Automatic 3-D Manipulation of Soft Objects by Robotic Arms With an Adaptive Deformation Model

David Navarro-Alarcon, *Member, IEEE*, Hiu Man Yip, *Student Member, IEEE*, Zerui Wang, *Student Member, IEEE*, Yun-Hui Liu, *Fellow, IEEE*, Fangxun Zhong, *Student Member, IEEE*, Tianxue Zhang, *Student Member, IEEE*, and Peng Li

***Abstract*—In this paper, we present a new feedback method to automatically servo-control the 3-D shape of soft objects with robotic manipulators. The soft object manipulation problem has recently received a great deal of attention from robotics researchers because of its potential applications in, e.g., food industry, home robots, medical robotics, and manufacturing. A major complication to automatically control the shape of an object is the estimation of its deformation properties, which determines how the manipulator's motion actively transforms into deformations. Note that these properties are rarely known beforehand, and its offline parametric identification is difficult and/or impractical to conduct in many applications. To cope with this issue, we developed a new algorithm that computes in real time the unknown deformation parameters of a soft object; this algorithm provides a valuable adaptive behavior to the deformation controller, something we cannot achieve with traditional fixed-model approaches. In contrast with most controllers in the literature, our new method can explicitly servo-control 3-D deformations (and not just 2-D image projections) in an entirely model-free way. To validate the proposed adaptive controller, we present a detailed experimental study with robotic manipulators.**

***Index Terms*—Adaptive control, deformable models, robots manipulators, shape control, visual servoing.**

I. Introduction

ROBOTIC manipulation of soft objects is needed in several economically important applications, for example the active shaping of food materials [1], handling and folding fabrics [2], assembling flexible automotive parts [3], manipulating cables or sutures [4], palpating organs and tissues [5], to name a few cases. Despite the recent progress in soft robotics, the automatic manipulation of compliant objects remains an open research problem [6]. There are several issues that complicate the automation of these types of tasks. Perhaps the most crucial issue is the difficulty to estimate (in real time) the deformation properties of a soft object; note that these properties greatly vary even among objects of the same category.

The object's deformation properties are necessary (even though coarsely approximated) to coordinate the motion of the mechanical system with the feedback measurements of the object's shape. However, note that in many situations the parametric identification of the deformation model is difficult to perform beforehand; to fully identify this model requires the manipulator to perform several testing physical interactions with the object, which may be impractical in many applications, e.g., in laparoscopic surgery. Even for the case in which the deformation parameters are exactly known, the computation of an accurate enough soft object model typically poses additional problems for real-time applications. In this paper, we aim to develop adaptive methods that can explicitly servo-control (i.e., with a measurable deformation error) the 3-D shape of a compliant object but in a *model-free* manner.

A. Related Work

In recent years, great progress has been obtained in the robotic manipulation of deformable linear objects such as compliant beams, cables, or surgical sutures (see [7]–[11]). The multi-finger grasping problem of soft bodies has been addressed in [12]–[14] using various approaches. Some other works have tackled the soft object manipulation but from a path-planning perspective (see [15]–[18]). Significant effort has been devoted to the automatic manipulation of rheological (i.e., viscoelastic) materials; the dynamic modeling, parametric identification, and robotic manipulation of this type of objects are addressed in [19]–[22]. In this paper, however, we are mainly interested in the manipulation of soft objects that exhibit (or that can be fairly modeled with) elastic deformations only.

There are some works in the literature that address the servo-control of elastic deformations, which is the main topic in this work. For example, the robotic manipulation of an extensible piece of fabric is formulated in [23] as a classical feedback control problem; the objective of this method is to position multiple 2-D feedback points to desired configurations. This noncollocated positioning problem is similarly studied in [24]–[26]. Note that to explicitly control the displacements of the deformable feature points, most methods in the literature require some *a priori* knowledge or calibration of the object's

Manuscript received December 13, 2015; accepted February 10, 2016. Date of publication March 9, 2016; date of current version April 1, 2016. This paper was recommended for publication by Associate Editor V. Kyrki and Editor B. J. Nelson upon evaluation of the reviewers' comments. This work was supported in part by the Hong Kong Research Grants Council under Grant 415011 and Grant CUHK6/CRF/13G, and in part by the Hong Kong Innovation and Technology Fund under Grant ITS/112/15FP.

The authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: david@cuhk.edu.hk; hmyip@mae.cuhk.edu.hk; zrwang@mae.cuhk.edu.hk; yhliu@mae.cuhk.edu.hk; fxzhong@mae.cuhk.edu.hk; txzhang@mae.cuhk.edu.hk; pli@mae.cuhk.edu.hk).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material consists of a video, viewable with Windows Media Player and VLC Player, demonstrating the performance of the proposed adaptive controller. All experiments are conducted with no knowledge of the deformation model. The authors conduct these experiments with a 6-DOF industrial manipulator (Staubli) and a 3-DOF surgical robot.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2016.2533639

deformation properties. When this information is not available, the stability of model-dependent controllers cannot be rigorously guaranteed.

Despite the clear limitations of a model-based design, not many works have addressed the *simultaneous* online estimation and servo-control of elastic deformations. In [27], we developed a model-free controller that computes in real time the deformation Jacobian matrix of a compliant object using the Broyden update rule [28]. To compute the Jacobian matrix, this numeric method only requires measurements of the input motions and output deformations (for visual servoing applications of this method, see [29] and [30]). Recently, in [31], we proposed an energy-motivated method to servo-control deformations without any knowledge of the deformation and camera models; this method uses adaptive control techniques to estimate the unknown parameters. However, it must be remarked that both [27] and [31] formulate the problem in terms of image (pixel) coordinates; therefore, these methods cannot explicitly control the 3-D configuration of the object, but only its 2-D image projections.

B. Our Contribution

To contribute to this economically important problem, in this paper we present a new adaptive method to automatically manipulate an unknown elastic object. In contrast with most controllers in the literature, our method can explicitly servo-control the 3-D shape without the need to identify the object's deformation model. For that, we developed an algorithm that estimates in real time the vector of unknown deformation parameters; this algorithm provides a valuable adaptive behavior to the controller, something we cannot achieve with fixed-model approaches. To automatically manipulate the object, we propose a deformation feature vector that decomposes the feedback manipulation task into positioning and shape terms; this allows us to simultaneously control the object's final position and its relative deformations. We prove the stability of our method with Lyapunov theory and provide simple algorithms to facilitate its implementation. Experimental results are presented to validate the proposed control method.[1]

The rest of the paper is organized as follows. In Section II, we derive the mathematical models. Section III presents the proposed control method. In Section IV, we define the deformation vector. Section V presents the conducted experiments. In Section VI, we give the conclusion and future work.

II. Mathematical Modeling

A. Robot Kinematics

Consider M kinematically controlled serial robotic manipulators. We denote the vector of joint displacements and end-effector Cartesian position of the jth robot by $\mathbf{q}_j \in \mathbb{R}^{G_j}$ and $\mathbf{x}_j = \mathbf{x}_j(\mathbf{q}_j) : \mathbb{R}^{G_j} \mapsto \mathbb{R}^3$, respectively, for $G_j \geq 3$. The differential kinematic equation of the robot is

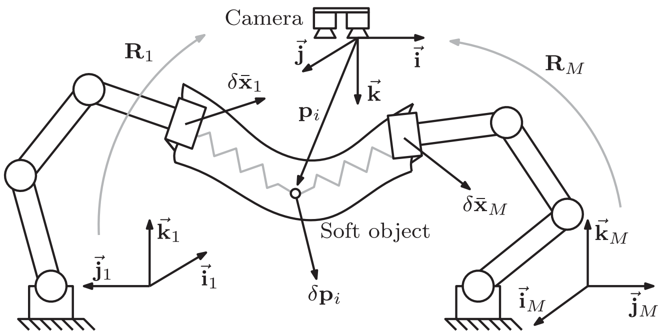


Fig. 1. Conceptual representation of the soft object manipulation setup.

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

where $\partial \mathbf{x}_j / \partial \mathbf{q}_j \in \mathbb{R}^{3 \times G_j}$ denotes the robot's Jacobian matrix.

Assumption 1 (Robot's structure and motion): In this paper, we make the following modeling assumptions:

1) The kinematic model of each robot is exactly known and available for control purposes.
2) The robots are kinematically controlled, meaning that the (input) velocity vector $\dot{\mathbf{q}}_j(t)$ can be *exactly* set with the servo-controller at any time instant t.
3) The velocity servo-controller renders infinitely stiff behavior to the robots; interactions with nonrigid objects do not affect the robots' trajectories.
4) The robots' motion is smooth, and its joint and end-effector velocities are bounded, i.e., $\|\dot{\mathbf{q}}_j\| \leq c_k$ and $\|\dot{\mathbf{x}}_j\| \leq c_l$, for positive bounding scalars $c_i > 0$.

Remark 1: The control methods proposed in this work are designed for typical kinematically controlled rigid manipulators (see, e.g., [33]); we make the above assumptions to model the properties that exhibit these systems. Note that by considering velocity control inputs, the method that we propose can be easily implemented in most commercial robot manipulators, whereas methods designed for torque-controlled systems require a special low-level torque/current control interface (see, e.g., [34]), which is generally not available in most systems.

B. Quasi-Static Deformation Model

Consider that the robots physically interact (in a point-contact manner) with a compliant object whose exact deformation model is unknown. The configuration of the object is measured by a fixed stereo camera that observes K deformable feature points located over the surface (see Appendix A). Let the vector $\mathbf{p}_i \in \mathbb{R}^3$ denote the 3-D position of the ith feature point defined with respect to the camera's coordinate frame $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. The robots' position in terms of the camera's frame is

$$\bar{\mathbf{x}}_j = \mathbf{R}_j \mathbf{x}_j + \mathbf{t}_j \in \mathbb{R}^3 \tag{2}$$

for $\mathbf{R}_j \in \mathbb{R}^{3\times3}$ and $\mathbf{t}_j \in \mathbb{R}^3$ as a constant rotation matrix and constant translation vector, respectively. See Fig. 1 for a conceptual representation.

[1] A weaker version of the proposed online algorithm was presented in [32]. However, in contrast with our new multirobot method, this early image-based approach can only control 2-D projections of a 3-D shape.

In order to control deformations of soft objects, we must find an expression that (approximately) predicts how the active motion of the robots affects the measurements of the feature points. To this end, we restrict our attention to soft object manipulation tasks that satisfy the following conditions.

Assumption 2 (Active soft object manipulation): We locally consider the following for quasi-static (slow) manipulation:

a) the soft object exhibits (or can be fairly modeled with) elastic deformations only, i.e., it does not present any rheological deformations [21];
b) the dominant effect is the object's potential energy, i.e., the object's inertial effects are minimal and, therefore, can be neglected from the model;
c) the robots rigidly grasp the soft object (with no slippage or loose contact) during the whole manipulation task; the contact point with the object is approximated by $\bar{\mathbf{x}}_j$;
d) each of the robots' coordinates $\bar{\mathbf{x}}_j^{\mathsf{T}} = [x_{\mathrm{xj}},\ x_{\mathrm{yj}},\ x_{\mathrm{zj}}]$ is interconnected with each of the point's coordinates $\mathbf{p}_i^{\mathsf{T}} = [p_{\mathrm{xi}},\ p_{\mathrm{yi}},\ p_{\mathrm{zi}}]$; motion can be continuously transmitted with a spring, as conceptually depicted in Fig. 1;
e) each deformable point $\mathbf{p}_i$ has a stable (i.e., positive and full-rank) stiffness matrix around its equilibrium point.

Remark 2: A model that considers only potential energy terms, as in Assumptions 2a and 2b, can be used in many practical applications that require to manipulate "mostly" elastic objects at "slow" speed. Some of these applications are, e.g., the positioning of fabrics [23], active bending of beams [7], [31], assembly of flexible parts [3], stretching of ligaments/tissues [35], to name a few cases.

To derive the quasi-static object model, let us first denote by $V \in \mathbb{R}$ the elastic energy of the spring connecting the robots with the feature point. We compute the equilibrium equations

$$\frac{\partial V}{\partial \mathbf{p}_i}^{\mathsf{T}} = \mathbf{0}_{3\times 1} \tag{3}$$

$$\sum_{j=1}^{M} \left(\frac{\partial V}{\partial \bar{\mathbf{x}}_j}^{\mathsf{T}} - \mathbf{f}_j \right) = \mathbf{0}_{3\times 1} \tag{4}$$

for $\mathbf{f}_j \in \mathbb{R}^3$ as the external force applied by the jth robot. To solve the above equations (and find the displacement relation between the grippers and feature point) requires exact knowledge of the deformation model and parameters of the object, which may not be available in many applications. To cope with this issue, we derive a linear model of the object in neighborhood around the static equilibrium [23]. To this end, we introduce $\delta\mathbf{p}_i = \mathbf{p}_i - \mathbf{p}_i^* \in \mathbb{R}^3$ and $\delta\bar{\mathbf{x}}_j = \bar{\mathbf{x}}_j - \bar{\mathbf{x}}_j^* \in \mathbb{R}^3$ to model the relative displacements of the feedback points and grasping points from the (undeformed) equilibrium, here denoted by $\mathbf{p}_i^* \in \mathbb{R}^3$ and $\bar{\mathbf{x}}_j^* \in \mathbb{R}^3$, respectively. By linearizing (3) around the equilibrium, we obtain

$$\underbrace{\frac{\partial^2 V}{\partial \mathbf{p}_i \partial \mathbf{p}_i}}_{\mathbf{A}_i} \delta\mathbf{p}_i + \sum_{j=1}^{M} \underbrace{\frac{\partial^2 V}{\partial \bar{\mathbf{x}}_j \partial \mathbf{p}_i}}_{\mathbf{B}_{i,j}} \delta\bar{\mathbf{x}}_j = \mathbf{0}_{3\times 1} \tag{5}$$

where $\mathbf{A}_i \in \mathbb{R}^{3\times 3}$ and $\mathbf{B}_{i,j} \in \mathbb{R}^{3\times 3}$ represent unknown constant stiffness matrices, whose partial derivatives are evaluated at the equilibrium. Solving (5) for $\delta\mathbf{p}_i$ yields

$$\delta\mathbf{p}_i = \mathbf{D}_i \delta\bar{\mathbf{x}} \tag{6}$$

for an extended vector $\delta\bar{\mathbf{x}} = [\delta\bar{\mathbf{x}}_1^{\mathsf{T}},\ \ldots,\ \delta\bar{\mathbf{x}}_M^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{3M}$ and a deformation matrix $\mathbf{D}_i \in \mathbb{R}^{3\times 3M}$ defined as

$$\mathbf{D}_i = -\mathbf{A}_i^{-1} \begin{bmatrix} \mathbf{B}_{i,1} & \cdots & \mathbf{B}_{i,M} \end{bmatrix}. \tag{7}$$

Note that $\mathbf{A}_i$ represents the stiffness matrix with respect to the displacements of $\mathbf{p}_i$, which from Assumption 2e we known is invertible.[2] We emphasize that our aim in this paper is not to develop a highly accurate model of the infinite-dimensional elastic body. Instead, we focus on developing a simple—yet sufficient for our control purposes—quasi-static displacements model, which describes how the slow motion of the robots transforms into deformations and presents linearly with respect to the unknown deformation parameters (this latter property is useful to design adaptive algorithms). To simplify notation, we introduce the extended vectors $\mathbf{p} \in \mathbb{R}^{3K}$ and $\mathbf{x} \in \mathbb{R}^{3M}$, which are defined as

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1^{\mathsf{T}} & \cdots & \mathbf{p}_K^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1^{\mathsf{T}} & \cdots & \mathbf{x}_M^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{8}$$

Remark 3: The local nature of (6) does not impose severe constraints to our slow-motion adaptive method since it continuously updates the model's parameters with real-time vision and position measurements, therefore extending its range of validity. We show in Section V how this simple model can be used to manipulate objects of various stiffness/shapes.

C. Vector of Deformation Parameters

Note that the proposed deformation model (6) can be linearly parameterized in terms of unknown terms, which contrast with other models in the literature, e.g., the rigidity functions proposed in [36] (we show in later sections that this linearity property results instrumental for the design of the adaptive controller). Let us define the coordinate-transformed deformation matrix $\mathbf{F}_i \in \mathbb{R}^{3\times 3M}$ as

$$\mathbf{F}_i = \mathbf{D}_i \operatorname{diag}(\mathbf{R}_1,\ \mathbf{R}_2,\ \ldots,\ \mathbf{R}_M) \tag{9}$$

which also includes the rotation matrices.[3] We group the elements of the K deformation matrices $\mathbf{F}_i$ into the vector of unknown constant parameters

$$\begin{aligned} \mathbf{a} = \big[\, & \mathbf{u}_1^{\mathsf{T}}\mathbf{F}_1 \quad \mathbf{u}_2^{\mathsf{T}}\mathbf{F}_1 \quad \mathbf{u}_3^{\mathsf{T}}\mathbf{F}_1 \quad \mathbf{u}_1^{\mathsf{T}}\mathbf{F}_2 \quad \mathbf{u}_2^{\mathsf{T}}\mathbf{F}_2 \quad \mathbf{u}_3^{\mathsf{T}}\mathbf{F}_2 \\ & \cdots \quad \mathbf{u}_1^{\mathsf{T}}\mathbf{F}_K \quad \mathbf{u}_2^{\mathsf{T}}\mathbf{F}_K \quad \mathbf{u}_3^{\mathsf{T}}\mathbf{F}_K \,\big]^{\mathsf{T}} \in \mathbb{R}^L \end{aligned} \tag{10}$$

where $\mathbf{u}_1^{\mathsf{T}} = [1,\ 0,\ 0]$, $\mathbf{u}_2^{\mathsf{T}} = [0,\ 1,\ 0]$, $\mathbf{u}_3^{\mathsf{T}} = [0,\ 0,\ 1] \in \mathbb{R}^3$, and for $L = 9MK$.

D. Problem Formulation

To avoid the open-loop deformation response of traditional robot-centered controllers (see, e.g., [37] and [38]), in our

[2] Having a stable full-rank matrix $\mathbf{A}_i$ physically means that the point $\mathbf{p}_i$ is elastically constrained (in all directions) by stiffness that brings the point back to equilibrium, a condition that is reasonable to assume.

[3] The translation vectors $\mathbf{t}_j$ are not needed to transform velocities between the coordinate frames; thus, we exclude them from the vector of parameters.

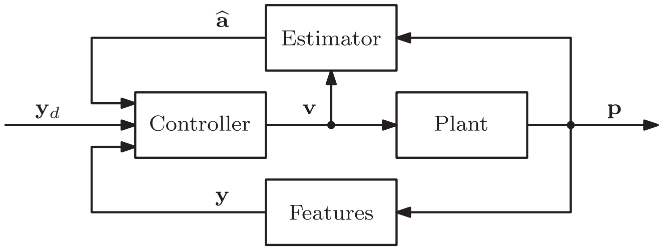


Fig. 2. Schematic representation of the proposed controller. In this figure, the plant system represents the soft object manipulated by the robots.

method, we use the measured feature points $\mathbf{p}_i$ to construct a vector of deformation features, which locally quantifies the object's deformations. We denote this feedback deformation vector by $\mathbf{y} = \mathbf{y}(\mathbf{p}) : \mathbb{R}^{3K} \mapsto \mathbb{R}^N$, for $N \leq 3M$ (in later sections, we give explicit definitions and examples of these deformation features). The time derivative of this vectorial function yields

$$\dot{\mathbf{y}} = \mathbf{J}\dot{\mathbf{x}} \tag{11}$$

where the matrix $\mathbf{J} \in \mathbb{R}^{N \times 3M}$, here called the deformation Jacobian matrix, satisfies

$$\mathbf{J} = \frac{\partial \mathbf{y}}{\partial \mathbf{p}}(\mathbf{p})\frac{\partial \mathbf{p}}{\partial \mathbf{x}}(\mathbf{x}) = \frac{\partial \mathbf{y}}{\partial \mathbf{p}}(\mathbf{p})\left[\mathbf{F}_1^{\mathsf{T}} \quad \cdots \quad \mathbf{F}_K^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{12}$$

To guarantee the full row-rank of $\mathbf{J}$, we must 1) define the coordinates of $\mathbf{y}$ such that they provide a linearly independent characterization of the object's shape and 2) define points $\mathbf{p}_i$ such that its deformations transform independently from the other points. Since $\mathbf{y}$ is a user-defined function, we can always construct functions that satisfy condition 1; a simple way to satisfy condition 2 is to select points that are not too close to each other (i.e., with sufficient separation). Although the coordinates of $\mathbf{y}$ can be computed from sensor feedback, the matrix $\mathbf{J}$ cannot be analytically computed (specifically $\partial \mathbf{p}/\partial \mathbf{x}$) since the vector $\mathbf{a}$ is unknown.

Problem: Given a desired (constant and physically reachable) vector $\mathbf{y}_d \in \mathbb{R}^N$, design a velocity controller $\dot{\mathbf{x}}$ that asymptotically minimizes the error $\Delta\mathbf{y} = \mathbf{y} - \mathbf{y}_d \in \mathbb{R}^N$ without any knowledge of the vector of parameters $\mathbf{a}$.

III. Controller Design

A. Online Parameter Estimator

With our proposed control method (whose architecture is given in Fig. 2), we compute the adaptive vector of parameters $\widehat{\mathbf{a}} \in \mathbb{R}^L$ to iteratively approximate the deformation model. The main idea behind the algorithm is to vary $\widehat{\mathbf{a}}$ in the direction that minimizes a measurable model matching error. For that, we first parameterize the velocity of the feature points as

$$\dot{\mathbf{p}} = \left[\mathbf{F}_1^{\mathsf{T}} \quad \cdots \quad \mathbf{F}_K^{\mathsf{T}}\right]^{\mathsf{T}}\dot{\mathbf{x}} = \mathbf{W}\mathbf{a} \tag{13}$$

for $\mathbf{W}(\dot{\mathbf{x}}) \in \mathbb{R}^{3K \times L}$ as a known regression matrix. Next, we compute the model matching error $\mathbf{e} \in \mathbb{R}^{3K}$ as

$$\mathbf{e} = \mathbf{W}\widehat{\mathbf{a}} - \dot{\mathbf{p}} = \mathbf{W}\Delta\mathbf{a} \tag{14}$$

where the vector $\Delta\mathbf{a} = \widehat{\mathbf{a}} - \mathbf{a} \in \mathbb{R}^L$ represents the parameter estimation error. Note that $\mathbf{e}$ can be directly computed from sensor measurements and the estimated parameters and does not require explicit knowledge of $\mathbf{a}$.

The objective of this algorithm is to compute a vector $\widehat{\mathbf{a}}$ that asymptotically minimizes the quadratic function

$$Q = \mathbf{e}^{\mathsf{T}}\mathbf{e}/2. \tag{15}$$

We update the parameters $\widehat{\mathbf{a}}$ with the rule

$$\frac{d}{dt}\widehat{\mathbf{a}} = -\gamma\frac{\partial Q}{\partial \widehat{\mathbf{a}}}^{\mathsf{T}} = -\gamma\mathbf{W}^{\mathsf{T}}\mathbf{e} \tag{16}$$

where the scalar $\gamma > 0$ modulates the adaptation rate.

Proposition 1: The gradient descent estimator (16) provides a numerically stable computation of the parameters $\widehat{\mathbf{a}}$, which asymptotically minimizes the error $\mathbf{e}$.

Proof: Consider the Lyapunov-like quadratic function

$$H = \Delta\mathbf{a}^{\mathsf{T}}\Delta\mathbf{a}/2 \tag{17}$$

whose time derivative yields

$$\dot{H} = -\Delta\mathbf{a}^{\mathsf{T}}\gamma\mathbf{W}^{\mathsf{T}}\mathbf{e} = -\mathbf{e}^{\mathsf{T}}\gamma\mathbf{e} \leq 0. \tag{18}$$

This proves that the parameter's error $\Delta\mathbf{a}$ is nonincreasing [39]; thus, the numerical state vector $\widehat{\mathbf{a}}$ is bounded. To prove the convergence of the error vector $\mathbf{e} \to \mathbf{0}_{L\times 1}$, note that since we only consider slow (i.e., saturated) and smooth (i.e., differential) motion of the robot manipulators and object, therefore the scalar function $\ddot{H} = -2\mathbf{e}^{\mathsf{T}}\gamma\dot{\mathbf{e}}$ is also bounded. This means that (18) is uniformly continuous; asymptotic minimization of $\mathbf{e}$ directly follows using the Barbalat's lemma [40]. ■

Remark 4: The purpose of the gradient descent estimator is to compute a vector of parameters $\widehat{\mathbf{a}}$, which can be used to reproduce the output velocity $\dot{\mathbf{p}}$. Note, however, that this property does not necessarily means the identification of the true parameters. In our method, we use the estimated parameters to approximate the deformation Jacobian matrix. This matrix is needed for computing the velocity control input with classical kinematic controllers (see, e.g., [33], [41], and [42]).

Remark 5: The proposed online estimator updates the parameters based on continuous velocity measurements of *both* the deformable points and the robotic grippers. Therefore, this algorithm cannot be used to estimate deformations of rheological objects (e.g., as done in [20]) because the shape of these types of materials keeps changing even after the robots' motion have stopped; these time-varying deformations cannot be detected with our method.

B. Saturated Velocity Controller

To actively deform the soft object, in this work, we use a kinematic motion control design. Since the deformation Jacobian matrix (12) is unknown, we compute an estimation $\widehat{\mathbf{J}} \in \mathbb{R}^{N \times 3M}$ as

$$\widehat{\mathbf{J}} = \frac{\partial \mathbf{y}}{\partial \mathbf{p}}(\mathbf{p})\left[\widehat{\mathbf{F}}_1^{\mathsf{T}} \quad \widehat{\mathbf{F}}_2^{\mathsf{T}} \quad \cdots \quad \widehat{\mathbf{F}}_K^{\mathsf{T}}\right]^{\mathsf{T}} \tag{19}$$

where the matrices $\widehat{\mathbf{F}}_i \in \mathbb{R}^{3\times 3M}$ are constructed with the variable parameters $\widehat{\mathbf{a}}$ according to the definition (10), as in, e.g.,

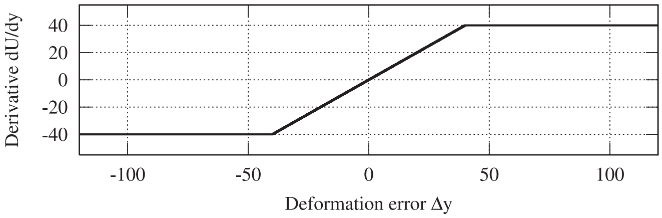


Fig. 3. Saturated potential action $\partial U_l / \partial y_l$, computed with $s_l = 40$.

[43] and [44]. The velocity control input is as

$$\dot{\mathbf{x}} = -\widehat{\mathbf{J}}^{+} \mathbf{K} \frac{\partial U}{\partial \mathbf{y}}^{\mathsf{T}} \tag{20}$$

where the matrix $\widehat{\mathbf{J}}^{+} \in \mathbb{R}^{3M \times N}$ represents the Moore–Penrose pseudoinverse, $\mathbf{K} \in \mathbb{R}^{N \times N}$ denotes a positive diagonal gain matrix, and $U(\Delta \mathbf{y}) = \sum U_l(\Delta y_l) : \mathbb{R}^N \mapsto \mathbb{R}$, for $l = 1, \ldots, N$, represents a $\mathcal{C}^1$ potential energy function with positive-definite terms U_l computed as [45]

$$U_l = \begin{cases} \frac{1}{2} |\Delta y_l|^2, & \text{for} \quad |\Delta y_l| < s_l \\ \frac{1}{2} s_l |\Delta y_l|, & \text{for} \quad |\Delta y_l| \geq s_l \end{cases} \tag{21}$$

for $s_l \in \mathbb{R}$ as positive saturation scalars. The continuous partial derivatives of U_l satisfy (see Fig. 3)

$$\frac{\partial U_l}{\partial y_l} = \begin{cases} \Delta y_l, & \text{for} \quad |\Delta y_l| < s_l \\ s_l \operatorname{sgn}(\Delta y_l), & \text{for} \quad |\Delta y_l| \geq s_l. \end{cases} \tag{22}$$

These scalars s_l determine the threshold at which the robots attain the maximum speed during manipulation; the value of s_l depends on the units of the deformation error Δy_l it monitors.

The specific design of the velocity-driven online estimator allows it to satisfy the differential relation

$$\dot{\mathbf{y}} = \widehat{\mathbf{J}} \dot{\mathbf{x}} \tag{23}$$

only after a few iterations of continuous slow motion. Note that these motions can be conducted locally around the starting configuration, thus avoiding large deformations to the object; we remark that this procedure must be performed only once in order to initialize the elements of the matrix $\widehat{\mathbf{J}}$.

To analyze the stability of the control law, we assume that the estimation algorithm has been initialized such that (23) is exactly satisfied. Substitution of the control input (20) into (23) enforces the closed-loop dynamical system

$$\dot{\mathbf{y}} = -\mathbf{K} \frac{\partial U}{\partial \mathbf{y}}^{\mathsf{T}}. \tag{24}$$

Consider U as a Lyapunov candidate function for (24), in which computing its time derivative we obtain $\dot{U} \leq 0$, which proves the stability of the system. Asymptotic convergence of $\Delta \mathbf{y}$ can be proved with the LaSalle principle [40]. Appendix B details the implementation of the controller.

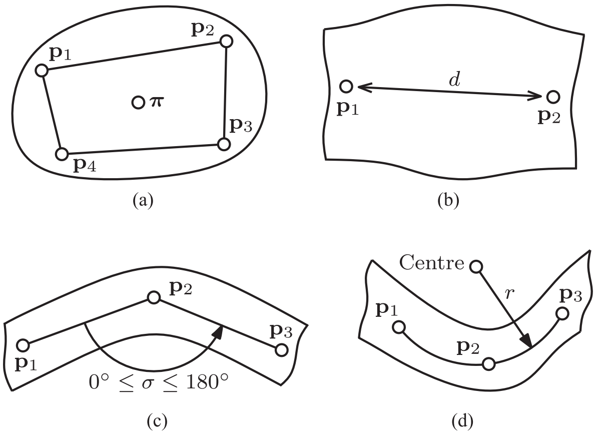


Fig. 4. Conceptual representations of the following proposed deformation features: (a) geometric center computed with $C = 4$, (b) compression distance between two points, (c) bending angle between two 3-D vectors, and (d) circumference that passes thought three deformable points.

IV. Feedback Deformation Vector

A. Deformation Problem

Traditional robotic manipulation tasks typically involve controlling the position vector of a rigid object, defined, e.g., at its *constant* center of mass. However, this same definition of robotic manipulation cannot be used with a deformable object, as its shape is easily affected by physical interactions with the robots and environment. To cope with this issue, we construct the coordinates y_k of the deformation vector by combining the feedback of two terms, a feedback position term and a feedback shape term, conceptually grouped as $\mathbf{y} = [\text{position}, \text{shape}]^{\mathsf{T}}$. The first term represents a 3-D vector, which we use to servo-control the object's position. The second term quantifies the internal displacements of the object; thus, it can be used to actively control relative deformations.

B. Position Term

1) Explicit Feature Point: The simplest way to control the 3-D position of the soft object is by actively driving a feedback point of interest into a desired location (see Fig. 1). In this way, we define three deformation coordinates as

$$\begin{bmatrix} y_k & y_{k+1} & y_{k+2} \end{bmatrix}^{\mathsf{T}} = \mathbf{p}_i. \tag{25}$$

2) Centroid: Alternatively, we can define the object's position in terms of the geometric center (i.e., the mean average position) of multiple deformable points [see Fig. 4(a)]. We compute the object's centroid $\boldsymbol{\pi} \in \mathbb{R}^3$ as

$$\boldsymbol{\pi} = (\mathbf{p}_1 + \mathbf{p}_2 + \cdots + \mathbf{p}_C) / C \tag{26}$$

and define three coordinates as $y_k = \pi_l$, for $l = 1, 2, 3$.

C. Shape Term

Besides controlling the object's position, we can also control the object's squeezing, bending, twisting, etcetera. We use the following *scalar* feedback features to achieve this.

1) Compression Distance: With this feature, the relative distance between two feature points of interest can be actively controlled. We use two feature points $\mathbf{p}_1$ and $\mathbf{p}_2$ to compute the object's compression distance $d \in \mathbb{R}$ as

$$d = \|\mathbf{p}_1 - \mathbf{p}_2\|. \tag{27}$$

For each distance, we define one deformation coordinate as $y_k = d$. See Fig. 4(b) for a conceptual representation.[4]

2) Folding Angle: This feature represents the measured angle between two intersecting lines computed with feature points located on the object's surface; this feature is useful to control the relative folding of an object. We use three feature points to compute the angle $\sigma \in \mathbb{R}$ as

$$\sigma = \arccos\left(\frac{(\mathbf{p}_1 - \mathbf{p}_2)\cdot(\mathbf{p}_3 - \mathbf{p}_2)}{\|\mathbf{p}_1 - \mathbf{p}_2\|\|\mathbf{p}_3 - \mathbf{p}_2\|}\right), \text{ for } 0^\circ \leq \sigma \leq 180^\circ. \tag{28}$$

For each measured angle, we define one deformation coordinate as $y_k = \sigma$. See Fig. 4(c) for a conceptual representation.

The point $\mathbf{p}_2$ represents the intersection between both lines; therefore, it should be selected around the object's bending point. To properly characterize the bending angle, the points $\mathbf{p}_1$ and $\mathbf{p}_3$ should be selected at the extreme of the deformable area of interest. Configurations around $\sigma \approx 0^\circ, \sigma \approx 180^\circ$ should be avoided to prevent singularity problems.

3) Normalized Curvature: The purpose of this feature is to quantify the "roundness" of the object's bending. The radius $r \in \mathbb{R}$ of the 3-D circumference that passes through the points $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ satisfies [46]

$$r = \frac{\|\mathbf{p}_1 - \mathbf{p}_2\|\|\mathbf{p}_2 - \mathbf{p}_3\|\|\mathbf{p}_3 - \mathbf{p}_1\|}{2\|(\mathbf{p}_1 - \mathbf{p}_2)\times(\mathbf{p}_2 - \mathbf{p}_3)\|}. \tag{29}$$

We can compute the normalized 3-D curvature $\kappa \in \mathbb{R}$ as

$$\kappa = \|\mathbf{p}_1 - \mathbf{p}_3\|/(2r), \quad \text{for} \quad 0 < \kappa \leq 1. \tag{30}$$

For this unitless scalar, a value close to zero represents flatness (as with the standard curvature metric), whereas a value close to one represents a circle with diameter $\|\mathbf{p}_1 - \mathbf{p}_3\|$. For each curve, we define one deformation coordinate as $y_k = \kappa$. See Fig. 4(d) for a conceptual representation.

To compute the object's curvature with (30), the points $\mathbf{p}_1$ and $\mathbf{p}_3$ must be approximately selected at the extreme positions of the curve, e.g., if the object is completely rounded, these points should represent the poles of the arc; the point $\mathbf{p}_2$ should be approximately selected between these two points.

Remark 6: Note that the proposed deformation features require the imaging system to continuously observe all the feedback feature points $\mathbf{p}_i$; therefore, these manipulation tasks must be locally defined. In addition, it is important to remark that although the selection of points is different from each deformation feature (i.e., there is no unified criterion), a useful guideline for these functionals is to select points that are not too close from each other (failure to do this may create linear dependency problems).

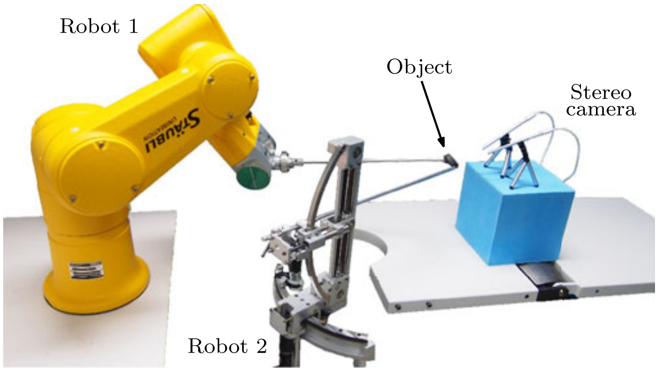


Fig. 5. Setup used for the automatic manipulation experiments.

V. Experimental Validation

A. Setup

We conduct this experimental study with two mechanical systems: a 6-degree-of-freedom (DOF) Staubli robot manipulator and a 3-DOF surgical robot [47] (see Fig. 5). Both manipulators are controlled in velocity mode. To measure the object's deformation, we place artificial markers on its surface and use two CCD cameras to measure the points $\mathbf{p}_i$. We use the OpenCV libraries to process the images and track the deformable feature points [48]. We implement the online adaptive estimator (16) with a gain $\gamma = 10^4$ (we analyze the sensitivity of this gain in following sections) and with an initial vector of parameters $\widehat{\mathbf{a}}(t_0) = [0, \ldots, 0]^\intercal$. To compute the input/output velocity vectors $\dot{\mathbf{x}}$ and $\dot{\mathbf{p}}$, we use standard numerical differentiation. To filter out noise from noisy velocity signals, here generically denoted by $\mathbf{v}_n$, we use the following first-order filter:

$$\dot{\mathbf{v}}(t) = -\xi\left(\mathbf{v}(t) - \mathbf{v}_n(t)\right) \tag{31}$$

where ξ is a positive gain, and $\mathbf{v}$ is the filtered signal. In our experiments, we used $\xi = 30$.

B. Simultaneous Three-Dimensional Position-Shape Control

In this two-robot (i.e., for $M = 2$) experimental study, we test the controller with different manipulation tasks and feedback features. As mentioned in Section IV, the incorporation of the shape term into the manipulation objective allows us to control (to some limited extent determined by the number of robots and feature points) the relative displacements of the soft object. To see this, let us first consider the opposite case where no shape term is used within the deformation vector, i.e., $\mathbf{y} = \mathbf{p}_1$. Fig. 6(a) depicts images of this single-point experiment. Note that for all the experiments presented in this section, the overlaid white and red circles represent the projected 3-D feedback and target positions, respectively.[5] From this figure, we can see that the robots can jointly control the position of the object; however, note that at the final configuration, the object is clearly overstretched by the robots.

[4] Note that $\boldsymbol{\pi}$ and d are only relevant for compressible/stretchable objects.

[5] We project onto the camera's image plane these 3-D vectors in order to visualize the positioning error.

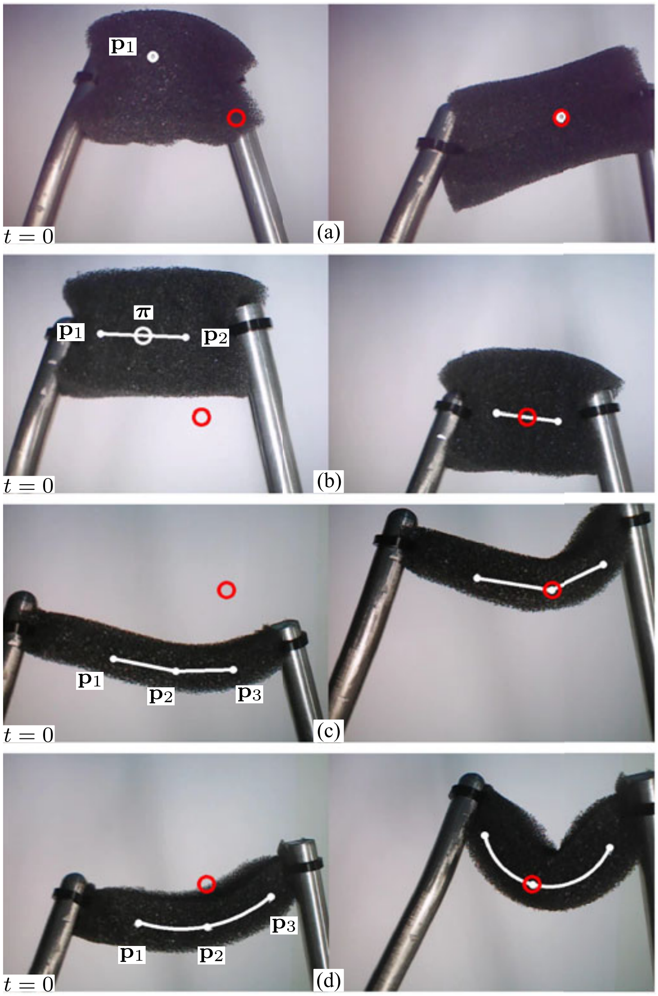


Fig. 6. Captured images of the initial (left) and final (right) configurations of the experiments with (a) one feedback point, (b) centroid (for $C = 2$) and compression distance, (c) angle and point, and (d) curvature and point.

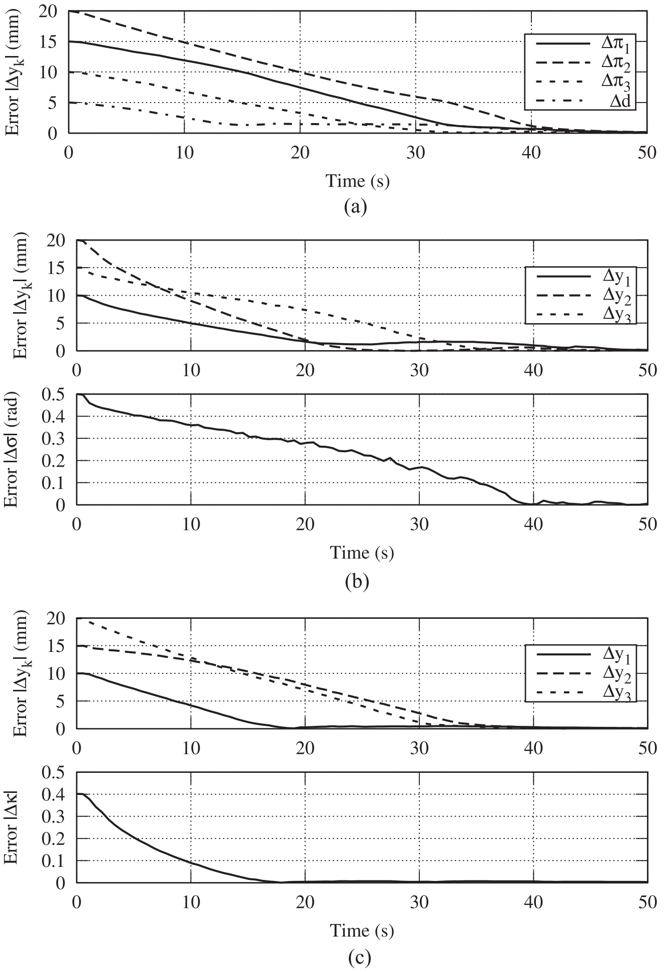


Fig. 7. Deformation errors of the experiments shown in Fig. 6. (a) Error coordinates $\Delta\mathbf{y}^\intercal = [\Delta\boldsymbol{\pi}, \Delta d]$ of the experiment in Fig. 6(b). (b) Error coordinates $\Delta\mathbf{y}^\intercal = [\Delta\mathbf{p}_2, \Delta\sigma]$ of the experiment in Fig. 6(c). (c) Error coordinates $\Delta\mathbf{y}^\intercal = [\Delta\mathbf{p}_2, \Delta\kappa]$ of the experiment in Fig. 6(d).

Now, we test our method with the two-point (i.e., $K = 2$) manipulation task shown in Fig. 6(b). The objective of this experiment is to servo-control the position of the middle point $\boldsymbol{\pi} = (\mathbf{p}_1 + \mathbf{p}_2)/2$ and the scalar distance $d = \|\mathbf{p}_1 + \mathbf{p}_2\|$. We define the deformation vector as

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{\pi}^\intercal & d \end{bmatrix}^\intercal \tag{32}$$

and the target configuration as $\mathbf{y}_d = \mathbf{y}(t_0) + [15,\ 20,\ 10,\ -5]^\intercal$. Fig. 7(a) shows the asymptotic minimization of the positioning/shape errors for this experiment; the resulting trajectories of the measured geometric features are shown in Fig. 8(a). From these results, we can see that robotic manipulators can automatically position the object into a target configuration while keeping a desired compression.

Next, we test our method with the task shown in Fig. 6(c), which uses three feedback points (i.e., $K = 3$) to manipulate a beam-like object. The objective in this experiment is to indirectly control the position of the point $\mathbf{p}_2$ and the scalar angle σ that characterizes the bending of the object. We define the deformation vector as

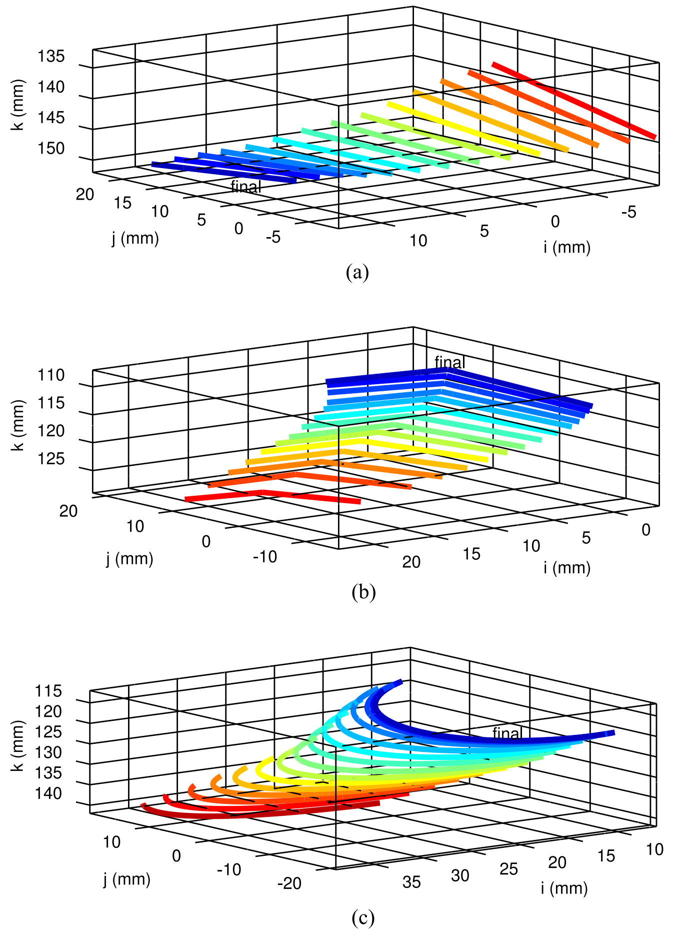
$$\mathbf{y} = \begin{bmatrix} \mathbf{p}_2^\intercal & \sigma \end{bmatrix}^\intercal \tag{33}$$

and the target as $\mathbf{y}_d = \mathbf{y}(t_0) + [10, -20, -5, 0.5]^\intercal$. Fig. 7(b) shows the asymptotic minimization of the errors for this experiment; the Cartesian trajectories of the 3-D points and its line features are shown in Fig. 8(b). These results show that the method can automatically control the object's position while simultaneously imposing a desired bending.

Finally, we evaluate our method with the three-point task shown in Fig. 6(d). The objective of this experiment is to position the deformable point $\mathbf{p}_2$ into a desired location while controlling the curvature κ of the object; we define the deformation vector as

$$\mathbf{y} = \begin{bmatrix} \mathbf{p}_2^\intercal & \kappa \end{bmatrix}^\intercal \tag{34}$$

and the target as $\mathbf{y}_d = \mathbf{y}(t_0) + [-15, -10, -20, 0.4]^\intercal$. Fig. 7(c) shows the asymptotic minimization of the deformation errors. Fig. 8(c) shows the 3-D Cartesian trajectories of the measured shape features.



(a)

(b)

(c)

Fig. 8. Spatial 3-D feature trajectories of the experiments shown in Fig. 7. (a) Trajectory of the compression feature of the experiment in Fig. 6(b). (b) Trajectory of the angle feature in Fig. 6(c). (c) Trajectory of the curvature feature of the experiment in Fig. 6(d).

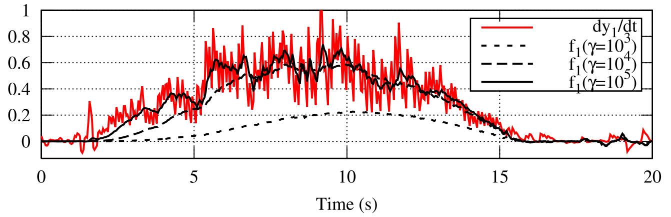
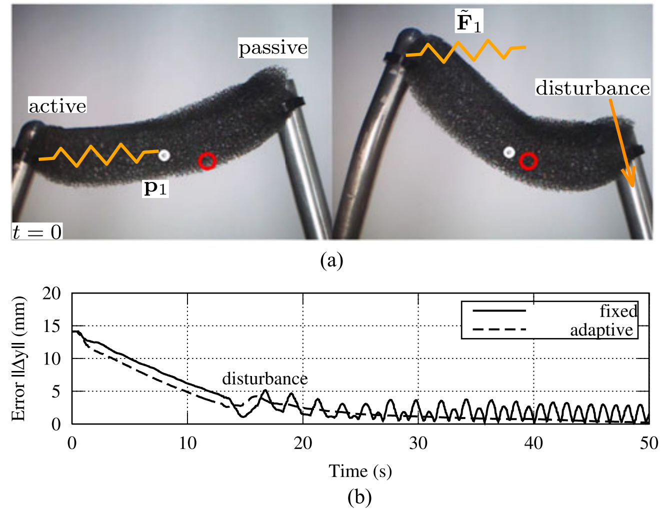


Fig. 9. Graphical comparison of the measured velocity $\dot{\mathbf{y}}$ (red line) and the flow vector $\mathbf{f} = \widehat{\mathbf{J}}\dot{\mathbf{x}}$ (black line) computed with the online estimator, for various values of the learning gain γ.



(a)

(b)

Fig. 10. Fixed/adaptive model comparison experiments. (a) Setup. (b) Norm of the error vector $\Delta\mathbf{y}$.

These simple shape terms can be used to monitor/control the object's relative deformations and, thus, avoid damaging the manipulated object. However, note that this method cannot predict/evaluate the stretching, bending, and compression forces exerted onto the object; this vision-based approach may be used in combination with a force sensor to improve safety.

C. Output Flow Estimation

We evaluate the performance of the proposed online estimator (16) with the two-point manipulation task depicted in Fig. 6(b). The purpose of this experiment is to qualitatively compare the profiles of the measured velocity vector $\dot{\mathbf{y}}$ and the flow vector $\mathbf{f} = \widehat{\mathbf{J}}\dot{\mathbf{x}} \in \mathbb{R}^N$. To this end, we command the manipulators to deform the object along a slow and smooth arbitrary trajectory while estimating the output deformation flows. Fig. 9 depicts the first coordinate of the measured and the estimated flows (i.e., $\dot{y}_1$ and $\dot{f}_1$, respectively) for different learning gains γ. From this figure, we can see that the proposed estimator can approximate the deformations of the object in real time (more specifically, it can approximate the directional change of the deformation features). We can also see that for "small" values (i.e., for $\gamma = 10^3$), the algorithm computes a low-pass filtered version of $\dot{\mathbf{y}}$, which is less susceptible to noisy measurements but poorly approximates rapidly changing configurations—note, however, that most deformation control applications require slow motions. This figure also shows that for "larger" values (i.e., for $\gamma = 10^5$), the algorithm can more accurately reproduce the output velocity $\dot{\mathbf{y}}$, but it also amplifies the measurement noise; special attention must be placed when tuning the online estimator as "large" learning gains tend to result in large oscillations. In our experiments, we obtained good results with values around $\gamma = 10^4$.

To tune the online algorithm, a useful method is to evaluate, for small motion $\|\dot{\mathbf{x}}\| > v_{\text{small}}$, the following metric:

$$\mathcal{A} = \int_{t_1}^{t_2} Q(\tau)\, d\tau \tag{35}$$

for Q defined as in (15). The scalar $\mathcal{A}$ quantifies the accuracy of the estimator for the time period $[t_1, t_2]$ (e.g., from the rising to the settling of $\mathbf{f}$), which allows us to select the learning gain γ that provides the desired response.

D. Fixed Model Versus Adaptive Model

Now, we analyze the differences between the proposed adaptive method and a fixed-model controller. We conduct this experiment with a single robot (i.e., for $M = 1$) and with the single-point manipulation task depicted in Fig. 10(a); similar to the experiment shown in Fig. 6(a), the manipulation objective is simply $\mathbf{y} = \mathbf{p}_1$. Note that in this experiment the right-hand side

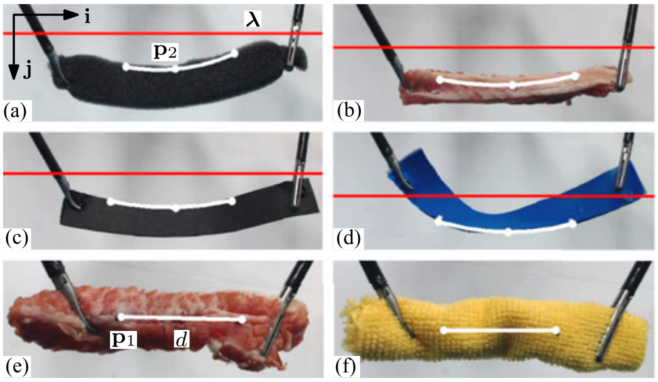


Fig. 11. Manipulation experiments with (a) foam, (b) meat, (c) cardboard, and (d) plastic; compression experiments with (e) meat and (f) towel.

gripper is not considered within the control algorithm; thus, it acts as a fixed manipulation point.

The purpose of experiment is to compare the performance of our adaptive controller and a controller that only considers a fixed (static) model of the soft object. To this end, we first estimate the object's local deformation properties, i.e., around the initial configuration. We compute the *constant* deformation matrix $\tilde{\mathbf{F}}_1 \in \mathbb{R}^{3\times 3}$, which is only valid around the initial configuration of the object (see Fig. 10(a) for a conceptual representation). We implement the following fixed-model velocity controller:

$$\dot{\mathbf{x}} = -\tilde{\mathbf{F}}^{-1}\mathbf{K}\frac{\partial U}{\partial \mathbf{y}}^{\mathsf{T}}. \tag{36}$$

In this experiment, we ask the active robot to drive the feature point into the target. To test the adaptivity of our method, the passive robot introduces a perturbation to the object (at $t = 13$ s), which changes the local deformation properties and, therefore, renders $\tilde{\mathbf{F}}_1$ invalid. Fig. 10(b) depicts the evolution of the error vector $\|\mathbf{y}\|$ for both controllers. This figure shows that after the unmodeled perturbation occurs, the fixed-model controller cannot drive the deformable feature point into the correct directions [this situation is conceptually depicted on the right-hand side of Fig. 10(a)].

E. Deformation Control With Various Materials

In this section, we conduct single-robot manipulation experiments with different materials and stiffness; we use objects made of foam (with a Young's modulus $E = 16.2$ kPa), pork meat ($E = 9.5$ kPa), cardboard ($E = 42.8$ kPa), plastic ($E = 26.4$ kPa), and fabric ($E = 12.1$ kPa). The objective of the first experiment [depicted in Fig. 11(a)–(d)] is to simultaneously control the position coordinates of $\mathbf{p}_2$ and the scalar curvature of the object. Since we only have three control inputs, thus, we define the deformation vector as

$$\mathbf{y} = \begin{bmatrix} p_{y2} & p_{z2} & \kappa \end{bmatrix}^{\mathsf{T}} \tag{37}$$

which only considers positioning along the $(\mathbf{j}, \mathbf{k})$ axes; we define the target configuration as $\mathbf{y}_d = \mathbf{y}(t_0) + [20, 15, 0.4]^{\mathsf{T}}$. Note that the target configuration for $\mathbf{p}_2$ is not a 3-D point but the 2-DOF line $\boldsymbol{\lambda}$ (represented by the overlaid red line) that is parallel to the $\mathbf{i}$-axis. Our aim in these experiments is to evaluate the performance of the shaping controller with different objects and stiffness. For that, we classify the soft objects in Fig. 11(a)–(d) into two categories: 1) stretchable-thick (foam and meat) and 2) nonstretchable-thin (cardboard and plastic).[6] Fig. 12(a) depicts the asymptotic minimization of the deformation errors for each object. In this figure, we can see that the control of the Cartesian coordinates performs generally the same (i.e., with a similar error slope) for all materials. The difference appears in the control of the bending error $\Delta\kappa$, in which the nonstretchable-thin objects exhibit a smoother minimization of $\Delta\kappa$ and the stretchable-thick objects show coupled effects in the bending control. This type of response can be explained by the fact that the latter objects present elastic deviations of its surface's points and since they are thicker require larger motion of the grippers to achieve the target configuration. Deformations of the nonstretchable-thin objects mostly comes along the transversal axis over its surface, which results in hinge-like motions, thus making easier the steering of its features. The objective of the second experiment [depicted in Fig. 11(e) and (f)] is to control the compression of the object and the position of $\mathbf{p}_1$. In this experiment, the robot manipulates foam, meat, and the towel; the deformation vector is defined as

$$\mathbf{y} = \begin{bmatrix} p_{y1} & p_{z1} & d \end{bmatrix}^{\mathsf{T}} \tag{38}$$

and the target configuration as $\mathbf{y}_d = \mathbf{y}(t_0) - [0,\ 0,\ 20]^{\mathsf{T}}$.[7] The asymptotic minimization of the position error $\|\Delta\mathbf{y}\|$ is shown in Fig. 12(b), which shows that the performance with these three stretchable/compressible objects is generally the equivalent. We remark that in this experimental study, we only found significant differences between materials when performing bending tasks.

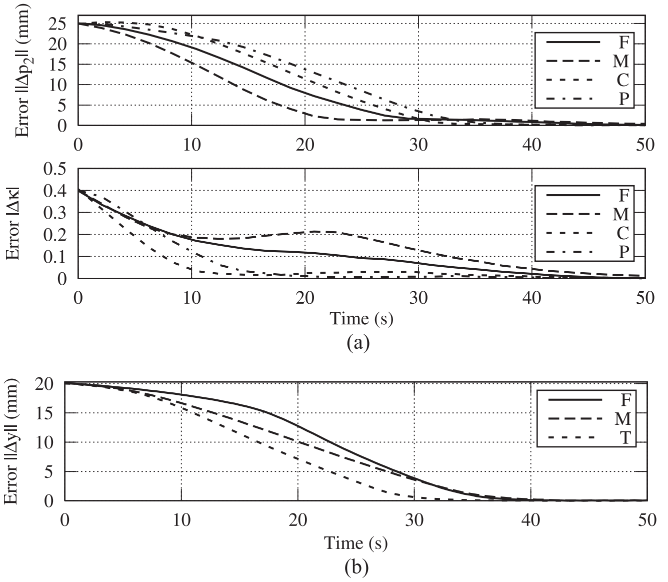


Fig. 12. Deformation errors of the experiments in Fig. 11, where the curve F stands for foam, M for meat, C for cardboard, P for plastic, and T for towel. (a) Position and curvature errors of the experiments in Fig. 11(a)–(d). (b) Position errors of the experiment in Fig. 11(e) and (f).

[6] Note that these objects mostly deform with transversal bending.

[7] To avoid clutter, we omit the target line in Fig. 11(e) and (f).

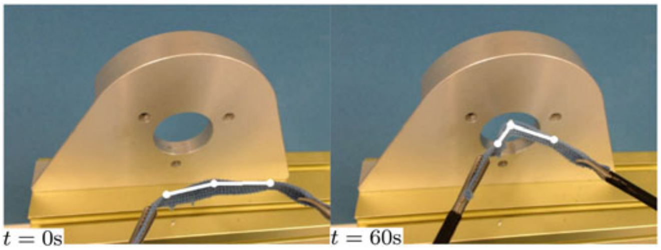


Fig. 13. Image-guided insertion of a soft object into a hole fixture.

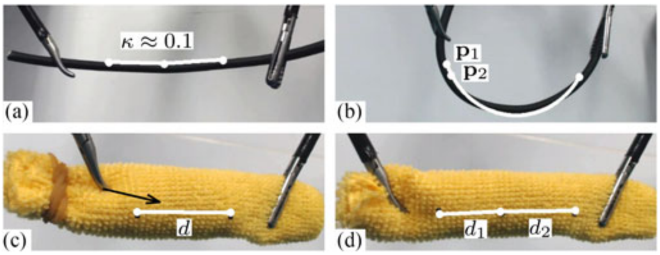


Fig. 14. Situations where the controller performs poorly. (a) Very small curvature. (b) Improper selection of points. (c) Nonrigid grasping. (d) Linearly dependent deformation features.

F. Case of Study: Image-Guided "Cloth in Hole"

In this section, we use the proposed method to automatically place a ribbon-like object inside a hole fixture, an application that requires to first fold the object and then guide its vertex into the target location. To perform this task, we construct the deformation vector $\mathbf{y} = [\mathbf{p}_2^\intercal, \sigma]^\intercal$ with point and angle features, as in Fig. 6(c). We use two robots to manipulate a thin piece of towel. The objective of this experiment is to insert the object 10 mm into the fixture with a folding angle of $80°$. Fig. 13 shows the captured initial and final images of a sample manipulation experiment. To guarantee that the object can be introduced into the fixture, we use higher feedback gain for the angle coordinate, i.e., $\mathbf{K} = 0.001\,\mathrm{diag}(\mathbf{I}_{3\times 3}, 4)$ (this helps to arrive at the hole with the desired bending angle). In the accompanying video, we illustrate the performance of the proposed approach with several objects (e.g., meat, foam, fabric, cable, cardboard, and plastic) and experiments.

G. Limitations of the Approach

Finally, we present and discuss situations that should be avoided when implementing the proposed manipulation approach; see Fig. 14 for a conceptual representation of these configurations. To measure (approximately) the shape of the object, in Section IV we proposed a set a deformation features; however, these features can only be used to describe/control the shape in local manner. This situation is depicted in Fig. 14(a), which shows an object with very small curvature, a configuration that should be avoided as singularities may arise (a similar situation arises with angles computed from almost parallel lines). Another issue that must be considered is proper selection of feature points. Fig. 14(b) depicts the curvature feature computed with points $\mathbf{p}_1$ and $\mathbf{p}_2$ very close to each other; from this figure, we can clearly see that κ poorly approximates the object's bending. To transmit motion to the feature points, the method that we propose requires that the robots rigidly grasp the object; point-like pushing interactions [as depicted in Fig. 14(c)] cannot, in general, be used to control deformations. Finally, we remark that the deformation feature $\mathbf{y}$ must be defined such that its coordinates locally provide a linearly independent (and representative) measurement of the object's shape. Fig. 14(d) depicts an example of two linearly dependent distances d_j computed along the same direction (this configuration should be avoided).

VI. Conclusion

In this paper, we have presented a new method to automatically servo-control 3-D elastic deformations by robotic manipulators. To avoid the identification of the object's deformation model, we first designed an online algorithm that iteratively estimates the unknown deformation parameters. Next, we proposed a model-free controller that exploits the estimated parameters to compute a velocity input that minimizes the feedback deformation vector. Then, several deformation features were proposed to characterize the manipulation task. Finally, we presented an experimental study with multiple manipulation tasks to validate the presented control approach.

The proposed algorithm estimates the relation between the manipulator's input motion and the 3-D velocity of deformable feature points. Note that the sole purpose of this algorithm is to compute a vector of parameters that can accurately reconstruct the measured output velocity, i.e., it does not guarantee the identification of the true deformation model. Besides estimating deformations, the proposed algorithm can also cope with uncalibrated kinematic transformations among the manipulators and the 3-D imaging system. The proposed controller uses the stereo vision to impose a servo-loop in terms of feedback 3-D shapes, something that is not straightforward to do with most image-based methods, as depth information is lost with the perspective projection of a point. In this work, we characterized the manipulation task using a feedback position term (which determines the 3-D position) and a feedback shape term (which determines the internal displacements). The simultaneous control of these two quantities is easier when multiple robots manipulate the object.

There are several limitations for the automatic control method that we proposed. For example, to measure the 3-D positions of the feature points, we need a calibrated stereo rig; this requirement may cause some problems for single camera applications, or when the calibration matrices are not available. Note that to compute the deformation features (which are locally defined), we must first place fiducial markers on the object, which may not be feasible in some applications. We remark that the method that we propose is designed for slow motions of the robots and cannot be used to control rapidly changing shapes. The approximation of the object's model as purely elastic may also limit the applicability of our method; the proposed algorithm can only be used with "mostly elastic" objects that present no rheological deformations. An important aspect to keep in mind when implementing this method is that the target configuration must be feasible, i.e., physically reachable with the manipulation system. Also note that different applications have different error acceptability, e.g., assembling soft parts in manufacturing may require higher accuracy than positioning elastic tissues in surgery.

In this work, we analytically formulated the problem from a simple Cartesian feature point perspective. As future work, we would like to develop simpler (and perhaps more computationally oriented) methods to control shapes without the need to use artificial markers. For that, we want to implement new feedback deformation functions, e.g., contours or surface areas, which can be directly computed from sensor feedback (using stereo vision or RGB-D sensors). Another interesting research direction is to model (and control) physical interactions between the manipulated soft object and a rigid environment; this approach can be used, for example, in assembly tasks with soft objects.

Appendix A
Algebraic Stereo Triangulation

Let the homogeneous vectors $\mathbf{m}_i = \mathbf{H}[\mathbf{p}_i^{\intercal}, 1]^{\intercal} \in \mathbb{R}^3$ and $\overline{\mathbf{m}}_i = \overline{\mathbf{H}}[\mathbf{p}_i^{\intercal}, 1]^{\intercal} \in \mathbb{R}^3$ represent the perspective projection of the feature point $\mathbf{p}_i$ on two image planes with camera matrices $\mathbf{H}, \overline{\mathbf{H}} \in \mathbb{R}^{3\times4}$. From the properties of the vector cross product, we know that the following relation satisfies

$$\underbrace{\begin{bmatrix}[m_i]\times\mathbf{H}\\ [\overline{m_i}]\times\overline{\mathbf{H}}\end{bmatrix}}_{\mathrm{s}}\begin{bmatrix}\mathbf{P}_i\\ 1\end{bmatrix} = \mathbf{0}_{6\times1} \qquad (39)$$

where $[\cdot]_\times \in \mathbb{R}^{3\times3}$ denotes the skew-symmetric cross product matrix operator. Let $\mathbf{l} = [l_1, l_2, l_3, l_4] \in \mathbb{R}^4$ represent the eigenvector of $\mathbf{S}^{\intercal}\mathbf{S}$ with the smallest eigenvalue; the Cartesian coordinates of the feature point are computed by

$$\mathbf{p}_i = \begin{bmatrix} l_1/l_4 & l_2/l_4 & l_3/l_4 \end{bmatrix}^{\intercal}. \qquad (40)$$

Appendix B
Implementation Algorithm

```
Initialize parameters $\widehat{\mathbf{a}} \leftarrow$ random values
**While** Deformation errors $|\Delta y_k| \geq \epsilon_k$ **do**
    Measure joint positions $\mathbf{q}_j$ and image points $\mathbf{m}_i, \overline{\mathbf{m}}_i$
    Compute point's 3-D positions $\mathbf{p}_i \leftarrow$ (40)
    Compute deformation controller $\dot{\mathbf{x}} \leftarrow$ (20)
    Solve the robot's velocity $\dot{\mathbf{x}}_j$ for $\dot{\mathbf{q}}_j$ in (1)
    Command joint motions $\dot{\mathbf{q}}_j$
    Update deformation parameters $\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} \leftarrow$ (16)
**end while**
```

We use the positive scalars ϵ_k to specify the error acceptability in different applications.

References

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

[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. Robot. Autom.*, 2011, pp. 3893–3900.

[3] E. Park and J. Mills, "Static shape and vibration control of flexible payloads with applications to robotic assembly," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 6, pp. 675–687, Dec. 2005.

[4] H. Wakamatsu, E. Arai, and S. Hirai, "Knotting/unknotting manipulation of deformable linear objects," *Int. J. Robot. Res.*, vol. 25, no. 4, pp. 371–395, Apr. 2006.

[5] P. Culmer, J. Barrie, R. Hewson, M. Levesley, M. Mon-Williams, D. Jayne, and A. Neville, "Reviewing the technological challenges associated with the development of a laparoscopic palpation device," *Int. J. Med. Robot. Comput.*, vol. 8, no. 2, pp. 146–159, 2012.

[6] D. Henrich and H. Wörn, Eds., *Robot Manipulation of Deformable Objects*. New York, NY, USA: Springer, 2000.

[7] D. Sun and Y. H. Liu, "Modeling and impedance control of a two-manipulator system handling a flexible beam," *ASME J. Dyn. Syst., Meas., Control*, vol. 119, no. 4, pp. 736–742, Dec. 1997.

[8] D. Sun and Y.-H. Liu, "Position and force tracking of a two-manipulator system manipulating a flexible beam," *J. Robot. Syst.*, vol. 18, no. 4, pp. 197–212, Mar. 2001.

[9] H. Wakamatsu and S. Hirai, "Static modeling of linear object deformation based on differential geometry," *Int. J. Robot. Res.*, vol. 23, no. 3, pp. 293–311, Mar. 2004.

[10] S. Yue and D. Henrich, "Manipulating deformable linear objects: Sensor-based skills of adjustment motions for vibration reduction," *J. Robot. Syst.*, vol. 22, no. 2, pp. 67–85, Feb. 2005.

[11] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 48–68, 2014.

[12] S. Hirai, T. Tsuboi, and T. Wada, "Robust grasping manipulation of deformable objects," in *Proc. IEEE Int. Symp. Assembly Task Planning*, 2001, pp. 411–416.

[13] K. G. Gopalakrishnan and K. Goldberg, "D-space and deform closure grasps of deformable parts," *Int. J. Robot. Res.*, vol. 24, no. 11, pp. 899–910, Nov. 2005.

[14] Y.-B. Jia, F. Guo, and H. Lin, "Grasping deformable planar objects: Squeeze, stick/slip analysis, and energy-based optimalities," *Int. J. Robot. Res.*, vol. 34, pp. 1–32, Feb. 2014.

[15] F. Lamiraux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *Int. J. Robot. Res.*, vol. 20, no. 3, pp. 188–208, Mar. 2001.

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

[17] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation," in *Proc. Int. Conf. Adv. Comput.-Human Interactions*, 2009, pp. 199–204.

[18] S. Patil, J. van den Berg, and R. Alterovitz, "Motion planning under uncertainty in highly deformable environments," in *Proc. Robot., Sci. Syst. Conf.*, 2011, pp. 1–8.

[19] M. Kimura, Y. Sugiyama, S. Tomokuni, and S. Hirai, "Constructing rheologically deformable virtual objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, vol. 3, pp. 3737–3743.

[20] 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. Robot. Autom.*, 2010, pp. 5120–5126.

[21] 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.

[22] A. Pettersson, T. Ohlsson, S. Davis, J. Gray, and T. Dodd, "A hygienically designed force gripper for flexible handling of variable and easily damaged natural food products," *Innov. Food Sci. Emerg. Technol.*, vol. 12, no. 3, pp. 344–351, Jul. 2011.

[23] 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.

[24] 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.

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

[26] J. Das and N. Sarkar, “Passivity-based target manipulation inside a deformable object by a robotic system with noncollocated feedback,” *Adv. Robot.*, vol. 27, no. 11, pp. 861–875, May 2013.

[27] 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.

[28] C. G. Broyden, “A class of methods for solving nonlinear simultaneous equations,” *Math. Comput.*, vol. 19, pp. 577–593, Oct. 1965.

[29] K. Hosoda and M. Asada, “Versatile visual servoing without knowledge of true Jacobian,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1994, vol. 1, pp. 186–193.

[30] M. Jagersand, O. Fuentes, and R. Nelson, “Experimental evaluation of uncalibrated visual servoing for precision manipulation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, vol. 4, pp. 2874–2880.

[31] D. Navarro-Alarcon, Y.-H. Liu, J. Romero, and P. Li, “On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments,” *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1462–1480, Jun. 2014.

[32] 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/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4977–4982.

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

[34] A. AlbuShffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, and G. Hirzinger, “The DLR lightweight robot: design and control concepts for robots in human environments,” *Ind. Robot: An Int. J.*, vol. 34, no. 5, pp. 376–385, 2007.

[35] P. Boonvisut and M. Cavusoglu, “Estimation of soft tissue mechanical parameters from robotic manipulation data,” *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 5, pp. 1602–1611, Oct. 2013.

[36] D. Berenson, “Manipulation of deformable objects without modeling and simulating deformation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4525–4532.

[37] J. K. Mills and J. G.-L. Ing, “Dynamic modeling and control of a multi-robot system for assembly of flexible payloads with applications to automotive body assembly,” *J. Robot. Syst.*, vol. 13, no. 12, pp. 817–836, Dec. 1996.

[38] 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.

[39] A. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*, 2nd ed. London, U.K.: Springer, 2000.

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

[41] B. Siciliano, “Kinematic control of redundant robot manipulators: A tutorial,” *J. Intell. Robot. Syst.*, vol. 3, no. 3, pp. 201–212, 1990.

[42] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization* (ser. Addison-Wesley Series in Electrical and Computer Engineering: Control Engineering). Boston, MA, USA: Addison-Wesley Longman, 1991.

[43] 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.

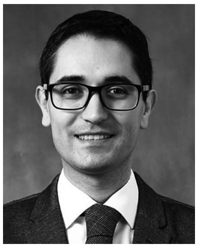
[44] C. C. Cheah, C. Liu, and J. J. E. Slotine, “Adaptive Jacobian vision based control for robots with uncertain depth information,” *Automatica*, vol. 46, no. 7, pp. 1228–1233, Jul. 2010.

[45] S. Arimoto, “A class of quasi-natural potentials and hyper-stable PID servo-loops for nonlinear robotic systems,” *Trans. SICE*, vol. 30, no. 9, pp. 1005–1012, 1994.

[46] C. Kimberling, *Triangle Centers and Central Triangles: By Clark Kimberling* (ser. Congressus Numerantum). Winnipeg, MB, Canada: Utilitas Mathematica, 1998.

[47] H. M. Yip, Z. Wang, D. Navarro-Alarcon, P. Li, Y.-H. Liu, and T. Cheung, “A new robotic uterine positioner for laparoscopic hysterectomy with passive safety mechanisms: Design and experiments,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3188–3194.

[48] G. Bradski, “The OpenCV library,” *Dr. Dobb’s J. Softw. Tools*, vol. 25, no. 11, pp. 120, 122–125, Nov. 2000.



David Navarro-Alarcon (S’06–M’14) received the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), Shatin, Hong Kong, in 2014.

He was a Postdoctoral Fellow with CUHK from 2014 to 2015, where he is currently a Research Assistant Professor. He is also an Associate Researcher with CUHK Shenzhen Research Institute, Shenzhen, China. His research interests include robot manipulation, medical robotics, and control engineering.

Hiu Man Yip (S’14) received the B.Eng. and M.Phil. degrees in mechanical and automation engineering from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2010 and 2013, respectively, where she has been working toward the Ph.D. degree since Fall 2013.

Her research interests include robotic systems design and development of medical devices and surgical robots.

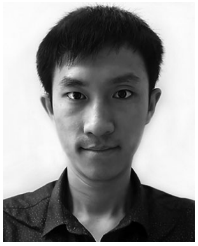
Zerui Wang (S’15) received the B.Eng. degree in reliability and system engineering from Beihang University, Beijing, China, in 2013. He is currently working toward the Ph.D. degree in mechanical and automation engineering with The Chinese University of Hong Kong, Shatin, Hong Kong.

His research interests include mechanical design of flexible joints and visual servoing.

Yun-Hui Liu (S’90–M’92–SM’98–F’09) received the B.Eng. degree in applied dynamics from Beijing Institute of Technology, Beijing, China, in 1985; the M.Eng. degree in mechanical engineering from Osaka University, Osaka, Japan, in 1989; and the Ph.D. degree in mathematical engineering and information physics from University of Tokyo, Tokyo, Japan, in 1992.

He was with the Electrotechnical Laboratory, Ministry of International Trade and Industry, Ibaraki, Japan, from 1992 to 1995. Since February 1995, he has been with The Chinese University of Hong Kong (CUHK), Shatin, Hong Kong, where he is currently a Professor with the Department of Mechanical and Automation Engineering and the Director of the CUHK T Stone Robotics Institute. He is also visiting State Key Lab of Robotics Technology and System, Harbin Institute of Technology, Harbin, China, and the Director of Joint Centre for Intelligent Sensing and Systems, National University of Defense Technology, Hunan, China, and CUHK. He has published more than 200 papers in refereed journals and refereed conference proceedings and was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, medical robotics, multifingered robot hands, mobile robots, sensor networks, and machine intelligence.

Dr. Liu has received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He is the Editor-in-Chief of *Robotics and Biomimetics* and an Editor of *Advanced Robotics*. He served as an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.



Fangxun Zhong (S'15) received the B.Eng. degree in automation from Beijing Institute of Technology, Beijing, China, in 2014 and the M.Sc. degree in mechanical and automation engineering from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2014, where he has been working toward the Ph.D. degree since Fall 2015.

His research interests include vision-based robot manipulation and control, surgical robots, and image processing.

Tianxue Zhang (S'15) received the B.Eng. degree in electrical engineering from Beijing Institute of Technology, Beijing, China, in 2014. She has been working toward the Ph.D. degree in mechanical and automation engineering with The Chinese University of Hong Kong, Shatin, Hong Kong, since Fall 2014.

Her research interests include robotic systems design and development of medical devices and interventional robots.

Peng Li received the B.Eng. degree in mechanical engineering from North Eastern University, Shenyang, China, in 2004 and the Ph.D. degree in mechatronics from Shenyang Institute of Automaton, Chinese Academy of Sciences, Shenyang, in 2010.

From 2009 to 2010, he was with The Chinese University of Hong Kong as a Research Assistant, where since 2010 he has been a Postdoctoral Researcher with the Department of Mechanical and Automation Engineering. His research interests include development of automatic surgical systems, medical devices, and pipeline robots.