Autonomous Field-of-View Adjustment Using Adaptive Kinematic Constrained Control with Robot-Held Microscopic Camera Feedback

DOI:
10.1109/ICRA57147.2024.10610663

Document Version
Accepted author manuscript

Published in:
2024 IEEE International Conference on Robotics and Automation, ICRA 2024

Autonomous Field-of-View Adjustment Using Adaptive Kinematic Constrained Control with Robot-Held Microscopic Camera Feedback

Hung-Ching Lin, Murilo Marques Marinho, Kanako Harada

***Abstract*— Robotic systems for manipulation in millimeter scale often use a camera with high magnification for visual feedback of the target region. However, the limited field-of-view (FoV) of the microscopic camera necessitates camera motion to capture a broader workspace environment. In this work, we propose an autonomous robotic control method to constrain a robot-held camera within a designated FoV. Furthermore, we model the camera extrinsics as part of the kinematic model and use camera measurements coupled with a U-Net based tool tracking to adapt the complete robotic model during task execution. As a proof-of-concept demonstration, the proposed framework was evaluated in a bi-manual setup, where the microscopic camera was controlled to view a tool moving in a pre-defined trajectory. The proposed method allowed the camera to stay 94.1% of the time within the real FoV, compared to 54.4% without the proposed adaptive control.**

I. Introduction

One major hurdle in robotic systems for manipulation tasks under microscopic view is providing situational awareness to operators or autonomous agents to close the task control loop effectively. A single fixed (or manually moved) high-magnification camera vision system is a common solution to this challenge. At this scale, due to the limited resolution of image sensors for digital zoom, the limited field-of-view of the camera, and the restricted working distance of lenses with high magnification, constant motion of the camera is often needed to capture essential points-of-view of the environment.

One pertinent example is *intravital imaging* [1]. This procedure involves installing a transparent observatory cranial window in mouse skulls with an 8 mm glass, enabling scientists to monitor human organoid growth in the mouse brain *in-vivo*. We have developed a multi-arm robotic platform [2] for performing scientific exploration experiments, showing that we can perform cranial window drilling in mock egg-shell trials and *ex-vivo* mice through teleoperation [2] and autonomously [3]. Thanks to the progress of these earlier works, we can explore the next step of the procedure, such as organoid implantation. This would demand a higher magnification camera with a shorter focus distance range (approximately $\pm$5 mm) and a smaller field of view (approximately 1 cm X 1 cm). Therefore, the task would effectively require camera motion to maintain the strict ~40cm focus distance while operating in the larger workspace.

This work was supported by JST Moonshot R&D JPMJMS2033.

Hung-Ching Lin and Kanako Harada are with the Department of Mechanical Engineering, the University of Tokyo, Tokyo, Japan. Emails:qlin1806, kanako@g.ecc.u-tokyo.ac.jp.

Murilo M. Marinho is with the Department of Electrical and Electronic Engineering, the University of Manchester, Manchester, UK. Email:murilo.marinho@manchester.ac.uk

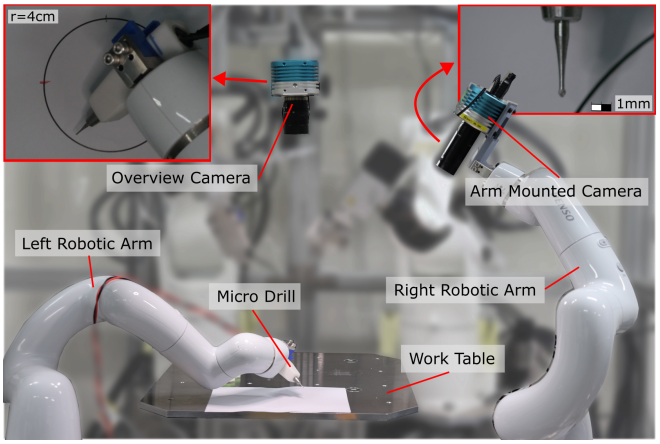

Fig. 1. The system configuration presented in this paper comprises two of the four arms from our AI-Robot Platform for Scientific Exploration [2]. On the left is an arm holding a drill. On the right is an arm holding a camera. Above the workspace is a overview camera that provides stationary view of the workspace.

In this context, we hypothesize that a viable solution is to mount a camera on the robotic arm, making it controllable. This approach attempts to ensure that the field of view remains focused on the tool, whether controlled via teleoperation or an autonomous agent. The concept is inspired by existing literature on *minimally invasive surgery* (MIS) in topics such as endoscope automation. In MIS systems, of which the da Vinci Surgical System [4] is an example, the robotic endoscope is controllable through teleoperation.

A major difficulty in this context is that a constrained workspace shared by many arms must have model-based self-collision avoidance [2]. The model comprises parameters related to the geometry of the robots and, additionally, in this work, the *extrinsics* of the mounted camera. In our target environment, owing to size constraints, an *offline* calibration procedure using additional sensors with large footprints, such as [5] and [6], is not viable. Instead, we are interested in exploiting camera information to calibrate the model *online*, through adaptive control [7].

A. Related works

A significant portion of the related work on camera automation is in the domain of MIS systems. For instance, Ali *et al.* [17] and Dardona *et al.* [18] focused on alleviating the surgeon's workload through novel methods for camera control input using eye gaze tracking and head-mounted display, respectively. In works toward the automation of the camera movement, King *et al.* [19] employed color

TABLE I

CAPABILITIES OF THE PROPOSED WORK IN CONTRAST WITH EXISTING LITERATURE.

	This work	[8]	[9]	[10]	[11]	[12]	[13]	[14]*	[15]	[16]
Visual servoing	✓	✓	✓	✓	✓	✓	✓	×	✓	✓
Marker-less	✓	✓	×	×	×	×	✓	△	×	×
Adaptive (kinematics)	✓	×	✓	✓	✓	✓	×	✓	×	✓
Adaptive (extrinsics)	✓	×	×	×	✓	×	✓	×	×	×
Workspace constraints	✓	✓	×	×	×	✓	×	×	×	×
FoV constraints	✓	✓	×	×	×	×	✓	×	✓	✓
Scale	mm	mm	cm	cm	cm	m	cm	mm	m	m

* In [14], marker-based tracking was used for obtaining ground-truth.

markers on effectors to track the tools in a surgical procedure, subsequently using this data to generate the control input for endoscopes based on a predefined set of rules. Extending [19], Eslamian *et al.* [20] implemented the method on the da Vinci Surgical System for comparison against the traditional clutch-based camera control. In a different approach, Ji *et al.* [21] proposed a method of identifying viewpoints of interest in a camera view to automatically re-center the camera learning from surgeons' demonstrations. In more related examples that also perform adaptation of robot parameters, Wang *et al.* [14] proposed a method that makes use of the property of the RCM-based robot for online calibration with an endoscopic camera end joint encoder. Outside of the MIS domain, early work by Hsu *et al.* [9] and Zergeroglu *et al.* [10] have both proposed methods for adaptive control of fixed cameras on planer robots. Later work by Cheah *et al.* in [11] also proposed a method for visual servoing using the shadow cast by the end-effector for parameter adaptation, and in [12], was extended to include force information and allow for tracking on a workspace constraint surface.

Regarding camera automation works that also took into account the FoV constraints, Chesi and Hung [22] proposed a method for path planning accounting for workspace constraints in addition to FoV constraints, and Bechlioulis *et al.* [15] demonstrated a visual servoing that is robust to depth measurement error. In a slightly different approach, the work by Zhong *et al.* [13]proposed a method of servoing without an explicit model of the robot and using a neural network to assist in state estimation.

We have shown in our prior work by Koyama *et al.* [8] that it is possible to effectively use the robot's model to automate the light-guide in an automated bimanual setup to provide consistent lighting conditions. Image tracking of the tooltip was used to move the robot to a set point but with no adaptation of the robot parameters. However, in such scenario, even minor discrepancies in robot parameters or change in effector weight and configuration often result in relatively large misalignment in the pose of the robot effector. Gonzales *et al.* [23] proposed a method for compliance compensation correcting for errors from weight or load on the robot and is strong evidence that robot misalignment can be difficult to model and impossible to calibrate offline for all trajectories. Marinho and Adorno [7] have recently introduced an adaptive constrained kinematic control that can utilize partial sensor information. In this work, our interest is to control the instrument with FoV constraints while the camera tracking of the tooltip is used to adjust the robot model along with the camera extrinsics. Capabilities of the works described herein are shown in Table I.

B. Statement of contributions

In this work, we propose a FoV camera automation framework with parameter adaptation by (1) modeling the problem with a proper partial measurement space definition, (2) obtaining the camera extrinsics Jacobian and projector to use it in an adaptive control law; and (3) assessing our system on tracking drill position in a proof-of-concept scenario.

II. PROBLEM STATEMENT

Consider the system in Fig. 1 as part of our robot platform for scientific exploration [2], from which we use two of the 8 DoF robotic branches, each one composed of a motor to rotate about the center of the workspace, a linear actuator to move along the radius of the rail, and a robotic arm (CVR038, Densowave, Japan). Let the first robot branch, $R1$, have configuration space $\boldsymbol{q}_1 \in \mathbb{R}^8$ and parameter space $\hat{\boldsymbol{a}}_1 \in \mathbb{R}^{44}$ holding a micro drill (MD1200, Braintree Scientific, USA). Let the second robot branch, $R2$, be the robotic arm (CVR038, Densowave, Japan) with joint values $\boldsymbol{q}_2 \in \mathbb{R}^8$ and parameter space $\hat{\boldsymbol{a}}_2 \in \mathbb{R}^{44}$ holding a robot-mounted camera (STC-HD853HDMI, Omron-Sentech, Japan) with microscopic lenses (VS-LDA75, VS Technology, Japan). The robotic system as a whole can be seen as having configuration space $\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_1^T & \boldsymbol{q}_2^T\end{bmatrix}^T$ and parameter space $\hat{\boldsymbol{a}} = \begin{bmatrix}\hat{\boldsymbol{a}}_1^T & \hat{\boldsymbol{a}}_2^T\end{bmatrix}^T$.

The image of the micro drill's workspace is obtained through the $R2$'s robot-mounted camera to provide the operator with a closed-up view of the task. To it, we add a U-Net [24] based keypoint detection tracking for the tip of the drill. The tracker outputs the center-adjusted pixel coordinates of the drill tip, i.e. $\boldsymbol{\rho}^{\text{oc}}(u, v) \in \mathbb{I}^2$.

A. Goal

In this work, we address a centralized kinematic controller that allows controlling $R1$ to follow a given trajectory $x_1(t)$ in the workspace while autonomously keeping the camera's FoV constraints [8] of the camera held by $R2$, and adapting

the robot's parameters, including the camera extrinsics, using the tooltip information (u, v).

III. Mathematical background

For the control and adaptation models we use (dual) quaternion algebra based on the notation of [25], [26], [7]. In a brief explanation for the notation used, a pose is represented by a unit dual quaternion, e.g. $\underline{\boldsymbol{\mathcal{S}}} \ni \underline{\boldsymbol{x}} = \boldsymbol{r} + 0.5\varepsilon \boldsymbol{t}\boldsymbol{r}$, where $\varepsilon \neq 0$ but $\varepsilon^2 = 0$. The translation is defined a the pure quaternion, e.g. $\mathbb{H}_p \ni \boldsymbol{t} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k}$, where $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$, and can be mapped into a vector with $\operatorname{vec}_3 \boldsymbol{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \in \mathbb{R}^3$ to represent a point in space. The rotation is defined by a unit quaternion, e.g. $\mathbb{S}^3 \ni \boldsymbol{r} = \cos(\phi/2) + \boldsymbol{v}\sin(\phi/2)$, that represents a rotation of ϕ radians about the unit vector $\boldsymbol{v} \in \mathbb{H}_p \cap \mathbb{S}^3$ and can be mapped into a vector with $\operatorname{vec}_4 \boldsymbol{r} = \begin{bmatrix} \cos(\phi/2) & v_x \sin(\phi/2) & v_y \sin(\phi/2) & v_z \sin(\phi/2) \end{bmatrix}^T \in \mathbb{R}^4$. A Plucker line is a pure dual quaternion with unit norm, e.g. $\mathcal{H}_p \cap \underline{\boldsymbol{\mathcal{S}}} \ni \underline{\boldsymbol{l}} = \boldsymbol{l} + \varepsilon(\boldsymbol{p}_l \times \boldsymbol{l})$, where $\boldsymbol{l} \in \mathbb{H}_p \cap \mathbb{S}^3$ is the line direction and $\boldsymbol{p}_l \in \mathbb{H}_p$ is a point in the line. Any $\boldsymbol{v} \in \mathbb{H}_p \cap \mathbb{S}^3$ can be rotated by $\boldsymbol{r}$ with the adjoint, namely $\operatorname{Ad}(\boldsymbol{r})\boldsymbol{l} = \boldsymbol{r}\boldsymbol{l}\boldsymbol{r}^*$, where $\boldsymbol{r}^* = \cos(\phi/2) - \boldsymbol{v}\sin(\phi/2)$, and $\operatorname{vec}_4 \boldsymbol{r}\boldsymbol{l}\boldsymbol{r}^* = \bar{\boldsymbol{H}}_4(\boldsymbol{r}) \overset{+}{\boldsymbol{H}}_4(\boldsymbol{r}^*) \operatorname{vec}_4 \boldsymbol{l}$.

If $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ are the configuration velocities of a robotic system and $\hat{\boldsymbol{a}} \in \mathbb{R}^p$ are the *estimated* kinematic parameters [7], then we can define the *estimated* rotation and translation Jacobians with $\operatorname{vec}_4 \dot{\hat{\boldsymbol{r}}} = \boldsymbol{J}_{\hat{r},q}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\hat{r},\hat{a}}\dot{\hat{\boldsymbol{a}}}$ and $\operatorname{vec}_4 \dot{\hat{\boldsymbol{t}}} = \boldsymbol{J}_{\hat{t},q}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\hat{t},\hat{a}}\dot{\hat{\boldsymbol{a}}}$, respectively.

IV. Proposed camera automation overview

The overall framework is illustrated in Fig. 2. The task-space controller receives the desired pose signal from either an operator or predefined trajectory and works to minimize task-space error of $R1$ using current robot parameters, while both $R1$ and $R2$ maintain the task constraints. The adaptive controller receives measurements of an image-processing algorithm to update the estimated robot parameters to reduce measurement error. The position of the tooltip of $R1$ is measured using the camera image from $R2$'s robot-mounted camera processed through a U-Net-based pipeline.

A. U-Net-based point tracking

A U-Net-based [24] keypoint detection algorithm was implemented with *segmentation_models.pytorch*[1] on images down-sampled to 576×576 pixels to output a confidence map for the tip of the drill held by $R1$. We use U-Net as it is a well established technique with an acceptable level of robustness. The network used *resnet18* as the encoder initialized with *ImageNet*[2] weight. A total of 2500 original and 17500 augmented images were provided with common augmentation techniques (e.g., blurs, rescaling, rotate, color shift, random gamma, brightness and contrast.) , with 9:1 train-validation split without a testing dataset for 30 epochs in training. Inference was then performed on a similarly down-sampled camera stream in real-time to obtain the drill tip pixel coordinate measurement $\boldsymbol{\rho}^{\text{oc}}(u, v) \in \mathbb{I}^2$.

[1] https://github.com/qubvel/segmentation_models.pytorch

[2] https://www.image-net.org/

B. Task controller

The task controller is based on [27], [26] with FoV constraints based on [8]. Let $\underline{\boldsymbol{\mathcal{S}}} \ni \hat{\underline{\boldsymbol{x}}}_i = \hat{\boldsymbol{r}}_i + 0.5\varepsilon\hat{\boldsymbol{t}}_i\hat{\boldsymbol{r}}_i$ and $\underline{\boldsymbol{\mathcal{S}}} \ni \hat{\underline{\boldsymbol{x}}}_{i,d} = \hat{\boldsymbol{r}}_{i,d} + 0.5\varepsilon\hat{\boldsymbol{t}}_{i,d}\hat{\boldsymbol{r}}_{i,d}$ be, respectively, the estimated effector poses and desired effector poses of the two robots Ri with $i \in \{1, 2\}$. We solve, at each time step to obtain a joint velocity control signal $\boldsymbol{u}_q = \begin{bmatrix} \boldsymbol{u}_{q_1}^T & \boldsymbol{u}_{q_2}^T \end{bmatrix}^T$, the following optimization problem

$$\begin{aligned} \boldsymbol{u}_q \in \underset{\dot{\boldsymbol{q}}}{\operatorname{argmin}}\ & \beta\left(\alpha f_{t,1} + (1-\alpha) f_{r,1} + \|\lambda\dot{\boldsymbol{q}}_1\|_2^2\right) \\ & + (1-\beta)\left(f_{t,2} + \|\lambda\dot{\boldsymbol{q}}_2\|_2^2\right) \\ \text{subject to}\ & \begin{matrix} \boldsymbol{W}_q\ \dot{\boldsymbol{q}} \preceq\ \boldsymbol{w}_q \\ \boldsymbol{B}_q\dot{\boldsymbol{q}} \preceq \boldsymbol{b}_q \end{matrix} \end{aligned} \tag{1}$$

in which $\lambda > 0 \in \mathbb{R}$ is the damping term, $f_{t,i} \triangleq \left\|\boldsymbol{J}_{\hat{t},q_i}\dot{\boldsymbol{q}}_i + \eta_q \operatorname{vec}_4(\tilde{\boldsymbol{t}}_i)\right\|_2^2$ is the cost function of the translation error $\tilde{\boldsymbol{t}} \triangleq \hat{\boldsymbol{t}}_i - \boldsymbol{t}_{i,d}$. In addition, $f_{r,i} \triangleq \|\boldsymbol{J}_{\hat{r},q_i}\dot{\boldsymbol{q}}_i + \eta_q \operatorname{vec}_4(\tilde{\boldsymbol{r}}_i)\|_2^2$ is the cost function of the switching rotational error

$$\tilde{\boldsymbol{r}_i} \triangleq \begin{cases} (\hat{\boldsymbol{r}}_i)^* \boldsymbol{r}_{i,d} - 1 & \text{if } \|\hat{\boldsymbol{r}}_i^* \boldsymbol{r}_{i,d} - 1\|_2 < \|\hat{\boldsymbol{r}}_i^* \boldsymbol{r}_{i,d} + 1\|_2 \\ (\hat{\boldsymbol{r}}_i)^* \boldsymbol{r}_{i,d} + 1 & \text{otherwise.} \end{cases}$$

The term $\eta_q \in (0, \infty) \subset \mathbb{R}$ is the proportional task error gain. The term $\alpha \in [0, 1] \subset \mathbb{R}$ is a weight factor for the rotation and translation cost functions. The term $\beta \in [0, 1] \subset \mathbb{R}$ is weight factor for the cost functions of each robot, in practice a "soft" prioritization of robot with larger weight. For this work we chose $\eta_q = 3$, $\alpha = 0.99$, and $\beta = 0.999$.

$R1$ is controlled by the desired task-space values $\boldsymbol{t}_{1,d}$ and $\boldsymbol{r}_{1,d}$, whereas $R2$ is autonomously moved by the optimization problem, with a small preference towards a neutral point-of-view given by $\boldsymbol{t}_{2,d}$.

Constraints: The inequality constraints have two components. The first, obtained with $\begin{pmatrix} \boldsymbol{W}_q, & \boldsymbol{w}_q \end{pmatrix}$, concerns joint-space constraints, namely position/velocity limits [7, Constraint (11)]. The other component obtained with $\begin{pmatrix} \boldsymbol{B}_q, & \boldsymbol{b}_q \end{pmatrix}$, is related to task-space constraints made of four parts. First, the collision avoidance of each robot with the environment implemented with point-to-plane and point-to-line distance constraints [26, Constraint (59), (32)], using 4 points positioned on each robot and the central work-stage modeled as a cylinder limited by one top plane. Second, the collision avoidance between the robots using 5 point-to-point distance constraints [26, Constraint (23)] with points defined around the robots effector and links. Third, the FoV constraint [8, the second row of Constraint (23)]. Lastly, a focal distance constraint implemented as a pair of complimentary point-to-point distance constraints with safe and forbidden zone direction [26, Constraint (17), (18)]

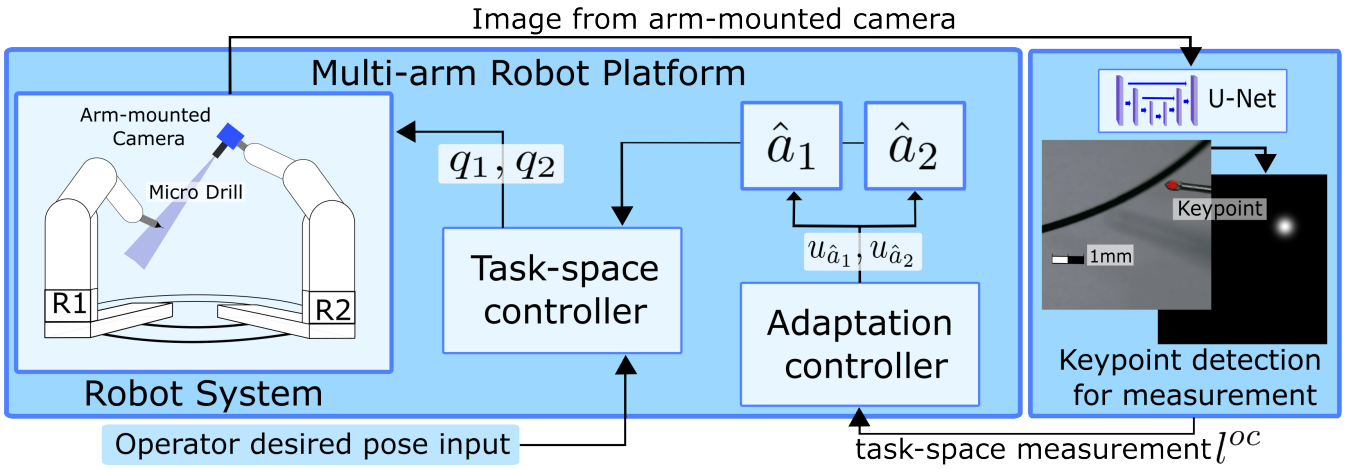

Fig. 2. Block diagram of the proposed system. The task-space controller receivesthe desired pose control signal to move R1 in a given trajectory, and both $R1$ and $R2$ to maintain the task constraints, including the FoV constraint. Meanwhile, a U-Net based tracking algorithm using images of the robot-mounted camera in $R2$ outputs the tooltip of the tool held by $R1$. Using these measurements, the adaptive controller updates the estimated parameters of the kinematic model of both robots, which include the camera extrinsics. That updated model is used by the task-space controller, and the task–adaptive loop is closed.

respectively imposed on points placed on the drill tip and camera optical center.

C. Adaptive controller

We solve, at each time step to obtain an adaptation signal $\boldsymbol{u}_{\hat{a}} = \left[\boldsymbol{u}_{\hat{a}_1}^T \; \boldsymbol{u}_{\hat{a}_2}^T\right]^T$, the following optimization problem [7]

$$\boldsymbol{u}_{\hat{a}} \in \underset{\dot{\hat{\boldsymbol{a}}}}{\operatorname{argmin}} \left\| \boldsymbol{J}_{\hat{y},\hat{a}}\dot{\hat{\boldsymbol{a}}} + \eta_a \tilde{\boldsymbol{y}} \right\|_2^2 + \left\| \boldsymbol{\Lambda}_{\hat{a}}\dot{\hat{\boldsymbol{a}}} \right\|_2^2 \qquad (2)$$

$$\text{subject to} \quad \begin{aligned} \boldsymbol{W}_{\hat{a}}\dot{\hat{\boldsymbol{a}}} &\preceq \boldsymbol{w}_{\hat{a}} \\ \boldsymbol{B}_{\hat{a}}\dot{\hat{\boldsymbol{a}}} &\preceq \boldsymbol{b}_{\hat{a}} \\ \boldsymbol{N}_{\hat{a}}\dot{\hat{\boldsymbol{a}}} &= \boldsymbol{0} \\ \tilde{\boldsymbol{x}}^T \boldsymbol{J}_{x,\hat{a}}\dot{\hat{\boldsymbol{a}}} &\leq 0 \end{aligned}$$

where $\eta_a \in (0, \infty)$ is the proportional controller gain, $\boldsymbol{\Lambda}_{\hat{a}}$is a positive definite diagonal damping matrix. The constraints defined by $(\boldsymbol{W}_{\hat{a}}, \boldsymbol{w}_{\hat{a}})$ and $(\boldsymbol{B}_{\hat{a}}, \boldsymbol{b}_{\hat{a}})$ are the parameter-space analogous of their configuration-space counterparts defined in (1). Lastly, the Lyapunov constraint is written with

$$\tilde{\boldsymbol{x}} \triangleq \left[\; \sqrt{\beta\alpha}\,\operatorname{vec}4\left(\tilde{\boldsymbol{t}}_1\right) \quad \sqrt{\beta\left(1-\alpha\right)}\,\operatorname{vec}4\left(\tilde{\boldsymbol{r}}_1\right) \quad \sqrt{1-\beta}\,\operatorname{vec}4\left(\tilde{\boldsymbol{t}}_2\right) \; \right]^T$$

and

$$\boldsymbol{J}_{x,\hat{a}} \triangleq \begin{bmatrix} \sqrt{\beta\alpha}\boldsymbol{J}_{t,1} & \boldsymbol{0} \\ \sqrt{\beta\left(1-\alpha\right)}\boldsymbol{J}_{r,1} & \boldsymbol{0} \\ \boldsymbol{0} & \sqrt{1-\beta}\boldsymbol{J}_{t,2} \end{bmatrix}.$$

In this paper, we propose a suitable measurement model, $\boldsymbol{y}$ (see (3)), an estimated measurement model, $\hat{\boldsymbol{y}}$ (see (5)), a measurement error, $\tilde{\boldsymbol{y}}$ (see (6)), the measurement-space Jacobian, $\boldsymbol{J}_{\hat{y},\hat{a}}$ (see (10)), and complementary task-space Jacobian, $\boldsymbol{N}_{\hat{a}}$ (see (11)), for single-shot camera measurements where the relation of one point in the image and its task-space counterpart is known.

V. Proposed Camera-Adaptive Methodology

This section contains the main technical contribution of this work. We start by modeling the camera extrinsics in Section V-A to obtain the measurement space, i.e. model what real-world information can be obtained from the camera. We then relate that model to a robot-mounted camera in Section V-B to obtain the estimated measurement-space. We then define the measurement error and obtain the related parametric-space Jacobian in Section V-C. Lastly, we obtain the parametric task-Jacobian projector in Section V-D.

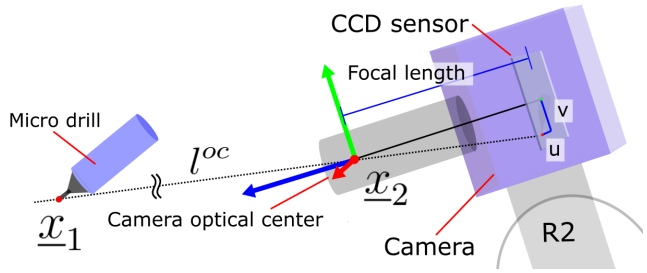

Fig. 3. Relevant elements for the camera extrinsics modeling of Section V. Notice that $\boldsymbol{l}^{\text{oc}} \in \mathbb{H}_p \cap \mathbb{S}^3$ is the task-space measurement, the direction of the line connecting the pixel $\boldsymbol{\rho}^{\text{oc}}(u,v) \in \mathbb{I}^2$ representing the tooltip of $R1$ and the tooltip itself, passing through the optical center of the camera. Using this information, we can adapt the estimated robot model and extrinsics to compensate for initial modeling inaccuracies.

A. Camera extrinsics model: measurement space

In this work, we consider the dual-quaternion model for a pin-hole camera, given as follows. Let a point in world-coordinates $\boldsymbol{t}^{\text{w}} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k}$ represent a un-occluded point of a world object, whose real coordinates are not measurable directly. The pin-hole camera model is such that $\boldsymbol{t}^{\text{w}}$, in world coordinates, and its pairing pixel[3] $\boldsymbol{\rho}^{\text{oc}}(u,v) = u\hat{\imath} + v\hat{\jmath}$, in image coordinates, are connected by a Plucker line that crosses the optical center, $\boldsymbol{p}_{\text{oc}}^{\text{w}}$, given by

$$\underline{\boldsymbol{l}}^{w}\left(u,v,\boldsymbol{p}_{\text{oc}}^{w}\right) \triangleq \boldsymbol{l}^{w}\left(u,v\right) + \varepsilon\left[\boldsymbol{p}_{\text{oc}}^{w} \times \boldsymbol{l}^{w}\left(u,v\right)\right],$$

where $\boldsymbol{p}_{\text{oc}}^{w}$ is the position of the camera frame, $\underline{\boldsymbol{x}}_{\text{oc}}^{w}$, obtained by decomposing the relation $\underline{\boldsymbol{x}}_{\text{oc}}^{w} = \boldsymbol{r}_{\text{oc}}^{w} + 0.5\varepsilon\boldsymbol{p}_{\text{oc}}^{w}\boldsymbol{r}_{\text{oc}}^{w}$,

[3]Note that the pairing pixel is obtained from the image-processing algorithm described in Section (IV-A).

commonly known as the camera extrinsics. The rotation of the optical center, $\boldsymbol{r}_{\text{oc}}^{w}$, can be used to map the line direction between the optical center frame and the world frame $\boldsymbol{l}^{w}(u,v) = \text{Ad}(\boldsymbol{r}_{\text{oc}}^{w})\,\boldsymbol{l}^{\text{oc}}(u,v)$.

Under the assumption of the pin-hole camera model, as illustrated in Fig. 4, and supposing that the camera constant[4] *intrinsics* are known[5], we can find a pixel in task-space, as seen by optical center, as

$$\boldsymbol{p}_{\rho}^{\text{oc}} \triangleq \boldsymbol{p}_{\rho}^{\text{oc}}(u,v) = us_x\hat{\imath} + vs_y\hat{\jmath} - f\hat{k},$$

where $f, s_x, s_y \in \mathbb{R}^{+} - \{0\}$ are the focal length, the $x-$axis size of the pixel, and the $y-$axis size the each pixel, respectively. This means that, from a given pixel, we can measure a line direction, with respect to the optical center, given by

$$\boldsymbol{y} \triangleq \boldsymbol{l}^{\text{oc}}(u,v) = \frac{\left(us_x\hat{\imath} + vs_y\hat{\jmath} - f\hat{k}\right)}{\left\|\boldsymbol{p}_{\rho}^{\text{oc}}\right\|}, \tag{3}$$

that is not singular even when $u = v = 0$ because $f > 0$, hence $\left\|\boldsymbol{p}_{\rho}^{\text{oc}}\right\| > 0 \;\forall\; u,\, v$.

B. Robot-mounted camera model: estimated measurement-space

Consider that the camera robot, $R2$, is modeled such that the camera reference corresponds to its end-effector, i.e.

$$\underline{\hat{\boldsymbol{x}}}_{\text{oc}}^{w} \triangleq \underline{\hat{\boldsymbol{x}}}_{2}(\boldsymbol{q}_2) = \hat{\boldsymbol{r}}_2 + \frac{1}{2}\varepsilon\hat{\boldsymbol{t}}_2\hat{\boldsymbol{r}}_2.$$

With the point of interest on $R1$ defined at the tooltip position $\hat{\boldsymbol{t}}_1$, obtained from $\underline{\hat{\boldsymbol{x}}}_{1}(\boldsymbol{q}_1) \triangleq \underline{\hat{\boldsymbol{x}}}_{1} = \hat{\boldsymbol{r}}_1 + 0.5\varepsilon\hat{\boldsymbol{t}}_1\hat{\boldsymbol{r}}_1$, we can define the estimated line as

$$\hat{\boldsymbol{l}}^{\text{oc}} \triangleq \hat{\boldsymbol{l}}_{2,1}(\boldsymbol{q}) = \frac{\hat{\boldsymbol{t}}_1 - \hat{\boldsymbol{t}}_2}{\left\|\hat{\boldsymbol{t}}_1 - \hat{\boldsymbol{t}}_2\right\|}, \tag{4}$$

which will never be singular because $\hat{\boldsymbol{t}}_2 \neq \hat{\boldsymbol{t}}_1$, that is, the tip of R1 can never physically be at the optical center given the collision-avoidance constraints. Lastly, we adjust the reference frame to be at the optical center, to match our measurement (3)

$$\hat{\boldsymbol{y}} \triangleq \hat{\boldsymbol{l}}^{\text{oc}} = \text{Ad}\left((\hat{\boldsymbol{r}}_2)^{*}\right)\hat{\boldsymbol{l}}_{2,1}. \tag{5}$$

C. Measurement error and adaptation Jacobians

We can define measurement error as the difference between (5) and (3), namely

$$\tilde{\boldsymbol{y}} \triangleq \hat{\boldsymbol{l}}^{\text{oc}} - \boldsymbol{l}^{\text{oc}}. \tag{6}$$

We use the error information to obtain the parameter Jacobian. We start from the time-derivative of (5)

$$\begin{aligned}\dot{\hat{\boldsymbol{l}}}^{\text{oc}} &= (\hat{\boldsymbol{r}}_2)^{*}\,\hat{\boldsymbol{l}}_{2,1}\hat{\boldsymbol{r}}_2 \\ &= \left(\dot{\hat{\boldsymbol{r}}}_2\right)^{*}\hat{\boldsymbol{l}}_{2,1}\hat{\boldsymbol{r}}_2 + (\hat{\boldsymbol{r}}_2)^{*}\,\dot{\hat{\boldsymbol{l}}}_{2,1}\hat{\boldsymbol{r}}_2 + (\hat{\boldsymbol{r}}_2)^{*}\,\hat{\boldsymbol{l}}_{2,1}\dot{\hat{\boldsymbol{r}}}_2,\end{aligned} \tag{7}$$

meaning that we need to find $\dot{\hat{\boldsymbol{r}}}_2$ and $\dot{\hat{\boldsymbol{l}}}_2$. The derivative of the rotation $\dot{\hat{\boldsymbol{r}}}_2$ is obtained with the rotation Jacobian, namely

$$\dot{\hat{\boldsymbol{r}}}_2 = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{J}_{\hat{r}_2,\hat{a}}\end{bmatrix}\dot{\hat{\boldsymbol{a}}}. \tag{8}$$

As for the line direction derivative, $\dot{\hat{\boldsymbol{l}}}_1$, we start from

$$\begin{aligned}\boldsymbol{h} &\triangleq \hat{\boldsymbol{t}}_1 - \hat{\boldsymbol{t}}_2 \implies \\ \dot{\boldsymbol{h}} &= \dot{\hat{\boldsymbol{t}}}_1 - \dot{\hat{\boldsymbol{t}}}_2 \implies \\ \text{vec}_4\,\dot{\boldsymbol{h}} &= \boldsymbol{J}_{\hat{t}_1,\hat{a}}\dot{\hat{\boldsymbol{a}}}_1 - \boldsymbol{J}_{\hat{t}_2,\hat{a}}\dot{\hat{\boldsymbol{a}}}_2 \\ &= \begin{bmatrix}\boldsymbol{J}_{\hat{t}_1,\hat{a}} & -\boldsymbol{J}_{\hat{t}_2,\hat{a}}\end{bmatrix}\dot{\hat{\boldsymbol{a}}}\end{aligned}$$

and substitute it in the derivative of (4) to obtain

$$\begin{aligned}\hat{\boldsymbol{l}}_{2,1} &= \frac{\boldsymbol{h}}{\|\boldsymbol{h}\|} = \boldsymbol{h}\,\|\boldsymbol{h}\|^{-1} \implies \\ \dot{\hat{\boldsymbol{l}}}_{2,1} &= \dot{\boldsymbol{h}}\,\|\boldsymbol{h}\|^{-1} - \boldsymbol{h}\,\|\boldsymbol{h}\|^{-2}\langle\boldsymbol{h},\dot{\boldsymbol{h}}\rangle\,\|\boldsymbol{h}\|^{-1} \implies \\ \text{vec}_4\,\dot{\hat{\boldsymbol{l}}}_{2,1} &= \|\boldsymbol{h}\|^{-1}\,\text{vec}_4\,\dot{\boldsymbol{h}} \\ &\quad \overbrace{+\frac{1}{2}\|\boldsymbol{h}\|^{-3}\overset{+}{\boldsymbol{H}}_4(\boldsymbol{h})\left[\overset{+}{\boldsymbol{H}}_4(\boldsymbol{h}) + \overset{-}{\boldsymbol{H}}_4(\boldsymbol{h})\right]}^{\boldsymbol{A}_1}\text{vec}_4\,\dot{\boldsymbol{h}} \\ &= \overbrace{\|\boldsymbol{h}\|^{-1}\begin{bmatrix}\boldsymbol{J}_{\hat{t}_1,\hat{a}} & -\boldsymbol{J}_{\hat{t}_2,\hat{a}}\end{bmatrix}}^{\boldsymbol{A}_2}\dot{\hat{\boldsymbol{a}}} \\ &\quad + \underbrace{\boldsymbol{A}_1\begin{bmatrix}\boldsymbol{J}_{\hat{t}_1,\hat{a}} & -\boldsymbol{J}_{\hat{t}_2,\hat{a}}\end{bmatrix}}_{\boldsymbol{A}_3}\dot{\hat{\boldsymbol{a}}} \\ &= \underbrace{(\boldsymbol{A}_2 + \boldsymbol{A}_3)}_{\boldsymbol{J}_{\hat{l}_{2,1},\hat{a}}}\dot{\hat{\boldsymbol{a}}}\end{aligned} \tag{9}$$

Substituting (8) and (9) in (7) results in

$$\begin{aligned}\dot{\hat{\boldsymbol{l}}}^{\text{oc}} &= \overbrace{\overset{-}{\boldsymbol{H}}_4\left(\hat{\boldsymbol{l}}_{2,1}\hat{\boldsymbol{r}}_2\right)\boldsymbol{C}_4\begin{bmatrix}\boldsymbol{0} & \boldsymbol{J}_{\hat{r}_2,\hat{a}}\end{bmatrix}}^{\boldsymbol{B}_1}\dot{\hat{\boldsymbol{a}}} \\ &\quad + \overbrace{\overset{-}{\boldsymbol{H}}_4(\hat{\boldsymbol{r}}_2)\overset{+}{\boldsymbol{H}}_4(\hat{\boldsymbol{r}}_2^{*})\boldsymbol{J}_{\hat{l}_{2,1},\hat{a}}}^{\boldsymbol{B}_2}\dot{\hat{\boldsymbol{a}}} \\ &\quad + \overbrace{\overset{+}{\boldsymbol{H}}_4\left(\hat{\boldsymbol{r}}_2^{*}\hat{\boldsymbol{l}}_{2,1}\right)\begin{bmatrix}\boldsymbol{0} & \boldsymbol{J}_{\hat{r}_2,\hat{a}}\end{bmatrix}}^{\boldsymbol{B}_3}\dot{\hat{\boldsymbol{a}}} \\ &= \underbrace{(\boldsymbol{B}_1 + \boldsymbol{B}_2 + \boldsymbol{B}_3)}_{\boldsymbol{J}_{\hat{l}_{\text{oc}},\hat{a}}}\dot{\hat{\boldsymbol{a}}} \\ \implies \boldsymbol{J}_{\hat{y},\hat{a}} &= \boldsymbol{B}_1 + \boldsymbol{B}_2 + \boldsymbol{B}_3.\end{aligned} \tag{10}$$

D. Parametric task-Jacobian projector, $\boldsymbol{N}_{\hat{a}}$

As stated in the adaptation law, the purpose of the parametric task-Jacobian projector exist to prevent disturbance in the unmeasured variables. Given the task-space measurement definition, the real translation of the drill in the world frame $\boldsymbol{t}_1$is unknown, therefore we can fix

$$\dot{\hat{\boldsymbol{t}}}(\hat{\boldsymbol{a}},\boldsymbol{q}) \implies \underbrace{\begin{bmatrix}\boldsymbol{J}_{\hat{t}_1,\hat{a}} & \boldsymbol{0}\end{bmatrix}}_{\boldsymbol{N}_{\hat{a},1}}\dot{\hat{\boldsymbol{a}}} = 0.$$

Additionally, we cannot know rotations about $\hat{\boldsymbol{l}}_{2,1}$, hence we will impose a 1 DoF constraint such that the angular velocity

[4]Considering intrinsics adaptation is the topic of ongoing future work.

[5]We used MATLAB's camera calibration `https://www.mathworks.com/help/vision/camera-calibration.html`.

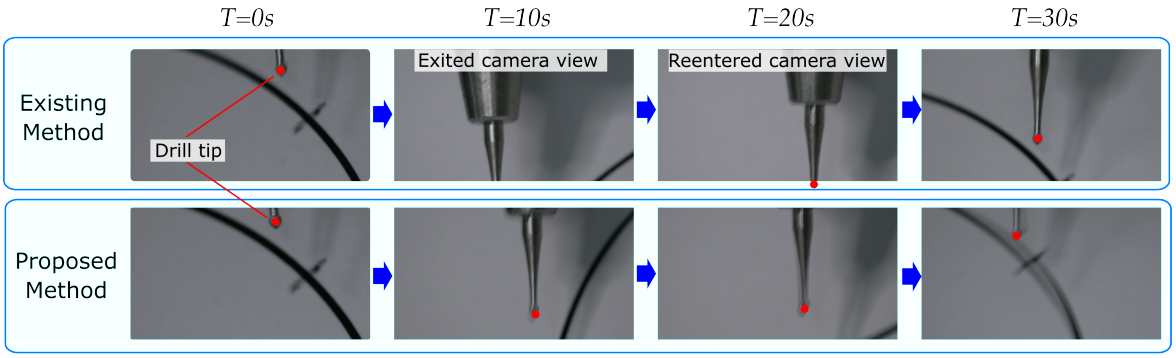

Fig. 4. Snapshots of the robot-mounted camera while $R1$ traverses a predefined trajectory. The drilltip frequently leaves the desired FoV when only kinematic constrained control is used, whereas it effectively stays within the desired FoV when using the proposed adaptive strategy.

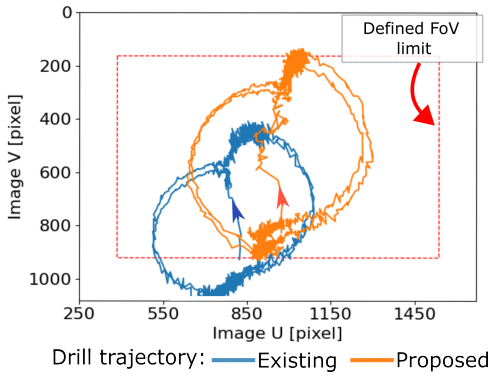

Fig. 5. The plot trace the trajectories of the drill tip under the robot-mounted camera for two rotational period.

$\boldsymbol{\omega}_l$ is perpendicular $\hat{\boldsymbol{l}}_{2,1}$, namely

$$\begin{aligned} &\langle \hat{\boldsymbol{l}}_{2,1}, \boldsymbol{\omega}_l(\hat{a}, q) \rangle = 0 \\ \implies &\left\langle \hat{\boldsymbol{l}}_{2,1}, 2\dot{\hat{\boldsymbol{r}}}_2 \hat{\boldsymbol{r}}_2^* \right\rangle = 0 \\ &\underbrace{2\,\mathrm{vec}_4\,\hat{\boldsymbol{l}}_{2,1}\,\bar{\boldsymbol{H}}\,(\hat{\boldsymbol{r}}_2^*)\,\begin{bmatrix}\boldsymbol{0} & \boldsymbol{J}_{\hat{r}_2,\hat{a}}\end{bmatrix}}_{\boldsymbol{N}_{\hat{a},2}} \dot{\hat{\boldsymbol{a}}} = 0 \end{aligned}$$

which are combined into

$$\boldsymbol{N}_{\hat{a}} = \begin{bmatrix}\boldsymbol{N}_{\hat{a},1}^T & \boldsymbol{N}_{\hat{a},2}^T\end{bmatrix}^T \implies \boldsymbol{N}_{\hat{a}}\dot{\hat{\boldsymbol{a}}} = 0. \tag{11}$$

VI. Experiments and Results

A set of experiments was conducted to evaluate the effectiveness of the proposed method with the setup shown in Fig. 1. We defined a (1150x750 pixels) subregion of the camera as the FoV as shown in Fig. 4, so that deviations of the FoV could be measured. We implemented two controllers. The first controller represents the existing model-based control strategy, whereas the second controller uses the proposed adaptive strategy to compensate for inaccuracies in the parameters of $R2$ arms, including camera extrinsics.

At the start of the experiment, we manually adjusted the initial pose of $R1$ at the start of the trajectory and $R2$ camera to point at the drill with the tip in view and in focus. We also established parameters for FoV constraint with ,$\theta_{safe} = 0.55°$ focus distance constraints with $d_{image} = 0.405$m, and adaptation gain with $\eta_a = 7$. The experiment involved automatically moving the drill tip along a pre-defined circular path with a radius r=4 cm and a period of 30s. The path followed by the tooltip of $R1$ in the image of the camera of $R2$ was used to calculate the proportion of time that the tool remained inside the robot-mounted camera's field.

TABLE II

Real FoV constraint maintenance

	Existing	Proposed
Duration Ratio	54.4%%	94.1%
Maximum Deviation ($\theta_{\text{FoV}} - \theta_{safe}$)	$\geq 0.68°$ *	0.27°

*due to the drill tip deviating off the camera view, only a lower bound value is available.

The experimental results are presented in Table II showing the percentage of time that the *real* drill tip remained within the FoV of the robot-mounted camera and the maximum deviation outside of the *real* FoV constraint. It is important to notice that the *estimated* FoV is always kept for both controllers. Nonetheless, under our proposed method, the system successfully retained the drill tip within the *real* FoV for 94.1% of the time, compared to the 54.4% achieved by the existing method. The difference is visible in Fig. 5, in which we show the trajectory of the drill tip from the perspective of the robot-mounted camera. Our proposed method consistently kept the drill tip within the projected rectangle, representing the FoV constraints. Conversely, the control group's FoV constraints have an offset pointing slightly above the actual drill tip, demonstrating a mismatch between the estimated and actual model of the robots and the need for parameter adaptation. After improving the frequency of the image-processing pipeline, currently at 32Hz, and the vibration seen in the robot-mounted camera sourcing from the robot's joint motion, likely from the weight of the camera assembly which is near the limit of the robot's payload, we expect to get better performance using the proposed strategy.

VII. Conclusions

In this paper, we introduced a framework for automated camera positioning, employing kinematic constraints to address the camera's field of view challenges. Simultaneously, we incorporated a parameter adaptation strategy by integrating keypoint detection into task-space measurements. Future work should incorporate additional task-space measurements for improved task performance. There is also an avenue to improve system functionality by autonomously controlling the neutral point-of-view.

REFERENCES

[1] H. Koike, K. Iwasawa, R. Ouchi, M. Maezawa, K. Giesbrecht, N. Saiki, A. Ferguson, M. Kimura, W. L. Thompson, J. M. Wells, A. M. Zorn, and T. Takebe, "Modelling human hepato-biliary-pancreatic organogenesis from the foregut–midgut boundary," *Nature*, vol. 574, no. 7776, pp. 112–116, Oct. 2019.

[2] M. M. Marinho, J. J. Quiroz-Omaña, and K. Harada, "Design and Validation of a Multi-Arm Robotic Platform for Scientific Exploration," Oct. 2022.

[3] E. Zhao, M. M. Marinho, and K. Harada, "Autonomous Robotic Drilling System for Mice Cranial Window Creation: An Evaluation with an Egg Model," Mar. 2023.

[4] G. Guthart and J. Salisbury, "The Intuitive/sup TM/ telesurgery system: overview and application," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, Apr. 2000, pp. 618–621 vol.1, iSSN: 1050-4729.

[5] S. Gharaaty, T. Shu, A. Joubair, W. F. Xie, and I. A. Bonev, "Online pose correction of an industrial robot using an optical coordinate measure machine system," *International Journal of Advanced Robotic Systems*, vol. 15, no. 4, p. 1729881418787915, Jul. 2018.

[6] C. Yu and J. Xi, "Simultaneous and on-line calibration of a robot-based inspecting system," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 349–360, Feb. 2018.

[7] M. M. Marinho and B. V. Adorno, "Adaptive Constrained Kinematic Control using Partial or Complete Task-Space Measurements," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3498–3513, Dec. 2022.

[8] Y. Koyama, M. M. Marinho, M. Mitsuishi, and K. Harada, "Autonomous Coordinated Control of the Light Guide for Positioning in Vitreoretinal Surgery," *CoRR*, vol. abs/2107.11985, 2022.

[9] L. Hsu and P. Aquino, "Adaptive visual tracking with uncertain manipulator dynamics and uncalibrated camera," in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, vol. 2, Dec. 1999, pp. 1248–1253 vol.2, iSSN: 0191-2216.

[10] E. Zergeroglu, D. Dawson, M. de Querioz, and A. Behal, "Vision-based nonlinear tracking controllers with uncertain robot-camera parameters," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 322–337, Sep. 2001.

[11] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive Tracking Control for Robots with Unknown Kinematic and Dynamic Properties," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 283–296, Mar. 2006.

[12] C. C. Cheah, S. P. Hou, Y. Zhao, and J.-J. E. Slotine, "Adaptive Vision and Force Tracking Control for Robots With Constraint Uncertainty," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 3, pp. 389–399, Jun. 2010.

[13] X. Zhong, X. Zhong, and X. Peng, "Robots visual servo control with features constraint employing Kalman-neural-network filtering scheme," *Neurocomputing*, vol. 151, pp. 268–277, Mar. 2015.

[14] Z. Wang, Z. Liu, Q. Ma, A. Cheng, Y.-h. Liu, S. Kim, A. Deguet, A. Reiter, P. Kazanzides, and R. H. Taylor, "Vision-Based Calibration of Dual RCM-Based Robot Arms in Human-Robot Collaborative Minimally Invasive Surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 672–679, Apr. 2018.

[15] C. P. Bechlioulis, S. Heshmati-alamdari, G. C. Karras, and K. J. Kyriakopoulos, "Robust Image-Based Visual Servoing With Prescribed Performance Under Field of View Constraints," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1063–1070, Aug. 2019.

[16] Z. Miao, H. Zhong, J. Lin, Y. Wang, Y. Chen, and R. Fierro, "Vision-Based Formation Control of Mobile Robots With FOV Constraints and Unknown Feature Depth," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 2231–2238, Sep. 2021.

[17] S. M. Ali, L. A. Reisner, B. King, A. Cao, G. Auner, M. Klein, and A. K. Pandya, "Eye gaze tracking for endoscopic camera positioning: an application of a hardware/software interface developed to automate Aesop," *Studies in Health Technology and Informatics*, vol. 132, pp. 4–7, 2008.

[18] T. Dardona, S. Eslamian, L. A. Reisner, and A. Pandya, "Remote Presence: Development and Usability Evaluation of a Head-Mounted Display for Camera Control on the da Vinci Surgical System," *Robotics*, vol. 8, no. 2, 2019.

[19] B. W. King, L. A. Reisner, A. K. Pandya, A. M. Composto, R. D. Ellis, and M. D. Klein, "Towards an Autonomous Robot for Camera Control During Laparoscopic Surgery," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 23, no. 12, pp. 1027–1030, Dec. 2013.

[20] S. Eslamian, L. A. Reisner, and A. K. Pandya, "Development and evaluation of an autonomous camera control algorithm on the da Vinci Surgical System," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 16, no. 2, p. e2036, 2020.

[21] J. J. Ji, S. Krishnan, V. Patel, D. Fer, and K. Goldberg, "Learning 2D Surgical Camera Motion From Demonstrations," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Aug. 2018, pp. 35–42, iSSN: 2161-8089.

[22] G. Chesi and Y. S. Hung, "Global Path-Planning for Constrained and Optimal Visual Servoing," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1050–1060, Oct. 2007.

[23] M. K. Gonzalez, N. A. Theissen, A. Barrios, and A. Archenti, "Online compliance error compensation system for industrial manipulators in contact applications," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102305, Aug. 2022.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015.

[25] B. V. Adorno, "Robot kinematic modeling and control based on dual quaternion algebra — Part I: Fundamentals," 2017.

[26] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, Oct. 2019.

[27] M. M. Marinho, B. V. Adorno, K. Harada, K. Deie, A. Deguet, P. Kazanzides, R. H. Taylor, and M. Mitsuishi, "A unified framework for the teleoperation of surgical robots in constrained workspaces," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2721–2727.