

Manipulation of Deformable Linear Objects Using Model Predictive Path Integral Control with Bidirectional Long Short-Term Memory Learning

Lukas Zeh^a, Johannes Meiwaldt, Zexu Zhou^b, Armin Lechler^c and Alexander Verl^d

Institute for Control Engineering of Machine Tools and Manufacturing Units, University of Stuttgart, Stuttgart, Germany

Keywords: DLO, MPPI, biLSTM, Manipulation, Control, Robotics.

Abstract: The manipulation of Deformable Linear Objects (DLOs) such as cables poses a significant challenge for automation due to their infinite degrees of freedom and non-linear dynamics. In this paper we present a machine learning based optimal control approach for the manipulation of DLOs. This approach is divided into two main components: modeling and control. For modeling the dynamics of the DLO, we propose a learning based approach using a bidirectional Long Short-Term Memory (biLSTM) network. The biLSTM network is trained on synthetic data generated by the MuJoCo physics engine. For manipulating the DLO, a model predictive control strategy that employs Model Predictive Path Integral (MPPI) control is selected. The proposed approach is evaluated through simulation and experiments. The results demonstrate the effectiveness of the proposed method in achieving accurate and efficient manipulation of DLOs.

1 INTRODUCTION

Flexible objects such as textiles, cables or ropes (Matsuno et al., 2006) can be found almost everywhere, both in everyday life and in the production environment. They belong to the class of deformable objects (Keipour et al., 2022). A sub-category of deformable objects are Deformable Linear Objects (DLOs). Examples of DLOs include cables, ropes and hoses. In the context of robotic applications, rigid bodies are typically assumed when gripping and manipulating objects. This assumption is valid as long as the deformation of the objects is negligible. However, when handling DLOs, the deformation of the object must be taken into account. The automated handling of flexible objects by robots is a research problem that has not yet been entirely solved (Zhu et al., 2022; Zhou et al., 2020).

The fundamental challenge in the manipulation of flexible objects, such as DLOs, is that an external force causes both a movement and a change in shape. Due to the infinite degrees of freedom of DLOs, modeling these nonlinearities during deformation is complex. Especially for real-time robotic manipulation

tasks, accurate and computationally efficient dynamic models are required. While both physics-based and data-driven approaches exist, each has its own advantages and disadvantages (Arriola-Rios et al., 2020).

To enable effective manipulation of DLOs, Model Predictive Control (MPC) has been successfully employed for planning and control in dynamic environments involving DLOs (Yan et al., 2020; Wang et al., 2022). MPC uses a predictive model to simulate and optimize control actions over a finite time horizon, making it suitable for systems with complex, time-varying dynamics. In the context of DLOs, where deformation must be anticipated and accounted for during manipulation, MPC can utilise a learned or physics-based model to generate feasible, optimized trajectories.

This publication investigates the potential of Model Predictive Path Integral (MPPI) control, a sampling based variant of MPC, for the manipulation of DLOs. Simulation data is generated to train the bidirectional Long Short-Term Memory biLSTM network to learn a model of the DLO dynamics offline. The model is then used in an MPPI controller to determine the trajectories for manipulating the DLO.

The contribution of our work can be summarized as follows:

- We contribute datasets, model architecture and model weights for modeling cables. The datasets, model architecture and model weights are avail-

^a  <https://orcid.org/0000-0003-2730-1383>

^b  <https://orcid.org/0009-0002-2163-2528>

^c  <https://orcid.org/0000-0002-4073-1487>

^d  <https://orcid.org/0000-0002-2548-6620>

able at <https://doi.org/10.18419/DARUS-5050>.

- We propose a framework for the manipulation of DLOs that utilizes a Model Predictive Path Integral Controller to manipulate deformable objects.
- We demonstrate the effectiveness of our proposed method in simulation and experiments.

The paper is organised as follows: In Section II, we review related work. The dataset is introduced in Section III, and our proposed framework is established in Section IV. In Section V, we present our simulation and experimental results.

2 RELATED WORK

The precise manipulation of Deformable Linear Objects (DLOs) requires a physics-based model that accounts for both deformation and an appropriate representation of object shape (Sanchez et al., 2018). Modeling approaches can generally be divided into physics-based and data-driven methods (Arriola-Rios et al., 2020).

2.1 Physics-Based Modeling Approaches

There are various physics-based modeling approaches for DLOs. Particle-based models describe DLOs as discrete particles whose positions change in accordance with Newton's laws under the influence of forces. In mass-spring-damper systems, these particles are connected by springs, and their physical parameters are described using parameters such as stiffness and damping (Schulman et al., 2013). Although these models are computationally efficient, they require precise parameterization, which limits their applicability to real-world industrial cables (Monguzzi et al., 2025).

Point-based dynamics (PBD), on the other hand, use geometric constraints to directly compute particle positions. They are more memory- and compute-efficient than mass-spring systems but less physically accurate (Arriola-Rios et al., 2020).

To achieve a more physically accurate representation, the DLO is discretized using Finite Element Methods (FEM) and the deformation equations are solved through numerical integration. However, FEM approaches are computationally intensive and require accurate material parameters (Koessler et al., 2021; Yin et al., 2021). As a result, they are generally unsuitable for real-time robotic manipulation tasks unless specific simplifications are made.

Other numerical methods make assumptions, such as the absence of large deformations, which limits their applicability in more dynamic tasks (Rabaetjé, 2003). Meanwhile, Jacobian-based approaches use local approximations to relate the movement of the robot to the deformation of the object. While these approaches are real-time capable, they only compute local deformation models (Zhu et al., 2022).

2.2 Data-Driven Approaches

Data-driven models have gained popularity due to their ability to capture the complex nonlinear dynamics of DLOs. These models are trained using either simulated (offline) data or real-world (online) data. When using simulated data, physical-based models are typically employed to generate the training data. The advantage of using simulated data is the ease and speed of data generation compared to collecting real-world data.

Several deep learning approaches have been proposed. For example, bidirectional Long Short-Term Memory (biLSTM) networks have been used to propagate DLO dynamics over time (Yan et al., 2020; Yang et al., 2022). The interaction-biLSTM proposed by Yang et al. outperformed a baseline biLSTM model in terms of accuracy, although with slightly reduced computational efficiency.

Graph Neural Networks (GNNs) have also been adopted to model DLO dynamics (Wang et al., 2022; Cao et al., 2024). In GNN-based methods, the DLO is represented as a graph of discrete capsule elements connected by physically motivated constraints such as bending stiffness, length restrictions, and collisions. The nodes represent DLO elements, and the edges capture the interactions between them.

Another approach uses radial basis function networks to estimate local deformation models via Jacobian matrices, encoding the relationship between DLO deformation and the robot end-effector position (Yu et al., 2023).

While these methods are capable of modeling the complex dynamics of DLOs, they typically require large datasets to achieve robust performance. It is therefore essential to assess whether models trained on simulation data generalize well enough for real-world robotic manipulation.

2.3 Model Predictive Control for DLO Manipulation

Model Predictive Control (MPC) is a strategy used for manipulating DLOs (Wang et al., 2022). It relies on predictive models to simulate object dynamics and

optimize control actions over a time horizon. The predictive model can be either physics-based or learned from data. MPC is particularly effective for manipulating deformable objects as it enables forward-looking planning that takes into account the evolution of the object’s shape.

A sampling-based variant of MPC, Model Predictive Path Integral (MPPI) control, was first introduced in (Williams et al., 2016) for the autonomous driving of a high-speed RC car. In (Williams et al., 2017), the authors generalized and formalized the MPPI approach, proposing a learning-based, information-theoretically grounded formulation. This extension makes MPPI applicable in data-driven and model-uncertain scenarios. Since then, MPPI has been applied in various robotic domains (Yan et al., 2020; Bhardwaj et al., 2021; Pezzato et al., 2025). The STORM framework, introduced in (Bhardwaj et al., 2021) is a fast, sampling based model predictive control framework that works directly in joint space. It enables real-time responses to complex manipulation tasks, including collisions, joint boundaries and uncertain perception, through GPU parallelization. (Pezzato et al., 2025) use a GPU-based physics simulator as the dynamic model for MPPI control. This allows high-contact tasks to be solved without explicit modeling or learning, offering a fast, flexible and robust solution in the presence of uncertainties. (Yan et al., 2020) used MPPI control to show the effectiveness of their Coarse-to-fine rope state estimation method. In their work the MPPI controller is used to estimate the optimal gripping point of the rope for manipulating the rope into a desired shape.

We extend existing works by applying MPPI control to manipulate different types of DLOs. We chose a biLSTM model for modeling the DLO dynamics based on the combination of high inference speed, acceptable accuracy and suitability for robust, flexible control. To ensure optimal performance, we conduct a hyperparameter search to identify the best model configuration. The biLSTM model is used in combination with a MPPI controller to generate optimal trajectories for DLO manipulation. The effectiveness of our approach is demonstrated through simulation and experimental results.

3 DATASET

In the following, we describe the dataset used to train the biLSTM model.

3.1 Simulation Environment

The datasets used to train the biLSTM model were generated using the MuJoCo (Todorov et al., 2012) physics engine. MuJoCo natively provides a plugin for the simulation of DLOs. In this plugin, DLOs are approximated as mass-spring systems. The DLO is modeled as a chain of mass points, which are connected by linear, torsional, and bending springs. The individual spring-mass elements are modeled in simplified manner as capsules with the corresponding physical properties. This approach saves time in the modeling process and also allows for a simple and intuitive implementation. Furthermore, it has the advantage that the parameters of the DLO can be easily and quickly adjusted, enabling a wide range of DLO variations to be simulated. To train the biLSTM model, a DLO with a length of 0.5 m was modeled in MuJoCo, consisting of 50 capsules with a diameter of 1 cm. The number of 50 capsules was chosen as a compromise between realistic behavior and computation time. The higher the number of capsules, the more degrees of freedom the system has. This increase in degrees of freedom leads to an almost exponential increase in computation time required to simulate the DLO. The parameters to be set in the simulation are the Young’s modulus [Pa], the shear modulus [Pa], and the damping [Nms/rad] between the individual capsules. Young’s modulus was chosen as 4×10^6 Pa, the Shear modulus as 1×10^6 Pa, and the damping was set to 1 Nms/rad. For the training of the biLSTM model, a simulation step time of 1 ms was chosen. The individual trajectories within the datasets have a length of 5 s. Influences of gravity, friction, and air drag are not considered in the simulation. Figure 1 shows the data generation process. For the simulation, the DLO is fixed at the right end with a welding condition, so that this end behaves like a clamped end. The left end of the DLO is manipulated by a robot arm. The robot arm performs a random trajectory in the xy-plane at a height of 0.15 m. For the data generation, the cable is manipulated from a straight line into a random shape by moving the left end of the DLO to a random position within the green box. The target position of the robot is chosen randomly for each trajectory within the range of $x \in [0.05, 0.35]$ m and $y \in [-0.2, 0.2]$ m (green area in Figure 1). The origin coordinate system is located in the center of the manipulated capsule. To generate a wider range of deformations, the DLO is also randomly rotated around the z-axis in the range of $\psi \in [-1, 1]$ rad. The target position range was chosen to avoid overstretching the DLO. During the data generation, the positions of the 50 capsules

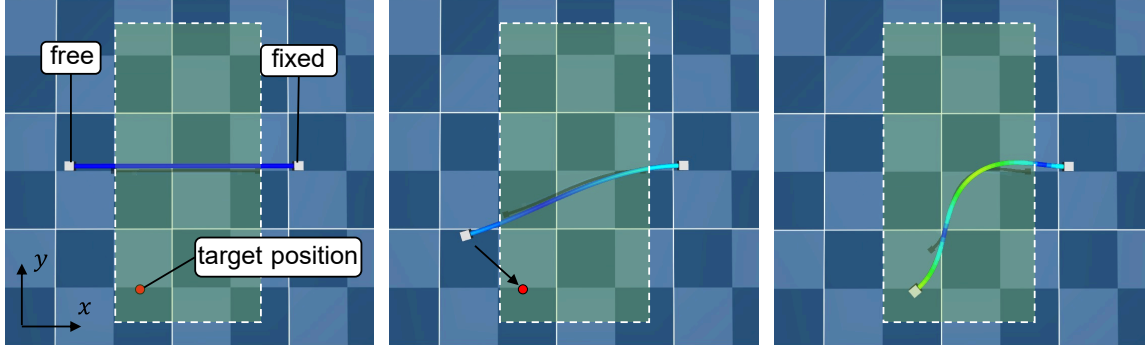


Figure 1: Dataset Generation. The cable is manipulated from a straight line to a random shape by moving the free end of the cable to a random position within the green box.

$\mathbf{X}_{DLO}$, the end effector position $\mathbf{X}_{TCP}$, and velocity $\mathbf{v}_{TCP}$ are recorded.

3.2 Representation of DLO in 2D

Since the manipulation takes place on a surface, we chose a representation of the DLO in 2D, similar to (Yan et al., 2020). For the training of the biLSTM model, the simulation data is reduced. Instead of using all 50 simulated capsules, in order to reduce the computing time, only every 5th capsule is used for training, resulting in a total of $n = 10$ capsules. The first and last capsules are also removed, as these are not needed for predicting the DLO dynamics. The position of the first capsule is described by the pose of the end effector $\mathbf{X}_{TCP}$. The position of the last capsule remains constant due to the welding condition. The position of the DLO can therefore be described as a sequence of points in 3D Cartesian space $\mathbf{X}_{DLO} \in \mathbb{R}^{n \times 2}$. For better generalization of the biLSTM model, the relative position of the capsules with respect to the end effector position $\mathbf{x}_{TCP}$ is used instead of the absolute position, computed as $\mathbf{x}_{r,i} = \mathbf{x}_i - \mathbf{x}_{TCP}$ for $i = 1, \dots, n$. For the calculation of the relative positions, only the x and y coordinates are used, as the z-coordinate is constant due to the fixed height of the end effector. The relative positions of the individual capsules of the DLO are described by

$$\mathbf{X}_{DLO} = (\mathbf{x}_{r,1}, \mathbf{x}_{r,2}, \dots, \mathbf{x}_{r,n}). \quad (1)$$

The advantage of this representation is the translational invariance, which allows the neural network to learn the deformation of the DLO not from the absolute positions, but by directly linking the deformation to the end effector position. The velocity of the capsules is described by the difference of the relative positions at time t and $t - 1$. The overall state of the DLOs is described by

$$\mathbf{S}_{DLO} = (\mathbf{X}_{DLO}, \dot{\mathbf{X}}_{DLO}). \quad (2)$$

The state of the end effector is described by the Cartesian position of the end effector, as well as the rotation of the end effector around the z-axis. The state of the end effector is therefore described in detail as follows:

$$\mathbf{S}_{TCP} = (\mathbf{X}_{TCP}, \dot{\mathbf{X}}_{TCP}) = ((x, y, z, \psi), (\dot{x}, \dot{y}, \dot{z}, \dot{\psi})). \quad (3)$$

The overall state of the system

$$\mathbf{S}_g = (\mathbf{S}_{DLO}, \mathbf{S}_{TCP}), \quad (4)$$

is obtained by combining the state of the DLO and the state of the end effector.

4 PROPOSED FRAMEWORK

In this section, we introduce the proposed framework for cable manipulation. The manipulation of the cable is done in 2D. First, an overview of the system used for the manipulation task is given. Then, the bidirectional Long-Short-Term-Memory (biLSTM) model for modeling the DLO dynamics is introduced. Finally, the Model Predictive Path Integral (MPPI) controller used to manipulate the DLO to the desired shape is described.

4.1 System Overview

The framework for cable manipulation, as displayed in Figure 2, consists of a biLSTM model and a MPPI controller. As an input for the system, the current state of the DLO $\mathbf{S}_{g,t}$ and the desired shape of the DLO $\mathbf{S}_{tar}$ are used. The current state of the DLO, $\mathbf{S}_{DLO}$, is defined by the relative positions and velocities of the capsules, as described in the previous section. The desired shape of the DLO is represented by the target position of the capsules. Based on the current state of the DLO and the desired shape, the MPPI controller generates a set of random trajectories $\mathbf{U}_i$. These trajectories are then sent to the biLSTM model, which

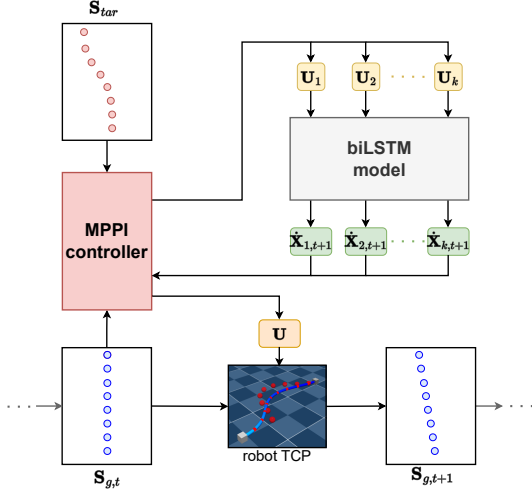


Figure 2: The proposed framework for cable manipulation uses a biLSTM model trained on synthetic data to predict DLO deformation. This prediction is passed to the MPPI controller, which computes the optimal robot control input.

predicts the velocities of the capsules for the next time step $\hat{\mathbf{x}}_i$. The information about the resulting deformation of the DLO is then used to calculate the cost function for the MPPI controller. The best trajectory is selected based on the cost function and sent to the robot for execution. This execution leads to a new state of the DLO $\mathbf{S}_{g,t+1}$, which is then used as input for the next iteration of the MPPI controller. The process is repeated until the DLO has reached the desired shape.

4.2 biLSTM Cable Model

As in the works of (Yang et al., 2022), (Yan et al., 2020), and (Gu et al., 2025), a biLSTM model is employed for modeling the DLO, as it has been shown to effectively capture its dynamic behavior. The biLSTM model is a type of recurrent neural network (RNN) that is particularly well-suited to sequence prediction tasks. The biLSTM is able to capture the dynamics and temporal dependencies of the DLO by processing the sequence of relative positions of the capsules and their velocities. Unlike standard RNNs, information flows in both temporal directions, allowing the model to use both past and future context for improved sequence understanding. This feature allows for more effective modeling of relationships along the DLO structure. This bidirectional processing enhances the LSTM’s ability to capture long-range interactions, improving its performance in sequential deformation modeling tasks. Compared to unidirectional networks such as MLPs, standard RNNs, or unidirectional LSTMs, biLSTMs have ad-

vantages in modeling the complex dynamics of deformable linear objects (Yang et al., 2022). To better capture the coupled dynamics, the biLSTM additionally incorporates the end-effector state as input, enabling the model to learn the interaction between actuator motion and DLO deformation. Thus, the complete system state $\mathbf{S}_g$ is provided as input to the biLSTM model. The general structure of the biLSTM model is shown in Figure 3. Its architecture consists of an input layer, one or more stacked biLSTM layers, and a fully connected output layer. This output layer predicts the capsule velocities for the next time step. The predicted velocities are then used to compute the cost function for the MPPI controller.

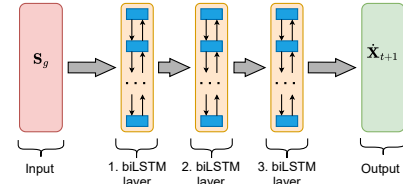


Figure 3: The biLSTM architecture used for modeling the DLO consists of an input layer, three biLSTM layers and an output layer.

4.2.1 Training

The biLSTM model is trained on the data generated in the simulation environment. The biLSTM model is trained using 10,000 trajectories. These trajectories are split into training and test data, with 80 % of the data used for training and 20 % for testing. The training is performed using the Adam optimizer and the Mean Squared Error (MSE) loss function. The MSE loss function is defined as:

$$\text{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (5)$$

where N is the number of nodes, y_i is the true velocity of the node i and $\hat{y}_i$ is the predicted velocity of the node i . To determine the optimal hyperparameters for the biLSTM model, both a random search and a grid search were performed. The random search was used to perform an initial narrowing down of the hyperparameters. The random search showed that the most effective models consistently used a hidden layer size of 256 or 512, were trained for up to 100 epochs, and employed a learning rate between $1e-5$ and $1e-3$. A low weight decay between $1e-7$ and $1e-5$ was also common among top-performing configurations. The number of biLSTM layers varied between 2 and 6, indicating that model depth was less critical compared to other parameters. Training the model for more than 50 epochs didn’t yield significant improvements,

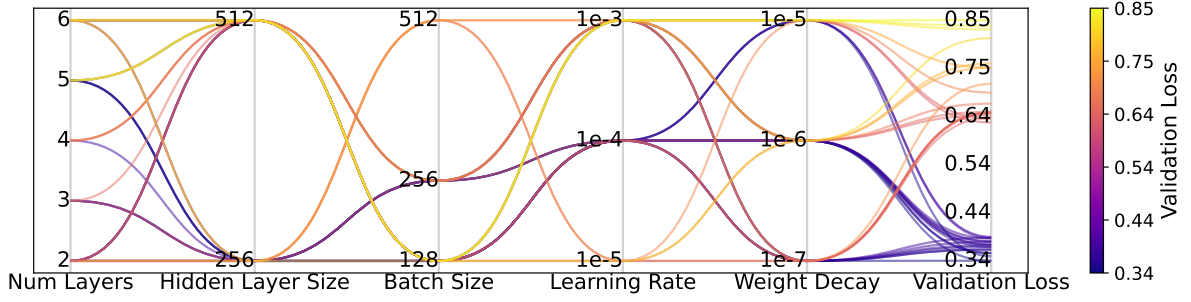


Figure 4: Performance of the biLSTM model in terms of validation loss for different hyperparameter combinations. Visualized are the top 10 % and the bottom 10 % of hyperparameter combinations during grid search.

Table 1: Hyperparameters of the biLSTM model, bold values are also used for grid search.

Hyperparameter	Values Random Search
biLSTM Layers	[1, 2 , 3, 4, 5, 6]
Hidden Layer Size	[8, 16, 32, 64, 128, 256 , 512 , 1024]
Epochs	[10, 20, 30, 40, 50 , 100]
Batch Size	[16, 32, 64, 128 , 256 , 512]
Learning Rate	[1e-3 , 1e-4 , 1e-5 , 1e-6]
Weight Decay	[1e-4, 1e-5 , 1e-6 , 1e-7]

suggesting that the model converged well within this range. In contrast, poor performance was associated with smaller hidden layer sizes, overly small learning rates, high weight decay values, and very small batch sizes. These results suggest that model capacity, sufficient training duration, and a well-tuned optimization setup are essential for achieving high prediction accuracy. Based on these findings, a subsequent grid search was then used to find the optimal hyperparameters in a smaller range. In the table 1, the hyperparameters of the biLSTM model are summarized. Figure 4 shows the model performance of various hyperparameter combinations, obtained through grid search, in terms of the validation loss.

Based on the hyperparameter study, the model with the best performance in terms of validation loss was selected. The model was trained using a learning rate of 1e-4 in combination with a weight decay of 1e-7. The model with the best performance was trained with batchsize 128 and consists of three biLSTM layers, as shown in Figure 3. Each biLSTM layer consists of 512 neurons (this is equivalent to a hidden layer size of 256 neurons in each direction). In the following section, the performance of this model is evaluated.

4.2.2 Model Evaluation

In order to use the biLSTM model in simulation or Model Predictive Control (MPC), a precise rollout prediction over multiple time steps is crucial. A roll-

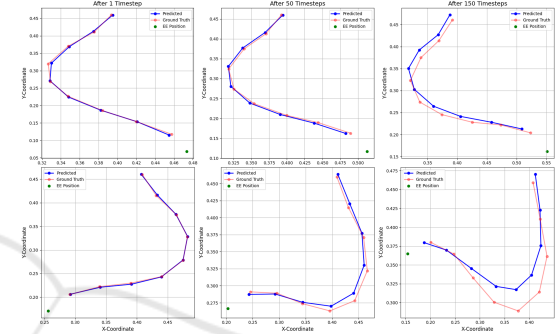


Figure 5: Shape-error $e_{\text{shape,biLSTM}}$ after 1, 50 and 150 timesteps. The blue line represents the model prediction, while the transparent red line represents the ground truth.

out is a sequence of predicted states over a certain time horizon, which is used to evaluate the model's performance in predicting the DLO dynamics. The quality of the dynamic model significantly influences the selection of optimal control sequences. The model quality is evaluated based on rollouts over 50 time steps (equivalent to 1 second) and 150 time steps (equivalent to 3 seconds). As a metric for the model quality, the average shape error e_{shape} and the average velocity error e_{vel} are used.

$$e_{\text{shape, biLSTM}} = \|\mathbf{x}_{\text{groundtruth}} - \mathbf{x}_{\text{pred}}\|_2, \quad (6)$$

$$e_{\text{vel, biLSTM}} = \frac{\|\dot{\mathbf{x}}_{\text{groundtruth}} - \dot{\mathbf{x}}_{\text{pred}}\|_2}{\|\dot{\mathbf{x}}_{\text{groundtruth}}\|_2} \times 100 \%. \quad (7)$$

The model is evaluated on 100 rollouts, each with a length of 150 time steps (3 seconds). The average shape error $e_{\text{shape, biLSTM}}$ and the average velocity error $e_{\text{vel, biLSTM}}$ are calculated over all rollouts. The model is able to predict the shape of the DLO with an average shape error $e_{\text{shape, biLSTM}}$ of 3.3 cm and the velocity with an average velocity error $e_{\text{shape, biLSTM}}$ of 61.59 % (similar to those in (Yang et al., 2022)). The error of both the shape and the velocity increases with the number of time steps. Figure 5 shows qualitative results of the shape error over 1, 50 and 150 time steps. The blue line represents the model prediction,

while the transparent red line represents the ground truth. The biLSTM model shows stable predictions, even over long time intervals. Structure, length and curvature are preserved, indicating a high model capacity and robust dynamic capture.

4.3 Model Predictive Path Integral for Manipulation

Model Predictive Control (MPC) is a well-established control strategy that has been successfully used to manipulate deformable objects (Wang et al., 2022; Yu et al., 2022). In this paper, the DLO is manipulated using a Model Predictive Path Integral (MPPI) based control strategy, similar to that in (Yan et al., 2020) and (Williams et al., 2016). MPPI is a sampling-based Model Predictive Control strategy particularly suited to handling complex systems and multiple objectives.

In this context, MPPI is employed to compute a control strategy that transforms an initial configuration into a desired target configuration through shape control. Actions are represented as target positions for the end effector (EE).

The MPPI algorithm is based on the principle of sampling multiple control sequences around a nominal sequence. A new control sequence is then generated as a weighted average of these control sequences. This new sequence is then used to construct the nominal control sequence for the next iteration.

A special feature of MPPI lies in the evaluation of the simulated trajectories. Each trajectory is assigned a cost value indicating how well the system performs under the respective control inputs. After simulating numerous future trajectories, each trajectory is assigned a unique set of disturbance values. The MPPI algorithm calculates a weighted sum of these disturbances. This nominal control sequence is initialized using one of two alternative mechanisms. When no prior solution exists, a zero-valued sequence spanning the planning horizon is used as the starting point. However, when a previous solution is available, a receding horizon approach is employed. In this approach, the prior solution is propagated forward by one timestep and the terminal control action is reset to zero. This warm-start methodology maintains solution continuity while adhering to the principles of Model Predictive Control.

Each trajectory is evaluated based on its cost value, with lower-cost trajectories receiving higher weights and thus have a greater influence on the control update. Specifically, the weight of a trajectory is determined by the exponential function of the negative ratio of its cost to a fixed parameter λ , also known

as temperature (Williams et al., 2017):

$$w_k = e^{-\frac{s_k}{\lambda}}, \quad (8)$$

where s_k represents the cost of trajectory k . The cost function used in MPPI consists of two terms: a shape cost and a control cost. The shape cost penalizes deviations from the target configuration and is defined using the Euclidean seminorm:

$$C_S = 0.5 \cdot \sum_{i=1}^N \left(e_{shape,i}^\top \cdot Q \cdot e_{shape,i} \right), \quad (9)$$

where $Q = \text{diag}(w_1, w_2, \dots, w_i)$ is a diagonal matrix assigning positive weights w_i to each feature point in the plane. The control cost penalizes excessive input effort and is given by:

$$C_R = 0.5 \cdot \sum_{i=1}^N \left(u_{b,i}^\top \cdot R \cdot u_{b,i} \right), \quad (10)$$

where R is the weighting matrix for the sampled control inputs u_b .

To compute a valid probability distribution over trajectories, the raw weights are normalized:

$$\tilde{w}_k = \frac{w_k}{\sum_{j=1}^K w_j}. \quad (11)$$

This normalization ensures that trajectories with lower costs contribute more strongly, while keeping the overall influence balanced.

The MPPI (Model Predictive Path Integral) algorithm proceeds as described in Algorithm 1. It begins with the initialization of a nominal control sequence $\mathbf{U} = \{u_0, u_1, \dots, u_{N-1}\}$, which is typically initialized to zero. In each iteration, a set of K trajectories is generated by sampling random disturbances δu_k for every time step across the prediction horizon. These disturbances are added to the nominal control sequence to create perturbed control sequences. These are then used in a Monte Carlo tree search. For generating the random disturbances δu_k , pink noise (Eberhard et al., 2023) is used. Each control sequence simulates the system's response to the disturbed input sequence. For deformable linear object (DLO) manipulation, this simulation is performed using the biLSTM model, which predicts the resulting DLO states $\hat{\mathbf{X}}_i$ based on the current system state $\mathbf{S}_{g,i}$ and the sampled control input. The resulting trajectory is evaluated using a cost function that measures the deviation from the target state $\mathbf{S}_{tar}$ as well as the control effort. The total cost of each trajectory s_k is computed, and the corresponding weight $\tilde{w}_k$ is derived as described above. The nominal control sequence is then updated using a cost-weighted average of the disturbances:

$$\mathbf{U} \leftarrow \mathbf{U} + \sum_{k=1}^K \tilde{w}_k \cdot \delta u_k. \quad (12)$$

This update shifts the nominal inputs towards those associated with lower-cost trajectories, thereby gradually improving control performance. The updated control input is applied to the system, which advances by one time step. The resulting new system state is recorded and used as input to the biLSTM model for the next iteration. The first element u_0 of the optimized control vector $\mathbf{U}$, produced by the MPPI controller, is applied to the robot or EE. The resulting system state is then updated. This process is repeated until a termination condition is met, either after a fixed number of iterations or when the distance error threshold is reached.

Data: Initial state $\mathbf{S}(g, t_0)$, model dynamics, cost function, prediction horizon N , number of control sequences K

Result: Optimized control input sequence $\mathbf{U}_{0..N-1}$

```

initialize control sequence  $\mathbf{U}_{0..N-1}$ ;
while target not reached do
    generate random disturbances  $\delta\mathbf{U}$ ;
    for control sequences  $k = 1..K$  do
        start at current state  $\mathbf{X}_{k,0} = \mathbf{X}(t_0)$ ;
        for horizon steps  $n = 0..N-1$  do
            input  $\mathbf{U}_{k,n} = \mathbf{U}_n + \delta\mathbf{U}_{k,n}$ ;
            next state
             $\mathbf{X}_{k,n+1} = \text{biLSTM}(\mathbf{X}_{k,n}, \mathbf{U}_{k,n})$ ;
            trajectory cost  $s_k = \text{control cost}$ 
             $C_R + \text{shape cost } C_S$ ;
        end
    end
    end
    for  $n = 0..N-1$  do
         $\mathbf{U}_n += \text{reward-weighted disturbance}$ ;
    end
    apply first input  $\mathbf{U}_0$  as control input;
    receive current state;
    check if target is reached;
end
    
```

Algorithm 1: MPPI Monte-Carlo-Algorithmus.

As shown in Figure 2, this complete process enables model-based manipulation of deformable objects by optimizing a control sequence that minimizes cost while adapting to the predicted system dynamics.

5 SIMULATION AND EXPERIMENTS

In this section, the simulation and experimental results are presented and discussed. The goal of the simulation and experiments is to analyze the behavior of



Figure 6: Experiment Setup.

the DLO and the performance of the MPPI controller. The simulation and experiments are performed with a Franka Emika Panda robot. First, the simulation and experimental setup is described. Then, the results are presented and discussed.

5.1 Simulation and Experimental Setup

1. **Simulation Setup:** The simulation environment is built using the MuJoCo physics engine. As in the simulation, the cable has a length of 50 cm and a diameter of 1 cm. Young's modulus is 4×10^6 Pa, the Shear modulus is 1×10^6 Pa and damping is set to 1 Nms/rad. One Franka Emika Panda robot moves one end of the cable so that the shape of the cable matches the desired shape. The other end of the cable is fixed.
2. **Experiment Setup:** The experimental setup is shown in Figure 6. A Franka Emika Panda robot is used to manipulate the cable so that the shape of the cable matches the desired shape. The other end of the cable is fixed using two zip ties. An Intel Realsense D435i RGB-D camera is used to track the shape of the cable. The biLSTM model and the MPPI controller are implemented on a Ubuntu 24.04 real-time desktop computer. The robot trajectories are sent to the robot for execution with a communication frequency of 1,000 Hz. The camera data is processed with 40 fps.

5.2 Simulation Results

In the following, the results of the simulation are presented. The simulation was performed using the MuJoCo model of the Franka Emika Panda robot provided by the MuJoCo physics engine. The left end of the DLO is firmly gripped by the end effector of the robot. The right end of the DLO is fixed. The initial pose of the robot is set to reflect the initial position

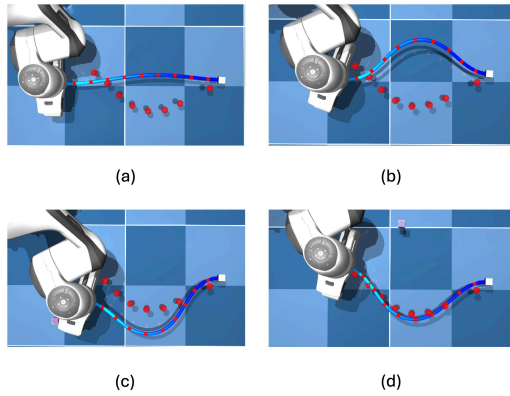


Figure 7: Steps of the shape control in the simulation environment.

of the DLO, meaning the robot is positioned so that the DLO is in a straight line. Each control output of the MPPI controller is set as the target position for a motion capture body. This body serves as the Cartesian target position for the inverse kinematics control of the robot. The control process was performed in the x-y-plane, while the z-position was kept constant at 10 cm. The rotation around the z-axis also received a direct control input from the MPPI output. The control values were limited to a range of -0.2 m to 0.2 m in the x-direction and 0 m to 0.35 m in the y-direction. This reflects the workspace during the generation of the training data. The parameters of the MPPI controller are displayed in Table 2. These parameters were determined empirically through a series of experiments.

First, the performance of the MPPI controller is evaluated by shaping the DLO into a U-shape. Figure 7 shows the process of the shape control in the simulation. In (a), the initial state of the DLO is shown. In (b), the DLO is deformed to the opposite side of the target position. In (c), the MPPI control could compensate for the initial deformation and move the DLO to the other side. In (d), the MPPI control has approached the target position and the trial has ended. 100 trials were performed, with a success criterion being a maximum deviation of 2 cm between the positions of the simulated capsules (represented by red dots on the cable) and their corresponding target points (represented by red dots on the plane). The success rate was 93 %, with an average time of 7.26 s per successful trial. A successful trial is defined as a trial in which the DLO is shaped into the desired U-shape within 30 seconds.

Additionally, the performance of the MPPI controller is evaluated over 1,000 trials of shaping the DLO into random goal shapes. The success rate was 20.5 %, with an average time of 13.3 s per successful

Table 2: MPPI parameters in simulation.

Parameter	Value
Horizon (H)	5
Time increment (dt)	0,2
Number of samples (N)	20
Temperature (λ)	0.002
Standard deviation of the disturbances (δ) for $[X, Y, Z, \text{Rot}(Z)]$	$[0.5; 0.5; 0; 3]$
Form error weighting (Q)	150
Control error weighting (R)	5

trial.

The simulation results show that the MPPI-based control in combination with a biLSTM dynamics model is able to shape the DLO into a desired target shape. The approach combines sample-based control with a neural network for modeling the DLO dynamics. By training with relative positions of the capsules, the approach can also be transferred to new scenarios. However, re-optimizing the MPPI parameters is necessary in scenarios where the target shape differs significantly from the one for which the controller was originally tuned. Additional parameterization is also required to handle severe deformations of the DLO effectively. One limiting factor is the computational demand of the biLSTM model, especially when processing a large number of fault samples. Increasing computing resources could improve the controller’s performance, as a larger sample size generally leads to greater accuracy and robustness. Another potential bottleneck is the current sampling strategy used by the MPPI controller. Using an adaptive sampling approach where exploration during the construction of the search tree focuses only on trajectories with high solution potential, might improve the results.

5.3 Experiment Results

The performance of the MPPI controller is evaluated across three different scenarios. In the first scenario, 2D shape control is performed on a 50 cm long cable with a diameter of 6 mm, equipped with 9 markers. The second scenario uses the same cable and marker setup, but shape control is conducted in 3D. The third scenario involves 2D shape control of a 50 cm long wire with a diameter of 1.5 mm, without any markers. While initial parameter tuning was performed in simulation, further adjustments were necessary during practical validation to compensate for discrepancies between simulated and real-world behavior. The MPPI controller parameters remain consistent across all scenarios and are listed in Table 3.

In all three scenarios, the cable is manipulated into

Table 3: MPPI parameters in experiment.

Parameter	Value
Horizon (H)	8
Time increment (dt)	0.02
Number of samples (N)	200
Temperature (λ)	0.002
Standard deviation of the disturbances (δ) for $[X, Y, Z, \text{Rot}(Z)]$	$[0.2; 0.2; 0; 0.02]$
Form error weighting (Q)	150
Control error weighting (R)	5

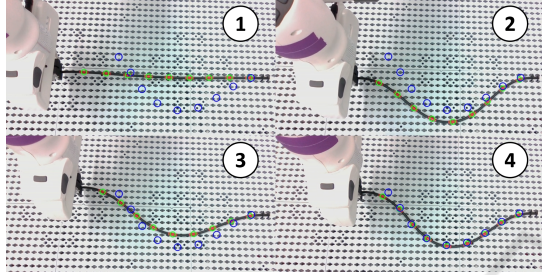


Figure 8: Experiment result of the 2D shape control of a 50 cm long cable with a diameter of 6 mm, marked with 9 markers. The cable is manipulated into a U-shape.

a U-shape. The goal of the shape control experiments is to shape the cable into the desired U-shape with a maximum deviation of 2 cm from the target position. The derivation of the target position is based on the difference between the measured positions of the markers on the cable and the target positions of the markers. In the first and second scenario, 9 markers along the cable are used to track the shape of the cable. The positions of these markers are tracked using a color filter. In the third scenario, no markers are used to track the shape of the cable. Instead, the FastDLO algorithm (Caporali et al., 2022) is used for shape estimation. Based on the estimated shape, 9 virtual markers are placed along the tracked shape in order to keep the process of the shape control as similar as possible to the first two scenarios.

In the first scenario, the cable is mounted directly on the working plane. In Figure 8, the process of a successful 2D shape control is shown. Indicated are the initial state of the cable and robot (1), two intermediate states (2) and (3), and the final state of the cable and robot (4), where the cable has been successfully shaped into the desired U-shape. To evaluate the performance of the MPPI controller, 20 trials were performed. The success rate was 85 %, with an average time of 15.7 s per successful trial. The fastest trial took 2.1 s, and the slowest took 45.2 s. In the second scenario, the cable is mounted on a 7 cm high platform. As in the first scenario, the 9 markers are

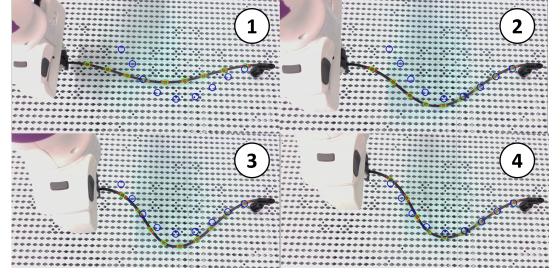


Figure 9: Experiment result of the 3D shape control of a 50 cm long cable with a diameter of 6 mm, marked with 9 markers. The cable is manipulated into a U-shape.

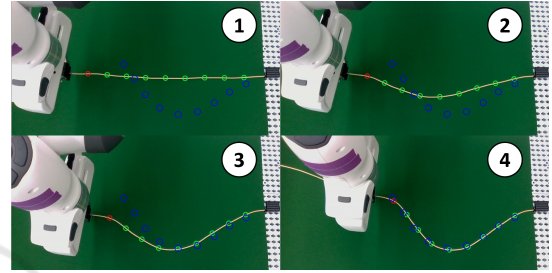


Figure 10: Experiment result of the 2D shape control of a 50 cm long wire with a diameter of 1.5 mm. The cable is manipulated into a U-shape.

used to track the shape of the cable. The positions of these markers are tracked by using a color filter. In Figure 9, the process of a successful 3D shape control is shown. The starting position of the robot remains the same as in the first scenario. This leads to the cable being slightly bent by gravity at the beginning, which can be seen in (1). In (2) and (3), two intermediate states of the shape control are displayed. The final state of the cable and robot can be seen in (4), where the cable has been successfully shaped into the desired U-shape.

Even though this scenario was not trained, the MPPI controller is able to manipulate the cable into a U-shape. The process of the shape control is not as robust as in the first scenario, leading to a higher failure rate. In the practical experiments, failure was defined as a trial in which the DLO is not shaped into the desired U-shape within 90 seconds. Also, the time to reach the target position is significantly higher. To evaluate the controller, 20 trials were performed. The success rate was 50 %, with an average time of 28.3 s per successful trial. The fastest trial took 10.4 s, and the slowest 80.5 s. In the third scenario, we used a thinner and more flexible wire with a diameter of 1.5 mm. This wire was selected to test the performance of the biLSTM model in respect to generalization. Like in the first scenario, the wire is mounted directly on the working plane. In this scenario, no

markers are used to track the shape of the wire, as we want to test the performance in a more real-world-like scenario. Instead, the FastDLO algorithm (Caporali et al., 2022) is used for shape estimation. Since the background of the testbench is white and the wire used is light yellow, we needed to use a greenscreen to be able to track the shape of the wire. In Figure 10, the process of a successful 2D shape control is shown. The starting position of the robot remains the same as in the first scenario. In (1), the initial state of the wire and Robot is shown. In (2) and (3), two intermediate states of the shape control are displayed. In (4), the final state of the wire and robot can be seen, where the wire has been successfully shaped into the desired U-shape.

Like in the second scenario, this scenario is less robust than the first scenario. Also leading to a higher failure rate and a longer time to reach the target position. To evaluate the controller, 20 trials were performed. The success rate was 60 %, with an average time of 25.4 s per successful trial. The fastest trial took 8.2 s and the slowest 65.3 s.

Additionally to the shapes shown in the figures, we also performed trials with different target shapes. The more the target shapes resemble a U-shape, the more the more likely the controller is to successfully shape the DLO into the desired shape. As the simulation results have shown, this control approach struggles especially with shapes requiring significant deformations. Since the cable and wire used in the experiments are pre-bend in one direction, the controller also struggles to deform the DLOs in the direction opposite to the pre-bend. Additionally, the approach is likely to fail if the DLO is deformed in the wrong direction at the beginning of the trial.

Besides these issues and limitations, the results of the experiments show that the MPPI-based control in combination with a biLSTM dynamics model, is able to shape the DLO into a desired target shape. The experiments have also shown that the biLSTM model is able to model different kinds of DLOs with very different properties. As mentioned before, the MPPI controller has to be retuned for scenarios that are not similar to the scenario for which the controller was initially tuned.

6 DISCUSSION AND FUTURE WORKS

In this paper, we presented a novel approach for the manipulation of deformable linear objects (DLOs) using a biLSTM model and a Model Predictive Path Integral (MPPI) controller. The approach combines

a neural network for modeling the DLO dynamics with a sampling-based control strategy. The biLSTM model is trained on a dataset of simulated DLO trajectories, which are generated using a MuJoCo model of the DLO. The model is able to predict the shape and velocity of the DLO over multiple time steps. The MPPI controller is used to manipulate the DLO into a desired target shape. The approach was evaluated in simulation and experiments using a Franka Emika Panda robot. The results show that the MPPI-based control in combination with a biLSTM dynamics model is able to shape the DLO into a desired target shape. The approach is able to generalize to different kinds of DLOs with different physical properties.

In the future, we want to tune the pretrained models with real data in order to minimize the sim to real gap and obtain more robust models. We also want to test different kinds of neural networks inside the Model Predictive Control loop, in order to evaluate if the approach becomes more robust or faster when faster or more accurate models are used. In respect to faster models, we want to test the performance of MLPs. In respect to more accurate models, we want to test the performance of GNNs.

In order to improve the performance of the MPPI controller, we want to investigate different sampling strategies and different cost functions.

ACKNOWLEDGEMENTS

This work was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016.

REFERENCES

- Arriola-Rios, V. E., Guler, P., Ficuciello, F., Kragic, D., Siciliano, B., and Wyatt, J. L. (2020). Modeling of Deformable Objects for Robotic Manipulation: A Tutorial and Review. *Frontiers in Robotics and AI*.
- Bhardwaj, M., Sundaralingam, B., Mousavian, A., Ratliff, N., Fox, D., Ramos, F., and Boots, B. (2021). STORM: An Integrated Framework for Fast Joint-Space Model-Predictive Control for Reactive Manipulation.
- Cao, B., Zang, X., Zhang, X., Chen, Z., Li, S., and Zhao, J. (2024). Shape Control of Elastic Deformable Linear Objects for Robotic Cable Assembly. *Advanced Intelligent Systems*, 6(7).
- Caporali, A., Galassi, K., Zanella, R., and Palli, G. (2022). FASTDLO: Fast Deformable Linear Objects Instance Segmentation. *IEEE Robotics and Automation Letters*, 7(4).

- Eberhard, O., Hollenstein, J., Pinneri, C., and Martius, G. (2023). Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*.
- Gu, F., Sang, H., Zhou, Y., Ma, J., Jiang, R., Wang, Z., and He, B. (2025). Learning Graph Dynamics with Interaction Effects Propagation for Deformable Linear Objects Shape Control. *IEEE Transactions on Automation Science and Engineering*.
- Keipour, A., Bandari, M., and Schaal, S. (2022). Deformable One-Dimensional Object Detection for Routing and Manipulation. *IEEE Robotics and Automation Letters*, 7(2).
- Koessler, A., Filella, N. R., Bouzgarrou, B., Lequievre, L., and Ramon, J.-A. C. (2021). An efficient approach to closed-loop shape control of deformable objects using finite element models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*.
- Matsuno, T., Tamaki, D., Arai, F., and Fukuda, T. (2006). Manipulation of deformable linear objects using knot invariants to classify the object condition based on image sensor information. *IEEE/ASME Transactions on Mechatronics*, 11(4).
- Monguzzi, A., Dotti, T., Fattorelli, L., Zanchettin, A. M., and Rocco, P. (2025). Optimal model-based path planning for the robotic manipulation of deformable linear objects. *Robotics and Computer-Integrated Manufacturing*, 92.
- Pezzato, C., Salmi, C., Trevisan, E., Spahn, M., Alonso-Mora, J., and Hernández Corbato, C. (2025). Sampling-Based Model Predictive Control Leveraging Parallelizable Physics Simulations. *IEEE Robotics and Automation Letters*, 10(3).
- Rabaetje, R. (2003). Real-time Simulation of Deformable Objects for Assembly Simulations. In *Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003 - Volume 18*.
- Sanchez, J., Corrales, J.-A., Bouzgarrou, B.-C., and Mezouar, Y. (2018). Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7).
- Schulman, J., Lee, A., Ho, J., and Abbeel, P. (2013). Tracking deformable objects with point clouds. In *2013 IEEE International Conference on Robotics and Automation*.
- Todorov, E., Erez, T., and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Wang, C., Zhang, Y., Zhang, X., Wu, Z., Zhu, X., Jin, S., Tang, T., and Tomizuka, M. (2022). Offline-Online Learning of Deformation Model for Cable Manipulation with Graph Neural Networks. *IEEE Robotics and Automation Letters*, 7(2).
- Williams, G., Drews, P., Goldfain, B., Rehg, J. M., and Theodorou, E. A. (2016). Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*.
- Williams, G., Wagener, N., Goldfain, B., Drews, P., Rehg, J. M., Boots, B., and Theodorou, E. A. (2017). Information theoretic MPC for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*.
- Yan, M., Zhu, Y., Jin, N., and Bohg, J. (2020). Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects.
- Yang, Y., Stork, J. A., and Stoyanov, T. (2022). Learning differentiable dynamics models for shape control of deformable linear objects. *Robotics and Autonomous Systems*, 158.
- Yin, H., Varava, A., and Kragic, D. (2021). Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54).
- Yu, M., Lv, K., Zhong, H., Song, S., and Li, X. (2023). Global Model Learning for Large Deformation Control of Elastic Deformable Linear Objects: An Efficient and Adaptive Approach. *IEEE Transactions on Robotics*, 39(1).
- Yu, M., Zhong, H., and Li, X. (2022). Shape Control of Deformable Linear Objects with Offline and Online Learning of Local Linear Deformation Models. In *2022 International Conference on Robotics and Automation (ICRA)*.
- Zhou, H., Li, S., Lu, Q., and Qian, J. (2020). A Practical Solution to Deformable Linear Object Manipulation: A Case Study on Cable Harness Connection. In *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE.
- Zhu, J., Cherubini, A., Dune, C., Navarro-Alarcon, D., Alambeigi, F., Berenson, D., Ficuciello, F., Harada, K., Kober, J., Li, X., Pan, J., Yuan, W., and Gienger, M. (2022). Challenges and Outlook in Robotic Manipulation of Deformable Objects. *IEEE Robotics and Automation Magazine*, 29(3).