

Task Controller For Performing Remote Centre Of Motion

Bassem Dahroug, Brahim Tamadazte, and Nicolas Andreff

FEMTO-ST, AS2M department, Univ. Bourgogne Franche-Comté/CNRS/ENSMM,
24 Rue Alain Savary, 25000 Besançon, France.

{bassem.dahroug, brahim.tamadazte, nicolas.andreff}@femto-st.fr

<http://www.femto-st.fr/en/>

Abstract. Remote centre of motion (RCM) is an essential movement done by the surgeon during minimal invasive surgery. While this motion, the centre point of incision hole should be superposed with the surgical tool in order to prevent patient harm. The paper presents the geometric modelling of this constrained motion in task-space (operational-space) for implementing a velocity controller with task priority form. The RCM constraints considered as the highest priority and the lower task is following a 3D reference path. The proposed controller shows the advantage of accurate motion control for executing complex movements within the patient body; since the results shows standard deviation around $0.026mm$ of path following task error and that of RCM equals zero.

1 Introduction

In recent years, the surgical assisted-robotics are gaining more demand for its benefits to the surgeon by providing ergonomic conditions in order to increase accuracy and reduce fatigue. The patients also benefit by a reduction in invasion, time and costs. These robotic systems enter the human body from a small incision (or trocar point) which presents physical constraints on the surgical tool motion. These constraints created by the incision wall reduce the tool degrees of freedom (DOF) to four (three rotations and one translation). The resultant motion from these constraints is called *remote center of motion* (RCM) or *fulcrum effect*. This type of motion could be achieved either with specific kinematic robot structure [10] or with software control [5]. This chapter is interested in the latter type because it is a generic method that could be applied regardless the robot structure, in condition that the robot DOF should be greater than 4-DOF. The latter condition ensures that the robot kinematic structure is redundant. The redundancy occurs when the manipulator joints number (its DOF) is greater than those required to execute the desired task which could be any kinematic or dynamic goal. The advantages of redundancy are increasing the robot manoeuvrability and dexterity that are useful to avoid singularity, joints limits and workspace obstacles, and to provide the concept of task priority [14].

Fulcrum effect is necessary during key-hole surgery, such as laparoscopy [7] [13] [11], eye surgery or ear surgery. The assistant-robot will help the surgeon

to decide the best strategy during the post-operative phase in order to reach the region of interest (ROI) inside the patient's body. These decisions are taken with the help of planning software which segments the medical images for creating a surface model of ROI and plans the optimal path/trajectory [8]. After that, during intra-operative phase, the navigation software collects continuously information about the robot internal state (i.e., the pose of its end-effector with respect to its base) and its relative pose with respect to organs. The manipulator relative pose is very important during the navigation phase and one way to estimate it is by using visual servoing control approach [3] which mimic the perception sense for the surgeon. This method uses real-time imaging (e.g., endoscope, optical coherence tomography or ultrasound) to detect, track and guide the instrument [9] [6]. Eventually, visual servoing method was combined with trajectory or path following controller for implementing an accurate velocity controller.

The difference between the two approaches is the notation of time. On one side, trajectory following parametrizes the geometric curve with time, where the velocity along the path is dependent on how the tool advances. In case that the tool does not reach the next scheduled point on the desired curve due to any difficulties during the previous points in the curve. Thereby the controller accelerates or tends to shortcut the defined curve in order to reduce the time delay. On the other side, path following is time independent and it is restricted to maintain the reference velocity profile even in lag conditions. The latter controller is useful in medical applications, especially during ablation or scanning processes. Since the surgeon can change the instrument velocity independently from the geometric curve and dependent on tissue-tool interaction. Path following is widely used for mobile robot but it is not frequently applied in medical application. A 2D path following proposed [18] for laser surgery. Non-holonomic control was proposed for executing a unicycle path following with high frequency. A 3D path following [16] proposed for guiding a magnetic helical swimmer.

This chapter discusses the problem of 3D path following combined with RCM constraints. It extends the proposed method in [5] to control the motion of rigid tool

2 CONTROL DESIGN

The proposed controller commands the robot velocity for performing 3D path following with RCM constraints. It achieves the objective with two task errors: the first prior task is the alignment of the tool with incision point, and 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 Remote Center of Motion Constraints

There are different reported methods in the literature for modelling and solving fulcrum problem. Trocar kinematics could be described with Euler angle representation as proposed in [13] for achieving an analytical solution. A more general solution was proposed in [7] to extend the robot Jacobian with quadratic optimization. Both methods have the objective to implement a Cartesian control.

An alternative technique is creating a "virtual" RCM [4] and the solution is derived from artificial intelligence based heuristic search. The first one is the distance (\mathbf{e}_1) between the tool tip and the target point inside the cavity. The second function is the cross-product (\mathbf{e}_2) between the rigid tool vector and the vector from RCM to target point. 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 a task function without respecting the other one (i.e., $(\mathbf{e}_1, \mathbf{e}_2) = (\mathbf{e}_1 \neq \mathbf{0}, \mathbf{0})$ or $(\mathbf{e}_1, \mathbf{e}_2) = (\mathbf{0}, \mathbf{e}_2 \neq \mathbf{0})$).

In later work [2], they described the trocar kinematics in vector form for one rigid link. The trocar point is assumed to be located along the link. This assumption means that the controller manage the situation where the tool is already inside the incision point, and it neglects the approaching phase where the instrument moves from its initial pose to the incision point. The solution in joint-space is achieved by gradient projection approach in closed-loop form. This is a good method to assure the priority between different tasks which are the manipulator kinematic and trajectory following.

In [15], the fulcrum effect is described in geometric way with stereo visual servoing for controlling the robot position from point-to-point. In the same direction, [1] formulated the RCM constraints as [2] and the formulation is extended to serial links. They extended the visual servoing task, which has the highest priority, by adding the trocar position error. Recent works synthesized the kinematic motion by using quaternion frameworks [12] or Lie algebra [17] for implementing velocity controller.

In the remain of this section, the mathematical formulation is presented for modelling the robot motion during the approaching and insertion phases, after that the solution for 3D path following.

Problem Statement Figure 1 shows the conceptual system with the different reference coordinate frames. The y -component of RCM frame (${}^r\mathbf{y}$) is assumed to be perpendicular to the tissue surface. The RCM constraints consequently allow the tool translation along the y -component of the current RCM frame (${}^r\mathbf{v}$) and angular rotation (${}^r\omega$) around the latter frame axes. Once 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 2.2). The problem becomes to achieved this motion, while respecting the RCM constraints, by applying the adequate end-effector velocity (${}^e\mathbf{v}$ and ${}^e\omega$)

Case 1: Tool Outside Incision Point This is the first phase for approaching to the fulcrum point. It is required to align the rigid tool with the incision point.

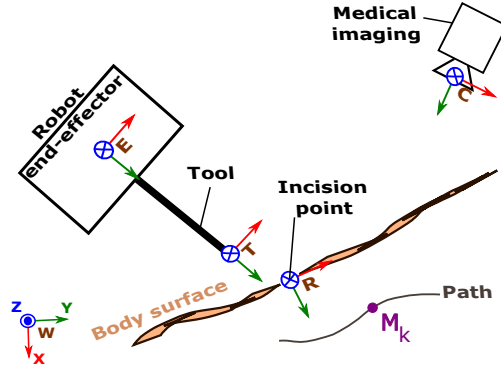


Fig. 1. Representation of different reference frames used in the modelling of the whole system.

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.

$$\mathbf{e}_1 = {}^e\mathbf{y} \wedge {}^e\mathbf{u}_{er} \quad (1)$$

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:

$$\dot{\mathbf{e}}_1 = {}^e\dot{\mathbf{y}} \wedge {}^e\mathbf{u}_{er} + \underbrace{{}^e\dot{\mathbf{y}} \wedge {}^e\mathbf{u}_{er}}_{=0} \quad (2)$$

The derivative of unit vector ${}^e\mathbf{u}_{er}$ with respect to time is calculated as follows [5]:

$${}^e\dot{\mathbf{u}}_{er} = \left(\frac{\mathbf{I}}{\|{}^e\mathbf{ER}\|} - \frac{{}^e\mathbf{u}_{er} {}^e\mathbf{u}_{er}^T}{\|{}^e\mathbf{ER}\|} \right) {}^e\mathbf{ER} \quad (3)$$

Where ${}^e\mathbf{ER}$ represents the linear velocity of incision point expressed in end-effector frame (${}^e\mathbf{ER} = {}^e\mathbf{v}_r$). The trocar velocity can be expressed in terms of end-effector velocity as:

$${}^e\mathbf{v}_r = -({}^e\mathbf{v}_e + {}^e\omega_e \wedge {}^e\mathbf{ER}) \quad (4)$$

By putting (4) in (3), the derivative of unit vector (${}^e\dot{\mathbf{u}}_{er}$) is represented as:

$${}^e\dot{\mathbf{u}}_{er} = \frac{-1}{\|{}^e\mathbf{ER}\|} (\mathbf{I} - {}^e\mathbf{u}_{er} {}^e\mathbf{u}_{er}^T) [\mathbf{I} - [{}^e\mathbf{ER}]_{\wedge}] \begin{bmatrix} {}^e\mathbf{v}_e \\ {}^e\omega_e \end{bmatrix} \quad (5)$$

where $-[{}^e\mathbf{ER}]_{\wedge}$ is the skew matrix of vector ${}^e\mathbf{ER}$ and $\mathbf{I}_{3 \times 3}$ is the identity matrix. By substituting (5) in (2), the derivative of first error is defined as:

$$\dot{\mathbf{e}}_1 = \underbrace{\frac{-1}{\|{}^e\mathbf{ER}\|} [{}^e\mathbf{y}]_{\wedge} (\mathbf{I} - {}^e\mathbf{u}_{er} {}^e\mathbf{u}_{er}^T)}_{\mathbf{L}_{e_1}^T} \underbrace{[\mathbf{I} - [{}^e\mathbf{ER}]_{\wedge}] \begin{bmatrix} {}^e\mathbf{v}_e \\ {}^e\omega_e \end{bmatrix}}_{{}^e\tau_e} \quad (6)$$

$$\dot{\mathbf{e}}_1 = \mathbf{L}_{e_1}^T {}^e\tau_e = -\lambda \mathbf{e}_1 \quad (7)$$

where $(\mathbf{L}_{e_1}^T)$ is the interaction matrix of alignment task, (λ) is a gain factor for alignment task and $({}^e\tau_e)$ is the control twist velocity of end-effector which gather the linear and angular velocities (${}^e\tau_e = \begin{bmatrix} {}^e\mathbf{v} \\ {}^e\omega \end{bmatrix}$). The latter control velocity is achieved by inverting the interaction matrix $(\mathbf{L}_{e_1}^{T\dagger})$ in (7) by psuedo-inverse or singular value decomposition (SVD) as:

$${}^e\tau_e = -\lambda \mathbf{L}_{e_1}^{T\dagger} \mathbf{e}_1 \quad (8)$$

In fact, these heavy computational methods can be avoided; since the null space of interaction matrix can be computed analytically as:

$$\begin{aligned} \ker(\mathbf{L}_{e_1}^T) &= \mathbf{I} - \mathbf{L}_{e_1}^{T\dagger} \mathbf{L}_{e_1}^T \\ &= \begin{bmatrix} {}^e\mathbf{u}_{er} & \mathbf{0} & \|{}^eR\| * ({}^e\mathbf{u}_{er} \wedge \mathbf{u}_{e_1}) - \|{}^eR\| * \mathbf{u}_{e_1} \\ \mathbf{0} & {}^e\mathbf{u}_{er} & \mathbf{u}_{e_1} \end{bmatrix} \end{aligned} \quad (9)$$

where $(\|{}^eR\|)$ is the euclidean norm of \mathbf{R} , and (\mathbf{u}_{e_1}) is the unit vector of \mathbf{e}_1 . Another task can be projected in the null space of first task by using the latter equation (9). During this case, the second task ($\mathbf{e}_2 = {}^eR - {}^eT$) is bringing the tool tip (T) to the incision point (R).

Case 2: Tool Inside Incision Point During this phase, the tool follows a pre-defined path and its velocity is determined by the path following algorithm (see Section 2.2). The tool tip velocity (${}^e\mathbf{v}_t$) is expressed with respect to end-effector frame as:

$${}^e\mathbf{v}_t = {}^e\mathbf{v}_e + {}^e\omega \wedge {}^e\mathbf{ET} \quad (10)$$

In the same way, the velocity of RCM frame is formulated with respect to end-effector frame as:

$${}^e\mathbf{v}_r = {}^e\mathbf{v}_e + {}^e\omega \wedge {}^e\mathbf{ER} \quad (11)$$

Putting (10) in (11), the RCM velocity is described in function of tool tip velocity:

$${}^e\mathbf{v}_r = {}^e\mathbf{v}_t + {}^e\omega \wedge \underbrace{{}^e\mathbf{ER} - {}^e\mathbf{ET}}_{= {}^e\mathbf{TR}} \quad (12)$$

Since the incision wall allows only the tool translation along the y -component of end-effector frame (Figure 2), the velocity of RCM frame equals:

$${}^e\mathbf{v}_r = {}^e\mathbf{v}_t + {}^e\omega \wedge {}^e\mathbf{TR} = -\gamma {}^e\mathbf{y} \quad (13)$$

where γ is a scalar factor to increase or decrease the velocity in y -axis. From latter equation (13), the angular velocity of end-effector is deduced as:

$${}^e\omega \wedge {}^e\mathbf{TR} = -({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y}) \quad (14)$$

$${}^e\omega = \frac{-1}{\|{}^e\mathbf{TR}\|^2} ({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y}) \wedge {}^e\mathbf{TR} + \mu {}^e\mathbf{TR} \quad (15)$$

The second term in the right hand side of latter equation (15) represents the tool rotation around its axis and it is neglected. Back substituting (15) in (14), we obtain:

$$\begin{aligned} {}^e\omega \wedge {}^e\mathbf{TR} &= \frac{-1}{\|{}^e\mathbf{TR}\|^2} (({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y}) \wedge {}^e\mathbf{TR}) \wedge {}^e\mathbf{TR} \\ &= \frac{1}{\|{}^e\mathbf{TR}\|^2} \left({}^e\mathbf{TR} {}^e\mathbf{TR}^T ({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y}) + ({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y}) {}^e\mathbf{TR}^T {}^e\mathbf{TR} \right) \\ &= \frac{{}^e\mathbf{TR} {}^e\mathbf{TR}^T ({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y})}{\|{}^e\mathbf{TR}\|^2} + ({}^e\mathbf{v}_t + \gamma {}^e\mathbf{y}) \end{aligned} \quad (16)$$

In order to obtain equation (16) equals (14), the choice of γ is determined as:

$$\gamma = -\frac{{}^e\mathbf{v}_t^T {}^e\mathbf{TR}}{{}^e\mathbf{y}^T {}^e\mathbf{TR}} \quad (17)$$

By placing (17) in (15), the control angular velocity of end-effector (${}^e\omega$) is calculated. Thereby, the control linear velocity of end-effector is determined by replacing (15) in (10) which gives:

$${}^e\mathbf{v}_e = {}^e\mathbf{v}_t - {}^e\omega \wedge {}^e\mathbf{ET} \quad (18)$$

During this case, the second task is determined by the path following controller. The path following error is projected in the null space of alignment task as used in (9).

2.2 3D Path Following

Problem Statement The desired geometric curve is generally defined by a planning algorithm [8], 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). Beside that it is required to determine the tool velocity along the desired path.

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

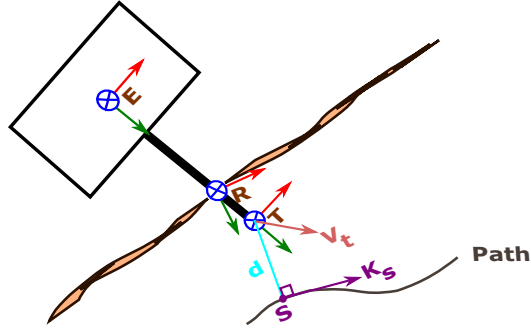


Fig. 2. Representation of different reference frames used in the path following.

Problem Resolution The projection of tool tip on the path provides the point (S) and the projected distance (19) which is required to be minimized.

$$\mathbf{d} = T - S \quad (19)$$

The time-derivative of previous equation is obtained as [5]:

$$\begin{aligned} \dot{\mathbf{d}} &= \mathbf{v}_t - \mathbf{v}_s \\ &= \mathbf{v}_t - \dot{s}\mathbf{K}_s \end{aligned} \quad (20)$$

where (\mathbf{v}_t) is the linear velocity of tool tip and (\mathbf{v}_s) is the linear velocity of projected point which is defined as the speed (\dot{s}) in the direction of the instantaneous unit vector (\mathbf{K}_s) that is tangent to the path. The instantaneous tangential vector (\mathbf{K}_s) is calculated in (21). (\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.

$$\begin{aligned} \mathbf{K}_s &= \frac{M_{k+1} - M_k}{\|M_{k+1} - M_k\|} \\ \mathbf{K}_s^+ &= \frac{M_{k+2} - M_{k+1}}{\|M_{k+2} - M_{k+1}\|} \\ \mathbf{K}_s^- &= \frac{M_k - M_{k-1}}{\|M_k - M_{k-1}\|} \end{aligned} \quad (21)$$

The derivative of instantaneous tangential vector is computed as:

$$\dot{\mathbf{K}}_s = \frac{d\mathbf{K}_s}{dt} = \frac{\partial \mathbf{K}_s}{\partial s} \frac{ds}{dt} = \frac{\mathbf{K}_s^+ - \mathbf{K}_s^-}{2 \Delta s} \dot{s} \quad (22)$$

The latter time-derivative is the instantaneous velocity to move from point M_k to M_{k+1} . It is also the perpendicular resultant (24) from the unit vector \mathbf{K}_s and the angular velocity ω that depends on the speed along the path and its curvature:

$$\omega = \dot{s}\mathbf{C}(s) \quad (23)$$

$$\dot{\mathbf{K}}_s = \dot{s}\mathbf{C}(s) \wedge \mathbf{K}_s \quad (24)$$

From (22) and (24), the path curvature ($\mathbf{C}(s)$) is calculated as:

$$\mathbf{C}(s) = -\mathbf{K}_s \wedge \frac{\mathbf{K}_s^+ - \mathbf{K}_s^-}{2 \Delta s} \quad (25)$$

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 obtained as:

$$\dot{\mathbf{d}}^T \mathbf{K}_s + \mathbf{d}^T \dot{\mathbf{K}}_s = 0 \quad (26)$$

In order to calculate the required speed along the path, the equation (20) is modified to:

$$\dot{\mathbf{d}} \mathbf{K}_s = \mathbf{v}_t \mathbf{K}_s - \dot{s} \mathbf{K}_s \mathbf{K}_s \quad (27)$$

By putting (24) and (27) in (26), the speed along the path is determined as:

$$\dot{s} = \frac{\mathbf{v}_t^T \mathbf{K}_s}{1 - \mathbf{d}^T (\mathbf{C}(s) \wedge \mathbf{K}_s)} \quad (28)$$

Back substituting (28) in (20), the velocity required to bring the tool tip on the path is defined as following which is the kinematic state-space representation:

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

The tool velocity is free to be chosen and a possible solution (30) to describe it 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.

$$\mathbf{v}_t = \alpha \mathbf{K}_s + \beta \mathbf{d} \quad (30)$$

Thereby, (30) gets into (29):

$$\dot{\mathbf{d}} = \alpha \left[1 - \frac{1}{1 - \mathbf{d}^T (\mathbf{C}(s) \wedge \mathbf{K}_s)} \right] \mathbf{K}_s + \beta \mathbf{d} \quad (31)$$

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

3 VALIDATION

A simulator was implemented for testing the proposed controller by deploying Octave or Matlab software. It is divided mainly into two phases: i) the first phase is approaching the robot to the incision point and align the the tool with the y -component of RCM frame, ii) the second phase is guiding the robot to perform the pre-defined 3D path. Figure 3 shows the block diagram of proposed controller while the second phase. At the first phase, the path following block is replaced by a simple point tracking controller. The desired path should be defined by the user at the beginning of simulation. During the control loop, the tool pose is simulated that is estimated by exteroceptive sensor (e.g., camera).

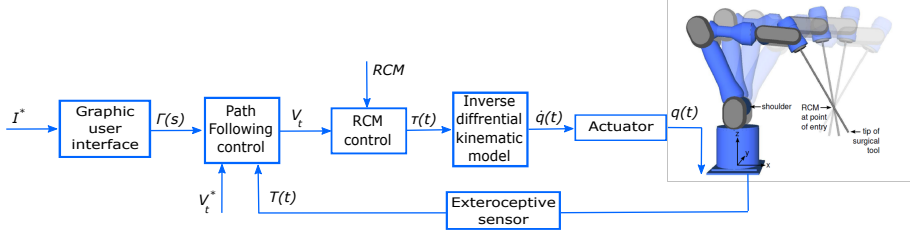


Fig. 3. Block diagram of RCM with path following control.

3.1 Implementation

The pseudo code (Algorithm 1) implements the block diagram in figure 3). The different reference coordinate frames (i.e., the RCM point, tool tip and end-effector frames with respect to world frame) are initialize by the function *initial_reference_frames()*. The function *generate_geometric_path()* creates the path with respect to incision point. At the start of control loop, the RCM variables are calculated in the function *compute_RCM_variables()*, as demonstrated previously in (6) and (7), for deducing the twist velocity of end-effector in order to accomplish the RCM alignment task.

If the control loop is executing the approach phase, the second task will bring the tool tip to the incision point ($\mathbf{e}_2 = {}^e \mathbf{TR}$). A simple point tracking controller is deployed for ensuring the exponential decay of second error and the control velocity of second task is deduced as, where *gamma* is a gain factor and $\mathbf{L}_{e_2}^{T\dagger}$ is the inverse of interaction matrix of this task:

$$\mathbf{L}_{e_2}^T = -[\mathbf{I}_{3 \times 3} \quad [{}^e R]_{\wedge}] \quad (32)$$

$${}^e \tau_e = -\gamma \mathbf{L}_{e_2}^{T\dagger} \mathbf{e}_2 \quad (33)$$

If the control loop is executing the insertion phase, the second task is the path following error ($\mathbf{e}_2 = \mathbf{d}$) and the control velocity of path following task is considered as mentioned in (30). The geometric solution of end-effector twist velocity is then calculated as (15) and (18) for accomplishing the second task.

After that, the solution of second task is projected in the null-space of first task (9) and the projected velocity is obtained as:

$${}^e \tau_{e_{ker}} = \mathbf{L}_{e_1 Ker} * \mathbf{L}_{e_1 Ker}^T * {}^e \tau_{e_2} \quad (34)$$

The latter secondary velocity is added to RCM alignment velocity (8), which could be chosen as any column of null-space matrix (9). The resultant velocity is the final solution that will be send to the robot.

Algorithm 2 computes the linear velocity of tool tip to follow the desired path. It begins by projecting the actual position of tool tip on the reference curve in order to determine the projected distance and the advancement direction. The resultant path following velocity is a combination of velocities in the both latter

Algorithm 1 Control loop for RCM constraints

```

 ${}^w\mathbf{M}_e \leftarrow \text{initial\_reference\_frames}(W, R, E, T)$ 
 $\Gamma \leftarrow \text{generate\_geometric\_path}()$ 
 $\text{approaching} \leftarrow \text{true}$ 
while not path end do
   $(\mathbf{e}_1, \mathbf{L}_{e_1}^T, {}^e\tau_{e_1}) \leftarrow \text{compute\_RCM\_variables}({}^w\mathbf{M}_e, {}^e\mathbf{u}_r, {}^e\mathbf{T}, \lambda)$ 
  if approach & (norm( $\mathbf{e}_2$ ) < 0.0001) then
     $\text{approaching} \leftarrow \text{false}$ 
  end if
  if approaching then
     $\mathbf{e}_2 \leftarrow {}^e\mathbf{TR}$ 
     $\mathbf{L}_{e_2}^T \leftarrow \text{interactive\_matrix}(I, [{}^eR]_{\wedge})$ 
     ${}^e\tau_{e_2} \leftarrow \text{control\_law}(\lambda, \gamma, \mathbf{e}_2, \mathbf{L}_{e_2}^{T\dagger})$ 
  else
     $(\mathbf{e}_2, {}^w\mathbf{v}_t) \leftarrow \text{path\_following}({}^w\mathbf{M}_e, {}^w\mathbf{T}, \Gamma)$ 
     ${}^w\omega_{e_2} \leftarrow \text{explicit\_solution}({}^e\mathbf{TR}, {}^e\mathbf{y}, {}^w\mathbf{v}_t)$ 
     ${}^e\tau_{e_2} \leftarrow \text{control\_law}({}^e\mathbf{v}_t, {}^e\omega_{e_2}, {}^e\mathbf{ET})$ 
  end if
   $\text{Ker}(\mathbf{L}_{e_1}) \leftarrow \text{projection\_null\_space}({}^e\mathbf{u}_r, {}^e\mathbf{R}, \mathbf{u}_{e_1})$ 
   $\text{ker}({}^e\tau_e) \leftarrow \text{projected\_velocities}(\text{Ker}(\mathbf{L}_{e_1}), {}^e\tau_{e_2})$ 
   ${}^e\tau_{e_{\text{control}}} \leftarrow \text{send\_robot\_velocities}({}^e\tau_e + \text{ker}({}^e\tau_e))$ 
   $\text{update\_pose}({}^w\mathbf{M}_e, {}^e\tau_{e_{\text{control}}})$ 
end while

```

directions. When the tool is far from the reference curve, this resultant velocity equals that of projection direction in order to bring back the tool tip as fast as possible to the desired path. Otherwise, the resultant path following velocity is determined as in (30) and the parameter (α) is obtained in the latter case as follows:

$$\alpha = \sqrt{(\beta\|\mathbf{d}\|)^2 + \mathbf{v}_{\text{tissue}}^2} \quad (35)$$

Algorithm 2 Control loop for 3D path following

```

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

```

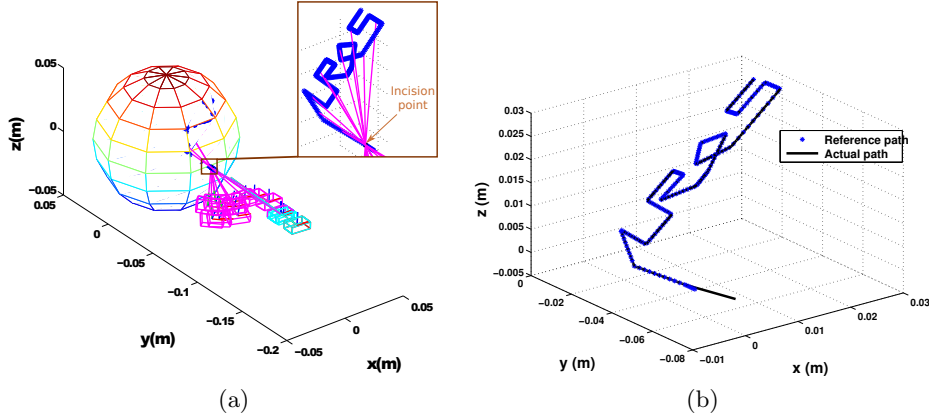


Fig. 4. (a) The end-effector motion during approach phase (cyan) and insertion phase (magenta); (b) the actual position of tool tip with respect to the desired path.

3.2 Results

A spherical workspace was chosen (Fig. 4.a) where the rigid tool navigated inside it. In fact, the fulcrum effect will produce a conical workspace within the spherical one. The desired 3D path could be defined in any form and it was constructed as shown in figure 4.b. In the latter figure displays the resultant tool tip position (black line) with respect to the reference path (blue dots).

Figure 5.a shows the RCM constraints error and path following error during the approaching phase and they decrease exponentially as designed. Figure 5.b presents the same errors during the insertion phase, where the RCM error is stable and the path following error has some picks but it stills smalls. Throughout the simulation, the standard deviation error of RCM constraints during the insertion phase is almost zero and that of path following is approximately $0.026mm$. These results are obtained with the parameters values $\lambda = 10$, $\gamma = 0.3$, $\beta = -10$, $v_{tissue} = 0.003m/sec$ and sampling time $0.1sec$.

The latter 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 6.a, the variables β and $v_{tissue} = 0.001m/sec$ are varied from $-1 * 10^{-6}$ to -10 and $0.1m/sec$ to $1 * 10^{-3}m/sec$, respectively. In Figure 6.b, the results are obtained by changing λ and γ from 2 to -0.1 and 2.5 to -0.1 , respectively.

4 CONCLUSIONS

This paper proposed a novel task priority controller in order to preform fulcrum effect with highest priority and path following task as second priority. It presented the geometric modelling of RCM constraints and path following scheme

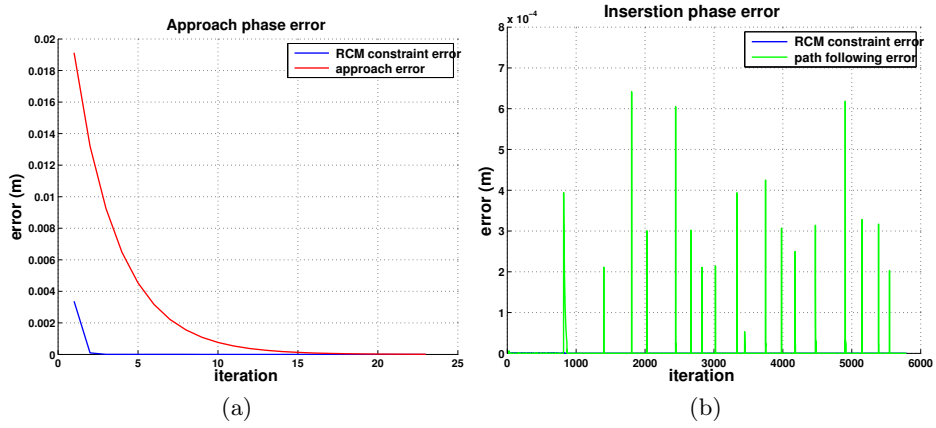


Fig. 5. Motion error during (a) approach phase and (b) path following.

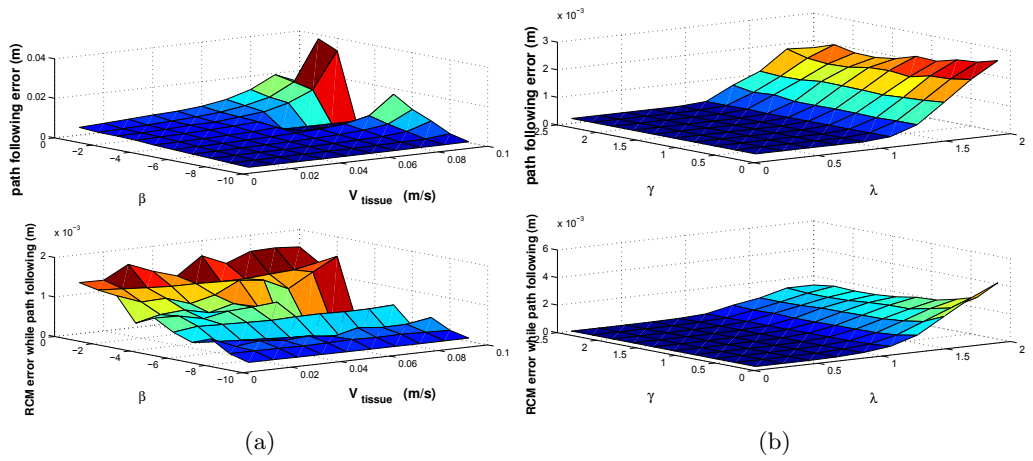


Fig. 6. Effect of system variables on the error (a) v_{tissue} vs β and (b) λ vs γ .

for deducing the control twist velocity. The proposed controller has a great flexibility to be integrated on any type of robot structure; since it is represented in task-space. It showed good results in terms of deviation error. This controller could be extended to handle curved tool. In addition, the RCM constraints may expand 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.

Appendix

4.1 Notation

The notations used within the paper are summarized in Table 2, for a better understanding.

References

1. Nastaran Aghakhani, Milad Geravand, Navid Shahriari, Marilena Vendittelli, and Giuseppe Oriolo. 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, 2013.
2. Hamidreza Azimian, Rajni V Patel, and Michael D Naish. On constrained manipulation in robotics-assisted minimally invasive surgery. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, pages 650–655, 2010.
3. Mahdi Azizian, Mahta Khoshnam, Nima Najmaei, and Rajni V Patel. 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, 2014.
4. Emad M Boctor, Robert J Webster III, Herve Mathieu, Allison M Okamura, and Gabor Fichtinger. Virtual remote center of motion control for needle placement robots. *Computer Aided Surgery*, 9(5):175–183, 2004.
5. Bassem Dahroug, Brahim Tamadazte, and Nicolas Andreff. 3d path following with remote center of motion constraints. In *13th International Conference on Informatics in Control, Automation and Robotics, ICINCO*, 2016.
6. L.-A. DufLOT, A. Krupa, B. Tamadazte, and N. Andreff. Towards ultrasound-based visual servoing using shearlet coefficients. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.
7. Janez Funda, Russell H Taylor, Benjamin Eldridge, Stephen Gomory, and Kreg G Gruben. Constrained cartesian motion control for teleoperated surgical robots. *IEEE Transactions on Robotics and Automation*, 12(3):453–465, 1996.
8. Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems*, pages 3–27. Springer, 2015.
9. Alexandre Krupa, Christophe Doignon, Jacques Gangloff, and Michel De Mathelin. 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, 2002.

Symbol	Description
\mathfrak{S}_W	world frame with the origin point W
\mathfrak{S}_B	robot base frame with the origin point B
\mathfrak{S}_E	end-effector frame with the origin point E
\mathfrak{S}_T	tool tip frame with the origin point T
\mathfrak{S}_R	RCM frame with the origin point R
\mathfrak{S}_C	camera frame with the origin point C
${}^w M_e$	homogeneous transformation matrix that describes the pose of \mathfrak{S}_E in \mathfrak{S}_W
${}^c \mathbf{v}_e$	linear velocity of \mathfrak{S}_E that is expressed in \mathfrak{S}_C
${}^c \boldsymbol{\omega}_e$	angular velocity of \mathfrak{S}_E that is expressed in \mathfrak{S}_C
${}^c \boldsymbol{\tau}_e$	velocity vector of \mathfrak{S}_E that groups its linear and angular velocities
$\mathbf{I}_{3 \times 3}$	identity matrix
${}^r \mathbf{y}$	the y -component of \mathfrak{S}_R
${}^r \mathbf{v}$	linear velocity of any point (subscript) that is expressed in \mathfrak{S}_R
${}^c \boldsymbol{\tau}_t$	linear velocity of \mathfrak{S}_T that is expressed in \mathfrak{S}_C
${}^e \mathbf{ER}$	vector between the origin points of \mathfrak{S}_E and \mathfrak{S}_R , expressed in \mathfrak{S}_E
${}^e \mathbf{u}_{er}$	unit vector of ${}^e \mathbf{ER}$ and expressed in \mathfrak{S}_E
${}^e \mathbf{v}_r$	linear velocity of \mathfrak{S}_R that is expressed in \mathfrak{S}_E
${}^e \mathbf{v}_e$	linear velocity of \mathfrak{S}_E that is expressed in its frame
${}^e \boldsymbol{\omega}_e$	angular velocity of \mathfrak{S}_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	the 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}

Table 2. symbols summary

10. Chin-Hsing Kuo, Jian S Dai, and Prokar Dasgupta. Kinematic design considerations for minimally invasive surgical robots: an overview. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 8(2):127–145, 2012.
11. Roderick CO Locke and Rajni V Patel. Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery. In *IEEE International Conference on Robotics and Automation*, pages 1900–1905, 2007.
12. Murilo M Marinho, Mariana C Bernardes, and Antônio PL Bó. 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, 2014.
13. Hermann Mayer, István Nagy, and Alois Knoll. Kinematics and modelling of a system for robotic surgery. In *On Advances in Robot Kinematics*, pages 181–190. Springer, 2004.
14. Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.
15. Takayuki Osa, Christoph Staub, and Alois Knoll. Framework of automatic robot surgery system using visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1837–1842, 2010.
16. Ali Oulmas, Nicolas Andreff, and Stéphane Régnier. Chained formulation of 3d path following for nonholonomic autonomous robots in a serret-frenet frame. In *American Control Conference (ACC)*, 2016.
17. Cong D Pham, Fernando Coutinho, Antonio C Leite, Fernando Lizarralde, Pal J From, and Rolf Johansson. 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, 2015.
18. Jean-Antoine Seon, Brahim Tamadazte, and Nicolas Andreff. Decoupling path following and velocity profile in vision-guided laser steering. *IEEE Transactions on Robotics*, 31(2):280–289, 2015.