A V-REP Simulator for the da Vinci Research Kit Robotic Platform

G. A. Fontanelli[1], M. Selvaggio[1], M. Ferro[2], F. Ficuciello[1], M. Vendittelli[3] and B. Siciliano[1]

***Abstract*— In this work we present a V-REP simulator for the da Vinci Research Kit (dVRK). The simulator contains a full robot kinematic model and integrated sensors. A robot operating system (ROS) interface has been created for easy use and development of common software components. Moreover, several scenes have been implemented to illustrate the performance and potentiality of the developed simulator.**
Both the simulator and the example scenes are available to the community as an open source software.

I. Introduction

The da Vinci Research Kit (dVRK) [1] is an open-source mechatronic platform obtained from the first-generation of Intuitive Surgical System and provided with controllers and software developed at Johns Hopkins University LCSR and Worcester Polytechnic Institute AIM Lab [2]. The research community sharing the dVRK is composed by over 30 research institutions across the world, which provides an idea of the importance of this open platform in technology-oriented research for robot-aided surgery. By default the daVinci is a master-slave robot scaling the motion, attenuating tremor and enhancing precision of the surgeon that remotely control the system. Despite teleoperation is the primary control mode, sensor-based shared automated control of robot trajectories is under development to augment surgeon abilities [3]. Thus, the open research platform is suitable to enhance both research in haptic teleoperation [4] and in semi-autonomus control [5]. In the last two decades, surgical simulation has strongly widespread thanks to the progress in robotic surgery [6]. Simulation and virtual reality support different research fields from industry to entertainment up to surgical robotics. In the surgical field, simulators are mainly developed for training to allow surgeons acquiring basic robotic skills as well as more complex maneuvers before performing live surgery. In the face of cost/effectiveness ratio, simulators are a solution to allow students learning the base techniques for robot-assisted surgery. The state of the art of simulators currently available is constituted by: *Robotic Surgery Simulator (RoSS)* [7], *SimSurgery Education Platform (SEP)* [8], *da Vinci Trainer* [9], *da Vinci Skills Simulator* [10], *Robotix Mentor* [11] and *Chiron* [12]. In [13] a comparative evaluation of some of these simulators is provided to help users in selecting an appropriate device for their needs. Besides the training capabilities of each

[1]Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università di Napoli Federico II giuseppeandrea.fontanelli@unina.it
[2]Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma
[3]Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni, Sapienza Università di Roma

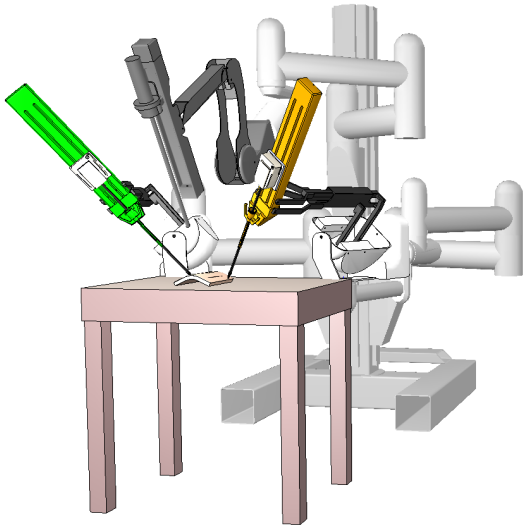

Fig. 1. The da Vinci Research Kit V-REP simulator.

system, all of them provide modules for EndoWrist manipulation, camera control, needle control and clutching, and a realistic representation of the da Vinci workspace. The described simulators are designed specifically for surgeons' training and do not provide a virtual reality simulator of the whole robot kinematics, namely Setup joints (SUJ), Patient Side Manipulators (PSMs), Endoscopic Camera Manipulator (ECM). On the other hand, simulating the robot is a solution providing a low cost and easy to access environment for the development and proof of new control strategies while minimizing the risk of testing new algorithms on such a complex system. Furthermore, it is a valuable tool for safely testing out new technology, e.g. new surgical tools and sensors [14], [15]. In a wide community sharing the dVRK a free simulator is a benefit for the progress of the research whereas the uniqueness of the robot, received as a donation from Intuitive Surgical, and the difficulty of replacing components in case of malfunctioning. Furthermore, a simulator will also profit educational purposes as it enables students to easily approach the system and work in total safety. A brief description of the most advanced open source software packages for robotic simulation can be found in [16]. The use of open source robot simulators allows creating a virtual model of a robot, simulating components and sensors and testing new tools design, control strategies and integrating learning in a simulation environments. In this paper, the dVRK simulator is realized using V-REP [17]. The choice is motivated by the versatility and simplicity of this software for multi-robot applications. V-REP is based on a distributed control architecture. 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. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave. Therefore, the simulator can be easily interfaced with the real surgeon master console, and new objects and robots can be imported in the scene by using a graphical interface. The developed simulator includes the kinematics of the SUJ, PSMs, ECM and the camera sensor and it is interfaced with the ROS framework. Moreover, four scenes are already created and ready for use.
The complete simulator, together with the four developed application scenes, is available at https://github.com/unina-icaros/dvrk-vrep.git.

The rest of the paper is organized as follows: In Sect. II the kinematics of the robotic arms included in the simulator are briefly described; in Sect. III the V-REP models are reported focusing on the simulated scene and on the simulator performance; Section IV describes the control architecture focusing on the ROS based infrastructure; in Sect. V we discuss four different scenes developed in the simulated environment as proof of concept to show the potentialities of the proposed simulator, while, Sect. VI concludes the paper.

II. The da Vinci Surgical System Kinematic Model

The full dVRK is a first-generation da Vinci Surgical System consisting of two/three PSMs, one ECM, and two Master Side Manipulators (MTMs). The slave side manipulators are mounted on a SUJ that allows the manual spatial positioning of the arm bases. We include in the simulated environment the patient side manipulators composed of two PSMs and an ECM mounted on the SUJ. In the next sections, a brief description of the arms kinematics is reported.

A. Setup Joints arm kinematics

The two PSMs and the ECM are mounted on the SUJ, an articulated robotic structure composed by three or, in the newest versions, four arms. The two PSMs are located at the end of two 6-degree-of-freedom (DoFs) arms (that we indicate hereafter as SUJ-PSMs) while the ECM is located at the end of a 4-DoFs arm (SUJ-ECM). All the robotic arms in the SUJ are not actuated by motors but it is possible to control breaks in each joint and read the angular position using potentiometers [1]. Denoting with $\boldsymbol{q}_{sp} = [q_{sp,1}, \ldots, q_{sp,6}]$ the vector of the SUJ-PSMs arms generalized coordinates, the homogeneous transformation matrix[1] $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AP}}(\boldsymbol{q}_{sp}) \in SE(3)$, representing the pose of the SUJ-PSMs end-effector frame $\mathcal{AP} : \{\boldsymbol{O}_{ap}; \boldsymbol{x}_{ap}, \boldsymbol{y}_{ap}, \boldsymbol{z}_{ap}\}$ with respect to the base frame $\mathcal{B} : \{\boldsymbol{O}_b; \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, can be easily computed applying the standard DH convention to the kinematic chain $\{J_1, \ldots, J_6\}$ of Fig. 2 (see Table I where $a_2 = 0.58$m, $a_3 = 0.56$m and $d_4 = 0.425$m). Moreover, denoting with $\boldsymbol{q}_{se} = [q_{se,1}, \ldots, q_{se,4}]$ the vector of the SUJ-ECM arm generalized coordinates, the pose of the SUJ-ECM end-effector frame $\mathcal{AE} : \{\boldsymbol{O}_{ae}; \boldsymbol{x}_{ae}, \boldsymbol{y}_{ae}, \boldsymbol{z}_{ae}\}$ with respect to the base frame $\mathcal{B} : \{\boldsymbol{O}_b; \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, defined by the homogeneous transformation matrix $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AE}}(\boldsymbol{q}_{se}) \in SE(3)$, can be computed considering only the first four rows of Table I. Notice that, two constant homogeneous transformation matrices $\boldsymbol{T}^{\mathcal{AP}}_{\mathcal{BP}} \in SE(3)$ and $\boldsymbol{T}^{\mathcal{AE}}_{\mathcal{BE}} \in SE(3)$ must be considered to complete che kinematics description, providing the transformation between $\mathcal{AP}$ and $\mathcal{AE}$ (respectively the last SUJ-PSM and SUJ-ECM frames) and the base frames $\mathcal{BP}$ and $\mathcal{BE}$ of the PSMs and of the ECM described in Sec. II-B and II-C (see Fig. 2).

[1]Hereafter, we use the matrix notation $\boldsymbol{T}^a_b$, where the superscript a denotes the frame in which vector components are expressed, the subscript b the current frame. *E.g.*, $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AP}}$ denotes the pose of the SUJ-PSM attach point expressed in the base frame.

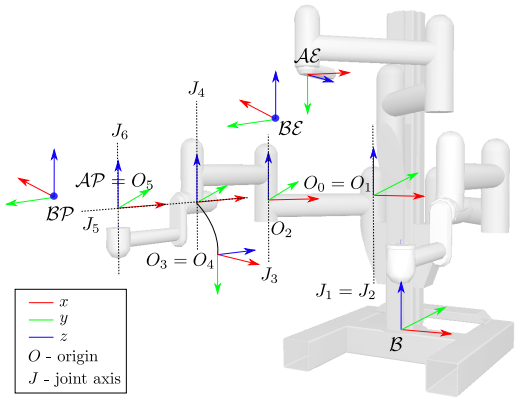

Fig. 2. SUJ kinematic description.

TABLE I
DH Parameters of the SUJ

link	joint	a_i	α_i	d_i	θ_i
1	P	0	0	$q_{se,1}$	–
2	R	a_2	0	–	$q_{se,2}$
3	R	a_3	0	–	$q_{se,3}$
4	R	0	$-\pi/2$	–	$q_{se,4}$
5	R	0	$\pi/2$	$-d_4$	$q_{se,5}$
6	R	0	0	–	$q_{se,6}$

TABLE II
DH Parameters of the PSM

link	joint	a_i	α_i	d_i	θ_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	$-\pi/2$	–	$q_{p,6}$

B. PSM arm kinematics

Each PSM is a 7-DoF actuated arm, which moves a surgical instrument about a Remote Center of Motion (RCM), *i.e.*, a fixed fulcrum point that is invariant with respect to the configuration of the PSM joints [18], [19]. The first 6 DoFs correspond to Revolute (R) or Prismatic (P) joints, combined in a RRPRRR sequence. The last DoF corresponds to the opening and closing motion of the gripper. The homogeneous transformation matrix $\boldsymbol{T}^{\mathcal{BP}}_{\mathcal{G}}(\boldsymbol{q}_p) \in SE(3)$ (where $\boldsymbol{q}_p = [q_{p,1}, \ldots, q_{p,6}]$ is the vector of the PSM generalized coordinates), representing the pose of the gripper frame $\mathcal{G} : \{\boldsymbol{O}_g; \boldsymbol{x}_g, \boldsymbol{y}_g, \boldsymbol{z}_g\}$ with respect to the base frame $\mathcal{BP} : \{\boldsymbol{O}_{bp}; \boldsymbol{x}_{bp}, \boldsymbol{y}_{bp}, \boldsymbol{z}_{bp}\}$, can be easily computed by choosing the origin of frame $\mathcal{BP}$ in the RCM point and applying the

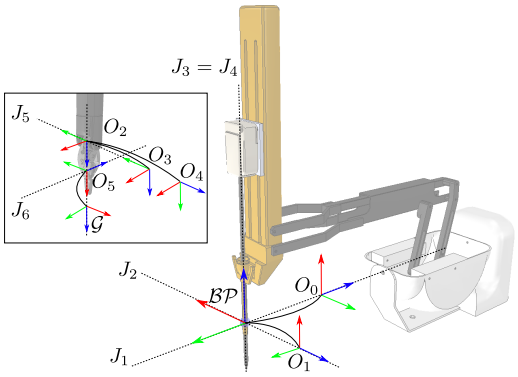

Fig. 3. PSM kinematic description.

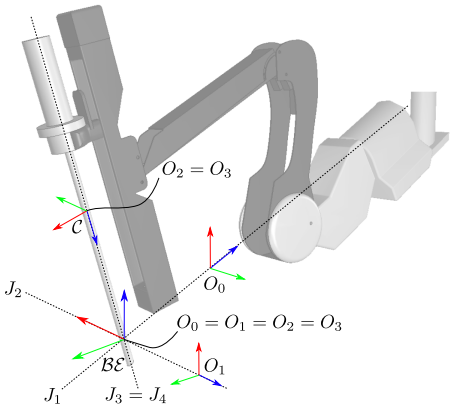

Fig. 4. ECM kinematic description.

standard DH convention to the kinematic chain $\{J_1, \ldots, J_6\}$ of Fig. 3 (see Table II, where $a_5 = 0.0091$ m).

C. ECM arm kinematics

The ECM is a 4-DoF actuated arm, which moves the endoscopic camera about the RCM through revolute and prismatic joints, combined in a RRPR sequence. The homogeneous transformation matrix $\boldsymbol{T}_{\mathcal{C}}^{\mathcal{BC}}(\boldsymbol{q}_e) \in SE(3)$ (where $\boldsymbol{q}_e = [q_{e,1}, \ldots, q_{e,4}]$), representing the pose of the camera frame $\mathcal{C} = \{\boldsymbol{O}_c; \boldsymbol{x}_c, \boldsymbol{y}_c, \boldsymbol{z}_c\}$ with respect to the base frame $\mathcal{BC} = \{\boldsymbol{O}_{bc}; \boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}, \boldsymbol{z}_{bc}\}$, can be easily computed by choosing the origin of frame $\mathcal{CB}$ in the RCM point and applying the standard DH convention to the kinematic chain $\{J_1, \ldots, J_4\}$ of Fig. 4 (parameters are given in Table III, where $d_4 = 0.007$m).

TABLE III
DH PARAMETERS OF THE ECM

link	joint	a_i	α_i	d_i	θ_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	$q_{e,4}$

III. V-REP MODEL

In this section, the simulator is described focusing on the robot structure and on the general performances. With reference to Fig. 1, our V-REP simulator is composed of a SUJ, two PSMs and one ECM. The robotic arms have been modeled starting from the CAD models included in the John Hopkins dVRK git webpage[2], except for the SUJ. Each robot link has been realized by including two type of mesh: (*i*) one visual mesh with structure and texture similar to the real robot link, (*ii*) one simplified convex dynamic and respondable mesh used to simulate dynamics and contacts[3]. With reference to Sect. II we realized the kinematic chain of each robotic arm by linking mesh and joints in a *joint-respondable-visual* sequence. For each respondable link of the two PSMs we included the dynamic parameters obtained by identification, as described in [19]. At the end of the endoscope link, two cameras have been included to simulate the binocular vision system of the real dVRK endoscope. We set a resolution for the cameras at 320×288 pixels, *i.e.*, half the resolution of the real endoscope, that results a trade-off option to have a good resolution and a good simulated sampling time.

The resulting complete robot is composed of 10178 triangles. Hence, considering a computer powered by a Intel I7-7770HQ processor, 16GB of ram and Nvidia GeForce 960M the scene is rendered at 45 fps and the dynamics is simulated at 200 Hz.

IV. CONTROL ARCHITECTURE

We designed the proposed V-REP simulator to be fully integrated into the dVRK control infrastructure. Hence, the high-level ROS framework has been used to link our simulator to the low-level control [2]. This allows the user to use the simulator in different modalities: (*i*) telemanipulated using the dVRK MTMs; (*ii*) in combination with the real robotic PSMs and ECM, to implement augmented reality 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

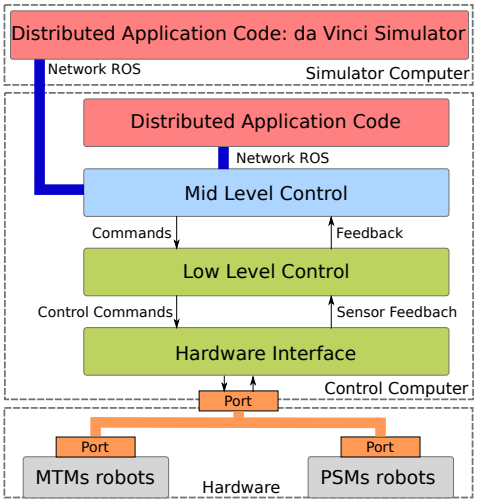

Fig. 5. Software architecture.

[2]https://github.com/jhu-dvrk

[3]Dynamic respondable shapes influence each other during dynamic collisions and are subject to gravity and inertial forces.

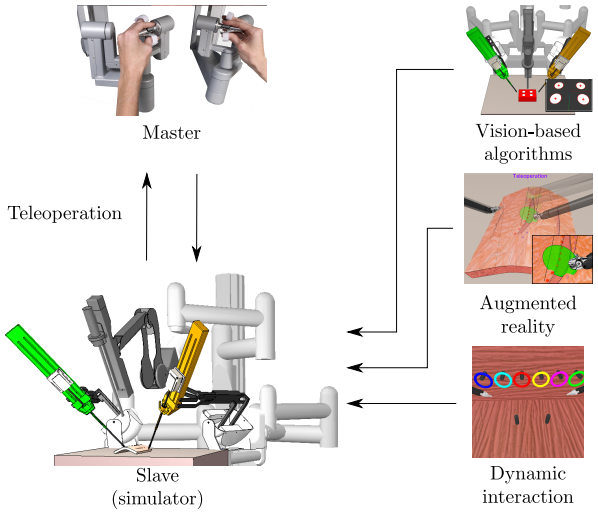

Fig. 6. Simulated environment with different application examples.

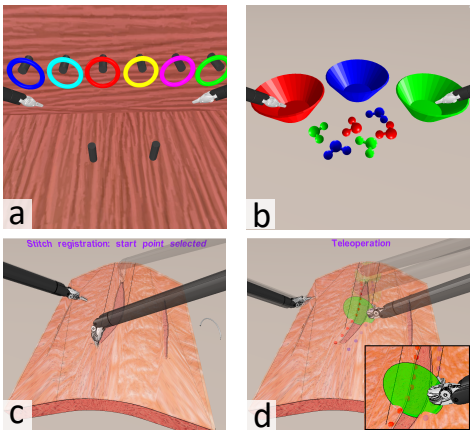

Fig. 7. Training and suturing scene setup. Upper row: two examples of training tasks, (a) peg on board; (b) pick and Place. Lower row: augmented reality suturing scene, (c) wound registration; (d) stitches planning and exececution.

the embedded scripts.

With reference to Fig. 5, the control architectures of the dVRK, described in detail in [20], is composed of: (*i*) a hardware interface to communicate with the embedded actuator controllers through the fire-wire bus, implementing the safety checks; (*ii*) a low level layer implementing all the algorithms for the inverse kinematics, impedance master control etc.; (*iii*) a mid-level layer implementing the ROS communication and the high level controllers. The communication between the da Vinci simulator, running in a dedicate computer, and the dVRK console is implemented through ROS topics. In detail, we use the *v_repExtRosInterface* to publish the state of the robot joints (PSMs, SUJ, ECM) and the gripper state for the PSMs. Moreover, the simulator subscribes to two topics *sensor_msgs::joint_state* to control the robots joints motion from ROS.

The computer configuration described in the previous section is able to stream cameras topics at 60 Hz[4]. The joints and objects topics are streamed at 220Hz.

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

V. Example scenes

The possibility to include different robots, dynamic objects, devices and sensors allows to easily extend the simulator capabilities through the creation of advanced V-REP scenes. In this work, we propose different scenes to show the potentialities addressing the implementation aspects, and representing common applications for robotic surgery research. Hence, we show the development of advanced control strategies, *e.g.*, visual servoing or vision-based object tracking, augmented reality and simulation of rigid objects dynamics and interaction (see Fig. 6).

In detail, we present:

- Two training scenes, developed to show the capability of the simulated robot interacting with rigid dynamic objects;
- A suturing scene, developed to show the possibility of integrating easily augmented reality information inside the simulated environment, e.g. to show an example of combination of autonomous and tele-operated task execution (assisted suturing);
- A needle tracking and a visual servoing scene, showing the potentiality in implementing advanced vision-based algorithms, through the information acquired from the simulated vision system.

A running simulation of each above-mentioned application can be found in the accompanying video of this work.

A. Training

In order to effectively use the dVRK in surgical scenarios, surgeons spend a huge amount of time training in simulation. Intuitive Surgical provides simulators (see Sect. I) embedding training modules for robotic skills enhancement in simple tasks and complete robotic procedures. In this context, simulation is very important since it can provide scores information about the surgeon skills. However, these simulators are costly and not completely exploitable by roboticists for research purposes. Novel engineers may equally need to train themselves to develop and test novel control strategies. To this end, we provide two V-REP scenes in which non-surgical training tasks are proposed, namely: *pick & place*, and *peg on board*. However, the high versatility of V-REP allows easy development and implementation of different training tasks and/or assistive strategies. Fig. 7 contains snapshots of the proposed scenes taken from the ECM left camera.

The scenes have been realized by developing and importing CAD models of the training setup into the scene. Contacts and interactions among objects have been simulated by creating respondable and simplified dynamic entities through the embedded V-REP functions. Moreover, a proximity sensor integrated between the needle driver pads

[4]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.

has been used to simulate objects grasping. The control architecture presented in Sect. IV allows to easily interface the simulated robot with the MTMs or other input devices (*e.g.* haptic devices).

B. Suturing

Suturing is one of the most delicate and stressful tasks in minimally invasive surgery mostly because of the reduced workspace, the high precision required, the lack of haptic perception and the complexity induced by artificial vision feedback. Hence, in the last decade different algorithms have been proposed to help the surgeon in this delicate procedure using automatic or assistive approaches [21]. In this scenario, the use of simulators can be useful and effective to test new control paradigms, to evaluate the surgeon's skills and also to provide surgeons with augmented reality information in an easy and powerful way.

In this work, we propose a scene designed for suturing in our simulated environment. A branch-top suturing phantom has been designed taking inspiration from real commercial phantoms. Moreover, some useful objects have been included in the scene (*i*) to give to the surgeon information about the insertion and extraction points for each stitch through overlayed objects; (*ii*) to give informations about the optimal stitch path using a semitransparent disk, with radius compatible with the chosen suturing needle; (*iii*) to give textual information using a banner integrated in the environment (see Fig. 7). All the above cited elements are easily included in the scene directly using the graphic interface and using custom scripts functions. Moreover, each described component is controllable from ROS via topic. In detail, a custom topic message has been used to send the spatial position, colour and number of spherical *drawing objects*; a *geometry_msgs::Pose* has been used to control the overlapped disk showing the optimal path; an *std_msgs::String* has been used to control the banner messages. Finally, a proximity sensor, integrated between the needle driver fingers, has been used to simulate the needle grasp in position, e.g., the needle is grasped when it is inside the needle driver fingers and the gripper is closed, while it is released when the gripper is opened. In this task, the needle pose w.r.t. the PSMs gripper is directly known from the *simGetObjectPosition* function included in V-REP. However, it can also be estimated using visual information, as described in the next section.

C. Needle tracking

To achieve autonomous or robot-assisted suturing with a surgical manipulator, it is necessary to know the current pose of the needle in the operating environment. This pose can be precisely estimated by fusing data coming from different sensory sources of the da Vinci System. While several methods have been proposed in literature, here our objective is to show the design of a simplified vision-based needle tracking scheme, starting from the information accessible through the simulator and the tools available in V-REP.

For this purpose, we designed an Extended Kalman Filter (EKF) to estimate the 6D pose of the needle [22], holded by one of the PSMs. The filter fuses kinematic information from the robot proprioceptive sensors with the visual information provided by one of the two cameras of the endoscope. The reasons below this choice are the typical high-rate information provided by the proprioceptive sensors, along with the capability of the camera to capture external disturbances, that can influence the pose of the needle with respect to the gripper. Based on this idea, we use the known velocity of the gripper, computed from the joint velocities of the PSM through differential kinematics, to predict an intermediate estimation of the needle pose. In this fashion, the needle and the gripper are assumed to be rigidly linked, so that the velocity of the needle can be recovered by simply transforming the velocity of the gripper. However, since the gripper-needle transformation is not rigid, a vision-based pose measurement is necessary to capture unexpected disturbances acting on the needle that can alter its pose, in order to refine such prediction. First, a vision-based detection of the projected ellipse is achieved through a preliminary RGB-segmentation performed on a circular Region Of Interest (ROI). The ROI is centered at the gripper position, and its radius delimits the spherical region where the needle is supposed to be. Then, the set of pixels resulting from the segmentation is used to robustly fit the projected ellipse of the needle on the image plane, with a least-square-based approach. Finally, the 6D needle pose measurement is recovered from the ellipse, based on geometric considerations.The simulated setup is intentionally simple, and is depicted in the top left view in Fig. 8: we considered a green-colored needle with a blue tip, to ehnance the vision-based reconstruction of the proper needle orientation. The main figure shows the image processing output of the 3D needle tracking scheme: the circular ROI is drawn in black, while the output of the vision-based detection and the projection of the estimated pose of the needle are drawn in red and blue, respectively. Finally, the projections of the reference frames of the gripper, as well as of the vision-based reconstructed and the estimated pose, are drawn superposed on the image (RGB triad for estimation and gripper, CMY triad for the measurement).

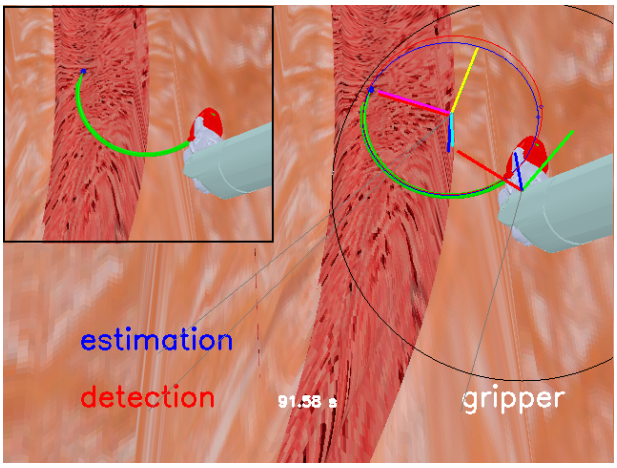

Fig. 8. Needle Tracking scene setup (top left) and image processing output.

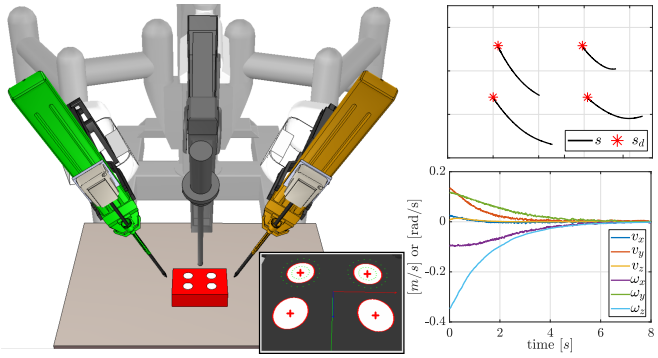

Fig. 9. Visual servoing scene setup. Right-top: image plane features; right-bottom: camera velocities.

D. Visual servoing

As an example of novel control strategies development and test, we show a visual servoing task. The task consists in autonomously regulating the pose of the ECM to track a desided object. Simulated images are streamed through customary ROS topics and are then processed to extract the needed visual features. For simplicity we used a red squared object endowed with four white circular blobs. Hence, a blob tracker has been implemented using VISP, a visual servoing software platform specifically designed for robot control [23]. The autonomous visual task aims at keeping the object of interest in a predefined image-space pose during the operation. We used a classical Image-Based Visual Servoing control scheme to keep the object blobs centers in a predefined pose on the image plane [24]. The resulting 6-dimensional camera velocity is projected in the manipulator joint space using the classical projected gradient control [25]. Fig. 9 contains a snapshot of the proposed scene and the results of the proposed autonomous camera regulation experiment.

VI. Conclusion and Future Work

In this work, a simulator of the full dVRK integrated in V-REP has been presented. The kinematic description of the dVRK arms has been provided and implemented on the simulated robot. The integration of our simulator with ROS allows controlling the robot using the real dVRK master device and developing advanced control strategies. To show the advantages and potentialities of the proposed simulator, we developed four different scenes. In future works, we will investigate the integration of deformable objects within V-REP using bullet engine [26] or SOFA [27] to extend the application scenarios.

References

[1] "da Vinci research kit (DVRK) wiki." [Online]. Available: http://research.intusurg.com/dvrkwiki

[2] 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 Int. Conf. on Robotics and Automation*, May 2014, pp. 6434–6439.

[3] 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*, 2018.

[4] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, "Enhancing bilateral teleoperation using camera-based online virtual fixtures generation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 1483–1488.

[5] J. M. Prendergast and M. E. Rentschler, "Towards autonomous motion control in minimally invasive robotic surgery," *Expert Review of Medical Devices*, vol. 13, no. 8, pp. 741–748, 2016.

[6] A. Moglia, V. Ferrari, L. Morelli, M. Ferrari, F. Mosca, and A. Cuschieri, "A Systematic Review of Virtual Reality Simulators for Robot-assisted Surgery," *Eur Urol*, pp. 1065–1080, 2016.

[7] 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," in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2008, pp. 479–480.

[8] "SimSurgery Educational Platform (SEP)." [Online]. Available: http://www.simsurgery.com

[9] "Mimic Simulation dV-Trainer." [Online]. Available: http://www.mimicsimulation.com/products/dv-trainer/

[10] "da Vinci Skills Simulator." [Online]. Available: https://www.intuitivesurgical.com/products/skills_simulator/

[11] "RobotiX Mentor Simbionix." [Online]. Available: simbionix.com/simulators/robotix-mentor/

[12] J. Sanchez-Margallo, J. P. Carrasco, L. Sanchez-Peralta, J. M. Cuevas, L. Gasperotti, D. Zerbato, and F. S.-M. L. Vezzaro, "A preliminary validation of the xron surgical simulator for robotic surgery," in *Int, Conf. of the Society for Medical Innovation and Technology*, 2013.

[13] R. Smith, M. Truong, and M. Perez, "Comparative analysis of the functionality of simulators of the da vinci surgical robot," *Surgical Endoscopy*, vol. 29, no. 4, pp. 972–983, Apr 2015.

[14] 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*, vol. 3, no. 3, pp. 2354–2361, 2018.

[15] 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," in *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 131–136.

[16] "Smashing Robotics." [Online]. Available: https://goo.gl/gH8VE4

[17] "V-REP simulator." [Online]. Available: http://www.coppeliarobotics.com/

[18] G. Guthart and J. Salisbury, "The intuitive™ telesurgery system: overview and application," in *IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 618–621.

[19] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da Vinci research kit robotic arms," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 1464–1469.

[20] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides, "Software architecture of the da vinci research kit," in *IEEE Int. Conf. on Robotic Computing*, 2017, pp. 180–187.

[21] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," *IEEE Int. Conf. on Robotics and Automation*, pp. 4178–4185, 2016.

[22] 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.

[23] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, 2005.

[24] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robot. Autom. Mag.*, pp. 82–90, 2006.

[25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer-Verlag London, 2009.

[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," in *IEEE Int. Conf. on Robotics and Biomimetics*, 2016, pp. 2204–2208.

[27] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A Multi-Model Framework for Interactive Physical Simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, ser. Studies in Mechanobiology, Tissue Engineering and Biomaterials, Y. Payan, Ed. Springer, 2012, vol. 11, pp. 283–321.