# Deformable Object Manipulation with a Tactile Reactive Gripper

by

Neha Sunil

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021



Author .....
Department of Mechanical Engineering
Aug 6, 2021

Certified by .....
Alberto Rodriguez
Associate Professor
Thesis Supervisor

Accepted by .....
Nicolas Hadjiconstantinou
Department Graduate Officer

# Deformable Object Manipulation with a Tactile Reactive Gripper

by

Neha Sunil

Submitted to the Department of Mechanical Engineering
on Aug 6, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

## Abstract

Deformable objects like cloth and cables are challenging for robots to manipulate due to their high-dimensionality and unpredictable dynamics. In previous work, Yu et. al (2019) [45] used a tactile sensor to estimate the pose of a cable within the grip while sliding along it. The authors used linear regression to model the cable sliding dynamics and used a linear quadratic regulator (LQR) controller to keep the cable centered within the grip. However, the underlying dynamics are not linear, so in this work, we explore controllers that take advantage of a non-linear underlying dynamics model. We use Gaussian process (GP) regression for the non-linear model which is used in three controllers in hardware experiments: (1) LQR with the GP model linearized about the target position and (2) time-varying LQR with the GP model linearized about the current state and (3) model predictive control with the full dynamics model and constraints on the state and input of our system over a finite horizon. We extend our framework for the more challenging task of cloth edge following by adjusting our hardware setup and developing a new perception system. We found that the time-varying LQR controller using the GP model performs similarly to the LQR controller with the linear regression model for following both cables and fabric edges.

Thesis Supervisor: Alberto Rodriguez
Title: Associate Professor

# Acknowledgments

Firstly, I would like to thank my advisor Professor Alberto Rodriguez for all the guidance and feedback. I am so grateful for his insights and encouragement while developing my research goals as well as the constant support through the past few years. I genuinely feel so lucky to be part of this lab and have truly enjoyed my time here. I am excited to see what the next few years will bring.

I would also like to thank my labmates at MCube and collaborators for all of their guidance and help. In particular, I would like to thank Siyuan and Shawn. They have each taught me so much and have been thoughtful, knowledgeable, and patient mentors.

I also want to thank all the professors in each class I have taken here at MIT. I appreciate all the work professors put into teaching and helping me learn, especially changing their curriculums during the pandemic. I have been able to directly apply so many of the concepts from my classes here to my research. I also want to thank all the TA's for their time and help.

Finally, I'd like to thank all my friends and family for their endless support. My friends have gotten me through the ups and downs, and have made the past few years at MIT so unforgettable and valuable. I will always treasure the memories from all of our trips, yoga classes (whether on Killian or in the Charles), and spontaneous adventures.

The research in this thesis was supported by the Amazon Research Awards(ARA), the Toyota Research Institute (TRI), the Google Faculty Research Awards, and the Office of Naval Research (ONR) [N00014-18-1-2815]. I have been supported by the National Science Foundation Graduate Research Fellowship [NSF-1122374], the Mechanical Engineering Distinguished Fellowship, and the David S.Y. and Patrick Wong Fellowship. This article solely reflects the opinions and conclusions of its authors and not Amazon, Toyota, Google, ONR, or NSF.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The robotic manipulation of highly deformable objects is a growing research area due to its applicability in a variety of fields such as cable routing in factories, assisted dressing or laundry folding in the home and healthcare industry, or suturing in robotic surgery. However, manipulating deformable objects is difficult due to their continuous and highly-underactuated nature.

To address these challenges, much of the work on manipulating deformable objects have utilized mechanical constraints [61, 66, 37]. For example a rope may be placed on a table, so that gravity and friction yield a quasistatic configuration of the cable. A gripper can then adjust the rope configuration, step by step, at a chosen pace.

In contrast, we are interested in real-time, dextrous manipulation by exploiting tactile perception. Specifically, we focus on the two tasks of following along a cable and the edge of a towel. Humans can grasp an object loosely between the thumb and forefinger and slide the fingers to a target position as a robust strategy to regrasp it. For example, when trying to find the plug-end of a loose headphone cable, one may slide along the cable until the plug is felt between the fingers (Fig. 1-1a). Similarly, when folding a bedsheet, one starts by grabbing a corner and tracing along the edge until reaching another corner.

Contour following on deformable objects is especially challenging because the object's shape changes dynamically with the sliding motion, and there are unpredictable factors such as kinks, variable friction, and external forces. Our goal is to manipulate

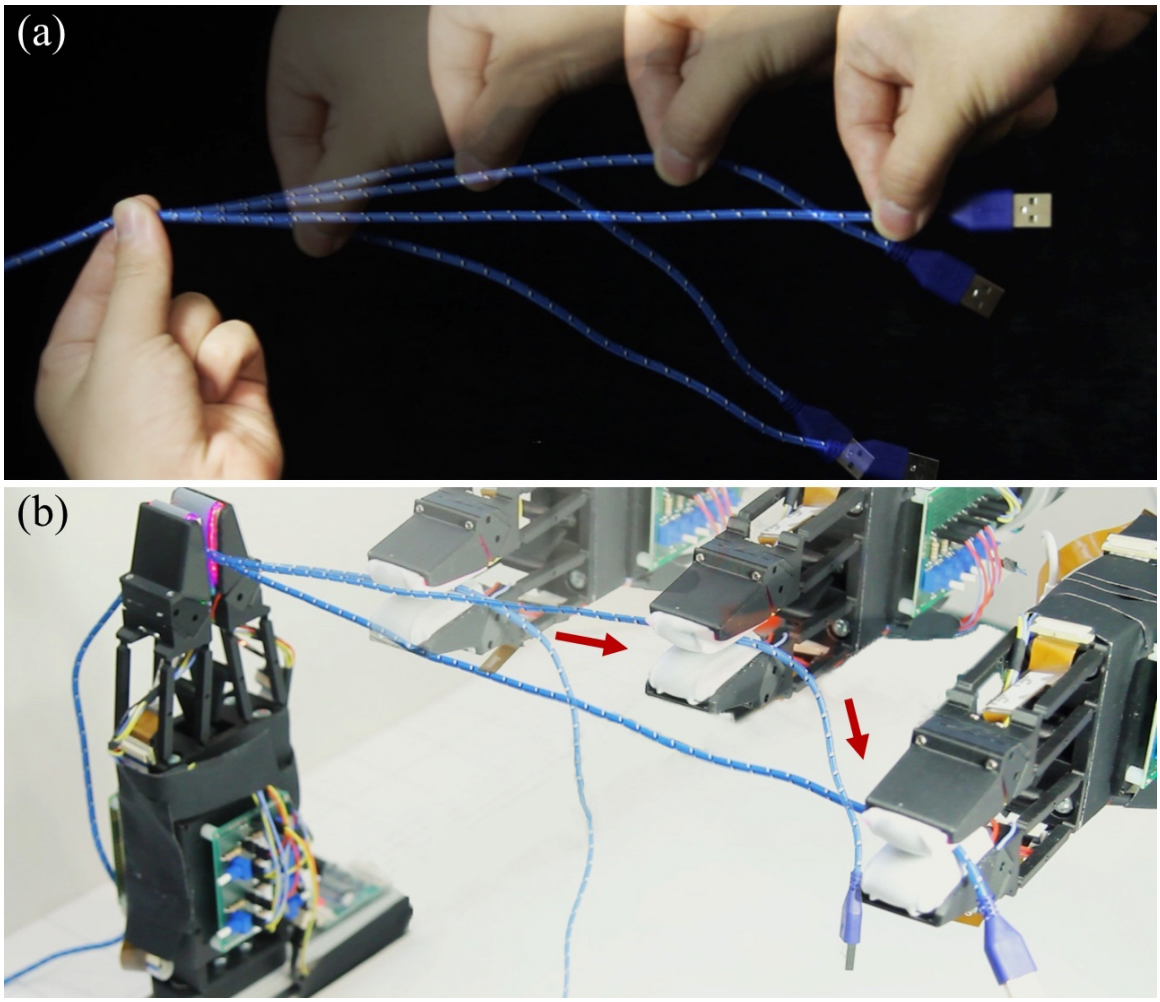


Figure 1-1: Following a cable with (a) human hands and (b) robotic grippers.

cables in real time, using a pair of grippers, with no added mechanical constraints. The cables are free to wiggle, swing, or twist, and our grippers must rapidly react using tactile feedback. In particular, we look at the task of picking one end of a cable with a gripper and following it to the other end with a second gripper, as shown in Fig. 1-1b.

Continuing off of the work from Yu et. al (2019) [45], we do not use vision in this research, relying on tactile sensing alone. While vision can be helpful, we are able to perform the task purely with tactile guidance. Deformable objects are easily occluded from view by grippers, by the environment, and often by itself. Tactile perception allows for precise localization of a grasped object. Tactile active perception, like when pulling from the two ends of a cable until it is in tension, can also be used to simplify

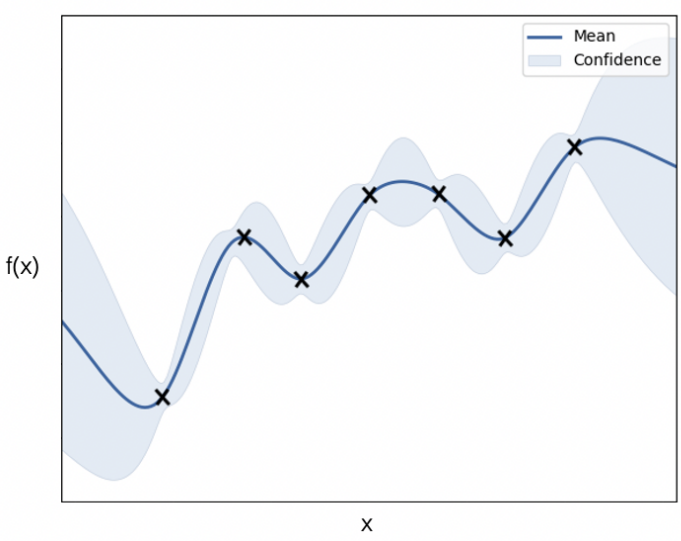


Figure 1-2: Example of GP Regression

perception such as in the case of a tangled rope.

For cable sliding, Yu et. al (2019) [45] learned a linear model of the cable sliding dynamics—how pulling angle affected the in-grip cable pose. They used a linear quadratic regulator (LQR) controller based on the linear model to keep the cable centered within the gripper to prevent it from falling while sliding in real time.

In this work, we will compare different control methods that take advantage of a nonparametric Gaussian process (GP) dynamics model to the LQR controller based on a linear dynamics model. Chapter 1 focuses on following cables while Chapter 2 covers towel-edge following. Both chapters discuss the perception and control strategies. While the general framework is the same for following both objects, cloth following is more challenging from both the perception and control perspectives.

## 1.1 Gaussian Process Models

Gaussian processes (GPs) are a Bayesian approach to function regression, allowing us to place a prior distribution over the space of functions. The approach has become one of the standard tools for solving non-linear regression problems [40]. The GP is a nonparametric method, meaning that the hypothesis space cannot be parameterized

by a finite set of parameters. As the number of data points increases, so do the number of model parameters. Figure 1-2 shows an example of a GP regression model given a set of data points.

The GP variables that can be tuned are the kernel type and the kernel hyperparameters. The hyperparameters include the lengthscale (which controls the global smoothness, or how quickly the function varies) and the variance. In terms of advantages for using this in a controller, GPs provides continuous and differentiable functions as well as uncertainty estimates at each point.

## 1.2 Related Work

### 1.2.1 Contour following

Contour following of rigid objects has been widely studied using both visual [27] and tactile perception [6, 54]. These techniques do not directly translate to deformable objects due to dynamic shape changes that are difficult to model, especially in real time.

The most similar contour following work to ours is by Hellman et al. (2017), who proposed a reinforcement learning approach to close a deformable ziplock bag with feedback from BioTac sensors [16]. The work demonstrated a robot grasping and following the edge of the bag. In contrast to our approach, they use a constant grasping force and discrete slow actions. As a consequence, they achieve a maximum speed of 0.5 cm/s, compared to 6.5 cm/s in our work.

### 1.2.2 Cable/rope manipulation

Manipulating deformable linear objects (DLOs) has attracted attention in the robotics community [19] with tasks including tying knots [36, 41], untangling [32, 13], insertion [53], reshaping [61, 66, 37], surgical suturing [33], or dynamic rope manipulation [58, 65]. Our approach to cable manipulation through tactile perception and control is fundamentally distinct from the existing literature and enables a larger potential

action space.

### Approaches

Much of the classical work in DLO manipulation involves perceiving the state of the DLO, simulating the dynamics of the DLO, or planning its motion. Visual perception for DLO state estimation is difficult given the infinite dimensional configuration space with the object's shape dynamically changing while often being occluded. Morita et al. (2003) described rope state topologically by listing intersections created by rope crossings [36]. Methods that more completely describe location along the entire length of the DLO often use non-rigid registration techniques to track the rope from a known initial state [48, 7]. Alternatively, an initial state estimate from a given point cloud can be refined to better align with the system dynamics [32, 22].

The most common methods for simulating DLOs use mass-spring models [52], energy minimization [4], or finite-element methods (FEM) [38]. These models can be computationally expensive and require knowledge of the DLO's physical properties such as rigidity, elasticity, and friction. Berenson (2013) avoids an explicit deformable object model by using an approximation to the Jacobian of the deformable object to drive object points to a target set [3]. Work by [59] also avoids complicated dynamics models by moving the cables at high enough speeds that they assume each rope segment follows the motion of the robot with a constant time delay.

Motion planning for DLOs has traditionally used sampling-based approaches such as a probabilistic roadmap (PRM) [24] or Rapidly-exploring Random Trees RRTs [28]. Moll and Kavraki (2006) used these methods to create local planners based on minimum energy curves [35]. For knot-tying, Saha et al. (2008) plan long-horizon, complex motions by simulating deformations of a rope in response to random external forces and placing configurations that would be part of the knot's topological forming sequence in a PRM [41]. McConachie et al. (2020) combines topological-based and sampling-based motion planning to move the object into a position where the local controller can complete the task [34].

Learning-based approaches can help simplify aspects of the problem. Given the

inherent difficulty of complete state estimation, some DLO manipulation works are trained directly from visual data without explicitly estimating the full state [13, 50, 37]. Nair et al. (2017) learns a pixel-level inverse dynamics model for a rope with self-supervised autonomous pick and place interactions [37]. More recently, [47, 12] use dense object descriptors to find pixel-wise correlations between images of ropes trained in simulation. Another class of work learns dynamics models for rope and uses them with Model Predictive Control (MPC). Work by Li et al. (2019) extends interaction networks to learn dynamics models [31], and Ebert et al. (2015) learns a video prediction model [10]. While these methods work for short-horizon tasks like shaping, planning for more complex, long-horizon tasks like knotting requires more guidance, for example learning from demonstration [43]. While these data-driven methods allow for faster computation, they are less generalizable for other DLO manipulation tasks.

### Rope Manipulation Skills

Due to their high dimensional dynamics, manipulating deformable linear objects is usually simplified by constraining their motion with external features, for example against a table [61, 66, 37], with additional grippers [33], or pegs [41]. Another common strategy involves limiting movements to long series of small deformations with pick and place actions [61, 37]. Thus, the dynamics of the system can be treated as quasistatic.

Furthermore, the action space in DLO manipulation literature is generally limited to those using fixed grasps of the DLO. Besides pick and place, other actions include following specific, potentially dynamic, trajectories [58], moving a segment of a rope using two grippers [34], insertion [53], and wrapping [67].

Few works exploit sliding along the DLO. Zhu et al. (2019) routed a cable around pegs using one end-effector that was attached to the cable end and another fixed end-effector that would passively let a cable slip through in order to pull out a longer length of cable [67]. This system, while allowing for sliding, loses the ability to sense and control the state of the cable at the sliding end, which can be in contact with

any point of the cable.

Jiang et al. (2015) traces cables in a wire harness using a gripper with rollers in the jaws [23]. This gripper passively adjusts grip force using springs to accommodate different sized cables. They sense and control the force perpendicular to the translational motion along the cable in order to follow the cable. The cables in our work are considerably smaller and less rigid, so such forces would be difficult to sense.

Furthermore, both of the specialized, passive end-effectors in the above two works have limited capabilities beyond sliding along the cable. In our work, the parallel jaw gripper used to follow a cable can also be used to insert the cable into a headphone jack [45], demonstrating the potential of this hardware setup for additional tasks.

Another example of work involving sliding with rope from Yamakawa et al. (2007) shows how our tactile perception framework could potentially be extended for the knot-tying task [60]. To pass one end of the rope through a loop, they leverage tactile sensing to roll the two rope ends relative to each other in between the fingers.

### 1.2.3 Cloth Manipulation

Many of the techniques in the DLO manipulation literature section above have also been applied on cloth. This extends to perception [7, 12], planning [34] and control [10]. McConachie et al. (2020) uses the cloth geodesic between the two grippers as a representation of object state for their planner while navigating around obstacles [34]. They assume the rest of the loose cloth will not get caught on obstacles as long as the geodesic moves around them. Thus, they can treat the cloth like a piece of rope.

#### Perception

Much of the work on cloth-centric perception for manipulation focuses on corner detection and ridge detection for determining grasp points for future manipulation tasks [55]. Corner detection can be done through computer vision techniques (e.g. Harris corner detection [15]) and is useful for tasks involving folding and unfolding.

Learning-based perception approaches allow for more flexibility. Ganapathi et al. (2020) train dense object descriptors using a cloth simulator to find pixel-wise correspondences across fabric in different configurations [11]. Qian et al. (2020) trains a network to segment edges and corners of cloth from a depth image by using a painted cloth as a ground truth label [39].

These techniques all work better for specific initial configurations of the fabric. Therefore, flattening and smoothing fabric is a common manipulation task. Flattening also allows for more robust clothing classification. Most of these papers use pick and place actions to flatten the fabric [46, 44, 11]. Ha and Song (2021) use a high-velocity flinging motion to unfold clothing [14].

#### Sliding

The sliding contour following skill used for cables is also relevant for fabric manipulation. Most work in fabric manipulation similarly uses the quasistatic assumption and incremental pick and place movements [56, 20, 11]. However, the sliding skill simplifies the task of finding two adjacent corners in order to fold a piece of fabric. Sahari et al. (2010) holds up a corner of the fabric and uses gravity to trace straight down to find the second corner without sensory feedback [42]. Similarly, Yuba et al. (2017) executes an open loop "pinch and slide" motion along the top edge of a piece of fabric [63].

### 1.2.4 Force and Tactile Sensing

Besides the knotting example from Yamakawa et al. (2007), force and tactile sensors can be seen in a variety of rope manipulation literature [60]. Abegg et al. (2000) uses a force torque sensor to detect changes in contact state, for example the rope moving from free space to contacting an edge of a rigid object [2]. Yue and Henrich (2002) detects vibration frequency of a rope using a force torque sensor before counteracting the vibration [64].

In our work, we use GelSight [62] for tactile perception to estimate the pose of the

cable in grasp. She et al. (2019) also uses it to approximate shear force by tracking the black markers on the sensor surface as a proxy for friction force while sliding [45]. Gelsight, and other vision-based tactile sensors, convert touch to vision by using a camera to visualize the deformation of the contact surface. With its high spatial resolution, this type of sensor shows unique advantages and has been successfully utilized in different robotic manipulation tasks, for instance, contour following [29], cutting [57], dish loading [25], in-hand manipulation [26], etc.

GelSight, in particular, has been used in various manipulation problems. Li et al. (2014) implemented a USB insertion task from random grasping poses based on object pose estimation feedback from the sensor [30]. Izatt et al. (2017) used the 3D point cloud from GelSight sensor in a Kalman filter to better register the position of a screwdriver in a peg-in-hole task [21]. Calandra et al. (2018) and Hogan et al. (2018) used the tactile images to evaluate the quality of a grasp and further infer better regrasp positions [5, 18]. Dong et al. (2018) used the sensor to predict slip and used it slip signal to modulate grasping forces while conducting a bottle cap screwing task [8]. Tian et al. (2019) proposed a tactile-based model predictive control method to reposition an object [49]. Dong and Rodriguez (2019) trained a tactile-based object insertion policy that could correct small misalignment between the object and the environment [9]. Hogan et al. (2020) designed closed-loop tactile controllers for dexterous table-top manipulation with dual-arm robotic palms, simultaneously controlling the contact state and the object state [17]. Wang et al. (2020) implemented a task of swing an elongated object to a target pose based on the learned friction, center of mass properties of the grasp object with the GelSight sensor [51].

# Chapter 2

# Cable Following

The goal of the cable following task is to use a robot gripper to grip the beginning of the cable with proper force and then control the gripper to follow the cable contour all the way to its tail end. The beginning end of the cable is initially firmly gripped by another fixed gripper during the cable following process.

Dynamically sliding along the edge of a cable is difficult because the object's shape changes unpredictably with the sliding motion, presenting a challenge for both perception and control. In previous work, Yu et. al (2019) [45] used a tactile sensor to estimate the pose of a cable within the grip while sliding along it. The authors used linear regression to model the cable sliding dynamics and used a linear quadratic regulator (LQR) controller to keep the cable centered within the grip. However, the underlying dynamics are not linear, so in this work, we explore controllers that take advantage of a non-linear underlying dynamics model.

We compare three controllers using the Gaussian Process (GP) to our linear model baseline: (1) LQR with the GP model linearized about the target position and (2) time-varying LQR (tvLQR) with the GP model linearized about the current state and (3) model predictive control (MPC) with the full dynamics model and constraints on the state and input of our system over a finite horizon. LQR has a closed form solution that is fast to compute compared to MPC. Only the MPC controller directly uses the non-linearized GP dynamics model.

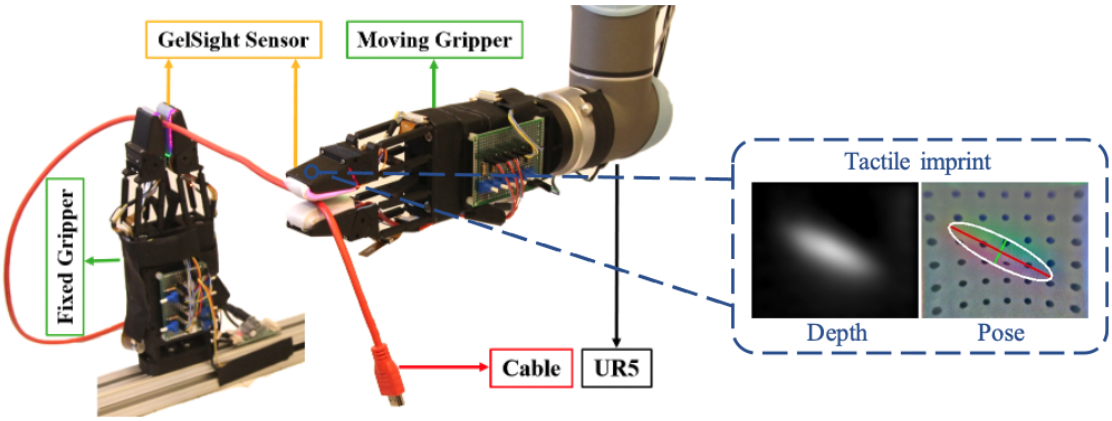


Figure 2-1: **Experimental setup.**

## 2.1 Methods

Both the fixed gripper and sliding gripper are outfitted with GelSight Sensors, but we only use the signal from the sliding gripper. The sliding gripper is attached to a UR5 robot arm (Fig. 2-1).

### 2.1.1 Perception

Figure 2-2 illustrates different cable pose estimations from tactile images. First, we compute depth images from the raw tactile images by estimating surface normal and applying Fast Poisson Solver (FPS) for integration [62]. Then, we extract the contact region by thresholding the depth image. Finally, we use Principal Component Analysis (PCA) on the contact region to get the principal axis of the imprint of the cable on the sensor.

### 2.1.2 Control

We model the dynamics as a planar pulling problem. The system's state is $\mathbf{x} = [y\ \theta\ \alpha]^T$, where $y$ is the cable height from the center line of the gripper, $\theta$ is the cable angle with respect to the gripper, and $\alpha$ is the angle between the two grippers (Fig. 2-3). The control input $\mathbf{u} = [\phi]$ is the pulling angle. The linear dynamic model is of the form:

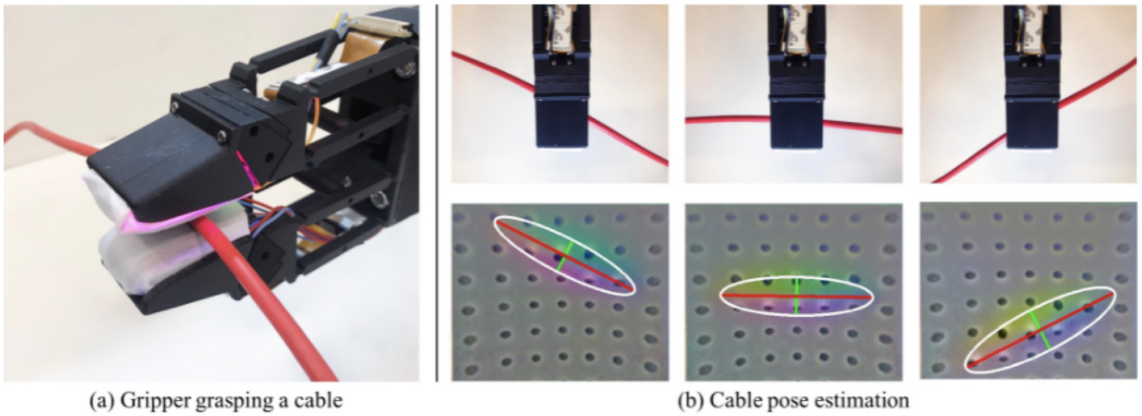


Figure 2-2: **Tactile perception.** (a) Gripper with GelSight sensors grasping a cable. (b) Top view of the gripper grasping different cable configurations and the corresponding cable pose estimations. The white ellipse shows the estimation of the contact region. The red and green lines show the first and second principal axes of the contact region, with lengths scaled by their eigenvalues.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{2.1}$$

where $A$ and $B$ are the linear coefficients of the model.

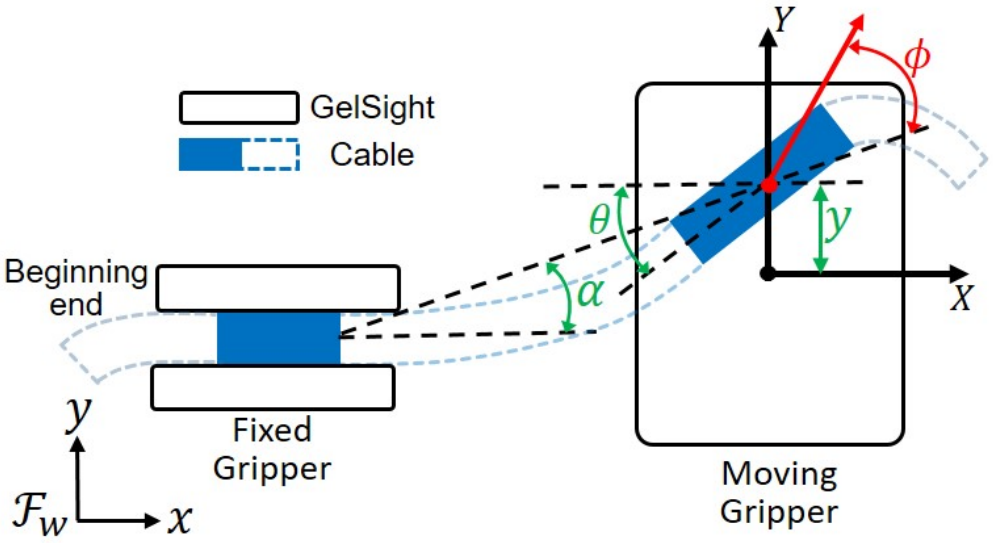


Figure 2-3: **Cable-gripper dynamics model**.

To learn a dynamics model, Yu et. al (2019) [45] collected data using a simple proportional controller supplemented with uniform noise. The dataset has 24,695

observations with a variety of starting states and pulling angles. Three separate GP models are fit to each state variable in $\dot{\mathbf{x}}$.

We tuned GP parameters on this dataset with the same hardware setup from [45], but decided to use a newer hardware setup (different grippers with different stiffness values) for our robot experiments. We collected 15,276 observations using a proportional controller ($\phi = -Ky$) from 20 following runs with this new hardware setup. We used the optimized parameters from the older dataset to train GP models with the new dataset.

We used the GPy [1] python package to perform Gaussian process regressions and tested a variety of their built-in kernels. Given that the training time increases with dataset size N as O($N^3$), we first compared kernels by training on 10% of the data and testing on the held-out testing set. We incrementally increased the percent of data used for training for the best-performing models. We also compared performance with and without automatic relevance determination (ARD), which sets a separate length-scale for each input. We used limited-memory BFGS optimization to tune hyperparameters. After determining the best kernels, we compared the performance of a sparse GP regression trained with the same kernel using a subset of the N training points. If the subset consists of M inducing points, the computation time becomes O($M^2N$) for training and O(M) for predicting the mean instead of O(N).

To linearize the GP model about a certain state $x_0$ and control input $u_0$, we differentiated the GP at this operating point using GPy's built-in predictive gradient function to get the $A$ and $B$ matrices. We then formulated an LQR controller to find the optimal feedback gain $K$, which we use to find the optimal input, $\mathbf{u} = -K\mathbf{x}$. For the LQR parameters, we start with $\mathbf{Q} = [10, 1, 0.1]$ and $\mathbf{R} = [0.1]$, since regulating $y$ and $\theta$ (so the cable does not fall) is more important than regulating $\alpha$ (to keep the cable straight). Additionally, if the cable moves too far in the y-direction, the tactile sensor will not be able to determine the cable pose estimate.

The first baseline controller is LQR using the linear dynamics model. The second LQR controller linearizes the GP model at the origin. The tvLQR controller linearizes the GP model about the current state.

Table 2.1: $R^2$ values of models trained on 10% of dataset with ARD.

| **Kernel** | $\dot{y}$ $R^2$ | $\dot{\theta}$ $R^2$ | $\dot{\alpha}$ $R^2$ |
|---|---|---|---|
| RBF | 0.873 | 0.861 | 0.752 |
| Rational Quadratic | 0.899 | 0.901 | 0.793 |
| Matern-3/2 | 0.895 | 0.894 | 0.828 |
| Matern-5/2 | 0.891 | 0.876 | 0.767 |
| Exponential | 0.896 | 0.897 | 0.859 |
| Linear | 0.707 | 0.556 | 0.592 |

For the MPC controller, we sampled potential actions using the cross-entropy method. The dynamics were defined using the GP model. The pulling direction was constrained between $-\pi/3$ and $\pi/3$. State constraints were imposed as well. When the controller could not find a feasible solution, we used the LQR controller with the linear model to find a default action.

## 2.2 Results

### Optimizing GP Parameters

As seen in Table 2.1, from the initial GP models using 10% of the data, we found that the RBF, Rational Quadratic, Matern-3/2, Matern-5/2, and Exponential kernels were worth training with a larger percent of the training data. We also found that ARD improved the model performance for every kernel, except the linear one (Table A.1). Full training results with larger percentages of data can be seen in Table A.2.

After incrementing the percent of the dataset used for training to 50%, we found that $\dot{y}$ was best modeled with a Matern-3/2 kernel using ARD with a MSE of 8.5 nm/s and $R^2$ of 0.977. Both $\dot{\theta}$ and $\dot{\alpha}$ performed best with an exponential kernel with ARD, resulting in MSEs of 0.005 deg/s and 0.071 deg/s and $R^2$ values of 0.982 and 0.962 respectively.

We also used the best performing kernels to train sparse GP models with 300 and 2000 inducing points. The $\dot{y}$ sparse models had MSEs of 52 nm/s and 22 nm/s and $R^2$ values of 0.863 and 0.941 respectively. We can run LQR to find K with the GP linearized at distinct operating points at 20 Hz for the 50% GP model and 60 Hz for

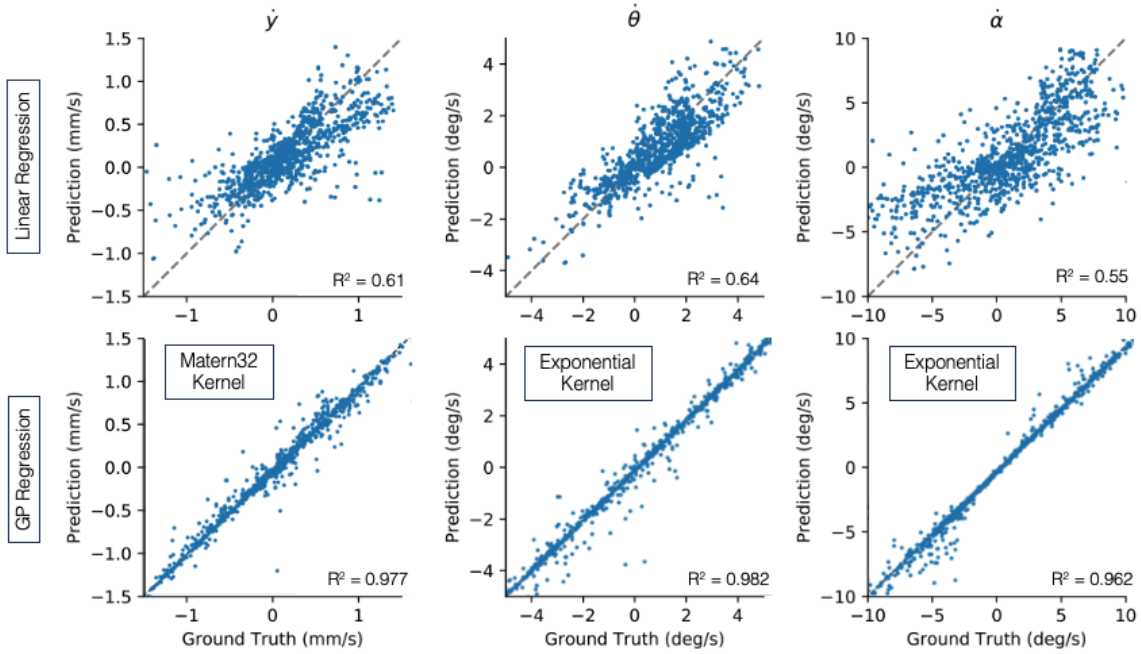


Figure 2-4: **Predicted vs. actual velocity** $\dot{y}$**,** $\dot{\theta}$, and $\dot{\alpha}$, of the generalized coordinates of the cable-gripper dynamics, as defined in Fig. 2-3 for linear and GP models.

the sparse GP model with 300 inducing points.

### GP Model with New Hardware

We trained four sparse GP models with 200, 400, 1000, and 2000 inducing points using the Matern-3/2 kernel for the $\dot{y}$ model and the exponential kernel for both $\dot{\theta}$ and $\dot{\alpha}$ models, all with ARD (Table 2.2). The y and $\theta$ components of the inducing points for the latter three models are plotted in Fig. 2-5. The inducing points, chosen by the optimizer in GPy, are mostly near the origin, with a few outside of the region of feasible states. This is useful since the covariance kernels have a zero-mean prior, meaning that far from the training points, the sampled functions would return to zero with a non-linear kernel. All of the sparse GP models were significantly more accurate than the linear model (Fig. 2-6).

Table 2.2: $R^2$ values from training sparse GP models on new hardware.

| **Num Inducing Points** | $\dot{y}$ $R^2$ | $\dot{\theta}$ $R^2$ | $\dot{\alpha}$ $R^2$ |
|---|---|---|---|
| 200 | 0.880 | 0.568 | 0.900 |
| 400 | 0.891 | 0.646 | 0.913 |
| 1000 | 0.917 | 0.693 | 0.938 |
| 2000 | 0.937 | 0.786 | 0.969 |

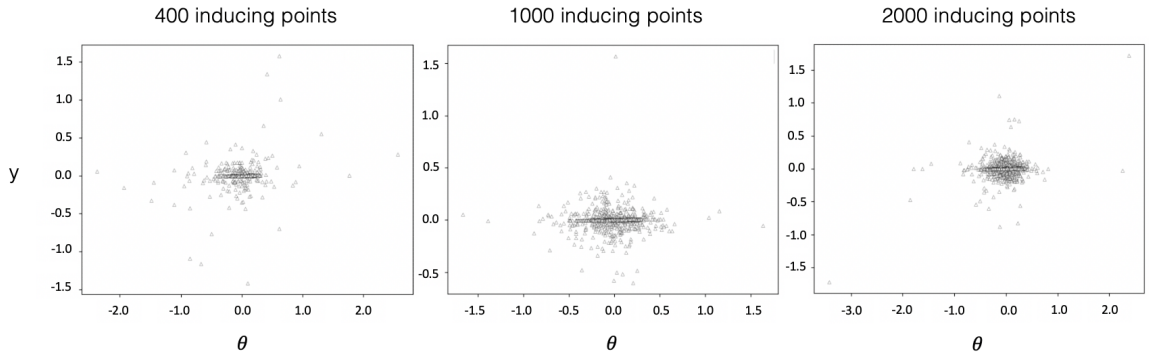


Figure 2-5: **Plot of $y$ and $\theta$ components of inducing points**.

### Controller Performance

To compare the different LQR controllers, we ran 5 trials of each and averaged the percent of the workspace traveled. The robot was stopped after traveling 0.65 m, the length of the workspace. Each controller was run at a nominal velocity of 0.02 m/s. The LQR controller based on the linear model traveled an average 93.4% of the workspace. The LQR controller based on the GP model with 200 inducing points linearized at the origin traveled an average 61.8% of the workspace. The tvLQR controller (based on the GP model with 400 inducing points) traveled an average 93.5% of the workspace. Even though the GP model is significantly more accurate than the linear model, none of the GP-based controllers significantly outperformed LQR using the linear model.

The LQR controller using the GP model linearized at the origin had smaller gains that worked as long as the cable was relatively centered. However, small kinks in the cable resulted in deviations that the controller could not correct for fast enough. This controller would likely be more effective with a less rigid cable with a good initial

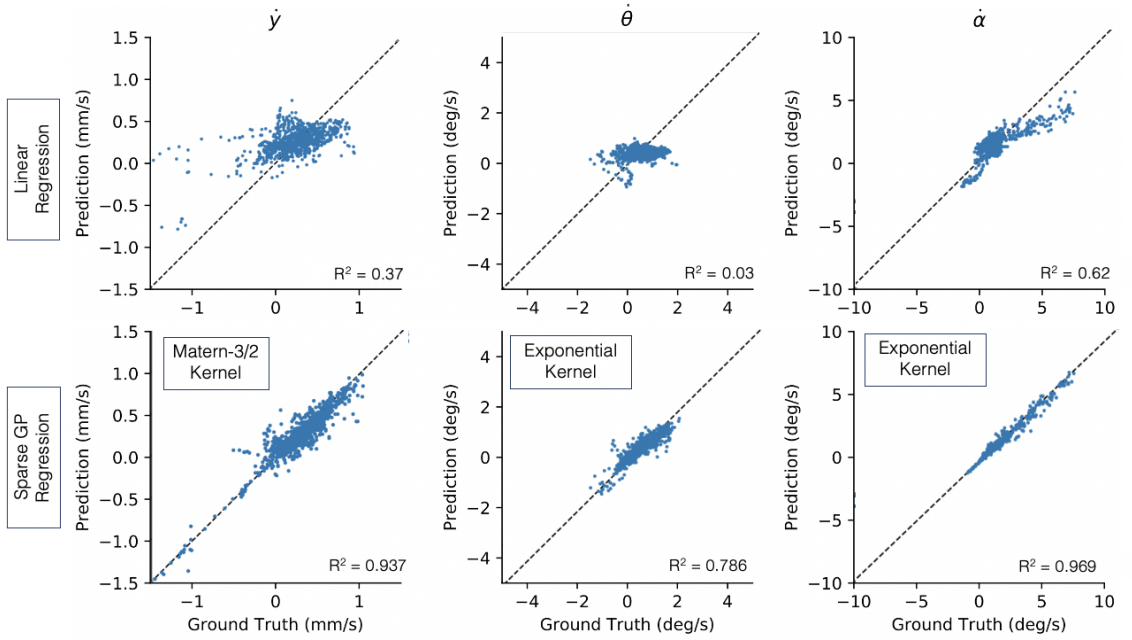


Figure 2-6: **Predicted vs. actual velocity** $\dot{y}$**,** $\dot{\theta}$, and $\dot{\alpha}$, of the generalized coordinates of the cable-gripper dynamics, as defined in Fig. 2-3 for linear and sparse GP models with new experimental hardware.

grasp near the origin.

The tvLQR controller required significant tuning to perform comparably to the linear model LQR controller. For the previous LQR controllers, we used $\mathbf{Q} = [10, 1, 0.1]$. For the TVLQR controller, we set $\mathbf{Q} = [10000, 1, 0.1]$ to properly prioritize keeping the cable centered within the grip. Figure 2-7 shows the difference in commanded actions in the world frame across two axes of the state space. We found that the number of inducing points in the GP model significantly impacted the performance of the controller. Looking at the plot of the inducing points (Fig. 2-5), the GP model may be overfitting to the points near the origin. Since the tvLQR controller uses predictive gradients, overfitting is problematic for this controller. Overfitting was especially evident for the 1000 and 2000 inducing point GP models. Furthermore, the initial training and testing data was collected by randomizing observations (individual timesteps) from only 20 runs. As we collected more data, we added this to the dataset. A larger dataset with more independent samples helped improve the generalizability of the GP model.

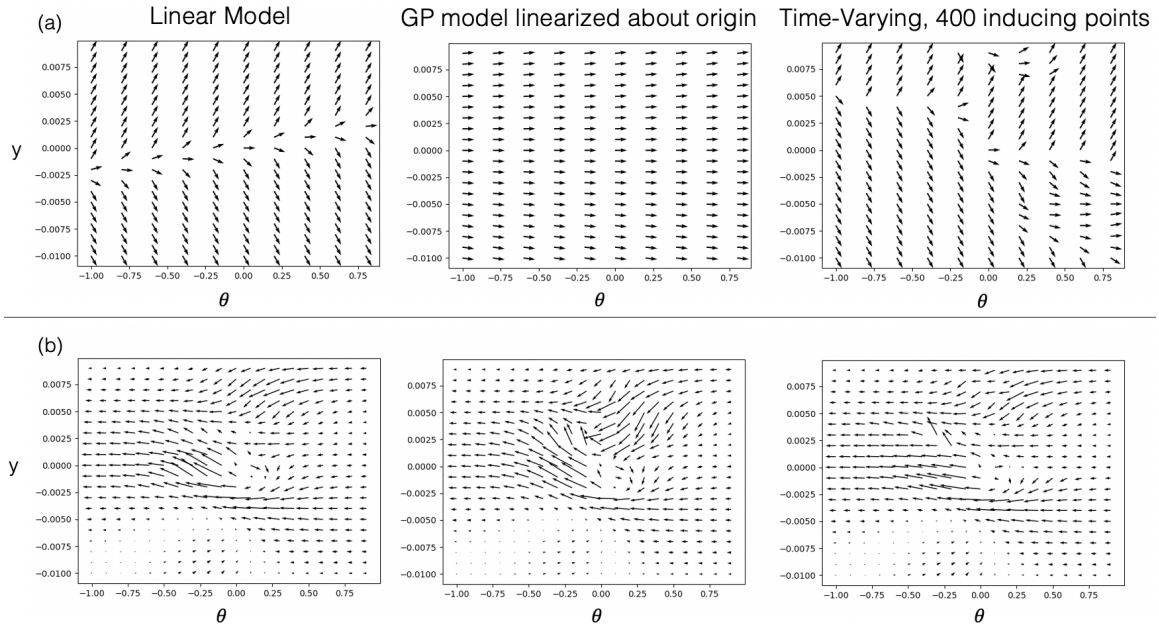


Figure 2-7: **Comparison of (a) commanded pulling angle in the world frame and (b) phase portraits across state space with fixed $\alpha$ using different controllers**. Note that since $\alpha$ is fixed to 0, the trajectories in the phase portrait are not representative of the movement of the cable.

Other less effective tuning approaches included multiplying each kernel with a linear kernel so the function would not return to a mean of zero in regions without data. Additionally, we tried creating a training set of points along a meshgrid of the input space and using K-Nearest Neighbors regression to assign output values to each node. Increasing the length-scale hyperparameter slowed how quickly the function varied. These methods did not improve performance beyond the linear model baseline.

For the MPC controller we settled on a time horizon of 5, 30 rollouts, 5 elites, and 4 iterations. We used the same quadratic cost from the LQR controller, but added the model uncertainty provided by the GP model to our cost function to avoid actions that resulted in future states in poorly fit regions of the GP model. When the controller was not able to find a solution given the state and action constraints, we used LQR with the linear model as the default action. This controller was able to consistently follow to the end of the workspace, but used the default action about

60% of the time. We believe that the GP model with 2000 inducing points was not accurate enough to consistently make reasonable predictions, requiring the default LQR controller to bring the cable back to the center of the gripper when it was too close the edge of the constraint space. While increasing the number of inducing points could result in a more accurate model, more points would not be tractable for real-time control.

## 2.3 Conclusion

We compare controllers based on a linear model and a nonlinear Gaussian process model to slide along a cable using a tactile-reactive gripper. The tvLQR controller performed comparably to the LQR controller based on the linear model, both following the cable for an average of 93% of the workspace distance. The controller linearized about the origin of the GP model performed the worst out of the three LQR controllers with an average following distance of 62% of the workspace. This controller was not able to quickly adapt to kinks in the cable and external perturbations that moved the cable away from the origin. We found that the GP model was not accurate enough for useful commanded actions from the MPC controller.

To improve the GP model for the controllers, we first want to collect more data with more independent following runs. We can also add more noise to the commanded action to have more data across the state space. Additional potential controllers include an iterative LQR control and an MPC controller with the dynamics model linearized about the current state.

Future tasks that can build off cable following include cable routing, untangling, and rope knotting. While all these tasks can feasibly be performed with only pick and place actions, sliding along the cable can speed up completion of the task and simplify the planning involved. Since constraining the DLO is crucial for many manipulation tasks, sliding while keeping the cable in tension allows for easy state estimation between the two grippers.

# Chapter 3

# Cloth Edge Following

In the cloth following task, a moving gripper with a fixed grip on the cloth pulls the cloth through a stationary gripper while attempting to keep the edge visible within the grasp of the stationary gripper, as shown in Fig. 3-1. The goal is to traverse as much of the fabric edge as possible without losing the grasp, and ideally while maintaining an estimate of the edge pose.

Applying the same cable pose estimation method (PCA) to soft cloths would be challenging because the contact region is no longer an ellipsoid. The fabric is relatively thin and covers a large area (see Fig. 3-2), unlike the cable which creates a distinct imprint. Traditional computer vision edge detection techniques were unsuccessful at consistently isolating the fabric edge due to the fabric texture and noise in the tactile image. Instead, we use a supervised data-driven method to estimate the pose of the fabric edge.

In terms of the pose controller, soft cloths are also more challenging than relatively stiff cables for robotic manipulation. We also found that the weight from the rest of the towel (the extra dimension compared to cables) made the towel much more likely to slip out of the grip.

We expect that leveraging extrinsic dexterity like the gravity of fabric for self-straightening or partial table support with external forces, will be important. But in general, the idea of following tactile features (like principal axis/edges) of deformable objects can serve as an alternative motion primitive to alleviate the complexity of state

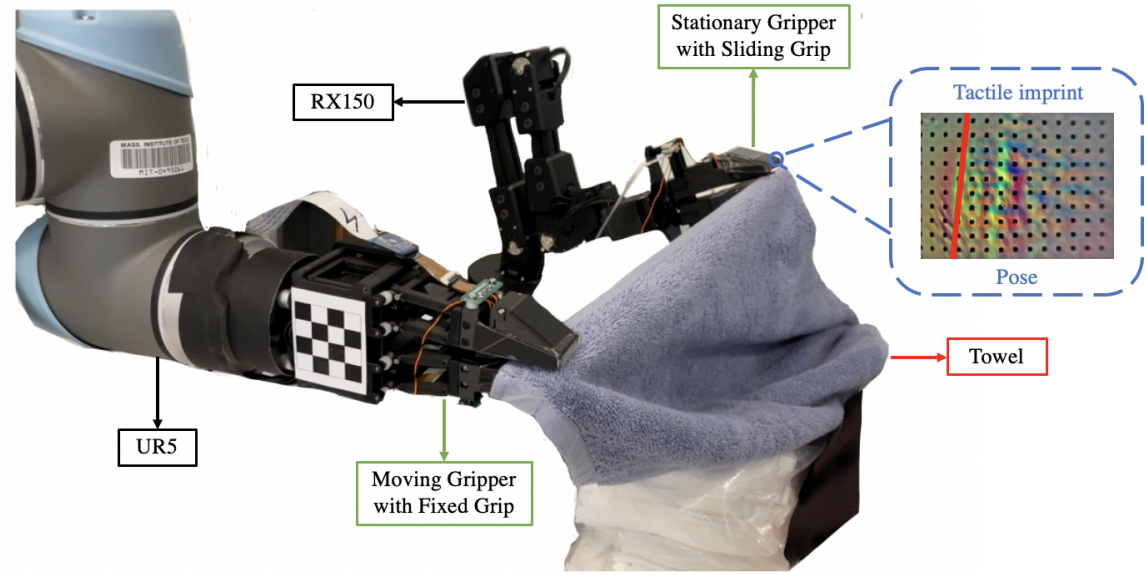


Figure 3-1: **Experimental setup.**

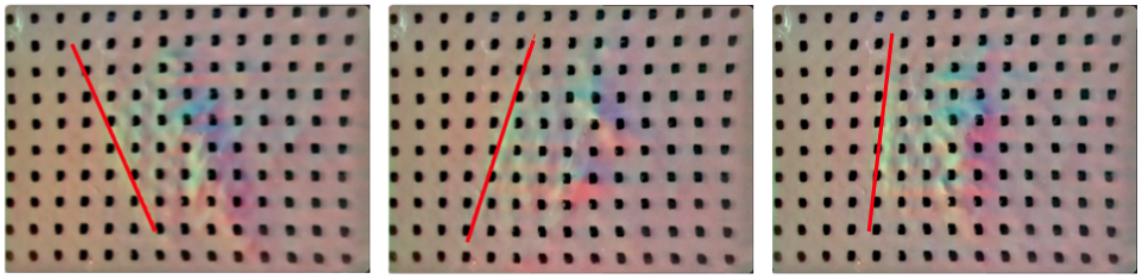

Figure 3-2: **Example tactile images of fabric edge.** The hand-drawn red lines indicate the location of the fabric edge.

and dynamic modeling. This chapter focuses on following using the most effective controllers from Chapter 2: LQR with a linear model and tvLQR with the GP model.

## 3.1 Methods

We tested several following configurations and discovered that pulling the cloth through a gripper was easier than tracing along an edge to maintain tension in the fabric. To address the issue of the extra weight pulling the towel out of the grasp, we placed supports under the stationary gripper to relieve some weight and tilted the gripper upward by 30 degrees (Fig. 3-1).

We also found that the grippers that worked well for cable following were not sensitive enough for fabric manipulation. We first increased the maximum gripping force from 10N to 20N. Additionally, we changed the gel composition to make it softer and therefore providing a more sensitive signal.

### 3.1.1 Perception

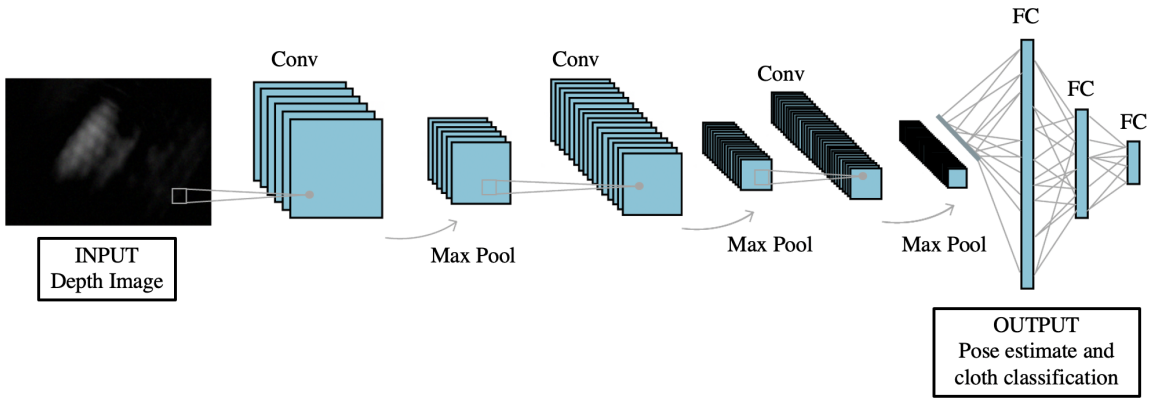


Figure 3-3: **Perception network architecture.**

Our perception system takes a down-sampled depth image as input and outputs (1) a classification of whether an edge is visible, the tactile sensor is entirely covered with fabric, or there is no fabric in the grip and (2) if an edge is visible, the x and y coordinates of the center of the edge and the orientation of the edge. The network we trained has 3 convolutional layers followed by max pooling, and 3 fully connected layers (Fig. 3-3).

We initially intended on using the raw tactile image as input to the network to save the computation time required to compute the depth image. We tried rendering a difference image from the augmented depth image (See B-1). However, we found that the reconstructions were not accurate enough for our purposes. Therefore, we use the depth images as input.

The dataset is generated by hand-labeling a series of training images (frames from a video), computing the depth image, augmenting the depth image by randomly varying the maximum depth threshold, and further augmenting the images by introducing

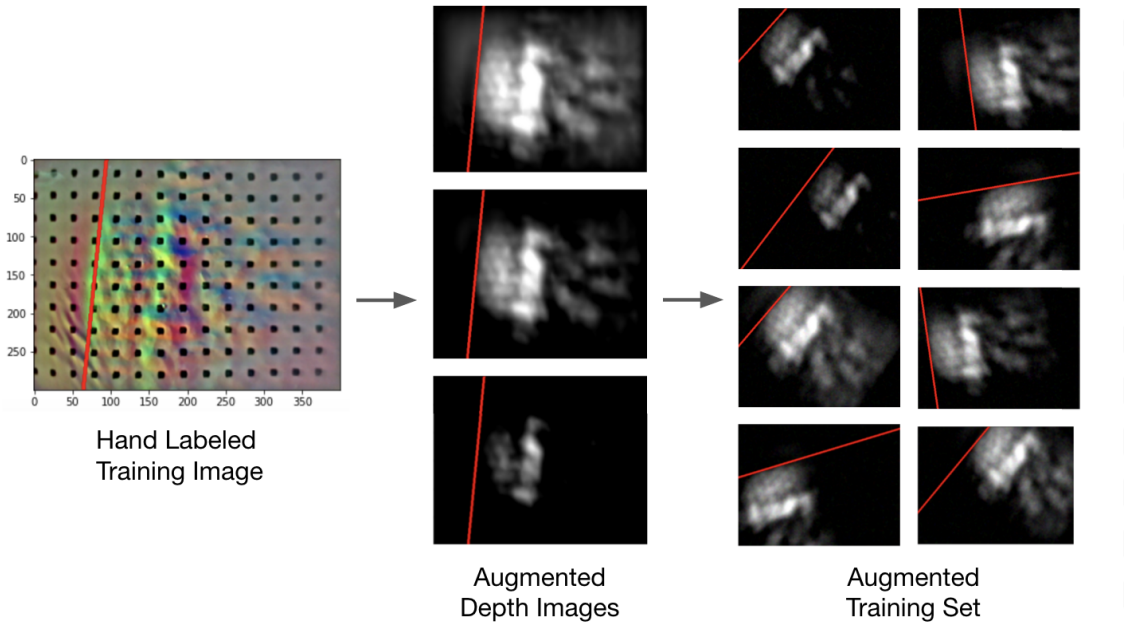


Figure 3-4: **Dataset generation.**

a random translation and rotation (Fig. 3-4). The human labeling process involves clicking two points to define the cloth edge if visible or classifying the image as all fabric or no fabric. The hand-labeled edge is transformed to stay consistent with the augmented image.

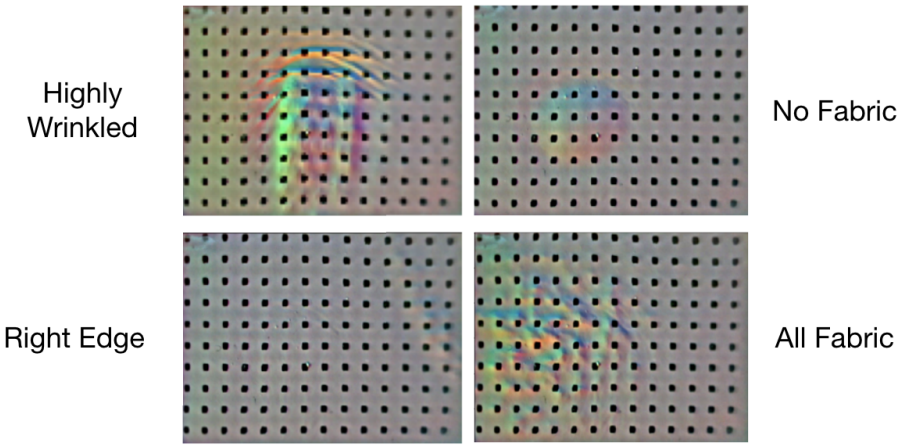


Figure 3-5: **Examples of special case fabric GelSight images.**

The final network we trained had 150 raw training images which we augmented to 30,000 depth images which were training inputs to the network. The training dataset

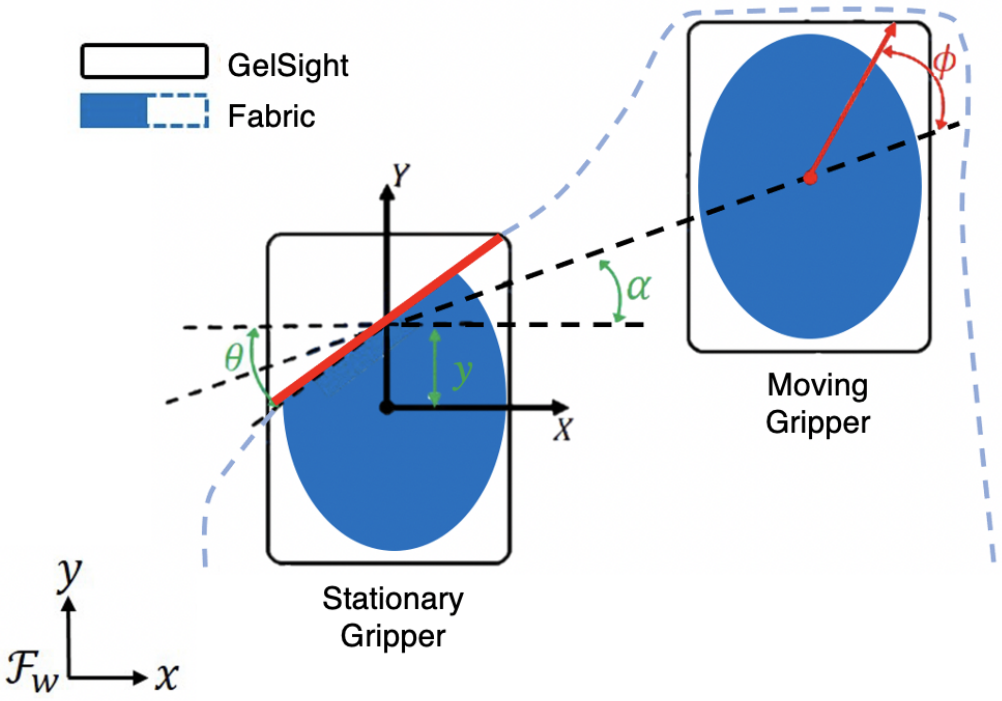


Figure 3-6: **Cloth-gripper dynamics model.**

specifically included images where the edge was close to the boundary of the sensor, the gel surface was wrinkled, the sensor was completely covered with fabric, and no fabric was touching the sensor (Fig. 3-5).

### 3.1.2 Control

We model the dynamics as shown in Fig 3-6. Like the cable following system, the system's state is $\mathbf{x} = [y\ \theta\ \alpha]^T$, where $y$ is the fabric edge height from the center line of the gripper, $\theta$ is the fabric edge angle with respect to the gripper, and $\alpha$ is the angle between the two grippers (Fig. 3-6). The control input $\mathbf{u} = [\phi]$ is the pulling angle. The linear dynamic model is of the form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{3.1}$$

where $A$ and $B$ are the linear coefficients of the model.

To learn a dynamics model, we collected data using a simple proportional controller ($\phi = -Ky$) supplemented with uniform noise. The dataset has 7100 observations from 30 different following runs with a variety of starting states and pulling angles. Three separate GP models are fit to each state variable in $\dot{\mathbf{x}}$.

We performed both linear regressions and sparse GP regressions with 500 inducing points and the best kernels from the cable following experiments (Matern-32 for $\dot{y}$ and exponential for $\dot{\theta}$ and $\dot{\alpha}$) with automatic relevance determination (ARD) to independently adjust the lengthscales for each input dimension.

We compare the two best performing controllers for cable following for cloth edge following. The first baseline controller is LQR using the linear dynamics model. The second tvLQR controller linearizes the GP model about the current state. For the LQR parameters, we use $\mathbf{Q} = [100000, 1, 0.1]$ and $\mathbf{R} = [0.1]$, to prioritize regulating $y$ and $\theta$ (so the cable does not fall).

## 3.2 Results

We were able to run the tactile perception network at 33 Hz. The network was reliable at categorizing the no fabric and all fabric cases. When the fabric is at the edge of the sensor, however, the categorization and pose estimation is less robust. The convexity of the gel surface results in the center of the sensor being the most sensitive.

The GP model with the Matern-32 kernel was not able to properly capture the dynamics of $\dot{y}$. It tended to predict a velocity close to zero no matter the initial state.

However, we did see significant improvement in the sparse GP models compared to the linear models for $\dot{\theta}$ and $\dot{\alpha}$ (Fig. 3-7). The linear model had MSEs of 2.08e-02 deg/s and 1.03e-03 deg/s and $R^2$ of 0.080 and 0.896 for $\dot{\theta}$ and $\dot{\alpha}$ respectively. The sparse GP models had MSEs of 4.45e-03 deg/s and 1.57e-04 deg/s and $R^2$ of 0.769 and 0.957 for $\dot{\theta}$ and $\dot{\alpha}$ respectively.

When we tested these controllers on the physical system, we found that both controllers performed comparably. We adjusted the y setpoint to be much closer to the inner edge than with cable following. With an optimal starting position, the LQR

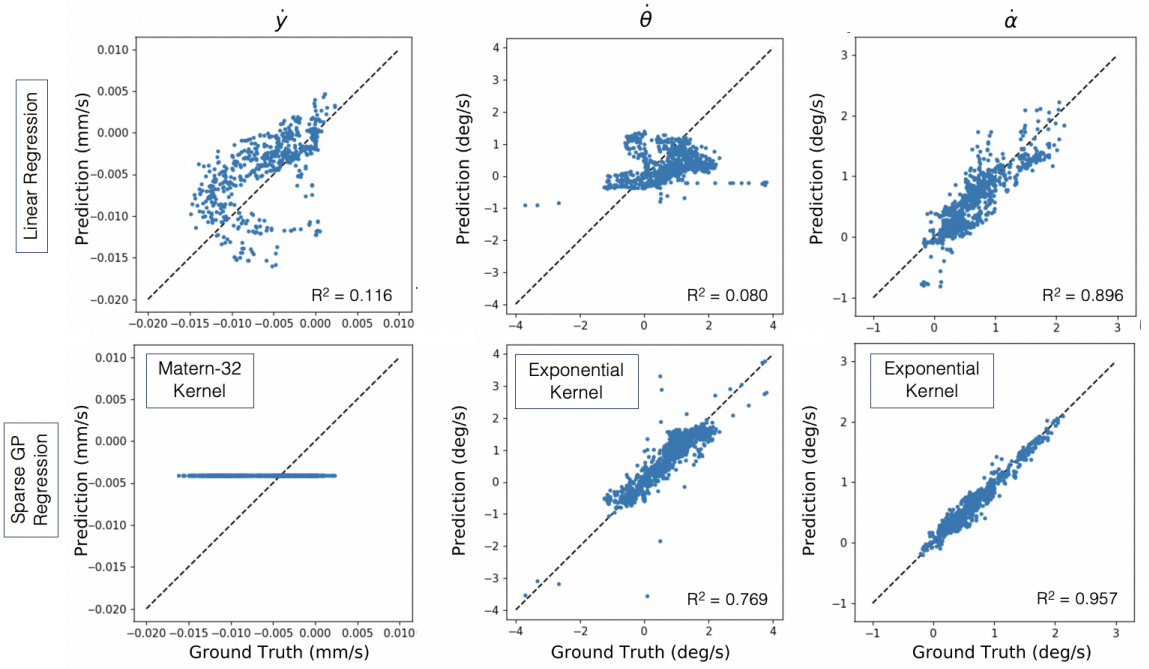


Figure 3-7: **Predicted vs. actual velocity** $\dot{y}$**,** $\dot{\theta}$, and $\dot{\alpha}$, of the generalized coordinates of the fabric-gripper dynamics, as defined in Fig. 3-6 for linear and sparse GP models.

controller based on the linear model consistently traveled to the end of the workspace in all five trials. When the initial position was in the center of the gripper (closer to the tip compared to the setpoint), the gripper traversed an average of 71.8% of the workspace (40 cm out of 56 cm).

We found that the tvLQR network with a Matern-32 kernel instead of a linear kernel (essentially Bayesian linear regression) for the $\dot{y}$ model prioritized keeping $\theta$ close to zero compared to the other state variables despite y having a larger cost 3-8. This is due to the model not being able to capture the dynamics of $\dot{y}$. However, performance was highly dependent on starting position. Ideally, the fabric starts with small $\theta$ and y, meaning the fabric edge is closely aligned with the inner edge of the sensor. When y becomes too large, the fabric quickly slips out of the grip due to the weight of the fabric. If the sensor is completely covered with fabric and cannot sense the edge, we found that the gripper often goes too far into the fabric. This allows the gripper to follow to the end of he workspace, but we no longer have a proper estimate of the fabric edge. With an initial position of all fabric on the sensor,

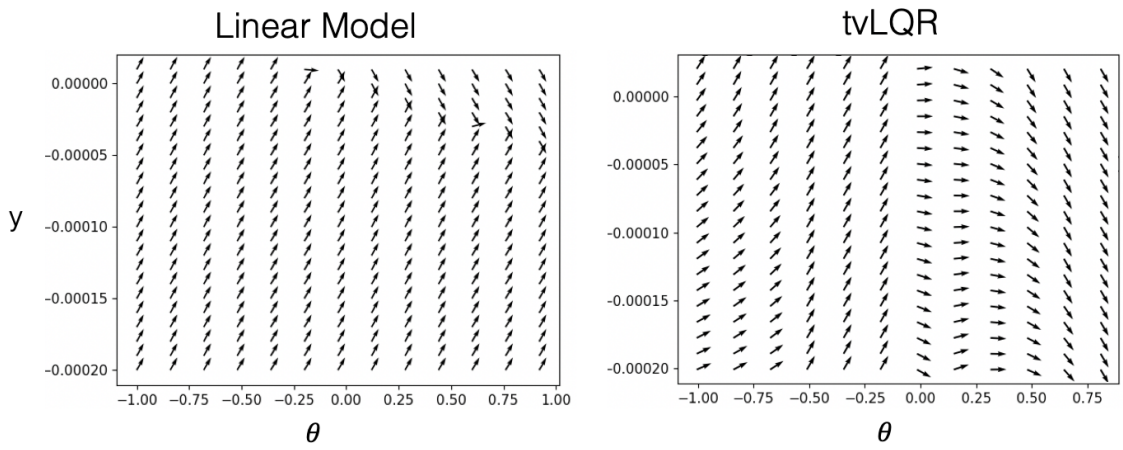


Figure 3-8: **Comparison of commanded pulling angle in the world frame and with fixed $\alpha$ using different controllers**.

The tvLQR controller consistently traveled to the end of the workspace while moving too far inwards in every trial. This could be fine for certain tasks or clothing types, especially narrow fabrics like a scarf or a tie where the distance traveled while sliding is more important than pose estimation. When the cloth edge was angled, it would quickly either slip out or move too far inward. LQR based on the linear model was less sensitive to angled starting positions.

The tvLQR network with a linear kernel for the $\dot{y}$ model had similar commanded actions to LQR based on the linear model, however it runs slower due to the time required to calculate the predictive gradient for the GP. Therefore, we found the linear model was favorable.

## 3.3 Conclusion

We found that our setup allowed the gripper to slide along the edge of the towel for an average of 40.0 cm with the LQR controller based on the linear model and 38.2 cm for the tvLQR controller based on the GP model. The tvLQR controller did not have a significant advantage over the other controller, but could potentially be improved with more training data for the GP. The larger problem we found, however, was that no kernel was able to predict $\dot{y}$ better than the linear kernel with the given input variables. A different formulation of state variables may prove useful. To improve

edge pose estimation in the future, we could use temporal information to improve the network performance. The fabric edge moves smoothly, so we could use filtering, motion-based computer vision techniques, or a recurrent neural network to leverage this knowledge. We could also add a gripping controller like in She et al. (2020) to grip harder when we are uncertain about the fabric edge position [45]. Another possibility is to add another degree of freedom to the gripper to "pinch" in order to change the location of the convexity, or the most sensitive region of the gripper.

We believe that the key to implementing our method of cloth edge following for future tasks with long travel lengths is a robust regrasp. It is worth noting that many dressing tasks only require sliding short distances. We also found that other cloth following strategies that involve keeping the cloth in a stable configuration while following the contour of the edge were much easier to implement, although potentially limiting for future tasks.

# Chapter 4

# Conclusions

In this work, we present a perception and control framework to tackle the task of slide along a cable and the edge of a piece of fabric. The tight integration of tactile feedback is key to turning the–a priori–complex task of manipulating a highly deformable object with uncontrolled variations in friction and shape, into an achievable task. We show that the LQR controller based on the linear dynamics model performs comparably or better than controllers based on a more complex, nonlinear GP model. The tvLQR controllers linearizing the GP at the current operating perform similarly to the original LQR controllers, while the controller that linearized the GP dynamics model about the origin and the MPC controller using the full GP both performed considerably worse for the cable following task.

Extending the cable following framework to cloth was challenging given the flatness complicating tactile pose detection and the weight of the cloth causing it to constantly slip from the grip. We used PCA on the depth image with cables, but trained a neural network to classify tactile images and determine the edge position, if visible. We also adjusted the experimental setup between cable and fabric edge following to make the task more feasible. One major limitation of the fabric following system compared to the cable following one is that the network is trained on a specific type of fabric. We will need a much larger training set with a variety of fabrics in order to generalize the perception system, even though we were able to perform cable following on a variety of fabrics given the gripping controller [45].

## 4.1 Discussions

Robotic manipulation has had an impact on a range of real-world tasks, such as pick-and-place and assembly. In most cases, the objects manipulated are rigid. Manipulation of deformable objects is more challenging since soft materials are represented by more complex states and follow more complex dynamics. A common approach to manipulating deformable objects is to iteratively transition between static stable states via pick-and-release sequences. This reduces complexity, but also makes manipulation inefficient. Natural manipulation of deformable objects observed from humans involves dynamic interactions such as sliding along an earbud cable to find the plug or sliding along the edge of a sheet to find its corner.

This “sliding" motion yields new challenges: cables are highly deformable with complex dynamics, and the operation requires real-time adjustments. Correspondingly, the design of control policies for this type of task becomes difficult. However, we can exploit the the local constraints imposed by the same sliding motion to simplify control, specially when supported by state feedback via advanced tactile sensors.

### 4.1.1 Global vs. Local Dynamics

From a global perspective, the state and dynamics of a deformable object are computationally challenging due to their large number of DOFs. Fig. 4-1 (left) shows the global view of an earphone cable and a piece of cloth. However, the state and dynamics are simplified for specific types of interaction. For example when the cable or piece of cloth are in tension, it is easier to control the task of sliding the fingers. The dimension of the deformable object is reduced by the same constraints imposed by the sliding motion. This makes it possible to design simple and efficient controllers for real-time robotic manipulation. In Fig. 4-1 (Right) the state of cable and the cloth is simplified in between the 2 grasping points.

The technique we demonstrate in this work bypasses the complexities of global state estimation and control by designing policies that rely only on local state feedback, which can be captured by tactile sensors. The technique to change the grasp

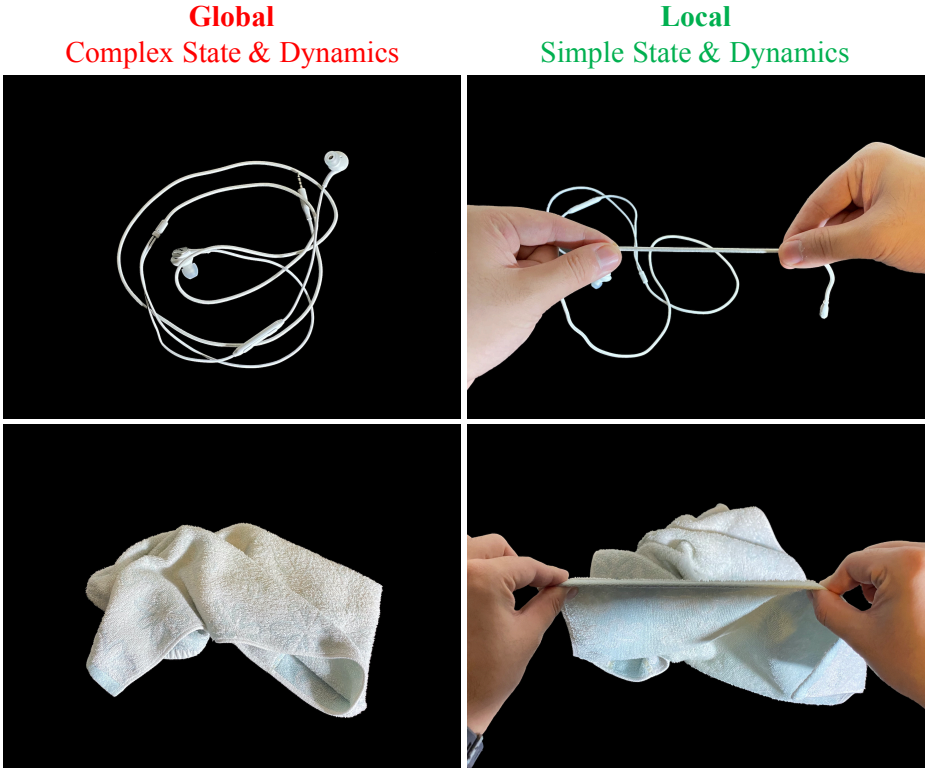


Figure 4-1: **Simplified state and dynamics from sliding.** The comparison of the global (left) and local (right) perspective of manipulating a piece of cable and fabric. From a global view, it is challenging to model the state and dynamics of a deformable objects due to the large number of degrees of freedom. However, the sliding motion adds constraints to the objects, simplifying the local state and dynamics. It enables fast and reactive manipulation skills.

on the cable by sliding the fingers can be thought of as a closed-loop primitive action (the "sliding regrasp") applicable to a range of objects (not just cables, but also rigid objects and other types of deformable objects, e.g., cloth). On the perception side, this requires a sensor that can track local motions of the local geometry at contact. On the control side, this requires a model of the local pulling-sliding dynamics.

## 4.2 Future Work

Beyond improving the perception and control systems with more training data, we are interested in pursuing research avenues that build upon this sliding skill. For example, sliding can enable finding a feature on a deformable object or tensioning a specific length of the object, so we can be confident of the state between the two grippers. We are also interested in using extrinsic dexterity such as gravity, dynamics, and environmental features because these are crucial tools people use every day to manipulate deformable objects. Additionally, we are working on integrating vision and tactile sensing for more robust and capable perception. We would eventually like to build a library of dextrous manipulation skills for deformable objects exploiting tactile perception. Using a task and motion planning (TAMP), we are interested in tasks including cable routing, knot tying, and folding, and dressing.

# Appendix A

# Tables

Table A.1: $R^2$ values of models trained on 10% of cable following dataset.

| **Kernel** | $\dot{y}$ $R^2$ | $\dot{\theta}$ $R^2$ | $\dot{\alpha}$ $R^2$ |
|---|---|---|---|
| RBF | 0.792 | 0.694 | 0.725 |
| RBF, ARD | 0.873 | 0.861 | 0.752 |
| Rational Quadratic | 0.847 | 0.794 | 0.790 |
| Rational Quadratic, ARD | 0.899 | 0.901 | 0.793 |
| Matern-3/2 | 0.808 | 0.761 | 0.772 |
| Matern-3/2, ARD | 0.895 | 0.894 | 0.828 |
| Matern-5/2 | 0.808 | 0.727 | 0.765 |
| Matern-5/2, ARD | 0.891 | 0.876 | 0.767 |
| Exponential | 0.845 | 0.791 | 0.79 |
| Exponential, ARD | 0.896 | 0.897 | 0.859 |
| Linear | 0.707 | 0.556 | 0.592 |

Table A.2: $R^2$ values of models trained on 40% of cable following dataset.

| **Kernel** | $\dot{y}$ $R^2$ | $\dot{\theta}$ $R^2$ | $\dot{\alpha}$ $R^2$ |
|---|---|---|---|
| RBF | 0.969 | 0.964 | 0.835 |
| Matern-5/2 | 0.973 | 0.979 | 0.835 |
| Exponential | 0.966 | 0.977 | 0.954 |

# Appendix B

# Figures

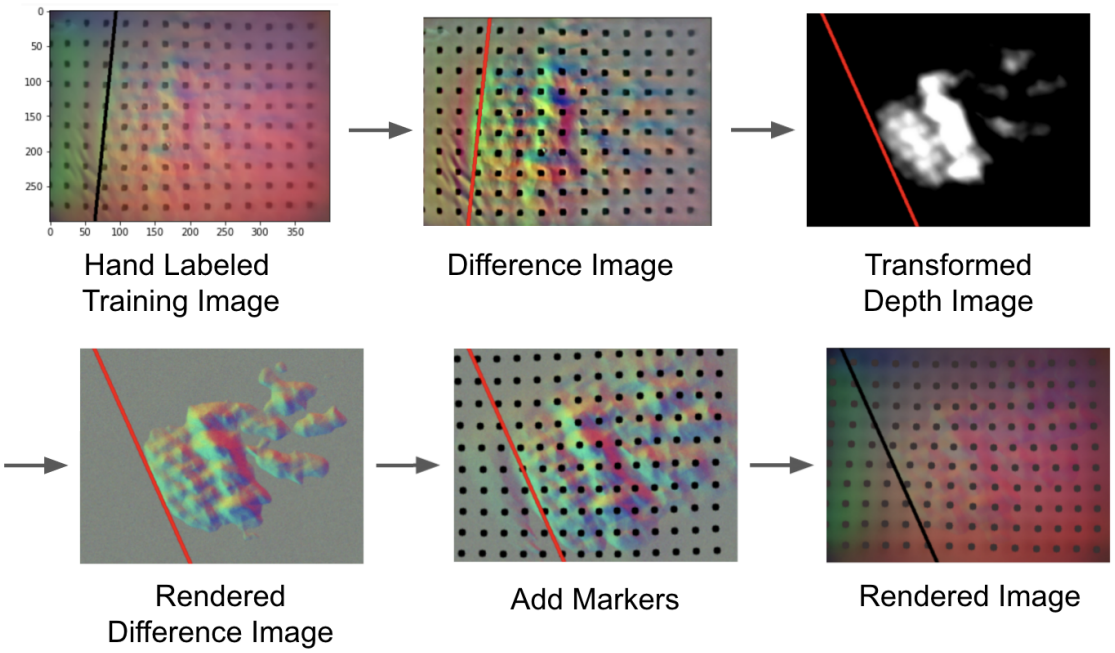


Figure B-1: **Rendered image dataset generation.** Initially, our fabric edge detection network was trained on rendered tactile images from the augmented depth images so we would not have to compute the depth image in real time. To do this, we first used the same process explained in Chapter 2 to transform and augment a depth image. Then, we rendered a difference image from the depth image, added markers warped using a thin-plate spline, and and added the initial frame to the difference image.

# Bibliography


[1] Gpy: The gaussian processes framework in python, since 2012.

[2] F. Abegg, A. Remde, and D. Henrich. Force- and Vision-Based Detection of Contact State Transitions. 2000.

[3] Dmitry Berenson. Manipulation of deformable objects without modeling and simulating deformation. In *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[4] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Transactions on Graphics*, 2008.

[5] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

[6] Ning Chen, Hong Zhang, and R Rink. Edge tracking using tactile servo. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 2, pages 84–89. IEEE, 1995.

[7] Cheng Chi and Dmitry Berenson. Occlusion-robust deformable object tracking without physics simulation. In *Intelligent Robots and Systems (IROS), 2019 IEEE International Conference on*. IEEE, 2019.

[8] Siyuan Dong, Daolin Ma, Elliott Donlon, and Alberto Rodriguez. https://arxiv.org/abs/1810.13381Maintaining Grasps within Slipping Bound by Monitoring Incipient Slip. In *IEEE ICRA*, 2018.

[9] Siyuan Dong and Alberto Rodriguez. Tactile-based insertion for dense box-packing. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[10] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2015.

[11] Aditya Ganapathi, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Jennifer Grannen, Minho Hwang, Ryan Hoque, Joseph E. Gonzalez, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images. *arXiv*, 2020.

[12] Aditya Ganapathi, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Ryan Hoque, Joseph E. Gonzalez, and Ken Goldberg. MMGSD: Multi-Modal Gaussian Shape Descriptors for Correspondence Matching in 1D and 2D Deformable Objects. In *Intelligent Robots and Systems (IROS), 2020 IEEE International Conference on*, 2020.

[13] Jennifer Grannen, Priya Sundaresan, Brijen Thananjeyan, Jeff Ichnowski, Ashwin Balakrishna, Vainavi Viswanath, Michael Laskey, Joseph Gonzalez, and Ken Goldberg. https://goldberg.berkeley.edu/pubs/CoRL-2020-Untangling-submitted.pdfLearning Robot Policies for Untangling Dense Knots in Linear Deformable Structures. In *4th Conference on Robot Learning (CoRL)*, 2020.

[14] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. *arXiv preprint arXiv:2105.03655*, 2021.

[15] C. Harris and M. Stephens. A Combined Corner and Edge Detector. 2013.

[16] Randall B Hellman, Cem Tekin, Mihaela van der Schaar, and Veronica J Santos. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8039205Functional contour-following via haptic perception and reinforcement learning. *IEEE transactions on haptics*, 11(1):61–72, 2017.

[17] Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[18] Francois R Hogan, Maria Bauza, Oleguer Canal, Elliott Donlon, and Alberto Rodriguez. Tactile regrasp: Grasp adjustments via simulated tactile transformations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2963–2970. IEEE, 2018.

[19] John E Hopcroft, Joseph K Kearney, and Dean B Krafft. https://journals.sagepub.com/doi/pdf/10.1177/027836499101000105A case study of flexible object manipulation. *The International Journal of Robotics Research*, 10(1):41–50, 1991.

[20] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Kumar Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. VisuoSpatial foresight for multi-step, multi-task fabric manipulation, 2020.

[21] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. Tracking objects with point clouds from vision and touch. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4000–4007. IEEE, 2017.

[22] Shervin Javdani, Sameep Tandon, Jie Tang, James O'Brien, and Pieter Abbeel. https://ieeexplore.ieee.org/document/5980431Modeling and perception of deformable one-dimensional objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.

[23] Xin Jiang, Yuki Nagaoka, Kazushi Ishii, Satoko Abiko, Teppei Tsujita, and Masaru Uchiyama. https://www.sciencedirect.com/science/article/pii/S0736584514001069 Robotized recognition of a wire harness utilizing tracing operation. *Robotics and Computer-Integrated Manufacturing*, 34:52–61, 2015.

[24] Lydia E. Kavraki, Petr Švestka, Jean Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 1996.

[25] Naveen Kuppuswamy, Alex Alspach, Avinash Uttamchandani, Sam Creasey, Takuya Ikeda, and Russ Tedrake. Soft-bubble grippers for robust and perceptive manipulation. *arXiv preprint arXiv:2004.03691*, 2020.

[26] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.

[27] Friedrich Lange, Patrick Wunsch, and Gerhard Hirzinger. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=680743Predictive Vision Based Control of High Speed Industrial Robot Paths. In *1998 IEEE/RSJ International Conference on Robotics and Automation (ICRA)*. IEEE, 1998.

[28] S M LaValle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *In*, 1998.

[29] Nathan F Lepora, Alex Church, Conrad De Kerckhove, Raia Hadsell, and John Lloyd. From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 4(2):2101–2107, 2019.

[30] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3988–3993. IEEE, 2014.

[31] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[32] Wen Hao Lui and Ashutosh Saxena. Tangled: Learning to untangle ropes with rgb-d perception. In *Intelligent Robots and Systems (IROS), 2013 IEEE International Conference on*, pages 837–844. IEEE, 2013.

[33] Hermann Mayer, Faustino Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jürgen Schmidhuber. http://people.idsia.ch/ tino/papers/mayer.iros06.pdfA system for robotic heart surgery that learns to tie knots using recurrent neural networks. *Advanced Robotics*, 22(13-14):1521–1537, 2008.

[34] Dale McConachie, Andrew Dobson, Mengyao Ruan, and Dmitry Berenson. Manipulating deformable objects by interleaving prediction, planning, and control. *International Journal of Robotics Research*, 2020.

[35] Mark Moll and Lydia Kavraki. Path planning for deformable linear objects. *IEEE Transactions on Robotics*, pages 625–636, 2006.

[36] Takuma Morita, Jun Takamatsu, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1242193Knot planning from observation. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 3, pages 3887–3892. IEEE, 2003.

[37] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. https://arxiv.org/abs/1703.02018 Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2146–2153. IEEE, 2017.

[38] Antoine Petit, Vincenzo Lippiello, and Bruno Siciliano. Real-time tracking of 3d elastic objects with an rgb-d sensor. page 3914–3921, 2015.

[39] Jianing Qian, Thomas Weng, Luxin Zhang, Brian Okorn, and David Held. Cloth region segmentation for robust grasp selection. In *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[40] P A Review, Gaussian Random, and Correlation Functions Tech. C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. c 2006 Massachusetts Institute of Technology. www.GaussianProcess.org/gpml. *Neural Networks*, 2006.

[41] Mitul Saha, Pekka Isto, and Jean-Claude Latombe. http://ai.stanford.edu/ latombe/papers/iser-06/paper.pdfMotion planning for robotic manipulation of deformable linear objects. *Experimental Robotics*, page 23–32, 2008.

[42] Khairul Salleh Mohamed Sahari, Hiroaki Seki, Yoshitsugu Kamiya, and Masatoshi Hikizu. Edge tracing manipulation of clothes based on different gripper types. *Journal of Computer Science*, 2010.

[43] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Learning from demonstrations through the use of non-rigid registration. In *Springer Tracts in Advanced Robotics*, 2016.

[44] Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Kumar Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, Katsu Yamane, Soshi Iba, John Canny, and Ken Goldberg. Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor. In *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[45] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *arXiv preprint arXiv:1910.02860*, 2019.

[46] Li Sun, Gerarado Aragon-Camarasa, Paul Cockshott, Simon Rogers, and J. Paul Siebert. A heuristic-based approach for flattening wrinkled clothes. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[47] Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Michael Laskey, Kevin Stone, Joseph E. Gonzalez, and Ken Goldberg. Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2020.

[48] Te Tang, Yongxiang Fan, Hsien-Chung Lin, and Masayoshi Tomizuka. https://ieeexplore.ieee.org/document/8206058State estimation for deformable objects by point registration and dynamic simulation. In *Intelligent Robots and Systems (IROS), 2017 IEEE International Conference on*. IEEE, 2017.

[49] Stephen Tian, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. Manipulation by feel: Touch-based control with deep predictive models. *arXiv preprint arXiv:1903.04128*, 2019.

[50] Angelina Wang, Thanard Kurutach, Kara Liu, Pieter Abbeel, and Aviv Tamar. Learning robotic manipulation through visual planning and acting. In *Robotics: Science and Systems (RSS)*, 2019.

[51] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. https://ras.papercept.net/proceedings/IROS20/1576.pdfSwingBot: Learning Physical Features from In-hand Tactile Exploration for Dynamic Swing-up Manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[52] Fei Wang, Etienne Burdet, Ronald Vuillemin, and Hannes Bleuler. Knot-tying with visual and force feedback for vr laparoscopic training. In *Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2005.

[53] Weifu Wang, Dmitry Berenson, and Devin Balkcom. An online method for tight-tolerance insertion tasks for string and rope. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2488–2495. IEEE, 2015.

[54] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. https://www.liebertpub.com/doi/pdfplus/10.1089/soro.2017.0052The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics*, 5(2):216–227, 2018.

[55] Bryan Willimon, Stan Birchfield, and Ian Walker. Model for unfolding laundry using interactive perception. In *IEEE International Conference on Intelligent Robots and Systems*, 2011.

[56] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations, 2019.

[57] Akihiko Yamaguchi and Christopher G Atkeson. Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1045–1051. IEEE, 2016.

[58] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm. In *Intelligent Robots and Systems (IROS), 2010 IEEE International Conference on*, pages 49–54. IEEE, 2010.

[59] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. Simple model and deformation control of a flexible rope using constant, high-speed motion of a robot arm. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2012.

[60] Yuji Yamakawa, Akio Namiki, Masatoshi Ishikawa, and Makoto Shimojo. https://ieeexplore.ieee.org/abstract/document/4399379One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007.

[61] Mengyuan Yan, Yilin Zhu, Ning Jin, and Jeannette Bohg. https://arxiv.org/pdf/1911.06283.pdf Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects. *arXiv preprint arXiv:1911.06283*, 2019.

[62] Wenzhen Yuan, Siyuan Dong, and Edward Adelson. https://www.mdpi.com/1424-8220/17/12/2762/htmGelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[63] Hiroyuki Yuba, Solvi Arnold, and Kimitoshi Yamazaki. Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty. *Advanced Robotics*, 2017.

[64] Shigang Yue and Dominik Henrich. Manipulating deformable linear objects: Sensor-based fast manipulation during vibration. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2002.

[65] Harry Zhang, Jeffrey Ichnowski, Daniel Seita, Jonathan Wang, and Ken Goldberg. https://arxiv.org/pdf/2011.04840.pdfRobots of the Lost Arc: Learning to Dynamically Manipulate Fixed-Endpoint Ropes and Cables. *arXiv preprint arXiv:2011.04840*, 2010.

[66] Jihong Zhu, Benjamin Navarro, Philippe Fraisse, André Crosnier, and Andrea Cherubini. https://ieeexplore.ieee.org/document/8593780 Dual-arm robotic manipulation of flexible cables. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 479–484. IEEE, 2018.

[67] Jihong Zhu, Benjamin Navarro, Robin Passama, Philippe Fraisse, André Crosnier, and Andrea Cherubini. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8851170Robotic Manipulation Planning for Shaping Deformable Linear Objects With Environmental Contacts. *IEEE Robotics and Automation Letters*, 5(1):16–23, 2019.