

# Portable dVRK: an augmented V-REP simulator of the da Vinci Research Kit

**Giuseppe Andrea Fontanelli<sup>1</sup>, Mario Selvaggio<sup>1</sup>, Marco Ferro<sup>2</sup>,**

**Fanny Ficuciello<sup>1</sup>, Marilena Vendittelli<sup>3</sup>, and Bruno Siciliano<sup>1</sup>**

<sup>1</sup>Università degli Studi di Napoli Federico II, Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, via Claudio, 21, 80125 Napoli NA, Italy.

{giuseppeandrea.fontanelli, mario.selvaggio, fanny.ficuciello, bruno.siciliano}@unina.it

<sup>2</sup>Sapienza Università di Roma, Dipartimento di Ingegneria Informatica, Automatica e Gestionale, via Ariosto, 25, 00185 Roma RM, Italy.

ferro@diag.uniroma1.it

<sup>3</sup>Sapienza Università di Roma, Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni, via Eudossiana, 18, 00184 Roma RM, Italy.  
marilena.vendittelli@uniroma1.it

---

*Abstract:* The da Vinci Research Kit (dVRK) is a first generation da Vinci robot repurposed as a research platform and coupled with software and controllers developed by research users. An already quite wide community is currently sharing the dVRK (32 systems in 28 sites worldwide). The access to the robotic system for training surgeons and for developing new surgical procedures, tools and new control modalities is still difficult due to the limited availability and high maintenance costs. The development of simulation tools provides a low cost, easy and safe alternative to the use of the real platform for preliminary research and training activities. The Portable dVRK, which is described in this work, is based on a V-REP simulator of the dVRK patient side and endoscopic camera manipulators which are controlled through two haptic interfaces and a 3D viewer, respectively. The V-REP simulator is augmented with a physics engine allowing to render the interaction of new developed tools with soft objects. Full integration in the ROS control architecture makes the simulator flexible and easy to be interfaced with other possible devices. Several scenes have been implemented to illustrate performance and potentials of the developed simulator.

---

*Keywords:* Robotic surgery simulators; Minimally invasive robotic surgery; Virtual reality

## 1 Introduction

Since 2012 Intuitive Surgical has started to donate to universities and research institutions core components of retired first generation da Vinci robot repurposed as a

research platform. The da Vinci Research Kit (dVRK)<sup>1</sup> couples this platform with software and controllers developed at Johns Hopkins University LCSR and Worcester Polytechnic Institute AIM Lab [1]. Currently, there are more than 30 systems available in 28 sites worldwide. This quite large community (already sharing the dVRK) witnesses the relevance of this platform in surgical robotics research. By targeting this system as the elective experimental testbed research aims at augmenting the surgeon's abilities [2] and ranges from haptic-based teleoperation control [3] to sensor-based shared autonomy [4].

Despite the importance and the current wide availability of the platform, a large part of the research community in the field does not have access to it. Even when available, a wise use of this resource is desirable in order to limit costs and difficulties in replacing components. In this perspective, simulation tools help in overcoming such issues, by developing new surgical tools [5, 6], integrating learning in simulation environments, and providing an easy-to-access educational tool to students.

Currently existing simulators are mainly oriented to surgeons' training [7]. The most relevant simulation systems are: *Robotic Surgery Simulator (RoSS)* [8], *Sim-Surgery Education Platform (SEP)*<sup>2</sup>, *da Vinci Trainer*<sup>3</sup>, *da Vinci Skills Simulator*<sup>4</sup>, *Robotix Mentor*<sup>5</sup> and *Chiron* [9]. Beside providing training utilities, each of these simulators allow EndoWrist manipulation, camera control, needle control and clutching, and a realistic representation of the surgical workspace. A useful guide to select the simulator that best fits the user's need is provided in [10].

Given that the above mentioned simulators are dedicated to surgeons' training they do not include a simulation model of the whole robot, i.e., Setup joints (SUS), Patient Side Manipulators (PSMs) and Endoscopic Camera Manipulator (ECM). On the other hand, simulation models of robotics systems can be easily obtained using open-source robotic simulators. For instance, in [11] authors integrated the dVRK system in the Gazebo simulation framework to develop and test a method for computing haptic forces for tele-operated surgical robots. Integration of the training capabilities of surgical simulators with the functionalities of open-source robotic simulators would serve a large community of users both in the robotics and in the surgical domain.

In this paper, we propose a portable dVRK simulator developed in V-REP [12]. With respect to the other robotic simulator frameworks that are currently available in the research community (e.g., Gazebo, Webots, ARGoS, Marilou), V-REP offers higher flexibility and ease of use in the simulation of multi-robot systems. It is also computationally efficient in terms of CPU usage and allows object mesh manipulation and optimization [13]. The control architecture is distributed and each object/model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution. In addition, V-

<sup>1</sup> <http://research.intusurg.com/dvrkwiki>

<sup>2</sup> <http://www.simsurgery.com>

<sup>3</sup> <http://www.mimicsimulation.com/products/dv-trainer/>

<sup>4</sup> [https://www.intuitivesurgical.com/products/skills\\_simulator/](https://www.intuitivesurgical.com/products/skills_simulator/)

<sup>5</sup> <https://simbionix.com/simulators/robotix-mentor/>

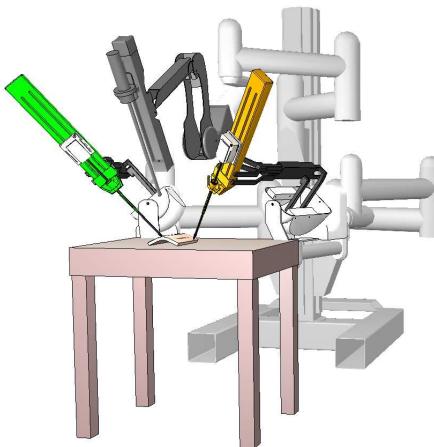


Figure 1  
The da Vinci Research Kit V-REP simulator.

REP supports C/C++, Python, Java, Lua, Matlab or Octave, and runs simulations based on the most commonly available dynamic engines, e.g., Bullet, ODE, Vortex, Newton. The presented simulator inherits such flexibility and is easily extendible. To show and highlight these properties, several scenarios are presented, where the simulator is easily interfaced with real input devices, such as the real surgeon master console, cheap haptic devices and a 3D vision system for virtual reality applications.

The developed simulator includes the kinematic models of the SUJ, PSMs, ECM and the camera sensor and it is interfaced with the ROS framework, as described in Sect. 2. Furthermore, to provide the user with a full immersion experience, a virtual reality headset is integrated together with low cost haptic interfaces as described in Sect. 3. In Sect. 4 four sample scenes, developed for manipulation of rigid dynamic objects, suturing, needle tracking and visual servoing tasks, are provided and ready for use. The potentiality of the simulator for integrating advanced instrument prototypes in a powerful and easy way is illustrated in Sect. 5. This is a very useful characteristic for design, testing and validation. Section 5.1 describes the integration of the simulator with a physics engine, Bullet Real-Time Physics Simulation<sup>6</sup>, to model soft contacts and deformable objects like tissues and organs.

## 2 V-REP Simulated Environment

The robot structure, composed of a SUJ, two PSMs and one ECM, constitutes the core of the dVRK V-REP simulator (Fig. 1) here described, together with its general performances. Starting from the CAD models available in the John Hopkins dVRK git webpage<sup>7</sup>, the robotic arms have been built by means of two types of mesh for each robot link: (i) one visual mesh with structure and texture similar to the real robot link, (ii) one simplified convex dynamic and respondable mesh used to simu-

<sup>6</sup> <https://pybullet.org/wordpress/>

<sup>7</sup> <https://github.com/jhu-dvrk>

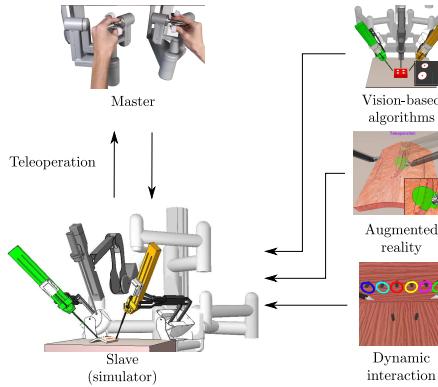


Figure 2  
Simulated environment with different application examples.

late dynamics and contacts<sup>8</sup>. The kinematic chain of each robotic arm is described in Sect. 2.2 and it is realized by linking meshes and joints in a *joint-respondable-visual* sequence. The dynamic parameters, obtained by the identification procedure performed in [14], are included for each respondable link of the two PSMs. Each PSM has been provided with standard laparoscopic instruments, such as the standard needle driver and the cadiere forceps. Two cameras have been included at the end of the endoscope to simulate the binocular vision system of the real dVRK endoscope. In order to have an acceptable simulated sampling time, while preserving a good resolution, a good trade-off is to set for the cameras half the resolution of the real endoscope, *i.e.* at  $320 \times 288$  pixels. The resulting complete robot is composed of 10178 triangles. With this settings the scene is rendered at 45 fps and the dynamics is simulated at 200 Hz with a computer powered by a Intel I7-7770HQ processor, 16 GB of ram and Nvidia GeForce 960M .

The V-REP simulator allows easily inclusion of different robots, dynamic objects, devices and sensors. These facilities allow creating advanced V-REP scenes that include control strategies, *e.g.*, visual servoing or vision-based object tracking, augmented reality and simulation of rigid objects dynamics and interaction (see Fig. 2).

## 2.1 Integration with the dVRK control software infrastructure

The V-REP simulator is designed to be fully integrable into the dVRK control infrastructure. To link our simulator to the dVRK low-level control software [1], the high-level ROS framework has been used. Therefore, the user can employ the simulator in different modalities: (*i*) the telemanipulated one, using the dVRK MTMs; (*ii*) in combination with the real robotic PSMs and ECM, to implement augmented reality simulations/algorithms; (*iii*) as standalone, by controlling the simulated robot using the ROS framework (*e.g.*, through C++, MATLAB and Python ROS nodes), or directly in V-REP using the embedded scripts.

The control software architecture of the dVRK is represented in Fig. 3 and is de-

<sup>8</sup> Dynamic respondable shapes influence each other during dynamic collisions and are subject to gravity and inertial forces.

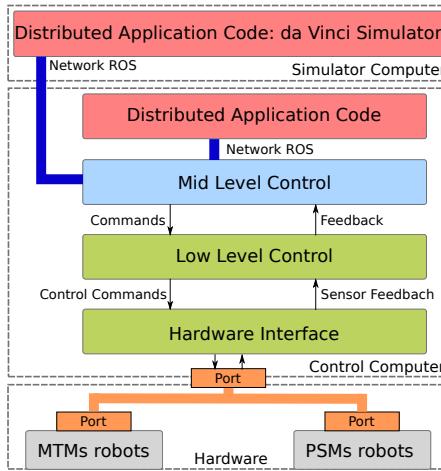


Figure 3  
Software architecture.

scribed more in details in [15]. The dVRK control architecture is composed by: (i) a hardware interface to communicate with the embedded actuator controllers through the fire-wire bus, implementing the safety checks, etc.; (ii) a low-level layer that implements all the algorithms for the inverse kinematics, master impedance control etc.; (iii) a mid-level layer that implements the ROS communication and the high level controllers. The dVRK simulator (running in a dedicate computer) and the dVRK console communicate through ROS topics. In particular, we use the *v\_repExtRosInterface* to publish the state of the simulated robot joints (PSMs, SUJ, ECM) and the gripper state for the PSMs. To control the robots joints motion from ROS, the simulator subscribes to two topics of *sensor\_msgs:joint\_state* type. Both joints and objects topics are streamed at 220Hz while cameras topics at 60 Hz<sup>9</sup>.

This architecture allows easily interfacing the simulator with the mid level control of the dVRK (to command the simulated robot through MTMs) or to other ROS-integrated input device (*e.g.*, haptic devices).

## 2.2 Kinematic Model of the dVRK Robotic Arms

In the following, we describe the kinematics of the patient-side manipulators involved in the presented simulator (*i.e.*, two PSMs and an ECM).

### 2.2.1 Setup Joints arm kinematics

The two PSMs and the ECM are mounted on the SUJ, that is an articulated robotic structure composed by three arms allowing the manual spatial positioning of the two PSMs and the ECM. The SUJ moving the PSMs are two 6-degrees-of-freedom (DoFs) arms (that we indicate hereafter as SUJ-PSMs) while the SUJ moving the ECM is a 4-DoFs arm (SUJ-ECM). The SUJ arms are not actuated. Nevertheless

<sup>9</sup> The simulation requires to be run in *threaded-rendering* mode, in order to decouple the rendering and the control scripts and speed up the execution.

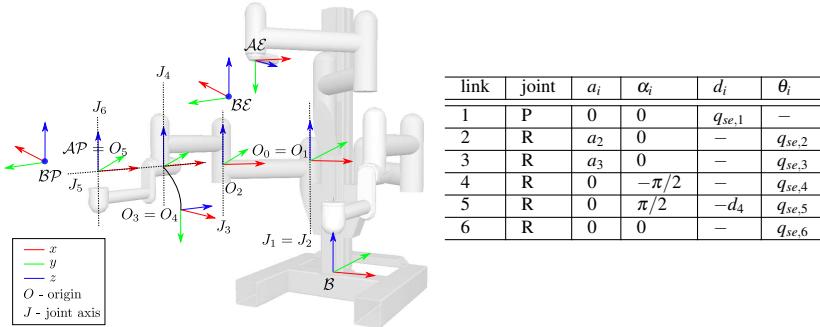


Figure 4  
SUJ kinematic description.

the angular position can be read thanks to embedded potentiometers<sup>10</sup>. The SUJ-PSMs arms generalized coordinates are given by  $q_{sp} = [q_{sp,1}, \dots, q_{sp,6}]$ . By applying the standard DH convention to the kinematic chain composed of  $\{J_1, \dots, J_6\}$  joints (Fig. 4), the homogeneous transformation matrix<sup>11</sup>  $T_{\mathcal{A}\mathcal{P}}(q_{sp}) \in SE(3)$ , representing the pose of the SUJ-PSMs end-effector frame  $\mathcal{A}\mathcal{P}$  :  $\{O_{ap}; x_{ap}, y_{ap}, z_{ap}\}$  with respect to the base frame  $\mathcal{B}$  :  $\{O_b; x_b, y_b, z_b\}$ , can be easily computed. The vector of the SUJ-ECM arm generalized coordinates is given by  $q_{se} = [q_{se,1}, \dots, q_{se,4}]$ . Therefore, the homogeneous transformation matrix  $T_{\mathcal{A}\mathcal{E}}(q_{se}) \in SE(3)$ , that defines the pose of the SUJ-ECM end-effector frame  $\mathcal{A}\mathcal{E}$  :  $\{O_{ae}; x_{ae}, y_{ae}, z_{ae}\}$  with respect to the base frame  $\mathcal{B}$  :  $\{O_b; x_b, y_b, z_b\}$ , can be computed considering only the first four rows of Table within Fig. 4. Notice that, two constant homogeneous transformation matrices  $T_{\mathcal{B}\mathcal{P}} \in SE(3)$  and  $T_{\mathcal{B}\mathcal{E}} \in SE(3)$  must be considered to complete the kinematics description, providing the transformation between  $\mathcal{A}\mathcal{P}$  and  $\mathcal{A}\mathcal{E}$  (respectively the last SUJ-PSM and SUJ-ECM frames) and the base frames  $\mathcal{B}\mathcal{P}$  and  $\mathcal{B}\mathcal{E}$  of the PSMs and of the ECM described in Sec. 2.2.2 and 2.2.3 (see Fig. 4).

## 2.2.2 PSM arm kinematics

The PSMs are two 7-DoFs actuated arms, where the first 6-DoFs correspond to Revolute (R) or Prismatic (P) joints, combined in a RRPRRR sequence. Each PSM arm moves a surgical instrument about a Remote Center of Motion (RCM) [14, 16]. The last DoF corresponds to the opening and closing motion of the gripper. By choosing the origin of frame  $\mathcal{B}\mathcal{P}$  in the RCM point and applying the standard DH convention to the kinematic chain  $\{J_1, \dots, J_6\}$  of Fig. 5, the homogeneous transformation matrix  $T_{\mathcal{G}\mathcal{P}}(q_p) \in SE(3)$  (where  $q_p = [q_{p,1}, \dots, q_{p,6}]$  is the vector of the PSM generalized coordinates), representing the pose of the gripper frame  $\mathcal{G}$  :  $\{O_g; x_g, y_g, z_g\}$  with respect to the base frame  $\mathcal{B}\mathcal{P}$  :  $\{O_b; x_{bp}, y_{bp}, z_{bp}\}$ , can be easily computed. The DH parameters are given in the table within Fig. 5, where  $a_5 = 0.0091$  m)

<sup>10</sup> <http://research.intusurg.com/dvrkwiki>

<sup>11</sup> Hereafter, we use the matrix notation  $T_b^a$ , where the superscript  $a$  denotes the frame in which vector components are expressed, the subscript  $b$  the current frame. For instance,  $T_{\mathcal{A}\mathcal{P}}^a$  denotes the pose of the SUJ-PSM attach point expressed in the base frame.

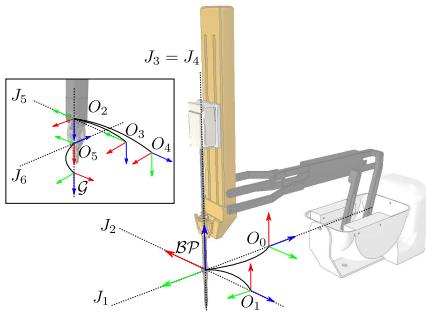


Figure 5  
PSM kinematic description.

link	joint	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	R	0	$-\pi/2$	—	$q_{p,1}$
2	R	0	$-\pi/2$	—	$q_{p,2}$
3	P	0	0	$q_{p,3}$	—
4	R	0	$\pi/2$	—	$q_{p,4}$
5	R	$a_5$	$-\pi/2$	—	$q_{p,5}$
6	R	0	$-\pi/2$	—	$q_{p,6}$

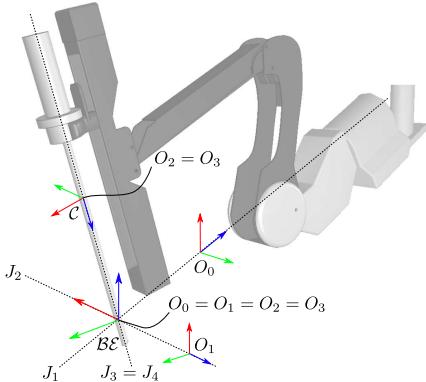


Figure 6  
ECM kinematic description.

link	joint	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	R	0	$-\pi/2$	—	$q_{e,1}$
2	R	0	$-\pi/2$	—	$q_{e,2}$
3	P	0	0	$q_{e,3}$	—
4	R	0	0	$d_4$	$q_{e,4}$

### 2.2.3 ECM arm kinematics

The ECM is a 4-DoF actuated arm, that moves the endoscopic camera about the RCM through revolute and prismatic joints, combined in a RRPR sequence. By choosing the origin of frame  $\mathcal{C}\mathcal{B}$  in the RCM point and applying the standard DH convention to the kinematic chain  $\{J_1, \dots, J_4\}$  of Fig. 6, the homogeneous transformation matrix  $T_{\mathcal{C}}^{\mathcal{B}}(q_e) \in SE(3)$  (where  $q_e = [q_{e,1}, \dots, q_{e,4}]$ ), representing the pose of the camera frame  $\mathcal{C} = \{O_c; x_c, y_c, z_c\}$  with respect to the base frame  $\mathcal{B}\mathcal{C} = \{O_{bc}; x_{bc}, y_{bc}, z_{bc}\}$ , can be easily computed. The DH parameters are given in the table within Fig. 6 where  $d_4 = 0.007\text{m}$ .

## 3 Integration with Oculus and Geomagic devices

In the perspective of providing an effective training tool for surgeons and a learning platform for students, it is useful to consider the possibility to interface the dVRK simulator with additional physical input devices, that may not be part of the original da Vinci robot platform. Specifically, in scenarios where the physical platform is not available, there could be the necessity of *reproducing* the surgeon console hosting the pair of master tool manipulators (MTMs), along with the two vision

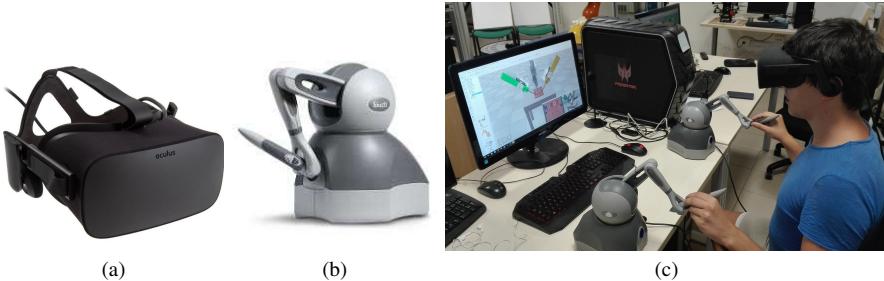


Figure 7

(a) The Head-mounted display of Oculus Rift and (b) the Geomagic Touch haptic device. The integration of the devices with the da Vinci simulator (c).

channels displaying images acquired from the ECM, employed to offer the surgeon the experience of a 3D vision.

To provide the user a fully immersive experience, the use of a virtual reality headset, such as the Oculus Rift<sup>12</sup> (see Fig. 7a), and a pair of haptic interfaces, such as the Geomagic Touch device<sup>13</sup> (see Fig. 7b), represents a non-expensive solution to reproduce the 3D vision system and the MTMs of the master console (see Fig 7c). By connecting the pair of cameras - placed at the end-effector of the ECM - with the Oculus head-mounted display (HMD) that can be freely moved in space, we can actually increase the potential of the framework and tele-operate the ECM through the movements of the user's head. On the other hand, the Geomagic Touch devices can be employed to tele-operate the PSMs of the robot and provide a feedback to the user rendering the interaction forces at the remote site. This scheme is detailed in [17] and summarized in the remainder of this section.

Implementing such functionalities is possible since Oculus and Geomagic devices come with a native SDK to develop custom applications to fully exploit the potentialities of the virtual reality headset and of the haptic interfaces, respectively. However, these SDKs make the overall system not easy to interface in Linux-based frameworks, as a full support and compatibility is granted only for Windows-based systems. Therefore, to allow this integration of devices, the overall distributed application code of the simulator is reconsidered to remove its ROS-dependent parts. Figure 8 shows the software architecture: an external application communicates with the Geomagic and Oculus devices through their corresponding libraries *Open-Haptics* and *LibOVR*, to read the state of the device and acquire specific information (e.g., tool position and velocity), or sends specific commands (e.g., rendering a given force feedback on the haptic tools). While the simulation is running, the application asks for the current joint configuration of the tele-operated PSMs end ECM, along with the current images acquired by the vision sensor objects, mounted at the end-effector of the ECM to simulate the endoscopic camera. The next two sections,

<sup>12</sup> <https://www.oculus.com/>

<sup>13</sup> <https://www.3dsystems.com/haptics-devices/touch>

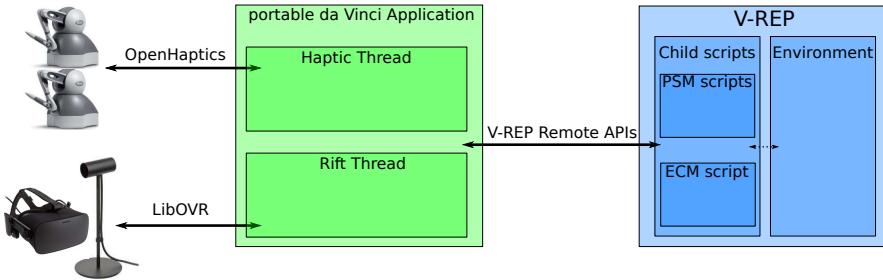


Figure 8

Modules and device communication scheme to interface Oculus Rift and Geomagic Touch devices to the simulator

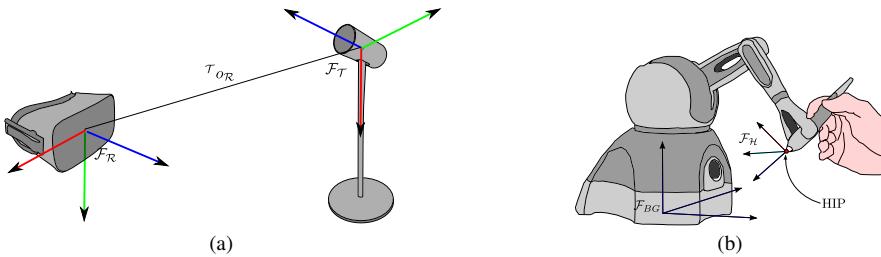


Figure 9

Reference frames of interest for (a) the Oculus Rift HMD and (b) the Geomagic Touch device, where the Haptic Interface Point (HIP) is highlighted.

provide additional details about the communication of Oculus Rift and Geomagic Touch devices with the proposed simulator.

### 3.1 Connecting the Oculus Rift device

The Oculus Rift hardware kit considers an infrared-based positional tracking system called Constellation, that provides an high-rate and accurate measurement of the HMD pose with respect to the tracker reference frame  $\mathcal{F}_{\mathcal{T}}$  (see Fig. 9a). As illustrated in Sect. 2.2.3, the ECM is a 4-DoF manipulator moving the end-effector (i.e., the endoscopic cameras) about the RCM. Therefore, it is not possible to assign an arbitrary pose to the cameras, as only 4 of the 6 space dimensions can be commanded. In this perspective, a user-enabled switching mechanism has been developed to alternatively command: (i) the orientation of the cameras, through the three revolute joints of the arm; (ii) the position along the longitudinal axis of the arm, corresponding to the z-axis of the camera frame  $\mathcal{C}$ , through the prismatic joint (as shown in Fig. 6). Specifically, by querying the Oculus SDK, we first extract the 6D velocity vector  $\mathcal{P}v_{\mathcal{R}}$ , denoting the linear and angular velocity of the Oculus display expressed in  $\mathcal{F}_{\mathcal{T}}$ . Applying the proper rotation, we then generate the velocity vector  $\mathcal{R}v_{\mathcal{R}}$  expressed in its own frame  $\mathcal{F}_{\mathcal{R}}$ . Finally, we extract the linear velocity component  $\mathcal{R}v_{\mathcal{R},z}$  and the angular velocity vector  $\mathcal{R}\omega_{\mathcal{R}}$ . To require that the ECM cameras move according to the velocities commanded by the motion of the Oculus display, we then set  $\mathcal{C}v_{\mathcal{C},z} = \mathcal{R}v_{\mathcal{R},z}$  and  $\mathcal{C}\omega_{\mathcal{C}} = \mathcal{R}\omega_{\mathcal{R}}$ . Denoting by  $J = [J_v, J_\omega]^T$

the  $6 \times 4$  Jacobian matrix of the ECM, where the linear and angular contributions  $J_V = [J_{v_x}, J_{v_y}, J_{v_z}]^T$  and  $J_\omega = [J_{\omega_x}, J_{\omega_y}, J_{\omega_z}]^T$  are highlighted, we designed a de-coupled control of the position and the orientation of the ECM cameras as follows:

$$\begin{cases} \dot{q}_{1,2,4} = J_\omega^\# {}^C \omega_C & , \text{if orientation control enabled} \\ \dot{q}_3 = {}^C v_{C,z} & , \text{if position control enabled} \end{cases} \quad (1)$$

being  $\dot{q}_{1,2,4} = [\dot{q}_1, \dot{q}_2, \dot{q}_4]^T$  the vector of the revolute joint velocities and  $\dot{q}_3$  the prismatic joint velocity, while  $J_\omega^\#$  denotes the pseudo-inverse matrix of  $J_\omega$ . We assume that the user can choose which control has to be enabled, through a keyboard input. When choosing a given control, the joint velocities involved in the unused scheme are set to 0. The values of joint velocities, computed this way, are used to directly command the joints of the ECM in the V-REP environment.

### 3.2 Connecting the Geomagic Touch devices

Each Geomagic Touch device is a 6DoF haptic interface equipped with joint encoders that measure the full 6D pose of the Haptic Interface Point (HIP) of the stylus held by the user (see Fig. 9b). The device also provides a 3-DoF force feedback, allowing the user to experience a virtual sense of touch and manipulate virtual objects or reproduce physical contacts of tele-operated objects.

The velocity vectors of the PSM end-effector (i.e., the gripper) and the HIP of the corresponding haptic device have to be kept consistent, to properly teleoperate the PSMs through the movements of the stylus. In detail, with reference to Fig. 9b and for each haptic device, we query the *OpenHaptics* library to extract the current linear and angular velocity  ${}^{BG} v_H$  of the HIP, expressed in base reference frame  $F_{BG}$ , and generate the velocity  ${}^H v_H$  by applying a rotation. Moreover, to require that the PSMs grippers move accordingly and be consistent with the gripper frame  $F_g$ , we set  ${}^g v_g = {}^H v_H$ .

To determine the desired joint velocity  $\dot{q}_p$  of the considered PSM, we compute the corresponding  $6 \times 6$  Jacobian matrix  $J_g$  in the gripper frame  $F_g$ , reconstructed from the DH Tables shown in Fig. 5. Therefore, the teleoperation of the PSM through the haptic device is achieved as

$$\dot{q}_p = J_g^{-1} {}^g v_g. \quad (2)$$

A typical issue in teleoperation tasks is the geometrical heterogeneity (usually referred to as *kinematic dissimilarity*) between master and slave workspaces (i.e., the haptic device and the PSM of the da Vinci system, respectively). Specifically, the Geomagic Touch has a limited workspace, due to the short length of the links and the finite positional ranges of the joints. However, the size and the kinematic chain of the PSM is different, thus also the corresponding workspace in which the end-effector (i.e., the gripper of the PSM) moves is distinct. A common workaround that handles this discrepancy considers the use of a *clutch-based* mechanism to enable/disable the tele-operation of the slave with the master device upon explicit command of the user. This way, when the HIP of the Geomagic Touch has reached the

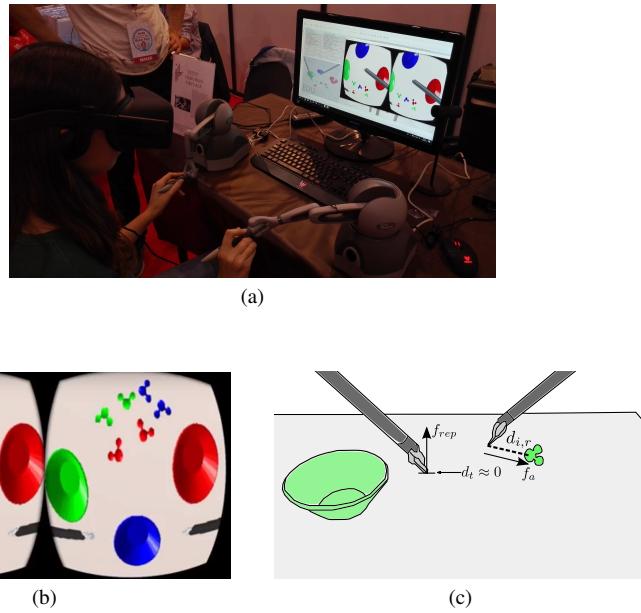


Figure 10

(a) A user interfacing with the da Vinci simulator to accomplish a training task. (b) The view seen by the user through the Oculus display. (c) A representation of the implemented repulsive force simulating the contact with the virtual table in the scene.

workspace limits of the device, the tele-operation can be disabled and the user can purposely relocate the stylus in a more favorable configuration to further move the HIP in the desired direction. This behaviour is implemented through one of the buttons mounted on the stylus of the Geomagic Touch.

Fig. 10a shows a user involved in a training task, built in the da Vinci simulator, through the use of the Oculus Rift display and the Geomagic Touch devices. The views on the two vision channels of the Oculus display is shown in Fig. 10b. The virtual scene considers a table and a set of object that can be grasped. Additional details about this training application scene are provided in Sect. 4. For demonstration purposes, we implemented a repulsive force  $f_{rep}$  on the planar surface of the table, to give the user the tactile experience of a contact of the gripper with a highly rigid object (see Fig. 10c). Future developments and improvements on the presented simulator will consider more complex dynamic interactions of the tele-operated grippers with the virtual objects in the scene. An exhaustive video showing the effectiveness of this application is available at the following link: <http://www.diag.uniroma1.it/~labrob/research/portable-DVRK.html>.

Finally, it is worth noticing that the simulation refresh rate of 220Hz, set for virtual joints and objects in the scene (see Section 2.1), is an appropriate value to render the haptic sensation of the user. In fact, human sensitivity to haptic experiences or pressure changes allows to feel regular vibrations from 200Hz to 500Hz [18] [19]. This is an operating condition satisfied by the Geomagic Touch devices, whose up-

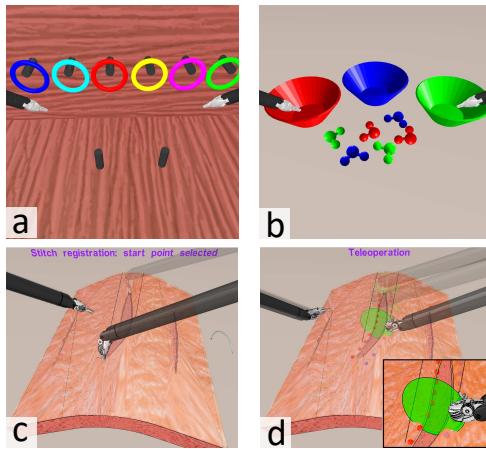


Figure 11

Training and suturing simulated environments: (a) peg on board; (b) pick and place; (c) augmented reality wound registration; (d) stitches selection and semi-autonomous execution.

date rate of the rendered forces can be set up to 1000Hz, thus providing an accurate and immediate response to the user.

## 4 Other Applications of the Simulator

Here, we introduce other potential applications of the proposed simulator:

- Training: the simulated robot is capable of interacting with dynamic rigid objects, thus is prone to be used for training purposes. Two example scenes are provided on this line.
- Augmented reality: it is possible to overlay additional information inside the simulated environment. An assisted suturing scene has been developed to show this possibility with an example of semi-autonomous task execution;
- Vision: the simulator can be used for advanced vision-based algorithms testing. A needle tracking and a visual servoing scene, in which the simulated vision system is exploited, are provided.

### 4.1 Training

Surgeons' training is an utmost requirement for an effective use of the daVinci system in real surgical scenarios. Most of the training time is spent in simulated environments. Simulators with embedded training modules are provided by Intuitive Surgical (see Sect. 1) for skills evaluation and enhancement. The training phase is essential to assess surgeon skills using scores information. However, engineers using the dVRK for research purposes do not have direct access to these costly simulators. To speed up development and testing of novel control strategies, engineers might need to equally train themselves on the (possibly simulated) dVRK system.

To this end, two V-REP scenes have been developed for non-surgical training tasks, namely: *pick & place*, and *peg on board*. Thanks to the high V-REP versatility, these

tasks are easily modifiable through the GUI, even from a non expert user. Fig. 11 contains some snapshots of these scenes as seen from the ECM camera.

The scenes have been realized by importing the CAD models of the setup into the simulated environment. V-REP has collision detection and response functionalities thus allowing simulating interactions and contacts among objects. Imported parts can be used to generate collidable, simplified meshes (different from the rendered ones) for fast simulation performed by the underlying dynamic engine. Moreover, it is possible to simulate simplified objects grasping by embedding a proximity sensor between the needle driver pads used to detect object proximity. We extensively tested the simulator training capabilities by connecting the simulated environment to real MTMs though the architecture presented in Sect. 2.1. It is anyway possible to interface the simulator with other haptic devices as shown in Sect. 3.

## 4.2 Suturing

In this section, we propose an example of a suturing scene realized in our simulator. Suturing represents an important topic in minimally invasive surgery, mainly because some of the subtasks required to complete a suturing procedure can be automatized to reduce the time and improve the results for the patients. Replicating this task in simulation can help the engineer in the development of algorithms for suturing reducing his/her effort. Moreover, the use of simulators can be useful to evaluate the surgeon's skills and to give back advantageous information to the surgeon in augmented reality. The scene developed is composed by a branch-top suturing phantom that takes inspiration from commercial phantoms and a needle SH-Plus that can be easily grasped in position using the dVRK tools grippers. The grasping control needle has been developed using a proximity sensor available in the V-REP sensors list, integrated between the gripper fingers. In detail, the needle is grasped when the proximity sensors identify the needle inside the gripper fingers and the "close the gripper" action is sent to the simulator. In this scene, the position of the needle has been obtained using the *simGetObjectPosition* function but can also be obtained using visual techniques, as described in the next section. Moreover, we include in the scene some objects: (i) colored spherical *drawing objects* to highlight the insertion (blue) and extraction (red) points; (ii) a semitransparent disk with radius equal to the needle one indicating the stitch path to follow; (iii) text messages to give back to the user information about the current control state (see Fig. 11). We include all these objects directly using the V-REP GUI and custom scripts functions. Moreover, each model is controllable from ROS topic. In detail, the spatial position, color and number of all the drawing objects is controlled using a custom topic message; a *geometry\_msgs::Pose* has been used to send position and orientation of the optimal path disk; an *std\_msgs::String* has been used to control the text messages.

## 4.3 Tracking of a suturing needle

Among the common procedures executed by surgeons, suturing is particularly challenging, due to the high dexterity demanded in a typically restricted workspace. This makes the procedure tiring for the surgeon, as the performance can be affected by his conditions and fatigue. Therefore, to increase the degree of autonomy and

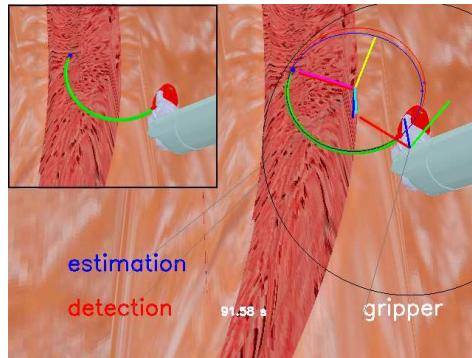


Figure 12  
Simulated setup and image processing steps, with focus on the needle (top left).

accuracy in the procedure itself, developing novel robot-assisted strategies becomes necessary. In particular, the employment of a surgical manipulator eases to design specific behaviours satisfying desirable properties, e.g., minimizing the stress on the patient’s tissue. This can be achieved by implementing proper control strategies, leading the needle held by the manipulator, to a reference configuration. To achieve this task, the availability of the pose of the instrument in the workspace is mandatory. Reconstructing such information is not trivial, since even when the needle is grasped, the only robot kinematics is not sufficient. Indeed, the grasping point on the needle surface is not unique, and external forces or slippages alter its relative pose with respect to the end-effector.

A possible solution to this problem considers a fusion of heterogeneous sensory data. While several methods have been proposed in literature, in the setup determined by our simulator we aim at designing a simplified vision-based needle tracking scheme, by using the visual information acquired by the cameras of the ECM and the kinematics data of the PSMs. This choice is motivated by the high-rate information of the joint encoders, and by the possibility to capture external disturbances, that can modify the pose of the needle, through camera images. The data are fused through an Extended Kalman Filter (EKF) to reconstruct the 6D needle pose [20], during the suturing procedure. The filter first builds an intermediate estimation of the pose, by reconstructing the velocity of the PSM gripper through differential kinematics, and assuming the needle rigidly linked to the end-effector, so that the pose of the needle can be reconstructed through velocity transformation. Nonetheless, the needle is not rigidly linked and external forces (e.g., interaction with tissue, slippages) can affect both position and orientation. Therefore, we process the camera images to detect the elliptical projection of the needle and extract a suitable visual measurement to correct the prediction in the update step of the filter. In particular, the ellipse detection is achieved through a simple RGB-based iamge segmentation, applied on a gripper-centered circular Region Of Interest (ROI), whose radius represents a projection of the spherical region of all the possible needle configurations. The set of pixels resulting from the segmentation are used to fit the corresponding ellipse on the image plane, through least-square estimation. Mathematical observations finally allow to reconstruct a measurement of the 6D pose of the needle from

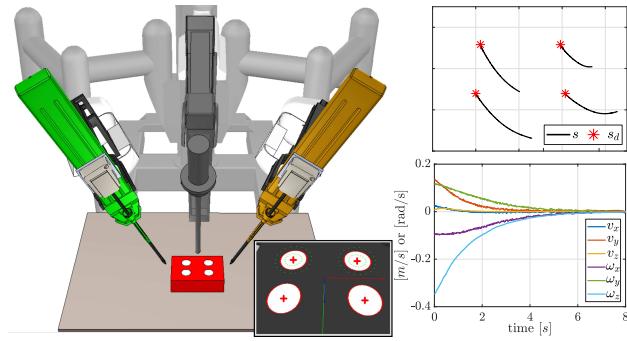


Figure 13

Visual servoing scene setup. Right-top: regulation of features on the image plane; right-bottom: 6D camera velocity converging to 0 as the desired features are approaching.

the ellipse, as explained in [21]. In the considered simulated scene (see top left view in Fig. 12), we adopted a green-colored needle along a blue tip, enforcing the vision-based pose reconstruction. The main picture in Fig. 12 shows some of the image processing entities used for the tracking: the black circle represents the ROI, the vision-based detected ellipse and the projected resulting estimation have red and blue color, respectively. The Figure also shows the corresponding reference frames

#### 4.4 Visual servoing

A visual servoing control scheme, validating the correctness of the simulated ECM, is presented. In particular, we aim at showing an Image-Based Visual Servoing (IBVS) for camera regulation, where one of the cameras of the ECM is controlled through inverse differential kinematics to regulate proper features on the image plane. For demonstrative purposes, we consider a red box with four white circles drawn on the top surface (see Fig. 13). The centroids of the circles are extracted with a blob tracker implemented in vision-based robot control software VISP [22], and used as image features to regulate. Defining a set of desired image coordinates, plots in Fig. 13 show that the circles centroids are successfully regulated through the IBVS scheme [23], where the 6D velocity of the camera is transformed to the joint velocity vector through the classical projected gradient control [24].

### 5 Integration with Advanced Instruments

The versatility of the proposed V-REP simulator allows including advanced robots and instruments in a powerful and easy way. In this section we discuss about the integration in the simulator of two novel surgical instruments we designed at the ICAROS center University of Naples Federico II. The possibility to simulate novel instruments improves the design work flow and provides the opportunity to test their performance with the help of surgeons before building them.

#### 5.0.1 The MUSHA Hand

The first instrument is our novel MUSHA hand (MH) (see Fig. 14, on the left). MH is a tree fingered under-actuated and miniaturized hand specifically designed for robotic laparoscopic surgery [25]. The hand aims at completely changing the

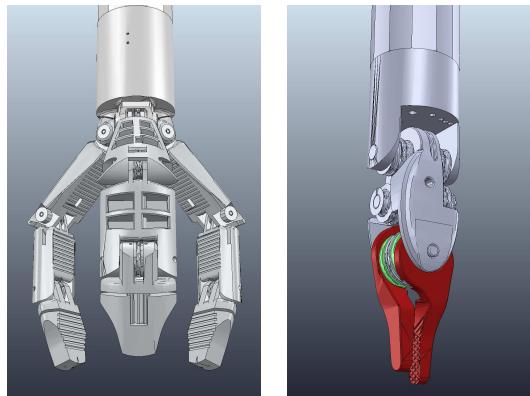


Figure 14

Advanced surgical instruments. Left: the MUSHA hand, Right: a novel needle driver with in-hand rolling capabilities.

surgical work-flow in MIRS by introducing advanced manipulation and sensing capabilities comparable to those of the human hand. The main purpose of MH is to gently interact with deformable organs to retract, manipulate and dissect them. As will be discussed in the next section, the integration of V-REP with Bullet physics engine allows evaluating the interaction between MH and a soft object.

### 5.0.2 A Novel Surgical Instrument With in-hand Rolling Capabilities

The second instrument is a new surgical needle driver that aims to reduce the surgeon mental and physical workload during difficult tasks such as suturing [5]. This instrument is provided with the ability to rotate in-hand the suturing needle to find the optimal orientation before the execution of each stitch. By adding an additional DoF to a standard needle driver tool we allow in-hand manipulation capabilities as those found in human hand during open surgery. We have integrated this advanced instrument (see Fig. 14, on the right) in the simulator for preliminary tests of the in-hand rolling capabilities as reported in [5]. In this case, the simulator has been used both for design optimization and to run a case study simulation to evaluate the percentage of cases in which this novel tool could be helpful in real suturing trajectories.

## 5.1 Integration with Other Physics Engines

In this section, we present a simulation that aims at evaluating the potential of our novel MUSHA Hand, with respect to classical tools, in selected tasks performed during adrenalectomy and colectomy procedures. The simulated tasks are organs mobilization, grasping, and measurements of critical dimensions of affected organs. Since the MH is still a prototype, not ready to be used in surgical environments, the evaluation of the conceptual design is obtained in a simulated environment by replicating qualitatively the task execution of standard laparoscopic tools in real environments. Fig. 15 presents the simulation environment. In more details, we exploit the possibility to integrate V-REP with other physics engine to extend its simulation potentialities. Bullet physics is chosen thanks to its ability to simulate

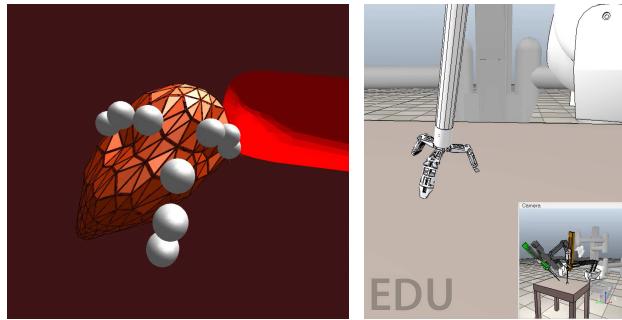


Figure 15

Simulation environment of the MH mounted on the dVRK robotic instrument. Left: Bullet physics scene containing the MH collision meshes (white spheres) and a deformable body representing the adrenal gland (orange); Right: V-REP scene of MH integrated with the dVRK robotic instrument.

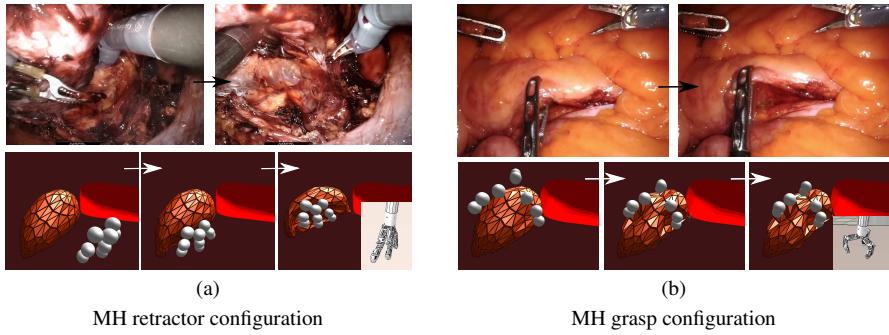


Figure 16

MH use cases: (a) retractor configuration to mobilize and lift organs during a laparoscopic adrenalectomy procedure; (b) grasp configuration to grab and pull organs during a colectomy procedure. Top: real surgical procedure, Bottom: simulated environment.

soft objects e.g. organs and tissues. The soft organ is simulated using a soft triangular mesh shape with elastic properties [26]. The organ has been anchored to the rigid scene in different points to simulate the interaction between the organ and the abdominal surfaces. The MUSHA hand collision model has been realized using three spheres for each finger simulating the hand phalanges. We have linked the bullet simulated scene to our VREP simulator through remote API functions to have at each time step the position of each hand phalanges w.r.t. the robot Remote Center of Motion (RCM). In Figs. 16a, 16b and 17, three different simulated scenes have been proposed where MH has been used in: (i) retractor configuration, to lift and mobilize organs; (ii) grasping configuration, to grab organs or tissues; (iii) caliper configuration to measure organs dimensions.

## 6 Discussion and Conclusions

In this work, a simulator of the full dVRK integrated in V-REP has been presented. The kinematics of the dVRK arms has been described and implemented in the sim-

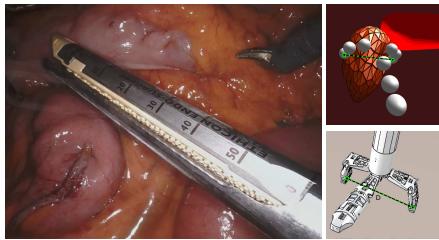


Figure 17

MH use case: caliper simulation. The hand direct kinematics can be used to estimate organs or tissues critical dimensions. Left: surgical scenario; Right top: bullet simulation; Right bottom: V-REP simulation.

ulated robot. The integration with ROS allows controlling the simulated robot using the real dVRK master device and developing advanced control strategies. A modified ROS-independent architecture also allows the integration of commercial interfaces for haptic feedback and virtual reality applications like, e.g., the Geomagic Touch and the Oculus Rift. To show the potentialities of the proposed simulator, four different scenes ready to use have been included.

Major limitations of the proposed simulator are realistic grasping of thin-shaped objects, such as the needle, and simulation of the interaction with deformable bodies. The former problem can be opportunely circumvented by disabling collisions and attaching the object rigidly to the hand. To address the second, the bullet physics engine has been connected to V-REP to simulate the interaction of tools with soft tissues and organs. A new tool for the manipulation of soft tissues is included in the simulator. As for the future, we aim to exploit bullet engine deformable body simulation capabilities recreating the deformable bodies visualization directly in V-REP.

### Acknowledgement

This project was partially supported by the POR FESR 2014-2020 National programme within BARTOLO project CUP B41C17000090007 and by the EC Seventh Framework Programme (FP7) within RoDyMan project 320992.

### References

- [1] 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. *IEEE Int. Conf. on Robotics and Automation*, pages 6434–6439, May 2014.
- [2] M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano. Passive virtual fixtures adaptation in minimally invasive robotic surgery. *IEEE Robotics and Automation Letters*, pages 3129 – 3136, 2018.
- [3] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell. Enhancing bilateral teleoperation using camera-based online virtual fixtures generation. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1483–1488, 2016.
- [4] J. M. Prendergast and M. E. Rentschler. Towards autonomous motion control in minimally invasive robotic surgery. *Expert Review of Medical Devices*, pages 741–748, 2016.

- [5] G. A. Fontanelli, M. Selvaggio, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano. A new laparoscopic tool with in-hand rolling capabilities for needle reorientation. *IEEE Robotics and Automation Letters*, pages 2354–2361, 2018.
- [6] G. A. Fontanelli, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano. A novel force sensing integrated into the trocar for minimally invasive robotic surgery. *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 131–136, 2017.
- [7] A. Moglia, V. Ferrari, L. Morelli, M. Ferrari, F. Mosca, and A. Cuschieri. A Systematic Review of Virtual Reality Simulators for Robot-assisted Surgery. *European Urology*, pages 1065–1080, 2016.
- [8] A. Baheti, S. Seshadri, A. Kumar, G. Srimathveeravalli, T. Kesavadas, and K. Guru. Ross: Virtual reality robotic surgical simulator for the da vinci surgical system. *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 479–480, 2008.
- [9] J. A. Sanchez-Margallo, J. P. Carrasco, L. Sanchez-Peralta, J. L. M. Cuevas, L. Gasperotti, D. Zerbato, and F. S.-M. L. Vezzaro. A preliminary validation of the xron surgical simulator for robotic surgery. *Int. Conf. of the Society for Medical Innovation and Technology*, 2013.
- [10] R. Smith, M. Truong, and M. Perez. Comparative analysis of the functionality of simulators of the da vinci surgical robot. *Surgical Endoscopy*, (4):972–983, Apr 2015.
- [11] A. Munawar and G. Fischer. Towards a haptic feedback framework for multi-dof robotic laparoscopic surgery platforms. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1113–1118, Oct 2016.
- [12] E. Rohmer, S. P. N. Singh, and M. Freese. Vrep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, Nov 2013.
- [13] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield. Feature and performance comparison of the v-rep, gazebo and argos robot simulators. In M. Giuliani, T. Assaf, and M. E. Giannaccini, editors, *Towards Autonomous Robotic Systems*, pages 357–368, Cham, 2018. Springer International Publishing.
- [14] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano. Modelling and identification of the da Vinci research kit robotic arms. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1464–1469, 2017.
- [15] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides. Software architecture of the da vinci research kit. *IEEE Int. Conf. on Robotic Computing*, pages 180–187, 2017.
- [16] G. S. Guthart and J. K. Salisbury. The intuitive<sup>TM</sup> telesurgery system: overview and application. *IEEE Int. Conf. on Robotics and Automation*, pages 618–621, 2000.
- [17] M. Ferro, D. Brunori, F. Magistri, L. Saiella, M. Selvaggio, and G. A. Fontanelli. A portable da vinci simulator in virtual reality. *Third IEEE International Conference on Robotic Computing (IRC 2019)*, pages 447–448, 2019.

- [18] J. Scheibert, S. Leurent, A. Prevost, and G. Debrégeas. The role of fingerprints in the coding of tactile information probed with a biomimetic sensor. *Science*, 323(5920):1503–1506, 2009.
- [19] L. Skedung, M. Arvidsson, J. Y. Chung, C. M. Stafford, B. Berglund, and M. W. Rutland. Feeling small: exploring the tactile perception limits. *Scientific reports*, 3:2617, 2013.
- [20] M. Ferro, G. A. Fontanelli, F. Ficuciello, B. Siciliano, and M. Vendittelli. Vision-based suturing needle tracking with extended kalman filter. *Computer/Robot Assisted Surgery workshop*, 2017.
- [21] D. López de Ipiña, P. R. S. Mendonça, and A. Hopper. Trip: A low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing*, 6(3):206–219, 2002.
- [22] E. Marchand, F. Spindler, and F. Chaumette. Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotic and Automation Magazine*, pages 40–52, 2005.
- [23] F. Chaumette and S. Hutchinson. Visual servo control, part i: Basic approaches. *IEEE Robotic and Automation Magazine*, pages 82–90, 2006.
- [24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer-Verlag London, 2009.
- [25] M. Selvaggio, G. A. Fontanelli, V. R. Marrazzo, U. Bracale, A. Irace, G. Breglio, L. Villani, B. Siciliano, and F. Ficuciello. The musha underactuated hand for robot-aided minimally invasive surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 15(3):e1981, 2019. e1981 rcs.1981.
- [26] F. Fazioli, F. Ficuciello, G. A. Fontanelli, B. Siciliano, and L. Villani. Implementation of a soft-rigid collision detection algorithm in an open-source engine for surgical realistic simulation. *IEEE Int. Conf. on Robotics and Biomimetics*, pages 2204–2208, 2016.