Towards Autonomous Endovascular Surgery: Development of Assistance Tools for Computer-Aided Interventions

Valentina Scarponi

Université de Strasbourg

École Doctorale MSII 269
Mathématiques, Sciences de l'Information et de l'Ingénieur

MIMESIS Team, INRIA
Laboratoire des sciences de l'Ingénieur, de l'Informatique et de l'Imagerie
ICube UMR 7357

ITI HealthTech Ph.D. Program

Towards Autonomous Endovascular Surgery: Development of Assistance Tools for Computer-Aided Interventions

Defended by
Valentina Scarponi

9 December 2024

This dissertation is submitted for the degree of
Doctor of Philosophy

Supervised by:

Pr. Stéphane Cotin — Inria Strasbourg, University of Strasbourg
Pr. Florent Nageotte — University of Strasbourg
Pr. Michel Duprez — Inria Strasbourg, University of Strasbourg

Committee Members:

Pr. Franziska Mathis-Ullrich — University of Erlangen–Nuremberg
Pr. Emmanuel Vander Poorten — University of KU Leuven
Pr. Philippe Cattin — University of Basel

Alla mia famiglia,
la mia certezza più grande

Acknowledgements

Tout d'abord, je souhaite remercier mes directeurs de thèse, Stéphane Cotin et Florent Nageotte, ainsi que mon encadrant, Michel Duprez, qui m'ont fait confiance dès le début et m'ont permis de développer ce projet.
Merci à Stéphane pour tous tes conseils, pour ton infinie patience, pour l'ambiance que tu as créée dans l'équipe et pour tout ce que tu m'as appris, tant sur le plan professionnel que personnel. Cette thèse a été un parcours qui m'a permis de grandir et de changer. Je garderai toujours en mémoire les expériences incroyables de ces années et j'en ferai un véritable trésor.
Merci à Florent pour ton soutien constant et ta rigueur, pour tes encouragements dans les moments les plus difficiles et pour ton aide précieuse dans le développement de ce travail, même face aux demandes de dernière minute. Ton soutien a été fondamental.
Merci à Michel pour tout le travail accompli et pour tes paroles réconfortantes dans les périodes sombres où il était difficile de garder la motivation pour avancer.

I would like to express my gratitude to the ITI HealthTech consortium for financing this project, with special thanks to Bernard Bayle and Nicole Kirsch for their trust and support throughout these years.

I would sincerely like to thank the members of my thesis committee: Franziska Mathis-Ullrich, Emmanuel Vander Poorten, and Philippe Cattin, for agreeing to participate in such an important step in my career. I would like to extend a special thanks to the reviewers, Franziska Mathis-Ullrich and Emmanuel Vander Poorten, for taking the time to read and review my manuscript. Your comments have been invaluable in improving the quality of this work.

Merci à l'équipe MLMS, qui a été comme une famille durant toutes ces années. Merci pour toutes les parties de coinche et de tarot, et pour avoir rendu les longues périodes de deadlines moins difficiles. Merci à Sid et Robin, qui m'ont accueillie dès le premier jour avec un "on va manger chez le Tunisien, tu viens avec nous ?" et qui ont été de véritables amis tout au long de ces années. Merci à Vincent pour toute la légèreté et la joie qu'il a apportées chaque jour. Merci à Nico d'avoir partagé avec moi les longues soirées au labo (et de m'avoir aidée à installer les drivers de la carte graphique au moins un million de fois !). Merci à Claire d'être devenue une amie et une confidente, et merci à Fred, à Pierre et à tous les autres pour tous les

rires partagés.

Grazie ai miei punti fissi di Strasburgo, Paola, Spyros, Beni e Amona, per avermi sempre ascoltata ed essermi sempre stati vicino. Siete stati la mia casa lontano da casa.

Grazie alla Glo, alla Lalla, alla Marti, alla Michi, a Jacky e a Jaco per le chiamate, per la presenza e l'amicizia che ci lega da quando siamo piccoli, e per avermi sempre fatto arrivare il vostro calore. Grazie alla Flotta per farmi sentire che il nostro legame non cambia mai e per essere cresciuti insieme a me. Siete da sempre una delle mie costanti più grandi.

A huge thank you to Nick for always supporting me and being by my side. Thank you for the hours you spent encouraging me, listening to me rehearse my presentations, and offering advice. Thank you for teaching me to believe more in myself. Your support has truly been fundamental.

Ma il ringraziamento più speciale va alla mia famiglia. Ai miei genitori, Roberta e Stefano, per avermi accompagnata in ogni singola fase di questo cammino e per essermi sempre stati vicini con il loro affetto incredibile, nonostante la distanza fisica. Grazie per non avermi mai fatto sentire sola. Grazie a mio fratello Simone, per essere sempre stato presente e per aver sempre trovato spazio e tempo per me. Grazie ai miei nonni, Elda, Maria, Benito e Carlo, per aver sempre sostenuto le mie scelte e non essersi mai persi nessuna delle tappe di questo percorso. Grazie alla zia Cry per avermi sempre strappato un sorriso e avermi sempre ricordato che qualunque cosa succeda "non casca il mondo" e all'allegria che l'Ali è riuscita a portarmi a Strasburgo. Grazie a Luchi e ai miei zii Gabri e Andrea per essere sempre stati pronti ad ascoltare tutte le mie storie. So per certo che senza tutti voi non sarei mai riuscita a portare a termine questo percorso. Siete davvero il mio porto sicuro da sempre.

Abstract

Summary of the thesis

Cardiovascular diseases are one of the leading causes of death worldwide, with an incidence of 17.9 million deaths per year. The primary therapeutic solution is endovascular interventions, which owe their success to significant advantages, such as minimal invasiveness and low costs. However, these procedures are limited by their complexity, requiring extensive clinician training and access to specialized facilities. To deliver treatment, the clinician must navigate long, thin-tube devices, such as catheters and guidewires, through the patient's arteries, controlling them from the proximal end positioned at the entrance of the patient's vessel. This already challenging task is further complicated by the limited guidance the clinician has available, which is only provided by 2D fluoroscopic images. Furthermore, the acquisition of these images requires the use of X-rays, dangerous for the health of both the patient and the clinician, and the visibility of vessels is dependent on contrast agents, which can be harmful to the patient's kidneys.

To address these limitations, this manuscript proposes solutions to facilitate the intervention, through the development of methods able to offer the clinician more support during certain phases of the intervention while automating others. A simplified procedure can indeed result in lower surgical time and, as a consequence, lower X-ray exposure for both the patient and the caregiver. Two main systems have been developed: one that enhances fluoroscopic images and another that autonomously navigates surgical tools. The first one is essentially an assistance system, which overlaps the classical fluoroscopic images with information about the anatomy that is being navigated and shows the clinicians the predicted outcome of their actions before they perform them. The second one is a Deep Reinforcement Learning controller which aims to autonomously perform the procedure by controlling an endovascular surgical robot. Currently, these robots function only as leader-follower devices, which are not able to provide additional support to the caregiver during the procedure.

In tests conducted on a phantom in the context of a user study, the enhanced fluoroscopic system allowed for a 56% intervention time reduction. The autonomous controller achieved a success rate of over 95%, even when tested on anatomies with characteristics completely different from the training models.

Résumé de thèse

Les maladies cardiovasculaires sont l'une des principales causes de mortalité dans le monde, avec un nombre estimé de 17,9 millions de décès par an. La solution thérapeutique principale consiste en des interventions endovasculaires, qui doivent leur succès à des avantages significatifs tels que la faible invasivité et les coûts réduits. Cependant, ces procédures sont limitées par leur complexité, nécessitant une formation approfondie des cliniciens et l'accès à des installations spécialisées. Pour effectuer le traitement, le clinicien doit manipuler des dispositifs longs et fins, tels que les cathéters et les guide-fils, à travers les artères du patient, en les contrôlant depuis leur extrémité proximale positionnée à l'entrée du vaisseau. Cette tâche déjà difficile est encore compliquée par le guidage limité à la disposition du clinicien, fourni uniquement par des images fluoroscopiques en 2D. De plus, l'acquisition de ces images nécessite l'utilisation de rayons X, dangereux pour la santé à la fois du patient et du clinicien, et la visibilité des vaisseaux dépend des agents de contraste, qui peuvent être toxiques pour les reins du patient.

Pour répondre à ces limitations, ce manuscrit propose des solutions pour faciliter l'intervention, par le développement de méthodes offrant au clinicien plus de soutien pendant certaines phases de l'intervention, tout en automatisant d'autres. Une procédure simplifiée peut en effet réduire le temps chirurgical et, par conséquent, diminuer l'exposition aux rayons X pour le patient et le clinicien. Deux systèmes principaux ont été développés : l'un qui améliore les images fluoroscopiques et un autre qui guide de manière autonome les outils chirurgicaux. Le premier est essentiellement un système d'assistance, qui superpose les images fluoroscopiques classiques avec des informations sur l'anatomie naviguée et montre au clinicien le résultat prédit de ses actions avant qu'il ne les exécute. Le second est un contrôleur basé sur l'apprentissage profond par renforcement, conçu pour effectuer de manière autonome la procédure en contrôlant un robot chirurgical endovasculaire. Actuellement, ces robots fonctionnent uniquement comme des dispositifs suiveurs, incapables d'offrir un soutien supplémentaire au clinicien durant la procédure.

Lors des tests effectués sur un fantôme dans le cadre d'une étude utilisateur, le système améliorant les images fluoroscopiques a permis une réduction de 56 % du temps d'intervention. Le contrôleur autonome a atteint un taux de réussite de plus de 95 %, même lorsqu'il a été testé sur des anatomies très différentes des modèles d'entraînement.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

E Young Modulus

Greek Symbols

ν Poisson ratio

Acronyms / Abbreviations

A2C Advantage Actor Critic

A3C Asynchronous Advantage Actor Critic

BEM Boundary Element Method

CT Computed Tomography

DDPG Deep Deterministic Policy Gradient

DDPG Deep Deterministic Policy Gradient

DL Deep Learning

DPG Deterministic Policy Gradient

DQN Deep Q-network

DRL Deep Reinforcement Learning

DSA Digital Subtraction Angiography

DTG Draw Tower Grating

EMsensors Electromagnetic sensor

FBG Fiber Bragg Grating

FDA Food and Drug Administration

FDM Finite Difference Method

FEM Finite Element Method

FOS Fiber optic sensors

MCF Multi-Core Fibers

MDP Markov Decision Process

ML Machine Learning

MRI Magnetic Resonance Images

NN Neural Network

ODE Ordinary Differential Equations

OFDR Optical Frequency Domain Reflectometry

PDE Partial Differential Equations

PPO Proximal Policy Optimization

PVW Principle of Virtual Work

RL Reinforcement Learning

RMSE Root Mean Square Error

ROI Region of Interest

$R-PCI$ Robotic Percutaneous coronaries intervention

SAC Soft Actor Critic

Chapter 1

Introduction

Endovascular diseases are nowadays the main cause of death overall, with an incidence of 17.9 million deaths per year *(World Health Organization (2021))*. These pathologies involve a large set of conditions affecting the heart and the blood vessels, including among others coronary heart diseases, cerebrovascular diseases and rheumatic heart diseases *(World Health Organization (2021))*. In the past two decades, the development of endovascular interventions revolutionized the treatment of these conditions, allowing the delivery of therapies that before required open surgery in a minimally invasive way. During endovascular interventions, thin tube devices, called catheters and guidewires, are inserted inside the vessel of the patient and navigated by the caregiver until reaching the target location. During the whole intervention, the clinician is guided by fluoroscopic images, which are continuous and real-time 2D X-ray images, which show the patient's internal structures.

Endovascular interventions include various techniques such as stenting, embolization and ablation *(Chi et al. (2020a))*. The range of conditions treated through endovascular procedures is increasing as new techniques and materials are developed, making endovascular interventions an important part of patients' vascular care for the foreseeable future. Compared to classical surgical approaches, these interventions produce fewer adverse events, less pain, faster recovery times, and shorter hospital stays, thus significantly reducing costs. However, the limiting factor of endovascular procedures is their complexity, requiring extensive training and access to specially equipped facilities.

The work presented in this manuscript focuses on the development of systems to provide support to the clinician during these procedures. In the following, details on the different phases of the intervention will be provided, with a focus on the main challenges encountered by the caregivers during the surgery. Section 1.1 introduces the procedure, the surgical instruments and their manipulation. Section 1.2 describes the current drawbacks of the intervention, which motivate the work presented in this manuscript, whose objectives are summarized in Section

1.3. Finally, Section 1.4 provides a detailed outline of the chapters composing the manuscript, and Section 1.5 describes the contributions accomplished throughout this work.

1.1 The procedure

To guarantee the success of the intervention, mastering the combined manipulation of the catheter and guidewire is crucial. Efficiently controlling the devices is a complex skill that requires long training and excellent dexterity, besides a good knowledge of instruments' characteristics. With the increasing adoption of endovascular interventions as a therapeutic solution for endovascular diseases, more and more variants of catheters and guidewires have been developed, allowing the broadening of the range of conditions that it is possible to treat with this technique. Choosing the correct catheter-guidewire combination is essential to ensure a good outcome of the surgery: this is why, to fully understand the procedure, it is important to know the devices' main characteristics. Most of the concepts about endovascular interventions reported in this Chapter are documented in *Endovascular Skills: Guidewire and Catheter Skills for Endovascular Surgery* book *(Schneider (2019))*.

1.1.1 The devices

Both the catheter and the guidewire present a particular tip shape chosen based on the function and the district of the body that needs to be navigated. Several shapes are available for each instrument, which results in a very large number of guidewire-catheter combinations. In Fig. 1.1 some examples of catheters and guidewires are shown. Each instrument presents different characteristics that make it appropriate for a specific task.
Considering the guidewire, the main properties include:

- *The length*, which can vary from 145 to 300 cm. The length must ensure the coverage of the cumulative distance from the insertion point to the target location.
- *The diameter*, which varies from 0.035 inches for the navigation of large vessels, (*e.g.,* the aorta), to 0.014 inches for the treatment of small vessels (*e.g.,* the coronaries).
- *The stiffness*, which is defined by an inner steel core. During each intervention, different guidewires with different stiffnesses are used. Often, guidewires that present a stiff shaft, are equipped with a softer tip to avoid damaging the wall of the vessels.
- *The coating*, which defines the friction coefficient of the instrument. The most common materials used to cover the core of the instrument are Teflon or silicone.

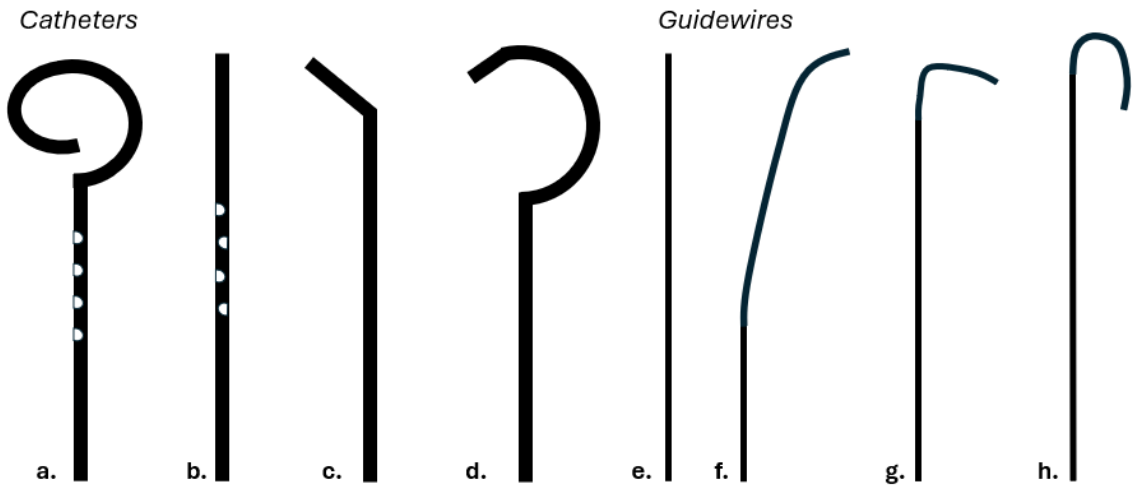

Fig. 1.1 Catheters and guidewires variations. a. Flush catheter b. Exchange catheter c. Simple curve selective catheter d. Double curve selective catheter e. Straight guidewire f. Angle M guidewire g. BK guidewire h. J 2mm guidewire.

- *The tip shape*. The tip of the device assumes different characteristics based on the function of the guidewire. Guidewires used in the first phase of the intervention are called *starting guidewires*, while the ones used to take a specific direction when a bifurcation is reached are named *selective guidewires* and often present a curved tip shape. These instruments are usually employed with a torque device, which takes the name of torque device, which, after being fixed on the shaft of the instrument, ensures that the torsion applied to this component is completely transmitted to the guidewire, simplifying the manipulation of the device. Finally, the *exchange guidewires* present either a straight or a curved tip. These devices are usually stiffer than the others and are used to maintain the reached position inside the vessel when the catheter needs to be exchanged: their stiffness indeed allows to safely withdraw the catheter that surrounds the guidewire and advance a new one, without risking the loss of the secured vascular access.

Catheters are characterized by:

- *The length*, which usually ranges from 65 cm to 100 cm. Also for this instrument, the length must be chosen to ensure the possibility of reaching the target location.
- *The outer diameter*, measured in French, ($1Fr = 0.33mm$) which usually ranges between 4 and 8 Fr. This dimension should be as small as possible to reduce the encumbrance inside the vessels and the damages but should, at the same time, allow to accomplish the task.

- *The tip shape*. Catheters present a more complex tip shape compared to guidewires, and also in this case the shape of the tip determines the function of the instrument. Catheters can be divided into three classes (Fig. 1.1): flush catheters, exchange catheters and selective catheters. The first class usually presents an end hole and multiple side holes to allow for a homogeneous injection of the contrast agent used during the intervention to visualize the vessel structures and are available in various shapes (Fig. 1.1 a), while the second class is usually straight and used when another guidewire must be placed in the artery (Fig. 1.1 b). Selective catheters present only an end hole and can be classified as simple-curved catheters or complex-curved catheters. The difference between these two devices is shown in Fig. 1.1 c and d: simple-curve catheters present only a single curve near their tip, which takes the name of *primary curve*, while complex-curve catheters are identified by two or three curves (*seconday* and *tertiary curve*.)

- *The traceability*, which defines their ability to follow the guidewire through tortuous vessels and around corners, without displacing it.

- *The pushability*, which characterizes how a force applied by the operator at the insertion point of the instrument is converted into movement of the tip of the instrument

- *The crossability*, which describes its ability to follow the guidewire across lesions

- *The torquability*, which defines how a rotation applied to the proximal end of the instrument, is transmitted to the tip.

- *The steerability*. Some catheters present a steerable distal portion, able to change its shape based on external controls, such as electromagnetic fields or cables. These catheters take the name of *steerable catheters (Fu et al. (2009))*. However, these devices will not be considered in this work and will be therefore not further discussed.

During the same intervention, several devices are exchanged based on the task that is being performed. Finally, it is important to mention that not all catheters and guidewires are compatible with each other, and it is therefore important to choose a set of devices suitable both for the application and for working together.

1.1.2 The phases of the intervention

The procedure starts with the patient placed on the operating table. The first step is the choice of the best location for the percutaneous access, which constitutes the access point to the vascular tree of the patient and the insertion point of the surgical instruments. This small incision is

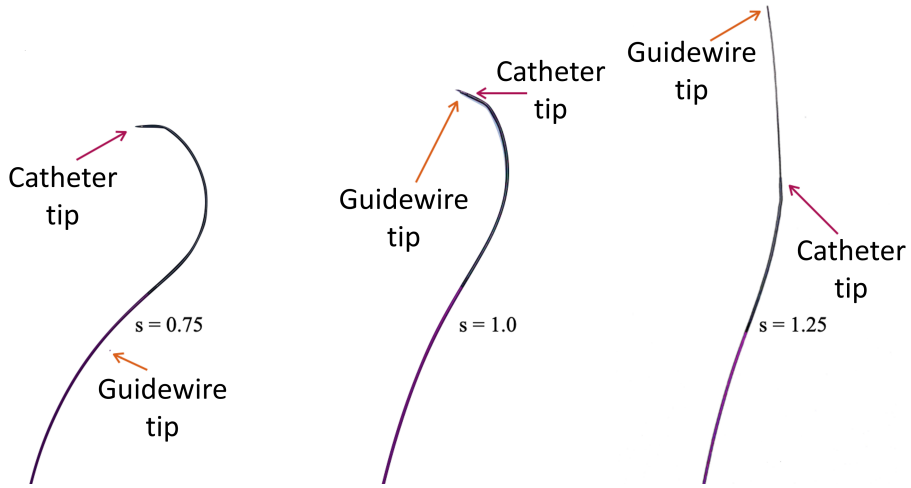

Fig. 1.2 Catheter shape modification as it is passed over the guidewire. The parameter 's' on each image indicates the local curvilinear coordinate of the guidewire within the catheter.

usually performed at the level of the femoral artery, the brachial artery, or the radial artery, but other options are available. Once the access point is identified, a needle is inserted to puncture the artery and a starting guidewire is advanced inside the needle. When the guidewire is sufficiently stable inside the vessel, the needle is withdrawn and a shaft is advanced over the guidewire. The shaft consists of a thin tube placed at the entry of the artery. It serves as the access point for the surgical tools and prevents bleeding during the procedure. When the access is secured, the navigation phase of the intervention starts, and the catheter and the guidewire are inserted into the vascular system of the patient. First of all, a starting guidewire is inserted through the needle. Its soft tip avoids puncturing the vessel wall. Once the guidewire position is secured, the catheter is advanced over the guidewire and the instruments are navigated together inside the vascular tree. The guidewire always preceeds the catheter during this phase, to avoid damaging the vessel wall. In this phase, the shape of the catheter is significantly modified by the guidewire. Indeed, when the guidewire is inserted inside the catheter, this latter assumes a relatively straight configuration, which helps to navigate straight vessels (Fig. 1.2).
The navigation of the devices toward the target location usually involves two main distinct stages:

- The *selective cannulation*, which involves the access from a large main vessel to small lateral vessels (Fig. 1.3 b).
- The *navigation of small downstream vessels*, which involves the navigation of several bifurcations to reach the target location (Fig. 1.3 c).

Each intervention requires these two steps to reach the target, but each procedure requires a specific chain of actions which can involve the repetition of the same operation several times

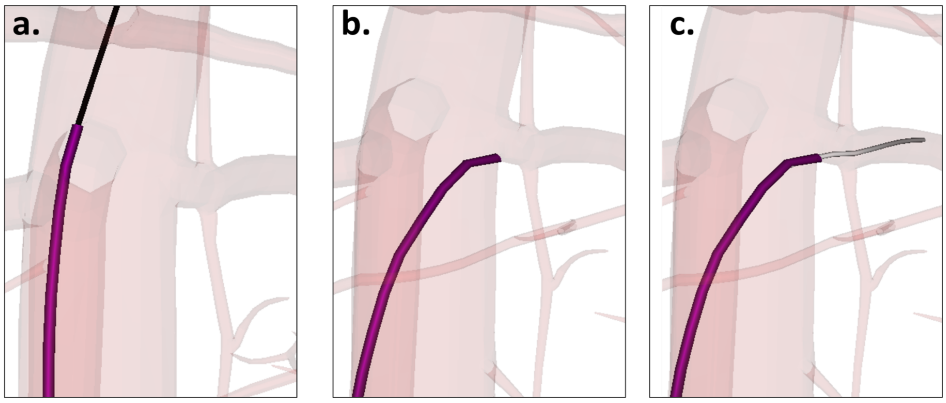

Fig. 1.3 Endovascular interventions steps. At the beginning of the intervention, the two devices are navigated together (a). Then, the guidewire is withdrawn, to let the catheter recover its original shape and place its tip at the entrance of the target vessel (*selective cannulation*, b). Finally a softer guidewire is advanced in the catheter and the downstream vessels can be navigated (c).

and the use of specific devices with varying complexity.

Considering for example the navigation of the renal arteries (Fig. 1.4 left), the procedure starts with the insertion of the devices into the femoral artery. In this case, a specific selective catheter, such as the Cobra catheter (*Merit Medical*), is used. At the beginning of the intervention, the catheter and the guidewire are navigated together, and the catheter, being advanced over the guidewire, assumes a straight configuration. The instruments are navigated together inside the vessel and they traverse the abdominal aorta until the ostium of the renal arteries is reached. At this stage, the guidewire is withdrawn to let the catheter reshape in its original configuration inside the main vessel. The particular shape of the chosen catheter permits to position the tip of the device at the entry point of the renal arteries. Once the access is secured, a soft guidewire is advanced inside the selective catheter and navigated inside the downstream renal arteries. This device presents a curved tip, that allows to navigate the correct branch when a bifurcation is reached.

In other cases, the intervention is more complex and more devices are required to reach the target location. It is the case for example of the coronary navigation (Fig. 1.4 right). Among the most suitable devices, the Judkins (*Merit Medical*) or the Amplatz (*Merit Medical*) catheters can be identified. Also in this scenario, the catheter and the guidewire are navigated together inside the aorta, and the straightening effect of the guidewire allows to obtain a configuration suited for the navigation of this main vessel. In this case, the navigation is continued through the aortic arch, and the coronary ostium, in which the coronaries originate, is reached. After the guidewire withdrawal and the placement of the tip of the selective catheter at the entrance of the target vessels, a second softer catheter and a guidewire are advanced through the selective

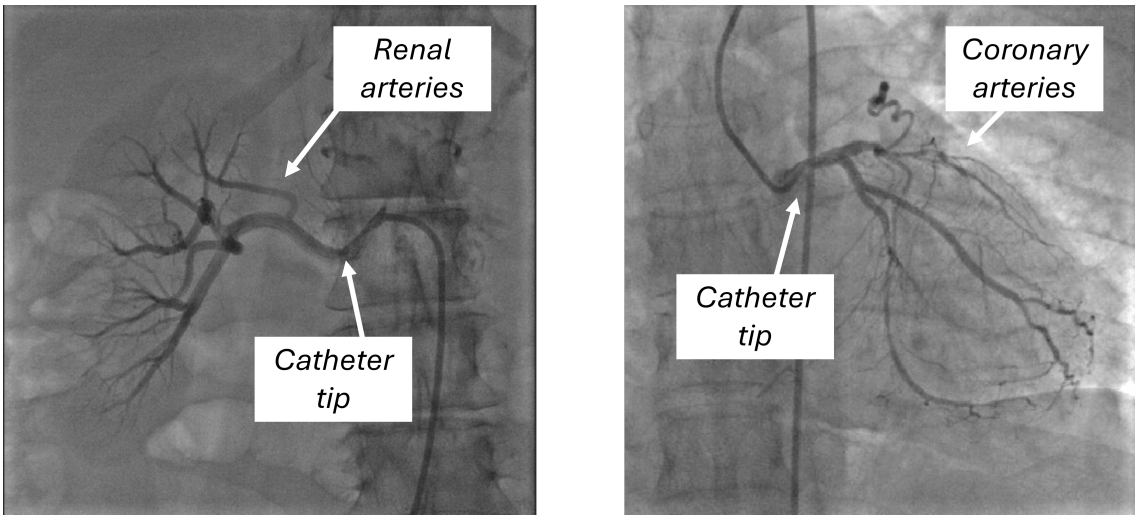

Fig. 1.4 Left: Contrasted fluoroscopic image of the renal arteries. Image from *Bhat et al. (2015)*, license: CC BY 3.0. Right: Contrasted fluoroscopic image of the coronary arteries. Image from *Ibrahim et al. (2012)*, license: CC BY 3.0. In both pictures, it is possible to identify the vessel structures and the selective catheter used to access the renal vessels and the coronary arteries.

catheter and navigated inside the coronaries. The guidewire presents an extremely soft tip, whose only purpose is avoiding puncturing the vessel wall, and it is directed by the catheter which presents a curved tip. Once again, it is the curvature of the catheter that allows the navigation of the branch leading to the target location.

1.1.3 The manipulation of the devices

The crucial aspect of endovascular interventions is the precise control of the tip of the devices. The correct positioning of this distal part of the instrument is the element that allows to safely reach the target location during all the stages of the procedure: in the navigation phase, in which it is important to keep the tip of the device as far as possible from the wall of the vessels to avoid damages, in the selective phase, in which the tip needs to be placed at the entrance of the target vessel, and in the navigation of the bifurcations, in which it is crucial to correctly direction the tip of the device to navigate the branch that leads to the target location. One of the primary challenges of the procedure is that the caregiver can only access the proximal part of the device, placed at the level of the percutaneous access: to place and orient the tip of the instrument, the clinician can only rotate and translate the tools from their basis, which, as the procedure advances, is progressively further away from the instrument distal end. This means that, due to the increasing constraints imposed by the vessels' geometry on the instruments, a control applied to the proximal end of the devices may not result in the desired motion of the tip of the instrument.

This already difficult control is further complicated by the reduced information the clinician has available during the intervention. During the whole procedure, the caregiver is only guided by 2D fluoroscopic images, acquired by means of X-rays. This kind of image can show the vessel structure only upon contrast agent injection (Fig. 1.5 a). Indeed, the vessels and the surrounding soft tissues present a similar X-ray absorption coefficient, which makes it difficult to distinguish these structures on fluoroscopic images. Contrast agents, which are typically iodine, are able to absorb a high amount of external X-rays and, when injected into the vessels, allow for the clear identification of the vascular anatomy on the fluoroscopic images. The fluoroscopic images used to guide the intervention are obtained through the *roadmap* technique, which generates real-time images of the blood vessels navigated by the instruments. These images are obtained by combining real-time fluoroscopic images with a static image of the vessel anatomy. The static angiographic image is used as a *roadmap* to observe how the devices are advancing inside the artery of the patient. This technique is particularly useful during the most delicate phases of the intervention, such as the identification of the vessel origin during selective cannulation or the passage through a stenosis. The static angiographic image can be obtained through several techniques. The most used, both in the diagnostic and planning phase, is the digital subtraction angiography (DSA). This technique is particularly interesting because it allows to eliminate from the fluoroscopic images the shadow produced by the overlapping tissues and bones. This is done by first acquiring a *mask*, which is a non-contrasted image of the region of interest (ROI), which only shows the anatomy (*e.g.,*, the bones, etc) and all the radiopaque bodies (*e.g.,* stents, surgical clip, etc.). A series of contrast images are then acquired in sequence while the contrast agent is injected: in these images, the anatomy, the radiopaque elements and the contrasted vessel are all present. The mask image is then subtracted pixel by pixel from the series of contrasted images, allowing to obtain images in which only the vascular anatomy is shown. Especially during the angiographic phase, the contrast agent is usually delivered through a flush catheter, which can present several holes to ensure a good liquid distribution.

1.2 Motivations

1.2.1 Main intervention drawbacks

The main drawbacks of endovascular intervention can be summarised as follows:

1. Fluoroscopic images are only 2D images: this might mask the knowledge about the direction of the tip or the vessel structure when these elements are outside the plane of the fluoroscopic image, making the control of the devices more challenging.

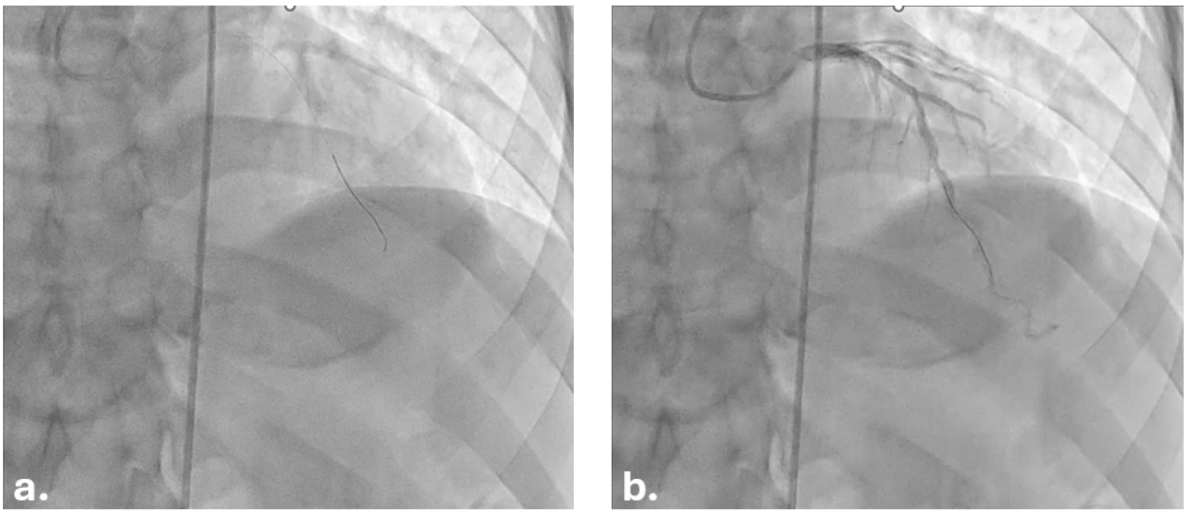

Fig. 1.5 Contrasted (a) vs non-contrasted (b) fluoroscopic image of the coronaries. The vessels are only visible upon contrast agent injection. Furthermore, the coronaries move: the static angiographic image used to build the roadmap might not be sufficient to properly observe the devices inside the wall of the vessel, and with each injection of contrast agent the vessels are visible only for a couple of seconds.

2. As already stated, the acquisition of fluoroscopic images requires the use of X-rays: during the procedure, both the patient and the clinician are exposed to these radiations. The health risks associated with high X-ray exposure are multiple and include cataracts, thyroid cancer, hematopoietic cancer, skin cancers, brain tumors, and sterility *(Schneider (2019))*. It is thus vital to reduce as much as possible the X-ray exposure for both the patient and the clinician (ALARA (*As Low As Reasonably Achievable*) principle *(Yeung (2019))*), who is exposed to this radiation multiple times per day. The guidelines indicate that for an adult, the maximum permissible dose is 5 rems (roentgen equivalent man $=$ 0.01 sievert) per year *(Schneider (2019))*.

3. A high dose of contrast agent is dangerous for the health of the patient's kidneys, leading to the risk of developing contrast-induced nephropathies, especially if pre-existing chronic kidney diseases are present. Other risk factors include advanced age, diabetes, and peripheral heart failure *(Hossain et al. (2018))*. To minimize this risk, the number of contrast agent injections during a single intervention is limited. The problem is that when navigating moving structures, such as the coronaries or the liver vessels, the static angiographic image acquired to build the roadmap is not sufficient to observe the movement of the devices properly, and a higher number of contrast agent injections is needed (Fig. 1.5).

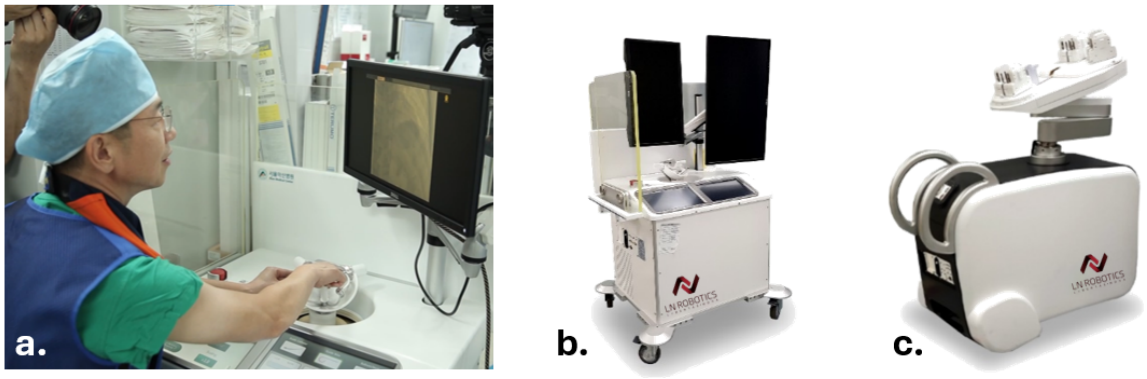

Fig. 1.6 Endovascular robots - Joystick that the clinician uses to control the endovascular robot (a), leader components of the system used by the caregiver (b) and follower part of the system responding to clinician's controls.

4. Finally, due to poor visualization, incorrect control of the devices may be applied, potentially resulting in puncturing or damaging the vessel walls.

The aforementioned problems pose some serious challenges during the intervention: the 2D view of 3D tools and anatomical structures can make it difficult to correctly control the devices on the first attempt. During selective cannulation, for example, the clinician needs to predict what is the best time to stop the advancement of the combined devices and to withdraw the guidewire, to let the catheter recover its original shape and reach with its tip the ostium of the target lateral vessel. Given the limited source of information the clinician has during the intervention, this operation can require several attempts and sometimes it needs to be repeated several times during a single intervention to reach the final target that can be, in the case of neuro-vascular interventions, more than 1 meter away from the insertion point. Similarly, when navigating small downstream vessels, it is necessary to navigate several bifurcations and for each bifurcation, the correct and precise steering of the device is essential to reach the target. This increases the intervention time and, as a consequence, the number of contrast agent injections and the X-ray exposure for both the patient and the clinician.

1.2.2 Current solution

The only partial solution that exists today is the use of interventional robots (Fig. 1.6). These systems have been developed to improve the surgical workflow during endovascular interventions *(Duan et al. (2023))* and several studies have demonstrated that robot-assisted interventions can help overcome the drawbacks of endovascular interventions *(Antoniou et al. (2011); Bonatti et al. (2014); Kazanzides et al. (2008))*, introducing many benefits both for the patient and the clinician. Having a mechanism that grasps the surgical instruments, these systems allow the

clinician to apply a precise control of the intervention tools and to perform small adjustments in the position and rotation of the devices. They can this way improve the intervention's accuracy, offering increased navigation precision and enhanced stabilization during the selective cannulation phase. They can also reduce the intraluminal vessel injury risk that can result from imprecise control of the devices due to the operator's hand tremors, enhancing the safety of the procedure. Furthermore, these systems can be teleoperated by the clinician, thus reducing the problem of X-ray exposure for the caregiver.
In the past two decades, various commercial endovascular systems have been proposed for electrophysiology and general vasculature applications. The most famous commercial systems include the CorPath (Corindus Vascular Robotics, MA, USA), the R-One (Robocath Inc., Rouen, France), the Sensei (Hansen Medical Inc., CA, USA), the Magellan (Hansen Medical Inc., CA, USA) and the Amigo (Catheter Precision Inc., NJ, USA). Corindus CorPath © 200 was the first robotic system to obtain Food and Drug Administration (FDA) approval in 2012 for coronary intervention. The second-generation CorPath is equipped with an extended reach arm, a disposable cassette, and a lead-shielded workstation. The disposable cassette, in particular, allows this system to manipulate some of the commercially available catheters. Thanks to this component, which can be easily sterilized, the simultaneous control of the guidewire, the guide-catheter balloon, and the stent catheter is possible. The main drawback of this system is the high cost of consumables and the lack of distal force feedback *(Duan et al. (2023); Mahmud et al. (2017); Pourdjabbar et al. (2017))*. Robocath R-One system has a similar design and functional capability and was also designed for robotic percutaneous coronaries interventions (R-PCI). The Hansen Sensei and Magellan systems have been designed for different purposes - respectively the study and treatment of cardiac electrophysiology and the treatment of peripheral vascular diseases - but they share a similar design. In particular, they achieve 6 DoF applying a guide sheath consisting of a steerable outer sheath and a steerable inner leader *(Kanagaratnam et al. (2008); Li et al. (2024a); Rafii-Tari et al. (2016))*. The main limitation of these systems, besides the high cost and the lack of haptic feedback, is that they require the manual placement of the interventional devices (balloons, stents) after surgical access has been established. Finally, the Amigo system allows the manipulation of the handle of the commercial standard endovascular procedure catheters.

Another category is constituted by the systems that use a magnetic field to control the instruments. One example is the Niobe (Stereotaxis Inc., MO, USA) and its upgraded version the Genesis (Stereotaxis Inc., MO, USA). These systems place two magnets on opposite sides of the patient bed. When these magnets rotate, the magnetic field changes and, as a consequence, the employed catheters and guidewires, which are specifically designed to modify their tip based on the magnetic field in which they are inserted, are adjusted accordingly *(Ernst*

et al. (2004); Huang and Miller (2019); Singh (2022)). Other systems, such as the Guidance Control and Imaging (CGCI) system (Magnetecs Corporation, CA, USA) and the Aeon Phocus (Aeon Scientific AG, Zurich, Switzerland) employ electromagnets: in this case, to modify the magnetic field, it is only necessary to change the electrical current that flows through the magnets, which allows for faster control of the devices *(Hwang et al. (2020); Pourkand and Abbott (2018))*.

All the mentioned systems are just leader-follower devices, manipulated by the caregiver through a console and master interface, such as the one shown in Fig. 1.6 a. None of these systems can therefore provide the clinician with more intraoperative support: to increase the application of these robots, it is necessary to augment their autonomy and their machine intelligence *(Duan et al. (2023))*.

1.3 Objectives

It is in this context that the work described in this manuscript takes place. The aim is to develop a system able to support the clinician during the whole intervention, developing algorithms that automate some phases of the intervention and that provide visual support in others. The final objective is to reduce the intervention time and, as a consequence, the number of contrast agent injections and the radiation intake for both the patient and the clinician. During the development of this work, two main topics have been addressed:

- The development of a system able to support the clinician during the selective cannulation phase, enriching the information provided by the fluoroscopic images. In particular, this system offers a real-time augmented view of fluoroscopic images in which besides the devices navigating the anatomy, the virtual shape of the catheter after guidewire withdrawal is shown, to help the clinician understand when it is the right moment to withdraw the guidewire to obtain a successful cannulation.
- The development of an autonomous control algorithm to navigate both the catheter and the guidewire toward the target location in all possible anatomies, both in static and dynamic cases.

To attain these objectives, several components are necessary, ranging from a reliable simulated environment, able to reproduce as accurately as possible the physics of real devices navigating the anatomy, to strategies to integrate real data into the simulation, from criteria that ensure the navigation toward the target location when a bifurcation is reached, to strategies to obtain a controller able to handle various anatomies.
The first system is essentially an assistance framework, in which the actions are decided and

executed by the clinician, supported by our enhanced fluoroscopic images. The core of the second system can be described through the control theory: this theory indeed is suited for those cases in which it is necessary to lead the system from the current state -*e.g.,* the insertion point - to the desired final state -*e.g.,* the target.

1.4 Outline

In this chapter, to provide the complete frame in which this work is developed, the general tools used during endovascular interventions and the general steps performed during the surgery were described. The main challenges faced by the clinicians during the intervention and the current technological innovations developed to try to overcome these limitations were also introduced.
The remaining of the manuscript is organized as follows.

Chapter 2 provides a short introduction to the fundamental concepts of control theory. As already mentioned, one of the systems developed in this work is essentially a control algorithm. This controller, which is an autonomous algorithm that interacts directly with the surgical devices and the anatomy, takes as input information regarding the anatomy to be navigated and the shape and position of the navigated instrument and gives as output the control needed to attain the objective. After introducing the basic concept of control theory in Section 2.1, details are provided on a specific category of control algorithms, which adopts Deep Reinforcement Learning (DRL) techniques to compute the optimal control. These algorithms present many common points with the classical optimal control theory. However, they are more suited for those cases in which it is difficult to build a precise representation of the problem and they were therefore employed in this work. Section 2.2 provides the fundamentals of DRL theory, in which an overview of deep learning (DL) principles and neural networks (NN) is provided in Section 2.2.1, and an introduction to the fundamental knowledge behind reinforcement learning (RL) algorithms and their essential characteristics is given in Section 2.2.2. Finally, Section 2.3 motivates the choice of a DRL algorithm for the development of our controller, comparing this strategy with the optimal control theory.

For the development of both systems, the accurate modeling of the physics of the devices and their contact with the anatomical structures is fundamental. The theoretical framework for building this model is provided by the finite element theory, whose main concepts are introduced in the first half of Chapter 3. After presenting the general formulation of the finite element method (FEM, Sections 3.1.1, 3.1.2), the principle of virtual work, which provides the mathematical fundamentals to apply the FEM to the mechanics of solids, is illustrated in Section 3.1.3. The second half of this Chapter describes the simulated environment, based on

the FEM, developed in this work. This environment is fundamental both for the creation of enhanced fluoroscopic images and for the development of the autonomous controller. In the first case, a reliable physical model is crucial to predict the shape of the catheter after guidewire withdrawal, without actually performing the task. In the second case, to then transfer the learned skill to the real world, it is necessary to reproduce as accurately as possible the physics of the instruments. For both applications, it is essential to maintain real-time performance. All the elements regarding the choice of the model for the instruments (Section 3.2.1), the modeling of the contacts between the devices and the vessel wall and the devices themselves (Section 3.2.3) are described in this chapter.

Both systems rely on the strong assumption of knowing the shape and position of the surgical instruments during the intervention. It is therefore fundamental to have a reliable method to obtain this information. Given their theoretical excellent characteristics, which include distributed measurements, electromagnetic immunity and reduced size, fiber Bragg grating (FBG) sensors were adopted. The reasons behind the use of these sensors are discussed in Section 4.1, along with the considered alternatives. The working principles of the FBGs are reported in Section 4.2 and the process adopted by commercial systems to reconstruct the shape of the sensor is described in the same section. However, FBGs, like all measurement systems, exhibit limitations and measurement errors. These limitations, discussed in Section 4.2.3, required new strategies to enhance the precision of the sensor. In this work, two main solutions were adopted, which will be described in Section 4.3.

After introducing all the tools used in the development of this work and the techniques adopted to overcome the encountered limitations, the remaining chapters describe in detail the systems presented in this work.

Chapter 5 describes the framework developed to enhance fluoroscopic images. First of all, the context in which this work takes place is presented (Section 5.1). Then, the specific details on the method implementation are provided in Section 5.2, in which our data-driven endovascular navigation technique is introduced. Finally, Section 5.3 provides details on the tests performed to evaluate the accuracy and usefulness of the proposed system.

Chapter 6 gives details on the first version of the autonomous controller developed in this work. After reviewing the methods and limitations of the studies presented in the literature (Section 6.1), the algorithm chosen as controller is described, along with the adopted libraries. Finally, some details on the organization of the project are provided (Section 6.2). The core and main novelty of the work is described in Section 6.3, which introduces our zero-shot reinforcement learning strategy, our nearly shape-invariant observation space, and our reward function. To develop these elements, a thorough study was conducted to fully understand the characteristics of the problem (Section 6.3.1). Indeed, to construct an effective DRL training

environment, a deep knowledge of the problem is fundamental. Finally, the trained controller was tested in a simulated environment on completely new anatomies and the results are reported in Section 6.4. The system described in this chapter was also tested in the real world and in an open loop to inspect the transferability of the learned control, and the results are reported in Section 6.5.

Chapter 7 expands the work presented in Chapter 6 to approach more realistic cases. In particular, in this work, the training strategy is improved to learn a control of the device even when the vascular anatomy is moving (Section 7.1), overcoming the limitation of the previous work which considered only static anatomies. For this purpose, it is necessary to know the movements of the vessels: the method developed by François Lecomte, a PhD student of MIMESIS team, is therefore integrated in this work. This strategy provides a technique to estimate the motion of the anatomy from single-view fluoroscopy images and it will be described in Section 7.2 for the sake of completeness. The combination of these two contributions makes it possible to automatically navigate across a moving vascular anatomy under fluoroscopic imaging, even without injecting a contrast agent. The method has been validated in two simulated scenarios, a beating heart and a liver subjected to breathing motion, and the results of these tests are reported in Section 7.3.

Chapter 8 describes the current research topic and presents a system able to simultaneously control two instruments, a catheter and a guidewire, to allow performing the autonomous cannulation of small lateral vessels. This work constitutes the natural evolution of the assistance system presented in Chapter 5 that provides support to the clinician precisely in the cannulation phase. This system is fused with an expansion of the control strategy for a single instrument reported in Chapters 6 and 7. The development of such a controller allows to obtain a complete pipeline for autonomous endovascular interventions: while the previously described controller permits the navigation of small downstream vessels, the algorithm presented in this chapter allows the automation of the cannulation phase of the intervention. To reach the lateral vessels with the tip of the catheter (Section 1.1.3), two elements are fundamental: the correct orientation of the tip of the combined devices and the correct tip shape (Section 8.1). While the first can be obtained with a technique similar to the previous one, the second requires a new strategy and a new criterion, described in Section 8.2. The criteria to obtain the correct tip orientation and the correct tip shape are finally fused together, to achieve a control strategy able to simultaneously manage a catheter and a guidewire (Section 8.3), and perform this way an autonomous cannulation.

Finally, Chapter 9 reports some final considerations about the work developed during this PhD project, along with the current limitation of the work and the possible solutions.

1.5 Contributions

Throughout the development of this doctoral thesis, the obtained results were published in the following papers:

- Scarponi, V., Verde, J., Haouchine N., Duprez, M., Nageotte, F. and Cotin S., *FBG-Driven simulation for virtual augmentation of fluoroscopic images during endovascular interventions*, Special Issue on AE-CAI | CARE | OR 2.0 of Wiley's Healthcare Technology Letters journal - *Winner of the Outstanding Paper Award*, *In press.*
- Scarponi, V., Lempte F., Duprez, M., Nageotte, F. and Cotin S., *Autonomous Guidewire Navigation in Dynamic Environments*, Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024)
- Scarponi, V., Duprez, M., Nageotte, F. and Cotin S., *A zero-shot reinforcement learning strategy for autonomous guidewire navigation*. Int J CARS 19, 1185–1192 (2024)
- Lecomte F., Scarponi V., Alvarez, P., Verde J., Dillenseger JL., Vibert E. and Cotin S. (2023). *Enhancing Fluoroscopy-Guided Interventions: a Neural Network to Predict Vessel Deformation without Contrast Agents*. 75-76. 10.31256/HSMR2023.38.

Furthermore, the work was presented at the following conferences and congresses:

- International Conference on Intelligent Robots and Systems (IROS 2024)
- International Conference on Information Processing in Computer-Assisted Interventions (IPCAI 2024)
- Joint AE-CAI | CARE | OR 2.0 workshop at MICCAI 2024
- Hamlyn Symposium on Medical Robotics (HSMR 2024)
- European Symposium on Vascular Biomaterials (ESVB 2023)
- Orthomanufacture 2024

In addition to the scientific research work, I covered the position of assistant professor in the following classes:

- *Continuum Mechanics, practical class (TD)*, academic years 2022-23, 2023-24.
- *Biomateriaux pour la santé - La Mécanique Continue (CM)*, academic years 2022-23, 2023-24.

- *Introduction to Scientific Reporting, (CM)*, academic year 2022-23.

During these three years of research, two side projects were started from this study:

- Timothée Portha's master thesis project (March-August 2022), focusing on the development of a prototype of a robot able to control the devices
- Francesco Dettori's master thesis project (September 2023-August 2024), focused on the development of a method to improve FBGS shape sensing system

Chapter 2

The control problem

As introduced in Chapter 1, besides reporting the details of the implementation of an assistance tool to facilitate the cannulation phase of the procedure, this manuscript aims to describe the strategies developed to obtain a controller able to autonomously steer the surgical devices during the two main phases of endovascular interventions, *i.e.,* the navigation of downstream vessels and the selective cannulation of small lateral branches while navigating large main vessels. The fundamental aspects of the control theory, necessary to understand the remaining of the manuscript, are therefore introduced in this chapter (Section 2.1). As motivated in the following sections, an intelligent controller that exploits Deep Reinforcement Learning (DRL) algorithms is adopted in this work. Section 2.2 will therefore cover the fundamental concepts of this discipline, which is gaining increasing attention in the control field.

2.1 Control theory

Control theory is a field of control engineering and applied mathematics that aims to develop a strategy to drive a system of interest to a certain state, by acting on its behavior. It is based on the assumption that all the phenomena can be described as dynamic systems, governed by a series of ordinary differential equations (ODEs). A dynamic system is a system whose state changes over time, usually in response to external stimulation or perturbations. This brief introduction to the control theory is based on the following books: *Modern control engineering (Ogata (2010)), Feedback systems. An Introduction for Scientists and Engineers (Astrom and Murray (2008)), Introduction to control theory and its application to computing systems (Abdelzaher et al. (2008)).*

Control theory appeared in the 19^{th} century and the first significant steps in the field were accomplished during the 20^{th} century thanks to the frequency domain response, whose main contributions are due to Hendrik W. Bode, and to the root-locus method, developed by Walter

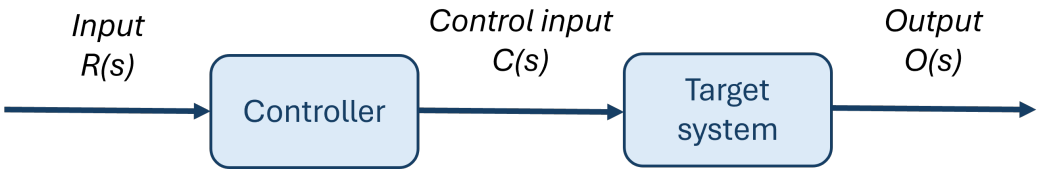

Fig. 2.1 Open-loop control representation.

R. Evans. These two methods constitute the core of the classical control theory which led to the development of the first control systems, characterized by good stability and performance criteria but unable to reach optimality. The blossoming of increasingly complex systems required the development of more sophisticated theories, able to deal with systems taking several inputs.

Nowadays, control engineering requires the simultaneous use of tools from physics and tools from operational research, such as optimization theories. This theory is applied to several engineering fields, ranging from the design of propulsion systems for aircraft to the study of robotic systems for assisted medical interventions.

Regardless of the application, all the control systems share the same general architecture, shown in Fig. 2.1. The system usually takes as input a signal $R(s)$, named *reference*, which represents the desired value of the output. This signal is given as input to a controller, that determines the needed control input $C(s)$ to reach the desired state. This signal acts on the target system, which, in response to this signal, changes its state and returns an output $O(s)$. This simple setting takes the name of *open-loop control*. In this case, the output has no effect on the control action: the output is neither measured nor fed back to the system. This means that independently from the current state, the same control will correspond to a certain input. It is therefore fundamental to accurately calibrate the system. The absence of influence of the output on the input of the controller means that even small disturbances will prevent the system from performing the desired task. To solve this problem, the open-loop configuration is substituted by the closed-loop one, shown in Fig. 2.2. These systems include a *feedback chain* which takes the output signal and feeds it back to the controller. This signal is usually subtracted from the reference signal $R(s)$ to compute an error $E(s) = R(s) - M(s)$, which constitutes in this case the input of the controller. These systems can maintain a prescribed relationship between the output $O(s)$ and the reference input and are able to handle possible disturbances. This characteristic usually allows to obtain an accurate control, even using slightly inaccurate components, such as imprecise sensors or models. The downside of this system is the difficult design: closed-loop control systems might be unstable, and it is therefore important to accurately design the components and anticipate the range of the input signal to avoid instabilities. For a closed loop system indeed the stability can be defined as the capability of the system to return a bounded

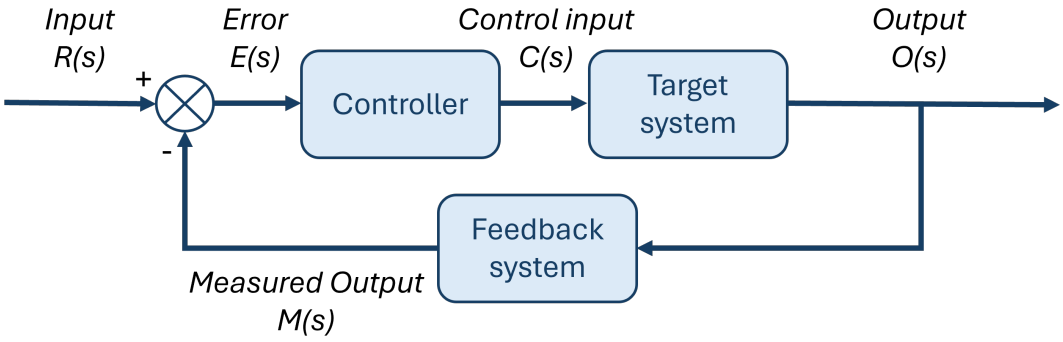

Fig. 2.2 Feedback control loop representation.

output when the input is bounded. Other important characteristics of a closed loop system are its accuracy, which measures the differences between the output and the reference, and its settling time, which estimates the time required to obtain a stable output.

The choice of the best design of the system is dependent on the application and on the usage conditions. Open-loop systems tend to be easier to design and maintain, more stable, less expensive and more convenient when it is difficult to measure the output but they become unreliable when disturbances occur or when the calibration deteriorates. Furthermore, open-loop control cannot be implemented when the system is intrinsically unstable.

Different control techniques exist today, with slightly different objectives and characteristics. Robust control, for example, directly deals with uncertainties. These controllers are designed by first setting up the possible range of modeling errors and then designing a controller able to provide the required output when the errors are in the expected range. Stochastic control, on the other hand, assumes disturbances and noises in the observations. However, when trying to efficiently perform a task, the best strategy is to use the optimal control theory. This strategy aims at performing a certain activity, by minimizing a given variable. An example is the choice of the optimal route, which does not only require finding a path between the starting and the ending point, but also requires to do it following the shortest path or the lowest fuel consumption. Considering endovascular interventions, it is important to perform the navigation task reducing as much as possible the trauma to the vessel wall. This means that the actions applied to the surgical devices should be the *minimal*, *i.e.,* useless actions, such as unnecessary rotations, should be minimized.

Mathematically, the optimal control problem can be formulated as the minimization of a function $f(x)$. A given x^* is then defined as a minimum if it satisfies the following conditions:

$$\begin{aligned} \nabla f(x^*) &= 0 \\ Hf(x^*) &> 0 \end{aligned} \tag{2.1}$$

where $\nabla f(x^*)$ represents the gradient of $f(x^*)$ and H the Hessian of $f(x^*)$.
When dealing with real-world problems, it is usually necessary to impose additional constraints, which can be defined as equality constraints or inequality constraints. The complete optimal control problem can then be defined as:

$$x^* = \min_{x \in \mathbb{R}^n} f(x) \qquad \text{with} \quad h_i(x) = 0, \quad g_j(x) \leq 0 \tag{2.2}$$

where $h_i(x)$ represent the $i-th$ equality constraints and $g_j(x)$ the $j-th$ inequality constraints. This problem is usually solved by employing the Lagrangian multipliers.
In certain cases, high-dimensional and non-linear problems cannot be solved using classical control theory and a new control category, referred by *de Silva (2009)* as intelligent control, has therefore been developed. These algorithms use various artificial intelligence approaches and can be classified based on the adopted strategy, which includes neural networks, Bayesian probability, machine learning and reinforcement learning control strategies. In particular, the last category raised more and more interest in the development of autonomous robotic systems in all engineering fields, providing effective solutions where traditional methods failed *(Buşoniu et al. (2018))*. DRL algorithms are particularly effective when the goal is to solve a problem that can be formulated as an optimal control task. These algorithms constitute a subfield of machine learning and they consider problems in which a computational agent learns to take the best decision to complete the desired task through a trial-and-error process *(Sutton and Barto (2018))*. As it will be explained in Section 2.3, optimal and deep reinforcement learning control problems share a lot of similarities. The application cases explore a wide range of disciplines, such as flow control *(Vignon et al. (2023))*, the control of the trajectory of vehicles *(He et al. (2024))*, the control of heating, air conditioning and ventilation in residential buildings *(Fu et al. (2023))*, and many more. In the medical field, these controllers helped in the development of various systems, such as controllers for lower limb rehabilitation exoskeletons *(Luo et al. (2023))* or controllers for the regulation of the real-time stimulation amplitude of electrical impulses used in deep brain stimulation, which showed promising results in the treatment of motor symptoms associated with Parkinson's disease *(Gao et al. (2023))*. Several works applied these techniques also to the autonomous control of catheters and guidewires during endovascular interventions *(Karstensen et al. (2020); Kweon et al. (2021); Tian et al. (2023); Tibebu et al. (0 14); Wang et al. (2022))*.

Inspired by these works and the encouraging results, deep reinforcement learning techniques have been applied also in the work presented in this manuscript for the development of an autonomous controller for endovascular interventions.

2.2 Deep Reinforcement Learning

In the past decade, machine learning (ML) techniques have gathered more and more interest, showing their ability to solve complex problems. The main potential of ML algorithms is their capability to detect patterns in the data and use them to perform some tasks. Among the different categories of machine learning, reinforcement learning (RL) achieved very good performances in the field of sequential decision-making, providing a formal framework for this task. Sequential decision-making is the process of using experience to decide the sequence of actions to take to reach a certain goal in an uncertain environment *(François-Lavet et al. (2018))*. The main idea behind RL is that an artificial agent can learn from experience by interacting with its environment. Its goal is to optimize an objective given in the form of a cumulative reward. When trying to solve a high-dimensional problem, the design of a RL algorithm might become very challenging *(Bellemare et al. (2012))*. To solve this issue, RL techniques have been combined with deep learning (DL) methods (DRL). Deep Learning is another fundamental branch of ML, whose methods can be classified into three main groups: *supervised learning*, which infers the classification or regression from labeled data, *unsupervised learning*, in which the inferences are drawn from input data without labels, and *semi-supervised learning*, in which a mixture of labeled and unlabeled data is used. DRL demonstrated its capability to solve complex tasks with little prior knowledge thanks to its ability to learn different abstraction levels from data and was successfully applied to various fields ranging from games *(Brown and Sandholm (2018))* to finance *(Deng et al. (2017))* and from self-driving cars *(Kiran et al. (2021))* to robotics *(Levine et al. (2016))*.
In the following, basic concepts of both DL (Section 2.2.1) and RL (Section 2.2.2) principles will be given, in order to provide a complete theoretical framework for DRL algorithms, introduced in Section 2.2.3. The following sections will be mainly based on *An introduction to Deep Reinforcement Learning* text *(François-Lavet et al. (2018))*.

2.2.1 Deep Learning

Deep learning algorithms constitute a class of machine learning algorithms in which a hierarchy of layers is used to transform the data given as input, in a more abstract and composite way. The strength of DL lies in the ability of these models to discover useful feature representations by themselves, needing a reduced amount of hand-tuning of the parameters. This is achieved thanks to the succession of different layers, each applying a non-linear transformation to its input data.
A DL model can be seen as a function $f = f(x;\theta)$ parametrized with $\theta \in \mathbb{R}^{n_\theta}$ ($n_\theta \in \mathbb{N}$). Let us consider now the simple neural network (NN) shown in Fig. 2.3. This network is composed

of three layers. The first one (l_x) constitutes the input of the network, composed of a column vector of size n_x ($n_x \in \mathbb{N}$). The second layer (l_h) transforms this input by a non-linear parametric function, as shown in Eq. (2.3).

$$\mathbf{h} = A(\mathbf{W}^{l_h} \cdot \mathbf{x} + \mathbf{b}^{l_h}) \tag{2.3}$$

Here, $\mathbf{W}^{l_h} = (w_{jk}^{l_h})$ represents a matrix of dimensions $n_h \times n_x$ ($n_h \in \mathbb{N}$) containing the weight between the precedent layer l_x and the current layer l_h. In particular, $w_{jk}^{l_h}$ is the weight between the $k-th$ node of l_x and the $j-th$ node of l_h. $\mathbf{b}^{l_h}$ is a bias term of dimensions n_h and A is the *activation function*. This function makes the transformation at each layer non-linear. Examples of the main activation functions are shown in Fig. 2.4.
Several subsequent hidden layers can follow the hidden layer l_h, each applying a transformation similar to the one in Eq. (2.3). The last layer (l_y), returns the output values y of size n_y:

$$\mathbf{y} = (\mathbf{W}^{l_y} \cdot \mathbf{h} + \mathbf{b}^{l_y}) \tag{2.4}$$

where $\mathbf{W}^{l_y} = (w_{jk}^{l_y})$ is a matrix of size $n_y \times n_h$ and $\mathbf{b}^{l_y}$ is of size n_y ($n_y \in \mathbb{N}$).
These layers are trained to minimize an error:

$$I[f] = \frac{1}{n} \sum_{i=1}^{n} L(y_i^*, f(x_i)), \tag{2.5}$$

where n is the number of data points and $L(\cdot,\cdot)$ represents the loss function. This function is one of the crucial components of ML algorithms and quantifies the difference between the predicted outputs y and the actual target values y^*. Based on the application, the choice of various loss functions is possible. Among the most common loss functions, the mean square error (MSE) and the cross-entropy loss can be mentioned. The former measures the average

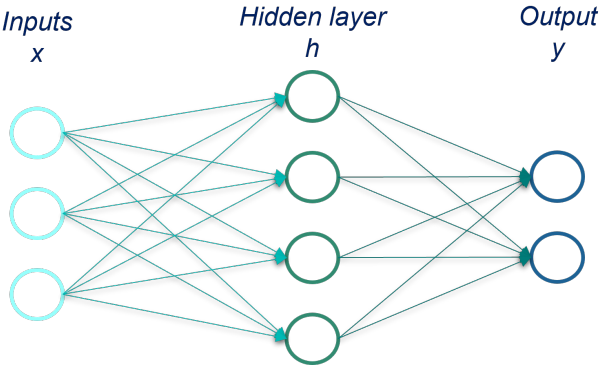

Fig. 2.3 Fully connected NN representation, composed of one input layer, one output layer and one fully connected hidden layer.

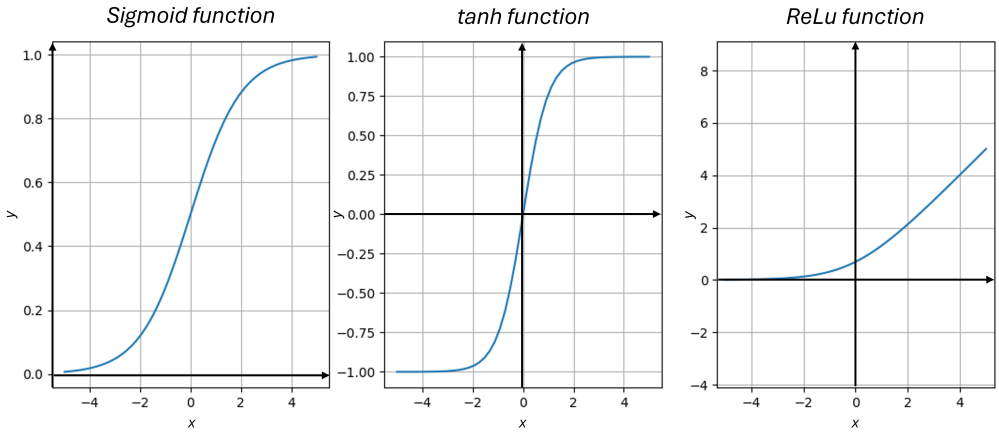

Fig. 2.4 Activation function examples. The sigmoid function can be observed on the left, the tanh function in the middle and the ReLu function on the right.

of the squares of the errors, while the latter measures the difference between two probability distributions.

To train the network, *i.e.,* to find the weights $\mathbf{W}^l = (w^l_{jk})$ and the biases $\mathbf{b}^l$, the most common technique is to compute the gradient of the loss function $L(\cdot,\cdot)$ with respect to the weights and the biases for a single input-output sample. This gradient is computed backward starting from the last layer to avoid the redundant calculation of intermediate terms and increase the efficiency. The whole process takes the name of *backpropagation (Rumelhart et al. (1986))* and is an application of the chain rule derivative to NNs.

2.2.2 Reinforcement Learning

The reinforcement learning problem is usually formulated as a Markov Decision Process (MDP) *(Bellman (1957))*, *i.e.* as a tuple $(\mathscr{S},\mathscr{A},P,r,\gamma)$ in which:

- $\mathscr{S}$ is a set of states;
- $\mathscr{A}$ is a set of actions;
- $P(s_{t+1}|s_t,a_t)$ is the probability density of the next state $s_{t+1}\in\mathscr{S}$ given the current state $s_t\in\mathscr{S}$ and action $a_t\in\mathscr{A}$;
- $r(s_t,a_t,s_{t+1})\in\mathscr{R}$ is the reward obtained by the agent for this transition. $\mathscr{R}$ is a continuous set of possible rewards in a range $R_{max}\in\mathbb{R}^+$;
- $\gamma\in[0,1)$ is the discount factor and it is a measure of how important future rewards are to the current state.

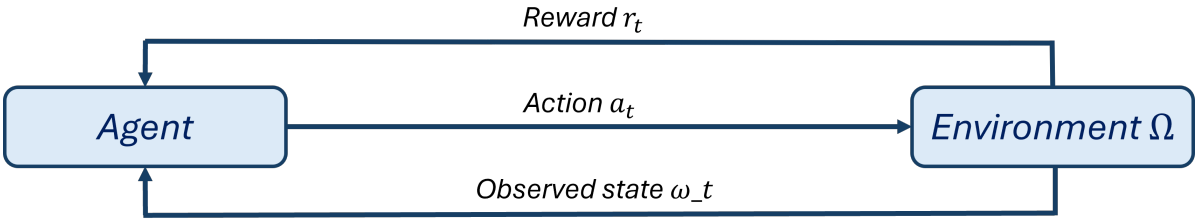

Fig. 2.5 Reinforcement Learning setting. At each time step, an agent takes an action $a_t \in \mathscr{A}$ that modifies the environment Ω. As a consequence of the action taken, the agent receives a reward $r(s_t, a_t, s_{t+1})$, observes a new state $\omega_{t+1} \in \Omega$ and the state transitions from s_t to s_{t+1}.

The general RL problem is formalized as a discrete time stochastic control process, in which an agent interacts with an environment Ω (Fig. 2.5). The agent starts from a state $s_0 \in \mathscr{S}$ and interacts with the environment at each time step through an action $a_t \in \mathscr{A}$. As a consequence, the system transitions from the state s_t to the state s_{t+1} and the agent receives an updated observation $\omega_t \in \Omega$ and a reward $r(s_t, a_t, s_{t+1}) \in \mathscr{R}$ that evaluates the chosen action *(Bellman (1957))*. This theoretical framework requires the control process to be *Markovian*, *i.e.,* the future state must only depend on the current observation, so that the agent has no interest in looking at the full story: $P[st+1|s_t] = P[st+1|s_1, ..., s_t]$. This implies that the observation space must be *fully observable*, which means that the observation space must coincide with the environment ($\omega_t = s_t$). In some scenarios, such as in the case of autonomous driving, the agent can only partially observe the environment, meaning that not all the features of the current state are available to the agent ($\omega_t \subseteq s_t$). In this case, the Markovian property is no longer respected and at each time step the agent receives an observation that does not allow to identify the state with certainty. The formal frame of this problem takes the name of *partially observable Markov decision process* (POMDP) and the 5-element tuple describing the classical MDP is expanded with two more elements: $(\mathscr{S}, \mathscr{A}, P, r, \Omega, O, \gamma)$. Here $\Omega = \{1, ..., N_\Omega\}$ represents a finite set of observations and $O : \mathscr{S} \times \Omega \to [0,1]$ defines a set of conditional observation probabilities. The main difference with respect to the classical formulation of the MDP is that in this case the observation $\omega_t \in \Omega$ received by the agent at each time step depends on the state of the environment with a probability $O(s_t, \omega_t)$.

At each time step, the action the agent chooses is based on a policy π, which maps the perceived state of the environment to actions. The policy is the real core of an RL agent and policies can be generally divided into 2 categories:

- *Deterministic*, in which the policy is described by $\pi(s) : \mathscr{S} \to \mathscr{A}$
- *Stochastic*, in which the policy is described as $\pi(s,a) : \mathscr{S} \times \mathscr{A} \to [0,1]$, where $\pi(s,a)$ denotes the probability that, being in the state s, the action a will be chosen

The aim of a RL algorithm is to maximize the total return $R(\tau)$ defined as:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t, s_{t+1}) \tag{2.6}$$

To do this, two main approaches are available:

- The value-based methods, in which the agent learns a value function, representing the expected cumulative reward from any given state
- The policy-based methods, in which the agent learns directly the optimal policy leading to maximizing the reward

The value-based methods

In this case, to generate the optimal policy $\pi^*(a,s)$, the RL agent needs to learn a quantitative evaluation of a state-action value. To this aim, two functions are available, the state-value function (or V-value) and the action-value function (or Q-value). The V-value function ($V^{\pi}(s)$) evaluates the expected return when following the policy π when being in the state s:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, \pi\right], \tag{2.7}$$

in which $\mathbb{E}$ represents the expected value operator. In this case, the optimal expected return $V^*(s)$ can be defined as:

$$V^*(s) = \underset{\pi \in \Pi}{max}\, V^{\pi}(s) \tag{2.8}$$

This represents the expected discounted reward if the agent in the state s follows the policy π^* thereafter.

The Q-value function, or action-value function ($Q^{\pi}(s,a)$) evaluates the expected return when following the policy π being in a state s and taking the action a:

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a, \pi\right] \tag{2.9}$$

Using Bellman equation, and indicating a possible next state as s', Eq. (2.9) can be rewritten as:

$$Q^{\pi}(s,a) = \sum_{s' \in \mathscr{S}} P(s,a,s')(R(s,a,s') + \gamma Q^{\pi}(s', a = \pi(s'))) \tag{2.10}$$

Here P represents the probability.
The optimal Q-value function $Q^*(s,a)$ can be defined as:

$$Q^*(s,a) = \underset{\pi \in \Pi}{max}\, Q^{\pi}(s,a) \tag{2.11}$$

This represents the expected discounted reward if the agent, given the state s and action a, follows the policy π^* thereafter. The main advantage of the Q-value function is that it permits to obtain directly the optimal policy $\pi^*(s,a)$, starting from $Q^*(s,a)$:

$$\pi^*(s,a) = \underset{a \in \mathscr{A}}{argmax}\, Q^*(s,a) \tag{2.12}$$

The Q-value and V-value functions are related by the advantage function, which describes how good the action a is compared to the expected return when following the policy π. The advantage function is defined as:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{2.13}$$

One of the simplest and most popular value-based algorithms is the Q-leaning algorithm *(Watkins (1989))*, which learns the value of an action in a particular state and builds a table that relates each action with the corresponding state.

The policy-based methods

Unlike the value-based method, the policy-based method aims at directly adjusting the policy based on a performance objective, which is usually the expected return. The optimal policy $\pi^*(s,a) \in \Pi$ can be expressed as *(Sutton et al. (2000))*:

$$\pi^*(s,a) = \underset{\pi \in \Pi}{argmax}\, \mathbb{E}[R(\tau)|\pi] \tag{2.14}$$

Among the many methods, one of the most known and used is the policy gradient method. The objective function can be defined as:

$$\mathscr{J}(\pi_\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \sum_{t=0}^{T} \gamma^t r_t(s_t, a_t, s_{t+1}) \tag{2.15}$$

Here, the policy is parametrized by θ and τ represents a trajectory. The goal here is to maximize this objective function depending on θ using gradient ascent, which allows to take a step toward

the ascending direction:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta \left(\mathscr{J}(\pi\theta_k) \right) \tag{2.16}$$

Considering a stochastic policy π, starting from the state s_0, the expected return from Eq. (2.7) can be written as:

$$V^\pi(s_0) = \int_{\mathscr{S}} \rho^\pi(s) \int_{\mathscr{A}} \pi(s,a) R'(s,a) da ds \tag{2.17}$$

where $R'(s,a) = \int_{s' \in \mathscr{S}} P(s,a,s') R(s,a,s')$ and $\rho^\pi(s)$ is the discounted state distribution given by $\rho^\pi(s) = \sum_{t=0}^{\infty} \gamma^t Pr\{s_t = s | s_0, \pi\}$.
Considering a differentiable policy π_θ, the policy gradient algorithm can be obtained *(Sutton et al. (2000))*:

$$\nabla_\theta V^{\pi_\theta}(s_0) = \int_{\mathscr{S}} \rho^{\pi_\theta}(s) \int_{\mathscr{A}} \nabla_\theta \pi_\theta(s,a) Q^{\pi_\theta}(s,a) da ds \tag{2.18}$$

This equation permits to adjust from the experience the policy parameters θ : $\Delta\theta \propto \nabla_\theta V^{\pi_\theta}(s_0)$. The simplest way to estimate $\nabla_\theta V^{\pi_\theta}(s_0)$ is to use a score function gradient estimator, known as REINFORCE *(Williams (1992))*. To derive a general method of estimating gradients from expectations, the likelihood ratio trick can be used:

$$\nabla_\theta \pi_\theta(s,a) = \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} = \pi_\theta(s,a) \nabla_\theta \log(\pi_\theta(s,a)) \tag{2.19}$$

Thanks to Eq. (2.19), Eq. (2.18) can be rewritten as:

$$\nabla_\theta V^{\pi_\theta}(s_0) = \mathbb{E}_{s \sim \rho^{\pi_\theta}, a \sim \pi_\theta} [\nabla_\theta \log(\pi_\theta(s,a)) Q^{\pi_\theta}(s,a)]. \tag{2.20}$$

If a deterministic policy $\pi(s) : \mathscr{S} \rightarrow \mathscr{A}$ is considered, the policy can be iteratively computed:

$$\pi_{k+1}(s) = \underset{a \in \mathscr{A}}{argmax}\, Q^{\pi_k}(s,a) \tag{2.21}$$

If the policy is differentiable, a simple and not computationally expensive option is to follow the policy in the direction of the gradient of Q:

$$\nabla_\theta V^{\pi_\theta}(s_0) = \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\nabla_\theta(\pi_\theta) \nabla_a (Q^{\pi_\theta}(s,a))|_{a=\pi_\theta(s)}] \tag{2.22}$$

This takes the name of Deterministic Policy Gradient (DPG) *(Silver et al. (2014))*.

For both the deterministic and the stochastic policy gradient, a value function is required to evaluate the current policy. A very interesting alternative approach was proposed by *(Konda and Tsitsiklis (1999))*, who proposed an actor-critic architecture, in which the actor represents the policy, while the critic the value function. In particular, the critic estimates the current

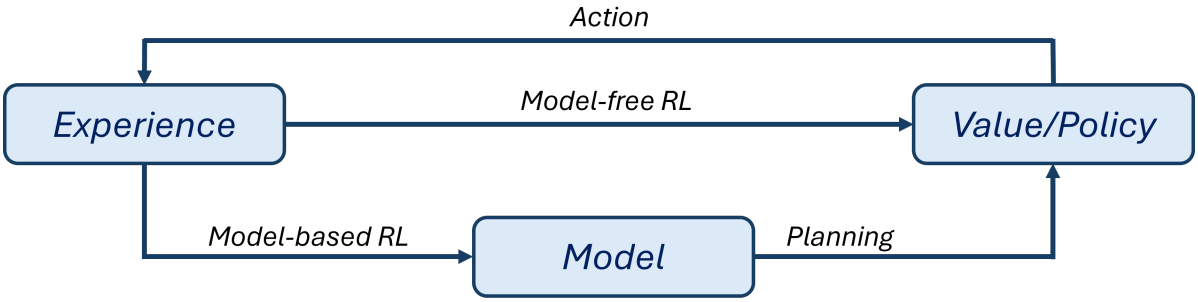

Fig. 2.6 Representation of model-free and model-based RL. In model-free RL, the policy is directly learned from the experience and a representation of either a value function or a policy is used to act on the environment. In model-based RL, a model of the environment is created and a policy is used to guide the actor in this environment.

action-value function for the policy $\pi : Q(s,a;\theta) \approx Q^{\pi}(s,a)$. Given a set of tuples $< s,a,r,s' >$, the current value $Q(s,a,\theta)$ is updated at every iteration towards the target value:

$$Y_k^Q = r + \gamma Q(s', a = \pi(s');\theta) \tag{2.23}$$

This method is straightforward to implement, but it uses a pure bootstrap strategy, as it uses its own value estimates of the following time steps to update the Q-function. This can lead to important computational times and instabilities. To solve this problem, methods that make use of a replay memory have been implemented *(Precup et al. (2000))*. On the other hand, the actor updates the policy by using the policy gradient theorem to adjust the parameters θ. For the stochastic case, the gradient is obtained from Eq. (2.20), with the sole difference that a behavior policy π_β different from π is used, which makes the gradient usually biased:

$$\nabla_\theta V^{\pi_\theta}(s_0) = \mathbb{E}_{s\sim\rho^{\pi_\beta},a\sim\pi_\beta}[\nabla_\theta \log(\pi_\theta(s,a))Q^{\pi_\theta}(s,a)]. \tag{2.24}$$

Both value-based and policy-based methods fall under the category of *model-free approaches*. Another important category of RL algorithms includes *model-based algorithms*. These algorithms rely on a model of the environment, of its dynamic, and of the reward function. The model is either explicitly given or learned from experience. When the model is available, the planning phase involves interacting with the environment to choose the proper action. Fig. 2.6 shows the main differences between a model-free and a model-based algorithm: while a model-free algorithm tries directly to create a representation of the policy or the value function that allows maximizing the reward, the model-based approach involves the extra step of building a representation of the environment with which the agent interacts. This representation

allows model-based RL algorithms to be faster than model-free algorithms in learning from experience, but when making decisions they can be slower because before taking an action, they always require an extra step to evaluate the outcome of their decision using their internal model. Furthermore, when dealing with real-world scenarios, obtaining an accurate model can be extremely challenging. Since model-based RL relies on these models to make decisions, differences between the model and the real world can lead to poor decision-making. Since model-free algorithms do not rely on this representation, they are more robust to the complexities of the real world. Accurately modeling the physics of surgical tools in contact with the vessel walls can be very challenging and a model-free approach has therefore been adopted in this work, to ensure both an increased robustness to the complexity of the environment and a faster decision-making process. For this reason, the model-based approach will not be discussed further. For more details please refer to *(François-Lavet et al. (2018))*.

Besides this classification, RL algorithms can be further categorized based on the availability of data (*online* vs *offline* algorithms) and on the policy used to generate data (*on-policy* vs *off-policy* methods). Regarding the first category, the algorithm is said to be *offline* if the agent learns from a limited amount of data already available in a given environment. In this case, the agent cannot interact further with the environment. If on the other hand, the problem setup involves the collection of data during the learning process, then the problem is classified as *online RL*. This setup is more complicated and the generalization capabilities of the algorithm are influenced by the *exploration/exploitation trade-off*. This is one of the key challenges of RL, implying the need to balance two opposite strategies. On one hand, exploitation involves using the best option based on the current knowledge of the system, while exploration involves the investigation of new options, missing the opportunity to exploit the current policy. The former permits capitalizing on the knowledge already gained, with the risk of having a partial understanding of the environment, while the latter can let the agent discover new features and optimal behaviors. A too-high exploitation can lead to a sub-optimal policy, while a too-high exploration can lead to very long training times. It is therefore important to correctly balance these two strategies to obtain the optimal policy in competitive training times.
Concerning the second categorization of RL algorithms, an algorithm is classified as on-policy if it attempts to improve or evaluate the policy that is used to make decisions, while an off-policy algorithm evaluates or improves a policy different from the one used to generate data *(Sutton and Barto (2018))*. An on-policy algorithm usually introduces a bias when it is used with samples taken from a replay memory, while an off-policy algorithm allows directly reusing these samples, making this last category more sample efficient.

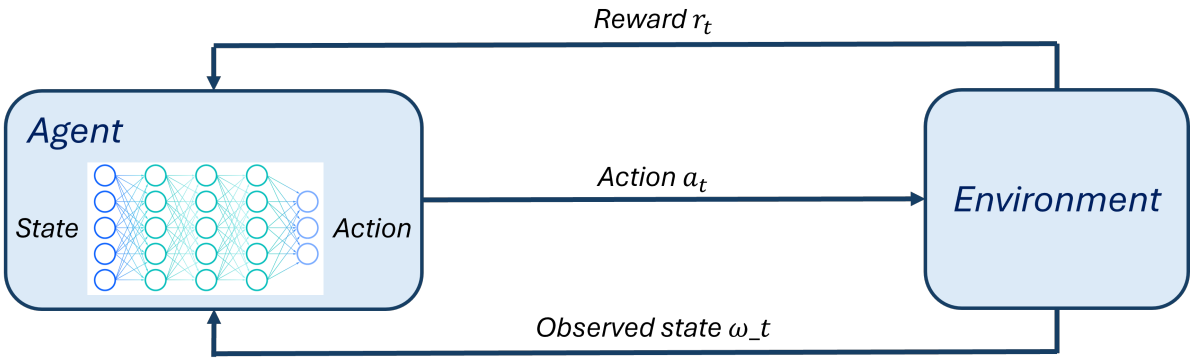

Fig. 2.7 Representation of DRL problem. In this case, the agent is substituted by a NN which, given the state as input, gives as output the action a_t. This action acts on the environment and triggers the transaction from the state s_t to the state s_{t+1}. As a consequence of this transaction, the agent receives a reward r_t and an updated observation of the environment ω_t.

2.2.3 Deep Reinforcement Learning

When dealing with real-world problems, the state space becomes high-dimensional and too difficult to solve with classical RL techniques. In this case, it is useful to model the agent by means of a NN, as shown in Fig. 2.7: in this scenario, the action to take (policy-based) or the value of a state (value-based) are estimated by means of a deep NN. In the case of the actor-critic algorithms, for example, both components are substituted by a NN. As a result, the advantages of DL are fused with the potentialities of RL algorithms: thanks to the introduction of a NN, an increase in the dimension of the action and state space does not require an exponential increase of input data anymore. The higher dimension of these spaces is indeed automatically handled by the neural network. The input can therefore consist of images acquired by a camera or a stream of data coming from a sensor. These data are elaborated by the neural network, which is able to find a relation that allows deciding which action to perform to optimize the objective, which can be for example the maximization of the score while playing a game. Furthermore, the NN can be trained incrementally, allowing the use of additional samples as the learning proceeds.

During the training, the process is the same as the one followed in the classical reinforcement learning process: at the time step t, an agent interacts with the environment by means of an action a_t, and receives as a response a new observation ω_t, which describes the new state of the environment after performing the action a_t, and a reward r_t to evaluate the taken decision. This information is then used by the NN constituting the agent to update its weights and converge towards the optimal behavior that allows to maximize the reward.

2.3 Deep Reinforcement Learning and Optimal Control

Two complementary approaches to solving the same problem

Deep Reinforcement Learning and optimal control strategy present many common points and are used to solve similar problems, under slightly different conditions. The first important common point is the field of application of these two theories, which deal with dynamic systems. To attain their objective, both methods aim to minimize a certain cost function, which is defined based on the specific application and depends on the action taken and the current state of the system. This cost function takes however two different configurations. DRL algorithms aim at maximizing the value function (Eq. (2.7) and Eq. (2.9)) and minimize the cost for any potential state, finding the optimal policy π^*. This policy can then be followed in all the possible states. In the optimal control framework, the aim is only to minimize the cost given some initial conditions. This process results in a sequence of actions to take, which can change as the initial conditions change. Another important point to take into account is that in the model-free DRL setting, no assumption is made on the state of the system and the direct aim is the computation of the optimal policy π^* which maximizes the reward. Optimal control theories, on the other hand, exploit the prior knowledge of the dynamic behavior of the system and incorporate it as a constraint.

Both these strategies have their advantages and disadvantages, which make them adapted for certain applications. To apply an optimal control strategy, it is necessary to model the dynamic of the studied system. The precision of the obtained controller will be strongly linked to the accuracy of this model. Obtaining the model of real-world systems is usually quite complex. Even when this model exists, it sometimes requires linearization, which introduces errors. However, if a model with acceptable accuracy exists, it becomes possible to switch between different costs, making it easy to change the objective of the control. Furthermore, even if the system deviates significantly from the operating conditions for which the controller was designed, an optimal control strategy can still provide a reliable foundation for finding the best solution, thanks to the model guiding the decision-making process.

DRL learning techniques on the other side, do not require a model of the dynamic of the studied system, making it possible to find the best control strategy even when a model cannot be obtained. This makes these techniques more robust to randomness and more adapted to deal with action constraints. The limitation of this method is that when the agent finds itself in a state that significantly deviates from the states experienced during the training, the policy can generate random decisions, leading to undesired and uncontrollable behavior. Another major drawback of this strategy is that a change in the objective requires restarting the training of the network from scratch and usually also requires modifying the engineered reward function.

When dealing with the navigation of catheters and guidewires inside the anatomy of the patient, developing a model able to reliably take into account the physics of these flexible devices in contact with the vessel wall and using this model to derive the cost function of the optimal control problem can be extremely challenging and lead to linearization errors which could lead to a wrong controller behavior. Furthermore, real-world environments require the development of a strategy that is as robust as possible to the uncertainties of the environment. Given the good capability of DRL controllers to deal with uncertainties and with problems difficult to model, this category has been chosen to design the controller developed in this work. This choice was also supported by the numerous works that adopt DRL techniques to develop autonomous controllers for endovascular navigation (see Section 6.1).

Chapter 3

The Simulated Environment

One of the fundamental aspects of this work is to accurately model the navigation of the devices inside the vessels. This is an essential requirement for both the systems developed in this work. When creating enhanced fluoroscopic images, our aim is to show the clinician the shape and the position of the devices in conditions different from the current ones. Without a reliable model of the surgical instruments, this system cannot be developed. Considering the autonomous navigation system, our aim is to develop our controller in a simulated environment and then transfer the learned behavior to the real world. As explained in Section 2.3, we designed a DRL-based controller, which requires an accurate training to learn the desired task. This requires several navigation episodes, in which the agent guides the devices from the insertion point to the target location and interacts with the environment. Performing these experiments in the real world can be very consuming both in terms of time and in terms of resources, mainly in the first steps of the learning process in which the agent only takes random actions. As a solution, we decided to perform the training of our algorithm in a simulated environment, which allows to reduce both the training time and the costs. For this to work and be transferable to the real world, the simulated environment must reproduce the physics of the real problem as accurately as possible. For this reason, at the start of the project, we focused our attention on the creation of this environment. Besides being able to correctly model the physics of the instrument, the simulation must also be able to handle the contacts between the devices and the wall of the vessel, and the devices themselves. An additional requirement to speed up the training process and allow the production of enhanced fluoroscopic images is to maintain real-time performances.

In this Chapter, details regarding the simulated environment and the choices made in this work will be provided. First, an introduction to the finite element method (FEM) is supplied in Section 3.1, which provides the mathematical tools used to model physical problems. The model adopted to simulate the physics of the devices and their discretization are then discussed

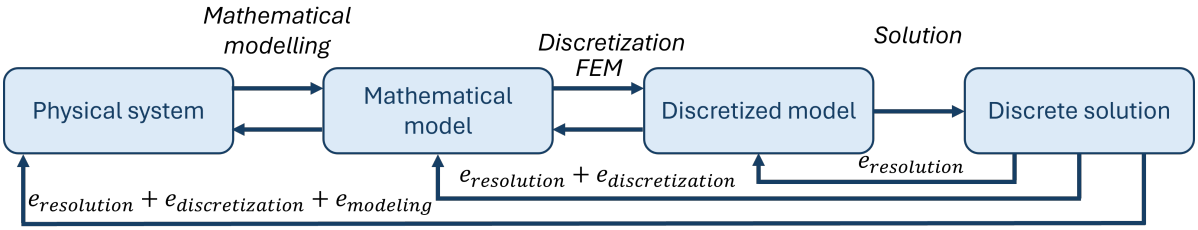

Fig. 3.1 Schematization of process followed to model a physical problem. The first step of the process involves the identification of a suitable mathematical model to simulate the behavior of the system. This mathematical model is then discretized, and starting from this discretized system, the discrete solution of the initial problem is computed. Each of these phases introduces an error: the model introduces differences between the real phenomenon and its representation ($e_{modeling}$), the discretization introduces an error between the continuous and the discrete system ($e_{discretization}$) and the computation of the solution introduces solution errors ($e_{resolution}$).

in Section 3.2.1 and Section 3.2.2 respectively. Finally, Section 3.2.3 describes how the contacts between the different simulated objects are handled in this work.

3.1 Model of physical problems

When the model of a physical phenomenon is necessary, the first fundamental step is identifying a suitable mathematical model to simulate and predict aspects of the system's behavior. This model is an abstraction of the physical reality and represents a tool to reduce the complexity of the original physical system, filtering out the information that is not relevant to the purpose of the study. Complex systems are usually represented by partial differential equations (PDE), which are equations containing an unknown function of two or more variables and their partial derivatives with respect to these variables. Analytically solving these problems is often impossible and this is why, to evaluate these equations, several techniques have been developed, such as the boundary element method, the meshfree method, the finite difference method (FDM), or the finite element method (FEM). This latter in particular offers several advantages that led to its use in a wide range of applications. This method indeed presents great flexibility, allowing the solution of problems involving complex geometries with complex boundary conditions and offers a solid theoretical foundation. FEM theory was developed in the mid-1990s and was originally conceived to solve complex elasticity and structural analysis systems. Today it is largely employed in all engineering fields, ranging from the automotive industry to the biomedical research. The core of this method lies in the discretization of a complex continuum domain in subdomains, called mesh, allowing to transform the solution of a complex PDE in the solution of a system of algebraic equations. A scheme summarizing

this whole process is shown in Fig. 3.1. Another very interesting method is constituted by the FDM, which class of numerical techniques to solve differential equations by approximating derivatives with finite differences. The great advantage of this method is the discretization of the domain is simple and regular subdomains.
The solution of a PDE involves two main steps:

- *The discretization of the spatial domain*, which is solved by means of the FEM, which allows to consider a problem with complex geometries. This leads to the composition of a system of linear or non-linear algebraic equations (Section 3.1.1).
- *The discretization of the temporal domain*, to solve the eventual dynamic of the problem. Since the discretization of the time domain is Cartesian, *i.e.* divided into a regular grid, this is solved using the FDM (Section 3.1.2)

3.1.1 The Finite Element Method

The computation of the solution of a PDE through the application of the FEM involves first of all the computation of the weak formulation of the problem, which consists in the calculation of an integral form of the PDE in which equations and conditions are no longer required to hold absolutely *(Ern and Guermond (2004))*. To obtain the weak formulation, the PDE is multiplied by a test function v, which is a smooth function of compact support, and integrated over the domain. Applying then Fubini's theorem, which allows the interchange of the order of integration, and the integration by parts, it is possible to transfer all the partial derivatives from the variable of the PDE to the test function v. The new formulation of the problem includes this way solutions that are not necessarily differentiable *(Evans (1998))*. The process that allows to pass from the strong formulation to the weak formulation is illustrated in the following in the case of a simple 1D problem, but the same principles can be applied to more complex problems. Let's consider the 1D Poisson equation:

$$\begin{cases} -u''(x) = f(x) \quad \text{in } (0,1) \\ u(0) = u(1) = 0, \end{cases} \tag{3.1}$$

where f is given, u is an unknown function in x, and u'' is the second derivative of u with respect to x. To obtain the weak formulation, Eq. 3.1 is multiplied by the smooth function v, which satisfies the boundary conditions ($v(0) = v(1) = 0$) and integrated over the domain:

$$-\int_0^1 u''(x)v(x)dx = \int_0^1 f(x)v(x)dx \tag{3.2}$$

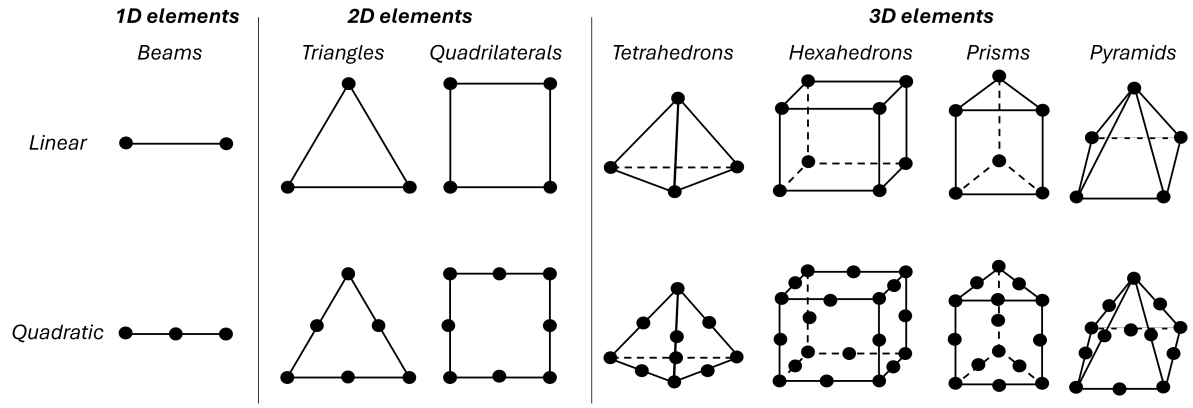

Fig. 3.2 Illustration of the different possible element types.

Applying the integration by parts, the partial derivatives are all transferred to v:

$$\begin{aligned} \underbrace{\int_0^1 f(x)v(x)dx}_{F(v)} &= -\int_0^1 u''(x)v(x)dx = \\ = -[u'(x)v(x)]_0^1 + \int_0^1 u'(x)v'(x)dx &= \underbrace{\int_0^1 u'(x)v'(x)dx}_{a(u,v)} \end{aligned} \tag{3.3}$$

in which $v(0) = v(1) = 0$ was used.
The same reasoning can be applied to the 2D case, as shown here below with the generalization of the Poisson problem to the 2D case:

$$\begin{cases} -\Delta u = f & in\ \Omega \subset \mathbb{R}^2 \\ u = 0 & in\ \delta\Omega, \end{cases} \tag{3.4}$$

Defining now a test function $v \in \mathbb{H}_0^1(\Omega)$, multiplying it by the two terms in Eq. (3.4) and integrating over the domain, the following expression is obtained:

$$\int_\Omega -\Delta u(x)v(x)dx = \int_\Omega f(x)v(x)dx, \qquad \forall v \in \mathbb{H}_0^1(\Omega) \tag{3.5}$$

Integrating by part, Eq. (3.5) becomes:

$$\underbrace{\int_\Omega -\nabla u(x)\nabla v(x)dx}_{a(u,v)} = \underbrace{\int_\Omega f(x)v(x)dx}_{F(v)}, \qquad \forall v \in \mathbb{H}_0^1(\Omega) \tag{3.6}$$

After writing the weak formulation of the problem, the next step is discretizing the domain, which transforms a problem with an infinite number of possible solutions into one with a finite number of unknowns or *degrees of freedom*. This means that instead of the infinite-dimension problem:

Find $u \in$ *to the Hilbert space* $\boldsymbol{H}_0^1 \mid \forall v \in \boldsymbol{H}_0^1, a(u,v) = F(v)$,

the following finite-dimension problem is solved:

Find $u \in V \mid \forall v \in V, a(u,v) = F(v)$, *where* V *is a finite-dimension subspace of* $\boldsymbol{H}_0^1$

Based on the nature of the problem and the shape of the solution space, different geometrical element types can be used to spatially discretize the domain, such as beams for 1D spaces, triangles and quadrilaterals for 2D spaces, and tetrahedra, hexahedra, prisms and pyramids for 3D spaces (Fig. 3.2). The ensemble of all the geometrical elements composes the mesh. The vertexes of the elements take the name of *nodes*, while the line connecting two vertexes takes the name of *edge*.

Usually the basis of the discretized solution space V can be written as $\{\phi\}_{i=1}^N$ and any function $v_k \in V$ can be seen as $v_k = \sum_{j=1}^N \alpha_j \phi_j$. This means that the approximate solution $u_h \in V$ to the original PDE can be obtained as $u_h = \sum_{j=1}^N u_j \phi_j$.

Restricting the weak formulation to the finite-element subspace, $a(u,v) = F(v)$ becomes:

$$a(u_h, v_h) = F(v_h) \tag{3.7}$$

Substituting $u_h = \sum_{j=1}^N u_j \phi_j$ and $v_h = \phi_i,\ i \in 1,..,N$, Eq. (3.7) becomes:

$$a(\sum_{j=1}^{N} u_j \phi_j,\ \phi_i) = F(\phi_i) \tag{3.8}$$

Calling $A_{ij} = a(\phi_j,\ \phi_i)$, $F_i = F(\phi_i)$ and $U = [u_1, u_2, ..., u_N]^T$, the discrete weak formulation (3.7) becomes a system of linear equations:

$$\mathbf{A}U = \mathbf{F} \tag{3.9}$$

When the test function used to derive the weak formulation and the basis function used in the domain discretization are the same as in the described case, the approach takes the name of *Galerkin Method*.

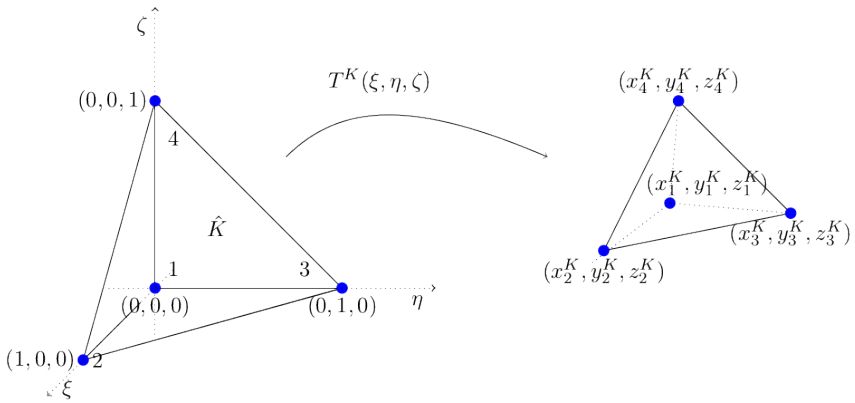

Fig. 3.3 Reference (left) and deformed (right) configuration of a tetrahedral element.

The basis functions, which also take the name of *shape function,* are the core of the FEM. These functions are usually chosen so that:

$$\phi_i(x_j) = \begin{cases} 1 & if\ x_i = x_j, \\ 0 & if\ x_i \neq x_j, \end{cases} \tag{3.10}$$

and so that the support of ϕ_i is only on cells adjacent to x_i. This particular choice simplifies the calculations, introducing several zero-terms in the matrix $\mathbf{A}$. When a basis function respects these properties, it takes the name of *Lagrangian basis function.* With the discretization of the solution domain, the solution is only computed on the nodes of the mesh, and to interpolate the solution between these discrete values, the shape functions are used. These shape functions are polynomial approximations of first, second or third degrees. The degrees of freedom are represented by the black nodes in Fig. 3.2.
Another point that it is important to keep in mind is that the shape functions ϕ can always be expressed in the coordinates of an undeformed configuration of the element (ξ,η,ζ) (Fig. 3.3), regardless of the deformation that the element has undergone. This means that it always exists a transformation T^K from a reference element $\hat{K}$ to the frame of an element K such that:

$$T^K : \hat{K} \to K, \qquad \hat{\xi} = (\xi,\eta,\zeta) \to x = T^K(\hat{\xi}) = \sum_{i=0}^{N} x_i \phi_i(\hat{\xi}) \tag{3.11}$$

T^K is bijective if $det(J) \neq 0$, where $det(J)$ is the determinant of the Jacobian of the transformation. This transformation will allow for the integration of the weak formulation by change of variable, permitting the integration on the reference element domain $d\xi, d\eta, d\zeta$ instead of on $d\Omega = dxdydz$. Furthermore, in the case of quadratic shape functions, some implementations

offer the possibility to geometrically discretize the space with quadratic cells, which are characterized by the possibility of having curved edges, thanks to the additional nodes present along the edges (second raw of Fig. 3.2). This differentiates them from the first-order elements, which can only have straight edges, and allows a better representation of certain types of phenomena.

3.1.2 The temporal discretization

Following the process explained in the previous paragraph, the initial partial differential equation is decomposed into a set of equations and assembled in the system shown in Eq. (3.9). When studying a dynamic system, this system will be composed of *ordinary differential equations* (ODE). To solve these equations, the adopted numerical method takes the name of *integration scheme (Butcher (2016))*. The goal of the method is to find an approximate solution to the problem:

$$\frac{dy}{dt} = f(t, y(t)) \tag{3.12}$$

The first step involves the discretization of the time domain of the ODE in small time steps, called dt. The solution to Eq. (3.12) will then describe how to go from the current time step t to the following time step $t + dt$.

All the integration schemes can be divided into two main categories:

- **Explicit schemes**, such as the Euler explicit scheme or the Runge-Kutta scheme. In the case of the Euler explicit scheme, the solution at the time step $t + dt$ is computed only based on information from the previous time step t:

$$y(t + dt) = y(t) + dt \cdot f(y(t)) \tag{3.13}$$

- **Implicit schemes**, such as the Euler implicit scheme or the Newmark Implicit scheme. In the case of the Euler implicit scheme, the solution at the time step $t + dt$ is computed based on information from the previous time step t and information from $t + dt$ itself:

$$y(t + dt) = y(t) + dt \cdot f(y(t + dt)) \tag{3.14}$$

Explicit schemes usually require only one iteration per time step and are generally easier to implement and less computationally demanding. However, they have the tendency to be unstable, thus requiring small time steps dt to reach the convergence. Implicit schemes on the other hand usually require several iterations per time step, but they ensure a higher stability, thus allowing for the choice of bigger time steps.

The system obtained thanks to the integration scheme, must now be solved by means of a *solver* to find the solution at time $t + dt$. Two different categories of solvers are available:

- **Direct Solvers**, which compute $\mathbf{A}^{-1}b$ in one single step. In this case, it is necessary to compute the inverse of the matrix $\mathbf{A}$, and for this purpose, several methods are present which can be more or less adapted to the specific case based on the structure of $\mathbf{A}$. Among the most employed, Cholesky, LU, or LDL and their sparse versions can be found.
- **Iterative Solvers**, which converge toward the solution gradually. At each iteration, the solution is computed more accurately, meaning that the error decreases with the number of iterations. Among the available methods, the conjugate gradient method and the minimal residual method can be mentioned.

Direct methods are usually able to solve well-conditioned and even some ill-conditioned problems. Iterative methods maintain a monotonic convergence even for large systems if the problem is well-conditioned, but for ill-conditioned problems, the convergence can require much more time.

Each of the steps here described, which allow the transition from a real physical system to a discrete solution, introduces an error, as shown in the scheme in Fig. 3.1. When the system of equations in Eq. (3.9) is solved by applying the integration schemes described in Section 3.1.2, an error, due to the numerical approximation is introduced ($e_{resolution}$). This error is the amount by which the discrete solution fails to satisfy the discretized model. The second source of error is due to the discretization of the mathematical model ($e_{discretization}$) and accounts for the amount by which the discretized model fails to satisfy the mathematical model. This is due to the choice of elements' type and dimension, and to the interpolation (*i.e.,* order of the elements). Finally, the last source of error is due to the differences between the mathematical model and the physical system ($e_{modeling}$): a model indeed can never represent all the aspects of a real system, but can only define a simplified version of it. When modeling the physics of a real system using the FEM it is therefore important to be aware of these sources of error, to try to minimize them as much as possible.

3.1.3 Model of mechanical problems

When applying the FEM to the mechanics of solids, one the most important formulation is the *principle of virtual work* (PVW) *(Hjelmstad (2007))*. This constitutes the starting point of the solution of any system involving deformable solids. The PVW states that:

> *The total virtual work done by all the forces acting on a system in static equilibrium is zero for a set of infinitesimal virtual displacements from equilibrium.*

Let us consider a deformable body Ω in a state of equilibrium and subjected to a traction force t on the boundary $\delta\Omega$ of its domain. This load induces the stress σ inside the domain of the body. Let's define a kinematically admissible virtual displacement field δu inside the body. A kinematically admissible virtual displacement defines a continuous and differentiable field, satisfying the condition $\delta u = 0$ on the boundary $\delta\Omega$. According to the PVW:

$$\underbrace{\int_\Omega \boldsymbol{\sigma}\delta\boldsymbol{\varepsilon}dV}_{\textit{External Virtual Work}} = \underbrace{\int_\Omega b\delta\mathbf{u}dV + \int_{\delta\Omega} t\delta\mathbf{u}dA}_{\textit{Internal Virtual Work}}, \qquad \forall\delta\mathbf{u} \tag{3.15}$$

where $\varepsilon = \frac{\nabla\mathbf{U}+\nabla\mathbf{U}^T}{2}$ is the symmetric part of the gradient of the displacements $\mathbf{U}$ and b represents the body forces. $\mathbf{U}$ contains the displacements of all the nodes of the mesh. This equation states that the stress, body force and traction are in equilibrium if and only if the internal virtual work equals the external virtual work for every virtual displacement field.
Since requiring that the *External Virtual Works = Internal Virtual Works* is a too strong requirement, we will only maintain *External Virtual Works = Internal Virtual Works* for a subset of virtual displacement fields, which leads to an approximate solution of the actual displacement field. In this sense, the PVW is a weak formulation of the equilibrium conditions.

The displacement $\mathbf{u}$ can be expressed as:

$$\mathbf{u} = \mathbf{H}_s\mathbf{U} \tag{3.16}$$

Here, the matrix $\mathbf{H}_s$ contains the shape functions ϕ expressed in the coordinates of an undeformed configuration of the element $(d\xi, d\eta, d\zeta)$ (Section 3.1.1).
The strain $\boldsymbol{\varepsilon}$ can be expressed as:

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{U} \tag{3.17}$$

Here $\mathbf{B}$ is a matrix containing the differential operator, and it depends on $(d\xi, d\eta, d\zeta)$.
Finally, the stress $\boldsymbol{\sigma}$ can be expressed as:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon} + \boldsymbol{\sigma}_0 \tag{3.18}$$

where $\mathbf{C}$ is the stiffness tensor, containing values related to the Young Modulus E and the Poisson ratio ν, and $\boldsymbol{\sigma}_0$ is the initial stress.
Considering all these elements, Eq. (3.9) can be rewritten as:

$$\mathbf{K}\mathbf{U} = \mathbf{R} \tag{3.19}$$

where $\mathbf{K} = \left(\int_\Omega \boldsymbol{\sigma}(\phi_i)dV : \boldsymbol{\varepsilon}(\phi_j)\right)_{ij}$ and $\mathbf{R} = \left(\int_\Omega b\phi_i dV + \int_{\partial\Omega} t\phi_i dA\right)_i$.

3.2 The Simulated Environment

Now that the fundamentals of the FEM have been introduced, it is possible to describe our simulation of endovascular navigation. The environment was implemented by exploiting SOFA framework *(Faure et al. (2012))*, which provides an open-source C++ library that allows the development of interactive real-time simulations. The strength of this software lies in the decomposition of a complex scene into simple components, which, given the open-source nature of the framework, can be modified according to the user's needs. Besides these great advantages, this framework was already used by several previous works developing autonomous systems for guidewire navigation *(Cho et al. (2021); Karstensen et al. (2020, 2023))*. All these elements position SOFA as an excellent candidate for the implementation of our simulated environment.

3.2.1 Catheters and guidewire modeling

As introduced in Chapter 1, catheters and guidewires are long, thin, flexible tube devices. Given these characteristics, it is possible to refer to the literature on the mechanical modeling of beams to simulate their physical behavior. A beam is a structural 1D element that primarily experiences loads applied laterally across the beam's axis, thus mainly undergoing bending. To model these structures, several theories are available, but when describing catheter-like devices, the most common models are the Euler-Bernoulli and the Kirchhoff theory and their generalizations *(Timoshenko and Goodier (1969))*.

Euler-Bernoulli theory provides a way of computing beams' load-carrying and deflection characteristics, through the following relationship:

$$\frac{d^2}{dx^2}\left(EI\frac{d^2y}{dx^2}\right) = q(x) \tag{3.20}$$

Here, $y(x)$ describes the deflection of the beam in the z direction at a point x of the beam, and $q(x)$ is a distributed load. EI represents the flexural rigidity, obtained as the product between the Young Modulus E and the moment of inertia I, which is calculated with respect to the axis perpendicular to the applied load. If the product EI is constant, Eq. (3.20) becomes:

$$EI\frac{d^4y}{dx^4} = q(x) \tag{3.21}$$

The main assumption behind Euler-Bernoulli's theory is that cross-sections of the beam that are plane and perpendicular to the neutral axis of the beam, remain plane and perpendicular

after deformation. This leads to the strong simplification of no shear deformation of the beam. Furthermore, the original theory is only valid for infinitesimal strains and small rotations.

Several variations of this approach have been developed through the years to model more complex cases, and have been designed as *geometrically exact beams* (*Boyer and Primault (2004); Cardona and Geradin (1988); Crisfield (1990)*, etc.).
Among the others, particular interest was raised by the Timoshenko beam theory *(Timoshenko and Goodier (1969))*. This apporach accounts for both the shear deformation and the rotational bending effect and it is particularly suitable to model composite or short thick beams. In this formulation, Euler-Bernoulli assumptions are softened and, while the plane sections are always considered plane, it is not required anymore for them to remain perpendicular to the neutral axis. The governing equations in this case are the following:

$$\begin{aligned} \frac{d^2}{dx^2}\left(EI\frac{d^2\varphi}{dx^2}\right) &= q(x) \\ \frac{dy}{dx} &= \varphi - \frac{1}{cAG}\frac{d}{dx}\left(EI\frac{d\varphi}{dx}\right) \end{aligned} \tag{3.22}$$

Here A is the cross-section area, G is the shear modulus, c is the Timoshenko coefficient and depends on the geometry and φ is the angle of rotation of the normal to the mid-surface of the beam. Combining the two equations in (3.22) and considering a homogeneous beam with constant cross-section, Eq. (3.23) is obtained.

$$EI\frac{d^4y}{dx^4} = q(x) - \frac{EI}{cAG}\frac{d^2q}{dx^2} \tag{3.23}$$

This equation relates the displacement y for the beam, to the mechanical properties of the beam and the load $q(x)$ applied to it.
Following Timoshenko theory, it is possible to express the bending moment M_{xx} and the shear force Q_x as a function of the displacement y and the rotation φ:

$$M_{xx} = -EI\frac{\delta\varphi}{\delta x}, \qquad Q_x = cAG\left(-\varphi + \frac{\delta y}{\delta x}\right) \tag{3.24}$$

For more details, please refer to *Elishakoff (2019)*.

Another interesting approach is provided by the Kirchoff Rod theory. This theory's success is partly attributed to Kirchhoff's discovery that the equations governing static deformations of the rod are analogous to those for the rotational motion of a rigid body *(O'Reilly and O'Reilly (2017))*. This theory leads to a set of equations that are easy to apply and simplify. For example,

Euler's theory of the elastica can be readily found starting from the rod theory *(O'Reilly and O'Reilly (2017))*. Kirchoff's approach makes three main assumptions *(Dill (1992))*:

1. Cross-sections remain plane, undistorted, and orthogonal to the axis of the rod
2. The transverse stress is zero
3. The bending moments and the twisting moment are proportional to the components of curvature and twist of the axis of the rod

Kirchoff's description of the kinematic of a cable with a constant circular cross-section and which is straight in the rest configuration is provided below *(Boyer et al. (2011))*. Let us denote $X \in [0, L]$ the material abscissa along the beam and $(\mathbf{t_i}(t, X))_{i=1,2,3}$ the moving orthonormal frame rigidly attached to the cross-section X. At time $t = 0$, the cable is at rest. Applying the assumption 1, any transformation of the cable can be decomposed in a translation, which changes the position field from $X \to \mathbf{r}(X)$ to $X \to \mathbf{r}^*(X) = \mathbf{r}(X) + \mathbf{d}(X)$, with $\mathbf{d}$ denoting a displacement field, and a rotation changing the orientation field $X \to \mathbf{R}(X)$ into:

$$\mathbf{R}^*(X) = e^{\Xi \hat{\mathbf{t}}_1^*} \Lambda_{t_1}(\mathbf{t}_1^*) \mathbf{R}(X) \tag{3.25}$$

Here, the transformation $X \to \mathbf{t}_1^*(X) = \frac{\delta_X \mathbf{r}^*}{||\delta_X \mathbf{r}^*||}$ is introduced, along with the rotation $\Lambda_{t_1}(\mathbf{t}_1^*)$, compatible with the Kirchhoff assumptions and with the rotation $e^{\Xi \hat{\mathbf{t}}_1^*}$ which denotes the rotation about the vector $\mathbf{t_1}^*$ of an angle Ξ.

Over the years, several variations of the Kirchoff theory have been derived. *Bergou et al. (2008)* developed a discrete elastic rod theory, based on *Langer and Singer (1996)* work, for a discrete geometric model of thin flexible rods with arbitrary cross-section and undeformed configuration. *Meier et al. (2014)* implemented a new finite element formulation of Kirchhoff's theory including the deformation states of axial tension, torsion and bending, allowing to account for large deformations. Among the others, Cosserat's *(Antman (2005))* model raised particular interest for the model of soft robots and surgical tools such as needles and wires *(Boyer and Porez (2015); Boyer et al. (2006); Renda et al. (2018); Rucker and Webster III (2011))*. This approach integrates into Kirchhoff's theory additional degrees of freedom related to the rotation of the rod's cross-sections, allowing the stretching and shear of the beam to be modeled. This approach assumes an incompressible and linear elastic material and a length of the beam which is much larger than the radius. Following Cosserat's model, the configuration of a micro solid of a soft body with respect to the spatial frame, is characterized by the position vector u and an orientation matrix $\mathbf{R}$, which are parametrized by the material abscissa $X \in [0, L]$ along the beam. The space configuration is thus defined as a curve $\mathbf{g}(\cdot) : X \to \mathbf{g}(x) \in SE(3)$,

Table 3.1 Comparison between Timoshenko and Cosserat beam models implemented in SOFA.

Force [N]	***Method***	***Coordinates*** [x,y][m]	***Error*** [x,y][%]
4.05×10^5	*Beliaev et al. (2015)*	[0.9885, -0.1381]	-
	Timoshenko Model	[0.9792, -0.1335]	[0.9, 3.3]
	Cosserat Model	[0.9887, -0.1366]	[0.02, 1.1]
1.093×10^7	*Beliaev et al. (2015)*	[0.4171, -0.8326]	-
	Timoshenko Model	[0.4179, -0.8196]	[0.2, 1.5]
	Cosserat Model	[0.4189, -0.8224]	[0.43, 1.22]
3.281×10^7	*Beliaev et al. (2015)*	[0.2403, -0.9246]	-
	Timoshenko Model	[0.2428, -0.9087]	[1.04, 1.72]
	Cosserat Model	[0.2424, -0.8994]	[0.87, 2.7]
2.953×10^8	*Beliaev et al. (2015)*	[0.07448, -1.170]	-
	Timoshenko Model	[0.0879, -1.1380]	[18.0, 2.7]
	Cosserat Model	[0.0809, -0.9663]	[8.6, 17.4]

where $SE(3)$ is the special Euclidean group of rigid body displacements in three dimensions and $\mathbf{g}$ is:

$$\mathbf{g} = \begin{pmatrix} \mathbf{R} & u \\ \mathbf{0}^T & 1 \end{pmatrix}$$

The strain state of the beam is defined by the vector field along the space configuration $\mathbf{g}(\cdot)$ given by $X \to \check{\xi}(X) = \mathbf{g}^{-1}\frac{\delta \mathbf{g}}{\delta X} = \mathbf{g}^{-1}\mathbf{g}' \in \mathfrak{se}(3)$, with $\mathfrak{se}(3)$ representing the Lie algebra. The components of this field in the body frame are expressed as:

$$\check{\boldsymbol{\xi}} = \begin{pmatrix} \tilde{k} & q \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3), \qquad \boldsymbol{\xi} = (k^T, q^T)^T \in \mathbb{R}^6$$

where $q(X)$ represents the linear strains, and $k(X)$ the angular strains. The tilde describes the isomorphism between 3-D vectors and skew-symmetric matrices.
The time evolution of $g(\cdot)$ is expressed by the twist vector $X \to \eta(X) \in \mathbb{R}^6$, which is defined as $\hat{\eta}(X) = \mathbf{g}^{-1}\frac{\delta \mathbf{g}}{\delta t} = \mathbf{g}^{-1}\dot{\mathbf{g}}$. The components of this field in the body frame are expressed as:

$$\hat{\eta} = \begin{pmatrix} \tilde{w} & v \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3), \qquad \eta = (w^T, v^T)^T \in \mathbf{R}^6$$

where $v(X)$ is the linear velocity and $w(X)$ the angular velocity at a given time instant.
Given these definitions, the strain of the beam ξ is linked to the position $\mathbf{g}$ by:

$$\mathbf{g}' = \mathbf{g}\check{\xi}, \tag{3.26}$$

and to the velocity by:

$$\eta' = \dot{\xi} - \mathrm{ad}_{\xi}\eta \tag{3.27}$$

where ad_{ξ} is the adjoint map of the strain vector $\boldsymbol{\xi}$.
For more details, please refer to *Renda et al. (2018)*, used as source of this mathematical description.

Both Timoshenko and Cosserat's theories are more complex than the basic Euler-Bernoulli formulation, but when the shear deformation component is not negligible, they provide more accurate results. This is why in this manuscript both these formulations have been compared. In particular, the results obtained with both the Timoshenko and the Cosserat models were compared with the results reported in *Beliaev et al. (2015)*. This paper presents the results of a bending test on a 1 *m*-long bar modeled with the Cosserat-Timoshenko approach, which the authors propose as a gold standard for comparing other models. The bar, directed along the x-axis, presents a Young Modulus of 1.96 *MPa*, a diameter of 5 *cm* and a Poisson ratio of 0.28. The bar is fully fixed at one end, while on the other end a force F is applied along the y-direction. The same conditions have been reproduced in a simulated environment developed in SOFA framework, which offers a stable implementation for both the Timoshenko and the Cosserat models. The results of this comparison are reported in Table 3.1.

These results show that both models are able to accurately reproduce the physics of a beam, with the Timoshenko beam model showing slightly better performances. For the remainder of this manuscript, despite the comparable performances, the Timoshenko beam model will be adopted to model the physics of catheters and guidewires. This choice was mainly driven by the presence of previous works adopting this model to simulate the physics of catheters and guidewires *(Meng et al. (2022))*, and by the difficulties in handling contacts with the Cosserat model *(Lenoir et al. (2006))*.
SOFA offers also a modified catheter and guidewire model which allows to simplify the model of the interactions between these two devices. This model is based on Timoshenko beam theory and will be discussed in detail in Section 3.2.5.

3.2.2 Finite Element Method

To solve the physics of the instrument, the device is defined as a set of beam elements, each delimited by two nodes having six degrees of freedom, three for the translation and three for the rotation. The stiffness matrix $\mathbf{K}^{(j)}$ of each element, based on three-dimensional beam theory *(Przemieniecki (1985))*, becomes:

$$\mathbf{K}^{(j)} = \frac{E}{l}\left[\begin{array}{cccccc|cccccc}
A & & & & & & \multicolumn{6}{c}{\multirow{5}{*}{\textit{Symmetric}}} \\
0 & \frac{12I_z}{l^2(1+\phi_y)} & & & & & & & & & & \\
0 & 0 & \frac{12I_y}{l^2(1+\phi_z)} & & & & & & & & & \\
0 & 0 & 0 & \frac{GJ}{E} & & & & & & & & \\
0 & 0 & \frac{-6I_y}{l(1+\phi_z)} & 0 & \frac{(4+\phi_z)I_y}{1+\phi_z} & & & & & & & \\
0 & \frac{6I_z}{l(1+\phi_y)} & 0 & 0 & 0 & \frac{(4+\phi_y)I_z}{1+\phi_y} & & & & & & \\
\hline
-A & 0 & 0 & 0 & 0 & 0 & A & & & & & \\
0 & \frac{-12I_z}{l^2(1+\phi_y)} & 0 & 0 & 0 & \frac{-6I_z}{l(1+\phi_y)} & 0 & \frac{12I_z}{l^2(1+\phi_y)} & & & & \\
0 & 0 & \frac{-12I_y}{l^2(1+\phi_z)} & 0 & \frac{6I_y}{l(1+\phi_z)} & 0 & 0 & 0 & \frac{12I_y}{l^2(1+\phi_z)} & & & \\
0 & 0 & 0 & \frac{-GJ}{E} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{E} & & \\
0 & 0 & \frac{6I_y}{l(1+\phi_z)} & 0 & \frac{(2-\phi_z)I_y}{1+\phi_z} & 0 & 0 & 0 & \frac{6I_y}{l(1+\phi_z)} & 0 & \frac{(4+\phi_z)I_y}{1+\phi_z} & \\
0 & \frac{6I_z}{l(1+\phi_y)} & 0 & 0 & 0 & \frac{(2-\phi_y)I_z}{1+\phi_y} & 0 & \frac{-6I_z}{l(1+\phi_y)} & 0 & 0 & 0 & \frac{(4+\phi_y)I_z}{1+\phi_y}
\end{array}\right] \tag{3.28}$$

Here, $G = \frac{E}{2\cdot(1+\nu)}$ represents the shear modulus, A is the cross-sectional area of the beam, and l is its length. I_z and I_y describe the cross-section moment of inertia along the z and y axis respectively, while ϕ_z and ϕ_y denote the shear deformation parameters, and are defined as :

$$\phi_z = \frac{12EI_y}{GA_{syz}l^2} = 24(1+\nu)\frac{A}{A_{sz}}\left(\frac{r_y}{l}\right)^2, \qquad \phi_y = \frac{12EI_z}{GA_{sy}l^2} = 24(1+\nu)\frac{A}{A_{sy}}\left(\frac{r_z}{l}\right)^2 \tag{3.29}$$

Here, A_{sz} and A_{sy} represent the shear area in respectively the z and y direction and r_z and r_y are the radii of gyration along the same directions. When the ratios $\frac{r_z}{l}$ and $\frac{r_y}{l}$ are small with respect to the unity, as in the case of slender beams, ϕ_z and ϕ_y become negligible. This leads to a formulation in which the effect of shear deformations is neglected *(Przemieniecki (1985))*, and it is the formulation adopted in this work: catheters and guidewires are indeed long and thin devices, in which the effect of shear deformation is negligible.

3.2.3 Contact modeling

As introduced in Section 3.1.3, the PVW states that when a system is in equilibrium, the total virtual work done by all the internal and external forces, including the mechanical constraints, acting on the system is zero. Considering a dynamic system, the PVW can be written as *(Faure*

et al. (2012)):

$$\underbrace{(\mathbf{M}+dt\frac{d\mathbf{f}}{d\dot{x}}+dt^2\frac{d\mathbf{f}}{dx})}_{\mathbf{A}}\Delta v=\underbrace{-dt(\mathbf{f}+dt\frac{d\mathbf{f}}{dx}v)}_{\mathbf{b}}+dt\mathbf{H}^T\lambda \tag{3.30}$$

Here, v represents the velocity of the system, the term $\mathbf{M}$ denotes the mass matrix, $\mathbf{H}^T\lambda$ is the factor of constraint forces and $\mathbf{f}=\mathbf{f}_{vol}+\mathbf{f}_{surf}$ indicates the forces acting on the system, composed by the body forces $\mathbf{f}_{vol}$ and the forces acting on the surface of the system $\mathbf{f}_{surf}$. In the frame of the PVW, Δv can be seen as the virtual variation of the velocity v, the term $dt\mathbf{H}^T\lambda$ as the virtual work produced by the constraints, the term $\mathbf{b}$ as the virtual work carried out by the body and surfaces force and the term $\mathbf{A}$ as a modified mass term, accounting for the dynamic of the system.

The term $\mathbf{H}^T\lambda$ appears when the objects composing the system are in contact among them. This term is given by the product between the matrix of the constraint directions $\mathbf{H}$ and the Lagrange multipliers λ. There are two possible types of constraints, bilateral and unilateral, represented respectively by the functions $\iota(x_1,x_2,..)$ and $\Upsilon(x_1,x_2,..)$. This allows to define $\mathbf{H}$ as $\mathbf{H}_i=\left[\frac{\delta\iota}{\delta x_i};\frac{\delta\Upsilon}{\delta x_i}\right]$. Considering a system with two objects in contact, Eq. (3.30) becomes for each object:

$$\begin{aligned}\mathbf{A}_1\Delta v_1&=b+dt\mathbf{H}_1^T\lambda\\ \mathbf{A}_2\Delta v_2&=b+dt\mathbf{H}_2^T\lambda\end{aligned} \tag{3.31}$$

To solve these equations, two steps are performed *(Faure et al. (2012))*:

1. The physics of each object is solved independently considering $\lambda=0$, *i.e.* considering that the two objects are not interacting. The terms Δv_1^{free} and Δv_2^{free} are computed solving the equations:

$$\begin{aligned}\mathbf{A}_1\Delta v_1^{free}&=b_1\\ \mathbf{A}_2\Delta v_2^{free}&=b_2\end{aligned} \tag{3.32}$$

2. The constraints are solved considering $b_1=b_2=0$, to compute the corrective change in velocity Δv_1^{corr} and Δv_2^{corr} through the equations:

$$\begin{aligned}\mathbf{A}_1\Delta v_1^{corr}&=dt\mathbf{H}_1^T\lambda\\ \mathbf{A}_2\Delta v_2^{corr}&=dt\mathbf{H}_2^T\lambda\end{aligned} \tag{3.33}$$

   Here the Lagrangian multipliers λ represent the force to be applied in the constrained space to satisfy the constrain laws $\iota(x_1,x_2,..)$ and $\Upsilon(x_1,x_2,..)$. The constrained system

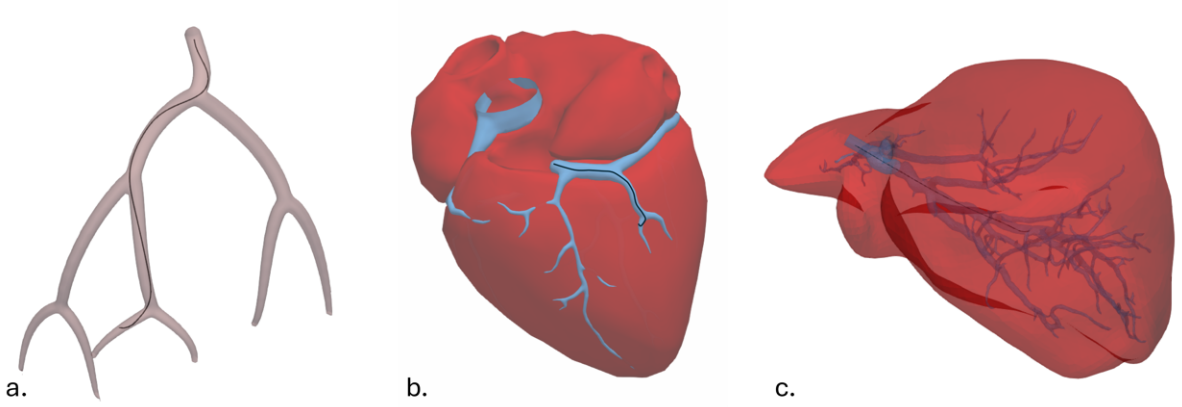

Fig. 3.4 Examples of the simulated environments developed in this work, showing the navigation of guidewires inside the vessel walls in three scenarios: a synthetic anatomy (a), the coronary arteries (b) and the liver vessels (c).

can be represented as:

$$\dot{\delta} = \mathbf{H}_1 v_1^{free} - \mathbf{H}_2 v_2^{free} + \underbrace{dt[\mathbf{H}_1 \mathbf{A}_1^{-1} \mathbf{H}_1^T + \mathbf{H}_2 \mathbf{A}_2^{-1} \mathbf{H}_2^T]}_{\mathbf{Z}} \lambda \tag{3.34}$$

Here, $\dot{\delta}$ is the constraint violation which can be computed from $\iota(x_1, x_2, ..)$ and $\Upsilon(x_1, x_2, ..)$, while $\mathbf{Z}$ has the dimension of a compliance.

Finally, the constraint problem is solved using the Gauss-Seidel algorithm *(Jourdan et al. (1998))*. After solving this linear system, the motion is computed as follows:

$$\begin{aligned} x_1 &= x_1^{free} + dt \cdot \Delta v_1^{corr}, \qquad \text{with} \quad \Delta v_1^{corr} = dt \mathbf{A}_1^{-1} \mathbf{H}_1 \lambda \\ x_2 &= x_2^{free} + dt \cdot \Delta v_2^{corr}, \qquad \text{with} \quad \Delta v_2^{corr} = dt \mathbf{A}_2^{-1} \mathbf{H}_2 \lambda \end{aligned} \tag{3.35}$$

Fig. 3.4 shows the results of the simulated environment here described in three different scenarios (a synthetic anatomy (a), the coronary arteries (b) and the liver vessels (c)). In each of these anatomies, the contacts between the device and the vessel walls are solved.

3.2.4 Time optimization strategies

Once the system of ODE describing the dynamic of the system has been built as explained in the previous section, the system must be solved following the steps presented in Section 3.1. For the temporal discretization of the dynamic discretized system, an Euler implicit scheme has been adopted. Given the particular geometry of the instruments to be modeled, in which each node of the mesh discretizing the tools is only connected to two other nodes, the matrix

A assumes a tridiagonal configuration. A block-tridiagonal solver *(Kolda and Bader (2009))*, which is a simplified form of the Gaussian elimination particularly optimized for the solution of this kind of system, can therefore be used to invert this matrix.
When simulating the physics of the devices navigating the vessels, due to the large number of contacts, the lightweight, and the high stiffness of the instruments, the matrix **A** can become ill-conditioned, causing convergence issues, which lead to long simulation times, despite the optimization strategies previously described. This is why, during the first phase of this project, a diagonal scaling *(Takapoui and Javadi (2016))* was applied to the matrix **A**, right before its inversion. A diagonal scaling is a particular type of matrix preconditioning, which consists of modifying a matrix in order to obtain a faster rate of convergence of the linear system that is being solved. This operation has a similar effect to adding mass to the system: it helps stabilize the system solution, but it also modifies the physics of the problem. To avoid this side effect, but to maintain at the same time the stabilizing effect, the diagonal scaling of **A** was applied only during the constraint solution step, before performing the computation of **Z**.

3.2.5 Catheter and guidewire interactions

The task of simulating the contacts between objects in an acceptable time frame is one of the most challenging tasks when applying the FEM. Indeed, adopting for example the strategy described in Section 3.2.3, the computational complexity of the contact solution increases as the number of contact points augments. In this work, two types of interactions are present: the interactions between the devices and the vessel wall and the interaction between the two instruments. During the intervention and, in particular, during the navigation phase, the catheter and the guidewire are constantly in contact because of their relative sizes and since they are inserted inside each other. Solving the contacts between these two instruments with the previously introduced method would involve a prohibitive computational cost, which would undermine the real-time characteristics of the simulation. This is the why to simulate the interaction between the instruments we adopted a different approach, with a strategy similar to *Lenoir et al. (2006)* and implemented in SOFA's BeamAdapter plugin. This technique replaces the model of two or more single devices inserted into each other with a composite model, which alters its properties based on the relative insertion of one instrument (usually the guidewire) inside the other (usually the catheter). When the guidewire is advanced inside the catheter, the shape of one instrument influences the shape of the other, based on their relative position (Fig. 1.2). This shape modification is generated by a change in the bending stiffness and bending moment in the region in which the two devices overlap. The transversal stiffness of this region

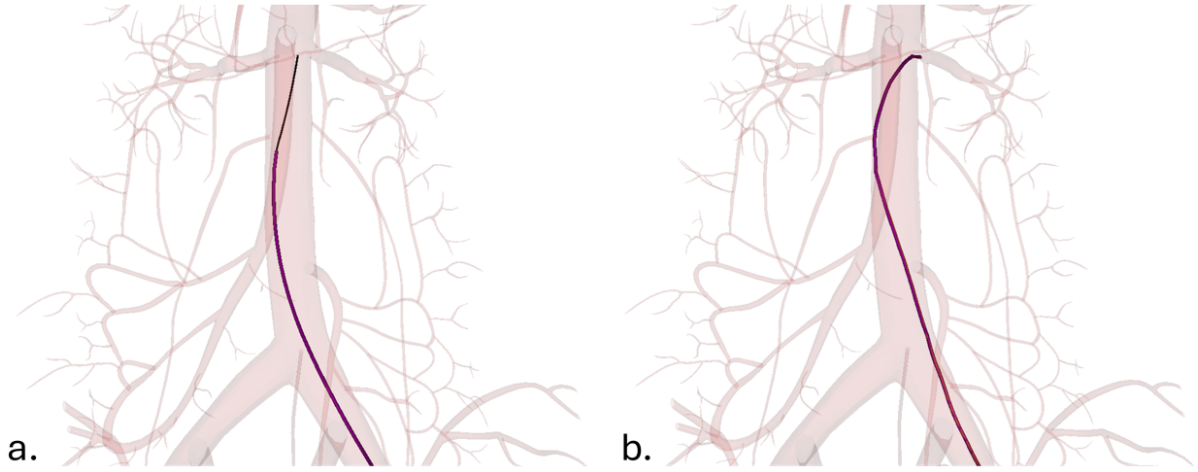

Fig. 3.5 Navigation of two instruments inside the blood vessels. On the right, the guidewire is inserted inside the catheter, while on the left the guidewire is withdrawn and the catheter recovers its original shape. In both cases, only one mechanical object is present in the scene, avoiding the need to consider too many contact points between the two instruments.

E_{trans} can be computed through the Halpin-Tsai *(Affdl and Kardos (1976))* equations:

$$E_{trans} = \frac{E_{cath}(1+\zeta h V_r)}{1-hV_r}, \quad h = \frac{E_{guide}-E_{cath}}{E_{guide}+\zeta E_{cath}} \tag{3.36}$$

here $V_r = \frac{V_{guide}}{V_{comb}}$, where V_{guide} is the volume of the guidewire and V_{comb} is the volume of the guidewire-catheter combination. ζ is a function of the material properties and geometry of the instruments, which can be found in lookup tables *(Affdl and Kardos (1976))*.
E_{trans} is used to update, in real-time, the stiffness of the composite model in the overlapped section, and the composite physical model reflects this change accordingly. This strategy allows the simulation of the interaction between the two surgical instruments, without calculating the computationally expensive collision between the two separate objects. The result of this strategy to compute the contacts can be observed in Fig. 3.5.

Chapter 4

The measurement system

Both the enhanced fluoroscopic image system and the autonomous control algorithms, which will be detailed in Chapters 5-8, rely on the strong assumption of knowing the 3D shape and position of the instruments navigating the vessels of the patient and the 3D shape of the anatomy. As introduced in Section 1.1.3, when needing to access small lateral vessels, it is fundamental to identify the best moment to withdraw the guidewire and allow the catheter to recover its original shape. At this stage, it is crucial to know the precise shape and position of the instruments inside the anatomy to perform a successful cannulation. During the navigation of small downstream vessels, it is necessary to correctly direct the tip of the instrument to navigate the desired branch of a bifurcation. To reach the target location, several bifurcations need to be overcome, each of them lying on a different plane which does not allow to observe them from a single 2D fluoroscopic image. Therefore, to use the systems proposed in this work in the real world, the first requirement is to have a reliable method to reconstruct both the 3D anatomy and the real-time shape and position of the instruments. While the 3D geometry can be retrieved from 3D preoperative images, such as computerized tomography angiography or magnetic resonance images, the acquisition of the real-time, 3D shape and position of the instruments requires a complex measurement system. Several choices are available today, ranging from the use of fluoroscopic images to the use of sensors such as electromagnetic (EM) sensors or optical fibers. All these possibilities are discussed in Section 4.1, with a particular focus on the adequate characteristics of optical fibers, which have been chosen in this work for the shape and position reconstruction of the instruments. Section 4.2 provides more details on optical fiber systems, and in particular on Fiber Bragg Grating systems (FBG), which as explained in Section 4.2, are particularly suited for strain measurements, from which the shape can be reconstructed, as detailed in Section 4.2.1. When applied to our specific case, this measurement technique revealed certain limitations (Section 4.2.3), highlighting the need for investigating potential solutions. In the medical field indeed, a clinically acceptable error is

in the order of $1-3$ *mm (Bourier et al. (2014, 2015))*, significantly lower than the accuracy demonstrated by the optical fiber system in some configurations. This is why, to address this issue, two main solutions were developed in the framework of this work, which are discussed in Section 4.3.

4.1 Sensor choice

With the development and increasing success of minimally invasive procedures, the need to provide surgeons with more information about the shape and position of surgical devices has become increasingly important, attracting growing interest from researchers. The first trivial approach involves the use of kinematic models, but the inherent deformable nature of interventional tools and their continuous collision with the anatomy make the real-time fine modeling of these devices and their interactions with the environment very challenging *(Webster III and Jones (2010))*, highlighting the need to explore alternative approaches. Three main emerging techniques for shape reconstruction were identified *(Shi et al. (2017))*: methods involving the use of particular imaging modalities, strategies adopting EM sensors and approaches using optical fibers.

The imaging modalities adopted for the shape reconstruction of surgical instruments range from fluoroscopic images to ultrasound and endoscopic imaging. Fluoroscopic images are particularly suited for reconstructing the shape of catheters and guidewires since these images are already available during endovascular procedures. *Wagner et al. (2016)* for example developed a method to obtain the 3D shape of catheters and guidewires starting from biplane angiography images. In particular, they implemented a bottom-up method using a flux-driven topology to extract the centerline of the surgical tools. The distance between adjacent segments and the distance with previous frames were minimized using Dijkstra's algorithm and the 3D shape was reconstructed using epipolar geometry. *Delmas et al. (2015)* proposed to use 3D curve segments instead of 3D point clouds calculated from the epipolar geometry, determining the most coherent curve in terms of continuity and curvature after the optimization from shape estimation. An interesting approach was also presented by *Canero et al. (2000)*, who suggested detecting the devices and the vessels through an energy minimization problem. Despite being very effective, all these methods require the use of biplane fluoroscopic images, *i.e.,* they require two simultaneous views to reconstruct the shape of the device. Obtaining these images in a real setup is very costly and requires a large radiation dosage *(Shi et al. (2017))*, limiting the applicability of these strategies. As a partial solution, some works *(Lobaton et al. (2013); Vandini et al. (2013))* integrate kinematic models or prior feature knowledge into monoplane fluoroscopic images, eliminating the need for specific instrumentation to acquire biplanar

fluoroscopic images. However, the use of this technique is limited by the need for multiple views of the observed region, which requires the movement of the C-arm used to acquire the fluoroscopic images *(Shi et al. (2017))*. Moving the C-arm during the intervention to reconstruct the shape of the device can disturb the execution of the intervention. At the same time, recovering depth information from monoplane fluoroscopic images and from a single view of the region of interest is an ill-posed problem that cannot be uniquely solved.

EM sensors have been largely used in the medical field thanks to their miniature size and their ability to function without a line of sight *(Shi et al. (2017))*. They exploit the principle of mutual induction to locate the receivers placed in a varying electromagnetic field, which is generated by an electromagnetic field generator. Their size allows for the integration along the body of continuum robots or catheters, and for the tracking of the distal tip of the device. In endovascular applications, they have been largely adopted for tracking steerable catheters. *Dore et al. (2012)* for example developed a method to fuse a physically-based simulation of the catheter kinematics with real-time in situ EM tracking, through a probabilistic fusion framework based on Kalman filtering. Very interesting is also the work of *Condino et al. (2012)*, which developed a catheter sensorised with two EM trackers, one placed at the distal tip of the instrument and the other positioned a few centimeters before. These two sensors allowed the reconstruction of the curvature of the distal part of the instrument, exploiting the theoretical and experimental result that the distal-end deflection profile of catheters can be approximated with a circular arch *(Ganji et al. (2009))*. The main limitation in the use of these systems lies in the working principle of EM sensors: indeed, when the EM field is disturbed by the presence of magnetic and conductive objects, these sensors produce important measurement errors. The operative field is full of these sources of disturbance (*e.g.,* electronic devices, metal instruments, etc.), significantly degrading the measurement precision of these sensors. Furthermore, EM sensors are very accurate around the center of the tracking volume, but when moving away from this area, the precision of the system diminishes *(Reichl et al. (2013))*. All these elements make it difficult to apply EM sensors in real clinical scenarios.

Fiber optic sensors (FOSs) can be divided into three main categories: light-intensity modulation-based FOS, interferometer-based FOS and FBG-based FOS. While the first two groups are well-suited for the measurement of forces, torque and pressure, FBGs are particularly adapted for the measurement of strain (Section 4.2). FOSs present many advantages, including small dimensions, high flexibility, electromagnetic immunity and absence of electrical connections *(Shi et al. (2017))*. Their lightness and small size allow their integration into medical tools with a reduced impact on the mechanical properties of the devices, and their electromagnetic immunity allows their use in the operative field, with no risk of accuracy deterioration. These characteristics make these sensors excellent candidates for reconstructing the shape and position

of surgical tools, and their use for this purpose has already been investigated in several studies. *Park et al. (2010)* for example equip an MRI-compatible biopsy needle with three optical fibers with two FBG sensors, achieving a good shape reconstruction accuracy (0.38 mm of root mean square error, RMSE). *Liu et al. (2015)* embedded into a cable-driven surgical robot designed for osteolysis treatment two optical fibers, each equipped with two FBGs, reaching an accuracy of 0.40 *mm*. FBGs have thus demonstrated their great precision when applied to stiff surgical tools, allowing for real-time shape estimation. Given these premises, FBGs have been adopted for the reconstruction of the shape and position of catheters and guidewires in this work.

4.2 Fiber Bragg Grating sensors

In the mid-1980s, the use of optical fibers for measuring bends and curvatures marked a significant technological advancement in sensor technology *(Lagakos et al. (1987))*. Since then, this technology has been developed and reached different fields, ranging from geothermal engineering *(Kori et al. (2020))* to the biomedical field *(Rohan et al. (2024))*. These sensors owe their success to their excellent characteristics, including their electromagnetic immunity, their reduced diameter ($\simeq 120\mu m$), their high interrogation speed, their multiplicity of sensing points, their simple interrogation system, and their high mechanical strength *(Al-Ahmad et al. (2020); Shi et al. (2017))*. In recent years, technological developments led to the evolution of techniques to reconstruct the strain starting from the signal measured by the optical fibers. The two most common techniques rely either on the FBG sensors or on the Rayleigh scattering technique. Both technologies reconstruct the strain starting from the reflection of wavelengths. The former relies on the periodic modification of fiber's refractive index, which describes the medium's light-bending ability, by means of optical absorption of the UV light *(Othonos (1997))*. The regions that underwent this modification are called *gratings*. When an incident light spectrum propagates inside the grating, the whole spectrum is propagated, except for a specific wavelength, called the Bragg wavelength (λ_B), which is reflected. When the fiber undergoes a deformation, the Bragg wavelength of each grating experiences a shift, and, starting from this shift, it is possible to reconstruct the fiber's strain and derive its deformed shape. Rayleigh scattering technique, on the other hand, does not require the use of gratings. In this case, the scattering of the light happens throughout the fiber length *(Froggatt et al. (2014); Megens et al. (2021))*. This technique provides a high spatial resolution and was adopted also in the biomedical domain *(Soller et al. (2005))*, but the lower costs of FBG-based interrogation technique, their high speed, and their high signal-to-noise ratio *(Al-Ahmad et al. (2020); Duncan et al. (2007))* make it a strong competitor. For these reasons, FBG-based technology was adopted during the development of this work. In the following, the basic principles behind

the shape reconstruction performed by means of the FBG sensors will be provided (Section 4.2.1, based on *Al-Ahmad et al. (2020)).*

4.2.1 Basic functioning principles and shape reconstruction

The characteristic Bragg wavelength of each grating can be obtained as $\lambda_B = 2n_{ef}\Lambda$, where n_{ef} is the optical fiber refraction index, and Λ is the grating period. When the fiber undergoes a deformation or is subjected to a temperature change ΔT, the fiber undergoes a shift in this wavelength, which can be linked to a strain variation inside the fiber. This strain variation can be either mechanical (ε) or related to ΔT. The change in the Bragg wavelength can be expressed as a function of these two contributions:

$$\frac{\lambda_B - \lambda_{B_0}}{\lambda_{B_0}} = \frac{\Delta\lambda_B}{\lambda_{B_0}} = S\varepsilon\Delta\varepsilon + S_T\Delta T \tag{4.1}$$

Here λ_{B_0} represents the unstrained Bragg wavelength, $\Delta\varepsilon$ represents the change in the strain with respect to the unstrained state, while S_ε and S_T describe the strain and the temperature sensitivity coefficients respectively.
In the case the fiber is subjected to bending, the mechanical strain ε, at the same time, can be used to find the curvature κ:

$$\varepsilon = -\kappa y \tag{4.2}$$

where y is the distance from the neutral bending axis, *i.e.,* the axis where the material of the beam is not under stress. This relation assumes symmetry with respect to the bending plane, no twisting and no buckling.
Nowadays, the most common setup for FBG sensors is the multi-core fibers (MCF) setting, in which several cores are inserted inside the same cladding, as shown in Fig. 4.1. In this configuration, the central core coincides with the neutral axis of the fiber: this core is thus only sensible to strain due to axial loads and temperature fluctuations. If the axial strain is negligible, then the wavelength shift due to temperature fluctuation $\Delta\lambda_{B,1}$ can be easily measured and compensated for the outer cores, allowing to rewrite Eq. (4.1) as:

$$\Delta_{\varepsilon_i} = \frac{\Delta\lambda_{B,i}}{S_\varepsilon\lambda_{B_0,i}} - \frac{\Delta\lambda_{B,1}}{S_\varepsilon\lambda_{B_0,1}} \tag{4.3}$$

From the configuration in Fig. 4.1, the relation between the strain ε and the curvature κ can be deducted:

$$\varepsilon_i = -\kappa\, r\, sin(\alpha_b - \frac{3}{2}\pi - \alpha_i) \tag{4.4}$$

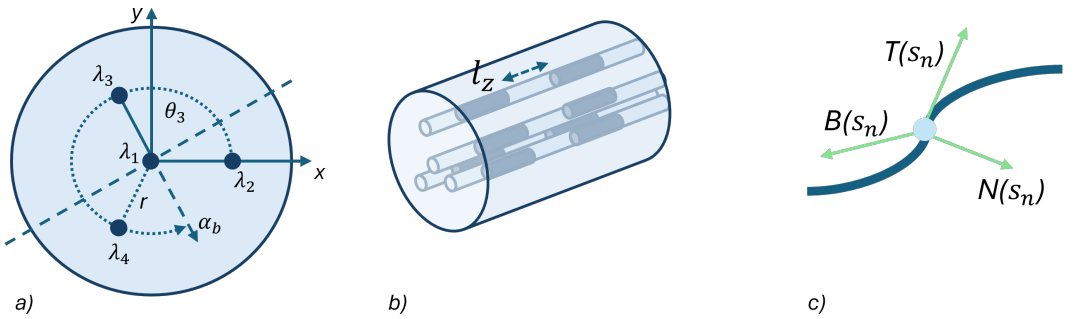

Fig. 4.1 MCF with 4 cores. a) Cross-section view of the fiber. λ_i is the wavelength of the i^{th} core, α_i describes the angle between the x-axis and the i^{th} core measured counter-clockwise, r represents the distance between the i^{th} core and the central axis, and α_b is the angle of the bending plane with respect to the axis x. b) Longitudinal view of the fiber, in which the distance l_z between two adjacent gratings is represented. c) Representation of the tangent vector **T**, the normal vector **N** and the binormal vector **B** to a point s_n of the optical fiber.

where r is the distance between the i^{th} core and the neutral axis, α_i describes the angle between the x-axis and the i^{th} core measured counter-clockwise and α_b is the angle of the bending plane with respect to the axis x. Considering this equation, if both twist and axial strain are considered negligible, determining κ and its direction α_b requires two unique strain measurements in different locations of the cross-section of the fiber. This means that only two outer cores are necessary to determine the two unknowns, but additional cores can enhance the precision of the measurements. If a fiber contains n FBGs distributed along its length, κ and α_b can be computed n times along the fiber, resulting in a set of curvatures $\kappa[n]$ and a set of bending angles $\alpha_b[n]$.

The first step for shape reconstruction is the measurement of a series of discrete FBG wavelengths $\lambda[n]$. For all sensor sets, the reference wavelength $\lambda_0[n]$ must be obtained, which represents the unstrained Bragg wavelength $\lambda_{B_0}[n]$. The wavelength of the central core $\lambda_c[n]$ is subtracted from both $\lambda[n]$ and $\lambda_0[n]$, obtaining the compensated wavelengths $\lambda_{comp}[n] = \lambda[n] - \lambda_c[n]$ and $\lambda_{comp,0}[n] = \lambda_0[n] - \lambda_{0,c}[n]$. Substituting $\lambda_{comp}[n]$ and $\lambda_{comp,0}[n]$ in Eq. (4.3), the strain ε_i can be computed and used in Eq. (4.4) to obtain the set of curvature $\kappa[n]$ and of bend angles $\alpha_b[n]$.

Defining now the arc length variable s, the curvature $\kappa(s)$ and the torsion $\tau(s) = \frac{d\alpha_b}{ds}$ describe how the curve's tangent **T**, the normal **N** and the binormal $\mathbf{B} = \mathbf{T} \times \mathbf{N}$ (Fig. 4.1 c) evolves along its length. Mathematically, this is defined using the differential Frenet-Serret formulae *(Banchoff and Lovett (2010))*. The Cartesian position $C_s(s)$ can then be obtained by integrating Eq. (4.5), which can be solved using various methods. For further details, please refer to

Fig. 4.2 Components of FBGS measurement system: the interrogator (left) and the optical fiber (right).

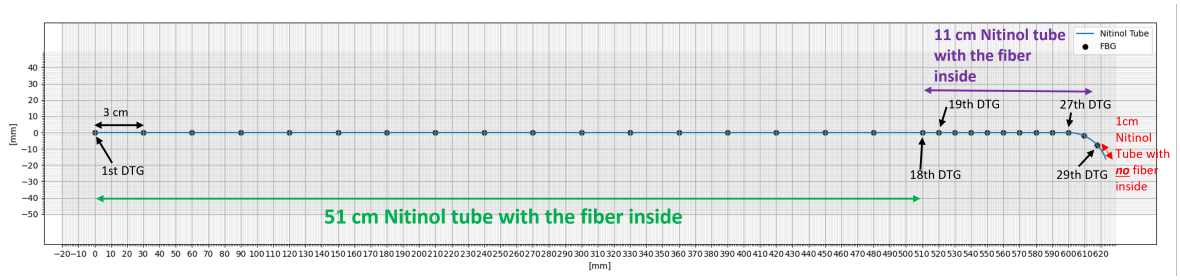

Fig. 4.3 Optical fiber design. The black dots represent the DTG inscribed in the optical fiber.

Al-Ahmad et al. (2020).

$$C_s(s) = C_{s,0} + \int_0^l \mathbf{T}(s)ds \tag{4.5}$$

Here $C_{s,0}$ is the Cartesian coordinate of the curve's base and l is the interrogated length.

4.2.2 Fiber design

In this work, the commercial system from FBGS International (Geel, Belgium), a leading company in the production and development of FBG technology, was employed. The hardware components of the system are shown in Fig. 4.2. The interrogator (on the left) is a measurement device based on the spectrometer technology. Its task is to send the light signal to the fiber (Fig. 4.2 right), connected in the front part of this device, and measure the reflected wavelengths. This system is complemented by the *ILLumiSense* software, which is used to process and visualize the wavelength collected by the FBGs and convert these data into strains. These data are taken as input by a second software, the *ShapeSense* software, which reconstructs the shape of the fiber based on the principles introduced in Section 4.2. The shape returned by this software is always defined in a reference frame fixed in the most proximal DTG.
The fiber adopted in this work is equipped with 29 4-core draw tower gratings (DTGs). A

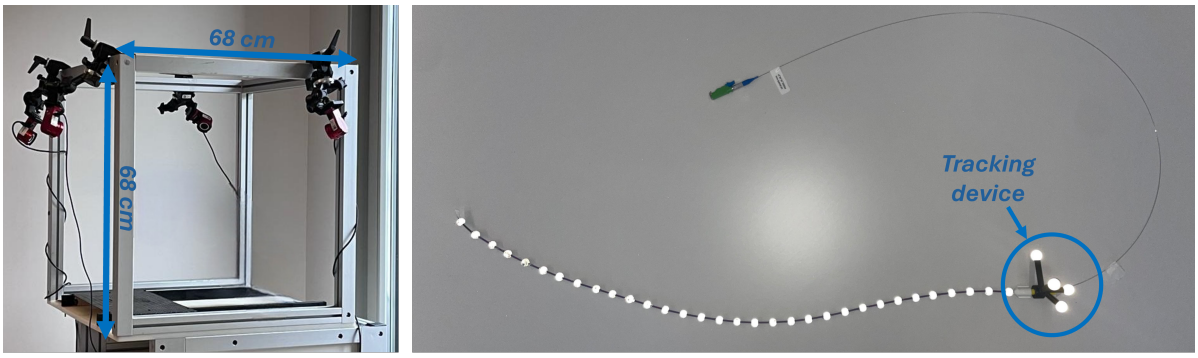

Fig. 4.4 OptiTrack acquisition system. On the left, the metallic cage used as support for the OptiTrack cameras and defining the acquisition area of the OptiTrack system is shown. On the right, the setup used to acquire the real-time position of the FBG optical fiber is depicted. The fiber is positioned into a catheter on which infrared refractive markers are positioned to allow the shape reconstruction by means of the OptiTrack system. A tracking device is positioned on the most proximal DTG to allow the registration of the shapes reconstructed by the OptiTrack system and by the ShapeSense software in the same reference frame.

DTG is a grating that underwent a particular production process that combines the drawing of the optical fiber with the writing of the grating. This special process ensures a high breaking strength and uniform coating coverage. To protect the fiber from mechanical stress, a 0.380 μm diameter Nitinol tube was used to cover the sensor. The design of the fiber was customized to meet the specific needs of our application. In particular, to ensure enough sensing length for the insertion in realistic vascular phantoms, a 64 cm-long fiber was designed. The distribution of the DTGs affects the precision of the sensor *(Shi et al. (2017))*: a high DTG density allows the detection of smaller curvatures, but at the same time, the number of DTGs that is possible to inscribe in the same fiber is limited. Indeed, each grating reflects a portion of the light that is passing through the fiber and when the number of gratings is too high, the amount of reflected light significantly reduces the light signal. These requirements led to the design of a fiber with two different regions: a first region composed of 18 3-cm-spaced gratings, and a second one formed by 11 1-cm-spaced gratings. The high density of the tip allows for an improved shape reconstruction in this region, which is the most significant for the purposes of this work. Finally, the distal region of the fiber presents a curvature of 0.033 $\frac{1}{mm}$ to reproduce the tip of selective guidewires. The final design of the fiber can be observed in Fig. 4.3.

4.2.3 System limitations

As introduced in Section 4.1, FBGs have demonstrated their great accuracy when applied to stiff surgical tools, allowing for real-time shape estimation. However, when the device presents

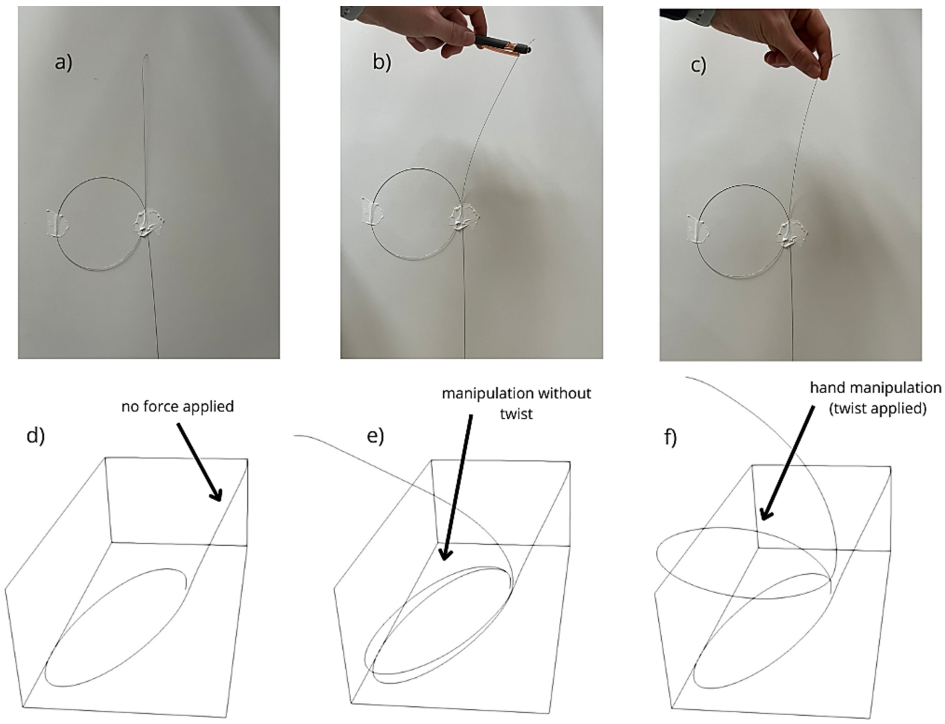

Fig. 4.5 FBGS ShapeSense software limitation. The top raw shows three different manipulations imposed on the sensor: no fiber manipulation (a), fiber bending (b) and fiber twisting (c). The bottom raw (d,e,f) shows the shape reconstructed by the ShapeSense software. In particular, in e and f, the shape reconstruction is compared with the shape reconstructed when no manipulation is applied to the fiber (a), showing how the system is robust to fiber bending (e) but an important error is introduced when twist is applied (f).

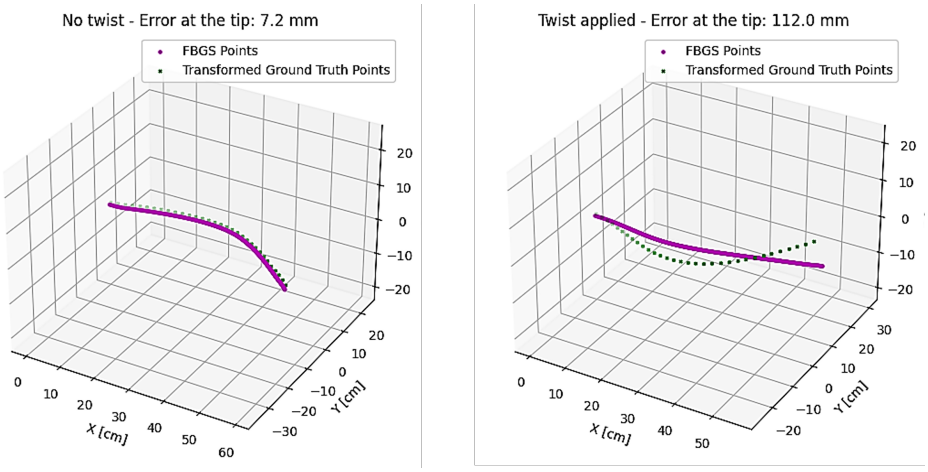

Fig. 4.6 Comparison between the shape reconstructed by ShapeSense software and the ground truth, acquired through the OptiTrack system, without (left) and with (right) twist.

a low stiffness, the performances of the system are sharply reduced *(Shi et al. (2017))*. This performance decrease was also observed in the context of this study and is demonstrated in Fig. 4.5. For this test, the optical fiber was placed in a circular configuration (Fig. 4.5 a) and the system was asked to reconstruct the detected shape in three different conditions: no manipulation, fiber bending (Fig. 4.5 b) and fiber twisting (Fig. 4.5 c). For each of these manipulations, the respective shape obtained through the ShapeSense software is reported in the second raw of Fig. 4.5. In particular, Fig. 4.5 e and f report, besides the reconstructed shape, the comparison with the shape obtained when no manipulation was applied. In all three configurations, the proximal part of the fiber is fixed, and only the distal part is manipulated with the described deformation: the reconstructed shapes should therefore lie always on the same plane and be fixed, and only the distal part of the fiber should show a difference in shape. This is the case of Fig. 4.5 e, in which the shapes a and b are reconstructed almost in the same position, while an important performance deterioration is reported when the fiber undergoes twist (Fig. 4.5 f). This performance deterioration was quantified with the help of an OptiTrack system (OptiTrack, NaturalPoint, Inc.), which is an advanced motion capture technology using a series of high-speed infrared cameras to track the position and orientation of markers placed in the sensing region (Fig. 4.4 left). For this purpose, 32 optical markers were placed on a catheter in which the optical fiber was inserted (Fig. 4.4 right). A tracker was fixed on the first DTG and rigidly connected to the first infrared marker. The tracker device, which can be observed in Fig. 4.4, is composed of a luer lock, able to secure the optical fiber, and of 4 arms of different lengths, each terminated with a refractive marker to allow the OptiTrack system to reconstruct at every instant a reference frame fixed on this device. This allows the registration of the shape acquired by the OptiTrack system and the shape obtained through the ShapeSense software in the same reference frame. To ensure the precise reconstruction of the position of the optical markers, 6 cameras were positioned around the perimeter of the top square of a cube-shaped metal cage, defining an acquisition volume of approximately $160 \times 10^3 cm^3$ (Fig. 4.4 left). Five different shapes were compared and a mean error at the tip of 7.2 *mm* was found when no twist was applied, which increased to 68 *mm* and 110 *mm* when a twist of 15^o and 45^o respectively was applied (Fig. 4.6)

4.3 Accuracy optimization strategies

During the navigation of vascular trees, the clinician controls the movement of catheters and guidewires by pushing or rotating them. Given the constraints imposed by the vessels on the surgical devices, their flexibility and length, the torsion applied to the proximal end of the surgical tools produces a high amount of twist in the devices themselves. The inaccuracies

introduced by the twist in the FBG measurements do not allow the use of the shape reconstructed by ShapeSense software for the purposes of this work: even a slight twist could cause the instrument's shape to be erroneously reconstructed outside the vessel. To solve this issue, we developed two different techniques: a first naive solution consisting of a rigid registration to ensure that the position of the reconstructed shape is inside the vessel lumen (Section 4.3.1), and a second more refined technique to reconstruct the shape of the instrument directly from the wavelength measured by the DTGs (Section 4.3.2). While the first technique still requires the use of the ShapeSense software, the second strategy bypasses this step and takes as input the raw data coming from each DTG.

4.3.1 Rigid registration

One of the main sources of error introduced by the FBG sensor when imposing a twist is the movement of the reference frame. Fig. 4.5 f depicts this issue, showing how imposing twist at the tip of the instrument, the shape of the fixed part of the fiber completely changes its position, but maintains the circular shape imposed on the fiber. This shows that when a twist is imposed on the sensor, the reference frame is not reliable anymore, but the reconstructed shape still presents characteristics related to the real shape of the instrument. The first strategy developed in this work to compensate for the error introduced by the twist exploits this characteristic and the knowledge of the anatomy to be navigated. Indeed, one of the main assumptions behind all the systems developed in this work is the availability of a 3D model of the navigated anatomy. Often, the vessels present a tortuous path: the curvature of the various parts of the anatomy and their sequence usually allow for their unique identification. Thanks to this characteristic, the shape reconstructed by the ShapeSense software was rigidly registered *(Birkfellner et al. (2007); Yaniv (2008); Zollei et al. (2001))* on the centerline of the vessels, extracted from its 3D model (Fig. 4.7). This strategy was implemented using the *RigidRegistration* method available in the *pycpd* Python library *(Gatti and Khallaghi (2022))*. This code implements the Coherent Point Drift algorithm *(Myronenko and Song (2010))*, and in addition to the adopted rigid registration method, includes two additional registration methods: the affine registration and the Gaussian-regularized non-rigid registration. Both in the choice of the library and in the parameters tuning, particular attention was paid to the minimization of the computational time, to maintain the real-time nature of the process. In particular, a maximum of 50 iterations and a tolerance of 0.5 *mm* was set for this problem. Since the curvature of each segment of the vascular tree can be calculated from the centerline and since the ShapeSense algorithm provides as output also the curvature of each DTG, this information was used to pass to the registration algorithm only the portion of the centerline corresponding to the detected high

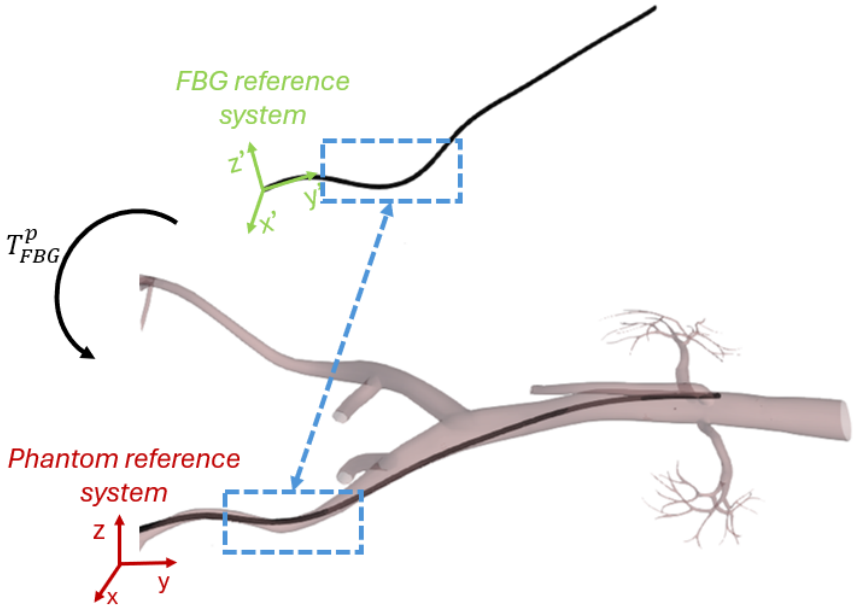

Fig. 4.7 Rigid registration of the fiber shape, reconstructed in FBG reference frame and affected by twist errors, in the reference frame of the phantom. The matrix T^p_{FBG}, computed using the coherent point drift algorithm, allows to perform this transformation. In the blue square, an example of the region presenting similar curvature and used for the rigid registration is shown.

curvatures. This strategy allowed to minimize registration errors and to reduce the convergence time of the algorithm.

However, the described technique only guarantees that the reconstructed shape is prevailing positioned inside the vessel's lumen, but it does not ensure that all the points constituting the shape are placed inside the vessel wall and it does not correct the errors introduced by the twist. Furthermore, when the diameter of the vessel is large, errors in the order of the vessel diameter are introduced. Since several configurations and positions satisfy the imposed constraint, the general accuracy of the system is connected to the dimensions of the vessels. A more general solution is therefore required to enhance the precision of this measurement system.

4.3.2 Neural Network correction

Given the limitations of the previously discussed method, an alternative strategy was explored. Based on prior research studies *(Ha et al. (2023); Manavi Roodsari et al. (2024); Roodsari et al. (2023))*, we applied deep learning techniques to obtain a network that, after the training, was able to reconstruct the shape of the device starting from the raw outputs of the optical fiber. In particular, this work was carried out in the context of the master thesis project conducted by Francesco Dettori, who completed his internship in our team in August 2024.

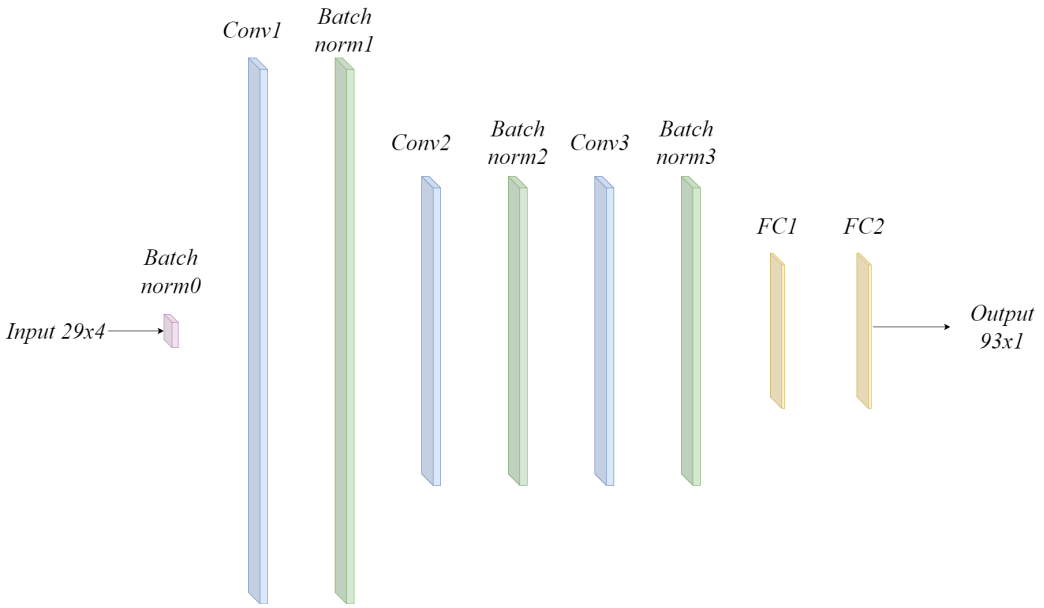

Fig. 4.8 Final configuration of the neural network used to reconstruct the shape of the optical fiber. The NN is composed of three convolutional layers, each followed by a batch normalization layer, and two fully connected layers (FC1, FC2). The NN takes as input an array of 29×4 wavelengths obtained from the FBGs raw data and gives as output an array of dimension 93×1 containing the relative position of 31 points along the shape, each provided as x, y, z displacement with respect to the previous point.

The neural network developed in this work, inspired by *Manavi Roodsari et al. (2024)* research, is composed of convolutional layers, suited to automatically and adaptively learn spatial hierarchies of features from inputs, followed by fully connected layers, which perform high-level analysis. The hyper-parameters of this network were defined by means of a grid search, which resulted in the choice of the architecture reported in Fig 4.8. The network is composed of three convolutional layers, each followed by a batch normalization layer and two fully connected layers. As shown in Fig. 4.8, the NN takes as input an array of dimension 29×4, which contains the Bragg wavelengths collected through FBGS optical fiber. The first convolutional layer and the respective batch normalization give as output an array of size $[-1, 2048, 27]$, which is reduced by the second and third convolutional layers, which outputs an array of size $[-1, 1024, 24]$ and $[-1, 1024, 22]$ respectively. The first fully connected layer returns an array of dimensions $[-1, 500]$, followed by the second and last fully connected layers which provide a 93×1 constituting the output of the network. The grid search led also to the definition of an adaptive learning rate and to the choice of a sigmoid activation function. Finally, the Adam optimizer *(Kingma and Ba (2014))*, an adaptive learning rate optimization algorithm, and the smooth L1 loss were adopted.

The 93×1 array which constitutes the output of the network provides the corresponding shape of the device in the form of x, y, z relative coordinates of the 31 points along the fiber. Each of these coordinates provides the relative position of one point with respect to the previous one, and the first of these coordinates defines the position of the first point with respect to the origin of the reference system in which the shape is reconstructed. The ground truth position of each of these points is provided by the OptiTrack system, with a setup similar to the one introduced in Section 4.2.3. This way the origin of the reference system of the shape reconstructed by the neural network coincides with the most proximal optical marker, which is rigidly connected to the tracking device described above: knowing the position and orientation of this tracking device, allows therefore to register the shape and position reconstructed by the NN in a known frame. The training datasets were acquired by collecting simultaneous data from both the OptiTrack system and the FBGS IllumiSense software. To this purpose, particular attention was paid to the synchronization of the data coming from the two measurement systems, which was automated through a Python script. Two different datasets were acquired, a generic one (*Generic_Dataset*) and a dataset containing *application specific* shapes (*ApplicationSpecific_Dataset*). The first dataset was collected by randomly manipulating the fiber to maximize the variability of acquired shapes. The second one was instead assembled imposing to the fiber a limited range of shapes, similar to the shapes encountered during the navigation of vascular trees, to understand the influence of an application-specific dataset on the system accuracy. For this purpose, the model shown in green in Fig. 4.9 was used for the data acquisition. This model reproduces the geometry of the 3D-printed vascular model shown in Fig. 4.9 in yellow, defining the same bifurcations and paths by means of small cylinders, which can be used to shape the fiber into the desired form. This design was conceived to allow to simultaneously acquire data with the OptiTrack system and the FBG fiber: indeed, to acquire the data with the OptiTrack system, the markers placed on the optical fiber (Fig. 4.4) must remain visible to the cameras. This would not be possible with a closed 3D-printed phantom like the one shown in yellow in Fig. 4.9. In both datasets, twist was introduced to inspect the capability of the network to compensate for this phenomenon.
Both the input (the wavelength) and output (OptiTrack positions) were separately preprocessed and fused in a unified data frame, consisting of a corresponding pair of model input and ground truth. In particular, the inputs were normalized using the mean μ and standard deviation σ from all the wavelengths of the dataset, using the formula:

$$z_{ij} = \frac{x_{ij} - \mu}{\sigma} \tag{4.6}$$

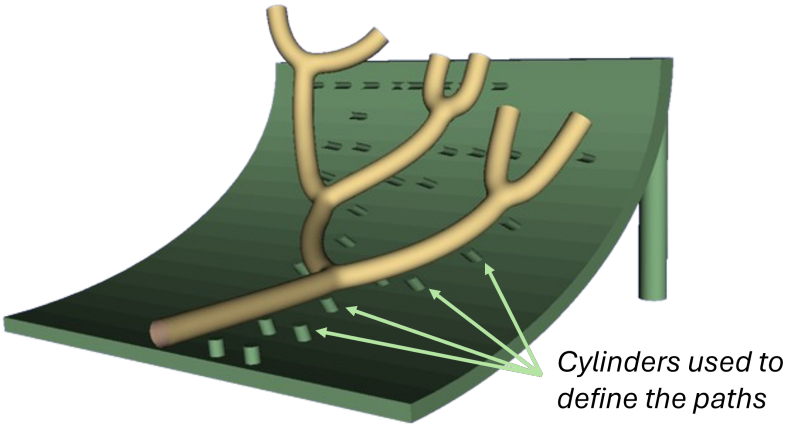

Fig. 4.9 Representation of the 3D printed model with paths used to acquire the shape of the ApplicationSpecific_Dataset. The tracks are defined by small cylinders and reproduce the shape of the phantom from which the paths were inspired, which is shown in yellow.

Here, x_{ij} is the unmodified wavelength of the j^{th} grating of the i^{th} DTG. Each data point is therefore transformed into a *z-score*, which represents the number of standard deviations the original value is away from the mean. This process, known as standardization or z-score normalization, ensures that the normalized data has a mean of 0 and a standard deviation of 1. The output data are initially acquired as a series of 32 coordinates representing the position of the 32 markers in the space. These points are first registered in the reference system defined by the tracking device (Fig. 4.4) through the rigid transformation:

$$p_{marker}^{fiber} = T_{OptiTrack}^{fiber} \cdot P_{marker}^{OptiTrack} \tag{4.7}$$

in which P_{marker}^{fiber} are the x, y, z coordinates of the markers in the fiber frame, $T_{OptiTrack}^{fiber}$ is the 4×4 homogeneous transformation matrix which allows passing from the OptiTrack to the fiber reference frame, obtained through the tracking device pose, and $p_{marker}^{OptiTrack}$ is the position of the marker in the OptiTrack reference system. These positions are then transformed to express the coordinates of each point P_i as relative positions with respect to the upstream marker P_{i-1}.

$$\Delta P_i = P_i - P_{i-1}, \quad i \in [2, ..32] \in \mathbb{Z} \tag{4.8}$$

The first marker, attached to the tracking device, constitutes the origin of the reference system.

The computer used for the training process was equipped with an Intel® Core™ i7-4790K CPU @ 4.00GHz, 16GB of RAM, and an NVIDIA Corporation GP104 [GeForce GTX 1070] dedicated graphics card with 8GB of memory. The training time for the two models

Table 4.1 Mean tip error and mean RMSE in the shape reconstruction predicted by the trained models *Generic_Model* (*G_Model*) and *ApplicationSpecific_Model* (*AS_Model*) and in the shape obtained through FBGS ShapeSense software.

Error	**G_Model**		**AS_Model**		**Shapesense**	
	No Twist	*Twist*	*No Twist*	*Twist*	*No Twist*	*Twist*
Mean tip error [mm]	31.9	32.1	20.9	23.6	17.0	87.2
Mean RMSE [mm]	11.0	9.5	7.0	5.9	9.4	32.3

(*Generic_Model* and *ApplicationSpecific_Model*) was approximately 40 hours to complete 50,000 epochs.

The network was tested on 5 different shapes acquired on the 3D printed model shown in Fig. 4.9 and recorded separately from the training dataset. For each of these 5 shapes, three configurations were considered: no twist, 90^o twist, and -90^o twist. Two metrics were evaluated: the mean of the Euclidean distance between the position of the 32^{nd} point reconstructed by the NN and its position in the ground truth (*tip error*) and the average RMSE between the real and predicted shape. The *Generic_Model* reported an average tip error of $31,9$ *mm* (no twist) and of 32 *mm* (twist) and an RMSE of 11.0 *mm* (no twist) and 9.5 *mm* (twist) in the same conditions, while the *ApplicationSpecific_Model* achieved a mean tip error of 20.9 *mm* (no twist) and 23.6 *mm* (twist) and an RMSE of 7.0 *mm* (no twist) and 5.9 *mm* (twist). In the same condition, FBGS ShapeSense software produced a mean tip error of 17.0 *mm* (no twist) and 87.2 *mm* (twist) and an RMSE of 9.4 *mm* (no twist) and 32.3 *mm* (twist). This test shows once again the good accuracy of FBGS ShapeSense software when no twist is applied to the fiber and an important performance drop when the twist is present. Both our trained models show a slightly lower precision compared to ShapeSense software when the fiber undergoes only bending, but the performances are maintained when the twist is applied to the fiber, demonstrating the robustness of our method to twist. Furthermore, *ApplicationSpecific_Model* outperforms *Generic_Model*, highlighting the importance of constructing a dataset tailored to the application.

Generic_Model was also tested in conditions similar to the ones of *Manavi Roodsari et al. (2024)* work. To this purpose, the network was first trained following a *random split method*, *i.e.* was trained and validated on a randomly selected part of the data and tested on the remaining part. In these conditions, our model obtained an RMSE of 1.3 *mm* and a median tip error of 1.6 *mm*, comparable to the 1.5 *mm* RMSE and 2.1 *mm* median tip error obtained by Manavi

et al.. A second test was performed using a dataset in which the samples were acquired continuously and the training and testing dataset were completely separate. In this case, our model obtained an RMSE of 5.7 *mm* and a median tip error of 9.1 *mm*, outperforming Manavi's model which reported an RMSE of 9.8 *mm* and a median tip error of 17.1 *mm*. It is important to note that there are significant differences between our work and that developed in Manavi's study. Specifically, their study utilized a 1-core, 30-cm long edge-FBG fiber, inscribed with 5 gratings, and used as input intensities from all wavelengths reflected by the fiber at the current time t, and at the two previous instants $t-1$ and $t-2$.

The results obtained in this research show that the adopted technique is suited for the development of a reliable method to reconstruct the shape of an FBG optical fiber, directly from the measured wavelengths. In particular, the results summarised in Table 4.1 show that our method maintains the same level of accuracy when the twist is imposed on the sensor, proving the robustness of our network to this manipulation. The accuracy of *ApplicationSpecific_Model* demonstrated how the precision of the network can be improved by acquiring data representing deformation similar to the ones that will be applied during the application of the system. However, the accuracy reached by our system is not yet sufficient for clinical applications. Therefore, strategies to improve the precision are necessary to exploit this measurement system during the autonomous navigation of catheters and guidewires. A first possibility involves the fusion of the registration technique presented in Section 4.3.1, with the strategy presented above. This would permit the merging of the advantages of both techniques: while the rigid registration would bound the error to the dimensions of the navigated vessel, the twist would be compensated by our neural network. Another option involves the correction of the 3D shape of endovascular devices reconstructed by our NN with information gathered from fluoroscopic images acquired from time to time. This has already been inspected by some previous studies *(Ourak et al. (2021))*, along with the possibility of fusing sparse monoplane fluoroscopic images, EM sensor information and FBG data *(Ha et al. (2021))*.

An important final consideration involves the choice of the commercial system used in this work. FBGs provide many great advantages, such as small dimensions and insensitivity to EM disturbances, making these devices a very good option for the reconstruction of the shape and position of endovascular devices. However, this technology is still in its early stage of development for medical applications given the required high accuracy. Even though FBGS is among the leaders in the production of optical fibers, alternative more expensive commercial systems based on the Rayleigh scattering technique (briefly presented in Section 4.2) already exist. It is the case of the Fiber Optic RealShape (FORS) technology, which is an exclusive of Philips company (Koninklijke Philips N.V.). This technology has already been applied for the reconstruction of the shape of endovascular devices *(Finnesgard et al.*

(2023); Panuccio et al. (2023)). Previous studies assessed the precision of the Philips FORS system, reporting a submillimeter accuracy for the tip location, and a milliradian accuracy for the orientation *(Megens et al. (2021))*. Furthermore, this technology has already been integrated into a commercial system, LumiGuide (Philips), which promises the possibility to obtain intraoperative, radiation-free, 3D guidance.

Chapter 5

Enhanced fluoroscopic images

As introduced in Section 1.2, given the limited amount of visual information and the difficult control of surgical devices, the cannulation phase of the intervention can be very challenging even for expert clinicians. Indeed, to place the tip of the catheter in the correct location, the caregiver needs to master the combined control of the devices and be able to predict which is the correct moment and position to withdraw the guidewire. To provide more support in this phase of the intervention, researchers are nowadays focusing on the development of Virtual Reality (VR) and Augmented Reality (AR) systems to enrich the information provided by fluoroscopic images or to develop systems for the training of new practitioners. Virtual reality methods aim to create an immersive environment for the users, which interacts with a fully virtual world by means of controllers. These systems are nowadays very common for entertainment purposes, but they are acquiring an increasing interest also in the medical field, where they can be employed for the training of new practitioners, to simulate interventions, or to practice new tools. Augmented reality, on the other side, overlays videos or images in a display representing the physical world. In these systems, simulated objects can interact with objects present in the real world. AR systems can for example overlay on real-world objects their structure, such as the elements composing an engine or the internal structures of a patient, or they can show the user instructions overlaid on the images of the real world. Furthermore, these elements have the potential to simplify the interactions between humans and machines, bypassing the need to use real physical controls. In the past 3 decades, several efforts have been made by the researchers to provide this kind of support to clinicians. The existing works will be discussed in Section 5.1, with a focus on supporting systems for endovascular interventions. Section 5.2 will then illustrate the solution proposed in this work. In particular, we present a system able to provide more informative fluoroscopic images, by fusing fluoroscopic image augmentation techniques and physics-based simulations. The resulting system overlays, on the fluoroscopic image, not only the virtual vascular anatomy but also the predicted shape of the catheter for a

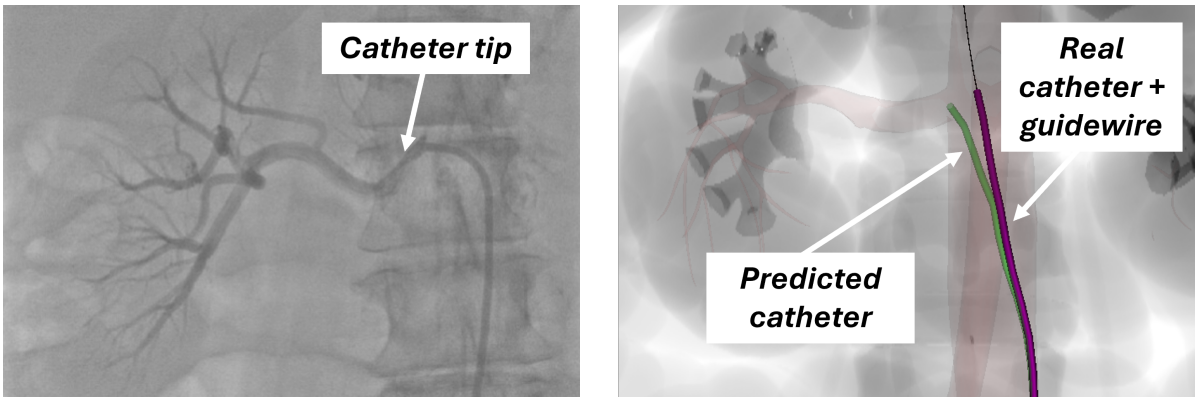

Fig. 5.1 Real fluoroscopic image showing catheter and guidewire navigation in the abdominal artery and renal artery cannulation (left) (license: CC BY 3.0). A synthetic fluoroscopic image (right) augmented with our predictive catheter model is used for assisted renal artery navigation.

predetermined action (Fig.5.1). The accuracy and the usefulness of this framework have been assessed through a validation and a user study, whose results are reported in Section 5.3.

The work described in this Chapter was presented at the *18th AE-CAI, 10th CARE and 6th OR 2.0 MICCAI workshop* in October 2024 and won the *Outstanding Paper Award*. It was also accepted for publication in the *Special Issue on AE-CAI | CARE | OR 2.0 of Wiley's Healthcare Technology Letters journal*.

5.1 Literature overview

Virtual reality and augmented reality systems are becoming increasingly popular in medical research, and when applied to the interventional radiology field, they usually consist in enriching fluoroscopic images with real-time or pre-operative information. One of the first works in the field appeared in 1996 and developed a VR simulator for interventional radiology training. In particular, this system combined digital medical images shown on a computer monitor, with a system able to provide force feedback, integrating tool position and tracking with computer graphics techniques, allowing to see the result on the screen *(Dawson et al. (1996))*. With the advancement of technology, it has been possible to develop increasingly complex systems, but given the critical field of application, which impacts patient health, and the great precision required by medical applications, the number of works addressing this topic is still limited. *Soulami et al. (2016)* created augmented fluoroscopy images to assist the clinician during transcatheter aortic valve implantation, exploiting the virtual reconstruction of the aortic root. *Kuhlemann et al. (2017)* conceived a system to visualize a 3D holographic view of the vascular system directly on the patient without any need for radiation. Initially, the patient's

surface is extracted and represented as a mesh, and the vascular tree shape is reconstructed from pre-operative CT data. These two models are then registered to the real-world object using magnetic tracking. An EM tracker placed at the tip of the instrument also allows to follow the movement of the device inside the blood vessels. The system has been tested on a vascular phantom model (Blue phantom FAST Trauma Full Torso Ultrasound Training Model), showing the clinician the vascular structures through Microsoft HoloLens technology. This framework constitutes a part of a larger project, aiming at reducing radiation exposure and contrast agent administration during endovascular aortic repair and at optimizing visualization during these procedures *(García-Vázquez et al. (2018))*. Besides the 3D model of the skin and vessels, the user can access 2D information, such as orthogonal views of computer tomography angiography and 2D images of the vascular model. The main limitation of this work lies in the use of an EM sensor to track the tip of the device: this information is indeed not sufficient to retrieve the complete shape of the device. This can lead to undesired effects, such as accidental damages to the vessel, due to a wrong manipulation of the devices caused by the lack of information on the orientation and position of the instrument. Another interesting study was conducted by *Salavitabar et al. (2022)*. This work describes an augmented reality visualization of a 3D angiography during congenital cardiac catheterization. In this case, the model of the vessel of the patient was reconstructed from 3D rotational angiograms and transmitted to a Microsoft Hololens system for intraprocedural visualization in the operative room. This allows the operator to easily manipulate and slice the model to visualize the structures of interest. The system was tested on 5 patients undergoing congenital cardiac catheterization. Only minor discrepancies were reported between the real anatomical structure and the model. The main limitation of this work is that the model was not registered on fluoroscopic images, requiring the clinician to get used to an additional source of information. To directly superimpose the vessel model on fluoroscopic images, *Guan et al. (2019)* developed a method able to register digital subtraction angiography images with their corresponding 3D models in reduced computational times, with a precision of less than 1 *mm* in the tested cases. In their work, they rely on a multi-channel convolutional neural network, which combines a convolutional neural network with a periodic vascular diameter variation model. The main drawback of this method lies in the definition of this model, which is possible only when a complete dataset containing all the phases of the periodic movement is available. *Li et al. (2024c)* proposed a deformable model-to-image registration framework to provide the clinician with augmented-reality images during endovascular interventions. This framework offers a complete pipeline, that starts with the segmentation of intra-operative fluoroscopic images. These images are then registered to the pre-operative 3D model reconstructed from CTA images and, through a deep residual U-Net network, the deformations of the 2D model are estimated

and reconstructed onto the pre-operative 3D model. The clinician can then access these images through an immersive visualization system. The 2D registration method was evaluated through the modified dice similarity coefficient value obtaining a value of 0.89 ± 0.02, demonstrating this way the accuracy of this system.
All these works present augmented reality systems that can provide physicians with more detailed, three-dimensional, information about the anatomy of the patient. However, they do not provide additional support to the procedure, as they are essentially visualization systems and not guidance systems.

Other works focus on the development of simulators, allowing to augment the safety and reduce the cost of the traineeship of new interventional radiologists. In this case, a critical aspect is the modeling of the physics of the devices, which must reproduce as accurately as possible the behavior of the real instruments. *Korzeniowski et al. (2017)* developed a virtual reality, real-time simulator for cardiovascular interventions that enables to perform a complete angioplasty. In this case, the devices are modeled exploiting the Cosserat model, and their properties are tuned based on the properties of real devices. In their work, *Wang and Wu (2021)*, developed a simulator of catheter ablation procedures. The devices are modeled as a set of rigid rods linked by discrete points and the beating of the heart is simulated to add realism. *Luboz et al. (2013)* developed a virtual environment teaching the core skills of vascular interventional radiology. In this work, the guidewire was modeled through linked rigid bodies. However, in their recent paper, *Li et al. (2024b)* claim that the main limitation of current training systems is in their limited fidelity of blood vessel and guidewire models, their lack of the necessary real-time responsiveness and their incompleteness in the modelization of vessel blood physics. As a solution, they propose a framework that integrates position-based dynamics shape matching and Cosserat elastic rods. To improve the modelization of the contact between the vessel wall and the instruments, they present a multi-level progressive continuous collision detection method. All these elements allow them to achieve the required accuracy in the modelization of both vascular and guidewire models.
All these studies propose simulated environments suitable for surgical training that exploit different physical models but do not aim to provide intraoperative support to the surgeon.

5.2 Predictive fluoroscopic images

To address the aforementioned limitations, and in particular the lack of intraoperative support to the clinician, we developed a framework that besides visually augmenting fluoroscopic images with 3D anatomical information, provides also additional information on the outcome of the clinician's actions, before actually performing them (Fig. 5.1). This additional support,

provided through real-time, predictive simulation, can reduce the number of failed attempts and, consequently, shorten the duration of the intervention, as well as the number of contrast agent injections and X-ray exposure for both the patient and the caregivers. To achieve this objective, it is not sufficient to have data regarding the anatomy of the patient, but it is crucial to have also a reliable physical model of the devices to predict their behavior in situations that have not yet occurred in the real world. In this work, we essentially focus on the change of shape of the catheter when the guidewire moves, but the concept and implementations are generic and could be applied to other actions.
The system is composed of the following parts:

- A shape sensing system to reconstruct the shape, in real-time, of the real catheter when it is inside the vascular anatomy. This is achieved by exploiting Fiber Bragg Grating (FBG) sensors, which are able to reconstruct the 3D shape of the instrument, as explained in Chapter 4

- A virtual environment in which the navigation of the instruments inside the vascular network is simulated. The physics of the catheter, the guidewire and their interaction are computed in real-time, along with the contacts with the vascular walls (see Chapter 3). This simulation is driven by a set of points sampled along the shape of the real catheter. The simulation guarantees that the model follows the clinician inputs and regularizes the data returned by the tracking system (Section 5.2). The simulated environment can be hosted on a remote computer, placed away from the operating table.

- A display system that provides feedback to the clinician during the intervention in the form of an enhanced fluoroscopic image, in which the original image is fused, in real-time, with the predictive simulation.

The idea of predictive augmented navigation is to provide the clinician with information on the outcome of his actions (device navigation, guidewire withdrawal, etc.), before actually performing them. The cannulation of the left renal artery is a good example to illustrate how the system works (Fig. 5.1). When approaching the region in which the secondary vessel originates, the system starts to show the shape of the catheter *as it would be* after guidewire withdrawal, helping the clinician maximize the chances of a successful one-shot cannulation. The framework developed in this work is essentially an assistance tool and it is represented in Fig. 5.2. The system takes as input 4 pieces of information: a model of the physics of the surgical tools, the shape of the devices, and in particular of the catheter, reconstructed using the FBGs sensors, the non-contrasted fluoroscopic images and the 3D model of the anatomy that is being navigated. This framework uses the physics-based model of the catheter and

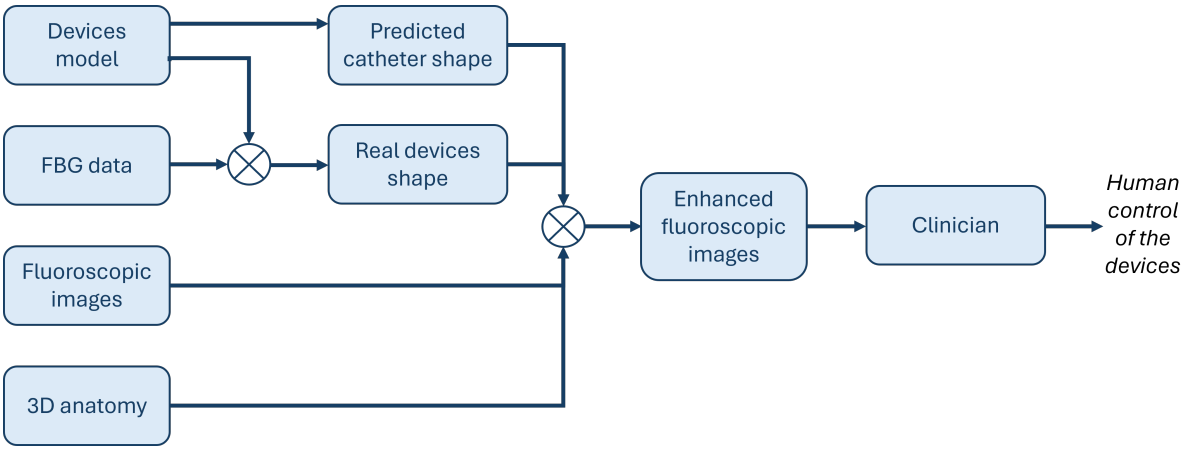

Fig. 5.2 Scheme describing the creation of the enhanced fluoroscopic images. The system takes as input 4 main pieces of information: the model of the physics of the surgical devices, the shape of the catheter reconstructed by the FBGs sensors, the non-contrasted fluoroscopic images and the 3D model of the anatomy that is being navigated. The physical model is used both to compute the predicted shape of the catheter after guidewire withdrawal and to regularize the data coming from the FBGs sensors. These two pieces of information are then overlapped with the 2D fluoroscopic image, along with the navigated anatomy, to create the enhanced fluoroscopic images. Guided by these images, the clinician performs the intervention with additional support.

guidewire described in Chapter 3 for two purposes: to simulate the shape of the catheter after guidewire withdrawal and to compensate for the errors introduced by the FBG system (see Section 4.2.3). This part of the work will be detailed in Section 5.2.1. These two pieces of information and the vessel shape are then overlapped with the fluoroscopic images (see Section 5.2.2) to produce the enhanced fluoroscopic images that the clinician will use as guidance to perform the cannulation task.

5.2.1 Data driven endovascular navigation

The system setup involves two elements: the replica in the simulated environment of the motion of the actual devices manipulated by the user and the computation, starting from this configuration, of additional information provided to the clinician directly on the fluoroscopic images (Section 5.2.2). To compute the predictive view, having a reliable physics-based model is crucial to anticipate the behavior of the device and show the clinician a truthful prediction to help her/him perform the task more efficiently. The prediction is computed at each time step, using the real device shape and position and accounting for the interaction between this simulated catheter and the real vessel walls, reproduced in the simulated environment. To achieve all of this, the model described in Chapter 3 was used. The mechanical properties of

the devices were determined based on real device's characteristics. The Young Modulus of all the devices was chosen by means of a bending test, performed both in the real and in the simulated environment. The Young Modulus of the models was iteratively changed until the value that minimized the difference between simulated and real data was found.
As already introduced, the real device shape is reconstructed by exploiting FBG data. The fiber is inserted inside the catheter and the two tools are rigidly attached by fixing the proximal part of the surgical tool to the sensor and paying attention to aligning the fiber tip with the tip of the catheter. Fixing the sensor and the surgical device only at the proximal end allows to reduce the twist imposed on the FBGs when navigating the catheter inside the vessels. The shape data are collected every 0.1 s. This shape information is then transformed in the same reference system as the anatomy through a rigid registration, following the process introduced in Section 4.3.1. This registration process alone allows to limit the reconstruction error to values in the order of the diameter of the vessel. In the considered case in particular, the abdominal aorta presents a diameter of $\simeq 3$ *cm*, but since the surgical tools are inserted inside the iliac artery, which has a diameter of $\simeq 4$ *mm*, the reconstruction error is here limited to this value. However, given the accuracy required by this application and the impossibility of compensating for twist errors with the adopted strategy (Section 4.3.1), it was essential to develop new strategies. The approach adopted in this work to solve this issue involved the development of a *regularisation technique* to correct the data coming from the FBG sensor and registered in the reference frame of the anatomy. To perform this regularisation, the physics-based model described in Chapter 3 was once again used. During the navigation of the devices inside the vessels, besides the predicted shape of the catheter, the simulation of the navigation of the real devices was therefore performed and the regularization was computed according to Equation 5.1. More specifically, this equation imposes the balance of forces:

$$f_{int} = f_{ext} + f_{correction} = f_{ext} + \alpha(x_{sim} - x_{FBG}) \tag{5.1}$$

Here, f_{int}, f_{ext}, $f_{correction}$ are respectively the internal, external and correction forces of the beam. In this first study, the vessel walls were considered rigid and the effect of blood flow was neglected: f_{ext} were therefore only constituted by the contact forces, computed as explained in Section 3.2.3. The correction term $f_{correction}$ was calculated by weighting the difference between the position of the devices computed by the simulation x_{sim} and the position measured by the sensors x_{FBG} by a constant α. We used $\alpha = 0.8$, which was tuned experimentally. Details on the accuracy of the reconstructed shape are reported in Section 5.3.1.

5.2.2 Visualization

The enhanced view superimposes the predicted virtual shape directly onto the fluoroscopic image. This choice was made to make our system more intuitive for clinicians. In clinical practice, indeed, the operator is guided by 2D fluoroscopic images. For this reason, even if the data computed by our system are 3D data, rather than displaying the model on a separate screen, the predictive model is shown on the same plane as the fluoroscopic images. Therefore, the 3D model of the virtual vascular anatomy augmented with the actual instruments' shape and the virtual catheter shape is registered on the live fluoroscopic images through rigid registration methods *(Birkfellner et al. (2007); Yaniv (2008); Zollei et al. (2001))*. For the registration of device shapes into the 3D vascular model, the method discussed in Section 4.3.1 has been used. This makes the system less intrusive since it does not require additional equipment and does not modify the operating setup to which the clinician is accustomed.

5.2.3 Implementation details

The development of this framework required the setup of a communication system between two different computers and the integration of the data gathered from the FBG sensors into the simulated environment. The simulated environment is created by a Python script, *CompleteScene.py* that imports all the needed libraries and plugins, and initializes all the components of the scene. In this case, the complete scene is composed of three surgical devices (the representation of the real catheter, the simulated catheter and the guidewire), the mesh of the navigated vessels, and the mesh of the organs surrounding the vessels. The latter are required to produce the simulated X-ray images. All these elements are initialized by calling the python class *Device*. The mechanical characteristics of the devices and geometries are set in a *JavaScript Object Notation (json)* file, *sceneCongif.json*. Besides creating all the simulated objects, this script also detects the region close to the target vessel in which the simulated catheter is shown, manages the simulation of the physics of the predicted catheter shape, handles the communication with the remote computer which collects the FBG data and integrates these data with the physical model of the real catheter. The FBG sensors measure data continuously at a frequency of 110 Hz. These data are continuously collected by the laptop placed close to the operating table. FBGS systems send a series of information to the receiving machine, which elaborates these data by means of a Python script that only stores the curvature and position information. The same Python script also applies the rigid transformation described in Section 4.3.1 and sends to the computer hosting the simulated environment the data already registered in the reference frame of the vessels. Every 0.1 s, the simulated environment sends a message to the remote computer asking for new data. When receiving this message, the remote computer applies the

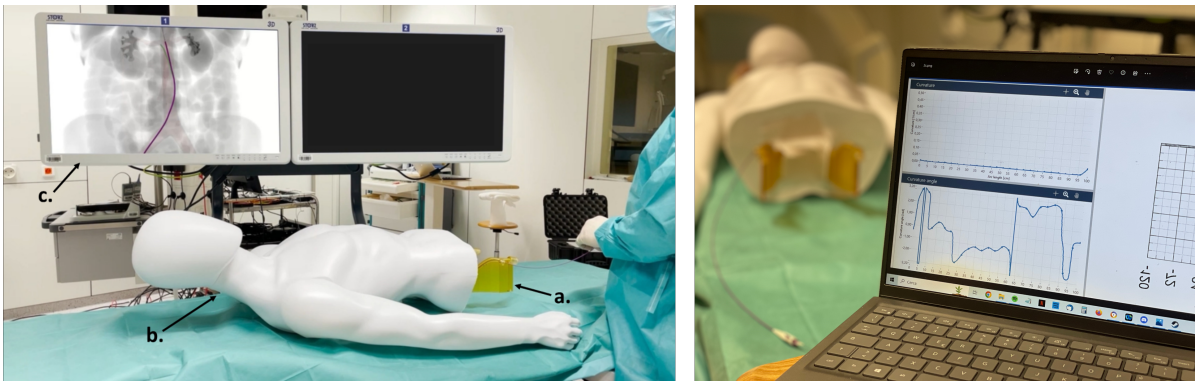

Fig. 5.3 Operating room setup. Left: The phantom (a) is inserted inside a mannequin (b), the laptop and the instrument to elaborate optical fiber data are placed far from the insertion point of the instruments (not visible in the image) and the screen (c) in which the predictive images are shown is placed to the left of the operating table. Right: Instrument shape acquisition by means of the FBG system provided by FBGS.

rigid transformation to the last available data and sends the required information. To ensure the data have been correctly received, the computer running the simulation sends a terminal string to the remote computer. This process ensures that the communication between the two machines runs smoothly, that no data is lost, and that no useless information is sent from the remote to the local workstation. Both the local and the remote computers are equipped with an IntelCore i7 processor running at 3.40 GHz.

The simulated environment adopted in this work is the one described in Chapter 3. The adopted physical and geometrical characteristics of the models are reported below. The catheter considered in this work is a Cobra catheter, suited for navigating abdominal aorta transversal vessels. The undeformed shape of the catheter into which the optical fiber was inserted was used as the rest shape of the model. An external diameter of 1.7 mm, an internal diameter of 1.4 mm and a length of 430 mm were measured. Special attention was paid to the catheter tip, characterized by a lower stiffness. After a fine-tuning of the parameters, a Young's Modulus of 3.4 MPa and 0.7 MPa were found for respectively the body and the tip of the catheter. Concerning the guidewire, a straight device with a diameter of 0.24 mm was chosen to be inserted inside the Cobra catheter. A length of 400 mm was adopted and a 500 MPa Young's Modulus was found through a bending test. To test the accuracy of the physical model described in Chapter 3, the combined shapes obtained both in the real world and in the simulated environment by inserting the guidewire inside the catheter were compared. A mean error of 2.0 ± 0.9 mm was measured between the two shapes. It is important to notice that this combined model is obtained by computing the interaction of the two separate models described in Chapter 3, as explained in Section 3.2.5.

5.3 System evaluation

In this section, all the technical characteristics of the framework are reported.
The complete setup of the operating room, including our instrumentation, is shown in Fig. 5.3. The usability of our framework is proven during renal artery cannulation. In this first study, an anatomical rigid 3D-printed vascular network was adopted and the effect of blood flow was neglected. To evaluate the robustness and usefulness of the system, a validation study, whose results are reported in Section 5.3.1 and a user study, reported in Section 5.3.2 were performed.

5.3.1 Validation Study

During the cannulation, the instruments are manipulated in order to place the catheter's tip at the entrance of the vessels to be reached, which in this case are the right and left renal arteries. These vessels have a diameter of 6 mm: to provide support during the procedure, the predictive model must produce a prediction whose error is lower than this value. For this reason, the accuracy of the model was tested by performing multiple insertions in a rigid 3D-printed phantom reproducing a human abdominal aorta anatomy (Fig. 5.4 a. and g.) and comparing the catheter shape predicted by our system (Fig. 5.4 in green) with the real catheter shape obtained after guidewire withdrawal (Fig. 5.4 in purple).

The experience involved the insertion of the guidewire and the catheter, equipped with the optical fiber, both in the right and in the left femoral artery and the navigation of the two instruments toward the origin of the renal arteries. During the navigation, the instruments were translated together and our system showed the current position of the instruments in real-time, exploiting the data collected through the optical fiber (Section 5.2.1), and the predicted catheter shape after guidewire withdrawal (Fig. 5.4 b, h).

For both the insertion in the right and in the left iliac artery, the navigation was stopped in 4 different locations and the accuracy of the system was measured by calculating the distance of the catheter shape predicted by our system and the real catheter shape obtained after guidewire withdrawal and measured through the optical fibers (Fig. 5.4 c, d, e, f, i, l, m, and n). Since the predicted catheter shape was only shown in the region in which the renal arteries originate, only device positions close to this region were considered. A total of 8 different locations were evaluated and are shown in Fig. 5.4. A mean 3D error of 2.4 ± 1.3 mm was found, with a mean of 1.3 ± 0.6 mm for the navigation from the right iliac artery to the right renal artery and 3.5 ± 0.8 mm from the left iliac artery toward the left renal artery. Since the 3D model is superimposed on 2D fluoroscopic images, also the 2D error on the same plane as the fluoroscopic images was computed. A total mean 2D error of 1.1 ± 0.7 mm was found, with a value of 0.7 ± 0.5 mm for the right navigation and 1.4 ± 0.6 mm for the left one. The values

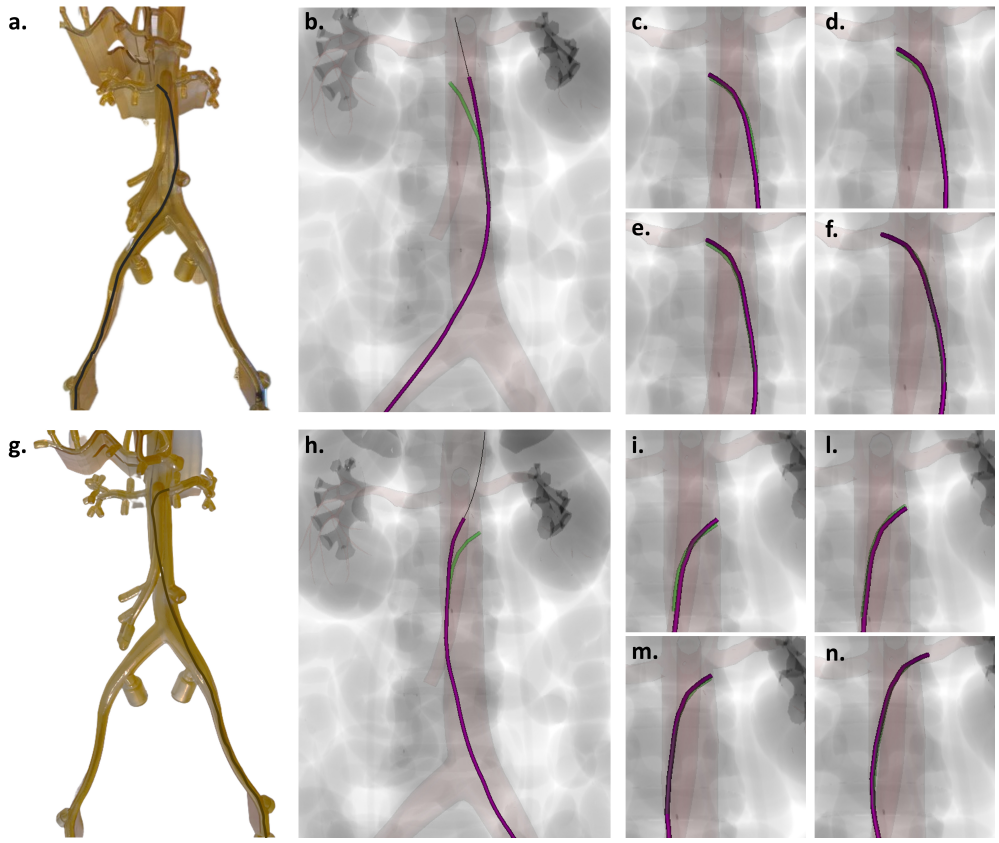

Fig. 5.4 Validation study. The guidewire and the catheter equipped with the optical fiber are inserted inside a phantom in the right (top row) and left (bottom row) iliac artery. The system, besides the real instruments' shape and position (Fig. b and h in purple), shows the predicted catheter shape after guidewire withdrawal (Fig.b and h in green). Fig. c, d, e, f, i, l, m, and n show the comparison between the predicted catheter shape computed by our system (in green) and the real catheter shape obtained after guidewire withdrawal (in purple) and acquired through the optical fiber system. A mean 3D error of 2.4$\pm$1.3 mm and 2D error of 1.1$\pm$0.7 mm were measured between the two shapes.

obtained for each test are reported in Table 5.1.

We believe that the higher errors found in the navigation from the left iliac artery toward the left renal artery are due to larger uncertainties in the shape reconstructed by the FBG sensor: since the phantom is rigid and the interior surface is not-smooth, the tip of the fiber can get stuck in some regions during the navigation, imposing non desired deformation of the catheter and possible twisting effect which can the be transmitted to the sensors.

Table 5.1 Validation study - Errors between the catheter shape computed by our system and the real catheter shape obtained after guidewire withdrawal in the locations shown in Fig. 5.4.

Right	***Error [mm]***		***Left***	***Error [mm]***	
insertion	*3D*	*2D*	***insertion***	*3D*	*2D*
Loc. c	1.9	0.5	*Loc. i*	2.5	1.2
Loc. d	0.7	0.4	*Loc. l*	3.5	2.4
Loc. e	0.9	0.5	*Loc. m*	4.4	1.3
Loc. f	1.8	1.5	*Loc. n*	3.5	0.9
Mean[mm]	1.3±0.6	0.7±0.5	**Mean[mm]**	3.5±0.8	1.4±0.6
Total Mean 3D Error[mm]		2.4±1.3	**Total Mean 2D Error[mm]**		1.1±0.7

5.3.2 User study

The usefulness of the system was assessed through a user study in which 4 different subjects were involved: three non-expert users and an experienced interventional radiologist. Each subject was introduced to the experimental setup composed of a rigid 3D-printed phantom representing the anatomy of the abdominal aorta, inserted inside a mannequin as shown in Fig. 5.3: only the access point, placed at the level of the femoral aorta was visible. The fluoroscopic images were shown on a screen placed in front of the operator, to reproduce as well as possible the setting of a real intervention. To avoid unnecessary X-ray exposure for the users, synthetic fluoroscopic images were generated using the Beer–Lambert law *(Pointon et al. (2023))* and the 3D mesh of the organs of the same subject from whom the anatomy of the abdominal aorta was obtained. The catheter (containing the optical fiber) and the guidewire manipulated during the experience are those described in Section 5.2.3. Each subject was required to perform a total of three experiences in three different scenarios without the possibility of repeating them to prevent a bias due to the learning curve. The aim of each experience was to perform a successful cannulation. At the beginning of the test, the catheter and the guidewire were inserted in the phantom's right renal artery. Then, manipulating both devices, the user performed the cannulation guided by fluoroscopic images containing for each trial different information:

- Classical fluoroscopic images, which do not show the anatomy of the vessels (Fig. 5.5 a). To reproduce as closely as possible the situation of real endovascular interventions, the 2D anatomy of the vessel was shown in the upper left part of the screen, to simulate the static road map typically used during these procedures.
- Fluoroscopic images augmented with the 3D model of the vessel (Fig. 5.5b)

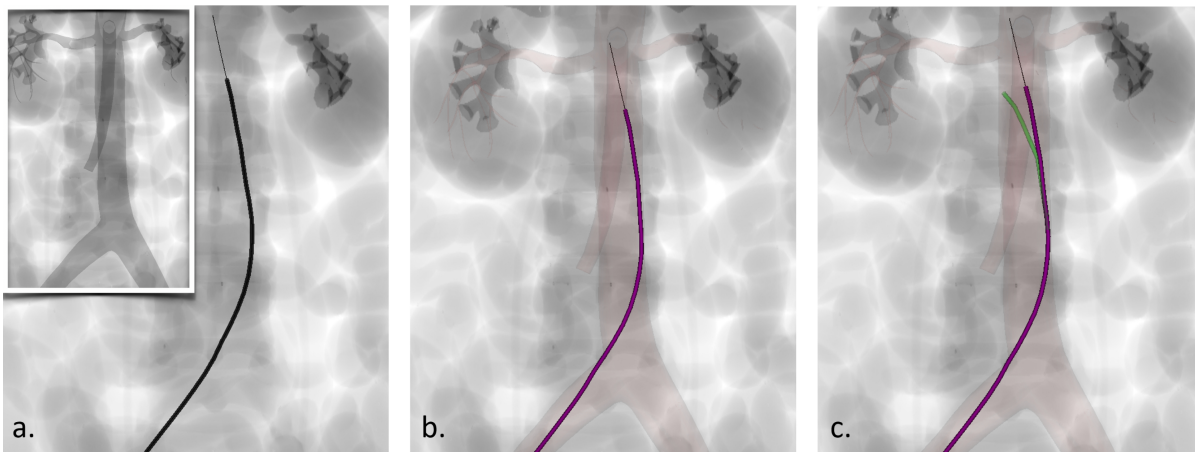

Fig. 5.5 User study illustration: a. First, the user is asked to navigate using the fluoroscopic image without contrast agent, seeing only a static roadmap in the upper left corner; b. Second, the user navigates using a fluoroscopic image augmented with the 3D vessel anatomy; c. Finally, the user tries to reach the target using our predictive navigation approach. Each user performed the cannulation three times, each time with one of the three image guidance shown in these pictures. For all the subjects, the use of our system reduced the time required to perform the cannulation.

- Fluoscopic images augmented by our predictive system (Fig. 5.5c). In this case, once the region close to the origin of the lateral vessels was reached, the predicted shape of the catheter (represented in green) started to be visualized.

During the experience, the user was observing the screen in which the movement of the combined devices was reconstructed in real-time as the instruments were navigated in the phantom. The user was required to manipulate the devices and withdraw the guidewire at the right time to perform the cannulation. For each subject and for each experience, the time elapsed from the insertion until the successful cannulation was measured. If, after guidewire withdrawal, the target was not reached, the guidewire was re-advanced inside the catheter and the guidewire withdrawal was repeated until the success of the operation. The results of the user study are reported in Table 5.2. Timings are measured in seconds, the value in parenthesis corresponds to the number of failed attempts.

A reduction in the time required to perform the cannulation can be observed from this table. In particular, the use of our system leads to an average procedural time reduction of 56% compared with classical fluoroscopic imaging guidance and 30% compared with the guidance of fluoroscopic images augmented with the 3D model of the vessel. Furthermore, a differentiation can be observed between the procedural time required by the expert user and the non-expert users, whose performances are leveled when using our system.

Table 5.2 User study results. For each user, the time required for the navigation is reported, while the number of failed attempts is shown in parentheses.

User ID	***Fluoroscopic images only***	***Fluoroscopic images + anatomy***	***Predictive navigation***
Non-Expert 1	81 sec.	52 sec.	34 sec.
Non-Expert 2	74 sec.	70 sec. (2)	34 sec.
Non-Expert 3	136 sec. (2)	49 sec.	38 sec.
Expert	29 sec.	20 sec.	17sec

5.4 Final considerations

In this work, a predictive system to support clinicians during endovascular interventions, in particular during the navigation and the cannulation tasks, is developed. These procedures are guided through 2D-fluoroscopy images in which the vessels are only visible upon contrast agent injection. Our system aims to help overcome this limitation by superimposing on the fluoroscopic images the information of the anatomy of the vessel, and augmenting the visual feedback with information about the outcome of the current action, before actually performing it. To the best of the authors' knowledge, this is the first time that such a system is developed. In the current work, real-time physical simulations are used together with optical fiber measurements to obtain a reliable 3D reconstruction of the current position and movement of the real devices on the fluoroscopic images. This shape, together with the physics-based model, is then used to compute a prediction able to support the clinician intra-operatively, especially during the cannulation. Real-time simulations are ensured through an optimization that helps reduce the computational time needed to resolve the contacts between the vessel wall and the instruments. The reliability of the model was tested through a validation study in which the predicted shape of the catheter was compared to the real shape of the device after guidewire withdrawal on the same plane as the fluoroscopic images that guide clinicians during the intervention. A mean error of 1.1 ± 0.7 mm was found, which is sufficient to offer support to the intervention.
Finally, the user study reported a reduced duration and a reduced number of trials to obtain a successful cannulation using our system, proving its usefulness. A reduced intervention time can indeed lead to lower radiation intake for both the patient and clinicians and to a lower number of contrast agent injections.

Chapter 6

Autonomous Guidewire Navigation in Static Environments

Efficiently and safely navigating the catheter and guidewire through the vascular anatomy is essential to ensure minimal exposure of the patient and clinician to X-ray radiations induced by the fluoroscopic imaging system. This requires a perfect knowledge of the anatomy, excellent control of the device, and a deep understanding of fluoroscopic visualization. Yet, even experienced clinicians can spend tens of minutes until reaching certain targets. As already introduced in Chapter 1, the only partial solution that exists today is the use of robotic systems, which remain only leader-follower devices controlled by the clinician. To provide clinicians with more support, current research focuses on the development of autonomous and semi-autonomous systems. To this end, several deep reinforcement learning (DRL)-based methods have been proposed. They generally take fluoroscopic images as input and predict a control action (rotation and translation) to be applied at the proximal end of the device. An overview of these systems will be provided in Section 6.1. However, all of these works usually present a limited generalization capability, a characteristic that is fundamental for an autonomous navigation system: blood vessels are indeed subjected to physiological movements and deformations and even the same anatomical structure can show important differences among patients. To address this issue, a particular training strategy has been developed in this work to obtain a controller that, after proper training, is able to navigate all the possible anatomical variations. The development of this work required different elements, such as the choice of the most suitable algorithm, the choice of an appropriate reinforcement learning framework, the choice of a convenient training environment, and the design of a proper reward function. All these choices will be discussed in Section 6.2. Section 6.3 describes the learning strategy developed in this work, while Section 6.4 reports the results obtained with the method described in this work when navigating static anatomies in a simulated environment. The

control algorithm has also been tested in the real world in open-loop. The adopted materials and the experimental setup of these tests are reported in Section 6.5, along with the obtained results.

The controller described in this Chapter was presented at the *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)* conference in June 2024, and it is the object of a publication in the *International Journal of Computer Assisted Radiology and Surgery (Scarponi et al. (2024))*.

6.1 Literature overview

Given the important advantages introduced by endovascular interventions, several research teams focused over the years on the development of strategies to optimize these interventions. An interesting approach was developed by *Schegg et al. (2022)*. In this work, the authors proposed a planning module responsible for the generation of feasible trajectories to reach the target location in coronary arteries. The method is based on a tree search combined with an inverse model obtained by convex optimization. Particular attention was paid to the modelization of the device and its contacts with the vessel walls, which were accounted for following Signorini's contact law. When tested on the planning of 70 different trajectories, the proposed algorithm was able to generate the correct trajectory in 90% of the cases. However, this system does not deal with the autonomous control of the device, but it only aims at developing a strategy to find the optimal path to follow to reach the target location in different scenarios, without trying to approximate the relationship between the applied control and the obtained device movement. Interesting steps toward this direction were made by the CASCADE project (2013-2016) *(Vander Poorten et al. (2016))*. This work focused on the development of a complete framework to autonomously steer catheters through the vasculature. In this project, all the aspects of the intervention have been considered, starting from the design of catheters to the vessel reconstruction, from the shape reconstruction to the decision-making and control of the instrument. Focusing the attention on the control algorithm, three versions have been developed. The first approach involved the control of only the distal part of the device. This section was treated as a continuum robot and modeled as a Cosserat rod. The Jacobian and the compliance matrix, which is the inverse of the stiffness matrix, were then computed. Starting from this information, position and force control schemes were derived. The control was formulated as a constraint-based approach, in which constraints on the position and pivoting were formulated to control the tip and the base of the distal part of the catheter, while for the force control, a velocity constant was imposed. The main drawback of this approach is that in order to apply it to the whole length of the catheter it is necessary to have a detailed representation of all the

interactions between the robot and the vessels, which is very complex. For this reason, for the control of the whole instrument and not only of its distal part, a minimum-energy approach was developed. This method was used to predict the new shape adopted by the catheter after a certain action was applied to it. The assumption behind this strategy is that the device will always assume configurations that minimize the combined deformation energy of the catheter and vessel walls. For this approach, a number of virtual inputs were applied to the device, and the corresponding catheter response was recorded. These data were then used to compute the kinematic of the object, used to compute the appropriate input to obtain the targeted output. The last approach presented in this project involved the use of machine-learning techniques, which were used to find the relationship between the input and the obtained catheter movement. The catheters were modeled with joint probability densities, choosing a Dynamic Gaussian Mixture Model. When challenged to follow the centerline of a vessel, the trained model was able to steer the catheter obtaining a mean Euclidean distance between the catheter tip and the centerline of 7.6 *mm*.

In this project, several techniques have been developed to approximate the relationship between the action applied to the device and its response to optimally control the instrument toward the target location. In recent years, with the explosion of deep reinforcement learning techniques in almost all engineering fields, many researchers have started to use DRL algorithms to automate endovascular interventions. The most common strategy involves feeding a convolutional neural network with 2D images of the device navigating the vessel structure and asking the controller to learn the control task based on this information. It is the case for example of the work developed by *Tian et al. (2023)*, which trained a Deep Deterministic Policy Gradient (DDPG) to obtain an autonomous navigation model. In this work, both the test and the training have been performed in the Cathsim simulated environment *(Jianu et al. (2023))*. The authors modified this framework to allow the simulation of passive devices, which can be controlled only from their proximal end. Two different algorithms were tested, a Proximal Policy Optimization (PPO) algorithm and a DDPG algorithm. Both algorithms were trained and tested on two different aortic arch types, obtaining a mean success rate of 70 % and 90% respectively, demonstrating the superiority of the DDPG algorithm for the navigation task. The main drawback of this work is that both the training and the testing are performed on the same anatomy, neglecting the generalization issue. *Wang et al. (2022)* proposed an autonomous guidewire delivery system, to simplify the work of the clinician when the navigation does not need complex control. In this work in particular they considered the navigation of the device from the iliac artery to a region close to the coronary ostium and for this purpose, a 3D phantom modeling these anatomical parts was employed. The system is based on the YOLOV5 deep learning algorithm, which, together with its previous versions, was specifically designed for real-time target detection

(Redmon et al. (2016); Redmon and Farhadi (2016)). The control algorithm takes as input images of the guidewire navigating the anatomy from multiple angles. The YOLOV5 algorithm, modified by the authors with an attention network, detects the position of the guidewire and the target location and the control algorithm decides which action to take to reach the target location. The trained model was tested feeding the network with new images of the guidewire navigating the phantom and in this case, the navigation was stopped when a region close to the heart was reached. The main limitation of this work lies once again in the use of the same model both for the training and for the testing of the algorithm, which strongly limits the application of the work. Following the same approach of training a deep reinforcement learning controller to navigate the surgical tools toward the target location with 2D images as inputs, *Karstensen et al. (2020)* developed a Deep Deterministic Policy Gradients algorithm with Hindsight Experience Replay (HER) for autonomous navigation in a 2D phantom. HER was developed to avoid the need to engineer complex reward functions thanks to sample-efficient learning from sparse and binary rewards *(Andrychowicz et al. (2017))*. The controller developed by *Karstensen et al. (2020)* takes as input the catheter state of the current and the previous three times steps, and the last three actions decided by the neural network. The most interesting part of this work is that while the training is performed in a simulated environment developed in SOFA, the tests are performed on a 3D-printed phantom. After 52500 training episodes, the algorithm reached a success rate of 96%. However, once again the training and testing are performed on the same geometry.

Besides the aforementioned limitations, another important drawback of these works is the need for continuous fluoroscopic images to perform the control task. This, besides exposing both the patient and the clinician to X-rays, can lead to ambiguities about the tip orientation, and therefore incorrect controls can be predicted. Furthermore, the learned control is also generally not compatible with a deformation of the anatomy. The development of neural controllers capable of realizing tasks not only in environments they have already explored but also in situations they have never seen before *(Kirk et al. (2023); Miranda et al. (2024))* remains an important challenge in DRL.

To address this issue in the context of endovascular interventions, *Kweon et al. (2021)* proposed a segment-wise learning approach to accelerate training using human demonstrations, transfer learning, and weight initialization. However, this approach still requires to train the network each time the environment is expanded or changed. In the work proposed by *Karstensen et al. (2023)*, the authors exploited recurrent neural networks to design a learning-based controller able to navigate various geometries. After 3.5 million exploration steps on various aortic arches, the controller reported a 75% success rate in navigating different aortic arches, which reduced to 29% when vessels from a real patient were used. Furthermore, the method requires

a different training for each vessel structure, which limits the applicability of the system when various structures need to be navigated. The same author proposed in 2022 a navigation system that was trained without human demonstration and compared to the movement performed by an expert user *(Karstensen et al. (2022))*. The model considered for the training and testing was the venous system of a porcine liver. The controller chosen for this work was a neural network trained using the DDPG algorithm and HER. The observation space was constituted by the current and the two previous guidewire positions, given as a set of 5 points along the device, the target location and the last two actions decided by the algorithm. When tested in the simulated environment in which the controller was trained, the algorithm showed a very high success rate (100%) and a behavior similar to the one shown by the clinician, which tends to continuously rotate clockwise and counterclockwise the device. The main limitation of this strategy is that when transferred to the testbench with the same ex-vivo liver, the performances of the controller dropped to 30%. An interesting work in the direction of transferring the control learned in a simulated environment to the real world, has been made by *Cho et al. (2022)*. In their work, they fed a neural network with 2D images of the guidewire navigating the anatomy. The peculiarity of the work lies in the training strategy developed. The controller was once again constituted by a DDPG network, but, instead of training the network from scratch, the controller was pre-trained through behavior cloning techniques. The expert data were collected by exploiting an expert algorithm developed by the authors in a previous work *(Cho et al. (2021))*. Once this phase was completed, the learning process was finalized by additionally training the total neural network with reinforcement learning in the virtual environment. This strategy allowed for a sharp reduction in the training time, making the system more efficient. The performances of the obtained controller were compared to the performances of the expert algorithm. Both the algorithms were required to navigate the training anatomy toward various target locations and successfully completed the task, but the trained controller achieved lower navigation time. Finally, the algorithm was tested on a real 3D-printed phantom presenting an anatomy with slight variations with respect to the training anatomy. The images of the phantom used as input of the controller were acquired through a camera placed on top of the phantom and also in this case the algorithm demonstrated its adaptability in the control task.
Another approach was developed by *Chi et al. (2020b)*. In their work, the authors implemented a generative adversarial imitation learning method aimed at learning to perform the catheterization of the brachiocephalic artery (BCA) and left common carotid artery (LCCA). In this work, they trained the network using expert demonstration on a reproduction of the aortic arch. On the training geometry, the system reported a success rate of 94.4% for the BCA cannulation and of 88.9% for the LCCA cannulation, but the performances were reduced to 69% for the BCA and 72.2% for the LCCA when the aortic type was changed. The same author proposed in a

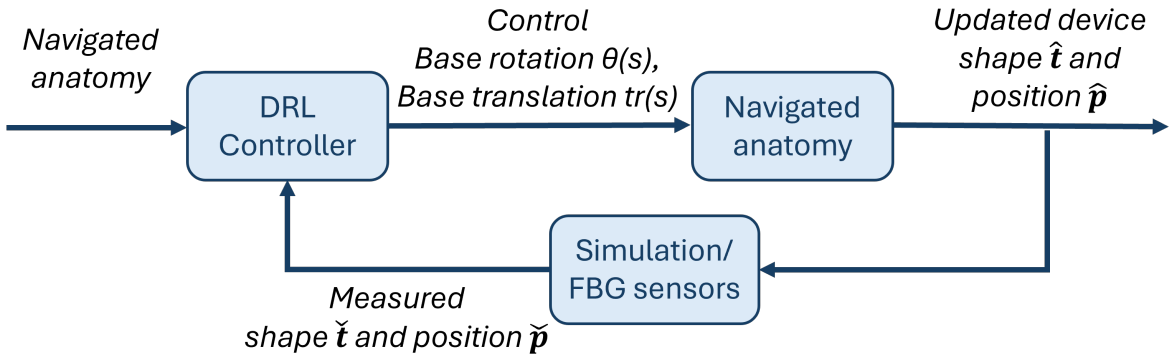

Fig. 6.1 Control loop specialized to the autonomous navigation task. The DRL controller takes as input two pieces of information: the anatomy that is being navigated and the shape $\breve{\mathbf{t}}$ and position $\breve{\mathbf{p}}$ of the devices. Given these two inputs, the DRL controller computes the rotation $\theta(s)$ and the translation tr to apply to the proximal end (base) of the device to navigate the desired branch. These controls will result in an updated device shape $\hat{\mathbf{t}}$ and position $\hat{\mathbf{p}}$, which will be retrieved from the simulation (simulated navigation) or measured by the FBG sensors (real-world scenario). Due to the imprecision of the measurement system, it is possible to measure $\breve{\mathbf{t}} \neq \hat{\mathbf{t}}$ and $\breve{\mathbf{p}} \neq \hat{\mathbf{p}}$.

previous work *(Chi et al. (2018))* a generalization method that did not involve the use of deep neural networks: using human demonstration, they trained a statistical model to perform the cannulation of the innominate aorta on a 3D model of the type I aortic arch and applied the same controller to variations of the same aortic arch type. This technique reported an average 98% success rate in new but very similar geometries, using human demonstrations for each new task.

All of these methods show a performance drop when tested on the new, unseen anatomies. Furthermore, these strategies were tested on different variations of the same anatomical structure, and do not provide a controller able to generalize the learned behavior on completely new vessel structures.

6.2 Setup of the DRL framework

In the study developed in this part of the work, the focus is on the navigation of a single device. To overcome the limitation of the works present in the literature, and in particular the lack of generalization capabilities and the use of fluoroscopic images as input of the system, we developed a DRL method that does not require the use of continuous fluoroscopic images and that is able to generalize the learned behavior to any vascular anatomy, assuming it is possible to navigate toward different target areas using the same guidewire tip shape. As a side

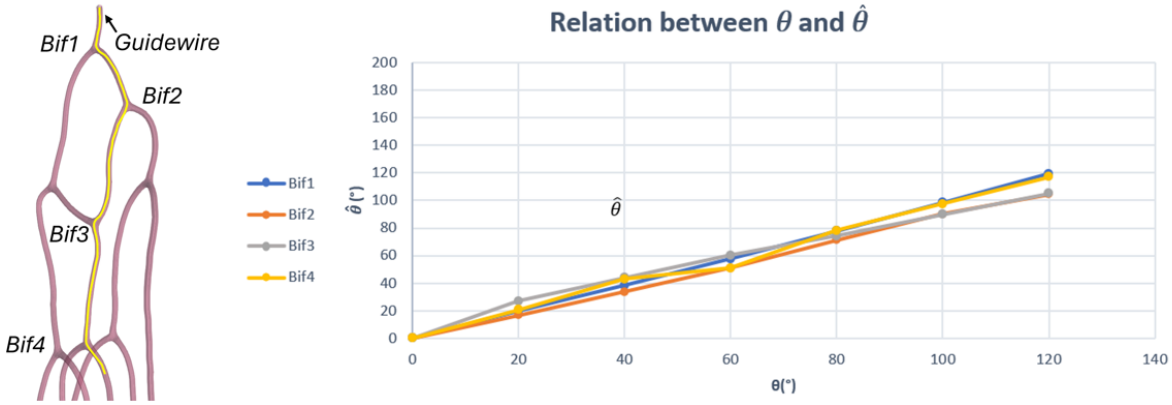

Fig. 6.2 Relationship between the rotation applied at the proximal part of the device and the obtained rotation at the tip when the guidewire is inserted in the anatomy shown in the left part of this image. The navigation was stopped after passing each of the 4 bifurcations and a total rotation θ of 120^o, divided into steps of 20^o, was applied to the proximal part of the device. For each of these steps, the obtained rotation at the tip $\hat{\theta}$ was measured and is represented in the graph on the right.

effect, our method is also able to handle shape variations due to physiological movements (e.g., breathing, cardiac motion).

The control algorithm developed in this work is represented by the control loop shown in Fig. 6.1. The core of the system lies in the DRL controller. Our control algorithm takes as input two pieces of information, which are the anatomy that is being navigated and the shape and position of the device inside the anatomy. As already detailed in Chapter 1, the necessary condition to navigate the desired branch that leads to the target is the ability to correctly orientate the tip of the device, which presents a particular shape. This desired tip direction is not directly given as input to the control algorithm, but it is the control algorithm itself that computes it, based on the anatomy that is being navigated, the position of the target, and the shape and position of the tip of the instrument. Given this information, the controller is able to compute the rotation $\theta(s)$ and the translation $tr(s)$ that needs to be applied to the devices in order to reach the target location. These actions are applied to the proximal part of the device and can result in a rotation $\hat{\theta}(s)$ and a translation $\hat{tr}(s)$ at the tip of the instrument that are different from the applied ones. The relationship between the rotation applied to the proximal part of the device and the rotation obtained at the tip is complex and depends on many different factors, which include, besides the applied rotation, the length of insertion of the guidewire inside the vessel, the curvature of the anatomy and the number of contacts. Before starting to develop the control algorithm described in this work, this relationship was studied by simulating in the SOFA environment the insertion of the guidewire inside the vessel

model shown in the left part of Fig 6.2. In this case, no friction was considered between the devices and the vessel wall. The navigation was stopped after each of the four bifurcations and a rotation θ of 120^o, divided into steps of 20^o, was applied at the proximal part of the device. At each step of the rotation, the obtained rotation at the tip of the device $\hat{\theta}$ was measured. The obtained graph is shown in the right part of Fig. 6.2. As can be evinced from this graph, based on the state of constraint that the vessel imposes on the instrument, the rotation obtained at the tip varies.

The measurement of this quantity in a simulated environment is straightforward and it can be measured from the parameters of the simulation itself, along with the device position $\check{\mathbf{p}}$ and shape $\check{\mathbf{t}}$. When switching to the real world, to avoid ambiguities due to 2D shape reconstruction from single-plane fluoroscopic images, we propose to rely on FBG shape-sensing technology to obtain the device shape $\check{\mathbf{t}}$ and position $\check{\mathbf{p}}$. However, as introduced in Chapter 4, these measurements are affected by errors that can lead to a measured shape $\check{\mathbf{t}} \neq \hat{\mathbf{t}}$ and to a measured position $\check{\mathbf{p}} \neq \hat{\mathbf{p}}$. This can cause problems in the control of the instrument, which must be robust to these uncertainties. However, since this initial phase has been implemented in a simulated environment, we will consider $\check{\mathbf{t}} = \hat{\mathbf{t}}$ and $\check{\mathbf{p}} = \hat{\mathbf{p}}$.

The development of the control algorithm required some design and implementation choices. As already detailed in Chapter 3, SOFA framework has been used to design the simulated environment as it provides reliable, real-time physical simulations and customizable components. Another fundamental choice involved the selection of the most appropriate deep reinforcement learning algorithm. Since many libraries provide ready-to-use codes of the most common DRL agents, to avoid the risk behind the implementation of a custom version of the algorithm, an already available code was adopted, which required the choice of the proper framework.

6.2.1 The choice of Soft Actor-Critic algorithm

In recent years, the increasing success of DRL led to the development of more advanced and efficient algorithms, each with specific characteristics that make them particularly well-suited for certain applications. The most classical DRL algorithm is the Deep Q-network (DQN) algorithm, which is the DRL version of the Q-learning approach, which builds a memory table $Q[s,a]$ which stores all possible combinations of states s and actions a. When the number of combinations is too high, building a table containing all possible combinations becomes prohibitive. To solve this issue, the table $Q[s,a]$ is approximated by a neural network, called DQN. This algorithm is very easy to implement, it improves sample efficiency thanks to the use of a replay buffer and it handles high dimensional spaces. However, it suffers from an overestimation of the Q-values, leading to potential instability, and it also demonstrates low generalization capabilities even in very similar environments *(Farebrother et al. (2018))*.

In 2016, the Asynchronous Advantage Actor-Critic (A3C) *(Mnih et al. (2016))* and its synchronous version, Advantage Actor-Critic (A2C) achieved great success thanks to their promise of training deep neural network policies reliably and without requiring a large amount of resources. In particular, A3C allows to asynchronously execute multiple agents in parallel, on multiple instances of the environment. The update scheme of these algorithms operates on fixed-length segments of experience and computes the estimated returns and advantage functions based on these experiences. The reduction in training time is roughly linear to the number of instances that run in parallel. The architecture of the neural network shares layers between the policy and the value function. Researchers did not find any evidence that the noise introduced by asynchrony provides any benefit in terms of performance and developed, therefore, the synchronous and deterministic version of this algorithm, the A2C. This algorithm presents the same benefits and structure of the A3C but waits for every actor to finish its experience segment before performing an update and averaging over all the actors. A2C generally requires a lot of data to learn efficiently and the performances of the algorithm tend to deteriorate when generalization is required *(Packer et al. (2018))*.
In 2017, the Proximal Policy Optimization (PPO) algorithm was developed *(Schulman et al. (2017))*. This algorithm and its variations showed better performances with respect to previous algorithms such as the A2C *(Schulman et al. (2017))* and were therefore largely adopted in several control tasks. PPO algorithm is easy to implement and shows very good stability. It is an online algorithm and it does not use a buffer, which means that it learns only from the current experience. The aim of this strategy is to avoid a too important difference between the current and the previous policy update, but it can result in low sample efficiency.
However, when dealing with autonomous control of catheters and guidewires, the literature research reported in Section 6.1 shows a predominant use of the Deep Deterministic Policy Gradient. This algorithm is an actor-critic algorithm and it constitutes a variant of the DPG algorithm (Section 2.2.2). It uses a Q-function estimator to enable off-policy learning and an actor to maximize this Q-function. It therefore incorporates aspects of both a deterministic actor-critic algorithm and a DQN. The interplay between these two networks makes the algorithm very difficult to stabilize *(Duan et al. (2016))*. In 2018, a new actor-critic algorithm, which takes the name of Soft Actor-Critic (SAC, *Haarnoja et al. (2018a)*), outperformed DDPG algorithm in several tests *(Haarnoja et al. (2018b))*. Given these promising characteristics, SAC algorithm has been adopted to develop the controller designed in this work.
SAC algorithm combines off-policy actor-critic training with a stochastic actor and aims to maximize the entropy of this actor with an entropy maximization objective. The SAC algorithm

optimizes the following objective function:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t) + \alpha\mathscr{H}(\pi(\cdot|s_t))] \tag{6.1}$$

in which $\rho_\pi(s_t)$ and $\rho_\pi(s_t,a_t)$ denote the state and state-action marginals of the trajectory distribution induced by a policy $\pi(a_t|s_t)$. With respect to the usual formulation of RL's objective functions, SAC introduces the entropy term $\mathscr{H}(\pi(\cdot|s_t))$, weighted by the temperature weight α. The entropy $\mathscr{H}$ encourages the exploration of the environment, avoiding repeatedly choosing an action that may exploit inconsistencies in the approximated Q-function. By choosing an appropriate value of α, it is possible to help the agent generalize its learned policy. Another important aspect of this algorithm is its ability to support continuous action spaces.
To optimize the objective function shown in Eq. (6.1), SAC uses three networks: a state value function V, a soft Q-function Q and a policy function π. The authors explain that having two separate networks for V and Q helps the convergence *(Haarnoja et al. (2018a))*. Each of the three networks aims to minimize a different error.
For the value network V, the function to be minimized is represented by:

$$J_v(\psi) = \mathbb{E}_{s_t\sim\mathfrak{D}}\left[\frac{1}{2}(V_\psi(s_t) - \mathbb{E}_{a_t\sim\pi_\phi}[Q_\theta(s_t,a_t) - \log\pi_\phi(a_t|s_t)])^2\right] \tag{6.2}$$

This function aims at minimizing the difference between the prediction of the V network and the expected prediction of the Q function, plus the entropy of the policy function π, which is represented by the negative logarithm of the policy.
The Q network tries to minimize:

$$J_Q(\varsigma) = \mathbb{E}_{(s_t,a_t)\sim\mathfrak{D}}\left[\frac{1}{2}\left(Q_\varsigma(s_t,a_t) - \hat{Q}(s_t,a_t)\right)^2\right] \tag{6.3}$$

with $\hat{Q}(s_t,a_t) = r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p}[V\overline{\psi}(s_{t+1})]$, where $\overline{\psi}(s_{t+1})$ is the target value function. This error aims to minimize the squared difference between the prediction of the Q function and the immediate reward plus the discounted expected value of the next state for all state-action pairs.
Finally, the policy is chosen to minimize the following:

$$J_\pi(\Upsilon) = \mathbb{E}_{s_t\sim\mathfrak{D},\varepsilon_t\sim\mathfrak{N}}[\log\pi_\Upsilon(f_\Upsilon(\varepsilon_t;s_t|s_t) - Q_\Upsilon(s_t,f_\Upsilon(\varepsilon_t;s_t))] \tag{6.4}$$

in which the authors used the reparameterization technique $a_t = f(\varepsilon_t; s_t)$ to make sure sampling from the policy is differentiable. Here, ε_t is an input noise. This function aims to make the policy distribution as similar as possible to the distribution of the Q function 6.4.

6.2.2 The choice of stable-baselines3 library

Different implementations of the most common DRL algorithms are available online. The two most complete libraries found during the development of this work, are RLlib *(Liang et al. (2018))* and stable-baselines3 *(Raffin et al. (2021))*. Both libraries are well-documented and are open-source, allowing the user to customize the components. RLlib was mainly developed for industrial production, it presents a comprehensive codebase and it is regularly updated. It supports several environments including Gymnasium ones. Gymnasium *(Towers et al. (2024))* is a standard API to communicate between learning algorithms and environments. While being comprehensive, it can sometimes be difficult to integrate custom applications in RLlib given the extension of the library, making it less suitable for research purposes. In this work, stable-baselines3 has been adopted as the code-base for the implementation of the SAC algorithm. This library is currently maintained, easily customizable and well-suited for research purposes. It provides a series of already implemented environments, but it also allows the user to easily integrate his own custom environment, which, to this purpose must follow the Gymnasium interface.

6.2.3 Implementation details

The use of stable-baselines3 library required a specific implementation of the code for this project, to be able to interface the simulated environment built in SOFA with the SAC agent available in stable-baselines3 library. The code has been therefore structured as follows. A script *scene.py* has been prepared to initialize all the simulation parameters and all the needed libraries. All the parameters of the simulation, from the mechanical properties of the instruments to the navigated anatomies, from the training parameters to the parameters of the neural network, are defined in *.json* files: *agentsConfig.json* for the agent, *rlConfig.json* for the action and observation space parameters, *bodyConfig.json* for the navigated anatomy and *sceneConfig.json* for the device's parameters. *scene.py* calls a python class, *Gym_env*, which, to comply with stable-baselines3 requirements, creates the SOFA environment following the structure of Gymnasium environments. This class handles the creation of the SOFA scene, the initialization of all the objects included in it and the initialization of the action and observation space. Finally it implements a function, *step*, required by Gymnasium API. This function handles the application of the action decided by the agent to the SOFA environment, calls the

simulation steps performed in SOFA and computes the new observation and reward function resulting from the action decided by the agent. It is in the definition of the appropriate observation space and reward function that lies the real heart of the method developed in this work and these aspects will be therefore discussed in Section 6.3. After the creation of the environment, *scene.py* calls another class, *SAC_agent*, which initializes the algorithm that will handle the training of SAC agent.
This particular configuration of the SOFA environment, which bypasses the usual internal management of SOFA simulation steps, does not allow the use of the built-in SOFA GUI during the training and testing of the algorithm. To overcome this limitation, a tool internally developed in the team, *SimulationSimpleDatabase*, was used. The use of this library allows to visualise, in real-time, the training and testing episodes of the algorithm, allowing to better inspect the performances of the algorithm.

6.3 Zero-shot reinforcement learning strategy

As anticipated in the previous section, the real core and novelty of the method developed in this work lies in the design of the reward function and observation space of the DRL algorithm. The current implementation of these elements was developed after a deep study of the problem, which was performed through a naive control algorithm not based on machine learning techniques, but only on geometrical considerations. In this first implementation of the work, three main assumptions were made: the vessels have a nearly constant radius, the anatomy is not moving or deforming during navigation and the bifurcations always have 2 exit vessels. This pattern was chosen as it is the most common in real vessels *(Singh et al. (2017))*.

6.3.1 Geometrical control algorithm

As already explained in Chapter 1, during the navigation, the guidewire is controlled by means of two actions, translation and rotation and when reaching a bifurcation, it is important to be able to correctly orientate the tip in order to navigate the desired branch. To do this, the first step is to compute the needed rotation at the tip ($\hat{\theta}$) and then compute which is the rotation θ to apply at the basis of the catheter to obtain the desired rotation at the tip. The computation of $\hat{\theta}$ involves four steps:

1. Definition of the path that connects the insertion point to the target location. To this purpose, the centerline of the anatomy is extracted using a *skeletonization* algorithm *(Gao et al. (2024); Tagliasacchi et al. (2012))*. The path to follow is defined starting from

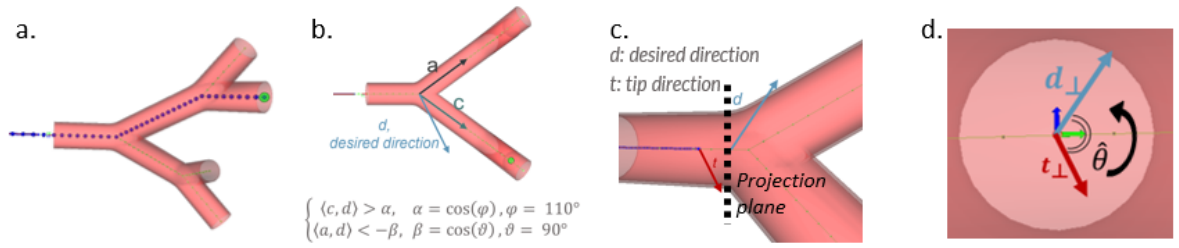

Fig. 6.3 a. Path definition b. Desired Direction definition c. & d. Vector projection.

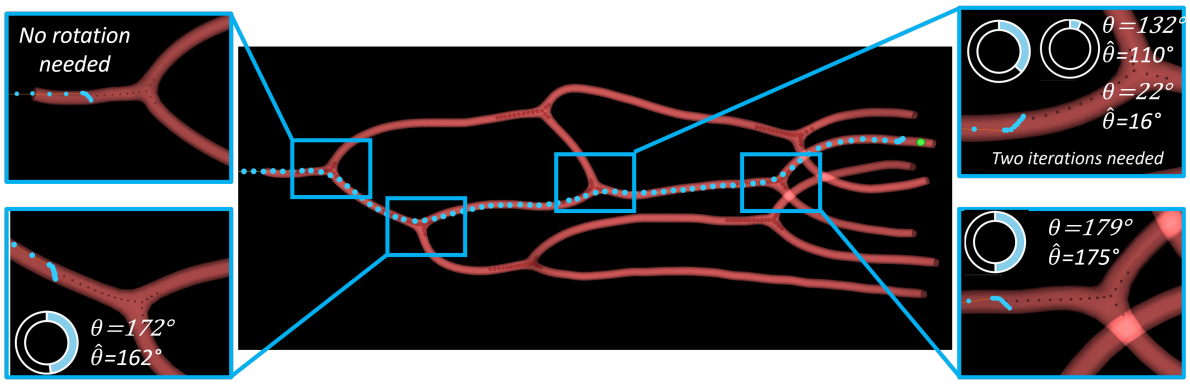

Fig. 6.4 Guidewire navigation in the simulated environment. At each bifurcation, the actions (*i.e.,* rotation) taken by the control algorithm are displayed as a circular diagram. From this image it can be observed how $\hat{\theta}$ is always smaller than θ.

the target location and navigating each bifurcation backward until reaching the insertion point (Fig. 6.3 a)

2. Definition of the desired tip direction **d**. This desired tip direction is defined solving an optimization problem that identifies a direction **d** as close as possible to the direction of the branch to be taken **c** and as far as possible from the direction of the undesired branches **a** (Fig. 6.3 b). This problem can be expressed as:

$$\text{Find } \mathbf{d} \text{ such that } \begin{cases} \mathbf{c}\cdot\mathbf{d} > \cos(110^o) \\ \mathbf{a}\cdot\mathbf{d} < -\cos(90^o) \end{cases} \tag{6.5}$$

3. Computation of $\hat{\theta}$: both the desired direction **d** and the tip direction **t** are projected onto a plane perpendicular to the centerline in a region immediately before the bifurcation (Figure 6.3 c). The angle between these two projections is then computed (Figure 6.3 d)

4. Definition of the direction of the rotation - clockwise or counterclockwise (Figure 6.3 d)

Given the complex $\theta - \hat{\theta}$ relationship (Section 6.2), the rotation applied at the proximal end can be different from the rotation obtained at the tip. For this reason, an error, defined as the difference between the desired direction and the direction of the tip after the rotation, is computed. If this error is lower than an ε, then the algorithm starts again to push forward the device. Otherwise, the second point of the process is repeated until the achievement of the correct orientation. The value of ε, equal to 20°, was experimentally defined. The algorithm was tested on several rigid anatomies with various subsequent bifurcations, proving its ability to navigate the guidewire toward the target location. The same unmodified algorithm was also used on a deformable vessel geometry, subjected to small deformations. Also in this scenario, the navigation was successful. This result can be explained thanks to the way the desired tip direction **d** is computed, which is robust to small differences between the navigated anatomy and the geometry used as input of the system: if the deformation is not too pronounced, then the algorithm can be effective, even without updating the geometry. An example of navigation is shown in Figure 6.4.
The analyses conducted to develop this trivial geometric algorithm and the obtained preliminary results set the foundation for the development of the learning strategy developed in this work.

6.3.2 A generalizable DRL controller

As already mentioned, our principal goal is to develop a deep reinforcement learning able to generalize to all the anatomies which is possible to navigate with a specific guidewire. To achieve this goal, the definition of the observation space, and selection of the training geometries, will prove essential to learn a generalizable control. Unlike previous works, we do not train our network on the test anatomy. Instead, we train it on a set of bifurcation patterns, not directly related to the test anatomies. We process the 3D shapes and compute their centerlines for both the test anatomy and training shapes. This computation is done automatically from the 3D vascular geometry. Then, at the beginning of each training episode, a target is chosen among a set of possible final states, typically (but not necessarily) placed near the end of each terminating branch. Then, a unique path connecting the starting point to the target location is computed from the centerlines of the training anatomy with the same technique described in Section 6.3.1. For each episode, the starting rotation of the guidewire around its axis and its orientation with respect to the centerline are chosen randomly to enhance the exploration of the environment. The controller is set to push the device at a constant speed during the whole episode and the agent can impose in a single time step a rotation $-30^{o} < \theta < 30^{o}$. The parameters of the DRL algorithm and details on the choice of the training geometries will be given in the following.

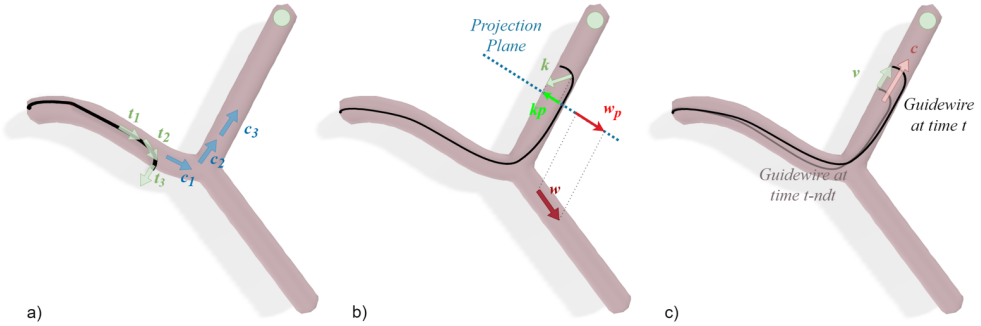

Fig. 6.5 Our observation space is composed of: 1) the scalar product between t_i, describing the tip, and c_i, describing the centerline: $t_i \cdot c_j$, with $i \in [1;3] \in \mathbb{Z}$ and $j \in [1;3] \in \mathbb{Z}$ (a), 2) the normalized distance between the tip of the guidewire and the target, 3) the chosen action, 4) the scalar product between a vector describing the curvature of the tip of the instrument (k_p) and the projection of the direction of the branch not leading to the target, on a plane Γ perpendicular to the centerline: $k_p \cdot w_p$ (b) 5) the scalar product between the velocity of the instrument v and a vector c describing the centerline: $v \cdot c_i$ (c).

Nearly shape-invariant observation space

To enforce the generalization of the learned control, we designed an observation space that is rotation and translation invariant but also shows little sensitivity to the shape variation of the bifurcation. The observation space allows the agent to build its internal representation of the state, which the controller uses to make its decision about the best action. By letting the agent build this representation from observations that describe its *relative* position to the environment, we can make it robust to a variety of transformations of the bifurcation shape. To this end, we propose to construct the observation space Ω as follows:

$$\Omega = \{\zeta_t, \zeta_{t-ndt}, \lambda_t, \lambda_{t-ndt}, a_t, \omega, d_v\}$$

- Let $\mathbf{t}_i, i \in [1,N]$ be the tangent vector at the coordinate $\mathbf{x}_i$ along the tip of the guidewire, and $\mathbf{c}_j, j \in [1,N]$ the tangent vector of the centerline at position $\mathbf{x}_j = \mathbf{x}_i + \mathbf{h}$. We define $\zeta_i = \mathbf{t_i} \cdot \mathbf{c_i} \; \forall \; i \in [1,N] \in \mathbb{Z}$.
- We then define $\zeta_m = [\zeta_1, \zeta_2, ..., \zeta_N]_m$, with $m \in \{t; t - ndt\}$.
- λ_t and λ_{t-ndt} represent the geodesic distance between the tip of the guidewire and the target at time t and $t - ndt$
- a_t is the action that determines the transition of the system from s_{t-ndt} to s_t

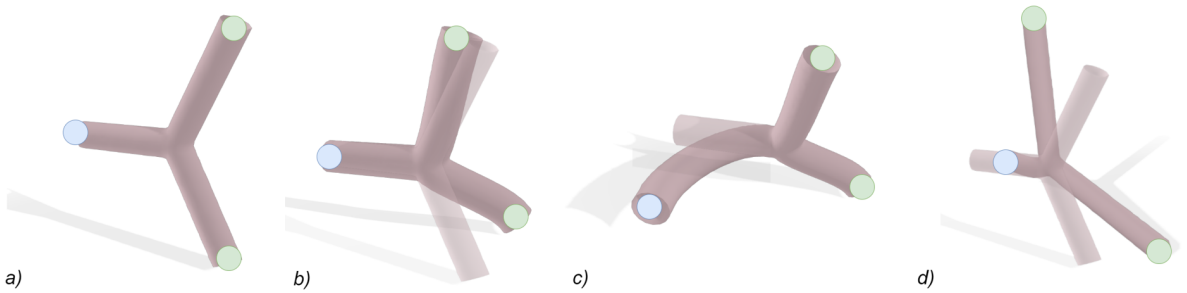

Fig. 6.6 Geometries used to study the sensitivity of our training with respect to changes in position (a), orientation (a) and shape (b,c,d). The blue dot marks the entry branch, the green ones the exit.

- $\omega = \mathbf{k}_p \cdot \mathbf{w}_p$, where $\mathbf{k}_p$ is the projection of the curvature vector $\mathbf{k}$ of the tip and $\mathbf{w}_p$ is the projection of the vector describing the direction of the wrong branch $\mathbf{w}$ onto a plane perpendicular to the centerline of the branch leading to the target (Fig. 6.5b).
- $d_v = \mathbf{v} \cdot \mathbf{c}$, where $\mathbf{v}$ describes the current velocity of the guidewire and $\mathbf{c}$ the tangent to the centerline near the tip of the guidewire (Fig. 6.5c).

The metrics composing this observation space have been chosen thanks to the analysis conducted during the development of the geometrical control algorithm (Section 6.3.1). It is important to notice that all the parameters used to build the observation space can be computed in both the virtual (training) environment and in a real setup. In the simulated environment, the parameters regarding the shape and position of the tip can be extracted from the parameters of the simulation. Since to apply the controller in the real world we will adopt FBGs, to reduce the effect of reconstruction errors, we rely on multiple grating measurements to define the tip orientation. As explained in Section 4.2.2, we designed a fiber with a high density of DTGs at the tip and we define the tip shape through a series of vectors ($\mathbf{t}_i$, $i \in [1,N] \in \mathbb{Z}$), each of them relying on the measure of multiple DTGs.

Another key element to learning the optimal action is the engineering of the reward function. We design our reward function as the weighted sum of three terms (Eq. (6.6)):

$$r(s_t, a_t) = \underbrace{\frac{2}{1+e^{5(\omega-0.1)}} - 1}_{a} + \underbrace{0.5\left(1 - \frac{\lambda_t}{\lambda_0}\right)}_{b} + \underbrace{(-0.2|a_t|)}_{c} \tag{6.6}$$

where λ_t and λ_0 are respectively the current and the initial distance between the tip and the target. a encourages the agent to obtain a tip direction ($\mathbf{k}_p$, Figure 6.5) opposite to $\mathbf{w}_p$. This function is a modified version of the sigmoid activation function, largely used in the deep learning field. In this modified version, the output γ is a decreasing function with

$\gamma \in [-1;1] \in \mathbb{R}$. The sigmoid function and its steepness have been chosen to reward the agent with a gradient that increases as it approaches optimal behavior. The final coefficients of this term were fine-tuned to ensure the best agent behavior. It is important to mention that the design of this term was inspired by the desired tip direction **d** computed in Eq. (6.5), which also tried to maximize the angle between the tip orientation and the direction of the wrong branch. The term b increases as the target is approached and it increases the reward of the agent when the actions lead to the navigation of the correct branch, and it further penalizes it when the wrong branch is taken. Finally, the term c discourages the agent from rotating the instrument when it is unnecessary.

Design of the training anatomies

To choose a proper dataset for the training, we studied the sensitivity of our method to the orientation, the position and the shape of the geometry. We trained our network on a single bifurcation (Fig. 6.6 a) and tested it in 5 variants of this bifurcation: a rotation of the model, a translation, a variation (var_1) of the shape of the exit branches (Fig. 6.6 b), a variation (var_2) of both the exit and entry branch shapes (Fig. 6.6 c) and an important variation (var_3) of the exit branches shape (Fig. 6.6 d).

Once trained, we first assessed that the agent was able to get a 100% success rate in the training anatomy. Then, the agent was required to perform the navigation 50 times in each of the 5 scenarios: 3 shape variations, rotation and translation of the training shape. The success rates were 100% (rotation), 100% (translation), 100% (var_1), 62% (var_2) and 86% (var_3). The high success rate reported by the controller when variations of the exit branch occur (var_1), demonstrates the robustness of our training to this parameter. However, when the differences become too important (var_3), a performance drop is reported. The biggest performance deterioration is however experienced when navigating var_2 anatomy, which reports a small variation of the exit vessels and an important variation of the entry branch, showing that our training is mostly sensitive to variations of the initial branch.

Therefore, and as we will illustrate in section 6.4, we can already anticipate that with only a few training shapes, we can maximize the chances of generalizing the learning to various complete anatomies.

6.4 Results

This section presents the training dataset (section 6.4.1) constituting the environment in which the SAC agent learned to control the guidewire following our training strategy. The controller was then used to navigate various complex unseen anatomies (section 6.4.2) to test its

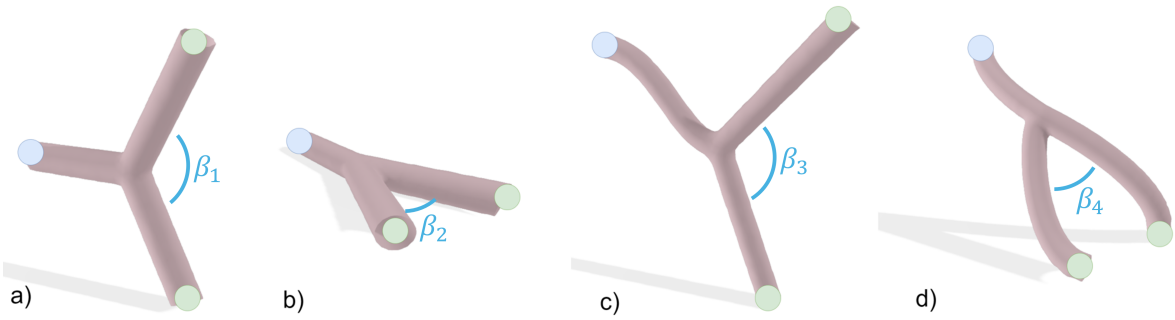

Fig. 6.7 Geometries chosen to train the agent and to cover the whole observation space. In each model, the blue point represents the starting point, and the green one the target location. β_i, $i \in [1,4]$ shows examples of the angles between the two exit vessels.

generalization capabilities. The guidewire selected for this work presents a 6 *mm* curved tip characterized by a curvature of 0.2 mm^{-1} and a total length of 310 *mm*. Both the training and the tests were performed on a computer equipped with 32 GB of RAM, a 24-core Intel(R) Core(TM) i7 running at 3.40 GHz and an NVIDIA GeForce GTX 1080 GPU.

6.4.1 Training anatomies

For the purpose of this work, a training dataset composed of only 4 constant radius anatomies was designed (Fig. 6.7). This was identified as the minimum dataset to respect the criteria described in section 6.3.2: each geometry presents different characteristics in terms of either the shape of the access vessel or the shape of the exit branch, allowing the agent to explore the shape variation observed in real vascular trees. Using this dataset, the agent was trained until the convergence, obtained after 120,000 time steps. Both the actor and the critic networks are composed of three 256-neuron layers. After tuning the hyperparameters, a learning rate of 10^{-4}, a batch size of 256, a buffer size of 10,000 and a discount factor of 0.98 were set, while the entropy coefficient was learned autonomously during the training. 1,000 steps were performed before starting the training of the model, which was updated at every time step.

At the beginning of each episode, one of the 4 anatomies was randomly chosen, along with the target location (green points in Fig. 6.7) and the guidewire was placed at the entrance of the selected vessel (blue points in Fig. 6.7) with a random orientation. The time required to complete the training was about 2 hours. The control learned by the controller was first tested on the training anatomies, under the same conditions used for the training itself. The navigation was performed on a total of 100 episodes and the controller obtained a success rate of 95%.

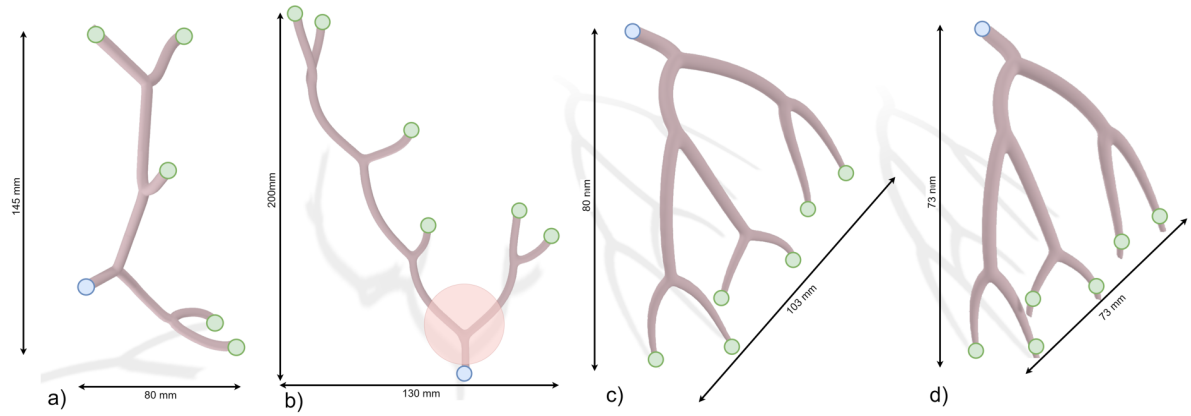

Fig. 6.8 Geometries used to test the algorithm. The blue dots represent the insertion points, while the green dots show all possible target locations. The red circle shows an example of a bifurcation region.

6.4.2 Navigation test on complex vascular trees

To test the efficacy of our method, the trained controller was used to navigate complex anatomies composed of various connected bifurcations. To this purpose, three different geometries were considered. The first anatomy was obtained by composing the training geometries (Fig. 6.8a) and aimed at demonstrating that learning on a single set of bifurcations can lead to a correct control on a combination of these bifurcations. The second anatomy has a constant vessel diameter, and contains up to 4 subsequent unseen bifurcations (Fig. 6.8b), while the third one is a reconstruction of the left coronary arteries, characterized by unseen bifurcation shapes and variable vessel radii (Fig. 6.8c). For each anatomy, the controller was tested for 50 episodes, each starting with the guidewire placed at the insertion point of the anatomy (blue points in Fig. 6.8) with a random initial orientation. An episode was considered finished either when the target was reached (green points in Fig. 6.8, *successful episode*) or when, navigating a

Table 6.1 Success rate obtained by the controller in the navigation of unseen anatomies. For each of these anatomies, the controller was required to navigate the geometry from the insertion point to the target location, which was randomly chosen at each episode. A total of 50 episodes per anatomy was performed.

Geometry	Success rate
Training geometries composition	98%
Unseen bifurcation composition	100%
Coronaries-like anatomy	94%
Deformed coronaries-like anatomy	90%

Table 6.2 Earth mover's distance between each bifurcation in the test anatomies (Fig. 6.8 b, c and d) and the most similar training bifurcation.

	Anatomy 1b	***Anatomy 1c***	***Anatomy 1d***
Bifurcation 1	0.54	1.21	1.12
Bifurcation 2	0.54	0.76	1.13
Bifurcation 3	1.06	0.78	1.44
Bifurcation 4	1.84	0.80	1.04
Bifurcation 5	1.62	1.09	0.6
Mean	1.04 ± 0.38		

bifurcation, the wrong branch was chosen (*failure*). In these complex anatomies, the control of the guidewire was entrusted to the agent only in the proximity of the bifurcation (*i.e.*, a region comprising an area of radius 20 *mm* and centered at the center of the bifurcation): indeed, outside these regions, the control is trivial and only consists of pushing forward the device. Every time a bifurcation was reached, an intermediate target was inserted in the branch that needed to be navigated to reach the final target. The results of these tests are reported in Table 6.1. In the anatomy composed of the training geometries, the controller obtained a success rate of 98%. The success rate in the geometry composed of unseen bifurcations was 100%, and 94% in the coronaries-like anatomy.

We then performed an additional test, on a deformed version of the coronary arteries (Fig. 6.8 d). This deformation corresponds to the maximum deformation amplitude observed during a cardiac motion. On that new unseen anatomy, our neural controller succeeded 45 times out of 50 trials to reach random targets(i.e. a 90% success rate).

Testing anatomies analysis

As an attempt to quantify how different the test anatomies are from the training set, we chose the earth mover's distance (EMD) which is a measure of dissimilarity between two shapes. For two identical shapes, this metric is equal to 0. Considering now two bifurcation shapes with a common entry vessel and with an angle between the two exit vessels (see β_i in Fig. 6.7 for a representation of this angle) of respectively 30° and 60°, the EMD is 0.82. In other words, a variation in bifurcation angle of 100% (which is highly significant) leads to an EMD of 0.82. To compute the EMD, each bifurcation of the test anatomy has been compared to the four training anatomies and to do that, the bifurcations have been reconstructed in the same reference system aligning the center of the two bifurcations and the entry vessels as accurately as possible. Table 6.2 reports the minimum EMD between each bifurcation in the test anatomies (Fig 6.8 b, c

and d) and the training bifurcations, giving, therefore, a measure of the difference between each bifurcation of the test anatomies and the most similar training anatomy. As can be observed from the table, all bifurcation shapes are highly different from the training dataset, with EMD values ranging from 0.54 to 1.84.

6.5 Open loop control

The simulated environment used in this work allows to accurately simulate the physics of real catheters and guidewires. Motivated by previous works *(Cho et al. (2022); Karstensen et al. (2020))* that used a similar virtual environment and successfully transferred the learned control to navigating a vascular phantom, we tested the algorithm described in the previous section in the real world.
This was possible thanks to the collaboration with LN Robotics (Seoul, Korea), a company producing robots for percutaneous coronary interventions. The great advantage of this system is its ability to control up to 4 simultaneous tools: this allows to support also complex percutaneous coronary interventions, that require switching to manual mode when using the principal competitor's robots. In addition, it also supports over-the-wire balloon catheters for chronic total occlusion procedures, catheter extension for patients with severe calcification, and additional tools for challenging situations. Similarly to other endovascular intervention robots, also this device operates as a leader-follower system, controlled by the clinician by means of a joystick (Fig. 6.9 a). The instruments are controlled through the mechanism shown in Fig. 6.9 b: a total of 8 cylinders, composing 6 roller blocks, move the instruments in response to the control imposed by the clinician. When the instruments need to be advanced, the roller blocks start to rotate around their axis, while when a rotation is required, the roller blocks displace vertically, either upward or downward. The movements of the cylinders are effectively transferred to the devices thanks to the silicon-like material used as a coating. In the current implementation, both the translation and the rotation of the instruments are speed-controlled. To actuate the robot imposing the action decided by our algorithm, the joystick needs to be excluded. For this reason, LN Robotics provided us with custom-made software integrated into the computer of the robot, which allows to control the movement of the roller blocks via UDP communication. The messages sent to the robot include several encoded information:

$$a/b/c/d_1d_2d_2/e_1e_2e_3$$

- a indicates the number of the message

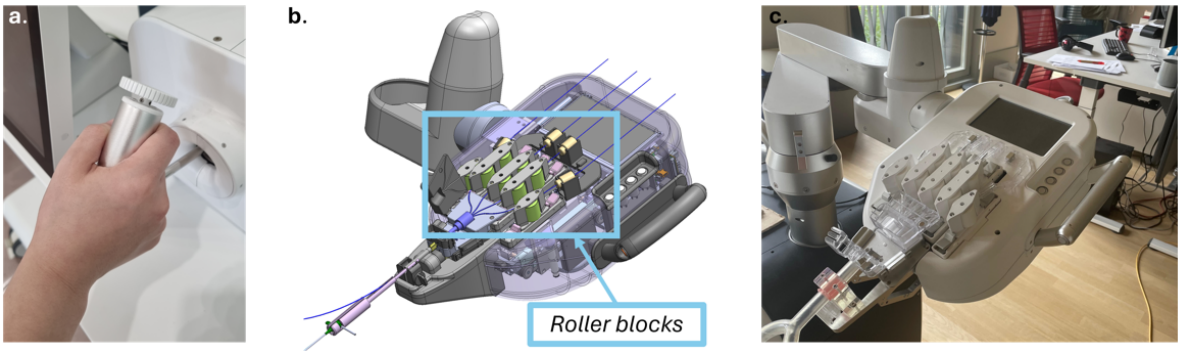

Fig. 6.9 a. Leader device to control the robotic device b. Design of the robotic follower device and of the roller blocks, the main actuator of the surgical devices. c. LN Robotics follower systems available in the laboratory.

- b allows to select which roller block will be activated, and therefore allows to select the channel in which the instrument to manipulate is inserted
- c is used to send the command to clamp the instrument inserted in channel b
- d_1 allows to move the instrument forward ($d_1 = 1$) or backward ($d_1 = 2$) at the velocity of $d_2 d_3$ $\frac{mm}{s}$, which ranges from 1 to 15 $\frac{mm}{s}$
- e_1 allows to rotate the instrument clockwise ($e_1 = 1$) or counterclockwise ($e_1 = 2$) at the velocity of $e_2 e_3 \times 10\,^o/s$, which ranges from 0 to 360 $^o/s$

It is then this software that handles the communication with the robot and translates the control command into a command that the system can apply.
The same software also collects information from the robot, encoded in a string as follows:

$$a/b/c_1c_2c_3c_4/d_1d_2/e_1e_2$$

- a indicates the number of the message
- b communicates on which channel the command is applied
- $c_1c_2c_3c_4$ indicates whether the instrument inserted in channel b is being clamped or released
- d_1d_2 indicated the displacement of the device in *mm*
- e_1e_2 indicates the displacement of the cylinders, which is a measure of the rotation applied to the instrument, in *mm*

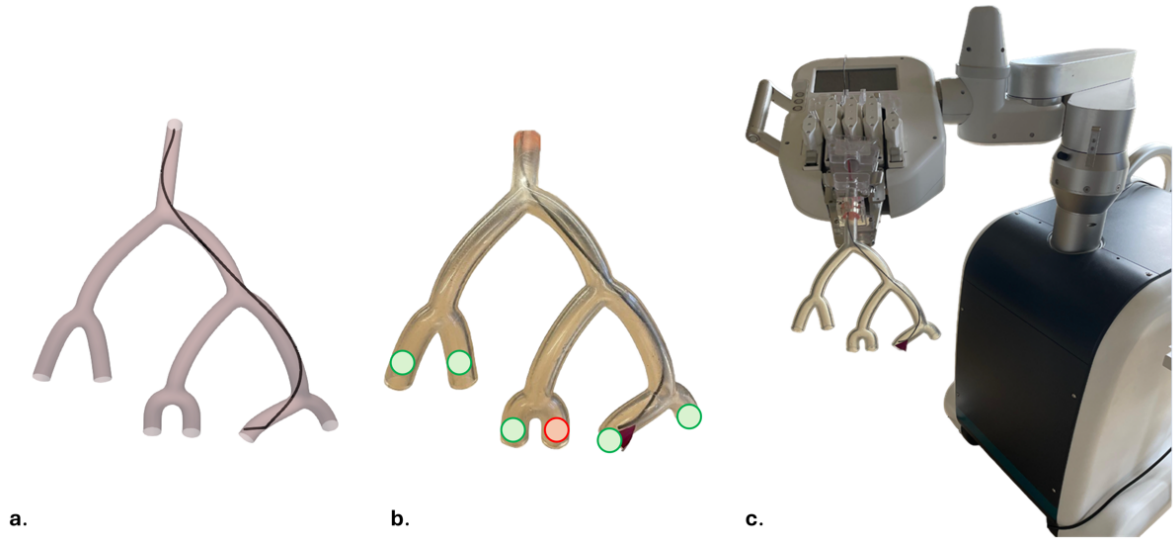

Fig. 6.10 Representation of the phantom geometry navigated by the catheter in the simulated environment (a) and in the real world (b,c). As can be observed from Fig a and b, the shape of the catheter modeled in the simulated environment is comparable to the shape assumed by the catheter in the real world. Fig c shows the complete setup, with the catheter navigated by the robot. The dots in Fig. b show the target location: the reached target are shown in green, while the failed attempt in red.

To generate the commands that will be sent to the robot, the algorithm and the strategy described in Section 6.3.2 have been used, including the training geometries and the guidewire. To test the model in the real world, a new test geometry was designed and 3D printed. The 3D printed model can be observed in Fig. 6.10 a. Also in this case the model was designed to present bifurcation substantially different from the training anatomies. Before 3D printing the model, a thickness of 3 *mm* was added around all the vessels. The obtained phantom is shown in the central part of Fig. 6.10.
To navigate this phantom, the Merit Impress Angiographic Catheter Amir Motarjeme has been chosen. This catheter presents a pre-curved tip, has a diameter of 5 *F* and a length of 100 *cm*. Using the described anatomical model and catheter, the algorithm was required to navigate this new anatomy toward all 6 exits, and the decided actions were registered in a *.txt* file. In this first implementation, the algorithm pushes the device forward at a constant speed and the controller handles the rotation of the instrument in regions close to the bifurcations, imposing a rotation of a certain angle. As already mentioned, the robot can be controlled by imposing the speed of translation and the speed of rotation of the instrument. The actions decided by the control algorithm were therefore converted to match the robot's command format. The control applied in this case, is an open loop control: at the beginning of each navigation, it was therefore important to position the catheter with the same translation and orientation as

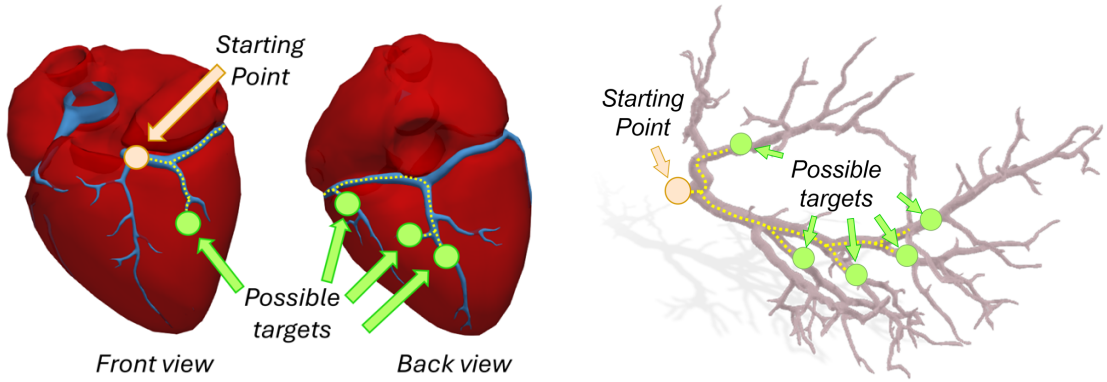

Fig. 6.11 Illustration of the different paths and targets the neural controller needs to navigate in the heart (left) and in the liver (right). The insertion point is shown in orange, and the randomly selected targets are shown in green. An example of a path is depicted as a dotted line.

the simulated one. Applying the command generated by our DRL controller in the simulated environment, the robot was able to reach 5 out of the six total target locations. The successfully reached targets are shown in Fig. 6.10 in green, while the failed case is depicted in red. In this unsuccessful scenario, the robot, controlled by the action decided by our algorithm, was able to correctly navigate the first two bifurcations and failed when the most distal one was reached.

This test proves the accuracy of the physical model developed in this work and the robustness of the control, which has the potential to be successfully transferred to the real world. However, the failed case shows that some additional work needs to be done to better parametrize the physical properties of the instrument and that a system to *close the control loop* is necessary.

6.6 Final considerations & current limitations

In this work, we developed an efficient method to train a neural controller able to perform navigation tasks on unseen complex anatomies, after being trained on a set of only 4 bifurcation geometries. We obtained an average 95% success rate when navigating new anatomies in the simulated environment, outperforming previous works that reported a success rate of 72% *(Chi et al. (2020b))* and 75% *(Karstensen et al. (2023))* when trying to generalize the task to unseen models. Our algorithm also showed promising results when transferring the learned control to the real world, in an open-loop system.

Given these encouraging results, we decided to expand this work and try to navigate more complex and realistic anatomies, both in static and dynamic conditions. Indeed the strongest limitations of this work are the synthetic nature of the test anatomies and the static conditions. We therefore challenged our algorithm to navigate two different anatomical models,

Table 6.3 Navigation results summary, in heart and liver, both in static and dynamic conditions.

Heart	
Conditions	Success rate [%]
Static[a]	100%
Dynamic	68%
Liver	
Conditions	Success rate [%]
Static[a]	36%
Dynamic	50%

the coronary arteries reconstructed from the MRI images of a real heart, and the venous system of the liver. The first one is a typical example of endovascular navigation, while the second is common in the diagnosis and treatment of portal hypertension. For the dynamic case, the movement of the beating heart was synthetically generated using the *InfinyToolkit* SOFA plugin, while the movement of the liver was reconstructed starting from fluoroscopic images, with a method that will be exposed in Section 7.2. For each of the two anatomies, the algorithm, trained as explained in Section 6.3, was required to perform the navigation from the insertion point until the target location a total of 100 times. For each episode, a different target was randomly selected among the 4 and 5 targets chosen in the heart and in the liver respectively (Figure 6.11). For the navigation of the coronary anatomies, since the dimensions of the vessel are compatible with the dimensions of the training anatomies, the same guidewire used during training was employed. The entry vessel of the liver, on the other hand, has a diameter that is double the diameter of the training anatomy: to perform the navigation in this geometry, it was, therefore, necessary to adopt a guidewire with a $6.5\,mm$-long tip and a $0.26\,mm^{-1}$ tip curvature.

The success rates obtained by our controller in these conditions are reported in Table 6.3. In the static coronary geometry, our controller was able to reach the target location in all considered cases, even if this anatomy was completely different from the geometries experienced during the training. This result demonstrates the robustness of our approach to geometrical variations. However, we observed an important performance drop in the navigation of the vessels of the beating heart. Considering now the liver, both in the static and in the dynamic case the controller failed to reach many target locations.
There are three main reasons underlying this behavior:

1. The controller has no information regarding the movement of the anatomy. This explains the important performance deterioration observed in the navigation of the moving

coronaries. Indeed, the method developed in this work does not provide any instrument to update the geometry of the vessels when a movement or a deformation occurs. This means that also the observation space remains unchanged, introducing differences between the geometry observed by the controller and the real geometry.

2. Even if the observation space was updated, the controller has never experienced moving anatomies during the training.

3. The dimension of the vessels. The controller developed in this work is able to successfully navigate the static coronaries because they respect the dimensions of the training anatomies. The important diameter difference in the liver geometry places the controller in an entirely new situation, which it cannot handle, resulting in a very low success rate.

Given these limitations, the work presented in this chapter was extended to try to overcome these drawbacks. The strategy that we developed is presented in the next chapter.

Chapter 7

Autonomous Guidewire Navigation in Dynamic Environments

As illustrated in Section 6.3, the algorithm presented in the previous chapter introduced three main assumptions: nearly constant vessel radius, branching patterns with one entry vessel and two exit vessels and static anatomies. The first assumption is not a limitation of the method but a consequence of using a unique guidewire during the navigation. With a constant tip shape, only vessels of a compatible diameter can be accessed. Furthermore, as highlighted in the previous chapter, bifurcation patterns are the most common *(Singh et al. (2017))*. However, these assumptions limit the applicability of our controller to the navigation of static anatomies, which present a vessel diameter similar to the vessel diameter of the training anatomies (Section 6.6).

This chapter presents the adopted strategies to overcome these limitations. In particular, we propose a training strategy to learn a control of the device generalizable to anatomies with different vessel dimensions and effective also when the vascular anatomy is moving and/or deforming (see Section 7.1). To this purpose, it is fundamental to be able to reconstruct the movement and deformation of the anatomies. The proposed controller is therefore integrated with the work developed by François Lecomte, a member of MIMESIS team. His strategy permits the estimation of the motion of the anatomy from single-view fluoroscopy images. For the sake of completeness, this method will be described in Section 7.2. The combination of the two proposed techniques makes it possible to automatically navigate across a moving anatomy under fluoroscopic imaging, even without injecting a contrast agent. Our results (see Section 7.3) illustrate the genericity of the training, and the good performance of our method, even when applied to complex, deforming anatomies only observed through 2D fluoroscopic imaging.

The controller described in this Chapter was presented at the *IEEE/RSJ International Conference on Intelligent Robots and Systems* in October 2024.

7.1 Guidewire control learning in dynamic environments

Our objective is to develop a generalized neural controller able to control the motion, in particular the rotation, of a guidewire through a complex, deforming vascular tree, from its insertion point until a given target is reached (see Fig. 7.6 right, 7.7 right), while it is advanced at a variable speed. This control is performed at the proximal end of the device, and accounts for both the device and anatomy deformations during navigation. Our learning method relies on five main elements:

1. An efficient DRL algorithm
2. A fast and accurate simulated environment that can be updated based on external input
3. An observation space robust to affine transformations of the anatomy
4. A specific reward function
5. An optimal choice of training anatomies.

Concerning the first two points, the same elements as in the first version of the algorithm were adopted. Given its promising characteristics and the encouraging results obtained on static anatomies with the previous controller, SAC algorithm has been chosen once again as DRL algorithm (Section 6.2.1). For the simulated environment, encouraged by the transferability demonstrated by the previous version of the controller, the same setting and characteristics have been maintained in this part of the work, with the sole exception of the used guidewire, which was chosen based on the navigated anatomies.
The main novelties introduced in this work, lie in the reinforcement learning strategy and in the design of the observation space. Details on these aspects will be provided in the following.

7.1.1 Reinforcement learning strategy

As illustrated in Chapter 6, the definition of the observation space, and choice of the training geometries, are essential to learn a generalizable control. In this work, we keep a similar strategy: we train the RL algorithm on a set of bifurcation patterns, unrelated to the test anatomies. This local vascular shape is represented by both a surface mesh and a centerline. The only assumption about the training shapes is that the diameter of the vessels is nearly

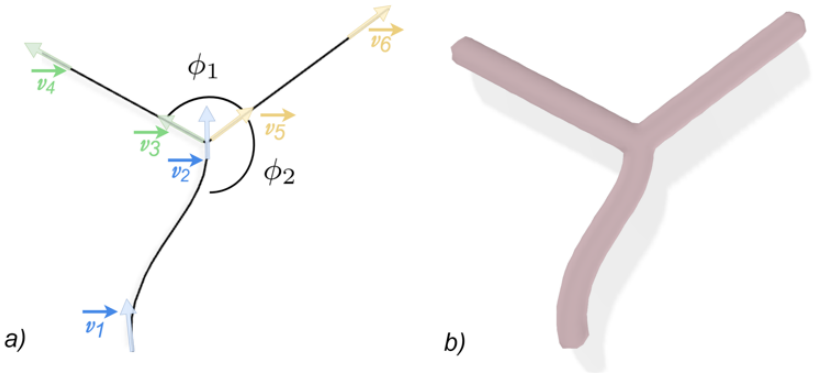

Fig. 7.1 Procedural training shape generation process. a) Centerline $C(\phi_i, v_j)$ of the vessel with $i \in \{1,2\}$ and $j \in \{1,..,6\}$. ϕ_1 and ϕ_2 are the angles between the bifurcation branches and v_j are the tangents to the centerline shape at each endpoint. b) 3D shape of the vessel obtained by extrusion of the centerlines C.

constant and that they have a Y-shaped topology. During the advancement of the device, the controller can impose a rotation $-30^o < \theta < 30^o$ every time the guidewire advances by 3 *mm*.

We then extend our previous work in two areas. First, we augment the training database by introducing shape variations of the training anatomy during the training process (i.e. similar to sim-to-real approaches). This shape variation is continuous throughout space and time, to avoid discontinuities in the displacement field that would cause problems in the simulation. Second, we formalize the shape generation process by making it procedural, rather than handcrafted. We characterize the 3D vessel shape from its centerline C from which it is extruded. C is defined as $C(\phi_i, v_j)$ with $i \in \{1,2\}$ and $j \in \{1,..,6\}$, where ϕ_1 and ϕ_2 define the angles between the bifurcation branches and v_j are the tangents of the centerline shape at each endpoint (see Fig. 7.1). Starting from the simplest geometry (a Y-shaped bifurcation with straight branches) we progressively deform this shape into a series of other shapes, by varying smoothly ϕ_i and v_j and maintaining a constant vessel diameter. Fig. 7.2 illustrates this process.

Let's call $\mathscr{B}$ the set of all the bifurcation shapes we generate through our process. We split $\mathscr{B}$ into a series of N subsets $\mathscr{B}_k$ of random length, such that $\cup_{k=1}^{N} \mathscr{B}_k = \mathscr{B}$. Each subset $\mathscr{B}_k$ represents a different range of shape variations, from small deformations to large ones. These shape variations are then used as training anatomies during the learning processes of our neural controller. For each training episode, a target is randomly selected, as well as a subset $\mathscr{B}_k$ of the varying training anatomy. The use of these subsets of shape variations during the training is crucial to allow the controller to experience anatomical movements in the learning process and recognize familiar situations while navigating new unseen anatomies subjected to unexplored movements.

The initial rotation of the guidewire around its axis and its orientation relative to the centerline

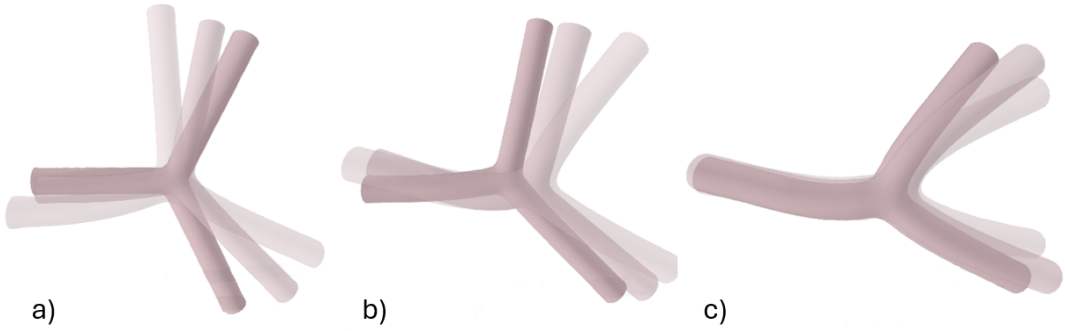

Fig. 7.2 Examples of $\mathscr{B}_k$ subsets. Each subset $\mathscr{B}_k$ represents a range of bifurcation shape variation and $\cup_{k=1}^{N}\mathscr{B}_k = \mathscr{B}$, where $\mathscr{B}$ defines the complete set of bifurcation shapes considered in this work.

are chosen randomly to enhance the exploration of the environment and, during each episode, the velocity of the device is also randomly modified. This velocity variation is introduced to once again help the agent observe the broadest possible range of situations during the training. During the navigation of moving anatomies, when the length of insertion is important it is indeed possible that even if the proximal end of the guidewire is pushed with a constant velocity, the tip of the device recedes due to the movement of the vessels. Modifying the speed of the guidewire during the training allows the simulation of these situations.

Nearly shape-invariant observation space

To enforce the generalization of the learned control, we propose as in the previous work (Section 6.3) an observation space that is rotation and translation invariant, but also shows little sensitivity to the shape variation of the bifurcation. This is achieved by defining observations that are relative to the position of the device in the environment. In this work, we expand the observation space by adding elements that permit to navigate geometries that are different both in shape and size, with the sole caution of using a guidewire compatible with the vessel diameter. The observation space Ω is constructed as follows:

$$\Omega = \{\zeta_t, \zeta_{t-ndt}, \lambda_t, \lambda_{t-ndt}, a_t, \omega, d_v\}$$

- Let $\mathbf{t}_i, i \in \{1,...,N\}$ be the tangent vector at the coordinate $\mathbf{x}_i$ along the tip of the guidewire, and $\mathbf{c}_j, j \in \{1,...,N\}$ the tangent vector of the centerline at position $\mathbf{x}_j = \mathbf{x}_i + \mathbf{h}$. We define $\zeta_i = \mathbf{t_i} \cdot \mathbf{c_i}\ \forall\, i \in \{1,...,N\}$. To handle dynamic environments, $\mathbf{c_i}$ must be updated. This does not require changing the observation space defined in the previous chapter but

necessitates estimating this change from live images during an intervention. The strategy to handle this challenge is described in Section 7.2.

- We then define $\zeta_m = [\zeta_1, \zeta_2, ..., \zeta_N]_m$, with $m \in \{t; t - ndt\}$.
- λ_t and λ_{t-ndt} represent the distance between the tip of the guidewire and the target at time t and $t - ndt$, normalized with respect to the initial distance to the target λ_0. λ_0 is defined as the target distance at the entrance of the bifurcation region (*i.e.,* a region comprising an area of radius 20 mm and centered at the center of the bifurcation).
- a_t is the action that determines the transition of the system from s_{t-ndt} to s_t.
- $\omega = \mathbf{k}_p \cdot \mathbf{w}_p$, where $\mathbf{k}_p$ and $\mathbf{w}_p$ are the projections of the vectors $\mathbf{k}$ and $\mathbf{w}$ onto a plane Γ perpendicular to the centerline of the branch leading to the target (see Fig. 7.3 b). $\mathbf{k}$ represents the radial vector of curvature located in the middle of the curved tip, and $\mathbf{w}$ is the vector describing the direction of the wrong branch. To be robust to different vessel dimensions, both in terms of vessel diameter and exit branch length, the proper choice of $\mathbf{w}$ and Γ normal vector (n_Γ) is crucial and has therefore been tuned in this new version of the controller. $\mathbf{w}$ norm is here proportional to the vessel diameter and its starting point is fixed at the center of the bifurcation, while n_Γ magnitude is chosen proportional to the squared diameter of the vessel and it originates from the projection of the guidewire distal end onto the centerline (see Fig. 7.3 b).
- $d_v = \mathbf{v} \cdot \mathbf{c}$, where $\mathbf{v}$ describes the current velocity of the guidewire inside the vessels and $\mathbf{c}$ the tangent to the centerline near the tip of the guidewire (see Fig. 7.3 c).

It is important to notice that once again all the parameters used to build the observation space can be computed in both the virtual (training) environment and in a real setup. The vessel geometry can be retrieved from preoperative images and updated intra-operatively (see Section 7.2) and the tip shape of the guidewire can be reconstructed from FBG data embedded in the catheter or guidewire.

Reward function

Another key element to learning the optimal action is the engineering of the reward function. Since the objective of the control did not change, the same reward function as the previous version of the algorithm was maintained:

$$r(s_t, a_t) = \underbrace{\frac{2}{1 + e^{5(\omega - 0.1)}} - 1}_{a} + \underbrace{0.5\,(1 - \lambda_t)}_{b} + \underbrace{(-0.2|a_t|)}_{c}, \tag{7.1}$$

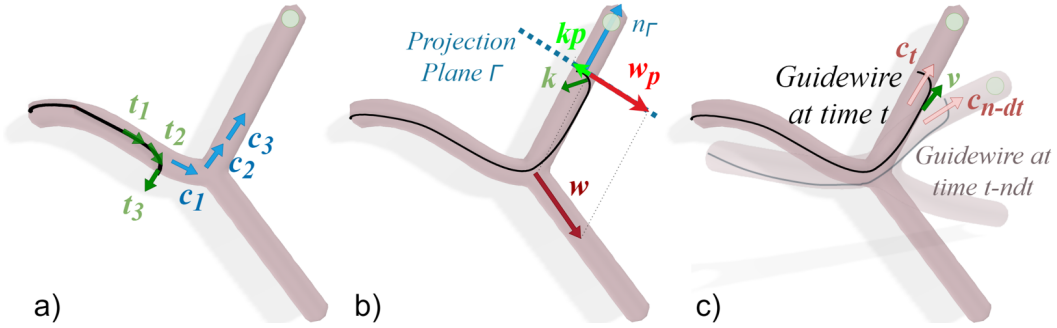

Fig. 7.3 Observation space: 1) dot product between $\mathbf{t}_i$ (tangents to the tip of the device) and $\mathbf{c}_j$, describing the downstream centerline, with $i \in \{1,...,3\}$ and $j \in \{1,..,3\}$ (a), 2) normalized distance between the tip of the guidewire and the target, 3) chosen action, 4) dot product between $\mathbf{k}_p$, describing tip's direction, and $\mathbf{w}_p$, describing the direction of the branch that does not lead to the target (b) 5) dot product between $\mathbf{v}$, describing the velocity of the guidewire and $\mathbf{c}$, describing the centerline (c).

where part a of the reward function encourages the agent to obtain a tip direction $\mathbf{k}_p$ opposite to $\mathbf{w}_p$ (see Fig. 7.3). Part b of the reward increases as the target is approached, while part c discourages the agent from rotating the instrument when it is unnecessary.

7.2 3D vascular motion estimation from fluoroscopic images

The observation space Ω described previously includes the relative position of the device tip with respect to the vessel centerline and target, among other things. When the shape and position of the vessels are changing (e.g. due to cardiac or respiratory motion) we must update this information such that the neural controller can perform optimally.

To recover this motion, it is necessary to use a real-time imaging modality that presents sufficient contrast between the vessels and the surrounding tissue. Fluoroscopy is the only imaging modality that meets these criteria and is currently used in the vast majority of endovascular interventions. However, it requires the injection of a contrast agent to be able to visualize vessels in the image, and it provides only a two-dimensional image.

Various methods have been devised to overcome this limitation and recover 3D motion from a single fluoroscopy, mostly in the context of free-breathing radiotherapy *(Nakao et al. (2022); Shao et al. (2023); Wei et al. (2020))*. While these methods demonstrate clinically relevant target localization accuracy, their use of a statistical motion model to generate training data limits them to recovering predetermined motion patterns, which can restrict their clinical applicability. Moreover, the accuracy of these methods to recover the shape of vessels has not yet been evaluated.

7.2.1 Fluoroscopy-based vessel motion prediction

To remediate the lack of methods suited for vessel shape estimation, the work by *Lecomte et al. (2023)* in the Mimesis team presented a method to recover, in real-time, the shape of the vessels from a single fluoroscopy. In addition, our method predicts the 3D displacement of the vascular network without requiring the injection of a contrast agent, which is a significant benefit for the patient. Even though fluoroscopy is a 2D projective imaging modality, it is still possible to recover 3D motion from fluoroscopy thanks to the penetrating nature of X-rays. A fluoroscopic image is formed by measuring rays' attenuation from the X-ray source through matter to the detector. This method, like the works cited above, uses a preoperative CT image as a prior on the patient's anatomy. The intra-operative fluoroscopic image is then processed to extract the motion of the anatomy relative to this preoperative CT scan. The fluoroscopic image formation process can be approximated by Eq. 7.2 below:

$$p(\mathbf{u}) \approx \int I(\mathbf{x}) \mathbf{dl_u}, \tag{7.2}$$

where p is the fluoroscopic image, $\mathbf{u}$ the pixel coordinate, I the CT scan of the patient, $\mathbf{x}$ the voxel coordinate along the projection ray, and l_u the projection ray from the X-ray source to the pixel u. Consequently, a deformation of I that produces a change in the value $p(\mathbf{u})$ will be visible in the fluoroscopic image. Only deformations of I that leave $p(\mathbf{u})$ unchanged, that is motion colinear to the projection ray l_u, cannot be observed in the fluoroscopic image.

This method seamlessly integrates into any fluoroscopy-guided intervention with routine pre-operative CT scan acquisition. Using the pre-operative CT scan of the patient, the vessels and other structures of interest are segmented and the planning of the intervention is performed by the clinician. In the current approach, it is assumed that the pose of the fluoroscopic imager with respect to the patient's anatomy is determined before the intervention, which is generally true.

7.2.2 Training data generation

The objective of the neural network is to predict a 3D displacement field from a single fluoroscopic image in which the anatomy is viewed in a deformed configuration (see Fig. 7.4). Since such data is impossible to obtain from actual interventions, both a 3D Displacement Vector Field (DVF) and the associated deformed fluoroscopic image, obtained as a Digitally Reconstructed Radiograph (DRR) have been generated. These DRR and DVF pairs are computed from the preoperative CT image of the patient, assuming a known pose of the C-arm. To enable the network to recover arbitrary deformations of the anatomy, the DVFs are generated in a

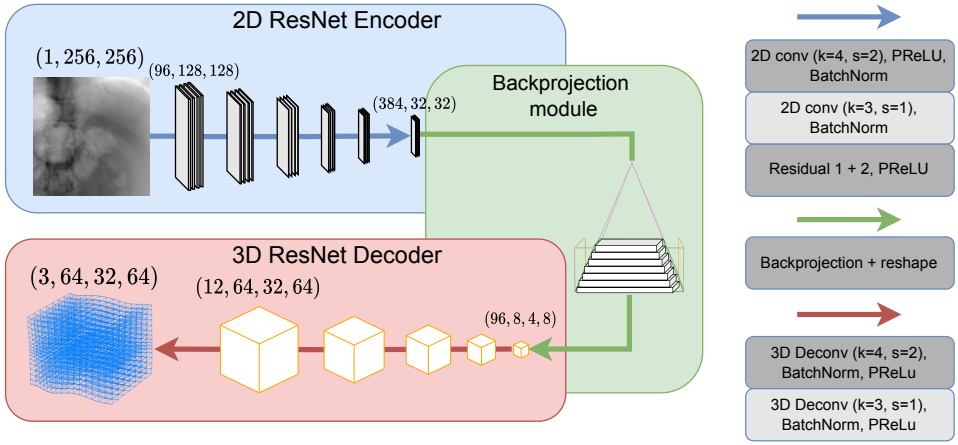

Fig. 7.4 Each block in the Encoder downscales the feature maps and increases their number by a factor of 2. In the Decoder, this is reversed. The last decoder layer transforms the 12 feature maps into a 3-channel 3D image.

randomized way. For the deformation to be realistic, the DVF must be diffeomorphic, meaning that no self-intersection occurs in the deformation. The Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework, which is tailored to meet these conditions, emerges as a promising solution *(Trouve et al. (2005))*. Within this framework, the DVF ϕ that registers a deformed image with a reference image is obtained by integrating a set of differential equations in time. At each step of the integration, the parameters of a velocity field $\mathbf{V}(t,\mathbf{x})$ are updated to minimize both an energy-based term and an image similarity term.

Trouve et al. (2005) show that the velocity field $\mathbf{V}(t,\mathbf{x})$ can be constructed as a sum of Gaussian kernels, each centered at control points $\mathbf{y_k}$ and associated with corresponding weights α_k:

$$\mathbf{V}(t,\mathbf{x}) = \sum_{k=1}^{N_{cp}} \alpha_k(t) \cdot \mathbf{K_k}(\mathbf{x}, \mathbf{y_k}(\mathbf{t})).$$

Here, $\mathbf{K_k}(t)$ denotes elements from a Reproducing Kernel Hilbert Space, in this case, Gaussian kernels. The control points $\mathbf{y_k}$, covariance matrices $\sigma_k \in \mathbb{R}^{3\times3}$, and weights α_k of the Gaussian kernels are randomized in our approach to compute a displacement field $\varphi(\mathbf{x})$, that does not depend on time. N_y control points $\mathbf{y_k}$ are sampled by generating random points in the volume, with a rejection process to avoid generating control points that are too close to each other, which would result in sharp variations of φ. Since α_k are independent of each other, a sample will typically contain a mixture of small, medium, and large displacements. However, to be robust to scenarios where no deformations occur, we randomly scale the DVF with a global scaling parameter chosen between -1 and 1. Additionally, covariance matrices σ_k are generated as $N_y \times 3 \times 3$ i.i.d. variables within a specified range. Finally, one DRR is generated

for each deformation of the preoperative CT using the DeepDRR framework *(Unberath et al. (2018))*.

The fully convolutional network architecture (detailed in Fig. 7.4) presented in this work is based on the architecture described in *Lecomte et al. (2023)*'s previous work, with some modifications. First, a convolutional encoder extracts a condensed feature representation from the 2D input image. The number of convolutional filters in the first layer is increased from 64 to 96, which slightly improved performances. Then, the 2D feature maps are transformed into 3D feature maps using back-projection instead of a simple reshaping as in *Lecomte et al. (2023)*'s previous work. The back-projection algorithm uses the C-arm camera matrix to transform the 2D feature maps in the 2D fluoroscopic image space to 3D feature maps in the preoperative 3D CT space. Owing to the local nature of convolutions, the feature maps output by the encoder are in the same space as the input image. The feature maps F extracted by the encoder are treated as a single-channel 3D image, where the original channel dimension of F corresponds to a third spatial dimension, corresponding to the depth along projection rays. This is motivated by the fact that the input image is formed by measuring the attenuation of rays traveling from the projection center to each pixel on a 2D detector, following Eq. 7.2. To transform the feature maps into the same space as the pre-operative CT scan, the projection matrix $\mathbf{P}$ is used. This matrix is formed by the intrinsic parameters of the C-arm and the C-arm pose with respect to the CT scan. First, a regular grid of voxel coordinates $\mathbf{G}$ in the 3D CT scan space is defined. Then $\mathbf{G}$ is converted to a grid of pixel coordinates $\mathbf{G_u} = \mathbf{PG}$ in 2D homogeneous coordinates u_i. u_0 and u_1 represent the height and width of the projection of the voxel on the detector plane, while u_2 represents the depth of the voxel along the projection ray. Since convolutions are local operations, F can be indexed in the same way as the input image, by normalizing G_u to the range $[-1,1]$. By interpolating F using G_u, a 3D volume F_{3D} in the same space as the preoperative CT scan is obtained. This process is described in Fig. 7.5, where F is back-projected to obtain F_{3D}. F_{3D} is non-zero only in the regions that are both in the CT volume and visible in the projection, tracing a square frustum.

F_{3D} is a three-dimensional image without a feature dimension. To transform F_{3D} into a 3D feature map, a feature dimension is created by splitting the depth dimension into a depth and a feature dimension. This choice is motivated by the fact that the depth dimension was originally the channel dimension of F before the back-projection process. Finally, F_{3D} is decoded into a 3D displacement field by a fully convolutional 3D decoder. This displacement field represents the non-rigid transform of the CT to the anatomy visible in the fluoroscopic image. The network is trained using an MSE Loss between the synthetic ground truth displacement field and the predicted displacement field.

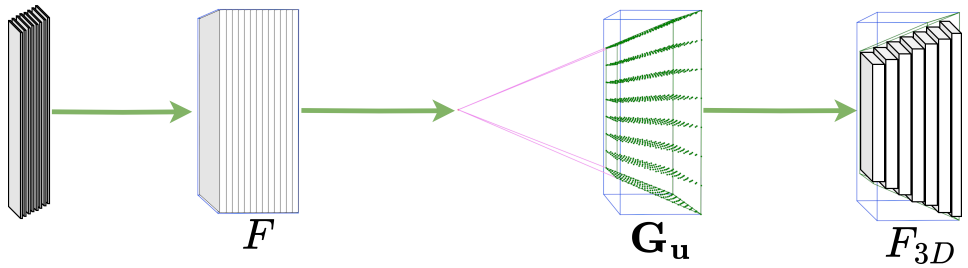

Fig. 7.5 The 2D feature maps are reshaped into a volume F in ray space. The volume is then sampled using ray space coordinates $\mathbf{G_u}$, in green, to obtain a volume F_{3D} in CT space. Elements of F_{3D} that are outside of the view frustum are set to zero.

Since the neural network receives a single 2D fluoroscopic image as input, the 3D motion estimation task is inherently an ill-posed problem. In particular, the 3D motion component in the direction of the projection rays cannot be observed in the image. According to Eq. 7.2, any motion along the projection rays l_u will not induce a change in image intensity. Indeed, the value of the line integral along l_u is not modified by a voxel displacement along p_u. To train the network, it was necessary to compute the MSE Loss only on the prediction components parallel to the image plane. This limits the ability of the network to predict the motion of the anatomy perpendicular to the image plane and induces errors in the prediction of the displacement field. As detailed below, our controller is, however, robust to this uncertainty.

7.3 Results

We test the performance of our new neural controller and fluoroscopy-based motion estimation in the same conditions in which our previous controller failed, *i.e.,* the moving coronary of a beating heart and the venous system of the liver subjected to breathing motion. Based on the method described in Section 7.1, we train the neural controller using a learning rate of 10^{-4}, a buffer size of 10,000 and a batch size of 256. The discount factor is set to 0.98 and the entropy coefficient is learned during the training. The actor and the critic networks are composed of three 256-neuron layers and the model, updated at every time step, is trained for 175,000 time steps, with a dt of $0.01\,s$. The whole training process only requires 6 hours of computation on an Intel(R) Core(TM) i7-13700KF processor with 32 GB of RAM. The training anatomies are generated as explained in Section 7.1 with a constant vessel diameter of $4\,mm$. A suitable guidewire is used to navigate the anatomies, with a $4.5\,mm$ long tip and a tip curvature of $0.38\,mm^{-1}$. For each anatomy and test case, the controller is evaluated on a total of 100 episodes, where an episode is defined as the navigation from the insertion point to the target location. An episode is considered successful if the guidewire, steered by the controller,

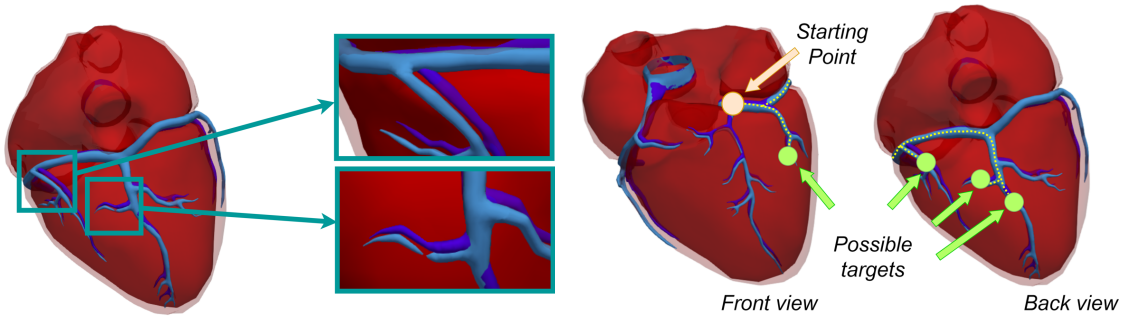

Fig. 7.6 Left. Visualization of the coronary motion during a cardiac cycle. The heart volume varies by about 12% in 1 second during a cardiac cycle. Right. Illustration of the different paths and targets the neural controller needs to navigate. The insertion point is shown in orange, and the randomly selected targets are shown in green. An example of a path is depicted as a dotted line.

reaches the target location. Four and five distinct target locations were chosen for respectively the heart and the liver, each involving the navigation of a minimum of 2 and a maximum of 4 bifurcations. For each test episode, a random target and a random starting rotation of the guidewire were chosen. We report in Table 7.1 the percentage of successful episodes for each test.

7.3.1 Navigation in coronary arteries during cardiac motion

In this example, we demonstrate the ability of our neural controller to navigate a dynamic environment without prior training on either this anatomy or this particular deformation (see Fig 7.6 left). As introduced in Section 6.6, the heart model was reconstructed from Magnetic Resonance imaging data but the motion was generated synthetically in SOFA environment, allowing to know the shape of the vascular tree and the centerline position at each time step. In this case, since the dimensions of the anatomy are similar to the dimensions of the training geometries, with an entry diameter of $3.8\,mm$, the same guidewire used during the training is adopted. The efficacy of the controller is tested in three different scenarios: a static case, in which the anatomy is not moving, a dynamic case in which the heart is beating, but the location of the centerline is not updated and a dynamic case in which the heart is beating and the position of the centerline is updated. For this anatomy, we selected 4 targets shown in the right part of Fig. 7.6 and randomly chose one for each of the 100 test episodes. In all test cases, our controller proves its ability to navigate the coronaries both in static (90% success rate, Table 7.1 a) and dynamic (97% success rate, Table 7.1 c) conditions, maintaining its performance also when navigating the anatomy without any knowledge about vessel deformation (89% success rate, Table 7.1 b). As shown in Table 6.3, the previous version of our controller reported

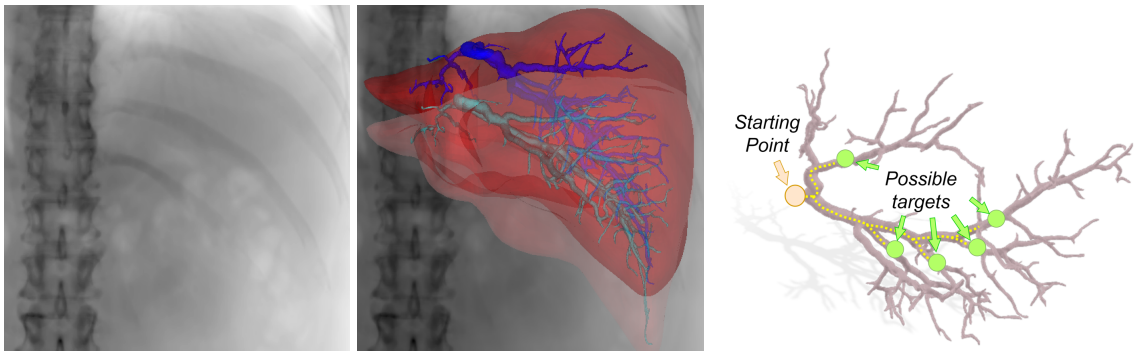

Fig. 7.7 Left: fluoroscopic image seen by the neural network. Center: preoperative position and shape of the liver and its venous system (opaque colors) and prediction of the 3D shape (in semi-transparent color) of both the liver shape and its vascular tree. The centerlines of the veins are also predicted, in real-time, and used by the neural controller. Right: Liver venous system with 5 different targets. The controller has to navigate the moving anatomy from the insertion point (in orange) up to the designated target, randomly chosen among the possible targets, in green.

a success rate of 68% in the same conditions: this increased success rate demonstrates the improved robustness of the newly developed method toward uncertainties and movements of the anatomy, partially due to the modified choice of $\mathbf{w}$ and n_Γ, which allow a more precise control of the tip orientation.

7.3.2 Navigation in hepatic veins during respiratory motion

In this section, we focus on a different clinical context, such as the endovascular treatment of hepatic venous outflow obstruction *(Ghibes et al. (2023))* or the endovascular treatment of portal hypertension *(Golowa and Cynamon (2012))*. The key difference when compared to the previous scenario is that, in this case, the vascular tree motion caused by the breathing of the subject is unknown, engaging ourselves in a true clinical scenario, in which the moving anatomy is only visible in fluoroscopic images, in 2D.

Using the neural network described in Section 7.2, we estimate, in real-time, the 3D position of the vessels' centerlines. Using a patient's abdominal CT scan, a training dataset, composed of 18,000 samples, was generated and used to train the motion prediction neural network, as described in Sec. 7.2.2. The test dataset contains a series of fluoroscopic images covering 5 inhale/exhale periods for a total of 50 samples. The main direction of motion is along the Inferior-Superior (IS) and Antero-posterior (AP) axes, with a small motion in the Left-Right (LR) direction, and a sliding motion of the organs against the thoracic cage. The trained network was evaluated using the Target Registration Error (TRE) on the hepatic veins centerlines and the hepatic veins mesh. Across the testing dataset, the mean displacement was $8.74 \pm 4.06\,mm$

Table 7.1 Navigation results summary, in heart and liver.

Heart			
Conditions	Success rate [%]		
	Previous controller	Our method	Random
Static[a]	100%	90%	15.6%
Dynamic, no centerline update[b]	68%	89%	
Dynamic, centerline update[c]	82%	**97%**	
Liver			
Conditions	Success rate [%]		
	Previous controller	Our method	Random
Static[d]	36%	89%	15%
Dynamic, no centerline update[e]	50%	24%	
Dynamic, centerline updated with our NN prediction[f]	50%	**93%**	

and $8.66 \pm 4.05\,mm$ while the mean TRE was $3.74 \pm 2.33\,mm$ and $3.84 \pm 2.37\,mm$ for the centerlines and the hepatic veins respectively. This error is not similar in each direction, since the motion along the direction perpendicular to the image plane is more difficult to estimate compared to the other two directions. This is reflected in the error of the network, which was, on average, below $2\,mm$ for the IS and LR directions and, on average, below $2.6\,mm$ for the AP direction.

We report in Table 7.1 the success rate of our controller, in similar conditions as presented for the heart: a static case, a dynamic case without centerline update, and a dynamic case in which the updated position of the anatomy is reconstructed from non-contrasted fluoroscopic images, thanks to our neural network. Given the dimension of the anatomy, a different guidewire, with a $6.5\,mm$-long tip and a tip curvature of $0.26\,mm^{-1}$ is used. For this second test anatomy, we chose 5 different target locations, as shown in Fig. 7.7 right, and randomly selected one for each of the 100 test episodes. In this more complex context, our controller demonstrates its efficacy with an 89% success rate (Table 7.1 d) in the static scenario, which is maintained

when transitioning to dynamic conditions. In this case, the agent reports a success rate of 93% (Table 7.1 f), while it shows an important performance drop (24% success rate, Table 7.1 e) when trying to navigate the dynamic anatomy without any information regarding the movement of the vessels (no centerline update in the observation space Ω). This shows the significance of our 3D vascular motion estimation.

7.4 Final considerations

In this chapter, we presented a neural controller, based on a deep reinforcement learning approach, able to navigate a guidewire in complex, unseen, moving anatomies with various dimensions. In addition, we proposed a method for estimating the 3D motion of the anatomy from single-view fluoroscopy images, even without the injection of a contrast agent. The combination of these two contributions makes it possible to automatically perform endovascular navigation in close to real-world conditions, as illustrated in two scenarios: a beating heart and a liver deformed under breathing motion. To the best of our knowledge, this has never been achieved before.

As illustrated by the results in Table 7.1, our new controller demonstrates its ability to navigate new complex anatomies, composed of various subsequent bifurcations, different in shapes and dimensions, with a mean success rate of 95% in the dynamic anatomies. This is very significant compared to the probability of reaching the designated targets when taking random actions which would lead to an average success rate of 15% for the two scenarios we have considered. This value can be obtained by considering the set $Z = \{z_1, z_2, ..., z_N\}$ of N different targets, each requiring the successful navigation of a number, b_{z_i}, of bifurcations. The mean probability of successfully reaching the target is equal to $\frac{1}{N}\sum_{j=1}^{N} 1/2^{b_{z_j}}$.

We compared our new controller with the agent described in Section 6.3, trained for 150,000 time steps of 0.01 s on anatomies with a $4\,mm$ diameter, consistent with the diameter of the training anatomies used in this work, allowing to use the same guidewire. The new controller outperforms our previous version in terms of both robustness to vessel motion (Table 7.1 c) and robustness to variations in vessel dimensions (Table 7.1, *Liver*). As stated in Section 6.6, the important performance loss shown by our previous controller when navigating the liver anatomy, can be explained by the difference in the dimension of the liver geometry, whose entry vessel presents a diameter of $7\,mm$, which almost doubles the diameter of the training anatomies. The high success rate obtained by our controller in the liver's static conditions (Table 7.1 d) demonstrates the adaptability of our new training strategy to anatomies presenting various bifurcation shapes and dimensions. However, when navigating a dynamic environment, the performances of the controller critically drop if the centerlines are not updated (see Table

7.1 e). Our neural network allows computing the new position of the vessels, thus reducing the difference between the real anatomy and the anatomy observed by the controller. In these conditions, our controller demonstrates its ability to navigate various dynamic anatomies both when a synthetic movement is generated (Table 7.1 c) and when the anatomy moves following real vessel movements (Table 7.1 f).

Chapter 8

Combined control of catheter and guidewire

The previous Chapters described the method developed in this work to autonomously perform the navigation of small downstream vessels operating a single medical device. To obtain a system able to automate all the phases of the intervention, the ability of this controller should be expanded to include also the selective cannulation of small lateral vessels. As already introduced in Chapters 1 and 5, during this stage, the combined control of a catheter and a guidewire is fundamental, as it is their interaction that allows reaching the entrance of small lateral vessels.

In literature, there is still a lack of work dealing with the combined navigation of a passive catheter combined with a guidewire. An interesting study was conducted by *Song et al. (2022)* that aimed at developing algorithms to make an endovascular robotic system autonomous. In their work, they used an RGB camera to track the surgical tools in vascular phantom environments and transformed these images in grayscale to conform to fluoroscopic images. The recognition of the surgical devices was performed by means of the YOLO *(Redmon et al. (2016))* algorithm. The management of the tools was entrusted to a deep reinforcement learning controller trained on the target task for 500,000 iterations. The x, y positions of both the catheter and the guidewire were used as input information for the agent, which was required to decide the needed actions to reach the target location. The training and testing were performed in two different vascular phantoms, the first one which only required pushing forward the devices, and the second one in which the instruments were manipulated to learn how to reach the supraortic vessels. Several targets were selected and for each of them, a different training was performed. During the testing phase, the algorithm achieved the target location in 96% of the cases. The main drawback of this strategy is that new training is required every time the target location changes, limiting the technique's applicability.

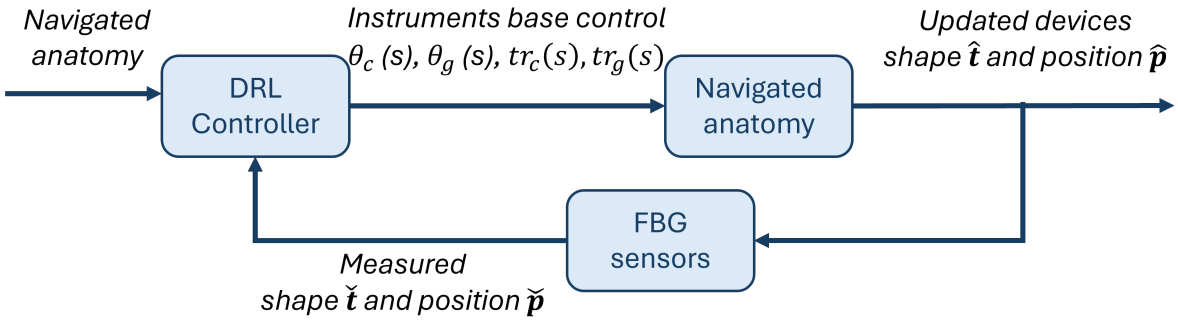

Fig. 8.1 Control loop specialized to the simultaneous navigation of two devices. Also in this case, the DRL controller takes as input two pieces of information: the anatomy that is being navigated and the shape $\check{\mathbf{t}}$ and position $\check{\mathbf{p}}$ of the devices. Given these two inputs, the DRL controller computes the rotation $\theta_c(s)$ and the translation tr_c to apply to the proximal end (base) of the catheter and the rotation $\theta_g(s)$ and the translation tr_g to apply to the proximal end (base) of the guidewire to reach the desired target. These controls will result in an updated device shape $\hat{\mathbf{t}}$ and position $\hat{\mathbf{p}}$, which will be retrieved from the simulation (simulated navigation) or measured by the FBG sensors (real-world scenario). Due to the limitations of the measurement system, it is possible to measure $\check{\mathbf{t}} \neq \hat{\mathbf{t}}$ and $\check{\mathbf{p}} \neq \hat{\mathbf{p}}$.

In our work, we aim at training a controller able to reach different target locations in different anatomies, by combining two instruments, a catheter and a guidewire. As in the previous version of the controller, we aim to obtain an algorithm that is also able to generalize the learned task to new, unseen anatomies. Our objective is detailed in Section 8.1, while the characteristics of this controller will be presented in Section 8.2 and 8.3. In these sections, the process that led to the development of the final system is described, along with the results, reported in Section 8.2.4 and 8.3.3. It is important to mention that this work constitutes the current research topic, and the results reported here are therefore only preliminary.

8.1 Problem statement

The purpose of this work is to design a controller able to simultaneously navigate two devices, enabling access to target locations that would be inaccessible with a single instrument.
The control loop describing this system is shown in Fig. 8.1. The input is constituted by the anatomy that needs to be navigated and the shape and position of the devices. With this knowledge, the controller decides which action to take in order to reach the desired location. In this case, since two instruments are managed, four distinct actions are possible: the rotation $\theta_c(s)$ and translation $tr_c(s)$ of the catheter and the rotation $\theta_g(s)$ and translation $tr_g(s)$ of the guidewire. These actions are imposed to the proximal part of the device, placed at the

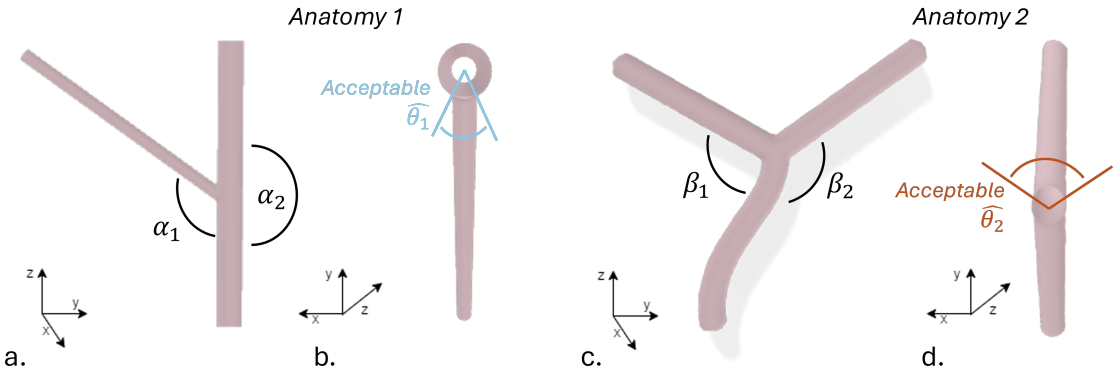

Fig. 8.2 Range of tip orientations that lead to reach the target location in two different anatomies. Anatomy 1 (a,b) presents an important angle difference between α_1 and α_2, an important difference between the diameter of the main branch and of the lateral vessel. In Anatomy 2 (c,d), $\beta_1 \simeq \beta_2$ and there is no diameter difference between the main branch and the target vessels. These elements results in an acceptable orientation angle range $\hat{\theta}$ smaller for anatomy 1 than for anatomy 2.

entry of the vessel. As before, since the surgical tools are inserted inside the vessels of the patient, the controls applied to the base of the instrument do not necessarily correspond to the same movement at the distal end of the tool ($\hat{\theta}_c(s) \neq \theta_c(s)$, $\hat{tr}_c(s) \neq tr_c(s)$, $\hat{\theta}_g(s) \neq \theta_g(s)$ and $\hat{tr}_g(s) \neq tr_g(s)$). Furthermore, in this case, the instruments are also interacting with each other and their relative movement leads to a change in the combined device shape. The new shape and position of the two interacting devices can be measured and given as input to the controller. This information can be easily retrieved in a simulated environment and when switching to the real world, it can be measured by means of FBG sensors. When applying the controller to the real world, it is also important to consider that the measured tip shape and position can be affected by errors due to the limitations of the sensor, leading to $\check{\theta}_c(s) \neq \hat{\theta}_c(s)$, $\check{tr}_c(s) \neq \hat{tr}_c(s)$, $\check{\theta}_g(s) \neq \hat{\theta}_g(s)$, $\check{tr}_g(s) \neq \hat{tr}_g(s)$. Since in this first implementation, the system is completely tested and developed in a simulated environment, these differences will not be considered.

In the previous version of the algorithm, the fundamental control consisted in finding the desired tip orientation to reach the target location and applying the correct rotation to the proximal end of the device to obtain it. This step remains fundamental also in the cannulation phase of the intervention and requires an even more precise tip orientation. When trying to reach the small lateral vessels indeed, given the important vessel diameter variation between the main and the target vessel and the important difference between the angles characterizing the bifurcation (see Fig. 8.2 a), the range of tip orientations that allows reaching the target is smaller than in the navigation of vessels with nearly constant diameters (Fig. 8.2).
Furthermore, the correct tip orientation is no longer sufficient to attain the desired objective,

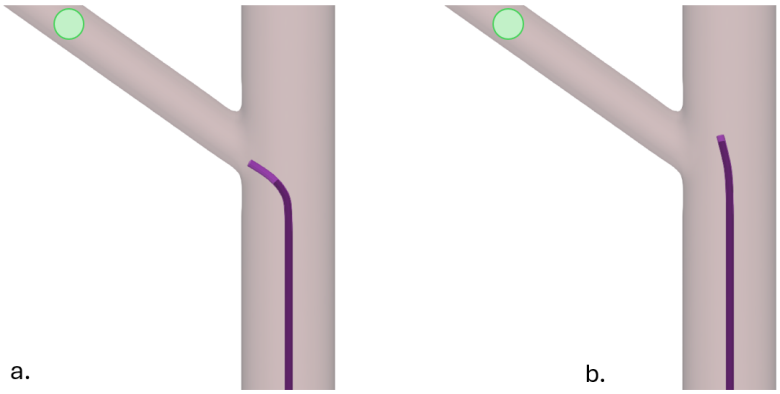

Fig. 8.3 Importance of the correct tip shape (a) to reach the target location placed in small lateral vessels (green dots): an incorrect tip shape leads to unsuccessful cannulation even when the tip orientation is correct (b).

but the tip shape also plays an important role in this case: as shown in Fig. 8.3, a correct tip orientation but a wrong tip shape lead to an unsuccessful navigation. The shape of the tip can be controlled by modifying the relative position of the guidewire inserted inside the catheter. These considerations highlight the necessity of controlling two key variables: the tip orientation and the tip shape, both defined by actions applied to the proximal end of the devices. Despite the need to obtain a more precise tip orientation, the control of this variable remains essentially the same as the one developed in Chapters 6 and 7, requiring only to impose stricter rules. For this reason, our initial focus was on the development of an algorithm to control only the tip shape, which constitutes the novelty of this work. The characteristics of the new controller are reported in Section 8.2, along with the developed criteria and the obtained results. Finally, Section 8.3 presents the results and key components of a controller obtained by integrating the principles for achieving the correct tip shape with those for ensuring proper device orientation.

8.2 Tip shape control strategy

Assuming that the tip of the instrument is already correctly oriented (*i.e.,* no rotation of the instrument is needed), the principal element that defines whether the tip of the catheter will enter the lateral vessel is the angle defined by the tip (α_{tip}, Fig. 8.4 b). This angle should be as similar as possible to the angle between the main branch of the vessel, navigated by the instrument, and the target branch (β_{an}, Fig. 8.4 a).
After the definition of this criterion, the control algorithm and its training environment can be designed. Once again our aim is to obtain a controller that is not only able to reach different targets in the training anatomies, but that can also apply the learned control to new unseen

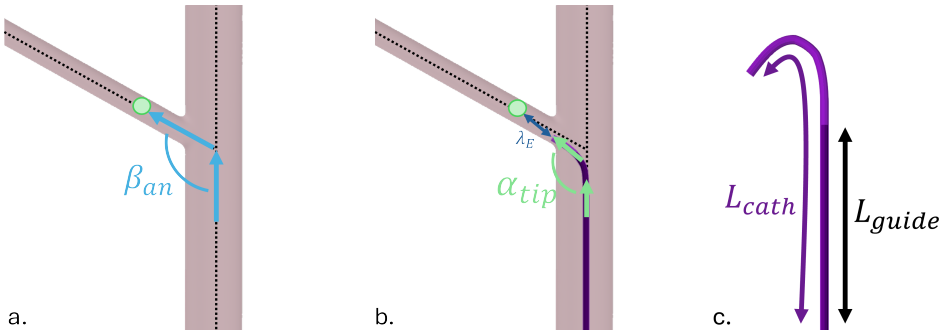

Fig. 8.4 Representation of the new terms of the observation. β_{an} describes the angle between the main branch and the branch leading to the target (a), α_{tip} represents the description of the tip shape (b), λ_E the Euclidean distance between the tip of the device and the target location (b) and $L_{rel} = \frac{L_{guide}}{L_{cath}}$ (c).

anatomies. Similarly to what was observed in the previous versions of the algorithm, three elements are essential to our purpose: the choice of an appropriate observation space, the accurate design of a reward function and the choice of the training anatomies. The selection of the most suitable vessel anatomies, together with the design of a proper observation space, allows the controller to observe during the training the highest possible number of states. This helps the agent to recognize during the testing phase conducted on new anatomies, situations similar to the ones experienced during the training. All these aspects will be described in the following sections.

8.2.1 Observation space

Starting from the formulation developed for the navigation of the small downstream vessels (Sections 6.3.2, 7.1.1), the observation space was modified in accordance with the current control requirements and takes the form:

$$\Omega = \{\zeta_t, \zeta_{t-ndt}, \lambda_{E_t}, \lambda_{E_{t-ndt}}, a_t, d_v, \mathscr{A}_t, \mathscr{A}_{t-ndt}, \mathscr{F}, L_{rel}\}$$

The terms $\zeta_t, \zeta_{t-ndt}, a_t$ and d_v, are defined consistently with those presented in Chapters 6 and 7, while the other terms present some differences:

- The geodesic normalized distance $\lambda_t, \lambda_{t-ndt}$ is here substituted with the Euclidean normalized distance at the current (λ_{E_t}) and previous ($\lambda_{E_{t-ndt}}$) time steps. This substitution has been performed because the Euclidean distance provides in this case more informative data than than the geodesic distance.

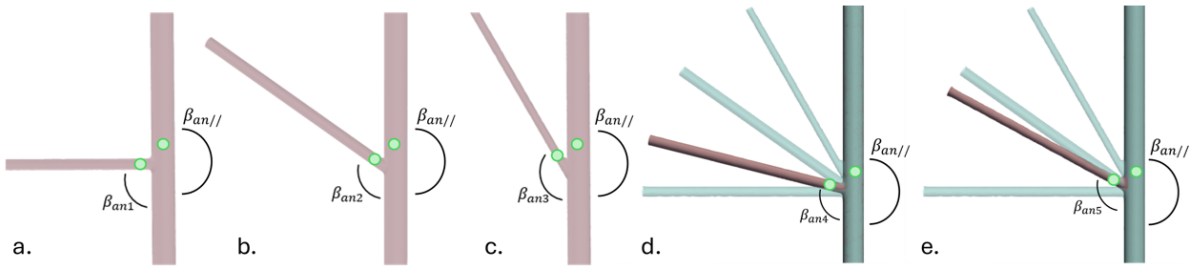

Fig. 8.5 Anatomies used for training (a, b, c) and testing (d, e) the controller. Each anatomy presents a different β_{an} angle with the small lateral vessel: $\beta_{an_1} = 90^o$, $\beta_{an_2} = 125^o$, $\beta_{an_3} = 145^o$, $\beta_{an_4} = 105^o$, and $\beta_{an_5} = 111^o$ and the same β_{an} with the main branch ($\beta_{an//} = 180^o$). All the possible targets are represented with green dots.

Both these quantities are normalized with respect to the initial target distance λ_{E_0}, computed at the beginning of each training episode.

- $\mathscr{A}_t = \cos(\alpha_{tip_t})$ and $\mathscr{A}_{t-1} = \cos(\alpha_{tip_{t-ndt}})$ are introduced, which represent the cosine of the angle α_{tip} at the current time step t and at the previous one $t - ndt$.
- $\mathscr{F} = \cos(\beta_{an})$ is added, which indicates the cosine of the angle β_{an}.
- L_{rel} which is obtained as the ratio between the length of the guide (L_{guide}) and the length of the catheter (L_{cath}).
- The term ω has been omitted because it provides information about the direction of the branch that does not lead to the target location, which is irrelevant to this task.

8.2.2 Reward Function

To account for the criteria described in Section 8.1, the reward function takes the form:

$$r(s_t, a_t) = \underbrace{\frac{2}{1 + e^{20|\rho|-6}} - 1}_{a} + \underbrace{\text{sgn}(dl)0.25\,(1 - \lambda_{E_t})}_{b} - \underbrace{0.5\,(1 - \lambda_{E_t})}_{c} \tag{8.1}$$

Here $\rho = \alpha_{tip} - \beta_{an}$, measured in degrees, represents the difference between the current tip angle and the angle characterizing the desired branch, sgn represents the sign function and $dl = \lambda_{E_{t-ndt}} - \lambda_{E_t}$. The term a is once again a modified version of the sigmoid function taking values in $[-1;1] \in \mathbb{R}$ and it rewards the agent with scores that approach $+1$ when ρ is close to 0. The module of the term b increases as the Euclidean distance λ_{E_t} decreases and takes positive values if $\lambda_{E_t} < \lambda_{E_{t-ndt}}$ and negative if $\lambda_{E_t} > \lambda_{E_{t-ndt}}$. Finally, the term c penalizes the agent at each time step, with a penalty that increases as the target is approached, pushing the

controller to reach the target as quickly as possible in situations where the optimal tip shape requires the guidewire to be advanced further than the catheter (*i.e.,* when the target is located in the large main vessel).
All the coefficients present in this formula were tuned through a trial-and-error process, in which the performances of the controller were evaluated as these parameters were modified, until finding a combination leading to the desired performances.

8.2.3 Choice of training anatomies and devices

The next important step is the definition of the training anatomies and of the devices that will be used to navigate these geometries.

For the purpose of this study, the shapes shown in Fig. 8.5 a,b,c have been adopted. These anatomies present a main vessel with a diameter of 20 *mm*, while the lateral branches have a diameter of either 10 or 7 *mm*. These lateral vessels form an angle β_{an} of 90^o, 125^o and 145^o with the main branch, which is completely straight ($\beta_{an//} = 180^o$). For each training episode, both a random anatomy and a random target, chosen from the possible options (Fig. 8.5, green dots), were selected.

To navigate all these anatomies using the same catheter combined with a guidewire, it was necessary to adopt instruments with a suitable shape. In this first implementation of the work, a straight guidewire and the catheter shown in Fig. 8.6 a were adopted. The combined shape of these two devices, for various values of L_{rel}, is shown in Fig. 8.6. To study the relationship between L_{rel} and α_{tip} during device insertion into vessels, a simulation was conducted where the guidewire was advanced through a catheter placed inside a straight tube with a diameter of 20 *mm*. At the start of the simulation, the catheter assumes its rest shape, with its tip in contact with the vessel wall. At each time step, the guidewire is advanced at a constant speed inside the catheter, and α_{tip} is measured. The obtained relationship is shown in Fig. 8.6. As it can be observed from this graph, a non-linear and variable relationship is present between these two parameters, with $80^o < \alpha_{tip} < 180^o$.

The choice of the instruments influences also the selection of the actions that the controller can apply to the instruments. Having two instruments introduces 4 possible controls: rotation of the catheter $\theta_c(s)$, rotation of the guidewire $\theta_g(s)$, translation of the catheter $tr_c(s)$ and translation of the guidewire $tr_g(s)$. In this first implementation, the rotation of instruments has not been considered to simplify the implementation of the control strategy, and $\theta_c(s)$ and $\theta_g(s)$ are therefore considered constant and equal to 0. During each episode, the navigation is started with a tip orientation that allows to reach the target branch under the condition that the combined device assumes the correct tip shape. The translation is controlled by setting the speed of the instruments. To reduce the number of controlled variables, a constant translation

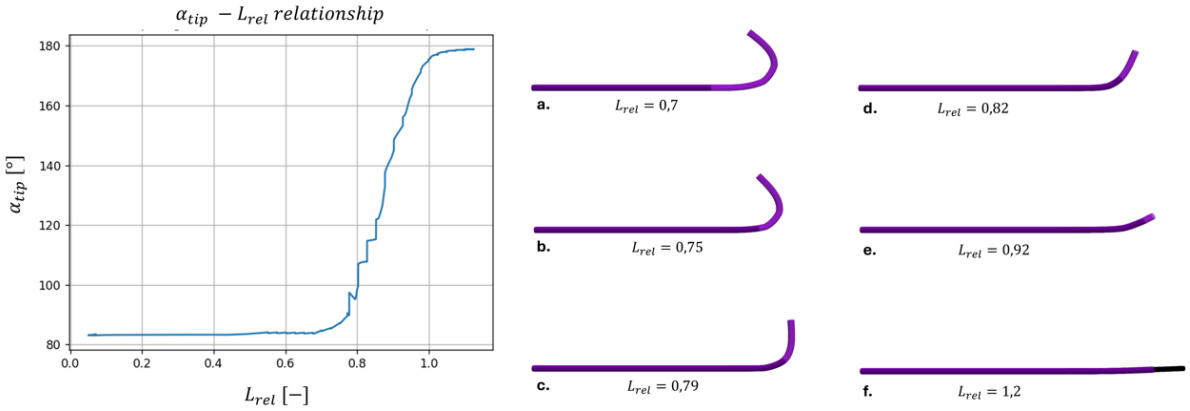

Fig. 8.6 Left: nonlinear relationship between the tip shape, described by α_{tip}, and the relative insertion of the guidewire inside the catheter (L_{rel}) when the devices are inserted inside a 20 mm diameter vessels. On the right, the combined free instrument shape for different values of L_{rel} is shown.

speed of 2 mm/s has been applied to the catheter, leaving to the agent only the control of the velocity of the guidewire. The action space of the DRL controller is therefore composed of one single parameter, which tunes the speed of the guidewire with values between -2 mm/s and 6 mm/s. Thanks to this large range of velocities, the agent can quickly modify the shape of the combined instrument, controlling the relative position of the guidewire inside the catheter.

8.2.4 Results

In this Section, the results obtained with the controller described above are reported.

The control algorithm chosen to train the agent was as before the SAC DRL algorithm. The controller was trained for 120,000 time steps with a learning rate of 10^{-4}, a batch size of 512, a buffer size of 10,000 and a discount factor of 0.98. The entropy coefficient was also in this case learned autonomously during the training. 1,000 steps were performed before starting the training of the model, which was updated at every time step. Both the tests and the training were performed on a computer equipped with 32 GB of RAM, a 24-core Intel(R) Core(TM) i7 running at 3.40 GHz and an NVIDIA GeForce GTX 1080 GPU.

The algorithm was tested both on the same anatomies on which it was trained and on new anatomies presenting different lateral branch angles β_{an}. The comparison between the train and test anatomies is shown in Fig. 8.5 d,e. The testing anatomies have an angle $\beta_{an_4} = 105^o$ and $\beta_{an_5} = 111^o$, with a main vessel diameter of 20 mm and a lateral vessel diameters of 10 mm. Both for the training and the test anatomies, the algorithm was evaluated on 50 episodes, choosing each time a random target location and a random anatomy. The controller was able to reach the target in all the tested cases, reporting a success rate of 100%. Since in this case the

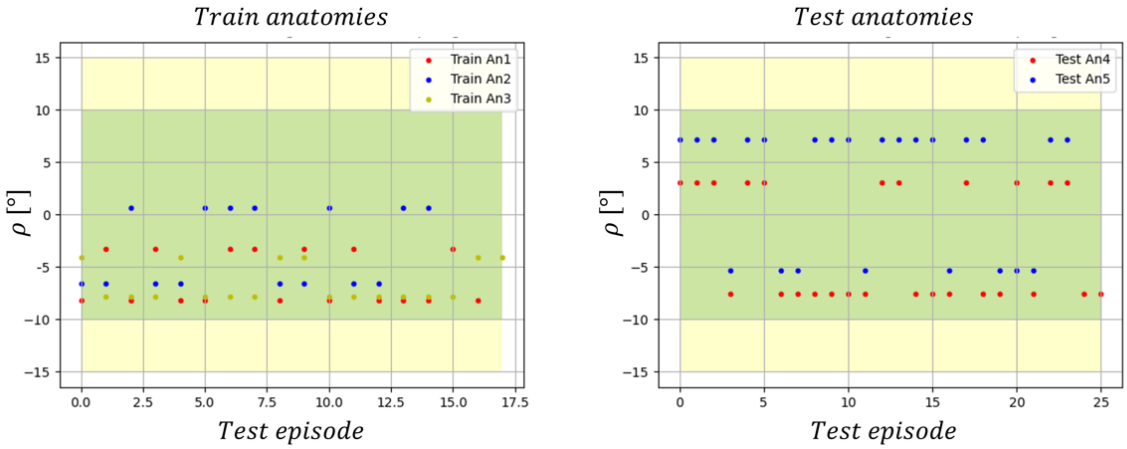

Fig. 8.7 Difference between the branch angle toward the target β_{an} and the tip angle α_{tip} at the end of each of the 50 test episodes, both in the training and in the testing anatomies. In both cases, $-10^o \leq \rho \leq 10^o$, where $\rho = \alpha_{tip} - \beta_{an}$, demonstrating the ability of the controller to obtain the optimal tip shape.

aim of the controller is not only to reach the target but to reach it with the correct tip shape, for each test episode, the angular difference $\rho = \alpha_{tip} - \beta_{an}$ was computed. Fig. 8.7 shows the obtained values both in the train and in the test anatomies. As it can be observed, in both cases $-10^o \leq \rho \leq 10^o$, with the mean of the absolute value of ρ taking a value of 5.6^o and 6.1^o in the case of the train and test anatomies respectively.

8.3 Tip shape and orientation control

The control strategy described in Section 8.2 was finally integrated with the control strategy developed in Chapters 6 and 7, to obtain a controller able both to manage the shape of a combined device resulting from the interaction of two separate tools, and to handle its rotation to reach the goal.

The characteristic of this approach is that the action space is now composed of two variables, one regulating the translation and one the rotation. As already explained in the previous section, when two instruments are controlled, 4 degrees of freedom are introduced. However, given the straight nature of the guidewire, imposing a rotation on this instrument would not produce any effect on the shape of the combined instrument. For this reason, the rotation $\theta_g(s)$ of the guidewire is considered constant and equal to zero, and the controller is required to handle only the rotation of the catheter with a proximal twist $\theta_c(s) \in [-30^o; 30^o]$. Concerning the translation, the choice made for the controller presented in Section 8.2 was maintained: both the catheter and the guidewire are manipulated imposing a certain translation speed and, while the

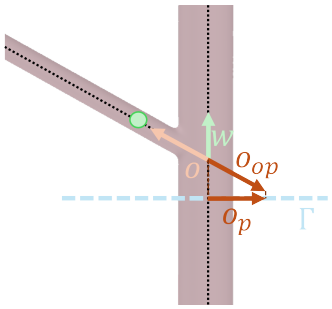

Fig. 8.8 Representation of the new term of the observation space $\omega_D = \mathbf{k}_p \cdot \mathbf{w}_p$. When $\mathbf{w} \perp \Gamma$, $\mathbf{w}_p = \mathbf{o}_p$, where $\mathbf{o}_p$ is the projection on the plane Γ of a vector $\mathbf{o}$, which has the same direction as the vector describing the direction leading to the target ($\mathbf{o}$), but with opposite orientation.

DRL algorithm handles the velocity of the guidewire imposing a velocity between $-2\ mm/s$ and $6\ mm/s$, a constant speed of $2\ mm/s$ is set for the catheter.

This new algorithm should implement the two aspects of the control developed in the previous works, *i.e.,* the control of the combined device orientation and shape. For this reason, the observation space and the reward function were designed to incorporate all the elements of the precedent versions of the system.

8.3.1 Observation space

The observation space presented in Section 8.2.1 is here expanded to include a modified version of the term ω, introduced in Sections 6.3.2 and 7.1.1 and excluded from the controller presented in Section 8.2. The observation space takes therefore the form:

$$\Omega = \{\zeta_t, \zeta_{t-ndt}, \lambda_{E_t}, \lambda_{E_{t-ndt}}, a_t, \omega_D, d_v, \mathscr{A}_t, \mathscr{A}_{t-ndt}, \mathscr{F}, L_{rel}\}$$

The term ω_D is obtained as $\omega_D = \mathbf{k}_p \cdot \mathbf{w}_p$, where $\mathbf{k}_p$ is the projection of the curvature vector $\mathbf{k}$ of the tip and $\mathbf{w}_p$, is the projection of the vector describing the direction of the wrong branch $\mathbf{w}$ onto a plane Γ perpendicular to the centerline of the main branch. When the wrong direction vector $\mathbf{w}$ is perpendicular to Γ (Fig. 8.8), the magnitude of $\mathbf{w}_p$ assumes values close to zero, making the computation of this vector, and of ω_D as a consequence, unreliable. To overcome this problem, $\mathbf{w}_p$ is taken in these cases as the projection onto Γ of a vector $\mathbf{o}$, which has the same direction as the vector describing the direction leading to the target, but with opposite orientation (Fig. 8.8).

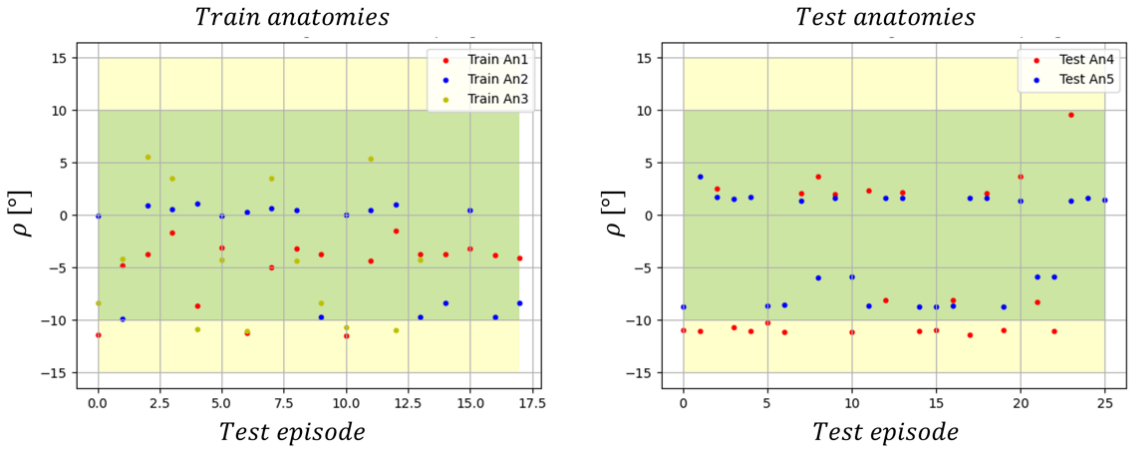

Fig. 8.9 Difference between the angle β_{an} between the main branch and the target branch and the tip angle α_{tip} at the end of each of the 50 test episodes, both in the training and in the testing anatomies. Both for the training and for the testing anatomies, ρ obtained as $\rho = \alpha_{tip} - \beta_{an}$ is included in the range $[-15^o; 15^o]$.

8.3.2 The reward function

To account for the criteria developed in the two previous versions of the controller, the reward function adopted for this work is given by the combination of the two precedent reward functions (Eq. (6.6), (8.1)) and takes the form:

$$r(s_t, a_t) = \underbrace{\frac{2}{1+e^{20|\rho|-6}} - 1}_{a} + \underbrace{\text{sgn}(dl)0.25\,(1-\lambda_{E_t})}_{b} - \underbrace{0.5\,(1-\lambda_{E_t})}_{c} + \underbrace{\frac{2}{2+e^{12*(\omega+0.65)}} - 0.5}_{d} - \underbrace{|a_{t_{rot}}| * 0.2}_{e}$$

The terms a, b and c have already been discussed in Section 8.2.2 and will be therefore not further examined. The term d is a modified version of the term a of Equation (6.6). The new coefficients included in this expression were introduced to enforce a more precise control on the tip orientation of the devices and aim at reducing the range of tip orientations $\hat{\theta}$ which are considered acceptable, as discussed in Section 8.1. These coefficients were found through a trial-and-error process. Finally, the term e discourages the algorithm from executing random, unnecessary rotations.

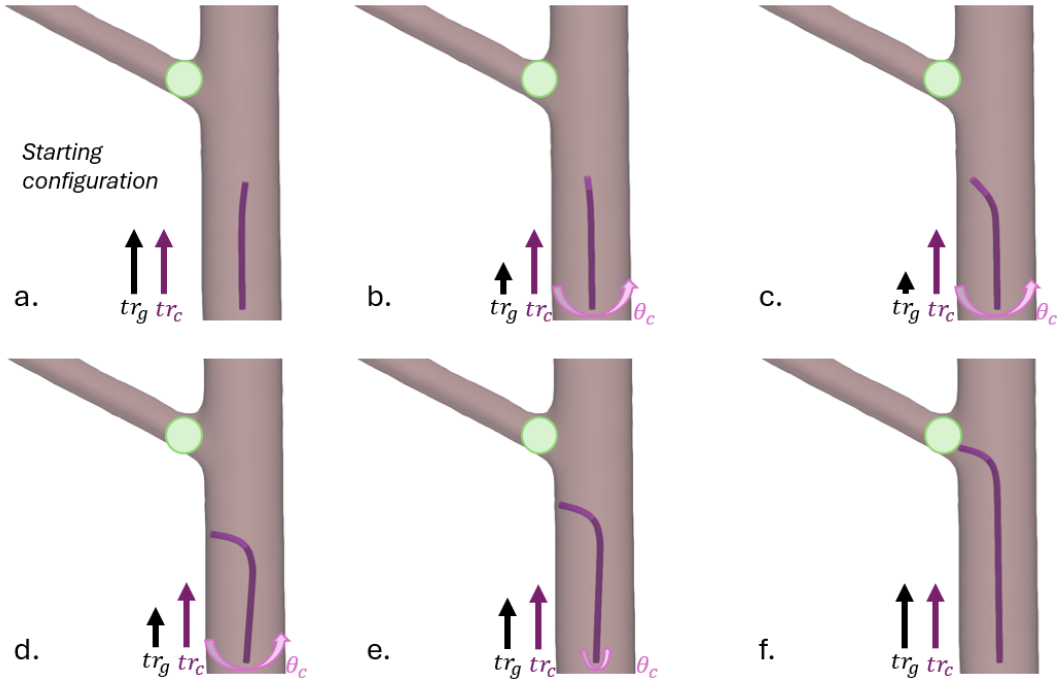

Fig. 8.10 Example of the controls decided by our control algorithm when a lateral branch needs to be reached. At the beginning of the navigation, the guidewire is inserted inside the catheter and the combined instrument assumes a straight configuration (a). The control algorithm then starts to push the catheter with a constant speed tr_c, and decides the speed to apply to the guidewire (tr_g) and the rotation to apply to the catheter (θ_c). b,c,d: $tr_g < tr_c$ to allow the tip of the combined instrument to assume a more curved configuration and catheter rotation θ_c. e: the desired tip orientation is almost reached, so the controller applies only a small θ_c, while the speed of the guidewire is regulated to fine tune the shape of the tip. f: The desired tip shape and orientation are obtained, so $tr_g = tr_c$ and $\theta_c = 0$. This configuration allows to reach the target.

8.3.3 Results

The training strategy presented in Section 8.2.4 was also adopted for this controller, with the only difference being that, instead of training the algorithm for 120,000 time steps, this system's learning process required 240,000 time steps. The anatomies shown in Fig. 8.5 a,b,c and the devices reported in Fig. 8.6 were used also in this case to train the agent. For each training episode, besides choosing a random anatomy and a random target among the available ones, a random initial orientation of the devices was also imposed, to allow the agent to explore the broadest possible range of conditions.

The obtained controller was tested both on the training anatomies and on the test anatomies (Fig 8.5 d and e). In both situations, it was tried on 50 episodes, choosing every time a random anatomy, a random target and a random initial tip orientation. An example of the control executed by the agent in a trial episode with the test anatomies is presented in Fig. 8.10,

showing different phases of the navigation. Our trained agent was able to correctly steer the combined devices toward the target location in all the considered cases, reaching a success rate of 100%. Fig. 8.9 reports the difference ρ between the obtained tip angle α_{tip} and the angle describing the anatomy β_{an} in all the experiments. A mean of the absolute value of ρ of 5.0^o and 6.1^o was obtained in the training and test anatomies respectively. However, for both tests, a slight reduction in accuracy was observed compared to the precision achieved when the controller managed only the guidewire speed: in 7 (training anatomies) and 12 (test anatomies) out of the total 50 episodes, ρ assumed values outside the range $[-10^o; 10^o]$. These values are however always included in the $[-15^o; 15^o]$ interval, demonstrating the capability of the algorithm to handle this new scenario.

8.4 Final considerations

In this Chapter, a deep reinforcement learning controller, able to simultaneously handle two instruments with different geometrical characteristics was developed. This controller integrates the control strategy described in the previous Chapters (Chapters 6 and 7), with a new control strategy that allows to learn the relationship between the tip shape of the combined device and the relative position of the guidewire inside the catheter.

The obtained agent demonstrated promising results on synthetic anatomies exhibiting characteristics that would make navigation with a single instrument impossible, paving the way for the development of a controller able to manage the combination of two devices in more complex and realistic anatomies. The aim of this controller is either to navigate the main vessel with a straight configuration of the combined device or to secure access to small lateral vessels by placing the tip of the catheter at the entrance of the target branch. Once this access is secured, a second softer guidewire can be advanced inside the catheter to proceed with the following phase of the procedure.

As already mentioned, the controller described in this chapter constitutes the current research topic, which is therefore not yet finalized. This work considers a simplified version of the problem, which is useful for individuating the parameters that influence a successful cannulation. Already in this simple setup, the tip of the catheter is in some cases in contact with the vessel wall while the device is advanced. This induces a change in the shape of the instrument, which is not caused by a modified relative position of the guidewire inside the catheter. Given the presence of the contacts, finding a mathematical relationship to model this phenomenon can be very challenging and introduce modeling errors. Furthermore, when dealing with more realistic scenarios, given the variable stiffness of both the catheter and the guidewire and the complex shape of real catheters (Fig. 1.1), the relationship between L_{rel} and α_{tip} becomes much more

complicated. These sources of non-linearity make a DRL controller a good candidate for this task.

Another interesting aspect of this work, is the similarity it introduces between the combined passive devices, and the active catheters, whose tip shape can be modified in response to external stimuli. This characteristic makes them able to reach certain targets which is difficult to attain with passive devices. Having a controller able to tune the combined shape of a passive catheter by modifying the position of the guidewire sliding in it can somehow simulate the use of an active device, maintaining the advantages of passive devices such as the reduced cost and the large availability.

Chapter 9

Conclusions and perspectives

The main focus of the work developed in this PhD thesis was to research and implement solutions aimed at assisting clinicians during endovascular interventions. Despite the many advantages of these procedures, their execution is usually very challenging. For these reasons, the work proposed in this manuscript aims to facilitate the execution of endovascular interventions, with the goal of reducing operation time and, consequently, minimizing X-ray exposure for both the patient and the clinician. Specifically, two main systems have been developed: an assistance system to support the clinician during the cannulation phase of the intervention, and a set of control algorithms designed to autonomously operate current market-available endovascular robots. These robots are indeed only leader-follower devices entirely controlled by the clinician and are not able to automate tasks or offer intraoperative support.

9.1 Contributions

The first important achievement of this work is the development of an accurate simulated environment to reproduce the physics of the medical devices navigating inside the vessel walls. This environment, described in Chapter 3, has been essential in all the phases of the work. We designed our environment focusing on two key aspects: the choice of the physical model to simulate the physics of flexible devices and the accurate simulation of their contacts with the navigated vessels. For the physics of the instruments, we selected the Timoshenko beam model, thanks to its ability to simulate thin tube devices subjected to twist and bending. Concerning the interaction model, we used two different strategies: a penalty-based approach to simulate the contacts between the devices and the vessel walls, and a composite model with varying mechanical characteristics to simulate the physics of the guidewire inserted inside the catheter as a single model.

Chapter 4 reports the details regarding the choice of the sensors used in this work to acquire the real-time shape of the medical tools. Specifically, in this work we adopted Fiber Bragg Grating sensors due to their promising characteristics, including compact size, high flexibility, and electromagnetic immunity, which make them well-suited for the integration in catheters and guidewires. However, this technology still has important limitations, causing shape reconstruction errors that negatively impact our methods. To overcome these limitations, we proposed two possible solutions: a rigid registration technique to ensure to reconstruct the shape of the devices inside the navigated vessels, and the development of a neural network capable of reconstructing the devices' shape from the raw sensor measurements. Both these strategies demonstrated their ability to improve the accuracy of the FBG measurements.

These algorithms were then used as building blocks for the assistance system developed to support the clinician during the selective cannulation phase of the intervention and for the design of the control algorithms.

The assistance system, presented in Chapter 5, aims to generate enhanced fluoroscopic images to support the clinician during the selective cannulation phase of small lateral vessels. This system superimposes, on the non-contrasted fluoroscopic images, the shape of the vessels and the outcome of the clinician's actions, before he/she performs them. In particular, we designed a system able to show the shape that the catheter would take if the guidewire was withdrawn at that specific moment (*predicted catheter shape*). Developing this assistance tool required a reliable simulated environment, which was used both to model the behavior of elements not present in the real world (*i.e.,* the predicted shape of the catheter) and to regularize the shape data collected by the FBG sensors. These data were post-processed with the rigid registration technique discussed in Chapter 4, and then fused with the shape computed by our simulation. This method demonstrated an accuracy of 1.1 ± 0.7 *mm* on fluoroscopic images plane while a user study performed on a phantom reported an intervention time reduction of 56% adopting our system, proving the usefulness of our framework.

Considering the development of the neural controllers, the objective of this research was to develop an autonomous algorithm able to navigate the devices toward the designated target locations. Given the diversity of the anatomies, our primary requirement was the ability to generalize the task to all possible geometrical variations.

For the development of this controller, we adopted a Deep Reinforcement Learning technique, relying on the Soft Actor critic algorithm. This algorithm introduces an entropy term in its objective function, helpful when trying to train a DRL controller able to generalize to new environments. However, the real core of our system lies in the training strategy, which involves three main elements: the choice of an appropriate observation space, the construction of a suited set of training anatomies and the design of the reward function. We built our observation

space to contain only terms describing the relative position of the instrument inside the anatomy, allowing the agent to recognize familiar situations even when navigating completely new environments. Thanks to our choice of training anatomies, constituted by only 4 bifurcation patterns, the agent was able to observe all the possible anatomical variations during the training. Finally, we designed our reward function to encourage the agent to apply controls to reach the correct tip orientation, avoiding useless device rotations. Following these principles, the controller described in Chapter 6 was developed. After training in a simulated environment, the controller was able to navigate unseen, complex, static, virtual anatomies with a success rate of 96%. We also tested this algorithm in a real-world scenario by generating control actions within a simulated environment that reproduced a 3D-printed phantom. We then applied these actions to a real medical device navigating the actual phantom using an endovascular robot. The phantom was designed to include a sequence of several bifurcation paths, leading to a total of 6 target locations. In these conditions, the robot, controlled in open loop, was able to reach 5 out of 6 targets.

However, this first implementation of the autonomous controller was designed to navigate static vessels, whereas real anatomical structures are continuously subjected to physiological movements. Additionally, comprehensive tests on the controller showed its limited capability to navigate anatomies with vessel diameters significantly different from those characterizing the training geometries. To overcome these limitations, we extended our training strategy to adapt the observation space, and we expanded our training dataset to include moving anatomies during the training process. We tested this new version of the controller, described in Chapter 7, in our simulated environment in two realistic scenarios, a beating heart and a liver deformed by breathing motion. In these moving, unseen anatomies with various vessel diameters, the controller was able to achieve an average success rate of 95%.

To create a comprehensive pipeline capable of automating all phases of the intervention, we developed a third version of the controller, able to operate two devices simultaneously. This controller was specifically designed to automate the selective cannulation procedure. Also in this case the agent, described in Chapter 8, was trained in our simulated environment using a limited set of synthetic anatomies featuring a small lateral vessel branching from the main navigated one. The controller was designed to generalize the learned behavior to new, unseen anatomies. After training, the agent was tested both on the training anatomies and on new simple geometries, demonstrating its ability to operate simultaneously two instruments in all the tested cases, laying the foundation for future research in more realistic anatomical contexts.

9.2 Discussion & Future work

The assistance system developed in this work and described in Chapter 5, demonstrated its usefulness in supporting clinicians during the cannulation phase of the intervention. The work presented in this manuscript constitutes a first implementation of the system, yet shows encouraging results in terms of usefulness and accuracy. To the best of the author's knowledge, this is the first time that such a system, able to superimpose on the fluoroscopic images not only the shape of the vessels but also the outcome of certain actions of the clinician before he/she actually performs them, is developed. All the systems presented in the literature, are either purely visualization systems *(García-Vázquez et al. (2018); Kuhlemann et al. (2017); Soulami et al. (2016))* or simulators *(Korzeniowski et al. (2017); Luboz et al. (2013); Wang and Wu (2021))* to teach the clinicians new skills.
To improve the system developed in this work, a first step would involve the integration of additional information in our assistance framework. In the current implementation, as clinicians are used to being guided by 2D-fluoroscopic images, our prediction is shown in the same 2D view. However, all the computations are performed in 3D, which we could exploit to provide additional feedback by indicating whether the catheter needs to be rotated to position its tip at the vessel entrance.
Additional potential developments involve the improvement of simulation realism, adding elements such as the effect of blood flow and vessel wall deformation. The blood circulation affects the cannulation phase of the intervention, causing movements of the tip of the catheter which are not accounted in the current implementation. Furthermore, the anatomy is currently considered undeformable, which is a good approximation in the case of renal artery cannulation but which limits the applicability of the system in other cases. Our simulated environment can already incorporate moving anatomies while maintaining real-time performance, and blood flow could be added too. However, the real challenge lies in acquiring the data to perform the simulation update, which must be accurately estimated in real-time directly from the patient. To obtain the movement of the anatomy, we could use the method developed by F. Lecomte (Section 7.2), which estimates the displacements of the vessels, even in non-contrasted fluoroscopic images. This strategy has already been used in this work to reconstruct the vessel motion of the structures navigated in Chapter 7. Considering instead the blood flow, collecting accurate real-time data can be very challenging and is still an open question.

The second system we presented (Chapter 6 and 7) is an autonomous controller, based on a deep reinforcement learning approach, able to navigate a guidewire in complex, unseen, moving anatomies with various dimensions. Fusing this work with the research of F. Lecomte, allowed to automatically perform endovascular navigation in close to real-world conditions involving moving anatomies whose movement was estimated from non-contrasted fluoroscopic

images.
In these conditions, our agent demonstrated good generalization capabilities, with success rates $> 95\%$, outperforming previously developed works which reported a success rate of 72% *(Chi et al. (2020b))* and 75% *(Karstensen et al. (2023))* when trying to generalize the task to unseen models in static conditions. Pushing the method even further, we designed a controller to simultaneously operate two devices, to access target locations impossible to reach using a single device (Chapter 8). Also in this case, the agent successfully learned the required task and generalized it to new, unseen anatomies. To the best of the author's knowledge, this is the first time a controller has been developed to automate this task and extend it to new geometries
The future directions of this research, listed below, focus on designing strategies to improve the developed controller and adapt it to real-world applications.
First of all, the algorithm's learning process could be enhanced by performing a grid search to identify the optimal network parameters and reduce, as a consequence, the training time. Considering the controller of the combined devices, an interesting development could involve the integration of the predicted tip shape of the catheter, computed by our cannulation assistance system, into the training process. This could help to learn the desired task more efficiently.
Secondly, the training strategy developed in this work could be refined by improving the realism of the simulation and by expanding the action space of the controller. Also in this case, the blood flow could be added to the simulation. Concerning the controller of a single instrument, the current implementation involves a constant positive guidewire speed. When the wrong branch is taken, the agent has no way to correct its trajectory. A possible adjustment strategy could be the incorporation of the speed of the device in the action space of the algorithm, giving the controller the possibility to pull backward the device when the wrong branch is navigated.
Several improvements can be made also in the simultaneous control of two instruments. First off all, the geometrical and mechanical characteristics of the catheter and guidewire can be adjusted to simulate real devices. In the initial implementation of the work, a simplified shape for the instruments was chosen to facilitate the development of the training strategy. However, to obtain a system transferable to the real world, this is a fundamental step. Additionally, more complex anatomies can be incorporated, and vessel deformation can be considered to better reflect real-world scenarios. Furthermore, to obtain a more precise controller and reduce the potential damage to the vessels, a strategy to reduce contact forces between the tip and the vessel wall could be implemented. Finally, the controller of a single device and the controller of the combined devices could be integrated into the same system, to obtain a strategy to autonomously navigate the medical tools from the insertion point to the target location.
To implement our control strategy in the real world, it is necessary to apply sim-to-real techniques to our learning strategy and establish a reliable experimental setup. A first step

toward this direction is presented in Section 6.5. However, the failed case shows that some additional work needs to be done to better parametrize the physical properties of the instrument. Furthermore, a closed-loop control strategy is necessary to reliably perform tests in the real world. To this purpose, the accuracy of the FBG system must be improved. A first possible method involves the refinement of our neural network correction technique (Section 4.3.2) by changing the network architecture or the training strategy. Another solution can entail the fusion of the two strategies described in Section 4.3, implementing a method to rigidly register on the centerline of the navigated vessel the shape reconstructed by our neural network. A third potential strategy, inspired by work present in the literature *(Ha et al. (2021); Ourak et al. (2021))*, would be to integrate the shape reconstructed by our neural network with additional information sources, such as electromagnetic sensors or fluoroscopic images.
Finally, to maximize the chances of transferring the control learned in the simulated environment to the real world, robust sim-to-real techniques must be integrated into the learning process of the agent, adding for example sources of noise in the observation space.

A common point for the future steps of both systems developed in this work is enhancing the realism of the simulated environment. While certain aspects, such as fine-tuning the material properties of the catheters and guidewires, can definitely improve the system's reliability, others — like accurately simulating blood flow or implementing a deformation model for the vessel wall — may introduce more complexity than benefits. These elements require intricate simulations that can significantly increase simulation time, potentially compromising the real-time requirements. Another aspect that is important to consider is that both systems are driven by data acquired from the real world (*e.g.,* the shape and position of the devices navigating the vessel provided by the FBG or the vessel motion reconstructed from the fluoroscopic images). These data intrinsically provide information about the vessel wall deformation and the blood flow, potentially reducing the need for more complex simulations.

Chapter 10

Brief summary in French

Les interventions endovasculaires représentent la principale solution thérapeutique pour traiter la majorité de pathologies cardiovasculaires, qui, avec une incidence de 17.9 millions de décès par an *(World Health Organization (2021))*, constituent la première cause de mortalité mondiale. Le succès de ces interventions est dû à leur efficacité et à leur caractère peu invasif. Cependant, la complexité des interventions endovasculaires constitue un facteur limitant, nécessitant une formation approfondie et l'accès à des installations spécialement équipées. Au cours de l'intervention, des dispositifs tubulaires fins, appelés cathéters et fil-guide (Fig. 10.1), sont navigués à l'intérieur des artères du patient pour atteindre la cible. Ces instruments sont manipulés manuellement depuis leur extrémité proximale, placée à l'extérieur du patient au niveau de l'incision, et présentent une tip avec une forme adaptée à l'anatomie naviguée. Pendant l'intervention, le seul retour visuel disponible pour localiser les instruments à l'intérieur du corps est fourni par des images fluoroscopiques 2D. L'ensemble de la procédure endovasculaire peut être divisé en trois étapes principales : la navigation dans les gros vaisseaux, la phase de canulation sélective, qui consiste à atteindre l'entrée de petits vaisseaux latéraux à partir de l'artère principale, et la navigation dans les petits vaisseaux latéraux. Dans toutes les phases de l'intervention, l'aspect crucial est de positionner correctement la pointe de l'instrument pour naviguer dans le vaisseau souhaité sans endommager la paroi vasculaire. Lors de la navigation dans les petits vaisseaux, plusieurs bifurcations doivent être surpassés, et pour parcourir la branche désirée, la pointe incurvée de l'outil chirurgical doit être correctement orientée. À ce stade, la longueur insérée du cathéter peut atteindre 1 mètre, plaçant le point de contrôle de l'instrument loin de sa pointe. De plus, les vaisseaux imposent des contraintes à l'instrument, rendant son contrôle très difficile. Durant la phase de canulation, deux instruments sont contrôlés ensemble pour atteindre l'entrée d'un petit vaisseau latéral. Lorsqu'un dispositif est inséré dans l'autre, la forme de chaque instrument et leur rigidité relative influencent leur forme combinée. Une fois la zone proche de la cible atteinte, le clinicien doit anticiper

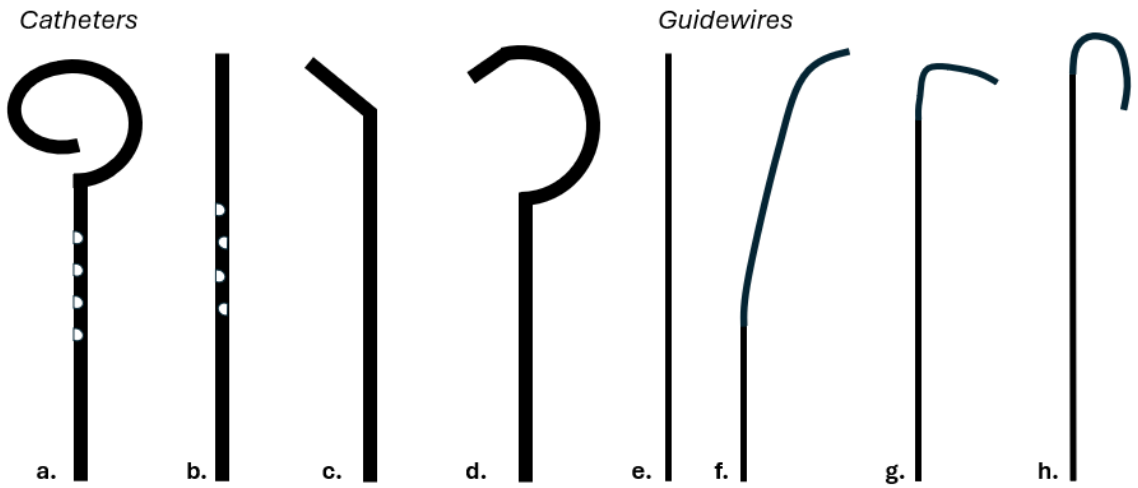

Fig. 10.1 Types de cathéters et de guides. a. Cathéter "flush" b. Cathéter d'échange c. Cathéter sélectif à courbe simple d. Cathéter sélectif à double courbe e. Guide droit f. Guide angulé M g. Guide BK h. Guide J de 2 mm

le meilleur moment pour arrêter l'avancée des dispositifs combinés et retirer le fil-guide, afin de laisser le cathéter reprendre sa forme d'origine et permettre à sa pointe d'atteindre l'ostium du vaisseau latéral cible. Cette opération peut nécessiter plusieurs tentatives, et il est parfois nécessaire de la répéter plusieurs fois au cours d'une même intervention. Un autre défi provient de la quantité limitée d'informations dont dispose le clinicien pendant la procédure, qui proviennent uniquement d'images 2D. De plus, l'acquisition de ces images implique l'utilisation de rayons X, dangereux pour la santé du patient et du clinicien, ainsi que l'utilisation de produit de contraste, nocif pour les reins du patient. Face à ces limitations, l'objectif de ce travail est de développer un système capable d'assister le clinicien tout au long de l'intervention, en concevant des algorithmes qui automatisent certaines phases de l'intervention et fournissent une aide visuelle dans d'autres. L'objectif final est de réduire le temps d'intervention et, par conséquent, le nombre d'injections de produit de contraste et l'exposition aux radiations, tant pour le patient que pour le clinicien. Ce travail est composé de trois axes principaux :

1. Le développement d'images fluoroscopiques augmentée pour assister le clinicien pendant la phase de canulation.

2. Le développement d'un algorithme de contrôle autonome pour manipuler un seul dispositif lors de la navigation dans les petits vaisseaux.

3. Le développement d'un algorithme de contrôle autonome capable de gérer la navigation combinée de deux dispositifs pendant la phase de canulation.

Le premier est essentiellement un système d'assistance, dans lequel les actions sont décidées et exécutées par le clinicien, soutenu par nos images fluoroscopiques améliorées. Le cœur des deuxième et troisième systèmes peut être décrit à travers la théorie du contrôle: cette théorie est en effet adaptée aux cas où il est nécessaire de guider le système de l'état actuel – par exemple, le point d'insertion – à l'état final souhaité – par exemple, la cible. Le contrôleur developpé dans ce travail est un algorithme autonome qui interagit directement avec les dispositifs chirurgicaux et l'anatomie. Il reçoit en entrée des informations concernant l'anatomie à naviguer ainsi que la forme et la position de l'instrument navigué et fournit en sortie le contrôle nécessaire pour atteindre l'objectif. Dans ce travail en particulier, nous avons adopté des techniques d'apprentissage par renforcement profond (Deep Reinforcement Learning, DRL) pour obtenir le contrôle optimal. Ces algorithmes présentent de nombreux points communs avec la théorie classique du contrôle optimal, mais ils sont mieux adaptés aux situations dans lesquelles il est difficile de construire une représentation précise du problème, comme dans le cas de la modélisation de la physique des cathéters et des guides naviguant à l'intérieur des vaisseaux. Dans ce cas, obtenir une modélisation précise peut en effet être très difficile et des erreurs peuvent être introduites entre le phénomène réel et sa représentation.

Pour développer l'ensemble des systèmes de cette étude, plusieurs composants sont nécessaires, allant d'un environnement simulé fiable, capable de reproduire aussi fidèlement que possible la physique des dispositifs réels naviguant dans l'anatomie, à des stratégies pour mesurer la forme et la position des cathéters et des guidewires et intégrer ces données dans la simulation, de critères garantissant la navigation vers la cible lorsqu'une bifurcation est atteinte, à des stratégies permettant d'obtenir un contrôleur capable de gérer diverses anatomies.

10.1 Environnement simulé

L'un des aspects fondamentaux de ce travail est de modéliser avec précision la navigation des dispositifs à l'intérieur des vaisseaux (Fig. 10.2). Cela constitue une exigence essentielle pour les deux systèmes développés dans cette étude. Lors de la création d'images fluoroscopiques améliorées, notre objectif est de montrer au clinicien la forme et la position des dispositifs dans des conditions différentes de celles actuelles. Sans un modèle fiable des instruments chirurgicaux, ce système ne peut pas être développé. En ce qui concerne le système de navigation autonome, notre objectif est de développer notre contrôleur dans un environnement simulé, puis de transférer le comportement appris au monde réel. Comme expliqué dans la section précédente, nous avons conçu un contrôleur basé sur le DRL, qui nécessite d'une phase d'entraînement pour apprendre la tâche souhaitée. Cela implique plusieurs épisodes de navigation, au cours desquels l'agent guide les dispositifs depuis le point d'insertion jusqu'à la

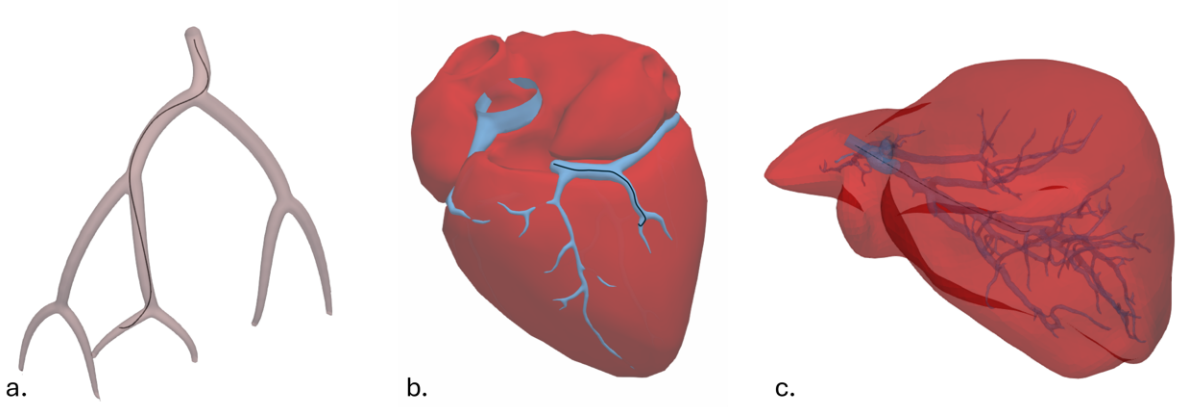

Fig. 10.2 Exemples des environnements simulés développés dans ce travail, montrant la navigation des guides à l'intérieur des parois vasculaires dans trois scénarios : une anatomie synthétique (a), les artères coronaires (b) et les vaisseaux hépatiques (c).

cible et interagit avec l'environnement. Réaliser ces expériences dans le monde réel peut être très coûteux, tant en termes de temps que de ressources, surtout lors des premières étapes de l'apprentissage, lorsque l'agent effectue uniquement des actions aléatoires. En solution, nous avons décidé d'effectuer l'entraînement de notre algorithme dans un environnement simulé, ce qui permet de réduire à la fois le temps d'entraînement et les coûts. Pour que cela fonctionne et soit transférable au monde réel, l'environnement simulé doit reproduire la physique du problème réel aussi précisément que possible.

10.1.1 Modèle de la physique des instruments chirurgicaux

Lorsque le modèle d'un phénomène physique est nécessaire, la première étape fondamentale consiste à identifier un modèle mathématique adapté pour simuler et prédire certains aspects du comportement du système (Fig. 10.3). Ce modèle est une abstraction de la réalité physique et sert à réduire la complexité du système physique d'origine en filtrant les informations non pertinentes pour l'objectif de l'étude. Les étapes suivantes impliquent la discrétisation du modèle mathématique trouvé et la résolution du système d'équations résultant de ce modèle. Chacune de ces phases introduit cependant une erreur : la modélisation introduit une erreur de modélisation ($e_{modeling}$), la discrétisation introduit une erreur de discrétisation ($e_{discretization}$), et la résolution introduit une erreur de résolution du système ($e_{resolution}$).

Les systèmes complexes sont généralement représentés par des équations aux dérivées partielles (EDP), qui sont des équations contenant une fonction inconnue de deux ou plusieurs variables ainsi que leurs dérivées partielles par rapport à ces variables. Résoudre ces problèmes analytiquement est souvent impossible; c'est pourquoi plusieurs techniques ont été développées

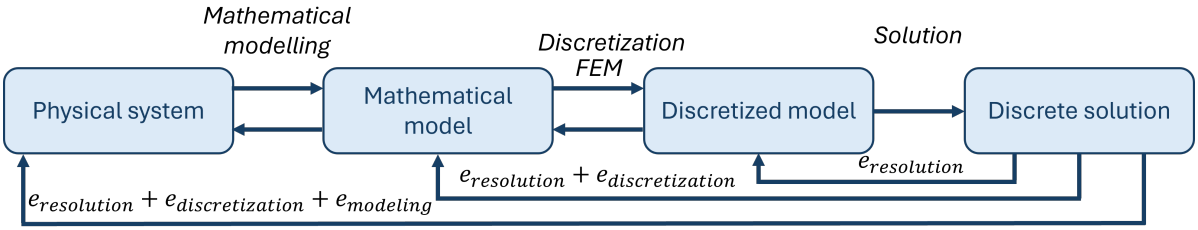

Fig. 10.3 Schématisation du processus suivi pour modéliser un problème physique. La première étape du processus consiste à identifier un modèle mathématique adapté pour simuler le comportement du système. Ce modèle mathématique est ensuite discrétisé, et, à partir de ce système discrétisé, on calcule la solution discrète du problème initial. Chacune de ces phases introduit une erreur : le modèle génère des différences entre le phénomène réel et sa représentation ($e_{modeling}$), la discrétisation introduit une erreur entre le système continu et le système discret ($e_{discretization}$), et le calcul de la solution engendre des erreurs de résolution ($e_{resolution}$).

pour évaluer ces équations. Parmi celles-ci, la méthode des différences finies et la méthode des éléments finis sont particulièrement intéressantes et sont utilisées dans ce travail pour la résolution des EDP et en particulier:

- La discrétisation du domaine spatial est réalisée par la méthode des éléments finis (MEF), qui permet de traiter un problème avec des géométries complexes. Cela conduit à la composition d'un système d'équations algébriques linéaires ou non linéaires.
- La discrétisation du domaine temporel, cartésienne, est réalisée par la méthode des différences finies (MDF), ce qui permet de résoudre la dynamique éventuelle du problème.

Nous avons d'abord comparé différents modèles pour la simulation de la physique des cathéters et des guides, et avons finalement choisi de modéliser nos dispositifs en utilisant le modèle de poutre de Timoshenko.
Pour nos deux applications, il est fondamental de maintenir une performance en temps réel : d'une part pour produire des images fluoroscopiques améliorées en temps réel, et d'autre part pour réduire le temps d'entraînement de l'algorithme de contrôle. Pendant la navigation, les dispositifs sont constamment en contact avec la paroi du vaisseau et entre eux. La tâche de simuler les contacts entre objets dans un délai acceptable est l'un des défis majeurs de l'application de la méthode des éléments finis (MEF). Pour modéliser les contacts entre les dispositifs et les parois du vaisseau, nous avons adopté un modèle à pénalités. La complexité computationnelle de ce calcul augmente avec le nombre de contacts. Pendant la navigation, lorsque le guide est inséré à l'intérieur du cathéter, le nombre de contacts entre ces deux instruments devient extrêmement élevé, ce qui rend l'utilisation de la méthode à pénalités

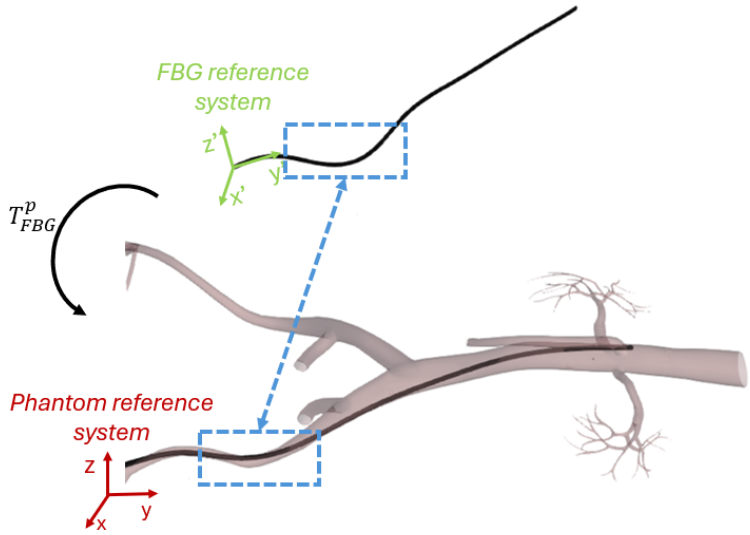

Fig. 10.4 Enregistrement rigide de la forme de la fibre, reconstruite dans le système de référence FBG et affectée par des erreurs de torsion, dans le système de référence du fantôme. La matrice T^p_{FBG}, calculée à l'aide du *coherent point drift algorithm*, permet d'effectuer cette transformation. Dans le carré bleu, un exemple de la région présentant une courbure similaire et utilisée pour l'enregistrement rigide est montré.

prohibitive. Pour cette raison, afin de modéliser les contacts entre les deux instruments, nous avons adopté un modèle composite avec des caractéristiques mécaniques variables pour simuler la physique du guide inséré à l'intérieur du cathéter en tant que modèle unique.

10.2 Le système de mesure

Le système d'imagerie fluoroscopique amélioré et les algorithmes de contrôle autonome, reposent sur la forte hypothèse de connaître la forme et la position en 3D des instruments naviguant dans les vaisseaux du patient, ainsi que la forme en 3D de l'anatomie. Lorsqu'il est nécessaire d'accéder à de petits vaisseaux latéraux, il est fondamental d'identifier le meilleur moment pour retirer le guide et permettre au cathéter de retrouver sa forme originale. À ce stade, il est crucial de connaître avec précision la forme et la position des instruments à l'intérieur de l'anatomie pour réaliser la cannulation. Lors de la navigation dans de petits vaisseaux, il est nécessaire de bien orienter la pointe de l'instrument pour accéder la branche souhaitée d'une bifurcation. Pour atteindre la cible, plusieurs bifurcations doivent être franchies, chacune se situant dans un plan différent, ce qui empêche de les observer toutes depuis une seule image fluoroscopique en 2D. Par conséquent, pour utiliser les systèmes proposés dans ce travail dans le monde réel, la première exigence est de disposer d'une méthode fiable pour reconstruire à

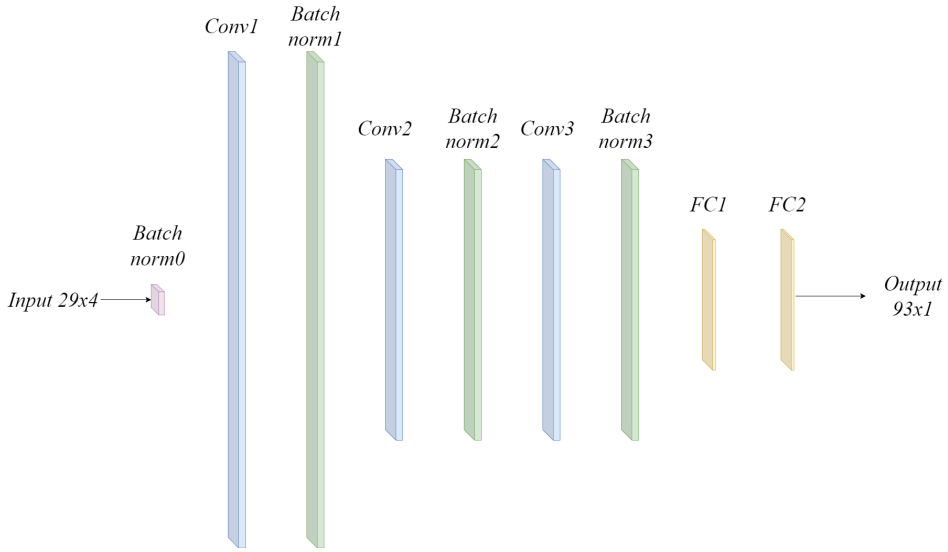

Fig. 10.5 Configuration finale du réseau de neurones utilisé pour reconstruire la forme de la fibre optique. Le réseau de neurones est composé de trois couches convolutionnelles, chacune suivie d'une couche de normalisation par lots, et de deux couches entièrement connectées (FC1, FC2). Le réseau de neurones prend en entrée un tableau de longueurs d'onde de dimension 29×4 obtenu à partir des données brutes des FBGs et donne en sortie un tableau de dimension 93×1 contenant la position relative de 31 points le long de la forme, chaque point étant fourni comme un déplacement x, y, z par rapport au point précédent.

la fois l'anatomie en 3D et la forme et la position en temps réel des instruments. Alors que la géométrie en 3D peut être obtenue à partir d'images préopératoires en 3D, telles que la tomodensitométrie ou l'imagerie par résonance magnétique (IRM), l'acquisition de la forme et de la position en 3D en temps réel des instruments nécessite un système de mesure complexe. Plusieurs choix sont disponibles aujourd'hui, allant de l'utilisation d'images fluoroscopiques à l'utilisation de capteurs comme les capteurs électromagnétiques (EM) ou les fibres optiques. Étant donné leurs excellentes caractéristiques théoriques, incluant des mesures distribuées, une immunité électromagnétique et une taille réduite, des capteurs à réseau de Bragg en fibre optique (FBG) ont été adoptés. Cependant, cette technologie présente encore des limitations importantes, causant des erreurs de reconstruction de forme qui impactent négativement nos méthodes. Pour surmonter ces limitations, nous avons proposé deux solutions possibles : une technique de recalage rigide pour garantir la reconstruction de la forme des dispositifs à l'intérieur des vaisseaux navigués, et le développement d'un réseau de neurones capable de reconstruire la forme des dispositifs à partir des mesures brutes des capteurs. Dans le premier cas, la forme reconstruite par les FBG est enregistrée de manière rigide *(Birkfellner et al. (2007); Yaniv (2008); Zollei et al. (2001))* sur la ligne centrale des vaisseaux, extraite de son modèle 3D (Fig. 10.4). Cependant, cette technique garantit uniquement que la forme

reconstruite est globalement positionnée à l'intérieur du vaisseau, mais elle ne garantit pas que tous les points constituant la forme se trouvent à l'intérieur de la paroi du vaisseau et ne corrige pas les erreurs introduites par la torsion. La deuxième méthode consiste à utiliser un réseau de neurones (RN) qui prend en entrée un array de dimension 29×4, contenant les longueurs d'onde de Bragg collectées par les fibres optiques, et donne en sortie la forme de l'instrument (Fig. 10.5). Ces deux stratégies ont démontré leur capacité à améliorer la précision des mesures FBG.

10.3 Images fluoroscopiques améliorées

Les systèmes de réalité mixte (RM) et de réalité augmentée (RA) capables d'améliorer et d'enrichir les informations fournies au clinicien lors des interventions endovasculaires deviennent de plus en plus populaires *(García-Vázquez et al. (2018); Soulami et al. (2016); Yoon et al. (2017))*. Cependant, les travaux présentés dans la littérature sont principalement des systèmes de visualisation ou des simulateurs qui n'offrent pas un support supplémentaire au clinicien. Pour surmonter ces limitations, nous avons développé un système qui non seulement superpose des informations anatomiques supplémentaires sur les images fluoroscopiques, mais qui montre également la forme prédite des dispositifs pour une action prédéterminée, offrant ainsi au clinicien une assistance en temps réel pendant l'intervention (Fig. 10.7 droite).

Plus précisément, nous avons mis au point une méthode de navigation endovasculaire basée sur les données, qui associe des modèles physiques à un système de détection de forme basé sur la technologie des FBGs. Notre système permet de prédire la position et la forme de l'outil dans des situations qui ne se sont pas encore produites. La cannulation de l'artère rénale gauche est un bon exemple pour illustrer le fonctionnement du système (Fig. 5.1). Lorsqu'on approche de la région où le vaisseau secondaire prend son origine, le système commence à afficher la forme du cathéter telle qu'elle serait après le retrait du guide, aidant ainsi le clinicien à maximiser les chances de réussite d'une cannulation en un seul essai.

Le système développé dans ce travail est essentiellement un outil d'assistance et prend en entrée quatre informations : un modèle de la physique des instruments chirurgicaux, la forme des dispositifs, en particulier celle du cathéter, reconstruite à l'aide des capteurs FBG, les images fluoroscopiques non contrastées, et le modèle 3D de l'anatomie naviguée. Ce système utilise le modèle basé sur la physique du cathéter et du guide décrit dans la Section 10.1 pour deux objectifs : simuler la forme du cathéter après retrait du guide et compenser les erreurs introduites par le système FBG. Les propriétés mécaniques des dispositifs ont été déterminées en fonction des caractéristiques réelles des instruments. Le module de Young de tous les dispositifs a été choisi au moyen d'un test de flexion, effectué à la fois dans l'environnement

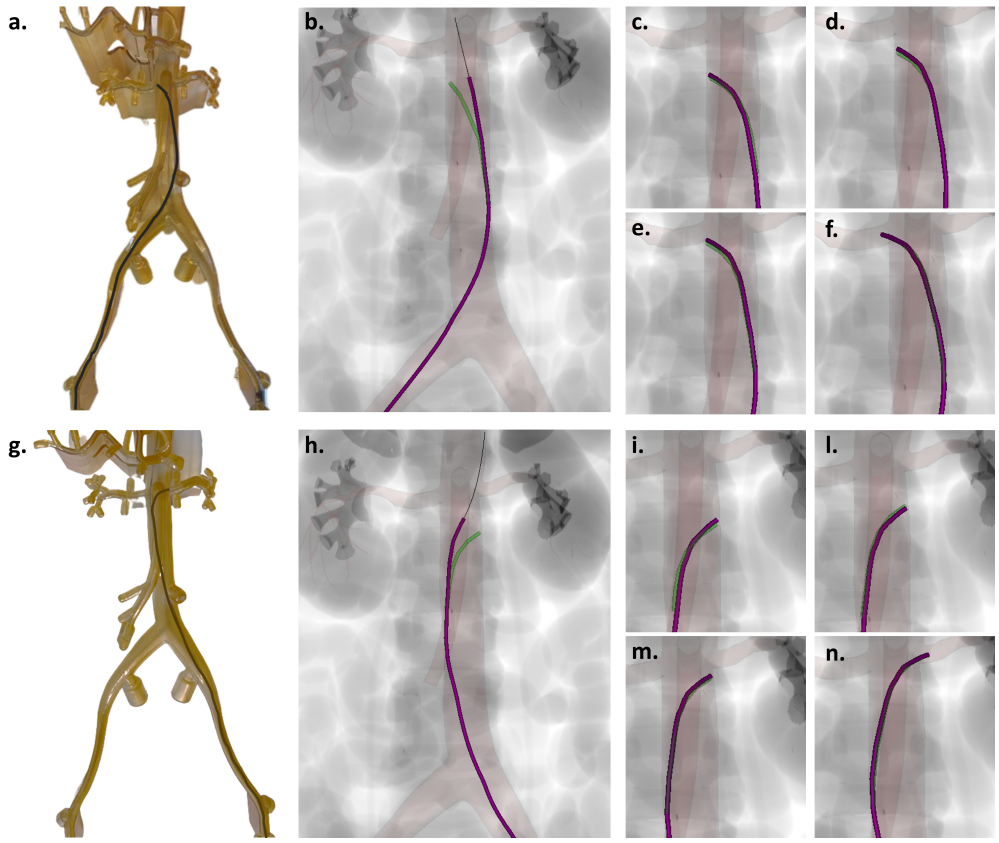

Fig. 10.6 Étude de validation. Le guide et le cathéter équipés de la fibre optique sont insérés dans un fantôme dans l'artère iliaque droite (en haut) et gauche (en bas). Le système, en plus de la forme et de la position réelles des instruments (Fig. b et h en violet), montre la forme prédite du cathéter après le retrait du guide (Fig. b et h en vert). Les Fig. c, d, e, f, i, l, m, et n montrent la comparaison entre la forme prédite du cathéter calculée par notre système (en vert) et la forme réelle du cathéter obtenue après le retrait du guide (en violet) et acquise par le système à fibre optique. Une erreur moyenne 3D de 2.4 ± 1.3 mm et une erreur 2D de 1.1 ± 0.7 mm ont été mesurées entre les deux formes.

réel et dans l'environnement simulé. Le module de Young des modèles a été modifié de manière itérative jusqu'à trouver la valeur qui minimisait la différence entre les données simulées et réelles.

Les données de forme du cathéter sont recueillies par le système FBG toutes les 0.1 secondes. Ces informations de forme sont ensuite transformées dans le même système de référence que l'anatomie par le enregistrement rigide introduit dans la section 10.2 et régularisées en utilisant la forme et la position simulées de l'instrument.

À notre connaissance, c'est la première fois qu'un système offrant ce type d'assistance est proposé : ce soutien supplémentaire peut réduire le nombre de tentatives échoué et, par conséquent, raccourcir la durée de l'intervention, ainsi que le nombre d'injections d'agent

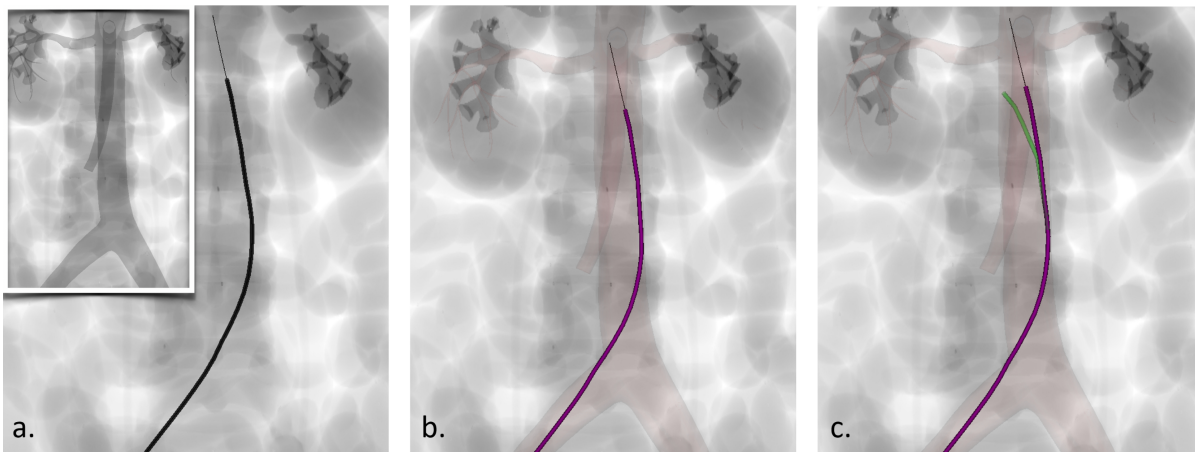

Fig. 10.7 Illustration de l'étude utilisateur : a. Tout d'abord, on demande à l'utilisateur de naviguer en utilisant l'image fluoroscopique sans agent de contraste, ne voyant qu'une *road-map* statique des vaisseaux dans le coin supérieur gauche ; b. Ensuite, l'utilisateur navigue en utilisant une image fluoroscopique augmentée de l'anatomie 3D des vaisseaux ; c. Enfin, l'utilisateur essaie d'atteindre la cible en utilisant notre approche de navigation prédictive. Chaque utilisateur a effectué la cannulation trois fois, chaque fois avec l'un des trois guidages d'image illustrés dans ces images. Pour tous les sujets, l'utilisation de notre système a réduit le temps nécessaire pour effectuer la cannulation.

de contraste et l'exposition aux rayons X pour le patient et les soignants. Pour évaluer la robustesse et l'utilité du système, une étude de validation a été réalisée, rapportant un erreur 3D de 2.4 $\pm$ 1.3 *mm* et une précision de 1.1 $\pm$ 0.7 *mm* sur le plan de l'image fluoroscopique, ainsi qu'une étude concernant quatre utilisateurs qui a montré une réduction moyenne du temps d'intervention de 56 % en utilisant notre système.

10.4 Algorithme de contrôle autonome

Lors de la navigation dans les petits vaisseaux, l'une des principales solutions adoptées aujourd'hui pour réduire le temps d'intervention est l'utilisation de systèmes robotiques *(Antoniou et al. (2011); Bonatti et al. (2014); Duan et al. (2023); Kazanzides et al. (2008))*, qui assurent un contrôle précis des dispositifs et permettent de réaliser l'intervention à distance de la table d'opération, réduisant ainsi l'exposition aux radiations pour le clinicien, mais pas pour le patient. Cependant, ces systèmes sont principalement des dispositifs maître-esclave, qui n'apportent pas de soutien supplémentaire au clinicien (Fig. 10.8). Pour surmonter cette limitation, au cours de ce projet de doctorat, nous avons proposé un système de navigation autonome dans le but d'automatiser les interventions endovasculaires. Plusieurs travaux présentés dans la littérature ont tenté de répondre à cette problématique, mais la plupart présente deux lacunes

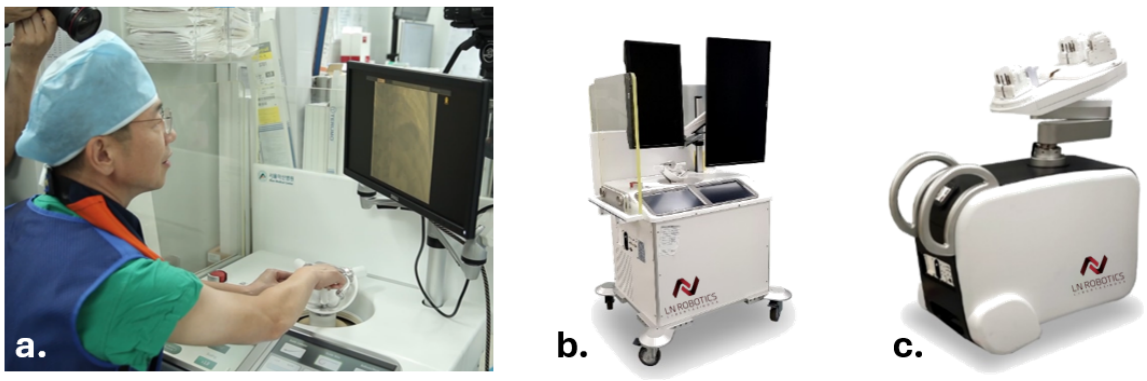

Fig. 10.8 Robots endovasculaires – Joystick utilisé par le clinicien pour contrôler le robot endovasculaire (a), composants de commande principaux du système utilisés par le soignant (b), et partie suiveuse du système qui répond aux commandes du clinicien.

communes *(Karstensen et al. (2020); Tian et al. (2023); Wang et al. (2022))*: l'utilisation d'images fluoroscopiques comme entrée du système, ce qui peut entraîner des ambiguïtés dans la reconnaissance de la position et de l'orientation du dispositif, et une reduite capacités de généralisation. La généralisation en particulier est très importante pour l'application clinique de ces systèmes. En effet, même une même structure anatomique peut présenter des variations de forme importantes d'un patient à l'autre, et même chez un seul patient, les vaisseaux peuvent subir des déplacements et des déformations importantes dus aux mouvements physiologiques. Pour surmonter cette limitation, *Karstensen et al. (2023)* ont utilisé des réseaux de neurones récurrents pour concevoir un contrôleur basé sur l'apprentissage, capable de naviguer dans diverses géométries. Après 3.5 millions d'étapes d'exploration sur divers arcs aortiques, le contrôleur a rapporté un taux de réussite de 75 % pour la navigation dans différents arcs aortiques, qui est tombé à 29 % lorsque des vaisseaux d'un patient réel ont été utilisés. *Chi et al. (2020b)* ont mis en œuvre une méthode d'apprentissage par imitation adversariale générative visant à réaliser la cathétérisation de l'artère brachio-céphalique et de l'artère carotide commune gauche. Ils ont entraîné le réseau en utilisant des démonstrations d'experts sur une reproduction de l'arc aortique et ont obtenu un taux de réussite de 72.2 % lorsque le type d'arc aortique a été modifié.

Dans ce travail, nous avons développé un algorithme basé sur l'apprentissage par renforcement profond (DRL), capable d'appliquer le comportement appris à n'importe quelle anatomie vasculaire, en supposant qu'il soit possible de naviguer vers différentes zones cibles à l'aide des outils choisis. Notre algorithme de contrôle prend en entrée deux informations : l'anatomie à naviguer, acquise par exemple à partir d'images IRM, et la forme des dispositifs obtenue grâce à l'utilisation de capteurs FBG. Le développement de ce contrôleur dans un environnement virtuel a constitué l'axe principal de cette thèse de doctorat, et trois versions

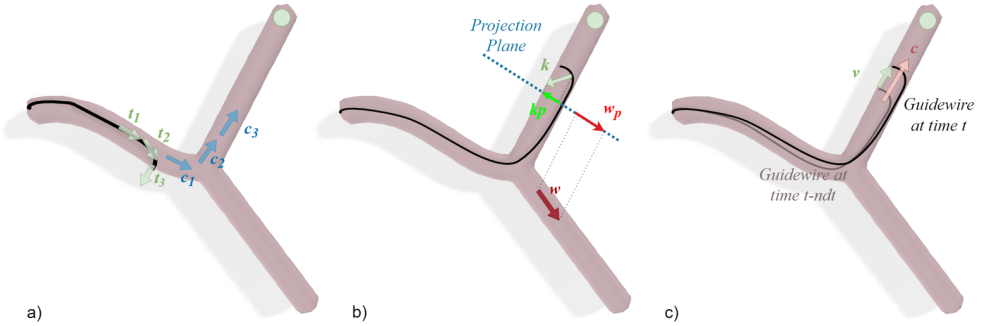

Fig. 10.9 Notre espace d'observation est composé de : 1) $t_i \cdot c_j$, avec $i \in [1;3] \in \mathbb{Z}$ et $j \in [1;3] \in \mathbb{Z}$ (a), 2) la distance normalisée entre la pointe du guide-fil et la cible, 3) l'action choisie, 4) $k_p \cdot w_p$ (b), 5) $v \cdot c_i$ (c).

différentes de l'algorithme ont donc été développées, capables de gérer des conditions de plus en plus complexes.

10.4.1 Navigation autonome du guide dans un environnement statique

La première version de l'algorithme de contrôle impliquait le développement d'un système qui, après l'entraînement, était capable d'effectuer la navigation dans toutes les anatomies compatibles avec le guide choisi. Trois hypothèses principales ont été faites : les vaisseaux ont un rayon presque constant, les bifurcations ont toujours 2 vaisseaux de sortie, et l'anatomie ne bouge pas et ne se déforme pas pendant la navigation.
Le cœur du système est composé de trois éléments :

- L'environnement simulé, dans lequel l'algorithme est entraîné, et qui est décrit dans la Section 10.1
- Un *espace d'observation presque invariant*, qui permet à l'agent d'observer uniquement des informations sur la position relative de l'instrument à l'intérieur des vaisseaux, et qui prend la forme suivante :

$$\Omega = \{\zeta_t, \zeta_{t-ndt}, \lambda_t, \lambda_{t-ndt}, a_t, \omega, d_v\}$$

  - Soit $\mathbf{t}_i, i \in [1,N]$ le vecteur tangent à la coordonnée $\mathbf{x}_i$ le long de la pointe du guidewire, et $\mathbf{c}_j, j \in [1,N]$ le vecteur tangent de la ligne centrale à la position $\mathbf{x}j = \mathbf{x}i + \mathbf{h}$. Nous définissons $\zeta_i = \mathbf{t_i} \cdot \mathbf{c_i}$ pour tout $i \in [1,N] \in \mathbb{Z}$.
  - Nous définissons ensuite $\zeta_m = [\zeta_1, \zeta_2, ..., \zeta_N]_m$, avec $m \in \{t; t-ndt\}$.
  - λ_t et λ_{t-ndt} représentent la distance géodésique entre la pointe du guidewire et la cible aux instants t et $t-ndt$.

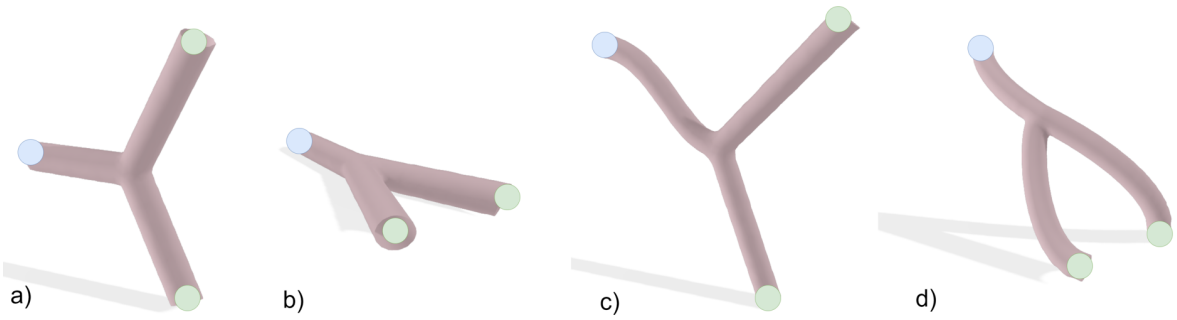

Fig. 10.10 Les géométries choisies pour entraîner l'agent et couvrir l'ensemble de l'espace d'observation. Dans chaque modèle, le point bleu représente le point de départ, et le point vert la position cible.

  - a_t est l'action qui détermine la transition du système de s_{t-ndt} à s_t.
  - $\omega = \mathbf{k}_p \cdot \mathbf{w}_p$, où $\mathbf{k}_p$ est la projection du vecteur de courbure $\mathbf{k}$ de la pointe du guide et $\mathbf{w}_p$ est la projection du vecteur décrivant la direction de la mauvaise branche $\mathbf{w}$ sur un plan perpendiculaire à la ligne centrale de la branche menant à la cible (Fig. 10.9 b).
  - $d_v = \mathbf{v} \cdot \mathbf{c}$, où $\mathbf{v}$ décrit la vitesse actuelle du guide-fil et $\mathbf{c}$ le vecteur tangent à la ligne centrale près de la pointe du guide-fil (Fig. 10.9 c).

- Un ensemble de bifurcation, utilisés pour entraîner l'agent et montrés dans la Figure 10.10.

- Une fonction de récompense, qui prend la forme suivante :

$$r(s_t, a_t) = \underbrace{\frac{2}{1+e^{5(\omega-0.1)}} - 1}_{a} + \underbrace{0.5\left(1 - \frac{\lambda_t}{\lambda_0}\right)}_{b} + \underbrace{(-0.2|a_t|)}_{c} \tag{10.1}$$

  Dans la fonction de récompense, la partie a encourage l'agent à obtenir une direction de la pointe $\mathbf{k}_p$ opposée à $\mathbf{w}_p$ (voir Fig. 10.9). La partie b de la récompense augmente à mesure que la cible est approchée, tandis que la partie c dissuade l'agent de faire pivoter l'instrument lorsque cela n'est pas nécessaire.

Notre algorithme d'apprentissage par renforcement, entraîné avec notre stratégie d'apprentissage *zero-shot learning*, a pu apprendre un contrôle généralisable à des anatomies vasculaires inconnues, sans besoin de réentraînement. Nous avons démontré notre méthode dans un environnement virtuel sur 5 systèmes vasculaires statiques différents, 4 synthétiques et un reproduisant les coronaires d'un cœur réel, et avons obtenu un taux de réussite moyen de 96%.

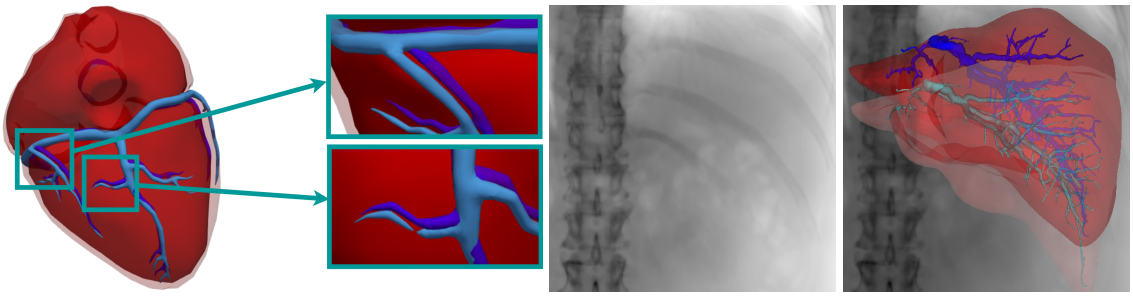

Fig. 10.11 Gauche : Visualisation du mouvement coronaire pendant un cycle cardiaque. Le volume cardiaque varie d'environ 12 % en 1 seconde pendant un cycle cardiaque. Droite : Anatomie du foie. La première image montre l'image fluoroscopique vue par le réseau neuronal. La seconde montre la position et la forme préopératoires du foie et de son système veineux (couleurs opaques) ainsi que la prédiction de la forme 3D (en couleur semi-transparente) à la fois de la forme du foie et de son arbre vasculaire. Les lignes centrales des veines sont également prédites, en temps réel, et utilisées par le contrôleur neuronal.

10.4.2 Navigation autonome dans des environnements dynamiques

Cette deuxième version du contrôleur visait à étendre l'applicabilité de notre algorithme précédemment développé à un plus large éventail d'anatomies. Le système précédent était en effet limité à la navigation dans des géométries statiques, avec un diamètre de vaisseau similaire à celui des anatomies d'entraînement. Pour surmonter ces limitations, nous avons étendu notre méthode pour apprendre à contrôler le dispositif même dans des anatomies vasculaires en mouvement et avec des dimensions différentes. La stratégie d'apprentissage par renforcement a donc été étendue dans deux domaines, tandis que l'espace d'observation et la fonction de récompense sont restés inchangés. Premièrement, nous avons augmenté la base de données d'entraînement en introduisant des variations de forme de l'anatomie d'entraînement au cours du processus d'entraînement (similaire aux approches sim-to-real). Cette variation de forme était continue dans l'espace et le temps afin d'éviter les discontinuités dans le champ de déplacement, qui auraient pu causer des problèmes dans la simulation. Deuxièmement, nous avons formalisé le processus de génération de formes en le rendant procédural, plutôt que manuel. De plus, dans ce travail, pour estimer le mouvement de l'anatomie naviguée, nous avons utilisé les méthodes développées par François Lecomte, membre de l'équipe MIMESIS, permettant de déduire le déplacement à partir d'images fluoroscopiques sans contraste *(Lecomte et al. (2023))*.

Le nouveau contrôleur a été testé dans deux scénarios simulés, un cœur battant, avec un mouvement généré dans un environnement simulé, et un foie déformé par le mouvement respiratoire, dont le mouvement a été estimé directement à partir d'images fluoroscopiques sans contraste (Fig. 10.11). Dans ces nouvelles anatomies dynamiques et inconnues, avec des

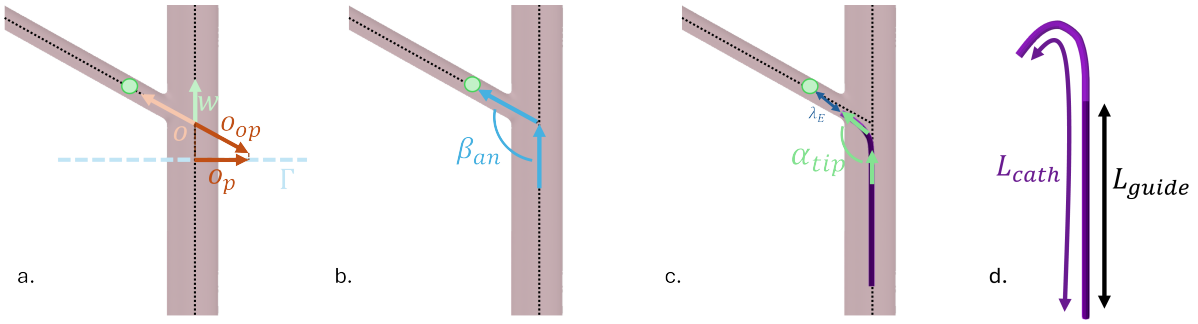

Fig. 10.12 Représentation des nouveaux termes de l'observation. $\omega_D = \mathbf{k}_p \cdot \mathbf{w}_p$ (a). Lorsque $\mathbf{w} \perp \Gamma$, $\mathbf{w}_p = \mathbf{o}_p$, où $\mathbf{o}_p$ est la projection sur le plan Γ d'un vecteur $\mathbf{o}$, qui a la même direction que le vecteur décrivant la direction menant à la cible ($\mathbf{o}$), mais avec une orientation opposée. β_{an} décrit l'angle entre la branche principale et la branche menant à la cible (b), α_{tip} représente la description de la forme de la pointe (c), λ_E la distance euclidienne entre la pointe de l'appareil et la position de la cible (b), et $L_{rel} = \frac{L_{guide}}{L_{cath}}$ (d).

dimensions de vaisseaux variables, le contrôleur a pu atteindre un taux de réussite moyen de 95%.

10.4.3 Cannulation sélective autonome

Dans le but de concevoir un système complet, capable d'automatiser toutes les phases de l'intervention, le contrôleur a été encore amélioré pour gérer le contrôle simultané de deux instrumemts, dans le but d'automatiser la procédure de cannulation sélective. Notre contrôleur a été entraîné sur un ensemble réduit d'anatomies synthétiques, présentant un petit vaisseau latéral se ramifiant à partir du vaisseau principal, en ayant pour objectif de généraliser le contrôle appris à de nouvelles anatomies inconnues.

À partir de la formulation développée pour la navigation des petits vaisseaux (Sections 10.4.1), l'espace d'observation a été modifié en fonction des exigences actuelles du contrôle et prend la forme suivante :

$$\Omega = \{\zeta_t, \zeta_{t-ndt}, \lambda_{E_t}, \lambda_{E_{t-ndt}}, a_t, \omega_D, d_v, \mathscr{A}_t, \mathscr{A}_{t-ndt}, \mathscr{F}, L_{rel}\}$$

Les termes ζt, $\zeta t - ndt$, a_t et d_v sont définis de manière cohérente avec ceux présentés dans la Section 10.4.1, tandis que les autres termes présentent quelques différences :

- Le terme ω_D est obtenu comme suit : $\omega_D = \mathbf{k}_p \cdot \mathbf{w}_p$, où $\mathbf{k}_p$ est la projection du vecteur de courbure $\mathbf{k}$ de la pointe et $\mathbf{w}_p$ est la projection du vecteur décrivant la direction de la mauvaise branche $\mathbf{w}$ sur un plan Γ perpendiculaire à la ligne centrale de la branche principale. Lorsque le vecteur de mauvaise direction $\mathbf{w}$ est perpendiculaire à Γ (Fig.

10.12 a), la magnitude de $\mathbf{w}_p$ prend des valeurs proches de zéro, rendant le calcul de ce vecteur, et de ω_D par conséquent, peu fiable. Pour contourner ce problème, $\mathbf{w}_p$ est dans ces cas pris comme la projection sur Γ d'un vecteur $\mathbf{o}$, qui a la même direction que le vecteur décrivant la direction menant à la cible, mais avec une orientation opposée (Fig. 10.12 a).

- La distance géodésique normalisée $\lambda_t, \lambda_{t-ndt}$ est ici remplacée par la distance euclidienne normalisée aux instants actuels (λ_{E_t}) et précédents ($\lambda_{E_{t-ndt}}$). Ce remplacement a été effectué car la distance euclidienne fournit, dans ce cas, des données plus informatives que la distance géodésique. Ces deux quantités sont normalisées par rapport à la distance à la cible initiale λ_{E_0}, calculée au début de chaque épisode d'entraînement.

- $\mathscr{A}_t = \cos(\alpha_{tip_t})$ et $\mathscr{A}_{t-1} = \cos(\alpha_{tip_{t-ndt}})$ sont introduits, représentant le cosinus de l'angle α_{tip} aux instants actuels t et précédents $t - ndt$.

- $\mathscr{F} = \cos(\beta_{an})$ est ajouté, ce qui indique le cosinus de l'angle β_{an}.

- L_{rel}, qui est obtenu comme le rapport entre la longueur du guide (L_{guide}) et la longueur du cathéter (L_{cath}).

La fonction de récompense prend la forme suivante:

$$r(s_t, a_t) = \underbrace{\frac{2}{1+e^{20|\rho|-6}} - 1}_{a} + \underbrace{\text{sgn}(dl)0.25\,(1-\lambda_{E_t})}_{b} - \underbrace{0.5\,(1-\lambda_{E_t})}_{c} + \underbrace{\frac{2}{2+e^{12*(\omega+0.65)}} - 0.5}_{d} - \underbrace{|a_{t_{rot}}| * 0.2}_{e}$$

Ici, $\rho = \alpha_{tip} - \beta_{an}$, mesuré en degrés, représente la différence entre l'angle actuel de la pointe et l'angle caractérisant la branche désirée, sgn représente la fonction signe et $dl = \lambda_{E_{t-ndt}} - \lambda_{E_t}$. Le terme a est une nouvelle version modifiée de la fonction sigmoïde prenant des valeurs dans $[-1;1] \in \mathbb{R}$ et récompense l'agent avec des scores proches de $+1$ lorsque ρ est proche de 0. Le module du terme b augmente à mesure que la distance euclidienne λ_{E_t} diminue et prend des valeurs positives si $\lambda_{E_t} < \lambda_{E_{t-ndt}}$ et négatives si $\lambda_{E_t} > \lambda_{E_{t-ndt}}$. Le terme c pénalise l'agent à chaque étape, avec une pénalité qui augmente à mesure que la cible est approchée, poussant le contrôleur à atteindre la cible le plus rapidement possible dans les situations où la forme optimale de la pointe nécessite que le guide soit avancé davantage que le cathéter (*i.e.,* lorsque la cible est située dans le grand vaisseau principal). Le terme d est une version modifiée du terme a de l'Équation (10.1). Les nouveaux coefficients inclus dans cette expression ont été

introduits pour renforcer un contrôle plus précis de l'orientation de la pointe des dispositifs et visent à réduire la plage des orientations de la pointe $\hat{\theta}$ considérées comme acceptables. Ces coefficients ont été déterminés par un processus d'essais et d'erreurs. Enfin, le terme e décourage l'algorithme d'exécuter des rotations aléatoires et inutiles.

Après l'entraînement, l'agent a été testé à la fois sur les anatomies d'entraînement et sur de nouvelles géométries simples, démontrant sa capacité à gérer le contrôle simultané de deux instruments dans tous les cas testés, posant ainsi les bases pour le développement de cette stratégie dans de nouvelles anatomies plus réalistes.

10.5 Conclusion

L'objectif principal du travail développé dans cette thèse de doctorat était de rechercher et de mettre en œuvre des solutions visant à assister les cliniciens lors des interventions endovasculaires. Malgré les nombreux avantages de ces procédures, leur exécution est généralement très complexe. C'est pourquoi le travail proposé dans ce manuscrit vise à faciliter la réalisation des interventions endovasculaires, dans le but de réduire le temps d'opération et, par conséquent, de minimiser l'exposition aux rayons X pour le patient et le clinicien. Plus précisément, deux systèmes principaux ont été développés : un système d'assistance pour soutenir le clinicien lors de la phase de canulation de l'intervention, et un ensemble d'algorithmes de contrôle conçus pour opérer de manière autonome les robots endovasculaires actuellement disponibles sur le marché. Ces robots sont en effet des dispositifs leader-suiveur entièrement contrôlés par le clinicien et ne sont pas capables d'automatiser des tâches ou d'offrir un soutien intraopératoire.

References

Abdelzaher, T., Diao, Y., Hellerstein, J. L., Lu, C., and Zhu, X. (2008). Introduction to control theory and its application to computing systems. *Performance Modeling and Engineering*, pages 185–215.

Affdl, J. C. H. and Kardos, J. L. (1976). The halpin-tsai equations: A review. *Polymer Engineering & Science*, 16(5):344–352.

Al-Ahmad, O., Ourak, M., Van Roosbroeck, J., Vlekken, J., and Poorten, E. V. (2020). Improved fbg-based shape sensing methods for vascular catheterization treatment. *IEEE Robotics and Automation Letters*, 5(3):4687–4694.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. (2017). Hindsight experience replay. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Antman, S. S. (2005). Problems in nonlinear elasticity. *Nonlinear Problems of Elasticity*, pages 513–584.

Antoniou, G. A., Riga, C. V., Mayer, E. K., Cheshire, N. J., and Bicknell, C. D. (2011). Clinical applications of robotic technology in vascular and endovascular surgery. *Journal of Vascular Surgery*, 53(2):493–499.

Astrom, K. J. and Murray, R. M. (2008). Feedback systems: An introduction for scientists and engineers.

Banchoff, T. and Lovett, S. (2010). *Differential geometry of curves and surfaces*. CRC Press.

Beliaev, M., Lalin, V., and Kuroedov, V. (2015). Geometrically nonlinear rods theory - comparison of the results obtained by cosserat-timoshenko and kirchhoff's rod theories. *Applied Mechanics and Materials*, 725-726:629–635.

Bellemare, M., Naddaf, Y., Veness, J., and Bowling, M. (2012). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47.

Bellman, R. (1957). A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684.

Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., and Grinspun, E. (2008). Discrete elastic rods. *ACM Transactions on Graphics*, 27.

Bhat, A., Kuang, Y. M., Gan, G. C. H., Burgess, D., and Denniss, A. R. (2015). An update on renal artery denervation and its clinical impact on hypertensive disease. *BioMed Research International*, 2015(1):607079.

Birkfellner, W., Figl, M., Kettenbach, J., Hummel, J., Homolka, P., Schernthaner, R., Nau, T., and Bergmann, H. (2007). Rigid 2d/3d slice-to-volume registration and its application on fluoroscopic ct images. *Medical physics*, 34:246–55.

Bonatti, J., Vetrovec, G., Riga, C., Wazni, O., and Stádler, P. (2014). Robotic technology in cardiovascular medicine. *Nature reviews. Cardiology*, 11.

Bourier, F., Fahrig, R., Wang, P., Santangeli, P., Kurzidim, K., Strobel, N., Moore, T., Hinkel, C., and AL-AHMAD, A. (2014). Accuracy assessment of catheter guidance technology in electrophysiology procedures: a comparison of a new 3d-based fluoroscopy navigation system to current electroanatomic mapping systems. *Journal of cardiovascular electrophysiology*, 25(1):74–83.

Bourier, F., Reents, T., AMMAR-BUSCH, S., Buiatti, A., Grebmer, C., Telishevska, M., Brkic, A., Semmler, V., Lennerz, C., Kaess, B., et al. (2015). Sensor-based electromagnetic navigation (mediguide®): How accurate is it? a phantom model study. *Journal of cardiovascular electrophysiology*, 26(10):1140–1145.

Boyer, F., De Nayer, G., Leroyer, A., and Visonneau, M. (2011). Geometrically Exact Kirchhoff Beam Theory: Application to Cable Dynamics. *Journal of Computational and Nonlinear Dynamics*, 6(4):041004.

Boyer, F. and Porez, M. (2015). Multibody system dynamics for bio-inspired locomotion: From geometric structures to computational aspects. *Bioinspiration & Biomimetics*, 10.

Boyer, F., Porez, M., and Khalil, W. (2006). Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. *IEEE Transactions on Robotics*, 22(4):763–775.

Boyer, F. and Primault, D. (2004). Finite element of slender beams in finite transformations: a geometrically exact approach. *International Journal for Numerical Methods in Engineering*, 59(5):669–702.

Brown, N. and Sandholm, T. (2018). Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359:418 – 424.

Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.

Buşoniu, L., de Bruin, T., Tolić, D., Kober, J., and Palunko, I. (2018). Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control*, 46:8–28.

Canero, C., Radeva, P., Toledo, R., Villanueva, J. J., and Mauri, J. (2000). 3d curve reconstruction by biplane snakes. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 563–566. IEEE.

Cardona, A. and Geradin, M. (1988). A beam finite element non-linear theory with finite rotations. *International Journal for Numerical Methods in Engineering*, 26(11):2403–2438.

Chi, W., Dagnino, G., Kwok, T. M. Y., Nguyen, A., Kundrat, D., Abdelaziz, M. E. M. K., Riga, C., Bicknell, C., and Yang, G.-Z. (2020a). Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2414–2420.

Chi, W., Dagnino, G., Kwok, T. M. Y., Nguyen, A., Kundrat, D., Abdelaziz, M. E. M. K., Riga, C., Bicknell, C., and Yang, G.-Z. (2020b). Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2414–2420.

Chi, W., Liu, J., Rafii-Tari, H., Riga, C., Bicknell, C., and Yang, G. Z. (2018). Learning-based endovascular navigation through the use of non-rigid registration for collaborative robotic catheterization. *International Journal of Computer Assisted Radiology and Surgery*, 13:855–864.

Cho, Y., Park, J., Choi, J., and Chang, D. E. (2022). Sim-to-real transfer of image-based autonomous guidewire navigation trained by deep deterministic policy gradient with behavior cloning for fast learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3468 – 3475. IEEE Robotics and Automation Society.

Cho, Y., Park, J.-H., Choi, J., and Chang, D. E. (2021). Image processing based autonomous guidewire navigation in percutaneous coronary intervention. In *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–6.

Condino, S., Ferrari, V., Freschi, C., Alberti, A., Berchiolli, R., Mosca, F., and Ferrari, M. (2012). Electromagnetic navigation platform for endovascular surgery: how to develop sensorized catheters and guidewires. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 8(3):300–310.

Crisfield, M. (1990). A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Computer Methods in Applied Mechanics and Engineering*, 81(2):131–150.

Dawson, S. L., Kaufman, J. A., and Meglan, D. (1996). An interactive training/simulator for interventional radiology. *Journal of Vascular and Interventional Radiology*, 7(1, Supplement):132–136.

de Silva, C. W. (2009). *Intelligent Control*, pages 4868–4891. Springer New York, New York, NY.

Delmas, C., Berger, M.-O., Kerrien, E., Riddell, C., Trousset, Y., Anxionnat, R., and Bracard, S. (2015). Three-dimensional curvilinear device reconstruction from two fluoroscopic views. In *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 9415, pages 100–110. SPIE.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664.

Dill, E. H. (1992). Kirchhoff's theory of rods. *Archive for History of Exact Sciences*, 44(1):1–23.

Dore, A., Smoljkic, G., Vander Poorten, E., Sette, M., Vander Sloten, J., and Yang, G.-Z. (2012). Catheter navigation based on probabilistic fusion of electromagnetic tracking and physically-based simulation. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3806–3811. IEEE.

Duan, W., Akinyemi, T., Du, W., Ma, J., Chen, X., Wang, F., Omisore, O., Luo, J., Wang, H., and Wang, L. (2023). Technical and clinical progress on robot-assisted endovascular interventions: A review. *Micromachines*, 14(1).

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR.

Duncan, R. G., Froggatt, M. E., Kreger, S. T., Seeley, R. J., Gifford, D. K., Sang, A. K., and Wolfe, M. S. (2007). High-accuracy fiber-optic shape sensing. In Peters, K. J., editor, *Sensor Systems and Networks: Phenomena, Technology, and Applications for NDE and Health Monitoring 2007*, volume 6530, page 65301S. International Society for Optics and Photonics, SPIE.

Elishakoff, I. (2019). *Handbook on Timoshenko-Ehrenfest Beam and Uflyand-Mindlin Plate Theories*. World Scientific Publishing Company Pte Limited.

Ern, A. and Guermond, J.-L. (2004). *Theory and practice of finite elements*, volume 159. Springer.

Ernst, S., Ouyang, F., Linder, C., Hertting, K., Stahl, F., Chun, J., Hachiya, H., Krumsdorf, U., Antz, M., and Kuck, K.-H. (2004). Modulation of the slow pathway in the presence of a persistent left superior caval vein using the novel magnetic navigation system niobe. *Europace : European pacing, arrhythmias, and cardiac electrophysiology : journal of the working groups on cardiac pacing, arrhythmias, and cardiac cellular electrophysiology of the European Society of Cardiology*, 6:10–4.

Evans, L. C. (1998). *Partial Differential Equations*. Providence: American Mathematical Society.

Farebrother, J., Machado, M. C., and Bowling, M. (2018). Generalization and regularization in dqn. *arXiv preprint arXiv:1810.00123*.

Faure, F., Duriez, C., Delingette, H., Allard, J., Gilles, B., Marchesseau, S., Talbot, H., Courtecuisse, H., Bousquet, G., Peterlik, I., and Cotin, S. (2012). *SOFA: A Multi-Model Framework for Interactive Physical Simulation*, volume 11, pages 283–321. Springer.

Finnesgard, E. J., Simons, J. P., Jones, D. W., Judelson, D. R., Aiello, F. A., Boitano, L. T., Sorensen, C. M., Nguyen, T. T., and Schanzer, A. (2023). Initial single-center experience using fiber optic realshape guidance in complex endovascular aortic repair. *Journal of Vascular Surgery*, 77(4):975–981.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, 11:219–354.

Froggatt, M. E., Klein, J. W., Gifford, D. K., and Kreger, S. T. (2014). Optical position and/or shape sensing. US8773650.

Fu, Q., Li, Z., Ding, Z., Chen, J., Luo, J., Wang, Y., and Lu, Y. (2023). Ed-dqn: An event-driven deep reinforcement learning control method for multi-zone residential buildings. *Building and Environment*, 242:110546.

Fu, Y., Liu, H., Huang, W., Wang, S., and Liang, Z. (2009). Steerable catheters in minimally invasive vascular surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(4):381–391.

Ganji, Y., Janabi-Sharifi, F., et al. (2009). Catheter kinematics for intracardiac navigation. *IEEE Transactions on Biomedical Engineering*, 56(3):621–632.

Gao, Q., Schmidt, S. L., Chowdhury, A., Feng, G., Peters, J. J., Genty, K., Grill, W. M., Turner, D. A., and Pajic, M. (2023). Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, ICCPS '23, page 44–55, New York, NY, USA. Association for Computing Machinery.

Gao, X., Loriot, S., and Tagliasacchi, A. (2024). Triangulated surface mesh skeletonization. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.6.1 edition.

García-Vázquez, V., Von Haxthausen, F., Jäckle, S., Schumann, C., Kuhlemann, I., Bouchagiar, J., Höfer, A.-C., Matysiak, F., Hüttmann, G., Goltz, J. P., et al. (2018). Navigation and visualisation with hololens in endovascular aortic repair. *Innov Surg Sci.*, 3(3):167–177.

Gatti, A. A. and Khallaghi, S. (2022). Pycpd: Pure numpy implementation of the coherent point drift algorithm. *Journal of Open Source Software*, 7(80):4681.

Ghibes, P., Artzner, C., Partovi, S., Hagen, F., Nadalin, S., and Grozinger, G. (2023). Endovascular treatment of symptomatic hepatic venous outflow obstruction post major liver resection. *BMC Gastroenterol*, 23(1).

Golowa, Y. and Cynamon, J. (2012). *Endovascular Treatment of Portal Hypertension*, chapter 85, pages 1095–1106. John Wiley & Sons, Ltd.

Guan, S., Meng, C., Xie, Y., Wang, Q., Sun, K., and Wang, T. (2019). Deformable cardiovascular image registration via multi-channel convolutional neural network. *IEEE Access*, 7:17524–17534.

Ha, X. T., Ourak, M., Al-Ahmad, O., Wu, D., Borghesan, G., Menciassi, A., and Vander Poorten, E. (2021). Robust catheter tracking by fusing electromagnetic tracking, fiber bragg grating and sparse fluoroscopic images. *IEEE Sensors Journal*, 21(20):23422–23434.

Ha, X. T., Wu, D., Ourak, M., Borghesan, G., Dankelman, J., Menciassi, A., and Poorten, E. V. (2023). Shape sensing of flexible robots based on deep learning. *IEEE Transactions on Robotics*, 39(2):1580–1593.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018a). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018b). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

He, Y., Liu, Y., Yang, L., and Qu, X. (2024). Deep adaptive control: Deep reinforcement learning-based adaptive vehicle trajectory control algorithms for different risk levels. *IEEE Transactions on Intelligent Vehicles*, 9(1):1654–1666.

Hjelmstad, K. D. (2007). *Fundamentals of structural mechanics*. Springer Science & Business Media.

Hossain, M., Costanzo, E., Cosentino, J., Patel, C., Qaisar, H., Singh, V., Khan, T., Cheng, J., Asif, A., and Vachharajani, T. (2018). Contrast-induced nephropathy: Pathophysiology, risk factors, and prevention. *Saudi Journal of Kidney Diseases and Transplantation*, 29:1.

Huang, S. K. S. and Miller, J. M. (2019). *Catheter Ablation of Cardiac Arrhythmias*. E-Book. Amsterdam, The Netherlands: Elsevier.

Hwang, J., Kim, J.-y., and Choi, H. (2020). A review of magnetic actuation systems and magnetically actuated guidewire- and catheter-based microrobots for vascular interventions. *Intell. Serv. Robot.*, 13(1):1–14.

Ibrahim, M., Sayed, S., Elasfar, A., Sallam, A., Fadl, M., and Baradai, A. (2012). Coronary fistula between the left anterior descending coronary artery and the pulmonary artery: Two case reports. *Journal of the Saudi Heart Association*, 24:253–6.

Jianu, T., Huang, B., Abdelaziz, M., Vu, M. N., Fichera, S., Lee, C.-Y., Berthet-Rayne, P., y Baena, F. R., and Nguyen, A. (2023). Cathsim: An open-source simulator for endovascular intervention.

Jourdan, F., Alart, P., and Jean, M. (1998). A gauss-seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 155(1):31–47.

Kanagaratnam, P., Koa-Wing, M., Wallace, D., Goldenberg, A., Peters, N., and Davies, D. (2008). Experience of robotic catheter ablation using a novel remotely steerable cahtere sheath. *Journal of interventional cardiac electrophysiology : an international journal of arrhythmias and pacing*, 21:19–26.

Karstensen, L., Behr, T., Pusch, T. P., Mathis-Ullrich, F., and Stallkamp, J. (2020). Autonomous guidewire navigation in a two dimensional vascular phantom. *Current Directions in Biomedical Engineering*, 6(1):20200007.

Karstensen, L., Ritter, J., Hatzl, J., Ernst, F., Langejürgen, J., Uhl, C., and Mathis-Ullrich, F. (2023). Recurrent neural networks for generalization towards the vessel geometry in autonomous endovascular guidewire navigation in the aortic arch. *Int. Journal of Computer Assisted Radiology and Surgery*, 18:1735–1744.

Karstensen, L., Ritter, J., Hatzl, J., Pätz, T., Langejürgen, J., Uhl, C., and Mathis-Ullrich, F. (2022). Learning-based autonomous vascular guidewire navigation without human demonstration in the venous system of a porcine liver. *International Journal of Computer Assisted Radiology and Surgery*, 17.

Kazanzides, P., Fichtinger, G., Hager, G., Okamura, A., Whitcomb, L., and Taylor, R. (2008). Surgical and interventional robotics - core concepts, technology, and design [tutorial]. *IEEE robotics & automation magazine / IEEE Robotics & Automation Society*, 15:122–130.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiran, B., Sobh, I., Talpaert, V., Mannion, P., Sallab, A., Yogamani, S., and Perez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–18.

Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. (2023). A survey of zero-shot generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76:201–264.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Konda, V. and Tsitsiklis, J. (1999). Actor-critic algorithms. In Solla, S., Leen, T., and Müller, K., editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Kori, T., Kori, A., Kori, A., and Nandi, S. (2020). A study on fiber bragg gratings and its recent applications. In Barolli, L., Xhafa, F., and Hussain, O. K., editors, *Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 880–889, Cham. Springer International Publishing.

Korzeniowski, P., White, R., and Bello, F. (2017). Vcsim3: a vr simulator for cardiovascular interventions. *International Journal of Computer Assisted Radiology and Surgery*, 13.

Kuhlemann, I., Kleemann, M., Jauer, P., Schweikard, A., and Ernst, F. (2017). Towards x-ray free endovascular interventions - using hololens for on-line holographic visualization. *Healthcare Technology Letters*, 4.

Kweon, J., Kim, K., Lee, C., Kwon, H., Park, J., Song, K., Kim, Y. I., Park, J., Back, I., Roh, J.-H., Moon, Y., Choi, J., and Kim, Y.-H. (2021). Deep Reinforcement Learning for guidewire navigation in coronary artery phantom. *IEEE Access*, 9:166409–166422.

Lagakos, N., Cole, J. H., and Bucaro, J. A. (1987). Microbend fiber-optic sensor. *Appl. Opt.*, 26(11):2171–2180.

Langer, J. and Singer, D. (1996). Lagrangian aspects of the kirchhoff elastic rod. *SIAM Review*, 38(4):605–618.

Lecomte, F., Scarponi, V., Alvarez, P., Verde, J., Dillenseger, J.-L., Vibert, E., and Cotin, S. (2023). Enhancing fluoroscopy-guided interventions: a neural network to predict vessel deformation without contrast agents. pages 75–76.

Lenoir, J., Cotin, S., Duriez, C., and Neumann, P. (2006). Interactive physically-based simulation of catheter and guidewire. *Computers & Graphics*, 30(3):416–422.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373.

Li, N., Wang, Y., Zhao, H., and Ding, H. (2024a). Robotic systems design in endovascular treatment. *IEEE Transactions on Medical Robotics and Bionics*, 6(2):367–383.

Li, P., Xu, B., Zhang, X., Fang, D., and Zhang, J. (2024b). Design and development of a personalized virtual reality-based training system for vascular intervention surgery. *Computer Methods and Programs in Biomedicine*, 249:108142.

Li, Z., Contini, L., Maria Ippoliti, A., Bastianelli, E., De Marco, F., Dankelman, J., and De Momi, E. (2024c). Deformable model-to-image registration toward augmented reality-guided endovascular interventions. *IEEE Sensors Journal*, 24(13):21750–21761.

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J. E., Jordan, M. I., and Stoica, I. (2018). RLlib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)*.

Liu, H., Farvardin, A., Pedram, S. A., Iordachita, I., Taylor, R. H., and Armand, M. (2015). Large deflection shape sensing of a continuum manipulator for minimally-invasive surgery. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 201–206. IEEE.

Lobaton, E. J., Fu, J., Torres, L. G., and Alterovitz, R. (2013). Continuous shape estimation of continuum robots using x-ray images. In *2013 IEEE international conference on robotics and automation*, pages 725–732. IEEE.

Luboz, V., Zhang, Y., Johnson, S., Song, Y., Kilkenny, C., Hunt, C., Woolnough, H., Guediri, S., Zhai, J., Odetoyinbo, T., Littler, P., Fisher, A., Hughes, C., Chalmers, N., Kessel, D., Clough, P., Ward, J., Phillips, R., How, T., Bulpitt, A., John, N., Bello, F., and Gould, D. (2013). Imagine seldinger: First simulator for seldinger technique and angiography training. *Computer Methods and Programs in Biomedicine*, 111(2):419–434.

Luo, S., Androwis Ph.D., G., Adamovich, S., Nunez, E., Su, H., and Zhou, X. (2023). Robust walking control of a lower limb rehabilitation exoskeleton coupled with a musculoskeletal model via deep reinforcement learning. *Journal of NeuroEngineering and Rehabilitation*, 20.

Mahmud, E., Pourdjabbar, A., Ang, L., Behnamfar, O., Patel, M. P., and Reeves, R. R. (2017). Robotic technology in interventional cardiology: Current status and future perspectives. *Catheterization and Cardiovascular Interventions*, 90(6):956–962.

Manavi Roodsari, S., Freund, S., Angelmahr, M., Seppi, C., Rauter, G., Schade, W., and Cattin, P. C. (2024). Deep learning-based approach for high spatial resolution fibre shape sensing. *Communications Engineering*, 3(1):19.

Megens, M., Leistikow, M. D., van Dusschoten, A., van der Mark, M. B., Horikx, J. J. L., van Putten, E. G., and Hooft, G. W. t. (2021). Shape accuracy of fiber optic sensing for medical devices characterized in bench experiments. *Medical Physics*, 48(7):3936–3947.

Meier, C., Popp, A., and Wall, W. A. (2014). An objective 3d large deformation finite element formulation for geometrically exact curved kirchhoff rods. *Computer Methods in Applied Mechanics and Engineering*, 278:445–478.

Meng, F., Guo, S., Zhou, W., and Chen, Z. (2022). Evaluation of an autonomous navigation method for vascular interventional surgery in virtual environment. In *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1599–1604.

Miranda, V. R. F., Neto, A. A., Freitas, G. M., and Mozelli, L. A. (2024). Generalization in deep reinforcement learning for robotic navigation by reward shaping. *IEEE Transactions on Industrial Electronics*, 71(6):6013–6020.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA. PMLR.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275.

Nakao, M., Nakamura, M., and Matsuda, T. (2022). Image-to-graph convolutional network for 2d/3d deformable model registration of low-contrast organs. *IEEE Transactions on Medical Imaging*, 41(12):3747–3761.

Ogata, K. (2010). *Modern Control Engineering*. Instrumentation and controls series. Prentice Hall.

O'Reilly, O. M. and O'Reilly, O. M. (2017). Kirchhoff's rod theory. *Modeling Nonlinear Problems in the Mechanics of Strings and Rods: The Role of the Balance Laws*, pages 187–268.

Othonos, A. (1997). Fiber bragg gratings. *Review of scientific instruments*, 68(12):4309–4341.

Ourak, M., Buck, S. D., Ha, X. T., Al-Ahmad, O., Bamps, K., Ector, J., and Poorten, E. V. (2021). Fusion of biplane fluoroscopy with fiber bragg grating for 3d catheter shape reconstruction. *IEEE Robotics and Automation Letters*, 6(4):6505–6512.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., and Song, D. (2018). Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*.

Panuccio, G., Torrealba, J., Rohlffs, F., Heidemann, F., Wessels, B., and Kölbel, T. (2023). Fiber optic realshape (fors) technology for endovascular navigation in severe tortuous vessels. *Journal of Endovascular Therapy*, 30(1):29–33. PMID: 35018862.

Park, Y.-L., Elayaperumal, S., Daniel, B., Ryu, S. C., Shin, M., Savall, J., Black, R. J., Moslehi, B., and Cutkosky, M. R. (2010). Real-time estimation of 3-d needle shape and deflection for mri-guided interventions. *IEEE/ASME Transactions On Mechatronics*, 15(6):906–915.

Pointon, J. L., Wen, T., Tugwell-Allsup, J., Létang, J. M., and Vidal, F. P. (2023). gVirtualXray (gVXR): Simulating X-ray radiographs and CT volumes of anthropomorphic phantoms. *Software Impacts*, 16:100513.

Pourdjabbar, A., Ang, L., Reeves, R., Patel, M., and Mahmud, E. (2017). The development of robotic technology in cardiac and vascular interventions. *Rambam Maimonides medical journal*, 8.

Pourkand, A. and Abbott, J. J. (2018). A critical analysis of eight-electromagnet manipulation systems: The role of electromagnet configuration on strength, isotropy, and access. *IEEE Robotics and Automation Letters*, 3(4):2957–2962.

Precup, D., Sutton, R., and Singh, S. (2000). Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*.

Przemieniecki, J. S. (1985). *Theory of matrix structural analysis*. Courier Corporation.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8.

Rafii-Tari, H., Riga, C. V., Payne, C. J., Hamady, M. S., Cheshire, N. J., Bicknell, C. D., and Yang, G.-Z. (2016). Reducing contact forces in the arch and supra-aortic vessels using the magellan robot. *Journal of Vascular Surgery*, 64(5):1422–1432.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.

Reichl, T., Gardiazabal, J., and Navab, N. (2013). Electromagnetic servoing—a new tracking paradigm. *IEEE transactions on medical imaging*, 32(8):1526–1535.

Renda, F., Boyer, F., Dias, J., and Seneviratne, L. (2018). Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533.

Rohan, R., Venkadeshwaran, K., and Ranjan, P. (2024). Recent advancements of fiber bragg grating sensors in biomedical application: a review. *Journal of Optics*, 53:282–293.

Roodsari, S. M., Huck-Horvath, A., Freund, S., Zam, A., Rauter, G., Schade, W., and Cattin, P. C. (2023). Shape sensing of optical fiber bragg gratings based on deep learning. *Machine Learning: Science and Technology*, 4(2):025037.

Rucker, D. C. and Webster III, R. J. (2011). Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Transactions on Robotics*, 27(6):1033–1044.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

Salavitabar, A., Whiteside, W., and Zampi, J. (2022). Feasibility of intraprocedural augmented reality visualisation of 3d rotational angiography in congenital cardiac catheterisation. *Cardiology in the Young*, pages 1–3.

Scarponi, V., Duprez, M., Nageotte, F., and Cotin, S. (2024). A zero-shot reinforcement learning strategy for autonomous guidewire navigation. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–8.

Schegg, P., Dequidt, J., Coevoet, E., Leurent, E., Sabatier, R., Preux, P., and Duriez, C. (2022). Automated planning for robotic guidewire navigation in the coronary arteries. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 239–246.

Schneider, P. (2019). *Endovascular Skills: Guidewire and Catheter Skills for Endovascular Surgery, Fourth Edition*. CRC Press.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, H.-C., Li, Y., Wang, J., Jiang, S., and Zhang, Y. (2023). Real-time liver motion estimation via deep learning-based angle-agnostic x-ray imaging. *Medical Physics*.

Shi, C., Luo, X., Qi, P., Li, T., Song, S., Najdovski, Z., Fukuda, T., and Ren, H. (2017). Shape sensing techniques for continuum robots in minimally invasive surgery: A survey. *IEEE Transactions on Biomedical Engineering*, 64(8):1665–1678.

Silver, D., Lever, G., Heess, N. M. O., Degris, T., Wierstra, D., and Riedmiller, M. A. (2014). Deterministic policy gradient algorithms. In *International Conference on Machine Learning*.

Singh, G. (2022). Chapter 6 - magnetic navigation systems. In Tishin, A. M., editor, *Magnetic Materials and Technologies for Medical Applications*, Woodhead Publishing Series in Electronic and Optical Materials, pages 177–192. Woodhead Publishing.

Singh, S., Ajayi, N., Lazarus, L., and Satyapal, K. S. (2017). Anatomic study of the morphology of the right and left coronary arteries. *Folia Morphologica*, 76(4):668 – 674.

Soller, B. J., Gifford, D. K., Wolfe, M. S., and Froggatt, M. E. (2005). High resolution optical frequency domain reflectometry for characterization of components and assemblies. *Opt. Express*, 13(2):666–674.

Song, H.-S., Yi, B.-J., Won, J. Y., and Woo, J. (2022). Learning-based catheter and guidewire-driven autonomous vascular intervention robotic system for reduced repulsive force. *Journal of Computational Design and Engineering*, 9(5):1549–1564.

Soulami, R. B., Verhoye, J.-P., Duc, H. N., Castro, M., Auffret, V., Anselmi, A., Haigron, P., and Ruggieri, V. G. (2016). Computer-assisted transcatheter heart valve implantation in valve-in-valve procedures. *Innovations (Phila)*, 11(3):193–200.

Sutton, R., Mcallester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inf. Process. Syst*, 12.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2018.

Tagliasacchi, A., Alhashim, I., Olson, M., and Zhang, H. (2012). Mean curvature skeletons. *Comput. Graph. Forum*, 31(5):1735–1744.

Takapoui, R. and Javadi, H. (2016). Preconditioning via diagonal scaling.

Tian, W., Guo, J., Guo, S., and Fu, Q. (2023). A DDPG-based method of autonomous catheter navigation in virtual environment. In *Proc. International Conference on Mechatronics and Automation*, pages 889–893.

Tibebu, A. T., Yu, B., Kassahun, Y., Vander Poorten, E., and Tran, P. T. (2014-10-14). Towards autonomous robotic catheter navigation using reinforcement learning.

Timoshenko, S. and Goodier, J. (1969). *Theory of Elasticity*. Engineering mechanics series. McGraw-Hill.

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al. (2024). Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*.

Trouve, A., Faisal Beg, M., Miller, M. I., and Younes, L. (2005). Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157.

Unberath, M., Zaech, J. N., Lee, S. C., Bier, B., Fotouhi, J., Armand, M., and Navab, N. (2018). DeepDRR – A Catalyst for Machine Learning in Fluoroscopy-Guided Procedures. *Lecture Notes in Computer Science*, 11073 LNCS:98–106.

Vander Poorten, E., Tran, P., Devreker, A., Gruijthuijsen, C., Portoles-Diez, S., Smoljkic, G., Strbac, V., Famaey, N., Reynaerts, D., Vander Sloten, J., and et al. (2016). Cognitive autonomous catheters operating in dynamic environments. *Journal of Medical Robotics Research*, 01(03):1640011.

Vandini, A., Giannarou, S., Lee, S.-L., and Yang, G.-Z. (2013). 3d robotic catheter shape reconstruction and localisation using appearance priors and adaptive c-arm positioning. In *International Workshop on Medical Imaging and Virtual Reality*, pages 172–181. Springer.

Vignon, C., Rabault, J., and Vinuesa, R. (2023). Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions. *Physics of Fluids*, 35(3):031301.

Wagner, M., Schafer, S., Strother, C., and Mistretta, C. (2016). 4d interventional device reconstruction from biplane fluoroscopy. *Medical Physics*, 43(3):1324–1334.

Wang, H. and Wu, J. (2021). A virtual reality based surgical skills training simulator for catheter ablation with real-time and robust interaction. *Virtual Reality and Intelligent Hardware*, 3(4):302–314.

Wang, S., Liu, Z., Shu, X., Cao, Y., Zhang, L., and Xie, L. (2022). Study on autonomous delivery of guidewire based on improved yolov5s on vascular model platform. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–6.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK.

Webster III, R. J. and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683.

Wei, R., Zhou, F., Liu, B., Bai, X., Fu, D., Liang, B., and Wu, Q. (2020). Real-time tumor localization with single x-ray projection at arbitrary gantry angles using a convolutional neural network (cnn). *Physics in Medicine & Biology*, 65(6).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8.

World Health Organization (2021). Cardiovascular diseases (cvds). https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

Yaniv, Z. (2008). Rigid registration. *Image-Guided Interventions: Technology and Applications. New York, NY: Springer*, pages 159–192.

Yeung, A. (2019). The'as low as reasonably achievable'(alara) principle: a brief historical overview and a bibliometric analysis of the most cited publications. *Radioprotection*.

Yoon, J., Chen, R., Refaey, K., Diaz, R., Reimer, R., Komotar, R., Quiñones-Hinojosa, A., Brown, B., and Wharen, R. (2017). Technical feasibility and safety of image-guided parieto-occipital ventricular catheter placement with the assistance of a wearable head-up display. *The international journal of medical robotics + computer assisted surgery : MRCAS*, 13.

Zollei, L., Grimson, E., Norbash, A., and Wells, W. (2001). 2d-3d rigid registration of x-ray fluoroscopy and ct images using mutual information and sparsely sampled histogram estimators. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, volume 2, pages II–II.