Robotica: page 1 of 23. © Cambridge University Press 2019
doi:10.1017/S0263574719001735

Unilaterally Constrained Motion of a Curved Surgical Tool

Bassem Dahroug*, Brahim Tamadazte and Nicolas Andreff

AS2M Department, FEMTO-ST Institute, Univ. Bourgogne Franche-Comté/CNRS, 24 Rue Alain Savary, 25000 Besançon, France
E-mails: brahim.tamadazte@femto-st.fr, nicolas.andreff@femto-st.fr

(Accepted November 1, 2019)

SUMMARY
Constrained motion is essential for varying robotics tasks, especially in surgical robotics, for instance, the case of minimally invasive interventions. This article proposes generic formulations of the classical bilateral constrained motion (i.e., when the incision hole has almost the same diameter as that of the tool) as well as unilaterally constrained motion (i.e., when the hole incision has a larger diameter compared to the tool diameter). One of the latter constraints is combined with another surgical task such as incision/ablation or suturing a wound (modeled here by 3D geometric paths). The developed control methods based on the hierarchical task approach are able to manage simultaneously the constrained motion (depending on the configuration case, i.e., bilateral or unilateral constraint) and a 3D path following. In addition, the proposed methods can operate with both straight or curved surgical tools. The proposed methods were successfully validated in various scenarios. Foremost, a simulation framework was proposed to access the performances of each proposed controller. Thereafter, several experimental validations were carried out. Both the simulation and experimental results have demonstrated the relevance of the proposed approach, as well as promising performances in terms of behavior as well as accuracy.

KEYWORDS: Constrained motion modeling; Bilateral and unilateral constraints; 3D path following; Hierarchical tasks; Medical robotics.

1. Introduction

Minimally invasive surgical (MIS) robotic systems enter into the human body either from a natural orifice (anal cavity, mouth, urethra, nasal cavity, etc.) or from an artificial (i.e., created) hole. In both cases, the robotic tool must avoid damaging the incision walls and the anatomical structures within the patient's body. Most of the medical purposes (e.g., laparoscopic[1,2] or eye[3,4] surgeries) consider that the incision hole diameter and that of the surgical tool are nearly equal.

Throughout this case, the tool linear motion is restricted locally along two axes of the incision frame (e.g., the x-axis and z-axis as shown in Fig. 1(a)). Hence, the center point of the incision hole (or the trocar center point) should constantly lay on the tool center line. The resulting constrained motion is typically called *Remote Center of Motion* (RCM) (or *fulcrum effect*).[5,6] Such constraints are mathematically modeled with equality constraint equations, also known as bilateral constraints in optimization theory.[7]

Nevertheless, bilateral constraints are excessively restrictive during other surgery types (e.g., in the nose[8,9] or the ear[10–12]) where the orifice size is bigger than that of the tool. It implies that the

* This work has been supported by the ANR μRoCS Project (ANR-17-CE19-0005-04).
Corresponding author. E-mail: bassem.dahroug@femto-st.fr

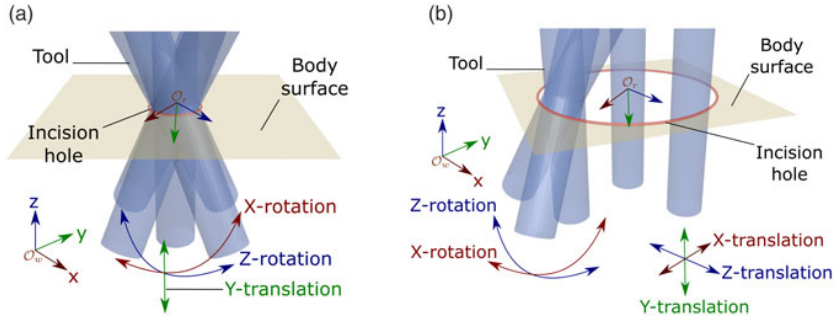


Fig. 1. A comparison between constrained motion: (a) RCM and (b) UCM.

orifice wall forms locally a cylinder around the instrument body. In this case, the tool is unconstrained within the hole before it hits the orifice wall (Fig. 1(b)). When the tool contacts the wall, its motion is constrained to slide along the wall contact point (i.e., performing the RCM movement at the contact point) to reach targets deeper inside the body. This motion is therefore restricted by inequality constraints and will be named within this paper as *Unilaterally Constrained Motion* (UCM).

A complex surgical task (e.g., suturing a wound, examining a region of interest, or excising pathological tissues) requires defining one of these constrained motions as a subtask. Indeed, this complex surgical task can be defined as a concatenation of several elementary subtasks, that is, a path following (representing an incision/ablation surgical gesture) under the RCM/UCM constraints. Setting up a hierarchy between these subtasks offers a solution to stack all of them. Thereby, the priority technique[13] based on the projection gradient method[14] is applied for projecting a secondary task into the null space of the primary task. This task hierarchical management ensures to find out the secondary solution that does not generate conflict with the primary solution. This formalism was used to build a complex task from individual subtasks as in ref. [15].

Moreover, surgical tasks in confined spaces, for instance, the middle ear cavity or the superior part of the sinuses, are complex assignments. They impose the usage of the curved tools to (i) increase the surgeon's dexterity and (ii) allow reaching distal regions that cannot be accessed with conventional straight instruments. This makes it even harder to cope with both the clinical task and the anatomical constraints. Thereby, it is necessary to formulate a *generic solution* for handling also curved surgical tools under geometric (bilateral or unilateral) constraints.

Consequently, the article contributions provide a unified approach to manipulate either a straight or curved tool for following a user-defined 3D path under unilateral or bilateral anatomical constraints. In the remainder of this paper, Section 2 discusses the state of the art dealing with constrained motion control as well as the reasons why our proposed method goes beyond the current methods. Section 3 recalls how a straight tool follows a reference path under bilateral constraints. After that, Section 4 proposes the formulation of bilateral constraints with a curved tool, while Section 5 discusses the model of a curved tool that follows a path under unilateral constraints. The proposed methods were validated in both numerical (simulation) and experimental scenarios as discussed in Section 6 and 7, respectively.

2. Overview and Contributions

2.1. Overview

From a mathematical point of view, RCM is considered as a linear (respectively, nonlinear) equality system, where the system function must be equal to zero (i.e., $f(\mathbf{x}) \leq \mathbf{0}$ and $f(\mathbf{x}) \geq \mathbf{0} \Leftrightarrow f(\mathbf{x}) = \mathbf{0}$, as depicted in Fig. 2(a)). In opposition, UCM is considered as a linear (respectively, nonlinear) inequality system, where the system function should be exclusively positive or negative (i.e., $f(\mathbf{x}) \leq \mathbf{0}$ xor $f(\mathbf{x}) \geq \mathbf{0}$, Fig. 2(b)).

RCM has been widely discussed in the literature from a mechanical viewpoint as, for instance, in refs. [5, 16]. Indeed, these works built a special kinematic structure in order to ensure the fulcrum effect constraints. The advantage of such a method is to physically impose the constrained motion the relatively easy formulation of the resulting control law. Despite that, a special structure does not provide enough flexibility to change the location of penetration point during the surgery. On

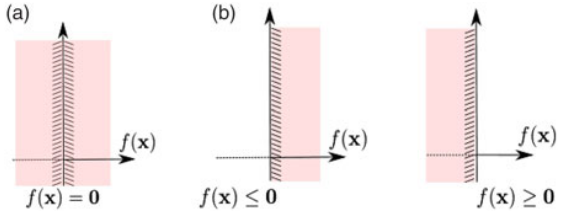


Fig. 2. A comparison from a mathematical viewpoint between (a) bilateral constraints and (b) unilateral constraints.

the opposition, a software controller can also achieve the required motion by guiding a general-purpose robot that is commercially present with a large range of models. The latter solution releases the mechanical constraints from the robot structure and adds the versatility of use thanks to the redundancy.

There are various methods reported in the literature for solving RCM by control, such as (i) extended Jacobian with quadratic optimization,[17] (ii) gradient projection approach in closed-loop scheme,[18] (iii) dual quaternion-based kinematic controller,[19] and (iv) constrained Jacobian represented with *Lie Algebra*.[20] These methods impose the bilateral constraints in the joint space (i.e., onto the robot kinematic Jacobian matrix $\mathbf{J}(\mathbf{q})$) for restricting the kinematics to $\underline{\mathbf{v}}_e = \mathbf{J}_{con}(\mathbf{q})\dot{\mathbf{q}}$. The new constrained Jacobian matrix $\mathbf{J}_{con} \in \mathbb{R}^{6\times n}$ is a transformation that maps the joints velocities $\dot{\mathbf{q}} \in \mathbb{R}^{n\times 1}$ into the end-effector twist vector $\underline{\mathbf{v}}_e \in se(3)$. However, this formulation requires an accurate knowledge of the robot kinematic model in order to implement the control law.

Furthermore, other authors have been proposed different approaches to formulate the constrained motion as (i) analytical solution based on trocar modeling with Euler angles representation,[21] (ii) geometric solution-based heuristic search,[22] and (iii) geometric constraint with stereo visual servoing.[1] These methods are more generic than the previous ones to perform the fulcrum effect. They are unlimited to a specific kinematic structure since they impose the bilateral constraints in the task space (i.e., constraints on the interaction matrix as $\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}\underline{\mathbf{v}}_e$). Indeed, the interaction matrix $\mathbf{L}_{\mathbf{s}} \in \mathbb{R}^{m\times 6}$ relates the features velocities $\dot{\mathbf{s}} \in \mathbb{R}^{m\times 1}$ and the end-effector twist vector. Such formulation does not necessarily require an accurate knowledge of the robot inverse kinematics.

Moreover, the previous methods applied a trajectory tracking scheme as a secondary task. However, trajectory tracking methods are less appropriate for medical applications as discussed in ref. [23], since the trajectory controller is limited in terms of accuracy and stability (e.g., when dealing with complex geometric curves with high curvatures) and more complicated to implement compared to a path-following scheme. Thereby, the previous work[6] assembled individual tasks (i.e., both the RCM and the path-following tasks) to perform a complex motion within the patient body by handling a straight tool. The projection gradient technique was also applied to guarantee the hierarchy/priority between the various tasks. Despite that this controller cannot manage the UCM case.

In the literature, constraining motion in a given direction was initially introduced by Rosenberg[24] to formulate the so-called *virtual fixtures*. This method was used to guide a teleoperated robot by applying a hybrid force/motion control. Furthermore, the various techniques associated with the virtual fixtures in medical applications are presented in a very interesting survey.[25] Note that the admittance control is derived from the virtual fixture method. It is also based on the kinematic and dynamic models of the robot as well as the interaction model with its environment. Indeed, this technique regulates the velocities/forces interacted with the environment by imposing a deviation from the desired motion. Besides that, it is considered as an indirect force control (i.e., the force measurement is not performed). Finally, this approach was implemented in the described work in this paper by deducing the interaction matrix that calculates the control velocities for achieving the UCM.

2.2. *Contributions*

This paper has the objective to formulate the UCM movement in an easy and versatile geometric form. Indeed, the previous work in ref. [6] handles a straight tool to perform a path following under bilateral constraint, while this article extended the controller capability by guiding a curved

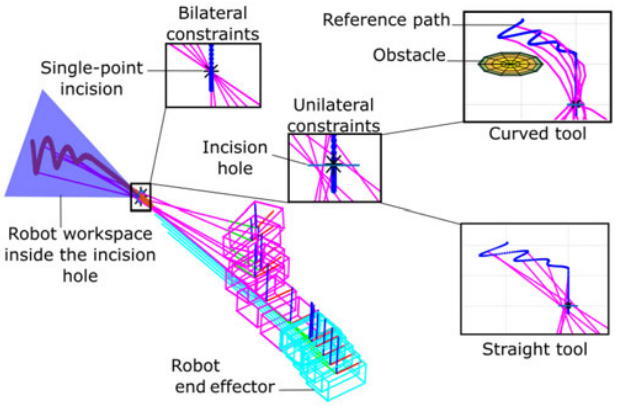


Fig. 3. A synthetic view of the various cases that can be handled by the proposed controller.

instrument to achieve a path following under unilateral constraint. The mathematical formulation in the next sections shows that a straight tool can be considered as an especial case of a curved one. Consequently, the new controller becomes more generic. It deduces the spatial velocity of the robot end effector for manipulating a rigid instrument through an orifice whose diameter can either be equal or larger than that of the tool.

Note that the curved geometry provides the system/surgeon with more dexterity for avoiding obstacles as shown in Fig. 3. This figure presents a general case where the tool first moves from its initial position toward the entrance of the incision hole (i.e., *outside phase*). Then, in a second phase (i.e., the *inside phase*), the tool follows a 3D path (e.g., an incision task, represented here without loss of generality by a spiral) under the constraints imposed by the incision hole. The workspace inside the incision hole is represented by the blue triangle in Fig. 3. Zoom was made on the incision hole in order to demonstrate the difference between the RCM and UCM movements. The upper left zoom plot entitled "*Bilateral constraints*" in this figure shows how the tool body always passes through the orifice center point. On the opposite, the central zoom plot entitled "*Unilateral Constraints*" demonstrates that the tool body moves within the orifice whose wall is represented by a blue horizontal line in this plot. The right zoom plots reveal how a curved tool has a better dexterity for avoiding obstacles, compared to a straight tool, under the unilateral constraints.

3. Background: Time-Independent Control of Straight Tools along a Surgical Path under Bilateral Constraints

3.1. Notations

In this paper, we opted for the notations and symbols summarized in Table I. For instance, bold nonitalic letters denote vectors (e.g., $\mathbf{x}$) and bold nonitalic uppercase letters represent matrices (e.g., $\mathbf{X}$). Scalar values and continuous-time functions will be denoted by italic letters (e.g., x). Also, a vector distance from a point a to a point b is written as a vector by $\mathscr{V}_{ab}$. The frame attached to an origin $\mathcal{O}_i$ is represented by $\Re_i$.

3.2. Medical gesture as a 3D path-following scheme

In MIS applications, the physician may need to excise or scan a pathological tissue by using an ablation tool (e.g., laser) or an optical endoscopic probe, respectively. Indeed, for better surgical task achievement, the tool tip should move accurately on the tissue with a velocity that is mostly independent of the path geometrical shape. Furthermore, the clinician does not possess precise knowledge of the tool tip velocity. Furthermore, he/she may need to change this velocity online (while the robot is executing the reference geometrical curve).

Consequently, instead of using a 3D trajectory tracking technique, we opted for a 3D path-following scheme more adequate to the clinical requirements.[23] Recently, in ref. [26], we demonstrated the preliminary benefits of this approach in the case of middle ear surgery (using a laboratory

Table I. Notations summary.

Symbol	Description
$\mathcal{O}_i$	Origin point of a 3D frame;
$\Re_i$	Frame attached to the origin point $\mathcal{O}_i$;
${}^i\mathbf{y}$	y-axis (basis vector) of $\Re_i$;
$\mathbf{I}_{n\times m}$	$n \times m$ identity matrix;
$\mathbf{X}_{n\times m}$	$n \times m$ matrix;
$\mathbf{x}$, $\mathbf{x}^\top$	Vector and its transpose, respectively;
$\mathbf{x}_1 \times \mathbf{x}_2$	Cross product between vectors $\mathbf{x}_1$ and $\mathbf{x}_2$;
$[\mathbf{x}]_\times$	Skew matrix associated to the vector $\mathbf{x}$;
$\|\mathbf{x}\|$	Euclidean norm of the vector $\mathbf{x}$;
$\dot{\mathbf{x}}$	Time derivation of the vector $\mathbf{x}$;
$\mathscr{V}_{ab}$	Vector between the $\mathcal{O}_a$ and $\mathcal{O}_b$;
$\dot{\mathscr{V}}_{ab}$	Time derivation of the vector $\mathscr{V}_{ab}$;
${}^i\mathbf{u}_{ab}$	Unit vector of ${}^i\mathbf{ab}$ expressed in $\Re_i$;
${}^i\dot{\mathbf{u}}_{ab}$	Time derivative of the unit vector ${}^i\mathbf{u}_{ab}$;
s_t	Length of a curve arc;
$\dot{s}_t$	Curvilinear speed of a point $\mathbf{p}$;
$S_t(s_t)$	Tool curved shape function of s_t;
$\mathbf{p}_{t'}$	Point along a curve (curved tool);
${}^e\mathbf{k}_t$	Instantaneous tangential unit vector of the tool shape;
$\mathbf{r}_{C_t}$	Curve (curved tool) radius at a given point;
$\mathbf{C}_t$	Curvature of the tool which is the inverse of the radius $\mathbf{r}_{C_t}$;
$\{\mathbf{k}, \mathbf{j}, \mathbf{i}\}$	Constructed basis using three unit vectors;
$\mathbf{d}_{rcm}$	Linear error of alignment task;
d_{rcm}	Projection of $\mathbf{d}_{rcm}$ along an axis;
d_{min}, d_{max}	Minimum and maximum distance d_{rcm} in case of UCM method, respectively;
$\mathbf{d}_{pf}$	Path-following error;
$\mathbf{p}_p$	Projected 2D point onto the path S_p to be followed;
Δt	Small time step;
$\mathbf{e}_{rcm}$	Angular error of the alignment task;
$\dot{\mathbf{e}}_{rcm}$	Time derivative of the angular RCM error;
${}^i\mathbf{v}_j$	Linear velocity of j expressed in i;
${}^i\omega_j$	Angular velocity of j expressed in i;
${}^i\underline{\mathbf{v}}_i$	Twist velocity vector of i composed of $({}^i\underline{\mathbf{v}}_i = \left[{}^i\mathbf{v}_i; {}^i\,\omega_i\right] \in \mathbb{R}^{6\times 1})$;
$\mathbf{v}_{adv}$	Advance velocity along a path;
$\mathbf{v}_{ret}$	Regulation velocity of the tool deviation;
$\mathbf{L}_{e_{rcm}}$	3×6 interaction matrix of the alignment task;
$\mathbf{L}^\dagger_{e_{rcm}}$	6×3 inverse matrix of $\mathbf{L}_{e_{rcm}}$;
λ	A positive scalar gain for the alignment task;
α_{obs}	Stiffness of a virtual spring;
σ_{min}, σ_{max}	Minimum and maximum stiffness of the virtual spring, respectively;
σ_{step}	Step of change of the virtual spring stiffness;
α, β	Gain coefficients for adjusting the priority between $\mathbf{v}_{adv}$ and $\mathbf{v}_{ret}$.

test bench). Thus, this section recalls the formulation of the 3D path-following controller which is based on the *Frenet–Serret* frame representation.

Let us consider that the path-following error $\mathbf{d}_{pf}$ as depicted in Fig. 4(a). This error is determined by projecting the tool tip $\mathcal{O}_t$ onto the reference path $\mathcal{S}_p(s_p)$, which results in a projected point $\mathbf{p}_p$. Thus, $\mathbf{d}_{pf}$ is obtained by

$$\mathbf{d}_{pf} = \mathcal{O}_t - \mathbf{p}_p. \tag{1}$$

At present, the challenge is to find the adequate controller that deduces the tool tip linear velocity $\mathbf{v}_t$ for minimizing the projection distance $\mathbf{d}_{pf}$ expressed in (1). To tackle this, the proposed controller decomposes the latter velocity $\mathbf{v}_t$ into two components (Fig. 4(b)): (i) the advance velocity $\mathbf{v}_{adv}$ along the path and (ii) the return velocity $\mathbf{v}_{ret}$ for regulating the tool deviation from the path.

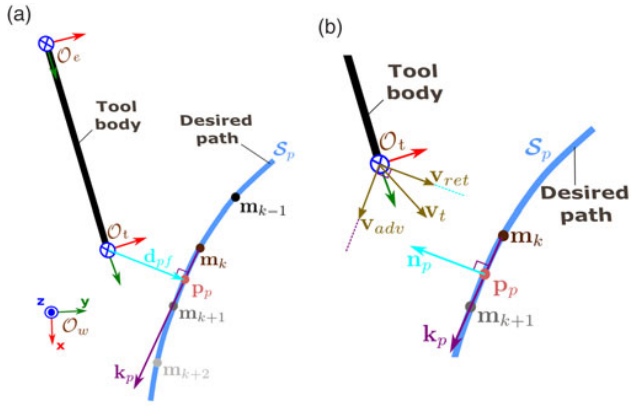


Fig. 4. The different reference frames and notions used during the path-following controller design, where (a) projection of the tool tip onto a geometric curve and (b) concept of path-following approach.

One can formulate the previous concept as follows:

$$\mathbf{v}_t = \underbrace{\alpha \mathbf{k}_p}_{=\mathbf{v}_{adv}} + \underbrace{\beta \mathbf{d}_{pf}}_{=\mathbf{v}_{ret}} \tag{2}$$

whereby α and β are the gain coefficients for adjusting the priority between the advance and return velocities, respectively, and $\mathbf{k}_p$ is the instantaneous tangential unit vector to the geometric curve (Fig. 4(b)).

A possible option is imposing a constant velocity to the tool as $\|\mathbf{v}_t\|^2 = v_{tis}^2$ in order to deduce the values of the gain coefficients α and β. Indeed, the velocity v_{tis} depends on the interaction between the surgical tool and the tissue homogeneity. Therefore, (2) can be reformulated as follows:

$$v_{tis}^2 = \alpha^2 \underbrace{\|\mathbf{k}_p\|^2}_{=1} + \underbrace{\beta^2 \|\mathbf{d}_{pf}\|^2}_{=\|\mathbf{v}_{ret}\|^2}. \tag{3}$$

So,

$$\text{if} \quad \begin{cases} \|\mathbf{v}_{ret}\|^2 < v_{tis}^2, \begin{cases} \beta = constant \\ \alpha = \sqrt{\|\mathbf{v}_{ret}\|^2 + v_{tis}^2} \end{cases} \\ \|\mathbf{v}_{ret}\|^2 > v_{tis}^2, \begin{cases} \beta = constant \\ \alpha = 0 \end{cases} \end{cases}. \tag{4}$$

The choice of the gain parameters is determined in function of the clinical needs (e.g., the velocity of the surgical tool with respect to the tissue type). For instance, if the tool is not far from the desired path, the first condition in (4) is chosen. Otherwise, the priority is returning the tool tip to the reference path (i.e., second condition in (4)).

The resultant control velocity of the tool tip (2) could be represented with respect to any desired frame. Then, by choosing the end-effector frame $\Re_e$, the twist velocity of end effector ${}^e\underline{\mathbf{v}}_e \in \mathbb{R}^{6\times 1}$ is related to the tool tip linear velocity ${}^e\mathbf{v}_t \in \mathbb{R}^{3\times 1}$ by the interaction matrix $\mathbf{L}_{e_{pf}} \in \mathbb{R}^{3\times 6}$ as follows:

$${}^e\mathbf{v}_t = \underbrace{\begin{pmatrix} \mathbf{I}_{3\times 3} & -\left[{}^e\mathscr{V}_{et}\right]_\times \end{pmatrix}}_{\mathbf{L}_{e_{pf}}} \underbrace{\begin{pmatrix} {}^e\mathbf{v}_e \\ {}^e\omega_e \end{pmatrix}}_{{}^e\underline{\mathbf{v}}_e} \tag{5}$$

whereby $\left[{}^e\mathscr{V}_{et}\right]_\times$ is the antisymmetric matrix of the vector ${}^e\mathscr{V}_{et}$ which begins at the end effector $\mathcal{O}_e$ and ends at the tool tip $\mathcal{O}_t$.

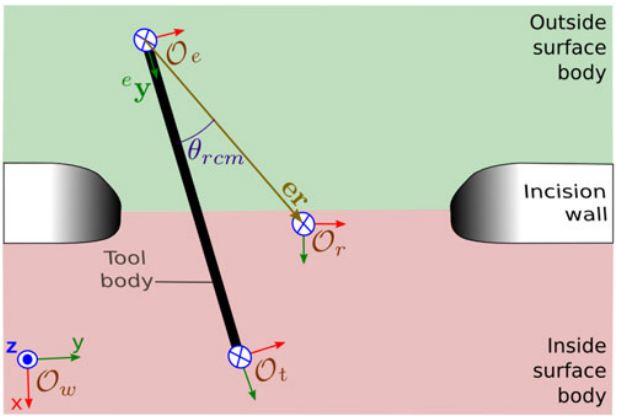


Fig. 5. The different reference frames and notions used within the formulation of the RCM (bilateral) problem.

3.3. RCM with straight tool

First, let us consider the following reference frames that are necessary to express the different transformations:

- $\Re_e$ attached to the robot end effector, having as origin the point $\mathcal{O}_e$;
- $\Re_t$ linked to the surgical tool with $\mathcal{O}_t$, the related origin point of the frame;
- $\Re_r$ fixed at the center of the incision point or natural orifice denoted $\mathcal{O}_r$.

In ref. [6], the alignment condition of RCM task was proposed as the geodesic error between two vectors: (i) the unit vector of $^e\mathscr{V}_{er}$ (i.e., $^e\mathbf{u}_{er}$) which is formed between the origin points $\mathcal{O}_e$ and the $\mathcal{O}_r$ (see Fig. 5) and (ii) the y-component of the end-effector frame $^e\mathbf{y}$. These vectors are expressed with respect to the end-effector frame $\Re_e$ (i.e., the left superscript). The RCM angular error is thus formulated as the cross product given by

$$\mathbf{e}_{rcm} = {}^e\mathbf{y} \times {}^e\mathbf{u}_{er}. \tag{6}$$

Recall that a velocity controller is needed to reduce the RCM angular error. Thereby, the time derivative of the unit vector $^e\mathbf{u}_{er}$ is computed as

$$^e\dot{\mathbf{u}}_{er} = \left(\frac{\mathbf{I}_{3\times 3}}{\|{}^e\mathscr{V}_{er}\|} - \frac{{}^e\mathbf{u}_{er}\,{}^e\mathbf{u}_{er}^\top}{\|{}^e\mathscr{V}_{er}\|} \right) {}^e\dot{\mathscr{V}}_{er} \tag{7}$$

where $\|\mathbf{x}\|$, $\mathbf{x}^\top$, and $[\mathbf{x}]_\times$ are the Euclidean norm, transpose, and the antisymmetric matrix associated with the vector $\mathbf{n}$, respectively, and $\mathbf{I}$ is a 3×3 identity matrix.

Besides that, the term $^e\dot{\mathscr{V}}_{er}$ in the latter equation represents the linear velocity of the incision frame $\Re_r$ with respect to the end-effector frame $\Re_e$ (i.e., $^e\dot{\mathscr{V}}_{er} = {}^e\mathbf{v}_r$). This linear velocity $^e\mathbf{v}_r$ can be related to the end-effector twist vector $^e\underline{\mathbf{v}}_e$ as

$$^e\mathbf{v}_r = \underbrace{\begin{pmatrix} \mathbf{I}_{3\times 3} & -[{}^e\mathscr{V}_{er}]_\times \end{pmatrix} \begin{pmatrix} {}^e\mathbf{v}_e \\ {}^e\omega_e \end{pmatrix}}_{{}^e\underline{\mathbf{v}}_e}. \tag{8}$$

Consequently, the unit vector $^e\dot{\mathbf{u}}_{er}$ is reformulated as

$$^e\dot{\mathbf{u}}_{er} = \underbrace{\frac{-1}{\|{}^e\mathscr{V}_{er}\|} \left(\mathbf{I}_{3\times 3} - {}^e\mathbf{u}_{er}\,{}^e\mathbf{u}_{er}^\top \right) \begin{pmatrix} \mathbf{I}_{3\times 3} & -[{}^e\mathscr{V}_{er}]_\times \end{pmatrix}}_{\mathbf{L}_{u_{er}}} {}^e\underline{\mathbf{v}}_e \tag{9}$$

where $\mathbf{L}_{u_{er}} \in \mathbb{R}^{3\times 6}$ is the interaction matrix that relates the rate of change of the unit vector $^e\dot{\mathbf{u}}_{er}$ to the spatial velocity of the end effector expressed in its own frame $^e\underline{\mathbf{v}}_e$.

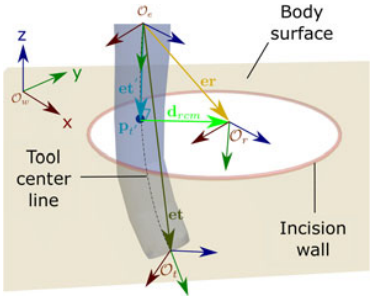


Fig. 6. A conceptual scheme to show the notions used for a curved tool with respect to the incision hole.

Note that a negative sign was added to (9) because the incision point frame $\Re_r$ moves in the opposite direction of that of the end effector $\Re_e$ for reducing the RCM error.

4. Time-Independent Control of a Curved Tool along a Surgical Path under Bilateral Constraints

The alignment condition of the tool body with the incision point in (6) is only valid in case of a straight tool since the vector ${}^e\mathscr{V}_{et}$ is always aligned with the basis ${}^e\mathbf{y}$. Thereby, if the tool is curved, the vector ${}^e\mathscr{V}_{et}$ is no longer aligned with the basis ${}^e\mathbf{y}$ (Fig. 6).

As a result, the alignment condition in (6) can be modified with the aim to be more generic for any tool shape. One can tackle this, by tracking $\mathbf{p}_{t'}$ which is the orthogonal projection of the incision point $\mathcal{O}_r$ onto the curved tool body. The resultant projection distance $\mathbf{d}_{rcm}$ can be considered as the RCM lateral error. It is thus evaluated as follows:

$$\mathbf{d}_{rcm} = \left(\mathbf{I}_{3\times 3} - {}^e\mathbf{u}_{et'}\ {}^e\mathbf{u}_{et'}^{\top}\right){}^e\mathscr{V}_{er} \tag{10}$$

whereby ${}^e\mathbf{u}_{et'}$ is the unit vector of ${}^e\mathscr{V}_{et'}$, which is formed between the origin point $\mathcal{O}_e$ and the projected point $\mathbf{p}_{t'}$.

After deducing the projected point $\mathbf{p}_{t'}$, the unit vector ${}^e\mathbf{u}_{et'}$ replaces the vector ${}^e\mathbf{y}$ in (6). Consequently, the RCM angular error can be rewritten as

$$\mathbf{e}_{rcm} = {}^e\mathbf{u}_{et'} \times {}^e\mathbf{u}_{er}, \tag{11}$$

and its time derivative is computed as

$$\dot{\mathbf{e}}_{rcm} = {}^e\mathbf{u}_{et'} \times {}^e\dot{\mathbf{u}}_{er} + {}^e\dot{\mathbf{u}}_{et'} \times {}^e\mathbf{u}_{er}. \tag{12}$$

Proposition 1. The rate of change of the alignment task is obtained as follows:

$$\dot{\mathbf{e}}_{rcm} = \underbrace{\left([{}^e\mathbf{u}_{et'}]_{\times}\mathbf{L}_{u_{er}} - [{}^e\mathbf{u}_{er}]_{\times}\mathbf{L}_{u_{et'}}\right)}_{\mathbf{L}_{e_{rcm}}} {}^e\underline{\mathbf{v}}_e \tag{13}$$

where $\mathbf{L}_{e_{rcm}} \in \mathbb{R}^{3\times 6}$ is the interaction matrix of the alignment task that relates the rate of change of the RCM angular error $\dot{\mathbf{e}}_{rcm}$ with the end-effector twist vector ${}^e\underline{\mathbf{v}}_e$.

Proof. In order to demonstrate Proposition 1, it is required to know all components of equation (12). The unit vector ${}^e\mathbf{u}_{er}$ can be determined trivially while its time derivative ${}^e\dot{\mathbf{u}}_{er}$ is formulated in (9). The vector ${}^e\dot{\mathbf{u}}_{et'} = \mathbf{L}_{u_{et'}}\ {}^e\underline{\mathbf{v}}_e$ is formally deduced in a similar way to (9). By substituting these variables in (12), we can find (13). □

Now, the expression of $\mathbf{L}_{u_{et'}}$ needs to be expressed.

Lemma 1. *The interaction matrix* $\mathbf{L}_{u_{et'}}$ *is computed as*

$$\mathbf{L}_{u_{et'}} = \frac{1}{\|{}^e\mathbf{et}'\|}\left(\mathbf{I}_{3\times 3} - {}^e\mathbf{u}_{et'}\ {}^e\mathbf{u}_{et'}^{\top}\right)\mathbf{L}_{et'} \tag{14}$$

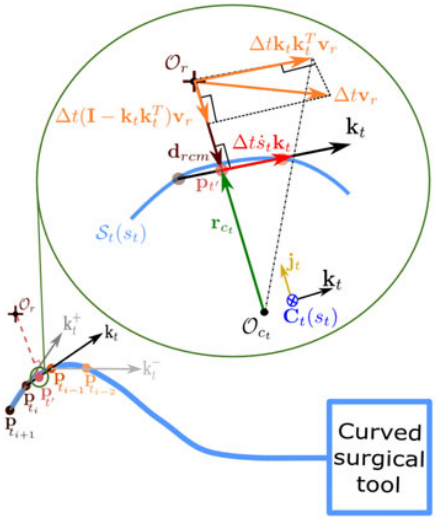


Fig. 7. The geometrical description for (i) projection of the incision center point $\mathcal{O}_r$ along the curved tool and (ii) computation of the velocity of the projected point $\mathbf{p}_{t'}$ along the curved tool.

where $\mathbf{L}_{et'}$ *is a* 3×6 *interaction matrix that relates the rate of change* ${}^e\dot{\mathscr{V}}_{et'}$ *with the end-effector twist vector* ${}^e\underline{\mathbf{v}}_e$.

Proof. The time derivative of the unit vector ${}^e\mathbf{u}_{et'}$ is deduced similar to (7) as

$$ {}^e\dot{\mathbf{u}}_{et'} = \frac{1}{\|{}^e\mathbf{et}'\|}\Big(\mathbf{I}_{3\times3} - {}^e\mathbf{u}_{et'}\,{}^e\mathbf{u}_{et'}^T\Big)\,{}^e\dot{\mathscr{V}}_{et'}. \tag{15} $$

The time derivative of the vector ${}^e\dot{\mathscr{V}}_{et'}$ represents the velocity of the projection point ${}^e\mathbf{p}_{t'}$ along the tool shape $\mathcal{S}_t(s_t)$, which is a function of the curve arc length s_t. It is also related to the end-effector twist vector ${}^e\underline{\mathbf{v}}_e$ as follows:

$$ {}^e\dot{\mathscr{V}}_{et'} = \mathbf{L}_{et'}\,{}^e\underline{\mathbf{v}}_e. \tag{16} $$

By substituting the latter equation in (15), the matrix $\mathbf{L}_{u_{et'}}$ is deduced as (14). □

The next lemma shows how to calculate the interaction matrix $\mathbf{L}_{et'}$.

Lemma 2. *The interaction matrix* $\mathbf{L}_{et'}$ *is computed as follows:*

$$ \mathbf{L}_{et'} = \frac{{}^e\mathbf{k}_t\,{}^e\mathbf{k}_t^\top}{1 + \mathbf{d}_{rcm}^\top\Big(\mathbf{C}_t(s_t) \times {}^e\mathbf{k}_t\Big)}\Big(\mathbf{I}_{3\times3} \quad -[{}^e\mathscr{V}_{er}]_\times\Big) \tag{17} $$

whereby $\mathbf{C}_t$ *is the tool curvature, and* ${}^e\mathbf{k}_t$ *is the instantaneous tangential unit vector on the tool shape.*

Proof. The time derivative of the vector ${}^e\dot{\mathscr{V}}_{et'}$ can be determined as

$$ \begin{aligned} {}^e\dot{\mathscr{V}}_{et'} &= \frac{\partial\,{}^e\mathscr{V}_{et'}}{\partial s_t}\frac{ds_t}{dt} \\ &= {}^e\mathbf{k}_t\dot{s}_t \end{aligned} \tag{18} $$

The time derivative of s_t is nothing more than the curvilinear speed of $\mathbf{p}_{t'}$ along the curved tool. Indeed, the tool shape is defined by the tool curvature $\mathbf{C}_t$, which is determined by a radius of curvature $\mathbf{r}_{C_t}$ and its center point $\mathcal{O}_{C_t}$, as depicted in Fig. 7, while the unit vector ${}^e\mathbf{k}_t$ could be approximated by two consecutive sample points on the tool body.

Now, the main problem becomes how to deduce the speed $\dot{s}_t$ in order to determine the time derivative ${}^e\dot{\mathscr{V}}_{et'}$. Therefore, a reference frame is placed at the center point $\mathcal{O}_{C_t}$, and its basis is formed by $\{{}^e\mathbf{k}_t,\ {}^e\mathbf{j}_t,\ {}^e\mathbf{i}_t\}$ as

$$ {}^{e}\mathbf{k}_t = \frac{{}^{e}\mathbf{p}_{t_{i-1}} - {}^{e}\mathbf{p}_{t_i}}{\|{}^{e}\mathbf{p}_{t_{i-1}} - {}^{e}\mathbf{p}_{t_i}\|}, \tag{19} $$

$$ {}^{e}\mathbf{j}_t = \frac{\mathbf{r}_{C_t}}{\|\mathbf{r}_{C_t}\|}, \tag{20} $$

$$ {}^{e}\mathbf{i}_t = {}^{e}\mathbf{j}_t \times {}^{e}\mathbf{k}_t. \tag{21} $$

The radius of curvature $\mathbf{r}_{C_t}$ is in the same direction as the basis ${}^{e}\mathbf{j}_t$ (Fig. 7), then it is defined as

$$ \mathbf{r}_{C_t} = r_{C_t}\, {}^{e}\mathbf{j}_t, \tag{22} $$

with $r_{C_t} = \|\mathbf{r}_{C_t}\| \in \mathbb{R}^+$.

Furthermore, the vector $\mathbf{d}_{rcm}$ is also collinear with ${}^{e}\mathbf{j}_t$, and its direction could be either in the same or in the opposite direction with/to that of the basis ${}^{e}\mathbf{j}_t$. Consequently, the vector $\mathbf{d}_{rcm}$ could be reformulated as follows:

$$ \mathbf{d}_{rcm} = \underbrace{\mathbf{d}_{rcm}^{\top}\, {}^{e}\mathbf{j}_t}_{=d_{rcm}}\, {}^{e}\mathbf{j}_t, \tag{23} $$

with

$$ d_{rcm} = \frac{\mathbf{d}_{rcm}^{\top}\mathbf{r}_{C_t}}{\|\mathbf{r}_{C_t}\|} \in \mathbb{R}. \tag{24} $$

Furthermore, the curvature $\mathbf{C}_t$ is defined in the same direction as ${}^{e}\mathbf{i}_t$, since

$$ \mathbf{C}_t(s_t) \times \mathbf{r}_{C_t} = {}^{e}\mathbf{k}_t, \tag{25} $$

and

$$ \begin{aligned} \|\mathbf{C}_t(s_t)\| &= \frac{1}{\|\mathbf{r}_{C_t}\|} \\ &= C_t \in \mathbb{R}^+, \end{aligned} \tag{26} $$

thereby,

$$ \mathbf{C}_t(s_t) = C_t \mathbf{i}_t. \tag{27} $$

As a result, the instantaneous curvature is calculated analytically by substituting (21) in the latter equation as

$$ \mathbf{C}_t(s_t) = \frac{\mathbf{r}_{C_t} \times {}^{e}\mathbf{k}_t}{\|\mathbf{r}_{C_t}\|^2}. \tag{28} $$

Let us assume a finite displacement applied to the incision center point $\mathcal{O}_r$ at a velocity ${}^{e}\mathbf{v}_r$ during a small period Δt, as depicted in Fig. 7. The resultant displacement from the linear velocity $({}^{e}\mathbf{v}_r\, \Delta t)$ of the incision frame is decomposed into:

1. A first component

$$ {}^{e}\mathbf{v}_{r_k} = {}^{e}\mathbf{k}_t\, {}^{e}\mathbf{k}_t^{\top}\, {}^{e}\mathbf{v}_r\, \Delta t \tag{29} $$

   which represents the projected component of ${}^{e}\mathbf{v}_r$ onto the vector ${}^{e}\mathbf{k}_t$ and it effects on the progress of $\mathbf{p}_{t'}$ along the tool shape, and
2. A second component

$$ {}^{e}\mathbf{v}_{r_j} = \left(\mathbf{I} - {}^{e}\mathbf{k}_t\, {}^{e}\mathbf{k}_t^{\top}\right) {}^{e}\mathbf{v}_r\, \Delta t \tag{30} $$

   which is the complement of the first component and it acts directly on the distance $\mathbf{d}_{rcm}$.

The first component is used to express analytically the curvilinear speed $\dot{s}_t$ based on *Thales* theorem. The position of the projected point $\mathbf{p}_{t'}(t + \Delta t)$ is found by the intersection of the radial line passing through the vector $\left(\mathcal{O}_{c_t}, \mathcal{O}_r + \Delta t\, {}^{e}\mathbf{k}_t\, {}^{e}\mathbf{k}_t^{\top}\, {}^{e}\mathbf{v}_r\right)$, as shown in Fig. 8. Therefore, the two triangles

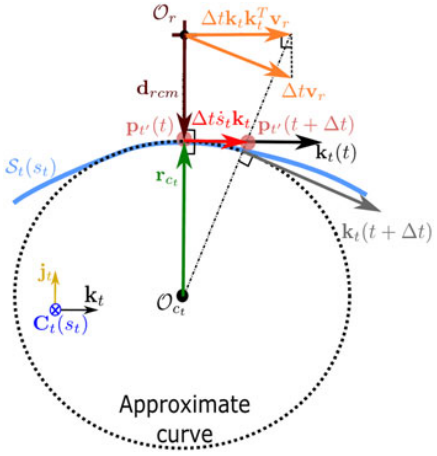


Fig. 8. Zoom on the geometric description of the tool curve and the incision point $\mathcal{O}_r$ at the instantaneous positions where the projected point $\mathbf{p}_{t'}$ is located between the origin of incision point and the center of curvature circle $\mathcal{O}_{c_t}$.

$\left(\mathcal{O}_{c_t}, \mathcal{O}_r, \mathcal{O}_r + \Delta t\, {}^e\mathbf{k}_t\, {}^e\mathbf{k}_t^\top\, {}^e\mathbf{v}_r\right)$ and $\left(\mathcal{O}_{c_t}, \mathbf{p}_{t'}(t), \mathbf{p}_{t'}(t+\Delta t)\right)$ are similar (Fig. 8) under the condition that $\Delta t << 1$. *Thales* theorem states that

$$\frac{\Delta t \dot{s}_t}{\Delta t\, {}^e\mathbf{k}_t^\top\, {}^e\mathbf{v}_r} = \frac{\mathbf{r}_{C_t}^\top\, {}^e\mathbf{j}_t}{{}^e\mathbf{r}_{C_t}^\top \mathbf{j}_t - \mathbf{d}_{rcm}^\top\, {}^e\mathbf{j}_t}. \tag{31}$$

By eliminating Δt from the left-hand side of (31), and by replacing the vectors $\mathbf{r}_{C_t}$ and $\mathbf{d}_{rcm}$ as in (22) and (23), respectively, from the right-hand side, allows writing:

$$\frac{\dot{s}_t}{{}^e\mathbf{k}_t^\top\, {}^e\mathbf{v}_r} = \frac{r_{C_t}\, {}^e\mathbf{j}_t^\top\, {}^e\mathbf{j}_t}{r_{C_t}\, {}^e\mathbf{j}_t^\top\, {}^e\mathbf{j}_t - d_{rcm}\, {}^e\mathbf{j}_t^\top\, {}^e\mathbf{j}_t}. \tag{32}$$

Knowing that ${}^e\mathbf{j}_t^\top\, {}^e\mathbf{j}_t = 1$, then (32) can be simplified as

$$\frac{\dot{s}_t}{{}^e\mathbf{k}_t^\top\, {}^e\mathbf{v}_r} = \frac{r_{C_t}}{r_{C_t} - d_{rcm}}. \tag{33}$$

Thereby, the curvilinear speed $\dot{s}_t$ is deduced by multiplying the right-hand side of the latter equation by the inverse of r_{C_t} as

$$\dot{s}_t = \left(\frac{1}{1 - d_{rcm} C_t}\right) {}^e\mathbf{k}_t^\top\, {}^e\mathbf{v}_r. \tag{34}$$

The term "$d_{rcm}C_t$" in (34) does not consider the direction of the vector $\mathbf{C}_t$, since $C_t = \|\mathbf{C}_t(s_t)\| \in \mathbb{R}^+$. Therefore, the curvature magnitude C_t could be reformulated by using (27) as follows:

$$\begin{aligned} C_t &= \mathbf{C}_t(s_t)^\top\, {}^e\mathbf{i}_t \\ &= \mathbf{C}_t(s_t)^\top ({}^e\mathbf{j}_t \times {}^e\mathbf{k}_t) \\ &= -\left(\mathbf{C}_t(s_t) \times {}^e\mathbf{k}_t\right)^\top\, {}^e\mathbf{j}_t \in \mathbb{R}. \end{aligned} \tag{35}$$

Now, by combining (35) with (24), the term $d_{rcm}C_t$ is reformulated as follows:

$$\begin{aligned} d_{rcm}C_t &= -\mathbf{d}_{rcm}^\top\, {}^e\mathbf{j}_t \left(\mathbf{C}_t(s_t) \times {}^e\mathbf{k}_t\right)^\top\, {}^e\mathbf{j}_t \\ &= -\mathbf{d}_{rcm}^\top \left(\mathbf{C}_t(s_t) \times {}^e\mathbf{k}_t\right) {}^e\mathbf{j}_t^\top\, {}^e\mathbf{j}_t. \end{aligned} \tag{36}$$

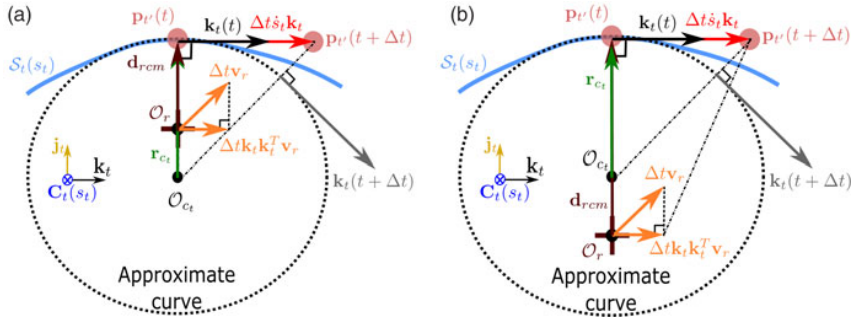


Fig. 9. Zoom on the geometric description of the tool curvature as well as the incision point $\mathcal{O}_r$ at the instantaneous positions where (a) the point $\mathcal{O}_r$ is located between the two points $\mathbf{p}_{t'}$ and $\mathcal{O}_{c_t}$ and (b) the point $\mathcal{O}_{C_t}$ is located within the segment formed between $\mathcal{O}_r$ and $\mathbf{p}_{t'}$.

Considering that ${}^e\mathbf{j}_t^\top\ {}^e\mathbf{j}_t = 1$, then (36) becomes

$$d_{rcm}C_t = -\mathbf{d}_{rcm}^\top\Big(\mathbf{C}_t(s_t)\ \times\ {}^e\mathbf{k}_t\Big). \tag{37}$$

Furthermore, by substituting (37) in (34) allows deducing the speed $\dot{s}_t$ as

$$\dot{s}_t = \frac{{}^e\mathbf{k}_t^\top\ {}^e\mathbf{v}_r}{1 + \mathbf{d}_{rcm}^\top\Big(\mathbf{C}_t(s_t) \times\ {}^e\mathbf{k}_t\Big)}. \tag{38}$$

After the formulation of the curvilinear speed $\dot{s}_t$, the velocity of projected point ${}^e\dot{\mathscr{V}}_{et'}$ is determined by injecting (38) in (18) as

$${}^e\dot{\mathscr{V}}_{et'} = \frac{{}^e\mathbf{k}_t\ {}^e\mathbf{k}_t^\top}{1 + \mathbf{d}_{rcm}^\top\Big(\mathbf{C}_t(s_t) \times\ {}^e\mathbf{k}_t\Big)}\ {}^e\mathbf{v}_r. \tag{39}$$

Finally, the matrix $\mathbf{L}_{et'}$ is expressed by replacing (8) in the latter equation. □

The latter solution is valid if and only if the projected distance $\mathbf{d}_{rcm}$ is smaller than the radius ${}^e\mathbf{r}_{C_t}$ (Fig. 8) because the *Thales* theorem cannot be used directly to formulate similar triangles. As consequence, it is necessary to satisfy the following conditions (see, Fig. 8)

- $\mathbf{p}_{t'} \in [\mathcal{O}_r, \mathcal{O}_{C_t}]$, and $\mathbf{d}_{rcm} \in \mathbb{R}^-$

or (as depicted in Fig. 9a)

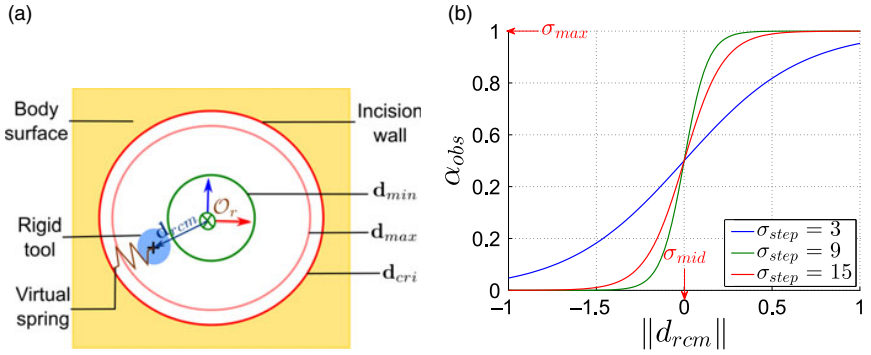
- $\mathcal{O}_r \in [\mathbf{p}_{t'}, \mathcal{O}_{C_t}]$, and $\mathbf{d}_{rcm} \in \mathbb{R}^+$

One can notice that when $\mathbf{d}_{rcm} = \mathbf{r}_{C_t}$, there is a numerical singularity because the right-hand side of (32) is divided by zero. Thereby, the implementation of RCM controller should consider this singular case.

Figure 9b shows another special case which could happen when

- $\mathcal{O}_{C_t} \in [\mathcal{O}_r, \mathbf{p}_{t'}]$ and $\mathbf{d}_{rcm} > \mathbf{r}_{C_t}$.

However, this kind of situation has not a simple geometric construction to find out an analytical expression of the curvilinear velocity $\dot{s}_t$ because the modeling of similar triangles cannot be identified in the aim to apply the *Thales* theorem. Indeed, this case uses a highly curved tool, which is not treated in this work, because it requires a specific numerical solution and also because these types of instruments are uncommon in surgery. For instance, if the vector distance $\mathbf{d}_{rcm}$ is around 1 mm, to satisfy this special case, the curvature radius $\mathbf{r}_{C_t}$ should be less than 1 mm.



The three regions of the incision hole

Sigmoid form of the stiffness function of the virtual spring

Fig. 10. A conceptual scheme of UCM movement.

Control of the error: A conventional proportional controller is applied to reduce exponentially the alignment task error as

$$\begin{aligned} \dot{\mathbf{e}}_{rcm} &= \mathbf{L}_{e_{rcm}} {}^{e}\underline{\mathbf{v}}_{e} \\ &= -\lambda \mathbf{e}_{rcm}. \end{aligned} \tag{40}$$

Thereby, the twist velocity of the end effector is determined as follows:

$${}^{e}\underline{\mathbf{v}}_{e} = -\lambda \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{e}_{rcm} \tag{41}$$

where $\mathbf{L}^{\dagger}_{e_{rcm}}$ is the inverse matrix of $\mathbf{L}_{e_{rcm}}$ and λ is a positive scalar gain to tune the decreasing rate of the angular error between both vectors ${}^{e}\mathbf{u}_{et'}$ and ${}^{e}\mathbf{u}_{er}$. Equation (40) is not only valid for a curved tool, but it is also applicable for a straight tool that can be considered as a special case.

5. Time-Independent Control of a Curved Tool along a Surgical Path under Unilateral Constraints

5.1. Unilaterally constrained motion

An intuitive solution is proposed to formulate the UCM issue by inserting a virtual spring between the tool body and the incision wall. Thereby, the incision diameter is divided into three regions as depicted in Fig. 10(a):

- ⋆ a *free region* around the point $\mathcal{O}_r$ by a minimum distance d_{min}, where the virtual spring is deactivated and the tool is free to move without any constraints;
- ⋆ a *safe region* bounded between the minimum distance d_{min} and a maximum one d_{max}, where the virtual spring is activated and its stiffness σ_{obs} increases gradually (with a step of σ_{step}) when the tool body approaches to d_{max};
- ⋆ a *critical region* where the stiffness of the virtual spring achieves its maximum value when the tool body passes d_{max}.

Control of the error: Admittance control is added to the control law (40) by inserting a virtual spring characterized by a stiffness α_{obs}. It is hence formulated as

$$\mathbf{L}_{e_{rcm}} {}^{e}\underline{\mathbf{v}}_{e} = -\alpha_{obs}\lambda \mathbf{e}_{rcm}. \tag{42}$$

This virtual spring behaves like a repulsive force to keep the tool body away from the incision wall. Its stiffness α_{obs} is chosen as a sigmoid function (Fig. 10(b)) for obtaining a continuous differentiable transition between the three distinct regions. Therefore, the stiffness α_{obs} is chosen as

$$\begin{cases} 0 & \|\mathbf{d}_{rcm}\| \leq d_{min} \\ \dfrac{\sigma_{max}}{1+e^{\left(\sigma_{step}\left(\|\mathbf{d}_{rcm}\|-\sigma_{min}\right)\right)}} & d_{min} < \|\mathbf{d}_{rcm}\| < d_{max} \\ 1 & d_{max} \leq \|\mathbf{d}_{rcm}\| \leq d_{cri} \end{cases}. \tag{43}$$

The latter behavior depends on the projected distance $\mathbf{d}_{rcm}$, which represents the tool body position with respect to the incision center point $\mathcal{O}_r$ (Fig. 10(a)). Finally, the control twist vector of the end effector is calculated as

$${}^{e}\underline{\mathbf{v}}_{e} = -\alpha_{obs}\lambda \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{e}_{rcm}. \tag{44}$$

5.2. Hierarchical assembly of subtasks

The task priority controller arranges the different subtasks depending on the relative pose of the surgical tool with respect to the incision hole. The controller deduces the spatial velocity of the robot end effector. Indeed, the tool motion is mainly divided into two phases. The *outside phase* is the first stage where the tool moves from its initial pose toward the incision hole. During this first phase, it is required that (i) the controller reduces the alignment error between the tool body and the incision hole and (ii) the distance between the tool tip and the incision center point should be regulated toward zero.

The *inside phase* is the second stage where the tool tip begins to follow a reference path under the RCM/UCM constraints. During this second phase, the controller deduces the linear and angular velocities for regulating the alignment error and the path-following error.

The hierarchy between the different subtasks is modified to the criteria of each phase. Note that a hierarchy is achieved by applying the projection gradient technique,[13] which allows projecting a secondary task in the null space of the first task. This projection provides an advantage to find out a solution that satisfies the secondary task without any conflict with the first one.

Outside phase. During this first phase, the alignment task (40) has the highest priority, while the second one is reducing the distance error $\mathbf{e}_{app} = {}^{e}\mathscr{V}_{tr}$ between the tool tip position $\mathcal{O}_t$ and that of the trocar point $\mathcal{O}_r$. This error is regulated by a conventional proportional controller as

$$-\gamma \mathbf{e}_{app} = \underbrace{\begin{pmatrix} \mathbf{I}_{3\times 3} & -[{}^{e}\mathscr{V}_{er}]_{\times} \end{pmatrix}}_{\mathbf{L}_{e_{app}}} \underbrace{\begin{pmatrix} {}^{e}\mathbf{v}_{e} \\ {}^{e}\omega_{e} \end{pmatrix}}_{{}^{e}\underline{\mathbf{v}}_{e}} \tag{45}$$

whereby γ is a positive gain factor that tunes the approach velocity to the trocar point, and $\mathbf{L}_{e_{app}} \in \mathbb{R}^{3\times 6}$ is the interaction matrix of the approach task.

The control twist velocity of the end effector, which satisfies both tasks, is expressed as follows

$${}^{e}\underline{\mathbf{v}}_{e} = -\lambda \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{e}_{rcm} - \gamma \left(\mathbf{I}_{3\times 3} - \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{L}_{e_{rcm}}\right) \mathbf{L}^{\dagger}_{e_{app}} \mathbf{e}_{app}. \tag{46}$$

Inside phase with RCM movement: Secondly, during this phase, the alignment task (40) is defined as the highest priority, while the path-following task (5) has a lower priority. Thus, the control twist velocity of the end effector is given by

$${}^{e}\underline{\mathbf{v}}_{e} = -\lambda \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{e}_{rcm} + \left(\mathbf{I}_{3\times 3} - \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{L}_{e_{rcm}}\right) \mathbf{L}^{\dagger}_{e_{pf}} {}^{e}\mathbf{v}_{t}. \tag{47}$$

Inside phase with UCM movement: During this phase, the hierarchy is defined in terms of the tool position within the incision hole. Therefore, the incision hole is divided into two main regions:

- ⋆ *Tool within the safe region:* When the norm of the projection distance $\|\mathbf{d}_{rcm}\|$ is smaller than the critical distance d_{cri}, the tool is free to move within this region. In this case, the first task is defined as the path following (5) and the lower priority is UCM task (42). Consequently, the end-effector twist vector is calculated as

$${}^{e}\underline{\mathbf{v}}_{e} = \mathbf{L}^{\dagger}_{e_{pf}} {}^{e}\mathbf{v}_{t} - \alpha_{obs}\lambda \left(\mathbf{I}_{3\times 3} - \mathbf{L}^{\dagger}_{e_{pf}} \mathbf{L}_{e_{pf}}\right) \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{e}_{rcm}. \tag{48}$$

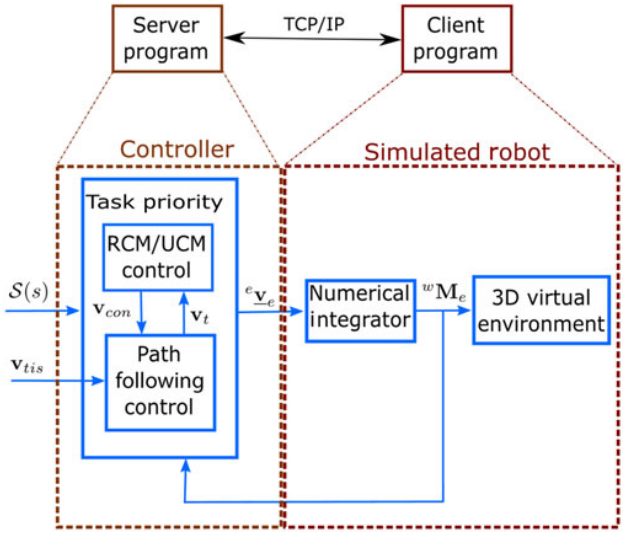


Fig. 11. Block diagram of the socket communication between the proposed controller and the simulated robot.

- *Tool within the dangerous region:* During this situation, the norm of the projection distance $\|\mathbf{d}_{rcm}\|$ is bigger than the critical distance d_{cri}. As consequence, the UCM task becomes the first priority and the path following is considered as the second one. In fact, this solution brings back the tool into the safe region, and the end-effector twist vector is redefined as

$$ {}^e\underline{\mathbf{v}}_e = -\alpha_{obs}\lambda \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{e}_{rcm} + \left(\mathbf{I}_{3\times 3} - \mathbf{L}^{\dagger}_{e_{rcm}} \mathbf{L}_{e_{rcm}}\right) \mathbf{L}^{\dagger}_{e_{pf}} \, {}^e\mathbf{v}_t. \tag{49} $$

- If the tool passes the dangerous region for some reason, the controller switches to RCM movement (47) in order to bring back the tool as quickly as possible toward the safe region.

6. Numerical Validation

A numerical simulator was developed as the first step to validate the functioning of the different methods before physical implementation. Therefore, the control architect was implemented to conceive the controller independent from the physical system (robot). Socket communication is implemented to establish the communication between the controller and the virtual robot, as depicted in Fig. 11. The next section will replace the simulated robot with a real robot.

The objective of the simulated tests discussed below is to demonstrate the influence of the model parameters (i.e., λ for the alignment task (40), γ for the approach task (45), and v_{tis} and β for the path-following controller (3)).

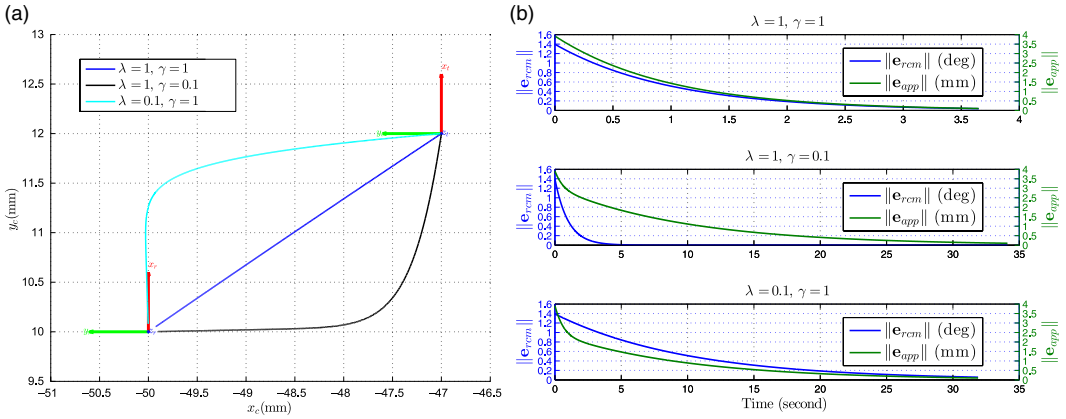
6.1. Outside phase: Control gain choice influence

A first test shows the influence of λ and γ for the first (40) and the second (45) tasks during the outside phase (46), as depicted in Fig. 12.

Throughout this first test, a straight tool was used during the various numerical trials where its tool tip frame $\Re_t$ is located at the top right of Fig. 12(a), while the incision frame $\Re_r$ is located at the bottom left.

The blue line, in Fig 12(a), represents the system behavior when $\lambda = 1$ and $\gamma = 1$. The system tends to reduce the alignment task error (40) faster than reducing the approach task error (45), if the value of λ is greater than γ (black line in Fig. 12(a)). In the opposite case, when γ is greater than λ, the approach task converges faster than the alignment task (cyan line in Fig. 12(a)).

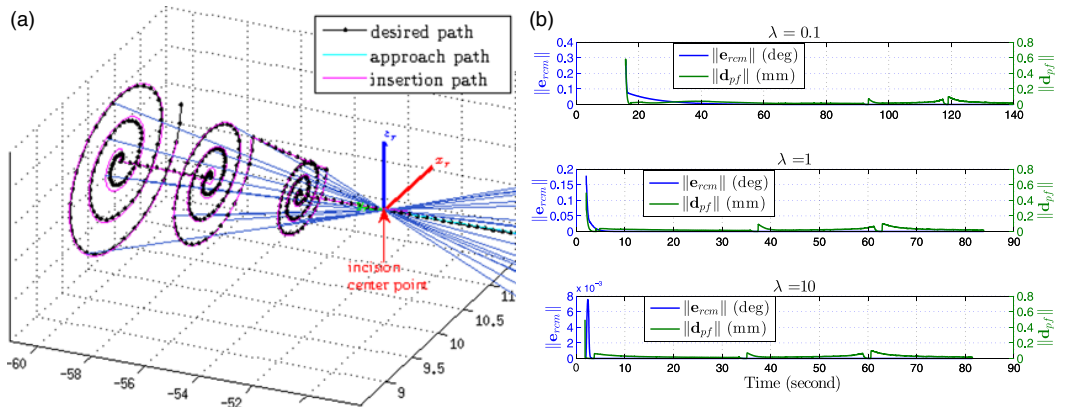
The evolution of the alignment task error and that of the approaching task are depicted in Fig. 12(b). It can be highlighted that for a faster exponential decay of these errors, one can increase the gain values.



Tool tip motion where the upper right corner is the initial position of the tool tip $\Re_t$ and the lower left corner is the incision hole center point $\Re_r$.

Output errors of the alignment task $\mathbf{e}_{rcm}$ and that of the approach task $\mathbf{e}_{app}$.

Fig. 12. The system performances during the outside phase while varying the values of λ and γ.



Zoom on the reference path and the actual path performed by the tool.

The angular RCM task error $\mathbf{e}_{rcm}$ with the path following task error $\mathbf{d}_{pf}$ during the inside phase

Fig. 13. Validation of the path-following task under the RCM constraint.

6.2. *Inside phase: 3D path following under bilateral (RCM) constraints*

This test simulates a driller (straight) tool in order to form a conical tunnel (i.e., incision hole). The 3D path is defined with respect to the incision frame $\Re_r$ as depicted in Fig. 13(a), and it is composed of (i) a linear portion along the basis ${}^r\mathbf{y}$ and its length is 13 mm and (ii) various steps of helical paths.

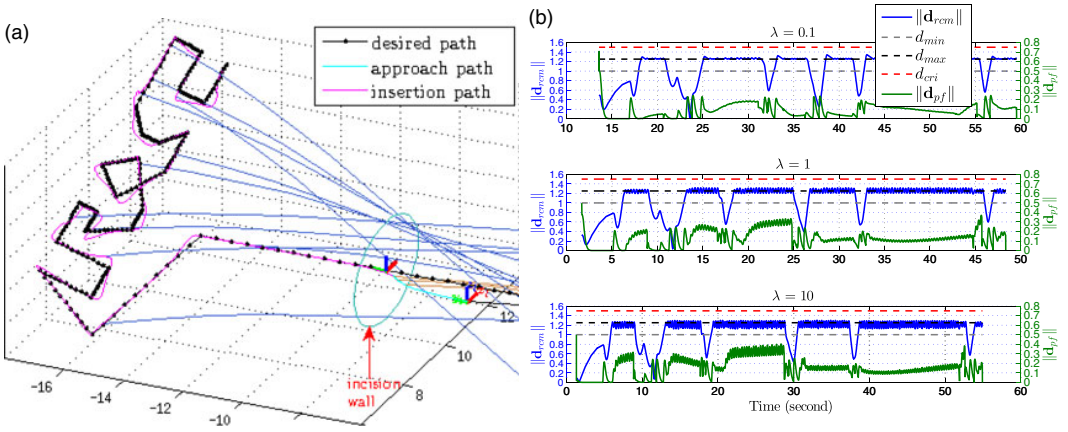
Throughout this phase, the task priority controller (47) arranges the alignment task as the highest priority, while the second one is the path following ensured the incision task. Thereby, this test focuses on the influence of the alignment gain λ, while the other gain values are constant, that is, $v_{tis} = 2$ mm/s, $\beta = -3$, and $T_e = 0.01$ *second* (sampling period).

The system performances are shown in Fig. 13(b), where λ increases from 0.1, 1, to 10, for the upper, middle, and lower plots, respectively. It can be highlighted that in the upper plot, the angular alignment error $\mathbf{e}_{rcm}$ is reduced exponentially as expected but it takes time to reduce and maintain it at zero. However, the middle and lower plots show that the error $\mathbf{e}_{rcm}$ reaches the zero quickly.

Besides that, one can observe that the task priority controller works well since the secondary (path following) task error does not influence that of the first (alignment) one. The summary of the angular error $\mathbf{e}_{rcm}$ and that of the path following $\mathbf{d}_{pf}$ during the different trials are presented in Table II, which summarizes the median, the mean, and the standard deviation (STD) errors of both tasks (i.e., $\mathbf{e}_{rcm}$ and $\mathbf{d}_{pf}$).

Table II. Summary of different tests achieved with constant gain values (i.e., $v_{tis} = 2$ *mm/second* and $\beta = -3$) while the gain λ varies from 0.1 to 10.

	RCM angular error (deg)			PF error (mm)		
λ	**Median**	**Mean**	**STD**	**Median**	**Mean**	**STD**
0.1	0	0.007	0.018	0.022	0.027	0.023
1	0	0.001	0.011	0.022	0.027	0.02
10	0	0	0.001	0.022	0.027	0.016



Zoom on the reference path and the actual path performed by the tool.

The UCM lateral error $\mathbf{d}_{rcm}$ with the path following task error $\mathbf{d}_{pf}$ during the inside phase while changing the value of λ

Fig. 14. Validation of the path-following task under the UCM constraint.

6.3. Inside phase: 3D path following under the unilateral (UCM) constraints

This test validates the use of a curved tool (with circular portion shape) instead of the straight one. The achieved task consists of the following of a reference incision path under the UCM constraints. The reference path is defined by a linear portion and the word $\mu RoCS$, as depicted in Fig. 14(a). This figure also shows the incision wall represented by a dark green circle.

During this phase, the task priority controller chooses between (48) or (49) depending on the tool position with respect to the incision wall. Therefore, this simulated test shows the influence of alignment gain λ on the system performances (Fig. 14(b)), while the other gain values are constant (i.e., $v_{tis} = 2$ mm/s, $\beta = -3$, and $T_e = 0.01$ s). When the weighting factor λ is small, the lateral alignment error $\mathbf{d}_{rcm}$ stays close to the border d_{max} with some overtaking (first plot in Fig. 14(b)). The values of d_{min}, d_{max}, and d_{cri} are constant during the different trials and then equal 1, 1.25, and 1.5 mm, respectively. However, if the value of λ is too big, the controller will provide high-velocity values. Consequently, the lateral alignment error $\mathbf{d}_{rcm}$ slightly oscillates around the border d_{max} (third plot in Fig. 14(b)) but the system still stable.

Besides that, the reference path of this test is discontinuous. It is generally difficult to follow accurately a sharp path. Thus, the path-following error $\mathbf{d}_{pf}$ is bigger compared to the previous test. However, the error $\mathbf{d}_{pf}$ can be reduced by smoothing the sharp corners of the reference path (make them curvy) and/or modifying the path-following gains (v_{tis}/β).

Table III summaries the lateral alignment error $\mathbf{d}_{rcm}$ and that of the path following $\mathbf{d}_{pf}$ during the different trials.

7. Experimental Validation

This section deals with experimental validation of the different concepts and control methods proposed in this paper. This experimental validation was performed using a laboratory test bench as depicted in Fig. 15, which is composed, among other items, of a 3PPSR parallel robotic system of six degrees of freedom (dof). Each dof is actuated thanks to a high-resolution DC motor and

Table III. Summary of different tests achieved with constant gain values (i.e., $v_{tis} = 2$ mm/s and $\beta = -3$), while the gain λ varies from 0.1 to 10.

	UCM lateral error (mm)			PF error (mm)		
λ	Median	Mean	STD	Median	Mean	STD
0.1	1.254	1.08	0.328	0.082	0.089	0.023
1	1.22	1.077	0.3	0.12	0.133	0.084
10	1.175	1.064	0.289	0.148	0.16	0.097

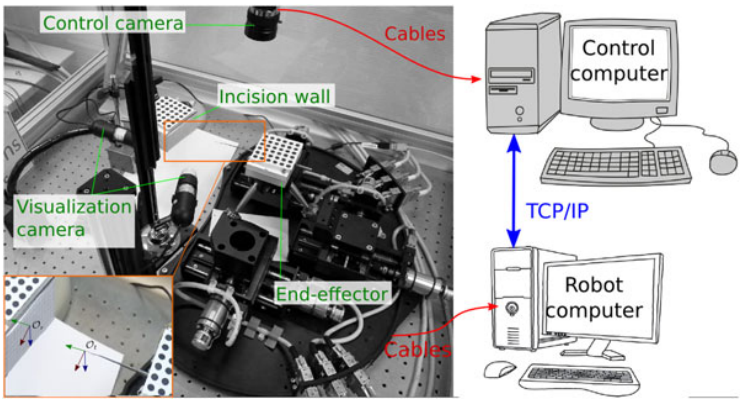


Fig. 15. Experimental setup configuration with a zoom on the tool tip and the incision wall.

high-accurate individual encoders. The robotic system is characterized with the following features[a] : translation ranges $(t_x, t_y, t_z)^\top_{max} = (50, 100, 12.7 \text{ mm})^\top$, rotation ranges $(r_x, r_y, r_z)^\top_{max} = (10°, 10°, 10°)^\top$, a linear resolution of 0.2 μm (repeatability of ± 0.5 μm), and an angular resolution of 0.0005° (repeatability of $\pm 0.0011°$). The low level of robot control (i.e., inner PID loop, static, and differential kinematic models) is done on a programmable logic controller, which communicates with a computer (a 2.33-GHz Xeon Intel CPU with a Windows distribution). Furthermore, the high-level control of the robot (i.e., task priority, both RCM and UCM, as well as the path-following controllers) is implemented on another computer (a 3.20-GHz i5 core Intel CPU with a Linux distribution), which sends the control velocities to the robot computer via TCP/IP protocol. The high-level computer is also used to communicate with a monocular camera for tracking and estimating the end-effector pose in real time (at the camera frame rate, i.e., 20 Hz).

Furthermore, the robot end effector carries a standard surgical tool (Fig. 16) that can be found actually in the operation room. It could be either a straight instrument (its diameter varies from 2.4 mm at its base and 0.5 mm at its tip) or a curved one (formed by a portion of a circle where its radius is around 16 mm and its diameter is 1 mm). Besides that, the incision wall is represented by a sheet of paper which acts as a fragile membrane. It is used to demonstrate that the tool does not tear the paper while following the desired path under the RCM or UCM tasks.

7.1. Implementation details

The proposed controller was implemented in C++ by using ViSP (Visual Servoing Platform) library.[27] The code is divided into C++ classes which are summarized in Algorithm 1.

The user (e.g., a surgeon and/or operator) starts by defining the reference path before the control loop begins (offline). The geometric curve is generally determined during the per-operative phase where a planning software deduces the optimal path[28–30] or the surgeon draws the curve on a tactile tablet [b].[31–33]

[a]The data sheet of the PI parallel robot *SpaceFAB SF-3000 BS* is available online https://www.physikinstrumente.com/en/products/parallel-kinematic-hexapods/hexapods-with-motor-screw-drives/sf-3000-bs-spacefab-1204400/

[b]μRALP (Micro-technologies and Systems for Robot-Assisted Laser Phonomicrosurgery). [online] http://www.microralp.eu/

Algorithm 1 Summarizing the controller code.

Result: robot velocity twist vector (${}^e\underline{\mathbf{v}}_e$)
parameter: initialize model parameters (λ, γ, v_{tis}, β)
Data: reference path $\left(\mathcal{S}_p(s_p)\right)$

$({}^e\mathbf{M}_t, {}^c\mathbf{M}_w, {}^c\mathbf{M}_r, {}^c\mathbf{M}_e) \longleftarrow$ initialize homogeneous matrices (image)
while *(! task accomplished)* **do**
  if PATH FOLLOWED == 1 *or* INTER == MAX **then**
    task accomplished
  else
    $({}^c\mathbf{M}_e, {}^c\mathbf{M}_r) \longleftarrow$ features tracking (image)
    $(\mathbf{e}_{rcm}, \mathbf{d}_{rcm}) \longleftarrow$ compute RCM/UCM variables $({}^e\mathbf{u}_{er}, {}^e\mathbf{u}_{et}, {}^e\mathbf{et})$
    $\mathbf{L}_{e_{rcm}} \longleftarrow$ alignment task (λ)
    if CHECK PHASE **then**
      outside phase
      $\mathbf{L}_{e_{app}} \longleftarrow$ point regulation (γ)
    else
      path following task
      $\mathbf{d}_{pf} \longleftarrow$ tool tip projection on the path $({}^c\mathbf{M}_e, {}^e\mathbf{M}_t, \mathcal{S}_p(s_p))$
      $(\mathbf{L}_{\mathbf{e}_{pf}}, \mathbf{v}_t) \longleftarrow$ compute tool tip linear velocity (v_{tis}, β)
      if END OF PATH **then**
        PATH PERFORMED $\longleftarrow$ 1
      end
    end
    arrange task priority
    ${}^e\underline{\mathbf{v}}_e \longleftarrow$ **projection of the 2nd task into the null-space of the 1st task**
    send the velocity to the robot $({}^e\underline{\mathbf{v}}_e)$
  end
end

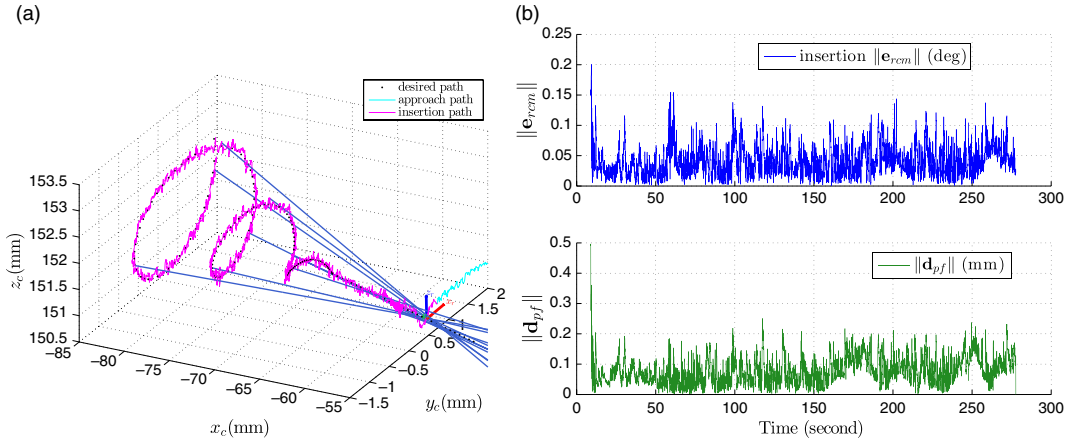
Fig. 16. The different surgical tools used during the experimental validation.

Afterward, the user initializes the parameters of each controller depending on the task to be performed. Thus, the control loop starts by updating the different homogeneous matrices representing the pose (position and rotation) of the various frames. Thereafter, the control velocity for maintaining the RCM/UCM task is then computed (40) or (42). Then, the control velocity of the second task is computed to converge toward the trocar point (45) or to follow the desired path (3).

Finally, the task priority controller manages the hierarchical priority between tasks as depicted in Section 5.2.

7.2. Results: 3D path following under bilateral (RCM) constraints

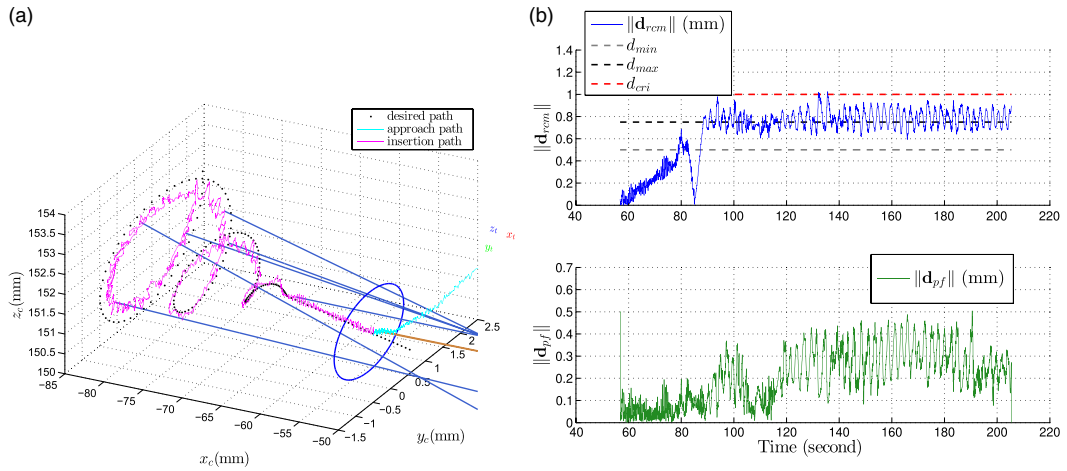
This test evaluates the proposed controller while using a straight tool for following a spiral path. Fig. 17(a) shows the general motion of the tool tip during the inside phase, while Fig. 17(b) presents



The path performed by a straight tool under the RCM constraints

The alignment error $\mathbf{e}_{rcm}$ and that of the path following $\mathbf{d}_{pf}$

Fig. 17. The results obtained from the parallel robot, during the inside phase, while using a straight tool under RCM constraints.



The path performed by a straight tool under the UCM constraints

The alignment error $\mathbf{e}_{rcm}$ and that of the path following $\mathbf{d}_{pf}$

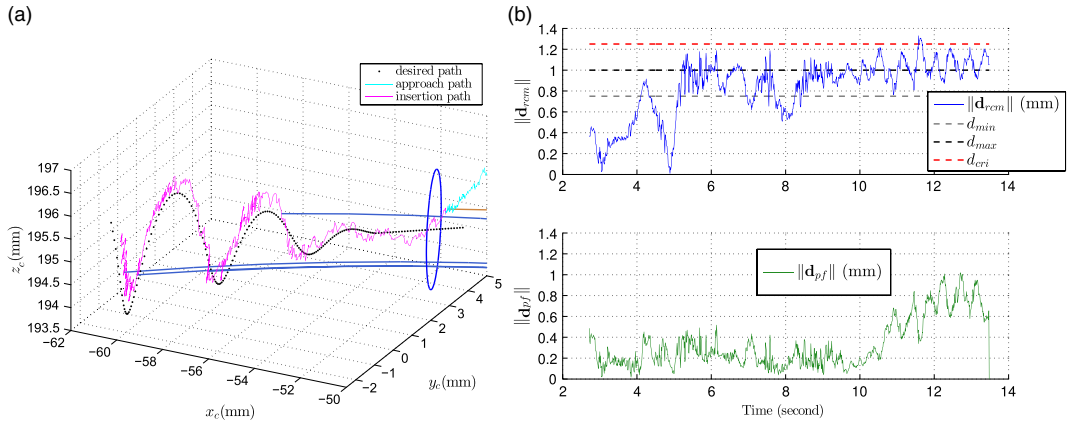
Fig. 18. The results obtained from the parallel robot while using a straight tool.

the system performances. As expected, the alignment error $\mathbf{e}_{rcm}$ maintains its value around zero, as depicted in the upper plot in Fig. 17(b). The error $\mathbf{e}_{rcm}$ was measured during the inside phase to be $0.04° \pm 0.02°$ (i.e., mean error $\pm$ STD error) and its median error was 0.035°.

The lower plot of Fig. 17(b) shows the evolution of the path following $\mathbf{d}_{pf}$. It was measured as 0.08 ± 0.05 mm, and its median error was 0.068 mm during the inside phase. The gain values used during this validation tests are fixed as follows: $\lambda = 0.8$, $\gamma = 1$, $v_{tis} = 4$ mm/s, $\beta = -8$, and $T_e = 0.05$ s.

7.3. Results: 3D path following under unilateral (UCM) constraints

Straight tool. This test deals with the validation of the proposed controller under a UCM constraint while using a straight tool. The reference path was chosen as a spiral curve with its maximum radius reaches 2 mm. Figure 18(a) shows the tool motion through the incision hole, where the boundary d_{cri} is represented by circle. Also, Fig. 18(b) shows the evolution of the positioning errors during the tool motion. As expected, the lateral alignment error $\mathbf{d}_{rcm}$ was limited between the boundaries $d_{min} = 0.5$, $d_{max} = 0.75$ and $d_{cri} = 1$ mm. Indeed, the $\mathbf{d}_{rcm}$ error was measured during the inside phase



The path performed by a curved tool under the UCM constraints

The alignment error $\mathbf{e}_{rcm}$ and that of the path following $\mathbf{d}_{pf}$

Fig. 19. The results obtained from the parallel robot while using a curved tool.

to be $0.67 \pm 0.23mm$ and its median value as 0.73 mm. Besides that, the path-following error $\mathbf{d}_{pf}$ was measured to be 0.208 ± 0.12 mm and its corresponding median value 0.207 mm. In fact, the lateral alignment error is coupled with the path following, as shown in Fig. 18(b) near the time 120 s, the slight oscillations of $\mathbf{d}_{rcm}$ error around the boundary d_{max} cause the $\mathbf{d}_{pf}$ to oscillate also. These results are produced with the following control gains: $\lambda = 0.8$, $v_{tis} = 2$ mm/s, $\beta = -4$, and $T_e = 0.05$ s.

Curved tool: In the second test, the straight tool was replaced by a curved one and the output results are presented in Fig. 19(a). The reference path is the same as the previous test, but it is rotated about the z-axis. Fig. 19(b) shows the system performances during the inside phase. The lateral alignment error $\mathbf{d}_{rcm}$ was measured as 0.83 ± 0.28 mm and its median value was 0.92 mm. The $\mathbf{d}_{rcm}$ error is bigger compared to the previous test due to the gain $\lambda = 0.5$ is smaller than the previous test. A bigger value of λ generates poses that are not reachable with the parallel robot since the robot reaches its workspace limit. Moreover, the path-following error $\mathbf{d}_{pf}$ was measured to be 0.33 ± 0.24 mm and its median value was 0.24 mm. These results are produced with the following control gains values: $v_{tis} = 2$ mm/s, $\beta = -3$, and $T_e = 0.01$ s.

8. Conclusion and Perspectives

This article demonstrated the proof of concept of a control scheme for constrained motions as RCM and UCM while using either straight-line or fixed-curve surgical tools. The proposed methods offer a generic formulation of the constrained motion problem with high flexibility, which means easy/intuitive integration in various systems or purposes since RCM and UCM constraints were described in the task space. In fact, this methodology does not require precise knowledge of the robot inverse kinematics.

Another "surgical" task was included in the problem formulation as a path-following scheme mimicking excision/ablation clinical interventions. Indeed, the surgeon can draw a predefined excision path using preoperative images that the tool must achieve under the constrained motion (RCM or UCM). A task-prioritizing paradigm was developed to manage the performing of two or more tasks without conflicts. Thereafter, the proposed methods were successfully validated (under various scenarios) both numerically using a developed simulator and experimentally using a 6-DoF robotic setup.

Finally, it would be valuable for the perspectives to investigate what is the optimal geometric form of the curved tool in order to achieve a maximum displacement while minimizing the collision risk with anatomical structures. Besides that, the proposed controller will be evaluated in *in vitro* and *ex vivo* experiments. Furthermore, the controller can be extended to include the robot dynamics (i.e., a force controller). Integrating a flexible tool will as well increase the dexterity and the maneuverability of the distal tool tip.

References

1. T. Osa, C. Staub and A. Knoll, "Framework of Automatic Robot Surgery System Using Visual Servoing," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010) pp. 1837–1842.
2. M. M. Dalvand and B. Shirinzadeh, "Remote Centre-of-Motion Control Algorithms of 6-RRCRR Parallel Robot Assisted Surgery System (PRAMiSS)," **In:** *IEEE International Conference on Robotics and Automation* (2012) pp. 3401–3406.
3. I. Fleming, M. Balicki, J. Koo, et al., "Cooperative Robot Assistant for Retinal Microsurgery," **In:** *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2008) pp. 543–550.
4. Y. Ida, N. Sugita, T. Ueta, et al., "Microsurgical robotic system for vitreoretinal surgery," *Int. J. Comput. Assist. Radiol. Surgery* **7**(1), 27–34 (2012).
5. R. H. Taylor, J. Funda, D. D. Grossman, et al.,"Remote Center-of-Motion Robot for Surgery," (1995), US Patent 5,397,323.
6. B. Dahroug, B. Tamadazte and N. Andreff, "Task Controller for Performing Remote Centre of Motion," **In:** *Lecture Note on Electrical Engineering*. Springer (2017).
7. A. Blumentals, B. Brogliato and F. Bertails-Descoubes, "The contact problem in Lagrangian systems subject to bilateral and unilateral constraints, with or without sliding Coulombs friction: A tutorial," *Multibody Syst. Dyn*. **38**(1), 43–76 (2016).
8. P. J. Swaney, J. M. Croom, J. Burgner, et al., "Design of a Quadramanual Robot for Single-Nostril Skull Base Surgery," **In:** *ASME Annual Dynamic Systems and Control Conference*, vol. 3 (American Society of Mechanical Engineers, 2012) pp. 387–393.
9. C. Girerd, K. Rabenorosoa and P. Renaud, "Combining Tube Design and Simple Kinematic Strategy for Follow-the-Leader Deployment of Concentric Tube Robots," **In:** *Advances in Robot Kinematics* (2016) pp. 23–31.
10. M. Miroir, Y. Nguyen, J. Szewczyk, et al., "Robotol: From design to evaluation of a robot for middle ear surgery," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 850–856.
11. K. Entsfellner, R. Tauber, D. B. Roppenecker, et al., "Development of universal gripping adapters: Sterile coupling of medical devices and robots using robotic fingers," **In:** *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2013) pp. 1464–1469.
12. B. Dahroug, B. Tamadazte, B. Weber, et al. "Review on otological robotic systems: Toward microrobot-assisted cholesteatoma surgery," *IEEE Rev. Biomed. Eng.* **11**, 125–142 (2018).
13. Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.* **6**(2), 3–15 (1987).
14. B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intel. Robot. Syst.* **3**, 201–212 (1990).
15. N. Mansard and O. Khatib, "Continuous Control Law from Unilateral Constraints," *IEEE International Conference on Robotics and Automation* (2008) pp. 3359–3364.
16. C.-H. Kuo, J. S. Dai and P. Dasgupta, "Kinematic design considerations for minimally invasive surgical robots: An overview," *Int. J. Med. Robot. Comput. Assist. Surg.* **8**(2), 127–145 (2012).
17. J. Funda, R. H. Taylor, B. Eldridge, et al., "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Trans. Robot. Autom.* **12**, 453–465 (1996).
18. H. Azimian, R. V. Patel, and M. D. Naish, "On Constrained Manipulation in Robotics-Assisted Minimally Invasive Surgery," **In:** *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics* (2010) pp. 650–655.
19. M. M. Marinho, M. C. Bernardes and A. P. Bó, "A Programmable Remote Center-of-Motion Controller for Minimally Invasive Surgery Using the Dual Quaternion Framework," **In:** *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics* (2014) pp. 339–344.
20. C. D. Pham, F. Coutinho, A. C. Leite, et al., "Analysis of a Moving Remote Center of Motion for Robotics-Assisted Minimally Invasive Surgery," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2015) pp. 1440–1446.
21. H. Mayer, I. Nagy and A. Knoll, "Kinematics and Modelling of a System for Robotic Surgery," **In:** *Advances in Robot Kinematics* (2004), pp. 181–190.
22. E. M. Boctor, R. J. Webster III, H. Mathieu, et al., "Virtual remote center of motion control for needle placement robots," *Comput. Aid. Surg.* **9**, 175–183 (2004).
23. B. Dahroug, J. A. Seon, A. Oulmas, T. Xu, B. Tamadazte, et al., "Some Examples of Path Following in Microrobotics," **In:** *IEEE International Conference on Manipulation, Automation and Robotics at Small Scales* (2018) pp. 1–6.
24. L. B. Rosenberg, "Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation," **In:** *IEEE Virtual Reality Annual International Symposium* (1993) pp. 76–82.
25. S. A. Bowyer, B. L. Davies and F. R. y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Trans. Robot.* **30**(1), 138–157 (2014).
26. B. Dahroug, B. Tamadazte and N. Andreff, "Visual Servoing Controller for Time-Invariant 3D Path Following with Remote Centre of Motion Constraint," **In:** *IEEE International Conference on Robotics and Automation* (2017) pp. 3612–3618.
27. E. Marchand, F. Spindler and F. Chaumette, "Visp for Visual Servoing: A Generic Software Platform with a Wide Class of Robot Control Skills," *IEEE Robot. Autom. Mag.* **12**, 40–52 (2005).

28. M. Kazemi, K. Gupta and M. Mehrandezh, "Path-Planning for Visual Servoing: A Review and Issues," **In:** *Visual Servoing via Advanced Numerical Methods*. Springer, 2010, pp. 189–207.
29. N. Gerber, B. Bell, K. Gavaghan, et al., "Surgical planning tool for robotically assisted hearing aid implantation," *Int. J. Comput. Assist. Radio. Surg.* **9**(1), 11–20 (2014).
30. A. Gasparetto, P. Boscariol, A. Lanzutti and R. Vidoni, "Path Planning and Trajectory Planning Algorithms: A General Overview," **In:** *Motion and Operation Planning of Robotic Systems*. Springer (2015), 3–27.
31. J. A. Seon, B. Tamadazte and N. Andreff, "Decoupling path following and velocity profile in vision-guided laser steering," *IEEE Trans. Robot.* **31**(2), 280–289 (2015).
32. R. Renevier, B. Tamadazte, K. Rabenorosoa, et al., "Endoscopic laser surgery: Design, modeling, and control," *IEEE/ASME Trans. Mechatron.* **22**(1), 99–106 (2017).
33. B. Tamadazte, R. Renevier, J. Séon, et al., "Laser beam steering along three-dimensional paths," *IEEE/ASME Trans. Mechatron.* **23**(3), 1148–1158 (2018).