GESRsim: Gastrointestinal Endoscopic Surgical Robot Simulator

Huxin Gao[1,2,3†], Zedong Zhang[3†], Changsheng Li[4], Xiao Xiao[5], Liang Qiu[1,7], Xiaoxiao Yang[6], Ruoyi Hao[2], Xiuli Zuo[6], Yanqing Li[6] and Hongliang Ren[2,1,3*]

***Abstract*—Robot-assisted gastrointestinal endoscopic surgery (GES) as a kind of natural orifice transluminal endoscopic surgery (NOTES) is the next-generation minimally invasive surgery (MIS). Besides, rendering certain autonomy to a Gastrointestinal Endoscopic Surgical Robot (GESR) is promising but highly challenging. Therefore, to accelerate the development and augment the autonomy of GESR, we use CoppeliaSim to develop the *first* robotic simulator for the GESR system (GESRsim) based on our previous design. The GESRsim provides several 3D models and kinematics of our designed manipulators and endoscopic snake bone. Additionally, we build several scenes for robotic GES training and then utilize different programming interfaces to perform teleoperation. Furthermore, several advanced control algorithms, including visual servoing (VS) and deep reinforcement learning (DRL), are implemented to verify the performance of the GESRsim.**

***Index Terms*—Robotic simulation, Robotic simulator, NOTES, Gastrointestinal endoscopic surgical robot, Autonomy, Visual servoing, Reinforcement learning**

I. Introduction

Robot-assisted gastrointestinal endoscopic surgery (GES), as a kind of natural orifice transluminal endoscopic surgery (NOTES), is experiencing rapid growth after robotic systems obtained successful worldwide applications in laparoscopic surgery [1], [2] of da Vinci Surgical System (Intuitive Surgical Inc., USA). Simulation is a low-cost and effective method for surgeons to learn basic robotic skills, especially teleoperation, before performing real surgery. For robotic laparoscopy, commercial robotic simulators with high fidelity include da Vinci Trainer, Robotic Surgery Simulator, Robotix Mentor™, etc. Similarly, we expect an increasing demand for GES simulators.

On the other hand, robotic systems are transitioning from teleoperation to autonomy because of the rapid technological development in the intelligent robotic control loop (sensing-planning-execution), including imaging [3], force sensing [4], artificial intelligence (AI) [5]. Currently, the most successful autonomous robotic systems in research are mobile robots and the self-driving vehicle is explosively entering the market. However, due to more complexity of surgical tasks and strict requirements for safety, there are no surgical robots with full autonomy[1] by now [7]. An increasing number of robotic surgical systems are endeavoring to achieve conditional autonomy [8] and high autonomy [9]. For higher

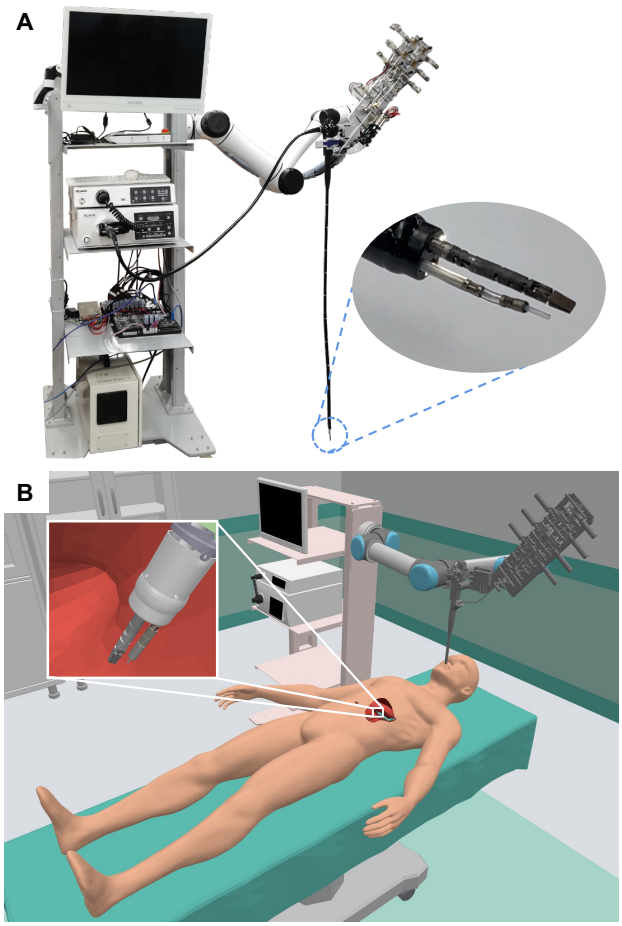

Fig. 1. GESR system overview. **A** The real robotic system. B The GESRsim. A patient is undergoing an ESD procedure. The end-effectors of the real and virtual GESR are shown in zoomed-in plots, respectively.

[1] Department of Biomedical Engineering, National University of Singapore, Singapore.
[2] Department of Electronic Engineering, The Chinese University of Hong Kong, Hongkong, China
[3] NUS (Suzhou) Research Institute, Suzhou, China
[4] School of Mechatronic Engineering, Beijing Institute of Technology, Beijing, China
[5] Department of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen, China
[6] Department of Gastroenterology, Qilu Hospital of Shandong University, Jinan, China
[7] Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore.
† Equal contribution
* Corresponding author: Hongliang Ren hlren@ieee.org, Department of Electronic Engineering, The Chinese University of Hong Kong

[1] The levels of autonomy for the surgical robot are defined in [6].

levels of autonomy, robotic simulation provides a low-risk solution to verify the perception, decision and action algorithms. Therefore, robotic simulator plays a significant role in the autonomy of surgical robots. Nevertheless, proprietary commercial simulators hinder global researchers from contributing to further developments.

Open-source robotic simulation platforms for surgical robots are emerging to solve this problem. For instance, after the open-source da Vinci Research Kit (dVRK) [10] was released, several open-source da Vinci simulators were developed. The Robot Operating System (ROS)-based dVRK provided a Gazebo [11] simulator. [12] used the Virtual Robot Experimentation Platform (V-REP) [13] to develop a whole da Vinci system, including the da Vinci manipulators and setup joints, and to provide some surgical training scenes. Following that, a reinforcement learning (RL) environment was established for da Vinci system called dVRL [14] to perform autonomous learning. Likewise, SurRoL [15] was also developed. Besides robotic laparoscopy, several robotic capsule simulators were also innovated. In [16], closed-loop control for magnetic capsule endoscopes was implemented in Gazebo. [17] utilized Unity [18] to develop an autonomous capsule robot and combined data-driven algorithms to conduct a perception-decision-action loop. In terms of neurosurgery, [19] developed a neurosurgical robot in V-REP for the burr-hole selection. However, there are no open-source simulators for the Gastrointestinal Endoscopic Surgical Robot (GESR) system.

A versatile and user-friendly robotic simulator is indispensable for developing the GESR. Our group previously developed a GESR system as shown in Fig. 1**A**. Based on the real robotic system, an open source-based GESR simulator (available upon request), namely GESRsim as shown in Fig. 1**B**, is developed using CoppeliaSim (previously V-REP) in this work. The main reason for choosing CoppeliaSim is its cross-platform compatibility (Linux, Mac and Windows) and high customizability through an application programming interface (API) with different languages (C/C++, Python, Java, Lua, Matlab or Octave). Therefore, the developed GESRsim is user-friendly and has strong compatibility with AI frameworks based on different languages, such as python-based PyTorch and Matlab AI toolkit.

The rest is organized as follows: Section II describes the kinematics of the whole GESRsim system, including flexible manipulator, flexible wrist, and endoscopic snake bone. Section III shows the software architecture. Section IV demonstrates the robotic GES training. Section V discusses autonomous control, including visual servoing and RL control. Section VI concludes the GESRsim system and future work.

II. The Kinematics of the GESRsim

The GESRsim mainly consists of a flexible manipulator (FM) [20], a flexible wrist (FW) [21] and an endoscope (Smart GS-60DQ, HUACO, China). The FM and the FW can pass through two endoscopic working channels with diameters of 3.8mm and 2.8mm, respectively, to perform gastrointestinal surgery, typically endoscopic submucosal dissection (ESD). The endoscope tip, a snake-bone structure (namely ESB), can be controlled to adjust the camera orientation. The FM end-effector can be integrated with different surgical tools for corresponding surgical tasks (e.g., grasping, suturing, etc.). In contrast, the FW end-effector is a high-frequency electrotome for tissue cutting.

A. Flexible Manipulator Kinematics

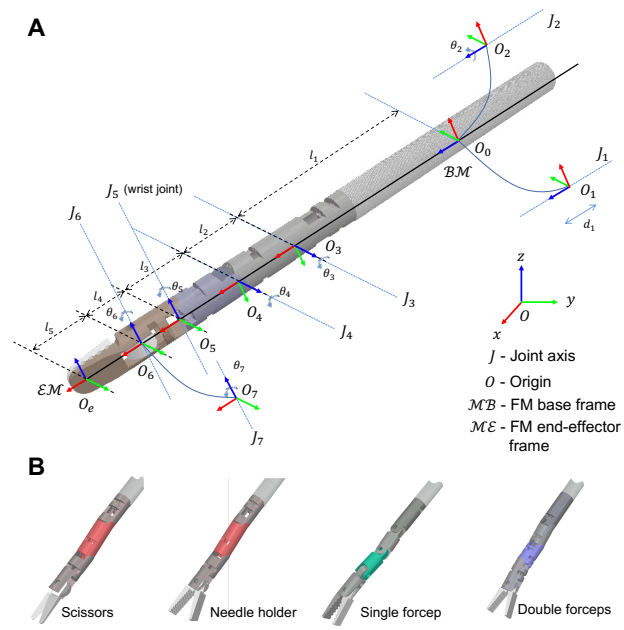

Fig. 2. Flexible manipulator. **A** FM kinematic illustration. **B** Four kinds of FMs with different surgical tools. From left to right: scissor; needle holder; single controllable forcep; double controllable forceps. Different colors are rendered on the middle part of the FMs to distinguish structure clearly. The same color means the same structure.

TABLE I
DH parameters of FMs

link	joint	a_{i-1}	α_{i-1}	d_i	θ_i
1	P	0	0	d_1	–
2	R	0	0	–	θ_2
3	R	l_1	$\pi/2$	–	$\pi/2+\theta_3$
4	R	l_2	0	–	θ_4
5	R	l_3	$\pi/2$	–	θ_5
6/7	R	l_4	0	–	θ_6/θ_7
$\mathcal{EM}$	–	l_5	0	–	–

Link 6 & 7 are the separate forceps which are both connected to link 5.

The FMs are linkage-based manipulators with 6 degrees of freedom (DoFs) and a diameter of 3.5mm, which can pass through a 3.8mm endoscopic working channel and has enough distal dexterity. In this work, we design four versions of FMs according to the function of the surgical tools as

shown in Fig. 2**B**. The scissors and needle holder are not required to pull and lift target tissue, so the wrist joint (J_5 in Fig. 2**A**) is canceled for compact size, and other joints are the same. On the contrary, the forceps lift target tissue to provide a submucosal view for effective cutting by the FW. Besides, we design two forceps with different structures, single and double controllable forceps, for multi-usage choices. Then, we utilize the modified DH method to describe the kinematics and integrate them into a unified one shown in Fig. 2**A**. The first 4 DoFs $\{J_1, \cdots, J_4\}$ in a PRRR sequence (R: revolute joint, P: prismatic joint) composes the manipulator's arm. The last 2 DoFs correspond to the pitching motion of the wrist and opening/closing motion. It is easy for users to modify the unified kinematics for the structure without a wrist. According to the DH parameters listed in Table I, the homogeneous transformation matrix ${}^{\mathcal{MB}}\boldsymbol{T}_{\mathcal{ME}} \in SE(3)$, representing the pose of the FM end-effector frame $\mathcal{ME}$ with respect to the base frame $\mathcal{MB}$, can be easily computed applying the modified DH convention to the kinematic chain $\{J_1, \cdots, J_6\}$.

B. Flexible Wrist Kinematics

To make the FW pass through a 2.8mm endoscopic working channel and work dexterously in operation, we use a compact cable-driven parallel structure to create a dexterous wrist with 2 DOFs $\{J_2, J_3\}$ and a diameter of 2.5mm. Besides, a translational DoF J_1 is added to enlarge the workspace. Since the forward kinematics of the parallel robot is complicated, we employ two serial revolute joints to approximate it, which obtains desirable performance in simulation. The alternative kinematics is depicted in Fig. 3**A** where the 3 DoFs are in a PRR sequence. According to the modified DH parameters listed in Table II, it is straightforward to solve the kinematic chain $\{J_1, \cdots, J_4\}$ to acquire the homogeneous transformation matrix ${}^{\mathcal{WB}}\boldsymbol{T}_{\mathcal{WE}} \in SE(3)$.

TABLE II
DH PARAMETERS OF FW

link	joint	a_{i-1}	α_{i-1}	d_i	θ_i
1	P	0	0	d_1	–
2	R	l_1	$-\pi/2$	–	$-\pi/2+\theta_2$
3	R	0	$\pi/2$	–	θ_3
$\mathcal{WE}$	–	l_2	0	–	–

Besides the alternative method, we also use a geometric method [22] to model the kinematics of FW's parallel structure more accurately. The spatial geometry of the cable-driven joint of the FW is illustrated in Fig. 3**B**. l_i is the length of the ith cable which can be expressed in (1).

$$l_i {}^b\boldsymbol{u}_i = \overrightarrow{{}^bB_iE_i} = {}^b\boldsymbol{b}_i - {}^b\boldsymbol{a}_i = {}^b\boldsymbol{R}_e {}^e\boldsymbol{b}_i + {}^b\boldsymbol{t}_e - {}^b\boldsymbol{a}_i, \quad (1)$$

where $\boldsymbol{u_i^b}$, a unit vector, indicates the direction of the vector $\overrightarrow{B_iE_i}$; B_i is the ith cable anchor point on the base frame and the corresponding vector is ${}^b\boldsymbol{a}_i$; E_i is the anchor point on the end-effector frame and the corresponding vector is ${}^e\boldsymbol{b}_i$; ${}^b\boldsymbol{R_e}$ and ${}^b\boldsymbol{t_e}$ denote the rotation matrix and translation vector of end-effector frame with respect to the base frame. Then, the unit vector can be calculated as follows.

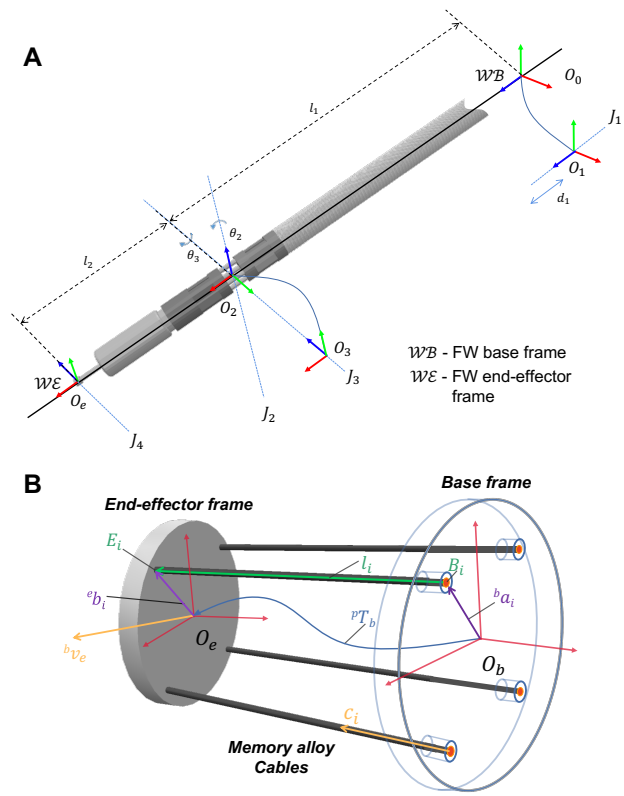

Fig. 3. Flexible wrist. **A** Alternative FW kinematic illustration. **B** FW kinematic illustration of the cable-driven parallel mechanism.

$${}^b\boldsymbol{u}_i = \frac{\overrightarrow{{}^bB_iE_i}}{\|\overrightarrow{{}^bB_iE_i}\|_2} = \frac{{}^b\boldsymbol{b}_i - {}^b\boldsymbol{a}_i}{\|\overrightarrow{{}^bB_iE_i}\|_2} = \frac{{}^b\boldsymbol{R}_e {}^e\boldsymbol{b}_i + {}^b\boldsymbol{t}_e - {}^b\boldsymbol{a}_i}{\|\overrightarrow{{}^bB_iE_i}\|_2} \quad (2)$$

The forward Jacobian matrix L_w of the FW is acquired through differentiating (2) as follows.

$$\boldsymbol{L_w} = \begin{bmatrix} {}^b\boldsymbol{u}_1^{\mathrm{T}} & ({}^b\boldsymbol{R}_e {}^e\boldsymbol{b}_1 \times {}^b\boldsymbol{u}_1)^{\mathrm{T}} \\ \vdots & \vdots \\ {}^b\boldsymbol{u}_m^{\mathrm{T}} & ({}^b\boldsymbol{R}_e {}^e\boldsymbol{b}_m \times {}^b\boldsymbol{u}_m)^{\mathrm{T}} \end{bmatrix}, \quad (3)$$

where m is the number of cables ($m = 4$ in this FW). Therefore, the mapping relationship between the end-effector velocity ${}^b\boldsymbol{v}_e$ and the cable velocity, $\boldsymbol{c} = [c_1, \cdots, c_m]^{\mathrm{T}}$, can be further obtained as follows.

$$\boldsymbol{c} = \boldsymbol{L_w} {}^b\boldsymbol{v_e} \quad (4)$$

C. Endoscopic Snake Bone Kinematics

The ESB is a cable-driven serial robot that consists of eighteen revolute joints. It is fixed at the distal end of the

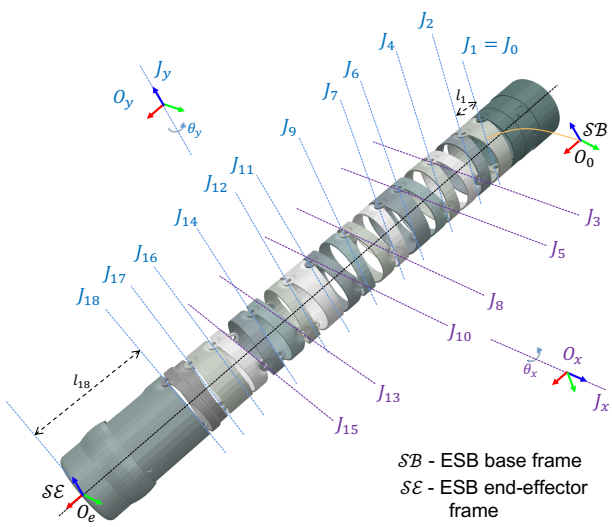

Fig. 4. Kinematic illustration of the endoscopic snake bone. Subscripts x, y indicate the link number.

TABLE III
DH PARAMETERS OF THE ESB

link	joint	a_{i-1}	α_{i-1}	d_i	θ_i
1	R	0	0	–	θ_1
2	R	l_1	0	–	θ_2
3	R	l_2	$-\pi/2$	–	θ_3
4	R	l_3	$\pi/2$	–	θ_4
5	R	l_4	$-\pi/2$	–	θ_5
6	R	l_5	$-\pi/2$	–	θ_6
7	R	l_6	0	–	θ_7
8	R	l_7	$-\pi/2$	–	θ_8
9	R	l_8	$\pi/2$	–	θ_9
10	R	l_9	$-\pi/2$	–	θ_{10}
11	R	l_{10}	$\pi/2$	–	θ_{11}
12	R	l_{11}	0	–	θ_{12}
13	R	l_{12}	$-\pi/2$	–	θ_{13}
14	R	l_{13}	$\pi/2$	–	θ_{14}
15	R	l_{14}	$-\pi/2$	–	θ_{15}
16	R	l_{15}	$\pi/2$	–	θ_{16}
17	R	l_{16}	0	–	θ_{17}
18	R	l_{17}	0	–	θ_{18}
$\mathcal{SE}$	–	l_{18}	0	–	–

endoscope. In the simulation, we assume that all joints can be controlled independently. As a result, its kinematics is also built using the modified DH method as shown in Fig. 4. The forward homogeneous transformation matrix ${}^{\mathcal{SB}}\boldsymbol{T}_{\mathcal{SE}} \in SE(3)$ can be solved according to the parameters listed in Table III.

III. GESRsim Architecture

Fig. 5 illustrates the GESRsim architecture. The built GESRsim comprises a robotic scene and a coding part. In the robotic scene, many components are provided in the format of *ttm*, including our designed manipulators (FMs and FW), a complete endoscope with snake bone, surgical training tools, human anatomic models and other auxiliary environment models in the operating room. For the coding part, we can use the built-in plugin (e.g., chai3d [23]), remote API and ROS interface to connect commercial haptic devices (Geomagic Touch^{TM}, 3D SYSTEMS) or our in-house developed master hand (Robo, China) for haptic rendering. Additionally, dominant AI frameworks, like PyTorch or Spinning Up (DRL), can communicate with the GESRsim via python remote API.

Based on the above robotic components and programming approach, two main applications are implemented in the GESRsim. For robotic GES training, some surgical training modules are established and can be used for teleoperation training; for autonomous control, some algorithms are implemented, including visual servoing and DRL. The details are introduced in sections IV and V.

IV. Robotic GES Training

One intention of building this simulator is to provide a low-cost and user-friendly platform for researchers and surgeons to master the real GESR system. In ESD, surgeons need to teleoperate the FM and the FW to grasp&lift and cut lesions, respectively. Therefore, we designed two basic training scenarios for the FM and the FW separately and one ESD scenario in the stomach for the dual-arm collaboration exercise as shown in Fig. 6. We firstly design a common peg-on-board task for the FMs with forceps as shown in Fig. 6**A**. According to software indication, the user needs to teleoperate the FM to grasp the rings and place them on the corresponding holders. Regarding the FW, its main function is to orient the end-effector. Hence, we designed three paths with tiered difficulty: plus –> infinity –> star, as shown in Fig. 6**B**. The operator must control the FW to follow a given path as precisely as possible. The last scene is a comprehensive training combining the FM and FW. The task is to cut a lesion in the stomach, which requires the skills acquired from the previous two training.

The design of training scenes in our GESRsim follows the principle of tiered difficulty. Users can start with separate FM and FW training, increase the difficulty as their skills improve, and finally perform dual-arm collaboration exercises. Besides, the establishment of training scenes is customizable in GESRsim, which means users can easily modify or create a training scene based on the provided models, such as manipulators, training tools, etc., to fit their requirements.

V. Autonomous Control

In practical operation, the primary control mode for the GESR system is teleoperation. However, the tendency of robotic systems is from teleoperation to semi-autonomy, even to complete autonomy. The GESRsim is a useful tool to verify different autonomous algorithms, such as [24] and [25]. Herein, we implement two advanced control algorithms in the GESRsim, including visual servo (VS) control and deep reinforcement learning (DRL) [26].

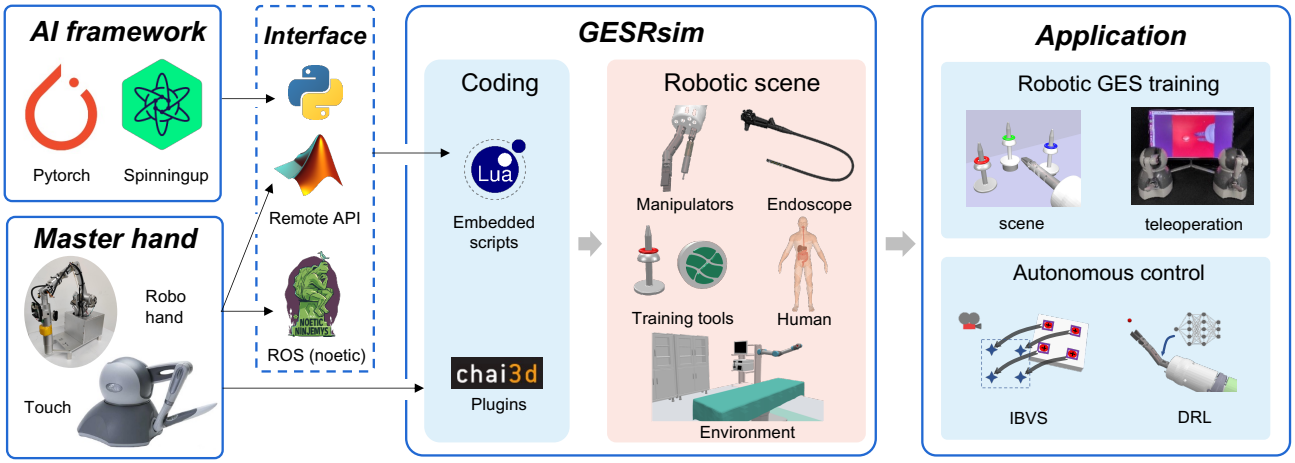

Fig. 5. GESRsim architecture. The GESR simulator consists of coding and robotic scene. The coding can be integrated with other AI frameworks (e.g., Pytorch, Spinning up, etc.) and hardware via a built-in interface. The robotic scene provides many objects to establish different surgical robot scenarios which can be used in robotic GES training and autonomous control.

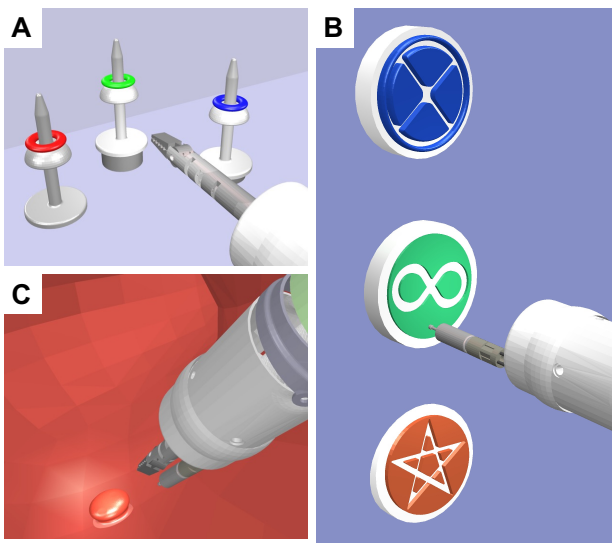

Fig. 6. Training scenes of the GESRsim. **A** Peg-on-board for the FM. **B** Path following for the FW. **C** ESD exercise for both FM and FW.

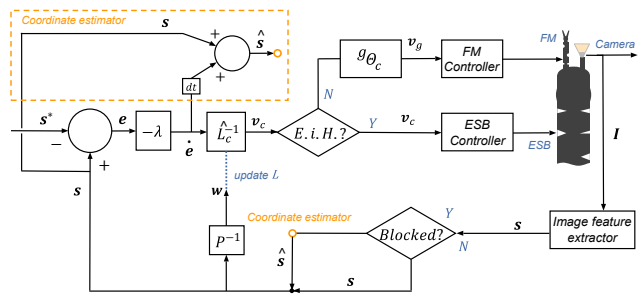

Fig. 7. Block diagram of IBVS control with eye-in-hand and eye-to-hand modes. Note that E.i.H. denotes eye-in-hand. Orange dashed box indicates the coordinate estimator.

A. Visual Servoing Control

VS is an effective approach to automatically controlling the endoscope to localize or follow targets in real-time [27], after efficient and intelligent image processing [28]. In the field of surgical image processing, numerical researches focus on instrument detection [29] or segmentation [30] and then the endoscope is controlled to follow instrument autonomously [31]. There are two VS control strategies: image-based visual servoing (IBVS) and position-based visual servoing (PBVS). Because the manipulator could be blocked in the endoscopic view, PBVS is difficult to execute in the confined and complex gastrointestinal tract. As a result, IBVS is adopted in this work.

In ESD, the GESR system is teleoperated to perform two procedures. Firstly, surgeons need to adjust the endoscope pose to make the target in the center of the endoscopic view and the reachable workspace of the FM and the FW. Secondly, the manipulators are teleoperated to grasp the target while the ESB is kept still. Based on these two steps, VS control needs to be executed on the ESB and FM, respectively, to automate the whole ESD procedure. Moreover, because of the different assembly positions of the endoscope relative to the ESB and the FM, the first step belongs to the eye-in-hand IBVS control for the ESB, and the second step belongs to the eye-to-hand IBVS control for the FM. To execute complete VS control for the GESRsim, we propose an integrated IBVS control strategy as shown in Fig. 7.

The detail of the control strategy is explained as follows. Taking Fig. 8**A** for example, at each time step, the endoscope in the GESRsim sends a detected image $\boldsymbol{I}$ with four red tracking points to the image feature extractor. The classical blob detection algorithm is utilized to acquire their 2D positions $\boldsymbol{s} = [x, y]$ in the image plane. Nonetheless, because of the close distance between the manipulator and endoscope, some tracking points can be blocked by the manipulators. In

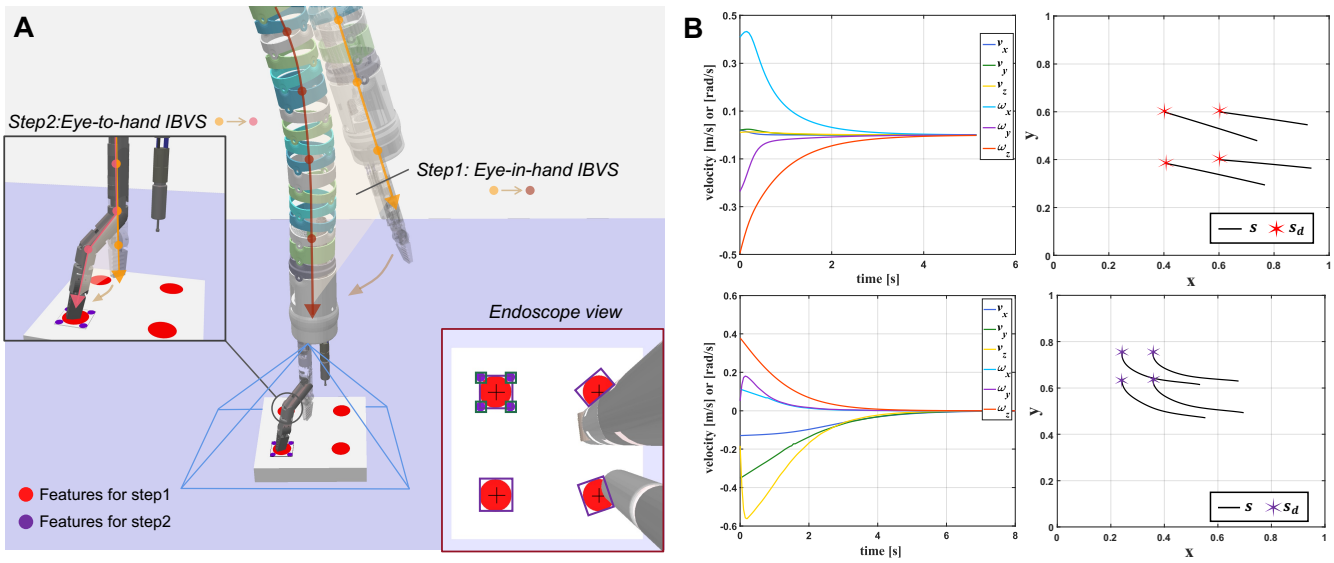

Fig. 8. Results of IBVS control. **A** The IBVS control process. Step1: eye-in-hand IBVS control, from yellow to brown. Tracking points are the red circles. Step2: eye-to-hand IBVS control, from yellow to pink. Tracking points are the purple circles. **B** Quantitative results. Top row: endoscope velocities and the red point positions in endoscopic view during step1; Botton row: gripper velocities and the purple point positions in endoscopic view during step2.

this case, we employ an estimated $\hat{\boldsymbol{s}}$ calculated at the last time step as the current tracking point (in the orange dashed box of Fig. 7). In other words, the control system will be open-loop until all tracking points are available again. Then, the corresponding 3D coordinates $\boldsymbol{w} = [X, Y, Z]$ can be computed by the inverse pinhole camera model $P^{-1}(s, Z)$ in (5). The depth information Z is directly read from the simulator via specific APIs, while the stereo endoscope can estimate it in the physical test.

$$\boldsymbol{w} = P^{-1}(s, Z) = \begin{cases} (0.5 - x)Z/k \\ (y - 0.5)Z/k \end{cases}, \tag{5}$$

where $k = 1/(2 \cdot \tan\alpha)$; α is half of the camera perspective view angle in CoppeliaSim. The 3D coordinates $\boldsymbol{w}$ will be used to update the image Jacobian matrix $\hat{L}_c$ [32] in (6), which is the mapping relationship between image feature velocity $\dot{\boldsymbol{s}} = [\dot{x}, \dot{y}]$ and endoscope velocity $\boldsymbol{v}_c$ in (8).

$$\hat{L}_c = k \cdot \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{-X}{Z^2} & \frac{-XY}{Z^2} & \frac{X^2}{Z^2} - 1 & \frac{Y}{Z} \\ 0 & \frac{-1}{Z} & \frac{Y}{Z^2} & 1 + \frac{Y^2}{Z^2} & \frac{-XY}{Z^2} & \frac{-X}{Z} \end{bmatrix} \tag{6}$$

$$\boldsymbol{v}_c = \hat{L}_c \dot{\boldsymbol{s}} \tag{7}$$

To exponential reduce the error $\boldsymbol{e}$ between the desired point position $\boldsymbol{s}^*$ and current point position, $\boldsymbol{e} = \boldsymbol{s} - \boldsymbol{s}^*$, the image velocity follows $\dot{\boldsymbol{s}} = -\lambda \boldsymbol{e}$ where λ is a proportional constant. Therefore, we obtain the endoscope velocity in the eye-in-hand control process as follows.

$$\boldsymbol{v}_c = -\hat{L}_c \lambda \boldsymbol{e} \tag{8}$$

In next eye-to-hand control process, an extra processing is required to map the endoscope velocity into the gripper velocity, $\boldsymbol{v}_g = {}^g\boldsymbol{\Theta}_c \boldsymbol{v}_c$, where ${}^g\boldsymbol{\Theta}_c$ is the transformation matrix [33] in (9).

$${}^g\boldsymbol{\Theta}_c = \begin{bmatrix} -{}^g\boldsymbol{R}_c & {}^g\boldsymbol{R}_c \boldsymbol{S}(-{}^g\boldsymbol{R}_c^{\mathrm{T}} {}^g\boldsymbol{t}_c) \\ 0 & -{}^g\boldsymbol{R}_c \end{bmatrix}, \tag{9}$$

where ${}^g\boldsymbol{R}_c$ and ${}^g\boldsymbol{t}_c$ are the rotation matrix and translation vector of the gripper with respect to endoscope; $\boldsymbol{S}$ is a skew-symmetric matrix [34]. Then, the acquired endoscope or gripper velocity is mapped to the joint velocities by the corresponding kinematics controller of the ESB and FM. Finally, the joint velocities will be applied to control the ESB and FM.

The qualitative and quantitative results of the proposed IBVS control are presented in Fig. 8 (**A** and B). Firstly, the ESB is automatically controlled to track the red points while the FM is still. From the curve plots, we find that the endoscope velocities converge within 4 seconds and make the red points evenly distributed in the endoscopic view. After that, the ESB is fixed, and the FM performs the eye-to-hand VS. Similarly, the FM can track the purple points on the upper left corner of the endoscopic view within 5 seconds. The simulation results prove that the integrated VS algorithm can be adopted in autonomous ESD applications.

B. Deep Reinforcement Learning

DRL is a promising AI technology to endow robots with human-level control [35]. The past 5 years have witnessed an explosion in the DRL research, especially in the fields of game [36] and robot control [37]. For surgical robots, some subtasks, such as suturing [38] and tissue cutting [39], have been performed by different state-of-the-art DRL algorithms. In the GESRsim, we also employed a popular DRL algorithm to train the control policy.

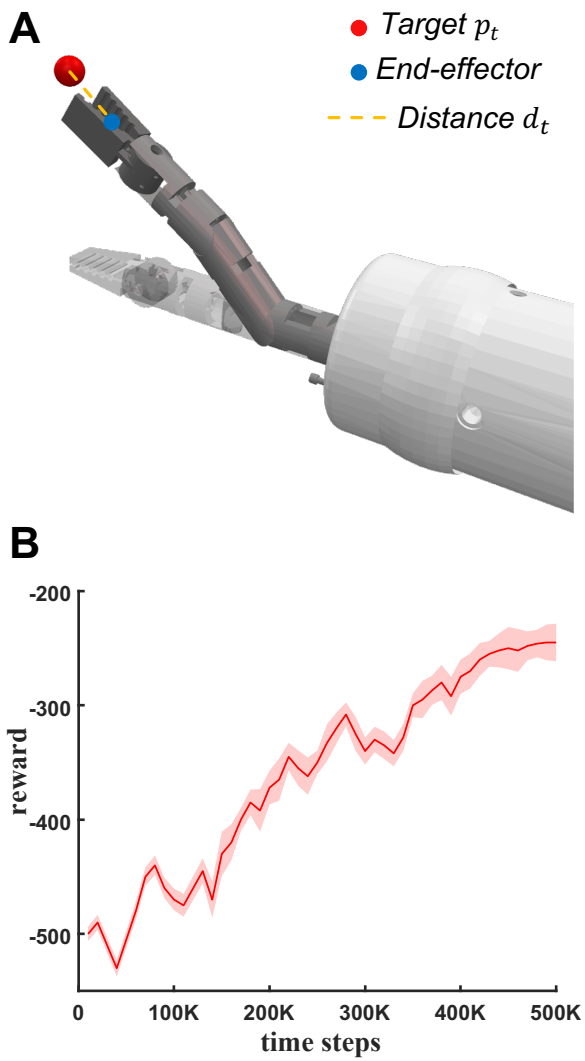

Fig. 9. "Reach" task. **A** Simulation setup. Needle holder-type manipulator is used to demonstrate the RL environment. The end-effector's frame is set at the center of forceps. **B** Learning curve. Solid lines denote the median performances. Shadings denote one standard deviation.

We establish a "Reach" environment as shown in Fig. 9**A** where the end-effector is required to reach any positions within its workspace autonomously. The agent's observation space o_t consists of the joint positions $\{q_t : \theta_1, \cdots, \theta_6\}$ listed in Table I, the previous joint velocities $\dot{q}_{t-1}$ and the target position p_t relative to the end-effector. The target position and the FM's joints are randomly preset in the FM's reachable workspace and the range of joint motion, respectively. The action a_t is the joint velocity $\dot{q}_t$. The reward function R in (10) is composed of two parts: the distance d_t between target and end-effector and the 2-norm of the actions. The latter component is used to avoid the motion jerk of the FM. For the RL agent, we employ PPO [40] algorithm, which is a modified version of the Actor-Critic algorithms. The reward during training starts at -500 and converges at aorund -250, as demonstrated in Fig. 9**B**. This shows that the agent's performance gradually improves during the training and keeps stable finally. The trained RL agent can guide the FM to complete the "Reach" task with desirable accuracy, which shows the potential of the autonomous control using DRL algorithms.

$$R(o_t, a_t) = -d_t - \|a_t\|_2 \tag{10}$$

VI. Conclusion and Future Work

In this work, we develop an open source based simulator for the GESR system, which is the first robotic simulator for GES or NOTES. The main kinematics models, including two manipulators and endoscopic snake bone, are established and provided. In addition, the GESRsim utilizes different APIs to use hardware and AI frameworks, which is user-friendly for developers to verify different control algorithms. This work has utilized VS control and RL algorithms to verify the performance of the GSERsim. Furthermore, according to the function of each manipulator, we create corresponding training scenes respectively and combine them to build a similar ESD scenario. In conclusion, the GESRsim is a versatile and user-friendly tool to accelerate the development of GESR and augment the autonomy of surgical robots.

In future work, we will try to integrate with some physics engines or other simulators, such as Vortex and SOFA [41], which allow the soft material simulation to increase the fidelity and realism of surgical scenarios. Additionally, an increasing number of state-of-the-art algorithms will be integrated for surgical robotic autonomy.

Acknowledgment

This work was supported by the National Key R&D Program of China under Grant 2018YFB1307700 (with sub-program 2018YFB1307703) from the Ministry of Science and Technology (MOST) of China, the Funding Scheme for Virtual Teaching and Learning (3212141), Hong Kong Research Grants Council (RGC) Collaborative Research Fund (C4063-18G:2300056), the Shun Hing Institute of Advanced Engineering (SHIAE project BME-p1-21, 8115064) at the Chinese University of Hong Kong (CUHK) and Hong Kong Research Grants Council (RGC-GRF) project 14211420.

References

[1] J. M. Kinross, S. E. Mason, G. Mylonas, and A. Darzi, "Next-generation robotics in gastrointestinal surgery," *Nature Reviews Gastroenterology & Hepatology*, vol. 17, no. 7, pp. 430–440, 2020.

[2] Y. Zhou, H. Ren, M. Q.-H. Meng, Z. T. H. Tse, and H. Yu, "Robotics in natural orifice transluminal endoscopic surgery," *Journal of Mechanics in Medicine and Biology*, vol. 13, no. 02, p. 1350044, 2013.

[3] L. Qiu and H. Ren, "Endoscope navigation and 3d reconstruction of oral cavity by visual slam with mitigated data scarcity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2197–2204, 2018.

[4] L. Zhang, K. S. Kumar, H. He, C. J. Cai, X. He, H. Gao, S. Yue, C. Li, R. C.-S. Seet, H. Ren, *et al.*, "Fully organic compliant dry electrodes self-adhesive to skin for long-term motion-robust epidermal biopotential monitoring," *Nature communications*, vol. 11, no. 1, pp. 1–13, 2020.

[5] L. Qiu, C. Li, and H. Ren, "Real-time surgical instrument tracking in robot-assisted surgery using multi-domain convolutional neural network," *Healthcare Technology Letters*, vol. 6, no. 6, pp. 159–164, 2019.

[6] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, *et al.*, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, p. eaam8638, 2017.

[7] J. Troccaz, G. Dagnino, and G.-Z. Yang, "Frontiers of medical robotics: From concept to systems to clinical translation," *Annual Review of Biomedical Engineering*, vol. 21, pp. 193–218, 2019.

[8] G. Fagogenis, M. Mencattelli, Z. Machaidze, B. Rosa, K. Price, F. Wu, V. Weixler, M. Saeed, J. E. Mayer, and P. E. Dupont, "Autonomous robotic intracardiac catheter navigation using haptic vision," *Science Robotics*, vol. 4, no. 29, p. eaaw1977, 2019.

[9] H. Saeidi, J. Opfermann, M. Kam, S. Wei, S. Leonard, M. Hsieh, J. Kang, and A. Krieger, "Autonomous robotic laparoscopic surgery for intestinal anastomosis," *Science Robotics*, vol. 7, no. 62, p. eabj2908, 2022.

[10] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci® surgical system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6434–6439, IEEE, 2014.

[11] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2149–2154, IEEE, 2004.

[12] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A V-rep simulator for the da Vinci research kit robotic platform," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pp. 1056–1061, IEEE, 2018.

[13] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1321–1326, IEEE, 2013.

[14] F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," *arXiv preprint arXiv:1903.02090*, 2019.

[15] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, "SurRoL: An open-source reinforcement learning centered and dVRK compatible platform for surgical robot learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1821–1828, IEEE, 2021.

[16] A. Z. Taddese, P. R. Slawinski, K. L. Obstein, and P. Valdastri, "Closed loop control of a tethered magnetic capsule endoscope," *Robotics Science and Systems: Online Proceedings*, vol. 2016, 2016.

[17] K. İncetan, I. O. Celik, A. Obeid, G. I. Gokceler, K. B. Ozyoruk, Y. Almalioglu, R. J. Chen, F. Mahmood, H. Gilbert, N. J. Durr, *et al.*, "VR-Caps: A virtual environment for capsule endoscopy," *Medical Image Analysis*, vol. 70, p. 101990, 2021.

[18] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, *et al.*, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[19] H. Gao, X. Xiao, L. Qiu, M. Q.-H. Meng, N. K. K. King, and H. Ren, "Remote-center-of-motion recommendation toward brain needle intervention using deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8295–8301, 2021.

[20] C. Li, Y. Yan, X. Xiao, X. Gu, H. Gao, X. Duan, X. Zuo, Y. Li, and H. Ren, "A miniature manipulator with variable stiffness towards minimally invasive transluminal endoscopic surgery," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5541–5548, 2021.

[21] H. Gao, X. Xiao, X. Yang, T. Zhang, X. Zuo, Y. Li, and H. Ren, "A miniature 3-dof flexible parallel robotic wrist using NiTi wires for gastrointestinal endoscopic surgery," in *2022 IEEE International Conference on Robotics and Automation (ICRA) Workshop*, IEEE, 2022.

[22] Z. Zake, S. Caro, A. S. Roos, F. Chaumette, and N. Pedemonte, "Stability analysis of pose-based visual servoing control of cable-driven parallel robots," in *International Conference on Cable-Driven Parallel Robots*, pp. 73–84, Springer, 2019.

[23] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, J. Warren, O. Khatib, and K. Salisbury, "The chai libraries," in *Proceedings of Eurohaptics 2003*, pp. 496–500, 2003.

[24] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[25] X. Li, Z. Serlin, G. Yang, and C. Belta, "A formal methods approach to interpretable reinforcement learning for robotic planning," *Science Robotics*, vol. 4, no. 37, 2019.

[26] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[27] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[28] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[29] L. C. García-Peraza-Herrera, W. Li, C. Gruijthuijsen, A. Devreker, G. Attilakos, J. Deprest, E. Vander Poorten, D. Stoyanov, T. Vercauteren, and S. Ourselin, "Real-time segmentation of non-rigid surgical tools based on deep learning and tracking," in *International Workshop on Computer-Assisted and Robotic Endoscopy*, pp. 84–95, Springer, 2016.

[30] Y. Sun, B. Pan, and Y. Fu, "Lightweight deep neural network for real-time instrument semantic segmentation in robot assisted minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3870–3877, 2021.

[31] X. Ma, C. Song, P. W. Chiu, and Z. Li, "Autonomous flexible endoscope for minimally invasive surgery with enhanced safety," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2607–2613, 2019.

[32] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[33] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *2000 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2741–2746, IEEE, 2000.

[34] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 525–532, 1998.

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[36] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[37] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, vol. 4, no. 26, 2019.

[38] V. M. Varier, D. K. Rajamani, N. Goldfarb, F. Tavakkolmoghaddam, A. Munawar, and G. S. Fischer, "Collaborative suturing: A reinforcement learning approach to automate hand-off task in suturing for surgical robots," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1380–1386, IEEE, 2020.

[39] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2D orthotropic gauze with deep reinforcement learning policies for tensioning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2371–2378, IEEE, 2017.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[41] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, *et al.*, "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pp. 283–321, Springer, 2012.