CathSim: An Open-Source Simulator for Endovascular Intervention

Tudor Jianu®, Graduate Student Member, IEEE, Baoru Huang®, Member, IEEE, Minh Nhat Vu®, Mohamed E. M. K. Abdelaziz®, Sebastiano Fichera®, Chun-Yi Lee®, Member, IEEE, Pierre Berthet-Rayne, Ferdinando Rodriguez y Baena®, Member, IEEE, and Anh Nguyen®

Abstract—Autonomous robots in endovascular operations have the potential to navigate circulatory systems safely and reliably while decreasing the susceptibility to human errors. However, there are numerous challenges involved with the process of training such robots, such as long training duration and safety issues arising from the interaction between the catheter and the aorta. Recently, endovascular simulators have been employed for medical training but generally do not conform to autonomous catheterization due to the lack of standardization and RL framework compliance. Furthermore, most current simulators are closed-source, which hinders the collaborative development of safe and reliable autonomous systems through shared learning and community-driven enhancements. In this work, we introduce CathSim, an open-source simulation environment that accelerates the development of machine learning algorithms for autonomous endovascular navigation. We first simulate the high-fidelity catheter and aorta with a state-of-the-art endovascular robot. We then provide the capability of real-time force sensing between the catheter and aorta in simulation. Furthermore, we validate our simulator by conducting two different catheterization tasks using two popular reinforcement learning algorithms, namely SAC and PPO. The experimental results show that our open-source simulator can mimic the behavior of real-world endovascular robots and facilitate the development of different autonomous catheterization tasks. Our simulator is publicly available at https://github.com/airvlab/cathsim.

Index Terms—Medical robotics, medical simulation, reinforcement learning.

I. INTRODUCTION

Endovascular intervention has been continuously evolving since the traditional approach of direct, open-cut surgery. It involves the use of a small incision that allows surgical equipment (such as catheters and guidewires) to be maneuvered within the vasculature. This type of minimally invasive surgery (MIS) provides numerous advantages where the patient benefits from reduced blood loss, shorter recovery time, lower postoperative pain, and diminished inflammatory response compared to the traditional approaches [1]. In typical clinical conditions, the catheter and guidewire are navigated to the diagnosis zone through the use of fluoroscopy, a medical visualization procedure that obtains real-time X-Ray images from the operating theater. Despite the relative advantages, endovascular intervention still presents some drawbacks such as lack of sensory feedback, surgeon exposure to radiation, and the need for highly dexterous manipulation [2].

To reduce the continuous risk of radiation imposed on the surgeon throughout the fluoroscopic procedure, many robotic systems with leader-follower teleoperation architecture have been proposed [3], [4], [5]. The surgeon actuates the leader device, from which, the information is mapped to the follower robot that executes the related action. The use of leader-follower robots allows the surgeon to perform the procedure remotely from a safe, radiation-free zone. Recent work has further focused on creating Magnetic Resonance (MR) safe robotic platforms [3], which eliminates the ionizing radiation exposure whilst allowing the soft tissue, such as the vasculature, to be visualized [6], [7]. Furthermore, in academic settings, the robotic system is developed to provide additional information to the surgeon through assistive features such as force/torque information [8], haptic feedback [9], and real-time segmentation and tracking [10].

Whilst recent robotic platforms for endovascular intervention demonstrate their assistive potential in the successful completion of the procedure, they share two problems: i) the lack of autonomy [20] and ii) increased duration of robotic procedure compared to its non-robotic counterpart [21]. Firstly, the surgeon operates within the tridimensional space of the vasculature whilst relying on the information provided by two-dimensional fluoroscopic images and haptic feedback. Additionally, the surgeon has to avoid inflicting extensive damage to the vasculature system, thus operating under mentally strenuous conditions which can be
diminished through the automation of the procedure. A more autonomous surgery would ideally inflict little damage whilst operating in a timely manner. However, this is not a trivial task in practice as it requires a complex vision, learning, and control system that can guarantee the safety of the procedure [20].

Recent developments in machine learning promise a greater degree of autonomy in many robotic systems. Such systems leverage deep learning architectures such as convolutional neural networks [22], [23], long-short-term memory [24], [25], [26], and generative adversarial imitation learning [10] to facilitate force estimation and catheter segmentation. Whilst those systems confer a lower level autonomy through the use of robotic assistive features, the higher level autonomy is left unaddressed. However, the navigation task has been addressed through many works [12], [16], [21]. The environment employed by those works makes tradeoffs between the use of physical and virtual environments, and they generally rely on closed-source environments which cannot be replicated by fellow researchers. Furthermore, albeit the use of simulated environments, they generally do not adhere to the Reinforcement Learning (RL) paradigm.

Endovascular navigation simulations have historically faced significant challenges, impacting their advancement and utility [27]. A primary issue, as detailed in Section II, has been the lack of standardized environments. This lack is exemplified by the closed-source nature of past simulators, which showcase algorithmic advancements but are not publicly available, thereby hindering replication and broader research contributions. Computational speed, critical for these simulations, often falls short in many existing platforms, lacking the efficiency needed for real-time response and complex algorithmic processing. The integration of a gymnasium interface [28], crucial for streamlined development and testing of reinforcement learning algorithms, is frequently absent in these simulators. Additionally, ease of installation and interaction is another shortcoming in past simulators, creating barriers to widespread adoption and hindering iterative development in research settings. Addressing these challenges is key to advancing endovascular navigation simulations, crucial for training and algorithm development in endovascular procedures. Our related work in Section II further explores these challenges and introduces ‘CathSim’, our response to these gaps in the field, offering an open-source, efficient, and highly realistic simulation environment, setting a new standard in the domain.

In this work, our goal is to provide a new minimally invasive surgery environment for endovascular procedures. We aim at facilitating the development of endovascular autonomy through the provision of a standardized environment that confers familiarity to the machine learning community. As such, we propose CathSim, a real-time simulation environment for autonomous cannulation based on MuJoCo [19]. We choose MuJoCo as the base simulator as it is a real-time and accurate physics engine that facilitates optimal control applications, hence well-suited for our task. Fig. 1 shows the overview of our simulator. We summarize our contributions and the potential usage of our simulator as follows:

1) We propose CathSim, a new open-source simulation environment for endovascular procedures.
2) We implement the baseline and provide the benchmark for autonomous cannulation tasks in our simulator using two popular RL algorithms.

II. RELATED WORK

Simulation Environments. In endovascular surgery simulation, environments are categorized into four types: synthetic, animal, virtual reality, and human cadaver, each with unique benefits and limitations [29], [30], [31], [32]. These simulations focus on skill development [29], [33], assistive features like haptic feedback [9], or utilize physical materials, which are often not suitable for RL-based environments. Recent advancements include the use of high-fidelity synthetic phantoms for imitation learning [21] and the SOFA simulation environment [13] tested on bi-dimensional synthetic phantoms [34]. However, these environments are typically closed-source, where past simulators showcasing algorithmic advancements are not publicly available, hindering replication and broader research contributions.

Autonomous Catheterization. Machine learning has facilitated the shift from assistive features to semi-autonomous navigation in autonomous catheterization [35]. Our research focuses on deep reinforcement learning (RL) for its role in complex decision-making, as proven in fields like autonomous driving. Many studies utilize RL, particularly using images from fluoroscopy [9], [12], [15], [16], [36]. While alternative methods, like the Dijkstra algorithm or breadth-first search exist [18], [37], model-free RL is suitable for managing the uncertainty and complexity of achieving higher autonomy. However, most research is still in the early stages of autonomy [35], making comprehensive autonomous navigation of the vascular system a challenging yet prospective target for reinforcement learning.

In Table I, we show a detailed comparison of current learning-based works that make use of environments based on a variety of physics engines. A limitation of such environments is that they are not publicly available, which hinders the reproducibility. CathSim, in contrast to other simulators, provides an open-source environment that is well-suited for training autonomous agents using different machine-learning
approaches. Based on MuJoCo’s [19] framework, our simulator offers an advanced simulation environment for real-time applications. Furthermore, our simulator provides real-time force sensing capability and high-fidelity realistic visualization of the aorta, catheter, and endovascular robots. In practice, CathSim can be used to train RL agents or serve as a practice platform for healthcare professionals.

Recent developments in gym-based simulation frameworks have integrated the SOFA platform, exemplified by the creation of SofaGym, a general framework for SOFA environments [38], and LapGym, tailored for laparoscopic surgery [39]. Both these simulators utilize Finite Element Methods (FEM) for simulating deformable objects and a Gym API, marking significant advancements in medical simulation. However, they do not support domain randomization, an essential aspect for robust machine learning model training. In contrast, our CathSim, specifically targeting endovascular navigation, leverages the MuJoCo physics engine [19] and introduces domain randomization. This capability, coupled with our focus on highly realistic aortic models derived from post-mortem anatomies and CT scans, distinctively positions CathSim in the medical simulation landscape. Our approach not only diverges from the broader application of SofaGym as a Gym wrapper for various SOFA environments and the laparoscopic surgery focus of LapGym, but also enriches the field of medical simulation, especially in endovascular navigation.

III. THE CATHSIM SIMULATOR

Our CathSim environment has three components: i) the follower robotic model for endovascular procedures [7], ii) the aortic arch phantoms, and iii) the catheter. Our simulator enables real-time simulation and supports the training of state-of-the-art learning algorithms.

A. Robot Simulation

In this work, we aim at transferring CathBot [3] to simulation with the purpose of autonomous agents training. We choose CathBot as it is a state-of-the-art robot and is not bounded by a commercial license. The design of CathBot follows the popular leader-follower architecture with the leader robot uses haptic feedback generated by the catheter’s interaction with the environment through the navigation system [9] and maintains an intuitive control that replicates human motion patterns such as insertion, retraction, and rotation. The follower robot mimics the leader motion, and it is made up of two pneumatic linear motors for translation, one pneumatic rotating stepper motor, and two pneumatic J-clamps for clamping the instrument while performing translational motions.

Given the linear mapping between the leader and the follower robot in CathBot’s design [3], we focus only on simulating the follower robot for simplicity. We simulate the follower robot by constructing four modular platforms that are attached to the main rail. On two of those platforms, a pair of clamps is set to secure the guidewire in place during the translational movements, whilst on the other two, rotary catheter and guidewire platforms are attached for performing the angular motions. The parts that account for the translational movements on the main rail as well as the clamps are joined using prismatic joints. Furthermore, revolute joints are used to bind the wheels, thus providing the catheter with rotational movements. The rotational aspect of the catheter implies frictional movement. That is, when the rotational movement is actuated, the clamps lock the catheter in place and rotate it. A similar procedure is carried out when the catheter is linearly displaced. However, this friction-reliant rotation is difficult to undertake in simulation, and therefore we assume a perfect motion throughout the system and actuate the joints directly.

We simulate the follower of the CathBot robot together with the aortic arch phantoms and the catheter. We chose to model these elements using MuJoCo’s [19] physics engine, given its stability and computational speed while enabling real-time interactions.

B. Aorta Simulation

To simulate the aorta, we first scan the silicone-based, transparent, anthropomorphic phantom, of the Type-I and Type-II aortic arch model (Elastrat Sarl, Switzerland) to create high-fidelity 3D mesh models. These models are derived from postmortem vascular casts using techniques from [40], [41], [42] thus representing an accurate anatomy. The concave mesh is then decomposed into a set of nearly convex surfaces using the volumetric hierarchical approximate decomposition [43] resulting in 1, 024 convex hulls for Type-I aortic arch and 470 convex hulls for Type-II aortic arch. The difference in the number of convex hulls is given by the difference in the measure of concavity of the two meshes [44]. The convex hull is used to model the collision, as it aids the computational process and allows the use of soft contacts by the physics engine [19]. We show the aorta simulation in Fig. 2.
the interaction between the aorta and catheter is represented through point contacts within a rigid body framework. Each contact point is defined by a spatial frame in a global coordinate system, where the primary axis aligns with the contact normal, crucial for calculating normal forces, and the remaining axes define the tangent plane for frictional force computations. The contact distance, a determinant of separation, contact, or penetration states between objects, influences the contact force calculations. In this rigid body model, the dynamics are governed by $M\ddot{v} + c = \tau + J^T f$, where $M$ represents the joint-space inertia, $\dot{c}$ the acceleration, and $c$ the bias force computed using the Recursive-Newton-Euler algorithm [46]. The applied force $\tau$, encompassing various force components such as fluid dynamics and actuation forces, and the constraint Jacobian $J$, which relates joint and constraint coordinates, are integral in determining the force interactions at these contact points. The rigid body assumption streamlines the simulation by focusing on fundamental mechanical principles, as deformations and complex material behaviors are not considered, ensuring computational efficiency and fidelity [47], [48].

IV. REINFORCEMENT LEARNING FOR AUTONOMOUS CANNULATION

We consider the task of autonomous cannulation, where our system represents an episodic Partially Observable Markov Decision Process (POMDP). Our agent, represented by the catheter, interacts with an environment $E$ represented by an aortic arch type. At each time step $t$, the agent, receives an observation $s_t$, chooses an action $a_t(s_t)$, receives a reward $r_t(s_t, a_t)$ and arrives in a new state $s_{t+1}$. The episode terminates when the agent reaches the goal position within the aorta $g \in G$.

A. Observations

We consider three types of observations for the experiments, namely Internal, Image, and Sequential. In the Internal setup, we fed the system extensive data such as position, velocity, center of mass inertia and velocity, actuator-generated force, and external forces on the body. For the Image observation, we utilized a virtual RGB camera positioned above the aortic phantom, producing $128 \times 128$ resolution grayscale images akin to X-ray images used in clinical procedures. Despite the possibility of higher resolution, our experiments indicated no performance improvement, only increased computational demand and memory requirements.

Moreover, due to the difficulty of inferring the catheter actions given one image as in the “Image” observation setup, we consider the “Sequential” observation by inserting the temporal dimension through the concatenation of three subsequent images $\{s_{t-2}, s_{t-1}, s_t\}$. This observation would take into account the temporal domain of the catheter action and potentially provide more information to the RL agent.

B. Actions and Rewards

The actions are represented by a vector $a_t \in \mathbb{R}^{21}$, where the first 20 elements are associated with motors that actuate the revolute joints inside the tip of the catheter plus the prismatic
joint responsible for the translational movement. Furthermore, the actions are normalized within an $[-1, 1]$ interval, giving a space of $a_t \in [-1, 1]$\textsuperscript{21}.

Given the sparsity of the reward function, we chose to convey more spatial information through reward shaping. Considering the navigation task, we provide the agent with an informational reward regarding the distance towards the target by computing the negative Euclidean distance between the head of the catheter $h$ and the goal $g$, such as $d(h, g) = ||h - g||$. If the agent is within a distance $\delta$ of the goal, the episode terminates and the agent receives a further reward of $r = 10$. The distance of $\delta = 8$ mm was selected as within this distance, the catheter tip is fully inserted within the artery.

$$r(h_t, g) = \begin{cases} 10 & \text{if } d(h_t, g) \leq \delta \\ -d(h_t, g) & \text{otherwise} \end{cases}$$

(1)

Whilst this reward function assists in agent convergence, it is also prone to local minima. An example would be the erroneous insertion of the catheter in another artery. In this case, the catheter would have to increase the distance from the target to achieve the goal objective.

C. Network Architectures

Considering the continuous action representation, we employ two state-of-the-art RL algorithms, namely PPO [49] and SAC [50] with the parameters utilized in [51]. A Multi-Layered Perceptron (MLP) based policy is used for the Internal observation and a Convolutional Neural Network (CNN) is used for the Image and Sequential observation. Note that, for simplicity, we follow [52] and use CNN for both Image and Sequential observations as they only have one and four input channels respectively where each channel represents a grayscale image. In our implementation, the MLP has two hidden layers of sizes $64$ with a tanh activation function. The CNN has three convolutional layers with a ReLU activation function. The ADAM optimizer is used to train all networks with a learning rate of 0.0003.

V. EXPERIMENTS

In this section, we perform intensive experiments to validate our CathSim. We start with the experiment to verify whether our simulator can mimic the behavior of the real-robot (CathBot). We then demonstrate how CathSim can be used for autonomous cannulation tasks using RL algorithms.

**Training Details.** The experiments were conducted on an NVIDIA RTX 2060 GPU (33MHz) system on an Ubuntu 22.04 LTS based operating system. Furthermore, the system contained an AMD Ryzen 7 5800X 8-Core Processor with a total of 16 threads with 64GB of RAM. All experiments used PyTorch, for the RL implementations we used stable baselines [53]. Each episode has two terminal states, one which is time-bound (i.e., termination of an episode upon reaching a number of steps) and one which is goal bound (i.e., the agent achieves the goal $g$).

A. Simulator Validation

1) **Setup:** We assess the validity of our simulator, considering the distribution of forces generated throughout cannulation of the brachiocephalic artery (BCA). Ideally, we want to compare the force measured by our simulator and the force measured in the real-world experiment setup [3]. Note that, in the experiments conducted in [3], a load cell (Mini40, ATI Industrial Automation, Apex, NC, USA) was used to capture the force generated by the interaction of the instruments with the silicon phantom.

To extract the force from our simulator, we manually perform the cannulation using the keyboard, then in each simulation time step, we extract the collision points between the catheter and the aorta along with the tridimensional force expressed as the normal force $f_n$ and frictional forces $f_s$ and $f_t$.

We further compute the magnitude of the force given the previous, such as given a time instance $t$, the magnitude of the force $f_t$ is given by:

$$f_t = \sqrt{f_s^2(t) + f_t^2(t) + f_z^2(t)}$$

We proceed by offering a comparison between the observed empirical distribution and a normal distribution derived from the real experiments conducted in [3]. We sample the same number of samples generated in our simulation and the real experiment in [3] for a fair comparison. We use the distributions to generate a cumulative distribution function $F_X(x) = P(X \leq x)$, which can be visualized in Fig. 5. We further assess the normality of the sampled data, followed by a comparison of the two given distributions.

2) **Results:** We begin our comparison with the Shapiro-Wilk test of normality on the data extracted from the simulator and, given a p-value of $p \approx 7.195 \times 10^{-17}$ and a statistic of 0.878, we conclude that the sampled data does not represent a normal distribution $X(\mu, \sigma)$. This can be visualized in Fig. 5, where the sampled data is plotted against a normal distribution. Furthermore, we assess the homoscedasticity of the sample distribution and normal distribution by using Levene's tests, which results in a statistic of $40.818$ and a p-value of $p \approx 2.898 \times 10^{-10}$, therefore, concluding that $\sigma_1^2 \neq \sigma_2^2$. Given the previous statistics (i.e., non-normal distribution and unequal variances), we select the non-parametric Mann-Whitney test to compare the given distributions. The resulting statistic given the test is $76076$, with a p-value of $p \approx 0.445$. Given that the p-value is higher than the threshold of $p = 0.05$, we can conclude that the differences in the distributions are merely given to chance and therefore the distributions can be considered as being part of the same population and thus convey that the force distribution of our simulator closely represents the distribution of forces encountered in the real-life system. Therefore, we can see that our CathSim successfully mimics the behavior of the real-world system.

B. Reinforcement Learning Results

1) **Setup:** We consider the autonomous catheterization of two principal arteries into the aortic arch, namely the brachiocephalic artery (BCA) and the left common carotid artery.
Within both setups, we position the catheter tip at the starting locations within the ascending aorta and terminate the training when the catheter is fully inserted within the artery. We follow the same procedure for both the Type-I and the Type-II aortic arches. Please see Fig. 4 for our experimental setup.

We trained the model for 600,000 time steps. Each episode started with a random catheter displacement of 1 mm, followed by navigation through the ascending aorta to reach the goal. If not achieved within 2,000 steps, the episode ended. Data on the aorta’s contact points and exerted force, as per Eq. (2), was gathered at each step to form a force heatmap overlaid on the RGB virtual image. The model’s performance was evaluated over 30 episodes. The maximum and mean forces for $n$ samples were computed as:

$$ f_{\text{max}} = \max_{t \in n} f_t, \quad f_{\text{mean}} = \frac{1}{n} \sum_{t=1}^{n} f_t $$

(3)

The training time ranged from 1 hour for PPO to 8 hours for SAC, making PPO approximately five times faster. Regardless, our simulator showed a performance of 60 FPS for the image-based environment and 80 FPS for the internal-based environment.

2) Quantitative Results: Table II shows the force, reward, and success rate of all methods. Type-I Aortic Arch experiments show that PPO relying on a sequential observation space achieved the greatest reward when cannulating the BCA target, although it shows the least performance when the target is LCCA. A more coherent reward has been achieved while using a singular image observation, where the cannulation of LCCA presents the greatest success (37%). In contrast, the cannulation of BCA presents close results to the sequential observation (83%). The performance gap between the cannulation of BCA and LCCA is mainly because of the start configuration and the position of the BCA and LCCA in the aortic arch. Naturally, from Fig. 4 we can see that it would be easier for both humans and RL agents to reach BCA rather than LCCA, since LCCA’s position is further away from the navigation direction and is surrounded by other vascular branches.

![Fig. 4. The starting configuration. The figure depicts the navigation task employed in the Type-I and Type-II Aortic Arches. The catheter is initially situated within the ascending aorta with the task of navigating towards the brachiocephalic artery (BCA) or the left common carotid artery (LCCA). The task finishes when the tip of the catheter is situated within proximity of 8 mm of the targets.](image)

![Fig. 5. The forces distributions are visualized through kernel density estimation (left) as well as through the cumulative distribution function (right). The figure depicts the similarities between the two distributions, where the empirical distribution gathered from the simulator shows a heavier tail than the normal distribution. Within the right subfigure, it can be observed that the cumulative distributions intersect at the 0.5 threshold, which signifies that the two distributions have a similar median ($\tilde{x}_1 \sim \tilde{x}_2$).](image)

Given the success rates of the cannulation, the catheterization of the Type-II aortic arch appears to be a more straightforward procedure. The task implies that the dependability of the vessel wall for maneuvering is reduced, leading to a more straightforward catheterization procedure. This phenomenon can be also observed in Fig. 6, where the catheter
exerted a greater amount of force on the aortic walls in order to reach the designated target. Overall, from our experiments, PPO shows better suitability for the task, especially when the observation is image-based.

3) Qualitative Result: Force frames from each time step help us calculate the mean of in-contact regions, which are then superimposed on the phantom model to generate force heatmaps (see Fig. 6). These maps show increased catheter force at bends required for target access. For example, the catheter often overshoots the LCCA target in Fig. 6 (c) whilst exerting greater force on the ascending aortic walls within Type-I Aortic Arch. Within the Type-II Aortic Arch, more force exerted on the wall between BCA and LCCA. Regardless, the Type-I Aortic Arch, poses more difficulties.

We further display the results of the force interaction in Fig. 7. Analyzing the figure, it can be seen that the force distribution is quite similar along both the Type-I and Type-II Aortic Arch, where most of the force is concentrated within the 0.0 to 0.2N range. Despite this, there are outliers which exert a force greater than 0.4N. Specifically, the BCA within the Type-I Aortic Arch has a notably wider interquartile range than the LCCA, suggesting that the BCA generally exerts more force than the LCCA.

At each time step, we compute the mean reward of the last 100 steps and display the results in Fig. 8. The algorithms show continuous learning with slight convergence. In our experiments, PPO with an image observation obtained the highest or close to the highest reward. Furthermore, it shows that an image-based observation that correlates to the medical procedure undertaken in real clinical scenarios is capable of reaching the most reward. While using the internal observation space, SAC performed reasonably well, managing to obtain the highest reward within the Type-I aortic arch when cannulating the LCCA target. Within all cases and coherent with the evaluation undertaken in the previous section, the sequential observation space yielded the least reward.

VI. DISCUSSION

CathSim serves as a framework for training and benchmarking autonomous catheterization agents, addressing a...
need for standardized evaluation identified in earlier studies [12], [15], [36]. It aims to establish a benchmark in the field of autonomous catheterization, utilizing sim-to-real transfer methods, similar to those in autonomous driving [21], to potentially reduce risks and costs in medical training. Beyond agent training, CathSim contributes to surgical planning and medical education, offering an open-source tool for research in surgical simulation.

Furthermore, there are challenges concerned with the sim-to-real transfer. Firstly, environmental variability, such as differences in blood flow and vessel wall properties in actual human vascular systems, presents a significant challenge. This variability can impact algorithm performance, necessitating adaptations to accommodate diverse physiological conditions. Secondly, the accuracy of X-ray imaging in real-world scenarios, a crucial aspect for catheter navigation, differs from simulated environments. This disparity can affect the algorithm’s effectiveness, highlighting the need for enhanced imaging simulation fidelity. Additionally, the real-time dynamic responses of the vascular system to catheter manipulation are complex and require algorithms to adapt in real-time, a feature that may be oversimplified in simulations. Importantly, our current simulation lacks deformable anatomical models, presenting a limitation in interaction accuracy. To address these challenges, further domain adaptation is crucial, ensuring that the algorithms are robust, accurate, and adaptable to the intricacies of real-world medical settings.

VII. CONCLUSION

We present CathSim, an open-source simulation environment designed as a comprehensive benchmarking platform for autonomous endovascular navigation. Our proposed simulator allows for the development and testing of a diverse range of algorithms for autonomous cannulation, eliminating the need for physical robotic systems and thereby reducing the associated risks and costs of experimental hardware. It serves a dual purpose as a development and benchmarking platform for computer scientists and roboticists, as well as a training platform for healthcare professionals. Furthermore, with the rapid advancements in medical robotics, CathSim provides a risk-free, controlled environment where innovative procedures and techniques can be tested and refined before their implementation in real-world clinical scenarios. By maintaining an open-source model, our simulator also encourages collaborative advancements and knowledge sharing within the scientific community.

REFERENCES

[1] I. Wamala, E. T. Roche, and F. A. Pigula, “The use of soft robotics in cardiovascular therapy,” Expert Rev. Cardiovasc. Ther., vol. 15, no. 10, pp. 767–774, 2017.
[2] O. M. Omosire, S. Han, J. Xiong, H. Li, Z. Li, and L. Wang, “A review on flexible robotic systems for minimally invasive surgery,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 1, pp. 631–644, Jan. 2022.
[3] D. Kundrat et al., “An MR-safe endovascular robotic platform: Design, control, and ex-vivo evaluation,” IEEE Trans. Biomed. Eng., vol. 68, no. 10, pp. 3110–3121, Oct. 2021.
[4] V. M. Pereira et al., “First-in-human, robotic-assisted neuroendovascular intervention,” J. Neurosurg. Surg., vol. 12, no. 4, pp. 338–340, 2020.
[5] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, “Continuum robots for medical applications: A survey,” IEEE Trans. Robot., vol. 31, no. 6, pp. 1261–1280, Dec. 2015.
[6] T. Heit et al., “Real-time magnetic resonance imaging–guided coronary intervention in a porcine model,” Sci. Rep., vol. 9, no. 1, p. 8663, 2019.
[7] M. E. M. K. Abdelaziz et al., “Toward a versatile robotic platform for fluoroscopy and MRI-guided endovascular interventions: A pre-clinical study,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 5411–5418.
[8] J. Konstantinova, A. Jiang, K. Althoefer, P. Dasgupta, and T. Nanayakkara, “Implementation of tactile sensing for palpation in robot-assisted minimally invasive surgery: A review,” IEEE Sensors J., vol. 14, no. 8, pp. 2490–2501, Aug. 2014.
[9] M. B. Molinero et al., “Haptic guidance for robot-assisted endovascular procedures: Implementation and evaluation on surgical simulator,” in Proc. IROS, 2019, pp. 5398–5403.
[10] A. Nguyen et al., “End-to-end real-time catheter segmentation with optical flow-guided warping during endovascular intervention,” in Proc. ICRAB, 2020, pp. 9967–9973.
[11] A. Juliani et al., “Unity: A general platform for intelligent agents,” 2018, arXiv:1809.02627.
[12] L. Karstensen, T. Behr, T. P. Pusch, F. Mathis-Ullrich, and J. Stallkamp, “Autonomous guidewire navigation in a two dimensional vascular phantom,” Curr. Dir. Biomed. Eng., vol. 6, no. 1, 2020, Art. no. 20200007.
[13] F. Faure et al., “SOFA: A multi-model framework for interactive physical simulation,” in Soft Tissue Biomechanical Modeling for Computer Assisted Surgery, pp. 283–321, 2012.
[14] R. Davis, R. Henschel, and G. Warburton, “A Timoshenko beam element,” J. Sound Vib., vol. 22, no. 4, pp. 475–487, 1972.
[15] T. Behr, T. P. Pusch, M. Siegfarth, D. Hüsener, T. Mörschel, and L. Karstensen, “Deep reinforcement learning for the navigation of neurovascular catheters,” Current Dir. Biomed. Eng., vol. 5, no. 1, pp. 5–8, 2019.
[16] O. M. Omosire, T. Akinzemi, W. Duan, W. Du, and L. Wang, “A novel sample-efficient deep reinforcement learning with episodic policy transfer for PID-based control in cardiac catheterization robots,” 2021, arXiv:2110.14941.
[17] E. Rohmer, S. P. Singh, and M. Freese, “V-REP: A versatile and scalable robot simulation framework,” in Proc. IROS, 2013, pp. 1321–1326.
[18] P. Scheggi et al., “Automated planning for robotic guidewire navigation in the coronary arteries,” in Proc. IEEE 5th Int. Conf. Soft Robot. (RoboSoft), 2022, pp. 239–246.
[19] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in Proc. IROS 2012, pp. 5026–5033.
[20] A. Marban, V. Srinivasan, W. Samek, J. Fernández, and A. Casals, “A neuro-recurrent-vision approach,” IEEE Trans. Haptics, vol. 10, no. 3, pp. 431–443, Jul.–Sep. 2017.
[21] A. Marban, V. Srinivasan, W. Samek, J. Fernández, and A. Casals, “Towards retrieving force feedback in robotic-assisted surgery: A supervised neuro-recurrent-attention approach,” IEEE Trans. Haptics, vol. 10, no. 3, pp. 431–443, Jul.–Sep. 2017.
[22] A. Marban, V. Srinivasan, W. Samek, J. Fernández, and A. Casals, “Towards retrieving force feedback in robotic-assisted surgery: A supervised neuro-recurrent-attention approach,” IEEE Trans. Haptics, vol. 10, no. 3, pp. 431–443, Jul.–Sep. 2017.
[23] A. I. Aviles, S. M. Alsaleh, E. Montseny, P. Sobrevilla, and A. Casals, “Towards meaningful robotic force feedback in robotic-assisted surgery,” IEEE Trans. Biomed. Eng., vol. 67, no. 5, pp. 1261–1280, May 2021.
[24] W. Chi et al., “Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2020, pp. 2414–2420.
[25] A. Marban, V. Srinivasan, W. Samek, J. Fernández, and A. Casals, “A recurrent convolutional neural network architecture for multi-sensor force estimation in robotic surgery,” Biomed. Signal Process. Control, vol. 50, pp. 134–150, Apr. 2019.
[26] C. Gao, X. Liu, M. Peven, M. Unberath, and A. Reiter, “Learning to see forces: Surgical force prediction with RGB-point cloud temporal convolutional networks,” in Proc. MICCAI Workshop, 2018, pp. 118–127.
[27] A. I. Aviles, S. M. Alsaleh, J. K. Hahn, and A. Casals, “A recurrent convolutional neural network architecture for multi-sensor force estimation in robotic surgery,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2020, pp. 2414–2420.
[28] A. Marban, V. Srinivasan, W. Samek, J. Fernández, and A. Casals, “A recurrent convolutional neural network architecture for multi-sensor force estimation in robotic surgery,” Biomed. Signal Process. Control, vol. 50, pp. 134–150, Apr. 2019.
