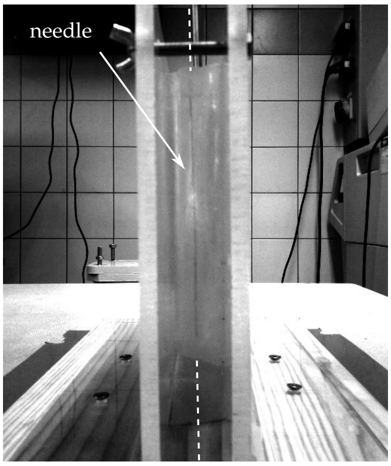
Because of manufacturing defects which are present in a compound lens, the image normally suffers from distortions that should be compensated before extracting feedback information. In addition, the camera image plane should correspond to the same plane of the tissue and its acrylic support. Hence, we perform a series of calibration and image processing procedures to identify the camera distortion parameters and rectify the images with respect to the tissue surface (see details in Appendix C).

Also, to be correctly visualized through the transparent tissue phantom, the needle should stay within a few millimeter of its surface. However, if the needle bevel is not correctly aligned with the phantom width direction, it may deflect during the insertion and hit one of the acrylic support plates. In this case, the needle not only diverges from the desired 2D insertion but also loses actuation since it is no longer surrounded by tissue to make it bend.

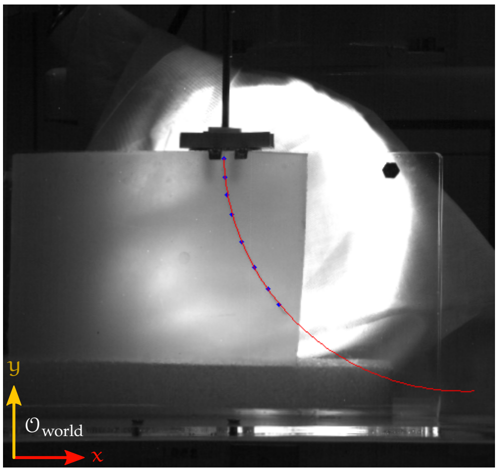
Hence, after the needle is attached to the robot, a calibration process is performed to minimize the out of plane motion. The process consists in doing several insertions with 0% duty-cycle—that is, pure insertions—and adjusting the needle orientation up to a few degrees until it can be completely inserted without going out of the plane more than a few millimeters, as shown in Fig. 5.6a. This procedure must be repeated every time the needle is detached from the end-effector drill chuck, or when the phantom support alignment is changed with respect to the xy-plane.

Finally, the natural curvature κ_{max} for the needle-tissue combination should be identified. For this, a series of pure insertions is performed and the images of each procedure are stored and evaluated offline. As exemplified in Fig. 5.6b, points along the needle shaft (blue) are manually selected and fitted to a circular curve (red). For each insertion, the curve fitting is obtained with an optimization through the `nlinfit` command from Matlab. This function implements a non-linear least-square method that uses Gauss-Newton descent algorithm with Levenberg-Marquardt modifications for global convergence. The final estimated natural curvature κ_{max} is given by the mean of all minimization results.

Since the natural curvature is a parameter that depends on both needle and tissue properties, during a set of experimental trials we employ the same tissue at multiple times in order to maintain the same experimental conditions for all insertions. However, care is taken to insert the needle at different locations each time, so that the holes cut from



(a) Lateral view of a needle completely inserted and aligned with the tissue phantom plane.



(b) Frontal view of a pure needle insertion (0% duty-cycle) used for the offline identification of the natural curvature. Observe that additional lighting is placed at the back of the phantom to increase needle visibility during the insertions.

Figure 5.6: Curvature identification procedure.

a previous insertion do not influence the subsequent experiments. To avoid accidental phantom rotation between trials, we use two metallic guides so that the phantom support can slide along the longitudinal direction while keeping its initial xy-plane alignment.

5.2.2 *Initialization procedure*

One may not forget the importance of including the physician in the system's decision process, not only to observe and interfere when an eminent fault is detected, but also to be able to indicate preferred paths and regions to be avoided based on his practice experience. Thus, a user interface was developed to allow the manual selection of the insertion environment (see Fig. 5.7), and to suspend the insertion procedure in case of potential danger to the patient.

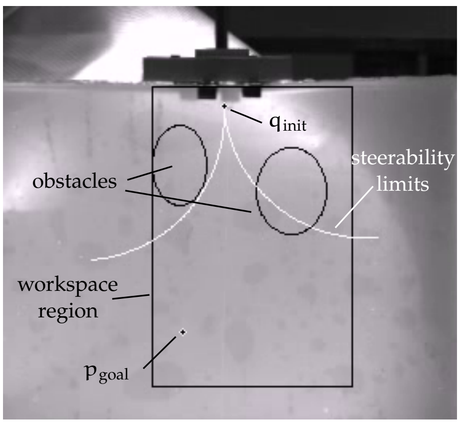
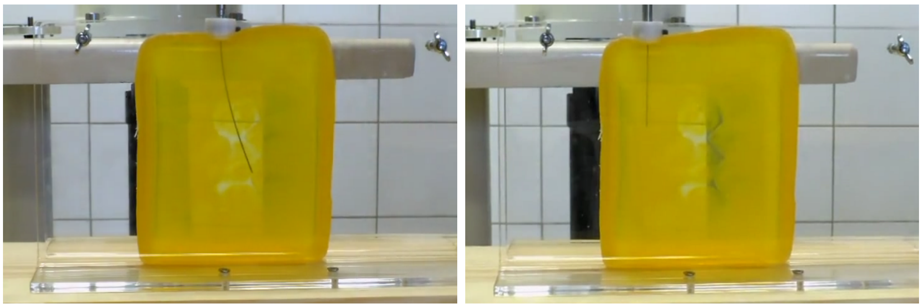


Figure 5.7: Manual selection of the insertion task through the user interface.



(a) DC = 0%, $\kappa = \kappa_{max}$ (b) DC = 100%, $\kappa = 0$

Figure 5.8: Needle trajectory when inserted into plastisol phantom with different duty-cycle references.

Hence, every experiment has an initialization step in which the planning task is defined: first, the operator selects the workspace region, obstacles positions and the needle initial configuration with mouse clicks on the camera images; then, the needle reachability region is plotted to help the operator define a reachable target from that given q_{init}. Finally p_{goal} is also selected with the mouse.

The obstacles may be used not only to indicate anatomical structures to be avoided, but also to obstruct risky paths according to the surgeons expertise. After the initialization, the path planning calculates the desired needle trajectory and the insertion procedure begins automatically.

5.3 RESULTS AND DISCUSSION

For evaluating the experimental setup and validating the proposed strategy for robot-assisted needle steering, we performed two sets of experiments. The first experiments correspond to open-loop needle insertions with the main objective of evaluating of the effects of the phantom material, needle geometry and duty-cycled rotation on the tip trajectory. In this trials, the adaptive motion planning module was suppressed, and the insertion movements were performed by the steering control using constant pre-defined DC references.

The second set of experiments correspond to closed-loop trials, with both planning and control modules working together. This experiments were performed with the objective of validating the proposed system, and verifying its robustness to uncertainties such as errors in the initial tip configuration and incorrect estimation of the natural curvature parameter.

5.3.1 *Open-loop insertions*

5.3.1.1 *Relation between duty-cycle and curvature*

Initially, we used the plastisol phantom and the larger needle to investigate the relation between duty-cycled rotation and needle curvature. For this, we performed single arc insertions with different DC references, and we expected to obtain a linear function between the resultant curvature and applied duty-cycle, as described in Chapter 2.

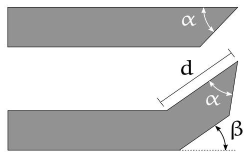


Figure 5.9: Simple beveled needles are defined by the angle tip α, while pre-bent beveled needles are also characterized by the bent angle β, and bend length d.

Table 5.1: Curvature radius obtained for different combinations of needle and tissue.

Phantom	Diameter	Tip	Parameters	r
Plastisol	0.8 mm	beveled	$\alpha = 16°$	25 cm
	0.508 mm	beveled	$\alpha = 20°$	15.5 cm
	0.8 mm	pre-bent	$\alpha = 16°$, $\beta = 30°$, $d = 4$ mm	10.7 cm
Agarose	0.508 mm	beveled	$\alpha = 20°$	10.5 cm
	0.508 mm	pre-bent	$\alpha = 20°$, $\beta = 40°$, $d = 3$ mm	6 cm

Indeed, the obtained results confirmed the relation given by equation (2.5): when no rotation is applied, the needle presented its maximum curvature (Fig. 5.8b), and as we raised the duty-cycle value, the needle curvature decreases linearly, approaching a straight trajectory in the limit when $DC = 100\%$ (Fig. 5.8b).

However, the combination of phantom and needle adopted in this experiment provided very low steerability, with a minimum curvature radius of approximately 25 cm. This curvature is not sufficient for evaluating the complete system with the path planning in closed-loop, since the system actuation could easily saturate, severely restricting the trajectories possibilities.

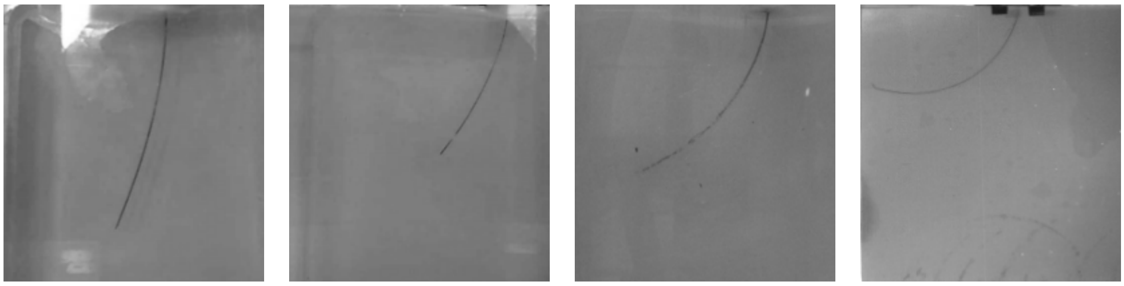
5.3.1.2 *Relation between needle-tissue combination and curvature*

To overcome the steerability limitations observed in our first choice of needle and tissue, we investigated the use of other combinations. Hence, we performed a series of single arc insertions with 0% duty-cycle, and used the identification procedure described in Subsection 5.2.1 to estimate the natural curvature for different needle-tissue sets.

Two different wire diameters were tested with two types of needle asymmetry: beveled and pre-bent tips. Beveled needles have a simple angled tip, while pre-bent needles have a bent section in the vicinity of the tip, which is also beveled (see Fig. 5.9). When manufacturing the needle prototypes, we tried to use the same α, β and d parameters, and the small variations obtained are due to the manual nature of the process. Table 5.1 presents the resultant curvature radius for the different combinations, while the insertion arcs are illustrated in Fig. 5.10.

The experiments allowed us to observe the effects of using different phantom materials and tip geometry in the insertion. For instance, we have observed that bigger diameters result in stiffer needles, which bend less. Consequently, as the needle diameter increases, the curvature decreases.

Also, the plastisol phantom presented high friction, making it very hard to insert the needle after a certain length. The force needed to continue inserting could get so high at



(a) $r = 25$ cm (b) $r = 15.5$ cm (c) $r = 10.7$ cm (d) $r = 6$ cm

Figure 5.10: Curvatures obtained for different combinations of phantom and needle geometry: *(a)* plastisol and beveled Ø0.8 mm needle, *(b)* plastisol and pre-bent Ø0.8 mm needle, *(c)* agarose and beveled Ø0.508 mm needle, *(d)* agarose and pre-bent Ø0.508 mm needle.

deeper insertions, that the needle would bend inside the telescopic tube before entering the tissue any further. This effect was specially evident when making insertions with the smaller diameter needle, which suffered permanent deformation before being inserted its full length into plastisol. Finally, the use of a pre-bent tip resulted in increased curvatures because they present larger asymmetry at the tip when compared to a simple beveled needle.

This observations motivated us to adopt the combination of agarose phantom and pre-bent Ø0.508 mm needle (which presented higher steerability) in the closed-loop experiments presented in the sequence.

5.3.2 *Closed-loop insertions*

5.3.2.1 *Validation results*

To evaluate the performance of the complete robot-assisted steering system, we carried out *in vitro* trials, this time in closed-loop, with the adaptive motion planning and the needle tracking integrated into the robotic system. We performed some automatic insertions with different scenarios to adjust the tracking threshold parameters, the number of simultaneous RRT trees, and the maximum number of samples in order to assure that the computational time needed by the adaptive motion planning to update the current needle configuration and desired path were smaller than a duty-cycle period.

At this point, we noticed that the effects of torsional stiffness had shown to be noticeable during some insertions. Moreover, we observed in some procedures that the needle was deviating from the desired 2D plane although the convergence to the target was always achieved in the camera image projections. In a severe case, the needle would even hit the acrylic support, loosing actuation. The needle tracking algorithm also failed to detect the needle tip if it was not close enough to the tissue surface, condition in which q_{tip} was no longer updated (see Fig 5.11). Hence, for evaluating the system, we only considered as valid trials those where the needle did not deviate considerably from the tissue phantom plane. Otherwise, the insertion was interrupted, the needle and phantom repositioned, and the trial restarted.

A sequence of four validation trials is presented in Fig. 5.12. In all of them, the needle was able to reach the target within an accuracy of 2 mm, which is acceptable in

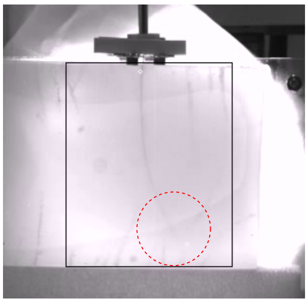
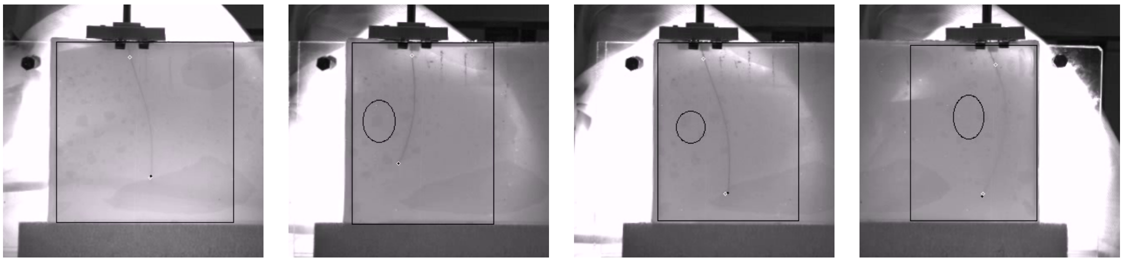


Figure 5.11: Loss of tracking after the needle deflected back in the direction of the acrylic support. The tip is no longer distinguishable inside the dashed region.



(a) Trial #1 (b) Trial #2 (c) Trial #3 (d) Trial #4

Figure 5.12: Final needle configuration for each validation scenario: *(a)* no obstacle, *(b)* target close to obstacle, *(c)* entry point in the right, and *(d)* entry point in the left.

most needle placement applications. The average tip placement error (mean ± standard deviation) is shown in Table 5.2, together with the RMS errors for the insertion control primitives. The good performance of these validation trials indicated the potential of the implemented closed-loop system, and motivated further tests to verify its robustness to modeling errors and other uncertainties of the insertion procedure.

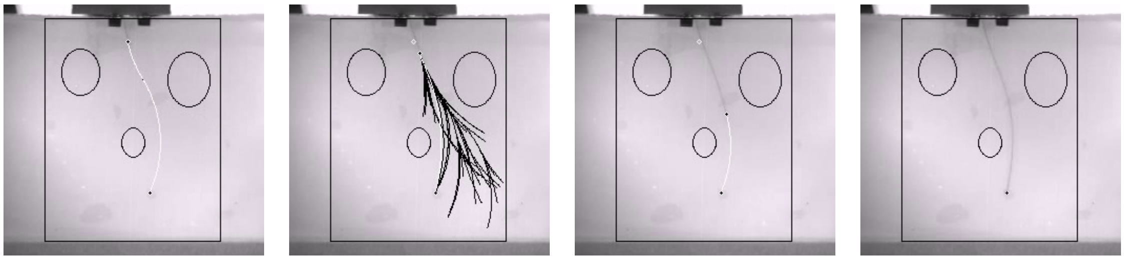
5.3.2.2 *Robustness results*

After the system validation, we tested its robustness by considering a κ_{max} approximately 9% smaller than the value we had previously identified offline. We also introduced an initial configuration error of approximately 7 mm and 38° to the value of q_{init}.

An image sequence of the performed needle insertion procedure is shown in Fig. 5.13, where we can observe the evolution of the insertion and how the desired trajectory

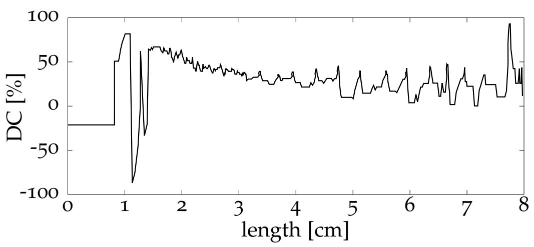
Table 5.2: Errors for *in vitro* validation trials.

Number of trials	4
Error to target (mm)	1.46 ± 0.75
β RMS error (deg)	0.0301 ± 0.0023
v RMS error (mm/s)	0.0886 ± 0.0116
x RMS error (mm)	0.0888 ± 0.0096
y RMS error (mm)	0.0021 ± 0.0003

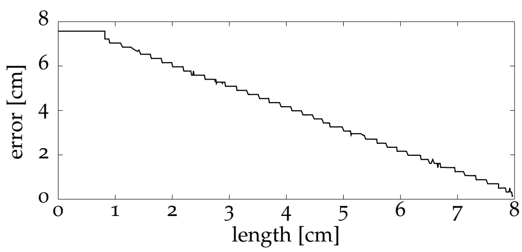


(a) $t = 0$ s (b) $t = 12$ s (c) $t = 63$ s (d) $t = 131$ s

Figure 5.13: Sequence of images from an insertion experiment. The black rectangle represents the workspace defined by the operator and the circles are considered to be obstacles. The white curve is the desired path given by the adaptive motion planning. The real needle can be seen as a shadow through the transparent phantom. *(a)* Initial planned path with induced error of approximately 7 mm and 38° in the initial configuration q_{init}. *(b)* A new RRT is built because the desired path was no longer feasible. *(c)* Progress of the insertion. The desired path is systematically recalculated from current q_{tip}. *(d)* Needle reaches the target with error of 1.43 mm.



(a) Duty-cycle reference.



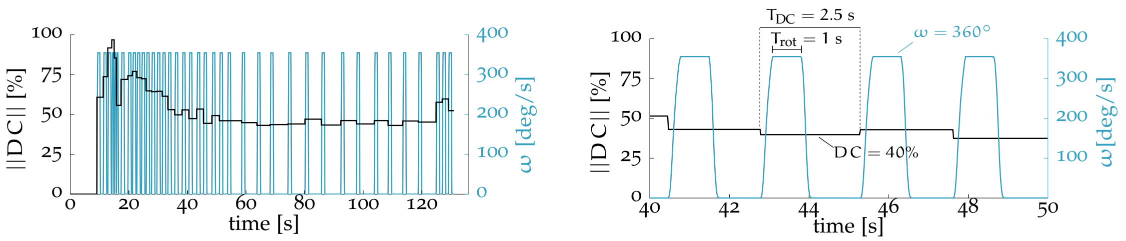
(b) Tip error to target.

Figure 5.14: DC reference and tip error, both parametrized in insertion length. The system starts operating in closed-loop after the needle was detected by the tracking system at 0.8 cm of insertion. Observe that the apparent discretization in *(b)* results from the delay in updating q_{tip}, caused by the needle tracking algorithm.

changes through the experiment, according to the current q_{tip} information. For this trial, the total needle insertion length was approximately 8 cm with final error of 1.43 mm to the desired point, which is compatible with the expected accuracy and with resolution of the camera used in the experiments. Fig. 5.14 shows both the decrease of the tip positioning error, and the duty-cycle reference that resulted in the needle trajectory shown in Fig. 5.13d. Note that negative DC reference values in Fig. 5.14 indicate a negative arc curvature. Every time there is a signal inversion in the DC reference, the needle bevel orientation is inverted by the duty-cycle control during the INIT state.

In Fig. 5.15 we can observe the ω signal applied by the steering control module. As expected, larger values of the duty-cycle reference (black) correspond to bigger proportions of time during which the angular velocity (blue) is different from zero. Also, observe that DC is kept constant for the duration of a cycle, and that, although $T_{rot} = 1$ s is constant, the cycle period T_{DC} varies according to changes in T_{ins}.

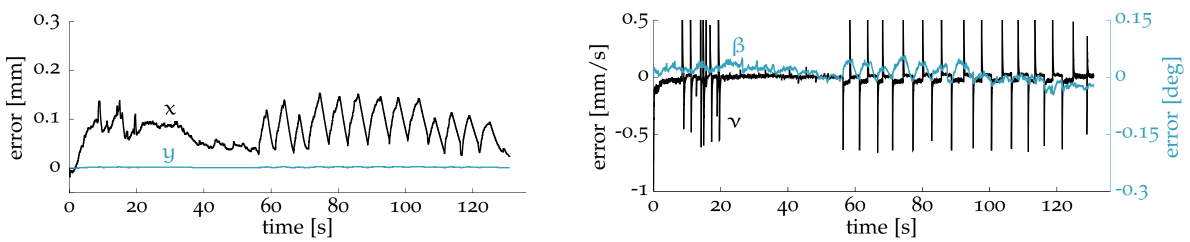
This experiment showed that the systematic feedback of q_{tip} provided by the needle tracking, combined with the trajectory intraoperative update, was able to compensate for



(a) Duty-cycled rotation velocity ω corresponding to the DC reference signal.

(b) Zoom showing the obtained duty-cycle period for a reference of DC = 40%.

Figure 5.15: Duty-cycle control results: DC reference (black) and correspondent ω output (blue). Observe that the discretization in the duty-cycle values results from the fact that DC is kept constant for the duration of a cycle.



(a) Control error for x and y.

(b) Control error for angle β and insertion velocity v.

Figure 5.16: Insertion control errors.

the induced errors and other possible system uncertainties like modeling errors, incomplete transmission of the robot movement to the needle tip, and etc.

Similarly to the performance evaluation from the previous chapter, the insertion control module presented good precision, with position errors for x and y of less then 0.2 mm, β error lower then 0.15 degree, and v errors in the order of 0.5 mm/s, as shown in Fig. 5.16. The small picks in the velocity errors represent changes in the v reference that are quickly compensated by the controller action.

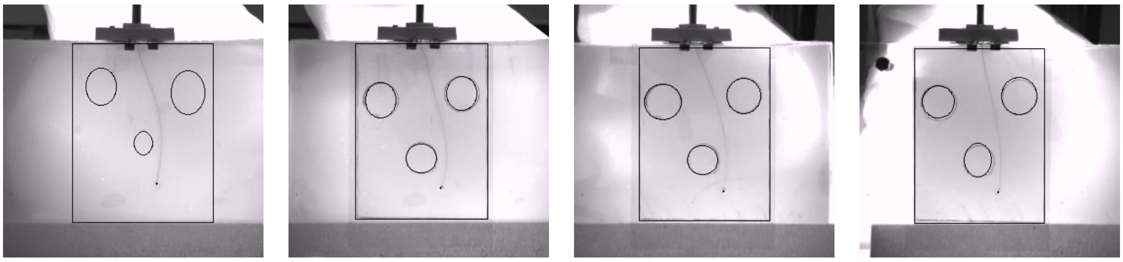
In order to perform a more consistent analysis of the obtained results, it is useful to perform many trials with similar conditions. To allow this, we have used a transparent film with the desired insertion environment printed on it. Every time the phantom was slid sideways for a new experiment, the same conditions could be repeated by translating the transparent film.

Thus, the same robustness experiment was repeated three more times with similar results. Table 5.3 shows the obtained values (mean ± standard deviation) for the final error to target and the RMS errors for the insertion control variables, while pictures of the complete insertions are shown in Fig. 5.17.

To conclude the robustness analysis, we performed two more insertions in a double-layered agarose phantom. This phantom consists of two parts of gelatin with in different concentrations, being the first layer equal to the previous used phantoms, and the second layer stiffer than the first. Consequently, the material property is not homogeneous, and the natural curvature changes as the needle crosses from one layer to the another. Using the offline identification procedure from Subsection 5.2.1, we estimated a minimum radius

Table 5.3: Errors for *in vitro* robustness trials with induced q_{init} and κ_{max} errors.

Number of trials	4
Error to target (mm)	1.34 ± 0.86
β RMS error (deg)	0.0350 ± 0.0088
ν RMS error (mm/s)	0.1305 ± 0.0365
x RMS error (mm)	0.0018 ± 0.0004
y RMS error (mm)	0.0889 ± 0.0093



(a) Trial #1 (b) Trial #2 (c) Trial #3 (d) Trial #4

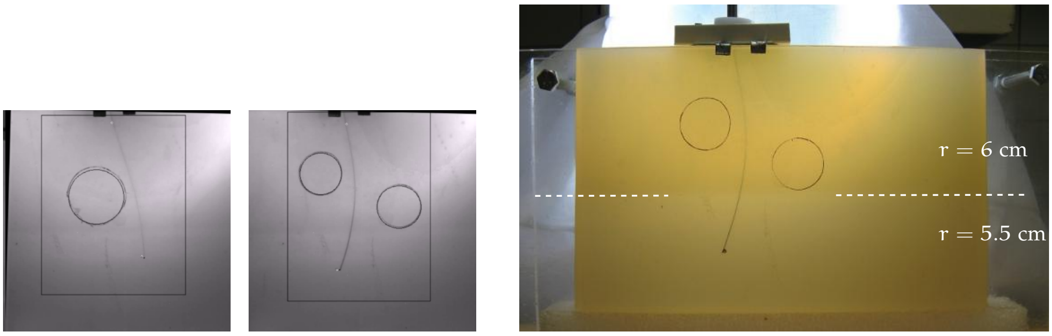
Figure 5.17: Final needle configuration for each robustness trial.

of curvature of $r = 5.5$ cm for the second layer, while the first layer is known to have $r = 6$ cm.

Similarly to the previous robustness trials, for both insertions performed, the needle was successfully positioned at the target with acceptable accuracy, although for the doubled-layered phantom, the final errors were slightly bigger (1.57 mm and 1.62 mm for trials 1 and 2, respectively). The two performed insertions and the two-layer phantom are shown in Fig. 5.18.

The presented *in vitro* experiments confirmed the feasibility of a robot-assisted needle steering system with intraoperative replanning strategy. They also indicated important aspects to be taken in consideration as the research development progress and the application moves from *in vitro* to *ex-* and *in vivo* experiments. For instance, medical real-time imaging is essential for guiding the needle in real animal tissue. The use of US images is especially attractive because it is safe, affordable and provides information related to tissue properties, target displacement and tool position (Abolhassani et al., 2007). Thanks to these advantages, and considering the context of this work within the USComp project, we consider it as the probable imaging modality for future developments, although in the presented *in vitro* tests we have used images from a standard CMOS camera.

Another important aspect to be considered is the identification of κ_{max}. The proposed approach is capable of compensating for underestimated values of this parameter as confirmed by the experiments, but if the parameter is overestimated, the actuation might saturate and still be unable to achieve the desired curve. Since real soft tissue is naturally non-homogeneous, the implementation of online curvature estimation should be considered.



(a) Trial #1 (b) Trial #2 (c) The dashed line shows the limit between layers.

Figure 5.18: Robustness test in non-homogeneous tissue. The obtained final positioning errors are *(a)* 1.57 mm, and *(b)* 1.62 mm. *(c)* Doubled-layered agarose phantom used for the last two robustness tests.

Also, dynamic workspaces were not tested, although the planning algorithm is compatible to changes in the scene. In the future, an additional module for active contour tracking in ultrasound images (e.g., (Li et al., 2011)) is foreseen to be combined with the needle tracking.

To conclude, the procedure for calibrating the needle insertion plane is not only unpractical, but also infeasible from a clinical point of view, since it requires multiple extra insertion before the actual procedure, going against the philosophy of minimal invasive interventions. Alternatives to give feedback information of the needle 3D configuration, such as sensors embedded in the needle tip, or 3D medical imaging, should be considered for the next development stages. Similarly, these solutions might be used to compensate torsional stiffness effects, as discussed in more detail in Chapter 6.

5.4 CONCLUSION

5.4.1 *Summary*

A robotically controlled needle steering system is normally comprised of a combination of physical systems and computational algorithms. This chapter explains how the computational modules described previously are combined together to form a robot-assisted closed-loop system with imaging feedback. More specifically, we present how the planning and control modules communicate and exchange information between themselves in order to define the manipulator behavior that will steer the needle according to a planned trajectory, while avoiding the predetermined obstacles and target.

We also introduce the experimental setup used to validate the system architecture, with a description of the calibration and initialization steps used for preparing the experimental trials. Lastly, the *in vitro* results are presented and discussed.

5.4.2 *Contributions*

The main contribution of this chapter is the proposal of a system architecture that combines path planning, feedback information, and control modules to form a robotically actuated needle steering system. The planning and control modules are contributions of this thesis and have been previously presented in Chapters 3 and 4, while the feedback information was provided by standard cameras that should be replaced by ultrasound or another kind of medical imaging system in future developments.

In addition, *in vitro* experiments validated our intraoperative replanning method and showed that even under disturbances, the needle was able to reach the target with satisfactory precision while avoiding obstacles. The results also indicated important aspects to be taken into consideration for future developments, such as torsional stiffness compensation, and online parameter estimation. Despite of these remarks, the results with *in vitro* tests were encouraging and presented our intraoperative replanning approach as a strong candidate for use in automatic needle steering solutions.

6 CONCLUSIONS AND PERSPECTIVES

6.1 CONCLUDING REMARKS

Needle steering can significantly improve the effectiveness of already existing medical interventions and also expand the applicability of needle-based techniques with the creation of new medical procedures that explore curved trajectories to access targets located deep into soft tissue and previously considered out of reach with straight-line trajectories. This thesis has focused on the development of a robot-assisted system for needle steering. More specifically, we have proposed a closed-loop method for automatic insertion of beveled needles using a manipulator arm and imaging feedback.

It consists in a motion planning module that uses imaging feedback from the current workspace and needle tip configurations to constantly adjust the planned path, and a steering control module, which controls the manipulator movements in order to perform a purely vertical insertion with duty-cycled rotation of the needle, resulting in trajectories with adjustable curvature.

The main contributions of this thesis include: (1) the development of an application-specific path planner that uses duty-cycled rotation and explicit geometry to produce paths with high success rate and fast calculation, (2) a replanning algorithm that uses feedback information to update the needle trajectory intraoperatively, resulting in a closed-loop control strategy, and (3) the proposal of a needle steering system that uses a commercially available manipulator arm, profiting from the already existing infrastructure to obtain a lower cost, more versatile and compact system.

In vitro experimental trials validated the proposed system, and showed that our systematic replanning strategy made the insertion procedure more robust to system uncertainties such as misplacement of the needle entry point, errors in parameter identification and modeling approximations. However, our implementation lacks 3D feedback of the tip configuration and consequently, only errors projected to a 2D plane could be corrected by the replanning. For validation purposes only, this was not a relevant issue since the tissue phantom is semi-transparent and allowed us to visualize the needle 2D configurations even when the effects of torsional stiffness caused needle to deviate from the insertion plane.

On the other side, in future clinical trials with 2D ultrasound, plane deviation becomes a critical problem since we may lose observation when the needle diverges from the medical imaging plane. A possible solution, is to incorporate a 2D stabilization controller (e.g., Swensen & Cowan, 2012; Kallem & Cowan, 2009) to our system. Another alternative, is to replace the planning algorithm by its 3D version, presented in Chapter 3, and fit the workspace limits to the 2D ultrasound slice thickness. In this case, the needle 3D configuration could be estimated from the applied inputs and bearing angle (Swensen, 2011), or measured by sensors attached to the needle tip, such as magnetic or Radio-Frequency Identification (RFID) based tracking solutions (by NDI Inc., and Amedo GmbH, respectively). One should also consider the use of 3D medical imaging (e.g., 3D ultrasound). Besides providing 3D feedback of the needle tip configuration, they also support three-dimensional workspaces which can result in a greater number of potential clinical applications.

As discussed in Chapter 3, convergence cannot be assured to our system like in any RRT-based solution. However, we have observed from simulations and tests that, thanks to the planner high success rate, the needle is very likely to reach the target if the system is not forced to work too close to saturation, that is, constantly inserting the needle with its maximum natural curvature.

For the presented experiments, the workspace was considered static during the whole procedure. However, the algorithms developed for the adaptive motion planning module are compatible with dynamic workspaces, which are expected during *in vivo* trials due to tissue swelling, respiratory motion, and other physiological effects. In the future, an additional module for active contour tracking in ultrasound images is foreseen to be combined with the needle tracking, providing feedback not only on the needle tip, but also on the workspace current configuration.

Some preliminary simulations have already suggested that the intraoperative replanning can successfully compensate for physiological movement if they involve small magnitude displacements at moderate velocities, and the tissue-needle combination presents good steerability. Bigger magnitude or higher frequency changes in the workspace, such as heart beating, would probably need to incorporate some kind of predictive model to the replanning strategy in order to estimate the motion of the tissue/obstacles, and consequently, avoid dead-end situations and saturation of the duty-cycled actuation.

Hence, needle steerability is an important issue to be considered when using beveled needle steering in a particular clinical application, specially when the workspace is very small, or too cluttered with obstacles to avoid. Although in our *in vitro* tests we obtained only 6 cm for the minimum radius of curvature, in the literature, this parameter has already achieved 1.5 cm in artificial tissue, and 3.4 cm in liver tissue (Majewicz et al., 2010). However, to reach targets in delicate regions of small organs, the curvatures obtained so far might not be enough. As mentioned in Chapter 1, the combination of beveled needles with other steering methods is an option already being considered for the achievement of higher steerability (Reed et al., 2011).

Another important aspect to be considered is the identification of the natural curvature. In this thesis, for experimental validation purposes, this parameter was experimentally determined with an offline minimization. However, using the same procedure to estimate κ_{max} in a clinical procedure would require extra needle insertions, contradicting the philosophy of minimally invasive interventions. Also, real soft tissue presents inhomogeneity, causing changes in the natural curvature as the needle is inserted. To improve robustness, online estimation of this parameter is a desirable feature to be implemented.

6.2 NEEDLE STEERING CHALLENGES

Needle steering presents enormous potential for improving the accuracy and applicability of percutaneous procedures, and recent research results have already provided great advances. However, a number of problems still remain open before needle steering can be moved on to clinical practice.

For instance, as observed by Reed et al. (2011), when a needle encounters a membrane with a shallow angle between them, instead of puncturing it and continuing with the planned path, the needle is likely to slide along the membrane, changing its expected trajectory. Since membrane layers are an inherent part of soft tissue and organs, overcoming such difficulty is important for the improvement of needle steering methods.

Also, as observed in Chapter 5 when we used a plastic-based tissue phantom, friction along the length of the needle can make it buckle. The friction forces increase linearly with the insertion length, and even more when the needle curves inside the tissue. Alternatively, we changed the phantom material and decreased the needle diameter to reduce the observed friction effects, but nevertheless, small buckling still occurs inside the support sheath, and it can affect the angle of insertion at the base, and the perception of already inserted length. This issue has motivated the investigation of other insertion methods, more specifically, a discrete solution which was inspired by the manual procedure performed by physicians and consists in alternating between two motions: grasp-push and release-retreat. However, this is still an ongoing work, and just one of many possible alternatives to deal with the buckling problem.

Another issue in needle steering involves undesired tissue damage. The insertion of a needle implies cutting through tissue, which normally causes bleeding and swelling. Minor tissue damage is intrinsic to any needle insertion, but needle steering has the potential of increased damage resultant from lateral tissue slicing. In longer insertions, the needle causes pressure against the tissue along its length. If too much force is exerted on the needle shaft, the needle can slice through tissue. The same happens when the needle is inserted through harder tissues or membranes: if the needle can not surpass the barrier, it buckles inside the tissue causing lateral slicing.

Although comparative studies (Majewicz et al., 2012) concluded that the insertion of beveled and pre-bent needles do not seem to damage tissue more than straight needles, there is still no consensus whether their rotation can increase tissue damage or not, which is an important concern for clinical use of the duty-cycle technique. Podder et al. (2005) investigated the effects of beveled needle rotation in plastic phantoms and did not observe significant enhance of damage. On the other side, Reed et al. (2011) observed that simultaneous insertion and rotation of a pre-bent needle results in a corkscrew cutting pattern in the tissue, even though the needle itself follows an apparent straight trajectory. However, it is still unclear if such cutting pattern causes significantly more damage to tissue. Other authors have also raised concerns about soft tissue injure. Although they conclude this should not prohibit the use of rotation as long as its effects are considered in risk analysis (Badaan et al., 2011), future research should investigate safe velocity limits, and new needle tip designs to minimize potential damage.

APPENDIX

A

QUATERNIONS AND DUAL QUATERNIONS

This appendix presents general concepts regarding quaternions and dual quaternions and their use in rigid motion representation. Some properties and mathematical definitions are presented, but for more detailed information we recommend the reading of textbooks (McCarthy, 1990; Kuipers, 1999). The notation used here and throughout the thesis was based in (Adorno, 2011), although most part of it is common in the current literature.

A.1 MATHEMATICAL BACKGROUND

A.1.1 *Quaternions*

Quaternions were introduced by Hamilton (1844) and can be seen as an extension of complex numbers, with three imaginary components $\hat{\imath}, \hat{\jmath}, \hat{k}$ defined as:

$$\hat{\imath} = \hat{\jmath}\hat{k} = -\hat{k}\hat{\jmath}, \qquad \hat{\jmath} = \hat{\imath}\hat{k} = -\hat{\imath}\hat{k}, \qquad \hat{k} = \hat{\imath}\hat{\jmath} = -\hat{\jmath}\hat{\imath}, \quad \text{and} \quad \hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = -1. \tag{A.1}$$

The quaternion $\mathbf{h}$ is defined as

$$\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}, \qquad h_i \in \mathbb{R}, i = 1, \ldots, 4 \tag{A.2}$$

The conjugate of a quaternion is given by

$$\mathbf{h}^* = h_1 - h_2\hat{\imath} - h_3\hat{\jmath} - h_4\hat{k}. \tag{A.3}$$

Let $\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$, and $\mathbf{h}' = h'_1 + h'_2\hat{\imath} + h'_3\hat{\jmath} + h'_4\hat{k}$ be two quaternions. The addition and multiplication of two quaternions are defined as:

$$\mathbf{h} \pm \mathbf{h}' = h_1 \pm h'_1 + \hat{\imath}\left(h_2 \pm h'_2\right) + \hat{\jmath}\left(h_3 \pm h'_3\right) + \hat{k}\left(h_4 \pm h'_4\right), \tag{A.4}$$

$$\begin{aligned}\mathbf{h}\mathbf{h}' &= \left(h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}\right)\left(h'_1 + h'_2\hat{\imath} + h'_3\hat{\jmath} + h'_4\hat{k}\right) \\ &= \left(h_1h'_1 - h_2h'_2 - h_3h'_3 - h_4h'_4\right) \\ &\quad + \hat{\imath}\left(h_1h'_2 + h_2h'_1 + h_3h'_4 - h_4h'_3\right) \\ &\quad + \hat{\jmath}\left(h_1h'_3 - h_2h'_4 + h_3h'_1 + h_4h'_2\right) \\ &\quad + \hat{k}\left(h_1h'_4 + h_2h'_3 - h_3h'_2 + h_4h'_1\right),\end{aligned} \tag{A.5}$$

whereas the norm of a quaternion $\mathbf{h}$ is given by

$$\|\mathbf{h}\| = \sqrt{\mathbf{h}\mathbf{h}^*} = \sqrt{\mathbf{h}^*\mathbf{h}}. \tag{A.6}$$

The set of quaternions $\mathbb{H}$ (for Hamilton) forms a group under the quaternion multiplication. They present associative and distributive properties, but are non-commutative.

To perform operations between matrix and quaternions, one must consider a previous parametrization of the quaternion into a vector. This is done with the vec operator, which takes each coefficient of the quaternion and stacks them in a vector; that is, it performs the one-by-one mapping vec : $\mathbb{H} \to \mathbb{R}^4$. Let $\mathbf{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$, then

$$\operatorname{vec}\mathbf{h} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \end{bmatrix}^T. \tag{A.7}$$

Commutativity in dual quaternion multiplication can be achieved with the use of Hamilton operators $\overset{+}{\mathbf{H}}$ and $\overset{-}{\mathbf{H}}$ (Akyar, 2008; Chou, 1992). For $\mathbf{h}, \mathbf{h}' \in \mathbb{H}$,

$$\operatorname{vec}\left(\mathbf{h}\mathbf{h}'\right) = \overset{+}{\mathbf{H}}(\mathbf{h}) \operatorname{vec}\mathbf{h}' = \overset{-}{\mathbf{H}}\left(\mathbf{h}'\right) \operatorname{vec}\mathbf{h}, \tag{A.8}$$

with $\overset{+}{\mathbf{H}}$ and $\overset{-}{\mathbf{H}}$ given by

$$\overset{+}{\mathbf{H}}(\mathbf{h}) = \begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & -h_4 & h_3 \\ h_3 & h_4 & h_1 & -h_2 \\ h_4 & -h_3 & h_2 & h_1 \end{bmatrix}, \qquad \overset{-}{\mathbf{H}}\left(\mathbf{h}'\right) = \begin{bmatrix} h'_1 & -h'_2 & -h'_3 & -h'_4 \\ h'_2 & h'_1 & h'_4 & -h'_3 \\ h'_3 & -h'_4 & h'_1 & h'_2 \\ h'_4 & h'_3 & -h'_2 & h'_1 \end{bmatrix}.$$

A.1.2 *Dual numbers*

Dual numbers were introduced by Clifford (1873), who proposed the dual unit ε to create a new algebra. In this algebra, ε is nilpotent and defined as:

$$\varepsilon \neq 0, \qquad \varepsilon^2 = 0. \tag{A.9}$$

A dual number $\underline{a} = a + \varepsilon a'$, consists of two parts; namely, the primary and the dual parts, typically composed of the same type of elements. The primary and the dual parts can be extracted using the operators $\mathcal{P}(\underline{a})$ and $\mathcal{D}(\underline{a})$, respectively (Adorno, 2011). Hence

$$\underline{a} = \mathcal{P}(\underline{a}) + \varepsilon\,\mathcal{D}(\underline{a}). \tag{A.10}$$

In Clifford's algebra, the sum and multiplication operations take into account the dual unit ε. Let $\underline{a}$ and $\underline{b}$ be two dual numbers, then

$$\underline{a} \pm \underline{b} = \mathcal{P}(\underline{a}) \pm \mathcal{P}(\underline{b}) + \varepsilon\left(\mathcal{D}(\underline{a}) \pm \mathcal{D}(\underline{b})\right), \tag{A.11}$$

$$\begin{aligned} \underline{ab} &= \left(\mathcal{P}(\underline{a}) + \varepsilon\,\mathcal{D}(\underline{a})\right)\left(\mathcal{P}(\underline{b}) + \varepsilon\,\mathcal{D}(\underline{b})\right) \\ &= \mathcal{P}(\underline{a})\,\mathcal{P}(\underline{b}) + \varepsilon\left(\mathcal{P}(\underline{a})\,\mathcal{D}(\underline{b}) + \mathcal{D}(\underline{a})\,\mathcal{P}(\underline{b})\right). \end{aligned} \tag{A.12}$$

A.1.3 *Dual Quaternions*

Dual quaternions , also introduced by Clifford in 1873, are dual numbers in which the primary and dual parts are quaternions. Thus, a dual quaternion $\underline{\mathbf{h}}$ is defined as

$$\begin{aligned} \underline{\mathbf{h}} &= \mathcal{P}(\underline{\mathbf{h}}) + \varepsilon\,\mathcal{D}(\underline{\mathbf{h}}), \\ &= \left(h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}\right) + \varepsilon\left(h_5 + h_6\hat{\imath} + h_7\hat{\jmath} + h_8\hat{k}\right) \end{aligned} \tag{A.13}$$

with $\mathcal{P}(\underline{\mathbf{h}}), \mathcal{D}(\underline{\mathbf{h}}) \in \mathbb{H}$. The set of dual quaternions is denoted by $\mathcal{H}$.

The conjugate of the dual quaternion $\underline{\mathbf{h}}$ is

$$\underline{\mathbf{h}}^* = \mathcal{P}(\underline{\mathbf{h}})^* + \varepsilon\,\mathcal{D}(\underline{\mathbf{h}})^* . \tag{A.14}$$

The multiplication of dual quaternions follows the same rules as for dual numbers, but respecting the quaternion operations. Analogously, the vec operator can also be extended to dual quaternions, performing the one-by-one mapping vec : $\mathcal{H} \to \mathbb{R}^8$, so that

$$\operatorname{vec} \underline{\mathbf{h}} = \begin{bmatrix} h_1 & \ldots & h_8 \end{bmatrix}^T . \tag{A.15}$$

A.2 RIGID BODY MOTION

A.2.1 *Rotations represented by quaternions*

Unit norm quaternions can be used to represent rotations. For instance, a rotation ϕ around the unit norm axis $\mathbf{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$, is given by a quaternion $\mathbf{r} \in \mathbb{H}$, such that

$$\mathbf{r} = \cos(\phi/2) + \sin(\phi/2)\mathbf{n}. \tag{A.16}$$

Let the quaternion $\mathbf{r}_0^0 = 1$ represent the initial orientation of a given reference frame $\mathcal{O}_0$. After N rotations, the final orientation is given by

$$\mathbf{r}_0^N = \mathbf{r}_0^1 \ldots \mathbf{r}_{N-1}^N, \tag{A.17}$$

where the subscript and superscript represent the original and final frames, respectively.

A.2.2 *Points and translations represented by quaternions*

A point or translation can be represented by a pure quaternion $\mathbf{p}$; that is, quaternions where the real part is equal to zero:

$$\mathbf{p} = p_x\hat{\imath} + p_y\hat{\jmath} + p_z\hat{k}. \tag{A.18}$$

Let $\mathbf{p}_0$ be a point with respect to the coordinate frame $\mathcal{O}_0$. If frame $\mathcal{O}_1$ is obtained rotating $\mathcal{O}_0$ by a quaternion $\mathbf{r}_0^1$, then point $\mathbf{p}_0$ with respect to $\mathcal{O}_1$ is given by

$$\mathbf{p}_1 = \mathbf{r}_0^{1*}\mathbf{p}_0\mathbf{r}_0^1. \tag{A.19}$$

Let $\mathbf{p}_0$ be a point with respect to $\mathcal{O}_0$, and consider that this point is rotated by a quaternion $\mathbf{r}_0$. Thus, the point's new position in $\mathcal{O}_0$ is given by

$$\mathbf{p}_0' = \mathbf{r}_0\mathbf{p}_0\mathbf{r}_0^*. \tag{A.20}$$

A.2.3 *Rigid motions represented by dual quaternions*

A rigid motion can be completely represented by the dual quaternion

$$\underline{\mathbf{q}} = \mathbf{r} + \varepsilon\frac{1}{2}\mathbf{p}\mathbf{r}, \tag{A.21}$$

where $\mathcal{P}\left(\underline{\mathbf{q}}\right) = \mathbf{r}$ is a unit norm quaternions representing a rotation, and $\mathcal{D}\left(\underline{\mathbf{q}}\right) = \frac{1}{2}\mathbf{p}\mathbf{r}$ is an unbounded norm quaternion indirectly representing a translation. The translations is given by the quaternion $\mathbf{p}$, which can be recovered with some quaternion operations:

$$\mathbf{p} = 2\,\mathcal{D}\left(\underline{\mathbf{q}}\right)\mathcal{P}\left(\underline{\mathbf{q}}\right)^{*}. \tag{A.22}$$

A sequence of rigid motions can be represented by a sequence of dual quaternion multiplications. Let $\underline{\mathbf{q}}_0^1$ be the dual quaternion that represents the transformation from frame $\mathcal{O}_0$ to $\mathcal{O}_1$, and $\underline{\mathbf{q}}_1^2$ to represent the transformation from frame $\mathcal{O}_1$ to $\mathcal{O}_2$. The resultant motion with respect to $\mathcal{O}_0$ is $\underline{\mathbf{q}}_0^2 = \underline{\mathbf{q}}_0^1\underline{\mathbf{q}}_1^2$. Extending this reasoning to N transformations, we have that

$$\underline{\mathbf{q}}_0^N = \underline{\mathbf{q}}_0^1 \cdots \underline{\mathbf{q}}_{N-1}^N, \tag{A.23}$$

where the subscript and superscript represent the original and final frames, respectively.

B MANIPULATOR KINEMATICS IN THE DUAL TASK SPACE

The configuration of a manipulator end-effector is given by a composition of rigid motions between the robot links. To facilitate the description of each robot link, a coordinate frame is normally affixed to it. The frames can be arbitrarily chosen as long as they are attached to the link they refer to. Nevertheless, a commonly used approach for selecting the coordinate frames is the Denavit-Hartenberg convention, which establishes a general and systematic method for the automatic modeling of serial robots.

This appendix gives a brief description of the standard D-H convention, and shows how it can be used to obtain the Forward and the Differential Kinematic Models of a serial manipulator using the dual quaternion notation, as proposed by Adorno (2011). It also presents the standard D-H parameters for the Adept Viper s650 robot, which is the platform we have used for validating our proposal through *in vitro* experiments.

B.1 THE DUAL TASK SPACE

From a mechanical point of view, a standard serial manipulator can be represented as a kinematic chain of serial links connected by means of joints. Each joint has one degree of freedom, either translational (prismatic joint) or rotational (revolute joint). For a manipulator with N joints numbered from 1 to N, there are $N+1$ links, numbered from 0 to N. Link 0 is the base of the manipulator, normally fixed, and link N bears the end-effector, as depicted in Fig. B.1. We consider the location of joint i to be fixed with respect to link $i-1$, so that when joint i is actuated, link i moves.

To each manipulator joint i, we associate a joint variable denoted θ_i. In the case of a revolute joint, θ_i is the rotation angle, whereas in a prismatic joint, it corresponds to the joint displacement. The vector of all joint variables $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \dots & \theta_N \end{bmatrix}^T \in \mathbb{R}^N$ is defined

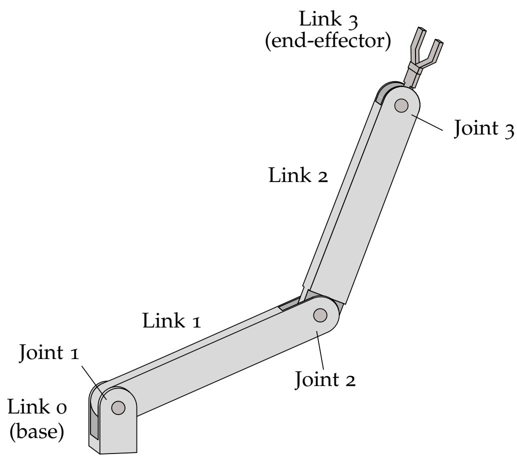


Figure B.1: Links and joints of a planar elbow manipulator with 3 DOF. The first link corresponds to the base, which is fixed, whereas the last one carries the end-effector.

in a space typically called *joint space*. On the other hand, the *task space* denotes the space in which the end-effector configuration is defined. One may consider many different representations for the task space. A common choice is to use Cartesian coordinates to specify the position in $\mathbb{R}^3$ and the rotation group SO^3 for orientation, resulting in a task space equal to $\mathbb{R}^3 \times SO^3$.

Another alternative is to use the set of dual quaternions $\mathcal{H}$, that has proved to be a useful representation to describe rigid motions (Yang, 1963; Bottema & Roth, 1979; McCarthy, 1990). Similarly to homogeneous transformation matrices, when using the dual quaternion notation, a single mathematical object is used to describe the complete rigid motion, and a sequence of rigid motions is represented by a sequence of multiplications. But instead of using a matrix, dual quaternions simultaneously describe positions and orientations in a single vector by means of the vec operator.

The difference between the dual quaternion representation and the homogeneous transformation matrices is not sensitive in terms of the number of calculations for either the kinematic model or the Jacobian (Adorno, 2011). Still, dual quaternions provide some attractive properties not found in homogeneous transformation matrices, such as the commutativity of the Hamilton operators which facilitate the calculations, and the possibility of using the same set of variables for both the forward kinematics and the robot control without any intermediate parametrization. The unified representation brings closer modeling and control, making the resultant theory more compact.

B.2 THE STANDARD DENAVIT-HARTENBERG CONVENTION

B.2.1 *Coordinate frames assignment*

For a given robot manipulator, the choice of the various coordinate frames is not unique, even when constrained to the requirements presented below. However, the final result will be always the same, regardless of the assignment of intermediate frames.

At first, for all links, we assign the z_i-axis to be the axis of actuation for joint $i+1$. Once we have established the z-axes for all the links, we establish the base frame. The origin of $\mathcal{O}_0$ can be chosen at any point on z_0. Then, x_0 and y_0 are chosen in any convenient manner so that we have a right-handed frame.

After frame 0 has been defined, we begin an iterative process starting from $i = 1$, in which we define frame i using frame $i-1$. In order to establish frame i, it is necessary to consider three possibilities: (I) the axes z_{i-1} and z_i are not coplanar, (II) the axes z_{i-1} and z_i intersect, (III) the axes z_{i-1} and z_i are parallel.

(I) z_{i-1} and z_i are not coplanar: it exists a unique line segment perpendicular to both z_{i-1} and z_i such that it connects both lines and it has minimum length. The line containing this common normal defines x_i, and the point where this line intersects z_i is the origin $\mathcal{O}_i$;

(II) z_{i-1} is parallel to z_i : there are infinitely common normals between them. To simplify, a usual method is to choose the normal that passes through $\mathcal{O}_{i-1}$ as the x_i-axis. Then, $\mathcal{O}_i$ is defined as the point where x_i intersects z_i;

(III) z_{i-1} intersects z_i: the axis x_i is chosen normal to the plane formed by z_{i-1} and z_i, with arbitrary positive direction. The natural choice for the origin of $\mathcal{O}_i$ is at the intersection of both z_{i-1} and z_i.

With both x_i and z_i axes defined, the y_i-axis is chosen so that a right-handed frame is obtained. At last, the end-effector frame is assigned normally with $\mathcal{O}_N$ located at the center of the tool. The x_N and y_N axes can be chosen freely to compose a right-handed frame.

In summary, the reference frame for a link i is laid out as follows:

z_i-axis	in the direction of the joint $i+1$ axis;
x_i-axis	parallel to the common normal to axis z_i and z_{i-1};
y_i-axis	follows the x_i- and z_i-axis to obtain a right-handed frame.

B.2.2 *Forward kinematic model*

Kinematics is a branch of mechanics which treats the phenomenon of motion without regard to its cause. In a kinematic model, there is no reference to mass or force; the concern is only with relative positions and their changes so that trajectories can be abstracted into purely mathematical expressions. The kinematics of manipulators involves the study of the geometric properties of the motion, and in particular how the various links move with respect to one another and with time.

The FKM model defines the configuration of the end-effector in the task space, given the configuration of the robot links in the joint space. The importance of the FKM in manipulator trajectory control is clear: it is in task space coordinates that specific tasks we desire are usually expressed; whereas it is in the joint coordinates that the actuator movements are described.

When the dual quaternion notation is used, the FKM, is given as

$$\underline{\mathbf{q}}_E = \underline{\mathbf{f}}(\theta), \tag{B.1}$$

where $\underline{\mathbf{q}}_E$ is the dual quaternion representing the pose of the robot's end-effector, θ is the vector of joints variables, and $\underline{\mathbf{f}} : \mathbb{R}^N \to \mathcal{H}$.

A link may be considered as a rigid body and to describe its location, a coordinate is affixed to it. Thus, each intermediate link is represented by an intermediate transformation, relating the configuration of link i with respect to the previous one in the chain. Consequently, we can write B.1 as a sequence of dual quaternion multiplications:

$$\underline{\mathbf{q}}_E = \underline{\mathbf{q}}_0^N = \underline{\mathbf{q}}_0^1 \underline{\mathbf{q}}_1^2 \cdots \underline{\mathbf{q}}_{N-1}^N, \tag{B.2}$$

with $\underline{\mathbf{q}}_i^j$ being the dual quaternion that represents the rigid motion from link i to link j.

In the Denavit-Hartenberg convention, the rigid motion that describes the pose of a link with respect to the previous one is represented by a product of four basic transformations: first, a rotation θ is performed around the z-axis, followed by a translation d along the z-axis; then a translation a is performed along the x-axis, followed by a rotation α around the x-axis. The D-H convention in dual quaternion space is straightforward and was introduced by Adorno (2011). It consists in multiplying the four dual quaternions that correspond to the D-H motion sequence, resulting in the dual quaternion:

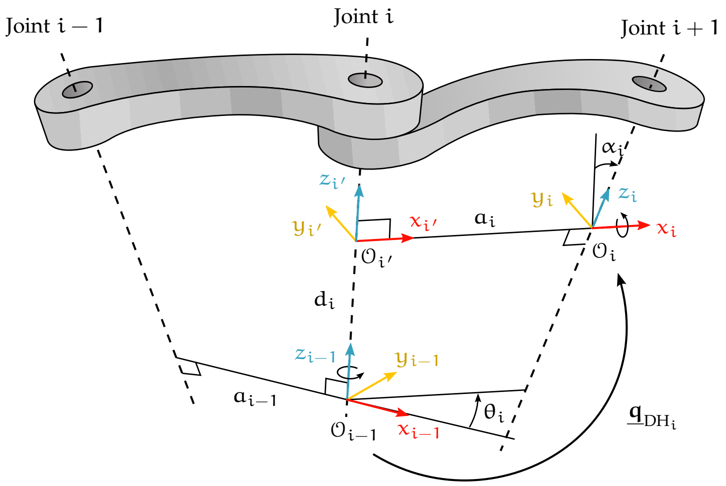


Figure B.2: Sequence of transformations for the standard D-H convention (adapted from (Spong et al., 2006)).

$$\underline{q}_{\mathrm{DH}} = \mathbf{r}_{z,\theta}\underline{\mathbf{p}}_{z,d}\underline{\mathbf{p}}_{x,a}\mathbf{r}_{x,\alpha}, \tag{B.3}$$

where $\mathbf{r}_{z,\theta}$ represents a pure rotation of θ around the z-axis (analogously for $\mathbf{r}_{x,\alpha}$), and $\underline{\mathbf{p}}_{z,d}$ represents a pure translation of d along the z-axis (analogously for $\underline{\mathbf{p}}_{x,a}$). Observe that $\underline{\mathbf{p}}_{z,d} = 1 + \varepsilon\,(d/2)\,\hat{k}$ and $\underline{\mathbf{p}}_{x,a} == 1 + \varepsilon\,(a/2)\,\hat{\imath}$. Expanding the quaternion multiplications leads to

$$\begin{aligned}
\mathcal{P}\left(\underline{q}_{\mathrm{DH}}\right) &= h_1 + \hat{\imath}h_2 + \hat{\jmath}h_3 + \hat{k}h_4 \\
\mathcal{D}\left(\underline{q}_{\mathrm{DH}}\right) &= -\frac{(dh_4 + ah_2)}{2} + \hat{\imath}\frac{(ah_1 - dh_3)}{2} \\
&\quad + \hat{\jmath}\frac{(dh_2 - +ah_4)}{2} + \hat{k}\frac{(dh_1 - ah_3)}{2}.
\end{aligned} \tag{B.4}$$

with

$$h_1 = \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\alpha}{2}\right), \qquad h_2 = \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\alpha}{2}\right),$$
$$h_3 = \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\alpha}{2}\right), \qquad h_4 = \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\alpha}{2}\right).$$

So, for each joint i, there is an associated $\underline{q}_{\mathrm{DH}_i}$ transformation, with the four quantities θ_i, d_i, α_i, and a_i denominated joint angle, link offset, link length, and link twist, respectively, as illustrated in Fig. B.2.

As a consequence, we can rewrite (B.2) so that the end-effector configuration is obtained from the standard D-H parameters as:

$$\underline{q}_E = \underline{q}_{\mathrm{DH}_1}\underline{q}_{\mathrm{DH}_2}\cdots\underline{q}_{\mathrm{DH}_N}, \tag{B.5}$$

where $\underline{q}_{\mathrm{DH}_i}$ is the transformation from link $i-1$ to link i.

B.2.3 *Analytical Jacobian*

While the FKM establishes the relationship between the joint variables and the end-effector configuration, the *analytical Jacobian* is a matrix that maps the joint velocities into the derivatives of the end-effector configuration vector.

Hence, considering that the end-effector is represented in the dual task space, the dual quaternion analytical Jacobian $\mathbf{J}_{\underline{q}_E}$ is the $8 \times N$ matrix that satisfies

$$\operatorname{vec} \dot{\underline{q}}_E = \mathbf{J}_{\underline{q}_E} \dot{\theta}, \tag{B.6}$$

with $\dot{\underline{q}}_E$ being the first derivative of the dual quaternion parameters that represent the end-effector configuration, and $\dot{\theta}$ is the joint velocity vector. It can be obtained by differentiating (B.2) as demonstrated in (Adorno, 2011) and shown as follows.

We have that the FKM is obtained from the D-H parameters by the expression

$$\underline{q}_E = \underline{q}_{DH_1} \underline{q}_{DH_2} \cdots \underline{q}_{DH_N}, \tag{B.7}$$

where $\underline{q}_{DH_i}$ is the transformation from link $i-1$ to link i given by B.4. By taking the first derivative of (B.7) we obtain

$$\dot{\underline{q}}_E = \sum_{i=0}^{N-1} \underline{q}_0^i \dot{\underline{q}}_i^{i+1} \underline{q}_{i+1}^N. \tag{B.8}$$

If we define $\underline{x}_i^{i+1}$ to be the dual quaternion that satisfies $\dot{\underline{q}}_i^{i+1} = (1/2)\, \underline{x}_i^{i+1} \underline{q}_i^{i+1}$, then B.8 becomes

$$\begin{aligned}
\dot{\underline{q}}_E &= \sum_{i=0}^{N-1} \underline{q}_0^i \left(\frac{1}{2} \underline{x}_i^{i+1} \underline{q}_i^{i+1} \right) \underline{q}_{i+1}^N \\
&= \frac{1}{2} \sum_{i=0}^{N-1} \underline{q}_0^i \underline{x}_i^{i+1} \underline{q}_i^N \\
&= \frac{1}{2} \sum_{i=0}^{N-1} \underline{q}_0^i \underline{x}_i^{i+1} \left(\underline{q}_0^i \right)^* \underline{q}_E.
\end{aligned} \tag{B.9}$$

Using the D-H convention, we have that $\underline{q}_i^{i+1} = \underline{q}_{DH_{i+1}}$. By differentiating (B.4) we obtain

$$\begin{aligned}
\dot{\underline{q}}_i^{i+1} &= \frac{d}{dt} \underline{q}_i^{i+1} \\
&= \mathcal{P}\left(\dot{\underline{q}}_i^{i+1} \right) + \varepsilon \, \mathcal{D}\left(\dot{\underline{q}}_i^{i+1} \right), \\
\mathcal{P}\left(\dot{\underline{q}}_i^{i+1} \right) &= \frac{\dot{\theta}_{i+1}}{2} \left(-h_4 - \hat{\imath} h_3 + \hat{\jmath} h_2 + \hat{k} h_1 \right), \\
\mathcal{D}\left(\dot{\underline{q}}_i^{i+1} \right) &= \frac{\dot{\theta}_{i+1}}{4} \left((-d h_1 + a h_3) + \hat{\imath}(-d h_2 - a h_4) \right. \\
&= \left. + \hat{\jmath}(-d h_3 + a h_1) + \hat{k}(-d h_4 - a h_2) \right).
\end{aligned} \tag{B.10}$$

As a consequence of (B.4) and (B.10),

$$\begin{aligned}\underline{x}_i^{i+1} &= 2\underline{\dot{q}}_i^{i+1}\left(\underline{q}_i^{i+1}\right)^* \\ &= \hat{k}\dot{\theta}_{i+1}.\end{aligned} \tag{B.11}$$

Thus, (B.9) becomes

$$\begin{aligned}\underline{\dot{q}}_E &= \frac{1}{2}\sum_{i=0}^{N-1}\underline{q}_0^i\hat{k}\left(\underline{q}_0^i\right)^*\underline{q}_E\dot{\theta}_{i+1} \\ &= \sum_{i=0}^{N-1}\underline{z}_i\underline{q}_E\dot{\theta}_{i+1},\end{aligned} \tag{B.12}$$

with $\underline{z}_i = \frac{1}{2}\underline{q}_0^i\hat{k}\left(\underline{q}_0^i\right)^*$.

Resolving explicitly,

$$\begin{aligned}\mathcal{P}\left(\underline{z}_i\right) &= \hat{\imath}\left(h_{i_1}h_{i_3}+h_{i_2}h_{i_4}\right) \\ &\quad+\hat{\jmath}\left(h_{i_3}h_{i_4}-h_{i_1}h_{i_2}\right) \\ &\quad+\hat{k}\left(\frac{h_{i_1}^2-h_{i_2}^2-h_{i_3}^2+h_{i_4}^2}{2}\right), \\ \mathcal{D}\left(\underline{z}_i\right) &= \hat{\imath}\left(h_{i_2}h_{i_8}+h_{i_6}h_{i_4}+h_{i_1}h_{i_7}+h_{i_5}h_{i_3}\right) \\ &\quad+\hat{\jmath}\left(h_{i_3}h_{i_8}+h_{i_7}h_{i_4}-h_{i_1}h_{i_6}-h_{i_5}h_{i_2}\right) \\ &\quad+\hat{k}\left(h_{i_4}h_{i_8}-h_{i_3}h_{i_7}-h_{i_2}h_{i_6}+h_{i_1}h_{i_5}\right),\end{aligned} \tag{B.13}$$

where $h_{i_1}\ldots h_{i_8}$are the coefficients of $\underline{q}_0^i$.

Recall that $\operatorname{vec}\underline{\dot{q}}_E = \mathbf{J}_{\underline{q}_E}\dot{\theta}$. By inspection of (B.12), we have that the columns of the analytical Jacobian $\mathbf{J}_{\underline{q}_E} = [\mathbf{j}_1 \ldots \mathbf{j}_N]$ are given by

$$\mathbf{j}_{i+1} = \operatorname{vec}\left(\underline{z}_i\underline{q}_E\right), \qquad i = 0,\ldots,N-1 \tag{B.14}$$

where $\underline{z}_i = \mathcal{P}\left(\underline{z}_i\right) + \varepsilon\,\mathcal{D}\left(\underline{z}_i\right)$, with its coefficients given by (B.13).

The dual quaternion analytical Jacobian matrix can be decomposed as

$$\mathbf{J}_{\underline{q}_E} \triangleq \begin{bmatrix}\mathbf{J}_{\mathcal{P}(\underline{q}_E)} \\ \mathbf{J}_{\mathcal{D}(\underline{q}_E)}\end{bmatrix}, \tag{B.15}$$

such that

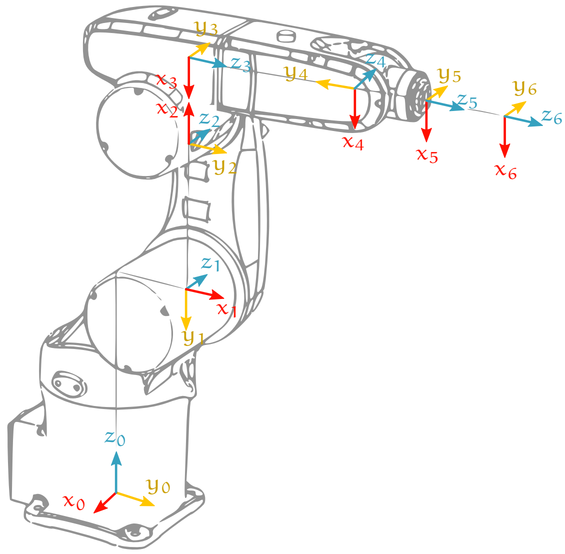
$$\begin{aligned}\operatorname{vec}\left(\mathcal{P}\left(\underline{\dot{q}}_E\right)\right) &= \mathbf{J}_{\mathcal{P}(\underline{q}_E)}\dot{\theta}, \\ \operatorname{vec}\left(\mathcal{D}\left(\underline{\dot{q}}_E\right)\right) &= \mathbf{J}_{\mathcal{D}(\underline{q}_E)}\dot{\theta}.\end{aligned} \tag{B.16}$$

B.3 ADEPT VIPER S650 D-H PARAMETERS

Table B.1: Standard D-H parameter for the Adept Viper s650.

i	θ_i [rad]	d_i [m]	a_i [m]	α_i [rad]
1	$-\frac{\pi}{2}$	0.075	0.335	0
2	0	0.270	0	$-\frac{\pi}{2}$
3	$\frac{\pi}{2}$	−0.090	0	π
4	$-\frac{\pi}{2}$	0	0.295	0
5	$\frac{\pi}{2}$	0	0	0
6	0	0	0.080*	0

∗0.150 if the force sensor is attached



C IMAGE PROCESSING PROCEDURES

In this appendix we present three image processing steps that should be performed in order to calibrate the camera system before being used in a feedback control loop. They consist in removing image distortion, rectifying the image plane to coincide to that of the tissue phantom, and identifying the image m/pixel ratio.

C.1 INTRINSIC PARAMETERS CALIBRATION

To compensate the effect of lens imperfections, a camera calibration should be done to identify its internal parameters, which are latter used to undistort the images. Hence, the first procedure consists in obtaining the camera intrinsic parameters, that is, its focal length $f = [\ f_x \quad f_y\ ]^T$, principal point $C = [\ C_x \quad C_y\ ]^T$, skew coefficient α, and radial and tangential distortions $k = [\ k_1 \quad k_2 \quad k_3 \quad k_4 \quad k_5\ ]^T$. Each pixel (x, y) in the original image camera corresponds to a pixel (u, v) in the corrected and rectified image according to the mapping

$$x' = \frac{(u - C_x)}{f_x}, \quad y' = \frac{(v - C_y)}{f_y}, \quad r^2 = u^2 + v^2,$$

$$\begin{aligned} x'' &= x'(1 + k_1 r^2 + k_2 r^4 + k_5 r^6) + 2k_3 x'y' + k_4(r^2 + 2x'^2), \\ y'' &= y'(1 + k_1 r^2 + k_2 r^4 + k_5 r^6) + 2k_4 x'y' + 2(r^2 + 2y'^2), \end{aligned}$$

$$\begin{aligned} x &= f_x x'' + \alpha y'' + C_x, \\ y &= f_y y'' + C_y. \end{aligned} \tag{C.1}$$

We used the Matlab Calibration Toolbox (Bouguet, 2003) to estimate such parameters from a set of images of a known chessboard pattern taken at different positions and perspectives (see Fig. C.1). The estimation computes the final intrinsic calibration parameters by minimizing the reprojection error of the known pixels through the gradient descent method.

The intrinsic parameters are used to transform the images to compensate for lens distortion using the inverse mapping of (C.1) as illustrated in Fig. C.2a-b.

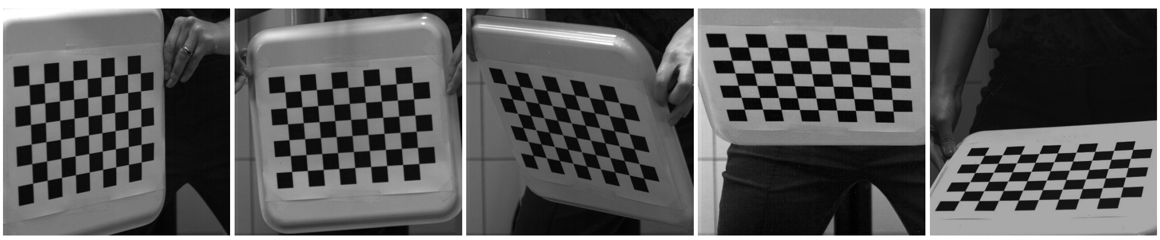
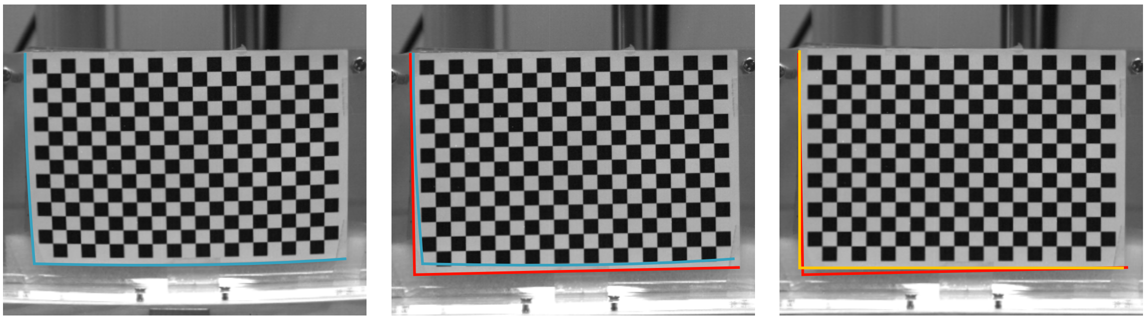


Figure C.1: Examples of images used to identify the camera intrinsic parameters.



(a) Original image (b) Undistorted (c) Rectified

Figure C.2: Calibration and rectification steps of the camera images. The colored segments correspond to the bottom-right edges of the chessboard pattern and are plotted to better illustrate how the applied transformations modify the final processed image: pre-processed image (blue), distortion compensation (red), and planar rectification (yellow).

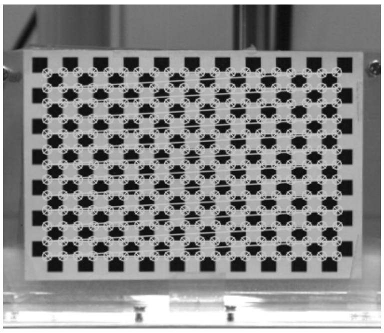


Figure C.3: Corners extracted from rectified image to calculate the images resolution [m/pixels].

C.2 PLANAR RECTIFICATION

The planar rectification procedure is responsible for making the camera image plane correspond to the same plane of the tissue phantom and its acrylic support. For this, a chessboard pattern with squares of known size is temporarily fixed in the acrylic plane. The square corners are extracted and used to build a planar rectification homography matrix (Park & Park, 2010) which is stored and used to rectify the images during the experiments.

For this, we use a pattern with 260 corners in total, being 20 in the x-axis direction and 13 in the y-axis direction. In the camera images, we find all corners using the Shi-Tomasi corner detection algorithm, implemented with the OpenCV library (Bradski, 2000). Then, we select the four outer corners and use a RANSAC-based robust method to obtain the perspective transformation that aligns them in a perfect rectangle. This transformation is defined by the homography matrix which is estimated with a least-square minimization of the back projection error, followed by a refining step using the Levenberg-Marquardt method. As a result, in the rectified image, all the squares from the chessboard pattern are aligned with the image reference frame axes (see Fig. C.2c).

C.3 IDENTIFICATION OF THE IMAGE RESOLUTION

After the rectification procedure is performed, the same squared pattern is also used to obtain the image resolution [m/pixel]. Since in the rectified image the squares have equal sides with known size, we re-extract the corners and apply a least-square minimization to obtain the m/pixel ratio. The obtained resolution value is used by the adaptive motion planner to convert the insertion length information from meters to image pixels.

BIBLIOGRAPHY

Abolhassani, N., Patel, R., & Moallem, M. (2007). Needle insertion into soft tissue: A survey. *Medical Engineering & Physics*, 29, 413–431.

Adorno, B. (2012). DQ Robotics Toolbox.

Adorno, B. V. (2011). *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration*. PhD thesis, Université Montpellier 2.

Adorno, B. V., Fraisse, P., & Druon, S. (2010). Dual position control strategies using the cooperative dual task-space framework. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3955–3960).

Akyar, B. (2008). Dual Quaternions in Spatial Kinematics in an Algebraic Sense. *Turkish Journal of Mathematics*, 32(4), 373–391.

Alterovitz, R., Branicky, M., & Goldberg, K. (2006). Constant-Curvature Motion Planning Under Uncertainty with Applications in Image-Guided Medical Needle Steering. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics* (pp. 319–334).

Alterovitz, R., Branicky, M., & Goldberg, K. (2008). Motion Planning Under Uncertainty for Image-Guided Medical Needle Steering. *The International Journal of Robotics Research*, 27(11-12), 1361–1374.

Alterovitz, R. & Goldberg, K. (2007). The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty.

Alterovitz, R., Goldberg, K., & Okamura, A. (2005a). Planning for Steerable Bevel-tip Needle Insertion Through 2D Soft Tissue with Obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1640–1645).

Alterovitz, R., Lim, A., Goldberg, K., Chirikjian, G., & Okamura, A. (2005b). Steering Flexible Needles Under Markov Motion Uncertainty. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1570–1575).

Alterovitz, R., Pouliot, J., Taschereau, R., Hsu, I.-C. J., & Goldberg, K. (2003). Needle insertion and radioactive seed implantation in human tissues: simulation and sensitivity analysis. In *Proceedings of the IEEE International Conference on Robotics and Automation* Taipei, Taiwan.

Badaan, S., Petrisor, D., Kim, C., Mozer, P., Mazilu, D., Gruionu, L., Patriciu, A., Cleary, K., & Stoianovici, D. (2011). Does needle rotation improve lesion targeting? *Internarional Journal of Medical Robotics and Computer Assisted Surgery*.

Barraquand, J. & Latombe, J. C. (1991). Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, (pp. 628–649).

Bassan, H., Hayes, T., Patel, R. V., & Moallem, M. (2007). A Novel Manipulator for 3D Ultrasound Guided Percutaneous Needle Insertion. *Computer Engineering*, (April), 10–14.

Benabid, A. L., Cinquin, P., Lavalle, S. M., Bas, J. F. L., Demongeot, J., & Rougemont, J. (1987). Computer-driven robot for stereotactic surgery connected to ct scan and magnetic resonance imaging: technologial design and preliminary results. *Applied Neurophysiology*, 50, 153–154.

Bernardes, M. C., Adorno, B. V., Poignet, P., Zemiti, N., & Borges, G. A. (2011). Adaptive path planning for steerable needles using duty-cycling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2545–2550).

Bottema, O. & Roth, B. (1979). *Theoretical kinematics*. North-Holland Pub. Co.

Bouguet, J. Y. (2003). Camera calibration toolbox for Matlab.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Chang, H. & Li, T. Y. (1995). Assembly Maintainability Study with Motion Planning. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1012–1019).

Chentanez, N., Alterovitz, R., Ritchie, D., Cho, L., Hauser, K., Goldberg, K., Shewchuk, J. R., & O'Brien, J. F. (2009). Interactive simulation of surgical needle insertion and steering. In *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques*, volume 28 (pp.ĩ).

Chou, J. C. K. (1992). Quaternion Kinematic and Dynamic Differential Equations. *IEEE Transactions on Robotics*, 8(1), 53–64.

Clifford, W. K. (1873). Preliminary sketch of biquaternions. In *Proceedings of the London Mathematical Society*, volume 4 (pp. 381–395).

Collins English Dictionary (2012). *Complete & Unabridged*. HarperCollins Publishers, 10th edition.

Corke, P. (1996). A robotics toolbox for matlab. *IEEE Robotics & Automation Magazine*, 3(1), 24 –32.

Corral, G., Ibáñez, L., & Nguyen, C. (2004). Robot control by fluoroscopic guidance for minimally invasive spine procedures. In *Proceedings of the International Congress and Exhibition on Computer-Assisted Radiology and Surgery*, number Cars (pp. 1–6).

Cowan, N., Goldberg, K., Chirikjian, G., Fichtinger, G., Alterovitz, R., & Reed, K. B. (2011). Robotic Needle Steering: Design, Modeling, Planning, and Image Guidance. In J. Rosen, B. Hannaford, & R. M. Satava (Eds.), *Surgical Robotics* (pp. 557–582). Springer US.

Dehghan, E., Wen, X., Zahiri-Azar, R., Marchal, M., & Salcudean, S. E. (2007). Modeling of needle-tissue interaction using ultrasound-based motion estimation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 709–16).

DiMaio, S. P., Fischer, G. S., Haker, S. J., Hata, N., Iordachita, I., Tempany, C. M., Kikinis, R., & Fichtinger, G. (2006). A System for MRI-guided Prostate Interventions. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 68–73).

DiMaio, S. P., Pieper, S., Chinzei, K., Fichtinger, G., & Kikinis, R. (2004). Robot assisted percutaneous intervention in open-MRI. In *Proceedings of the International Interventional MRI Symposium* (pp. 155).

DiMaio, S. P. & Salcudean, S. E. (2002). Simulated interactive needle insertion. In *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* (pp. 344–351).

DiMaio, S. P. & Salcudean, S. E. (2003a). Needle insertion modeling and simulation. *IEEE Transactions on Robotics and Automation*, 2, 2098–2105.

DiMaio, S. P. & Salcudean, S. E. (2003b). Needle steering and model-based trajectory planning. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 33–40).

DiMaio, S. P. & Salcudean, S. E. (2005a). Interactive Simulation of Needle Insertion Models. *IEEE Transactions on Biomedical Engineering*, 52(7), 1167–1179.

DiMaio, S. P. & Salcudean, S. E. (2005b). Needle steering and motion planning in soft tissues. *IEEE Transactions on Biomedical Engineering*, 52(6), 965–974.

Ding, J., Stoianovici, D., Petrisor, D., Mozer, P., Avila, R., Ibanez, L., Turner, W., Yankelvitz, D., Wilson, E., Banovac, F., & Cleary, K. (2008). Medical needle steering for lung biopsy: Experimental results in tissue phantoms using a robotic needle driver. In *Proceedings of the IEEE International Conference on Bioinformatics and Bioengineering* (pp. 1–5).

Dombre, E. & Khalil, W. (2007). *Modeling, performance analysis and control of robot manipulators*. ISTE Ltd.

Dorileo, E. A. G. (2011). *Needle Tip Location - Customized Approach*. Technical report, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, Montpellier, France.

Duindam, V., Alterovitz, R., Sastry, S., & Goldberg, K. (2008a). Screw-based motion planning for bevel-tip flexible needles in 3D environments with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2483–2488).

Duindam, V., Alterovitz, R., Sastry, S., & Goldberg, K. (2010). Three-dimensional Motion Planning Algorithms for Steerable Needles Using Inverse Kinematics. *The International Journal of Robotics Research*, 29(7), 789–800.

Duindam, V., Xu, J., Alterovitz, R., Sastry, S., & Goldberg, K. (2008b). 3D Motion Planning Algorithms for Steerable Needles Using Inverse Kinematics. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics* Guanajuato, Mexico.

Engh, J., Podnar, G., Khoo, S., & Riviere, C. (2006a). Flexible Needle Steering System for Percutaneous Access to Deep Zones of the Brain. In *Proceedings of the IEEE Annual Northeast Bioengineering Conference* (pp. 103–104).

Engh, J. A., Podnar, G., Kondziolka, D., & Riviere, C. N. (2006b). Toward effective needle steering in brain tissue. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1 (pp. 559–62).

Fichtinger, G., Kazanzides, P., Okamura, A. M., Hager, G. D., Whitcomb, L. L., & Taylor, R. H. (2008). Surgical and Interventional Robotics: Part II. *IEEE Robotics & Automation Magazine*, (September), 94–102.

Fischer, H. & Kutter, S. (2004). ROBITOM II: Robot for biopsy and therapy of the mamma. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 47 Suppl 1 (pp. 2–5).

Fu, Y., Liu, H., & Huang, W. (2009). Steerable catheters in minimally invasive vascular surgery. *Internarional Journal of Medical Robotics and Computer Assisted Surgery*, (5), 381–391.

Gerovich, O., Marayong, P., & Okamura, A. M. (2004). The effect of visual and haptic feedback on computer-assisted needle insertion. *Journal of the International Society for Computer Aided Surgery*, 9(6), 243–9.

Glozman, D. & Shoham, M. (2004). Flexible Needle Steering and Optimal Trajectory Planning for Percutaneous Therapies. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 137–144). Rennes, France.

Glozman, D. & Shoham, M. (2007). Image-Guided Robotic Flexible Needle Steering. *IEEE Transactions on Robotics*, 23(3), 459–467.

Hamilton, W. R. (1844). On a new species of imaginary quantities connected with a theory of quaternions. In *Proceedings of the Royal Irish Academy*, volume 2 (pp. 424–434).

Hauser, K., Alterovitz, R., Chentanez, N., Okamura, A., & Goldberg, K. (2009). Feedback Control for Steering Needles Through 3D Deformable Tissue Using Helical Paths. In *Proceedings of the Robotics: Science and Systems Conference*.

Heibel, T. H., Glocker, B., Paragios, N., & Navab, N. (2010). Needle tracking through higher-order MRF optimization. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (pp. 624–627).

Hong, J., Dohi, T., Hashizume, M., Konishi, K., & Hata, N. (2004). An ultrasound-driven needle-insertion robot for percutaneous cholecystostomy. *Physics in Medicine and Biology*, 49(3), 441–455.

Kallem, V. & Cowan, N. J. (2007). Image-guided control of flexible bevel-tip needles. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 3015–3020). Rome, Italy.

Kallem, V. & Cowan, N. J. (2009). Image Guidance of Flexible Tip-Steerable Needles. *IEEE Transactions on Robotics*, 25(1), 191–196.

Kataoka, H., Washio, T., Audette, M., & Mizuhara, K. (2001). A Model for Relations Between Needle Deflection, Force, and Thickness on Needle Penetration. In W. Niessen & M. Viergever (Eds.), *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 2208 of *Lecture Notes in Computer Science* (pp. 966–974).

Kataoka, H., Washio, T., & Chinzei, K. (2002). Measurement of the tip and friction force acting on a needle during penetration. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 2488 (pp. 216–223).

Kavraki, L. E., Svestka, P., Vestka, L. E. K. P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12, 566–580.

Kettenbach, J., Kronreif, G., & Figl, M. (2005). Robot-assisted biopsy using computed tomography-guidance: initial results from in vitro tests. *Investigative Radiology*, 40(4), 219–228.

Ko, S. Y. & Rodriguez y Baena, F. (2012). Trajectory following for a flexible probe with state/input constraints: An approach based on model predictive control. *Robotics and Autonomous Systems*, 60(4), 509–521.

Krieger, A., Susil, R. C., Ménard, C., Coleman, J. A., Fichtinger, G., Atalar, E., & Whitcomb, L. L. (2005). Design of a novel MRI compatible manipulator for image guided prostate interventions. *IEEE Transactions on Biomedical Engineering*, 52(2), 306–13.

Kuffner, J. J. & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 995–1001).

Kuipers, J. B. (1999). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, 1st edition.

Kwoh, Y., Hou, J., Jonckheere, E., & Hayati, S. (1988). A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35(2), 153–160.

Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers.

LaValle, S. M. (1998). Rapidly-Exploring Random Trees: A New Tool For Path Planning.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press.

LaValle, S. M. & Kuffner, J. J. (1999). Randomized kinodynamic planning. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 473–479).

Li, B. N., Nguyen, P. B., Ong, S. H., Qin, J., Yang, L. J., & Chui, C. K. (2009). Image Processing and Modeling for Active Needle Steering in Liver Surgery. In *Proceedings of the International Asia Conference on Informatics in Control, Automation and Robotics* (pp. 306–310).

Li, T., Krupa, A., & Collewet, C. (2011). A robust parametric active contour based on fourier descriptors. In *Proceedings of the International Conference on Image Processing* (pp. 1037 –1040).

Lobaton, E., Zhang, J., Patil, S., & Alterovitz, R. (2011). Planning Curvature-Constrained Paths to Multiple Goals Using Circle Sampling. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1463–1469).

Lozano-Perez, T. (1983). Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2), 108 –120.

Majewicz, A., Marra, S., van Vledder, M., Lin, M., Choti, M., Song, D., & Okamura, A. (2012). Behavior of tip-steerable needles in ex vivo and in vivo tissue. *IEEE Transactions on Biomedical Engineering*, 59(10), 2705 –2715.

Majewicz, A., Wedlick, T. R., Reed, K. B., & Okamura, A. M. (2010). Evaluation of Robotic Needle Steering in ex vivo Tissue. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2068–2073).

Mallapragada, V. & Sarkar, N. (2009). Robot-assisted real-time tumor manipulation for breast biopsy. *IEEE Transactions on Robotics*, 25(2), 316–324.

Mallapragada, V. G., Sarkar, N., & Podder, T. K. (2008). Autonomous coordination of imaging and tumor manipulation for robot assisted breast biopsy. In *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* (pp. 676–681).

Maurel, W. (1999). *3D modeling of the human upper limb including the biomechanics of joints, muscles and soft tissues*. PhD thesis, École Polythechnique Fédérale de Lausanne.

Maurin, B., Barbe, L., & Bayle, B. (2004). In vivo study of forces during needle insertions. In *Proceedings of the Medical Robotics, Navigation and Visualisation Scientific Workshop* (pp. 1–8).

McBeth, P. B., Louw, D. F., Rizun, P. R., & Sutherland, G. R. (2004). Robotics in neurosurgery. *The American Journal of Surgery*, 188(4), 68–75.

McCarthy, J. M. (1990). *An Introduction to Theoretical Kinematics*. MIT Press.

Minhas, D., Engh, J., Fenske, M., & Riviere, C. (2007). Modeling of needle steering via duty-cycled spinning. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2756–2759).

Misra, S., Ramesh, K. T., & Okamura, A. (2008a). Modeling of Tool-Tissue Interactions for Computer-Based Surgical Simulation : A Literature Review. *Presence*, 17(5), 463– 491.

Misra, S., Reed, K. B., Douglas, A. S., Ramesh, K. T., & Okamura, A. (2008b). Needle-tissue interaction forces for bevel-tip steerable needles. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 224–231).

Misra, S., Reed, K. B., Schafer, B. W., Ramesh, K. T., & Okamura, A. (2009). Observations and Models for Needle-Tissue Interactions. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2687–2692). Kobe, Japan.

Misra, S., Reed, K. B., Schafer, B. W., Ramesh, K. T., & Okamura, A. (2010). Mechanics of Flexible Needles Robotically Steered through Soft Tissue. *The International Journal of Robotics Research*, (pp. 1640–1660).

Mittal, S. & Deb, K. (2007). Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 3195–3202).

Neubach, Z. & Shoham, M. (2010). Ultrasound-Guided Robot for Flexible Needle Steering. *IEEE Transactions on Biomedical Engineering*, 57(4), 799–805.

Nienhuys, H.-W. & van Der Stappen, A. F. (2004). A Computational Technique for Interactive Needle Insertions in 3D Nonlinear Material. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2061–2067).

Okamura, A., Simone, C., & O'Leary, M. D. (2004). Force modeling for needle insertion into soft tissue. *IEEE Transactions on Biomedical Engineering*, 51(10), 1707–1716.

Okazawa, S. H., Ebrahimi, R., Chuang, J., Rohling, R. N., & Salcudean, S. E. (2006). Methods for segmenting curved needles in ultrasound images. *Medical Image Analysis*, 10(3), 330–42.

Okazawa, S. H., Ebrahimi, R., Chuang, J., Salcudean, S. E., & Rohling, R. N. (2005). Hand-held steerable needle device. *IEEE/ASME Transactions on Mechatronics*, 10(3), 285–296.

Park, S.-Y. & Park, G. G. (2010). Active calibration of camera-projector systems based on planar homography. In *Proceedings of the International Conference on Pattern Recognition* (pp. 320 –323).

Park, W., Kim, J., Zhou, Y., Cowan, N. J., Okamura, A., & Chirikjian, G. (2005). Diffusion-based motion planning for a nonholonomic flexible needle model. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 4611–4616).

Park, Y. & Elayaperumal, S. (2010). Real-Time Estimation of 3-D Needle Shape and Deflection for MRI-Guided Interventions. *IEEE/ASME Transactions on Mechatronics*, 15(6), 906–915.

Patil, S. & Alterovitz, R. (2010). Interactive Motion Planning for Steerable Needles in 3D Environments with Obstacles. In *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* (pp. 893–899).

Patil, S., van den Berg, J. P., & Alterovitz, R. (2011). Motion planning under uncertainty in highly deformable environments. In *Proceedings of the Robotics: Science and Systems Conference*.

Paul, S., Graves, C. M., Slocum, A., Gupta, R., & Walsh, C. J. (2012). Towards a Compact Robotically Steerable Thermal Ablation Probe. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 709–714).

Pettré, J., Laumond, J. P., & Siméon, T. (2003). A 2-stages locomotion planner for digital actors. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (pp. 258–264).

Pham, H.-L., Perdereau, V., Adorno, B. V., & Fraisse, P. (2010). Position and Orientation Control of Robot Manipulators Using Dual Quaternion Feedback. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Podder, T. K., Clark, D. P., Sherman, J., Fuller, D., Messing, E. M., Rubens, D. J., Strang, J. G., Zhang, Y. D., Ng, W. S., & Yu, Y. (2005). Effects of tip geometry of surgical needles: an assessment of force and deflection. In *Proceedings of the European Medical and Biological Engineering Conference*, volume 11 (pp. 5–8).

Reed, K. B., Kallem, V., Alterovitz, R., Goldberg, K., Okamura, A., & Cowan, N. J. (2008). Integrated Planning and Image-Guided Control for Planar Needle Steering. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 819–824).

Reed, K. B., Majewicz, A., Kallem, V., Alterovitz, R., Goldberg, K., Cowan, N. J., & Okamura, A. M. (2011). Robot-assisted needle steering. *IEEE Robotics & Automation Magazine*, 18(4), 35–46.

Reed, K. B., Okamura, A., & Cowan, N. J. (2009). Modeling and control of needles with torsional friction. *IEEE Transactions on Biomedical Engineering*, 56(12), 2905–16.

Ren, H., Vasilyev, N. V., & Dupont, P. E. (2011). Detection of curved robots using 3D ultrasound. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2083–2089).

Romano, J. M., Webster III, R. J., & Okamura, A. (2007). Teleoperation of Steerable Needles. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 934–939).

Ryu, S. C., Quek, Z. F., Renaud, P., Black, R. J., Daniel, B. L., & Cutkosky, M. R. (2012). An optical actuation system and curvature sensor for a MR-compatible active needle. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1589–1594).

Schwartz, J. T. & Sharir, M. (1983). On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4(3), 298–351.

Sears, P. & Dupont, P. (2006). A Steerable Needle Technology Using Curved Concentric Tubes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2850–2856). Beijing, China.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. Springer-Verlag, 2nd edition.

Siméon, T., Laumond, J. P., Geem, C. V., & Cortes, J. (2001). Computer aided motion: Move3d within molog. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1494–1499).

Simone, C. & Okamura, A. M. (2002). Modeling of needle insertion forces for robot-assisted percutaneous therapy. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2085–2091).

Smolen, J. & Patriciu, A. (2009). Model based stabilization of soft tissue targets in needle insertion procedures. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5084–5087).

Song, G. & Amato, N. M. (2001). Using motion planning to study protein folding pathways. In *Proceedings of the Annual International Conference on Computational Biology* (pp. 287–296).

Song, S.-E., Cho, N. B., Iordachita, I. I., Guion, P., Fichtinger, G., & Whitcomb, L. L. (2011). A Study of Needle Image Artifact Localization in Confirmation Imaging of MRI-guided Robotic Prostate Biopsy. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 4834–4839).

Spong, M., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control*. John Wiley & Sons.

Stoianovici, D., Cadeddu, J. A., Demaree, R. D., Basile, S. A., Taylor, R. H., Whitcomb, L. L., Sharpe Jr., W. N., & Kavoussi, L. R. (1997). A novel mechanical transmission applied to percutaneous renal access. In *Proceedings of the ASME Dynamic Systems and Control Division* (pp. 401–406).

Su, H., Zervas, M., Cole, G. a., Furlong, C., & Fischer, G. S. (2011). Real-time MRI-guided needle placement robot with integrated fiber optic force sensing. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1583–1588).

Swensen, J. P. (2011). *Torsional dynamics and rotational estimation of tip-steerable needles*. PhD thesis, The Johns Hopkins University.

Swensen, J. P. & Cowan, N. J. (2012). Torsional dynamics compensation enhances robotic control of tip-steerable needles. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1601–1606).

Torabi, M., Hauser, K., Alterovitz, R., Duindam, V., & Goldberg, K. (2009). Guiding Medical Needles Using Single-Point Tissue Manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 4254–4259). Kobe, Japan.

Torres, L. G. & Alterovitz, R. (2011). Motion Planning for Concentric Tube Robots Using Mechanics-Based Models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5153–5159).

Tran, D. & Rohling, R. N. (2010). Automatic Detection of Lumbar Anatomy in Ultrasound Images of Human Subjects. *IEEE Transactions on Biomedical Engineering*, (c), 1–9.

van den Berg, J. P., Patil, S., & Alterovitz, R. (2010). LQG-based planning, sensing, and control of steerable needles. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics* (pp. 373–389).

Van Geem, C., Simeon, T., Laumond, J. P., Bouchet, J. L., & Rit, J. F. (1999). Mobility analysis for feasibility studies in CAD models of industrial environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3 (pp. 1770–1775).

Vancamberg, L., Sahbani, A., Muller, S., & Morel, G. (2010). Needle Path Planning for Digital Breast Tomosynthesis Biopsy. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2062–2067).

Vancamberg, L., Sahbani, A., Muller, S., & Morel, G. (2011). Needle Path Planning for Digital Breast Tomosynthesis Biopsy Using a Heterogeneous Model. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 5749–5755).

Walsh, C. J., Franklin, J., Slocum, A. H., & Gupta, R. (2011). Design of a robotic tool for percutaneous instrument distal tip repositioning. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2097–100).

Webster III, R. J. & Jones, B. A. (2010). Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review. *The International Journal of Robotics Research*, 29(13), 1661–1683.

Webster III, R. J., Kim, J., Cowan J, N., Chirikjian, G., & Okamura, A. (2006a). Nonholonomic modeling of needle steering. *The International Journal of Robotics Research*, 25(5-6), 509–525.

Webster III, R. J., Memisevic, J., & Okamura, A. (2005). Design considerations for robotic needle steering. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 3588–3594).

Webster III, R. J., Okamura, A., & Cowan, N. J. (2006b). Toward Active Cannulas: Miniature Snake-Like Surgical Robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2857–2863).

Webster III, R. J. & Romano, J. M. (2009). Mechanics of precurved-tube continuum robots. *IEEE Transactions on Robotics*, 25(1), 67–78.

Wedlick, T. R. & Okamura, A. M. (2009). Characterization of pre-curved needles for steering in tissue. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 1200–1203).

Wood, N. A., Shahrour, K., Ost, M. C., & Riviere, C. N. (2010a). Closed-Loop Control of Needle Steering for Percutaneous Kidney Access. In *ICRA Workshop on Snakes, Worms and Catheters: Continuum and Serpentine Robots for Minimally Invasive Surgery*, number 4 (pp. 48–50).

Wood, N. A., Shahrour, K., Ost, M. C., & Riviere, C. N. (2010b). Needle Steering System using Duty-Cycled Rotation for Percutaneous Kidney Access. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5432–5435).

Xu, J., Duindam, V., Alterovitz, R., & Goldberg, K. (2008). Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring Random Trees and backchaining. In *Proceedings of the IEEE International Conference on Automation Science and Engineering* (pp. 41–46).

Yan, K. G., Podder, T., Ng, W. S., & Yu, Y. (2007). 'Smart' Needle for Percutaneous Surgery: Influential Factor Investigation. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 461–464).

Yang, A. T. (1963). *Application of quaternion algebra and dual numbers to the analysis of spatial mechanisms*. Phd, Columbia University.

Zhang, J., Wei, W., Ding, J., Roland, J. T. J., Manolidis, S., & Simaan, N. (2010). Inroads toward robot-assisted cochlear implant surgery using steerable electrode arrays. *Otology & Neurotology*, 31(8), 1199–206.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

Bernardes, M. C., Adorno, B. V., Poignet, P., & Borges, G. A. (2012). Semi-automatic needle steering system with robotic manipulator. In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 1595–1600).

Bernardes, M. C., Adorno, B. V., Poignet, P., Zemiti, N., & Borges, G. A. (2011a). Adaptive path planning for steerable needles using duty-cycling. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2545–2550).

Bernardes, M. C., Adorno, B. V., Poignet, P., Zemiti, N., & Borges, G. A. (2011b). Path plan- ning for steerable needles using duty-cycled spinning. In Proceedings of the International Skills Conference (pp. 44–47).

Bernardes, M. C., Adorno, B. V., Zemiti, N., Poignet, P., & Borges, G. A. (2011c). Path planning for steerable needles using duty-cycling. In Proceedings of the International Congress and Exhibition on Computer-Assisted Radiology and Surgery (pp. 293–294).

INDEX