

Towards latent space based manipulation of elastic rods using autoencoder models and robust centerline extractions

Jiaming Qi ^a, Guangfu Ma^a, Peng Zhou ^b, Haibo Zhang ^c, Yueyong Lyu^a and David Navarro-Alarcon ^b

^aSchool of Astronautics, Harbin Institute of Technology, Harbin, People's Republic of China; ^bDepartment of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, KLN, Hong Kong; ^cNational Defense Key Laboratory of Space Intelligent Control Technology, Beijing Institute of Control Engineering, Beijing, People's Republic of China

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 robots configuration and the object's shape features is used to deal with the latter's (typically unknown) mechanical properties. The proposed approach computes the rods 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.

ARTICLE HISTORY

Received 17 February 2021
Revised 28 June 2021
and 29 September 2021
Accepted 5 October 2021

KEYWORDS

Robotics; visual servoing; deformable objects; autoencoder; self-organizing network

1. Introduction

Controlling the shape of soft objects automatically with robot manipulators is highly valuable in many applications, such as food processing [1], robotic surgery [2], cable assembly [3], and household works [4]. Although great progress has been achieved in recent years, shape control remains an open problem in robotics [5]. One of the most crucial issues that hamper the implementation of these types of controls is the difficulty to obtain a meaningful and efficient feedback representation of the objects 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 [6]. At present, traditional methods are roughly divided into two categories: local

and global descriptors. Local descriptors may use centroids, distances, angles, curvatures [7] to describe geometric characteristics. But, these features must be hard-coded and can only represent a limited type of shape with showing part of the characteristics of the object. Global descriptors produce a generic representation of the objects overall shape. An example method under this category is the Point Feature Histogram (PFH) reported in [8], which 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 [9, 10]. However, these methods are limited by the acquisition accuracy of the RGB-D camera, and the computational cost is high. A method based on linearly parameterized (truncated) Fourier series was also proposed to represent the objects contour [11]. This parameterization idea was generalized in [12], where more shape representations were analyzed and implemented. While, the above geometric-based techniques generate high-dimensional feature vectors, which may not be the most effective representation method in soft object deformation.

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 [13, 14]. Force and position measurements of a three-finger gripper manipulating a soft object were used in [6] as input to a network, which produced and predicted the objects 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 [15]. Growing neural gas was used in [16] to represent deformable shapes. However, the network structures of the above methods are complicated without a satisfactory operation speed. Note that real-time performance is also an indicator of shape representation, not just accuracy. This is different from traditional computer vision. A more efficient method is to generate feedback features automatically (e.g. used in the soft object simulation [17]) and combining them with dimension reduction techniques [18].

Latent space approaches have been widely used in data reduction, and image analysis [19] because they can encode high-dimensional data into a meaningful internal representation with low-dimensional variables by constructing highly flexible generators. A feature extractor based on the convolutional autoencoder was developed to obtain a low-dimensional latent space from tactile sensing data [20]. Convolutional neural networks were used to learn the physical model in the latent space [21] to build the inverse kinematics of a rope [22]. Deep autoencoder was used for content-based image retrieval by using semantic hashing to constructing short binary codes [23], which allows much too accurate image matching and reduces the dimension of the image feature. Despite its valuable (and reduction) feature extraction properties, deep autoencoder has not yet been utilized in soft object manipulation tasks. Although [24] proposed a variety of methods, including autoencoder for shape representation, the author did not form a complete control system, and [24] was based on the learning while our proposed method is from the perspective of control.

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 Figure 1). The coordinates obtained are denoted by:

$$\bar{\mathbf{c}} = [\mathbf{c}_1^T, \dots, \mathbf{c}_N^T]^T \in \mathbb{R}^{2N}, \mathbf{c}_i = [u_i, v_i]^T \in \mathbb{R}^2 \quad (1)$$

where N represents the number of points that make up the centerline, u_i and v_i represents the coordinates of the i th ($i = 1, \dots, N$) point in the image frame.

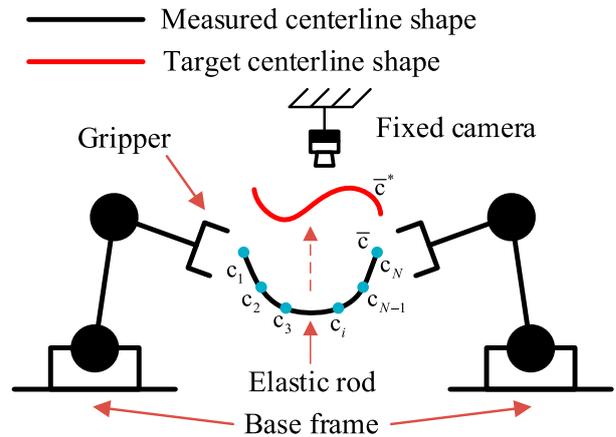


Figure 1. Schematic diagram of the elastic rod shape deformation. The camera is utilized to determine shape feature \mathbf{s} in real time, and within the designed controller the robot automatically deform the real-time shape denoted by $\bar{\mathbf{c}}$ of elastic rods into the target shape $\bar{\mathbf{c}}^*$.

- 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 $\Delta \mathbf{r}_k \in \mathbb{R}^q$ [25] 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 cannot meet the above requirements and need pre-processing of data, which will deteriorate the systems 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 [26], 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 Figure 2. The points in the white area are defined by $\bar{\mathbf{m}} = [\mathbf{m}_1^T, \dots, \mathbf{m}_M^T]^T \in \mathbb{R}^{2M}$, $\mathbf{m}_i \in \mathbb{R}^2$ 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, $SOM : 2M \rightarrow 2N$.

Remark 3.1: 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 [12] is utilized to reorder the centerline. This process will not take too much time because N is small.

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 [24]. 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 $\mathbf{s} = \mathbf{f}_{\theta_1}(\bar{\mathbf{c}}) = \text{sig}(\mathbf{W}_1 \bar{\mathbf{c}} + \mathbf{b}_1)$, 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 $\hat{\bar{\mathbf{c}}} =$

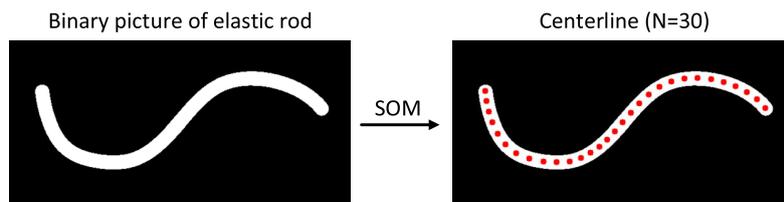


Figure 2. Schematic diagram of SOM-based centerline extraction. The white area in the left side represents the area of elastic rod (clustering area), and the red points in the right side represent the obtained centerline points (clustering points) (in this figure, $N = 30$).

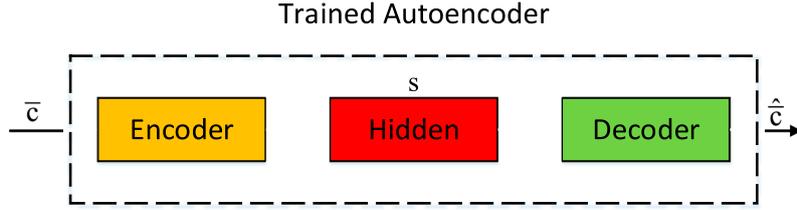


Figure 3. Structure of DAE with the centerline $\bar{\mathbf{c}}$ as the input, and \mathbf{s} is defined as the shape feature used for DJM approximation and controller design.

$\mathbf{g}_{\theta_2}(\mathbf{s}) = \text{sig}(\mathbf{W}_2\mathbf{s} + \mathbf{b}_2)$, 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:

$$\{\theta_1^*, \theta_2^*\} = \arg \min_{\theta_1, \theta_2} \sum_{k=1}^N L(\mathbf{c}_i, \mathbf{g}_{\theta_2}(f_{\theta_1}(\mathbf{c}_i))) \quad (2)$$

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}_{θ_1} . The workflow of DAE is shown in Figure 3.

For DAE, the reconstruction output $\hat{\mathbf{c}}$ 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).

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

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

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

Differentiating (4) concerning time variable t yields:

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

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 $\dot{\mathbf{r}}$, i.e. deformation Jacobian matrix (DJM) [11]. 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:

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

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

$$\mathbf{y}_k = \Delta \mathbf{s}_k = \mathbf{s}_k - \mathbf{s}_{k-1}, \quad \mathbf{u}_k = \Delta \mathbf{r}_k = \mathbf{r}_k - \mathbf{r}_{k-1} \quad (7)$$

Broyden algorithms are as follows:

- (1) R1 update formula [27]:

$$\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} + \frac{(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)\mathbf{u}_k^T}{\mathbf{u}_k^T\mathbf{u}_k} \quad (8)$$

This form has a simple structure and fast calculation speed.

- (2) SR1 update formula [27]:

$$\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} + \frac{(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)^T}{\mathbf{u}_k(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)^T} \quad (9)$$

The structure of SR1 is similar to R1, but the calculation accuracy is higher.

- (3) DFP update formula [28]:

$$\hat{\mathbf{J}}_k = \hat{\mathbf{J}}_{k-1} + \frac{(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)\mathbf{y}_k^T + \mathbf{y}_k(\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)^T}{\mathbf{u}_k\mathbf{y}_k^T} - \frac{\mathbf{y}_k^T\mathbf{y}_k}{\|\mathbf{u}_k\mathbf{y}_k^T\|} (\mathbf{y}_k - \hat{\mathbf{J}}_{k-1}\mathbf{u}_k)\mathbf{u}_k^T \quad (10)$$

DFP is a rank two quasi-Newton method, which is efficient for solving nonlinear optimization.

(4) BFGS update formula [29]:

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

It is recognized with the best numerical stability.

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

Remark 3.2: The robot is assumed to manipulate elastic rods at low speed. Thus, the deformation of the elastic rods is relatively slow. On the basis, DJM \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 , DJM $\hat{\mathbf{J}}_k$ has been assumed to be exactly approximated by (8)–(11), so that the shape-motion difference model is satisfied:

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

A model predictive controller [30] 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 DJM $\hat{\mathbf{J}}_k$ and (12), the predicted deformation output at time $k + w$ is shown below:

$$\mathbf{s}_{k+w}^p = \mathbf{s}_k + \hat{\mathbf{J}}_k \cdot \mathbf{u}_{k+w} \quad (13)$$

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 DJM as follows:

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

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

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

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:

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

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 DJM will be affected. Taking derivative of (16) with respect to \mathbf{u}_k , the gradient ∇ is calculated as follows:

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

where $\beta = e^{-\rho}$. By setting $\nabla = 0$, the velocity command \mathbf{u}_k is derived [31]:

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

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

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{u}_k \quad (19)$$

Following (12), it yields:

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

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

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

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, $\lim_{t \rightarrow \infty} \mathbf{s}_k = \mathbf{s}_k^*$. 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 [32]. The workflow of the proposed framework is presented in Algorithm 1. Figure 4 gives the block diagram of the proposed manipulation framework.

Remark 3.3: The velocity controller (18) and DJM estimators (8)–(11) only require visual feedback data without any additional sensors, prior knowledge of the system model, and the requirement to calibrate the camera.

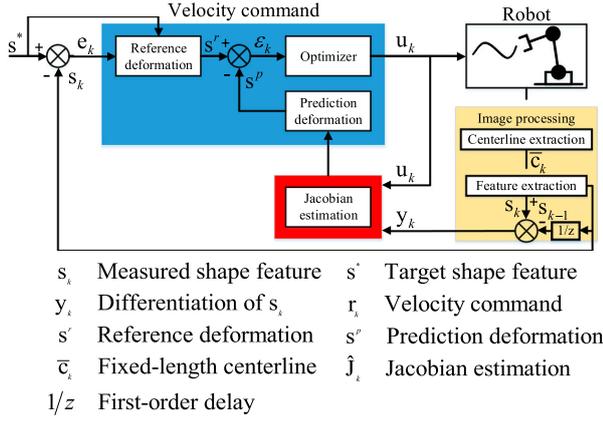


Figure 4. The block diagram of the proposed manipulation framework.

Algorithm 1 Workflow of the proposed framework.

Require: *Threshold; Max;*

- 1: Give a target shape \bar{c}^* represented by s^* ;
 - 2: Conduct small deformations around the starting configuration to initialize \hat{j}_0 and start the manipulation;
 - 3: $k = 0$
 - 4: **while** $\|e_k\| \geq \text{Threshold}$ and $k < \text{Max}$ **do**
 - 5: Record the current position r_k and velocity u_k ;
 - 6: Record the current shape \bar{c}_k ;
 - 7: Calculate the current shape feature s_k by trained DAE;
 - 8: Calculate error $e_k = s^* - s_k$;
 - 9: Update $u_k, \leftarrow (18)$;
 - 10: Update $\hat{j}_k, \leftarrow (8) - (11)$;
 - 11: Dual-arm robot moves using the updated u_k ;
 - 12: $k = k + 1$;
 - 13: **end while**
-

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 [33] by using the minimum energy principle [34], 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 Figure 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.

For better illustration, $\|\bar{c} - \hat{c}\|$ is defined by the reconstruction error between the feedback shape \bar{c} and the reconstruction shape \hat{c} . The detailed comparison is presented in Figure 6 among DAE, Principal Component Analysis (PCA) [18], Fourier [11] and NURBS [12]. We ensure that the dimension of the shape feature s has increased to the limit of each method. p and k determine the dimension of s of DAE and PCA, and k is set to be equal to p for the fairness of competition. Detailed feature dimensions are set: $p \in \{1, 15\}$ for DAE, $k \in \{1, 15\}$ for PCA, $p \in \{6, 10, \dots, 62\}$ for Fourier, and $p \in \{4, 6, \dots, 32\}$ for NURBS. Note that the feature dimension calculation of Fourier and NURBS is different from DAE and PCA. The dimension of Fourier is $4n + 2$, and that of NURBS is $2n + 2$, where n represents the fitting order. We set $n \in \{1, 15\}$ here to be compared fairly with DAE and PCA. The results in Figure 6 show that DAE has the best accuracy in each dimension. DAE with $p = 15$ has the best reconstruction accuracy, while it has the lowest accuracy with $p = 1$. This means that p is too low to represent the elastic rod fully. It can be seen that when the dimension increases to a certain extent, the fitting accuracy of DAE is no longer significantly improved, which also means that DAE has a specific upper fitting limit. Theoretically, this limit can be improved without increasing the feature dimension by increasing the layers of the neural network and increasing the data set. And DAE is compared with Fourier [11] and NURBS [12], the latter two are geometric-based methods. Figure 6 shows that DAE also performs better than Fourier and NURBS. The feature dimension of DAE is less than that of Fourier and NURBS under similar accuracy, significantly enhancing the system's stability. Although, Fourier and NURBS can also achieve good results in high-dimensional situations, this may affect the robustness and stability of the system. And there is a trade balance between feature extraction accuracy and feature dimensions in shape deformation. For more information about overdetermined visual servoing, we refer the readers to [32] for detailed stability analysis. Considering the trade balance of system controllability and shape representation, DAE with $p = 4$ is used in the following sections.

4.2. Validation of the Jacobian estimation

In this section, we command the robot to move along the circular trajectory around the center (0.4, 0.4) with

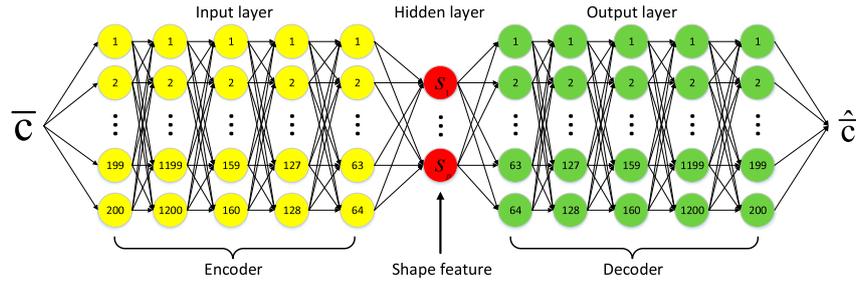


Figure 5. Structure of DAE comprised of MLPs as the basic blocks of Encoder and Decoder. The centerline $\bar{\mathbf{c}}$ is fed into the trained DAE to generate shape feature denoted by \mathbf{s} .

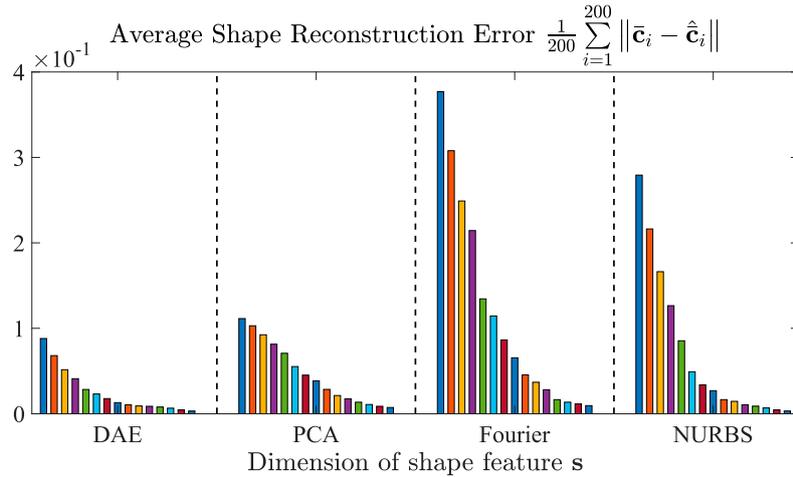


Figure 6. Average shape reconstruction error comparison of 200 shape sets among DAE, PCA [18], Fourier [11] and NURBS [12]. The dimension of each method is: $p \in \{1, 15\}$ for DAE, $k \in \{1, 15\}$ for PCA, $p \in \{6, 10, \dots, 62\}$ for Fourier, and $p \in \{4, 6, \dots, 32\}$ for NURBS arranging from left to right.

known smooth $\Delta \mathbf{r}_k$. The goal is to compare the accuracy of estimating DJM among R1, SR1, DFP, and BFGS, and that obtained with Least Square Method [35] and Inverse Matrix Calculation [18]. Different from initializing $\hat{\mathbf{J}}_0$ with a random matrix, we command the robot to move with small local motions (the motions are ensured not to be collinear) and run the estimator to get a suitable $\hat{\mathbf{J}}_0$. Two error criteria (22) are introduced to quantify the performance of such estimation:

$$T_1 = \|\mathbf{s}_k - \hat{\mathbf{s}}_k\|, T_2 = \|\Delta \mathbf{s}_k - \hat{\mathbf{J}}_k \Delta \mathbf{r}_k\| \quad (22)$$

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

$$\hat{\mathbf{s}}_k = \hat{\mathbf{s}}_{k-1} + \hat{\mathbf{J}}_k \Delta \mathbf{r}_k, \hat{\mathbf{s}}_0 = \mathbf{s}_0 \quad (23)$$

where $\hat{\mathbf{J}}_k$ is estimated by R1, SR1, DFP, BFGS, Least Square Method [35] and Inverse Matrix Calculation [18]. $\hat{\mathbf{s}}_0$ and \mathbf{s}_0 are the initial values of $\hat{\mathbf{s}}_k$ and \mathbf{s}_k , respectively. The evolution of T_1 and T_2 for the motions $\Delta \mathbf{r}_k$ executed by the robot are demonstrated in Figure 7. It can be concluded that BFGS owns the best estimation performance amongst the compared algorithms (viz. R1, SR1,

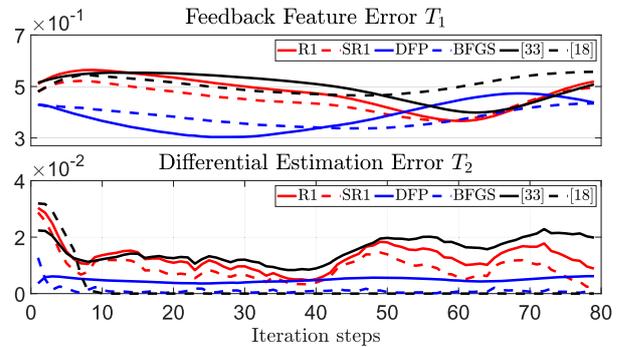


Figure 7. Profiles of the criteria T_1 and T_2 that are computed along the circular trajectory around the center (0.4, 0.4) among R1, SR1, DFP, BFGS, Least Square Method [35] and Inverse Matrix Calculation [18].

DFP, [18, 35]). The results corroborate that, when starting deformation, BFGS can calibrate and update DJM to identify the pseudo-physical parameters of the elastic rods online. Specifically, BFGS can accurately estimate the shape feature \mathbf{s}_k , and its differential changes $\Delta \mathbf{s}_k$ in the latent space, which can guide the robot with the estimated DJM.

4.3. Manipulation of elastic rods

In this section, we validate the performance of the proposed shape controller by commanding the robot to deform the elastic rods into the desired constant shape $\bar{\mathbf{c}}^*$ (which is transformed into a desired shape feature \mathbf{s}^*). The error criterion (24) is utilized to assess the deformation performance.

$$T_3 = \|\bar{\mathbf{c}}_k - \bar{\mathbf{c}}^*\| \quad (24)$$

The progress of the cable deformation among R1, SR1, DFP and BFGS combined with the velocity command (18) is depicted in Figure 8. The curve of T_3 and $\Delta \mathbf{r}_k$ are shown in Figure 10(a). 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. To better demonstrate the usefulness and effectiveness of the latent space approach in shape deformation, we compare the error minimization performance of DAE with that of NURBS [12], Fourier [11], and PCA [18] with the DJM adaptively estimated by BFGS with the velocity controller (18). From Figure 9, it is concluded that four algorithms all complete the shape deformation task. Figure 10(b) shows that DAE provides the fastest error minimization, while NURBS and Fourier give a similar

performance. The above results prove the effectiveness of DAE in shape deformation than those that do not exploit latent space.

5. Experimental results

Various experiments with two UR5 that support velocity control mode are conducted, as shown in Figure 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.

5.1. Image processing

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

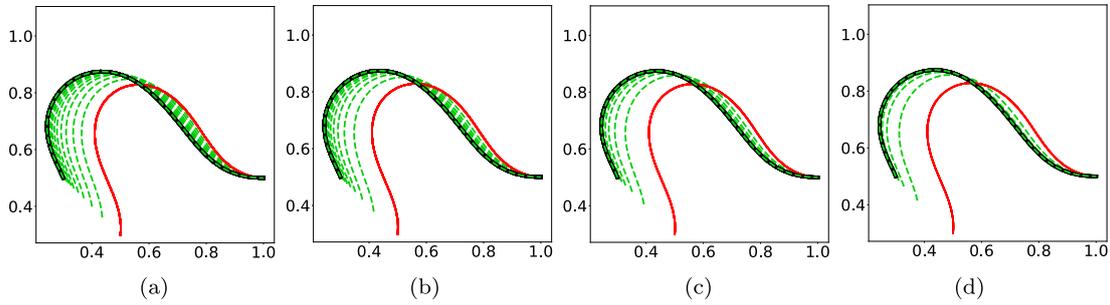


Figure 8. Profiles of the shape deformation simulation among R1, SR1, DFP and BFGS combined with DAE and the velocity controller (18). Red solid curve represents the initial shape, green dashed curves represent transitional trajectories, and the black solid curve represents the target shape $\bar{\mathbf{c}}^*$ represented by \mathbf{s}^* . The deformation trajectories display every three frames. (a) R1, (b) SR1, (c) DFP, (d) BFGS.

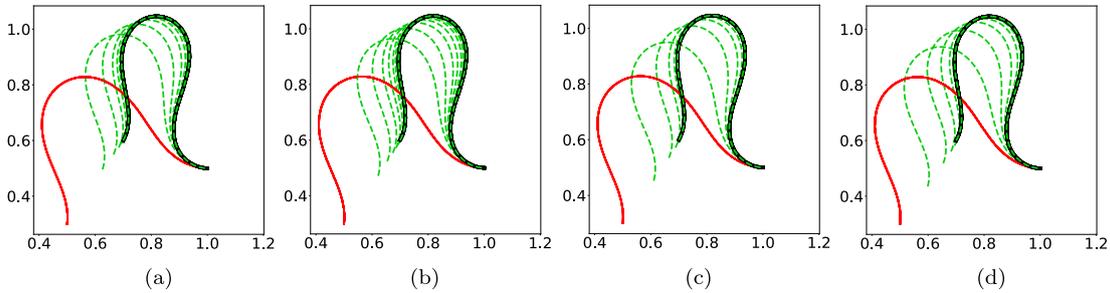


Figure 9. Profiles of the shape deformation trajectories among DAE, NURBS [12], Fourier [11] and PCA [18] combined with BFGS and the velocity controller (18). Red solid curve represents the initial shape, green dashed curves represent transitional trajectories, and the black solid curve represents the target shape $\bar{\mathbf{c}}^*$ represented by \mathbf{s}^* . The deformation trajectories display every three frames. (a) PCA, (b) Fourier, (c) NURBS, (d) DAE.

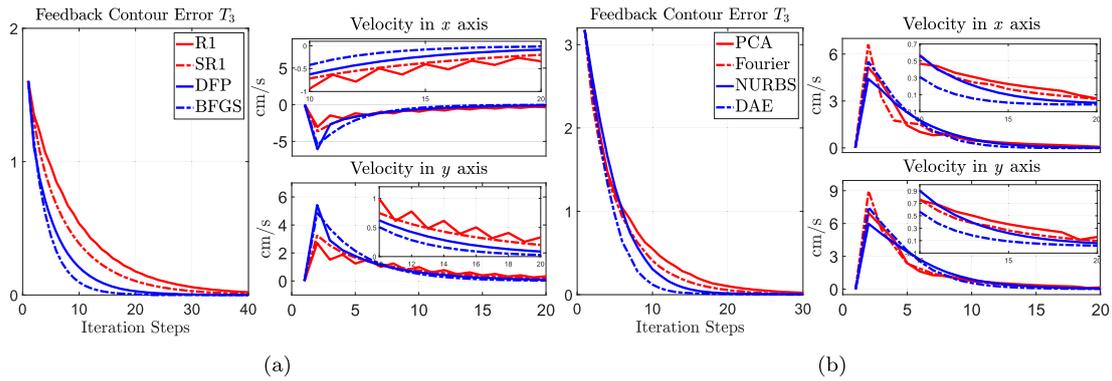


Figure 10. (a) The profiles of the criterion T_3 and velocity command $\Delta \mathbf{r}_k$ among R1, SR1, DFP and BFGS within manipulation. (b) The profiles of the criterion T_3 and velocity command $\Delta \mathbf{r}_k$ among DAE, NURBS [12], Fourier [11] and PCA [18].

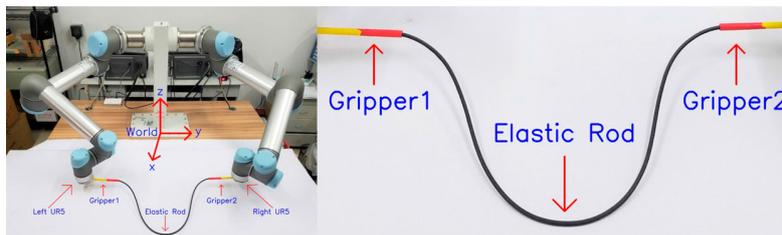


Figure 11. Experimental setup comprised of two UR5 which support velocity control mode.

First, the proposed SOM-based method is compared with two other centerline extraction algorithms. The first one is based on the *OpenCV/thinning* developed in Reference [12], and the second one is based on CL, which is the traditional clustering method [36]. All algorithms need to provide an ordered, fixed-length $N = 50$, equidistant centerline for the fairness of competition. The adjacent distance along the rod of each point is approximately the same, which is an important evaluation index. As the CL-based and SOM-based algorithms only generate an unordered fixed-length centerline, the sorting algorithm [12] is used to sort unordered centerlines. For SOM, an open-source toolbox provided by [37] is utilized. It can be seen from Figure 12 that Reference and SOM have similar extraction performance as both shapes look the same. However, Reference sorts all the points first and then performs down-sampling; thus, its calculation time

is slower than that of SOM and CL (see Table 1), which is unfriendly to the real-time requirements of the system. As for CL in Figure 12(b), the points on the left and right sides are densely arranged, and the points in the middle part are arranged loosely. This does not meet the requirements of the equidistant arrangement. From the perspective of extraction accuracy, $\text{SOM} = \text{Reference} > \text{CL}$. From the perspective of calculation time, $\text{SOM} < \text{CL} < \text{Reference}$, see Table 1. Thus, SOM has the best performance with the fastest extraction speed, while CL-based has the worst performance. One reason is that

Table 1. Comparison results among three centerline extraction algorithms with $N = 50$.

	Reference [12]	CL [36]	SOM
Time (Second)	1.68	0.98	0.38

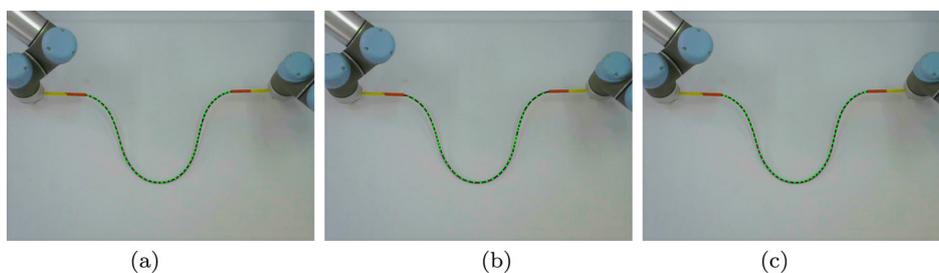


Figure 12. Comparison of three centerline extraction algorithms, including reference [12], CL-based [36] and the proposed SOM-based. (a) Reference [12], (b) CL [36], (c) Proposed SOM.

the 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. Thus, the SOM-based centerline extraction algorithm is used by considering the reasons for accuracy and speed.

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

- (1) First, segment the red areas nearby 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 Figure 13(a)).
- (2) Next, identify the rod in ROI, remove the noise, and obtain a binary image of the rod using OpenCV morphological opening algorithm (see Figure 13(b)).

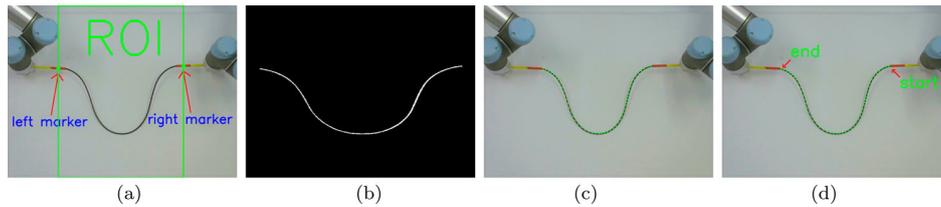


Figure 13. Image processing steps. (a) ROI selection, (b) Thresholding, (c) Centerline (SOM), (d) Centerline sorting.

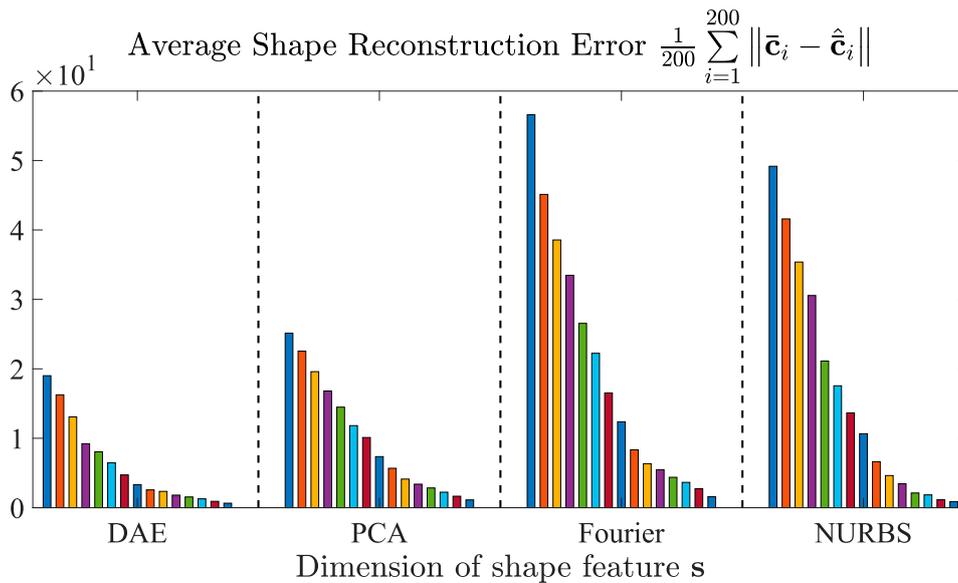


Figure 14. Average shape reconstruction error comparison of 200 shape sets among DAE, PCA [18], Fourier [11] and NURBS [12]. The dimension of each method is: $p \in \{1, 15\}$ for DAE, $k \in \{1, 15\}$ for PCA, $p \in \{6, 10, \dots, 62\}$ for Fourier, and $p \in \{4, 6, \dots, 32\}$ for NURBS arranging from left to right.

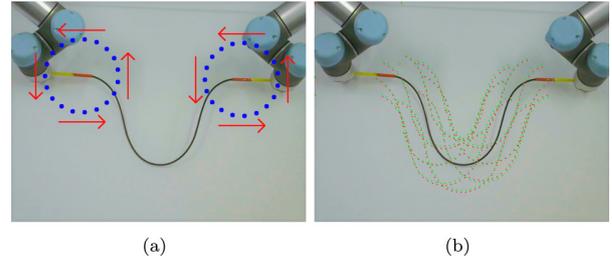


Figure 15. DJM \hat{J}_k validation framework. (a) Motion trajectory of the end-effectors of dual-arm UR5. (b) Comparison between the visually measured cable profile (green dot line) and its reconstruction shape obtained by DAE (red dot line) with $p = 4$.

- (3) Subsequently, use the proposed SOM-based algorithm to get an unordered centerline with $N = 50$ (see Figure 13(c)).
- (4) Finally, apply the sorting algorithm [12] to get an ordered centerline (see Figure 13(d)). The starting

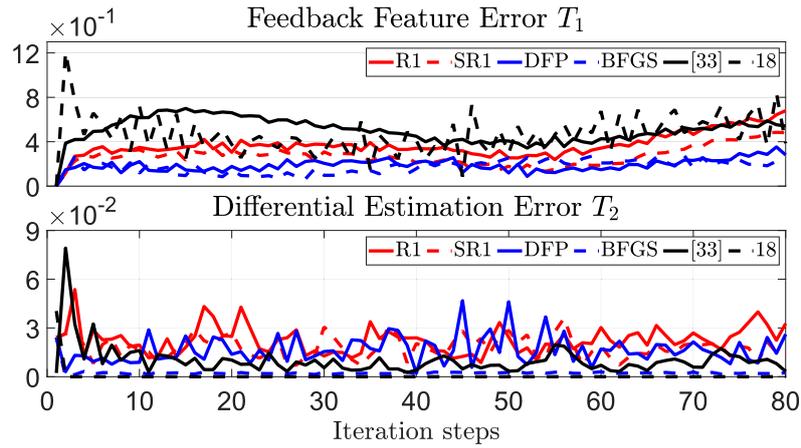


Figure 16. Profiles of the criteria T_1 and T_2 that are computed along the circular trajectory among R1, SR1, DFP, BFGS, Least Square Method [35] and Inverse Matrix Calculation [18].

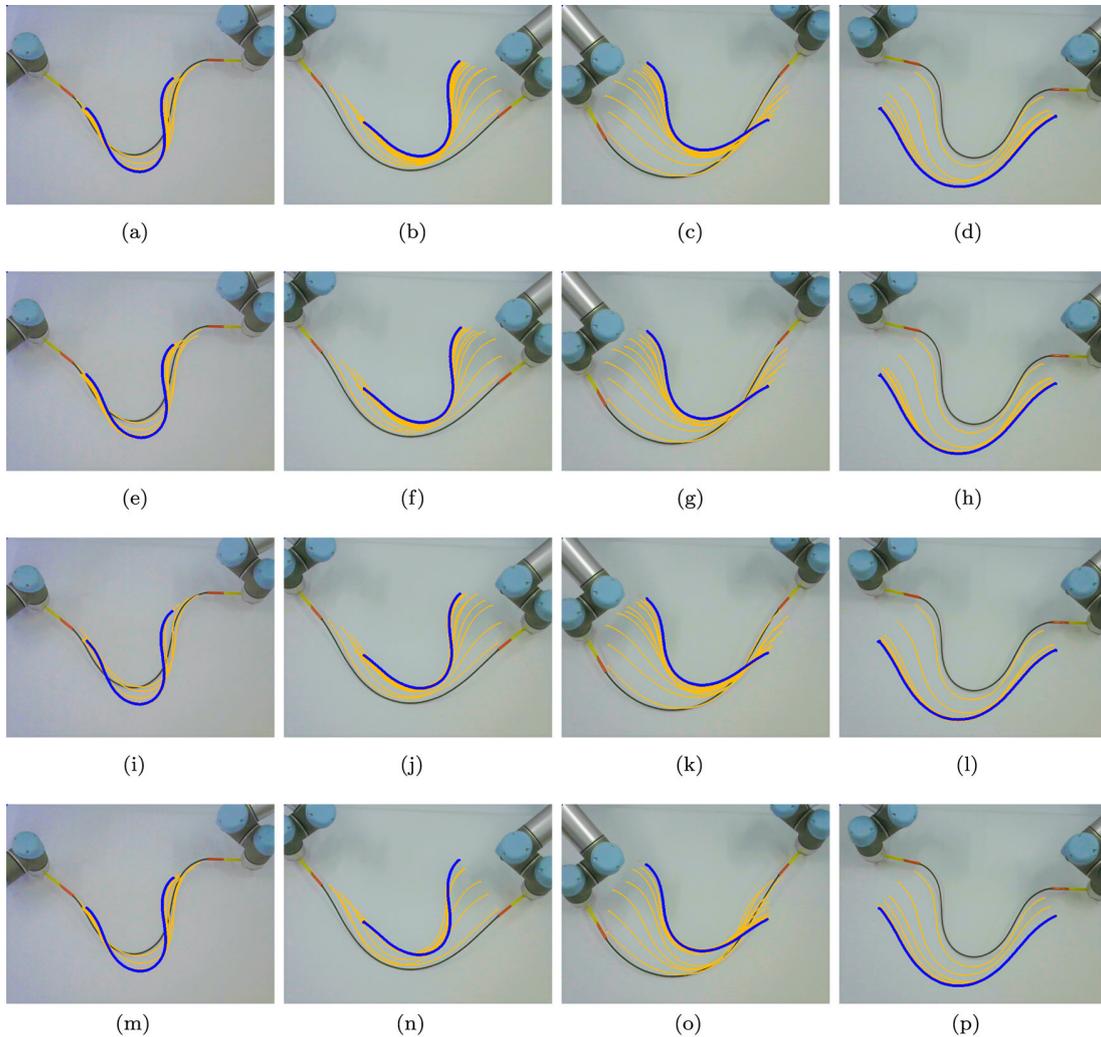


Figure 17. Initial (black solid line), transition (orange solid line) and target (blue solid line) configurations in the four shape deformation experiments manipulated by dual-arm UR5. Each row represents the different initial and target shape, while each column represents different DJM estimators, i.e. R1, SR1, DFP and BFGS. (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.

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 Figure 5. Batch-Normalization-1D (BN) is added after each layer. Figure 14 shows that DAE with $p = 15$ has the highest reconstruction accuracy while that of $p = 1$ is the worst. In the same feature-dimensional, DAE is better than PCA [18]. We also compare DAE with Fourier [11] and NURBS [12]. The results in Figure 14 give that DAE still has the best reconstruction accuracy with lower-dimensional feature vectors than Fourier and NURBS. The above results are consistent with the simulation results, which fully proves the effectiveness of DAE in the shape representation. In the following sections, we uniformly use DAE with $p = 4$.

5.3. Validation of the Jacobian estimation

Similar to Section 4.2, dual-arm UR5 is commanded to move along a fixed circular trajectory, as depicted in Figure 15(a). The shape reconstruction performance of DAE with $p = 4$ is accurate in the experiment, as shown

in Figure 15(b). Same as before, R1, SR1, DFP and BFGS are compared with [38] and [18]. The results in Figure 16 show that BFGS has a minimal approximation error, which confirms that BFGS has a superior performance in estimating the unknown DJM.

5.4. Manipulation of elastic rods

Similar to Section 4.3, we conducted an experimental research where the dual-arm UR5 manipulates the elastic rod to various target shapes. We describe these experiments as Exp1 ... Exp4 with different initial and target shapes. Considering safety, the saturation of $\Delta \mathbf{r}$ is set to, $|\Delta r_{i1}| \leq 0.01$ m/s, $|\Delta r_{i2}| \leq 0.01$ m/s and $|\Delta r_{i3}| \leq 0.1$ rad/s, $i = 1, 2, 3$, respectively.

Figure 17 presents the active shape deformation trajectories (represented by the green centerlines) of the elastic rod towards the target shape (represented by the red centerline). This figure qualitatively compares the centerline trajectories of different target shapes (each in a separate row) with the four DJM estimators (each in a separate column). Figure 18 shows the time evolution of the shape error T_3 and the respective velocity command $\Delta \mathbf{r}_k$. From these results, it can be seen that BFGS has the best control effect with the fastest convergence speed and has excellent adaptability to various conditions in the shape deformation, which is consistent

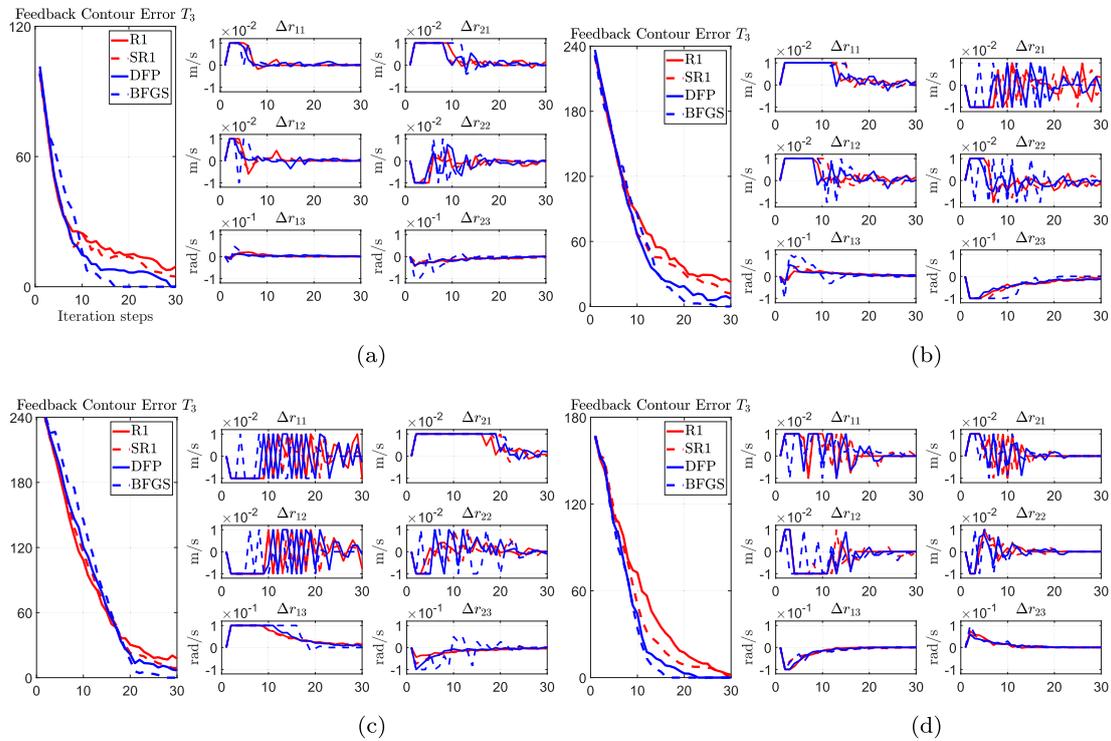


Figure 18. Profiles of the criterion T_3 and velocity command $\Delta \mathbf{r}_k$ among R1, SR1, DFP and BFGS within four shape deformation experiments. (a) Experiment1 result, (b) Experiment2 result, (c) Experiment3 result, (d) Experiment4 result.

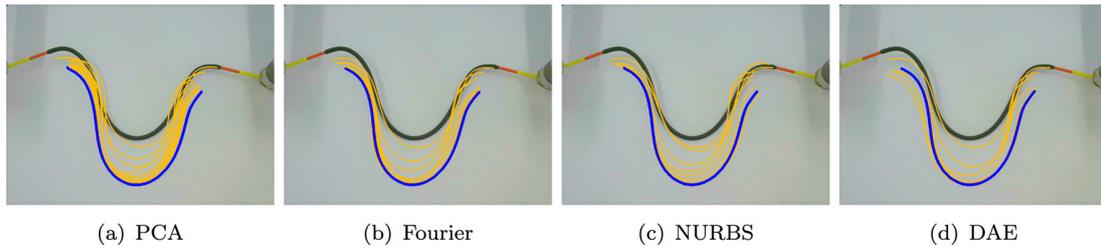


Figure 19. Initial (black solid line), transition (orange solid line) and target (blue solid line) configurations in the comparison experiment manipulated by dual-arm UR5 among DAE, NURBS [12], Fourier [11] and PCA [18] combined with BFGS and the velocity controller (18). (a) Fourier, (b) NURBS, (c) DAE.

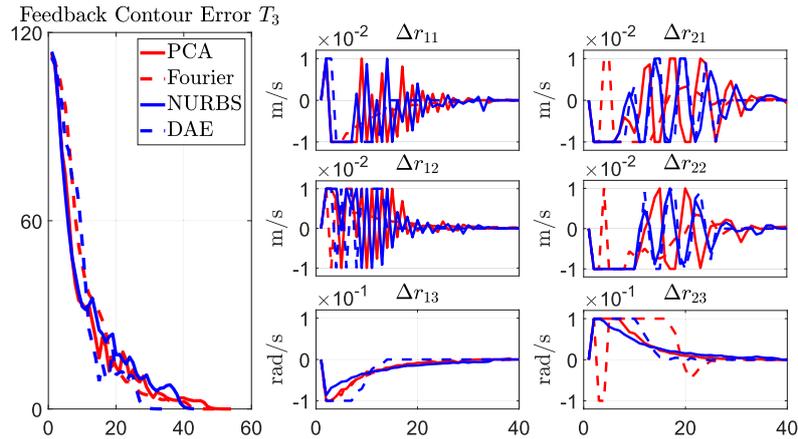


Figure 20. Profiles of the criterion T_3 and velocity command $\Delta \mathbf{r}_k$ among DAE, NURBS [12], Fourier [11] and PCA [18] combined with BFGS and the velocity controller (18).

with the simulation results. Same as before, we compare the error minimization performance of DAE with that of NURBS [12], Fourier [11] and PCA [18] with the DJM adaptively estimated by BFGS with the velocity controller (18). Figure 19 shows that the shape deformation trajectories among four shape feature extraction algorithms. Figure 20 shows that DAE provides the fastest error minimization compared to the other methods. The above results prove the effectiveness of the proposed framework in the shape deformation and prove the effectiveness of DAE in the shape representation than those that do not exploit latent space.

6. Conclusions

A framework for the deformation control of elastic rods is proposed without any prior physical knowledge. It includes shape feature extraction, DJM 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 DJM 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 DJM 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 DJM 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.

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.

Acknowledgments

Jiaming Qi contributes to the conception of the study, writes the manuscript and performs the experiment; Guangfu Ma helps perform the analysis with constructive discussions; David Navarro-Alarcon and Peng Zhou contribute significantly to analysis and manuscript preparation; Haibo Zhang and Yueyong Lyu give some constructive suggestions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported in part by the Key-Area Research and Development Program of Guangdong Province (2020B090928001), in part by the Hong Kong Research Grants Council [grant number 14203917], in part by the Jiangsu Industrial Technology Research Institute Collaborative Funding Scheme under grant 43-ZG9V, and in part by The Hong Kong Polytechnic University [grant numbers ZZH] and YBYT].

Notes on contributors

Jiaming Qi received the MSc in Integrated Circuit Engineering from Harbin Institute of Technology, Harbin, China, in 2018. In 2019, he was a visiting PhD student at The Hong Kong Polytechnic University. He is currently pursuing the PhD degree with Control Science and Engineering, Harbin Institute of Technology, Harbin, China. His current research interests include data-driven control for soft object manipulation, vision-servoed control, robotics and control theory.

Guangfu Ma received the PhD and MS 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.

Peng Zhou (S'20) was born in China. He received the MSc degree in software engineering from the School of Software Engineering, Tongji University, Shanghai, China, in 2017 and is currently pursuing his the PhD degree in robotics in The

Hong Kong Polytechnic University, Kowloon, Hong Kong. His research interests include deformable object manipulation, motion planning and robot learning.

Haibo Zhang received the PhD and MS 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 Beijing Institute of Control Engineering.

Yueyong Lyu is now an associate research fellow in the Department of Control Science and Engineering, Harbin Institute of Technology, China. He also got his Bachelor, Master and PhD degree from Harbin Institute of Technology respectively in 2002, 2008 and 2013. His interests of research mainly focus on spacecraft guidance, navigation and control, especially in spacecraft formation flying, on-orbit service, and so on.

David Navarro-Alarcon (GS'06-M'14-SM'19) received the PhD 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 at the Department of Mechanical Engineering, Principal Investigator of the Robotics and Machine Intelligence Laboratory, and Investigator at the Research Institute for Smart Ageing. Before joining PolyU, he was a Postdoctoral Fellow and then Research Assistant Professor at the CUHK T Stone Robotics Institute, from 2014 to 2017. He has had visiting appointments at the University of Toulon in France and the Technical University of Munich in Germany. His current research interests include perceptual robotics and control theory.

ORCID

Jiaming Qi  <http://orcid.org/0000-0003-2655-6835>

Peng Zhou  <http://orcid.org/0000-0002-7020-0943>

Haibo Zhang  <http://orcid.org/0000-0002-8309-2558>

David Navarro-Alarcon  <http://orcid.org/0000-0002-3426-6638>

References

- [1] 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. IEEE; 2002. p. 1457–1464.
- [2] Abolmaesumi P, Salcudean SE, Zhu WH, et al. Image-guided control of a robot for medical ultrasound. *IEEE Trans Robot Autom.* 2002;18(1):11–23.
- [3] Tang T, Wang C, Tomizuka M. A framework for manipulating deformable linear objects by coherent point drift. *IEEE Robot Autom Lett.* 2018;3(4):3426–3433.
- [4] 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). IEEE; 2019. p. 47–52.
- [5] Navarro-Alarcon D, Liu Yh, Romero JG. On the visual deformation servoing of compliant objects: uncalibrated

- control methods and experiments. *Int J Robot Res.* 2014;33(11):1462–1480.
- [6] Cretu AM, Payeur P, Petriu EM. Soft object deformation monitoring and learning for model-based robotic hand manipulation. *IEEE Trans Syst Man Cybern B.* 2011;42(3):740–753.
 - [7] Navarro-Alarcon D, Yip HM, Wang Z, et al. Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model. *IEEE Trans Robot.* 2016;32(2):429–441.
 - [8] Rusu RB, Marton ZC, Blodow N, et al. Persistent point feature histograms for 3d point clouds. In: *Proceedings 10th International Conference Intel Autonomous System (IAS-10); Baden-Baden, Germany; 2008.* p. 119–128.
 - [9] Rusu RB, Blodow N, Beetz M. Fast point feature histograms (fpfh) for 3d registration. In: *2009 IEEE International Conference on Robotics and Automation; 2009.* p. 3212–3217.
 - [10] Hu Z, Han T, Sun P, et al. 3-d deformable object manipulation using deep neural networks. *IEEE Robot Autom Lett.* 2019;4(4):4255–4261.
 - [11] Navarro-Alarcon D, Liu YH. Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-d image contours. *IEEE Trans Robot.* 2017;34(1):272–279.
 - [12] Qi J, Ma W, Navarro-Alarcon D, et al. Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls. Preprint 2020. Available from: arXiv:200806896.
 - [13] Yeo N, Lee K, Venkatesh Y, et al. Colour image segmentation using the self-organizing map and adaptive resonance theory. *Image Vis Comput.* 2005;23(12):1060–1079.
 - [14] Han H, Paul G, Matsubara T. Model-based reinforcement learning approach for deformable linear object manipulation. In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE).* IEEE; 2017. p. 750–755.
 - [15] Yan M, Zhu Y, Jin N, et al. Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE Robot Autom Lett.* 2020;5(2):2372–2379.
 - [16] Valencia AJ, Nadon F, Payeur P. Toward real-time 3d shape tracking of deformable objects for robotic manipulation and shape control. In: *2019 IEEE SENSORS.* IEEE; 2019. p. 1–4.
 - [17] Alvarez N, Yamazaki K, Matsubara T. An approach to realistic physical simulation of digitally captured deformable linear objects. In: *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN).* IEEE; 2016. p. 135–140.
 - [18] Zhu J, Navarro-Alarcon D, Passama R, et al. Vision-based manipulation of deformable and rigid objects using sub-space projections of 2d contours. *Robot Autom Syst.* 2020.
 - [19] Hoff PD, Raftery AE, Handcock MS. Latent space approaches to social network analysis. *J Am Stat Assoc.* 2002;97(460):1090–1098.
 - [20] Polic M, Krajacic I, Lepora N, et al. Convolutional autoencoder for feature extraction in tactile sensing. *IEEE Robot Autom Lett.* 2019;4(4):3671–3678.
 - [21] Ebert F, Finn C, Dasari S, et al. Visual foresight: model-based deep reinforcement learning for vision-based robotic control. Preprint 2018. Available from: arXiv:181200568.
 - [22] Nair A, Chen D, Agrawal P, et al. Combining self-supervised learning and imitation for vision-based rope manipulation. In: *2017 IEEE International Conference on Robotics and Automation (ICRA).* IEEE; 2017. p. 2146–2153.
 - [23] Krizhevsky A, Hinton GE. Using very deep autoencoders for content-based image retrieval. In: *ESANN.* Vol. 1; Citeseer; 2011. p. 2.
 - [24] Zhou P, Zhu J, Huo S, et al. Lasesom: a latent and semantic representation framework for soft object manipulation. *IEEE Robot Autom Lett.* 2021;6(3):5381–5388.
 - [25] Cherubini A, Navarro-Alarcon D. Sensor-based control for human-robot collaboration: fundamentals, challenges and opportunities. *Front Neurobot.* 2021;14:113.
 - [26] Zahra O, Navarro-Alarcon D. A self-organizing network with varying density structure for characterizing sensorimotor transformations in robotic systems. In: *Towards Autonomous Robotic Systems; 2019.* p. 167–178.
 - [27] Broyden CG. A class of methods for solving nonlinear simultaneous equations. *Math Comput.* 1965;19(92):577–593.
 - [28] Nocedal J. Updating quasi-newton matrices with limited storage. *Math Comput.* 1980;35(151):773–782.
 - [29] Dennis JE, Moré JJ. A characterization of superlinear convergence and its application to quasi-newton methods. *Math Comput.* 1974;28(126):549–560.
 - [30] Ouyang B, Mo H, Chen H, et al. 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).* IEEE; 2018. p. 613–618.
 - [31] Mo H, Ouyang B, Xing L, et al. Automated 3-d deformation of a soft object using a continuum robot. *IEEE Trans Autom Sci Eng.* 2020.
 - [32] Hutchinson S, Chaumette F. Visual servo control, part i: basic approaches. *IEEE Robot Autom Magaz.* 2006;13(4):82–90.
 - [33] 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. IEEE; 1995. p. 433–438.
 - [34] Hamill P. A student's guide to lagrangians and hamiltonians. Cambridge, UK: Cambridge University Press; 2014.
 - [35] Navarro-Alarcon D, Liu YH, Romero JG, et al. Model-free visually servoed deformation control of elastic objects by robot manipulators. *IEEE Trans Robot.* 2013;29(6):1457–1468.
 - [36] Park DC. Centroid neural network for unsupervised competitive learning. *IEEE Trans Neural Netw.* 2000;11(2):520–528.
 - [37] Vettigli G. Minisom: minimalistic and numpy-based implementation of the self organizing map. 2018. Available from: <https://github.com/JustGlowing/minisom/>
 - [38] Navarro-Alarcon D, Liu Y. A dynamic and uncalibrated method to visually servo-control elastic deformations by fully-constrained robotic grippers. In: *IEEE International Conference Model-free Visually Servoed Deformation Control of Elastic Objects by Robot Manipulator on Robotics and Automation; 2014.* p. 4457–4462.