2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A Dynamic and Uncalibrated Method to Visually Servo-control Elastic Deformations by Fully-constrained Robotic Grippers

David Navarro-Alarcon and Yun-hui Liu

Abstract— **In this paper, we address the set-point deformation control of elastic objects by fully-constrained grippers. We propose an uncalibrated Lyapunov-based algorithm that iteratively estimates the deformation Jacobian matrix, with no prior knowledge of the deformation and camera models. With this new method we show how, by combining pose information of the grippers with several visual measurements, we can independently control elastic deformations of unknown objects. We report experiments with a 6-DOF robot manipulator to validate this control approach.**

I. Introduction

In recent years, the problem of the active deformation control of compliant objects has received significant attention from the robotics research community. This interest in controlling deformations seems to come from the recent advances in surgical robotics [1], the progress in home/personal-robotics [2], and the automation of new applications that require to manipulate soft bodies, e.g. handling food materials [3], or suturing [4]. Note that in many applications (specially those in the medical field), the manipulated objects usually have uncertain deformation properties; this situation clearly complicates the practical implementation.

The research community has made significant contributions since the work [5], which to our knowledge, reports one of the earliest efforts to comprehensively address this topic. Some researchers have studied the active shaping of rheological materials, see e.g. [6], [7]. To control deformations, these methods must first identify the object's visco-elastic properties. For objects that exhibit elastic deformations only, some researchers have proposed explicit feedback control methods to indirectly position deformable feature points, see e.g. [8]–[11]. These model-based controllers also require prior calibration of deformation and vision models.

Irrespectively of the material's properties, most controllers in the literature require a-priori knowledge of a deformation model, albeit approximated. Very few works address the simultaneous online estimation and control of elastic deformations (note that standard visual servoing, e.g. classical [12] or uncalibrated methods [13], do not address active deformation). Moreover, most existing methods only use the manipulator's linear displacements to actively control deformations; since these control algorithms do not consider rotations/torsions, they can only impose a limited set of shapes onto the object. For example, [7] presents a method to control one-dimensional deformations (i.e. linear compressions), whereas [8]–[10] formulate the deformation controller considering plane motion of the manipulator (these methods can only control up-to two degree-of-freedom (DOF) with a single manipulator). Recently, [11] proposed a method to control deformations with the 6-DOF motion of the end-effector; this controller, however, requires a-priori knowledge of the object's deformation properties.

In this paper, we present a new image-based approach to servo-control elastic deformations. By incorporating the gripper's attitude and several visual measurements, our method can independently control elastic deformations with the 6-DOF motion of the manipulator—something we can not achieve with existing controllers including our previous formulations [14], [15]. The uncalibrated method that we present does not require identification of the deformation and camera models. For that, we derive a Lyapunov-based algorithm that computes the unknown deformation Jacobian matrix in real-time. We report a 6-DOF experimental study to validate the feasibility of our new set-point deformation controller.

The rest of this manuscript has the following structure: In Section II we derive the mathematical models. Section III details our control method. In Section IV we report the experiments. Section V gives conclusions and future work.

II. Mathematical Modelling

A. Manipulator model

Consider a serial non-redundant manipulator with revolute joints and an exactly known kinematic structure. We denote the vectors of joint displacements by $\mathbf{q} \in \mathbb{R}^n$ and end-effector pose (position + orientation) by $\mathbf{x} \in \mathbb{R}^n$. The time-derivative of the pose vector yields the expression

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}}, \tag{1}$$

where the matrix $\frac{\partial \mathbf{x}}{\partial \mathbf{q}}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ denotes the analytical Jacobian matrix of the manipulator.

In this paper we only consider kinematically-controlled manipulators, e.g. as reported in [16]. Therefore, the angular velocity $\dot{\mathbf{q}}$ physically stands for the control input to the system. Note that in real experimental systems this velocity input typically presents saturation.

B. Deformation model

We consider the case where the robot manipulator physically interacts with soft objects that exhibit elastic deformations only. To actively modify the object's shape, the

D. Navarro-Alarcon and Y.-H. Liu. Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin NT, HKSAR. dnavarro,yhliu[at]mae.cuhk.edu.hk

This work has support from the the Hong Kong RGC (grant numbers 415011, CUHK6/CRF/13G); the Hong Kong ITF (grant number ITS/475/09), and the Shun Hing Institute of Advanced Engineering.

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

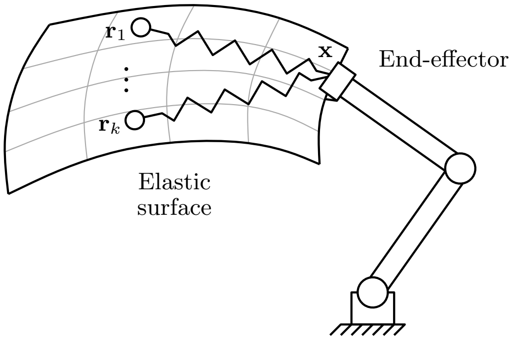


Fig. 1. Conceptual representation of the object's deformation model.

manipulator has a robotic gripper which rigidly grasps the elastic body in a fully-constrained manner. Physically, this situation means that the motion of the manipulator's tip (in any linear or rotational direction) produces deformations on the body.

To represent the elastic configuration of the object, we use k feature points $\mathbf{r}_i \in \mathbb{R}^3$, for $i = 1 \dots, k$, located over the surface of the object (see Fig. 1 for a conceptual representation). We *locally* model the quasi-static relation between the pose $\mathbf{x}$ and each feature point $\mathbf{r}_i$ by

$$\mathbf{r}_i = \mathbf{C}_i \delta\mathbf{x}, \tag{2}$$

where $\mathbf{C}_i \in \mathbb{R}^{3\times n}$ represents a constant deformation matrix that maps the displacements of the n-dimensional pose to the deformation of the feature points. The vector $\delta\mathbf{x} = \mathbf{x} - \overline{\mathbf{x}} \in \mathbb{R}^n$ models the relative pose displacement, with $\overline{\mathbf{x}}$ as a constant reference. For ease of presentation, we group all the feature points in the extended feature vector

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1^\top & \cdots & \mathbf{r}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{3k}. \tag{3}$$

C. Visual feedback model

Consider m uncalibrated fixed cameras that observe the object. In our approach, we use multiple independent vision sensors to monitor (and eventually control) the object's shape from different perspectives. For ease of presentation, we assume that the point $\mathbf{r}_i$ projects onto one image plane only[1]. We denote the visual feedback of the point $\mathbf{r}_i$ by

$$\mathbf{s}_i = \begin{bmatrix} \mu_i & \nu_i \end{bmatrix}^\top \in \mathbb{R}^2, \tag{4}$$

where $\mu_i \in \mathbb{R}$ and $\nu_i \in \mathbb{R}$ denote image pixel coordinates. We group all the image points $\mathbf{s}_i$ into the extended vector

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_1^\top & \cdots & \mathbf{s}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{2k}. \tag{5}$$

See Fig. 2 for a conceptual representation of this setup.

In our approach, we *locally* model the projection of the feature point $\mathbf{r}_i$ onto the jth image plane by

$$\mathbf{s}_i = \mathbf{M}_j \mathbf{r}_i + \mathbf{b}_j \in \mathbb{R}^2, \tag{6}$$

with the matrix $\mathbf{M}_j \in \mathbb{R}^{2\times 3}$ and vector $\mathbf{b}_j \in \mathbb{R}^2$ as the constant calibration terms. We must remark that we use this affine model [17] to obtain a simple and linearly parametrisable expression of the visual deformation feedback; in Section III we derive an online estimator that exploits this useful property (we show that the use of this simple model does not impose severe implementation constraints to our method since it *continuously updates* variable parameters based on image measurements).

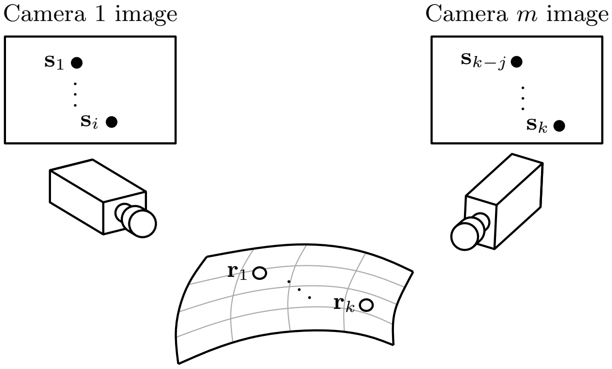


Fig. 2. Conceptual representation of the k visual feedback points provided by the m uncalibrated cameras, where $1 \leq i \leq j \leq k$.

The time-derivative of the extended visual feedback vector $\mathbf{s}$ yields the differential expression

$$\dot{\mathbf{s}} = \mathbf{L}\dot{\mathbf{x}}, \tag{7}$$

which relates the input motion of the grippers with the output optical flow. We define the projection-deformation constant matrix $\mathbf{L} \in \mathbb{R}^{2k\times n}$ by

$$\mathbf{L} = \begin{bmatrix} \mathbf{M}_1\mathbf{C}_1 \\ \vdots \\ \mathbf{M}_m\mathbf{C}_k \end{bmatrix}, \tag{8}$$

where m denotes the number of cameras, and k the number of feature points. Now, let us group all the elements of the matrix $\mathbf{L}$ into the vector of constant parameters

$$\boldsymbol{\theta} = \begin{bmatrix} L_{1,1} & L_{1,2} & \cdots & L_{2k,n-1} & L_{2k,n} \end{bmatrix}^\top \in \mathbb{R}^g, \tag{9}$$

where $g = 2kn \in \mathbb{R}$ represents the number of unknown components, and $L_{i,j} \in \mathbb{R}$ denotes an element at the ith row jth column of the matrix $\mathbf{L}$.

D. Deformation feature

To obtain a quantitative metric of the object's shape (and thus avoid the open-loop response of robot-centred methods, e.g. [18]), we introduce the vision-dependent function

$$\mathbf{y} = \mathbf{y}(\mathbf{s}) \in \mathbb{R}^h; \quad \text{for } h \leq n, \tag{10}$$

which we call the deformation feature vector. The coordinates of this smooth function can represent, e.g. visual displacements, angles, curvatures, line features, etcetera (see [15]). By time-differentiating this function we obtain

$$\dot{\mathbf{y}} = \mathbf{J}(\mathbf{s})\dot{\mathbf{x}}, \tag{11}$$

[1] We can easily extend this formulation to multiple feedback of one point.

where $\mathbf{J}(\mathbf{s}) \in \mathbb{R}^{h\times n}$ represents the *deformation Jacobian matrix* and has the following definition:

$$\mathbf{J}(\mathbf{s}) = \frac{\partial \mathbf{y}}{\partial \mathbf{s}}(\mathbf{s})\mathbf{L}. \tag{12}$$

Note that we can not exactly compute this matrix since it contains unknown terms, namely, the matrix $\mathbf{L}$.

III. Controller Design

As with traditional visual servoing approaches [12], we design our controller in terms of the end-effector velocities, therefore, we define $\mathbf{v} = \dot{\mathbf{x}} \in \mathbb{R}^n$ as the new control variable.

Problem. Given a constant deformation feature $\mathbf{y}_d \in \mathbb{R}^h$, design an uncalibrated velocity controller $\mathbf{v}$ that asymptotically minimises the error $\Delta\mathbf{y} = \mathbf{y} - \mathbf{y}_d \in \mathbb{R}^h$, with no a-priori information of the deformation and camera models.

A. Lyapunov-based estimator

In order to minimise the deformation error $\Delta\mathbf{y}$, we must first estimate how the gripper's linear and rotational motions actively deform the elastic object. For that, in this section we present an adaptive algorithm that computes an online estimation of the vector of parameters $\boldsymbol{\theta}$; we denote this adaptive vector by $\widehat{\boldsymbol{\theta}} \in \mathbb{R}^g$. Note that with the vector $\widehat{\boldsymbol{\theta}}$ we can construct an estimation of the unknown projection-deformation matrix $\mathbf{L}$, and in turn, an estimation of the deformation Jacobian matrix $\mathbf{J}(\mathbf{s})$.

To introduce our adaptive algorithm, consider first the flow estimation error vector

$$\mathbf{e} = \mathbf{f} - \dot{\mathbf{y}} \in \mathbb{R}^h, \tag{13}$$

for a vector $\mathbf{f} \in \mathbb{R}^h$ computed as

$$\mathbf{f} = \frac{\partial \mathbf{y}}{\partial \mathbf{s}}(\mathbf{s})\widehat{\mathbf{L}}\dot{\mathbf{x}}. \tag{14}$$

The matrix $\widehat{\mathbf{L}} \in \mathbb{R}^{2k\times n}$ represents an estimation of the unknown matrix $\mathbf{L}$. The upper symbol $\widehat{\star}$ indicates that we construct this term with the variable parameters $\widehat{\boldsymbol{\theta}}$. By substituting the right-hand sides of the flows (11) and (14) into the error (13), we can equivalently re-write the error as

$$\begin{aligned}\mathbf{e} &= \frac{\partial \mathbf{y}}{\partial \mathbf{s}}(\mathbf{s})\left(\widehat{\mathbf{L}} - \mathbf{L}\right)\dot{\mathbf{x}}, \\ &= \mathbf{A}(\mathbf{s}, \dot{\mathbf{x}})\Delta\boldsymbol{\theta}, \end{aligned} \tag{15}$$

where $\Delta\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}} - \boldsymbol{\theta} \in \mathbb{R}^g$ denotes the parameters' estimation error, and $\mathbf{A}(\mathbf{s}, \dot{\mathbf{x}}) \in \mathbb{R}^{h\times g}$ represents a known regression matrix which has the following simple definition:

$$\mathbf{A}(\mathbf{s}, \dot{\mathbf{x}}) = \frac{\partial \mathbf{y}}{\partial \mathbf{s}}(\mathbf{s}) \begin{bmatrix} \dot{\mathbf{x}}^\top & \cdots & \mathbf{0}_{1\times n} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{1\times n} & \cdots & \dot{\mathbf{x}}^\top \end{bmatrix}. \tag{16}$$

Proposition 1: Consider that for slow (viz. saturated) and smooth motion of the manipulator, the bounds $\|\dot{\mathbf{x}}\| \leq \varepsilon_1$ and $\|\ddot{\mathbf{x}}\| \leq \varepsilon_2$ simultaneously satisfy for positive scalars $\varepsilon_j \in \mathbb{R}$. In this situation, the parameter's update rule

$$\dot{\widehat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\mathbf{A}^\top(\mathbf{s}, \dot{\mathbf{x}})\mathbf{K}\mathbf{e}, \tag{17}$$

with symmetric and positive matrices $\boldsymbol{\Gamma} \in \mathbb{R}^{g\times g}$ and $\mathbf{K} \in \mathbb{R}^{h\times h}$, stably computes a matrix $\widehat{\mathbf{L}}$ that asymptotically minimises the error $\mathbf{e}$.

Proof: Considering the linearly parametrised error (15), we can re-write the update rule as

$$\dot{\widehat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\mathbf{A}^\top(\mathbf{s}, \dot{\mathbf{x}})\mathbf{K}\mathbf{A}(\mathbf{s}, \dot{\mathbf{x}})\Delta\boldsymbol{\theta}. \tag{18}$$

To prove the stability of the online estimator, we introduce the energy-like functional

$$\mathcal{V} = \frac{1}{2}\Delta\boldsymbol{\theta}^\top\boldsymbol{\Gamma}^{-1}\Delta\boldsymbol{\theta} \in \mathbb{R}, \tag{19}$$

whose time-derivative along (18) satisfies

$$\begin{aligned}\dot{\mathcal{V}} &= -\Delta\boldsymbol{\theta}^\top\mathbf{A}^\top(\mathbf{s}, \dot{\mathbf{x}})\mathbf{K}\mathbf{A}(\mathbf{s}, \dot{\mathbf{x}})\Delta\boldsymbol{\theta}, \\ &= -\mathbf{e}^\top\mathbf{K}\mathbf{e}, \\ &\leq 0. \end{aligned} \tag{20}$$

From the Lyapunov's direct method [19], we see that the update rule provides a stable (thus bounded) computation of $\widehat{\boldsymbol{\theta}}$. Note that since $\|\dot{\mathbf{x}}\|$ and $\|\ddot{\mathbf{x}}\|$ have upper bounds, then, simple computations can show that $\|\ddot{\mathcal{V}}\|$ has also upper bounds. From the Barbalat's lemma [19], we prove that as time approaches infinity, the flow error $\mathbf{e} \to \mathbf{0}_{h\times 1}$. ■

In our method, we compute the Jacobian matrix in real-time and with no model calibration as follows:

$$\widehat{\mathbf{J}}(\mathbf{s}) = \frac{\partial \mathbf{y}}{\partial \mathbf{s}}(\mathbf{s})\widehat{\mathbf{L}} \in \mathbb{R}^{h\times n}. \tag{21}$$

Remark 1: The update rule (17) does not guarantee the identification of the true parameters $\boldsymbol{\theta}$ or the Jacobian matrix $\mathbf{J}(\mathbf{s})$. We design our iterative algorithm with the objective of *continuously* satisfying the kinematic relation:

$$\dot{\mathbf{y}} = \widehat{\mathbf{J}}(\mathbf{s})\dot{\mathbf{x}}. \tag{22}$$

This approach resembles the method in [20].

B. Dynamic velocity control

In our method, we use the *smooth* velocity control input

$$\mathbf{v} = \widehat{\mathbf{J}}^{+}(\mathbf{s})\mathbf{p}, \tag{23}$$

where the matrix $\widehat{\mathbf{J}}^{+}(\mathbf{s}) \in \mathbb{R}^{n\times h}$ stands for the Moore–Penrose pseudoinverse. The vector $\mathbf{p} \in \mathbb{R}^h$ represents a dynamic state variable—with physical interpretation of visual momenta, see [21]—which we compute by the rule

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{Q}}{\partial \mathbf{y}}^\top(\Delta\mathbf{y}) - \mathbf{B}\mathbf{p}, \tag{24}$$

for $\mathcal{Q}(\Delta\mathbf{y}) \in \mathbb{R}$ as a positive-definite function with a unique equilibrium at $\Delta\mathbf{y} = \mathbf{0}_{h\times 1}$. The symmetric and positive matrix $\mathbf{B} \in \mathbb{R}^{h\times h}$ represents damping.

Proposition 2: The dynamic velocity control input (23) asymptotically minimises the deformation error $\Delta\mathbf{y}$.

Proof: By substituting the controller (23) into the deformation plant (22) we obtain the closed-loop system

$$\begin{bmatrix} \dot{\mathbf{y}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{h\times h} & \mathbf{I}_{h\times h} \\ -\mathbf{I}_{h\times h} & -\mathbf{B} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial \mathbf{y}}^\top \\ \frac{\partial \mathcal{H}}{\partial \mathbf{p}}^\top \end{bmatrix}, \tag{25}$$

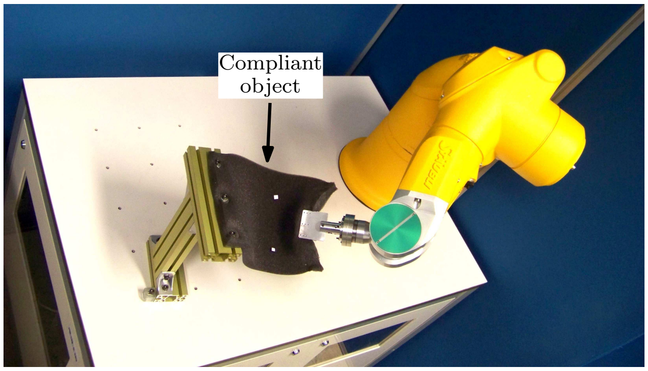


Fig. 3. The setup used in our experimental study.

with an energy-like function

$$\mathcal{H}(\Delta\mathbf{y}, \mathbf{p}) = \mathcal{Q}(\Delta\mathbf{y}) + \frac{1}{2}\mathbf{p} \cdot \mathbf{p} \in \mathbb{R}, \tag{26}$$

whose time-derivative satisfies

$$\dot{\mathcal{H}}(\Delta\mathbf{y}, \mathbf{p}) = -\mathbf{p}^\top \mathbf{B}\mathbf{p} \leq 0. \tag{27}$$

This expression shows that the scalar $\mathcal{H}$ monotonically decreases along trajectories of the system (25). To prove the asymptotic minimisation of the error $\Delta\mathbf{y}$, we analyse the equilibria of (25). These equations show that as time approaches infinity, the error $\Delta\mathbf{y} \to \mathbf{0}_{h\times 1}$ [19]. ■

IV. Experiments

A. Setup

Fig. 3 shows the 6-DOF ($n = 6$) robot manipulator that we use to conduct the experimental study. This mechanical system has an open architecture servo-controller that allows to set the joint velocity in real-time (see [15] for details). The manipulator's gripper rigidly grasps a $2 \times 12 \times 25$ mm soft foam sheet; we place artificial markers on the surface of this object. To acquire the visual feedback, we use two C310 Logitech cameras.

B. Case of study

We test the performance of our control method with two different 4-DOF deformation tasks (this situation corresponds to $\mathbf{y} = [y_1, y_2, y_3, y_4] \in \mathbb{R}^4$). In the first deformation task we only consider a single view ($m = 1$) of the object; the experiment consists in simultaneously position two deformable points $\mathbf{s}_1$ and $\mathbf{s}_2$ (see Fig. 4). We construct the deformation feature vector with the pixel coordinates

$$\mathbf{y} = \begin{bmatrix} \mathbf{s}_1^\top & \mathbf{s}_2^\top \end{bmatrix}^\top. \tag{28}$$

In the second task we use two cameras ($m = 2$) to observe the upper and side views of the object; the experiment consists in simultaneously position one deformable point and two angle features (see Fig. 5). We construct the deformation vector with four visual feedback points $\mathbf{s}_i$, and define its pixel-angle coordinates by

$$\mathbf{y} = \begin{bmatrix} \mathbf{s}_1^\top & \arccos(\mathbf{u}_1 \cdot \mathbf{l}_1) & \arccos(\mathbf{u}_2 \cdot \mathbf{l}_2) \end{bmatrix}^\top, \tag{29}$$

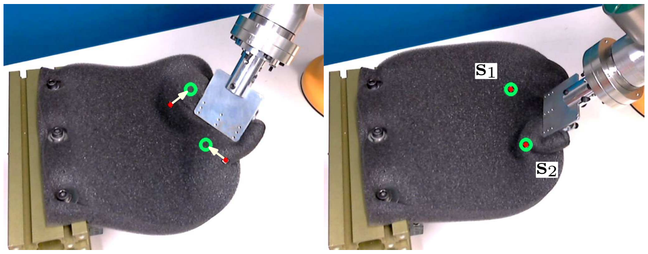
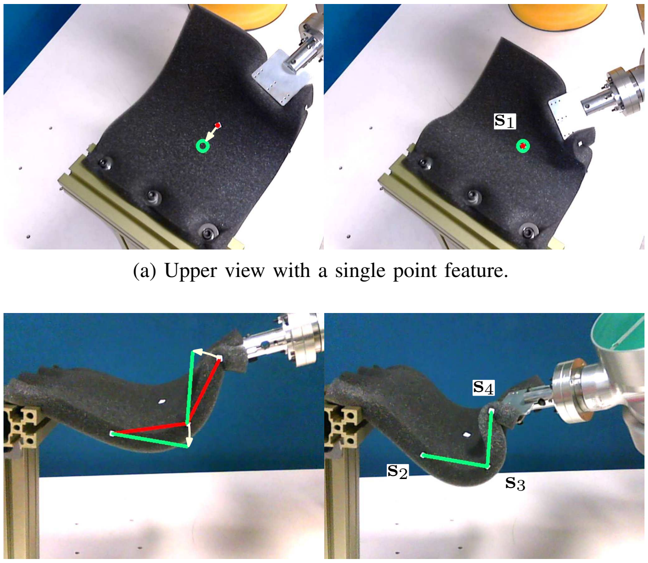


Fig. 4. Snapshots of the initial and final configurations of the single-view ($m = 1$) point positioning deformation experiment.



(a) Upper view with a single point feature.

(b) Side view with two angle features.

Fig. 5. Snapshots of the initial and final configurations of the two-view ($m = 2$) point-angle deformation experiment.

where the constant unit vectors $\mathbf{u}_i \in \mathbb{R}^2$ denote arbitrary references. We compute the feedback-dependent directional vectors $\mathbf{l}_i \in \mathbb{R}^2$ as

$$\mathbf{l}_i = \frac{\mathbf{s}_{i+1} - \mathbf{s}_{i+2}}{\|\mathbf{s}_{i+1} - \mathbf{s}_{i+2}\|}; \quad \text{for } i = 1, 2. \tag{30}$$

To minimise the deformation error $\Delta\mathbf{y}$, we implement the potential control action

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{y}}^\top (\Delta\mathbf{y}) = \lambda \mathbf{sat}(\Delta\mathbf{y}), \tag{31}$$

where $\lambda = 0.5$ represents a proportional feedback gain and $\mathbf{sat}(\cdot) : \mathbb{R}^4 \mapsto \mathbb{R}^4$ denotes a saturation function that satisfies $-\boldsymbol{\alpha} \leq \mathbf{sat}(\cdot) \leq \boldsymbol{\alpha}$, for $\boldsymbol{\alpha} \in \mathbb{R}^4$ as the bound vector. We set the components of $\boldsymbol{\alpha}$ to $\alpha_i = 20$ px and $\alpha_j = 0.2$ rad for the point and angle features, respectively. To compute the dynamic variable $\mathbf{p}$, we use initialise it with the vector $\mathbf{p}(0) = \mathbf{0}_{h\times 1}$, and use a matrix $\mathbf{B} = 5\mathbf{I}_{4\times 4}$. Note that these values for $\mathcal{Q}(\Delta\mathbf{y})$ and $\mathbf{B}$ provide a highly dissipative (hence smooth) behaviour to the system.

To implement the Lyapunov-based estimator, we initialise the vector of parameters with $\widehat{\boldsymbol{\theta}}(0) = [1, \ldots, 1]^\top$, and use a tuning matrix $\boldsymbol{\Gamma} = 30000\mathbf{I}_{g\times g}$. For our two cases of study,

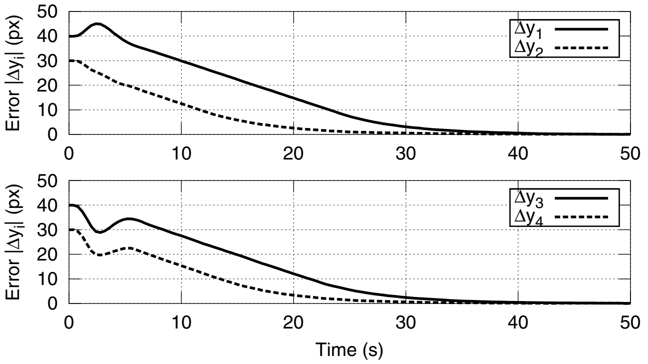


Fig. 6. Magnitude of the point deformation errors Δy_i of the single-view task shown in Fig. 4.

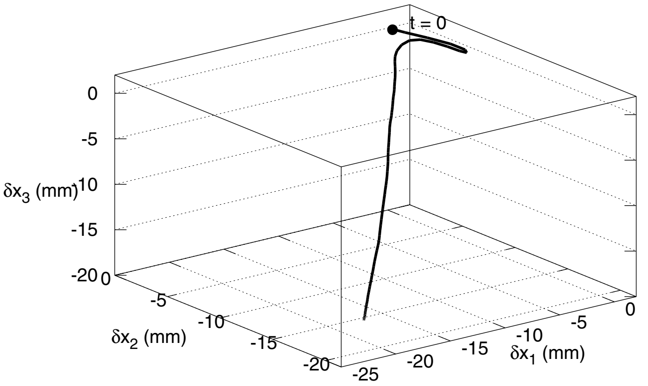


Fig. 7. Trajectory of the relative Cartesian displacements of the single-view task shown in Fig. 4.

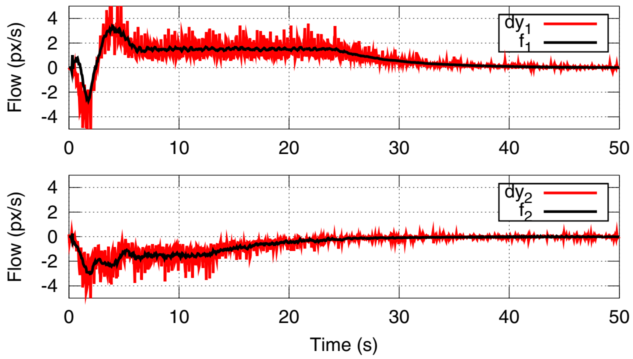


Fig. 8. Comparison of the measured and estimated deformation flows for the experiment shown in Fig. 4. First and second coordinates.

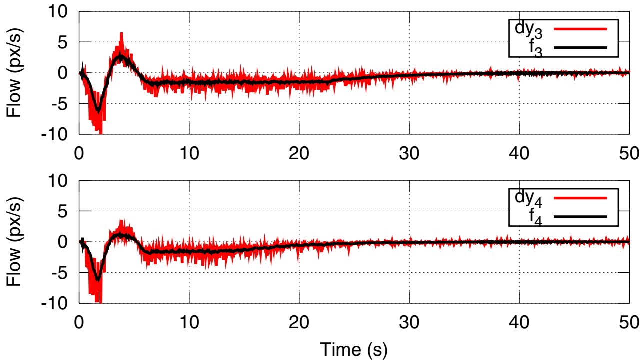


Fig. 9. Comparison of the measured and estimated deformation flows for the experiment shown in Fig. 4. Third and fourth coordinates.

we use the flow estimation gain matrices $\mathbf{K} = \mathbf{I}_{4\times4}$ and $\mathbf{K} = \mathrm{diag}(1, 1, 100, 100)$, respectively.

C. Results

Fig. 4 and Fig. 5 show snapshots of the conducted experiments. The overlaid red and green "curves" represent the feedback deformation features and the desired constant targets, respectively.

We first present the results of the single-view point positioning experiment. Fig. 6 depicts the asymptotic minimisation of the four deformation errors Δy_i. Fig. 7 shows the resulting Cartesian displacements of the gripper. To demonstrate the accuracy of our uncalibrated method, in Fig. 8 and Fig. 9 we present graphical comparisons of the visually measured deformation flow $\dot{\mathbf{y}}$ and the flow estimated by $\mathbf{f} = \widehat{\mathbf{J}}(\mathbf{s})\dot{\mathbf{x}}$.

Now, we present the results of the two-view point-angle deformation experiment. Fig. 10 depicts the asymptotic minimisation of the pixel and radian deformation errors. Finally, Fig. 11 shows the relative displacements of the gripper. To the best of our knowledge, this paper reports for the *first time* the simultaneous control of 4 independent deformation DOF by a *single* manipulator. In [15], we also report the control of 4-DOF, but jointly performed by two manipulators. The control of deformation via rotations results instrumental to our method. Approaches like [8] only consider plane motion, thus can only control up-to 2-DOF per robot. We must emphasise that the grasping configuration plays a crucial role in our formulation. Also note that the manipulator can only drive the features to physically reasonable targets.

D. Multimedia material

In the accompanying video, we illustrate the performance of our controller with different deformation control experiments. This video presents a simple procedure to obtain a good enough initial value for $\widehat{\boldsymbol{\theta}}$.

V. Conclusions

In this paper, we presented a visual servoing method to servo-control deformations of unknown elastic objects with the 6-DOF motion of the manipulator. To cope with the uncertain vision/deformation models, we first derived an algorithm that online estimates the Jacobian matrix. Next, we presented a dynamic-state feedback velocity control law which provides smooth trajectories to the manipulator. Finally, we reported experimental results to validate the controller.

In this new method, we incorporate the orientation of the fully-constrained grippers into the estimation algorithm. This feature allows us to: (i) increase the number of controllable deformation DOF, and (ii) extend the dimension of the Jacobian's kernel. In our early study [14], we noted that $n = 3$ has a restricted number of reachable configurations

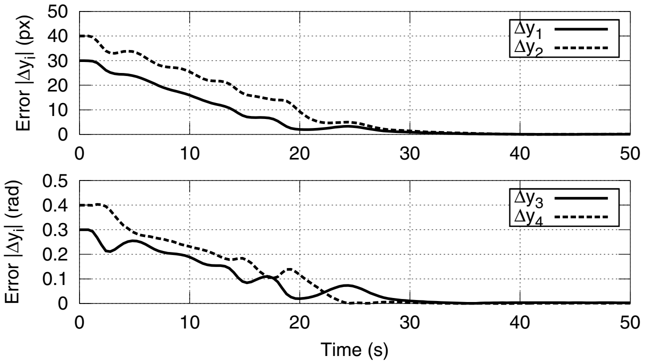


Fig. 10. Magnitude of the point and angle deformation errors Δy_i of the two-view task shown in Fig. 5.

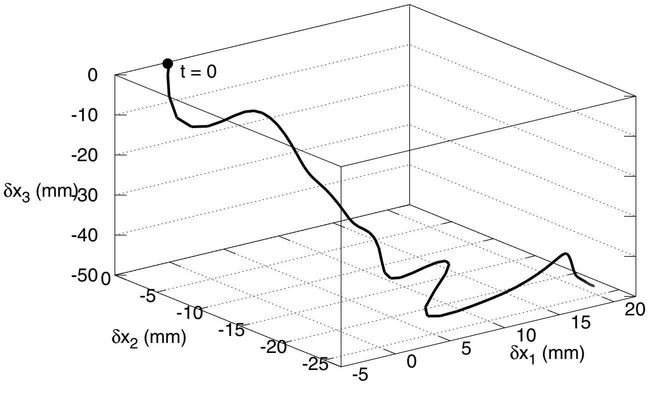


Fig. 11. Trajectory of the relative Cartesian displacements of the two-view task shown in Fig. 5.

(singularities tend to arise in this situation). In contrast with our previous estimator [15], the stability (convergence) of our new online algorithm can be proved with Lyapunov-like theory; we must remark that this method requires slow motion of the manipulator and proper filtering of the visually measured flow signals.

As future work, we want to use the stereo vision system to estimate the 3D deformation of the object. Additionally, we would like to extend this uncalibrated controller to perform shared (robot-human) deformation tasks.

References

[1] V. Mallapragada, N. Sarkar, and T. Podder, "Toward a robot-assisted breast intervention system," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1011–1020, Dec. 2011.

[2] M. Cusumano-Towner, A. Singh, S. Miller, J. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011, pp. 3893–3900.

[3] Z. Wang and S. Hirai, "Modeling and estimation of rheological properties of food products for manufacturing simulations," *J. Food Eng.*, vol. 102, no. 2, pp. 136 – 144, Jan. 2011.

[4] M. Saha and P. Isto, "Manipulation planning for deformable linear objects," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1141–1150, Dec. 2007.

[5] D. Henrich and H. Wörn, Eds., *Robot manipulation of deformable objects*, ser. Advanced manufacturing. New York: Springer-Verlag, 2000.

[6] S. Tokumoto and S. Hirai, "Deformation control of rheological food dough using a forming process model," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, 2002, pp. 1457–1464.

[7] M. Higashimori, K. Yoshimoto, and M. Kaneko, "Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2010, pp. 5120–5126.

[8] S. Hirai and T. Wada, "Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model," *Robotica*, vol. 18, no. 1, pp. 3–11, Jan. 2000.

[9] M. Torabi, K. Hauser, R. Alterovitz, V. Duindam, and K. Goldberg, "Guiding medical needles using single-point tissue manipulation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009, pp. 2705–2710.

[10] S. Kinio and A. Patriciu, "A comparative study of Hinf and PID control for indirect deformable object manipulation," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, 2012, pp. 414–420.

[11] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2013, pp. 1–8.

[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] J. Piepmeier, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 143–147, Feb. 2004.

[14] D. Navarro-Alarcon and Y.-H. Liu, "Uncalibrated vision-based deformation control of compliant objects with online estimation of the Jacobian matrix," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2013, pp. 4977–4982.

[15] D. Navarro-Alarcon, Y.-H. Liu, J. 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.

[16] 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.

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

[18] Y.-H. Liu and D. Sun, "Stabilizing a flexible beam handled by two manipulators via pd feedback," *IEEE Trans. Autom. Control*, vol. 45, no. 11, pp. 2159–2164, Nov. 2000.

[19] J.-J. Slotine and W. Li, *Applied Nonlinear Control*, 1st ed. New Jersey: Prentice Hall, 1991.

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

[21] R. Ortega, A. van der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *IEEE Control Syst. Mag.*, vol. 21, no. 2, pp. 18–33, Apr. 2001.