2014 5th IEEE RAS & EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
August 12-15, 2014. São Paulo, Brazil

A programmable remote center-of-motion controller for minimally invasive surgery using the dual quaternion framework

Murilo M. Marinho, Mariana C. Bernardes, Antônio P. L. Bó

Abstract— **Within robotic-aided laparoscopy, one important research topic concerns the teleoperation of open chain serial link manipulators. As opposed to specialized surgical robots, a serial robot might be used for different procedures with small changes on the setup, lowering the involved costs and possibly increasing the acceptance of robotic systems in clinical settings. However, such robots do not present kinematic constraints that guarantee that the tool movements are projected on the pivoting point defined by the incision (RCM). In such scenario, the RCM must be assured by software. This paper presents a novel control strategy for laparoscopic tools attached to robotic manipulators that makes use of a programmable RCM. The tool movement references are generated intuitively by the surgeon, while the RCM is maintained by software using a dual quaternion-based kinematic controller. The method is evaluated in a simulated surgical environment and presented satisfactory results, both in terms of RCM control, tool positioning, and good performance under human operation.**

I. Introduction

Minimally Invasive Surgery (MIS), is an operation technique based on the access via several small incisions in the patient's body, being consequently less invasive than open surgery used for the same purpose. MIS interventions are significantly becoming the preferred approach since they offer outstanding advantages like less pain, smaller scars and faster recovery time [1].

One of the main difficulties in MIS is that the endoscope is manipulated by an assistant instead of by the main surgeon himself, therefore the main surgeon's hand-eye coordination is disturbed. Also, the incision acts as a pivoting point causing the instruments movements to be reduced from 6 to 4 degrees of freedom (DOF). Consequently, the surgeon hand movements about the incision are mirrored and scaled relative to the instrument tip. In addition, tactile information is lost for tissue manipulation since there is no direct contact between the tissue and the surgeon's hands. As a result, MIS procedures have a long learning curve for the physician and hinder the use of MIS techniques in complex surgical tasks because of longer procedure times, more difficult manipulation of instruments and torturous ergonomics [2].

A. Related Work

The goal of surgical robotics is not to replace surgeons with robots, but to provide versatile tools that may extended their ability to treat patients [3]. In this light, many robotic devices have been proposed to overcome some of the mentioned disadvantages and allow the development of more complex MIS procedures. In the laparoscopy scenario, surgical robots have to generate a Remote Center of Motion (RCM) that should coincide with the pivoting point on the patient. Such robots are usually divided in two categories: specialized and general purpose. Specialized robots such as the most known Da Vinci and many others [4], generate the RCM mechanically for the safety it provides. However, they have the disadvantage of being high cost and have restricted use to the specific application they have been developed for. On the other hand, cheaper general purpose robots, such as serial link manipulators, can also be used in laparoscopy if the RCM is properly generated by software. As this category evolves to share the same level of safety as the other, it may provide not only pivot flexibility but increased versatility. Hence, software RCM control has been the subject many recent works [5]–[7]. In this paper, we also focus on this category of RCM generation.

One of the interesting characteristics of the procedures in MIS is that they only cover small distances and must be performed slowly. Combining this fact with the use of serial link manipulators, we are able to bring the advances of kinematic control to the robotic MIS development. Recent works synthesize an interesting update to kinematic control in recent years: the use of the unit dual quaternion framework to represent rigid motions without the singularity issues of other minimal representations [8], [9]. Among its advantages, this representation allows a straightforward way to obtain geometric parameters, resulting in a more intuitive derivation controllers and path planners.

Despite the existence of many works focusing on the use of serial link manipulators in robotic laparoscopy, many of the controllers are case dependent and cannot be easily extended to the general manipulator [10]–[12]. Using task-priority resolutions and an elegant Jacobian devised specifically to the RCM generation, Azimian *et al* [7] created a controller for the general case.

Although being able to maintain the RCM error negligible, their controller showed considerable tool-tip deviation from the desired path. This is a severe issue when considering MIS applications, as it means the tool may sometimes move in an arbitrary direction inside the patient's body. Another issue is that the rotation of the tool around the axis of the trocar is not specified as a reference, being arbitrary chosen by matrix inversions in the controller.

B. Proposed work

In this work we propose a kinematic controller in the unit dual quaternion space for controlling an endoscope in

The authors are with LARA - Automation and Robotics Laboratory, Universidade de Brasília, Caixa Postal 4386, 70919-970 Brasília-DF, Brazil {murilomarinho,bernardes,antonio.plb}@lara.unb.br

laparoscopic procedures. The choice of the dual quaternion framework justifies by the fact that it is a compact mathematical tool that presents many advantages in the point of view of feedback control. The definition of both position and orientation in a common vector simplifies the design of the controller and avoids the computation of complex transformations to obtain the orientation error. Since this formulation relies on the use of quaternions, it is also non singular, unlike other representations based on sequence of angle rotations (e.g., Euler angles, Roll-Pitch-Yaw).

The proposed controller receives simple references from a haptic device and generates the RCM with negligible error, while evading singularities using task-priority. In addition, our method is not model dependent, being applicable to any general case manipulator. To the best of the authors' knowledge, this is the first controller to encircle all those characteristics.

Finally, a last issue commonly neglected by other works is the generation of references for the laparoscopy controllers. This subject is usually left unnoticed, but surgeons usually have strong expectations about system easiness and performance, while presenting low tolerance for interfaces that impede their work. Hence, we propose a controller that is able to receive simple movement references from the surgeon such as up-down, right-left, in-out commands, and transparently transform them in tool tip displacements, while dealing with the RCM restriction and mirrored movements internally.

II. Mathematical Background

We begin by recalling dual quaternions and their basic algebra when representing rigid transformations to establish the notation used in this work, followed by a brief review of existing work in the kinematic control of robotic manipulators.

A. Dual Quaternions

The dual quaternions are the basic building blocks of the kinematic control theory implemented in this work. We begin by defining $\hat{\imath}$, $\hat{\jmath}$ and $\hat{k}$ as the three imaginary components of a quaternion such that $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = -1$ and $\hat{\imath}\hat{\jmath}\hat{k} = -1$.

Hence, a general quaternion $\mathbf{x}$ is given by

$$\mathbf{x} = q_1 + q_2\hat{\imath} + q_3\hat{\jmath} + q_4\hat{k},$$

and its conjugate is defined as $\mathbf{x}^* \triangleq q_1 - q_2\hat{\imath} - q_3\hat{\jmath} - q_4\hat{k}$.

The norm of a quaternion $\mathbf{x}$ is $\| \mathbf{x} \| = \sqrt{\mathbf{x}\mathbf{x}^*}$. An arbitrary rotation of a rigid body by an angle θ around an axis $\mathbf{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$ is represented by the unit norm quaternion

$$\mathbf{r} = \cos\left(\frac{\theta}{2}\right) + \mathbf{n}\sin\left(\frac{\theta}{2}\right).$$

A translation described by $\mathbf{t} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k}$ can be associated to a rotation $\mathbf{r}$ in order to represent the complete rigid motion. This is represented by the unit dual quaternion

$$\underline{\mathbf{x}} = \mathbf{r} + \varepsilon\frac{1}{2}\mathbf{t}\mathbf{r},$$

and its conjugate is defined as $\underline{\mathbf{x}}^* \triangleq \mathbf{r}^* + \varepsilon(\frac{1}{2}\mathbf{t}\mathbf{r})^*$; where ε is nilpotent; i.e., $\varepsilon \neq 0$ but $\varepsilon^2 = 0$. The translator operator obtains the translation quaternion from the dual quaternion, that is translation$(\underline{\mathbf{x}}) = \mathbf{t}$.

The P operator returns a quaternion that represents the pure part of a dual quaternion $\underline{\mathbf{x}}$, that is $P(\underline{\mathbf{x}}) \triangleq \mathbf{r}$. On the other hand, the D operator returns a quaternion that stands for the dual part of a dual quaternion $\underline{\mathbf{x}}$, meaning $D(\underline{\mathbf{x}}) \triangleq \frac{1}{2}\mathbf{t}\mathbf{r}$. The logarithm of $\underline{\mathbf{x}}$ is $\log \underline{\mathbf{x}} \triangleq \frac{\theta\mathbf{n}}{2} + \epsilon\frac{\mathbf{t}}{2}$, while the exponential of $\underline{\mathbf{x}}$ is

$$\exp \underline{\mathbf{x}} \triangleq P(\exp \underline{\mathbf{x}}) + \epsilon D(\underline{\mathbf{x}})P(\exp \underline{\mathbf{x}})$$

$$P(\exp \underline{\mathbf{x}}) \triangleq \cos \| P(\underline{\mathbf{x}}) \| + \frac{\sin \| P(\underline{\mathbf{x}}) \|}{\| P(\underline{\mathbf{x}}) \|} P(\underline{\mathbf{x}}),$$

if $\| P(\underline{\mathbf{x}}) \| \neq 0$; otherwise $P(\exp \underline{\mathbf{x}}) \triangleq 1$.

The vec operator maps a given dual quaternion $\underline{\mathbf{x}}$ into an eight-dimensional column vector; i.e.,

$$\mathrm{vec}(\underline{\mathbf{x}}) \triangleq \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 \end{bmatrix}^T.$$

B. Kinematic Control

The forward kinematics model of a manipulator robot can be easily obtained from the Denavit-Hartenberg parameters directly in dual quaternion space [8]. However, a closed-form solution for the inverse kinematics does not exist for the general case. In order to overcome this limitation, we use the mapping between the manipulator joint velocities $\dot{\theta} \in \mathbb{R}^{n\times 1}$ and the generalized end-effector velocity $\mathrm{vec}\,\dot{\underline{\mathbf{x}}}_{\text{eff}} \in \mathbb{R}^{8\times 1}$ given by

$$\mathrm{vec}\,\dot{\underline{\mathbf{x}}}_{\text{eff}} = \mathbf{J}\dot{\theta}, \tag{1}$$

where $\mathbf{J} \in \mathbb{R}^{8\times n}$ is called the manipulator analytical Jacobian [13] and depends on the current robot posture. As (1) is a simple linear mapping, it is common practice to invert the Jacobian and use it in closed loop controllers that exponentially reduce the error between the current pose $\underline{\mathbf{x}}_{\text{eff}}$ and the desired pose $\underline{\mathbf{x}}_d$

$$\dot{\theta} = \mathbf{K}\mathbf{J}^{inv}\,\mathrm{vec}(\underline{\mathbf{x}}_{\text{eff}} - \underline{\mathbf{x}}_d), \tag{2}$$

with a positive definite $\mathbf{K}$ that affects the rate of convergence. Some research has been done in devising controllers directly in dual quaternion space starting from (2), branching to cooperative manipulation frameworks [8] and a H_∞ control design [9]. One very interesting result of [9] is the introduction of a transformation invariant error metrics. Modifying (2) to use such error metrics results in

$$\dot{\theta} = \mathbf{K}\mathbf{N}^{inv}\,\mathrm{vec}(1 - \underline{\mathbf{x}}^*_{\text{eff}}\underline{\mathbf{x}}_d), \tag{3}$$

in which $\mathbf{N} = \bar{\mathbf{H}}(\underline{\mathbf{x}}_d)\mathbf{C}_8\mathbf{J}$, $\bar{\mathbf{H}}(\underline{\mathbf{x}}_d)$ is the Hamilton operator [14] of $\underline{\mathbf{x}}_d$, and $\mathbf{C}_8 = diag(1,-1,-1,-1,1,-1,-1,-1)$. Another interesting fact is that regular kinematic control strategies can be applied on (3) with minor modifications, in order to solve the known problems concerning task-space singularities and redundancy.

In the neighborhood of a task-space singularity, $\mathbf{J}$ and $\mathbf{N}$ become ill conditioned, which results in high values for the joint velocities when the Moore-Penrose pseudoinverse is

directly applied. For this reason, earlier works summarized in [15] suggest the use of damped pseudoinverses, allowing a trade-off between end effector pose tracking error and maximum joint velocities norm near singular configurations.

In medical procedures, however, neither tracking error nor high joint velocities are acceptable. In this light, we use the Moore-Penrose pseudoinverse of the Jacobian–denoted by $\mathbf{J}^\dagger$ or $\mathbf{N}^\dagger$–and use the manipulator redundancy to avoid singularities and their effects.

Manipulator redundancy is the existence of more degrees of freedom in the manipulator than the necessary to perform a given task, which allows the controller to use those extra degrees of freedom in the execution of secondary tasks. This is usually performed by projecting a secondary objective function of the joint variables $c(\theta)$ in the nullspace of the primary task [16]. By doing so, (3) becomes

$$\dot{\theta} = \mathbf{K}\mathbf{N}^\dagger \operatorname{vec}(1 - \underline{\mathbf{x}}^*_{\text{eff}}\underline{\mathbf{x}}_d) + \mathbf{P}\mathbf{K}_c\mathbf{J}_c(c(\theta) - c_d), \quad (4)$$

where $\mathbf{P} = (\mathbf{I} - \mathbf{N}\mathbf{N}^\dagger)$ is a nullspace projector in which $\mathbf{I}$ is the identity matrix, $\mathbf{K}_c$ is a positive definite gain matrix for the secondary objective, and $\mathbf{J}_c = \frac{\partial c(\theta)}{\partial \theta}$ is the constraint Jacobian. Also, c_d is the desired value for the secondary objective function. By using (4), we will have the convergence $c(\theta) = c_d$ as long as the secondary objective does not conflict with the first [15].

Our interest is to use a suitable $c(\theta)$ so that (4) evades singularities. For that purpose, there are some functions described in the literature [17], [18]. It is not important which technique is chosen as long as it is well tuned to evade kinematic singularities.

III. Remote Center of Motion Camera Controller

In laparoscopic procedures, the surgical instrument must be inserted through the trocar into the patients body. The trocar constraints the movement of the surgical instrument to 4 degrees of freedom, as shown in Fig. 1.

At the beginning of the procedure, the endoscope is inserted manually into the patient. After manual setup, the point on the endoscope coincident with the same translation as the trocar will be considered the end effector with initial pose $\underline{\mathbf{x}}_d(0)$ as shown in Figure 1. Its z-axis given by the endoscope shaft. This choice of end effector will allow an easy derivation of the controller.

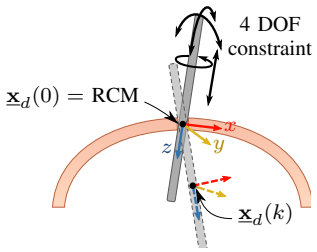


Fig. 1: The RCM constraint and initial positioning.

A. Reference generation

In order to move an endoscope camera intuitively, all motion should be performed with respect to its image point-of-view. For this purpose, at each instant k we acquire a left-right rotation $\mathbf{r}_\alpha(k)$, an up-down rotation $\mathbf{r}_\beta(k)$, an "around-itself" rotation $\mathbf{r}_\gamma(k)$, and an in-out translation along the endoscope axis $\mathbf{t}_z(k)$; all of them with respect to the initial pose of the end effector $\underline{\mathbf{x}}_d(0)$ which is the same as the RCM. Then, we define the end effector desired pose

$$\underline{\mathbf{x}}_d(k) = \underline{\mathbf{x}}_d(0)\mathbf{r}_\alpha(k)\mathbf{r}_\beta(k)\mathbf{r}_\gamma(k)(1 + \epsilon\frac{1}{2}\mathbf{t}_z(k)). \quad (5)$$

By using (5), it is easy to see that all $\underline{\mathbf{x}}_d(k)$ respect the RCM constraint while moving the camera according to the control references.

B. Kinematic controller

As neither RCM generation errors nor endoscope positioning errors are acceptable, both objectives should have the same priority in the kinematic control loop. In order to prevent unacceptable large velocities near singularities, the secondary task should evade them.

For this purpose, we use the references (5) to assure the RCM point and reposition the endoscope with the same priority. Those references are sent to the discrete time implementation of (4)

$$\begin{aligned}\theta(k+1) = \theta(k) &+ \mathbf{K}\mathbf{N}^\dagger \operatorname{vec}(1 - \underline{\mathbf{x}}^*_{\text{eff}}(k)\underline{\mathbf{x}}_d(k)) \\ &+ \mathbf{P}\mathbf{K}_c\mathbf{J}_c(c(\theta) - c_d),\end{aligned} \quad (6)$$

with a $c(\theta)$ for singularity evasion, such as [17], [18].

C. Unit dual quaternion path planner

Between two sequential references $\underline{\mathbf{x}}_d(l)$ and $\underline{\mathbf{x}}_d(l+1)$, there is no guarantee that any intermediary pose of the endoscope will keep the RCM by only using (5) as input for (6). This becomes more evident when $\underline{\mathbf{x}}_d(l)$ is far from $\underline{\mathbf{x}}_d(l+1)$. The solution is to interpolate $\underline{\mathbf{x}}_d(l)$ and $\underline{\mathbf{x}}_d(l+1)$ with $N \in \mathbb{N} - \{0\}$ points that also generate the RCM.

Consider then that the interpolation is made in an instance l, in which we store $\underline{\mathbf{x}}(l)$, the end effector pose at instant l, and $\underline{\mathbf{x}}_d(l)$, the desired end effector pose at instant l. They will be, respectively, the beginning and ending poses in our interpolation. Then, we need to obtain the constrained description of $\underline{\mathbf{x}}(l)$ as if it was given by (5). That is, the rotation $\mathbf{r}^l_{\text{rcm}}$ and the constrained translation $\mathbf{t}^l_{\text{rcm}}$ so that $\underline{\mathbf{x}}_{\text{rcm}}\mathbf{r}^l_{\text{rcm}}\underline{\mathbf{t}}^l_{\text{rcm}}$ equals $\underline{\mathbf{x}}(l)$. For that purpose, we note that

$$\underline{\mathbf{x}}(l) = \underline{\mathbf{x}}_{\text{rcm}}\underline{\mathbf{x}}^l_{\text{rcm}}, \quad (7)$$

for some $\underline{\mathbf{x}}^l_{\text{rcm}}$ that describes the motion from $\underline{\mathbf{x}}_{\text{rcm}}$ to $\underline{\mathbf{x}}(l)$. From (7), we can obtain $\mathbf{r}^l_{\text{rcm}}$ by noticing

$$\begin{aligned}\underline{\mathbf{x}}^l_{\text{rcm}} &= \underline{\mathbf{x}}^*_{\text{rcm}}\underline{\mathbf{x}}(l) \\ \therefore \mathbf{r}^l_{\text{rcm}} &= \mathcal{P}(\underline{\mathbf{x}}^*_{\text{rcm}}\underline{\mathbf{x}}(l)).\end{aligned} \quad (8)$$

With (8), we can obtain the translation from the RCM to $\underline{\mathbf{x}}(l)$ as given by

$$\mathbf{t}^l_{\text{rcm}} = \text{translation}((\underline{\mathbf{x}}^l_{\text{rcm}}\mathbf{r}^l_{\text{rcm}})^*\underline{\mathbf{x}}(l)).$$

Due to the RCM constraint, $\mathbf{t}^l_{\text{rcm}}$ can only be a translation in the z-axis. However, numerical and kinematic inaccuracies may cause it to be some

$$\mathbf{t}^l_{\text{rcm}} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k},$$

with $t_x \neq 0$ and $t_y \neq 0$. To correctly interpolate points between two poses that maintain the RCM constraint, we need the starting and ending poses to also keep the constraint. So, instead of using $\underline{\mathbf{x}}(l)$ as-is, we force $t_x = t_y = 0$ to obtain

$$\mathbf{t}^l_{\text{rcm}} = t_z\hat{k} \implies \underline{\mathbf{t}}^l_{\text{rcm}} = (1 + \frac{1}{2}\epsilon t_z\hat{k}).$$

We then define $\underline{\mathbf{x}}'(l) = \underline{\mathbf{x}}_{\text{rcm}}\mathbf{r}^l_{\text{rcm}}\underline{\mathbf{t}}^l_{\text{rcm}}$, which it is $\underline{\mathbf{x}}(l)$ shifted in space so that its z-axis coincides with the RCM point.

In the case of $\underline{\mathbf{x}}_d(l)$, we can use (5) to see that

$$\mathbf{r}^d_{\text{rcm}} = \mathbf{r}_x(l)\mathbf{r}_y(l)\mathbf{r}_z(l) \quad \text{and} \quad \mathbf{t}^d_{\text{rcm}} = \text{translation}(\underline{\mathbf{t}}_z(l)).$$

Now that the descriptions the rotations and constrained translation of $\underline{\mathbf{x}}'(l)$ and $\underline{\mathbf{x}}_d(l)$ were found in relation to the RCM, we then find the intermediary points between them.

To find the incremental rotation, we begin by decomposing the relative rotation between initial and final rotations into $N+1$ equal partial rotations $\mathbf{r}_{\text{inc}}$. By accumulating those $N+1$ rotations we go from the current rotation $\mathbf{r}^l_{\text{rcm}}$ to $\mathbf{r}^d_{\text{rcm}}$, that is

$$\mathbf{r}^l_{\text{rcm}}(\mathbf{r}_{inc})^{N+1} = \mathbf{r}^d_{\text{rcm}} \implies (\mathbf{r}_{inc})^{N+1} = (\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}},$$

then we can use the log operator to obtain $\mathbf{r}_{inc}$

$$(N+1)\log(\mathbf{r}_{inc}) = \log((\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}})$$
$$\therefore \mathbf{r}_{inc} = \exp(\frac{1}{N+1}\log((\mathbf{r}^l_{\text{rcm}})^*\mathbf{r}^d_{\text{rcm}})). \quad (9)$$

And the incremental translation in the z-axis is simply given by

$$\mathbf{t}_{\text{inc}} = \left(\frac{1}{N+1}\right)(\mathbf{t}^d_{\text{rcm}} - \mathbf{t}^l_{\text{rcm}}). \quad (10)$$

Therefore, at each interpolation step $m \in \mathbb{N}$ in the interval $[1, N+1]$, we compose both (9) and (10) to obtain

$$c(i) = \mathbf{r}^l_{\text{rcm}}\mathbf{r}^{\{m\}}_{\text{inc}}(1 + \frac{1}{2}\epsilon(\mathbf{t}^l_{\text{rcm}} + \mathbf{t}^{\{m\}}_{\text{inc}})), \quad (11)$$

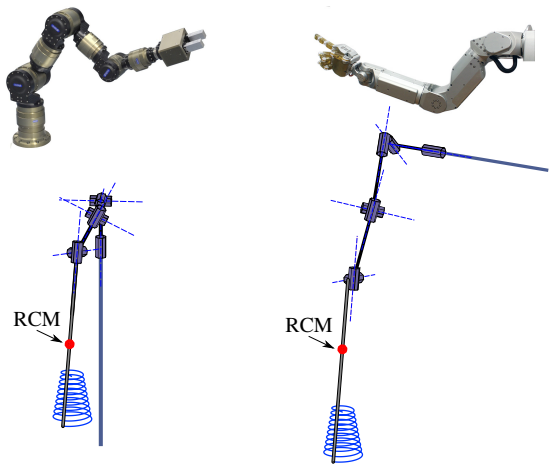
so that the interpolated path is given by

$$\underline{\mathbf{x}}'(l) = \underline{\mathbf{x}}_{\text{rcm}}c(0) \rightarrow \cdots \rightarrow \underline{\mathbf{x}}_{\text{rcm}}c(N+1) = \underline{\mathbf{x}}_d(l).$$

IV. Experimental Evaluation

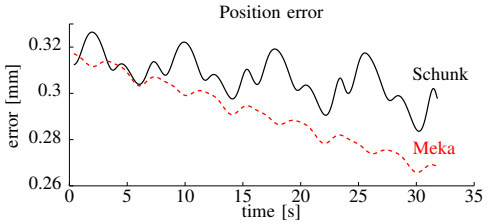
A. Validating the controller

To validate the control strategy and verify its flexibility, we performed computational simulations with two different commercial robots: a Schunk manipulator and a Meka antropomorfic arm. The Schunk LWA3 is a 7DOF manipulator arm with modular architecture, built by Schunk GmbH. The Meka A2 arm, on its turn, is a 7DOF compliant manipulator which is part of the Mekabot Humanoid robot. It is manufactured by Meka Robotics LLC and is a human safe product, intended to be used in cooperative robotics.

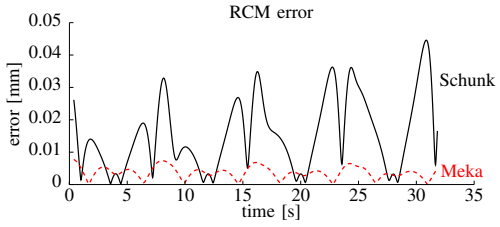


(a) Schunk (b) Meka

Fig. 2: Simulations with a conical helix trajectory tracking.



(a) Tool-tip error to path



(b) RCM point error

Fig. 3: Errors for both Schunk (solid) and Meka (dashed).

In the simulations, a laparoscopic tool was considered to be attached to the end effector of the robots. A predefined conical helix path is given as reference to the tool tip position and the same kinematic controller presented in Section III-B is used to control both manipulators, changing only the robot DH parameters accordingly.

Figure 2 shows the tool tip trajectory for both robots. Even though the two robotic platforms have very distinct design and kinematic models, they both behaved as expected and our proposed controller was able to track the desired path while keeping the RCM pivot constraint as shown in Fig. 3.

The first plot shows the tool tip position error when following the desired trajectory, while the second plot shows the deviation of the tool from the initial RCM point. The obtained error values are negligible during the whole procedure, especially those for the RCM point, which are smaller than 0.05 mm.

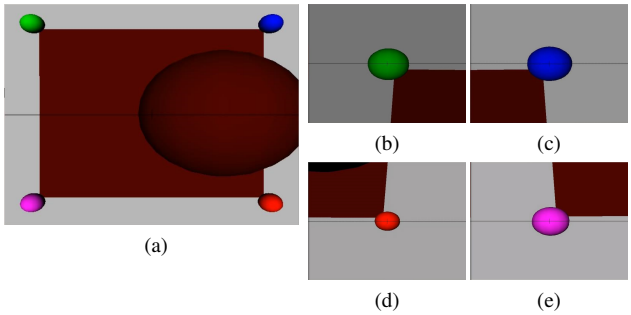


Fig. 4: Relevant camera views. (a) Initial camera position, (b) 1^{st}, (c) 2^{nd}, (d) 3^{rd}, and (e) last objectives.

B. Validating the reference generation

After the initial validation through the helix path tracking, we performed experiments with 20 subjects[1] with no prior medical training. They had to operate an endoscope during a surgical task in a simulated environment.

The subjects used the translational degrees of freedom of an Omega 7 haptics interface to move the camera. The deviations around the initial translation were transformed into ($\mathbf{t}_z$,$\mathbf{r}_\alpha$,$\mathbf{r}_\beta$) and used in (5). As the device has force feedback only on the translational degrees of freedom, this choice allowed us to add a viscosity parameter in the hand movement, both reducing hand tremor and helping slowing the users' movements.

The initial camera view is shown in Fig. 4a, with colored spheres of same radius positioned in the corners of a 5×5 cm square. The user should move around the environment to find the position at which the camera would show the same images and in the same order as Fig. 4b-4e; in the shortest time possible, while avoiding collision with the big obstacle sphere positioned between objectives 2 and 3. When the camera position was within a 1 mm tolerance from the current target, the objective was considered reached and the user should move to the next target. Such small tolerance was selected to increase the difficulty of the task, for a more reliable evaluation of the control system. The experimental setup is shown in Fig. 5. The constants of the kinematic controller (6) were selected as $\mathbf{K} = 60.0$, $\mathbf{K}_c = 7$, and $c_d = 1$; to allow both stability and good responsiveness. The objective function $c(\theta)$ was selected as in [17].

The experimental evaluation process was threefold. First, before each user started the experiment, questions were asked to quantify his previous knowledge concerning laparoscopic procedures, surgical simulators and haptic/game interfaces. Those with no experience in such systems were given two minutes to interact with the experimental setup, while the others had no prior training before executing the task. The goal of this practicing phase was to investigate the learning curve and intuitiveness of the proposed control approach by comparing a novice group with a more experienced group that already had familiarity with similar systems.

Secondly, during the execution of the task, we stored data regarding the desired end effector tracking error, the distance

[1]See provided video for a trial example

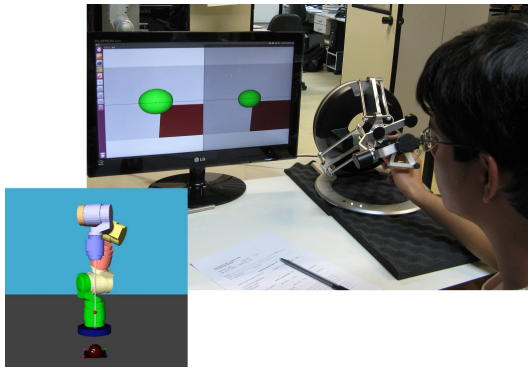


Fig. 5: Subject 18 performing the experiment.

TABLE I: Performance results for 20 trials.

End Effector RMS Error [mm]	0.3398 ± 0.0163
RCM RMS Error [mm]	0.00710 ± 0.00706
Minimal Distance to Obstacle [mm]	12.52 ± 5.03
Completion Time [s]	68.03 ± 30.58

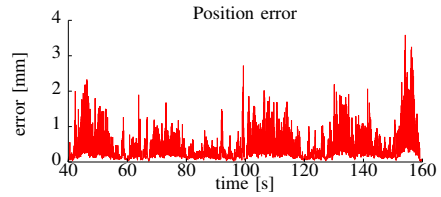
to the obstacle sphere, the RCM error, and the instant each target was reached for all 20 trials. This information was later analyzed for performance evaluation purposes.

Lastly, at the end of the exercise, we asked each subject to evaluate its ease (1 = very difficult, 2 = difficult, 3 = easy, 4 = very easy) and how well he considered his hand movements were translated into camera movements (1 = very bad, 2 = bad, 3 = good, 4 = very good) so that we could qualify the user interaction with the system.

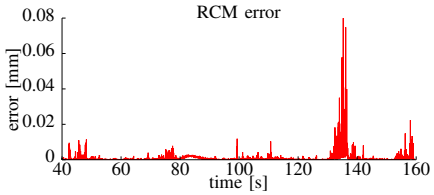
In relation to the system intuitiveness and learning curve, we observed that even with only a few minutes of training, the novice users already presented a similar performance when compared to users from the experts group. All test subjects were able to successfully finish the exercise with a reasonable completion time, having an average of 68.03 s and standard deviation of 30.58 s. The trainees completed the task within 84 seconds in average, while the others took 52 seconds. Also, all users from both groups were able to correctly avoid the obstacle during the test.

With respect to the controller performance and considering all 20 subjects from both groups, the average performance (mean $\pm$ standard deviation) is shown in Table I. The RCM error was kept below 0.37 mm in all experiments, while the end effector positioning error was no bigger than 4.82 mm. The resulting end effector tracking error, RCM error, and performed trajectory in one of the trials are illustrated in Fig. 6, and 7 respectively.

Regarding their interaction with the system, the users evaluated how well the system performed their intended movements with an average of 3.68 points (between good and very good). The most common complaint was the lack of depth perception caused by the flat image of the endoscope. Even so, no subject collided the endoscope with the obstacle. As a result of the system ease of use, the users gave an average of 3.11 points (between easy and very easy) in the task ease scale, even with the small camera position tolerance



(a) End effector error



(b) RCM error

Fig. 6: Performance parameters during one of the trials

and the lack of depth perception they themselves observed.

V. Conclusions and Future work

In this work, we presented a novel intuitive algorithm for robotic control of laparoscopic tools using programmable RCM. Our method is based on the dual quaternion framework which has strong advantages over other kinematic control strategies, such as non-singular representations, more intuitive control laws, control signals obtained directly from the end effector representation, and less expensive computations when compared to homogeneous matrices.

The proposed control strategy is flexible and can be used to different setups with open chain serial link manipulators without significant changes in the algorithms. The controller is also designed to minimize undesirable effects found in previous methods, such as large end effector tracking errors and unrestricted rotations of the tool around its axis.

In order to validate the proposed system, two experiments were performed. First, we used simulations based on two robot models (Schunk LWA3 and Mekabot A2 compliant arm) and controlled them to follow an helical tool tip trajectory. Afterward, the simulated robot was inserted into a 3D environment emulating a surgical task, and coupled with a haptic interface. Twenty subjects participated on tests based on this setup to evaluate the system performance and intuitiveness. The results presented on this paper show that not only the controller can track trajectories with negligible positioning and RCM errors, but also that it can provide effective performance under human control, with a user-friendly and natural operation.

In future works, we intend to evaluate the controller with physicians with different levels of expertise in laparoscopic surgery. Further studies will also focus on the development of an improved singularity avoidance algorithm, and the implementation of the proposed method in a physical experimental setup including a real A2 Meka compliant arm.

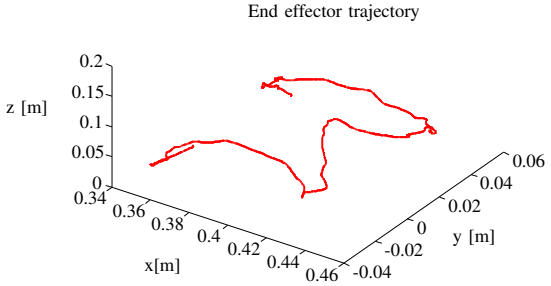


Fig. 7: End effector trajectory

References

[1] G. Fichtinger, P. Kazanzides, A. M. Okamura, G. D. Hager, L. L. Whitcomb, and R. H. Taylor, "Surgical and Interventional Robotics: Part II," *IEEE Robotics & Automation Mag.*, pp. 94–102, 2008.

[2] J. E. N. Jaspers, "Simple Tools for Surgeons. Design and Evaluation of Mechanical Alternatives for "Robotic" Instruments for Minimally Invasive Surgery," PhD thesis, Technische Universiteit Delft, 2006.

[3] R. H. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery," *IEEE Trans. Robotics and Automation*, vol. 19, no. 5, pp. 765–781, 2003.

[4] C.-h. Kuo and J. S. Dai, "Robotics for Minimally Invasive Surgery: A Historical Review from the Perspective of Kinematics," in *Int. Symp. History of Machines and Mechanisms*, 2009, pp. 337–354.

[5] R. C. O. Locke and R. V. Patel, "Optimal Remote Center-of-Motion Location for Robotics-Assisted Minimally-Invasive Surgery," in *Proc. IEEE Int. Conf. Robotics and Automation*, no. April, 2007, pp. 10–14.

[6] M. M. Dalvand and B. Shirinzadeh, "Remote Centre-Of-Motion Control Algorithms of 6-RRCRR Parallel Robot Assisted Surgery System (PRAMiSS)," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012, pp. 3401–3406.

[7] H. Azimian, S. Member, R. V. Patel, and M. D. Naish, "On Constrained Manipulation in Robotics-Assisted Minimally Invasive Surgery," in *Proc. IEEE/RAS-EMBS Int. Conf. Biomedical Robotics and Biomechatronics*, Japan, 2010, pp. 650–655.

[8] B. Adorno, A. Bo, P. Fraisse, and P. Poignet, "Towards a cooperative framework for interactive manipulation involving a human and a humanoid," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011.

[9] L. Figueredo, B. Adorno, J. Ishihara, and G. Borges, "Robust kinematic control of manipulator robots using dual quaternion representation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2013.

[10] M. Michelin, P. Poignet, and E. Dombre, "Dynamic task/posture decoupling for minimally invasive surgery motions: simulation results," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2004, pp. 3625–3630.

[11] H.-j. Cha and B.-j. Yi, "Modeling of a Constraint Force at RCM point in a Needle Insertion Task," in *Proc. IEEE Int. Conf. Mechatronics and Automation*, 2011, pp. 2177–2182.

[12] H. Mayer, I. Nagy, and A. Knoll, "Kinematics and Modelling of a System for Robotic Surgery," in *On Advances in Robot Kinematics*. Springer, 2004, p. 10.

[13] H.-L. Pham, V. Perdereau, B. V. Adorno, and P. Fraisse, "Position and Orientation Control of Robot Manipulators Using Dual Quaternion Feedback," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 658–663.

[14] B. Akyar, "Dual Quaternions in Spatial Kinematics in an Algebraic Sense," *Turkish Journal of Mathematics*, vol. 32, no. 4, pp. 373–391, May 2008.

[15] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robotics and Automation*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[16] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, pp. 868–871, Dec. 1977.

[17] S. Chiaverini and B. Siciliano, "Kinematic analysis and singularity avoidance for a seven-joint manipulator," *American Control Conference*, pp. 2300–2305, 1990.

[18] J. Kim, G. Marani, W. K. Chung, and J. Yuh, "A general singularity avoidance framework for robot manipulators: task reconstruction method," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2004.