Fourier-Based Shape Servoing: A New Feedback Method to Actively Deform Soft Objects into Desired 2-D Image Contours

David Navarro-Alarcon, *Member, IEEE*, and Yun-Hui Liu, *Fellow, IEEE*

***Abstract*—This paper addresses the design of a vision-based method to automatically deform soft objects into desired two-dimensional shapes with robot manipulators. The method presents an innovative feedback representation of the object's shape (based on a truncated Fourier series) and effectively exploits it to guide the soft object manipulation task. A new model calibration scheme that iteratively approximates a local deformation model from vision and motion sensory feedback is derived; this estimation method allows us to manipulate objects with unknown deformation properties. Pseudocode algorithms are presented to facilitate the implementation of the controller. Numerical simulations and experiments are reported to validate this new approach.**

***Index Terms*—Adaptive control, robots manipulators, shape control, soft objects, visual servoing.**

I. Introduction

The feedback control of soft object deformations has many promising applications in growing fields, such as medical robotics (e.g., manipulating soft tissues [1], [2]), automated food processing (shaping compliant food materials [3], [4]), 3C industry (positioning cables [5], [6]), and the garment industry (folding fabrics [7]), to name a few cases. To servo-control deformations, a robotic system must continuously measure the object's configuration, typically with a vision sensor, and use its feedback data to compute a trajectory that deforms the object into a desired shape. We refer to this type of manipulation task as *shape servoing*, an approach that contrasts with standard (fixed-camera) visual servoing [8], in that the servo loop is formulated in terms of the object's shape and not in terms of the robot's pose.

Recently, in [9]–[11], we proposed methods to automatically position deformable point features of objects toward desired targets (this positioning problem was first addressed in [12] and later extended in [13]–[15]). These approaches can control deformations in a model-free manner; however, they can only be used with structured objects that have a well-defined texture. Furthermore, when the objective is to deform the body into a desired image contour, these methods may not perform well as they are designed to control displacements of individual point features. The design of explicit shape servo-controllers present many challenges, such as the characterization of the object's shape as a feedback signal, the limited number of manipulation points to enforce a desired shape, and the estimation of the deformation properties to coordinate the robot's motion with the shape measurements. Our aim in this paper is to develop a new method to cope with the above-mentioned issues.

There are some works that have previously addressed shape control. For example, the development of a forming machine for food dough is presented in [16]; this image-guided system can predict deformations of rheological materials while controlling its shape. A method to estimate and control contours of soft objects with a multiple robotic fingers is proposed in [17]; however, this work requires objects to have a well-structured (gridded) surface. An approach to control the profile of an elastic body by several manipulators is reported in [18]; this work only conducted simulations. In [19], a method to position fabrics into image profiles is presented; the same group in [20] developed a planning algorithm for robotic laundry. A method to manipulate extensible fabrics and ropes is presented in [21]; this model-based method approximates the deformation properties with a diminishing rigidity model.

There are some relevant visual servoing works that also use two-dimensional (2-D) contours. An early example is given in [22], where polar coordinates are used for representing the observed shape and to compute the controller. Other methods are given, e.g., in [23] (where a controller based on a correspondence/collineation algorithm is proposed), [24] (where the controller is computed with image moments of contours), and [25] (where the object's boundaries are approximated with polynomials). Note, however, that these visual servoing methods do not address shape control of deformable objects. Its objective is (generally) to match the target and observed object's contours by performing rigid motions on a robotic camera.

This paper presents a new feedback method to automatically deform soft objects into desired 2-D image contours. The object's shape is characterized with a compact vector of Fourier coefficients, a useful approach that enables the specification of the deformation tasks with frequency terms and not only with displacements (as with traditional methods [12]). To deal with the input–output coordination problem, a model calibration scheme that iteratively estimates the deformation model from sensor feedback is derived; this algorithm allows us to manipulate objects without having to compute its full deformation model beforehand.[1] We provide several algorithms to facilitate the implementation of the controller. Numerical simulations and experiments are presented to validate the method.

Compared with our previous works [9]–[11], this paper has the following original contributions: 1) It introduces a novel feedback characterization of deformable shapes—based on truncated Fourier series—and effectively utilizes it to perform closed-loop manipulation of soft objects (to the best of our knowledge, this innovative *shape vector* has never been used for controlling deformable shapes); and 2) it presents

Manuscript received July 25, 2017; accepted September 23, 2017. Date of publication November 13, 2017; date of current version February 5, 2018. This paper was recommended for publication by Associate Editor C. Bergeles and Editor A. Billard upon evaluation of the reviewers' comments. This work was supported in part by the Hong Kong Research Grants Council (HK RGC) under Grant 14203917, in part by the Hong Kong Polytechnic University (HKPolyU) under Grant 4-ZZHJ, and in part by the Chinese University of Hong Kong Shun Hing Institute of Advanced Engineering (CUHK SHIAE) under Grant 8115053. *(Corresponding author: David Navarro-Alarcon.)*

D. Navarro-Alarcon is with Hong Kong Polytechnic University, Hong Kong (e-mail: dna@ieee.org).

Y.-H. Liu is with Chinese University of Hong Kong, Hong Kong (e-mail: yhliu@mae.cuhk.edu.hk).

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.2017.2765333

[1] Note that most previous works (e.g., [12]–[21]) are model-based, i.e., the deformation model must be computed prior the active control task.

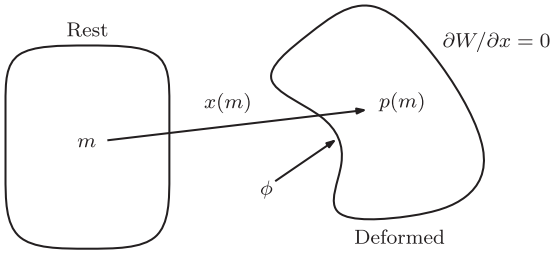

Fig. 1. Conceptual representation of the object's deformations.

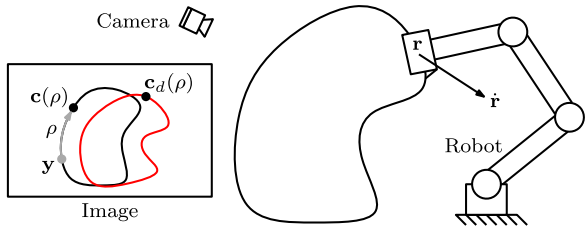

Fig. 2. Conceptual representation of the soft object manipulation setup.

a new estimation–recalibration algorithm that continuously approximates the model with multiple sensory data points (collected along the trajectory) and not just with an instantaneous measurement as in our previous work (this new algorithm is more robust to the noisy image measurements that are common in most visual servo applications).

The rest of this paper is organized as follows. Section II introduces the problem's preliminaries, Section III describes the new method, Section IV presents the results, and Section V gives the final conclusion.

II. Preliminaries

A. Continuum Deformation Model

Let us introduce m to model the rest position of a point, given in material coordinates over the object domain Π. When an external force ϕ is applied, the point m is displaced to a different position $p(m)$; the relative deformation is represented by the displacement vector field $x(m) = p(m) - m$. The object's potential energy satisfies the energy balance [26]

$$W = \frac{1}{2}\int_{\Pi} E(m) : \Sigma(m)\,\mathrm{d}m - \int_{\Pi} \phi \cdot x(m)\,\mathrm{d}m \tag{1}$$

where E and Σ represent the elastic strain and stress tensors, respectively, and $(:)$ denotes the double tensor contraction. The left-hand-side term in (1) represents the stored energy; the right-hand-side term represents the externally applied work. For deformable objects, the shape is determined by the following static equilibrium equation (see Fig. 1):

$$\partial W/\partial x = 0. \tag{2}$$

B. Problem Formulation

Consider a robotic system manipulating a soft object. Throughout this paper, we denote by $\mathbf{r} \in \mathbb{R}^M$ the vector of M manipulation coordinates that have a measurable influence in the object's deformation. Typically, $\mathbf{r}$ may contain the robot's plane motions that are parallel to the camera plane. Depending on the camera/grasping setup, the "depth" motion and/or a gripper rotation may additionally be considered. To derive our new method, the following setup assumption must be made.[2]

Assumptions: Consider the following task assumptions.

1) The robot manipulator is kinematically controlled, i.e., its servo-controller renders stiff behaviors and precisely sets $\mathbf{r}(t)$ at any time instant t.
2) The soft object is rigidly grasped by the manipulator without any loose contact during the whole task.
3) The object is manipulated with quasi-static robot motions such that its shape is determined by the potential energy terms only.[3]

[2]If these are violated: 1) the object may be wrongly deformed; 2) our new algorithm may be unstable; 3) the deformation model may not be estimated.

[3]Note that for slow motions, the object's inertial, rheological, and viscous effects are minimal, therefore, these can be safely neglected for our model.

A static camera system is used to continuously observe the object's shape, here represented with the parametric curve $\mathbf{c}(\rho) = [u(\rho), v(\rho)]^\intercal$, for u and v as the image coordinates, and ρ as the parametric domain. The curve $\mathbf{c}(\rho)$ is computed relative to a visible origin point $\mathbf{y}$ that can physically represent, e.g., the manipulator's grasping point or a static fixture. Note that the analytical expression of $\mathbf{c}(\rho)$ is not known, as this curve only represents the captured 2-D image contour, see Fig. 2.

Problem statement: Given a desired contour $\mathbf{c}_d(\rho)$, design a model-free method [i.e., without any knowledge of the equilibrium model (2)] to automatically deform the object such that the feedback shape $\mathbf{c}(\rho)$ approximates $\mathbf{c}_d(\rho)$.

III. Methods

A. Feedback Shape Parameters

Since no information is available about the object's model, we set the range of the contour's parameter to $\rho = [0, 2\pi)$, and approximate $\mathbf{c}(\rho)$ with the following truncated Fourier series:

$$\mathbf{c}(\rho) = \sum_{j=1}^{N} \begin{bmatrix} a_j & b_j \\ c_j & d_j \end{bmatrix} \begin{bmatrix} \cos(j\rho) \\ \sin(j\rho) \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{3}$$

for $N > 0$ as the number of harmonics taken into account, and $a_j, b_j, c_j, d_j, e,$ and f as the Fourier coefficients. Note that (3) can be linearly parametrized as

$$\mathbf{c}(\rho) = \sum_{j=1}^{N} \underbrace{\begin{bmatrix} \cos(j\rho) & \sin(j\rho) & 0 & 0 \\ 0 & 0 & \cos(j\rho) & \sin(j\rho) \end{bmatrix}}_{\mathbf{F}_j} \mathbf{z}_j + \begin{bmatrix} e \\ f \end{bmatrix}$$

$$= \mathbf{G}(\rho)\mathbf{s} \tag{4}$$

where the vector $\mathbf{s} = [\mathbf{z}_1^\intercal, \ldots, \mathbf{z}_N^\intercal, e, f]^\intercal \in \mathbb{R}^P$, for $P = 4N + 2$ and $\mathbf{z}_j^\intercal = [a_j, b_j, c_j, d_j]$ groups the shape parameters of the contour, and $\mathbf{G}(\rho) = [\mathbf{F}_1, \ldots, \mathbf{F}_N, \mathbf{I}] \in \mathbb{R}^{2\times P}$ represents a regression-like matrix. Using L sample data points of the parameter ρ and its corresponding image points on the contour $\mathbf{c}(\rho)$, we construct the following "long" structures:

$$\vec{\mathbf{c}} = \begin{bmatrix} \mathbf{c}(\rho_1)^\intercal & \cdots & \mathbf{c}(\rho_L)^\intercal \end{bmatrix}^\intercal \in \mathbb{R}^{2L} \tag{5}$$

$$\vec{\mathbf{G}} = \begin{bmatrix} \mathbf{G}(\rho_1)^\intercal & \cdots & \mathbf{G}(\rho_L)^\intercal \end{bmatrix}^\intercal \in \mathbb{R}^{2L\times P}. \tag{6}$$

The vector of shape parameters $\mathbf{s}$ is computed from sensor feedback at every iteration as

$$\mathbf{s} = \left(\vec{\mathbf{G}}^\intercal \vec{\mathbf{G}}\right)^{-1} \vec{\mathbf{G}}^\intercal \vec{\mathbf{c}}. \tag{7}$$

To invert the matrix $\vec{\mathbf{G}}^\intercal\vec{\mathbf{G}}$, a sufficient number of data points $\rho_1, \ldots, \rho_L$ must be used such that $2L > P$, see e.g., [27].

B. Approximation of the Local Deformation Model

The shape parameters $\mathbf{s}$ computed with the Fourier approximation (3) can be used to reconstruct the object's contour $\mathbf{c}$ with the parameter $\rho = [0, 2\pi)$. However, this static relation only satisfies for the specific position $\mathbf{r}$ of the robot where the vector $\mathbf{s}$ is computed. By keeping the number of harmonics constant during the manipulation task, it is reasonable to expect that a small change in the robot's position produces a small change in the shape parameters. From this observation, the position–shape relation is *locally* approximated (around the current operating point, assumed to be regular and with no singular configurations) with the model

$$\delta\mathbf{s} = \mathbf{Q}\delta\mathbf{r}. \tag{8}$$

where $\mathbf{Q} \in \mathbb{R}^{P\times M}$ represents a constant matrix modeling the relative deformation of the object during manipulation, and the vectors $\delta\mathbf{s} = \mathbf{s} - \bar{\mathbf{s}}$ and $\delta\mathbf{r} = \mathbf{r} - \bar{\mathbf{r}}$ denote the relative change of the shape parameters and the robot position, respectively, with respect to the initial configurations $\bar{\mathbf{s}}$ and $\bar{\mathbf{r}}$.

To deform the object into a target shape, our method does not estimate the full deformation model. Instead, it continuously computes local approximations of (8), which are updated as the object's configuration evolves from one shape into another. To this end, let us first denote by δs_i the ith scalar element of the vector $\delta\mathbf{s}$, and by $\mathbf{q}_i^\intercal \in \mathbb{R}^M$ the ith row-vector of the matrix $\mathbf{Q}$, such that

$$\delta s_i = \delta\mathbf{r}^\intercal\mathbf{q}_i \tag{9}$$

To estimate the object's model, we collect $T > M$ sampling data points of δs_i and $\delta\mathbf{r}$ around the operating point and group them into the terms

$$\boldsymbol{\sigma}_i = \begin{bmatrix} \delta s_i(t_1) \\ \vdots \\ \delta s_i(t_T) \end{bmatrix} \in \mathbb{R}^T, \quad \mathbf{R} = \begin{bmatrix} \delta\mathbf{r}(t_1)^\intercal \\ \vdots \\ \delta\mathbf{r}(t_T)^\intercal \end{bmatrix} \in \mathbb{R}^{T\times M}. \tag{10}$$

The ith energy function is computed at the time instant t as

$$J_i(t) = \left(\|\delta\mathbf{r}(t)^\intercal\widehat{\mathbf{q}}_i - \delta s_i(t)\|^2 + \|\mathbf{R}\widehat{\mathbf{q}}_i - \boldsymbol{\sigma}_i\|^2\right)/2 \tag{11}$$

where $\widehat{\mathbf{q}}_i \in \mathbb{R}^M$ denotes an estimation of $\mathbf{q}_i$. The left term in (11) represents the model error for the current (time changing) measurement and the right term represents the error for the old (static) data points. The purpose of the functional J_i is to quantify the accuracy of the estimated model (9); the error for all the parameters can be computed as $J = \sum_{i=1}^{P} J_i$. The estimated vector $\widehat{\mathbf{q}}_i$ is updated according to the rule

$$\dot{\widehat{\mathbf{q}}}_i = -\gamma\frac{\partial J_i}{\partial\widehat{\mathbf{q}}_i}^\intercal = -\gamma\mathbf{H}^\intercal\begin{bmatrix} \mathbf{R}\widehat{\mathbf{q}}_i - \boldsymbol{\sigma}_i \\ \delta\mathbf{r}^\intercal\widehat{\mathbf{q}}_i - \delta s_i \end{bmatrix} \tag{12}$$

for $\gamma > 0$ as a tuning gain, and the matrix $\mathbf{H} \in \mathbb{R}^{(T+1)\times M}$ constructed as

$$\mathbf{H} = \begin{bmatrix} \delta\mathbf{r}(t_1) & \cdots & \delta\mathbf{r}(t_T) & \delta\mathbf{r} \end{bmatrix}^\intercal = \begin{bmatrix} \mathbf{R}^\intercal & \delta\mathbf{r} \end{bmatrix}^\intercal. \tag{13}$$

The implementation of the update rule is described in Algorithm 1, where ε represents a scalar to control the algorithm's accuracy, and $\widehat{\mathbf{q}}_i$ is initialized with a null vector since the model is unknown.

Proposition: Consider that locally, the model (8) closely approximates the object's relative deformations. For a number M of linearly independent displacement vectors $\delta\mathbf{r}(t_k)$ such that the matrix $\mathbf{H}^\intercal$ has full row-rank, the update rule (12) asymptotically minimizes the error $\Delta\mathbf{q}_i = \widehat{\mathbf{q}}_i - \mathbf{q}_i$.

Proof: Consider the quadratic energy function

$$V_i = \Delta\mathbf{q}_i^\intercal\Delta\mathbf{q}_i/(2\gamma). \tag{14}$$

Algorithm 1: UpdateRule(ε).

```
if estimator is uninitialized then
    Initialize r̄ ← r, s̄_i ← s_i, q̂_i ← 0_M
end if
Compute relative vectors δr ← r − r̄, δs_i ← s_i − s̄_i
Compute energy functional J_i ←(11)
if J_i > ε then
    Update deformation model d/dt q̂_i ←(12)
end if
return q̂_i
```

Using the expressions $\boldsymbol{\sigma}_i = \mathbf{R}\mathbf{q}$ and (9), the time derivative of V_i along trajectories of (12) satisfies

$$\begin{aligned} \dot{V}_i &= \Delta\mathbf{q}_i^\intercal\dot{\widehat{\mathbf{q}}}_i/\gamma = -\Delta\mathbf{q}_i^\intercal\mathbf{H}^\intercal\begin{bmatrix} \mathbf{R}\widehat{\mathbf{q}}_i - \mathbf{R}\mathbf{q}_i \\ \delta\mathbf{r}^\intercal\widehat{\mathbf{q}}_i - \delta\mathbf{r}^\intercal\mathbf{q}_i \end{bmatrix} \\ &= -\Delta\mathbf{q}_i^\intercal\mathbf{H}^\intercal\begin{bmatrix} \mathbf{R}^\intercal & \delta\mathbf{r} \end{bmatrix}^\intercal\Delta\mathbf{q}_i = -\Delta\mathbf{q}_i^\intercal\mathbf{H}^\intercal\mathbf{H}\Delta\mathbf{q}_i \\ &\leq 0 \end{aligned} \tag{15}$$

and, since $\mathbf{H}^\intercal$ has full row-rank, $\mathbf{H}^\intercal\mathbf{H}$ thus is positive definite. This proves the asymptotic stability of the error $\Delta\mathbf{q}_i$ [28]. ■

C. Iterative Model Recalibration

Accurate deformation models are difficult to compute due to its high dimension, nonlinear behavior, and configuration-dependent properties. The full characterization of a soft body typically requires evaluation of the deformation response around the desired operating range (see, e.g., [29]), which may be impractical to do in applications where (*a-priori*) testing deformations cannot be performed as they may damage the object, e.g., when manipulating tissues.

As a solution to this problem, the proposed method computes a local deformation model and recalibrates it at different local regions. It first computes an initial approximation of (8) around the starting configuration $\mathcal{O}_j$ and then, as the robot moves into a new configuration $\mathcal{O}_{j+1}$, the model is recalibrated based on new data. The advantages of using this approach are 1) its computational burden is considerably smaller (and thus suitable for real-time applications) compared to the full model identification since only a simple linear model needs to be computed, and 2) it robustifies the estimation since the model is updated with new feedback data collected around the new configuration.

The proposed recalibration scheme is given in pseudocode in Algorithm 2 and is depicted in Fig. 3. In this implementation, the arbitrary scalar α determines the separation between each sampling points; this separation is evaluated by computing the magnitude $\|\mathbf{d}\|$ between the positions $\delta\mathbf{r}(t_1)$ and $\delta\mathbf{r}(t_2)$. The point $\mathbf{o}$ represents the midpoint of the T collected data points $\delta\mathbf{r}(t_k)$, and is used to define the centre of a ball with radius Ω, where the model is assumed to be valid (in our study, we determined these values experimentally).

D. Shape Servo-Controller

Let us first represent the desired parametric contour $\mathbf{c}_d(\rho)$ as a "long" vector of L sample points

$$\vec{\mathbf{c}}_d = \begin{bmatrix} \mathbf{c}_d(\rho_1)^\intercal & \cdots & \mathbf{c}_d(\rho_L)^\intercal \end{bmatrix}^\intercal \in \mathbb{R}^{2L}. \tag{16}$$

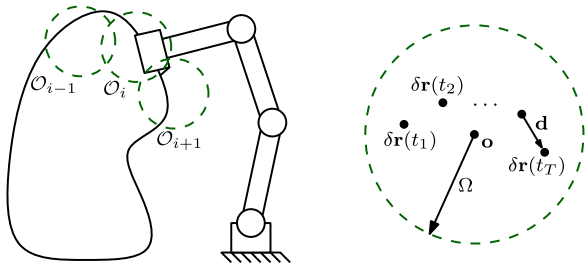

Fig. 3. Representation of the local regions computed with the T data points.

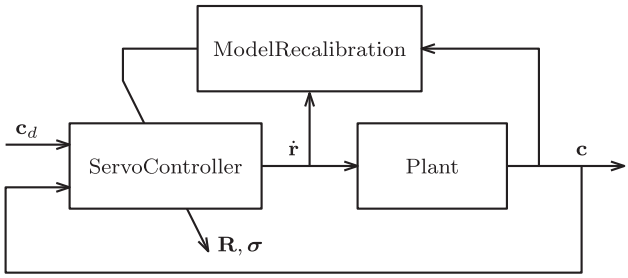

Fig. 4. Schematic representation of the shape servo-controller.

Algorithm 2: ModelRecalibration(T, α).

Initialize $\delta\mathbf{r}^{old} \leftarrow \delta\mathbf{r}(t_{k-1})$, $pts \leftarrow 0$
while $pts < T$ **do**
  Compute distance $\mathbf{d} \leftarrow \delta\mathbf{r}(t_k) - \delta\mathbf{r}^{old}$
  if $\|\mathbf{d}\| > \alpha$ **then**
    Stack data $\mathbf{R} \leftarrow \delta\mathbf{r}(t_k)$, $\boldsymbol{\sigma}_i \leftarrow \delta s_i(t_k)$
    Update $\delta\mathbf{r}^{old} \leftarrow \delta\mathbf{r}(t_k)$, $pts \leftarrow pts + 1$
  end if
end while
Compute midpoint $\mathbf{o} \leftarrow \frac{1}{T}\sum_{k=1}^{T} \delta\mathbf{r}(t_k)$
return $\mathbf{R}, \boldsymbol{\sigma}_i$

Algorithm 3: ServoController$(\vec{\mathbf{c}}_d, \varepsilon, T, \alpha, \Omega, \eta)$.

Conduct small testing deformations around the starting configuration, and compute an initial model $\mathbf{Q}$
repeat
  Measure position $\mathbf{r}$ and contour $\mathbf{c}(\rho)$
  Compute feedback shape parameters $\mathbf{s} \leftarrow$(7)
  Compute deformation model $\widehat{\mathbf{q}}_i \leftarrow$ UpdateRule(ε)
  Compute controller $\dot{\mathbf{r}} \leftarrow$(19)
  if Outside local region $\|\mathbf{r} - \mathbf{o}\| > \Omega$ **then**
    $\mathbf{R}, \boldsymbol{\sigma}_i \leftarrow \text{Async}(\text{ModelRecalibration}(T, \alpha))$
  end if
until $\|\Delta\mathbf{s}\| < \eta$

Then, by considering the same number N of harmonics as in (4), the desired shape vector is computed as

$$\mathbf{s}_d = \left(\vec{\mathbf{G}}^{\intercal}\vec{\mathbf{G}}\right)^{-1}\vec{\mathbf{G}}^{\intercal}\vec{\mathbf{c}}_d. \tag{17}$$

The shape error $\Delta\mathbf{s} = \mathbf{s} - \mathbf{s}_d$ represents the difference between the Fourier coefficients of the feedback and target shapes. This P-dimensional error is directly used to drive the motion of the manipulator such that $\mathbf{s}_d$ is approximated by the observed object at the steady state. This Fourier-based controller contrasts with previous methods, e.g., [9]–[12], in that the latter can only perform manipulation tasks represented with point displacements, whereas this new frequency-motivated approach provides more flexibility in the task specification.

The control law is designed using a Jacobian-based approach [30]. The matrix $\mathbf{Q}$ represents the Jacobian matrix, which is constructed with the P adaptive vectors

$$\widehat{\mathbf{Q}} = \begin{bmatrix}\widehat{\mathbf{q}}_1 & \cdots & \widehat{\mathbf{q}}_P\end{bmatrix}^{\intercal} \in \mathbb{R}^{P\times M}. \tag{18}$$

The velocity control input of the robot is computed as

$$\dot{\mathbf{r}} = -\lambda\left(\widehat{\mathbf{Q}}^{\intercal}\widehat{\mathbf{Q}}\right)^{-1}\widehat{\mathbf{Q}}^{\intercal}\,\text{sat}(\Delta\mathbf{s}) \tag{19}$$

where λ is a feedback gain and $\text{sat}(\cdot) : \mathbb{R}^P \mapsto \mathbb{R}^P$ is a vectorial saturation function. It is well known that the overdetermined controller (19) (with more error outputs than controllable inputs) cannot guarantee global asymptotic stability of $\|\Delta\mathbf{s}\|$; this vector will typically converge to a steady-state error whose magnitude depends on the feasibility of the target. For more information about overdetermined visual servoing, we refer the reader to [31] for a classical stability analysis or to [32] for an experimental performance evaluation.

The objective of $\text{sat}(\cdot)$ is to guide the system toward the target through incremental errors that can be locally achieved without converging to an intermediate local minimum. A simple test to determine the feasibility of a *local* target is to evaluate $\|(\widehat{\mathbf{Q}}^{\intercal}\widehat{\mathbf{Q}})^{-1}\widehat{\mathbf{Q}}^{\intercal}\,\text{sat}(\Delta\mathbf{s})\| \leq \zeta$ (values smaller than the arbitrary scalar ζ can tell if the robot is approaching to a local minimum). Note that this model-free test method cannot, in general, be used to determine the feasibility of a final shape from the starting configuration.

A schematic representation of the controller is depicted in Fig. 4, and its implementation is given in Algorithm 3. The variables Ω and η are used to determine the size of the local region and to specify the error acceptability, respectively. At the beginning of the method, small testing deformations are conducted to obtain a rough initial estimation of $\mathbf{Q}$. The function Asyn() is used to represent an asynchronous call of the ModelRecalibration() routine, which in our method runs in a parallel and slower thread than the main servo-loop; this function is only called when the robot leaves the local region; thus, the model needs to be recalibrated.

IV. Results

A. Numerical Simulation

A circular deformable object is computed using a discrete mesh with 200 points (note that this numerical object is used as a "black box"). The soft body is rigidly attached to a fixed surface, while a 3-DOF planar robot rigidly grasps a point on its surface. The algorithm is written in C++ and the object/robot are visualized with OpenCV.

We first evaluate the performance of the model recalibration scheme. For that, the object's contour is approximated with $N = 5$ harmonics, which gives total of $P = 22$ feedback Fourier coefficients. The algorithm is implemented considering $T = 5$ sampling points, with a local region of $\Omega = 40$ pixels, and a learning gain of $\gamma = 5 \times 10^{-6}$. Fig. 5 shows the results of estimating the object's shape around different configurations. The black profile represents the observed contour $\mathbf{c}(\rho)$, whereas the green profile represents the contour estimated as

$$\widehat{\mathbf{c}} = \mathbf{G}(\rho)(\widehat{\mathbf{Q}}\delta\mathbf{r} + \bar{\mathbf{s}}). \tag{20}$$

From these figures, we can see that as the robot moves away from $\mathcal{O}_j$ (i.e., t_2 and t_3), the estimated shape deviates from the real one. The computed model is corrected as new data are collected around $\mathcal{O}_{j+1}$. Fig. 6 depicts the minimization process for the energy function

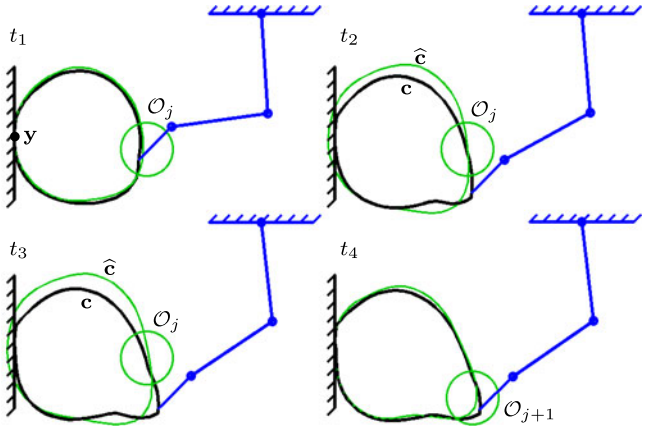

Fig. 5. Shape estimation for different configurations from $t = t_1$ to $t = t_4$.

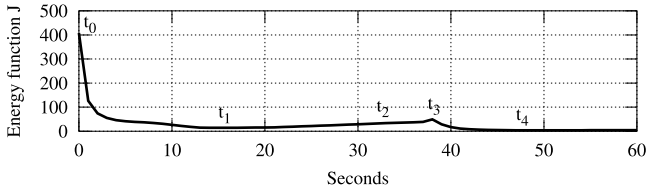

Fig. 6. Minimization process of the energy function J.

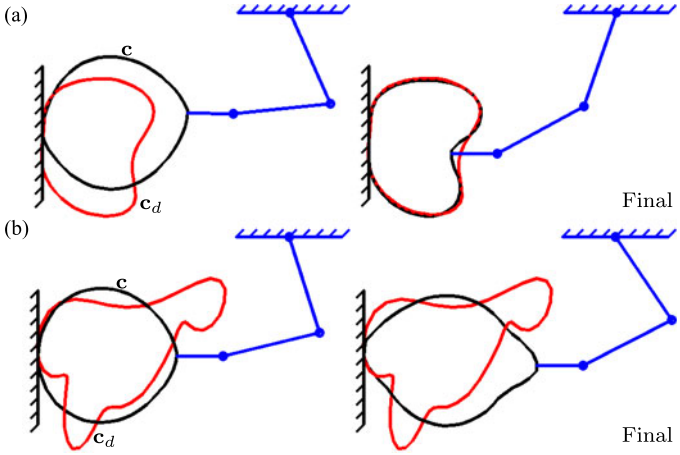

Fig. 7. Initial and final configurations of the shape control simulation.

J. During the estimator's initialization at $t = t_0$, only the $T = 5$ data points are available and no knowledge of the model is assumed (i.e., $\widehat{\mathbf{q}}_i = \mathbf{0}_M$). As J starts to increase, the algorithm recalibrates the model to improve the accuracy. The controller (19) is implemented with saturation bounds of $|\,\mathrm{sat}(\Delta s_i)| \leq 3$ and gain $\lambda = 0.1$. Two tasks are considered: The first is a feasible object compression task, whereas the second is a task with an infeasible target shape. Fig. 7 depicts the initial and final object's configurations, where the red curves represent the target profiles $\mathbf{c}_d(\rho)$. Fig. 8 depicts the evolution of the magnitude $\|\Delta \mathbf{s}\|$ for both tasks. These curves show that, for the former task, the error is asymptotically minimized, whereas for the latter one the error converges to a large local minimum.

B. Experiments

The performance of the proposed method is evaluated by conducting a series of experiments using a 2-DOF (i.e., $M = 2$) motorized linear

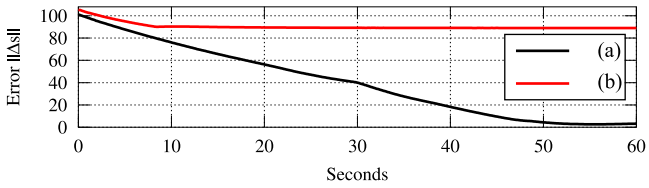

Fig. 8. Evolution of the error $\|\Delta \mathbf{s}\|$ for the tasks shown in Fig. 7(a) and (b).

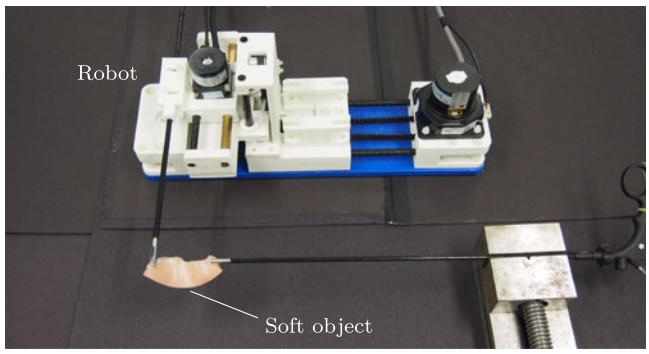

Fig. 9. Experimental setup.

Fig. 10. Feedback contour $\mathbf{c}(\rho)$ computed with the captured image.

stage that manipulates soft silicon phantom models. Fig. 9 shows the experimental robot, whose joints are controlled using a position-stepping mode. The object is manipulated by one controllable gripper, while a second passive gripper is used as a supporting point. A Logitech C920 camera captures images of the object; these are processed with a Linux-based PC at 30 frame per second. The same software and tuning parameters of the previous numerical simulation are used in this manipulation experiments.

The contour is computed from a binary image of the object, where the reference point $\mathbf{y}$ represents the grasping point of the passive gripper. Fig. 10 shows the segmented image and the parametric contour $\mathbf{c}(\rho)$ (blue curve). Note that since only $N = 5$ harmonics are used, the contour $\mathbf{c}(\rho)$ does not exactly match the profile of the object. Using a "small" number N is useful for the method since it allows us to 1) reduce the computational complexity of the shape parametrization (7) (making it suitable for real-time applications) and 2) only feed the most shape-significant errors Δs_i into the motion controller (19) (this is particularly important as a high-dimensional vector $\Delta \mathbf{s}$ characterising many contour details might be difficult to minimize with a limited number of controllable grippers). For clarity's sake, this blue (feedback) contour is not displayed in the following experiments, only the target contour is shown.

The performance of the controller is evaluated with multiple experiments using different sizes of objects. At the beginning of each task, the controller only has access to the $T = 5$ local sampling points that are collected with small testing motions, as described in Algorithm 3. Fig. 11 depicts the initial and final configurations for these experi-

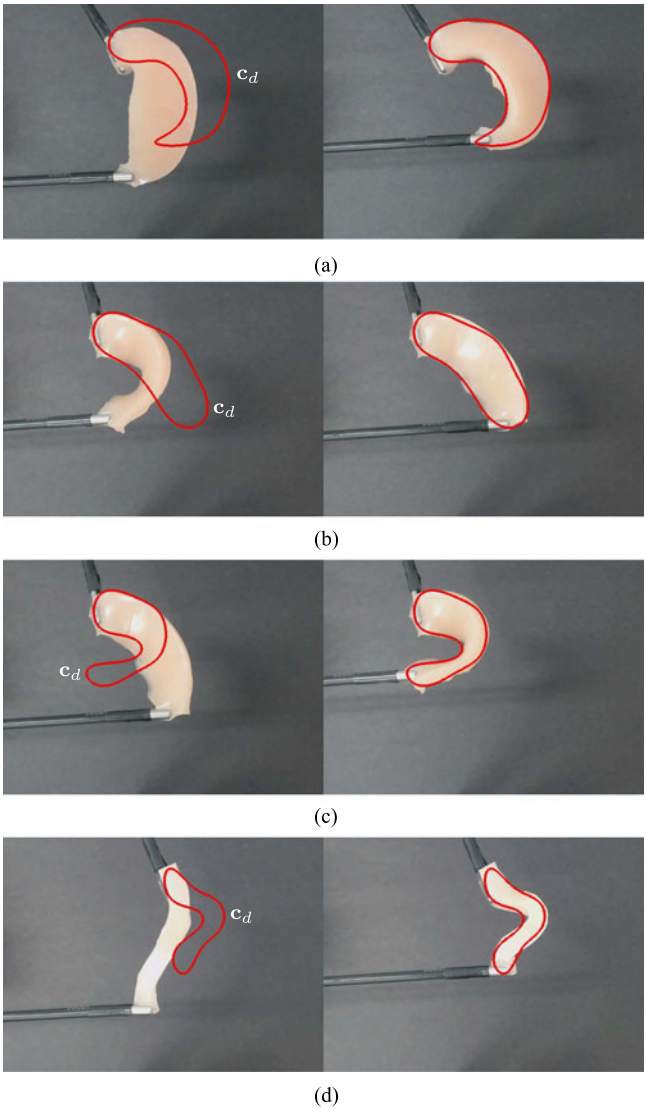

Fig. 11. Initial (left) and final (right) configurations for various experiments.

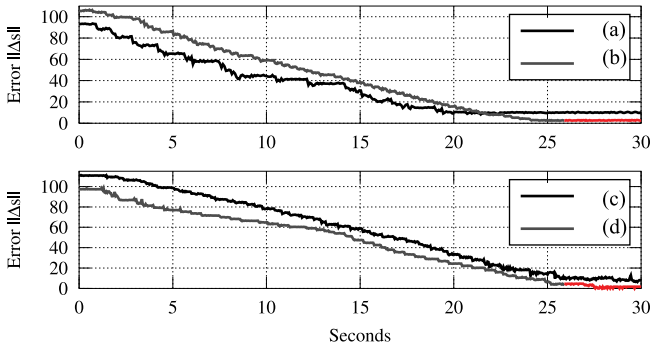

Fig. 12. Minimization of $\|\Delta\mathbf{s}\|$ for the experiments shown in Fig. 11(a)–(d).

ments. The red curve represents the constant target shape $\mathbf{c}_d$, which is transformed into a desired vector $\mathbf{s}_d$ using (17). In this experimental study, the shape servoing tasks are conducted without the identification of the object's deformation model. The minimization of the magnitude error $\|\Delta\mathbf{s}\|$ for each experiment is shown in Fig. 12. These results corroborate that the new Fourier-based task specification can be used

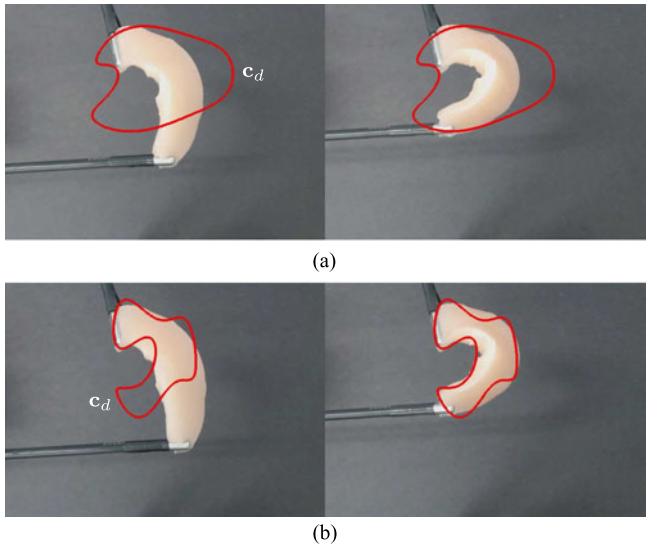

Fig. 13. Experiments with infeasible target shapes that result in local minima.

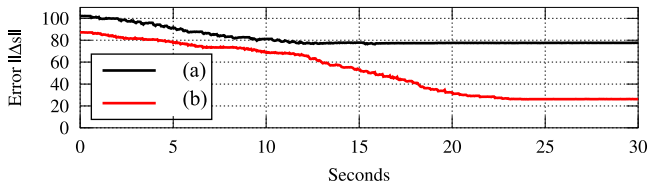

Fig. 14. Evolution of $\|\Delta\mathbf{s}\|$ for the experiments shown in Fig. 13(a) and (b).

to actively deform soft objects into desired 2-D shapes; note that this is achieved without using any special texture/fiducials on the object.

To obtain a good contour match (e.g., the results depicted in Fig. 11), the desired shape must be physically reachable with the setup being used, namely the object being shaped and the robotic manipulator. Similarly to the result shown in Fig. 7(b), Fig. 13 now presents two examples of target shapes that cannot be accurately enforced onto that particular object and with a single manipulation point (these targets may require, e.g., stretching/shaping of the object from multiple points). Note, however, that for this situation, the controller can still be used to manipulate the object into coarsely approximated target; the desired contour is not exactly enforced and instead the object arrives at a configuration that numerically represents a large local minimum. This scenario can be clearly seen in Fig. 14, which shows that the evolution of the magnitude error $\|\Delta\mathbf{s}\|$ during these experiments converges to a steady-state error comparatively larger than those in Fig. 12.

By using a small number of harmonics (five in this study), the object's feedback contour $\mathbf{c}(\rho)$ is coarsely approximated since high-frequency components (i.e., small details) are filtered out. This simplification of the feedback contour can be used by the proposed Fourier-based method to manipulate objects with complex profiles. Fig. 15 depicts two manipulation experiments using a complete phantom uterus model that has considerably more details than the simpler partial objects used in Fig. 11. These experiments are performed with one active gripper and two passive supporting grippers. For these types of manipulation tasks, the local target contour $\mathbf{c}_d$ approximately represents the boundary, where the object is to be positioned by the robot. Fig. 16 shows the time evolution of the shape error $\|\Delta\mathbf{s}\|$ for these manipulation experiments. Note that since the object shape does not exactly match the target contour, the same steady-state errors arise as with the previous experiment.

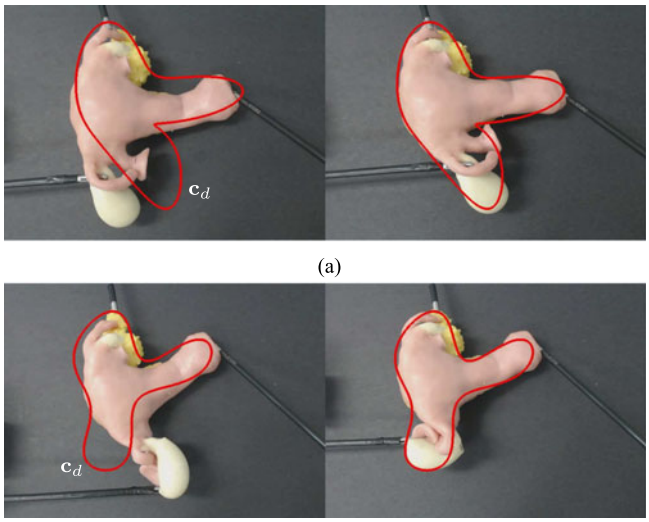

Fig. 15. Manipulation experiments of soft objects with complex shapes.

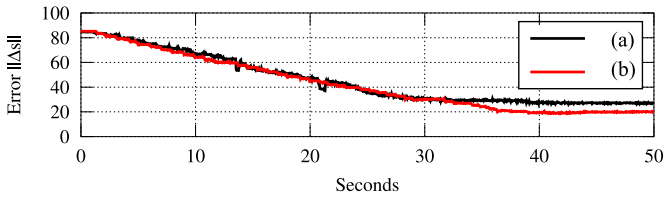

Fig. 16. Minimization of the shape error $\|\Delta \mathbf{s}\|$ for the manipulation experiments with complex objects shown in Fig. 15(a) and (b).

V. Conclusion

In this paper, we present a new vision-based control method to automatically manipulate deformable objects. First, a compact feedback characterization of deformable object shapes is developed. Next, an adaptive update rule that estimates the object's model is derived. Then, the shape servoing controller and its implementation algorithm are presented. Finally, numerical and experimental results are reported to validate the approach.

Our method introduces a new paradigm to specify closed-loop deformation tasks based on truncated Fourier series of 2-D image contours, an approach that contrasts with previous methods that typically use the displacements of feature points to define the task. The core idea behind the controller design is to establish an explicit shape servo loop by feeding back a vector of Fourier coefficients. This design provides more flexibility for describing the feedback shape and its target configuration (e.g., $\mathbf{c}_d(\rho)$ can be roughly sketched on a screen), and allows us to manipulate soft objects with unstructured surfaces (i.e., without special fiducials markers or texture). To avoid the full parametric identification of the object's model, an online algorithm that approximates the deformation properties is proposed. The key idea behind the algorithm is to iteratively recalibrate an approximated linear model computed from multiple data points collected along the local trajectory. The estimated coarse linear model is used within an overdetermined kinematic control law to steer the object toward the desired target shape. In our experimental study, we found out that the use of the saturation function helps us to avoid large steady-state errors, as no specific error coordinate dominates the whole trajectory.

The proposed method has many limitations. For example, the segmentation of the manipulated object may be complicated to perform for scenes with poor illumination, similar background/object colors, blurred edges, etc. (in our experiments, we simplified this situation by using high contrast between the object and the background). A clear disadvantage of the proposed *model-free* method is its inability to determine whether a final target shape is physically reachable or not before the task is performed (note that the current control formulation assumes that the object can be deformed into $\mathbf{c}_d$, and along a connected motion trajectory). Although the method does not require to use fiducial markers, it does need to observe one clearly distinguishable feature point (i.e., $\mathbf{y}$) to define the origin of the feedback contour (in our experiments, we used the position of the static gripper). The objective of the iterative model estimator is to compute a local approximation of the relation between the robot's motion and the shape parameters. However, it must be remarked that since the estimator computes a linearized model that is valid only locally, a single approximation cannot, in general, be used to model the whole properties of the object. Finally, the design of the robot motion controller is not restricted to plane motions; however, note that the proposed method can only control 2-D image projections of the observed shapes.

For future work, we would like to perform shape control experiments but now we are using multiple active manipulators; this new setup will allow us to deform the object into more complex shapes as compared with a single manipulation point. Also, we would like to test the performance of our controller but using other model estimators, e.g., velocity-based algorithms, such as the Broyden update rule [32]. An interesting research direction is to use the feedback shape data to compute a deformation model using the finite element method; the estimated physical model could be used to more accurately predict the feasibility of a given target $\mathbf{c}_d(\rho)$. The natural extension of this paper is to work on the development of a new manipulation approach but to deform objects into desired 3-D shapes/surfaces. For that, coarse physical models and active 3-D vision systems might be needed. We encourage readers to work into this direction.

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. Torabi, K. Hauser, R. Alterovitz, V. Duindam, and K. Goldberg, "Guiding medical needles using single-point tissue manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 2705–2710.
[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] 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.
[5] 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.
[6] 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.
[7] M. Bell, "Flexible object manipulation," Ph.D. dissertation, Dartmouth College, Hanover, NH, USA, 2010.
[8] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
[9] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1457–1468, Dec. 2013.
[10] 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, 2014.
[11] D. Navarro-Alarcon *et al.*, "Automatic 3D manipulation of soft objects by robotic arms with adaptive deformation model," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 429–441, Apr. 2016.

[12] 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.
[13] M. Shibata and S. Hirai, "Soft object manipulation by simultaneous control of motion and deformation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2460–2465.
[14] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators," *J. Intell. Robot. Syst.*, vol. 62, pp. 3–27, 2011.
[15] S. Kinio and A. Patriciu, "A comparative study of H_∞ and PID control for indirect deformable object manipulation," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2012, pp. 414–420.
[16] 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.
[17] A.-M. Cretu, P. Payeur, and E. M. Petriu, "Soft object deformation monitoring and learning for model-based robotic hand manipulation," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 740–753, Jun. 2012.
[18] J. Das and N. Sarkar, "Passivity-based target manipulation inside a deformable object by a robotic system with noncollocated feedback," *Adv. Robotics*, vol. 27, no. 11, pp. 861–875, May 2013.
[19] 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.
[20] S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 249–267, 2011.
[21] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4525–4532.
[22] C. Collewet and F. Chaumette, "A contour approach for image-based control on objects with complex shape," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2000, vol. 1, pp. 751–756.
[23] E. Malis, G. Chesi, and R. Cipolla, "2 1/2D visual servoing with respect to planar contours having complex and unknown shapes," *Int. J. Robot. Res.*, vol. 22, no. 10-11, pp. 841–853, 2003.
[24] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.
[25] A. Y. Yazicioglu, B. Calli, and M. Unel, "Image based visual servoing using algebraic curves applied to shape alignment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 5444–5449.
[26] S. Martin, "Flexible, unified and directable methods for simulating deformable objects," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2015.
[27] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems* (ser. Scripta series in mathematics). Great Falls, MT, USA: Winston, 1977.
[28] J.-J. Slotine and W. Li, *Applied Nonlinear Control*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1991.
[29] 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.
[30] 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.
[31] 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.
[32] 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.