



**POLITECNICO**  
**MILANO 1863**

**SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE**

## Autonomous Endovascular Guidewire Navigation in a Simulated Unity Environment

TESI DI LAUREA MAGISTRALE IN  
BIOMEDICAL ENGINEERING - INGEGNERIA BIOMEDICA

**Andrea Montalbano, 247461**

**Advisor:**

Prof. Elena De Momi, Prof.  
Emmanuel Vander Poorten

**Co-advisors:**

Angela Peloso  
Mouloud Ourak

**Academic year:**

2025-2026

**Abstract:** Endovascular procedures represent a widely adopted minimally invasive approach for the treatment of cardiovascular diseases, such as carotid artery stenosis (CAS). These interventions rely on the navigation of flexible instruments through complex vascular anatomies to reach specific target sites. Despite their clinical advantages, they still present several challenges, including risks of complications such as vessel perforation, as well as prolonged exposure of the surgical team to ionizing radiation. Moreover, procedural outcomes strongly depend on the operator's experience. In particular, anatomical variability, such as different aortic arch configurations, significantly affects the difficulty of catheterization and the success rate of the procedure. These limitations have motivated the development of robotic and autonomous solutions aimed at improving precision, reducing operator dependency, and enhancing procedural safety. In this context, this thesis presents the development of a simulation framework for AI-based autonomous endovascular navigation. The system is implemented in Unity and reproduces a realistic geometry of the aortic arch, enabling the simulation of guidewire navigation under different anatomical conditions. The guidewire is modeled as a flexible structure composed of interconnected elements, enabling its interaction with vessel walls. A reinforcement learning approach based on Proximal Policy Optimization (PPO) and Soft-Actor Critic (SAC) is adopted to train an agent to autonomously navigate the guidewire from the descending aorta to target branches of the aortic arch, mimicking clinical scenarios such as carotid artery stenting, which represents a technically demanding step for inexperienced operators. Additionally, the framework includes a dedicated actuator simulation module for analyzing the control system independently of the anatomical environment. The results demonstrate the feasibility of learning effective navigation policies within a simulated environment, highlighting the potential of AI-based approaches for robotic catheterization. This work represents an initial step within a broader research effort carried out at the Robot-Assisted Surgery (RAS) research group at KU Leuven.

**Key-words:** Endovascular navigation, Reinforcement learning, Aortic arch, Simulation

# 1. Introduction

This section introduces the clinical and technological context of the work. In particular, it provides an overview of carotid artery stenosis, its epidemiology, and catheter-based treatment strategies, highlighting the limitations of current approaches and motivating the development of less invasive, more automated catheter-based solutions.

## 1.1. Clinical Background

The head and neck receive the majority of their arterial blood supply from the carotid arteries [1]. The common carotid arteries (CCAs) (Figure 1) constitute the primary vessels delivering blood to the brain and face. The right CCA originates from the brachiocephalic artery in the neck, whereas the left CCA arises from the aortic arch in the thorax. The carotids are critical landmarks in vascular, neurosurgical, and head-and-neck procedures, as inadvertent injury can result in cerebral ischemia or hemorrhage. Furthermore, pathologies concerning the carotid itself like Carotid Stenosis (CS) can reduce the blood flow to the brain, which may lead to an increased risk of cerebrovascular events [2].

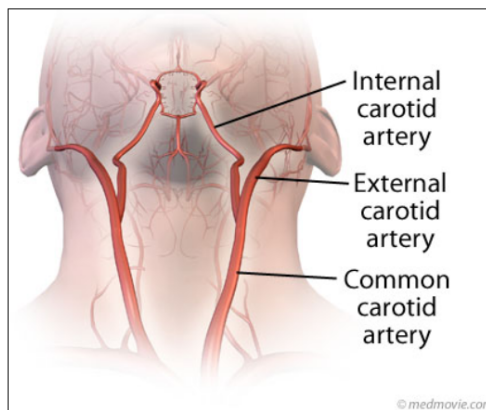


Figure 1: Common, external and internal carotid artery visualization [3].

## Carotid Stenosis

Carotid stenosis is defined as the narrowing or blockage of the carotid arteries caused by plaque buildup [2] (Figure 2). Atherosclerotic plaque formation is a chronic insidious process with a number of potential contributors to the formation of such plaques. The definition of atherosclerosis is not only limited to abnormal deposition of lipid but also includes a chronic, complex, inflammatory process. Alteration of shear stress is an important initiating factor in atherosclerosis. The process is initiated at the abluminal surface, where metabolic, mechanical, or physical injury causes disruption of endothelial integrity. The resultant endothelial dysfunction can lead to intimal stress associated with an abnormal biochemical signal. Over time, disruption of the tunica media also occurs [4]. Despite CS can generally affect every part of the carotids, in most cases it associated with Internal Carotid Artery (ICA) stenosis. Early detection of stenosis of carotid artery is crucial to prevent potential stroke linked to ICA stenosis [4].

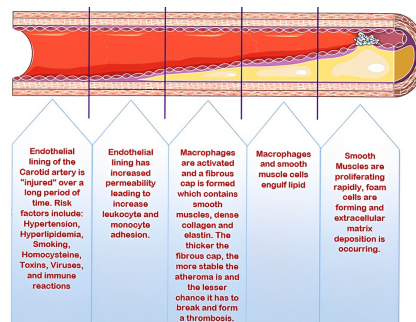


Figure 2: Pathogenesis of carotid artery stenosis [2].

The global prevalence of significant carotid artery stenosis (defined as  $\geq 50\%$  narrowing) is estimated at around

1.8% in men and 1.2% in women and increases with age [5]. The annual occurrence rate ranges from 2% to 6% [6]. The cumulative risk of stroke following the onset of carotid stenosis may be as high as 15% in the first year, 26% at 2 years, and 30% at 5 years [7]. Moreover, it is responsible for approximately 7% of all cases of ischemic strokes [8]. Considering the delicate position and vital functions of the carotid arteries, interventions are always challenging and technically demanding.

## Carotid Stenosis Treatment

Nowadays carotid revascularization procedures comprise Carotid endarterectomy (CEA) and Carotid artery stenting (CAS) (Figure 3).

CEA firstly involves opening the carotid artery, removing the atherosclerotic plaque, and carefully closing the artery. Monitoring of cerebral perfusion during the procedure is critical, and in some cases, shunts may be used to maintain blood flow to the brain during the surgery [9]. CEA also shows some contraindications in which it would be better to not pursue surgery treatment:

- Severe comorbid conditions with limited life expectancy,
- Complete occlusion of the ipsilateral ICA,
- Uncorrectable coagulopathy.

In patients deemed high risk for surgical intervention due to unfavorable anatomy such as high carotid bifurcation, prior neck irradiation, or significant comorbidities, CAS may offer a safer and equally effective alternative.

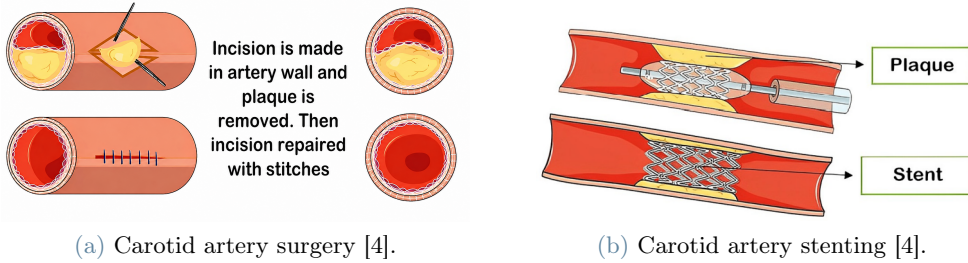


Figure 3: Carotid stenosis treatment procedures.

CAS is a minimally invasive endovascular procedure designed to re-establish cerebral perfusion by performing balloon angioplasty and placing a stent to maintain vessel patency. The procedure can be performed via a transfemoral or transcarotid approach, with selection guided by patient anatomy, comorbidities, and risk profile. In particular, during the transfemoral approach, under ultrasound guidance, percutaneous access to the common femoral artery is obtained using a micropuncture kit and the Seldinger technique. A femoral angiogram should be performed to confirm appropriate entry into the common femoral artery and assess suitability for a vascular closure device at the procedure's conclusion. Using fluoroscopy, the wire is navigated carefully into the aortic arch. With an angled catheter in a left anterior oblique projection, the CCA is selectively cannulated, and the anatomy is mapped. The guidewire is advanced into the ECA, followed by advancement of the catheter itself. Over a stiffer wire, a shuttle sheath or guide catheter is advanced into the CCA for stability. Predilation with a low-profile angioplasty balloon is performed cautiously, with close attention to hemodynamic changes. After predilation, an appropriately sized self-expanding stent is deployed across the lesion. Dedicated carotid arteriography is performed in multiple projections to identify the stenotic lesion and assess collateral circulation. Upon completion, carotid and cerebral angiography confirm stent patency, adequate flow restoration, and absence of distal emboli. All devices are removed, and the arteriotomy site in the femoral artery is closed using a vascular closure device [10]. The transfemoral approach remains the most commonly used method for CAS, offering excellent maneuverability and familiarity for most interventionists. However, it can be limited by the tortuosity of the aortic arch, particularly in the presence of a type-3 aortic arch.

## Aortic Arch Configurations

The aortic arch is the section of the aorta between the ascending and descending aorta. The aortic arch has 3 major branches. The brachiocephalic trunk is the first branch of the aortic arch and supplies blood to the right arm, right head, and neck. The left common carotid artery is the second branch of the aortic arch, which supplies blood to the left side of the head of the neck. The last branch of the aortic arch is the left subclavian artery that distributes blood to the left arm [11]. Based on the geometric configuration of the aortic arch and its branches, the aortic arch can be classified into type-1/type-2/type-3 aortic arch (Figure 4). The classification is done according to the Madwhal criteria [12], the most relevant one in clinical field.

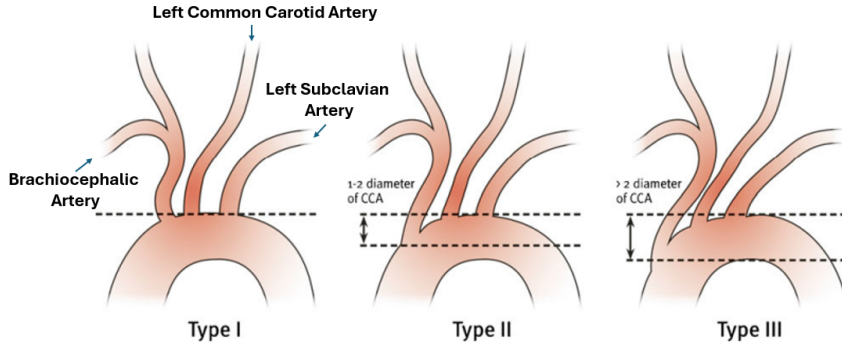


Figure 4: Aorch types based on Madwhal classification [13].

More specifically, the criteria is based on the distance between the higher point of the aortic arch and the starting point of the brachiocephalic trunk. The range of the criteria is described in Table 1. The variability

| Criteria             | Description                                                                                                                                                                                      |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Type I arch</b>   | The brachiocephalic trunk originates within a distance of $< 1$ diameter of the left common carotid artery (LCCA) from the horizontal plane of the outer arch curvature (i.e., top of the arch). |
| <b>Type II arch</b>  | The brachiocephalic trunk originates at a distance within 1 and 2 LCCA diameters from the horizontal plane of the outer arch curvature (i.e., top of the arch).                                  |
| <b>Type III arch</b> | The brachiocephalic trunk originates at a distance $> 2$ LCCA diameters from the horizontal plane of the outer arch curvature (i.e., top of the arch).                                           |

Table 1: Anatomic criteria and methods for Aortic Arch Classification (modified from [12]) [14].

in aortic arch morphology significantly influences catheter stability, device selection, and technical success rates during carotid artery stenting. Consequently, the ability to safely and efficiently navigate complex vascular geometries has become a central challenge in modern endovascular procedures.

## 1.2. Endovascular Navigation

Endovascular navigation refers to the process by which flexible instruments, such as catheters and guidewires, are controlled and advanced through the vascular network in order to reach specific anatomical targets for diagnostic or therapeutic purposes. This process takes place in environments characterized by complex geometries, limited visibility, and continuous mechanical interaction with biological tissues, making navigation a critical component of minimally invasive cardiovascular procedures. To successfully undertake endovascular navigation procedures both pre-operative and intra-operative phases are critical (Figure 5).

### Pre-operative and Intra-operative Phase

During the pre-operative phase, patient-specific three-dimensional imaging modalities, most commonly computed tomography angiography (CTA), are acquired to obtain a detailed representation of the vascular anatomy. These images enable the visualization of vessel geometry, branching patterns, diameters, and pathological regions of interest. Based on pre-operative images, it is possible to reconstruct the 3D anatomic model, which is then used for target definition and path planning. Importantly, the pre-operative anatomical model represents a static approximation of the patient’s vasculature and does not account for intraoperative factors such as vessel deformation, cardiac motion, or tool–tissue interactions. As a result, while pre-operative imaging provides comprehensive spatial information, it cannot be directly used for real-time guidance without integration with intraoperative sensing modalities. The intra-operative phase, instead, encompasses the execution of the endovascular procedure and the real-time navigation of interventional instruments within the patient’s vasculature. While this process is conventionally performed manually by skilled clinicians, recent advancements are driving the development of robotic systems for endovascular navigation, with the goal of improving procedural accuracy and safety.

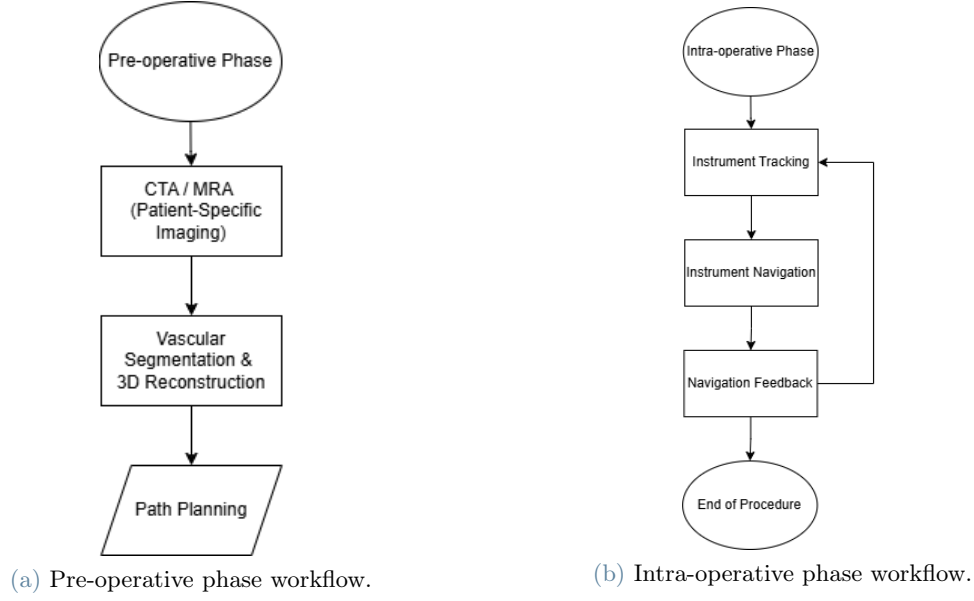


Figure 5: Pre-operative and intra-operative workflow of endovascular procedures.

## Robotic Systems for Endovascular Navigation

Robotically controlled catheter systems (Table 2) have been developed in order to mitigate limitations of manual navigation such as radiation exposure, precision, tremors, fatigue and added operator discomfort [15]. They are mainly classified as electromechanical or magnetic depending on the actuation technology. Electromechanically actuated commercial platforms include the Sensei and Magellan robotic catheter systems (Auris Surgical Robotics), Amigo remote catheter system (Catheter Precision Inc), and CorPath vascular robotic system (Corindus Vascular Robotics). The Niobe magnetic navigation system (Stereotaxis USA) is a commercially available magnetically controlled robotic system.

| System   | Access           | Control | Haptic | Motion | Off-shelf | Use                  | Level   |
|----------|------------------|---------|--------|--------|-----------|----------------------|---------|
| Sensei   | Robotic catheter | Remote  | No     | 3-DOF  | No        | Cardiac, EVAR        | In vivo |
| Magellan | Robotic catheter | Remote  | No     | 7-DOF  | No        | Peripheral           | In vivo |
| Amigo    | Robotic catheter | Remote  | No     | 3-DOF  | No        | Cardiac (RFA)        | In vivo |
| CorPath  | Bedside unit     | Remote  | No     | 3-DOF  | Yes       | Coronary             | In vivo |
| Niobe    | Magnetic system  | Remote  | No     | 3-DOF  | No        | Coronary, peripheral | In vivo |

Table 2: Overview of robotic systems for endovascular navigation. Modified from [16].

Abbreviations: DOF = degree of freedom, EVAR = endovascular aneurysm repair, RFA = radiofrequency ablation.

### 1.3. Clinical Problem

Even if endovascular catheter-based procedures are becoming an established treatment for cardiovascular diseases, they show some limitations and challenges to overcome [17]. They are related to occasional complications including perforations, thrombosis, and dissection in parent artery, as well as distal embolization of thrombus [18]. Moreover, angiography requires intravascular contrast agent administration, which can sometimes lead to nephrotoxicity [19]. Lastly, surgeon and his team are exposed to a high cumulative dose of X-ray radiations, which is a risk factor for cancer. Even if radiation risks are already minimized by the use of protective equipment, the heavy equipment itself can lead to orthopedic complications, it suggesting the needs for an alternative solution [20]. As mentioned before, robotic systems can mitigate some of the current challenges but, still, they have limitations. The outcome of the procedure is still limited by human error and operator skill set [21]. Furthermore, these robotic systems also consist of user interfaces such as buttons and joysticks,

requiring skills that are different to those used in current clinical practice. In addition, coordinated control in terms of simultaneous pushing/pulling and rotation of the instrument is technically challenging, especially within complex vascular anatomies, such as in the presence of a type-3 aortic arch. In this context, the increasing procedural complexity and the need for enhanced precision and safety have motivated the development of autonomous navigation strategies for endovascular interventions. Autonomous control aims to reduce operator dependency, minimize human error, improve reproducibility, and potentially decrease radiation exposure and procedure time. Traditionally, autonomous or semi-autonomous navigation has been addressed through classical control strategies based on kinematic modeling [22], rule-based decision systems, image-based feedback control [23], and optimization-driven path planning [24]. These approaches rely on predefined models of catheter–tissue interaction, vessel geometry reconstruction from imaging data, and deterministic control laws to guide instrument motion. While such methods can be effective in structured or well-modeled environments, their performance may degrade in the presence of anatomical variability, nonlinear mechanical interactions, and uncertain or partially observable conditions. To overcome these limitations, artificial intelligence, particularly machine learning and reinforcement learning techniques, has recently emerged as a promising framework for autonomous endovascular navigation [17]. By learning control policies directly from data or interaction with the environment, AI-based methods can potentially adapt to complex, nonlinear dynamics and patient-specific anatomical variations. AI and machine learning are indeed quickly developing in the medical field for disease prediction and diagnosis, providing support and helping the medical team to intervene faster and with more accuracy.

#### 1.4. State Of The Art of AI Based Autonomous Navigation

Autonomous navigation is nowadays experimentally tested for different purpose in vitro, in silico and in ex-vivo set-up [17]. The goal of this technique mainly focus on fully autonomous navigation of the instruments, trajectory optimization and shared navigation. It is based on machine learning models trained on dataset composed either or both of anatomic images and position data. Machine learning methods can be divided in three main categories: supervised machine learning, unsupervised machine learning and reinforcement learning. Furthermore, it is possible to identify Learn from Demonstrations algorithms, which solve non deterministic challenges with supervised approach.

Supervised machine learning involves training an algorithms based on a labeled dataset (the output is known) in order to predict the label of new instances (prediction tasks) [25]. Learn from Demonstration is a variant of supervised learning, where input data are given by an expert demonstrator. It can also be used as precursor of reinforcement learning to help the algorithm to learn faster by imitating the expert behavior. Unsupervised machine learning involves training algorithms on unlabeled datasets, with the goal of discovering hidden structures or patterns in the data. Finally, in reinforcement learning (RL), an agent interacts with an environment by observing its state and performing actions. Each action influences the environment, which returns feedback in the form of a reward signal. The objective of the agent is to learn an optimal policy that maximizes the expected cumulative reward over time by mapping states to actions.

Twelve studies were selected to review the state of the art in AI-based autonomous endovascular navigation. Studies were included if they presented at least one artificial intelligence method applied to autonomous navigation tasks. To ensure a broad representation of the current state of the art, papers covering different AI approaches were considered. Publications not written in English were excluded. Among the selected ones, nine works adopted reinforcement learning approaches, two employed supervised learning methods, and one used an unsupervised learning strategy. The reinforcement learning methods included A3C, DDPG, DQN, Dueling DQN, HER,  $PI^2$ , PPO, GAIL, and Rainbow [26–34]. In some cases, reinforcement learning was combined with Learning from Demonstration (LfD) techniques, such as Behavioral Cloning, GAIL, or human demonstrations, to improve the learning process [29, 31, 33]. Supervised approaches included CNN and YOLOv5s-based methods [35, 36], while the unsupervised approach was based on Gaussian Mixture Models combined with LfD [37].

The reviewed studies were performed either in simulation or in controlled physical setups. In silico environments were used in two works [26, 34], while the majority of studies relied on in vitro platforms, often based on transparent or simplified vascular phantoms. Simulation environments vary across studies. The SOFA framework is used in some works [34], while Unity-based simulations are adopted in others [26]. The considered tasks ranged from guidewire or catheter navigation in idealized vessel models to more anatomically specific scenarios, including coronary navigation, porcine liver vessels, right atrium navigation, and aortic arch catheterization. A summary of the reviewed studies is reported in Table 3.

The performance metrics adopted across the literature are heterogeneous, making direct comparison between studies difficult. The most common evaluation criteria are success rate (5/12) and procedure time (5/12), while other works report contact force (3/12), speed (3/12), acceleration, path length (2/12), or image-based detection metrics. This variability reflects the absence of standardized benchmarks for AI-based autonomous endovascular navigation and highlights the early-stage nature of the field.

Among the reviewed works, only a limited subset specifically addresses the aortic arch [26, 29, 31, 34, 37]. These studies focus on tasks such as guidewire over-arching, catheter and guidewire navigation, trajectory optimization, and selective catheterization of supra-aortic branches. The aortic arch represents a particularly challenging anatomical region due to its curvature, branching structure, and high inter-patient variability. These aspects are especially relevant for procedures such as carotid artery stenting, where the instruments must be navigated from the descending aorta toward the supra-aortic vessels.

| Reference                                | Method                | Environment | Task                                             | Evaluation (results)                                                                       |
|------------------------------------------|-----------------------|-------------|--------------------------------------------------|--------------------------------------------------------------------------------------------|
| <b>Reinforcement Learning Approaches</b> |                       |             |                                                  |                                                                                            |
| Meng 2021 [26]                           | A3C                   | In silico   | Guidewire over-arching (Unity)                   | n/a                                                                                        |
| Meng 2022 [34]                           | A3C                   | In silico   | Guidewire and catheter over-arching (SOFA)       | Insertion time = 68.81 s; Contact force = 22.15 mN                                         |
| Behr 2019 [27]                           | DDPG + HER + HD       | In vitro    | Catheter navigation (idealized vessel platform)  | Success rate = 89.6%                                                                       |
| You 2019 [28]                            | Dueling DQN           | In vitro    | Catheter navigation (right atrium)               | Distance = $4.70 \pm 1.59$ mm; Success rate = 87%                                          |
| Chi 2018 [29]                            | PI <sup>2</sup> + LfD | In vitro    | Trajectory optimization (aortic arch)            | RMSE = 3.65 mm; Mean force = 0.0609 N; Speed std = 2.22; Max acc = 315.76; Path = 35.06 mm |
| Kweon 2021 [30]                          | Rainbow               | In vitro    | Guidewire navigation (LAD)                       | Success rate < 95%; Duration = $21.21 \pm 11.47$ s                                         |
| Chi 2020 [31]                            | PPO + GAIL            | In vitro    | Selective catheterization (BCA, LCCA)            | Mean force = 0.34 N; Max force = 0.76 N; Speed = 1.03; Path = 55.7 mm; Duration = 52.1 s   |
| Karstensen 2022 [32]                     | DDPG + HER            | In vitro    | Catheter navigation (porcine liver)              | Success = 100% (sim), 30% (bench)                                                          |
| Cho 2022 [33]                            | DDPG + BC             | In vitro    | Guidewire navigation (idealized vessel platform) | Average time = 35.06 s                                                                     |
| <b>Supervised Learning Approaches</b>    |                       |             |                                                  |                                                                                            |
| Zhao 2019 [35]                           | CNN                   | In vitro    | Guidewire navigation (designed vessels)          | Success = 95%; Time = 31.7 s                                                               |
| Wang 2022 [36]                           | YOLOv5s               | In vitro    | Guidewire delivery (coronary)                    | AP = 81.7% (guidewire), 98.4% (heart); mAP = 90.0%                                         |
| <b>Unsupervised Learning Approaches</b>  |                       |             |                                                  |                                                                                            |
| Rafi 2013 [37]                           | GMM + LfD             | In vitro    | Robot-assisted catheterization (aortic arch)     | Mean speed = 4.00; Max speed = 54.5; Mean acc = 73.1; Max acc = $1.4 \times 10^3$          |

**Table 3:** Overview of representative studies on autonomous catheter and guidewire navigation, grouped by learning paradigm. Rows highlighted in light blue correspond to studies specifically addressing the aortic arch.

*Abbreviations:* RMSE = root mean square error; AP = average precision; mAP = mean average precision; BCA = brachiocephalic artery; LCCA = left common carotid artery.

Despite these contributions, several limitations remain: 1) to the best of the author’s knowledge, none of the reviewed studies explicitly trains and evaluates autonomous navigation models on Type-3 aortic arch configurations, which are associated with increased anatomical complexity and procedural difficulty; 2) generalization across different aortic arch geometries remains largely unexplored. Only one reviewed study [31] considered training and validation across different aortic arch types; 3) although LfD techniques have been combined with RL, existing works typically employ a single demonstration-based strategy. The combined use of multiple LfD approaches, such as Behavioral Cloning and GAIL within the same framework, remains underexplored. The state of the art of AI-based autonomous navigation gives evidence that in the future it may be possible to navigate catheters and wires through a patient’s vasculature to the target site, requiring minimal assistance from an operator. If proven to be safe, there would be several benefits. It is plausible that in clinical specialties facing a shortage of highly-trained operators, there may be a reduced need for their expertise, potentially leading to greater accessibility of endovascular treatments globally [17]. Furthermore, time procedures would probably decrease and accuracy would increase.

## 1.5. Objective

The primary objective of this thesis is to investigate AI-driven approaches for autonomous endovascular navigation across different anatomical scenarios through the development of a Unity-based simulation framework. In particular, the work aims to address several gaps identified in the literature by:

1. extending the evaluation to more complex anatomical configurations, including Type-3 aortic arches;
2. investigating the generalization capabilities of the proposed approach across different anatomical scenarios within a unified simulation framework;

3. exploring the combined use of multiple Learning from Demonstration (LfD) techniques together with reinforcement learning to improve navigation performance.

The thesis also focuses on the design of the actuation behaviour of the robotic catheterization platform available at the Robot-Assisted Surgery Research Group, Department of Mechanical Engineering, KU Leuven, thereby establishing a foundation for the future transfer and validation of learning-based navigation strategies on the laboratory's robotic catheterization system.

## 2. Material & Methods

This section presents the materials and methods adopted to design and implement the proposed system. The chapter details the actuation logic, the simulation environment, and the AI-based solutions used throughout this work.

### 2.1. Control Strategy for Coordinated Actuation of the Catheterization System

The robotic system currently available in the laboratory controls three endovascular instruments: the guidewire, the catheter, and the introducer sheath. The guidewire is the thinnest instrument and it is primarily used to explore the vascular system, providing a guiding path for the catheter to follow. The catheter advances coaxially over the guidewire, enabling controlled navigation toward the target region while improving stability and support. The introducer sheath represents the outermost component of the system, ensuring vascular access and providing mechanical support during the navigation procedure. Each instrument is actuated by a pair of units, as shown in Figure 6 (guidewire by Active Unit C and D, catheter by Active Unit Y and Passive Unit B, introducer sheath by Active unit A and Passive Unit X). The active units are a catheterization system based on braided sleeve grippers. They allow both linear movement of the instrument in longitudinal axis direction and rotation of the instrument around its longitudinal axis direction (2 Degree Of Freedom (DOF)). The novel of this system is the innovative gripping system (Figure 7). The mechanism relies on the use of sleeve surrounding the instrument, which can be extended or compressed to grip or release the instrument. The sleeve is hold at the two extremities by two sleeve holders. Both sleeve holders' positions are fixed through the use of two magnets. One sleeve holder is then connected to a pneumatic cylinder which can move the sleeve holder back and forward. If the distance between the two sleeve holders increases the sleeve's diameter decreases and the instrument is gripped [38]. The passive units, instead, remain permanently gripped on the instrument and allow only translational movement (1 DOF).

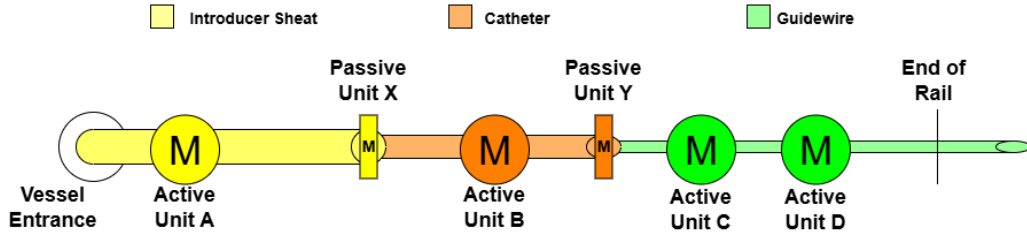


Figure 6: Laboratory robotic system simplified schematic.

The aforementioned catheter driver technology shows the following advantages:

- The gripping force is distributed over the instrument for a length equal to the sleeve length;
- It is possible to control the gripping force regulating the distance between the sleeve holders;
- The grippers are modular, since the magnets are strong enough to lock the detachable sleeve holders in place while being weak enough for a user to pull them out.

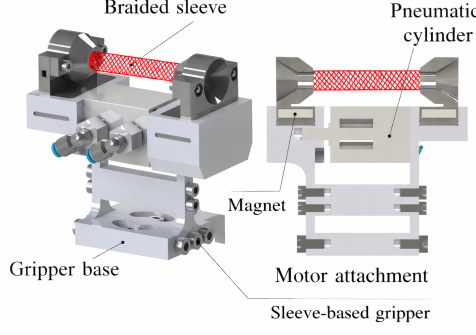


Figure 7: Sleeve-based gripping system [38].

A high-level control architecture has been developed as a preliminary step toward the integration of the AI navigation framework with the robotic platform. This modular design will support future simulation studies under conditions that closely resemble the real experimental setup, easing the transition from algorithm development to robotic validation.

## Actuators Movement Constraints

In order to implement an efficient and coordinated high-level catheterization logic, a set of system specifications and operational constraints must be respected. The specifications define the operational capabilities of the robotic platform, whereas the constraints represent conditions that must never be violated during execution.

### Specifications

1. Units can translate and rotate simultaneously.
2. Rotation can be performed both clockwise and counterclockwise, with a range of  $0^\circ$ – $360^\circ$ .
3. Displacements and rotations are defined independently for each instrument.
4. Passive units always grip the instrument.
5. Since passive units always grip the instrument, any displacement of a passive unit directly results in the same displacement of the corresponding instrument.

### Constraints

1. At least one unit must always maintain the grip on each instrument.
2. A minimum safety distance between units must always be preserved.
3. To avoid instrument bending outside the vessel (kinking phenomenon), a minimum distance before pushing must be ensured between:
  - (a) the entrance of the phantom and the closest ON-grip unit linked to the introducer sheath;
  - (b) the passive unit linked to the introducer sheath and the closest ON-grip unit linked to the catheter;
  - (c) the passive unit linked to the catheter and the closest ON-grip unit linked to the guidewire.
4. The maximum translational motion of each instrument depends on the exposed length outside the phantom.
5. The position of the end of the guidewire must not exceed the position of the last unit linked to the guidewire.

For the sake of clarity, the overall control logic is divided into three phases: Calibration Phase, Linear Translation Phase, and Rotation Phase. Although translation and rotation are described separately, they can be executed simultaneously during operation. The schematic shown in Figure 6 is used as a reference for the following block diagrams.

## Calibration Phase

In the calibration phase the parameters (desired displacement, rotation and anti-kinking distance set by the user) of the simulation are processed for each instrument and the units are moved in order to be ready for the next phase. First, based on the maximum available displacement of the instrument (length of the instrument outside the vessel), the goal displacement is computed (Constraint 4). Secondly, during the calibration phase for pushing (Figure 8a), if necessary, every active unit is moved toward the vessel entrance until the instrument anti-kinking distance is respected (Constraint 3). During the calibration phase for pulling (Figure 8b), instead, the active units are moved toward the end of rail to respect the same constraint. Unlike the other instruments, which are manipulated by one active unit and one passive unit that continuously maintains the grip, the guidewire is driven by two active units. Therefore, at least one unit must always remain in the gripping state to do not violate Constraint 1.

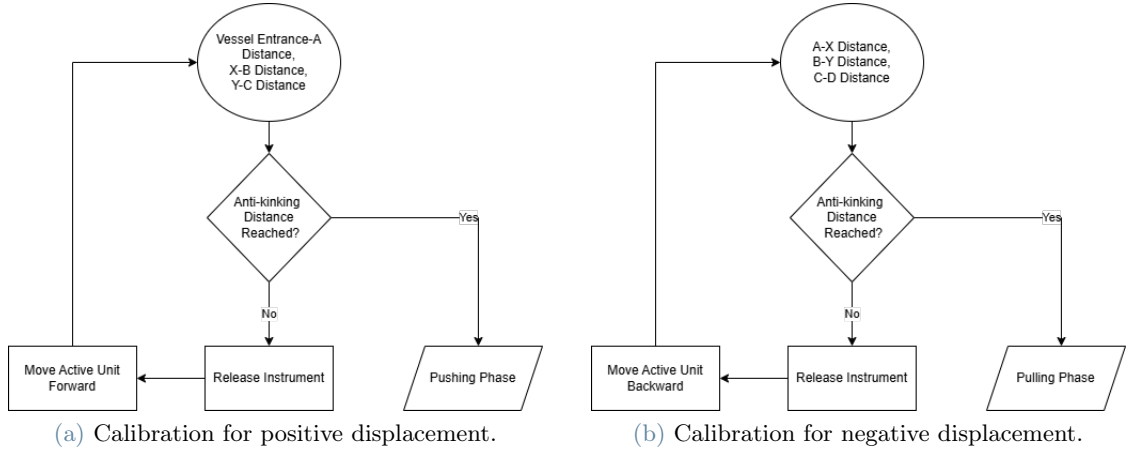


Figure 8: Calibration phase blocks diagram.

## Linear Translation Phase

The linear translation phase defines the motion of the system along the longitudinal axis of the instrument, based on the sign of the desired displacement. A positive displacement corresponds to a pushing action, while a negative displacement results in a pulling motion. During this phase, the actuation units grip the instrument and translate it until the target displacement is reached. If the desired displacement has not yet been achieved and the units approach a potential collision, the instrument is released and the units are repositioned to a safe distance (Constraint 2), provided that the anti-kinking constraint (Constraint 3) is still satisfied. Since all the units can move simultaneously, it is important to always check the relative distance between the actuators even during the repositioning part. This allows the system to re-engage and continue the motion iteratively until the required displacement is completed. The overall workflow of the linear translation phase is illustrated in Figure 9.

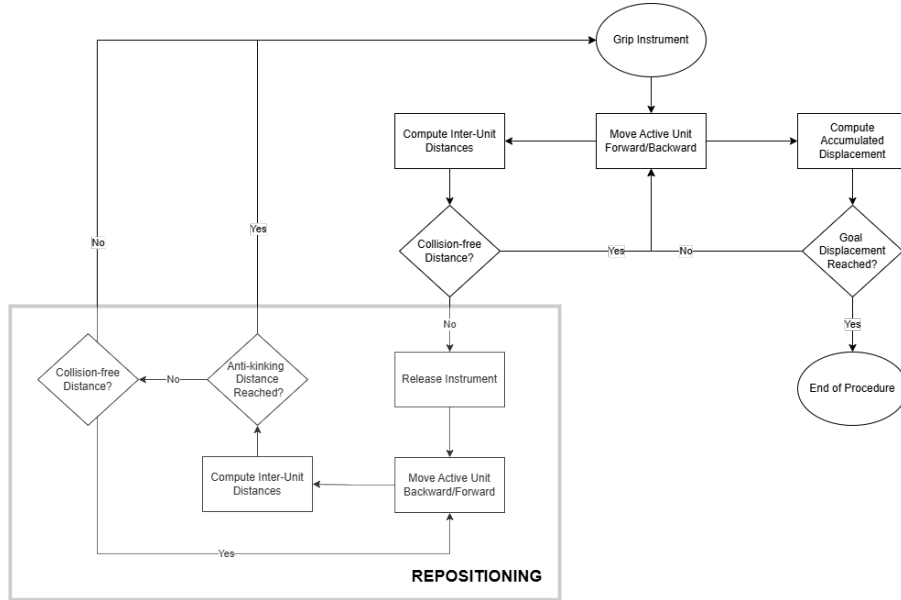


Figure 9: Linear Translation phase blocks diagram.

When the distal end of the guidewire approaches Active Unit D (Figure 10), the standard advancement procedure is temporarily interrupted and a dedicated repositioning strategy is executed to manage Constraint 5. Active Unit D initially maintains the grip on the guidewire while Active Unit C is translated forward until the minimum safe distance from Passive Unit Y is reached (Constraint 2). Active Unit C then establishes the grip on the guidewire, allowing Active Unit D to move proximally until the anti-kinking distance is restored (Constraint 3). Once the nominal configuration has been re-established, the standard advancement workflow resumes, with Active Unit D pushing the guidewire forward. The procedure continues until either the desired displacement is achieved or the maximum available displacement is reached.

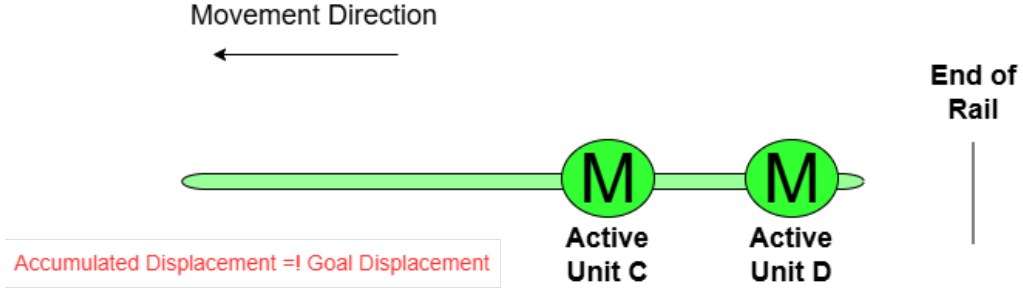


Figure 10: Special case of the general Linear Translation workflow.

## Rotation Phase

Finally, the rotation phase defines the phase in which the instrument get rotated by the active units until the desired rotation is reached. Even if it is described separately from the translation phase, it happens simultaneously. The workflow of the rotation phase is shown in Figure 11.

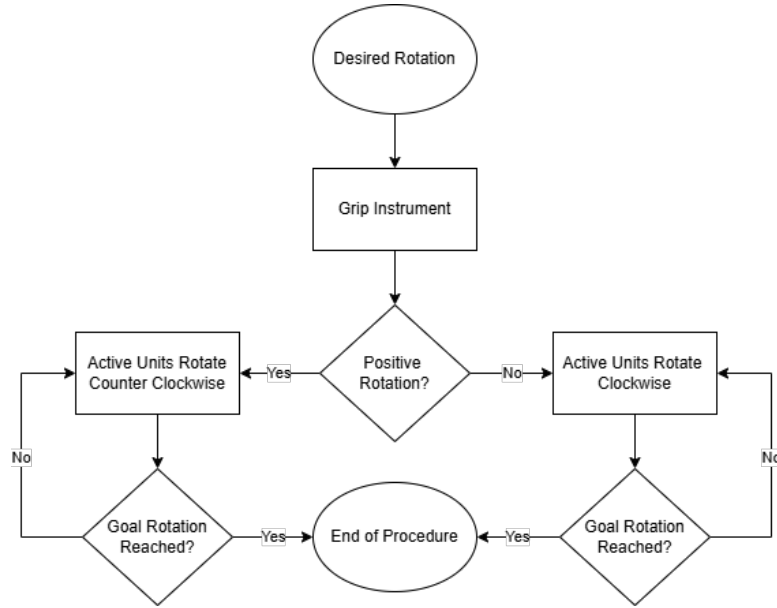


Figure 11: Rotation phase blocks diagram.

## 2.2. Unity Simulation Environment

The simulation environment was developed using Unity Engine, version *6000.0.40f1*, a real-time 3D development platform widely adopted for interactive simulations and robotics research. The software has been chosen taking in consideration that the following aspects would coexist and work simultaneously:

- **3D Environment Design**
- **Physical Interaction and Collision Handling**
- **Flexible Object Modeling**
- **Graphical User Interface**
- **Python Integration**
- **Machine Learning Integration**

Based on these six metrics Unity Engine has been chosen as the best trade-off of metrics implementation capability and simplicity of use. Furthermore, the choice of a simulation-based approach in this work is not incidental but motivated by practical and methodological considerations. Reproducing and iterating on a physical laboratory setup for catheter manipulation can be time-consuming, costly, and constrained by hardware availability and safety requirements. In addition, the large amount of labeled data required for training and validation is difficult to obtain in real-world set-ups. Simulation enables rapid prototyping and systematic exploration of control strategies by allowing key parameters, environmental conditions, and system configurations to be modified efficiently and reproducibly. Moreover, a simulated environment facilitates the exploration of a broader design and control space within a limited time frame, which would be impractical to achieve using

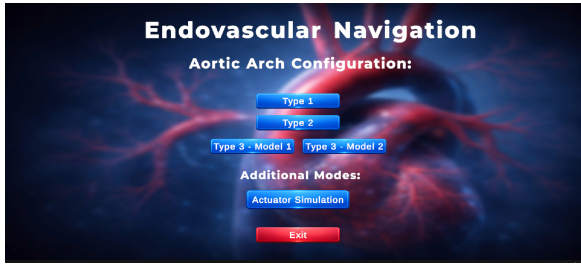
physical hardware alone. In addition, simulation provides a suitable foundation for training learning-based control strategies, whose policies can subsequently be transferred to real-world systems following a sim-to-real paradigm.

## Graphical User Interface

The graphical user interface (GUI) has been designed to provide a clear and intuitive interaction with the simulation environment, allowing the user to easily configure and launch different scenarios.

The entry point of the application is represented by an initial menu (Figure 12a), where the user can select the desired anatomical configuration of the aortic arch. The available options include Type 1, Type 2, and two variants of Type 3 models. In addition to the main anatomical scenarios, the interface also includes access to an *Additional Mode*, the *Actuator Simulation*, which allows to simulate the movement's logic of the six actuators controlling the three instruments. An exit button is also provided to allow the user to close the application at any time.

Once a specific aortic arch type is selected, a dedicated interface is displayed (Figure 12b), where the user is required to choose the control modality. Two options are available: *Manual Control* and *Autonomous Control*. This step clearly separates the definition of the anatomical environment from the control strategy to be adopted during the simulation.



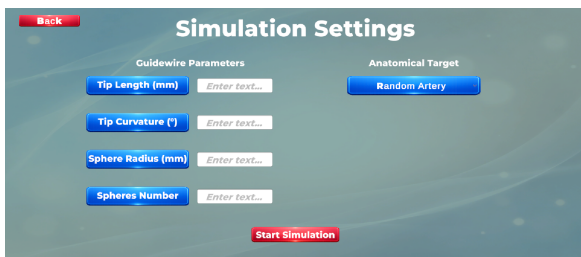
(a) GUI initial menu.



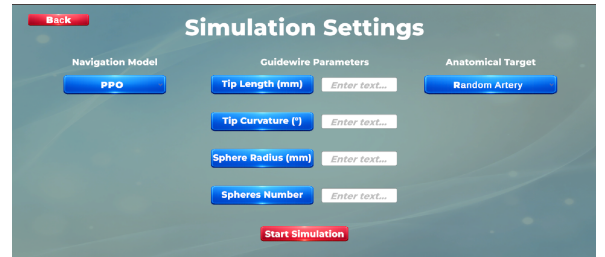
(b) Manual/Autonomous control selection interface.

Figure 12: Graphic User Interface (menu and selection).

When the manual control mode is selected, the interface switches to the simulation settings page (Figure 13a). In this configuration, the user is required to define the main geometric parameters of the guidewire, including tip length, tip curvature, sphere radius, and the number of spheres used to model the distal portion of the device. Additionally, the anatomical target can be selected through a dedicated drop-down menu. The autonomous control mode interface (Figure 13b), instead, in addition to the guidewire parameters and target selection described in the manual case, includes a field for selecting the AI based navigation model. In both cases, once the button *Start* is pressed, the simulation is launched.



(a) Manual navigation parameters settings.



(b) Autonomous navigation parameters settings.

Figure 13: Graphic User Interface (parameters settings).

Finally, the *Additional Mode* of the initial menu allows the user to test the high level control logic of the catheterization system previously discussed.

Once this mode is selected, the user is presented with a dedicated interface that allows the definition of control parameters for the introducer sheath, catheter, and guidewire. For each instrument, the user can specify the anti-kinking distance, desired rotation, and desired displacement. These parameters enable direct control of the actuation system and allow the user to test different motion strategies. The interface also provides two different visualization modes that can be dynamically selected through dedicated buttons. The *Top View* (Figure 14a)

offers a global perspective of the actuator system, allowing the user to observe the relative positioning and coordination of all units along the rail. In contrast, the *Guidewire View* (Figure 14b) focuses on the distal portion of the system, providing a more detailed visualization of the guidewire behavior during actuation.

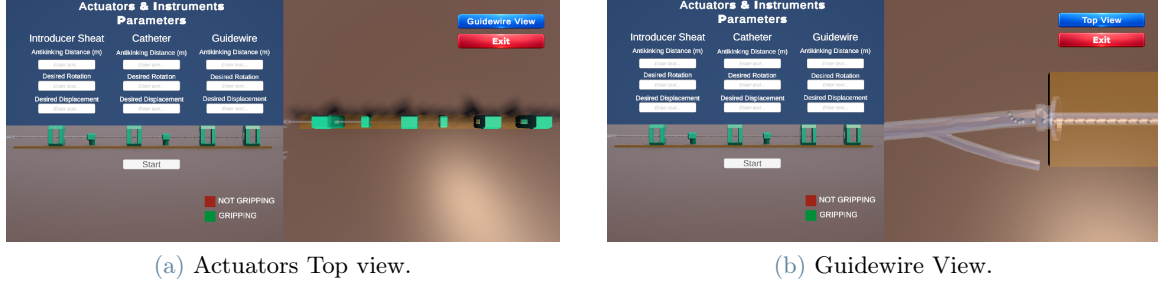


Figure 14: Graphical User Interface (actuator simulation).

### Python-Unity Communication

The units movement logic is implemented in Python, while the visualization is handled by Unity programming language C#. Real time Python-C# communication is then needed to exchange data and run the simulation properly. In this work, it is possible thanks to TCP/IP socket (Figure 15). A socket is a software abstraction that enables data exchange between two processes through a communication network. TCP socket communication follows a client-server architecture. In Python, a socket is created and configured to work as a server. The server binds to an address (HOST) and a port, and then starts listening for connection requests from a client. Upon simulation start, the Unity application initializes a TCP client socket and actively requests a connection to the Python server by addressing its IP and port. Once the request arrives, the server accepts it and the communication channel is established and ready for data exchange. Data are exchanged between the Python control layer and the Unity simulation using a JSON-based message format. The messages are serialized as JSON string, then encoded in bytes to be sent through the socket. The receiver, which can be both Unity or Python, decodes the bytes in JSON string and convert them to the original message. The choice of introducing this type of communication is driven by the will of keeping the robotic system's logic in Python, in order to easier adapt it to real experimental set up.

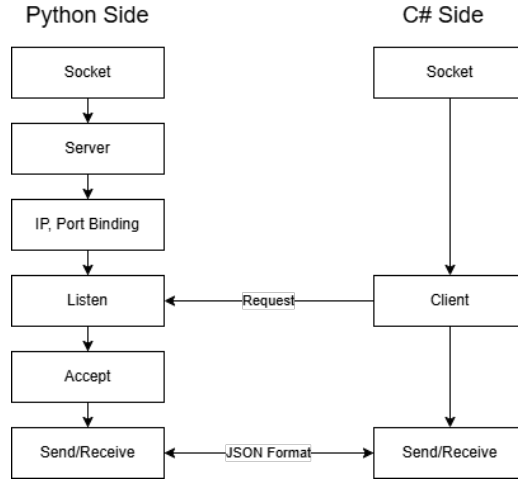


Figure 15: Python-C# communication scheme.

### 3D Modeling of the System's Components

The components of the systems have been designed in Blender, an open source software for 3D modeling, and then imported into the Unity project. The active units' width dimension is 6 cm, while the passives one is 3.5 cm. The overall system is placed over a 1 meter length rail. The introducer sheath and the catheter are designed as rigid hollow tube, while the guidewire implementation will be further discussed in the next chapter. As this work focuses on guidewire navigation and interaction, only the guidewire will be considered in the following modeling, path planning, and navigation analysis.

### 2.3. Guidewire Modelling

To achieve a proper and realistic navigation of the instrument, the design of the guidewire has been implemented in such a way (Figure 16) that it is possible for the guidewire to collide with the internal walls of the vascular phantom, following his natural path and never exiting the boundaries. For this purpose, it was not possible to treat the guidewire as a single rigid body. A system of spheres linked to each other has been created to reproduce the guidewire. To better understand the concept, it is necessary to introduce few Unity components:

- **Rigidbody:** it enables the objects to act under the control of physics. The Rigidbody can receive forces and torque to make objects move in a realistic way.
- **Sphere Collider:** it is a built-in sphere-shaped collider. Different Sphere Collider attached to different objects allow them to collide realistically without let any of them pass through each other.
- **Fixed Joint:** it restricts an object so that its movement is dependent upon another object's movement.

It is possible to divide the chain of spheres that generates the guidewire in different parts (Figure 16): Guidewire Core Sphere, Guidewire Tip, Guidewire Support Spheres and Guidewire Length Visualization Spheres, each of them with a proper function.

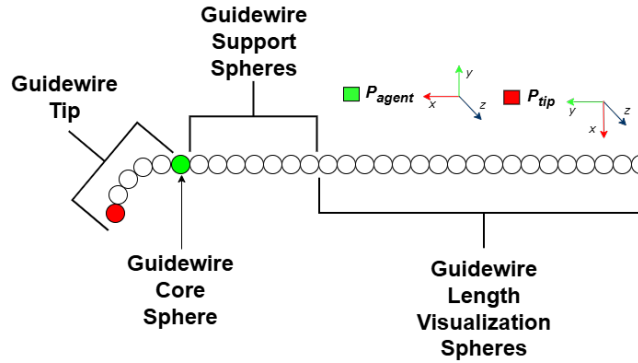


Figure 16: Guidewire design scheme.

#### Guidewire Core Sphere

The **Core Sphere** is the heart of the guidewire and it is composed of a single sphere, to which are transmitted the commands of pushing/pulling and rotation. So, this sphere has the function of moving the whole system that represents the guidewire. The parameters of the Rigidbody component of the core sphere are essential to make the system work. First, a higher mass is assigned to enable it to pull the other elements. Furthermore, Linear and Angular Damping have been tuned to completely dump the linear and angular oscillations of the other spheres and, consequentially, of the whole guidewire, to better mimic the real behavior. Indeed, in real-life scenario, once the units stop pushing or rotating, the instrument is stucked in that last position, without further rotating or retrieving.

#### Guidewire Tip

The **Guidewire Tip** is the curved tip which allows to access the target vessel by orientating the curvature to the direction of the vessel. It has been implemented in Unity via script through the closed-form solutions to the inverse kinematics of continuum robots with one segment. The overall model is the Piecewise Constant Curvature (PCC) model and represents a continuum robot as a sequence of PCC elements, where all states except for bending are set to zero (torsion and axial extension) and is a special case of the piecewise constant strain model [39]. The model needs for three parameters: the arch length  $L$ , the bending angle  $\theta$  and the azimuth angle  $\phi$ . Then, it is possible to compute the position and orientation of the tip respect to the base.

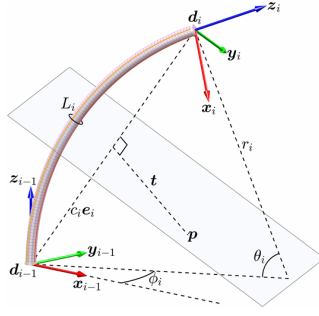


Figure 17: Costant Curvature Model [39].

Taking as reference Figure 17, the bending radius  $r$  is derived as follows:

$$r = L/\theta;$$

Considering  $\phi=0$ , the coordinates of a point on the circular arc of radius  $r$  laying on the  $x_{i-1} - z_{i-1}$  plane centered at  $[r \ 0 \ 0]^T$  are  $p = [r(1 - \cos \theta) \ 0 \ r \sin \theta]^T$ . This motion includes a rotation  $R_y(\theta)$  around the  $+y_{i-1}$  axis by the angle  $\theta$ . Rotating the entire arc about the  $+z$  plane by an angle  $\phi$  (corresponding to the rotation angle of the catheter driver) moves the robot out of the  $x_{i-1} - z_{i-1}$  plane producing a transformation from arc base to the tip equal to:

$$T = \underbrace{\begin{bmatrix} R_z(\phi) & 0 \\ 0 & 1 \end{bmatrix}}_{\text{Rotation}} \underbrace{\begin{bmatrix} R_y(\theta) & p \\ 0 & 1 \end{bmatrix}}_{\text{Inplane Transformation}}$$

Introducing the curvature  $k$ , given that  $k = 1/r$  and  $\theta = ks$  (where  $s \in [0, L]$ ), it is possible to compute the transformation matrix for every point along the arc of length  $L$ :

$$T = \begin{bmatrix} \cos \phi \cos(\kappa s) & -\sin \phi & \cos \phi \sin(\kappa s) & \frac{\cos \phi (1 - \cos(\kappa s))}{\sin \phi (1 - \cos(\kappa s))} \\ \sin \phi \cos(\kappa s) & \cos \phi & \sin \phi \sin(\kappa s) & \frac{\sin \phi (1 - \cos(\kappa s))}{\sin(\kappa s)} \\ -\sin(\kappa s) & 0 & \cos(\kappa s) & \frac{\kappa}{1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inside the simulation environment, the spheres are spawned according to the PCC model. Every sphere of the arch shows a lower mass respect to the core sphere, but a high value of linear and angular damping have been set to further reduce the residual oscillations. With this configuration of the tip, the rotation of the Core Sphere around the guidewire extension axis generates a proper rotation of the tip around the same axis.

## Guidewire Support Spheres

The **Support Spheres** are a set of 7 spheres that are not viewable in the environment but have a critical role in ensuring the right behavior of the guidewire. They are placed behind the Core Sphere and their function is to stabilize the movement of the overall guidewire. Once the walls of the phantom get hit, these spheres preserve the direction of movement and prevent the guidewire from bending over itself and going backward. They have relative low mass compared to the Core Sphere and low Linear and Angular Damping to allow the Core Sphere to rotate and pull the all system.

## Guidewire Length Visualization Spheres

The remaining length of the guidewire is implemented via script. Extending the number of support spheres to cover the entire length was not feasible, as the computational cost required to continuously update the RigidBody positions and to solve the associated Fixed Joint constraints would have been excessively high. Since these spheres are placed only for visualization purpose, they show no physic components as RigidBody or Sphere Collider.

## 2.4. Path Planning

The path planning stage is based on the extraction of vascular centerlines, followed by the construction of a topological graph representing the connectivity of the vessel network, followed by the generation of optimal paths used as reference trajectories for training the learning-based navigation agent. Although the procedure is explained referred to a single vascular model, it has been conducted for all the available vascular models used within the simulation.

### Centerlines Extraction

By centerline it is defined a one-dimensional representation of a tubular anatomical structure, defined as the locus of points equidistant from the structure boundaries. It describes the medial axis of the lumen and preserves the topological and geometric properties of the original three-dimensional model, providing a compact and computationally efficient description of the vessel course. Centerline extraction has been conducted through the "Extract Centerline" module available in 3D Slicer, an open source software for processing and visualization of medical images. The extracted centerlines (Figure 18) were further processed to smooth high-frequency geometric artifacts and eliminate local zig-zag deviations resulting from the discrete sampling of the centerline. The core steps of the processing are the applied filtering methods:

- **Spline Interpolation:** this technique was applied to each individual centerline branch in order to reconstruct a smooth continuous curve from the discretely sampled and uniformly resampled points. The spline fitting process jointly considers all points along the branch, enforcing global geometric smoothness and continuity of the first derivative, thereby reducing medium-scale zig-zag artifacts while preserving the overall vessel course and branch endpoints;
- **Moving Avarage Smoothing:** this technique was applied to further suppress residual high-frequency spatial oscillations. For each point along a branch, the filter replaces the point coordinates with the average of neighboring points within a fixed-size window, performing a local low-pass filtering operation. The window size controls the degree of smoothing, while branch endpoints were explicitly preserved to maintain topological consistency at bifurcations.

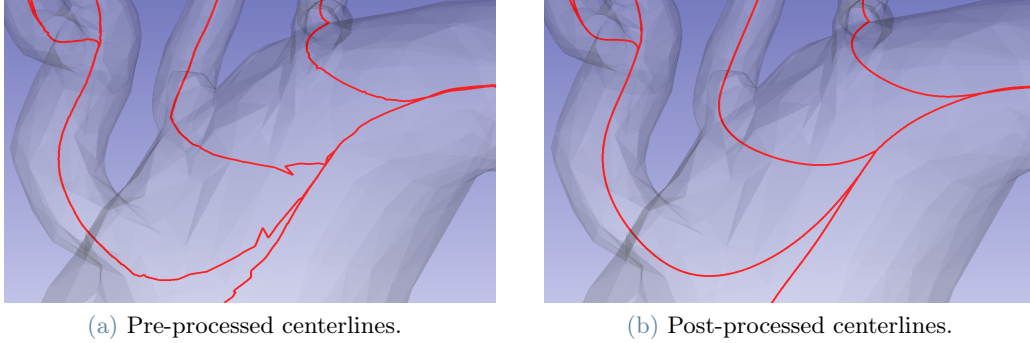


Figure 18: Centerlines post-processing effect.

### Graph Generation

Once the centerlines have been extracted and post-processed, the following stage in order to set up the implementation of the navigation algorithm has been the conversion of the centerlines in a topological graph (Figure 19). Although centerlines already provide a geometric representation of the vascular structure, they are further converted into a graph to explicitly model the topological connectivity of the vessel network. This representation enables the identification of branching points and the definition of discrete navigation decisions, which are not directly captured by raw centerlines. Moreover, the graph structure allows the efficient computation of optimal paths using standard algorithms such as Dijkstra's algorithm. The resulting paths provide structured and consistent reference trajectories, which are particularly suitable for guiding and training the learning-based navigation agent. Within a graph two main components are defined:

- **Node:** it corresponds to a point of topological interest inside the network, such as branch endpoints or bifurcation;
- **Edge:** it represents an individual vessel branch connecting two nodes in the graph and is associated with the geometric centerline segment between them.

Each node is associated with a degree, defined as the number of incident edges. Nodes will never be associated with degree = 2. Indeed, if a point is identified by degree = 2, it means that it is connected with only two other points. Consequentially, it can never be either a bifurcation (degree  $\geq 3$ ) or an endpoint (degree = 1). Finally, it will be a point of an edge.

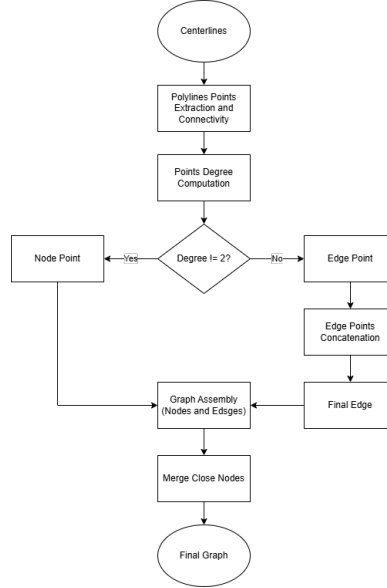


Figure 19: Centerlines to graph generation workflow.

## Path Generation

By converting the centerline network into a graph, the navigation problem is reformulated as a path planning task on a discrete topological structure. The graph has been imported inside the simulation environment and overlapped to the vascular model already present in the scene. Path generation is implemented through the **Dijkstra** algorithm, which finds the path of minimum total length between two given node P and Q within a weighed graph [40]. Simultaneously exploring different routes from the starting node P to the target node Q, it allows to find the shortest path choosing between the route associated with the lowest cost. Even if multiple paths are viable to reach the target, the algorithm ensures that the shortest one will be picked. In the proposed work the cost of the edges is associated with the arc length of the edge itself. Furthermore, since both the catheter tip (start point) and the target position are spawned at runtime, they do not necessarily coincide with graph nodes. In most cases, they lie on the interior of a vessel segment rather than exactly at a bifurcation or endpoint represented by a node. For this reason, a projection-based strategy is adopted. The start and target positions are first projected onto the closest graph edge, and their locations are expressed in terms of curvilinear distance along that edge. Each projection therefore splits the corresponding edge into two virtual sub-segments, connecting the projected point to the two adjacent graph nodes. To compute the shortest path between these two projected points, all possible combinations between the endpoints of the start edge and the endpoints of the target edge are evaluated (Figure 20). For each combination, a standard Dijkstra search is performed on the graph nodes, and the corresponding partial distances from the projected points to the selected endpoints are added. The combination yielding the minimum total path length is selected as the final solution. Finally, the complete path is reconstructed by concatenating:

1. the sub-segment from the start projection to the chosen start-side node,
2. the sequence of graph edges returned by Dijkstra,
3. the sub-segment from the final node to the target projection.

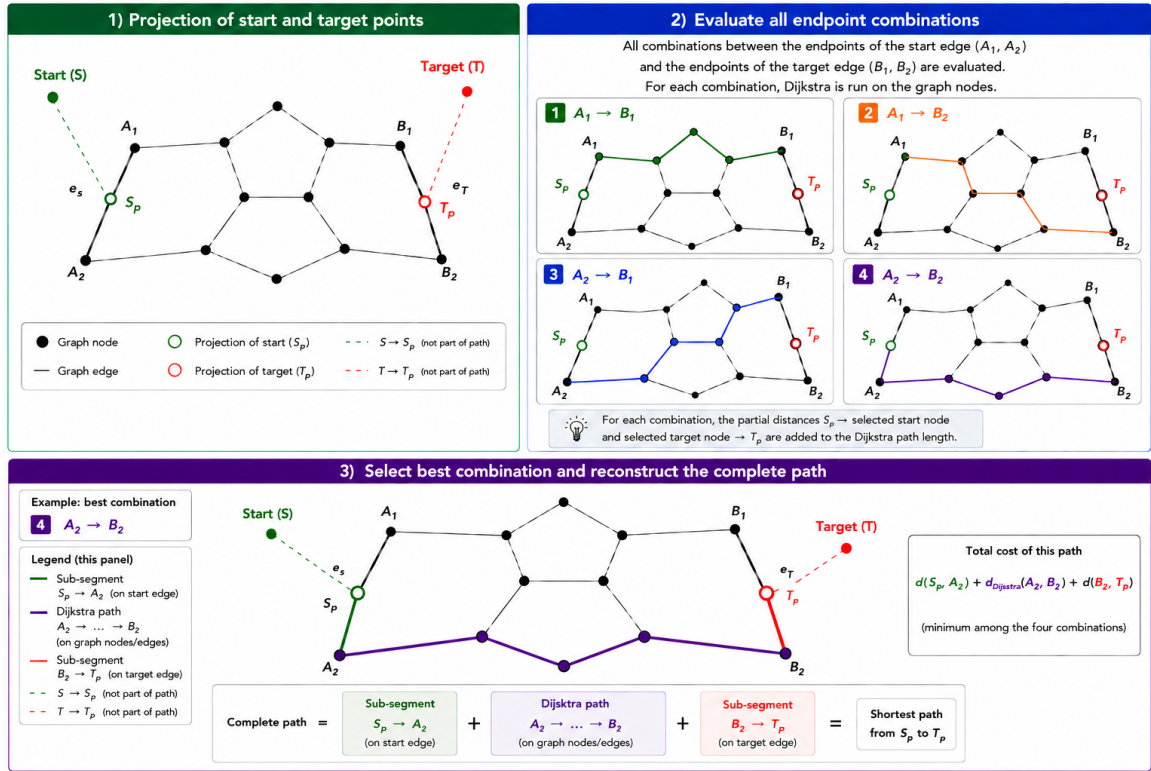


Figure 20: Overview of the proposed shortest-path reconstruction method. The optimal combination is selected according to the minimum total cost, and the final path is reconstructed by combining edge sub-segments with the Dijkstra shortest path on the graph.

## 2.5. AI Based Navigation

AI-based navigation has been conducted inside the Unity Engine simulation environment. Training algorithms in the Unity environment has been possible by installing the Unity ML-Agents package (Figure 21). It allows the user to convert any Unity scene into a learning environment and train character behaviors. More specifically, the package allows the user to:

- Define Agents: entities, or characters, whose behavior will be learned. Agents are entities that generate observations (through sensors), take actions, and receive rewards from the environment.
- Define Behaviors: entities that specify how an agent should act. Multiple agents can share the same Behavior and a scene may have multiple Behaviors.
- Record demonstrations: to show the behaviors of an agent within the Editor. It is possible to use demonstrations to help train a behavior for that agent.
- Embed a trained model in the scene via the Unity Sentis inference engine.

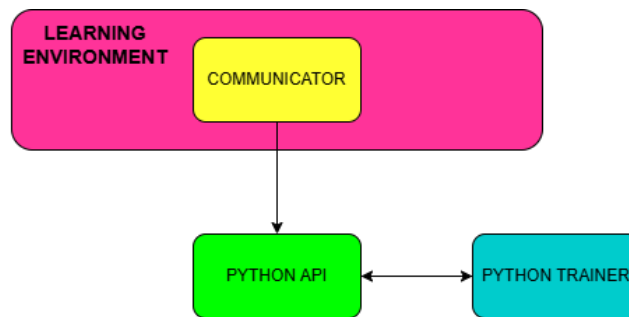


Figure 21: Simplified block diagram of ML-Agents.

Python trainers within the ML-Agents package include RL algorithms like PPO and Soft-Actor-Critic (SAC), or learning from demonstrations algorithms like BC and GAIL. Hyperparameters and settings of the neural networks can then be easily adjusted by the user in a dedicated configuration files.

## Reinforcement Learning

In order to learn a complex task such as endovascular navigation characterized by continuous control, delayed rewards, and dynamic environment interactions, RL algorithms have been chosen to update the parameters of the policy (neural network) used to map states to actions. Unlike supervised learning, where a model learns a direct mapping from input data to predefined target labels, the considered task does not provide a single correct action for each state. Instead, the agent must continuously interact with a simulated environment. At each time step, the agent observes the current state, including spatial configuration, orientation, and environmental constraints, and must output a control signal to actuate the guidewire. The interaction loop between the agent and the environment is illustrated in Figure 22. The objective is not to replicate fixed labeled outputs, but to maximize a cumulative reward function over time. Furthermore, the anatomical navigation scenario involves structural complexity, including vessel bifurcations, constrained motion, and the need for forward and backward maneuvers. Such behaviors require strategic adaptation rather than purely reactive control. RL enables the agent to implicitly learn navigation strategies that balance exploration and exploitation while respecting environmental constraints. Finally, classical path planning algorithms alone would not be sufficient for this task. While graph-based planners can provide high-level navigation guidance, they do not account for the continuous physical interaction between the guidewire and the simulated anatomy. RL complements planning by learning low-level control policies that operate within the physical simulation.

The Reinforcement Learning paradigm can be formulated as an optimization problem. These class of algorithms normally optimizes the  $J(\omega)$  function, show in (1). Typically, it is present a reward function  $r(s_t, a_t)$  that depends on the action  $a_t$  and state  $s_t$  at the timestep  $t$ . This reward function is summed over a sequence of timesteps to get the cumulative reward. Finally, the definition of  $J(\omega)$  becomes the expected value of the cumulative reward over trajectories created by our RL agent.

$$J(\omega) = E_{\tau \sim p_{\omega}(\tau)} \left[ \sum_t r(s_t, a_t) \right] \quad (1)$$

where  $\omega$  denotes the trainable parameters of the policy network,  $\tau$  represents a trajectory sampled from the distribution  $p_{\omega}(\tau)$  induced by the policy, and  $E_{\tau \sim p_{\omega}(\tau)}[\cdot]$  denotes the expectation over all trajectories generated by the current policy. Normally, RL algorithms that do Policy Optimization use gradient ascent in this objective function, as shown in (2).

$$\omega^* = \arg \max_{\omega} J(\omega) \quad (2)$$

To apply gradient ascent, the  $J(\omega)$  function is differentiated with respect to the parameters  $\omega$ , expressed in (3). Inside the (3) equation, the gradients are calculated by multiplying the gradient of the log of policy function  $\pi_{\omega}(a_t | s_t)$  and the cumulative reward  $(\sum_{t'=t}^T r(s_{t'}, a_{t'}))$ .

$$\nabla J(\omega) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\omega} \log \pi_{\omega}(a_t | s_t) \left( \sum_{t'=t}^T r(s_{t'}, a_{t'}) \right) \quad (3)$$

In the proposed work, RL algorithms SAC and PPO are used to train two different model. Indeed, despite their similar nature in terms of application domain, their optimization mechanisms are structurally incompatible within a unified training loop. In the context of the proposed navigation task, the role of both algorithms is conceptually identical: given a structured formulation of the problem defined by observations (state representation), a reward function, and a continuous action space, each method implements a distinct parameter update strategy to optimize the neural network and maximize the expected cumulative reward over time.

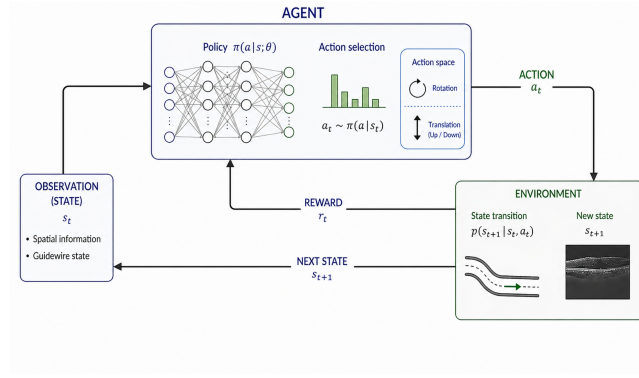


Figure 22: Reinforcement learning pipeline for endovascular navigation. The agent receives a state representation encoding spatial information and guidewire configuration, outputs continuous control actions (translation and rotation), and is trained through reward signals provided by the environment based on navigation performance.

### Proximal Policy Optimization (PPO)

PPO addresses the challenges of stability and sample efficiency in policy optimization by introducing a clipped surrogate objective function [41], as shown in (4). This function prevents large policy updates by clipping  $1 - \epsilon < r_t(\omega) < 1 + \epsilon$ , promoting more stable training.  $r_t(\omega)$  is the coefficient between the policy and the old policy and it comes from importance sampling. The importance sampling trick basically updates values collected by old policies to match what we would have got if we had used the current policy. This means that we can do multiple updates on the same batch of data and remain on-policy.

$$L^{CPI}(\omega) = E \left[ \frac{\pi_{\omega}(a_t | s_t)}{\pi_{\omega_{old}}(a_t | s_t)} \hat{A}_t \right] = E \left[ r_t(\omega) \hat{A}_t \right] \quad (4)$$

The other component of the surrogate loss is the Generalized Advantage Estimator (GAE)  $\hat{A}_t$ , (5). GAE computes the advantage function interpolating the Monte Carlo and Bellman Backup approximation. It is possible to tune the interpolation by changing the hyperparameter  $\lambda$ , where  $\lambda = 0$  is the same as Bellman, and  $\lambda = 1$  is the same as Monte Carlo. In PPO, the normal value is  $\lambda = 0.9$ , so it is more approximated to the Monte Carlo estimation.

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \dots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \quad (5)$$

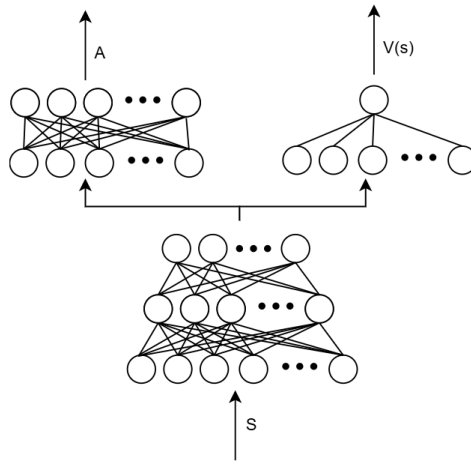


Figure 23: PPO Neural Network architecture [41].

PPO is an Actor-Critic algorithm that uses a single neural network with two heads: one for the actor (policy) and another for the critic (Value function), as shown in Figure 23. This neural network is optimized by minimizing

a loss function with 3 components, where the surrogate loss is one of them. Equation (6) defines the loss  $L$  used in PPO:

$$L = L^{CLIP} - c_1 L^{VF} + c_2 \mathcal{H}(\pi_\omega(\cdot | s_t)) \quad (6)$$

$$L^{CLIP}(\omega) = \hat{E}_t \left[ \min \left( r_t(\omega) \hat{A}_t, \text{clip}(r_t(\omega), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right] \quad (7)$$

$$L^{VF} = (V_\omega(s_t) - V_t^{\text{targ}})^2 \quad (8)$$

where  $L$  denotes the overall PPO loss function,  $L^{CLIP}$  is the clipped surrogate objective used to update the policy,  $L^{VF}$  is the value function loss, and  $\mathcal{H}(\pi_\omega(\cdot | s_t))$  is the entropy of the policy distribution. The coefficients  $c_1$  and  $c_2$  weight the contributions of the value function and entropy terms, respectively. Furthermore,  $r_t(\omega)$  is the probability ratio between the current and previous policies and  $\epsilon$  is the clipping parameter limiting policy updates. In Equation (8),  $V_\omega(s_t)$  is the value predicted by the critic for state  $s_t$ , while  $V_t^{\text{targ}}$  is the corresponding target value used for training.

Equation (7) updates the parameters of the policy, based on the computed advantage and clips the policy update within a range, equation (8) updates the critic part by comparing the estimated value of the reward with the actual obtained reward. Finally, the last term takes into account for entropy, defining the exploration capabilities of the policy.

### Soft Actor-Critic (SAC)

SAC is an Actor-Critic algorithm that uses the two separated neural networks for the actor (policy) and the critic (Q-function), as shown in Figure 24.

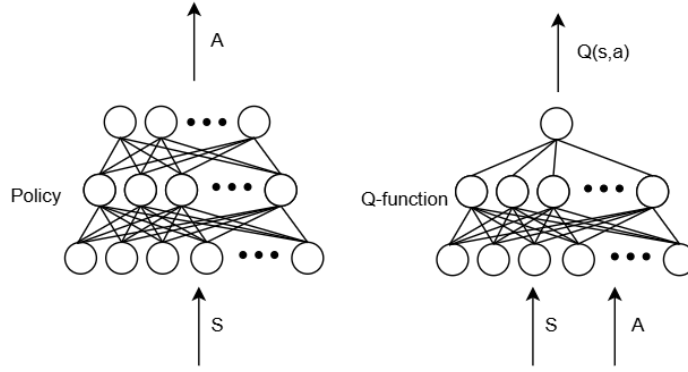


Figure 24: SAC Neural Networks architecture [42].

SAC incorporates the entropy measure of the policy into the reward to encourage exploration, altering the original  $J(\omega)$  [42], as shown in (9). This results in a policy that acts as randomly as possible while it is still able to succeed at the task.

$$J(\omega) = E_{\tau \sim p_\omega(\tau)} \left[ \sum_t r(s_t, a_t) + \alpha \mathcal{H}(\pi_\omega(\cdot | s_t)) \right] \quad (9)$$

To maximize this function, two loss functions to be minimized have been introduced for both neural networks. Firstly, to minimize equation (10) (Actor network) it would be either possible to reduce the value of  $\log(\pi)$  or to increase the  $Q$ -function. It also has been proven that updating the policy weights to minimize this loss maximizes the RL objective. In (10)  $\phi$  denotes the trainable parameters of the actor network,  $\pi_\phi$  is the policy parameterized by  $\phi$ ,  $\mathcal{D}$  is the replay buffer containing previously collected transitions,  $\alpha$  is the temperature coefficient controlling the trade-off between reward maximization and exploration and  $Q_\omega(s_t, a_t)$  is the state-action value estimated by the critic network.

$$L_1 = J_\pi(\phi) = E_{s_t \sim \mathcal{D}} [E_{a_t \sim \pi_\phi} [\alpha \log \pi_\phi(a_t | s_t) - Q_\omega(s_t, a_t)]] \quad (10)$$

Secondly, to learn the Q-function network, the authors utilize a Soft Q-learning method. This method is basically the Bellman Backup with an additional entropy term, as shown in equation (11). Therefore, the second loss ( $L_2 = J_Q(\omega)$ ) is the Mean Squared Error between the left and right side of the equation (11). The entropy component results in adding a bonus reward to actions with less probability.

$$Q(s_t, a_t) \leftarrow r(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1} | s_{t+1}) \quad (11)$$

Another interesting property of SAC is its off-policy nature. This means that the policy is updated with data that were collected by an old policy, making it an off-policy method. Off-policy methods tend to be more unstable; however, its data efficiency and Q-learning style value estimation make SAC a good candidate to train robotic agents.

## Agent, Observations and Actions

In order to train the RL algorithms, it has been needed to correctly set up the environment and to define the observations (state at every time step), the actions and the reward function to be maximized by the agent. Firstly, the graph has been imported into the simulation environment and overlapped with the vessel model. By doing this, the generated paths for the task appear inside the phantom, thus keeping fidelity to the real position of the centerlines with respect to the model geometry. Furthermore, the path was discretized with a constant step size of 0.5 mm. In the proposed work the agent is implemented as the **Core Sphere** of the guidewire, representing the controllable endovascular instrument navigating through the model. The learning logic is encapsulated in a custom script derived from the ML-Agents Agent class, which is attached directly to the sphere. This design allows the agent to perceive its state, execute actions, and receive rewards within the simulated anatomical environment.

The observation space consists of 34 features describing the state of the guidewire and its relation to the planned trajectory. All quantities are expressed in the local reference frame of the agent through the inverse rotation matrix  $\mathbf{R}^{-1} \in R^{3 \times 3}$ . The local position of the guidewire tip (3 observations) is computed as  $\mathbf{o}_{\text{tip,pos}} = \mathbf{R}^{-1}(\mathbf{p}_{\text{tip}} - \mathbf{p}_{\text{agent}})$ . The local orientation of the guidewire tip (3 observations) is represented as  $\mathbf{o}_{\text{curve}} = \mathbf{R}^{-1}\mathbf{d}_{\text{curve}}$ , where  $\mathbf{d}_{\text{curve}} \in R^{3 \times 1}$  denotes the local axis of the guidewire tip used to represent its bending direction. The observation associated with the local target point (3 observations) is computed as  $\mathbf{o}_{\text{target}} = \mathbf{R}^{-1}(\mathbf{p}_{\text{target}} - \mathbf{p}_{\text{closest}})$ , where  $\mathbf{p}_{\text{closest}}$  denotes the closest point along the reference trajectory to the guidewire tip, and  $\mathbf{p}_{\text{target}}$  denotes a target point sampled 4 mm ahead along the path and used to guide local navigation. The future sampled points along the trajectory (15 observations) are defined as  $\mathbf{o}_{\text{future},k} = \mathbf{R}^{-1}(\mathbf{p}_k - \mathbf{p}_{\text{closest}})$ , where  $\mathbf{p}_k$  denotes the  $k$ -th future point sampled at increasing arc-length distances after the local target point, with  $k \in \{1, \dots, 5\}$ . The local tangent direction of the trajectory (3 observations) is represented as:

$$\mathbf{o}_{\text{tangent}} = \mathbf{R}^{-1} \left( \frac{\mathbf{p}_{\text{target}} - \mathbf{p}_{\text{closest}}}{\|\mathbf{p}_{\text{target}} - \mathbf{p}_{\text{closest}}\|} \right) \quad (12)$$

The final navigation goal (3 observations) is represented as  $\mathbf{o}_{\text{goal}} = \mathbf{R}^{-1}(\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{closest}})$ . The normalized direction vector from the guidewire tip to the local target point (3 observations) and the Euclidean distance between them (1 observation) are computed as:

$$\mathbf{o}_{\text{dir}} = \mathbf{R}^{-1} \left( \frac{\mathbf{p}_{\text{target}} - \mathbf{p}_{\text{tip}}}{\|\mathbf{p}_{\text{target}} - \mathbf{p}_{\text{tip}}\|} \right), \quad o_{\text{dist}} = \|\mathbf{p}_{\text{target}} - \mathbf{p}_{\text{tip}}\| \quad (13)$$

Finally, in every iteration step, the agent output is represented by two continuous actions:

1. **Pushing/Pulling along the guidewire extension direction.**
2. **Rotating clockwise/counter clockwise with 360° range of motion.**

The actions can be both performed simultaneously and they are the only two actions allowed in real-case scenario.

## Reward Function

The definition of the reward function is crucial for the purpose of completing the task. It directly shapes the learning dynamics by guiding the agent toward desirable behaviors while discouraging inefficient or unsafe actions. First, the theoretical desired behavior has been identified as a combination of three factors:

1. Progress along the path in order to proceed toward the target,
2. Orientation of the tip toward the centerlines in order to facilitate navigation through vascular bifurcations,
3. Minimization of procedure duration.

Based on the above mentioned constraints different types of negative and positive rewards have been implemented to induce the correct behavior of the agent.

More specifically, to address the condition of point 1, at every time step:

- Reward = +0.0007 is added if the cumulative progress along the path is greater than the progress of the previous step; otherwise a reward = -0.0002 is added.

- Reward = +0.2 if the agent reaches  $\frac{1}{4}$  of the total path, reward = +0.5 if the agent reaches  $\frac{1}{2}$  of the total path, reward = +1 if the agent reaches  $\frac{3}{4}$  of the total path, reward = 3 if the target reaches the target. This method is the "Milestone" approach and it is useful in navigating through complex environment, when the global target is difficult to reach, especially for the first phase of the training.

To deal with the condition of point 2, at every time step:

- Reward =  $-(DistanceToPath * 0.0001)$  is added, with "DistanceToPath" being the distance between the tip of the guidewire and the closest point along the path.

As it regards point 3, instead, at every time step:

- Reward = -0.0003 is added.

Finally, two different conditions have been implemented to terminate unsuccessful episodes:

- Reward = -5 and end of the episode if no cumulative progress along the path is observed for more than 5000 steps.
- Reward  $\in [-5, -2]$  based on the agent's progress along the path and end of the episode if the distance between the tip and the closest point along the path overcome a defined threshold.

## Learning from Demonstration

Since in learning controls the learning task are less well defined, no direct training data exist which could be used to learn a controller in a supervised way. In such a setting, typical for reinforcement learning, learning a task from scratch can require a prohibitively time consuming amount of exploration of the state-action space in order to find a good policy. Learning from demonstration (LfD) refers to the process used to transfer new skills to a machine by relying on demonstrations from a user, thus drastically reducing the time for the algorithm to correctly update the policy parameters. In LfD, the demonstrations can take various forms, requiring the design of interfaces, devices, and control techniques that can be used for efficient collection of data. Some of these can be linked to the social learning process in human interactions, such as observational learning and kinesthetic teaching [43]. In the proposed work the demonstration recording approach is observational teaching inside the simulation environment. It refers to the observation of a demonstrator achieving a task with the goal of reproducing it. The agent is moved manually by the user, while the episodes are recorded with the shape of state-actions pairs. Within the proposed work, two different LfD algorithms have been used: Behavioral Cloning (BC) and GAIL. They are used both separately and in combination to speed-up the training of the core algorithms PPO and SAC. Although BC and GAIL are both imitation learning approaches, they rely on fundamentally different optimization principles. Their combination is therefore not redundant but complementary, both from an optimization and a theoretical perspective. Indeed, BC enforces local imitation trying to copy expert actions given a determined state, while GAIL enforces global behavior imitation trying to match the distribution of the action-state pairs along the all trajectories.

### Behavioral Cloning (BC)

Behavioral Cloning is the simplest approach of imitation learning in RL. Assumed that the user collected from a running demonstrator policy  $\pi^b$  a dataset:

$$D = \{(s_1^t, a_1^t, \dots, s_H^t, a_H^t)\}_{t=1}^T \quad (14)$$

so that  $(s_1^t, a_1^t, \dots, s_H^t, a_H^t)$  a trajectory rollout of  $\pi^b$ . The demonstrator dataset does not include label rewards. BC attempts to fit a policy  $\pi^c$  to match the distribution of actions observed in  $D$ . It works with a supervised learning approach, in which  $\pi^c$  tries to predict an action  $a$  given  $(s, a) \in \mathcal{D}$  [44]. The natural choice for  $\pi^c$  model the empirical action distribution in  $D$ :

$$\pi_h^c(a | s) := \begin{cases} \frac{T_h(s, a)}{T_h(s)} & \text{se } T_h(s) > 0, \\ \text{unif}(\mathcal{A}) & \text{se } T_h(s) = 0. \end{cases} \quad (15)$$

where  $T_h(s, a)$  is the probability of the action  $a$  given the state  $s$  based on the expert demonstrations, while  $T_h(s) = 0$  if the state has not been observed in the recorded trajectories.

### General Adversarial Imitation Learning (GAIL)

GAIL formulates imitation learning as a minimize-maximization optimization problem between:

- A policy  $\pi_\theta$ , parameterized by  $\theta$ ,
- A discriminator  $D_\phi(s, a)$ , parameterized by  $\phi$ .

The discriminator aims to distinguish expert state–action pairs from those generated by the learned policy. The policy, on the contrary, attempts to generate state–action pairs that are indistinguishable from expert demonstrations [45]. The objective function is:

$$\min_{\pi} \max_D E_{(s,a) \sim \pi_E} [\log D(s,a)] + E_{(s,a) \sim \pi} [\log (1 - D(s,a))] - \lambda H(\pi), \quad (16)$$

where:

- $D(s,a) \in (0,1)$  is the probability that  $(s,a)$  originates from the expert,
- $H(\pi)$  denotes the causal entropy of the policy,
- $\lambda > 0$  is a regularization coefficient.

The entropy term is defined as:

$$H(\pi) = E_{\pi} [-\log \pi(a | s)], \quad (17)$$

which encourages stochasticity and exploration.

### 3. Experimental Set-up

#### Anatomical Dataset

In total, four different anatomic models acquired by CT scan have been used to train and validate the models (Figure 25). The training phase has been performed on a type-3 aortic arch according to the Madwhal criteria, which represents the most challenging configuration of the aortic arch for navigation. Validation then occurred over three different aortic arch model, in type-1, type-2 and type-3 configuration.

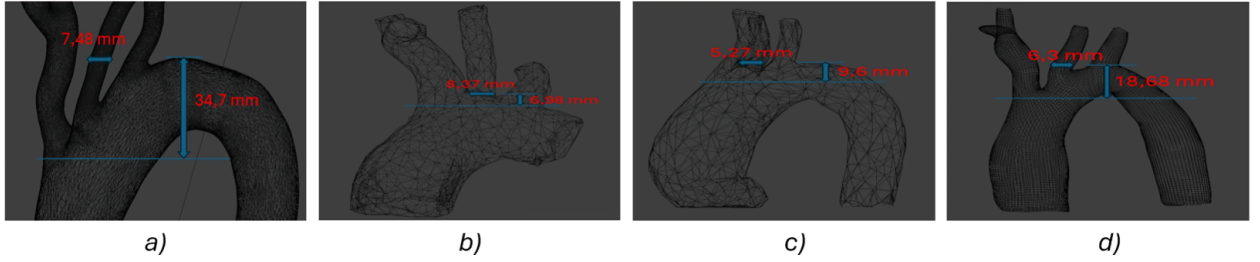


Figure 25: Aortic arch models included in the dataset. The training set includes a Type III (M3<sub>1</sub>) configuration (a), while the validation set comprises Type I (M1) [46] (b), Type II (M2) [47] (c), and Type III (M3<sub>2</sub>) [48] (d) geometries.

#### Experimental Task

The task of the proposed experiment (Figure 26) is to train a RL model to correctly perform a guidewire autonomous over-arching of the aortic arch without any need for external support inside the Unity simulation environment. More specifically, to mimick the real-case scenario of transcatheter procedures such as carotid artery stenting through transfemoral approach, the navigation starting point is located along the descending treat of the aorta. The target point, instead, is placed along the brachiocephalic artery, the left common carotid and the right subclavian artery. Depending on the specific target point, the model should be able to navigate the guidewire from the descending treat of the aorta to the aortic arch and, subsequently, to successfully cannulate the guidewire in the right branch.

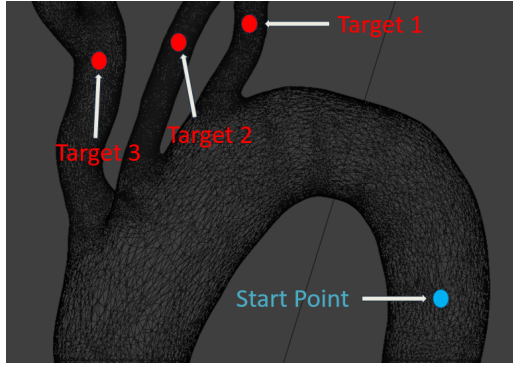


Figure 26: Task illustration: start point, target points.

## Training Hyperparameters Setting

Hyperparameters are configuration variables defined prior to the training process that govern the learning dynamics of a model but are not directly optimized during training. Their selection significantly influences convergence speed, training stability, and final performance, and they are typically determined through empirical tuning or prior domain knowledge. Many hyperparameters are common to different algorithms, while others apply specifically to selected one. The configuration of parameters and hyperparameters for PPO and SAC is listed in Table 4, while BC and GAIL ones are shown in Table 5. When both methods are used jointly (BC and GAIL), the parameters must be properly configured for each of them.

| Hyperparameter | Description                                   | PPO    | SAC    |
|----------------|-----------------------------------------------|--------|--------|
| Batch size     | Experiences per gradient update               | 512    | 256    |
| Buffer size    | PPO: collected steps; SAC: replay buffer size | 4096   | 200000 |
| Learning rate  | Step size of gradient updates                 | 3e-4   | 3e-4   |
| $\beta$        | Entropy regularization                        | 0.001  | –      |
| $\epsilon$     | PPO clipping parameter                        | 0.2    | –      |
| $\lambda$      | GAE parameter                                 | 0.95   | –      |
| Epochs         | Optimization passes                           | 3      | –      |
| $\tau$         | Target network update                         | –      | 0.005  |
| Entropy coeff. | SAC exploration parameter                     | –      | 0.5    |
| Steps/update   | Env steps between updates                     | –      | 10     |
| Warmup steps   | Steps before updates start                    | –      | 10000  |
| Normalize      | Observation normalization                     | Yes    | Yes    |
| Hidden units   | Units per layer                               | 128    | 256    |
| Layers         | Number of layers                              | 2      | 3      |
| Reward scale   | Scaling factor                                | 1      | 1      |
| $\gamma$       | Discount factor                               | 0.99   | 0.99   |
| Max steps      | Training steps                                | 600000 | 600000 |
| Time horizon   | Steps before adding to buffer                 | 128    | 512    |

Table 4: Hyperparameter configuration used for PPO and SAC.

| Hyperparameter | Description                          | BC                           | GAIL                       |
|----------------|--------------------------------------|------------------------------|----------------------------|
| Strength       | Reward scaling factor                | 0.4 (alone), 0.3 (with GAIL) | 0.3 (alone), 0.1 (with BC) |
| $\gamma$       | Discount factor                      | –                            | 0.99                       |
| Learning rate  | Step size of gradient updates        | –                            | 3e-4                       |
| Steps          | Optimization steps on demonstrations | 80000                        | –                          |
| Demo path      | Number of demonstration trajectories | 25 demos                     | 25 demos                   |

Table 5: Hyperparameter configuration used for BC and GAIL.

## Evaluation Metrics

The performance of the proposed reinforcement learning agents is evaluated using quantitative metrics that assess both learning behavior and task efficiency.

The **average cumulative reward** per episode is monitored throughout training to assess convergence behavior and learning stability:

$$\bar{R} = \frac{1}{N} \sum_{i=1}^N \left( \sum_{t=0}^{T_i} r_t^{(i)} \right) \quad (18)$$

where  $r_t^{(i)}$  denotes the reward received at timestep  $t$  during episode  $i$ ,  $T_i$  represents the duration of episode  $i$ , and  $N$  is the number of considered episodes.

In addition, the **episode length**, defined as the number of steps required to complete an episode, is analyzed as an indicator of navigation efficiency and trajectory optimization.

The **success rate** is defined as the percentage of episodes in which the agent successfully reaches the target branch without violating predefined safety constraints. This metric provides a direct measure of task completion reliability.

The **average time to target** is used to evaluate navigation efficiency. It is computed as the total elapsed simulation time required for the agent to reach the desired vascular branch. Given the clinical relevance of procedural duration in endovascular interventions, this metric represents a key performance indicator.

To statistically evaluate the differences among the considered algorithms, appropriate non-parametric statistical tests were applied. Differences in success rate were analyzed using the **Chi-square** test. When a statistically significant difference was detected, pairwise comparisons were performed using **Fisher's exact test**, which is more appropriate for binary outcomes and small sample sizes. For procedure time, the **Mann-Whitney U** test was used for statistical comparisons with respect to the PPO baseline model, which was used as reference. In both cases, Holm correction was applied to adjust the p-values and control the family-wise error rate due to multiple comparisons. The **success rate** and the **average time to target** are computed over 20 independent episodes for each target branch.

## 4. Results

### PPO Based Algorithms Training Performance

All PPO-based configurations demonstrated stable convergence during training (Figure 27). The learning curves show a progressive increase in cumulative reward accompanied by a reduction in episode length, indicating improved efficiency in reaching the target. Final success rates were consistently high across PPO configurations, although differences were observed in convergence speed and completion time.

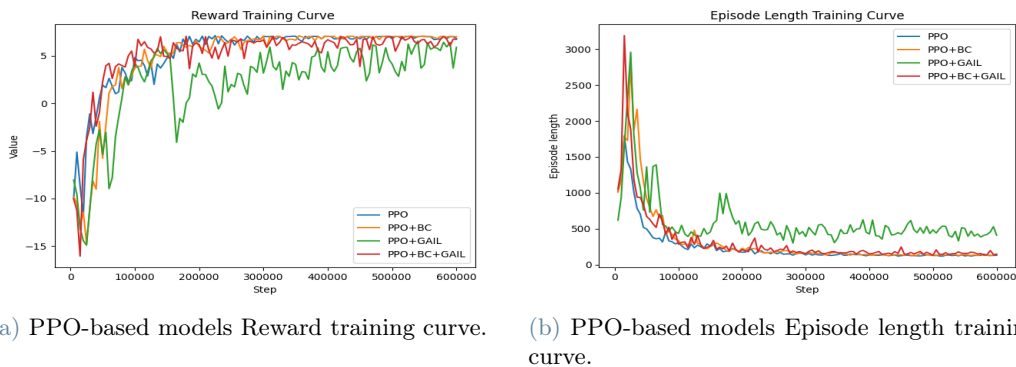


Figure 27: PPO Learning phase dynamics.

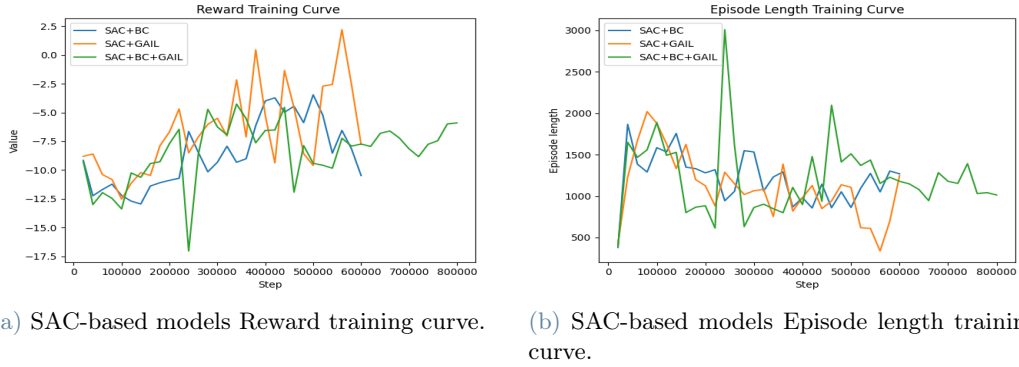
Among the tested setups, the best-performing configuration achieved a success rate of almost 100% for every target with a low average procedure time.

| Algorithm(s)       | Success Rate |          |          | Procedure Time (s) |                    |                   |
|--------------------|--------------|----------|----------|--------------------|--------------------|-------------------|
|                    | Target 1     | Target 2 | Target 3 | Target 1           | Target 2           | Target 3          |
| <b>PPO</b>         | 100%         | 100%     | 100%     | $22.62 \pm 1.00$   | $42.16 \pm 1.91$   | $24.79 \pm 1.31$  |
| <b>PPO+BC</b>      | 95%          | 100%     | 100%     | $23.69 \pm 1.98$   | $45.90 \pm 3.03$   | $26.02 \pm 6.12$  |
| <b>PPO+GAIL</b>    | 100%         | 85%      | 95%      | $51.48 \pm 8.57$   | $133.78 \pm 20.24$ | $49.34 \pm 19.07$ |
| <b>PPO+BC+GAIL</b> | 100%         | 75%      | 100%     | $25.04 \pm 1.80$   | $53.89 \pm 3.94$   | $35.12 \pm 13.99$ |

**Table 6:** Training performance of PPO-based configurations on the Type-3 aortic arch. Success rate and completion time (mean  $\pm$  standard deviation) are computed over 20 evaluation episodes. Target 1: cannulation in the left subclavian artery; Target 2: cannulation in the left common carotid artery; Target 3: cannulation in the brachiocephalic artery.

### SAC Based Algorithms Training Performance

In contrast, SAC-based configurations exhibited unstable training behavior (Figure 28). Learning curves revealed significant oscillations in cumulative reward and inconsistent reductions in episode length. Despite extensive hyperparameter tuning, SAC did not reach satisfactory convergence on the Type-3 anatomy.



**Figure 28:** SAC Learning phase dynamics.

| Algorithm(s)       | Success Rate | Procedure Duration |
|--------------------|--------------|--------------------|
| <b>SAC+BC</b>      | 0%           | N/a                |
| <b>SAC+GAIL</b>    | 0%           | N/a                |
| <b>SAC+BC+GAIL</b> | 0%           | N/a                |

**Table 7:** Training performance of SAC-based configurations on the Type-3 aortic arch. Success rate and completion time (mean  $\pm$  standard deviation) are computed over 20 evaluation episodes.

### Validation on Unseen Anatomies

Following the training phase conducted on the Type-3 aortic arch, the best-performing PPO-based configurations were evaluated on unseen anatomical models in order to assess generalization capabilities. Validation was performed on Type-1, Type-2 and Type-3 aortic arch geometries, which were not encountered during training. Results on the Type-1 anatomy show consistently high success rates across configurations, with moderate variability in completion time depending on the target. Similarly, validation on the more complex Type-2 and Type-3 anatomies demonstrates the ability of PPO-based policies to adapt to different geometric configurations, although variations in performance are observed across individual targets.

Detailed results for each target are reported in Tables 8, 9 and 10 for Type-1, Type-2 and Type-3 anatomies, respectively.

| Algorithm(s)       | Success Rate |          |          | Procedure Time (s) |                  |                  |
|--------------------|--------------|----------|----------|--------------------|------------------|------------------|
|                    | Target 1     | Target 2 | Target 3 | Target 1           | Target 2         | Target 3         |
| <b>PPO</b>         | 100%         | 100%     | 60%      | $5.47 \pm 1.47$    | $9.36 \pm 0.89$  | $11.18 \pm 1.37$ |
| <b>PPO+BC</b>      | 100%         | 100%     | 100%     | $6.23 \pm 0.72$    | $10.88 \pm 1.21$ | $10.36 \pm 0.30$ |
| <b>PPO+GAIL</b>    | 85%          | 85%      | 10%      | $9.28 \pm 1.67$    | $29.39 \pm 5.62$ | $43.57 \pm 0.66$ |
| <b>PPO+BC+GAIL</b> | 90%          | 100%     | 75%      | $5.44 \pm 0.89$    | $10.36 \pm 0.84$ | $11.28 \pm 1.23$ |

**Table 8:** Validation performance of PPO-based configurations on the Type-1 aortic arch. Success rate and completion time (mean  $\pm$  standard deviation) are computed over 20 evaluation episodes. Target 1: cannulation in the left subclavian artery; Target 2: cannulation in the left common carotid artery; Target 3: cannulation in the brachiocephalic artery.

| Algorithm(s)       | Success Rate |          |          | Procedure Time (s) |                  |                  |
|--------------------|--------------|----------|----------|--------------------|------------------|------------------|
|                    | Target 1     | Target 2 | Target 3 | Target 1           | Target 2         | Target 3         |
| <b>PPO</b>         | 100%         | 85%      | 100%     | $10.60 \pm 3.19$   | $12.79 \pm 3.03$ | $13.89 \pm 0.79$ |
| <b>PPO+BC</b>      | 100%         | 80%      | 100%     | $9.33 \pm 0.57$    | $12.93 \pm 0.94$ | $13.65 \pm 0.94$ |
| <b>PPO+GAIL</b>    | 40%          | 70%      | 100%     | $33.17 \pm 10.90$  | $29.01 \pm 5.49$ | $21.85 \pm 1.99$ |
| <b>PPO+BC+GAIL</b> | 100%         | 80%      | 100%     | $9.66 \pm 0.99$    | $13.61 \pm 2.49$ | $13.91 \pm 0.77$ |

**Table 9:** Validation performance of PPO-based configurations on the Type-2 aortic arch. Success rate and completion time (mean  $\pm$  standard deviation) are computed over 20 evaluation episodes. Target 1: cannulation in the left subclavian artery; Target 2: cannulation in the left common carotid artery; Target 3: cannulation in the brachiocephalic artery.

| Algorithm(s)       | Success Rate |          |          | Procedure Time (s) |                  |                   |
|--------------------|--------------|----------|----------|--------------------|------------------|-------------------|
|                    | Target 1     | Target 2 | Target 3 | Target 1           | Target 2         | Target 3          |
| <b>PPO</b>         | 100%         | 100%     | 95%      | $7.20 \pm 0.13$    | $10.46 \pm 1.41$ | $14.61 \pm 1.65$  |
| <b>PPO+BC</b>      | 100%         | 100%     | 95%      | $8.93 \pm 0.59$    | $12.36 \pm 0.62$ | $19.95 \pm 0.60$  |
| <b>PPO+GAIL</b>    | 100%         | 85%      | 80%      | $20.94 \pm 2.80$   | $42.85 \pm 8.02$ | $57.81 \pm 23.03$ |
| <b>PPO+BC+GAIL</b> | 100%         | 100%     | 95%      | $7.57 \pm 0.19$    | $11.13 \pm 0.37$ | $15.52 \pm 0.93$  |

**Table 10:** Validation performance of PPO-based configurations on the Type-3 aortic arch. Success rate and completion time (mean  $\pm$  standard deviation) are computed over 20 evaluation episodes. Target 1: cannulation in the left subclavian artery; Target 2: cannulation in the left common carotid artery; Target 3: cannulation in the brachiocephalic artery.

## Statistical Analysis of RL Algorithms

The success rate analysis, which represents a binary outcome (success or failure), was performed using a global comparison among the models based on the chi-square test (Table 11). When significant differences were observed, pairwise comparisons were conducted using Fisher’s exact test. Holm correction was applied to adjust the p-values and control for multiple comparisons, ensuring a rigorous evaluation of statistical significance across model configurations (Table 12, Table 13 and Table 14).

For the procedure time analysis, pairwise comparisons were performed separately using the Mann–Whitney U-test, considering only successful episodes. Since procedure time may vary across target branches, comparisons were conducted separately for each target to avoid aggregation bias. In this case, statistical comparisons were conducted with respect to the PPO baseline model, which was used as reference. Holm correction was applied to adjust the p-values and control the family-wise error rate due to multiple comparisons (Table 15, Table 16 and Table 17).

| Phantom Model | Metric       | p-value           |
|---------------|--------------|-------------------|
| Type 1        | Success Rate | $p < 0.001^{***}$ |
| Type 2        | Success Rate | $p < 0.001^{***}$ |
| Type 3        | Success Rate | $p = 0.010^*$     |

Table 11: Global statistical comparison of success rate across RL models on Type-1, Type-2 and Type-3 phantoms using the global chi-square test. Statistical significance was defined as follows: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ .

|             | PPO     | PPO+BC          | PPO+GAIL        | PPO+BC+GAIL |
|-------------|---------|-----------------|-----------------|-------------|
| PPO         | /       | 0.018*          | 0.006**         | 1.000       |
| PPO+BC      | 0.018*  | /               | $< 0.001^{***}$ | 0.026*      |
| PPO+GAIL    | 0.006** | $< 0.001^{***}$ | /               | 0.003**     |
| PPO+BC+GAIL | 1.000   | 0.026*          | 0.003**         | /           |

Table 12: Pairwise statistical comparison of success rate between models on the Type-1 phantom using Fisher’s exact test with Holm correction. Values represent adjusted p-values. Statistical significance: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ .

|             | PPO     | PPO+BC  | PPO+GAIL | PPO+BC+GAIL |
|-------------|---------|---------|----------|-------------|
| PPO         | /       | 1.000   | 0.003**  | 1.000       |
| PPO+BC      | 1.000   | /       | 0.008**  | 1.000       |
| PPO+GAIL    | 0.003** | 0.008** | /        | 0.008**     |
| PPO+BC+GAIL | 1.000   | 1.000   | 0.008**  | /           |

Table 13: Pairwise statistical comparison of success rate between models on the Type-2 phantom using Fisher’s exact test with Holm correction. Values represent adjusted p-values. Statistical significance: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ .

|             | PPO   | PPO+BC | PPO+GAIL | PPO+BC+GAIL |
|-------------|-------|--------|----------|-------------|
| PPO         | /     | 1.000  | 0.367    | 1.000       |
| PPO+BC      | 1.000 | /      | 0.367    | 1.000       |
| PPO+GAIL    | 0.367 | 0.367  | /        | 0.367       |
| PPO+BC+GAIL | 1.000 | 1.000  | 0.367    | /           |

Table 14: Pairwise statistical comparison of success rate between models on the Type-3 phantom using Fisher’s exact test with Holm correction. Values represent adjusted p-values. Statistical significance: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ .

|             | Target 1        | Target 2        | Target 3 |
|-------------|-----------------|-----------------|----------|
| PPO+BC      | $< 0.001^{***}$ | $< 0.001^{***}$ | 0.551    |
| PPO+GAIL    | $< 0.001^{***}$ | $< 0.001^{***}$ | 0.065    |
| PPO+BC+GAIL | 0.320           | $< 0.001^{***}$ | 0.826    |

Table 15: Pairwise statistical comparison of procedure time across target branches for the Type-1 phantom using the Mann–Whitney U-test with Holm correction. Values represent adjusted p-values with respect to the PPO baseline. Statistical significance: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ . Target 1: left subclavian artery; Target 2: left common carotid artery; Target 3: brachiocephalic artery.

|                    | Target 1   | Target 2   | Target 3   |
|--------------------|------------|------------|------------|
| <b>PPO+BC</b>      | 0.020*     | 0.095      | 0.546      |
| <b>PPO+GAIL</b>    | < 0.001*** | < 0.001*** | < 0.001*** |
| <b>PPO+BC+GAIL</b> | 0.093      | 0.095      | 0.903      |

**Table 16:** Pairwise statistical comparison of procedure time across target branches for the Type-2 phantom using the Mann–Whitney U-test with Holm correction. Values represent adjusted p-values with respect to the PPO baseline. Statistical significance: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ . Target 1: left subclavian artery; Target 2: left common carotid artery; Target 3: brachiocephalic artery.

|                    | Target 1   | Target 2   | Target 3   |
|--------------------|------------|------------|------------|
| <b>PPO+BC</b>      | < 0.001*** | < 0.001*** | 0.001**    |
| <b>PPO+GAIL</b>    | < 0.001*** | < 0.001*** | < 0.001*** |
| <b>PPO+BC+GAIL</b> | < 0.001*** | < 0.001*** | 0.152      |

**Table 17:** Pairwise statistical comparison of procedure time across target branches for the Type-3 phantom using the Mann–Whitney U-test with Holm correction. Values represent adjusted p-values with respect to the PPO baseline. Statistical significance: \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ . Target 1: left subclavian artery; Target 2: left common carotid artery; Target 3: brachiocephalic artery.

## Tip Geometry Configuration

The optimal tip configuration, defined according to the constant curvature model, was evaluated in terms of both diameter and tip length. The results, summarized in Table 18, show that the best-performing configuration varies across different aortic arch anatomies. In particular, both geometric parameters were found to significantly influence navigation performance, suggesting that a single tip design may not be optimal for all anatomical scenarios.

For each aortic model, a specific combination of diameter, length and curvature yielded the highest success rate, highlighting the importance of tailoring the guidewire tip geometry to the vascular structure. Interestingly, although the agent was trained using a fixed tip configuration, the optimal parameters identified during validation differed depending on the anatomical setting.

| Phantom Model | Diameter (mm) | Length (mm) | Curvature ( $\theta$ ) | Bending Radius (mm) |
|---------------|---------------|-------------|------------------------|---------------------|
| <b>Type 1</b> | 2             | 6,5         | 80°                    | 4,66                |
| <b>Type 2</b> | 1,8           | 5           | 150°                   | 1,91                |
| <b>Type 3</b> | 1,6           | 10          | 100°                   | 5,73                |

**Table 18:** Optimal tip configuration for each phantom model, defined in terms of diameter, length, and curvature angle  $\theta$  according to the constant curvature model.

## 5. Discussion

This section provides an interpretation of the obtained results, with the aim of highlighting the main findings and their implications in the context of autonomous endovascular navigation. In particular, the performance of the proposed approaches is analyzed across different aortic arch configurations, together with the limitations observed in specific learning strategies. Finally, the results are discussed with respect to the current literature, focusing on anatomical variability, generalization capabilities, and the role of imitation learning in reinforcement learning frameworks.

### Performance of PPO-based approaches

Across all anatomical configurations, PPO-based approaches demonstrated generally robust and consistent performance, although statistically significant differences were observed among models.

For the Type-1 configuration, high success rates were achieved for the left subclavian artery and left common carotid artery across all models, reflecting their relatively simple geometry and favorable branching angles. In contrast, the brachiocephalic artery exhibited higher variability due to its more complex orientation and the proximity to adjacent branches, which increases the likelihood of incorrect vessel selection. In this setting, PPO achieved consistently high performance, while PPO+GAIL showed a clear degradation in both success rate and efficiency. The inclusion of Behavioral Cloning led to slightly improved robustness in terms of success rate, whereas the addition of GAIL did not provide consistent benefits.

A similar trend was observed in the Type-2 configuration. While most targets remained easily accessible, increased variability was observed for the left common carotid artery due to its bovine-like anatomical configuration, which introduces ambiguity in branch selection. Also in this case, PPO maintained stable performance, while PPO+GAIL consistently underperformed. The remaining configurations (PPO+BC and PPO+BC+GAIL) showed comparable success rates and only minor, target-dependent differences in procedure time.

In the Type-3 configuration, both for training and validation phantoms, all models achieved high success rates across targets, with no significant pairwise differences. Interestingly, the brachiocephalic artery showed improved performance compared to the Type-1 configuration, likely due to the increased spatial separation between branches, which reduces navigation ambiguity. However, differences in navigation efficiency remained evident, with PPO+GAIL consistently resulting in longer procedure times. The impact of Behavioral Cloning remained variable across targets, without providing a consistent improvement.

Overall, these results highlight that PPO provides stable and reliable performance across different anatomical configurations. The integration of GAIL consistently degrades performance, both in terms of success rate and navigation efficiency. In contrast, the effect of Behavioral Cloning depends on the considered metric. While success rates remain comparable to PPO across most targets, a slight increase in procedure time is observed when BC is used alone. Interestingly, when combined with GAIL, the execution time becomes comparable to PPO, suggesting that the addition of GAIL compensates for the slight increase in procedure time introduced by Behavioral Cloning.

## SAC Failure Analysis

SAC did not achieve stable learning behavior in the considered environment and was therefore excluded from the final quantitative comparison. During training, the agent frequently produced erratic navigation strategies and failed to consistently reach the target vessels. In particular, the learned policy struggled to maintain trajectories aligned with the reference centerline and often generated unstable guidewire motions.

This behavior can be attributed to the interaction between the characteristics of SAC and the constraints of the proposed navigation task. SAC is an off-policy algorithm based on experience replay and entropy maximization. Although these properties are beneficial in many continuous control problems, they may become disadvantageous in endovascular navigation, where the action space is highly constrained and the guidewire must perform smooth and precise directional adjustments inside narrow vascular structures. Excessive exploration can therefore produce unstable trajectories and make it difficult for the agent to learn consistent branch-selection strategies. Another relevant factor is the sequential nature of the task. The reward function is strongly related to progressive advancement along the centerline and local alignment with the target path. As a consequence, successful behavior requires temporally coherent actions over long horizons. Since SAC updates the policy using replayed transitions collected from previous interaction phases, the learning process may be affected by samples that are less representative of the current policy behavior. This can hinder the acquisition of stable navigation strategies along the centerline.

The integration of imitation learning techniques, such as Behavioral Cloning and GAIL, did not improve SAC training stability. This suggests that an initial demonstration-based guidance was not sufficient to compensate for the instability observed during off-policy learning. Although several hyperparameter configurations were tested, SAC remained highly unstable across trials. Therefore, while further tuning may improve its behavior, the observed failure is likely related not only to parameter selection but also to the mismatch between the algorithmic properties of SAC and the requirements of the proposed navigation task.

## Implications With Respect to Literature Gaps

The obtained results directly address several limitations identified in the current literature:

1. The evaluation on Type-3 aortic arch configurations demonstrates the feasibility of training and validating reinforcement learning policies in anatomically complex scenarios, which have been largely overlooked in previous works;

2. The analysis across multiple aortic arch types provides insight into the generalization capabilities of the learned policies. The consistent performance trends observed across different anatomical configurations indicate that the proposed approach is able to partially generalize to unseen geometries, addressing a key limitation of existing methods.
  3. Finally, the combined use of multiple imitation learning strategies was investigated. The results show that Behavioral Cloning leads to comparable performance with respect to PPO, while the inclusion of GAIL does not provide consistent benefits and may even degrade performance. Interestingly, their combination does not result in a clear improvement, suggesting that the interaction between imitation learning strategies does not lead to predictable or synergistic effects.
- Several factors may explain this behavior. First, the limited number of expert demonstrations (25 trajectories) may be sufficient to provide partial guidance through BC, but insufficient for more complex frameworks such as GAIL, which typically require larger datasets. Second, the demonstrations were generated manually and therefore exhibit variability, which may hinder the extraction of a consistent policy during imitation learning. Finally, the strong geometric guidance provided by the centerline-based path constrains the exploration space, enabling PPO to efficiently learn the task through interaction alone and reducing the potential benefit of imitation learning, both individually and in combination.

## Limitation of The Study

Despite the promising results, several limitations should be acknowledged:

- **Absence of blood flow simulation.** The environment does not include hemodynamic effects or fluid-structure interactions, which may influence guidewire dynamics in real procedures.
- **Rigid vessel walls.** Vascular walls are modeled as rigid, neglecting arterial compliance and deformation under device interaction.
- **Simplified guidewire model.** The guidewire is represented using a simplified mechanical model that does not fully capture torsion, distributed stiffness, and friction effects.
- **Simplified procedural strategy.** The navigation assumes direct advancement toward the target, without modeling more complex clinical maneuvers such as push-pull strategies or energy-based reconfiguration.

## 6. Conclusions

This work presented a simulation framework for autonomous endovascular navigation, integrating physics-based modeling, control, and learning within a unified Unity environment. The system includes a guidewire representation, a catheter actuation model derived from a real three-instrument set-up, and a modular architecture that supports both reinforcement learning and externally defined control strategies.

The framework enables the full navigation pipeline, from anatomical modeling and path planning to real-time control and policy evaluation. In particular, reinforcement learning approaches based on PPO, SAC and their variants were investigated and evaluated across multiple aortic arch configurations. The results demonstrate the feasibility of learning effective navigation strategies in complex vascular environments, highlighting the robustness of PPO-based methods and the challenges associated with alternative learning approaches.

Overall, this work provides a flexible and scalable platform for the development and evaluation of autonomous navigation strategies, representing a step toward the integration of learning-based methods in endovascular robotics.

## Future Developments

Future work will focus on improving both the realism of the simulation environment and the applicability of the proposed framework to real-world scenarios. Several extensions can enhance the physical and procedural fidelity of the environment. These include the incorporation of hemodynamic effects and fluid-structure interactions, which would allow a more accurate representation of the forces acting on the guidewire. In addition, modeling deformable vessel walls would enable the simulation to account for arterial compliance and its influence on device behavior. The guidewire model could also be further refined by including more advanced mechanical properties, such as torsion, distributed stiffness, and frictional interactions along the vessel walls. From a procedural perspective, future developments will consider more advanced manipulation strategies, such as push-pull maneuvers and reorientation techniques, allowing the agent to better exploit the intrinsic mechanical properties of the guidewire.

Another key direction is the transition from simulation to real-world applications. Since the actuation model is derived from a real three-instrument system available in the laboratory, the proposed framework provides a

solid foundation for experimental validation. Developments will focus on transferring the learned policies to physical setups and evaluating their performance under controlled laboratory conditions. A central challenge in this process is the sim-to-real gap, which arises from discrepancies between simulated and real environments. These include modeling inaccuracies, unmodeled physical phenomena, sensor noise, latency in control signals, and uncertainties in actuation. The integration of real sensor feedback and hybrid control strategies, combining learning-based policies with classical control methods, may further improve stability and safety during deployment.

In this context, simulation remains a crucial component, as it allows large-scale training in a safe and controlled environment, significantly reducing the risks and costs associated with real-world experimentation. Bridging the sim-to-real gap will be a critical step toward the development of reliable and clinically applicable autonomous endovascular navigation systems.

## References

- [1] F. Jozsa and S. Munakomi. Anatomy, head and neck: Carotid arteries. In *StatPearls [Internet]*. StatPearls Publishing, Treasure Island (FL), 2025. URL <https://www.ncbi.nlm.nih.gov/books/NBK545238/>. Updated 2025 Dec 9.
- [2] A. Ismail, S. Ravipati, D. Gonzalez-Hernandez, H. Mahmood, A. Imran, E. J. Munoz, S. Naeem, Z. U. Abdin, and H. F. Siddiqui. Carotid artery stenosis: A look into the diagnostic and management strategies, and related complications. *Cureus*, 15(5):e38794, May 2023. doi: 10.7759/cureus.38794.
- [3] Medmovie. Carotid artery anatomy. [https://medmovie.com/library\\_id/3101/topic/cvml\\_0276i/summary/](https://medmovie.com/library_id/3101/topic/cvml_0276i/summary/), n.d. Accessed: 2026-02-18.
- [4] S. C. Bir and R. E. Kelley. Carotid atherosclerotic disease: A systematic review of pathogenesis and management. *Brain Circulation*, 8(3):127–136, September 2022. doi: 10.4103/bc.bc\_36\_22.
- [5] I. A. Sebastian, K. Barakhanov, and A. Ganesh. Symptomatic cervical carotid artery stenosis: Evolving paradigms in risk stratification and intervention. *Annals of Indian Academy of Neurology*, 28(1):1–9, January 2025. doi: 10.4103/aian.aian\_838\_24.
- [6] Y. Morishima, M. Kawabori, Y. M. Ito, M. Gekka, K. Furukawa, Y. Niiya, and M. Fujimura. Validity of e-pass score for evaluating perioperative minor complications associated with carotid endarterectomy. *Neurologia Medico-Chirurgica (Tokyo)*, 65(1):9–14, January 2025. doi: 10.2176/jns-nmc.2024-0035.
- [7] L. Guo, J. Zhang, K. Lv, X. Li, M. Guo, and C. Li. Multimodal monitoring of cerebral perfusion in carotid endarterectomy patients: a computational fluid dynamics study. *Frontiers in Neurology*, 15:1455401, December 2024. doi: 10.3389/fneur.2024.1455401.
- [8] S. Dharmakidari, P. Bhattacharya, and S. Chaturvedi. Carotid artery stenosis: Medical therapy, surgery, and stenting. *Current Neurology and Neuroscience Reports*, 17(10):77, August 2017. doi: 10.1007/s11910-017-0786-2.
- [9] S. Munakomi and P. Theetha Kariyanna. Carotid artery surgery. In *StatPearls [Internet]*. StatPearls Publishing, Treasure Island (FL), 2025. URL <https://www.ncbi.nlm.nih.gov/books/NBK441947/>. Updated 2025 May 2.
- [10] A. Kimyaghalam and D. T. Baril. Carotid artery stenting. In: *StatPearls [Internet]*, Nov 2025. [Updated 2025 Nov 7]. Treasure Island (FL): StatPearls Publishing; 2025 Jan-. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK470541/>.
- [11] J. D. Kelley and J. V. Ashurst. Anatomy, thorax, aortic arch. In: *StatPearls [Internet]*, Aug 2023. [Updated 2023 Aug 8]. Treasure Island (FL): StatPearls Publishing; 2025 Jan-. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK499911/>.
- [12] S. Madhwal, V. Rajagopal, D. L. Bhatt, C. T. Bajzer, P. Whitlow, and S. R. Kapadia. Predictors of difficult carotid stenting as determined by aortic arch angiography. *Journal of Invasive Cardiology*, 20(5): 200–204, May 2008.
- [13] D. Kotelis, M. S. Bischoff, B. Jobst, H. von Tengg-Kobligk, U. Hinz, P. Geisbüsch, and D. Böckler. Morphological risk factors of stroke during thoracic endovascular aortic repair. *Langenbeck's Archives of Surgery*, 397(8):1267–1273, 2012. doi: 10.1007/s00423-012-0997-6.

- [14] M. M. Marrocco-Trischitta and M. Glauber. Implications of different definitions for aortic arch classification provided by contemporary guidelines on thoracic aortic repair. *Interactive Cardiovascular and Thoracic Surgery*, 32(6):950–952, 2021. doi: 10.1093/icvts/ivab029.
- [15] Ameya Pore, Zhen Li, Diego Dall’Alba, Albert Hernansanz, Elena De Momi, Arianna Menciassi, Alicia Casals Gelpí, Jenny Dankelman, Paolo Fiorini, and Emmanuel Vander Poorten. Autonomous navigation for robot-assisted intraluminal and endovascular procedures: A systematic review. *IEEE Transactions on Robotics*, 39(4):2529–2548, 2023. doi: 10.1109/TRO.2023.3269384.
- [16] Lucinda Cruddas, Guy Martin, and Celia Riga. Robotic endovascular surgery: current and future practice. *Seminars in Vascular Surgery*, 34(4):233–240, 2021. ISSN 0895-7967. doi: <https://doi.org/10.1053/j.semvascsurg.2021.10.002>. URL <https://www.sciencedirect.com/science/article/pii/S0895796721000594>.
- [17] Harry Robertshaw, Lennart Karstensen, Benjamin Jackson, Hadi Sadati, Kawal Rhode, Sebastien Ourselin, Alejandro Granados, and Thomas C. Booth. Artificial intelligence in the autonomous navigation of endovascular interventions: a systematic review. *Frontiers in Human Neuroscience*, Volume 17 - 2023, 2023. ISSN 1662-5161. doi: 10.3389/fnhum.2023.1239374. URL <https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2023.1239374>.
- [18] Klaus A Hausegger, Peter Schedlbauer, Hannes A. Deutschmann, and Kurt Tiesenhausen. Complications in endoluminal repair of abdominal aortic aneurysms. *European Journal of Radiology*, 39(1):22–33, 2001. ISSN 0720-048X. doi: [https://doi.org/10.1016/S0720-048X\(01\)00339-4](https://doi.org/10.1016/S0720-048X(01)00339-4). URL <https://www.sciencedirect.com/science/article/pii/S0720048X01003394>.
- [19] Michael R. Rudnick, Stanley Goldfarb, Lewis Wexler, Philip A. Ludbrook, Mary J. Murphy, Elkan F. Halpern, James A. Hill, Michael Winniford, Martin B. Cohen, Douglas B. VanFossen, and for the Iohecol Cooperative Study. Nephrotoxicity of ionic and nonionic contrast media in 1196 patients: A randomized trial. *Kidney International*, 47(1):254–261, 1995. ISSN 0085-2538. doi: <https://doi.org/10.1038/ki.1995.32>. URL <https://www.sciencedirect.com/science/article/pii/S0085253815587899>.
- [20] Pei Ho, Stephen W.K. Cheng, P.M. Wu, Albert C.W. Ting, Jensen T.C. Poon, Clement K.M. Cheng, Joseph H.M. Mok, and M.S. Tsang. Ionizing radiation absorption of vascular surgeons during endovascular procedures. *Journal of Vascular Surgery*, 46(3):455–459, 2007. ISSN 0741-5214. doi: <https://doi.org/10.1016/j.jvs.2007.04.034>. URL <https://www.sciencedirect.com/science/article/pii/S0741521407006374>.
- [21] Mohammad Mofatteh. Neurosurgery and artificial intelligence. *AIMS Neuroscience*, 8(4):477–495, 2021. ISSN 2373-7972. doi: 10.3934/Neuroscience.2021025. URL <https://www.aimspress.com/article/doi/10.3934/Neuroscience.2021025>.
- [22] Priyanka Rao, Quentin Peyron, Sven Lilge, and Jessica Burgner-Kahrs. How to model tendon-driven continuum robots and benchmark modelling performance. *Frontiers in Robotics and AI*, 7, 02 2021. doi: 10.3389/frobt.2020.630245.
- [23] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *Robotics Automation Magazine, IEEE*, 13:82 – 90, 01 2007. doi: 10.1109/MRA.2006.250573.
- [24] Lydia Kavraki and Steven LaValle. Motion planning. pages 139–162, 01 2016. doi: 10.1007/978-3-319-32552-1\_7.
- [25] Tammy Jiang, Jaimie L. Gradus, and Anthony J. Rosellini. Supervised machine learning: A brief primer. *Behavior Therapy*, 51(5):675–687, 2020. ISSN 0005-7894. doi: <https://doi.org/10.1016/j.beth.2020.05.002>. URL <https://www.sciencedirect.com/science/article/pii/S0005789420300678>.
- [26] Fanxu Meng, Shuxiang Guo, Wei Zhou, and Zhengyang Chen. Evaluation of a reinforcement learning algorithm for vascular intervention surgery. In *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1033–1037, 2021. doi: 10.1109/ICMA52036.2021.9512675.
- [27] Tobias Behr, Tim Pusch, Marius Siegfarth, Dominik Hüsener, Tobias Mörschel, and Lennart Karstensen. Deep reinforcement learning for the navigation of neurovascular catheters. *Current Directions in Biomedical Engineering*, 5:5–8, 09 2019. doi: 10.1515/cdbme-2019-0002.
- [28] H. You, E. Bae, Y. Moon, et al. Automatic control of cardiac ablation catheter with deep reinforcement learning method. *J Mech Sci Technol*, 33:5415–5423, 2019.

- [29] Wenqiang Chi, Jindong Liu, Mohamed E. M. K. Abdelaziz, Giulio Dagnino, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Trajectory optimization of robot-assisted endovascular catheterization with reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3875–3881, 2018. doi: 10.1109/IROS.2018.8593421.
- [30] Jihoon Kweon, Kyunghwan Kim, Chaehyuk Lee, Hwi Kwon, Jinwoo Park, Kyoseok Song, Young In Kim, Jeeone Park, Inwook Back, Jae-Hyung Roh, Youngjin Moon, Jaesoon Choi, and Young-Hak Kim. Deep reinforcement learning for guidewire navigation in coronary artery phantom. *IEEE Access*, 9:166409–166422, 2021. doi: 10.1109/ACCESS.2021.3135277.
- [31] Wenqiang Chi, Giulio Dagnino, Trevor M. Y. Kwok, Anh Nguyen, Dennis Kundrat, Mohamed E. M. K. Abdelaziz, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2414–2420, 2020. doi: 10.1109/ICRA40945.2020.9196912.
- [32] Lennart Karstensen, Jacqueline Ritter, Johannes Hatzl, Torben Pätz, Jens Langejürgen, Christian Uhl, and Franziska Mathis-Ullrich. 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(11):2033–2040, November 2022. ISSN 1861-6410. doi: 10.1007/s11548-022-02646-8. URL <https://europepmc.org/articles/PMC9515141>.
- [33] Yongjun Cho, Jae-Hyeon Park, Jaesoon Choi, and Dong Eui Chang. Sim-to-real transfer of image-based autonomous guidewire navigation trained by deep deterministic policy gradient with behavior cloning for fast learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3468–3475, 2022. doi: 10.1109/IROS47612.2022.9982168.
- [34] Fanxu Meng, Shuxiang Guo, Wei Zhou, and Zhengyang Chen. 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, 2022. doi: 10.1109/ICMA54519.2022.9856107.
- [35] Yan Zhao, Shuxiang Guo, Yuxin Wang, Jinxin Cui, Youchun Ma, Yuwen Zeng, Xinke Liu, Yuhua Jiang, Youxinag Li, Liwei Shi, and Nan Xiao. A cnn-based prototype method of unstructured surgical state perception and navigation for an endovascular surgery robot. *Medical Biological Engineering Computing*, 57, 06 2019. doi: 10.1007/s11517-019-02002-0.
- [36] Shuang Wang, Zheng Liu, Xiongpeng Shu, Yongfeng Cao, Ling Zhang, and Le Xie. 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, 2022. doi: 10.1109/ROBIO55434.2022.10011829.
- [37] Hedyeh Rafi-Tari, Jindong Liu, Su-Lin Lee, Colin Bicknell, and Guang-Zhong Yang. Learning-based modeling of endovascular navigation for collaborative robotic catheterization. In Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 369–377, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40763-5.
- [38] Omar Al-Ahmad, Mouloud Ourak, Johan Vlekken, and Emmanuel Vander Poorten. Force control with a novel robotic catheterization system based on braided sleeve grippers. *IEEE Transactions on Medical Robotics and Bionics*, 5(3):602–613, 2023. doi: 10.1109/TMRB.2023.3291026.
- [39] Yucheng Li, David H. Myszka, and Andrew Murray. The kinematics of constant curvature continuum robots through three segments. *IEEE Robotics and Automation Letters*, 8(11):7631–7638, 2023. doi: 10.1109/LRA.2023.3320946.
- [40] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1): 269–271, 1959.
- [41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL <http://arxiv.org/abs/1707.06347>.
- [42] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018. URL <http://arxiv.org/abs/1812.05905>.

- [43] Sylvain Calinon. *Learning from Demonstration (Programming by Demonstration)*, pages 1–8. 05 2018. ISBN 978-3-642-41610-1. doi: 10.1007/978-3-642-41610-1\_27-1.
  - [44] Andrew Wagenmaker, Perry Dong, Raymond Tsao, Chelsea Finn, and Sergey Levine. Posterior behavioral cloning: Pretraining bc policies for efficient rl finetuning, 2025. URL <https://arxiv.org/abs/2512.16911>.
  - [45] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *CoRR*, abs/1606.03476, 2016. URL <http://arxiv.org/abs/1606.03476>.
  - [46] CU ModeRn Unity Anatomical PRogram. Aortic arch. <https://sketchfab.com/3d-models/aortic-arch-f21267a127284ea4995424836e11ae27>, n.d. Sketchfab. Licensed under Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA).
  - [47] l8aart. Thoracic aorta. <https://www.thingiverse.com/thing:488215>, n.d. Thingiverse. Accessed: 2026-04-09.
  - [48] Soma3D. Aortic arch normal development. <https://sketchfab.com/3d-models/aortic-arch-normal-development-460eb397b0cc4713a48bc3a138bad77b>, n.d. Sketchfab. Licensed under Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA).
- of supplementary material.

## Abstract in lingua italiana

Le procedure endovascolari rappresentano un approccio minimamente invasivo ampiamente adottato per il trattamento delle malattie cardiovascolari, come la stenosi dell'arteria carotide (CAS). Questi interventi si basano sulla navigazione di strumenti flessibili attraverso anatomie vascolari complesse per raggiungere specifici siti target. Nonostante i loro vantaggi clinici, presentano ancora diverse criticità, tra cui il rischio di complicazioni come la perforazione dei vasi, nonché una prolungata esposizione del team chirurgico a radiazioni ionizzanti. Inoltre, gli esiti procedurali dipendono fortemente dall'esperienza dell'operatore.

In particolare, la variabilità anatomica, come le diverse configurazioni dell'arco aortico, influisce significativamente sulla difficoltà della cateterizzazione e sul tasso di successo della procedura. Queste limitazioni hanno motivato lo sviluppo di soluzioni robotiche e autonome volte a migliorare la precisione, ridurre la dipendenza dall'operatore e aumentare la sicurezza della procedura.

In questo contesto, la presente tesi presenta lo sviluppo di un framework di simulazione per la navigazione endovascolare autonoma basata su intelligenza artificiale. Il sistema è implementato in Unity e riproduce modelli realistici dell'arco aortico, consentendo la simulazione della navigazione della guida (guidewire) in diverse condizioni anatomiche. La guida è modellata come una struttura flessibile composta da elementi interconnessi, permettendo l'interazione con le pareti dei vasi.

Un approccio di reinforcement learning basato su Proximal Policy Optimization (PPO) e Soft Actor-Critic (SAC) è adottato per addestrare un agente a navigare autonomamente la guida dall'aorta discendente verso i rami target dell'arco aortico, simulando scenari clinici come lo stenting carotideo. Inoltre, il framework include un modulo dedicato alla simulazione dell'attuazione, utilizzato per analizzare il sistema di controllo indipendentemente dall'ambiente anatomico. I risultati dimostrano la fattibilità dell'apprendimento di politiche di navigazione efficaci all'interno di un ambiente simulato, evidenziando il potenziale degli approcci basati su intelligenza artificiale per la cateterizzazione robotica. Questo lavoro rappresenta un primo passo all'interno di un più ampio progetto di ricerca condotto presso il gruppo di Robot-Assisted Surgery (RAS) della KU Leuven.

**Parole chiave:** Navigazione Endovascolare, Reinforcement Learning, Arco Aortico, Simulazione

## Acknowledgements

I would like to express my sincere gratitude to Professor Elena De Momi for giving me the opportunity to undertake this research experience at KU Leuven. Her support and trust have made this valuable academic and personal journey possible.

I would also like to thank Professor Emmanuel Vander Poorten for welcoming me into the RAS Group and for his guidance throughout this research project.

A special thanks goes to my tutor, Mouloud Ourak, for his constant support, patience, and invaluable guidance during every stage of this project. His technical expertise, constructive feedback, positive approach and willingness to share knowledge were fundamental to the development of this research and greatly contributed to my academic growth.

I am also grateful to all the members of the RAS Group for making my time at KU Leuven such a rewarding experience. Their kindness and enthusiasm for created a welcoming and intellectually stimulating environment that greatly contributed to both my personal and professional growth.

I would also like to thank my Politecnico di Milano tutor, Angela Peloso, for her support and valuable guidance throughout this experience.