A Unified Controller for Region-reaching and Deforming of Soft Objects

Zerui Wang, Xiang Li, David Navarro-Alarcon, and Yun-hui Liu

Abstract— Emerging applications of robotic manipulation of deformable objects have opened up new challenges in robot control. While several control techniques have been developed to manipulate deformable objects, the performance of existing methods is commonly limited by several issues: 1) implicit assumption that the physical contact between the end effector and the object is always maintained; 2) requirements of exact parameters of deformation model, which are difficult to obtain; and 3) stability or effectiveness in scenarios of low-speed manipulation. This paper presents a new control scheme for robotic manipulation of deformable objects, which allows the robot to automatically contact then manipulate the deformable object by assessing the status of deformation in real time. Instead of designing multiple controllers and switching among them, the proposed method smoothly and stably integrates two control phases (i.e. region reaching and active deforming) into a single controller. The stability of the closed-loop system is rigorously proved with the consideration of uncertain robot dynamics, unknown deformation model, and also uncalibrated cameras. Hence, the proposed control scheme guarantees the feasibility, the autonomous capability, and the stability of active deformation of deformable objects. Experimental results in a medical robot system (da Vinci Research Kit) have been presented to illustrate the performance of the proposed controller.

I. Introduction

Object manipulation is the most fundamental and common task when a robot is interacting with its surrounding environment [1]. The robotic manipulation with rigid object has been extensively and successfully studied in terms of motion control [2], motion planning [3], force control [4], etc, with applications in a number of traditional fields, e.g., pick and place in PCB manufacturing line, painting and welding in vehicle assembly line and so on [5]. However, the manipulation of deformable objects is inevitable in areas like food industry [6], robotic laundry [7], 3C industry [8], and robotic surgery [9], in which robots have to interact with objects that are inherently "soft".

The objective of robotic manipulation of deformable objects can be defined as providing desired configurations (e.g., shape) by the interaction with robots. Several works have been reported in the literature for robotic manipulation of deformable objects. Using mass-spring model, people studied the deformation control of one dimensional and two dimensional objects [8], [10]–[12]. For better accuracy, people used finite element analysis to model the deformable objects [6], [13], [14]. However, the computation is too intensive to be used in the real time control. Therefore, these approaches suffer either from the pool accuracy or from the intensive computation. Moreover, the approaches need prior knowledge (e.g., parameters) of the object deformation model, which makes it less applicable for deformation control. On the other hand, instead of relying on deformation models with parameter identification, in [15]–[17], people used numerical methods to estimate the deformation property so as to control it. Some others used regressor-based methods to control the deformation, and meanwhile online estimate the deformation parameters [18], [19]. The advantage of these approaches comes from the fact that they do not need to identify the parameters of deformation model beforehand.

The aforementioned results commonly require the exact parameters of deformation model, which is usually varying with different material, dimension, and size of deformable objects and hence difficult to obtain. While the adaptive control is able to estimate the unknown parameters online [18], [19], existing adaptive controllers are limited to kinematic levels, in the sense that the manipulation speed of the robot is low such that the robot dynamics is negligible. Moreover, all the works mentioned above are based on an implicit assumption that the robot end-effector is connected with the deformable object initially and rigidly during the manipulation, which actually implies that a certain level of human assistance is still required to initiate or re-establish the physical contact between the robot end effector and the object whenever it is lost.

To overcome the limitations, a new vision-based adaptive controller is proposed for robotic manipulation of deformable objects in this paper, which is able to improve the autonomous capability of existing results and also guarantee the dynamic stability of closed-loop systems in the presence of unknown parameters of deformation model and camera model. In particular, the proposed controller enables the robot to automatically contact then actively deform the deformable object into a desired configuration, by using the vision feedback to assess the deformation in real time. Instead of having two standalone controllers for contact and active deformation separately and switching among them, the proposed method provides a unified formation by integrating two control phases into a single controller in a smooth and

This work is supported in part by the HK RGC under grants 415011, 14203917, and CUHK6/CRF/13G, by the HK ITF under grants ITS/112/15FP and ITT/012/15GP, by the VC's Discretionary Fund through CURI grant 4930763 and 4930725, by the CUHK SHIAE under grant 8115053.

Z. Wang, X. Li and Y.-H. Liu are with the Department of Mechanical and Automation Engineering, the T Stone Robotics Institute, The Chinese University of Hong Kong, HKSAR.

D. Navarro-Alarcon is with the Department of Mechanical Engineering, Hong Kong Polytechnic University, HKSAR.

stable manner. Novel adaptation laws are developed to update the unknown parameters of robot dynamic model, deformation model of objects, and camera model concurrently and separately. The dynamic stability of the closed-loop system is rigorously proved with Lyapunov methods, such that the proposed controller is theoretically grounded even for high-speed applications. An experimental study is conducted with a medical robot system (the da Vinci Research Kit) to demonstrate the performance of the proposed method.

The rest part of this paper is organized as follows: Section II details the related background, Section III presents the novel control scheme, Section IV shows experimental results, and Section V concludes the paper.

II. Background

A. Robot Kinematics and Dynamics

Consider a vision-based robotic manipulation system shown in Fig. 1. A camera is employed to observe the feature point on the deformable object, and the robot end-effector is controlled to contact and manipulate the deformable object into a desired configuration. Let $\mathbf{r} = [r_1\ r_2\ r_3]^\intercal \in \mathbb{R}^3$ denote the position vector of the robot end-effector in task space, and $\mathbf{q} \in \mathbb{R}^n$ denote the robot joint positions. Taking the time derivate of $\mathbf{r}$, we can obtain a velocity mapping as follows:

$$\dot{\mathbf{r}} = \mathbf{J}_r(\mathbf{q})\dot{\mathbf{q}} \tag{1}$$

where $\mathbf{J}_r(\mathbf{q}) \in \mathbb{R}^3 \times \mathbb{R}^n$ is the Jacobian matrix from the joint space to the task space.

It is well known that the dynamic of a robot manipulator can be formulated in the following form:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \underbrace{\left(\frac{1}{2}\dot{\mathbf{H}}(\mathbf{q}) + \mathbf{S}(\mathbf{q},\dot{\mathbf{q}})\right)}_{\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{2}$$

where $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^n \times \mathbb{R}^n$ is called robot inertia matrix, which is symmetric positive definite, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ is a vector of centripetal and Coriolis forces, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ represents gravitational forces, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the control input, and $\boldsymbol{\tau}_e \in \mathbb{R}^n$ denote the interaction torque because of the physical interaction between the end effector and the object. Note that, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ can be further decomposed into two components, where $\mathbf{S}(\mathbf{q},\dot{\mathbf{q}})$ is skew-symmetric. In addition, $\boldsymbol{\tau}_e = \mathbf{J}_r^\intercal(\mathbf{q})\mathbf{f}_e$ where $\mathbf{f}_e \in \mathbb{R}^3$ is the interaction force exerted on the deformable object which is also bounded. Note that the force reduces to zero when the end effector does not contact the object.

In this paper, it is assumed that the dynamic parameters are unknown, which will be estimated online by exploring the following property.

Property 1. Given a proper definition of the unknown parameter vector $\boldsymbol{\theta}_d$ describing the physical properties, the terms $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$, and $\mathbf{g}(\mathbf{q})$, all depend linearly on $\boldsymbol{\theta}_d$, and can be linearly parameterized as follows [20]:

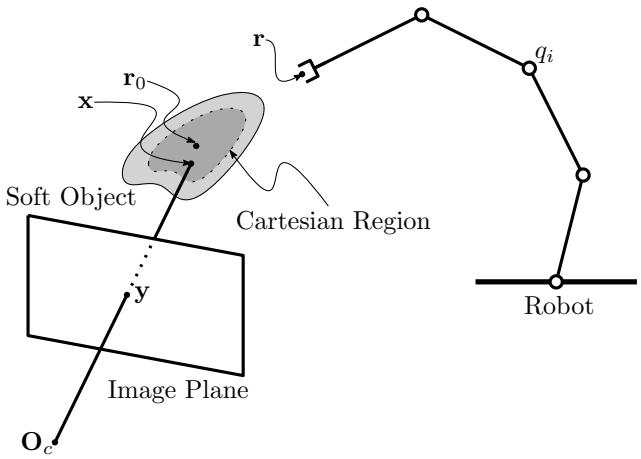

Fig. 1. Illustration shows the deformable object manipulation setup. The robot manipulator does not contact the object to be manipulated initially.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}) = \mathbf{Y}_d(\mathbf{q},\dot{\mathbf{q}},\dot{\mathbf{q}}_r,\ddot{\mathbf{q}}_r)\boldsymbol{\theta}_d \tag{3}$$

where $\mathbf{q}_r$ is called nominal reference.

B. Deformation Model and Camera Model

In this work, we model the deformable object as a homogeneous elastic body which has no rheological properties as [21]:

$$\mathbf{x} = \mathbf{D}\mathbf{r} + \mathbf{d} \tag{4}$$

where $\mathbf{x} = [x\ y\ z]^\intercal \in \mathbb{R}^3$ denote the position vector of a feature point on the surface of a deformable object in Cartesian-space, $\mathbf{D} \in \mathbb{R}^3 \times \mathbb{R}^3$ and $\mathbf{d} \in \mathbb{R}^3$ are constant matrix and vector, respectively, representing the object deformation properties.

To measure the deformation, we use a fixed camera to get the vision feedback without calibration (intrinsic and extrinsic) in an eye-to-hand manner. Let $\mathbf{y} = [u\ v]^\intercal \in \mathbb{R}^2$ denote the position vector of the feature point projection on the image plane. Under the perspective projection model [22], the projection of the feature point on to the image plane can be given as follows:

$$\begin{bmatrix}\mathbf{y}\\ 1\end{bmatrix} = \frac{1}{{}^c z}\mathbf{K}[\mathbf{R}\ \mathbf{t}]\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix} \tag{5}$$

where ${}^c z$ is the depth of the feature point with respect to the camera frame, $\mathbf{K} \in \mathbb{R}^3 \times \mathbb{R}^3$ is an upper triangular constant matrix associating with the camera intrinsic parameters, and $\mathbf{R} \in \mathbb{R}^3 \times \mathbb{R}^3$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation matrix and translation vector, respectively, representing the camera extrinsic parameters.

Multiplying ${}^c z$ on the both sides of (5), we can rewrite it as follows: $[{}^c z\mathbf{y}, {}^c z]^T = \mathbf{W}\mathbf{r} + \mathbf{M}\mathbf{d} + \mathbf{K}\mathbf{t}$, where $\mathbf{M} = \mathbf{K}\mathbf{R}$ and $\mathbf{W} = \mathbf{M}\mathbf{D} = \mathbf{K}\mathbf{R}\mathbf{D}$. Let $\mathbf{w}_i^\intercal$ denote the ith row vector of $\mathbf{W}$, $\mathbf{m}_i^\intercal$ denote the ith row vector of $\mathbf{M}$, and

t_z denote z component of the translation vector $\mathbf{t}$. Then ${}^c z$ can be computed as follows:

$$ {}^c z = \mathbf{w}_3^\intercal \mathbf{r} + \mathbf{m}_3^\intercal \mathbf{d} + t_z \tag{6} $$

By differentiating (5), we can obtain the following velocity mappings:

$$ \dot{\mathbf{y}} = \frac{1}{{}^c z} \mathbf{Q}_D \dot{\mathbf{q}} \tag{7a} $$

where $\mathbf{Q}_D \in \mathbb{R}^2 \times \mathbb{R}^n$ is called combined depth-independent image-deformation Jacobian matrix in the following form:

$$ \mathbf{Q}_D = \begin{bmatrix} \mathbf{w}_1^\intercal - u\mathbf{w}_3^\intercal \\ \mathbf{w}_2^\intercal - v\mathbf{w}_3^\intercal \end{bmatrix} \mathbf{J}_r(\mathbf{q}) \tag{8} $$

Therefore, the combined image-deformation Jacobian can be defined $\mathbf{J}_y = \frac{1}{{}^c z}\mathbf{Q}_D$.

The exact information about the deformation model is usually unknown and difficult to obtain, because it varies with different material, shape, and size of deformable objects. In addition, the exact camera model is obtained with identification or calibration techniques, which is laborious. Even the camera model can be obtained with sufficient accuracy, it is not flexible to re-calibrate the camera every time when it is adjusted to suit a new deformable object which is to be manipulated. Moreover, the unknown depth information cannot be updated together with other unknown parameters using adaptive control techniques [23], as it is inversely proportional to other parameters, thereby cannot be extracted to form a lumped vector.

In this paper, the unknown depth information and the unknown parameters of the camera model and deformation model will be estimated by exploring the following two properties, respectively.

Property 2. ${}^c z\dot{\mathbf{y}}$ is linear in sets of constant values such as robot DH parameters and camera intrinsic and extrinsic parameters, and it can be linearly parameterized as follows [24]:

$$ {}^c z\dot{\mathbf{y}} = \mathbf{Y}_z(\dot{\mathbf{y}}, \mathbf{r})\boldsymbol{\theta}_z \tag{9} $$

where $\dot{\mathbf{y}}$ is the velocity of the feature point projection, $\mathbf{Y}_z(\dot{\mathbf{y}}, \mathbf{r})$ is called the depth regressor matrix, and $\boldsymbol{\theta}_z$ represents the unknown depth parameters.

Property 3. For any 3×1 vector $\boldsymbol{\rho}$, the product $\mathbf{Q}_D\boldsymbol{\rho}$ can be be linearly parameterized in the following form:

$$ \mathbf{Q}_D\boldsymbol{\rho} = \mathbf{B}(\boldsymbol{\rho}, \mathbf{q}, \mathbf{y})\boldsymbol{\theta} \tag{10} $$

where $\mathbf{B}(\boldsymbol{\rho}, \mathbf{q}, \mathbf{y})$ is a regression matrix without depending on the robot kinematics parameters, the camera parameters (intrinsic and extrinsic), and the object deformation parameters, and $\boldsymbol{\theta}$ is a constant vector of the unknown parameters.

III. Adaptive Control for Contacting and Manipulation

In this section, a new adaptive controller is proposed to enable the robot to automatically contact then actively deform the deformable object into a desired configuration, together with novel adaption laws for updating the unknown parameters of the robot dynamic model, objection deformation model, and camera model concurrently and separately. Two types of feedback information (i.e. Cartesian-space and image-space feedback) are employed for the operations of contacting and manipulation respectively and also integrated into a single controller.

A. Regional Feedback

In this work, we make the use of the region reaching control technique [25] for regulating the robot end-effector to reach a specific region. The region is specified in Cartesian-space and also attached to the deformable object, such that the end effector contacts the object after it enters the Cartesian-space region. Let $\mathbf{r}_0 = [r_{01}\ r_{02}\ r_{03}]^\intercal$ denote the centroid of the region. Then the region-function is defined in the following form:

$$ f(\mathbf{r}) = \begin{bmatrix} (r_1 - r_{01})^{n_r} \\ (r_2 - r_{02})^{n_r} \\ (r_3 - r_{03})^{n_r} \end{bmatrix}^\intercal \begin{bmatrix} R_1^{-n_r} \\ R_2^{-n_r} \\ R_3^{-n_r} \end{bmatrix} - 1 \tag{11} $$

where $R_i, i \in \{1, 2, 3\}$ are positive constants representing the radiuses along three principal axes, respectively, and $n_r \geq 2$ is the order of the region which is also an even integer. As the order of the region increase, the region shape more looks like a cuboid with rounded corners (as shown in Fig. 2). Therefore, the shape, the size, and the position of the region can be changed to suit different deformable objects by varying the parameters of the region function.

Then, a potential energy function for the region reaching controller can be specified in the following form:

$$ P_r(\mathbf{r}) = \frac{k_{P_r}}{N} \left[\max(0, f(\mathbf{r}))\right]^N \tag{12} $$

where k_{P_r} is a positive constant, and $N \geq 4$ is an even integer, which is the order of the function.

Partial differentiation of the potential energy (12) with respect to $\mathbf{r}$ results in:

$$ \begin{aligned} \Delta\boldsymbol{\epsilon}_r(\mathbf{r}) &= \left(\frac{\partial P_r(\mathbf{r})}{\partial \mathbf{r}}\right)^\intercal \\ &= k_{P_r} \left[\max(0, f(\mathbf{r}))\right]^{N-1} \left(\frac{\partial f(\mathbf{r})}{\partial \mathbf{r}}\right)^\intercal \end{aligned} \tag{13} $$

which represents the Cartesian-space regional feedback. From (13), the region error $\Delta\boldsymbol{\epsilon}_r(\mathbf{r})$ is nonzero where $f(\mathbf{r}) > 0$ and hence drives the end effector to enter the region and also contact the deformable object, and it automatically reduces to zero where $f(\mathbf{r}) \leq 0$, that is, inside the region The Cartesian-space region error will be used to achieve the operation of contacting.

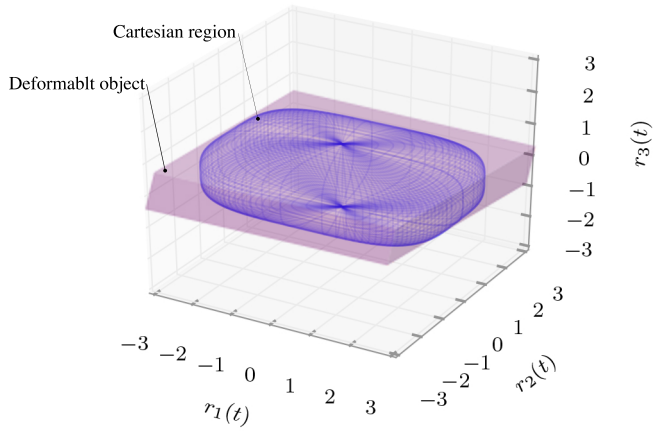

Fig. 2. A Cartesian region (blue mash) represents the volume that a deformable object (pink cuboid) occupies in the Cartesian-space.

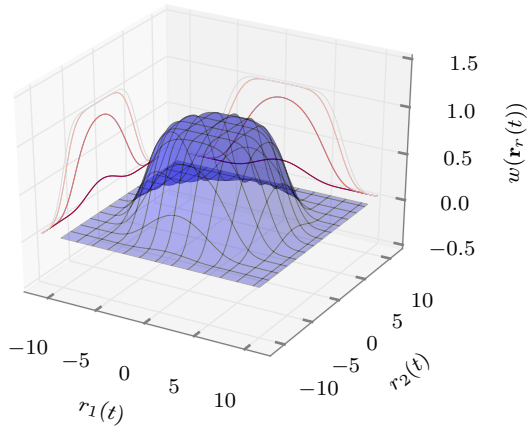

Fig. 3. A weighting factor for phase switch. The value is 0 when the end-effector is out of the region and is 1 when the end-effector is inside the region (contacts the object).

Next, the vision is only employed for the active deformation after the robot end effector stably contacts the deformable object. To monitor the position of the end effector with respect to the Cartesian-space region, a weighting factor is constructed as follows:

$$w_r(\mathbf{r}) = 1 - \frac{1}{(\kappa^{n_r}-1)^{16}} \cdot \left\{\min\left[0, [\min(0, f(\mathbf{r}))]^4 - (\kappa^{n_r}-1)^4\right]\right\}^4 \quad (14)$$

where $0 < \kappa < 1$ is a constant, which controls steepness of the function. That is, $w_r(\mathbf{r}) = 0$ when $f(\mathbf{r}) > 0$, and it smoothly increases to 1 when $f(\mathbf{r}) \leq 0$. A two dimensional visualization of the weighting factor is shown in Fig. 3.

By using the weighting factor, a composite vector is defined in the image-space $\mathbf{y}_c = w_r(\mathbf{r})\mathbf{y}_d + (1 - w_r(\mathbf{r}))\mathbf{y}$, where $\mathbf{y}_d$ is the desired (constant) position for the feature point on the image plane. Then the image-space regional feedback is specified as follows:

$$\Delta\boldsymbol{\epsilon}_y = \mathbf{y} - \mathbf{y}_c \quad (15)$$

From (14), when the end effector does not contact the deformable object, $w_r(\mathbf{r}) = 0$, $\mathbf{y}_c = \mathbf{y}$, and hence $\Delta\boldsymbol{\epsilon}_y = \mathbf{0}$, that is, the vision feedback is not used; when the end effector contacts the object, $w_r(\mathbf{r}) = 1$, $\mathbf{y}_c = \mathbf{y}_d$, and hence $\Delta\boldsymbol{\epsilon}_y \neq \mathbf{0}$, hence, the vision feedback is activated.

B. Adaptive Controller Design

With the regional feedback from both Cartesian-space and image-space, a unified controller is proposed to achieve the operations of contacting and active deformation simultaneously, in the presence of uncertain robot dynamics, unknown deformation model, and uncalibrated cameras.

The controller is proposed as follows:

$$\begin{aligned}\boldsymbol{\tau} = {} & \mathbf{Y}_d(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}}_d - k_g\mathbf{sgn}(\mathbf{s}) - \mathbf{K}_s\mathbf{s} \\ & - k_r\mathbf{J}_r^\intercal(\mathbf{q})\Delta\boldsymbol{\epsilon}_r(\mathbf{r}) - k_y\hat{\mathbf{J}}_y^\intercal(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_q)\Delta\boldsymbol{\epsilon}_y \end{aligned} \quad (16)$$

where $\hat{\boldsymbol{\theta}}_d$ is an estimation of unknown constant vector $\boldsymbol{\theta}_d$, and $\mathbf{s}$ is a sliding vector, which is introduced as follows:

$$\begin{aligned}\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = {} & \dot{\mathbf{q}} - \hat{\mathbf{J}}_y^+(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_q)\dot{\mathbf{y}}_c + \alpha_r\mathbf{J}_r^+(\mathbf{q})\Delta\boldsymbol{\epsilon}_r(\mathbf{r}) \\ & + \alpha_y\hat{\mathbf{J}}_y^+(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_q)\Delta\boldsymbol{\epsilon}_y \end{aligned} \quad (17)$$

and $\mathbf{sgn}(\cdot)$ is a sign function, and k_g is a positive constant. The term $-k_g\mathbf{sgn}(\mathbf{s})$ will be used to deal with the interaction force. $\hat{\mathbf{J}}_y^+(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_q)$ is the pseudo-inverse of the image-deformation Jacobian estimation $\hat{\mathbf{J}}_y(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_q)$, which can be constructed by the online updated vectors $\hat{\boldsymbol{\theta}}_z$ and $\hat{\boldsymbol{\theta}}_q$. Note that $\Delta\boldsymbol{\epsilon}_r$ and $\Delta\boldsymbol{\epsilon}_y$ are employed in the operations of contacting and manipulation respectively, and the image-space regional feedback is only activated when the robot end effector has stably contacted the deformable object. Whenever the contact is lost during the manipulation, the manipulation is suspended (i.e. $\Delta\boldsymbol{\epsilon}_y = \mathbf{0}$), and the Cartesian-space regional feedback is triggered (i.e. $\Delta\boldsymbol{\epsilon}_r \neq \mathbf{0}$) to enable the robot to re-contact the object again. While different feedback information is employed in the local region, the combination of regional feedback guarantee the autonomous contacting then manipulation in a unified controller. Since the weighting factor (14) is smooth, the transition between different feedback is also continuous without any hard switching.

From Property 2, the product of depth estimation error ${}^c\hat{z}\dot{\mathbf{y}} - {}^cz\dot{\mathbf{y}}$ and the image velocity $\dot{\mathbf{y}}$ can be linearized in the following form:

$${}^c\hat{z}\dot{\mathbf{y}} - {}^cz\dot{\mathbf{y}} = \mathbf{Y}_z(\dot{\mathbf{y}}, \mathbf{r})\Delta\boldsymbol{\theta}_z \quad (18)$$

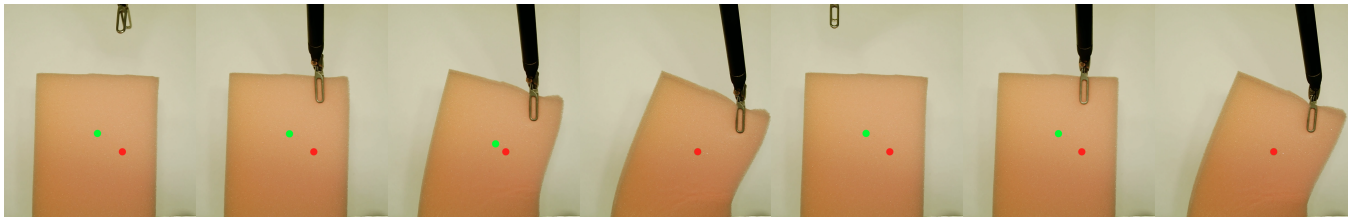

Fig. 4. Sequence of snapshots during two experiment scenarios. From $it = 60$ to $it = 99$, the robot was manually pushed away from the phantom. The green point represents the tracked point and the red point denotes the desired position. (it denotes iteration)

TABLE I
Values and Status of the Errors and Adaption Laws

Operation	Feedback	w_r	$\Delta\boldsymbol{\epsilon}_r$	$\Delta\boldsymbol{\epsilon}_y$	$\hat{\boldsymbol{\theta}}_z$ & $\hat{\boldsymbol{\theta}}_q$
Contact	Cartesian	0	-	0	Suspended
Manipulation	Image	1	0	-	Updating

where $\Delta\boldsymbol{\theta}_z = \hat{\boldsymbol{\theta}}_z - \boldsymbol{\theta}_z$.

Similarly, from Property 3, the product of the combined depth-independent image-deformation Jacobian matrix estimation error and the joint velocity can be linearized as:

$$\hat{\mathbf{Q}}_D\dot{\mathbf{q}} - \mathbf{Q}_D\dot{\mathbf{q}} = \mathbf{Y}_q(\dot{\mathbf{q}}, \mathbf{q}, \mathbf{y})\Delta\boldsymbol{\theta}_q \tag{19}$$

where $\Delta\boldsymbol{\theta}_q = \hat{\boldsymbol{\theta}}_q - \boldsymbol{\theta}_q$.

Then, the uncertain parameters $\hat{\boldsymbol{\theta}}_d$, $\hat{\boldsymbol{\theta}}_z$, and $\hat{\boldsymbol{\theta}}_q$ can be updated as:

$$\frac{d}{dt}\hat{\boldsymbol{\theta}}_d = -\boldsymbol{\Gamma}_d\mathbf{Y}_d^\intercal(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{s} \tag{20a}$$

$$\frac{d}{dt}\hat{\boldsymbol{\theta}}_z = -\hat{Z}\boldsymbol{\Gamma}_z\mathbf{Y}_z^\intercal(\dot{\mathbf{y}}, \mathbf{r}(r))\Delta\boldsymbol{\epsilon}_y \tag{20b}$$

$$\frac{d}{dt}\hat{\boldsymbol{\theta}}_q = \hat{Z}\boldsymbol{\Gamma}_q\mathbf{Y}_q^\intercal(\dot{\mathbf{q}}, \mathbf{q}, \mathbf{y})\Delta\boldsymbol{\epsilon}_y \tag{20c}$$

where $\hat{Z} = {k_y}^c\hat{z}^{-1}$. $\boldsymbol{\Gamma}_d$, $\boldsymbol{\Gamma}_z$, and $\boldsymbol{\Gamma}_q$ are positive-definite and diagonal gain matrices.

In particular, (20a)-(20c) are developed to update the dynamic parameters, the depth parameters, and the deformation parameters, respectively. When the end effector does not contact the deformable object, $\Delta\boldsymbol{\epsilon}_y = \mathbf{0}$, and hence both $\frac{d}{dt}\hat{\boldsymbol{\theta}}_z$ and $\frac{d}{dt}\hat{\boldsymbol{\theta}}_q$ are equal to zero. That is, the updating for the estimated Jacobian matrix $\hat{\mathbf{J}}_y(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_y)$ is suspended. It is reasonable since both the vision feedback and the estimated Jacobian matrix $\hat{\mathbf{J}}_y(\hat{\boldsymbol{\theta}}_z, \hat{\boldsymbol{\theta}}_y)$ are only used in the phase of manipulation, after the end effector stably contacts the deformable object. The variations of control terms in different operations are summarized in TABLE I.

Theorem 1. For the dynamics (2), the controller (16) together with adaptation laws (20a)-(20c) realizes the contact and active deformable of deformable objects, i.e., the asymptotic stability of both the Cartesian-space region error and the image-space region error is guaranteed ($\Delta\boldsymbol{\epsilon}_r \to \mathbf{0}$, $\Delta\boldsymbol{\epsilon}_y \to \mathbf{0}$ as $t \to \infty$).

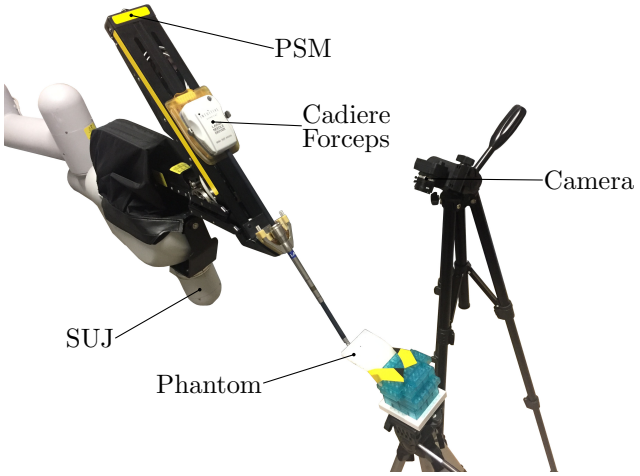

Fig. 5. The experimental setup consists of a PSM with Cardiere Forceps installed, a camera (uncalibrated), and a piece of phantom (unknown physical characteristics).

The proof is given in Appendix I.

IV. Experimental Studies

A. Experimental Setup

To validate the proposed control method, we used the da Vinci Research Kit (dVRK) as our experimental platform (as shown in Fig. 5). The dVRK is a tele-surgical research platform consisting of the mechanical components donated by Intuitive Surgical Inc. and the open source electronics and software libraries (cisst/SAW) developed by researchers in Johns Hopkins University [26]. The robot has two Patient Side Manipulators (PSM), two Master Tool Manipulators (MTM), and one Endoscope Camera Manipulator. Since the dVRK in the Chinese University of Hong Kong uses a retired da Vinci Classic System as its mechanical components, it also has Setup Joints (SUJ) for holding PSMs and the ECM.

In this work, we conducted the experimental studies using a single PSM with a Cadiere Forceps (Intuitive Surgical Inc.) installed. During the experiment, the vision feedback is provided by a Point Grey Firefly camera (FMVU-03MTC-CS). The deformable object is simulated by a piece of sponge. The phantom has the shape of a cuboid (95mm×65mm×10mm), and is flexible with homogeneous mechanical properties. To track the position of the feature point, we implemented

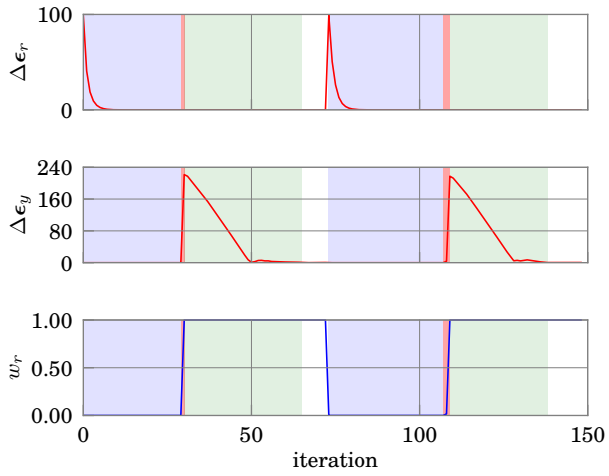

Fig. 6. The experimental results. Light blue region represents the region reaching phase, light green represents active deforming phase, and light red represents transition phase.

Lucas-Kanade optical flow algorithm using the OpenCV library [27]. The experimental software is written in Python using rospy as a ROS package [28]. Further, to mitigate noisy image velocity calculated by the numeric differentiation, we implemented a first low-pass filter.

B. Experimental Procedure

We used a tripod to hold the phantom, whose position is known with respect to the base frame of PSM, and defined the Cartesian region by taking it as a reference. We selected a feature point to be tracked and marked the a desired point on the image plane that needs to be coincided. The snapshots of the experiment are shown in Fig. 4. We first ran the algorithm to let the robot contact then manipulate the phantom to the desired pose, i.e., overlaying the green point onto the red point. Then, we interrupted the the movement by manually pushing the robot away from its desired pose. Then, we released the robot and let it to re-contact and re-manipulate the deformable object.

C. Experimental Results

The results of the experimental study is shown in Fig. 6. The whole experimental study can be divided into two parts. In the first part, the robot first moved towards to the object (from $it = 0$ to $it = 30$). Once the robot end-effector entered the Cartesian region and contacted the object, the Cartesian-space regional error $\Delta\boldsymbol{\epsilon}_r$ converged to 0 and the value of the weighting factor increased from 0 to 1. Meanwhile, the image-space regional error was activated and the robot started to actively deform the object using the vision feedback. The image-space regional error $\Delta\boldsymbol{\epsilon}_y$ converged to 0 at $it = 66$. Then, at $it = 73$, we interrupted the active deformation by manually moving the end-effector away from the object, so that the value of the weighting factor dropped from 1 to 0. Once we released the robot, it moved towards to the object again to re-contact then re-manipulate the object. The Cartesian-space regional error $\Delta\boldsymbol{\epsilon}_r$ and the image-space regional error $\Delta\boldsymbol{\epsilon}_y$ converged to 0 again at $it = 110$ and $it = 139$, respectively.

V. Conclusion

In this paper, a new control scheme has been developed for robotic manipulation of deformable objects, which allows the robot to automatically contact then manipulate the deformable object by assessing the status of deformation in real time. Instead of designing multiple controllers and switching between them, the proposed method smoothly and stably integrate two control phases (i.e. region reaching and active deforming) into a single controller. The stability of the closed-loop system is rigorously proved with the consideration of uncertain robot dynamics, unknown deformation model, and also uncalibrated cameras. Hence, the proposed control scheme guarantees the autonomous capability, the feasibility, and the stability of active deformation of deformable objects. Experimental results in a medical robot system (da Vinci Research Kit) have been presented to illustrate the performance of the proposed controller. The proposed method provides a solution to robotic manipulation of deformable objects and can be easily extended to many other growing field such as food processing, 3C manufacturing, and automatic laundry.

Appendix I
Stability Analysis

Injecting Eq. (16) into (2) and considering Property 1, we obtain the closed dynamics in the following form:

$$\begin{aligned}\mathbf{H}\dot{\mathbf{s}} + \mathbf{C}\mathbf{s} = \boldsymbol{\tau}_e + \mathbf{Y}_d\Delta\boldsymbol{\theta}_d - k_g\mathbf{sgn}(\mathbf{s}) - \mathbf{K}_s\mathbf{s} \\ - k_r\mathbf{J}_r^\intercal\Delta\boldsymbol{\epsilon}_r - k_y\hat{\mathbf{J}}_y^\intercal\Delta\boldsymbol{\epsilon}_y\end{aligned} \tag{21}$$

where $\Delta\boldsymbol{\theta}_d = \hat{\boldsymbol{\theta}}_d - \boldsymbol{\theta}_d$.

Introduce the following positive-definite function:

$$\begin{aligned}V = \frac{1}{2}\mathbf{s}^\intercal\mathbf{H}\mathbf{s} + k_rP_r(\mathbf{r}) + \frac{1}{2}k_y\Delta\boldsymbol{\epsilon}_y^\intercal\Delta\boldsymbol{\epsilon}_y \\ + \frac{1}{2}\Delta\boldsymbol{\theta}_d^\intercal\boldsymbol{\Gamma}_d^{-1}\Delta\boldsymbol{\theta}_d + \frac{1}{2}\Delta\boldsymbol{\theta}_z^\intercal\boldsymbol{\Gamma}_z^{-1}\Delta\boldsymbol{\theta}_z + \frac{1}{2}\Delta\boldsymbol{\theta}_q^\intercal\boldsymbol{\Gamma}_q^{-1}\Delta\boldsymbol{\theta}_q\end{aligned} \tag{22}$$

Differentiating V and substituting (13), (15), (20a), (20b), (20c) into $\dot{V}$ result in:

$$\begin{aligned}\dot{V} = \mathbf{s}^\intercal\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^\intercal\dot{\mathbf{H}}\mathbf{s} + k_r\Delta\boldsymbol{\epsilon}_r^\intercal\dot{\mathbf{r}} + k_y\Delta\boldsymbol{\epsilon}_y^\intercal(\dot{\mathbf{y}} - \dot{\mathbf{y}}_c) \\ - \Delta\boldsymbol{\theta}_d^\intercal\boldsymbol{\Gamma}_d^{-1}\boldsymbol{\Gamma}_d\mathbf{Y}_d^\intercal\mathbf{s} + \Delta\boldsymbol{\theta}_q^\intercal\boldsymbol{\Gamma}_q^{-1}\hat{Z}\boldsymbol{\Gamma}_q\mathbf{Y}_q^\intercal\Delta\boldsymbol{\epsilon}_y \\ - \Delta\boldsymbol{\theta}_z^\intercal\boldsymbol{\Gamma}_z^{-1}\hat{Z}\boldsymbol{\Gamma}_z\mathbf{Y}_z^\intercal\Delta\boldsymbol{\epsilon}_y\end{aligned} \tag{23}$$

The first two terms in $\dot{V}$ can be rewritten as follows:

$$\begin{aligned}
&\mathbf{s}^\intercal\mathbf{H}\dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^\intercal\dot{\mathbf{H}}\mathbf{s} = \mathbf{s}^\intercal\left\{\mathbf{H}\dot{\mathbf{s}} + \mathbf{C}\mathbf{s}\right\} \\
&= \mathbf{s}^\intercal\mathbf{Y}_d\Delta\boldsymbol{\theta}_d + \mathbf{s}^\intercal\boldsymbol{\tau}_e - k_g\mathbf{s}^\intercal\mathbf{sgn}(\mathbf{s}) - \mathbf{s}^\intercal\mathbf{K}_s\mathbf{s} \\
&\quad - (\dot{\mathbf{q}} - \hat{\mathbf{J}}_y^+\dot{\mathbf{y}}_c)^\intercal \cdot \left(k_r\mathbf{J}_r^\intercal\Delta\boldsymbol{\epsilon}_r + k_y\hat{\mathbf{J}}_y^\intercal\Delta\boldsymbol{\epsilon}_y\right) \\
&\quad - \left(\alpha_r\mathbf{J}_r^+\Delta\boldsymbol{\epsilon}_r + \alpha_y\hat{\mathbf{J}}_y^+\Delta\boldsymbol{\epsilon}_y\right)^\intercal \cdot \left(k_r\mathbf{J}_r^\intercal\Delta\boldsymbol{\epsilon}_r + k_y\hat{\mathbf{J}}_y^\intercal\Delta\boldsymbol{\epsilon}_y\right)
\end{aligned} \tag{24}$$

The above coupling terms are zeros all the time since $\Delta\boldsymbol{\epsilon}_r$ is nonzero only when $f(\mathbf{r}) \geq 0$ ($w(\mathbf{r}) = 0$, i.e., outside the region), where $\Delta\boldsymbol{\epsilon}_y = \mathbf{y} - \mathbf{y}_c = \mathbf{y} - (w(\mathbf{r})\mathbf{y}_d + (1 - w(\mathbf{r}))\mathbf{y}) = \mathbf{y} - \mathbf{y} = 0$.

$$\left(\alpha_r\mathbf{J}_r^+\Delta\boldsymbol{\epsilon}_r\right)^\intercal \cdot \left(k_y\hat{\mathbf{J}}_y^\intercal\Delta\boldsymbol{\epsilon}_y\right) = 0 \tag{25a}$$

$$\left(\alpha_y\hat{\mathbf{J}}_y^+\Delta\boldsymbol{\epsilon}_y\right)^\intercal \cdot (k_r\mathbf{J}_r^\intercal\Delta\boldsymbol{\epsilon}_r) = 0 \tag{25b}$$

The term $(\hat{\mathbf{J}}_y^+\dot{\mathbf{y}}_c)^\intercal \cdot (k_r\mathbf{J}_r^\intercal\Delta\boldsymbol{\epsilon}_r)$ is also zero, since when $\Delta\boldsymbol{\epsilon}_r \neq \mathbf{0}$ ($f(\mathbf{r}) \geq 0$ or $w(\mathbf{r}) = 0$), the robot end-effector hasn't contacted the deformable object, or just contacted the object without significant deformation, and thus it is reasonable to assume that the velocity of the feature point on the deformable object is zero, i.e. $\dot{\mathbf{y}} = \mathbf{0}$. Then, $\dot{\mathbf{y}}_c = \dot{\mathbf{y}} = \mathbf{0}$.

In $\dot{V}$, after substituting the closed-loop equation, we will obtain the term $\mathbf{s}^\intercal\boldsymbol{\tau}_e - k_g\mathbf{s}^\intercal\mathbf{sgn}(\mathbf{s})$ where $\boldsymbol{\tau}_e$ is bounded. It is assumed that the upper bound of $\boldsymbol{\tau}_e$ is b_e, then we have $\mathbf{s}^\intercal\boldsymbol{\tau}_e - k_g\mathbf{s}^\intercal\mathbf{sgn}(\mathbf{s}) \leq -(k_g - b_e)\|\boldsymbol{s}\|$. Therefore, if k_g is chosen sufficiently large such that $k_g > b_e$, we have the term $\mathbf{s}^\intercal\boldsymbol{\tau}_e - k_g\mathbf{s}^\intercal\mathbf{sgn}(\mathbf{s}) \leq 0$.

Considering all above, it is obtained that:

$$\begin{aligned}
\dot{V} &= \mathbf{s}^\intercal\boldsymbol{\tau}_e - k_g\mathbf{s}^\intercal\mathbf{sgn}(\mathbf{s}) - \mathbf{s}^\intercal\mathbf{K}_s\mathbf{s} \\
&\quad - k_r\alpha_r\Delta\boldsymbol{\epsilon}_r^\intercal(\mathbf{r})\Delta\boldsymbol{\epsilon}_r - k_y\alpha_y\Delta\boldsymbol{\epsilon}_y^\intercal\Delta\boldsymbol{\epsilon}_y \\
&\leq -\mathbf{s}^\intercal\mathbf{K}_s\mathbf{s} - k_r\alpha_r\Delta\boldsymbol{\epsilon}_r^\intercal(\mathbf{r})\Delta\boldsymbol{\epsilon}_r - k_y\alpha_y\Delta\boldsymbol{\epsilon}_y^\intercal\Delta\boldsymbol{\epsilon}_y \\
&\leq 0
\end{aligned} \tag{26}$$

Since from (22), V is positive definite, and from (26), $\dot{V}$ is negative semi-definite, V is bounded, such that $\mathbf{s}$, P_r, $\Delta\boldsymbol{\epsilon}_y$, $\Delta\boldsymbol{\theta}_d$, $\Delta\boldsymbol{\theta}_q$, $\Delta\boldsymbol{\theta}_z$ are bounded. The boundedness of P_r ensures the boundedness of f. Therefore, $\mathbf{r}$ is also bounded. Since f is specified as scalar functions with continuous partial derivative, the boundedness of $\mathbf{r}$ ensures the boundedness of the partial derivative. Since both f and its partial derivative are bounded, the region error $\Delta\boldsymbol{\epsilon}_r$ and its partial derivative with respect to $\mathbf{r}$ are bounded, as observed from (12). The boundedness of $\Delta\boldsymbol{\theta}_d$, $\Delta\boldsymbol{\theta}_q$, and $\Delta\boldsymbol{\theta}_z$ indicate that $\hat{\boldsymbol{\theta}}_d$, $\hat{\boldsymbol{\theta}}_q$, and $\hat{\boldsymbol{\theta}}_z$ are bounded.

Because w_r is bounded ($0 \leq w_r \leq 1$) and $\Delta\boldsymbol{\epsilon}_y$ is bounded, $\mathbf{y}$ is bounded. Since $w_r \in \mathcal{C}^2$, $\frac{\partial w_r}{\partial \mathbf{r}}$ is bounded. When $0 < w_r < 1$, the interactive force between the robot end-effector and the deformable tissue can be written as $\mathbf{D}_*\dot{\mathbf{r}} + \mathbf{K}_*\mathbf{r} = \mathbf{f}_e$, in which $\mathbf{D}_*$ and $\mathbf{K}_*$ are constants and $\mathbf{f}_e$ is bounded such that $\dot{\mathbf{r}}$ is bounded. The bounded $\frac{\partial w_r}{\partial \mathbf{r}}$ and $\dot{\mathbf{r}}$ indicate the boundedness of $\dot{w}_r$.

When $w_r = 0$, the robot end-effector does not contact the object yet so that $\dot{\mathbf{y}} = 0$ and hence $\dot{\mathbf{y}}_c = -\dot{\mathbf{y}} = 0$ is bounded. When $w_r = 1$, $\dot{\mathbf{y}}_c = 0$ is bounded. When $0 < w_r < 1$, the robot end-effector just contacts the object without significant deformation, and thus it is reasonable to assume that the $\dot{\mathbf{y}} = 0$ is bounded. Therefore, $\dot{\mathbf{y}}_c$ is bounded. The bounded $\hat{\boldsymbol{\theta}}_q$ and $\hat{\boldsymbol{\theta}}_z$ indicate the boundedness of $\hat{\mathbf{J}}_y^+$. $\mathbf{J}_r^+$ is also bounded. Therefore, $\dot{\mathbf{q}}_r$ is bounded, together with the bounded $\mathbf{s}$, we can conclude the boundedness of $\dot{\mathbf{q}}$. The bounded $\dot{\mathbf{q}}$ guarantees that $\dot{\mathbf{r}}$ and $\dot{\mathbf{y}}$ are bounded.

As mentioned above, $\frac{\partial \Delta\boldsymbol{\epsilon}_r}{\partial \mathbf{r}}$ is bounded and $\dot{\mathbf{r}}$ is bounded, thus $\Delta\dot{\boldsymbol{\epsilon}}_r$ is bounded. The boundedness of $\dot{\mathbf{y}}_c$ and $\dot{\mathbf{y}}$ ensure the boundedness of $\Delta\dot{\boldsymbol{\epsilon}}_y$. Therefore, $\Delta\boldsymbol{\epsilon}_r$ and $\Delta\boldsymbol{\epsilon}_y$ are uniformly continuous. From (26), we have $\Delta\boldsymbol{\epsilon}_r \in L_2$ and $\Delta\boldsymbol{\epsilon}_y \in L_2$; from [29], it is shown that $\Delta\boldsymbol{\epsilon}_r \to 0$ and $\Delta\boldsymbol{\epsilon}_y \to 0$, which means the robot end-effector moves into the Cartesian-space region and the position of the feature point moves to the desired set point.

References

[1] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in IEEE Int. Conf. Robot. Autom., vol. 1, 2000, pp. 255–262.

[2] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," Int. J. Rob. Res., vol. 14, no. 1, pp. 19–36, 1995.

[3] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," Int. J. Rob. Res., vol. 21, no. 3, pp. 233–255, 2002.

[4] S. Part, "Impedance control: An approach to manipulation," J. Dyn. Syst. Meas. Contr., vol. 107, p. 17, 1985.

[5] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, A mathematical introduction to robotic manipulation. CRC press, 1994.

[6] 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, 2011.

[7] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," Int. J. Rob. Res., vol. 31, no. 2, pp. 249–267, 2012.

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

[9] R. H. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery," IEEE Trans. Robot. Autom., vol. 19, no. 5, pp. 765–781, 2003.

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

[11] 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, 2000.

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

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

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

[15] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2013, pp. 4525–4532.
[16] 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. Rob., vol. 29, no. 6, pp. 1457–1468, 2013.
[17] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in IEEE Int. Conf. Robot. Autom., May 2017, pp. 2146–2153.
[18] D. Navarro-Alarcon, Y.-h. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," Int. J. Rob. Res., vol. 33, no. 11, pp. 1462–1480, 2014.
[19] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," IEEE Trans. Rob., vol. 32, no. 2, pp. 429–441, 2016.
[20] J.-J. E. Slotine and W. Li, Applied nonlinear control. Prentice-Hall, Inc., 1991.
[21] R. Ogden, "Non-linear elastic deformations," Eng. Anal. Bound. Elem., vol. 1, no. 2, p. 119, 1984.
[22] D. Forsyth and J. Ponce, Computer Vision: A Modern Approach, 2002.
[23] X. Li and C. C. Cheah, "Global task-space adaptive control of robot," Automatica, vol. 49, no. 1, pp. 58–69, 2013.
[24] C. C. Cheah and X. Li, Task-Space Sensory Feedback Control of Robot Manipulators, 2015, vol. 73.
[25] C.-C. Cheah, D. Q. Wang, and Y. C. Sun, "Region-reaching control of robots," IEEE Trans. Rob., vol. 23, no. 6, pp. 1260–1264, 2007.
[26] P. Kazanzidesf, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci surgical system," in IEEE Int. Conf. Robot. Autom., 2014, pp. 6434–6439.
[27] G. Bradski et al., "The opencv library," Doctor Dobbs Journal, vol. 25, no. 11, pp. 120–126, 2000.
[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in IEEE Int. Conf. Robot. Autom. Open-Source Software Workshop, vol. 3, no. 3.2, 2009, p. 5.
[29] S. Arimoto, Control theory of nonlinear mechanical systems. Oxford University Press, Inc., 1996.