Noname manuscript No.
(will be inserted by the editor)

Towards Latent Space Based Manipulation of Elastic Rods using Autoencoder Models and Robust Centerline Extractions

Jiaming Qi · **Guangfu Ma** ·
Peng Zhou · **Haibo Zhang** · **Yueyong Lyu** ·
David Navarro-Alarcon*

Received: date / Accepted: date

Abstract The automatic shape control of deformable objects is a challenging (and currently hot) manipulation problem due to their high-dimensional geometric features and complex physical properties. In this study, a new methodology to manipulate elastic rods automatically into 2D desired shapes is presented. An efficient vision-based controller that uses a deep autoencoder network is designed to compute a compact representation of the object's infinite-dimensional shape. An online algorithm that approximates the sensorimotor mapping between the robot's configuration and the object's shape features is used to deal with the latter's (typically unknown) mechanical properties. The proposed approach computes the rod's centerline from raw visual data in real-time by introducing an adaptive algorithm on the basis of a self-organizing network. Its effectiveness is thoroughly validated with simulations and experiments.

Keywords Robotics · Visual Servoing · Deformable Objects · Autoencoder · Self-Organizing Network · Model Predictive Control

1 Introduction

Controlling the shape of soft objects automatically with robot manipulators is highly valuable in many applications, such as food processing [25], robotic surgery [1], cable assembly [24], and household works [23]. Although great progress has been achieved in recent years, shape control remains an open problem in robotics [13]. One of the most crucial

J. Qi, G. Ma and Y. Lyu are with the Harbin Institute of Technology, Harbin, 150001, China.

P. Zhou and D. Navarro-Alarcon are with the Hong Kong Polytechnic University, Hung Hom, KLN, Hong Kong. Corresponding author: dna@ieee.org

Haibo Zhang is with the Beijing Institute of Control Engineering, Beijing, 100190, China. National Key Laboratory of Science and Technology on Space Intelligent Control, Beijing, 100190, China.

Jiaming Qi
Harbin Institute of Technology, Harbin, 150001, China.
Tel.: +86-18646086707
E-mail: qijm_hit@163.com

issues that hamper the implementation of these types of controls is the difficulty to obtain a meaningful and efficient feedback representation of the object's configuration in real-time. However, given the intrinsic high-dimensional nature of deformable objects, standard vision-based control algorithms (e.g., based on simple point features) cannot be used as they cannot properly capture the objects' state. In this work, a solution is provided to this problem.

The configuration of rigid objects can be fully described by six degrees of freedom. However, representing the configuration of soft objects is difficult as they have infinite-dimensional geometric information. Therefore, a simple and effective feature extractor that can characterize these objects in an efficient (i.e., compact) manner should be designed [3]. At present, traditional methods are roughly divided into two categories: local and global descriptors. Local descriptors may use centroids, distances, angles, curvatures [14] to describe geometric characteristics. However, these features must be "hard-coded." In turn, they can only provide a fixed type of representation. Global descriptors produce a generic representation of the object's shape. An example method under this category is the Point Feature Histogram (PFH) reported in [21]. PFH forms a multi-dimensional histogram to represent the overall shape of a soft object. Subsequent efforts developed PFH into the Fast Point Feature Histograms (FPFH), which reduces computation time [8, 20]. A method based on linearly parameterized (truncated) Fourier series was also proposed to represent the object's contour [12]. This parameterization idea was generalized in [19], where more shape representations were analyzed and implemented.

Learning-based solutions have received considerable attention due to their potential to learn (in latent space) shape representations of virtually any type of object from data observations only [30]. Force and position measurements of a three-finger gripper manipulating a soft object were used in [3] as input to a network, which produced and predicted the object's contour (even for unknown objects). A coarse-to-fine shape representation was also proposed on the basis of spatial transformer networks, which allowed it to obtain good generalization properties without expensive ground truth observations [29]. Growing neural gas was used in [26] to represent deformable shapes. In [18], a feature extractor based on the convolutional autoencoder was developed. This method was used to obtain a low-dimensional latent space from tactile sensing data.

Traditional methods for manipulating soft objects [7] typically need to identify the complex physical properties of the object. This requirement hinders their application in practice. Algorithms based on latent spaces present a feasible solution, as they can effectively extract low-dimensional features from a high-dimensional shape space. For example, convolutional neural networks were used to build the inverse kinematics of a rope [11], learn the physical model of a soft object in the latent space without any prior knowledge of the object [5], and estimate the rope's state and combine it with model predictive control [29]. However, none of these works has been used to establish an explicitly shaped servo-loop with a latent space representation. This idea has not been sufficiently explored in the soft manipulation literature.

In the current work, a new solution to the manipulation problem of the elastic rod is proposed. The novel contributions of this study are listed as follows.

1. A centerline extraction algorithm based on self-organizing maps (SOM) is presented for slender elastic rods.
2. A shape feature extraction algorithm is designed using the deep autoencoder network (DAE). The proposed method is used to represent the elastic rod with finite-dimensional feature vectors.

3. Detailed simulations and experiments are conducted to validate the effectiveness of the proposed method.

To the best of the authors' knowledge, this work is the first attempt wherein a shape servo-controller uses DAE to establish an *explicit* shape servo-loop. The remainder of this study is organized as follows. The preliminaries are presented in Section 2. The overall deformation control implementation process is discussed in Section 3. Various visually servoed deformation tasks of elastic rods are shown in Sections 4 and 5. Conclusions and future work are provided in Section 6.

2 PRELIMINARIES

Notation. Column vectors are denoted with bold small letters $\mathbf{v}$ and matrices with bold capital letters $\mathbf{M}$. Time evolving variables are represented as $\mathbf{m}_k$, where the subscript k denotes the discrete time instant. $\mathbf{E}_n$ is an $n \times n$ identity matrix.

The deformation control scheme of a robot manipulating the elastic rod based on visual servoing is investigated. The following conditions are provided to foster an understanding among readers:

– A fixed camera is used to measure the centerline of the elastic rod, namely, eye-to-hand configuration (depicted in Fig. 1). The coordinates obtained are denoted by:

$$\bar{\mathbf{c}} = \left[\mathbf{c}_1^T, \ldots, \mathbf{c}_N^T\right]^T \in \mathbb{R}^{2N} \ \mathbf{c}_i = \left[u_i, v_i\right]^T \in \mathbb{R}^2 \tag{1}$$

where N represents the number of points that make up the centerline, u_i and v_i represents the coordinates of the ith $(i = 1, \cdots, N)$ point in the image frame.
– Before the experiment, the robot has tightly grasped the elastic rod; that is, object grasping is not the research field of this article. Measurement loss is also not a problem during the manipulation process.
– The robot supports velocity control mode, which can accurately execute the given desired kinematic commands $\Delta\mathbf{r}_k \in \mathbb{R}^q$ [22] and satisfy the incremental position motions $\mathbf{r}_k = \mathbf{r}_{k-1} + \Delta\mathbf{r}_k$.
– The robot manipulates the elastic rod at low speeds, so the shape is uniquely determined by elastic potential energy.

Problem Statement. Without any prior physical characteristics of elastic rods, design a model-free vision-based controller which commands the robot to deform the elastic rod into the desired shape in the 2D image space.

3 Methods

3.1 Robust SOM-based Centerline Extraction Algorithm

Slender elastic rods whose lengths are much larger than their diameters are used as the research object. Therefore, the centerline describes the shape of the elastic rods. Given that the centerline generally comprises center-points for elastic rods, it should be fixed-length, ordered, and equidistant for subsequent feature extraction and controller design. Although some centerline extraction algorithms are used in the literature, e.g., *OpenCV/thinning*, they

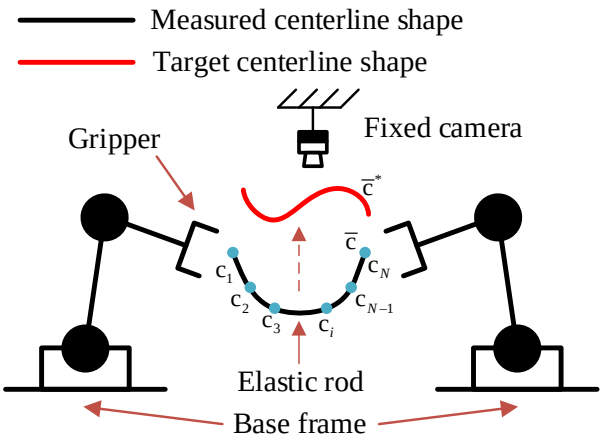


Fig. 1: Schematic diagram of the elastic rod shape deformation. The camera is utilized to determine shape feature $\mathbf{s}$ in real time, and within the designed controller the robot automatically deform the real-time shape denoted by $\bar{\mathbf{c}}$ of elastic rods into the target shape $\bar{\mathbf{c}}^*$.

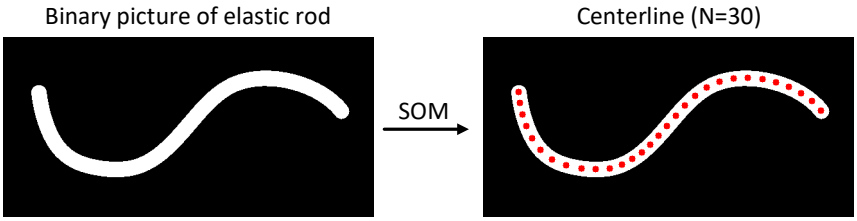


Fig. 2: Schematic diagram of SOM-based centerline extraction. The white area in the left side represents the area of elastic rod (clustering area), and the red points in the right side represent the obtained centerline points (clustering points) (in this figure, $N = 30$).

cannot meet the above requirements and need pre-processing of data, which will deteriorate the system's real-time performance.

In this article, SOM is utilized to achieve real-time 2D centerline extraction of elastic rods without artificial marker points. SOM is a neural network trained in an unsupervised learning manner [10], which is originally used for dimensionality reduction of high-dimensional data. Here, it is used as a clustering algorithm. It generates a fixed number of clustering points from the image data of the elastic rods. Finally, the centerline is composed of the clustering points. The input of SOM is the white area where the elastic rod is located in the binary image, as shown in Fig. 2. The points in the white area are defined by $\bar{\mathbf{m}} = \left[\mathbf{m}_1^T, \ldots, \mathbf{m}_M^T\right]^T \in \mathbf{R}^{2M}$, $\mathbf{m}_i \in \mathbf{R}^2$ represents coordinates of each point in the image frame, and $M \gg N$. With the clustering nature of SOM, a fixed-length equidistant centerline can be obtained, namely, $SOM : 2M \to 2N$.

Remark 1 The proposed SOM-based centerline extraction is only used in the experiment and not for simulation. The centerline extracted by SOM is not guaranteed to be ordered, so the sorting algorithm [19] is utilized to reorder the centerline. This process will not take too much time because N is small.

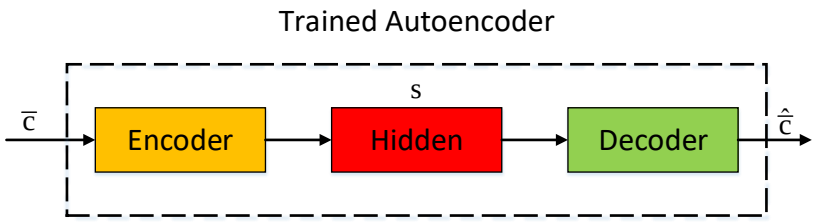


Fig. 3: Structure of DAE with the centerline $\bar{\mathbf{c}}$ as the input, and $\mathbf{s}$ is defined as the shape feature used for deformation Jacobian matrix approximation and controller design.

3.2 Feature Extraction

A controller that can deform the real-time shape $\bar{\mathbf{c}}$ of elastic rods into the target shape $\bar{\mathbf{c}}^*$ can be designed using the centerline extracted by SOM. However, the centerline cannot be directly inputted into the system. Given its high dimensionality, it will make the system run slowly and may even cause many adverse effects, e.g., loss of control. Thus, designing a shape feature extraction algorithm for elastic rods to reduce the feature dimension and represent the centerline effectively is necessary.

In this article, DAE is used to extract shape features $\mathbf{s} \in \mathbb{R}^p$ from the high-dimensional centerline $\bar{\mathbf{c}} \in \mathbb{R}^{2N}$. DAE is an artificial neural network trained in an unsupervised-learning manner, which can automatically learn latent features from unlabeled data [31]. DAE comprises three parts, an Encoder that projects the input into the hidden layer, a hidden layer describing the latent feature $\mathbf{s}$, and a Decoder that reconstructs the latent feature into the original input. Formally, the centerline $\bar{\mathbf{c}} \in \mathbb{R}^{2N}$ is fed into DAE and mapped to the hidden layer through the nonlinear transformation $\mathbf{s} = \mathbf{f}_{\theta_1}(\bar{\mathbf{c}}) = sig(\mathbf{W}_1\bar{\mathbf{c}} + \mathbf{b}_1)$, where parameter set $\theta_1 = \{\mathbf{W}_1, \mathbf{b}_1\}$. $\mathbf{W}_1$ is a $k \times 2N$ weight matrix, $\mathbf{b}_1$ is a vector of bias and sig is a *sigmoid* activation function, $s(\bar{c}) = \frac{1}{1+e^{-\bar{c}}}$. The latent feature $\mathbf{s}$ is input into the Decoder to generate a reconstruction $\hat{\bar{\mathbf{c}}}$ with $2N$ dimensions through the deterministic equation $\hat{\bar{\mathbf{c}}} = \mathbf{g}_{\theta_2}(\mathbf{s}) = sig(\mathbf{W}_2\mathbf{s} + \mathbf{b}_2)$, with $\theta_2 = \{\mathbf{W}_2, \mathbf{b}_2\}$. The parameters of θ_1 and θ_2 of the DAE are designed to minimize the average error of reconstruction, which is defined as:

$$\{\theta_1^*, \theta_2^*\} = \arg\min_{\theta_1,\theta_2} \sum_{k=1}^{N} L\left(\mathbf{c}_i, g_{\theta_2}\left(f_{\theta_1}(\mathbf{c}_i)\right)\right) \tag{2}$$

where θ_1^* and θ_2^* are the ideal parameters, and L is usually a mean square error. Once the Autoencoder is trained, the centerline $\bar{\mathbf{c}}$ is input into the network, and the low-dimensional shape feature $\mathbf{s} \in \mathbb{R}^p$ can be obtained through nonlinear transformations $\mathbf{f}_{\theta_1}$. The workflow of DAE is shown in Fig. 3.

For DAE, the reconstruction output $\hat{\bar{\mathbf{c}}}$ is not the focus, and only the Encoder is utilized to provide the shape feature $\mathbf{s} \in \mathbb{R}^p$ once the DAE is trained. At present, DAE has various forms. In this paper, multilayer perceptron (MLP) is used, given its ability to handle 2D data efficiently. The size of shape feature dimension p can also be selected due to a trade-off balance. A small p will improve the system's controllability, e.g., $p < q$. However, a large p will enhance the representation accuracy of centerlines. In the simulation and experiment, the effect of various p on the shape representation capabilities is presented.

3.3 Approximation of the Local Deformation Model

Given that regular (i.e., mechanically well-behaved) elastic objects are considered, the centerline $\bar{\mathbf{c}}$ is extremely dependent on the robot command $\mathbf{r} \in \mathbb{R}^q$ can be defined as the joint angles or end-effector's pose in this study. The relationship between $\bar{\mathbf{c}}$ and $\mathbf{r}$ can be represented by the following unknown function (3).

$$\bar{\mathbf{c}} = \mathbf{h}(\mathbf{r}) \tag{3}$$

Following (3), the overall kinematics model from robot command $\mathbf{r}$ to shape feature $\bar{\mathbf{c}}$ can be constructed as follows:

$$\mathbf{s} = \mathbf{f}_{\theta_1}(\mathbf{h}(\mathbf{r})) \tag{4}$$

Differentiating (4) concerning time variable t yields:

$$\dot{\mathbf{s}} = \mathbf{J}(t)\dot{\mathbf{r}} \tag{5}$$

where $\mathbf{J}(t) = \partial \mathbf{s}/\partial \mathbf{r} \in \mathbb{R}^{p\times q}$ represents a Jacobian-like matrix that describes the mapping between the feature change speed $\dot{\mathbf{s}}$ and the velocity command $\dot{\mathbf{r}}$. The properties of elastic rods are unknown, so the analytical form of $\mathbf{J}(t)$ cannot be obtained. Discretizing (5) yields the first-order format as follows:

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \mathbf{J}_k \Delta \mathbf{r}_k \tag{6}$$

where $\Delta \mathbf{r}_k = \mathbf{r}_k - \mathbf{r}_{k-1} \in \mathbf{R}^q$. The application of DAE as feature extraction is the focus of this study. Accordingly, the simple Broyden algorithms are used to computes local approximations of $\mathbf{J}_k$ in real-time. Define the following differential signal:

$$\mathbf{y}_k = \mathbf{s}_k - \mathbf{s}_{k-1}\ \ \mathbf{u}_k = \Delta \mathbf{r}_k = \mathbf{r}_k - \mathbf{r}_{k-1} \tag{7}$$

Broyden algorithms are as follows:

1. R1 update formula [2]:
$$\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} + \frac{\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)\mathbf{u}_k^T}{\mathbf{u}_k^T\mathbf{u}_k} \tag{8}$$
This form has a simple structure and fast calculation speed.
2. SR1 update formula [2]:
$$\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} + \frac{\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)^T}{\mathbf{u}_k\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)^T} \tag{9}$$
The structure of SR1 is similar to R1, but the calculation accuracy is higher.
3. DFP update formula [15]:
$$\begin{aligned}\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} &+ \frac{\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)\mathbf{y}_k^T + \mathbf{y}_k\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)^T}{\mathbf{u}_k\mathbf{y}_k^T} \\ &- \frac{\mathbf{y}_k^T\mathbf{y}_k}{\left\|\mathbf{u}_k\mathbf{y}_k^T\right\|}\left(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k\right)\mathbf{u}_k^T\end{aligned} \tag{10}$$
DFP is a rank two quasi-Newton method, which is efficient for solving nonlinear optimization.

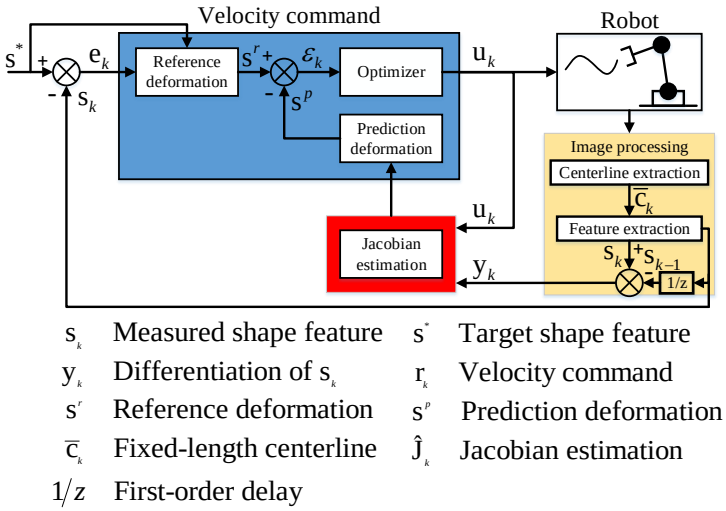


Fig. 4: The block diagram of the proposed real-time deformation control strategy.

4. BFGS update formula [4]:

$$\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} - \frac{\hat{\mathbf{J}}_{k-1}\mathbf{u}_k\mathbf{u}_k^T\hat{\mathbf{J}}_{k-1}^T}{\mathbf{u}_k\mathbf{u}_k^T\hat{\mathbf{J}}_{k-1}^T} + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{u}_k\mathbf{y}_k^T} \tag{11}$$

It is recognized with the best numerical stability.

When a new data pair $(\mathbf{y}_k, \mathbf{u}_k)$ enters the system, the deformation Jacobian matrix $\hat{\mathbf{J}}_k$ can be updated with the above estimators.

Remark 2 The robot is assumed to manipulate elastic rods at low speed. Thus, the deformation of the elastic rods is relatively slow. On the basis, the deformation Jacobian matrix $\mathbf{J}_k$ can be estimated online as the formula (4) is assumed to be smooth.

3.4 Shape Servoing Controller

At the discrete-time instant k, the deformation Jacobian matrix $\hat{\mathbf{J}}_k$ has been assumed to be exactly approximated by (8)(9)(10)(11), so that the shape-motion difference model is satisfied:

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \hat{\mathbf{J}}_k \cdot \mathbf{u}_k \tag{12}$$

A model predictive controller [16] is utilized to minimize the shape deformation error $\mathbf{e}_k = \mathbf{s}^* - \mathbf{s}_k$ between the measured feature $\mathbf{s}_k$ and a constant target feature $\mathbf{s}^*$. With the estimated deformation Jacobian matrix $\hat{\mathbf{J}}_k$ and (12), the predicted deformation output at time $k+w$ is shown below:

$$s_{k+w}^p = s_k + \hat{\mathbf{J}}_k \cdot \mathbf{u}_{k+w} \tag{13}$$

where $w \in [0,H]$ represents the length of prediction horizon, and $\mathbf{u}_{k+w} = \mathbf{r}_{k+w} - \mathbf{r}_k$. The reference deformation trajectory at time $k+w$ is calculated to ensure smooth deformation of elastic rods and the estimation accuracy of deformation Jacobian matrix as follows:

$$\mathbf{s}_{k+w}^r = \mathbf{s}^* - e^{-\rho w} \cdot \mathbf{e}_k \tag{14}$$

where ρ is a positive constant. Error ε between the reference and the prediction deformation at instant $k+w$ is defined as follows:

$$\varepsilon_{k+w} = \mathbf{s}_{k+w}^r - \mathbf{s}_{k+w}^p = \left(1 - e^{-\rho w}\right) \mathbf{e}_k - \hat{\mathbf{J}}_k \mathbf{u}_{k+w} \tag{15}$$

Velocity command $\mathbf{u}_k$ is assumed to remain constant from k to $k+w$ and can be calculated by minimizing ε from k to $k+H$, as shown below:

$$\min \frac{1}{2} \left(\sum_{w=0}^{H} \alpha^w \left\| \left(1 - e^{-\rho w}\right) \mathbf{e}_k - w \hat{\mathbf{J}}_k \mathbf{u}_k \right\|^2 + \mathbf{u}_k^T \mathbf{Q} \mathbf{u}_k \right) \tag{16}$$

where $0 < \alpha \leq 1$, and $\mathbf{Q}$ is a symmetric and positive definite matrix used to adjust the command $\mathbf{u}_k$. When the command $\mathbf{u}_k$ is too large, it will cause the robot to move too fast and the manipulated object will oscillate. In turn, the estimation accuracy of the deformation Jacobian matrix will be affected. Taking derivative of (16) with respect to $\mathbf{u}_k$, the gradient ∇ is calculated as follows:

$$\nabla = \sum_{w=0}^{H} -w \alpha^w \hat{\mathbf{J}}_k^T \left(\left(1 - \beta^w\right) \mathbf{e}_k - w \hat{\mathbf{J}}_k \mathbf{u}_k \right) + \mathbf{Q} \mathbf{u}_k \tag{17}$$

where $\beta = e^{-\rho}$. By setting $\nabla = 0$, the velocity command $\mathbf{u}_k$ is derived:

$$\begin{aligned} \mathbf{u}_k &= \left(a \hat{\mathbf{J}}_k + \hat{\mathbf{J}}_k^{T+} \mathbf{Q}\right)^+ (b - c) \mathbf{e}_k \\ a &= \left(H^2 \alpha^H - 2b\right) / \ln \alpha \\ b &= \left(H \alpha^H \ln \alpha - \alpha^H + 1\right) / \ln^2 \alpha \\ c &= \left(H (\alpha\beta)^H \ln (\alpha\beta) - (\alpha\beta)^H + 1\right) / \ln^2 (\alpha\beta) \end{aligned} \tag{18}$$

Thus, at each time instant, the incremental position command is calculated as follows:

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{u}_k \tag{19}$$

Following (12), it yields:

$$\mathbf{e}_k - \mathbf{e}_{k-1} = -\hat{\mathbf{J}}_k \Delta \mathbf{r}_k \tag{20}$$

$\hat{\mathbf{J}}_k$ is assumed to be a full column rank, and substituting (18) into (20) yields:

$$\left(a \mathbf{E}_n + \hat{\mathbf{J}}_k^{T+} \mathbf{Q} \hat{\mathbf{J}}_k^+\right) (\mathbf{e}_k - \mathbf{e}_{k-1}) + (b - c) \mathbf{e}_k = 0 \tag{21}$$

As $a > 0, b - c > 0$, and $\hat{\mathbf{J}}_k^{T+} \mathbf{Q} \hat{\mathbf{J}}_k^+$ is a positive-definite matrix, the error $\mathbf{e}_k$ asymptotically converges to error, namely, $\lim_{t \to \infty} \mathbf{s}_k = \mathbf{s}_k^*$. However, when the reachability of the desired goal $\mathbf{s}_k^*$ is not satisfied, $\hat{\mathbf{J}}_k$ may not be a column full-rank matrix. The feedback error $\|\mathbf{e}_k\|$ may only converge to the neighborhood near the origin. For such under-actuated visual servo control tasks, guaranteeing the global asymptotic convergence is challenging [9]. The block diagram of the proposed real-time deformation control strategy is shown in Fig. 4.

Remark 3 The velocity controller (18) and deformation Jacobian estimators (8)(9)(10)(11) only require visual feedback data without any additional sensors, prior knowledge of the system model, and the requirement to calibrate the camera.

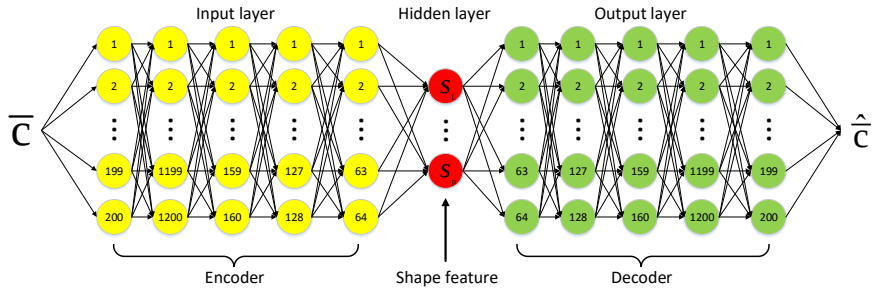


Fig. 5: Structure of DAE comprised of MLPs as the basic blocks of Encoder and Decoder. The centerline $\bar{\mathbf{c}}$ is fed into the trained DAE to generate shape feature denoted by $\mathbf{s}$.

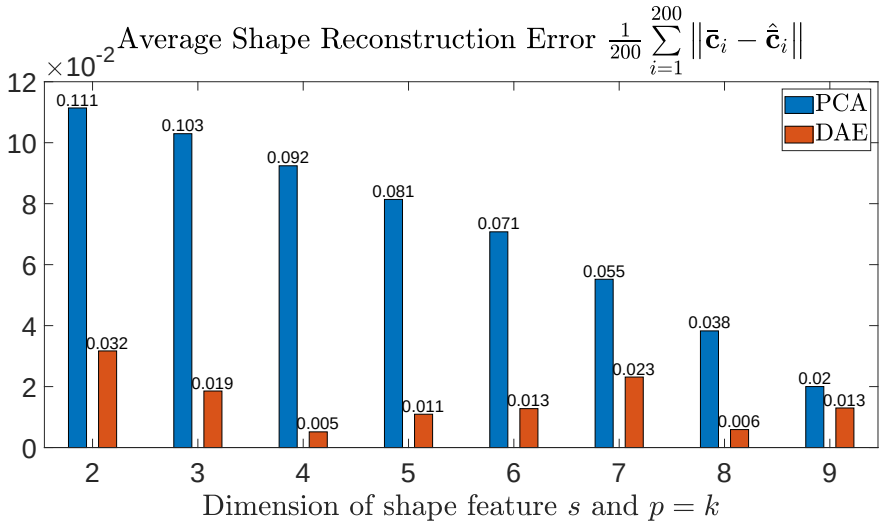


Fig. 6: Average shape reconstruction error comparison between DAE and PCA among 200 shape sets in the simulation.

4 SIMULATION RESULTS

The following case is considered: one end of an elastic rod is rigidly grasped by a planar robot (2DOF) and the other end is static. For brevity, the robot is not shown in the figures. The cable simulator is simulated as in [28] by using the minimum energy principle [6], and publicly available at https://github.com/q546163199/shape_deformation/tree/master/python/package/shape_simulator. All numerical simulations are implemented in Python.

4.1 Feature Extraction Comparison

In this section, 40,000 samples ($N = 100$) utilized to train the DAE are generated by randomly moving the robot. As previously mentioned in Section 3.2, DAE comprises MLPs, as shown in Fig. 5. DAE is implemented on PyTorch and trained by adopting an ADAM optimizer with an initial learning rate of 0.001 and a batch size of 500. RELU activation functions are adopted in the Encoder and Decoder.

In Fig. 6, the reconstruction error between the simulated shape $\bar{\mathbf{c}}$ and the shape $\hat{\bar{\mathbf{c}}}$ obtained from DAE or PCA is denoted by $\left\|\bar{\mathbf{c}} - \hat{\bar{\mathbf{c}}}\right\|$. k and p determine the dimension of shape feature $\mathbf{s}$ obtained from PCA [32] and DAE, respectively. For the fairness of competition, k is set to be equal to p for validating the feature extraction and reconstruction performance of PCA and DAE under the same shape feature dimension. The result shows that in each case, the reconstruction performance of DAE is better than that of PCA. For DAE, the results show that $p = 4$ has the best reconstruction performance, followed by $p = 6$ and $p = 2$. This finding indicates that p is too low to represent the elastic rod fully. Considering the trade balance of system controllability and shape representation performance, DAE with $p = 4$ is used in the following sections.

4.2 Validation of the Jacobian Estimation

In this section, four deformation Jacobian estimators, namely, R1, SR1, DFP, and BFGS, are considered, and their effectiveness is evaluated. The planar robot grasps one end of the simulated rod and conducts a counterclockwise circular motion with center (0.4, 0.4), as shown in Fig. 7a. Different from the simple initialization of $\hat{\mathbf{J}}_0$, e.g., the identity matrix, the robot moves in the initial sampling area (the motions are ensured not to be collinear and close to the starting point) to initialize $\hat{\mathbf{J}}_0$. The initialization accuracy of deformation Jacobian matrix is improved using this method. This method can also reduce the possibility of singular problems under the deformation Jacobian matrix in the manipulation process. In turn, the safety of operations is enhanced. Two error criteria (22) are utilized to compare each deformation Jacobian estimator qualitatively.

$$T_1 = \|\mathbf{s}_k - \hat{\mathbf{s}}_k\| \quad T_2 = \left\|\Delta\mathbf{s}_k - \hat{\mathbf{J}}_k \Delta\mathbf{r}_k\right\| \tag{22}$$

where $\mathbf{s}_k$ is feedback shape feature generated by DAE, $\hat{\mathbf{s}}_k$ is calculated by (23).

$$\hat{\mathbf{s}}_k = \hat{\mathbf{s}}_{k-1} + \hat{\mathbf{J}}_k \Delta\mathbf{r}_k \tag{23}$$

The deformation Jacobian estimators and the shape reconstruction accuracy of DAE are verified, as depicted in Fig. 7b. The results show that the shape reconstruction accuracy of DAE ($p = 4$) is well, proving the effectiveness of DAE in the shape representation. The plots of T_1 and T_2 during the circular motion are demonstrated in Fig. 8. For the T_1 curve, all the four deformation Jacobian estimators can accurately update the deformation Jacobian matrix $\hat{\mathbf{J}}_k$, and the average error of BFGS is the smallest. For T_2, BFGS has no apparent fluctuations, and the estimation accuracy is the best. DFP is second-best, and R1 and SR1 share a similar pattern, consistent with the theoretical analysis. The above analyses prove that, when starting deformation, BFGS can calibrate and update the deformation Jacobian matrix in time to identify the pseudo-physical parameters of the elastic rods. Specifically, BFGS can estimate the change direction of shape feature $\mathbf{s}$ in the latent space.

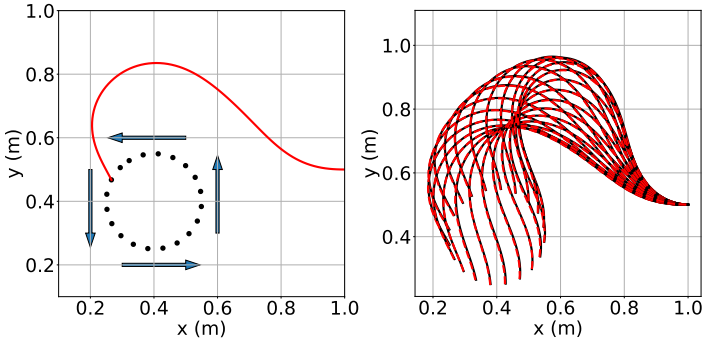


Fig. 7: Deformation Jacobian matrix $\hat{\mathbf{J}}_k$ validation framework. (a) Motion trajectory of robot's end-effector. (b) Comparison between the simulated cable profile (black solid line) and its reconstruction shape obtained by DAE (red dashed line) with $p = 4$.

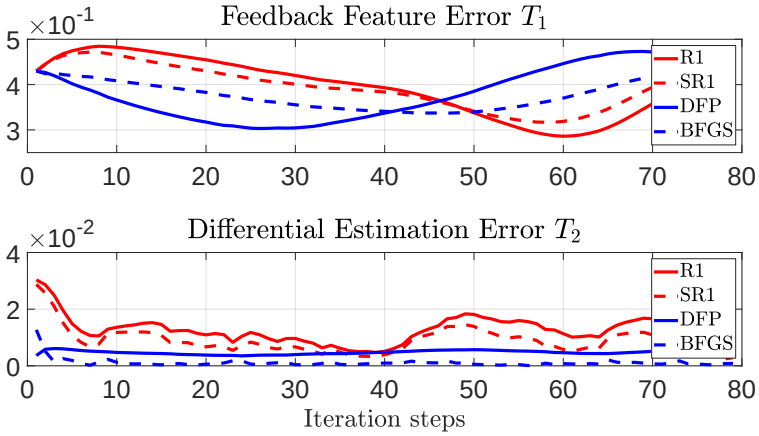


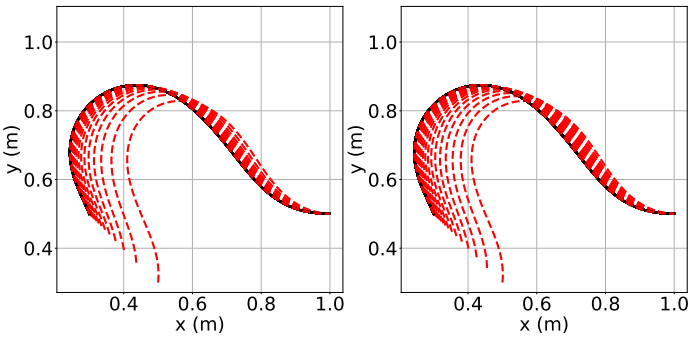
Fig. 8: Profiles of the criteria T_1 and T_2 that are computed along the circular trajectory around the center (0.4, 0.4).

4.3 Manipulation of Elastic Rods

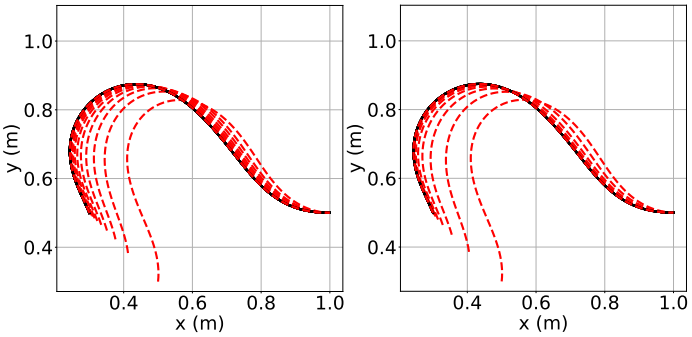
In this section, the robot is commanded by the velocity controller (18) to deform the elastic rods into the desired constant shape $\bar{\mathbf{c}}^*$, corresponding to $\mathbf{s}^*$. The error criterion (24) is utilized to assess the deformation performance.

$$T_3 = \|\bar{\mathbf{c}}_k - \bar{\mathbf{c}}_k^*\| \tag{24}$$

The progress of the cable deformation under the velocity command (18) on the basis of R1, SR1, DFP and BFGS is depicted in Fig. 9. The curve of T_3 and the velocity command $\Delta\mathbf{r}_k$ are shown in Fig 10, and the detailed time comparison is provided in Table 1. Both figures show that BFGS is the best method with the shortest convergence time and smallest deformation error, followed by DFP, and the effects of R1 and SR1 are similar.



(a) R1 result (b) SR1 result



(c) DFP result (d) BFGS result

Fig. 9: Profiles of of the shape deformation simulation among R1, SR1, DFP and BFGS (red dashed curves represent the initial and transitional trajectories, and the black solid curve represents the target shape $\bar{\mathbf{c}}^*$ with shape feature $\mathbf{s}^*$).

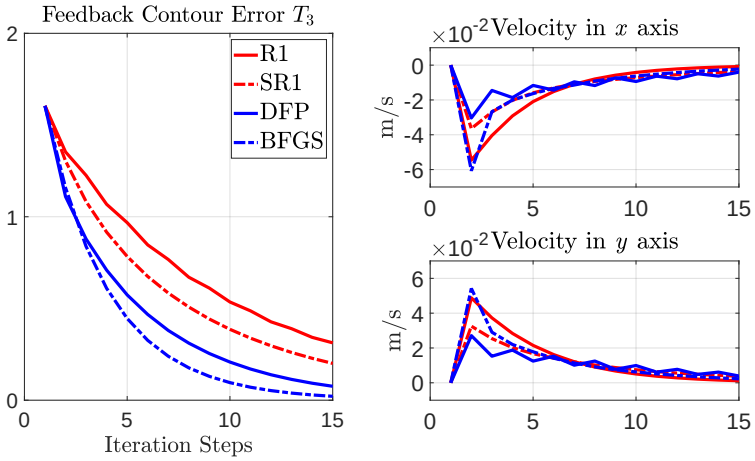


Fig. 10: Profiles of the criterion T_3 and velocity command $\Delta\mathbf{r}_k$ among R1, SR1, DFP and BFGS within manipulation.

Table 1: Results among R1, SR1, DFP and BFGS

	R1	SR1	DFP	BFGS
Steps	58	48	32	22
Time (second)	33.64	27.84	18.56	12.76

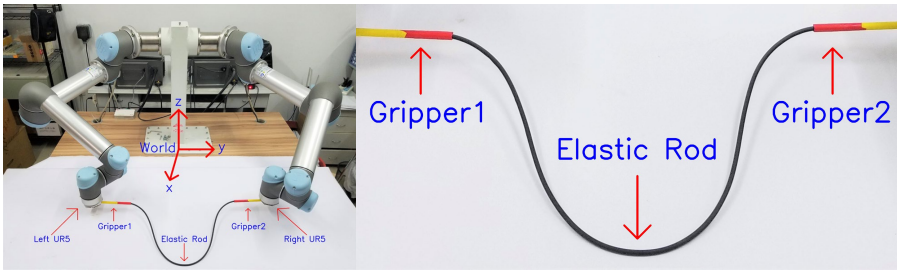


Fig. 11: Experimental setup comprised of two UR5 which support velocity control mode.

5 EXPERIMENTAL RESULTS

Various experiments with two UR5 that support velocity control mode are conducted, as shown in Fig. 11. $\Delta \mathbf{r} = [\Delta \mathbf{r}_1^T, \Delta \mathbf{r}_2^T]^T \in \mathbb{R}^6$. Δr_{i1} and Δr_{i2}, $i = 1,2$ represent the linear velocity of end-effector along x-axis and y-axis of each UR5 in the world frame. Δr_{i3}, $i = 1,2$ represents the angular velocity of the sixth joint of each UR5 along the direction parallel to the z axis in the world frame. A Logitech C270 camera is used to capture the rod's image and combined with OpenCV to process on the Linux PC at 30 fps. The deformation trajectories display once every two frames to compare the convergence effects of each algorithm. An experimental video can be downloaded here https://github.com/q546163199/experiment_video/raw/master/paper2/video.mp4.

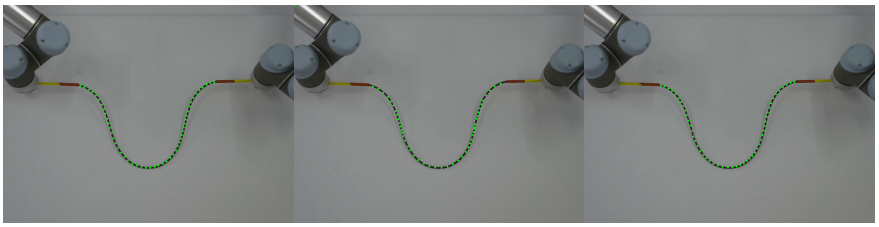
Table 2: Comparison results among three centerline extraction algorithms with $N = 50$

	Reference [19]	CL [17]	SOM
Time (Second)	1.68	0.98	0.38

5.1 Image Processing

This section verifies the proposed SOM-based centerline extraction algorithm and describes the image processing steps.

First, the SOM-based method proposed in this article is compared with two other centerline extraction algorithms. The first one is based on the *OpenCV/thinning* developed in Reference [19], and the second one is based on CL, which is the traditional clustering method [17]. For the fairness of competition, all algorithms need to provide an ordered, fixed-length $N = 50$, equidistant centerline. As the CL-based and SOM-based algorithms



(a) Reference [19] (b) CL [17] (c) Proposed SOM

Fig. 12: Comparison of three centerline extraction algorithms, including reference [19], CL-based [17] and the proposed SOM-based.

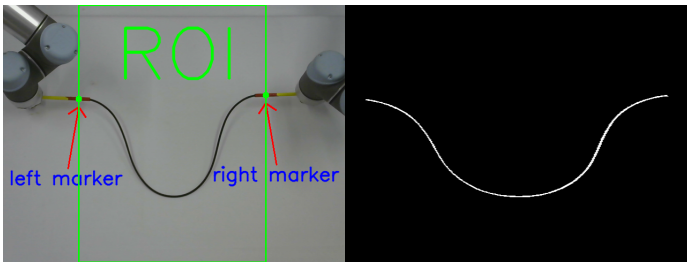
only generate an unordered fixed-length centerline, the sorting algorithm [19] is used to sort unordered centerlines. For SOM, an open-source toolbox provided by [27] is utilized. The SOM-based algorithm is the fastest, and it can efficiently perform centerline extraction, as shown in Table 2. One reason is that the provided SOM toolbox is already highly optimized. Another reason is that the centerline produced by CL-based and SOM-based clustering algorithms has the advantage of fixed-length and equidistant-sampling. This method saves more time than [19], which sorts all the points and then perform down-sampling. The proposed SOM-based algorithm and [19] have similar centerline extraction accuracy, and CL-based has the worst performance, as shown in Fig. 12. Thus, in terms of accuracy, considering that the processing speed of SOM is the fastest, the SOM-based centerline extraction algorithm is used.

Second, the relevant image processing for centerline extraction is provided. The overall process (shown in Fig. 13) is as follows:

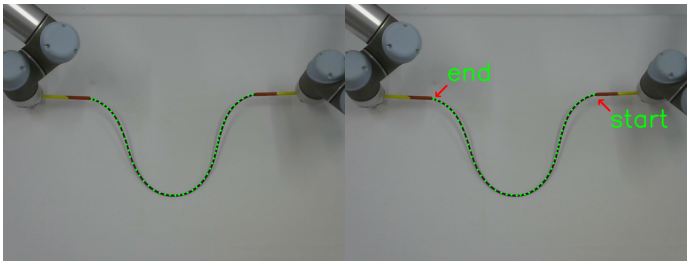
1. First, segment the red areas nearby the Gripper1 and Gripper2 on the basis of HSV color space and mark them as two green marker points. Then, segment the region of the interest (ROI) containing the rod following both green marker points (see Fig. 13a).
2. Next, identify the rod in ROI, remove the noise, and obtain a binary image of the rod using OpenCV morphological opening algorithm (see Fig. 13b).
3. Subsequently, use the proposed SOM-based algorithm to get an unordered centerline with $N = 50$ (see Fig. 13c).
4. Finally, apply the sorting algorithm [19] to get an ordered centerline (see Fig. 13d). The starting point is the closest point to the right marker point on the centerline.

5.2 Feature Extraction Comparison

Similar to Section 4.1, 40,000 samples are generated in the same way. The structure of DAE is similar with Section 4.1, as shown in Fig. 5. Batch-Normalization-1D (BN) is added after each layer. From the comparison results shown in Fig. 14, the reconstruction accuracy of DAE is still better than PCA. For DAE, the accuracy of $p = 4$ is the best, and $p = 5$ is the second best. This finding is consistent with the simulation results. The results prove the effectiveness of the proposed DAE in the shape feature extraction. Thus DAE with $p = 4$ is used in the following sections.



(a) ROI selection (b) Thresholding



(c) Centerline extraction (SOM) (d) Coordinate sorting

Fig. 13: Image processing steps.

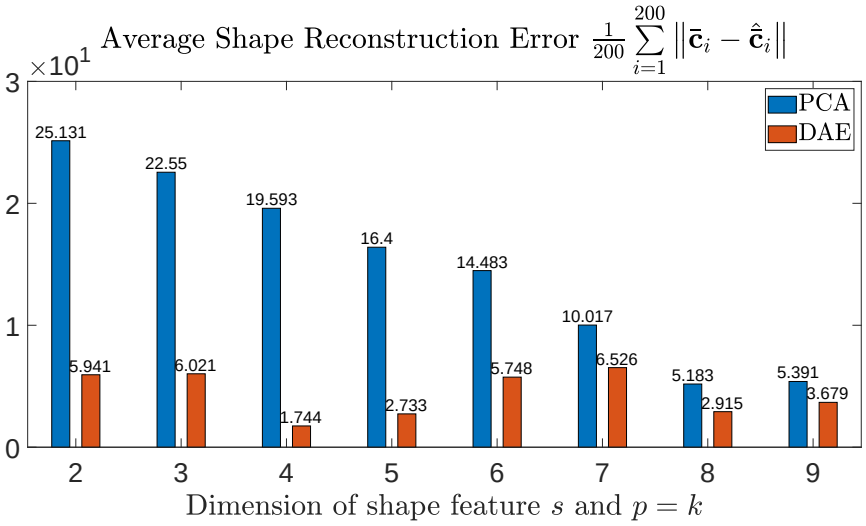
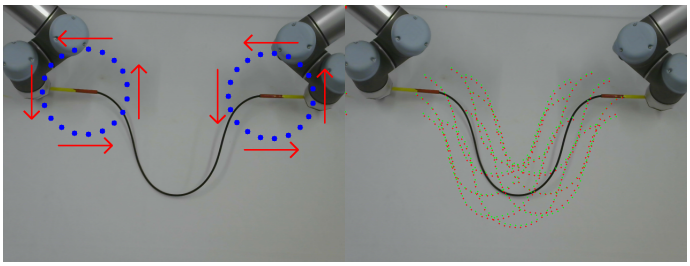


Fig. 14: Average shape reconstruction error comparison between DAE and PCA among 200 shape sets in the experiment.



(a) (b)

Fig. 15: Deformation Jacobian matrix $\hat{\mathbf{J}}_k$ validation framework. (a) Motion trajectory of robot's end-effector. (b) Comparison between the visually measured cable profile (green dot line) and its reconstruction shape obtained by DAE (red dot line) with $p = 4$.

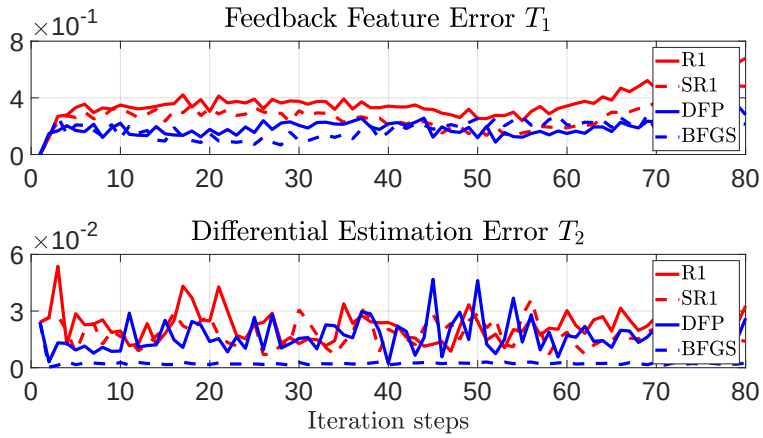


Fig. 16: Profiles of the criteria T_1 and T_2 that are computed along the circular trajectory.

5.3 Validation of the Jacobian Estimation

Similar to Section 4.2, two UR5 are commanded to move along a fixed circular trajectory, as depicted in Fig. 15a. The shape reconstruction performance of DAE is accurate in the experiment, as shown in Fig. 15b. Thus, whether for simulation or experiment, DAE can be applied in the shape representation. The comparison results depicted in Fig. 16 show that BFGS has a minimal deformation Jacobian approximation error, which validates its effectiveness and adaptability in different regions.

5.4 Manipulation of Elastic Rods

Similar to Section 4.3, a desired shape $\bar{\mathbf{c}}^*$ should be given in advance. The following steps are conducted to obtain the feasible target shape.

– First, the robot is moved to a position while avoiding the singular shapes, e.g., straight.

- Second, the current shape of the elastic rod is recorded by the camera and denoted by $\bar{\mathbf{c}}^*$ as the target shape.
- Third, the target shape $\bar{\mathbf{c}}^*$ is fed into the trained DAE to get target shape feature denoted by $\mathbf{s}^*$.
- Fourth, the robot moves back to the initial position and starts the deformation with the given $\mathbf{s}^*$.

Considering safety, the saturation of $\Delta\mathbf{r}$ is set to, $|\Delta r_{i1}| \leq 0.01m/s$, $|\Delta r_{i2}| \leq 0.01m/s$ and $|\Delta r_{i3}| \leq 0.1rad/s, i = 1,2,3$, respectively. Four experiments with different initial and target shapes are conducted to verify the proposed algorithm's effectiveness.

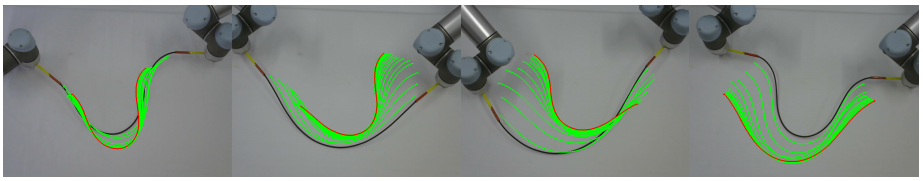
In the proposed algorithm, the elastic rod can be deformed using two UR5 to the desired shape accurately without damaging the object during the deformation process, as depicted in Fig 17. The profiles of T_3 and the velocity command $\Delta\mathbf{r}_k$ are shown in Fig. 18. Corresponding to the simulation results, BFGS still has the most significant control effect, the fastest convergence speed, without apparent fluctuations (large instantaneous deformation). Thus, BFGS has excellent adaptability and robustness to various conditions in the shape deformation issue. In this study, the experiment uses two UR5. Overall, the results prove the effectiveness of the proposed algorithm for the multi-manipulator shape deformation issue.

6 Conclusions

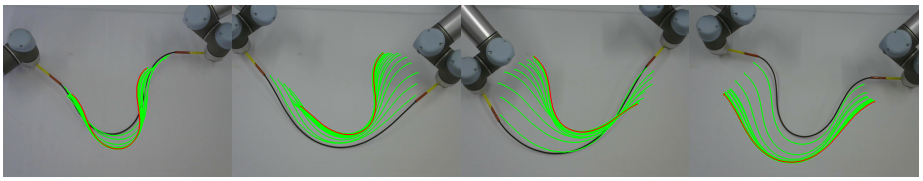
A framework for the deformation control of elastic rods is proposed without any prior physical knowledge. It includes shape feature extraction, deformation Jacobian matrix estimation, and a robust SOM-based centerline extraction algorithm. First, new shape features based on DAE are utilized to represent the elastic rod's centerline in the low-dimensional latent space. Second, the performance of four deformation Jacobian estimators (R1, SR1, DFP, and BFGS) is evaluated. Third, the velocity controller is derived and the system stability is proven. Finally, the effectiveness and feasibility of the proposed algorithm are validated by the numerical and experimental results.

DAE is used in this study to map the high-dimensional geometric information of elastic rods flexibly into a low-dimension latent space. The proposed feature extraction algorithm has better shape representation capabilities than the traditional PCA. It also does not require any artificial markers, making it widely applicable to practical situations. Broyden algorithms are used to approximate the deformation Jacobian matrix in real-time. In this way, the physical parameters and camera models are not identified. From the results, BFGS has the advantages of simple structure, fast calculation speed, and accurate approximation performance. Simultaneously, a robust SOM-based centerline extraction algorithm with a fast calculation speed and high extraction accuracy is designed. The overall system is completely calculated from the visual feedback data, without any prior physical characteristics of the elastic rod and the requirement to calibrate the camera.

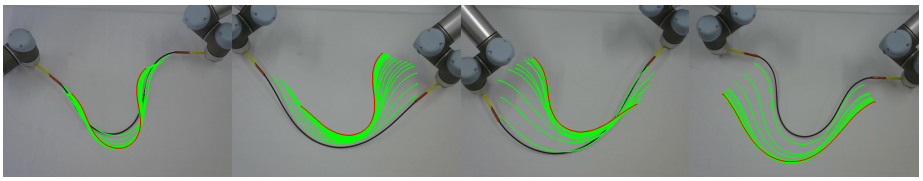
The proposed method also has some limitations. First, the manipulated object is only soft elastic objects, e.g., carbon fiber rod. Thus the proposed algorithm is not suitable for inelastic items, e.g., plasticine and rope. Second, although DAE has a good shape representation ability, it needs an extensive and rich-enough dataset to train itself, which has particular difficulties in practical applications. Third, the approximation of deformation Jacobian matrix based on Broyden algorithms is easy to fall into the local optimum, which may generate the destructive operation, such as over-tension and over-compression in the manipulation process.



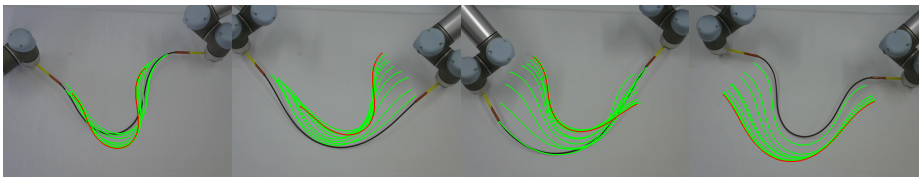
(a) Experiment1-R1 (b) Experiment2-R1 (c) Experiment3-R1 (d) Experiment4-R1



(e) Experiment1-SR1 (f) Experiment2-SR1 (g) Experiment3-SR1 (h) Experiment4-SR1



(i) Experiment1-DFP (j) Experiment2-DFP (k) Experiment3-DFP (l) Experiment4-DFP

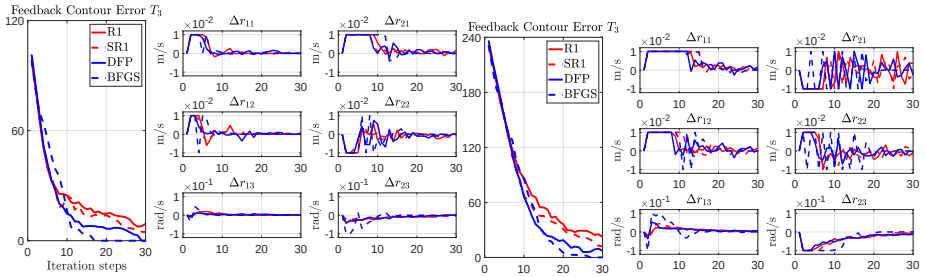


(m) Experiment1-BFGS (n) Experiment2-BFGS (o) Experiment3-BFGS (p) Experiment4-BFGS

Fig. 17: Initial (black solid line), transition (green solid line) and target (red solid line) configurations in the four shape deformation experiments which have a variety of different initial and target shape with dual-UR5 robot among R1, SR1, DFP and BFGS.

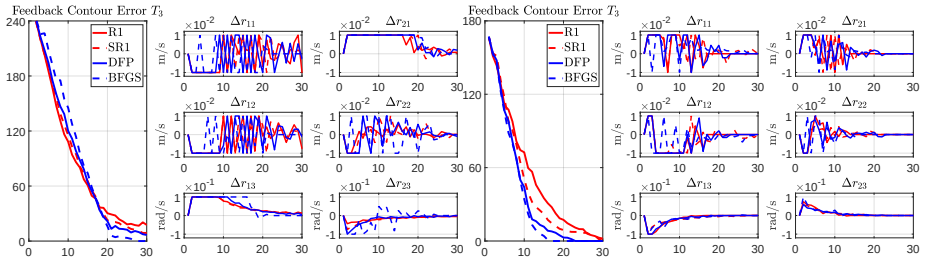
In the future, 3D deformation tasks will be included to manipulate more complex shapes, e.g., M-shaped and spiral. Moreover, the existing DAE needs to be improved to be suitable for different scenarios and materials. Path planning should be considered to avoid possible destructive operations during the manipulation process.

Acknowledgements This work was supported in part by the Germany/Hong Kong Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the German Academic Exchange Service under grant G-PolyU507/18, in part by the Research Grants Council of Hong Kong under grant number 14203917, in part by the Key-Area Research and Development Program of Guangdong Province 2020 under project 76.



(a) Experiment1 result

(b) Experiment2 result



(c) Experiment3 result

(d) Experiment4 result

Fig. 18: Profiles of the criterion T_3 and velocity command $\Delta \mathbf{r}_k$ among R1, SR1, DFP and BFGS within four shape deformation experiments.

References

1. Abolmaesumi, P., Salcudean, S.E., Zhu, W.H., Sirouspour, M.R., DiMaio, S.P.: Image-guided control of a robot for medical ultrasound. IEEE Transactions on Robotics and Automation **18**(1), 11–23 (2002)
2. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. Mathematics of computation **19**(92), 577–593 (1965)
3. Cretu, A.M., Payeur, P., Petriu, E.M.: Soft object deformation monitoring and learning for model-based robotic hand manipulation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **42**(3), 740–753 (2011)
4. Dennis, J.E., Moré, J.J.: A characterization of superlinear convergence and its application to quasi-newton methods. Mathematics of computation **28**(126), 549–560 (1974)
5. Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., Levine, S.: Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. arXiv preprint arXiv:1812.00568 (2018)
6. Hamill, P.: A student's guide to Lagrangians and Hamiltonians. Cambridge University Press (2014)
7. Henrich, D., Wörn, H.: Robot Manipulation of Deformable Objects (2000)
8. Hu, Z., Han, T., Sun, P., Pan, J., Manocha, D.: 3-d deformable object manipulation using deep neural networks. IEEE Robotics and Automation Letters **4**(4), 4255–4261 (2019)
9. Hutchinson, S., Chaumette, F.: Visual servo control, part i: Basic approaches. IEEE Robotics and Automation Magazine **13**(4), 82–90 (2006)
10. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological cybernetics **43**(1), 59–69 (1982)
11. Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., Levine, S.: Combining self-supervised learning and imitation for vision-based rope manipulation. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2146–2153. IEEE (2017)
12. Navarro-Alarcon, D., Liu, Y.H.: Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-d image contours. IEEE Transactions on Robotics **34**(1), 272–279 (2017)
13. Navarro-Alarcon, D., Liu, Y.h., Romero, J.G., Li, P.: On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments. The International Journal of Robotics Research **33**(11), 1462–1480 (2014)

14. Navarro-Alarcon, D., Yip, H.M., Wang, Z., Liu, Y.H., Zhong, F., Zhang, T., Li, P.: Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model. IEEE Transactions on Robotics **32**(2), 429–441 (2016)
15. Nocedal, J.: Updating quasi-newton matrices with limited storage. Mathematics of computation **35**(151), 773–782 (1980)
16. Ouyang, B., Mo, H., Chen, H., Liu, Y., Sun, D.: Robust model-predictive deformation control of a soft object by using a flexible continuum robot. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 613–618. IEEE (2018)
17. Park, D.C.: Centroid neural network for unsupervised competitive learning. IEEE Transactions on Neural Networks **11**(2), 520–528 (2000)
18. Polic, M., Krajacic, I., Lepora, N., Orsag, M.: Convolutional autoencoder for feature extraction in tactile sensing. IEEE Robotics and Automation Letters **4**(4), 3671–3678 (2019)
19. Qi, J., Ma, W., Navarro-Alarcon, D., Gao, H., Ma, G.: Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls. arXiv preprint arXiv:2008.06896 (2020)
20. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217 (2009). DOI 10.1109/ROBOT.2009.5152473
21. Rusu, R.B., Marton, Z.C., Blodow, N., Beetz, M.: Persistent point feature histograms for 3d point clouds. In: Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany, pp. 119–128 (2008)
22. Siciliano, B.: Kinematic control of redundant robot manipulators: A tutorial. Journal of intelligent and robotic systems **3**(3), 201–212 (1990)
23. Sun, P., Hu, Z., Pan, J.: A general robotic framework for automated cloth assembly. In: 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 47–52. IEEE (2019)
24. Tang, T., Wang, C., Tomizuka, M.: A framework for manipulating deformable linear objects by coherent point drift. IEEE Robotics and Automation Letters **3**(4), 3426–3433 (2018)
25. Tokumoto, S., Hirai, S.: Deformation control of rheological food dough using a forming process model. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 2, pp. 1457–1464. IEEE (2002)
26. Valencia, A.J., Nadon, F., Payeur, P.: Toward real-time 3d shape tracking of deformable objects for robotic manipulation and shape control. In: 2019 IEEE SENSORS, pp. 1–4. IEEE (2019)
27. Vettigli, G.: Minisom: minimalistic and numpy-based implementation of the self organizing map. GitHub.[Online]. Available: https://github.com/JustGlowing/minisom/
28. Wakamatsu, H., Hirai, S., Iwata, K.: Modeling of linear objects considering bend, twist, and extensional deformations. In: Proceedings of 1995 IEEE International Conference on Robotics and Automation, vol. 1, pp. 433–438. IEEE (1995)
29. Yan, M., Zhu, Y., Jin, N., Bohg, J.: Self-supervised learning of state estimation for manipulating deformable linear objects. IEEE Robotics and Automation Letters **5**(2), 2372–2379 (2020)
30. Yeo, N., Lee, K., Venkatesh, Y., Ong, S.H.: Colour image segmentation using the self-organizing map and adaptive resonance theory. Image and Vision Computing **23**(12), 1060–1079 (2005)
31. Zhou, P., Zhu, J., Huo, S., Navarro-Alarcon, D.: Lasesom: A latent representation framework for semantic soft object manipulation. arXiv preprint arXiv:2012.05412 (2020)
32. Zhu, J., Navarro-Alarcon, D., Passama, R., Cherubini, A.: Vision-based manipulation of deformable and rigid objects using subspace projections of 2d contours. Rob. Auton. Syst. (2020)