

# Contour Moments Based Manipulation of Composite Rigid-Deformable Objects With Finite Time Model Estimation and Shape/Position Control

Jiaming Qi, Guangfu Ma , Jihong Zhu , Member, IEEE, Peng Zhou , Graduate Student Member, IEEE, Yueyong Lyu, Haibo Zhang, and David Navarro-Alarcon , Senior Member, IEEE

**Abstract**—The robotic manipulation of composite rigid-deformable objects (i.e., those with mixed nonhomogeneous stiffness properties) is a challenging problem with clear practical applications that, despite the recent progress in the field, it has not been sufficiently studied in the literature. To deal with this issue, in this article, we propose a new visual servoing method that has the capability to manipulate this broad class of objects (which varies from soft to rigid) with the same adaptive strategy. To quantify the object's infinite-dimensional configuration, our new approach computes a compact feedback vector of 2-D contour moments features. A sliding mode control scheme is then designed to simultaneously ensure the finite-time convergence of both the feedback shape error and the model estimation error. The stability of the proposed framework (including the boundedness of all the signals) is rigorously proved with Lyapunov theory. Detailed simulations and experiments are presented to validate the effectiveness of the proposed approach. To the best of the author's knowledge, this is the first time that contour moments along with finite-time control have been used to solve this difficult manipulation problem.

**Index Terms**—Contour moments, deformable objects, robotics, sliding mode control (SMC), visual-servoing.

## I. INTRODUCTION

THE manipulation of composite rigid-deformable objects (CRDO) is currently an open research problem in robotics (see our recent survey [1]). The ubiquitous nature of CRDO motivates the development of suitable manipulation strategies, which can be used in various application scenarios, e.g., food industry [2], robot-surgery assistance [3], cable assembly [4], and household works [5]. To automate these types of tasks, there are four main technical problems: 1) Efficient feedback shape representations; 2) Estimation of the sensorimotor model of the robot-object system. 3) Design of shaping controls that can timely minimize the deformation error. 4) Adjustment of dynamic performance that can improve the manipulation process. Although great progress has been achieved in this problem in recent years, the research of control methods for CRDO has not been sufficiently studied in the literature and validated experimentally. Our aim in this work is to provide a unified solution to the above issues.

Manuscript received June 4, 2021; revised September 24, 2021; accepted October 25, 2021. Recommended by Technical Editor H. Wang and Senior Editor G. Alici. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B 090928001, in part by the Hong Kong Research Grants Council under Grant 14203917, and in part by the PROCORE France/Hong Kong Joint Research Scheme sponsored by the RGC and the Consulate General of France in Hong Kong under Grant F-PolyU503/18. (Corresponding author: David Navarro-Alarcon.)

Jiaming Qi, Guangfu Ma, and Yueyong Lyu are with the Astronautics School, Harbin Institute of Technology, Harbin 150001, China (e-mail: qjim\_hit@163.com; magf@hit.edu.cn; lyvy@hit.edu.cn).

Jihong Zhu is with the Cognitive Robotics, Delft University of Technology (TU Delft), 2628 CD Delft, Netherlands (e-mail: j.zhu-3@tudelft.nl).

Haibo Zhang is with the Beijing Institute of Control Engineering, Beijing 100094, China (e-mail: zhanghb502@163.com).

Peng Zhou and David Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: jeffery.zhou@connect.polyu.hk; david.navarro-alarcon@polyu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TMECH.2021.3126383>.

Digital Object Identifier 10.1109/TMECH.2021.3126383

1083-4435 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

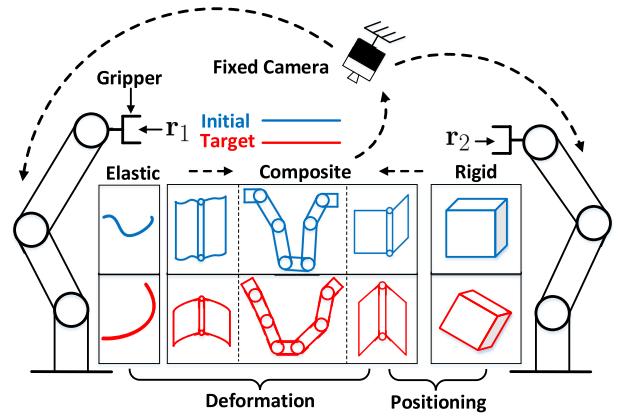
visual contours by replacing the region integral with a curvilinear integral, which results in a reduced computation cost. Note that this promising approach has not yet been used in shape servoing.

To automatically manipulate CRDO, it is essential to know how the robot's motion result in changes of the object's shape; This relation is captured by the so-called deformation Jacobian matrix (DJM) [1]. Due to the complexity of CRDO, the exact (i.e., analytical) calculation of the DJM might be difficult to obtain, a situation that is further complicated by uncertainties in the object's mechanical properties. Therefore, most works make use of numerical algorithms to estimate DJM in real-time. Some examples of this approach include the Broyden update rules [13], online optimization methods [14], flow-based adaptive algorithms [15], deep neural networks [16], etc. Note that these previous methods have been used to estimate the DJM of objects with "mostly" elastic properties. In [17], a shape servoing controller is proposed for both rigid and deformable objects that estimates the DJM with a least-squares method over a sliding window. However, in this and most previous works, the stability of such online algorithms is not rigorously proved (the design of their update rules is generally decoupled from the motion control laws). This condition limits the robustness of existing methods and cannot guarantee the approximation accuracy of DJM.

The design of standard visual servoing methods (including shape servoing ones) typically relies on imposing an asymptotically stable equilibrium of the visual-guided task [18] [19]. However, this closed-loop stability property does not ensure that the feedback shape error will converge to zero in a finite amount of time. This is undesirable for soft object manipulation tasks that demand precisely shaping actions with a deterministic convergence (e.g., in industrial productions). Combining finite-time control with sliding mode control (SMC) [20] is a feasible solution to deal with these issues, as it enables to specify the convergence of the error and is robust to model and parameter uncertainty [21]. Despite its valuable (and controllable) stability properties, this advanced control technique has not yet been utilized in soft object manipulation tasks [1].

In this article, we present an efficient shape servoing method to manipulate CRDO with a dual-arm robot. For that, we propose an innovative finite-time control scheme that ensures the robust minimization of both the shape error (which evaluate the manipulation task) and the model estimation error (which represents the estimation of the DJM) with a specified dynamic adjustment. A self-tuning adaptive update law is also presented to improve the approximation accuracy of DJM, which combined with classical Slotine-Li's  $\sigma$ -modification method [22] to avoid parameter drift. Although SMC has been previously used in visual servoing tasks [23][24], this is the first time (to the best of the authors' knowledge) that it has been used in a soft object manipulation task. The main contributions of this article are threefold:

- 1) Contour moments are used (for the first time) as the descriptor to represent the shape of CRDO (elastic, anisotropic, and time-varying physical properties).
- 2) A finite-time SMC (FTSMC) method is proposed (and rigorously analysed) to simultaneously perform the shape servoing task and to estimate the DJM.



**Fig. 1.** Representation of dual-arm robot manipulating CRDO. In general, the manipulation tasks includes deformation tasks and positioning tasks. The goal is to design the velocity controller to command dual-arm robot to manipulate CRDO into the desired configurations.

- 3) Detailed simulations, experiments and quantitative comparisons are conducted to validate our new method.

## II. PROBLEM FORMULATION

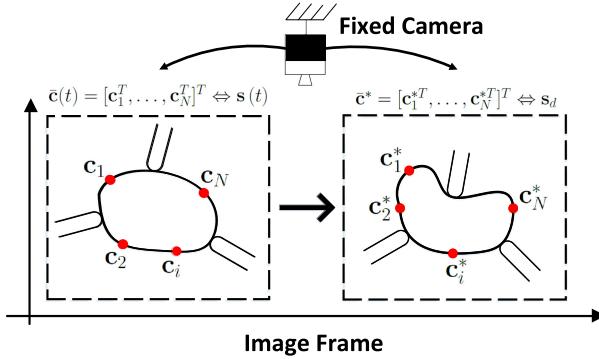
In this work, the vision-based manipulation task considered in the work is conceptually depicted in Fig. 1. Column vectors are denoted with bold small letters  $\mathbf{v}$ , and matrices with bold capital letters  $\mathbf{M}$ ; The symbol  $(\cdot)^+$  represents the matrix pseudoinverse.

### A. Dual-Arm Robot Model

Consider the configuration of a dual-arm robot, where the vector of joint angles and end-effector pose of each robot are denoted by  $\mathbf{q}_i \in \mathbb{R}^d$  and  $\mathbf{r}_i = \mathbf{r}_i(\mathbf{q}_i) \in \mathbb{R}^q$ , respectively, where  $i = 1, 2$  and  $q \leq d$ . We define the augmented vectors  $\mathbf{r} = [\mathbf{r}_1^T, \mathbf{r}_2^T]^T \in \mathbb{R}^{2q}$  and  $\mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T]^T \in \mathbb{R}^{2d}$ . The differential kinematics of the dual-arm robot are  $\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ , where  $\mathbf{J}(\mathbf{q}) = \text{diag}\{\frac{\partial \mathbf{r}_1}{\partial \mathbf{q}_1}, \frac{\partial \mathbf{r}_2}{\partial \mathbf{q}_2}\} \in \mathbb{R}^{2q \times 2d}$  represents the standard Jacobian matrix, which is assumed to be exactly known. In this article, we consider that the robots are kinematically controlled (i.e. that the robot's joint positions/velocities can be exactly determined). Furthermore, we assume that the trajectories of the robot are free from collisions with either the environment or among its arms.

### B. Visual-Deformation Model

Let us define the state of the CRDO as  $\mathbf{m} \in \mathbb{R}^\theta$ . The relation between the robot's pose  $\mathbf{r}$  and  $\mathbf{m}$  is represented by the (unknown) nonlinear mapping  $\mathbf{m} = \mathbf{f}_m(\mathbf{r})$ , which models the mechanical properties of the CRDO, e.g., deformable structures, rigid parts, and composites of both, etc. In this article, the object's image contour  $\bar{\mathbf{c}} = [\mathbf{c}_1^T, \dots, \mathbf{c}_N^T]^T \in \mathbb{R}^{2N}$ , for  $\mathbf{c}_i = [u_i, v_i]^T \in \mathbb{R}^2$ , is utilized to represent the configuration of the CRDO, where  $N$  represents the number of points that comprise the contour, and  $u_i$  and  $v_i$  represent the pixel coordinates of the  $i$ th ( $i = 1, \dots, N$ ) point in the image frame, see Fig. 2. Thus, the relation between  $\bar{\mathbf{c}}$  and  $\mathbf{m}$  can be described by the



**Fig. 2.** Contour description in the 2-D manipulation space. The contour is defined in the image frame, where  $\bar{c}$  represents the real-time feedback contour and  $\bar{c}^*$  represents the desired contour.

nonlinear mapping,  $\bar{c} = f_c(\mathbf{m})$  (which captures the perspective geometry of the object's points in the boundary).

To actively control the shape of CRDOs, the robot's end-effectors must rigidly grasp the object in advance. Motion of the controllable grippers' pose  $\mathbf{r}$  result in changes in the object's image contour  $\bar{c}$ . We model this visual-motor relation with the following expression:

$$\bar{c} = f_c(f_m(\mathbf{r})) = f_c(f_m(\mathbf{r}_1, \mathbf{r}_2)). \quad (1)$$

Note that the dimension  $2N$  of the observed contour  $\bar{c}$  is generally large, thus, it is inefficient to directly use it in a shape controller as it contains redundant information. In our approach, we use the contour information  $\bar{c}$  to construct a compact feature vector, here denoted by  $\mathbf{s} \in \mathbb{R}^p$ , for  $p \ll 2N$  that characterizes the object's shape but with significantly fewer feedback coordinates. The relation between the robot's pose and such shape descriptor is modeled as follows:

$$\mathbf{s} = f_s(\bar{c}) = f_s(f_c(f_m(\mathbf{r}_1, \mathbf{r}_2))). \quad (2)$$

By computing the time derivative of (2), we obtain the first-order dynamic model

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{f}_s}{\partial \mathbf{r}_1} \dot{\mathbf{r}}_1 + \frac{\partial \mathbf{f}_s}{\partial \mathbf{r}_2} \dot{\mathbf{r}}_2 = \mathbf{J}_s(\mathbf{r}) \dot{\mathbf{r}} \quad (3)$$

where  $\mathbf{J}_s(\mathbf{r}) = [\frac{\partial \mathbf{f}_s}{\partial \mathbf{r}_1}, \frac{\partial \mathbf{f}_s}{\partial \mathbf{r}_2}] \in \mathbb{R}^{p \times 2q}$  represents DJM, which relates the velocity of the end-effectors with the shape feature changes. As the mechanical properties of CRDOs are complex and difficult to obtain, these Jacobian matrices need to be numerically estimated. Note that the deformations of CRDO (not considering rheological objects) are only related to its own potential energy, the contact force with the manipulator and without the manipulation sequence of the manipulator. These conditions guarantee the establishment of (1). The quasi-static (3) holds when the materials properties of objects do not change significantly during manipulations, as DJM  $\mathbf{J}_s(\mathbf{r})$  represents the velocity mapping between the objects and the robot motions. And it is mentioned that this article only considers about the low/medium speed deformation of the object. For rapid deformation, this is not the category of shape-servoing, which is generally achieved by the fixed trajectory movement of the robot.

**Problem Statement:** Design a velocity-based controller for a dual-arm robot to automatically deform the CRDO toward the desired image shape (characterized by the feature vector  $\mathbf{s}_d$ ) while simultaneously estimating the unknown matrix  $\mathbf{J}_s(\mathbf{r})$ .

### C. Mathematical Properties

Before presenting our main result, some important properties are here introduced.

**Lemma 1:** (see [25]) Define the function  $\text{sig}^k(x) = |x|^k \text{sgn}(x)$ , which holds for any  $x \in \mathbb{R}$ , where  $k > 0$ , and satisfies the relation

$$\frac{d}{dt} \text{sig}^k(x) = k|x|^{k-1} \dot{x}, \quad k \geq 1. \quad (4)$$

**Lemma 2:** (see [26]) The inequality  $0 \leq |x| - x \tanh(x/\varepsilon) \leq \delta\varepsilon$  holds for any  $\varepsilon > 0$  and for any  $x \in \mathbb{R}$ , where  $\delta = 0.2785$  is a constant that satisfies  $\delta = e^{-(\delta+1)}$ .

**Assumption 1:** The DJM is divided into two components, namely,  $\mathbf{J}_s(\mathbf{r}) = \hat{\mathbf{J}}_s + \tilde{\mathbf{J}}_s$ , where  $\hat{\mathbf{J}}_s$  is the estimation of  $\mathbf{J}_s(\mathbf{r})$  and  $\tilde{\mathbf{J}}_s$  is the approximation error.

**Assumption 2:** The approximation error of the DJM is bounded  $\|\tilde{\mathbf{J}}_s\|^2 \leq \beta$ , for  $\beta$  as a unknown positive constant.

**Definition 1:** Given an arbitrary vector  $\mathbf{x} \in \mathbb{R}^n$ , let us introduce the following vectorial power definitions [25]:

$$\text{sig}^k(\mathbf{x}) = [\text{sig}^k(x_1), \dots, \text{sig}^k(x_n)]^T \in \mathbb{R}^n$$

$$|\mathbf{x}|^k = \text{diag} \left\{ |x_1|^k, \dots, |x_n|^k \right\} \in \mathbb{R}^{n \times n}.$$

### III. FEATURE EXTRACTION ALGORITHM

In this article, we use the contour moments [27] of  $\bar{c}$  to construct the compact feature vector  $\mathbf{s}$ . For digital images, the ordinary moment (of order  $i+j$ ) is defined as follows:

$$h_{ij} = \sum_{k=1}^N u_k^i v_k^j \Delta m_k, \quad i, j = 0, 1, \dots \quad (5)$$

where  $N$  is the number of points on the contour and  $\Delta m_k = \|\mathbf{c}_k - \mathbf{c}_{k-1}\|$  represents the distance between two adjacent pixels on the contour. The central moment of order  $i+j$  of the contour is given as

$$\eta_{ij} = \sum_{k=1}^N (u_k - \bar{u})^i (v_k - \bar{v})^j \Delta m_k, \quad i, j = 0, 1, \dots \quad (6)$$

where  $\bar{u}$  and  $\bar{v}$  are the central coordinates, i.e.,  $\bar{u} = h_{10}/h_{00}$  and  $\bar{v} = h_{01}/h_{00}$ . Similar to the classical Hu moments [11], the following seven contour moments are computed:

$$\bar{\varphi}_1 = \eta_{20} + \eta_{02}, \quad \bar{\varphi}_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\bar{\varphi}_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\bar{\varphi}_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2$$

$$\bar{\varphi}_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left( \begin{array}{l} (\eta_{30} + \eta_{12})^2 \\ -3(\eta_{21} + \eta_{03})^2 \end{array} \right)$$

$$\begin{aligned}
& + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left( \frac{3(\eta_{30} + \eta_{12})^2}{-(\eta_{21} + \eta_{03})^2} \right) \\
\bar{\varphi}_6 &= (\eta_{20} - \eta_{02}) \left( (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right) \\
& + 4\eta_{11}(\eta_{12} + \eta_{30})(\eta_{21} + \eta_{03}) \\
\bar{\varphi}_7 &= (3\eta_{21} - \eta_{03})(\eta_{12} + \eta_{30}) \left( \frac{(\eta_{12} - \eta_{30})^2}{-3(\eta_{21} + \eta_{03})^2} \right) \\
& - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left( \frac{3(\eta_{12} + \eta_{30})^2}{-(\eta_{21} + \eta_{03})^2} \right) \quad (7)
\end{aligned}$$

which are invariant to translation and rotation. However,  $\bar{\varphi}_k$  in (7) have a very large magnitude. A common approach to rectify these variables is to use its logarithmic form

$$\bar{s}_k = |\log(|\bar{\varphi}_k|)|, k \in [1, 7]. \quad (8)$$

Note, however, that the contour moments  $\bar{s}_k$  above (for  $k = 1, \dots, 7$ ) describe the object's shape in a rotation and translation invariant manner. For the considered bimanual shape servoing task, such information might not be sufficient to deform and position the object into a desired configuration. Therefore, further incorporating the object's centroid coordinates  $\bar{u}$  and  $\bar{v}$  (which encodes position) within the shape feature descriptor

$$\bar{s}_8 = \bar{u}, \bar{s}_9 = \bar{v}. \quad (9)$$

The angle of the contour's principal axis (which encodes orientation in the plane) [28], is also used in the shape descriptor

$$\bar{s}_{10} = \frac{1}{2} \tan^{-1} \left( \frac{2\eta_{11}}{\eta_{20} - \eta_{02}} \right). \quad (10)$$

As the magnitudes of the ten shape coordinates  $\bar{s}_k, k$  are different, we normalize them in a  $\pm 1$  range. For the contour moments, this is done as

$$s_i = \frac{\bar{s}_i - \frac{1}{7} \sum_{i=1}^7 \bar{s}_i}{\max(\bar{s}_i) - \min(\bar{s}_i)}, i = 1, \dots, 7. \quad (11)$$

For centroid coordinates as

$$s_8 = \frac{2\bar{s}_8 - C_w}{C_w}, s_9 = \frac{2\bar{s}_9 - C_h}{C_h} \quad (12)$$

where  $C_w$  and  $C_h$  represent the width and height of the camera, respectively, e.g.,  $640 \times 320$  and  $1920 \times 1080$ . For the principal axis angle, it is simply done as

$$s_{10} = \bar{s}_{10}/\pi. \quad (13)$$

Finally, the total shape feature vector is constructed as follows:  $\mathbf{s} = [s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}]^T \in \mathbb{R}^{10}$ . As seen from (5) to (13),  $\mathbf{s}$  only depends on the image contour. Slight illumination and contrast changes will not significantly affect its computation as the contour moment replaces the calculation of the region integral by a curvilinear integral, which represents a more robust alternative than the classical image moments [11]. And as the proposed contour moment breaks the translation and rotation invariance in [11], this guarantees the uniqueness of  $\mathbf{s}$ . The computation of the proposed feedback shape features is given in Algorithm 1.

---

**Algorithm 1:** Shape Feature  $\mathbf{s}$  Calculation Process.

---

**Require:** Fixed-size contour  $\bar{\mathbf{c}}$ ;

- 1: Calculate the difference-operator  $\Delta m_k, k = 1, \dots, N$ ;
  - 2: Calculate the ordinary moment  $h_{ij} \leftarrow (5)$ ;
  - 3: Calculate the central moment  $\eta_{ij} \leftarrow (6)$ ;
  - 4: Calculate contour moments  $\bar{\varphi}_1, \dots, \bar{\varphi}_7 \leftarrow (7)$ ;
  - 5: Data compression is conducted in (8);
  - 6: Calculate central coordinates  $\leftarrow (9)$ ;
  - 7: Calculate principal angle  $\leftarrow (10)$ ;
  - 8: Normalize shape feature  $\leftarrow (11), (12)$ , and (13);
  - 9: **return** Shape feature  $\mathbf{s}$ ;
- 

#### IV. CONTROLLER DESIGN

Two methods are here described to control the shape of the CRDO: Linear SMC (LSMC) and finite-time SMC (FTSMC). In the rest of this article, we denote the robot's control input as  $\mathbf{u} = \dot{\mathbf{r}}$

##### A. Linear Sliding Mode Control

Let us first define the error variables

$$\mathbf{e}_1 = \mathbf{s} - \mathbf{s}_d, \mathbf{e}_2 = \dot{\mathbf{s}} - \hat{\mathbf{J}}_s \mathbf{u} \quad (14)$$

and its time derivatives

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{s}} - \dot{\mathbf{s}}_d, \dot{\mathbf{e}}_2 = \ddot{\mathbf{s}} - \hat{\mathbf{J}}_s \dot{\mathbf{u}} - \hat{\mathbf{J}}_s \mathbf{u} \quad (15)$$

which we use to construct the linear sliding surfaces [29]

$$\sigma_1 = \mathbf{K}_1 \mathbf{e}_1 + \dot{\mathbf{e}}_1, \sigma_2 = \mathbf{K}_2 \mathbf{e}_2 + \dot{\mathbf{e}}_2 \quad (16)$$

for  $\mathbf{s}_d$  as the desired shape feature, and  $\mathbf{K}_k$  as symmetric positive-definite constant matrices. Considering (15) and Assumption 1, we can compute the time derivative of  $\sigma_1$  as

$$\dot{\sigma}_1 = \mathbf{K}_1 \hat{\mathbf{J}}_s \mathbf{u} + \mathbf{K}_1 \tilde{\mathbf{J}}_s \mathbf{u} - \mathbf{K}_1 \dot{\mathbf{s}}_d + \dot{\mathbf{e}}_1 \quad (17)$$

and design the following velocity control input:

$$\mathbf{u} = \hat{\mathbf{J}}_s^+ \mathbf{K}_1^{-1} (-\sigma_1 + \mathbf{K}_1 \dot{\mathbf{s}}_d - \ddot{\mathbf{e}}_1) \quad (18)$$

where  $\hat{\mathbf{J}}_s^+$  denotes the pseudoinverse of the adaptive matrix  $\hat{\mathbf{J}}_s$ . Since there is no power term and sign function,  $\mathbf{u}$  is continuous without chattering. To quantify the shape tracking error, we introduce the quadratic function  $V_1(\sigma_1) = \frac{1}{2} \sigma_1^T \sigma_1$ , whose time-derivative satisfies

$$\dot{V}_1(\sigma_1) = \sigma_1^T \dot{\sigma}_1 = -\sigma_1^T \sigma_1 + \sigma_1^T \mathbf{K}_1 \tilde{\mathbf{J}}_s \mathbf{u}. \quad (19)$$

From Assumption 2 and considering Young's inequality, we can obtain the following relation:

$$\sigma_1^T \mathbf{K}_1 \tilde{\mathbf{J}}_s \mathbf{u} \leq \lambda_{\mathbf{K}_1} \|\sigma_1\|^2 / 4 + \beta \|\mathbf{u}\|^2 \quad (20)$$

where  $\lambda_{\mathbf{K}_1}$  denotes the maximum eigenvalue of  $\mathbf{K}_1$ . Substitution of (20) into (19) yields

$$\dot{V}_1(\sigma_1) \leq -(1 - \lambda_{\mathbf{K}_1}/4) \|\sigma_1\|^2 + \beta \|\mathbf{u}\|^2. \quad (21)$$

With this method, the DJM is adaptively computed as

$$\dot{\mathbf{J}}_s = (\ddot{\mathbf{s}} - \hat{\mathbf{J}}_s \dot{\mathbf{u}} - \mathbf{K}_2^{-1} \varpi) \mathbf{u}^+$$

$$\varpi = -\sigma_2 - \ddot{\mathbf{e}}_2 - \sigma_2^{T+} \tanh\left(\|\mathbf{u}\|^2/\chi\right) \hat{\beta} \|\mathbf{u}\|^2 \quad (22)$$

where the variable  $\hat{\beta}$  is updated with the adaptive rule:

$$\dot{\hat{\beta}} = \tanh\left(\|\mathbf{u}\|^2/\chi\right) \|\mathbf{u}\|^2 - \gamma \hat{\beta} \quad (23)$$

for  $\chi$  and  $\gamma$  as positive constants.  $\ddot{\mathbf{s}} - \hat{\mathbf{J}}_s \dot{\mathbf{u}}$  indicates the update of  $\mathbf{s}$  and  $\mathbf{u}$  to  $\hat{\mathbf{J}}_s$ . The term  $\varpi$  and  $\tanh$  are used to compensate the estimation error of  $\hat{\mathbf{J}}_s$  and improve the system manipulability. The classical Slotine-Li's  $\sigma$ -modification method [22] is added in (23) to avoid parameter drift.

*Proposition 1:* Consider the dynamic system (3) in closed-loop with the adaptive controller (18), (22), (23). Given desired shape vector  $\mathbf{s}_d$ , there exists an appropriate set of control parameters that ensure that: (1) all signals in the closed-loop system remain uniformly ultimately bounded (UUB), and (2) the deformation error  $\mathbf{e}_1$  asymptotically converges to a compact set around zero.

*Proof:* Consider the energy-like function

$$V_2(\sigma_1, \sigma_2, \tilde{\beta}) = V_1(\sigma_1) + \frac{1}{2} \sigma_2^T \sigma_2 + \frac{1}{2} \tilde{\beta}^2 \quad (24)$$

for  $\tilde{\beta} = \beta - \hat{\beta}$ . Considering the Young's inequality,  $\tilde{\beta} \hat{\beta} \leq (\beta^2 - \tilde{\beta}^2)/2$  and invoking (21)–(24), we can show that the time derivative of  $V_2$  satisfies

$$\begin{aligned} \dot{V}_2(\sigma_1, \sigma_2, \tilde{\beta}) &\leq -\left(1 - \frac{\lambda_{\mathbf{K}_1}}{4}\right) \|\sigma_1\|^2 - \|\sigma_2\|^2 - \frac{\gamma}{2} \tilde{\beta}^2 + b \\ &\leq -a V_2(\sigma_1, \sigma_2, \tilde{\beta}) + b \end{aligned} \quad (25)$$

where  $a = \min((2 - \lambda_{\mathbf{K}_1}/2), 2, \gamma)$ ,  $b = \beta\delta\chi + \gamma\beta^2/2 > 0$ . Refer to (24),  $V_2$  is decreasing so that it has an upper boundary. The state signals  $\sigma_1$ ,  $\sigma_2$ , and  $\tilde{\beta}$  are endowed with asymptotic stability and remain UUB [30] by selecting a matrix  $\mathbf{K}_1$  that ensures that  $a > 0$ . This implies that the shape error  $\mathbf{e}_1$  asymptotically converges to a compact set around zero and ensures that the Jacobian estimation error  $\hat{\mathbf{J}}_s$  is bounded [31]. For further development, we give the upper bound for  $V_2$  and apply the comparison lemma [32] to (25) yields:  $V_2(t) \leq e^{-at} V_2(0) + \frac{b}{a}$ . Through  $V_2$ , it satisfies  $V_2 \geq \frac{1}{2} \|\sigma\|^2$  where  $\sigma = [\sigma_1^T, \sigma_2^T]^T$ . Thus,  $\sigma$  satisfies  $\|\sigma\| \leq e^{-\frac{a}{2}t} \sqrt{2V_2(0)} + \sqrt{\frac{2b}{a}}$ . Thus, when  $t \rightarrow \infty$ ,  $\|\sigma\| \rightarrow \sqrt{\frac{2b}{a}}$ . The convergence set of  $\sigma$  depends on  $\gamma$  and  $\chi$ , as  $\beta$  and  $\delta$  are constants. Although the above results validate that  $\mathbf{e}_1$  only converges to a local minimum near the equilibrium point, we can adjust the parameters to make this compact set small. Therefore, it can be considered that LSMC can deform the CRDO to the desired shape asymptotically. ■

### B. Finite-Time Sliding Mode Control

Many practical applications demand a tight timing performance, which cannot be accomplished by simply increasing the control gain; FTSMC is designed to address these issues. The nonsingular terminal sliding surface [33] is given as

$$\begin{aligned} \sigma_1 &= \mathbf{e}_1 + \alpha_1 \text{sig}^{p_1}(\dot{\mathbf{e}}_1) + \beta_1 \text{sig}^{q_1}(\mathbf{e}_1) \\ \sigma_2 &= \mathbf{e}_2 + \alpha_2 \text{sig}^{p_2}(\dot{\mathbf{e}}_2) + \beta_2 \text{sig}^{q_2}(\mathbf{e}_2) \end{aligned} \quad (26)$$

where  $p_1 \in (1, 2)$ ,  $q_1 > p_1$ ,  $p_2 \in (1, 2)$ ,  $q_2 > p_2$  and  $\alpha_1, \alpha_2, \beta_1, \beta_2 > 0$  are all positive constants. Invoking (15) and Lemma 1, the time derivative of  $\sigma_1$  is given by

$$\dot{\sigma}_1 = \hat{\mathbf{J}}_s \mathbf{u} + \tilde{\mathbf{J}}_s \mathbf{u} - \dot{\mathbf{s}}_d + \alpha_1 p_1 |\dot{\mathbf{e}}_1|^{p_1-1} \ddot{\mathbf{e}}_1 + \beta_1 q_1 |\mathbf{e}_1|^{q_1-1} \dot{\mathbf{e}}_1 \quad (27)$$

which we use to design the following velocity control input:

$$\mathbf{u} = \hat{\mathbf{J}}_s^+ \begin{pmatrix} -\alpha_1 p_1 |\dot{\mathbf{e}}_1|^{p_1-1} \ddot{\mathbf{e}}_1 - \varepsilon_1 \text{sgn}(\sigma_1) \\ -\beta_1 q_1 |\mathbf{e}_1|^{q_1-1} \dot{\mathbf{e}}_1 + \dot{\mathbf{s}}_d \end{pmatrix}. \quad (28)$$

As (28) has a sgn function,  $\mathbf{u}$  has a certain chattering phenomenon, which is unfavorable to the robot in practice. sgn can be replaced with tanh to reduce the chattering effects. However, this will affect the overall convergence performance of the system. Therefore, there is a trade balance in suppressing chattering and tuning system dynamic characteristics. With the above controller, we obtain a new expression for  $\dot{V}_1$

$$\dot{V}_1(\sigma_1) = \sigma_1^T \dot{\sigma}_1 = -\varepsilon_1 \sigma_1^T \text{sgn}(\sigma_1) + \sigma_1^T \tilde{\mathbf{J}}_s \mathbf{u} \quad (29)$$

where by considering the Young's inequality

$$\sigma_1^T \tilde{\mathbf{J}}_s \mathbf{u} \leq \|\sigma_1\|^2/4 + \beta \|\mathbf{u}\|^2 \quad (30)$$

we can show (after some algebraic operations) that the following relation is satisfied:

$$\dot{V}_1(\sigma_1) \leq -\varepsilon_1 \|\sigma_1\| + \|\sigma_1\|^2/4 + \beta \|\mathbf{u}\|^2. \quad (31)$$

Finally, the adaptive rule for the DJM is given as

$$\begin{aligned} \dot{\hat{\mathbf{J}}}_s &= \left( \ddot{\mathbf{s}} - \hat{\mathbf{J}}_s \dot{\mathbf{u}} + \varepsilon_2 \text{sgn}(\sigma_2) + \varpi \right. \\ &\quad \left. + \alpha_2 p_2 |\dot{\mathbf{e}}_2|^{p_2-1} \ddot{\mathbf{e}}_2 + \beta_2 q_2 |\mathbf{e}_2|^{q_2-1} \dot{\mathbf{e}}_2 \right) \mathbf{u}^+ \\ \varpi &= \sigma_2^{T+} \left( \tanh\left(\|\mathbf{u}\|^2/\chi\right) \hat{\beta} \|\mathbf{u}\|^2 + \frac{1}{4} \|\sigma_1\|^2 \right). \end{aligned} \quad (32)$$

$\ddot{\mathbf{s}} - \hat{\mathbf{J}}_s \dot{\mathbf{u}}$  and  $\varpi$  are similar as that in (22) and  $\text{sgn}(\sigma_2)$  compensates for the estimation error of  $\hat{\mathbf{J}}_s$ . And,  $\alpha_2 p_2 |\dot{\mathbf{e}}_2|^{p_2-1} \ddot{\mathbf{e}}_2$  and  $\beta_2 q_2 |\mathbf{e}_2|^{q_2-1} \dot{\mathbf{e}}_2$  guarantee the finite-time estimation of  $\hat{\mathbf{J}}_s$  and the overall stability combined with the controller (28).

*Proposition 2:* Consider the dynamic system (3) in closed-loop with the adaptive controller given by (28), (32), (23). For this system, all state signals are semiglobal practical finite time stable (SGPFS) [34], and the shape error  $\mathbf{e}_1$  converges to a compact set within a finite time without any singularities during the task.

*Proof:* Consider the energy-like function (24) computed with the sliding surface (26). By using the following square completion property:

$$-\gamma \tilde{\beta}^2/2 \leq \gamma/8 - |\tilde{\beta}|/2 \quad (33)$$

along with (31), we can show that the time derivative of  $V_2$  satisfies the following relations:

$$\begin{aligned} \dot{V}_2 &\leq -\varepsilon_1 \|\sigma_1\| - \varepsilon_2 \|\sigma_2\| - \gamma \tilde{\beta}^2/2 + (\beta\delta\chi + \gamma\beta^2/2) \\ &\leq -a V_2^{\frac{1}{2}} + b \end{aligned} \quad (34)$$

where  $a = \min(2\varepsilon_1, 2\varepsilon_2, \gamma)$ ,  $b = \beta\delta\chi + \gamma\beta^2/2 + \gamma/8 > 0$ . By selecting appropriate parameters that ensure  $a > 0$ , then  $V_2$

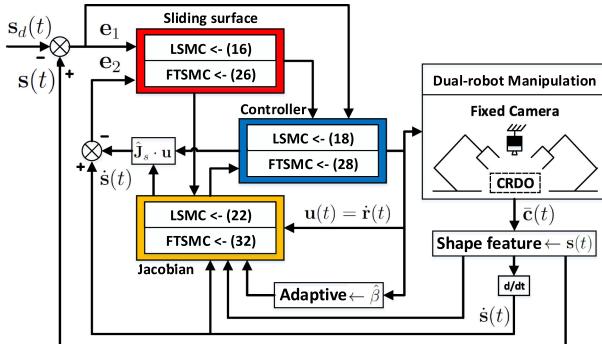


Fig. 3. Block diagram of the proposed manipulation framework.

converges to a compact set  $\{V_2 | V_2 \leq \Omega^2, \text{ time } t \geq T_s\}$  within the convergence time  $T_s$  calculated by [34]

$$T_s = \frac{2}{av} \left( V_2^{\frac{1}{2}} (\mathbf{X}(0)) - \Omega \right), \quad \Omega = \frac{b}{(1-v)a} \quad (35)$$

where  $\mathbf{X}(0) = [\sigma_1^T(0), \sigma_2^T(0), \tilde{\beta}(0)]^T \in \mathbb{R}^{2p+1}$  and  $v \in [0, 1]$ . Then,  $\mathbf{X}$  remains in the compact set defined by  $\Omega_v = \{\mathbf{X} | \|\mathbf{X}\| \leq \sqrt{2}\Omega, t \geq T_s\}$ , which means that  $\sigma_1, \sigma_2$ , and  $\tilde{\beta}$  are SGPFS. As  $V_2 \geq \frac{1}{2}\|\sigma\|^2$ , it yields  $\|\sigma\| \leq \sqrt{\frac{2b}{(1-v)a}}$ . By adjusting  $a$  and  $b$ , we can make the convergence range of  $V_2$  smaller. The above analysis implies that the shape error  $e_1$  and estimation error  $e_2$  converge to a compact set around zero within finite time [33]. In the traditional terminal SMC [35], there exists  $|e_i|^w, i = 1, 2$  and  $w < 0$ . Thus,  $\mathbf{u}$  cannot be guaranteed to be bounded when  $|e_i|$  is near the origin, namely, numerical singularity. For the proposed velocity command (28), it does not contain any negative fractional power since  $p_i \in (1, 2), q_i > p_i, i = 1, 2$ ; thus it is singularity-free. The above results validate that the CRDO can be deformed into the target configuration within finite time along NFTSM without singularity and all signals as SGPFS.

LSMC and FTSMC are the overdetermined controllers, which means that  $e_1$  only converges to a local minimum effected by the feasibility of  $s_d$  [9]. A local minimum is inevitable for the Jacobian-based approach in the shape deformation. ■

The workflow of the proposed framework is given in Algorithm 2. Fig. 3 gives the block diagram of the proposed dual-arm robot manipulation framework.

*Remark 1:* FTSMC aims to adjust the dynamic performance of the system without changing the control gain, rather than simply speeding up. In contrast, the object needs a slow deformation sometimes.

## V. EXPERIMENTS

To validate the effectiveness of our proposed framework, we conducted an experimental study with a dual-arm robotic platform (composed of two UR5 robots) manipulating several types of CRDO: Elastic cable, sponge, plastic folder, nonhomogeneous (NH) beam, articulated wallet, and a rigid box, see Fig. 4. We fix two plastic sticks on the end-effector to connect CRDO, see Fig. 5. The sticks are fixed in the support, see

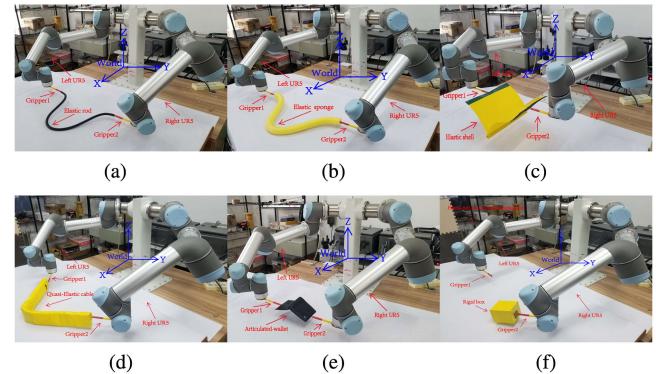


Fig. 4. Dual-arm UR5 experiment platform manipulating CRDO. (a) Elastic cable. (b) Linear sponge. (c) Plastic folder. (d) NH beam. (e) Articulated wallet. (f) Rigid box.

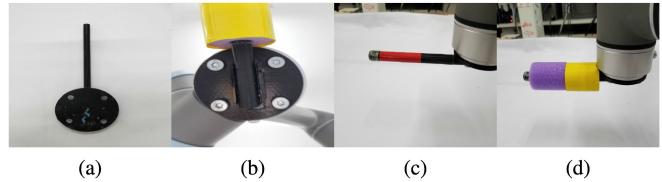


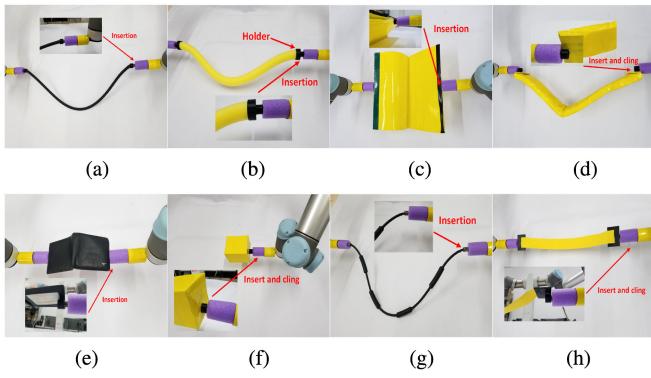
Fig. 5. Base connection with end-effector.

### Algorithm 2: Workflow of the Proposed Framework.

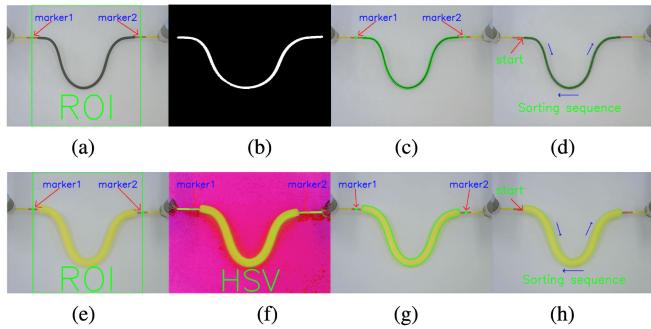
**Require:** Threshold; Max; Sw: default=1;

- 1: Give a target shape feature  $s_d$  calculated by  $\bar{c}^*$ ;
- 2: Conduct small deformations around the starting configuration to initialize  $\hat{J}_s(0)$  and start the manipulation;
- 3:  $k = 0, \hat{\beta}(0) \leftarrow 0.001$
- 4: **while**  $\|e_1\| \geq \text{Threshold}$  and  $k < \text{Max}$  **do**
- 5: Record the current position  $r$  and velocity  $u$ ;
- 6: Record the current contour  $\bar{c}$  of the object;
- 7: Calculate the current shape feature  $s \leftarrow (5) - (13)$ ;
- 8: Calculate error signals  $e_1$  and  $e_2 \leftarrow (14)$ ;
- 9: Calculate  $\sigma_i, (16) \leftarrow Sw = 1$  or  $(26) \leftarrow Sw = 2$ ;
- 10: Update the adaptive term  $\hat{\beta} \leftarrow (23)$ ;
- 11: Update  $u, (18) \leftarrow Sw = 1$  or  $(28) \leftarrow Sw = 2$ ;
- 12: Update  $\hat{J}_s, (22) \leftarrow Sw = 1$  or  $(32) \leftarrow Sw = 2$ ;
- 13: Dual-arm robot moves using the updated  $u$ ;
- 14:  $k = k + 1$ ;
- 15: **end while**

Fig. 5(a). The supports are fixed in the end-effector by screws, see Fig. 5(b) and (c). Marker sponges cover stick outside to define ROI, see Fig. 5(d). Various holders are printed by a 3-D printer to connect CRDO with end-effector, see Fig. 6. As we did not consider the object grasping, we need to connect CRDO and end-effector in advance. The velocity input is constructed as  $u = [u_1^T, u_2^T]^T \in \mathbb{R}^6$ , where the coordinates of  $u_i^T = [u_{i1}, u_{i2}, u_{i3}]$  represent the linear and angular velocities along the  $x, y$ , and  $z$  axes, respectively. The maximum speed for linear motions is set to  $|u_{ij}| \leq 0.01 \text{ m/s}$  (for  $j = 1, 2$ ), and for angular motions to



**Fig. 6.** Detailed connections of CRDO with the end-effector. (a) Cable. (b) Sponge. (c) Folder. (d) Beam. (e) Wallet. (f) Box. (g) Rod. (h) Cloth.

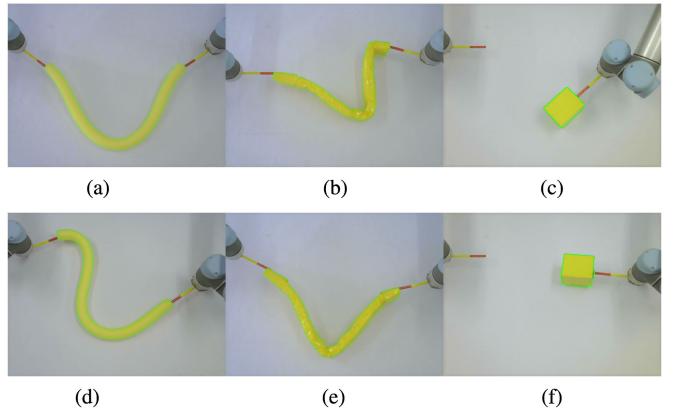


**Fig. 7.** Image processing for generating a fixed-sampled closed contour of various objects. (a) ROI selection. (b) Binary. (c) Extraction. (d) Fixed-sampled. (e) ROI selection. (f) HSV. (g) Extraction. (h) Fixed-sampled.

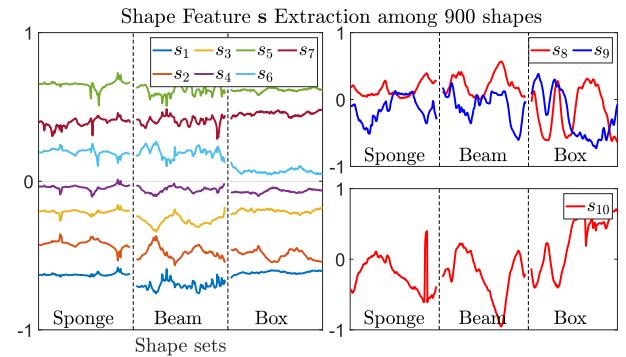
$|u_{i3}| \leq 0.1 \text{ rad/s}$ . The experimental environment is programmed in ROS/Python. A Logitech C270 camera is used to capture the shape of the objects. All image processing algorithms are implemented in a Linux PC at 30 FPS with OpenCV. The experimental video can be downloaded from: [https://github.com/q546163199/experiment\\_video/tree/master/paper3/video.mp4](https://github.com/q546163199/experiment_video/tree/master/paper3/video.mp4)

### A. Image Processing

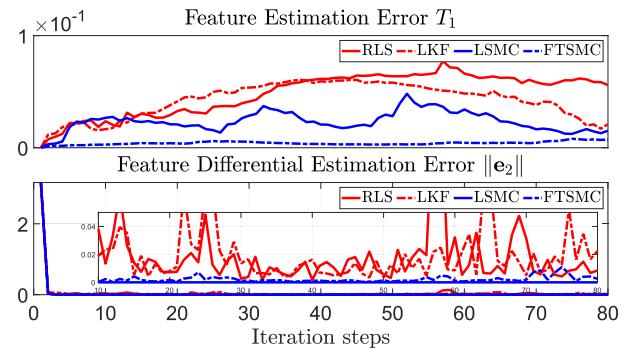
The contour extraction process (depicted in Fig. 7) is as follows: The centroids of red areas near the grippers are first extracted and marked by marker1 and marker2, respectively. The object's region of interest (ROI) is the rectangle defined by both markers. The manipulated black objects' binary image is extracted from the ROI by using OpenCV's morphological algorithms for denoising. We transform their image into HSV for the manipulated yellow objects and then conduct mask processing to obtain the binary image. The contour of the objects (either black or yellow) is then extracted by using OpenCV's *findcontour*. The point closest to marker1 is recorded as the starting point of the contour and used to arrange the contour clockwise to get a new contour. Finally, a fixed-step (set to  $N = 300$ ) algorithm [10] is used to construct the contour  $\bar{c}$  with a constant number of points  $c_i$ .



**Fig. 8.** Different shapes of various objects manipulated by dual-arm UR5. (a) Linear sponge. (b) NH beam. (c) Rigid box. (d) Linear sponge. (e) NH beam. (f) Rigid box.



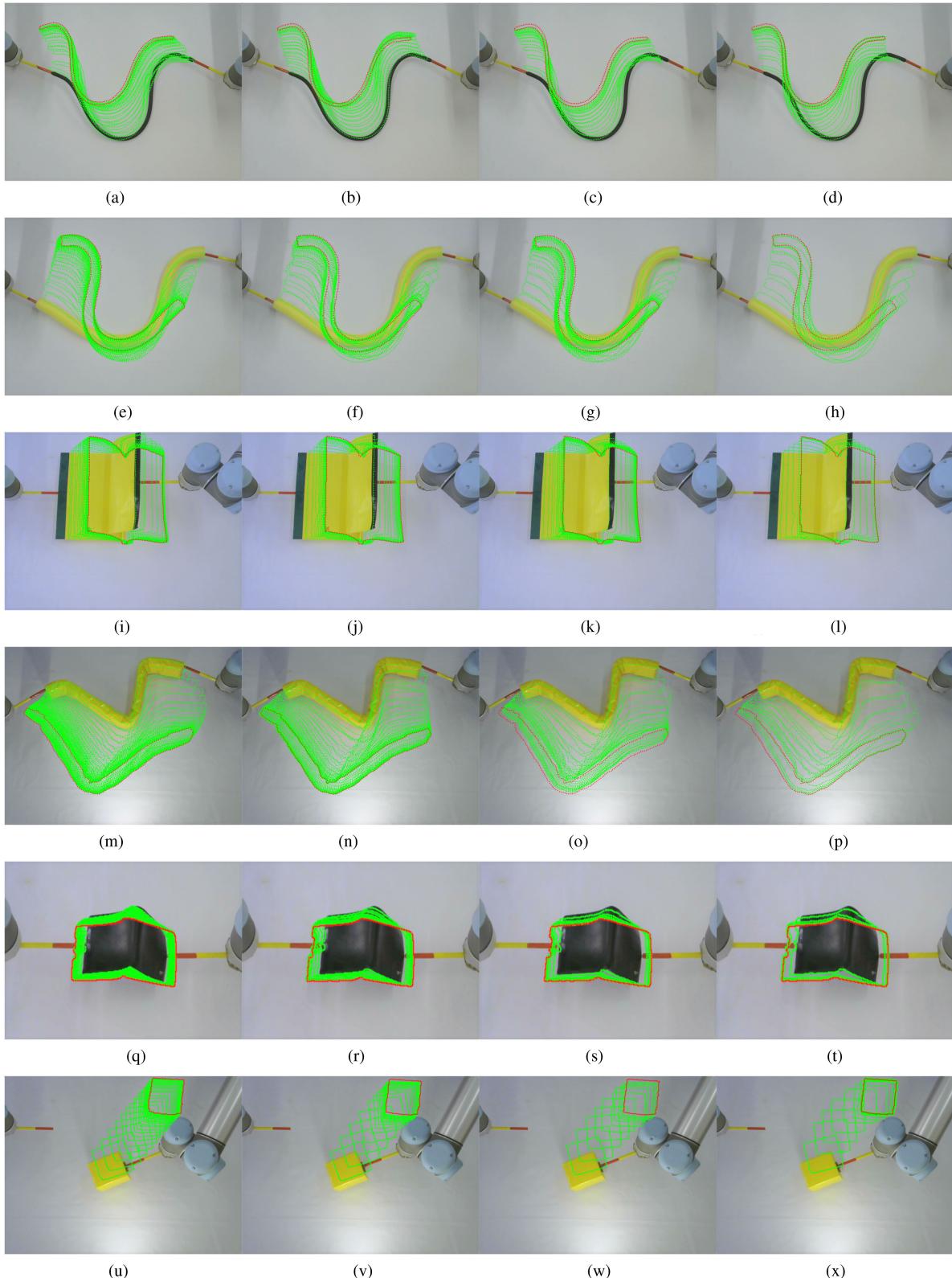
**Fig. 9.** Contour moments extraction (5)–(13) of 900 shapes among sponge, beam and box. Each group has 300 shapes, separately.



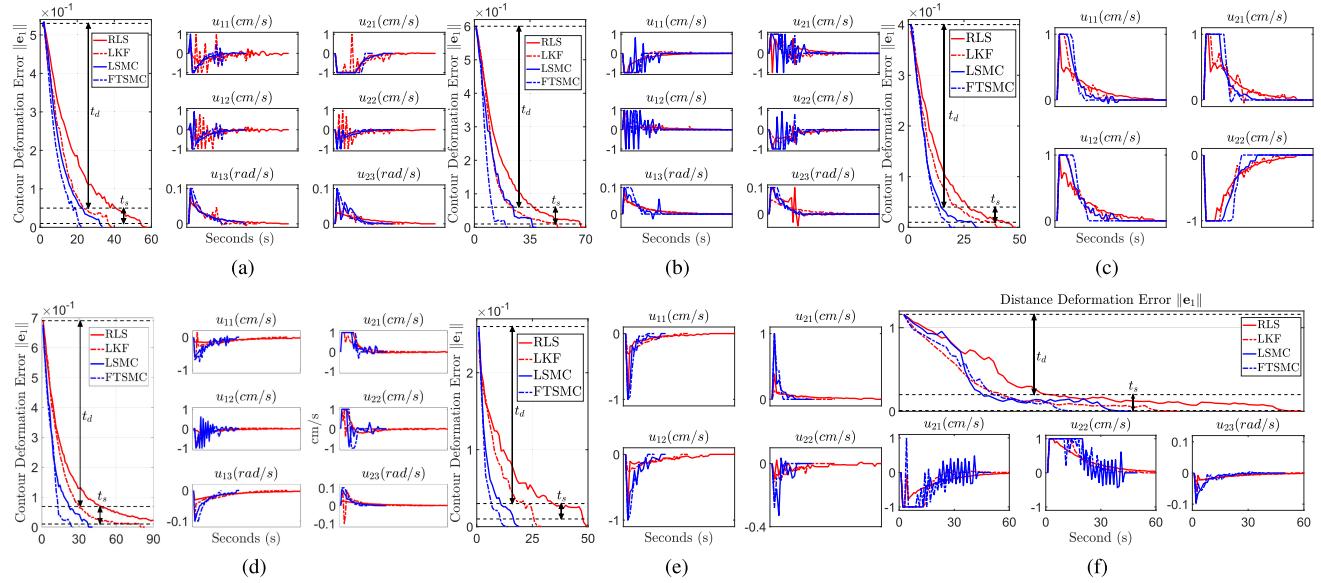
**Fig. 10.** Profiles of the criteria  $T_1$  and  $\|e_2\|$  with dual-arm UR5 executing a given trajectory  $r$ . It gives the estimation effect of  $\hat{J}_s$  among RLS [36], LKF [37], LSMC (22) and FTSMC (32), respectively.

### B. Validation of the Contour Moments Extraction

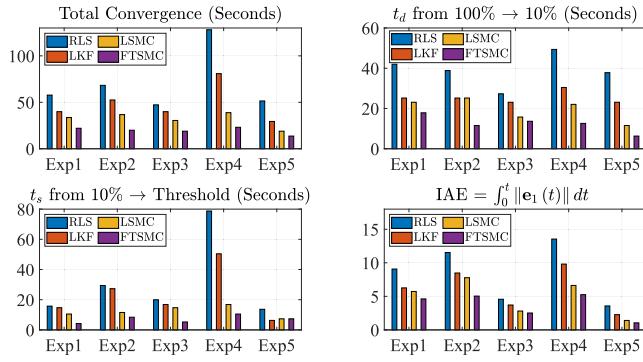
The robotic platform is commanded to continuously deform three CRDO into various shapes, see Fig. 8 (such shaping actions are demonstrated in the accompanying multimedia file). The contour moments that are extracted from these robot-object motions are depicted in Fig. 9, which shows that as shape



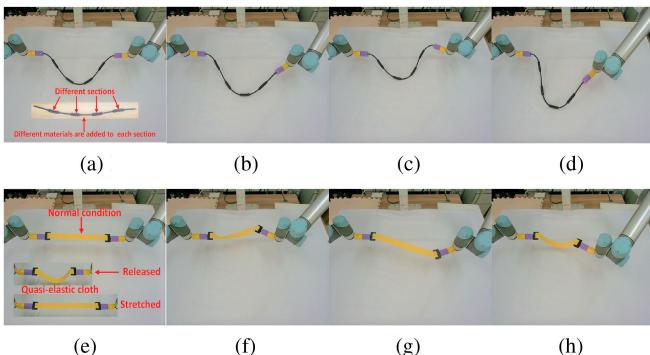
**Fig. 11.** Initial (black solid line), transition (green solid line) and target (red solid line) contour deformation trajectories from Exp1 to Exp5. Exp6 represents the positioning of the rigid box. All six experiments are conducted among RLS [36], [10], LKF [37], [10], LSMC (18)–(22) and FTSMC (28)–(32). (a) Exp1-Cable RLS. (b) Exp1-Cable LKF. (c) Exp1-Cable LSMC. (d) Exp1-Cable FTSMC. (e) Exp2-Sponge RLS. (f) Exp2-Sponge LKF. (g) Exp2-Sponge LSMC. (h) Exp2-Sponge FTSMC. (i) Exp3-Folder RLS. (j) Exp3-Folder LKF. (k) Exp3-Folder LSMC. (l) Exp3-Folder FTSMC. (m) Exp4-Beam RLS. (n) Exp4-Beam LKF. (o) Exp4-Beam LSMC. (p) Exp4-Beam FTSMC. (q) Exp5-Wallet RLS. (r) Exp5-Wallet LKF. (s) Exp5-Wallet LSMC. (t) Exp5-Wallet FTSMC. (u) Exp6-Rigid-Box RLS. (v) Exp6-Rigid-Box LKF. (w) Exp6-Rigid-Box LSMC. (x) Exp6-Rigid-Box FTSMC.



**Fig. 12.** Profiles of contour deformation error  $\|e_1\|$  of Exp1-Exp5, distance deformation error  $\|e_1\|$  of Exp6 and velocity command  $u$  among four methods, namely, RLS [36], [10], LKF [37], [10], LSMC (18)–(22), and FTSMC (28)–(32). (a) Exp1 Result. (b) Exp2 Result. (c) Exp3 Result. (d) Exp4 Result. (e) Exp5 Result. (f) Exp6 Result.



**Fig. 13.** Performance comparison in the deformation tasks (Exp1 to Exp5).



**Fig. 14.** Detailed composition description of the heterogeneous rod and the quasi-elastic cloth with various deformations. (a) Rod. (b) Case1. (c) Case2. (d) Case3. (e) Cloth. (f) Case1. (g) Case2. (h) Case3.

changes, the features' profiles also change. This figure shows that the contour moments can smoothly represent the infinite-dimensional object's shape.

### C. Estimation of the DJM

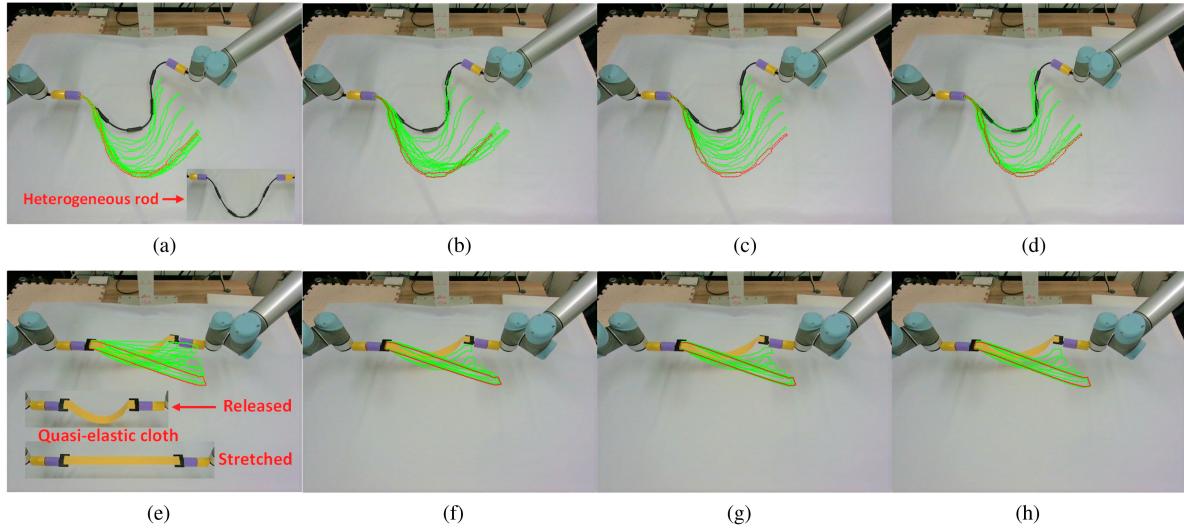
In this section, we command the robot to execute slow and smooth trajectories  $\dot{r}(t)$  with the proposed LSMC and FTSMC controllers. The purpose of this test is to compare the accuracy of the Jacobian estimation  $\hat{\mathbf{J}}_s$  obtained with these methods, and that obtained with traditional recursive least square (RLS) [36] and linear Kalman filter (LKF) [37]. For that, we initialize  $\hat{\mathbf{J}}_s(0)$  with a random matrix and then perform small local motions and run the estimator to obtain an “good-enough” matrix  $\hat{\mathbf{J}}_s$ . To quantify the accuracy of such estimation, we introduce the following metric:

$$T_1 = \|\mathbf{s} - \hat{\mathbf{s}}\| \quad (36)$$

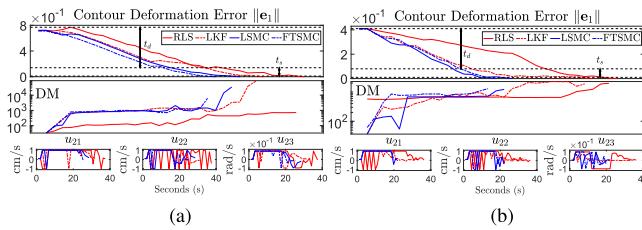
where  $\hat{\mathbf{s}}$  is the estimated feature that is updated as follows:

$$\dot{\hat{\mathbf{s}}} = \hat{\mathbf{J}}_s \cdot \mathbf{u}, \quad \hat{\mathbf{s}}(0) = \mathbf{s}(0) \quad (37)$$

where  $\hat{\mathbf{J}}_s$  is estimated by RLS, LKF, LSMC, and FTSMC.  $\hat{\mathbf{s}}(0)$  and  $\mathbf{s}(0)$  are the initial values of  $\hat{\mathbf{s}}$  and  $\mathbf{s}$ , respectively. Fig. 10 illustrates the evolution of  $T_1$  and  $\|\mathbf{e}_2\|$  for the motion  $\dot{r}$  executed by the robot in the previous test. It can be seen that FTSMC provides the best approximation among the considered algorithms (viz. RLS, LKF, LSMC). These results corroborate that the proposed FTSMC can accurately predict the shape features  $\mathbf{s}$  and its differential change  $\dot{\mathbf{s}}$ , enabling to guide the robot with the estimated matrix  $\hat{\mathbf{J}}_s$ .



**Fig. 15.** Initial (black solid line), transition (green solid line) and target (red solid line) contour deformation trajectories of Exp7 and Exp8. Both experiments are conducted among RLS [36], [10], LKF [37], [10], LSMC (18)–(22), and FTSMC (28)–(32). (a) Exp7-Rod RLS. (b) Exp7-Rod LKF. (c) Exp7-Rod LSMC. (d) Exp7-Rod FTSMC. (e) Exp8-Cloth RLS. (f) Exp8-Cloth LKF. (g) Exp8-Cloth LSMC. (h) Exp8-Cloth FTSMC.



**Fig. 16.** Profiles of contour deformation error  $\|e_1\|$  of Exp7 and Exp8 and velocity command  $u$  among RLS [36][10], LKF [37][10], LSMC (18)–(22), and FTSMC (28)–(32). Note that  $y$  of DM is displayed in logarithm, thus the actual fluctuation is more obvious. (a) Exp7 Result. (b) Exp8 Result.

#### D. Shape Servoing With CRDO

In this section, we validate the performance of the proposed shape controller by commanding the robot to deform six objects (with time-varying mechanical properties): Elastic cable, linear sponge, plastic folder, NH beam (composed of different sponges, which is consistent with CRDO), articulated wallet, rigid box into a desired contour  $\bar{c}^*$  (which is transformed into a desired feature vector  $s_d$ ). We denote the experiments with these objects as Exp1...Exp6, respectively. We compare the error minimization performance of the FTSMC/LSMC methods with that of a classical visual servoing controller (i.e. à la Chaumette) with the DJM adaptively estimated with RLS [36] and LKF [37].

Additionally to the shape error  $\|e_1\|$ , we use the following four indices to analyze the performance of the system:

- 1) Convergence step  $T_{\max}$  indicates the total deformation time consumed by the system.
- 2) Decay time  $t_d$  indicates the time required to decrease from  $\|e_1\|$  to 10% of its initial value  $\|e_1(0)\|$ .
- 3) Settling time  $t_s$  indicates the time required to decrease from 10% of  $\|e_1(0)\|$  to 0.01.

4) Integrated absolute error (IAE)  $IAE = \int_0^t \|e_1\| dt$  indicates the cumulative error and shows the energy consumption of the system.

**Fig. 11** illustrates the active shaping motions (represented by the moving green contours) of the objects toward the desired configuration (represented by the red contour). From the analysis of the FTSMC in (35), we can see that the parameters  $\varepsilon_1, \varepsilon_2, \gamma, \chi$  determine the minimization performance  $\sigma_i \rightarrow 0$ . The parameter  $\varepsilon_1$  represents a control gain with similar function to  $K_1$  in (16). As there exists an adjustable power term in (28), the accuracy and speed of the system can be adjusted without increasing  $\varepsilon_1$ . This valuable property is lacking in the other three methods, as they must increase the “proportional-like” control gain to achieve a comparable convergence rate (which in turn may generate a large input velocity  $u$ ). This figure qualitatively depicts the contour trajectories of the six objects (each in a different row) with the four controllers (each in a different column). **Fig. 12** quantitatively depicts the time evolution of the shape error  $\|e_1\|$  and the respective driving control inputs  $u$ . From these temporal profiles, we can see that the proposed FTSMC provides the best control performance compared with the fastest error minimization to the other methods. LSMC the second-best, while RLS and LKF give a similar performance. A summary of the performance metrics for the conducted manipulation experiments is depicted in **Fig. 13**. This figure quantitatively shows that the proposed control method achieves the best relative to the time instance  $t_d$  (which means that the elastic cable can be coarsely deformed into the target configuration), the instance  $t_s$  (a property introduced by the terminal attractor  $\text{sig}(\cdot)$ ), and the IAE index (which demonstrates the moderate consumption of energy).

Two materials are added to demonstrate the effectiveness of the proposed framework for the active deformation of CRDO further, i.e., heterogeneous rod (Exp7) with various materials

with different thicknesses and sizes in each section, and quasi-elastic cloth (Exp8) with different physical states under released with plasticity and stretched with elasticity, respectively. Both objects have time-varying physical characteristics, and the detailed compositions are presented in Fig. 14. The deformability measure  $DM = \sqrt{\|\hat{\mathbf{J}}_s \cdot \hat{\mathbf{J}}_s^T\|}$  is introduced to evaluate the estimation of DJM [18]. Fig. 15 shows the deformations of Exp7 and Exp8 with the same four methods as before. Fig. 16 confirms that FTSMC provides superior performance in the active deformation of heterogeneous objects. The drastic changes of DM are caused by the heterogeneous (time-varying) characteristics of the rod and cloth in the manipulation process.

*Remark 2:* Discrete difference [38] and Levant differentiator [39] are used to obtain  $\dot{s}$  and  $\ddot{s}$  in the experiments. In addition, the tactile sensor can be used to measure friction between gripper (e.g., Robotiq-85) and objects. Then it monitors whether grippers and objects are connected in real-time.

*Remark 3:* High bending curvatures may render (22), (32) discontinuous, which causes singularities in  $\hat{\mathbf{J}}_s$ . Thus,  $\beta$  should be large-enough to ensure Assumption 2, which in turn, enlarges the system's convergence area; This might be reflected in oscillations around the origin [40].

## VI. CONCLUSION

In this article, we present a new visual servoing framework to control the shape of CRDO with dual-arm robots. To characterize the objects' infinite-dimensional shape, a compact feature vector is constructed with contour moments from the observed 2-D image of the object. A new finite-time sliding mode controller is proposed to automatically deform the object into a desired shape and simultaneously estimate the DJM; The stability of this method is rigorously analyzed. To validate the performance of our new method, we report a detailed experimental study with a robotic system manipulating CRDO.

The proposed method has some limitations. First, this controller cannot be used for purely *plastic* objects, such as certain food materials or clay. Second, although FTSMC controller provides the best performance, due to the plenty of control parameters, sometimes it is difficult to adjust them to obtain a satisfactory performance. The hard saturation adopted in this article is used to limit  $u$ , however, we do not consider its impact on the system's stability. For future work, our team is currently extending the current method to perform 3-D manipulation tasks. Also, we are incorporating shape planning capabilities into the framework so as to conduct complex multi-action shaping tasks (e.g., packing nonrigid objects).

## REFERENCES

- [1] J. Zhu *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robot. Autom. Mag.*, 2021.
- [2] S. Tokumoto and S. Hirai, "Deformation control of rheological food dough using a forming process model," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, 2002, pp. 1457–1464.
- [3] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 4, pp. 2016–2026, Aug. 2021.
- [4] E. J. Park and J. K. Mills, "Static shape and vibration control of flexible payloads with applications to robotic assembly," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 6, pp. 675–687, Dec. 2005.
- [5] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3893–3900.
- [6] H. Wang, B. Yang, J. Wang, X. Liang, W. Chen, and Y. Liu, "Adaptive visual servoing of contour features," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 811–822, Apr. 2018.
- [7] D. Navarro-Alarcon *et al.*, "Automatic 3D manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 429–441, Apr. 2016.
- [8] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognit.*, vol. 37, no. 1, pp. 1–19, 2004.
- [9] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2D image shapes," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 272–279, Feb. 2018.
- [10] J. Qi, W. Ma, D. Navarro-Alarcon, H. Gao, and G. Ma, "Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls," 2021, *arXiv:2008.06896*.
- [11] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [12] W. Chen, M. Li, and X. Su, "Error analysis about CCD sampling in fourier transform profilometry," *Optik - Int. J. Light Electron Opt.*, vol. 120, no. 13, pp. 652–657, 2009.
- [13] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 254–261, Apr. 2018.
- [14] R. Lagneau, A. Krupa, and M. Marchal, "Automatic shape control of deformable wires based on model-free visual servoing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5252–5259, Oct. 2020.
- [15] D. Navarro-Alarcon and Y. Liu, "A dynamic and uncalibrated method to visually servo-control elastic deformations by fully-constrained robotic grippers," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 4457–4462.
- [16] X. Li, X. Su, and Y.-H. Liu, "Vision-based robotic manipulation of flexible PCBs," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 6, pp. 2739–2749, Dec. 2018.
- [17] J. Zhu, D. Navarro-Alarcon, R. Passama, and A. Cherubini, "Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours," *Robot. Auton. Syst.*, vol. 142, 2021, Art. no. 103798.
- [18] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Visual tracking control of deformable objects with a FAT-based controller," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1673–1681, Feb. 2022.
- [19] A. Cherubini and D. Navarro-Alarcon, "Sensor-based control for human-robot collaboration: Fundamentals, challenges and opportunities," *Front Neurorobot.*, vol. 14, 2021, Art. no. 113.
- [20] S. Li, A. Ghasemi, W. Xie, and Y. Gao, "An enhanced IBVS controller of a 6DoF manipulator using hybrid PD-SMC method," *Int. J. Control Automat. Syst.*, vol. 16, no. 3, pp. 844–855, 2018.
- [21] Y. Chang, L. Li, Y. Wang, and K. You, "Toward fast convergence and calibration-free visual servoing control: A new image based uncalibrated finite time control scheme," *IEEE Access*, vol. 8, pp. 88333–88347, 2020.
- [22] J.-J. E. Slotine *et al.*, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991, vol. 199, no. 1.
- [23] B. Ahmadi, W.-F. Xie, and E. Zakeri, "Robust cascade vision/force control of industrial robots utilizing continuous integral sliding mode control method," *IEEE/ASME Trans. Mechatronics*, to be published, doi: 10.1109/TMECH.2021.3067619.
- [24] F. Li and H.-L. Xie, "Sliding mode variable structure control for visual servoing system," *Int. J. Automat. Comput.*, vol. 7, no. 3, pp. 317–323, 2010.
- [25] M. Van, S. S. Ge, and H. Ren, "Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control," *IEEE Trans Cyber.*, vol. 47, no. 7, pp. 1681–1693, Jul. 2017.
- [26] M. M. Polycarpou and P. A. Ioannou, "A robust adaptive nonlinear control design," in *Proc. Amer. Control Conf.*, 1993, pp. 1365–1369.
- [27] H. L. Lei and S. Yang, "Fusion of image contour moments and fourier descriptors for the hand gesture recognition," in *Proc. Adv. Mater. Res.*, vol. 433, 2012, pp. 5188–5192.
- [28] D. Zheng, H. Wang, J. Wang, X. Zhang, and W. Chen, "Toward visibility guaranteed visual servoing control of quadrotor UAVs," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 3, pp. 1087–1095, Jun. 2019.

- [29] V. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Autom. control*, vol. 22, no. 2, pp. 212–222, Apr. 1977.
- [30] J. Ma, S. S. Ge, Z. Zheng, and D. Hu, "Adaptive NN control of a class of nonlinear systems with asymmetric saturation actuators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1532–1538, Jul. 2015.
- [31] Z. Qu, *Robust Control of Nonlinear Uncertain Systems*. Hoboken, NJ, USA: Wiley 1998.
- [32] P. Ioannou and J. Sun, *Robust Adaptive Control*. North Chelmsford, Chelmsford, MA, USA: Courier Corp., 2012.
- [33] L. Yang and J. Yang, "Nonsingular fast terminal sliding-mode control for nonlinear dynamical systems," *Int. J. Robust Nonlinear Control*, vol. 21, no. 16, pp. 1865–1879, 2011.
- [34] C. Xue, X. Yu, W. He, and C. Sun, "Finite-time neural impedance control for an uncertain robotic manipulator," in *Proc. 34rd Youth Academic Annu. Conf. Chin. Assoc. Automat.*, 2019, pp. 42–46.
- [35] Y. Feng, X. Yu, and Z. Man, "Non-singular terminal sliding mode control of rigid manipulators," *Automatica*, vol. 38, no. 12, pp. 2159–2167, 2002.
- [36] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Proc. IEEE/RSJ Int. Conf. Robots Intell. Syst.*, vol. 1, 1994, pp. 186–193.
- [37] J. Qian and J. Su, "Online estimation of image jacobian matrix by kalman-bucy filter for uncalibrated stereo vision feedback," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2002, pp. 562–567.
- [38] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*. Berlin, Germany: Springer, 2013, vol. 12.
- [39] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *Int. J. Control.*, vol. 76, no. 9/10, pp. 924–941, 2003.
- [40] D. Navarro-Alarcon, J. Qi, J. Zhu, and A. Cherubini, "A Lyapunov-stable adaptive method to approximate sensorimotor models for sensor-based control," *Front Neuorobot.*, vol. 14, 2020, Art. no. 59.

**Jiaming Qi** received the M.Sc. degree in integrated circuit engineering from the Harbin Institute of Technology, Harbin, China, in 2018. He is currently working toward the Ph.D. degree in Control Science and Engineering with the Harbin Institute of Technology, Harbin, China.

In 2019, he was a visiting Ph.D. student with The Hong Kong Polytechnic University. His current research interests include robotic manipulation, data-driven control, and visual servoing.

**Guangfu Ma** received the Ph.D. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 1993 and 1987, respectively.

He was with the Harbin Institute of Technology, where he became an Associate Professor in 1992, and a Professor in 1997, where he currently teaches and performs research in optimal control, spacecraft attitude control, and aerospace control systems. He is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology.

**Jihong Zhu** (Member, IEEE) received the M.Sc. degree in systems and control from TU Delft, Delft, Netherlands, in 2015, and the Ph.D. degree in robotics from the University of Montpellier, Montpellier, France, in 2020.

He conducted his Doctoral Research with LIRMM, France, and in 2019, he visited The Hong Kong Polytechnic University. He is currently a Postdoc with Cognitive Robotics department, TU Delft and Honda Research Institute, Europe. His research interests include: robot learning and manipulation of soft objects.

**Peng Zhou** (Graduate Student Member, IEEE) was born in China. He received the M.Sc. degree in software engineering from the School of Software Engineering, Tongji University, Shanghai, China, in 2017. He is currently working toward the Ph.D. degree in robotics with The Hong Kong Polytechnic University, Kowloon, Hong Kong.

His research interests include deformable object manipulation, motion planning, and robot learning.

**Yueyong Lyu** received the bachelor's, master's, and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2002, 2008 and 2013, respectively.

He is currently an Associate Research Fellow with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, China. His interests of research mainly focus on spacecraft guidance, navigation and control, especially in spacecraft formation flying, on-orbit service, etc.

**Haibo Zhang** received the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2013 and 2009, respectively.

He performs research in space manipulator dynamics and compliance capture control, and spacecraft relative motion control. He is currently a Senior Engineer with the Beijing Institute of Control Engineering, Beijing, China.

**David Navarro-Alarcon** (Senior Member, IEEE) received the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), Shatin, Hong Kong, in 2014.

Since 2017, he has been with The Hong Kong Polytechnic University (PolyU), Hung Hom, Hong Kong, where he is an Assistant Professor with the Department of Mechanical Engineering, Principal Investigator with the Robotics and Machine Intelligence Laboratory, and Investigator with the Research Institute for Smart Ageing. Before joining PolyU, he was a Researcher with the CUHK T Stone Robotics Institute, from 2014 to 2017. He has had visiting appointments with the University of Toulon in France and the Technical University of Munich in Germany. His current research interests include perceptual robotics and control theory.