A Unity-based Da Vinci Robot Simulator for Surgical Training

Ke Fan[1], Aldo Marzullo[2], Nicolò Pasini[1], Alberto Rota[1], Matteo Pecorella[1], Giancarlo Ferrigno[1] and Elena De Momi[1]

Abstract— **The development of the Robot-Assisted Minimally Invasive Surgery (RAMIS) imposes an increasing demand for surgical training platforms, especially low-cost simulation-based surgical training through the creation of new open-source modules. For this goal, a da Vinci Surgical robot simulator based on Unity Physics Engine is developed. The simulator is integrated with da Vinci Research Kit (dVRK), robot kinematic models and multiple sensors. The Robot Operating System (ROS) interface is embedded for better integration with ROS based software components. The simulator can provide interactive information such as haptic feedback with master input devices. An application of a virtual fixture is implemented to test and verify the performance of the simulator. The results show that the simulator has high expansibility and support interactive training tasks well.**

I. INTRODUCTION

Practical training is always a necessary part of modern surgical education [1]. Besides enhancement of surgical theory, surgical training helps surgeons with psychomotor skills and experience [2]. This has become more and more important since lots of new surgical instruments and devices are increasingly introduced into modern surgery. Inexperienced surgeons have limited preparation to employ these new techniques. For example, the RAMIS has obtained great advances in the last two decades, especially the development of the da Vinci surgical system (Intuitive Surgical, CA, USA). RAMIS allows performing complex surgical operations with high accuracy and flexibility while minimizing the post-operatory impact on the patient. Although surgical robots like the da Vinci surgical system have simplified the surgical process through master-slave bilateral control, stereo vision and delicate ergonomics, teleoperating the robotic manipulators in a confined space is still a task of great challenges due to risky collisions with the anatomic structure. This set demands of innovative methods to improve surgeons' ability of surgical robot operation and surgical skills training. The surgical training program based on real objects such as 3D-printed organs and cadavers has been widely used [3]. In early surgical training practices such as laparoscopic procedures, live animals were widely used. Due to ethical problems as high cost, Dry lab laparoscopic simulator was used as a substitute for live animals [4]. Later, the development of computer and sensor-based system motivated more complex and customised surgical training of full surgical procedures. For instance, a Transoral Robotic Surgery (TORS) training platform with a series of collision sensors presented in [5] shows the benefits of surgical training. However, this real system based method is of significant limitations due to the high cost of building the surgical scenes including diverse robotic hardware.

Simulation-based surgical training through lower cost, validated, virtual surgical training modules is helpful to teaching new surgeons surgical procedures and techniques [6]. It has been validated that the using of repetitive Virtual Reality (VR) tasks before starting practice on patients can greatly improve the learning process [7]. A VR simulator on the MicrovisTouch platform was designed for ophthalmic surgery with limited degrees of freedom [8]. In addition, Schill *et al.* [9] and Lam *et al.* [10] also proposed their own virtual simulators for ophthalmic surgery training with good performance. However, they did not specify the surgical robots or instruments. Wang *et al.* [11] presented an interactive VR-based simulator with visual guidance for catheter ablation. The simulator also focused on modelling of the organ and might suffer from the hardware compatibility. The da Vinci surgical system is the most successful RAMIS system which can provide intuitive interaction on motor inputs and human operators.

The da Vinci Research Kit (dVRK) is an open-source system, composed of hardware and software that is wildly adopted for research based on the first-generation da Vinci system [12]. Researchers have developed various innovative applications and algorithms [13], [14] based on this shared research platform. Yan *et al.* [15] proposed a dynamic simulation based on dVRK for modelling of the parallelograms, springs and counterweight. A simulation for manipulation, visualization, and interaction of soft bodies and robots was presented in [16]. A surgical training platform of the da Vinci system also requires significant system integration and a unified control framework for customized algorithm development. In addition, lots of robotic assistance methods were researched in previous literature such as studies of haptic guidance, virtual fixtures and active constraints, which showed promising enhancement in safety, accuracy and decreasing in learning load [17] [18]. For example, operators may suffer from bad hand-eye coordination in complex surgical tasks. Haptic assistance can help operators focus on the tasks rather than motor skills and other aspects of the mechanical setup[19]. Therefore, surgical training should benefit from haptic assistance.

This paper proposes the development of a virtual simu-

[1]Ke Fan, Nicolò Pasini, Alberto Rota, Matteo Pecorella, Giancarlo Ferrigno and Elena De Momi are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy. ke.fan@polimi.it

[2]Aldo Marzullo is with the Department of Mathematichs and Computer Science, University of Calabria, Rende, Italy.

lation platform composed of full da Vinci surgical system components for surgical training, as shown in Fig.1. The simulator is a telesurgical system. The Master Tool Manipulators (MTMs) are used as the haptic input device. The dynamic simulations of three Patient Side Manipulators (PSMs), Setup Joints (SUJs) and the Endoscopic Camera Manipulator (ECM) are provided in this simulator. Furthermore, the haptic function is integrated into the simulator. In order to validate the potentiality for surgical training of the simulator, we developed a scenario of the virtual fixture. Virtual fixtures are control strategies used in robotic surgery that restrict or adjust the motion of the surgical tool with respect to predefined paths, trajectories [20]. We studied the effect of haptic assistance in path following by ten non-medical subjects. In addition, Various surgical training scenarios can be established so that the simulator provides low-cost access to the development of innovative control strategies. We build the simulator based on the Unity physic engine. Users can develop their own algorithms towards every component of the robot through Unity embedded scripts, ROS nodes or customized remote clients. The communication between the simulator and ROS is realized by ROS# based on the *rosbridge* suite. ROS# [21] enables the communication between Unity and ROS. New surgical components or scenarios can be easily imported into the simulator.

This paper is organized as follows: Section II describes the details of the setup of the simulator such as framework and communication methods. Then, the validation approach of a virtual fixture is introduced; Section III illustrates the experimental results and discussion; Section IV is the conclusion and future work.

II. Methods

The simulator is composed by two parts: the software part (the virtual scene) as shown in Fig.1 and the hardware part (the da Vinci console), which is used to control the simulated robot.

A. System Architecture

The standard da Vinci surgical system is composed of mechanical hardware and control software that supplies directly bilateral control to the manipulators on both the patient side and the surgeon side.

The simulator consists of dynamic simulation part, teleoperation part, force feedback (wrench) generation part, Graphical User Interface (GUI) and rodsbridge suite for communication between Unity and dVRK. The dynamic simulation part provides the physics of each virtual component with kinematics and collision configuration. The manipulators are modelled in the CAD software and imported as the Unified Robot Description Format (URDF) files. In Unity, the kinematic chain of each manipulator is realized by the hinge joint component for revolute joints and the configurable joint component for prismatic joints. The teleoperation is realized by the da Vinci console with MTMs. The da Vinci console consists of a stereo display, three foot pedals, two MTMS with 7 active joints and a passive gripper.

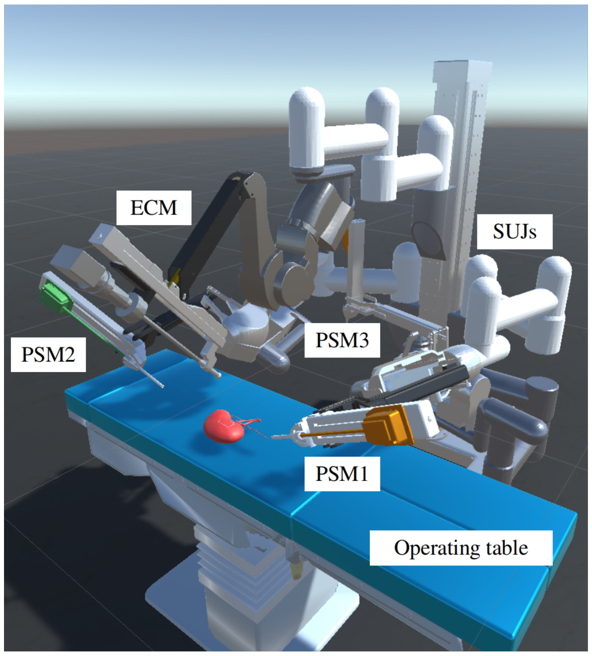

Fig. 1. The da Vinci surgical simulator in Unity scene including three PSMs, a ECM, SUJs and an operating table.

The MTMs can provide force feedback. The teleoperation scaling is adjustable. The controller hardware is connected by the firewire with a safety relay check. On the Unity side, two cameras (resolution of 640×480) are mounted on the end of the ECM link to simulate the binocular vision of the real da Vinci robot endoscope. The view of the simulated endoscope is displayed in the real console to realize the stereo display. The force feedback generation part aims to improve the surgical training performance of the simulator. The generation algorithm is a high-level controller which can be developed using C++, Python outside of Unity or inside of Unity with C# scripts.

B. Communication Between Components

The communication block diagram of the simulator is shown in Fig. 2. The dVRK is a hybrid of the low-level controllers and hardware, which has a component-based software architecture [12]. Different modules and functions can be tailor-made with configuration files (JSON files) in dVRK part. In this context, we customize four kinematic configuration files for simulating the state of PSMs and the ECM. For example, the PSM kinematic file leverages the PSM device interface embedded in the dVRK to trick dVRK to regard the simulated PSM as the real PSM connected to the system. The controllers and manipulators are connected by sending JSON-based messages over User Datagram Protocol (UDP). The *rosbridge* suite is used to connect dVRK and the Unity environment after pointing the IP address of the ROS master. Joint state messages can be subscribed by the scripts in Unity. The wrench message generated in Unity can also be published to MTMs. In the Unity environment, the surgeon can set the task states and parameters. The environment runs

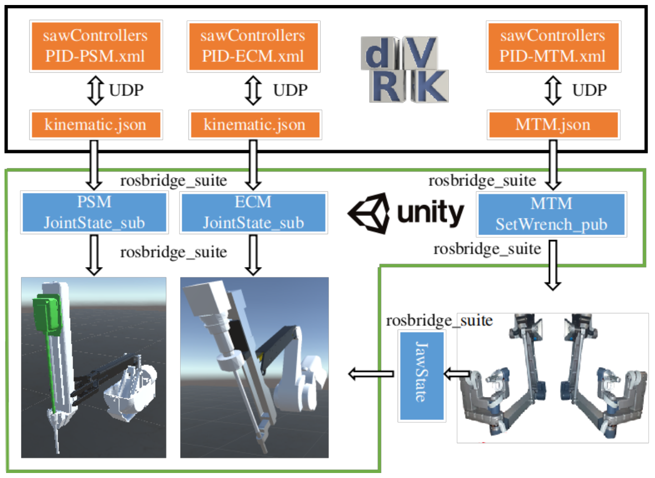

Fig. 2. The communication between dVRK software, Unity environment and MTMs.

at 25 Hz. Furthermore, the performance of the surgeon such as the moving velocity and deviation form the reference path can be collected by the simulator for computing the intensity of force feedback to provide haptic assistance.

C. Virtual Fixture

In order to demonstrate the potentiality and performance of the simulator in surgical training, an application of the virtual fixture is shown in this section. When performing a surgical task through the teleoperated robotic system, the surgeon commands may not align to the requirements of the surgical task. The concept of virtual fixtures provides the possibility of eliminating the biased commands from operators. The guidance force computed for the virtual fixture in this section is of viscous type, as its magnitude depends on the velocity of the end-effector in space. A more detailed discussion can be found in the previous work [22]. The force magnitude G at each time step is, in this case, calculated as:

$$G = \operatorname{sgn}\left(G_{max}, \|b \cdot \boldsymbol{v}\|\right) \tag{1}$$

$\boldsymbol{v}$ is the velocity vector of the end-effector in space. The $\operatorname{sgn}(\alpha, \beta)$ is a selection function shown as

$$\operatorname{sgn}(\alpha,\beta) = \begin{cases} \alpha, & \text{if } \alpha > \beta \\ \beta, & \text{otherwise} \end{cases} \tag{2}$$

b is an anisotropic coefficient obtained as follows:

$$b = B \cdot \sqrt{\frac{1 - \boldsymbol{v} \cdot \boldsymbol{\varphi}}{2}} \tag{3}$$

Where $\boldsymbol{\varphi}$ is the deviation, denoted as the vector going from the end-effector to the closest point in the path, while B represents the viscosity coefficient.

The direction of the guidance force is continuously changing with the tool tip motion as shown in Fig. 3. The force direction $\vec{\boldsymbol{G}}$ is computed as:

$$\vec{\boldsymbol{G}} = \begin{cases} \boldsymbol{\varphi} & if \quad \vec{\boldsymbol{v}} \cdot \boldsymbol{\varphi} < 0 \\ rotate(\vec{\boldsymbol{v}}, \theta, \boldsymbol{n}) & otherwise \end{cases} \tag{4}$$

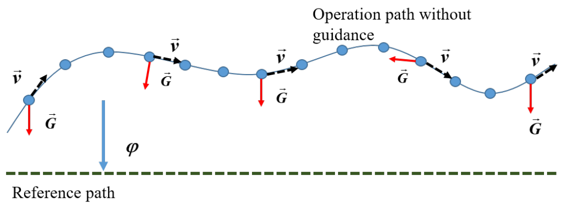

Fig. 3. The direction of the guidance force.

where the $rotate(\cdot)$ function rotates the velocity direction vector $\vec{\boldsymbol{v}}$ around the axis denoted by vector $\boldsymbol{n}$ of θ degrees. Equation 3 shows that if the end-effector is moving away from the path, the force direction is aligned with the deviation direction. Conversely, the vector rotation is applied solely in the cases where the end-effector approaches the path. In this context, the angle and axis of rotation are defined as:

$$\theta = (1 + \vec{\boldsymbol{v}} \cdot \boldsymbol{\varphi}) \cdot \frac{\pi}{2} \tag{5}$$

$$\boldsymbol{n} = \vec{\boldsymbol{v}} \times \boldsymbol{\varphi} \tag{6}$$

Finally, the guidance force $\boldsymbol{G}$ is calculated as:

$$\boldsymbol{G} = G \cdot \vec{\boldsymbol{G}} \tag{7}$$

Fig. 4 shows the schematic diagram of the computation of the guidance force $\boldsymbol{G}$.

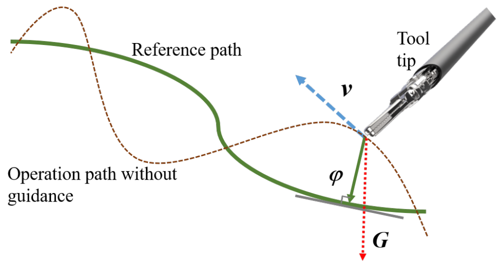

Fig. 4. The concept of the viscous-based virtual fixture.

D. Experiments

The experimentation setup consists of a virtual simulation environment in Unity and two MTMs as the haptic devices. To keep the structural integrity of the simulator, the haptic control loop and 3D graphics are implemented in the same environment on one computer with 3.2GHz Core i7 processor (Intel Corp.) and a GeForce GTX 1650 (Nvidia Corp.). The haptic control loop was implemented as a modularized Unity component in C# on a real-time Xenomai-patched Linux kernel. The two-way communication between the haptic control loop and the haptic device is realized by UDP to exchange data such as the tools' pose, forces and tool's velocity. The haptic rate is 1 kHz and the 3D display refresh rate is 25Hz. The detailed implementation in the simulator is shown in Fig.5. Fig. 5 (b) shows that a operator is operating the task at the console. The simulator can support different camera positions for external monitoring of the tasks, as

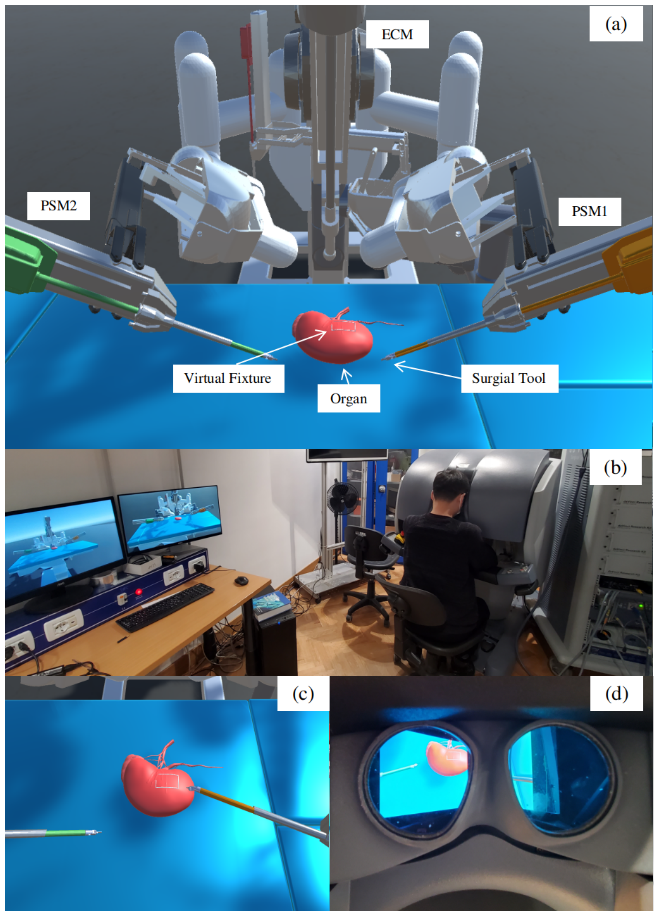

Fig. 5. The virtual fixture in the Unity scene. Figure (a) is the overview of the training task. Figure (b) shows the user operating MTMs at the da Vinci teleoperation station. Figure (c) shows the view of surgeons on the console with the stereo display. Figure (d) shows the stereo display of the console

shown in Fig.5 (a). The operator's view includes information about depth since the acquisition of the virtual stereo camera is sent to the oculars at the master console, as in Fig. 5 (c). Fig. 5 (d) is the stereo viewer on the console.

We got ten male right-handed users without any operating experience to experience the virtual fixture task in the simulator to validate the performance of the simulator. The experimental task is implemented in the simulator as shown in Fig. 6. The task requires both hand-eye coordination and manual dexterity of users. Since the assistance force is computed based on both tool's velocity and deviation, the user will not be affected by force when staying static. This can relieve mental stress of users during the task. Each user was told to move the surgical tool through MTMs along a curved reference path from point 1 to point 5 for ten times and attempts to avoid possible deviation from the path. The curved path is designed to span along the three Cartesian dimensions. Users need to do 90° rotation for three times at point 2, point 3 and point4. For Instance, the tool is on the position shown in Fig. 6 and moving along the direction represented as the blue line. The assistance force would be like the yellow line to pull the tool back to the path. We

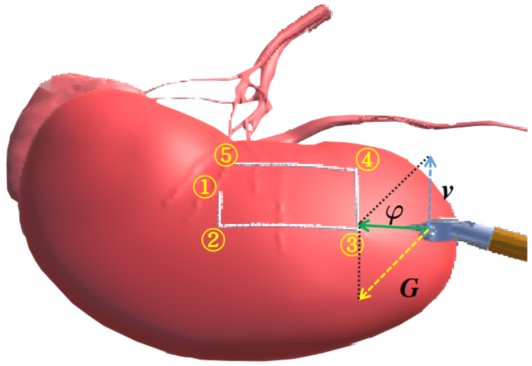

Fig. 6. Details of the training task. Users need to move the tool along the white path, starting from point 1 and ending at point 5.

adopted B=0 Ns/m, B=20 Ns/m, B=40 Ns/m, B=60 Ns/m according to equation (2) respectively for the same task so that the assistance intensity for each subject increased. The parameters used for experiments are presented in table I. The maximum guidance force was set to 2N, 5N and 8N for three assistance experimental methods. The objective of constraining maximum force is to prevent the assistance force from being too aggressive. For each repetition, we

TABLE I
PARAMETERS OF THE VIRTUAL FIXTURE

Param.	Values			
B (Ns/m)	0	20	40	60
Gmax (N)	0	2	5	8

measured the deviation error and the completion time. The deviation error can be formulated as

$$e_{jn} = \frac{\sum_{i=1}^{i=K} \varphi_{ijn}}{K} \tag{8}$$

Where φ_{ijn} represents the scalar of deviation at time step i during the repetition j for subject n. Then, we calculate the mean performance of all the subjects as

$$e = \frac{\sum_{n=1}^{n=10} \sum_{j=1}^{j=10} e_{jn}}{jn} \tag{9}$$

The results are shown in the next section.

III. Results and Discussion

The training data collected for each repetition are shown in Fig.7. The blue scatters represent the average deviation error of all subjects in each repetition. The bars represent the 20th to 80th percent of data in each repetition. When B=0, which means no assistance force, the deviation error decreases with increasing of the repetition times slightly. In this case, the subjects learn the task from repetition merely and the learning curve shows large fluctuations. When B=20, which means a small assistance force is implemented on users, the deviation error decreases compared to the case without

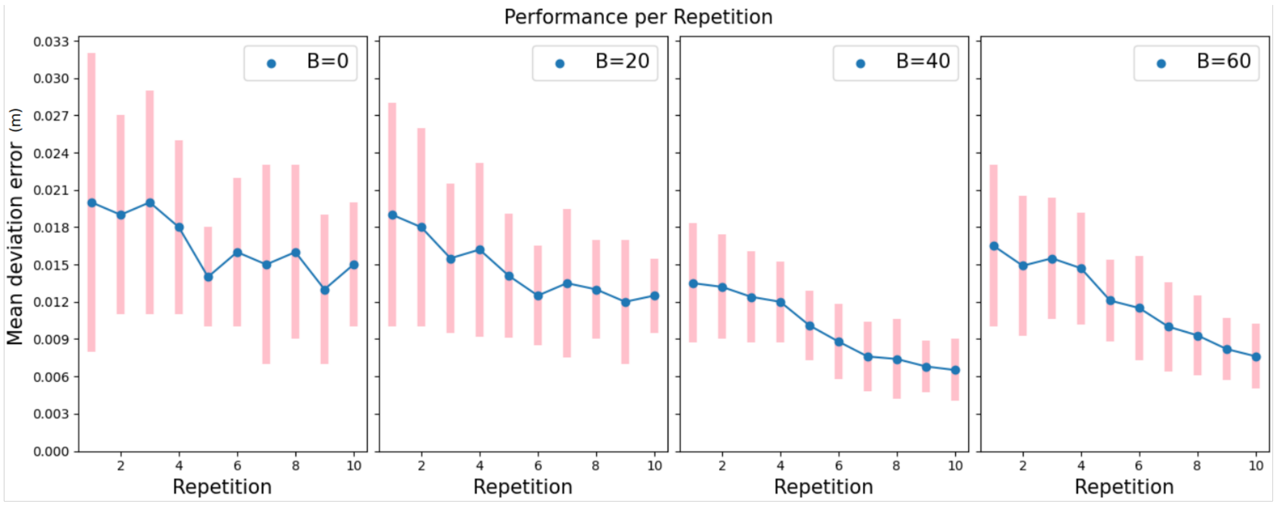

Fig. 7. The mean deviation error from reference path of four cases with different assistant force intensity. Blue dots represent the mean deviation error of all subjects in each repetition. The bars shows the 20th to 80th percent of data in each repetition.

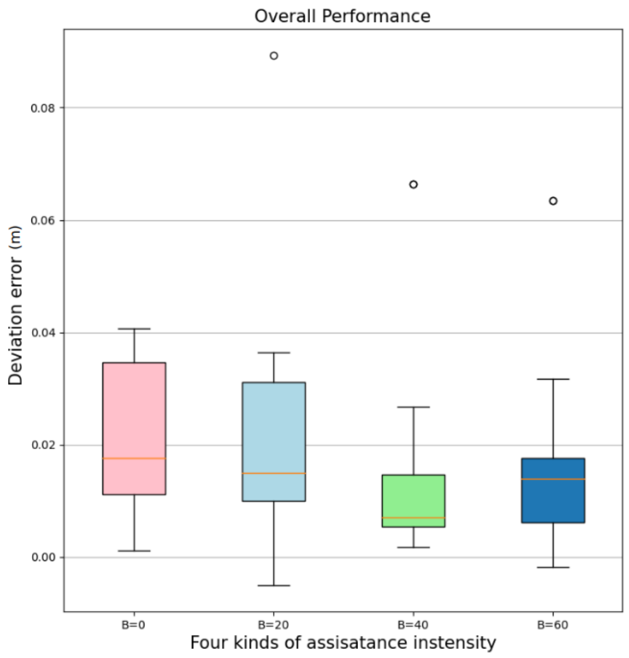

Fig. 8. The deviation error from reference path of four cases with different assistance force intensity. With parameter B increasing, the assistant force increases.

assistance force, but the enhancement is limited. When B=40, which means a larger assistance force is implemented on subjects, a significant progress in terms of deviation error is observed in both error distribution and mean deviation error as the increasing of repetition times. When B=60, the deviation error tends to increase a bit, which means an overlarge assistance force may not improve the performance. Too large force on subjects shows more oscillating and may disturb subjects.

The overall performance of four cases is visualized in Fig. 8. For each subject, ten acquisitions were recorded.

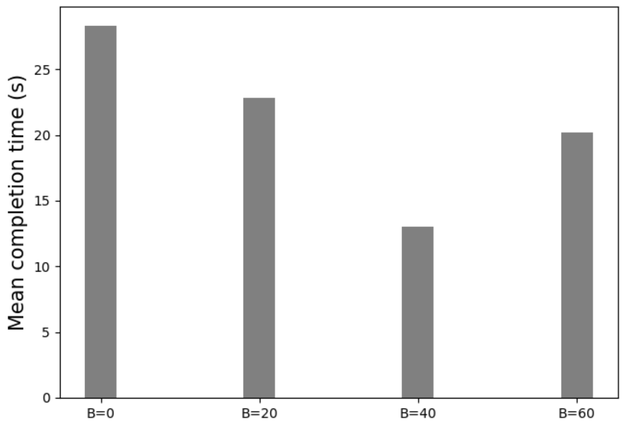

Fig. 9. The mean completion time for four cases. The assistance force increases with the parameter B increasing

The overall performance is represented as the distribution of the deviation error during all repetitions of all subjects. The results suggest that the virtual fixture have reduced deviation in cases with three different force intensity compared with cases without assistance force. As shown in the figure, the deviation decreases with the force increasing. However, When the force is too large (B=60 Ns/m), the virtual fixture does not demonstrate improvements compared with the middle force (B=40 Ns/m). During the experiments with B=60 Ns/m, it is observed some subjects show difficulty in completing the task efficiently. It also can be observed in Fig. 9. The task with a large assistance force (B=60 Ns/m) is more time-consuming than the case with a middle assistance force (B=40 Ns/m). This is because of oscillating and disturbing on subjects with large force. This also means the intensity of assistant force should consider individual difference. Furthermore, the variability of the deviation error has been seen reduced in three cases with assistance force among the subjects compared to the case without assistance force.

Overall, the results show that the demonstrated virtual fixture with proper parameters can enhance the surgical task operating process, and supply efficient guidance for users during the training process. As shown in this example application, the simulator can support tailor-made tasks. It can reduce the risk of breaking the real da Vinci robot, especially when testing some high level control strategies.

IV. CONCLUSIONS AND FUTURE WORK

In this work, a Unity-based simulator of da Vinci surgical robot is proposed. External algorithms can be developed and connected to the simulator via the ROS framework. The simulator is an interactive environment that can benefit surgical training. The application of a virtual fixture highlighting the potentialities of developing different control algorithms. The simulator can be found at https://github.com/fanke1213812138/Davinci-simulator-Unity.

In the future, machine learning algorithms based on this simulation framework may be developed, with the aim of extending the function of simulation transferring it to real robots.

ACKNOWLEDGMENT

We received suggestions and help on the dVRK from Anton Deguet. Thank to our subjects for participating in the experiment.

REFERENCES

[1] J. Dankelman, C. K. A. Grimbergen, and H. G. Stassen, "New technologies supporting surgical intervenltions and training of surgical skills - a look at projects in europe supporting minimally invasive techniques," *IEEE Engineering in Medicine and Biology Magazine*, vol. 26, no. 3, pp. 47–52, 2007.

[2] N. Enayati, A. M. Okamura, A. Mariani, E. Pellegrini, M. M. Coad, G. Ferrigno, and E. De Momi, "Robotic assistance-as-needed for enhanced visuomotor learning in surgical robotics training: An experimental study," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6631–6636.

[3] M. Shahbazi, S. F. Atashzar, C. Ward, H. A. Talebi, and R. V. Patel, "Multimodal sensorimotor integration for expert-in-the-loop telerobotic surgical training," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1549–1564, 2018.

[4] G. M. Fried, L. S. Feldman, M. C. Vassiliou, S. A. Fraser, D. Stanbridge, G. Ghitulescu, and C. G. Andrew, "Proving the value of simulation in laparoscopic surgery," *Annals of surgery*, vol. 240, no. 3, p. 518, 2004.

[5] R. Geoghegan, J. Song, A. Singh, T. Le, A. Abiri, and A. H. Mendelsohn, "Development of a transoral robotic surgery training platform¡sup¿*¡/sup¿," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 5851–5854.

[6] A. E. Abdelaal, M. Sakr, A. Avinash, S. K. Mohammed, A. K. Bajwa, M. Sahni, S. Hor, S. Fels, and S. E. Salcudean, "Play me back: A unified training platform for robotic and laparoscopic surgery," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 554–561, 2019.

[7] R. M. Satava, "Virtual reality surgical simulator," *Surgical endoscopy*, vol. 7, no. 3, pp. 203–205, 1993.

[8] J. Luo, P. Kania, P. P. Banerjee, S. Sikder, C. J. Luciano, and W. G. Myers, "A part-task haptic simulator for ophthalmic surgical training," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, 2016, pp. 259–260.

[9] M. A. Schill, C. Wagner, M. Hennen, H.-J. Bender, and R. Männer, "Eyesi–a simulator for intra-ocular surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 1999, pp. 1166–1174.

[10] C. K. Lam, K. Sundaraj, and M. N. Sulaiman, "Virtual reality simulator for phacoemulsification cataract surgery education and training," *Procedia Computer Science*, vol. 18, pp. 742–748, 2013.

[11] H. Wang, S. Jiang, and J. Wu, "A virtual reality based simulator for training surgical skills in procedure of catheter ablation," in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 2018, pp. 247–248.

[12] 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)*, 2014, pp. 6434–6439.

[13] X. Ma, C. Song, P. W. Chiu, and Z. Li, "Visual servo of a 6-dof robotic stereo flexible endoscope based on da vinci research kit (dvrk) system," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 820–827, 2020.

[14] 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)*, 2021, pp. 1821–1828.

[15] Y. Wang, R. Gondokaryono, A. Munawar, and G. S. Fischer, "A convex optimization-based dynamic model identification package for the da vinci research kit," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3657–3664, 2019.

[16] A. Munawar, N. Srishankar, and G. S. Fischer, "An open-source framework for rapid development of interactive soft-body simulations for real-time training," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6544–6550.

[17] R. Kikuuwe, N. Takesue, and H. Fujimoto, "A control framework to generate nonenergy-storing virtual fixtures: Use of simulated plasticity," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 781–793, 2008.

[18] J. Aleotti, S. Caselli, and M. Reggiani, "Evaluation of virtual fixtures for a robot programming by demonstration interface," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 4, pp. 536–545, 2005.

[19] M. A. Finan, M. E. Clark, and R. P. Rocconi, "A novel method for training residents in robotic hysterectomy," *Journal of robotic surgery*, vol. 4, no. 1, pp. 33–39, 2010.

[20] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2001, pp. 1419–1420.

[21] M. Bischoff, "Rossharp," *https://github.com/siemens/ros-sharp*.

[22] N. Enayati, E. C. Alves Costa, G. Ferrigno, and E. De Momi, "A dynamic non-energy-storing guidance constraint with motion redirection for robot-assisted surgery," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4311–4316.