Machine Intelligence Research
www.mi-research.net

22(4), August 2025, 696-712
DOI: 10.1007/s11633-025-1566-0

Online Iterative Learning Enhanced Sim-to-Real Transfer for Efficient Manipulation of Deformable Objects

Zuyan Chen[1] Jian-An Huang[2,3] Juha Röning[1] Leopoldo Angrisani[4] Shuai Li[1,5]

[1] Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu 90570, Finland

[2] Faculty of Medicine, University of Oulu, Oulu 90220, Finland

[3] Faculty of Biochemistry and Molecular Medicine, University of Oulu, Oulu 90220, Finland

[4] Department of Information Technology and Electrical Engineering (DIETI), University of Napoli Federico II, Naples 80125, Italy

[5] VTT, Technical Research Centre of Finland, Oulu 90590, Finland

Abstract: Deformable manipulation has attracted a lot of attention in the field of robotics, especially in medical applications. However, manipulating deformable objects faces various challenges, mainly including their complex dynamic properties and unpredictable nonlinear deformations. It is difficult to provide a basis for deformable object measurements without effective control methods that provide intelligent and accurate position control, and this research also provides a premise for deformable object measurements. To address these issues, this paper proposes an online iterative perception policy (IPP) method, which does not require large-scale deep network training. This method is able to perceive transformations through an iterative process, and achieve efficient and accurate control of deformable objects. Extensive experiments in the simulation environment and the real scene are conducted to validate the effectiveness and superiority of the proposed method, as well as to compare with advanced algorithms (linear–quadratic regulator (LQR), sliding mode control (SMC), model predictive control (MPC), and heuristic). The experimental results reveal that IPP outperforms other approaches in terms of convergence, stability, robustness and flexibility in both the simulation and real-world scenarios, regardless of textile properties or initial conditions.

Keywords: Deformable object manipulation, online policy, transformer, deep learning, perception policy.

Citation: Z. Chen, J. A. Huang, J. Röning, L. Angrisani, S. Li. Online iterative learning enhanced sim-to-real transfer for efficient manipulation of deformable objects. *Machine Intelligence Research*, vol.22, no.4, pp.696–712, 2025. http://doi.org/10.1007/s11633-025-1566-0

1 Introduction

Deformable object research has attracted much attention in the field of robotics, particularly for medical applications[1, 2]. Due to their flexible and non-rigid properties, deformable objects, such as soft tissues, organs and surgical materials, are crucial targets for robotic manipulation. In medical scenarios, effective manipulation of those deformable entities is of fundamental relevance for applications, for instances, minimally invasive surgery, robotic suturing, tissue repair and rehabilitation assistance[3]. This technological breakthrough is expected to improve the precision, safety and efficiency of medical operations, thereby driving changes in healthcare[4].

One of the primary challenges in manipulating deformable objects lies in their complex dynamic and unpredictable nature[5, 6]. Deformable objects, as opposed to rigid objects, exhibit complex, nonlinear deformations when subjected to external forces. The degrees of deformation are influenced not only by the magnitude and direction of applied forces but also by the material properties of the object, such as elasticity and shear modules[7, 8]. Furthermore, accurately describing these deformations with mathematical models is particularly challenging[9, 10]. All of these challenging problems have caused great difficulties in the measurement of deformable objects, and there is an urgent need for a control strategy and framework to accurately control deformable objects in order to provide a research basis for the measurement of deformable objects. Conventional approaches rely on precise modeling and real-time perception, which are essential for dealing with deformable objects that also present significant obstacles[11, 12]. Deformable objects in medical scenarios are often composed of complex and flexible materials, demanding models that integrate a wide range of physical characteristics, including elasticity, viscosity and plasticity[13]. Traditional rigid body mechanics are insufficient to address these complexities. In addition, real-time

Research Article
Special Issue on Embodied Intelligence
Manuscript received on December 19, 2024; accepted on May 23, 2025
Recommended by Associate Editor Long Cheng
Colored figures are available in the online version at https://link.springer.com/journal/11633

perception of the deformable object's state demands high-resolution sensing capabilities, such as vision, force feedback, or multi-modal sensor fusion. However, the data from these sensors is often noisy and computationally expensive to process, further complicating the perception and modeling tasks[14].

To handle with such difficulties and challenges, online iterative perception policy (IPP), an online iterative perception policy method that could perceive the actuation space and the real space at each iteration and does not require massive deep network training, is hence proposed. The main features of IPP are as follows:

1) Effective and lightweight: IPP adopts a two-layer transformer-based encoder-decoder architecture and could be deployed on the RTX 4 070 platform and run in real time.

2) Adaptive: The perception policy could perceive the relationship between the execution space and the real space in each iteration, and could effectively achieve the goal after changing the scene and environment.

3) Robust: The policy applies delta action as input, which ensures that the system can still approach the target stably after changing the actuator and power system.

Compared with the traditional control methods, the proposed method does not depend on a specific system model and has better adaptability to systems with different parameters. Unlike data-driven approaches, the proposed method does not require a large quantity of data for training and frequent interactions with the environment, and its lightweight design enables it to be implemented on edge devices and run in real time. The main contributions of the proposed method are as follows:

1) A lightweight neural network model is constructed for describing the mapping relationship from manipulator's action space to the motion trajectory of the end of a deformable object for specific experimental scenarios.

2) A perception transformation module is proposed as a way to bridge the gap between deformable objects with different properties and between simulation and real environment, such design avoids re-training of neural network after the property changes in deformable objects or environment.

3) An iterative control strategy is proposed that can be deployed on edge-side devices and can operate in real-time, dynamically adjusting the movements of a robotic arm as a means of manipulating the trajectory of a deformable object to reach a certain target point.

The subsequent organization of the article is as follows: Section 2 presents previous relevant work, while Section 3 constructs the research question. Subsequently, the specific implementation of IPP is developed in Section 4. Section 5 analyzes the experimental results of IPP in both simulator and real scenarios, and Section 6 is the conclusions.

2 Related works

Manipulation of deformable objects is a complicated and crucial research area in robotics with a wide range of applications in areas such as healthcare, manufacturing and autonomous systems[15–17]. Various control strategies have been proposed to address this challenge, which can be broadly categorized into conventional control methods, optimal control and heuristic methods and data-driven techniques.

2.1 Conventional control methods

Traditional approaches to deformable object manipulation are deeply grounded in control theory, with several methods sharing the common objective of performance optimization despite their different implementations and trade-offs. One popular method is the linear quadratic regulator (LQR), which is designed to minimize a quadratic cost function in linear systems[18, 19]. Although LQR provides a theoretical solution for systems with linear dynamics, its direct application to deformable objects is often difficult due to their inherent nonlinearity and high-dimension information of these systems[20]. To address these limitations, researchers have developed enhanced variants, such as time-varying LQR, which performs local linearization of the nonlinear dynamics[21]. This kind of approaches allow the control strategy to retain optimal solution while adapting to the evolving state of the system, thus bridging the gap between linear optimal control and the complicate inherence in deformable object manipulation[22].

Model predictive control (MPC) is another type of the control paradigm. MPC formulates the control problem as an optimization task over a finite prediction horizon, where the objective is to optimize a performance metric while satisfying state and input constraints[23]. Such framework naturally supports nonlinear dynamics via continuously updating the optimization problem based on real-time measurements[24]. In the context of deformable object manipulation, MPC can efficiently handle the system's limitations and complex behavior, thereby providing a more flexible and adaptive control solution[25]. However, this benefit comes at the cost of increased computational complexity, which creates substantial challenges for real-time implementation, particularly in high dimension applications[5].

Sliding mode control (SMC), is a nonlinear control method, could ensure robust operation in the face of uncertainties and external disturbances[26]. SMC achieves this objective via forcing system state to converge and remain on a specific sliding surface, substantially reducing the impact of model inaccuracies and external perturbations[27]. In applications that involve deformable objects, this robustness is particularly valuable; however, the inherent phenomenon that rapid oscillations are around the sliding surface, can impair performance and reduce control smoothness[28]. Consequently, various modifications and extensions of SMC have been proposed to reduce

chattering while maintaining its desirable robustness properties[29].

There are also some iterative control frameworks to retrieve control input that maximize performance metrics[30]. These techniques have presented promise in achieving precise control but often face scalability and real-time capability challenges in high-dimensional and complex environments[31–33]. Optimization-based iterative control frameworks are also frequently used with some success for the control of objects with unknown dynamics[34–36]. In contrast, heuristic control approaches use predetermined rules or task-specific insights to direct manipulation[37]. These approaches are frequently motivated by physical dynamics or actual observation, providing computationally efficient solutions for less complex cases[38, 39]. While heuristic methods can deliver satisfactory performance in structured environments, their adaptability to dynamic and highly variable conditions remains limited[40].

2.2 Data-driven techniques

Deep learning has evolved in recent years as a useful technique for manipulating deformable objects, employing data-driven models to address complicated and nonlinear dynamics without the need for explicit mathematical modeling[41]. Reinforcement learning (RL)-based procedures have shown particular effectiveness in learning control policies through repeated interaction with the environment, producing outstanding outcomes in tasks such as grabbing, folding and stretching flexible materials[42]. This framework leverages the ability to model complex and nonlinear dynamics in a data-driven manner without relying on explicit mathematical formulas. The iterative structure of RL facilitates adaptation to varying task circumstances and uncertainties, making it a promising candidate for solving the issue associated with deformable objects manipulation[43]. Some methods utilize image processing to obtain the state of the system, then learn via combining the experts′ operation data, and interact with the environment until a satisfying result is retrieved[41, 44, 45]. Another kind of RL-based approaches focus more on the design of the agent′s action space and value function, which could standardize the predicted operation and find the optimal action under specific state[46, 47].

Moreover, deep learning methods integrating convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have indicated effectiveness in processing multi-modal sensor inputs, such as vision and force feedback, enabling precise perception and control in complex tasks[48–52]. These architectures are constructed to feature extraction and temporal modeling, which are critical for understanding and manipulating deformable objects with time-varying properties. One of these methods is direct end-to-end modeling, which creates a mapping from robot action execution to acquired image[48, 53]. Another popular approach is to design a latent space, decrease the system′s feature from high dimension to low dimension space, and then construct a system dynamics mapping in such space[54, 55]. Despite their potential, data-driven approaches face significant limitations, including the requirement for large amounts of training data, limited adaptability, and poor generalization to new environments or objectives. However, their ability to describe highly nonlinear and high-dimensional systems provides an exciting possibility for future advances in deformable object manipulation.

3 Problem setup

The purpose of this study is to promote the development of manipulator in the field of healthcare, especially the manipulation of deformable objects utilizing artificial intelligence and advanced control methods. To achieve this goal, a prototype scenario is constructed here. The details are illustrated in Fig. 1, there is a manipulator

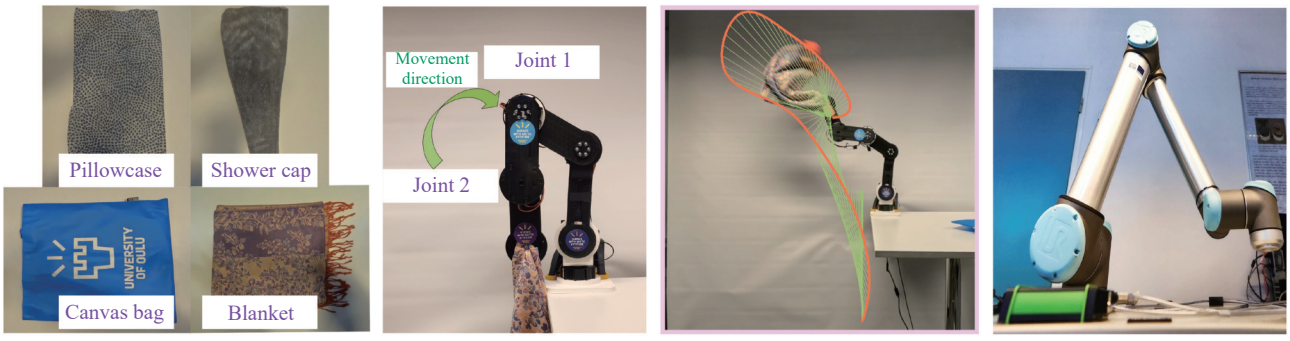


Fig. 1 Schematic diagram of the experimental scenario. (a) Four deformable textiles used for the experiment with different properties: a pillowcase, a shower cap, a canvas bag and a blanket, respectively. (b) The main body of the manipulator utilized for the experiment is in the center, with two movable joints. (c) The robot flings the textile as presented in the diagram on the right, and the movement of the manipulator changes the fabric′s end trajectory. (d) The lab UR5 robotic arm, which is not used in the real scenario experiments in this paper in order to better validate the stability and robustness of the algorithm, and the proposed method′s deployment performance on a much lower performance robotic arm. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

capable of grasping a corner of deformable fabric (e.g., a towel) and subsequently propelling the diagonal end of the towel through a specified spatial location at high speed, which will be achieved via rational joint trajectory design and dynamics control. In contrast to rigid-body manipulation, this process entails the control of a deformable fabric with an estimated infinite number of degrees of freedom. The motion of the robotic arm induces fluctuations, local strains and nonlinear deformations in the towel, significantly deviating from the behavior of a simple model of a rigid object.

In order to validate the effectiveness of the proposed method, a robotic arm assembled with robotic joint motors that are capable of delivering a maximum of 17 Nm is utilized as the experimental subject. In this paper, instead of applying the commercial UR5 robotic arm as shown in Fig. 1, the deployment is carried out on a much lower performance robotic arm with the aim of better testing the stability and robustness of the proposed method. The utilized manipulator has two movable joints, one is joint 1, located above to simulate a human shoulder, and the other one is joint 2, located below to mimic human wrist. The movable range of joint 1 is $[0^\circ, 120^\circ]$, while joint 2 is $[0^\circ, 90^\circ]$. Position loop of the actuator motors is applied for control, and hence robot's action space can be obtained as $a = (J_1, J_2)$. In addition, four textiles with different properties are used in this paper to test the reliability of the proposed algorithm, respectively pillowcase, shower cap, blanket and canvas bag. A segment of the deformable textile is fixed at the end of the robotic arm, which flings the textile through joint movements and then applies a key point algorithm to obtain the trajectory $T = \mathcal{F}(a)$ of the diagonal end of the textile, in which $\mathcal{F} : \mathbf{R}^2 \to \mathbf{R}^{2\times 1\,000}$ present the observed mapping from action to trajectory space. The goal of the task is to move the cloth's end trajectory past a predetermined target point P^*. The error distance metric $\mathcal{D}(T, P^*)$ is then retrieved by spline interpolation to obtain the shortest distance of the trajectory from the target point, and the objective of this problem is described as follows:

$$\begin{aligned} &\min \quad \mathcal{D}(T, P^*) = T(t^*) - P^* \\ &t^* = \arg\min \|\mathcal{F}(a, t) - P^*\| \\ &\text{s.t.} \quad J_1 \in [0^\circ, 120^\circ], \quad J_2 \in [0^\circ, 90^\circ]. \end{aligned} \tag{1}$$

4 Approach

When people try to swing various fabrics, it is always difficult to reach a satisfactory trajectory or obtain the target position on the first attempt. However, once a person has systematically tried and trained, he or she could fling a given fabric to a specific position in a limited number of attempts. Not only that, even when the type of fabric changes, they could achieve the goal in fewer attempts via relying on the perception. Gymnasts, for instance, trained in dance ribbons, are capable of adapting swiftly through their perceptions even when ribbons change.

Conventional methods utilize explicit modeling, require more detailed dynamics models, and their performance is closely related to the quality of the linearization method, while often relying on manual design or optimization of the error-to-action mapping. Such methods have a wide application base and excellent performance in tasks with adequate models and relatively stable environmental conditions. Data-driven methods, on the other hand, use end-to-end learning, do not rely on explicit models, learn system laws directly from data, do not require linearization, and can automatically establish error-to-action mappings. When dealing with complex and nonlinear behaviors, the data-driven approach shows better adaptability, especially in the scene of manipulating flexible objects, and is able to achieve a certain degree of adaptive adjustment through interaction. The two types of methods have their own advantages, but for the nonlinear manipulation task of flexible objects studied in this paper, the data-driven method is more suitable for its needs in practice.

According to the ability to perceive deformable objects, the IPP (as shown in Fig. 2) is proposed for manipulating different kinds of deformable objects to achieve target objective. Assuming T_i as the trajectory of cloth end while manipulator's i times action is a_i. After the action a_i of the manipulator at i times, the trajectory T_i could be obtained. The current trajectory T_i and 0 delta action, as inspired by [44], are then input into the delta perception transformer to acquire the predicted trajectory $\hat{T}_i$. The real and predicted trajectories will be utilized to obtain the parameters $A = \{s, R, t\}$ of perception transform, and target point P^* is transformed into the perceptual space $\hat{P}^*$ according to the parameters. Since then, the delta action space is sampled according to the normal distribution, and the distance metric between the predicted trajectories and target point in the perception space $\hat{P}^*$ is performed via delta perception transformer, and the closest distance one is selected as the optimal delta action δa_i^*. Finally, a new round of action a_{i+1} will be updated and move to the next iteration until the target point P^* is achieved or the iteration is stopped. Sections 4.1–4.4 illustrate each essential part of the proposed algorithm.

4.1 Delta perception transformer

Delta perception transformer is a neural network with an encoder-decoder architecture with transformer as the backbone. Current trajectory T_i, the action a_i perform to generate the trajectory, and the delta action δa_i is the inputs of delta perception transformer. The output of the network is a predicted trajectory based on the above in-

puts. During each iteration, the network is utilized to obtain the parameters of perception transform, as well as to generate predicted trajectories with random sampling delta actions.

4.1.1 Trajectories and delta actions

The trajectory used in the proposed method is a two-dimensional set of points located in the x-z plane of length 1 000, i.e., $T_i = \{p_i^j \mid j = 1, \cdots, 1\,000\}$. Specifically, while the manipulator is performing the action, the camera will capture a total of 120 frames of images at a frequency of 60 Hz, and the point set of 120 trajectories is acquired by the key target tracking algorithm (CoTracker3[56]). Subsequently, the trajectory is extended to 1 000-point sets via bicubic interpolation method. Finally, the acquired pixel trajectories are mapped into domain of $[-1, 1]$. In the simulation environment, 1 000 simulation steps are utilized and the obtained trajectories are also mapped between $[-1, 1]$.

In order to enable the neural network to acquire better results, the underlying action a_i space of both manipulator joints is regularized to the $[0, 1]$ space. In addition, to the delta action δa_i of the two joints is obtained via sampling two Gaussian distributions with mean 0 and standard deviations of 0.035 and 0.023 respectively. After acquiring the data needed for model input, the action a_i and delta action δa_i data will be concatenated and then fed into delta perception transformer along with the trajectory data.

4.1.2 Network and loss

As is indicated in Fig. 3, the delta perception transformer is a transformer-based architecture specifically designed to predict trajectory increments (delta action) in sequences. Its input consists of information about the current trajectory and action changes, which are processed through an input embedding and positional encoding layer before being fed to the transformer encoder to extract contextual features. The encoder output is then passed through a transformer decoder with a mask, which com-

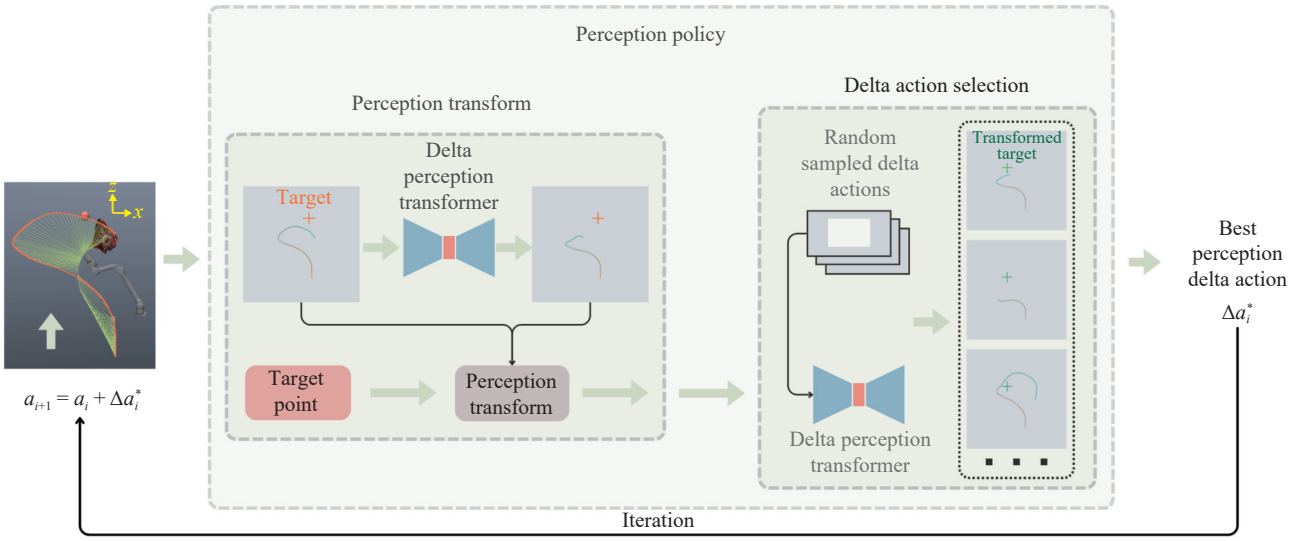


Fig. 2 The framework of iterative perception policy. In each iteration of the policy, the trajectory of the cloth end is obtained through the camera and converted into coordinate points in the $x - z$ plane. The actual trajectory is then predicted by the delta perception transformer, and the parameters of the perception transformer are calculated from these two different trajectories. After the target point is transformed in perception, the optimal delta action is selected for the next iteration by random delta action sampling and combining the transformer prediction results. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

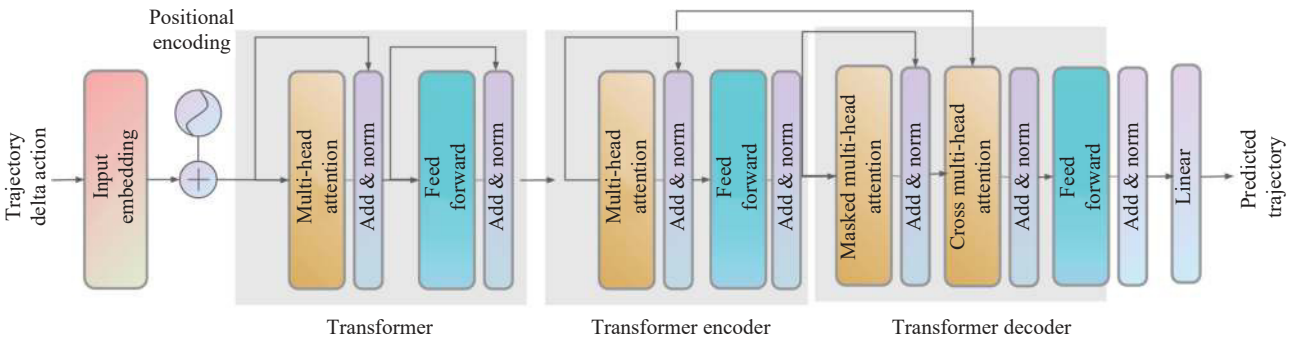


Fig. 3 The architecture of delta perception transformer, a transformer-based neural network for predicting trajectory according to delta actions. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

bines historical information and encoder memory to gradually generate a predicted sequence. Finally, through a linear layer, the model outputs the next trajectory prediction. The network parameters of delta perception transformer are illustrated in Table 1. The input to the network consists of two variables, a trajectory of dimension (batch size, 1 000, 2), and an action context of dimension (batch size, 4). After the trajectory is input to the model, it will first go through a linear embedded layer that maps it into a feature tensor with the dimension (batch size, 1 000, 128). The action context also passes through a linear embedding layer, which is extracted into the feature space and subsequently concatenated with the trajectory feature tensor after being expanded. The spliced tensor will feed into the positional encoding layer and then into the transformer encoder layer for further extraction of the encoded spatial feature tensor. The encoded tensor is in turn decoded by the transformer decoder and then mapped back into the trajectory space by a linear connection layer.

The network is trained using simulated data obtained from MUJOCO[57] and the mean squared errors between the predicted and real trajectory as well as smoothness constraint are applied as the loss, which is formulated as

$$\mathcal{L} = \underbrace{\frac{1}{BN}\sum_{b=1}^{B}\sum_{i=1}^{N}(\hat{p}_{b,i}-p_{b,i})^2}_{\text{MSE}} + \underbrace{\lambda\frac{1}{B(N-1)}\sum_{b=1}^{B}\sum_{i=2}^{N}\|\hat{p}_{b,i}-\hat{p}_{b,i-1}\|}_{\text{Smoothness constraint}} \tag{2}$$

where B represents the training batch size while λ means the weight of smoothness constraint. The AdamW[58] with learning rate 0.001 and weight decay of 1×10^{-6} is utilized during training process as model parameter optimizer.

4.2 Perception transform

Perception transform is a mapping method for transforming real spatial data into perceptual space. Consistent with human perception, it generates a set of transformation parameters $A=\{s,R,t\}$ by analyzing the correlation between real and predicted trajectories, which in turn maps the target points P^* into the perceptual space $\hat{P}^*$. Assume the relationship between real and predicted trajectories as follows:

$$p_i^{nn} = sRp_i^{real}+t, \quad s\in\mathbb{R},\ R\in\mathbb{R}^{2\times2},\ t\in\mathbb{R}^2 \tag{3}$$

where s is the scale factor, R means the rotation matrix while t indicates the translation. In order to obtain these transform parameters, the pre-process for removing the influence of data difference is applied here as

$$\begin{aligned}\hat{p}_{real} &= p_{real}-\frac{1}{N}\sum_{i=1}^{N}p_i^{real}\\ \hat{p}_{nn} &= p_{nn}-\frac{1}{N}\sum_{i=1}^{N}p_i^{nn}.\end{aligned} \tag{4}$$

An iterative form of solving for these transformation parameters is utilized here, where a metric function $E(A)$ is constructed here to evaluate the merits of these variables:

$$E(A)=\sum_{i=1}^{N}\left\|p_i^{nn}-(sRp_i^{real}+t)\right\|^2 \tag{5}$$

and in each iteration, the rotation matrix R is found by singular value decomposition:

$$\begin{aligned}U\Lambda V^{\mathrm{T}} &= \sum_{i=1}^{N}\hat{p}_i^{real}(\hat{p}_i^{nn})^{\mathrm{T}}\\ R &= VU^{\mathrm{T}}.\end{aligned} \tag{6}$$

The scale factor s is calculated as

Table 1 The structure of delta perception transformer

Layers	Input shape	Output shape	The number of parameters
Delta perception transformer	[(1, 1 000, 2), (1, 4)]	(1, 1 000, 2)	927 490
Trajectory embedded	(1, 1 000, 2)	(1, 1 000, 128)	384
Context embedded[1]	(1, 4)	(1, 128)	640
Positional encoding	(1 000, 1, 128)	(1 000, 1, 128)	–[2]
Transformer encoder	(1 000, 1, 128)	(1 000, 1, 128)	396 800
Transformer decoder	(1 000, 1, 128)	(1 000, 1, 128)	529 408
Output layer	(1, 1 000, 128)	(1, 1 000, 2)	258

[1] The output of context embedded layer will be expanded as the same dimension with the output of trajectory embedded layer and then processed for concatenating. [2] The non-parametric sinusoidal positional encoding method is utilized in this paper.

$$s = \frac{\sum_{i=1}^{N}[\hat{p}_i^{nn} \cdot (R\hat{p}_i^{real})]}{\sum_{i=1}^{N}\left\|\hat{p}_i^{real}\right\|^2}. \tag{7}$$

The translation t is obtained by

$$t = \frac{1}{N}\sum_{i=1}^{N} p_i^{nn} - sR\frac{1}{N}\sum_{i=1}^{N} p_i^{real}. \tag{8}$$

The evaluation function is calculated once at the end of each iteration until the evaluation function value satisfies the condition, or the maximum number of iterations is reached and then iteration ends. Subsequently, target point P^* in real space will be mapped into perceptual space $\hat{P}^*$:

$$\hat{P}^* = sRP^* + t. \tag{9}$$

The acquired target points in the perceptual space will be involved in the selection of delta action, which is utilized to evaluate the benefits from different delta actions.

4.3 Iterative delta action selection

At the initial phase of the fabric flinging task, the manipulator first runs the task 5 times with randomized actions and the textile end trajectories under these actions are recorded. After the target position is set, the distance between each existing trajectory and the target position is measured by spline interpolation, and the action of the closest trajectory is selected as the initial action a_0. The initial trajectory T_0, the initial action a_0, and the 0 delta action $\delta a_0 = 0$ are then input into the delta perception transformer to obtain its predicted trajectory $\hat{T}_0$. Thereby the parameters A of perception transform can be retrieved from the real and predicted trajectories, and thus the target position P^* in the real space will be mapped to the perception space $\hat{P}^*$.

In addition, the actions of the two joints in manipulator will be mapped into the interval $[0, 1]$ based on the range of motion $[0, 120°]$ of the first joint and $[0, 90°]$ of the second joint. The delta action $\delta a_{i,j}$ of the two joints will be obtained by random sampling in two Gaussian distributions with mean 0 and standard deviation 0.035 and 0.023, respectively. All delta actions are passed through the delta perception transformer to retrieve the predicted trajectories $\hat{T}_{i,j}$, and the distance $\mathcal{D}_{i,j}(\hat{T}_{i,j}, \hat{P}^*)$ to the target point in the perception space is calculated for each trajectory. Then the delta action δa_i^* corresponding to the predicted trajectory with the shortest distance $\text{argmin}(\mathcal{D}_{i,j})$ is selected as the action input increment, and the action $a_{i+1} = a_i + \delta a_i^*$ of the robotic arm at the next moment is calculated, which is described as

$$\begin{aligned}\delta a_i^* = &\arg\min_j \left\| f_{nn}(a_i, \delta a_{i,j}) - \hat{P}^* \right\| = \\ &\arg\min_j \left\| f_{nn}(a_i, \delta a_{i,j}) - sRP^* - t \right\|. \end{aligned} \tag{10}$$

In the experiments, this paper adopts $d \leq 0.02\,\text{m}$ as the policy termination condition, and the maximum number of iterations is limited to 20. The pseudo code is illustrated in Algorithm 1.

Algorithm 1. Iterative perception policy

Input: Target position P^*

Initialize action a_0 // Section 4.3
while $t < t_{max}$ **do**
  $T_i = experiment(a_i)$
  $d_i = \mathcal{D}(T_i, P^*)$
  if $d_i \leq d_{stop}$ **then**
    Break
  end if
  $\hat{T}_i = \text{transformer}\ (T_i, a_i, \mathbf{0})$
  $A = \text{perception}\ (\hat{T}_i, T_i)$ // Section 4.2
  $\hat{P}^* = \text{transform}\ (A, P^*)$ // Section 4.2
  $\delta a_{i,j} = \text{sampling}\ ()$ // Section 4.3
  $\hat{T}_{i,j} = \text{transformer}\ (T_i, a_i, \delta a_{i,j})$
  $j^* = \text{argmin}_j\ (\mathcal{D}(\hat{T}_{i,j}, \hat{P}^*))$
  $a_{i+1} = a_i + \delta a_i^*$
end while

4.4 Training and system setup

In order to retrieve enough training data, this paper builds a simulation environment in MUJOCO, which applies Unitree Z1 as the experimental robotic arm and a square deformable cloth as the flinging object. The manipulator has two movable joints, where the motion range of first joint is $[0, 120°]$ while the second joint is $[0, 90°]$. The initial position of each joint is 0, and the flinging action is performed via setting the target value of the position loop. The specifics of the simulation environment are indicated in Fig. 4, with a Unitree Z1 manipulator with a piece of textile hanging from its end at the zero position. In the initial situation, the textile is vertically downward, and when the two joints of the arm move, the textile will be thrown up, thus producing different trajectories at its end. During data collection, the movable space of both joints was divided equally into 300 grids, and a total of 300^2 actions were collected and all the trajectories of the fabric were recorded. And in training process, each trajectory is randomly matched to 20 nearby trajectories in a data pair, so hence 20 grid distances are set to 3σ, and two normal distributions of joint actions with standard deviation of 0.035 and 0.023 are retrieved, respectively. The ratios of the training and test sets are 0.8 and 0.2 each, and the test set is obtained via sampling the average interval in all the data area.

5 Experimental results

In the experimental part, this paper validates and compares the proposed algorithms through simulation and real environment experiments respectively. The simulation part will test the robustness and accuracy of the proposed method through multiple points, while the real environment part will evaluate the adaptability and generalization ability of the method through deformable objects with different attributes.

To validate the effectiveness of the proposed method, a comparative study was conducted with four representative controllers: LQR[20], SMC[29], MPC[59] and heuristic[60] method. The specific parameter settings and control law designs are shown in Table 2. Since the system included in this scenario is nonlinear, the linearized system function is used here, and the parameters with better results are selected for experiments based on multiple attempts. All methods share the same input: The normalized joint angles x and the error distance d, which is used to form the error vector $e = [d,\, d]^{\mathrm{T}}$ (with MPC directly taking the target point as the objective state). Based on these inputs, different control strategies are applied to generate the control input. In the LQR approach, the controller computes the optimal feedback gain K by solving the continuous algebraic riccati equation (CARE) with matrices A, B, Q and R, and then applies the control law $u = -K\,e$ to update the state using the linear model. In contrast, the MPC method uses a finite-horizon prediction model and formulates a quadratic cost function to optimize the control sequence, of which only the first control input u_0 is applied. The SMC strategy

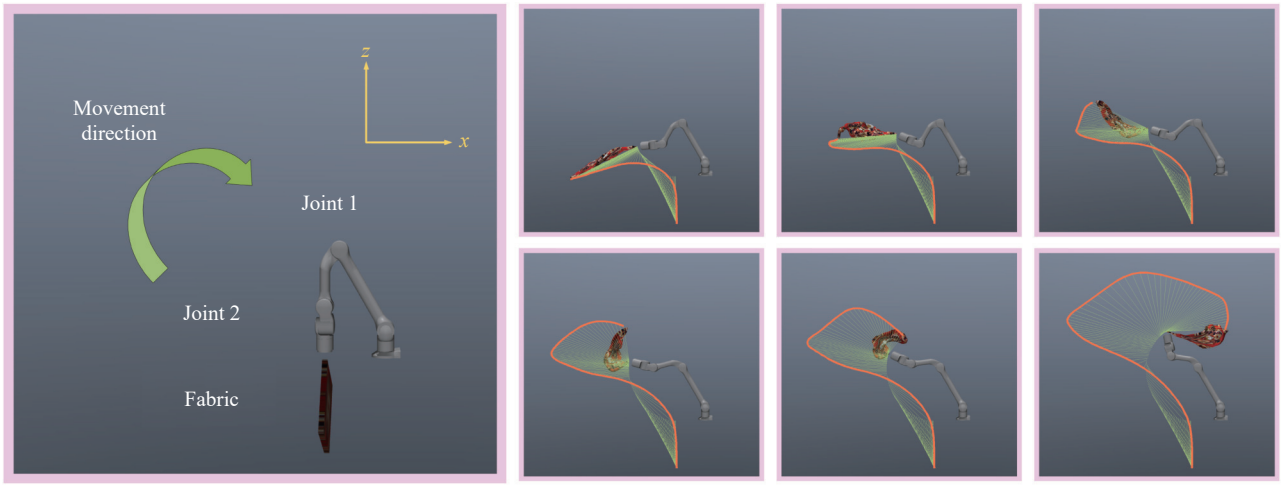


Fig. 4 Schematic diagram of the simulation environment. The main characteristic of the simulation environment is a Unitree Z1 robot located at zero point with a deformable textile hanging from the end of the robot. In the initial state of the robot arm, the textile is vertically downward, and as the two joints of the robot arm move, the textile is thrown with the force generated by the robot arm, so that the end of the textile appears to have different trajectories. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

Table 2 The control law and parameter setting in each compared method

Methods	Control law	Parameters
LQR[20]	$u = -K\,e,\ e = [d,d]^{\mathrm{T}},\ K = R^{-1}B^{\mathrm{T}}P,$ $A^{\mathrm{T}}P + PA - PB\,R^{-1}B^{\mathrm{T}}P + Q = 0.$	$A = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 \end{pmatrix},\ B = \begin{pmatrix} 0 & 0 \\ 0.15 & 0.15 \end{pmatrix},$ $Q = \mathrm{diag}(1, 0.1),\quad R = 0.01.$
MPC[59]	$x_{t+1} = A\,x_t + B\,u_t,\quad J = \sum_{t=0}^{N-1}\left[(x_{t+1} - P^*)^{\mathrm{T}}Q(x_{t+1} - P^*) + u_t^{\mathrm{T}}R\,u_t\right].$	$A = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 \end{pmatrix},\ B = \begin{pmatrix} 0 & 0 \\ 0.1 & 0.1 \end{pmatrix},$ $Q = \mathrm{diag}(1, 0.1),\quad R = 0.01,\quad N = 10.$
SMC[29]	$e = [d,d]^{\mathrm{T}},\quad s = Qe,$ $\eta = \eta_{\min} + (\eta_{\max} - \eta_{\min})\,(1 - \exp(-\lVert e\rVert)),$ $u_{\mathrm{eq}} = -B^{\dagger}(Ax),\quad u_{\mathrm{dis}} = -\eta\ \mathrm{sat}(s/\phi),$ $u = \mathrm{clip}\,(u_{\mathrm{eq}} + u_{\mathrm{dis}}, -1, 1)\,.$	$A = \begin{pmatrix} 1 & 0.05 \\ 0 & 1 \end{pmatrix},\ B = \begin{pmatrix} 0 & 0 \\ 0.02 & 0.02 \end{pmatrix},$ $Q = \mathrm{diag}(1, 0.1),\quad \eta_{\min} = 0.1, \eta_{\max} = 1.0.$
Heuristic[60]	$\delta = (\mathrm{rand}(2) - 0.5) \times s,$ $x_{i+1} = x_i + \delta,$ $s \leftarrow c \times s.$	Step size $s = 0.5,$ cooling rate $c = 0.9.$

maps the error to a sliding surface $s = Qe$ and calculates an adaptive sliding gain. It then combines an equivalent control $u_{\text{eq}} = -B^{\dagger}(Ax)$ with a discontinuous control component to form the final control input. Finally, the heuristic method generates the current state via adding a random offset to generate a candidate state.

5.1 Simulation environment

In the simulation section, the proposed algorithm is firstly evaluated on multiple target points so as to verify the validity and reliability of the algorithm, and then compared with various algorithms to highlight the advantages of the algorithm.

5.1.1 Multi-target points validation

Target points located in four directions are selected for the experiment: below the left, above the left, directly above and to the right of the robot. The results of the experiment are presented in Fig. 5.

The convergence graphs on the left indicate a consistent reduction in the distance between the deformable object's endpoint trajectory and the target point over successive iterations. In task 1, the initial distance is approximately 15 cm. The method exhibits steady, near-linear convergence, reaching close to 0 cm after 9 iterations. In task 2, where the initial distance is around 20 cm, the distance decreases with fluctuations but ultimately converges to under 5 cm by the final iteration. Notably, in tasks 3 and 4, the method achieves superior convergence performance. Starting at 20 cm and 15 cm respectively, the distance drops sharply to near 0 cm within the first 3rd–5th iterations, demonstrating rapid convergence and computational efficiency. These results highlight the method's ability to consistently minimize error distances across different initial conditions while maintaining fast and efficient convergence.

The visualization of iterative processes on the right reveals the progressive refinement of the deformable object's endpoint trajectory. In the initial iterations, the orange trajectory lines show significant deviation from

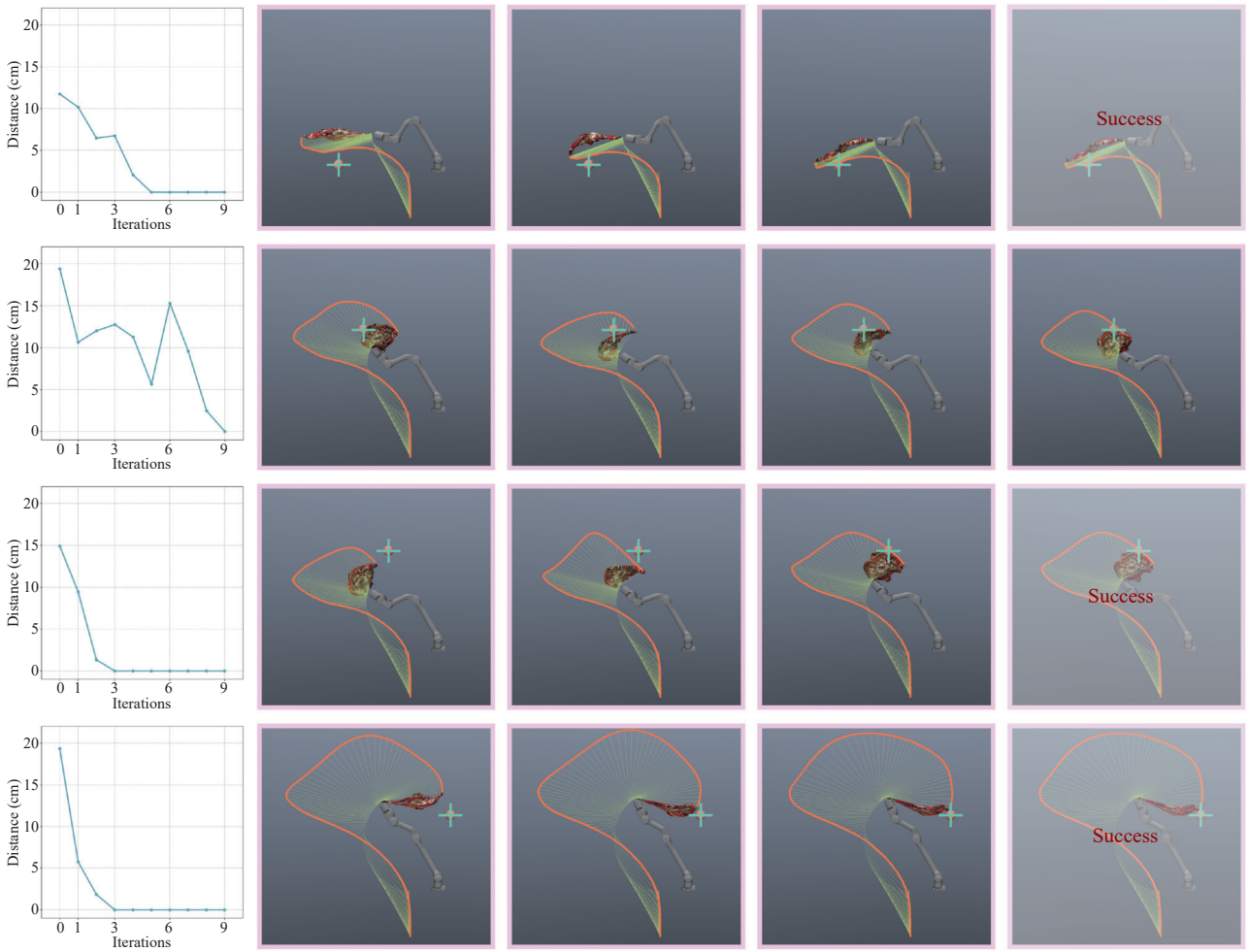


Fig. 5 Multi-target points validation. The left side indicates the convergence plots of the error distances for different target points, and it can be found that most of the error distances converge to near 0 cm after 10 iterations. On the right side is the plot of textile end trajectories at different iteration moments, at first the trajectories are far away from the position of the target point, but after a few iterations, they all pass by or close to the target point. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

the target point (marked by the green cross). This reflects the limitations of the initial state control. As iterations progress, the endpoint trajectory gradually aligns with the target point. In tasks 3 and 4, the endpoint passes through or very close to the target point by the 3rd–5th iterations, with smoother and more stable trajectories observed. Even in the cases where the experiment terminates without reaching the exact target (marked "success"), the final errors are constrained to an acceptable range of approximately 1–3 cm, demonstrating high precision. These findings validate the method's ability to iteratively optimize control inputs, achieving accurate and smooth trajectory tracking for deformable object manipulation.

5.1.2 Algorithm comparison

In order to validate the performance advantage of IPP, here IPP is compared with other four commonly utilized advanced algorithms (LQR, MPC, SMC and heuristic), and again multiple target points are selected for the experiment, and the results are indicated in Fig. 6.

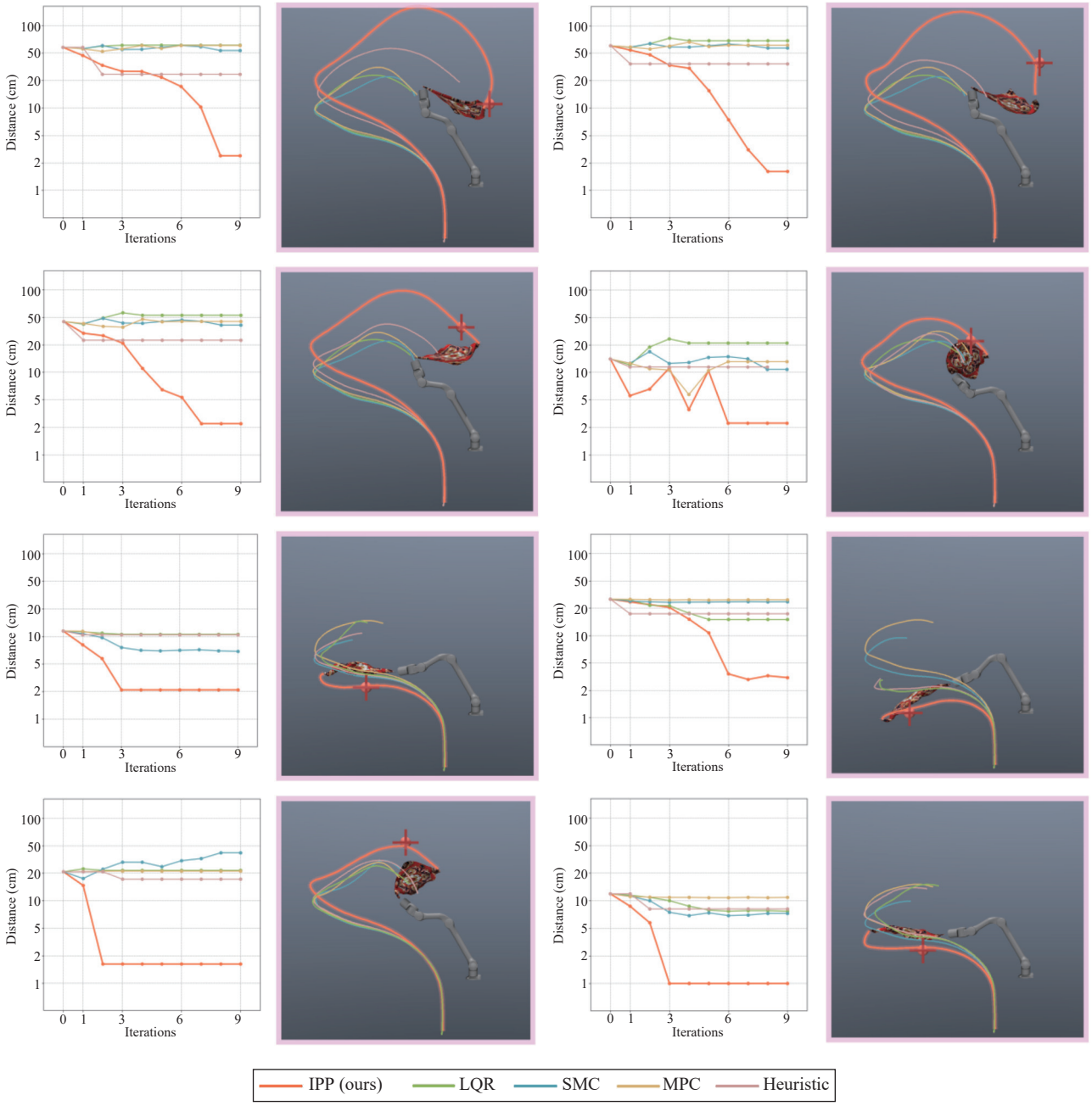


Fig. 6 Comparison diagram of each algorithm – IPP(ours), LQR[20], SMC[29], MPC[59] and heuristic[60], in simulation environment. The left side illustrates the convergence of the error distance of the textile end trajectories from the target point under the control of each algorithm, and it could be found that IPP is substantially ahead. The right side is the plot of the end trajectory at the last iteration, and the IPP algorithm then mostly passes the target point at the end, while the other algorithms have a large gap. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

The convergence graphs on the left clearly demonstrate that the IPP method achieves significantly swifter and more effective distance error reduction compared to the other methods. Across all experiments, IPP method (orange): Starting from a relatively large initial distance, the distance error decreases dramatically within the first 3–5 iterations, achieving near-zero errors by the final iteration. This rapid convergence highlights the efficiency of the proposed approach. LQR (green) and SMC (blue): Both methods exhibit nearly constant distance error throughout the iterations, indicating limited performance in reducing the endpoint-target distance. MPC method (yellow): While showing some improvement in certain experiments, MPC′s convergence is relatively slow, and the final distance error remains notably higher than that of the IPP method. Heuristic method (brown): The error remains almost unchanged, demonstrating an inability to optimize the deformable object′s behavior effectively. Overall, the IPP method consistently outperforms the other methods in both error reduction and convergence speed, underscoring its computational efficiency and superior optimization capabilities.

The trajectory visualizations at the right side offer a clear comparison of endpoint trajectory under each control method. The trajectories from IPP exhibit a smooth and gradual convergence toward the target point. At the final iterations, the trajectories either pass through or closely approach the target, indicating high precision and effective control. The trajectories from LQR, SMC and MPC deviate significantly from the target point and fail to present meaningful convergence. The paths remain inconsistent, with notable errors persisting throughout the iterations, reflecting the limitations of these methods in handling complex nonlinear dynamics. Heuristic method displays little to no improvement over the iterations, remaining far from the target. This result highlights the ineffectiveness of heuristic control for deformable object manipulation tasks. The IPP method′s ability to iteratively guide the endpoint toward the target demonstrates its superior trajectory-control performance, particularly when managing the nonlinear behavior of deformable objects.

5.2 Real scene

In the real scenario experiments, four deformable textiles with different properties (pillowcase, shower cap, blanket and canvas bag) are experimented and then compared with other algorithms.

5.2.1 Various textiles experiment

Four types of fabrics with different properties are utilized here to evaluate the performance of the proposed algorithm in real scene. The results are illustrated in Fig. 7.

The convergence graphs indicate consistent and swift error reduction across different deformable objects, demonstrating the method′s efficiency. For the pillowcase and shower cap, the initial error exceeds 30 cm, but the distance reduces sharply within the first 3 iterations, demonstrating highly efficient convergence. Similarly, for the blanket and canvas bag, despite their greater initial error and structural complexity, the distance decreases steadily over the iterations. By iteration 3, the trajectory successfully approaches the target, verifying the method′s reliable performance across different materials and geometries. Despite the presence of real-world factors such as gravity, friction and external disturbances, the method maintains consistent and reliable performance. The error distance converges to the target across all objects cases, showcasing the system′s ability to remain robust under varying conditions.

The visualizations on the right reveal the improvement of trajectories across iterations, explaining the method′s trajectory optimization capabilities. For the pillowcase and shower cap, the initial trajectories (orange lines) exhibit noticeable deviations from the target (green cross). However, as iterations progress, the trajectories are iteratively improved, ultimately converging to the target position. In the case of the blanket and canvas bag, which are characterized by higher stiffness and complex deformations, the control task is more challenging. Nonetheless, the manipulator progressively adjusts the manipulation strategy, resulting in trajectories that approach the target with a high degree of accuracy. The smooth and continuous adjustments further confirm the method′s effectiveness in handling dynamic deformations. These observations confirm the method′s ability to precisely control endpoint trajectories and adapt to dynamic and uncertain real-world conditions.

5.2.2 Algorithm comparison

Advanced algorithms (LQR[20], SMC[29], MPC[59] and heuristic[60]) are also utilized here to validate the performance of the proposed IPP. The comparison results are indicated in Fig. 8. In addition, this paper compares order reduction LQR[61], MPC[62] and SMC[63] for a more detailed performance comparison with the proposed method. The specific form is as follows: the fabric is divided into 50 grids of length and width, totaling $n = 2\,500$ prime points. Then $q \in \mathbb{R}^{2n}$ and $\dot{q} \in \mathbb{R}^{2n}$ are the location and speed of prime points. $M \in \mathbb{R}^{2n\times 2n}$, $C \in \mathbb{R}^{2n\times 2n}$, $K \in \mathbb{R}^{2n\times 2n}$, $B_{full} \in \mathbb{R}^{2n\times 2}$ are mass matrix, damping matrix, stiffness matrix, and input matrix, respectively. Hence the dynamic equation is described as $M\ddot{q} + C\dot{q} + Kq = B_{full}u$. A modal decomposition and normalization approach is utilized, $\Phi = [\phi_1, \cdots, \phi_r] \in \mathbb{R}^{2n\times 2}$, $\Phi^{\mathrm{T}} M \Phi = I_r$. Then the matrix will be projected into the low-dimensional space M_r, C_r, K_r and B_r, and the system state A and input matrix B in the low-dimensional space can be obtained.

$$A = \begin{bmatrix} 0 & I \\ -M_r^{-1}K_r & -M_r^{-1}C_r \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ M_r^{-1}B_r \end{bmatrix}.$$

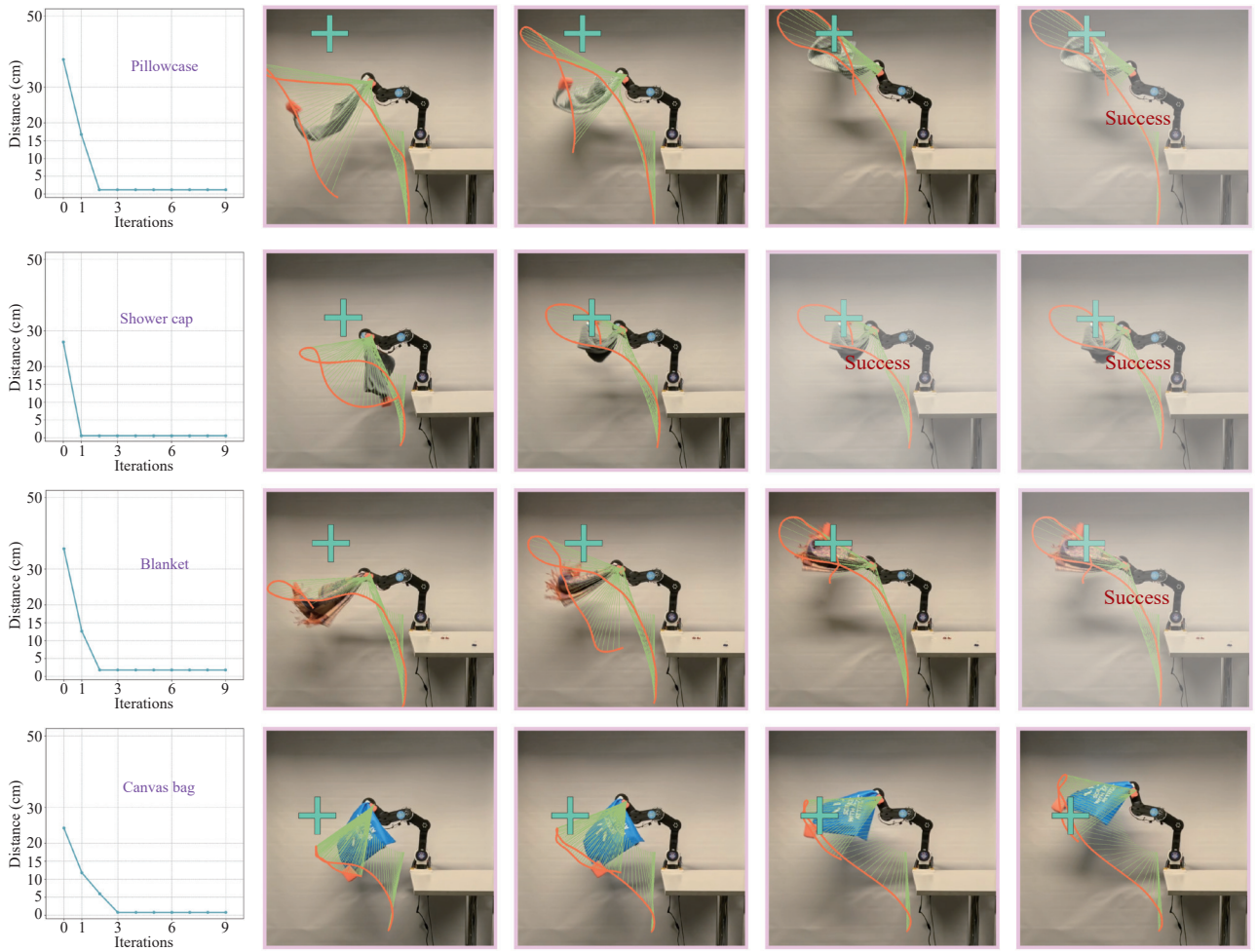


Fig. 7 Diagram of the proposed algorithm's performance on four different fabrics in real scene. The convergence plots on the left side indicate that the proposed algorithm is able to converge below 3 cm in each textile task. The iteration plot on the right side indicates that the trajectory of the textile is able to pass the target point after several rounds of IPP iterations. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

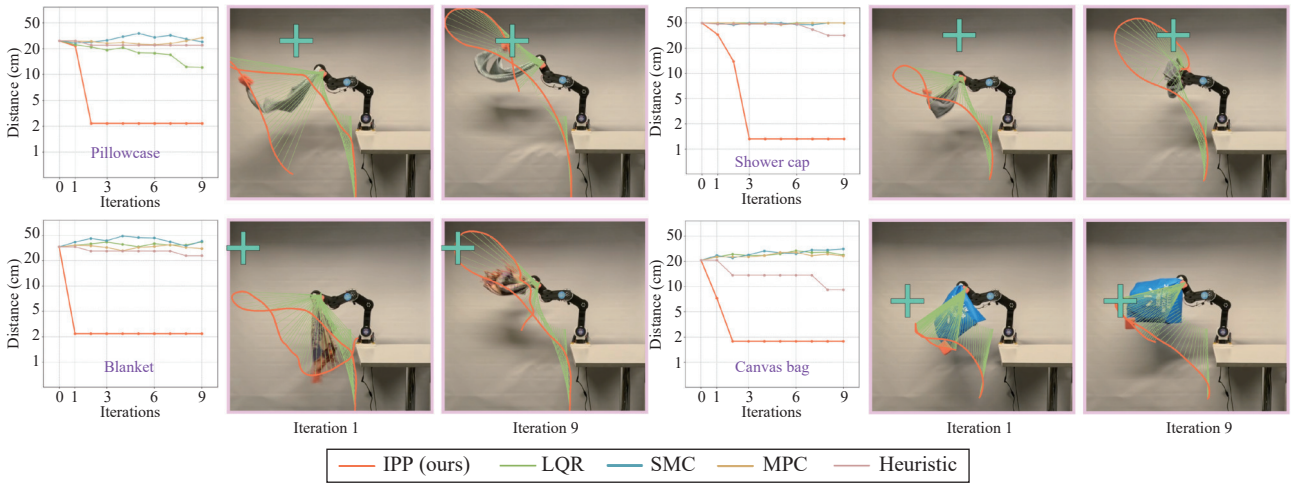


Fig. 8 Comparison diagram of each algorithm – IPP(ours), LQR[20], SMC[29], MPC[59] and heuristic[60] in real scene. The convergence figures on the left indicate that IPP converges swiftly and with far better results than the other algorithms. The iterative trajectory diagram on the right side illustrates that IPP is capable of allowing trajectory to pass by or close to the target point after iterations even when the initial position is away from the target point. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

The convergence graphs on the left provide a clear illustration of the reduction in error distance over iterations. The IPP method demonstrates superior convergence performance, with the initial error starting over 25 cm and rapidly decreasing to below 5 cm within 3 iterations. The error then stabilizes, maintaining precision for the remaining of the iterations. In contrast, the LQR and SMC methods present little reduction in distance error, with their curves remaining nearly flat, however LQR achieved better result in pillowcase experiment. Similarly, the MPC method exhibits a slower rate of convergence, with errors reducing gradually but still remaining significantly higher than those achieved by the IPP method. The heuristic method, meanwhile, shows almost no improvement, indicating a failure to optimize the endpoint trajectory effectively. These results confirm the IPP method′s efficiency in achieving rapid and consistent convergence compared to the other control strategies.

The experiments also highlight the robustness and adaptability of the proposed IPP method across diverse deformable objects and experimental conditions. Regardless of the initial state, object shape, or material properties, the IPP method consistently achieves convergence and precise trajectory control. Its performance remains stable even in the presence of external disturbances such as gravity and friction, which are unavoidable in real-world scenarios. In contrast, the alternative methods, particularly LQR, SMC and MPC, struggle to adapt to the nonlinear dynamics and environmental uncertainties, which leads to suboptimal performance. The heuristic method exhibits the greatest limitations, demonstrating poor adaptability and negligible trajectory optimization. These findings validate the IPP method′s strong robustness and generalization in handling deformable object manipulation.

5.3 Statistical analysis

In order to verify the effectiveness and utility of the proposed method, a total of 100 experiments were done in experimental and real scenarios, and the specific statistical results are shown in Table 3. Six metrics are utilized in Table 3, which are successful rate, average iterations, convergence speed per iteration, average distance convergence, average error distance and standard deviation of error distance. Since most of the current work is on the manipulation of 2D or linear deformable objects, and there are fewer cases for 3D deformable object manipulation, in this paper, the neural network and iterative policy in the proposed method are also utilized as one of the comparison methods.

From the results, IPP (this paper) consistently outperforms all baseline methods and the simple version without the perception transform (“this paper without perception transform”) on every metric. Notably, “this paper” achieves a success rate of 78.57%, considerably higher than the simple version (55.36%) and far exceeding all other baselines (each below 32%). This substantial difference underscores the pivotal role of the perception transform in enhancing task success. In terms of average iterations, IPP converges in 5.13 iterations, which is lower than both “this paper without perception transform” (8.60) and the remaining baselines (all above 7.54). This efficiency is further reflected in the convergence speed per iteration: At 2.22 cm, it indicates that each iteration under IPP reduces the error distance more effectively. The superior convergence behavior is also evidenced by a lower average distance convergence (7.31 cm) and a smaller final average error distance (3.88 cm) than all other methods, demonstrating both faster and more accurate control. As reported in the bottom three rows of Table 3, incorporating an order reduced model into the classical controllers does yield the modest gains over their conventional counterparts (e.g., MPC′s success rate rises from 17.9% to 31.0%, and LQR's success rate rises from 14.3% to 27.6%), and their average iterations change slightly (from 7.8–8.0 steps to 7.1–8.7 steps). However, these order reduced schemes still fall far short of our proposed method′s performance 78.6% success rate, just 5.1 iterations on average, 2.22 cm per step convergence speed.

Table 3 The statistic comparison of each algorithm in six metrics

Methods	Successful rate (%)	Average iterations	Convergence speed per iteration (cm)	Average distance convergence (cm)	Average error distance (cm)	Std. error distance (cm)
This paper	78.57	5.13	2.22	7.31	3.88	3.13
This paper without perception transform	55.36	6.57	1.21	4.69	6.50	5.04
LQR[20]	14.29	8.00	0.53	2.10	9.09	4.52
MPC[59]	17.85	7.80	0.50	1.74	9.45	4.60
SMC[29]	28.57	7.54	0.61	2.53	8.66	4.24
Heuristic[60]	31.25	7.88	0.71	2.42	6.12	3.40
Order reduced LQR[61]	27.59	7.69	1.01	5.05	6.21	4.22
Order reduced MPC[62]	31.03	7.10	0.50	5.12	8.13	3.22
Order reduced SMC[63]	24.14	8.69	0.87	4.32	7.55	3.98

This gap highlights that, although modal reduction can improve classical controllers slightly, our end-to-end approach delivers substantially higher reliability, faster convergence, and greater robustness to parameter choices.

Moreover, IPP exhibits a relatively low standard deviation (3.13 cm) in the final error distance, indicating stable and consistent performance across all trials. In contrast, removing the perception transform component not only reduces the success rate but also increases the number of required iterations and worsens the final accuracy. These findings confirm that the perception transform module is integral to the robustness and precision of IPP, emphasizing its effectiveness in deformable object manipulation.

6 Conclusions

In this paper, an online IPP is proposed for the prototype application of deformable object manipulation in the medical field. The proposed algorithm could achieve real-time iterative control of deformable objects with a small amount of neural network training. The proposed algorithm is experimented in simulation and real environments, and the experimental results indicate that IPP has excellent performance in terms of convergence, adaptability and robustness. The work in this paper also provides some research basis for the measurement of deformable objects. In future work, IPP will be extended to be applied in 3D space, while the neural network architecture and perceptual transformation component will be improved to employ to a wider range of scenarios. Moreover, more practical scenarios will be applied to the deployment and will validate the stability and reliability of IPP.

Acknowledgements

Open access funding provided by University of Oulu, Finland.

Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

Open Access

References

[1] C. Dai, G. Shan, H. Liu, C. Ru, Y. Sun. Robotic manipulation of sperm as a deformable linear object. *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2799–2811, 2022. DOI: 10.1109/TRO.2022.3158200.

[2] C. Dai, Z. Zhang, Y. Lu, G. Shan, X. Wang, Q. Zhao, C. Ru, Y. Sun. Robotic manipulation of deformable cells for orientation control. *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 271–283, 2020. DOI: 10.1109/TRO.2019.2946746.

[3] R. Laezza, R. Gieselmann, F. T. Pokorny, Y. Karayiannidis. ReForm: A robot learning sandbox for deformable linear object manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, Xi'an, China, pp. 4717–4723, 2021. DOI: 10.1109/ICRA48506.2021.9561766.

[4] X. Liang, F. Liu, Y. Zhang, Y. Li, S. Lin, M. Yip. Real-to-sim deformable object manipulation: Optimizing physics models with residual mappings for robotic surgery. In *Proceedings of IEEE International Conference on Robotics and Automation*, Yokohama, Japan, pp. 15471–15477, 2024. DOI: 10.1109/ICRA57147.2024.10610263.

[5] L. Wijayarathne, Z. Zhou, Y. Zhao, F. L. Hammond. Real-time deformable-contact-aware model predictive control for force-modulated manipulation. *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3549–3566, 2023. DOI: 10.1109/TRO.2023.3286070.

[6] W. He, Z. Li, C. L. P. Chen. A survey of human-centered intelligent robots: Issues and challenges. *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 602–609, 2017. DOI: 10.1109/JAS.2017.7510604.

[7] P. Mitrano, A. LaGrassa, O. Kroemer, D. Berenson. Focused adaptation of dynamics models for deformable object manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, London, UK, pp. 5931–5937, 2023. DOI: 10.1109/ICRA48891.2023.10161366.

[8] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, F. Mathis-Ullrich. Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects. *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5338–5345, 2024. DOI: 10.1109/LRA.2024.3382529.

[9] F. Liu, E. Su, J. Lu, M. Li, M. C. Yip. Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics. *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3964–3971, 2023. DOI: 10.1109/LRA.2023.3264766.

[10] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, E. Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011. DOI: 10.1109/TRO.2011.2158251.

[11] B. P. Murphy, F. Alambeigi. A surgical robotic frame-

work for safe and autonomous data-driven learning and manipulation of an unknown deformable tissue with an integrated critical space. *Journal of Medical Robotics Research*, vol. 8, no. 01n02, Article number 2340001, 2023. DOI: 10.1142/S2424905X23400019.

[12] L. Cheng, Z. G. Hou, M. Tan. Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model. *Automatica*, vol. 45, no. 10, pp. 2312–2318, 2009. DOI: 10.1016/j.automatica.2009.06.007.

[13] H. Yin, A. Varava, D. Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, vol. 6, Article number 54, 2021. DOI: 10.1126/scirobotics.abd8803.

[14] X. Lin, Y. Wang, J. Olkin, D. Held. SoftGym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Proceedings of the 4th Conference on Robot Learning*, Cambridge, USA, pp. 432–448, 2020.

[15] F. Gu, Y. Zhou, Z. Wang, S. Jiang, B. He. A survey on robotic manipulation of deformable objects: Recent advances, open challenges and new frontiers, [Online], Available: https://arxiv.org/abs/2312.10419, 2023.

[16] A. Billard, D. Kragic. Trends and challenges in robot manipulation. *Science*, vol. 364, Article number 6446, 2019. DOI: 10.1126/science.aat8414.

[17] P. Jiménez. Survey on model-based manipulation planning of deformable objects. *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 154–163, 2012. DOI: 10.1016/j.rcim.2011.08.002.

[18] C. Dai, Y. Sun. Robotic orientation control of linear-shaped sperm. *Robotic Manipulation of Reproductive Cells*, C. Dai, Y. Sun, Eds., Cham, Germany: Springer, pp. 53–70, 2023. DOI: 10.1007/978-3-031-52730-2_5.

[19] H. Yin, M. C. Welle, D. Kragic. Policy learning with embedded Koopman optimal control. *Proceedings of Machine Learning Research*, vol. 144, pp. 1–14, 2021.

[20] J. Barbic, J. Popović. Real-time control of physically based simulations using gentle forces. *ACM Transactions on Graphics*, vol. 27, Article number 5, 2008. DOI: 10.1145/1409060.1409116.

[21] A. De Marco. Investigation of A Model-based Approach for Dynamical Manipulation of Deformable Objects, Ph. D. dissertation, Polytechnic University of Turin, Italy, 2023.

[22] N. Sunil. Deformable Object Manipulation with A Tactile Reactive Gripper, Master dissertation, Massachusetts Institute of Technology, USA, 2021.

[23] M. Yu, K. Lv, C. Wang, Y. Jiang, M. Tomizuka, X. Li. Generalizable whole-body global manipulation of deformable linear objects by dual-arm robot in 3-D constrained environments. *The International Journal of Robotics Research*, vol. 44, no. 4, pp. 607–639, 2025. DOI: 10.1177/02783649241276886.

[24] M. Yu, K. Lv, H. Zhong, S. Song, X. Li. Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach. *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 417–436, 2023. DOI: 10.1109/TRO.2022.3200546.

[25] N. Dehio, Y. Wang, A. Kheddar. Dual-arm box grabbing with impact-aware MPC utilizing soft deformable end-effector pads. *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5647–5654, 2022. DOI: 10.1109/LRA.2022.3158433.

[26] F. Ding, J. Huang, Y. Wang, T. Fukuda, T. Matsuno. Adaptive sliding mode control for manipulating deformable linear object with input saturation. In *Proceedings of IEEE International Conference on Mechatronics and Automation*, Chengdu, China, pp. 1862–1867, 2012. DOI: 10.1109/ICMA.2012.6285105.

[27] A. Delgado, C. A. Jara, F. Torres, C. M. Mateo. Control of robot fingers with adaptable tactile servoing to manipulate deformable objects. In *Proceedings of Advances in Robotics: Second Iberian Robotics Conference*, Cham, Germany, pp. 81–92. 2016. DOI: 10.1007/978-3-319-27146-0_7.

[28] P. Zhou, P. Zheng, J. Qi, C. Li, H. Y. Lee, A. Duan, L. Lu, Z. Li, L. Hu, D. Navarro-Alarcon. Reactive human–robot collaborative manipulation of deformable linear objects using a new topological latent control model. *Robotics and Computer-integrated Manufacturing*, vol. 88, Article number 102727, 2024. DOI: 10.1016/j.rcim.2024.102727.

[29] F. Ding, J. Huang, Y. Wang, T. Matsuno, T. Fukuda. Vibration damping in manipulation of deformable linear objects using sliding mode control. *Advanced Robotics*, vol. 28, no. 3, pp. 157–172, 2014. DOI: 10.1080/01691864.2013.861769.

[30] D. Shi, H. Hu, C. Yang, Z. Lu, Q. Li. A learning system for deformable object cooperative manipulation. *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 8453–8464, 2025. DOI: 10.1109/TASE.2024.3486063.

[31] Y. Zhang, F. Liu, X. Liang, M. Yip. Achieving autonomous cloth manipulation with optimal control via differentiable physics-aware regularization and safety constraints. In *Proceedings of IEEE International Conference on Robotics and Automation*, Yokohama, Japan, pp. 9931–9938, 2024. DOI: 10.1109/ICRA57147.2024.10611111.

[32] H. Jeon, M. H. Choi. Interactive motion control of deformable objects using localized optimal control. In *Proceedings of IEEE International Conference on Robotics and Automation*, Rome, Italy, pp. 2582–2587, 2007. DOI: 10.1109/ROBOT.2007.363854.

[33] J. Barbič, M. da Silva, J. Popović. Deformable object animation using reduced optimal control. In *Proceedings of ACM SIGGRAPH Papers*, New Orleans, USA, Article number 53, 2009. DOI: 10.1145/1576246.1531359.

[34] W. He, T. Meng, D. Huang, X. Li. Adaptive boundary iterative learning control for an Euler–Bernoulli beam system with input constraint. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1539–1549, 2018. DOI: 10.1109/TNNLS.2017.2673865.

[35] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu. Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018. DOI: 10.1109/TIE.2018.2803773.

[36] W. He, T. Meng, X. He, C. Sun. Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances. *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1524–1535, 2019. DOI: 10.1109/TCYB.2018.2808321.

[37] D. Tong, A. Choi, L. Qin, W. Huang, J. Joo, M. K. Jawed. Sim2Real neural controllers for physics-based robotic deployment of deformable linear objects. *The International Journal of Robotics Research*, vol. 43, no. 6, pp. 791–810, 2024. DOI: 10.1177/02783649231214553.

[38] X. Lin, C. Qi, Y. Zhang, Z. Huang, K. Fragkiadaki, Y. Li, C. Gan, D. Held. Planning with spatial-temporal abstrac-

tion from point clouds for deformable object manipulation. In *Proceedings of the 6th Conference on Robot Learning*, Auckland, New Zealand, pp. 1640–1651, 2022.

[39] S. Chen, Y. Liu, S. W. Yao, J. Li, T. Fan, J. Pan. DiffSRL: Learning dynamical state representation for deformable object manipulation with differentiable simulation. *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9533–9540, 2022. DOI: 10.1109/LRA.2022.3192209.

[40] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, S. Song. Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, London, UK, pp. 5872–5879, 2023. DOI: 10.1109/ICRA48891.2023.10161546.

[41] G. Salhotra, I. C. A. Liu, M. Dominguez-Kuhne, G. S. Sukhatme. Learning deformable object manipulation from expert demonstrations. *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8775–8782, 2022. DOI: 10.1109/LRA.2022.3187843.

[42] C. Yang, Y. Jiang, Z. Li, W. He, C. Y. Su. Neural control of bimanual robots with guaranteed global stability and motion precision. *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1162–1171, 2017. DOI: 10.1109/TII.2016.2612646.

[43] C. Yang, Z. Li, R. Cui, B. Xu. Neural network-based motion control of an underactuated wheeled inverted pendulum model. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004–2016, 2014. DOI: 10.1109/TNNLS.2014.2302475.

[44] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, S. Song. Iterative residual policy: For goal-conditioned dynamic manipulation of deformable objects. *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 389–404, 2024. DOI: 10.1177/02783649231201201.

[45] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, F. Mathis-Ullrich. Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery. *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 560–567, 2023. DOI: 10.1109/LRA.2022.3227873.

[46] J. Matas, S. James, A. J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Proceedings of the 2nd Annual Conference on Robot Learning*, Zürich, Switzerland, pp. 734–743, 2018.

[47] Y. Wu, W. Yan, T. Kurutach, L. Pinto, P. Abbeel. Learning to manipulate deformable objects without demonstrations. In *Proceedings of the 16th Robotics: Science and Systems*, Corvalis, USA, 2020.

[48] Q. Tan, Z. Pan, L. Gao, D. Manocha. Realtime simulation of thin-shell deformable materials using CNN-based mesh embedding. *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2325–2332, 2020. DOI: 10.1109/LRA.2020.2970624.

[49] C. Li, Z. Ai, T. Wu, X. Li, W. Ding, H. Xu. DeformNet: Latent space modeling and dynamics prediction for deformable object manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, Yokohama, Japan, pp. 14770–14776, 2024. DOI: 10.1109/ICRA57147.2024.10611243.

[50] Y. Zhang, L. Cheng, R. Cao, H. Li, C. Yang. A neural network based framework for variable impedance skills learning from demonstrations. *Robotics and Autonomous Systems*, vol. 160, Article number 104312, 2023. DOI: 10.1016/j.robot.2022.104312.

[51] R. Cao, L. Cheng, W. Xue. Iterative assist-as-needed control with skill learning for physical human-robot interaction. *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 7052–7057, 2023. DOI: 10.1016/j.ifacol.2023.10.557.

[52] R. Cao, L. Cheng, C. G. Yang, Z. Dong. Iterative assist-as-needed control with interaction factor for rehabilitation robots. *Science China Technological Sciences*, vol. 64, no. 4, pp. 836–846, 2021. DOI: 10.1007/s11431-020-1671-6.

[53] Y. Zhang, S. Li. A neural controller for image-based visual servoing of manipulators with physical constraints. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5419–5429, 2018. DOI: 10.1109/TNNLS.2018.2802650.

[54] J. A. Preiss, D. Millard, T. Yao, G. S. Sukhatme. Tracking fast trajectories with a deformable object using a learned model. In *Proceedings of International Conference on Robotics and Automation*, Philadelphia, USA, pp. 1351–1357, 2022. DOI: 10.1109/ICRA46639.2022.9812189.

[55] Y. Zhang, S. Li, X. Zhou. Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit. *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3573–3582, 2019. DOI: 10.1109/TIE.2018.2851960.

[56] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, C. Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos, [Online], Available: https://arxiv.org/abs/2410.11831, 2024.

[57] E. Todorov, T. Erez, Y. Tassa. MuJoCo: A physics engine for model-based control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 5026–5033, 2012. DOI: 10.1109/IROS.2012.6386109.

[58] I. Loshchilov, F. Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.

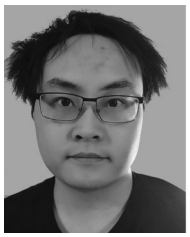
[59] Y. Tang, X. Chu, J. Huang, K. W. S. Au. Learning-based MPC with safety filter for constrained deformable linear object manipulation. *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2877–2884, 2024. DOI: 10.1109/LRA.2024.3362643.

[60] P. J. M. van Laarhoven, E. H. L. Aarts. Simulated annealing. *Simulated Annealing: Theory and Applications*, P. J. M. Laarhoven, E. H. L. Aarts, Eds., Dordrecht, The Netherlands: Springer, pp. 7–15, 1987. DOI: 10.1007/978-94-015-7744-1_2.

[61] J. C. Wu, J. N. Yang, W. E. Schmitendorf. Reduced-order H_∞ and LQR control for wind-excited tall buildings. *Engineering Structures*, vol. 20, no. 3, pp. 222–236, 1998. DOI: 10.1016/S0141-0296(97)00081-3.

[62] S. A. Davari, D. A. Khaburi, F. Wang, R. M. Kennel. Using full order and reduced order observers for robust sensorless predictive torque control of induction motors. *IEEE Transactions on Power Electronics*, vol. 27, no. 7, pp. 3424–3433, 2012. DOI: 10.1109/TPEL.2011.2179812.

[63] B. Bandyopadhyay, M. Patil. Reduced order SMC design for output tracking of arbitrary signal. *Output Tracking in Non-minimum Phase Systems: A Reduced Order Sliding Mode Design Approach*, B. Bandyopadhyay, M. Patil, Eds., Cham, Germany: Springer, pp. 75–99, 2024. DOI: 10.1007/978-3-031-70988-3_4.



Zuyan Chen received the B. Eng. degree in communication engineering from Jiangxi University of Science and Technology, China in 2020, and the M. Res. degree in electronic and electrical engineering from Swansea University, UK in 2024. He is currently a Ph. D. degree candidate in computer science and engineering with the University of Oulu, Finland.

His research interests include robot force control and machine learning.

E-mail: zuyan.chen@oulu.fi

ORCID iD: 0000-0001-7134-6453

Jian-An Huang received the Ph. D. degree in reproducible surface-enhanced Raman spectroscopy (SERS) based on plasmonic nanopillars from City University of Hong Kong, China in 2018. He then worked as a research associate at the University of Hong Kong, China, to build a scanning near-field optical microscope and a Raman spectrometer to study relaxation of micro/nanostructured indium gallium nitride light-emitting diode (In/GaN LED) as well as low-dimensional plasmonic/semiconductor hybrid photodetectors, and then joined Italian Institute of Technology (iit) at Genova, Italy in 2021 as a postdoc for plasmonic biosensor development. He is an assistant professor in University of Oulu, his research combines nanoparticle manipulation and microfluidic SERS for single-cell, single-particle and single-molecule biosensing.

His research interests include plasmonics, nanophotonics, optofluidics and biophysics.

E-mail: jianan.huang@oulu.fi

Juha Röning received the M. Sc. degree (Hons.) in engineering, and the Ph. D. and Licentiate degrees (Hons.) in technology from the University of Oulu, Finland in 1983 and 1992, respectively. He is the Head of the Biomimetics and Intelligent Systems Group, Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland. He has authored or co-authored over 300 papers. He is a Fellow of SPIE and the International Society for Computers and Their Applications.

His research interests include data mining, intelligent systems, mobile robots, and software security.

E-mail: juha.roning@oulu.fi

Leopoldo Angrisani received the M. Sc. degree in electronic engineering from University of Salerno, Italy in 1993, and the Ph. D. degree in electrical engineering from University of Napoli Federico II, Italy in 1997. He is a full professor of electrical and electronic measurements with the Department of Information Technology and Electrical Engineering, University of Napoli Federico II, Italy, where he is also the Chair of the Board of the Ph. D. Program, Information and Communication Technology for Health (ICTH). From 2016 to 2023, he was the Head of CeSMA, Center of Advanced Metrological and Technological Services, University of Napoli Federico II, Italy.

His research interests include the internet of things and cyber-physical measurement systems, green soft-growing sensors, measurement sustainability, measurement uncertainty, measurements for Industry 4.0, communication systems, and networks test and measurement.

E-mail: leopoldo.angrisani@unina.it

Shuai Li received the B. Eng. degree in precision mechanical engineering from the Hefei University of Technology, China in 2005, the M. Eng. degree in automatic control engineering from the University of Science and Technology of China, China in 2008, and the Ph. D. degree in electrical and computer engineering from the Stevens Institute of Technology, USA in 2014. He is currently a full professor with the Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland, and also with the VTT-Technical National Research Center of Finland, Finland.

His research interests include dynamic neural networks, robotics, machine learning, and autonomous systems.

E-mail: shuai.li@oulu.fi (Corresponding author)

ORCID iD: 0009-0004-4499-0023