Virtual Reality Simulation of a Robotic Laparoscopic Surgical System

Alexios Karadimos[1], Anthony Tzes[2,3], Nikolaos Evangeliou[2] and Evangelos Dermatas[4]

2022 8th International Conference on Control, Decision and Information Technologies (CoDIT) | 978-1-6654-9607-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/CODIT55151.2022.9804076

Abstract— **Virtual reality simulation of robotic-assisted minimal invasive procedures reveals interesting issues related to the perception, control and manipulation of laparoscopic tools with emphasis given to the pivot trajectories and the Remote-Center- of-Motion (RCM) constrained motion planning. In this paper, the Gazebo simulator under Robot Operating System (ROS) allows the inclusion of hardware-in-the-loop for Minimally Invasive Surgery (MIS) procedures. The RCM constraint is addressed through the transformation of the surgical task space into the robot's taskspace, while addressing the robot's manipulability. Emphasis is given in calculating various geometric paths to be followed by the robot during surgery. Simulations were conducted using the ROS framework and the MoveIt kinematic planner using the RRTConnect path planning algorithm to evaluate the efficacy proposed scheme.**

I. Introduction

Robot-assisted minimally invasive surgery (RA-MIS) has revolutionized the surgical theatre, in terms of accuracy, pre-operative planning and reduction of blood loss [1]–[3]. This comes at the cost of RA-MIS requiring the inserted robotic tools to respect the RCM point constraint, to avoid patient discomfort and long post-operative stay [4].

Thus, in order to perform RCM-conforming MIS procedures, industrial overactuated robotic manipulators are utilized, to respect the inherited constraint and maintain adequate intra-operative workspace coverage. In [5], a controller is designed to minimize the deviation from the RCM point using kinematics with the dual quaternion framework. A similar approach is studied in [6], where a controller is designed for an active wrist robot resulting in reduced exerted pressure on the abdominal wall and respecting the RCM constraint while executing the trajectory with smaller error. Overall, a smaller RCM error minimizes the force exerted to the patient and ensures the minimally invasive characteristic of the operation. In [7], this deviation error is studied more thoroughly and is used to study the force interaction model of the surgical tool and the abdominal wall.

Moreover, prior to utilizing online feedback to minimize the RCM error, adequate trajectories can be pre-operatively computed, to allow for smooth motion of the robotic manipulators end-effector. In [8] such trajectories are estimated during a robot-assisted endoscopic surgery using the Least Squares algorithm. This algorithm calculates the intersection of surgical tool axes measured from a number of consecutive robot configurations. A different approach is presented in [9], where the trajectory estimations are performed using an Extended Kalman Filter. In [10] a direct adaptive control system is presented, where a geometrical estimator is used to estimate the outer penetration of the surgical tool w.r.t. the fulcrum point, which is then used in a PI controller and a trajectory generator for pivot motions.

Additionally, an autonomous method for firm laparoscopic tool exchanging and grasping is an emerging research topic. Micro-motions of the tool, owing to insufficient grasping, can result in severe tissue damage to the fulcrum point that needs be resolved [11], [12].

In this paper, a holistic approach for perception and navigation of a complete RA-MIS system is highlighted. The Virtual-Reality (VR) framework relies on Gazebo, ROS and MoveIT offering several advantages to the practicing physician [13]–[16]. The surgeon can detect, grasp and manipulate a laparoscopic surgical tool through extended simulations prior to the actual surgical operation. The highlight of the developed robot-assisted surgical approach is depicted in Figure 1. To achieve this objective, the developed simulation environment comprising mainly of a high efficiency industrial robotic arm is capable of:

- Visually detecting the scene and laparoscopic tools, while calculating the relative position and orientation of the center of mass of each tool through stereoscopic vision.
- Calculating the contact points on the tool, on which the fingers of the gripper will be placed to ensure a firm grasp (force closure).
- Calculating the trajectory to be executed when the tool is inserted in the trocar, since its motion is RCM-constrained. The latter refers to the motion constraint of one point of the inserted tool coinciding with the RCM point.

This paper emphasizes on robotic-assisted MIS simulation, which is important for future works relevant to surgeon training [16] virtual-reality surgical teleoperation [14], [17]. The paper is organized as follows: In Section II the main components of the system, including their fundamental kinematic principles are presented. In Section III, a visual servoing environment is simulated for firm tool grasping and path planning, to be followed by Simulation Studies and Conclusions in Sections IV and V respectively.

[1] Department of Electrical and Computer Engineering, University of Patras, Greece a.karadimos@ieee.org

[2] Electrical Engineering, New York University Abu Dhabi, Abu Dhabi, 129188, UAE

[3] Center for Artificial Intelligence and Robotics, New York University Abu Dhabi, Abu Dhabi, 129188, UAE

[4] Department of Computer Engineering & Informatics, University of Patras, Greece

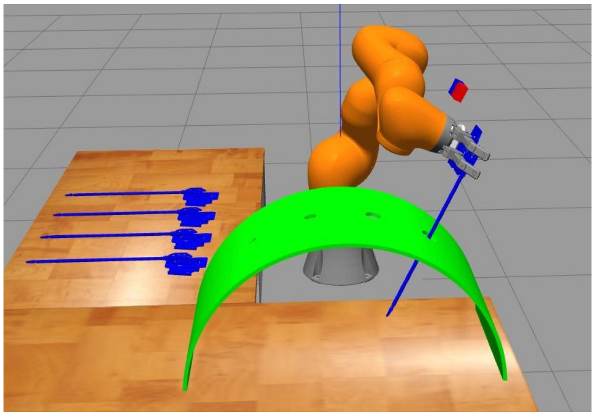

Fig. 1. Simulation setup in Gazebo ROS for MIS using a) the Kuka iiwa robotic arm, b) the Barrett hand gripper, c) the laparoscopic tool, and d) table with laparoscopic tools and a mounting dock.

II. Laparoscopic Robotic System

The proposed robotic system comprises of three parts, namely the robotic arm, the gripper, and an actuated laparoscopic tool.

A. Robot Manipulator

The 7 Degree-of-Freedom (DoF) KUKA iiwa 14 lwr was utilized for this medical simulation, due to its positioning and force feedback accuracy in future real-world scenarios. Before designing the trajectories that the manipulator will have to execute, it is crucial to first study the kinematic equations of the robot, as well as the singularity points that occur from these equations and the mechanical structure and the constraints that must be satisfied.

RCM Constraint. The Remote Center of Motion (RCM) constraint is encountered frequently in MIS. The RCM point, also known as fulcrum point, trocar point, incision point and/or insertion point, is a virtual point in space around which the robot is constrained to rotate (pivot). The RCM constraint means that at all times one point along the axis of the surgical tool must coincide with the RCM point. This constraint reduces the number of DoF from 6 (position and orientation in Cartesian 3D space) to only four:

1) **insertion** and **retraction**: translation along the $\hat{\mathbf{r}}$ vector of the spherical coordinate system of the Fulcrum Reference frame
2) **roll**: rotation around the $\hat{\mathbf{z}}$ unit vector by an angle ϕ
3) **pitch**: rotation around the $\hat{\mathbf{y}}$ unit vector by an angle θ
4) **yaw**: rotation around the $\hat{\mathbf{x}}$ unit vector by an angle ψ

The RCM constraint does not have a direct implication in the Inverse Kinematics (IK) solution, because it only reduces the taskspace relying on a specific set of robot positions and orientations.

Elbow-up constraint The IK typically provides 8 different solutions for a desired pose (position and orientation) of Kuka arm, owing to the increased dexterity. This offers more flexibility per case, as a different solution can be chosen based on user defined conditions (i.e. previous robot pose, collision avoidance). In this paper, there are user-defined cases. The first one refers to choosing a specific robot configuration for picking surgical tools from a robot-side cart. This case does not impose any motion constraints and, thus, any of the IK solutions can be implemented. Secondly, the insertion and pivoting of the surgical tool at the mounting dock is chosen, which impose the RCM constraint and a constraint of collision avoidance between the robot arm and the mounting dock. In the latter use case, to satisfy the collision avoidance, all path planning kinematic solutions are in a elbow-up configuration at all times.

There are two ways to mathematically describe the elbow-up constraint, either using the distance between the robot base and the third link or by using the relative angle of the base link and the third link. From the Kuka iiwa specifications, the following calculations are implemented using the Denavit-Hartenber (DH) distance parameters $d_1 = 360$mm and $d_3 = 420$mm. The distance constraint is $d_{\min} \leq d \leq d_{\max}$, where $d_{\min} = \sqrt{d_1^2 + d_3^2} = 553$mm and $d_{\max} = d_1 + d_3 = 780$mm.

The distance-based description of the elbow-up constraint can not be easily forward-transformed to a description that uses the robot's forward kinematic transformations and reference frames, although it can be easily used after the IK solution to check which solutions satisfy this distance constraint. The angle-based description is preferred, because it directly describes the orientation that Kuka's third reference frame must have, with respect to the reference frame of the base. Thus for each orientation angle θ (pitch, yaw, roll) the following constraint must be satisfied $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$. When the IK problem is solved, some of the solutions may be rejected from the elbow-up constraint. For each solution the forward kinematics up to the third joint must be calculated. Note that although the forward kinematics from the universal frame to the end-effector should be the same for all solutions, the same does not hold for the forward kinematic transformations of the intermediate links, since ${}^0T_3 = {}^UT_0^{-1}\ {}^UT_{tcp}\ {}^3T_{tcp}^{-1}$.

Let the calculated ${}^0T_3$ be ${}^0T_3 = \begin{bmatrix} {}^0R_3 & {}^0\mathbf{p}_3 \\ 0 & 1 \end{bmatrix}$, then the distance that must satisfy the inequality $d_{\min} \leq d \leq d_{\max}$ is $d = \|{}^0\mathbf{p}_3\|$.

In a similar way, the angle version of the elbow-up constraint can be used to check if a solution is accepted. Let the orientation matrix ${}^0R_3$ of the calculated pose ${}^0T_3$ have the following form

$$ {}^0R_3 = [{}^0\hat{\mathbf{x}}_3 \quad {}^0\hat{\mathbf{y}}_3 \quad {}^0\hat{\mathbf{z}}_3] $$

then the angle that must satisfy the inequality $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$ is

$$ \theta = \operatorname{acos}\left(\frac{{}^0\hat{\mathbf{z}}_3 \cdot {}^0\hat{\mathbf{z}}}{\|{}^0\hat{\mathbf{z}}_3\|\|{}^0\hat{\mathbf{z}}\|}\right) $$

where ${}^0\hat{\mathbf{z}}$ is the unit vector of the z-axis of the frame $\{0\}$ with respect to itself, i.e. ${}^0\hat{\mathbf{z}} = [0, 0, 1]^\top$

B. Dexterous 3-finger Hand

The task for the selected gripper is to firmly grasp the laparoscopic tool implying the use of at least a three finger (contact points) configuration to achieve force closure. The used Barrett Hand BH8-282 has 3 fingers, where the first 2 fingers have 3 DoF each and the third one has 2 DoF. Additionally, each finger and the palm of the gripper have arrays of tactile sensors which can be used to measure force and torque. These measurements are important to determine a good grasp using a force control system, as shown in Figure 2, which is mainly comprised of a Proportional-Derivative (PD) controller and the fingers Inverse Kinematics (IK) and forward Kinematics (FK).

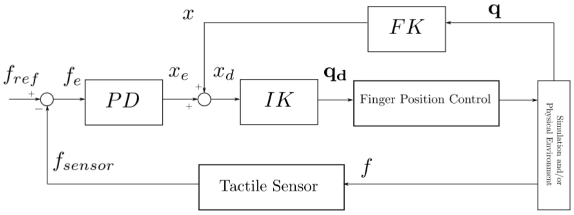

Fig. 2. Force control for a single gripper finger

III. Perception and Navigation of Laparoscopic Tool

A. Perception

For visualization we assume in the simulation that the surgical tools are blue-colored and the mounting dock, where the tools will be placed, is green. These colors make the scene and object recognition much easier without the need of more advanced image processing and/or machine learning recognition algorithms.

The calculation of the candidate grasping points is a problem where we seek to find the points that lie on the contour of the detected object such that the gripper can make a good grasp of the object preferably with force closure. The adopted method is implemented by using an expanding **growing circle**. This circle is initiated with a small radius and has at all times its center at the center of mass of the detected tool. At each step of the method the radius is incremented by a fixed amount and then we check if this circle has any intersection with the contour of the detected tool. If no intersection is found the method proceeds with a new radius. The method is terminated when at least three intersections are found owing to the three finger-configuration of the hand, as shown in Figure 3. In this Figure, the yellow points are the grasp points and the thin black circumscribed circle is the growing circle that was used to calculate them.

The set of intersection points $\mathbb{G}$ is calculated from

$$\mathbb{G} = \underset{(x,y)}{\arg\max} I_1(x,y) \odot I_2(x,y).$$

Given one binary image $I_1(x,y)$ which contains the contour of the detected tool (all white pixels are the contour) and one second binary image $I_2(x,y)$ which contains the growing circle, the set of intersection points are the (x,y) coordinates where the Hadamard product $I_1(x,y) \odot I_2(x,y)$ (element-wise product) of the two binary images is maximum.

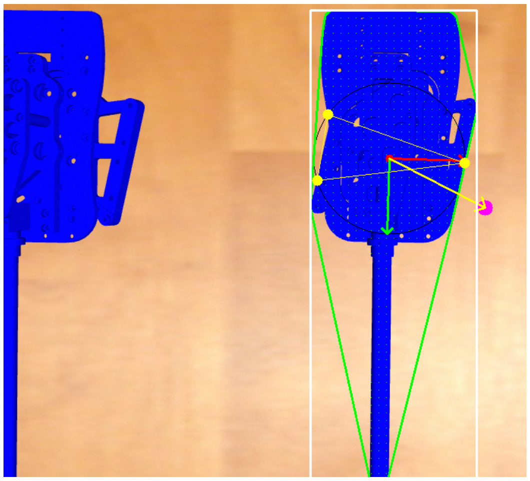

Fig. 3. Image based visual servoing and calculation of grasp points.

Several times multiple pixels are returned from this method due to the ”pixelated nature” of curves. To filter all these pixels and get only one pixel for each intersection point we initialize a list of the final points with the first pixel found and then iterate over all of the other pixels. For each pixel we check if it has a Manhattan distance larger than a small threshold (e.g. 3 pixels) from the pixels that are already added in the list of the final pixels. If two pixels have a Manhattan distance larger than a selected threshold then we consider them to be two different intersection points. Figure 4 shows the execution steps of the algorithm that calculates the candidate grasping points from the intersections of a growing circle and the contour of the detected surgical tool.

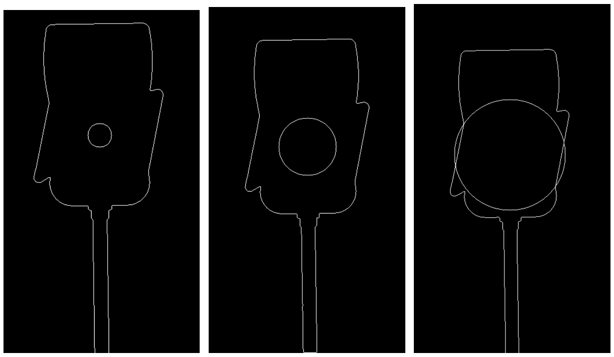

Fig. 4. Candidate grasping point computations.

Image based servoing uses features extracted from the image, as well as positions on the image plane and drives the robot in such a way so that the video frame is changed from an initial view to a final, desired view.

The image based visual servoing control system depicted in Figure 5 consists of the **Image Controller**, the **Robot** and the **Feature Extractor**.

The adopted PD-structure of the Image controller outputs commands to be executed in the plant. These commands are to be used to control the robot in task space, whereas the internal controller of the robot drives each joint to the

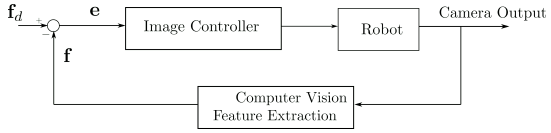

Fig. 5. Image based visual servoing closed loop control

desired angle. The feedback used to calculate the error for the controller, uses the camera's output and, based on that, it calculates the yellow vector (shown in Figure 3) from the detected tool's center of mass to the center of the image frame (feature extraction [18]), as $\mathbf{x}[k+1] = \mathbf{x}[k] + \mathbf{u}[k]$ where $\mathbf{x}[k] = [x, y, z, \theta, \phi, \psi]^\top$. The discrete PID control law is given by

$$\mathbf{u}[k] = K_p \mathbf{e}[k] + K_i \sum_{i=0}^{k-1} \mathbf{e}[i] + K_d\left(\mathbf{e}[k] - \mathbf{e}[k-1]\right)$$

where $\mathbf{e}[k] = [e_x, e_y, 0, e_\theta, 0, 0]^\top$ (assuming 2D motion of the end-effector, parallel to the table with the laparoscopic tools). The system's performance is shown in Figure 6 where the utilized controller parameters are $K_p = 0.9, K_d = 0.2$. The parameters were tuned empirically, starting first with tuning trials of the K_p parameter and then with the K_d parameter. On the top part in the error graphs several spikes appear due to the the sudden temporary detection of a nearby surgical tool. On the bottom image, these spikes are filtered out, and only the error graphs of the visual servoing of one tool are shown.

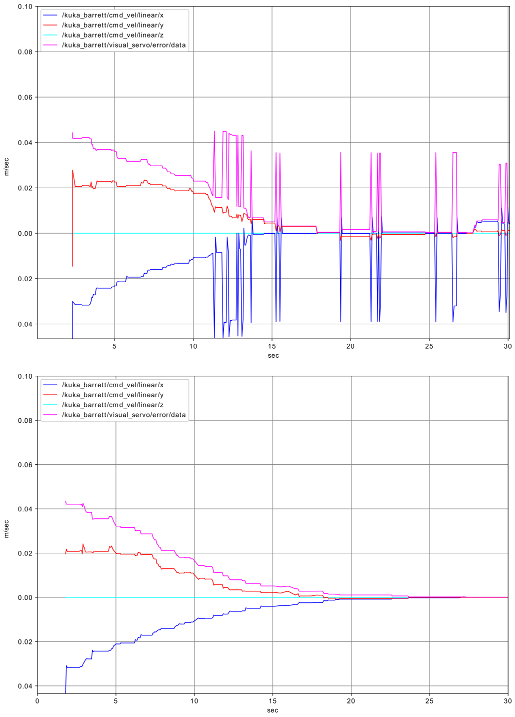

Fig. 6. Visual servo controller error diagrams.

B. Path Planning

Manipulating a laparoscopic tool with a robot requires to properly map the fulcrum point. To do this, the surgeon's hand motions are inverted and scaled with respect to the RCM point, which lies approximately on the center of the incision. As already mentioned, apart from the scaling and inversion, laparoscopic procedures add an additional motion constraint that demands at each time one point of the laparoscopic tool to coincide with the RCM point.

Before designing the path planning, it is important to define the laparoscopic tool pose. The pose is given by the following transformation matrix

$$ {}^FT_B = \begin{bmatrix} {}^FR_B & {}^F\mathbf{p}_B \\ \mathbf{0} & 1 \end{bmatrix}, \quad {}^FR_B = \begin{bmatrix} \hat{\mathbf{x}}_B & \hat{\mathbf{y}}_B & \hat{\mathbf{z}}_B \end{bmatrix}$$

and the orientation vectors can be calculated using spherical coordinate unit vectors

$$\hat{\mathbf{x}}_B = \begin{bmatrix} -\sin\phi \\ \cos\phi \\ \theta \end{bmatrix}, \hat{\mathbf{y}}_B = \begin{bmatrix} \cos\theta\cos\phi \\ \cos\theta\sin\phi \\ -\sin\theta \end{bmatrix}, \hat{\mathbf{z}}_B = \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}$$

The position of the point B is given in spherical coordinates by:

$$ {}^F\mathbf{p}_B = \begin{bmatrix} \rho\sin\beta\cos\alpha \\ \rho\sin\beta\sin\alpha \\ \rho\cos\beta \end{bmatrix} = \rho\hat{\mathbf{r}}$$

where $r = \rho$ is the outside penetration of laparoscopic tool, $\theta = \beta$ is the altitude angle and $\phi = \alpha$ is the orientation angle.

After having defined the tool's pose with respect to the fulcrum reference frame, a geometric path can be designed such that for each point the matrix ${}^FT_B$ is computed. Subsequently, this will be converted to the robot's reference frames i.e. ${}^0T_7 = {}^UT_0^{-1}\,{}^UT_F\,{}^FT_B\,{}^7T_{TCP}^{-1}$ which are then used as input to the IK-problem. The simplest trajectory that is extensively simulated and studied in IV is the line segment trajectory which can be defined in parametric form:

$$\begin{cases} x_F = s_l + x_{F1} = (1-s)x_{F1} + s x_{F2} \\ y_F = s_m + y_{F1} = (1-s)y_{F1} + s y_{F2} \\ z_F = s_n + z_{F1} = (1-s)z_{F1} + s z_{F2} \end{cases}$$

where $(x_{F1}, y_{F1}, z_{F1}), (x_{F2}, y_{F2}, z_{F2})$ are the start and end points respectively of the line segment. Other geometric trajectories (e.g. line segments, circular, helical trajectories etc.) are presented in more detail in [19].

After obtaining a desirable path, the next step is to make sure the robot will execute the trajectory while satisfying the RCM constraint. To do that, an RCM-error metric must be used to implement a control system that guarantees that the laparoscopic tool is always aligned with the fulcrum point. To calculate this error, the line of the long axis of the surgical tool must be first defined. Thus, two points are calculated using the transformation of the surgical tool, which is attached to the robot's TCP on the end-effector. Let the following be the pose of the surgical tool with respect to the global reference frame: ${}^{U}T_{T0} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

Using this pose, let A, B be the points such that

$$\overrightarrow{O_F A} = \mathbf{p}, \text{ and } \quad \overrightarrow{O_F B} = \mathbf{p} + \hat{\mathbf{x}},$$

where O_F is the origin point of the fulcrum reference frame. The the line of interest is defined as the line l that passes through the points A and B. Since both the origin of the fulcrum reference frame and the line of the long axis of the tool are known, then the alignment error can be calculated as the distance of the line l from the point O_F

$$e_{rcm} = d(l, O_F) = \frac{\|\overrightarrow{O_F A} \times \hat{\mathbf{x}}\|}{\|\hat{\mathbf{x}}\|}$$

. In real-case scenarios the exact position and orientation of the uncertain fulcrum points are estimated as in [8], [9], [20].

The error e_{rcm} can provide further information if the distance of the line l (or of the $\hat{\mathbf{x}}$ axis) from the fulcrum point is seen from a different perspective. If this distance is seen from a plane that is perpendicular to $\hat{\mathbf{x}}$, then the line is seen as a point and the distance $d(l, O_F)$ is seen as a distance between 2 points, as illustrated in Figure 7. This perspective of the RCM error (will also be referenced as yz-error e_{yz}) is more useful because it decomposes the error distance in 2 components $e_y = e_{yz} sin(\psi)$ and $e_z = e_{yz} cos(\psi)$ that can be used to correct the goal pose of the robot so that it fixes the RCM misalignment. The angle ψ that is used to split e_{rcm} in the two components, is already known from the robot's pose and it is the yaw (also known as spin) angle of the surgical tool. Similar approaches to calculate this error/distance are presented in [6]–[8].

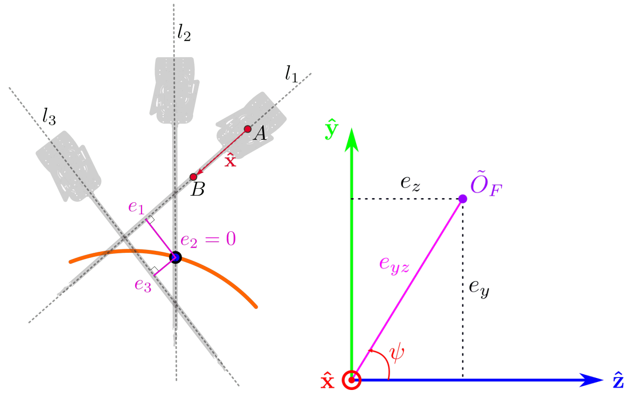

Fig. 7. (left) Geometric calculation of the RCM alignment error e using the distance between the line l and the RCM point, (right) RCM error calculation in yz plane. The RCM error ([5]) or yz-error is the distance between the line of the $\hat{\mathbf{x}}$ vector (here seen as a point) and the estimated position of the origin of the fulcrum reference frame $\tilde{O}_F$

If $e_{rcm} \leq 1$mm then the surgical tool axis passes through the fulcrum point and executes RCM trajectories, otherwise the robot is considered to have slipped from alignment, it does not execute RCM motion and probably generates a force $f \propto e$ which exerts pressure to the tissue around the incision point. This in turn can have negative side-effects in patient's recovery and, thus, needs to be highly avoided. In [6] a different distance e_s is calculated along the x-axis of the fulcrum reference frame for the RCM error, as shown in Figure 8. Using this deviation distance e_s the force interaction between the surgical tool and the abdominal wall is $\|\mathbf{f}_s\| = \lambda e_s$, where λ is the elasticity of the abdominal wall and can be measured experimentally and $e_s = \frac{1}{cos\gamma} e$.

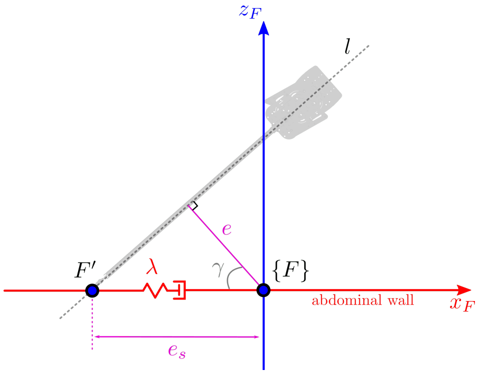

Fig. 8. Force interaction model of the laparoscopic tool and the abdominal wall around the fulcrum point

Using the estimated RCM error $\tilde{e}_{rcm}$ and the estimated position of the origin of the fulcrum reference frame, an adaptive motion control system can be designed that corrects the trajectory to avoid RCM misalignment. The proposed control system is illustrated in Figure 9 following along a similar pivoting motion control system in [10], whereas similar control schemes have appeared in [5], [21].

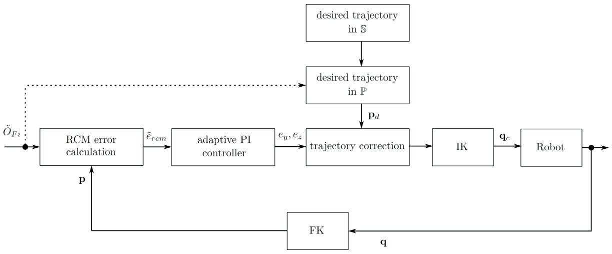

Fig. 9. RCM tracking adaptive control system.

IV. Simulation Studies

A line segment trajectory is selected inside the surgical taskspace which will then be transformed via the fulcrum transformation to a trajectory that the robotic arm can execute. Rather than defining the boundary points of this line, we define a segmented trajectory using the coordinates of the start point and a direction of this line. In Figure 10, the overall motion is decomposed into: (a) a preparatory path to achieve elbow-up pose, (b) a transformed line segment trajectory for the end-effector to follow, (c) a transformed line segment trajectory with respect to the tool base frame, (d) the fulcrum reference frame 2 $\{F_2\}$, (e) a line segment trajectory in surgical taskspace, and (f) the line of the axis along the length of the surgical tool, which is used to calculate the distance error (RCM deviation). The collision-free path was calculated using the RRTConnect path-planning algorithm [22].

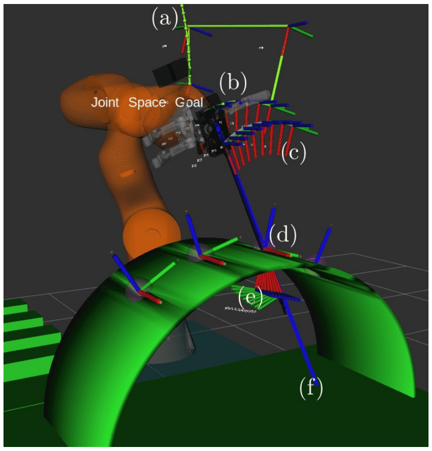

Fig. 10. Line segment trajectory creation.

In Figure 11 the RCM error values are shown. The *x*-axis shows the ROS time in seconds and the *y*-axis the RCM error in logarithmic scale. The robot starts from the home position where the RCM error is larger, then approaches the fulcrum point, where the error decreases and then it executes the line segment pivot trajectory where the error is bounded in the magnitude of less than a micrometer then stays constant, then changes again during the execution of the reverse line-segment trajectory and then finally it stays constant while the robot stays still in the initial insertion position. The RCM error while the robot is inserted but still and the error while the robot executes a pivot trajectory is shown in Table I.

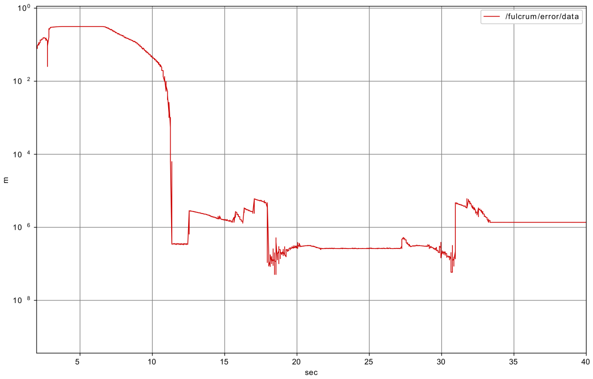

Fig. 11. Achieved RCM error in meters (logarithmic scale) with respect to simulation time (seconds).

TABLE I
REPEATABILTY AND AVERAGE TRAJECTORY ERROR

	Average [μm] (accuracy)	Standard Deviation [μm] (repeatability)	sample size
while pivoting	2.11	1.61	2309
while inserted and still	0.29	0.29	2696

In Figure 12, the repeatability from 10 iterations is shown where the left diagram indicates the distribution of all measurements, the middle one shows distribution of measurements while the robot was pivoting, and the rightmost the distribution of measurements upon robot insertion.

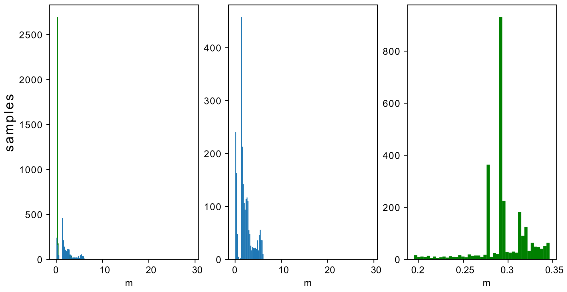

Fig. 12. RCM error distributions.

The virtual reality simulation scenario allowed the examination of the manipulability $\mathscr{M} = \sqrt{det(JJ^\top)}$ index throughout the execution of the trajectory, as shown in Figure 13. It should be mentioned that a similar dexterity metric $\mathscr{D}$ has been reported in [23] which also includes information about the proximity of the robot from the taskspace boundary.

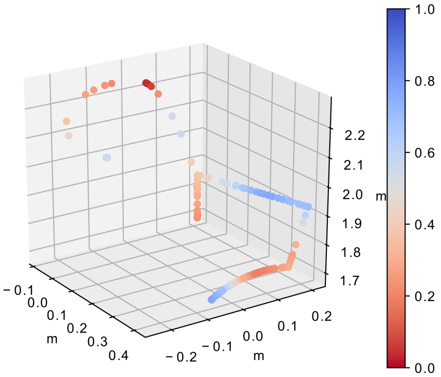

Fig. 13. Manipulability $\mathscr{M} = \sqrt{det(JJ^\top)}$ evolution during executed trajectory.

The developed software is available at [24] . A video demo of this paper can be found at [25] and the following Table is indicative of the presented results in the video.

V. Conclusions

Simulation enhanced Robotic MIS can reveal issues related to several factors affecting the operation including the robot's positional and orientation accuracy, minimizing the operating time, addressing the RCM constraint while selecting suitable path planning algorithms, and researching into the motion's repeatability and manipulability of the robot. The generated research has provided the framework

TABLE II
VIDEO-ENHANCED SIMULATION RESULTS

Simulation studies	Trajectories with **RRTConnect**	
	elbow-up preparatory path	
10 Experiments	Elbow-up Start pose planning time (sec)	Execution status
Average	0.1742	1
Standard deviation	0.0490	-
10 Experiments	Elbow-up preparation path planning time (sec)	Execution status
Average	0.1170	1
Standard deviation	0.0842	-
	Approach & Insertion	
10 Experiments	Approach fulcrum 2 path planning time (sec)	Execution status
Average	0.1164	1
Standard deviation	0.0880	-
10 Experiments	Insertion path planning time (sec)	Execution status
Average	0.2495	1
Standard deviation	0.0789	-
	Line segment pivot trajectories	
10 Experiments	Line segment path planning time (sec)	Execution status
Average	1.8090	1
Standard deviation	2.4214	-
10 Experiments	Reverse line segment path planning time (sec)	Execution status
Average	5.3566	0.7
Standard deviation	0.0868	-

for addressing these issues and allows the control designer to fine tune the suggested controllers.

REFERENCES

[1] V. Vitiello, S.-L. Lee, T. P. Cundy, and G.-Z. Yang, "Emerging robotic platforms for minimally invasive surgery," *IEEE reviews in biomedical engineering*, vol. 6, pp. 111–126, 2012.

[2] E. A. Arkenbout, P. W. Henselmans, F. Jelínek, and P. Breedveld, "A state of the art review and categorization of multi-branched instruments for NOTES and SILS," *Surgical endoscopy*, vol. 29, no. 6, pp. 1281–1296, 2015.

[3] C. E. Díaz, R. Fernández, M. Armada, and F. García, "A research review on clinical needs, technical requirements, and normativity in the design of surgical robots," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 4, p. e1801, 2017.

[4] C. Gruijthuijsen, L. Dong, G. Morel, and E. V. Poorten, "Leveraging the fulcrum point in robotic minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2071–2078, 2018.

[5] M. M. Marinho, M. C. Bernardes, and A. P. Bó, "A programmable remote center-of-motion controller for minimally invasive surgery using the dual quaternion framework," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE, 2014, pp. 339–344.

[6] E. Bauzano, V. Muñoz, I. Garcia-Morales, and B. Estebanez, "Control methodologies for endoscope navigation in robotized laparoscopic surgery," in *International Conference on Research and Education in Robotics*. Springer, 2009, pp. 11–22.

[7] E. Bauzano, V. Munoz, I. Garcia-Morales, and B. Estebanez, "Active wrists endoscope navigation in robotized laparoscopic surgery," in *2009 IEEE International Conference on Mechatronics*. IEEE, 2009, pp. 1–6.

[8] L. Dong and G. Morel, "Robust trocar detection and localization during robot-assisted endoscopic surgery," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4109–4114.

[9] C. Gruijthuijsen, L. Dong, G. Morel, and E. Vander Poorten, "Leveraging the fulcrum point in robotic minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2071–2078, 2018.

[10] V. F. Muñoz, J. Gómez-de Gabriel, I. García-Morales, J. Fernández-Lozano, and J. Morales, "Pivoting motion control for a laparoscopic assistant robot and human clinical trials," *Advanced Robotics*, vol. 19, no. 6, pp. 694–712, 2005.

[11] S. Abeywardena, Q. Yuan, A. Tzemanaki, E. Psomopoulou, L. Droukas, C. Melhuish, and S. Dogramadzi, "Estimation of tool-tissue forces in robot-assisted minimally invasive surgery using neural networks," *Frontiers in Robotics and AI*, vol. 6, 2019.

[12] T. Liu and M. C. Cavusoglu, "Needle grasp and entry port selection for automatic execution of suturing tasks in robotic minimally invasive surgery," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 552–563, 2016.

[13] M. J. Fiedler, S.-J. Chen, T. N. Judkins, D. Oleynikov, and N. Stergiou, "Virtual reality for robotic laparoscopic surgical training." *Studies in Health Technology and Informatics*, vol. 125, pp. 127–129, 2007.

[14] Y. Jo, Y. J. Kim, M. Cho, C. Lee, M. Kim, H.-M. Moon, and S. Kim, "Virtual reality-based control of robotic endoscope in laparoscopic surgery," *International Journal of Control, Automation and Systems*, vol. 18, no. 1, pp. 150–162, 2020.

[15] H. R. Patel, "Simulation training in laparoscopy and robotic surgery," *Journal of Visualized Surgery*, vol. 3, 2017.

[16] H. R. Patel and J. V. Joseph, *Simulation training in laparoscopy and robotic surgery*. Springer Science & Business Media, 2012.

[17] O. A. Van der Meijden and M. P. Schijven, "The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review," *Surgical endoscopy*, vol. 23, no. 6, pp. 1180–1190, 2009.

[18] N. Aghakhani, M. Geravand, N. Shahriari, M. Vendittelli, and G. Oriolo, "Task control with remote center of motion constraint for minimally invasive robotic surgery," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 5807–5812.

[19] A. Karadimos, "Perception, control and path planning of robotic laparoscopic surgical system," Master's thesis, Electrical and Computer Engineering, University of Patras, Greece, 2022. [Online]. Available: http://nemertes.library.upatras.gr/jspui/handle/10889/15791

[20] B. Rosa, C. Gruijthuijsen, B. Van Cleynenbreugel, J. V. Sloten, D. Reynaerts, and E. V. Poorten, "Estimation of optimal pivot point for remote center of motion alignment in surgery," *International journal of computer assisted radiology and surgery*, vol. 10, no. 2, pp. 205–215, 2015.

[21] C. D. Pham, F. Coutinho, A. C. Leite, F. Lizarralde, P. J. From, and R. Johansson, "Analysis of a moving remote center of motion for robotics-assisted minimally invasive surgery," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1440–1446.

[22] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[23] G. Gras, K. Leibrandt, P. Wisanuvej, P. Giataganas, C. A. Seneci, M. Ye, J. Shang, and G.-Z. Yang, "Implicit gaze-assisted adaptive motion scaling for highly articulated instrument manipulation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4233–4239.

[24] A. Karadimos, "Github - karadalex/surgery_robotics_kuka_barrett," 2022. [Online]. Available: https://github.com/karadalex/surgery_robotics_kuka_barrett

[25] ——, "Thesis demo1 - robotic surgical tool manipulator," 2022. [Online]. Available: https://youtu.be/lfV1vdHf7bk