

Analyzing Local Models for the Robotic Manipulation of Deformable Linear Objects

Project thesis

of

Mohamed Musthafa Palamadathil Kozhisseri

04. April 2025

Supervisor: Georg Rabenstein, M.Sc.
Auditor: Prof. Dr.-Ing. K. Graichen

Chair of Automatic Control
Faculty of Engineering
Friedrich-Alexander-Universität Erlangen-Nürnberg

Declaration

I certify that I have prepared this work without outside assistance and without using sources and tools other than those stated, in particular the usage of AI tools. The work has not been submitted in the same or similar form to any other examining authority and has been accepted by them as part of an examination. All statements that are taken literally or analogously are marked as such.

Erlangen, 04. April 2025

Mohamed Musthafa
Palamadathil Kozhisseri

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objective	2
1.3	Related Works	2
2	Methodology	5
2.1	Local Deformation Model	5
2.2	Path Planning	7
2.2.1	Basic Algorithm	8
2.2.2	Advanced Algorithm	9
3	Implementation	13
3.1	Simulation Setup	13
3.1.1	Task A: Cable Manipulation Without Contacts	14
3.1.2	Task B: Cable Manipulation With a Single Contact	16
3.1.3	Task C: Cable Manipulation With Multiple Contacts	20
4	Conclusion	27
4.1	Summary	27
4.2	Future Scope	28
	Bibliography	29

1 Introduction

This chapter explains the motivation for studying the manipulation of deformable linear objects (DLOs) and the objective of this thesis. It also gives an overview of the existing methods, showing their strengths and limits. This work focuses on using local deformation models for cable shaping tasks in constrained environments.

1.1 Motivation

The change of an object in size and shape under applied forces is called deformation. Robotic manipulation of deformable objects such as cables, clothes, and rubber is a significant yet complex task in automation. Their applications are extensive in industrial automation, medical robotics, and logistics, prioritizing the need for effective control methods [1].

Deformable linear objects (DLOs) such as cables and ropes have infinite degrees of freedom and exhibit nonlinear deformation behaviors, which make automatic handling very challenging. Compared to rigid objects, DLOs do not have a fixed geometry, and their shape continuously varies under manipulation. This raises challenges for modeling, perception, and control, particularly in applications like cable assembly and household robotics [2][1].

Traditional physics-based models such as the Finite Element Method (FEM), Mass-spring systems, and Position Based Dynamics (PBD) offer physically interpretable models. These methods require accurate physical parameters, which are difficult to estimate. Some physics-based methods are computationally expensive, which makes them unsuitable for real-time robotic control [2]. On the other hand, data-driven models provide much more flexibility and adaptability, but require a larger amount of training data. While learning-based techniques have demonstrated success in simulation, they typically lack robustness in real-world applications [3][2].

Another popular approach is a method based solely on online estimation of local linear deformation models, in which a small deformation of the DLO is linear with respect to a small movement of the end-effector. The deformation model is then approximated by a locally estimated Jacobian matrix, which is comparatively simpler to formulate. However, they are usually task-specific and only accurate for local configurations [4]. The potential usage of local linear models in cable manipulation tasks is investigated in this work.

1.2 Objective

This thesis is primarily built upon the existing work in [5], which focuses on an approach using a local deformation model. The proposed method utilizes intermediate configurations, planning algorithms to learn the DLO's behavior and manipulate it without any prior knowledge of its physical properties. The approach is evaluated using the SOFA simulation framework [6].

The existing work [5] successfully deploys a local linear model in simple cable shaping tasks. The key goal of this thesis is to analyze the feasibility, adaptability, and performance of a local linear model in cable shaping under diverse task conditions. This includes evaluating how well the model generalizes when the task involves routing around obstacles, getting entangled, or following multi-step paths to achieve complex cable configurations.

By extending the original framework, this work introduces an additional level of complexity by manipulating the cable in environments with obstacles. Furthermore, the proposed methodology explores adaptive path planning, periodically updating the local deformation model to account for the nonlinear behavior of the DLO over time. This thesis seeks to demonstrate that local linear deformation models, when trained and applied with the right strategy, can be a powerful alternative.

1.3 Related Works

Existing modeling approaches can be categorized as physics-based, data-driven, and hybrid models. There are several approaches to physics-based DLO modeling like the Finite Element Method (FEM) [7] [8], continuous elastic rod models [9] [10], or multi-body models [11] [12]. All these methods demand substantial knowledge of the manipulated object, such as its mesh structure, Young's modulus, shear modulus, or mass. Additionally, they rely on precise perception systems capable of identifying all components of the object's complex state representation [13].

Data-driven models are a flexible and computationally efficient alternative to physics-based models. There are various methods in this domain like using Reinforcement Learning [14] [15] or Neural Networks [16] [17]. The limitations of these methods are low generalization for unseen data, and the need for large datasets [4]. Local linear deformation modeling is an interesting approach from the data-driven domain, which is easier to implement. These models estimate the local Jacobian matrix that linearly maps the local deformations of the cable to the motion of the robot's end-effector. The following provides a brief description of a few of these approaches.

Jin et al. [5] introduce the SPR-RWLS framework, combining structure-preserved registration and robust least squares to estimate the local Jacobian. This method shows robustness

under sensor noise but is limited to simple manipulation tasks. Almaghout et al. [18] propose a planar method, using the diminishing rigidity property to estimate the Jacobian matrix, which improves accuracy. However, it does not generalize the deformations well in 3D. In contrast, Li et al. [19] present a fully online method using recursive least squares that adapts in real-time integrated with a second Jacobian matrix, estimated to approximate the center of the DLO geometry. This method succeeds in 3D manipulation without any prior calibration. Similarly, Zhu et al. [3] propose a vision-based approach using 2D contours and an adaptive receding window to recalculate the Jacobian that compensates for the nonlinear behavior of the DLO.

Several hybrid approaches have been proposed to improve DLO manipulation by combining offline learning with online refinement. Yu et al. [20] use a neural network to estimate the local Jacobian offline, updating it online during manipulation. However, this method requires extensive pretraining. Similarly, Cao et al. [21] integrate offline GNN-based learning with online updates using least squares. In addition, the system is equipped with a model predictive controller, demonstrating real-world applicability but still reliant on accurate pretraining. Despite the variety of approaches, most works focus on simple path planning objectives in unconstrained environments. This work investigates the potential usage of local linear deformation models in more complicated tasks such as cable routing with additional obstacles in the environment.

2 Methodology

In this chapter, the framework for the cable manipulation task is presented. First, it is demonstrated how the cable deformation can be approximated using local linear models. Afterwards, a path planning scheme based on this model is introduced. The framework also introduces an adaptive cable shaping algorithm to meet task requirements with accuracy.

2.1 Local Deformation Model

Data for the gripper movements and the corresponding cable configurations are collected and used for approximating the local deformation model. It is assumed that a cable is held by N_g robotic grippers. A gripper pose

$$\tilde{\mathbf{r}}_i = [x, y, z, \phi, \theta, \psi] \in \mathbb{R}^6 \quad (2.1)$$

is defined by translations and rotations. The configurations of N_g grippers, each with 6 degrees of freedom, at time t are described as

$$\mathbf{r}(t) = [\tilde{\mathbf{r}}_1, \tilde{\mathbf{r}}_2 \dots, \tilde{\mathbf{r}}_{N_g}]^T \in \mathbb{R}^{N_g \times 6} \quad (2.2)$$

A cable configuration is described as a sequence of N_t uniformly distributed points. A typical cable configuration is depicted in Figure 2.1, in which the red dots indicate the tracking points and the green dots indicate the robotic grippers. Each tracking point

$$\tilde{\mathbf{c}}_i = [x, y, z] \in \mathbb{R}^3 \quad (2.3)$$

is represented by its 3D position. The collection of N_t points along the cable representing the cable configuration at time t is described as the following.

$$\mathbf{c}(t) = [\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \dots, \tilde{\mathbf{c}}_{N_t}]^T \in \mathbb{R}^{N_t \times 3} \quad (2.4)$$

A local deformation model uses the difference in cable configuration with respect to gripper movement. The change in gripper movements

$$\delta \mathbf{r}(t) = [\delta \tilde{\mathbf{r}}_1(t) \dots \delta \tilde{\mathbf{r}}_{N_g}(t)]^T \quad (2.5)$$

is described by their change in pose, with $\delta \mathbf{r}(t) = \mathbf{r}(t) - \mathbf{r}(t-1)$. The corresponding deformation of the cable

$$\delta \mathbf{c}(t) = [\delta \tilde{\mathbf{c}}_1(t) \dots \delta \tilde{\mathbf{c}}_{N_t}(t)]^T \quad (2.6)$$

is the positional difference between the tracking points, with $\delta\mathbf{c}(t) = \mathbf{c}(t) - \mathbf{c}(t-1)$. The desired local linear model [5] mapping the relationship between $\delta\mathbf{r}(t)$ and $\delta\mathbf{c}(t)$ is described below.

$$\delta\mathbf{c}(t) = \frac{\delta\mathbf{c}}{\delta\mathbf{r}}(t)\delta\mathbf{r}(t) = \left[\frac{\delta c_1}{\delta r}(t) \dots \frac{\delta c_{N_f}}{\delta r}(t) \right]^T \delta\mathbf{r}(t) \quad (2.7a)$$

$$= \mathbf{A}(t)\delta\mathbf{r}(t) \quad (2.7b)$$

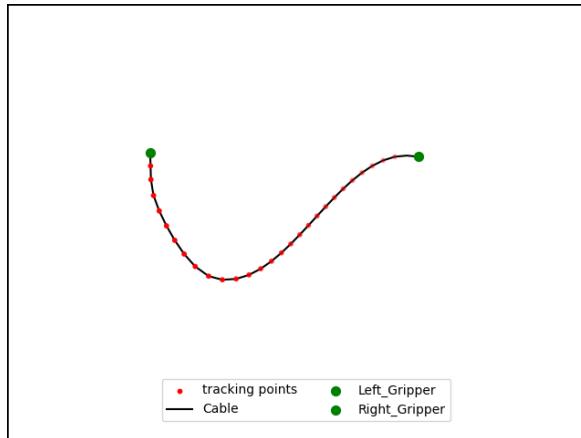


Figure 2.1: A cable represented by a number of points on it held by two robotic grippers.

A DLO has infinite degrees of freedom and exhibits nonlinear deformation behavior. This makes it hard to formulate a time-varying global model. If the motions of the robotic grippers are small enough, the resulting deformations will be small and the local deformation model can be approximated as a linear function. In (2.7b), $\mathbf{A}(t) \in \mathbb{R}^{N_t \cdot 3 \times N_g \cdot 6}$ is a time-varying Jacobian matrix, that represents the relationship between the movement of the robotic grippers and the configuration of the cable. This matrix is estimated based on recorded cable configuration data by randomly moving the grippers m times. These datasets \mathbf{C} and \mathbf{R} are represented as the following.

$$\mathbf{C} = [\mathbf{c}(0) \dots \mathbf{c}(m)] \in \mathbb{R}^{m \times N_t \times 3} \quad (2.8a)$$

$$\mathbf{R} = [\mathbf{r}(0) \dots \mathbf{r}(m)] \in \mathbb{R}^{m \times N_g \times 6} \quad (2.8b)$$

The $m-1$ resulting deformations $\delta\mathbf{C}$ to the corresponding gripper movements $\delta\mathbf{R}$

$$\delta\mathbf{C} = [\delta\mathbf{c}(0) \dots \delta\mathbf{c}(m-1)] \in \mathbb{R}^{m-1 \times N_t \times 3} \quad (2.9a)$$

$$\delta\mathbf{R} = [\delta\mathbf{r}(0) \dots \delta\mathbf{r}(m-1)] \in \mathbb{R}^{m-1 \times N_g \times 6} \quad (2.9b)$$

form the training datasets for the local deformation model. The relationship $\delta\mathbf{C} = \mathbf{A} \cdot \delta\mathbf{R}$, can be described as the deformation behavior. Instead of directly estimating the matrix \mathbf{A} , the pseudo-inverse $\mathbf{G} = \mathbf{A}^\dagger \in \mathbb{R}^{N_g \cdot 6 \times N_t \cdot 3}$ is calculated. The resulting linear formulation is

$\delta\mathbf{R} = \mathbf{G} \cdot \delta\mathbf{C}$. The pseudo-inverse matrix \mathbf{G} can be estimated by solving the optimization problem

$$\mathbf{G} = \arg \min_{\mathbf{G}} \|\mathbf{G} \cdot \delta\mathbf{C} - \delta\mathbf{R}\|^2 \quad (2.10a)$$

$$= \sum_{t=0}^{m-1} \arg \min_{\mathbf{G}} \|\mathbf{G} \cdot \delta\mathbf{c}(t) - \delta\mathbf{r}(t)\|^2 \quad (2.10b)$$

by minimizing the sum of squared means between the predicted and the actual gripper displacements.

A control law for cable shaping can now be derived based on the matrix \mathbf{G} . The difference between the desired cable configuration $\mathbf{c}_D = [\tilde{\mathbf{c}}_{D,1}, \tilde{\mathbf{c}}_{D,2}, \dots, \tilde{\mathbf{c}}_{D,N_t}]^T \in \mathbb{R}^{N_t \times 3}$ and the current cable configuration $\mathbf{c}_C = [\tilde{\mathbf{c}}_{C,1}, \tilde{\mathbf{c}}_{C,2}, \dots, \tilde{\mathbf{c}}_{C,N_t}]^T \in \mathbb{R}^{N_t \times 3}$ is the desired configuration difference $\delta\mathbf{c}_D = \mathbf{c}_D - \mathbf{c}_C$ needed to achieve the goal. The required control law is determined using the estimated matrix \mathbf{G} as the following.

$$\delta\mathbf{r}_D = \mathbf{G} \cdot \delta\mathbf{c}_D \quad (2.11)$$

However, this data-driven model only approximates the deformation behavior locally and is valid only for small displacements. For bigger distances between \mathbf{c}_C and \mathbf{c}_D , the model may generate a control path that exceeds the valid range of the model. This can be rectified by limiting the magnitude of the calculated control law $\delta\mathbf{r}_D$ by setting an upper bound u . The control path $\delta\mathbf{r}_D$ can be interpreted as a vector and the unit vector will point in the direction in which the grippers should move. The unit vector is calculated as $\hat{\delta\mathbf{r}}_D = \frac{\delta\mathbf{r}_D}{\|\delta\mathbf{r}_D\|}$, where $\|\delta\mathbf{r}_D\|$ is the norm of the vector $\delta\mathbf{r}_D$. The control law $\delta\mathbf{r}_D$ is regulated to the upper bound u to ensure accuracy of the local model.

$$\delta\mathbf{r}_D = \hat{\delta\mathbf{r}}_D \cdot u \quad (2.12)$$

During more complex tasks, like cable routing with obstacles, the deformation behavior can change. Therefore, the model has to be updated regularly during the operation and a method to guide the cable to the desired configuration is required.

2.2 Path Planning

This section presents the path planning strategy employed for cable manipulation. Two planning algorithms are introduced: a basic algorithm for simple tasks and an extended scheme for complex scenarios involving intermediate configurations. Figure 2.2 provides an overview of the proposed path planning strategy. After initialization, the program enters a closed loop of data collection, model updating, and control path execution. At each step, the mean squared error is evaluated. If the final configuration is not reached, the loop continues with new data, executing iterative gripper motions.

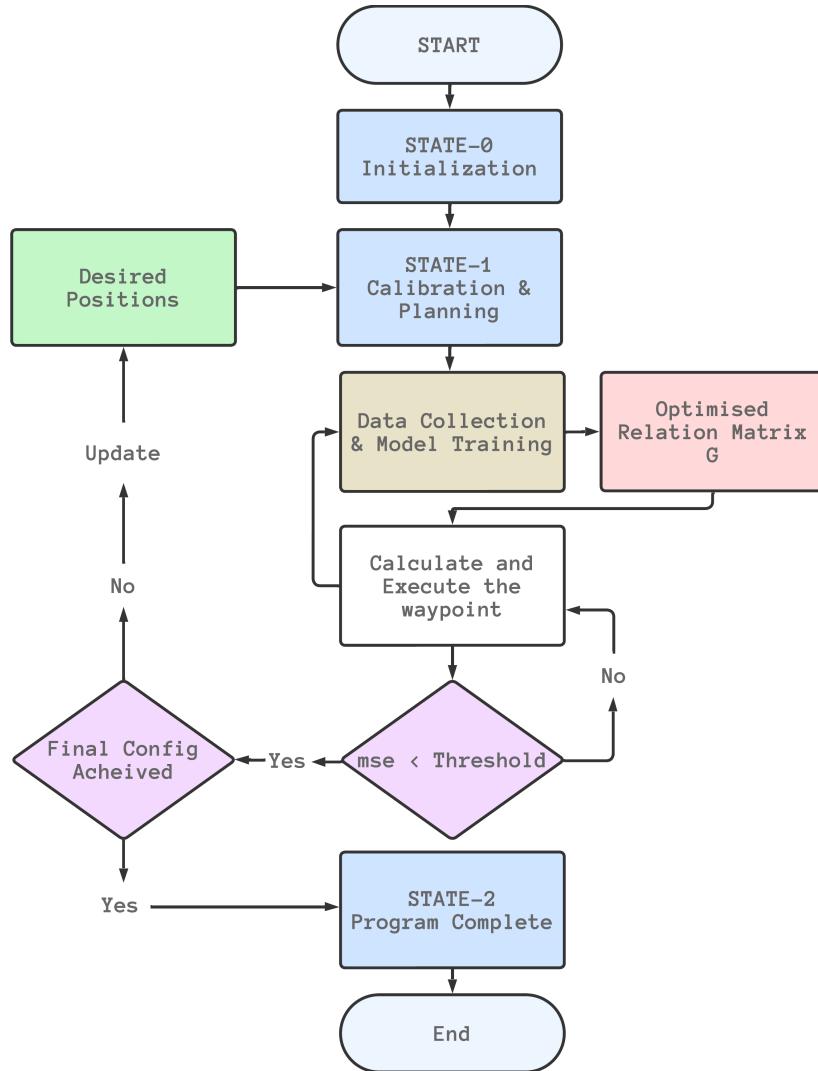


Figure 2.2: State Diagram

2.2.1 Basic Algorithm

The basic path planning approach for cable shaping is presented in Algorithm 1. The algorithm begins by initializing the desired configuration \mathbf{c}_D , which is recorded by manually moving the grippers. Then, an initial gripper movement $\mathbf{r}_{initial}$ is performed and the training datasets (2.9b) and (2.9a) are collected. These datasets form the base for estimating the matrix \mathbf{G} that represents the local linear mapping between the gripper movement and the cable deformation by solving (2.10a).

Based on \mathbf{G} , the algorithm iteratively calculates and executes the required gripper control (2.11) if the mean squared error between the configurations \mathbf{c}_C and \mathbf{c}_D

$$mse(\mathbf{c}_C, \mathbf{c}_D) = \frac{1}{N_t} \sum_{i=0}^{N_t} \|\tilde{\mathbf{c}}_{C,i} - \tilde{\mathbf{c}}_{D,i}\|^2 \quad (2.13)$$

Algorithm 1 Basic Algorithm

```

1: Input the prerecorded  $\mathbf{c}_D$  and initialize  $\epsilon$ 
2: Perform  $\mathbf{r}_{initial}$  and collect  $\delta\mathbf{C}$  and  $\delta\mathbf{R}$ .
3: Compute the relationship matrix  $\mathbf{G}$ 
4: while  $mse(\mathbf{c}_C, \mathbf{c}_D) > \epsilon$  do
5:   Compute  $\delta\mathbf{r}_D \leftarrow \mathbf{G} \cdot \delta\mathbf{c}_D$ .
6:   Execute  $\delta\mathbf{r}_D$  and collect  $\delta\mathbf{c}_A$ .
7:   Append  $\delta\mathbf{r}_D$  and  $\delta\mathbf{c}_A$  to datasets  $\delta\mathbf{C}$  and  $\delta\mathbf{R}$ .
8: end while

```

is above a predefined threshold ϵ . In each iteration, the executed gripper motion $\delta\mathbf{r}_D(t)$ and the corresponding achieved deformation $\delta\mathbf{c}_A(t)$ are recorded and added to the datasets. The iterative process is repeated until the error between the current and the desired configuration falls below a threshold ϵ .

2.2.2 Advanced Algorithm

The proposed control law can only be used for local path planning. It might therefore fall short in cable routing scenarios where collisions with obstacles have to be considered. In such complex cases, a single desired configuration is insufficient to guide the system. To account for this, the scheme shown in algorithm 2, extends the basic approach to more complex tasks with multiple goal configurations.

Algorithm 2 Advanced Algorithm

```

1: Input a set of prerecorded configurations  $\mathbf{C}_D$ ,  $t_{max}$  and  $n_{max}$ .
2: Perform  $\mathbf{r}_{initial}$  and collect  $\delta\mathbf{C}$  and  $\delta\mathbf{R}$ .
3: Compute the relationship matrix  $\mathbf{G}$  and set  $t \leftarrow 0$ .
4: for  $\mathbf{c}_{D,i}$  in  $\mathbf{C}_D$  do
5:   while  $mse(\mathbf{c}_C, \mathbf{c}_{D,i}) > \epsilon$  do
6:     Compute  $\delta\mathbf{r}_D \leftarrow \mathbf{G} \cdot \delta\mathbf{c}_{D,i}$ .
7:     Execute  $\delta\mathbf{r}_{D,i}$  and collect  $\delta\mathbf{c}_{A,i}$ .
8:     Append  $\delta\mathbf{r}_{D,i}$  and  $\delta\mathbf{c}_{A,i}$  to datasets  $\delta\mathbf{C}$  and  $\delta\mathbf{R}$ .
9:      $t \leftarrow t + 1$ 
10:    if  $t > t_{max}$  then
11:      Recalculate matrix  $\mathbf{G}$  using  $\delta\mathbf{C}$  and  $\delta\mathbf{R}$ .
12:      Set  $t \leftarrow 0$ .
13:    end if
14:    if  $n > n_{max}$  then
15:      Discard all data at index  $i < (n - n_{max})$  from the datasets.
16:    end if
17:  end while
18: end for

```

In this approach, instead of a single goal configuration, a sequence of p cable configurations

$$\mathbf{C}_D = [\mathbf{c}_{D,1} \dots \mathbf{c}_{D,p}] \in \mathbb{R}^{p \times N_t \times 3} \quad (2.14)$$

are introduced. The first $p - 1$ configurations are the intermediate configurations, which incrementally guide the cable to reach the desired configuration at index p . Each intermediate configuration is chosen such that it is locally reachable from the previous configuration as in Figure 2.3. The algorithm iterates over each configuration $\mathbf{c}_{D,i}$ in \mathbf{C}_D , which employs a similar approach as algorithm 1. The scheme computes and executes the control law $\delta\mathbf{r}_{D,i}$ using matrix \mathbf{G} until mse falls below the predefined threshold ϵ .

The model is retrained periodically every t_{max} iterations using the updated datasets. This ensures that the local linear model remains accurate. However, if set too high, the model update is delayed, which may lead to inaccurate control paths. Additionally, n_{max} is defined as the maximum number of data points to ensure that the data is recent and relevant. For complex tasks, lower values of n_{max} may cause the model to predict the path inaccurately, due to the limited data variety. Likewise, much larger values may include data which is no longer relevant, degrading overall performance.

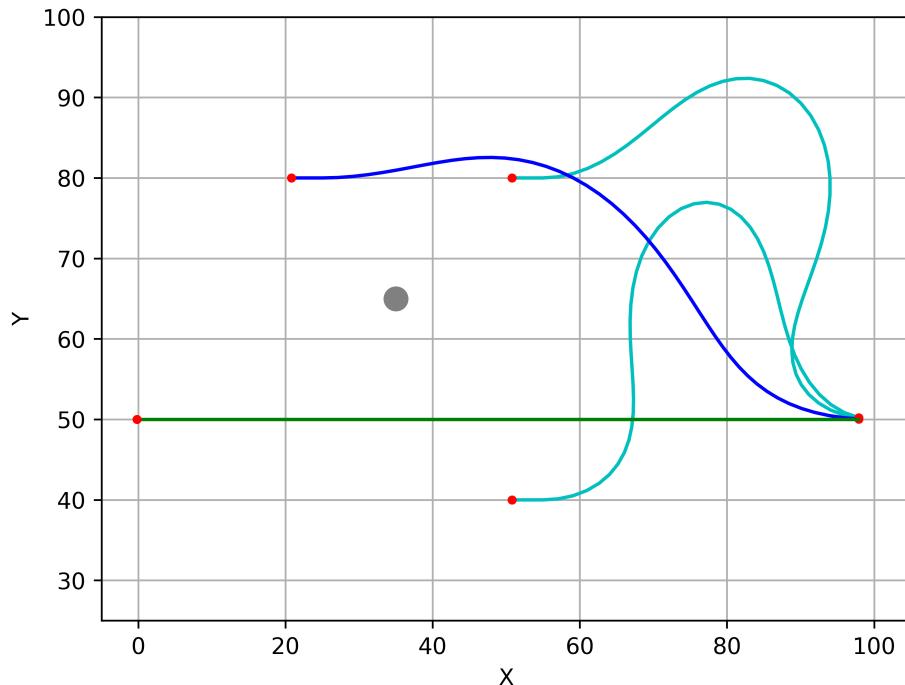


Figure 2.3: Initial configuration(Green) vs Intermediate configurations(Cyan) vs Final Configuration(Blue)

In some cases, multiple initial random motions might be necessary to have enough data variety. This is crucial for the model to generalize local models more accurately for complex configurations. Additionally, achieving all the intermediate configurations precisely is not always necessary. Therefore multiple error thresholds can be introduced, higher error threshold for intermediate configurations and lower for the desired configuration.

To summarize, the algorithm combines local models with adaptive retraining by updating \mathbf{G} periodically and data regulation by limiting the data size to n_{max} . This balance ensures robust performance across complex deformations.

3 Implementation

This chapter presents the implementation of the proposed methodology in a simulated environment. The planning method is implemented in Python and evaluated in the SOFA simulation framework. SOFA enables precise, physics-based simulation of deformable linear objects, providing the flexibility to model, control, and visualize complex cable behaviors under manipulation. This makes it an ideal platform for evaluating the developed framework. Three simulation setups with variations in task and complexity are introduced to evaluate the performance of the proposed algorithm.

3.1 Simulation Setup

The setup is simulated in 3D to closely reflect realistic cable manipulation scenarios. Each end of the cable is held by a robotic gripper. These grippers are denoted as the left gripper $\tilde{\mathbf{r}}_L$, and the right gripper $\tilde{\mathbf{r}}_R$. Throughout the tasks, only $\tilde{\mathbf{r}}_L$ is manipulated, and this simplifies (2.2) to $\mathbf{r}(t) = [\tilde{\mathbf{r}}_L] \in \mathbb{R}^{1 \times 4}$. Likewise, the simulation uses a model with 4 degrees of freedom: 3 translations and rotation around the z-axis, which reduces (2.1) to $\tilde{\mathbf{r}}_L = [x, y, z, \psi] \in \mathbb{R}^4$. The initial configuration of the cable is denoted as $\mathbf{c}_{initial}$. Since the cable length l_{cable} is different for each variation of the experiment, N_t is also different for each setup. The maximum data size n_{max} and the period t_{max} to retrain the model are selected based on the complexity of the task. The initial displacement for the left gripper $\tilde{\mathbf{r}}_{L,initial}$ for data recording is selected to have enough data variety for better generalization of the local Jacobian matrix \mathbf{G} .

To analyze the effect of environmental constraints on cable manipulation, three variations of the simulations are implemented. These variations include manipulation in free space without any environmental contact, routing around a single contact, and navigating through multiple contact points. All three variations execute gripper movements based on algorithm 2 to achieve the goal. The algorithm computes and executes the control path to reach the goal configurations $\mathbf{c}_{D,i}$ in \mathcal{C}_D (2.14) until the error falls below the threshold ϵ . The *mse* (2.13) between the current cable configuration \mathbf{c}_C after each execution and the desired configuration $\mathbf{c}_{D,i}$ is monitored for evaluation. For clarity, the term configuration is abbreviated as ‘Cfg’ in all figures and tables.

3.1.1 Task A: Cable Manipulation Without Contacts

In the first task, the cable is manipulated in an environment with no contacts. The primary objective is to evaluate the model performance in a collision-free environment. The length of the cable l_{cable} is 100 units. It is represented by $N_t = 48$ uniformly distributed tracking points. The cable configuration at time t is given as $\mathbf{c}(t) = [\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \dots, \tilde{\mathbf{c}}_{48}]^T \in \mathbb{R}^{48 \times 3}$. A single desired configuration $\mathbf{c}_{D,1}$ is given as input $\mathbf{C}_D = [\mathbf{c}_{D,1}]^T \in \mathbb{R}^{1 \times 48 \times 3}$. The initial configuration $\mathbf{c}_{initial}$ and the desired configuration \mathbf{C}_D for this task are presented in Figure 3.1.

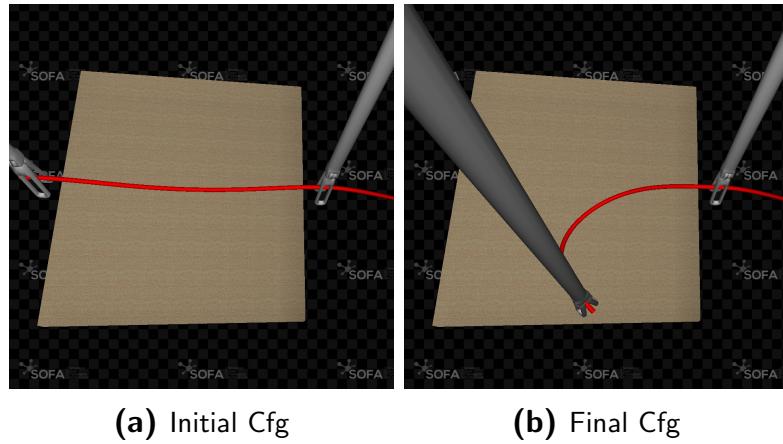


Figure 3.1: Initial Configuration $\mathbf{c}_{initial}$ vs Desired Configuration \mathbf{C}_D

Since the task is simple, the maximum data size is set to $n_{max} = 2000$ and to ensure the local linear model is accurate, the matrix \mathbf{G} is periodically recalculated every $t_{max} = 1000$ steps. The error threshold ϵ is set to 1. The parameters selected for this task are summarized in Table 3.1.

Table 3.1: Task A Setup Parameters

Parameter	Value
Cable Length l_{cable}	100 units
Tracking Points N_t	48
Initial Gripper Pose $\tilde{\mathbf{r}}_L$	[4, 50, 85, 0°]
Initial Displacement $\tilde{\mathbf{r}}_{Linitial}$	[20, 20, 85, 45°]
Maximum Data Size n_{max}	2000
Training Interval t_{max}	1000
Error Threshold ϵ	1
No. of goal configurations p	1

Control Path and Error Analysis

The control path followed by the left gripper is presented in Figure 3.2. The progression of the *mse* values in Table 3.2, shows a converging error trend. The initial error after executing the input path $\tilde{r}_{L,initial}$ is high at 75.62, indicating a significant difference between the configurations $c_{D,1}$ and c_C . A slight increase in error is observed at the first intermediate stage as shown in Figure 3.2b, which could be explained by a lack of data in this early stage.

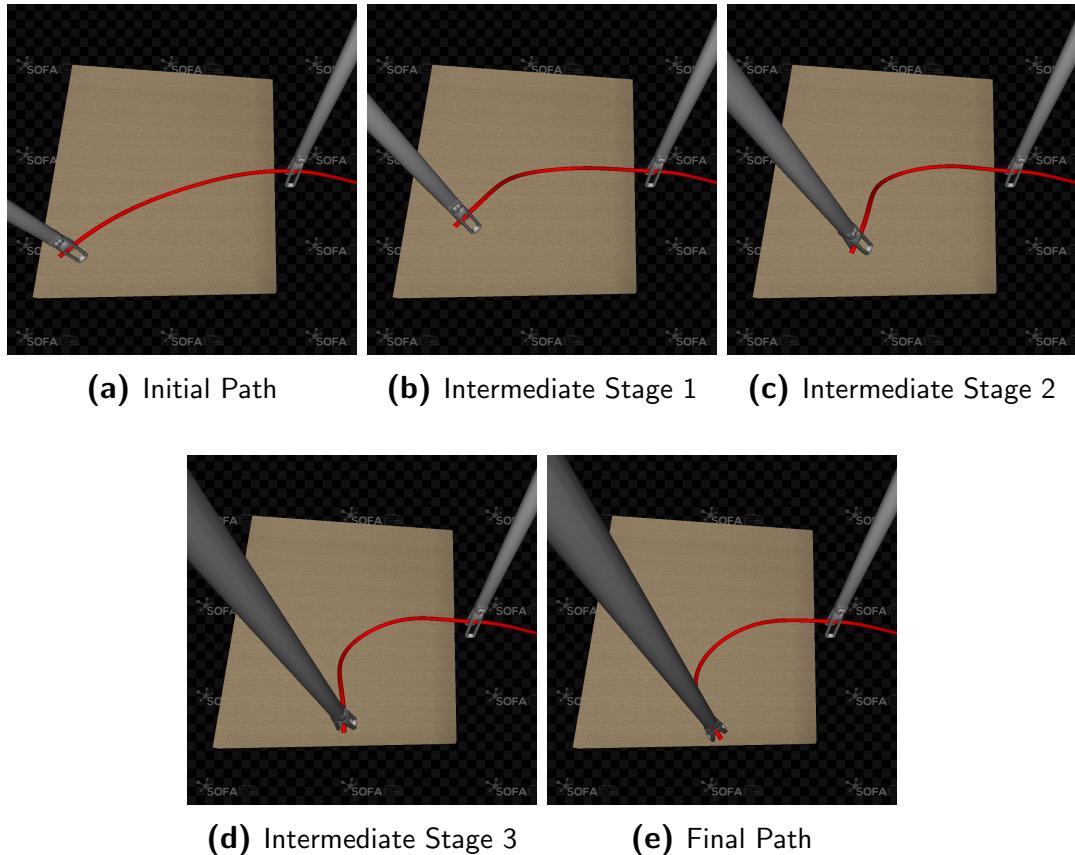
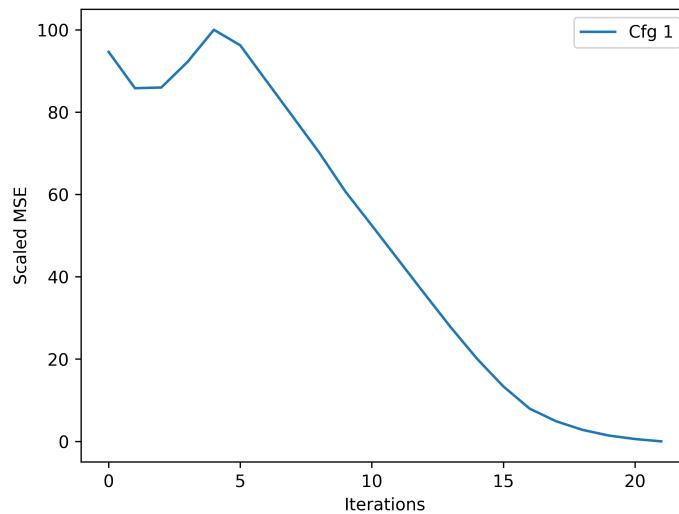
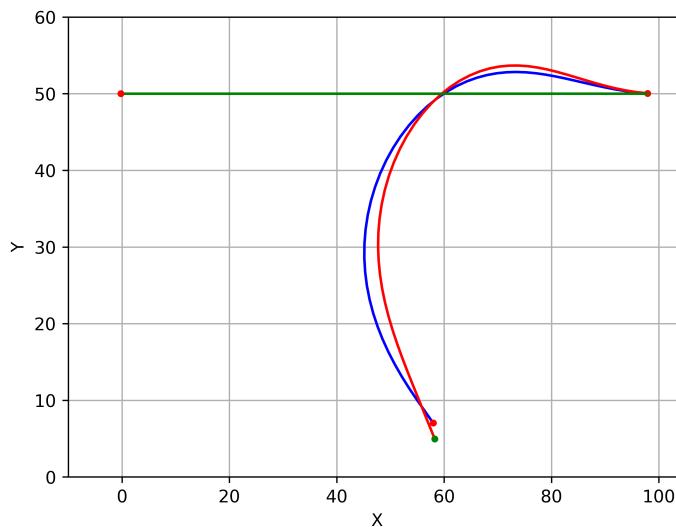


Figure 3.2: Control Path for Task A

As the simulation continues, the model iteratively reduces the *mse* to 49.60 in the second intermediate stage, and further reduces it to 12.80 in the third stage. The reduction demonstrates that the retraining of the local model at regular intervals helps to improve accuracy. Eventually, the *mse* drops significantly to 0.77 in the final position, which is below the predefined threshold $\epsilon = 1$. The *mse* trend for this task is depicted in Figure 3.3, which shows convergence with slight oscillations. A graphical comparison of the final achieved configuration c_A and the desired configuration $c_{D,1}$ is shown in Figure 3.4. This demonstrates the effectiveness of the proposed control strategy in a contact-free environment.

Table 3.2: MSE Values at Different Transition Stages in 3.2

Transition Stage	MSE Value
After Initial Path ($\tilde{r}_{L,initial}$)	75.62
At intermediate stage 1	79.85
At intermediate stage 2	49.60
At intermediate stage 3	12.80
At Final Position	0.77

**Figure 3.3:** MSE Trend**Figure 3.4:** Graphical comparison of c_A (Red), $c_{D,1}$ (Blue) and $c_{initial}$ (Green)

3.1.2 Task B: Cable Manipulation With a Single Contact

In this task, the cable is manipulated in an environment with a single contact. The primary objective is to move the cable around a contact, while collision between cable and

the contact must be avoided. The length of the cable l_{cable} and the number of equally distributed tracking points N_t along the cable are the same as in the first task, and a cable configuration at time t is $\mathbf{c}(t) \in \mathbb{R}^{48 \times 3}$. This task is a bit complex compared to the last task and cannot be completed with a single desired input configuration. Therefore, a set of three desired configurations is given as input $\mathbf{C}_D = [\mathbf{c}_{D,1}, \mathbf{c}_{D,2}, \mathbf{c}_{D,3}]^T \in \mathbb{R}^{3 \times 48 \times 3}$. Figure 3.5 shows the initial configuration, two intermediate configurations and a desired configuration. When the model achieves an intermediate configuration $\mathbf{c}_{D,i}$, the corresponding achieved configuration $\mathbf{c}_{A,i}$ will act as the initial configuration to drive the model to the next configuration $\mathbf{c}_{D,i+1}$.

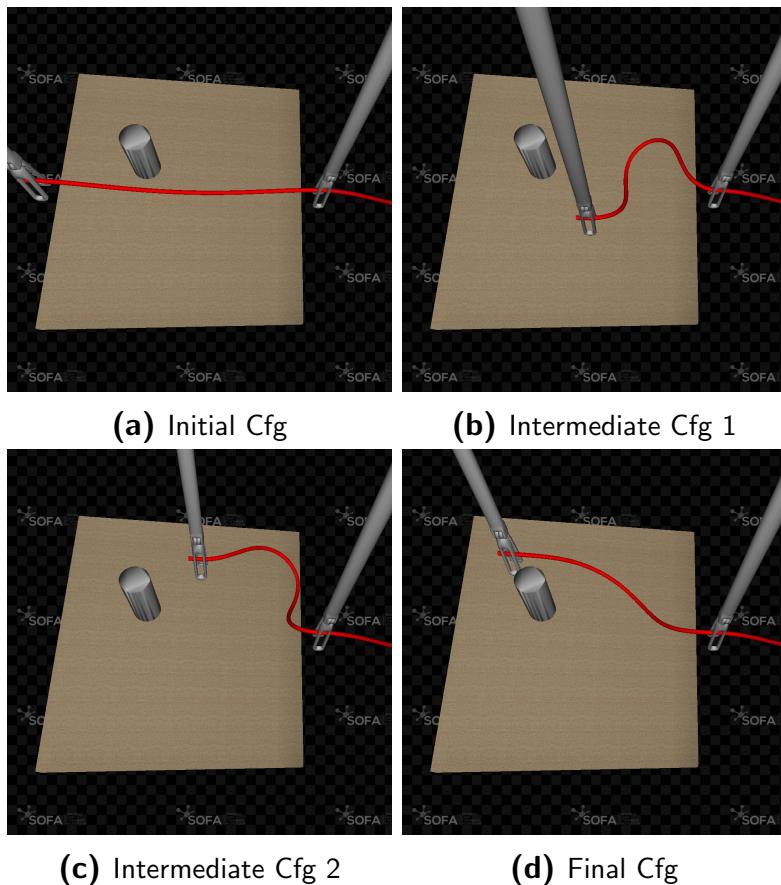


Figure 3.5: Initial Configuration $\mathbf{c}_{initial}$ vs Desired Configurations $\mathbf{c}_{D,i}$

It was observed from the last task how limited data will cause the model to compute and follow an inaccurate control path. Such inaccuracies cannot be tolerated in this task, as they may cause the cable to collide with the contact. To account for this, the maximum data size is set to $n_{max} = 3000$, and the model is periodically retrained every $t_{max} = 1000$ steps. The error threshold ϵ for this task is also set to 1, as the number of configurations to achieve is only three. Table 3.3 summarizes the simulation parameters used in Task B.

Table 3.3: Task B Parameters

Parameter	Value
Cable Length l_{cable}	100 units
Tracking Points N_t	48
Initial Gripper Pose \tilde{r}_L	[4, 50, 85, 0°]
Initial Displacement $\tilde{r}_{L,initial}$	[30, 40, 85, 0°]
Maximum Data Size n_{max}	3000
Training Interval t_{max}	1000
Error Threshold ϵ	1
No. of goal configurations p	3

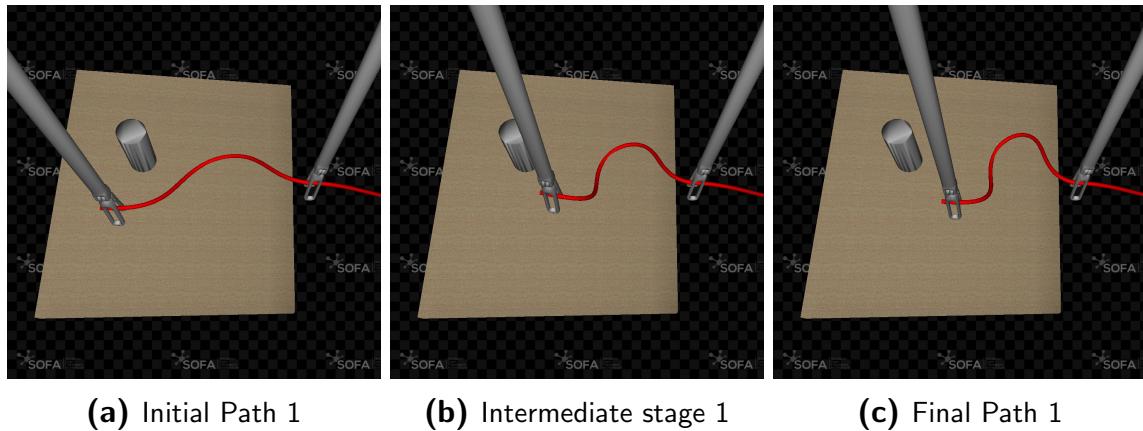
Control Path and MSE Analysis

The control path followed by the gripper for all three input configurations is depicted in Figure 3.9, which shows that the task was successful in avoiding collisions. The *mse* values presented in Table 3.4 highlight the performance of the model for this task. For intermediate configuration 1, the error decreases from 84.71 to 0.93, indicating the first configuration was reached successfully. A similar trend is observed in intermediate configuration 2, where the initial error is significantly higher at 257.68. However, the model successfully reduces this to 0.20, demonstrating its ability to adapt even in the presence of large initial error. The final configuration initially shows a high error of 198.49. It drops to 69.59 and further to 0.88 upon final convergence, following a similar trend.

Figure 3.10 illustrates that the configurations 1 and 2 exhibit smooth and rapid convergence, while configuration 3 shows slower but consistent reduction in *mse*. A graphical representation of the achieved configurations $c_{A,i}$ and the goal configurations $c_{D,i}$ are illustrated in Figure 3.11. These results confirm that the proposed model can robustly handle a task with one contact when sufficient training data is provided and the model is regularly updated.

Table 3.4: MSE Values at Different Transition Stages Experiment 1

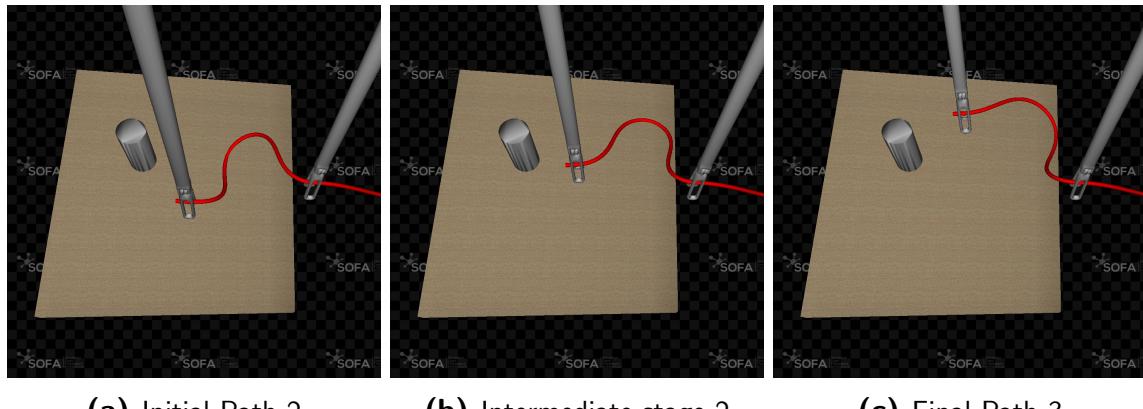
Configuration Transition	Stage	MSE Value
Intermediate Cfg 1	After Initial Path 1	84.71
	At intermediate stage 1	10.84
	At Final Path 2	0.93
Intermediate Cfg 2	After Initial Path 2	257.68
	At intermediate stage 2	56.82
	At Final Path 3	0.20
Final Cfg	After Initial Path 3	198.49
	At intermediate stage 3	69.59
	At Final Position	0.88

Figure 3.6: Initial Cfg to Intermediate Cfg 1

(a) Initial Path 1

(b) Intermediate stage 1

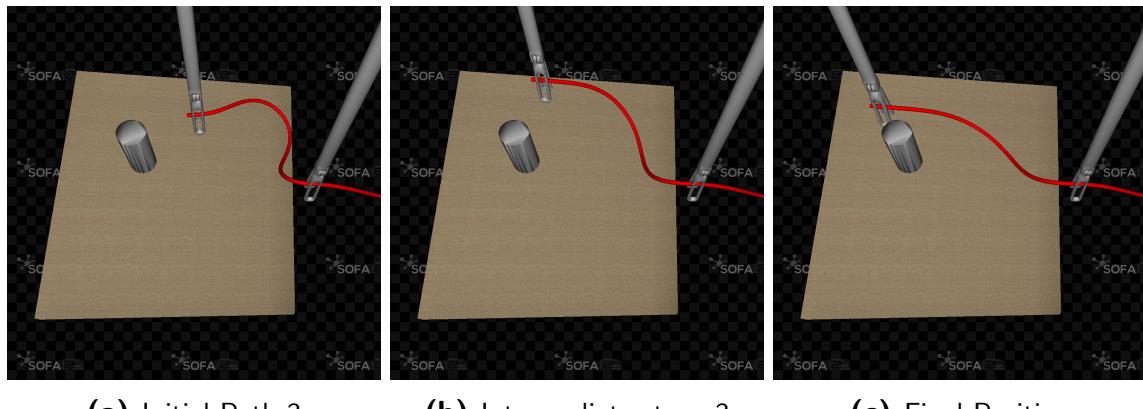
(c) Final Path 1

Figure 3.7: Intermediate Cfg 1 to Intermediate Cfg 2

(a) Initial Path 2

(b) Intermediate stage 2

(c) Final Path 3

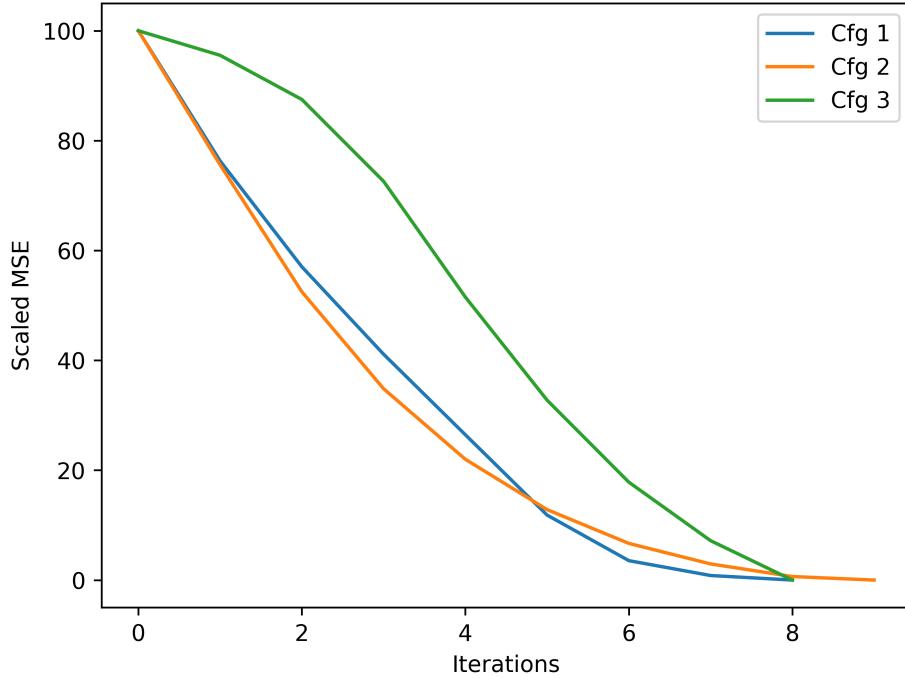
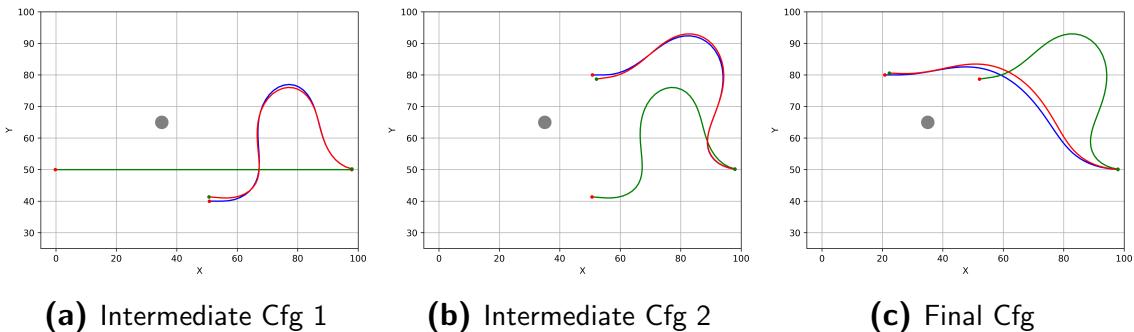
Figure 3.8: Intermediate Cfg 2 to Final Cfg

(a) Initial Path 3

(b) Intermediate stage 3

(c) Final Position

Figure 3.9: Control path progression across all stages: Intermediate Cfg 1, 2, and Final Cfg

**Figure 3.10:** MSE trends for all Configurations**Figure 3.11:** $c_{A,i}$ (Red) Vs $c_{D,i}$ (Blue) Vs $c_{initial}$ (Green) Vs Contact (Grey) : $c_{A,i}$ for $c_{D,i}$ is $c_{initial}$ for $c_{D,i+1}$

3.1.3 Task C: Cable Manipulation With Multiple Contacts

The model is further evaluated in an environment with two contacts, increasing the complexity. For cable routing tasks between two physical contacts, the cable is more likely to get stuck on the contact. The performance of the local linear model based control law is now evaluated in a more complex scenario. The length of the cable l_{cable} is 180 units and the number of tracking points N_t along the cable is 85, and the respective cable configuration at time t is $\mathbf{c}(t) = [\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \dots, \tilde{\mathbf{c}}_{85}]^T \in \mathbb{R}^{85 \times 3}$. The task is to manipulate the cable from the initial configuration Figure 3.12a to the configuration shown in Figure 3.12j. A total of nine desired configurations are given as input $\mathbf{C}_D = [\mathbf{c}_{D1}, \mathbf{c}_{D2}, \dots, \mathbf{c}_{D9}]^T \in \mathbb{R}^{9 \times 85 \times 3}$.

The maximum data size n_{max} is set to 5000, and the model is periodically retrained every 1000 steps. The error threshold ϵ for this task is set to 5 for the first seven interme-

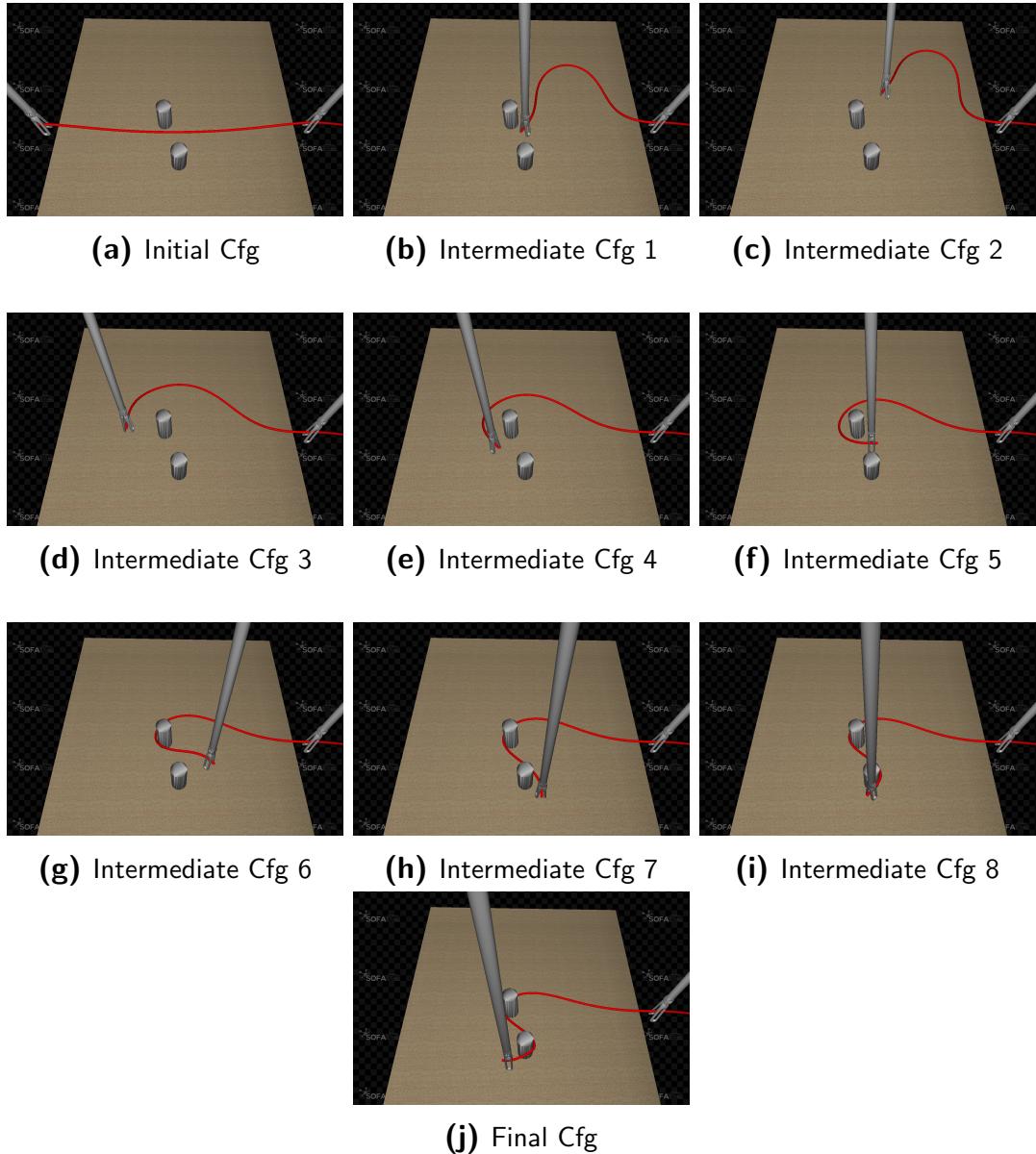


Figure 3.12: Initial Configuration $c_{initial}$ vs Desired Configurations $c_{D,i}$

diate configurations and 0.5 for the final two configurations. Different thresholds are used because it is not always necessary to achieve all the intermediate configurations precisely. However, higher accuracy should be achieved in the final stages to properly guide the cable to the desired configuration. To reinforce this, the intermediate configurations in the final stages are chosen relatively closer in comparison to the early stages. For the previous tasks, only one input was given as the initial displacement. However, for this task, two initial paths are given to ensure the model has enough data variation to achieve the task. The parameters selected for this task are summarized in Table 3.5.

Table 3.5: Parameters for Task C

Parameter	Value
Cable Length I_{cable}	180 units
Tracking Points N_t	85
Initial Gripper Position \tilde{r}_L	[4, 50, 85, 0°]
Initial Displacements $\tilde{r}_{Linitial}$	[70, 65, 30, 0°], [75, 60, 30, 45°]
Maximum Data Size n_{max}	5000
Training Interval t_{max}	1000
Error Threshold ϵ	5, 0.5
No. of goal configurations p	9

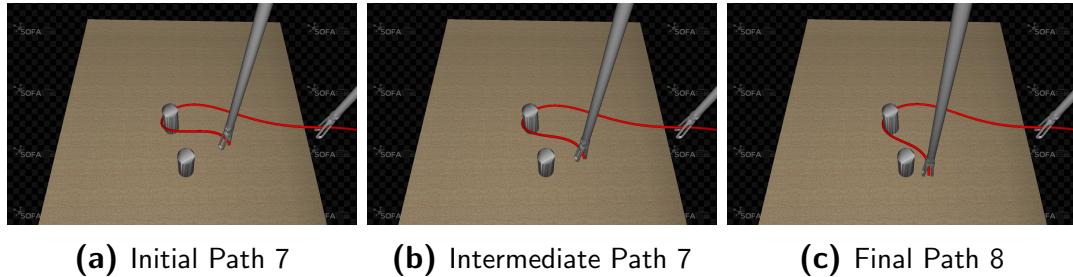
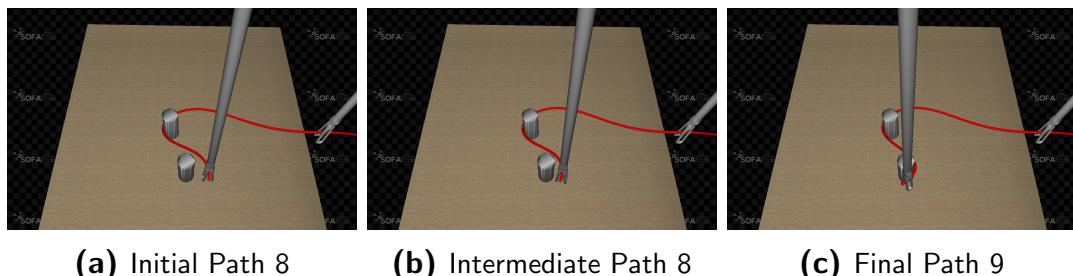
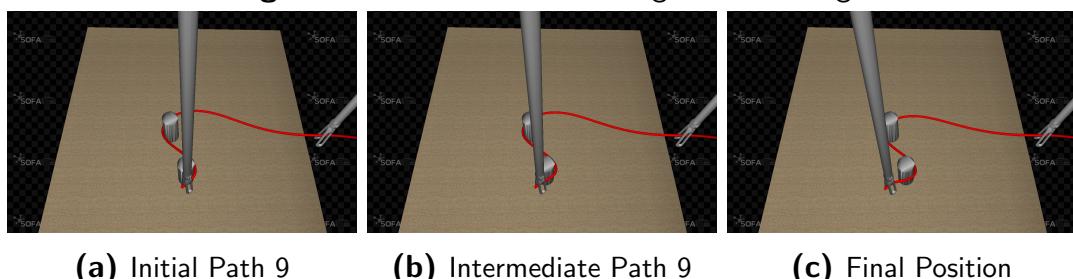
Control Path and MSE Analysis

The progression of mse values across the nine configuration transitions in Table 3.6 illustrates the performance and adaptability of the local linear model under challenging conditions with multiple contacts. The path followed for the first six configurations follow a similar pattern to those in tasks A and B. The later configurations are stuck on the contacts, and this behavior is explored. The transitions from the sixth configuration to the final configuration is illustrated, in Figure 3.16. It is observed that, particularly during transitions to configurations 2 and 3, where error differences are 213.93 and 578.05, respectively are high. These high values are attributed to the large difference between the input intermediate configurations. The transition error is much smaller in later stages as specified earlier.

The model was periodically retrained and, was able to reach these configurations consistently below error values $\epsilon = 5$. The final configuration reaching a minimal mse of 0.14, well within the stricter threshold $\epsilon = 0.5$ set for the final goal. The mse trend in Figure 3.18 shows that all configurations exhibit a steady convergence, with minor oscillations in configurations 4 and 5. The comparison between the achieved configurations and the given inputs is illustrated in 3.17. This demonstrates the framework's capability to adapt in tasks with entanglement, when sufficient data variations and training intervals are provided. Overall, the results validate the feasibility of using local deformation models for complex cable manipulation tasks in constrained environments and highlight the importance of model retraining and multi-step path planning.

Effect of Hyperparameters n_{max} and t_{max}

The performance of the local deformation model is influenced by two hyperparameters, the maximum data size n_{max} and the retraining interval t_{max} . The model performance in the third task under different n_{max} is illustrated in Table 3.7. It was observed that the selection of lower values n_{max} leads to inaccurate control paths and potential collisions or oscillatory behaviors. This can be observed for n_{max} values 1000, 2000, and 3000. Larger

Figure 3.13: transition from Cfg 6 to Cfg 7**Figure 3.14:** transition from Cfg 7 to Cfg 8**Figure 3.15:** transition from Cfg 8 to final Cfg**Figure 3.16:** Control Path Progression: Cfg 6 to Final Cfg

n_{max} enables the retention of more data, allowing the model to generalize deformations better. However, a very large n_{max} may not be optimal due to the presence of outdated data. These results validate that effective tuning of n_{max} based on the task is necessary for complex tasks.

The effect of t_{max} was also analyzed for Task A. The mse trends for different t_{max} values 800, 1000, and 2000 are presented in Figure 3.19. Lower values of t_{max} lead to unnecessary and frequent model retraining before sufficient data is accumulated. On the other hand, larger values delay model updates, which results in the model no longer accurately reflecting the local behavior. Both of these effects can be observed for t_{max} values 800 and 2000, respectively, in Figure 3.19. Nevertheless, the model still converged in all cases. The result shows that control paths can deviate from the target path, which can lead to failure in complex or contact-rich scenarios. Careful tuning of t_{max} ensures that the model adapts at appropriate intervals, maintaining both accuracy and stability.

Table 3.6: MSE Values at Different Transition Stages

Configuration Transition	Stage	MSE Value
Intermediate Cfg 1	After Initial Path 1	45.36
	At Intermediate Stage 1	13.50
	At Final path 2	1.68
Intermediate Cfg 2	After Initial Path 2	213.93
	At Intermediate Stage 2	61.86
	At Final path 3	4.85
Intermediate Cfg 3	After Initial Path 3	578.05
	At Intermediate Stage 3	130.46
	At Final path 4	2.89
Intermediate Cfg 4	After Initial Path 4	49.43
	At Intermediate Stage 4	13.36
	At Final path 5	3.44
Intermediate Cfg 5	After Initial Path 5	32.85
	At Intermediate Stage 5	12.69
	At Final path 6	4.78
Intermediate Cfg 6	After Initial Path 6	46.39
	At Intermediate Stage 6	22.38
	At Final path 7	3.36
Intermediate Cfg 7	After Initial Path 7	27.32
	At Intermediate Stage 7	15.00
	At Final path 8	2.77
Intermediate Cfg 8	After Initial Path 8	6.40
	At Intermediate Stage 8	3.55
	At Final Position 9	0.44
Final Cfg	After Initial Path 9	10.08
	At Intermediate Stage 9	4.94
	At Final position	0.14

Table 3.7: Effect of Varying n_{max} on MSE Across Configurations

n_{max}	Cfg 1	Cfg 2	Cfg 3	Cfg 4	Cfg 5	Cfg 6	Cfg 7	Cfg 8	Cfg 9
1000	1.68	4.23	1.67*	2.80	3.94*	Fail	—	—	—
2000	1.68	4.85	3.97	4.09	2.59*	0.83	Fail	—	—
3000	1.68	4.85	2.89	3.54	1.61*	4.98	4.41	Fail	—
4000	1.68	4.85	2.89	3.44	1.08*	3.62	3.75	0.40	0.30
5000	1.68	4.85	2.89	3.44	4.78	3.36	2.77	0.44	0.14
6000	1.68	4.85	2.89	3.44	4.78	2.42	4.24	0.26	0.29

*Inaccurate gripper action observed. Fail indicates failure due to collision or oscillatory behavior.

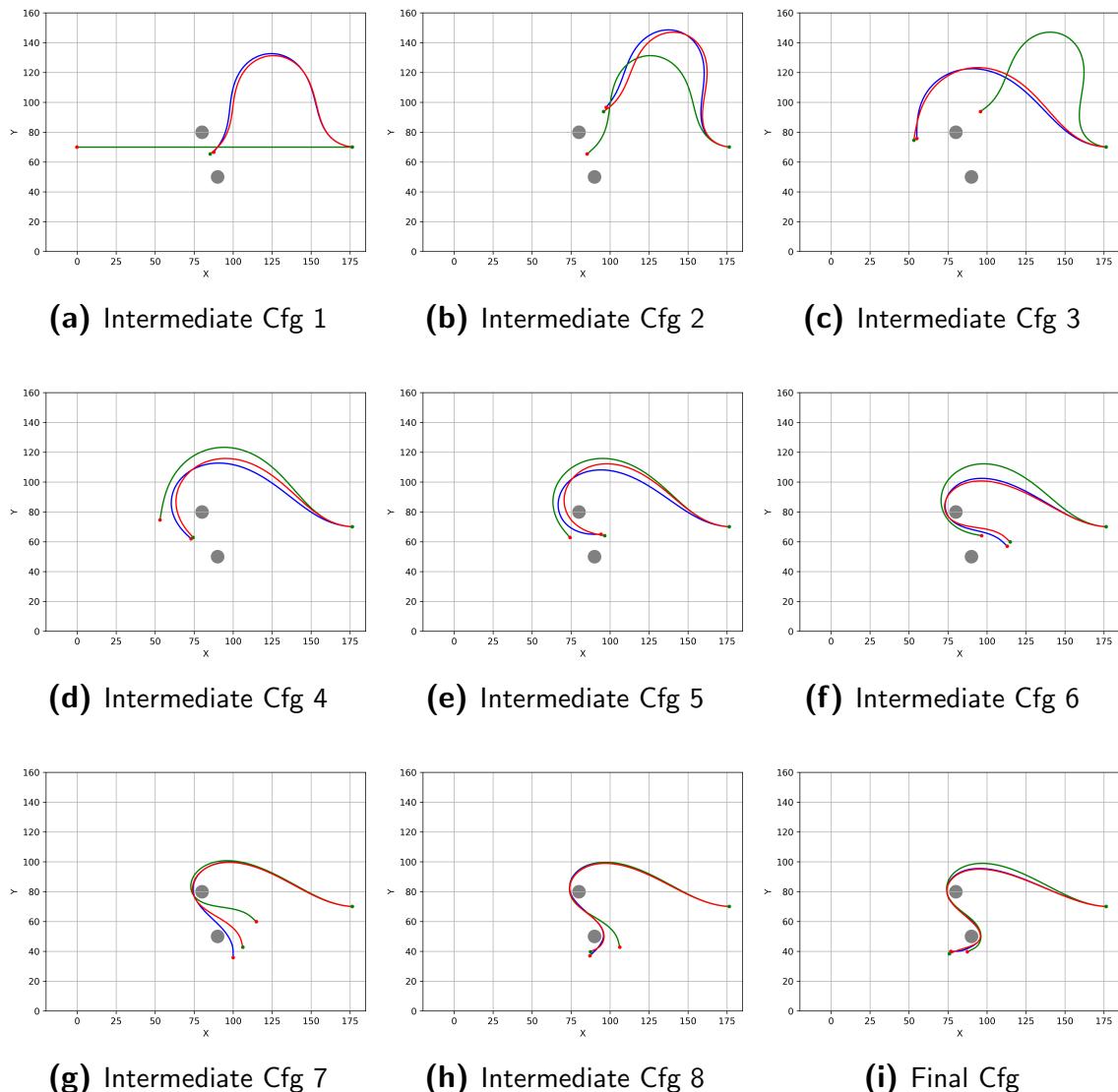
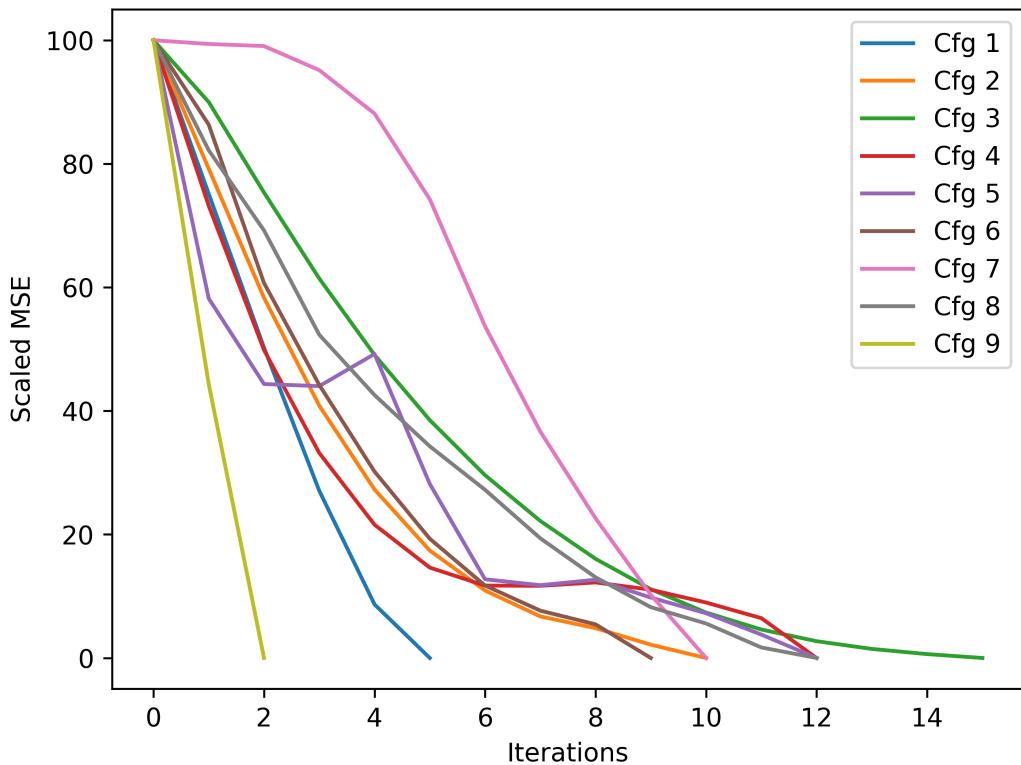
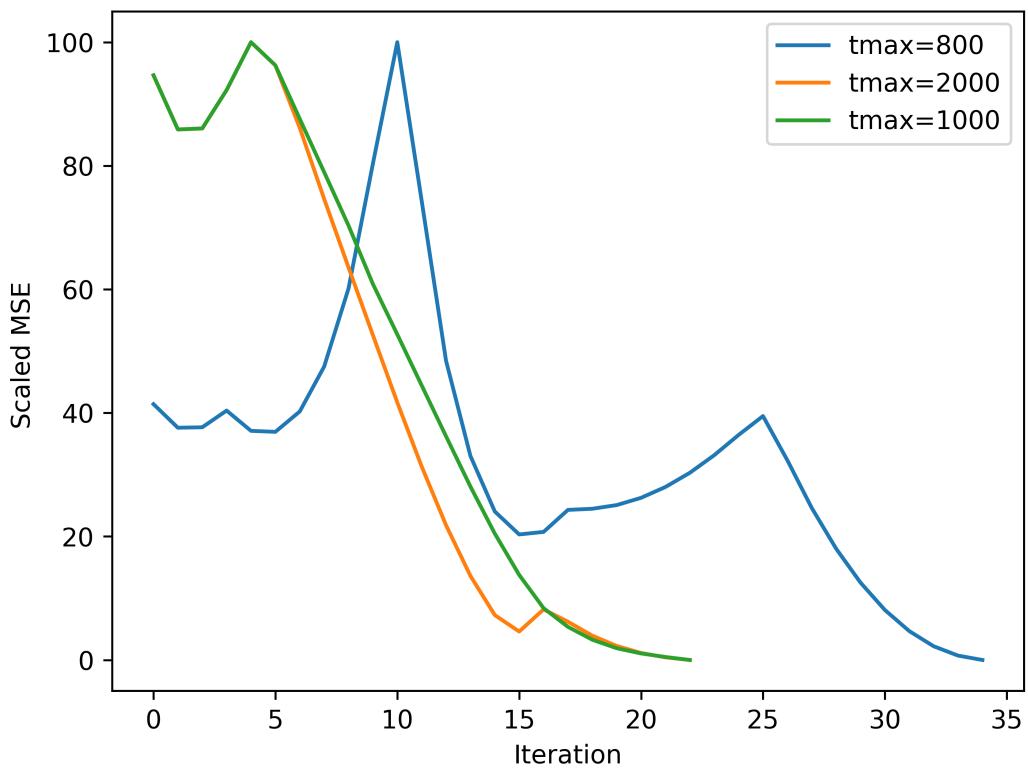


Figure 3.17: c_{Ai} (Red) Vs c_{Di} (Blue) Vs $c_{initial}$ (Green) Vs Contacts (Grey) : c_{Ai} for c_{Di} is $c_{initial}$ for c_{Di+1}

**Figure 3.18:** MSE trend for all Configurations**Figure 3.19:** MSE trend for different t_{max} values

4 Conclusion

This chapter concludes the thesis by summarizing the key contributions and experimental findings. It also outlines potential directions for extending the proposed framework to address current limitations and explore new capabilities in robotic manipulation of deformable linear objects.

4.1 Summary

In this thesis, a data-based method for robotic shape control of DLOs was proposed. The approach leveraged local linear models to control cable shapes through a multi-step adaptive planning algorithm. The system was implemented in the SOFA simulation framework. A local Jacobian-based model was developed to estimate the relationship between gripper actions and the resulting deformations. Using this model, an advanced algorithm incorporating flexibility and adaptability was developed. The proposed framework was evaluated in three tasks with increasing complexity: manipulation without contact, with a single contact, and with multiple contacts. The proposed method was deployed successfully in complex tasks with proper tuning and sufficient intermediate configurations. The effects of hyperparameters, the maximum data size n_{max} and retraining interval t_{max} , were also analyzed. Results showed that improper tuning of these parameters can lead to instability or failure, supporting the need for task-specific tuning.

A key limitation of the current approach is its dependence on manually selected intermediate configurations and the need for the initial calibration. For even more complex tasks that involve more contacts, a large number of intermediate configurations may be required. Overall, the results validate the feasibility of using local deformation models for complex cable manipulation tasks. The proposed framework offers a promising alternative to traditional model-based approaches that rely heavily on accurate physical parameters for planning.

4.2 Future Scope

Several directions remain unexplored for future development.

- **Automatic Generation of Intermediate Configurations:** In contact-rich environments, manually specifying intermediate configurations is not practical. Future work can focus on algorithms that automatically generate intermediate configurations, negating the need for manual recording.
- **Real-World Testing:** The simulated environment does not consider real-world robotics, where data is collected using sensors. Such data is often noisy and less structured than in simulation environments. Therefore, testing in the real world is necessary.
- **Multi-Gripper Manipulation:** Extending the control strategy to manipulate both grippers simultaneously could enhance task efficiency and robustness.
- **Integration with Global Planners:** Combining the local deformation model with global motion planning frameworks could enable full end-to-end execution of complex manipulation tasks.

In conclusion, this work lays a foundation for shape manipulation of DLOs using local linear models, offering a direction for the future.

Bibliography

- [1] J. Sanchez, J. A. Corrales Ramon, B. C. BOUZGARROU, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *The International Journal of Robotics Research*, vol. 37, pp. 688 – 716, 2018.
- [2] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, p. eabd8803, 05 2021.
- [3] J. Zhu, D. Navarro-Alarcon, R. Passama, and A. Cherubini, "Vision-based manipulation of deformable and rigid objects using subspace projections of 2d contours," 06 2020.
- [4] M. Yu, K. Lv, H. Zhong, S. Song, and 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, pp. 417–436, 2023.
- [5] S. Jin, C. Wang, and M. Tomizuka, "Robust deformation model approximation for robotic cable manipulation," 11 2019, pp. 6586–6593.
- [6] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, *SOFA: A Multi-Model Framework for Interactive Physical Simulation*, 06 2012, vol. 11.
- [7] E. Yoshida, K. Ayusawa, I. G. Ramirez-Alpizar, K. Harada, C. Duriez, and A. Kheddar, "Simulation-based optimal motion planning for deformable object," 06 2015, pp. 1–6.
- [8] 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018.* IEEE, 2018. [Online]. Available: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8574473>
- [9] A. Theetten, L. Grisoni, C. Andriot, and B. Barsky, "Geometrically exact dynamic splines," 01 2006.
- [10] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *International Journal of Robotics Research*, vol. 33, no. 1, pp. 48–68, Jan. 2014.

- [11] N. Umetani, R. W. Schmidt, and J. Stam, "Position-based elastic rods," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, ser. SCA '14. Eurographics Association, 2015, pp. 21–30.
- [12] F. Liu, E. Su, J. Lu, M. Li, and M. Yip, "Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–8, 07 2023.
- [13] P. Kicki, M. Bidziński, and K. Walas, "Learning quasi-static 3d models of markerless deformable linear objects for bimanual robotic manipulation," 2023. [Online]. Available: <https://arxiv.org/abs/2309.07609>
- [14] R. Laezza, R. Gieselmann, F. T. Pokorny, and Y. Karayiannidis, "Reform: A robot learning sandbox for deformable linear object manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4717–4723.
- [15] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," 2021. [Online]. Available: <https://arxiv.org/abs/2011.07215>
- [16] Y. Yang, J. A. Stork, and T. Stoyanov, "Learning differentiable dynamics models for shape control of deformable linear objects," *Robotics and Autonomous Systems*, vol. 158, p. 104258, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889022001518>
- [17] R. Lee, M. Hamaya, T. Murooka, Y. Ijiri, and P. Corke, "Sample-efficient learning of deformable linear object manipulation in the real world through self-supervision," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 11 2021.
- [18] K. Almaghout and A. Klimchik, "Planar shape control of deformable linear objects," *IFAC-PapersOnLine*, vol. 55, pp. 2469–2474, 10 2022.
- [19] A. Szymko, P. Kicki, and K. Walas, "Calibrationless bimanual deformable linear object manipulation with recursive least squares filter," *IEEE Access*, vol. PP, pp. 1–1, 01 2024.
- [20] M. Yu, H. Zhong, and X. Li, "Shape control of deformable linear objects with offline and online learning of local linear deformation models," 05 2022, pp. 1337–1343.
- [21] B. Cao, X. Zang, X. Zhang, Z. Chen, S. Li, and J. Zhao, "Shape control of elastic deformable linear objects for robotic cable assembly," *Advanced Intelligent Systems*, vol. 6, 04 2024.