

RCM-constrained manipulator trajectory tracking using differential kinematics control

Omar Rayyan¹, Vinicius Gonçalves¹, Nikolaos Evangeliou² and Anthony Tzes¹

Abstract—This paper proposes an approach for controlling surgical robotic systems, while complying with the Remote Center of Motion (RCM) constraint in Robot-Assisted Minimally Invasive Surgery (RA-MIS). In this approach, the RCM-constraint is upheld algorithmically, providing flexibility in the positioning of the insertion point and enabling compatibility with a wide range of general-purpose robots. The paper further investigates the impact of the tool's insertion ratio on the RCM-error, and introduces a manipulability index of the robot which considers the RCM-error that it is used to find a starting configuration. To accurately evaluate the proposed method's trajectory tracking within an RCM-constrained environment, an electromagnetic tracking system is employed. The results demonstrate the effectiveness of the proposed method in addressing the RCM constraint problem in RA-MIS.

I. INTRODUCTION

Minimally Invasive Surgery (MIS) involves performing an operation through small incisions, with the goal of minimizing tissue disruption. This approach is attributed to reducing patients' trauma, resulting in faster recovery and lower hospitalization costs [1]. The advent of Robot-Assisted MIS (RA-MIS) in recent years has further propelled these benefits, allowing surgeons to perform more precise movements with greater agility [2]–[5].

In RA-MIS, the workspace of the robot's tool-tip is confined to the interior space of the patient's body, which can only be accessed through a designated insertion point. Once the tool-tip of the robot passes through this point, all subsequent movements made by the robot must adhere to the constraints imposed by its location, avoiding any contact with its surrounding boundary. This kinematic constraint resulting from the fulcrum point is widely recognized in the field of RA-MIS as the RCM-constraint. Any deviation from this constraint can lead to an unsafe contact force being exerted on the patient's skin, potentially causing more extensive wounds and longer recovery times. Therefore, minimizing drift from the RCM while maintaining precise tool alignment is of high importance when performing a surgical operation.

The methods employed to maintain the RCM-constraint in RA-MIS are generally categorized in mechanical and software-based approaches. In mechanical-based approaches, the RCM-constraint is maintained inherently through the mechanical structure of the robot. A widely adopted mechanical mechanism makes use of the dual-parallelogram design. In such a system, the RCM-constraint is upheld by ensuring

the intersection of the adjacent sides of parallelogram linkages [6]. This mechanism is also utilized by the da Vinci surgical system, one of the leading commercialized surgical systems nowadays. On the other hand, software-based approaches control the manipulator while actively ensuring the RCM-constraint is maintained through the utilization of software-based algorithms.

Mechanical approaches, although robust, come with several limitations. Their reliance on the physical design of the robot restricts them into having a single fixed RCM. This results in a significantly reduced adaptability, as it mandates alignment of the insertion point with the RCM prior to each operation. It is also worth mentioning that mechanically constraining a robot to an RCM greatly limits the available space above the insertion point, thereby reducing the range of motion for a robotic arm and obstructing access [7].

Software approaches, offer a greater flexibility to surgeons by enabling them to position the RCM directly on any insertion point algorithmically, eliminating the need for an ideal operation table positioning [7]. Moreover it can potentially accommodate non-typical insertion points on the patient's body and its inherent non-reliance on specialized robot mechanisms, enables the achievement of the RCM-constraint on a wide range of manipulators.

The main objective of this paper is to propose an enhanced framework for actively controlling a kinematically redundant manipulator while adhering to the RCM-constraint. The proposed control approach makes use of the task function approach and operates at a joint-velocity control level. Within the robot-task's function framework, the tool-tip tracking task is treated as a hard constraint while also ensuring that the task dynamics align with the RCM-constraint, ultimately minimizing any deviation from the fulcrum point. In Subsection II-D, comparisons will be done between the presented approach and other approaches also trying to maintain the RCM-constraint algorithmically. Moreover, the accuracy of the proposed method is verified using an external position measurement system rather than using the inherent feedback of the robot.

The core contributions of this paper are: a) Formulation of a minimal task function that considers the RCM-constraint in Subsection (II-B), b) Analysis of the tool insertion ratio effect on the RCM-error in Subsection (II-E), c) Quantification of a manipulability index which considers the RCM-error in Subsection (II-F), that is then used to generate an initial configuration, and d) Validation of the tool-tip trajectory tracking accuracy through the usage of Aurora electromagnetic tracking system in Subsection (III-B).

¹O. Rayyan, V. Gonçalves and A. Tzes are with Center for Artificial Intelligence and Robotics (CAIR), New York University Abu Dhabi (NYUAD), 129188, Abu Dhabi, UAE.

²N. Evangeliou is with Electrical Engineering, NYUAD, UAE.

²Corresponding author's email: vmg6973@nyu.edu.

II. METHODOLOGY

A. Mathematical Notation

All vectors are considered column vectors. For a vector/matrix M , M^T represents its transpose. $0_{n \times m}$ and $I_{n \times n}$ represents the $n \times m$ zero matrix and n identity matrix, respectively. $\|\cdot\|$ represents the Euclidean norm of a vector and $\dot{\mathbf{v}}$ the time derivative of a vector $\mathbf{v}(t)$. For two vectors \mathbf{a} , \mathbf{b} , the symbol $\mathbf{a} \times \mathbf{b}$ represents their cross product.

B. Forward and Differential Kinematics for the RCM-Constraint

Let $\mathbf{x}_T, \mathbf{y}_T, \mathbf{z}_T$ and \mathbf{p}_T be 3D-vectors representing the x, y, z axis and center, respectively, of a frame \mathcal{F}_T attached to the tool-tip, written in a fixed, world frame \mathcal{F}_W . Under the assumption that a cylindrical tool is present, the point \mathbf{p}_T is at the tip of the tool, aligned with the cylinder's axis of rotation axis. The vector \mathbf{z}_T is also aligned with this axis. Subsequently, these vectors can be computed using forward kinematics as a function of the joint position configuration \mathbf{q} . Furthermore, let \mathbf{p}_F be the fixed position of the RCM point, also expressed as \mathcal{F}_W .

In Figure 1 an illustration of the RCM constraint in RAMIS is depicted. The grey objects represent the end-effector and tool of the robot, in three different configurations. The tool is inserted into the yellow volume through a fulcrum point \mathbf{p}_F . The tool-tip position \mathbf{p}_T can be moved as long as one of the points of the tool contains the static point \mathbf{p}_F . It can be derived that in order to satisfy the RCM point constraint, there must exist a scalar $\lambda > 0$ such that

$$\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F = \lambda \mathbf{z}_T(\mathbf{q}). \quad (1)$$

The scalar λ represents how much the tool is inserted into the RCM, and is positive if and only if the tool is inserted. Pre-multiplying this equation by \mathbf{x}_T^T and then by \mathbf{y}_T^T and using the orthogonality condition on $\mathbf{x}_T, \mathbf{y}_T, \mathbf{z}_T$, (2) holds.

$$\begin{aligned} \mathbf{x}_T(\mathbf{q})^T (\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F) &= 0, \\ \mathbf{y}_T(\mathbf{q})^T (\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F) &= 0. \end{aligned} \quad (2)$$

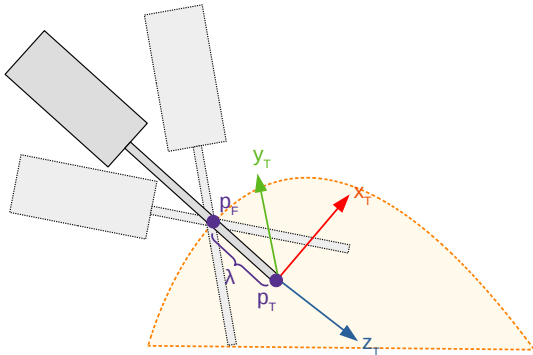


Fig. 1. Illustration of the RCM-constraint.

The benefit of using (2) over (1) is that the latter does not have the variable λ . Clearly, (2) can be written as $\mathbf{r}_F(\mathbf{q}) = 0_{2 \times 1}$, for a function $\mathbf{r}_F : \mathbb{R}^n \mapsto \mathbb{R}^2$. This is the *task function* for the RCM-constraint that will be used for controlling the robot. Note that the RCM-constraint requires a robot configuration with a minimum of two degrees of freedom, one for translational movements along the insertion axis and another for rotating the insertion axis alongside the fulcrum point. These two degrees of freedom align with the number of components of the proposed task function.

In order to implement the controller, the task Jacobian will be expressed as $J_F(\mathbf{q}) \triangleq \frac{\partial \mathbf{r}_F}{\partial \mathbf{q}}(\mathbf{q})$. The first row of this task Jacobian, relative to the first component of \mathbf{r}_F , will be derived, and the second one comes using a very similar reasoning. For this, let $J_v(\mathbf{q})$ and $J_\omega(\mathbf{q})$ be the linear and angular velocity Jacobians, respectively, for the tooltip frame \mathcal{F}_T written in \mathcal{F}_W . From standard differential kinematics [8] it is known that

- (i) $\frac{d}{dt} \mathbf{p}_T(\mathbf{q}) = J_v(\mathbf{q}) \dot{\mathbf{q}}$.
- (ii) Let $\boldsymbol{\omega}$ be the angular velocity of the tooltip frame. It is known that $\boldsymbol{\omega}(\mathbf{q}, \dot{\mathbf{q}}) = J_\omega(\mathbf{q}) \dot{\mathbf{q}}$. Furthermore, it is also known that $\frac{d}{dt} \mathbf{x}_T(\mathbf{q}) = \boldsymbol{\omega}(\mathbf{q}, \dot{\mathbf{q}}) \times \mathbf{x}_T(\mathbf{q})$. Consequently, $\frac{d}{dt} \mathbf{x}_T(\mathbf{q}) = (J_\omega(\mathbf{q}) \dot{\mathbf{q}}) \times \mathbf{x}_T(\mathbf{q})$.

Furthermore, let $r_{T,x}(\mathbf{q}) \triangleq \mathbf{x}_T(\mathbf{q})^T (\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F)$ be the first entry of \mathbf{r}_T . On one hand, by the chain rule

$$\frac{d}{dt} r_{T,x} = \frac{\partial r_{T,x}}{\partial \mathbf{q}} \dot{\mathbf{q}}. \quad (3)$$

On the other hand, by differentiating the expression of $r_{T,x}(\mathbf{q})$ using the results (i) and (ii), 4 can be derived

$$\frac{d}{dt} r_{T,x} = \left((J_\omega \dot{\mathbf{q}}) \times \mathbf{x}_T \right)^T (\mathbf{p}_T - \mathbf{p}_F) + \mathbf{x}_T^T J_v \dot{\mathbf{q}}, \quad (4)$$

in which the dependencies in $\mathbf{x}_T, \mathbf{y}_T, \mathbf{p}_T, J_v$ and J_ω on \mathbf{q} are omitted. Now, the mixed product formula is used: $(\mathbf{a} \times \mathbf{b})^T \mathbf{c} = (\mathbf{b} \times \mathbf{c})^T \mathbf{a}$. Applying this result into (4) and factoring out $\dot{\mathbf{q}}$, 5 can be formulated

$$\frac{d}{dt} r_{T,x} = \left(\mathbf{x}_T^T J_v + (\mathbf{x}_T \times (\mathbf{p}_T - \mathbf{p}_F))^T J_\omega \right) \dot{\mathbf{q}}. \quad (5)$$

Comparing (3) with (5), then $\frac{\partial r_{T,x}}{\partial \mathbf{q}} = \mathbf{x}_T^T J_v + (\mathbf{x}_T \times (\mathbf{p}_T - \mathbf{p}_F))^T J_\omega$. Similarly, the Jacobian for the component $r_{T,y}$ is obtained, by replacing \mathbf{x}_T with \mathbf{y}_T in the formula. Stacking these two Jacobians the desired Jacobian for $J_F(\mathbf{q})$ can be obtained. Equation 6 summarizes the above formulations.

$$\begin{aligned} \mathbf{r}_F &= \begin{bmatrix} \mathbf{x}_T^T (\mathbf{p}_T - \mathbf{p}_F) \\ \mathbf{y}_T^T (\mathbf{p}_T - \mathbf{p}_F) \end{bmatrix}. \\ J_F &= \begin{bmatrix} \mathbf{x}_T^T J_v + (\mathbf{x}_T \times (\mathbf{p}_T - \mathbf{p}_F))^T J_\omega \\ \mathbf{y}_T^T J_v + (\mathbf{y}_T \times (\mathbf{p}_T - \mathbf{p}_F))^T J_\omega \end{bmatrix}. \end{aligned} \quad (6)$$

C. Controller Design

The controller design is based on the task to track a tooltip trajectory $\mathbf{p}_{T,d}(t)$, while keeping the RCM error to 0, $\mathbf{r}_F = 0_{2 \times 1}$. For that, the *task function approach* [9] is incorporated. In this approach, it is assumed that the control objective can be codified with m differentiable functions

$\mathbf{r}_i : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^{k_i}$ of the configuration \mathbf{q} and time t as $\mathbf{r}_i(\mathbf{q}, t) = 0_{k_i \times 1}$ for $i = 1, 2, \dots, m$, i.e, the control objective is translated into zeroing all the functions \mathbf{r}_i . Then, a *task dynamics specification* is imposed for the purpose of choosing the control inputs so that $\frac{d}{dt}\mathbf{r}_i = -K_i\mathbf{r}_i$ holds along the system trajectories. This case implies that all of the $r_i \rightarrow 0$ exponentially. Therefore, in order to use this approach it is necessary to (i) choose the functions \mathbf{r}_i according to the task, (ii) select the gains K_i , that controls how fast the functions \mathbf{r}_i converge to 0 and how aggressive the controllers are and (iii) select the control inputs to meet the equation $\frac{d}{dt}\mathbf{r}_i = -K_i\mathbf{r}_i$.

For the step (i), for the first task, the task function $\mathbf{r}_T(\mathbf{q}, t) \triangleq \mathbf{p}_T(\mathbf{q}) - \mathbf{p}_{T,d}(t)$ will be used. For the second task, the function \mathbf{r}_F defined on the previous subsection will be used. The choice for step (ii) is discussed in Section III. Finally, in step (iii) the control input must be chosen. Using the chain rule and the assumption that $\dot{\mathbf{q}} = \mathbf{u}$, the task dynamics specifications $\frac{d}{dt}\mathbf{r}_i = -K_i\mathbf{r}_i$ become

$$\begin{aligned} J_v(\mathbf{q})\mathbf{u} &= -K_T\mathbf{r}_T(\mathbf{q}, t) + \dot{\mathbf{p}}_{T,d}(t) \\ J_F(\mathbf{q})\mathbf{u} &= -K_F\mathbf{r}_F(\mathbf{q}) \end{aligned} \quad (7)$$

in which the chain rule was used together with the fact that $\frac{\partial}{\partial \mathbf{q}}\mathbf{r}_T = J_v$ as well as that $\frac{\partial}{\partial t}\mathbf{r}_T = -\dot{\mathbf{p}}_{T,d}$.

In this work, since the motion $\mathbf{p}_{T,d}(t)$ will be relatively slow, this feedforward term can be omitted. So, once this term is removed, (7) becomes a system of linear equation for the variable $\dot{\mathbf{q}} = \mathbf{u}$. Considering both matricial equations for the two tasks $3 + 2 = 5$ equations are extracted, whereas for the Kuka LBR IIWA robot $n = 7$ joints are present. So, assuming that all the rows of the matrices J_v and J_F are linearly independent, there exist $7 - 5 = 2$ additional Degrees of Freedom (DoF) and there are infinite solutions for (7). However, it may be the case that some rows of J_v are linearly dependent, or very close to be linearly dependent to the rows of J_F . This means that there is a conflict between executing the task dynamic specifications for following the tooltip trajectory and for keeping the RCM-constraint.

In order to solve both issues (infinite solutions and conflicts), the problem of computing $\mathbf{u} = \dot{\mathbf{q}}$ will be formulated as the quadratic program

$$\begin{aligned} \mathbf{u}(\mathbf{q}, t) &= \arg \min_{\boldsymbol{\mu}} \|J_F(\mathbf{q})\boldsymbol{\mu} + K_F\mathbf{r}_F(\mathbf{q})\|^2 + \epsilon\|\boldsymbol{\mu}\|^2 \\ &\text{such that } J_v(\mathbf{q})\boldsymbol{\mu} = -K_T\mathbf{r}_T(\mathbf{q}, t) \end{aligned} \quad (8)$$

in which ϵ is a small positive constant (in this work, $\epsilon = 10^{-6}$). This formulation solves both the infinite solution problem and the conflict problem. Essentially, the program formulates the problem as follows: it treats the tooltip tracking task as a hard constraint, since the respective task dynamics specification is a constraint, and then, considering this, it tries to both satisfy the task dynamics specification for the RCM-constraint (trying to minimize $\|J_F(\mathbf{q})\boldsymbol{\mu} + K_F\mathbf{r}_F(\mathbf{q})\|^2$) and maintaining a low velocity (trying to minimize $\|\boldsymbol{\mu}\|^2$). However, since ϵ is small, it prioritizes more having a better control of the RCM than having a small norm. These characteristics guarantee that the

solution is unique (trying to minimize $\|\boldsymbol{\mu}\|^2$ guarantees strict convexity of the formulation and consequently uniqueness of solution) and that if there is a conflict between the tooltip tracking task and the RCM task, the former should have more priority. Furthermore, since (8) is a quadratic program without inequality constraints, it can be solved analytically by solving the system

$$\begin{bmatrix} J_F^\top J_F + \epsilon I_{n \times n} & J_v^\top \\ J_v & 0_{3 \times n} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} -K_F J_F^\top \mathbf{r}_F \\ -K_T \mathbf{r}_T \end{bmatrix} \quad (9)$$

in which $\boldsymbol{\gamma}$ is the Lagrange multiplier.

D. Comparison with Other Approaches

In this section, the presented approach is compared with others that also use the task function framework to formulate the problem. The focus will be on the description of the RCM task only. To make the comparison easier, their task functions are rewritten into equivalent forms.

In [10], the formulation of the RCM-constraint is based on the tangent plane equation at the insertion point, and a task function is derived based on this plane. The selected task function is equivalent to the task function $\mathbf{r}_{F,1}(\mathbf{q}) \triangleq \lambda \mathbf{z}_T(\mathbf{q}) - (\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F)$. Note that having this task function vanish is equivalent to having (1). However, it is assumed that λ is constant, which is not the case in the proposed approach.

In [7], the RCM-constraint task is defined to be $\mathbf{r}_{F,2}(\mathbf{q}) \triangleq (I - \mathbf{z}_T \mathbf{z}_T^\top)(\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F(\mathbf{q}))$. This task function incorporates an extra task component when compared to the proposed task function which includes 2 components, matching the DoF required by the RCM-constraint. This means the task function of [7] exhibits a redundant task component. The proposed task function \mathbf{r}_F is related to this task function $\mathbf{r}_{F,2}$ by the expression $\mathbf{r}_F = [\mathbf{x}_T^\top \mathbf{r}_{F,2} \ \mathbf{y}_T^\top \mathbf{r}_{F,2}]^\top$.

In [11], the RCM-constraint task is defined to be $\mathbf{r}_{F,3}(\mathbf{q}, \lambda) \triangleq \lambda \mathbf{z}_T(\mathbf{q}) - (\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F)$. This task function is equivalent to $\mathbf{r}_{F,1}$, except that λ is no longer constant. This requires λ to be continuously evolved according to the feedback in the closed loop. In addition to that, this approach requires the usage of a transformation matrix to partition the joint velocities, which is not necessary in the proposed approach.

E. The Effect of the Tool Insertion

From (1), pre-multiplying by \mathbf{z}_T to obtain that the *tool insertion* λ can be given as $\lambda(\mathbf{q}) = \mathbf{z}_T(\mathbf{q})^\top (\mathbf{p}_T(\mathbf{q}) - \mathbf{p}_F)$. Let L be the length of the tool. An important index is the ratio between how much is inserted into the RCM, λ and how much is outside, $L - \lambda$. This ratio, $\rho(\mathbf{q}) \triangleq |(L - \lambda(\mathbf{q})) / \lambda(\mathbf{q})|$ is important. Essentially, the higher is this number, the more difficult it is to control the robot because a small movement in the tool-tip may require a large movement for the robot as whole.

This is better visualized by considering how a small change in the tooltip position $\|d\mathbf{p}_T\|$ is related to a small change in a point \mathbf{p}_E that connects the tool to the robot, that is, $\|d\mathbf{p}_E\|$, as shown in Figure 2. Since the interest is

on studying how the insertion affects the movement, it is assumed that in this movement the insertion does not change, so $d\lambda = 0$. From (1), one can take the differential from both sides, use the fact that $d\mathbf{p}_F = 0$ (since the RCM is constant) and $d\lambda = 0$ (it is assumed that the insertion does not change) to obtain:

$$d\mathbf{p}_T = \lambda d\mathbf{z}_T \quad (10)$$

and thus $\|d\mathbf{p}_T\| = |\lambda| \|d\mathbf{z}_T\|$. Furthermore, since it is possible to write $\mathbf{p}_T = \mathbf{p}_E + L\mathbf{z}_T$, taking the differential from both sides

$$d\mathbf{p}_T = d\mathbf{p}_E + Ld\mathbf{z}_T \quad (11)$$

and thus from (10) and (11) the result $d\mathbf{p}_E = (\lambda - L)d\mathbf{z}_T$ is obtained and, consequently, $\|d\mathbf{p}_E\| = |\lambda - L| \|d\mathbf{z}_T\|$. Therefore, $\|d\mathbf{p}_E\|/\|d\mathbf{p}_T\| = |\lambda - L|/|\lambda| = \rho$. This result means that ρ measures, in a given configuration, how much a (small) movement in the tooltip generates of movement in a specific point of the robot (the one that connects the robot to the tool) or, equivalently, how the respective velocities relate. So, for example, $\rho = 5$ means that having $\|\dot{\mathbf{p}}_T\| = 1\text{cm/s}$ on the tooltip will require, roughly, that \mathbf{p}_E moves with velocity $\|\dot{\mathbf{p}}_E\| = 5\text{cm/s}$, i.e., small velocities on the tooltip generate large velocities on the robot which, then, requires large velocities on the joints. Indeed, as it will be shown in the experimental results, generally when ρ increases the RCM error \mathbf{r}_F also increases. This implies that, as a general rule, the tool should be designed so $\rho(\mathbf{q})$ is as small as possible.

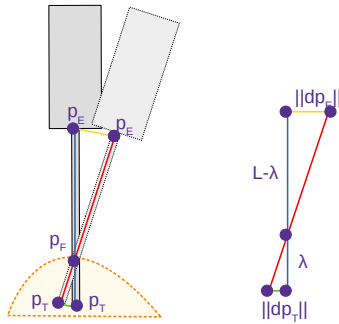


Fig. 2. Geometric illustration of the fact that $\|d\mathbf{p}_E\|/\|d\mathbf{p}_T\| = (L - \lambda)/\lambda = \rho$. If $\|d\mathbf{p}_E\|$ and $\|d\mathbf{p}_T\|$ are small, the two triangles formed are very close to be right triangles, similar to each other, and thus, $\|d\mathbf{p}_E\|/(L - \lambda) = \|d\mathbf{p}_T\|/\lambda$.

F. Selecting the Starting Configuration

A crucial parameter in the surgical room is the positioning of the robotic equipment with respect to the patient's body and the surrounding physicians. For every operation there always exists an optimal initial joint space configuration for a robotic arm that allows for optimal workspace coverage and dexterity.

This issue can be quantified using the *manipulability* index. Suppose that, at a configuration \mathbf{q}_0 and with a RCM

\mathbf{p}_F , it is desired to execute a linear velocity \mathbf{v} on the tooltip. It is assumed that the RCM error is zero at \mathbf{q}_0 . So, the joint velocity $\dot{\mathbf{q}}$ can be decided by solving the system of equations $J_F(\mathbf{q}_0, \mathbf{p}_F)\dot{\mathbf{q}} = \mathbf{0}_{2 \times 1}$ (keep the RCM error in 0) and $J_v(\mathbf{q}_0)\dot{\mathbf{q}} = \mathbf{v}$ (move according to this linear velocity). Note that it is written $J_F(\mathbf{q}_0, \mathbf{p}_F)$ to stress that the RCM Jacobian also depends on the RCM \mathbf{p}_F . Let $J(\mathbf{q}_0, \mathbf{p}_F) = J \triangleq [J_v^T J_F^T]^T$. Then, it is not difficult to show that the smallest solution $\dot{\mathbf{q}}$ (in the Euclidean norm) to these two equations has Euclidean norm $\|\dot{\mathbf{q}}\| = \sqrt{\mathbf{v}^T M \mathbf{v}}$, in which $M(\mathbf{q}_0, \mathbf{p}_F)$ is the top 3 rows and columns of the matrix $(JJ^T)^{-1}$. So, if the eigenvalues of the matrix $M(\mathbf{q}_0, \mathbf{p}_F)$ are large, this means that in order to execute a linear velocity at that configuration \mathbf{q}_0 and RCM \mathbf{p}_F , large joint velocities are required. So, it is reasonable to choose a configuration \mathbf{q}_0 and RCM \mathbf{p}_F to operate around so M has small eigenvalues.

It is necessary, therefore, to optimize both on \mathbf{q}_0 and \mathbf{p}_F . In order to simplify the problem, the RCM choice will be attached to the choice of \mathbf{q}_0 as follows: $\mathbf{p}_F = \mathbf{p}_T(\mathbf{q}_0) - \lambda_0 \mathbf{z}_T(\mathbf{q}_0)$, in which $\lambda_0 = 0.1m$ is fixed. This means that the RCM is selected as being 10cm above the tooltip along the tool at the starting configuration. With this, one can optimize only on \mathbf{q}_0 , by minimizing the maximum eigenvalue (spectral radius) of M while considering other constraints as the tooltip position being inside a region in the workspace, the configuration being within the joint limits and also that the configuration should make the tool be pointing downwards.

III. EXPERIMENTAL STUDIES

A. Experimental Setup

To verify the accuracy of the proposed controller, it was tested on a 7-DOF KUKA LBR iiwa-14 R820 robotic manipulator, shown in Figure 3 using KUKA's Fast Research Interface (FRI) [12] which offers joint-positions control. Considering the imposed constraint in FRI of manipulating the serial robotic arm through joint-positions, and given that the proposed controller operates on the joint-velocity level, a virtual joint-velocity control method was employed. After determining the joint-velocities $\dot{\mathbf{q}}_n$, numerical integration is performed using the preceding joint-positions \mathbf{q}_{n-1} to compute the subsequent joint-positions \mathbf{q}_n , applying the formula $\mathbf{q}_n = \mathbf{q}_{n-1} + \frac{\dot{\mathbf{q}}_n}{\text{frequency}}$. This allows the controller to be adapted to work with the KUKA's FRI joint-position control interface at a frequency of 250 Hz.

In order to mimic the surgical instruments used in RA-MIS procedures, a tool was attached to the end-effector of the manipulator, having a length of $L = 0.4m$. The RCM constraint was then established at a distance of $\lambda_0 = 0.1m$ from the tool-tip. As a result, an initial insertion ratio of $\rho(\mathbf{q}_0) = \frac{0.3m}{0.1m} = 3$ was established. To generate an initial configuration that maximizes manipulability, the annealing algorithm, described in Subsection II-F, was used to get $\mathbf{q}_0 = [35.5^\circ, 81.9^\circ, -92.2^\circ, -92.0^\circ, 82.1^\circ, 91.2^\circ, -72.0^\circ]^T$, which corresponds to a starting position $\mathbf{p}_{T,0} = [0.562m, -0.095m, -0.126m]$. This starting configuration can be seen in Figure 3. The fulcrum point \mathbf{p}_F is then

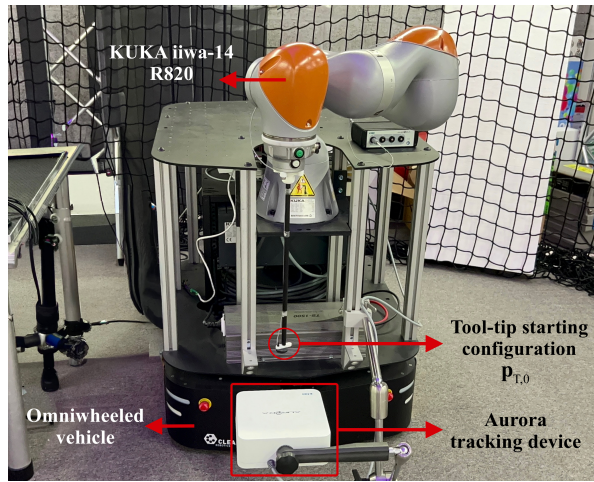


Fig. 3. Initial Configuration of the Manipulator ($\mathbf{p}_{T,0}$) in the setup environment

established to be at a distance $\lambda = \frac{L}{1+\rho(q_0)}$ from tool-tip of this starting configuration. The robot was then instructed to trace the helical trajectory described by the parametric equation:

$$\mathbf{p}_{T,d}(t) = \mathbf{p}_{T,0} + \begin{bmatrix} 0.03\alpha(t) \cos(\frac{\pi}{5}t) \\ 0.03 \sin(\frac{\pi}{5}t) \\ 0.06 \sin(\frac{\pi}{10}t) - 0.04\alpha(t) \end{bmatrix} \quad (12)$$

in which $\alpha(t) = \max(1, t/5)$, t is in seconds and $\mathbf{p}_{T,d}(t)$ in meters. The function $\alpha(t)$ was incorporated in the helical path parametric equation to ensure a smooth transition from the initial tooltip position to the initial desired tooltip position (so $\mathbf{p}_{T,d}(0) = \mathbf{p}_{T,0}$). The helical path was chosen as it serves as an ideal approximation of suturing motion - a very common task in surgical procedures [13]. The parameters K_T and K_F in (7) were set to $K_T = 14s^{-1}$ and $K_F = 27s^{-1}$.

B. Results and Accuracy Evaluation

To evaluate the accuracy of the tool-tip tracking along the helical path in (12), two distinct measurement methods were used to continuously determine the tool-tip's position. The primary method involves using the manipulator's joint position feedback for calculating the tip position via the forward kinematics formulation. The secondary method involves the use of an external tracking device to assess the accuracy of the forward kinematics estimation of the first method [14], [15]. Specifically, a version of the Aurora electromagnetic tracking system from NDI with RMS positioning accuracy of 0.8 mm and measurement frequency of 40Hz was used. The tracking of the tool-tip, evaluated using the two layers of accuracy measurements, is visualized in Figure 4. Additionally, Figure 5 illustrates the positioning error during helical path execution, calculated as $\|\mathbf{r}_T(\mathbf{q}(t), t)\|$. On average, when calculated using the forward-kinematics model estimation, the positioning error relative to the desired trajectory (12) amounts to 0.78 mm. However, when evaluated using the

Aurora device measurements, the positioning error averages 1.1 mm along the path.

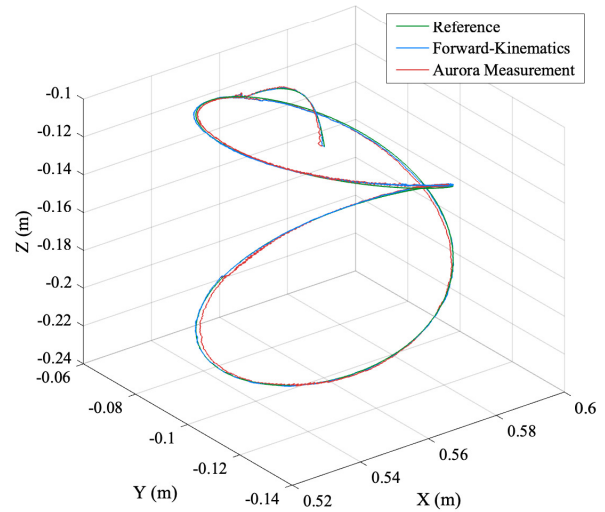


Fig. 4. Tool-Tip Position Visualization using the forward-kinematics model and the Aurora device measurement

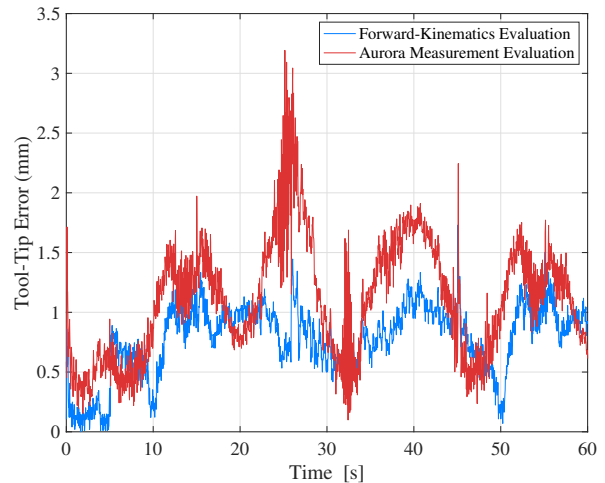


Fig. 5. Tool-Tip Position Error evaluated using the forward-kinematics model and the Aurora device measurement

Figure 6 shows the RCM error throughout the tracked trajectory, calculated as $\|\mathbf{r}_F(\mathbf{q}(t))\|$ and equal to the Euclidean distance between the tool and the fulcrum point \mathbf{p}_F . With an initial insertion ratio $\rho(q_0) = 3$, the RCM following had an average error of 1.5 mm. To verify the proposed hypothesis in Subsection II-E, the initial RCM position was later modified to become a distance of $\lambda_0 = 0.2m$ from the tool-tip. As a result, the initial insertion ratio became $\rho(q_0) = \frac{0.2m}{0.2m} = 1$. The error of the manipulator's RCM from the fulcrum point at this insertion ratio is depicted in Figure 6. On average, the RCM error at $\rho(q_0) = 1$ was 0.4 mm, a 73.3% decrease from that at $\rho(q_0) = 3$.

A visualization of random achieved manipulator configurations during RCM validation is depicted in Figure 7. Finally, a video of the experiment can

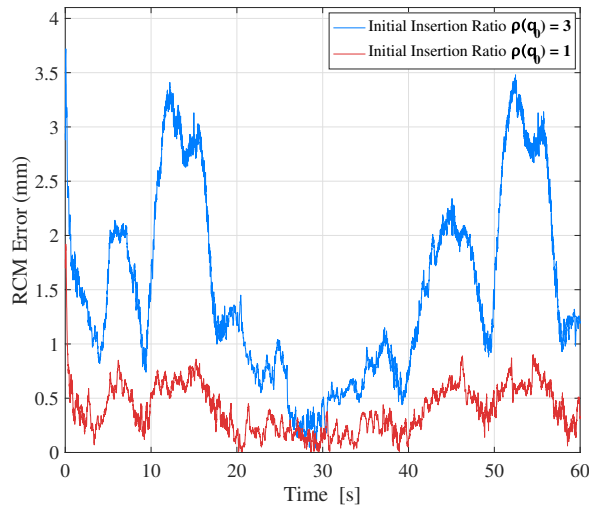


Fig. 6. RCM position error $\|r_F(\mathbf{q}(t))\|$ at an initial insertion ratio of $\rho(\mathbf{q}_0) = 3$ and $\rho(\mathbf{q}_0) = 1$ evaluated using the forward-kinematics model

be seen in <https://youtu.be/Ybg34tYok9U> and the relevant open-sourced implementation is located at <https://github.com/RISC-NYUAD/RCMController>.

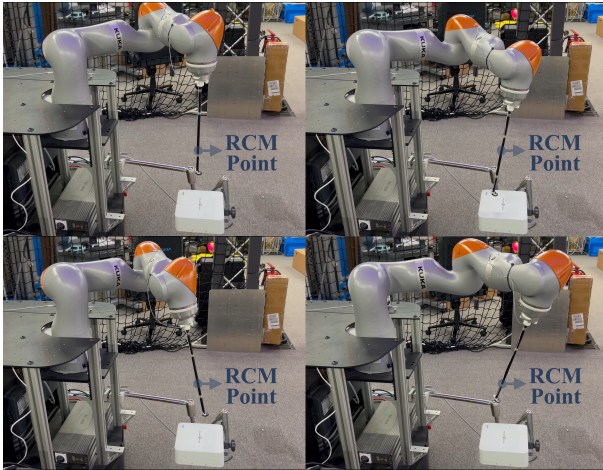


Fig. 7. RCM Visualization

IV. CONCLUSIONS

In this paper, a kinematic control approach for RCM-constrained tool-tip trajectory tracking is developed. The approach is based on a novel task function formulation, which includes a minimal representation of the RCM constraint. Furthermore, the impact of the tool's insertion ratio ρ into the system, as well as, the selection of the starting configuration are discussed. Experimental results showcase the applicability of the approach mainly, but not limited to, RA-MIS operations.

ACKNOWLEDGEMENTS

This work was partially performed in the Kinesis Lab, Core Technology Platform (CTP) facility of NYUAD.

This work was partially supported by the NYUAD Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010.

REFERENCES

- [1] K. Fuchs, "Minimally invasive surgery," *Endoscopy*, vol. 34, no. 02, pp. 154–159, 2002.
- [2] 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, 2013.
- [3] 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, pp. 1281–1296, 2015.
- [4] N. Evangeliou and A. Tzes, "Development of an sma-actuated redundant robotic platform for minimally invasive surgery," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechanics (BioRob)*, pp. 353–358, 2016.
- [5] K. Xu, J. Zhao, and M. Fu, "Development of the sjtu unfoldable robotic system (surs) for single port laparoscopy," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2133–2145, 2015.
- [6] C.-H. Kuo and J. S. Dai, "Robotics for minimally invasive surgery: a historical review from the perspective of kinematics," in *International Symposium on History of Machines and Mechanisms: Proceedings of HMM 2008*, pp. 337–354, Springer, 2009.
- [7] J. Colan, A. Davila, K. Fozilov, and Y. Hasegawa, "A concurrent framework for constrained inverse kinematics of minimally invasive surgical robots," *Sensors*, vol. 23, no. 6, 2023.
- [8] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, Springer London, 2010.
- [9] C. Samson, M. Borgne, and B. Espiau, *Robot Control: The Task Function Approach*. Oxford engineering science series, Clarendon Press, 1991.
- [10] H. Azimian, R. V. Patel, and M. D. Naish, "On constrained manipulation in robotics-assisted minimally invasive surgery," in *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics*, pp. 650–655, 2010.
- [11] H. Sadeghian, F. Zokaie, and S. Hadian Jazi, "Constrained kinematic control in minimally invasive robotic surgery subject to remote center of motion constraint," *Journal of Intelligent & Robotic Systems*, vol. 95, 09 2019.
- [12] K. Chatzilygeroudis, M. Mayr, B. Fichera, and A. Billard, "iiwa_ros: A ROS Stack for KUKA's IIWA robots using the Fast Research Interface," 2019.
- [13] A. Karadimos, A. Tzes, N. Evangeliou, and E. Dermatas, "Virtual reality simulation of a robotic laparoscopic surgical system," in *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, pp. 1573–1579, 2022.
- [14] J. Hummel, M. Figl, C. Kollmann, H. Bergmann, and W. Birkfellner, "Evaluation of a miniature electromagnetic position tracker," *Medical physics*, vol. 29, no. 10, pp. 2205–2212, 2002.
- [15] J. Hummel, M. Figl, W. Birkfellner, M. R. Bax, R. Shahidi, C. Maurer, and H. Bergmann, "Evaluation of a new electromagnetic tracking system using a standardized assessment protocol," *Physics in Medicine & Biology*, vol. 51, no. 10, p. N205, 2006.