2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Autonomous Endoscope Control Algorithm with Visibility and Joint Limits Avoidance Constraints for da Vinci Research Kit Robot

Rocco Moccia[1], Fanny Ficuciello[1]

Abstract— **This paper presents a novel autonomous endoscope control method for the dVRK's Endoscopic Camera Manipulator (ECM), which allows the camera to track the surgical instruments on the Patient Side Manipulator (PSM). An Image-based Visual Servoing (IBVS) is enforced by the addition of a visibility constraint that ensures the identified surgical tool remains in the camera's Field Of View (FOV) for the continued availability of image feedback and a joint limits avoidance constraint that prevents the ECM from exceeding its joint limits. The work relies on an optimization approach, with constraints performed using the Control Barrier Functions concept (CBFs). The goal is to minimize the surgeon's cognitive and physical workload by removing the time-consuming job of camera reorientation, offering an enforced method compared to the traditional IBVS endoscopic camera controller.**

I. Introduction

In minimally invasive surgery, images are the main feedback the surgeon can use to correctly operate on the surgical site, and endoscopes play a crucial role in this scenario. Typically, a human assistant manually manipulates the endoscope, which adds challenges caused by unavoidable fatigue, resulting in unstable and poor camera performance. Surgical robots introduce significant improvements to classical minimally invasive surgical procedures [1]. In particular, the da Vinci®robotic system is the most widely used robotic system for Robot-assisted Minimally Invasive Surgery (MIRS). The da Vinci robot frequently employs a stereo endoscope, which provides a view of the surgical site, processed by a vision cart, to the surgeon through a stereo visor, with the tips of the tools aligned with the surgeon's hands, which grasp the master controllers. The endoscope is mounted on a robotic arm, called the Endoscopic Camera Manipulator (ECM). During the surgical procedure, the surgeon can control the ECM using a pedal clutch interface, which allows switching between the tools and camera control. Despite all these improvements, some disadvantages remain. In the absence of a human assistant, the surgeon is forced to control both the tools and the camera, manually switching between the two. The surgeon must frequently stop the operations and moves the camera to optimize the Field-Of-View (FOV) during the transitions between different tasks. Therefore, this act can introduce cognitive workload and time lost, which can significantly compromise the effectiveness of the procedure [2]. Moreover, the limited camera FOV poses other possible risks. In particular, the surgical tool can easily go out of focus, increasing the possibility of tissues damages. To overcome these limitations, robotic research is advancing toward the development of autonomous endoscope movements, improving camera control to minimize procedure time and surgical effort [3]. The paper in [4] offers a practical overview of the camera control algorithm recently implemented on the da Vinci Research Kit (dVRK) endoscope, an open-source mechatronic platform, constituted by the first generation of the da Vinci®robot and specifically used for academic purposes. One of the first approaches to autonomous camera control for the da Vinci surgical system is proposed in [5]. The strategy helps the surgeon eliminate unneeded movements by maintaining the view centered on the surgical instruments of the PSM. Nevertheless, camera calibration, which uses kinematics data to establish the tool's 3D location and converts it to 2D camera pixel coordinates, has a significant impact on the process. The authors in [6] introduce the concept of visual servoing for camera control of the dVRK, keeping a tracked surgical instrument within the bounds of the endoscopic camera image. The method in this case reconstructs the scene's 3D point cloud using stereo camera calibration to determine the image's coordinates and measures the tool's 3D positions using ArUco markers. The authors in [7] describe the SCAN (Technology for Camera Autonomous Navigation) system, which continuously moves the scene by tracking the tools and adjusting the endoscope's position accordingly. The view is defined by the scene center modality that the surgeon selects using a foot pedal. The study in [8] investigates Learning from Demonstrations (LfD), which uses endoscopic images seen during expert demonstrations to produce a predictive model that recommends camera positions at the start of surgery. All of the aforementioned approaches employ information from images to identify and track the visibility target, which is then used to control the ECM. Visual servoing has a fundamental requirement: the visual target must stay in the camera's image plane during the entire operation. Visibility violations can occur due to the presence of occlusions, especially if the visual feature is already near the image plane's border, or from a non-optimized trajectory that is built without considering the target's visibility. Moreover, the loss of target visibility may allow the robot to approach joint limits, causing additional problems. The visual target's visibility is crucial for giving real-time feedback on the target of interest that is utilized to create control inputs. As a result, explicitly accounting for the visibility problem in controller design is one technique to increase the robustness of visual servoing. This paper introduces a new autonomous endoscope control algorithm for the dVRK's ECM, which allows the camera to monitor

[1]The authors are with the Department of Information Technology and Electrical Engineering, Università degli Studi di Napoli Federico II, 80125 Napoli, Italy. Corresponding author's email: rocco.moccia@unina.it.

the PSM surgical instruments. The proposed method claims to provide a high-quality autonomous camera control, standing above the traditional visual control. The methodology employs the Image-based Visual Servoing (IBVS), which is enforced by the addition of two constraints that ensure the identified surgical instrument remains in FOV for the continued availability of image input and prevents the ECM from exceeding its joint limits. The method uses an optimization-based approach with constraints definition and execution achieved utilizing the concept of Control Barrier Functions (CBFs). CBFs have recently been utilized successfully to guarantee the safety of robots in dynamic environments, from applying dynamically-feasible limits, obstacle avoidance, and inter-collision avoidance for swarms of robots to adaptive cruise control of autonomous vehicles [9], [10], [11], [12]. By applying Lyapunov-like functions, the CBF-based technique is used to provide the control input enforcing the desired efficiency and safety properties into the behavior of the controlled robot [13]. The method in this work makes direct use of laparoscopic images captured by the dVRK endoscopic camera. The PSM tool semantic segmentation is accomplished using the deep learning approach presented in [14]. The information is then used to define an IBVS controller, configured as a nominal controller. The visibility and the joint limits avoidance constraints are defined using the CBFs technique. Ultimately, a convex optimization problem minimally modifies the nominal controller to satisfy the given constraints. The method is then rigorously tested on the ECM of the dVRK robot and in a simulation environment utilizing CoppeliaSim software, developed in [15], [16], [17].

II. Nominal Image-based Visual Servoing Controller

The kinematics of the dVRK's ECM and the basic unconstrained IBVS controller are described in this section.

A. System description

The ECM is a 4 Degrees of Freedom (DoFs) actuated arm moving about a Remote Center of Motion (RCM) with the following sequence RRPR, with R as revolute and P as prismatic joints. The base frame $\mathcal{F}_b$ is chosen as coincident with the RCM point with $\boldsymbol{q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_4]$ as the generalized joint coordinates and kinematic description shown in Fig. 1. The ECM end-effector pose, described by the homogeneous transformation $\boldsymbol{T}_{\boldsymbol{ee}}^{\boldsymbol{b}}$ between end-effector frame $\mathcal{F}_{ee}$ and RCM frame $\mathcal{F}_b$, can be computed using the Denavit-Hartenberg (DH) convention (Table I, with $l_1 = 0.3822\ m$ and $l_2 = 0.3829\ m$) to the kinematic chain. Moreover, joint limits are: $\boldsymbol{q}_{max} = [90°, 65°, 0.235\ m, 90°]$ and $\boldsymbol{q}_{min} = [-90°, -45°, 0.0\ m, -90°]$. To connect the camera frame $\mathcal{F}_c$ with the end-effector frame $\mathcal{F}_{ee}$, another transformation $\boldsymbol{T}_{\boldsymbol{c}}^{\boldsymbol{ee}}$ is required.

B. Nominal IBVS control law

The camera is placed on the manipulator in an eye-in-hand configuration. The reference frame $\mathcal{F}_c$ is collinear with the camera's optical axis, and its origin is located at the

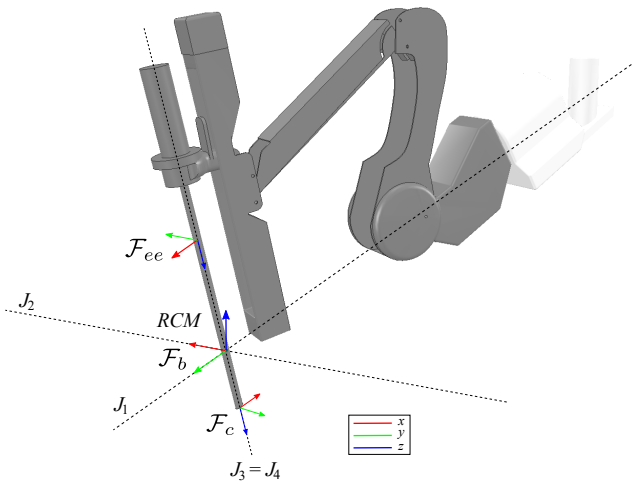


Fig. 1: ECM arm kinematics and frame definition. $J_1, \ldots, J_4$ represent the actuation axes of the ECM. $\mathcal{F}_b$: RCM base frame; $\mathcal{F}_{ee}$: end-effector frame; $\mathcal{F}_c$: camera frame.

TABLE I: DH parameters of ECM arm

link	joint	a_i	α_i	d_i	θ_i	offsets
1	R	0	$\pi/2$	–	q_1	$\pi/2$
2	R	0	$-\pi/2$	–	q_2	$-\pi/2$
3	P	0	$\pi/2$	q_3	–	$-l_1$
4	R	0	0	–	q_4	l_2

camera's principal point. The technique described in [14] provides the identification of the PSM tooltip. A deep learning technique identifies a visual feature vector $\boldsymbol{s}(t) \in \mathbb{R}^{2m}$, made of $m = 5$ image plane points, and then an Extended Kalman Filter (EKF) completes the tracking process. The principal point's coordinates $(\boldsymbol{c}_x, \boldsymbol{c}_y)$ and the pixel size to focal length ratios $(\boldsymbol{f}_x, \boldsymbol{f}_y)$ are the intrinsic camera parameters. The 2D projection of the PSM tooltip ($\boldsymbol{u}$, $\boldsymbol{v}$ in pixels) on the image plane can be calculated as follows:

$$\begin{cases} \boldsymbol{X} = (\boldsymbol{u} - \boldsymbol{c}_x) \,/\, \boldsymbol{f}_x, \\ \boldsymbol{Y} = (\boldsymbol{v} - \boldsymbol{c}_y) \,/\, \boldsymbol{f}_y. \end{cases} \tag{1}$$

Consequently, the time-varying visual feature and its velocity in the image plane are: $\boldsymbol{s}(t) = [\boldsymbol{X}, \boldsymbol{Y}]$ and $\dot{\boldsymbol{s}}(t)$. The IBVS control law can be built so that the image space error goes to zero asymptotically: $\boldsymbol{e}(t) = \boldsymbol{s}_d - \boldsymbol{s}(t)$, where $\boldsymbol{s}_d$ is the target feature parameters vector. The error vector's derivative may be written in terms of camera velocity as: $\dot{\boldsymbol{e}}(t) = \boldsymbol{L}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{z}_c)\boldsymbol{\varepsilon}^c$, where $\boldsymbol{\varepsilon}^c$ is the linear and angular velocity of the camera reference frame $\mathcal{F}_c$. While, $\boldsymbol{L}_{\boldsymbol{s}} \in \mathbb{R}^{2m \times 6}$ is the interaction matrix, depending on vector $\boldsymbol{s}$ and the depth of the camera feature points $\boldsymbol{z}_c$ in $\mathcal{F}_c$. Therefore, the interaction matrix of a set of points is $\boldsymbol{L}_{\boldsymbol{s}} = [\boldsymbol{L}_{\boldsymbol{s_1}}, \ldots, \boldsymbol{L}_{\boldsymbol{s_m}}]^T$, with $\forall i = 1, \ldots, m$:

$$\boldsymbol{L}_{\boldsymbol{s_i}} = \begin{bmatrix} -\frac{1}{\boldsymbol{z}_c} & 0 & \frac{\boldsymbol{X}}{\boldsymbol{z}_c} & \boldsymbol{XY} & -(1+\boldsymbol{X}^2) & \boldsymbol{Y} \\ 0 & -\frac{1}{\boldsymbol{z}_c} & -\frac{\boldsymbol{Y}}{\boldsymbol{z}_c} & (1+\boldsymbol{Y}^2) & -\boldsymbol{XY} & -\boldsymbol{X} \end{bmatrix}. \tag{2}$$

As a result, the camera velocity $\boldsymbol{\varepsilon}^c$ is obtained as:

$$\boldsymbol{\varepsilon}^c = -\boldsymbol{L}_s^*(\boldsymbol{s}, \boldsymbol{z}_c)\boldsymbol{K}_s\boldsymbol{e}, \tag{3}$$

where $\boldsymbol{K}_s$ is a positive definite matrix and $\boldsymbol{L}_s^*(\boldsymbol{s}, \boldsymbol{z}_c) \in \mathbb{R}^{6\times 2m}$ is the left pseudo-inverse matrix of the interaction matrix $\boldsymbol{L}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{z}_c)$. Therefore, assuming the reference velocity in joint space $\dot{\boldsymbol{q}}_n$, the unconstrained IBVS controller is provided by [18]:

$$\dot{\boldsymbol{q}}_n = \boldsymbol{J}^*(\boldsymbol{q}) \begin{bmatrix} \boldsymbol{R}_c & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_c \end{bmatrix} \boldsymbol{\varepsilon}^c, \tag{4}$$

where $\boldsymbol{J}^*(\boldsymbol{q}) \in \mathbb{R}^{4\times 6}$ is the left pseudo-inverse matrix of the the ECM Jacobian, $\boldsymbol{R}_c \in \mathbb{R}^{3\times 3}$ as the rotation matrix of $\mathcal{F}_c$ with respect to $\mathcal{F}_b$, and $\boldsymbol{\varepsilon}^c$ is the camera twist, made of linear and angular velocities, expressed in $\mathcal{F}_c$. The previously established control law allows the ECM camera to be aligned to the PSM tooltip. Nevertheless, if the visual target moves out from the camera's FOV, the controller will unavoidably fail. The CBFs technique, as described in the Section III, allows us to alter the IBVS control law such that the PSM tooltip projection is always within the camera FOV and simultaneously avoiding ECM joint limits during the movement.

III. Control Barrier Functions Approach

This section focuses into the CBF principles and exposes the constrained IBVS control law, which is implemented utilizing visibility and joint limit avoidance constraints.

A. CBFs Approach

The CBF-based method is frequently employed to guarantee the safety of a dynamic system. The safety is achieved by maintaining a task variable in a specified set $\mathcal{C}$, ensuring the set's forward invariance, and enabling the system to remain in the safe set at all times. In particular, the CBFs proprieties of a smooth function $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}$ ensure the forward invariance of its super-level set and the accomplishment of the related task [9]. A robot with n-DoFs is modeled as a non-linear affine control system: $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{\theta}$, where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the system state, $\boldsymbol{\theta} \in \mathcal{U} \subseteq \mathbb{R}^p$ is the control input in a set of admissible control values $\mathcal{U}$, f and g are Lipschitz-continuous vector fields. From definition of CBFs in [9], a continuously differentiable function $\boldsymbol{h} : \mathcal{X} \subseteq \mathbb{R}_{\geq 0} \to \mathbb{R}$ with a super-level set $\mathcal{C} \subset \mathcal{X}$ is defined as a time-varying control barrier function if there exists a Lipschitz continuous extended class $\mathcal{K}_\infty$ function γ for all $\delta \in \mathcal{X}$, such that:

$$\sup_{\boldsymbol{\theta}\in\mathcal{U}} \overbrace{\left\{ \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}} g(\boldsymbol{x})\boldsymbol{\theta} \right\}}^{\frac{\delta \boldsymbol{h}(\boldsymbol{x},\boldsymbol{\theta})}{\delta \boldsymbol{q}}} \geq -\gamma(\boldsymbol{h}). \tag{5}$$

This condition states that by applying the restrictions on the control input, CBFs guarantee forward invariance and asymptotic stability of the set $\mathcal{C}$. Any Lipschitz continuous controller $\dot{\boldsymbol{\theta}}$ that meets the criterion (5) for the robotic system yields the set $\mathcal{C}$ forward invariant if $\boldsymbol{h}$ is a CBF on $\mathcal{X}$. The set $\mathcal{C}$ is asymptotically stable in $\mathcal{X}$ as well. Finally, assuming the robot already has a nominal control input, the minimum variation of the control input subject to a set of constraints enforcing task execution derived by solving the following optimization problem posed to synthesize a velocity controller $\boldsymbol{\theta}^*$:

$$\begin{aligned} \underset{\boldsymbol{\theta}\in\mathcal{U}}{\text{minimize}} \quad & \|\boldsymbol{\theta} - \boldsymbol{\theta}_n\|^2 \\ \text{subject to} \quad & \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}} g(\boldsymbol{x})\boldsymbol{\theta} \geq -\gamma(\boldsymbol{h}). \end{aligned} \tag{6}$$

The γ function aims to modulate CBF activity depending on whether more aggressive or cautious behavior is necessary. The $\gamma > 0$ in this work is a scalar variable, and the nominal controller is less limited for large γ values, so the CBF is less conservative.

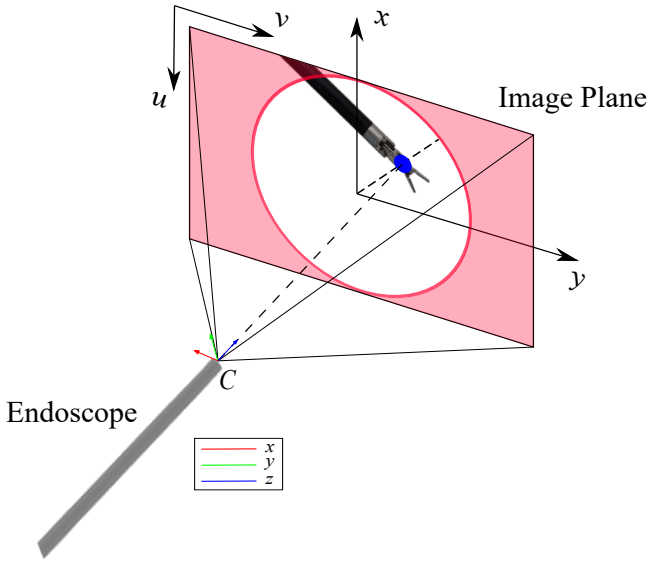


Fig. 2: The endoscopic camera and its image plane. The blue point identifies the surgical instrument. The forbidden region is highlighted in red, while the image FOV is represented in white.

B. Constrained IBVS control law

The ECM robot has 4 Dofs and its kinematic model is considered, assuming that its joint velocities are controllable. In this case, the state $\boldsymbol{x}$ is the vector of joint coordinates $\boldsymbol{q}$, the input $\boldsymbol{\theta}$ is the vector of joint velocities $\dot{\boldsymbol{q}}$ and the nominal controller $\boldsymbol{\theta}_n$ is the IBVS control law $\dot{\boldsymbol{q}}_n$. The visibility constraint is achieved by specifying a FOV with a circle centered on the image and a radius of $r = H/2$, where H is the image height in pixels. As a result, the FOV in this paper is different from the full image plane, resulting in a more stringent safety criterion. The constraint is then achieved by requiring the visual points associated with the PSM to remain within the safe-set circle. The safe-set is represented by any point within the FOV circle, and the minimum distance between each feature and the circle is calculated, defining the circle points in polar coordinates as illustrated in Fig. 2. The $\boldsymbol{h}$-functions $\boldsymbol{h}_s \in \mathbb{R}^{1\times 4}$ is defined as follows $\forall i = 1, \ldots, m$:

$$\boldsymbol{h}_{s,i} = -\frac{1}{2}||\boldsymbol{s} - \boldsymbol{s}_{\boldsymbol{e}}||^2, \tag{7}$$

where, $\boldsymbol{s}_{\boldsymbol{e}}$ is the point on minimum distance on the FOV circle. This enables the computation of the $\boldsymbol{h}$-function derivative

with respect to $\boldsymbol{q}$:

$$\frac{\delta \boldsymbol{h}_{s,i}}{\delta \boldsymbol{q}} = \frac{\delta \boldsymbol{h}_{s,i}}{\delta \boldsymbol{s}} \boldsymbol{J}_L(\boldsymbol{s}, \boldsymbol{z}_c, \boldsymbol{q}) = 2(\boldsymbol{s} - \boldsymbol{s_e})^T \boldsymbol{J}_L(\boldsymbol{s}, \boldsymbol{z}_c, \boldsymbol{q}). \quad (8)$$

where $\boldsymbol{J}_L(\boldsymbol{s}, \boldsymbol{z}_c, \boldsymbol{q})$ is defined as:

$$\boldsymbol{J}_L(\boldsymbol{s}, \boldsymbol{z}_c, \boldsymbol{q}) = \frac{\delta \boldsymbol{s}}{\delta \boldsymbol{q}} = \boldsymbol{L_s}(\boldsymbol{s}, \boldsymbol{z}_c) \begin{bmatrix} \boldsymbol{R}_c^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_c^T \end{bmatrix} \boldsymbol{J}(\boldsymbol{q}). \quad (9)$$

In the same manner, the joint limits avoidance is accomplished by forcing joint variables to stay within an upper and a lower bound. The $\boldsymbol{h}$-functions $\boldsymbol{h} \in \mathbb{R}^{4\times 4}$ and for each joint $i = 1, \ldots, 4$:

$$\boldsymbol{h}_{lim,i} = \frac{(\boldsymbol{q_{max,i}} - \boldsymbol{q_i})(\boldsymbol{q_i} - \boldsymbol{q_{min,i}})}{(\boldsymbol{q_{max,i}} - \boldsymbol{q_{min,i}})^2}, \quad (10)$$

where, $\boldsymbol{q_i}$ is the i-th joint variable, $\boldsymbol{q_{max,i}}$ and $\boldsymbol{q_{min,i}}$ are the upper and lower bound of the i-th joint. In this case:

$$\frac{\delta \boldsymbol{h}_{lim,i}}{\delta \boldsymbol{q}} = \frac{(\boldsymbol{q_{max,i}} + \boldsymbol{q_{min,i}} - 2\boldsymbol{q_i})}{(\boldsymbol{q_{max,i}} - \boldsymbol{q_{min,i}})^2}. \quad (11)$$

Finally, the optimization problem that determines the minimum variation of the IBVS controller required to complete the two tasks may be expressed as follows:

$$\begin{aligned} &\underset{\dot{\boldsymbol{q}}}{\text{minimize}} \quad \|\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_n\|^2 \\ &\text{subject to} \quad \boldsymbol{A}(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{q}, t))\dot{\boldsymbol{q}} \geq \boldsymbol{b}(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{q}, t)), \end{aligned} \quad (12)$$

where, $\boldsymbol{A} \in \mathbb{R}^{(4+m)\times 4}$ and $\boldsymbol{b} \in \mathbb{R}^{(4+m)\times 1}$ defined as:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\delta \boldsymbol{h}_{lim}}{\delta \boldsymbol{q}} \\ \frac{\delta \boldsymbol{h}_s}{\delta \boldsymbol{q}} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} -\gamma(\boldsymbol{h}_{lim}) \\ -\gamma(\boldsymbol{h}_s) \end{bmatrix}. \quad (13)$$

The formulation proposed is indeed a quadratic programming (QP) problem. Furthermore, the optimization problem may be solved relatively quickly thanks to its convexity. Because of this, this method also works well in applications with strict real-time needs.

IV. Experimental Validation

The experimental validation for the proposed methodology is discussed in this section. It provides a description of the protocol and a discussion of the outcomes, which are acquired on the dVRK robot while taking into account the ECM kinematic description given in Section II-A.

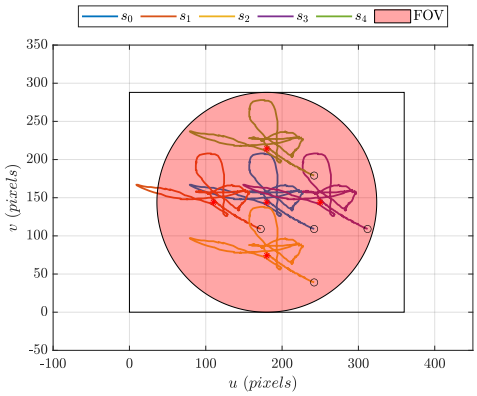
A. Experimental protocol and results

The same experiments are performed on the CoppeliaSim physics simulators and then recreated on the real dVRK robotic platform. The open-source control software, developed in [19], [20], used to connect with the dVRK robot, is based on the Robotics Operating System (ROS) software layer. While the robot model is handled on the CoppeliaSim scene using embedded scripts and ROS topics for simulation testing, the optimization problem is addressed using IBM ILOG CPLEX. The kinematics data from the dVRK are recorded at 200 Hz, while the tool location is approximated by the vision-based system at a camera frame rate of 25 Hz. The EKF method overcomes this limitation by giving tool poses at 200 Hz. To speed up processing, the tool segmentation is conducted utilizing GPU implementation on an NVIDIA®GTX 1080 Ti, as demonstrated in [14]. The dVRK ROS control layer provides the camera calibration matrix, which includes principal point coordinates and pixel size to focal length ratios. During the tests, the CBF method's γ value is empirically set at $\gamma = 10.0$. The session is divided into two phases, each beginning with the definition of 5 points and the related feature vector $\boldsymbol{s}$ on the image plane collected via surgical tool detection, as illustrated in Fig. 3. The PSM is moved in teleoperation mode using the dVRK console. A desired trajectory is then recorded using the desired feature vector $\boldsymbol{s}_d$, corresponding to the image center, as:

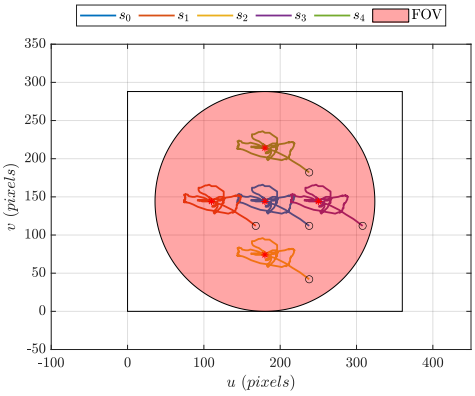
Fig. 3: The Figure shows the original frame, the related binary mask, and the point identification in the image plane.

$$\boldsymbol{s}_d = \begin{bmatrix} 0 & -0.224 & 0 & 0.224 & 0 \\ 0 & 0 & -0.224 & 0 & 0.224 \end{bmatrix}. \quad (14)$$

The interaction matrix $\boldsymbol{L_s}$ depends on $\boldsymbol{s}$ and $\boldsymbol{z}_c$, with the variable $\boldsymbol{z}_c$ as the third coordinate vector of each feature point, which is estimated from surgical tool tracking and homogeneous transformation between PSM and ECM base frames. The FOV circle has a radius of $r = 144$ pixels and is centered in the image plane. The first evaluation test ("Unconstrained IBVS control test") uses a standard IBVS control that, as shown in Section II-B, asymptotically decreases the image space error $\boldsymbol{e}(t) = \boldsymbol{s}_d - \boldsymbol{s}(t)$ to zero, moving the tool to the image center. The second evaluation test ("CBFs-constrained IBVS control test") considers the same desired trajectory, but the ECM control law adopts the suggested CBFs technique, keeping the chosen features inside the FOV and avoiding ECM joint limits, as indicated in Section III-B. The feature trajectories on the image plane are shown in Fig. 4(a) and Fig. 4(b). The red circle denotes the FOV, while the black lines represent the boundaries obtained from the image size. Figure 4(a) shows the pixel trajectories for "Unconstrained IBVS control test" using only the nominal IBVS control law and no constraints. Figure 4(b) depicts the pixel trajectories for "CBFs-constrained IBVS control test", using the same $\boldsymbol{s}_d$ and the restricted IBVS control law with visibility and joint limit avoidance constraints. The experimental results for "Unconstrained IBVS control test" are summarized in Fig. 5. The figure, in particular, describes the ECM joint values during the test, taking into consideration the joint restrictions for the first two joints. It displays the error norm as well as the produced ECM camera velocities as a consequence of the IBVS controller. Figure 6 summarizes the experimental results for "CBFs-constrained IBVS control test". The figure illustrates the ECM joint positions during



(a) Unconstrained IBVS control test



(b) CBFs-constrained IBVS control test

Fig. 4: Pixel trajectories with the image size shown by the black box. The $*$ markers denote the intended position, whereas the o denote the position of the starting features.

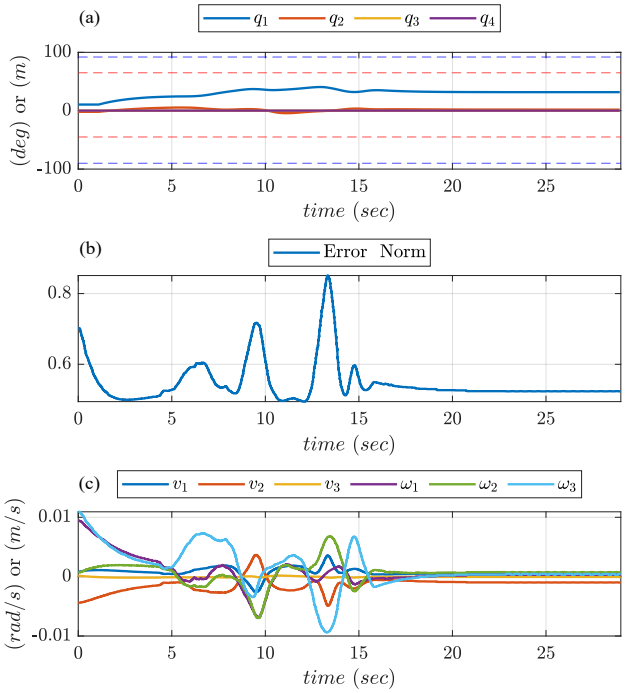


Fig. 5: Unconstrained IBVS control test: (a) joint position values and joint limits for the first two joint: $\boldsymbol{q}_{1lim} = [90°, -90°]$, $\boldsymbol{q}_{2lim} = [65°, -45°]$; (b) feature error norm; (c) ECM camera velocities.

the test, as in the previous test, considering the joint limits for the first two joints. It displays the error norm as well as the ECM camera velocities. Furthermore, the figure shows the trends of the CBF $\boldsymbol{h}$-functions as they relate to the visibility constraint.

B. Discussion

The "Unconstrained IBVS control test" imposes no constraints on feature vector movement inside the image plane. The IBVS task is completed, as shown in Fig. 4(a), but since no constraints are enforced, the feature trajectories can leave the FOV or ultimately reach dangerously near to its margins. In this test, all of the feature points begin at an initial pixel (o) and go to the desired point ($*$). In this scenario, the $\boldsymbol{s}_1$ feature's trajectory (red line) is out of the FOV throughout the test, and the $\boldsymbol{s}_4$ trajectory (green line) is also dangerously near to the FOV circle. This means that the point marked on the surgical tool may potentially exit the image plane and become invisible to the camera. As a consequence, the IBVS task is expected to fail. Since the features at least stay within the image plane (black box), the nominal controller remains functional in this case. However, the IBVS task's performance is unavoidably reduced. During the test, as shown in Fig. 5(b), the error norm tries to decrease to zero, completing the task, but it still exhibits some spikes (0.8 in the $5 - 15$ sec time interval). The camera velocities derived from the IBVS controller are shown in Fig. 5(c). Finally, only the first two joints contribute to the ECM motions during the test, and the joint position values stay within their joint limits, as illustrated in Fig. 5(a). Nevertheless, joint boundaries can potentially be approached because of no restrictions, particularly if the IBVS is about to fail. The IBVS nominal controller is enforced by imposing two extra constraints on features visibility and joint limits avoidance during the "CBFs-constrained IBVS control test". The IBVS task is successfully completed with a limited computational time of 2.913 ms. Unlike the previous test, all feature trajectories stay within the FOV circle region for the entire duration of the experiment. As shown in Fig. 4(b), all feature points in this test start at an initial pixel (o) and travel to the desired location ($*$), but they all keep a safe distance from the margins of the FOV circle. As shown in Fig. 6(b), the feature error norm in this experiment progressively declines to zero. Additionally, it displays spikes (0.1 at $15 - 20$ secs), especially when certain features ($\boldsymbol{s}_1$, $\boldsymbol{s}_2$, and $\boldsymbol{s}_4$) approach the FOV circle. The CBF $\boldsymbol{h}$-functions trends on visibility constraints in Fig. 6(d), which show that at that time period, $\boldsymbol{h}_{s1}$, $\boldsymbol{h}_{s2}$, and $\boldsymbol{h}_{s4}$ values slightly decline, also attest to this. The other parameters, however, are always kept at high values. As can be seen in Fig. 6(c), the camera velocities do not significantly exceed those of the previous test in an effort to keep the feature in the FOV. Finally, as shown in Fig. 6(a), the joint position values remain within their joint limits. Joint limits, however, are inaccessible due to the enforced constraint. The aforementioned findings attest

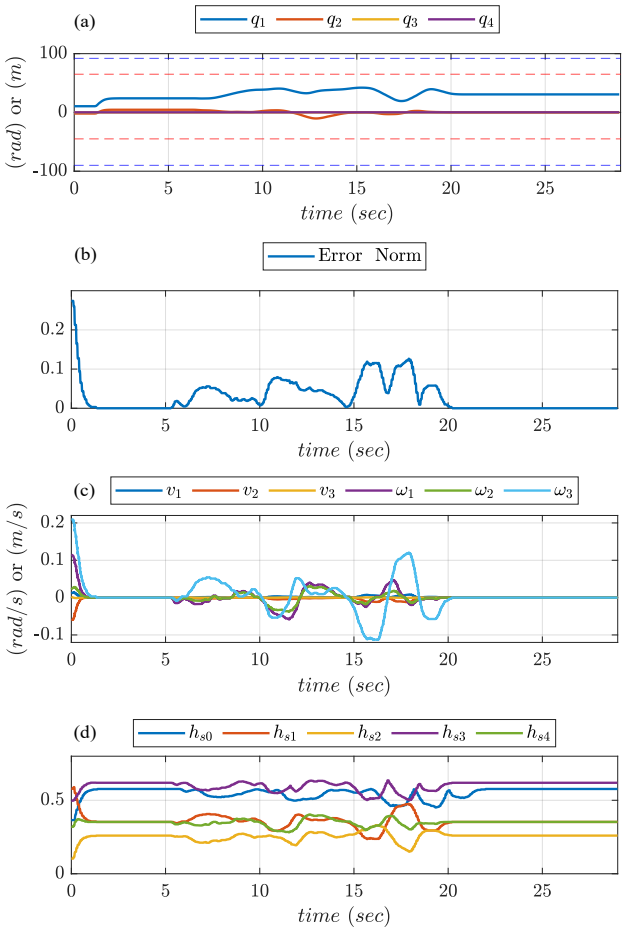


Fig. 6: CBFs-constrained IBVS control test: (a) joint position values and joint limits for the first two joint: $\boldsymbol{q}_{1lim} = [90°, -90°]$, $\boldsymbol{q}_{2lim} = [65°, -45°]$; (b) feature error norm; (c) ECM camera velocities; (d) CBF $\boldsymbol{h}$-functions as they relate to the visibility constraint.

to the constrained controller's superior performance over the basic IBVS law. The suggested approach ensures that every feature of the identified surgical instrument remains inside the FOV and never leaves the image plane, preventing a failure of the autonomous camera control that might compromise the surgical operation.

V. Conclusion

In this paper, a new autonomous endoscope control technique for the dVRK's ECM is presented. This technique enables the camera to follow the PSM's surgical tool. The methodology is detailed in the paper, starting with the deep learning algorithm used to detect surgical tool, the basic IBVS nominal controller, and the CBFs method used to provide visibility and joint limits avoidance constraints. The method is tested on the dVRK robot in two sessions, each of which included and excluded the CBFs constraints. The reported results attest to the CBFs approach's strong performance, providing high-quality autonomous camera control. Future works will include more extensive and realistic surgical experiments, as well as a user study including both novice and experienced surgeons.

References

[1] P. Rodrigues Armijo, C. K. Chun-Kai Huang, T. Carlson, D. Oleynikov, and K. Siu, "Ergonomics analysis for subjective and objective fatigue between laparoscopic and robotic surgical skills practice among surgeons," *Surgical Innovation*, vol. 27, no. 1, pp. 81–87, 2020.

[2] A. Pandya, L. Reisner, B. King, N. Lucas, A. Composto, M. Klein, and R. D. Ellis, "A review of camera viewpoint automation in robotic and laparoscopic surgery," *Robotics*, no. 3, pp. 310–329, 2014.

[3] T. Da Col, G. Caccianiga, and M. Catellani *et al.*, "Automating Endoscope Motion in Robotic Surgery: A Usability Study on da Vinci-Assisted Ex Vivo Neobladder Reconstruction," *Frontiers in Robotics and AI*, vol. 8, no. 707704, 2021.

[4] C. D'Ettorre, A. Mariani, A. Stilli, F. Rodriguea y Baena, P. Valdastri, A. Deguet, P. Kazanzides, R. h. Taylor, G. Fischer, S. P. DiMaio, A. Menciassi, and D. Stoyanov, "Accelerating surgical robotics research: A review of 10 years with the da vinci research kit," *IEEE Robotics & Automation Magazine*, vol. 28, no. 4, pp. 56–78, 2021.

[5] S. Eslamian, L. A. Reisner, B. W. King, and A. K. Pandya, "Towards the implementation of an autonomous camera algorithm on the da vinci platform," *Medicine Meets Virtual Reality Conference MMVR*, pp. 118–123, 2016.

[6] C. Molnar, T. D. Nagy, R. N. Elek, and T. Haidegger, "Visual servoing-based camera control for the da vinci surgical system," *IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 107–112, 2020.

[7] T. Da Col, A. Mariani, A. Deguet, A. Menciassi, P. Kazanzides, and E. De Momi, "SCAN: System for camera autonomous navigation in robotic-assisted surgery," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2996–3002, 2020.

[8] J. J. Ji, S. Krishnan, V. Patel, D. Fer, and K. Goldberg, "Learning 2D surgical camera motion from demonstrations," *IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 35–42, 2018.

[9] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," *18th European Control Conference (ECC)*, pp. 3420–3431, 2019.

[10] G. Notomista and M. Egerstedt, "Constraint-Driven Coordinated Control of Multi-Robot Systems," *American Control Conference (ACC)*, pp. 1990–1996, 2019.

[11] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[12] G. Notomista, S. Mayya, M. Selvaggio, M. Santos, and C. Secchi, "A Set-Theoretic Approach to Multi-Task Execution and Prioritization," *IEEE International Conference on Robotics and Automation*, pp. 9873–9879, 2020.

[13] F. Ferraguti, C. Talignani Landi, A. Singletary, H. C. Lin, A. Ames, C. Secchi, and M. Bonfe, "Safety and efficiency in robotics: The control barrier functions approach," *IEEE Robotics & Automation Magazine*, pp. 2–14, 2022.

[14] R. Moccia, C. Iacono, B. Siciliano, and F. Ficuciello, "Vision-Based Dynamic Virtual Fixtures for Tools Collision Avoidance in Robotic Surgery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1650–1655, June 2020.

[15] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A V-REP Simulator for the da Vinci Research Kit robotic Platform," *7th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1056–1061, 2018.

[16] M. Ferro, D. Brunori, F. Magistri, L. Saiella, M. Selvaggio, and G. A. Fontanelli, "A portable da vinci simulator in virtual reality," *3rd IEEE International Conference on Robotic Computing*, pp. 447–448, 2019.

[17] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "Portable dVRK: an augmented V-REP simulator of da Vinci Research Kit," *Acta Polytechnica Hungarica*, vol. 16, no. 8, pp. 79–98, 2019.

[18] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. New York, NY, USA: Springer-Verlag, 2009.

[19] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci® Surgical System," *IEEE International Conference on Robotics and Automation*, pp. 6434–6439, 2014.

[20] dVRK Wiki. [Online]. Available: http://research.intusurg.com/dvrkwiki