3D Path Following with Remote Center of Motion Constraints

Bassem Dahroug, Brahim Tamadazte and Nicolas Andreff
FEMTO-ST Institute, AS2M department, Univ. Bourgogne Franche-Comté/CNRS/ENSMM,
4 Rue Alain Savary, 25000 Besançon, France
firstname.lastname@femto-st.fr

Keywords: Bilateral Remote Center of Motion Constraints, 3D Path Following, Medical Robotics.

Abstract: The remote center of motion (RCM) is an essential issue during minimal invasive surgery where the surgeon manipulates a medical instrument inside the human body. It is important to assure that the tool should not apply forces on the incision wall in order to prevent patient harm. The paper shows a geometric method computing the intended robot velocity vector for respecting the RCM constraints. In addition, the proposed solution deals with the latter constraints as the highest task priority. A second task function is added, which is projected in the null space on the first task, to follow a 3D path inside the cavity. As result, this method helps the surgeon to execute more sophisticated motion within the patient body with high accuracy; since the results shows standard deviation around $0.004mm$ and $0.089mm$ of RCM task error and positioning task error, respectively.

1 INTRODUCTION

The surgical assisted-robotics have been getting more demands over the last years as they help by providing ergonomic conditions for increasing accuracy and reducing fatigue. Moreover, the patients benefit from a reduced invasion, time and costs. The assistance will help the surgeon in performing more complex motion inside the patient's body, getting over the physical constraints and navigating in unknown environment. In fact, the robot not only needs information about its internal state, which is the pose of its end-effector with respect to its base, but it is also required information regarding the relative pose of the organs.

During the navigation phase, visual servoing control approach (Azizian et al., 2014) allows to mimic the perception sense for the surgeon. This approach uses real-time imaging (e.g., endoscope, optical coherence tomography or ultrasound) to detect, track and guide the instrument (Krupa et al., 2002) (Duflot et al., 2016). The navigation software may include other advanced options such as virtual- and augmented- reality to enhance the visualization and guidance process. But the essential control task is guiding the instrument motion for following a desired geometric path or trajectory. The difference between these two latter notations is the solution convergence. In case the tool is retarded to reach the scheduled point due to any throughout the previous point. On one hand, the trajectory following controller tends to accelerates its velocity and to shortcut the desired trajectory, especially when it is defined with acute curvature, in order to reduce the time delay. On the other hand, path following controller maintains its motion along the geometric path with the intended velocity profile even in lag conditions. The latter controller is useful for medical applications, especially during ablation process. Since path following controller guarantees independent instrument velocity from the geometric path and depend on the interaction between the ablation tool and the tissue type. Path following is widely used for mobile robot but it is not frequently applied into medical application. A 2D path following proposed (Seon et al., 2015) for laser surgery. They applied non-holonomic control for executing a unicycle path following with high frequency. A 3D trajectory following and pose estimation methods (Nageotte et al., 2006) proposed for controlling an instrument to perform automatic suturing during laparoscopic surgery. In general, surgical assisted-robots help the surgeon to perform more complex gestures and become less invasive.

Minimal invasive robotic systems enter into the human body from a small incision which presents physical constraints on the surgical tool motion. These constraints are created by the incision wall which reduces the tool degrees of freedom (DOF) to four DOF (three rotations and one translation). The resultant motion from these constraints is called *remote center of motion* (RCM), *trocar constraints*, *bi-*

lateral constraints or *fulcrum effect*. This type of motion could be achieved either with specific kinematic robot structure (Kuo et al., 2012) or with software control (Dalvand and Shirinzadeh, 2012). This article is focusing on the software control type because it is a generic method that could be applied regardless of the robot structure, in condition that the robot DOF should be greater than 4-DOF. This condition ensures that the robot kinematic structure is redundant. The redundancy occurs when the manipulator joints number (i.e., its DOF) is greater than those required to execute a desired task. Such task could be any kinematic or dynamic goal. The advantage of redundancy is increasing the robot manoeuvrability and dexterity that could be useful to avoid singularity, joints limits, workspace obstacles and it provides the concept of task priority (Nakamura et al., 1987).

For software RCM resolution, there are different reported methods in the literature: extended Jacobian with quadratic optimization (Funda et al., 1996), artificial intelligence based heuristic search (Boctor et al., 2004), analytical solution based on trocar modelling with Euler angle representation (Mayer et al., 2004), isotropy-based kinematic optimization (Locke and Patel, 2007), gradient projection approach in closed-loop form (Azimian et al., 2010), dual quaternion-based kinematic controller (Marinho et al., 2014), and constrained Jacobian represented with Lie algebra (Pham et al., 2015). For solving RCM with a visual servoing scheme, the reported methods are: geometric constraint with stereo visual servoing for controlling the robot position from point-to-point (Osa et al., 2010), and extended Jacobian solution for manipulating serial end-effector (Aghakhani et al., 2013). These presented techniques are used to maintain the task function of fulcrum effect and additional tasks may be added to extend the robot functionality.

As far as we know, the problem of RCM constraints combined with the 3D path following has not been properly addressed yet.The previous researches discussed the modelling of trocar kinematics only or combined with trajectory following. However, the main new add value of this article is formulating a new method to maintain bilateral constraints while following a 3D pre-defined path. The method controls the motion of rigid tool with visual feedback and it describes the bilateral constraints in vector form with task hierarchy, as shown in Section 2. The control laws are tested in simulator and the results are presented in Section 3.

2 CONTROL DESIGN

The proposed controller commands the robot velocity for performing 3D path following with bilateral constraints. It achieves the objective with two task errors: (i) the first prior task is the alignment of the tool with incision point, and (ii) the second task error is the position of tool tip with respect to the required path. It is also has two operation modes to accomplish a desired 3D path: (i) approaching phase where the tool aligns itself with the trocar point, and (ii) insertion phase where the trocar point should be located along the tool.

2.1 Notation

The notations used within the paper are summarized in Table 1, for a better understanding.

2.2 Remote Center of Motion Constraints

2.2.1 Problem Statement

On one hand, the tool is free to move when it is outside the incision point. On the other hand, the tool movement is restricted when it passes the incision hole. During the latter motion, the RCM constraints allow the tool translation along the y-component of the current RCM frame (${}^{r}\mathbf{v}$) and angular rotation (${}^{r}\boldsymbol{\omega}$) around the latter frame axes. The y-component of RCM frame (${}^{r}\mathbf{y}$) is assumed to be perpendicular to the tissue surface (Figure 1). The tool tip velocity with respect to the medical imaging system (${}^{c}\mathbf{v}_{t}$) is determined by the position-based path following control (see Section 2.3). Therefore, the problem becomes to achieved this motion by applying the adequate end-effector velocity (${}^{r}\mathbf{v}$ and ${}^{r}\boldsymbol{\omega}$) while maintaining the RCM constraints.

In (Boctor et al., 2004), two heuristic functions were used to define the RCM constraints. The first one is the distance ($\mathbf{e}_1 = T - P$) between the tool tip (T) and the target point (P) inside the cavity. The second function is the cross-product ($\mathbf{e}_2 = \mathbf{ET} \wedge \mathbf{RP}$) between the rigid tool vector ($\mathbf{ET}$) and the vector from RCM to the target point ($\mathbf{RP}$). The weakness of this method is not arranging the heuristic functions in task priority mode. Therefore, the system could converge to a solution to satisfy the one function without respecting the other one (i.e., $(\mathbf{e}_1, \mathbf{e}_2) = (\mathbf{e}_1 \neq \mathbf{0}, \mathbf{0})$ or $(\mathbf{e}_1, \mathbf{e}_2) = (\mathbf{0}, \mathbf{e}_2 \neq \mathbf{0})$).

Table 1: symbols summary.

Symbol	Description
$\Im_W$	world frame with the origin point W
$\Im_B$	robot base frame with the origin point B
$\Im_E$	end-effector frame with the origin point E
$\Im_T$	tool tip frame with the origin point T
$\Im_R$	RCM frame with the origin point R
$\Im_C$	camera frame with the origin point C
${}^{w}M_e$	homogeneous transformation matrix that describes the pose of $\Im_E$ in $\Im_W$
${}^{c}\mathbf{v}_e$	linear velocity of $\Im_E$ that is expressed in $\Im_C$
${}^{c}\omega_e$	angular velocity of $\Im_E$ that is expressed in $\Im_C$
${}^{c}\tau_e$	velocity vector of $\Im_E$ that groups its linear and angular velocities
$\mathbf{I}_{3\times3}$	identity matrix
${}^{r}\mathbf{y}$	the y-component of $\Im_R$
${}^{r}\mathbf{v}$	linear velocity of any point (subscript) that is expressed in $\Im_R$
${}^{c}\tau_t$	linear velocity of $\Im_T$ that is expressed in $\Im_C$
${}^{e}\mathbf{ER}$	vector between the origin points of $\Im_E$ and $\Im_R$, expressed in $\Im_E$
${}^{e}\mathbf{u}_{er}$	unit vector of ${}^{e}\mathbf{ER}$ and expressed in $\Im_E$
${}^{e}\mathbf{v}_r$	linear velocity of $\Im_R$ that is expressed in $\Im_E$
${}^{e}\mathbf{v}_e$	linear velocity of $\Im_E$ that is expressed in its frame
${}^{e}\omega_e$	angular velocity of $\Im_E$ that is expressed in its frame
$\mathbf{e}_1$ and $\mathbf{e}_2$	alignment task error and second task error
$\mathbf{L}_{e_1}^T$	interactive matrix of alignment task error
λ	gain factor for alignment task error
$\mathbf{u}_{e_1}$	unit vector of alignment task error
γ	gain factor for second task error
$\mathbf{v}_{\perp_y}$	linear velocity perpendicular on ${}^{e}\mathbf{y}$
Γ	geometric path to be followed
M_k	k^{th} point on the path
$\mathbf{d}$ and $\dot{\mathbf{d}}$	projection distance between the tool tip and the path, and its time-derivative
S	the projected point on the path
$\mathbf{v}_s$	linear velocity of S along the path
$\dot{s}$	the speed of S along the path
$\mathbf{K}_s$	unit vector between two consecutive points along the path
$\mathbf{v}_{tissue}$	desired linear velocity along the tissue
β	gain factor for reducing $\mathbf{d}$
α	gain factor for $\mathbf{v}_{tissue}$

2.2.2 Case 1: Tool Outside Incision Point

This is the first phase for getting close to the fulcrum point. It is required to align the rigid tool with the incision point. To achieve this task, the error between the y-component of end-effector frame (${}^{e}\mathbf{y}$) and the unit vector oriented from end-effector origin point to incision origin point (${}^{e}\mathbf{u}_{er}$) should be equal to zero (1), where ($\wedge$) is the cross product between these two vectors.

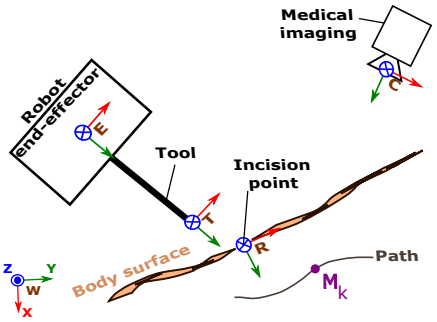


Figure 1: Representation of different reference frames used in the modelling of the whole system.

$$\mathbf{e}_1 = {}^{e}\mathbf{y} \wedge {}^{e}\mathbf{u}_{er} = \mathbf{0} \tag{1}$$

This task tracks the incision point and the end-effector in order to align both of them. In order to ensure exponential error decay, the control equation is $\dot{\mathbf{e}}_1 = -\lambda\mathbf{e}_1$, thereby the time-derivative of (1) is calculated as:

$$\dot{\mathbf{e}}_1 = {}^{e}\mathbf{y} \wedge {}^{e}\dot{\mathbf{u}}_{er} + \underbrace{\underbrace{{}^{e}\dot{\mathbf{ER}}}_{=0} \wedge {}^{e}\mathbf{u}_{er}}_{=0} \tag{2}$$

the time-derivative of vector ${}^{e}\mathbf{ER}$ represents the linear velocity of incision point expressed in end-effector frame (${}^{e}\dot{\mathbf{ER}} = {}^{e}\mathbf{v}_r$). This velocity must be equivalent to zero, and consequently the formulation (2) is reduced. The derivative of unit vector ${}^{e}\mathbf{u}_{er}$ with respect to time is calculated as follows:

$$\begin{aligned} {}^{e}\dot{\mathbf{u}}_{er} &= \frac{\|{}^{e}\mathbf{ER}\|\,{}^{e}\dot{\mathbf{ER}} - {}^{e}\mathbf{ER}\frac{d\|{}^{e}\mathbf{ER}\|}{dt}}{\|{}^{e}\mathbf{ER}\|^2} \\ \textit{where } \frac{d}{dt}\|{}^{e}\mathbf{ER}\| &= \frac{{}^{e}\mathbf{ER}^T\,{}^{e}\dot{\mathbf{ER}}}{\sqrt{{}^{e}\mathbf{ER}^T\,{}^{e}\mathbf{ER}}} \end{aligned} \tag{3}$$

and it is simplified as follows:

$$\begin{aligned} {}^{e}\dot{\mathbf{u}}_{er} &= \frac{{}^{e}\dot{\mathbf{ER}}}{\|{}^{e}\mathbf{ER}\|} - \frac{{}^{e}\mathbf{ER}\,{}^{e}\mathbf{ER}^T\,{}^{e}\dot{\mathbf{ER}}}{\|{}^{e}\mathbf{ER}\|^3} \\ &= \left(\frac{\mathbf{I}}{\|{}^{e}\mathbf{ER}\|} - \frac{{}^{e}\mathbf{u}_{er}\,{}^{e}\mathbf{u}_{er}^T}{\|{}^{e}\mathbf{ER}\|}\right){}^{e}\dot{\mathbf{ER}} \end{aligned} \tag{4}$$

The trocar velocity can be expressed in terms of end-effector velocity as:

$${}^{e}\mathbf{v}_r = -({}^{e}\mathbf{v}_e + {}^{e}\omega_e \wedge {}^{e}\mathbf{ER}) \tag{5}$$

By putting (5) in (4), the derivative of unit vector (${}^{e}\dot{\mathbf{u}}_{er}$) is represented as:

$${}^{e}\dot{\mathbf{u}}_{er} = \frac{-1}{\|{}^{e}\mathbf{ER}\|}(\mathbf{I} - {}^{e}\mathbf{u}_{er}\,{}^{e}\mathbf{u}_{er}^T)[\mathbf{I} - [{}^{e}\mathbf{ER}]_\wedge]\begin{bmatrix} {}^{e}\mathbf{v}_e \\ {}^{e}\omega_e \end{bmatrix} \tag{6}$$

where $-[{}^{e}\mathbf{ER}]_\wedge$ is the skew matrix of vector ${}^{e}\mathbf{ER}$ and $\mathbf{I}_{3\times3}$ is the identity matrix. By substituting (6) in (2),

the derivative of first error is defined as:

$$\dot{\mathbf{e}}_1 = \underbrace{\frac{-1}{\|{}^e\mathbf{ER}\|}[{}^e\mathbf{y}]_\wedge(\mathbf{I} - {}^e\mathbf{u}_{er}\,{}^e\mathbf{u}_{er}^T)[\mathbf{I} - [{}^e\mathbf{ER}]_\wedge]}_{\mathbf{L}_{e_1}^T}\underbrace{\begin{bmatrix} {}^e\mathbf{v}_e \\ {}^e\omega_e \end{bmatrix}}_{{}^e\tau_e} \tag{7}$$

$$\dot{\mathbf{e}}_1 = \mathbf{L}_{e_1}^T\,{}^e\tau_e = -\lambda\mathbf{e}_1 \tag{8}$$

where ($\mathbf{L}_{e_1}^T$) is the interaction matrix, (λ) is a gain factor for alignment task and ($^e\tau_e$) is the control velocity of end-effector which gather the linear and angular velocities ($^e\tau_e = \begin{bmatrix} {}^e\mathbf{v} \\ {}^e\omega \end{bmatrix}$). This control velocity is achieved by inverting the interaction matrix ($\mathbf{L}_{e_1}^{T^\dagger}$) by singular value decomposition (SVD) in (8):

$${}^e\tau_e = -\lambda\mathbf{L}_{e_1}^{T^\dagger}\mathbf{e}_1 \tag{9}$$

A possible solution of end-effector velocity vector is calculated in (10) to bring the alignment error task in the null space.

$$\mathbf{L}_{e_1}^T\begin{bmatrix} 0 \\ \frac{\lambda}{{}^e\mathbf{y}^T\,{}^e\mathbf{ER}}\mathbf{e}_1 \end{bmatrix} = -\lambda\mathbf{e}_1 \tag{10}$$

The constraints are extended by adding another task ($\mathbf{e}_2 = {}^eR - {}^eT$) that brings the tool tip (T) to the incision point (R). The second task error is projected in the null space of first task error that its interaction matrix in nullity ($\mathbf{L}_{e_{1_{Ker}}}^T$) is defined as:

$$\mathbf{L}_{e_{1_{Ker}}}^T = \begin{bmatrix} {}^e\mathbf{u}_{er} & \mathbf{0} & \|{}^eR\| * ({}^e\mathbf{u}_{er} \wedge \mathbf{u}_{e_1}) & -\|{}^eR\| * \mathbf{u}_{e1} \\ \mathbf{0} & {}^e\mathbf{u}_{er} & \mathbf{u}_{e_1} & {}^e\mathbf{u}_{er} \wedge \mathbf{u}_{e_1} \end{bmatrix} \tag{11}$$

The latter projection is also valid for the second case where the tool moves inside the hole, where ($\|{}^eR\|$) is the euclidean norm of $\mathbf{R}$, and ($\mathbf{u}_{e_1}$) is the unit vector of $\mathbf{e}_1$.

2.2.3 Case 2: Tool Inside Incision Point

During this phase, the tool follows a pre-defined path and its velocity ($^e\mathbf{v}_t$) is determined by the path following algorithm (see Section 2.3). The tool tip velocity is transmitted to the end-effector as:

$${}^e\mathbf{v}_t = {}^e\mathbf{v}_e + {}^e\omega \wedge {}^e\mathbf{ET} \tag{12}$$

The incision wall allows only the tool translation along the y-component of end-effector frame (Figure 2). The mathematical representation of RCM constraint is:

$${}^e\mathbf{v}_r \wedge {}^e\mathbf{y} = 0 \tag{13}$$

The linear velocity vector of incision point is projected on the y-component of end-effector in order to find its solution and maintain the bilateral constraint (13):

$$(\mathbf{I} - {}^e\mathbf{y}\,{}^e\mathbf{y}^T){}^e\mathbf{v}_r = 0 \tag{14}$$

$${}^e\mathbf{v}_r = {}^e\mathbf{v}_e + {}^e\omega \wedge {}^e\mathbf{ER} \tag{15}$$

Putting (12) in (15), the RCM velocity is described in terms of tool tip velocity:

$${}^e\mathbf{v}_r = {}^e\mathbf{v}_t + {}^e\omega \wedge \underbrace{{}^e\mathbf{ER} - {}^e\mathbf{ET}}_{={}^e\mathbf{TR}} \tag{16}$$

By substituting (16) in (14), the equation (17) is divided into two parts. The first one is the linear velocity perpendicular to $^e\mathbf{y}$ and the second is the angular velocity that is reduced because $^e\omega \wedge {}^e\mathbf{TR}$ is perpendicular to $^e\mathbf{y}$.

$$\underbrace{(\mathbf{I} - {}^e\mathbf{y}\,{}^e\mathbf{y}^T){}^e\mathbf{v}_t}_{\mathbf{v}_{\perp_y}} + \underbrace{(\mathbf{I} - {}^e\mathbf{y}\,{}^e\mathbf{y}^T)({}^e\omega \wedge {}^e\mathbf{TR})}_{-{}^e\mathbf{TR} \wedge {}^e\omega} = 0 \tag{17}$$

The angular velocity of end-effector ($^e\omega$) is calculated by:

$${}^e\omega = \frac{\mathbf{v}_{\perp_y} \wedge {}^e\mathbf{TR}}{\|{}^e\mathbf{TR}\|^2} = \frac{\mathbf{v}_{\perp_y} \wedge {}^e\mathbf{y}}{\|{}^e\mathbf{TR}\|} \tag{18}$$

Thereby, the linear velocity of end-effector is determined by replacing (18) into (12) and accordingly this will result to the following:

$${}^e\mathbf{v}_e = {}^e\mathbf{v}_t - {}^e\omega \wedge {}^e\mathbf{ET} \tag{19}$$

The second task error in this case is determined by the path following ($\mathbf{e}_2 = {}^e\mathbf{T} - {}^e\mathbf{T}^*$) that is the error between the actual tool tip position vector ($^e\mathbf{T}$) and the desired one ($^e\mathbf{T}^*$). The error is projected in the null space of first task as used in (11).

2.3 3D Path Following

2.3.1 Problem Statement

The desired geometric path is generally defined by a planning algorithm (Gasparetto et al., 2015), for avoiding obstacles and generating the shortest distance between the initial point and the target one, or simply by the surgeon drawing on input device, such as tablet [1].

During the robot motion, the perpendicular distance ($\mathbf{d}$) between the tool tip and the desired path points should be maintained to zero (Figure 2). In addition, it is required to determine the tool velocity along the desired path.

2.3.2 Problem Resolution

The projection of tool tip on the path provides the point (S) and the projected distance (20) which is required to be as minimal as possible.

$$\mathbf{d} = T - S \tag{20}$$

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

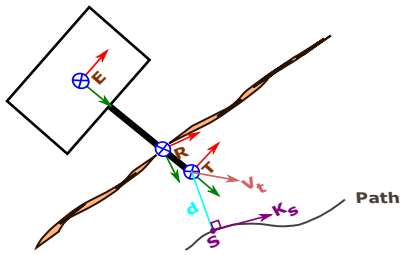


Figure 2: Representation of different reference frames used in the path following.

The time-derivative of (20) is obtained in (21). The projected point velocity ($\mathbf{v}_s$) is defined as the speed ($\dot{s}$) in the direction of the instantaneous unit vector ($\mathbf{K}_s$) that is tangent to the path.

$$\begin{aligned}\dot{\mathbf{d}} &= \dot{\mathbf{T}} - \dot{\mathbf{S}} \\ &= \mathbf{v}_t - \mathbf{v}_s \\ &= \mathbf{v}_t - \dot{s}\mathbf{K}_s\end{aligned} \tag{21}$$

The instantaneous tangential vector ($\mathbf{K}_s$) is calculated in (22). ($\mathbf{K}_s^+$) and ($\mathbf{K}_s^-$) are the previous and next tangential vectors, respectively, and (M_k) is the k^{th} point on the geometric path.

$$\begin{aligned}\mathbf{K}_s &= \frac{M_{k+1} - M_k}{\|M_{k+1} - M_k\|} \\ \mathbf{K}_s^+ &= \frac{M_{k+2} - M_{k+1}}{\|M_{k+2} - M_{k+1}\|} \\ \mathbf{K}_s^- &= \frac{M_k - M_{k-1}}{\|M_k - M_{k-1}\|}\end{aligned} \tag{22}$$

The derivative of instantaneous tangential vector is computed as:

$$\dot{\mathbf{K}}_s = \frac{d\mathbf{K}_s}{dt} = \frac{\partial \mathbf{K}_s}{\partial s}\frac{ds}{dt} = \frac{\mathbf{K}_s^+ - \mathbf{K}_s^-}{2 \triangle s}\dot{s} \tag{23}$$

The latter time-derivative is the instantaneous velocity vector to move from point M_k to M_{k+1}. It is also the perpendicular resultant (25) from the cross product of the unit vector $\mathbf{K}_s$ and the angular velocity ω, which depends on the speed along the path and its curvature:

$$\omega = \dot{s}\mathbf{C}(s) \tag{24}$$

$$\dot{\mathbf{K}}_\mathbf{s} = \dot{s}\mathbf{C}(s) \wedge \mathbf{K}_s \tag{25}$$

From (23) and (25), the path curvature ($\mathbf{C}(s)$) is calculated as:

$$\mathbf{C}(s) = -\mathbf{K}_s \wedge \frac{\mathbf{K}_s^+ - \mathbf{K}_s^-}{2 \triangle s} \tag{26}$$

Since the projected distance is perpendicular on the tangential vector ($\mathbf{d}^T\mathbf{K}_s = 0$), then the time-derivative of the latter expression is concluded as:

$$\dot{\mathbf{d}}^T\mathbf{K}_s + \mathbf{d}^T\dot{\mathbf{K}}_s = 0 \tag{27}$$

In order to calculate the required speed along the path, (21) is modified to:

$$\dot{\mathbf{d}}\mathbf{K}_s = \mathbf{v}_t\mathbf{K}_s - \dot{s}\mathbf{K}_s\mathbf{K}_s \tag{28}$$

By putting (25) and (28) in (27), the speed along the path is determined as:

$$\dot{s} = \frac{\mathbf{v}_t^T\mathbf{K}_s}{1 - \mathbf{d}^T(\mathbf{C}(s) \wedge \mathbf{K}_s)} \tag{29}$$

Back substituting (29) in (21), the velocity required to bring the tool tip on the path is defined as following which is the kinematic state-space representation:

$$\dot{\mathbf{d}} = \left(\mathbf{I} - \frac{\mathbf{K}_s\mathbf{K}_s^T}{1 - \mathbf{d}^T(\mathbf{C}(s) \wedge \mathbf{K}_s)}\right)\mathbf{v}_t \tag{30}$$

The velocity profile of tool is to be set freely. A possible solution (31) is describing the tool velocity as two components: the first one to advance the tool along the path, and the second to reduce the distance between the tool and the path.

$$\mathbf{v}_t = \alpha\mathbf{K}_s + \beta\mathbf{d} \tag{31}$$

Thereby, (31) gets into (30):

$$\dot{\mathbf{d}} = \alpha[1 - \frac{1}{1 - \mathbf{d}^T(\mathbf{C}(s) \wedge \mathbf{K}_s)}]\mathbf{K}_s + \beta\mathbf{d} \tag{32}$$

As result the control problem becomes to determine the gain coefficients (α and β).

3 VALIDATION

3.1 Implementation

Algorithm 1 realizes the RCM motion and it is divided mainly into two phases. The first phase is getting the robot close to the incision point and align the tool with the y-component of RCM frame. The second phase is guiding the robot to perform the pre-defined 3D path. The function $generate\_geometric\_path()$ creates the path with respect to the incision point. The first task error is computed as shown in (1). In the control loop, the robot velocity is obtained analytically (10) and the projection in the null space of first task (11). The projected velocity control vector is:

$${}^e\tau_{e_{ker}} = \mathbf{L}_{e_{1_{Ker}}} * \mathbf{L}^T_{e_{1_{Ker}}} * {}^e\tau_e \tag{33}$$

During the first phase, the second task error brings the tool tip to the incision point and the interaction matrix of this task is determined as, where its dimension is 3×6:

$$\mathbf{L}^T_{e_2} = -\begin{bmatrix}\mathbf{I}_{3\times3} & [{}^e\mathbf{R}]_\wedge\end{bmatrix} \tag{34}$$

The control velocity is computed to ensure exponential decay of second task error that is projected in the null-space of first task error, and (γ) is a gain factor of second task:

$$ {}^{e}\tau_{e} = -\lambda\gamma \mathbf{L}_{e_2}^{T^{\dagger}} \mathbf{e}_2 \tag{35} $$

During the insertion phase, the second task is the path following error and the control velocity is considered as mentioned in (18) and (19).

Algorithm 1: Control loop for RCM constraints.

$^{w}M_e \leftarrow initalization\_reference\_frames(W,R,E,T)$
$\Gamma \leftarrow generate\_geometric\_path()$
$(\mathbf{e}_1,\mathbf{e}_2) \leftarrow initial\_task\_errors({}^{e}\mathbf{y},\ {}^{e}\mathbf{u}_r,\ {}^{e}\mathbf{TR})$
$(approaching, inserting) \leftarrow (true, false)$
while *not path end* **do**
  ${}^{e}\tau_{e_{control}} \leftarrow analytical\_solution(\lambda, \mathbf{e}_1, {}^{e}\mathbf{y}, {}^{e}\mathbf{ER})$
  if $approach$ & $(norm(\mathbf{e}_2) < 0.0001)$ **then**
    $(approaching, inserting) \leftarrow (false, true)$
  end if
  if $approaching$ **then**
    $\mathbf{e}_2 \leftarrow {}^{e}\mathbf{TR}$
    $\mathbf{L}_{e_2}^{T} \leftarrow interactive\_matrix(I, [{}^{e}R]_{\wedge})$
    ${}^{e}\tau_{e} \leftarrow control\_law(\lambda, \gamma, \mathbf{e}_2, \mathbf{L}_{e_2}^{T^{\dagger}})$
  else
    $\mathbf{e}_2 \leftarrow {}^{e}\mathbf{T} - {}^{e}\mathbf{T}^{*}$
    ${}^{w}\mathbf{v}_t \leftarrow path\_following(\Gamma, {}^{w}\mathbf{T})$
    ${}^{w}\omega_e \leftarrow explicit\_solution({}^{e}\mathbf{TR}, {}^{e}\mathbf{y}, {}^{e}\mathbf{v}_{\perp_y})$
    ${}^{e}\tau_{e} \leftarrow control\_law({}^{e}\mathbf{v}_t, {}^{e}\omega_e, {}^{e}\mathbf{ET})$
  end if
  $\mathbf{L}_{e_{1_{Ker}}} \leftarrow projection\_null\_space({}^{e}\mathbf{u}_r, {}^{e}\vec{R}, \mathbf{u}_{e_1})$
  ${}^{e}\tau_{e_{ker}} \leftarrow projected\_velocities(\mathbf{L}_{e_{1_{Ker}}}, {}^{e}\tau_e)$
  ${}^{e}\tau_{e_{control}} \leftarrow send\_robot\_velocities({}^{e}\tau_{e_{ker}}, {}^{e}\tau_e)$
  $\mathbf{e}_1 \leftarrow update\_variables({}^{e}\mathbf{y}, {}^{e}\mathbf{u}_r)$
end while

Algorithm 2 computes the linear velocity of tool tip to follow the desired path. It gives priority to reach the path when the tool is far from it. When the error is relatively small, the calculated velocity (31) is the resultant velocity between that of tool tip and that along the path. The parameter (α) is obtained in the latter case as follows:

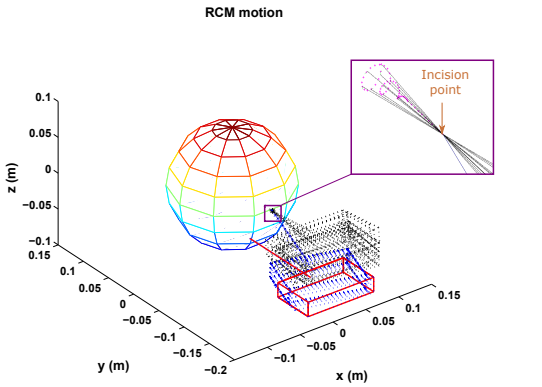
$$ \alpha = \sqrt{(\beta\|\mathbf{d}\|)^2 + \mathbf{v}_{tissue}^2} \tag{36} $$

Algorithm 2: Control loop for 3D path following.

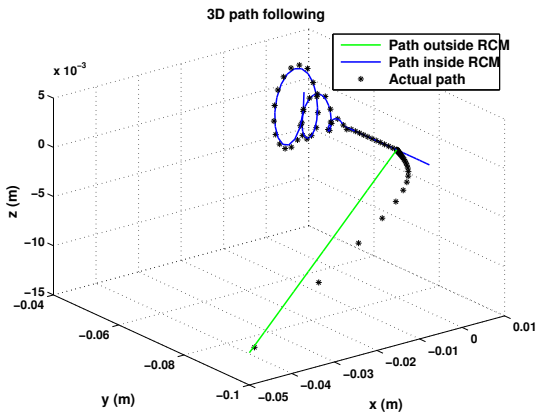
$(M_k, M_{k+1}) \leftarrow nearest\_point(\Gamma, \mathbf{T})$
$(\mathbf{K}_s, \mathbf{S}, \mathbf{d}) \leftarrow projection(M_k, M_{k+1}, \mathbf{T})$
if $(\beta\|\mathbf{d}\|)^2 > \mathbf{v}_{tissue}^2$ **then**
  $\alpha \leftarrow 0$
else
  $\alpha \leftarrow compute(\beta, \mathbf{d}, \mathbf{v}_{tissue})$
end if
$\mathbf{v}_t \leftarrow required\_velocity(\alpha, \beta, \mathbf{d})$

3.2 Results

A spherical workspace was chosen (Figure 3.a) in which the rigid tool navigates. The RCM will produce a conical workspace within the spherical one. Therefore, the desired 3D curve is defined as straight line from the incision point to the starting point of helical path. Figure 3.b presents the resultant tool tip position with respect to the 3D geometric path (in blue) and the shortest way between the initial position of tool tip and the incision point (in green). Throughout the tested simulation, the standard deviation error of RCM constraints during the insertion phase is around $0.004mm$ and that of path following is approximately $0.089mm$. These results in figure 4 are obtained with parameters values $\lambda = 0.3$, $\gamma = 0.3$, $\beta = -10$, $v_{tissue} = 0.001m/sec$ and sampling time $0.1sec$. Figure 4.a shows the RCM constraints error and positioning error during the approaching phase. Both errors are decreased exponentially as designed. Figure 4.b presents the same errors during the insertion phase where the RCM error is stable and the positioning error is oscillating due to the gain parameters.

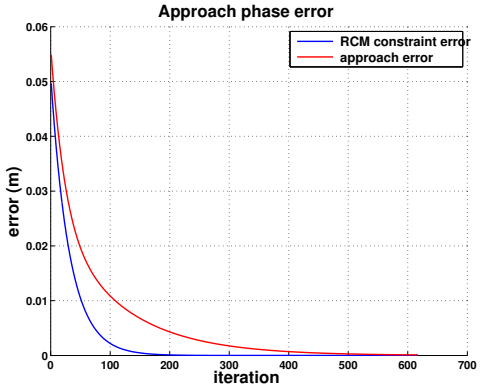


(a)

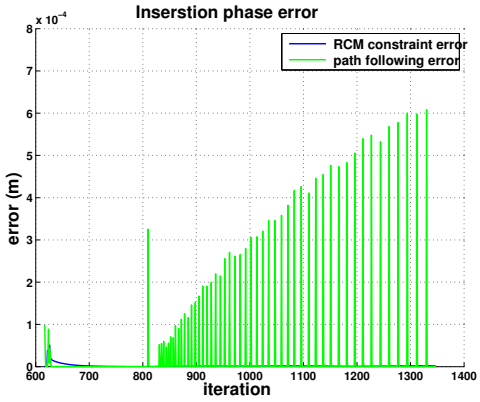


(b)

Figure 3: (a) The end-effector motion during approach phase (doted blue) and insertion phase (doted black); (b) the position of tool tip with respect to the path.



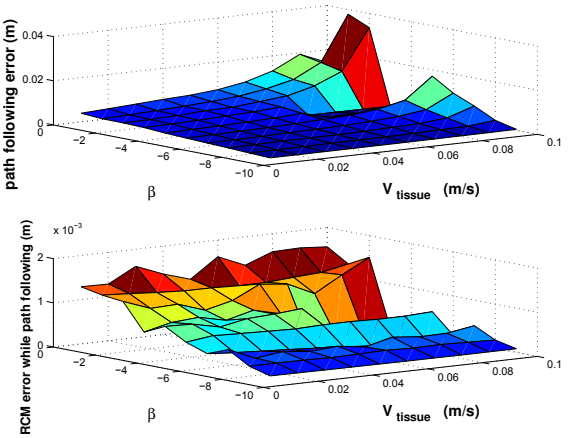
(a)



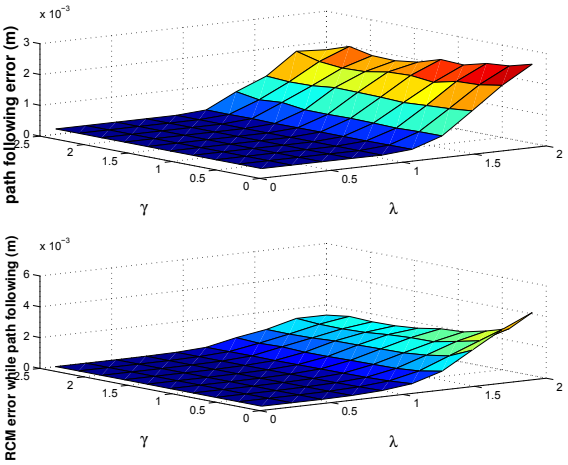
(b)

Figure 4: Motion error during (a) approach phase and (b) path following.

These coefficients effect the system performance and the problem becomes to choose the right values for these system variables. In order to visualize this effect in 3D surface, the error is calculated while varying two parameters and the others are fixed. In Figure 5.a, the variables β and v_{tissue} are varied from $-1*10^{-6}$ to -10 and 0.1 to $1*10^{-3}m/sec$, respectively. In Figure 5.b, the results are obtained by changing λ and γ from 2 to -0.1 and 2.5 to -0.1, respectively.



(a)



(b)

Figure 5: Effect of system variables on the error (a)v_{tissue} vs β and (b)λ vs γ.

4 CONCLUSIONS

The article presented a detailed method to arrange more than one task in hierarchical form, whereby the highest priority is the bilateral constraints and the second one is 3D path following task. The proposed method implements the controller for the usage of rigid tool but it could be modified easily in order to be adapted with other tool shape. This controller is useful for medical application, such as ENT (ear, nose and throat) surgeries and laparoscopic surgery; since it is accurate to follow the 3D path and maintain the trocar kinematics. It will be extended to consider unilateral RCM constraints where the incision hole is bigger than the tool diameter and the instrument has more space to move before it hits the incision wall.

ACKNOWLEDGEMENTS

This work is conducted with a financial support from the project NEMRO (ANR-14-CE17-0013-01) funded by the ANR and the financial support of the Franche-Comté region (FRANCHIR), France. It is also performed in the framework of the Labex ACTION (ANR-11-LABEX-01-001).

REFERENCES

Aghakhani, N., Geravand, M., Shahriari, N., Vendittelli, M., and Oriolo, G. (2013). Task control with remote center of motion constraint for minimally invasive robotic surgery. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5807–5812.

Azimian, H., Patel, R. V., and Naish, M. D. (2010). On constrained manipulation in robotics-assisted minimally invasive surgery. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 650–655.

Azizian, M., Khoshnam, M., Najmaei, N., and Patel, R. V. (2014). Visual servoing in medical robotics: a survey. part i: endoscopic and direct vision imaging–techniques and applications. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 10(3):263–274.

Boctor, E. M., Webster III, R. J., Mathieu, H., Okamura, A. M., and Fichtinger, G. (2004). Virtual remote center of motion control for needle placement robots. *Computer Aided Surgery*, 9(5):175–183.

Dalvand, M. M. and Shirinzadeh, B. (2012). Remote centre-of-motion control algorithms of 6-rrcrr parallel robot assisted surgery system (pramiss). In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3401–3406.

Duflot, L.-A., Krupa, A., Tamadazte, B., and Andreff, N. (2016). Towards ultrasound-based visual servoing using shearlet coefficients. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Funda, J., Taylor, R. H., Eldridge, B., Gomory, S., and Gruben, K. G. (1996). Constrained cartesian motion control for teleoperated surgical robots. *IEEE Transactions on Robotics and Automation*, 12(3):453–465.

Gasparetto, A., Boscariol, P., Lanzutti, A., and Vidoni, R. (2015). Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems*, pages 3–27. Springer.

Krupa, A., Doignon, C., Gangloff, J., and De Mathelin, M. (2002). Combined image-based and depth visual servoing applied to robotized laparoscopic surgery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 323–329.

Kuo, C.-H., Dai, J. S., and Dasgupta, P. (2012). Kinematic design considerations for minimally invasive surgical robots: an overview. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 8(2):127–145.

Locke, R. C. and Patel, R. V. (2007). Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery. In *IEEE International Conference on Robotics and Automation*, pages 1900–1905.

Marinho, M. M., Bernardes, M. C., and Bó, A. P. (2014). A programmable remote center-of-motion controller for minimally invasive surgery using the dual quaternion framework. In *5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 339–344.

Mayer, H., Nagy, I., and Knoll, A. (2004). Kinematics and modelling of a system for robotic surgery. In *On Advances in Robot Kinematics*, pages 181–190. Springer.

Nageotte, F., Zanne, P., Doignon, C., and de Mathelin, M. (2006). Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2364–2369.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15.

Osa, T., Staub, C., and Knoll, A. (2010). Framework of automatic robot surgery system using visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1837–1842.

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

Seon, J.-A., Tamadazte, B., and Andreff, N. (2015). Decoupling path following and velocity profile in vision-guided laser steering. *IEEE Transactions on Robotics*, 31(2):280–289.