Robotic and Haptic Assistance for Kidney Access during Percutaneous Nephrolithotomy

by

Olivia Wilz

A thesis submitted to the School of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the degree of

Master of Applied Science in Mechanical Engineering

Faculty of Engineering and Applied Science
University of Ontario Institute of Technology
Oshawa, Ontario, Canada
August 2021

Thesis Examination Information

Submitted by:

Olivia Wilz

Master of Applied Science in Mechanical Engineering

Thesis Title:

Robotic and Haptic Assistance for Kidney Access during Percutaneous Nephrolithotomy

An oral defense of this thesis took place on August 16, 2021 in front of the following examining committee:

Examining Committee

Chair of Examining Committee	Dr. Martin Agelin-Chaab
Research Supervisor	Dr. Carlos Rossa
Nominal Research Supervisor	Dr. Scott Nokleby
Examining Committee Member	Dr. Matthew Harker
Thesis Examiner	Dr. Ruth Milman

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

PERCUTANEOUS nephrolithotomy (PCNL) is a procedure to remove kidney stones through an incision in the patient's back. Gaining kidney access is the most challenging task in this procedure and the cause of many complications.

This thesis presents semi-autonomous solutions for improving PCNL procedure outcomes. First, a cyber-physical PCNL simulator incorporating haptic feedback and simplifying surgeon mental workload via teleoperation of a nephroscope, the surgeon controls the tooltip position while a robotic agent controls its orientation.

This thesis then explores how subtask automation can further improve the procedure. A multi-objective path planning algorithm is implemented to generate multiple suitable paths for kidney access, from which an expert surgeon selects one for execution. The robotic agent then steers the tool along the path autonomously. A further advancement adds a tool/tissue interaction model which determines tool bending; thereby providing accurate trajectory tracking. The concepts are validated experimentally in ex-vivo and phantom tissues.

Keywords: robotic surgery, cyber-physical simulator, semi-autonomous, tool bending, percutaneous nephrolithotomy, path planning, haptic feedback

Author's Declaration

I declare that the thesis presented here is my original work that I have authored. It is a true copy of the thesis.

I grant the University of Ontario Institute of Technology the right to lend this thesis, for the purpose of scholarly research, to other individuals or institutions. Additionally, I grant the University of Ontario Institute of Technology the right to reproduce by means of photocopying or other, this thesis, partially or fully per the request of other institutions or individuals for the purpose of scholarly research. I acknowledge this thesis will be made publicly available through electronic means.

Statement of Contribution

Parts of the work described in this thesis have been published in peer-reviewed journals, and conferences as follows:

Chapter 1

O. Wilz and C. Rossa, "Advances in Semi-Autonomous Percutaneous Nephrolithotomy," International Conference on Smart Multimedia (ICSM), Springer, Marseille, France 2021. [submitted]

Chapter 2

O. Wilz, B. Sainsbury, and C. Rossa, "Constrained haptic-guided shared control for collaborative human–robot percutaneous nephrolithotomy training," Mechatronics, vol. 75, p. 102528, 2021. https://doi.org/10.1016/j.mechatronics.2021.102528

Chapter 3

O. Wilz, B. Sainsbury, and C. Rossa, "Multiobjective Path Planning for Autonomous Robotic Percutaneous Nephrolithotomy via Discrete B-spline Interpolation," International conference on Systems, Man, and Cybernetics (SMC), IEEE, Melbourne, Australia 2021, [accepted]

Chapter 4

O. Wilz, B. Kent, B. Sainsbury, and C. Rossa, "Multiobjective Trajectory Tracking of a Flexible Tool during Robotic Percutaneous Nephrolithotomy," in Robotics and Automation Letters, 2021. https://doi.org/10.1109/LRA.2021.3102946

Also selected for presentation at

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021.

I conceptualized the concept of robotic assistance for PCNL, developed the mathematical formulation for all chapters included in this thesis, developed the experimental setup and protocols, programmed the robotic system and implemented its controller, wrote all publications as well as this manuscript and created the content such as figures

and tables for each chapter with the except for those figures that have been identified as being reproduced from outside sources. Brayden Kent provided the matlab code for the NSGA-II algorithm used in section 3.2 chapter 3 and section 4.2 chapter 4. Brayden Deboon created the formulation for incorporating joint limits analytically into the inverse kinematics used in chapter 2. Ben Sainsbury provided funding and consulting. Carlos Rossa assisted in theoretical analysis, experiment design, and editing and revising the manuscript.

Acknowledgments

I WANT to extend my sincere gratitude to my supervisor Dr. Carlos Rossa for his exceptional guidance, patience, and understanding. I have learned so much more than I expected over the past two years, and have vastly improved my writing thanks to the countless detailed and thorough revisions provided by Dr. Rossa. I am grateful for all of his suggestions when I got stuck in my research; for answering all of my questions seemingly regardless of what time of day I asked them; and for his tireless enforcement of the Canadian spelling of words. With his support and assistance, I always endeavoured to do my very best and have accomplished more than I knew I was capable of.

Additionally, I want to thank my fellow biomechatronics lab members, Maciej Łącki, Brayden Deboon, Brayden Kent, Rick Hao Tan, Conor McDermott, and Ben DeBoer for all of their help reading my manuscripts. As well as their friendship and conversations that helped to make even the toughest days a little easier.

Finally, I want to thank Will and Natascha for all of their support, and for being my biggest cheerleaders.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations	
AR	Augmented Reality
CT	Computed Tomography
DOF	Degree(s) of Freedom
NCCT	Non-contrast-enhanced Computed Tomography
PCNL	Percutaneous Nephrolithotomy
RCM	Remote Centre of Motion
SWL	Shockwave Lithotripsy
US	Ultrasonography
VR	Virtual Reality

Symbols (all chapters)	
$\dot{\square}$	indicates the first time directive
$\ddot{\square}$	indicates the second time directive
$\square^T$	indicates the vector or matrix is transformed

Symbols in **Chapter 2**	
${}^{i-1}_{i}\mathbf{T}$	robot joint frame transformation from $i-1$ to i
$c(\cdot)$	cosine
$s(\cdot)$	sine
θ_i	angle of i^th joint about rotational axis
α_{i-1}	angular change of rotational axis from joint $i-1$ to i
a_{i-1}	joint axis distance from previous joint to current joint
d_i	joint displacement along rotational axis
j	DOF of the robotic manipulator
${}^{j}_{t}\mathbf{p}$	tool-tip displacement from robot end-effector
${}^{j}_{c}\mathbf{p}$	constrained point displacement from robot end-effector

${}^0_t\mathbf{p}$	Cartesian position of tool-tip relative to robot base frame
${}^0_c\mathbf{p}$	Cartesian position of constrained point relative to robot base frame
d_t	tool length
d_c	displacement along tool axis to constrained point
l	distance between constrained point and tooltip
$\mathbf{x}$	the column vector containing the Cartesian position of the tooltip and constrained point
$\mathbf{\Theta}$	the column vector of all the joint angles θ_i
$\dot{\mathbf{x}}$	velocity vector of the tooltip and constrained point
$\dot{\mathbf{\Theta}}$	vector containing the angular velocities of the joints
$\mathbf{\Theta}$	initial joint angles
$\mathbf{J}$	the Jacobian constructed based on tooltip and constrained point velocity equations
θ_i^u	maximum joint angle of joint i
θ_i^ℓ	minimum joint angle of joint i
$\upsilon_i(\theta_i)$	joint i remapped onto the tangent function
$\vartheta_i(\upsilon_i)$	remapped joint i when returned to the joint space
$\mathbf{J}_c(\upsilon)$	the constrained Jacobian matrix
$\mathbf{J}^\dagger(\upsilon)$	the pseudo-inverse of $\mathbf{J}(\upsilon)$
$\mathbf{I}$	an identity matrix
μ	a scalar damping constant, it has a small positive value
υ_d	the inverse kinematics solution, a desired set of joint angles
$\mathbf{x}_d$	desired tooltip and constrained point coordinates
ϵ	error between desired tooltip and constrained point locations and their actual locations
N	total number of demonstration data points
ξ_d	the set of all demonstration data points
ξ_d^n	demonstration data point n
ξ	data points comprising the potential field workspace
k	total number of data points used to define the potential field

$\phi_i^n(\xi)$	potential energy field based on demonstration point n
ϕ_0^n	initial potential energy for point n
S^n	stiffness of potential energy field for point n
ξ_i	the i^{th} data point in ξ
$\omega_i^n(\xi)$	weighting element for data point n
σ^n	the smoothing parameter for data point n
$\mathbf{\Phi}(\xi)$	weighted sum of all individual potential field elements
$\psi^n(\dot{\xi})$	damping field for data point n
$\mathbf{D}^n$	dissipative gain for data point n
$\dot{\xi}$	first time derivative of ξ
$\mathbf{\Psi}(\xi, \dot{\xi})$	weighted sum of all dissipative elements
$\mathbf{F}$	haptic feedback force
∇	gradient operator
ξ_t^m	tooltip data point m from an experimental trial
t_{tot}	total time to complete the task
M	total data points for an experimental trial
ξ_L	path length
ν_m	tooltip velocity at point m
δt	time step between data points
ν_μ	average tooltip velocity
a_m	tooltip acceleration at point m
a_μ	average tooltip acceleration
ξ_I	idealized user trajectory
ε_m	deviation of point m from idealized trajectory
ε_μ	average deviation from ideal trajectory
ξ_c	Cartesian coordinates for the phantom kidney stone
TA	targeting accuracy

Symbols in **Chapter 3**

$N_{i,j}$	basis function for B-spline curve

d	degree of the B-spline
$m+1$	size of the knot vector
$n+1$	number of anchor points
$\mathbf{t}$	knot vector
t_i	knot i in the knot vector
$C(t)$	B-spline curve
$\mathbf{P}_i$	anchor point i
q	number of internal knots
D	dimension
f_1, f_2, f_3, f_4	cost functions
a, b	starting point and ending point of curve
T	matrix containing finite element coordinates of the phantom kidney
$C_R(j)$	resampled B-spline curve
ℓ	number of points in $C_R(j)$
L	tool length
$C_{RT}(j,i)$	coordinates of the discretized tool shaft
z	a displacement along the tool shaft
K	tissue stiffness
$s(z)$	displacement of the tool from the original entry path
$U(d)$	tissue compression potential energy
θ	angular offset of the tool
x_t	x coordinate of the tool-base (robot end-effector)
$x_{OP}(z)$	a function describing the x coordinate of the original entry path at the displacement z along the tool axis
z_1, z_2	displacement of the tissue entry point and exit point along the tool axis
$\mathbf{n_{p1}}$, $\mathbf{n_{p2}}$	parent population members
$\mathbf{n_{c1}}$, $\mathbf{n_{c2}}$	child population members
B	value determines crossover between to parent population members
δ_l, δ_r	values determine the mutation of a child member

μ_c, μ_m	randomly generated crossover and mutation values
ζ_c, ζ_m	crossover and mutation factors
$\mathbf{n}_{UB}$, $\mathbf{n}_{LB}$	parameter upper and lower bounds
$\tilde{f}(\mathbf{n})$	normalized objective function
R_k	dimensional vector
$s_1(\mathbf{n})$, $s_2(\mathbf{n})$	euclidean and normal distances for the population

Symbols in **Chapter 4**

z	a displacement along the tool shaft
ν	displacement of the tool
n	maximum number of vibration modes being considered
$q_i(z)$	vibration mode i
$g_i(d)$	modal coordinate i
γ_i	constant used in bending formulation
β_i	constants for a cantilever beam with one fixed and one free end
L	tool length
$\Pi(d)$	total energy in the system
$U_d(d)$	potential energy stored in the tool due to bending
$U_t(d)$	tissue compression potential energy
$EI(z)$	flexural rigidity
E_s, E_{tool}	Young's modulus of elasticity of the sheath and tool
I_s, I_{tool}	second moment of inertia for the sheath and the tool
P^t	coordinates of a point given relative to the tool reference frame
P^0	coordinates of a point given relative to the base reference frame
θ_1	orientation of the end-effector
θ_2	mounting angle of the tool to the end-effector
K	tissue stiffness
$s(d, z)$	displacement of the tool from the original entry path
θ	angular offset of the tool
x_t	x coordinate of the tool-base (robot end-effector)

$x_{OP}(z)$	a function describing the x coordinate of the original entry path at the displacement z along the tool axis
z_1, z_2	displacement of the tissue entry point and exit point along the tool axis
ϕ_{ij}, ω_{ij}, ψ_i	placeholder variables for tool bending energy and tissue compression equilibrium energy components
M	matrix of the energy equilibrium equations
μ_{ij}	equation for a single entry in matrix M
f_1, f_2, f_3, f_4	cost functions
$P_G^0(i)$	reference position
$P_E^0(i)$	tooltip position
P_t^0	robot end-effector position

Chapter 1

Advances in Semi-Autonomous Percutaneous Nephrolithotomy

NEPHROLITHIASIS also known as kidney stones or urinary calculi are a common ailment around the world, affecting between 1.7 to 14.8 % members of any given population [1–4]. Kidney stones are formed in the kidneys from mineral build-up, primarily calcium and other minerals [1, 4, 5]. A depiction of the renal filtration system and mineral deposits is given in Fig. 1.1. While many small kidney stones can pass down the urinary tract with little to no discomfort, larger kidney stones cause intense discomfort and pose a risk of blocking urine flow. In these cases, medical interventions are required [1, 6]. A variety of treatment options are available for kidney stones, they range from non-invasive to minimally invasive [1]. Although open surgical lithotomy was used in the past for kidney stone management, this has been made obsolete through the adoption of less invasive solutions [1].

A patient suspected of having a kidney stone undergoes imaging to confirm the diagnosis; ultrasonography (US) and computed tomography (CT) scans are both common imaging tools used for this purpose [1, 7, 8]. X-rays and MRI have also been used as imaging methods for diagnostics and determining the stone sizes, although they are not used as frequently as US and CT imaging modalities. US imaging passes acoustic waves through a patient's tissue. The waves are reflected by parts of the tissue and the US probe measures the reflected response to generate an image [7]. US is affordable and free from ionizing radiation but it is not effective for accurately diagnosing and estimating stone size, which often leads to smaller kidney stones not being identified, or an overestimation in stone size [1, 5, 7, 8]. Non-contrast computed tomography (NCCT) is identified as a gold standard for diagnosing kidney stones [5, 7, 8]. During a CT scan, an x-ray reflective contrast agent is injected into a patient while x-ray images are taken from several angles to provide detailed images of internal anatomy. The contrast agent permeates soft tissues through the vascular system, making them more visible on the NCCT scan. However, in NCCT the contrast is not injected since the goal is primarily to image kidney stones that do not benefit from the contrast and can be obscured by it [7, 8]. NCCT provides clearer images with more details compared to US, although it is generally more expensive and exposes patients to ionizing radiation [1, 5, 7, 8].

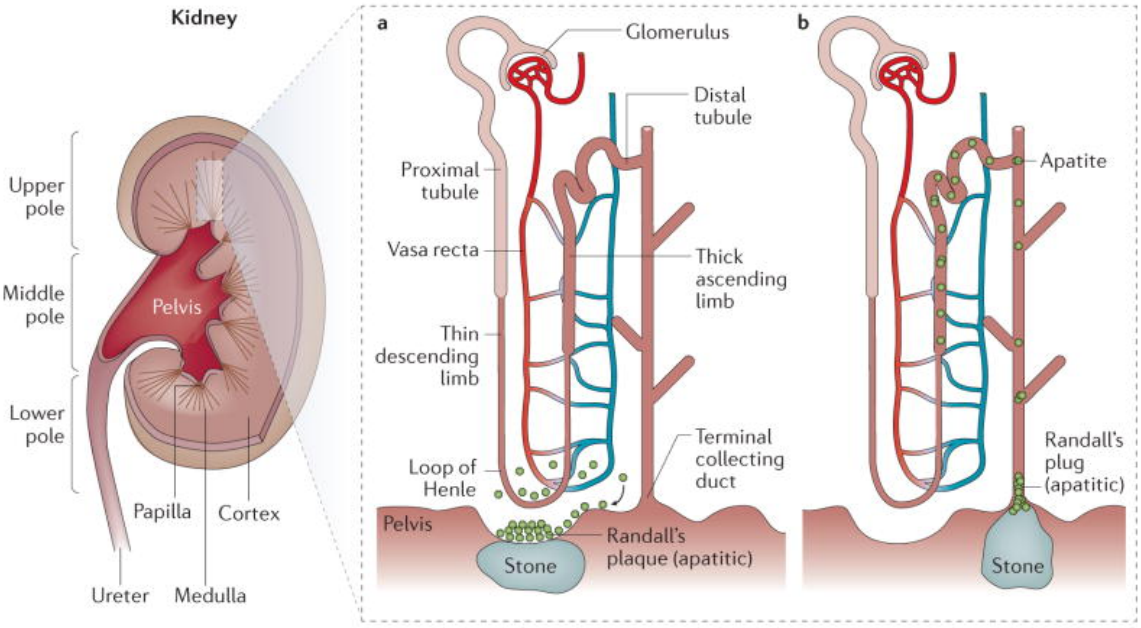


Figure 1.1: Kidney anatomy and filtering system. The general kidney anatomy is shown on the left with the location of kidney poles identified. Two mechanisms are shown for possible kidney stone formation: in (a) the calcium phosphate is deposited into the interstitium, once it reaches the papillary surface, calcium oxalate deposits in the urine continue to grow the stone. In (b), calcium oxalate deposits can plug the renal tubules and begin to form a stone. [1]

Prior to the use of a stone-removal procedure, medical imaging is used to determine the size, shape, and location of the kidney stones, which assists in determining the appropriate treatment option. The treatment option used for each patient then depends on the size and location of the urinary calculi. The three most commonly used treatment options are shockwave lithotripsy (SWL), ureteroscopic fragmentation and retrieval, and percutaneous nephrolithotomy (PCNL) [1, 5]. Individual practitioners use their discretion to determine the correct treatment for each scenario.

1.1 Kidney Stone Treatment Options

The most commonly used treatment option is SWL, making up approximately 40% to 50% of medical interventions for kidney stones worldwide since the majority of stones

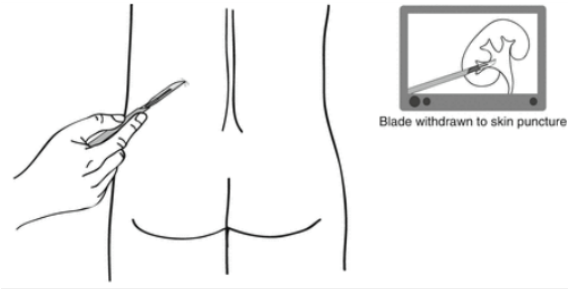
[1] Reprinted by permission from Springer Nature, Nature Reviews Disease Primers, Saeed R. Khan et al [1] © Macmillan Publishers Limited 2016.

are relatively small [1]. Shockwave lithotripsy is the use of a high-energy acoustic wave that is powerful enough to break-up the kidney stones. The stone fragments are then passed naturally or SWL is combined with ureteroscopic removal. Guidelines indicate that this treatment is appropriate for stones in the lower pole of the kidney (see Fig. 1.1 for what region of kidney makes up the lower pole) and less than 15 mm in size or for stones in the upper or middle poles if they are less than 20 mm [1, 5]. Imaging modalities like ultrasound or fluoroscopy are commonly used during SWL to determine the precise locations of each stone so they may be accurately targeted with the acoustic wave.

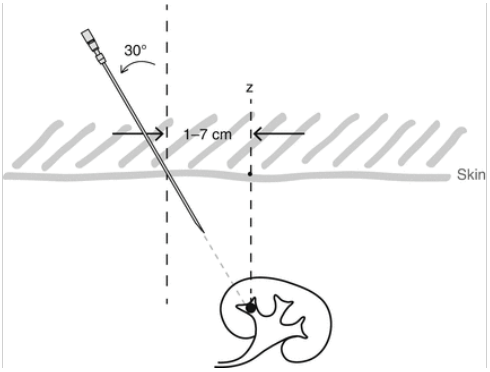
Ureteroscopic interventions are also frequently applied treatment methods, with occurrences of roughly 30% to 40% worldwide, they are generally applied to larger kidney stones [1]. These procedures involve inserting an endoscope through the urethra and guiding it inside the kidney to gain access to the stones and potentially provide additional tools or other assistance in stone retrieval or removal [1, 5]. These intervention methods are often used consecutively with other interventions such as shockwave lithotripsy and PCNL.

Percutaneous nephrolithotomy is the leading intervention for large kidney stones (usually greater than 20 mm), irregularly shaped ones (like staghorn calculi), or when other intervention methods have failed [1, 5, 9, 10]. It is also a procedure frequently used when a patient has anatomical abnormalities that make alternative treatment options nonviable [9]. Since PCNL is a more invasive treatment option for more serious or complicated instances of kidney stones, it is not practiced as frequently as SWL or ureteroscopy; it is the treatment option used in 5% to 10% of renal calculi cases [1].

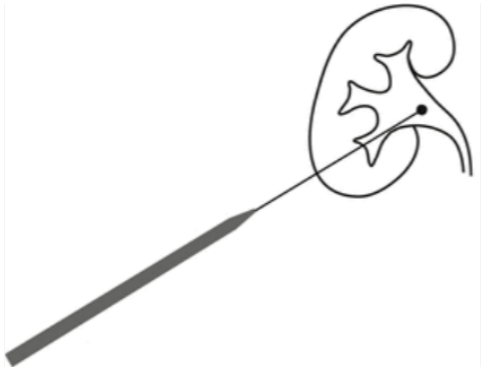
This minimally invasive procedure involves making a small incision in a patient's back, see Fig. 1.2(a), the exact location of the incision depends on the locations of the kidney stones although it is more common to enter the lower pole of a kidney [1, 9]. A needle is inserted into the incision to puncture the kidney and enter the renal pelvis through a calyx, see Fig. 1.2(b). This is referred to as the puncture step and is considered the most challenging and critical component of PCNL [9]. 2D imaging modalities are



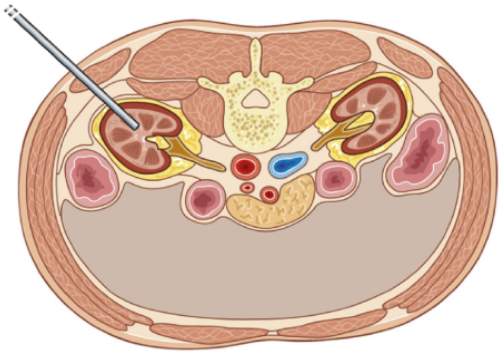
(a) The incision is made in the patient's back. [1]



(b) A needle is inserted through the puncture. [1]



(c) The guide wire is used to guide the dilator as it is inserted to expand the entry path into the kidney, a sheath is then inserted.[2]



(d) The nephroscope is inserted through the sheath into the kidney, tools are inserted through the nephroscope.[3]

Figure 1.2: Basic steps for performing PCNL, beginning with the incision and puncture, followed by guide wire insertion and dilation of the entry path and inserting a sheath. With the sheath inserted stone fragmentation and removal occurs.

used to correctly align the tool for the puncture and to execute it, ultrasound and fluoroscopy are the two widely accepted imaging methods for PCNL [1]. Due to the challenging nature of gaining kidney access, it is one of the most common causes of injury and excessive bleeding during PCNL [11, 12].

Once access to the kidney has been established, a guidewire is inserted through the entry path and into the ureter, see Fig. 1.2(c) [9, 11]. With the guidewire in place, a dilator is used to expand the entry path into the kidney, it needs to be expanded to

[1]Reprinted by permission from Springer Nature, Percutaneous Nephrostomy, David Webb [13] © Springer International Publishing Switzerland 2016. Cropped slightly from originals.

[2]Reprinted by permission from Springer Nature, Routine PCNL, David Webb [14] © Springer International Publishing Switzerland 2016. Cropped slightly from original.

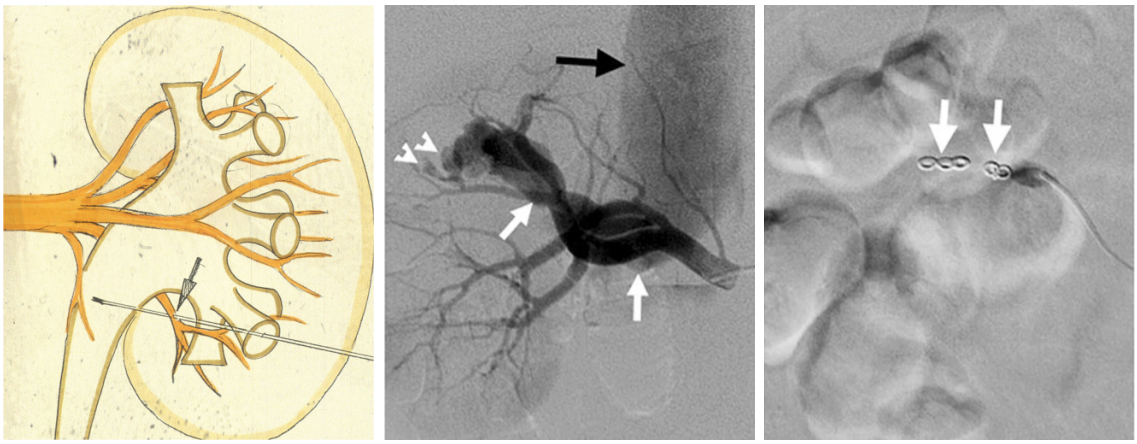
[3]Reprinted by permission from Springer Nature, Positioning During PNL, András Hozenk [15] © Springer International Publishing Switzerland 2020.

create room for larger tools and a sheath. Once the path has been expanded with the dilator, a sheath is put in place to maintain this opening and the dilator is removed. The nephroscope can now be inserted through the sheath [14], Fig. 1.2(d). The nephroscope can now navigate within the kidney to reach various kidney stones.

The next step in the procedure is to break-up and remove the kidney stones. Sometimes stones are small enough that they can be removed directly, although most must be fragmented prior to removal [16]. Shockwaves are used to fragment the stones within the kidney similar to SWL, generally ultrasonic lithotripsy is used for stone fragmentation, although pneumatic or laser stone fragmentation methods are sometimes necessary for particularly hard calculi [9]. While the stones are being fragmented, the kidney is irrigated to remove small stone fragments as they are created [14]. Forceps may also be used to remove larger kidney stone fragments through the nephroscope [14].

Although PCNL is a highly effective kidney stone treatment option, it is considerably more invasive than SWL and ureteroscopic fragmentation and retrieval; therefore, it has higher complication rates in particular related to bleeding and infections [1,5,9,10]. Most of the complications related to PCNL are caused by mistakes in tool steering when gaining kidney access. During the puncture step, it is possible to miss the kidney and instead damage the surrounding tissue. This includes puncturing the pleural leading to a pneumothorax, hydrothorax, or hemothorax, or puncturing nearby organs such as the colon [17–20]. It is also possible to incorrectly puncture the kidney; causing a vascular injury from the puncture of the renal pelvis or a calyceal infundibulum, this is a possible result of not entering through a calyx [1, 17–20]. An example of a possible cause for a vascular puncture and the resulting injury is shown in Fig. 1.3. Any of these described errors in tool steering during the puncture step can lead to excessive bleeding and possible infection [17, 18, 21]. Barring the puncture step, excessive bleeding can be caused by the sharp angulation of the nephroscope or sheath during PCNL [17, 18], therefore, careful control of the orientation is vital.

Since PCNL is not practiced as frequently as other kidney stone treatment options,



(a) Example of a puncture that causes a vascular injury. The tool punctures the vasculature and the renal pelvis.[1]

(b) A patient's renal artery was punctured during PCNL, there are two pseudoaneurysms and a large arteriovenous fistula. [2]

(c) The kidney after treating arterial injuries, excessive bleeding was stopped, pseudoaneurysms were embolized. [2]

Figure 1.3: Possible cause of vascular puncture during PCNL puncture step, and the resulting bleeding complications from a patient who received a vascular injury during PCNL.

training opportunities are limited, further exacerbating the existing challenges for novices when learning the procedure, in particular the initial puncture of the kidney. Despite being the leading intervention for severe instances of renal calculi, only 11% of urologists successfully gain access to the kidney stones themselves [22]. A variety of studies have explored how many practice procedures must be performed to reach proficiency, the reported number of trials recommended before a novice surgeon is consider proficient varies from 30 to 100 attempts [6, 23, 24].

Thus, it is integral to patient safety to provide novices with numerous effective training opportunities, or additional real-time assistance methods during the procedure while thoroughly minimizing risks to patients. A variety of training options have been created throughout the years to assist new surgeons in gaining the surgical skills required to perform PCNL. These training options include both more traditional options that are a physical facsimile to part of the human body and more technologically advanced options such as simulators.

[1]Reprinted from European Urology, vol. 51, Maurice Stephan Michel, Lutz Trojan, Jens Jochen Rassweiler, Complications in Percutaneous Nephrolithotomy, p.899-906, Copyright 2016, with permission from Elsevier [18]

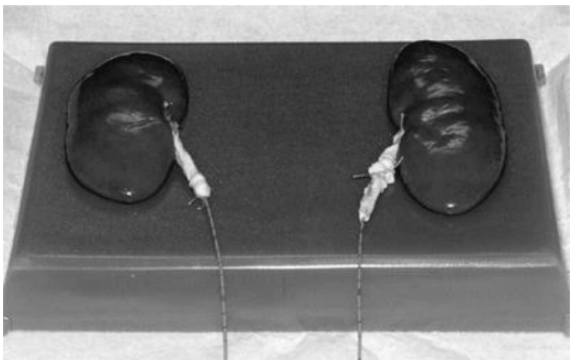
[2]Reproduced under CC BY from [25], cropped slightly from originals

1.2 Conventional Physical Training Options

Physical models are frequently used to train surgeons for numerous procedures including PCNL. Training models are generally categorized as either wet models, which are composed of real tissue, and dry models which are composed of synthetic materials [6, 23, 26, 27]. Wet models include a variety of ex-vivo and in-vivo animal tissue, human cadavers, while dry models include synthetic tissue phantoms, and large bench models [6, 23, 26–28]. Wet and dry models are capable of providing similar training opportunities, although the precise benefits of each vary [6, 23, 26].

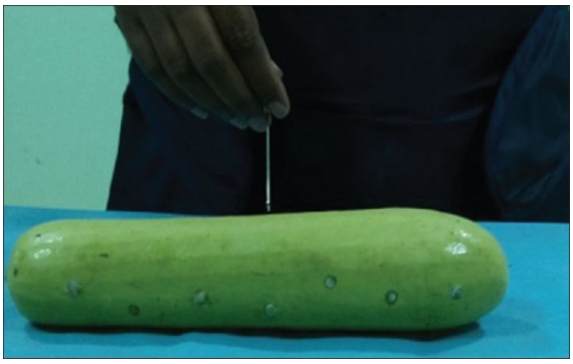
1.2.1 Wet training models

Wet models provide considerably realistic training scenarios as they are made of animal or human tissues, and provide accurate force feedback with relatively accurate anatomy depending on the model. Small wet models are composed of ex-vivo animal tissue, porcine is the most common selection [27]. A small wet model may be comprised of a pair of animal kidneys suspended in a material like silicone [28], or placed inside additional animal tissue to represent the anatomical structures surrounding the kidney [6, 29–31]; two of these small wet models are shown in Fig. 1.4(a) and 1.4(b). This type of wet model provides training opportunities for gaining kidney access [6, 9, 24], and opportunities to train other PCNL subtasks such as stone scavenging, drainage catheter insertion; applying an imaging modality like ultrasound, and fundamental surgical skills like incisions and sutures [28]. Some wet models are small samples of porcine or poultry tissue that are used to train surgical dexterity and other basic skills such as suturing. One of the largest benefits of small wet models is their affordability and availability since ex-vivo porcine tissue can be sourced from local butchers. However, these are low fidelity models which do not represent any particular part of the kidney or other anatomy related to the procedure.



(a) Catheterized porcine kidneys layed out on a silicone base before being enclosed in silicone. [1]

(b) Catheterized porcine kidney on the porcine flesh used to represent the patients back. [2]



(c) Vegetable model used to train puncture step of the procedure. [3]

(d) Anesthetized pig used to train entire PCNL procedure. [4]

Figure 1.4: A variety of wet training models for PCNL.

Larger wet models are also frequently used during later training phases once basic surgery skills and knowledge have been acquired, they allow trainees to practice full procedures and all skills related to PCNL. Similar to smaller wet models, they provide accurate force feedback and other tissue characteristics. Large wet models include human cadavers which provide the most accurate anatomy with no risk to the patient. Although, medical cadavers can be challenging to acquire and this minimizes training opportunities with them [6, 34]. In addition to cadavers, live anesthetized pigs, as

[1]Reprinted by permission from Springer Nature, Ex vivo training for percutaneous renal surgery, Walter Ludwig Strohmaier et al [28] © Springer-Verlag 2005.

[2]Reprinted by permission from Springer Nature, New ex vivo organ model for percutaneous renal surgery using a laparoendoscopic training box: the sandwich model, Stephan Jutzi et al [31] © Springer-Verlag Berlin Heidelberg 2013.

[3]Reproduced under CC BY-NC-SA 3.0 from [32]

[4]Reprinted from BJU International, vol. 106, Mahesh Desai, Ravindra Sabnis, Veeramani Muthu, et al, Percutaneous renal access training: content validation comparison between a live porcine and a virtual reality (VR) simulation model, p.1753-1756, Copyright 2010, with permission from John Wiley and Sons [33]. Cropped slightly from original.

shown in Fig. 1.4, are used to practice the procedure [33]. Training with live animals provides highly realistic scenarios with the opportunity to train on a patient that bleeds, breaths, and presents other physical phenomena which are experienced when working with live patients [23,27]. Live animals can be difficult to store and transport; additionally, there are ethical and legal issues that must be considered when working with live animals [23].

While not frequently studied, some early training for gaining kidney access may be performed on fruit or vegetable models (aubergines and watermelons are common choices) an example of this is shown in Fig. 1.4(c) [6, 32]. Although these are not animal-derived tissue specimens like others presented in this section, they are still similar enough to other wet model examples to be included here.

The biggest drawbacks of wet model training are their limited uses. Many of the models can only be used once which drastically limits training opportunities [6,23,27]. The materials for the smaller wet models must be sourced and the models constructed for each use, which presents a significant financial requirement and requires manpower to create enough models for several training opportunities. Live animals pose even greater problems in regard to their transport, storage, and care. Moreover, live animals require anesthetic resources and manpower during the procedure which requires notable planning efforts for a single-use training scenario while posing ethical and legal issues [6, 23, 27]. Finally, while human cadavers pose excellent training opportunities there are limited training scenarios available with these and they only provide a single training use each.

1.2.2 Dry training models

An alternative to wet models are dry models, these provide repeated usage and are generally easier to transport [6, 26]. These include some low fidelity tissue phantoms that do not closely represent any particular anatomy related to PCNL and as such, they are used primarily to train individual subtasks and skills required to complete the entire procedure [26, 35]. Matsumoto in [36] found that low-fidelity models are

(a) The URO Mentor physical training simulator. Source: reproduced from [37]

(b) Medskills Standard Scope Trainer. Source: reproduced from [38]

Figure 1.5: Three types of physical dry training models.

effective training devices as long as the skills trained on them accurately represented the skills required in actual practice. To training on the entire PCNL procedure in an operating room environment, larger physical models are necessary to provide trainees the ability to repeatedly train for PCNL. A few different larger bench models exits such as the URO Mentor from *3D Systems* [37] and the Medskills Standard Scope Trainer [38], depicted in Fig. 1.5(a) and 1.5(b) respectively. These larger bench models allow trainees to practice the entire procedure of PCNL rather than individual skills or subtasks. Some of the more advanced physical simulators like the URO Mentor can mimic a variety of tactile and physical phenomena while providing imaging simulated imaging feedback [37]. While these models represent a slightly higher initial cost than wet models, their repeated usability makes them a common and effective training tool [6, 23, 26, 35]. The drawback associated with many bench models and tissue phantoms is in the tactile feedback they provide, which does not mimic the feedback experienced during percutaneous nephrolithotomy. Additionally, bench models only allow for a few select scenarios or parts of a procedure to be experienced, and this may limit the training opportunities even further [6, 23].

A desirable training model allows for a procedure to be attempted repeatedly while also providing accurate tactile and image feedback. To this end, virtual simulators have been created to provide additional training opportunities for PCNL.

1.3 PCNL Simulation Training Options

PCNL simulators provide a safe and effective training method, allow repeated usage, and can simulate a variety of scenarios. PCNL simulators are able to mimic a variety of physical phenomena, such as breathing, that are encountered during the procedure and may not be present in dry or wet models [26]. They are also able to simulate a variety of scenarios including abnormal kidney anatomy, various kidney stone sizes and placements, and possible complications [6]. Additionally, unlike smaller physical models, they allow the user to practice the entire procedure in most simulators [6, 26]. Further, simulators can record participant performance, tool trajectories, and other measures of performance to provide quantitative and qualitative feedback on performance for the procedure.

A frequently used simulator is the PERC Mentor^{TM}, while sometimes referred to as a virtual reality system, it employs a cyber-physical approach by implementing an interactive physical model as well as computer-based simulated components. Thus, the system is more accurately described as an augmented reality (AR) training simulator, see Fig. 1.6(a) [6, 26, 39–41]. The PERC Mentor^{TM} has demonstrated to be a highly effective training tool for PCNL [40, 41]. Augmented reality creates detailed training scenarios with some tactile feedback from the physical model; although, the available scenarios may be limited by the physical model [26].

Another type of simulator that has been implemented for PCNL training is virtual reality (VR). This allows surgeons to see and interact with the tools and imaging they may experience in an operating room similar to augmented reality. However, the benefit of a more fully virtual platform is in its flexibility in procedure and scenarios. VR systems like the VirtaMed UroS^{TM} Fig. 1.6(c) or the Marion Surgical K181 PCNL Simulator [26] Fig. 1.6(b) are able to provide a wide variety of training scenarios while providing some tactile feedback to the user through a haptic device representing the tool. Both VR and AR simulators can provide data to the user as they record the full procedure including tool movements, completion time, etc. In particular, the benefit

(a) PERC Mentor™ augmented reality simulator. Source: reproduced from [37]

(b) Marion Surgical K181 PCNL virtual reality simulator. [1]

(c) VirtaMed UroS™ virtual reality simulator. Source: reproduced from [42]

Figure 1.6: Virtual and augmented reality PCNL training simulators.

of simulators is their reusability. Individual scenarios can be attempted several times, or adjusted slightly each time, providing more training opportunities for novices. Some of the drawbacks of simulated training systems are the fidelity of tactile information or other physical feedback [6,26]. While both AR and VR generally incorporate either a physical model or some amount of haptic feedback that may be similar to what is experienced during a procedure, these areas still need improvements to be more akin to what is experienced when working with actual tissue [43]. Further, these simulators are generally expensive which creates a higher barrier of entry compared to some of the physical models described earlier.

While a novice will have received significant training on models and simulators, the fine-tuning of their skills with the procedure will be completed on real-world patients. Ideally, the shift to human patients should be made as safe as possible. To that end, robotic assistance can ease the transition while continuing to provide additional feedback or assisting the surgeons during the procedure.

[1] Reproduced under CC BY from [26]. Cropped slightly

1.4 Assistance types available during PCNL

Due to the risks associated with PCNL and the challenges in performing the procedure while gaining competency, assistance can help offset the inexperience of a novice surgeon. These assistance methods are applied in real-time in the operating room to both train a novice and further ensure the safety of the procedure. Assistance can be sought through a variety of means such as visual assistance through augmented reality, physical assistance through haptic feedback, or by semi or fully automating the procedure. Semi-autonomous in the context of surgery can refer to the automation of surgical subtasks: gaining kidney access, scavenging for stones, and more. Alternatively, fully autonomous surgery paradigms include one or more robotic manipulators to perform the procedure with little to no input from the surgeon. Some of these assistance methods have been proposed in literature to be applied to PCNL, while others have been applied to a variety of other surgical procedures and demonstrate qualities that may be suitable for application to PCNL [44–48, 48–54].

Due to the challenges encountered in PCNL from inexperience to challenging patient anatomies, the application of assistance during the procedure is a natural solution when bridging the gap from novice to expert or providing additional help to experts during particularly complex scenarios. A commonly studied type of assistance during PCNL is augmented reality. Some of the earlier uses of augmented reality are applied by using preoperative imaging to create 3D models of the kidney and then projecting this 3D model onto other imaging options to provide the surgeon with more data during the procedure [55–57]. In [55, 56] an Ipad is used as a camera and display, it overlays a correctly oriented 3D model of the kidney on the live image of the procedure while a novice gains kidney access. Similarly, [57] projects the 3D model onto ultrasound images during the procedure to better visualize the kidney's internal structures. Ferraguti et al. propose two different forms of assistance in [58]. First, they apply augmented reality to overlay a patient's anatomical structures in a VR display. Second, they use a robot manipulator to apply haptic assistance, guiding the user to the correct puncture site and assisting in maintaining tool orientation during

the puncture step.

Robot assistance is becoming increasingly common in a variety of surgical procedures due to its precision and consistency. Robotics can aid novices when learning PCNL while maintaining patient safety, or assist expert surgeons in particularly complex scenarios. For example, [59] controlled for out-of-plane motion using robotic assistance, thus, accounting for the patient's breathing during the puncture step. This method of robotic assistance aims to simplify the procedure somewhat for the surgeon performing it by allowing a robot to handle an aspect of the procedure. Likewise, robots can be used to automated surgical subtasks such as [60] where the kidney puncture is performed autonomously.

A clinically implemented version of robotic assistance exists in the form of teleoperative platforms. These allow a surgeon to perform the procedure with increased precision and control. These systems may implement tremor reduction to reduce unintended micro-movements by the surgeon, or through gesture reduction which allows a surgeon to precisely control a surgical instrument with a small workspace. In recent years this form of robotic assistance has been implemented for robotic nephrolithotomy and robotic pyelolithotomy, although these intervention methods are generally only used for more complex cases of nephrolithiasis [61–63]. This method of robot assistance does not provide additional assistance other than increasing a surgeon's control and precision during the procedure. A surgeon still needs to be skilled in order to perform robotic nephrolithotomy and robotic pyelolithotomy since they will receive no guidance or other help from the robot.

Several robotic assistance methods have been applied to medical interventions other than PCNL. These assistance methods can include teleoperative robotic imaging control such as fluoroscopy or ultrasound steering [44–46]. These systems provide precision, stability, and repeatability of the imaging modality. Additional robotic assistance based on visual servoing can greatly benefit from stable imaging with accurate positioning information from the robotic manipulator. Another form of assistance aims to simplify the procedure or increase patient safety by applying constraints

through a teleoperated robotic platform [47, 48]. These constraints aim to simplify the procedure by maintaining the orientation and/or position of a tool, such as the entry point, or by adding environmental constraints that ensure collision-free motion of the tool [48–50]. Simplifying a procedure can also be achieved through the automation of surgical subtasks. Automation of this kind usually requires a surgeon to set up and align the robotic system and supervise the autonomous execution of the subtask. This form of semi-autonomous assistance has been explored for blunt dissection, soft tissue retraction, oral and maxillofacial surgery, and more [51–54].

A less autonomous approach to robotic assistance can be sought through tactile feedback during teleoperation tasks. This form of feedback provides additional information to the surgeon which can be beneficial for novices and experts. Haptic assistance is able to keep a human in control of a procedure, which is often lost in more autonomous assistance methods. One application of haptic feedback for medical interventions is in providing environmental cues to assist a surgeon in avoiding sensitive anatomical structures such as the placenta during fetal laser surgery, vein walls during endovascular procedures, and implementing a working region for brain tumour removal [64–66]. Alternatively, haptic feedback can restrict motion even further by guiding a user along a desired trajectory [67, 68]. Tactile feedback has also been explored to communicate tissue characteristics to the surgeon. This can include the tissue forces to mimic the tactile sensations felt during manual surgery, or to identify different tissue types such as cancerous tissues [69–72]. While haptic assistance has been shown to be beneficial for a variety of medical procedures, it has not been explored in much detail for PCNL outside of training in simulators.

The assistance methods applied to PCNL focus primarily on training scenarios such as those provided in simulators. Some efforts have been made to include additional visual information to novices while performing the puncture step of the procedure. Nevertheless, PCNL training and clinical use would benefit from more advanced frameworks to improve a surgeon's performance during PCNL.

1.5 Proposed Framework

Robotic assistance methods for PCNL have not been thoroughly explored, despite the widespread clinical acceptance of PCNL and the documented challenges and safety concerns. Particularly, little focus is applied to bridging the gap between training platforms and a novice's first real-world interventions. Furthermore, haptic feedback has not been suitably explored and evaluated for its effectiveness in teaching the PCNL procedure. The training framework proposed in this thesis aims to provide a framework to simplify the procedure for the user. Additionally, haptic feedback from an expert demonstration is implemented to teach a novice the correct trajectory while keeping them in control of the operation. This framework can be implemented both on tissue phantoms or other training situations, or during some of the novice's first interventions to provide a smooth transition for the novice from training scenarios to the operating room. A skill assessment method is created to determine the effectiveness of training an unskilled user with haptic feedback. The results from a minimal set of training and evaluation samples found that the haptic feedback proposed in this thesis is a suitable method of training and improves a user's performance even once the haptic feedback is removed.

In addition to haptic feedback for training, robotic assistance is also proposed. Tool path planning and tracking have remained unaddressed in literature relating to robot-aided PCNL. This is addressed in this thesis through the use of a multi-objective optimizer that determines an optimal path through the entry point in the kidney to a desired goal location. The path is represented with a uniform B-spline, which requires only a few anchor points to determine the entire path, simplifying the dimensionality of the optimization problem. The multi-objective optimizer considers a unique set of four cost functions that are selected specifically for the application of gaining access to kidney stones during PCNL. These four objectives include trajectory smoothness, tool proximity to obstacles, tissue strain energy at the entry point, and path length. Each of the four objectives serves to plan an appropriate path that considers patient safety.

Some of the important factor that are considered by surgeons when moving from open surgeries to minimally invasive ones are recovery times and possible complications. PCNL is no exception. Studies have shown that decreasing the size of the nephroscope leads to less bleeding and potentially shorter recovery times. Small nephroscopes, also referred to as micro nephroscopes, are used more frequently in paediatric patients. The smaller size of these tools leads to potential bending of the nephroscope itself. Similarly, the puncture needle may also undergo bending while gaining kidney access. This may lead to trajectory tracking errors during more autonomous approaches. To address this issue a trajectory tracking algorithm is developed in this thesis. It employs a novel tool/tissue interaction model to determine the possible tool bending along a trajectory. This model is combined with a multi-objective optimizer to solve the inverse problem of finding a suitable tool pose where the tip location traces the desired trajectory. This framework is then evaluated through simulations and physical trials on ex-vivo porcine tissue to demonstrate its ability to compensate for tool and tissue deformation during the procedure. This algorithm may also allow for more flexible tools to be used in the future with semi-autonomous frameworks. A more flexible tool would likely cause reduced tissue trauma and provides more manoeuvrability within the kidney, thus, requiring fewer incisions and entry points for more severe cases of nephrolithiasis.

1.6 Thesis Objectives and Outline

This thesis focuses on creating a semi-autonomous and autonomous robotic framework to train novices and simplify the PCNL procedure for experts. The first framework proposed in this thesis provides haptic feedback based on a predefined trajectory to train a novice or assist an expert surgeon during kidney access. Haptic guidance for PCNL has not been explored in literature in detail, despite the documented challenges in successfully gaining kidney access. This guidance is combined with a constrained kinematic environment for the robot arm to reduce the user's workload, thereby, further simplifying the procedure.

The second framework focuses on automating kidney access. First, an optimal tool trajectory is planned using preoperative information. This trajectory is then implemented as an automated subtask where the robot performs the puncture step and reaches the kidney stones autonomously. To further automate gaining kidney access, a tool/tissue interaction model is used to determine potential tool bending and compensate for it with a trajectory tracking algorithm.

This thesis is organized as follows:

Chapter 2 presents the cyber-physical PCNL framework proposed to train novices and has three complementary contributions. First, a teleoperation framework is used where the position and orientation of the robot are decoupled and the operator only controls the Cartesian position of the tooltip. The second contribution lies in the implementation of the haptic assistance. The simulator creates a potential field of forces based on an expert demonstration provided in the physical slave environment. This potential field is then used to determine the magnitude and direction of haptic feedback force to apply. The final contribution proposes a method of evaluating the effectiveness of the framework proposed in this chapter. Sixteen participants took part in a total of five trials each. A unique set of evaluation criteria are created to evaluate the individual performance of each participant and demonstrated the effectiveness of training with haptics.

Chapter 3 presents a path planner that serves as the foundation for automated robotic PCNL. Multi-objective non-dominated sorting genetic algorithm II (NSGA-II) is proposed to plan a B-spline curve through a set of anchor points that represents the desired tooltip trajectory from the entry point to the stone location. The outputs of the algorithm are the optimal spline coefficients that minimize the trajectory length, tissue displacement, and trajectory smoothness, while maximizing the distance to obstacles.

Chapter 4 takes the concept one step further and proposes two contributions. First, a novel tool/tissue model is proposed to describe the deflection of a flexible tool during robot-aided PCNL. The second contribution involves using the model in a multi-

objective optimizer to solve the inverse kinematic problem that allows the deformed tool to follow a predefined trajectory given multiple objectives.

Chapter 5 presents the general conclusion of the work proposed here and recommendations for future work.

Chapter 2

Constrained Haptic-Guided Shared Control for Collaborative Human-Robot Percutaneous Nephrolithotomy Training

Mechatronics, 0957-4158, vol. 75, p. 102528, 2021. DOI: 10.1016/j.mechatronics.2021.102528

2.1 Introduction

THIS chapter introduces a teleoperative framework for PCNL (percutaneous nephrolithotomy) training. The cyber-physical simulator proposed here assists surgeons in two ways: first, by providing haptic feedback to the operator to help teach the optimal manoeuvers to gain kidney access; second, by reducing the procedure workload through surgical subtask automation using a robotic agent.

2.1.1 Haptic Feedback

Haptic feedback has been explored for training in surgical scenarios other than PCNL. While some debate still exists on the effectiveness of haptic assistance and feedback, literature suggests that when haptic feedback is task-specific, it provides additional useful information to a surgeon leading to reduced learning times, improved task performance, quality, dexterity, improved retention rates [73–75], and better acceptance of simulator training by professionals [76]. The most frequent implementation of haptic feedback is by creating virtual fixtures to prevent a tool from reaching certain areas in a workspace to protect sensitive tissues [77–79]. Another form of haptic feedback aims to mimic the tactile forces felt during the procedure, these generally incorporate force sensors into the surgical instruments and recreate these forces in a teleoperative setup [80]. While these forms of haptic feedback have been shown to be beneficial for experienced surgeons during teleoperated surgery, they are not meant to teach a surgical task.

To address this problem, haptic feedback is proposed to guide a surgeon along a predefined path in [67, 81], and specifically to guide a novice through the insertion task of PCNL in [58]. Kidney access was decomposed into different subtasks and a specific form of haptic feedback was provided during each task. The framework required the surgeon to maintain tool orientation during the insertion step, and as such, it requires significant skill and knowledge from the surgeon. Further, haptic cues for gaining kidney access were not suitably assessed for what benefits they provided

during the procedure.

In addition to providing haptic feedback, training can be made easier by simplifying the teleoperative environment. This concept may be explored in surgical subtask automation such as tool orientation, stone fragmentation, stone removal, etc., while the surgeon learns to perform a complementary task. Subtask automation allows the workload to be shared between the operator and the robotic agent. In addition, data acquired from expert surgeons can define an optimal tool path, and subsequently, a robotic agent can assist the surgeon in following that path during surgery. In this context, less experienced surgeons would benefit from the expertise of more experienced surgeons.

2.1.2 Tool orientation subtask automation

Typically, gaining access to kidney stones during PCNL is done under two-dimensional image guidance. It becomes challenging to visualize and mentally recreate the three-dimensional anatomy of the kidney and the relative location of stones and tools. To simplify the workload, the entry point into the kidney can be constrained to minimize damage to the surrounding tissue. With the assistance of a robotic manipulator, a remote centre of motion (RCM) can be implemented to constrain the entry point in this fashion.

Remote centers of motion have been explored for their applicability to medical procedures, primarily through physical assistive devices that restrict a surgeon's available motion [82–85]. Nevertheless, some instances of RCMs used within robotic frameworks do exists. For example, Garcia et al. [86] applied a RCM to a robotic manipulator mounted on a mobile robot. A remote centre of motion is proposed for implementation in a surgical environment in [87]. While the RCM is beneficially in simplifying the procedure for a novice, the inverse kinematic framework described in this chapter proposes to combine a RCM constraint with joint limit constraints incorporated into the inverse kinematic framework of the robotic manipulator to ensure patient safety while minimizing the mental workload for the surgeon.

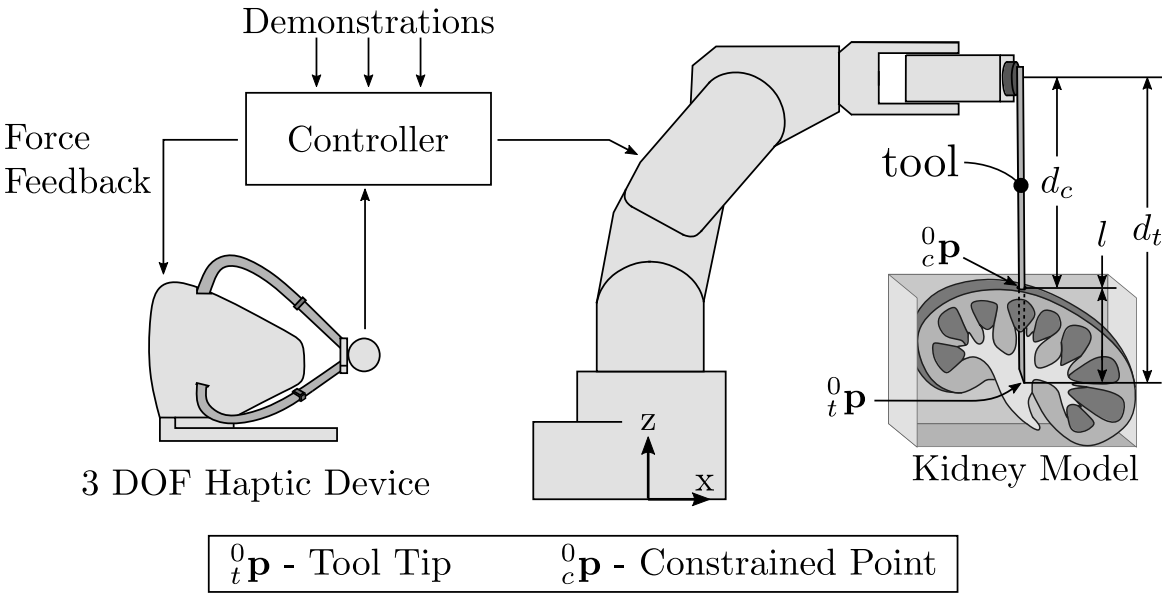


Figure 2.1: The general setup of cyber-physical robot-aided PCNL training framework. The 3 DOF (degrees of freedom) haptic device provides position information to the controller while receiving force feedback information. The controller sends joint angles to the 6 DOF robotic arm, allowing the 3 DOF haptic device to control the 6 DOF robotic arm.

2.1.3 A cyber-physical simulator for PCNL training

This chapter aims to merge haptics and robotic assistance to implement a novel cyber-physical simulator for PCNL training. These two forms of robotic assistance work alongside the operator in a collaborative fashion. With the teleportation scheme shown in Fig. 2.1, the robotic agent helps the operator follow a predetermined kidney access path while maintaining the remote centre of motion. Further, the incorporation of haptic feedback allows the surgeon to compensate for trajectory tracking errors while maintaining full control of the procedure.

The cyber-physical simulator system is shown in Fig. 2.1, where a 3 degree of freedom (DOF) haptic device is used to control a 6 DOF robotic arm. The inverse kinematics constrain the entry point in the tissue to the tool shaft, thus the surgeon only controls the Cartesian position of the tooltip. The controller determines the force feedback to send to the user and updates the robot joint angles based on the inverse kinematics.

This framework has three complementary parts. First, teleoperation is used to decouple the position and orientation of the robot such that the operator only controls

the Cartesian position of the tooltip while the robot ensures that the position of the entry point in the tissue remains unchanged. This is the idea behind a remote centre of motion. The RCM is incorporated into the inverse kinematics by considering the entry point as the tip of a tool with variable length, while the joint limits are included analytically in the formulation using a saturation function for the joint speeds. This is the focus of section 2.2.

The second part of the framework implements haptic assistance. The simulator bases the potential field on an expert demonstration provided in the physical slave environment, where the demonstrator controls the robot through the haptic device, and the stiffness of the potential energy field is dependent on the proximity to the phantom tissue. This potential field is then used to determine the magnitude and direction of haptic feedback force to apply. This is discussed in section 2.3.

Finally, a method to evaluate the effectiveness of the framework is proposed in this chapter.

The remainder of this chapter is organized as follows. Section 2.4 focuses on user trials along with the validation of the analytical joint limitations. The results are discussed in detail in section 2.5. The overall performance of the contributions is discussed in section 2.6, along with potential future improvements.

2.2 Inverse Kinematic Formulation for Constrained Tool Orientation during PCNL

During PCNL a long thin tool is inserted into a small incision into the patient's back to gain access to the kidney; this requires the control of the position of the tooltip as well as the orientation of the tool to ensure that it continues to pass through the entry point during insertion. Once inside the kidney, multiple kidney stone locations may be reached through the same entry point. Thus the path of the tooltip is not a straight path and navigation within the kidney is necessary. While some deviation from the entry point is acceptable this should be minimized to reduce the risk of

additional tissue damage. To this end, the entry point in the kidney can be used to constrain the tool's orientation.

In the context of robot-assisted PCNL, these two points – that is the tooltip position and entry point – are sufficient to solve for the inverse kinematics of the robotic manipulator using a remote centre of motion. This remote centre of motion (RCM) allows the user to navigate the tooltip within the kidney to reach a desired goal location. Since the robot will control the tool to ensure that it passes through the two points in space, the inverse kinematics need to be developed in terms of six Cartesian coordinates (two 3D points) rather than three Cartesian coordinates defining the position and three angles defining the orientation. The inverse kinematics should take into account the joint limits of the manipulator so as not to exceed them during operation.

To configure the inverse kinematics as described above, the forward kinematics must be determined to define the RCM and the tooltip position based on the joint angles of the manipulator.

2.2.1 Forward Kinematics

The forward kinematics of the robot arm is derived using the modified Denavit-Hartenberg convention which specifies a set of homogeneous transformations ${}^{i-1}_{i}\mathbf{T}$ that expresses the position and orientation of the kinematic chain's i^{th} joint with respect to joint $i-1$ as:

$$ {}^{i-1}_{i}\mathbf{T} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1} $$

where $c(\cdot)$ stands for $\cos(\cdot)$ and $s(\cdot)$ for $\sin(\cdot)$, θ_i is the angle of the i^{th} joint about its rotational axis, α_{i-1} is the angle from the previous $(i-1)$ rotational axis to the current (i^{th}) rotational axis sometimes called *link twist*, a_{i-1} is the distance from the

previous joint axis to the current joint axis (generally the previous link length) and d_i is the displacement of the current link along its axis of rotation also known as the *link offset*.

The transformation matrices are multiplied, as shown in (2.2) and (2.3) where j is the total number of degrees-of-freedom (DOF). The tooltip ${}^{j}_{t}\mathbf{p} \in \mathbb{R}^{4\times 1}$ and constrained point ${}^{j}_{c}\mathbf{p} \in \mathbb{R}^{4\times 1}$ can be described relative to the j^{th} reference frame by using only the fourth column of (2.1), since their orientations are not necessary. The tooltip position ${}^{0}_{t}\mathbf{p} \in \mathbb{R}^{4\times 1}$ is

$$ {}^{0}_{t}\mathbf{p} = \prod_{i=1}^{j} ({}^{i-1}_{i}\mathbf{T})\ {}^{j}_{t}\mathbf{p}. \tag{2.2} $$

The constrained point position ${}^{0}_{c}\mathbf{p} \in \mathbb{R}^{4\times 1}$ is

$$ {}^{0}_{c}\mathbf{p} = \prod_{i=1}^{j} ({}^{i-1}_{i}\mathbf{T})\ {}^{j}_{c}\mathbf{p}. \tag{2.3} $$

Since the constrained point is along the tool shaft its location can be defined in the same way as the tooltip with the only notable difference being the tool length d_t and d_c as shown in Fig. 1. The tool itself will have a constant length, while the distance to the constrained point is variable depending on the insertion depth l of the tool inside the kidney such that $d_c = d_t - l$. This means that ${}^{j}_{t}\mathbf{p}$ can be defined as:

$$ {}^{j}_{t}\mathbf{p} = \begin{bmatrix} a_j & -d_t s\alpha_j & d_t c\alpha_j & 1 \end{bmatrix}^T, \tag{2.4} $$

where the transpose is denoted by $(\cdot)^T$, and ${}^{j}_{c}\mathbf{p}$ can be defined relative to ${}^{j}_{t}\mathbf{p}$ as:

$$ {}^{j}_{c}\mathbf{p} = \begin{bmatrix} a_{i-1} & -d_c s\alpha_j & d_c c\alpha_j & 1 \end{bmatrix}^T = {}^{j}_{t}\ \mathbf{p} - \begin{bmatrix} 0 & -l s\alpha_j & l c\alpha_j & 0 \end{bmatrix}^T, \tag{2.5} $$

The vectors created in (2.2) and (2.3) together contain six equations for describing the tool's tip and constrained point positions relative to the base. Altogether they make up the forward kinematic solution. These equations are used to find the inverse kinematic formulas; however, the results need to be bounded by the joint limits of the

manipulator. To take the joint limits into account during the inverse kinematic model, a nonlinear saturation function is implemented that limits the speed of a given joint when it approaches its limit. This is implemented as a nonlinear gain in the Newton Raphson method. Which has the added benefit of providing a unique and feasible solution rather than multiple solutions where a particular configuration needs to be selected.

2.2.2 Constrained Inverse Kinematics

The tooltip and constrained point positions and velocities can both be found using the current joint angles and angular velocities. To find the constrained inverse kinematics the opposite must be done, such that the joint angles can be found based on the tooltip and constrained point positions.

Let p_i be the i^{th} entry in vector $\mathbf{p}$ defined in (2.2) and (2.3). One can define the task space vector $\mathbf{x} \in \mathbb{R}^{6\times 1}$ as

$$\mathbf{x} = [{}^0_t p_1 \quad {}^0_t p_2 \quad {}^0_t p_3 \quad {}^0_c p_1 \quad {}^0_c p_2 \quad {}^0_c p_3]^T. \tag{2.6}$$

Normally, the Jacobian is constructed using three equations describing the position and three describing the orientation of the manipulator. In (2.6), however, the Jacobian is constructed from the six position equations, three for the tooltip position and three for the remote centre of motion. Generally, when constructing a Jacobian matrix for a robotic manipulator, three equations for position and three for orientation are used, rather than 6 position equations.

These equations define the position of the tool; it is important to note that while this allows the 6 DOF manipulator to be controlled with a 3 DOF device, only 5 DOF are being accounted for since the tool can be rotated about its longitudinal axis which does not affect its Cartesian position. Let:

$$\mathbf{\Theta} = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_j]^T. \tag{2.7}$$

be the joint space vector of the manipulator ($\in \mathbb{R}^{j\times 1}$). The Jacobian describes the relationship between $\dot{\mathbf{x}}$ (the vector of the Cartesian velocities) and $\dot{\mathbf{\Theta}}$ (the vector of the angular velocities) as

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{\Theta}}, \tag{2.8}$$

here the time derivative is denoted by the $(\dot{})$ operator. In (2.8), $\mathbf{J} \in \mathbb{R}^{6\times j}$ is defined as:

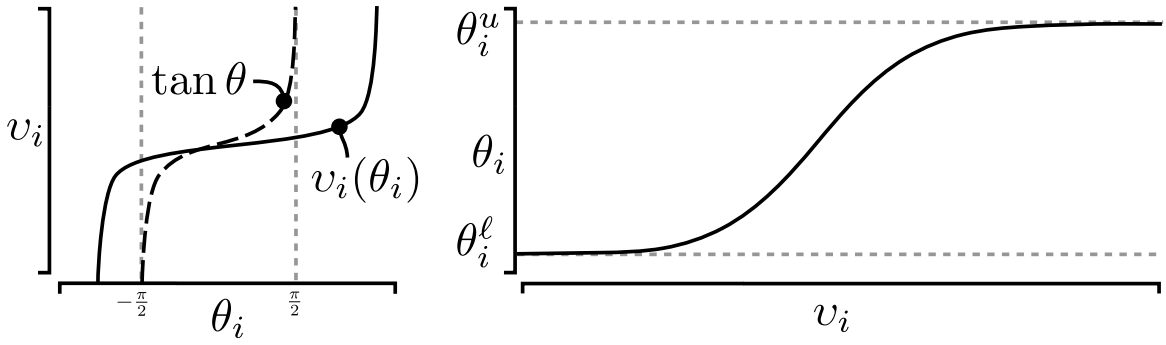
$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \mathbf{\Theta}} = \begin{bmatrix} \frac{\partial^0_t p_1}{\partial\theta_1} & \frac{\partial^0_t p_1}{\partial\theta_2} & \cdots & \frac{\partial^0_t p_1}{\partial\theta_j} \\ \frac{\partial^0_t p_2}{\partial\theta_1} & \frac{\partial^0_t p_2}{\partial\theta_2} & \cdots & \frac{\partial^0_t p_2}{\partial\theta_j} \\ \frac{\partial^0_t p_3}{\partial\theta_1} & \frac{\partial^0_t p_3}{\partial\theta_2} & \cdots & \frac{\partial^0_t p_3}{\partial\theta_j} \\ \frac{\partial^0_c p_1}{\partial\theta_1} & \frac{\partial^0_c p_1}{\partial\theta_2} & \cdots & \frac{\partial^0_c p_1}{\partial\theta_j} \\ \frac{\partial^0_c p_2}{\partial\theta_1} & \frac{\partial^0_c p_2}{\partial\theta_2} & \cdots & \frac{\partial^0_c p_2}{\partial\theta_j} \\ \frac{\partial^0_c p_3}{\partial\theta_1} & \frac{\partial^0_c p_3}{\partial\theta_2} & \cdots & \frac{\partial^0_c p_3}{\partial\theta_j} \end{bmatrix}. \tag{2.9}$$

Joint angles must be found for the manipulator that satisfy the desired tooltip and constrained point positions. These joint angles must be achievable for the physical manipulator. In this formulation, the joint limits will be considered during the inverse kinematics to find a feasible solution. This is done by converting the joint angles to a different variable that saturates as it approaches a joint limit. The variable is then converted back to the original joint space where it is now bounded by the limits.

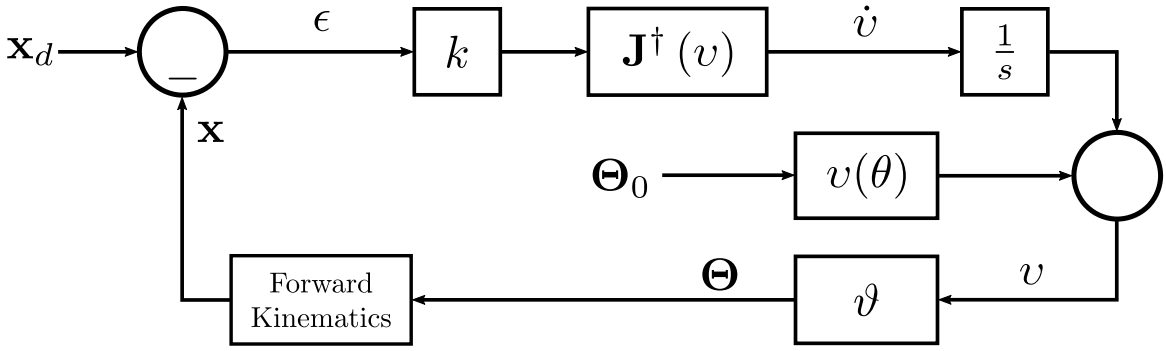
First, let θ_i^u and θ_i^ℓ be the upper and lower limits of joint i respectively. The desired transformation function that converts the joint angles to a new space must be continuously increasing within the open interval $\left(\theta_i^\ell \ \theta_i^u\right)$. The arctangent function is one that meets these criterion. It is used by linearly mapping θ_i from the joint limits $\left(\theta_i^\ell \ \theta_i^u\right)$ to the open interval $\left(-\frac{\pi}{2} \ \frac{\pi}{2}\right)$ by

$$\upsilon_i(\theta_i) = \tan\left(\frac{\pi(2\theta_i - \theta_i^u - \theta_i^\ell)}{2(\theta_i^u - \theta_i^\ell)}\right) \tag{2.10}$$

which is shown in Fig. 2.2(a) on the left-hand side. The inverse of (2.10) converts the



(a) Joint angle saturation function



(b) Constrained inverse kinematics overview

Figure 2.2: The inverse kinematics of the system, left (a): the arctangent function as given in (2.10), where υ_i is the remapped joint angle; and right (a): converting the joint angles back to the joint space (2.11) where they are restricted by the joint limits. (b) shows the control loop of the inverse kinematics, it takes the desired tooltip position $\mathbf{x}_d$ and initial joint angles $\mathbf{\Theta}_0$ as inputs and outputs the corresponding joint angles $\mathbf{\Theta}$ by minimizing error ϵ, k is a constant gain and the pseudo-inverse, constrained Jacobian is $\mathbf{J}^\dagger(\upsilon)$.

transformed joint angle back into physical joint angles θ_i as follows

$$\vartheta_i(\upsilon_i) = \theta_i(\upsilon_i) = \frac{\theta_i^u - \theta_i^\ell}{\pi} \tan^{-1}(\upsilon_i) + \frac{\theta_i^u + \theta_i^\ell}{2}. \tag{2.11}$$

Note that θ_i is now bounded between the joint limits θ_i^ℓ and θ_i^u as shown in Fig. 2.2(a) on the right hand side and is now denoted as ϑ_i.

To compute the inverse kinematics, a new Jacobian has to be constructed based on the modified joint space. The constrained Jacobian $\mathbf{J_c}$ is now computed as the partial derivative of the task space for the transformed joint space as

$$\mathbf{J}_c(\upsilon) = \frac{\partial \mathbf{x}}{\partial \upsilon} = \mathbf{J}(\mathbf{\Theta}) d\vartheta, \tag{2.12}$$

where $\mathbf{J}(\mathbf{\Theta})$ is defined in (2.9) and $d\vartheta$ is a $j \times j$ diagonal matrix given by

$$d\vartheta = \begin{bmatrix} \frac{\partial \vartheta_1}{\partial \upsilon_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial \vartheta_2}{\partial \upsilon_2} & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\partial \vartheta_j}{\partial \upsilon_j} \end{bmatrix}, \tag{2.13}$$

and

$$\frac{\partial \vartheta_i}{\partial \upsilon_i} = \frac{\theta_i^u - \theta_i^\ell}{\pi(1+\upsilon_i^2)}. \tag{2.14}$$

Now the inverse solution of (2.8) can be found considering the change of variable and the joint limits as:

$$\dot{\upsilon} = \mathbf{J}^\dagger(\upsilon)\dot{\mathbf{x}} \tag{2.15}$$

where $\mathbf{J}^\dagger(\upsilon)$ is the pseudo-inverse, constrained, Jacobian matrix defined as

$$\mathbf{J}^\dagger(\upsilon) = \mathbf{J}_c(\upsilon)^T \left[\mathbf{J}_c(\upsilon)\mathbf{J}_c(\upsilon)^T + \mu\mathbf{I}\right]^{-1} \tag{2.16}$$

where $\mathbf{I} \in \mathbb{R}^{6\times 6}$ is an identity matrix and $\mu \in \mathbb{R}^+ << 1$ is a damping constant scalar used to avoid possible discontinuity of the pseudo-inverse at a singular configuration of the manipulator.

Let υ_d be a solution to a desired Cartesian position of the tooltip and constrained point $\mathbf{x}_d \in \mathbb{R}^{6\times 1}$, and ϵ define the error between the desired and actual Cartesian position as

$$\epsilon = \mathbf{x}_d - \mathbf{x}. \tag{2.17}$$

A proportional control law in the form of

$$\dot{\upsilon} = k\mathbf{J}^\dagger(\upsilon)\epsilon \tag{2.18}$$

guarantees that $\epsilon = \mathbf{x}_d - \mathbf{x} \to 0$ (that is $\mathbf{x}_d = \mathbf{x}$), and thus υ approaches υ_d provided that the constant k is positive. This gives a least squared solution to (2.15) when the Jacobian is full rank. The solution ensures that the $\dot{\upsilon}$ is minimized given that

(2.13) increases monotonically in the open interval $(\theta_i^\ell \ \theta_i^u)$. The inverse kinematics process is shown in Fig. 2.2(b), where the desired position and constrained point $\mathbf{x}_d$ is given as an input along with the initial joint angles $\mathbf{\Theta}_0$. The error ϵ, calculated as the difference between the desired position $\mathbf{x}_d$ and the current position $\mathbf{x}$, is multiplied by the constant gain and the pseudo-inverse Jacobian $\mathbf{J}^\dagger$ which gives the change in angle $\dot{v}$. Integrating the change in angle and adding the initial joint values provides the new joint values which just need to be converted to the joint space through (2.11). This loop is repeated until $\epsilon \cong 0$ at which point the joint angles are sent to the manipulator.

The inverse kinematics has been defined based on the forward kinematics while applying the joint limits. The forward kinematics is defined based on joint angles and the physical structure of the manipulator. A 6 DOF (or greater) robotic arm can now be controlled using a 3 DOF haptic device through the inverse kinematics. The next step is to generate haptic feedback based on the tooltip position and expert demonstrations. The haptic feedback can then be used to guide the user through the procedure.

2.3 Haptic Assistance from Demonstrations

To guide a user based on a previously demonstrated trajectory, the trajectory first needs to be collected. While a demonstration is given by one user through the cyber-physical simulator, the tooltip positions are recorded. These points provide information about the path the demonstrator used to accomplish the specified task.

The next step is to generate force feedback based on the demonstration to guide a user; to this end, the well-known concept of potential fields (introduced in [88]) is proposed. The use of potential fields allows the haptic feedback provided to be time-invariant; this is important because it keeps the surgeon in full control of the operation, allowing them to make active adjustments during the procedure such as: correct for possible changes in the environment, deal with issues that did not appear in preoperative imaging, or a change in the position of the kidney, kidney stones, or

the patient. The entry point is, however, not influenced by possible misaligments or changes from preoperative imaging since it is determined by the surgeon during the procedure.

By controlling the interaction impedance, that is, the ratio of applied force to the magnitude of deviation from the reference trajectory, the haptic feedback can be made more or less compliant in certain regions. This allows a human and a robotic agent to act cooperatively towards accomplishing a task such as following a given path.

The assistive forces are calculated based on a non-parametric potential field function [88]. These forces are applied by the haptic device and its spatial impedance is captured by a potential function gradient and curvature.

2.3.1 Potential Force Fields from Trajectory Demonstrations

Potential fields have been proposed to guide robotic manipulators through autonomous tasks. However, these potential fields can also be applied to generate haptic feedback forces during teleoperation. Since the forces are time-invariant and include bounded force ranges, the stability of the robot is guaranteed when in contact with passive environments.

The tooltip position is used as an index to calculate the force that needs to be applied to the user based on the location of the tooltip in 3D space.

All of the sampled reference trajectory data points are concatenated in a single vector $\xi_d \in \mathbb{R}^{3\times N}$, where N is the total number of data points from all reference trajectories. The n^{th} column in ξ_d is given by

$$\xi_d^n = \begin{bmatrix} {}^0_t p_1 & {}^0_t p_2 & {}^0_t p_3 \end{bmatrix}^T. \tag{2.19}$$

The potential field has to be created for a specified workspace whose points in 3D space are denoted $\xi \in \mathbb{R}^{3\times k}$ in which ξ represents the 3D Cartesian coordinate of

a point within the defined workspace boundary and k is the total number of points used to represent the workspace. The higher the k value is, the higher the resolution and smoothness of the potential field, at the cost of computational time and memory usage.

A potential energy field ϕ is generated for demonstration point n as follows:

$$\phi_i^n(\xi) = \phi_0^n + \frac{1}{2} S^n \left(\xi_i - \xi_d^n\right)^T \left(\xi_i - \xi_d^n\right), \quad \forall n \in 1 \dots N, \forall i \in 1 \dots k \tag{2.20}$$

where the stiffness and the initial potential energy for the n^{th} data point, are defined as S^n and ϕ_0^n respectively and ξ_i is the i^{th} entry in ξ.

A Gaussian Kernal is used to create a weighting element for each demonstration point as

$$\omega_i^n(\xi) = e^{-\frac{1}{2(\sigma^n)^2}\left(\xi_i - \xi_d^n\right)^T \left(\xi_i - \xi_d^n\right)} \quad \forall n \in 1 \dots N, \forall i \in 1 \dots k, \tag{2.21}$$

in which, $\sigma^n \in \mathbb{R}^+$ is a smoothing parameter that controls the region of influence for the n^{th} data point. Using weighted sums, the potential field $\mathbf{\Phi} \in \mathbb{R}^{1 \times k}$ at ξ can be computed using element-wise multiplication and division as:

$$\mathbf{\Phi}(\xi) = \frac{\sum_{n=1}^{N} \omega^n(\xi)\, \phi^n(\xi)}{\sum_{n=1}^{N} \omega^n(\xi)}. \tag{2.22}$$

One can also implement a dissipative field in the form of a controllable damper parameter to take into account the robot's speed in the haptic forces. The damping element ψ^n is created for each demonstration point n as follows:

$$\psi^n\left(\dot{\xi}\right) = \mathbf{D}^n \dot{\xi} \tag{2.23}$$

where $\mathbf{D}^n$ is a dissipative gain and $\dot{\xi}$ is the velocity at ξ. These damping elements are then combined similarly to the way that the potential fields are combined to generate a dissipative field $\mathbf{\Psi} \in \mathbb{R}^{1 \times k}$ by computing the weighted sums for element ξ using

element-wise multiplication and division as:

$$\mathbf{\Psi}\left(\xi,\dot{\xi}\right)=\frac{\sum_{n=1}^{N}\omega^{n}\left(\xi\right)\psi^{n}\left(\dot{\xi}\right)}{\sum_{n=1}^{N}\omega^{n}\left(\xi\right)}. \tag{2.24}$$

Finally, the dissipative field and the potential field are combined to create a force field that will be used to apply a force that guides the user along a desired trajectory $\xi_{\mathbf{d}}$. The force field is generated by subtracting the dissipative field from the negative gradient of the potential field as follows:

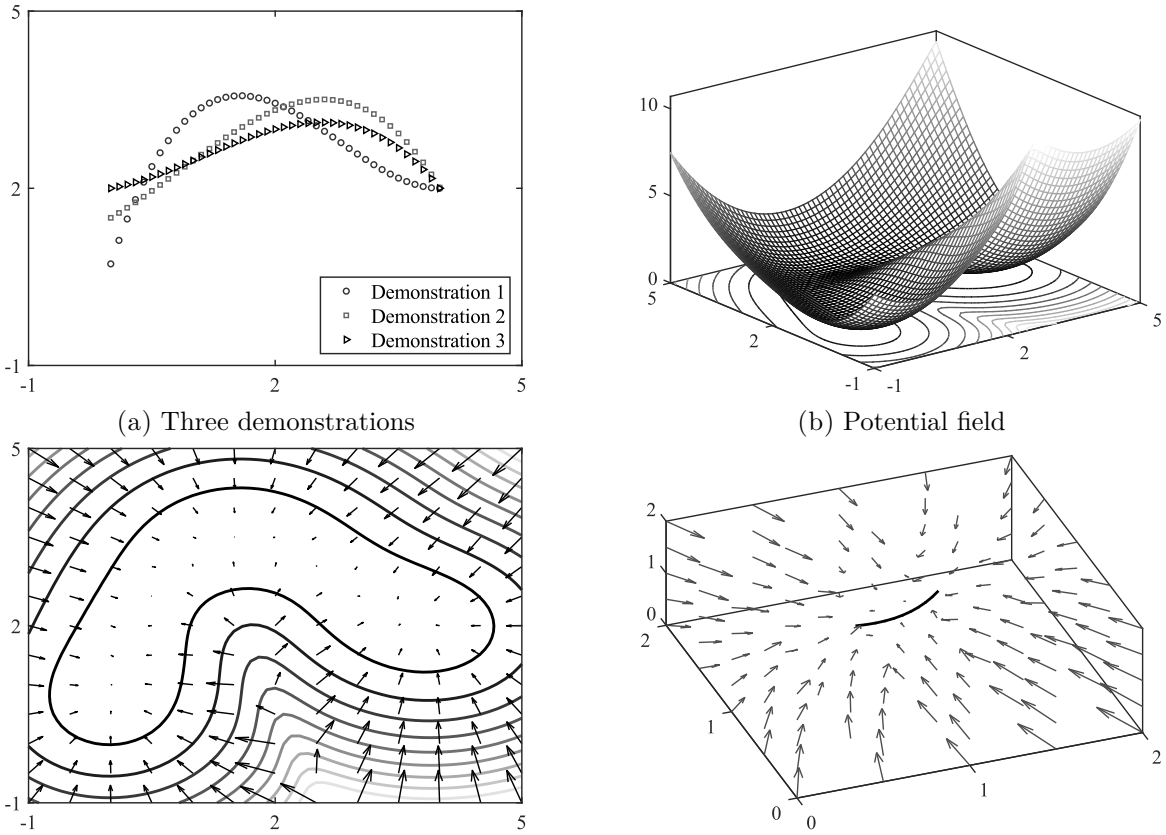
$$\mathbf{F}=-\nabla\mathbf{\Phi}\left(\xi\right)-\mathbf{\Psi}\left(\xi,\dot{\xi}\right) \tag{2.25}$$

where ∇ is the gradient. In Fig. 2.3(b) an example of a potential field $\mathbf{\Psi}$ is shown, while in Fig. 2.3(c)&(d) the arrows are the force $\mathbf{F}$. The force field will now be used as a lookup table when providing haptic assistance.

The reference trajectories are obtained from demonstrations given by experts. These demonstration points, however, are not evenly distributed in 3D space. This means that the sample points may be clustered together in one area while being spread apart widely in another. This unevenness can lead to some areas being too heavily weighted. Because of this, the demonstrations should be re-sampled so that all of the points are evenly spread apart. Alternatively, an optimization method can be used to assign the ideal value of stiffness $\mathbf{S}$ and smoothing parameter σ to each demonstration to generate a smooth potential field such as in [88].

Since each demonstration point is composed of three Cartesian coordinates, the potential field will exist in the fourth dimension. This increase in dimensionality is shown in Fig. 2.3(a)&(b), part (a) shows several two-dimensional demonstration points while part (b) shows the potential field that was built for these sample demonstrations; note that the potential field is three dimensional.

After the gradient is taken of the potential field the dimensionality returns to that of the demonstrations, as shown in 2.3(c)&(d) the arrows representing the gradient



(a) Three demonstrations

(b) Potential field

(c) Gradient of Potential Field

(d) 3D example

Figure 2.3: Potential field construction starting with 2D reference trajectories (a) shows three sample demonstrations that may be used as reference trajectories when constructing the potential field, in (b) the potential field for the three reference trajectories is shown along with its projection onto the x-y plane at the bottom, the z-axis here is the magnitude of potential energy. Plot (c) represents the gradient of the potential field. The length of the arrow is determined by the magnitude of the gradient at that point. Plot (d) shows an example of a three dimensional reference trajectory (the black line in the centre), where the arrows again represent the gradient that is taken of the potential field, these describe the haptic feedback for the given reference trajectory.

of the potential field, are all two-dimensional for the 2D demonstrations and three-dimensional for the 3D demonstrations.

2.3.2 Using potential fields for haptic assistance in PCNL

The haptic device receives two inputs, one from the user and the other from the force field; this is shown in Fig. 2.4(b). The force feedback is applied to the user, guiding them along the desired trajectory. The 3 DOF haptic device outputs three Cartesian coordinates to the inverse kinematics which determine the six joint angles configuration to reach the desired location while constraining the entry point. Thus, the 3 DOF haptic device is can be used to control the 6 DOF robotic manipulator. Given a set of reference trajectories, the potential field forces can now be used to provide haptic assistance based on the tooltip location. As the user controls the robot's tooltip using a haptic device, they receive feedback based on its location to help direct them towards the optimal path. The force field does not control the robot directly and a human is kept in full control of the actual robot position.

2.4 Experimental Validation

The experimental setup is shown in Fig. 2.4(a). The setup consists of a 3-DOF (degree of freedom) haptic device (the Novint Falcon); a 6-DOF robotic manipulator, the Meca500R from Mecademic (Montreal, CA); and a phantom kidney kindly provided by Marion Surgical. The outside of the phantom kidney consists of a hard plastic casing that is open at the top and has a clear acrylic pane in the front, while the kidney consists of soft silicone. At the centre front of the kidney there is an opening through the silicone from the top into the kidney, through which users are expected to enter the kidney model. The resulting path is shown in Fig. 2.4(a), between points ⑥ ⑦. The same kidney model was used for all experimental trials. The modified Denavit-Hartenberg parameters used in the inverse kinematics can be found in Table 2.1 and the joint limits are summarized in Table 2.2.

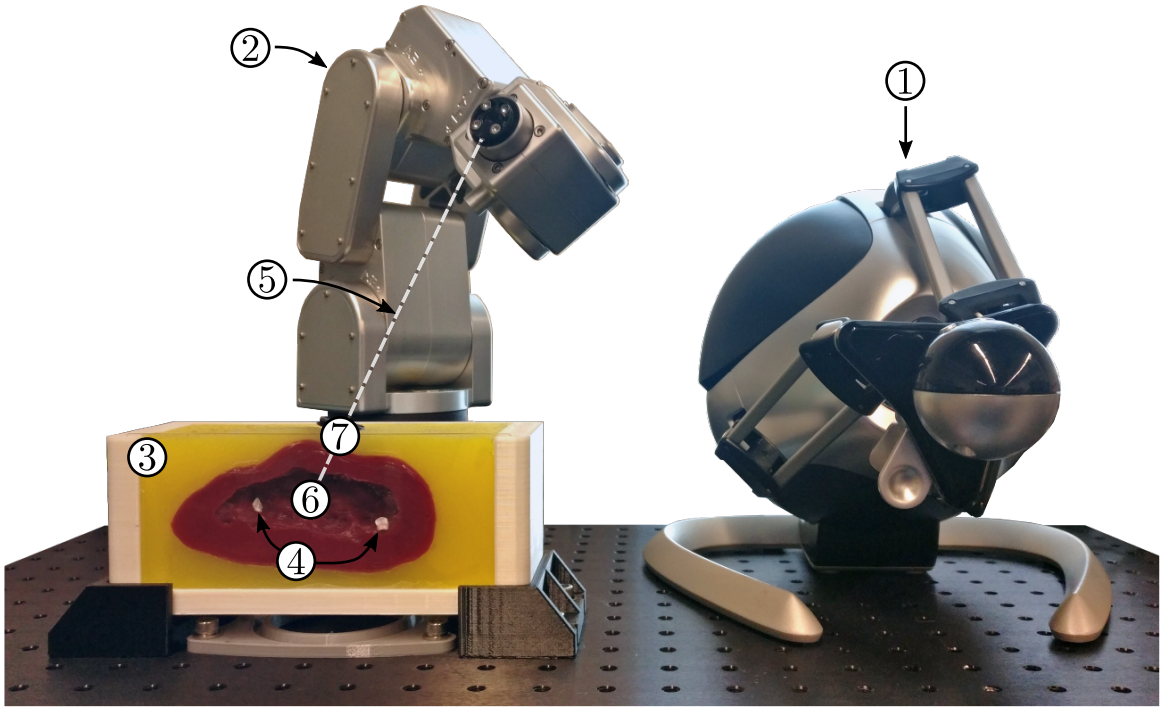
Table 2.1: Modified Denavit-Hartenberg Parameters for Meca500

i	1	2	3	4	5	6	t	c	**unit**
α_{i-1}	0	-90	0	-90	90	-90	-90	-90	deg
a_{i-1}	0	0	135	38	0	0	0	0	mm
d_i	135	0	0	120	0	70	$-L$	$-\ell$	mm
θ_i	θ_1	θ_2-90	θ_3	θ_4	θ_5	θ_6	0	0	deg

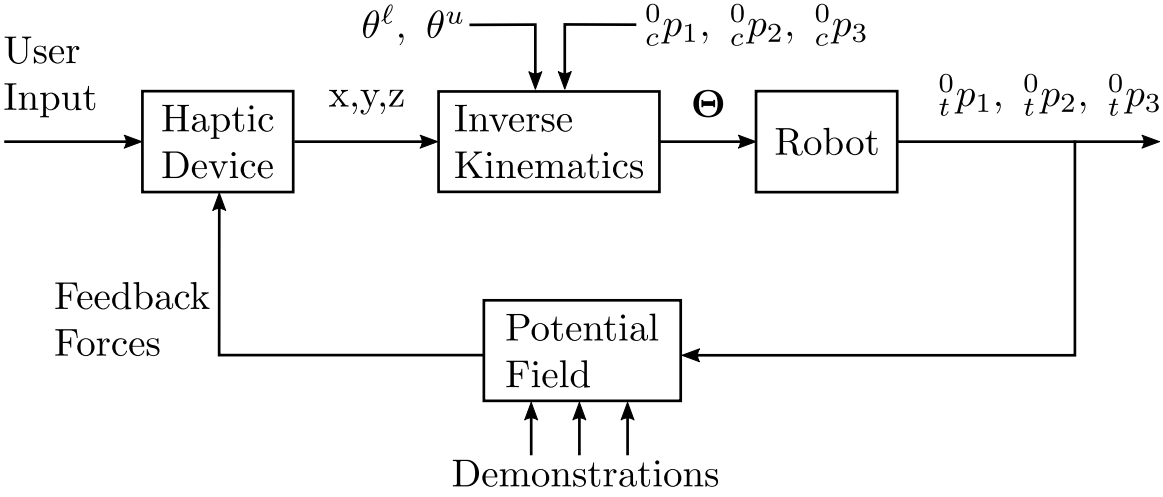
The control loop used during experimental trials with haptic feedback can be seen in Fig. 2.4(b); for those trials without haptic feedback there is simply no feedback loop. During the user trials, the haptic device sends its 3D position to the computer which then processes it and uses it as the desired tooltip position in the robot's inverse kinematics. A demonstration of the desired tool path is first executed where the 3D Cartesian coordinates of the tooltip are collected. The tooltip information is determined by applying the forward kinematics to the collected joint angles; this assumes that the tool is perfectly rigid. The data from the demonstration is used to generate the potential field that will be used to provide the haptic feedback. A user familiar with the system provides the initial demonstration. Once the potential field has been constructed, the robot tooltip position is input into the potential field function to obtain the required haptic assistance force to be applied to the haptic device. The computer uses an *Intel i7* processor and *GeForce RTX 2080* GPU. Communication between the computer, the haptic device, and the robot is handled by the Robot Operating System (ROS).

2.4.1 Joint Limit Experimental Validation

An experiment is performed to validate the effectiveness of the proposed constrained inverse kinematics. Two trials need to be conducted to observe the effects of the joint limits. The first only uses the default joint limits of the manipulator (see Table 2.2, *Default* rows), these limits are determined by the manufacturer based on avoiding collisions or the rotational limit of the motors used when constructing the manipulator. During the second experiment, joint 1 is further limited (see Table 2.2, *Modified* rows) from -3.05 to 3.05 rad to -1 to 0.4 rad to demonstrate the response of the inverse



(a) Experimental set-up used to perform the user trials with the robotic arm, the haptic device and the phantom kidney



(b) Control loop used during experiments

Figure 2.4: Experimental setup (a), does not depict a clinical setting, rather represents the equipment and relative position used during the user trials including ① the haptic device, ② the robot, ③ the kidney model, ④ phantom kidney stones in the kidney model, ⑤ the tool used during experiments, ⑥ the tooltip, ⑦ the constrained point along the tool; (b) shows a general loop of how the system operates and provides haptic feedback to the user based on the robot's position.

Table 2.2: Robot joint limits given in radians for the joint limit validation experiments

joint i	1	2	3	4	5	6
Default θ_i^ℓ	-3.05	-1.22	-2.36	-2.97	-2.01	$-\pi$
Default θ_i^u	3.05	1.57	1.22	2.97	2.01	π
Modified θ_i^ℓ	-1.00	-1.22	-2.36	-2.97	-2.01	$-\pi$
Modified θ_i^u	0.4	1.57	1.22	2.97	2.01	π

kinematics as a particular joint approaches its limit.

The tooltip reference trajectory during these experiments is a path defined such that under the *Default* joint constraints the joints do not reach their limits. The trajectory can be found in the bottom right of Fig. 2.5.

Fig. 2.5 shows that as a joint approaches its limit, it begins to saturate. To achieve the desired position, the inverse kinematics finds an alternative joint configuration to compensate for the limited joint. It can be noted that the constrained and unconstrained angles are initially the same, and as time goes on the angles of each joint begin to change relative to the original; thus, compensating for the limitation imposed on joint 1.

2.4.2 Constrained Workspace Evaluation

The second set of evaluations simulated the reachable workspace inside the kidney model. The constrained tooltip workspace was determined by considering the phantom kidney volume and determining which points within it are achievable when the RCM is set at the entry point. The previously validated inverse kinematics was used to attempt to reach points within the phantom kidney volume. This data was used to create the tooltip workspace shown in Fig. 2.6.

The workspace shown considers a tooltip position in the robot's workspace when the angle between the tool and the surface of the kidney is no less than 20°. All of the axes and coordinates are given relative to the robot's reference frame which considers the origin to be at the centre of the robot's base.

As can be seen, most of the kidney volume can be accessed through the same entry

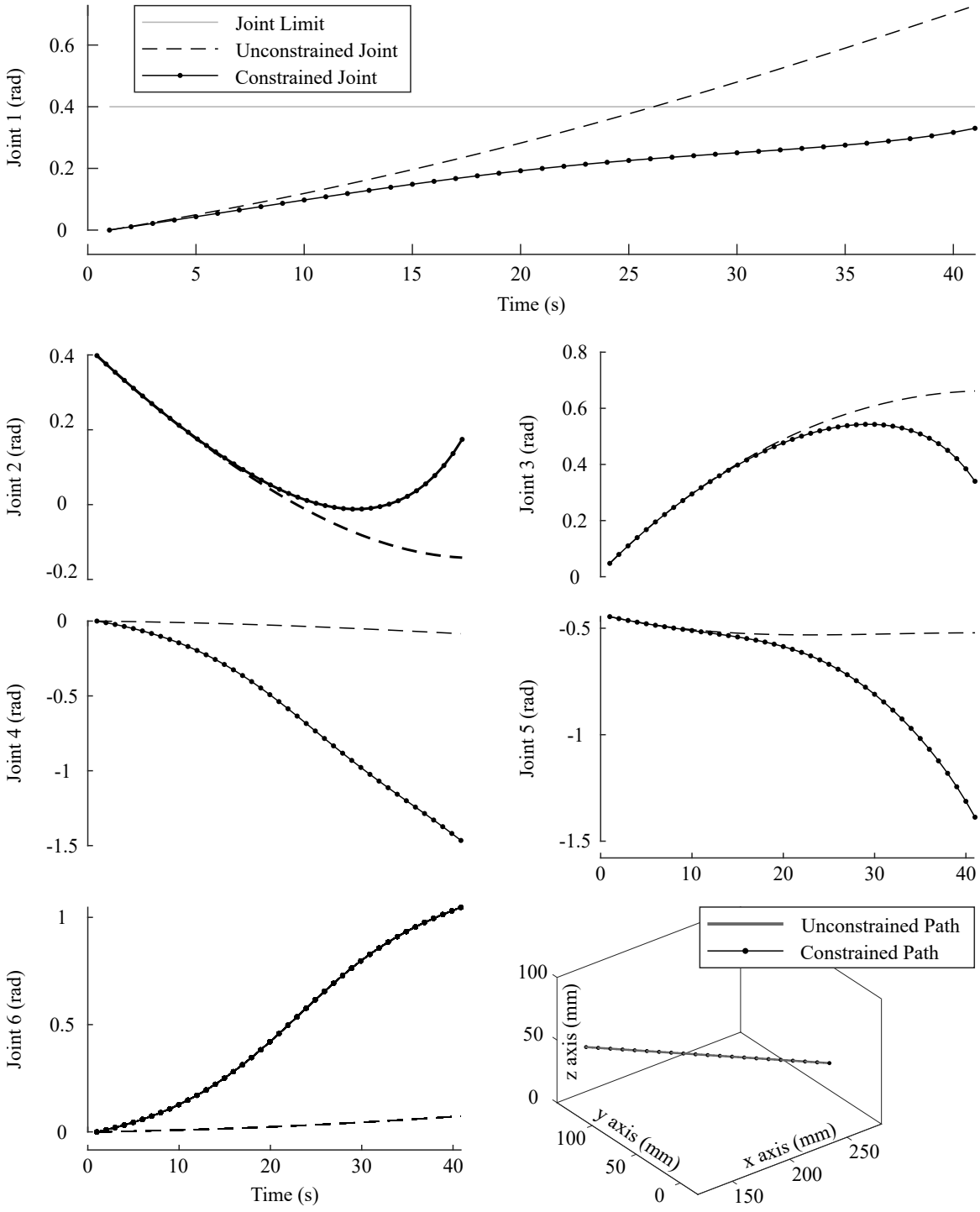
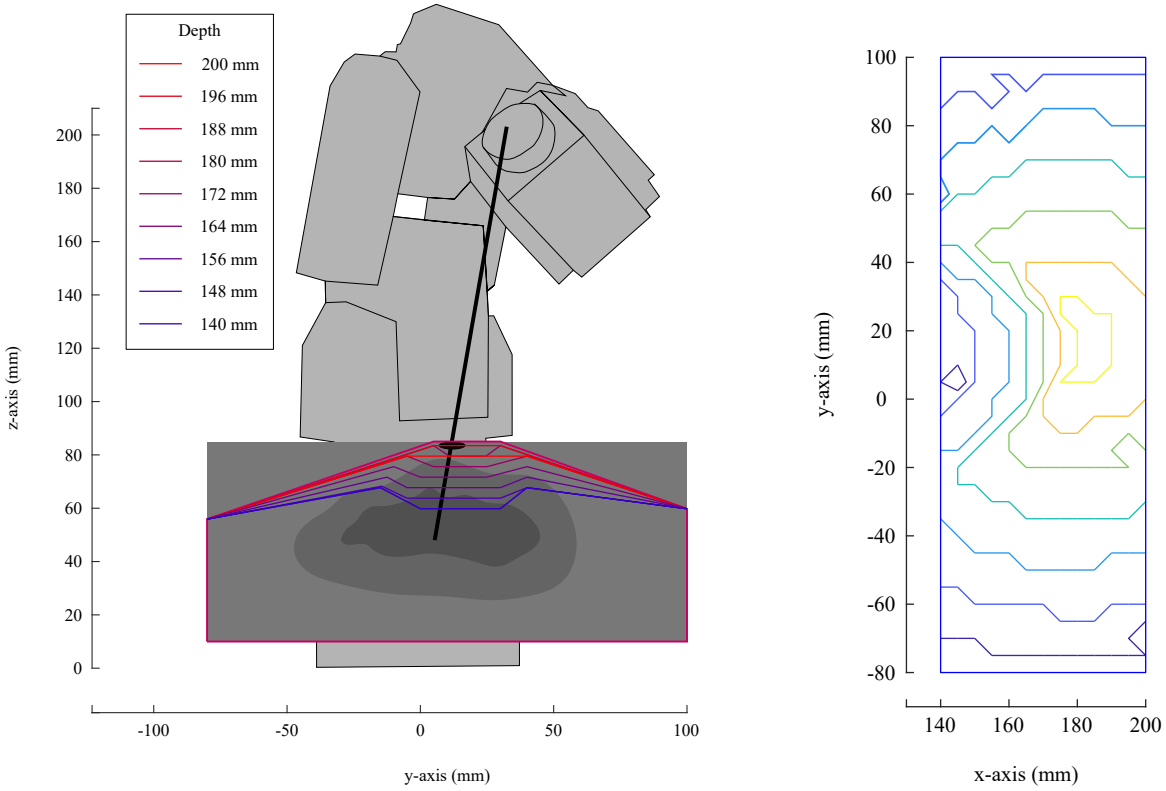


Figure 2.5: The results from the constrained inverse kinematics. In the 2D graphs, a dashed line depicts the joint angle when no additional joint constraints were added, a dotted line depicts the joint angles after an additional joint limit was applied to joint 1, and a solid grey line shows the added joint constraint. In the 3D plot, the trajectory of the experiment is shown.



(a) tooltip constrained workspace, considering the phantom-kidney volume.

(b) Contour plot of the top surface of the workspace.

Figure 2.6: The above plots show the workspace of the constrained tool within the phantom kidney. On the left, plot (a), is a plot of the outline of the workspace (in red), with contours (ranging from red to blue) included for the top surface as it changes depending on depth. While plot (b) depicts the contour of the workspace surface in the x-y plane.

point. The peak of the workspace (yellow region of the contour plot) is the location of the entry point. There is an additional concavity in the workspace behind the entry point; this is due to the kinematic constraints of the robot.

2.4.3 Experimental Scenarios for User Trials

Three different experiments were performed to validate the functionality of the system. They were done both with and without haptic feedback. Participants have a full view of the phantom kidney and robot arm during all of the experimental trials. There were sixteen participants, 12 males, 4 females ranging in age from 18 to 31 with the average age being 24. No exclusion criteria are used to disqualify participants. The participants also do not have prior knowledge of the experimental setup or procedure.

They are separated into 3 groups. The participants in each group were exclusive to that group and only provided data for their assigned group. Each participant completes 5 trials in total, the first 3 trials are considered training trials while the last 2 trials are used to evaluate the performance of each group. Participants performed all 5 of their trials consecutively with no significant breaks in between trials. Every user completed the same scenario in each trial; the goal in each trial is to reach a particular kidney stone which is in the same location for all experimental trials. The user who provided the demonstration data for the potential field was not a participant in the actual tests; they only provided the demonstration data.

Group 1 (6 participants): The first group performed all 5 trials without any haptic assistance. The data collected from the last two of these trials is the control data to compare the other groups against.

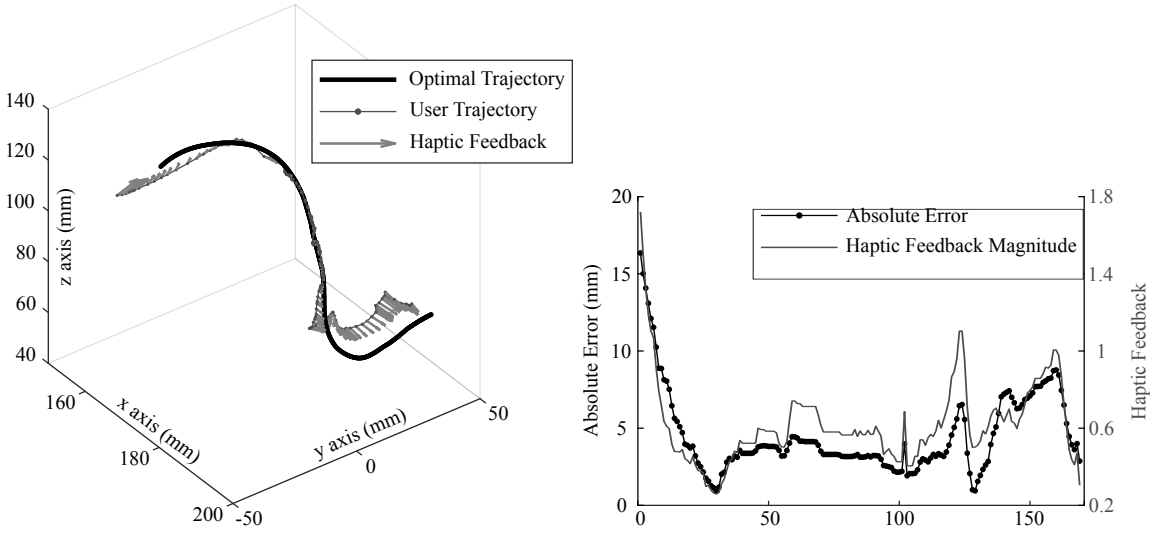
Group 2 (5 participants): The second group had haptic feedback during all 5 trials. This provided information on the effectiveness of haptic assistance when compared to those without it.

Group 3 (5 participants): Finally, the third group completed their 3 training rounds with the haptic assistance and then two evaluation trials without assistance. This was done to evaluate how well participants learned from the haptic assistance and to analyze if there were any lasting improvements in skill compared to those without haptic feedback.

The stiffness of the potential field used to derive the haptic feedback was kept constant during all trials.

2.4.4 Experimental Procedure for User Trials

Users are expected to use the haptic device to control the six DOF robotic manipulator. A sheath located on the top surface of the kidney model is the entry point into the tissue ⑦ in Fig. 2.4(a). Users were requested to manipulate the tooltip through the sheath towards the phantom kidney stones ④ in Fig. 2.4(a). The procedure had



(a) 3D plot of a user trial with haptic feedback. (b) User error and the haptic feedback.

Figure 2.7: Plot (a) provides an example of haptic feedback provided to the user during one of their user trials; the optimal trajectory is shown as the solid black line, the user's trajectory is the dark grey dotted line, and the haptic feedback are light grey arrows. The bottom plot (b) exhibits the absolute error between the user's trajectory and the optimal path as the dotted black line (its y-axis is on the left), while the magnitude of the haptic feedback is given as a grey line (its y-axis is on the right).

3 phases:

1. The tool was initially located outside of the kidney model. During this phase, the constrained point was just above the tooltip and was moved along with it such that the tool remains vertical. The 3D position of the haptic device controlled the position of the tooltip;

2. Once the user had positioned the tooltip inside the sheath, the user pressed a button and the tooltip location was recorded as the entry point in the tissue. The constrained point was then fixed and assigned the recorded entry point position.

3. The user further manoeuvred the tooltip towards the kidney stones. The 3D position of the haptic device controlled the 3D position of the tooltip while the constrained inverse kinematics ensured that the tool shaft passed through the constrained point.

An example of the haptic feedback provided to a user is shown in Fig. 2.7. In Fig.

2.7(a), the haptic feedback (grey arrows) is always directed towards the optimal path (the black solid line). Plot (b) shows the absolute error between the user's trajectory and the optimal path.

The magnitude of the applied haptic feedback was roughly proportional to the error. Due to how the potential field was constructed, with a slight pull along the optimal trajectory and varying spatial stiffness, the haptic feedback was similar to but not perfectly proportional to the user's error. When the user is in phase 1, moving towards the entry point, the haptic feedback provided should feel like a funnel due to the stiffness of the potential field being kept low and linearly increased up to the entry point. As the stiffness increases, larger force feedback is felt when the user diverges from the ideal path. At the entry point and through the flesh of the kidney model, the stiffness was kept constant; this is the highest stiffness used when generating the force feedback. While inside the kidney, phase 3, the feedback resembles a tube-like structure with relatively firm walls. The stiffness is linearly decreased somewhat once the tooltip has passed through the sheath, then the stiffness is constant inside the kidney.

2.4.5 Assessment Criteria for User Trials

Six different metrics are used to compare the experimental results between groups of participants. These metrics are largely based on those described in [89], which uses a set of evaluation criteria during teleoperation to assess a user's skill level for robot assisted minimally invasive surgery.

To begin, let the collected data points of the tooltip be $\xi_t \in \mathbb{R}^{3\times M}$ where M is the total number of collected data points for one trial, now the m^{th} column can be represented by:

$$\xi_t^m = \begin{bmatrix} {}^0_t p_1 & {}^0_t p_2 & {}^0_t p_3 \end{bmatrix}^T. \tag{2.26}$$

The 5 assessment criteria can now be defined as:

Criterion 1 is the time to complete task, t_{tot}, *i.e.*, the total time needed to reach the

goal;

Criterion 2 is the distance the tooltip travels, *i.e.*, the path length ξ_L, given by:

$$\xi_L = \sum_{m=2}^{M} |\xi_t^m - \xi_t^{m-1}|, \tag{2.27}$$

where the magnitude of a vector $\left[\xi_t^m - \xi_t^{m-1}\right]$ is given as:

$$|\xi_t^m - \xi_t^{m-1}| = \sqrt{\left(\xi_t^m - \xi_t^{m-1}\right)^T \left(\xi_t^m - \xi_t^{m-1}\right)}. \tag{2.28}$$

The magnitude of the tooltip's velocity $\nu \in \mathbb{R}^{1\times M}$ at every point can be computed as the first time derivative of ξ_t by:

$$\nu_m = \left| \frac{\xi_t^{m+1} - \xi_t^{m-1}}{2\delta t} \right| \ \forall m \in 1 \ldots M, \tag{2.29}$$

where δt is the time step from one data-point to the next, assuming the time step is constant for all data-points.

Criterion 3 can now be defined as the average velocity ν_μ, i.e.:

$$\nu_\mu = \overline{\nu} = \frac{1}{M} \sum_{m=1}^{M} \nu_m. \tag{2.30}$$

The magnitude of motion acceleration $a \in \mathbb{R}^{1\times M}$ can be defined similar to velocity with the main difference being that acceleration requires the second time derivative of ξ_t as opposed to the first time derivative. Therefore, the vector of motion acceleration magnitudes can be computed by:

$$a_m = \left| \frac{\nu_{m+1} - \nu_{m-1}}{2\delta t} \right| \ \forall m \in 1 \ldots M, \tag{2.31}$$

Criterion 4 is defined as the mean acceleration, that is:

$$a_\mu = \overline{a} = \frac{1}{M} \sum_{m=1}^{M} a_m \tag{2.32}$$

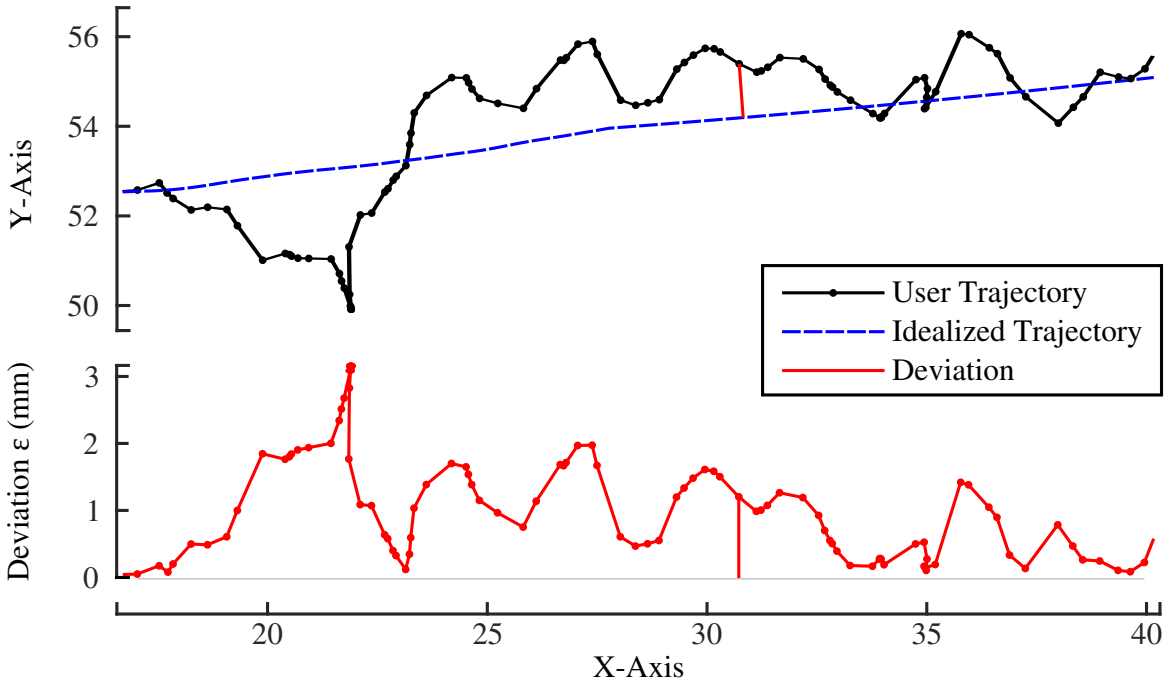


Figure 2.8: An example of a user's actual trajectory compared with their idealized trajectory. The dashed blue line is the idealized trajectory while the dotted black line is a user's actual trajectory during one trial.

Criterion 5 is related to the user's deviation from an idealized version of their trajectory. The idealized version ξ_I removes small redundancies and tremors leaving a close approximation to what the user intended to do. It is created by applying the Savitzky-Golay filter, which is a digital filter that employs a moving window and fits a polynomial curve to sets of data points within the window in order to filter out noise. The Savitzky-Golay filter is employed here to generate a smooth version of a user's trajectory so deviation from their intended path can be calculated. An example of an filtered trajectory can be seen in Fig. 2.8, where the smooth blue line is the ideal smoothed trajectory based on the actual trajectory (the dotted black line).

Using the idealized version of the path the magnitude of the user's deviation $\varepsilon \in \mathbb{R}^{1\times M}$ for every point is defined as:

$$\varepsilon_m = \left|\xi_t^m - \xi_I^i\right| \quad \forall m \in 1 \dots M, \tag{2.33}$$

where ξ_I^i is the closest point along ξ_I. The average deviation is given by:

$$\varepsilon_\mu = \overline{\varepsilon} = \frac{1}{M} \sum_{m=1}^{M} \varepsilon_m. \tag{2.34}$$

Criterion 6 is the targeting accuracy for each user. This considers the approximate location of the phantom calculi ξ_c and finds the closest data point from the user trial. As such the targeting accuracy TA can be defined as:

$$TA = \left| \xi_c - \xi_t^k \right|, \tag{2.35}$$

where $\exists k \in M \ni \left| \xi_c - \xi_t^k \right| \leq |\xi_c - \xi_t^m| \, \forall m \in M$.

These criteria can now be used to evaluate the performance of the three different experimental groups to determine the effectiveness of the haptic feedback on performance, skill acquisition, and path consistency between experimental trials.

2.5 Results and Discussion

The results from the last two trials of every user can be found in Fig. 2.9 as box plots. At the top left of Fig. 2.9 is the total time to complete each trial, the red lines indicate the median time for each group to complete their evaluation trials. The blue boxes span from the first quartile to the third quartile and indicate a median of the lower and upper half of the dataset respectively, this box describes the range of most participants times. The whiskers on each box reach the maximum and minimum times that fall within the interquartile range. This first evaluation criterion does not have outliers which would be marked by a red plus sign. Outlying datapoints are those outside of the interquartile range, the boxplots for average acceleration and targeting accuracy both have an outlier in group 3.

Second from the top on the left, is the path length as determined by (2.27). At the bottom left of the figure is (2.30) which is the average velocity per trial between the user's actual motion and the idealized version of their trajectory. At the top

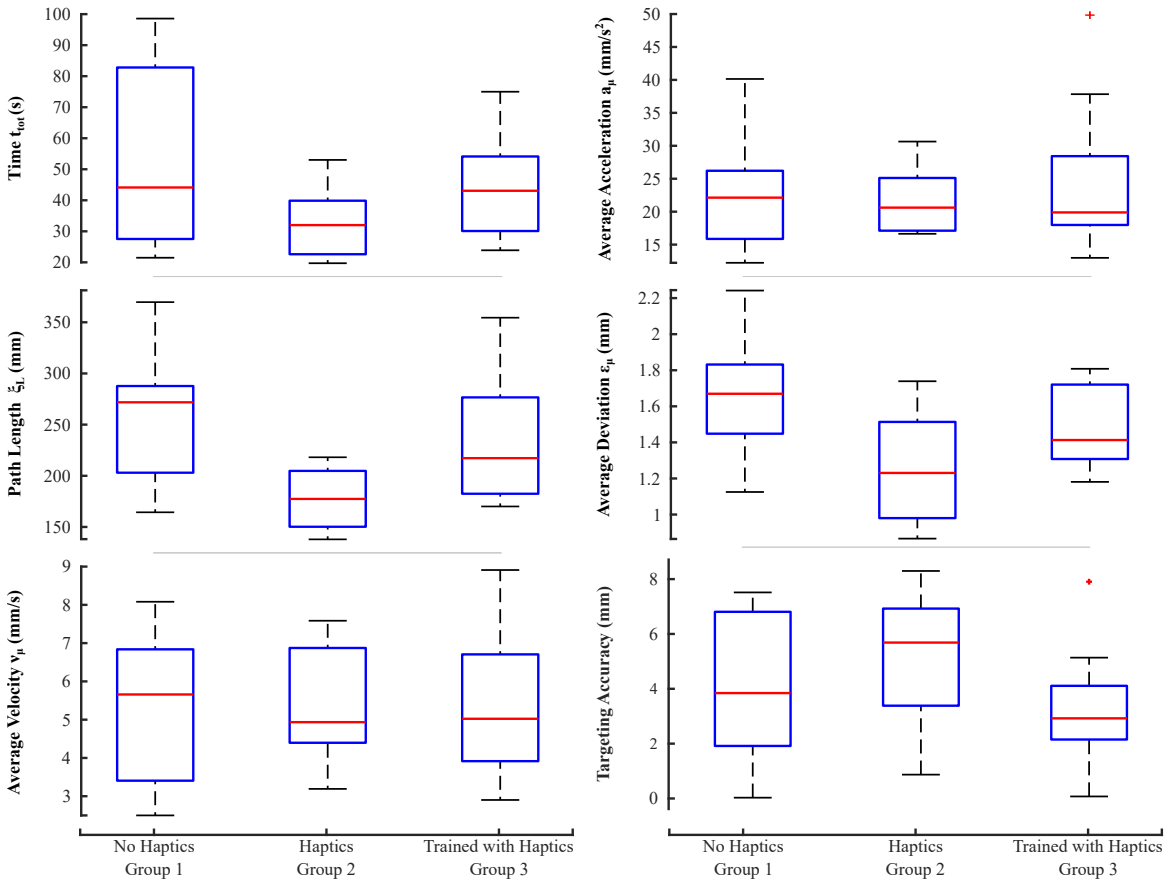


Figure 2.9: Box plots of the experimental results for the last two trials of every user. The data is separated into the three groups of users, those who received no haptic assistance ('No Haptics'), those who completed it with haptic assistance ('Haptics'), and those who completed their training rounds with haptic assistance but the evaluation rounds without haptic assistance ('Trained with Haptics').

right of the figure is the average acceleration (2.32) per trial. Average deviation is the centre plot on the right in Fig. 2.9, which was calculated in (2.34). Lastly, the targeting accuracy is shown on the bottom right; the results are similar between each of the three groups. However, more consistent targeting was achieved by group 3, this suggests that the training with haptics leads to a much better understanding of the target position and the particular motion required to reach it. It should be noted that while the tool was considered to be rigid; it did bend when approaching the phantom calculi, which can lead to less accurate results when comparing the tooltip location to a static location in space. The average and standard deviation for each assessment criterion per group are summarized in Table 2.3. For each evaluation criteria the table

Table 2.3: Average and standard deviation for each assessment criterion and targeting accuracy (denoted TA)

	unit	Group 1	Group 2	Group 3
t_{tot}	sec	52.07 ± 28.09	32.98 ± 10.89	44.49 ± 16.83
ξ_L	mm	257.65 ± 64.92	178.32 ± 29.81	232.97 ± 58.84
ν_μ	mm/s	5.28 ± 1.91	5.32 ± 1.49	5.31 ± 2.01
a_μ	mm/s^2	22.58 ± 8.65	21.66 ± 5.12	24.53 ± 11.46
ε_μ	mm	1.63 ± 0.32	1.28 ± 0.33	1.48 ± 0.23
TA	mm	1.13 ± 3.03	2.22 ± 1.26	0.72 ± 0.94

contains the mean for the respective group along with its standard deviation.

2.5.1 Discussion

The experimental results obtained from a total of 32 trials and 3 independent groups reveal that, overall, users who receive haptic assistance perform better on average according to the assessment criteria specified earlier. The results further suggest that haptics is an effective way of developing surgical skills in the context of teleoperated PCNL.

The most evident benefit of haptic assistance can be seen in the time taken to complete the procedure as shown in Fig. 2.9. The addition of haptic feedback (group 2) made all trials more consistent when compared to those without haptic feedback (group 1). This claim is further supported by the fact that group 3 outperformed group 1. The results indicate that the subjects from group 3, who are trained with haptic feedback and are evaluated without it, did better than those who never experienced the haptic assistance. This suggests that group 3 demonstrates the degree to which users can learn from the haptic feedback.

Path length is another indicator of the effectiveness of the haptic assistance on user performance. Both the mean and the variability of the path length decrease with haptic assistance. Additionally, the lower median and variability of group 3 compared to group 1 is recurrent evidence of learning from the haptic assistance.

The mean deviation from the idealized path for each user trial follows the same trends as described for time and path length. A lower average deviation and range suggests that fewer redundant motions occurred during the procedure. It further implies that the path taken is smoother compared to those without haptic assistance or training (such as group 1). Finally, the average acceleration and velocity decrease with haptic assistance and training, which is another indication of smoother trajectories and motion during the procedure.

The targeting accuracy criterion indicates the effectiveness of learning from the haptic feedback. Once the haptic feedback was removed users were able to reach the phantom kidney stone with very consistent results. Group 2, who had haptics during every trial, did manage to target the kidney stone with slightly higher accuracy than the participants in group 1. Although participants in group 1 have a lower median targeting accuracy, they have a standard deviation larger than double the standard deviation of group 2, as can be seen in table 2.3.

2.6 Concluding Remarks

Although PCNL is the leading procedure used for the management of urinary calculi for large and irregularly shaped urinary stones, it remains a challenging procedure to learn and perform efficiently. A fully automated system would pose an increased risk to patients due to unmodelled aspects of the surgical environment. In addition, they require accurate alignment of the optimal path to patient anatomy. This challenge is considered in [58], they found a 15.8 mm position error and 4.12° orientation error from alignment. This chapter introduces a collaborative human-robot teleoperative training framework to assist the surgeon and teach surgical skills. It has two integrant parts: The constrained inverse kinematics that decouples tooltip orientation and position using a remote centre of motion, and haptic assistance from past demonstration(s) based on data collected from the slave. This way, uncertainties and unmodelled dynamics of the environment are accounted for and the haptic feedback reflects the same conditions faced by the experts demonstrating the procedure.

Current literature primarily evaluates performance in virtual environments, while the experiments performed here consider a cyber-physical environment. Sixteen participants took part in a total of five trials each. The participants were each placed in one of three groups; the first group received no haptic assistance, the second group received haptic assistance for all trials, and the third group is trained using haptic assistance for the first three trials and completed the last two evaluation trials without any haptic assistance.

Constraining the entry point in the kidney effectively allowed users to use a 3-DOF device for a procedure that required 6-DOF. This allows the workload to be shared between the surgeon and the robot while preserving the surgeon's control over the tool-path. This reduces the cognitive load on the surgeon during the procedure which makes it possible for them to be more receptive towards haptic feedback [90].

It was found that the analytical joint limits functioned as anticipated; as a joint approaches a limit it begins to saturate and the inverse kinematics returns an alternative joint configuration to reach the desired point. Experimental results obtained with haptic feedback consistently have shorter path lengths, shorter time for the procedure, a lower average deviation, and more consistent velocity and acceleration during the procedure. The group trained with haptic assistance performed better than those without, however, not as well as those who had haptic feedback for all trials. By most evaluation metrics they had a worse median value and larger variability; these problems were even more exaggerated in the group who never experienced any haptic feedback, thereby suggesting that haptic feedback reduces redundant motions during a procedure and can help teach inexperienced users.

While haptic feedback is beneficial for training purposes, automating the subtask of gaining kidney access or tool steering with a robotic agent can provide necessary precision and control to the procedure. An integral component in the autonomous execution of a task is in the trajectory required for execution. This is particularly significant when considering surgical procedures such as PCNL.

Chapter 3

Multi-objective Path Planning via Discrete B-spline Interpolation

AS DESCRIBED in **Chapter 1**, inadequate tool steering leads to a higher chance of complications and increased recovery times due to tissue puncture or damage. These complications, while not fatal, lead to longer recovery times and excessive bleeding [17–20]. Thus, partially or fully automating kidney access through robotic assistance can improve the procedure success rates and increase patient safety [91, 92].

Path planning is integral to any automated surgical procedure. A carefully planned path allows the robot to perform its task effectively and precisely with minimal risks to the outcome of a procedure. There are a few common traits of a desirable path regardless of application, such as being relatively short, smooth, and avoiding obstacles. During PCNL (percutaneous nephrolithotomy) specifically, the ideal path is the one that provides access to the kidney while avoiding surrounding tissue and other anatomical structures that are at risk of unnecessary puncture due to their proximity to the kidney. It is also desirable to reduce tissue damage, which can be quantified as reducing the overall tissue displacement for a planned trajectory. While the literature for path planning in robotic surgery is extensive, little research has been done for PCNL path planning. PCNL differs from some of these procedures including in the assumptions made during the planning process such as that the tool is rigid throughout the procedure and that the tool-tissue interaction mostly occurs along the entry path [26].

To address these issues, a novel multi-objective path planner for autonomous kidney stone access during PCNL is presented in this chapter. NSGA-II is proposed to plan a B-spline curve through a set of anchor points that represents the desired tooltip trajectory from the entry point to the stone location. The multi-objective non-dominated sorting genetic algorithm II (NSGA-II) is proposed to plan a B-spline curve that will be used as the tool trajectory during PCNL. Here, NSGA-II is used to determine the anchor point locations for a uniform B-spline curve. The optimal path minimizes path length, tissue potential energy due to tissue compression, and path smoothness while maximizing the distance to obstacles. The outputs of the algorithm are the optimal

spline coefficients.

This chapter is organized as follows. Section 3.1 provides a brief overview of documented path planning algorithms in robotic surgery and section 3.2 introduces the B-splines and NSGA-II optimizer. The multi-objective optimization is first evaluated through a set of simulations in Sec. 3.3 and demonstrates the effectiveness of the path planning for a variety of goal locations within the kidney. The simulation shows that the planned trajectories result in minimal tissue deformation, are relatively short and smooth, and do not collide with the internal kidney structures. Finally, the trajectories are put into practice through physical experiments to demonstrate their applicability, this is discussed further in Sec. 3.4.

3.1 PCNL Optimal Path Planning

A wide variety of path planning algorithms have been applied to path planning in robotic surgery. These algorithms include several classical methods as well as evolutionary approaches. Some approaches are designed for real-time path planning where only the tool's immediate surroundings and the desired goal location are required. These include artificial potential fields, collision cones, vector field histograms, and dynamic windows [93,94]. Real-time path planners are fast enough to provide quick responses during real world scenarios such as autonomous vehicle path planning. They are computationally effective, however, they are best suited to local path planning problems since they do not necessarily return an optimal path, but rather a *feasible path* that is primarily concerned with obstacle avoidance. Other planning methods, such as roadmap and grid-based approaches, are used to describe the environment as a whole, including obstacles and collision-free paths, and are frequently used in conjunction with additional search methods such as A^* or an evolutionary approach to return an *ideal path* [93, 94].

Evolutionary approaches to path planning are fast and often computationally efficient. For multi-objective optimization problems, metaheuristic approaches are gen-

erally preferred as they produce a better spread along the Pareto front and more non-dominated solutions compared with classical optimization approaches. Evolutionary optimizers tend to search the variable and objective spaces more than classical approaches, although this is very dependant on the algorithms in use. Evolutionary optimizers include a wide variety of swarm optimizers such as particle swarm optimization, ant colony optimization, bacterial foraging, bee colony optimization, and several lesser-known swarm algorithms [93,94]. Alternatively, genetic algorithms, fuzzy logic, and neural networks can also be used for path planning.

Surgical path planning must take into account a variety of objectives, thus, a multi-objective optimizer is a natural choice for the task of planning an autonomous trajectory for PCNL. The algorithm must consider multiple objectives to plan an ideal path and provide not one but multiple paths that are safe and effective in reaching the kidney stones. Providing multiple suitable optimized paths allowing a medical professional to make the decision for the final tradeoff of objectives which ensures the safety of the procedure and keeps a human in the loop. A frequently used and highly effective multi-objective algorithm, NSGA-II is well-known for its computational efficiency and performance. For example, in [95], NSGA-II is used to optimize the path planning task for an unspecified autonomous robot, where the paths are represented using splines to provide smooth trajectories.

In multi-objective optimizers, a high number of optimization variables can increase the computational time significantly. Therefore, to limit the dimension and the complexity of the problem, it is preferable if only a few coordinates are necessary to define the paths. For example, to maintain a smooth trajectory, a curve can be interpolated based on anchor points created by NSGA-II. Alternatively, B-splines can be used to represent these curves using only a few anchor points.

To determine a solution for the optimal tool trajectory, multiple factors need to be considered. Some of these factors may negatively impact one another requiring a trade-off between them. A single objective optimizer could be used here by implementing a weighted sum of objectives, however, these algorithms would only return

a single solution and when multiple actual cost functions are present it is more likely to become trapped in local minima, which do not allow for wider exploration. Furthermore, this removes a human from selecting the actual path that will be executed, if the resulting path is determined to not be suitable the algorithm may need to be run several times. Another consideration for weighted sums is that nuances regarding the individual performance of cost functions are lost which can negatively impact the resulting trajectory. A more suitable approach for problems with multiple objectives is a multi-objective optimizer. A multi-objective optimization algorithm finds solutions that satisfy each of the objectives; it returns a set of solutions that optimizes each objective to a varying degree rather than a single solution that it believes to be optimal. This has the added benefit of keeping a human in the loop of the procedure planning process when applied to PCNL. Since a medical expert would make the final selection for which path to choose based on their own expertise. This ensures the procedure's safety is maintained since a medical expert could determine that none of the proposed solutions are suitable.

Within multi-objective optimization, the solution set returned at the end should be the set of non-dominated solutions i.e., the Pareto front. A Pareto front is the subset of solutions that are not dominated by other solutions present. A solution is considered dominated if it is worse with respect to all objectives compared to another member. The set of solutions which are not dominated by any of the other population members are returned as the final results from the multi-objective optimizer. Multi-objective optimizers consider each objective separately rather than a weighted sum of the objectives. The task at hand is to find the Pareto optimal solutions, such that the solutions can be scrutinized and selected for execution. From the solution set returned by the optimizer, the user has the ability to determine an acceptable trade-off when selecting a solution.

Applying a multi-objective optimizer to path planning first requires a discretization of the problem in order for the optimizer to generate possible optimal paths. A B-spline representation of a trajectory requires a minimal number of control points, also known

as anchor points, to define the curve.

3.2 PCNL Trajectory B-spline Representation

B-splines are well-suited to obstacle avoidance path planning tasks as they are computationally efficient and able to provide smooth continuous trajectories from only a few anchor points. B-splines are a piece-wise polynomial function constructed from a set of basis functions given as

$$N_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} N_{i+1,j-1}(t) \tag{3.1}$$

where $j = 1, 2, \ldots, d$, and d is the degree of the B-spline being constructed. Basis functions construct the underlying curves that will be multiplied by the anchor points and added together to create the overall B-spline. For a first degree spline the basis functions are straight lines, when multiplied by the anchor points the resulting B-spline is a set of straight lines connecting each anchor point, while the basis functions for second degree splines are parabolas. Lower degree splines remain closer to the anchor points used to generate them but are less smooth than higher order splines which tend to follow their original anchor points less directly. Additionally, the number of anchors and knots used to generate various degree B-splines differs, this is discussed in more detail later.

The degree of the spline is calculated with

$$d = m - n - 1 \tag{3.2}$$

for a set of $n + 1$ control points and a knot vector $\mathbf{t} \in \mathbb{R}^{1 \times m+1}$. When $j = 0$ the basis function takes the form of

$$N_{i,0} = \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where t_i is the i^{th} entry in the knot vector which must be a non-decreasing series and each $t_i \in [0, 1]$. The final curve is then calculated as

$$C(t) = \sum_{i=1}^{n} \mathbf{P}_i N_{i,d}(t), \tag{3.4}$$

where $\mathbf{P}_i \in \mathbb{R}^D$ are the coordinates of control point i for the dimension D, and $N_{i,d}(t)$ is the corresponding basis function.

Splines are also well-suited to be used within an optimizer, as a relatively complex curve requires only a few control points $\mathbf{P}$ to define it, thus keeping the dimensionality of the problem low. A 3^{rd} degree B-spline while be used as it is still relatively smooth, and each of the anchor points are able to more directly control the final curve compared to a higher order spline. The objective of the optimizer is therefore to determine the coordinates of a limited number of control points such that the path created by them optimizes for a set of pre-defined cost functions.

3.2.1 Mutli-objective control point optimization cost functions

NSGA-II uses four objectives when planning the ideal path for PCNL:

1) Path Length is to be minimized. For a 2 dimensional scenario $C(t)$ contains the piecewise equations $x(t)$ and $y(t)$ which describe the x and y coordinates along the curve. Thus the path length can be defined as

$$f_1 = \int_a^b \sqrt{1 + \left(\frac{y'}{x'}\right)^2}\, dx \tag{3.5}$$

where $x' = \frac{dx}{dt}$, $y' = \frac{dy}{dt}$ and $a \in \mathbb{R}^D$ and $b \in \mathbb{R}^D$ are the start and end points of the path respectively and $C(t)$ is the equation of path created using B-splines.

2) Obstacle Distance describes the distance between the tool shaft and obstacles through the entire path trajectory. Maximizing the distance to an obstacle is important to avoid collisions and keep the surgical tool as far away as possible from sensitive

tissue during the procedure. To this end, the patient anatomy can be discretized as a set of voxels or pixels depending on the dimension of the problem. The centre of each voxel or pixel is stored in a matrix $T \in \mathbb{R}^{D\times\kappa}$ where κ is the total number of points used to describe the obstacle space. The distance to obstacles along the path must be maximized. First the path $C(t)$ is resampled such that there is a discrete set of evenly spaced points describing the path, this is denoted $C_R(j) \in \mathbb{R}^{D\times\ell}$ where ℓ is the total number of points used to describe the re-sampled path. The total distance to closest obstacles along the path can now be quantified as

$$f_2 = \sum_{j=1}^{\ell} \min ||C_R(j) - T||. \tag{3.6}$$

It may be beneficial to consider the distance of the entire tool to its surrounding obstacles, with the tool pose known for each point along $C_R(j)$, (3.6) must be calculated and summed for the entire tool shaft. Thus (3.6) becomes

$$f_2 = \sum_{z=0}^{L} \sum_{j=1}^{\ell} \min ||C_{RT}(j,i) - T||, \tag{3.7}$$

where L is the total length of the tool and $C_{RT}(j,z)$ is a point at a distance z along the tool shaft for the j^{th} point along the resampled path $C_R(j)$; and when $z = L$ $C_{RT}(j,L) = C_R(j)$.

3) Path Smoothness is to be minimized to ensure that the path is smooth and free from unnecessary jerk. The smoothness of the path further ensures patient safety by minimizing rapid changes in motion that could cause additional tissue damage. The smoothness is quantified by

$$f_3 = \int_a^b \frac{\left(Q(t)/dt\right)^2}{\sqrt{x'^2 + y'^2}} dt \tag{3.8}$$

where

$$Q(t) = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}}; \tag{3.9}$$

here $x'' = \frac{d^2x}{dt^2}$ and $y'' = \frac{d^2y}{dt^2}$ [96].

4) Tissue Compression Energy is a measure of how much energy is stored in the patient tissue due to its compression for the entire desired tool trajectory. During PCNL, it is assumed that the tissue compression occurs along the tool's entry point in the kidney only. The tissue compression energy for a single point along the trajectory is given as

$$U(d) = \frac{1}{2} K \int_{z_1}^{z_2} s(z)^2 \mathrm{d}z, \tag{3.10}$$

where z_1 and z_2 are the entry and exit depths along the tool axis respectively and

$$s(d,z) = \begin{cases} (z\cos\theta + x_t - x_{OP}(z)) & \text{if } z_1 \leq z \leq z_2 \\ 0 & \text{otherwise,} \end{cases} \tag{3.11}$$

where, as shown in Fig. 3.1(a), z is the distance along the tool axis, x_t is the horizontal coordinate of the tool base (where it is attached to the robot end-effector), θ is the angle of the tool, and $x_{OP}(z)$ is an equation describing the original entry path into the kidney.

The optimization variables are the coordinates for the anchor points, these anchor points are used in addition to the goal point and three constant points used to define a straight entry trajectory when creating the B-spline. Moving forward, the anchor points created by the optimization algorithm may be referred to as the internal anchors. If five internal anchors are used in a 2-dimensional scenario, there is a total of 10 optimization variables ($D\times$number of internal anchors).

A curve is generated for each of these population members and is then evaluated on the four cost functions described above. The evaluation of the cost functions requires that the tool pose is known. During autonomous PCNL kidney access, the tool is attached to the robot arm at its end-effector and the robot steers the tool from outside the tissue such that the tip follows the path to the end, that is, the location of the calculi. The inverse kinematics is presented in **Chapter 2**. This is simulated during NSGA-II to evaluate the cost functions for every iteration along the discretized path.

The results of the cost functions determine the performance of each individual and

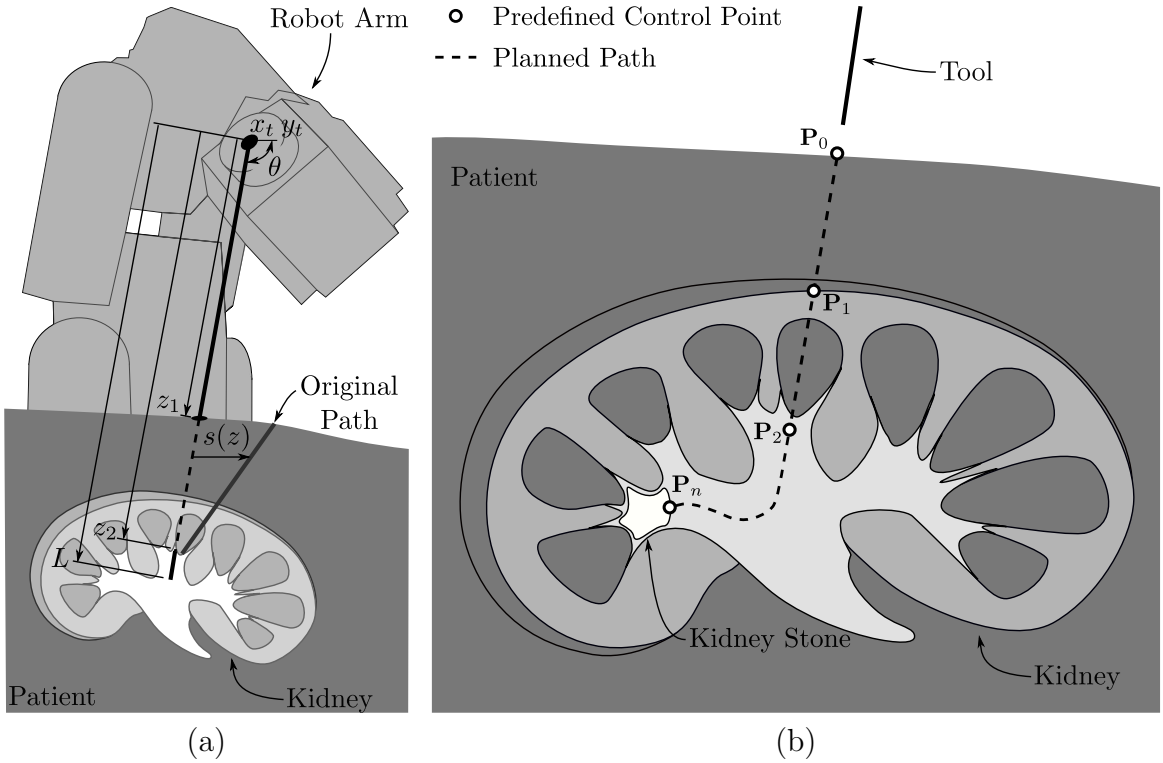


Figure 3.1: Automated robotic PCNL overview, (a) shows a robot arm holding a nephroscope and how the tool enters the kidney, (b) shows 4 predefined constant control points and a possible B-spline created between these points leading to the kidney stone.

affect the chance of a population member being used to generate the child population for the next iteration of the algorithm. The initial population is randomly generated, while each subsequent iteration of NSGA-II applies crossover and mutation schemes to the previous population to generate a new one. NSGA-II iterates until a stopping criterion is reached, in this case, a specified number of iterations. Once NSGA-II reaches the desired number of iterations, it returns the final population, the size of the final population is the same as that of the initial population or smaller, because the algorithm rejects solutions from the final set if they are not considered Pareto optimal.

Having defined all objective functions, let us shift the focus to the multi-objective optimizer.

3.2.2 Multi-objective optimizer outline

NSGA-II is a widely used and recognized multi-objective optimization algorithm. It operates on the principle of evolution, a population of potential solutions is evaluated using fitness functions and the highest-scoring members are more likely to produce the next child generation. The algorithm iterates until stopping criteria are met, typically by evaluating convergence to the Pareto front.

Population members are altered to explore the potential combinations of parameters. This is achieved using the fundamental transformations of a genetic algorithm: crossover and mutation. Crossover is performed first, where two parent population members ($\mathbf{n_{p1}}, \mathbf{n_{p2}}$) are used to create the two child members ($\mathbf{n_{c1}}, \mathbf{n_{c2}}$) using the following equations [97]:

$$\mathbf{n_{c1}} = 0.5((1 - B)\mathbf{n_{p1}} + (1 + B)\mathbf{n_{p2}}) \tag{3.12}$$

$$\mathbf{n_{c2}} = 0.5((1 + B)\mathbf{n_{p1}} + (1 - B)\mathbf{n_{p2}}) \tag{3.13}$$

where B is calculated using a random variable $0 \leq \mu_c \leq 1$ for each parameter and the user-defined crossover rate ζ_c as shown in [97–100], it is determined as follows:

$$B = \begin{cases} (2\mu_c)^{1/(\zeta_c+1)} & \text{if } \mu_c \leq 0.5 \\ (2(1 - \mu_c))^{1/(\zeta_c+1)} & \text{if } \mu_c > 0.5. \end{cases} \tag{3.14}$$

Polynomial mutation is then applied to the child population [98]. Mutation is achieved by using a random variable $0 \leq \mu_m \leq 1$ for each parameter and the user-defined rate ζ_m. If the random variable $\mu_m \leq 0.5$, the update variable is evaluated as [98],

$$\delta_l = (2\mu_m)^{(1/(\zeta_m+1))} - 1 \tag{3.15}$$

and the expression for the mutation becomes [98],

$$\mathbf{n_{c1}} = \mathbf{n_{c1}} + \delta_l(\mathbf{n_{c1}} - \mathbf{n}_{UB}) \tag{3.16}$$

$$\mathbf{n_{c2}} = \mathbf{n_{c2}} + \delta_l(\mathbf{n_{c2}} - \mathbf{n}_{UB}); \tag{3.17}$$

Alternatively, should $\mu_m > 0.5$ then the update variable becomes,

$$\delta_r = 1 - (2(1 - \mu_m))^{(1/(\zeta_m+1))} \tag{3.18}$$

and the children are mutated using,

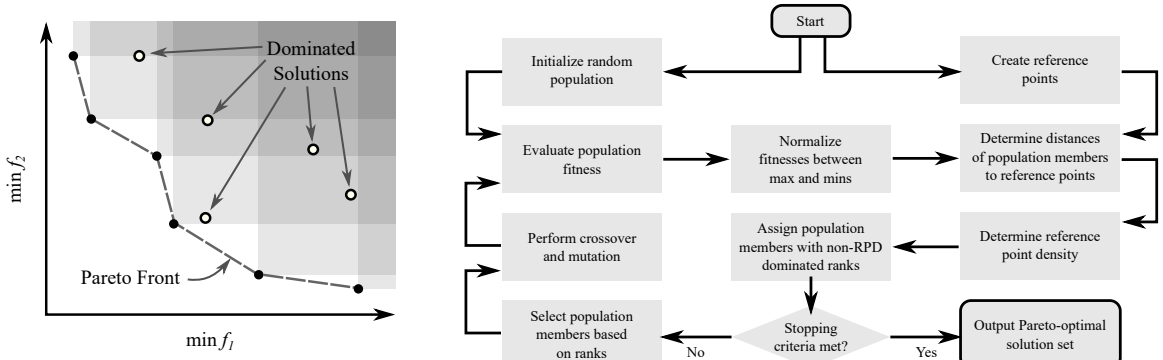
$$\mathbf{n_{c1}} = \mathbf{n_{c1}} + \delta_r(\mathbf{n}_{LB} - \mathbf{n_{c1}}) \tag{3.19}$$

$$\mathbf{n_{c2}} = \mathbf{n_{c2}} + \delta_r(\mathbf{n}_{LB} - \mathbf{n_{c2}}) \tag{3.20}$$

assuming that there are upper $\mathbf{n}_{UB}$ and lower $\mathbf{n}_{LB}$ bounds that a parameter may take.

Through an iteration of the optimizer, each population member has its fitness determined by evaluating the cost functions. The maximum and minimum fitness values for each cost function is used to normalize the fitness values of the population members. The normalization process remaps the maximum and minimum of a cost function to 1 and 0 respectively. Thus, the population member with the lowest value for the cost function being considered has a value of zero once it is normalized, while the population member with the highest cost function value now has a one in its place. All population members that do not have the highest or lowest cost function value have their fitness value linearly remapped between 0 and 1 based on the overall maximum and minimum values for the specified fitness function.

When a larger number of objectives are required, the process can converge on a local minimum, where the true Pareto front is not well represented. The authors in [101]



(a) Example of a Pareto front

(b) NSGA-II algorithm process

Figure 3.2: A non-dominated solution set when minimizing two cost functions f_1 and f_2 is shown by black dots in (a), dominated solutions are shown as white circles with black borders, and (b) depicts a flow-chart for the non-RPD-dominated NSGA-II algorithm

propose implementing reference point dominance (RPD). The algorithm creates evenly distributed points along the objective space hyperplane, applies additional penalties when determining the non-dominated solutions. Penalties are determined by evaluating the density of population members near the reference points and convergence to the ideal point. The RP-fronts are determined using the Euclidean distance

$$s_1(\mathbf{n}) = \frac{\|\tilde{f}(\mathbf{n})^T R_k\|}{\|R_k\|} \tag{3.21}$$

and the normal distance

$$s_2(\mathbf{n}) = \left\| \tilde{f}(\mathbf{n}) - s_1(\mathbf{n}) \left(\frac{R_k}{\|R_k\|} \right) \right\| \tag{3.22}$$

with $\tilde{f}(\mathbf{n})$ being the normalized objective function, and R_k is an M-dimensional direction vector [101], these can be evaluated to their scalar value for every member in the population.

Including the parameters described in (3.21) and (3.22) in the algorithm ensures good convergence and diversity of the potential solutions respectively. For further information on the RPD algorithm refer to [101].

Using these distances, the Pareto front can be determined. A visual example of a

Pareto front for the minimization of two cost functions is given in Fig. 3.2(a). The child population is constructed from Pareto front solutions. The process repeats until the stopping criteria of the optimizer is met, then the process is complete. The stopping criteria, at a minimum, consists of a maximum number of iterations but can also check the number of fitness function calls, change in the variables for convergence, or fitness level. The full sequence of the optimizer is shown in Fig. 3.2(b).

3.3 Simulation Results

The proposed path planning method is first evaluated through a set of simulations. The simulations are performed for a simplified 2D version of the kidney and its anatomy, based on the phantom kidney used for the physical experiments in **Chapter 2**, and in a later section, see Fig. 3.6. Additionally, three different goal points are considered during four scenarios, each goal is selected to demonstrate the algorithm's ability to explore the environment.

Scenario 1: evaluates the performance of the algorithm on **Goal 1** which is located on the bottom left of the simulated kidney anatomy. No obstacles are considered other than the kidney walls.

Scenario 2: is the same as scenario 1 with the exception that it considers **Goal 2** which is located on the bottom right.

Scenario 3: considers **Goal 3**, located on the top left.

Scenario 4: includes an additional obstacle to demonstrate the optimizer's ability to plan around obstacles, it considers **Goal 1**.

Fig. 3.4(b)-(e) depicts each of the goals and scenarios described above with the resulting paths from the optimization process. These simulations are performed on an *Intel i7* processor with 64 GB of RAM and *GeForce RTX 2080* GPU.

During PCNL, the trajectory into the kidney is a straight path from the incision to where it enters a calyx. To ensure this entry trajectory remains straight and that

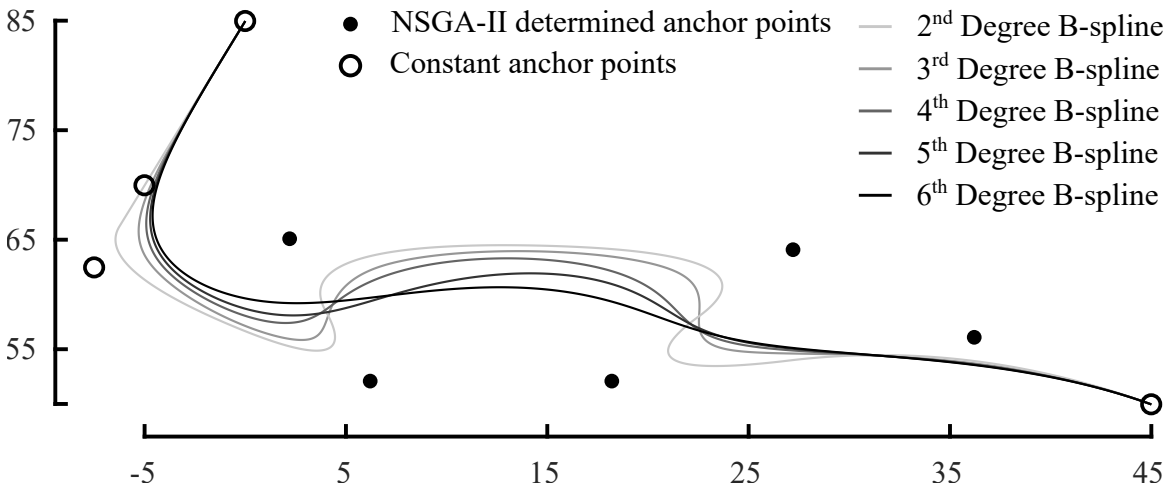


Figure 3.3: Example of a B-spline with a total of 9 anchor points, constant anchor points are shown with circles, the first 3 ensure a relatively straight entry trajectory, the last one represents the goal point. The location of the internal anchors are determined by the NSGA-II algorithm (black dots). The degree of splines are depicted for 2^{nd} to 6^{th} degree splines ranging in color from light grey to dark grey respectively.

the trajectory within the kidney is smooth and continuous along this first part of the trajectory, the first three control points for the B-spline are constant and determined in advance based on the desired entry trajectory. The final point for the uniform B-spline, or the goal point for the path, is also maintained as a constant for all of the B-splines. The NSGA-II algorithm controls the coordinates for five points between the three points defining a straight entry path and the final goal point see Fig. 3.3. The degree of a B-spline curve is determined by (3.2) which depends on the number of anchor points and the size of the knot vector. The degree of the B-spline can be increased through knot repetition at the beginning and end of the knot vector. For example: consider a 4^{th} degree ($d = 4$) B-spline curve is created through 9 anchor points ($n = 8$). Equation (3.2) is rearranged as $m = d + n + 1$, so the knot vector size can be determined as $m = 4 + 8 + 1 = 13$ and the knots at the beginning and end are repeated $d + 1$ times leaving 4 internal knots. It is important to consider the number of internal knots $\mathbf{t_{internal}} \in \mathbb{R}^{1\times q}$ necessary when using knot multiplicity $q = m - 2d - 1$ and $q \geq 0$. Thus for a 4^{th} degree B-spline curve, at least 5 control points are necessary ($q = 0$).

For the 2D scenario with 5 anchor points, there will be a total of 10 variables that

Table 3.1: NSGA-II algorithm parameters

Simulation parameters		Workspace [mm]	
Dimension	10	x range	y range
Iterations	500		
Number of Objectives	4	min -60	min 40
Population Size	60	max 60	max 80

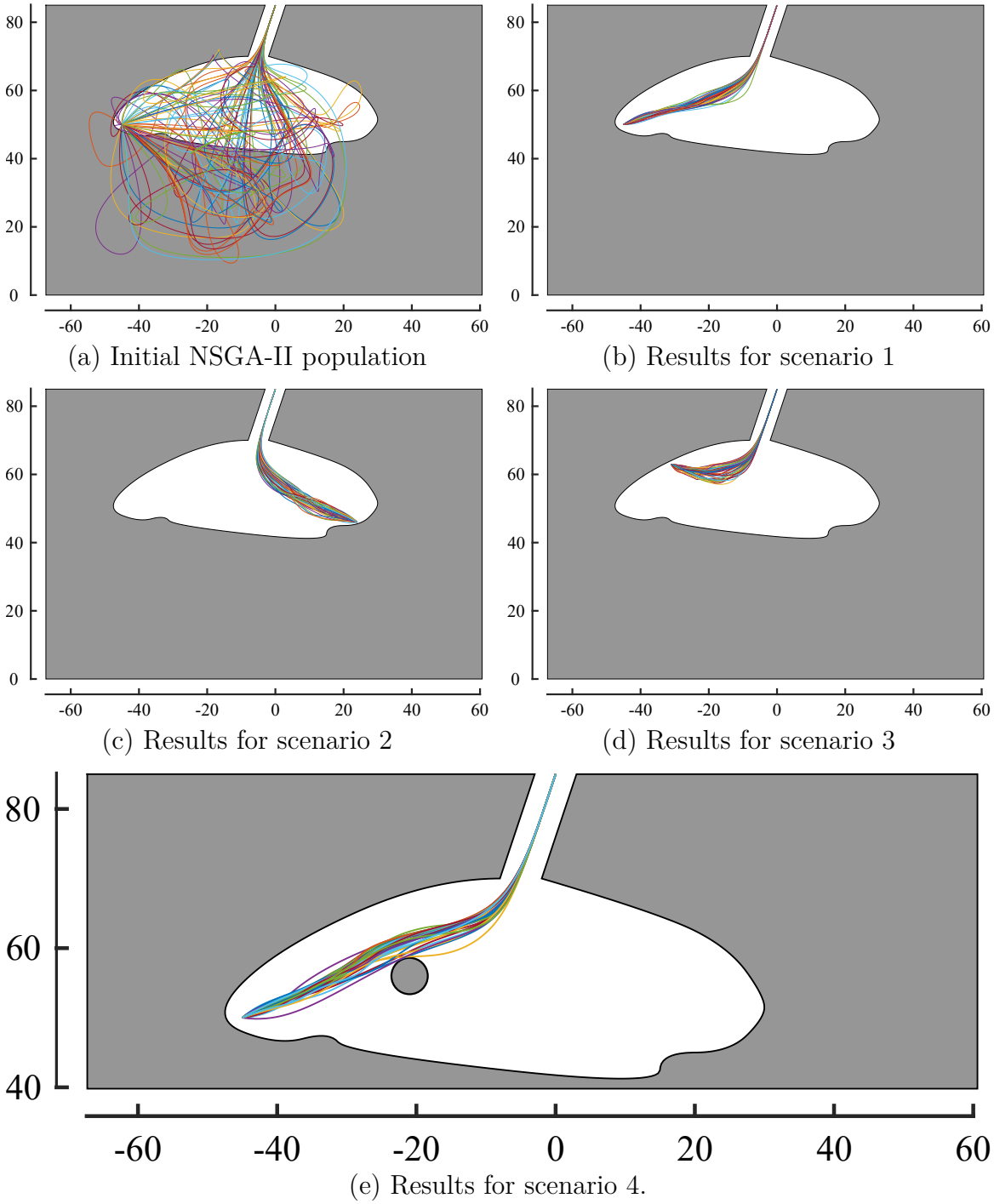
NSGA-II controls (the x and y coordinates for each point). The NSGA-II simulation parameters used during the optimization process, are found in Table 3.1.

The results from the NSGA-II optimization for each simulation scenario can be seen in Fig. 3.4. In subfigure (a) is an example of a random initial population. Subfigures (b) to (d) show the final Pareto optimal solutions for each goal point. As it can be seen, the Pareto optimal solutions are all smooth curves that are relatively short and do not collide with any obstacles. Subfigure (e) shows the Pareto optimal solution for scenario 4, i.e., goal 1 when an obstacle is present, demonstrating the algorithm's ability to create paths that still navigate around more complex surroundings with smooth trajectories.

From the Pareto optimal solution set, one member must be chosen to execute the task. This is a benefit of using a multi-objective optimizer that returns multiple solutions as it keeps a human in the loop of the planning procedure. Since all of the cost function values have been optimized to some degree, the generated solutions all represent acceptable trajectories. From these solutions, the operator can discard paths that are closer to obstacles than others, apply a maximum acceptable threshold to tissue compression, etc.

The cost function values are normalized relative to their respective minimums and maximums to compare them more easily. Furthermore, three out of the four cost functions must be minimized. For consistency, obstacle distance is inverted after normalization, thus in table 3.2, lower values of obstacle distance are more desirable.

Selecting the optimal path from the Pareto front is then achieved in two steps. First, a subset of the final Pareto optimal solutions is selected. This is shown in Table 3.2



(a) Initial NSGA-II population

(b) Results for scenario 1

(c) Results for scenario 2

(d) Results for scenario 3

(e) Results for scenario 4.

Figure 3.4: The results from NSGA-II Path planning, (a) shows an example of a randomly initialized population, (b)–(d) are the Pareto optimal solution sets returned for scenarios 1 through 3, (e) shows the Pareto optimal solutions for scenario 4 i.e. when an obstacle is present in the environment.

for scenario 3. The subset is selected by applying an upper threshold of 0.65 and 0.72 to the normalized tissue energy and obstacle distance cost functions, respectively. Following this, the cost function values for each member are compared and members whose cost function values are too similar to a previous member are discarded. Cost function similarity is determined by the root-mean-squared of the difference between two members' fitness values.

In the second step, a single solution is then selected from the subset by inspecting each individual solution path visually, and by analyzing their cost function values. Since all paths tend to be relatively short, the other three cost functions may be given more weight during the decision. Obstacle distance is given priority.

Alternatively, a weighted sum could be used at this junction to automatically select the final result. Obstacle distance would likely be given the highest weight and path length and path smoothness would likely be given lower weights since all of the final paths are similar in length and are shown to be very smooth. Although, using a weighted sum takes the human out of the planning process, and this can pose a significant risk for surgeries. Medical experts may be more familiar with what trade-offs would be acceptable and they may be difficult to describe simply as thresholds or weighted sums. There is also a risk that the multi-objective optimizer does not return any paths that are suitable, either due to the random nature of the algorithm, the particular scenario being considered, or the algorithm parameters being used. For these reasons it is integral to maintain a human in the loop of decision making, especially in the planning phases of a medical procedure.

The selected result for scenario 3 has been highlighted in light grey in Table 3.2 (member 32). This path is being selected since it has the second-lowest obstacle distance and shows good results in the other cost function values. Although member 32's highest value is the length of this path, this value is not the longest path generated across all Pareto solutions. This indicates a good trade-off between cost functions.

The resultant tool poses required to follow each of the optimal tooltip paths from each scenario are presented in Fig. 3.5. The tool poses determine the tissue compression

Table 3.2: Subset of the normalized cost function values for scenario 3

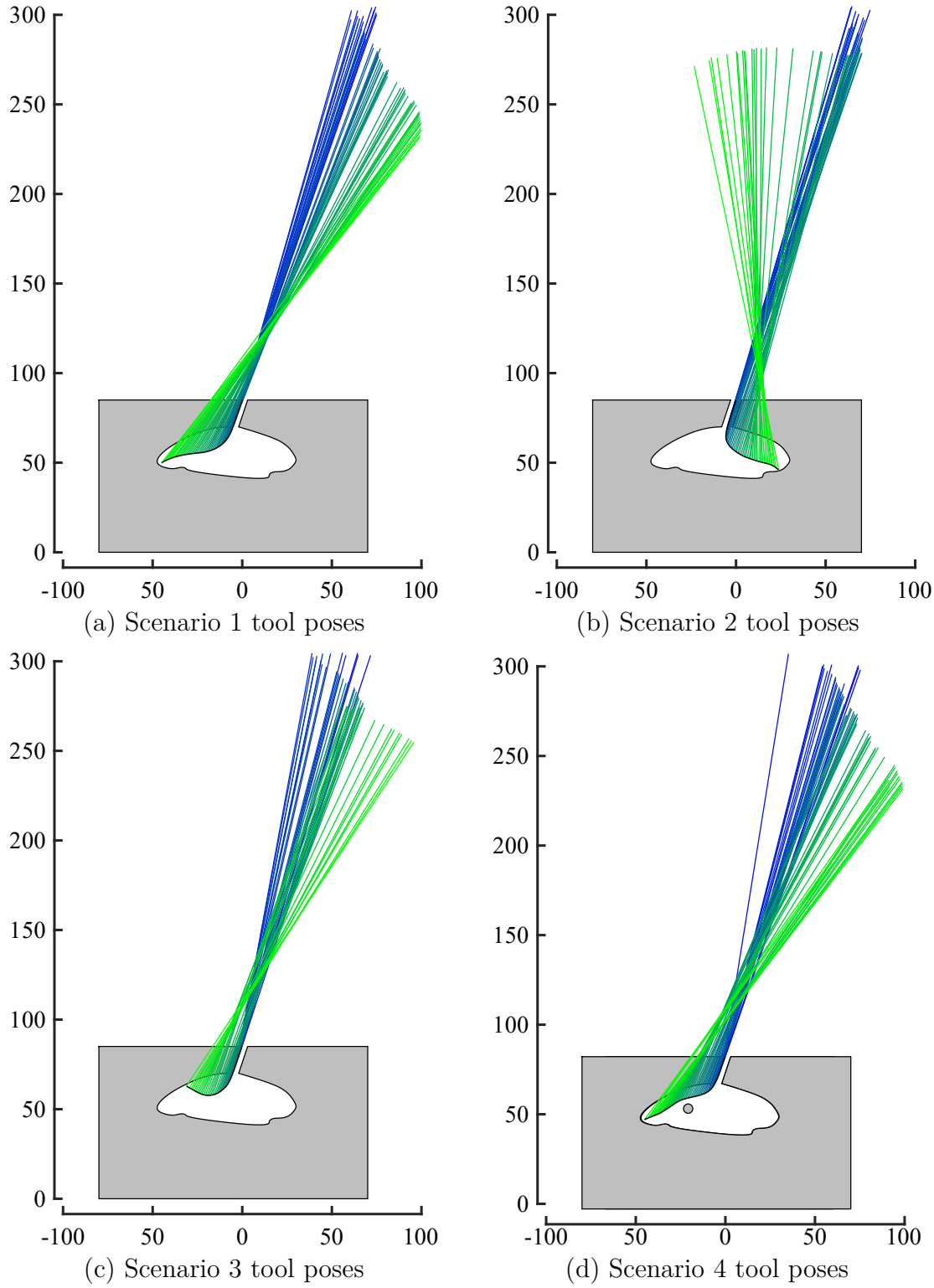
Member	Path length	Tissue energy	Path smoothness	Obstacle distance
4	0.2488	0.1471	3.130×10^{-10}	0.4201
7	0.6501	0.0394	9.563×10^{-9}	0.0441
8	0.3784	0.3950	1.987×10^{-12}	0.5035
9	0.5804	0.2587	4.951×10^{-12}	0.3939
11	0.1045	0.3088	1	0.3995
15	0.4690	0.2867	1.458×10^{-11}	0.1650
22	0.7817	0.4025	0	0.2561
30	1	0	5.681×10^{-12}	0
32	0.8116	0.3601	1.299×10^{-11}	0.0004
46	0.6750	0.4377	1.464×10^{-13}	0.6152

during the NSGA-II, they are also used to perform the physical experiments, described in the next section. From the simulation results, both the tooltip and the tool shaft do not intersect with the simulated obstacles. It is also worth noticing that even though the tool shaft is not constrained to the entry point in the tissue, the optimal path keeps the tool shaft close to its entry point to avoid tissue damage.

3.4 Experimental Evaluation

The path planning algorithm is also validated through physical trials. The paths selected during simulations are executed using the same 6-DOF (degree of freedom) robotic arm presented in **Chapter 2** (Meca 500, Mecademic, Montréal, Canada) in the same kidney phantom. The tool used in the physical experiments is a 218.7 mm long square brass rod with a width of 3.2 mm. An electromagnetic tool tracking system (Aurora, Northern Digital, Waterloo, Canada) records the tool's tip during the trials, see Fig. 3.6. The robot arm steers the tool base such that its tip follows the desired path. The robot's inverse kinematic is presented in detail in **Chapter 2**.

Each of the selected paths from the previous section is executed on this experimental setup to ensure that the paths are feasible. Each path is run three times. The recorded tooltip trajectory for each run is shown along with the reference trajectory in Fig. 3.7.



(a) Scenario 1 tool poses

(b) Scenario 2 tool poses

(c) Scenario 3 tool poses

(d) Scenario 4 tool poses

Figure 3.5: The results of the path planning algorithm, an ideal path is selected for each scenario and plotted in (a)–(d) above along with the tool poses generated during the path planning process in order to reduce tissue compression. The tool poses are shown in blue at their starts and become green as they approach the goal point.

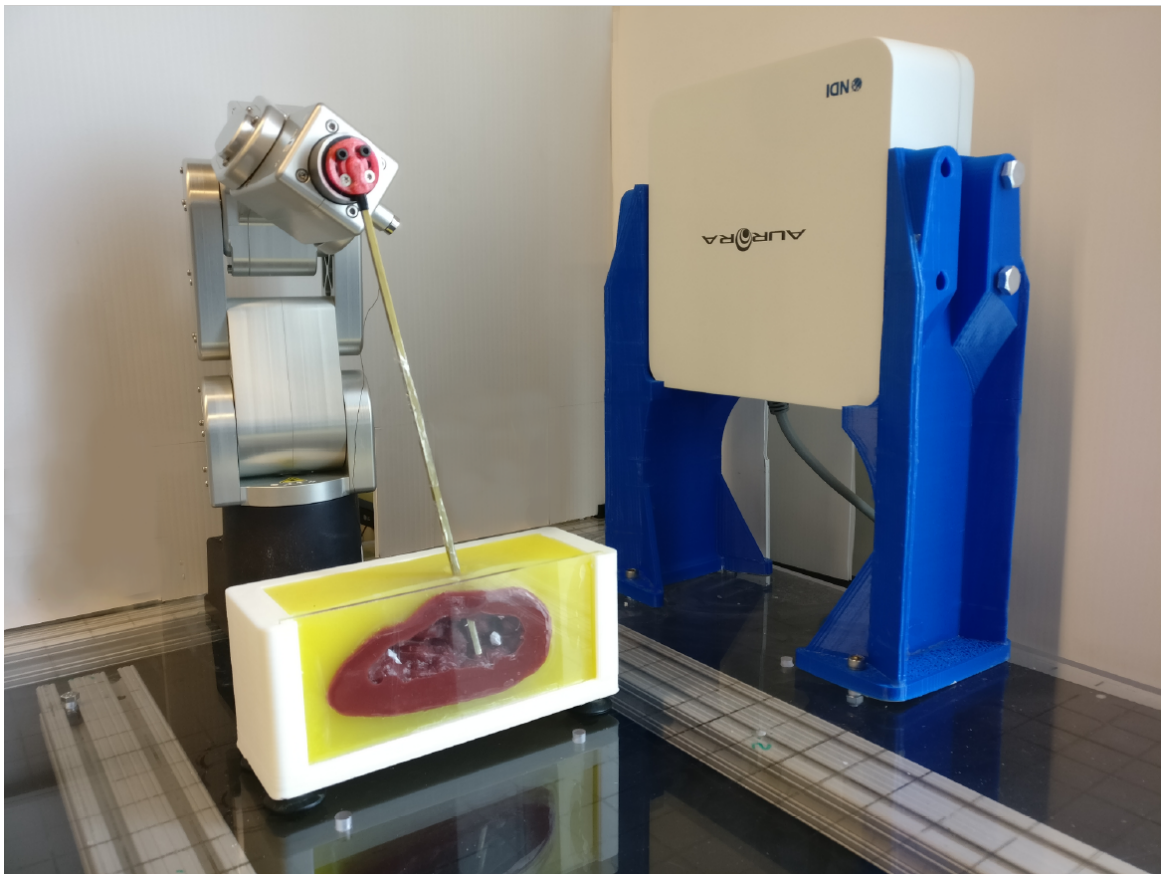


Figure 3.6: Experimental setup used to conduct the experiments on a phantom kidney model. A brass rod is used as the nephroscope. The 6-degree-of-freedom robot steers the tool base such that the tooltip follows the predefined trajectory. The tooltip location is recorded by the electromagnetic tracking system.

While the obstacle is shown in the figure, note that it is not present in the phantom kidney and is only considered during the planning of scenario 4.

All paths follow their reference trajectories, entering the kidney on a straight line, thanks to the first 3 anchor points, and then following a curved line around the obstacle towards the goal. Both the tooltip and the tool shaft successfully avoid the obstacle and the kidney walls and the tooltip reaches the desired goal. Lateral tool motion is also minimized by the algorithm to limit tissue compression. The increased tracking error as the tool approaches the goal is the outcome of the assumption that the tool is perfectly rigid, which is not the case for the brass tool used in the experiments. Additionally, some error may be caused by electromagnetic interference in the tracking process.

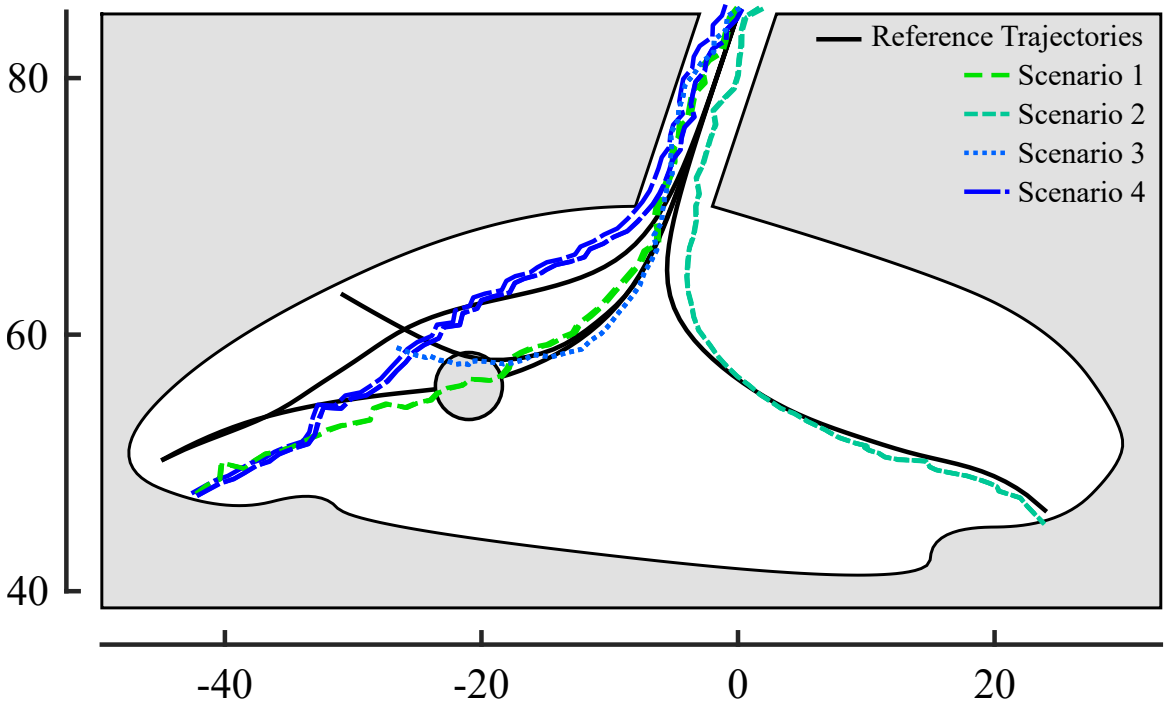


Figure 3.7: Each of the selected paths are plotted along with the tooltip tracking results from the physical experimental trials. Scenario 4 (the dark blue path) is the only trajectory that needs to manoeuvre around the obstacle.

3.5 Conclusion

This chapter proposes a path planning framework for PCNL using B-spline representation with the multi-objective optimizer NSGA-II. The resulting trajectories meet several requirements: being short and smooth while avoiding obstacles and high tissue compression. The path planning algorithm is validated on four scenarios, demonstrating its ability to plan an appropriate trajectory given different circumstances. A single trajectory is selected from the Pareto optimal solution set that is returned by the algorithm, this selection process keeps a human in the loop of planning the procedure to further ensure patient safety. Each of the selected trajectories is evaluated with three physical trials, where a robot arm controls the tool and advances it along the planned path towards the goal. The physical trials demonstrate the ability of the system to avoid obstacles while minimizing tissue damage at the entry point.

A well-planned trajectory is key to a successful and safe autonomous procedure.

Although, even the best-planned trajectories cannot ensure a safe procedure if the robotic agent is unable to accurately track the desired trajectory. For rigid tools, there is a straightforward solution. However, when considering much narrower or more flexible tools accurately determining the tool poses is significantly more challenging.

Chapter 4

Multi-objective Trajectory Tracking of a Flexible Tool during Robotic Percutaneous Nephrolithotomy

THE PREVIOUS chapter presented a multi-objective path planning algorithm for percutaneous nephrolithotomy (PCNL). Two assumptions were made in the algorithm, first, that the tooltip position is known; second, that the tool is rigid and straight. The above assumptions simplify the inverse kinematics of the robot significantly: there exists a straightforward relationship between the tool's base position and orientation and the tooltip position. However, it is desirable to use thinner, more flexible nephroscopes during the procedure to limit tissue damage and other possible complications. Meaning, for the desired tooltip position, the inverse kinematics becomes ill-posed, as there are several robot end-effector positions and orientations that bring the tooltip to the desired location while compensating for the tool bending.

Robotic assistance can be used to steer the tool while compensating for tool bending. In this scenario, the surgeons take on a more supervisory role in the operating room rather than a hands-on role. To achieve this level of autonomy, the robot must follow the planned path exactly, relying on accurately modelling the tool-tissue interaction. To reduce the potential for tissue trauma, using a flexible nephroscope can be beneficial. Flexible nephroscopes are recommended for some morbidly obese patients and during second-look procedures that follow PCNL [102]. They also offer additional manoeuvrability within the kidney providing better stone clearance with fewer incisions. While smaller nephroscopes are recommended for paediatric patients [6], the smaller diameter of the nephroscope means they are more prone to bending. However, using a flexible nephroscope poses additional challenges for a human to control in 3D space while receiving only 2D visual feedback from a fluoroscope.

If the robotic manipulator holds the base of the flexible tool as opposed to a rigid tool such as the one presented in the previous chapter, it creates kinematic redundancies. An appropriate solution will satisfy other qualitative and quantitative requirements such as reducing the potential for additional tissue trauma at the entry point, maintaining a smooth motion with minimal redundant movements, and ensuring the tool's deformation is reduced minimize the risk of snapping motions and improve tracking accuracy. When objectives like these are at odds, there often exists no global

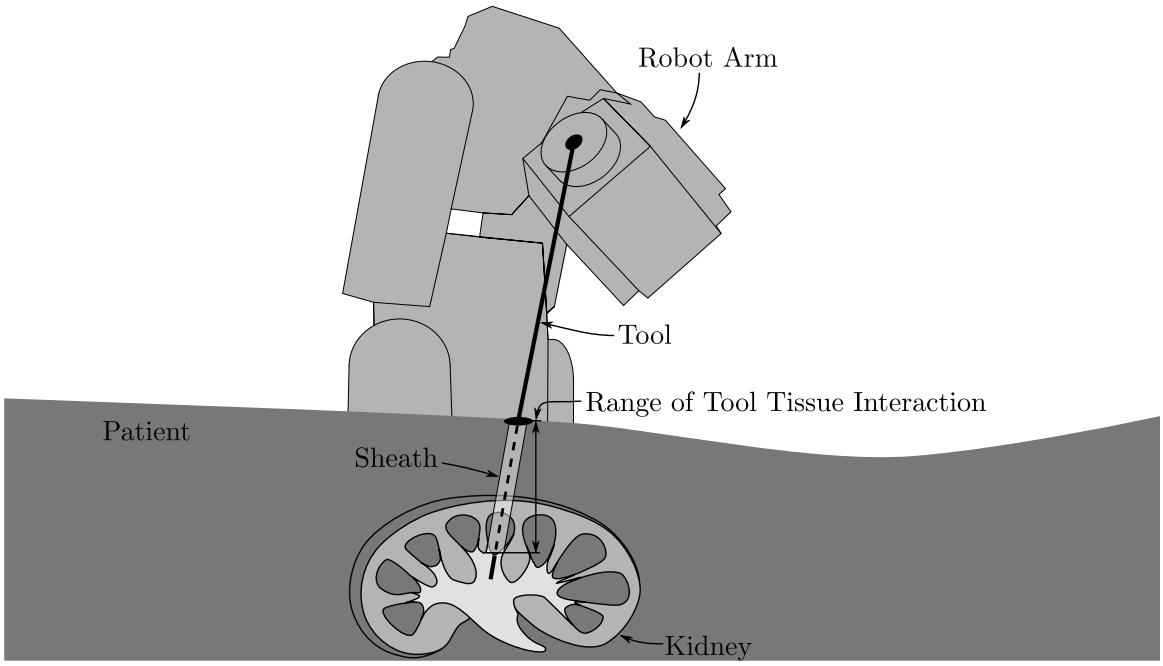


Figure 4.1: Overview of the proposed system, the flexible nephroscope is inserted into the kidney through the patient's back, the other end of the tool is attached to the end-effector of the robot arm which controls its position and orientation.

optimum between them and a compromise is necessary. This can be achieved by a multi-objective optimization algorithm, which creates multiple solutions for the tool position and orientation that are biased towards each objective to varying degrees.

This chapter proposes to use the NSGA-II algorithm presented in the previous chapter in combination with a flexible tool model to follow a predefined path during robot-aided PCNL. Given a desired tip trajectory, the multi-objective algorithm, described in Sec. 4.1, uses the forward solution to the tool bending model (Sec. 4.2) to calculate different positions and orientations of the end-effector that minimize four objective functions, namely (1) tissue displacement, (2) end-effector displacement, (3) variations in tool strain energy, and (4) tooltip placement error. The optimizer provides a variety of solutions to the steering problem, allowing the operator to choose from a set of results giving them the ability to determine the trade-off between objectives. This is different from **Chapter 2** as the multi-objective optimizer is used in place of an inverse solution to the tool-bending model.

The results of the trajectory tracking algorithm are validated through simulation and

experiments performed on ex-vivo porcine tissue, these are described in Sec. 4.4. The 25 physical experiments reported in Sec. 4.4.2 show an average trajectory tracking error of 2.03 mm during all trials. Finally, Sec. 4.5 discusses the general performance of the proposed framework.

4.1 Modelling Tool Bending during PCNL through Minimum Potential Energy

Accurately modelling the tool/tissue interaction has been the focus of extensive research in areas such as needle steering [103]. A nonholonomic model was used in [104–106] to reduce needle deflection during insertion. Similarly, in [107] the deflection of a needle is predicted primarily based on the characteristics of the tissue using the Rayleigh–Ritz method and a multilayer tissue model to estimate tool deflection. The authors in [108] use a combination of a biomechanics-based model, optimal rotation control, and online curvature estimation to steer a needle. Rayleigh-Ritz is a method of closely approximating the eigenvalues of a system using a direct numerical method. In the case of tool bending the eigenvalues are also known as modal coordinates and are needed to predict the bending of the tool.

It is clear that tool bending can have a negative impact on any surgical task, kidney access during PCNL is no exception. Accurately modelling the tool bending is integral to the safety and success of PCNL. To that end, the tool is modelled as a cantilever beam, similar to the model used in [109], with one end clamped to the robot end-effector and the other end free. The following formulation demonstrates this method in 2D, which will be expanded to 3D in future work. The formulation considers three assumptions:

- **Assumption 1,** the system is quasi-static model therefore mass and friction forces are neglected;
- **Assumption 2,** axial deformation along the tool axis is neglected;

- **Assumption 3,** tissue does not deform axially.

These assumptions hold true for the portions of the PCNL procedure considered here, additionally the tool does not undergo an axial load during standard PCNL. Since velocity and acceleration are both kept relatively low they have minimal impact on the tool model, thus, allowing a quasi static model to be employed.

The bending of the tool is given as the weighted sum of n vibration modes. This modal model determines the tool bending and tissue compression simultaneously through minimum potential energy without having to explicitly model the distributed load. The vibration modes are the first n mode shapes of free vibration for a cantilever beam with one fixed and one free end. The boundary conditions of the beam must be satisfied by trial beam deflection functions, i.e., eigenfunctions that represent the first n vibration modes of the beam [109, 110]. Additionally, the eigenfunctions must be differentiable up to the highest order of the partial differential equations of the beam; thus ensuring a numerical solution. The beam deflection $\nu(d, z)$ at a given insertion depth d and shaft location z is given by [103, 109]:

$$\nu\left(d, z\right) = \sum_{i=1}^{n} q_i(z) g_i(d), \tag{4.1}$$

where $q_i(z)$ is the displacement induced by the i^{th} vibration mode and $g_i(d)$ is the weight d, i.e., the modal coordinates. If L is the length of the beam, the eigenfunctions $q_i(z)$ can be normalized such that the maximum deflection is equal to unity and is given by [111]:

$$q_i(z) = \frac{1}{k_i}\left[\sin\left(\beta_i \frac{z}{L}\right) - \sinh\left(\beta_i \frac{z}{L}\right) - \gamma_i\left(\cos\left(\beta_i \frac{z}{L}\right) - \cosh\left(\beta_i \frac{z}{L}\right)\right)\right], \tag{4.2}$$

with the constant γ_i defined as:

$$\gamma_i = \frac{\sin\beta_i + \sinh\beta_i}{\cos\beta_i + \cosh\beta_i}, \tag{4.3}$$

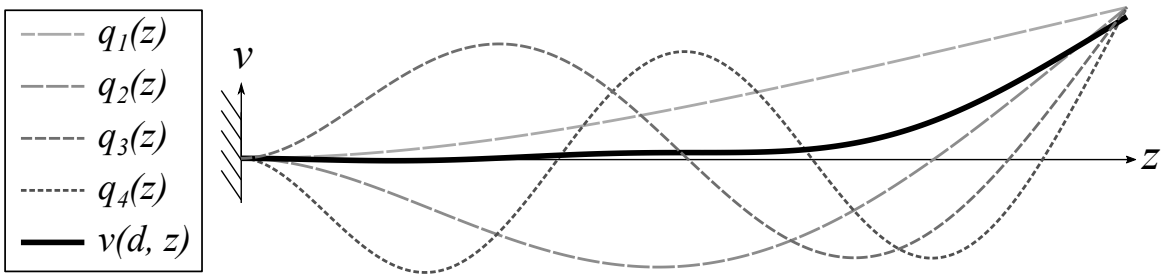


Figure 4.2: Tool bending vibration modes. The horizontal axis is the tool's axis, the dashed gray lines are the first four vibration modes of the beam, and the solid black line is an example of the weighted sum of vibration modes.

and the constant k_i computed by:

$$k_i = \sin\beta_i - \sinh\beta_i - \gamma_i\left(\cos\beta_i - \cosh\beta_i\right). \tag{4.4}$$

The constants β_i are given in [111] for a cantilever beam as $\beta_1 = 1.875$, $\beta_2 = 4.694$, $\beta_3 = 7.855$, $\beta_4 = 10.996$ and $\beta_i \simeq \pi\left(i - 0.5\right) \forall\ i > 4$, thus allowing (4.3) and (4.4) to be evaluated numerically. The summation in (4.1) is shown in Fig. 4.2, where the dashed and dotted lines represent the first 3 vibration modes of the beam and the solid black line is a weighted sum of the deflection functions. The modal coordinates are unknown and must be determined at each tool pose. The approximate beam deflection is now fully described by the above equations with the exception of the weights $g_i(d)$, which will be determined in the following section.

4.1.1 System Energy Equilibrium

During PCNL, a rigid sheath is placed in an incision made in the patient's back, the nephroscope is then inserted through the sheath and into the kidney. The nephroscope is constrained by the sheath, however, it is considered free within the kidney, see Fig. 4.1. When the flexible tool is inserted through the tissue and is given a linear or angular displacement from the original entry path, the tissue compresses due to the force/torque from the tool, which in turn causes it to bend as well. This interaction is shown in Fig. 4.3.

The modal coordinates in (4.1) can be determined by minimizing the energy in the

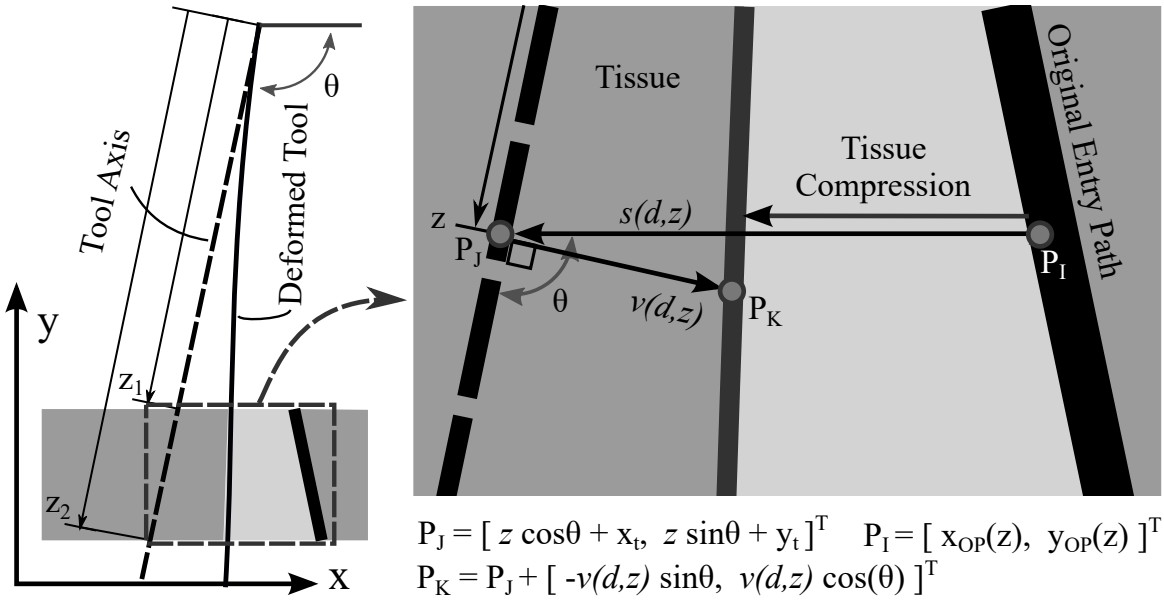


Figure 4.3: The displacement $s(d, z)$ between the tool axis and the original entry path is shown on the right. The angle θ describes the angular offset of the tool relative to the horizontal. A depth z along the tool axis gives P_I as the corresponding point along the original entry path, P_J as the corresponding location of the tool axis, and P_K as the location of the deformed tool, all three points are given relative to the base reference frame. The tool axis coincides with the undeformed tool, z_1 and z_2 are the z displacements along the tool axis for the entry and exit locations respectively.

system [109]. This takes into account the work done by an input force or torque, which causes the tool to bend, hence the tool/tissue interaction is a coupled effect and they must be solved concurrently. Equilibrium is attained by adding the energy stored in the tool due to it bending to the energy or work created by the tissue compression. To begin, the total energy within the tool/tissue model is:

$$\Pi\left(d\right) = U_d\left(d\right) + U_t\left(d\right) \tag{4.5}$$

where $U_t(d)$ is energy stored in the tissue due to compression and $U_d(d)$ is the energy stored within the tool due to bending, which can be described by:

$$U_d\left(d\right) = \frac{1}{2}\int_0^L EI(z)\left(\frac{\partial^2 \nu(d,z)}{\partial z^2}\right)^2 \mathrm{d}z, \tag{4.6}$$

where E is the tool's Young's modulus of elasticity, I is the second moment of inertia of the beam about its longitudinal axis, and $\frac{\partial^2}{\partial z^2}$ denotes the second derivative with

respect to z.

Accounting for the rigid sheath in this formulation requires changing the flexural rigidity of the tool, hereafter denoted by $EI(z)$, so the tool does not bend within the sheath. These values are changed when the depth z is between the start z_1 and end z_2 of the sheath. Thus, the flexural rigidity is:

$$EI(z) = \begin{cases} E_s I_s & \text{if } z_1 \ \leq \ z \ \leq \ z_2 \\ E_{tool} I_{tool} & \text{otherwise,} \end{cases} \tag{4.7}$$

where I_s and E_s as well as I_{tool} and E_{tool} are determined based on the geometry and characteristics of the sheath and tool, respectively with $I_s \approx I_{tool}$ and $E_s >> E_{tool}$.

The potential energy due to tissue compression $U_t(d)$ is slightly more complex. Tissue displacement is given relative to a base frame while the tool has a separate coordinate frame. The two frames need to be related to properly describe the tissue compression.

4.1.2 Tissue Potential Energy Calculation

The tool thus far has been described relative to its own reference frame, i.e., the robot's end-effector. The tissue location, on the other hand, is described relative to a base reference frame, i.e., the robot's base frame. Any coordinates or other variables will be denoted with the superscript $()^0$ or $()^t$ if given relative to the base or tool frame, respectively. Fig. 4.4 demonstrates the conversion of reference frames for a point P_1^t. Since the position and orientation of the tool need to be found relative to the base frame, it is most efficient to describe all points relative to the base frame.

Consider a point P_1^t given relative to the tool reference frame as $P_1^t = \begin{bmatrix} z_1^t & v_1^t \end{bmatrix}^T$, where $()^T$ denotes the transpose, z_1^t is a displacement along the tool's axis, and v_1^t refers to a displacement perpendicular to the tool's axis as in Fig. 4.4. In order to translate P_1^t to the base frame, the position $[x_t^0 \ y_t^0]^T$ and orientation θ_1 of the end-effector need to be known, along with θ_2 the mounting angle of the tool to the end-effector (see Fig. 4.4). If $\theta = \theta_1 + \theta_2$, then the coordinates of P_1^0 relative to the robot's base reference

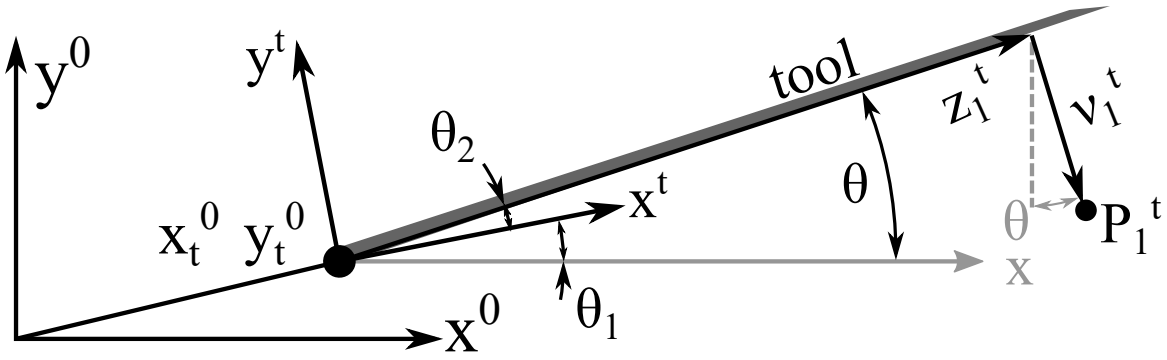


Figure 4.4: Translation of a point from tool frame to the base frame. θ_2 is the mounting angle between the end-effector and the tool, θ_1 is the orientation of the end-effector, z_1^t is the displacement of a point along the tool axis, ν_1^t is the displacement of a point perpendicular to the tool i.e., the deflection described in (4.1), and x_t^0, y_t^0 are the coordinates of the tool frame.

frame as:

$$P_1^0 = \begin{bmatrix} x_1^0 \\ y_1^0 \end{bmatrix} = \begin{bmatrix} x_t^0 + z_1^t \cos\theta + v_1^t \sin\theta \\ y_t^0 + z_1^t \sin\theta - v_1^t \cos\theta \end{bmatrix} \tag{4.8}$$

The tissue displacement for a point along the tool $z_1 \leq z \leq z_2$ is calculated as $s(d,z) - \sin\theta\nu(d,z)$, where $\nu(d,z)$ is the tool deflection and $s(d,z)$ is the distance between the original entry path and the tool axis, i.e., the tissue displacement from the original entry path to the tool axis is

$$s(d,z) = \begin{cases} (z\cos\theta + x_t - x_{OP}(z)) & \text{if } z_1 \leq z \leq z_2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

in which $x_{OP}(z)$ represents the x coordinate of the entry path at the specified point z along the tool shaft. See Fig. 4.3.

Finally, the potential energy stored in the tissue is:

$$U_t(d) = \frac{1}{2} K \int_{z_1}^{z_2} \left(s(d,z) - \sin\theta\nu(d,z)\right)^2 \mathrm{d}z, \tag{4.10}$$

where K is the stiffness of the tissue, and θ is the angular offset of the tool relative to the horizontal axis of the base frame as shown in Fig. 4.3 and Fig. 4.4. This formulation assumes that the tissue is homogeneous, heterogeneous tissue formulations

can be explored further in future work.

4.1.3 Minimizing Potential Energy through Modal Coordinates

The modal coordinates in (4.1) can now be determined through the Rayleigh-Ritz method. This method is a way of approximating the eigenvalues of a system directly, in this case, it is applied to determine the modal coordinates. $g_i(d)$ must ensure that $\delta\Pi_i(d) = 0$, where the infinitesimal difference is given by δ. Therefore, in order to minimize the energy within the system with unknown $g_i(d)$, the variation of $\Pi(d)$ with respect to $g_i(d)$ is set to zero by:

$$\delta\Pi_i(d) = \frac{\partial}{\partial g_i(d)}\left(U_d(d) + U_t(d)\right) \;=\; 0. \tag{4.11}$$

The potential energy from (4.6) can be inserted into the previous equation to give:

$$\begin{aligned}\delta\Pi_i(d) = \frac{\partial}{2\;\partial g_i(d)}\Bigg(&\int_0^L EI(z)\left[\frac{\partial^2}{\partial z^2}\sum_{i=1}^{n} q_i(z)g_i(d)\right]^2 \mathrm{d}z \\ &+K\int_{z_1}^{z_2}\left[s(d,z) - \sin\theta\sum_{i=1}^{n} q_i(z)g_i(d)\right]^2 \mathrm{d}z\Bigg) = 0.\end{aligned} \tag{4.12}$$

The partial derivative can be taken with respect to $g_j(d)$, allowing the above to be simplified to:

$$\begin{aligned}\int_0^L EI(z)\left(\sum_{i=1}^{n}\ddot{q}_i(z)g_i(d)\right)\ddot{q}_j\,\mathrm{d}z \;-\; K\sin\theta\int_{z_1}^{z_2} s(d,z)q_j(z)\mathrm{d}z \\ +\; K\sin^2\theta\int_{z_1}^{z_2}\left(\sum_{i=1}^{n} q_i(z)g_i(d)\right)q_j(z)\mathrm{d}z = 0,\end{aligned} \tag{4.13}$$

in which ($\ddot{}$) denotes the second derivative with respect to z. The first term can now

be expanded and written as:

$$g_1(d)\int_0^L EI(z)\ddot{q}_j(z)\ddot{q}_1(z)\mathrm{d}z + g_2(d)\int_0^L EI(z)\ddot{q}_j(z)\ddot{q}_2(z)\mathrm{d}z \\ \cdots + g_n(d)\int_0^L EI(z)\ddot{q}_j(z)\ddot{q}_n(z)\mathrm{d}z. \tag{4.14}$$

For simplicity, three new variables are created, i.e.:

$$\phi_{ij} = \int_0^L EI(z)\ddot{q}_i(z)\ddot{q}_j(z)\mathrm{d}z, \quad \omega_{ij} = \int_{z_1}^{z_2} q_i(z)q_j(z)\mathrm{d}z, \\ \text{and } \psi_i = \int_{z_1}^{z_2} s(d,z)q_i(z)\mathrm{d}z. \tag{4.15}$$

The above terms can be written in matrix form as a function of $g_i(d)$ since all terms other than the weights can be numerically evaluated. When taking the partial derivative in (4.11), n equations with n unknown variables are created, that is the weights $g_i(d)$. Thus, if the matrix M is full rank there exists a non-trivial solution for $g_i(d)$. The matrix allows the system to be rewritten as:

$$M\left[g_1(d),\, g_2(d), \cdots, g_n(d)\right]^T = K\sin\theta\left[\psi_1,\, \psi_2, \cdots, \psi_i\right]^T.$$

If $\mu_{ij} = \phi_{ij} + K\sin^2\theta\omega_{ij}$,

$$M = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n1} & \mu_{n2} & \cdots & \mu_{nn} \end{bmatrix}, \tag{4.16}$$

which gives the solution to the modal coordinates $g_i(d)$ through:

$$\begin{bmatrix} g_1(d) \\ g_2(d) \\ \vdots \\ g_n(d) \end{bmatrix} = M^{-1}\left(K\sin\theta\begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_i \end{bmatrix}\right), \tag{4.17}$$

where M^{-1} denotes the inverse of matrix M.

Now that the deflected tool shape can be found, the position and orientation of the end-effector need to be determined to reach a desired tooltip position. This is an ill-posed problem making the analytical solution nonviable. To find a set of viable solutions, a multi-objective optimization method is implemented to take into account various objective functions.

4.2 Multi-Objective Trajectory Tracking

To determine a solution for the position and orientation of the end-effector, multiple factors need to be considered. Some of these factors may negatively impact one another requiring a trade-off between them. For these types of problems, multi-objective optimization is utilized. The algorithm finds solutions that satisfy each of the objectives; it returns a set of solutions that optimizes each objective to a varying degree - in multi-objective optimization, these are the non-dominated solutions i.e., the Pareto front. From the solution set returned by the optimizer, the user has the ability to determine an acceptable trade-off when selecting a solution. Multi-objective optimizers consider each objective separately rather than a weighted sum of the objectives. The task at hand is to find the Pareto optimal solutions, such that the solutions can be scrutinized and selected for execution.

4.2.1 Objective Functions during Trajectory Tracking

A reference trajectory is discretized as a collection of n points, equally spaced along its length. Let index $1 \leq i \leq n$ denote the i^{th} point along the trajectory. Four cost functions are now defined as:

Minimize tooltip error: Minimize the error of tooltip position and goal position along the trajectory. Consider the reference position to be $P_G^0(i)$ and the estimated tooltip position to be $P_E^0(i)$. The Euclidean distance between them is used to deter-

mine the fitness as

$$f_1 = \sqrt{\sum \left[P_G^0(i) - P_E^0(i)\right]^2}. \tag{4.18}$$

Minimize robot end-effector displacement. This will reduce the motion of the robot end-effector and as a result the motion of the tool, thereby ensuring that the algorithm has a preference for poses that are close to the previous pose of the robot. The robot's end-effector position at the instant i can be described with $P_t^0(i)$, and the difference between its current position and the previous position is given as:

$$f_2 = \sqrt{\sum \left(P_t^0(i) - P_t^0(i-1)\right)^2}, \tag{4.19}$$

which should minimize the overall motion of the end-effector with the desired goal of minimizing erratic or unnecessary motion of the tool when following the desired path.

Minimize changes in the tool strain energy. While the tool is meant to bend slightly, the purpose of this cost function is to avoid sudden changes in the tool deflection to maintain stability and avoid snapping motions. Let the third cost function be the difference in tool potential energy from its previous position to its current position

$$f_3 = U_d(d_i) - U_d(d_{i-1}) \tag{4.20}$$

where the tool strain energy $U_d(d)$ is defined in (4.6).

Minimize energy stored in the tissue. This cost function minimizes tissue displacement along the entry point in the tissue, i.e., by minimizing the energy stored in the tissue due to compression as defined in (4.10). Thus:

$$f_4 = U_t(d_i). \tag{4.21}$$

This cost function can reduce potential tissue trauma by limiting the extent to which it undergoes deformations.

The NSGA-II from **Chapter 3** can now be used to return the optimal end-effector

Table 4.1: NSGA II trajectory tracking variable bounds

	Unit	Minimum	Maximum
x_t^0	mm	-100	100
y_t^0	mm	100	350
θ	deg	-150	-30

position and orientation for the robot that minimizes the cost functions described above. Each point along the trajectory is optimized separately, meaning a separate population is generated for each point along the trajectory. The tool pose and orientation are determined for the first point along the trajectory with the optimizer, then the process is repeated for each subsequent point until the end of the path is reached.

To determine the necessary tool position and orientation the following parameters are used in NSGA-II. The initial population is comprised of 100 members, each member contains three variables: the position coordinates of the end-effector x_t^0, y_t^0, and θ as described in the previous section and shown in Fig. 4.4. The parameters are initialized randomly between boundary values provided in Table 4.1.

Each of the cost functions is evaluated for each individual of the population. The fitness of each individual influences the subsequent population generated. To identify the impact of each cost function and its necessity they are evaluated through a set of simulations.

4.3 Simulation Results

To validate the proposed framework, several simulations are performed to justify the necessity of each cost function described in Sec 4.2. Each simulation is performed once on a target trajectory defined as a straight line inside the tissue with a length of 40 mm. The following tool parameters are used: $L = 231.8$ mm, $E_{tool} = 3.2$ GPa, and $I_{tool} = 25.2549$ mm^4. The tissue stiffness is $K = 27.5$ kPa, which is the same stiffness used for the physical experiments (the procedure for determining this value is given in the following section). Four different scenarios are considered:

Table 4.2: Mean and standard deviation for each cost function for simulations 1 through 4. Light grey indicates cost functions being used in the optimizer.

Case	f_1 $[\mu m]$	$f_2[mm]$	f_3 $[mJ]$	f_4 $[mJ]$
1	2.92 ± 1.64	14.0 ± 21.4	55.2 ± 115	1230 ± 2860
2	8.50 ± 4.24	4.34 ± 10.7	0.48 ± 0.33	9.27 ± 6.01
3	36.9 ± 47.2	4.76 ± 10.5	2.29 ± 5.37	41.2 ± 94.4
4	30.7 ± 57.1	3.88 ± 8.81	1.01 ± 4.77	17.4 ± 78.1

Simulation 1: The only cost function considered in the optimizer is the tooltip error f_1 defined in (4.18). This cost function is necessary to have the tool follow a desired path.

Simulation 2: The only cost functions considered are the tooltip error f_1 defined in (4.18) and the tissue compression f_4 defined (4.21). The effects of adding the tissue compression should be evident when compared to the previous trial.

Simulation 3: Here three cost functions are considered, i.e., the tooltip error f_1 (4.18), the tissue compression f_4 (4.21), and the tool strain energy f_3 (4.20).

Simulation 4: Here all four cost functions f_1, f_2, f_3, and f_4 are implemented in the optimizer concurrently.

Simulations are performed using an *Intel i7* processor, 64 GB of RAM and *GeForce RTX 2080* GPU. The results in Fig. 4.5 depicts the deformed tool shape for each pose along the executed trajectory. The figure portrays the effect of each cost function on the tool pose. The tool poses from simulation 1 are erratic and present alternating motion between poses. Simulations 2 and 3 have results similar to each other, however, their tool poses are significantly less erratic compared to simulation 1 since f_4 constrains the original entry path. Finally, the inclusion of f_2 in simulation 4 results in the least erratic tool base position, with tool poses creating a smooth tool motion. The mean and standard deviation of the cost functions for the selected results of the 4 simulation trials are shown in Table 4.2. The tool poses are selected for each point along the path from the final Pareto optimal sets returned by the optimizer, the population member with the lowest tooltip error is selected as the best solution.

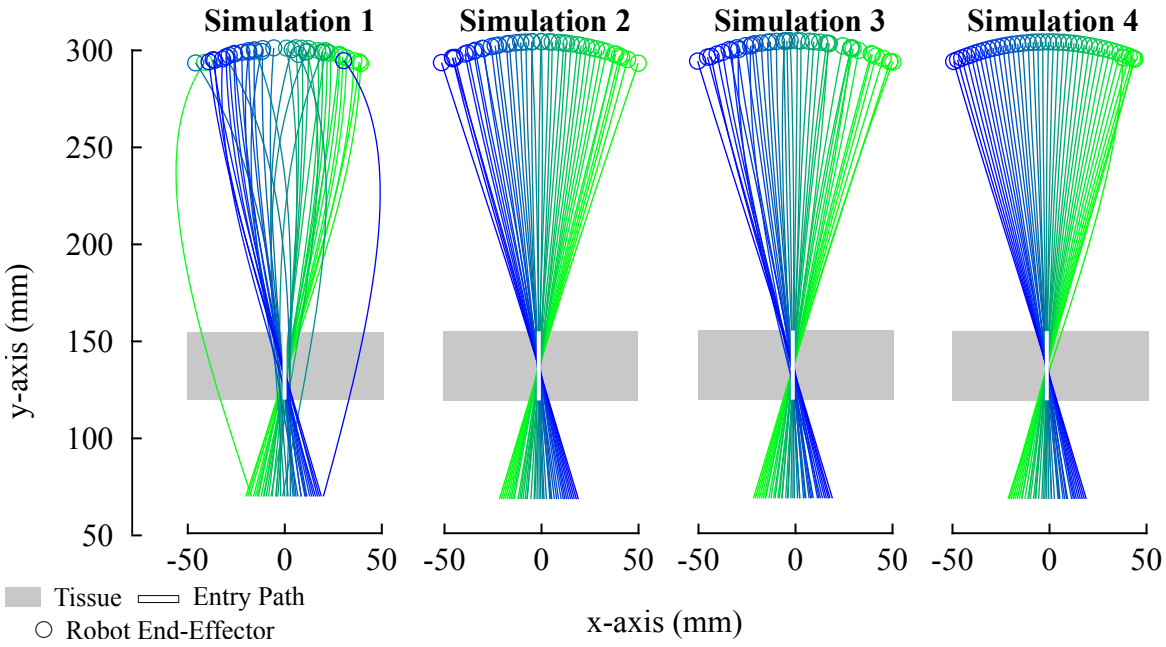


Figure 4.5: Simulation results demonstrating the effect of each cost function. The grey rectangles represent the tissue and the solid white line represents the original path through the tissue. Each tool pose along the trajectory is shown, green represents the first pose and each pose becomes more blue towards the end. Circles represent the position of the robot end-effector which generally moves from right to left (green to blue respectively) as the trajectory moves in the opposite direction.

The results show that there is a large difference between simulations 1 and 2 in tissue compression. This indicates that the cost function f_4 ensures that the tool passes more directly through the original path through the tissue, thereby reducing the potential for tissue damage. When considering simulation 3, Table 4.2 shows a slight increase in strain energy, i.e., f_3, for simulation 3 when compared to 2, however, it is still an improvement on simulation 1. Finally, simulation 4 includes f_2 which aims to minimize the movement of the tool base. This cost function appears to be effective in reducing the overall motion of the end-effector; thereby reducing unnecessary jerk or other undesirable motions.

4.4 Experimental Results

The experimental setup used to validate the proposed method again uses the 6-DOF (degree of freedom) robotic arm (Meca 500, Mecademic, Montréal, Canada), and an

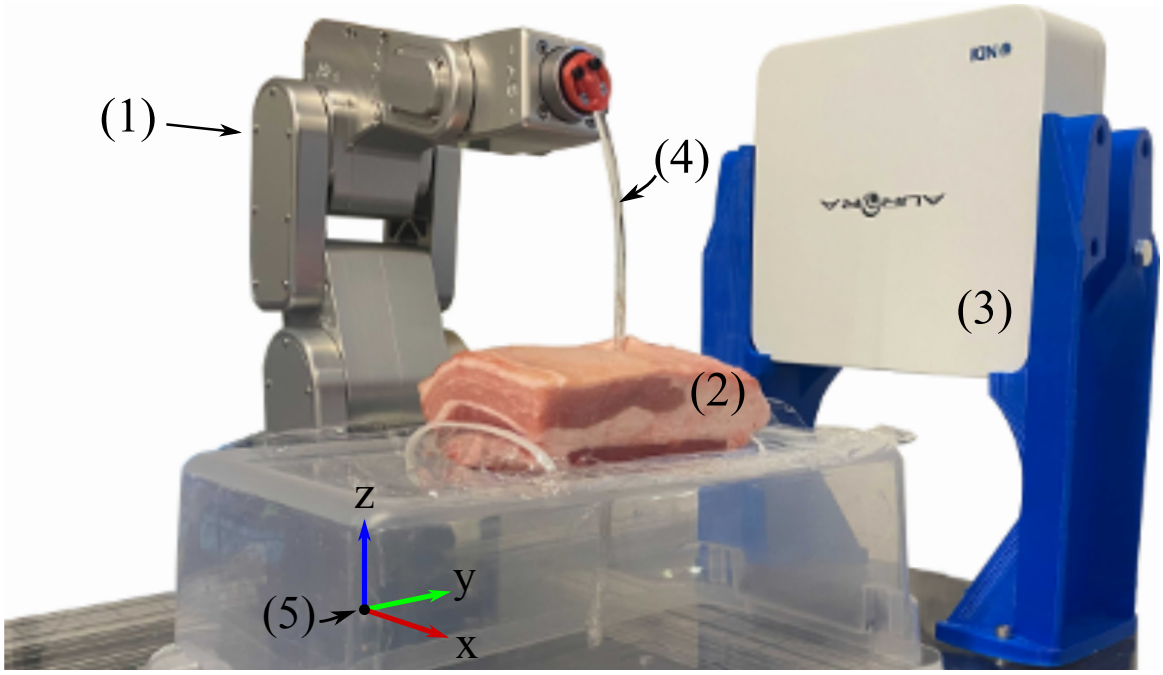


Figure 4.6: The experimental setup showing the robotic arm (1) with the tool attached at the end-effector, the porcine tissue sample (2), the electromagnetic tracking system (3), the acrylic tool with embedded sensor (4), and the robot base frame (5).

electromagnetic tool tracking system (Aurora, Northern Digital, Waterloo, Canada) as pictured in Fig 4.6. The experiments were performed on ex-vivo porcine tissue that included the skin, layers of muscle and fat, and connective tissue. The electromagnetic tracking system uses a small sensor embedded in the tip of the tool, the other end of which is attached to the robot end-effector. Due to potential eddy currents interfering with the electromagnetic tracking system, the tool is a 231.8 mm acrylic rod with a 3.2 GPa Young's modulus and 2.38 mm radius. The flexible acrylic rod demonstrates the capability of the proposed formulation to enable the use of more flexible tools in the future with a semi-autonomous platform. All of the experiments begin with the tool already inserted through a straight entry path in the tissue. The tissue is kept stationary during each trial while the robot manoeuvres the tool along the trajectory. The sensor is within the electromagnetic field it returns the 3D tip coordinates with an approximate accuracy of 0.70 mm, at a frequency of 14.7 Hz.

Three pre-defined tooltip trajectories are created to induce both translation and rotation of the tool's base, as shown in Fig. 4.7(a). The trajectories are expressed relative

Table 4.3: Mean and standard deviation of tip error in mm

Tissue Depth	Trajectory 1	Trajectory 2	Trajectory 3
20 mm	1.21 ± 1.06	1.28 ± 1.03	2.45 ± 1.30
35 mm	2.23 ± 0.94	2.97 ± 0.73	-

to the robot's base, located at $x = 0$ and $y = 0$. The tissue is 120 mm above the robot's base, and the trajectories are on average 50 mm below the tissue. The robot's end-effector starts at position $x = 0$ and $y = 280$ mm. Two different porcine tissue samples having different thicknesses of 20 and 35 mm are used. For each tissue and trajectory, a total of five trials are performed, thus there are 5 experimental scenarios and 25 trials in total.

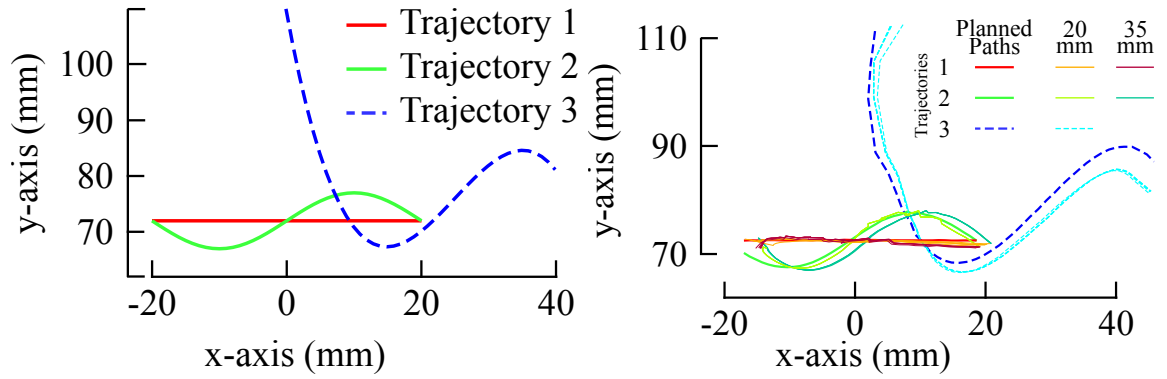
For each experimental scenario, the optimizer determines the corresponding robot's end-effector position and orientation according to the four cost functions defined earlier. The robot is then controlled to follow the calculated end-effector position while the tooltip position is measured.

4.4.1 Model Identification

The flexural rigidity of the tool and sheath are known. The tissue stiffness K, however, must be determined prior to experiments. To this end, the tool is inserted into the tissue initially without the sheath present, the robot end-effector is controlled to follow a predefined straight line while the electromagnetic tracking system collects 3D tip position information. The measured robot end-effector trajectory is input to the tool-bending model. The tooltip position is calculated for a range of potential tissue stiffnesses and compared with the experimental results. The stiffness value yielding the lowest least square error between the model and measured tip positions is selected. The obtained stiffness value is $K = 27.5$ kPa.

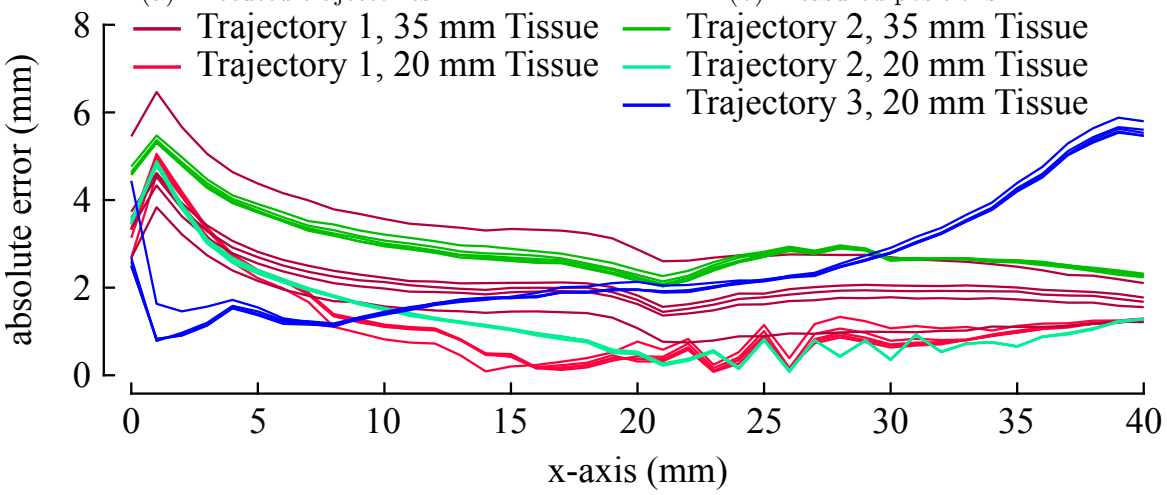
4.4.2 Experimental Results

The error between simulated and measured tooltip positions is calculated for each point along the trajectory. Table 4.3 summarizes the mean error and standard devi-



(a) Executed trajectories.

(b) Measured positions.



(c) Error during the experimental trials.

Figure 4.7: The trajectories used are shown in (a), trajectory 1 is red, trajectory 2 is green, and trajectory 3 is blue. The measured tooltip positions are plotted in (b). The error for each trial of each trajectory is shown in (c), red lines indicate trajectory 1, green for trajectory 2, and blue for trajectory 3. Note that the x-axes have been aligned in (c) to better show the results since each trajectory spans 40 mm.

ation for each experimental scenario. Note that out-of-plane error is neglected since a 2D scenario is being considered, it is presumed that a 3D scenario would have a slightly higher error rate due to the increased dimensionality. The results are consistent across trials with an overall mean error of 2.03 mm. Based on the error for each point along the trajectories, as Fig. 4.7 shows, the framework is most accurate close to the entry point, the error appears to increase as the distance between the tooltip and the original entry point increases.

4.5 Conclusion and Discussion

Proposed in this chapter is a model-based multi-objective optimizer used to predict the bending of the nephroscope. The optimizer aims to minimize four cost functions to accurately trace the predefined trajectory.

Four simulation scenarios are used to validate the purpose of each cost function for the multi-objective optimizer. It is found that the tissue displacement and the tool strain energy are highly correlated, however, both are necessary to accurately trace the trajectory. The 25 experiments performed on samples of porcine tissue show an average tracking error of 2.03 mm with a standard deviation of 1.23 mm. Furthermore, the tracking error tends to increase with tissue thickness, the error also increased with the distance from the entry path. Overall the results indicate that the proposed framework can accurately trace a predefined path with minimal error.

Chapter 5

Conclusions and Recommendations

GAINING kidney access is one of the most challenging components of PCNL (percutaneous nephrolithotomy) and is integral to patient outcomes. Incorrect tool steering when gaining kidney access is the leading cause of many complications related to PCNL. This thesis proposed robotic assistance to both teach novice surgeons and to provide assistance during PCNL through sub-task automation. First, haptic assistance and subtask automation are implemented to help simplify the procedure while training novices. The haptic assistance is derived from an expert demonstration. A novel inverse kinematic formulation is used to create a remote centre of motion (RCM) at the entry point in the tissue, which simplifies the task of controlling the tool such that the surgeon only controls the 3D tooltip position. Additionally, this allows a 6 degree of freedom (DOF) robotic arm to be controlled by a 3 DOF haptic device. Through a set of evaluation criteria, the effectiveness of training with the haptic feedback is demonstrated. Users are able to achieve better results when the haptic feedback is applied, and once haptic feedback is removed, those trained with haptic feedback perform better than those who never experienced the haptic feedback.

While the ideal path for applying haptic feedback is based on an expert demonstration, this approach may not be the most suitable in an autonomous scenario. To that end, a multi-objective optimizer (NSGA-II) is applied to plan a variety of appropriate paths for the robotic agent to follow. The algorithm considers four objective functions: path length, tissue displacement, obstacle proximity, and path smoothness. Each objective is used specifically for gaining kidney access during PCNL. The algorithm aims to minimize the overall path length since longer paths require longer procedure times and more tool movement puts the patient at increased risks. Minimizing tissue displacement reduces the risk for tissue trauma at the entry point; tissue trauma at the entry point can lead to additional bleeding and infection. Maximizing the distance to obstacles ensures a collision-free execution of the algorithm, further ensuring that there is no additional tissue trauma during the procedure. Finally, minimizing the path smoothness parameter ensures that there are no sudden movements or jerks during the procedure. This algorithm is able to provide an ideal set of paths from

which the expert surgeon can select one for execution. This optimization method keeps a surgeon in control of the trade-offs that the various final solutions offer, for example: since all of the path lengths are minimized and may only vary slightly in their final length, this may not be as important of a factor for the surgeon when selecting the final path. The surgeon may choose to select a path that provides the least tissue displacement or maintains the furthest distance from obstacles.

With an appropriate path planned, a robot agent can autonomously execute the task of gaining kidney access. Although, an additional challenge to consider when performing this task autonomously is accurately tracking the trajectory of the tool. This problem can be further complicated by tool bending that may occur as a result of its interaction with the tissue at the entry point. A tool/tissue interaction model is then developed to accurately determine the bending of the tool for a given position and orientation of the base. The tool is modelled as a cantilever beam with one fixed and one free end, and the bending of the tool can be described by a weighted sum of the vibration modes. Conversely, the exact displacement of the tissue is unknown without the tool bending, thus it is modelled as a function of the tool displacement. These two models are combined as potential energy functions. The weights, or modal coordinates, for the tool bending, are determined by minimizing the energy in the system. This formulation was evaluated on a set of simulations which demonstrated that while some of the cost functions may have similar effects on the path planning all four cost functions need to be implemented to ensure patient safety and accurate trajectory tracking. Then NSGA-II is used to find suitable tool poses that account for the bending such that the tip follows predefined trajectories. Physical experiments demonstrated the accuracy of this model for a real-world scenario.

Each of the above contributions aim to provide robotic assistance to PCNL, particularly while gaining access to kidney stones. Their effectiveness has been thoroughly evaluated through experiments. The inverse kinematics, from **Chapter 2**, successfully decoupled orientation from position, while haptic feedback proved to be an excellent addition when teaching novices. Nevertheless, the proposed framework can be

improved, specifically regarding the visual feedback.

The cyber-physical simulator proposed in **Chapter 2** does not recreate the visual feedback that would be provided in an operating room. The phantom kidney used during the experiments gives users a full visual view of both the outside and inside of the kidney, except when they were entering the kidney. During PCNL, there is generally only 2D image feedback from a fluoroscope or ultrasound machine. Future work should focus on limiting the visual information to ultrasound and incorporating a semi-autonomous visual servoing system. This semi-autonomous addition to the framework could further reduce the physical and mental workload on a novice by automating control of the ultrasound probe.

Another improvement can be sought by integrating the proposed framework into Marion Surgical's K181 PCNL surgical simulator [26]. The simulator is able to generate accurate scenarios based on individual patient's preoperative imaging. These scenarios will allow expert surgeons to create the optimal trajectory in the simulator. The trajectory is then be used to create the force feedback to guide a novice and teach them how to perform the procedure on the patient. Expert surgeons would be able to mentor and teach novices remotely through the simulators.

Semi-autonomous visual servoing would also provide additional feedback on kidney stone locations and could be implemented to further improve tool tracking during the procedure. The ultrasound image information could be incorporated to adjust the accuracy of the tool bending mode. By providing online tooltip tracking, the tool bending model parameters could be updated and used to adjust the tool pose to maintain accurate tool positioning.

The tool/tissue interaction model demonstrated accurate tooltip tracking for 2D scenarios. A key aspect of future work should replace the tool bending model with a compatible 3D version. Other than the tool bending model, the remaining formulation of the tool/tissue interaction model can be converted to a 3-dimensional scenario with minimal adjustments. The 3D formulation is expected to have a slightly higher absolute error as a result of possible deflection in an additional direction. The 3D

tool/tissue interaction model combined with semi-autonomous visual servoing is anticipated to prove highly effective in accurate trajectory tracking and implementing autonomous kidney access. This framework may also be implemented to additional steps during the procedure since it allows a robotic agent to accurately execute a carefully planned task.

Finally, possible complications during the procedure come from changes in the position of the kidney or misalignment of the simulated patient position compared to the patient's position in the operating room. Further, patient breathing, other natural phenomena, or the tool can displace the kidney during the procedure. Accounting for these position errors is beneficial as it would further improve patient safety and allow a robotic agent to account for the position errors in an online environment. The optimal trajectory would be adjusted based on visual feedback from ultrasound images to potentially realign the trajectory to the new target position. The visual feedback could also be used to adjust for other position errors within the kidney or caused by breathing. Making small online adjustments to an optimal path would improve autonomous tool steering during PCNL and simplifies the challenges of accurately aligning the kidney with the planned trajectory.

The semi-autonomous PCNL frameworks outlined in this thesis provide a variety of possible improvements for both training and live scenarios. Combining this framework with the possible advancements identified in this thesis, could vastly improve the outcomes from PCNL by providing additional training opportunities and automating subtasks during the procedure. This simplifies the mental and physical workload on the surgeon and is capable of ensuring a safe and accurate procedure. The addition of the tool bending model allows smaller more flexible tools to be used during autonomous PCNL. Narrower tools have been demonstrated to reduce bleeding and tissue damage since they require smaller incisions and are less likely to puncture larger veins or arteries. These improvements can one day lead to a fully autonomous kidney stone removal system, which has the potential to greatly improve the outcomes of percutaneous nephrolithotomy.

References

[1] S. R. Khan, M. S. Pearle, W. G. Robertson, G. Gambaro, B. K. Canales, S. Doizi, O. Traxer, and H.-G. Tiselius, "Kidney stones," *Nature Reviews Disease Primers*, vol. 2, no. 1, pp. 1–23, 2016.

[2] C. D. Scales, A. C. Smith, J. M. Hanley, and C. S. Saigal, "Prevalence of kidney stones in the united states," *European urology*, vol. 62, no. 1, pp. 160–165, 2012.

[3] G. Zeng, Z. Mai, S. Xia, Z. Wang, K. Zhang, L. Wang, Y. Long, J. Ma, Y. Li, S. P. Wan, W. Wu, Y. Liu, Z. Cui, Z. Zhao, J. Qin, T. Zeng, Y. Liu, X. Duan, X. Mai, Z. Yang, Z. Kong, T. Zhang, C. Cai, Y. Shao, Z. Yue, S. Li, J. Ding, S. Tang, and Z. Ye, "Prevalence of kidney stones in china: an ultrasonography based cross-sectional study," *BJU international*, vol. 120, no. 1, pp. 109–116, 2017.

[4] T. Alelign and B. Petros, "Kidney stone disease: an update on current concepts," *Advances in urology*, vol. 2018, 2018.

[5] J. B. Ziemba and B. R. Matlaga, "Guideline of guidelines: kidney stones," *BJU international*, vol. 116, no. 2, pp. 184–189, 2015.

[6] A. Aydın, A. Al-Jabir, B. Smith, and K. Ahmed, *Training in Percutaneous Nephrolithotomy*. Singapore: Springer Singapore, 2020, pp. 195–202. [Online]. Available: https://doi.org/10.1007/978-981-15-0575-1_21

[7] M. Vijayakumar, A. Ganpule, A. Singh, R. Sabnis, and M. Desai, "Review of techniques for ultrasonic determination of kidney stone size," *Research and*

reports in urology, vol. 10, p. 57, 2018.

[8] T. Kanno, M. Kubota, H. Sakamoto, R. Nishiyama, T. Okada, Y. Higashi, and H. Yamada, "The efficacy of ultrasonography for the detection of renal stone," *Urology (Ridgewood, N.J.)*, vol. 84, no. 2, pp. 285–288, 2014.

[9] D. Rodríguez and D. E. Sacco, "Minimally invasive surgical treatment for kidney stone disease," *Advances in chronic kidney disease*, vol. 22, no. 4, pp. 266–272, 2015.

[10] R. Takazawa, S. Kitayama, and T. Tsujii, "Appropriate kidney stone size for ureteroscopic lithotripsy: When to switch to a percutaneous approach," *World journal of nephrology*, vol. 4, no. 1, p. 111, 2015.

[11] I. M. Sabler, I. Katafigiotis, O. N. Gofrit, and M. Duvdevani, "Present indications and techniques of percutaneous nephrolithotomy: What the future holds?" *Asian journal of urology*, vol. 5, no. 4, pp. 287–294, 2018.

[12] R. Munver, F. C. Delvecchio, G. E. Newman, and G. M. Preminger, "Critical analysis of supracostal access for percutaneous renal surgery," *The Journal of urology*, vol. 166, no. 4, pp. 1242–1246, 2001.

[13] D. R. Webb, *Percutaneous Nephrostomy*. Cham: Springer International Publishing, 2016, pp. 55–82. [Online]. Available: https://doi.org/10.1007/978-3-319-22828-0_4

[14] ——, *Routine PCNL*. Cham: Springer International Publishing, 2016, pp. 83–110. [Online]. Available: https://doi.org/10.1007/978-3-319-22828-0_5

[15] A. Hoznek, *Positioning During PNL*. Singapore: Springer Singapore, 2020, pp. 43–52. [Online]. Available: https://doi.org/10.1007/978-981-15-0575-1_6

[16] T. Knoll, F. Daels, J. Desai, A. Hoznek, B. Knudsen, E. Montanari, C. Scoffone, A. Skolarikos, and K. Tozawa, "Percutaneous nephrolithotomy: technique," *World Journal of Urology*, vol. 35, no. 9, pp. 1361–1368, 2017.

[17] I. Kyriazis, V. Panagopoulos, P. Kallidonis, M. Ozsoy, M. Vasilas, and E. Liatsikos, "Complications in percutaneous nephrolithotomy.(report)," *World Journal of Urology*, vol. 33, no. 8, pp. 1069–1077, 2015-08-01.

[18] M. S. Michel, L. Trojan, J. J. Rassweiler, and A. Breda, "Complications in percutaneous nephrolithotomy," *European Urology*, vol. 51, no. 4, pp. 899,906, 2007-04.

[19] S. Rais-Bahrami, J. I. Friedlander, B. D. Duty, Z. Okeke, and A. D. Smith, "Difficulties with access in percutaneous renal surgery," *Therapeutic Advances in Urology*, vol. 3, no. 2, pp. 59–68, 2011.

[20] C. Chu, S. Masic, M. Usawachintachit, W. Hu, W. Yang, M. Stoller, J. Li, and T. Chi, "Ultrasound-guided renal access for percutaneous nephrolithotomy: A description of three novel ultrasound-guided needle techniques," *Journal of Endourology*, vol. 30, 09 2015.

[21] A. R. El-Nahas, A. A. Shokeir, A. M. El-Assmy, T. Mohsen, A. M. Shoma, I. Eraky, M. R. El-Kenawy, and H. A. El-Kappany, "Post-percutaneous nephrolithotomy extensive hemorrhage: A study of risk factors," *The Journal of Urology*, vol. 177, no. 2, pp. 576–579, 2007.

[22] V. G. Bird, B. Fallon, and H. N. Winfield, "Practice patterns in the treatment of large renal stones," *Journal of endourology*, vol. 17, no. 6, pp. 355–363, 2003.

[23] C. Ng, "Training in percutaneous nephrolithotomy: The learning curve and options," *Arab journal of urology*, vol. 12, no. 1, pp. 54–57, 2014.

[24] J. de la Rosette, M. Laguna, J. Rassweiler, and P. Conort, "Training in percutaneous nephrolithotomy—a critical review," *European urology*, vol. 54, no. 5, pp. 994–1003, 2008.

[25] S.-Y. Lee, M.-L. Wang, Y.-C. Wong, C.-H. Wu, and L.-J. Wang, "Prolonged international normalized ratio and vascular injury at divisional level predict

embolization failures of patients with iatrogenic renal vascular injuries," *Scientific reports*, vol. 9, no. 1, pp. 17 108–10, 2019.

[26] B. Sainsbury, M. Łącki, M. Shahait, M. Goldenberg, A. Baghdadi, L. Cavuoto, J. Ren, M. Green, J. Lee, T. D. Averch, and C. Rossa, "Evaluation of a virtual reality percutaneous nephrolithotomy (pcnl) surgical simulator," *Frontiers in Robotics and AI*, vol. 6, p. 145, 2020.

[27] A. Al-Jabir, A. Aydin, H. Al-Jabir, M. S. Khan, P. Dasgupta, and K. Ahmed, "Current status of wet lab and cadaveric simulation in urological training: A systematic review," *Canadian Urological Association Journal*, vol. 14, no. 11, p. E594, 2020.

[28] W. L. Strohmaier and A. Giese, "Ex vivo training model for percutaneous renal surgery," *Urological research*, vol. 33, no. 3, pp. 191–193, 2005.

[29] A. Häcker, G. Wendt-Nordahl, P. Honeck, M. Michel, P. Alken, and T. Knoll, "A biological model to teach percutaneous nephrolithotomy technique with ultrasound-and fluoroscopy-guided access," *Journal of endourology*, vol. 21, no. 5, pp. 545–550, 2007.

[30] M. Vijayakumar, S. Balaji, A. Singh, A. Ganpule, R. Sabnis, and M. Desai, "A novel biological model for training in percutaneous renal access," *Arab journal of urology*, vol. 17, no. 4, pp. 292–297, 2019.

[31] S. Jutzi, F. Imkamp, M. A. Kuczyk, U. Walcher, U. Nagele, and T. R. W. Herrmann, "New ex vivo organ model for percutaneous renal surgery using a laparoendoscopic training box: the sandwich model," *World journal of urology*, vol. 32, no. 3, pp. 783–789, 2014.

[32] M. Sinha and V. Krishnamoorthy, "Use of a vegetable model as a training tool for pcnl puncture," *Indian journal of urology: IJU: journal of the Urological Society of India*, vol. 31, no. 2, p. 156, 2015.

[33] S. Mishra, A. Kurien, A. Ganpule, V. Muthu, R. Sabnis, and M. Desai, "Percutaneous renal access training: content validation comparison between a live porcine and a virtual reality (vr) simulation model: Percutaneous renal access training," *BJU international*, vol. 106, no. 11, pp. 1753–1756, 2010.

[34] P. Ahmed, Kamran, M. Aydin, Abdullatif, F. Dasgupta, Prokar, F. U. Khan, Muhammad Shamim, and F. U. McCabe, John E., "A novel cadaveric simulation program in urology," *Journal of surgical education*, vol. 72, no. 4, pp. 556–565, 2015.

[35] I. Badash, K. Burtt, C. A. Solorzano, and J. N. Carey, "Innovations in surgery simulation: a review of past, current and future techniques," *Annals of translational medicine*, vol. 4, no. 23, 2016.

[36] E. D. Matsumoto, "Low-fidelity ureteroscopy models," *Journal of endourology*, vol. 21, no. 3, pp. 248–251, 2007.

[37] "Uro/perc mentor platforms," 3D Systems, 2017. [Online]. Available: https://simbionix.com/simulators

[38] "Mediskills standard scope trainer – sst – endourological skills-training model," Medi Skills Models, 2015. [Online]. Available: https://www.mediskills.com/products/scope-trainer/

[39] S. M. Botden, S. N. Buzink, M. P. Schijven, and J. J. Jakimowicz, "Augmented versus virtual reality laparoscopic simulation: what is the difference?" *World journal of surgery*, vol. 31, no. 4, pp. 764–772, 2007.

[40] B. E. Knudsen, E. D. Matsumoto, B. H. Chew, B. Johnson, V. Margulis, J. A. Cadeddu, M. S. Pearle, S. E. Pautler, and J. D. Denstedt, "A randomized, controlled, prospective study validating the acquisition of percutaneous renal collecting system access skills using a computer based hybrid virtual reality surgical simulator: phase i," *The Journal of urology*, vol. 176, no. 5, pp. 2173–2178, 2006.

[41] A. Papatsoris, T. Shaikh, D. Patel, A. Bourdoumis, C. Bach, N. Buchholz, J. Masood, and I. Junaid, "Use of a virtual reality simulator to improve percutaneous renal access skills: a prospective study in urology trainees," *Urologia internationalis*, vol. 89, no. 2, pp. 185–190, 2012.

[42] "Virtamed uros™," VirtaMed, 2021. [Online]. Available: https://www.virtamed.com/en/medical-training-simulators/uros/

[43] A. Lelevé, T. McDaniel, and C. Rossa, "Haptic training simulation," *Frontiers in Virtual Reality*, vol. 1, 2020.

[44] S. Chen, Z. Li, Y. Lin, F. Wang, and Q. Cao, "Automatic ultrasound scanning robotic system with optical waveguide-based force measurement," *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, no. 6, pp. 1015–1025, 2021.

[45] R. Elek, T. D. Nagy, D. A. Nagy, B. Takács, P. Galambos, I. Rudas, and T. Haidegger, "Robotic platforms for ultrasound diagnostics and treatment," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 1752–1757.

[46] E. Fomekong, S. E. Safi, and C. Raftopoulos, "Spine navigation based on 3-dimensional robotic fluoroscopy for accurate percutaneous pedicle screw placement: a prospective study of 66 consecutive cases," *World neurosurgery*, vol. 108, pp. 76–83, 2017.

[47] J. Sandoval, G. Poisson, and P. Vieyres, "A new kinematic formulation of the rcm constraint for redundant torque-controlled robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4576–4581.

[48] Z. Wang, Z. Liu, Q. Ma, A. Cheng, Y.-h. Liu, S. Kim, A. Deguet, A. Reiter, P. Kazanzides, and R. H. Taylor, "Vision-based calibration of dual rcm-based robot arms in human-robot collaborative minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 672–679, 2017.

[49] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic active constraints for surgical robots using vector-field inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, 2019.

[50] M. M. Marinho, K. Harada, A. Morita, and M. Mitsuishi, "Smartarm: Integration and validation of a versatile surgical robotic system for constrained workspaces," *The international journal of medical robotics and computer assisted surgery*, vol. 16, no. 2, pp. e2053–n/a, 2020.

[51] R. Elek, T. D. Nagy, D. Á. Nagy, T. Garamvölgyi, B. Takács, P. Galambos, J. K. Tar, I. J. Rudas, and T. Haidegger, "Towards surgical subtask automation—blunt dissection," in *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2017, pp. 000 253–000 258.

[52] T. D. Nagy, M. Takács, I. J. Rudas, and T. Haidegger, "Surgical subtask automation—soft tissue retraction," in *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2018, pp. 000 055–000 060.

[53] Q. Ma, E. Kobayashi, H. Suenaga, K. Hara, J. Wang, K. Nakagawa, I. Sakuma, and K. Masamune, "Autonomous surgical robot with camera-based markerless navigation for oral and maxillofacial surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 1084–1094, 2020.

[54] S.-Y. Woo, S.-J. Lee, J.-Y. Yoo, J.-J. Han, S.-J. Hwang, K.-H. Huh, S.-S. Lee, M.-S. Heo, S.-C. Choi, and W.-J. Yi, "Autonomous bone reposition around anatomical landmark for robot-assisted orthognathic surgery," *Journal of Cranio-Maxillofacial Surgery*, vol. 45, no. 12, pp. 1980–1988, 2017.

[55] J. J. Rassweiler, M. Müller, M. Fangerau, J. Klein, A. S. Goezen, P. Pereira, H.-P. Meinzer, and D. Teber, "ipad-assisted percutaneous access to the kidney using marker-based navigation: Initial clinical experience," *European urology*, vol. 61, no. 3, pp. 628–631, 2011.

[56] M. Müller, M.-C. Rassweiler, J. Klein, A. Seitel, M. Gondan, M. Baumhauer, D. Teber, J. J. Rassweiler, H.-P. Meinzer, and L. Maier-Hein, "Mobile augmented reality for computer-assisted percutaneous nephrolithotomy," *International journal of computer assisted radiology and surgery*, vol. 8, no. 4, pp. 663–675, 2013.

[57] Z.-C. Li, K. Li, H.-L. Zhan, K. Chen, J. Gu, and L. Wang, "Augmenting intraoperative ultrasound with preoperative magnetic resonance planning models for percutaneous renal access," *Biomedical engineering online*, vol. 11, no. 1, pp. 1–16, 2012.

[58] F. Ferraguti, M. Minelli, S. Farsoni, S. Bazzani, M. Bonfe, A. Vandanjon, S. Puliatti, G. Bianchi, and C. Secchi, "Augmented reality and robotic-assistance for percutaneous nephrolithotomy," *IEEE robotics and automation letters*, vol. 5, no. 3, pp. 4556–4563, 2020.

[59] I. Paranawithana, H.-Y. Li, S. Foong, U.-X. Tan, L. Yang, T. S. K. Lim, and F. C. Ng, "Ultrasound-guided involuntary motion compensation of kidney stones in percutaneous nephrolithotomy surgery," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 1123–1129.

[60] H.-Y. Li, I. Paranawithana, Z. H. Chau, L. Yang, T. S. K. Lim, S. Foong, F. C. Ng, and U.-X. Tan, "Towards to a robotic assisted system for percutaneous nephrolithotomy," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 791–797.

[61] R. Swearingen, A. Sood, R. Madi, Z. Klaassen, K. Badani, J. S. Elder, K. Wood, A. Hemal, and K. R. Ghani, "Zero-fragment nephrolithotomy: a multi-center evaluation of robotic pyelolithotomy and nephrolithotomy for treating renal stones," *European urology*, vol. 72, no. 6, pp. 1014–1021, 2017.

[62] K. R. Ghani, Q.-D. Trinh, W. Jeong, A. Friedman, Y. Lakshmanan, M. Menon, and J. S. Elder, "Robotic nephrolithotomy and pyelolithotomy with utilization

of the robotic ultrasound probe," *International braz j urol*, vol. 40, no. 1, pp. 125–126, 2014.

[63] R. Madi and A. Hemal, "Robotic pyelolithotomy, extended pyelolithotomy, nephrolithotomy, and anatrophic nephrolithotomy," *Journal of endourology*, vol. 32, no. S1, pp. S–73, 2018.

[64] C. Gruijthuijsen, R. Colchester, A. Devreker, A. Javaux, E. Maneas, S. Noimark, W. Xia, D. Stoyanov, D. Reynaerts, J. Deprest, S. Ourselin, A. Desjardins, T. Vercauteren, and E. Vander Poorten, "Haptic guidance based on all-optical ultrasound distance sensing for safer minimally invasive fetal surgery," *Journal of medical robotics research*, vol. 3, no. 03n04, p. 1841001, 2018.

[65] M. B. Molinero, G. Dagnino, J. Liu, W. Chi, M. E. Abdelaziz, T. M. Kwok, C. Riga, and G.-Z. Yang, "Haptic guidance for robot-assisted endovascular procedures: implementation and evaluation on surgical simulator," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5398–5403.

[66] S. Seung, H. Choi, J. Jang, Y. S. Kim, J.-O. Park, S. Park, and S. Y. Ko, "Virtual wall–based haptic-guided teleoperated surgical robotic system for single-port brain tumor removal surgery," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 231, no. 1, pp. 3–19, 2017.

[67] L. Xiong, C. B. Chng, C. K. Chui, P. Yu, and Y. Li, "Shared control of a medical robot with haptic guidance," *International journal for computer assisted radiology and surgery*, vol. 12, no. 1, pp. 137–147, 2016.

[68] O. Halabi and Y. Halwani, "Design and implementation of haptic virtual fixtures for preoperative surgical planning," *Displays*, vol. 54, pp. 9–19, 2018.

[69] Y.-Y. Juo, A. Abiri, J. Pensa, S. Sun, A. Tao, J. Bisley, W. Grundfest, and E. Dutson, "Center for advanced surgical and interventional technology multi-

modal haptic feedback for robotic surgery," in *Handbook of robotic and image-guided surgery*. Elsevier, 2020, pp. 285–301.

[70] B. Kent, A. Cusipag, and C. Rossa, "Tissue discrimination through force-feedback from impedance spectroscopy in robot-assisted surgery," in *Smart Multimedia*, ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 274–285.

[71] A. Abiri, Y.-Y. Juo, A. Tao, S. J. Askari, J. Pensa, J. W. Bisley, E. P. Dutson, and W. S. Grundfest, "Artificial palpation in robotic surgery using haptic feedback," *Surgical endoscopy*, vol. 33, no. 4, pp. 1252–1259, 2019.

[72] Q. Ouyang, J. Wu, S. Sun, J. Pensa, A. Abiri, E. Dutson, and J. W. Bisley, "Bio-inspired haptic feedback for artificial palpation in robotic surgery," *IEEE Transactions on Biomedical Engineering*, 2021.

[73] M. O'Malley, A. Gupta, M. Gen, and Y. Li, "Shared control in haptic systems for performance enhancement and training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 75–85, 2006.

[74] C. Wagner and R. Howe, "Mechanisms of performance enhancement with force feedback," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. IEEE, 2005, pp. 21–29.

[75] L. Panait, E. Akkary, R. L. Bell, K. E. Roberts, S. J. Dudrick, and A. J. Duffy, "The role of haptic feedback in laparoscopic simulation training," *Journal of Surgical Research*, vol. 156, no. 2, pp. 312–316, 2009.

[76] S. Botden, F. Torab, S. Buzink, and J. Jakimowicz, "The importance of haptic feedback in laparoscopic suturing training and the additive value of virtual reality simulation," *Surgical endoscopy*, vol. 22, no. 5, pp. 1214–1222, 2008.

[77] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics Research*, S. Thrun, R. Brooks, and

H. Durrant-Whyte, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 49–64.

[78] A. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Industrial Robot: An International Journal*, vol. 31, no. 6, pp. 499,508, 2004-12-01.

[79] S. A. H. Perez, K. Harada, and M. Mitsuishi, "Haptic virtual fixtures to assist endonasal micro robotic surgery through virtual reality simulation," in *2018 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*. IEEE, 2018-12, pp. 1,3.

[80] H. Su, W. Shang, G. Li, N. Patel, and G. S. Fischer, "An mri-guided telesurgery system using a fabry-perot interferometry force sensor and a pneumatic haptic device," *Annals of Biomedical Engineering*, vol. 45, no. 8, pp. 1917–1928, Aug 2017.

[81] A. Chowriappa, R. Wirz, Y. W. Seo, A. Reddy, T. Kesavadas, P. Scott, and K. Guru, "A predictive model for haptic assistance in robot assisted trocar insertion," in *2013 World Haptics Conference (WHC)*. IEEE, 2013-04, pp. 121,126.

[82] Y. Yang, S. Jiang, Z. Yang, W. Yuan, H. Dou, W. Wang, D. Zhang, and Y. Bian, "Design and analysis of a tendon-based computed tomography–compatible robot with remote center of motion for lung biopsy," *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, vol. 231, no. 4, pp. 286–298, 2017.

[83] Y. He, P. Zhang, H. Jin, Y. Hu, and J. Zhang, "Type synthesis for remote center of motion mechanisms based on coupled motion of two degrees-of-freedom," *Journal of mechanical design (1990)*, vol. 138, no. 12, 2016.

[84] M. Hadavand, A. Mirbagheri, S. Behzadipour, and F. Farahmand, "A novel remote center of motion mechanism for the force-reflective master robot of haptic

tele-surgery systems," *The international journal of medical robotics + computer assisted surgery*, vol. 10, no. 2, pp. 129–139, 2014.

[85] K.-Y. Kim, H.-S. Song, J.-W. Suh, and J.-J. Lee, "A novel surgical manipulator with workspace-conversion ability for telesurgery," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 200–211, 2013.

[86] M. A. R. Garcia, R. A. Rojas, and F. Pirri, "Object-centered teleoperation of mobile manipulators with remote center of motion constraint," *IEEE robotics and automation letters*, vol. 4, no. 2, pp. 1745–1752, 2019.

[87] H. Sadeghian, F. Zokaei, and S. Hadian Jazi, "Constrained kinematic control in minimally invasive robotic surgery subject to remote center of motion constraint," *Journal of intelligent & robotic systems*, vol. 95, no. 3-4, pp. 901–913, 2018.

[88] S. Khansari-Zadeh and O. Khatib, "Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors," *Autonomous Robots*, vol. 41, no. 1, pp. 45,69, 2017-01.

[89] K. Liang, Y. Xing, J. Li, S. Wang, A. Li, and J. Li, "Motion control skill assessment based on kinematic analysis of robotic end-effector movements," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 1, pp. n/a,n/a, 2018-02.

[90] M. Zhou, D. Jones, S. Schwaitzberg, and C. Cao, "Role of haptic feedback and cognitive load in surgical skill acquisition," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 51, no. 11, pp. 631–635, 2007.

[91] L.-M. Su, D. Stoianovici, T. W. Jarrett, A. Patriciu, W. W. Roberts, J. A. Cadeddu, S. Ramakumar, S. B. Solomon, and L. R. Kavoussi, "Robotic percutaneous access to the kidney: comparison with standard manual access," *Journal of Endourology*, vol. 16, no. 7, pp. 471–475, 2002.

[92] B. Challacombe, A. Patriciu, J. Glass, M. Aron, T. Jarrett, F. Kim, P. Pinto, D. Stoianovici, N. Smeeton, R. Tiptaft, L. Kavoussi, and P. Dasgupta, "A randomized controlled trial of human versus robotic and telerobotic access to the kidney as the first step in percutaneous nephrolithotomy," *Computer aided surgery*, vol. 10, no. 3, pp. 165–171, 2005.

[93] P. Raja and S. Pugazhenthi, "Optimal path planning of mobile robots: A review," *International journal of physical sciences*, vol. 7, no. 9, pp. 1314–1320, 2012.

[94] M. N. Zafar and J. Mohanta, "Methodology for path planning and optimization of mobile robots: A review," *Procedia computer science*, vol. 133, pp. 141–152, 2018.

[95] F. Ahmed and K. Deb, "Multi-objective path planning using spline representation," in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 1047–1052.

[96] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles," *IEEE transactions on automation science and engineering*, vol. 7, no. 1, pp. 167–172, 2010.

[97] Z. Zhao, B. Liu, C. Zhang, and H. Liu, "An improved adaptive nsga-ii with multi-population algorithm," *Applied Intelligence*, vol. 49, no. 2, pp. 569–580, 2019.

[98] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 1, pp. 1–28, 2014.

[99] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[100] K. Kumar and K. Deb, "Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems," *Complex syst*, vol. 9, pp. 431–454, 1995.

[101] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y.-S. Ong, "A new decomposition-based nsga-ii for many-objective optimization," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 48, no. 7, pp. 1191–1210, 2017.

[102] M. A. Beaghler, M. W. Poon, J. W. Dushinski, and J. E. Lingeman, "Expanding role of flexible nephroscopy in the upper urinary tract," *Journal of Endourology*, vol. 13, no. 2, pp. 93–97, 1999, pMID: 10213102.

[103] C. Rossa and M. Tavakoli, "Issues in closed-loop needle steering," *Control Engineering Practice*, vol. 62, pp. 55–69, 2017.

[104] W. Park, J. S. Kim, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian, "Diffusion-based motion planning for a nonholonomic flexible needle model," in *Int. Conference on Robotics and Automation*. IEEE, 2005, pp. 4600–4605.

[105] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006.

[106] M. Khadem, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli, "Geometric control of 3d needle steering in soft-tissue," *Automatica*, vol. 101, pp. 36–43, 2019.

[107] H. Lee and J. Kim, "Estimation of flexible needle deflection in layered soft tissues with different elastic moduli," *Medical & Biological Engineering & Computing*, vol. 52, no. 9, pp. 729–740, 2014.

[108] P. Moreira and S. Misra, "Biomechanics-based curvature estimation for ultrasound-guided flexible needle steering in biological tissues," *Annals of Biomedical Engineering*, vol. 43, no. 8, pp. 1716–1726, 2015.

[109] C. Rossa, M. Khadem, R. Sloboda, N. Usmani, and M. Tavakoli, "Adaptive quasi-static modelling of needle deflection during steering in soft tissue," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 916–923, 2016.

[110] M. Khadem, B. Fallahi, C. Rossa, R. S. Sloboda, N. Usmani, and M. Tavakoli, "A mechanics-based model for simulation and control of flexible needle insertion in soft tissue," in *International Conference on Robotics and Automation*, May 2015, pp. 2264–2269.

[111] G. G. Genta, *Vibration dynamics and control*, ser. Mechanical engineering series. New York: Springer, 1993.