

Attaching a Surgical Insertion Tool onto an Industrial Manipulator

Yan Wang¹, Abhijeet Thakan¹, Kenechukwu Mbanisi¹, Tianyu Cheng², and Yuqi Jiang²

Abstract—Robot-assisted minimally-invasive surgeries (RAMIS) have achieved significant commercial success during the last decade. Increasingly more researchers have started working on improving the performance of RAMIS. However, most RAMIS robots require specifically designed mechanism, which makes it expensive and not accessible to most researchers. In this work, we converted a general industrial robot into a surgical robot. To achieve this, we first made a mounting for the insertion tool of da Vinci Research Kit (dVRK) and its actuators. Second, we developed a software package consisting of the kinematics, trajectory generation, and teleoperation subject to the remote center of motion constraints. We also built the Gazebo model for the simulation of the robot. Experimental results show the integrated system and the developed software package are capable of teleoperation in Gazebo, and the actuation system is able to actuate the insertion tool correctly.

I. INTRODUCTION

Robot-assisted minimally-invasive surgeries (RAMIS) have surged in the past decade because of its advantages over conventional laparoscopic surgeries, such as 3D vision, motion scaling, intuitive movements, visual immersion, and tremor filtration [1]. This growth in popularity has led to the increase in research interest to improve the system performance in order to achieve better clinical outcomes.

RAMIS are conducted by inserting thin-wristed surgical instruments into the body of the patients through small incisions. A crucial requirement for laparoscopic surgeries is the remote center of motion (RCM) constraint at the incisions (entry point on the body). To achieve this, customized mechanisms have been designed by various surgical robot manufacturers. A family of RCM mechanisms based on intersecting motion planes are introduced in [2]. The da Vinci Surgical System by Intuitive Surgical, Inc., which is the most commercially successful robot for assisted minimally-invasive surgery, uses a double four-bar parallelogram to achieve the RCM [3]. Raven, a surgical robot developed by researchers from University of Washington, also uses a spherical mechanism to realize the RCM, in which the first four joint axes intersect at the surgical port location [4]. While interests in RAMIS increases, the high cost of existing off-the-shelf surgical robot platforms may stand as a limitation in its growth in research and education.

Existing literature shows efforts have been made for adapting general-purpose robot manipulators for robot-assisted

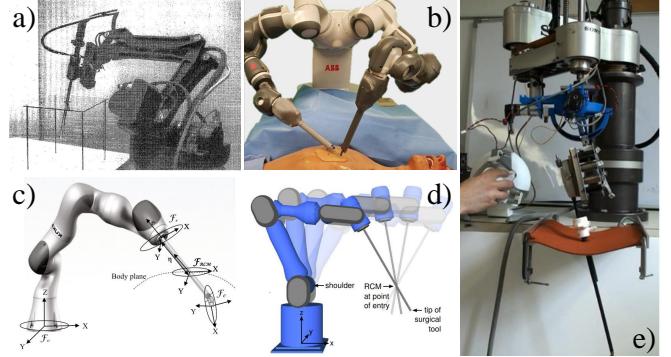


Fig. 1. Previous work trying to attach surgical insertion tools on general-purpose robot arms. a) Parallel Linkage Remote-Center-of-Motion (PLRCM) surgical robot [5]. b) ABB Yumi collaborative robot with a customized surgical gripper [6]. c) 7-DOF iiwa KUKA manipulator with a customized insertion tool [7]. d) Mitsubishi PA10-7C general-purpose 7-DOF manipulators with custom-designed end-effector modules [8]. e) SCARA robot with a customized wrist and the insertion tool of the da Vinci Surgical Robot [9].

surgical applications. This often includes adapting the robot with a surgical-grade insertion tool at the end-effector. Funda *et al.* presented an approach for constrained Cartesian motion control for teleoperated surgical robot with Parallel Linkage Remote-Center-of-Motion (PLRCM) surgical robot as shown in Fig. 1(a), designed and built at IBM [5]. Locke *et al.* presented a work using a general-purpose Mitsubishi PA10-7C, which is a 7-DOF manipulator with custom-designed end-effector modules as shown in Fig. 1(d), and an optimization-based method to solve the RCM constraint [8]. Marchese *et al.* used a SCARA robot with a customized wrist and the insertion tool of the da Vinci Surgical Robot as shown in Figure 1(e) to realize the functionality of RAMIS [9]. Sanchez *et al.* attached a surgical gripper to an ABB Yumi collaborative robot as shown in Figure 1(b) and achieved melanoma extraction by teleoperation on HTC VIVE handles [6]. Hamid *et al.* attached a customized insertion tool on a 7-DOF iiwa KUKA manipulator as shown in Figure 1(c) and proposed a way to resolve the RCM constraints [7]. All of the aforementioned works adapt industrial robots for surgical applications. However, they either use tools that are not suitable for RAMIS or require customized insertion tools, which makes it hard for other researchers to duplicate.

In this work, we propose a low-cost framework to adapt a standard industrial manipulator as a surgical assistive robot. Specifically, we developed the actuation and mounting platform to attach a surgical insertion tool to the end effector of the ABB IRB-120 robot. We then performed the kinematics and trajectory module that guarantees the RCM constraints.

*All the authors contribute to this work equally

¹Yan Wang, Abhijeet Thakan, Kenechukwu Mbanisi are with the Robotics Engineering Program, Worcester Polytechnic Institute, Worcester, MA 01609, USA

²Tianyu Cheng, and Yuqi Jiang are with the Department of Mechanical Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA

We demonstrate the performance of our integrated system in a teleoperation experiment using the master tool manipulator (MTM) of the dVRK in simulation and finally perform a motion comparison of the insertion tool in simulation and on the physical hardware.

II. SYSTEM STRUCTURE

In this section, the mechanical system of the mounting between the insertion tool and the end-effector of the industrial arm is first described. Second, the ROS-based [10] software package is introduced. Third, the simulation of the integrated system in Gazebo [11] is presented.

A. Mechanical system

The goal of our proposed design is to create an adaptive mounting structure with driving mechanisms which connects the surgical insertion tool from da Vinci Surgical System to the ABB IRB-120 robot.

There are several requirements for the servos to drive the spin wheels of the da Vinci surgical tool. First, the servos should be lightweight to reduce the extra load on the robot, and compact for easier design and installation. The servos should have a motion range of at least 180° , and be able to rotate through the motion range within 1 second. Finally, according to [9], the servos should provide at least 0.18 Nm torque to the spin wheels during typical suturing maneuvers.

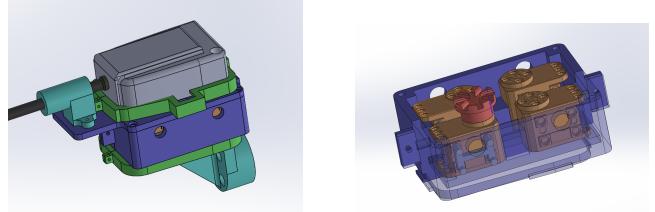
Based on the requirements above, we selected the Dynamixel XL320 servos. This servo meets the weight, dimension, torque and speed requirements of our system, and has a relatively high resolution at the same time. Compared to regular analog servos, it receives and sends digital command via a TTL serial bus, therefore, all of the four servos are connected through serial bus. Also, the servos provides several feedback parameters including position, torque and temperature. The major parameters of XL320 servos are listed at Table I.

TABLE I
MAJOR PARAMETERS OF DYNAMIXEL XL320

Item	Specifications
Resolution	0.29°
Running Range	0 - 300°
Weight	16.7 g
Dimensions (W × H × D)	24 mm × 36 mm × 27 mm
Stall Torque	0.39 Nm @ 7.4 V, 1.1 A
No Load Speed	114 rpm @ 7.4 V, 0.18 A

For the mechanical mounting system, we designed a modular structure for the connection, which includes a motor box, two mount structures designed to connect ABB robot end-effector and the insertion instrument of the da-Vinci Research Kit (dVRK) [12], and several position control components (see Fig. 2).

The motor box in the middle is the power and control system, which comprises four servo motors placed in an exact order and relative position to match the four rotating control plates of the surgical tool. For convenient fitting, we divided the motor box into two half with suitable grooves and



(a) Assembled CAD model with surgical tool mounted
(b) Transparent view to show inner composition for servos setting and flange

Fig. 2. CAD model of the insertion tool mounting assembly



Fig. 3. 3D-printed insertion tool mounting assembly

boss features, which could fix the position of motors along vertical direction. Then we added several support beams with a "H" shape to fix the distance between the motors' sidewall and the box's inner wall. Therefore, we have the four servo motors arranged at the exact relative position. Furthermore, a simple flange connector was designed and manufactured to connect the motors and the rotating control plates from the surgical tool, which allows the control of the 4-DOF motion of the surgical tool by actuating the servos. Then, there are two mounting components. First one is the cover of motor box, which has a surface to mate with the surgical tool. The second is the corner component which connects the back of motor box to the robot end-effector. Both mounting components are simply connected using pairs of grooved buckles which can lock their position by the help of raised edges. As described above, the two mounting components are modular and separate with respect to the main control box, for ease of modification for different connections. Fig. 3 shows the tool mounting assembly attached to the da-Vinci Surgical insertion tool.

B. Software Framework

1) *Kinematics and Trajectory Generation:* As shown in Figure 4, we implemented a ROS node to manage the kinematics and trajectory generation of the robot and surgical tool. This node provides absolute and relative position control interfaces in joint space and Cartesian space. It also provides interfaces to inquire robot current state.

2) *Servo Control:* As shown in Figure 5, all the XL320 servos are serially connected to OpenCM 9.04, which is a microcontroller unit (MCU) based on 32bit ARM Cortex-M3. An external power source provides 7.4 V to the MCU

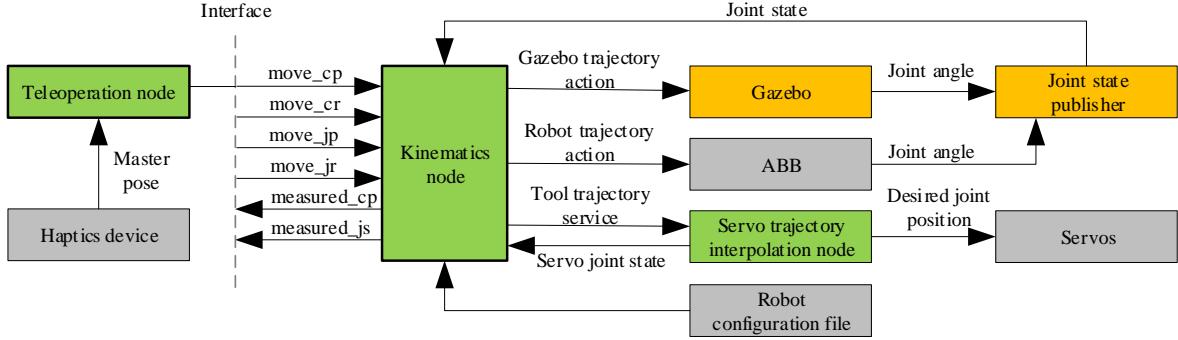


Fig. 4. Software structure of the system

and the servos. The MCU is connected to the control PC via USB. The control software running on the MCU communicates on the ROS network via ROSserial (<http://wiki.ros.org/rosserial>).

There are several steps to set up the servo control. First, every XL320 servo should be labeled with identical IDs and unified TTL bus baud rate, and the TTL port on the MCU should be set with same baud rate as the servos. This is done using the R+ Manager 2.0, which is a software that manages the Dynamixel controllers and servos. The e-Manual is available at <http://emanual.robotis.com/docs/en/software/rplus2/manager/>. The control software is developed and uploaded to the MCU via Arduino IDE. The software is designed to initialize the MCU as a ROS subscriber node which listens to a servo command topic published by the servo trajectory interpolation node on the ROS network (see Fig. 4).

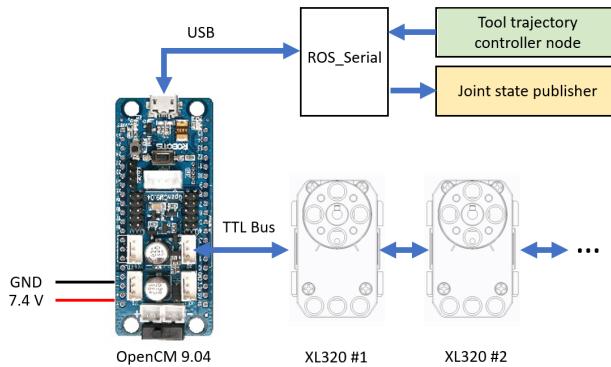


Fig. 5. Servo control diagram

The servo trajectory interpolation node interfaces with the kinematics node on a ROS service protocol and receives desired tool trajectory. The node then performs a mapping from the dVRK-ROS joint coordinates q_{5-7}^d to the servo motor joint coordinates using a coupling matrix defined in the user guide of the dVRK for wrist joint actuation. The coupling matrix is defined as A_m^d :

$$q_{5-7}^d = A_m^d q_{5-7}^m \quad (1)$$

$$\text{where } A_m^d = \begin{bmatrix} 1.0186 & 0 & 0 \\ -0.8306 & 0.6089 & 0.6089 \\ 0 & -1.2177 & 1.2177 \end{bmatrix}.$$

3) Interfacing with the Robot Controller: To communicate with the robot controller device, a ROS server environment has to be set up and configured on the controller. First, the server codes were uploaded to the robot controller and then task settings were configured using the ABB FlexPendant. A detailed setup procedure is available on the ABB ROS wiki page (<http://wiki.ros.org/abb/Tutorials>). Once the environment is set up, the main control PC running the ROS Master is then connected to the robot controller via Ethernet, thus creating a communication link over ROS. The robot controller has a fixed IP address (192.168.125.1) on the network.

4) Teleoperation Interface: Beyond enabling trajectory tracking with our system, our reach goal in this work is to enable direct teleoperation of the surgical industrial robot using standard input devices such as a Geomagic Touch Haptic device. For this, we first install the hardware driver from OpenHaptics SDK [13]. A ROS package for robot teleoperation using the Geomagic Touch is available on the WPI AIM Lab Github repository (<https://github.com/WPI-AIM/wpi-dvrk-ros>). Detailed methodology of the teleoperation implementation will be discussed in the final report.

C. Simulation

It is a good practice to test new features or methods in simulation before they are deployed on the actual robot to minimize the risk of injury to human or damaging the hardware. For this work, we used a Gazebo model for the mount, the surgical tool and the ABB IRB 120 robot. All the three components are independent of each other as a separate model to maintain modularity. We created a Gazebo model for both the surgical insertion tool and the mount equipped with so as to make it easier to control through ROS.

For any robot to work in Gazebo simulation a URDF file of the robot has to be created. URDF is in XML file format which has all the description of the robot including the robot name, link name, link position, link orientation, joint name, joint position, joint orientation, etc. This file contains not only the visuals for the model but also its

physical parameters. Since Gazebo is a physics engine it also simulates the dynamics of the robot and not just the kinematics making it closer to reality.

This URDF file can be either created beforehand and stored in the system or generated in runtime of the simulation by using Xacro files. As its name implies, Xacro is a macro language, i.e. it has small modules called Macros which can be expanded to larger XML expressions. Having a modular system is important for a number of reasons one of which is to make the components transportable. If this system would be used on a new robot, it should work without making major changes. This will also save time when porting the components to a new system. The Xacro package has provisions to call different URDF files and create a combined model in simulation. Similar to URDF, Xacro is also in XML file format. This Xacro file can generate a URDF file of the combined system in runtime and show it in simulation. This way modularity is assured and the structure stays clean.

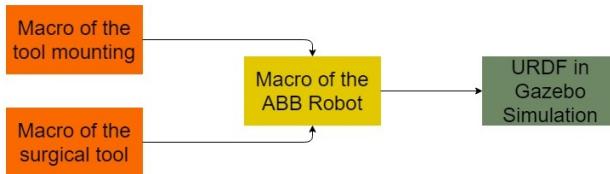


Fig. 6. Modular Approach

For modular approach we included the tool and mount Macros in the Macro of the ABB IRB 120 robot. This ensured that the robot base frame is fixed with the world frame so that it does not show any error during runtime. The Macros of the tool and mount were included in the Macro of the ABB robot and then given the required parameters like the joint between the robot end effector and the base of mounting and the joint between the end of mount and the base of the surgical tool. Visuals were also created to show the RCM constraint.

III. CONSTRAINED MANIPULATION WITH RCM CONSTRAINT

This section presents the kinematic modeling and algorithms for the constrained manipulation with RCM constraint. The basic kinematics and trajectory generation algorithms are referred from [14].

A. Forward kinematics

The forward kinematics of a robot modeled with the screw theory [14] has the form

$$\mathbf{T}(\boldsymbol{\theta}) = \prod_{k=1}^{10} e^{[\mathbf{S}_k]\theta_k} \mathbf{M} \quad (2)$$

where $\mathbf{M} \in SE(3)$ is the end-effector frame pose in home configuration in task space. $\mathbf{S}_i = (\boldsymbol{\omega}_i, \mathbf{v}_i)$ is the screw axes, where $\boldsymbol{\omega}_i$ is the joint axis vector and \mathbf{v}_i is calculated by $\mathbf{v}_i =$

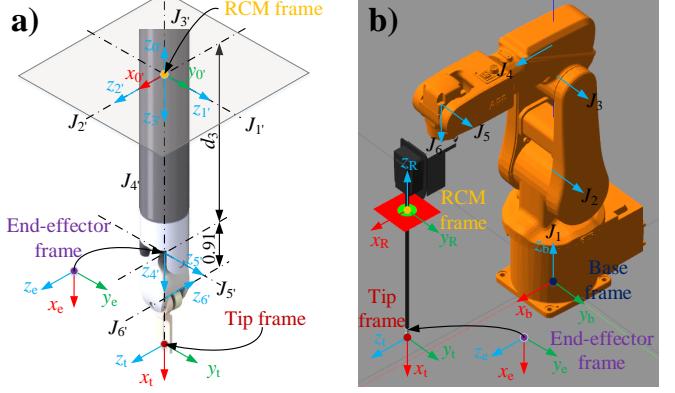


Fig. 7. Frame and axis definition of the robot. a) Intraoperative frames and axes. b) ABB robot frames and axes

$-\boldsymbol{\omega}_i \times \mathbf{q}_i$, where \mathbf{q}_i is an arbitrary point on the corresponding rotation axis.

To get the kinematics of the end-effector of the ABB robot and the jaw tip of the insertion tool, their corresponding frame poses in home configuration, M_e and M_j , are needed. By inspection,

$$M_j = \begin{bmatrix} 0 & 0 & 1 & 0.91 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.73 \\ 0 & 0 & 0 & 1 \end{bmatrix}, M_e = \begin{bmatrix} 0 & 0 & 1 & 0.89 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.73 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The screw axes are listed in Table II.

TABLE II
SCREW AXIS OF EACH JOINT IN THE BASE FRAME OF THE ABB ROBOT

i	$\boldsymbol{\omega}_i$	\mathbf{q}_i	\mathbf{v}_i
1	(0, 0, 1)	(0, 0, 0)	(0, 0, 0)
2	(0, 1, 0)	(0, 0, 0.29)	(-0.29, 0, 0)
3	(0, 1, 0)	(0, 0, 0.56)	(-0.56, 0, 0)
4	(1, 0, 0)	(0, 0, 0.63)	(0, 0.63, 0)
5	(0, 1, 0)	(0.302, 0, 0.63)	(-0.63, 0, 0.302)
6	(1, 0, 0)	(0.374, 0, 0.63)	(0, 0.63, 0)
4'	(1, 0, 0)	(0.89, 0, 0.73)	(0, 0.73, 0)
5'	(0, 1, 0)	(0.89, 0, 0.73)	(-0.73, 0, 0.89)
6'	(0, 0, 1)	(0.900, 0, 0.73)	(0, -0.9, 0)

B. Inverse Kinematics

TABLE III
SCREW AXIS OF EACH JOINT OF THE INTRACORPOREAL SECTION IN RCM FRAME

i	$i + 1$	$\boldsymbol{\omega}_i$	\mathbf{q}_i	\mathbf{v}_i
1'	2'	(0, 1, 0)	(0, 0, 0)	(0, 0, 0)
2'	3'	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
3'	4'	(0, 0, 0)	(0, 0, 0)	(0, 0, -1)
4'	5', 6	(0, 0, -1)	(0, 0, 0)	(0, 0, 0)
5'	6'	(0, 1, 0)	(0, 0, 0)	(0, 0, 0)
6'	-	(1, 0, 0)	(0, 0, 0.0091)	(0, 0.0091, 0)

The inverse kinematics of the robot subject to the RCM constraints is based on [15]. In this method, the problem is broken down into two subproblems. First, we solve the intracorporeal kinematics, in which several virtual joints are built with respect to the RCM frame. Joints 1' and 2' are revolute joints rotating around y -axis and x -axis of the RCM frame, respectively. Joint 3' is a prismatic joint moving along the opposite direction of the z -axis of the RCM frame. Joints 4', 5' and 6' are real joints, which are consistent with the modeling in the forward kinematics modeling in task space but transformed to the RCM frame. The screw axes of the intraopereal kinematics are listed in Table. III. The jaw tip pose in the RCM frame when the intraopereal kinematics is in home position is

$$M_j^R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -0.019 \\ 0 & 0 & 0 & 1 \end{bmatrix}, M_e^R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

By solving the inverse kinematics of the jaw tip relative to the RCM frame, we can get the joint angles of joints 1' to 6'. Furthermore, with the joint angles of joints 1' to 4', the pose of the end-effector frame in the RCM T_e^R can be calculated, using forward kinematics. Next, the pose of the end-effector in task space can be obtained by $T_e = T_R T_e^R$, where T_R is the transformation of the RCM frame in task space. Second, the joint angles of joint 1 to 6 can be solved by the inverse kinematics of the end-effector in task space. The inverse kinematics for both the intracorporeal and extracorporeal sections is calculated in numerical ways in [14].

C. Trajectory Generation

The motion is decoupled into rotational motion and transitional motion by $\mathbf{X} = (\mathbf{R}, \mathbf{p})$. And the robot will follow a straight line and meanwhile rotate around a constant axis in body frame. The path can be described by

$$\mathbf{p}(s) = \mathbf{p}_{start} + s(\mathbf{p}_{end} - \mathbf{p}_{start}) \quad (5)$$

$$\mathbf{R}(s) = \mathbf{R}_{start} \exp(\log(\mathbf{R}_{end} \mathbf{R}_{start}^{-1})s) \quad (6)$$

where $s \in [0, 1]$, \mathbf{X}_{start} is the starting pose and \mathbf{X}_{end} is the end pose.

We use quintic function to generate trajectory in Cartesian space in which both of the starting and ending velocity and acceleration can be constrained to be 0. A trajectory with execution time of T can be described by

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (7)$$

where $a_0 = 0$, $a_1 = 0$, $a_2 = \frac{2}{T^2}$, $a_3 = \frac{1}{2}$, $a_4 = \frac{2}{T}$ and $a_5 = -\frac{2}{T^2}$.

IV. EXPERIMENTS AND DISCUSSION

Two experiments are conducted to validate the capability of the developed system. First, we carry out a teleoperation experiment of the integrated system in Gazebo with the MTM of the dVRK as the input device. Second, a motion

comparison experiment of the insertion instrument actuated by our developed system and its Gazebo simulation.

A. Teleoperation of the integrated system in Gazebo with MTM

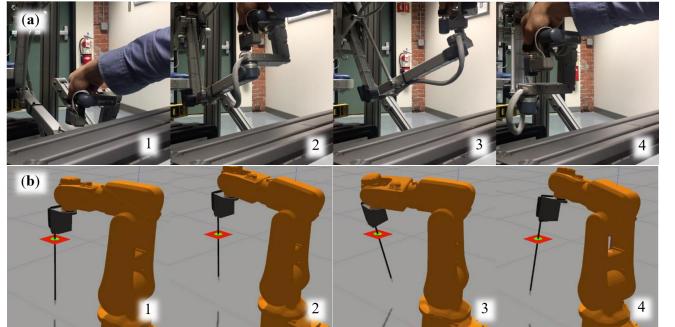


Fig. 8. Translational motion of teleoperation of the integrated system in Gazebo with the MTM of dVRK. a) MTM of dVRK. b) integrated surgical system in Gazebo

In this experiment, we use the MTM of the dVRK to teleoperate the integrated system in Gazebo. The orientation of MTM's tip \mathbf{R}_{MTM} is assigned to the orientation of the tip of our integrated system \mathbf{R}_{system} directly. And the position of MTM's tip \mathbf{p}_{MTM} is mapped to the position of the tip of our integrated system \mathbf{p}_{system} with respect to the RCM frame with a scaling coefficient α . The positions of MTM's tip and our integrated system's tip are both subtracted by offsets, \mathbf{p}_o^{MTM} and \mathbf{p}_o^{system} .

$$\begin{aligned} \mathbf{R}_{system} &= \mathbf{R}_{MTM} \\ \mathbf{p}_{system} &= \alpha(\mathbf{p}_{MTM} - \mathbf{p}_o^{MTM}) + \mathbf{p}_o^{system} \end{aligned} \quad (8)$$

Figure 8 shows the results of translational motion of the teleoperation experiment. We can see the tip of our integrated system follows the translational motion of MTM's tip correctly subject to the RCM constraints.

Figure 9 shows the results of rotational motion of the teleoperation experiment. We can see the tip of our integrated system follows the rotational motion of MTM's tip correctly subject to the RCM constraints, and the openness and closeness also function correctly.

B. Motion comparison of the insertion instrument actuated by our developed system and its Gazebo simulation

Figure 10 shows the motion comparison of the insertion instrument actuated by our developed system and its corresponding Gazebo simulation. We can see the real insertion instrument actuated by our developed system can reach to same orientation of its corresponding Gazebo simulation, which demonstrates the effectiveness of our developed actuation system.

V. CONTRIBUTION

VI. CONCLUSION

A robot-assisted surgical system is developed based on a general industrial robot (ABB irb120) and the insertion

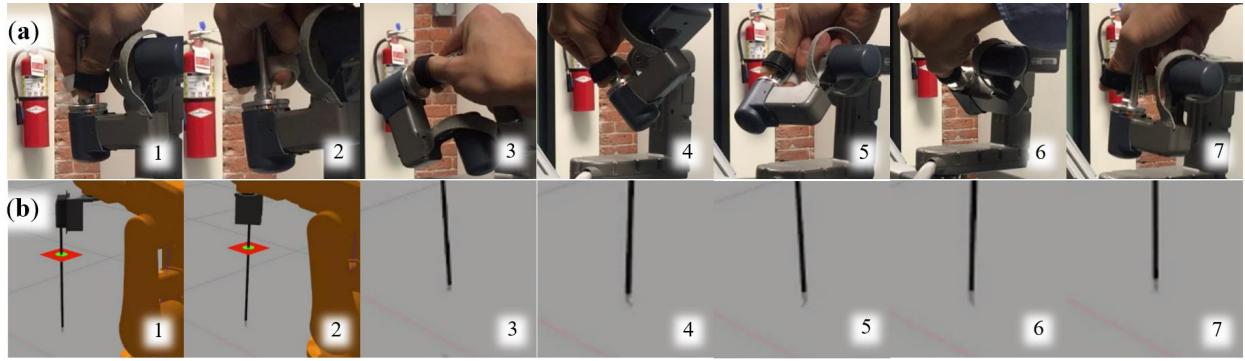


Fig. 9. Orientation motion of teleoperation of the integrated system in Gazebo with the MTM of dVRK

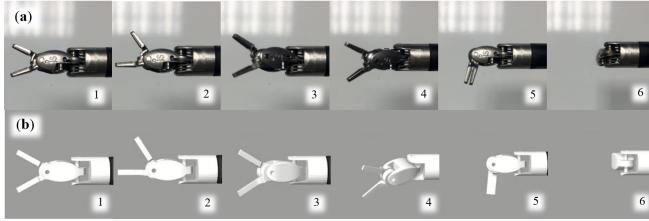


Fig. 10. Motion comparison of the insertion instrument actuated by our developed system and its corresponding Gazebo simulation. a) the insertion instrument of dVRK. b) the simulation of the insertion tool in Gazebo

TABLE IV
GROUP CONTRIBUTION

Task	Name
Kinematics and software	Yan Wang
Mechanical design and manufacturing	Yuqi Jiang
Manufacturing and embedded system of servos	Tianyu Cheng
Servo trajectory control	Kenechukwu Mbanisi
Gazebo simulation	Abhijeet Thakan

instrument of the da Vinci Surgical Robot. A customized mounting device for the insertion instrument is fabricated based on 3D printing. Actuation for the insertion instrument is integrated to the system with four off-the-shelf servos and their corresponding control board. A ROS-based software package is developed which includes kinematics and trajectory generation subject to the RCM constraints. A teleoperation experiment of the integrated system in Gazebo is carried out with the MTM of the dVRK as the input device, which demonstrates the capability of the system. The actuation system for the insertion tool is proven to be able to mimic the same motion of the simulation in Gazebo. Unfortunately, due to the software problem in the robot pendant, we are not able to demonstrate this work on the real robot.

However, there are some corner cases that can make the robot kinematics fail and the trajectory tracking response is not fast enough. These can be the future work of this project.

- [2] J. Li, G. Zhang, A. Müller, and S. Wang, “A family of remote center of motion mechanisms based on intersecting motion planes,” *Journal of Mechanical Design*, vol. 135, no. 9, p. 091009, 2013.
- [3] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, “Modelling and identification of the da vinci research kit robotic arms,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1464–1469, IEEE, 2017.
- [4] M. J. Lum, D. C. Friedman, G. Sankaranarayanan, H. King, K. Fodero, R. Leuschke, B. Hannaford, J. Rosen, and M. N. Sinanan, “The raven: Design and validation of a telesurgery system,” *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1183–1197, 2009.
- [5] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, “Constrained cartesian motion control for teleoperated surgical robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.
- [6] N. Sanchez-Tamayo and J. P. Wachs, “Collaborative robots in surgical research: a low-cost adaptation,” in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 231–232, ACM, 2018.
- [7] H. Sadeghian, F. Zokaei, and S. H. Jazi, “Constrained kinematic control in minimally invasive robotic surgery subject to remote center of motion constraint,” *Journal of Intelligent & Robotic Systems*, pp. 1–13, 2018.
- [8] R. C. Locke and R. V. Patel, “Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery,” in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1900–1905, IEEE, 2007.
- [9] A. Marchese and H. Hoyt, “Force sensing and haptic feedback for robotic telesurgery,” *Worcester polytechnic institute*, 2010.
- [10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.
- [11] N. P. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *IROS*, vol. 4, pp. 2149–2154, Citeseer, 2004.
- [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 *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6434–6439, IEEE, 2014.
- [13] D. Systems, “Openhaptics.” <https://3dsystems.teamplatform.com/pages/102774?t=r4nk8vqwa91>, 2017 (accessed November 30, 2018).
- [14] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.
- [15] H. Azimian, R. V. Patel, and M. D. Naish, “On constrained manipulation in robotics-assisted minimally invasive surgery,” in *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, pp. 650–655, IEEE, 2010.

REFERENCES

- [1] J. H. Palep, “Robotic assisted minimally invasive surgery,” *Journal of Minimal Access Surgery*, vol. 5, no. 1, p. 1, 2009.