Pre-operative Offline Optimization of Insertion Point Location for Safe and Accurate Surgical Task Execution

Francesco Cursi*,[1,2], Petar Kormushev[2]

***Abstract*— In robotically assisted surgical procedures the surgical tool is usually inserted in the patient's body through a small incision, which acts as a constraint for the motion of the robot, known as remote center of Motion (RCM). The location of the insertion point on the patient's body has huge effects on the performances of the surgical robot. In this work we present an offline pre-operative framework to identify the optimal insertion point location in order to guarantee accurate and safe surgical task execution. The approach is validated using a serial-link manipulator in conjunction with a surgical robotic tool to perform a tumor resection task, while avoiding nearby organs. Results show that the framework is capable of identifying the best insertion point ensuring high dexterity, high tracking accuracy, and safety in avoiding nearby organs.**

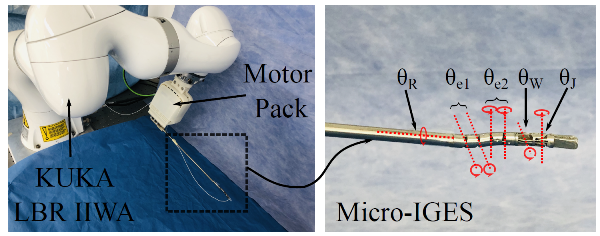
I. Introduction

In robotically assisted surgical procedures, the surgical tool is inserted inside the patient's body through small incisions. This, however, restricts the motion of the robot, which is not allowed to move tangentially to the hole plane. Yet, it can only pivot and translate in the insertion direction about the insertion point. The insertion point generates the Remote Center of Motion (RCM), whose constraint leads the robotic system to lose two Degrees of Freedom (DOFs) [1].

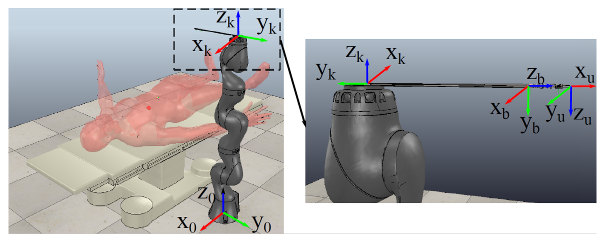
Different approaches have been developed to overcome the issue of loss of mobility due to the RCM motion, and they can typically be divided into design-based and control-based strategies [2]. However, the location of the insertion point, which is responsible for the RCM, highly affects the performances of the surgical robot.

Thus far, few works have focused on finding the optimal insertion point for a surgical operation. In [3] the authors used CT images to manually identify the set of possible port locations and then applied a search space method among the possible port locations to find the best one. In this work, however, the robot could achieve the RCM motion thanks to its mechanical design. In [2], instead, the robotic setup consists of a serial-link manipulator and a surgical tool. This work is concerned with the problem of choosing a location for the RCM relative to the manipulator in order to maximize the system performances.

Most of the works focus only the system's dexterity, without assessing the motion accuracy or possible presence of organs and obstacles that might interfere with the surgical task. The main contribution of this manuscript is thus to provide a pre-operative offline framework for finding the optimal insertion point location in order to guarantee safe and accurate surgical task execution, when operating with a surgical robot. This work focuses on the use of a KUKA robot, on which the Micro-IGES surgical robotic tool [4] is attached, to simulate a tumor resection task (Figure 1). However, the framework can be generalized to any other robotic structure and task.



(a) The KUKA+Micro-IGES robot in the real environment



(b) Simulation environment for the KUKA+Micro-IGES robot

Fig. 1: The KUKA+MicroIGES robot in: 1a) the real environment; 1b) the simulation environment.

The framework consists of two main parts: the control strategy, and the optimization strategy. For the control problem, the Hierarchical Quadratic Programming (HQP) [5] approach is adopted, since it allows multiple prioritized tasks to be optimally solved while satisfying additional constraints, such as joint limits and avoiding obstacles (like organs or bones). With regards to the optimization, a space search optimization is employed to identify the best port location in a defined space. A custom-made fitness function for the optimization is introduced which allows motion accuracy, system's dexterity, and safety in avoiding collision with nearby organs to be considered. However, the optimal configuration identified by the solver may reside in a region where small deviations from the optimal may lead to much worse performances. Consequently, a resilience to error strategy to find an optimal solution in a neighborhood that guarantees good performances is also introduced. Being it an offline pre-operative framework, results are shown in this paper based on a simulation environment, which allows the

*Corresponding author
[1]Hamlyn Centre, Imperial College London, Exhibition Road, London, UK.
[2]Robot Intelligence Lab, Imperial College London, Exhibition Road, London, UK.
Email: [f.cursi17,p.kormushev]@imperial.ac.uk

capabilities of the robot to be assessed when many different insertion point locations are assigned.

The paper is structured as follows. In Section II the Micro-IGES robotic surgical tool is described and the kinematic model of the whole KUKA and Micro-IGES robot is derived. Section III describes the proposed framework, presenting the control strategy, the optimization method, and the resilience to error approach. Results are then shown in Section IV and, finally, conclusions are drawn in Section V.

II. Robotic System Description

In this section an overview of the robotic system setup is presented, describing the kinematic models of the Micro-IGES surgical robotic tool, of the KUKA LBR IIWA robotic arm, and of the whole system.

A. Micro-IGES Kinematic Model

A detailed description of the Micro-IGES robot can be found in [4], [6]. Because of the generally nonlinear motor to joint (and joint to motor) mapping $\boldsymbol{q}_u = \boldsymbol{f}(\boldsymbol{\theta}_u)$, being $\boldsymbol{\theta}_u$ the vector of motor positions and $\boldsymbol{q}_u$ the vector of joint positions, the kinematic model of the robot, with respect to its base frame ($\{RF_b\}$ in Figure 1b), can be rewritten as ${}^b\boldsymbol{T}_u = {}^b\boldsymbol{T}_q(\boldsymbol{q}_u) = {}^b\boldsymbol{T}_u(\boldsymbol{\theta}_u)$ and ${}^b\boldsymbol{v}_u = {}^b\boldsymbol{J}_q(\boldsymbol{q}_u)\dot{\boldsymbol{q}}_u = {}^b\boldsymbol{J}_q(\boldsymbol{q}_u)\boldsymbol{L}(\boldsymbol{q}_u)\dot{\boldsymbol{\theta}}_u = {}^b\boldsymbol{J}_u(\boldsymbol{\theta}_u)\dot{\boldsymbol{\theta}}_u$, where ${}^b\boldsymbol{T}_u \in \mathbb{R}^{4\times4}$ is the Cartesian end-effector pose, ${}^b\boldsymbol{v}_u \in \mathbb{R}^6$ the Cartesian end-effector twist (linear and angular velocities vector), ${}^b\boldsymbol{J}_q$ is the Cartesian task Jacobian with respect to the joint variables, and ${}^b\boldsymbol{J}_u$ the Jacobian with respect to the motor values. The matrix $\boldsymbol{L}$, with $L_{i,j} = \frac{\partial q_i}{\partial \theta_j}$, is the motor to joint differential matrix [7]. In this work we employed the same motor to joint mapping as in [8]. The control problem for the Micro-IGES can therefore be formulated as a function of the motor values, which can be directly measured and controlled. The system's state is expressed by $\boldsymbol{\theta}_u = \begin{bmatrix}\theta_R & \theta_{e1} & \theta_{e2} & \theta_W & \theta_J\end{bmatrix}^T$, representing the Roll, Elbow 1, Elbow 2, Wrist, and Jaws motor values.

B. KUKA Robotic Arm

The KUKA LBR IIWA is an articulated industrial robot with 7 joints. Because of its serial-link manipulator design and direct transmission (the motors are placed directly at the joints), the kinematic model of this robot can be easily computed by means of the Denavit-Hartenberg convention. Therefore, the kinematics of the KUKA can be expressed with respect to its base frame $\{RF_0\}$ as ${}^0\boldsymbol{T}_k = {}^0\boldsymbol{T}_k(\boldsymbol{q}_k)$ and ${}^0\boldsymbol{v}_k = {}^0\boldsymbol{J}_k(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k$, where $\boldsymbol{q}_k \in \mathbb{R}^7$ is the vector of KUKA's joint positions.

C. Total System Kinematics

The total robotic system is composed by the Micro-IGES and the KUKA robots (Figure 1), with the Micro-IGES motor package directly attached to the KUKA end-effector. Due to the lack of a proper CAD model of the motor pack, in this work the Micro-IGES is considered to be directly attached to the KUKA's end-effector, just neglecting a vertical translation. In total, the system has 12 Degrees of Freedom (DOF), 7 for the KUKA and 5 for the Micro-IGES and the robot's state is then described by $\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_k & \boldsymbol{\theta}_u\end{bmatrix}^T$.

The robot end-effector pose, with respect to the KUKA base, can be computed as $\boldsymbol{T}(\boldsymbol{q}) = {}^0\boldsymbol{T}_k(\boldsymbol{q}_k)\,{}^k\boldsymbol{T}_b\,{}^b\boldsymbol{T}_u(\boldsymbol{\theta}_u)$, where ${}^0\boldsymbol{T}_k$ is the KUKA end-effector pose, ${}^k\boldsymbol{T}_b \in \mathbb{R}^{4\times4}$ is a fixed transformation matrix from the KUKA end-effector to the Micro-IGES base frame, and ${}^b\boldsymbol{T}_u$ is the Micro-IGES pose with respect to its base.

Being $\boldsymbol{\omega}_k = \boldsymbol{J}_{\omega,k}\dot{\boldsymbol{q}}_k \in \mathbb{R}^3$ the KUKA's end-effector angular velocity, with $\boldsymbol{J}_{\omega,k} \in \mathbb{R}^{3\times7}$ and $\boldsymbol{J}_{\omega,u} \in \mathbb{R}^{3\times5}$ the KUKA and micro-IGES's orientation Jacobian, $\boldsymbol{J}_{v,k} \in \mathbb{R}^{3\times7}$, $\boldsymbol{J}_{v,k} \in \mathbb{R}^{3\times5}$ the Jacobians of for the linear velocity of each robot, it then results that the macro-micro manipulator's end-effector velocities can be computed as:

$$\begin{aligned}\dot{\boldsymbol{P}} &= \begin{bmatrix}{}^0\hat{\boldsymbol{J}}_{v,k} & {}^0\boldsymbol{J}_{v,u}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{q}}_k \\ \dot{\boldsymbol{\theta}}_u\end{bmatrix} = {}^0\boldsymbol{J}_{v,tot}\dot{\boldsymbol{q}} \\ \boldsymbol{\omega} &= {}^0\boldsymbol{\omega}_k + {}^0\boldsymbol{\omega}_u = \begin{bmatrix}{}^0\boldsymbol{J}_{\omega,k} & {}^0\boldsymbol{J}_{\omega,u}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{q}}_k \\ \dot{\boldsymbol{\theta}}_u\end{bmatrix} = {}^0\boldsymbol{J}_{\omega,tot}\dot{\boldsymbol{q}}\end{aligned}, \tag{1}$$

with $\hat{\boldsymbol{J}}_{v,k} = \boldsymbol{J}_{v,k} + \begin{bmatrix}\boldsymbol{J}_{\omega,1}\times\boldsymbol{P}_{u,k} & \dots & \boldsymbol{J}_{\omega,7}\times\boldsymbol{P}_{u,k}\end{bmatrix}$.

The total system's end-effector twist is then computed as:

$$\boldsymbol{v} = \begin{bmatrix}\dot{\boldsymbol{P}} \\ \boldsymbol{\omega}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{v,tot} \\ \boldsymbol{J}_{\omega,tot}\end{bmatrix}\dot{\boldsymbol{q}} = \begin{bmatrix}\boldsymbol{J}_{KUKA} & \boldsymbol{J}_{Iges}\end{bmatrix}\dot{\boldsymbol{q}} = \boldsymbol{J}_{tot}\dot{\boldsymbol{q}}\ , \tag{2}$$

with $\mathbf{J_{KUKA}} \in \mathbb{R}^{6\times7}$ and $\mathbf{J_{Iges}}\mathbb{R}^{6\times5}$ describing the contribution of each robot to the motion.

III. Method

In this section, the proposed pre-operative framework is presented, introducing the control strategy and the optimization method.

A. Surgical Task Plan

Identifying the optimal insertion point location is typically task specific. In this work we are roughly simulating a tumor resection task, which starts with the whole KUKA and micro-IGES manipulator at the home position, with the KUKA upright and the micro-IGES straight. Once the insertion point location is specified, the whole manipulator is required to reach the keyhole. In this phase of the motion, the micro-IGES is not enabled, but kept straight. Only the final accuracy in reaching the insertion point is considered, since generally in a real scenario the robot is manually moved to the insertion point. The insertion point location dictates the direction along which the surgical tool will be inserted. In fact, during the insertion phase, the tip moves in a straight line connecting the insertion point to the tumor, while keeping the RCM and the controller tries to keep the micro-IGES straight, as it would be in a real surgical operation. Once the targeted tumor is reached, the whole system is commanded to get to a desired cutting orientation. In this work, the cutting orientation is perpendicular to the tumor's plane. Finally, the tumor is cut, tracking its contour with the desired cutting orientation, while keeping the RCM.

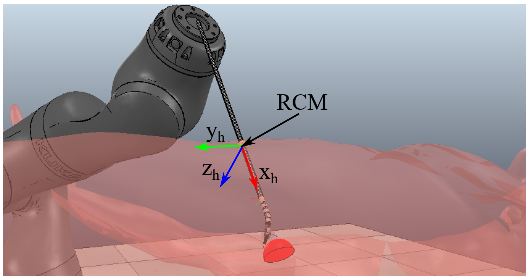


Fig. 2: Remote Center of Motion representation

B. Motion Control

In order to optimally exploit the hyper redundancy of the whole manipulator, different motion subtasks can be specified. Each subtask can be assigned a certain priority. This ensures that the subtasks do not conflict with each other, guaranteeing optimality in the control. Different approaches exist in the literature that address the redundancy problem in robotic surgery [9]–[11]. Recently, especially in the field of humanoid robotics, where researchers have to deal with high level of redundancy, HQP [5] is widely used to solve stacks of prioritized tasks subjected to different constraints (both equality and inequality). In a surgical application like tumor resection at least two motion subtasks can be specified: keeping the RCM fixed and follow a desired path (in order to remove the tumor). However, during the surgical task the robot may be required to avoid possible obstacles such as bones or organs, that may be in the proximity of the tumor. Therefore, beside keeping the RCM and accurately tracking a desired path, obstacle avoidance may be assigned as an additional subtask.

1) Remote Center of Motion: The RCM is a point where only rotations and insertion motion can be achieved. During laparoscopic procedures, for instance, the RCM coincides with the location of the hole on the patient's body where the surgical tool is then inserted, as in Figure 2. Therefore, the motion at the RCM needs to be constrained along the surface tangential directions (y, z in Figure 2). Consequently, the primary motion subtask is to constrain the RCM and the corresponding cost function can be specified as $||\tilde{\boldsymbol{v}}_{RCM} - \boldsymbol{J}_{RCM}\dot{\boldsymbol{q}}||^2$, where $\tilde{\boldsymbol{v}}_{RCM}$ is a desired RCM velocity. $\boldsymbol{J}_{RCM}$ allows the velocity of the RCM to be computed based on the robot kinematics.

The position of the RCM with respect to the robot base is computed as $\boldsymbol{P}_{RCM} = {}^0\boldsymbol{P}_k + {}^0\boldsymbol{R}_k\, {}^k\boldsymbol{P}_{RCM,k}$, with ${}^k\boldsymbol{P}_{RCM,k}$ being the RCM position with respect to the KUKA's end-effector, in the end-effector's frame. By deriving with respect to time, as in (1), it turns out that:

$$\begin{aligned}\dot{\boldsymbol{P}}_{RCM} = &{}^0\boldsymbol{J}_{v,k}\dot{\boldsymbol{q}}_k + \\ &\begin{bmatrix}{}^0\boldsymbol{J}_{\omega,1} \times {}^0\boldsymbol{P}_{RCM,k} & \ldots & {}^0\boldsymbol{J}_{\omega,7} \times {}^0\boldsymbol{P}_{RCM,k}\end{bmatrix}\dot{\boldsymbol{q}}_k = \\ &\begin{bmatrix}{}^0\hat{\boldsymbol{J}}_{RCM} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{q}}_k \\ \dot{\boldsymbol{\theta}}_u\end{bmatrix} = {}^0\boldsymbol{J}_{RCM}\dot{\boldsymbol{q}}\end{aligned} \tag{3}$$

To simplify the control problem, in this work the RCM motion is expressed with respect to the insertion point frame in Figure 2. Therefore $\boldsymbol{J}_{RCM} = \begin{bmatrix}{}^h\boldsymbol{R}_0\, {}^0\boldsymbol{J}_{v,RCM} \\ {}^h\boldsymbol{R}_0\, {}^0\boldsymbol{J}_{\omega,RCM}\end{bmatrix}$, with ${}^0\boldsymbol{R}_h = {}^h\boldsymbol{R}_0^T$ the rotation matrix of the hole frame with repect to the KUKA base frame, and ${}^0\boldsymbol{J}_{v,RCM}, {}^0\boldsymbol{J}_{\omega,RCM}$ the linear and orientation part of the RCM Jacobian. Thanks to this transformation, assuming the hole fixed, the RCM motion to consider is only along the y, z components. Consequently, $\tilde{\boldsymbol{v}}_{RCM} = \begin{bmatrix}0 & 0\end{bmatrix}^T$ in order to guarantee the RCM motion in the tangential directions to be fixed and $\boldsymbol{J}_{RCM} \in \mathbb{R}^{2\times 12}$. However, this formulation also allows non-null hole velocities to be specified, as it could be in case of breathing.

2) Path Tracking: The secondary motion subtask is executing a desired motion, which, in our case is to have the total system's end-effector follow the desired path during insertion and tumor resection. Given a reference Cartesian path and its corresponding desired velocity $\tilde{\boldsymbol{v}}_P$, with $m_p \leq 6$, from (2) the cost function associated to the secondary motion task can be expressed as $||\tilde{\boldsymbol{v}}_P - \boldsymbol{J}_{tot}\dot{\boldsymbol{q}}||$. In this work the path tracking subtask is expressed in terms of both Cartesian position and orientation, therefore $m_p = 6$ and considers both the approaching motion to reach the tumor inside the patient and the path tracking to cut the tumor.

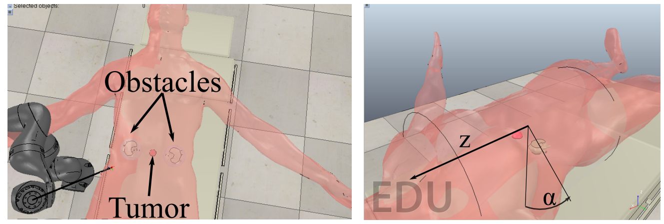


Fig. 3: Simulation environment with the tumor resection task showing the kidneys acting as obstacles to avoid and the body approximation used for the optimization strategy.

3) Obstacle Avoidance: Figure 3 shows the simulation environment with the obstacles being the kidneys of the patient. In order to avoid the obstacles, a repulsive field is generated. Since the dynamics of the system is not considered in this work, the repulsive field is expressed in terms of velocities. For sake of simplicity, a spherical bounding volume for the organs is generated. Any more complex shape can also be used. The bounding volumes are responsible for the repulsive fields, which are considered to have the following form:

$$\boldsymbol{v}_{rep} = k_1 e^{-k_2 \frac{\boldsymbol{d}^T\boldsymbol{d}}{\delta^2}} \frac{\boldsymbol{d}}{||\boldsymbol{d}||}\ , \tag{4}$$

where $k_1 = 20, k_2 = 5$ are user-defined constants on the strength and extension of the repulsive field, $\delta = 5$ cm is the radius of the bounding volume, and $\mathbf{d} \in \mathbb{R}^3$ is the distance of a point from the center of the bounding volume. In order to ensure that the surgical tool steers away from the obstacle, the repulsive field is considered to act only on the two points on the tool that are closer to each organ. The tool is therefore discretized into 34 points, 28 on the shaft and 6 on the articulated part (one for each joint). During the motion, it is continuously checked which points are closer to the two obstacles, and the two closest ones are considered. These two points will be at distances $\boldsymbol{d}_1, \boldsymbol{d}_2$ from the organs to which the two repulsive velocities $\boldsymbol{v}_{obs,1}, \boldsymbol{v}_{obs,2}$, computed from (4), will correspond. Moreover, the two points are associated with the Jacobians ${}^0\boldsymbol{J}_{obs,1}, {}^0\boldsymbol{J}_{obs,2}$ with respect to the whole system's base frame.

In order to avoid stalling situations when the repulsive velocities are in the opposite direction of the motion of the two points, only the components of the repulsive field perpendicular to the direction of motion of the two points are taken into consideration. Even though the robot manages to move unimpeded into the obstacle if traveling radially, this motion would be penalized in the optimization.

C. Prioritized Motion Control

In order to guarantee accurate execution of the specified tumor resection task, while avoiding collisions with surrounding obstacles, the HQP problem is formulated as:

$$\begin{aligned} &\text{for } n = 1\dots 3 \\ &\dot{\boldsymbol{q}}_n = \arg\min_{\dot{\boldsymbol{q}}} \quad \frac{1}{2}||\tilde{\boldsymbol{v}}_n - \boldsymbol{J}_n\dot{\boldsymbol{q}}||^2 + \frac{1}{2}||\Lambda\dot{\boldsymbol{q}}||^2 \\ &\qquad \text{s.t} \quad \boldsymbol{J}_1\dot{\boldsymbol{q}} = \boldsymbol{J}_1\dot{\boldsymbol{q}}_1 \\ &\qquad\qquad \vdots \\ &\qquad \boldsymbol{J}_{n-1}\dot{\boldsymbol{q}} = \boldsymbol{J}_{n-1}\dot{\boldsymbol{q}}_{n-1} \\ &\qquad \frac{\boldsymbol{q}_m - \boldsymbol{q}}{dt} \leq \dot{\boldsymbol{q}} \leq \frac{\boldsymbol{q}_M - \boldsymbol{q}}{dt} \end{aligned} \quad , \quad (5)$$

where $n = 1$ corresponds to guaranteeing RCM motion ($\boldsymbol{J}_1 = \boldsymbol{J}_{RCM},\ \tilde{\boldsymbol{v}}_1 = \tilde{\boldsymbol{v}}_{RCM}$) and $n = 2$ is for the path tracking ($\boldsymbol{J}_2 = \boldsymbol{J}_{tot},\ \tilde{\boldsymbol{v}}_2 = \tilde{\boldsymbol{v}}_P$). For the obstacle avoidance $n = 3$, the task is specified as $\boldsymbol{J}_3 = \boldsymbol{J}_{obs} = \begin{bmatrix} \boldsymbol{J}_{obs,1} \\ \boldsymbol{J}_{obs,2} \end{bmatrix}$ and $\tilde{\boldsymbol{v}}_3 = \boldsymbol{v}_{obs} = \begin{bmatrix} \boldsymbol{v}_{obs,1} \\ \boldsymbol{v}_{obs,2} \end{bmatrix}$, which guarantees that the two critical points move away from the obstacles. The HQP formulation guarantees that lower priority subtasks (n) are solved by exploiting the system's redundancies and without conflicting with higher priority subtasks ($n-1$). This means that the solution of high priority subtasks needs to be included as an equality constraint when solving for lower priority subtasks ($\boldsymbol{J}_{n-1}\dot{\boldsymbol{q}} = \boldsymbol{J}_{n-1}\dot{\boldsymbol{q}}_{n-1}$). In our formulation, the highest priority subtask is constraining the RCM, the secondary subtask is path tracking, and obstacle avoidance is the lowest priority one. For the primary RCM motion task, $\boldsymbol{J}_0 = \boldsymbol{0}$. $\boldsymbol{\Lambda} = diag(\lambda_{\mathbf{k}}, \lambda_{\mathbf{u}})$ is a diagonal weighting matrix which is used to reduce joint velocities when the systems are close to singularity. The last inequality in (5) guarantees that the joints and motor commands of the whole system are within their bounds ($\boldsymbol{q}_m, \boldsymbol{q}_M$). dt is the motion sampling time.

D. Optimization of Insertion Point Location

With regards to choosing the location of the insertion point, it is important to consider the information provided by the surgeons, who make their decision depending on the surgery to be undertaken and the anatomy of the patient. Therefore, in our framework the surgeon can specify a certain region on the patient's body where the insertion point might be. This region is then discretized and a space search optimization is performed. In this work we approximate the torso of the patient as a cylinder with specified radius and assume that the insertion point will be on the surface of the cylinder. For this reason, we choose two independent variables for the search space: the distance from the center of the torso z and the angle around the torso α (Figure 3). We are assuming that $z \in [-15, 15]$ cm and $\alpha \in [90, 210]°$, with a discretization of 5 cm and 5°, respectively. The search space optimization will iterate through the defined discretized locations and choose the best one.

1) Fitness Function: In order to find the optimal insertion point location, a fitness function needs to be chosen and minimized. This function must be defined such as to take into consideration different factors such as the RCM motion, the tip positioning, the system's dexterity, and the obstacle avoidance while performing the desired surgical task.

To perform the path tracking and cut the tumor, the path is discretized in N_t time steps, and (5) is used to find the joint commands at each time step. The tracking accuracy can therefore be expressed in terms of the value of the cost function achieved by solving (5). It is worth mentioning, that the number of time steps (and thus the total completion time) is fixed and it is the same for any insertion point location. Assuming that $\dot{\boldsymbol{q}}^*$ is the optimal joint commands from (5), the cost at each time step $t = 0 \dots N_t$ is computed as:

$$c_t = \frac{1}{2}||\tilde{\boldsymbol{v}}_{RCM} - \boldsymbol{J}_{RCM}\dot{\boldsymbol{q}}^*|| + \frac{1}{2}||\tilde{\boldsymbol{v}}_P - \boldsymbol{J}_{tot}\dot{\boldsymbol{q}}^*|| \quad , \quad (6)$$

with each $c_t \in [0, \infty]$. Then the average $\bar{c}$ and the maximum c_M costs can be obtained as $\bar{c} = \frac{1}{N_t}\sum_{t=0}^{N_t} c_t$ and $c_M = max(c_0, \dots, c_{N_t})$. The cost fitness function is then defined as $f_c = \frac{1}{\bar{c}c_M} \geq 0$: configurations with small tracking costs, would have high fitness cost values. Such choice is made in order to penalize configurations that might have overall smaller tracking costs $\bar{c}$, but large instantaneous peaks c_M.

Beside the motion accuracy, it is also important to have configurations that improve the dexterity of the system. Generally, the manipulability ellipsoid [12] is used in robotics to assess the dexterity of a system. Different measures exist to assess the dexterity of a robot, as reported in [13], but their derivations only consider one single motion task. Therefore, in this work the following modification is carried out. If joint limits were not imposed and obstacle avoidance not accounted for, the solution of (5) would be the same as the one obtained from the Gradient Projection Method and would be defined by $\dot{\boldsymbol{q}} = \boldsymbol{J}_1^\dagger\tilde{\boldsymbol{v}}_1 + (\boldsymbol{J}_2\boldsymbol{\mathcal{P}}_1)^\dagger(\tilde{\boldsymbol{v}}_2 - \boldsymbol{J}_2\boldsymbol{J}_1^\dagger\tilde{\boldsymbol{v}}_1)$ [14], with $\boldsymbol{\mathcal{P}}_1 = \boldsymbol{I} - \boldsymbol{J}_1^\dagger\boldsymbol{J}_1$, and $\tilde{\boldsymbol{v}}_1, \boldsymbol{J}_1$ correspond to the RCM motion task, and $\tilde{\boldsymbol{v}}_2, \boldsymbol{J}_2$ to the path tracking task. In order to take into account joint limits, a penalization function is used [13]. Each Jacobian column $j = 1 \dots 12$ is penalized for each task $n = 1, 2$ such that $\boldsymbol{J}_{n,j} = s_j\boldsymbol{J}_{n,j}$ and $s_j = \frac{1 - e^{-4K\frac{(q_{M,j}-q_j)(q_j-q_{m,j})}{(q_{M,j}-q_{m,j})^2}}}{1-e^{-K}}$, where $K = 10$ is a positive constant. This allows the contribution of a specific joint to be zero when it reaches its joint limits. From the definition of the

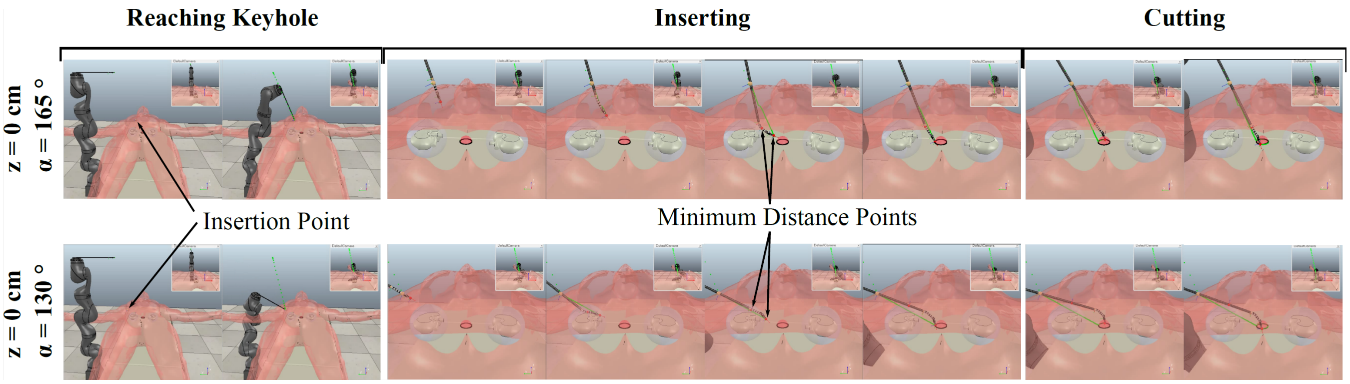


Fig. 4: Snapshots of the surgical task plan at two different insertion point locations.

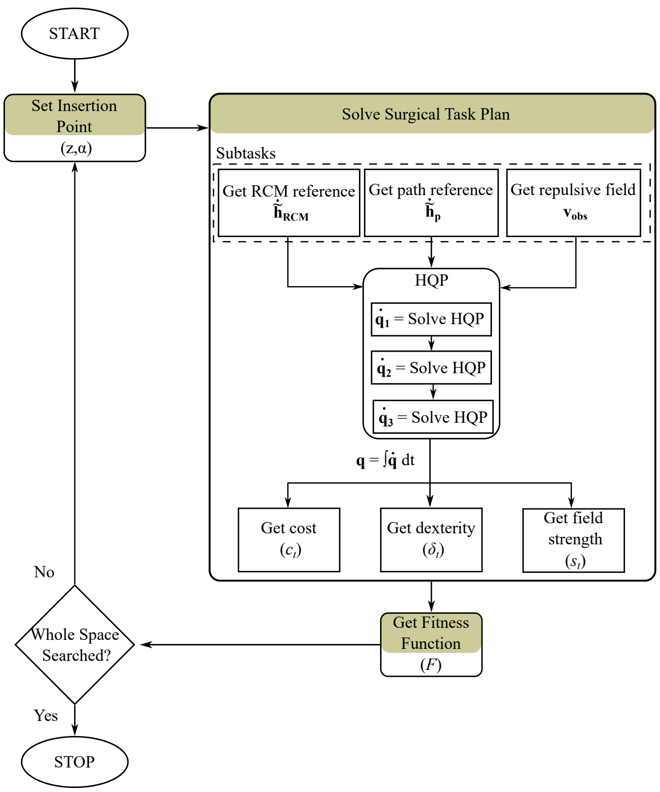


Fig. 5: Flow chart showing the framework to identify the optimal insertion point location. For each insertion point location z, α, the desired motion task is solved and the associated fitness function F computed.

manipulability ellipsoid, following the Jacobian penalization, the multitask manipulability ellipsoid can be defined as $\dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} = \begin{bmatrix} \tilde{\boldsymbol{v}}_1^T & \tilde{\boldsymbol{v}}_2^T \end{bmatrix} \boldsymbol{H} \begin{bmatrix} \tilde{\boldsymbol{v}}_1 \\ \tilde{\boldsymbol{v}}_2 \end{bmatrix}$ with $\boldsymbol{H} \in \mathbb{R}^{8 \times 8}$. If only one single motion task is specified, this formulation reduces to the traditional manipulability ellipsoid formulation. To assess the system's dexterity, the order-independent manipulability measure is used [13]. Due to the time discretization of the motion, at each time step the dexterity measure is defined as:

$$\delta_t = (det(\boldsymbol{H}))^{1/n_j} \ , \tag{7}$$

with $n_j = 12$ being the number of joints of the system, and $\delta_t \in [0, \infty]$. The average $\bar{\delta} = \frac{1}{N_t} \sum_{t=0}^{N_t} \delta_t$ and the minimum $\delta_m = min(\delta_0, \ldots, \delta_{N_t})$ dexterities are computed and utilized to define the dexterity fitness function $f_d = \bar{\delta}\delta_m \geq 0$: configurations with high dexterity maximize the dexterity fitness function.

To account for the obstacle avoidance, the optimal insertion point location should ensure that the robot passes far from the obstacles, and, therefore, the repulsive fields for the two critical points should be small. At each time step, the total repulsive field strength is computed as:

$$s_t = ||\boldsymbol{v}_{obs,1}|| + ||\boldsymbol{v}_{obs,2}|| \tag{8}$$

and then the average $\bar{s} = \frac{1}{N_t} \sum_{t=0}^{N_t} s_t$ and the maximum $s_M = max(s_0, \ldots, s_{N_t})$ are used to define the field strength fitness function $f_s = \frac{1}{\bar{s} s_M} \geq 0$, which decreases as the tool gets closer to the organs. Similarly to the cost, this function penalizes configurations that might have overall small repulsive fields $\bar{s}$, but large instantaneous peaks s_M.

Finally, the fitness function to minimize in order to find the optimal insertion point location is computed as:

$$F = -f_c f_d f_s \ , \tag{9}$$

with the negative sign being added to ensure the minimization of the fitness function. The fitness function is considered to be unitless and the possible differences in the order of magnitude in the measures used do not affect the optimization, since they are not summed together but rather multiplied or divided in the computation of F. Figure 5 shows the flow chart of the framework and Figure 4 shows some snapshots of the surgical task plan for two different insertion point locations.

2) Resilience to Errors: One issue that may arise from the optimization, is that the optimal insertion point location identified by the solver may reside in a region where small deviations from the optimal may lead to much worse performances, yielding higher costs and lower dexterities. This is what could happen in case of positioning errors of the surgical tool on the patient's body. In order to overcome it, the following adjustment is made.

Given an insertion point location defined by $L^0 = [z^0, \ \alpha^0]$, neighbouring regions in the search space to this location are considered. In this work we consider three neighbouring regions with sizes of $l_z = 1, 3, 5$ cm and $l_\alpha = 1, 3, 5°$ for the z and α variables respectively. Each

Algorithm 1 Algorithm for penalization of the costs, dexterities, and filed strengths based on neighbouring regions.

function $F^* \leftarrow$ REGIONPENALTY(L^0, l_z, l_α)
$\triangleright$ Get neighbouring regions
  $\mathcal{R} \leftarrow$ getRegions(L^0, l_z, l_α)
  $S = 12$ $\triangleright$ Total Number of neighbouring locations
$\triangleright$ Get values for current configuration
  $\bar{c}_0, c_{M,0}, \bar{\delta}_0, \delta_{m,0}, \bar{s}_0, s_{M,0} \leftarrow$ SolveMotionTask(L^0)
  $s = 1$
  for $L^s \in \mathcal{R}$ **do**
    $\bar{c}_s, c_{M,s}, \bar{\delta}_s, \delta_{m,s}, \bar{s}_0, s_{M,0} \leftarrow$ SolveMotionTask(L^s)
    $s = s + 1$
  end for
  $\Delta_1 = \max(|\bar{c}_1 - \bar{c}_0|, \dots, |\bar{c}_S - \bar{c}_0|)$
  $\Delta_2 = \max(|c_{M,1} - c_{M,0}|, \dots, |c_{M,S} - c_{M,0}|)$
  $\Delta_3 = \max(|\bar{\delta}_1 - \bar{\delta}_0|, \dots, |\bar{\delta}_S - \bar{\delta}_0|)$
  $\Delta_4 = \max(|\delta_{m,1} - \delta_{m,0}|, \dots, |\delta_{m,S} - \delta_{m,0}|)$
  $\Delta_5 = \max(|\bar{s}_1 - \bar{s}_0|, \dots, |\bar{s}_S - \bar{s}_0|)$
  $\Delta_6 = \max(|s_{M,1} - s_{M,0}|, \dots, |s_{M,S} - s_{M,0}|)$
$\triangleright$ Penalize current configuration values
  $\left.\begin{array}{l} \bar{c}_0^* = \bar{c}_0\Delta_1 \\ c_{M,0}^* = c_{M,0}\Delta_2 \end{array}\right\} f_c^* = \frac{1}{\bar{c}_0^* c_{M,0}^*}$
  $\left.\begin{array}{l} \bar{\delta}_0^* = \frac{\bar{\delta}_0}{\Delta_3} \\ \delta_{m,0}^* = \frac{\delta_{m,0}}{\Delta_4} \end{array}\right\} f_d^* = \bar{\delta}_0^* \delta_{m,0}^*$
  $\left.\begin{array}{l} \bar{s}_0^* = \bar{s}_0\Delta_5 \\ s_{M,0}^* = s_{M,0}\Delta_6 \end{array}\right\} f_s^* = \frac{1}{\bar{s}_0^* s_{M,0}^*}$
  return $F^* = -f_c^* f_d^* f_s^*$
end function

neighbouring region $\mathcal{R}$ is a set of 4 new locations such that $\mathcal{R} = \{L^s, s = 1 \dots 4 \mid L^1 = [z^0 + l_z,\ \alpha^0 + l_\alpha],\ L^2 = [z^0 - l_z,\ \alpha^0 + l_\alpha],\ L^3 = [z^0 - l_z,\ \alpha^0 - l_\alpha],\ L^4 = [z^0 + l_z,\ \alpha^0 - l_\alpha]\}$. In total, 12 neighbouring locations are added. This choice of the neighbouring region is determined by the assumption the the decision variables for the insertion point locations are just the displacement z and the angle α.

For each neighbouring location L^s in $\mathcal{R}$, the motion task is solved and $\bar{c}, c_M, \bar{\delta}, \delta_m, \bar{s}, s_M$ are computed. Then, the deviations of the costs, dexterities and field strengths of each neighbouring location with respect to the initial location L^0 are computed. The initial location L^0 is associated with $\bar{c}_0, c_{M,0}, \bar{\delta}_0, \delta_{m,0}, \bar{s}_0, s_{M,0}$. The maximum deviations are then used as penalization terms. In this way, configurations leading to large deviations will be associated with higher penalized costs and field strengths and smaller penalized dexterities. Algorithm 1 describes the penalization process.

IV. RESULTS

Due to the chosen discretization of the optimization variables z, α for the space search, in total 175 possible insertion point locations were analyzed. Figure 6 shows the distributions of the costs, dexterities, and filed strengths and of the proposed fitness functions in the discretized space. It can be noticed that the insertion point locations near the organs have larger tracking costs $\bar{c}, c_M$ and repulsive field strengths $\bar{s}, s_M$ throughout the whole motion. Larger tracking costs occur also for large angles α because those insertion point locations are much further from the robot base and lead to more joints reaching their limits. The optimal location identified by our framework results to be $z = -15$ cm and $\alpha = 165°$, to which corresponds a value of the fitness function $F = -13.3 \cdot 10^9$. The fact that the optimal location resides on the border of the search space is due to the organs being in the middle of the patient's body. The safest configurations to avoid the organs are those at the borders of z or those at large angles α. However, those at larger angles lead to poor performances in terms of dexterity and tracking accuracy.

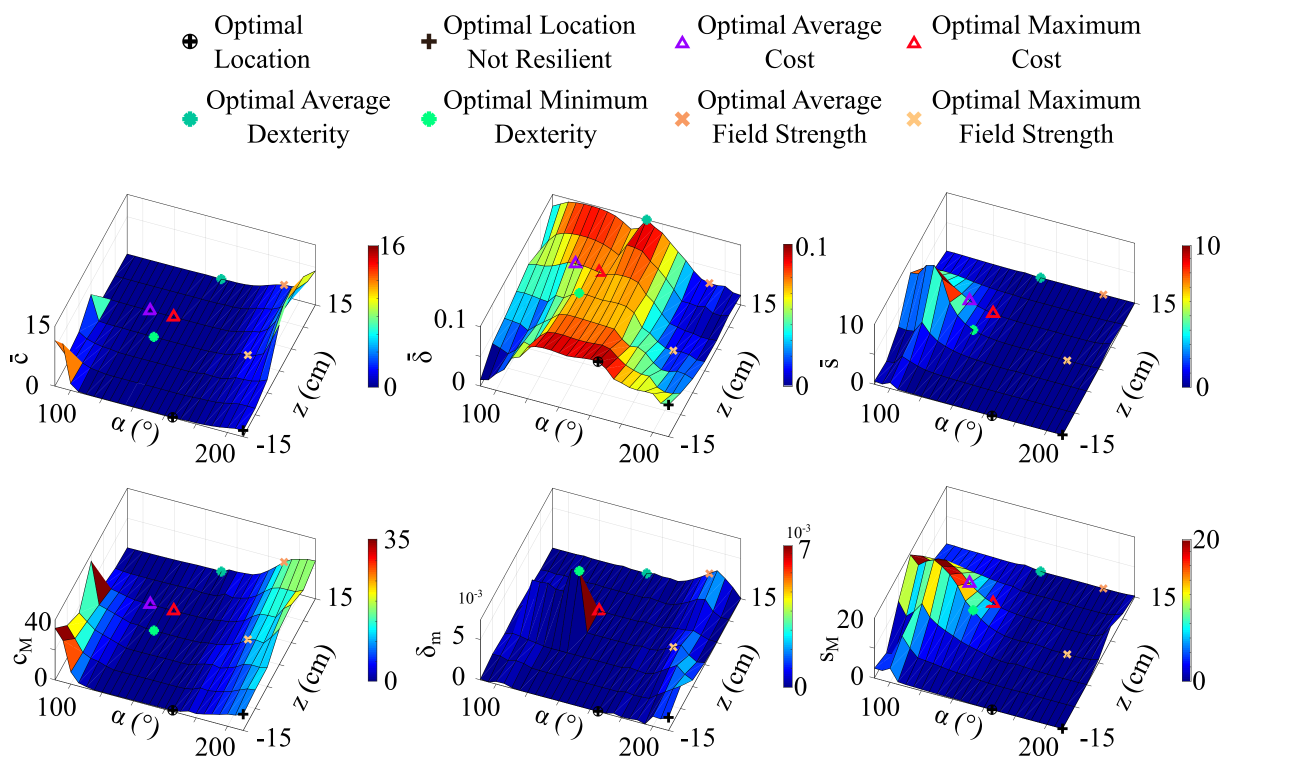
Table Ia reports the costs, dexterities, and field strength values for the locations optimizing each measure independently. Both from Figure 6 and Table Ia it is clear that optimizing for one single measure independently may lead to much worse values in the other measures of interest. For instance, the location $z = 15$ cm, $\alpha = 190°$ which minimizes the average field strength during the motion, has very high costs. This leads to poor overall performances and small fitness function values (Figure 6).

The optimal solution found $z = -15$ cm and $\alpha = 165°$, instead, allows good dexterities, low costs and field strengths to be achieved, thus guaranteeing accurate path tracking and safe distancing from the obstacles. Furthermore, it is worth noticing the effect of taking into consideration neighbouring regions in order to reduce the deterioration of the performances due to inaccurate positioning of the insertion point. If the neighbouring regions were not considered, the optimal insertion point location would be $z = -15$ cm and $\alpha = 210°$. However, as reported in Table Ib, this location resides in a not safe region. In fact, large deviations occur in its surrounding locations, especially in the dexterity measures. The optimal port location, instead, is in a safer region, with much smaller variations.

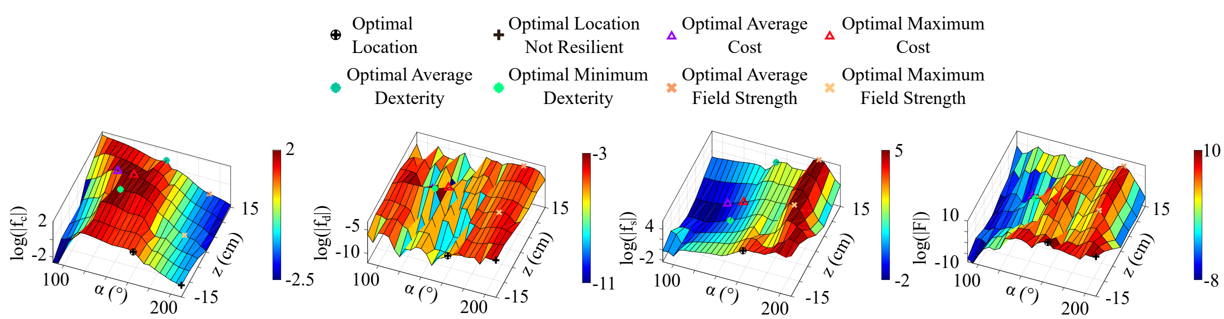
V. CONCLUSIONS

In conclusion, in this work we proposed an offline pre-operative framework for identifying the best location of the insertion point of a surgical robotic tool on a patient's body, in order to improve safety and accuracy of the surgical task execution. The framework considers different aspects to ensure safety during the surgical procedure, such as accurate tracking of the RCM and of the desired path, system's dexterity, and avoidance of nearby organs. Results show that the location of the insertion point has strong effects on the performances of the robotic system, justifying the need for an optimal location.

The proposed framework is task-specific, so it needs to be run every time a different surgical task is specified. However, being it pre-operative and offline, this is not a major limitation. Moreover, it can be used for different surgical tasks and different robots, just by adapting the



(a) Average and maximum costs, average and minimum dexterities, and average and maximum field strengths



(b) The cost, dexterity, field strength, and the overall fitness functions.

Fig. 6: Distributions of the costs, dexterities, and field strengths used to compute the fitness function in the discretized insertion point locations space.

TABLE I: Cost, dexterities, field strengths, and fitness function values for each optimal port location, and maximum deviations in the neighbouring regions.

(a) Cost, dexterities, field strengths, and fitness function values for each optimal port location.

Name	**Location**		$\bar{\mathbf{c}}$	$\mathbf{c_M}$	$\bar{\delta}$	$\delta_{\mathbf{m}}$	$\bar{\mathbf{s}}$	$\mathbf{s_M}$	$\lvert\mathbf{F}\rvert$
	$\mathbf{z(cm)}$	$\alpha(^\circ)$							
Optimal Average Cost	5	120	**0.0365**	0.2302	0.0796	$7.21 \cdot 10^{-7}$	1.4041	6.5605	0.0069
Optimal Maximum Cost	5	135	0.0464	**0.1520**	0.0758	$1.56 \cdot 10^{-5}$	0.3403	1.9778	32.78
Optimal Average Dexterity	15	150	0.1232	1.2404	**0.1012**	$3.43 \cdot 10^{-8}$	0.0185	0.0663	$1.38 \cdot 10^{3}$
Optimal Minimum Dexterity	0	130	0.0469	0.2078	0.0743	**0.0074**	0.8930	6.2803	$1.43 \cdot 10^{5}$
Optimal Average Field Strength	15	190	3.0620	18.8779	0.0224	0.0021	**0.0010**	0.0024	$1.57 \cdot 10^{7}$
Optimal Maximum Field Strength	0	190	1.9406	11.1727	0.0197	0.0011	0.0011	**0.0023**	$3.11 \cdot 10^{7}$
Optimal Location	-15	165	0.0970	0.7749	0.0946	$1.41 \cdot 10^{-5}$	0.0051	0.0185	$\mathbf{13.3 \cdot 10^{9}}$
Optimal Location Not Resilient	-15	210	1.6734	10.846	0.0535	0.0016	0.0012	0.0029	$3.47 \cdot 10^{9}$

(b) Maximum deviations in the neighbouring regions in terms of average and maximum cost (Δ_1, Δ_2),average and minimum dexterity (Δ_3, Δ_4), average and maximum repulsive field strength (Δ_5, Δ_6).

Name	$\mathbf{\Delta_1}$	$\mathbf{\Delta_2}$	$\mathbf{\Delta_3}$	$\mathbf{\Delta_4}$	$\mathbf{\Delta_5}$	$\mathbf{\Delta_6}$
Optimal Location	0.0266	$1.41 \cdot 10^{-5}$	0.0889	1.2583	0.0086	0.0389
Optimal Location Not Resilient	0.0290	0.0014	0.4243	2.3701	0.0011	0.0087

kinematic model and reformulating the fitness function, if necessary.

Future work will focus on improving the framework by including more organs and approximating them with more realistic anatomical shapes, generating more complex repulsive fields, and better discretizing the patient's body.

ACKNOWLEDGMENT

The authors would like to thank Digby Chappell from Robot Intelligence Lab, Imperial College London, for providing valuable feedback and discussion.

REFERENCES

[1] 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, 5 2013, pp. 5807–5812.

[2] R. C. O. Locke and R. V. Patel, "Optimal Remote Center-of-Motion Location for Robotics-Assisted Minimally-Invasive Surgery," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 4 2007, pp. 1900–1905.

[3] A. L. Trejos, R. V. Patel, I. Ross, and B. Kiaii, "Optimizing port placement for robot-assisted minimally invasive cardiac surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 4, pp. 355–364, 12 2007.

[4] J. Shang, K. Leibrandt, P. Giataganas, V. Vitiello, C. A. Seneci, P. Wisanuvej, J. Liu, G. Gras, J. Clark, A. Darzi, and G.-Z. Yang, "A Single-Port Robotic System for Transanal Microsurgery—Design and Validation," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1510–1517, 2017.

[5] A. Escande, N. Mansard, and P.-b. Wieber, "Hierarchical quadratic programming : Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[6] F. Cursi and G.-Z. Yang, "A Novel Approach for Outlier Detection and Robust Sensory Data Model Learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11 2019, pp. 4250–4257.

[7] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1994.

[8] K. Leibrandt, P. Wisanuvej, G. Gras, J. Shang, C. A. Seneci, P. Giataganas, V. Vitiello, A. Darzi, and G.-Z. Yang, "Effective Manipulation in Confined Spaces of Highly Articulated Robotic Instruments for Single Access Surgery," *Icra 2017*, vol. 2, no. 3, pp. 1–1, 2017.

[9] J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 6 1996.

[10] H. Azimian, R. V. Patel, and M. D. Naish, "On constrained manipulation in robotics-assisted minimally invasive surgery," in *2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2010*, 2010, pp. 650–655.

[11] F. Alambeigi, S. Sefati, and M. Armand, "A Convex Optimization Framework for Constrained Concurrent Motion Control of a Hybrid Redundant Surgical System," in *Proceedings of the American Control Conference*, vol. 2018-June. Institute of Electrical and Electronics Engineers Inc., 8 2018, pp. 1158–1165.

[12] T. Yoshikawa, "Force Control of Robot Manipulators," in *IEEE International Conference on Robotics and Automation*, 2000.

[13] D. Zhang, F. Cursi, and G.-Z. Yang, "WSRender: A Workspace Analysis and Visualization Toolbox for Robotic Manipulator Design and Verification," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3836–3843, 10 2019.

[14] S. Chiaverini, G. Oriolo, and I. D. Walker, "Kinematically Redundant Manipulators," in *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 245–268.