

Visual Servoing Controller for Time-Invariant 3D Path Following with Remote Centre of Motion Constraint

Bassem Dahroug - Brahim Tamadazte - Nicolas Andreff

Abstract—The Remote Centre of Motion (RCM) is an essential movement for most of the medical robotic systems during the minimal invasive surgeries. In literature, there are many references which had analysed the techniques for modelling the problem of RCM constraint. However, only very few have discussed the RCM with Path Following Controller which ensures the complexity of surgical tasks. Therefore, this article is focusing on presenting the Task Priority Controller. Such task controller deploys exteroceptive sensor (visual servoing scheme) in order that the surgical tool follows the reference path while maintaining the RCM constraints. The experimental results showed good performance of the proposed controller.

Index Terms—Medical Robotics, Remote Centre of Motion, Path Following.

I. INTRODUCTION

Generally, a medical robot gets into the patient body from an incision point, the incision walls do limit the robot motion. Consequently, the surgical tool rotates around the three axes of the penetration point (i.e., the x -, y - and z -axis) and it is only permitted to translate along one direction. Such constrained motion caused by the incision walls is called the Remote Centre of Motion (RCM) or fulcrum effect. In order to overcome such constrained motion, there are two possible solutions either special robot kinematic structure or software controller. The first solution is to build a special robot structure that has at four three Degrees Of Freedom (DOF) to satisfy the RCM constraints. The RCM mechanisms [1] have the advantages of reducing potential hazard due to their kinematic structure that ensures the pivoting motion and their simple controller. However, they do not provide enough flexibility to change the location of penetration point.

On the opposite, the RCM software guides a general purpose robot that is generally redundant for performing different surgical tasks simultaneously into the patient body. For instance, the surgeon needs to reach specific Region Of Interest (ROI), for scanning an organ or removing tissues, while maintaining RCM and avoiding anatomical structures. Therefore, the software controller is the focal point of this article due to its flexibility for achieving complex tasks. It is necessary to set a priority list of the objectives. The task priority technique [2] was originally implemented for solving the redundancy issue. However, as shown by this paper, this technique is useful for defining complex tasks that are assembled by

several individual tasks (i.e., the RCM movement and the path following tasks).

During the robot motion control, there are basically two types of controllers for guiding the manipulator over a reference curve, either a path following or a trajectory tracking controllers. The difference between both controllers is the dependency on time. The trajectory tracking controller is considered as a time-dependent, where the reference curve is parametrized with time. On the contrary, the path following controller is time-invariant, where the curve is defined without any temporal constraints (using only xyz -coordinates). In fact, the exact time to finish the curve is not a factor of significant nature for the surgeon. What really matters is that the instrument follows exactly the intended curve. the surgeon may also need to change the tool velocity independently from the curve shape, size or curvature. The instrument velocity is indeed dependent on the tissue-tool interaction (deformation) which could be variable due to the potential inhomogeneity of the tissue. In addition, the tool velocity profile may change when the surgeon scans a ROI, or moves from one ROI to another for inspection. For the above reasons, the path following controller is a good option for our application; since it separates the geometric curve from the velocity profile.

This article discusses the RCM issue to implement a control software for performing fulcrum task along with time-invariant 3D path following based on visual servoing controller. Section II reviews the fulcrum effect and the path following scheme in the literature. After that, Section III shows the contribution of this paper by proposing the mathematical models of both issues (i.e., RCM and path following), and applying the projection gradient method for solving these tasks in hierarchical form. Then Section IV shows the experimental setup and discusses the obtained results by the proposed vision-based controller.

II. BACKGROUND

A. Kinematic Constraint of Fulcrum Effect

The kinematic description of fulcrum effect could be represented either in the joint-space [3] [4] or in the task-space [5] [6]. The joint-space representation needs a precise knowledge about the robot kinematic model and the robot/patient registration in order to map the required task into the joint-space. A proprioceptive sensor is usually used for the control feedback. In this case, the controller would not be able to detect and fix the error, since the error comes from the inverse static model

which is the input of the control loop. On the contrary, the task-space representation does not need an accurate knowledge of the robot model because the inverse differential model is integrated in the control loop. An exteroceptive sensor is commonly used as the control feedback. If the exteroceptive sensor is not accurate, it will affect the system performance. Therefore, there is a trade-off between the accuracy of robot mathematical model and that of sensor.

The authors in [4] deployed the projection gradient framework to propose a velocity controller that fulfils the fulcrum task and the trajectory tracking task. The model reported in [4] was extended in [7] for manipulating multiple links inside the incision hole. But they deployed the task augmented technique in order to determine the control velocity. This technique does not guarantee the hierarchical form. It does not give the priority for specific task. It attempts to find out a solution that satisfy both tasks at the same time. Therefore, the projection gradient technique is a better candidate compared to the task augmented method; since it ensures that the fulcrum task has the highest priority and the second task has a lower priority.

The authors in [5] proposed a stereo-vision controller to move a surgical tool during a laparoscope surgery. The geometric model of fulcrum effect is integrated in the Image-Based Visual Servoing (IBVS) control loop. However, the proposed controller was completed by a time-dependent point-to-point task.

Inspired by the cited works, a visual servoing controller with monocular camera will be deployed for performing the RCM movement with a rigid tool. The proposed controller in our article allows to execute the fulcrum effect as the highest priority. The RCM constraints are formulated by a geometric method in order to be easily described in the task-space which allows to overcome the lack of knowledge with respect to the robot kinematics.

B. Path Following Related Work

Most of the previously-reported works with respect to path following controller were implemented for mobile robot applications. Such works had used various techniques for solving the path following problem with non-holonomic constraints, such as robust non-linear controller [8], chain form [9] and backstepping filter based on Lyapunov [10]. Such controllers are working well with non-holonomic system (i.e., ground, marine and air vehicles) in 2D and 3D spaces. However, there are few works which had applied a path following controller for medical applications, especially those which are combining visual servoing with path following (e.g., [11] and [12]). Fewer works had discussed path following with RCM constraints [6].

Occasionally in the literature, we detect a confusion of terminology between the path following and the trajectory tracking. For instance, in [11], the authors are discussing the path following with a proposed controller functioning as time-variant controller with feed-forward controller.

In [6], we had proposed a 3D path following scheme for holonomic system which is characterized by a local frame in order to take advantage of the time-invariant controller. The proposed scheme deduces the tool tip velocity in the local

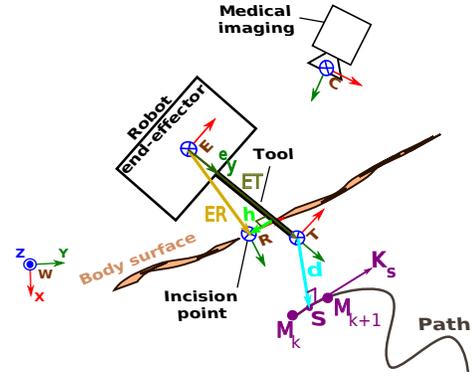


Fig. 1. Conceptual scheme of the system with the various reference frames (i.e., \mathfrak{S}_w world frame, \mathfrak{S}_e end-effector frame, \mathfrak{S}_t tool tip frame, \mathfrak{S}_r RCM frame, and \mathfrak{S}_c camera frame).

frame of the reference curve. This velocity is then transformed to the end-effector frame for determining its velocity while maintaining the RCM constrains.

III. PROPOSED METHOD

The proposed controller extends the previous work [6] by formulating the RCM error with one dimensional equation (Section III-A). The new formulation of RCM task allows to liberate more dimensions to satisfy the secondary task. Such secondary task is the path following scheme. The stability study of path following task is also presented (Section III-B). Finally, the task hierarchy is obtained by deploying the projection gradient technique (Section III-C).

A. Remote Center of Motion Modelling

The different reference frames used in the equations are shown in figure 1. The fulcrum effect restricts two dimensions of the tool linear motion (i.e., the translation along the x - and z -axis) and it allows only one dimension translation along the y -axis. Indeed, the RCM constraint ensures that the misalignment error between the tool centre line (assumed to be straight line) and the centre point of incision reference equals to zero. This error is measured as the perpendicular distance between the incision point and the line formed by the tool (h in Fig. 1).

Alternatively, this error is determined by the angular error between the tool vector (\mathbf{ET}) and the vector formed by the RCM point and any point on the tool (in our case, the tool base point (\mathbf{ER}) is chosen, i.e., $\mathbf{ET} \times \mathbf{ER}$ is intrinsic to the RCM constraint and decorrelates it from the path following task). This is basically the geodesic error between the two vectors and it was used in [6] for defining:

$$e_{RCM} = e_y \times e_{u_{er}} \quad (1)$$

whereby e_y is the y -component of the end-effector frame and $e_{u_{er}}$ is the unit-vector of \mathbf{ER} expressed in the end-effector frame.

This three dimensional error, which constrains 2-DOF, could reduce its dimension into one, if the formulation is expressed

in the appropriate reference frame. Moreover, the direction of the error is numerically ill-defined when the error goes to zero. As a consequence, we propose here the following one dimensional error:

$$e_{RCM} = 1 - {}^e\mathbf{y}^T {}^e\mathbf{u}_{er} \quad (2)$$

The advantage of this new form is liberating more DOF that will be used later to satisfy another task (i.e., the path following task) with the projection gradient method.

The time-derivative of misalignment error between the tool and the RCM point (2) is evaluated as:

$$\dot{e}_{RCM} = - {}^e\mathbf{y}^T {}^e\dot{\mathbf{u}}_{er} \quad (3)$$

The time-derivative of unit-vector ${}^e\mathbf{u}_{er}$ derives as [6]:

$${}^e\dot{\mathbf{u}}_{er} = \frac{1}{\|{}^e\mathbf{ER}\|} (\mathbf{I} - {}^e\mathbf{u}_{er} {}^e\mathbf{u}_{er}^T) {}^e\dot{\mathbf{ER}} \quad (4)$$

whereby $\mathbf{I}_{3 \times 3}$ is the identity matrix, and ${}^e\dot{\mathbf{ER}}$ represents the linear velocity vector of the RCM frame expressed in the end-effector frame (${}^e\dot{\mathbf{ER}} = {}^e\mathbf{v}_r$). The linear velocity of RCM point (${}^e\mathbf{v}_r$) is also defined in terms of end-effector linear velocity (${}^e\mathbf{v}_e$) and its angular velocity (${}^e\omega_e$), as:

$$\begin{aligned} {}^e\mathbf{v}_r &= -({}^e\mathbf{v}_e + {}^e\omega_e \times {}^e\mathbf{ER}) \\ &= -[\mathbf{I}_{3 \times 3} \quad -[{}^e\mathbf{ER}]_{\times}] \begin{bmatrix} {}^e\mathbf{v}_e \\ {}^e\omega_e \end{bmatrix} \end{aligned} \quad (5)$$

Back substituting (5) and (4) in (3) reformulates the geometric RCM constraint as a kinematic one:

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

whereby $\mathbf{L}_{e_{RCM}}^T$ is the interaction matrix of misalignment error with dimension 3×6 and ${}^e\tau_e$ is the end-effector twist vector. The null-space of the interaction matrix, which has a dimension of 6×5 , can be analytically computed as:

$$\begin{aligned} \ker(\mathbf{L}_{e_{RCM}}^T) &= \mathbf{I}_{6 \times 6} - \mathbf{L}_{e_{RCM}}^T \mathbf{L}_{e_{RCM}} \\ &= \begin{bmatrix} {}^e\mathbf{u}_{er} & \mathbf{k}_1 & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{k}_3 \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{k}_2 & {}^e\mathbf{u}_{er} & \mathbf{k}_1 \end{bmatrix} \end{aligned} \quad (7)$$

where $\mathbf{L}_{e_{RCM}} = (\mathbf{L}_{e_{RCM}}^T)^T$, $\mathbf{k}_1 = {}^e\mathbf{y} \times {}^e\mathbf{u}_{er}$, $\mathbf{k}_2 = {}^e\mathbf{u}_{er} \times {}^e\mathbf{y} \times {}^e\mathbf{u}_{er}$ and $\mathbf{k}_3 = \|\mathbf{ER}\| * \mathbf{k}_2$. The aim to use the latter equation (7) is to avoid subsequent costly numerical computation (e.g., Pseudo-inverse or singular value decomposition).

After having formulated the RCM constraint in the state form (6), the projection gradient will be used in order to find a solution for the state variable ${}^e\tau_e$.

B. 3D Path Following

The reference curve is defined as a set of points in the 3D space. The curve is simply represented in xyz -coordinates and it does not need to be parametrized in function of time or any other parameters. This is exactly what a surgeon defines the reference curve in a 3D navigation system/planner. The core of path following task is minimizing the projection distance vector ($\mathbf{d} = \mathbf{T} - \mathbf{S}$) (Fig. 1) which is the distance between

the tool tip in the 3D space (\mathbf{T}) and its closest point projected on the reference curve (\mathbf{S}). Then, the time-derivation of the projected distance ($\dot{\mathbf{d}}$) is the resultant velocity that brings back the tool tip to the reference curve. Such velocity is the difference between the tool tip velocity (\mathbf{v}_t) and the advanced velocity along the path (\mathbf{v}_s) [6]:

$$\begin{aligned} \dot{\mathbf{d}} &= \dot{\mathbf{T}} - \dot{\mathbf{S}} \\ &= \mathbf{v}_t - \mathbf{v}_s \end{aligned} \quad (8)$$

The advanced velocity ($\mathbf{v}_s = \dot{s}\mathbf{k}_s$) is decomposed into the required speed (\dot{s}) and the instantaneous tangential vector (\mathbf{k}_s) along the reference curve. The tool tip moves into the plane spanned by the two vectors (i.e., \mathbf{d} and \mathbf{k}_s). An orthogonal basis is formed from $\{\mathbf{d}, \mathbf{k}_s, \mathbf{d} \times \mathbf{k}_s\}$. In addition, the speed along the path (\dot{s}) could be represented in terms of i) the tool tip velocity (\mathbf{v}_t), ii) the path curvature ($\mathbf{C}(s)$), iii) the instantaneous tangential vector (\mathbf{k}_s), and iv) the projection distance (\mathbf{d}) as proposed in [6]:

$$\dot{s} = \frac{\mathbf{v}_t^T \mathbf{k}_s}{1 - \mathbf{d}^T (\mathbf{C}(s) \times \mathbf{k}_s)} \quad (9)$$

Back substituting (9) in (8), the time-derivative of the error becomes:

$$\dot{\mathbf{d}} = \left(\mathbf{I} - \frac{\mathbf{k}_s \mathbf{k}_s^T}{1 - \mathbf{d}^T (\mathbf{C}(s) \times \mathbf{k}_s)} \right) \mathbf{v}_t \quad (10)$$

The tool velocity profile ($\mathbf{v}_t(t)$) in (10) is free to be chosen and independently from path parameters. There is a possible control solution [6] which consists of choosing the tool velocity as the resultant velocity of two components:

$$\mathbf{v}_t = \alpha \mathbf{k}_s + \beta \mathbf{d} \quad (11)$$

The first term in the Right Hand Side (RHS) of equation (11) represents the required velocity to advance the tool on the path. The second term in the RHS of equation (11) represents the necessary velocity to bring back the tool on the path, if the tool is deviated from the path. The weighting coefficients (α and β) in equation (11) provide a priority choice between the two velocity components.

1) *Stability Condition:* Let us consider the following Lyapunov Candidate:

$$V = \frac{1}{2} \mathbf{d}^T \mathbf{d} \Rightarrow \begin{cases} V > 0 \text{ when } \mathbf{d} \neq \mathbf{0} \\ V = 0 \text{ when } \mathbf{d} = \mathbf{0} \end{cases} \quad (12)$$

In order to ensure the stability, the derivative of the latter equation (12) should be a negative scalar when $\mathbf{d} \neq \mathbf{0}$ and null when $\mathbf{d} = \mathbf{0}$.

$$\dot{V} = \frac{1}{2} (\mathbf{d}^T \dot{\mathbf{d}} + \dot{\mathbf{d}}^T \mathbf{d}) = \mathbf{d}^T \dot{\mathbf{d}} \quad (13)$$

By applying the controller (11) on the kinematics (10), the resultant velocity of projection distance becomes:

$$\dot{\mathbf{d}} = \alpha \left[1 - \frac{1}{1 - \mathbf{d}^T (\mathbf{C}(s) \times \mathbf{k}_s)} \right] \mathbf{k}_s + \beta \mathbf{d} \quad (14)$$

By replacing (14) in (13), \dot{V} is then reduced to (15), since \mathbf{k}_s is perpendicular to \mathbf{d} :

$$\dot{V} = \beta \mathbf{d}^T \mathbf{d} = 2\beta V \quad (15)$$

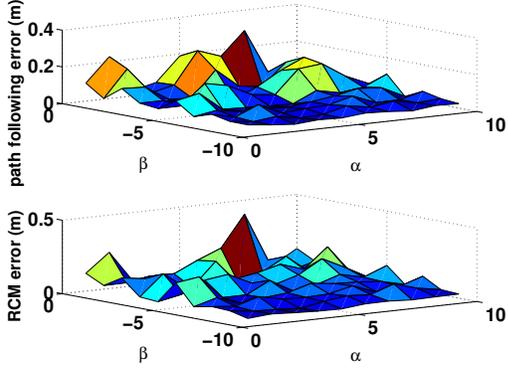


Fig. 2. Effect of parameters (α and β) on the system performance with respect to the path following error (upper figure) and the RCM error (lower figure).

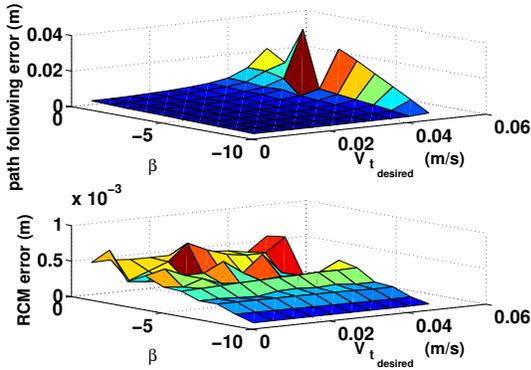


Fig. 3. Effect of parameters ($v_{t_{des}}$ and β) on the system performance in terms of the path following error (upper figure) and RCM error (lower figure).

In order to achieve asymptotic convergence, the coefficient (β) should be set as a negative coefficient. The parameter (α) does not effects the system stability but rather it changes its performance. Figure 2 shows the differences in performance with respect to the path following and the RCM errors. Each parameter is changed separately from the other, by fixing one and modifying the other, within the following ranges $-1 * 10^{-6} < \beta < -10$ and $1 * 10^{-6} < \alpha < 10$. The results in this figure were obtained by the simulator proposed in [6].

In order to ensure that the tool tip is not driven away from the reference path, α is then formulated in function of β and $v_{t_{des}}$ which represents the desired velocity along the path [6]:

$$\alpha = \begin{cases} \sqrt{\beta^2 \|\mathbf{d}\|^2 + v_{t_{des}}^2}, & \beta^2 \|\mathbf{d}\|^2 < v_{t_{des}}^2 \\ 0, & \beta^2 \|\mathbf{d}\|^2 > v_{t_{des}}^2 \end{cases} \quad (16)$$

The above representation shows a better performance (Fig. 3 compared to Fig. 2). Such good performance consists of the significant reduction of the errors of the path following and the RCM, whereby $v_{t_{des}}$ varying from 0.0001 to 0.05(m/s). These results are also obtained from the simulator.

C. Control Scheme

This part puts all together for implementing the task priority controller. For ensuring an exponential decrease of misalign-

ment error (6), the control law has the following form:

$$\dot{e}_{RCM} = -\lambda e_{RCM} \quad (17)$$

$$-\lambda e_{RCM} = \mathbf{L}_{e_{RCM}}^T e_{\tau_e} \quad (18)$$

The control solution of RCM problem is obtained by the projection gradient technique [2] [13] as:

$$e_{\tau_e} = -\lambda \mathbf{L}_{e_{RCM}}^{T+} e_{RCM} + [\mathbf{I} - \mathbf{L}_{e_{RCM}}^{T+} \mathbf{L}_{e_{RCM}}^T] \mathbf{b}_1 \quad (19)$$

where \mathbf{b}_1 is a free vector that is used to project the secondary task (i.e., path following task) in the null-space of fulcrum task.

The output of path following controller is the second task that should be represented in form of the vector (\mathbf{b}_1) for solving the state variable (e_{τ_e}). The tool tip velocity is determined by (11). It is described in the end-effector frame (${}^e \mathbf{v}_t$) and it is related with the end-effector velocity vector (${}^e \tau_e$) as:

$${}^e \mathbf{v}_t = \underbrace{[\mathbf{I} - [{}^e \mathbf{E} \mathbf{T}]_{\times}]}_{\mathbf{L}_d^T} \underbrace{\begin{bmatrix} {}^e \mathbf{v}_e \\ {}^e \omega_e \end{bmatrix}}_{e_{\tau_e}} \quad (20)$$

where \mathbf{L}_d^T is the interaction matrix associated to the path following task. Thus, the path following task would require that:

$$\mathbf{b}_1 = \mathbf{L}_d^{T+} e_{\mathbf{v}_t} \quad (21)$$

Back substituting (21) in (19), the resultant equation (22) is the control solution that satisfies both conditions (18) and (20), and arranges the two conditions in a task priority manner.

$$e_{\tau_e} = -\lambda \mathbf{L}_{e_{RCM}}^{T+} e_{RCM} + [\mathbf{I} - \mathbf{L}_{e_{RCM}}^{T+} \mathbf{L}_{e_{RCM}}^T] \mathbf{L}_d^{T+} e_{\mathbf{v}_t} \quad (22)$$

IV. EXPERIMENTAL VALIDATION

The system configuration used to test the proposed controller is shown in figure 4. The system includes a *SpaceFAB SF-3000 BS*¹ parallel robot (3PPSR) whose end-effector is holding a rigid tool that has cone shape with its base diameter of 2.4mm and its tip of 0.5mm. A monocular camera is also employed to detect the end-effector pose. A 3D printed part was also developed to simulate the workspace into which the tool navigates through a hole of diameter of 6mm.

A. Implementation Issues and Results

The software was implemented in C++ and using Visual Servoing Platform (ViSP) library [14] for camera calibration, image processing and pose estimation. The control algorithm is represented as a flow chart in figure 5. At the beginning, the camera parameters (i.e., intrinsic and extrinsic parameters) are estimated during the calibration phase in order to build its interaction matrix. Afterwards, the features extraction from the acquired image is done to initialize the end-effector pose and that of penetration point. In fact, the four points are deployed in order to estimate the instantaneous end-effector pose with

¹Addition information about 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/>

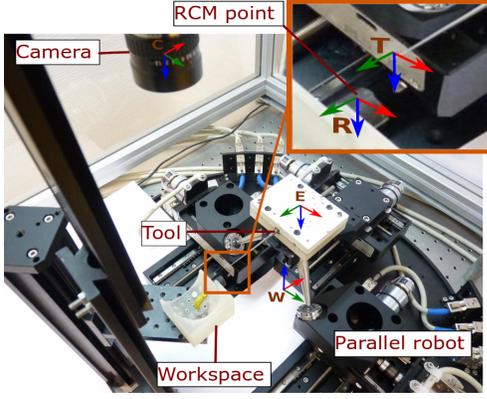


Fig. 4. Experimental system configuration with the various reference frames.

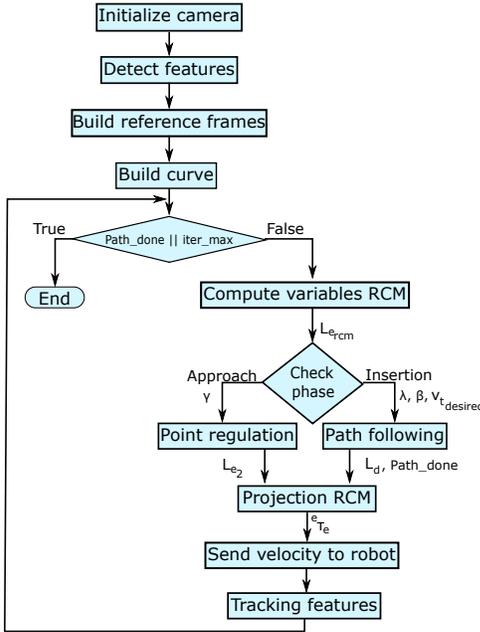


Fig. 5. Flow chart diagram of control algorithm.

respect to the camera reference frame by using Dementhon algorithm [15]. The pose of other reference frames could also be deduced by homogeneous transformation; since the end-effector pose is known.

Thereafter, the user defines the reference path and then the control algorithm performs its closed loop to guide the tool during two phases. The first phase is approaching to the incision point and the second one is inserting the tool through the hole in order for the tool tip to follow the desired path. Therefore, the loop starts with computing the RCM variables (i.e., (2), (6) and (18)), and then the program determine the tool state. If the tool is approaching to a point, the control law is similar to (18):

$$-\gamma \mathbf{e}_2 = \underbrace{[\mathbf{I} \quad -[{}^e\mathbf{ER}]_x]}_{\mathbf{L}_{e_2}^T} \underbrace{\begin{bmatrix} e_{v_e} \\ e_{\omega_e} \end{bmatrix}}_{e_{\tau_e}} \quad (23)$$

whereby γ is a positive gain coefficient for approaching to the RCM point, and $\mathbf{e}_2 = \mathbf{TR}$ is the distance between the

incision point and the tool tip. If the tool is inside the incision hole, the path following controller deduces the tool tip velocity to follow the reference path (11). The tangential vector \mathbf{k}_s is deduced from two consecutive points of reference curve (i.e., \mathbf{M}_k and \mathbf{M}_{k+1}):

$$\mathbf{k}_s = \frac{\mathbf{M}_{k+1} - \mathbf{M}_k}{\|\mathbf{M}_{k+1} - \mathbf{M}_k\|} \quad (24)$$

Afterwards, the projection gradient is computed to determine the control velocity of end-effector (22) in order to send it to the robot. At the end of the loop, the camera acquires new image that is treated to estimate the new pose of end-effector. The loop continues to execute these steps until the tool finishes the intended path or the number of iterations that is supposed to be saturated.

The choice of system gain coefficients (λ , γ , β and $v_{t_{des}}$) leads to change the system performance as mentioned above. The first two parameters (λ and γ) effect the angular and linear velocities of end-effector, respectively. The last two coefficients (β and $v_{t_{des}}$) determine the linear velocity components for bringing back and advancing, respectively, the tool tip to the path.

The reference path could be defined in any shape by xyz -coordinates, as the user so decides. The proposed controller was tested on two types of 3D curve: a zigzag path (Fig. 6) and a spiral path (Fig. 7). Figure 6(a) shows the difference between the reference zigzag path and the actual one that was done by the tool tip. Figure 6(b) shows the errors of RCM and path following during the two phases (approach and insertion phases). The errors are decreased exponentially as designed. On one hand, the RCM error is measured as the projected distance of the penetration point onto the centre line of tool body. The mean value of RCM error was approximately $0.557mm$ and its standard deviation was around $0.18mm$. On the other hand, the path following error is measured as the projected distance of the tool tip onto the reference geometric curve. The mean value of path following error was around $0.311mm$ and its standard deviation was about $0.17mm$. These results are obtained with $\lambda = 0.8$, $\gamma = 0.5$, $\beta = -1.25$ and $v_{t_{des}} = 0.75mm/second$.

In figure 7(a), we can observe that the tool tip does not follow the path with great accuracy, especially at the end of path. However, figure 7(b) shows that task priority is working well since the path following error increased while the RCM error remained constant. The mean errors of RCM and path following were measured about $0.356mm$ and $0.334mm$, respectively, during the insertion phase. While the standard deviation of RCM and path following were calculated around $0.158mm$ and $0.271mm$, respectively, during the same phase. These results are obtained with $\lambda = 0.8$, $\gamma = 0.5$, $\beta = -1.5$ and $v_{t_{des}} = 1mm/second$.

A small oscillation is also observed along the z -direction in both cases (zigzag and spiral paths) due to the error of pose estimation with four points. A grid with more points, which are not located in the same plan, is a possible solution in order to increase the pose estimation accuracy.

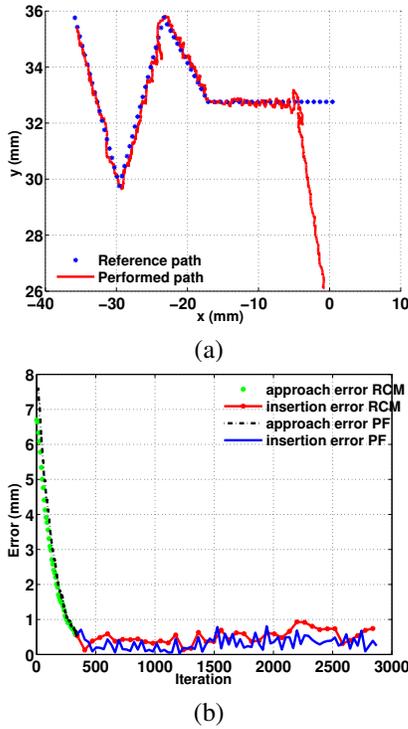


Fig. 6. (a) Comparison between the reference zigzag path and the performed one, (b) the error of RCM and path following along the robot motion.

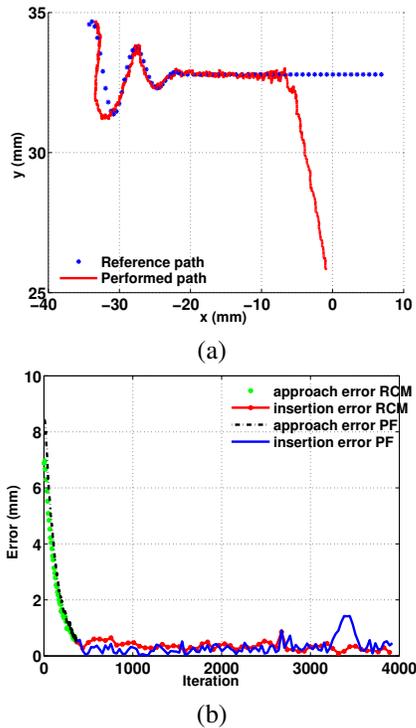


Fig. 7. Task priority verification: (a) comparison between the reference helical path and the performed one, (b) the error of RCM and path following along the robot motion.

V. CONCLUSION

This article presents a visual servoing controller for performing fulcrum task as the highest priority and 3D path following task as lower priority. The input for the controller is images from monocular camera and the output is the velocity of end-effector in the task-space. This controller has the advantage to be integrated easily into any generic-purposely robot for executing the desired tasks. The experimental results show good performance to maintain RCM constrains and to follow the reference path. However, the proposed controller could be improved by introducing a robust control. Such controller could also be extended to handle a flexible tool for increasing the reachable workspace.

REFERENCES

- [1] C.-H. Kuo, J. S. Dai, and P. Dasgupta, "Kinematic design considerations for minimally invasive surgical robots: an overview," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, no. 2, pp. 127–145, 2012.
- [2] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [3] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.
- [4] H. Azimian, R. V. Patel, and M. D. Naish, "On constrained manipulation in robotics-assisted minimally invasive surgery," in *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob) 2010*, 2010, pp. 650–655.
- [5] 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 (IROS) 2010*, 2010, pp. 1837–1842.
- [6] B. Dahroug, B. Tamadazte, and N. Andreff, "3d path following with remote center of motion constraints," in *13th International Conference on Informatics in Control, Automation and Robotics, ICINCO*, vol. 1, 2016, pp. 84–91.
- [7] N. Aghakhani, M. Geravand, N. Shahriari, M. Vendittelli, and G. Oriolo, "Task control with remote center of motion constraint for minimally invasive robotic surgery," in *IEEE International Conference on Robotics and Automation (ICRA) 2013*, 2013, pp. 5807–5812.
- [8] L. Lapiere and B. Jouvencel, "Robust nonlinear path-following control of an auv," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 89–102, 2008.
- [9] C. Samson, "Control of chained systems application to path following and time-varying point-stabilization of mobile robots," *IEEE Transactions on Automatic Control*, vol. 40, no. 1, pp. 64–77, 1995.
- [10] B. Zhu and W. Huo, "3-d path-following control for a model-scaled autonomous helicopter," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1927–1934, 2014.
- [11] F. Nageotte, P. Zanne, C. Doignon, and M. de Mathelin, "Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2364–2369.
- [12] A. Oulmas, N. Andreff, and S. Régnier, "Chained formulation of 3d path following for nonholonomic autonomous robots in a serret-frenet frame," in *American Control Conference (ACC)*, 2016.
- [13] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.
- [14] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.
- [15] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International journal of computer vision*, vol. 15, no. 1-2, pp. 123–141, 1995.