2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Adaptive Image-based Positioning of RCM Mechanisms Using Angle and Distance Features

David Navarro-Alarcon, Hui Man Yip, Zerui Wang, Yun-hui Liu, Weiyang Lin and Peng Li

Abstract— **In this paper, we address the positioning problem of remote centre of motion (RCM) mechanisms with uncalibrated image feedback from a monocular camera. Nowadays, RCM mechanisms are widely used in minimally invasive robotic surgery due to their ability to distally rotate a tool around a fixed entry port; note that in most surgical applications, the tools are typically controlled by manual/teleoperated motion commands given by a human user. In this paper, we depart from the traditional manual control scheme and derive sensor-based methods to automatically position the manipulated tool using real-time image feedback. To this end, we first characterise the mechanism's 3-DOF configuration with the angle of the image projected tool and scalar distances between feature points. To cope with uncertainty in the camera's calibration parameters, we propose two gradient descent estimators that adaptively compute the unknown Jacobian matrix; the stability of these algorithms is proved with Lyapunov theory. Finally, we derive a kinematic image-based controller and evaluate its performance with several positioning experiments.**

I. Introduction

Nowadays, remote centre of motion (RCM) mechanisms are widely used in minimally invasive robotic surgery due to their ability to distally manipulate instruments around a fixed entry port [1]. In most applications, the manipulated tools are typically controlled by manual motion commands provided by the user, e.g. through a joystick or haptic interface (see [2]–[4]); the user determines the desired configuration based on the visual feedback provided by an endoscopic camera that observes the scene (see Fig. 1). In order to *automatically* control the pose of RCM mechanisms, a key step is to integrate this image feedback within the low-level positioning servo-loop. Our goal in this paper is precisely to develop these types of sensor-based automatic control schemes (particularly, we are interested in developing adaptive methods that can cope with uncertainties in the calibration parameters of the imaging system).

There are some works in the literature that address the automatic image-based positioning of laparoscopic instruments, see [5] for a comprehensive review. In [6], a method to control the position of a suturing tool with two uncalibrated cameras is presented—similarly, the control methods proposed in [7], [8] also use stereo cameras to position laparoscopic instruments. The single-camera 3D positioning problem has been studied by some researchers, e.g. Krupa et al. [9] proposed a method to regulate the relative distance between an instrument (that continuously projects a laser pattern) and an organ of interest. A path following controller for laparoscopic instruments is proposed in [10]; this single-camera approach computes the reference image points through the forward projection model of the desired trajectories and makes use of a special 6 point maker to estimate the location of the pivot point.

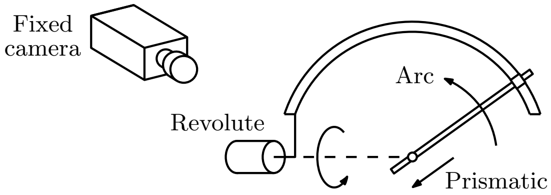

Fig. 1. Conceptual representation of a fixed camera observing the motion of a 3-DOF remote centre of motion mechanism.

To control the position of an instrument, the use of stereo vision seems a reasonable approach. However, note that monocular endoscopic cameras are still the most commonly used imaging systems in surgical procedures. One of the major challenges to position/guide instruments using a monocular camera is to obtain 3D pose information from a 2D image (it is well-known that the perspective projection of a point removes one DOF). The projected images of tools manipulated by RCM mechanisms satisfy certain geometric properties that allow us to characterise/register its pose, not in terms of Cartesian coordinates, but in terms of other features such as image lines and distances.

To contribute to this important problem, in this paper we propose a new method to servo-position RCM mechanisms using the visual feedback from an uncalibrated monocular camera. For that, we first construct an image feature vector using the angle of the projected tool and scalar distances between image points; this vector allows us to measure and independently control 3-DOF of the system. To coordinate the mechanism with the uncalibrated measurements, we derive energy-motivated algorithms that compute the Jacobian matrix in real-time without any knowledge of the calibration parameters. We experimentally validate the proposed control method with a robotic prototype.

The rest of the paper is organised as follows: Section II introduces the mathematical models, Section III derives the controller, Section IV presents the experiments, and Section V gives final conclusions.

II. Modelling

This section presents the mathematical models of the control problem addressed in the manuscript.

All the authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, HKSAR. Corresponding author e-mail: dnavarro@mae.cuhk.edu.hk

This work is supported in part by the HK RGC (grant numbers 415011, CUHK6/CRF/13G) and the HK ITF (grant number ITS/020/12FP).

978-1-4799-9993-4/15/$31.00 ©2015 IEEE

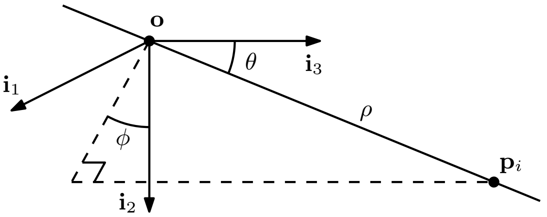

Fig. 2. The spherical coordinate system of the mechanism, where we measure the orientation angle ϕ over the $(\mathbf{i}_1, \mathbf{i}_2)$ plane.

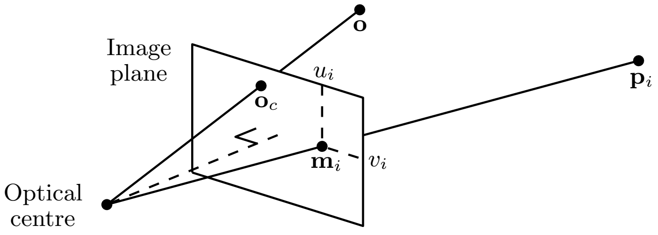

Fig. 3. Perspective projection of a point $\mathbf{p}_i$ onto the image plane.

A. Notation

We denote vectors by bold small letters (e.g. $\mathbf{v} \in \mathbb{R}^N$) and matrices by bold capital letters (e.g. $\mathbf{M} \in \mathbb{R}^{N \times M}$); we use $\mathbf{f} = \mathbf{f}(\mathbf{g})$ to denote both, the coordinate vector $\mathbf{f} \in \mathbb{R}^M$ and the vectorial function $\mathbf{f}(\mathbf{g}) : \mathbb{R}^N \mapsto \mathbb{R}^M$.

B. Kinematics

Consider an RCM mechanism with three controllable degrees of freedom, namely the pitch and yaw angles of the tool and its respective linear insertion. We use the standard spherical coordinates to represent the configuration of the manipulated tool; in this coordinate system, the RCM point represents the origin. Let the vector $\mathbf{q} \in \mathbb{R}^3$, denote the system's configuration which we define as

$$\mathbf{q} = \begin{bmatrix} \theta & \phi & \rho \end{bmatrix}^\intercal \tag{1}$$

for $\theta, \phi \in \mathbb{R}$ as the orientation angles, and $\rho \in \mathbb{R}$ as the insertion distance.

Using the configuration vector $\mathbf{q}$, we can calculate the Cartesian position of a point $\mathbf{p}_i \in \mathbb{R}^3$ located on the mechanism's tool as follows:

$$\mathbf{p}_i = \rho \begin{bmatrix} \sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi) & \cos(\theta) \end{bmatrix}^\intercal \tag{2}$$

Note that for any other feature point located along the tool, we can simply express its position vector as $\mathbf{p}_j = \lambda \mathbf{p}_i$, for $\lambda \in \mathbb{R}$ as a known scaling factor. See Fig. 2 for a graphical representation of the coordinate system.

C. Perspective Projection of a Point

Consider a fixed monocular camera that observes the manipulated tool. Let the vector $\mathbf{m}_i \in \mathbb{R}^2$ represents the 2D image measurements of the point $\mathbf{p}_i$

$$\mathbf{m}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\intercal \tag{3}$$

with pixel coordinates $u_i, v_i \in \mathbb{R}$. Along this work we model that the origin $\mathbf{o}_c \in \mathbb{R}^2$ of the pixel coordinates corresponds to the RCM point $\mathbf{o} \in \mathbb{R}^3$ projected onto the image plane. In a later section, we present a simple method to compute this origin point from measured image lines.

The perspective projection of the point $\mathbf{p}_i$ onto the image plane satisfies the well-known expression [11]

$$\begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = \frac{1}{z_i} \mathbf{C} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \tag{4}$$

where the matrix $\mathbf{C} \in \mathbb{R}^{3 \times 4}$ denotes the constant calibration matrix of the camera. The scalar functional $z_i \in \mathbb{R}$ represents the depth (measured along the optical axis of the camera) of the imaged point with respect to the camera's optical centre, and satisfies

$$z_i = \mathbf{c}_3^\intercal \begin{bmatrix} \mathbf{p}_i^\intercal & 1 \end{bmatrix}^\intercal \tag{5}$$

for $\mathbf{c}_i^\intercal \in \mathbb{R}^4$ as the ith row vector of the matrix $\mathbf{C}$. Fig. 3 depicts a conceptual representation of this model.

D. Feedback Geometric Features

To control the pose of the mechanism, in this work we use the measurements of two image points, which we group into an extended image measurement vector $\mathbf{m} \in \mathbb{R}^4$

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_1^\intercal & \mathbf{m}_2^\intercal \end{bmatrix}^\intercal \tag{6}$$

With these measurements, we construct a vector of geometric features which locally characterises the configuration (i.e. the perceived image pose) of the system. We denote this feature vector by $\mathbf{y} \in \mathbb{R}^3$, and locally compute its coordinates as follows:

$$\mathbf{y} = \begin{bmatrix} s & r & d \end{bmatrix}^\intercal \tag{7}$$

The scalar functional $s(\mathbf{m}_1, \mathbf{m}_2) : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}$ denotes the relative angle of the tool projected onto the image plane; we locally compute the angle from image measurements as

$$s = \operatorname{atan2}(v_2 - v_1, u_2 - u_1) \tag{8}$$

The scalar functional $r(\mathbf{m}_1) : \mathbb{R}^2 \mapsto \mathbb{R}$ represents the distance of the point $\mathbf{m}_1$ from the image origin; the scalar functional $d(\mathbf{m}_1, \mathbf{m}_2) : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}$ represents the projection of the distance between the two feature points; we compute these distance features as follows:

$$r = \|\mathbf{m}_1\|, \qquad d = \|\mathbf{m}_2 - \mathbf{m}_1\| \tag{9}$$

Fig. 4 depicts a conceptual representation of these features.

Remark 1. With the proposed feedback features, we can only control the mechanism in local regions away from singular configurations. This, however, does not impose severe constraints to many applications that require 'small' motions. Singularities arise e.g. when the manipulated tool is parallel/perpendicular to the image plane.

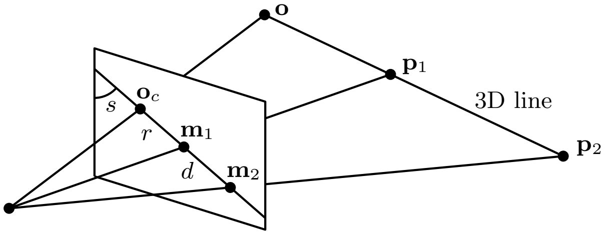

Fig. 4. Conceptual representation of the feedback geometric features.

E. Vector of Parameters

Lemma 1. *The projection matrix* $\mathbf{C}$ *of the RCM mechanism has 9 independent degrees of freedom.*

Proof. Consider the projection of the (arbitrary) RCM point $\mathbf{o} = \mathbf{0}_{3\times 1}$ onto the image plane

$$\begin{bmatrix}\mathbf{0}_{2\times 1}\\ 1\end{bmatrix} = \frac{1}{z_o}\mathbf{C}\begin{bmatrix}\mathbf{0}_{3\times 1}\\ 1\end{bmatrix} \tag{10}$$

for $z_o \in \mathbb{R}$ as the depth of $\mathbf{o}$. By solving the above equation and considering the constraint $z_o \neq 0$, we can show that

$$\begin{bmatrix}c_{1,4} & c_{2,4} & c_{3,4}\end{bmatrix}^{\intercal} \cong \kappa\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\intercal} \tag{11}$$

where $c_{i,j}$ represents the (i,j) element of the matrix $\mathbf{C}$. We use the symbol $\cong$ to express that we can only estimate $\mathbf{C}$ up-to an arbitrary scaling factor $\kappa \neq 0$. Expression (11) removes 3 degrees of freedom (2 come from the translated image origin, and 1 comes from the scale-independent homogeneous transformation) from the calibration matrix $\mathbf{C}$. ■

We set $c_{1,4} = c_{2,4} = 0$, $c_{3,4} = 1$, and group the 9 independent elements into the vector $\mathbf{a} \in \mathbb{R}^9$ as follows:

$$\mathbf{a} = \begin{bmatrix}c_{1,1} & c_{1,2} & c_{1,3} & c_{2,1} & c_{2,2} & c_{2,3} & \cdots & c_{3,3}\end{bmatrix}^{\intercal} \tag{12}$$

With the unknown calibration parameters $c_{i,j}$, we also construct the terms $\mathbf{G} \in \mathbb{R}^{2\times 3}$ and $\mathbf{g} \in \mathbb{R}^3$ as follows:

$$\mathbf{G} = \begin{bmatrix}c_{1,1} & c_{1,2} & c_{1,3}\\ c_{2,1} & c_{2,2} & c_{2,3}\end{bmatrix}, \quad \mathbf{g}^{\intercal} = \begin{bmatrix}c_{3,1} & c_{3,2} & c_{3,3}\end{bmatrix} \tag{13}$$

F. Problem Formulation

In this work, we aim to develop adaptive image-based methods to control the configuration of the mechanism. We derive the first-order kinematic plant by computing the time-derivative of the feature vector (7)

$$\dot{\mathbf{y}} = \mathbf{J}\mathbf{v} \tag{14}$$

where $\mathbf{v} = \dot{\mathbf{p}}_1 \in \mathbb{R}^3$ denotes the velocity control input (defined in terms of the known Cartesian velocity), and $\mathbf{J} \in \mathbb{R}^{3\times 3}$ denotes the standard Jacobian matrix

$$\mathbf{J} = \frac{\partial \mathbf{y}}{\partial \mathbf{m}}\frac{\partial \mathbf{m}}{\partial \mathbf{p}_1} \tag{15}$$

Problem. Given a constant 3-DOF feature vector $\mathbf{y}_d \in \mathbb{R}^3$, design an image-based kinematic control law $\mathbf{v}$ that asymptotically minimises the feedback error $\Delta\mathbf{y} = \mathbf{y} - \mathbf{y}_d \in \mathbb{R}^3$, without any knowledge of the parameters $\mathbf{a}$.

Remark 2. In order to control the motion of a system based on feedback images, the traditional visual servoing approach [12] requires exact knowledge of the calibration matrix $\mathbf{C}$. This requirement may pose problems to applications with unknown/uncertain parameters, e.g. to laparoscopic surgery where an endoscopic camera, with a typically uncertain pose, provides the visual feedback of the scene (note that surgeons may also manually roll the camera to adjust the scene).

III. Image-based Adaptive Controller

In this section, we describe the design of an image-based method to control the position of the mechanism.

A. Computation of the Imaged RCM

We numerically estimate the image position of the RCM point by iteratively computing a point $\mathbf{o}_c$ that minimises, in the least-squares sense, the distance to H previous image lines. Let $\mathbf{n}(t_i) \in \mathbb{R}^2$ denote a unit vector perpendicular to the line that passes through the image points $\widetilde{\mathbf{m}}_1(t_i)$ and $\widetilde{\mathbf{m}}_2(t_i)$, where $\widetilde{\mathbf{m}}_j$ denotes the original (i.e. the non origin translated) image coordinates of $\mathbf{m}_j$. We compute the point $\mathbf{o}_c$ as follows:

$$\mathbf{o}_c = \left(\sum_{i=1}^{H}\mathbf{n}(t_i)\mathbf{n}^{\intercal}(t_i)\right)^{-1}\sum_{i=1}^{H}\mathbf{n}(t_i)\mathbf{n}^{\intercal}(t_i)\widetilde{\mathbf{m}}_1(t_i) \tag{16}$$

B. Projection-based Adaptive Estimation

In this section, we present an algorithm to compute a vector of adaptive the parameters, here denoted by $\widehat{\mathbf{a}} \in \mathbb{R}^9$. In this method, we vary $\widehat{\mathbf{a}}$ in order to minimise the ith projection error $\mathbf{e}_i \in \mathbb{R}^2$ defined as [13]

$$\mathbf{e}_i = \widehat{\mathbf{G}}\mathbf{p}_1(t_i) - \mathbf{m}_1(t_i)\left(\widehat{\mathbf{g}}^{\intercal}\mathbf{p}_1(t_i) + 1\right) \tag{17}$$

where we construct the 'upper-hat' terms $\widehat{\mathbf{G}} \in \mathbb{R}^{2\times 3}$ and $\widehat{\mathbf{g}} \in \mathbb{R}^3$ from the adaptive vector $\widehat{\mathbf{a}} = [\widehat{a}_1, \ldots, \widehat{a}_9]^{\intercal}$, e.g. we compute $\widehat{\mathbf{g}}^{\intercal} = [\widehat{a}_7, \widehat{a}_8, \widehat{a}_9]$. To guarantee the estimation of parameters, in this algorithm we must use $P \geq 6$ observations (in other words, old measurements) of the image point $\mathbf{m}_1(t_i)$ and the position vector $\mathbf{p}_1(t_i)$, for $t_i = 1, 2, \ldots, P$.

Note that the error (17) physically represents the scaled difference between the estimated perspective projection and the measured image feedback of a feature point. To see this, we can use the intuitive 'hat notation' to equivalently express (17) as $\mathbf{e}_i = \widehat{z}_1(t_i)\left(\widehat{\mathbf{m}}_1(t_i) - \mathbf{m}_1(t_i)\right)$ [1].

We define the projection error as in (17) in order to obtain a measurable expression that satisfies linearity with respect to the parameters' error vector $\Delta\mathbf{a} = \widehat{\mathbf{a}} - \mathbf{a} \in \mathbb{R}^9$. To see this, we can rewrite (after some algebra) $\mathbf{e}_i$ as follows:

$$\begin{aligned}\mathbf{e}_i &= \widehat{\mathbf{G}}\mathbf{p}_1(t_i) - \mathbf{m}_1(t_i)\left(\widehat{\mathbf{g}}^{\intercal}\mathbf{p}_1(t_i) + 1\right)\\ &\qquad - \underbrace{z_1(t_i)\mathbf{m}_1(t_i) + \mathbf{m}_1(t_i)z_1(t_i)}_{(=0)},\\ &= \left(\widehat{\mathbf{G}} - \mathbf{G}\right)\mathbf{p}_1(t_i) - \mathbf{m}_1(t_i)\left(\widehat{\mathbf{g}}^{\intercal} - \mathbf{g}^{\intercal}\right)\mathbf{p}_1(t_i),\\ &= \mathbf{W}_i\Delta\mathbf{a}\end{aligned} \tag{18}$$

[1] In this simplified expression, we define $\widehat{z}_1(t_i) = \widehat{\mathbf{g}}^{\intercal}\mathbf{p}_1(t_i) + 1$ and $\widehat{\mathbf{m}}_1(t_i) = \frac{1}{\widehat{z}_1(t_i)}\widehat{\mathbf{G}}\mathbf{p}_1^{\intercal}(t_i)$

for a known regression matrix $\mathbf{W}_i(\mathbf{m}_1(t_i), \mathbf{p}_1(t_i)) \in \mathbb{R}^{2\times 9}$ that depends only on the measured observations, and has the following structure:

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{p}_1^\mathsf{T}(t_i) & \mathbf{0}_{1\times 3} & -u_1(t_i)\mathbf{p}_1^\mathsf{T}(t_i) \\ \mathbf{0}_{1\times 3} & \mathbf{p}_1^\mathsf{T}(t_i) & -v_1(t_i)\mathbf{p}_1^\mathsf{T}(t_i) \end{bmatrix} \tag{19}$$

Finally, we compute the scalar quadratic functional

$$Q = \frac{1}{2}\sum_{i=1}^{P} \mathbf{e}_i^\mathsf{T}\mathbf{e}_i \tag{20}$$

which quantifies the accuracy of the estimated projection model. To adaptively update the vector of parameters, we use the following gradient descent rule:

$$\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} = -\mathbf{\Gamma}\frac{\partial Q}{\partial \widehat{\mathbf{a}}}^\mathsf{T} \tag{21}$$

Proposition 1. *For 6 linearly independent sets of observation vectors $\mathbf{m}_1(t_i)$ and $\mathbf{p}_1(t_i)$, the projection-based update rule* (21) *enforces a stable (passive) system which asymptotically estimates the vector of parameters* $\mathbf{a}$.

Proof. To analyse the stability of the this method, consider the quadratic Lyapunov-like function

$$U = \frac{1}{2}\Delta\mathbf{a}^\mathsf{T}\mathbf{\Gamma}^{-1}\Delta\mathbf{a} \tag{22}$$

whose time-derivative satisfies

$$\dot{U} = -\Delta\mathbf{a}^\mathsf{T}\mathbf{M}^\mathsf{T}\mathbf{M}\Delta\mathbf{a} \leq 0 \tag{23}$$

for a constant matrix $\mathbf{M} \in \mathbb{R}^{2P\times 9}$ defined as

$$\mathbf{M} = \begin{bmatrix} \mathbf{W}_1^\mathsf{T} & \mathbf{W}_2^\mathsf{T} & \cdots & \mathbf{W}_P^\mathsf{T} \end{bmatrix}^\mathsf{T} \tag{24}$$

Equation (23) proves the numerical stability of the algorithm [14]. To prove the minimisation of the error $\Delta\mathbf{a}$, let us rearrange 9 row vectors of $\mathbf{M}$ as follows:

$$\begin{bmatrix} \mathbf{p}_1^\mathsf{T}(t_1) & * & * \\ \mathbf{p}_1^\mathsf{T}(t_2) & * & * \\ \mathbf{p}_1^\mathsf{T}(t_3) & * & * \\ \mathbf{0}_{1\times 3} & \mathbf{p}_1^\mathsf{T}(t_1) & * \\ \mathbf{0}_{1\times 3} & \mathbf{p}_1^\mathsf{T}(t_2) & * \\ \mathbf{0}_{1\times 3} & \mathbf{p}_1^\mathsf{T}(t_3) & * \\ \mathbf{p}_1^\mathsf{T}(t_4) & 0_{1\times 3} & -u_1(t_4)\mathbf{p}_1^\mathsf{T}(t_4) \\ \mathbf{p}_1^\mathsf{T}(t_5) & 0_{1\times 3} & -u_1(t_5)\mathbf{p}_1^\mathsf{T}(t_5) \\ \mathbf{p}_1^\mathsf{T}(t_6) & 0_{1\times 3} & -u_1(t_6)\mathbf{p}_1^\mathsf{T}(t_6) \end{bmatrix} \tag{25}$$

where $*$ denotes 'unimportant' terms. For 6 non-collinear sets of vectors $\mathbf{m}_1(t_i)$ and $\mathbf{p}_1(t_i)$, the matrix $\mathbf{M}$ has full column rank, thus (23) is dissipative with respect to $\Delta\mathbf{a}$ [14]. ■

C. Flow-based Adaptive Estimation

In this section, we present an algorithm which, in contrast to the previous projection-based method, directly estimates the Jacobian matrix from optical flow measurements. The proposed algorithm iteratively computes a varying matrix $\widehat{\mathbf{J}} \in \mathbb{R}^{3\times 3}$ that closely approximates the following input-output differential relation:

$$\dot{\mathbf{y}}(t) \approx \widehat{\mathbf{J}}(t)\dot{\mathbf{p}}_1(t) \tag{26}$$

This approach resembles the rationale of other estimation algorithms, e.g. the Broyden method (see [15]–[17] for some applications in visual servoing).

To design the online estimation algorithm, we first define the scalar energy functional

$$F = \frac{1}{2}\left\|\widehat{\mathbf{J}}\dot{\mathbf{p}}_1 - \dot{\mathbf{y}}\right\|^2 \tag{27}$$

which provides a measurable metric of the algorithm's accuracy. Note that we can directly compute this functional from sensory feedback (more specifically, by numerically differentiating $\mathbf{y}$), the estimator's state and velocity input.

With the proposed flow-based method, we iteratively vary the (i, j) element of the Jacobian matrix $\widehat{\mathbf{J}}$ with the following gradient descent rule:

$$\frac{\mathrm{d}}{\mathrm{d}t}\widehat{J}_{ij} = -\gamma\frac{\partial F}{\partial \widehat{J}_{ij}} \tag{28}$$

where $\gamma \in \mathbb{R}$ denotes a positive tuning gain. This update rule continuously minimises the difference between the visually measured flow vector $\dot{\mathbf{y}}$ and the flow numerically computed with $\widehat{\mathbf{J}}\dot{\mathbf{q}}$.

In order to analyse the stability of the update rule (28), we make the following modelling assumptions: we consider that the robot moves at low-speed such that the elements of the Jacobian matrix change slowly. Thus, we *locally* model that its time rate satisfy $\frac{\mathrm{d}}{\mathrm{d}t}J_{ij} \approx 0$. Note that this approximated model represents applications where a 'high-speed' control loop commands 'low-speed' motions of a mechanism.

Proposition 2. *For slow and continuous motion of the robot, the flow-based update rule* (28) *provides a stable (passive) computation of the terms $\widehat{J}_{ij}$.*

Proof. Consider the ith (for $i = 1, 2, 3$) quadratic energy-like function

$$R_i = \frac{1}{2\gamma}\sum_{j=1}^{3}\Delta J_{ij}^2 \tag{29}$$

where $\Delta J_{ij} = \widehat{J}_{ij} - J_{ij} \in \mathbb{R}$ denotes the estimation error. The time-derivative of R_i yields

$$\dot{R}_i = -\left(\sum_{j=1}^{3}\Delta J_{ij}\dot{p}_{1,j}\right)^2 \tag{30}$$

which proves passivity of the closed-loop systems [14], and therefore, the numerical stability of the flow-based estimator. ■

Remark 3. Note that this numerical method does not aim to identify the true camera calibration matrix $\mathbf{C}$. The sole objective of the algorithm is to compute a time-varying matrix that closely approximates, at every time instant t, the differential mapping given in (26); we can see this property in (30), where a flow error drives the monotonic minimisation of the Lyapunov function R_i.

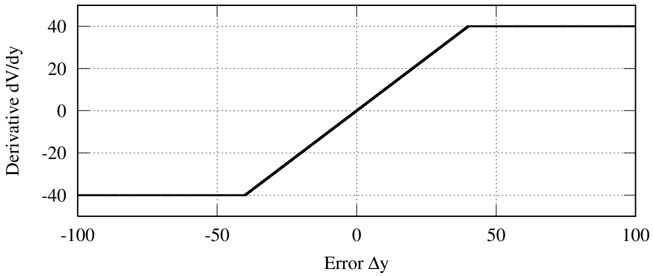

Fig. 5. The saturated potential action $\partial V_i / \partial y_i$, computed with $b_i = 40$.

D. Uncalibrated Image-based Controller

To control the motion of the RCM mechanism based on images, we use a Jacobian-based kinematic control design [18]. To implement the controller, we first compute the estimated Jacobian matrix $\widehat{\mathbf{J}}$ with any of the previous estimation algorithms (i.e. directly with the rule (28), or constructed via the adaptive parameters $\widehat{\mathbf{a}}$ with the rule (21)). We compute the velocity input with the following control law:

$$\mathbf{v} = -k\widehat{\mathbf{J}}^{-1} \frac{\partial V}{\partial \mathbf{y}}^{\mathsf{T}} \tag{31}$$

where $k \in \mathbb{R}$ denotes a positive feedback gain, and the scalar $V(\Delta \mathbf{y}) = \sum V_i(\Delta y_i) : \mathbb{R}^3 \mapsto \mathbb{R}$ represents a potential energy function with positive-definite terms V_i computed as

$$V_i = \begin{cases} \frac{1}{2}\|\Delta y_i\|^2, & \text{for} \quad \|\Delta y_i\| < b_i \\ \frac{1}{2} b_i \|\Delta y_i\|, & \text{for} \quad \|\Delta y_i\| \geq b_i \end{cases} \tag{32}$$

for $b_i \in \mathbb{R}$ as a positive saturation scalar (see Fig. 5 for a graphical representation of $\partial V_i / \partial y_i$).

To analyse the stability properties of this uncalibrated kinematic controller, we assume that the online estimation algorithm accurately computes the unknown Jacobian matrix, in other words, for $\mathbf{J}\widehat{\mathbf{J}}^{-1} \approx \mathbf{I}_{3\times 3}$, the image-based controller (31) enforces the closed-loop system

$$\dot{\mathbf{y}} = -k \frac{\partial V}{\partial \mathbf{y}}^{\mathsf{T}} \tag{33}$$

Consider $V(\Delta \mathbf{y})$ as a Lyapunov function for (33); by computing its time-derivative we obtain $\dot{V} = -k\|\partial V / \partial \mathbf{y}\|^2 \leq 0$, which proves asymptotic stability of the error $\Delta \mathbf{y}$

IV. Experiments

This section presents the conducted experimental study.

A. Setup

The system that we use is a laboratory prototype designed for surgical procedures [19] (see Fig. 6). The joints of this system physically represent the coordinates θ, ϕ, and ρ. We control the velocity of this mechanism using a Linux-based PC with an embedded Galil DMC-4040 control board. Visual feedback of the scene is captured with a Point Grey Firefly camera at 60 fps; we use the OpenCV libraries to process the captured images and to track the feature points (represented by two artificial markers placed on the tool). The overall motion control system has a servo-loop of 10 milliseconds.

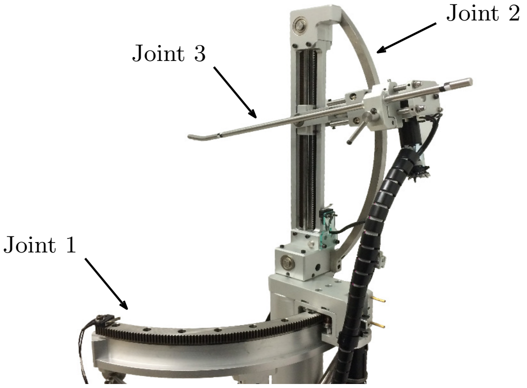

Fig. 6. The RCM robotic prototype used for the experiments.

We implement the estimators with no knowledge of the calibration matrix. Thus, we initialise with null terms the adaptive vector $\widehat{\mathbf{a}}(0) = [0, \ldots, 0]^{\mathsf{T}}$ and matrix $\widehat{\mathbf{J}}(0) = \mathbf{0}_{3\times 3}$. We implement the adaptive rules with the tuning gains $\mathbf{\Gamma} = \mathrm{diag}(10^{-5}\mathbf{I}_{6\times 6}, 10^{-8}\mathbf{I}_{3\times 3})$ and $\gamma = 0.001$, and the velocity controller with gain $k = 0.1$ and bounds $b_i = 10$. To compute the rule (21), we collect $P = 10$ random observation points. We iteratively compute the imaged RCM point $\mathbf{o}_c$ with $H = 50$ old lines.

B. Regulation Experiments

We test the performance of the projection-based and flow-based adaptive methods with two regulation tasks, both with initial errors of $[|\Delta s|, |\Delta r|, |\Delta d|] = [40, 80, 60]$. Fig. 7 and Fig. 8 show the captured initial and final images of these experiments. The overlaid blue lines represent the feedback features $\mathbf{y}$ and the overlaid black lines represent the target features $\mathbf{y}_d = [s_d, r_d, d_d]^{\mathsf{T}}$. Fig. 9 and Fig. 10 show the magnitude of the error coordinates $\Delta \mathbf{y} = [\Delta s, \Delta r, \Delta d]^{\mathsf{T}}$ obtained with both experiments.

Fig. 11 and Fig. 12 qualitatively depict the properties of both gradient descent estimators; e.g. Fig. 11 compares the profiles of the image point $\mathbf{m}_1 = [u_1, v_1]^{\mathsf{T}}$ with the point

$$\boldsymbol{\pi} = [\pi_u, \pi_v]^{\mathsf{T}} = \frac{\widehat{\mathbf{G}}\mathbf{p}_1}{(\widehat{\mathbf{g}}^{\mathsf{T}}\mathbf{p}_1 + 1)} \tag{34}$$

computed with the projection-based estimator during the experiment shown in Fig. 9; on the other hand, Fig. 12 compares the profiles of the measured optical flow vector $\dot{\mathbf{y}}$ with the vector

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^{\mathsf{T}} = \widehat{\mathbf{J}}\dot{\mathbf{p}}_1 \tag{35}$$

computed with the flow-based estimator during the experiment shown in Fig. 10. Note that since $\dot{\mathbf{y}}$ is computed by numerical differentiation, its profile presents considerable noise. This, however, does not affect the performance of the estimator, since the use of a 'small' gain γ helps to remove noise from the output flow estimation (compare the red and black profiles).

From these results, we can see that the both (conceptually different) gradient descent adaptive algorithms can be used

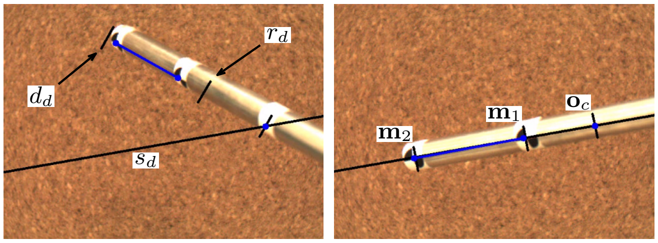

Fig. 7. Initial and final captured images of the projection-based experiment.

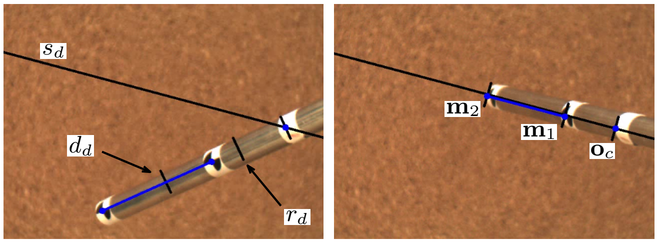

Fig. 8. Initial and final captured images of the flow-based experiment.

to compute the Jacobian matrix in real-time. The flow-based method provides a more accurate (instantaneous) Jacobian computation, however in contrast with the projection-based method, it cannot asymptotically estimate the calibration parameters $\mathbf{a}$.

C. Image-based 3D Pose Control

With the proposed image feature vector $\mathbf{y}$, we can (locally) register the 3D pose of the tool with a single monocular camera. In this experiment, we test the accuracy of our image-based 3D positioning method. For that, we take a learning-by-showing approach, i.e. we first bring the tool to desired configurations and register its reference image features $\mathbf{y}_d$, then, we use the image-based controller to position the tool into the reference 3D poses. For implicitly, we conduct this experiment using the flow-based adaptive estimator.

Fig. 13 depicts snapshots of the control experiment; the upper-left figure depicts the initial configuration, where each of the overlaid black lines represent the reference (i.e. previously registered) image features. Fig. 14 shows the magnitude of the 3D pose error $\|\Delta\mathbf{p}\| = \|\mathbf{p}_2 - \mathbf{p}_d\|$ obtained during this experiment, where $\mathbf{p}_d$ denotes reference Cartesian position (note that the vector $\mathbf{p}_d$ is never used in our control method, here we only use this reference vector for comparison purposes). These results shows that the proposed controller can visually reposition the tool into the reference configurations without any additional devices (e.g. a laser as in [9]) or pose estimation algorithms (e.g. [20]), with a positioning error smaller than 4mm.

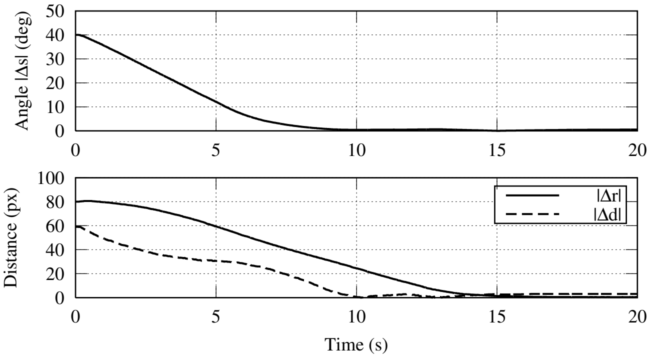

Fig. 9. Error coordinates of the experiment shown in Fig. 7.

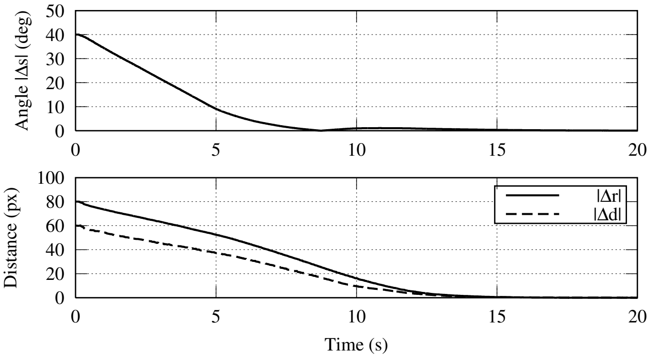

Fig. 10. Error coordinates of the experiment shown in Fig. 8.

V. Conclusions

In this paper, we presented a method to control the 3-DOF configuration of an RCM mechanism using uncalibrated image feedback from a single monocular camera. In our method, we first characterised the system's pose using angle and distance features. Next, to cope with the uncertain camera parameters, we proposed two gradient descent algorithms to online estimate the Jacobian matrix. Then, we derived a kinematic control law to automatically drive the system into desired image features. Finally, we presented experimental results to validate our method.

The method that we propose allows to register a 3D configuration from images of a single camera. Although our controller is driven by the image feature error $\Delta\mathbf{y}$, our experiments show that the steady-state pose error $\|\Delta\mathbf{p}\|$ of the system is less than 4mm. Certainly, the features that we use are locally defined, however, many laparoscopic applications do not require the instrument to perform 'large motions'. The use of the gradient descent algorithms to iteratively compute the Jacobian matrix allows to, not only control the mechanism with uncalibrated feedback, but also to arbitrarily relocate the camera to observe different regions of interest (which is common during surgical procedures).

As future work, we would like to conduct image-based experiments in an ex-vivo laparoscopic surgical setting; we want to directly use the feedback of a conventional endoscopic camera to control the motion of a robotised surgical instrument.

References

[1] C.-H. Kuo, J. S. Dai, and P. Dasgupta, "Kinematic design considerations for minimally invasive surgical robots: an overview," *Int. J. Med. Robot. Comp.*, vol. 8, no. 2, pp. 127–145, Jan. 2012.

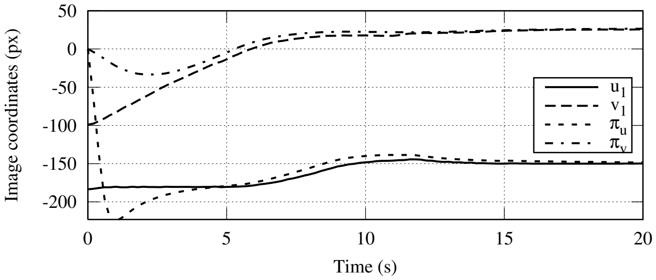

Fig. 11. Online estimation of the projection model.

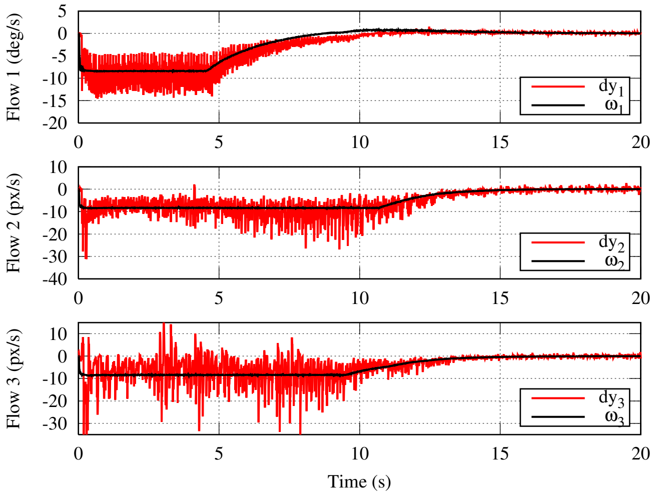

Fig. 12. Online estimation of the output flow.

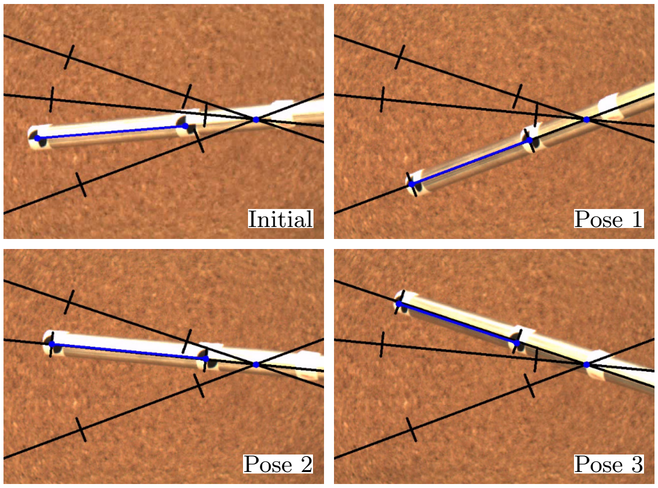

Fig. 13. Image-based registration of different 3D poses.

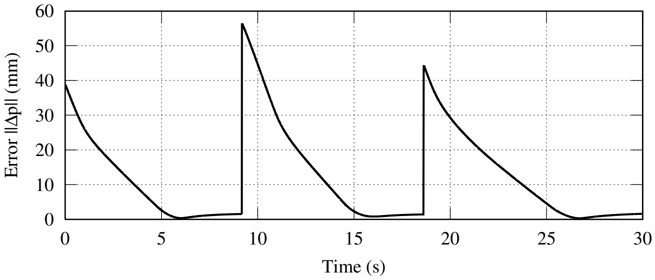

Fig. 14. Pose error $\|\Delta \mathbf{p}\|$ for the different experiments shown in Fig. 13.

[2] C. Delgorge, F. Courreges, L. Bassit, C. Novales, C. Rosenberger, N. Smith-Guerin, C. Bru, R. Gilabert, M. Vannoni, G. Poisson, and P. Vieyres, "A tele-operated mobile ultrasound scanner using a light-weight robot," *IEEE Trans. Inf. Technol. Biomed.*, vol. 9, no. 1, pp. 50–58, Mar. 2005.

[3] K.-Y. Kim, H.-S. Song, J.-W. Suh, and J.-J. Lee, "A novel surgical manipulator with workspace-conversion ability for telesurgery," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 1, pp. 200–211, Feb. 2013.

[4] D. Stoianovici, C. Kim, F. Schafer, C.-M. Huang, Y. Zuo, D. Petrisor, and M. Han, "Endocavity ultrasound probe manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 914–921, Jun. 2013.

[5] M. Azizian, M. Khoshnam, N. Najmaei, and R. V. Patel, "Visual servoing in medical robotics: a survey. Part I: endoscopic and direct vision imaging — techniques and applications," *Int. J. Med. Robot. Comp.*, vol. 10, no. 3, pp. 263–274, 2014.

[6] P. Hynes, G. Dodds, and A. Wilkinson, "Uncalibrated visual-servoing of a dual-arm robot for mis suturing," in *Proc. IEEE Int. Conf. Biomedical Robotics and Biomechatronics*, 2006, pp. 420–425.

[7] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 1837–1842.

[8] D.-L. Chow, R. Jackson, M. Cavusoglu, and W. Newman, "A novel vision guided knot-tying method for autonomous robotic surgery," in *Proc. Int. Conf. Automation Science and Engg.*, 2014, pp. 504–508.

[9] A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE Trans. Robot. and Automat.*, vol. 19, no. 5, pp. 842–853, Oct. 2003.

[10] F. Nageotte, P. Zanne, C. Doignon, and M. de Mathelin, "Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2006, pp. 2364–2369.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2004.

[12] F. Chaumette and S. Hutchinson, "Visual servo control. Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[13] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, Aug. 2006.

[14] J.-J. Slotine and W. Li, *Applied Nonlinear Control*, 1st ed. Upper Saddle River, NJ: Prentice Hall, 1991.

[15] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 1, 1994, pp. 186–193.

[16] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, 1997, pp. 2874–2880.

[17] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Trans. Robot.*, vol. 26, no. 6, pp. 1457–1468, Aug. 2013.

[18] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. 10, no. 2, pp. 47–53, Jun. 1969.

[19] H. M. Yip, P. Li, D. Navarro-Alarcon, Z. Wang, and Y. hui Liu, "A new circular-guided remote center of motion mechanism for assistive surgical robots," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, Dec 2014, pp. 217–222.

[20] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar feature points," *Comput. Vis. Image Underst.*, vol. 63, no. 3, pp. 495–511, May 1996.