



POLITECNICO DI MILANO
DEPARTMENT OF ELECTRONICS, INFORMATION AND
BIOENGINEERING DOCTORAL PROGRAM IN BIOENGINEERING

NOVEL PATH PLANNING AND AUTONOMOUS CONTROL
METHODS FOR NEEDLE STEERING SYSTEMS IN KEYHOLE
NEUROSURGERY

DOCTORAL THESIS

Author:
Alice Segato

Supervisors:

Prof. Elena De Momi
Prof. Francesco Calimeri

Tutor:

Prof. Giuseppe Pozzi

The Coordinator of the PhD Program:

Prof. Andrea Aliverti

Milano, August 2021

XXXIV

This thesis is dedicated to Pepito

"Somewhere, something incredible is waiting to be known."
Carl Sagan

Acknowledgements

I would like to thank my supervisor, Prof. Elena De Momi; I thank her for mentoring my professional endeavours. She has been a constant source of inspiration, has always encouraged scientific thinking, and instilled a sense of discipline in my work. I would like to thank my co-supervisor Francesco Calimeri. I would like to thank my PhD colleagues Marco Vidotto, Chiara Di Vece and Alessandro Casella. I would like to thank the partners of the Eden2020 project consortium for their valuable advice and collaboration during the project activities. In particular, clinicians Valentina Pieri, Antonella Castellano and Marco Riva, vets Marco Trovatelli and Prof. Davide Zani and engineers Stefano Galvan, Riccardo Secoli and Prof. Ferdinando Rodriguez Y Baena. I would like to thank all the Master thesis students with whom I had the pleasure to collaborate Luca Sestini, Marco Di Marzo, Sara Zucchelli, Valentina Corbetta and Irene Testa.

I would like to thank my friend Marina for the innumerable happy moments, the friendly support, the talks, the laughs, the happy hours and the meaningful experiences lived together. I would also like to thank Alessandro for sharing good wine and good conversation with me. I would like to thank Martina for the lovely evenings spent in the company of our friends Campari. I would like to thank Lorenzo for the beautiful emotions during these months full of stimuli and adventures. Thank you for always encouraging me never to give up and to reach this moment. I would like to thank my dear friend Fede for always being my best accomplice, for many years of warm friendship, calm guidance and being a firm rock of refuge.

I would like to thank my beloved puppy Pepito and his four-legged friends Mya and Luna. My Loyal, kind and loving friends that left paw prints on my heart.

Finally, I would like to thank my family for their continuous support during these three years. Without the inspiration, drive, and support you have given me, I might not achieve this important goal. Thanks Mum for being my rock and my example to follow and thanks Dad for always believing in me and pushing me. Thank you to my brother Andrea who, even if with irony and cynicism, has always proved to be at my side. Finally, I would like to thank my brother Fabio.

Ringraziamenti

Vorrei ringraziare il mio supervisore, la Prof.ssa Elena De Momi; la ringrazio per avermi fatto da mentore nei miei sforzi professionali. È stata una costante fonte di ispirazione e ha sempre incoraggiato il pensiero scientifico e instillato un senso di disciplina nel mio lavoro. Ringrazio, inoltre il mio co-supervisore Francesco Calimeri. Ringrazio i miei colleghi di dottorato Marco Vidotto, Chiara Di Vece e Alessandro Casella. Vorrei ringraziare tutti i partner del progetto Eden2020 per i loro preziosi consigli e la loro collaborazione durante le attività del progetto. In particolare, i clinici Valentina Pieri, Antonella Castellano e Marco Riva, i veterinari Marco Trovatelli e Davide Zani e gli ingegneri Stefano Galvan, Riccardo Secoli e il Prof. Ferdinando Rodriguez Y Baena. Ringrazio tutti i tesisti con cui ho avuto il piacere di collaborare Luca Sestini, Marco Di Marzo, Sara Zucchelli, Valentina Corbetta e Irene Testa.

Ringrazio la mia amica Marina per gli innumerevoli momenti felici, il supporto amichevole, le chiacchiere, le risate e le esperienze significative vissute insieme. Ringrazio anche Alessandro per aver condiviso con me del buon vino e delle belle chiacchierate. Ringrazio Martina per le belle serate passate in compagnia dei nostri amici Campari. Ringrazio Lorenzo per avermi fatto vivere tante belle emozioni in questi mesi pieni di stimoli e avventure. Grazie per avermi sempre incoraggiata a non mollare mai e a guardare avanti a testa alta per arrivare fino a questo momento. Vorrei ringraziare il mio caro amico Fede per essere sempre il mio miglior complice, per tanti anni di vera amicizia e di essere una solida roccia di rifugio.

Ringrazio il mio amato cucciolo Pepito e le sue amiche a quattro zampe Mya e Luna. Amici leali, gentili e affettuosi che hanno lasciato impronte di zampe nel mio cuore.

Infine, vorrei ringraziare la mia famiglia per il loro continuo sostegno durante questi tre anni. Senza l'ispirazione, la spinta e il sostegno che mi avete dato, non avrei potuto raggiungere questo importante obiettivo. Grazie mamma per essere la mia roccia e il mio esempio da seguire e grazie papà per aver sempre creduto in me e avermi sempre spronato. Grazie a mio fratello Andrea che, anche se con ironia e cinismo, ha sempre dimostrato di essere al mio fianco. Infine vorrei ringraziare mio fratello Fabio.

Summary

THE spread of the keyhole neurosurgery procedure is due to its capability to minimise patient trauma, risk of infection and recovery time. These features make keyhole neurosurgery applicable for a wide range of patients who are not suitable for open surgery. However, performing these procedures using traditional rigid needles may result in the impossibility to reach targets without colliding with anatomical obstacles (e.g., bones) or sensitive anatomical tissues (e.g., vital organs and vessels).

Over the last two decades, different research groups have focused their efforts on the development of needles able to steer inside the tissue. These needles can perform curvilinear trajectories planned to maximise the distance from sensitive anatomical structures to be avoided and reach targets otherwise inaccessible via rectilinear insertion paths. Differently from conventional needles, for which the insertion path can be planned and performed by the clinician on the basis of the target location and the patient anatomy, the complex kinematics of steerable needles make the manual path planning unbearable requiring the aid of automatic or semiautomatic path planning solutions. The pre-operative and intra-operative accuracy, and the robotic motion control robustness directly contribute to treatment outcomes and postoperative recovery.

On this background, the overall goal of this PhD thesis is to describe a safe and effective keyhole neurosurgery automatic planning framework for flexible neurosurgical needles. The framework is intended to support neurosurgeons during: 1) the pre-operative procedure to quickly determine the most appropriate surgical entry point and trajectory through the brain with minimised risk, 2) the intra-operative procedure to react to a dynamic environment, and 3) the robotic needle insertion to provide an appropriate control strategy.

In particular, the contributions of this PhD work are:

1. *An overview of different path planning techniques used for surgical steerable needle/catheters.* A systematic and careful literature search in major databases such as Pubmed, Scopus and Web of Science was carried out using "steerable catheter/needle" and "path planning" as main keywords. The analysis carried out for each paper concerns the clinical aim, the path planning method, the validation

procedure and the quantitative results. 84 studies out of 337 were identified, which actually made use of path planning algorithms for steerable needle/catheters. Such Paradigms Include graph-based, sampling-based, mathematical-based, learning-based, reasoning based and multifusion-based methods.

2. *A pre-operative path planning framework that couples learning-based and reasoning-based techniques in order to improve path planning performance.* In particular, an inductive learning model, relying on demonstrations performed by expert operators, is in charge of generating a set of paths as candidate solutions; then a deductive reasoning module selects the "best" starting point, according to explicit knowledge modeled over domain experts suggestions. This kind of coupling allows to transfer to the automated path planner most of the knowledge available at human level: the inductive learning module "catches", via demonstrations, expert capabilities that are hard to explicitly express (e.g., visual-spatial, bodily-kinesthetic), while the deductive module formally encodes what has been elaborated by the experts upon long-lasting practice (e.g., domain knowledge, best practices). Eventually, we assess the viability of the proposed approach, with in-silico test, proving that it stands or even outperform state-of-the-art approaches in terms of safety distance-from-obstacles respect, smoothness, and computational time.
3. *A realistic, time-bounded simulator based on position-based dynamics simulation that mocks brain deformations due to catheter insertion.* It maximises the probability of intra-operative path planning success by accounting for uncertainty in deformation models. The simulator uses a novel approach with respect to the literature, and it has proved to be a close match with real brain deformations through validation using recorded deformation data of in-vivo animal trials with a mean mismatch of $4.73 \pm 2.15\%$. The stability, accuracy, and real-time performance make this model suitable for creating a dynamic environment for keyhole neurosurgery path planning and intra-operative control.
4. *A novel path-replanner able to generate an obstacle-free and curvature bounded path at each time step during insertion, accounting for a constrained target pose and intra-operative anatomical deformation.* The generated path is achieved with inverse reinforcement learning methods for planning intra-operative feasible trajectory in the dynamic needle-tissue interactive keyhole neurosurgery environment. Simulation results performed on a realistic dataset show that our re-planning method can guide a needle with bounded curvature to a predefined target pose with an average targeting error of 1.34 ± 0.52 mm in position and 3.16 ± 1.06 degrees in orientation under a deformable simulated environment, with a re-planning time of 0.02 sec and a success rate of 100%. The target is considered reached with a tolerance of 2.5 mm in position and 10° in orientation.
5. *A motion controller that can steer a programmable bevel-tip needle to produce the desired trajectory in 3D.* This is performed thanks to the integration of the path planning model developed in 3D Unity into a system that consists of a programmable bevel-tip needle, a needle tracking via electromagnetic sensors and a needle steering robot. The integration is performed with the support of the Robotic

Operating System. In-gel validation shown the feasibility of the method, with a 100% success rate on reaching the target and 3.88 mm of error in positioning (with a tolerance of 5 mm), demonstrating performances comparable with the state-of-art of other controllers applied to the same needle steering system. The Programmable Bevel-tip Needle (PBN) is a multi-segment steerable needle under development within the EU EDEN2020 project. It is composed of four axially interlocked slender sections, robotically actuated to develop specific tip configurations that allow the needle to steer in the space. In this PhD dissertation, the PBN is considered as a case study for the presented methods.

Sommario

Lo sviluppo della procedura di neurochirurgia keyhole è dovuto alla sua capacità di ridurre al minimo il trauma del paziente, il rischio di infezione e il tempo di recupero. Queste caratteristiche rendono la neurochirurgia keyhole applicabile a una vasta gamma di pazienti non adatti alla chirurgia aperta. Tuttavia, l'esecuzione di queste procedure, utilizzando i tradizionali aghi rigidi, può comportare l'impossibilità di raggiungere gli obiettivi senza scontrarsi con ostacoli anatomici (ad esempio, le ossa) o tessuti anatomici sensibili (ad esempio, organi vitali e vasi sanguigni).

Negli ultimi due decenni, diversi gruppi di ricerca hanno concentrato i loro sforzi sullo sviluppo di aghi in grado di orientarsi all'interno dei tessuti. Questi aghi possono eseguire traiettorie curvilinee, pianificate per massimizzare la distanza da strutture anatomiche sensibili da evitare, e raggiungere obiettivi altrimenti inaccessibili tramite percorsi di inserimento rettilinei. A differenza degli aghi convenzionali, per i quali il percorso di inserimento può essere pianificato ed eseguito dal medico sulla base della posizione del bersaglio e dell'anatomia del paziente, la complessa cinematica degli aghi orientabili rende infattibile la pianificazione manuale, richiedendo quindi l'aiuto di soluzioni automatiche o semiautomatiche. La precisione pre-operatoria e intra-operatoria e la robustezza del controllo del movimento robotico contribuiscono direttamente ai risultati del trattamento e al recupero post-operatorio.

Su queste basi, l'obiettivo generale di questa tesi di dottorato è quello di descrivere un quadro sicuro ed efficace di pianificazione automatica di neurochirurgia keyhole per aghi neurochirurgici flessibili. Il sistema intende supportare i neurochirurghi durante: 1) la procedura pre-operatoria per determinare rapidamente il punto di ingresso chirurgico più appropriato e la traiettoria attraverso il cervello con un rischio ridotto al minimo, 2) la procedura intra-operatoria per interagire con un ambiente dinamico, e 3) l'inserimento dell'ago robotico per fornire una strategia di controllo adeguata. In particolare, i contributi di questo lavoro di dottorato sono:

1. *Una panoramica delle diverse tecniche di pianificazione del percorso utilizzate per gli aghi/cateteri chirurgici orientabili.* Una ricerca sistematica e attenta della letteratura nelle principali basi di dati come Pubmed, Scopus e Web of Science

è stata effettuata utilizzando "steerable catheter/needle" e "path planning" come parole chiave importanti. L'analisi effettuata per ogni articolo riguarda l'obiettivo clinico, il metodo di pianificazione del percorso, la procedura di validazione e i risultati quantitativi. Sono stati identificati 84 studi su 337 che hanno effettivamente fatto uso di algoritmi di pianificazione del percorso per aghi/cateteri orientabili. Tali paradigmi includono metodi basati su grafi, campionamento, matematica, apprendimento, ragionamento e multifusione.

2. *Un sistema di pianificazione del percorso pre-operatorio che incorpora tecniche basate sull'apprendimento e sul ragionamento al fine di migliorare le prestazioni di pianificazione del percorso.* In particolare, un modello di apprendimento induttivo, basato su dimostrazioni eseguite da operatori esperti, è incaricato di generare un insieme di percorsi come soluzioni candidate; successivamente, un modulo di ragionamento deduttivo seleziona il "miglior" punto di partenza, secondo la conoscenza esplicita modellata sui suggerimenti degli esperti del dominio. Questo tipo di accoppiamento permette di trasferire al pianificatore automatico di percorsi la maggior parte della conoscenza disponibile a livello umano. Il modulo di apprendimento induttivo "cattura", tramite dimostrazioni, le capacità degli esperti che sono difficili da esprimere esplicitamente (ad esempio, visuale-spaziale, fisico-cinestetico), mentre il modulo deduttivo codifica formalmente ciò che è stato elaborato dagli esperti dopo una lunga pratica (ad esempio, la conoscenza del dominio, le migliori pratiche). Infine, valutiamo la fattibilità dell'approccio proposto, in simulazione, dimostrando che è in grado di competere o addirittura superare le soluzioni presenti nello stato dell'arte in termini di distanza di sicurezza dagli ostacoli, linearità e tempo di calcolo.
3. *Un simulatore realistico, incentrato sulla simulazione dinamica basata sulla posizione, che simula le deformazioni del cervello dovute all'inserimento del catetere e massimizza la probabilità di successo della pianificazione del percorso intra-operatorio tenendo conto dell'incertezza nei modelli di deformazione.* Il simulatore utilizza un nuovo approccio rispetto alla letteratura, e ha dimostrato di avere una stretta corrispondenza con le deformazioni cerebrali reali attraverso la convalida utilizzando i dati di deformazione registrati in esperimenti su animali in vivo con un mismatch medio del $4.73 \pm 2.15\%$. La stabilità, la precisione e le prestazioni in tempo reale rendono questo modello adatto a creare un ambiente dinamico per la pianificazione del percorso KN e la guida intra-operatoria.
4. *Un nuovo ripianificatore di percorsi in tempo reale in grado di generare, ad ogni passo temporale durante l'inserimento, un percorso privo di ostacoli e di curvatura, tenendo conto di una posa vincolata dell'obiettivo e della deformazione anatomica intra-operatoria.* Il percorso generato è ottenuto con metodi di apprendimento di rinforzo inverso per la pianificazione della traiettoria intra-operatoria. I risultati di simulazione eseguiti su un set di dati realistici dimostrano che il nostro metodo di ripianificazione può guidare un ago con curvatura limitata a una posizione di destinazione predefinita con un errore medio di posizionamento di 1.34 ± 0.52 mm in posizione e 3.16 ± 1.06 gradi in orientamento in un ambiente deformabile, con un tempo di ripianificazione di 0.02 sec e un tasso di successo

del 100%. L'obiettivo è considerato raggiunto con una tolleranza di 2.5 mm per la posizione e 10° per l'orientamento.

5. *Un controllore di movimento che può guidare un ago orientabile per produrre la traiettoria desiderata in 3D.* Questo viene realizzato grazie all'integrazione del modello di pianificazione del percorso sviluppato in 3D Unity in un sistema che consiste in un ago orientabile, un tracciamento dell'ago tramite sensori elettromagnetici e un robot che guida l'ago. L'integrazione viene eseguita con il supporto del Robotic Operating System. La validazione in-gel ha mostrato la fattibilità del metodo con un tasso di successo del 100% nel raggiungere il bersaglio e 3.88 mm di errore nel posizionamento (con una tolleranza di 5 mm), dimostrando prestazioni paragonabili allo stato dell'arte di altri controllori applicati allo stesso sistema di guida dell'ago. Il Programmable Bevel-tip Needle (PBN) è un ago orientabile a più segmenti in fase di sviluppo nell'ambito del progetto EU EDEN2020. È composto da quattro sezioni sottili interconnesse assialmente, azionate roboticamente per sviluppare specifiche configurazioni della punta che permettono all'ago di sterzare nello spazio. In questa tesi di dottorato, il PBN è considerato come un caso di studio per i metodi presentati.

List of publications

International Journals

1. Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., ... and De Momi, E. (2021). Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. *IEEE Robotics and Automation Letters*.
2. Segato, A., Marzullo, A., Calimeri, F., and De Momi, E. (2020). Artificial intelligence for brain diseases: A systematic review. *APL bioengineering*, (4(4), 041503).
3. Segato, A., Corbetta, V., Di Marzo, M., Pozzi, L., and De Momi, E. (2020). Data augmentation of 3D brain environment using Deep Convolutional Refined Auto-Encoding Alpha GAN. *IEEE Transactions on Medical Robotics and Bionics*, (3(1), 269-272).
4. Segato, A., Pieri, V., Favaro, A., Riva, M., Falini, A., De Momi, E., and Castellano, A. (2019). Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei. *Frontiers in Robotics and AI*, (6, 70).
5. Favaro, A., Segato, A., Muretti, F., and De Momi, E. (2021). An Evolutionary-Optimized Surgical Path Planner for a Programmable Bevel-Tip Needle. *IEEE Transactions on Robotics*.

International Journals - Under Review

1. Segato, A., Calimeri, F., Testa I., Corbetta, V., Riva, M., and De Momi, E. A Hybrid inductive learning-based and deductive reasoning-based 3-D path planning method in complex environments. *Autonomous Robots*. (resubmitted after major revisions on November 2021).
2. Segato, A., Li, Z., Favaro, A., and De Momi, E. (2021) Steerable needles path planning for minimally invasive surgery: a systematic review. *IEEE Transactions on Robotics*. (submitted on May 2021)

-
3. Antonella, C., Valentina, P., Segato, A., Marco, R., Vidotto, M., Nicolò, P., ... and Falini, A. In vivo MRI measurement of microstructural constraints for Convection-Enhanced Delivery within the brain. *Scientific Reports* (submitted on June 2021)
 4. Segato, A., Di Marzo, M., Zucchelli, S., Galvan, S., Secoli, R., and De Momi, E. Inverse Reinforcement Learning Intra-operative Path Planning for Steerable Needle. *IEEE Transactions on Biomedical Engineering (T-BME)* (resubmitted after major revisions on November 2021)
 5. Segato, A., Di Marzo, M., Galvan, S., De Momi, E., and Secoli, R. Toward Autonomous Control of Needle Steering Systems: An Inverse Reinforcement Learning Approach. *IEEE Transactions on Robotics (T-RO)* (submitted on August 2021)

International Conferences/Workshops with Peer Review

1. Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., Galvan, S., Secoli, R., and De Momi, E. (2021). Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. In *2021-International Conference on Intelligent Robots and Systems. Prague*
2. Segato, A., Corbetta, V., Zangari, J., Z., Perri, S., Calimeri, F., and De Momi, E. (2021). Optimized 3D path planner for steerable catheters with deductive reasoning. In *2021 IEEE International Conference on Robotics and Automation, ICRA 2021 China*, (pp. 1-7).
3. Segato, A., Sestini, L., Castellano, A., and De Momi, E. (2020, May). GA3C reinforcement learning for surgical steerable catheter path planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2429-2435). IEEE.
4. Segato, A., Irene, T., and De Momi, E. (2020). 3D Neurosurgical Simulator for Training Robotic Steerable Catheter Agents Using Generative Adversarial Imitation Learning. In *2020 International Conference on Robotics and Automation (ICRA)*. France.
5. Segato, A., Corbetta, V., Francesco, C., and De Momi, E. (2020). Inductive and Deductive Reasoning for Robotic Steerable Needle in Neurosurgery. In *2020-International Conference on Intelligent Robots and Systems*. USA.
6. Antonella, C., Segato, A., Valentina, P., Vidotto, M., Nicolò, P., Marco, R., ... and Falini, A. (2020). In vivo MRI measurement of microstructural constraints for direct drug delivery within the brain. In *2020-ISMIRM and SMRT Virtual Conference and Exhibition*.
7. Li, Z., Segato, A., Favaro, A., Jenny, D., and De Momi, E. (2020). A Heuristic-Sliding-Window-based RRT Path Planning for Endovascular Catheterization.
8. Favaro, A., Valentina, P., Segato, A., Andrea, F., De Momi, E., and Antonella, C. (2018). Steerable needle DBS path planning safeguards deep nuclei and white

matter tracts. In Joint Workshop on New Technologies for Computer/Robot Assisted Surgery (CRAS). *GBR*.

Others

1. Segato, A., Corbetta, V., Zangari, J., Z., Perri, S., Calimeri, F., and De Momi, E. Towards solving path planning in keyhole neurosurgery with Answer Set Programming. The 37th International Conference on Logic Programming.
2. Segato, A., Di Marzo, M., Zucchelli, S., Galvan, S., Secoli, R., and De Momi, E. Intra Operative Planning Framework for Surgical Flexible Needle with Generative Adversarial Imitation Learning. The 13th Hamlyn Symposium on Medical Robotics 2021.

List of Abbreviations

A

ACO	Ant Colony Optimization. 23, 35
AD	Alzheimer's Disease. 19
AFT	Adaptive Fractal Tree. 25, 29
AHFT	Adaptive Hermite Fractal Tree. 4, 22, 25, 29, 80
AI	Artificial Intelligence. 38, 49, 69, 77
APF	Artificial Potential Field. XXIII, 25, 29, 39, 40
ASP	Answer Set Prigramming. XXIV, 13, 38, 39, 53, 57, 58, 65, 116, 117

B

BA	Bresenham's algorithm. 33
BB	Bubble Bending. 29, 80
BFS	Breadth First Search. XXIII, 24, 25, 39, 40
BIT-Star	Batch Informed Trees star. 3, 80
BIT-Star	Batch Informed Trees star. XXIII, 25, 29, 39, 40

C

CED	Convection Enhanced Delivery. XXIV, XXV, 2, 3, 59–61, 64, 65, 80
CNN	Convolutional Neural Network. XXIII, 35–37, 85
CoM	Center of Mass. 72, 73
CSF	Cerebrospinal Fluid. 68
CST	Corticospinal Tract. XXIV, 59, 60
CT	Computed Tomography. 9, 20, 71
CTA	Computer-Tomographic Angiography. 9

CTN	Concentric Tube Needle. 3, 79
D	
DBS	Deep Brain Stimulation. XXIV, XXV, 3, 19, 42, 59–61, 64, 65, 79
DD	Drug Delivery. 13, 19
dDQN	dueling Deep Q-learning. XXIII, 35–37
DP	Dynamic Programming. XXIII, 23, 31, 33, 39, 40
DQN	Deep Q-learning Network. XXIII, 23, 35–37, 80
DR	Deductive Reasoning. XXIV, XXV, 50, 53, 57, 61–65
DRL	Deep Reinforcement Learning. 49, 53, 80, 83, 86, 122
E	
EA	Evolutionary Algorithm. XXIII, 35, 36
EBB	Extended Bubble Bending. XXVIII, 80, 88–90, 92, 94, 105
EDEN2020	Enhanced Delivery Ecosystem for Neurosurgery in 2020. 105
EM	Electromagnetic. 20
F	
FBG	Fiber Bragg Grating. 105, 106
FBN	Flexible Bevel-tip Needle. 4, 80
FEM	Finite State Element. 20, 26, 33, 68, 76, 77, 80, 113
FFE	Fast-Field-Rcho. 70, 71
FFNN	Feedforward Neural Networks. XXIII, 35, 36, 39, 40, 42
G	
GA	Genetic Algorithm. XXIII, 13, 23, 35, 36, 39, 40
GAIL	Generative Adversarial Imitation Learning. XXIII, XXIV, XXVI, XXIX, 35–37, 53, 56, 57, 83–85, 94, 97–100, 102, 114, 123
GB	Graph-Based. 79
GBO	Gradient Based Optimization. XXII, 12, 31, 32, 80
GPU	Graphics Processing Unit. 29
I	
ICG	Improved Conjugate Gradient. 39

ihDP	infinite horizon Dynamic Programming. XXII, 32, 33, 39, 41
IK	Inverse Kinematics. XXII, XXIII, 31–33, 39, 40, 42
IL	Inductive Learning. XXIV, 49, 50, 53, 57, 59, 62
ILDR	Inductive Learning-based and Deductive Reasoning-based. XVIII, XXV, 52, 60–65
IRL	Inverse Reinforcement Learning. 6, 7, 80, 83, 86, 94, 97, 100, 102, 105, 112–114, 123
K	
KN	Keyhole Neurosurgery. 2–4, 6, 19, 45, 60, 67, 77–80, 88, 94, 96, 98, 101, 110–114
L	
LB	Learning-Based. 80
LfD	Learning from Demonstration. XXIII, 35–37, 80, 96, 97
LITT	Laser-induced Thermotherapy. 29
LPA*	Lifelong Planning A-star. 22, 25, 79
LSTN	left subthalamic nucleus. XXIV, 50, 59, 60
M	
MB	Mathematical-Based. 80
MCH	Composite Hydrogel. 68, 77
MDP	Markov Decision Process. 23, 33, 39, 41
MFN	magnetically guided flexible needle. 3, 80
MIS	Minimally Invasive Surgery. 1, 5, 6, 13, 18, 19, 25, 45, 95, 96, 113, 114
ML	Machine Learning. 4, 5, 60, 80, 96, 122, 123
MR	Magnetic Resonance. 59, 68, 70, 71, 74
MRA	Magnetic Resonance Angiography. 25
MRE	Magnetic Resonance Elastography. 73
MRI	Magnetic Resonance Imaging. 9, 20, 37
MS	Moduli Space. 25, 29
MSM	Mass-Spring Model. 80
N	
NN	Neural network. XXIII, 13, 36, 85, 123
NNG	Nearest Neighbor Graph. 24, 25
P	

PBD	Position based dynamics. XXV, XXXI, 6, 67–72, 76, 77, 80, 81, 86, 101, 113, 114, 118, 120–123
PBN	Programmable Bevel-tip steerable Needle. XXV, 2, 3, 5, 6, 29, 69, 71, 73, 79–81, 87, 89, 96–98, 101, 105, 106, 111, 114
PD	Parkinson’s Disease. 19
POP	Path-of-Probability. 34
PPO	Proximal Policy Optimization. 83, 85, 100, 122, 123
PRM	Probabilistic Road Map. XXII, XXIII, 12, 25, 26, 39–41
PSO	Particle Swarm Optimization. XXIII, 35, 36
R	
RFA	Radio-Frequency Ablations. 19
RG-RRT	Reachability Guided Rapidly-exploring Random Tree. 25, 27, 28
RL	Reinforcement Learning. XXIII, XXIV, 4, 36, 37, 45, 53, 56, 80, 85, 96, 97, 100, 113, 122, 123
ROS	Robotic Operating System. 87, 102
RRT	Rapidly-exploring Random Tree. XXII, XXIII, 12, 22, 25–29, 39–41, 61, 62
RRT*	Rapidly-exploring Random Tree-star. 79
RRT-Con	Rapidly-exploring Random Tree-Connect. 4, 80
RRT-star	Rapidly-exploring Random Tree Star. 28
RSI	Radioactive source implantation. 19
RT	Re-planning Time. 81
S	
SAC	Soft Actor Critic. 83, 85, 122
SB	Sampling-Based. 79
SMR	Stochastic Motion Roadmap. XXIII, 25, 30, 39–41
SOTA	State-Of-The-Arts. 45, 77, 80, 110
SPF	Shortest Path First. 24
SR	Success Rate. 81
T	
TA	Tumor Ablation. 4, 80
TOE	Target Orientation Error. 81
ToF	Time-of-Flight. XXVI, 59, 70, 72
TPE	Target Position Error. 81

TRPO	Trust Region Policy. 56, 57
U	
UDQL	Universal Distributional Q-learning. 37
US	Temporal Bone Surgery. 19
US	Ultra Sound. 20, 28
USi	Ultrasound Image. 9, 20

Contents

List of Abbreviations	XII
1 Introduction	1
1.1 PROBLEM STATEMENT	1
1.1.1 Minimal invasive surgery	1
1.1.2 Steerable needle	2
1.1.3 Path planning	3
1.1.4 Motion control	4
1.2 AIM OF THE THESIS	5
2 Steerable needles path planning for minimally invasive surgery: a systematic review	8
2.1 INTRODUCTION	9
2.1.1 Type of needle steering	9
2.2 PROBLEM STATEMENT AND MOTIVATION	11
2.2.1 Path planning problem	11
2.2.2 The path planning algorithm taxonomy	12
2.2.3 Path planner in minimally invasive surgery	13
2.3 METHODS	14
2.4 EVALUATION METRICS	18
2.5 RESULTS	18
2.5.1 Graph-based methods	24
2.5.2 Sampling-based methods	25
2.5.3 Mathematical-based methods	31
2.5.4 Learning-based methods	35
2.5.5 Reasoning-based methods	38
2.5.6 Multifusion-based methods	39
2.6 DISCUSSION	42
2.7 CONCLUSION	44

3	A Hybrid inductive learning-based and deductive reasoning-based 3-D pre-operative path planning method in complex environments	46
3.1	INTRODUCTION	46
3.2	RELATED WORK	48
3.3	MATERIALS AND METHODS	50
3.3.1	Problem statement	50
3.3.2	ILDR architecture	52
3.4	EXPERIMENTAL PROTOCOL	59
3.4.1	Neurosurgical environment	59
3.4.2	Neurosurgical simulator	60
3.4.3	Experimental validation	61
3.4.4	Hardware specification	62
3.4.5	IL Training Strategy	62
3.4.6	Statistical analysis	62
3.5	RESULTS	63
3.5.1	Convection enhanced delivery	63
3.5.2	Deep brain stimulation	63
3.5.3	Computational time	63
3.6	DISCUSSION AND CONCLUSION	65
3.6.1	Clinical translation	66
3.6.2	Future directions	66
4	Position-based Dynamics Simulator of Brain Deformations for Intra-Operative Path Planning and Control in Keyhole Neurosurgery	67
4.1	INTRODUCTION	67
4.2	RELATED WORK	68
4.3	OBJECTIVE	69
4.4	MATERIALS and METHODS	69
4.4.1	Position-based dynamics approach	69
4.4.2	Case study: keyhole neurosurgery	69
4.4.3	Computational setup	74
4.5	RESULTS	74
4.5.1	White matter calibration	74
4.5.2	Deep gray matter calibration	76
4.5.3	Brain deformation model validation	76
4.6	DISCUSSION	76
4.7	CONCLUSION	77
5	Deep and Inverse Reinforcement Learning Intra-operative Path Planning for Steerable Needle	78
5.1	INTRODUCTION	78
5.2	RELATED WORK	79
5.3	OBJECTIVE	80
5.4	MATERIALS AND METHODS	81
5.4.1	Moving Agent - Flexible needle	81
5.4.2	Dynamic Environment - Needle-Tissue and Tissue-Tissue interactions	81

5.4.3	Problem: intra-operative path planning	82
5.4.4	Actions and Observations	82
5.4.5	Reward Function	83
5.4.6	DRL and IRL Model Construction	83
5.4.7	Simulation Framework	86
5.4.8	Integration into Robot Operating System	87
5.4.9	Hardware Specification	88
5.5	EXPERIMENTAL SETUP	88
5.5.1	Experimental protocol	88
5.5.2	Validation	90
5.6	RESULTS	92
5.7	DISCUSSION	93
5.8	CONCLUSION	94
6	Toward Autonomous Control of Needle Steering Systems: An Inverse Reinforcement Learning Approach	95
6.1	INTRODUCTION	95
6.2	RELATED WORK	96
6.3	OBJECTIVE	97
6.4	MATERIALS AND METHODS	98
6.4.1	Control Problem	98
6.4.2	IRL as Control	98
6.4.3	Training Environment	101
6.4.4	Training strategy	102
6.4.5	Integration with the physical robotic system	102
6.4.6	Simulated Environment	104
6.4.7	Real Environment	105
6.5	EXPERIMENTAL SETUP	105
6.5.1	In-silico experiments: training on experimental dataset	106
6.5.2	In-Vitro Experiment	106
6.6	EXPERIMENTAL ANALYSES	106
6.7	RESULTS	107
6.7.1	In-Silico Study	107
6.7.2	In-Vitro Experiment	109
6.8	DISCUSSION	109
6.9	CONCLUSION	110
7	Conclusion	111
7.1	Thesis contribution	112
7.2	Impact and future perspective	113
8	Appendices	115
8.1	Path Planning Systematic Review Keywords	115
8.2	Computing the radius of curvatures	115
8.3	Answer Set Programming	116
8.4	Position Based Dynamic	118
8.4.1	Position-based Dynamics Approach	118

8.5	Deep Reinforcement Learning	122
8.5.1	SAC	122
8.5.2	PPO	123
8.6	Inverse Reinforcement Learning	123
8.6.1	GAIL	123
8.7	Electronic Supplementary Material	123
8.7.1	Neurosurgical Simulator	123
8.7.2	Brain Deformable Tissue PBD Simulator	123
	Bibliography	124

List of Figures

1.1	Synthetic Graphical abstract of the PhD thesis. The research hypotheses that guide this PhD work are presented	5
1.2	Extended Graphical abstract of the PhD thesis. The research hypotheses that guide this PhD work are presented, along with the performed activities and the solutions employed to validate the proposed methods.	7
2.1	Steerable Needles. Example of (A) Concentric tube needle consisting of three tubes [66]. (B) Bevel tip needle [277]. (C) Pre curved stylet [194]. (D) Programmable bevel consisting of four interlocked segments [145]. (E) Example of a tendon actuated active tip-steered needle with a ball joint and a conical tip [262]. (F) Magnetically controlled needle [257]. (G) Guidewire [56]. (H) Duty-cycle bevel tip [180]. (I) Micro/Nano robots [288]. (L) Hydraulically driven catheter [125]. (M) Fracture-direct waterjet [23]. (N) Ablation needle [41]. The cross-section of needles is shown on the upper right. (part of the image is courtesy of the [263]).	10
2.2	PRISMA Flow Diagram of Systematic Review Identification, Screening, Eligibility and Inclusion. 84 studies were included in the final analysis out of the 324 screened.	15
2.3	Papers Analysis Method. Schematization of the analysis carried out for each paper	17
2.4	Benchmark Measure. Path planning performance measures for steerable needles/catheter	18
2.5	Clinical aim results Distribution. Distribution of the specific anatomical area, pathology and surgical procedure based on all the analyzed published article	19
2.6	Path Planner Framework Results Distribution. Distribution of specific surgical procedure, risk analysis, type of query, type of experiments, type of environment, the method used for tissue modelling and tool based on all the analyzed published article	20

2.7	Validation Results Distribution. Distribution of specific steerable catheter/needle, sensors and type of data used based on all the analyzed published article	21
2.8	3D Path Planning Methods for Robotic Steerable Needle. Divided in Graph-based, Sampling-based, Mathematical-based, Learning-based, Reasoning-based and Multifusion-based, and their overall and yearly distribution from 2005 to 2021	21
2.9	Graph-based methods. Specific algorithms of graph-based methods, applied in the context of path planning for steerable needles, are illustrated	24
2.10	Sampling-based methods. Specific algorithms of sampling-based methods, applied in the context of path planning for steerable needles, are illustrated. These algorithms are divided into two more detailed parts: active and passive. Active means algorithms such as RRT that can obtain a feasible path to the target with its own processing procedure, choosing at each step a point in the space. Passive means algorithms such as PRM that first generate a road map from the beginning to the goal, then a search algorithm is used where the solution is chosen from all possible ones on the map	26
2.11	Mathematical-based methods. Specific algorithms of mathematical-based methods, applied in the context of path planning for steerable needles, are illustrated. These algorithms are divided into three more detailed parts: linear, nonlinear and dynamic programming. Linear programming refers to an algorithm, such as IK, that can achieve the best result in a mathematical concept whose requirements are expressed by linear relationships. Nonlinear problems arise when the objective or constraints cannot be expressed as linear functions without sacrificing some essential nonlinear feature of the real problem. This category was further divided into the constrained and unconstrained problem. In an unconstrained problem, such as GBO, minima are only found at stationary points. In a constrained problem, such as Spline, minima can be found at nonstationary points. Dynamic Programming means algorithm, such as ihDP, which subdivide a problem into several connected subproblems. A subproblem can, in turn, be subdivided into smaller, similar subproblems and so on until the subproblem is of unitary size.	32

2.12 **Learning-based methods.** Specific algorithms of learning-based methods, applied in the context of path planning for steerable needles, are illustrated. These algorithms are divided into three more detailed parts: EA, NN and RL. EA include GA, PSO and Ant Colony Optimization (ACO), EA was proposed to overcome the problem where traditional linear programming and dynamic programming frequently fail to solve NP-hard problems with a large number of variables. NN, such as CNN and FFNN was first introduced for obstacle avoidance as well as navigation and then became widely used and implemented in path planning in various sectors. NN approach aims to generate a dynamic landscape in a neural-like form. RL algorithms, such as LfD and in particular Deep RL algorithms, such as GAIL DQN, dDQN and UDQN received more attention, and there are increasing applications of it in robot path planning 36

2.13 **Reasoning-based methods.** Reasoning-based methods. Specific reasoning based methods, applied in the context of path planning for steerable needles, are illustrated. They are part of the “logic programming” category, in particular Answer Set Programming (ASP) 38

2.14 **Multifusion-based methods.** Specific algorithms of multifusion-based methods, applied in the context of path planning for steerable needles, are illustrated. The combined approaches involve graph-based methods, such as BFS algorithm, sampling-based methods, such as APF, PRM, SMR, RRT, RRT-connect RRT-star and BIT-Star algorithms, mathematical-based method, such as DP, IK and Spline algorithms, and learning-based methods, such as FFNN and GA algorithms 40

3.1 **Agent and 3D Complex Environment.** a)(on the left) The moving agent kinematic constraints are the needle diameter (d_{out}) and the maximum curvature (k_{max}) that it can perform. At $t - th$ time step it can translate (\mathbf{p}_{agent_t}) and rotate (\mathbf{R}_{agent_t}), performing an action a_t , from its configuration, \mathbf{T}_{agent_t} . b)(on the right) The 3D complex environment is represented by a brain structure, the obstacle space (C_{obst}), the free space (C_{free}), the agent configuration (\mathbf{T}_{agent}). 51

- 3.2 **Path Planning Problem and Method.** a) Schematic representation of the 3D path planning problem elements, start configuration ($\mathbf{T}_{start_k} = \mathbf{T}_{agent_0}$) and target configuration ($\mathbf{T}_{target} = \mathbf{T}_{agent_{n-1}}$). The task is to find the optimal path ($Q = \{\mathbf{T}_{agent_0}, \mathbf{T}_{agent_1}, \dots, \mathbf{T}_{agent_{n-1}}\}$). At every t -th time step \mathbf{T}_{agent_t} and $\mathbf{T}_{agent_{t+1}}$ are connected by straight segments, as an admissible sequence of "agent configurations", taking into account the "obstacle space", C_{obst} . b) The proposed path planning algorithm exploits two merged approaches: inductive learning-based and deductive reasoning-based approaches. The expert indicates N start configurations, (\mathbf{T}_{start_k} with $k \in N$), the target configuration, (\mathbf{T}_{target}) the rules and performs a set of demonstrations. While the agent acts and observes in the virtual environment. The expert's demonstrations and the agent's observations feed the inductive approach that generates a set of optimal trajectories $\{Q_k^{IL}\}$ with k equal to the N start configuration. The rules and the kinematic constraints feed the deductive approach that extracts the best path Q^{ILDR} 52
- 3.3 **ILDR Architecture.** The expert gives in input the dataset (C_{free} and C_{obst}), the start and target configuration (\mathbf{T}_{start} and \mathbf{T}_{target}) and the kinematics constraints (d_{out} and k_{max}). The IL model is trained through a loop that starts with paths generated by an expert ($\{Q^{Manual}\}$) and paths ($\{Q^G\}$) performed by the network generator (G) based on a RL approach. With a GAIL approach a discriminator (D) with its network (D_w), taking in input the expert and generator policies (π_E and π_θ) the starting value of the policy's parameters and of the discriminator (θ_0 and w_0) makes a comparison of these two paths generating an intrinsic reward ($r^{in}(\mathbf{T}_{agent}, a)$) based on similarity score (r_{sim}), updating the agent's policy π . The loop continues until the generator, moving in the environment (Env) with actions (a_i), collecting observations (d_{tot} , d_{min} , d_{avg} , c_{max} and α_{tr}) and computing an extrinsic reward ($r^{ex}(\mathbf{T}_{agent}, a)$), can produce a path similar to the expert's demonstrations and that respects the kinematic constraints. Once the IL model is trained, it generates a set of optimal paths; the weights ($w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$) and the kinematic constraints are taken as input by the DR classifier that extracts the best path with an approach based on ASP. Applying hard and soft constraints obtaining the best path Q^{ILDR} 53
- 3.4 **CED and DBS Environemnts.** In (a) sagittal, axial and coronal view of the CED environment are reported. Defining accordingly the obstacle space ($C_{obst} = \{\text{ventricles, thalamus, pallidum and vessels}\}$), the free space ($C_{free} = \text{gyri}$) and the target space ($C_{target} = \text{tumor}$). In (b) sagittal, axial and coronal view of the DBS environment are reported. Defining accordingly the obstacle space ($C_{obst} = \{\text{ventricles, thalamus, pallidum, vessels and CST}\}$), the free space ($C_{free} = \text{gyri}$) and the target space ($C_{target} = \text{LSTN}$). 59

3.5 **Neurosurgical simulator.** It is possible to: (1) start a simulation, (2) select the procedure between the DBS and CED, (3) input the physical characteristics of the needle, d_{out} and k_{max} , and the parameters to be maximised or minimised in the selection of the best path, (4) select starting and target configurations, \mathbf{T}_{start_k} and \mathbf{T}_{target} , (5) perform manually the trajectories with a joystick controller or let the agent to perform trajectories autonomously with ILDR and finally (6) visualise all the generated trajectories. 60

3.6 **Schematic representation of the experimental protocol workflow** Each environment Env_s (with $1 \leq s \leq 2$), takes in input the start configurations, $\mathbf{T}_{start_k}^s$ (with $1 \leq k \leq 10$), the target configuration, \mathbf{T}_{target}^s , the weights, $w_{c_{max}}^s, w_{d_{min}}^s, w_{d_{tot}}^s$ and the kinematic constraints of the moving agent, d_{out} and k_{max} . j experiments, EXP_j (with $1 \leq j \leq 5$), were conducted for each approach: Manual, DR, RRT* (only for DBS Environment) and ILDR. A pool of surgical paths, $\{path_k\}$ is generated, and the optimal one, $path_j^s$, is selected. 61

3.7 **Comparison between the different approaches.** For both CED (a) and DBS (b) environment, expert’s rules, and needle kinematic constraints are reported in the upper part. a) shows a comparison between Manual, DR, and ILDR approaches in the CED environment. b) shows a comparison between Manual, DR, RRT*, and ILDR in the DBS environment. The results for both considered scenarios and used approaches are reported in terms of the minimum (d_{min}) and the mean (d_{avg}) distance from the critical obstacles, the total path length (d_{tot}) and the curvature (c_{max}) calculated over the five best paths for each approach. P-values were calculated using Wilcoxon matched-pairs signed-rank test. 64

3.8 **Comparison between DR and ILDR approaches.** For both CED (scenario 1) and DBS (scenario 2) environment, one example of the obtained path is shown. In particular, a) shows a comparison between DR and ILDR approaches in the CED environment. b) shows a comparison between DR and ILDR in the DBS environment. The results for both considered scenarios show an increase in safety ($\gg d_{min}$) and smoothness ($\ll c_{max}$) and a reduction in length ($\ll d_{tot}$) for the proposed method (ILDR). 65

4.1 **Workflow of the experimental study.** 70

4.2 **Experimental Setup.** (A) Illustration of the PBN catheter showing the four segments interlocked by means of Dovetail joints. The four segments can slide on each other along the center axis, creating an offset that allows the steering of the catheter (courtesy of Leibinger, 2016). (B) Experimental setup with the hydrogel phantom and the needle presented in [161]. (C) Basic deformable parallelepiped-shaped phantom used for the initial setting of PBD simulation parameters for the brain white matter and the 3D model of the reconstructed catheter with an outer diameter of 2.5 mm. 71

4.3	Human Brain Dataset: (A) Representation of the brain structures segmented on the normalized 3D T1-weighted images and the 3D high-resolution ToF-MRA. (B) Particle model of brain white matter. (C) Particle model of deep gray matter structures.	72
4.4	Ovine Brain Dataset (A) Coronal, sagittal and axial view of post-operative CT on ovine brain and in-vivo experiment. (B) Ovine brain pre-operative and intra-operative 3DT1 and deformation field analysis. (C) 3D segmented dataset for the ovine brain. (D) Particle model for the ovine brain.	73
4.5	Needle Deformation Field of in-vivo Insertion (A) To sample the deformation field, first we set the viewing axes so that the z-axis is roughly parallel to the catheter insertion direction. (B) Then we take our samples from the catheter hole boundary in the x-z plane (green points) (C) and y-z plane (orange point).	74
4.6	Needle Displacement Results. (A) Displacement obtained for white matter on the calibration parallelepiped and comparison with the one obtained in [161]. (B) Heatmap representing the displacement of the different deep gray matter structures at various insertion depths (the maximum insertion depth reached is considered enough to be clinically relevant). (C) Five experiments reporting the deformation of the simulated ovine model (yellow) and the relative in-vivo reference (blue) as a function of the insertion depth. The obtained <i>errors</i> are considered acceptable as they are under a threshold fixed to 10%.	75
5.1	Agent and Dynamic Environment A The moving agent kinematic constraints are: the catheter diameter (d_{out}) and the maximum curvature (k_{max}) that it can perform. At $t - th$ time step it can translate (\mathbf{p}_{agent_t}) and rotate (\mathbf{R}_{agent_t}), from its configuration, \mathbf{q}_{agent_t} . B The environment is represented by a brain structure, the obstacle space (C_{obst}), the free space (C_{free}).	82
5.2	Models Construction A The SAC model is trained estimating the policy (π) and value function in every epoch, using the actual policy to interact with the environment, performing action a_t taking observation o_t , calculating the TD_{error} and returning the action probability density function. B PPO control policy, modelled as a NN (actor) and a second NN for estimating value functions (critic). C The PPO+GAIL+camera model is trained through a loop that starts with paths generated by an expert ($\{Q^{Manual}\}$) and paths ($\{Q^G\}$) performed by the network generator. With a GAIL approach, a discriminator with its network, taking in input the expert and generator policies (π_E and π_θ), makes a comparison of these two paths generating an intrinsic reward (r^{in}), updating the agent's policy (π). The loop continues until the generator, moving in the environment with actions (a_t), collecting observations with a visual encoder based on a CNN and computing an extrinsic reward (r^{ex}), can produce a path similar to the expert's demonstrations and that respects the kinematic constraints. D The PPO+GAIL+raycast model is trained as the model described in C, with the only difference being that the observations' nature is not visual but binary (collision or no collision).	84

- 5.3 **The simulation framework architecture** is composed by the “Segmentation” module that takes in input the patient dataset (MRI and CT) and gives in output the withe matter, target (*target*) and obstacles (*obst*) segmented meshes. The segmented meshes are the input of the “PDB simulator” of brain tissue deformation, that produces in output a (1) static environment composed by the target space and the obstacle space (C_{target}, C_{obst}) which are the input together with expert manual demonstration (Q^{manual}), start (\mathbf{q}_{start}) and target (\mathbf{q}_{target}) configuration and kinematics constraint(k_{max}) of the (2) “Training” the (3) "Pre-operative path planning” modules. The (2) and (3) returns respectively the trained model and the pre-operative path (Q^{pre}) both input of the (4) “Intra-operative path planning” module. At this point the (4) can start to interact with a Dynamic Environment (5) sending the agent current configuration (\mathbf{q}_{agent_t}) and generating (6) the agent next step ($\mathbf{q}_{agent_{t+1}}$). This configuration is sent to the “Motion Control” module, which estimates the curvature the needle as to achieve(K_{des_t}) translated into the motion of the PBN. The configuration of the robot is registered by the sensors in the form of curvature(k_{real_t}) and translated as the real configuration of the tip ($\mathbf{q}_{real_{t+1}}$). The estimation of the real tip configuration allows computing the (7) error with respect to the agent configuration previously computed ($\mathbf{q}_{agent_{t+1}}$). This error affects the (8) fast re-planning that, in combination with the trained model, returns (4) the new configuration the needle has to achieve performing re-positioning, taking account of the committed error and trying to predict the future bias. The procedure continues in a loop until the target is reached or an obstacle is touched. 85
- 5.4 **The intra-operative path**, Q^{intra} , starts in \mathbf{q}_{agent_0} which coincides with the real robot initial configuration (\mathbf{q}_{real_0}), the pre-operative path (Q^{pre}) initial configuration (\mathbf{q}_{real_0}) which in turn coincides with the selected starting configuration (\mathbf{q}_{start_0}). The intra-operative path planning algorithm generates a new configuration (\mathbf{q}_{agent_t}) at every t step, most of the time different from (\mathbf{q}_{ref_t}) due to the real robot configuration (\mathbf{q}_{real_t}), where the agent perform re-positioning updating its current configuration with the real robot configuration $\mathbf{q}_{agent_t} \leftarrow \mathbf{q}_{ref_t}$. The last configuration ($\mathbf{q}_{agent_{n-1}}$) reaches the target configuration and space (\mathbf{q}_{target_t} and Q_{target_t}), avoiding obstacles (Q_{obst_t}), all deformed with respect to the initial configurations due to the simulated brain deformation. 88

- 5.5 **Experimental protocol.** The training phase, for SAC and PPO+GAIL models, takes in input m start (\mathbf{q}_{start_m}) and target (\mathbf{q}_{target_m}) configurations and expert manual path (Q_m^{manual}) (with $1 \leq m \leq 70$). For all the proposed approaches SAC, PPO+GAIL+camera, PPO+GAIL+raycast and EBB the preoperative phase takes in input k \mathbf{q}_{start_k} and \mathbf{q}_{target_k} , (with $1 \leq k \leq 5$) and the kinematic constraints of the moving agent, d_{out} and k_{max} and produces k pre-operative path (Q_k^{pre}). The produced paths Q_k^{pre} , together with the maximum number of steps allowed (t_{max}) and the update frequency (t_{update}) are the inputs of j experiments, EXP^j (with $1 \leq j \leq 2$), where EXP^1 correspond to experiments without repositioning of the robot during the procedure while EXP^2 corresponds to experiments simulating the repositioning of the robot. Finally the output of each approaches are the intra-operative paths: $Q_k^{SAC,j}$, $Q_k^{GAIL+C,j}$, $Q_k^{GAIL+R,j}$ and $Q_k^{EBB,j}$ 90
- 5.6 **Results A** Pre-operative path, Q_k^{pre} (red), and Intra-operative results, Q_k^{intra} plotted as a function of the path, k , for qualitative comparison (intra-operative tests were conducted on a total of 5 path, with $1 \leq k \leq 5$). The intra-operative path is reported for each approach: $Q_k^{GAIL+PPO+ray}$ (blue), $Q_k^{GAIL+PPO+cam}$ (green), Q_k^{EBB} (black). **B** Shows TOE, TPE, d_{min} , d_{avg} and d_{tot} . **C** Shows a comparison between GAIL+PPO+ray, GAIL+PPO+cam and EBB approach in KN environment. The results for both considered scenarios and used approaches are reported in terms of the minimum distance (d_{min}) and the average distance (d_{avg}) distance from the critical obstacles, the total path length (d_{tot}) calculated over the k paths for each approach. P-values ($* \leq 0.05$ and $n.s. \geq 0.05$) were calculated using Wilcoxon matched-pairs signed-rank test. 93
- 6.1 **Block diagrams showing the different control systems for PBNs applied in the Eden2020 context.** (A) Representation of the ‘‘Surgeon-in-the-Loop’’ control system presented in [275]. The surgeon commands the desired needle curvature $k_{des}(t)$ from which the inverse model calculates the best needle shape. The motor control system then computes the required actuation of the PBN to achieve the desired needle shape at an insertion speed of $v(t)$. The PBN’s resultant tip pose $\mathbf{q}(t)$ is displayed to the surgeon alongside a reference pre-planned tip pose $\mathbf{p}_{ref}(t)$ via a visual interface. (B) Representation of the ‘‘Re-planning as Control’’ control system presented in [209]. ‘‘Re-planning as Control’’ framework removes the need of a user guiding the needle insertion, thus avoiding external influence on the resulting accuracy. The system automatically steers the needle along paths that avoid obstacles of known location. (C) Representation of our ‘‘Inverse Reinforcement Learning (IRL) as control’’ system. IRL algorithm replaces the surgeon by computing the required curvatures once the information about the PBN tip position is received. (A) Courtesy of Watts et al. [275] and (B) Courtesy of Pinzi et al. [209] 97

- 6.2 **Training Environment.** The IRL control policy, π , is trained through a loop that starts with the policy (π_E), generated by an expert and the policy (π_G) performed by the network generator. With a GAIL approach a discriminator with its network, taking in input the expert and generator policies (π_E and π_G), makes a comparison of these two paths generating an intrinsic reward (r^{in}), updating the learner policy (π). The loop continues until the generator, moving in the environment with actions (a_t), collecting observations and computing rewards (r_t), can produce a policy similar to the expert’s one. The training environment is represented by deformable mesh object of the brain (obtained with a PBD approach) and the model of the PBN. Here the body frame coordinates of the PBN is shown. Angular and linear velocities, $\omega_x, \omega_y, \omega_z$, and v are shown along with the corresponding curvatures k_2, k_1 , and torsion τ (assumed to be zero for the EDEN2020 PBN needle). 99
- 6.3 **ROS Integration. (A)** In order to establish the communication between *Unity* and *ROS*, we use ROS Sharp (*ROS#*). The communication is established through the activation of nodes on which both *Unity* and *ROS* can publish messages or read messages already published to make this communication the *package Rosbridge* is required. *Rosbridge* provides a *JSON API* to *ROS* functionality for non-*ROS* programs. The front-ends that interface with *Rosbridge* includes a *WebSocket* server for web browsers interaction. **(B)** The communication protocol between the “IRL as Control” based on *Unity Engine* and the robotic environment based on *ROS*, with the messages exchanged with the aid of *ROS#*. *Unity* is a publisher of the node */desired_curvature* and a subscriber of the node */needle_pose*. On the contrary, *ROS* is a subscriber of the node */desired_curvature* and a publisher of the node */needle_pose*. In this way, *Unity* can send information about the curvature the needle has to follow to the robot, and the robot can return to *Unity* the needle position. **(C)** Thanks to *ROS*, it was possible to connect the “IRL as Control” with the front end of the EDEN2020 system. The round boxes indicate a topic, while the squared ones are nodes. *ROS*’s meshes are provided to *Unity* thanks to the message the */front_end_proxy/segmentation_mesh* of type the */MeshArray*. Soft tissue modelling is applied to the different objects to generate the dynamic environment and take account of soft tissue deformation. *ROS* also passes user inputs such as start pose, (\mathbf{q}_{start}) and target pose (\mathbf{q}_{target}). The desired curvature ($k_{des}(t)$) is sent to the *motion_control* topic, which sends the information received to the desired curvature-offset conversion “inverse module”. These offsets are sent directly to the robot, moving the catheter by the desired displacement. FBG sensors return the exact pose ($\mathbf{q}(t)$) of the catheter tip once the movement is performed. The FBG’s feedback is provided to the *Unity* module thanks to the *tool_tracking* topic. 103

6.4 **Simulated Environment.** The “IRL as control” approach is integrated with the EDEN2020 simulator of the robotic system. The “IRL as control” outputs the desired curvature values. These curvature values are passed to the inverse model of the needle, which returns the required offsets between the needle segments to achieve that curvature. These required offsets serve as the inputs to the low-level controller, which generates the motion profiles (containing cyclic motions) for the simulated motors to move the segments and create the commanded offset. The expected curvature of the needle, based on the forward model of the needle converting segment offset to curvature, as well as the actual pose of the needle can then be shown to the operator via the visual interface and sent to the “IRL as control” as input. Part of the image is courtesy of Matheson *et al.* [176] 104

6.5 **Real Environment.** The “IRL as control” approach integrated with the EDEN2020 robotic system. The “IRL as control” outputs the desired curvature values. These curvature values are passed to the inverse model of the needle, which returns the required offsets between the needle segments to achieve that curvature. These required offsets serve as the inputs to the low-level controller, which generates the motion profiles (containing cyclic motions) for the motors to move the segments and create the commanded offset. The curvature of the needle, as well as the actual pose of the needle, measured by the FBG sensors, can then be shown to the operator via the visual interface and sent to The “IRL as control” as input. 105

6.6 **Experimental Setup.** (A) 7% by weight swine gelatin. (B) 2.5 mm outer diameter biocompatible Programmable Bevel-tip Needle. (C) Full experimental setup: the PBN is inserted into gelatin and tracked by the FBG sensors. The neuromate ©robot (Renishaw Plc), an actuation box controls each segment of the PBN and encoders measure their insertion length. (D) The PBN sensorized with four multi-core fibers that have FBG; each catheter segment has one fiber. The experiment utilizes four multi-core fibers, fan-out boxes, interrogator and coupler (Courtesy of Khan et al. [143]). (E) The Visual interface integrated into the commercial neurosurgical planning and intraoperative software neurosinspire™(Renishaw plc). (F) FBG sensors visual interface. 107

6.7 **In-silico results,** $Q_{k,j,g}^{silico}$ plotted as a function of the path, k , (in-silico tests were conducted on a total of 5 path, with $1 \leq k \leq 5$), as a function of the experiment type, j , ($j = 1$ without re-positioning, $j = 2$ with re-positioning) and as a function of maximum curvature, g , (with $g = 1$ when $k_{max} = 0.005$, $g = 2$ when $k_{max} = 0.007$, and $g = 3$ when $k_{max} = 0.014$). All the graphs show the pre-operative path, Q_k^{pre} , and the target pose, \mathbf{q}_{target} . Target position error TPE and Target orientation error TOE are reported for each result. 108

6.8 **Dataset and In-vitro Results** (A)Sagittal, axial and coronal view of the segmented brain objects are reported. In red the obstacles and in white the Gyri where the PBN can move freely. (B) In-silico insertion of the path with $k=1$. (C) In-vitro insertion of the path with $k=1$. (D) In-silico, $Q_{i,j,k}^{silico}$ and In-vitro, $Q_{i,j,k}^{vitro}$, results plotted as a function of the path, k , (in-vitro tests were conducted only for the first path, $k = 1$), as a function of the experiment type, j ($j = 1$ without re-positioning, $j = 2$ with re-positioning) and as a function of the maximum curvature, g (with $g = 2$ so $k_{max} = 0.007$). Both graphs show the preoperative path, Q_k^{pre} and the target pose \mathbf{q}_{target} . In the first graph (above), in-silico and in-vitro trajectories are compared following the experiment without re-positioning, while in the second case the conditions are the same except that the experiment was conducted with re-positioning. Target position error TPE and Target orientation error TOE are reported for each result. 109

8.1 **Graphic Representation of The General Idea of The PBD Algorithm,** (a), (b), (c) and (d) illustrate the four main phases of the algorithm (courtesy of Müller et al. [187]). 120

8.2 **Projection of the constraint** $C(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2| - d$, where the corrections $\Delta\mathbf{p}_i$ are weighted according to the inverse masses $w_i = 1/m_i$ (courtesy of Müller et al. [188]). 121

List of Tables

2.1	Needles category	10
2.2	Benchmark measures explained	18
2.3	Methods explained	21
2.4	Graph Based methods	25
2.5	Sampling Based methods	30
2.6	Mathematical Based methods	34
2.7	Learning Based methods	38
2.8	Reasoning Based methods	39
2.9	Multifusion Based methods	42
3.1	Related Work	48
3.2	Training parameters.	56
3.3	Expert constraints and rules.	62
3.4	Training parameters	63
3.5	Results in term of computational time are shown.	63
4.1	Adimensional parameters set for different brain structures as a result of the fine-tuning procedure	76
5.1	State of the art.	79
5.2	Reward Function parameters values	83
5.3	SAC training parameters	87
5.4	PPO+GAIL training parameters	87
5.5	Pre-operative results	92
5.6	Intra-operative results	92
5.7	Intra-operative obstacles avoidance	93
6.1	Demonstrations recording	102
6.2	Training parameters	102
6.3	In-silico results	108
6.4	In-vitro results	109

8.1 Keywords for Systematic Review 115

CHAPTER 1

Introduction

1.1 PROBLEM STATEMENT

1.1.1 Minimal invasive surgery

Nowadays, the incidence of cerebral diseases is exponentially increasing. According to a new United Nations report, one in six people has a mental illness, such as Alzheimer's, Parkinson's and brain tumours [1]. This growth is why the number of cutting-edge technology development and studies about brain surgery is growing. Among the most critical research challenges in this field is finding and targeting deep parts of the brain using brain surgery.

Brain surgery is a necessary, highly invasive procedure that allows accessing the brain to treat various conditions, including tumours, aneurysms, epilepsy and many other complicated diseases. It is possible to apply various techniques to proceed with the surgical procedure both for therapeutic and diagnoses reasons. Being very invasive procedures, the risks for patients are very high, yet brain surgery saves the lives of millions of people every year. In particular, in open brain surgeries, the cranial box's opening caused damage to the healthy tissues surrounding the target and an increase in the possibility of undesired side effects such as infections or other complications. These phenomena may cause higher healthcare costs, mainly related to longer hospitalisation times and expensive postoperative treatments [224].

The progress made by research in this field in recent years has made it possible to find new operating techniques that avoid opening the cranial box and thus minimise the risk of infection. Current technical advances allow performing, for the variety of the techniques, Minimally Invasive Surgery (MIS), which allows the reduction of the incision during the operation, thus increasing the accuracy and the quality of the procedure [208]. Minimally invasive surgery is used not only in the neurosurgical field,

but also in other fields and is considered the gold standard due to the improved imaging techniques and surgical instruments used to perform it. These factors increase precision and accuracy during the surgical procedure, drastically reducing complications and achieving better postoperative results than conventional craniotomy [292].

Keyhole neurosurgery

A procedure that is making its way into the world of neurosurgery is Keyhole Neurosurgery (KN). KN is a minimally invasive procedure performed to reach targets located deep inside the brain through a tiny hole in the skull, called a "burr hole" or "key-hole" [129].

1.1.2 Steerable needle

The use of standard instrumentation to perform the KN is his main drawback. Indeed, the limited typology of surgical trajectories that a rigid catheter can perform and the limited number of targets that can reach, due to the impossibility for a rigid probe to curve into the brain, make this approach excluded for all the kinds of operation that expects to arrive at anatomical sites in deep and complex areas that a straight path cannot reach, as Convection Enhanced Delivery (CED). CED is a technique for implanting a catheter into a brain tumour in order to infuse specific drugs. The risks involved in using a rigid catheter are the generation of haemorrhage, infection and hardware malfunction in combination with leakage of cerebral fluid, especially when the trajectory overtakes delicate parts such as blood vessels or ventricles.

These complications and constraints can be significantly minimised by using a steerable catheter. They are probe with the possibility to make curvilinear paths, thanks to flexible parts that compose them. The chance to curve provides more significant degrees of freedom for the respective trajectory, allowing reaching deep brain areas that are not reachable by straight trajectories. The technologies that develop steering needles can overcome the limits imposed by the straight, rigid probe. Indeed, these catheter types enhance targeting, promote the study of novel treatment methods, and reduce complication rates. Moreover, the generation of curvilinear trajectories through steerable probes allows an enhancement of the path's accuracy and an improvement of the safety of the procedures since the distance of the probe to healthy tissues can be increased [247].

The programmable bevel-tip needle

Currently, the EDEN2020 European project has developed an innovative Programmable Bevel-tip steerable Needle (PBN) for KN. The PBN is manufactured with four axially interlocked thin sections, which are robotically driven at the base of the needle so that they slide over each other along the main needle direction. The needle has a diameter of 2.5 mm. Each section has a hollow lumen with an outer diameter of 0.3 mm that can be used for drug delivery, cyst evacuation or optically based diagnostic detection. The tip of each segment is ground to form a blunt profile at an angle of approximately 30° from the neutral insertion axis of the needle. Further details on the fabrication of the PBN can be found in [274]. The needle tip changes according to the offset generated

between adjacent sections and determines the driving direction and degree of curvature. The PBN is considered a case study for the methods described in this thesis.

1.1.3 Path planning

As mentioned previously, the use of steerable needles in neurosurgery provides the advantage over the standard rigid probe allowing it to reach deep brain areas without damaging the surrounding tissues. However, differently from conventional needles, for which the insertion path can be planned and performed by the clinician on the basis of the target location and the patient anatomy, the complex kinematics of steerable needles make the pre- and intra- operative path planning unbearable requiring the aid of automatic or semi-automatic path planning solutions [57].

Pre-operative path planning

In general, pre-operative path planning algorithms calculate the optimal path from a selected entry point on the cerebral cortex to a destination point, e.g. the tumour, avoiding obstacles [229]. Indeed, the brain is characterised by a high density of vulnerable tissues, and any damage to these structures could compromise the surgical procedure's success. Therefore, pre-operative path planning algorithms applied to neurosurgery should define an optimal path within the grey and white matter while avoiding obstacles identified by critical brain structures such as blood vessels, thalamus, ventricles and dorsal pallidum. In order for the planner to avoid obstacles, it is necessary to specify in advance the criteria to be followed. For example, maximising the distance the catheter must maintain from obstacles performs a safer trajectory while minimising the trajectory's length will make the operation less invasive. It is also essential to assess the feasibility of the trajectories computed. In this case, algorithms should consider the maximum curvature that the probe can perform and its diameter.

Intra-operative path planning

Moreover, during the intra-operative path planning phase, a good planning algorithm should also consider the movements that obstacles make inside the brain during the operation. These deformations, caused by the insertion of the catheter, if not simulated or taken into account correctly, can cause errors in planning as they can move the obstacle on the trajectory that was thought to be safe and thus negatively affect the outcome of the operation.

A number of approaches for planning KN interventions have been proposed to assist the surgeon in the planning process. Such paradigms include graph-based, sampling-based and mathematical-based methods. In [191], Niyaz *et al.* exploit a graph-based method for efficient path re-planning of Concentric Tube Needle (CTN) validated in-silico but without a proper dynamic environment and brain tissue modelling. In [240]¹, a sampling-based path planner is proposed for a PBN for Deep Brain Stimulation (DBS). The same was done by Hong *et al.* [119] but for a magnetically guided flexible needle (MFN) [205]. In [79], Favaro *et al.* proposed a solution, for CED, based on a Batch

¹Segato, A., Pieri, V., Favaro, A., Riva, M., Falini, A., De Momi, E., and Castellano, A. (2019). Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei. *Frontiers in Robotics and AI*, (6, 70).

Informed Trees star (BIT-Star) approach. These last three methods only proposed a pre-operative solution without taking into account the dynamicity of an intra-operative environment. In [78], Fauser *et al.* proposed a solution using an Rapidly-exploring Random Tree-Connect (RRT-Con) [151]. The proposed algorithm was validated in-silico without a proper dynamic environment accounting for uncertainty in needle-tissue interaction and needle/tissue deformation. Pinzi *et al.* in [210] proposed an Adaptive Hermite Fractal Tree (AHFT) approach. This work was extended in [209] for fast re-planning, taking into account the environment changes such as tissue deformation. In [25] Bano *et al.* presented a mathematical-based method for generating pre-operative paths for neurosurgical Flexible Bevel-tip Needle (FBN). This method was later validated by Ko *et al.*, in [146]. Even in this work, during the in-silico validation, the framework didn't consider a dynamic environment. In [273], a heuristic-based search algorithm is introduced to find an optimal path of the needle for neurosurgical Tumor Ablation (TA).

To the best of our knowledge, there is currently no implementation of learning-based framework in the KN for pre or intra-operative path planning of steerable needles. The main reason is that Machine Learning (ML) methods, applicable in this context, such as on-policy Reinforcement Learning (RL), require free exploration in the environment which is not feasible in-vivo (with this statement we do not consider the off-policy ones for this type of application since especially in a dynamic environment the agent needs to explore or have pre-recorded data of in-vivo steerable needle insertion, which are difficult to get as they are not yet used in the clinic). However, by exploiting the idea that surgeons utilize pre-operative 3D images for pre-planning robotic platforms, a simulated environment can be implemented as a "Brain Digital Twin" to perform exploration and trajectory optimization training. The idea of Digital Twin [259] is used instead of a pure simulation, because enable real-time system reflections, interaction between physical and virtual spaces, and automatic model evolution with updated data feed. What we need, therefore, is a simulator that is as close to reality as possible and that can provide us with the dynamism to train reliable on-policy RL models.

1.1.4 Motion control

Precision is of paramount importance in steerable needle insertion that can not be done directly by the surgeon. Indeed, specialised devices are required to insert and guide these needles into the tissue. These are robotic devices that assist the human during insertion with precise and constant speed to prevent the catheter from flexing under the surgeon's excessive force, who may inadvertently bend them. A significant component required to bring these robotically actuated steerable needles into use is the development of appropriate control strategies to achieve an accurate insertion.

Two main approaches are frequently used to control needle steering systems: control strategies based on kinematic/dynamic model of the needle and methods that use path re-planning as control. In the first group, different techniques have been proposed such as a sliding mode control technique based on nonholonomic unicycle model of the needle [226], a nonlinear adaptive control scheme is proposed for stabilizing steerable needles [185] and model predictive control to explicitly consider input and state constraints that arise from the unique mechanism of motion of the probe [147]. In

the second group, different approaches were evaluated for different needle designs to address effects such as tissue deformation and target motion. One of the most recent works for PBN based designs is a modified version of the ‘‘Bubble Bending’’, proposed by Pinzi *et al.* in [209].

In addition to these two types of approaches based on kinematic/dynamic models and methods that use path re-planning to control the needle, recent advances in ML demonstrate the ability to automate the control of needle steering systems [264]. A learning-based approach can add further advantages such as increased stability of the procedure, reduced cognitive workloads, as well as facilitating supervised and collaborative autonomy of MIS procedure.

1.2 AIM OF THE THESIS

The aim of this PhD thesis consists of developing a pre-operative curvilinear path planner for steerable needles, a realistic time-bounded simulator that mocks brain deformations due to needle insertion, an intra-operative path-replanner accounting for a constrained target pose, an intra-operative anatomical deformation and a motion controller that can steer a needle to produce the desired trajectory. An synthetic graphical abstract of this PhD thesis is shown in Figure 1.1. The proposed methods consider the EDEN2020 PBN and robotic system as a case study.

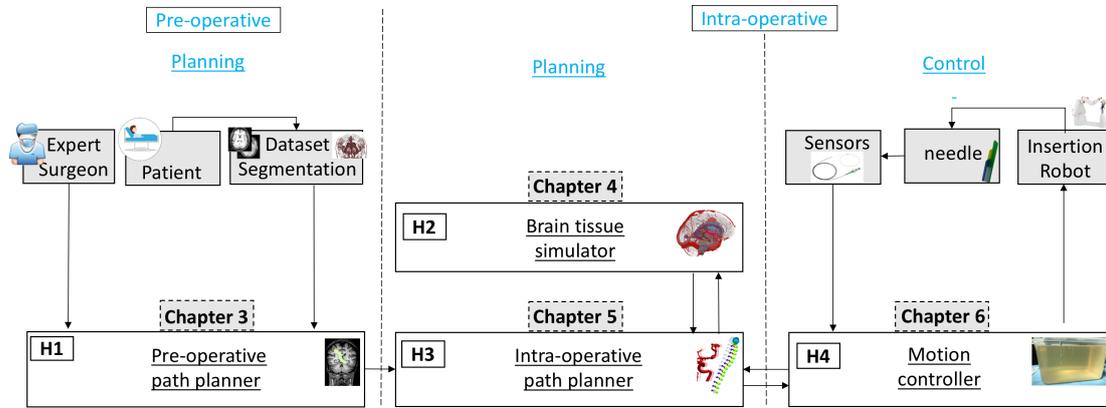


Figure 1.1: *Synthetic Graphical abstract of the PhD thesis. The research hypotheses that guide this PhD work are presented*

Starting from the open technical challenges, the research hypotheses that guide the PhD work can be summarized as follows:

1. **Hypothesis 1:** The curvilinear pre-operative path planner can solve the planning query by computing a kinematically feasible solution for the steerable needle.
 - (a) The path is computed in accordance with the optimization criteria of minimum path length, curvature and maximum obstacle clearance (**Hypothesis 1.1**)
 - (b) The computational time, safety distance-from-obstacles and smoothness can be kept consistent with standard pre-operative planning algorithms (**Hypothesis 1.2**)

1.2)

2. **Hypothesis 2:** The simulator realistically mocks brain deformations due to needle insertion.
 - (a) The deformations proves to be a close match with the real brain (**Hypothesis 2.1**)
 - (b) The stability, accuracy, and real-time performance can be kept consistent with standard dynamic simulators (**Hypothesis 2.2**)
3. **Hypothesis 3:** The presented intra-operative steerable needle path re-planner is fast (**Hypothesis 3.1**) and accurate (**Hypothesis 3.2**).
4. **Hypothesis 4:** The motion controller can steer a needle to produce the desired trajectory.
 - (a) The needle is inserted following the desired trajectory (**Hypothesis 4.1**)
 - (b) The accuracy and robustness can be kept consistent with standard needle insertion control strategies. (**Hypothesis 4.2**)

To investigate the aforementioned hypotheses, specific research activities have been performed, which are thoroughly discussed in this PhD dissertation.

To properly contextualize the research topics expressed in Chapter 2 a systematic literature survey was carried out by reviewing methods for steerable needles path planning in MIS. The survey focuses on the analysis of the clinical aim and background, the path planning method, the validation procedure and the quantitative results.

In Chapter 3, **Hypothesis 1** is investigated, and a new method for pre-operative path planning is presented using the PBN as a case study. **Hypothesis 1.1** is investigated by proposing a deductive reasoning approach according to explicit knowledge modeled over domain experts suggestions while **Hypothesis 1.2** is addressed implementing an inductive learning approach relying on demonstrations performed by expert operators. The method was validated against manual and automatic solutions from literature in-silico KN insertions.

In Chapter 4, **Hypothesis 2** is investigated, and a new method for simulating a dynamic environment that mocks brain deformations due to catheter insertion is presented. **Hypothesis 2.1** is addressed by fine-tuning the deformation parameters with deformation data for PBN insertion in a composite hydrogel phantom, while **Hypothesis 2.2** is addressed by implementing a Position based dynamics (PBD) approach. The method was validated using recorded brain deformation on in-vivo animal KN insertions of a PBN.

In Chapter 5, **Hypothesis 3** is investigated, and a new method to support neurosurgeons during the intra-operative procedure to react to a KN dynamic environment is presented. **Hypothesis 3.1** and **Hypothesis 3.2** are addressed by implementing an Inverse Reinforcement Learning (IRL) method. The method was validated in a dynamic environment against a solution from literature in-silico KN insertions.

In Chapter 6, **Hypothesis 4** is investigated, and a new method to control the deflection of a flexible needle tip to achieve an accurate path is presented. **Hypothesis 4.1** and **Hypothesis 4.2** are addressed by implementing an IRL method. Validation was performed through in-silico trials and in-vitro experiments in phantom-brain gelatin.

In Chapter 7, the conclusions, the scientific implications and the future perspective of this PhD work are reported and discussed.

Appendices are reported in Chapter 8.

An extended graphical abstract of this PhD thesis is shown in Figure 1.2.

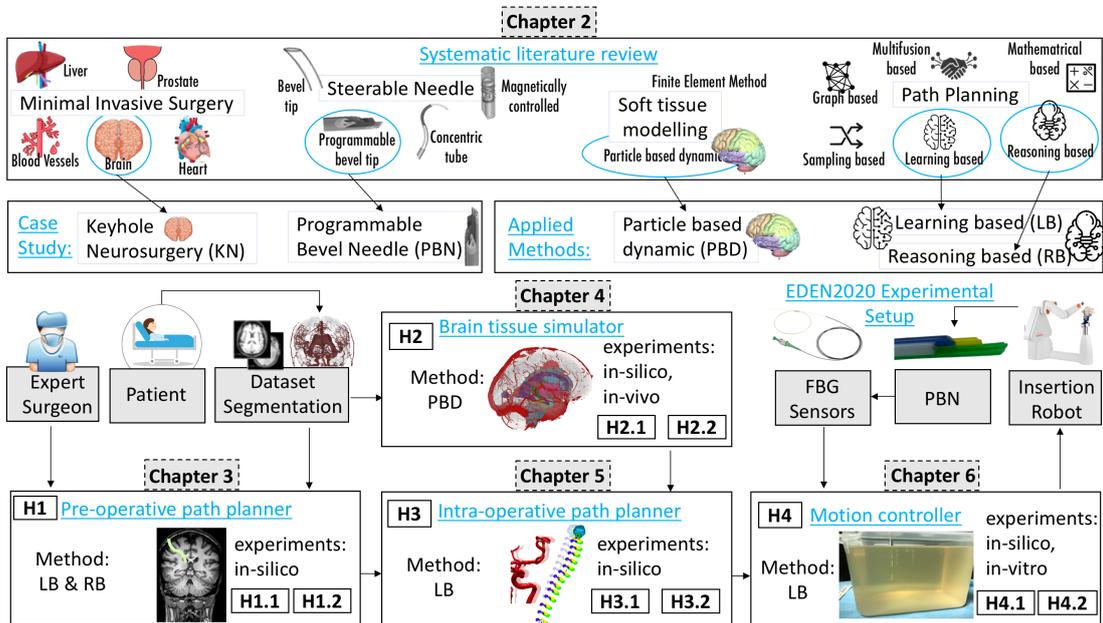


Figure 1.2: *Extended Graphical abstract of the PhD thesis. The research hypotheses that guide this PhD work are presented, along with the performed activities and the solutions employed to validate the proposed methods.*

Steerable needles path planning for minimally invasive surgery: a systematic review

Over the last two decades, different research groups have focused their efforts on the development of needles able to steer inside the tissue. Differently from conventional needles, for which the insertion path can be planned and performed by the clinician, the complex kinematics of steerable needles make the path planning unbearable requiring the aid of automatic or semi-automatic path planning solutions. In the present chapter, The PhD thesis is contextualized by proposing an overview of different path planning techniques used for surgical steerable needle/catheters, along with a review of important clinical applications.

A systematic and careful literature search in major databases such as Pubmed, Scopus and Web of Science was carried out using “steerable catheter/needle” and “path planning” as main keywords. Further references were integrated by cross-referencing from key articles. The analysis carried out for each paper concerns the clinical aim and background, the path planning method, the validation procedure and the quantitative results.

84 studies out of 337 were identified, which actually made use of path planning algorithms for steerable needle/catheters. Such paradigms include graph-based, sampling-based, mathematical-based, learning-based, reasoning based and multifusion-based methods. The most common surgical procedure was the biopsy, the most common anatomical area was the liver, and the most common pathology was cancer. Only half of the framework has a re-plan strategy for an intra-operative path planner. Most of the algorithm were validate in silico. Bevel tip was the most frequently used needles, followed by programmable bevel tip needles. 3D model data were the most frequently used input features. Electromagnetic sensor and camera were the most frequently used sensors.

Regarding the path planning method, we witnessed a growth, in the latest ten years,

of the number of studies proposing steerable needle/catheter as a new possible tool in operating surgery and proposing their own method to plan the optimal trajectory. Despite the remarkable performance and the level of readiness reached in simulations by many solutions reported in this survey, what is often missing is a thorough evaluation of the planning performance in conditions that mimic a real needle insertion through, e.g., in-vivo or ex-vivo experiments. This would make it possible to assess the feasibility of the proposed algorithm in the presence of tissue and modelling uncertainties which represents, in our opinion, one of the big open questions in steerable needle path planning. The systematic review resulted in a journal paper [2]¹ currently under review.

2.1 INTRODUCTION

In current clinical practice, a growing number of minimally invasive procedures relies on the use of needles, such as biopsies, brachytherapy for radioactive seeds placement, abscess drainage, and drug infusion. Compared to standard open surgeries, the needle's small diameter allows to access the targeted anatomy inflicting limited tissue damage and thus reducing the risks for the patient and speeding up the recovery.

Over the last two decades, different research groups have focused their efforts on the development of needles able to steer inside the tissue. These needles can perform curvilinear trajectories, maximizing the distance from sensitive anatomical structures to be avoided and can reach targets that would otherwise be inaccessible via rectilinear insertion paths.

Differently from conventional needles, for which the insertion path can be planned and performed by the clinician on the basis of the target location and the patient anatomy, the complex kinematics of steerable needles make the path planning unbearable requiring the aid of automatic or semi-automatic path planning solutions. Automatic planners often require the definition of the starting and target point as inputs from the clinician, as well as the patient anatomy that can be obtained from standard imaging modalities as X-ray fluoroscopy, Ultrasound Image (USi), Computed Tomography (CT), Computer-Tomographic Angiography (CTA) or Magnetic Resonance Imaging (MRI). Imaging information can also be used intra-operatively as feedback to reduce possible needle insertion inaccuracies arising from tissue inhomogeneities, obstacles, and target movements, as well as uncertainty in needle motion modelling.

This Chapter proposes a survey on the principal methods proposed in the literature over the last two decades for steerable needle path planning, which aims to investigate research trends and challenges.

2.1.1 Type of needle steering

Different steerable needle designs and approaches have been proposed in the literature [263], including base manipulators [100], tissue manipulators [260], a number of different bevel-tip needles [278] (such as duty-cycled bevel-tipped needles [69] and programmable bevel-tip needles [275]), pre-curved stylet [194], concentric-tube needles [95], tendons actuated needle [262], active cannulas [290], magnetically controlled

¹Segato, A., Li, Z., Favaro, A., and De Momi, E. (2021) Steerable needles path planning for minimally invasive surgery: a systematic review. *IEEE Transactions on Robotics*. (submitted on May 2021)

needle [205], hydraulically driven catheter [78], fracture-direct waterjet needle [22], radiofrequency ablation needle [286] and micro/nano robots [164]. All steerable needle designs and approaches of the reviewed article [263] are reported in Figure 2.1 and described in Table 2.1. This survey aims to provide an integrated, synthesized overview of the path planning of those needles and their derivations.

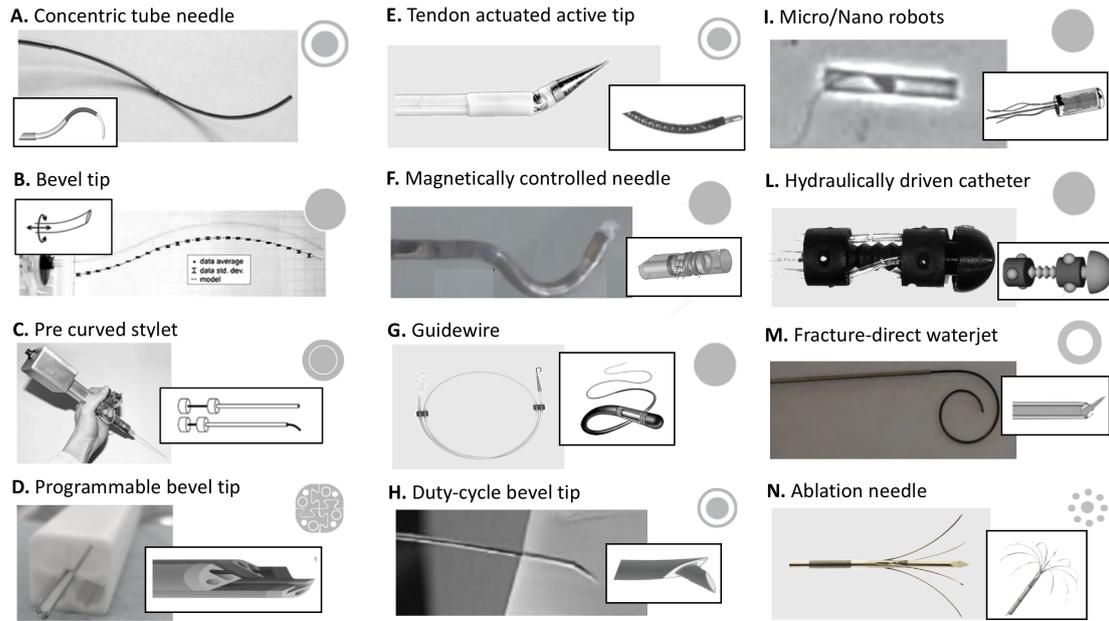


Figure 2.1: Steerable Needles. Example of (A) Concentric tube needle consisting of three tubes [66]. (B) Bevel tip needle [277]. (C) Pre curved stylet [194]. (D) Programmable bevel consisting of four interlocked segments [145]. (E) Example of a tendon actuated active tip-steered needle with a ball joint and a conical tip [262]. (F) Magnetically controlled needle [257]. (G) Guidewire [56]. (H) Duty-cycle bevel tip [180]. (I) Micro/Nano robots [288]. (L) Hydraulically driven catheter [125]. (M) Fracture-direct waterjet [23]. (N) Ablation needle [41]. The cross-section of needles is shown on the upper right. (part of the image is courtesy of the [263]).

Table 2.1: Needles category

Needle	definition
concentric tube needle [66]	By means of rotating and extending the tubes with respect to one another, cannula curvature, tip position, and tip orientation can be adapted. Since the actuation forces are generated internally, these instruments do not need environmental contact to steer.
bevel tip [277]	These needles are typically controlled by translating or rotating the needle along its longitudinal axis. It is often assumed that asymmetric interaction forces at the tip will cause a bending effect with constant curvature, although this assumption does not necessarily hold for more compliant soft tissue.
pre curved stylet [194]	On following the stylet path with the cannula, tissue reaction forces will cause the cannula to bend in the same direction. The degree of steering can be controlled by varying the exposed length of the stylet. Once in place, the actuation mechanism and stylet can be retracted, leaving only the cannula in place. From a control perspective, a reorientation of the steering direction can be achieved in its retracted state, minimizing tissue interference and torsion.

2.2. PROBLEM STATEMENT AND MOTIVATION

programmable tip [145]	bevel	A quick developing steerable needle design, based on biomimetic concepts, makes the use of a programmable bevel tip. The latest version of this device consists of four interlocked segments that can slide along one another. Each segment is actuated by a linear motor at the base and allows the respective part of the tip configuration to change. Tip control and steering occurs on behalf of the relative offsets of the interlocked segments.
tendon actuated tip [262]	active	Tendon actuated tip steering techniques have been integrated in a variety of medical instrument designs, such as laparoscopic tools or endovascular guidewires. Tip actuation occurs outside the patient and the mechanical propagation of actuation signals can, for instance, be achieved by cable or wire. The needle tip is placed on the top of a flexible cannula. In between, any type of compliant structure or joint mechanism can be used. Cannula bending results from asymmetric tip–tissue interaction forces.
magnetically controlled needle [257]		A permanent magnet at the proximal end of a flexible needle is steered by an external magnetic field, and the resultant tip-deflection angle bends the flexible body like a bevel-tip needle.
guidewire [56]		The guidewire is the device used to guide the catheter into place during CVC insertions. This is a single wire with a hub at one end that is inserted into the catheter prior to placement. The stylet is used to add rigidity to the catheter during insertion.
duty-cycle [180] bevel tip		The configuration of this needle looks similar to a bevel tip needle and can bend more than a standard bevel-tip needle. The curvature of the needle is adjusted using the “duty cycling” control strategy, where the needle is simultaneously rotated and advanced.
micro/nano robots [288]		The design of an efficient nano/microscale machine requires a swimming strategy that operates under Brownian motion. Since traditional power components and batteries are not possible at these small scales, innovative and bio-inspired design principles are needed to meet the power and locomotion requirements. Typically, these tiny machines rely on either chemically-powered motors that convert locally supplied fuels to force and movement or externally-powered motors that mostly utilize magnetic and ultrasound energies to drive their motion.
hydraulically catheter [125]	driven	Another actuation principle is that the catheters are composed of one or several hydraulic bellows and the distal tips are steered by hydraulic pressure of bellows. The variation of hydraulic pressure inside the bellows modifies their lengths and, thus the bending of the catheter. The catheter is filled with water and its bending angle is controlled by the suction of water.
fracture-direct waterjet [23]		The principle behind this needle is to first control the direction of the tissue fracture with a water jet, after which the needle will follow during the next insertion. In this work, the direction of tissue fracture is controlled by an angled water jet nozzle and water jet speed control, and then the flexible Nitinol needle follows. By changing the speed of the water jet and thus the depth of cut, the radius of curvature can be controlled.
ablation needle [41]		Catheters designed to deliver radiofrequency energy to tissues during ablation procedures. With a wide portfolio offering, they include multi-curve options and lateral deflection features to address the needs of diverse anatomies.

2.2 PROBLEM STATEMENT AND MOTIVATION

2.2.1 Path planning problem

We considered the path planning problem as in [136, 141].

The workspace $\mathcal{W} = \mathbb{R}^N$ (with $N = 2$ or $N = 3$) represents the environment where the robot \mathcal{R} can move. We define $\mathcal{O} \subset \mathcal{W}$ the part of the workspace occupied by obstacles. The configuration space, or \mathcal{C} -space, is the space of all possible robot configurations defined considered robot’s kinematics. The part of the configuration space occupied by obstacles can be defined as:

$$\mathcal{C}_{obs} = \{\mathbf{q} \in \mathcal{C} | \mathcal{R}(\mathbf{q}) \cap \mathcal{O} \neq \emptyset\} \quad (2.1)$$

where with $\mathcal{R}(\mathbf{q}) \subset \mathcal{W}$ we refer to the points in the workspace occupied by the robot in the specific configuration \mathbf{q} . The set of configurations that avoid obstacle collision is $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$.

Definition 1: Path planning

Robots are assumed to operate in N -dimension (\mathcal{R}^N) space. Given a function $\tau(t) \rightarrow \mathcal{R}^N, t \in [0, T]$ and a starting and goal configuration $\mathbf{q}_I, \mathbf{q}_G \in \mathcal{C}_{free}$ in a way that $\tau(0) = \mathbf{q}_I$ and $\tau(T) = \mathbf{q}_G$. The path planning problem, consists in computing a continuous process without break so that $\tau(t) \in \mathcal{C}_{free}$ for all $t \in [0, T]$.

Definition 2: Optimal path planning

Given a planning query and a cost function, c linked to case-specific optimization criteria, $c : \Sigma \rightarrow \mathbb{R}_{\geq 0}$, where Σ represents the set of all paths, the optimal path planner is the one able to compute a feasible path τ^* such that $c(\tau^*) = \min\{c(\tau) : \tau \text{ is feasible}\}$. If no such path exists, report failure.

Definition 3: Trajectory planning

Path planning and trajectory planning are two separate concepts. The first one concerns the computation of a continuous curve in the configuration space from \mathbf{q}_I to \mathbf{q}_G while trajectory planning refers to the problem of taking the result of the path planning algorithm and define how to move along this path in terms of velocities and accelerations. A trajectory is thus a set of states associated with time [284].

2.2.2 The path planning algorithm taxonomy

Algorithms of path planning have been arising since the last century; the methodologies have different characteristics and can be used for different robots and environments. This review divides path planning algorithms into six categories.

- Graph-based algorithms, such as Dijkstra algorithm [62] and A-star [110], are based on the discrete approximation of the planning problem. Many methods represent the environment as a square graph, or an irregular graph [134], or a Voronoi diagram [92]. A search is performed in order to find an optimal path. These algorithms are known to be “resolution-complete”, as they can determine whether a solution exists in a finite time, and “resolution-optimal” since they can estimate the best path, according to length, given the specific resolution of the approximation. This approach may also be used for identifying a restricted area where further optimisation refinements can be performed [124].
- Sampling-based algorithms, such as Probabilistic Road Map (PRM) [140, 142] and Rapidly-exploring Random Tree (RRT) [156], are a type of path planner that generates the robot’s path by sampling sequentially different points in the robot’s workspace and gradually constructs a data structure representing collision-free paths. They feature probabilistic completeness, i.e. if a solution to the path planning problem exists, they will eventually find it.
- Mathematical-based algorithms include linear algorithms, such as Spline Interpolation [139] and optimal control, such as Gradient Based Optimization (GBO) [31]. These methods model the environment (kinematic constraints) and the system (dynamic) and then restrict the cost function with all kinematic and dynamic constraints that are inequalities or equations to obtain an optimal solution. Further, under this category fall optimization-based motion planning methods, the

canonical examples being, e.g., CHOMP [219], Trajectory optimization [231], STOMP [132], and ITOMP [197].

- Learning-based methods originate from imitating how humans or other natural creatures behave or think. Evolutionary Algorithms were propounded to solve the problem where traditional linear programming and dynamic programming often fail to solve with a large number of variables. The proposed and implemented Evolutionary method the Genetic Algorithm (GA) [117]; Another subcategory of learning-based path planning method is Neural network (NN) [97].
- Reasoning-based method represents an effective tool for knowledge representation and reasoning. Thanks to its declarative approach where a problem is modelled by means of a logic program composed of a collection of rules. An example is represented by Answer Set Programming (ASP) [170].
- Multifusion-based method is an approach that has recently emerged to improve the performance of 3D path planning algorithms; algorithms can benefit from each other in this way. Typically, algorithms tend to merge in a layer-by-layer fashion and aim to plan an optimal path (with better real-time or non-local optimal performance). For example, an algorithm that tends to fall into a local minimum can be combined with another to overcome this limitation. Therefore, this study classifies these types of algorithms, which are introduced by combining several algorithms together to achieve better performance, as multi-fusion based algorithms.

2.2.3 Path planner in minimally invasive surgery

High precision and accuracy in reaching target locations inside the human body are necessary for the success of minimally invasive procedure, such as tissue sample removal (biopsy), brachytherapy, and localized Drug Delivery (DD). Flexible/steerable needles may enable the surgeon to reach targets deep inside solid organs while avoiding sensitive structures (e.g. blood vessels). Path planning in MIS has to meet multiple requirements at the same time in order to achieve the proper level of safety and efficacy. These requirements can be the needle technical specifications or the translation of clinical aspects into specific constraints.

Conceptually, the requirements can be divided into *strict* and *soft* constraints [75]. Strict constraints are conditions that must be satisfied (e.g., the needle should not intersect any obstacle, or the path curvature should not exceed the maximum degree of curvature admissible by the needle) and define the available workspace for the path planning algorithms, limiting the path search to regions where solutions can lie, excluding the unfeasible ones. Instead, soft constraints can be considered as the optimisation objectives and are included in the cost function to drive the path search toward the best solution, among those already selected as feasible through the application of hard-constraints. The cost function to be minimized generally takes the following form:

$$F(x) = \sum_{i=1}^N k_i * f_i(x)$$

with f_i representing a specific soft constraint opportunely weighted by a factor k_i , which is often chosen empirically through multiple experiments or on the basis of the

clinicians' suggestions.

Different planners include different objectives but, generally, one or more of the following are included:

- minimization of the path length
- maximization of the distance from one or multiple anatomical obstacles
- maximization of the probability to reach the target

Sometimes, the minimization of the path curvature is also considered, together with reducing its standard deviation to provide a smoother path and reduce the effort required to control the needle insertion.

This survey will discuss different objectives considered in the reviewed literature.

2.3 METHODS

According to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines, a systematic literature review was performed. In particular, Pubmed, Scopus and Web of Science databases were searched to identify all potentially relevant studies back to January 1, 2005. The search queries were carefully built with the guidance of a professional librarian using search terms related to 'steerable catheter/needle' and 'path planning'. A comprehensive list of the keywords used for the search is reported in Appendix 8.1. All biomedical studies that evaluated path planning for steerable needles/catheters were included; duplicates were discarded using the *EndNote* reference management software. Following the elimination of duplicates, a careful screening of titles and abstracts was made to identify papers relevant to our research topic. Any work that matched at least one of the following exclusion criteria was crossed out:

- (i) no full-text available,
- (ii) not this application,
- (iii) conference abstracts,
- (vi) books,
- (vii) book chapters,
- (viii) non-English language.

After a proper check of full-texts and references, a total of 84 articles/reviews were identified as eligible and hence included in this systematic review, as reported in Figure 2.2. Any article appearing to help our research was included and classified; nevertheless, we decided not to cover papers concerning catheters/needles modelling, tracking and control. The analysis carried out for each paper follows the scheme represented in Figure 2.3, where we have analyzed the clinical aim and background, the path planning method, the validation procedure and the quantitative results. Data considered from each study were:

- (1) First author, year of publication
- (2) Single or multi query planner
- (3) Automatic or semi-automatic planner
- (4) Pre or intra-operative planner
- (5) Risk assessment
- (6) Clinical aim
- (7) Pathology, anatomical area
- (8) Needle, sensor
- (9) Needle kinematic

PRISMA Flow Diagram of Systematic Review

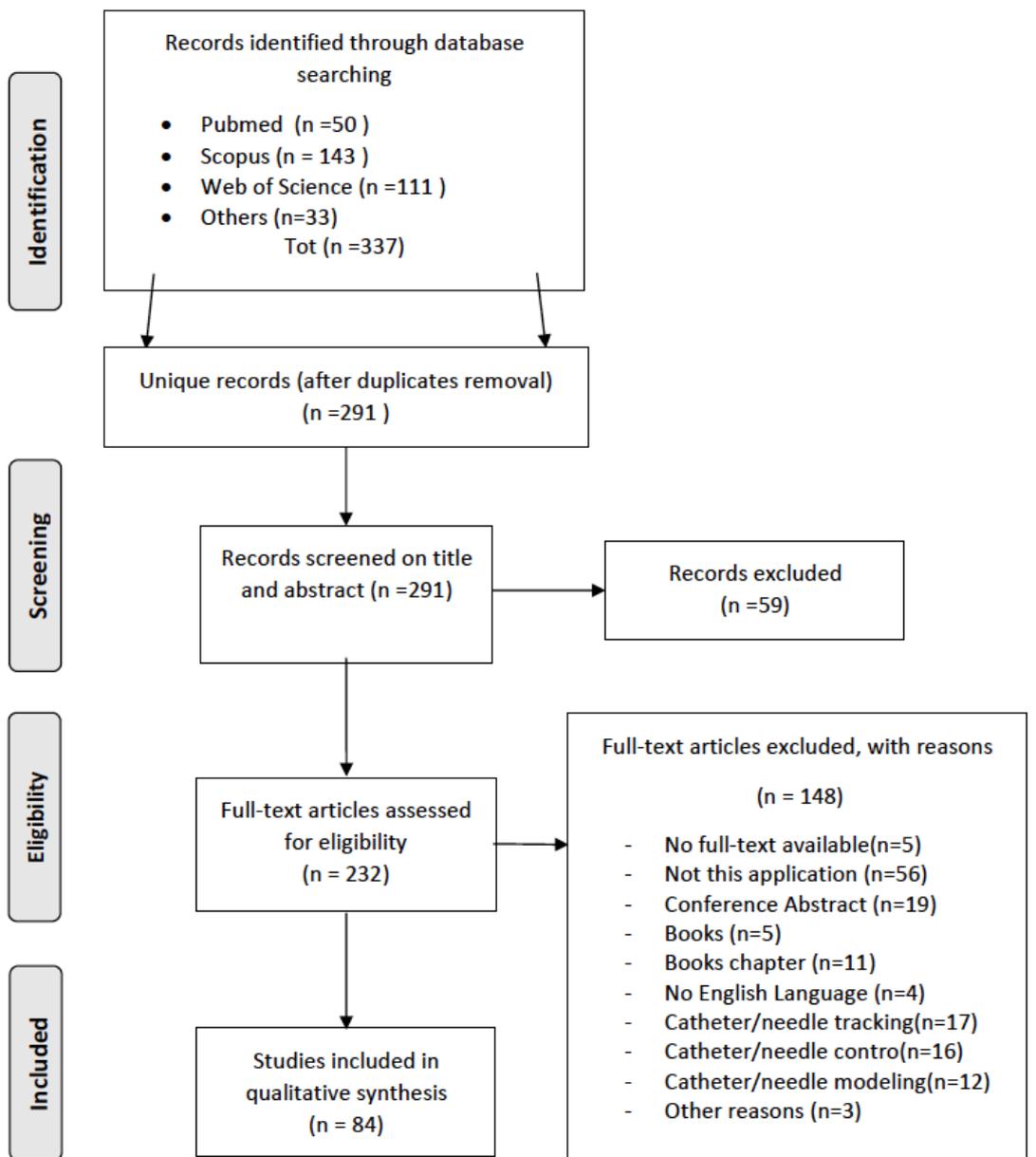


Figure 2.2: PRISMA Flow Diagram of Systematic Review Identification, Screening, Eligibility and Inclusion. 84 studies were included in the final analysis out of the 324 screened.

constraints (10) Path Planning method (11) Type of data (12) Dataset (12) Static or dynamic environment (14) Tissue modelling (15) Validation in-silico/in-vitro/ex-vivo/in-vivo (16) Tool (17) Benchmark measure (18) Results

On this basis, we computed the distribution of all published articles within the domains of the path planning algorithm. We considered a quantitative synthesis to be inappropriate due to the heterogeneity in the applications. A qualitative synthesis of

results is hence provided next by means of a narrative approach.

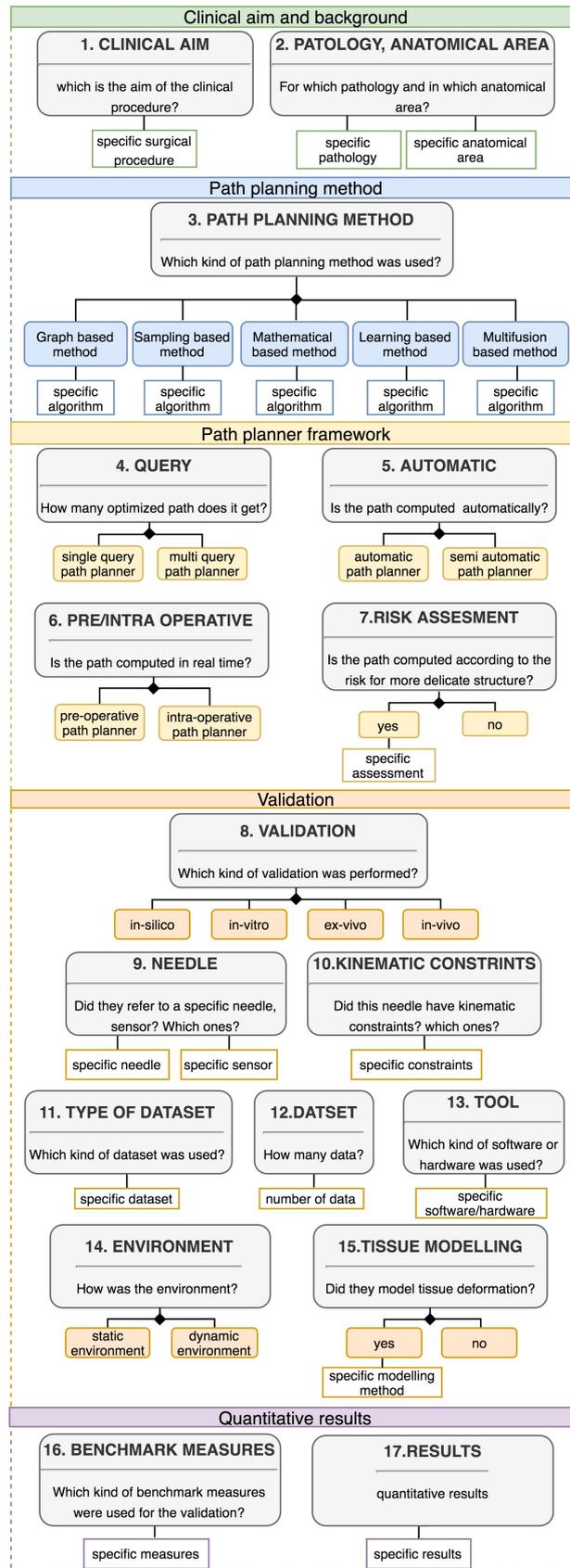


Figure 2.3: Papers Analysis Method. Schematization of the analysis carried out for each paper

2.4 EVALUATION METRICS

For all path planning in MIS applications, performance measurement is an essential task. Benchmark measures used for the evaluation of the reviewed studies are in Figure 2.4 and explained in Table 2.2. Time Complexity (*time*), path length (*length*), targeting error (*tError*), tracking error (*trError*) success rate (*SR*) and number of generated optimal paths (*#p*) are metrics widely used to evaluate performance in path planning task. Furthermore, in the case of path planning in MIS, other parameters are evaluated due to the high risk of the procedure in avoiding delicate anatomical areas or reaching targets at a certain angle that facilitate their treatment. These parameters are the probability of obstacles avoidance (P_a), the minimum (d_{min}) and average (d_{avg}) distance from the obstacles, and the target's insertion angle (α_{angle}).

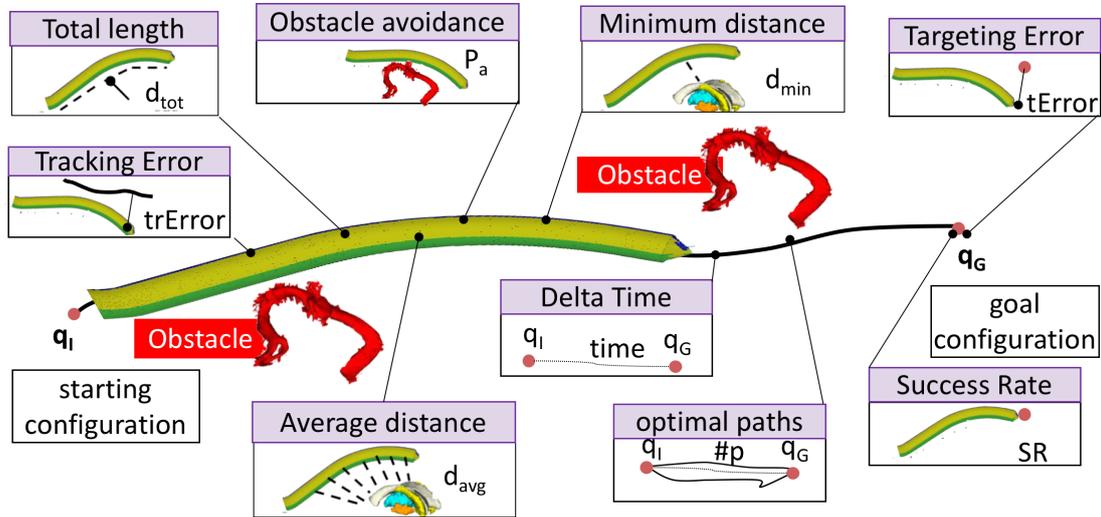


Figure 2.4: **Benchmark Measure.** Path planning performance measures for steerable needles/catheter

Table 2.2: **Benchmark measures explained**

Benchmark measures	definition
time (Time Complexity)	is the computational complexity that describes the amount of time it takes to run an algorithm.
tError (Targeting Error)	is the target position accuracy that usually influences the surgery consequence significantly.
trError (Tracking Error)	is the tracking position accuracy along the whole motion plan.
P_a (Probability of avoidance)	describes the risk of colliding with obstacles.
SR (Success Rate)	measures the algorithm's robustness in reaching the target within the constraints.
d_{min} (minimum distance)	is the minimum distance between the needle/catheter body with obstacles.
d_{avg} (average distance)	is the average distance between the needle/catheter body with obstacles.
$\#p$ (number of paths)	describes the number of generated optimal paths in one trail.
α_{target} (target insertion angle)	is the target orientation accuracy.
k_{mean} (mean curvature)	is the mean value of curvature that describes the bending extent of a trajectory.

2.5 RESULTS

Out of the 337 citations initially identified in the selected databases, 84 were selected by title/abstract and full-text screening. Regarding the clinical aim and background,

as synthesized in Figure 2.5, steerable needle/catheter are applied for different surgical procedures, such as DBS, DD, Radio-Frequency Ablations (RFA), brachytherapy, tissue ablation, biopsy, Radioactive source implantation (RSI), stent placement, drainage, Temporal Bone Surgery (US) and more in general for MIS and KN, in the various anatomical area of the patients such as brain, liver, heart, breast, lung, blood vessels and prostate, with a wide variety of pathology, such as cancer, Parkinson’s Disease (PD), Alzheimer’s Disease (AD), arrhythmia, atrial fibrillation, heart disease, brain disease, stenosis, aneurysm, vascular disease. The most common surgical procedure, after a generic MIS procedure, was the biopsy, the most common anatomical area was the liver, and the most common pathology was cancer.

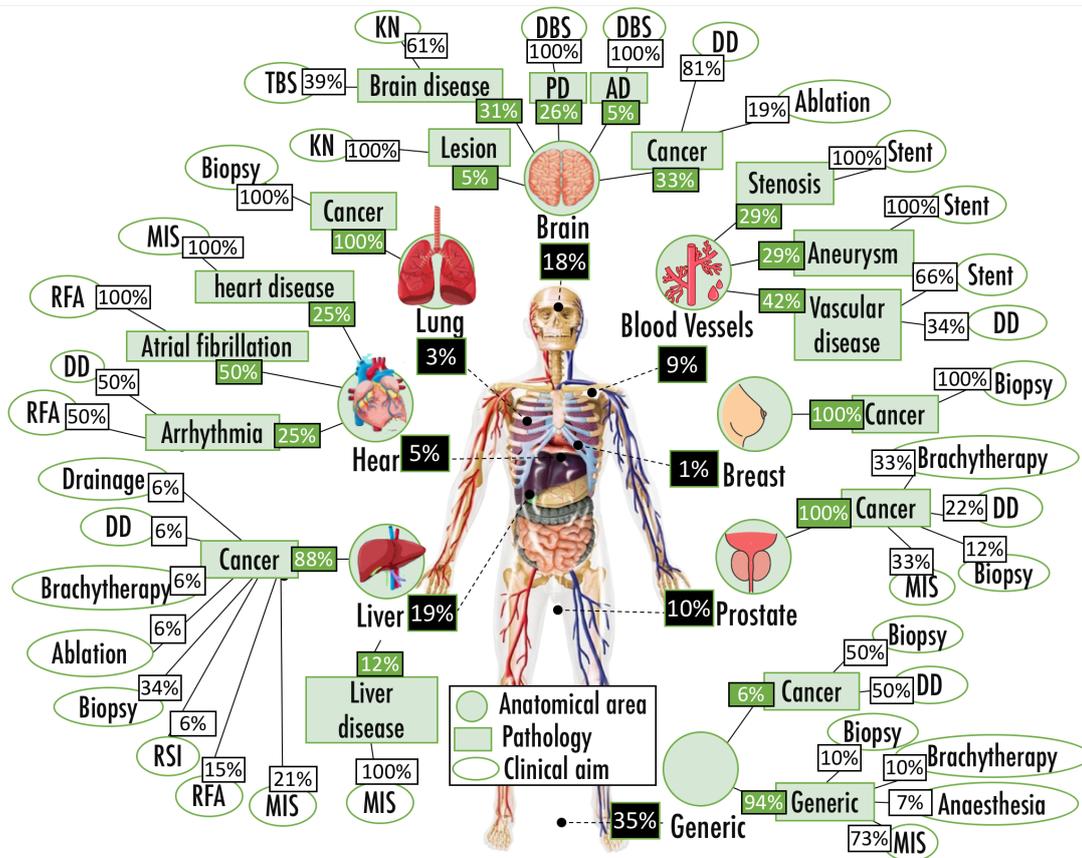


Figure 2.5: Clinical aim results Distribution. Distribution of the specific anatomical area, pathology and surgical procedure based on all the analyzed published article

As shown in Figure 2.6, the path planner framework in which the algorithms were run are all pre-operative path planner, and only half of the framework has a re-plan strategy for an intra-operative path planner. Most of the proposed path planners are automatic, with 6% of semi-automatic path planners that involve the surgeon’s assistance during the planning of the path/trajectory. Only 26% of the planner provides a risk map or a risk analysis for the most delicate structure involved in the procedure. 77% of the planners were single query planners, where only one optimal path is provided at the end of the computation. In contrast, the others were multi-query planners, intended for those conditions when there is the need to search for different queries on the same

static environment. Regarding the validation of the algorithms, 71% of the developed methods were validated in-silico, 24% in-vitro and 5% ex-vivo; none of the cited works was validated in-vivo. The tools used for creating the environment for the validation were *Unity3D*, *3DSlicer* and *Matlab* software for the in-silico validation, phantoms for the in-vitro validation and biological tissue for the ex-vivo validation. Only 43% of the investigated works were validated in a dynamic environment and simulated tissue deformation with noise or Finite State Element (FEM) for in-silico validation, gelatin and silicon for the in-vitro validation and biological tissue for the ex-vivo validation.

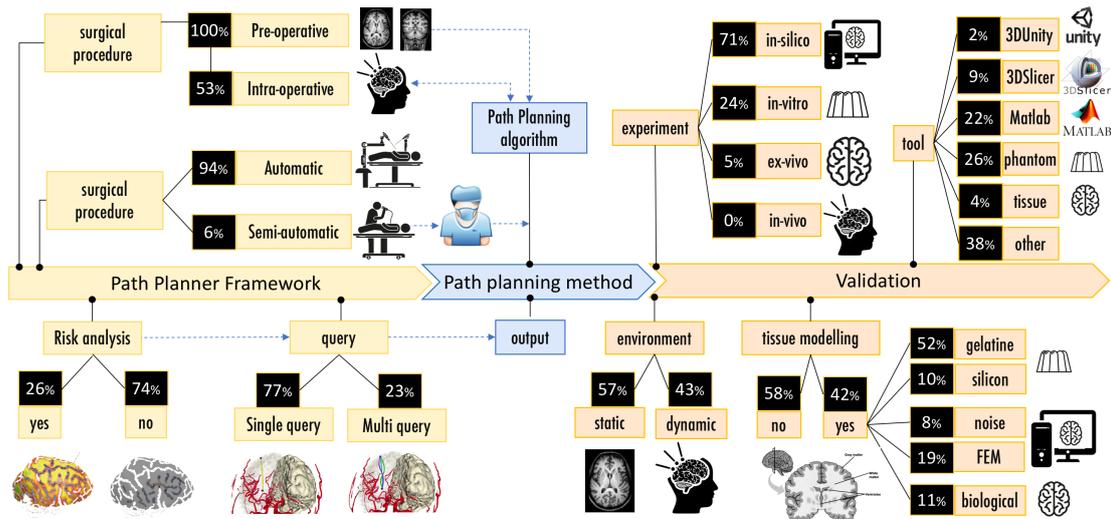


Figure 2.6: Path Planner Framework Results Distribution. Distribution of specific surgical procedure, risk analysis, type of query, type of experiments, type of environment, the method used for tissue modelling and tool based on all the analyzed published article

As shown in Figure 2.7, commonly used type of catheters/needles were bevel tip needle, concentric tube needle, tendon actuated active tip, magnetically controlled needle, micro/nano robots, hydraulically driven catheter, active cannula, pre curved stylet, programmable bevel tip, guidewire, duty-cycle bevel tip needle, radiofrequency ablation needle and fracture-direct waterjet. Bevel tip was the most frequently used needles (49%), followed by programmable bevel tip needles (25%). Commonly used type of data were CT, MRI, 3D model, 2D map and USi. 3D model data (48%) were the most frequently used input features. Commonly used sensors were Electromagnetic (EM) sensor, Ultra Sound (US), force sensor, and camera. EM(39%) and camera (39%) were the most frequently used sensors.

As shown in Figure 2.8, regarding the path planning method, we witnessed a growth, in the latest ten years, of the number of studies proposing steerable needle/catheter as a new possible tool in operating surgery and proposing their own method to plan the optimal trajectory. Such paradigms include graph-based methods (5%), sampling-based methods (41%), mathematical-based methods (24%), learning-based methods(13%), reasoning based methods (2%) and multifusion-based methods (15%). Algorithms used for each of these categories are described in the following Subsections and summarized in Table 2.3.

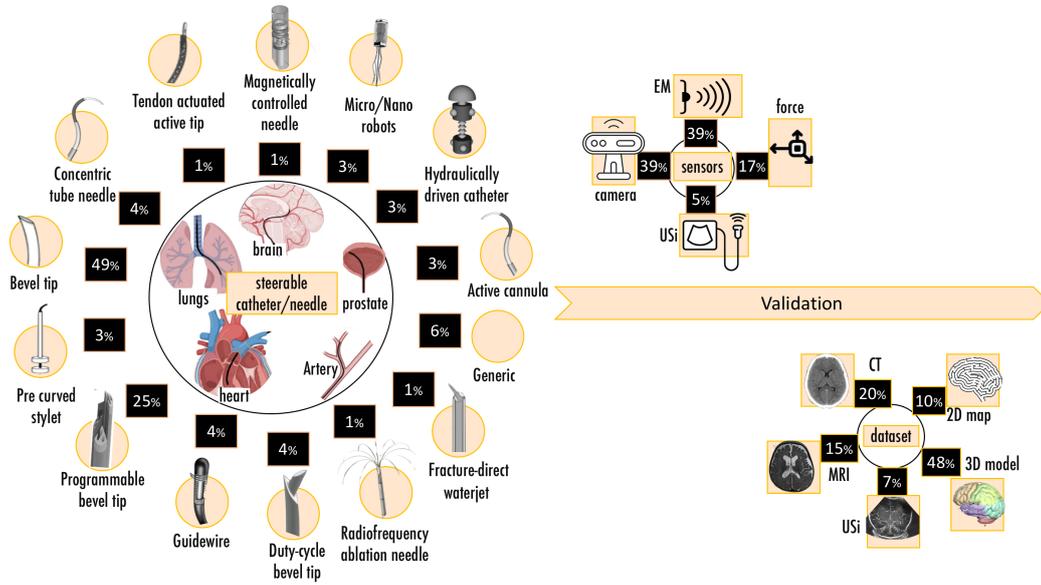


Figure 2.7: Validation Results Distribution. Distribution of specific steerable catheter/needle, sensors and type of data used based on all the analyzed published article

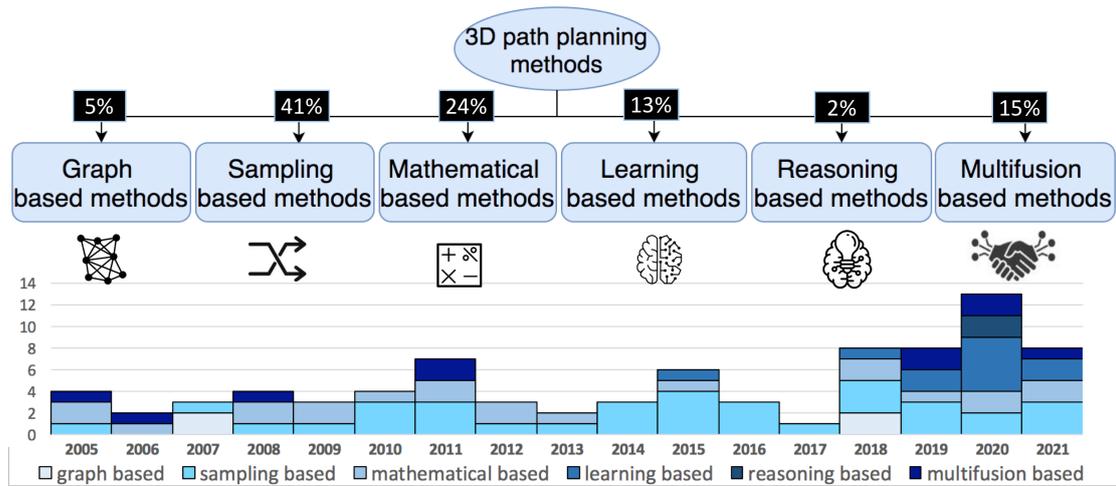


Figure 2.8: 3D Path Planning Methods for Robotic Steerable Needle. Divided in Graph-based, Sampling-based, Mathematical-based, Learning-based, Reasoning-based and Multifusion-based, and their overall and yearly distribution from 2005 to 2021

Table 2.3: Methods explained

Algorithm	definition
Graph-based methods	
Dijkstra	Dijkstra algorithm [62] is an algorithm for finding the shortest paths between nodes in a graph, which is also called as Shortest Path First (SPF) algorithm. In each iteration, it chooses the node with the minimal cost to expand the tree.
A-star	A-star is an extension of Dijkstra’s algorithm [110], which reduces the total number of states implementing an heuristic estimation of the cost from the current state to the target state. A-star can converge very fast and ensures optimality as well.

2.5. RESULTS

NNG	Nearest neighbor graph algorithm, also called Lifelong Planning A-star (LPA*), is an incremental heuristic search algorithm based on A-star, which is first proposed in [148]. NNG can adapt to dynamic changes in the graph without recalculating the entire graph.
BFS	Breadth First Search algorithm [60] explores a graph by expanding the most promising node chosen according to a specified rule. By using a heuristic that attempts to predict how close the end of a path is to a solution, the paths which are judged to be closer to a solution are extended first. Due to its greedy strategy, the path cost from the start point to current point is ignored.
Heuristic	A heuristic technique, or a heuristic, is any approach to problem solving or self-discovery that employs a practical a not guaranteed optimal method , but is nevertheless enough to achieve an immediate and short-term goal or approximation.
Sampling-based methods	
RRT	Rapidly exploring random tree method is first proposed by LaValle [156]. The method attempted to solve route planning problems under holonomic, non-holonomic and kinodynamic constraints. RRT quickly searches the configuration space to generate a path connecting the start node and the end node. At each step a new node is sampled; if the extension from the sampled node to the nearest node is successful, a new node is added.
RRT-connect	The RRT-connect [151] is a structures that incrementally expands two RRTs rooted at the start and the target configurations. These two RRTs explore the space advancing towards each other with a simple greedy heuristic until they connect.
RG-RRT	Reachability Guided Rapidly-exploring Random Tree are a combination of RRT and a reachability-guided sampling heuristic that construct the path through a sequential connection of arcs with variable curvature bounded within the maximum curvature achievable
arc-RRT	the arc-RRT is an extension of RG-RRT The objective of the algorithm is to find a combination of circular arcs capable of taking the robot from its initial configuration to a final configuration while respecting the system constraints. An arc is defined by the robot's curvature and its two extremity configurations. The final extremity of each arc should correspond to the next arc's initial extremity, not only in position but also in orientation, to have the path continuity.
RRT-star	The RRT-star [138] represent an "anytime" variation of the original RRT: it quickly identifies an initial feasible plan, then, as the plan execution process, it improves the plan toward an optimal solution. This is obtained by rewiring the RRT by removing redundant edges and keeping the shortest or minimum-cost path. As with RRT, an initial feasible solution is found quickly but, in addition, RRT* almost surely converges to an optimal solution.
AFT	The Adaptive Fractal Tree takes advantage of the fractal theory and the architecture of graphics processing units (GPUs) to parallelize the planning process and enhance the computation performance and achieve online replanning.
AHFT	AFT are the basis for the Adaptive Hermite Fractal Tree where the fractal structure was combined with optimized geometric Hermite curves that allow to perform a path computation accounting for the heading (i.e. the SE(3) pose) of both the start and target points. Although developed and tested only for an preoperative neurosurgical scenario, also AHFT is suitable for GPU parallelization for a rapid intraoperative replanning.
BB	Bubble Bending [158] is based on the online modification of a predefined path. It assumes the path behaves like an elastic band, deforming to accommodate external forces and reacting with internal forces to keep its original shape.
APF	Artificial Potential Field algorithm [144] defines a potential function to free space and simulates the value of as a particle reacting to the force due to the potential field. The potential function is composed of an attractive force and a repulsive force, which represent the effect from the target and obstacles, respectively.
PRM	A probabilistic roadmap is a network graph of possible paths in a given map based on free and occupied spaces. The basic idea of PRM is to take random samples from the robot's configuration space, checking if they are in free space, and use a local planner to attempt to connect these configurations to other nearby configurations. The start and end configurations are added, and a graph search algorithm is applied to the resulting graph to determine a path between the start and target configurations.
SRM	Stochastic Motion Road Map is defined by initially sampling the configuration states in a random manner to build a collision-free connectivity map. Then, for each sampled state i , the planner computes the optimal action to be performed so that to maximize the probability of reaching the target being i the current needle state. A sequence of optimal control actions that maximize the probability to reach the target without colliding with obstacles is computed.
MS	A moduli space parametrizes a family of geometric paths such that points in moduli space represent geometric paths and nearby points represents objects with similar structure. Continuous motion in moduli space corresponds to a smooth deformation of the geometric object and constraints define hypersurfaces.
Mathematical-based methods	

2.5. RESULTS

MDP	In mathematics, a Markov decision process is a discrete-time stochastic control process. It provides a mathematical framework for modelling decision-making in situations where outcomes are partly casual and partly under the supervision of a decision-maker. Markov Decision Process (MDP)s are useful for studying optimization problems solved via dynamic programming.
DP	The basic concept of dynamic programming is to divide a problem into several related subproblems. A subproblem can in turn be subdivided into similar subproblems and so on until the subproblem is of unit size. In route planning, the problem solved by Dynamic Programming (DP) is to calculate the distance of the destination from all points on the map. The distance to the next points (problem) is calculated based on the pre-calculated distance to the nearest points (subproblem). Thus the entire problem is divided into subproblems, with a unitary subproblem such as the distance of the target from itself being zero.
ihDP	infinite horizon dynamic programming problems are characterized by the fact that the number of stage is infinite.
GBO	Gradient-based optimization method iteratively search a minimum of a dimensional target function. The target function is thereby approximated by a terminated Taylor series expansion around : The actual optimization is performed iteratively.
Penalty	The Penalty Function formulate the planning problem as a constrained nonlinear optimization problem that is locally minimized using a penalty method that converts the formulation to a sequence of unconstrained optimization problems.
BA	The Bresenham algorithm is a line-drawing algorithm that determines the points in an n-dimensional array that should be selected to form a close approximation to a direct line between two points. After generating a possible trajectory with a low cost, the Bresenham algorithm is used to determine the set of voxels, which are penetrated by the trajectory [153].
IK	Inverse kinematics is the mathematical process of computing the variable parameters of the joints required to position the end of a kinematic chain, such as the robotic manipulation or the skeleton of an animation character, in a given position and orientation relative to the start of the chain. IK transforms the motion plan into joint actuator trajectories for the robot.
arc	arc interpolation are arc going through given data points and satisfying certain continuity conditions.
Spline	Splines are piecewise polynomials going through given data points and satisfying certain continuity conditions.
Learning-based methods	
GA	Holland [118] firstly introduced Genetic Algorithm, and now it is the most popular population-based optimization method. The basic version of GA defines a cost function to evaluate potential resolutions. Then a partially random crossover operator takes two parents from the population set and recombines them in some manner. The mutation operator tries to change the solutions and tries to obtain a valid solution to escape local optimality.
ACO	Animals such as ants may be able to establish the shortest route from their colony to the food source and back home through group cooperation, researchers imitate the procedure and have proposed the Ant Colony Optimisation method. Ant Colony Optimization (ACO) introduces two basic ideas, which are "track intensity" and "visibility" to form the transition probability that ultimately decides which route to take, thus to formulate the shortest path. ACO aims to find the best route by evaluating the density at each step. The process works flexibly in dynamic environments, needing only to change the representation of "track intensity" of a certain edge.
PSO	Particle Swarm Optimization is a computational method that optimises a problem by iteratively trying to improve a potential solution with respect to a given quality measure. It solves a problem by having a population of candidate solutions, here called particles, and moving these particles in the search space according to a simple mathematical formula about particle position and velocity.
RL	Reinforcement learning is probably the most complex learning system, which requires that the machine be equipped with systems and tools that can improve one's learning and, above all, understand the characteristics of the surrounding environment. Reinforcement learning algorithms aim to determine the ideal behavior within a specific context based on simple reward feedback on their actions.
DQN	Deep Q-learning Network replaces the normal Q-table with a NN. Rather than mapping a state-action pair to a Q-value, a NN maps input states to pairs (action, Q-value). One of the interesting things about Deep Q-learning Network (DQN) is that the training process uses 2 neural networks. These networks have the same structure but different weights. At each step, the weights of the main network are copied to the target network. Using both networks leads to greater stability in the learning process and helps the algorithm to learn more efficiently.
LfD	In the context of robotics and automation, learning from demonstration is the paradigm in which robots acquire new expertise by learning to imitate an expert.
dDQN	Dueling Deep Q-learning is an algorithm that adds an advantage function, which is a relation function between a learning network and Q function as well as value function in the DQN reinforcement learning method [272]
GAIL	Generative Adversarial Imitation Learning is an inversive reinforcement learning algorithm. It is based on Generative Adversarial Networks (GANs). GAIL could be defined as a model-free imitation learning algorithm.

FFNN	A Feed Forward Neural Network is an artificial neural network in which the connections between nodes does not form a cycle. The feed forward model is the most basic form of NN, as information is only elaborated in one direction. While data can pass through multiple hidden nodes, it always moves in one direction and never backwards.
CNN	The convolutional neural network consists of multiple elements, such as convolution layers, clustering layers and fully connected layers, and is designed to autonomously and adaptively learn spatial feature hierarchies through a backpropagation algorithm.
Reasoning-based methods	
ASP	with Answer Set Programming, computational problems in a large variety of scenarios can be explicated by means of simple and elegant logic programs consisting of a set of rules; solutions to a problem instance are then found by computing the semantics of such programs combined with the representation, usually expressed using factual rules, of the instance at hand.

2.5.1 Graph-based methods

As observable in Table 2.4, different works, in the context of path planning for steerable needles, exploit graph-based methods. As schematized in Figure 2.9, these algorithms are Dijkstra, A-star, Nearest Neighbor Graph (NNG), Breadth First Search (BFS) and Heuristic. The description of BFS is also included in this Section because it is a method that falls into this category but, in the context of path planning for steerable needles, it is used in conjunction with other algorithms. Therefore it is discussed in the multifusion-based methods, Section 2.5.6.

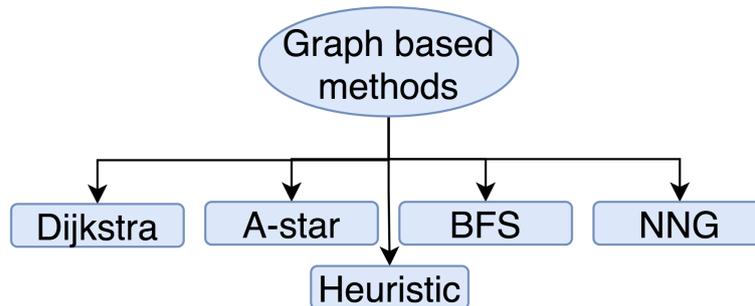


Figure 2.9: Graph-based methods. Specific algorithms of graph-based methods, applied in the context of path planning for steerable needles, are illustrated

Dijkstra

Dijkstra algorithm [62] is an algorithm for finding the shortest paths between nodes in a graph, which is also called as Shortest Path First (SPF) algorithm. In each iteration, it chooses the node with the minimal cost to expand the tree. In [230] and [67], an SPF algorithm is implemented for a guidewire in endovascular interventional procedures. Simulation results indicate that the proposed algorithm provides accurate paths of guidewires in a vessel phantom. The exploration strategy of Dijkstra is based on a cost function, which aims to find the minimal cost path. The performance of its solution is better compared to BFS.

A-star

As an extension of the Dijkstra algorithm, A-star achieves better performance by considering heuristics to guide its search in [110]. In [50], a path planning algorithm based on A-star search for micro-robot in the cerebrovascular system is proposed. Simulations show that the proposed method is able to identify the shortest path automatically on Magnetic Resonance Angiography (MRA) image.

Nearest neighbor graph

NNG algorithm, also called LPA*, is an incremental heuristic search algorithm based on A-star, which is first proposed in [148]. NNG can adapt to dynamic changes in the graph without recalculating the entire graph, which could be used for efficient path re-planning [191].

Best-First search

The BFS [60] explores a graph by expanding the most promising node chosen according to a specified rule. Using a heuristic that attempts to predict how close the end of a path is to a solution, the paths judged to be closer to a solution are extended first. It can be advantageous in real-time scenarios, but most of the path planning methods for MIS are pre-operative planning, where optimality and safety are considered the essential objectives.

Heuristic

In [273], a heuristic-based search algorithm is introduced to find an optimal path of the needle for neurosurgical tumor ablation, and a better performance than BFS and Dijkstra is shown.

Table 2.4: *Graph Based methods*

Author, Year	Clinical Aim	Pathology, Anatomy	Method	Query	Auto/ Semi/ Intra	Risk	Validation	Needle, Sensor	Needle Kinematic	Data	Dataset	Tool	Env	Tissue modelling	Benchmark measure	Results	
[230] Schafer, 2007	vascular surgery (stent)	vascular disease, blood vessels	Dijkstra	single	yes	intra	no	in-vitro	guidewire, no sensor	$d(0.4mm)$	CTA, 3D model	1 image, 1 scene	phantom	dynamic	yes (silicon)	time[sec]	3
[67] Egger, 2007	vascular surgery (stents)	stenosis, blood vessels	Dijkstra	single	yes	pre	no	in-silico	guidewire, no sensor	-	CT, 3D model	1 image, 1 scene	MeVisLab	static	no	length[mm], time[sec]	96.54, 1.05
[50] Chang, 2018	DD	-, blood vessels	A-star	single	yes	pre	no	in-silico	micro/nano robots, no sensor	-	MRA	2 images	-	static	no	time[sec]	3253
[191] Niyaz, 2018	KN	-, brain	NNG	single	yes	intra	no	in-silico ¹ , ex-vivo ²	concentric tube needle	-	3D model	1 scene	-, phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (3D printed)	tError ¹ [mm], tError ² [mm]	0.0009, 0.0076
[273] Wankhede, 2019	tumor ablation	cancer, brain	heuristic	single	yes	intra	no	in-silico	-, -	-	3D model	1 scene	MATLAB	static	no	time[sec]	40

2.5.2 Sampling-based methods

As observable in Table 2.5, different works, in the context of path planning for steerable needles, exploit sampling-based methods. As schematized in Figure 2.10, these algorithms are RRT and its derivatives, which are defined as RRT-connect, Reachability Guided Rapidly-exploring Random Tree (RG-RRT), arc-RRT, RRT-star and Batch Informed Trees star (BIT-Star). Similarly, Artificial Potential Field (APF), Moduli Space (MS), Adaptive Fractal Tree (AFT) and its evolution AHFT. Finally, also PRM and Stochastic Motion Roadmap (SMR) are discussed. The description of PRM is also included in this Section because it is a method that falls into this category but, in the

context of path planning for steerable needles, it is used in conjunction with other algorithms. Therefore discussed in the multifusion-based methods, Section 2.5.6.

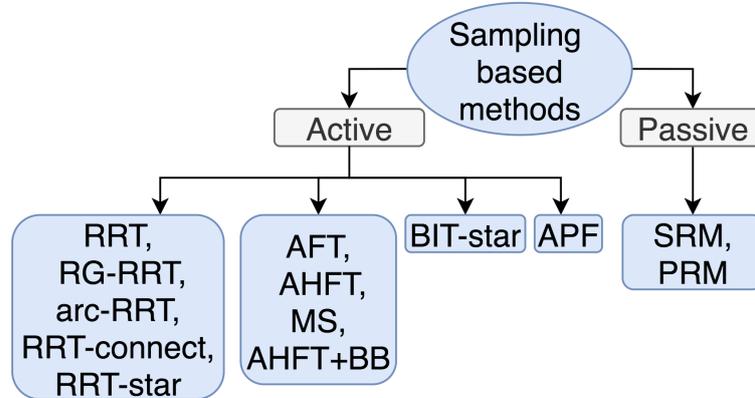


Figure 2.10: Sampling-based methods. Specific algorithms of sampling-based methods, applied in the context of path planning for steerable needles, are illustrated. These algorithms are divided into two more detailed parts: active and passive. Active means algorithms such as RRT that can obtain a feasible path to the target with its own processing procedure, choosing at each step a point in the space. Passive means algorithms such as PRM that first generate a road map from the beginning to the goal, then a search algorithm is used where the solution is chosen from all possible ones on the map

Rapidly-exploring random tree

The first implementation of an RRT path planner for steerable needles in 3D with obstacles was proposed by Xu *et al.* [282] for a bevel-tip needle. The planner randomly samples the space of control inputs obtaining a trade-off between exploration complexity and completeness. The method is also tested to define the target point, a feasible entry point inside a user-defined entry area, starting the search from the target rather than from the start point in a back chaining fashion. The solution was tested in a simulated undeformable environment with ideal spherical obstacles.

In [283], Xu *et al.* proposed a multi query planner for prostate cancer based on an RRT algorithm for a pre-operative static environment. This work was later extended from Van *et al.* [265] with an RRT based intra-operative path planner.

In [266], Vancaberg *et al.* proposed a solution based on an RRT for breast cancer biopsy. It was performed with a bevel tip needle in a 3D dynamic environment with tissue deformation modelled with FEM.

In [202], Patil *et al.* proposed a solution for bevel-tip needles in 2D deformable environments by accounting for uncertainty in deformation models, noisy sensing data and unpredictable actuation through a simulator of deformable environments. Firstly, an RRT algorithm is used to generate a set of candidate motion plans. Then a high quality plan with the highest estimated probability of successfully avoiding obstacles and reaching the goal region is selected among them. This method was verified in 2D simulations, demonstrating a computational time in the order of minutes.

Sun *et al.* in [250] proposed a high-frequency planner based on multiple independent RRTs executed in parallel at every insertion step from the current needle tip position to

the target point. This allows computing, at every time step, a motion plan that asymptotically approaches the global optimal. At each iteration step, the best plan is estimated according to two optimization objectives: the probability of successfully reaching the target without colliding with obstacles and minimising the path length. This plan is opportunely adjusted considering sensory information and linear feedback control. Uncertainty deriving from errors in needle kinematics, sensing noise and obstacles motion is propagated over the next insertion steps to estimate the path uncertainty from the start to the goal position. The feasibility of the proposed solution was tested in simulation in a liver biopsy scenario, where the vasculature represents the sensitive anatomical obstacle, using a duty-cycle bevel-tip needle as a case study. Hoelscher *et al.* presented a RRT-based planning method aiming to accurately reaching targets in most regions of the lung [116]. Unlike typical forward planning approaches, research developed in [116] introduces a backward planning approach that starts at the target and aims to find a piercing site. The reported results show that the backward approach can achieve a higher success rate and higher efficiency.

RRT-connect

Fausser *et al.* [78] proposed a method for the planning problem in $SE(3)=R^3 \times SO(3)$ (c-space in 3D Cartesian space) using an RRT variants: the RRT-Connect [151]. The RRT-Connect is a structure that incrementally expands two RRTs rooted at the start and the target configurations. These two RRTs explore the space advancing towards each other with a simple greedy heuristic until they connect. This method (k-B-RRT-Connect) corresponds to an extension of [201] into RRT-Connect, where the requirement of matching the initial and target needle pose is solved as a Dubins path problem.

RG-RRT and arc-RRT

Patil *et al.* [201] proposed a solution for duty-cycling steerable needles able to plan and control the needle motion in a closed-loop fashion, guaranteeing obstacles avoidance and uncertainties compensation. Confining the search on the subset of point in the workspace that meets the needle's kinematic constraints, this solution allows a path replanning suitable for online procedures ensuring clinically acceptable error of less than 3mm, assessed through experiments in ex-vivo and tissue samples. In this work, the planning method used was the RG-RRT, originally proposed in [200]. RG-RRT are a combination of RRT and a reachability-guided sampling heuristic that construct the path through a sequential connection of arcs with variable curvature bounded within the maximum curvature achievable by the needle. A bespoke distance metric overlooks points of the configuration space that are not reachable from a given state, considering the needle's maximum curvature. In [200] the authors express the planning objective in terms of minimizing the insertion length and maximizing the clearance from obstacles. RG-RRT was also at the base of the work of Vrooijink *et al.* in [267], the needle tip pose is used by a RRT-based motion planner to compute a feasible needle path to the target. The motion planner is sufficiently fast such that re-planning can be performed repeatedly in a closed-loop manner, under a needle insertion speed of $3mm/s$. This enables the system to correct for perturbations in needle motion and movement in obstacle and target locations. The path re-planning algorithm was tested in a soft-tissue phantom. While Caborni *et al.* in [45] proposed a 2D implementation of the RG-RRT

with applications in the neurosurgical field.

The idea of connecting arcs with bounded curvature is at the base of the geometric-based arc planner (arc-RRT) proposed by Bernardes *et al.* for 2D intraoperative path planning in [33] with a method that demonstrated robustness in coping with modelling uncertainties, tissue deformation and inhumanities. Performance has been later enhanced in [32, 34] by implementing an arc-RRT where sampling is performed in the control inputs rather than in the configuration space.

In [289], Zhao *et al.* proposed a fast computation of RG-RRT that features a greedy heuristic strategy based on the depth-first search. This search approach stresses the target's search to encourage the connection through a linear path rather than via arcs of bounded curvature. The best solution is determined on the basis of a cost function that fosters, in addition to paths length and distance from obstacles, also both a reduced magnitude and a limited variation in the path curvature. The solution was later extended in a dynamic motion planner that addresses the uncertainty related to the movement of obstacles and target [290]. The method was developed for a novel flexible tool composed of a bevel-tip needle inserted into a flexible cannula, allowing the needle to freely rotate without torsional friction between the needle shaft and the tissue fostering a more precise control of the needle tip orientation. The planner features solutions to address the problem of planning convergence and large path detour that may arise in the presence of target and obstacles unpredicted motions. When tested in a simulated scenario similar to [282] resulted in being robust and suitable for real-time replanning.

In [152] Kuntz *et al.* developed a hybrid approach to planning the motion of a multi-lumen transoral lung access system designed to perform biopsies in the lung. The system is composed of three serial stages: a bronchoscope to access the bronchial tubes, a concentric tube to enter the lung parenchyma, and a bevel-tip needle to steer in the parenchyma and reach the nodule avoiding lung vessels. The motion plan consists of a sequence of collision-free configurations of the multi-lumen system and associated control inputs. An initial sample-based approach defines the path for the first stage to reach the desired position in the bronchial tubes as a sequence of piece-wise linear curves. The second stage is then actuated, and a mechanic-based kinematic model [227] is used to infer the tip frame of the concentric tube robot when it touches the parenchyma. The third stage then follows a path in the parenchyma estimated through the RRT approach proposed in [201]. The system was tested in a simulated lung biopsy scenario. The method has been successively extended [87]. From lung CT data, a cost map is introduced encoding the costs associated with traversing the specific voxel determined as the probability of for that voxel to be part of the lung vessel tree. As time allows, the planner estimates more paths minimizing the accumulated cost in an anytime manner. A fast and robust two-step path planning approach is adapted in [154] for a US-guided flexible needle, where RG-RRT is considered as a base, followed by an in-depth path analysis via Kalman filter by accounting for uncertainties. This proof-of-concept is validated in phantoms and ex-vivo tissue, and the promising results show that a low targeting error is achieved.

RRT-star and BIT-star

The Rapidly-exploring Random Tree Star (RRT-star) [138] represents an “anytime” variation of the original RRT: it quickly identifies an initial feasible plan, then, as the

plan execution process, it improves the plan toward an optimal solution. This is obtained by rewiring the RRT by removing redundant edges and keeping the shortest or minimum-cost path. As with RRT, an initial feasible solution is found quickly, but, in addition, RRT-star almost surely converges to an optimal solution.

In [119], an RRT-star based path planner is proposed for a magnetically guided flexible needle for DBS [205] aiming to minimize the insertion length while maximizing the obstacle clearance.

In [79], Favaro *et al.* proposed a solution based on an BIT-Star approach that runs within a dynamic subset of the original configuration space. This subset is shaped as an ellipsoidal volume, heuristically defined so that to change according to the planning phase in view of confining the search only where better solutions can lie. This approach reduces the BIT-Star complexity and computational time. Path length, curvature and obstacle clearance are optimized according to a cost function, while control errors are included in the planning phase, increasing the path uncertainty over the insertion length.

Artificial potential field

APF algorithm [144] defines a potential function to free space and simulates the vehicle as a particle reacting to force due to the potential field. The potential function is composed of an attractive force and a repulsive force, which represent the effect of the target and obstacles, respectively.

A potential-field-based path planning technique is used in [63] for needle insertion into deformable tissue. A discrete potential field algorithm based on 3D anatomical structures is proposed in [166] to plan the needle path in minimally invasive surgery with an in-vitro validation on a gelatin phantom. A potential-field-based path planning in the presence of soft tissue deformation based on bio-heat transfer is focused on in [24]. Girerd *et al.* [96] use a 3D point cloud representation for tubular structure and compute a repulsive force to ensure that the concentric tube needle tip remains inside the contour.

Adaptive fractal trees

Parallel path computation is also used in the AFT proposed by Liu *et al.* in [172] for a PBN. This method is an evolution of a precedent work [173] proposed by the same author based on MS algorithm and takes advantage of the fractal theory and the architecture of Graphics Processing Unit (GPU) to parallelize the planning process and enhance the computation performance and achieve online replanning, as demonstrated with simulated 3D liver needle insertions. AFT are the basis for the AHFT later proposed by Pinzi *et al.* in [210] where the fractal structure was combined with optimized geometric Hermite curves that allow performing a path computation accounting for the heading (i.e. the SE(3) pose) of both the start and target points. Although developed and tested only for a preoperative neurosurgical scenario, also AHFT is suitable for GPU parallelization for a rapid intraoperative replanning. This work is extended in [209] via a combination between AHFT and Bubble Bending (BB) for fast replanning, taking into account the environment changes such as tissue deformation. Furthermore, the AHFT is also applied into Laser-induced Thermoablation (LITT) in [211] and states that the steerability of needles has the potential to improve LITT procedure efficacy.

Stochastic motion roadmap

SMR is defined by initially sampling the configuration states in a random manner to build a collision-free connectivity map. Then, for each sampled state i , the planner computes the optimal action to be performed so that to maximize the probability of reaching the target being i the current needle state. A sequence of optimal control actions that maximize the probability to reach the target without colliding with obstacles is computed. The same SMR planned pre-operatively is used intraoperatively for fast path re-computation. This approach tends to foster lengthy but safer insertion paths. The method, originally tested in randomized simulation, was later tested in combination with an integrated needle-steering system composed of a needle pose estimator, a torsion compensator and a planar controller (to guarantee the needle to move on the ideal insertion plane) for the evaluation of its performance in a brachytherapy scenario with artificial and ex-vivo tissues [221]. In [18], Alterovitz *et al.* proposed a 2D motion planning framework that addresses the uncertainty in the needle motion. This is included by considering a stochastic motion model of the needle, which represents the probability of moving from one state to another as a result of a specific control action.

Table 2.5: Sampling Based methods

Author, Year	Clinical Aim	Pathology, Anatomy	Method	Query	Auto/ Semi	Pre/ Intra	Risk	Validation	Needle, Sensor	Needle Kinematic	Data	Dataset	Tool	Env	Tissue modelling	Benchmark measure	Results
[63] Di Maio, 2005	anaesthesia, biopsy, brachytherapy	-	APF	single	yes	intra	no	in-silico ¹ , in-vitro ²	- , no sensor	-	-	-	- ¹ , phantom ²	dynamic	yes ¹ (FEM), yes ² (gelatin)	tError[mm] ¹ < 2.5, tError[mm] ² < 1	
[282] Xu, 2008	MIS	- prostate	RRT	single	yes	pre	no	in-silico	bevel tip, no sensor	-	3D model	1 scene	-	static	no	SR[%], time[sec]	100, 621.4
[18] Alterovitz, 2007	anaesthesia, biopsy, brachytherapy	-	SMR	single	yes	pre	no	in-silico	bevel tip, no sensor	-	2D map	1 map	-	static	no	SR[%]	87
[283] Xu, 2009	MIS	cancer, prostate	RRT	multi	yes	pre	no	in-silico	bevel tip, no sensor	-	3D model	1 scene	-	static	no	time[sec]	57573
[200] Patil, 2010	cancer therapy	cancer, prostate	RG-RRT	multi	yes	pre	no	in-silico	bevel tip, no sensor	$k_{max}(0.02mm^{-1})$	3D model	3 scenes	-	static	no	SR[%], time[sec]	100, 0.02
[266] Vancam, 2010	biopsy	cancer, breast	RRT	multi	yes	intra	no	in-silico	bevel tip, no sensor	-	3D model	1 scene	VTK	dynamic	yes(FEM)	tError[mm] #p, time[sec]	0.3, 30, 50
[265] Van, 2010	cancer therapy	cancer, prostate	RRT	single	yes	intra	no	in-silico	bevel tip, no sensor	-	3D model	1 scene	-	static	yes(noise)	tError[cm], length[cm]	0.24, 11.25
[221] Reed, 2011	brachytherapy, tumor ablation	cancer, liver	SMR	multi	yes	intra	no	in-silico ^{1,2} , ex-vivo ³	bevel tip, camera	$d^1(0.37mm)$, $k_{max}^1(0.016mm^{-1})$, $d^{2,3}(0.58mm)$, $k_{max}^{2,3}(0.018mm^{-1})$	3D model	1 scene	matlab ^{1,2} , goat liver ³	static, dynamic	no ^{1,2} , yes(biological) ³	tError ¹ , tError ²	0.79±0.18, 1.11±0.25
[202] Patil, 2011	MIS	-	RRT	single	yes	intra	no	in-silico	- no sensor	-	2d map	1 map	-	dynamic	yes(FEM)	SR[%]	97
[33] Bernardes, 2011	MIS	-	arc-RRT	single	yes	intra	no	in-silico	duty-cycle, no sensor	$k_{max}(0.16cm^{-1})$	2D USi	1 image	-	dynamic	no	tError[mm], time[msec]	0.2, 0.038
[45] Caborni, 2012	KN	- brain	RG-RRT	multi	yes	intra	yes (risk map)	in-silico ¹ , in-vitro ²	PBN, EM	$d^1(4mm)$, $k_{max}^1(0.025mm^{-1})$, $d^2(4.5mm)$, $k_{max}^2(0.007mm^{-1})$	MRI	-	Matlab ^{1,2} , phantom ²	static ¹ , dynamic ²	no ¹ , yes(gelatin) ²	time ¹ [sec], SR ¹ [%], tError ² [mm]	27.50, 100, 1.15±0.63
[34] Bernardes, 2013	MIS	-	arc-RRT	single	yes	intra	no	in-silico ¹ , in-vitro ²	duty-cycle, camera	$d(0.508mm)$, $k_{max}(20^\circ)$	-	-	- , phantom ²	dynamic ¹ , dynamic ²	no ¹ , yes ² (gelatin)	tError ¹ [mm], SR ¹ [%], length ¹ [cm], time ¹ [ms], tError ² [mm]	0.36, 99.8, 12.23±0.74, 1.9±4.33, 1.3±0.9
[32] Bernardes, 2014	MIS	-	arc-RRT	single	yes	intra	no	in-silico	duty-cycle, no sensor	$k_{max}(0.25cm^{-1})$	3D model	1 scene	-	dynamic	no	SR[%], tError[mm], time[sec]	100, 0.83±0.47, 13.17±1.66
[267] Vrooijink, 2014	surgical therapy	cancer, -	RG-RRT	single	yes	intra	no	in-vitro	bevel tip, ultrasound	$d(0.5mm)$, $k_{max}(30^\circ)$	2D USi	-	phantom	dynamic	yes(gelatin)	tError[mm]	2.16 ± 0.88
[201] Patil, 2014	surgical therapy	cancer, liver	RRT	single	yes	intra	yes	in-vitro ¹ , ex-vivo ²	(prebend) bevel tip, EM	$d^1(0.92mm)$, $k_{max}^1(0.11cm^{-1})$, $d^2(0.88mm)$, $k_{max}^2(0.15cm^{-1})$	3D model ¹ , CT ²	4 scene ¹ , 1 image ²	phantom ¹ , porcine tissue ²	dynamic	yes ¹ (gelatin), yes ¹ (biological)	tError ¹ [mm], tError ² [mm], length ² [mm], time[sec]	1.07±0.59, 2.38±1.02, 15.5, 1
[152] Kunis, 2015	biopsy	cancer, lung	RRT	single	yes	pre	yes	in-silico	concentric tube, no sensor	$k_{max}(0.121m^{-1})$	CT, 3D model	1 image, 1 scene	-	static	no	SR[%], time[sec]	95, 2
[289] Zhao, 2015	MIS	-	RG-RRT	single	yes	pre	no	in-silico	active canula, -	$k_{max}(0.02mm^{-1})$	3D model	1 scene	Matlab	static	no	time[sec], length[mm]	1.3±0.0001, 215.54±6.14

2.5. RESULTS

[290] Zhao, 2016	MIS	-	RG-RRT	single	yes	intra	no	in-silico	active can- nula + bevel tip, -	$k_{max}(0.02mm^{-1})$	3D model	1 scene	Matlab	dynamic	yes(noise)	tError[mm], time[sec], length[mm]	1.47±0.37, 3.3±22.1, 1197.1±5.2
[250] Sun, 2015	biopsy	cancer, liver	RRT (with HFR)	single	yes	intra	no	in-silico	bevel tip, imaging/ EM	$k_{max}(cm^{-1})$	3D model	1 scene	-	dynamic	no	SR[%], length[cm]	98, 11.13±0.59
[173] Liu, 2015	biopsy, RSI	DD, cancer, liver	MS	multi	yes	pre	no	in-silico	bevel tip, no sensor	-	CT	1 image	-	dynamic	yes(FEM)	#p, time[sec]	70, 0.01
[172] Liu, 2016	MIS	- liver	AFT	multi	yes	pre	no	in-silico	PBN, no sensor	$k_{max}(0.014mm^{-1})$	CT, 3D model	1 image, 1 scene	Matlab	static	no	tError[mm], time[sec]	1.45±1.19, 5.15±0.048
[281] Xiong, 2016	PNI	- liver	APF	single	semi	pre	-	in-silico	bevel tip, no sensor	-	3D model	1 scene	-	static	no	-	-
[166] Li, 2017	MIS	-	APF	single	yes	intra	no	in-silico ¹ , in-vitro ²	bevel tip, ultrasound	$d(0.5mm)$, $k_{max}(30^\circ)$, $l(200mm)$	3D model ¹ , 2D US ²	1 scene ¹ , 1 image ²	- phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (gelatin)	tError ¹ [mm], tError ² [mm]	0.29, 1.15±0.56
[78] Fauser, 2018	temporal bone surgery	brain (Cochlea ¹ , retro-labyrinthine ² , semicircular canal ³)	RRT-connect	multi	yes	intra	yes	in-silico	bevel tip/ hydraulically driven, no sensor	$d^1(0.5mm)$, $k_{max}^1(0.05cm^{-1})$, $d^{2,3}(1mm)$, $k_{max}^{2,3}(0.05cm^{-1})$	CT, 3D model	40 images, 100 scene	3DSlicer	static	no	# ¹ , # ² , # ³ , time ¹ [sec], time ² [sec], time ³ [sec], SR ¹ , SR ² [%], SR ³ [%]	2635, 760, 4, 131.75, 760, 0.2, 95, 100, 55
[87] Fu, 2018	biopsy	cancer, lung	RRT	single	yes	pre	yes(risk map)	in-silico	bevel tip	$k_{max}(0.121m^{-1})$	CT, 3D model	1 images, 1 scene	3DSlicer	static	no	SR[%]	100
[79] Favaro, 2018	DD	cancer, brain	BIT-star	single	yes	pre	yes	in-silico	PBN, no sensor	$d(1.25mm)$, $k_{max}(0.014mm^{-1})$	MRI, 3D Model	1 image, 1 scene	3DSlicer	static	no	length[mm], d_{min} [mm], d_{avg} [mm], time[sec]	86.2, 1.37, 4.75, 45.1
[119] Hong, 2019	DBS	PD/AD, brain	RRT-star	single	yes	pre	yes	in-silico	magnetically controlled needle, no sensor	$k_{max}(0.025mm^{-1})$	MRI, 3D model	1 image, 1 scene	CAD	static	no	SR[%], d_{min} [mm], d_{avg} [mm], time[sec]	100, 2, 4, 1.4
[210] Pinzi, 2019	DD	cancer, brain	AHFT	multi	yes	pre	no	in-silico	PBN, no sensor	-	MRI	1 image	3DSlicer, MATLAB	static	no	SR[%], time[sec]	94.23, 22
[24] Bahwini, 2019	biopsy, ablation brachytherapy	cancer, liver	APF	single	yes	intra	no	in-silico	-	-	3D models	2 scene	-	dynamic	yes(FEM)	-	-
[209] Pinzi, 2021	DBS, DD	PD/cancer, brain	AHFT+ BB	single	yes	intra	yes	in-silico ¹ , in-vitro ²	PBN, EM	$d(1.25mm)$, $k_{max}(0.014mm^{-1})$	MRI	1	Matlab ¹ , phantom ²	Dynamic	yes(FEM) ¹ , yes(gelatin) ²	time ¹ , time ² [sec], SR ¹ , SR ² [%], tError ¹ [mm], tError ² [mm], α_{target}^1 , α_{target}^2	0.53±0.03, 0.51±0.02, 93.6, 100, 0.64±0.46, 1.81±0.51, 3.25±5.23, 5.9±1.42
[211] Pinzi, 2021	LITT	MTLE, brain	AHFT	multi	yes	pre	yes	in-silico	PBN, no sensor	$d(1.25mm)$, $k_{max}(0.014mm^{-1})$	CT, MRI	5 images, 5 images	-	static	no	length[mm], k_{mean}^{-1} [mm]	111±7.5, 0.6±0.03
[154] Lapouge, 2020	biopsy, brachytherapy, DD	-	RG-RRT	single	yes	intra	no	in-silico ¹ , ex-vivo ²	bevel tip, ultrasound	$d(0.45mm)$	US images	3 images	- ¹ , ²	dynamic	no ¹ , yes ² (gelatin)	tError ¹ [mm], tError ² [mm]	1.5±0.9, 1.7±0.8
[96] Girerd, 2020	tumor removal	tumor, MIS	APF	single	yes	intra	no	in-silico	concentric tube, camera	length(300mm), $k_{max}(0.04mm^{-1})$	CT	1 image	-	static	no	-	-
[116] Hoelscher, 2021	biopsy	cancer, lung	RRT	multi	yes	pre	yes (risk map)	in-silico	bevel tip	$d(1mm)$, length(100mm), $k_{max}(0.01mm^{-1})$	CT	3 images	-	static	no	SR[%], time[sec]	92.53±0.37, 2.48

2.5.3 Mathematical-based methods

As observable in Table 2.6, different works, in the context of path planning for steerable needles, exploit mathematical-based methods. As schematized in Figure 2.11, these algorithms are Penalty functions, Inverse Kinematics (IK), DP, Arc interpolation, Spline interpolation and GBO.

Penalty function

The Penalty Function formulates the planning problem as a constrained nonlinear optimization problem that is locally minimized using a penalty method that converts the formulation to a sequence of unconstrained optimization problems. Alterovitz *et al.* proposed a planning algorithm to generate desired needle trajectories within a 2D plane in [15], based on the Penalty Function method. This method was then tested on a phantom in [133], where Kallem *et al.* design and demonstrate a nonlinear image-based observer-controller pair to drive a flexible bevel-tip needle to a desired 2D plane based

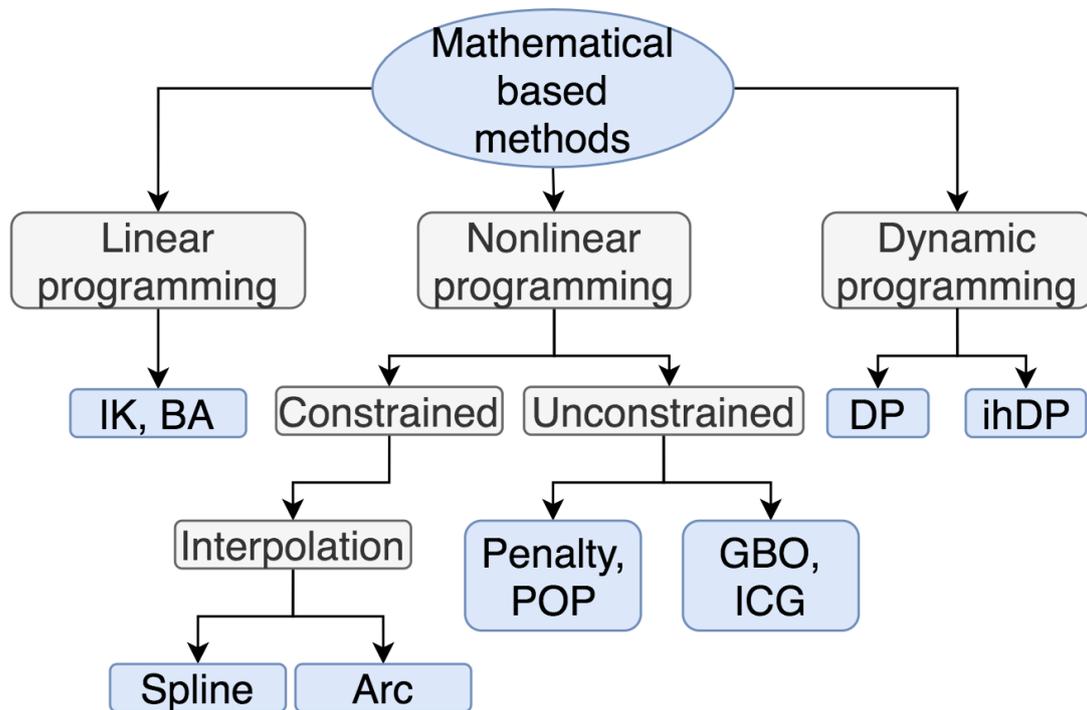


Figure 2.11: Mathematical-based methods. Specific algorithms of mathematical-based methods, applied in the context of path planning for steerable needles, are illustrated. These algorithms are divided into three more detailed parts: linear, nonlinear and dynamic programming. Linear programming refers to an algorithm, such as IK, that can achieve the best result in a mathematical concept whose requirements are expressed by linear relationships. Nonlinear problems arise when the objective or constraints cannot be expressed as linear functions without sacrificing some essential nonlinear feature of the real problem. This category was further divided into the constrained and unconstrained problem. In an unconstrained problem, such as GBO, minima are only found at stationary points. In a constrained problem, such as Spline, minima can be found at nonstationary points. Dynamic Programming means algorithm, such as infinite horizon Dynamic Programming (ihDP), which subdivides a problem into several connected subproblems. A subproblem can, in turn, be subdivided into smaller, similar subproblems and so on until the subproblem is of unitary size.

on Penalty Function.

Inverse kinematics

IK is the mathematical approach to calculating the variable joint parameters required to position the end of a kinematic chain, such as a robot manipulator or the skeleton of an animated character, in a given position and orientation relative to the start of the chain. IK transforms the motion plan into joint actuator trajectories for the robot. If it is assumed that the needle has follow-the-leader motion in the tissue, the path of the needle can be represented by its shape, and hence the path planning can be achieved by solving the inverse kinematics. The first IK based 3D motion planning algorithm was introduced by Park *et al.* [198] and used diffusion of a stochastic differential equation to generate a family of solution paths. The authors also describe several extensions to avoid obstacles. Duindam *et al.* [64, 65] proposed a solution for estimating catheter

pathways totally described in geometric terms and inspired by the Paden-Kahan sub-problem, an explicit solution to inverse kinematics used as an alternative to the implicit expression provided by the Denavit-Hartenberg parameters. The method was tested in a simplified environment and showed a high speed in the path computation but with limited obstacle avoidance capabilities. Glozman *et al.* [99] model the interaction between a flexible needle and the surrounding tissue using virtual springs to compute local deformations with ex-vivo experiments on chicken breast muscle and animal liver. Wang *et al.* in [270] addressed the 3D motion planning problem for steerable needles based on IK and in [269] proposed to adjust the curved path within two parallel lines and then establish the optimum distance between the two parallel lines so that the generated moving path of the needle has the shortest length with the least number of needle rotations. Then they investigate the insertion problem in a deformable environment. The work in [165] presents a dynamic path planning based on needle inverse kinematics, which also considers the soft tissue deformation via FEM modelling. In [52], the path planning and obstacle avoidance is solved via the inverse kinematics based on experimentally verified needle path curvature. Lacking passive needle segment modelling and tissue modelling, the paths are repeatedly re-planned.

Bresenham algorithm

The Bresenham's algorithm (BA) is a line-drawing algorithm that determines the points in an n-dimensional array that should be picked to form a close approximation to a direct line between two points. In [153], a risk map is defined, and a possible trajectory is generated from every vertex of the surface model to the target point, aiming to calculate the best trajectory with the lowest overall risk. After generating a possible trajectory with a low cost, the BA is used to determine the set of voxels, which are penetrated by that trajectory.

Dynamic programming

In mathematics, a Markov decision process is a discrete-time stochastic control process. It provides a mathematical approach for modelling decision making in scenarios where outcomes are partly casual and partly under a decision maker control. MDPs are useful for studying optimization problems solved via dynamic programming. Given a medical image with segmented obstacles and target, Alterovitz *et al.*, in [14], formulate the planning problem as a Markov Decision Process based on an efficient discretization of the state space, model motion uncertainty using probability distributions, and compute optimal steering actions using Dynamic Programming. In [21], Asadian *et al.*, present a new 2D motion planner for steering flexible needles inside relatively rigid tissue. This approach uses a nonholonomic system approach, which models tissue-needle interaction, and formulates the problem as a Markov Decision Process that is solvable using ihDP. In [255] Tan *et al.* proposed a method using MDP for flexible needle navigation that can perform robust path planning and steering under the circumstance of complex tissue-needle interactions. With DP achieves higher accuracy and probability of success in avoiding obstacles under complicated and uncertain tissue-needle interactions.

Arc and spline interpolation

In [199] Park *et al.* adopted a new method using reference splines with the Path-of-Probability (POP) algorithm to path planning of flexible needles with bevel tips and demonstrated how the algorithm could be used for feedback control of flexible needles. In [184] Moreira *et al.* presented a new three-dimensional flexible needle steering system that presented a new flexible three-dimensional needle guidance system that integrates optimal guidance control, an ultrasound-based needle tracking system, a needle deflection model, online needle curvature estimation and offline curvature estimation based on biomechanical properties. The path is calculated by an algorithm that uses splines to find a path that goes from the insertion point to the target location, while avoiding the obstacle. The system is evaluated by experiments in gelatin phantoms and biological tissues. In [291] a path planning method based on a mechanical model is proposed. Zhao *et al.* proposed a method using an equal-radius arc, which can avoid big return in the path to meet the physical characteristics of the flexible needle, reduce large return in the path, and reduce harm on the tissue.

Gradient based optimization

Gradient based optimization strategies iteratively search a minimum of a dimensional target function. The target function is thereby approximated by a terminated Taylor series expansion around: The actual optimization is performed iteratively. In [25] Bano *et al.* presented a gradient method for generating pre-operative paths for a neurosurgical flexible probe. Since the flexible probe is modelled as a nonholonomic system, a deterministic continuous curvature path planning scheme capable of avoiding obstacles is developed for smooth steering of its tip. Multiple paths are generated with the aim to minimize damage to the tissue and the risk to the patient. This method was later validated by Ko *et al.*, in [146] where the performance of a 4-mm diameter flexible needle was tested, in-vitro on a phantom, on curvilinear trajectories generated with the new path planning algorithm, using the closed-loop control strategy.

Table 2.6: *Mathematical Based methods*

Author, Year	Clinical Aim	Pathology, Anatomy	Method	Query	Auto/ Semi/ Intra	Risk	Validation	Needle, Sensor	Needle Kinematic	Data	Dataset	Tool	Env	Tissue modelling	Benchmark measure	Results
[15] Alterovitz, 2005	biopsy, DD, brachytherapy	cancer, prostate	Penalty	single	yes	intra	no	in-silico	bevel tip, no sensor	-	MRI, 3D model	1 image, 1 scene	OpenGL	dynamic	yes(FEM)	time[sec] 60
[198] Park, 2005	MIS	-	IK	multi	yes	pre	no	in-silico	bevel tip, no sensor	$k_{max}(0.05mm^{-1})$	-	-	static	no	-	-
[99] Glozman, 2006	MIS	-	IK	single	yes	intra	no	in-silico ¹ , ex-vivo ²	PBN, force sensor, camera	$d(0.71mm)$	x-ray	2 images	chicken muscle ² , animal liver ²	dynamic	yes(stiffness) ¹ , yes(biological) ²	-
[14] Alterovitz, 2008	brachytherapy	cancer, prostate	MDP (ihDP)	single	yes	intra	no	in-silico	bevel tip, no sensor	$k_{max}^1(0.2mm^{-1})$, $k_{max}^2(0.4mm^{-1})$	USi	1 image	-	dynamic	yes	SR%, time[sec] 73.7, 67-110
[64] Duindam, 2008	MIS	-	IK	multi	yes	pre	yes	in-silico	bevel tip, no sensor	-	3D model	1 scene	-	static	no	time[sec] <1
[133] Kallem, 2009	biopsy, brachytherapy	-	Penalty	single	yes	intra	no	in-silico ¹ , in-vitro ²	bevel tip, no sensor	$d(0.7mm)$, $k_{max}(45^\circ)$	-	-	Matlab ¹ , phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (gelatin)	length ² [cm] 12, SR ² [%], tError ² [m/r] 98.56, 1
[65] Duindam, 2009	MIS	-	IK	single	yes	pre	no	in-silico	-	-	3D model	1 scene	-	static	no	-
[199] Park, 2010	MIS	-	spline (POP)	single	yes	pre	no	in-silico	bevel tip, no sensor	$k_{min}(0, 157cm^{-1})$	3D model	1 scene	Matlab	static	no	time[sec] 1.6
[21] Asadian, 2011	MIS	-	MDP(ihDP)	single	yes	pre	no	in-silico ¹ , in-vitro ²	bevel tip, EM	$d(1.27mm)$, $k_{max}(22^\circ)$	2D map	1 map	Matlab ¹ , phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (gelatine)	time ¹ [sec] 2166.5, tError ² [m/r] 5
[270] Wang, 2011	MIS	-	IK	single	yes	pre	no	in-silico	bevel tip, no sensor	-	3D model	1 scene	-	static	no	length[cm] 29.05, time[sec] 5

[25] Bano, 2012	KN	lesion, brain	GBO	multi	yes	pre	yes (risk map)	in-silico	PBN, no sensor	$d(4mm)$ $k_{max}(0.014m^{-1})$	MRI	1 image	Matlab	static	no	-	-
[146] Ko, 2012	KN	brain disease, brain	GBO	single	yes	intra	yes	in-silico ¹ , in-vitro ²	PBN, EM	$d(4mm)$ $k_{max}(0.014mm^{-1})$	-	-	Labview ¹ , phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (gelatin)	trError ² [m]	$\theta \pm 0.64$
[269] Wang, 2013	MIS	-	IK	single	yes	intra	no	in-silico ¹ , in-vitro ^{2,3}	bevel tip, camera	$d^{1,2}(0.7mm)$, $k_{max}^{1,2}(39.6^\circ)$, $d^3(0.5mm)$ $k_{max}^3(46.2^\circ)$	3D model	1 scene	phantom ^{1,2}	dynamic	yes ¹ (mass- spring), yes ^{2,3} (gelatine)	length ¹ [mm] tError ¹ [mm] length ² [mm] tError ² [mm] length ³ [mm] tError ³ [mm]	147.008, 1.2, 1.3, 1.1, 256.2, 1.8
[184] Moreira, 2015	MIS	-	spline	single	yes	intra	no	in-vitro ¹ , ex-vivo ²	bevel tip, ultrasound	$d(0.5mm)$ $k_{max}(30^\circ)$	USi	3 images	phantom ¹ , chicken tissue ²	dynamic	yes ¹ (gelatin), yes ² (biological)	tError ¹ [mm] tError ² [mm]	0.42 ± 0.17
[255] Tan, 2018	RFA	cancer, liver	MDP(DP)	single	yes	intra	no	in-silico ¹ , in-vitro ²	bevel tip	RITA RFA	CT	1 image	Matlab ¹ , phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (gelatin)	P_2^2 [%], SR ² [%]	68.2, 92.4
[291] Zhao, 2018	biopsy	cancer, liver	arc	single	yes	intra	no	in-vitro	bevel tip, ultrasound	$d(1.2mm)$, $k_{max}(20^\circ)$, length(120mm)	USi, 3D model	1 image, 1 scene	phantom	dynamic	yes(gelatin)	-	-
[165] Li, 2020	MIS	-	IK	single	yes	intra	no	in-vitro	bevel tip, camera	$d(0.58mm)$, $k_{max}(30^\circ)$, length(150mm)	-	1 scene	Matlab, phantom	dynamic	yes(FEM), yes(PVA)	tError[mm] d_{min} [mm]	1.2, 1.16
[52] Chen, 2020	biopsy, brachytherapy, ablation	-	IK	single	yes	intra	no	in-vitro	bevel tip, camera	$d(1.2mm)$, $k_{max}(0.03mm^{-1})$	3D model	1 scene	Matlab	dynamic	yes(gelatine)	tError[mm]	2
[153] Kunz, 2021	keyhole neuro- surgery	tumor, brain	BA	multi	yes	pre	yes (risk map)	in-silico	-	-	CT, MRI	1 images, 1 images	ROS	static	no	time[sec]	12.3 ± 1

2.5.4 Learning-based methods

As observable in Table 2.7, different works, in the context of path planning for steerable needles, exploit learning-based methods. As schematized in Figure 2.12, these algorithms are GA, Particle Swarm Optimization (PSO), ACO, Learning from Demonstration (LfD), DQN, UDQN, dueling Deep Q-learning (dDQN), Convolutional Neural Network (CNN), Generative Adversarial Imitation Learning (GAIL) and Feedforward Neural Networks (FFNN).

GA, PSO and ACO

EA includes GA, PSO and ACO.

EA is a stochastic search approach that imitates natural biological evolution and social behavior. The firstly proposed and now widely implemented Evolutionary method is GA. Holland [118] firstly introduced GA, and now it is the most popular population-based optimization method. The basic version of GA defines a cost function to evaluate the potential solutions. Then a partly random crossover operator takes two parents from the population set and recombines them in some way. The mutation operator tries to modify the solutions and aims to achieve a valid solution in order to escape local optimality. In [22] Path planning is accomplished through a GA, and the efficacy of waterjet steerable needle is tested for different paths. The article’s key finding is that the radius of curvature of the steerable water jet needle can be adjusted by a fixed angle of the water jet tip and varying the speed of the water jet outlet to control the cutting depth. Li *et al.* [168] proposed a fast path planning approach under the steerable catheter curvature constraint via a local GA optimization. The reported results showed the planner’s ability to satisfy the robot curvature constraint while keeping a low computational time cost compared with sampling-based methods. Animals such as ants could manage to establish the shortest path from their colony to the feeding source and back home by group cooperation; researchers mimic the behavior and proposed ACO method. ACO introduces two basic concepts, which are “intensity of trail” and “visibility” to form the transition probability, which at last decides which way to go, thus formulating the shortest path. ACO aims to find the best path by evaluating the density in each step.

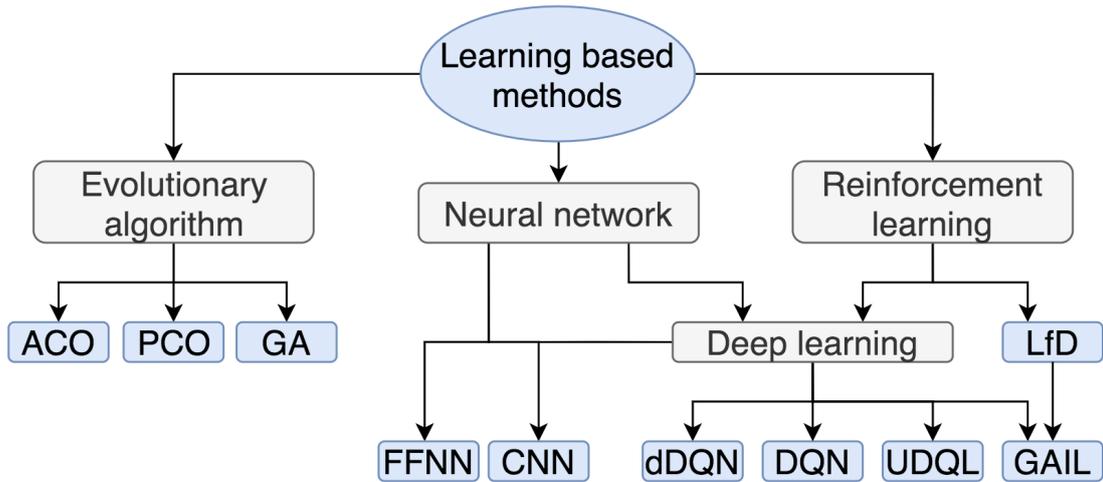


Figure 2.12: Learning-based methods. Specific algorithms of learning-based methods, applied in the context of path planning for steerable needles, are illustrated. These algorithms are divided into three more detailed parts: Evolutionary Algorithm (EA), NN and RL. EA include GA, PSO and Ant Colony Optimization (ACO), EA was proposed to overcome the problem where traditional linear programming and dynamic programming frequently fail to solve NP-hard problems with a large number of variables. NN, such as CNN and FFNN was first introduced for obstacle avoidance as well as navigation and then became widely used and implemented in path planning in various sectors. NN approach aims to generate a dynamic landscape in a neural-like form. RL algorithms, such as LfD and in particular Deep RL algorithms, such as GAIL DQN, dDQN and UDQN received more attention, and there are increasing applications of it in robot path planning

The process runs flexibly in dynamic environments, only needing to change the representation of “intensity of trail” of a certain edge. In [91], Gao *et al.*, proposed an improved ant colony algorithm to plan a vascular optimal three-dimensional path with overall consideration of factors such as catheter diameter, vascular length, diameter as well as the curvature and torsion. PSO is a computational method that optimises a problem by iteratively trying to improve a candidate solution with respect to a given quality measure. It solves a problem by having a population of candidate solutions, here called particles, and moving these particles in the search space according to a simple mathematical formula about particle position and velocity. In [46], Cai *et al.* proposed PSO approach for percutaneous puncture with a flexible needle to guarantee accuracy and avoid obstacles in the puncture process. The proposed method was validated in-vitro on a phantom. A PSO-based path planning approach is also developed in [256] for a bevel tip flexible needle. Nevertheless, the proposed work is only validated in an ideal simulation scene composing of regular sphere obstacles.

Reinforcement learning

RL is an exciting field of Machine Learning that’s attracting a lot of attention and popularity. It requires that the machine be equipped with systems and tools that can improve one’s learning and, above all, understand the characteristics of the surrounding environment. RL algorithms aim to determine the ideal behavior within a specific context based on simple reward feedback on their actions. In [53], Chi *et al.* proposed a pilot study of the implementation of trajectory optimisation to robot-assisted catheterisation

in simulated environments. The initial robot control policies were set by expert demonstrations. RL, based on a LfD approach, was implemented to optimize the policies for dynamic flow simulations, adaptation to different vascular phantoms and catheterization tasks. The proposed robotic approach interacts with vascular phantoms with spatial feedback from a 3D tracking system. The results show a significant refinement of catheter paths by the proposed approach, resulting in shorter overall lengths and fewer contact forces, which can potentially reduce risks in endothelial wall damages, embolization, and stroke. One of the core concepts in RL is the Deep Q-Learning algorithm. DQN is indeed one of the first algorithm showing success while applying deep learning to RL. In [159], proposed by Lee *et al.*, a DQN algorithm is utilized to learn the control policy for flexible needle steering with needle-tissue interaction. In the simulation, the agent (needle) can be controlled with 2 degrees of freedom (bevel direction rotation and insertion) and received negative rewards when it collides with obstacles, goes out of range, or exceeds a predefined number of rotations. During the training, the agent demonstrates the accuracy and efficiency of the learned policy through feedback scores in every episode. In [242]², obtained 2D and 3D environments from MRI images processing and implemented a DQN algorithm to create a path planning model, able to generalize on different patients anatomies. The curvilinear trajectories obtained are compared to the ones obtained by A-star and RRT-star algorithms for a neurosurgical environment. The proposed method achieves state-of-the-art performances in terms of obstacle avoidance, trajectory smoothness and computational time, proving this algorithm as a valid planning method. In [254], Tan *et al.* proposed an approach for flexible needle insertion for biopsy and radio-frequency ablation. The proposed a new path planning framework using multi-goal DQN to address difficulties in uncertain needle-tissue interactions and improve the robustness of the robot-assisted insertion process. The proposed framework utilizes a new algorithm called Universal Distributional Q-learning (UDQL) to learn a stable driving policy and perform risk management by visualising the learned Q-value distribution. To further improve robustness, the universal value function approximation is leveraged in the UDQL training process to further maximise generalisation and link to diagnosis by adapting fast re-planning and transfer learning. In [286], You *et al.* implemented a RL method for automated control of a catheter by a robot. This study aims to show that such a robot can learn to manipulate a catheter to reach a target in a simulated environment and subsequently control a catheter in an actual environment. In this learning method, a dDQN method based on DQN, is used together with a CNN to learn the image data. In [239]³ a learning-by-imitation method is applied in which the agent learns to perform the desired trajectory through a set of demonstrations given by an experienced surgeon. The approach chosen to accomplish this task is the exploitation of GAIL which is a technique that uses demonstrations by experts and learns both the politics and the reward function of the unknown environment. This learning system is applied to a 3D brain surgery simulator that has been developed with *Unity3D* and integrated with *ML-Agents Toolkit* [131].

²Segato, A., Sestini, L., Castellano, A., and De Momi, E. (2020, May). GA3C reinforcement learning for surgical steerable catheter path planning. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2429-2435). IEEE.

³Segato, A., Irene, T., and De Momi, E. (2020). 3D Neurosurgical Simulator for Training Robotic Steerable Catheter Agents Using Generative Adversarial Imitation Learning. In 2020 International Conference on Robotics and Automation (ICRA). France.

Table 2.7: Learning Based methods

Author, Year	Clinical Aim	Pathology, Anatomy	Method	Query	Auto/Semi	Pre/Intra	Risk	Validation	Needle, Sensor	Needle Kinematic	Data	Dataset	Tool	Env	Tissue modelling	Benchmark measure	Results
[91] Gao, 2015	vascular surgery (stent)	vascular disease, blood vessel (leg)	ACO	single	yes	intra	no	in-vitro	-, EM	$d^1(0.2mm)$, $d^2(0.5mm)$	CT, 3D model	1 image, 1 scene	VTK, phantom	dynamic	yes	time ¹ [sec], time ² [sec]	10.53, 3.88
[53] Chi, 2018	vascular surgery (stent)	Standard ¹ , Aneurysm ² , Stenosis ³ , blood vessel	RL(LfD)	single	semi	intra	no	in-silico, in-vitro	pre curved stylet, camera/EM	-	CT, 3D model	1 image, 3 scene	matlab, phantom	static, dynamic	no, yes(gelatin)	RMSE ¹ [mm], RMSE ² [mm], RMSE ³ [mm], length ¹ [mm], length ² [mm], length ³ [mm]	2.98, 211.6, 3.16, 264.1, 2.64, 301.3
[286] You, 2019	DD, RFA	arrhythmia, heart	DRL (CNN+dQN)	single	yes	intra	yes	in-silico ¹ , in-vitro ²	tendon actuated active tip, EM	-	3D model	1 scene	Unity3D ¹ , phantom ²	static ¹ , dynamic ²	no ¹ , yes ² (silicon)	SR ¹ [%], d_{avg}^1 [mm], SR ² [%], d_{avg}^2 [mm]	96, 4.44±0.72, 87, 4.70±1.59
[159] Lee, 2019	MIS	-	RL(DQN)	single	yes	pre	yes	in-silico	bevel tip, no sensor	$d(0.83mm)$	CT, 2D map	1 map, 1 model	-	static	no	SR[%]	93.6
[46] Cai, 2020	MIS	-	PSO	single	yes	intra	no	in-silico ¹ , in-vitro ²	bevel tip, digital camera	$d(0.6mm)$	3D Model	1 scene	Matlab ¹ , phantom ²	static ¹ , dynamic ²	no ¹ , yes(gelatin) ²	tError ¹ [mm], tError ² [mm]	1.36, <4
[242] Segato, 2020	DBS, DD	PD/cancer, brain	RL(DQN)	single	yes	pre	yes	in-silico	PBN	$d(2.5mm)$, $k_{max}(0.014mm^{-1})$	MRI, CT, 3D model	7 images, 7 images, 7 scenes	3DSlicer	static	no	length[mm], d_{min} [mm], d_{avg} [mm], time[sec]	0.28, 0.5, 0.78, 0.087
[22] Babaias, 2020	biopsy, brachytherapy, breast surgery	-, lung/breast	GA	single	yes	intra	no	in-silico ¹ , in-vitro ²	fracture-direct waterjet camera	$d(1.6mm)$	-	-	SEBS ¹ , phantom ²	dynamic	yes ¹ , yes(gelatin) ²	-	-
[239] Segato, 2020	DBS, DD	PD/cancer, brain	GAIL	single	yes	pre	yes	in-silico	PBN	$d(2.5mm)$, $k_{max}(0.014mm^{-1})$	MRI, CT, 3D model	1 images, 1 images, 1 scenes	Unity3D	static	no	length[mm], d_{min} [mm], d_{avg} [mm], time[sec], k_{mean} [°]	55.65±23.1, 0.02±0.03, 11.78±4.57, 2±0.73, 14
[254] Tan, 2020	biopsy, RFA	cancer, liver	RL(UDQL)	single	yes	intra	yes	in-silico, in-vitro	flexible, camera	$d(2.304mm)$	CT, 3D model	1 image, 1 scene	-, phantom	static, dynamic	no, yes(gelatin)	SR[%], F_a [%]	86.4, 90.5
[168] Li, 2021	vascular surgery (stent)	stenosis, aneurysms, blood vessels	GA	single	yes	pre	no	in-silico	-	-	2D images, 3D models	8 images, 5 scenes	-	static	no	time[sec], SR[%], d_{min} [mm]	0.191±0.102, 1~5.5, 100
[256] Tan, 2021	biopsy, anesthesia, DD	-	PSO	single	yes	pre	no	in-silico	bevel tip	$k_{max}(0.02mm^{-1})$	3D model	2 scenes	-	static	no	tError[mm]	<1

2.5.5 Reasoning-based methods

As observable in Table 2.8 few, but recent works, in the context of path planning for steerable needle, exploit reasoning-based methods. As schematized in Figure 2.13 this algorithm is ASP. ASP allows to model in a purely declarative fashion the environment and the steerable needle which moves in it using a set of logic rules, hence creating a flexible methodology which can be customized by surgeons depending on the single patient.

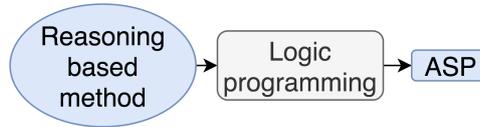


Figure 2.13: Reasoning-based methods. Reasoning-based methods. Specific reasoning based methods, applied in the context of path planning for steerable needles, are illustrated. They are part of the “logic programming” category, in particular Answer Set Programming (ASP)

In particular, in [237]⁴, Segato *et al.*, exploited ASP semantics to model the brain environment and thus implement an Artificial Intelligence (AI) agent able to move within it, satisfying requirements, which can be customized depending on the specific application and based on the preferences expressed by domain experts, as surgeons and clinicians. Results show the ability of the ASP program to find the minimum number of

⁴Segato, A., Corbetta, V., Francesco, C., and De Momi, E. (2020). Inductive and Deductive Reasoning for Robotic Steerable Needle in Neurosurgery. In 2020-International Conference on Intelligent Robots and Systems. USA.

steps to reach the target and also the possibility to obtain different trajectories and thus optimizing distinct characteristics. In subsequent work, [243]⁵, they then developed a method, based on ASP optimization and classification, to support neurosurgeons at quickly determining the most appropriate surgical trajectory through the brain matter with minimized risk.

Table 2.8: Reasoning Based methods

Author, Year	Clinical Aim	Pathology, Anatomy	Method	Query	Auto/ Semi/ Pre/ Intra	Risk	Validation	Needle, Sensor	Needle Kinematic	Data	Dataset	Tool	Env	Tissue modelling	Benchmark measure	Results	
[237] Segato, 2020	DBS, DD	PD/cancer, brain	ASP	single	yes	pre	yes	in-silico	PBN	$d(2.5mm)$, $k_{max}(0.014mm)$	MRI	1 image	-	static	no	$d_{min}[mm]$, $d_{avg}[mm]$, $k_{mean}[mm]$	1, 21, 36
[243] Segato, 2021	DBS, DD	PD/cancer, brain	ASP	single	semi	pre	yes	in-silico	PBN	$d(2.5mm)$, $k_{max}(0.014mm)$	MRI, CT, 3D Model	1 image, 1 image, 1 scene	Unity3D	static	no	length[mm], $d_{min}[mm]$, $d_{avg}[mm]$, $k_{mean}[^\circ]$	41.2±1.3, 3.2±0.28, 28.3±3.57, 0.01±0.003

2.5.6 Multifusion-based methods

As observable in Table 2.9, different works, in the context of path planning for steerable needle, exploit multifusion-based methods. Multi-fusion-based algorithms manage problems where a single approach is usually proposed and cannot work to find an optimal path individually. For example, PRM cannot generate an optimal path itself. APF often jump into local minimum; graph-based optimal algorithms need preknown environmental skeleton information. Mathematical-based algorithms tend to be time consuming and unable to solve NP-hard problem with varying environments; thus researchers try to introduce a combination of different approaches to form a fast searching and global optimal algorithm. As schematized in Figure 2.14 the combined approaches, used in this context, involve BFS, Improved Conjugate Gradient (ICG), APF, PRM, SMR, DP, Heuristic, RRT, RRT-connect RRT-star, BIT-Star, IK, Spline, FFNN and GA.

BFS / ICG+APF

In [228], a BFS algorithm for the 3D path planning of micro-devices from the source to the destination in the cardiovascular system is applied to a force map, which is created with APF. Nevertheless, it ignores the physiological and technological constraints, such as the micro-device capability. A combination method of APF and improved conjugate gradient is proposed in [167] for trajectory planning of needle insertion for prostate cancer.

PRM / SMR+dynamic programming

In [15], Alterovitz *et al.* proposed a 2D motion planning framework for a bevel-tip needle that addresses the uncertainty in the needle motion. This is included by considering a PRM, which represents the configuration states in a random manner to build a collision-free connectivity map. Then, for each sampled state i , the planner computes the optimal action to be performed so that to maximize the probability of reaching the target being i the current needle state. This represents a MDP, and it is solved using ihDP [245]. The same approach but starting with SMR is defined, by [220] Reed *et al.*,

⁵Segato, A., Corbetta, V., Zangari, J., Z., Perri, S., Calimeri, F., and De Momi, E. (2021). Optimized 3D path planner for steerable catheters with deductive reasoning. In 2021 IEEE International Conference on Robotics and Automation, ICRA 2021 China, (pp. 1-7).

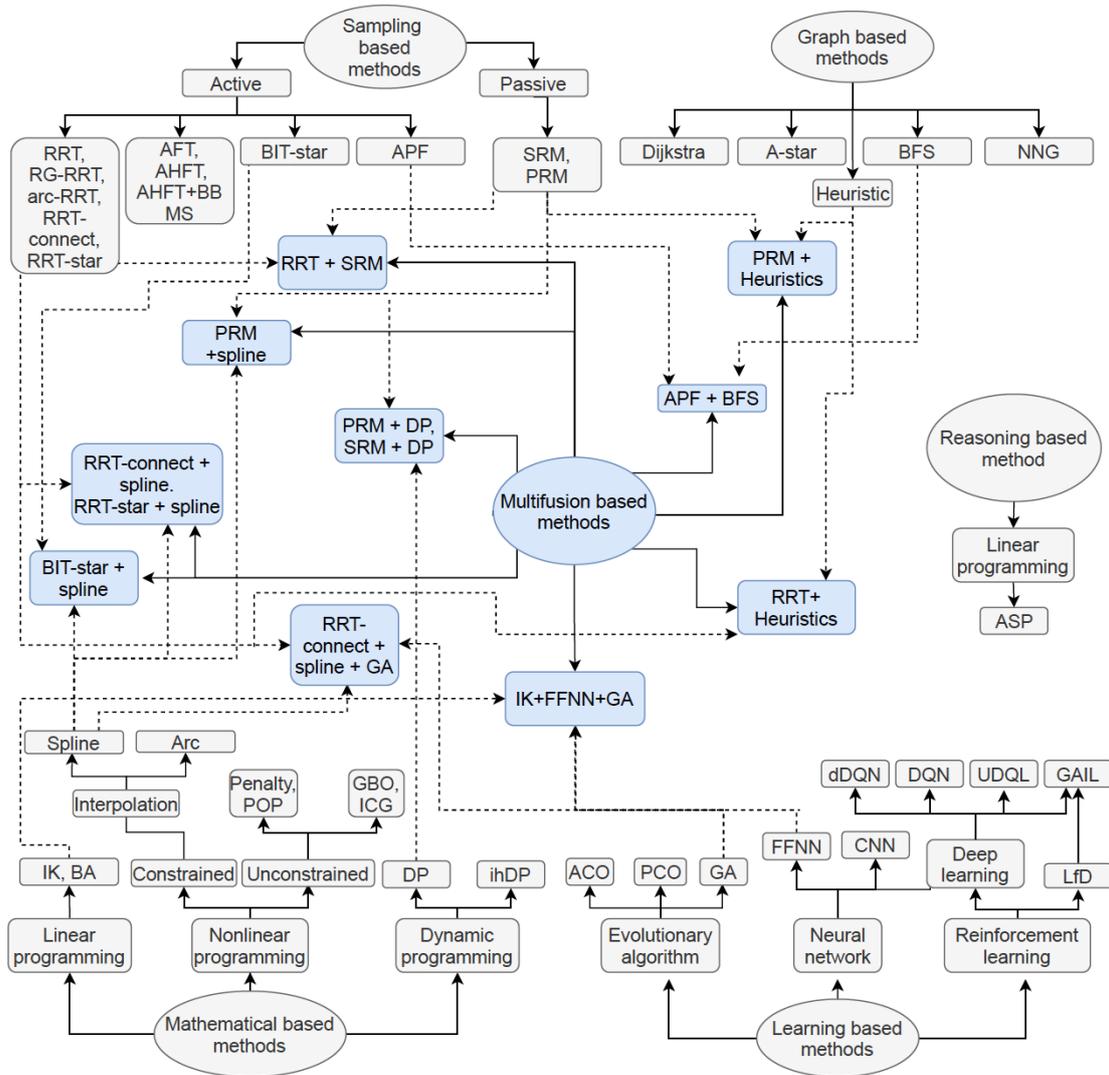


Figure 2.14: Multifusion-based methods. Specific algorithms of multifusion-based methods, applied in the context of path planning for steerable needles, are illustrated. The combined approaches involve graph-based methods, such as BFS algorithm, sampling-based methods, such as APF, PRM, SRM, RRT, RRT-connect RRT-star and BIT-Star algorithms, mathematical-based method, such as DP, IK and Spline algorithms, and learning-based methods, such as FFNN and GA algorithms

by initially sampling the configuration states in a random manner to build a collision-free connectivity map, solved using ihDP formulated as a MDP.

PRM / RRT+heuristic

Lobaton *et al.* developed an approach for optimal, collision-free path planning to visit multiple goals in any order [174]. The method builds a roadmap by sampling circles of constant curvature and generates a feasible transition between circles. The multiple goals objective takes the shape of a multi-query planning problem, solved by the algorithm using a minimum directed Steiner tree on the roadmap. The optimal path can visit all the target locations while minimizing the extent of tissue cut, i.e. the path length. Simulations are performed in 2D and 3D static environments ignoring possible tissue deformations. This article [11], by Aghdam, aims to design a 2-D path planner, based on RRT+Heuristic, for a steerable bevel-tip needle inside soft tissue to automatically reach multiple targets in the presence of obstacles while minimizing a clinician-specified cost function that is considered to be the needle insertion length. For both the studies, simulations are performed in 2D and 3D static environments ignoring possible tissue deformations.

RRT+SMR

The use of an RRT algorithm together with a SMR algorithm was firstly proposed by Alterovitz *et al.* in [17] and consists of an optimal motion planning that performs a trade-off between random-sampling exploration of the configuration space and refinement of the existing path according to a user-specified parameter. It guarantees probabilistic completeness and asymptotic converge to the optimal solution sparing computational time focusing the refinement procedure a path that can reach the target configuration.

PRM+spline

In enhanced PRM for pliable needle robotic surgery is proposed in [249] by Sudhakara *et al.*. The classical PRM method is enhanced using shape preserving Piece-wise Cubic Hermite Interpolation (PCHIP) technique, used to generate smooth trajectories, considering direction constraints on both source and target positions. Simulation experiments with a static obstacle environment are carried out to show the validity of the algorithm. However, the simulations are only performed in 2D environments, and the processing time is still heavy.

BIT-star / RRT-star / RRT-connect+spline

In [80], Favaro *et al.* proposed a solution based on an BIT-star + spline approach that runs within a dynamic subset of the original configuration space. In [77], Fauser *et al.* extended a previous work [78] by including a further step of translation from Bezier curves into circular arcs, demonstrating in a further increase in the distance from obstacles in simulated temporal bone surgery and intraluminal catheter insertion. In [240]⁶ proposed a solution based on an RRT-star approach that runs within a dynamic

⁶Segato, A., Pieri, V., Favaro, A., Riva, M., Falini, A., De Momi, E., and Castellano, A. (2019). Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei. *Frontiers in Robotics and AI*, (6, 70).

subset of Spline. This approach reduces the RRT-star complexity and the computational time. Path length, curvature and obstacle clearance are optimized according to a cost function while control errors are included in the planning phase. Simulated insertions were performed to test the method in complex DBS scenarios.

RRT-star+spline+GA

An evolution of the method proposed in [80] is reported in [82]. The method relies on a new heuristic, limiting the search space to a subset of reachable needle configuration considering its maximum degree of curvature. In addition, the path planning benefits from a novel evolutionary optimization procedure intended to correct the final needle path according to optimality criteria to minimise the path length, the magnitude and the variability of the path curvature, and keep the obstacle avoidance the larger as possible.

FFNN+IK+spline

In [36], Bian *et al.* proposed a self-learning algorithm based on a FFNN of IK for control of intracardiac robotic ablation catheters using splines. Performing an in-vitro validation on a phantom.

Table 2.9: Multifusion Based methods

Author, Year	Clinical Aim	Pathology, Anatomy	Method	Query	Auto/ Semi	Pre/ Intra	Risk	Validation	Needle, Sensor	Needle Kinematic	Data	Dataset	Tool	Env	Tissue modelling	Benchmark measure	Results	
[16] Alterovitz, 2005	MIS	-	PRM DP	+	single	yes	pre	no	in-silico	bevel-tip, no sensor	$d(2.5cm)$	2D map	1 map	-	static	no	-	
[228] Sabra, 2006	vascular surgery	cardiovascular system, brain	BFS APF	+	single	yes	pre	no	in-silico	micro robots, no sensor	-	MRI	1 image	Matlab	static	no	-	
[220] Reed, 2008	MIS	-	SMR DP	+	single	semi	intra	no	in-silico, in-vitro	bevel tip, camera	$d(0.61mm), k_{max}(0.16cm^{-1})$	2D model	1 scene	phantom	dynamic	yes(gelatine)	-	
[174] Lobaton, 2011	MIS	-	PRM + Heuristic	+	multi	yes	pre	no	in-silico	bevel tip, no sensor	-	2D map, 3D model	2 maps, 1 scene	-	static	no	-	
[17] Alterovitz, 2011	MIS	-	RRT SMR	+	single	yes	pre	no	in-silico	concentric tube, no sensor	-	3D model	1 scene	-	static	no	time[sec]	183
[167] Li, 2014	brachytherapy	cancer, prostate	APF ICG	+	single	yes	pre	no	in-silico	bevel tip, no sensor	-	MRI, 3D model	1 image, 1 scene	-	static	no	time[sec]	0.548
[36] Bian, 2015	RFA	atrial fibrillation, heart	IK + GA + FFNN	+	single	semi	intra	no	in-vitro	catheter, EM	$d(5mm)$	-	-	matlab, phantom	dynamic	yes(acrylic)	tError[mm]	0.12 ± 0.5
[80] Favaro, 2018	DBS	PD, brain	BIT-star + spline	+	multi	yes	pre	yes	in-silico	PBN	$d(2.5mm), k_{max}(0.014mm^{-1})$	MRI, CT	1 image, 1 image	3DSlicer	static	no	$d_{min}(mm)$	2.3
[249] Sudhakara, 2018	DD, biopsy	cancer	PRM + spline	+	single	yes	pre	no	in-silico	pre curved stylet, no sensor	-	2D map	1 map	Matlab	static	no	time[sec], length[cm]	691, 104.2
[77] Fauser, 2019	temporal bone ¹ /heart ² surgery	brain ¹ , aorta ²	spline + RRT-connect	+	multi	yes	pre	yes	in-silico	Hydraulically driven catheter ¹ , guidewire ²	$d^1(1.5cm), k_{max}^1(0.05mm^{-1}), l^1(6.8cm), d^2(3.6mm), k_{max}^2(0.1mm^{-1}), l(40mm)$	CT, 3D model	20 images ¹ , 40 images ² , 20 scene ² , 40 scene ²	-	static	no	SR ¹ , SR ² [%], #p ¹ , #p ² , time ¹ , time ² [msec], d_{avg}^1 , d_{avg}^2 [mm]	93,100 \pm 152.2, 8 \pm 3.2, 23.06 \pm 11.8, 0.6 \pm 0.1, 2.7 \pm 0.4, 3.7 \pm 0.5
[240] Segato, 2019	DBS	PD, brain	spline + RRT-star	+	multi	yes	pre	yes	in-silico	PBN, no sensor	$d(2.5cm), k_{max}(0.014mm^{-1})$	MRI, CT, 3D model	10 images, 10 images, 10 scenes	3Dslicer, Matlab	static	no	SR[%], time[sec], d_{min} [mm], d_{avg} [mm], α_{target} [$^\circ$]	99, 147.75, 0.75, 14, 80
[111] Aghdam, 2020	brachytherapy, biopsy	cancer, prostate	RRT+ Heuristic	+	multi	yes	intra	no	in-silico	bevel tip, no sensor	$k_{max}(0.2cm^{-1})$	2D maps	3 maps	Matlab	static	no	tError[mm], time[sec]	<1, 0.38 \pm 0.08
[82] Favaro, 2021	DD	cancer, brain	spline + RRT-start + GA	+	multi	yes	pre	yes	in-silico	PBN	$d(2.5cm), k_{max}(0.014mm^{-1})$	CT, MRI, 3D model	1 image, 1 image, 1 scene	3Dslicer, Matlab	static	no	time[sec], d_{min} [mm], d_{avg} [mm]	17.6, 0.25, 5.94

2.6 DISCUSSION

In recent years, steerable needles and catheters have increasingly caught the attention of many researchers in the neuroscience/medical field. Automatic or semi-automatic path

planning algorithms for steerable needles, in particular, have been proposed for finding ways to increase the quality and precision of targeting a specific anatomical area in order to improve surgical treatments. A systematic review of the recent path planning algorithm for robotic steerable needle and catheters was presented in this work. The analysis carried out for each paper concerns the clinical aim and background, the path planning method, the validation procedure and the quantitative results.

From the results, we can state that the most common surgical procedure was the biopsy, the most common anatomical area was the liver, and the most common pathology was cancer. Only half of the framework has a re-plan strategy for an intra-operative path planner. Only a quarter of the planner provides a risk map or risk analysis for the most delicate structure involved in the procedure and the possibility to get more than one optimized solution. Most of the algorithms were validated in silico. None of the cited works were validated in-vivo. Regarding the simulated environment, the most used tools were *Unity3D*, *3DSlicer* and *Matlab*. Less than half of the works were validated in a dynamic environment. Bevel tip was the most frequently used needles, followed by programmable bevel tip needles. 3D model data were the most frequently used input features. Electromagnetic sensors and cameras were the most frequently used sensors.

We identified six main categories for path planning with steerable needles that have been proposed in the literature: graph-based, sampling-based, mathematical-based, learning-based, reasoning based and multifusion-based. Graph methods are based on the discrete approximation of the planning problem. Only a few works among those analyzed, about 5%, have used this type of method, probably by exploiting its ability to ensure a final path that is "resolution complete" and "resolution optimal". But these methodologies exhibit a high computation time, $O(m \log n) \leq T \leq O(n^2)$, as the discretization of the environment becomes finer. For this reason, we can infer that they are not suitable for surgical applications, as high-resolution 3-D datasets are normally used. Sampling-based methods are built on the random sampling of the workspace. Most of the work among those analyzed, about 45%, relies on sampling-based approaches to find the optimal path. This because the random initial guess ensures escaping of local minimum, and this kind of algorithms does not rely much on environmental representation. Mathematical-based algorithms aim to describe the whole workspace in a mathematic form, with the advantage of describing all constraints with differential equations. This type of method is the second most used, with 24% application among the proposed work. This type of algorithms can easily represent the dynamic constraints of delicate tissues and the kinematic constraints of steerable needles. The problem of its applicability is related to the high computational cost proportional to the polynomial equation. Learning-based method can excellently deal with complex and dynamic unstructured constraints. In recent years we can see a growth in their use in the field of minimally invasive steerable needle surgery. This because they are more flexible than graph-based and sampling-based methods; indeed, they allow one to directly include all expected constraints and optimality criteria (obstacle clearance, kinematic constraints, minimum path length) in the optimization process, without the need for subsequent refinement steps, which are time-consuming. However, the training phase required by these approaches needs a great number of medical images and data not always available or accessible. Reasoning-based approaches have increased capability of explicitly repre-

sent domain knowledge; however, only two works make use of this approach because a path planning system for complex environments based only on a deductive reasoning-based method or similar approaches might be insufficient, as current implementations cannot handle an excessive increase of the search space and generalise on different environments. Multifusion-based algorithms synthesize several algorithms' advantages together to achieve global optimal and cost minimum. This kind of algorithms import the idea of complementation, that is, merging the merits of several algorithms. In the analyzed works, we have noticed that these algorithms can achieve several goals simultaneously for optimal path selection, such as maintaining a minimum distance from delicate structures, reducing path length and trajectory curvature. We also found a reduction in computational costs, $O(m \log n) \leq T$, as in [36, 82, 167], which is very important for real-time use in an intra-operational environment.

Based on the discussions we have just had, the most promising categories of path planning algorithms for steerable needles are Learning based, Reasoning based and Multifusion based, the latter intended as a combination of the first two.

2.7 CONCLUSION

The complex kinematics of steerable needles requires automatic solutions for the path planning task. Planners are asked to find feasible paths solving a multi-objective planning problem that requires meeting specific criteria such as the maximization of the distance from anatomical obstacles and the minimization of the path lengths. Many solutions have been proposed in the literature. This work aims to provide an integrated, synthesized overview of the current state of knowledge. For each paper analyzed, we answered the following questions: Which is the aim of the clinical procedure? For which pathology and in which anatomical area? Which kind of path planning method was used? How many optimized paths does it get? Is the path computed automatically? Is the path computed in real-time? Is the path computed according to the risk for more delicate structures? Which kind of validation was performed? Did they refer to a specific needle, sensor? Which ones? Did this needle have kinematic constraints? Which ones? How was the environment? Did they model tissue deformation? Which kind of benchmark measures were used for the validation? And which are the corresponding quantitative results. In this way, solutions are analyzed in terms of the context of the application, the methodology used, their efficiency, and the possibility of performing intra-operative replanning and dealing with uncertainties in needle motion and/or in the environment. The approaches used by authors to validate their solutions are also reported. The structure of analysis and synthesis proposed could allow the reader quickly identify the works in the literature that may represent their gold standard suitable to be applied in their clinical context. In addition, thanks to the synthesis performed and the immediate mode of exposure proposed, it could give the possibility to identify more quickly some aspects to be improved such as real-time planning, complete information expressing, and complex environments modelling.

The great possibilities associated with steerable needles in terms of therapy outcome and patient safety make path planning a topic of great interest for researchers and engineers. Many solutions have been proposed to cope with this task grouped in graph-based, sampling-based, mathematical-based, learning-based, reasoning based and multifusion-based solution. The structure of analysis and synthesis proposed could

ensure the reader quickly identifies the works in the literature that may represent their gold standard suitable to be applied in their clinical context. In addition, thanks to the synthesis performed and the immediate mode of exposure proposed, it could give the possibility to identify more quickly some aspects to be improved such as real-time planning, complete information expressing, and complex environments modelling. A relevant aspect evidenced in several of the works herein reported consists in the uncertainty associated with needle motion and tissue deformation that can arise during needle deployment and jeopardise the insertion accuracy. Despite the remarkable performance and the level of readiness reached in simulations by many solutions reported in this survey, what is often missing is a thorough evaluation of the planning performance in conditions that mimic a real needle insertion through, e.g., in-vitro or ex-vivo experiments. This would make it possible to assess the feasibility of the proposed algorithm in the presence of tissue and modelling uncertainties, which represents one of the big open questions in steerable needle path planning. Recently, big efforts have been made to explore new optimization strategies and methods, e.g. the use of RL. Still, the high complexity of needle kinematics combined with the uncertainties and the dynamic nature of the environment in which they operate lead path planning open to further relevant improvements. This work recommends a combination of multifusion-based path planning and environment modelling methods.

This comprehensive analysis influenced this thesis on steerable catheters for key-hole neurosurgery, leading us to apply mainly Learning based, Reasoning based and Multifusion based algorithms, the latter being a combination of the first two. The proposed algorithms were compared with methods that represent the State-Of-The-Arts (SOTA) in the KN context, which are mainly sampling-based approaches, based on the metrics described in Section 2.4 and identified as indicative in the MIS context. The proposed planning approaches for pre- and intra-operative procedures were validated both in simulation, in a static and dynamic environment, and in vitro in gelatin.

A Hybrid inductive learning-based and deductive reasoning-based 3-D pre-operative path planning method in complex environments

Traditional path planning methods, such as sampling-based and iterative approaches, allow for optimal path's computation in complex environments. Nonetheless, environment exploration is subject to rules which can be obtained by domain experts and could be used for improving the search.

In this chapter we investigate **Hypothesis 1**, integrating inductive techniques that generate path candidates with deductive techniques that choose the preferred ones. In particular, an inductive learning model is trained with expert demonstrations and with rules translated into a reward function, while logic programming is used to choose the starting point according to some domain expert's suggestions. We discuss, as use case, 3-D path planning for neurosurgical steerable needles. Results show that the proposed method computes optimal paths in terms of obstacle clearance and kinematic constraints compliance (**Hypothesis 1.1**), and is able to outperform state-of-the-art approaches in terms of safety distance-from-obstacles respect, smoothness, and computational time (**Hypothesis 1.2**). The application of the method resulted in a journal paper [1]¹ currently under review.

3.1 INTRODUCTION

Moving agents, such as mobile robots, are increasingly being employed in many complex environments. Starting from the initial applications of mobile robots to manu-

¹Segato, A., Calimeri, F., Testa I., Corbetta, V., Riva, M., and De Momi, E. A Hybrid inductive learning-based and deductive reasoning-based 3-D path planning method in complex environments. *Autonomous Robots*. (resubmitted after major revisions on November 2021).

facturing industries, a variety of robotic systems have been developed, and they have shown their effectiveness in performing different kinds of tasks, including smart home environments [12], airports [293], shopping malls [135], and manufacturing laboratories [51].

Nowadays, path planning is fruitfully employed in many fields, such as entertainment, medicine, mining, rescuing, education, military, space, agriculture, robots for elderly persons, automated guided vehicles, for transferring goods in a factory, unmanned bomb disposal robots and planet exploration robots [222]. Apart from robotic applications, path planning finds use in planning the routes on circuit boards, obtaining the hierarchical routes for networks in wireless mobile communication, planning the path for digital artists in computer graphics, reconnaissance robots and in computational biology to understand probable protein folding paths [217].

In recent years, path planning has been also widely used in surgery [9]. In current clinical practice, a growing number of minimally invasive procedures rely on the use of needles, such as biopsies, brachithery for radioactive seeds placement, abscess drainage and drug infusion [244]. With respect to standard open surgeries, the small diameter of the needle allows to access the targeted anatomy inflicting limited tissue damage and thus reducing the risks for the patient and speed up the recovery.

Over the last two decades, different research groups have focused their efforts on the development of needles able to autonomously steer inside the tissue. These needles can perform curvilinear trajectories planned to maximize the distance from sensitive anatomical structures to be avoided and reach targets otherwise inaccessible via rectilinear insertion paths [271]. Accurate placement of the needle tip inside tissue is challenging, especially when the target moves and anatomical obstacles must be avoided. Moreover, the complex kinematics of steerable needles [81] make the path planning challenging requiring the aid of automatic or semi-automatic path planning solutions.

In general, moving agents in a static or dynamic known environment means finding one or more admissible paths from a starting configuration to a target configuration, avoiding obstacles and some movement possibilities, identified as kinematic constraints. The path planning problem is part of a larger class of "scheduling" and "routing problems", and it is known to be non-deterministic polynomial-time hard and complete [192]. In general, path planning algorithms performance can be evaluated over two main characteristics: "completeness" and "optimality". An algorithm is said to be complete if it can find a solution in a definite interval of time, provided that the solution exists, or failure otherwise. An algorithm is optimal if no other algorithm uses less time or space complexity. Given a path, path length is defined as the total distance covered by the moving agent from the starting position to the target, path safety represents the distance from the path to the nearest obstacle, and computation time is the time required to compute a path.

In this work, we propose a framework that couples inductive and deductive techniques in order to improve path planning performances. In particular, an inductive learning model, relying on demonstrations performed by expert operators, is in charge of generating a set of paths as candidate solutions; a deductive reasoning module selects then the "best" starting point, according to explicit knowledge modeled over domain experts suggestions. Interestingly, this kind of coupling allows us to transfer to the automated path planner some of the knowledge available at human level: the induc-

Table 3.1: *Related Work*

<i>Method</i>	<i>Optimal</i>	<i>Complete</i>	<i>Scalable</i>	<i>Computation time</i>	<i>Complex environment</i>	<i>Smooth path</i>
Graph-based	✓	✓	×	↑	×	×
Sampling-based	×	×	✓	↓	✓	×
Optimisation-based	✓	✓	×	↑	✓	✓
Heuristic-based	×	×	✓	↑	×	×
Learning-based	×	×	✓	↓	✓	✓
Reasoning-based	✓	✓	×	↓	×	×
ILDR	✓	✓	✓	↓	✓	✓

✓=yes, ↑=no, ↑=high, ↓=low

tive learning module “catches” via demonstrations expert capabilities that are hard to explicitly express (e.g., visual-spatial, bodily-kinesthetic), while the deductive module formally encodes what has been elaborated by the experts upon long-lasting practice (e.g., domain knowledge, best practices). Furthermore, the deductive technique based on a declarative formalism grants several advantages: on the one hand, it makes the knowledge easy to maintain and update; on the other hand, it allows us to provide the final user with a highly customizable tool with real time visual feedback, as she can decide what is important for choosing the best path, and to what extent, for each case at hand.

Eventually, we assess the viability of the proposed approach in a use case, namely 3-D path planning for neurosurgical steerable needle, proving that it stands or even outperform state-of-the art solutions.

3.2 RELATED WORK

Several approaches for path planning have been proposed in literature: graph-based, sampling-based, optimisation-based, heuristic-based, learning-based, reasoning-based methods. These methods are described below, and summarised in Table 3.1, according to optimality (i.e. an algorithm is known to be and “optimal” since it can estimate the best path, according to a certain criteria, given the specific resolution of the approximation); completeness (i.e. an algorithms is known to be “complete”, as it can determine whether a solution exists in finite time); scalability (i.e. an algorithms is known to be “scalable” as it can plan a path in a reasonable time even if the search space increase in size); computational time (i.e. the execution time to obtain a solution); the ability to plan within a complex environment (i.e. the environment is composed of many obstacles with elaborate shapes, narrow passages and tangled locations); and the ability to obtain a smooth path (i.e. able to minimise the along the curvature).

Dijkstra algorithm [62] and A* [110] are graph-based methods based on the discrete approximation of the planning problem. Many methods represent the environment as a square graph, or as an irregular graph [134], or a Voronoi diagram [92]. A search is performed in order to find an optimal path. These algorithms are known to be “resolution-complete”, as the one proposed in [88] that provides more guarantees also in terms of efficiency, and “resolution-optimal”. This approach may also be used for identifying a restricted area where further optimisation refinements can be performed [124]. Notably, even though graph-based methods are relatively simple to

implement, they require considerable computational time as the environment size increases [27] or it becomes complex. Tangent graph-based planning methods for a given environment build a graph by selecting collision-free common tangents between the obstacles. These methods allow more accurate path planning around curved obstacles without errors caused by polygonal approximation; however, these methods are not always suitable when considering the kinematics limitations of a moving agent and require additional optimisation steps [261] to obtain a smooth path.

Sampling-based methods are based on a random sampling of the working space, with the aim of significantly reducing the computational time. Rapidly-exploring Random Tree (RRT) [157] is an exploration algorithm for quick search in high-dimensional spaces, more efficient than brute-force exploration of the state space. In fact, this class of methods is scalable and capable of planning in a complex environment. Its enhanced versions, RRT* [79, 128] and bidirectional-RRT [137] are “probabilistically complete” since the probability to find an existing solution tends to one, as the number of samples goes to infinity, and “asymptotically optimal”, as they can refine an initial raw path by increasing the sampling density of the volume.

Paths computed with the approaches mentioned above can be further refined using Bezier curves [121], splines [155], polynomial basis functions [214], or with optimisation-based methods such as evolutionary algorithms, simulated annealing, and particle swarm [35] to obtain a smooth path. These approaches have the advantage of working properly in complex environments, as demonstrated in [82]²; however, they require higher computational time than the sampling-based methods.

AI has been increasingly used while dealing with path planning tasks [74] in the last decade. Heuristic-based techniques such as greedy [246] and genetic algorithm [279] are two examples of AI approaches; they belong both to the class of optimisation procedure. Greedy algorithms for path planning are often used in combination with other approaches. This kind of algorithms fails to find the optimal solution, as it takes decisions merely on the basis of the information available at each iteration step, without considering the overall picture. Genetic algorithms are also used to generate solutions for path optimisation problems based on operations like mutation, crossover and selection. For this kind of algorithms, the most relevant limit is computational time, as it significantly increases with the search space. Completeness depends on the heuristic function.

Learning-based methods are more flexible than graph-based and sampling-based methods [282]; indeed, they allow one to directly include all expected constraints and optimality criteria (obstacle clearance, kinematic constraints, minimum path length) in the optimisation process, without the need for subsequent refinement steps, which are time-consuming and may still not lead to the optimal path [241]³. [183] and [171] showed that Deep Reinforcement Learning (DRL) is suitable for solving path planning problems, and several studies [181, 182, 253] applied DRL in path planning. [196] made use of the DRL approach to a grid path planning problem with promising results on small environments. Inverse Reinforcement Learning, also known as an Inductive Learning (IL) technique [178] that is a process where the learner discovers rules by ob-

²Favaro, A., Segato, A., Muretti, F., and De Momi, E. (2021). An Evolutionary-Optimized Surgical Path Planner for a Programmable Bevel-Tip Needle. *IEEE Transactions on Robotics*.

³Segato, A., Sestini, L., Castellano, A., and De Momi, E. (2020, May). GA3C reinforcement learning for surgical steerable catheter path planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2429-2435). IEEE.

serving examples, has been applied to a wide range of domains, including autonomous driving [280], robotic manipulation [58] and grid-world planning [190]. In general learning-based methods are not optimal or complete, the computational time is not high, and it is not increasing when the search space increases. They perform well in complex environments even if it is dynamic because they don't need prior information about obstacles.

Reasoning-based approaches for path planning have been successfully designed, providing high-level methods like in [169]. They are optimal and complete [212] successfully solved the path planning problem, [73, 102] encoded multi-agents pathfindings and [72] used a deductive reasoning-based approach to control and plan the actions of multiple housekeeping robots whose aim is to tidy up a house in the shortest possible time and to avoid collisions between themselves and other obstacles. Reasoning-based approaches have the capability of explicitly representing domain knowledge; however, a path planning system for complex environments based only on a deductive reasoning-based method or similar approaches might be insufficient, as current implementations cannot handle an excessive increase of the search space and generalise on different environments [72].

3.3 MATERIALS AND METHODS

In this work, we propose a novel approach for path planning in 3-D complex environments that combines IL and Deductive Reasoning (DR); we refer to it as the *ILDR* method. In this way we can exploit all the advantages of the first technique (scalability, low computation time, capacity to plan in a complex environment taking into account kinematic constraints of the robot providing a smooth path if necessary) and the second technique (completeness, optimality, capability of explicitly encoding medical knowledge, solid theoretical bases coupled with a fully declarative nature; this allows to produce formal specifications that are already executable without the need for additional algorithmic coding, thus fostering fast prototyping and easing the interaction with domain experts). The novel aspects of this approach is that it not only includes all the fundamental requirements of the path planning task, but take also into account the expert's knowledge to fully understand the decision-making process that guides the optimal path selection.

To test viability and effectiveness of our approach, we have chosen keyhole neurosurgery as a case study. In this context, path planning is crucial when the pathological target (e.g., a tumor or left subthalamic nucleus (LSTN)) is located in deep brain areas and cannot be safely reached by a flexible probe. Thus, the optimization criteria involved in finding the best path in this case scenario are: the maximisation of the minimum and average distance with to obstacles, so as to avoid delicate structures in the brain, the minimisation of the length and curvature of the path, so as to limit damage to the brain matter.

3.3.1 Problem statement

Moving agent

Let us consider an "agent", showed in Figure 3.1a, moving in a 3D complex environment. The agent in this work is the tip of a steerable needle, a new biomimetic flexible

probe (PBN) [44], that can translate and rotate in space. Its kinematic constraints are the “diameter”, d_{out} , and the “maximum curvature”, k_{max} .

Environment

The “3D complex environment” (Env) is showed in Figure 3.1b. The “configuration space”, C -space, is the set of all the possible t “agent configurations”, \mathbf{T}_{agent_t} , with $t \in C$ -space. The agent configurations are described by poses, denoted as 4×4 transformation matrices:

$$\mathbf{T}_{agent_t} = \begin{pmatrix} \mathbf{R}_{agent_t} & \mathbf{p}_{agent_t} \\ \mathbf{0} & 1 \end{pmatrix}$$

where $\mathbf{p}_{agent_t} \in \mathbb{R}^3$ is the tip position and $\mathbf{R}_{agent_t} \in SO(3)$ is the orientation relative to a world coordinate frame.

The “obstacle space”, $C_{obst} \in C$ -space, is the space occupied by obstacles. The “free space” $C_{free} \in C$ -space, is the set of possible configurations ($\mathbf{T}_{agent} = \mathbf{T}_{free} \in C_{free}$; $\mathbf{T}_{agent} \neq \mathbf{T}_{obst} \in C_{obst}$). Moreover we can define: the “start configurations” $\mathbf{T}_{start_k} \in C_{free}$ with $k \in 1..N$, the “target space” $C_{target} \in C_{free}$, that is a subspace of “free space” which denotes where we want the needle to move to, and the “target configuration” $\mathbf{T}_{target} \in C_{target}$.

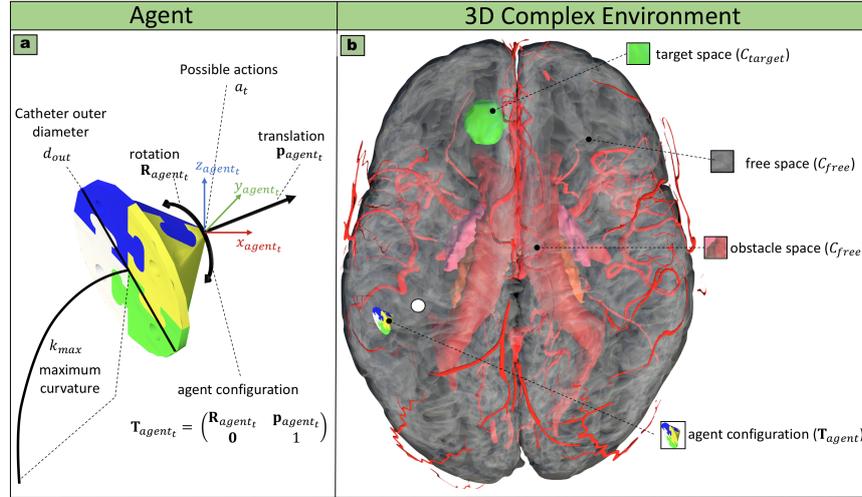


Figure 3.1: Agent and 3D Complex Environment. a)(on the left) The moving agent kinematic constraints are the needle diameter (d_{out}) and the maximum curvature (k_{max}) that it can perform. At t – th time step it can translate (\mathbf{p}_{agent_t}) and rotate (\mathbf{R}_{agent_t}), performing an action a_t , from its configuration, \mathbf{T}_{agent_t} . b)(on the right) The 3D complex environment is represented by a brain structure, the obstacle space (C_{obst}), the free space (C_{free}), the agent configuration (\mathbf{T}_{agent}).

Actions

The agent at every t -th time step can take an action $a_t = (x_t, y_t, z_t, \alpha_t, \beta_t, \gamma_t)$, moving towards the \mathbf{T}_{target} , where x, y and z are the axes and α, β, γ the angles around the axes respectively. Actions moving the agent toward an \mathbf{T}_{obst} or outside the Env are considered inadmissible.

Path planning problem

The path planning problem, described in Figure 3.2a, can be formulated as follows: given the “start configurations” \mathbf{T}_{start_k} , $k = 1 : N$ and a “target configuration”, \mathbf{T}_{target} , the task is to find a path, $Q = \{\mathbf{T}_{agent_0}, \mathbf{T}_{agent_1}, \dots, \mathbf{T}_{agent_{n-1}}\}$, $\mathbf{T}_{agent_0} = \mathbf{T}_{start_k}$, $\mathbf{T}_{agent_{n-1}} = \mathbf{T}_{target}$, and \mathbf{T}_{agent_t} and $\mathbf{T}_{agent_{t+1}}$ are connected by straight segments, as an admissible sequence of “agent configurations”.

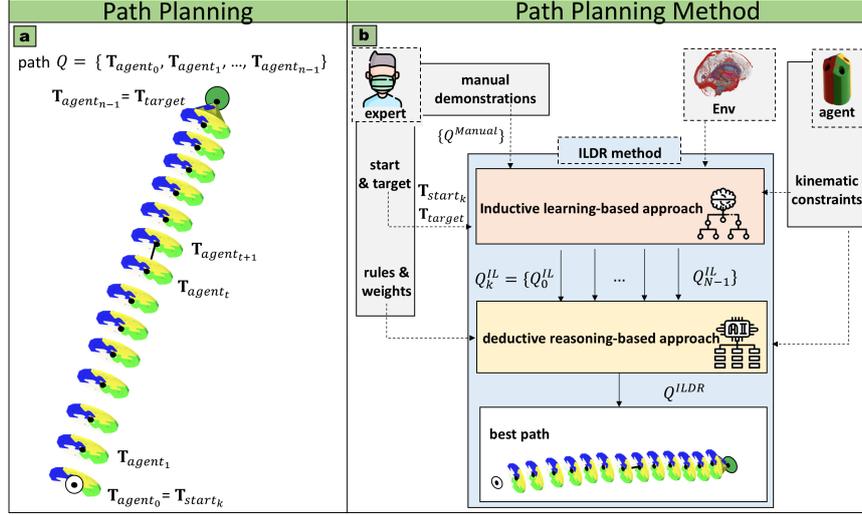


Figure 3.2: Path Planning Problem and Method. a) Schematic representation of the 3D path planning problem elements, start configuration ($\mathbf{T}_{start_k} = \mathbf{T}_{agent_0}$) and target configuration ($\mathbf{T}_{target} = \mathbf{T}_{agent_{n-1}}$). The task is to find the optimal path ($Q = \{\mathbf{T}_{agent_0}, \mathbf{T}_{agent_1}, \dots, \mathbf{T}_{agent_{n-1}}\}$). At every $t - th$ time step \mathbf{T}_{agent_t} and $\mathbf{T}_{agent_{t+1}}$ are connected by straight segments, as an admissible sequence of “agent configurations”, taking into account the “obstacle space”, C_{obs} . b) The proposed path planning algorithm exploits two merged approaches: inductive learning-based and deductive reasoning-based approaches. The expert indicates N start configurations, (\mathbf{T}_{start_k} with $k \in N$), the target configuration, (\mathbf{T}_{target}) the rules and performs a set of demonstrations. While the agent acts and observes in the virtual environment. The expert’s demonstrations and the agent’s observations feed the inductive approach that generates a set of optimal trajectories $\{Q_k^{IL}\}$ with k equal to the N start configuration. The rules and the kinematic constraints feed the deductive approach that extracts the best path Q^{ILDR} .

3.3.2 ILDR architecture

The herein proposed approach to path planning in 3-D complex environments, referred to as the ILDR method, is summarized in Figure 3.2b. The best path is obtained by means of a learning approach that generates a set of path candidates and an expert classifier that selects the “preferred” one, according to a set of rules set by the domain expert.

Inductive learning model

As shown in Figure 3.3, the inductive component iteratively trains the moving agent to generate optimal paths, $\{Q_k^{IL}\}$, thanks to expert’s manual demonstrations, $\{Q^{Manual}\}$.

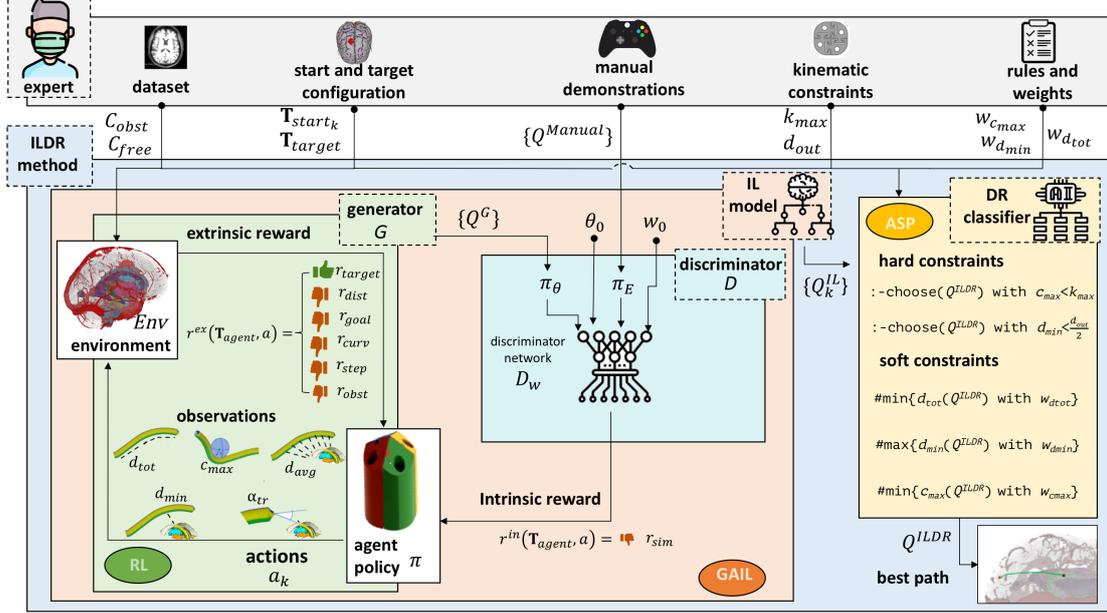


Figure 3.3: ILDR Architecture. The expert gives in input the dataset (C_{free} and C_{obst}), the start and target configuration (\mathbf{T}_{start} and \mathbf{T}_{target}) and the kinematics constraints (d_{out} and k_{max}). The IL model is trained through a loop that starts with paths generated by an expert ($\{Q^{Manual}\}$) and paths ($\{Q^G\}$) performed by the network generator (G) based on a RL approach. With a GAIL approach a discriminator (D) with its network (D_w), taking in input the expert and generator policies (π_E and π_θ) the starting value of the policy's parameters and of the discriminator (θ_0 and w_0) makes a comparison of these two paths generating an intrinsic reward ($r^{in}(\mathbf{T}_{agent}, a)$) based on similarity score (r_{sim}), updating the agent's policy π . The loop continues until the generator, moving in the environment (Env) with actions (a_i), collecting observations (d_{tot} , d_{min} , d_{avg} , c_{max} and α_{tr}) and computing an extrinsic reward ($r^{ex}(\mathbf{T}_{agent}, a)$), can produce a path similar to the expert's demonstrations and that respects the kinematic constraints. Once the IL model is trained, it generates a set of optimal paths; the weights ($w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$) and the kinematic constraints are taken as input by the DR classifier that extracts the best path with an approach based on ASP. Applying hard and soft constraints obtaining the best path Q^{ILDR}

Generator

With several successful applications in robot control, RL that searches for optimal policies by interacting with the environment becomes one potential solution to learn how to plan a path. [241]⁴, based on DRL, demonstrated its capability of directly learning the complicated policy of planning a path for steerable needle.

In the proposed approach, the agent interacts with the environment, Env . At each time step (t), the agent updates its current state (\mathbf{T}_{agent}) and selects an action (a_t), according to the policy (π), such that:

$$\pi(\mathbf{T}_{agent}, Env) = a_t \quad (3.1)$$

In response, the agent receives the next state ($\mathbf{T}_{agent_{t+1}}$) and observations (explained in the next paragraph). The goal of the agent is to determine an optimal policy π allowing it to take actions inside the environment, maximizing, at each t , the cumulative extrinsic

⁴Segato, A., Sestini, L., Castellano, A., and De Momi, E. (2020, May). GA3C reinforcement learning for surgical steerable catheter path planning. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2429-2435). IEEE.

reward $r^{ex}(\mathbf{T}_{agent}, a)$. The latter is associated to the real, full state of the system and to the agent's observations. In this way, the moving agent successfully learns to follow the best path in accomplishing its task and its constraints of maximum curvature, k_{max} , and diameter, d_{out} .

Observations. The observations collected step-by-step by the agent while exploring the environment are: the length (d_{tot}) of the path, the minimum (d_{min}) and average (d_{avg}) distances from the obstacle space $\{C_{Obst}\}$, the maximum curvature of the path (c_{max}), the target angle (α_{target}) and the computational time (T).

For each path $Q^G = \{\mathbf{T}_{agent_0}, \mathbf{T}_{agent_1}, \dots, \mathbf{T}_{agent_{n-1}}\}$ where $\mathbf{T}_{agent_0} = \mathbf{T}_{start_k}$ $k = 1 : N$, the observations are:

- Path length (d_{tot}): the distance, d , between any two positions can be calculated based on the Euclidean distance:

$$d(\overline{\mathbf{p}_{agent_t} \mathbf{p}_{agent_{t+1}}}) = \|\mathbf{p}_{agent_t} - \mathbf{p}_{agent_{t+1}}\| \quad (3.2)$$

with $t \in 1, \dots, n, n = \|Q^G\|$ and

$$d_{tot} = \sum_{t=0}^{n-1} d(\overline{\mathbf{p}_{agent_t} \mathbf{p}_{agent_{t+1}}}) \quad (3.3)$$

- Minimum distance (d_{min}): given the the line segment ($\overline{\mathbf{p}_{agent_t} \mathbf{p}_{agent_{t+1}}}$) and the m obstacles represented by the occupied configurations $\mathbf{T}_{obst_j} = \begin{pmatrix} \mathbf{R}_{obst_j} & \mathbf{p}_{obst_j} \\ \mathbf{0} & 1 \end{pmatrix}$ with $j \in 1, \dots, m, m = \|C_{obst}\|$, the minimum distance of the path from the nearest obstacle indicating the level of safety is the minimum length between the line segments and the closest obstacle, such that:

$$d_{min} = \min\{d(\overline{\mathbf{p}_{agent_t} \mathbf{p}_{agent_{t+1}}}, \mathbf{p}_{obst_j})\} \forall t, \forall j \quad (3.4)$$

- Average distance (d_{avg}) of the path from all the nearest m obstacles: it is calculated over the whole length of the path with respect to all the obstacles, such that:

$$d_{avg} = \frac{1}{n \cdot m} \sum_{t=1}^n \sum_{j=1}^m d(\overline{\mathbf{p}_{agent_t} \mathbf{p}_{agent_{t+1}}}, \mathbf{p}_{obst_j}) \forall t, \forall j \quad (3.5)$$

- Maximum curvature (c_{max}) of the path: curvature c of a path in a 3-D space is the inverse of the radius r_t of a sphere passing through 4 positions of the path ($\mathbf{p}_{agent_t}, \mathbf{p}_{agent_{t+1}}, \mathbf{p}_{agent_{t+2}}, \mathbf{p}_{agent_{t+3}}$) computed for each t (the method is explained comprehensively in Appendix 8.2). Subsequently, the maximum curvature can be extracted as follows:

$$c_t = \frac{1}{r_t} \quad (3.6)$$

$$c_{max} = \max_{0 \leq t < n-3} c_t \quad (3.7)$$

- Target angle (α_{target}): given the 3-D unit vector representing the agent direction $\overrightarrow{\mathbf{P}_{agent_t} \mathbf{P}_{agent_{t+1}}}$ and the one representing the target direction $\overrightarrow{\mathbf{P}_{agent_t} \mathbf{P}_{target}}$, the target angle is defined as follows:

$$\alpha_{target} = \arccos(\overrightarrow{\mathbf{P}_{agent_t} \mathbf{P}_{agent_{t+1}}} \cdot \overrightarrow{\mathbf{P}_{agent_t} \mathbf{P}_{target}}) \quad (3.8)$$

Reward function (Algorithm 1) the reward function associated with each time step t is shaped in order to make the agent learn to optimise the path, according to three main requirements:

1. path length minimisation
2. obstacle clearance maximisation
3. moving agent's kinematic constraints respect

The reward $r^{ex}(\mathbf{T}_{agent_t}, a)$ is defined as:

$$r^{ex}(\mathbf{T}_{agent_t}, a_t) = \begin{cases} r_{goal_t} & \text{if } \mathbf{T}_{agent_t} = \mathbf{T}_{target} \\ r_{obst_t} & \text{if } \mathbf{T}_{agent_t} = \mathbf{T}_{obst} \\ r_{step_t} + r_{dist_t} + r_{curv_t} + r_{target_t} & \text{otherwise} \end{cases} \quad (3.9)$$

- A positive constant reward, r_{goal_t} , is given upon reaching the target.
- A negative constant reward, r_{obst_t} , is given if an obstacle collision is detected.
- A negative reward, $r_{step_t} = -\frac{1}{S_{max}}$, is given at each step t of the agent in order to obtain a reduction in the computational time (T), where S_{max} corresponds to a fixed threshold representing the maximum number of steps t .
- A negative constant reward, r_{dist_t} , is added whenever the minimum value of d_{min_t} is lower than a predefined safe distance ($d_{safe} = \frac{d_{out}}{2}$), corresponding to the occupancy of the moving agent. This reward aims to maximise the d_{min_t} to reduce the risk of collision and increase the safety rate of the path.
- A negative constant reward, r_{curv_t} , is assigned if the current path curvature c_t overcomes the maximum value of the curvature of the moving agents specified by the user (k_{max}).
- Finally, a negative reward, $r_{target_t} = -\frac{\alpha_{target_t}}{\alpha_{max}} r_{deg}$, is added to minimise α_{target_t} in order to further minimise c and T parameters. The value of this reward is proportional to the ratio between α_{target_t} and the maximum angle α_{max} .

The parameters of the reward function are reported in Table 3.2. The optimal parameters' value have been obtained by fine-tuning with grid-search procedure. Sub-optimal parameter configurations caused the agent to learn and apply inappropriate actions, e.g. moving too close to obstacles, going in the opposite direction with respect to the target, choosing non-optimal paths in terms of distance.

In this way, with the generator, G , we are able to obtain new paths, Q^G .

Algorithm 1 Reward function - pseudocode for each episode

```

1: for  $t$  in  $S_{max}$  do
2:   Compute action  $a_t$ 
3:   Collect Observations  $d_{min_t}, d_{avg_t}, d_{tot_t}, c_t$  and  $\alpha_{target_t}$ 
4:    $addReward(-1/S_{max})$ 
5:   if  $\mathbf{T}_{agent_t} = \mathbf{T}_{target}$  then
6:      $addReward(1)$ 
7:      $newEpisode()$ 
8:   else if  $\mathbf{T}_{agent_t} = \mathbf{T}_{obst}$  then
9:      $addReward(-1)$ 
10:     $newEpisode()$ 
11:  end if
12:  if  $d_{min_t} < d_{safe}$  then
13:     $addReward(-0.001)$ 
14:  end if
15:  if  $c_t > k_{max}$  then
16:     $addReward(-0.001)$ 
17:  end if
18:  if  $\alpha_{target_t} \neq NaN$  then
19:     $addReward(-\alpha_{target_t}/\alpha_{max} * 0.0001)$ 
20:  end if
21: end for

```

Table 3.2: Training parameters.

r_{goal_t}	r_{obst_t}	r_{curv_t}	r_{dist_t}	S_{max}	α_{max}	r_{deg}
1	-1	-0.001	-0.001	5000	180	-0.001

Discriminator

In GAIL [115] the task of the discriminator, D , is to distinguish between the paths, $\{Q^G\}$, generated by G with a RL approach, and the demonstrated paths $\{Q^{Manual}\}$. When D cannot distinguish $\{Q^G\}$ from $\{Q^{Manual}\}$, then G has successfully matched the demonstrated path, $\{Q^{Manual}\}$, reaching a level equal to or higher than one of the expert thanks to the additional observations collected and extrinsic reward received.

As showed in Algorithm 2, the proposed path planning approach receives in input the expert’s paths, $\{Q^{Manual}\}$, the path generated from the Generator, $\{Q^G\}$, and the initialization of the policy’s parameters, θ_0 , and of the discriminator, w_0 . The path $\{Q^G\}$ fits a parameter policy π_θ (where θ represent weights), while the manual demonstration, $\{Q^{Manual}\}$, fits an expert policy, π_E . The discriminator network D_w (w , weights) learns to distinguish the generated policy, π_θ , from the expert one, π_E . The parameters w of D_w are updated in order to maximise Equation 3.10:

$$\max_{\nabla_w} E_{\pi_\theta} [\nabla_w \log (D_w (\mathbf{T}_{agent}, a))] + E_{\pi_E} [\nabla_w \log (1 - D_w (\mathbf{T}_{agent}, a))] \quad (3.10)$$

where ∇_w is the gradient, $E[]$ is the expectation value with respect to a policy, π_θ , or to the expert policy, π_E , $D_w (\mathbf{T}_{agent}, a)$ is the Discriminator network that evaluates the state (\mathbf{T}_{agent}) and the action (a).

Trust Region Policy (TRPO) [232] assures that the change of parameter θ between

$\pi_\theta + 1$ and π_θ is limited:

$$\theta = \min_{\nabla_\theta} E_{\pi_\theta} [\nabla_\theta \log \pi_\theta(a|\mathbf{T}_{agent}) \cdot Q(\mathbf{T}_{agent}, a)] - \lambda H(\pi_\theta) \quad (3.11)$$

where ∇_θ is the gradient, $H(\pi_\theta) = E_{\pi_\theta} [-\log \pi(a|\mathbf{T}_{agent})]$ is the causal entropy and its value by $\lambda \in \mathfrak{R}$, $\lambda > 0$.

Finally the discriminator, D , updates the agent’s policy π to be close to the expert policy π_E :

$$\pi = \arg \max_{\pi_\theta} E_{\pi_{\theta_i}} [-\log(1 - D_w(\mathbf{T}_{agent}, a))] \quad (3.12)$$

using an intrinsic reward, defined as $r^{in}(\mathbf{T}_{agent}, a) = r_{sim}$. The similarity reward r_{sim} proportional to $-\log(1 - D_w(\mathbf{T}_{agent}, a))$ is an increasing reward when the results of D_w approaches $r_{sim} = 1$, i.e. D_w is not able to discriminate well the two paths. The inverse happens when D_w approaches $r_{sim} = 0$, i.e. it is able to discriminate well between the two paths, and the reward goes to 0.

The trained IL model generates the paths $\{Q_k^{IL}\} k = 1 : N$.

Algorithm 2 Path planning with GAIL

Input: Expert paths $\{Q^{Manual}\} \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0

1: **for** $i=0,1,2,\dots$ **do**

2: Generated paths $\{Q^G\} \sim \pi_{\theta_i}$

3: Update the D_w parameters from w_i to w_{i+1} with the gradient ∇_w :

4:

$$E_{\pi_{\theta_i}} [\nabla_w \log (D_w(\mathbf{T}_{agent}, a))] + E_{\pi_E} [\nabla_w \log (1 - D_w(\mathbf{T}_{agent}, a))]$$

5: Take a policy step from θ_i and θ_{i+1} with TRPO cost function $\log(D_{w_{i+1}}(\mathbf{T}_{agent}, a))$ with the gradient ∇_θ :

6:

$$E_{\pi_{\theta_i}} [\nabla_\theta \log \pi_\theta(a|\mathbf{T}_{agent})Q(\mathbf{T}_{agent}, a)] - \lambda H(\pi_\theta)$$

7: Compute reward $r^{in}(\mathbf{T}_{agent}, a) = -\log(1 - D_w(\mathbf{T}_{agent}, a))$

8: **return** π

9: **end for**

Deductive Reasoning Classifier

As already introduced above, the optimal path is selected among the N paths given by IL and the paths produced by the inductive module by means of an ASP-based DR Classifier [39, 94, 163]. Among modern declarative formalisms, ASP, originally developed in the field of logic programming and non-monotonic reasoning, has become widely used in AI, and it is recognized as an effective tool for Knowledge Representation and Reasoning (KRR). This especially due to its high expressiveness and the ability to deal with incomplete knowledge, and also because of the availability of robust and efficient solvers; more details on ASP are given in Appendix 8.3.

A domain expert provided the knowledge that we encoded into a logic program consisting of rules and constraints and readable by the ASP solver. For each instance of the path planing problem, the ASP program is hence coupled to a proper representation of the pool of paths calculated by the IL model, $\{Q_k^{IL}\}$ (the “input”) and fed to the ASP

solver, thus choosing the best one, $\{Q^{ILDR}\}$. In the following, we provide more details about the encoded deductive path selector.

A path Q_x^{ILDR} cannot be selected if the maximum curvature c_{max} (Equation 3.7) measured for Q_x^{ILDR} is bigger than the k_{max} , as it can be seen in the following code snippet (In order to fully understand the syntax, we have provided some examples in Appendix 8.3):

```
:- choose( $Q_x^{ILDR}$ ), curvature( $c_{max}$ ), maxCurve( $Q_x^{ILDR}$ ,  $k_{max}$ ),  $k_{max} < c_{max}$ .
```

where the atom $\text{maxCurve}(Q_x^{ILDR}, c_{max})$ couples the path Q_x^{ILDR} with the maximum curvature c_{max} .

A path Q_x^{ILDR} must be discarded also if it approaches the sensitive structures at a minimum distance d_{min} (Equation 3.4) smaller than the radius ($r = \frac{d_{out}}{2}$) of the moving agent:

```
:- choose( $Q_x^{ILDR}$ ), radius( $r$ ), distObst( $Q_x^{ILDR}$ ,  $d_{min}$ ),  $d_{min} < r$ .
```

Along with these hard constraints, that make inappropriate paths to be discarded, we identified several criteria for expressing preferences among admissible ones. In particular, starting from the expert's knowledge, we designed some "soft constraints", and three corresponding weights ($w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$, $w \in \mathbb{R} > 0$) used either express preferences towards paths that feature minimum (or maximum) adherence to the criteria.

1. Minimization of the path length (d_{tot}):

```
#minimize{ $d_{dot} @ w_{d_{tot}}$ ,  $Q_x^{ILDR}$ : choose( $Q_x^{ILDR}$ ), length( $Q_x^{ILDR}$ ,  $d_{tot}$ )}
```

where d_{tot} is the length of path Q_x^{ILDR} .

2. Maximization of the distance from obstacles:

```
#maximize{ $d_{min} @ w_{d_{min}}$ ,  $Q_x^{ILDR}$ : choose( $Q_x^{ILDR}$ ), distObst( $Q_x^{ILDR}$ ,  $d_{min}$ )}
```

3. Minimization of the curvature of the moving agent:

```
#minimize{ $c_{max} @ w_{c_{max}}$ ,  $Q_x^{ILDR}$ : choose( $Q_x^{ILDR}$ ), maxCurve( $Q_x^{ILDR}$ ,  $c_{max}$ )}
```

Some additional insights are provided in description of Algorithm 3.

Algorithm 3 Classifying Path with ASP

Input: $\{Q_k^{IL}\}$, k_{max} , r , $w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$. **Output:** Q^{ILDR} .

```
1: for x in  $\{Q_k^{IL}\}$  do
2:   Compute  $d_{min_x}$ ,  $d_{tot_x}$  and  $c_{max_x}$ 
3:   if :- choose( $Q_x^{ILDR}$ ), curvature( $c_{max_x}$ ), maxCurve( $Q_x^{ILDR}$ ,  $K_{max_x}$ ),  $k_{max} < c_{max_x}$ . then
4:     if :- choose( $Q_x^{ILDR}$ ), radius( $r$ ), distObst( $Q_x^{ILDR}$ ,  $d_{min_x}$ ),  $d_{min} < r$ . then
5:       Sort  $\{Q^{ILDR}\}$  by:
6:         #min{ $d_{dot_x} @ w_{d_{tot}}$ .  $Q_x^{ILDR}$ : choose( $Q_x^{ILDR}$ ), length( $Q_x^{ILDR}$ ,  $d_{tot_x}$ )}.
7:         #max{ $d_{min_x} @ w_{d_{tot}}$ .  $Q_x^{ILDR}$ : choose( $Q_x^{ILDR}$ ), distObst( $Q_x^{ILDR}$ ,  $d_{min_x}$ )}.
8:         #min{ $c_{max_x} @ w_{c_{max}}$ .  $Q_x^{ILDR}$ : choose( $Q_x^{ILDR}$ ), maxCurve( $Q_x^{ILDR}$ ,  $c_{max_x}$ )}.
9:       end if
10:    end if
11: end for
return  $Q_0^{ILDR}$ 
```

It is worth noting that the purely declarative nature of ASP easily allows fine-tuning the desiderata by combining the constraints. Not only the system can easily be improved if new or more specific knowledge is available from the experts, but the user can change the behavior of the classifier at will

when in use. The provided interface gives the user the possibility to compose the desiderata and repeat this step multiple times, after the IL model has generated the output, $\{Q_k^{IL}\}$ (agnostic to the human-chosen desiderata), changing the inputs until she is satisfied with the obtained path. In this case, new weights and more (or less) restrictive constraints (i.e., increasing k_{max} or d_{out}) can be set. Furthermore, the user can decide to take into account all, some, or none of the encoded preferences, depending on the specific application; if she chooses to apply more than one of these rules, then also different priorities can be set, expressed by the weights. Hence, the capability to customize the set of rules and (hard/soft) constraints to use for each case study makes our tool highly flexible and generalizable.

3.4 EXPERIMENTAL PROTOCOL

Criteria for defining the “best” surgical path are several; their importance depends on the application at hand. In our experiments, performed in static simulated environments, we focused on DBS and CED, which are two relevant applications of steerable needles in keyhole neurosurgery.

3.4.1 Neurosurgical environment

3-D brain structures for CED and DBS environment were identified on two datasets: 1) Time-of-Flight (ToF) Magnetic Resonance (MR) for vessels visualisation, 2) T1 for brain cortex, skull surface, arterial blood vessels, ventricles and all the relevant deep grey matter structures visualisation, segmented through FreeSurfer Software [84] and 3-D Slicer [207].

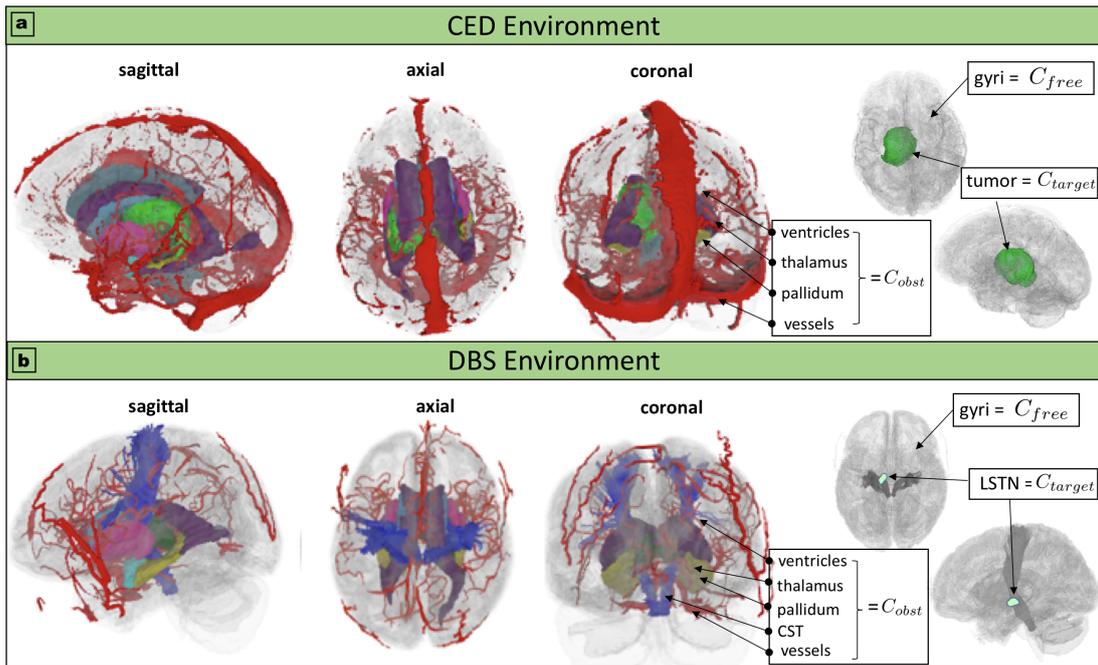


Figure 3.4: CED and DBS Environments. In (a) sagittal, axial and coronal view of the CED environment are reported. Defining accordingly the obstacle space ($C_{obst} = \{\text{ventricles, thalamus, pallidum and vessels}\}$), the free space ($C_{free} = \text{gyri}$) and the target space ($C_{target} = \text{tumor}$). In (b) sagittal, axial and coronal view of the DBS environment are reported. Defining accordingly the obstacle space ($C_{obst} = \{\text{ventricles, thalamus, pallidum, vessels and Corticospinal Tract (CST)}\}$), the free space ($C_{free} = \text{gyri}$) and the target space ($C_{target} = \text{LSTN}$).

- As reported in Figure 3.4a, in CED the target space is the tumor= C_{target} , surrounded by different essential structures (ventricles, thalamus, pallidum and vessels), that represent the obstacle space, C_{obst} , while gyri represent the free space, C_{free} .

- As reported in Figure 3.4, in DBS the target space is the LSTN= C_{target} , located in the central brain core. The obstacle space, C_{obst} , is represented by relevant structures (ventricles, thalamus, pallidum, vessels and CST), while gyri represent the free space, C_{free} .

3.4.2 Neurosurgical simulator

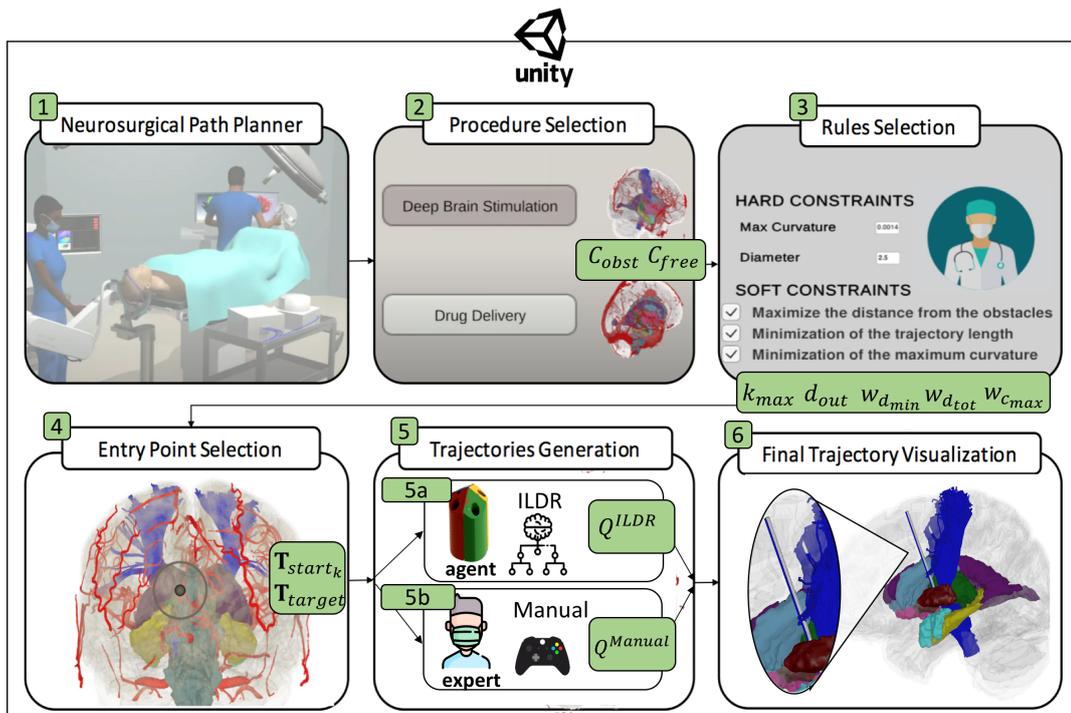


Figure 3.5: Neurosurgical simulator. It is possible to: (1) start a simulation, (2) select the procedure between the DBS and CED, (3) input the physical characteristics of the needle, d_{out} and k_{max} , and the parameters to be maximised or minimised in the selection of the best path, (4) select starting and target configurations, T_{start_k} and T_{target} , (5) perform manually the trajectories with a joystick controller or let the agent to perform trajectories autonomously with ILDR and finally (6) visualise all the generated trajectories.

A planning tool is implemented in 3D Unity [101] and integrated with ML-Agents Toolkit [131] that allows to visualize the 3D segmented risk structures of the brain of the patient derived from that data. We designed and developed a neurosurgical simulator, i.e. a "Brain Digital Twin", described in Figure 3.5 and shown in the animation (Appendix 8.7.1) to create the environment, collect manual demonstrations by the expert surgeon with a joystick controller (with a combination of the translation along the z axis and the rotation around the x and y axes) and train the moving agent. First of all the surgeon is asked to choose the curvature, k_{max} , and diameter, d_{out} , and the parameters he wants to prioritize in the selection of the best trajectory, $w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$, used either to minimize or to maximize the rule expression. The surgeon can then select a target configuration in the brain (T_{target}), e.g. on the tumor and N start configurations (T_{start_k}), on the brain cortex. Once the start and target configurations are defined, it is possible to proceed to either a manual or automatic, pre- or intra-operative procedure. The difference between these last two is the dynamism of the environment in representing the needle-tissue interaction and so the non-holonomic constraints that affect the behaviour and the needle movements. The surgeon can in fact proceed with a static environment or with a dynamic environment whose model is based on a Positions Based Dynamics simulator [238]⁵ used to emulate the brain tissues deformations in KN. The needle model is considered as particle system [193]. For the validation a pre-operative procedure with a

⁵Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., ... and De Momi, E. (2021). Position-

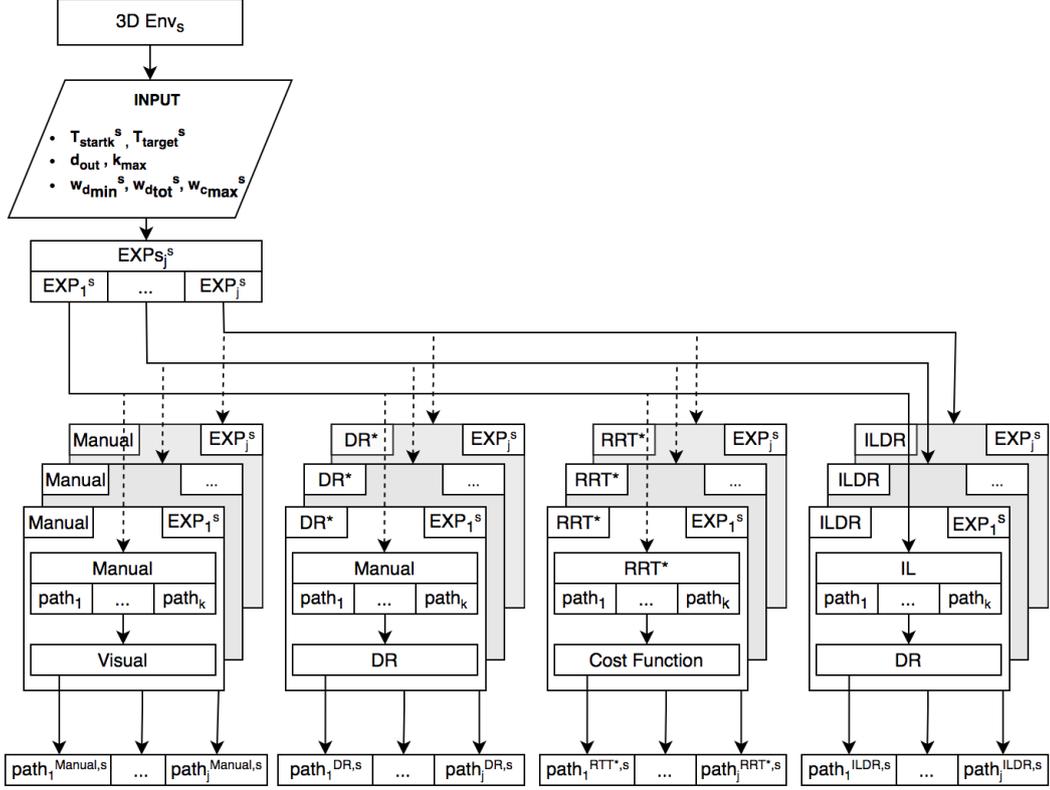


Figure 3.6: Schematic representation of the experimental protocol workflow Each environment Env_s (with $1 \leq s \leq 2$), takes in input the start configurations, $\mathbf{T}_{start_k}^s$ (with $1 \leq k \leq 10$), the target configuration, \mathbf{T}_{target}^s , the weights, $w_{c_{max}}^s, w_{d_{min}}^s, w_{d_{tot}}^s$ and the kinematic constraints of the moving agent, d_{out} and k_{max} . j experiments, EXP_j^s (with $1 \leq j \leq 5$), were conducted for each approach: Manual, DR, RRT* (only for DBS Environment) and ILDR. A pool of surgical paths, $\{path_k\}$ is generated, and the optimal one, $path_j^s$, is selected.

static environment was considered. It is assumed that the motion of the needle tip fully determines the motion of the needle ("follow-the-leader" deployment) with a combination of the translation along the z axis and the rotation around the x and y axes.

3.4.3 Experimental validation

The results' assessment for both scenarios, CED and DBS, is based on the comparison of the proposed method, ILDR, with the Manual and DR approaches. Moreover, in the DBS scenario, ILDR was tested against the RRT* algorithm.

As shown in Figure 3.6, an expert surgeon (age: 37, performed surgical procedures: 2440) was asked to select, for each environment Env_s , 10 desired start configurations, $\mathbf{T}_{start_k}^s$, on the brain cortex, a target configuration, \mathbf{T}_{target}^s , on the target space, C_{target} , and the weights, $w_{c_{max}}^s, w_{d_{min}}^s, w_{d_{tot}}^s$ for the rules prioritisation, reported for both scenarios in Table 3.3. j experiments, EXP_j^s (with $1 \leq j \leq 5$), were conducted for each one of the four approaches: Manual, DR, RRT* (only for DBS Environment) and ILDR.

- Manual approach: For each EXP_j^s , the surgeon was asked to generate a pool of surgical paths, $\{path_k\}$ (with $1 \leq k \leq 10$), and choose the optimal one, $path_j^{Manual,s}$, based on his expertise.

based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. IEEE Robotics and Automation Letters.

Table 3.3: Expert constraints and rules.

	$d_{out}(mm)$	$k_{max}(mm^{-1})$	$w_{d_{min}}$	$w_{d_{tot}}$	$w_{c_{max}}$
CED	2.5	0.014	9	6	6
DBS	2.5	0.014	1	4	2

- DR approach: For each EXP_j was considered the same pool of surgical paths generated in the manual approach, $\{path_k\}$, and the optimal one, $\{path_j^{DR,s}\}$, was selected with the DR classifier, using rules, weights and kinematic constraints given in input by the surgeon.
- RRT* approach: For each EXP_j the pool of paths, $\{path_k\}$, was generated with the RRT* algorithm. The optimal one, $\{path_j^{RRT^*,s}\}$, was selected with a Cost Function F_{cost} , to be minimised:

$$F_{cost}(\{path_k\}) = \begin{cases} \infty & \text{if } d_{min} \leq 0 \\ \infty & \text{if } c_{max} > k_{max} \\ w_{d_{min}} \frac{1}{d_{min}} + w_{d_{tot}} \frac{1}{d_{tot}} + w_{c_{max}} \frac{c_{max}}{k_{max}} & \text{otherwise} \end{cases} \quad (3.13)$$

Using rules, weights and kinematic constraints given in input by the surgeon. For more information on the implementation of this approach please refer to our previous work [240]⁶.

- ILDR approach: For each EXP_j the pool of paths, $\{path_k\}$, was generated with the IL model. The optimal one, $\{path_j^{ILDR,s}\}$, was selected with the DR classifier, using rules, weights and kinematic constraints given in input by the surgeon.

For each path, $\{path_k^s\}$, we calculated:

- The length (d_{tot}) of the path, as described in Equation 3.3;
- The minimum (d_{min}) and the mean (d_{avg}) distances of the path with respect to all the obstacles, as described in Equations 3.4 and 3.5;
- The maximum curvature (c_{max}) of the path, as described in Equation 3.7.

3.4.4 Hardware specification

Experiments were performed on a Linux machine equipped with a 6-core i7 CPU, 16GB of RAM and 1 NVIDIA Titan XP GPU with 12GB of VRAM.

3.4.5 IL Training Strategy

The training phase, for IL models, for each Environment, takes in input w start (\mathbf{T}_{start_w} , with $1 \leq w \leq 20$) and target z (\mathbf{T}_{target_z} , with $1 \leq z \leq 5$) configurations and y expert manual path (Q_y^{manual} , with $1 \leq z \leq 10$) for each start and target. The number of manual demonstrations $\|x = 1000$, (with $1 \leq x \leq \|w \times \|z \times \|y$), is obtained by combining the number of demonstrations ($\|y$) provided by the expert user for each couple of start and target ($\|w \times \|z$). At every episode randomly, a new \mathbf{q}_{start_w} was chosen among the available ones along with its relative \mathbf{q}_{target_z} . Table 3.4 presents the training parameters values referred to the CED and DBS models.

3.4.6 Statistical analysis

All the performance metrics (d_{tot} , d_{min} , d_{avg} and c_{max}), extracted from the path, were analysed employing Matlab (The MathWorks, Natick, Massachusetts, R2020a). Lilliefors test has been initially applied for data normality. Due to the non-normality of data distribution, pairwise comparison was performed with the Wilcoxon matched-pairs signed-rank test. Differences were considered statistically significant at p-value < 0.05 .

⁶Segato, A., Pieri, V., Favaro, A., Riva, M., Falini, A., De Momi, E., and Castellano, A. (2019). Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei. *Frontiers in Robotics and AI*, (6, 70).

Table 3.4: Training parameters

<i>Parameter</i>	<i>Value</i>	<i>Parameter</i>	<i>Value</i>
beta	5.0e-4	max steps	1.0e5
batch size	64	buffer size	256

3.5 RESULTS

3.5.1 Convection enhanced delivery

Figure 3.7a shows a comparison between Manual, DR and ILDR approaches in terms of d_{min} , d_{avg} , c_{max} and d_{tot} calculated over the best path of left hemisphere (for each approach, the criteria for the selection of the best path have been described in Section 3.4.3). DR approach keeps a greater d_{min} and d_{avg} from obstacles and a significantly lower d_{tot} (p-value < 0.01) of the path with respect to Manual approach following the rules dictated by the expert. ILDR approach keeps a significantly greater d_{avg} (p-value < 0.05) from obstacles and a significantly lower c_{max} (p-value < 0.01) and d_{tot} (p-value < 0.01) with respect to DR approach and Manual approach.

Figure 3.8a shows the visual comparison between one resulting path obtained with DR and ILDR approaches considering the previously mentioned optimisation criteria (safety, smoothness and path length), that result better for the ILDR approach even by visualization inspection.

3.5.2 Deep brain stimulation

Figure 3.7b shows a comparison between Manual, DR, RRT* and ILDR approaches in terms of d_{min} , d_{avg} , c_{max} and d_{tot} calculated over the best path of left hemisphere (for each approach, the criteria for the selection of the best path have been described in Section 3.4.3). DR approach keeps lower d_{tot} and c_{max} of the path than the Manual approach following the rules dictated by the expert who gives more importance to these two parameters in this case. While ILDR approach keeps a significantly greater d_{min} (p-value < 0.01) and d_{avg} (p-value < 0.01) from obstacles and a lower c_{max} and d_{tot} than the DR approach. The comparison between ILDR and RRT* approaches is showed in terms of d_{min} , d_{avg} , c_{max} and d_{tot} calculated over the best path of left hemisphere. ILDR approach keeps a significantly greater d_{min} (p-value < 0.0001) and d_{avg} (p-value < 0.0001) from obstacles and a lower c_{max} and a significantly lower d_{tot} (p-value < 0.01) than the RRT* approach.

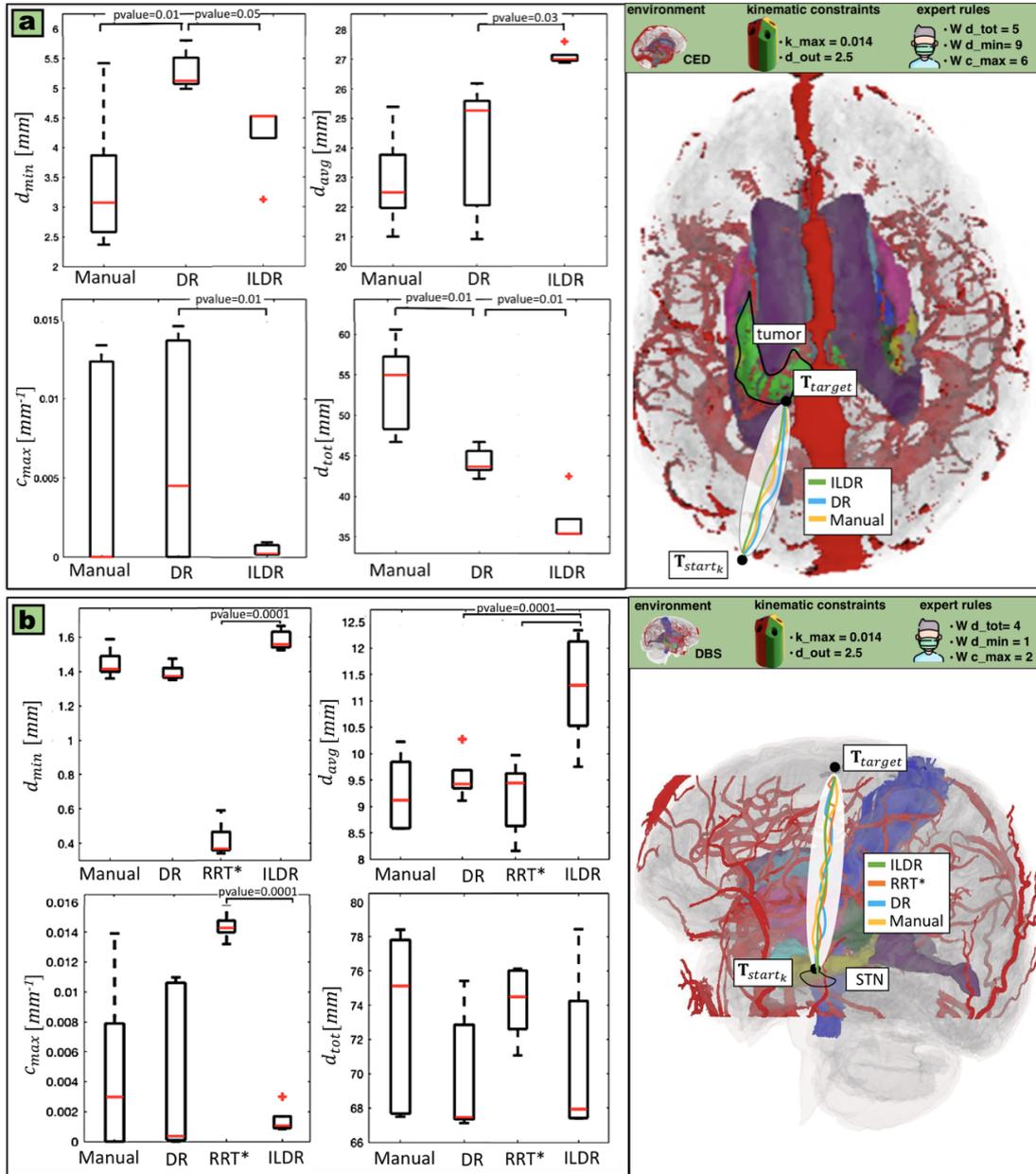
Figure 3.8b shows the visual comparison between one resulting path obtained with DR and ILDR approaches considering the previously mentioned optimisation criteria (safety, smoothness and path length), that result better for the ILDR approach even by visualization inspection.

3.5.3 Computational time

In Table 3.5, the computational time T for all the analysed approaches is reported. The ILDR approach is twice as fast compared to the Manual one and keeps much lower computational time than the state-of-the-art sampling-based method RRT*. Although this value is not essential for the proposed pre-operative procedure that is performed off-line, this demonstrates that the proposed method may be potentially applicable to an intra-operative procedure requiring a fast planning time.

Table 3.5: Results in term of computational time are shown.

Method	25th(s)	Median(s)	75th(s)
Manual	15.58	17.94	21.98
DR	11.25	15.00	32.69
RRT*	35.93	61.54	78.87
ILDR	8.02	8.06	8.10



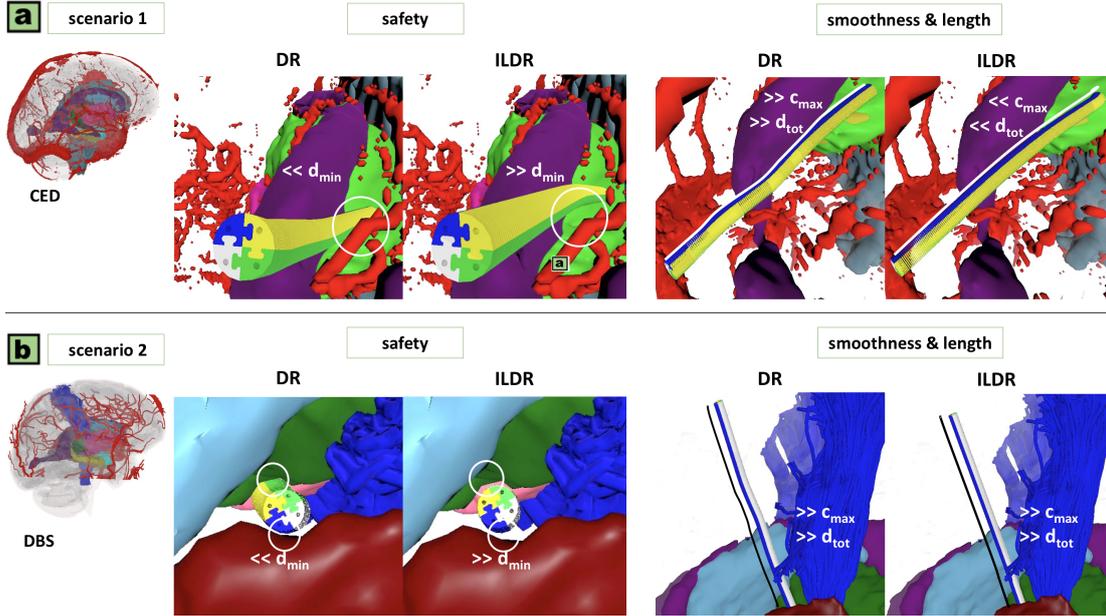


Figure 3.8: Comparison between DR and ILDR approaches. For both CED (scenario 1) and DBS (scenario 2) environment, one example of the obtained path is shown. In particular, a) shows a comparison between DR and ILDR approaches in the CED environment. b) shows a comparison between DR and ILDR in the DBS environment. The results for both considered scenarios show an increase in safety ($\gg d_{min}$) and smoothness ($\ll c_{max}$) and a reduction in length ($\ll d_{tot}$) for the proposed method (ILDR).

3.6 DISCUSSION AND CONCLUSION

The present work proposes a novel automatic path planning approach, called ILDR, for a moving agent in a complex environment. The complexity of the environment represents the worst-case scenario that grants the applicability and reliability of the method. In our experiments, ILDR performed better in terms of obstacle clearance and moving agent kinematic constraints compliance when tested against the Manual approach, the DR classification approach and the RRT* algorithm. By simultaneously optimising paths according to all the requested features, the proposed method outperforms state-of-the-art approaches in terms of path safety, path length, and computational time.

Our method succeeds in obtaining the optimal paths that can be followed to reach a specific target according to rules set by an expert. This approach allows to fully exploit an expert’s knowledge: he first performs the demonstrations used to train the GAIL model and then selects the constraints and their priorities, which ultimately lead to the choice of the best path with ASP.

It is worth noting that one of the main contributions of the present work consists in the integration of an inductive learning-based approach with a deductive reasoning-based approach. The inductive learning-based method allows the agent to learn the policy by a set of demonstrations provided by an expert, who can introduce in a path planning algorithm all his requirements and knowledge that cannot always be possible in graph- or sampling-based approaches unless additional optimisation steps are applied with additional computational time. Explicit programming cannot fully cover the complexity of the environment (represented by the human brain in this case, due to the presence of delicate and very complicated anatomical structure, narrow passages), the number of parameters and possible complications that have to be considered during the path planning. For this reason we implemented a DR classifier with a user interface, as described in the final part of Section 3.3.2, where the experts can express their individual preferences assigning different weights, thus creating a priority list for maintaining different path planning optimisation criteria (i.e., giving more priority to path safety than path length) while visualizing the trajectory and changing the criteria in real time. The DR method is implemented using ASP, that

allowed us enjoy several advantages. First of all, even if the whole machinery can be embedded into a graphical user interface for user's convenience, under the hood we are dealing with knowledge explicitly expressed via a declarative formalism: modifying and adapting the criteria for dropping unwanted paths and selecting the preferred one(s) is rather easy. Furthermore, as the specification are formally encoded, once the optimization criteria are well-established, we are ensured that the best path is actually chosen, and if more than one are present with the same "score", then picking one or the other is completely indifferent. If, for some reasons, this turns to be not the case, this means that the criteria should be modified, which, as already stated, can be easily done, especially given that the resulting framework allows one to straightforwardly experiments with this respect.

3.6.1 Clinical translation

As part of the EU's Horizon EDEN2020 project, the current study proposes a novel automatic planner for steerable needles in keyhole-neurosurgery. Given the environment of the brain, a surgeon-defined start, and a target, the proposed method can provide an optimal path, according to predefined features as insertion length, clearance from safety regions, as blood vessels and eloquent morpho-functional landmarks, and compliance to the needle's kinematics limits. It is intended to provide a state-of-the-art combined technology platform for minimally invasive surgery. When tested against the RRT* approach, the proposed method performed better in terms of path smoothness and clearance from safety regions, significantly decreasing the length and with a sensibly lower computational time. Accordingly to the possibility to perform curvilinear path for STN and tumor targeting, the proposed algorithm allows optimising the fundamental aspects of the DBS and CED and to maximising both the effectiveness and safety of the procedure.

3.6.2 Future directions

The proposed methodology favours high applicability and generalisability, as it could potentially be applied to different path planning problems. Future perspectives may include the exploitation of this automatic path planner method in many applications additional to keyhole neurosurgical procedures. The development of a surgical simulator defines an example of applications based on anatomical topology but not on anatomical dimensions. It is easy to see that different rules and constraints can be defined upon expert suggestion, thus making our methodology highly customisable and paving the way to extensions to additional 3-D complex environments, beyond brain surgery. Hence, this approach is widely application-independent and can be adapted to other use cases for path planning in a complex environment, where an expert has a crucial role. The application to a totally different context would require a thorough consultation with domain experts and the creation, if missing, of a specific simulation environment. Nonetheless, once the setup is created for a particular domain, as in the current study with brain surgery, it is easily applicable to different problems in such domain, by simply modifying or adding rules.

Position-based Dynamics Simulator of Brain Deformations for Intra-Operative Path Planning and Control in Keyhole Neurosurgery

Many tasks in robot-assisted surgery require planning and controlling manipulators' motions that interact with highly deformable objects. In this chapter we investigate **Hypothesis 2**. This study proposes a realistic, time-bounded simulator based on PBD simulation that mocks brain deformations due to catheter insertion for pre-operative path planning and intra-operative guidance in keyhole surgical procedures. It maximizes the probability of success, in achieving the planned target, by accounting for uncertainty in deformation models, noisy sensing, and unpredictable actuation. The PBD deformation parameters were initialized on a parallelepiped-shaped simulated phantom to obtain a reasonable starting guess for the brain white matter. They were calibrated by comparing the obtained displacements with deformation data for catheter insertion in a composite hydrogel phantom. Knowing the gray matter brain structures' different behaviors, the parameters were fine-tuned to obtain a generalized human brain model. The brain structures' average displacement was compared with values in the literature. The simulator's numerical model uses a novel approach with respect to the literature, and it has proved to be a close match with real brain deformations (**Hypothesis 2.1**) through validation using recorded deformation data of in-vivo animal trials with a mean mismatch of $4.73 \pm 2.15\%$. The stability, accuracy, and real-time performance make this model suitable for creating a dynamic environment for KN path planning, pre-operative path planning, and intra-operative guidance (**Hypothesis 2.2**). The application of the method resulted in a journal paper [1]¹.

4.1 INTRODUCTION

KN is a minimally invasive procedure performed to reach targets located deep inside the brain through a tiny hole in the skull, called "burr hole" or "keyhole" [130]. In KN, catheters are inserted into the brain

¹Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., ... and De Momi, E. (2021). Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. IEEE Robotics and Automation Letters.

for biopsy and therapies, as drug delivery or electrical stimulation following predefined paths to avoid damage to delicate structures.

Path planning plays an essential role in various fields of application and research. In its most general form, the task is to define a path for some moving entity between a start and a goal in an environment, *e.g.*, the trajectory of a catheter between keyhole and target location in an MR image. Nonetheless, the difference between the pre-operative planning in a static condition where no deformation is considered and the intra-operative dynamic condition where the anatomy deformations can significantly impact the planned trajectory's accuracy and relevance, since both the target organs and the obstacles such as vessels can move [109]. Due to the difficulty, tasks involving uncertainty and highly deformable objects are still routinely completed manually rather than automatically or semi-autonomously using robot assistance. Automating these tasks could increase productivity and improve outcomes by decreasing the time and costs associated with manual operation while simultaneously increasing accuracy and precision.

When considering the brain, there are three primary sources of deformation: the pulsation of brain vessels, the brain shifts [59, 112, 114, 223] (which could be caused by (i) Cerebrospinal Fluid (CSF) loss through the keyhole, resulting in surgical accuracy deterioration, (ii) pharmaceuticals administered during the intervention and (iii) tissue resection [108]), and the interaction between the catheter and the brain tissue. However, the pulsation of the brain vessels is not relevant in minimally invasive surgery [268].

4.2 RELATED WORK

Synthetic phantoms and virtual models of the human brain have been proposed as a tool to measure / compute these deformations and compensate for them. Such models are essential to reproduce geometries and structures and provide material formulations that could accurately replicate such a complex organ's mechanical behavior. Many works in the literature have studied brain deformations during neurosurgery [111, 122] and brain biomechanical characterization [85, 86, 160, 161]. Most of the physics-based brain deformation models are based on FEM but differ from each other in the choice of the constitutive equation of the model that defines the material chosen for the soft tissue simulation. In particular, [160] presents a comprehensive comparative study to help researchers select suitable materials for soft tissue simulation and their interactions with surgical instruments. The authors matched the stiffness of gelatin and Composite Hydrogel (MCH) phantoms to the one of a porcine brain and analyzed needle insertions in terms of insertion forces, displacements, and deformations. In [86], the authors perform a characterization of brain tissues under various test conditions (including humidity, temperature, strain rate); they present a rigorous experimental investigation of the human brain's mechanical properties through ex-vivo tests, covering both gray and white matter. Brain material characterization is instrumental in providing inputs to a mathematical model describing brain deformation during a surgical procedure. The work of Forte et al. [85] focuses on developing an accurate numerical model for predicting brain displacement during procedures and employs a tissue mimic of MCH to capture human tissue complexity. This mimic is designed to reproduce the dynamic mechanical behavior in a range of deformation rates suitable for surgical procedures. The investigation supports the proposal of a hybrid formulation of porous-hyper-viscoelastic material for brain displacement simulation. Further work of Leibinger et al. [161] measures internal displacements and strains in three dimensions within a soft tissue phantom at the needle interface, providing a biologically inspired solution causing significantly fewer damages to surrounding tissues.

As an alternative for the phantoms and models mentioned above, neurosurgical simulators based on FEM brain tissue deformation model have been developed. Some of these are based on the optimization of the implicit Euler method, such as the one presented in [120]. Others like the NeuroTouch [61], a commercial neurosurgical simulator, use the explicit time integration scheme, require small time steps to keep the computation stable. Implicit integration schemes [13], on the other hand, have the advantage of unconditional stability but are much more computationally expensive [30].

Recently, PBD [186] has gained considerable popularity due to its simplicity, high efficiency, unconditional stability, and real-time performance [28]. PBD approach eliminates the overshooting problem of equilibrium configurations achievement in force-based methods and simplifies the implementation process. Although PBD is not as accurate as force-based methods, its efficiency and controllability outperform those simulating medical procedures by far while providing visually plausible results [48].

4.3 OBJECTIVE

To the authors' knowledge, this Chapter presents the first simulator of realistic brain tissue displacement using PBD modeling designed for catheter insertion during KN. We further validate it using recorded brain deformation on an in-vivo animal model. The simulator developed in Unity with NVIDIA FleX [6] backend provides a suitable system for AI framework to train autonomous control and path planners. Moreover, we make our simulator publicly available on [GitLab](#) to encourage the adoption of the method. The simulator is shown in the animation (Appendix 8.7.2)

4.4 MATERIALS and METHODS

4.4.1 Position-based dynamics approach

PBD is a simulation approach that computes the time evolution of a dynamic system by directly updating positions, as first described by Müller et al. in [188]. Simulated objects are discretized as clusters of particles described by their positions \mathbf{p}_i and velocities \mathbf{v}_i , subject to a set of J positional constraints $C_j(\mathbf{p}_i, \dots, \mathbf{p}_n) \succ 0$ (symbol \succ denotes either $=$ or \geq). In the PBD approach, deformation computation becomes a constrained optimization problem. Soft bodies simulation behavior and performance are not only influenced by the relative position, dimension, and the number of particles in space but also by the constraints acting among particles. For example, large deformations of soft bodies are usually achieved by defining positional constraints among adjacent particles' rigid clusters. This kind of constraint is called *region-based shape matching*, as described in [30]. The method is explained comprehensively in Appendix 8.3. A realistic elastic behavior is obtained by appropriately selecting cluster parameters; hence, we focused on the initialization of the PBD parameters defining clusters for all the structures present in the scene: *cluster spacing* (i.e., the distance between adjacent clusters), *cluster radius* (i.e., the radius of each cluster region) and *cluster stiffness* (i.e., the extent to which adjacent clusters are constrained to each other). Other parameters of the PBD model can impact soft body behavior and require tuning according to [186]; therefore, to simplify the model, parameter values were kept fixed as in [48]. The NVIDIA FleX is a particle-based simulation library for real-time, realistic visual effects. The objects in the framework are modeled as clusters of particles connected by various constraints, broadly based on PBD. Since NVIDIA FleX does not directly support the Unity Game Engine, we used uFleX, a Unity asset integrated low-level Flex native library.

4.4.2 Case study: keyhole neurosurgery

Figure 4.1 summarizes the workflow of our experimental study made of two calibration phases, one for the white matter performed with an in-vitro experiment on a phantom and one for deep gray matter performed with an in-silico experiment on a human brain dataset. The last phase is the validation conducted through in-vivo experiments on ovine brains. Assuming that the ovine brain is quite similar to the human brain except for proportion. The different steps are detailed below:

Catheter

This project was developed in the context of the EU's RIA Horizon project codename EDEN2020, aiming to advance the current state of the art in neurosurgical technology. The catheter model used in this work is a Programmable Bevel-Tip Needle (PBN) [43]. The catheter has an outer diameter of 2.5 mm, as shown in Figure 4.2A.

Data acquisition and processing

Various models were created to carry out our simulations:

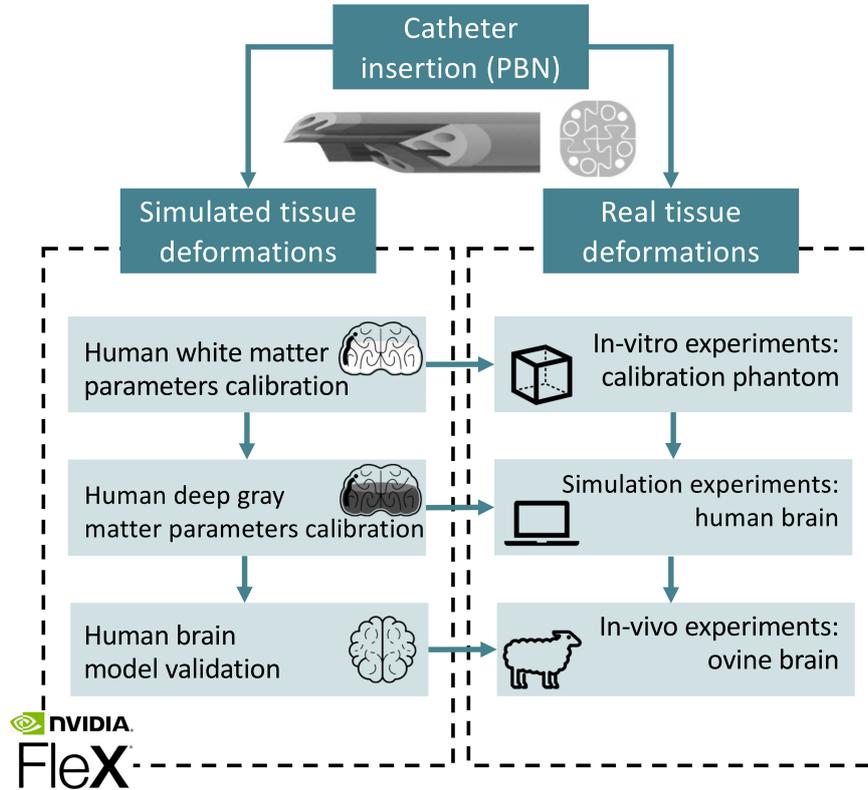


Figure 4.1: Workflow of the experimental study.

Calibration brain phantom We created a parallelepiped-shaped phantom ($50 \times 17.5 \times 17.5$ mm) using Blender [2] to initialize the PBD simulation parameters for the brain white matter. Its dimensions have been chosen to match the experimental setup of [161], shown in Figure 4.2B. Figure 4.2C shows the obtained simulated phantom and catheter.

Human brain High-resolution MR images were acquired on one healthy subject (male, aged 38 yo). The research ethical committee of Vita-Salute San Raffaele University and IRCCS San Raffaele Scientific Institute, Milan, Italy, approved the study (ethical approval n. 80/INT/2016), and the subject provided signed informed consent before MR acquisition. The MR protocol included:

- a 3D T1-weighted sagittal Fast-Field-Rcho (FFE) with selective water excitation (Proset technique) [103] acquired with the following parameters: repetition time/echo time [TR/TE] 12/5.9 ms; flip angle, 8° ; acquisition matrix, 320×299 ; voxel size, $0.8 \times 0.8 \times 0.8$ mm; thickness, $0.8/0$ mm gap; SENSitivity-Encoding [SENSE] reduction factor, $R=2$; 236 slices; acquisition time, 5 min 19 s;
- a 3D high-resolution ToF MR angiography (MRA) acquisition to visualize flow within the arterial vessels, acquired with the following parameters: TR/TE 23/3.45 ms; flip angle, 18° ; acquisition matrix, 500×399 ; acquired voxel size, $0.4 \times 0.5 \times 0.9$ mm; reconstructed voxel size, $0.3 \times 0.3 \times 0.45$ mm; thickness, $0.45/-0.45$ mm gap; SENSE factor, $R = 2$; 210 slices; acquisition time, 8 min 33 s.

On the normalized 3D T1-weighted images, we segmented the amygdala, brain stem, caudate, cerebellum, cerebral cortex (gyri and sulci), hippocampus, pallidum, putamen, thalamus, and ventricles employing FreeSurfer Software [3, 83]. Additionally, we segmented the arterial blood vessels from the 3D high-resolution ToF-MRA using 3D Slicer© platform [5, 206] to obtain the final set of brain structures ($BS = bs_1, \dots, bs_m$, with $m = \#BS$). Figure 4.3A shows the final result.

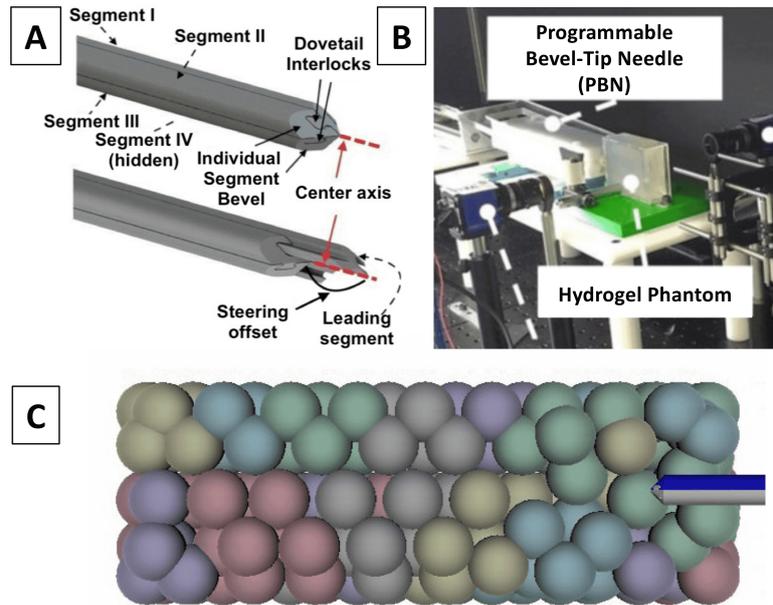


Figure 4.2: Experimental Setup. (A) Illustration of the PBN catheter showing the four segments interlocked by means of Dovetail joints. The four segments can slide on each other along the center axis, creating an offset that allows the steering of the catheter (courtesy of Leibinger, 2016). (B) Experimental setup with the hydrogel phantom and the needle presented in [161]. (C) Basic deformable parallelepiped-shaped phantom used for the initial setting of PBD simulation parameters for the brain white matter and the 3D model of the reconstructed catheter with an outer diameter of 2.5 mm.

Ovine brain During the experiments carried out in Università degli Studi di Milano Ospedale Veterinario Universitario within the EU’s Horizon EDEN2020 project, five animals underwent MR at 1.5 T (Achieva, Philips Healthcare) before and after the procedure (PBN insertion), including CT (Figure 4.4A) and conventional 3DT1 FFE (TR/TE:25/5ms) to guide the neurosurgical planning (Figure 4.4B). We segmented the brain on the normalized 3D T1-weighted images, as shown in Figure 4.4C. All animals were treated in accordance with the European Communities Council directive (2010/63/EU), to the laws and regulations on animal welfare enclosed in D.L.G.S. 26/2014. Ethical approval for this study was obtained by the Italian Health Department with authorization n 635/2017.

Simulation parameters

First, we calibrated the simulation parameters for brain white and deep gray matter with an empirical procedure based on calibration experiments and observations of the resulting visual and numerical deformations. These parameters were then validated by comparing the obtained deformations with those in in-vivo tests on the ovine brain. Catheter-tissue interaction is modeled as a contact problem, handled by the Unity engine’s default collision detection and response implementation.

White matter parameters calibration We created a scene in Unity containing the previously mentioned parallelepiped-shaped simulated phantom. To describe the entire model as deformable, we constrain all the particles to fall within at least one cluster by imposing *cluster radius* to be at least half of *cluster spacing* (set to [5, 35] mm), as proposed in [252]. Consequently, *cluster radius* is restricted to the range [2.5, 35] mm to keep the simulation stable; the upper limit is coincident with the *cluster spacing* one to maintain an overlap between the various clusters. Hence, the final deformation is smooth without burdening the computational cost. Conversely, *cluster stiffness* is left free to vary within the entire acceptable range [0,1] with a trial and error process to find the most suitable value. Indeed, the

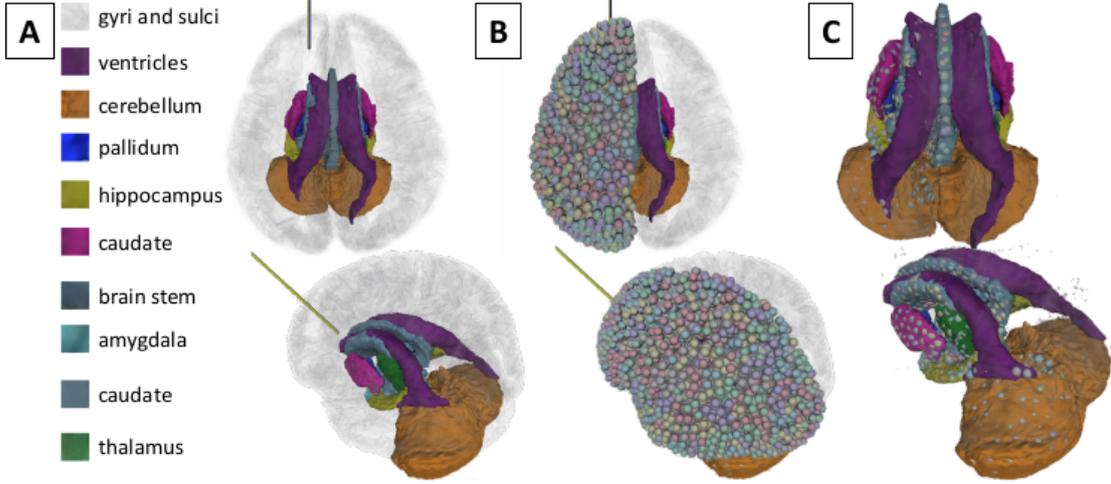


Figure 4.3: Human Brain Dataset: (A) Representation of the brain structures segmented on the normalized 3D T1-weighted images and the 3D high-resolution ToF-MRA. (B) Particle model of brain white matter. (C) Particle model of deep gray matter structures.

model's stiffness does not depend only on the defined stiffness parameter but also on the time step, number of solver iterations, number of clusters, and shape-matching constraints. This involves considering the stiffness parameters in relation to the entirety of the system and highlights how the PBD parameters, in general, do not have a direct physical meaning [48].

To initialize the parameters, we performed 8 insertions of the catheter inside the simulated phantom, $insertion^k$ (with $1 \leq k \leq 8$). Following the experimental work of Leibinger et al. [161], we have set the following conditions: the catheter is inserted in a straight trajectory with a constant speed of 0.5 mm/s , without offset at the tip. The temporal evolution of all particles' average displacement was evaluated at 31.5 mm depth along the insertion axis. Last, the sampling of the displacement was calculated with a fixed space interval of 3.4 mm .

As result, at each frame f of the simulation the following metrics were calculated:

- The penetration depth (Δ_{depth_f}) of the k insertion:

$$\Delta_{depth_f}^k = \|\mathbf{q}_{init}^k - \mathbf{q}_f^k\| \quad (4.1)$$

where \mathbf{q}_{init}^k is the catheter's starting x, y, z position, \mathbf{q}_f^k is the position of the tip of the catheter at frame f , and $\|\cdot\|$ is the Euclidean distance.

For each interval of 3.4 mm we calculated:

- The average displacement (Δ_{disp_f}) of the N particles of the parallelepiped-shaped phantom at the depth of 31.4 mm :

$$\Delta_{disp_f}^k = \frac{1}{N} \sum \|\mathbf{p}_{init}^k - \mathbf{p}_f^k\| \quad (4.2)$$

where $\mathbf{p}_{i,init}^k$ represents the starting position of the N particles, and $\mathbf{p}_{i,f}^k$ their position at frame f .

- The displacement ($\Delta_{centerDisp_f}$) of phantom particles Center of Mass (CoM):

$$\Delta_{centerDisp_f} = \|\mathbf{c}_{init} - \mathbf{c}_f\| \quad (4.3)$$

where \mathbf{c}_{init}^k represents the starting position of the particles CoM, and \mathbf{c}_f their position at frame f .

We averaged the measured variables across the 8 experiments. Subsequently, starting from the displacement results reported in [161], we fine-tuned the PBD model parameters to achieve comparable values for the brain white matter.

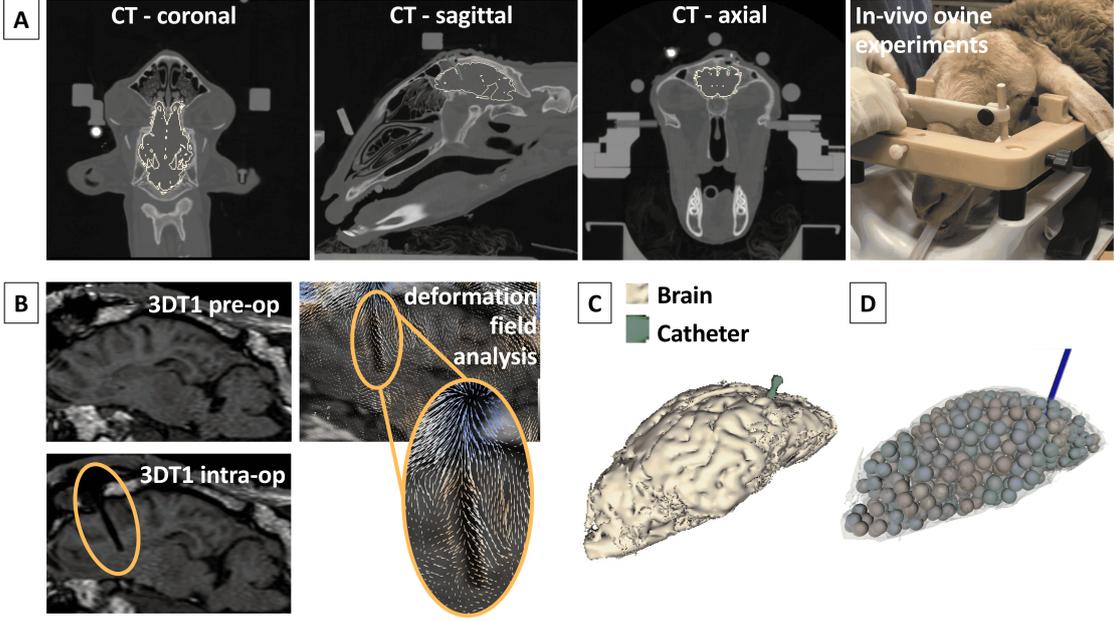


Figure 4.4: Ovine Brain Dataset (A) Coronal, sagittal and axial view of post-operative CT on ovine brain and in-vivo experiment. (B) Ovine brain pre-operative and intra-operative 3DT1 and deformation field analysis. (C) 3D segmented dataset for the ovine brain. (D) Particle model for the ovine brain.

Deep gray matter parameters calibration We tuned the gray matter parameters following the procedure of parameter initialization of the white matter and knowing the differences in the various brain structures' behaviors [105, 126, 127, 179, 189, 285]. White matter presents a real shear modulus higher than gray matter. The same applies to the viscosity, represented by the imaginary shear modulus, leading to an isotropic complex shear modulus of white matter around 1.25 kPa [105], and a higher stiffness with respect to gray matter. Other brain components were analyzed through a 3D multi-frequency Magnetic Resonance Elastography (MRE), and values of the complex shear modulus between 1.058 kPa for the thalamus and 0.649 kPa for the caudate were observed. The stiffness values for the cerebellum and the brainstem are slightly higher, as measured by [59]. According to this possible subdivision of the brain into areas with similar stiffness present in the literature, we tuned the FleX asset parameters of the NVIDIA FleX framework to give a simulation behavior as close as possible to reality.

We created a new Unity scene containing all the previously segmented brain structures to validate the chosen simulation parameters. Figure 4.3B shows the application of the FleX model to gyri and sulci; on the other hand, in Figure 4.3C, it is possible to observe the different brain structures with their flexible particle system. We selected one desired starting position for the catheter, \mathbf{q}_{init} , on the brain cortex and a target position, \mathbf{q}_{goal} , for the left hemisphere and to manually perform an insertion from \mathbf{q}_{init} towards \mathbf{q}_{goal} using a hand controller. For each *insertion*^k (with $1 \leq k \leq 8$), we evaluated the average displacement of each brain structure's centers.

For each frame f of the simulation we calculated:

- The penetration depth (Δ_{depth_f}) of the k insertion, as defined in Equation 4.1;
- The displacement (Δ_{disp_f}) of the CoM c of each brain structure ($\{BS\}$):

$$\Delta_{disp_{i,f}}^k = \|\mathbf{c}_{i,init}^k - \mathbf{c}_{i,f}^k\| \quad (4.4)$$

where $\mathbf{c}_{i,init}^k$ represents the CoM position of the i bs when the PBN tip is at the start position \mathbf{q}_{init}^k ; conversely, $\mathbf{c}_{i,f}^k$ represent the CoM position at frame f .

Subsequently, we computed the mean displacement averaging the displacements of the CoM in all the frames f , all structures bs_i , and all the experiments k .

Brain deformation model validation We computed the displacement of points surrounding the catheter inserted in one ovine brain to validate the calibration parameters. First, using ImFusion Suite [4], we rigidly registered the post-operative MR (after catheter insertion) to the pre-operative one. After that, we performed a non-rigid registration using the Demons algorithm [203] and normalized cross-correlation (NCC) as a similarity metric. To obtain points for evaluation, we aligned the axes to the catheter cylinder on the post-operative image. Subsequently, we extracted the displacement field; on each plane, we picked $z = 1, \dots, 5$ equally distanced points on each side $s = 1, \dots, 4$ of the hole created from the catheter insertion. The deformation field is sampled from the perimeter of the cylinder hole created by the insertion, visible on the MR. To select points on the perimeter, we set the viewing axes so that the z -axis is in line with the insertion direction, shown in Figure 4.5A. Next, we selected 5 points on each side of the cylinder hole in $x - z$ plane (green points in Figure 4.5B) and $y - z$ plane (orange points in Figure 4.5C) resulting in 20 points in total. After that, we simulated the segmented ovine brain into the Unity scene using the particle model (Figure 4.4D). $\forall z, s$ we measured the displacements $\Delta disp_f^k$ (Eq. 4.2).

Mismatch and error The reference displacement and the calculated one of the mean displacements of each z point in each side s were compared obtaining the mean displacement in the area around the catheter insertion. In order to objectively compare the results obtained in the validation phase, the $Mismatch[mm]$ is calculated by computing the mean squared errors of the average displacements, according to this equation:

$$Mismatch = \frac{\sum_{i=1}^n (disp_{reference_i} - disp_{calculated_i})^2}{n} \quad (4.5)$$

obtaining the $error[\%]$, according to this equation:

$$error = \frac{Mismatch}{max(displ)} \quad (4.6)$$

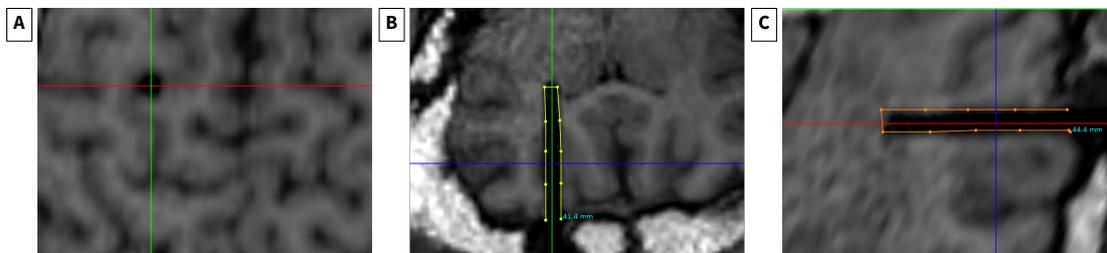


Figure 4.5: Needle Deformation Field of in-vivo Insertion (A) To sample the deformation field, first we set the viewing axes so that the z -axis is roughly parallel to the catheter insertion direction. (B) Then we take our samples from the catheter hole boundary in the x - z plane (green points) (C) and y - z plane (orange point).

4.4.3 Computational setup

The simulation environment was developed and tested in Unity 2019.2 using NVIDIA FleX on a workstation equipped with an Intel Core i7 6800k processor, 32 GB RAM, Titan Xp GPU by NVIDIA Corporation with CUDA 10.1.

4.5 RESULTS

4.5.1 White matter calibration

This section reports the results related to the calibration of the white matter simulation parameters and their tuning. The curve shown in Figure 4.6A represents the temporal evolution of the average displace-

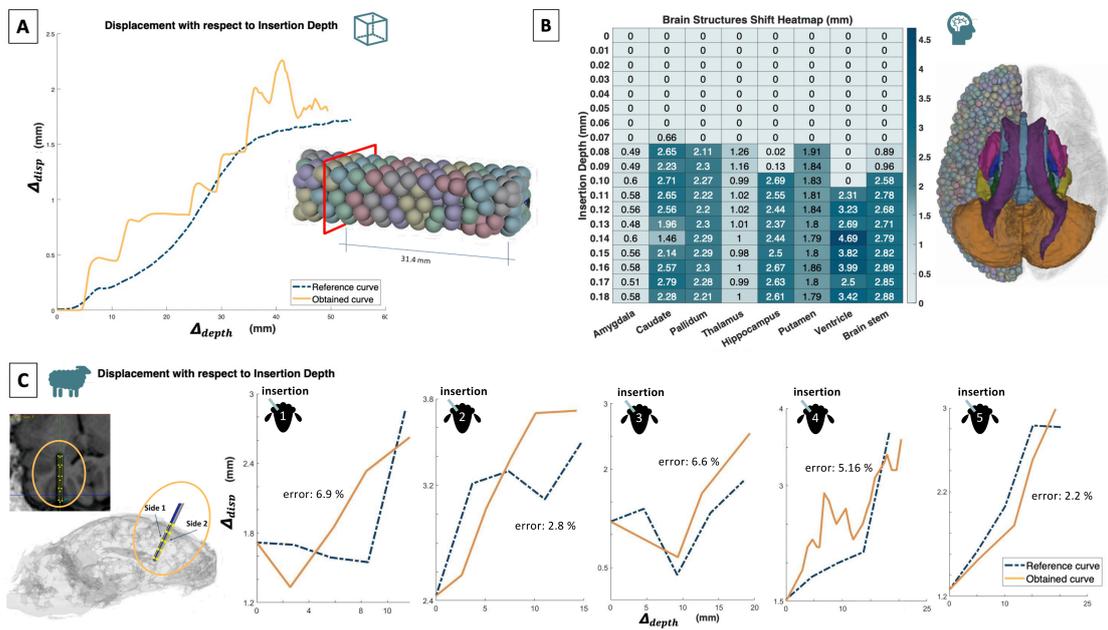


Figure 4.6: Needle Displacement Results. (A) Displacement obtained for white matter on the calibration parallelepiped and comparison with the one obtained in [161]. (B) Heatmap representing the displacement of the different deep gray matter structures at various insertion depths (the maximum insertion depth reached is considered enough to be clinically relevant). (C) Five experiments reporting the deformation of the simulated ovine model (yellow) and the relative in-vivo reference (blue) as a function of the insertion depth. The obtained errors are considered acceptable as they are under a threshold fixed to 10%.

Table 4.1: *Adimensional parameters set for different brain structures as a result of the fine-tuning procedure*

<i>Flex Objects</i>	<i>Particle Spacing</i>	<i>Cluster Spacing and Radius</i>	<i>Cluster Stiffness</i>	<i>Link Radius</i>	<i>Link Stiffness</i>
Amygdala	0.005	0.005	0.0005	0.005	0.001
Caudate	0.005	0.005	0.0005	0.005	0.001
Pallidum	0.005	0.006	0.0005	0.005	0.001
Hippocampus	0.005	0.005	0.0005	0.005	0.001
Thalamus	0.005	0.005	0.001	0.005	0.001
Ventricle	0.0065	0.0065	0.001	0.005	0.001
Putamen	0.0065	0.0065	0.0005	0.005	0.001
Gyri	0.0066	0.0066	0.002	0.009	0.001
Sulci	0.007	0.007	0.001	0.005	0.001
Brain Stem	0.007	0.007	0.0005	0.005	0.001
Cerebellum	0.006	0.006	0.002	0.0065	0.001

ment of all the particles situated at a depth of 31.4 *mm* within the tissue phantom. We computed the displacements at a fixed interval of 0.34 *mm* while inserting the catheter. We compared this temporal evolution with the one obtained in the experiment presented in [161]. We observed that parameter tuning led to a deformation range comparable with the reference one, with an *error* equal to 6.16% (considered acceptable under a threshold fixed to 10%).

4.5.2 Deep gray matter calibration

Tab. 4.1 reports the optimal values for cluster spacing, radius, and stiffness parameters obtained with the fine-tuning procedure for the various brain structures. Figure 4.6B summarizes the effect of the catheter insertion in the simulated deep gray matter. On the *x*-axis, we reported the various brain structures, whereas the different insertion depths (*i.e.*, the increment on the *y*-direction starting from the anterior limit of the brain) are represented on the *y*-axis. Each cell of the grid provides the value of the particular brain structure center’s displacement at the relative insertion depth. The shift of the structures obtained with the calibration reflects the expected values reported in [161]. This suggests that a proper tuning of the tissue deformation parameters has been performed.

4.5.3 Brain deformation model validation

We confronted the displacement obtained during five different catheter insertions simulated on our ovine model and the ones obtained in the in-vivo experiments for the model validation. Figure 4.6C shows this comparison considering the insertion depth. It can be noticed that the deformation ranges are comparable, as shown by the mean mismatch of $4.73 \pm 2.15\%$ obtained by computing the mean squared errors of the average displacements (with an average re-planning latency of 0.02 *sec*). Hence, the model can be applied to different types of datasets and represents a good surrogate for the modelization of deformation induced by the catheter insertion in different cerebral areas.

4.6 DISCUSSION

This Chapter presents a model able to account for the brain’s dynamic behavior during keyhole neurosurgery. It is established that the accuracy of FEM results would benefit from using a higher mesh resolution, but this would come at the expense of degradation in computation time [287]. Conversely, the mesh-free PBD approach has the advantage of avoiding the time-consuming generation of high-quality meshes, which represents the major bottleneck in FEM simulations (especially in those involving large deformations). Since we target a patient-specific context, this represents a relevant advantage because the mesh would have to be constructed every time for each patient. Furthermore, thanks to its direct manipulation of positions, the PBD approach can efficiently handle collision constraints. Probe-tissue

interaction can thus be effectively treated as a collision problem, thus allowing to deal with any input probe position without requiring the explicit definition of the contacting surface. The same does not apply to FEM simulations, where the enforcement of contact constraints would introduce degradation of the performances and stability issues. The proposed approach relies on the *region-based shape matching* constraint to model large deformations of soft tissues. This implementation's main drawbacks are the dependence of the deformable behavior on time step size and iteration count and the fact that PBD parameters do not have a direct physical meaning. The preliminary calibration of the main deformation parameters on a distinct geometry was performed offline to find reasonable initialization values. Before applying the deformation model, simulation parameters were refined with a fine-tuning procedure on the final structures of interest to improve parameter values to describe patient-specific features. Indeed, the PBD approach is controlled by a high number of parameters that, if fine-tuned, can provide a more realistic simulation. We have demonstrated that by tuning a subset of parameters on a MCH phantom can lead to matching real deformations. We obtained a PBD model able to replicate the typical behavior of the brain. This is demonstrated by comparing the deformations produced by our model applied to the ovine brain and the corresponding real deformations, obtaining an average error of 4.73%. This fine-tuning process of all parameters is out of the scope of this work, and it will be explored in future development. We expect that such parameters will be able to model another patient's brain by simply replacing the tuned meshes in the publicly available project with the new specific brain anatomical models. We do not expect that the chosen parameters (given in Table 4.1) will be able to model other parts of the brain that were not tuned (*e.g.*, subthalamic nucleus). However, with proper tuning of their values, it will be possible to account for intra-brain variability. Overall, the stability and latency of the proposed are satisfactory. To improve the stability of the simulated environment, it would be optimal to increment the total number of particles (thus increasing the degrees of freedom) or force a particle to lie at the real fiducials' exact location.

4.7 CONCLUSION

Through this work, we have presented a realistic, time-bounded simulator that mocks brain deformations during keyhole surgical procedures, where a catheter/needle is inserted into the brain. The simulator's numerical model has used a novel approach with respect to the literature, and it has proved to be a close match with real brain deformations through validation using recorded deformation data of in-vivo animal trials. The simulator represents a critical component for the development and training of AI systems such as autonomous control or intra-operative path planners in the context of KN, which will be explored and presented in future works. Future work could also extend the study to tumour tissue modelling, which we preferred to avoid due to the complexity and heterogeneity in stiffness depending on tumour type and location (*e.g.* adherence of tumour to the surrounding normal brain). In any case, there are many studies in the SOTA [113,204] that could provide valid parameter initialisations, provided that real needle insertions can then be performed in the tumours for the fine-tuning and validation of these parameters.

Deep and Inverse Reinforcement Learning Intra-operative Path Planning for Steerable Needle

This Chapter presents a safe and effective keyhole neurosurgery intra-operative planning framework for flexible neurosurgical robots. The framework is intended to support neurosurgeons during the intra-operative procedure to react to a dynamic environment. In this chapter we investigate **Hypothesis 3**, proposing a system that integrates inverse reinforcement learning path planning algorithm combined with 1) a pre-operative path planning framework for fast and intuitive user interaction, 2) a realistic, time-bounded simulator based on Position-based Dynamics (PBD) simulation that mocks brain deformations due to catheter insertion and 3) a simulated robotic system. Simulation results performed on a human brain dataset show that the inverse reinforcement learning intra-operative planning method can guide a steerable needle with bounded curvature to a predefined target pose with an average targeting error of 1.34 ± 0.52 (25th=1.02, 75th=1.36) mm in position and 3.16 ± 1.06 (25th=2, 75th=4.94) degrees in orientation under a deformable simulated environment, with a re-planning time of 0.02 sec and a success rate of 100% (**Hypothesis 3.1**). With this work, we demonstrate that the presented intra-operative steerable needle path planner is able to avoid anatomical obstacles while optimising surgical criteria. The results demonstrate that the proposed method is fast and can securely steer flexible needles with high accuracy and robustness (**Hypothesis 3.2**). The application of the method resulted in a journal paper [4]¹ currently under review.

5.1 INTRODUCTION

The spread of the KN procedure is due to its capability to minimize patient trauma, risk of infection and recovery time [8]. These features make KN applicable for a wide range of patients who are not suitable for open surgery [107]. However, performing these procedures using traditional rigid needles may result in the impossibility to reach targets without colliding with anatomical obstacles(e.g., bones) or sensitive anatomical tissues(e.g., vital organs and vessels).

¹Segato, A., Di Marzo, M., Zucchelli, S., Galvan, S., Secoli, R., and De Momi, E. Inverse Reinforcement Learning Intra-operative Path Planning for Steerable Needle. IEEE Transactions on Biomedical Engineering (T-BME) (resubmitted after major revisions on November 2021)

Table 5.1: *State of the art.*

Authors	Method	Algorithm	Intra operative	Dynamic env	Tissue modelling	Needle	Clinical aim
[191] Niyaz, 2018	GB	LPA*	YES	NO	NO	CTN	KN
[240] Segato, 2019	SB	RRT*	NO	NO	NO	PBN	DBS
[119] Hong, 2019	SB	RRT*	NO	NO	NO	MFN	DBS
[79] Favaro, 2018	SB	BIT*	NO	NO	NO	PBN	CED
[45] Caborni, 2012	SB	RG-RRT	YES	NO	NO	PBN	KN
[78] Fauser, 2018	SB	RRT-Con	YES	NO	NO	FBN	KN
[210] Pinzi, 2019	SB	AHFT	NO	NO	NO	PBN	CED
[209] Pinzi, 2021	SB	EBB	YES	YES	YES	PBN	CED
[25] Bano, 2011	MB	GBO	NO	NO	NO	PBN	KN
[146] Ko, 2012	MB	GBO	YES	NO	NO	PBN	KN
[273] Wankhede, 2019	MB	heuristic	YES	NO	NO	FBN	KN
[242] Segato, 2020	LB	DQN	NO	NO	NO	PBN	DBS
Presented approach	LB	IRL	YES	YES	YES	PBN	KN

Over the last two decades, different research groups have focused their efforts on the development of needles able to steer inside the tissue. These needles can perform curvilinear trajectories planned to maximize the distance from sensitive anatomical structures to be avoided and reach targets otherwise inaccessible via rectilinear insertion paths [42]. Differently from conventional needles, for which the insertion path can be planned and performed by the clinician on the basis of the target location and the patient anatomy, the complex kinematics of steerable needles make the path planning unbearable requiring the aid of automatic or semiautomatic path planning solutions. The intra-operative accuracy and robustness directly contribute to treatment outcomes and postoperative recovery.

Hence, there is a need to develop a robust path planning framework to further improve the robustness in obstacle avoidance and risk management under the complexity and uncertainty in needle–tissue interaction and needle/tissue deformation that facilitate large targeting errors.

5.2 RELATED WORK

A number of systems for planning KN interventions have been proposed to assist the surgeon in the planning process. The main idea of these studies is to describe curvilinear trajectory planning methods for risk reduction. Table 5.1 summarizes and compares the here presented related work in terms of methodology, intra-operative applicability (some can only be used in pre-operative phases as they do not have a re-planning strategy), type of validation (presence of a dynamic environment), validation reliability (presence of brain tissue modelling), type of steerable needle used and the KN clinical aim.

GB methods

in [191], Niyaz *et al.* exploit a GB method called LPA*. LPA* can adapt to dynamic changes in the graph without recalculating the entire graph. In this work, LPA* is used for efficient path re-planning of CTN validated in-silico but without a proper dynamic environment and brain tissue modelling.

SB methods

In [240]², an Rapidly-exploring Random Tree-star (RRT*) based path planner is proposed for a PBN for DBS aiming to minimize the insertion length while maximizing the obstacle clearance. The same

²Segato, A., Pieri, V., Favaro, A., Riva, M., Falini, A., De Momi, E., and Castellano, A. (2019). Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei. *Frontiers in Robotics and AI*, (6, 70).

was done by Hong *et al.* [119] but for a MFN [205]. In [79], Favaro *et al.* proposed a solution, for CED, based on a BIT-Star approach. These last three methods only proposed a pre-operative solution without taking into account dynamics generated in an intra-operative environment. In [78], Fauser *et al.* proposed a solution using an RRT-Con [151]. The proposed algorithm was validated in-silico but without mocking intraoperative dynamics, accounting for uncertainty in needle–tissue interaction and needle/tissue deformation.

Pinzi *et al.* in [210] proposed an AHFT approach that was developed and tested only for a pre-operative neurosurgical scenario. This work was extended in [209] via a combination between AHFT and BB for fast re-planning, named Extended Bubble Bending (EBB), taking into account tissue deformations.

MB methods

in [25] Bano *et al.* presented a GBO method for generating pre-operative paths for neurosurgical FBN. This method was later validated by Ko *et al.*, in [146] where the performance of a 4-mm diameter PBN was tested. Even in this work, during the in-silico validation, the framework didn't consider a dynamic environment. In [273], a heuristic-based search algorithm is introduced to find an optimal path of the needle for neurosurgical TA.

LB methods

In [242]³, Segato *et al.* implemented a DQN algorithm³, to create a path planning model, able to generalize on different patients anatomies. The proposed method achieves state-of-the-art performances in terms of obstacle avoidance, trajectory smoothness and computational time, proving this algorithm as a valid pre-operative planning method for complex environments, but still not accounting for tissue deformations as in a real clinical scenario.

In the field of path planning for steerable needles, applied to other surgical interventions, LB are being used. In [159], proposed by Lee *et al.*, a DQN algorithm is utilized to learn the control policy for needle steering with needle-tissue interaction. In [254], Tan *et al.* proposed an approach for biopsy and radio-frequency ablation. They proposed a multi-goal DQN to overcome the difficulties in uncertain needle–tissue interactions and enhance the robustness of robot-assisted insertions process. In [286], You *et al.* implemented a RL method, to manipulate a catheter to reach a target in a simulated environment and subsequently control a catheter in a real clinical setting. DRL algorithms, such as DQN, received more attention, and there are increasing applications of these in robot path planning [90]. The agent obtains knowledge through the exploration of an environment and learns using a process of trial and error. The DRL method shows advantages in path planning because it requires less prior information about the environment. So, where the environment is difficult to model, a RL approach proves its advantage. In the field of ML, many types of research have been conducted to combine LfD with RL generating an IRL algorithm [20]. Using policies trained from human demonstrations to initialize RL, while updating them through interacting with the environment, could efficiently reduce the time for policy optimization.

5.3 OBJECTIVE

This study investigates DRL and IRL methods for intra-operative re-planning where dynamics generated by needle-tissue interaction in a KN procedure were accounted. Unlike SOTA algorithms, such as FEM [120] and Mass-Spring Model (MSM) [98], the dynamic information linked to targets and obstacles, predicted with a PBD approach, can reduce the needle insertion errors. The advantages of the proposed learning based planning algorithms are the following. DRL and IRL realise sequential decision making under uncertainties through end-to-end learning without explicitly for modelling the system uncertainty. With the introduction of manual demonstration with the IRL approach, human expertise is embedded in the planning system. As a case study, we integrate the proposed planning models into a

³Segato, A., Sestini, L., Castellano, A., and De Momi, E. (2020, May). GA3C reinforcement learning for surgical steerable catheter path planning. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2429-2435). IEEE.

simulated framework that consists of a PBN and we treat the needle as a particle system. The proposed method is therefore generalisable to other steerable needles, which can be represented as particle systems. The experiments are designed to validate the effectiveness of the proposed algorithms in terms of Target Position Error (TPE), Target Orientation Error (TOE), Success Rate (SR) and Re-planning Time (RT). Moreover, the fact our intra-operative path planning framework can be efficiently integrated into a robotic system for needle steering provides that the algorithm can work in a practical clinical scenario. This work can be used to develop a virtual training system and assist the surgeon in planning intra-operative paths.

5.4 MATERIALS AND METHODS

5.4.1 Moving Agent - Flexible needle

Let us consider the PBN as the agent, Figure 5.1A. It is assumed that the motion of the needle tip fully determines the motion of the needle ("follow-the-leader" deployment) with a combination of the translation along the z axis and the rotation about the x and y axes. The pose of needle segment is defined as: $\mathbf{q}_{needle} = [x, y, z, \alpha, \beta, \gamma]^T$, with α, β, γ the angles around the x, y and z axes respectively.

The agent configurations are described by poses, denoted as 4×4 transformation matrices:

$$\mathbf{q}_{agent_i} = \begin{pmatrix} \mathbf{R}(\mathbf{q}_{needle}) & \mathbf{p}(\mathbf{q}_{needle}) \\ \mathbf{0}^T & 1 \end{pmatrix} \quad (5.1)$$

where $\mathbf{p}(\mathbf{q}_{needle}) = [x, y, z]^T \in \mathbb{R}^3$ is the agent position and $\mathbf{R}(\mathbf{q}_{needle}) = [\alpha, \beta, \gamma]^T \in SO(3)$ is the orientation. Where $SO(3)$ is the group of all rotations about the origin of three-dimensional Euclidean space \mathbb{R}^3 under the operation of composition [71].

5.4.2 Dynamic Environment - Needle-Tissue and Tissue-Tissue interactions

For the construction of the simulated "3D Dynamic Environment", Figure 5.1B, let us define:

- The "configuration space", C -space, as the set of all the possible t "agent configurations", \mathbf{q}_{agent_t} , with $t \in C$ -space.
- The "obstacle space", $C_{obst} \subset C$ -space, as the space occupied by obstacles.
- The "free space" $C_{free} \subset C$ -space, as the set of possible agent configurations, $\mathbf{q}_{agent} \in C_{free}$; $\mathbf{q}_{agent} \ni C_{obst}$.
- The "start configurations" $\mathbf{q}_{start_k} \in C_{free}$ with $k \in 1..N$
- The "target space" $C_{target} \subset C_{free}$, that is a subspace of "free space" which denotes where we want the catheter to move to, and the "target configuration" $\mathbf{q}_{target} \in C_{target}$.

The advantage of a re-path planner is accounting for real-time information of needle-tissue interactions, such as the movements of the target space C_{target_t} and configuration \mathbf{q}_{target_t} , and the obstacles space C_{obst_t} . PBD approach computes the time evolution of a dynamic system by directly updating positions, as first described by Müller et al. in [188]. Simulated brain structures are discretized as clusters of particles described by their positions \mathbf{p}_i and velocities \mathbf{v}_i , subject to a set of positional constraints $C_j(\mathbf{p}_i, \dots, \mathbf{p}_n) \geq 0$. We remind the reader of our work [238]⁴ for further details on the numerical model that mocks brain deformations during keyhole surgical procedures.

⁴Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., ... and De Momi, E. (2021). Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. IEEE Robotics and Automation Letters.

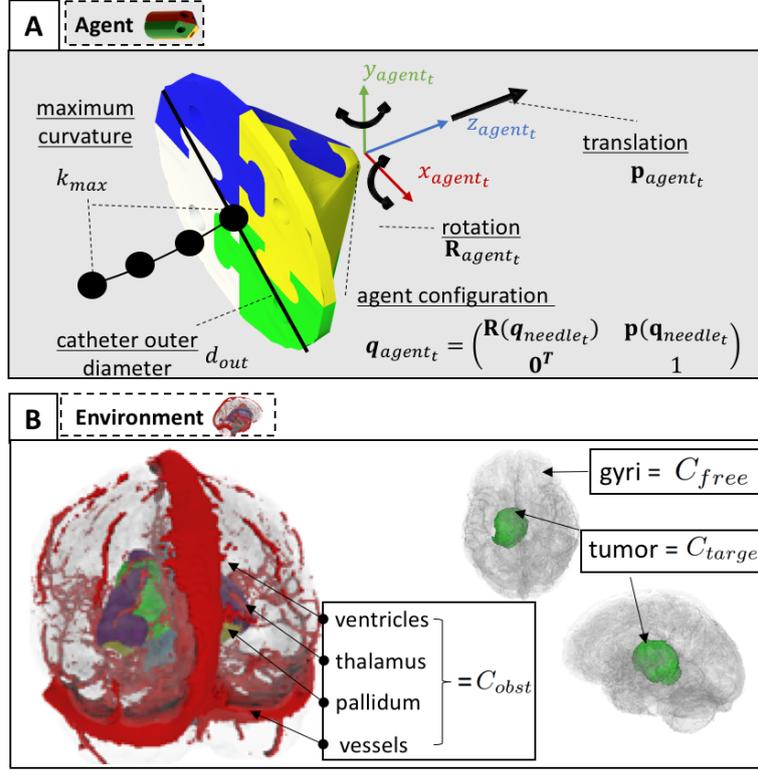


Figure 5.1: Agent and Dynamic Environment **A** The moving agent kinematic constraints are: the catheter diameter (d_{out}) and the maximum curvature (k_{max}) that it can perform. At $t - th$ time step it can translate (\mathbf{P}_{agent_t}) and rotate (\mathbf{R}_{agent_t}), from its configuration, \mathbf{q}_{agent_t} . **B** The environment is represented by a brain structure, the obstacle space (C_{obst}), the free space (C_{free}).

5.4.3 Problem: intra-operative path planning

The path planning problem is described as follows: we define k steps with $k = 1 : N$. Given the “start configurations” \mathbf{q}_{start_k} and a “target configuration”, \mathbf{q}_{target_t} . Since the motions of target and obstacles need to be considered, the configuration of needle-tissue interaction are also provided (C_{target_t} and C_{obst_t}). The procedure of needle insertion is divided into n time sequences. The outputs of the algorithm is a path, as an admissible sequence of “agent configurations”:

$$Q_k^{intra} = \{\mathbf{q}_{agent_0}, \mathbf{q}_{agent_1}, \dots, \mathbf{q}_{agent_{n-1}}\} \quad (5.2)$$

where $n = \#Q^{intra}$, $\mathbf{q}_{agent_0} = \mathbf{q}_{start_k}$, and $\mathbf{q}_{agent_{n-1}} = \mathbf{q}_{target_t}$.

5.4.4 Actions and Observations

The agent at every $t - th$ time step can take an action $a_t = [x, y, z, \alpha, \beta, \gamma]$, moving towards the target, \mathbf{q}_{target_t} , with a combination of the translation along its z axis and the rotation about its x and y axes. Actions moving the agent toward obstacles, C_{obst} or outside the Env, C_{free} , are considered inadmissible.

At every action, a_t , the agent collects observations, o_t , which corresponds to:

- the cumulative steps number t
- an obstacle collision $\mathbf{q}_{agent_t} \in C_{obst_t}$
- the achievement of the target $\mathbf{q}_{agent_t} == \mathbf{q}_{target_t} + tol$, where tol represent a certain tolerance (e.g. for minimally invasive neurosurgical treatment we can set a tol of 2.5 mm).

- how the target is reached i.e. the Euclidean difference between the needle's final position ($\mathbf{p}(\mathbf{q}_{needle})$) and the target position ($\mathbf{p}(\mathbf{q}_{target})$), $TPE = \|\mathbf{p}(\mathbf{q}_{target}) - \mathbf{p}(\mathbf{q}_{needle})\|$

5.4.5 Reward Function

The reward function, $R(\tau) = r_t$, associated with each time step, t , is shaped in order to make the agent learn to optimize the path, according to three main requirements:

- agent steps number minimization
- obstacle avoidance
- target position error minimization

The reward r_t is defined as:

$$r_t = \begin{cases} r_{stepmax} & \text{if } t \geq t_{max} \\ r_{obst} + r_{step} & \text{if } \mathbf{q}_{agent_t} \in C_{obst_t} \\ r_{target} + r_{TPE} + r_{step} & \text{if } \mathbf{q}_{agent_t} == \mathbf{q}_{target_t} \\ r_{step} & \text{otherwise} \end{cases} \quad (5.3)$$

- A negative reward, $r_{stepmax}$, is given if the the cumulative steps number (t) overcomes the predefined maximum number of steps allowed (t_{max}).
- A negative reward, $r_{step} = -\frac{1}{t_{max}}$, is given at each step t of the agent in order to obtain a reduction in the computational time.
- A negative reward, r_{obst} , is given if a collision is detected between the agent (\mathbf{q}_{agent_t}) and the obstacles (C_{obst_t}).
- A positive reward, r_{target} , is given upon reaching the target (\mathbf{q}_{target_t}).
- A negative reward is given, $r_{TPE} = -\|\mathbf{p}(\mathbf{q}_{target}) - \mathbf{p}(\mathbf{q}_{needle})\|$, upon reaching the target in order to minimize the difference between the target ($\mathbf{p}(\mathbf{q}_{target})$) and the catheter final position ($\mathbf{p}(\mathbf{q}_{needle})$).

The optimal parameters of the r_t , obtained with an empirical method, are reported in Table 5.2.

Table 5.2: Reward Function parameters values

$r_{stepmax}$	r_{obst}	r_{target}	t_{max}
-1	-1	+3	5000

5.4.6 DRL and IRL Model Construction

We want to design a robust intra-operative policy that can achieve the target configuration \mathbf{q}_{target_t} from $k = 1 : N$ different starting configuration, \mathbf{q}_{start_k} . At every time step t of the intra-operative procedure the trained model takes in input the current position of the agent (\mathbf{q}_{agent_t}) and the observations (o_t) and performs an action (a_t) according to the policy (π). In response, the agent receives in output from the model the next state $\mathbf{q}_{agent_{t+1}}$.

Three different approaches were investigated to train the policy generating 1 DRL-based model, based on Soft Actor Critic (SAC) (explained in Appendix 8.5.1), and 2 IRL-based models, both based on Proximal Policy Optimization (PPO) (explained in Appendix 8.5.2), to train a controlling policy based on expert trajectories with a GAIL approach (explained in Appendix 8.6.1). The three approaches are detailed here below.

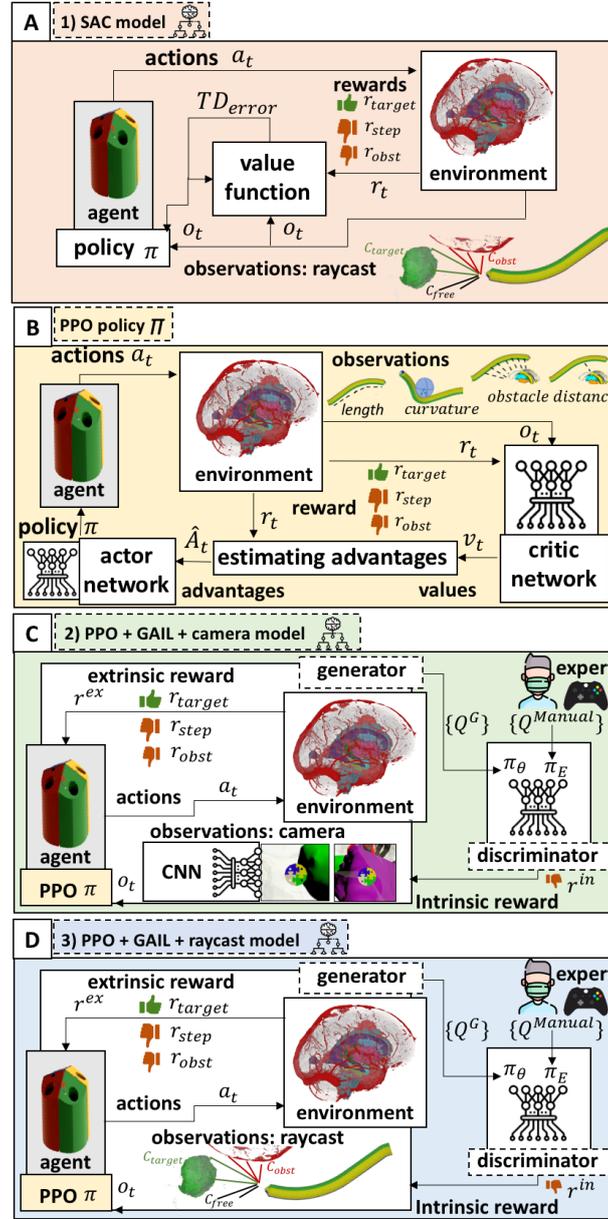


Figure 5.2: Models Construction **A** The SAC model is trained estimating the policy (π) and value function in every epoch, using the actual policy to interact with the environment, performing action a_t taking observation o_t , calculating the TD_{error} and returning the action probability density function. **B** PPO control policy, modelled as a NN (actor) and a second NN for estimating value functions (critic). **C** The PPO+GAIL+camera model is trained through a loop that starts with paths generated by an expert ($\{Q^{Manual}\}$) and paths ($\{Q^G\}$) performed by the network generator. With a GAIL approach, a discriminator with its network, taking in input the expert and generator policies (π_E and π_θ), makes a comparison of these two paths generating an intrinsic reward (r^{in}), updating the agent's policy (π). The loop continues until the generator, moving in the environment with actions (a_t), collecting observations with a visual encoder based on a CNN and computing an extrinsic reward (r^{ex}), can produce a path similar to the expert's demonstrations and that respects the kinematic constraints. **D** The PPO+GAIL+raycast model is trained as the model described in C, with the only difference being that the observations' nature is not visual but binary (collision or no collision).

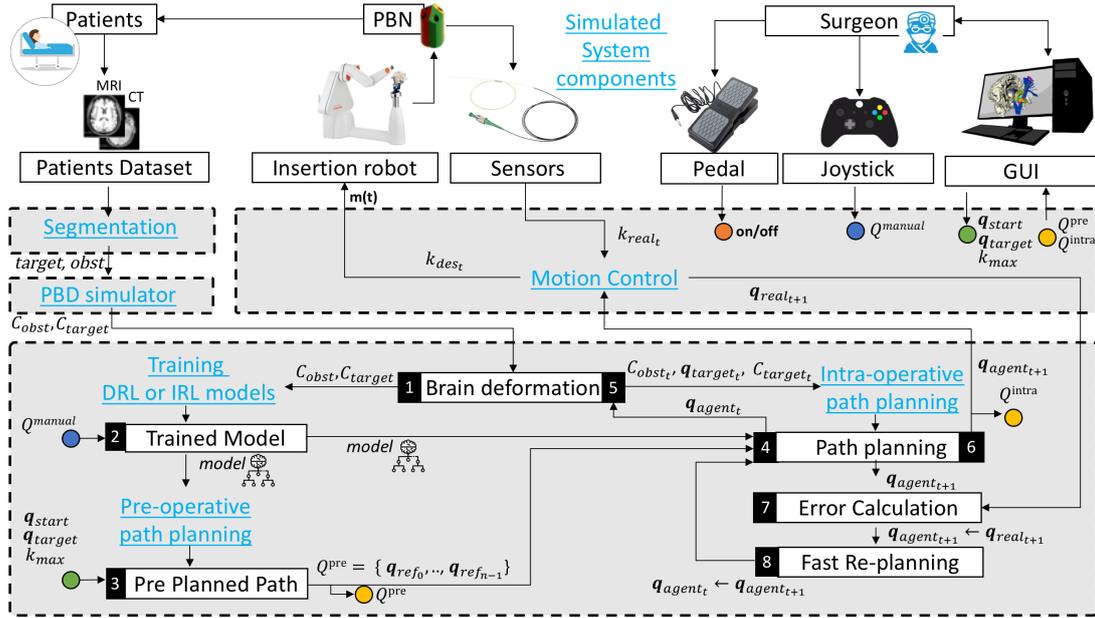


Figure 5.3: The simulation framework architecture is composed by the “Segmentation” module that takes in input the patient dataset (MRI and CT) and gives in output the white matter, target (target) and obstacles (obst) segmented meshes. The segmented meshes are the input of the “PBD simulator” of brain tissue deformation, that produces in output a (1) static environment composed by the target space and the obstacle space (C_{target} , C_{obst}) which are the input together with expert manual demonstration (Q^{manual}), start (q_{start}) and target (q_{target}) configuration and kinematics constraint (k_{max}) of the (2) “Training” the (3) “Pre-operative path planning” modules. The (2) and (3) returns respectively the trained model and the pre-operative path (Q^{pre}) both input of the (4) “Intra-operative path planning” module. At this point the (4) can start to interact with a Dynamic Environment (5) sending the agent current configuration (q_{agent_t}) and generating (6) the agent next step ($q_{agent_{t+1}}$). This configuration is sent to the “Motion Control” module, which estimates the curvature of the needle as to achieve (K_{des_t}) translated into the motion of the PBN. The configuration of the robot is registered by the sensors in the form of curvature (k_{real_t}) and translated as the real configuration of the tip ($q_{real_{t+1}}$). The estimation of the real tip configuration allows computing the (7) error with respect to the agent configuration previously computed ($q_{agent_{t+1}}$). This error affects the (8) fast re-planning that, in combination with the trained model, returns (4) the new configuration the needle has to achieve performing re-positioning, taking account of the committed error and trying to predict the future bias. The procedure continues in a loop until the target is reached or an obstacle is touched.

SAC model

During every epoch in the SAC model training, shown in Figure 5.2A, the actual agent policy (π) interacts with the environment moving in it with action (a_t), collecting observations (o_t), and computing a reward (r_t). After this, o_t and r_t are used to estimate a value for the temporal difference error (TD_{error}) [149], that is used to weigh the π , optimizing it in the directions of the actions that decrease the TD_{error} value.

PPO+GAIL+camera model

The PPO+GAIL+camera model relies on the PPO policy (Figure 5.2B). The PPO+GAIL+camera model, represented in Figure 5.2C, is based on the classic GAIL paradigm in which the RL-based generator network is represented by the PPO control policy and the observations’ nature is visual, as successfully applied in several vision based navigation approaches [37]. For this reason, a visual encoder (a CNN) is used to feed information from the camera sensor to the actor NN. With a GAIL approach, the dis-

criminator network, taking in input the expert and generator policies (π_e and π_θ), makes a comparison of these two paths generating an intrinsic reward (r_{in}) based on similarity score, updating the agent's policy (π). The loop continues until the generator, moving in the environment with actions (a_t), collecting observations, and computing an extrinsic reward (r_{ex}), can produce a path similar to the expert's demonstrations.

PPO+GAIL+raycast model

The PPO+GAIL+raycast model, shown in Figure 5.2D has the same structure of the PPO+GAIL+camera model with the difference that the observations' provided by the raycast, working as sensor, is not image based but binary input (collision or no collision).

5.4.7 Simulation Framework

The framework architecture shown in Figure 5.3 demonstrates the major process of implementing the proposed path planning system based on DRL or IRL, including segmentation and 3D mesh model construction, soft tissue modelling, agent training, pre-operative phase, intra-operative phase with fast re-planning and motion control. However, the proposed intra-operative algorithm could adapt other workflows with different clinical procedures.

Segmentation

Anonymized medical images (MRI/CT) of a healthy subject from a patient dataset acquired within the context of the Horizon 2020 EDEN2020 European project [49] were segmented (*FreeSurfer* Software) (A Healthy patient dataset was used with the intent to have a generic dataset to avoid the overfitting of the obtained model on a subject-specific pathological brain tumour environment). We performed segmentation of the white and grey matter (C_{free}) and key anatomical structures, such as the brain vessels and ventricles (C_{obst}) as mesh objects. Then the 3D models of relevant structures (white matter and obstacles) were constructed (*3D Slicer* Software).

PBD simulator

We used the GPU-based particle simulation *NVIDIA Flex* library for real-time needle-tissue and tissue-tissue PBD modelling of the brain, as described in details in [238]⁵.

Training DRL and IRL models

Our setup consists of pre and intra-operative path planning simulator based on *Unity Game Engine* with the training phase involving learning with SAC and mimicking the learned expert policy (π_E), taking in input the expert manual path (Q^{manual}), with PPO+GAIL, in order to obtain a trained model. Table 5.3 and Table 5.4 present the training parameters values referred to the SAC and PPO+GAIL models.

Pre-operative path planning

The trained model is firstly used to obtain the pre-operative path (Q^{pre}). Considering in input the entry configuration (q_{start}), the target configuration (q_{target}), the curvature kinematic constraint (k_{max}) and the static environment constituted by obstacles (C_{obst}) and the target structure (C_{target}).

Intra-operative path planning

As presented in Algorithm 4 and illustrated in Figure 5.4 the intra-operative planning takes as input the first pose of the pre-operative path (q_{ref0}), the maximum number of steps allowed (t_{max}), and the update frequency (t_{update}) of the agent's configuration, q_{agent_t} , with respect to the robot's configuration, q_{real_t} .

⁵Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., ... and De Momi, E. (2021). Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. *IEEE Robotics and Automation Letters*.

Table 5.4: PPO+GAIL training parameters

Table 5.3: SAC training parameters

<i>Parameter</i>	<i>Value</i>
max steps	1.0e6
batch size	32
buffer size	5.0e5
init-entcoef	0.5
strength	1.0
gamma	0.9
memory size	256
sequence length	128

<i>Parameter</i>	<i>Value</i>
beta	5.0e-4
max steps	1.0e5
batch size	64
buffer size	256
strength	1.0
gamma	0.99
curiosity strength	0.02
curiosity encoding size	128
GAIL strength	1.0
GAIL gamma	0.99
GAIL encoding size	128

The process starts by setting the initial configuration of the agent (\mathbf{q}_{agent_0}) to the initial configuration of the pre-operative path (\mathbf{q}_{ref_0}). For each step (t), until the ending conditions (target reached or obstacle collided) are not met:

- The dynamic environment is updated with the PBD algorithm by returning the new position and deformation of the obstacles (C_{obst_t}), target structure (C_{target_t}) and target configuration (\mathbf{q}_{target_t}).
- The trained model returns the next step of the agent ($\mathbf{q}_{agent_{t+1}}$) considering the updated environment and the current configuration of the agent.
- Every t_{update} step, the agent configuration is updated with the real configuration of the robot ($\mathbf{q}_{real_{t+1}}$) and the re-positioning of the agent is performed $\mathbf{q}_{agent_t} \leftarrow \mathbf{q}_{real_t}$.
- Finally, the output is the intra-operative path, Q^{intra} .

Motion control

During the intra-operative planning, the agent’s configuration is sent to the Motion control module (based on the mechanics-based model for 3-D steering of programmable bevel-tip needles proposed in [234]), which estimates the curvature the needle as to achieve (k_{des_t}). The insertion robot motor places the needle in the estimated configuration by the trained model. The estimation of the real tip configuration allows computing the error with respect to the previously computed desired configuration. This error affects the fast re-planning that, in combination with the trained model, returns a new configuration the needle has to achieve, taking into account the committed error and trying to predict the future bias. The procedure continues in a loop until the target is reached or an obstacle is collided.

Simulated system components

The PBN is a passive catheter with a finite orientation velocity and a bio-inspired insertion mechanism. Particularly, the offset measured at the tip between its four segments influences its bending direction.

5.4.8 Integration into Robot Operating System

In order to establish the communication between the planning model developed in Unity and the medical device Robotic Operating System (ROS) ROS Sharp (ROS#) was used. ROS# is a set of open-source software libraries and tools in C# for communicating with ROS from .NET applications, in particular Unity.

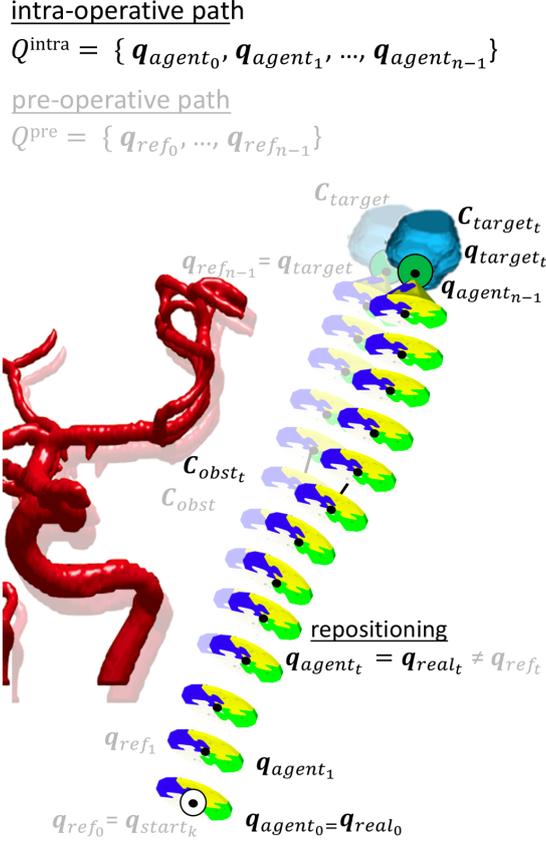


Figure 5.4: The intra-operative path, Q^{intra} , starts in \mathbf{q}_{agent_0} which coincides with the real robot initial configuration (\mathbf{q}_{real_0}), the pre-operative path (Q^{pre}) initial configuration (\mathbf{q}_{real_0}) which in turn coincides with the selected starting configuration (\mathbf{q}_{start_0}). The intra-operative path planning algorithm generates a new configuration (\mathbf{q}_{agent_t}) at every t step, most of the time different from (\mathbf{q}_{ref_t}) due to the real robot configuration (\mathbf{q}_{real_t}), where the agent perform re-positioning updating its current configuration with the real robot configuration $\mathbf{q}_{agent_t} \leftarrow \mathbf{q}_{ref_t}$. The last configuration ($\mathbf{q}_{agent_{n-1}}$) reaches the target configuration and space (\mathbf{q}_{target_t} and Q_{target_t}), avoiding obstacles (Q_{obst_t}), all deformed with respect to the initial configurations due to the simulated brain deformation.

5.4.9 Hardware Specification

We performed our experiments on a Windows machine equipped with a 6-core i5-9500 CPU (3.00 GHz), 16GB of RAM and 1 NVIDIA GeForce RTX 2060 SUPER GPU.

5.5 EXPERIMENTAL SETUP

5.5.1 Experimental protocol

As analysed, in Section 5.2, the only approach that presents a complete validation of the pre-operative path planning procedure with a static environment and intra-operative path planning procedure with a dynamic environment, for a KN steerable needle, is the EBB method recently proposed by Pinzi *et al.* [209]. This represented our SOTA. We recreated the same experimental setup used in [209], and performed the same number of experiments, performing 5 pre- and intra- operative path planning procedure on a healthy subject taken from the same dataset, starting from the same entry configurations and arriving at the same target configurations.

Algorithm 4 Intra-operative path planning

Input: $\mathbf{q}_{ref_0}, t_{max}, t_{update}$
Output: $Q^{intra} = \{\mathbf{q}_{agent_0}, \dots, \mathbf{q}_{agent_{n-1}}\}$

- 1: $Q^{intra} \leftarrow \{\mathbf{q}_{ref_0}\}$
- 2: $t \leftarrow 0$ // step counter
- 3: $target \leftarrow \mathbf{false}$ // target reached
- 4: $obstacle \leftarrow \mathbf{false}$ // obstacle collided
- 5: $\mathbf{q}_{agent_0} \leftarrow \mathbf{q}_{ref_0}$
- 6: **while** $\neg target$ **and** $\neg obstacle$ **and** $t \leq t_{max}$ **do**
- 7: $C_{obst_t}, C_{target_t}, \mathbf{q}_{target_t} \leftarrow \mathbf{PBD}(\mathbf{q}_{agent_t})$
- 8: $\mathbf{q}_{agent_{t+1}} \leftarrow \mathbf{MODEL}(\mathbf{q}_{agent_t}, C_{obst_t}, C_{target_t})$
- 9: **if** $\mathbf{q}_{agent_{t+1}} == \mathbf{q}_{target_t}$ **then**
- 10: $target \leftarrow \mathbf{true}$
- 11: **else if** $\mathbf{q}_{agent_{t+1}} \in C_{obst}$ **then**
- 12: $obstacle \leftarrow \mathbf{true}$
- 13: **else if** $t == t_{update}$ **then**
- 14: $\mathbf{q}_{real_{t+1}} \leftarrow \mathbf{ROBOT}(\mathbf{q}_{agent_{t+1}})$
- 15: $\mathbf{q}_{agent_{t+1}} \leftarrow \mathbf{q}_{real_{t+1}}$
- 16: $t \leftarrow 0$
- 17: **end if**
- 18: $t \leftarrow t + 1$
- 19: $\mathbf{q}_{agent_t} \leftarrow \mathbf{q}_{agent_{t+1}}$
- 20: $Q^{intra} \leftarrow Q^{intra} + \mathbf{q}_{agent_t}$
- 21: **end while**
- 22: **return** Q^{intra}

Training

As shown in Figure 5.5 the training phase, for SAC and PPO+GAIL models, takes in input m start (\mathbf{q}_{start_m}) and target (\mathbf{q}_{target_m}) configurations and expert manual path (Q_m^{manual}) (with $1 \leq m \leq 70$), the $\#m$ is obtained by combining the demonstrations provided by three expert users. At every episode, a new \mathbf{q}_{start_m} was chosen among the available ones along with its relative \mathbf{q}_{target_m} .

Pre-operative

The pre-operative analysis was performed in a static environment without the link with the needle steering robot (or its simulator) to test the ability of the models to generate a feasible path. For all the proposed approaches SAC, PPO+GAIL+camera and PPO+GAIL+raycast the pre-operative phase takes in input k \mathbf{q}_{start_k} and \mathbf{q}_{target_k} , (with $1 \leq k \leq 5$) and the kinematic constraints of the moving agent, $d_{out} = 2.5mm$ and $k_{max} = 0.007mm^{-1}$ (according to the PBN kinematics constraints) and produces k pre-operative path (Q_k^{pre}).

Intra-operative

On the other hand, the intra-operative protocol involved the needle steering robot simulator and a dynamic environment. In addition to the proposed approaches, we compared our learning-based methods with a sampling-based method, EBB [209], which uses the same experimental setup and shows comparable functionality and performance as analysed in Section 5.2. The produced pre-operative paths Q_k^{pre} , together with the maximum number of steps allowed (t_{max}) and the update frequency (t_{update}) are the inputs of j experiments, EXP^j (with $1 \leq j \leq 2$), where EXP^1 correspond to experiments without repositioning of the robot during the procedure while EXP^2 corresponds to experiments simulating the repositioning of the robot. Finally the output of each approaches are the intra-operative paths: $Q_k^{SAC,j}$, $Q_k^{GAIL+C,j}$, $Q_k^{GAIL+R,j}$ and $Q_k^{EBB,j}$.

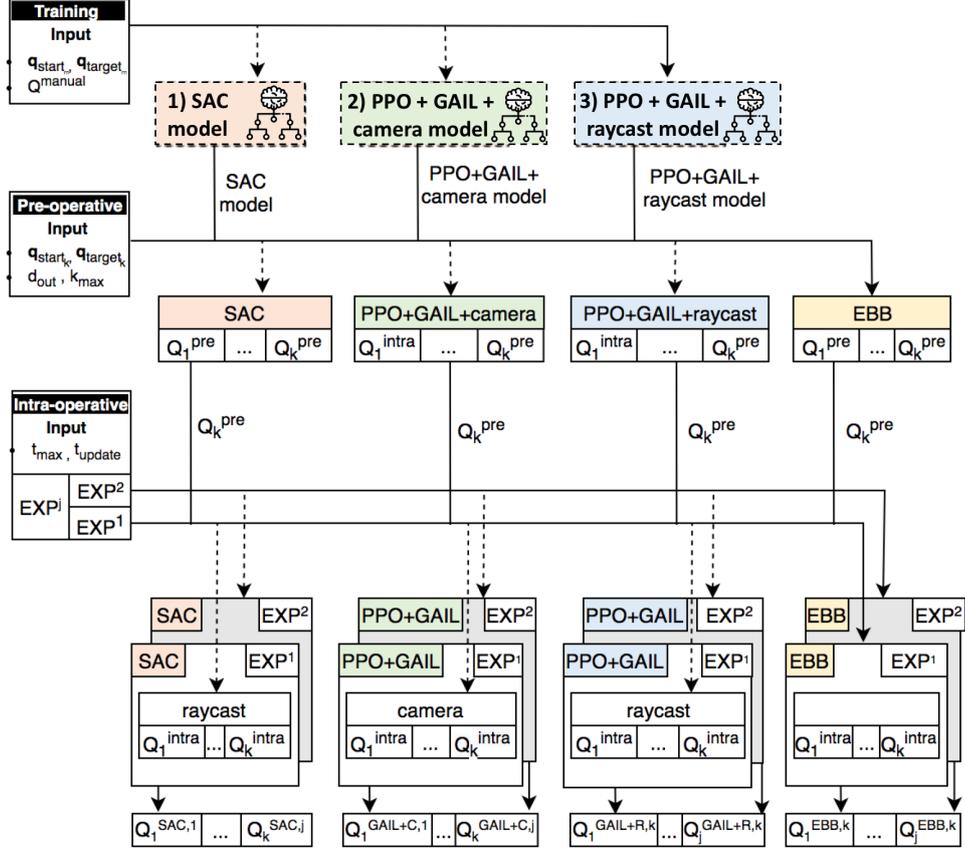


Figure 5.5: Experimental protocol. The training phase, for SAC and PPO+GAIL models, takes in input m start (\mathbf{q}_{start_m}) and target (\mathbf{q}_{target_m}) configurations and expert manual path (Q_m^{manual}) (with $1 \leq m \leq 70$). For all the proposed approaches SAC, PPO+GAIL+camera, PPO+GAIL+raycast and EBB the preoperative phase takes in input k \mathbf{q}_{start_k} and \mathbf{q}_{target_k} , (with $1 \leq k \leq 5$) and the kinematic constraints of the moving agent, d_{out} and k_{max} and produces k pre-operative path (Q_k^{pre}). The produced paths Q_k^{pre} , together with the maximum number of steps allowed (t_{max}) and the update frequency (t_{update}) are the inputs of j experiments, EXP^j (with $1 \leq j \leq 2$), where EXP^1 correspond to experiments without repositioning of the robot during the procedure while EXP^2 corresponds to experiments simulating the repositioning of the robot. Finally the output of each approaches are the intra-operative paths: $Q_k^{SAC,j}$, $Q_k^{GAIL+C,j}$, $Q_k^{GAIL+R,j}$ and $Q_k^{EBB,j}$.

5.5.2 Validation

For each pre-operative path, Q_k^{pre} , and intra-operative path, Q_k^{intra} , the quantitative analysis considers the following four indexes:

The re-planning time (RT [sec])

i.e. average computational time spent by the re-planner to calculate a new trajectory:

$$RT = \frac{1}{N} \sum (T_{update}^k - T_{real}^k) \quad (5.4)$$

where T_{update}^k represents the time at which the replanner has calculated the new trajectory, T_{real}^k the time at which the replanner has received the information about the real catheter position, N the total number of trajectory updates, and k the number of the specific update.

The success rate (SR[%])

i.e. the ratio between insertions reaching the target and the total number of simulated insertions:

$$SR = \frac{I_{target}}{I_{tot}} \quad (5.5)$$

where I_{win} represents the number of the insertions reaching the target \mathbf{q}_{target} , and I_{tot} the total number of insertions. This is a measure of the algorithm's robustness in reaching the target within the anatomical constraints.

The target orientation error (TOE[°])

i.e. the difference between the needle's final orientation and the target orientation:

$$TOE = \arccos(\mathbf{fw}_{target} \cdot \mathbf{fw}_{needle}) * \frac{180}{\pi} \quad (5.6)$$

where \mathbf{fw}_{target} and \mathbf{fw}_{needle} representing the 3-D normalized forward vectors of the target and the needle, respectively, at the end of the simulation. Given the 3×3 rotation matrices \mathbf{R}_{target} and \mathbf{R}_{needle} representing the orientation of target and needle respectively, the forward vectors represent the third column of the matrices in a z - forward world coordinate system: $\mathbf{fw}_{target} = \mathbf{R}_{target}[001]^T$ and $\mathbf{fw}_{needle} = \mathbf{R}_{needle}[001]^T$

The target position error (TPE[mm])

i.e. the Euclidean difference between the needle's final position ($\mathbf{p}(\mathbf{q}_{needle})$) and the target position ($\mathbf{p}(\mathbf{q}_{target})$):

$$TPE = \|\mathbf{p}(\mathbf{q}_{target}) - \mathbf{p}(\mathbf{q}_{needle})\| \quad (5.7)$$

Moreover, for each intra-operative path, Q_k^{intra} , the quantitative analysis considers three additional indexes:

The total length of the path (d_{tot} [mm])

i.e. the total distance covered by the insertion from the starting configuration to the target configuration:

$$d_{tot} = \sum_{t=0}^{n-1} d(\overline{\mathbf{p}(\mathbf{q}_{needle_t})\mathbf{p}(\mathbf{q}_{needle_{t+1}})}) \quad (5.8)$$

where the distance, d , between any two configurations can be calculated based on the Euclidean distance:

$$d(\overline{\mathbf{p}(\mathbf{q}_{needle_t})\mathbf{p}(\mathbf{q}_{needle_{t+1}})}) = \|\mathbf{p}(\mathbf{q}_{needle_t}) - \mathbf{p}(\mathbf{q}_{needle_{t+1}})\| \quad (5.9)$$

with $t \in 1, \dots, n, n = \#Q_k^{intra}$

The minimum distance from obstacles (d_{min} [mm])

i.e. the minimum distance of the path from the nearest obstacle indicating the level of safety, calculated as the minimum length between a line segments connecting two consecutive configuration of the path and the closest obstacle, such that:

$$d_{min} = \min\{d(\overline{\mathbf{p}(\mathbf{q}_{needle_t})\mathbf{p}(\mathbf{q}_{needle_{t+1}})}, \mathbf{p}_{obst_j})\} \forall_t, \forall_j \quad (5.10)$$

given the the line segment $\overline{\mathbf{p}(\mathbf{q}_{needle_t})\mathbf{p}(\mathbf{q}_{needle_{t+1}})}$ and the m obstacles represented by the occupied configurations $\mathbf{T}_{obst_j} = \begin{pmatrix} \mathbf{R}_{obst_j} & \mathbf{p}_{obst_j} \\ \mathbf{0} & 1 \end{pmatrix}$ with $j \in 1, \dots, m, m = \#C_{obst}$

The average distance from obstacles (d_{avg} [mm])

i.e. the average distance (d_{avg}) of the path from all the nearest m obstacles, that is calculated over the whole length of the path with respect to all the obstacles, such that:

$$d_{avg} = \frac{1}{n \cdot m} \sum_{t=1}^n \sum_{j=1}^m d(\mathbf{p}(\mathbf{q}_{needle_t})\mathbf{p}(\mathbf{q}_{needle_{t+1}}), \mathbf{p}_{obst_j}) \forall t, \forall j \quad (5.11)$$

5.6 RESULTS

Concerning pre-operative results, Table 5.5 shows a comparison between SAC, PPO+GAIL+camera and PPO+GAIL+raycast in terms of TPE, TOE and SR calculated over the Q_k^{pre} . The re-planning time is not reported because in the pre-operative simulation there is no catheter re-positioning. Both the PPO+GAIL approaches keep a lower TPE and TOE and a greater SR than the SAC approach.

Table 5.5: Pre-operative results

Method	TPE(mm)				TOE(mm)				SR (%)
	25 th	avg	75 th	std	25 th	avg	75 th	std	
Q_k^{SAC}	3.44	4.48	5.53	2.94	13.68	14	14.35	0.94	20
Q_k^{GAIL+C}	0.18	0.22	0.32	0.1	5.87	7.82	10.94	3.01	100
Q_k^{GAIL+R}	0.15	0.16	0.18	0.06	6.33	8.69	11.91	3.2	100

Concerning intra-operative results, shown qualitatively in Figure 5.6A, Table 5.6 shows a quantitative comparison between SAC, PPO+GAIL+camera, PPO+GAIL+raycast and EBB in terms of TPE, TOE, SR and RT calculated over the Q_k^{intra} . SAC failed in performing all the insertion with SR of 0%, for not reaching the target. As a consequence values of TPE and TOE for SAC cannot be computed (/). Both the PPO+GAIL approaches keep a greater SR than the SAC approach. Additionally, the PPO+GAIL+raycast keeps a lower TPE and TOE than the PPO+GAIL+camera, above all when re-positioning is introduced. Finally, the PPO+GAIL+raycast approach keeps comparable performance, with respect to the EBB approach concerning the TPE and TOE, in accordance with fixed tolerance values of 10° and 2.5 mm set arbitrary defined by [209] for this kind of experiments, but keeping a higher SR and an extremely lower RT. Figure 5.6B shows a representation of the performance metrics TPE, TOE, d_{min} , d_{avg} and d_{tot} .

Table 5.6: Intra-operative results

Method	Re-pos	TPE(mm)				TOE(mm)				SR (%)	RT (s)
		25 th	avg	75 th	std	25 th	avg	75 th	std		
Q_k^{SAC}	no	/	/	/	/	/	/	/	/	0	/
Q_k^{GAIL+C}	no	1.00	1.58	1.63	0.73	2.12	3.51	4.95	1.52	100	/
Q_k^{GAIL+R}	no	0.93	1.52	1.55	0.89	2.00	3.15	4.94	1.81	100	/
Q_k^{SAC}	yes	/	/	/	/	/	/	/	/	0	0.02
Q_k^{GAIL+C}	yes	1.90	2.44	2.95	0.96	1.91	2.87	3.31	1.22	100	0.02
Q_k^{GAIL+R}	yes	1.02	1.34	1.36	0.52	2.00	3.16	4.94	1.06	100	0.02
Q_k^{EBB}	yes	/	0.65	/	0.46	/	3.25	/	5.23	93.6	0.02

Concerning the obstacle avoidance of the intra-operative results, Table 5.7 and Figure 5.6C show a comparison between PPO+GAIL+camera, PPO+GAIL+raycast and EBB in terms of d_{avg} , d_{min} and

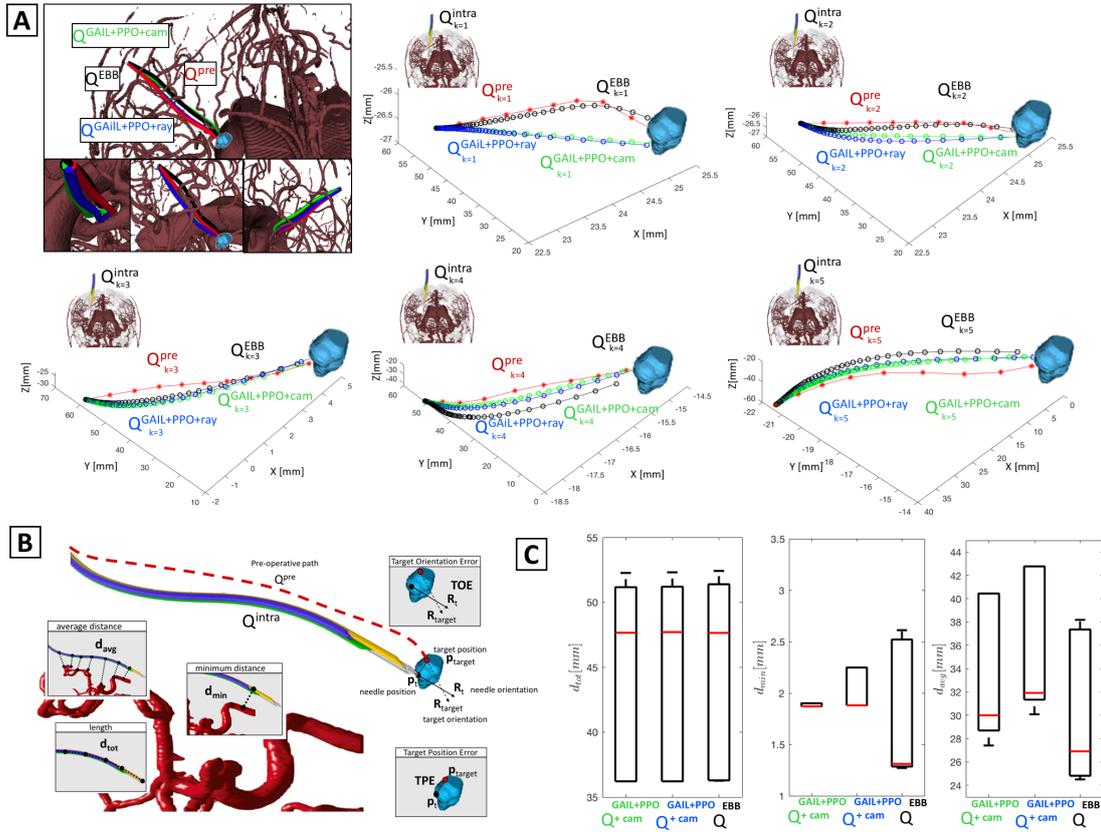


Figure 5.6: Results **A** Pre-operative path, Q_k^{pre} (red), and Intra-operative results, Q_k^{intra} plotted as a function of the path, k , for qualitative comparison (intra-operative tests were conducted on a total of 5 path, with $1 \leq k \leq 5$). The intra-operative path is reported for each approach: $Q_k^{GAIL+PPO+ray}$ (blue), $Q_k^{GAIL+PPO+cam}$ (green), Q_k^{EBB} (black). **B** Shows TOE, TPE, d_{min} , d_{avg} and d_{tot} . **C** Shows a comparison between GAIL+PPO+ray, GAIL+PPO+cam and EBB approach in KN environment. The results for both considered scenarios and used approaches are reported in terms of the minimum distance (d_{min}) and the average distance (d_{avg}) distance from the critical obstacles, the total path length (d_{tot}) calculated over the k paths for each approach. P -values ($* \leq 0.05$ and $n.s. \geq 0.05$) were calculated using Wilcoxon matched-pairs signed-rank test.

d_{tot} calculated over the Q_k^{intra} . The PPO+GAIL+raycast approach keeps a greater d_{min} and significantly greater d_{avg} (p -value $< 1.29e^{-04}$) from obstacles and a similar d_{tot} than the EBB approach.

Table 5.7: Intra-operative obstacles avoidance

Method	Re-pos	d_{avg} (mm)	d_{min} (mm)	d_{tot} (mm)
PPO+GAIL+camera	yes	33.43 ± 6.42	1.88 ± 0.02	44.63 ± 7.86
PPO+GAIL+raycast	yes	35.61 ± 6.35	2.17 ± 0.63	44.65 ± 7.88
EBB	yes	30.32 ± 6.75	1.79 ± 0.69	44.73 ± 7.91

5.7 DISCUSSION

The proposed path planning framework trained with the PPO+GAIL+raycast proposed approach can:

- Rapidly identify the finest surgical pre-operative path

- Simulate a realistic brain soft tissue deformation
- Be fast and can securely steer flexible needles with high accuracy and robustness
- Work adequately in dynamic intra-operative real-time scenario

TOE and TPE outcomes respect the corresponding tolerance values of 10° and 2.5 mm set for this type of experiment. d_{min} , d_{avg} outcomes demonstrated that the GAIL algorithm can securely steer in dynamic real-time scenarios. The outstanding value of RT provides that the GAIL algorithm can work adequately in dynamic real-time scenarios. Moreover, the results obtained are close to the already tested Extended Bubble Bending ones, supporting its possible application in intra-operative neurosurgical procedures. Finally, SR underlines the goodness of the proposed method. The comparison between our method and the direct optimisation method (EBB) shows that, in a dynamic environment and during intra-operative planning, where the time of execution is of the essence, the proposed method is more efficient in terms of re-planning time. The planner can avoid both static and dynamic obstacles without prior information about the environment. Moreover, contrary to methods based on direct optimisation, the space and time complexity of the proposed approach do not grow with the dynamicity.

5.8 CONCLUSION

The presented work assessed the performance of a new online steerable needle path-planner, based on an IRL approach, able to avoid anatomical obstacles while optimizing surgical criteria. This method is particularly suited for surgical procedures demanding high accuracy on the desired goal pose under tissue deformations and real-world inaccuracies.

Although our work has verified the effectiveness of IRL methods in KN path planning problems, there are a number of issues that need to be carefully resolved before these methods can be meaningfully implemented in a clinical setting. Existing IRL methods require an accurate model to be given beforehand or estimated from data. This is infeasible when such a model is lacking, or accurate estimation of the model is infeasible directly from expert demonstrations, particularly in clinical settings where the model always involves a large volume of continuous states and actions. It is thus valuable to apply IRL methods that are capable of estimating the rewards and model dynamics simultaneously. These issues are left for our future work together with the design of the control strategies necessary to achieve the planning behaviour in a high-fidelity simulation or a real surgical planning task. Since the state of the environment may not be perfectly observed by the agent, one may have to use a framework of partially observable Markov Decision Process for decision making under conditions of measurement uncertainty of the true state of the environment.

Toward Autonomous Control of Needle Steering Systems: An Inverse Reinforcement Learning Approach

Robotic-guided needles aim to reach a predefined target accurately by controlling the deflection of the needle tip during insertion. In doing so, they can reduce trauma to the patient by avoiding sensitive regions and increasing positioning accuracy. This class of needles has complicated kinematics than straight needles, often requiring complex control strategies to model the needle-tissue interaction. In this chapter we investigate **Hypothesis 4**, implementing an inverse reinforcement learning method to achieve an autonomous control of the tip pose. Using a generative adversarial imitation learning approach, this study aims to show that such a robot can learn to manipulate a catheter to reach a target in a simulated environment and subsequently control a catheter in an actual environment. A series of in-silico and in-vitro needle insertion experiments are performed in order to validate the controller performance in terms of success rate and targeting accuracy. Through these experiments, the study verifies that a learning model can be implemented in a robot system to control a biologically inspired 3d steering needle (**Hypothesis 4.1**, **Hypothesis 4.2**). The application of the method resulted in a journal paper [5]¹ currently under review.

6.1 INTRODUCTION

MIS involves numerous surgical techniques which significantly limit the incision size and subsequently reduce the risk of infection, blood loss, and wound healing time. Therefore, MIS is applicable for a wide range of patients who are not suitable for open surgery [89]. Needle insertion is one of the most crucial clinical approaches in MIS for biopsy, ablation, brachytherapy and fluid delivery and extraction treatments. The success of these procedures is defined by the accuracy in reaching the target while avoiding critical vital structures (e.g. vessels). In the past decade, steerable needle systems have been introduced to achieve high position accuracy at the target, reduce trauma to the tissue by avoiding puncturing critical

¹Autonomous Control of Needle Steering Systems: An Inverse Reinforcement Learning Approach. IEEE Transactions on Robotics (T-RO) (submitted on August 2021)

organs, an extensive analysis of their design can be found in [263]. This is of particular importance in KN, where straight line trajectories can be undesirable for, e.g., deep brain stimulation due to anatomical reasons, as targets can be blocked by eloquent brain tissue [70]. KN procedures can be aided by steerable devices, as such devices can accurately achieve brain target placement, increase the workspace of the needle, as well as avoid critical areas or obstacles along the way, including nerves, vessels and bones [195]. However, similar to other percutaneous approaches, the accuracy and robustness of flexible needle insertions are limited by unmodeled properties which affects the control of these system during the insertion (e.g. tissue deformation, tissue–needle interaction, etc.) [8]. The needle insertion accuracy and robustness directly contribute to treatment outcomes and postoperative recovery. Recent effort has been applied to develop robotic needle navigation systems and steerable robotic needle technologies so as to perform nonstraight paths, allowing the needle to avoid the anatomical obstacles and increase tip placement accuracy [263]. A significant component required to bring these robotically actuated steerable needles into use is the development of appropriate control strategies to achieve an accurate insertion.

6.2 RELATED WORK

Two main approaches are used to control needle steering systems: control strategies based on kinematic/dynamic model and methods that use path re-planning as a control paradigm.

Different techniques have been proposed in the first group according to the needle steering design [262]. For bevel-tip needle design, early works of Webster *et al.* shown a sliding mode control technique based on nonholonomic unicycle model that was validated in phantom and ex vivo liver tissue in the presence of tissue deformation [226]. A similar control strategy was used in combination with ultrasound image guidance [76] to achieve target position accuracy. A nonlinear adaptive control strategy was proposed by Motaharifar *et al.* [185] to account for unmodeled needle-tissue interactions, validated in-silico, where achieved better performance of existing feedback linearization-based techniques. For more complex needle system designs, such as the PBN, multiple strategies were adopted to counteract unmodelled dynamics of the tissue-needle interaction. Geometric-based controls were explored, ranging from pure model-based paradigm with feedback linearisation [147,235] to adaptive controls [236] where the tissue-needle interaction nonlinearities were learned during the insertion of the needle.

In the second group, different approaches were evaluated for different needle designs to address effects such as tissue deformation and target motion, both of which can significantly affect the achievable targeting accuracy [201]. Most fast path planners employ numerical solvers, thus providing more robustness to system non-linearity than first group methods. One of the most recent works for PBN based designs is a modified version of the “Bubble Bending”, which has been adapted to the surgical environment to be able to compensate for both tissue deformation and real-world uncertainties, proposed by Pinzi *et al.* in [209]. In [176], Matheson *et al.* proposed a hybrid controller which combines a cyclic and a direct control motions with a human-in-the loop strategy. With this approach the human operator can correct the performance in real-time thank to a visual interface and a teleoperated joystick.

In addition to these two types of approaches based on kinematic/dynamic models and methods that use path re-planning to control the needle, recent advances in ML demonstrate the ability to automate the control of needle steering systems [264]. A learning based approach can add further advantages such as increased stability of the procedure, reduced cognitive workloads, as well as facilitating supervised and collaborative autonomy of MIS procedure. In the field of KN, there are not significant applications, but in relation to flexible needles used for other surgical procedures, trained models were used to partially automate needle insertion tasks [54,215]. In some work, LfD based models have been used to extract needle movement patterns [55,215] and needle-tissue contact force patterns [216] from surgeon demonstrations. In the field of ML, many researches have been conducted to combine LfD with RL [104, 150, 225, 248]. Using policies trained from human demonstrations to initialize RL, while updating them through interacting with the environment, could efficiently reduce the time for policy optimization.

However, to the best of our knowledge, there is currently no implementation of LfD-RL framework in the field of control needle steering systems for KN. The main reason is that RL requires free exploration of actions in the physical setup, which is not feasible in invivo clinical setup. However, surgical robotic platforms can learn a control strategy in simulated environments with preoperative 3D images,

as demonstrated in [53]. In Appendix 8.5 and 8.6 a brief recap of the RL, IRL and GAIL theoretical background is proposed.

6.3 OBJECTIVE

This study aims to control a complex model of steerable needle, such a PBN using a controller modeled with GAIL [115], which is one of the LfD-RL approaches also defined as IRL method. The advantages of the proposed learning as control approach are the following. RL and IRL realise sequential decision making under uncertainties through end-to-end learning without the need for modelling the system uncertainty. Moreover, since approaches based on human motor control have been shown to achieve very good performance, what we propose is to incorporate through the IRL method the behaviour of the expert through demonstrations. With the introduction of manual demonstration with the IRL approach, the ‘human expertise’ is embedded in the control system. In this work, the PBN is considered as a case study for the presented method considering the complex model of the system. The PBN design consists of four axially interlocked segments, robotically actuated to develop specific tip configurations that allow the needle to steer in the brain tissue. The proposed approach was compared with two previous work [209, 275] respectively following control strategies based on kinematic/dynamic models (Figure 6.1A) and re-planning to control the needle (Figure 6.1B), to confirm if can steer the flexible needle in 3-dimensional model while achieving accuracy and robustness.

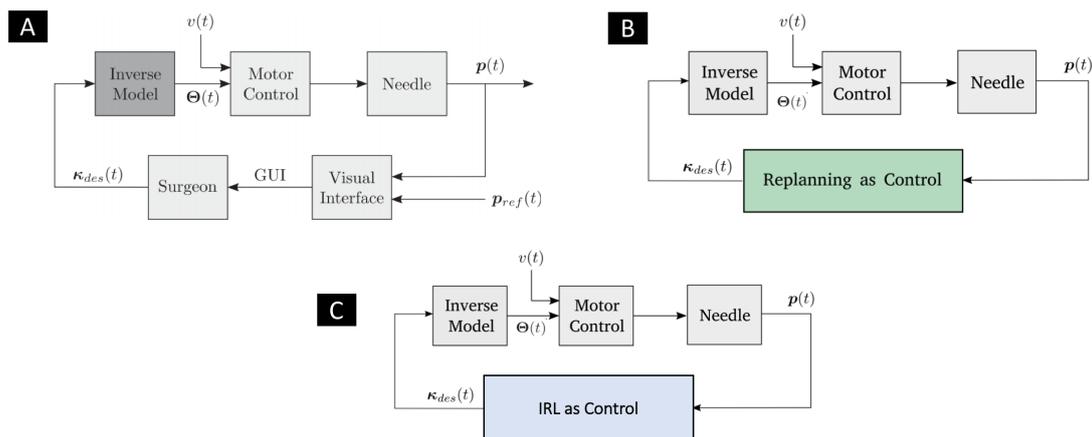


Figure 6.1: Block diagrams showing the different control systems for PBNs applied in the *Eden2020* context. (A) Representation of the “Surgeon-in-the-Loop” control system presented in [275]. The surgeon commands the desired needle curvature $\kappa_{des}(t)$ from which the inverse model calculates the best needle shape. The motor control system then computes the required actuation of the PBN to achieve the desired needle shape at an insertion speed of $v(t)$. The PBN’s resultant tip pose $\mathbf{q}(t)$ is displayed to the surgeon alongside a reference pre-planned tip pose $\mathbf{p}_{ref}(t)$ via a visual interface. (B) Representation of the “Re-planning as Control” control system presented in [209]. “Re-planning as Control” framework removes the need of a user guiding the needle insertion, thus avoiding external influence on the resulting accuracy. The system automatically steers the needle along paths that avoid obstacles of known location. (C) Representation of our “Inverse Reinforcement Learning (IRL) as control” system. IRL algorithm replaces the surgeon by computing the required curvatures once the information about the PBN tip position is received. (A) Courtesy of Watts et al. [275] and (B) Courtesy of Pinzi et al. [209]

6.4 MATERIALS AND METHODS

A dynamic simulation of KN environment and a needle model considered as particle system were designed to train the control model with an “IRL as Control” approach. The integration of the “IRL as Control” model and the simulator of a physical robotic system, previously presented in [176] was performed in order to substitute the human controller. Finally, the proposed “IRL as Control” model is integrated with a physical robotic system for validation by measuring success rate and the target accuracy of the robot control.

6.4.1 Control Problem

We defined the control problem for the case of PBN according to Fig. 6.1C: given a “starting pose” \mathbf{q}_{start} (e.g. entry pose in the brain) and a “target pose”, \mathbf{q}_{target} (e.g. location for a biopsy), the “IRL as Control” controller generates desired steering curvatures, \mathbf{k}_{des} , which acts as reference trajectories for the lower level control, which embed the forward and inverse dynamic model of the needle, described in [275].

6.4.2 IRL as Control

The main objective of the proposed controller is to learn a policy (π), which is able to reproduce the skill of an human expert in controlling the robotic needle system and, at the same time, to improve the driving skill by using the feedback provided by the training environment first, and by the real physical environment later. The approach is schematized in Figure 6.2. For this reason π is trained with a GAIL model. Since the expert’s policy, π_E , cannot be directly observed, the learner recovers the policy from the expert’s demonstrations Q^{Manual} . The discriminator network, taking in input the expert and generator policies (π_E and π_G) makes a comparison of these two paths generating an intrinsic reward (r_{in}) based on similarity score, updating the control policy (π). The loop continues until the Generator, moving in the environment with actions (a_t), collecting observations (o_t), and computing reward (r_t), can produce a policy similar or better than the human expert.

The observation o_t used for learning the robot’s policy include the pose of the PBN considered as particle system moving in the space.

$$o_t = \mathbf{q}(t) = \begin{pmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0}_{0 \times 3} & 1 \end{pmatrix} \quad (6.1)$$

where $\mathbf{p} \in \mathbb{R}^3$ is the tip position of the needle and $\mathbf{R} \in SO(3)$ is the orientation of the needle. The simulation composes the discrete model, and the needle’s tip location is calculated according to the motion values that change following the actions per one step, a_t .

The action a_t , corresponds to the desired curvatures, \mathbf{k}_{des} , required to reach a certain pose, $\mathbf{q}(t)$, starting from the previous pose, $\mathbf{q}(t-1)$.

$$a_t = \mathbf{k}_{des} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \quad (6.2)$$

where k_1 and k_2 are two curvature values defined in the Parallel Transport Frame (PTF) as it represents the best kinematic framing for the PBN [236].

The reward function $R(\tau) = \{r_t, obstacle, target\}$, associated with each time step, t , is shaped to make the agent learn the optimal trajectory Q , according to three main requirements: (1) agent steps number minimization, (2) obstacle avoidance, (3) target position error minimization. The reward r_t is defined as:

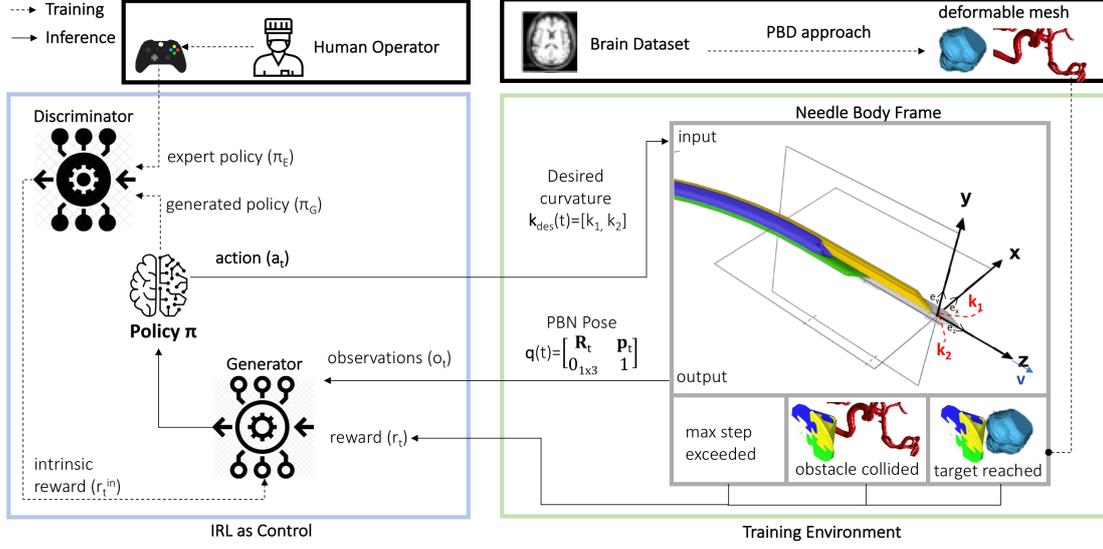


Figure 6.2: Training Environment. The IRL control policy, π , is trained through a loop that starts with the policy (π_E), generated by an expert and the policy (π_G) performed by the network generator. With a GAIL approach a discriminator with its network, taking in input the expert and generator policies (π_E and π_G), makes a comparison of these two paths generating an intrinsic reward (r^{in}), updating the learner policy (π). The loop continues until the generator, moving in the environment with actions (a_t), collecting observations and computing rewards (r_t), can produce a policy similar to the expert's one. The training environment is represented by deformable mesh object of the brain (obtained with a PBD approach) and the model of the PBN. Here the body frame coordinates of the PBN is shown. Angular and linear velocities, ω_x , ω_y , ω_z , and v are shown along with the corresponding curvatures k_2 , k_1 , and torsion τ (assumed to be zero for the EDEN2020 PBN needle).

$$r_t = \begin{cases} r_{step_{max}} & \text{if } t \geq t_{max} \\ r_{obst} + r_{step} & \text{if } \mathbf{p}_t \in C_{obst_t} \\ r_{target} + r_{TPE} + r_{step} & \text{if } \|\mathbf{p}_{target} - \mathbf{p}_t\| \leq tol \\ r_{step} & \text{otherwise} \end{cases} \quad (6.3)$$

- A negative reward, $r_{step_{max}}$, is given if the cumulative steps number (t) overcomes the predefined maximum number of steps allowed (t_{max}).
- A negative reward, $r_{step} = -\frac{1}{t_{max}}$, is given at each step t of the agent in order to obtain a reduction in the computational time.
- A negative reward, r_{obst} , is given if a collision is detected between the needle (\mathbf{p}_t) and the obstacles (C_{obst_t}). Where C_{obst_t} are the cluster of points defining the surface of the segmented meshes, labeled as obstacles.
- A positive reward, r_{target} , is given upon reaching the target (\mathbf{p}_{target}). The target is reached with a tolerance (tol) of 2.5 mm, with respect to the target position, which is within the acceptable accuracy for minimally invasive neurosurgical treatment and comparable to the results in [209].
- A negative reward is given, $r_{TPE} = -\|\mathbf{p}_{target} - \mathbf{p}_t\|$, upon reaching the target in order to minimize the difference between the target position (\mathbf{p}_{target}) and the needle final position (\mathbf{p}_t).

The boolean variable *obstacle* is defined as:

$$obstacle \leftarrow \text{true if } \mathbf{p}_t \in C_{obst_t} \quad (6.4)$$

The boolean variable *target* is defined as:

$$target \leftarrow \mathbf{true} \text{ if } \|\mathbf{p}_{target} - \mathbf{p}_t\| \leq tol \quad (6.5)$$

The Generator is involved during both the training and inference phases. As showed in Algorithm 5, the Generator takes as input \mathbf{q}_{start} and the maximum number of steps allowed t_{max} . The process starts by setting the initial pose of the needle ($\mathbf{q}(0)$) in \mathbf{q}_{start} . For each step (t), until the ending conditions (target reached or obstacle collided) are not met:

- The dynamic environment is updated with the PBD algorithm by returning the new position and deformation of the obstacles (C_{obst_t}), target structure (C_{target_t}) and target configuration (\mathbf{q}_{target_t}).
- The Reward function ($R(\tau)$) returns the reward (r_t) and the value of the boolean variables (*target* and *obstacles*)
- The policy function, initially trained using the RL PPO algorithm [233] and then optimized following the IRL GAIL [115] algorithm, takes in input the r_t and returns the action (\mathbf{k}_{dest}) which corresponds to the curvatures the needle has to follow.
- The pose of the needle $\mathbf{q}(t)$ is updated with the configuration of the tip of the needle, with the output of a simplified model in the case of the simulation environment or with the data form the sensing in the case of the physical robot. Finally, the output is the path, Q .

Algorithm 5 Generator

Input: $\mathbf{q}_{start}, t_{max}$

Output: $Q = \{\mathbf{q}(0), \dots, \mathbf{q}(n-1)\}$

```

1:  $t \leftarrow 0$  // step counter
2:  $target \leftarrow \mathbf{false}$  ▷ target initialization
3:  $obstacle \leftarrow \mathbf{false}$  ▷ obstacle initialization
4:  $\mathbf{q}(0) \leftarrow \mathbf{q}_{start}$ 
5:  $Q \leftarrow \{\mathbf{q}(0)\}$ 
6: while  $\neg target$  and  $\neg obstacle$  and  $t \leq t_{max}$  do
7:    $C_{obst_t}, C_{target_t}, \mathbf{q}_{target_t} \leftarrow \mathbf{PBD}(\mathbf{q}(t))$ 
8:    $r_t, target, obstacle \leftarrow \mathbf{R}(\tau)(\mathbf{q}(t), C_{obst_t}, C_{target_t}, \mathbf{q}_{target_t})$  ▷ Reward Function
9:    $\mathbf{k}_{dest} \leftarrow \pi(r_t)$  ▷ get the action ( $a_t = \mathbf{k}_{dest}$ ) from the Control's policy  $\pi$ 
10:   $\mathbf{q}(t) \leftarrow \mathbf{PBN}(\mathbf{k}_{dest})$  ▷ perform the action and get the observation  $o_t = \mathbf{q}(t)$ 
11:   $t \leftarrow t + 1$ 
12:   $Q \leftarrow Q + \mathbf{q}(t)$ 
13: end while
14: return  $Q$ 
    
```

The Discriminator is involved only during the training phase. As showed in Algorithm 6, the Discriminator receives in input the expert's paths, $\{Q^{Manual}\}$ and the initialization of the policy's parameters, θ_0 , and of the discriminator, w_0 . Until the Generator can produce a policy similar or better than the expert's behaviour $\pi_G \sim \pi_E$: -The path $\{Q\}$, obtained running the Generator, Algorithm 5, as inner loop, fits a parameter policy π_G , while the manual demonstration, $\{Q^{Manual}\}$, fits an expert policy, π_E . - The discriminator network D_w (w , weights) learns to distinguish the generated policy, π_G , from the expert one, π_E . The parameters w of D_w are updated in order to maximize Equation 8.8. -The discriminator, D , updates the agent's policy π to be close to the expert policy π_E using an intrinsic reward, defined with TRPO cost Function [232]:

$$r^{in}(\mathbf{q}(t), k_{dest}) = \log(D_{w_{i+1}}(\mathbf{q}(t), k_{dest}))$$

Finally the Discriminator return the control's policy π .

¹implemented in Unity ML-Agents Toolkit: <https://github.com/Unity-Technologies/ml-agents>

Algorithm 6 Discriminator

Input: Expert policy $\pi_E \sim \{Q^{Manual}\}$, initialize the policy and discriminator parameters θ_0, w_0

Output: Control's policy, π

```

1:  $\pi_G \leftarrow PPO$ 
2: while  $\pi_G \sim \pi_E$  do
3:    $\pi_G \sim \{Q\}$  ▷  $Q$  is obtained from the Generator
4:    $E_{\pi_G} [\log(D_w(\mathbf{q}(t), \mathbf{k}_{des_t}))] + E_{\pi_E} [1 - \log(D_w(\mathbf{q}(t), \mathbf{k}_{des_t}))]$  ▷ Update the  $D_w$  parameters
5:    $r^{in}(\mathbf{q}(t), \mathbf{k}_{des_t}) = -\log(1 - D_w(\mathbf{q}(t), \mathbf{k}_{des_t}))$  ▷ Compute the intrinsic reward,  $r^{in}$ 
6:    $\pi = \arg \max_{\pi_\theta} E_{\pi_{\theta_i}} [r^{in}]$  ▷ Update control' policy,  $\pi$ 
7: end while
8: return  $\pi$ 

```

6.4.3 Training Environment

To implement a learning policy, (π) from the simulation into a robot's experimental environment, the collisions between the needle model movements and brain model are simulated using the *Unity Engine* (<http://www.unity3d.com>).

PBD model

The needle-tissue and tissue-tissue deformation of the obstacles (C_{obst_t}), target structure (C_{target_t}) and target configuration (\mathbf{q}_{target_t}) are simulated using a PBD approach based on the *NVIDIA Flex* "flex for Unity Plugin" (the reader is referred to our previous work [238]² for further details on the simulator that mocks brain deformations during KN procedures). In the training environment, the PBD model is used to detect the collision of the PBN model.

Needle motion model

The motion of the needle is treated as moving particle defined in the Parallel Transport Frame as in [236]. Figure 6.2 shows the body frame coordinates of the needle, which it is assumed to move along a curve *EXP* parametrised in s :

$$EXP(s) = e^{(s\Pi)} \quad (6.6)$$

where s represents the arc length parameter while Π a 4×4 matrix defined as:

$$\Pi = \begin{pmatrix} \Omega & \mathbf{V} \\ 0 & 0 \end{pmatrix} \quad (6.7)$$

with Ω representing the rotational velocity while \mathbf{V} represents the linear velocity. The definition of Ω presents the Parallel Transport Frame Curvatures k_2 and k_1 as they describe the change of the tangent vector e_z in the e_y and e_x directions with Z-forward configuration:

$$\Omega \rightarrow \begin{bmatrix} 0 & 0 & -k_2 \\ 0 & 0 & k_1 \\ -k_1 & k_2 & 0 \end{bmatrix} \quad (6.8)$$

Pfaffian constraints on the linear velocity affect the kinematic model of steerable needle motion by simplifying the PBN tip curve equation:

$$v_x = 0, \quad v_y = 0, \quad v_z = \frac{dp}{dt} = v \quad (6.9)$$

With this simplification, we can compute the needle pose at time t by only knowing the tip pose at time $t - 1$, the curvatures k_2 and k_1 , and the linear velocity v on the z - axis:

$$\mathbf{q}(t) = e^{(s\Pi)} * \mathbf{q}(t - 1) \quad (6.10)$$

²Segato, A., Di Vece, C., Zucchelli, S., Di Marzo, M., Wendler, T., Azampour, M. F., ... and De Momi, E. (2021). Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. *IEEE Robotics and Automation Letters*.

6.4.4 Training strategy

The medical image dataset used in this work is derived by the repository [49], and it is composed of MRI/CT imaging dataset of one healthy subject. We performed segmentation of the skull and key anatomical structures, such as the brain vessels and ventricles, to obtain meshes which we labeled as obstacles (C_{obst}). We defined 70 test cases with k different pair of \mathbf{q}_{start} and \mathbf{q}_{target} poses. We collected 350 demonstrations Q^{Manual} (5 repetitions for each test case) from three expert users (right-handed - all subject provided signed informed consent prior to the testing phase - due to covid19, it was not possible to involve clinical expert). To record the demonstrations and user' commands, we used a 2D joystick (xbox 360) which projects into normalized axes the \mathbf{k}_{des} defined in Eq. 6.8, as in the user interface of [234]. For each test case, we defined the maximum curvatures achievable by the needle as $\mathbf{k}_{max} = [k1_{max}, k2_{max}]$, linearly mapped to the maximum value of 1 to the joystick inputs. The three users gained driving skills of the system by training with it for 21 hours. The demonstrations were collected by randomizing the order of test cases. Table 6.1 summarizes the mean reward and the number of episodes recorded by the three different users during demonstrations.

Table 6.1: *Demonstrations recording*

Method	$v(t)(mm/s)$	$\mathbf{k}_{max}(mm^{-1})$	Mean Reward	Episodes
IRL as Control	0.8	0.005	2.30	158
IRL as Control	0.8	0.007	2.71	265
IRL as Control	0.8	0.014	2.75	266

Users' demonstrations were passed to GAIL agents for training. Compared to traditional IRL algorithms, which recover the reward function from the features of data without calculating the optimal policy, GAIL agents are easier to train and they output both reward signal and policy. In addition, GAIL overcomes the distribution mismatch issue caused by multimodal behaviors of demonstrations [258], Table 6.2 presents the training parameters values.

Table 6.2: *Training parameters*

<i>Parameter</i>	<i>Value</i>	<i>Parameter</i>	<i>Value</i>
beta	5.0e-4	max steps	1.0e5
batch size	64	buffer size	256
strength	1.0	gamma	0.99
curiosity strength	0.02	curiosity encoding size	128
GAIL strength	1.0	GAIL gamma	0.99
GAIL encoding size	128		

6.4.5 Integration with the physical robotic system

The software of robotic system described in [234] is designed using ROS, that is employed as framework to interconnect different hardware and software modules. In order to establish the communication between our IRL control implemented in *Unity Engine* and the physical robot, we used ROS Sharp (ROS#). ROS# is a set of open-source software libraries and tools in C# for communicating with ROS from .NET applications. As shown in Figure 6.3A, the communication is established through the activation of nodes on which both *Unity Engine* and ROS can publish messages or read messages. Figure 6.3B show the communication protocol between the "IRL as Control" running on *Unity Engine* and the robotic environment based on ROS, with the messages exchanged with the aid of ROS#.

Integration with EDEN2020 Robotic System

With ROS, it was possible to connect the "IRL as Control" with the front end of the EDEN2020 system, shown in Figure 6.3C. Brain meshes that will create the simulation environment are provided to *Unity*.

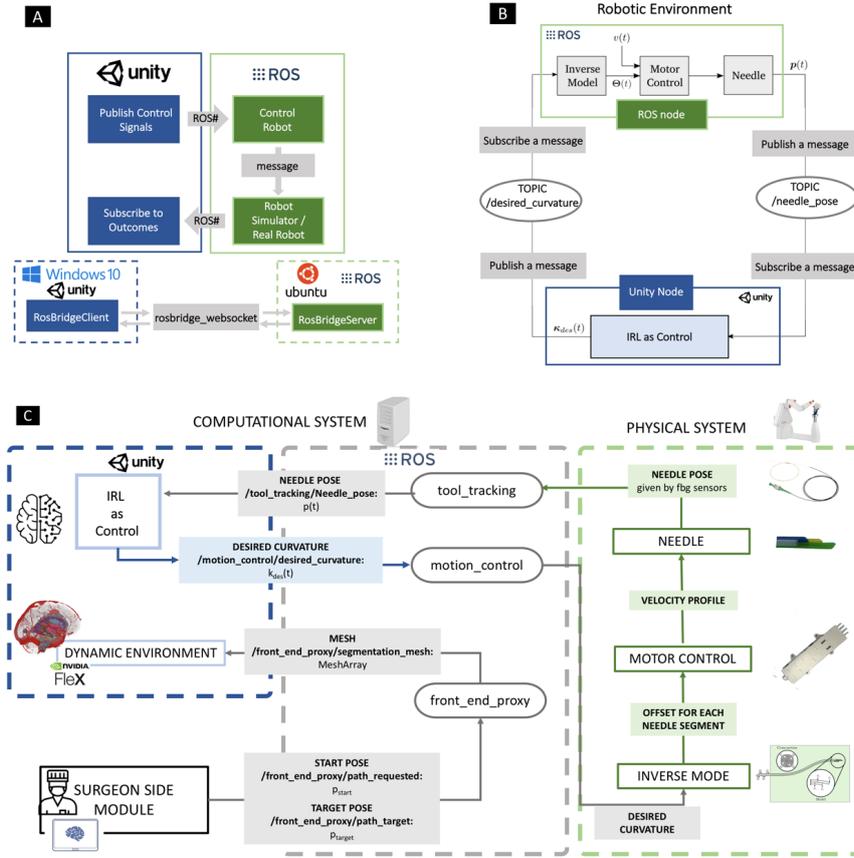


Figure 6.3: ROS Integration. (A) In order to establish the communication between Unity and ROS, we use ROS Sharp (ROS#). The communication is established through the activation of nodes on which both Unity and ROS can publish messages or read messages already published to make this communication the package Rosbridge is required. Rosbridge provides a JSON API to ROS functionality for non-ROS programs. The front-ends that interface with Rosbridge includes a WebSocket server for web browsers interaction. (B) The communication protocol between the “IRL as Control” based on Unity Engine and the robotic environment based on ROS, with the messages exchanged with the aid of ROS#. Unity is a publisher of the node /desired_curvature and a subscriber of the node /needle_pose. On the contrary, ROS is a subscriber of the node /desired_curvature and a publisher of the node /needle_pose. In this way, Unity can send information about the curvature the needle has to follow to the robot, and the robot can return to Unity the needle position. (C) Thanks to ROS, it was possible to connect the “IRL as Control” with the front end of the EDEN2020 system. The round boxes indicate a topic, while the squared ones are nodes. ROS’s meshes are provided to Unity thanks to the message the /front_end_proxy/segmentation_mesh of type the /MeshArray. Soft tissue modelling is applied to the different objects to generate the dynamic environment and take account of soft tissue deformation. ROS also passes user inputs such as start pose, (\mathbf{q}_{start}) and target pose (\mathbf{q}_{target}). The desired curvature ($k_{des}(t)$) is sent to the motion_control topic, which sends the information received to the desired curvature-offset conversion “inverse module”. These offsets are sent directly to the robot, moving the catheter by the desired displacement. FBG sensors return the exact pose ($\mathbf{q}(t)$) of the catheter tip once the movement is performed. The FBG’s feedback is provided to the Unity module thanks to the tool_tracking topic.

Soft tissue modelling is applied to the different objects to generate the dynamic environment and take account of soft tissue deformation. ROS also passes user inputs such as start pose, (\mathbf{q}_{start}) and target pose (\mathbf{q}_{target}). The desired curvature ($k_{des}(t)$) is sent to curvature-offset conversion “inverse module”. These

offsets are sent directly to the robot, moving the catheter by the desired displacement. However, the robot carries delays due to the robot’s mechanics and the catheter encounters’ resistance when driving in the real environment. For this reason, FBG return the exact position of the catheter tip once the movement is performed. FBG return the exact pose ($q(t)$) of the catheter tip once the movement is performed. In this way, it is possible to calculate the error between the previously predicted pose ($q(t - 1)$) and the actual one, allowing the controller to re-position and correct the pose and still achieve the target. The procedure continues in a loop until the target is reached or an obstacle is touched.

6.4.6 Simulated Environment

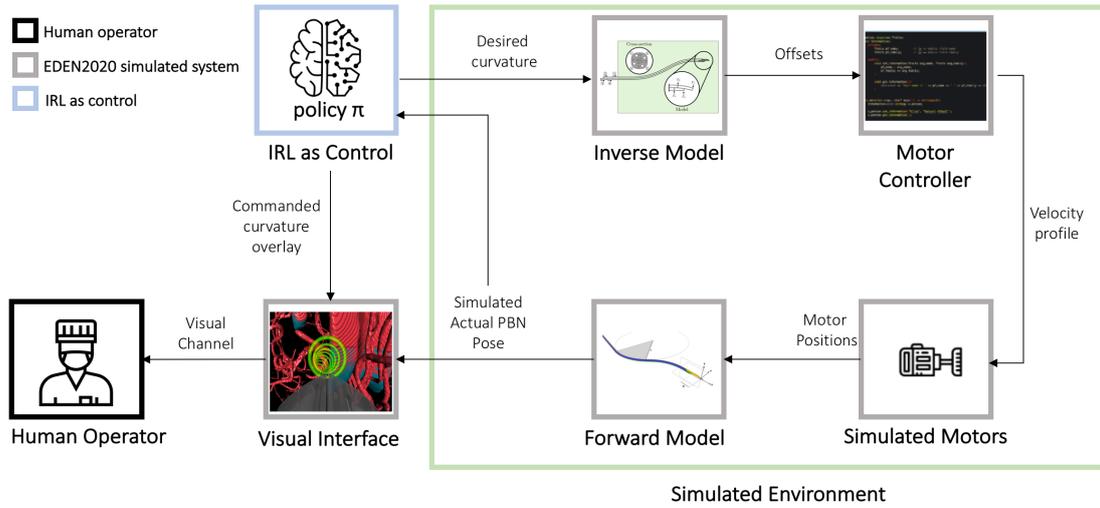


Figure 6.4: Simulated Environment. The “IRL as control” approach is integrated with the EDEN2020 simulator of the robotic system. The “IRL as control” outputs the desired curvature values. These curvature values are passed to the inverse model of the needle, which returns the required offsets between the needle segments to achieve that curvature. These required offsets serve as the inputs to the low-level controller, which generates the motion profiles (containing cyclic motions) for the simulated motors to move the segments and create the commanded offset. The expected curvature of the needle, based on the forward model of the needle converting segment offset to curvature, as well as the actual pose of the needle can then be shown to the operator via the visual interface and sent to the “IRL as control” as input. Part of the image is courtesy of Matheson et al. [176]

We first integrated the “IRL as control” with the a robotic system simulator, shown in Figure 6.4, which is composed by:

- The inverse model module: the desired curvature values are passed to the inverse model of the needle, which returns the required offsets between the needle segments to achieve that curvature as detailed in [276].
- The robotic mechanism model: the motor positions are defined via trapezoidal numerical integration of the velocity values, with a time step of 0.05 s. Each motor velocity profile assumed instantaneous acceleration.
- The Forward model: The pose of the needle is calculated using the forward mechanics-based model of the needle as detailed in [275]. This model estimates the curvatures of the needle given the current segment configuration during the motion.

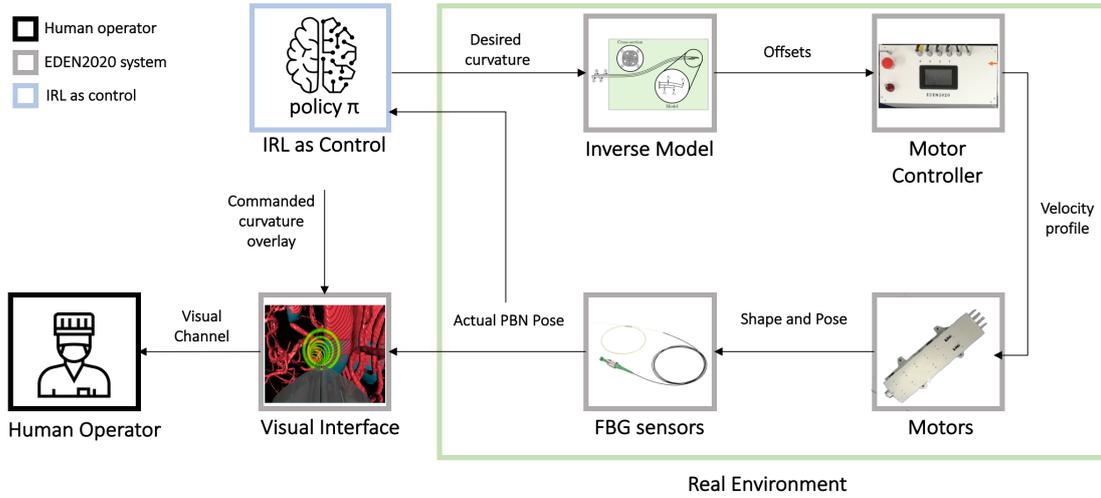


Figure 6.5: Real Environment. The “IRL as control” approach integrated with the EDEN2020 robotic system. The “IRL as control” outputs the desired curvature values. These curvature values are passed to the inverse model of the needle, which returns the required offsets between the needle segments to achieve that curvature. These required offsets serve as the inputs to the low-level controller, which generates the motion profiles (containing cyclic motions) for the motors to move the segments and create the commanded offset. The curvature of the needle, as well as the actual pose of the needle, measured by the FBG sensors, can then be shown to the operator via the visual interface and sent to The “IRL as control” as input.

6.4.7 Real Environment

To implement the policy model from the simulation learning to a real physical robotic system, the Enhanced Delivery Ecosystem for Neurosurgery in 2020 (EDEN2020) actual robot environment’s experiment is performed as shown in Figure 6.5, which is composed by:

- The inverse model: same as the simulation environment.
- The motor controller: generates the motion profiles (containing cyclic motions) for the motors to move the segments and create the commanded offset. The cyclic controller is largely based on the work presented in [177]
- The motor: The physical system uses Maxon high-precision DC brushed motors, each equipped with a rotary magnetic encoder. For each the net speed of the needle is 1 mm/s, and the maximum speed of any segment when moving to create an offset is 5 mm/s.
- The Fiber Bragg Grating (FBG) sesors: FBG are utilized as sensors to measure the pose of a multi-segment needle. The reconstruction technique that provides the pose of such a fiber is presented in [143].

6.5 EXPERIMENTAL SETUP

We compared our “IRL as Control” method with a “Re-planning as Control” presented by Pinzi et.al. [209] where a EBB sampling-based algorithm was used to drive a PBN with the physical robotic system described in [234]. The EBB shows to be comparable in terms of functionality and performance. The medical image dataset used is the same as the one used for the training phase, described in Section 6.4.4 A pre-operative path Q^{pre} for each experimental case was defined by a clinical expert. In addition, for each path we define the entry pose in the brain as well the target location, respectively \mathbf{q}_{start} and \mathbf{q}_{target} .

6.5.1 In-silico experiments: training on experimental dataset

The in-silico experiments involved a set of k pre-operative paths Q_k^{pre} (with $1 \leq k \leq 5$) as the inputs of j experiments, EXP^j (with $1 \leq j \leq 2$), where EXP^1 correspond to experiments without re-positioning of the robot during the procedure while EXP^2 corresponds to experiments simulating the re-positioning of the robot. The velocity $v(t)$ was fixed to 0.8 mm/s (empirical tuning according to [209]). While we used g different values of maximum curvatures \mathbf{k}_{max}^g (with $1 \leq g \leq 3$) corresponding respectively to $\mathbf{k}_{max}^1 = 0.005 \text{ mm}^{-1}$ (as in [198]), $\mathbf{k}_{max}^2 = 0.007 \text{ mm}^{-1}$ (as in [45]) and $\mathbf{k}_{max}^3 = 0.014 \text{ mm}^{-1}$ (as the maximum curvatures achievable by the current prototype of PBN). The output of each approaches are the in-silico paths: $Q_{k,j,g}^{silico}$.

6.5.2 In-Vitro Experiment

The workflow of the physical robotic system is shown in Figure 6.5. The in-vitro experiments, involved the needle insertions, were performed on phantom-brain gelatin (7% by weight swine gelatin - Il Molino Chiavazza Gelatin [81]) to assess the performance of the proposed solution in conditions which replicate real insertion scenarios, shown in Figure 6.6A. In-vitro trials were performed using a four-segment PBN design, the details of which can be found in [275]. The PBN, shown in Figure 6.6B, is formed by four segments featuring one lumen of 0.3 mm each, and an overall outer diameter of 2.5 mm for all segments together. The experimental setup is reported in Figure 6.6C and described hereinafter. The insertion of the needle was driven by the robotic system described in Secoli et al. [234]. The needle driver is integrated within the neuromate©(Renishaw Plc) end-effector. The needle driver is composed by four linear actuators (one per each PBN segment). As shown in Figure 6.6D, the catheter is sensorized with four multi-core fibers that have FBG; each catheter segment has one fiber to track the needle tip pose and the curvature. Figure 6.6E shown the Visual interface integrated into the commercial neurosurgical planning and intraoperative Software Neurosinspire™(Renishaw plc) while Figure 6.6F shown the FBG sensors visual interface. One ($k = 1$ with respect to the in-silico experiments) pre-operative paths $Q_{k=1}^{pre}$ is the inputs of j experiments, EXP^j (with $1 \leq j \leq 2$), where EXP^1 correspond to experiments without re-positioning of the robot during the procedure while EXP^2 corresponds to experiments simulating the re-positioning of the robot. The velocity $v(t)$ was fixed to 0.8 mm/s , while we used $\mathbf{k}_{max}^{g=2} = 0.007 \text{ mm}^{-1}$ as value of maximum curvature ($g = 2$ with respect to the in-silico experiments). The choice of \mathbf{k}_{max} was made on the basis of the results obtained from the in-silico experiments. The output of each approaches are the in-vitro paths: $Q_{k=1,j,g=2}^{vitro}$.

6.6 EXPERIMENTAL ANALYSES

For each in-silico path, $Q_{k,j,g}^{silico}$, and in-vitro path, $Q_{k=1,j,g=2}^{vitro}$, the quantitative analysis considers the following four metrics:

- The computation time (CT [sec]) *i.e.* average computational time spent by the controller to calculate a new pose:

$$RT = \frac{1}{N} \sum (T_{update}^k - T_{real}^k) \quad (6.11)$$

where T_{update}^k represents the time at which the controller has calculated the new pose, T_{real}^k the time at which the controller has received the information about the needle pose, N the total number of pose updates, and k the number of the specific update.

- The success rate (SR[%]) *i.e.* the ratio between insertions reaching the target and the total number of simulated insertions:

$$SR = \frac{I_{target}}{I_{tot}} \quad (6.12)$$

where I_{win} represents the number of the insertions reaching the target \mathbf{p}_{target} , and I_{tot} the total number of insertions. This is a measure of the controller's robustness in reaching the target within the anatomical constraints.

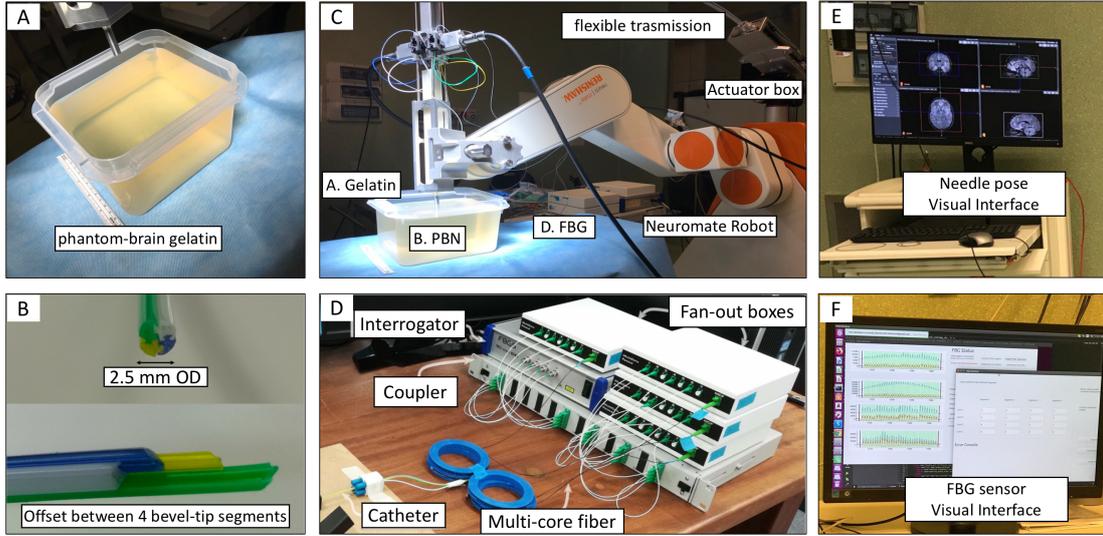


Figure 6.6: Experimental Setup. (A) 7% by weight swine gelatin. (B) 2.5 mm outer diameter biocompatible Programmable Bevel-tip Needle. (C) Full experimental setup: the PBN is inserted into gelatin and tracked by the FBG sensors. The neuromate robot (Renishaw Plc), an actuation box controls each segment of the PBN and encoders measure their insertion length. (D) The PBN sensorized with four multi-core fibers that have FBG; each catheter segment has one fiber. The experiment utilizes four multi-core fibers, fan-out boxes, interrogator and coupler (Courtesy of Khan et al. [143]). (E) The Visual interface integrated into the commercial neurosurgical planning and intraoperative software neurosinpire™ (Renishaw plc). (F) FBG sensors visual interface.

- The target orientation error (TOE[°]) i.e. the difference between the needle’s final orientation and the target orientation:

$$TOE = \arccos(\mathbf{fw}_{target} \cdot \mathbf{fw}_t) * \frac{180}{\pi} \quad (6.13)$$

where \mathbf{fw}_{target} and \mathbf{fw}_t representing the 3-D normalized forward vectors of the target and the needle, respectively, at the end of the simulation. Given the 3×3 rotation matrices \mathbf{R}_{target} and \mathbf{R}_t representing the orientation of target and needle respectively, the forward vectors represent the third column of the matrices in a z – forward world coordinate system:

$$\mathbf{fw}_{target} = \mathbf{R}_{target}[0, 0, 1]^T \quad \mathbf{fw}_t = \mathbf{R}_t[0, 0, 1]^T$$

- The target position error (TPE[mm]) defined as the euclidean difference between the needle’s final position and the target position:

$$TPE = \|\mathbf{p}_{target} - \mathbf{p}_t\| \quad (6.14)$$

6.7 RESULTS

6.7.1 In-Silico Study

The Table 6.3 compares “IRL as Control” and “Re-planning as Control” according to the metrics TPE, TOE, SR and RT calculated over the $Q_{k,j,g}^{silico}$. The “IRL as Control” with re-positioning approaches keep a lower TPE and TOE and a greater SR than the “IRL as Control” without re-positioning. Additionally, the “IRL as Control” with $\mathbf{k}_{max} = 0.007$ approach keep a lower TPE and TOE than the “IRL as Control” with $\mathbf{k}_{max} = 0.005$ and $\mathbf{k}_{max} = 0.014$ approaches, above all when re-positioning is introduced.

Finally, the “IRL as Control” approach keeps comparable performance with respect to the “Re-planning as Control” approach concerning the TPE and TOE but keeping a higher SR and an extremely lower RT. In Figure 6.7 are shown in-silico results, $Q_{k,j,g}^{silico}$ plotted as a function of the path, k , as a function of the experiment type, j , and as a function of maximum curvature, g . Target position error TPE and Target orientation error TOE are reported for each result.

Table 6.3: In-silico results

Method	Re-pos	$v(t)(mm/s)$	$k_{max}(mm^{-1})$	TPE (mm)	TOE ($^{\circ}$)	SR(%)	RT(s)
IRL as Control	no	0.83	0.005	1.89 ± 0.78	3.65 ± 1.06	100	/
IRL as Control	no	0.83	0.007	1.52 ± 0.89	3.15 ± 1.81	100	/
IRL as Control	no	0.83	0.014	1.93 ± 1.09	$4,13 \pm 1,89$	100	/
IRL as Control	yes	0.83	0.005	2.57 ± 0.72	3.19 ± 0.80	100	0.02
IRL as Control	yes	0.83	0.007	1.34 ± 0.52	3.16 ± 1.06	100	0.02
IRL as Control	yes	0.83	0.014	1.43 ± 0.65	3.40 ± 1.33	100	0.02
Re-planning as Control	yes	0.4	0.014	0.65 ± 0.46	3.25 ± 5.23	93.6	0.53

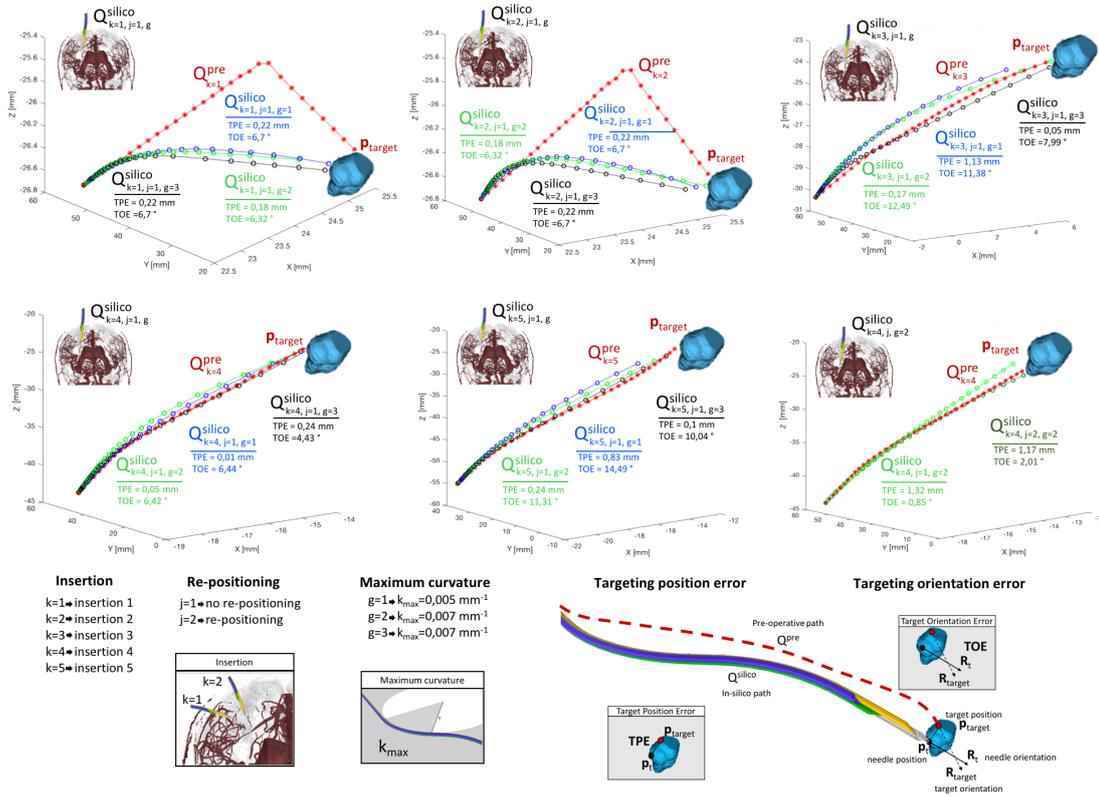


Figure 6.7: In-silico results, $Q_{k,j,g}^{silico}$ plotted as a function of the path, k , (in-silico tests were conducted on a total of 5 path, with $1 \leq k \leq 5$), as a function of the experiment type, j , ($j = 1$ without re-positioning, $j = 2$ with re-positioning) and as a function of maximum curvature, g , (with $g = 1$ when $k_{max} = 0.005$, $g = 2$ when $k_{max} = 0.007$, and $g = 3$ when $k_{max} = 0.014$). All the graphs show the pre-operative path, Q_k^{pre} , and the target pose, q_{target} . Target position error TPE and Target orientation error TOE are reported for each result.

6.7.2 In-Vitro Experiment

Concerning in-vitro results, Table 6.4 shows a comparison between “IRL as Control” and “Re-planning as Control” in terms of TPE, TOE, SR and RT calculated over the $Q_{k=1,j,g=2}^{vitro}$. The “IRL as Control” with re-positioning approaches keep an higher TPE and TOE and a greater SR than the “IRL as Control” without re-positioning. Finally, the “IRL as Control” approach didn’t keeps comparable performance with respect to the “Re-planning as Control” approach concerning the TPE and TOE but keeping an extremely lower RT. In Figure 6.8 are shown in-vitro results, $Q_{k,j,g}^{vitro}$ plotted as a function of the path, k , as a function of the experiment type, j , and as a function of maximum curvature, g . Target position error TPE and Target orientation error TOE are reported for each result.

Table 6.4: In-vitro results

Method	Re-pos	$v(t)(mm/s)$	$k_{max}(mm^{-1})$	TPE (mm)	TOE ($^{\circ}$)	SR(%)	RT(s)
IRL as Control	no	0.83	0.007	3.88	8.01	100	/
IRL as Control	yes	0.83	0.007	5.19	7.36	100	0.02
Re-planning as Control	yes	0.4	0.014	1.81 ± 0.51	5.9 ± 1.42	100	0.51

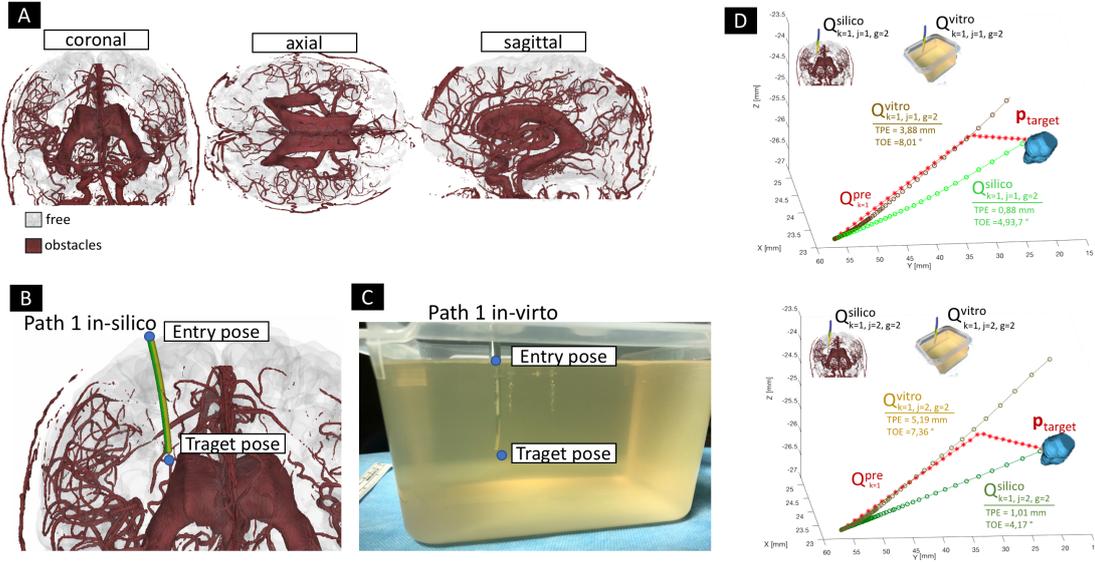


Figure 6.8: Dataset and In-vitro Results (A) Sagittal, axial and coronal view of the segmented brain objects are reported. In red the obstacles and in white the Gyri where the PBN can move freely. (B) In-silico insertion of the path with $k=1$. (C) In-vitro insertion of the path with $k=1$. (D) In-silico, $Q_{i,j,k}^{silico}$ and In-vitro, $Q_{i,j,k}^{vitro}$, results plotted as a function of the path, k , (in-vitro tests were conducted only for the first path, $k = 1$), as a function of the experiment type, j ($j = 1$ without re-positioning, $j = 2$ with re-positioning) and as a function of the maximum curvature, g (with $g = 2$ so $k_{max} = 0.007$). Both graphs show the preoperative path, Q_k^{pre} and the target pose q_{target} . In the first graph (above), in-silico and in-vitro trajectories are compared following the experiment without re-positioning, while in the second case the conditions are the same except that the experiment was conducted with re-positioning. Target position error TPE and Target orientation error TOE are reported for each result.

6.8 DISCUSSION

The simulation results of TOE and TPE comply with the corresponding tolerance values of 10° and 2.5 mm set for this type of experiment. Moreover, the results obtained are close to those of the “Re-planning

as Control” approach, supporting its possible application in intra-operative neurosurgical procedures. Furthermore, it is strict about noticing that, we adapted our system to a higher velocity. This means that with a lower velocity value (as the one used “for Re-planning as Control”), the “IRL as Control”, increasing the number of re-planning and consequently the accuracy, could decrease the error in position and orientation .

While the TPE and TOE reported for in-vitro experiments are considered acceptable for the SOTA [177], in fact, SR underlines the goodness of the proposed method.

Considering the real-time applicability, we have measured the RT. For the worst-case scenario, we have observed that it takes 0.02 sec to produce a new location and update the trajectory for the control pose of the IRL as Control’s agent. The exceptional RT value proves that the “IRL as Control” can work adequately in dynamic real-time scenarios.

The comparison between our method and the direct optimisation method (“Re-planning as Control”) shows that, in a dynamic environment and during intra-operative planning, where execution time is essential, the proposed method is more efficient in terms of re-planning time. Our control approach can avoid both static and dynamic obstacles. Moreover, contrary to methods based on direct optimisation, the spatial and temporal complexity of the proposed approach does not increase with dynamism.

6.9 CONCLUSION

A robot for keyhole neurosurgery procedures can minimise the effort spent by the doctors and help perform the surgery with minimum invasiveness. The development of an autonomous needle insertion system has the potential to reduce surgical time and improve accuracy in positioning. This study represents the first attempt to use an inverse reinforcement learning approach to control a complex needle steering system, where unmodelled dynamics between needle and tissue are hard to describe by classical mechanical approaches.

We demonstrated that a virtual training environment provide a good first approximation for learning control strategy to drive such complex system. The “IRL as Control” was successfully integrated with the real physical robotic system. For experimentation, the EDEN2020 robotic suite for keyhole neurosurgery that controls a new developed bio-inspired programmable bevel tip needle was used for this purpose.

The “IRL as Control” model underwent experiments with the same conditions in both simulation and the physical robotic system, of which performances were compared with previous works using the same needle model [209, 275]. In a simulated environment, we bench-marked our approach with the literature that uses path “Re-planning as Control” with geometric techniques. The results demonstrated that the proposed control framework could steer the flexible needle by performing better in success rate and computational time while maintaining the same standard of accuracy and robustness in target position and orientation error.

The results of the in-vitro assessment confirm that the policy learned from a simplified needle simulation environment through experiments can be applied to a real physical robotic system, but the differences between the simulation and the real physical system has impacted the control performances with a lower success rates and not optimal error distance between the tip and the target.

The “IRL as Control” approach was successfully applied in other contexts, such as suturing and knot tying [26, 123] where demonstrated improvements by overcoming some limits of geometric-based control approaches, such as providing sequential decision-making under uncertainties through end-to-end learning without the need to model the system uncertainty. However, the “IRL as Control” approaches has still limitation on "transfer and learn", due to the gap between the simulated environment and physical system. In future developments, to reduce this gap, we will acquire expert surgeon demonstrations on real physical robotic system, with a setup that will include ex-vivo samples to achieve a real mock of KN surgeries.

In conclusion, we would like to point out that we are aware that the use of only one subject limits the presentation of the results. As future work, we would like to extend the study to more subjects to analyse the generalisability of the control model produced.

CHAPTER 7

Conclusion

Over the last two decades, different research groups have focused their efforts on the development of needles able to autonomously steer inside the tissue. These needles perform curvilinear trajectories planned to maximise the distance from sensitive anatomical structures to be avoided and reach targets otherwise inaccessible via rectilinear insertion paths. Accurate placement of the needle tip inside tissue is challenging, especially when the target moves and anatomical obstacles must be avoided. Moreover, the complex kinematics of steerable needles make the insertion challenging, thus requiring the aid of automatic path planning solutions and control strategy.

This PhD dissertation presents an overview of different path planning techniques used for surgical steerable needle/catheters, a pre-operative path planner, a realistic simulator that mocks brain deformations due to catheter insertion, a novel intra-operative path re-planner and motion controller that can steer a PBN to produce the desired trajectory. The solutions were developed and tested, considering the programmable bevel tip needle as a case study. The latter consists of a multi-segment steerable needle under development within the EU EDEN2020 project and designed for percutaneous intervention in neurosurgery (drug infusion and in-situ optical diagnosis).

In Chapter 2, the planning problem for steerable needles in minimally invasive surgery is contextualised with a systematic review of the methods proposed in the literature for the computation of curvilinear paths.

In Chapter 3, a hybrid inductive learning / deductive reasoning 3-D path planning method for optimized paths computation is proposed (**Hypothesis 1**). The solution is validated through simulated insertions in neurosurgical scenarios demonstrating improvements in terms of path length, curvature and obstacle avoidance with respect to state of the art (**Hypothesis 1.1**) as well as computational time, safety distance-from-obstacles and smoothness suitable for pre-operative planning (**Hypothesis 1.2**).

Chapter 4 addresses the problem of simulating a dynamic environment that mocks brain deformations due to catheter insertion (**Hypothesis 2**). The method was validated using recorded brain deformation of in-vivo animal KN insertions of a PBN demonstrating a close match with real brain deformation (**Hypothesis 2.2**), as well as stability, accuracy, and real-time performance suitable to maximises the probability of intra-operative path planning success by accounting for uncertainty in needle-tissue and tissue-tissue deformation.

In Chapter 5, an inverse reinforcement learning method to support neurosurgeons during the intra-

operative procedure to react to a KN dynamic environment is presented (**Hypothesis 3**). The solution is validated in simulation in a dynamic environment demonstrating improvements in terms of fast re-planning and secure needle steering by accounting for a constrained target pose, intra-operative anatomical deformation, and accuracy and robustness suitable to maximise the probability of intra-operative path planning success.

Chapter 6 addressed the problem of steering a needle to produce the desired trajectory (**Hypothesis 4**). An IRL control method is proposed. The motion controller and control strategy are validated through in-silico trials and in-vitro experiments in phantom-brain gelatin, demonstrating that the needle is inserted following the desired trajectory (**Hypothesis 4.1**), as well as the accuracy and robustness can be kept consistent with standard needle insertion control strategies (**Hypothesis 4.2**).

7.1 Thesis contribution

This PhD research led to:

- *A systematic literature review on the planning problem for steerable needles in minimally invasive surgery for the computation of curvilinear paths.*

The presented systematic review contributes to provide an integrated, synthesized overview of the current state of knowledge. The great possibilities associated with steerable needles in terms of therapy outcome and patient safety make path planning a topic of great interest for researchers and engineers. Many solutions have been proposed to cope with this task grouped in graph-based, sampling-based, mathematical-based, learning-based, reasoning based and multifusion-based solutions. The structure of the analysis and the proposed synthesis could allow the reader to quickly identify the literature works that may represent their gold standard suitable for applications in their clinical context. In addition, thanks to the synthesis performed and the immediate mode of exposure proposed, it could give the possibility to identify more quickly some aspects to be improved such as real-time planning, complete information expressing, and complex environment modelling.

- *A new pre-operative curvilinear path planner, exploiting an expert's knowledge, able to compute, in a time consistent with standard pre-operative path planners, a kinematically feasible path for a steerable needle in accordance with the optimization criteria of minimum path length and curvature, and maximum distance from obstacles.*

The presented pre-operative curvilinear path planner for steerable needles succeeds in obtaining the optimal paths that can be followed to reach a specific target according to rules set by an expert. This approach allows to fully exploit an expert's knowledge. It is worth noting that one of the main contributions of the present work consists in the integration of an inductive learning-based approach with a deductive reasoning-based approach. The inductive learning-based method allows the agent to learn the policy by a set of demonstrations provided by an expert, who can introduce in a path planning algorithm all his requirements and knowledge that cannot always be possible in graph- or sampling-based approaches unless further optimisation steps are applied requiring additional computational time. Explicit programming cannot fully cover the complexity of the environment (represented by the human brain in this case), the number of parameters and possible complications that have to be considered during the path planning. For this reason, we implemented a deductive reasoning classifier with a user interface where the experts can express their individual preferences by assigning different weights, thus creating a priority list for maintaining different path planning optimisation criteria (i.e., giving more priority to path safety than path length) while visualizing the trajectory and changing the criteria in real time. When tested in simulation against a literature approach, the proposed method performed better in terms of path smoothness and clearance from safety regions, significantly decreasing the length and with a sensibly lower computational time.

- *A realistic, time-bounded simulator that mocks brain deformations during keyhole surgical procedures, where a catheter/needle is inserted into the brain.*

The presented simulator is able to account for the brain's dynamic behavior during keyhole neurosurgery. Thanks to its direct manipulation of positions, the PBD approach can efficiently handle collision constraints. Probe-tissue interaction can thus be effectively treated as a collision problem, allowing to deal with any input probe position without requiring the explicit definition of the contacting surface. The same does not apply to FEM simulations, where the enforcement of contact constraints would introduce degradation of the performances and stability issues. The simulator's numerical model has used a novel approach with respect to the literature, and it has proved to be a close match with real brain deformations through validation using recorded deformation data of in-vivo animal trials. The simulator represents a contribution to the development and training of AI systems such as autonomous control or intra-operative path planners in the context of KN.

- *A fast and accurate intra-operative path re-planner able to react to a KN dynamic environment while optimizing surgical criteria*

The presented IRL method is able to perform intra-operative re-planning of a trajectory in the dynamic needle-tissue interactive KN environment. The advantage of using a learning-based approach is that IRL realises sequential decision making under uncertainties through end-to-end learning without the need for modeling the system uncertainty. Moreover, with the exploitation of manual demonstrations, human expertise is embedded in the planning system. The algorithm can smoothly plan a path intra-operatively while meeting constraints on both target position and orientation. Furthermore, this system reacts smoothly to ongoing tissue-tissue and needle-tissue deformations. Its ability to compensate for local changes, while maintaining constraints on its global objectives, removes the risk of long path deviations that affect the most common sampling-based techniques proposed in the literature. When tested in simulation against other learning-based approaches and a sampling-based approach proposed in literature, the proposed method performed better in terms of success rate and re-planning time while maintaining the standard of accuracy and robustness in terms of target position and orientation error.

- *A robust strategy to control the deflection of a flexible needle to achieve an accurate KN robot-assisted insertion process* The presented IRL approach is able to control a catheter that moves in a brain model using a robot, exploiting its capacity to solve problems that are difficult to analytically model and explicitly observe. A simulation environment and catheter model similar to an actual robot environment are designed to learn catheter model control by making the simulation move the model to the designated target. This control model is then implemented into an actual robot to observe the success rate of robot control. The advantage of the proposed approach is that the surgeon's decision could be involved in the model updates and transfer learning to generate a patient-specific treatment program. When tested in simulation, against other control strategies based on kinematic/dynamic model of the needle and strategies that use path re-planning as control, the proposed control framework can steer the flexible needle in 3D model under continuous state space performing better in terms of success rate and computational time while maintaining the same standard of accuracy and robustness in terms of target position and orientation error. Based on the computer simulation, the proposed control framework can achieve high accuracy and robustness in steering flexible needle insertion. Based on in-vitro results, the study confirms that the policy learned from a simplified catheter simulation environment through experiments can be applied to the actual robot to control the catheter model movements. But, due to the differences between the simulation and the actual environment, the models applied to the robot have lower success rates and maintain a further average distance between the tip and the target.

7.2 Impact and future perspective

The PhD dissertation proposed methods to address two open issues in the automation of steerable needle minimally invasive interventions: the path planning and the motion control. To address MIS automatic pre- and intra-path planning and control problems, we proposed RL exploiting its capacity to solve problems that are difficult to analytically model and explicitly observe. To effectively reduce the learning

time, we chose IRL approaches by initialising them with policies learned from human expert demonstrations. Instead of learning from scratch, the robot improves the initial policy based on the demonstrations to reproduce the desired surgical tasks. To do so, GAIL agents were trained to imitate latent patterns that existed in human demonstrations, which can deal with the mismatch distribution caused by multi-modal behaviours.

The proposed IRL methods endow robots to autonomously execute path planning and motion control tasks. However, it is unfeasible for MIS procedures to repeat the experiments on the surgical robotic platform for over a million times during the training phases. To this end, we were able to implement a PBD brain tissue simulator and demonstrate, that the agent can be first trained in a simulation environment and transferred to a real robotic system.

As a case study, we have chosen KN because, in our opinion, it represents one of the most challenging MIS procedures and the PBN because it has one of the most challenging steering design systems. The fact that we have managed to apply the proposed approaches to a delicate surgical environment and a complex steering system design means that the proposed solutions can potentially be used for different MIS procedures and with needles featuring different designs and/or steering mechanisms (e.g., standard bevel-tip needles or duty-cycle bevel-tip needles), provided that the needle kinematic limits are known.

Based on the approaches and results presented in this PhD thesis, we point out two promising research directions: the development of autonomous surgical systems and of intelligent surgical training systems. We believe that these research lines are key to augment the capabilities of current surgical systems and to develop new tools that will benefit medical personnel and patients.

This PhD thesis demonstrated that model-free reinforcement learning is a promising direction toward generalizable automated surgical performance in minimally invasive procedure, but progress has been slowed by the lack of efficient and realistic learning environments. Moreover, beyond precision, robustness, safety and automation, it is also necessary to carefully consider the legal and ethical considerations of artificial intelligence in surgery.

8.1 Path Planning Systematic Review Keywords

A comprehensive list of the keywords used for the PRISMA search is reported in Table 8.1.

Table 8.1: *Keywords for Systematic Review*

Database	Query
SCOPUS	TITLE-ABS-KEY ("path planning" or "trajectory planning") AND ("flexible needle" OR "flexible catheter" OR "steerable catheter" OR "steerable needle ") AND ("pre-operative" OR "intra-operative") AND "minimal invasive surgery" AND "keyhole surgery") AND (LIMIT-TO (DOCTYPE , "ar") OR LIMIT-TO (DOCTYPE , "cp")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (EXACTKEYWORD , "path tracking") OR EXCLUDE (EXACTKEYWORD , "segmentation") OR EXCLUDE (EXACTKEYWORD , "path modeling"))
WEB OF SCIENCE	(("path planning" or "trajectory planning") AND ("flexible needle" OR "flexible catheter" OR "steerable needle" OR "steerable catheter") AND ("pre-operative" OR "intra-operative") AND "minimal invasive surgery" AND "keyhole surgery") NOT "segmentation" NOT "path tracking" NOT "path modeling") AND LANGUAGE: (English) AND DOCUMENT TYPES: (Article OR Proceedings Paper)
PUBMED	("path planning"[Title/Abstract/MeSH] OR "trajectory planning"[Title/Abstract/MeSH]) AND ("steerable needle"[Title/Abstract] OR "flexible needle"[Title/Abstract] OR "steerable catheter"[Title/Abstract] OR "flexible catheter"[Title/Abstract]) AND ("minimal invasive surgery"[All Fields] AND "keyhole neurosurgery"[All Fields]) NOT "segmentation"[Title/Abstract/MeSH] NOT "path tracking"[Title/Abstract/MeSH] NOT "path modeling"[Title/Abstract/MeSH] NOT Review[ptyp] AND English[lang]

8.2 Computing the radius of curvatures

In this appendix, the method used for computing the radius of curvature of the path is explained in algorithm 7. The method is based on the computation of the equation of the sphere that passes through four successive positions. The equation of a generic 3D sphere is determined by the equation 8.1 where the four coefficients D, E, F, G are unknown constant.

$$x^2 + y^2 + z^2 + D(x) + E(y) + F(z) + G = 0 \quad (8.1)$$

The radius of this sphere is the radius of curvature. First of all, it is important to note that four positions defined a single sphere if they are not coplanar [213]. In the opposite case, no spheres pass through the

four positions, or there are infinity numbers of them. For this reason, is necessary to verify if the four positions of the trajectory are coplanar. And this is true if and only if the determinant A of the matrix indicated in the algorithm is equal to zero. Since the positions are on the sphere, the substitution of their coordinates in equation 8.1 generates a system of four equations where the unknowns are the coefficients D, E, F, G [218]. For resolving this system the Cramer's rule is applied. Then the radius of the sphere can be computed.

8.3 Answer Set Programming

With ASP, computational problems in a large variety of scenarios can be described by means of simple and elegant logic programs consisting of a set of rules; solutions to a problem instance are then found by computing the semantics of such programs combined with the representation, usually expressed using factual rules, of the instance at hand.

One of the main advantages of ASP consists of its purely declarative nature: rather than focusing on algorithm design and coding, and thus on *how to solve* a computational problem at hand, with ASP one can focus on *how to describe* such computational problem (or how its solutions should look like), completely avoiding the need for explicitly express the steps to be executed. In turn, order of statements in a ASP logic program is immaterial: explicit updates in the problem specification can be more easily incorporated, thus fostering advantages such as fast prototyping, quick error detection and modularity. Besides, a clean model-theoretic semantics grants correctness; intuitively, an ASP program can be seen

Algorithm 7 Calculation of radius curvature

1: Input: four successive position of the path $\mathbf{p}_{agent_1}(x_1, y_1, z_1)$, $\mathbf{p}_{agent_2}(x_2, y_2, z_2)$, $\mathbf{p}_{agent_3}(x_3, y_3, z_3)$, $\mathbf{p}_{agent_4}(x_4, y_4, z_4) \in Q = \{\mathbf{T}_{agent_0}, \mathbf{T}_{agent_1}, \dots, \mathbf{T}_{agent_{n-1}}\}$.

2: Output: Ray of the sphere r_i that passes through the input positions.

3: $A = \det \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix}$

4: **if** $A == 0$ **then**

5: $r_i = 0$

▷ The four positions are coplanar

6: **else**

▷ Four positions determine a unique sphere if and only if they are not coplanar

7: $T = \det \begin{vmatrix} (x_1^2 + y_1^2 + z_1^2) & x_1 & y_1 & z_1 \\ (x_2^2 + y_2^2 + z_2^2) & x_2 & y_2 & z_2 \\ (x_3^2 + y_3^2 + z_3^2) & x_3 & y_3 & z_3 \\ (x_4^2 + y_4^2 + z_4^2) & x_4 & y_4 & z_4 \end{vmatrix}$

▷ D, E, F and G are determined through the Cramer's

rule

8: **for** $i = 1$ **in** 4 **do**

9: $t_i = -(x_i^2 + y_i^2 + z_i^2)$

10: **end for**

11: $D = \det \begin{vmatrix} t_1 & y_1 & z_1 & 1 \\ t_2 & y_2 & z_2 & 1 \\ t_3 & y_3 & z_3 & 1 \\ t_4 & y_4 & z_4 & 1 \end{vmatrix}$

$E = \det \begin{vmatrix} x_1 & t_1 & z_1 & 1 \\ x_2 & t_2 & z_2 & 1 \\ x_3 & t_3 & z_3 & 1 \\ x_4 & t_4 & z_4 & 1 \end{vmatrix}$

12: $F = \det \begin{vmatrix} x_1 & y_1 & t_1 & 1 \\ x_2 & y_2 & t_2 & 1 \\ x_3 & y_3 & t_3 & 1 \\ x_4 & y_4 & t_4 & 1 \end{vmatrix}$

$G = \det \begin{vmatrix} x_1 & y_1 & z_1 & t_1 \\ x_2 & y_2 & z_2 & t_2 \\ x_3 & y_3 & z_3 & t_3 \\ x_4 & y_4 & z_4 & t_4 \end{vmatrix}$

13: $C = -\left(\frac{D}{2}, \frac{E}{2}, \frac{F}{2}\right)$

▷ Coordinate of the centre of the sphere

14: $r_i = \frac{1}{2} \sqrt{(D^2 + E^2 + F^2 - 4G)}$

▷ Radius of the sphere

15: **end if**

as a formal yet executable description of the problem. The basic construct of ASP is a rule, that has a general form of $Head \leftarrow Body$; the $Body$ is a logic conjunction in which nonmonotonic negation may appear, and $Head$ can be either an atomic formula or a logic disjunction. Rules are interpreted according to common sense principles: roughly, the intuitive semantics of a rule corresponds to an implication. The answer set semantics associates a problem specification with none, one, or many intended models, called *answer sets*, each one corresponding to a solution; an ASP program that models a computational problem, coupled with a proper representation of an instance of such problem, can be fed to an implementation of ASP, called ASP solver, in order to actually compute all corresponding answer sets. Efficient and reliable ASP solvers exist, such as DLV/DLV2 [10, 162] and clingo [93].

For a full description of ASP syntax and semantics, along with examples of its applications in academy and industry, we refer the reader to [47, 170] and the vast literature. We briefly recall here some very basic preliminaries useful to understand the herein proposed approach.

A variable or a constant is a term. Variables are denoted by strings starting with some uppercase letter, while constants can either be integers, strings starting with some lowercase letter or quoted strings. If t_1, \dots, t_k are terms (either constants or variables) and p is a predicate symbol of arity k , then $p(t_1, \dots, t_k)$ is an atom of arity k .

A literal l is of the form a or $\text{not } a$, where a is an atom; in the former case l is positive, negative otherwise. A rule is of the following form:

$$a_0 | \dots | a_h : -b_1, \dots, b_n \text{ not } b_{n+1}, \dots, b_m. \quad (8.2)$$

On the left, the symbol “|” connects atoms that are part of a disjunction in the head, whereas comma separated literals in the right side, i.e., the body, are part of a conjunction. An ASP program is a finite set of rules.

A *fact* is a rule with empty body, and represents a piece of information known to be true (typically, facts stand for the knowledge granted before reasoning, or represent the instance of a problem); usually, a fact is immediately followed by the “.” symbol (i.e., the implication symbol “:-” is omitted). A constraint is a rule with empty head; hard (“strong” or “classical”) and soft (“weak”) constraints can be specified in order to cut out undesired models and express preferences, respectively. Weak constraints are expressed with the symbol $:\sim$ instead of $:-$, that is one used for hard constraints. These latter are conditions that must be satisfied, whereas soft constraints represent conditions that should be fulfilled; intuitively, when a solution violates a soft constraint it pays a cost: this induces an ordering among solutions that allows one to express minimization and/or maximization criteria.

ASP enjoys several additional language features for easing knowledge representation; we mention here *choice rules*, that are a compact way for expressing disjunction of atoms that must adhere to some cardinality conditions and *aggregates*, that can be used for compact representations of properties and inductive definitions using sets of propositions [19]. The scientific community agreed on a standard language [47]; furthermore, in addition to the standard, several flavours of ASP are supported by solvers, featuring additional constructs such as the `#minimize` and `#maximize` statements for expressing preferences in optimization problems similarly to what can be done via weak constraints.

ASP is a very expressive formalism; indeed, in [68] it is proved that disjunctive logic programs under answer set semantics capture the complexity class Σ_2^P (that is, they allow us to express every property which is decidable in non-deterministic polynomial time with an oracle in NP), and weak constraints make ASP well-suited to represent a wide class of problems (including, e.g., NP optimization problems) in a very natural and compact way [40].

The following example briefly illustrates how the common “Guess&Check” paradigm is used for modelling problem with ASP.

Example 1. *Let us consider the well-known problem of 3-colorability, which consists of the assignment of three colors to the nodes of a graph in such a way that adjacent nodes always have different colors; this problem is known to be NP-complete [38]. Suppose that the nodes and the arcs are represented by a set \mathcal{F} of facts with predicates *node* (unary) and *arc* (binary), respectively. Then, the following ASP program allows us to determine the admissible ways of coloring the given graph with the three given colors.*

```

r1: color(X, red) | color(X, green) | color(X, blue) :- node(X) .
r2: :- arc(X, Y), color(X, C), color(Y, C) .

```

Rule r_1 (guess) above states that every node of the graph must be colored as red or green or blue; r_2 (check) forbids the assignment of the same color to any couple of adjacent nodes. The minimality of answer sets semantics guarantees that every node is assigned only one color. Thus, there is a one–one correspondence between the solutions of the 3-coloring problem for the instance at hand and the answer sets of $\mathcal{F} \cup \{r_1, r_2\}$: the graph represented by \mathcal{F} is 3-colorable if and only if $\mathcal{F} \cup \{r_1, r_2\}$ has some answer set.

The following example illustrates the use of weak constraints for expressing preferences while dealing with optimization problems.

Example 2. Let us consider the same 3-colorability problem of Example 1. Imagine that we know that some of our graphs are not 3-colorable; as already stated, for such graphs our programs would have no answer sets (meaning that there is no admissible solution). Nevertheless, for such cases we would like to have admissible colorings, even if not complete. The following program, that is slightly different from the previous, allows us to determine all partial colorings.

```

r3: color(X, red) | color(X, green)
    | color(X, blue) | noColor(X) :- node(X) .
r2: :- arc(X, Y), color(X, C), color(Y, C) .

```

Here, rule r_1 of Example 1 is replaced with rule r_2 ; note that now, for the program $\mathcal{F} \cup \{r_3, r_2\}$, admissible answer sets exist with some nodes with no color assigned. These contain answer sets with no node colored at all, others with some colored, and up to others with all node colored, in case the graph is 3-colorable. We can state that we prefer a solution over another one if it features a higher number of colored nodes by means of the following weak constraint:

```

r4: :-~ node(X), noColor(X) . [ 1@1, X]

```

Now, for a solution of $\mathcal{F} \cup \{r_3, r_2, r_4\}$, while if it violates r_2 is inadmissible, and thus discarded, if it violates r_4 then it is assigned a cost as specified in the square brackets. In this latter case, for each node X that has no color, the solution costs 1 at level 1. Weight and level can be constant values or variables appearing in the body of the constraint; furthermore, weights are additive, grouped by levels, among all constraints in the program: in this simple example, just a weak constraint is present, but one can make use of more in order to express complex sets of desiderata. Higher levels correspond to more important desiderata.

8.4 Position Based Dynamic

8.4.1 Position-based Dynamics Approach

The PBD method is a simulation approach that omits the velocity and acceleration layer, which are instead taken into account by the most common simulation algorithm of object's deformations to focus on positions. By applying this approach, the objects are represented by a particle system, a set of particles that might be a single point in the space or, like in our case, a sphere characterised by a radius. This characteristic differentiates the PBD approach from classical dynamics approaches used in the world of graphics that formulate the change of momentum of a system as a function of applied forces, and evolve positions through numerical integration of accelerations and velocities, generally characterising objects with meshes (*i.e.*, vertices and information on how the vertices are connected) or other surface representation. PBD, instead, compute the time evolution of a dynamic system by directly updating the positions of the particles that composed the objects, as first described by Muller et al. in [187]. The meshless representation involves that the particles keep track of their original position, goal position, and mass. Indeed, simulated objects are discretised like a set of particles, particularly by a set of N particles

and a set of M constraints. Each particle i is described by position \mathbf{p}_i , velocity \mathbf{v}_i and mass \mathbf{m}_i . The initial and final positions are essential because they are needed to calculate the particle's displacement and know the precise particle's final location. The mass is required in order to calculate the centre of mass of the object. A constraint j is defined by the five attributes: cardinality n_j , scalar constraint function $C_j : \mathbb{R}^3 n_j \rightarrow \mathbb{R}$, set of indices $\{i_1, \dots, i_{n_j}\}, i_k \in [1, \dots, N]$, stiffness parameter $k_i \in [0, \dots, 1]$, type unilateral or type bilateral. In particular, in the last case, constraint j with type *bilateral* is satisfied if $C_j(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{n_j}}) = 0$. If its type is unilateral then it is satisfied if $C_j(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{n_j}}) \geq 0$. The stiffness parameter k_j defines the strength of the constraint in a range from zero to one [30].

Given this data and a time step Δt , the simulation proceeds as described by Algorithm 8. The positions \mathbf{x}_1 and the velocities \mathbf{v}_1 of the particles, in the first instant of time, need to be specified (1)-(3) because the algorithm simulates a system that is second order in time. These steps must be performed before the simulation starts to ensure a correct initialisation. An explicit forward Euler integration step on the velocities and the positions estimates for each particle is performed in lines from (5) to (11), taking into account the external forces applied to the system. In particular, line (6) allows external forces to be attached to the system if some of the forces cannot be converted into position constraints. The new locations \mathbf{p}_i are not assigned to the positions directly but are only used as predictions. Moreover, speeds can be damped if necessary, as shown in line (8). In line (13), non-permanent constraints are generated. It is the case of possible collisions with other objects which change from step to step during the simulation, and therefore they need to be generated at the beginning of each time step. It is performed continuous collision detection through the use of the original and predicted positions of each particle. The solver (15)-(18) then iteratively corrects the predicted positions such that they satisfy the M_{coll} external as well as the M internal constraints. Finally, the corrected positions \mathbf{p}_i are used to update the positions (21) and the velocities (20). Unlike traditional schemes, these integration steps do not extrapolate blindly into the future, but move the vertices to a physically valid configuration that the constraint solver calculates.

Algorithm 8 Position-based dynamics

```

1: for all vertices  $i$  do
2:   initalize  $\mathbf{x}_i = \mathbf{x}_i^0, \mathbf{v}_i = \mathbf{v}_i^0, w_i = 1/m_i$ 
3: end for
4: loop
5:   for all vertices  $i$  do
6:      $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{ext}(\mathbf{x}_i)$ 
7:   end for
8:   dampVelocities ( $\mathbf{v}_1, \dots, \mathbf{v}_N$ )
9:   for all vertices  $i$  do
10:     $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$ 
11:   end for
12:   for all vertices  $i$  do
13:    genCollConstraints ( $\mathbf{x}_i \leftarrow \mathbf{p}_i$ )
14:   end for
15:   loop
16:     solverIteration times
17:     projectConstraints ( $C_1, \dots, C_{M+M_{Coll}}, \mathbf{p}_1, \dots, \mathbf{p}_N$ )
18:   end loop
19:   for all vertices  $i$  do
20:     $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i) / \Delta t$ 
21:     $\mathbf{x}_i \leftarrow \mathbf{p}_i$ 
22:   end for
23:   velocityUpdate ( $\mathbf{v}_i, \dots, \mathbf{v}_N$ )
24: end loop
    
```

For this reason, the scheme is unconditionally stable. However, it is not possible to classify this integration as an explicit or implicit scheme because if only one iteration of the solver is performed for each time step, this will behave more like an explicit scheme, but, as the number of iterations increase,

the algorithm behaves more like an implicit scheme. The velocities are updated in line (23) because they are the velocities of colliding vertices, so it is necessary to modify them according to friction and restitution coefficients, along with the positions. In this way, the simulation will produce the correct behaviour of a second-order system.

The Solver

The algorithm predicts that an object modelled with PBD starts the simulation unaltered, *i.e.*, in the basic shape it should have, as shown graphically in (a) in Figure 8.1.

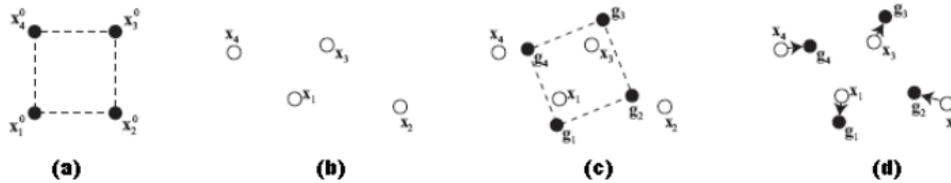


Figure 8.1: *Graphic Representation of The General Idea of The PBD Algorithm, (a), (b), (c) and (d) illustrate the four main phases of the algorithm (courtesy of Müller et al. [187].*

The particles lose their shape and change their position after applying a force or after the collision with another object (b). In particular, the velocity and acceleration of these particles will be altered due to these external stimuli. In the case of gravity, for example, all particles will have an almost equal alteration of the parameters, while in the case of external forces affecting a circumscribed portion of the object, only some particles will move, and others will remain in their initial position or will be less affected by this force. The goal of the position-based solver is to satisfy the equation $C(\mathbf{p} + \Delta\mathbf{p}) = 0$ finding a $\Delta\mathbf{p}$ that corrects the particles positions. For that reason, the constraints $C_M + C_{M_{coll}}$ and the estimates $\mathbf{p}_1, \dots, \mathbf{p}_N$ of the new positions the points should have are the inputs to the solver. Through a system of nonlinear equations, the solver will try to modify these estimates so that they satisfy all the constraints. The equations will be non-linear because even a simple distance constraint $C(\mathbf{p}_i, \mathbf{p}_{i+1}) = |\mathbf{p}_i - \mathbf{p}_{i+1}| - d$ will produce a non-linear equation and the constraints that place an inequality between the various positions will, in turn produce, inequalities. Therefore, an iteration inspired by the Gauss-Seidel algorithm is used to solve this system of a certain complexity, *i.e.*, each constraint will be solved independently of the others, and in succession, it will consider the positions changes of previous constraints. The Gauss-Seidel algorithm is not used in its entirety because it only requires linear equations to be solved, whereas in the case analysed there is non-linearity. It repeatedly iterates through all the constraints, and then it projects the particles to valid locations concerning the given constraint, considering one at a time, as shown in (c). For a single constraint, the position correction $\Delta\mathbf{p}$ is computed by solving the linearised equation:

$$C(\mathbf{p} + \Delta\mathbf{p}) \approx C(\mathbf{p}) + \nabla_{\mathbf{p}} C^T(\mathbf{p}) \Delta\mathbf{p} = 0 \quad (8.3)$$

The position correction are performed only in the direction of $\nabla_{\mathbf{p}} C(\mathbf{p})$ the linear momentum and the angular momentum are conserved, satisfying the principle of virtual work. The corrections $\Delta\mathbf{p}_1, \dots, \Delta\mathbf{p}_N$ are weighted according to the inverse masses $w_i = 1/m_i$, as shown in this equation:

$$\Delta\mathbf{p}_i = w_i \lambda \nabla_{\mathbf{p}_i} C(\mathbf{p}) \quad (8.4)$$

where λ is a Lagrange multiplier. In this way the changes and displacements made to the points will be immediately visible to the operator from a graphical point of view, which would not have been possible if a Jacobian algorithm, which is very popular for this type of operation, had been used. Finally, in (d), the particles will convert their own original shape into the current shape and they will move to their respective target positions where the applied external stimulus actually moved them, as shown in Figure 8.2.

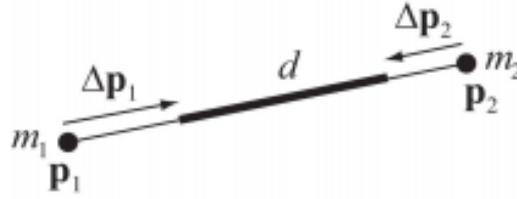


Figure 8.2: Projection of the constraint $C(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2| - d$, where the corrections $\Delta \mathbf{p}_i$ are weighted according to the inverse masses $w_i = 1/m_i$ (courtesy of Müller et al. [188]).

This process is repeated at each instant throughout the simulation. Even though the process can lead to oscillations if the order is not kept constant in over-constrained situations, this significantly accelerates convergence because pressure waves can propagate through the material in a single solver step, which affects that depends on the order in which the constraints are solved. It is possible to introduce energy function $\mathbf{E}(\mathbf{p})$ constraints by imposing $\mathbf{C}(\mathbf{p}) = \mathbf{E}(\mathbf{p}) = 0$ for each energy function, which are based on continuum mechanics. The advantage of using energy functions instead of geometric constraints is that thanks to the continuum-based formulation we use, physical phenomena like a lateral contraction, anisotropy or elastoplasticity can be simulated since the characteristic deformation behaviour of these effects are encoded in the energy functions and their gradients, as described by Jan Bender et al. in [29].

Collision handling

One advantage of the PBD approach is the simplicity with collision response can be managed. This plainness is especially important in the field of minimally invasive surgery in which this project is involved: managing collisions between the catheter and the rest of the environment, and between the environment itself, in a quickly and reliably way makes this method particularly suitable for real-time applications. Dealing with collisions in a virtual simulation involves analysing when objects enter into a collision and determining which objects, if any, are actually colliding with each other, *i.e.*, *collision detection*, and how objects respond to that specific event, *i.e.*, *collision response*, applying appropriate forces to colliding objects in order to make their behaviour as realistic as possible. In line (17) of the Algorithm 8 the M_{coll} collision constraints are generated at each time step, while the first M constraints, generated by the representation of the object, are fixed during all the simulation time. The number of M_{coll} depends on the number of colliding vertices and varies during the simulation, as described in [188]. The collisions could be both continuous and static. For continuous collision handling, it is necessary to test for each vertex i if the ray $\mathbf{x}_i \rightarrow \mathbf{p}_i$ is entering into an object in order to understand if there is any collision with that specific object. If this is the case the entry point \mathbf{q}_c and the surface normal \mathbf{n}_c at this position are computed and an *inequality* constraint with constraint function $C(\mathbf{p} = (\mathbf{p} - \mathbf{q}_s) \cdot \mathbf{n}_c$ and stiffness $k = 1$ is added to the list of constraints. The case just stated is the case of continuous collision, where the ray $\mathbf{x}_i \rightarrow \mathbf{p}_i$ do not lie entirely in the object. Instead, continuous collision detection has failed if this is not the case, and static collision must be managed. The surface point \mathbf{q}_s which is closest to \mathbf{p}_i point and the surface normal \mathbf{n}_s at this position are computed, adding to the list of constraints also in this case the *inequality* constraint calculated above and the stiffness coefficient $k = 1$. In order to make the simulation faster, the generation of collision constraint is done outside of the solver loop. The colliding vertices' velocities can be modified in step (23) of the Algorithm 8 since they also deal with friction and restitution behaviours. In the case in which we have dynamic collision, in other words there is an impulse that is transferred to the collision partners, the rules handling above must be reconsidered. If we have two dynamic colliding objects it is possible to obtain the correct response by simulating both objects with N vertices and M constraints, which are the input to the algorithm and simply represent two or more independent objects. In this instance the *inequality* constraint that manages the movement of a point \mathbf{q} of an objects in the area described by three points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ belonging to the other object is characterised by constraint function $C(\mathbf{q}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \pm(\mathbf{q} - \mathbf{p}_1) \cdot [(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)]$, which keeps the point \mathbf{q} on the correct side of the triangle conserving the linear and angular momentum. Collisions can be missed if the solver works with

a fixed collision constraint set in a particular scenario, but the artefacts are negligible according to [188].

Shape matching technique

In the PBD approach, deformation computation becomes a constraint-function optimization problem. The problem is equivalent to finding the minimum change in kinetic energy that satisfies the constraints, which is consistent with Gauss’s principle of least constraint [175], as shown in the following equations:

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \Delta \mathbf{x}^T \mathbf{M} \Delta \mathbf{x} \\ &\text{subject to} && C_i(\mathbf{x} + \Delta \mathbf{x}) = 0, \quad i = 1, \dots, n \end{aligned} \quad (8.5)$$

This implementation follows that soft bodies simulation behaviour and performance are not only influenced by the relative position, dimension, and number of particles in space but also by the constraints acting among particles. For example, large deformations of soft bodies are usually achieved by defining positional constraints among adjacent particles’ rigid clusters. This kind of constraint is called *region-based shape matching*, as described in [30]. In order to model the elastic deformation of the considered object in a way that could give it a characteristic more or less soft behaviour, there is a stiffness parameter for each shape-matching constraint. It is detected when a particle belonging to a rigid shape has been deformed beyond a selected threshold. Whether this has happened, this deformation is mixed back into the particle’s local-space rest position according to a plastic creep coefficient. Concerning convergence, Shape-matching convergence is dependent on the total number of particles in every shape. This yoke makes the PBD with the shape matching method integration more suitable to smaller rigid objects and less suitable for large shapes. Buoyancy is also affected by the number of particles in the shape because the more particles there are, the slower the convergence is, making the rigid body behave as if it were heavier than what it is. Consequently, our mass ratios do not correspond to real-world submergence depths.

As explained in this section, the final result is a method capable of representing most of the characteristic behaviours that an object, in reality, can assume, for example, fluids, cloths, soft bodies, rigid bodies, even coexisting in the same scene. In our case, to model the brain, we will focus on soft materials, which simulate the natural behaviour of grey and white matter, as if it were jelly.

8.5 Deep Reinforcement Learning

RL [251] is an interesting field of ML that’s attracting a lot of attention and popularity. It requires that the machine systems understand the characteristics of the surrounding environment. In RL, an agent interacts with an environment (ε). At each time step (t), the agent receives, from the environment, its current state (s_t) and selects an action (a_t) from a set of possible actions (A), according to its policy (π), such that $\pi(s_t) = a_t$. In response, the agent receives the next state (s_{t+1}) and a scalar reward (r_t), according to a predefined reward function. The goal of the agent is to determine an optimal policy π^* allowing it to take actions inside the environment, maximizing the overall cumulative reward. We analyzed and tested the performances of two DRL algorithms: SAC and PPO.

8.5.1 SAC

SAC is one of the most efficient model-free algorithms utilized in real-world robotic learning [106]. SAC is an off-policy model-free DRL algorithm which means it can learn from experiences collected at any time during the past. SAC is based on the maximum entropy reinforcement learning framework, which considers the entropy augmented objective:

$$J(\pi) = \mathbb{E}_{\pi} \left[\sum_t r_t(s_t, a_t) - \alpha \log(\pi(a_t | s_t)) \right] \quad (8.6)$$

The optimal policy maximizes the expected return (first summand) and the expected entropy (second summand). The trade-off between the two is controlled by the non-negative temperature parameter α , which can be learned automatically instead of treating it as a hyperparameter.

8.5.2 PPO

PPO is a policy gradient method for reinforcement learning which represents a balance between ease of implementation and tuning, and sample complexity, trying to compute an update at each step that minimizes the cost function while ensuring a relatively small deviation from the previous policy [233]. The most up-to-date variant of the algorithm uses the “clip” objective function not typically found in other algorithms:

$$L(\pi) = \hat{E}_t \left[\min \left(c_t(\pi) \hat{A}_t, \text{clip} \left(r_t(\pi), 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_t \right) \right] \quad (8.7)$$

where \hat{E}_t denotes the empirical expectation over timesteps, c_t is the ratio of the probability under the new and old policies, respectively, \hat{A}_t is the estimated advantage at time t , and ε is a hyperparameter, usually 0.1 or 0.2.

8.6 Inverse Reinforcement Learning

IRL is a recently developed ML framework in which the reward function is inferred by the expert demonstrations and then used with RL methods to optimize it [7]. IRL seeks a valid policy (π) and a reward function ($R(\tau)$) under which the expert’s behaviour is ideal by optimizing the agent π in the inner loop while searching for the $R(\tau)$ in the outer loop using expert’s trajectories.

8.6.1 GAIL

GAIL rewards the agent for behaving similarly to a set of demonstrations using an adversarial approach [115] in which the discriminator NN is used to distinguish between demonstration and agent actions. During the training process, the agent gets better at mirroring the demonstrations, while the discriminator gets stricter learning how to differentiate the two different types of demonstrations. In this way, the final policy (π) produces states (s) and actions (a) similar to the demonstrations. The goal of GAIL is to find the saddle point of the Equation:

$$\min_{\pi} \max_D \mathbb{E}_{\pi} [\log D(x)] + \mathbb{E}_{\pi_E} [1 - \log D(s, a)] - \lambda H(\pi) \quad (8.8)$$

where π_E is the expert model or the expert trajectories generated by the expert model, $D(s, a)$ is the Discriminant network, and H is a policy regulariser where $\lambda > 0$.

8.7 Electronic Supplementary Material

8.7.1 Neurosurgical Simulator

Below is the link to the neurosurgical simulator: <https://youtu.be/OM4x4W91WkE>

8.7.2 Brain Deformable Tissue PBD Simulator

Below is the link to the brain deformable tissue PBD simulator: <https://youtu.be/yKEhqbJ8LGY>

Bibliography

- [1] American Brain Foundation. <https://www.americanbrainfoundation.org/diseases/>. Accessed: 2021-03-02.
- [2] Blender. <https://www.blender.org>. Accessed: 2021-02-01.
- [3] FreeSurfer. <http://surfer.nmr.mgh.harvard.edu>. Accessed: 2021-02-01.
- [4] ImFusion. <https://www.imfusion.com/>. Accessed: 2021-02-01.
- [5] Slicer. <https://www.slicer.org>. Accessed: 2021-02-01.
- [6] NVIDIA gameworks | Nvidia FleX, Online; Accessed June 24, 2020. <https://developer.nvidia.com/flex>.
- [7] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.
- [8] Niki Abolhassani, Rajni Patel, and Mehrdad Moallem. Needle insertion into soft tissue: A survey. *Medical engineering & physics*, 29(4):413–431, 2007.
- [9] Louaï Adhami and Ève Coste-Manière. Optimal planning for minimally invasive surgical robots. *IEEE Transactions on Robotics and Automation*, 19(5):854–863, 2003.
- [10] Weronika T Adrian, Mario Alviano, Francesco Calimeri, Bernardo Cuteri, Carmine Dodaro, Wolfgang Faber, Davide Fuscà, Nicola Leone, Marco Manna, Simona Perri, et al. The asp system dlv: advancements and applications. *KI-Künstliche Intelligenz*, 32(2-3):177–179, 2018.
- [11] Afsoon Nejati Aghdam and Peter Xiaoping Liu. A novel path planner for steerable bevel-tip needles to reach multiple targets with obstacles. *IEEE Transactions on Instrumentation and Measurement*, 69(10):7636–7645, 2020.
- [12] Mustafa Al-Khawaldeh, Ibrahim Al-Naimi, Xi Chen, and Philip Moore. Ubiquitous robotics for knowledge-based auto-configuration system within smart home environment. In *2016 7th international conference on information and communication systems (ICICS)*, pages 139–144. IEEE, 2016.
- [13] Jérémie Allard, Hadrien Courtecuisse, and François Faure. Implicit fem solver on gpu for interactive deformation simulation. In *GPU computing gems Jade Edition*, pages 281–294. Elsevier, 2012.
- [14] Ron Alterovitz, Michael Branicky, and Ken Goldberg. Motion planning under uncertainty for image-guided medical needle steering. *The International journal of robotics research*, 27(11-12):1361–1374, 2008.
- [15] Ron Alterovitz, Ken Goldberg, and Allison Okamura. Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 1640–1645. IEEE, 2005.
- [16] Ron Alterovitz, Andrew Lim, Ken Goldberg, Gregory S Chirikjian, and Allison M Okamura. Steering flexible needles under markov motion uncertainty. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1570–1575. IEEE, 2005.

- [17] Ron Alterovitz, Sachin Patil, and Anna Derbakova. Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 3706–3712. IEEE, 2011.
- [18] Ron Alterovitz, Thierry Siméon, and Ken Goldberg. The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. In *Robotics: Science and systems*, 2007.
- [19] Mario Alviano and Wolfgang Faber. Aggregates in answer set programming. *KI-Künstliche Intelligenz*, 32(2-3):119–124, 2018.
- [20] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, page 103500, 2021.
- [21] Ali Asadian, Mehrdad R Kermani, and Rajni V Patel. Robot-assisted needle steering using a control theoretic approach. *Journal of Intelligent & Robotic Systems*, 62(3):397–418, 2011.
- [22] Mahdiah Babaiasl, Fan Yang, Stefano Boccelli, and John P Swensen. Fracture-directed waterjet needle steering: Design, modeling, and path planning. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 1166–1173. IEEE, 2020.
- [23] Mahdiah Babaiasl, Fan Yang, and John P Swensen. Towards water-jet steerable needles. In *2018 7th IEEE international conference on biomedical robotics and biomechatronics (biorob)*, pages 601–608. IEEE, 2018.
- [24] Tariq Bahwini, Yongmin Zhong, and Chengfan Gu. Path planning in the presence of soft tissue deformation. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 13(4):1603–1616, 2019.
- [25] Sophia Bano, Seong Young Ko, and Ferdinando Rodriguez y Baena. Smooth path planning for a biologically-inspired neurosurgical probe. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 920–923. IEEE, 2012.
- [26] Yotam Barnoy, Molly O’Brien, Will Wang, and Gregory Hager. Robotic surgery with lean reinforcement learning. *arXiv preprint arXiv:2105.01006*, 2021.
- [27] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.
- [28] J Bender, D Koschier, P Charrier, and D Weber. Position-based simulation of continuous materials. *Computers & Graphics*, 44, 07 2014.
- [29] Jan Bender, Dan Koschier, and Daniel Weber Patrick Charrier. Position-based simulation of continuous materials. *Comput. Graph.*, 44(C):1–10, November 2014.
- [30] Jan Bender, Matthias Müller, Miguel A. Otaduy, Matthias Teschner, and Miles Macklin. A survey on position-based simulation methods in computer graphics. *Computer Graphics Forum*, 33(6):228–251, 2014.
- [31] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.
- [32] Mariana C Bernardes, Bruno V Adorno, Geovany A Borges, and Philippe Poignet. 3d robust online motion planning for steerable needles in dynamic workspaces using duty-cycled rotation. *Journal of Control, Automation and Electrical Systems*, 25(2):216–227, 2014.
- [33] Mariana C Bernardes, Bruno V Adorno, Philippe Poignet, Nabil Zemiti, and Geovany A Borges. Adaptive path planning for steerable needles using duty-cycling. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2545–2550. IEEE, 2011.
- [34] MC Bernardes, Bruno Vilhena Adorno, Philippe Poignet, and GA Borges. Robot-assisted automatic insertion of steerable needles with closed-loop imaging feedback and intraoperative trajectory replanning. *Mechatronics*, 23(6):630–645, 2013.
- [35] Eva Besada-Portas, Luis de la Torre, M Jesus, and Bonifacio de Andrés-Toro. Evolutionary trajectory planner for multiple uavs in realistic scenarios. *IEEE Transactions on Robotics*, 26(4):619–634, 2010.
- [36] Ge Bian, Michael Lipowicz, and Grant H Kruger. Self-learning of inverse kinematics for feedforward control of intracardiac robotic ablation catheters. In *2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 72–77. IEEE, 2015.
- [37] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.
- [38] Andreas Brandstädt, Van Bang Le, and Thomas Szyczak. The complexity of some problems related to graph 3-colorability. *Discret. Appl. Math.*, 89(1-3):59–73, 1998.
- [39] Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

- [40] Francesco Buccafurri, Nicola Leone, and Pasquale Rullo. Strong and weak constraints in disjunctive datalog. In Jürgen Dix, Ulrich Furbach, and Anil Nerode, editors, *Logic Programming and Nonmonotonic Reasoning, 4th International Conference, LPNMR'97, Dagstuhl Castle, Germany, July 28-31, 1997, Proceedings*, volume 1265 of *Lecture Notes in Computer Science*, pages 2–17. Springer, 1997.
- [41] E Clif Burdette, D Caleb Rucker, Punit Prakash, Chris J Diederich, Jordon M Croom, Clyde Clarke, Philipp Stolka, Titania Juang, Emad M Boctor, and Robert J Webster III. The acusitt ultrasonic ablator: the first steerable needle with an integrated interventional tool. In *Medical Imaging 2010: Ultrasonic Imaging, Tomography, and Therapy*, volume 7629, page 76290V. International Society for Optics and Photonics, 2010.
- [42] Jessica Burgner-Kahrs, D Caleb Rucker, and Howie Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.
- [43] C Burrows, R Secoli, and FR y Baena. Experimental characterisation of a biologically inspired 3d steering needle. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 1252–1257, 2013.
- [44] Christopher Burrows, Riccardo Secoli, and Ferdinando Rodriguez y Baena. Experimental characterisation of a biologically inspired 3d steering needle. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 1252–1257. IEEE, 2013.
- [45] Chiara Caborni, Seong Young Ko, Elena De Momi, Giancarlo Ferrigno, and Ferdinando Rodriguez y Baena. Risk-based path planning for a steerable flexible probe for neurosurgical intervention. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, pages 866–871. IEEE, 2012.
- [46] Chenxu Cai, Chunsheng Sun, Ying Han, and Qinhe Zhang. Clinical flexible needle puncture path planning based on particle swarm optimization. *Computer methods and programs in biomedicine*, 193:105511, 2020.
- [47] Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Marco Maratea, Francesco Ricca, and Torsten Schaub. Asp-core-2 input language format. *Theory Pract. Log. Program.*, 20(2):294–309, 2020.
- [48] M Camara, Susmita C., A Darzi, and P Pratt. Soft tissue deformation for surgical simulation: a position-based dynamics approach. *International Journal of Computer Assisted Radiology and Surgery*, 11:919–928, 2016.
- [49] Antonella Castellano and Andrea Falini. Eden2020 human brain mri datasets for healthy volunteers (0.1) [data set]. *Zenodo*, 2019.
- [50] Yuchou Chang, Xiaoyan Wang, Zhenzhou An, and Haifeng Wang. Robotic path planning using a* algorithm for automatic navigation in magnetic resonance angiography. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 734–737. IEEE, 2018.
- [51] Chiu-Hung Chen, Tung-Kuan Liu, and Jyh-Horng Chou. A novel crowding genetic algorithm and its applications to manufacturing robots. *IEEE Transactions on Industrial Informatics*, 10(3):1705–1716, 2014.
- [52] Yuyang Chen, Haozhe Yang, Xu Liu, and Kai Xu. Design and kinematic modeling of a novel steerable needle for image-guided insertion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2400–2406. IEEE, 2020.
- [53] Wenqiang Chi, Jindong Liu, Mohamed EMK Abdelaziz, Giulio Dagnino, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Trajectory optimization of robot-assisted endovascular catheterization with reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3875–3881. IEEE, 2018.
- [54] Wenqiang Chi, Jindong Liu, Hedyeh Rafii-Tari, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Learning-based endovascular navigation through the use of non-rigid registration for collaborative robotic catheterization. *International journal of computer assisted radiology and surgery*, 13(6):855–864, 2018.
- [55] Wenqiang Chi, Hedyeh Rafii-Tari, Christopher J Payne, Jindong Liu, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. A learning based training and skill assessment platform with haptic guidance for endovascular catheterization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2357–2363. IEEE, 2017.
- [56] HCM Clogenson, J Dankelman, and JJ van den Dobbelsteen. Steerable guidewire for magnetic resonance guided endovascular interventions. *Journal of Medical Devices*, 8(2), 2014.
- [57] Noah J Cowan, Ken Goldberg, Gregory S Chirikjian, Gabor Fichtinger, Ron Alterovitz, Kyle B Reed, Vinutha Kallem, Wooram Park, Sarthak Misra, and Allison M Okamura. Robotic needle steering: Design, modeling, planning, and image guidance. In *Surgical robotics*, pages 557–582. Springer, 2011.

- [58] Whitney Crooks, Gabrielle Vukasin, Maeve O’Sullivan, William Messner, and Chris Rogers. Fin ray® effect inspired soft robotic gripper: From the robosoft grand challenge toward optimization. *Frontiers in Robotics and AI*, 3:70, 2016.
- [59] E De Momi, G Ferrigno, G Bosoni, P Bassanini, P Blasi, G Casaceli, D Fuschillo, L Castana, M Cossu, G Russo, and F Cardinale. A method for the assessment of time-varying brain shift during navigated epilepsy surgery. *International journal of computer assisted radiology and surgery*, 11, 07 2015.
- [60] Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of a. *Journal of the ACM (JACM)*, 32(3):505–536, 1985.
- [61] Sébastien Delorme, Denis Laroche, Robert DiRaddo, and Rolando F Del Maestro. Neurotouch: a physics-based virtual simulator for cranial microneurosurgery training. *Operative Neurosurgery*, 71(suppl_1):ons32–ons42, 2012.
- [62] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [63] Simon P DiMaio and Septimiu E Salcudean. Needle steering and motion planning in soft tissues. *IEEE Transactions on Biomedical Engineering*, 52(6):965–974, 2005.
- [64] Vincent Duindam, Ron Alterovitz, Shankar Sastry, and Ken Goldberg. Screw-based motion planning for bevel-tip flexible needles in 3d environments with obstacles. In *2008 IEEE international conference on robotics and automation*, pages 2483–2488. IEEE, 2008.
- [65] Vincent Duindam, Jijie Xu, Ron Alterovitz, Shankar Sastry, and Ken Goldberg. 3d motion planning algorithms for steerable needles using inverse kinematics. In *Algorithmic Foundation of Robotics VIII*, pages 535–549. Springer, 2009.
- [66] Pierre E Dupont, Jesse Lock, Brandon Itkowitz, and Evan Butler. Design and control of concentric-tube robots. *IEEE Transactions on Robotics*, 26(2):209–225, 2009.
- [67] Jan Egger, Zvonimir Mostarkic, Stefan Großkopf, and Bernd Freisleben. A fast vessel centerline extraction algorithm for catheter simulation. In *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS’07)*, pages 177–182. IEEE, 2007.
- [68] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Adding disjunction to datalog. In Victor Vianu, editor, *Proceedings of the Thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 24-26, 1994, Minneapolis, Minnesota, USA*, pages 267–278. ACM Press, 1994.
- [69] Johnathan A Engh, Davneet S Minhas, Douglas Kondziolka, and Cameron N Riviere. Percutaneous intracerebral navigation by duty-cycled spinning of flexible bevel-tipped needles. *Neurosurgery*, 67(4):1117–1123, 2010.
- [70] Johnathan A Engh, Gregg Podnar, Douglas Kondziolka, and Cameron N Riviere. Toward effective needle steering in brain tissue. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 559–562. IEEE, 2006.
- [71] Kenth Engø. On the bch-formula in so (3). *BIT Numerical Mathematics*, 41(3):629–632, 2001.
- [72] Esra Erdem, Erdi Aker, and Volkan Patoglu. Answer set programming for collaborative housekeeping robotics: representation, reasoning, and execution. *Intelligent Service Robotics*, 5(4):275–291, 2012.
- [73] Esra Erdem, Volkan Patoglu, and Zeynep G Saribatur. Integrating hybrid diagnostic reasoning in plan execution monitoring for cognitive factories with multiple robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2007–2013. IEEE, 2015.
- [74] David Erdos, Abraham Erdos, and Steve E Watkins. An experimental uav system for search and rescue challenge. *IEEE Aerospace and Electronic Systems Magazine*, 28(5):32–37, 2013.
- [75] Caroline Essert, Claire Haegelen, Florent Lalys, Alexandre Abadie, and Pierre Jannin. Automatic computation of electrode trajectories for Deep Brain Stimulation: A hybrid symbolic and numerical approach. *International Journal of Computer Assisted Radiology and Surgery*, 7(4):517–532, 2012.
- [76] Bita Fallahi, Carlos Rossa, Ron S Sloboda, Nawaid Usmani, and Mahdi Tavakoli. Sliding-based switching control for image-guided needle steering in soft tissue. *IEEE Robotics and Automation Letters*, 1(2):860–867, 2016.
- [77] Johannes Fauser, Romol Chadda, Yannik Goergen, Markus Hessinger, Paul Motzki, Igor Stenin, Julia Kristin, Thomas Klenzner, Jörg Schipper, Stefan Seelecke, et al. Planning for flexible surgical robots via bezier spline translation. *IEEE Robotics and Automation Letters*, 4(4):3270–3277, 2019.

- [78] Johannes Fauser, Georgios Sakas, and Anirban Mukhopadhyay. Planning nonlinear access paths for temporal bone surgery. *International journal of computer assisted radiology and surgery*, 13(5):637–646, 2018.
- [79] Alberto Favaro, Leonardo Cerri, Stefano Galvan, Ferdinando Rodriguez Y Baena, and Elena De Momi. Automatic optimized 3d path planner for steerable catheters with heuristic search and uncertainty tolerance. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–16. IEEE, 2018.
- [80] Alberto Favaro, Leonardo Cerri, Davide Scorza, and Elena De Momi. Automatic multi-trajectory planning solution for steerable catheters. In *2018 International Symposium on Medical Robotics (ISMR)*, pages 1–6. IEEE, 2018.
- [81] Alberto Favaro, Riccardo Secoli, Ferdinando Rodriguez y Baena, and Elena De Momi. Model-based robust pose estimation for a multi-segment, programmable bevel-tip steerable needle. *IEEE Robotics and Automation Letters*, 5(4):6780–6787, 2020.
- [82] Alberto Favaro, Alice Segato, Federico Muretti, and Elena De Momi. An evolutionary-optimized surgical path planner for a programmable bevel-tip needle. *IEEE Transactions on Robotics*, 2021.
- [83] B Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.
- [84] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.
- [85] A Forte, S Galvan, and D Dini. Models and tissue mimics for brain shift simulations. *Biomechanics and Modeling in Mechanobiology*, 17:1–13, 2017.
- [86] AE Forte, SM Gentleman, and D Dini. On the characterization of the heterogeneous mechanical response of human brain tissue. *Biomechanics and Modeling in Mechanobiology*, 16:907–920, 2017.
- [87] Mengyu Fu, Alan Kuntz, Robert J Webster, and Ron Alterovitz. Safe motion planning for steerable needles using cost maps automatically extracted from pulmonary images. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4942–4949. IEEE, 2018.
- [88] Mengyu Fu, Oren Salzman, and Ron Alterovitz. Toward certifiable motion planning for medical steerable needles. *arXiv preprint arXiv:2107.04939*, 2021.
- [89] KH Fuchs. Minimally invasive surgery. *Endoscopy*, 34(02):154–159, 2002.
- [90] Junli Gao, Weijie Ye, Jing Guo, and Zhongjuan Li. Deep reinforcement learning for indoor mobile robot path planning. *Sensors*, 20(19):5493, 2020.
- [91] Ming-ke Gao, Yi-min Chen, Quan Liu, Chen Huang, Ze-yu Li, and Dian-hua Zhang. Three-dimensional path planning and guidance of leg vascular based on improved ant colony algorithm in augmented reality. *Journal of medical systems*, 39(11):1–11, 2015.
- [92] Santiago Garrido, Luis Moreno, Mohamed Abderrahim, and Fernando Martin. Path planning for mobile robot navigation using voronoi diagram and fast marching. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2376–2381. IEEE, 2006.
- [93] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot asp solving with clingo. *Theory and Practice of Logic Programming*, 19(1):27–82, 2019.
- [94] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New generation computing*, 9(3-4):365–385, 1991.
- [95] Hunter B Gilbert, D Caleb Rucker, and Robert J Webster III. Concentric tube robots: The state of the art and future directions. *Robotics Research*, pages 253–269, 2016.
- [96] Cedric Girerd, Andrey V Kudryavtsev, Patrick Rougeot, Pierre Renaud, Kanty Rabenorosoa, and Brahim Tamadazte. Slam-based follow-the-leader deployment of concentric tube robots. *IEEE Robotics and Automation Letters*, 5(2):548–555, 2020.
- [97] Roy Glasius, Andrzej Komoda, and Stan CAM Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 8(1):125–133, 1995.
- [98] Daniel Glozman and Moshe Shoham. Flexible needle steering and optimal trajectory planning for percutaneous therapies. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 137–144. Springer, 2004.
- [99] Daniel Glozman and Moshe Shoham. Flexible needle steering for percutaneous therapies. *Computer Aided Surgery*, 11(4):194–201, 2006.
- [100] Daniel Glozman and Moshe Shoham. Image-guided robotic flexible needle steering. *IEEE Transactions on Robotics*, 23(3):459–467, 2007.

- [101] Will Goldstone. *Unity game development essentials*. Packt Publishing Ltd, 2009.
- [102] Rodrigo N Gómez, Carlos Hernández, and Jorge A Baier. A compact answer set programming encoding of multi-agent pathfinding. *IEEE Access*, 9:26886–26901, 2021.
- [103] D Grasso, C Borreggine, D Melchionda, C Bristogiannis, LP Stoppino, and L Macarini. Role of 3d mri with proset technique in the evaluation of lumbar radiculopathy. *J Biol Regul Homeost Agents*, 27(3):817–25, 2013.
- [104] Florent Guenter, Micha Hersch, Sylvain Calinon, and Aude Billard. Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics*, 21(13):1521–1544, 2007.
- [105] J Guo, S Hirsch, A Fehlner, S Papazoglou, M Scheel, J Braun, and I Sack. Towards an elastographic atlas of brain anatomy. *PLOS ONE*, 8(8):1–10, 2013.
- [106] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.
- [107] Giselle G Hamad and Myriam Curet. Minimally invasive surgery, 2010.
- [108] N Hamzé, A Bilger, C Duriez, S Cotin, and C Essert. Anticipation of brain shift in deep brain stimulation automatic planning. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3635–3638, 2015.
- [109] N Hamzé, I Peterlík, S Cotin, and C Essert. Preoperative trajectory planning for percutaneous procedures in deformable environments. *Computerized Medical Imaging and Graphics*, 47:16 – 28, 2016.
- [110] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [111] T Hartkens, D Hill, AD Castellano-Smith, DJ Hawkes, M Jr, AJ Martin, WA Hall, H Liu, and CL Truwit. Measurement and analysis of brain deformation during neurosurgery. *IEEE transactions on medical imaging*, 22:82–92, 02 2003.
- [112] P Hastreiter, C Rezk-Salama, G Soza, M Bauer, G Greiner, R Fahlbusch, O Ganslandt, and C Nimsky. Strategies for brain shift evaluation. *Medical Image Analysis*, 8(4):447 – 464, 2004.
- [113] Haralampos Hatzikirou, Andreas Deutsch, Carlo Schaller, Matthias Simon, and Kristin Swanson. Mathematical modelling of glioblastoma tumour development: a review. *Mathematical Models and Methods in Applied Sciences*, 15(11):1779–1794, 2005.
- [114] DL Hill, CR Jr Maurer, RJ Maciunas, JA Barwise, JM Fitzpatrick, and Wang MY. Measurement of intraoperative brain surface deformation under a craniotomy. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI'98*, pages 51–62. Springer Berlin Heidelberg, 1998.
- [115] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.
- [116] Janine Hoelscher, Mengyu Fu, Inbar Fried, Maxwell Emerson, Tayfun Efe Ertop, Margaret Rox, Alan Kuntz, Jason A Akulian, Robert J Webster III, and Ron Alterovitz. Backward planning for a multi-stage steerable needle lung robot. *IEEE Robotics and Automation Letters*, 6(2):3987–3994, 2021.
- [117] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.
- [118] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [119] Ayoung Hong, Quentin Boehler, Roman Moser, Ajmal Zemmar, Lennart Stieglitz, and Bradley J Nelson. 3d path planning for flexible needle steering in neurosurgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 15(4):e1998, 2019.
- [120] Wenguo Hou, Peter X Liu, Minhua Zheng, and Shichao Liu. A new deformation model of brain tissues for neurosurgical simulation. *IEEE Transactions on Instrumentation and Measurement*, 69(4):1251–1258, 2019.
- [121] Michael Hoy, Alexey S Matveev, and Andrey V Savkin. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(3):463–497, 2015.
- [122] J Hu, X Jin, JB Lee, L Zhang, V Chaudhary, M Guthikonda, KH Yang, and AI King. Intraoperative brain shift prediction using a 3d inhomogeneous patient-specific finite element model. *Journal of Neurosurgery*, 106(1):164–169, 2007.
- [123] Yang Hu, Lin Zhang, Wei Li, and Guang-Zhong Yang. Robotic sewing and knot tying for personalized stent graft manufacturing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 754–760. IEEE, 2018.

- [124] Haomiao Huang, Gabriel M Hoffmann, Steven L Waslander, and Claire J Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *2009 IEEE international conference on robotics and automation*, pages 3277–3282. IEEE, 2009.
- [125] Yoshinori Inoue and Koji Ikuta. Hydraulic driven active catheters with optical bending sensor. In *2016 IEEE 29th international conference on micro electro mechanical systems (MEMS)*, pages 383–386. IEEE, 2016.
- [126] C Johnson, J Holtrop, M McGarry, J Weaver, K Paulsen, J Georgiadis, and B Sutton. 3d multislabs, multishot acquisition for fast, whole-brain mr elastography with high signal-to-noise efficiency. *Magnetic resonance in medicine*, 71, 2014.
- [127] CL Johnson, S Hillary, M D.J. McGarry, AT Anderson, GR Huesmann, BP Sutton, and NJ Cohen. Viscoelasticity of subcortical gray matter structures. *Human Brain Mapping*, 37(12):4221–4233, 2016.
- [128] Matthew Jordan and Alejandro Perez. Optimal bidirectional rapidly-exploring random trees. In *Computer Science and Artificial Intelligence Laboratory Technical Report*, 2013.
- [129] Leo Joskowicz. Advances in image-guided targeting for keyhole neurosurgery. *Touch Briefings Reports, Future Directions in surgery London, UK*, 2007:1–4, 2006.
- [130] Leo Joskowicz. Advances in image-guided targeting for keyhole neurosurgery: a survey paper. *Touch Briefings reports, Future directions in surgery*, 2, 2006.
- [131] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.
- [132] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *2011 IEEE international conference on robotics and automation*, pages 4569–4574. IEEE, 2011.
- [133] Vinutha Kallem and Noah J Cowan. Image guidance of flexible tip-steerable needles. *IEEE Transactions on Robotics*, 25(1):191–196, 2009.
- [134] Vinutha Kallem, Adam T Komoroski, and Vijay Kumar. Sequential composition for navigating a nonholonomic cart in the presence of obstacles. *IEEE Transactions on Robotics*, 27(6):1152–1159, 2011.
- [135] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. An affective guide robot in a shopping mall. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 173–180, 2009.
- [136] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 6 2011.
- [137] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.
- [138] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt. In *2011 IEEE International Conference on Robotics and Automation*, pages 1478–1483. IEEE, 2011.
- [139] Nikos I Katevas, Spyros G Tzafestas, and Christos G Pneumatikatos. The approximate cell decomposition with local node refinement global path planning method: Path nodes refinement and curve parametric interpolation. *Journal of Intelligent and Robotic Systems*, 22(3):289–314, 1998.
- [140] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and automation*, 14(1):166–171, 1998.
- [141] Lydia E. Kavraki and Steven M. LaValle. Motion planning. In *Springer Handbook of Robotics*. 2016.
- [142] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [143] Fouzia Khan, Abdulhamit Donder, Stefano Galvan, Ferdinando Rodriguez y Baena, and Sarthak Misra. Pose measurement of flexible medical instruments using fiber bragg gratings in multi-core fiber. *IEEE sensors journal*, 20(18):10955–10962, 2020.
- [144] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.
- [145] Seong Young Ko, Brian L Davies, and Ferdinando Rodriguez y Baena. Two-dimensional needle steering with a “programmable bevel” inspired by nature: Modeling preliminaries. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2319–2324. IEEE, 2010.

- [146] Seong Young Ko and Ferdinando Rodriguez y Baena. Toward a miniaturized needle steering system with path planning for obstacle avoidance. *IEEE Transactions on Biomedical Engineering*, 60(4):910–917, 2012.
- [147] Seong Young Ko and Ferdinando Rodriguez y Baena. Trajectory following for a flexible probe with state/input constraints: An approach based on model predictive control. *Robotics and Autonomous systems*, 60(4):509–521, 2012.
- [148] Sven Koenig and Maxim Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, volume 1, pages 968–975. IEEE, 2002.
- [149] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.
- [150] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, pages 3232–3237. IEEE, 2010.
- [151] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.
- [152] Alan Kuntz, Luis G Torres, Richard H Feins, Robert J Webster, and Ron Alterovitz. Motion planning for a three-stage multilumen transoral lung access system. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3255–3261. IEEE, 2015.
- [153] Christian Kunz, Maximilian Gerst, Pit Henrich, Max Schneider, Michal Hlavac, Andrej Pala, and Franziska Mathis-Ullrich. Multimodal risk-based path planning for neurosurgical interventions. *Journal of Medical Devices*, 2021.
- [154] Guillaume Lapouge, Philippe Poignet, and Jocelyne Troccaz. Towards 3d ultrasound guided needle steering robust to uncertainties, noise and tissue heterogeneity. *IEEE Transactions on Biomedical Engineering*, 2020.
- [155] Boris Lau, Christoph Sprunk, and Wolfram Burgard. Kinodynamic motion planning for mobile robots using splines. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2427–2433. IEEE, 2009.
- [156] S M LaValle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *In*, 1998.
- [157] Steven M LaValle and James J Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. Citeseer, 2000.
- [158] Chi-Tai Lee and Ching-Chih Tsai. 3d collision-free trajectory generation using elastic band technique for an autonomous helicopter. In *FIRA RoboWorld Congress*, pages 34–41. Springer, 2011.
- [159] Yonggu Lee, Xiaoyu Tan, Chin-Boon Chng, and Chee-Kong Chui. Simulation of robot-assisted flexible needle insertion using deep q-network. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 342–346. IEEE, 2019.
- [160] A Leibinger, AE Forte, Z Tan, MJ Oldfield, F Beyrau, D Dini, and F Rodriguez Y Baena. Soft tissue phantoms for realistic needle insertion: A comparative study. *Annals of biomedical engineering*, 44(8):2442–2452, 2016.
- [161] A Leibinger, MJ Oldfield, and F Rodriguez Y Baena. Minimally disruptive needle insertion: a biologically inspired solution. *Interface Focus*, 6(3), 2016.
- [162] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The dlV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic (TOCL)*, 7(3):499–562, 2006.
- [163] Nicola Leone and Francesco Ricca. Answer set programming: A tour from the basics to advanced development tools and industrial applications. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 308–326. Springer, 2015.
- [164] Jinxing Li, Berta Esteban-Fernández de Ávila, Wei Gao, Liangfang Zhang, and Joseph Wang. Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification. *Science Robotics*, 2(4), 2017.
- [165] Murong Li, Dedong Gao, Yong Lei, and Tian Xu. Dynamic path planning for bevel-tip flexible needle insertion into soft tissue based on a real-time finite element model. *Mathematical Problems in Engineering*, 2020, 2020.

- [166] Pan Li, Shan Jiang, Dong Liang, Zhiyong Yang, Yan Yu, and Wei Wang. Modeling of path planning and needle steering with path tracking in anatomical soft tissues for minimally invasive surgery. *Medical engineering & physics*, 41:35–45, 2017.
- [167] Pan Li, Shan Jiang, Jun Yang, and Zhiyong Yang. A combination method of artificial potential field and improved conjugate gradient for trajectory planning for needle insertion into soft tissue. *J Med Biol Eng*, 34(6):568–573, 2014.
- [168] Zhen Li, Jenny Dankelman, and Elena De Momi. Path planning for endovascular catheterization under curvature constraints via two-phase searching approach. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–9, 2021.
- [169] Vladimir Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138(1-2):39–54, 2002.
- [170] Vladimir Lifschitz. *Answer set programming*. Springer, 2019.
- [171] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [172] Fangde Liu, Arnau Garriga-Casanovas, Riccardo Secoli, and Ferdinando Rodriguez y Baena. Fast and adaptive fractal tree-based path planning for programmable bevel tip steerable needles. *IEEE Robotics and Automation Letters*, 1(2):601–608, 2016.
- [173] Fangde Liu, Joshua Petersen, and Ferdinando Rodriguez y Baena. Parallel moduli space sampling: Robust and fast surgery planning for image guided steerable needles. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 626–631. IEEE, 2015.
- [174] Edgar Lobaton, Jinghe Zhang, Sachin Patil, and Ron Alterovitz. Planning curvature-constrained paths to multiple goals using circle sampling. In *2011 IEEE International Conference on Robotics and Automation*, pages 1463–1469. IEEE, 2011.
- [175] Miles Macklin, Matthias Müller, Nuttapon Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.
- [176] Eloise Matheson and Ferdinando Rodriguez y Baena. Biologically inspired surgical needle steering: Technology and application of the programmable bevel-tip needle. *Biomimetics*, 5(4):68, 2020.
- [177] Eloise Matheson, Riccardo Secoli, Christopher Burrows, Alexander Leibinger, and Ferdinando Rodriguez y Baena. Cyclic motion control for programmable bevel-tip needles to reduce tissue deformation. *Journal of Medical Robotics Research*, 4(01):1842001, 2019.
- [178] Ryszard S Michalski. A theory and methodology of inductive learning. In *Machine learning*, pages 83–134. Elsevier, 1983.
- [179] A Midawi, N Huda, Y Kisaka, R Lazor, and A Gerlich. Measurement of local mechanical properties in pipeline steel welds using instrumented indentation and digital image correlation. In *CWA Journal 2015 Special Conference Edition*, 2015.
- [180] Davneet S Minhas, Johnathan A Engh, Michele M Fenske, and Cameron N Riviere. Modeling of needle steering via duty-cycled spinning. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2756–2759. IEEE, 2007.
- [181] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems*, pages 2419–2430, 2018.
- [182] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.
- [183] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [184] Pedro Moreira and Sarthak Misra. Biomechanics-based curvature estimation for ultrasound-guided flexible needle steering in biological tissues. *Annals of biomedical engineering*, 43(8):1716–1726, 2015.
- [185] Mohammad Motaharifar, Heidar A Talebi, Farzaneh Abdollahi, and Ahmad Afshar. Nonlinear adaptive output-feedback controller design for guidance of flexible needles. *IEEE/ASME Transactions on Mechatronics*, 20(4):1912–1919, 2014.

- [186] M Müller, B Heidelberger, M Hennix, and J Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, 2007.
- [187] M Müller, B Heidelberger, M Teschner, and M Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, 2005.
- [188] M Müller, B Heidelberger, M Teschner, and M Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, 2005.
- [189] M Murphy, J Huston, C Jack, K Glaser, M Senjem, J Chen, A Manduca, J Felmlee, and R Ehman. Measuring the characteristic topography of brain stiffness with magnetic resonance elastography. *PLoS one*, 8:e81668, 2013.
- [190] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Policy gradient with value function approximation for collective multiagent planning. In *Advances in Neural Information Processing Systems*, pages 4319–4329, 2017.
- [191] Sherdil Niyaz, Alan Kuntz, Oren Salzman, Ron Alterovitz, and Siddhartha Srinivasa. Following surgical trajectories with concentric tube robots via nearest-neighbor graphs. In *International Symposium on Experimental Robotics*, pages 3–13. Springer, 2018.
- [192] Marek Obitko. Genetic algorithms. *Internet publication*, 1998.
- [193] David O’Brien, Susan Fisher, and Ming C Lin. Automatic simplification of particle system dynamics. In *Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No. 01TH8596)*, pages 210–257. IEEE, 2001.
- [194] Stephen Okazawa, Richelle Ebrahimi, Jason Chuang, Septimiu E Salcudean, and Robert Rohling. Hand-held steerable needle device. *IEEE/ASME Transactions on Mechatronics*, 10(3):285–296, 2005.
- [195] Andrew Palisch, Raj G Patel, Christina Gutowski, Adam C Zoga, Phillip Colucci, Brian J O’Hara, Catherine C Roberts, and John Abraham. Analysis of needle type for musculoskeletal lesion biopsy: results of a novel steerable needle. *Current Orthopaedic Practice*, 27(4):393–399, 2016.
- [196] Aleksandr I Panov, Konstantin S Yakovlev, and Roman Suvorov. Grid path planning with deep reinforcement learning: Preliminary results. *Procedia computer science*, 123:347–353, 2018.
- [197] Chonhyon Park, Jia Pan, and Dinesh Manocha. Itomp: Incremental trajectory optimization for real-time replanning in dynamic environments. In *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.
- [198] Wooram Park, Jin Seob Kim, Yu Zhou, Noah J Cowan, Allison M Okamura, and Gregory S Chirikjian. Diffusion-based motion planning for a nonholonomic flexible needle model. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4600–4605. IEEE, 2005.
- [199] Wooram Park, Yunfeng Wang, and Gregory S Chirikjian. The path-of-probability algorithm for steering and feedback control of flexible needles. *The International journal of robotics research*, 29(7):813–830, 2010.
- [200] Sachin Patil and Ron Alterovitz. Interactive motion planning for steerable needles in 3d environments with obstacles. In *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics*, pages 893–899. IEEE, 2010.
- [201] Sachin Patil, Jessica Burgner, Robert J Webster, and Ron Alterovitz. Needle steering in 3-d via rapid replanning. *IEEE Transactions on Robotics*, 30(4):853–864, 2014.
- [202] Sachin Patil et al. Motion planning under uncertainty in highly deformable environments. *Robotics science and systems: online proceedings*, 2011.
- [203] Xavier Pennec, Pascal Cachier, and Nicholas Ayache. Understanding the “demon’s algorithm”: 3d non-rigid registration by gradient descent. In Chris Taylor and Alain Colchester, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI’99*, pages 597–605, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [204] Marina Pérez-Aliacar, Mohamed H Doweidar, Manuel Doblaré, and Jacobo Ayensa-Jiménez. Predicting cell behaviour parameters from glioblastoma on a chip images. a deep learning approach. *Computers in Biology and Medicine*, page 104547, 2021.
- [205] Andrew J Petruska, Fabio Ruetz, Ayoung Hong, Luca Regli, Oguzkan Sürücü, Ajmal Zemmar, and Bradley J Nelson. Magnetic needle guidance for neurosurgery: Initial design and proof of concept. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4392–4397. IEEE, 2016.
- [206] S Pieper, M Halle, and R Kikinis. 3d slicer. *Proceedings of the IEEE international symposium on biomedical imaging*, page 632–635, 2004.

- [207] Steve Pieper, Michael Halle, and Ron Kikinis. 3d slicer. In *2004 2nd IEEE international symposium on biomedical imaging: nano to macro (IEEE Cat No. 04EX821)*, pages 632–635. IEEE, 2004.
- [208] PK Pillay, SJ Hassenbusch, and R Sawaya. Minimally invasive brain surgery. *Annals of the Academy of Medicine, Singapore*, 22(3 Suppl):459–463, 1993.
- [209] Marlene Pinzi, Stefano Galvan, Thomas Watts, Riccardo Secoli, and Ferdinando Rodriguez y Baena. Path replanning for orientation-constrained needle steering. *IEEE Transactions on Biomedical Engineering*, 2021.
- [210] Marlene Pinzi, Stefano Galvan, and Ferdinando Rodriguez y Baena. The adaptive hermite fractal tree (ahft): a novel surgical 3d path planning approach with curvature and heading constraints. *International journal of computer assisted radiology and surgery*, 14(4):659–670, 2019.
- [211] Marlene Pinzi, Brian Hwang, Vejay N Vakharia, John S Duncan, Ferdinando Rodriguez y Baena, and William S Anderson. Computer assisted planning for curved laser interstitial thermal therapy. *IEEE Transactions on Biomedical Engineering*, 2021.
- [212] Julian J Portillo, Carmen L Garcia-Mata, Pedro R Márquez-Gutiérrez, and Rogelio Baray-Arana. Robot platform motion planning using answer set programming. In *LA-NMR*, pages 35–44, 2011.
- [213] George B Purdy and Justin W Smith. Lines, circles, planes and spheres. *Discrete & Computational Geometry*, 44(4):860–882, 2010.
- [214] Zhihua Qu, Jing Wang, and Clinton E Plaisted. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, 20(6):978–993, 2004.
- [215] Hedyeh Rafii-Tari, Jindong Liu, Su-Lin Lee, Colin Bicknell, and Guang-Zhong Yang. Learning-based modeling of endovascular navigation for collaborative robotic catheterization. In *International conference on medical image computing and computer-assisted intervention*, pages 369–377. Springer, 2013.
- [216] Hedyeh Rafii-Tari, Christopher J Payne, Jindong Liu, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Towards automated surgical skill evaluation of endovascular catheterization tasks based on force and motion signatures. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1789–1794. IEEE, 2015.
- [217] Purushothaman Raja and Sivagurunathan Pugazhenth. Optimal path planning of mobile robots: A review. *International journal of physical sciences*, 7(9):1314–1320, 2012.
- [218] Babu Ram. *Engineering mathematics*. Pearson Education India, 2009.
- [219] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494. IEEE, 2009.
- [220] Kyle B Reed, Vinutha Kallem, Ron Alterovitz, Ken Goldberg, Allison M Okamura, and Noah J Cowan. Integrated planning and image-guided control for planar needle steering. In *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics*, pages 819–824. IEEE, 2008.
- [221] Kyle B Reed, Ann Majewicz, Vinutha Kallem, Ron Alterovitz, Ken Goldberg, Noah J Cowan, and Allison M Okamura. Robot-assisted needle steering. *IEEE robotics & automation magazine*, 18(4):35–46, 2011.
- [222] Ambrose Robert, Kumar Vijay, Wilcox Brian, Yuh Jun-ku, Lavery David, et al. *Robotics: state of the art and future challenges*. World Scientific, 2008.
- [223] DW Roberts, A Hartov, FE Kennedy, MI Miga, and KD Paulsen. Intraoperative brain shift and deformation: A quantitative analysis of cortical displacement in 28 cases. *Neurosurgery*, 43(4):749–758, 2000.
- [224] John D Rolston, Corinna C Zygourakis, Seunggu J Han, Catherine Y Lau, Mitchel S Berger, and Andrew T Parsa. Medical errors in neurosurgery. *Surgical neurology international*, 5(Suppl 10):S435, 2014.
- [225] Eric Rombokas, Evangelos Theodorou, Mark Malhotra, Emo Todorov, and Yoky Matsuoka. Tendon-driven control of biomechanical and robotic systems: A path integral reinforcement learning approach. In *2012 IEEE International Conference on Robotics and Automation*, pages 208–214. IEEE, 2012.
- [226] D Caleb Rucker, Jadav Das, Hunter B Gilbert, Philip J Swaney, Michael I Miga, Nilanjan Sarkar, and Robert J Webster. Sliding mode control of steerable needles. *IEEE Transactions on Robotics*, 29(5):1289–1299, 2013.
- [227] Daniel Caleb Rucker. *The mechanics of continuum robots: model-based sensing and control*. Citeseer, 2011.
- [228] Wael Sabra, Matthew Khouzam, Arnaud Chanu, and Sylvain Martel. Use of 3d potential field and an enhanced breadth-first search algorithms for the path planning of microdevices propelled in the cardiovascular system. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3916–3920. IEEE, 2006.

- [229] N Sariff and Norlida Buniyamin. An overview of autonomous mobile robot path planning algorithms. In *2006 4th student conference on research and development*, pages 183–188. IEEE, 2006.
- [230] Sebastian Schafer, Vikas Singh, Kenneth R Hoffmann, Peter B Noël, and Jinhui Xu. Planning image-guided endovascular interventions: guidewire simulation using shortest path algorithms. In *Medical Imaging 2007: Visualization and Image-Guided Procedures*, volume 6509, page 65092C. International Society for Optics and Photonics, 2007.
- [231] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.
- [232] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.
- [233] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [234] Riccardo Secoli, Eloise W. Matheson, Marlene Pinzi, Stefano Galvan, Thomas E. Watts, Abdulhamit Donder, and Ferdinando Rodriguez y Baena. A modular robotic platform for precision neurosurgery with a programmable bevel-tip needle. *IEEE Transactions on Robotics*, Under review.
- [235] Riccardo Secoli and Ferdinando Rodriguez y Baena. Closed-loop 3d motion modeling and control of a steerable needle for soft tissue surgery. In *2013 IEEE International Conference on Robotics and Automation*, pages 5831–5836. IEEE, 2013.
- [236] Riccardo Secoli and Ferdinando Rodriguez y Baena. Adaptive path-following control for bio-inspired steerable needles. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechanics (BioRob)*, pages 87–93. IEEE, 2016.
- [237] Alice Segato, Valentina Corbetta, Calimeri Francesco, and Elena De Momi. Inductive and deductive reasoning for robotic steerable needle in neurosurgery. In *2020-International Conference on Intelligent Robots and Systems*. USA, 2020.
- [238] Alice Segato, Chiara Di Vece, Sara Zucchelli, Marco Di Marzo, Thomas Wendler, Mohammad Farid Azampour, Stefano Galvan, Riccardo Secoli, and Elena De Momi. Position-based dynamics simulator of brain deformations for path planning and intra-operative control in keyhole neurosurgery. *IEEE Robotics and Automation Letters*, 2021.
- [239] Alice Segato, Testa Irene, and Elena De Momi. 3d neurosurgical simulator for training robotic steerable catheter agents using generative adversarial imitation learning. In *2020 International Conference on Robotics and Automation*. FRA, 2020.
- [240] Alice Segato, Valentina Pieri, Alberto Favaro, Marco Riva, Andrea Falini, Elena De Momi, and Antonella Castellano. Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei. *Frontiers in Robotics and AI*, 6:70, 2019.
- [241] Alice Segato, Luca Sestini, Antonella Castellano, Baena, and Elena De Momi. Ga3c reinforcement learning for surgical steerable catheter path planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
- [242] Alice Segato, Luca Sestini, Antonella Castellano, and Elena De Momi. Ga3c reinforcement learning for surgical steerable catheter path planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2429–2435. IEEE, 2020.
- [243] Alice Segato, Corbetta Valentina, Jessica Zangari, Simona Perri, Francesco Calimeri, and Elena De Momi. Optimized 3d path planner for steerable catheters with deductive reasoning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 0–7. IEEE, 2021.
- [244] Chaoyang Shi, Xiongbiao Luo, Peng Qi, Tianliang Li, Shuang Song, Zoran Najdovski, Toshio Fukuda, and Hongliang Ren. Shape sensing techniques for continuum robots in minimally invasive surgery: A survey. *IEEE Transactions on Biomedical Engineering*, 64(8):1665–1678, 2016.
- [245] David K. Smith. Dynamic Programming and Optimal Control. Volume 1. *Journal of the Operational Research Society*, 1996.
- [246] Moshe Sniedovich. Dijkstra’s algorithm revisited: the dynamic programming connexion. *Control and cybernetics*, 35:599–620, 2006.
- [247] Richard F Spaide. Steerable and flexibly curved probes, November 27 2012. US Patent 8,317,778.

- [248] Freek Stulp, Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. Learning to grasp under uncertainty. In *2011 IEEE International Conference on Robotics and Automation*, pages 5703–5708. IEEE, 2011.
- [249] Priyanka Sudhakara, Velappa Ganapathy, and Karthika Sundaran. Trajectory planning using enhanced probabilistic roadmaps for pliable needle robotic surgery. In *2018 International Conference on Recent Trends in Electrical, Control and Communication (RTECC)*, pages 61–64. IEEE, 2018.
- [250] Wen Sun, Sachin Patil, and Ron Alterovitz. High-frequency replanning under uncertainty using parallel sampling-based motion planning. *IEEE Transactions on Robotics*, 31(1):104–116, 2015.
- [251] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [252] E Tagliabue, D Dall’Alba, E Magnabosco, C Tenga, I Peterlik, and P Fiorini. Position-based modeling of lesion displacement in ultrasound-guided breast biopsy. *Int J Comput Assist Radiol Surg.*, 14(8):1329–1339, 2019.
- [253] Lei Tai, Giuseppe Paolo, and Ming Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36. IEEE, 2017.
- [254] Xiaoyu Tan, Yonggu Lee, Chin-Boon Chng, Kah-Bin Lim, and Chee-Kong Chui. Robot-assisted flexible needle insertion using universal distributional deep reinforcement learning. *International journal of computer assisted radiology and surgery*, 15(2):341–349, 2020.
- [255] Xiaoyu Tan, Pengqian Yu, Kah-Bin Lim, and Chee-Kong Chui. Robust path planning for flexible needle insertion using markov decision processes. *International journal of computer assisted radiology and surgery*, 13(9):1439–1451, 2018.
- [256] Zhen Tan, Hua-Geng Liang, Dan Zhang, and Qing-Guo Wang. Path planning of surgical needle: A new adaptive intelligent particle swarm optimization method. *Transactions of the Institute of Measurement and Control*, page 0142331221998832, 2021.
- [257] Libo Tang, Yonghua Chen, and Xuejian He. Magnetic force aided compliant needle navigation and needle performance analysis. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 612–616. IEEE, 2007.
- [258] Voot Tangkaratt, Bo Han, Mohammad Emtiyaz Khan, and Masashi Sugiyama. Variational imitation learning with diverse-quality demonstrations. In *International Conference on Machine Learning*, pages 9407–9417. PMLR, 2020.
- [259] Martin Tomko and Stephan Winter. Beyond digital twins—a commentary. *Environment and Planning B: Urban Analytics and City Science*, 46(2):395–399, 2019.
- [260] Meysam Torabi, Kris Hauser, Ron Alterovitz, Vincent Duijndam, and Ken Goldberg. Guiding medical needles using single-point tissue manipulation. In *2009 IEEE International Conference on Robotics and Automation*, pages 2705–2710. IEEE, 2009.
- [261] Benjamn Tovar, Rafael Murrieta-Cid, and Steven M LaValle. Distance-optimal navigation in an unknown environment without sensing distances. *IEEE Transactions on Robotics*, 23(3):506–518, 2007.
- [262] Nick J van de Berg, Jenny Dankelman, and John J van den Dobbela. Design of an actively controlled steerable needle with tendon actuation and fbg-based shape sensing. *Medical engineering & physics*, 37(6):617–622, 2015.
- [263] Nick J van de Berg, Dennis J van Gerwen, Jenny Dankelman, and John J van den Dobbela. Design choices in needle steering—a review. *IEEE/ASME Transactions on Mechatronics*, 20(5):2172–2183, 2014.
- [264] Jur Van Den Berg, Stephen Miller, Daniel Duckworth, Humphrey Hu, Andrew Wan, Xiao-Yu Fu, Ken Goldberg, and Pieter Abbeel. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *2010 IEEE International Conference on Robotics and Automation*, pages 2074–2081. IEEE, 2010.
- [265] Jur Van Den Berg, Sachin Patil, Ron Alterovitz, Pieter Abbeel, and Ken Goldberg. Lqg-based planning, sensing, and control of steerable needles. In *Algorithmic Foundations of Robotics IX*, pages 373–389. Springer, 2010.
- [266] Laurence Vancamberg, Anis Sahbani, Serge Muller, and Guillaume Morel. Needle path planning for digital breast tomosynthesis biopsy. In *2010 IEEE International Conference on Robotics and Automation*, pages 2062–2067. IEEE, 2010.
- [267] Gustaaf J Vrooijink, Momen Abayazid, Sachin Patil, Ron Alterovitz, and Sarthak Misra. Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images. *The International journal of robotics research*, 33(10):1361–1374, 2014.

- [268] M Wagshul, P Eide, and J Madsen. The pulsating brain: A review of experimental and clinical studies of intracranial pulsatility. *Fluids and barriers of the CNS*, 8:5, 01 2011.
- [269] Jianjun Wang, Xiangpeng Li, Jinjin Zheng, and Dong Sun. Dynamic path planning for inserting a steerable needle into a soft tissue. *IEEE/ASME Transactions on Mechatronics*, 19(2):549–558, 2013.
- [270] Jianjun Wang, Dong Sun, Jinjin Zheng, and Wen Shang. Optimal path planning for inserting a steerable needle into tissue. In *2011 IEEE International Conference on Information and Automation*, pages 39–44. IEEE, 2011.
- [271] Junchen Wang, Takashi Ohya, Hongen Liao, Ichiro Sakuma, Tianmiao Wang, Iwai Tohnai, and Toshinori Iwai. Intravascular catheter navigation using path planning and virtual visual feedback for oral cancer treatment. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 7(2):214–224, 2011.
- [272] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.
- [273] Ajeet Wankhede, Likhita Madiraju, Dipam Patel, Kevin Cleary, Chima Oluigbo, and Reza Monfaredi. Heuristic-based optimal path planning for neurosurgical tumor ablation. In *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10951, page 109512H. International Society for Optics and Photonics, 2019.
- [274] Thomas Watts, Riccardo Secoli, and F Rodriguez y Baena. Modelling the deformation of biologically inspired flexible structures for needle steering. In *Mechanism, Machine, Robotics and Mechatronics Sciences*, pages 67–80. Springer, 2019.
- [275] Thomas Watts, Riccardo Secoli, and Ferdinando Rodriguez y Baena. A mechanics-based model for 3-d steering of programmable bevel-tip needles. *IEEE Transactions on Robotics*, 35(2):371–386, 2018.
- [276] Thomas Watts, Riccardo Secoli, and Ferdinando Rodriguez y Baena. Needle steerability measures: Definition and application for optimized steering of the programmable bevel-tip needle. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 59–64. IEEE, 2018.
- [277] Robert J Webster, Jasenka Memisevic, and Allison M Okamura. Design considerations for robotic needle steering. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3588–3594. IEEE, 2005.
- [278] Robert J Webster III, Jin Seob Kim, Noah J Cowan, Gregory S Chirikjian, and Allison M Okamura. Non-holonomic modeling of needle steering. *The International Journal of Robotics Research*, 25(5-6):509–525, 2006.
- [279] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.
- [280] Markus Wulfmeier, Dominic Zeng Wang, and Ingmar Posner. Watch this: Scalable cost-function learning for path planning in urban environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095. IEEE, 2016.
- [281] Jing Xiong, Zeyang Xia, and Yangzhou Gan. Human-robot collaborated path planning for bevel-tip needle steering in simulated human environment. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5672–5675. IEEE, 2016.
- [282] Jijie Xu, Vincent Duintam, Ron Alterovitz, and Ken Goldberg. Motion planning for steerable needles in 3d environments with obstacles using rapidly-exploring random trees and backchaining. In *2008 IEEE international conference on automation science and engineering*, pages 41–46. IEEE, 2008.
- [283] Jijie Xu, Vincent Duintam, Ron Alterovitz, Jean Pouliot, J Adam M Cunha, I-Chow Hsu, and Ken Goldberg. Planning fireworks trajectories for steerable medical needles to reduce patient trauma. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4517–4522. IEEE, 2009.
- [284] Liang Yang, Juntong Qi, Dalei Song, Jizhong Xiao, Jianda Han, and Yong Xia. Survey of Robot 3D Path Planning Algorithms, 2016.
- [285] Z Yin, AJ Romano, A Manduca, RL Ehman, and J 3rd. Huston. Stiffness and beyond: What mr elastography can tell us about brain structure and function under physiologic and pathologic conditions. *Top Magn Reson Imaging*, 27(5):305–318, 2018.
- [286] Hyeonseok You, EunKyung Bae, Youngjin Moon, Jihoon Kweon, and Jaesoon Choi. Automatic control of cardiac ablation catheter with deep reinforcement learning method. *Journal of Mechanical Science and Technology*, 33(11):5415–5423, 2019.

- [287] J Zhang, Y Zhong, and C Gu. Deformable models for surgical simulation: A survey. *IEEE Reviews in Biomedical Engineering*, 11:143–164, 2018.
- [288] Mingjun Zhang, Tzyh-Jong Tarn, and Ning Xi. Micro/nano-devices for controlled drug delivery. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pages 2068–2073. IEEE, 2004.
- [289] Yan-Jiang Zhao, Bardia Konh, Mohammad Honarvar, Felix Orlando Maria Joseph, Tarun K Podder, Parsaoran Hutapea, Adam P Dicker, and Yan Yu. 3d motion planning for robot-assisted active flexible needle based on rapidly-exploring random trees. *J. Autom. Control Eng*, 3(5), 2015.
- [290] Yan-Jiang Zhao, Wen-Qiang Wu, Yong-De Zhang, Rui-Xue Wang, Jing-Chun Peng, and Yan Yu. 3d dynamic motion planning for robot-assisted cannula flexible needle insertion into soft tissue. *International Journal of Advanced Robotic Systems*, 13(3):121, 2016.
- [291] Yue Zhao, Hongyu Qi, Dandan Li, Jing Jin, and Qiucheng Wang. Biopsy needle puncture path planning method based on 3d ultrasound images. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pages 1169–1174. IEEE, 2018.
- [292] Houguang Zhou, Yu Zhang, Ling Liu, Yanyan Huang, Yuping Tang, Jingjing Su, Wei Hua, Xu Han, Jianzhong Xue, and Qiang Dong. Minimally invasive stereotactic puncture and thrombolysis therapy improves long-term outcome after acute intracerebral hemorrhage. *Journal of neurology*, 258(4):661–669, 2011.
- [293] Jie-Hua Zhou, Ji-Qiang Zhou, Yong-Sheng Zheng, and Bin Kong. Research on path planning algorithm of intelligent mowing robot used in large airport lawn. In *2016 international conference on information system and artificial intelligence (ISAI)*, pages 375–379. IEEE, 2016.