UNIVERSITY OF CALIFORNIA

Los Angeles

Data-driven Robotic Manipulation of Deformable Objects Using Tactile Feedback: From Model-free to Model-based Approaches

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mechanical Engineering

by

Yi Zheng

2023

ABSTRACT OF THE DISSERTATION

Data-driven Robotic Manipulation of Deformable Objects Using Tactile Feedback:
From Model-free to Model-based Approaches

by

Yi Zheng
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2023
Professor Veronica Santos, Chair

Perceiving and manipulating deformable objects with the sense of touch are essential skills in everyday life. However, it remains difficult for robots to autonomously manipulate deformable objects using tactile sensing because of numerous perception, modeling, planning, and control challenges. We believe this is partially due to two fundamental challenges: (1) Establishing a physics-based model describing physical interactions between deformable tactile sensors and deformable objects is difficult; (2) Modern tactile sensors provide high-dimensional data, which is beneficial for perception but impedes the development of practical planning and control strategies. To address these challenges, we developed systematic frameworks for the tactile-driven manipulation of deformable objects that integrates state-of-the-art tactile sensing with well-established tools used by other robotics communities.

In Study #1, we showed how a robot can learn to manipulate a deformable, thin-shell object via tactile sensor feedback using model-free reinforcement learning methods. A page flipping

task was learned on a real robot using a two-stage approach. First, we learned nominal page flipping trajectories by constructing a reward function that quantifies functional task performance from the perspective of tactile sensing. Second, we learned adapted trajectories using tactile-driven perceptual coupling, with an intuitive assumption that, while the functional page flipping trajectories for different task contexts (page sizes) might differ, similar tactile sensing feedback should be expected.

In Study #2, we showed how a robot can use tactile sensor feedback to control the pose and tension of a deformable linear object (elastic cable). For a cable manipulation task, low-dimensional latent space features were extracted from high-dimensional raw tactile sensor data using unsupervised learning methods, and a dynamics model was constructed in the latent space using supervised learning methods. The dynamics model was integrated with an optimization-based, model predictive controller for end-to-end, tactile-driven motion planning and control on a real robot.

In summary, we developed frameworks for the tactile-driven manipulation of deformable objects that either circumvents sensor modeling difficulties or constructs a dynamics model directly from tactile feedback and uses the model for planning and control. This work provides a foundation for the further development of systematic frameworks that can address complex, tactile-driven manipulation problems.

The dissertation of Yi Zheng is approved.

Ankur Mehta

Jacob Rosen

Dennis Hong

Veronica Santos, Committee Chair

University of California, Los Angeles

2023

To my beloved family . . .

My parents, Huichang and Yukun, who taught me

the importance of hard work, perseverance as well as optimism.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions 2

2 Autonomous Learning of Page Flipping Movements via Tactile Feedback **4**

2.1 Abstract 4

2.2 Introduction 5

2.3 Related Work 9

2.4 Learning to Manipulate a Thin-shell Object via Tactile Sensor Feedback 11

2.4.1 Dynamic Movement Primitives 12

2.4.2 Model-based Relative Entropy Policy Search 14

2.4.3 Relevance of Tactile Information to a Page Flipping Task 15

2.4.4 Learning a Nominal Trajectory from Tactile Feedback 18

2.4.5 Adapting Learned Nominal Trajectories to a Novel Context 20

2.4.6 Representation of the Tactile Sensing Traces 22

2.5 Experimental Procedure and Evaluation 24

2.5.1 Experimental Set-up 25

2.5.2 Learning a Nominal Trajectory from Tactile Feedback 26

2.5.3 Impact of Chosen Representation of Tactile Sensing Traces 32

2.5.4 Adapting Learned Nominal Trajectories to a Novel Context 38
2.6 Conclusion 44
2.6.1 Summary of Contributions 44
2.6.2 Limitations and Future Work 45

3 End-to-End Model Predictive Control Framework for Tactile-Driven Manipulation of Deformable Linear Objects 47

3.1 Introduction 48
3.2 Related Work 52
3.2.1 Model-based Reinforcement Learning 52
3.2.2 Robotic Manipulation of Deformable Linear Objects 54
3.2.3 Simulation of Tactile Sensor Behavior 56
3.3 Framework for the Tactile-driven Model Predictive Control of Deformable Linear Objects 58
3.3.1 Gaussian Process Models 60
3.3.2 Uncertainty Propagation 63
3.3.3 Tractable Model Predictive Control Formulation 64
3.3.4 Unsupervised Learning of Dynamics Using Probabilistic Generative Modeling 65
3.4 Experimental Procedure and Evaluation 74
3.4.1 Experimental Setup 76
3.4.2 Collecting and Augmenting Observation-Action Pairs for Training 78

3.4.3 Learning a Latent Space Dynamics Model from Tactile Sensor Data Using a Static Inference Model . . . 79
3.4.4 Learning a Latent Space Dynamics Model from Tactile Sensor Data Using a Sequential Inference Model . . . 88
3.4.5 Comparing the Performance of the Static and Sequential Inference Models 93
3.4.6 Performing Model Predictive Control Using a Learned Latent Space Dynamics Model . . . 95
3.5 Conclusion and Future Work . . . 101
3.6 Appendix . . . 103

4 Summary and Conclusion . . . 106
4.1 Contributions . . . 106
4.2 Future Work . . . 107
4.2.1 Extending Manipulation Tasks from 2D to 3D and Incorporating Mechanics Models . . . 107
4.2.2 Hybrid Framework for Simulation of Tactile Sensor Behavior . . . 108
4.2.3 Learning Policies from Multimodal Sensor Systems . . . 109

References . . . 109

LIST OF FIGURES

2.1 Three joints (J2, J4, J6) of a 7-DOF Kinova robot arm are controlled to perform page flipping movements in the y-z plane using two fingers outfitted with deformable, multimodal BioTac tactile sensors. Large and small notebook pages (shown) were held by rigid binders placed on a flat support surface parallel to the x-y plane. Passive motion capture markers are attached to the binders for tracking displacement. 6

2.2 Representative snapshots of the three categories of page flipping trajectories described in Section 2.4.3 are shown in 15 sec increments and the supplemental video. *(Top)* The Semi-circular trajectory represents the ideal page flipping movement. *(Middle)* For the Warping trajectory, page warping is especially pronounced in subfigure *(f)* and page snapping results in the page configuration shown in subfigure *(g)*. *(Bottom)* For the Aggressive trajectory, the binder is pulled aggressively from subfigures *(d)* through *(g)*, resulting in large and numerous displacements of the binder in the y-z plane defined in Figure 3.1. 16

2.3 Representative tactile sensing traces from the artificial apical tuft (electrodes 7, 8, 9, 10 in Fig. 2.4) are shown in arbitrary units (AU) for the *(a)* top and *(b)* bottom fingers during the flipping of small pages when using the three categories of trajectories (15 rollouts each) described in Section 2.4.3. The semi-circular trajectories (green) generate smoother tactile signals than both warping trajectories (blue) and aggressive trajectories (red). As indicated by the periods shaded in gray, spikes in the tactile signals occur near the end of the page flipping movement for the warping trajectories and near the beginning and end of the movement for the aggressive trajectories. 17

2.4 *(a)* The multimodal BioTac sensor is comprised of a rigid core, elastomeric skin, and fingernail (Image from [28]). *(b)* The rigid core of the BioTac is shown with impedance electrodes individually numbered. The red ellipse highlights the artificial apical tuft (flat region of the distal phalanx), where contact is made with the page in the majority of cases. 22

2.5 Learning curves are shown for learning nominal trajectories for flipping *(a)* small pages over the course of 25 policy updates and *(b)* large pages over the course of 41 policy updates. Mean and variance are presented for batches of 10 rollouts. 25

2.6 Individual reward components from Equation 2.15 are shown for the nominal trajectory learning curves in Figure 2.5 for flipping *(a)* small pages over the course of 25 policy updates and *(b)* large pages over the course of 41 policy updates. Mean and variance are presented for batches of 10 rollouts. This figure illustrates that the improvement of tactile-related reward components plays a major role in the improvement of the overall reward function as compared to any losses resulting from binder displacement. 26

2.7 Data for each of 19 electrodes are shown for the bottom finger for learning a nominal trajectory for flipping a large page. A total of 15 rollouts are shown for each the initial policy after a single policy update (red) and final policy after 41 policy updates (green). The initial policy generates undesired spikes in the electrode signals during the page warping period shaded in gray. 27

2.8 Low frequency pressure data are shown for the *(a)* top and *(b)* bottom fingers for learning a nominal trajectory for flipping a large page. A total of 15 rollouts are shown for each the initial policy after a single policy update (red) and final policy after 41 policy updates (green). The initial policy generates undesired spikes in the low frequency pressure data during the page warping period shaded in gray. 31

2.9 Distributions of reward function samples are shown for two representations of tactile sensing traces: *(a)* artificial apical tuft, and *(b)* PCA eigenvalues. Fifteen rollouts were performed for each of six trajectories on a binder containing small pages: an ideal semi-circular trajectory (*Functional*), three trajectories that cause page warping and snapping, and two aggressive trajectories. The *Warped_1* trajectory is the nominal trajectory learned for large pages, but purposely applied to small pages. 34

2.10 Learning curves are shown for the simplified experiment on adaptation of the learned nominal trajectories to a novel context using the *(a)* artificial apical tuft and *(b)* PCA eigenvalue representations of tactile sensing traces. The goal position for small pages is known a priori. Mean and variance are presented for batches of 10 rollouts. 36

2.11 Distributions of tactile sensing traces are shown for the simplified experiment on adaptation of the learned nominal trajectories to a novel context using the *(a)* artificial apical tuft and *(b)* PCA eigenvalue representations of tactile sensing traces. The goal position for small pages is known a priori. Tactile data are shown for a nominal trajectory learned for large pages and applied to large pages (green) and to small pages prior to adaptation learning (red), for 10 rollouts each. Three independent learning trials (9 policy updates each) show how the tactile sensing traces change as the initial nominal trajectory for large pages is adapted to small pages during adaptation learning. 37

2.12 Learning curves are shown for the full experiment on adaptation of the learned nominal trajectories to a novel context using the PCA eigenvalue representation of tactile sensing traces. The goal position for small pages is not known a priori and must be learned. Mean and variance are presented for batches of 10 rollouts. 39

2.13 Distributions are shown for the PCA eigenvalue representation of tactile sensing traces for the full experiment on adaptation of the learned nominal trajectories to a novel context. The goal position for small pages is not known a priori and must be learned. Distributions (mean and variance) are shown for a nominal trajectory learned for large pages and applied to large pages (blue) and to small pages prior to adaptation learning (red), for 10 rollouts each. Three independent learning trials (16 policy updates each) show how the distributions change as the initial nominal trajectory for large pages is adapted for small pages after learning. The distribution of tactile feedback is also shown for the ideal case of a nominal trajectory learned for small pages and applied to small pages (green). 40

2.14 Distributions are shown for reward function samples using the PCA eigenvalue representation of tactile sensing traces for the full experiment on adaptation of the learned nominal trajectories to a novel context. Goal position is not known a priori and must be learned. Distributions are shown for a nominal trajectory learned for large pages and applied to small pages prior to adaptation learning (red), for 10 rollouts each. Three independent learning trials (16 policy updates each) show how the distributions change as the initial nominal trajectory for large pages is adapted for small pages after learning. The distribution of reward function samples is also shown for the ideal case of a nominal trajectory learned for small pages and applied to small pages (green). . . . 43

3.1 A 7-DOF robot arm controls the pose angle and tensile state of an elastic cable using a gripper outfitted with the vision-based GelSight tactile sensor [112]. One end of the tethered cable is manipulated by the gripper through a rigid rig. The left and right columns show different cable states and their tactile representations, respectively. . . . 49

3.2 Two different types of inference models were considered for capturing the latent space dynamics of the observation-action pairs. (L) The static inference model (section 3.3.4.1) assumes that each transition tuple is i.i.d. Representative tactile sensor data are shown as outputs of the generative process. (R) The sequential inference model (section 3.3.4.2) considers temporal relationships within the tactile sensor data. Solid and dashed arrows represent generative processes and inference functions, respectively. Grey and white shading denote whether the variables are observed or unknown, respectively. Squares and circles indicate whether variables are treated as deterministic or stochastic, respectively. 68

3.3 Experimental procedure: (Left) We collected tactile observation – robot action pairs for use as training data. (Center) System dynamics were mapped from a high-dimensional tactile sensor data space to a low-dimensional latent space. (Right) The learned latent space dynamics model was integrated with an optimization-based model predictive control framework. 75

3.4 Experiment setup: The robot gripper moves within a 2D plane defined by the **x**- and **y**-axes. The top finger of the gripper is outfitted with a camera-based tactile sensor (GelSight). The configuration of the elastic cable is specified as the angle α between the cable and the **x**-axis, and the distance **d** between the grasping point and the anchor point. The Aruco marker is used solely for visualization within the latent space and is not used for training. 77

3.5 The training and validation loss curves are shown for three random seeds for the GP-based static inference model of latent space dynamics. From left to right, the first three plots show the training loss for the encoder-decoder reconstruction loss, GP latent space transition likelihood, and negative entropy encoder regularization, respectively (terms **I**, **II**, and **III** in Eqn. Equation Equation (3.24)). The plot on the far right shows the validation loss defined in Section 3.4.3. 79

3.6 For the GP-based static inference model of latent space dynamics, the original GelSight marker displacement vector fields (Row 1) are shown alongside the tactile sequences reconstructed from the learned dynamics model at epochs 2, 18, and 30 (Rows 2, 3, and 4, respectively). The solid arrows in Rows 2-4 are reconstructed by sending the GP predictive mean to the decoder network. The shaded arrows are reconstructed by sending 20 predictions sampled from the GP predictive distribution to the decoder network. The data shown are for the worst performing trial (purple trace) in Figure 3.5. 82

3.7 (Row 1) The latent space structure is shown, parameterized by α and $\mathbf{d}$, after mapping all observations from the validation dataset through the trained encoder network at epochs 2, 18, and 30 for the GP-based static inference model of latent space dynamics. The colormaps for α and $\mathbf{d}$ are based on data from the Aruco marker shown in Figure 3.4. (Row 2) All reconstructed and original sequences of validation data, as encoded in the latent space, are shown at epochs 2, 18, and 30. The start and end of an individual trajectory (purple trace in Figure 3.5) are highlighted with a red cross and blue triangle, respectively. 85

3.8 The training and validation loss curves are shown for the GP-based sequential inference model of latent space dynamics. For brevity, the recurrent state space model was trained only once using 26 epochs. The SVGP was then trained with the latent state encoded by the inference model **q**. The plot on the left shows the reconstruction loss (red) and KL divergence regularization loss (blue) (terms **I** and **II** from Eqn. Equation Equation (3.29)). The plots in the center and on the right show the SVGP negative log likelihood and validation loss, respectively, for the same three random seeds used to evaluate the GP-specific static inference model. 87

3.9 For the GP-based sequential inference model of latent space dynamics, the original GelSight marker displacement vector fields (Row 1) are shown alongside the tactile sequences reconstructed from the learned dynamics model at epochs 2, 38, and 40 (Rows 2, 3, and 4, respectively). The solid arrows in Rows 2-4 are reconstructed by sending the GP predictive mean to the decoder network. The shaded arrows are reconstructed by sending 20 predictions sampled from the GP predictive distribution to the decoder network. The data shown are for the red trace from the center and right plots in Figure 3.8. 90

3.10 (Row 1) The latent space structure is shown, parameterized by α and **d**, after mapping all observations from the validation dataset through the trained recurrent state space model at epochs 2, 38, and 40 for the GP-based sequential inference model of latent space dynamics. The colormaps for α and **d** are based on data from the Aruco marker shown in Figure 3.4. (Row 2) All reconstructed and original sequences of validation data, as encoded in the latent space, are shown at epochs 2, 38, and 40. The start and end of an individual trajectory (red trace from the center and right plots in Figure 3.8) are highlighted with a red cross and blue triangle, respectively. 91

3.11 The distribution of differences between the original and reconstructed latent space trajectories are shown on a normalized scale to enable the direct comparison of the GP-specific static and sequential inference models. Differences for 50k sample sequences are shown for prediction windows of 30, 40, and 50 datapoints. For each plot, the blue line denotes the mean while the bottom and top bounds of the box denote the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme datapoints that are not considered as outliers. 94

3.12 Snapshots are shown of the real robot (Rows 1 and 3) and corresponding the latent space (Rows 2 and 4) for each of two manipulation experiments. The first trial required 9 MPC iterations; snapshots are shown for MPC iterations 1, 3, 5, 7 and 9. The second trial required 15 MPC iterations; snapshots are shown for MPC iterations 1, 4, 7, 10, 13, and 15. The black and red circles represent the initial and goal states, respectively, in the latent space after being encoded from the corresponding GelSight marker displace vector fields. The blue solid lines and ellipsoids indicate the predicted trajectory mean and uncertainty, respectively, at each planning step as outputted by the MPC solver. 96

3.13 The distribution of differences between the starting point for each MPC iteration and the predicted end point of the prior MPC iteration are shown for 15 repetitions of each of the two manipulation experiments shown in Figure 3.12. For each plot, the red line denotes the mean while the bottom and top bounds of the box denote the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme datapoints that are not considered as outliers. 99

3.14 The detailed network structure is shown for the sequential inference model described in Section 3.4.4 and on the right of Figure 3.2. The top two networks on the left describe the inference model. The top two networks on the right describe the generative process. The recurrent neural networks on the bottom left connect the inference model and generative process. 105

LIST OF TABLES

2.1 Binder displacement distances are reported as mean (standard deviation) from 15 roll-outs for each combination of page size and learning stage. “Before learning” refers to the initial policy after a single policy update. “After learning” refers to the final policy after 25 and 41 policy updates for the small and large pages, respectively. 30

3.1 Encoder architecture for learning latent space dynamics using a static inference model. 104

3.2 Decoder architecture for learning latent space dynamics using a static inference model. 104

ACKNOWLEDGMENTS

Firstly, I would like to express my deep gratitude towards my advisor, Dr. Veronica J. Santos, for all of her support through out my whole PhD life. When I first attended the lab meeting of the Biomechatronics lab, Dr. Santos had just moved to UCLA and the lab was at the very beginning stage of development. I am very happy and proud that after all these years I made substantial contributions to fulfill the lab's mission, which is to bring the sense of touch into robotics to do cool stuff, with the support from Dr. Santos. Before I came to UCLA, I barely knew anything about robotics, and I was very thankful that Dr. Santos gave me the chance, freedom and time to explore interesting topics in robotics.

I would also like to thank my doctoral committee members Professor Ankur Mehta, Professor Jacob Rosen, and Professor Dennis Hong, for their guidance and time. My research has greatly benefited from their thought-provoking presentations, inspiring lectures, and collaborative research discussions.

I want to say thanks to everyone who were either a part of the Biomechatronics Laboratory or members from other robotics labs in the department for helping me both with my academic career but also for being there for me as a friend. Specifically, I would like to thank Dr. Xiaoyu Wang, Dr. Shengxin Jia, Dr. Eunsuk Chong, Dr. Eric Peltola, Dr. Lionel Zhang, Huajing Zhao from my lab, Dr. Junjie Shen from RoMeLa, and Dr. Daisy Yayun Du from the Structures-Computer Interaction Lab. I am sure that I could not have made it through without all of your support.

For Chapter 2, I would like to express my deep gratitude to Prof. Jan Peters at Technical University of Darmstadt and Prof. Filipe Veiga at George Mason University. Prof. Jan Peters leads the Intelligent Systems Laboratory at Technical University of Darmstadt, which in my opinion is one of the best robotics labs in the world. With the help of Prof. Santos I am very fortunate

to visit this lab for a year, it is at this lab I systematically studied the theory of reinforcement learning and their application in real world robotics, and developed the idea for a project that eventually leads to my first publication. Prof. Filipe Veiga was a doctoral student in the lab then and he was my mentor during my visit there, he helped me in every aspect from finding housing to formalizing research ideas. After I went back to UCLA I still keep working and having discussions with Prof. Filipe Veiga about my research project, from experiment data analysis all the way to manuscript writing. He offered so many insightful advices that not only helps me finish my first research project, but also benefits and inspires greatly my future PhD path as well as the mindset of how to be a scientific researcher. I would also like to say my thanks to other members of the Intelligent Systems Laboratory and other friends I met during my visit there: Dr. Boris Belousov, Dr. Hany Abdulsamad, Dr. Dorothy Koert, Dr. Fabio Muratore etc., from the Intelligent Systems Laboratory, and Dr. Zhenghao Wu now at XJTLU, Dr. Maohua Zhang now at The Pennsylvania State University, Yunlong Song now at ETH and Tianming Qiu now at TUM, with whom I had many insightful, beneficial and inspiring discussions not only just about research, but also about life. Additionally, we thank Casey Asaoka for assistance with the experimental setup. This work was supported in part by Office of Naval Research Award #N00014-18-1-2814.

For Chapter 3, I would like to give my special thanks to Prof. She Yu at Purdue, Dr. Shaoxiong Wang and Dr. Siyuan Dong at MIT. Prof. She Yu and Dr. Shaoxiong Wang designed the robotic gripper and vision-based tactile sensor used in the project, without which the accomplishment of this project would be impossible. And I had many insightful discussions with them which inspired me to ask the right question and find the right recipe for this project. I would also like to thank Dr. Xiaoyu Wang from my lab and Dr. WenZhong Yan from the lab of Professor of Ankur Mehta, they gave me invaluable advices regarding the proper experiment setup with which I can work out a meaningful problem with a meaningful framework. This work was supported in part by Office

of Naval Research Award #N00014-18-1-2814.

Finally and most importantly I would like to thank my friends and family. There has never been a single moment during my PhD life where I did not receive the support and love from my family. There are times during a PhD life where it is hard to see the light at the end of tunnel, and I just want to say thanks to my family for accompanying me all the way to the end. I have made friends with many people during my PhD life in UCLA, they inspire me to continue the passion for the robotics field but also they have always been there for me whenever there I need to go through hardships related to my PhD. I just want to say thank you for helping me get through the tunnel at the end.

VITA

2016–2023	Graduate Student Researcher, UCLA Biomechatronics Lab, Mechanical and Aerospace Engineering Department, UCLA, Los Angeles, California
2018–2019	Visiting Student Researcher, Technical University of Darmstadt, Hessen, Germany
2014–2016	M.S. Mechanical Engineering, UCLA, Los Angeles, California
2014	B.S. Reliability and Systems Engineering, Beihang University, Beijing, China

PUBLICATIONS

Zheng, Y., Veiga, F., Peters, J., & Santos, V. J. (2022). Autonomous learning of page flipping movements via tactile feedback IEEE Trans on Robotics, 2022:38(5):2734-2749, with podium presentation at the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyoto, Japan.

CHAPTER 1

Introduction

1.1 Motivation

The sense of touch is ubiquitous in the everyday life of human beings. People rely on the sense of touch to feel the world and explore their surroundings. When people use touch to perceive and manipulate an object, the contact between the hands and the object or the environment provides information about the interaction that can be leveraged to recognize objects or make decisions. In particular, there are many cases in which touch is critical to manipulating deformable objects in our daily life. For example, we use the finger and palm to feel the magnitude and direction of tension when tightening a pair of shoe laces, or we use our hands to feel the shape and hardness of dough when making pizza, etc.

To bring robots into the daily life of human beings, we believe it is important to equip robots with the capability of leveraging the sense of touch to interact with deformable objects intelligently. Currently, there are few works on the tactile-driven manipulation of deformable objects, and numerous modeling, planning, and control challenges remain to be solved for realizing such a goal. One fundamental challenge is that many of the existing frameworks for planning and control in robotics were developed under the assumption of a world filled with rigid bodies. However, for the tactile-driven manipulation of deformable objects, both state-of-the-art tactile sensors and the objects are deformable. Hence, it is extremely difficult to establish a physics-based model to describe

physical interactions between the sensor and the object, and to apply the rich, well-established theories for planning and control, accordingly. Another fundamental challenge is that, for many modern tactile sensors, the tactile information is provided in the form of high-dimensional data, as with camera-based tactile sensors. Intuitively, high-dimensional tactile data can benefit perception since high-dimensional data usually means enhanced spatial resolution. Yet, high-dimensional tactile data can impede the development of practical and effective planning and control methods because it is difficult to directly describe how the system state changes as a consequence of a control action. Therefore, to address these challenges and develop robots capable of the tactile-driven manipulation of deformable objects, we conducted studies to construct systematic frameworks for tactile perception, modeling, planning, and control that were inspired by many of the recent advances in robotics and machine learning.

1.2 Contributions

This dissertation aims to develop systematic frameworks for the tactile-driven manipulation of deformable objects. We propose to bring recent technical advances from machine learning, optimization, and control theory into the construction of the systematic frameworks such that the difficulty of hard-to-model and high-dimensional tactile feedback related to deformable object manipulation can be circumvented or practically addressed for real robot experiments. We believe our works provide initial foundations for developing systematic frameworks that can handle more complex, tactile-driven deformable object manipulation problems for semi-autonomous robots in our physical world and everyday life.

Chapter 2 presents a framework that combines model-free learning and imitation learning for the tactile-driven manipulation of thin-shell deformable objects. We demonstrated the effective-

ness of the framework on a real robot for the task of flipping a notebook page. The robot relied directly on tactile feedback in order to learn a functional behavior after starting from an non-functional behavior. Moreover, for similar tasks with a different task context (page size), tactile feedback generated from the functional behavior learned for one task context was used to guide the adaptation of robot behavior for a novel task context.

Chapter 3 presents a framework that uses both unsupervised learning and supervised learning for the tactile-driven manipulation of deformable linear objects. We demonstrated the effectiveness of the framework on a real robot for the task of controlling the pose and tension of an elastic cable. Data-driven approaches were leveraged in order to identify a system dynamics model for physical interactions between a deformable tactile sensor and the elastic cable. A recurrent neural network and probabilistic generative modelling techniques were leveraged in order to extract low-dimensional latent space features from high-dimensional raw tactile sensor data. The recurrent neural network was then used to guide the supervised learning of a Gaussian process model, which has fewer parameters than a recurrent neural network but retains sufficient representative power. The Gaussian process model that encoded the latent space dynamics was further integrated with an optimization-based, model predictive control framework for motion planning and control. The overall framework has an "end-to-end" flavor since the input information is high-dimensional raw tactile data and the output is a manipulation behavior.

Chapter 4 summarizes the dissertation and presents opportunities for future work that build on the research in this dissertation.

CHAPTER 2

Autonomous Learning of Page Flipping Movements via Tactile Feedback

2.1 Abstract

Robotic manipulation is challenging when both the objects being manipulated and the tactile sensors are deformable. In this work, we addressed the interplay between the manipulation of deformable objects, tactile sensing, and model-free reinforcement learning on a real robot. We showed how a real robot can learn to manipulate a deformable, thin-shell object via feedback from deformable, multimodal tactile sensors. We addressed the learning of a page flipping task using a two-stage approach. For the first stage, we learned nominal page flipping trajectories for two page sizes by constructing a reward function that quantifies functional task performance from the perspective of tactile sensing. For the second stage, we learned adapted trajectories using tactile-driven perceptual coupling, with an intuitive assumption that, while the page flipping trajectories for different task contexts (page sizes) might differ, similar tactile feedback should be expected from functional trajectories for each context. We also investigated the quality of information encoded by two different representations of tactile sensing data: one based on the artificial apical tuft of bio-inspired tactile sensors, and another based on PCA eigenvalues. The results and effectiveness of our learning framework were demonstrated on a real 7-DOF robot arm

and gripper outfitted with tactile sensors.

2.2 Introduction

Manipulation skills are important human capabilities. With these skills, humans are able to tackle a wide range of tasks requiring different levels of dexterity, using a large variety of objects, and having distinct desired outcomes. For tasks requiring dexterity, the sense of touch plays a major role in enabling the prediction of key state transitions that occur during manipulation actions. Distinct tactile patterns are associated with transitions such as the making and breaking of contact with an object, or changes in weight during the lift and replacement of a grasped object. Predicting such state transitions allows the human to detect and react to undesired events that produce deviations from the desired task states. This capability is particularly relevant when vision is occluded and the task state cannot be visually inferred [46]. When the sense of touch is taken away, complementary sensory mechanisms such as vision are often insufficient for completing manipulation tasks with the same level of functional performance. This importance of the sense of touch is demonstrated by a human perception study in which a subject, whose sense of touch at the fingertip is temporarily impaired by anesthetization takes much longer to execute a match-lighting task that seems trivial before the anesthesia [45]. Although one could argue that this demonstrates that humans can still perform manipulation tasks using complementary sensory mechanisms, it also demonstrates the importance of tactile sensing for tasks requiring dexterity.

To achieve performance with dexterity comparable to the humans, robots could be equipped with tactile sensors that provide rich information about the contact interactions between themselves and their environment [120], [22]. Once equipped with such sensors, we believe that the ability to complete complex manipulation tasks is dependent on how three learning challenges are

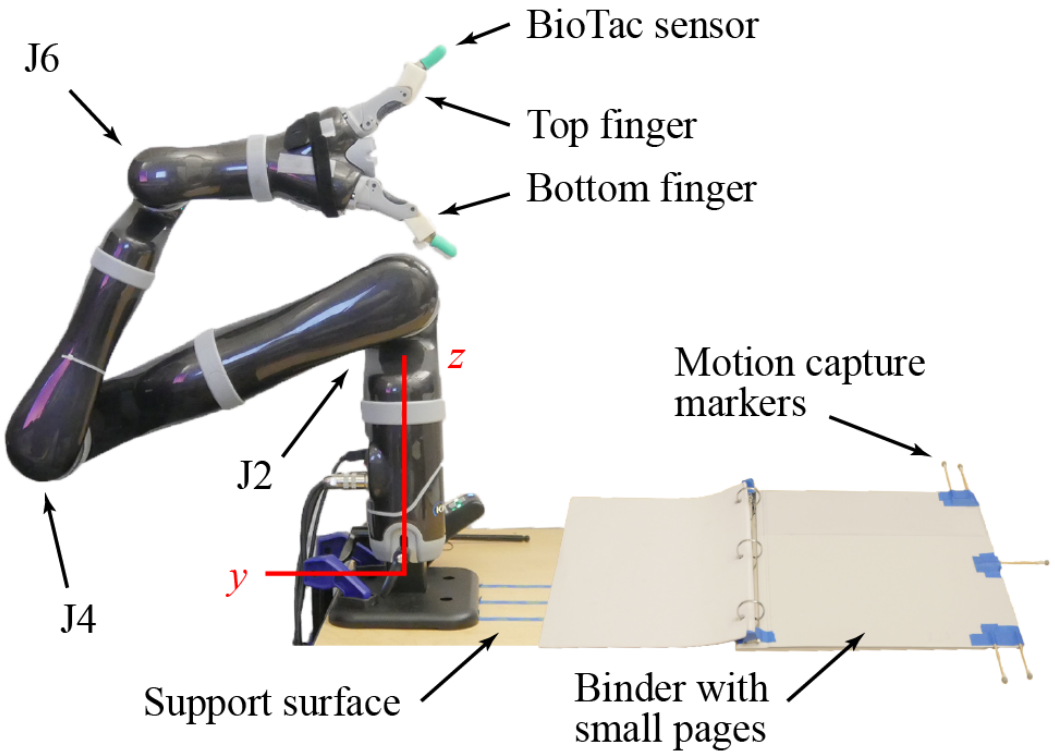

Figure 2.1: Three joints (J2, J4, J6) of a 7-DOF Kinova robot arm are controlled to perform page flipping movements in the y-z plane using two fingers outfitted with deformable, multimodal BioTac tactile sensors. Large and small notebook pages (shown) were held by rigid binders placed on a flat support surface parallel to the x-y plane. Passive motion capture markers are attached to the binders for tracking displacement.

addressed: Firstly, the robot needs to learn how to associate specific task state transitions with corresponding sensory events and use this association, or mapping, to detect undesirable states and evaluate the functional performance of the task. Second, once the ability to detect undesirable states through sensory events is acquired, the next problem is to learn which actions to perform in order to compensate for the observed sensory deviations and to return to states associated with acceptable functional task performance. Examples of such corrective actions are incremental adjustments to grasp forces after sensing slippage between the fingertips and the surface of a grasped

object, or adjustments of wrist positions to make sure a scraping tool maintains sufficient contact with a tilted surface [96]. Finally, once the previous two learning challenges have been overcome, it is crucial to achieve generalization of these capabilities to novel contexts or scenarios [51, 98].

In the past, a major bottleneck for solving the three aforementioned learning challenges lay in the limitations of tactile sensing hardware technology. Traditional tactile sensors composed of pressure sensing arrays have low deformation capabilities [42], which may not encode sufficient information for tasks that require detailed knowledge of the finger-surface interactions. In recent years, efforts have been made to develop deformable tactile sensors that can provide multimodal and high-resolution spatial information, such as the BioTac [116], GelSight [122] and TacTip [114]. Although several successful applications of these sensors exist for tasks such as slip detection [106, 43] or object property classification [61], there have been few examples of their application to more general forms of object manipulation [107, 55, 25, 99]. When the manipulated object is highly deformable (e.g., thin-shell objects such as paper), the application of tactile sensing seems to be even more rarer.

Currently, there are two key challenges that limit the application of state-of-the-art tactile sensors to dexterous object manipulation problems: (i) the dynamics of the interactions between the sensors and the manipulated objects are nontrivial to model, especially for cases where both the sensor and the object are deformable, and (ii) in most cases, it is difficult to quantify the overall functional performance of a manipulation task solely based on tactile information that is inherently localized to finger-object interactions.

To overcome the aforementioned challenges, we propose an approach that begins by deploying a model-free reinforcement learning process seeded via human demonstrations that is then guided by a tactile-based reward function in order to learn a nominal movement trajectory for a specific task context. Using model-free reinforcement learning, we do not require explicit models

of the system components (e.g., the tactile sensors, the manipulated objects, and the interactions between them) and are able to extract the necessary information solely from the tactile and proprioceptive data acquired during the process, as shown in other motor skill acquisition tasks using real robots [50], [23]. Once the nominal movement trajectory has been learned, the corresponding sensor readings are considered as the nominal sensing traces. In order to generalize the learned movements to different task contexts, we operate under the assumption that, for tasks where the trajectories required for each context are different, the resultant nominal sensing traces associated with the functional behavior should still be similar. Under this assumption, we can use the differences between sensing traces in order to adapt the movement trajectory to a different context. Such an adaptation is achieved by a separate reinforcement learning process, where adjustments to the nominal trajectory are learned using the differences between the nominal sensing traces and the actual sensor traces, acquired during execution of the learned movement trajectory in a new context. We use our approach to tackle a notebook page flipping task, where both the tactile sensors and manipulated objects are highly deformable. Different contexts result from using different notebook page sizes that require different movement trajectories to flip pages at an acceptable performance level.

Our work contributes to the development of new reinforcement learning approaches for the manipulation of deformable objects while explicitly leveraging state information encoded in tactile sensor data. More specifically, the contributions of this paper are the following: (i) we show that a nominal trajectory with functional behavior can be learned using model-free reinforcement learning and a tactile-based reward function, (ii) we achieve the adaptation of such functional behaviors to a novel context by relying solely on the differences between tactile sensing traces generated by a nominal trajectory and those generated for a novel context, and (iii) our learning approach demonstrates the manipulation of highly deformable thin-shell objects with a real robot.

Section 2.3 outlines related work. Section 2.4 details the manipulation task and our general methodology. Section 2.5 provides a description of the hardware used in our experiments as well as a discussion of our experimental results. Section 2.6 summarizes contributions and limitations of this work, and suggests directions for future work.

2.3 Related Work

The three topics that interplay in this paper are object deformability, modeling of tactile sensors, and object manipulation in robotics. In this section, we provide brief introductions to related work on each of these topics.

Robotic manipulation of objects (rigid or deformable) has been an active area of research for quite some time. Regarding the manipulation of rigid object, several efforts have focused on grasping or contour-following tasks, with approaches that rely on tactile sensing to enable the detection of salient discrete events (e.g. slip detection, stability estimation, force thresholding) [58, 93, 107], or utilizing tactile information as continuous feedback signals that drive corrective actions [55, 25].

While the manipulation of rigid objects has been extensively investigated, the same cannot be said for the manipulation of deformable objects, especially when considering approaches that leverage tactile sensing. Currently, state-of-the-art approaches to the manipulation of deformable objects rely predominantly on visual sensing [77], with tactile sensing mainly being explored for the classification of object properties [121, 123] or shape estimation [78, 111]. Such limited application of tactile sensing to the manipulation of deformable objects is not surprising, considering that it is extremely difficult to accurately model deformable objects, deformable tactile sensors, and their complex interactions during contact.

Nonetheless, some efforts have been made to model deformable linear objects (DLOs), [67]

such as ropes and cables [11, 83], as well as thin-shell objects, such as paper [26] and garments [73]. Still, all of these approaches either require complex models of internal physical states of deformable objects that are difficult to deploy in real robot experiments, or require sufficiently accurate complementary sensing mechanisms (e.g., fixed visual tracking markers) instead of tactile sensing.

A recent work by She, et al. uses tactile sensing to manipulate a deformable object. A deformable, computer vision-based tactile sensor (GelSight [122]) is applied to a cable following task [84]. The authors used a model-based approach for manipulation because the tactile images from the sensor enabled the state of the cable to be continuously observed throughout task execution. In this work, we sought to manipulate a deformable thin-shell object (notebook page). We employed a deformable tactile sensor (BioTac [116]) that does not enable a direct observation of system state for this particular manipulation task, and so we elected to use a model-free reinforcement learning approach.

Typically, endowing a robot with predictive tactile sensing capabilities has been approached as a forward modeling problem, which is nontrivial for deformable tactile sensors that engender soft contact (e.g. BioTac [116], GelSight [122], TacTip [114]). Successful predictive approaches include building latent space dynamics models for the BioTac sensor using deep representation learning to enable object surface servoing [97], and training deep recurrent neural networks to predict sequences of future GelSight tactile images from the current tactile image and applied control actions for the implementation of a model predictive control framework [99]. Note that in both of these examples, the objects being manipulated were rigid and the manipulative actions could be accurately observed and evaluated based on the tactile sensor information.

Recent manipulation approaches attempt to capitalize on the recent successes of reinforcement learning. Some approaches leverage simulation in order to pre-learn policies that are then trans-

ferred to real robotic systems [108, 39]. Other approaches begin directly with real robotic systems either by first learning how to evaluate the quality of their actions and using the quality assessments to guide the learning [18], or by focusing on a single manipulation action and only considering one object [105].

In order to deploy reinforcement learning on real robots and tackle more complex manipulation tasks, we elected to use policy representations with a limited number of parameters to encode the movement of the robot. Several policy representations have been proposed, including deterministic representations such as dynamic movement primitives (DMPs) [80] and probabilistic representations such as probabilistic movement primitives [70] or Gaussian mixture regression (GMR) [15]. In this work, we use the DMP framework mainly due to its successful application to motor skill learning problems with real robots [23].

2.4 Learning to Manipulate a Thin-shell Object via Tactile Sensor Feedback

We partition the learning challenges into two sub-problems in order to show, first, that a robot can learn a page flipping task using quantitative performance measures based on tactile sensing, and second, that deviations from expected tactile sensor feedback can be used to adapt nominal actions to different contexts. First, we learn nominal trajectories leading to the functional behavior of page flipping. To learn these nominal trajectories efficiently, we bootstrap a model-free reinforcement learning process seeded by human demonstrations via kinesthetic teaching. The reinforcement learning process is guided by a reward function based on tactile signals and motion tracking data. The tactile signals provide information about the contact state between the fingertips and grasped notebook pages. The motion capture data tracks the movement of the notebook for the evaluation of task performance.

Second, after learning a nominal page flipping trajectory, we learn an additional tactile-based feedback term that adapts the nominal trajectory to a different-sized notebook (a different context). The additional feedback term is denoted as a perceptual coupling term [49] and is in fact a separate correction policy. As previously mentioned, the correction policy is learned based on the assumption that, while the nominal movement trajectories for different page sizes might differ, the sensing traces corresponding to functional behaviors should remain similar. Hence, the correction policy for a novel page size should adapt the movement trajectory such that it reproduces the nominal sensing traces corresponding to the functional page flipping behavior generated by learning with the nominal page size. While this correction policy is also learned via a model-free reinforcement learning process, the reward function that guides this process is now purely based on tactile information.

In this section, we provide a brief introduction to the DMPs policy representation (Section 2.4.1) and the reinforcement learning algorithm used for both learning sub-problems (Section 2.4.2). We then describe a qualitative study aimed at establishing the relevance of tactile information to the page flipping task (Section 2.4.3). Leveraging insights from the qualitative study, we describe how we use tactile and marker tracking information to learn the nominal movement trajectories (Section 2.4.4) and how we use tactile sensor feedback exclusively to adapt the nominal trajectory to a different page size (Section 2.4.5). Finally, we describe several alternative representations for tactile information that ensure that maximal tactile information is provided to the reinforcement learning process in a computationally efficient (e.g. low-dimensional) manner (Section 2.4.6).

2.4.1 Dynamic Movement Primitives

In order to learn a nominal movement trajectory, the parameters of a trajectory representation are adjusted to reproduce a demonstrated trajectory and fine-tuned by a reinforcement learning

algorithm. In this work, we choose the Dynamic Movement Primitives (DMPs) as the parametric representation of a trajectory [80]. A DMP typically consists of (i) a transformation system (trajectory generator), (ii) a phase system, and (iii) a nonlinear forcing function. We also include a gating system to scale the magnitude of the forcing term [91]. We choose a transformation system for discrete movements [40] [91]

$$\begin{bmatrix} \dot{z} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} (\alpha_y(\beta_y(y_g - y) - z) + vf(s))/\tau \\ z/\tau \end{bmatrix}, \tag{2.1}$$

where y, y_g are the actual position and goal position of a robot movement respectively, α_y is a spring constant, β_y is a damping constant, τ is a temporal scaling factor of the movement duration, and $f(s)$ is a non-linear forcing function of a phase variable s that determines the shape of the robot trajectory.

The forcing function is defined as

$$f(s) = \frac{\sum_{i=1}^{N} \Phi_i(s)\omega_i}{\sum_{i=1}^{N} \Phi_i(s)} s(y_g - y_0), \tag{2.2}$$

with Gaussian kernel

$$\Phi_i(s) = \exp\left((s - c_i)^2/h_i\right). \tag{2.3}$$

where c_i and h_i represent the center and width of the Gaussian kernel, respectively. The forcing function is scaled by the difference between the start y_0 and goal position y_g, and by a gating variable. The gating variable v evolves as a sigmoid system [91] scaled by a time constant α_v,

$$\dot{v} = -\alpha_v v(1 - v/v_{\max}). \tag{2.4}$$

In addition, the forcing function depends on the phase variable s instead of explicitly depending on time. The phase variable evolves as a constant decaying system [91]

$$\dot{s} = -1/\tau. \tag{2.5}$$

Note that each degree of freedom for the robot has its own transformation system and forcing function. The synchronization of the multiple degrees of freedom is achieved via a shared phase variable.

To encode a demonstration movement y_{demo} as a DMP, the weights $\{\omega\}_i$ associated with the forcing function need to be adapted such that the generated robot movement matches the recorded human demonstration used to seed the learning process. The initial fitting of the DMP weights is achieved by solving the linear regression problem

$$\{\omega\}_i = \text{argmin}_{\{\omega\}_i} \sum_s (f_{\text{target}}(s) - f(s)) \tag{2.6}$$

where f_{target} is the target forcing function (human demonstration) and is computed by integrating the transformation system (Equation 2.2) using variables extracted from the demonstration.

2.4.2 Model-based Relative Entropy Policy Search

We rely on reinforcement learning to fine-tune the initial movement trajectory that matches the demonstration and then to learn the correction policy to adapt the nominal movement trajectory. Specifically, we use an information-theoretic policy search approach: Model-based Relative Entropy Policy Search (MORE) [1]. By bounding the KL-divergence of two subsequent policy search distributions $KL(\pi(\theta)||q(\theta))$ and the co-variance matrix shrinkage of $\pi(\theta)$, MORE achieves an effective trade-off of exploration and exploitation.

For MORE, the learning problem of maximizing the reward function under the expectation of generated trajectory samples can be formulated as

$$\max_{\pi} J(\pi) = \int R(\theta)\pi(\theta)d\theta \quad \text{s.t.} \tag{2.7}$$

$$\int \pi(\theta) \log \frac{\pi(\theta)}{q(\theta)} d\theta \leqslant \varepsilon \tag{2.8}$$

$$-\int \pi(\theta)\log\pi(\theta)d\theta \leqslant \beta \tag{2.9}$$

$$\int \pi(\theta)d\theta = 1. \tag{2.10}$$

with the updated policy $\pi(\theta)$, KL-divergnce constraints and entropy bound constraints.

With an additional constraint that $\pi(\theta)$ is a proper probability distribution, the Lagrangian dual for this constraint optimization problem can be obtained in closed form and yields the following solution

$$\pi(\theta) \propto q(\theta)^{\left(\frac{\eta}{\eta+\gamma}\right)} e^{\left(\frac{R(\theta)}{\eta+\gamma}\right)}. \tag{2.11}$$

The new policy $\pi(\theta)$ is a geometric average of the current policy $q(\theta)$ and an exponential transformation of the reward function. The Lagrangian dual variables η and γ serve as "temperature" parameters that weight each sample drawn using the current policy.

MORE fits a quadratic surrogate model to reward function samples $R_\theta \approx \theta^T R\theta + \theta^T r + r_0$ and assumes that the current policy search distribution is Gaussian $q(\theta) = N(\theta|\mu,\Sigma)$. The new policy search distribution can be obtained in closed form as

$$\pi(\theta) = N(\theta|Ff, F(\eta+\gamma)) \tag{2.12}$$

where

$$F = (\eta\Sigma^{-1} - 2R)^{-1} \tag{2.13}$$

$$f = \eta\Sigma^{-1}\mu + r. \tag{2.14}$$

In practice, η needs to be restricted such that F is positive definite.

2.4.3 Relevance of Tactile Information to a Page Flipping Task

To gain insights into which page flipping behaviors are detectable via tactile sensing, multiple sets of DMPs parameters are fitted using different human demonstrations via kinesthetic teaching.

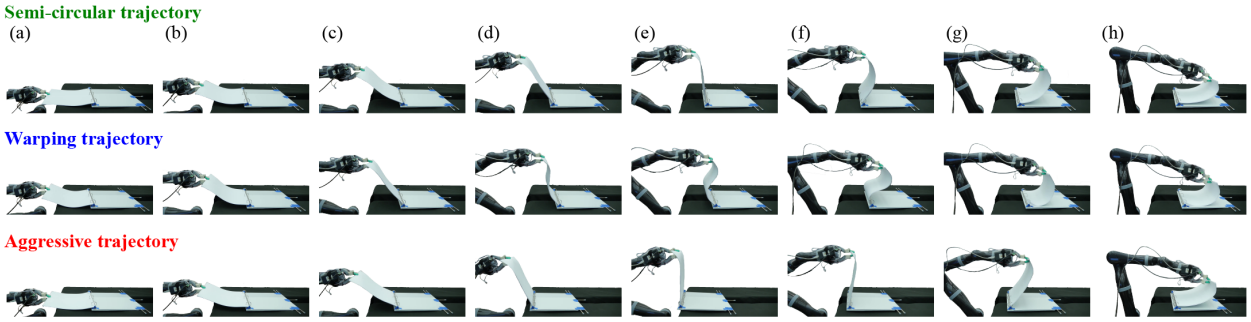

Figure 2.2: Representative snapshots of the three categories of page flipping trajectories described in Section 2.4.3 are shown in 15 sec increments and the supplemental video. *(Top)* The Semi-circular trajectory represents the ideal page flipping movement. *(Middle)* For the Warping trajectory, page warping is especially pronounced in subfigure *(f)* and page snapping results in the page configuration shown in subfigure *(g)*. *(Bottom)* For the Aggressive trajectory, the binder is pulled aggressively from subfigures *(d)* through *(g)*, resulting in large and numerous displacements of the binder in the y-z plane defined in Figure 3.1.

The demonstrated page flipping movements can be categorized into three groups (Figure 2.2):

1. *Semi-circular Trajectories:* In these demonstrations, the robot flips the pages with a relatively semi-circular movement, where the radius of the semi-circle is approximately equal to the width of the page.

2. *Warping Trajectories:* In these demonstrations, the trajectory is either mostly horizontal and parallel to the binder's support surface, or it will begin along a semi-circular path and then move downward toward the support surface prematurely, prior to the page being fully flipped. These trajectories cause the page to warp, leading to "page snapping" as the curvature of the page abruptly changes.

3. *Aggressive Trajectories:* In the initial period of these trajectories, the robot pulls the page

excessively. The unnecessarily large movement can undesirably slide the binder along the support surface or lift the binder off the support surface.

We use a deformable, multimodal tactile sensor called the BioTac (SynTouch, Inc., Montrose, CA, USA) to record tactile data ((Figure 2.4). Each BioTac measures low frequency pressure (P_{dc}), high frequency pressure (P_{ac}), data from 19 impedance electrodes (E), internal temperature (T_{dc}), and temperature flux (T_{ac}). All tactile sensing channels are provided at 100 Hz except for the high frequency pressure data, which are provided at 2200 Hz. For each of two fingers, 44 tactile signals are sampled at 100 Hz. In this work, we use the low frequency pressure (P_{dc}) and impedance electrodes (E) only.

After executing the DMPs for each trajectory category, several repeatable patterns can be observed in the tactile sensor data recorded by the deformable, multimodal BioTac sensor [116] used

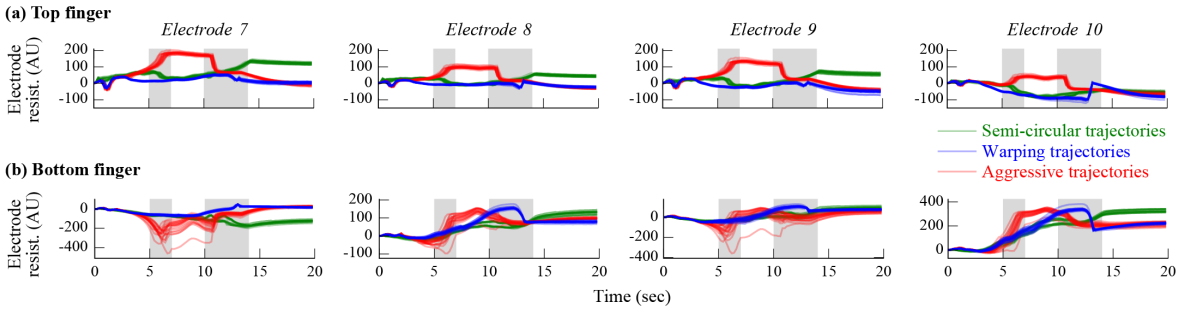

Figure 2.3: Representative tactile sensing traces from the artificial apical tuft (electrodes 7, 8, 9, 10 in Fig. 2.4) are shown in arbitrary units (AU) for the *(a)* top and *(b)* bottom fingers during the flipping of small pages when using the three categories of trajectories (15 rollouts each) described in Section 2.4.3. The semi-circular trajectories (green) generate smoother tactile signals than both warping trajectories (blue) and aggressive trajectories (red). As indicated by the periods shaded in gray, spikes in the tactile signals occur near the end of the page flipping movement for the warping trajectories and near the beginning and end of the movement for the aggressive trajectories.

in the experimental evaluation (Figure 2.3). For semi-circular trajectories, the tactile signals captured during the page flipping movement are relatively smooth, with very few, if any, movements of the binder during the execution of the trajectory. For the warping trajectories, the page snapping causes spikes in the low frequency pressure signal (P_{dc}) and in the electrode voltages (E) provided by the BioTac sensors. As the horizontal gripper trajectory moves closer to the binder's support surface, the more severe the page warping and snapping, making the spikes in the tactile signals more pronounced. For aggressive trajectories, signal spikes are observed when the binder is pulled toward the robot and hits the border of the support surface, when the binder is lifted from and returned to the support surface, and when the gripper moves to flip the page. In addition, aggressive pulling of the notebook pages can tear the page, effectively damaging the notebook.

Based on the above observations, it is clear that undesirable events can be detected as large shifts or transient spikes in several of the tactile sensor data streams. The signal spikes can be interpreted as contact state instabilities during the page flipping movements caused by sub-optimal trajectories. By design, optimal trajectories will attempt to minimize abrupt changes in the tactile signals in order to maintain stable contact throughout the page flipping movement. In order to track undesired gross movement of the binder, which cannot be fully characterized by tactile signals, passive motion capture markers are attached to the binder. Optimal trajectories will also attempt to minimize the movement of the markers, thus minimizing the pulling of the binder.

2.4.4 Learning a Nominal Trajectory from Tactile Feedback

For the first learning sub-problem, we learn a functional movement for flipping pages of a notebook by relying on tactile and marker tracking information. We begin by fitting the parameters of a DMP to a demonstration of a warping trajectory, as this category of trajectories exhibited the undesirable behavior during the pilot study presented in Section 2.4.3. We then use MORE

to further optimize the trajectory for improved functional performance. The policy distribution parameter θ in the problem formulation specified in Section 2.4.2 corresponds to the DMP weights $\{\omega\}_i$.

We design a novel reward function that simultaneously enforces that the contact areas between the fingers and notebook pages remain stable throughout the trajectory, and that the movement of the binder is minimized. Consider $i \in [1,2,\ldots,n]$ BioTac signal instances recorded during the page flipping trajectory with a sampling frequency of ϕ, and with low frequency pressure channels denoted as P_{dc}, electrode voltages denoted as E, and total displacement distances of the binder markers denoted as D_{markers}, the reward function is defined as

$$R(\tau) = R_{P_{\text{dc}}} + R_E + R_{\text{markers}} \tag{2.15}$$

where

$$R_{P_{\text{dc}}} = -\alpha(\Gamma_1 + \Gamma_2)^2 \tag{2.16}$$

penalizes trajectories with large shifts in the P_{dc} channels by considering the maximum shift captured during the trajectory for each of the d fingers, where

$$\Gamma_d = \max_i \left(\left| \frac{{}^d P_{\text{dc}}^{i+1} - {}^d P_{\text{dc}}^{i}}{\phi} \right| \right), i \in [1,2,\ldots,n]. \tag{2.17}$$

In a similar fashion,

$$R_E = -\sigma(\Lambda_1 + \Lambda_2)^2 \tag{2.18}$$

penalizes large shifts in the electrode values by considering the maximum average shift captured across the 19 electrodes for each of the d fingers, where

$$\Lambda_d = \max_i \left(\frac{1}{19\phi} \sum_{j=1}^{19} |E_j^{i+1} - E_j^i| \right), i \in [1,2,\ldots,n]. \tag{2.19}$$

Finally, marker movement is also penalized via

$$R_{\text{markers}} = -\lambda D_{\text{markers}}. \tag{2.20}$$

Note that the pressure, electrode, and marker movement reward terms are scaled by $-\alpha$, $-\sigma$ and $-\lambda$ respectively. The overall reward $R(\tau)$ depends quadratically on the tactile signals of each individual finger and linearly on the displacement distance of the binder.

The reward function defines the task accomplishment via salient tactile features throughout the trajectory. For the page flipping task in this work, a salient tactile signal indicating overall task success/failure at the end of the trajectory is not pronounced. However, there may be other tasks, such as the closure of a ziplock bag, where a salient tactile signal, such as a "click" upon bag closure might exist. For those cases, one could add an additional term to the reward function that acknowledges overall task accomplishment.

2.4.5 Adapting Learned Nominal Trajectories to a Novel Context

For the second learning sub-problem, we show that nominal page flipping trajectories learned for the first sub-problem can be adapted to different page sizes while relying solely on tactile information. In contrast to the first learning sub-problem that considered binder displacement in the reward function, the second learning sub-problem does not use any visual feedback related to binder displacement. Specifically, we use the tactile sensing traces produced by executing the optimal page flipping trajectory learned in Section 2.4.4 for a specific page size as the nominal tactile sensing traces.

We still require that pages of different sizes be flipped with semi-circular trajectories for functional behavior. As such, we propose that, while the movement trajectory needs to adapt to different page sizes, the tactile sensing traces should remain constant. The robot then learns how to adapt the nominal trajectory to a different page size by trying to match the new tactile sensing traces to the nominal sensing traces. In this manner, we extend learning based on a single demonstration to a different task context.

In order to adapt the nominal trajectories, the nominal DMP needs to be modified to adapt to step-based tactile signals. In this paper, we leverage "Perceptual Coupling Dynamic Movement Primitives" [49], also known as "Associative Skill Memories" [71]. After defining the nominal signal trace instance $S_{nom}(s)$ and the current sensing trace instance $S_{cur}(s)$, the adaptation actions are decided based on the difference between the nominal and current signal traces (i.e., the perceptual coupling term) during the execution of the current page flipping trajectory on pages with a different size. Since concurrent reactions to sensing trace differences depends on which sensing channel diverged from the nominal sensing trace, it is necessary to maintain separate weights for each sensing channel in the adaptation policy.

Therefore, we model the adaptation policy as a mixture of Gaussians that takes the sensing trace differences as inputs and adds the adaptation policy to the nonlinear forcing function of the nominal DMP as shown

$$\hat{f}(s) = f(s) + \sum_{j=1}^{m} \sum_{k=1}^{n} \hat{\omega}_{jk} e^{\frac{(s-c_k)^2}{h_k}} (S^j_{\text{nom}}(s) - S^j_{\text{cur}}(s))s. \tag{2.21}$$

Here, m represents the total number of sensing channels used as tactile feedback, n represents the total number of basis functions for each tactile channel, and $\{\hat{\omega}\}_{jk}$ represents the learnable weights of the adaptation policy for a single degree of freedom of the robot arm, which are also trained using the MORE algorithm. Since this task involves three robot joints (described further in Section 2.5.1), the total dimensionality of the learning problem is $m \times n \times 3$. For simplicity and better synchronization of the nominal DMP and the perceptual coupling element, the centers $\{c\}_k$ and widths $\{h\}_k$ of the Gaussian kernels are set to be identical across all three robot joints. Considering the above, a natural reward function to learn the adaptation policy is the sum of squared differences between the nominal and the current sensing traces over the course of an entire page flipping trajectory. In other words, we set the reward function equal to the square of the perceptual

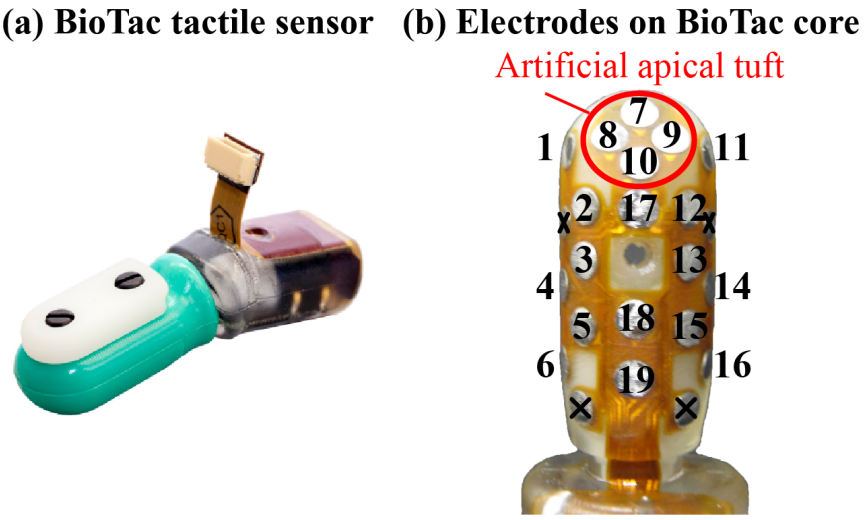

Figure 2.4: *(a)* The multimodal BioTac sensor is comprised of a rigid core, elastomeric skin, and fingernail (Image from [28]). *(b)* The rigid core of the BioTac is shown with impedance electrodes individually numbered. The red ellipse highlights the artificial apical tuft (flat region of the distal phalanx), where contact is made with the page in the majority of cases.

coupling term.

$$R(\tau) = -[S_{\text{nom}}(\tau(s)) - S_{\text{cur}}(\tau(s))]^2 \tag{2.22}$$

2.4.6 Representation of the Tactile Sensing Traces

Our choice of representation of the tactile sensing traces requires a careful balance between richness of tactile information for effective learning and computational tractability for deployment on a real robot. In one extreme case, we could naively use all sensing channels from both BioTacs on the gripper, resulting in 88 total tactile sensing channels (1 P_{dc} channel, 19 E channels, 2 channels associated with temperature, and 22 high frequency pressure values.

Hypothesizing that some of the native BioTac sensing channels would not be necessary for learning the page flipping task, we consider two possible representations for the sensing traces.

The first representation that we consider is comprised of a subset of the complete set of BioTac channels. Upon inspection of the page flipping trajectories, we observed that electrodes 7, 8, 9, and 10, located on the artificial apical tuft (flat surface of the distal phalanx) of the BioTac (Figure 2.4b), are stimulated most strongly and most often during contact with the page in the majority of trials. By focusing only on the P_{dc}, E_7, E_8, E_9, and E_{10} values from each finger, we reduce the dimensionality of the representation of tactile sensing traces from 88 to 10 values.

To further reduce complexity, we average signals over the four electrodes for each individual finger, which results in a total of 4 values (one P_{dc} value and one mean electrode value $\overline{E}_{7-10}$ per finger) sampled at 100 Hz. We believe that averaging over the four apical tuft electrodes is reasonable since we are primarily interested in the average skin deformation of that specific area of the sensor. To appropriately scale P_{dc} and $\overline{E}_{7-10}$, we normalize the data on each signal individually using sensing traces collected from the nominal trajectory learning experiments and assume that they provide a reasonable range for sensing traces that the robot can experience during page flipping.

The second representation that we consider uses Principal Component Analysis (PCA) to reduce the complete set of BioTac channels to a subset that captures most of the variance in the tactile sensing traces that can be leveraged for learning. Again, we begin by focusing only on the P_{dc}, E_7, E_8, E_9, and E_{10} values from each finger. Upon normalizing the signals as described previously, we pool the tactile signals across both fingers and apply PCA to further reduce the representation of tactile sensing traces from 10 dimensions to 3 dimensions.

Note that different sensing channels on different fingers (2 P_{dc} + 8 electrodes) form the state vector. Prior to performing PCA, the data from each signal channel are individually normalized. This normalization is performed to ensure that the PCA results will not be biased by large magnitude changes resulting from differences in measurement units, measurement ranges, or channel

sensitivity.

The first three principal components explain 96% of the total variance in the original 10-D tactile sensing space. The 1st, 2nd, and 3rd principal components explain 45.8%, 38.7%, and 11.5% of the total variance, respectively. We obtain the PCA projection matrix once at the start of the experiment. During runtime, the 10-D normalized BioTac signals are passed through the PCA projection matrix comprised of the first three principal components in order to yield three PCA eigenvalues for learning.

2.5 Experimental Procedure and Evaluation

In this section, we present the experimental procedures that were used to evaluate our approach and discuss the results of those experiments. First, we describe the hardware setup used in our experiments in Section 2.5.1. We then present the training procedure and results for the nominal trajectory learning sub-problem in Section 2.5.2. In Section 2.5.3, we present the results of a simplified version of the trajectory adaptation learning sub-problem and assess the impact of the choice of the representation of the tactile sensing traces introduced in Section 2.4.6. In this simplified version of the trajectory adaptation learning sub-problem, some partial knowledge of the novel task context (novel page size) is provided in order to make the adaptation policy learning problem more tractable while we focused on the assesment choice of tactile sensing trace representation. Finally, in Section 2.5.4, we present the results of the complete trajectory adaptation learning sub-problem, without the benefit of a priori knowledge of novel page size. For this final, complex learning experiment, we used the PCA eigenvalue representation of the tactile signals (Section 2.4.6) generated from the nominal trajectory (Section 2.4.4) learned for large pages.

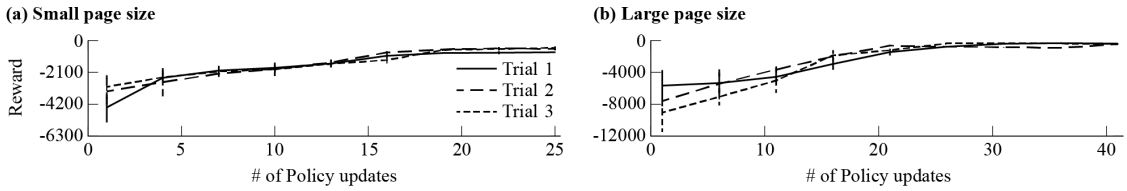

Figure 2.5: Learning curves are shown for learning nominal trajectories for flipping *(a)* small pages over the course of 25 policy updates and *(b)* large pages over the course of 41 policy updates. Mean and variance are presented for batches of 10 rollouts.

2.5.1 Experimental Set-up

For all experiments, we used a 7 degree-of-freedom (DOF) robot arm (JACO, Kinova, Boisbriand, Quebec, Canada) outfitted with a 4-DOF, three-digit gripper (KG-3, Kinova, Boisbriand, Quebec, Canada) (Figure 3.1). The ulnar digit was removed from the gripper in order to enable a two-digit precision grip. Each fingertip was equipped with a BioTac tactile sensor, as introduced in Section 2.4.3.

The robot was commanded to grasp and flip two different sizes of notebook pages (small page: 8.5” x 11”, large page: 11” x 11”). Retroreflective markers and six T-Series cameras sampled at 100 Hz (Vicon, Culver City, CA, USA) were used to track a rigid binder, containing the notebook pages, that was placed on a support surface parallel to the x-y plane (Figure 3.1). The binder displacement values in the y-z plane were used by the reward function described in Section 2.4.4 for learning a nominal trajectory for the page flipping task.

As shown in Figure 2.2, the page flipping movement occurs within the y-z plane defined in Figure 3.1. Through purposeful placement of the robot arm with respect to the binder, we simplify the policy learning problem. Specifically, we operate the robot arm within the y-z plane only. We control only joints 2, 4, and 6 (Figure 3.1) and constrain all remaining joints, thereby reducing the

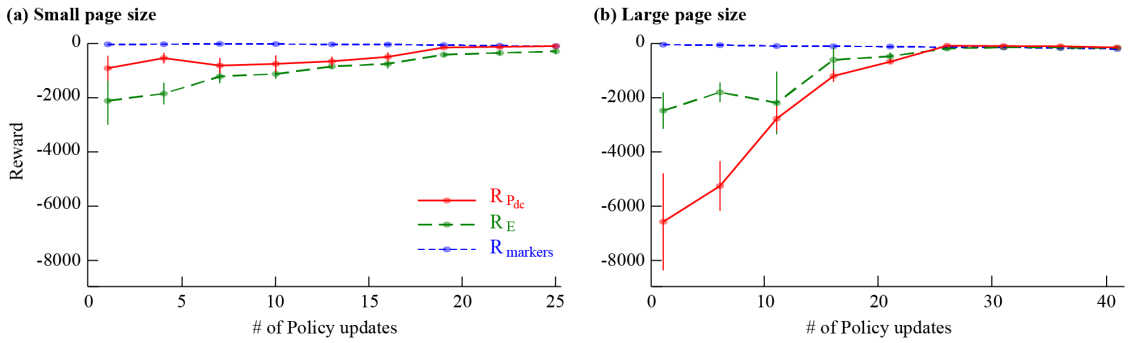

Figure 2.6: Individual reward components from Equation 2.15 are shown for the nominal trajectory learning curves in Figure 2.5 for flipping *(a)* small pages over the course of 25 policy updates and *(b)* large pages over the course of 41 policy updates. Mean and variance are presented for batches of 10 rollouts. This figure illustrates that the improvement of tactile-related reward components plays a major role in the improvement of the overall reward function as compared to any losses resulting from binder displacement.

dimensionality of the policy weights to be tuned during learning.

2.5.2 Learning a Nominal Trajectory from Tactile Feedback

Using the proposed framework introduced in Section 2.4.1 and Section 2.4.2, along with the reward functions defined from Equation 2.15 to Equation 2.20 in Section 2.4.4, we learn nominal trajectories for flipping pages of two different sizes. In each learning trial, the robot first moves to a "home" position and grips a pre-set stack of 20 pages.The home position is determined from a single human demonstration per page size at the start of the experimental session. An experimenter kinesthetically teaches the robot by grasping the robot and guiding it through a suboptimal Warping trajectory, as described in Section 2.4.3. Throughout the kinesthetic teaching, joint angles and joint angular velocities are recorded at 50 Hz. The kinematic data from the human demonstration are used to initialize ten parameters for each of three DMPs (one DMP for each of joints 2, 4, and 6 of

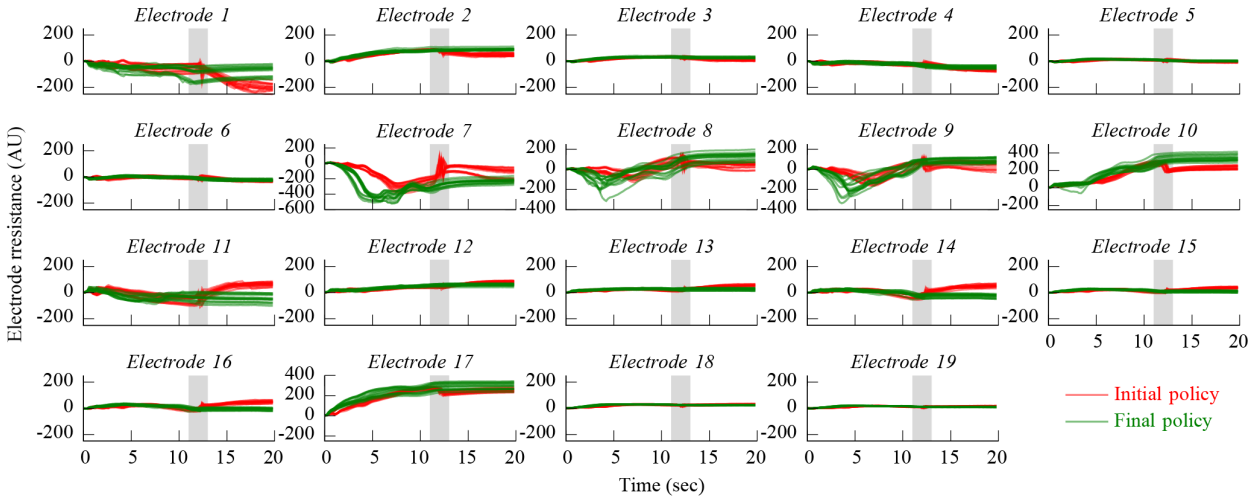

Figure 2.7: Data for each of 19 electrodes are shown for the bottom finger for learning a nominal trajectory for flipping a large page. A total of 15 rollouts are shown for each the initial policy after a single policy update (red) and final policy after 41 policy updates (green). The initial policy generates undesired spikes in the electrode signals during the page warping period shaded in gray.

the robot arm).

Upon initializing the DMP parameters, a reinforcement learning process, introduced in Section 2.4.2, refines the DMP parameters using the reward function defined in Section 2.4.4. The MORE policy search space has a dimensionality of 30 (three DMPs, each with ten parameters). The MORE ε and β parameters are set to 0.1 and 0.075, respectively. To fill a sample buffer, a total of 40 rollouts are executed and corresponding rewards are generated based on the initial policy distribution, which was defined as a multivariate Gaussian with mean values set equal to the initial DMP weights and a diagonal covariance matrix that was tuned based on preliminary data. Once the sample buffer is filled, the policy distribution is updated every five rollouts.

The learning curves for the nominal trajectory learning experiments are shown in Figure 2.5 for the small and large pages. Rewards from the updated policies are shown in increments of 3

policy updates (10 rollouts each) for three independent learning trials for the small pages, and in increments of 5 policy updates (10 rollouts each) for three independent learning trials for the large pages.

Since the reward values converge to zero, we see that a policy is successfully refined and learned for each of the two page sizes. The supplemental video shows that the robot learns page flipping trajectories that do not induce undesired page warping or page snapping, thereby avoiding spikes in the tactile signals by design of the reward function.

For brevity, we present tactile sensing traces for the nominal trajectory learning sub-problem for large pages only. Reflecting local deformation of the fluid-filled BioTac fingerpad, data from 19 electrodes are presented for the bottom finger in Figure 2.7. A total of 15 rollouts are shown for two policies: the initial policy after a single policy update and the final policy after 41 policy updates. The corresponding P_{dc} data are presented in Figure 2.8. The results for learning a nominal trajectory for small pages are similar to Figures 2.7 and 2.8 except that the final policies are learned after only 25 policy updates. Figures 2.7 and 2.8 show that rollouts of the initial policy, shown in red, generate undesired spikes in the tactile sensing traces during the page warping period shaded in gray. However, the rollouts of the final policy, shown in green, result in much smoother tactile sensing traces characterized by a significant reduction in spiking behaviors. The associated reduction in page warping is also demonstrated in the supplemental video.

Figure 2.6 and Table 2.1 show that tactile state is more relevant and plays a more significant role in the nominal trajectory learning process than binder displacement. Figure 2.6 illustrates how individual components of the reward function defined in Equation 2.15 contribute to the learning process and change over the course of learning. The improvement in the tactile reward components ($R_{P_{\text{dc}}} + R_E$) play a major role in the improvement of the overall reward function. We also observe that the tactile reward components for the large pages are worse when compared with those for the

small pages. Given that large pages are heavier than small pages, it makes sense that the negative tactile consequences of page warping and page snapping would be more pronounced in the tactile reward components for the large pages.

Table 2.1 compares the mean and standard deviation of binder displacement distances before and after the learning process. Binder displacement values are reported from 15 rollouts for each combination of page size (small or large) and learning stage (before or after learning). Although Table 2.1 shows that binder displacements are larger after learning, the negative effects of binder displacement on the overall reward function are dwarfed by the significant improvements in the tactile reward components, indicating that the learning process is dominated by tactile state, as desired.

Tactile state is more relevant for learning nominal trajectories for two main reasons. First, the magnitudes of the changes in tactile data due to page warping and page snapping are greater than those for marker movement resulting from binder displacement. Second, we chose the magnitudes of the scaling factors α, σ, λ in Equations 2.16, 2.18, and 2.20, respectively, such that marker movement would be considered but not heavily weighted in the overall reward function (Equation 2.15). If we were to increase λ to more heavily weight marker movement than tactile state, we would contradict our definition of what constitutes a functional page flipping behavior, as trajectories that result in page warping and page snapping would be improperly rewarded. Specifically, α was -0.0075 and λ was -2.5 for both page sizes, and σ was -0.0125 for the large page size and -0.05 for the small page size. The scaling constant σ was increased manually for the small page size in order to compensate for the fact that smaller tactile signal spikes result from the snapping of smaller pages.

Although we show that it is possible to learn functional page flipping behaviors for different page sizes using tactile information, some limitations were observed. For example, if the initial

Table 2.1: Binder displacement distances are reported as mean (standard deviation) from 15 roll-outs for each combination of page size and learning stage. "Before learning" refers to the initial policy after a single policy update. "After learning" refers to the final policy after 25 and 41 policy updates for the small and large pages, respectively.

	Before learning	After learning
Small pages	32.7 (0.5) mm	96.0 (3.7) mm
Large pages	24.0 (5.3) mm	184.9 (9.9) mm

policy produces a trajectory that is too low and close to the support surface, the pages warp and then contact the binder during the page warping period. As a result, transient perturbations due to page snapping are absorbed by the friction between the binder and the pages, and are not sensed by the BioTacs, whose tactile signals will be smooth. The policy search then gets stuck in a local optimum in which the trajectories appear to maximize the reward function when, in fact, the page flipping behaviors are unacceptable. We acknowledge that the reward function may not capture the tactile consequences of all possible notebook page flipping trajectories. To address this, one could supplement tactile sensing with a complementary sensing modality, such as vision, during learning.

Another example of a limitation is the aforementioned binder displacement after learning. Sometimes, the increased displacement of the binder after learning results in a learned trajectory that is not perfectly semi-circular (as seen in the supplemental video). One possible reason for this result is that perturbations around the set of DMP weights that generate perfectly semi-circular trajectories, can actually cause the robot to move toward the support surface prematurely, resulting in page warping. Page warping would cause the rewards to deteriorate significantly due to the dominant role of the tactile reward components. Thus, page warping is avoided through

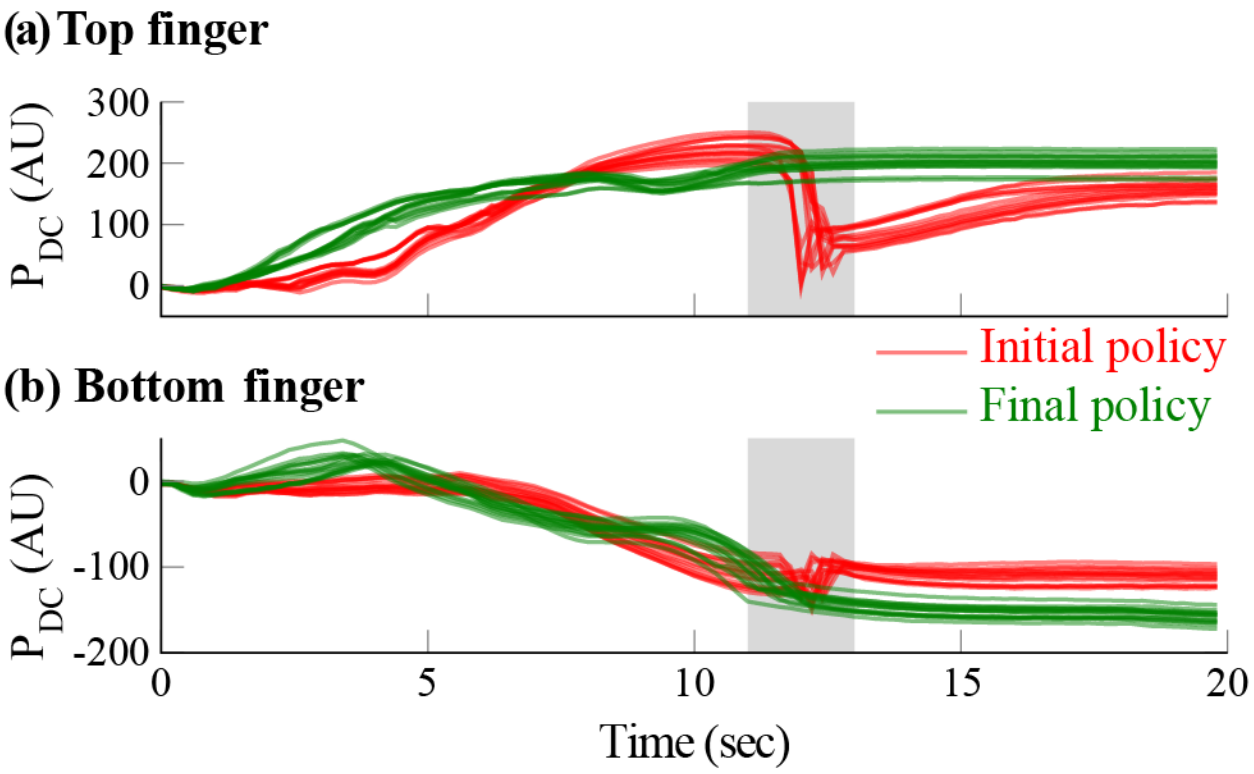

Figure 2.8: Low frequency pressure data are shown for the *(a)* top and *(b)* bottom fingers for learning a nominal trajectory for flipping a large page. A total of 15 rollouts are shown for each the initial policy after a single policy update (red) and final policy after 41 policy updates (green). The initial policy generates undesired spikes in the low frequency pressure data during the page warping period shaded in gray.

learned trajectories that are not perfectly semi-circular. Specifically, minor perturbations in DMP weights from semi-circular trajectories can result in task performance and rewards having a large variance. A large variance in the reward function values will be deemed undesirable during policy updates, especially if the values of β and γ, which bound the KL-divergence and entropy reduction constraints of the MORE algorithm, are set to make the learning process risk-averse.

As is commonly done when performing reinforcement learning experiments on a real robot,

we tuned hyperparameters in order to ensure that the learning process would converge within a reasonable number of samples. Specifically, the hyperparameters were tuned such that the initial policy distribution would have sufficient variance to generate samples of DMP weights that would, in turn, generate page flipping movements with different degrees of page warping and snapping. In addition, covariance matrix values were increased for robot joint activations that were observed to be especially sensitive to changes in DMP weights during different phases of the page flipping trajectories. This variance in behavior provides a wide and meaningful range of page flipping behaviors and reward function samples that enable productive policy updates.

2.5.3 Impact of Chosen Representation of Tactile Sensing Traces

Once we successfully learned nominal trajectories for the page flipping task for both page sizes, we paused to examine the impact of the choice of representation of the tactile information on a simplified version of the sub-problem for learning adapted trajectories. Using the methods described in Section 2.4.6, we sought to reduce the dimensionality of the tactile sensing traces before attempting the full experiment on the adaptation of the learned nominal trajectories to a novel context. Specifically, we investigated how the values of the reward function samples were affected by two different simplified representations of the tactile sensing traces: (i) one mean P_{dc} and one mean electrode value $\overline{E}_{7-10}$ for the artificial apical tuft, per finger, and (ii) three PCA eigenvalues.

Figure 2.9 shows the reward function values for the two different representations of the tactile sensing traces. The reward function samples (specified for learning adapted trajectories in Section 2.4.5) are the result of 15 rollouts performed for each of six trajectories on a binder containing small pages. An ideal semi-circular trajectory is denoted as *Functional*. Three trajectories causing page warping and page snapping are denoted as *Warped_1*, *Warped_2* and *Warped_3*. The *Warped_1*

trajectory is special in that it is the nominal trajectory learned for large pages, but purposely applied to small pages (a different task context). One aggressive trajectory (*Aggressive_1*) pulls the binder upwards and away from the support surface. Another aggressive trajectory (*Aggressive_2*) pulls the binder toward the base of the robot arm.

Two comments can be made about the similarity in reward distributions between the *Functional* and *Aggressive_2* trajectories. First, the *Functional* trajectory was provided by a human demonstration, which could have resulted in a small degree of aggressiveness since the demonstrator had to manually move the robot arm in order to flip the page. Second, from a utilitarian perspective, a trajectory that is labeled *Aggressive_2* could be acceptable if it is close enough to a trajectory that is deemed *Functional*.

As expected, the desired *Functional* trajectories generate the best reward distributions, regardless of the representation of tactile sensing traces (Figure 2.9). For both the apical tuft and PCA eigenvalue representations, the reward distributions can be used to distinguish between functional page flipping trajectories and those that cause undesired page warping and snapping. Unfortunately, the reward distributions for the *Functional* and *Aggressive_2* trajectories overlap, which make these two categories of trajectories more difficult to be distinguished from one another when using the tactile-based reward function specified in Section 2.4.5 with either of the two representations.

Nonetheless, the PCA eigenvalue representation generates a broader reward landscape compared to that generated by the apical tuft representation (Figure 2.9). In particular, the PCA eigenvalue representation results in a larger difference in mean reward function values between the *Functional* and the *Aggressive_2* trajectories than the apical tuft representation. As a result, functional and non-functional behaviors can be better distinguished when using the PCA eigenvalue representation. It is possible that, by averaging the four electrode measurements across the artifi-

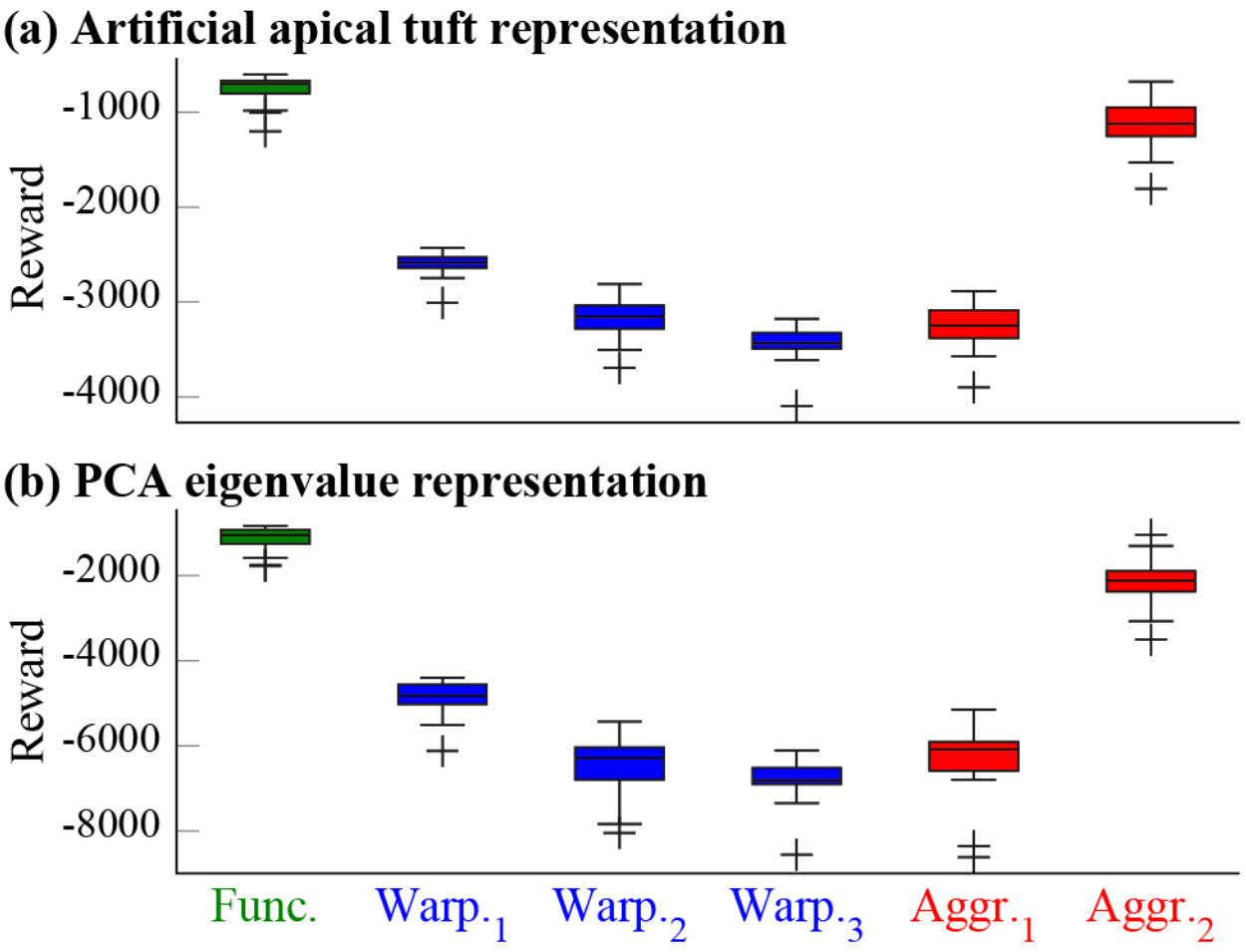

Figure 2.9: Distributions of reward function samples are shown for two representations of tactile sensing traces: *(a)* artificial apical tuft, and *(b)* PCA eigenvalues. Fifteen rollouts were performed for each of six trajectories on a binder containing small pages: an ideal semi-circular trajectory (*Functional*), three trajectories that cause page warping and snapping, and two aggressive trajectories. The *Warped_1* trajectory is the nominal trajectory learned for large pages, but purposely applied to small pages.

cial apical tuft, we lose information that may have encoded differences in page flipping behaviors.

In order to further test the impact of the tactile sensing representations, the first batch of experiments for learning an adaptation policy is conducted under the assumption that the goal position for small pages is known a priori. Specifically, the joint-specific values of the goal position parameter y_g in the perceptual coupling DMPs are set to the goal position values that were obtained from the human demonstration for small pages. Three independent learning trials are conducted with each tactile sensing trace representation. We initialize all DMP weights $\{\omega\}_i$ to the weights learned for large pages and all perceptual coupling feedback weights $\{\hat{\omega}\}_i$ are set to zero. The number of Gaussian basis functions in the perceptual coupling term is set to three. Since three robot joints are subject to control, the learning process explores a 36-D space (4 tactile traces $\times$ 3 basis functions per tactile trace $\times$ 3 robot joints) for the apical tuft representation and a 27-D space ($3 \times 3 \times 3$) for the PCA eigenvalue representation.

Figure 2.10 shows the learning curves using the apical tuft and PCA eigenvalue tactile sensing representations, respectively. For both tactile sensing representations, the MORE algorithm enables learning, as evidenced by an increase in the mean and the maintenance of a relatively small variance for the distribution of reward function samples. These results suggest that the perceptual coupling term in Equation 2.22 enables the tactile feedback to drive the adaptation of the initial trajectory intended for flipping large pages toward that necessary for flipping small pages.

While the learning curves are similar for both representations (Figure 2.10), the resultant adapted page flipping trajectories are quite different. Our observation is that the adapted trajectories are more aggressive when learned with the apical tuft representation than with the PCA eigenvalue representation. While avoiding page warping and snapping, the aggressive trajectories pull the binder closer toward the base of the robot before initiating page flipping and can even result in the binder hitting the edge of the support surface. The gentler page flipping trajectories

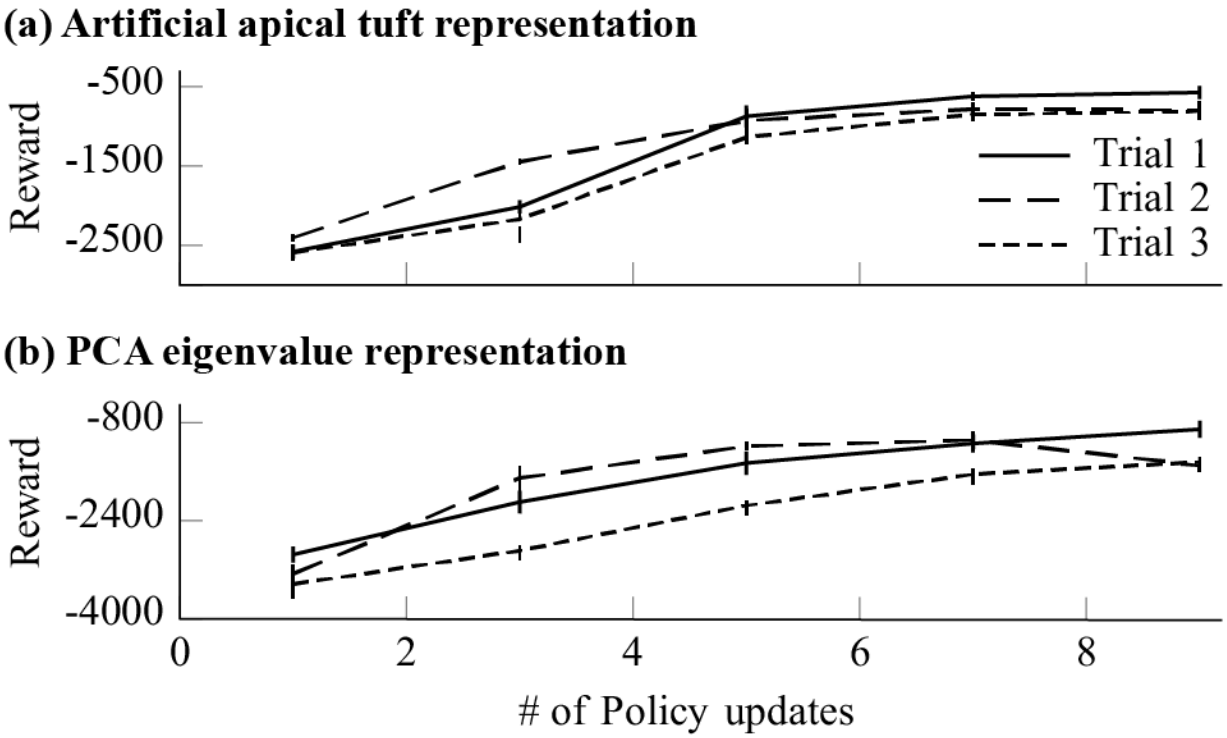

Figure 2.10: Learning curves are shown for the simplified experiment on adaptation of the learned nominal trajectories to a novel context using the *(a)* artificial apical tuft and *(b)* PCA eigenvalue representations of tactile sensing traces. The goal position for small pages is known a priori. Mean and variance are presented for batches of 10 rollouts.

learned with the PCA eigenvalue representation result in smaller displacements of the binder toward the base of the robot. Differences between the trajectories learned using the apical tuft and PCA eigenvalue representations can be seen in the supplementary video. Just as the PCA eigenvalue representation was preferred for distinguishing the functional trajectories from the aggressive trajectories (Figure 2.9), we conclude that the PCA eigenvalue representation is also preferred for learning adapted trajectories.

Figure 2.11 shows in greater detail how the tactile sensing traces for both representations change as the adapted trajectory is learned. The nominal tactile sensing traces are taken from

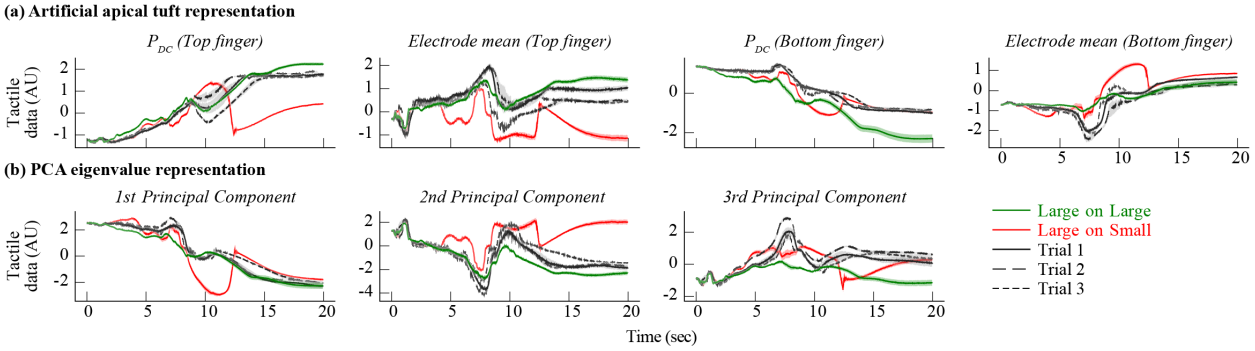

Figure 2.11: Distributions of tactile sensing traces are shown for the simplified experiment on adaptation of the learned nominal trajectories to a novel context using the *(a)* artificial apical tuft and *(b)* PCA eigenvalue representations of tactile sensing traces. The goal position for small pages is known a priori. Tactile data are shown for a nominal trajectory learned for large pages and applied to large pages (green) and to small pages prior to adaptation learning (red), for 10 rollouts each. Three independent learning trials (9 policy updates each) show how the tactile sensing traces change as the initial nominal trajectory for large pages is adapted to small pages during adaptation learning.

10 rollouts of a nominal trajectory learned for large pages (green). When the nominal trajectory learned for large pages is directly applied to small pages (a different task context), a much different set of tactile sensing traces results before any learning takes place (red). The results from three independent learning trials are shown, with each trial being comprised of 10 rollouts and 9 policy updates.

The adaptation of the tactile sensing traces encouraged by the perceptual coupling term in Equation 2.22 is most clearly illustrated in Figure 2.11 for the 1st and 2nd principal components of the PCA eigenvalue representation, which combine to explain 84.5% of the total variance in the original 10-D tactile sensing space. After learning to adapt the initial nominal trajectory for large pages to small pages, the tactile sensing traces for the the learning trials converge toward those

for the ideal case in which the nominal trajectory learned for large pages is appropriately applied to large pages. For the apical tuft representation, the adaptation of the tactile sensing traces after learning is most clearly shown for the top finger of the gripper in the P_{dc} and $\overline{E}_{7-10}$ data.

2.5.4 Adapting Learned Nominal Trajectories to a Novel Context

Based on the encouraging results described in Section 2.5.3, we adopted the PCA eigenvalue representation for the tactile sensing traces for the full experiment on adaptation of the learned nominal trajectories to a novel context. For the full experiment, we no longer provide any information about page size. As a result, the goal position y_g now becomes another axis in the policy search space. We show that a nominal trajectory learned for large pages can be successfully adapted to an unknown, novel page size (small, in this case) using perceptual coupling driven by a 3-D PCA eigenvalue representation of tactile feedback. In the simplified version of the sub-problem for learning adapted trajectories, the goal positions that were provided a priori were encoded in joint space. The learning problem becomes much harder when the joint-specific goal positions are no longer provided. Without the provision of such joint-specific constraints, it is possible that naive sampling of trajectories could lead to damage of the robot or movements that do not flip the page at all. To address this issue, we leverage the fact that the page flipping trajectories lie within a 2-D plane. As seen in Figure 3.1, the z-coordinates for the support surface and binder are constant. Accordingly, we assume that the z-coordinate of the goal position will be constant for the gripper regardless of page size. Since the y-coordinate of the gripper will vary according to page size, we represent different goal positions using the gripper's final y-coordinate.

During learning, the MORE algorithm samples values for the gripper's goal y-coordinate at the end of the page flipping trajectory. The pair of goal (y,z) coordinates for the gripper is then transformed into goal positions in joint space via an inverse kinematics solver. Using the PCA

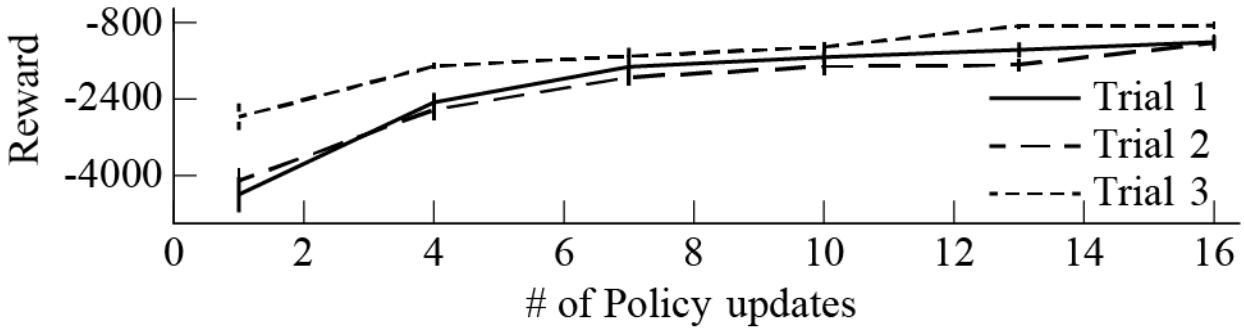

Figure 2.12: Learning curves are shown for the full experiment on adaptation of the learned nominal trajectories to a novel context using the PCA eigenvalue representation of tactile sensing traces. The goal position for small pages is not known a priori and must be learned. Mean and variance are presented for batches of 10 rollouts.

eigenvalue representation of the tactile sensing traces, the learning algorithm searches a 28-D space (3 tactile traces $\times$ 3 basis functions per tactile trace $\times$ 3 robot joints $+$ goal y-coordinate. As before for the simplified experiment on learning adapted trajectories (Section 2.5.3), all perceptual coupling feedback weights $\{\hat{\omega}\}_i$ are set to zero. We use the final y-coordinate of the gripper from the learned nominal trajectory to generate the initial estimate for y_g in joint space.

As shown by the learning curves in Figure 2.12, learning of the adapted trajectories was successful for the full experiment in which a nominal trajectory learned for large pages was applied to a novel task context (small pages). The initial trajectory rollouts result in aggressive movements in which the robot lifts the binder off of the support surface and drags the binder farther than necessary for small pages. As the adapted trajectory is learned using tactile-driven perceptual coupling, the distributions of reward function values improve, as reflected by the increase in mean and decrease in variance. After 16 policy updates, the small page is flipped gently, without lifting the binder from the support surface, and with less displacement of the binder.

Prior to adaptation learning, ten rollouts of a nominal trajectory learned for large pages and

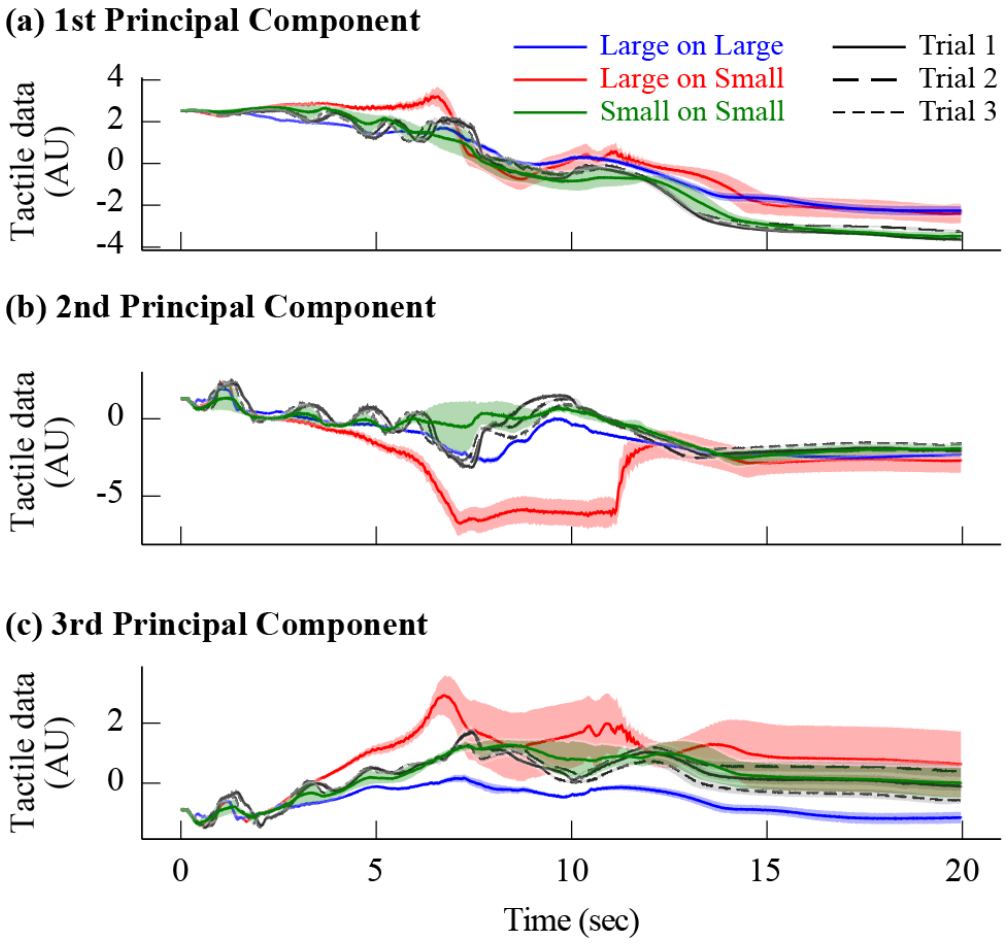

Figure 2.13: Distributions are shown for the PCA eigenvalue representation of tactile sensing traces for the full experiment on adaptation of the learned nominal trajectories to a novel context. The goal position for small pages is not known a priori and must be learned. Distributions (mean and variance) are shown for a nominal trajectory learned for large pages and applied to large pages (blue) and to small pages prior to adaptation learning (red), for 10 rollouts each. Three independent learning trials (16 policy updates each) show how the distributions change as the initial nominal trajectory for large pages is adapted for small pages after learning. The distribution of tactile feedback is also shown for the ideal case of a nominal trajectory learned for small pages and applied to small pages (green).

applied to small pages result in a mean value of -48.2 mm for the y-coordinate of the goal position. The mean value for the y-coordinate of the goal position is 40.0 mm for the ideal case in which a nominal trajectory learned for small pages is applied small pages. The final policies from three independent learning trials (16 policy updates each) result in mean values for the y-coordinate of the goal position of -40.8 mm, -41.0 mm, and -41.8 mm. In all cases, all standard deviation values were less than 0.03 mm. The mean value of the goal y-coordinate decreases by at least 6 mm (approximately 12%) as the trajectory is adapted from large pages to small pages. By the end of three independent learning trials, the mean goal y-coordinates are most similar to that for the ideal case in which a nominal trajectory learned for small pages is applied to small pages. This illustrates that the y-coordinate of the goal position is also learned and is successfully adapted from a value suited for large pages to a value appropriate for small pages. This demonstrates that a learned nominal trajectory can be successfully adapted to a novel task context using only the tactile sensing traces of a functional behavior as a reference.

Figure 2.13 shows in greater detail how the tactile sensing traces for the PCA eigenvalue representation change as the adapted trajectory is learned. Tactile sensing trace distributions (mean and variance) are shown in red for 10 rollouts of a nominal trajectory learned for large pages and naively applied to small pages (a novel task context). The ideal tactile sensing traces are shown in green for a nominal trajectory learned for small pages and applied appropriately to small pages. As desired, after 16 policy updates, the tactile sensing traces for the three independent learning trials converge upon those for the ideal case after learning the adapted trajectory and goal position using tactile-driven perceptual coupling.

For comparison, tactile sensing traces are shown in blue for a nominal trajectory learned for large pages that is applied appropriately to large pages. First, we see that task context does affect the tactile feedback, as exemplified by the slight differences between the tactile sensing traces

for the rollouts that do not require adaptation, but are learned for different page sizes (blue for large pages, green for small pages). Nonetheless, the tactile feedback for the rollouts that do not require adaptation (blue, green) are more similar to one another than to the tactile feedback for the rollouts that do require adaptation (red). This supports our assumption that, while the page flipping trajectories for different page sizes might differ, similar tactile sensing traces should be expected from functional trajectories for each of the page sizes. Second, we see that the learning trials that adapt to small pages lead to tactile sensing traces that are most similar to those from rollouts for small pages that do not require adaptation (ideal green case). This trend is most clearly visible for the 3rd principal component.

Figure 2.14 compares distributions of reward function values for different cases of trajectory rollouts. Reward function samples are shown in red for 10 rollouts of a nominal trajectory learned for large pages and naively applied to small pages (a novel task context). Reward function samples are shown in green for the ideal case in which a nominal trajectory learned for small pages is applied appropriately to small pages.

After 16 policy updates, the reward function samples for the three independent learning trials generally converge upon those for the ideal case after learning the adapted trajectory and goal position. The improvement in reward function values is most clearly seen in the boxplots for the 2nd and 3rd principal components. For the 1st principal component, it was initially surprising to see little improvement in reward function values with learning. We believe this may be caused by the fact that the trajectory is being adapted from a nominal trajectory learned for a different task context. Some of the undesired properties of the initial trajectory may remain prevalent in the adapted trajectory and are reflected in the 1st principal component of the tactile feedback representation.

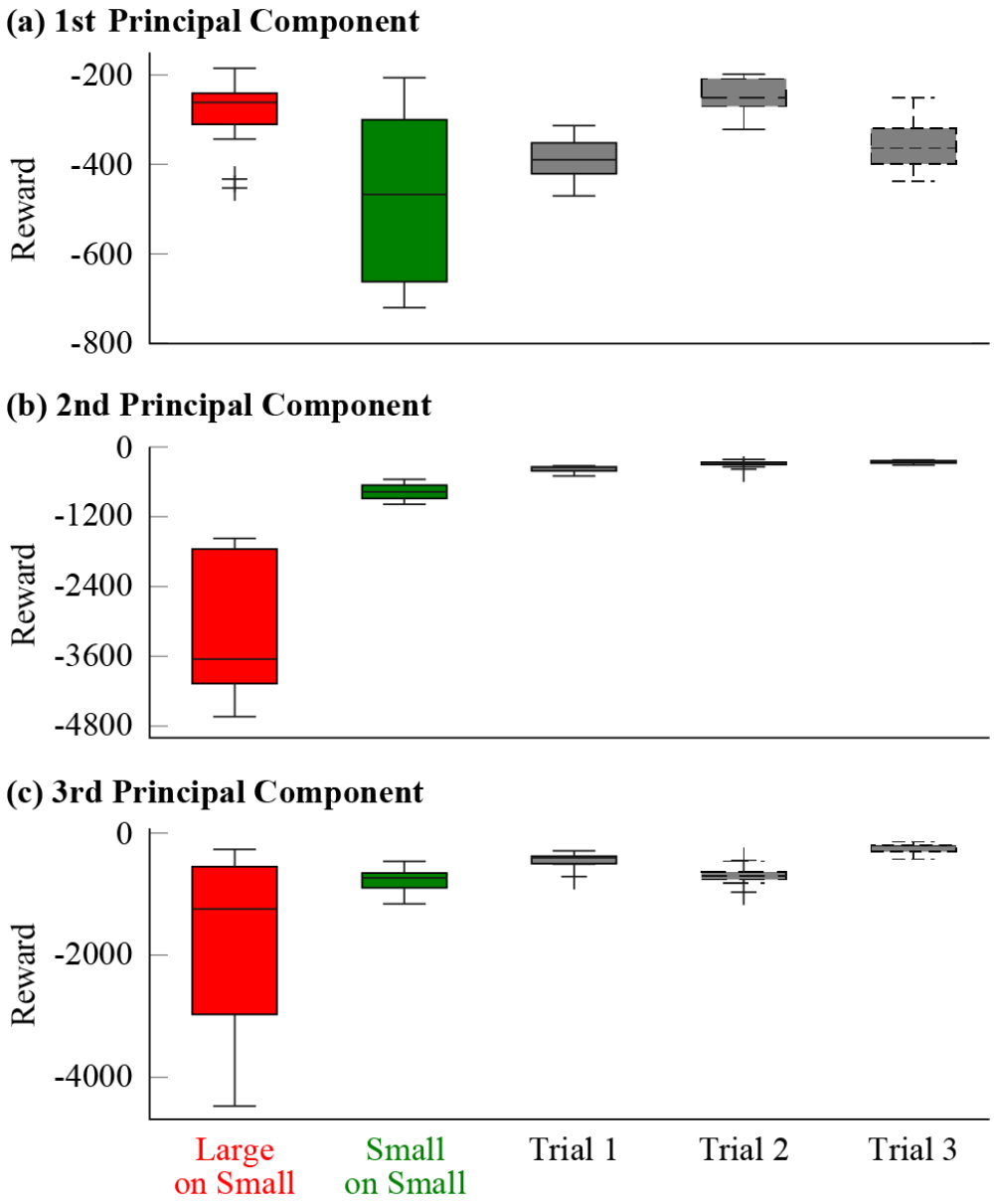

Figure 2.14: Distributions are shown for reward function samples using the PCA eigenvalue representation of tactile sensing traces for the full experiment on adaptation of the learned nominal trajectories to a novel context. Goal position is not known a priori and must be learned. Distributions are shown for a nominal trajectory learned for large pages and applied to small pages prior to adaptation learning (red), for 10 rollouts each. Three independent learning trials (16 policy updates each) show how the distributions change as the initial nominal trajectory for large pages is adapted for small pages after learning. The distribution of reward function samples is also shown for the ideal case of a nominal trajectory learned for small pages and applied to small pages (green).

2.6 Conclusion

With experiments on real robots, we demonstrated a learned manipulation of deformable, thin-shell objects via a page flipping task. We showed that the functional performance of the task can be quantified from the perspective of tactile sensing. We also verified our intuitive assumption that there exist tactile features that can be used to adapt learning to novel task contexts for the manipulation of deformable objects. This insight could facilitate the design of tactile-based controllers for more complex manipulation tasks involving deformable objects and deformable tactile sensors.

2.6.1 Summary of Contributions

In this paper, we demonstrated the ability for a real robot to learn how to manipulate a deformable thin shell and adapt the learned functional behavior to other task contexts. More specifically, we demonstrated that a real robot can learn a page flipping task via tactile information. We addressed the learning of this task using a two-stage approach. For the first learning sub-problem, we learned nominal page flipping trajectories by constructing a reward function that quantifies functional task performance and is driven by tactile feedback. Nominal trajectories were learned specifically for small or large pages using human demonstrations via kinesthetic teaching.

For the second learning sub-problem, we learned adapted trajectories by constructing a reward function that used tactile-driven perceptual coupling. We assumed that, while the page flipping trajectories for different task contexts (page sizes) might differ, similar tactile feedback should be expected from functional trajectories for each of the contexts. We performed a simplified experiment on adaptation of the learned nominal trajectories to a novel context in which the goal position for small pages was known a priori. Using this simplified case, we compared two different representations of tactile sensing traces and concluded that a PCA eigenvalue representation encodes

essential tactile information to enable learning. Finally, we performed a full experiment on adaptation of the learned nominal trajectories to a novel context in which the goal position for small pages had to be additionally learned. We showed that functional behaviors for different task contexts shared features in the tactile feedback that enabled successful learning of adapted trajectories via tactile-driven perceptual coupling.

2.6.2 Limitations and Future Work

One limitation of this work is that, for practical purposes, we reduced the control of the 7-DOF robot arm to three joints such that the page flipping movement would be constrained to a 2-D plane. If all 7 DOFs of the robot were enabled, the learning algorithm might encounter regions of the policy parameter space associated with unnecessarily complex robot motions, such as the twisting of notebook pages through wrist rotation. Defining an effective reward function based on raw tactile sensor data becomes very challenging for such complex scenarios.

Another limitation is that this work does not address additional factors, such as object texture, that might affect the generalizability of a tactile-driven policy. Further investigations are needed to assess the applicability of the "tactile invariance" notion to other tasks and scenarios, when more factors that might affect tactile sensor signals are introduced. Specifically, we believe that one interesting direction is to investigate tactile invariance not only in the context of a specific task, but rather taking it to a higher level of abstraction. For example, many contact manipulation tasks can be decomposed into a sequence of different subtasks (primitives). If they exist, intra-subtask tactile invariances could be used to compose a skill with varying task context, or speed up the learning of a new skill [68, 92].

Another interesting line of investigation is the use of multiple sensing modalities to capture task-relevant features across different task contexts. If low-level representations that encode infor-

mation related to “task invariance” could be extracted from high-dimensional multimodal sensory data, the representations could enable the generalization of learned policies to new task instances more efficiently [53].

CHAPTER 3

End-to-End Model Predictive Control Framework for Tactile-Driven Manipulation of Deformable Linear Objects

Manipulating deformable objects is a common activity in everyday life, yet it remains difficult for robotic systems. Numerous modeling, planning, and control challenges remain for the manipulation of deformable objects by robots endowed with tactile sensors. This work provides a systematic framework for the autonomous, tactile-driven manipulation of deformable linear objects. Our framework leverages advancements in non-parametric supervised learning, model-based reinforcement learning, and optimization-based model predictive control (MPC). We demonstrate the framework for the task of manipulating the pose angle and tensile state of an elastic cable using high-dimensional image data from a deformable, vision-based tactile sensor. A real 7-DOF robot arm and gripper were used to grasp a tethered elastic cable and perform random planar motions in order to collect observation-action pairs. The observation-action pairs were used to develop a low-dimensional latent space dynamics model using a variational non-parametric data-driven approach. The learned latent space dynamics model was used to predict the latent space representations of the high-dimensional tactile sensor data, which encoded the pose and tensile state of the elastic cable's tensile state in response to a control action. Eventually, the dynamics model was embedded in a nonlinear MPC framework for the tactile-driven feedback control of the elastic cable. We demonstrate that a real robot was able to plan trajectories to transition from an initial pose angle

and tensile state to a goal pose angle and tensile state while relying solely on tactile information. Our novel, end-to-end framework has the potential to be scaled to more complex tactile-driven manipulations of deformable linear objects.

3.1 Introduction

Many daily activities, such as unplugging a cellphone charger, tightening a pair of shoelaces, disentangling a rack of cables, etc., necessitate the manipulation of deformable linear objects (DLOs). Yet, the autonomous manipulation of deformable linear objects, such as a cables and ropes, remains challenging for robotic systems. Typically, the manipulation of deformable objects, in general, requires an integrated framework of modelling, perception, planning, and control. Such a requirement undoubtedly introduces compounding challenges and complexity. Therefore, despite its importance, the manipulation of deformable objects has, historically, been investigated less frequently than the manipulation of rigid objects.

Recent achievements in the field of computer graphics have enhanced the feasibility of expensive, nonlinear, numerical simulation of elastic deformable objects. Additionally, progress in reinforcement learning has enabled the planning and control of complex physical interactions between a robotic agent and its environment, either in simulation or directly in the real world environment. Together, these advancements provide valuable modelling techniques and data-driven paradigms that can overcome some of the challenges of modelling, planning and control for the manipulation of deformable objects.

However, in order to implement these modeling and data-driven paradigms, one must operate under a few assumptions and constraints. First, the simulation environment must include a model that is sufficiently accurate for the task of interest [27, 63]. Second, the state of the deformable

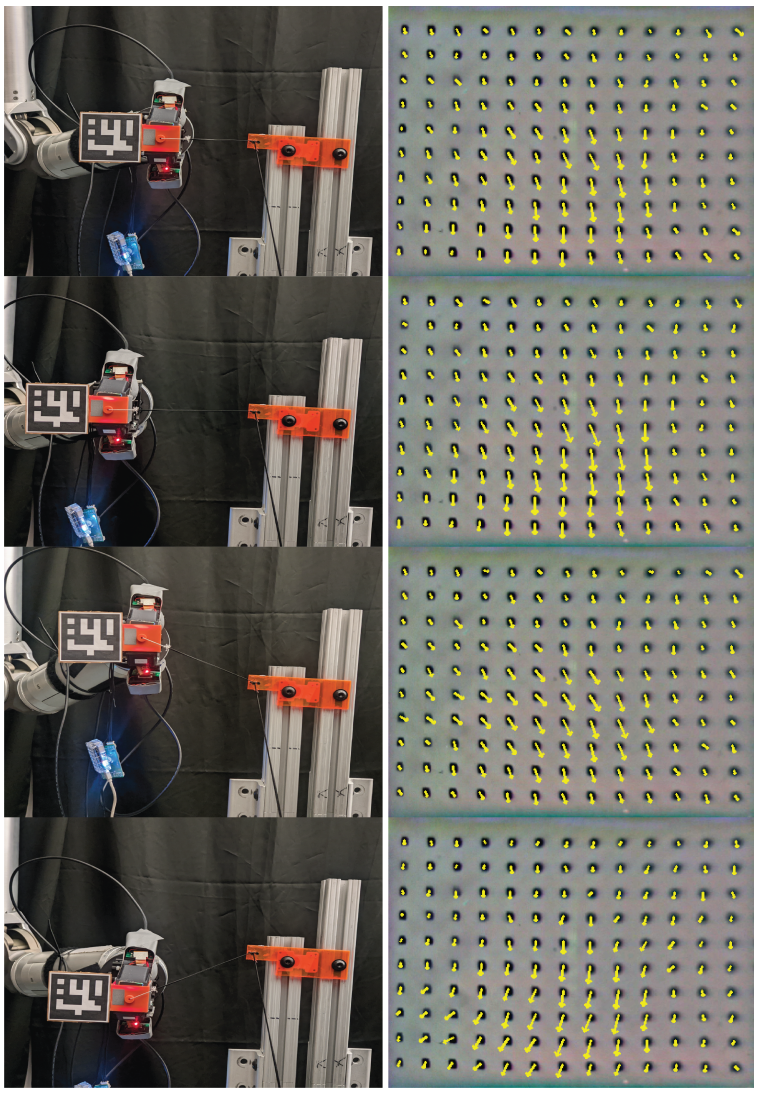

Figure 3.1: A 7-DOF robot arm controls the pose angle and tensile state of an elastic cable using a gripper outfitted with the vision-based GelSight tactile sensor [112]. One end of the tethered cable is manipulated by the gripper through a rigid rig. The left and right columns show different cable states and their tactile representations, respectively.

object must either be directly accessible from the simulation output [126], or observable from real-world sensor data, such as images [34].

Although computer vision has been used to great effect for the manipulation of DLOs, the use of tactile sensors for the manipulation of DLOs is less common in the robotics literature. One likely reason is that vision-dependent approaches have focused on tasks that change the configuration, or pose, state of DLOs. Computer vision has been used to plan and control the manipulation of deformable linear objects for knot-disentangling [94, 109], rope casting [59], and surgical thread suturing [60]. In such cases, tactile sensors can be disadvantageous due to their relatively narrow field-of-view as compared to cameras, or entirely irrelevant, as when the task does not involve the control of the tensile state of the DLO.

A computer vision-based approach is useful when a DLO is slack, but not when the DLO becomes taut and visually straight due to nonzero tension. In order to simulate changes in the tensile state of a DLO, one needs a high-fidelity simulation of the DLO as well as the physical interactions between the DLO and tactile sensors. The simulation of deformable tactile sensors is a highly challenging research topic by itself, and state-of-the-art models of deformable tactile sensors are not easily integrated with state-of-the-art planning and control pipelines. In this work, we leverage tactile sensors to address the visual ambiguity of a taut DLO in order to perceive pose angle and tensile state simultaneously.

The ultimate goal of this work is to develop a principled optimization-based model predictive control (MPC) framework that enables the robotic manipulation of deformable linear objects (DLOs) using tactile data. Our proposed framework consists of three components: (i) a deep representation model that maps high-dimensional tactile sensor observations related to pose and tensile states to a compact, low-dimensional, latent space, (ii) a supervised learning model to encode tactile-related transition dynamics in the compact latent space with sufficient representational

power, and (iii) an optimization-based MPC framework that uses the learned dynamics model in latent space in order to perform motion planning and control, for which an off-the-shelf nonlinear optimization solver can be leveraged.

To the best of our knowledge, this work is the first to drive an optimization-based model predictive controller on a real robot using a system dynamics model identified directly from high-dimensional tactile observations. This work addresses the following observations from the robotics literature:

- While prior research has traditionally focused on the control of DLO configuration only [118, 94, 109], we aim to control simultaneously the pose angle and tensile state of a DLO.
- While "end-to-end" reinforcement learning has been accomplished using vision-based approaches [57, 35, 36], there are few works that use tactile data as inputs to "end-to-end" learning frameworks and further demonstrate the results for a real robot manipulation task. Our proposed framework has an "end-to-end" nature because high-dimensional tactile data are used as inputs to plan robot behaviors that control the pose and tensile state of a DLO.
- Our proposed approach bridges the tactile-driven manipulation community with other robotics communities where well-established technical tools, such as MPC, are available. Such a bridge is beneficial for advancing the use of tactile data in robot manipulation applications.

The remainder of the manuscript is organized as follows: Section 3.2 outlines related work. Section 3.3 introduces our proposed framework and individual technical components. Section 3.4 describes the development and deployment of the framework on a real robot. Section 3.5 summarizes the limitations, novel contributions, and future directions of this work.

3.2 Related Work

In this section, we introduce related work and distinguish our approach from prior studies for the following topics: model-based reinforcement learning, the robotic manipulation of deformable linear objects, and the simulation of tactile sensor behavior.

3.2.1 Model-based Reinforcement Learning

Reinforcement learning methods have been successfully demonstrated for a number of robotics applications either on a real system or in simulation, such as dynamic quadruped locomotion across non-homogeneous terrains [20, 65, 52, 76], in-hand manipulation to solve Rubik's Cubes [69], aggressive maneuvers of simulated race-car driving [31, 89] and agile drone flights [90]. Many of these results were achieved using model-free deep reinforcement learning, in which the agent learns a policy function represented by a neural network through direct interactions with the environment.

Traditionally, the samples of agent-environment interactions often rely heavily on computer vision data, even in the case of the in-hand manipulation of Rubik's Cubes [69]. For the tactile-driven manipulation of objects, the complex contact dynamics between deformable objects and deformable tactile sensors can render the construction of physics-based simulations infeasible. It can also be impractical to collect large numbers of samples of agent-environment interactions for the tactile sensing modality.

In model-based reinforcement learning, the agent infers a predictive model of the world from sensor data obtained via interaction with real environment and then uses the predictive model for planning and control [74]. Model-based approaches offer some potential advantages over model-free approaches. For example, instead of relying solely on propagated rewards from repetitive

rollouts, richer signals (i.e. sequential dependencies) from the learned transition model can be leveraged to increase sampling efficiency.

Importantly, the use of model-based approaches can bridge reinforcement learning methods with state-of-the-art technical tools for planning and control. In turn, such a bridge could enable the principled treatment of properties such as state or input constraints and system stability, which cannot currently be addressed directly or can only be represented as penalty terms in reward functions when using model-free methods [19, 103]. Some studies [115, 4, 125] proposed locally linear models for transitions in the latent space and planned for actions using linear quadratic regulators. However, the locally linear formulation limits the application of this approach to tasks with relatively well-defined dynamics because of a lack of representative power. For our purposes, well-defined contact dynamics for physical interactions between DLOs and deformable tactile sensors are not yet established.

More recent works [35, 9, 36] propose to learn non-local nonlinear dynamics models such that the asymptotic performance of model-based methods can match model-free exemplars in complex image-based domains. Our work takes inspiration from [35, 9, 36] in order to develop a variational, latent space dynamics learning method such that the latent space dynamics generated from high-dimensional, noisy tactile sensor data from real robot experiments can be captured and further incorporated into an optimization-based MPC module.

Both [35] and [36] use convolutional neural networks to construct latent space models from raw RGB camera images, and then use recurrent neural networks to encode the transition dynamics in the latent space. While this structure offers a rich representation from which to learn, the complex structure cannot be directly incorporated into an optimization-based MPC framework. Moreover, in [35] and [36], the learned latent space dynamics model is not used directly for planning. Rather, a different reward prediction network is used that takes the latent state as input. On a real robot,

it may be unrealistic to learn different reward prediction networks for different scenarios, such as different initial and goal conditions for a tactile-driven manipulation task. Thus, in this work, we aimed to learn a transition dynamics model that could be used directly for planning.

In [9], the state transition dynamics in latent space and the reward prediction model are modelled through Gaussian processes (GPs) instead of neural networks. Nonetheless, as with [35, 36], the learned latent space dynamics are not used directly to perform planning. In order to incorporate GPs into a variational inference framework, [9] treats each observation-action pair from the sequence as independent and identically distributed instead of taking sequential relationships into consideration, and accordingly formulates the training objective based on probabilistic generative modelling principles.

In preliminary studies, when we applied the approaches of [35, 9, 36] to our tactile sensor data, we observed that neural networks and GPs could easily overfit observations and did not sufficiently capture latent space dynamics for planning. We suspect that the tactile sensor from real robot experiments were noisier and less Markovian as compared to sensor data from a structured, one-dimensional pendulum swing-up scenario [9]. As such, our work considers sequential relationships in the data, as will be discussed in Section 3.4. Finally, while evaluations were performed in simulation environments in [35, 9, 36], we evaluated our proposed framework on a real robot platform.

3.2.2 Robotic Manipulation of Deformable Linear Objects

The robotic manipulation of deformable linear objects is an active area of research. State-of-the-art approaches to the manipulation of deformable linear objects currently rely predominantly on computer vision [119]. In [59], a self-supervised framework is proposed to tune differentiable simulators for dynamic rope casting behavior using camera data collected from real robot inter-

actions, specifically the endpoint of the rope as obtained from visual inspection. In [94], a robust knot untangling system was developed, in which constant visual monitoring and refinement according to RGB-D images were key components. In [118], a vision-based system was developed that enabled a robot to manipulate a deformable linear object until it matched a goal state that was represented visually using segments inferred from top-down images of the rope configuration.

In contrast, the application of tactile sensing for the manipulation of deformable linear objects is less common. Most works related to the robotic manipulation of DLOs use tactile sensing for perception, but not for planning or control. Examples of tactile perception include shape estimation [79], texture classification [124, 37], and the estimation of a physical properties, such as those unique to different liquids in a container [41]. There are a few exceptions in which tactile sensing is used for the manipulation of deformable objects. In [85], a system was developed to perform a planar cable following task using a linear quadratic regulator whose linear model coefficients were inferred directly from camera-based tactile sensor data about cable state. The prior work was extended to a cloth unfolding task in [95], where the folding state of a cloth garment was first classified via vision-based tactile sensor data, and then a hierarchical state-machine performed motion primitives based on the classified state.

This work differs from [85, 95] significantly when considering the information that is gleaned from the camera-based tactile sensor data. For the tasks of cable following [85] and cloth unfolding [95], intuitive, salient, local geometric features about pose state can be obtained directly from tactile images (e.g. position and orientation of a cable or cloth edge). As a result, interactions between the robot fingerpad and target object can be considered linear time invariant, and linear quadratic regulators can be used for system identification, planning, and control. Similar to [85, 95], this work results in tactile consequences that are geometrically aligned with actual robot motion. However, [85, 95] did not consider tensile state in addition to pose. Additionally, we

aim to develop a framework that has the potential to be extended to more complex manipulations of deformable objects, without relying on the experimental assumption that local features can be easily and directly obtained from high-dimensional tactile data.

The study presented in [100] is most closely related to this work, but involves the manipulation of rigid objects instead of deformable objects. The authors used camera-based tactile sensor data from real robot-object interactions and a complex neural network in order to build a model of system dynamics in the form of an image sequence prediction network. Then an evolutionary strategy method [8] was used for planning control actions for manipulation tasks such as positioning a ball bearing. In this work, we conducted an investigation of system dynamics in a low-dimensional latent space, and we integrated the latent space dynamics model with an optimization-based model predictive control framework, whose parameters could be extended for future works.

3.2.3 Simulation of Tactile Sensor Behavior

Innovations in tactile sensor design continue, with recent designs leveraging marker-tracking with a low-cost camera [56], tracking of dense optical flows [82], and measurement of electrode voltage changes within a conductive fluid [54]. While tactile sensor technology and data analysis techniques advance, the development of task-relevant models of tactile sensor behavior remains a challenge. The difficulty of developing tactile sensor models from first-principle physics or by collecting tactile data via real robot-environment interactions negates the widespread use of sim-to-real paradigms and well-established model-based planning and control methods for tactile-driven manipulation. In order to leverage tactile sensor data for planning and control, one needs a model that captures and predicts how a given robot action directly changes the state of the tactile sensor as well as the state of the object. The dynamics model can be difficult to build when both the tactile sensor and object are deformable, and especially when multiphysics modeling is required.

Progress is being made in the important area of tactile sensor modeling and simulation. In [113], a simulator for a vision-based tactile sensor was developed based on RGB-D image rendering heuristics [86]. The simulator was demonstrated for the task of manipulating rigid objects having simple shapes (e.g. cube, sphere) in the plane with homogeneous contact geometry. However, in this work, contact between the deformable tactile sensor fingerpad and elastic cable is affected by both the pose and tensile state of the cable.

In [87], the authors employed a linear displacement relationship and superposition principle in order to compute marker motion for an elastomeric, camera-based tactile sensor (GelSight). The linear superposition principle requires that the initial displacement of the fingerpad caused by external loads be known a priori. In this work, such displacement information is not known a priori.

In [81], finite element model simulations based on hyperelastic material models and the ideal pinhole camera model were used to generate a supervised learning dataset. The goal of the simulator was to estimate the distribution of contact force across an elastomeric, camera-based tactile sensor from tactile images acquired during real robot-object interactions. The construction of the dataset required a regularized indentation process such that the contact force distribution and node displacement field could be extracted at each point along the indentation trajectory. In this work, physical interactions with the elastic cable were relayed to the elastomeric tactile sensor fingerpad through the grasp of a rigid rig and not through direct indentation of the cable into the fingerpad. Furthermore, both normal and shear forces from manipulations of the elastic cable affected the tactile sensor data.

3.3 Framework for the Tactile-driven Model Predictive Control of Deformable Linear Objects

In this work, we aim to develop a principled optimization-based model predictive control framework for the tactile-driven manipulation of deformable linear objects. In order to realize such a goal we need to overcome two challenges: (i) How to identify a system dynamics model that accurately captures the effects of robot control actions on high-dimensional tactile sensor data; (ii) How to ensure that the system dynamics model is computationally feasible for integration with an optimization-based MPC framework.

In order to tackle these two challenges, we employed a combination of unsupervised representation learning and supervised learning. First, we used deep probabilistic generative modeling techniques to allow for the automatic discovery of dynamics in a low-dimensional latent space from high-dimensional observation-action pairs. Second, the generative neural network was used to guide the training of a Gaussian process (GP) model. With proper approximation techniques, the GP model was made computationally feasible for integration with an optimization-based MPC framework for planning and control on a real robot.

We briefly introduce the problem formulation at the highest level here. At a time step t we denote the high-dimensional raw tactile sensor observation as $\mathbf{o}_t$, the control action $\mathbf{u}_t$ and the underlying low-dimensional latent state as $\mathbf{x}_t$. The parameterized inference function that estimates the low-dimensional latent state is $q_\Theta(\mathbf{x}_t|\mathbf{o}_t)$, with Θ representing the weights and bias of the neural networks. The dynamics model describing the change of low-dimensional latent state caused by a control action is identified and denoted as $\mathbf{x_{t+1}} = f(\mathbf{x_t}, \mathbf{u_t}|\Theta)$, where the conditioning symbol is used to emphasize that the latent space dynamics model depends on the inference function $\mathbf{q}$.

Denoting the reference trajectory in the latent space as $\mathbf{X_r} = \{\mathbf{x}_0, \ldots, \mathbf{x}_N\}$, the general MPC

formulation of an optimal control problem is expressed as the following:

$$\min_{\Pi(\mathbf{x})} \quad \mathbb{E}\left(l_f\left(\mathbf{x}_N - \mathbf{x}_N^r\right) + \sum_{t=0}^{N-1} l\left(\mathbf{x}_i - \mathbf{x}_i^r, \mathbf{u}_i\right) \right) \tag{3.1a}$$

$$\text{s.t.} \quad \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i|\Theta), \tag{3.1b}$$

$$\mathbf{u}_i = \pi_i(\mathbf{x}_i), \tag{3.1c}$$

$$Pr(\mathbf{u}_i \in \mathscr{U}) \geq p_u, \tag{3.1d}$$

$$\mathbf{x}_0 = \mathbf{q}(\mathbf{o}_0) \tag{3.1e}$$

The optimization is carried out over a sequence of input policies $\Pi(\mathbf{X}) = \{\pi_0(\mathbf{x}), \ldots, \pi_N(\mathbf{x})\}$ with terminal loss $l_f\left(\mathbf{x}_N - \mathbf{x}_N^r\right)$ and intermediate loss $l\left(\mathbf{x}_i - \mathbf{x}_i^r, \mathbf{u}_i\right)$, where N is the length of the planning window. A stochastic optimization framework also allows the treatment of chance constraints Eqn. Equation (3.1d) with principled approaches, which is a prescription of maximum probability constraint violation. At the start of the optimization process we obtain the initial latent state $\mathbf{x}_0$ from the current tactile observation $\mathbf{o}_0$, and the optimization is carried out with the latent space dynamics $\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i|\Theta)$. It should be noted that the unsupervised representation learning model is not used directly in the optimization process except to provide the initial latent state. Nonetheless, the model is still important to the optimization process because the quality of the latent space dynamics function f heavily depends on the model.

In this section, we provide background on Gaussian process models (Section 3.3.1) and an approximation technique (Section 3.3.2) that make GP models practical for an optimization-based MPC framework. We present how to replace the original MPC formulation in Eqn. Equation (3.1) with a tractable approximation (Section 3.3.3). We then introduce the probabilistic generative modeling principles in Section 3.3.4) and the unsupervised representation learning methods explored in this work (Sections 3.3.4.1 and 3.3.4.2).

3.3.1 Gaussian Process Models

Gaussian process models provide a data-efficient framework for nonlinear regression problems. A GP is a collection of random variables, such that any finite number of which have a joint Gaussian distribution [75]. For a dynamical system, we denote the collected state-action triples as $\{\mathbf{x}_{i_i+1}, \mathbf{x}_{t_i}, \mathbf{u}_{t_i}\}_{i=1,\ldots,M}$, Hence, the training dataset for a GP model is

$$\mathscr{D} := \{\mathbf{X} := [(\mathbf{x}_{t_1}, \mathbf{u}_{t_1})^\intercal, \ldots, (\mathbf{x}_{t_M}, \mathbf{x}_{t_M})^\intercal] \in \mathbb{R}^{M\times(n_x+n_a)},$$
$$\mathbf{Y} := [\mathbf{x}_{t_1+1}^\intercal, \ldots, \mathbf{x}_{t_M+1}^\intercal] \in \mathbb{R}^{M\times n_x}\}$$

For the state dimension $n_x > 1$, given the input data pair $(\mathbf{x}_t, \mathbf{u}_t)$, we assume that each dimension of GP output is conditionally independent. A Gaussian process models the output data using a joint normal distribution. For each output dimension $l \in \{1, \ldots, n_x\}$, a prior mean function $m^l(\cdot)$ and prior pair-wise covariance function $k^l(\cdot,\cdot)$ are specified, which results in the following joint normal distribution,

$$[\mathbf{Y}]_{\cdot,l} \sim N(m^l(\mathbf{X}), K^l_{\mathbf{XX}} + I\sigma^2_{n_l}) \tag{3.2}$$

where $K^l_{\mathbf{XX}}$ is the covariance matrix of the data. More specifically, suppose $\hat{\mathbf{x}}_t = (\mathbf{x}_t, \mathbf{u}_t)^\intercal$, we have $m^l(\mathbf{X}) = [m^l(\hat{\mathbf{x}}_{t_1}), \ldots, m^l(\hat{\mathbf{x}}_{t_M})]$ and $[K^l_{\mathbf{XX}}]_{ij} = k^l(\hat{\mathbf{x}}_{t_i}, \hat{\mathbf{x}}_{t_j})$. In practice, the prior mean function $m^l(\cdot)$ is often set to zero and the kernel function captures the features of the data. In this work, we chose the commonly used squared exponential kernel,

$$k^l(\hat{\mathbf{x}}_{t_i}, \hat{\mathbf{x}}_{t_j}) = \sigma^2_{f_l} \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}_{t_i} - \hat{\mathbf{x}}_{t_j})^\intercal W_l (\hat{\mathbf{x}}_{t_i} - \hat{\mathbf{x}}_{t_j})\right) + \delta_{ij}\sigma^2_{n_l} \tag{3.3}$$

where W_l is the diagonal length scale matrix determining how relevant each input dimension l is with respect to the output data, $\sigma^2_{f_l}$ is the signal variance, and $\sigma^2_{n_l}$ is the noise variance. As such, the hyperparameters are $(\{W_l\}, \sigma^2_{f_l}, \sigma^2_{n_l})$. Optimization of the hyperparameters are achieved through Bayesian inference with the log marginal likelihood as the evidence [75].

For prediction, the joint distribution of training data and a test data point $\mathbf{x}_*$ in an output dimension l can be written as

$$\begin{bmatrix} [\mathbf{Y}]_{\cdot,l} \\ [\mathbf{y}_*]_{\cdot,l} \end{bmatrix} \sim N\left(\begin{bmatrix} m^l(\mathbf{X}) \\ m^l(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} K^l_{\mathbf{XX}} + I\sigma^2_{n_l} & K^l_{\mathbf{X}\mathbf{x}_*} \\ K^l_{\mathbf{x}_*\mathbf{X}} & K^l_{\mathbf{x}_*\mathbf{x}_*} \end{bmatrix} \right) \tag{3.4}$$

Given the training inputs, the predicted output of the test data in dimension l is expressed with the resultant conditional distribution $p([\mathbf{y}_*]_{\cdot,l}|[\mathbf{Y}]_{\cdot,l},\mathbf{X}) = N(\mu_l(\mathbf{x}_*),\Sigma_l(\mathbf{x}_*))$

$$\mu_l(\mathbf{x}_*) = K^l_{\mathbf{x}_*\mathbf{X}}(K^l_{\mathbf{XX}} + I\sigma^2_{n_l})^{-1}[\mathbf{Y}]_{\cdot,l} \tag{3.5a}$$

$$\Sigma_l(\mathbf{x}_*) = K^l_{\mathbf{x}_*\mathbf{x}_*} - K^l_{\mathbf{x}_*\mathbf{X}}(K^l_{\mathbf{XX}} + I\sigma^2_{n_l})^{-1}K^l_{\mathbf{X}\mathbf{x}_*} \tag{3.5b}$$

The final multivariate GP prediction for the output is expressed by stacking the mean and covariance functions for each individual output dimension, as follows,

$$\Psi = [\mu_1(\mathbf{x}_*),\ldots,\mu_M(\mathbf{x}_*)], \tag{3.6a}$$

$$\Omega = diag(\Sigma_1(\mathbf{x}_*),\ldots,\Sigma_M(\mathbf{x}_*)) \tag{3.6b}$$

For the above, full GP model, the complexity of evaluating the mean and covariance in Eqn. Equation (3.5) are $\mathscr{O}(n^2)$ and $\mathscr{O}(n^3)$ respectively, and it is, in principle, to incorporate all training points for the evaluations in Eqn. Equation (3.5). Unfortunately, for the large amounts of data required for MPC on real robots, such complexity limits the use of GP models for our application. A second issue with the full, dense GP model is that the predictive distributions in Eqn. Equation (3.5) are derived with deterministic inputs. In order to fully align with the MPC framework and transcribe multi-step predictions from learned dynamics, it is necessary to consider stochastic (tactile) inputs. We will now briefly outline techniques for solving both issues.

3.3.1.1 Stochastic Variational Gaussian Process Models

The marginal likelihood of the full dense GP does not attain a mini-batch formulation due to the $\mathcal{O}(n^3)$ complexity. To alleviate this issue, a set of inducing points are introduced. The inducing points jointly form the GP prior with the full training dataset and the test data point. The motivation is that the information from the full training dataset can be effectively summarized by the inducing dataset. The marginal likelihood then factorizes into:

$$p(y) = \int p(y|\mathbf{X})p(\mathbf{X}|\mathbf{x}_m)p(\mathbf{x}_m)d\mathbf{X}d\mathbf{x}_m \tag{3.7}$$

We can apply the variational approximation principle, which is further described in Section 3.3.4, and notice that if the variational approximation distribution is chosen as $q(\mathbf{X}, \mathbf{x}_m) = p(\mathbf{X}|\mathbf{x}_m)q(\mathbf{x}_m)$, the conditional probability $p(\mathbf{X}|\mathbf{x}_m)$ naturally cancels out in the derivation of the evidence lower bound (ELBO $\mathbb{L}$):

$$\begin{aligned}\mathbb{E}_{q(\mathbf{X},\mathbf{x}_m)}[\log p(\mathbf{X})] = &\underbrace{\mathbb{E}_{q(\mathbf{X},\mathbf{x}_m)}[\log \frac{p(y|\mathbf{X})p(\mathbf{x}_m)}{q(\mathbf{x}_m)}]}_{ELBO} + \\ &D_{KL}[q(\mathbf{X}, \mathbf{x}_m)||p(\mathbf{X}, \mathbf{x}_m|y)]\end{aligned} \tag{3.8}$$

Importantly, due to the expectation formulation, we can now use stochastic gradient descent with mini-batch sampling to train the GP model. After the maximization of the ELBO, predictive distributions similar to Eqn. Equation (3.5) can be obtained from the joint prior. Now, the dominant computational expense depends on the m inducing points $\mathbf{x}_m$, and the evaluation complexity becomes $\mathcal{O}(nm^2)$, which provides a significant speed-up when $m \ll n$. In this work, we use the stochastic variational Gaussian process (SVGP) model in preparation for eventual integration with an optimization MPC framework. For additional details regarding SVGP models, we refer the reader to [64, 101].

3.3.2 Uncertainty Propagation

As stated in Section 3.3.1, inputs to GP models are assumed to be deterministic. As a result, the Gaussian output distribution is useful for predicting one timestep ahead. However, since the MPC framework in the latent space normally requires predicting several timesteps ahead, a stochastic output from one timestep will be used as the (stochastic) input for the next prediction.

Suppose the joint distribution for the state-action pair at a specific timestamp t is Gaussian, i.e. $p(\mathbf{s_t}, \mathbf{a_t}) = N(\tilde{\mu}, \tilde{\Sigma})$, then the probabilistic state transition function will have the form

$$p(f(\mathbf{s_t}, \mathbf{a_t})|\tilde{\mu}, \tilde{\Sigma}) = \int p(f(\mathbf{s_t}, \mathbf{a_t})|\mathbf{s_t}, \mathbf{a_t}) p(\mathbf{s_t}, \mathbf{a_t}) d\mathbf{s_t} d\mathbf{a_t} \tag{3.9}$$

This represents an integration of a nonlinear mapping of Gaussian-distributed random variables to the GP prediction mean and covariance with Eqn. Equation (3.5a) and Equation (3.5b), which, in general, is non-Gaussian and cannot be computed analytically. To make the learned GP dynamics tractable for an optimization-based MPC framework, a first-order Taylor approximation is applied to the GP predictive distribution with the uncertain input:

$$\mu_{pred} = \mu(\tilde{\mu}) \tag{3.10a}$$

$$\Sigma_{pred} = \Sigma(\tilde{\mu}) + \nabla(\mu_{pred}) \tilde{\Sigma} \nabla(\mu_{pred})^{\mathsf{T}} \tag{3.10b}$$

For further details, readers are referred to [16, 24].

A more complicated method that can capture the uncertainty propagation more accurately is also proposed in [24]. For this work, we find that the Taylor approximation offers a reasonable trade-off between computation speed and representation accuracy of the transition dynamics. However, for tactile sensors whose noise can be better characterized and/or for more high-consequence manipulation scenarios, the selection of the uncertainty propagation method could be significant and a more complicated method such as in [24] should be considered.

3.3.3 Tractable Model Predictive Control Formulation

With the tools and approximations introduced above, we can replace the original MPC formulation in Eqn. Equation (3.1) with a tractable MPC formulation. First, we restrict the controller class to be a linear state feedback controller. This allows us to avoid the infinite dimensional optimization problem of searching over a general form of feedback control policy $\pi_i(\mathbf{x})$ at each planning step.

$$\mathbf{u}_i = \mathbf{K}(\mu_i^{\mathbf{x}} - \mathbf{x}_i^r) + \mathbf{v}_i \tag{3.11}$$

The feedback gain $\mathbf{K}$ is selected heuristically and the optimization is conducted with respect to the feedforward component $\mathbf{v}_i$. With the control law in Eqn. 3.11, the mean and covariance of the stochastic input to the GP in Eqn. Equation (3.9) can expressed as

$$\tilde{\mathbf{u}} = [\mu_i^{\mathbf{x}}, \mathbf{K}\mu_i^{\mathbf{x}}]^\top, \quad \tilde{\Sigma} = \begin{bmatrix} \Sigma_i^{\mathbf{x}} & \Sigma_i^{\mathbf{x}}\mathbf{K}^\top \\ \mathbf{K}\Sigma_i^{\mathbf{x}} & \mathbf{K}\Sigma_i^{\mathbf{x}}\mathbf{K}^\top \end{bmatrix} \tag{3.12}$$

The first order Taylor approximation in Eqn. Equation (3.10) can be applied accordingly.

The task we consider in this work is analogous to a reaching task that involves an initial state and goal state. Thus, the cost function is chosen to be the quadratic distance of every random state on the trajectory with respect to the goal point $\mathbf{x}_g$

$$\mathbb{E}(l(\mathbf{x}_i, \mathbf{x}_g)) = ||\mu_i^{\mathbf{x}} - \mathbf{x}_g||_Q^2 + \mathbf{tr}[Q\Sigma_i^{\mathbf{x}}] \tag{3.13}$$

We also bound the control signal through chance constraints in order to demonstrate the full flexibility of the framework, similar to [38]. With an application of Boole's inequality [6], a probabilistic polytopic constraint on a Gaussian random variable $Pr(\mathbf{u}_1 \leq \mathbf{K}\mathbf{x} + \mathbf{v} \leq \mathbf{u}_2) \geq \delta$ can be made deterministic, as follows,

$$\begin{aligned} \mathbf{K}\mu_{\mathbf{x}} + \mathbf{v} &\leq \mathbf{u}_2 - \sqrt{2\mathbf{K}^\top\Sigma_{\mathbf{x}}\mathbf{K}}\Phi^{-1}(\delta) \\ \mathbf{K}\mu_{\mathbf{x}} + \mathbf{v} &\geq \mathbf{u}_1 + \sqrt{2\mathbf{K}^\top\Sigma_{\mathbf{x}}\mathbf{K}}\Phi^{-1}(\delta) \end{aligned} \tag{3.14}$$

where Φ^{-1} denotes the inverse quantile function of a standard univariate Gaussian distribution.

Eventually we obtain a tractable MPC formulation with the multiple shooting scheme [7]

$$
\begin{aligned}
\min_{\mathbf{v}_{0:N-1},\mu^{\mathbf{x}}_{0:N},\Sigma^{\mathbf{x}}_{0:N}} \quad & \sum_{i=0}^{N-1} \left(||\mu^{\mathbf{x}}_i - \mathbf{x}_g||^2_Q + \mathbf{tr}[Q\Sigma^{\mathbf{x}}_i] \right) \\
\text{s.t.} \quad & \mu^{\mathbf{x}}_{i+1} = \mu_{pred}(\mu^{\mathbf{x}}_i, \mathbf{u}_i), \\
& \Sigma^{\mathbf{x}}_{i+1} = \Sigma_{pred}(\mu^{\mathbf{x}}_i, \Sigma^{\mathbf{x}}_i, \mathbf{u}_i), \\
& \mathbf{u}_i = \mathbf{K}\mu^{\mathbf{x}}_i + \mathbf{v}_i, \\
& Pr(\mathbf{u}_i \in \mathscr{U}) \geq p_u, \ \textit{according to Equation}(3.14) \\
& \mu^{\mathbf{x}}_0 = q(\mathbf{o}_0), \ \Sigma^{\mathbf{x}}_0 = 0
\end{aligned}
\tag{3.15}
$$

3.3.4 Unsupervised Learning of Dynamics Using Probabilistic Generative Modeling

In order to perform MPC on a real robot for our tactile-driven manipulation task, we need a system dynamics model describing how the state of the elastic cable (as encoded by high-dimensional tactile sensor data) changes due to control actions. In turn, to make the dynamics model computationally feasible for MPC, the model needs to be represented in a low-dimensional latent space instead of the high-dimensional tactile sensor data space. This requirement naturally falls within the class of problems necessitating unsupervised feature extraction, which is commonly tackled with probabilistic generative modeling [48].

In probabilistic generative modeling we assume a set of observed variables $\mathbf{O} = \{\mathbf{o}\}_i$ is independent and identically distributed (i.i.d.) and generated from an unknown random process involving an unobserved random variable $\mathbf{x}$. Probabilistically, the process factorizes into two steps: (1) A value of $\mathbf{x}$ is sampled from a prior distribution $p_\theta(\mathbf{x})$; (2) A value of the observation is sampled from the conditional distribution $p_\theta(\mathbf{o}|\mathbf{x})$. For applications with high-dimensional data, such as

image and speech waveforms, neural networks are commonly used for modelling the density functions, where both the unobserved random variable $\mathbf{x}$ and neural network parameters θ are unknown to us.

The learning objective of probabilistic general modeling is to search over parameter θ such that the modelled marginal data distribution $\int p_\theta(\mathbf{o}|\mathbf{x})p_\theta(\mathbf{x}) = p_\theta(\mathbf{o})$ matches the unknown true data distribution. However, in general, directly computing the marginal probability distribution through integration is non-analytical. In order to circumvent this problem and to develop an inference scheme for the marginal distribution, Bayes rule is applied. Applying Bayes rule to the marginal distribution $p(\mathbf{o}) = \frac{p(\mathbf{o},\mathbf{x})}{p(\mathbf{x}|\mathbf{o})}$, introducing another auxiliary distribution $q(\mathbf{x})$, and noting that $\log p(\mathbf{o}) = \mathbb{E}_{q(\mathbf{x})}[p(\mathbf{o},\mathbf{x})]$, the log marginal likelihood can be rewritten as:

$$\begin{aligned} \log p_\theta(\mathbf{x}) = D_{KL}(q(\mathbf{x})||p(\mathbf{x}|\mathbf{o})) + \mathbb{E}_{q(\mathbf{x})}[p(\mathbf{o}|\mathbf{x})] \\ - D_{KL}(q(\mathbf{x})||p(\mathbf{x})) \end{aligned} \tag{3.16}$$

It should be noted that the generative distribution is modelled using a neural network with parameter θ. If the auxiliary function is modelled to be dependent upon the observation $\mathbf{o}$ using a neural network with parameter ϕ, similar to the generative distribution, the first term on the right-hand side of Eqn. Equation (3.16) becomes the Kullback-Leibler (KL) divergence of the auxiliary posterior from the true posterior. The remaining terms on the right-hand side of Eqn. Equation (3.16) are called the evidence lower bound (ELBO), which consists of the reconstruction error of the observation from the encoder-decoder structure and the KL divergence of the encoder from the unobserved state prior. Maximization of the ELBO with respect to θ and ϕ will approximately maximize the marginal observation distribution $p(\mathbf{o})$ and minimize the KL divergence of the auxiliary posterior from the true posterior.

The generative inference framework described above is "static" in the sense that observations are assumed to be i.i.d. However, tactile sensor data generated from task-specific action sequences

on a real robot are likely to be temporally correlated. Such temporal relationships may need to be considered when developing an inference framework for sequential data.

Consider a sequence of random observed variables $\mathbf{o}_{1:T} = \{\mathbf{o}_t \in \mathbb{R}^{n_o}\}$, unobserved variables $\mathbf{x}_{1:T} = \{\mathbf{x}_t \in \mathbb{R}^{n_x}\}$, and control sequence $\mathbf{u}_{1:T} = \{\mathbf{u}_t \in \mathbb{R}^{n_u}\}$. For control-related tasks, we are interested in the generative process of $\mathbf{o}_{1:T}$ and $\mathbf{x}_{1:T}$ given control action $\mathbf{u}_{1:T}$. We assume that $\mathbf{u}_{1:T}$ is deterministic and, therefore, only focus on modelling the conditional marginal probability $\int p(\mathbf{o}_{1:T}, \mathbf{x}_{1:T}|\mathbf{u}_{1:T}) p_\theta(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) = p(\mathbf{o}_{1:T}|\mathbf{u}_{1:T})$.

After imposing a causal relationship and applying the chain rule, the conditional joint distribution can be factored as follows,

$$p(\mathbf{o}_{1:T}, \mathbf{x}_{1:T}|\mathbf{u}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{o}_t|\mathbf{o}_{1:t-1}, \mathbf{x}_{1:t}, \mathbf{u}_{1:t}) \\ p(\mathbf{x}_t|\mathbf{o}_{1:t-1}, \mathbf{x}_{1:t-1}, \mathbf{u}_{1:t}) \tag{3.17}$$

Without any further assumptions, the exact posterior is non-causal and attains the following general form,

$$p(\mathbf{x}_{1:T}|\mathbf{o}_{1:T}, \mathbf{u}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{o}_{1:T}, \mathbf{u}_{1:T}) \tag{3.18}$$

The formulation of the inference model $q(\mathbf{x}_{1:T})$ should also maintain the structure of the exact posterior:

$$q(\mathbf{x}_{1:T}|\mathbf{o}_{1:T}, \mathbf{u}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{o}_{1:T}, \mathbf{u}_{1:T}) \tag{3.19}$$

After applying the formulation in Eqns. Equation (3.17), Equation (3.18) and Equation (3.19) to the log marginal decomposition Eqn. Equation (3.16), we obtain the ELBO for a sequential

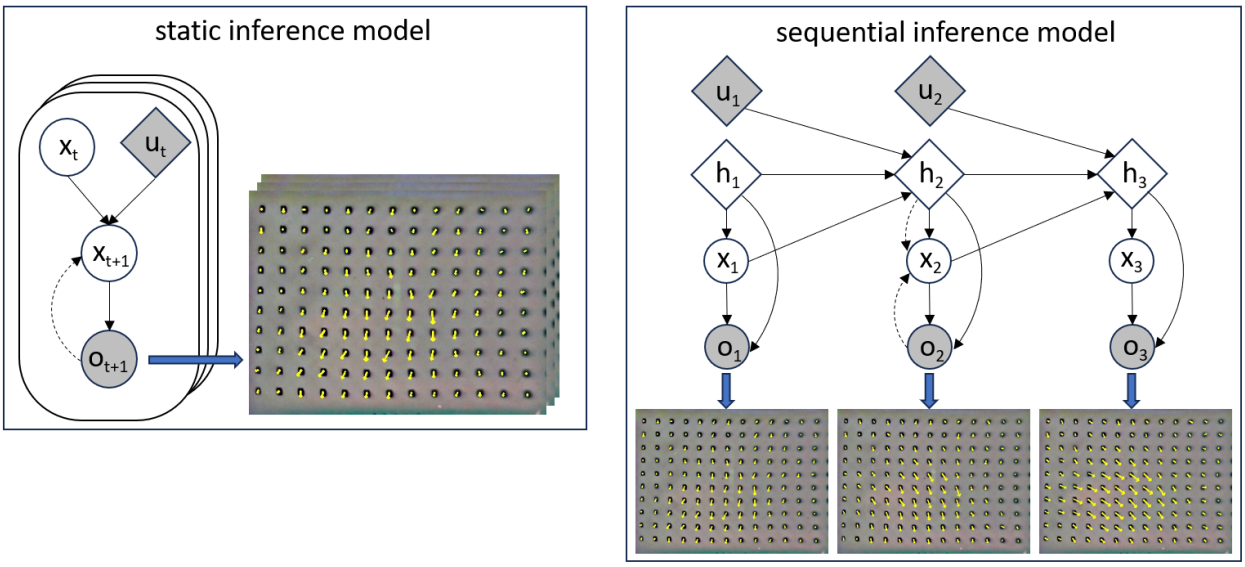

Figure 3.2: Two different types of inference models were considered for capturing the latent space dynamics of the observation-action pairs. (L) The static inference model (section 3.3.4.1) assumes that each transition tuple is i.i.d. Representative tactile sensor data are shown as outputs of the generative process. (R) The sequential inference model (section 3.3.4.2) considers temporal relationships within the tactile sensor data. Solid and dashed arrows represent generative processes and inference functions, respectively. Grey and white shading denote whether the variables are observed or unknown, respectively. Squares and circles indicate whether variables are treated as deterministic or stochastic, respectively.

inference model as the following:

$$\begin{aligned}\mathbb{L}(\theta,\phi,\mathbf{o}_{1:T},\mathbf{u}_{1:T}) =& \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{o}_{1:T},\mathbf{u}_{1:T})}[\log p_\theta(\mathbf{o}_{1:T},\mathbf{x}_{1:T}|\mathbf{u}_{1:T})] - \\ & \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{o}_{1:T},\mathbf{u}_{1:T})}[\log q_\phi(\mathbf{z}_{1:T}|\mathbf{x}_{1:T},\mathbf{u}_{1:T})] \\ =& \sum_{t=1}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_{1:t}|\mathbf{o}_{1:T},\mathbf{u}_{1:T})}[\log p_\theta(\mathbf{o}_t|\mathbf{o}_{1:t-1},\mathbf{x}_{1:t},\mathbf{u}_{1:t}] - \\ & \sum_{t=1}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_{1:t-1}|\mathbf{o}_{1:T},\mathbf{u}_{1:T})}[D_{KL}(q_\phi(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{o}_{1:T},\mathbf{u}_{1:T})|| \\ & p_\theta(\mathbf{x}_t|\mathbf{o}_{1:t-1},\mathbf{x}_{1:t-1},\mathbf{u}_{1:t})]\end{aligned} \tag{3.20}$$

The ELBO for the sequential inference model contains a reconstruction accuracy term and a regularization term, as for the ELBO for a static inference model (Eqn.Equation (3.16)). The regularization term in Eqn. 3.20 for the sequential inference model has an interesting interpretation. Specifically, $p_\theta(\mathbf{x}_t|\mathbf{o}_{1:t-1},\mathbf{x}_{1:t-1},\mathbf{u}_{1:t})$ can be viewed as the "predictive" distribution because it outputs the probability of the latent state at step t based on information up to step $t-1$ and a control signal applied up to step t. At the same time, $q_\phi(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{o}_{1:T},\mathbf{u}_{1:T})$ can be viewed as the "update and smoothing" distribution because it infers the probability of the latent state at step t with future observations and control signals in addition to past latent state information. By minimizing the KL divergence between $p_\theta(\mathbf{x}_t|\mathbf{o}_{1:t-1},\mathbf{x}_{1:t-1},\mathbf{u}_{1:t})$ and $q_\phi(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{o}_{1:T},\mathbf{u}_{1:T})$, the ELBO for the sequential inference model aims to match the "predictive" distribution with the "update and smoothing" distribution.

Many of the recent works in the generative modelling literature use specific instances of the general definition of ELBO expressed in Eqn. Equation (3.20). By applying more conditional independence assumptions to the generative model, the exact posterior $p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{o}_{1:T},\mathbf{u}_{1:T})$, approximate posterior $q_\phi(\mathbf{x}_t|\mathbf{x}_{1:t-1},\mathbf{o}_{1:T},\mathbf{u}_{1:T})$, and ELBO in Eqn. Equation (3.20) can be simplified further. We refer the interested reader to [33].

In Sections 3.3.4.1 and 3.3.4.2, we describe the formulations that we used to investigate whether

a "static" or "sequential" inference model, implemented with Gaussian processes, would be more appropriate for our goal of tactile-driven manipulation of deformable linear objects.

3.3.4.1 Static Model of Latent Space Dynamics – Deep Gaussian Process Model

The structure of the static inference model is shown in Figure 3.2 (left). Our use of this inference framework was inspired by [10]. As will be discussed in Section 3.4.3, we show that a GP-specific static inference model fails to capture the latent space dynamics accurately enough for the the tactile-driven manipulation of DLOs in this work.

To infer latent space dynamics using probabilistic generative modelling with a static inference framework, each transition tuple $\{\mathbf{o}_{t:t+1}, \mathbf{x}_{t:t+1}, \mathbf{u}_t\}$ is treated as i.i.d. and we maximize the observation transition probability $p(\mathbf{o}_{t+1}|\mathbf{o}_t, \mathbf{u}_t)$ as follows.

$$\begin{aligned}
\log p(\mathbf{o}_{t+1}|\mathbf{o}_t, \mathbf{u}_t) &= \log \frac{p(\mathbf{o}_{t+1}, \mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{o}_t, \mathbf{u}_t)}{p(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{o}_{t+1}, \mathbf{o}_t, \mathbf{u}_t)} \\
&= \log \frac{p(\mathbf{o}_{t+1}, \mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{o}_t, \mathbf{u}_t)p(\mathbf{x}_t|\mathbf{o}_t, \mathbf{u}_t)}{p(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{o}_{t+1}, \mathbf{o}_t, \mathbf{u}_t)} \\
&= \log \frac{p(\mathbf{o}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)p(\mathbf{x}_t|\mathbf{o}_t)}{p(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{o}_{t+1}, \mathbf{o}_t, \mathbf{u}_t)}
\end{aligned} \tag{3.21}$$

After applying an assumption of conditional independence and Bayes rule, the posterior distribution on the denominator can be decomposed as

$$\begin{aligned}
p(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{o}_{t+1}, \mathbf{o}_t, \mathbf{u}_t) &= p(\mathbf{x}_{t+1}|\mathbf{o}_{t+1}, \mathbf{o}_t, \mathbf{u}_t)p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{o}_{t+1}, \mathbf{o}_t, \mathbf{u}_t) \\
&\approx p(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})p(\mathbf{x}_t|\mathbf{o}_t)
\end{aligned} \tag{3.22}$$

Introducing the variational distribution q on $\{\mathbf{x}_{t+1}, \mathbf{o}_{t+1}\}$ we obtain the following:

$$\begin{aligned}
\log p(\mathbf{o}_{t+1}|\mathbf{o}_t, \mathbf{u}_t) = \mathbb{E}_{q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})}[&\log \frac{p(\mathbf{o}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)}{q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})} + \\
&\log \frac{q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})}{p(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})}]
\end{aligned} \tag{3.23}$$

Given that $\mathbb{E}_q[\log \frac{q}{p}] = D_{KL}(q||p)$, we obtain the GP-specific ELBO for the static inference model as follows:

$$\begin{aligned}\mathbb{L} = &\underbrace{\mathbb{E}_{q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})} \log(p(\mathbf{o}_{t+1}|\mathbf{x}_{t+1}))}_{\textbf{(I)}: \text{Reconstruction accuracy}} + \\ &\underbrace{\mathbb{E}_{q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})} \log(p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t))}_{\textbf{(II)}: \text{Latent space dynamics}} + \\ &\underbrace{\mathbb{E}_{q(\mathbf{x}_{t+1}|\mathbf{o}_{\mathbf{t+1}})} - \log(q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1}))}_{\textbf{(III)}: \text{Entropy regularization}}\end{aligned} \tag{3.24}$$

The ELBO expression in Eqn. 3.24 contains three terms: (**I**) reconstruction accuracy, (**II**) latent space dynamics, and (**III**) entropy regularization. Importantly, (**II**) is the cost term that we use GP to represent because the cost term describes the likelihood of latent state transitions over the expectation of the encoder distribution. Maximization of the cost term (**II**) is equivalent to maximization of the mean of GP marginal likelihood evaluated with Monte Carlo samples generated from the encoder distribution $q(\mathbf{x}_{t+1}|\mathbf{o}_{t+1})$. The remaining terms (**I**, **III**) in the ELBO are computed with the encoder $q(\mathbf{x}|\mathbf{o})$ and decoder $p(\mathbf{o}|\mathbf{x})$, which are modelled with neural networks. All model parameters for the GP model and neural networks are trained jointly using stochastic gradient descent.

3.3.4.2 Sequential Model of Latent Space Dynamics – Deep Recurrent Guided Variational Gaussian Process Model

The structure of the sequential inference model is shown in Figure 3.2 (right). The motivation for this structure is to combine the stochastic state transition of the traditional state space model with the nonlinearity of recurrent neural networks (RNNs) in order to create a system capable of modeling time series data in a complex domain. The use of an RNN also allows the deterministic hidden state from long short-term memory (LSTM) cells or gated recurrent units (GRUs) to act as

an internal memory, which is crucial for capturing sequential relationships within complex time series data. The combination is highlighted by the deterministic hidden states $\{\mathbf{h}_{1:T}\}$ that split the model into a deterministic path and stochastic path (Figure 3.2, right). This approach was discovered and explored by the model-based reinforcement learning community [21, 30, 35, 14], but has not been widely applied to experiments on real robots. In this work, we use the model structure in [35], but we present the derivation of the ELBO in a more systematic way, as inspired by [33, 29].

With the incorporation of the deterministic latent states $\{\mathbf{h}_{1:T}\}$, and the conditional independence assumption (Figure 3.2, right), the generative process can be factorized as follows,

$$\begin{aligned} p(\mathbf{o}_{1:T}, \mathbf{x}_{1:T}, \mathbf{h}_{1:T} | \mathbf{u}_{1:T-1}) &= p(\mathbf{o}_{1:T} | \mathbf{x}_{1:T}, \mathbf{h}_{1:T}) p(\mathbf{x}_{1:T}, \mathbf{h}_{1:T} | \mathbf{u}_{1:T-1}) \\ &= \prod_{t=1}^{T} p(\mathbf{o}_t | \mathbf{x}_t, \mathbf{h}_t) p(\mathbf{x}_t | \mathbf{h}_t) \\ &\quad p(\mathbf{h}_t | \mathbf{x}_{t-1}, \mathbf{h}_{t-1}, \mathbf{u}_{t-1}) \end{aligned} \tag{3.25}$$

With Bayesian networks, every variable is conditionally independent of its non-descendants given its parents. As a result, the true posterior for the graphical model shown in Figure 3.2 (right) can be factorized as follows,

$$\begin{aligned} p(\mathbf{x}_{1:T}, \mathbf{h}_{1:T} | \mathbf{u}_{1:T-1}, \mathbf{o}_{1:T}) = \prod_{t=1}^{T} & p(\mathbf{x}_t | \mathbf{h}_t, \mathbf{o}_t) \\ & p(\mathbf{h}_t | \mathbf{h}_{t-1}, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{o}_t) \end{aligned} \tag{3.26}$$

It should be noted that the information flow related to $\mathbf{h}_t$ is deterministic. Thus, with known parents, there is no stochasticity remaining in $\mathbf{h}_t$, and $p(\mathbf{h}_t | \mathbf{h}_{t-1}, \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ in Eqn. Equation (3.25) and $p(\mathbf{h}_t | \mathbf{h}_{t-1}, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{o}_t)$ in Eqn. Equation (3.26) each reduce to the same delta function centered at the RNN output. If we impose the same structure assumption on the inference function,

we get the following,

$$q(\mathbf{x}_{1:T}, \mathbf{h}_{1:T} | \mathbf{u}_{1:T-1}, \mathbf{o}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{x}_t | \mathbf{h}_t, \mathbf{o}_t) p(\mathbf{h}_t | \mathbf{h}_{t-1}, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \tag{3.27}$$

The GP-specific ELBO for the sequential inference model can be computed as

$$\begin{aligned} \mathbb{L} = & \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{1:t}, \mathbf{h}_{1:t} | \mathbf{u}_{1:t-1}, \mathbf{o}_{1:t})} [\log p(\mathbf{o}_t | \mathbf{x}_t, \mathbf{h}_t)] - \\ & \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{1:t-1}, \mathbf{h}_{1:t} | \mathbf{u}_{1:t-1}, \mathbf{o}_{1:t})} [D_{KL}(q(\mathbf{x_t} | \mathbf{h_t}, \mathbf{o_t}) || p(\mathbf{x_t} | \mathbf{h_t}))] \end{aligned} \tag{3.28}$$

Given that $\mathbf{h}_t$ is the output of the RNN , $\mathbf{h}_t$ is essentially a function of $(\mathbf{x}_{1:t-1}, \mathbf{u}_{1:t-1})$. Applying this observation to Eqn. Equation (3.28), we obtain the final representation of the GP-specific ELBO for the sequential inference model,

$$\begin{aligned} \mathbb{L} = & \underbrace{\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{1:t} | \mathbf{u}_{1:t-1}, \mathbf{o}_{1:t})} [\log p(\mathbf{o}_t | \mathbf{x}_{1:t}, \mathbf{u}_{1:t-1})]}_{\textbf{(I)}: \text{ Reconstruction accuracy}} - \\ & \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{1:t-1} | \mathbf{u}_{1:t-1}, \mathbf{o}_{1:t})} [D_{KL}(q(\mathbf{x_t} | \mathbf{x_{1:t-1}}, \mathbf{o_t}, \mathbf{u_{1:t-1}}) || \\ & \underbrace{p(\mathbf{x_t} | \mathbf{x_{1:t-1}}, \mathbf{u_{1:t-1}}))]}_{\textbf{(II)}: \text{ KL Regularization}} \end{aligned} \tag{3.29}$$

The ELBO expression in Eqn. 3.29 contains two terms: (**I**) reconstruction accuracy and (**II**) KL regularization. Comparing the GP-specific form of ELBO for the sequential inference model (Eqn. Equation (3.29)) with the general form of ELBO for sequential inference models (Eqn. Equation (3.20)), the GP-specific simplification of the sequential inference model used in this work can be identified clearly. Considering the reconstruction accuracy term (**I**) in Eqn. 3.29, we assume that the latent state and observation at the current timestep depend on previous latent states and control signals but not on previous observations. Considering the KL regularization term (**II**) in Eqn. 3.29, all future ($t+1$ to T) information about the observation data sequence is discarded such

that $q(\mathbf{x}_{1:t}|\mathbf{u}_{1:t-1}, \mathbf{o}_{1:t})$ acts only as an "update" distribution. Since $q(\mathbf{x}_{1:t}|\mathbf{u}_{1:t-1}, \mathbf{o}_{1:t})$ still contains information up to and including the current timestep, the temporal relationship of the observation data sequence remains partially encoded.

To use this GP-specific formulation of the sequential inference model, we first train the model with sequences of collected tactile data. Then we use the output of the model in latent space as a training dataset to guide the Gaussian process. As will be discussed in Section 3.4.4, we show that a GP-specific sequential inference model, as described here, outperforms a GP-specific static inference model, as described in Section 3.3.4.1.

3.4 Experimental Procedure and Evaluation

In this section, we describe our experimental procedure and address two questions. For Q1, we ask, when learning a latent space dynamics model based on high-dimensional tactile sensor data, is a sequential inference model (Section 3.3.4.2) preferred over a simpler static inference model (Section 3.3.4.1)? For Q2, we ask, how well does a latent space dynamics model learned from high-dimensional tactile sensor data work in an optimization-based model predictive control framework for the manipulation of deformable linear objects on a real robot?

In Sections 3.4.1 and 3.4.2, we describe the hardware setup and data collection procedure, respectively, for the real robot experiments. In Sections 3.4.3 and 3.4.4, we address Q1 by describing the training procedures and results for learning a latent space dynamics model using a static inference model and sequential inference model, respectively. Finally, in Section 3.4.6, we address Q2 and evaluate the use of a learned latent space dynamics model within an optimization-based MPC framework for a tactile-driven manipulation task on a real robot. Our experimental procedure is illustrated at a high level in Figure 3.3.

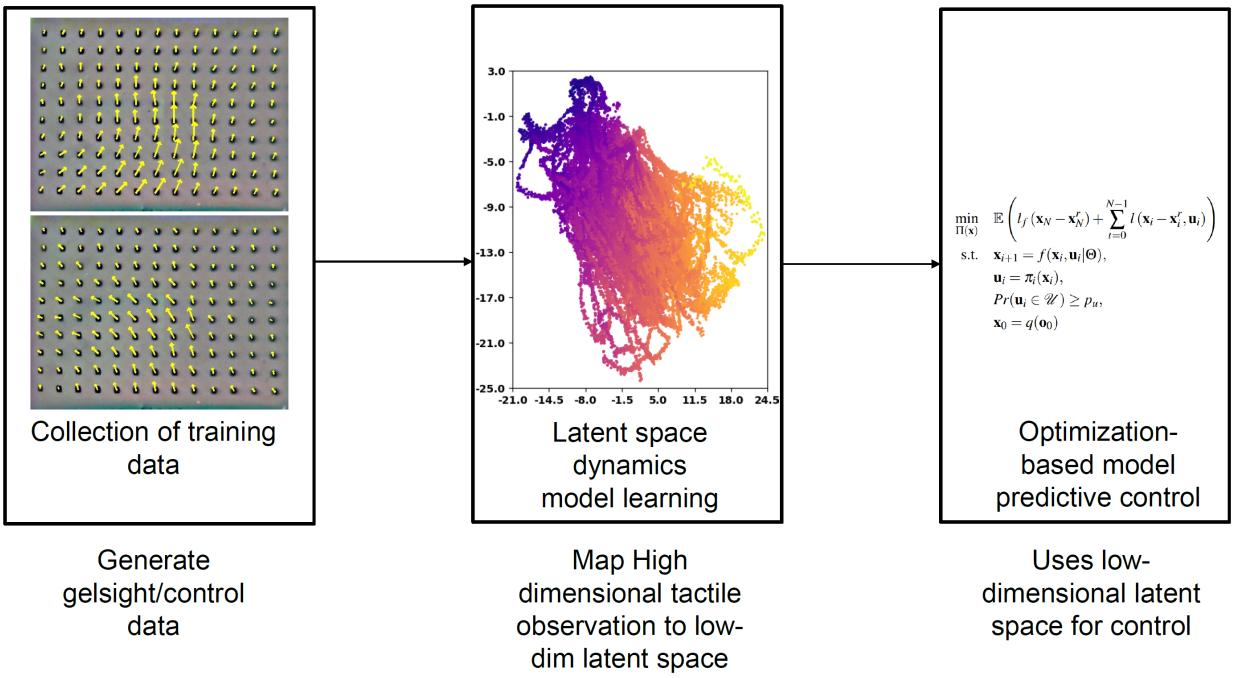

Figure 3.3: Experimental procedure: (Left) We collected tactile observation – robot action pairs for use as training data. (Center) System dynamics were mapped from a high-dimensional tactile sensor data space to a low-dimensional latent space. (Right) The learned latent space dynamics model was integrated with an optimization-based model predictive control framework.

3.4.1 Experimental Setup

A 7-degree-of-freedom (DOF), tendon-driven robot arm (WAM Arm, Barrett Technology) was used to manipulate an elastic cable in a plane (Figure 3.4). The fixed end of the elastic cable was attached to the base of a rigid frame. The cable was passed over a rigid rod, which will be hereafter referred to as the "anchor point" in our system. The grasped end of the elastic cable was attached to a rigid rig that could be easily grasped by a 1-DOF, parallel gripper [85]. Each fingerpad of the gripper was outfitted with an elastomeric camera-based tactile sensor (wedge GelSight sensor [112]).

The anchor point serves as the origin for an $\mathbf{x}-\mathbf{y}$ plane, in which the robot moves the grasped end of the elastic cable or "grasp point". The $(\mathbf{x},\mathbf{y})$ coordinates of the grasp point are parameterized in terms of cable length $\mathbf{d}$ and cable angle α. Cable length $\mathbf{d}$ is defined as the distance between the grasp point and the anchor point. Cable angle α is measured positive counter-clockwise from the $\mathbf{x}$-axis to the stretched cable. An Aruco marker attached to the rigid rig is used to track $\mathbf{d}$ and α. The Aruco marker data are not used for learning a latent space dynamics model or model predictive control, and are only used for post hoc visualizations of the latent space after a latent space dynamics model has been learned.

Importantly, the $(\mathbf{d},\alpha)$ pair contains information about both pose state and tensile state of the elastic cable. Specifically, $\mathbf{d}$ and α capture magnitude and direction information, respectively, about the tensile force vector acting along the length of the elastic cable. We believe that tactile sensors are uniquely suited for encoding tensile force vector information that cannot be gleaned from images of taut (and, therefore, straight) cable configurations without a priori knowledge of the elastic properties of the cable.

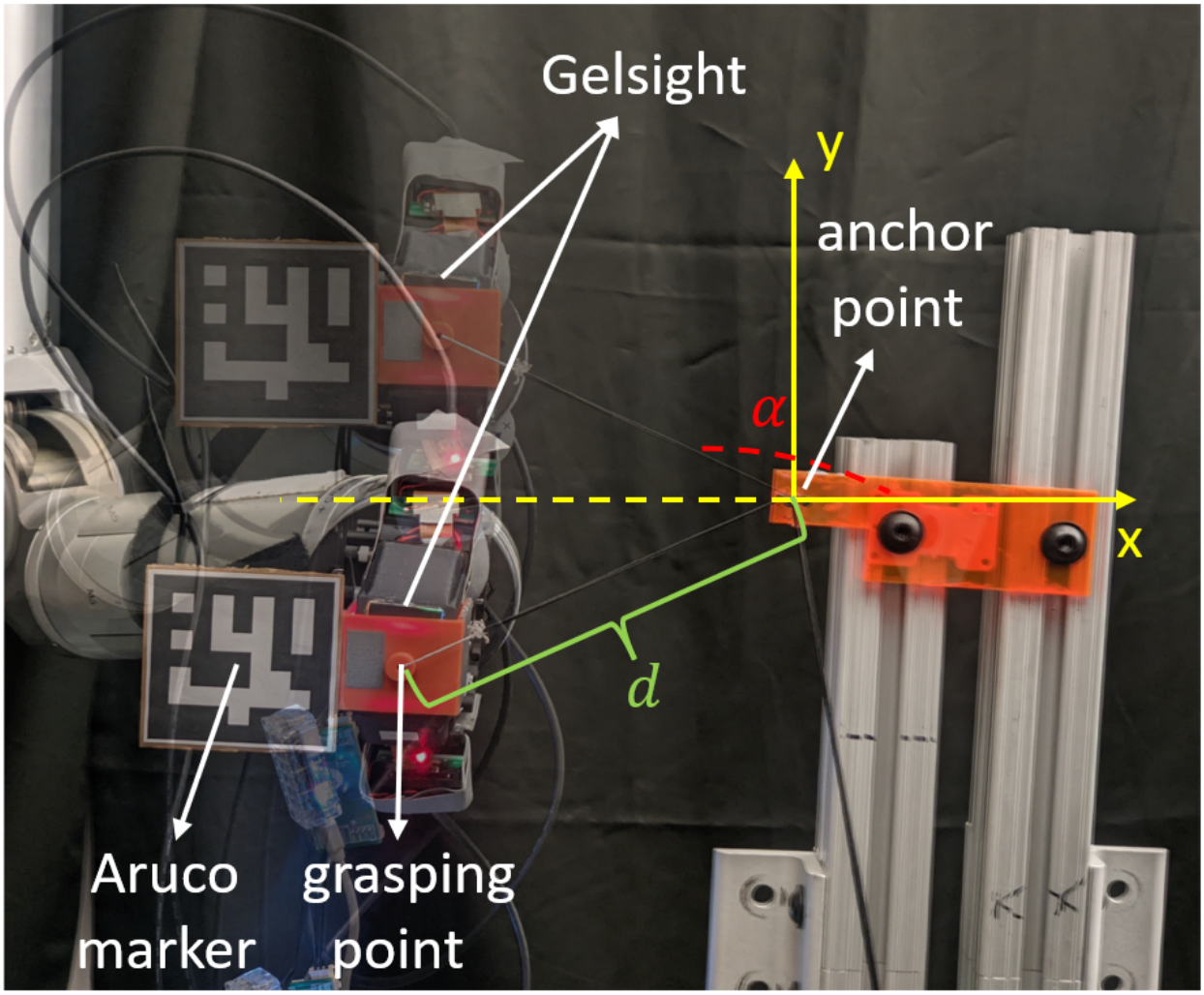

Figure 3.4: Experiment setup: The robot gripper moves within a 2D plane defined by the **x**- and **y**-axes. The top finger of the gripper is outfitted with a camera-based tactile sensor (GelSight). The configuration of the elastic cable is specified as the angle α between the cable and the **x**-axis, and the distance **d** between the grasping point and the anchor point. The Aruco marker is used solely for visualization within the latent space and is not used for training.

3.4.2 Collecting and Augmenting Observation-Action Pairs for Training

In order to learn a latent space dynamics model, we collected observation-action pairs using a total of 260 rollouts on the real robot system shown in Figure 3.4. Prior to each rollout, baseline tactile sensor data were collected with the grasp point close to the anchor point such that the elastic cable was not quite taut. We then commanded the robot to move to a starting position where the cable was minimally stretched, and began sending velocity commands in Cartesian space. Marker displacement vector fields were extracted from the raw GelSight image data. We used the GelSight marker displacement vector fields and Cartesian velocity vector for each timestep t as an observation-action pair $\{\mathbf{o}_t, \mathbf{u}_t\}$.

Velocity commands in the Cartesian space were designed to randomly sample combinations of cable angle α and cable length $\mathbf{d}$. First, the $[0^\circ, 360^\circ]$ range was divided into six non-overlapping sectors 60° wide. The direction of the velocity command vector was sampled from each of the six sectors using a resolution of 5°. The magnitude of the velocity command vector was set at 3 cm/s for all rollouts. Each rollout lasted for 90 sec.

We augmented the original observation-action pair dataset by dividing each rollout into multiple sequences, each having a manually selected sequence length of 50 datapoints. Sequences were sampled at each timestep until insufficient datapoints remained for a 50-datapoint sequence (i.e. sampling occurred at $[t_1, t_2, \cdots, t_{T-50+1}]$). We also reversed the data sequences and injected noise into the displacement vector (magnitude and direction) of each marker. The noise for each marker's displacement vector was sampled from a normal distribution having a mean and standard deviation that were calculated from a 30 sec batch of tactile sensor data, collected after the gripper reached and remained briefly at a commanded location where the cable was moderately taut. The original dataset was augmented through the inclusion of the reversed and noise-injected sequences.

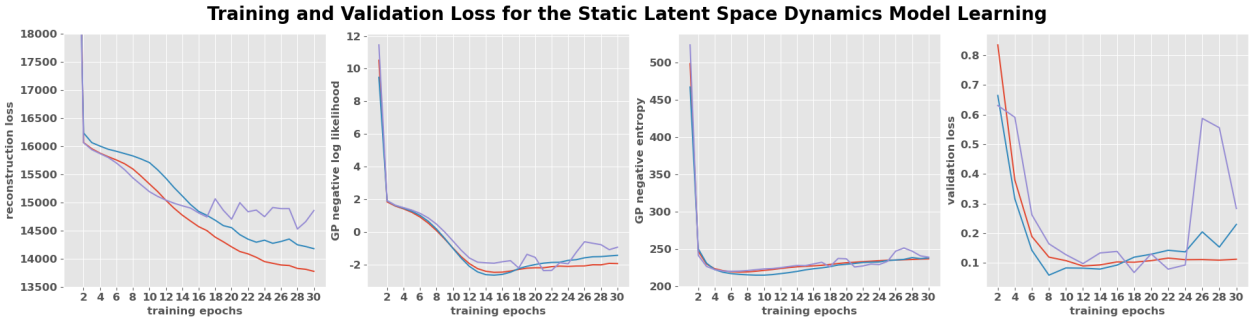

Figure 3.5: The training and validation loss curves are shown for three random seeds for the GP-based static inference model of latent space dynamics. From left to right, the first three plots show the training loss for the encoder-decoder reconstruction loss, GP latent space transition likelihood, and negative entropy encoder regularization, respectively (terms **I**, **II**, and **III** in Eqn. Equation (3.24)). The plot on the far right shows the validation loss defined in Section 3.4.3.

All control action and tactile sensor data were recorded at a frequency of 5 Hz for each 90 sec rollout. After augmentation, we had approximately 180k observation-action pairs and 40k observation-action pairs in the training and validation datasets, respectively.

3.4.3 Learning a Latent Space Dynamics Model from Tactile Sensor Data Using a Static Inference Model

In this section, we describe the methods and results for learning a latent space dynamics model from high-dimensional tactile sensor data using a GP-based static inference model. Applying the physical intuition that the cable state for our manipulation task has two "general coordinates" (magnitude and direction), we defined the latent state as a two-dimensional real vector $\mathbf{x}_t \in \mathbb{R}^2$. The observation $\mathbf{o}_t$ is the stacked displacement vector (magnitude and direction) for all 130 markers on the elastomeric GelSight fingerpad. Hence, $\mathbf{o}_t$ has a dimensionality of 260.

Both the encoder $\mathbf{q}_\phi$ and decoder $\mathbf{p}_\theta$ were modelled as feedforward neural networks. Details

on the structure of the encoder and decoder can been found in the Appendix (Section 3.6). The number of inducing points for the stochastic variational Gaussian process was set at 300. For the SVGP model, we assumed a constant, nonzero mean for the prior distribution. The constant unknown mean becomes an additional hyperparameter alongside the length scales, signal variance, and noise variance introduced in Section 3.3.1.

The neural networks and SVGP were implemented using PyTorch [72] and GPytorch [32], respectively. Neural network parameters and SVGP hyperparameters were jointly optimized using stochastic gradient descent via Adam [47]. The learning rates for the neural network paramters and GP hyperparameters were set to $5e^{-4}$ and $1e^{-3}$, respectively. We trained the model for 30 epochs, and performed validation every 2 epochs.

The training loss was implemented as in Eqn. Equation (3.24). Given that the input vector has a much higher dimensionality (260-D) than the latent state vector (2-D), we expected that the training loss would be dominated by the reconstruction loss. It should be noted that our ultimate goal is not to absolutely minimize reconstruction loss, but rather to learn a GP-based inference model that can capture dynamics in the latent space well enough to achieve our tactile-driven manipulation task later.

We defined the validation loss as the difference between the original and reconstructed latent space trajectories for a given (50-datapoint) observation-action sequence sampled from the validation dataset. The original latent space trajectory was obtained by sending the complete observation-action sequence to the encoder. The reconstructed latent space trajectory was obtained by sending the first encoded latent state and complete control action sequence to the SVGP in order to reconstruct the remainder of the sequence in the latent space. The overall validation loss at each validation epoch was defined as the average of validation losses computed for every observation-action sequence in the validation dataset.

To check whether neural network initialization and training data shuffling influenced the training, we examined the training experiment results for three different random initialization seeds. We will refer to the results from each seed as a trial. The training loss (encoder-decoder reconstruction loss, GP latent space transition likelihood, and negative entropy encoder regularization) and validation loss curves for the three trials are shown for the SVGP static inference model in Figure 3.5. The reconstruction loss decreases consistently for the majority of the 30 epochs. As expected, the magnitude of the reconstruction loss is much greater than that for the latent space transition likelihood and the negative entropy encoder regularization terms.

For all three trials, the validation loss stopped decreasing during the training. For one trial (purple trace), the validation loss increased significantly at epoch 24. This indicates that the difference between the original and reconstructed latent space trajectories dramatically worsened at epoch 24. We also observed that the GP latent space transition likelihood, and negative entropy encoder regularization terms began increasing in the early or middle stages of the training process. This indicates that the latent space dynamics prediction was getting worse and the uncertainty of the GP model was increasing.

The training and validation loss trends in Figure 3.5 suggest that the latent space dynamics model using a GP-based static inference model is overfitted to the encoder-decoder reconstruction and does not capture the latent space dynamics well. To further investigate this issue, we compared an original tactile sensor data sequence with reconstructed sequences at different epochs.

Figure 3.6 shows an original sequence of GelSight marker displacement vector fields from the validation dataset (Row 1) as compared to sequences reconstructed from the learned dynamics model at epochs 2, 18, and 30 (Rows 2, 3, and 4, respectively). The data are shown for the worst-performing trial (purple trace) shown in Figure 3.5. Epoch 18 marks the epoch where the lowest validation loss was obtained for this particular trial. This particular rollout sequence, comprised

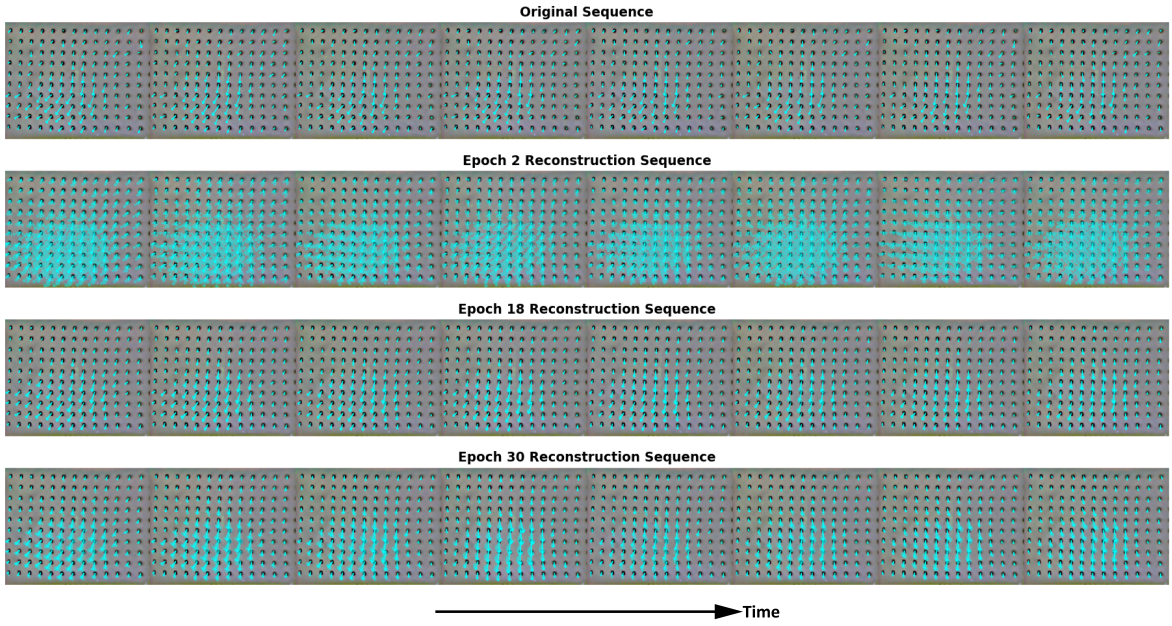

Figure 3.6: For the GP-based static inference model of latent space dynamics, the original GelSight marker displacement vector fields (Row 1) are shown alongside the tactile sequences reconstructed from the learned dynamics model at epochs 2, 18, and 30 (Rows 2, 3, and 4, respectively). The solid arrows in Rows 2-4 are reconstructed by sending the GP predictive mean to the decoder network. The shaded arrows are reconstructed by sending 20 predictions sampled from the GP predictive distribution to the decoder network. The data shown are for the worst performing trial (purple trace) in Figure 3.5.

of 50 datapoints, lasts for 10 sec. We show the marker displacement vector fields in 1.25 sec increments, beginning with $t = 0$ sec on the far left and ending at $t = 10$ sec on the far right.

The tactile sensor data reconstruction is performed similarly to how the validation tests are conducted. We set the initial latent state of the GP by encoding the first recorded marker displacement vector field into the latent space. Then we predict the GP distributions in the 2-D latent space for each timestep in the sequence using the latent state from the previous timestep, the known control action, and the learned static inference model of the latent space dynamics. For visualization purposes, the mean and 20 random samples from the GP predictive distributions are sent to the decoder for reconstruction in the 260-D space of marker displacement vector fields. The mean displacement vectors are shown with solid arrows and the sampled vectors are shown with shaded arrows in all reconstructed sequences in Figure 3.6.

By comparing the original sequence and the reconstructed sequence at epoch 2 (Figure 3.6), we observe that the solid arrows reconstructed from the latent state means match those of the original sequence fairly well. This suggests that the GP predictive distribution has already captured the transition dynamics of the mean relatively well in the latent state space, and that the decoder is doing a decent job with reconstruction. These observations are supported by the significant decrease in the reconstruction loss and GP latent space transition likelihood from the training loss curves in Figure 3.5.

From epoch 2 to epoch 18, we see further decreases in the reconstruction loss and GP latent space transition likelihood, which reflects continued improvements in the reconstruction and latent space predictions, respectively. Likewise, the regions spanned by the shaded arrows in the reconstructions from epoch 18 (Figure 3.6) shrink substantially as compared to those from epoch 2. This indicates that the covariance kernel of the latent space dynamics model has narrowed around its predictive mean.

From epoch 18 to epoch 30, we observe a degradation in the performance of the latent space dynamics model reflected in the reconstructed sequences (Figure 3.6). By epoch 30, the solid arrows that represent the predictive means of the marker displacement vector fields no longer match well with those from the original sequence. The regions spanned by the shaded arrows in the reconstructions from epoch 30 have increased as compared to those from epoch 18. This reflects an increase in the uncertainty of the latent space dynamics model from epoch 18 to epoch 30 that correspond with increases in the the GP latent space transition likelihood and validation loss in Figure 3.5.

Thus far, the results and analysis seem to indicate that the learned model at epoch 18 provides a good combinatorial performance for both the latent space dynamics modelling and the reconstruction. However, a closer look at the corresponding latent space structure at epoch 18 suggests otherwise. Row 1 of Figure 3.7 shows the latent space structure, parameterized by α and $\mathbf{d}$, after mapping all observations from the validation dataset through the trained encoder network at epochs 2, 18, and 30. Row 2 of Figure 3.7 shows all reconstructed and original trajectories, as encoded in the latent space, at epochs 2, 18, and 30. The start and end of an individual trajectory (purple trace in Figure 3.5) are highlighted with a red cross and blue triangle, respectively.

From Figure 3.7, we can comment on three key aspects of the latent space structure: shape, distribution of configurational states, and continuity. First, the geometric shape of the latent space for epoch 2 aligns best with our physical tuition. The shape is an arc, which makes sense for the range of cable angles that were randomly sampled during data collection.

Second, the distribution of configurational states, as reflected by the colormaps, make physical sense for both α and $\mathbf{d}$. For α, the values transition from 140° at the top of the arc toward 230° at the bottom of the arc. In this sense, the latent space structure parallels the sampled motions of the gripper within the second and third quadrants of the planar $\mathbf{x}-\mathbf{y}$ workspace (Figure 3.4). For

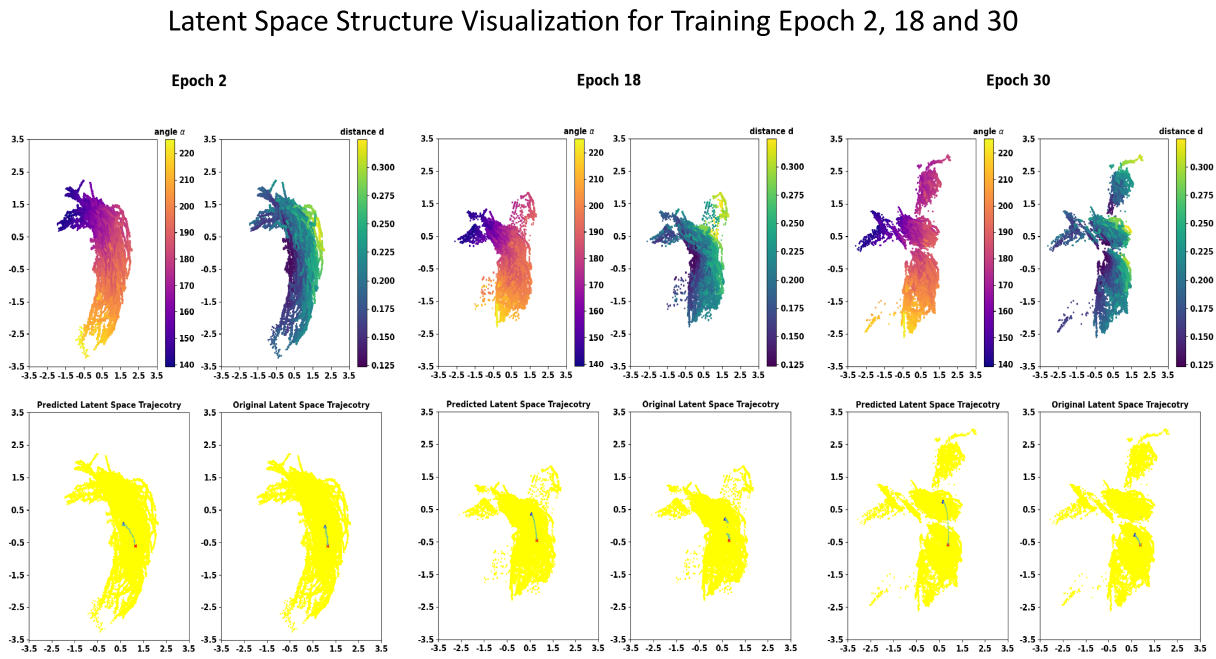

Figure 3.7: (Row 1) The latent space structure is shown, parameterized by α and $\mathbf{d}$, after mapping all observations from the validation dataset through the trained encoder network at epochs 2, 18, and 30 for the GP-based static inference model of latent space dynamics. The colormaps for α and $\mathbf{d}$ are based on data from the Aruco marker shown in Figure 3.4. (Row 2) All reconstructed and original sequences of validation data, as encoded in the latent space, are shown at epochs 2, 18, and 30. The start and end of an individual trajectory (purple trace in Figure 3.5) are highlighted with a red cross and blue triangle, respectively.

d, the values increase as one moves radially outward from the virtual center of the arc. Again, the latent space structure parallels the sampled motions of the gripper that placed the grasping point at different radial distances relative to the fixed anchor point.

Third, we observe that the latent space for epochs 18 and 30 are more disjointed when compared to that for epoch 2. Although the sampled motions of the gripper comprise a relatively continuous region in the α and **d** space, only epoch 2 exhibits the most continuous latent space structure of the three representative epochs. At epoch 30, the latent space structure is comprised of several disjoint sets.

Furthermore, the similarity between the reconstructed and original trajectory highlighted in Row 2 of Figure 3.7 is the greatest for epoch 2. For epoch 18, the original trajectory is disjoint. For epoch 30, the original trajectory for is significantly shorter than the reconstructed trajectory.

Although results were only shown in Figure 3.6 and Row 2 of Figure 3.7 for one random trial (purple trace in Figure 3.5), similar findings were made for the other two random trials described in Figure 3.5.

Figures 3.5, 3.6, and 3.7 illustrate that it is not straightforward to identify a satisfying latent space dynamics model using the GP-specific static inference model. The training and validation loss (Figure 3.5) and sample reconstruction of marker displacement vector fields (Figure 3.6) would suggest that epoch 18 yields an effective latent space dynamics model. However, the latent space structure for epoch 18 (Figure 3.7) has some undesirable attributes when considering downstream integration with an optimization-based model predictive control framework.

Although results were only shown in Figure 3.6 and Row 2 of Figure 3.7 for one random trial (purple trace in Figure 3.5), similar findings were made for the other two random trials described in Figure 3.5.

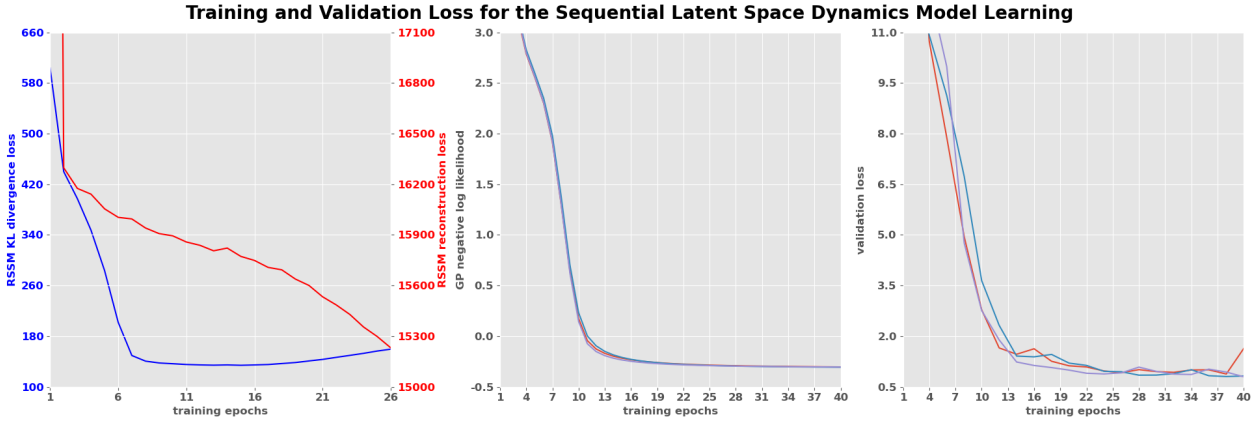

Figure 3.8: The training and validation loss curves are shown for the GP-based sequential inference model of latent space dynamics. For brevity, the recurrent state space model was trained only once using 26 epochs. The SVGP was then trained with the latent state encoded by the inference model **q**. The plot on the left shows the reconstruction loss (red) and KL divergence regularization loss (blue) (terms **I** and **II** from Eqn. Equation (3.29)). The plots in the center and on the right show the SVGP negative log likelihood and validation loss, respectively, for the same three random seeds used to evaluate the GP-specific static inference model.

Overall, we believe the results presented above suggest that the static inference model treats each observation-action pair as i.i.d., which may be an inappropriate assumption for the noisy tactile sensor data generated from real robot experiments in this work. To better capture the latent space dynamics, we hypothesize that learning an inference model that captures temporal relationships in the tactile sensor data will lead to improved performance.

3.4.4 Learning a Latent Space Dynamics Model from Tactile Sensor Data Using a Sequential Inference Model

In this section, we describe the methods and results for learning a latent space dynamics model from high-dimensional tactile sensor data using a GP-based sequential inference model. The encoder $\mathbf{q}(\mathbf{x}_t|\mathbf{h}_t,\mathbf{o}_t)$, decoder $\mathbf{p}(\mathbf{o}_t|\mathbf{x}_t,\mathbf{h}_t)$, and generative model of the stochastic latent state $\mathbf{p}(\mathbf{x}_t|\mathbf{h}_t)$ in Eqn. Equation (3.28) are modelled with feedforward neural networks. The deterministic latent state transition probability $\mathbf{p}(\mathbf{h}_t|\mathbf{h}_{t-1},\mathbf{x}_{t-1},\mathbf{u}_{t-1})$ in Eqn. Equation (3.27) is modelled with a gated recurrent unit (GRU) network. Details on the network structure of the sequential inference model can be found in the Appendix (Section 3.6).

The observation vector $\mathbf{o}_t$ and latent state vector $\mathbf{x}_t$ for the sequential inference model are the same as those for the static inference model. For the sequential inference model, the dimensionality of the deterministic latent state space $\mathbf{h}$ is set to 8.

As with the GP-specific static inference model (Section 3.3.4.1), the learning rate was set to $5e^{-4}$ for all neural network parameters for the GP-specific sequential inference model. However, the learning rate for the SVGP hyperparameters was increased to $5e^{-3}$ based on findings from preliminary experiments. The preliminary experiments showed that an SVGP learning rate of $1e^{-3}$ significantly slowed training and more epochs were required for the training to converge.

The number of inducing points for the SVGP for the GP-specific sequential inference model was decreased to 100 from the 300 that were used for the GP-specific static inference model. The reduction of the number of inducing points was done for two reasons. First, an SVGP with 300 inducing points is too computationally expensive for a downstream optimization-based model predictive controller. Second, we wanted to investigate whether the consideration of temporal relationships in the tactile sensor data would lead to improvements in overall performance even if the representational power of the SVGP was purposely degraded.

For brevity, we only trained the recurrent state space model once using 26 epochs. As in Section 3.3.4.1, we trained the SVGP for 40 epochs and performed validation every 2 epochs. In order to directly compare trial-specific results between the static and sequential inference models, we used the same three random seeds as in Section 3.3.4.1. The same definition of validation loss for the static and sequential inference models. That is, validation loss was defined as the difference between the original latent space trajectory from the encoder $\mathbf{q}$ and the reconstructed latent space trajectory from the SVGP.

The training loss for the recurrent state space model and SVGP as well as the validation loss are shown in Figure 3.8. For the training of the recurrent state space model, the magnitude of reconstruction loss (red) decreased dramatically from epoch 1 to epoch 2 and dominated the rest of the training process. It is also notable that the KL divergence regularization loss (blue) decreased consistently for the first few epochs. This indicates that the "predictive" distribution $\mathbf{p}(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{u}_{1:t-1})$ and the "update" inference distribution $q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{o}_t, \mathbf{u}_{1:t-1})$ began distinct from one another but became more similar as training continued.

The behaviors of the GP latent space transition likelihood and validation loss terms (center and right plots, respectively, in Figure 3.8) are much better for the sequential inference model than for the static inference model (Figure 3.5). Even after a significant drop in both the GP latent space

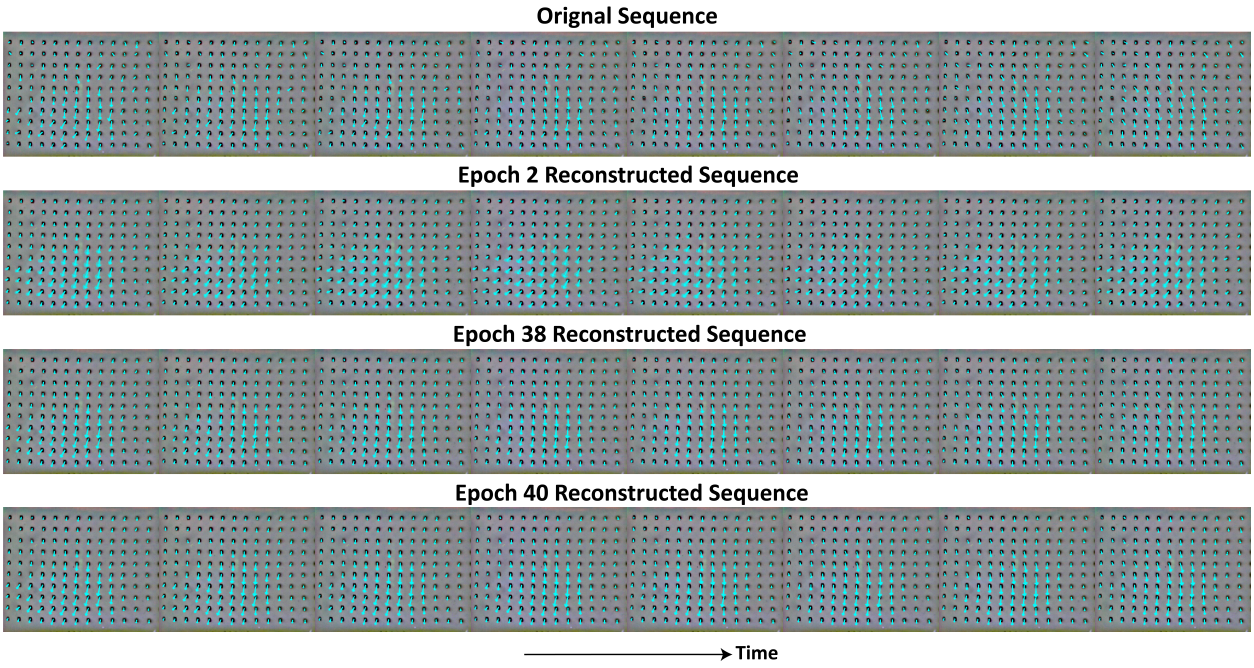

Figure 3.9: For the GP-based sequential inference model of latent space dynamics, the original GelSight marker displacement vector fields (Row 1) are shown alongside the tactile sequences reconstructed from the learned dynamics model at epochs 2, 38, and 40 (Rows 2, 3, and 4, respectively). The solid arrows in Rows 2-4 are reconstructed by sending the GP predictive mean to the decoder network. The shaded arrows are reconstructed by sending 20 predictions sampled from the GP predictive distribution to the decoder network. The data shown are for the red trace from the center and right plots in Figure 3.8.

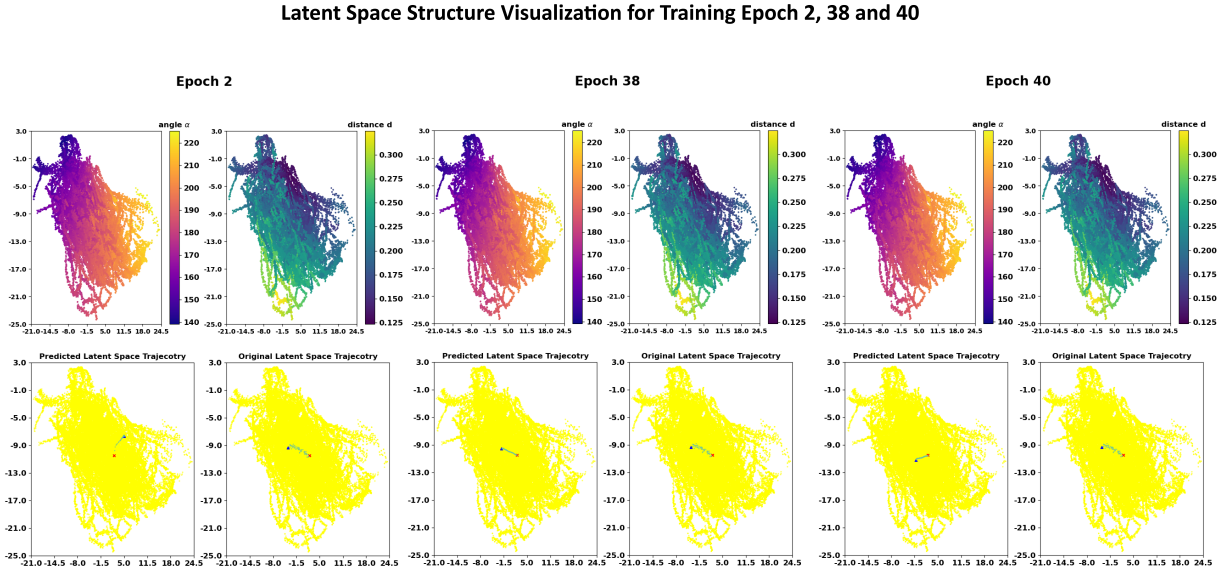

Figure 3.10: (Row 1) The latent space structure is shown, parameterized by α and $\mathbf{d}$, after mapping all observations from the validation dataset through the trained recurrent state space model at epochs 2, 38, and 40 for the GP-based sequential inference model of latent space dynamics. The colormaps for α and $\mathbf{d}$ are based on data from the Aruco marker shown in Figure 3.4. (Row 2) All reconstructed and original sequences of validation data, as encoded in the latent space, are shown at epochs 2, 38, and 40. The start and end of an individual trajectory (red trace from the center and right plots in Figure 3.8) are highlighted with a red cross and blue triangle, respectively.

transition likelihood and validation loss terms over the first 10 epochs of training for the sequential inference model (Figure 3.8), the terms continue to decrease at a slower rate for the remainder of the training process.

For all three random seeds, the lowest validation loss was observed by epoch 38 or epoch 40. Unlike the static inference model, the sequential inference model continues to improve upon its ability to capture the latent space dynamics as training continues. Furthermore, it does not appear as if the sequential inference model was overfit to the training data.

Figure 3.9 shows an original sequence of GelSight marker displacement vector fields from the validation dataset (Row 1) as compared to sequences reconstructed from the learned dynamics model at epochs 2, 38, and 40 (Rows 2, 3, and 4, respectively). The data are shown for the GP-specific sequential inference model for one random seed (red trace from the center and right plots in Figure 3.8). At the initial phase of the training (epoch 2), the uncertainty associated with the SVGP predictive covariance is already much less for the sequential inference model (Figure 3.9) as compared to that for the static inference model (Figure 3.6). While the uncertainty increased with training from epoch 18 to epoch 30 for the static inference model (Figure 3.6), the uncertainty does not increase with training for the sequential inference model. As seen in Figure 3.9, the reconstructed marker displacement vector fields in epochs 38 and 40 match those of the original tactile sensor data quite well.

Figure 3.10 facilitates a closer look at the corresponding latent space structure. Row 1 of Figure 3.10 shows the latent space structure, parameterized by α and $\mathbf{d}$, after mapping all observations from the validation dataset through the trained recurrent state space model at epochs 2, 38, and 40. Row 2 of Figure 3.10 shows all reconstructed and original sequences of validation data, as encoded in the latent space, at epochs 2, 38, and 40. The start and end of an individual trajectory (red trace from the center and right plots in Figure 3.8) are highlighted with a red cross and blue triangle, respectively.

During the early stages of training (epoch 2), the reconstructed and original trajectories in the latent space are dissimilar (Row 2 of Figure 3.10). The similarity between the reconstructed and original trajectories is the greatest for epoch 38. As expected from the slight increase in the validation loss for the corresponding (red) random seed in Figure 3.8, the similarity between the reconstructed and original trajectories worsens slightly for epoch 40 as compared to epoch 38.

With regards to shape and distribution of configurational states, the latent space structure for

the sequential inference model (Figure 3.10) does not align with physical intuition as directly as the latent space structure for the static inference model (Figure 3.7). Nonetheless, the sampled cable angles and lengths smoothly transition across the latent space, as evidenced by the color transitions in Row 1 of Figure 3.10. Additionally, the relatively continuous nature of the latent space structure is promising for the downstream integration with the optimization-based model predictive control framework.

3.4.5 Comparing the Performance of the Static and Sequential Inference Models

In this section, we directly compare the performance of the GP-based static inference model to that of the GP-based sequential inference model. We first chose the model with the lowest validation loss for each of the three experiment trials for both the static and sequential inference model learning experiments. For the static inference model, we ran the training dataset through the best performing encoder-decoder. For the sequential inference model, we ran the same training dataset through the best performing recurrent state space model. The resulting latent space structures were normalized so that we could directly compare reconstructed trajectories between the static and sequential inference models.

For each model, we sampled a total of 50k sequences from the validation dataset. Each sequence was comprised of 50 datapoints. For each sequence, we calculated the difference between the original and reconstructed latent space trajectories, similar to those highlighted in Row 2 of Figures 3.7 and 3.10. Finally, we computed the mean of the differences for prediction windows of 30, 40, and 50 datapoints using the results pooled from all three random seeds.

Figure 3.11 shows the distribution of differences between the original and reconstructed latent space trajectories on a normalized scale to enable the direct comparison of the GP-specific static and sequential inference models. The distributions of differences for 50k sample sequences are

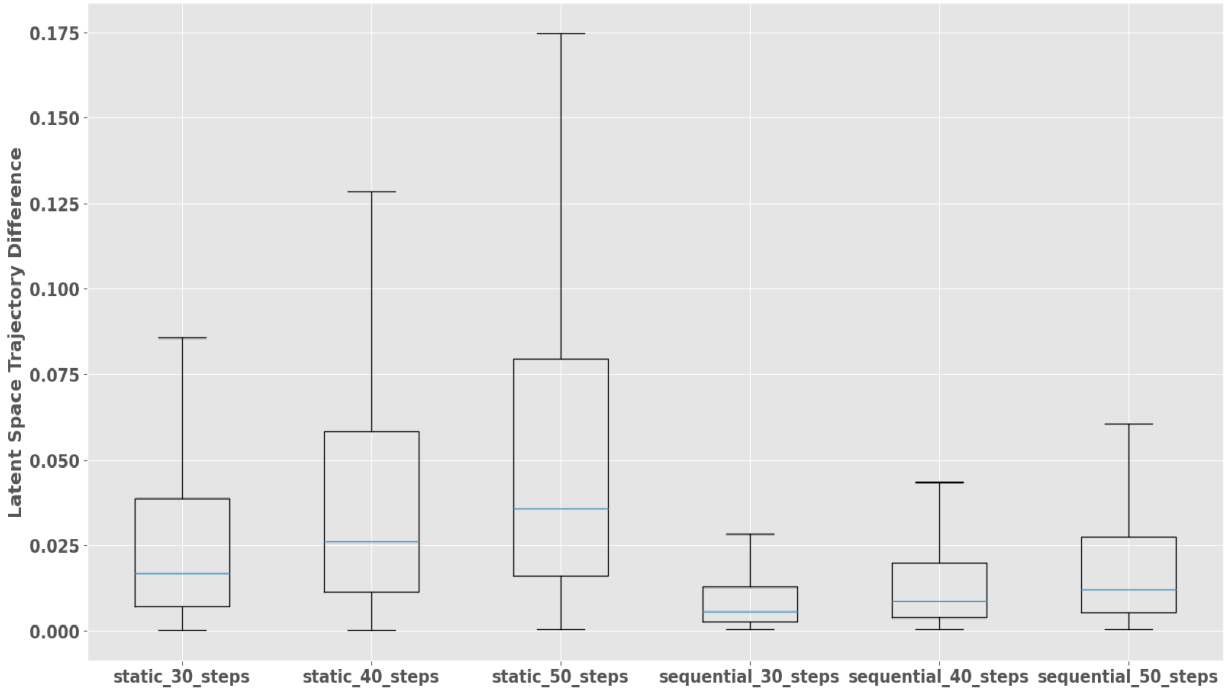

Figure 3.11: The distribution of differences between the original and reconstructed latent space trajectories are shown on a normalized scale to enable the direct comparison of the GP-specific static and sequential inference models. Differences for 50k sample sequences are shown for prediction windows of 30, 40, and 50 datapoints. For each plot, the blue line denotes the mean while the bottom and top bounds of the box denote the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme datapoints that are not considered as outliers.

shown for prediction windows of 30, 40, and 50 datapoints. For each prediction window length, the distributions of differences are narrower and have smaller mean magnitudes for the sequential inference model as compared to the static inference model.

As expected, for both types of models, the distribution of differences widens and the mean magnitude increases as the prediction window length increases. However, the sequential inference model appears to be more robust to increases in prediction window length than the static inference model. The distribution of differences for the largest prediction window for the sequential inference model is still narrower and has a smaller mean magnitude as compared to the smallest prediction window for the static inference model (right-most and left-most boxplots, respectively, in Figure 3.11).

Recall that the SVGP trained for the sequential inference model only used 100 inducing points as compared to the 300 inducing points used for the static inference model. Still, Figures 3.7- 3.11 all indicate that the GP-specific sequential inferential model outperforms the GP-specific static inferential model. By accounting for temporal relationships in the sequences of observation-action pairs, we can more effectively capture the latent space dynamics in a way that can be practically integrated with an optimization-based model predictive control framework.

3.4.6 Performing Model Predictive Control Using a Learned Latent Space Dynamics Model

To demonstrate our complete end-to-end model predictive control framework, we implemented the tractable MPC formulation with the multiple schooting scheme [7], as shown in Eqn. 3.15. Leveraging an automated differentiation tool for nonlinear optimization [2], we used a primal-dual interior point method [110] to solve the non-convex nonlinear programming problem.

We set the planning horizon of the MPC to be 25 datapoints, with the aim of balancing the

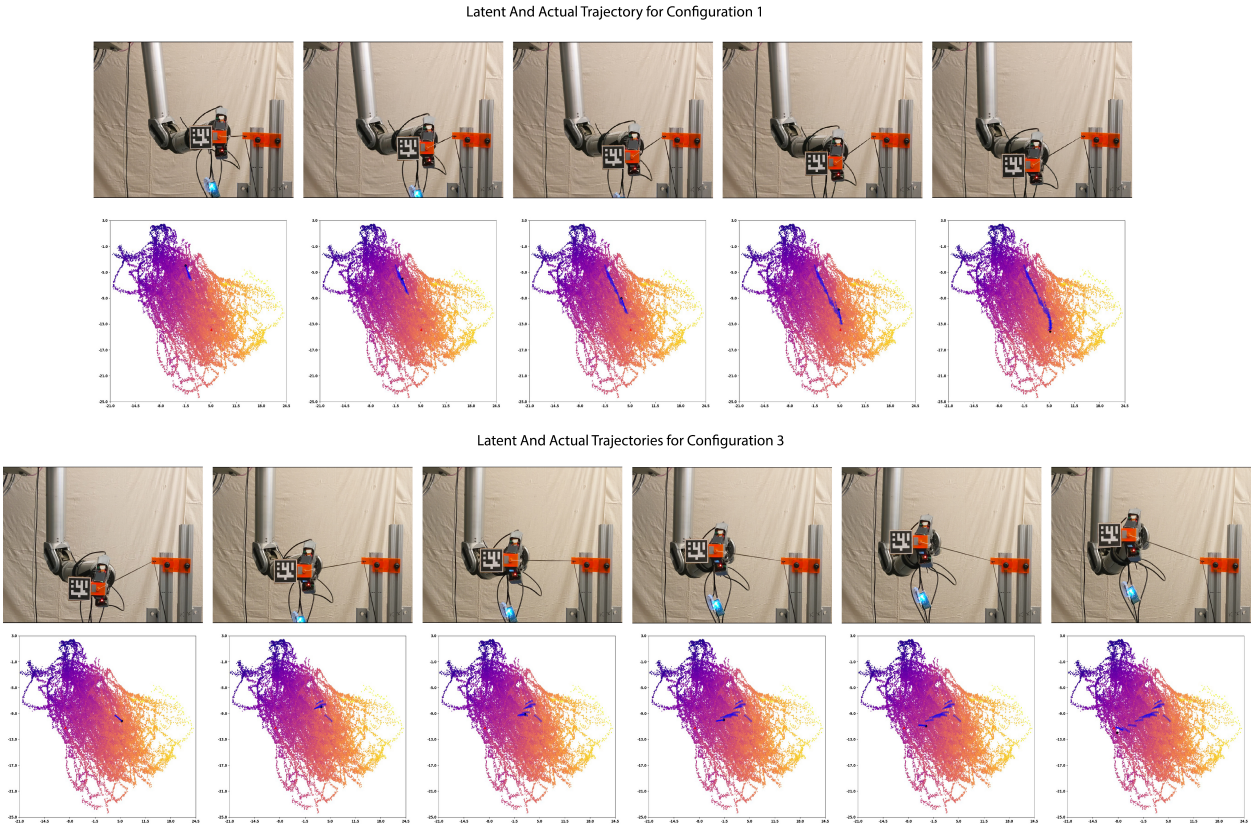

Figure 3.12: Snapshots are shown of the real robot (Rows 1 and 3) and corresponding the latent space (Rows 2 and 4) for each of two manipulation experiments. The first trial required 9 MPC iterations; snapshots are shown for MPC iterations 1, 3, 5, 7 and 9. The second trial required 15 MPC iterations; snapshots are shown for MPC iterations 1, 4, 7, 10, 13, and 15. The black and red circles represent the initial and goal states, respectively, in the latent space after being encoded from the corresponding GelSight marker displace vector fields. The blue solid lines and ellipsoids indicate the predicted trajectory mean and uncertainty, respectively, at each planning step as outputted by the MPC solver.

computational time necessary to solve one MPC planning iteration with the length of the planning horizon. The greater the length of the planning horizon, the greater the computational expense in order to predict one complete sequence.

We evaluated the performance of the MPC after integrating the GP-specific sequential inference model learned in Section 3.4.4 for capturing the latent space dynamics. Each MPC experimental trial consisted of a rollout from a random initial state (cable angle and cable pose) to a random goal state. First, the robot gripper was commanded to an initial configuration. The initial GelSight marker displacement vector field was sent through the recurrent neural network state space model in order to obtain the initial state in the latent space. The initial state in the latent space was used as the initial SVGP state $\mu_0^{\mathbf{x}}$ of the MPC formulation (Eqn. Equation (3.15)). Next, the robot gripper was commanded to a goal configuration. We followed the same procedure as described for the initial configuration in order to obtain the goal SVGP state $\mathbf{x}_g$ in Eqn. Equation (3.15).

Having identified the initial and goal SVGP states in the latent space, we commanded the robot gripper to move back to the initial configuration and begin iteratively solving the MPC problem in order to move to the goal configuration using only the high-dimensional tactile sensor data as inputs. After completing each MPC planning iteration, the resultant control sequence is sent to the robot. After the control action sequence has been applied and the robot gripper has achieved a new configuration, the corresponding state in latent space is obtained in order to initialize $\mu_0^{\mathbf{x}}$ for the next iteration of MPC planning.

Unlike the traditional MPC convention in which only the first control action is sent to the robot while the rest of control action sequence is discarded, we sent the complete control action sequence obtained from the MPC solver to the robot. We elected to sent complete control action sequences to the robot because the SVGP dynamics model is highly nonlinear. Additionally, the SVGP used in this work uses 100 inducing points, which is significantly more than in other applications of

SVGP in MPC frameworks [38]. As a result, the computational time necessary to solve each MPC iteration takes longer (approximately 3 sec) is greater than that for traditional implementations of MPC. To ensure that the robot could complete each tactile-driven manipulation experiment within a reasonable amount of time, we elected to send complete control action sequences to the robot.

We provide both qualitative and quantitative results of the MPC experiments for two representative tactile-driven manipulation experiments. Figure 3.12 shows snapshots of the real robot and the latent space for each of two manipulation experiments. The two trials were selected for visualization purposes because the initial-goal states require different trajectories across the latent space. The black and red circles represent the initial and goal states, respectively, in the latent space after being encoded from the corresponding GelSight marker displace vector fields. The blue solid lines and ellipsoids indicate the predicted trajectory mean and uncertainty, respectively, at each planning step as outputted by the MPC solver.

For the first trial (Rows 1 and 2 of Figure 3.12), the robot required 9 MPC iterations in order to complete the tactile-driven manipulation task. Throughout the trial, the starting point for each MPC iteration was relatively close to the predicted end point of the prior MPC iteration. For the second trial (Rows 3 and 4 of Figure 3.12), the robot required 15 MPC iterations in order to complete the manipulation task. For the early MPC iterations, the starting point for each MPC iteration was relatively far from the predicted end point of the prior MPC iteration. This resulted in a disjointed latent space trajectory for the second trial.

Figure 3.13 shows the distributions of differences between the starting point for each MPC iteration and the predicted end point of the prior MPC iteration. Data are shown for 15 repetitions of each of the two manipulation experiments. The distribution of differences for the second manipulation experiment (corresponding to Rows 3 and 4 in Figure 3.12) is wider and has a higher mean magnitude as compared to the first manipulation experiment (corresponding to Rows 1 and 2 in

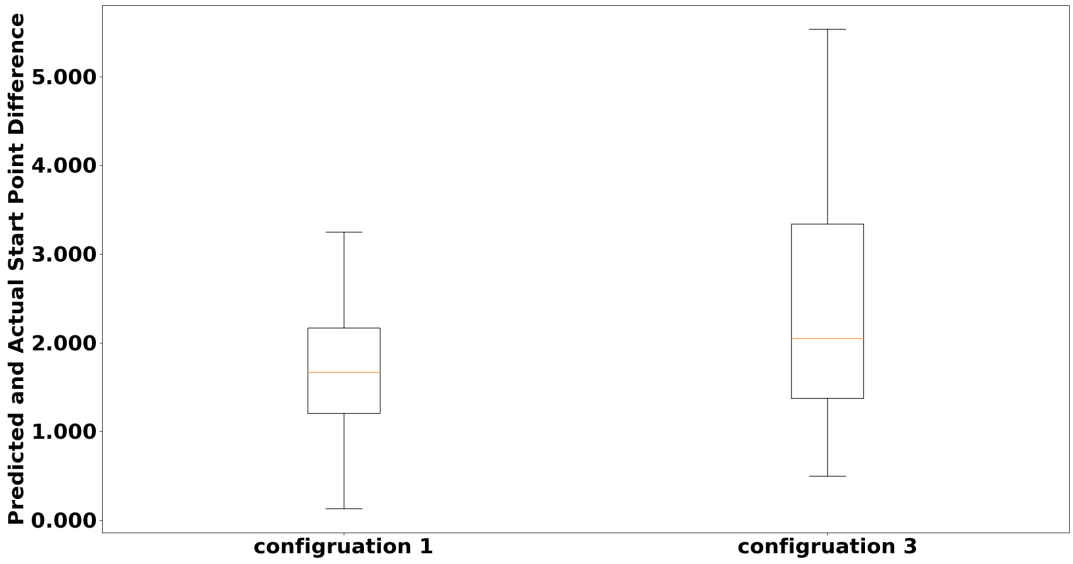

Figure 3.13: The distribution of differences between the starting point for each MPC iteration and the predicted end point of the prior MPC iteration are shown for 15 repetitions of each of the two manipulation experiments shown in Figure 3.12. For each plot, the red line denotes the mean while the bottom and top bounds of the box denote the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme datapoints that are not considered as outliers.

Figure 3.12). Nonetheless, the MPC solver built upon the GP-specific sequential inference model was still successful in generating control action sequences that achieved the tactile-driven manipulation task. Given that this work is one of the few to perform the manipulation of deformable linear objects using high-dimensional tactile sensor data on a real robot, we believe our results demonstrate the effectiveness of our proposed end-to-end model predictive control framework,.

There are a number of limitations that likely contributed to the degraded performance observed for the second manipulation experiment (Rows 3 and 4 in Figure 3.12). First, the training data that was collected per Section 3.4.2 may be insufficient for fully capturing the latent space dynamics. Due to the time-consuming nature of data collection, we elected to sample from a larger region of the workspace of the robot gripper while sacrificing full coverage of the $[0^\circ, 360^\circ]$ velocity commands at each intermediate waypoint between the initial and goal states. This compromise during data collection may have limited the generalizability of the latent space dynamics model when using it for the MPC experiments.

Second, we observed multiple sources of noise in the experimental setup: noise in the tactile sensor data and noise in the robot movements. With regards to noise in the tactile sensor data, the tracking of each individual GelSight marker relies on RGB value magnitude thresholding, which is noisy. As a result, the GelSight marker displacement vectors were observed to oscillate even when the robot was not moving. Such noise-based oscillations were unavoidably encoded into the latent space dynamics.

With regards to noise in the robot movements, we observed shaking of the tendon-driven robot arm when the low-level controller was engaged to send control action sequences to the robot. This phenomenon likely contributed greatly to the differences between the starting point for each MPC iteration and the predicted end point of the prior MPC iteration (Figure 3.13). As an example, even if the robot movement is repeated in order to traverse the same trajectory in the latent space,

the tactile sensor data generated by one long movement will be different from that generated by multiple short movements along the same trajectory. The differences in the observation-action pair sequences can also degrade the performance of the learned latent space dynamics model.

At a more fundamental theoretical level, we did not explicitly regularize or impose any geometric properties on the latent space. According to a recent work in the unsupervised learning community [13], it could be beneficial to craft a latent space structure that might generalize better in terms of encoding the high-dimensional observations from the original physical space.

3.5 Conclusion and Future Work

This work presents a principled framework that leverages well-established system identification and model predictive control theory for the tactile-driven manipulation of deformable linear objects on a real robot. Original observations are in the form of high-dimensional data (260-D) from a camera-based tactile sensor. Using principles and techniques from unsupervised learning and probabilistic generative modelling, the dynamics of how tactile observations change due to robot control actions are identified in a latent space having a much lower dimensionality.

We compared two different inference techniques in order to identify the latent space dynamics. First, we considered a static inference model that treats each observation-action pair from the data sequence as i.i.d. Then we considered a sequential inference model that considers temporal relationships within sequences of data. We concluded that a sequential inference model was more effective at capturing latent space dynamics accurately for the tactile-driven manipulation task in this work.

We used latent space trajectories generated from the trained sequential inference model in order to guide the training of a stochastic variational Gaussian process. Finally, we integrated the SVGP

with an optimization-based MPC framework in order to control the pose and tensile state of an elastic cable in experiments with a real robot.

To the best of our knowledge, we believe that this work includes the first successful demonstration of an end-to-end model predictive control framework for the tactile-driven manipulation of deformable linear objects using a real robot. We believe that this work advances the integration of tools and techniques across different subfields of robotics. For example, we extended the application of unsupervised learning and probabilistic generative modelling techniques from traditional application domains in vision, natural language, and simulated robotics environment [29, 35, 36] to manipulation tasks on real robots. As another example, we integrated well-established technical tools from other subfields in machine learning and controls into tactile-driven robotics research. The principled framework developed for controlling the pose and tensile state of an elastic cable using tactile sensing could inspire and be extended to more complex tactile-driven manipulation tasks.

The limitations of this work motivate a number of future research directions. For example, we observed that the GelSight marker displacement vector data and robot movements were noisy. If modifications to the sensor and robot hardware are not possible, then algorithmic efforts could be dedicated to enhanced filtering of the raw tactile sensor data or modifications to the unsupervised learning process that enable the handling of noisy observation-action sequences.

Another limitation of this work is that, although SVGP is used to alleviate the computational burden of MPC, performing multiple-shooting nonlinear optimization with SVGP is still computationally expensive and time consuming for realtime applications. A fundamental theoretical and practical question arises from this limitation: For a quadratic cost function with system feasibility constraints represented by a SVGP, could a better convergence rate be proven and could a specific nonlinear optimization solver that takes into account the structure of the SVGP dynamics model

be implemented? Recent works related to optimization with Reproducing Hilbert Kernel Space (RHKS) might be useful to consider [5] given that a Gaussian Process is a specific example of RHKS.

Finally, we acknowledge that we constrained the manipulation task to a 2D plane and used a constant speed for all robot gripper motions. This was done in order to make the tactile-driven manipulation task feasible on a real robot and without any simulations available for modeling how the tensile state of an elastic cable affects camera-based tactile sensor data. We believe that the principled framework could be extended to manipulation tasks that involve all 6 DoFs in Cartesian space. One could pursue the representation of the infinite-dimensional configuration space of an elastic cable using a low-dimensional tensile state vector [12, 66]. To release the constraint on the control signal and learn richer system dynamics models, one challenging future direction would be the development of realistic, physics-based simulations of multimodal tactile sensors. Specifically, a combination of machine learning approaches and multiphysics finite element methods [104, 62] could lead to high-impact results that would benefit a number of fields.

3.6 Appendix

This section provides details on the encoder (Table 3.1) and decoder (Table 3.2) architecture for the static inference model described in Section 3.4.3.

We also provide details on the network structure (Figure 3.14) used for the sequential inference model described in Section 3.4.4.

Table 3.1: Encoder architecture for learning latent space dynamics using a static inference model.

Input: stacked Gelsight marker displacement vector fields (batch $\cdot$ 260)
Linear(256 $\rightarrow$ 512) + ReLu
Linear(512 $\rightarrow$ 256) + ReLu
Linear(256 $\rightarrow$ 128) + ReLu
μ: Linear(256 $\rightarrow$ 2), σ: Exp(Linear(256 $\rightarrow$ 2)$\cdot$0.5)

Table 3.2: Decoder architecture for learning latent space dynamics using a static inference model.

Input: 2-dimensional latent state vector
Linear(2 $\rightarrow$ 128) + ReLu
Linear(128 $\rightarrow$ 256) + ReLu
Linear(256 $\rightarrow$ 512) + ReLu
Linear(256 $\rightarrow$ 260)

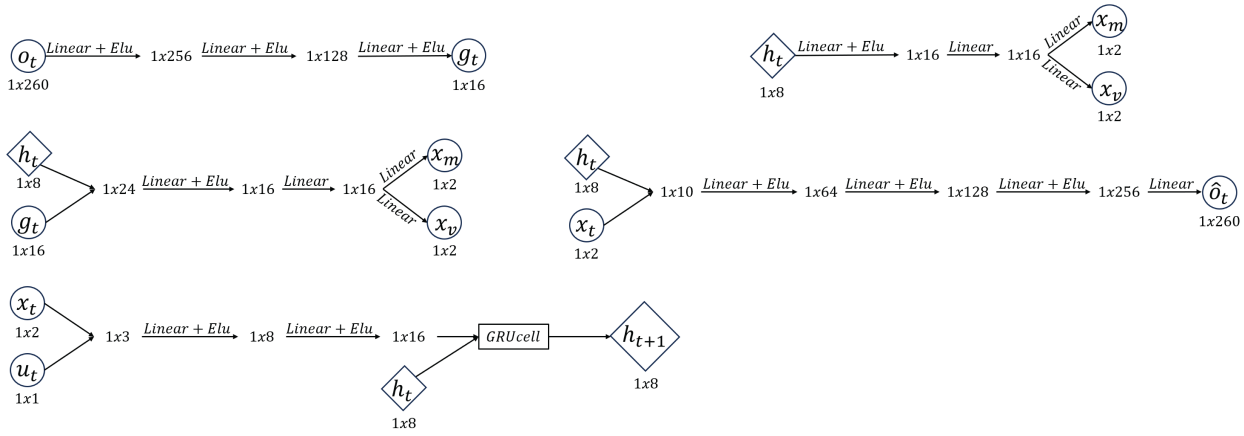

Figure 3.14: The detailed network structure is shown for the sequential inference model described in Section 3.4.4 and on the right of Figure 3.2. The top two networks on the left describe the inference model. The top two networks on the right describe the generative process. The recurrent neural networks on the bottom left connect the inference model and generative process.

CHAPTER 4

Summary and Conclusion

The work presented in this dissertation addresses two fundamental challenges related to the tactile-driven manipulation of deformable objects: (1) Establishing a physics-based system model describing interactions between a deformable sensor and deformable object is very difficult; (2) Modern tactile sensors provide information with high-dimensional data, which is beneficial for perception but impedes the development of effective control methods. By incorporating technical tools from machine learning and optimization, we can circumvent modelling difficulties or construct dynamics models directly from high-dimensional tactile data and use such models for the purpose of planning and control. We believe our work can serve as a foundation for new systematic frameworks that can handle more complex tactile-driven manipulation problems.

4.1 Contributions

Development of a novel model-free framework with reinforcement learning approaches for the manipulation of deformable objects while explicitly leveraging state information encoded in tactile sensor data: We show that a nominal robot trajectory having a functional behavior can be learned using model-free reinforcement learning and a tactile-based reward function. Moreover, we can adapt learned functional behaviors to novel contexts by relying solely on the differences between tactile sensor data generated by a learned nominal trajectory and those gener-

ated for a novel context. Our tactile-driven reinforcement learning approach was demonstrated on a real robot for the manipulation of highly deformable thin-shell objects (notebook pages).

Development of a novel model-based framework for the "end-to-end" tactile-driven manipulation of the pose and tensile state of a deformable linear object: Deep probabilistic general modeling techniques were used to map high-dimensional tactile sensor observations to a low-dimensional latent space. In order to retain representative power of the system dynamics in the low-dimensional latent space, we used recurrent neural networks to account for temporal dependencies of sequential data from real robot sensor-object interactions, instead of treating each observation-action pair independently. The generative neural network was then used to guide the training of a non-parametric supervised learning method, which was finally integrated with an optimization-based, model predictive control framework. Our framework was demonstrated on a real robot for the manipulation of both pose and tension for deformable linear objects (elastic cables).

4.2 Future Work

In this section, we discuss possible future works to advance tactile-driven manipulation capabilities for real robots. Some ideas for future studies are natural extensions of the presented work. Other ideas are interesting and important directions that we think are worth exploring in order to further advance the field of tactile-driven manipulation.

4.2.1 Extending Manipulation Tasks from 2D to 3D and Incorporating Mechanics Models

The tasks presented in Chapters 2 and 3 were purposely constrained to two dimensions in order to make the development and testing of the new frameworks tractable. However, most tasks

in unstructured environments are three-dimensional in nature. The presented frameworks could be extended for 3D manipulation tasks that require the planning and control of manipulation behaviors for all 6 DOFs in the Cartesian space.

Manipulation tasks could also be considered for a larger variety of deformable objects, such as rubber rods and resistance bands. For such deformable objects, one could leverage physics-based models that incorporate well-established mechanics theory for the stretching, bending, and twisting of deformable objects [3, 44]. At this time, there are very few works in the field of robotics that explicitly leverage such energy-based mechanical models for the manipulation of deformable objects [12, 66, 102]. It would be really interesting to investigate how tactile sensing, physics-based models, and control theory could be combined to achieve the tactile-driven manipulations of more complex deformable objects in the full 3D environment.

4.2.2 Hybrid Framework for Simulation of Tactile Sensor Behavior

The construction of accurate simulations of deformable, often multimodal, tactile sensors is a crucial step for pushing the field of tactile-driven manipulation forward. Analytical or physics-based models could open new avenues for learning tactile-based manipulation policies in simulation and transferring the policy learned in simulation to a real robotic system. Moreover, the process of constructing an accurate model could enhance understanding of the physics of tactile sensing mechanisms.

Currently, there are only a few works aimed at exploring this approach [113, 87, 81, 117]. However, the application of the approach has, thus far, been limited to manipulation tasks such as pendulum swing-up, ball repositioning, and peg insertion with rigid objects. An interesting direction to explore would be to combine multi-physics simulation and machine learning [104, 62] in order to build accurate simulations for more complex tactile-driven manipulation tasks, such

as those involving deformable objects. One could also explore trade-offs between spatiotemporal resolution and ease of simulation when designing new tactile sensors.

4.2.3 Learning Policies from Multimodal Sensor Systems

Humans leverages multimodal sensing systems in order to perceive the world and manipulate objects all the time. While tactile sensing provides rich, high-resolution information about finger-object contact, its information is inherently local and expensive to obtain. Policies might be more robust and generalizable if they were learned through a combination of complementary senses such as touch and vision, through which a robot could obtain a more global estimate of the object or environment [88, 17].

REFERENCES

[1] A. Abdolmaleki, R. Lioutikov, J. Peters, N. Lau, L. P. Reis, and G. Neumann. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems 28*, pages 3537–3545. 2015.

[2] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. doi: 10.1007/s12532-018-0139-4.

[3] S. Antman. *Nonlinear Problems of Elasticity*. Applied Mathematical Sciences. Springer New York, 2005. ISBN 9780387276496. URL https://books.google.com/books?id=_MagNyCXNqMC.

[4] Ershad Banijamali, Rui Shu, Mohammad Ghavamzadeh, Hung Hai Bui, and Ali Ghodsi. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, 2017.

[5] Dimitris Bertsimas and Nihal Koduri. Data-driven optimization: A reproducing kernel hilbert space approach. *Operations Research*, 70, 03 2021. doi: 10.1287/opre.2020.2069.

[6] Lars Blackmore, Masahiro Ono, and Brian C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011. doi: 10.1109/TRO.2011.2161160.

[7] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems*. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)61205-9. URL https://www.sciencedirect.com/science/article/pii/S1474667017612059. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[8] P. De Boer, Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. In *Annals of Operations Research*, volume 134, pages 19–67, 2004.

[9] Nathanael Bosch, Jan Achterhold, Laura Leal-Taixé, and Jörg Stückler. Planning from images with deep latent gaussian process dynamics. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 640–650. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/bosch20a.html.

[10] Nathanael Bosch, Jan Achterhold, Laura Leal-Taixé, and Jörg Stückler. Planning from images with deep latent gaussian process dynamics. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger,

editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 640–650. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/bosch20a.html.

[11] T. Bretl and Z. McCarthy. Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *Intl J of Robotics Research*, 33(1):48–68, 2014.

[12] Timothy Bretl and Zoe McCarthy. Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *The International Journal of Robotics Research*, 33(1):48–68, 2014. doi: 10.1177/0278364912473169. URL https://doi.org/10.1177/0278364912473169.

[13] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021. URL https://arxiv.org/abs/2104.13478.

[14] Lars Buesing, Theophane Weber, Sébastien Racanière, S. M. Ali Eslami, Danilo Jimenez Rezende, David P. Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, and Daan Wierstra. Learning and querying fast generative models for reinforcement learning. *CoRR*, abs/1802.03006, 2018. URL http://arxiv.org/abs/1802.03006.

[15] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans Sys, Man, and Cybernetics, Part B (Cybernetics)*, 37(2): 286–298, 2007.

[16] Joaquin Quinonero Candela, Agathe Girard, Jan Larsen, and Carl Edward Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 2, pages II–701. IEEE, 2003.

[17] Arkadeep Narayan Chaudhury, Timothy Man, Wenzhen Yuan, and Christopher G. Atkeson. Using collocated vision and tactile sensors for visual servoing and localization. *IEEE Robotics and Automation Letters*, 7(2):3427–3434, 2022. doi: 10.1109/LRA.2022.3146565.

[18] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal. Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1960–1966, 2016.

[19] Jason Choi, Fernando Castañeda, Claire Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. In *Robotics: Science and Systems*, 07 2020. doi: 10.15607/RSS.2020.XVI.088.

[20] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023. doi: 10.1126/scirobotics.ade2256. URL https://www.science.org/doi/abs/10.1126/scirobotics.ade2256.

[21] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/b618c3210e934362ac261db280128c22-Paper.pdf.

[22] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE Trans Robot*, 26(1):1–20, 2010.

[23] M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2:1–142, 2011.

[24] Marc Deisenroth, Dieter Fox, and Carl Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:408–423, 02 2015. doi: 10.1109/TPAMI.2013.218.

[25] S. Dong and A. Rodriguez. Tactile-based insertion for dense box-packing. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, pages 7953–7960, 2019.

[26] C. Elbrechter, R. Haschke, and H. Ritter. Folding paper with anthropomorphic robot hands using real-time physics-based modeling. In *IEEE-RAS Intl Conf on Humanoid Robots (Humanoids)*, pages 210–215, 2012.

[27] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, and Stéphane Cotin. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In Yohan Payan, editor, *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, volume 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pages 283–321. Springer, June 2012. doi: 10.1007/8415_2012_125. URL https://hal.inria.fr/hal-00681539.

[28] J. Fishel, G. Lin, and G. Loeb. *BioTac® Product Manual, V21*. SynTouch LLC, 2018.

[29] Marco Fraccaro. *Deep Latent Variable Models for Sequential Data*. PhD thesis, Technical University of Denmark, 2018.

[30] Marco Fraccaro, Søren Kaae Sø nderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran As-

sociates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/208e43f0e45c4c78cafadb83d2888cb6-Paper.pdf.

[31] Florian Fuchs, Yunlong Song, Elia Kaufmann, Davide Scaramuzza, and Peter Dürr. Super-human performance in gran turismo sport using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(3):4257–4264, 2021. doi: 10.1109/LRA.2021.3064284.

[32] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[33] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends® in Machine Learning*, 15(1-2):1–175, 2021. ISSN 1935-8237. doi: 10.1561/2200000089. URL http://dx.doi.org/10.1561/2200000089.

[34] Jennifer Grannen, Priya Sundaresan, Brijen Thananjeyan, Jeffrey Ichnowski, Ashwin Balakrishna, Minho Hwang, Vainavi Viswanath, Michael Laskey, Joseph Gonzalez, and Ken Goldberg. Untangling dense knots by learning task-relevant keypoints. In *Conference on Robot Learning*, 2020.

[35] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565, 2019.

[36] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

[37] Negin Heravi, Wenzhen Yuan, Allison M. Okamura, and Jeannette Bohg. Learning an action-conditional model for haptic texture generation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11088–11095, 2020. doi: 10.1109/ICRA40945.2020.9197447.

[38] Lukas Hewing, Alexander Liniger, and Melanie N. Zeilinger. Cautious nmpc with gaussian process dynamics for autonomous miniature race cars. In *2018 European Control Conference (ECC)*, pages 1341–1348, 2018. doi: 10.23919/ECC.2018.8550162.

[39] Sandy H.Huang, Martina Zambelli, Jackie Kay, Murilo F. Martins, Yuval Tassa, Patrick M. Pilarski, and Raia Hadsell. Learning gentle object manipulation with curiosity-driven deep reinforcement learning, 2019. URL https://arxiv.org/abs/1903.08542.

[40] H. Hoffmann, P. Pastor, D. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *IEEE Intl Conf on Robotics and Automation*, pages 2587–2592, 2009.

[41] Hung-Jui Huang, Xiaofeng Guo, and Wenzhen Yuan. Understanding dynamic tactile sensing for liquid property estimation. In *Robotics: Science and Systems*, 2022.

[42] N. Jamali, M. Maggiali, F. Giovannini, G. Metta, and L. Natale. A new design of a fingertip for the icub hand. In *2015 IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, pages 2705–2710, 2015.

[43] J. W. James, N. Pestell, and N. F. Lepora. Slip detection with a biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 3(4):3340–3346, 2018.

[44] M.K. Jawed, A. Novelia, and O.M. O'Reilly. *A Primer on the Kinematics of Discrete Elastic Rods*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, 2018. ISBN 9783319769653. URL https://books.google.com/books?id=0_RZDwAAQBAJ.

[45] Roland Johansson. The effects of anesthesia on motor skills, 2011. URL https://www.youtube.com/watch?v=0LfJ3M3Kn80.

[46] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359, 2009.

[47] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[48] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019. doi: 10.1561/2200000056. URL https://doi.org/10.1561/2200000056.

[49] J. Kober, B. Mohler, and J. Peters. Learning perceptual coupling for motor primitives. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 834–839, 2008.

[50] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *Int J Robot Res*, 32(11):1238–1274, 2013.

[51] A. Kupcsik, M. P. Deisenroth, J. Peters, A. P. Loh, P. Vadakkepat, and G. Neumann. Model-based contextual policy search for data-efficient generalization of robot skills. *Artif Intell*, 247:415–439, 2017.

[52] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47): eabc5986, 2020. doi: 10.1126/scirobotics.abc5986. URL https://www.science.org/doi/abs/10.1126/scirobotics.abc5986.

[53] Michelle A. Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020. doi: 10.1109/TRO.2019.2959445.

[54] Ziwei Lei, Xutian Deng, Yi Wang, Zhihao Li, Xiaohui Xiao, Dong Han, Fei Chen, and Miao Li. A biomimetic tactile palm for robotic object manipulation. *IEEE Robotics and Automation Letters*, 7(4):11500–11507, 2022. doi: 10.1109/LRA.2022.3202653.

[55] N. F. Lepora, A. Church, C. de Kerckhove, R. Hadsell, and J. Lloyd. From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor. *IEEE Robotics and Autom Letters*, 4(2):2101–2107, 2019.

[56] Nathan F. Lepora, Yijiong Lin, Ben Money-Coomes, and John Lloyd. Digitac: A digit-tactip hybrid tactile sensor for comparing low-cost high-resolution robot touch. *IEEE Robotics and Automation Letters*, 7(4):9382–9388, 2022. doi: 10.1109/LRA.2022.3190641.

[57] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, jan 2016. ISSN 1532-4435.

[58] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard. Learning of grasp adaptation through experience and tactile sensing. In *2014 IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 3339–3346, 2014.

[59] Vincent Lim, Huang Huang, Lawrence Yunliang Chen, Jonathan Wang, Jeffrey Ichnowski, Daniel Seita, Michael Laskey, and Ken Goldberg. Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8282–8289, 2022. doi: 10.1109/ICRA46639.2022.9811651.

[60] Bo Lu, Henry K. Chu, K. C. Huang, and Li Cheng. Vision-based surgical suture looping through trajectory planning for wound suturing. *IEEE Transactions on Automation Science and Engineering*, 16(2):542–556, 2019. doi: 10.1109/TASE.2018.2840532.

[61] S. Luo, J. Bimbo, R. Dahiya, and H. Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54 – 67, 2017.

[62] Quan Khanh Luu, Nhan Huu Nguyen, and Van Anh Ho. Simulation, learning, and application of vision-based tactile sensing at large scale. *IEEE Transactions on Robotics*, 39(3):2003–2019, 2023. doi: 10.1109/TRO.2023.3245983.

[63] Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. Non-smooth newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics*, 38:1–20, 10 2019. doi: 10.1145/3338695.

[64] Alexander Graeme De Garis Matthews. *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge, 2017. URL https://www.repository.cam.ac.uk/handle/1810/278022.

[65] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822.

[66] Itamar Mishani and Avishai Sintov. Real-time non-visual shape estimation and robotic dual-arm manipulation control of an elastic wire. *IEEE Robotics and Automation Letters*, 7(1): 422–429, 2022. doi: 10.1109/LRA.2021.3128707.

[67] M. Moll and L. E. Kavraki. Path planning for deformable linear objects. *IEEE Trans Robot*, 22(4):625–636, 2006.

[68] Tetsuya Narita and Oliver Kroemer. Policy blending and recombination for multimodal contact-rich tasks. *IEEE Robotics and Automation Letters*, 6(2):2721–2728, 2021. doi: 10.1109/LRA.2021.3061982.

[69] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019. URL http://arxiv.org/abs/1910.07113.

[70] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic movement primitives. In *Proc Intl Conf on Neural Information Processing Sys - Vol. 2*, page 2616–2624, 2013.

[71] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal. Towards associative skill memories. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 309–315, 2012.

[72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[73] V. Petrík, V. Smutný, P. Krsek, and V. Hlaváč. Physics-based model of a rectangular garment for robotic folding. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, pages 951–956, 2016.

[74] Athanasios S Polydoros and Lazaros Nalpantidis. Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems*, 86(2):153–173, 2017.

[75] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

[76] N. Rudin, Hendrik Kolvenbach, Vassilios Tsounis, and Marco Hutter. Cat-like jumping and landing of legged robots in low gravity using deep reinforcement learning. *IEEE Transactions on Robotics*, 38:317–328, 2021.

[77] J. Sanchez, Juan. Corrales, B. Bouzgarrou, and Y. Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018.

[78] J. Sanchez, C. M. Mateo, J. A. Corrales, B. Bouzgarrou, and Y. Mezouar. Online shape estimation based on tactile sensing and deformation modeling for robot manipulation. In *2018 IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, pages 504–511, 2018.

[79] Jose Sanchez, Carlos M. Mateo, Juan Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Online shape estimation based on tactile sensing and deformation modeling for robot manipulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 504–511, 2018. doi: 10.1109/IROS.2018.8594314.

[80] S. Schaal. *Dynamic Movement Primitives–A Framework for Motor Control in Humans and Humanoid Robotics*, pages 261–280. Springer Tokyo, Tokyo, 01 2006.

[81] Carmelo Sferrazza and Raffaello D'Andrea. Sim-to-real for high-resolution optical tactile sensing: From images to three-dimensional contact force distributions. *Soft Robotics*, 9(5): 926–937, 2022. doi: 10.1089/soro.2020.0213. URL https://doi.org/10.1089/soro.2020.0213. PMID: 34842455.

[82] Carmelo Sferrazza and Raffaello D'Andrea. Design, motivation and evaluation of a full-resolution optical tactile sensor. *Sensors*, 19(4), 2019. ISSN 1424-8220. doi: 10.3390/s19040928. URL https://www.mdpi.com/1424-8220/19/4/928.

[83] A. Shah, L. Blumberg, and J. Shah. Planning for manipulation of interlinked deformable linear objects with applications to aircraft assembly. *IEEE Trans Autom Sci Engin*, 15(4): 1823–1838, 2018.

[84] Y. She, W. Shaoxiong, D. Siyuan, N. Sunil, A. Rodríguez, and E. Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40 (12-14):1385–1401, 2021.

[85] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021. doi: 10.1177/02783649211027233. URL https://doi.org/10.1177/02783649211027233.

[86] Dave Shreiner, Graham Sellers, John M. Kessenich, and Bill M. Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*. Addison-Wesley Professional, 8th edition, 2013. ISBN 0321773039.

[87] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2361–2368, 2022. doi: 10.1109/LRA.2022.3142412.

[88] Paloma Sodhi, Michael Kaess, Mustafa Mukadam, and Stuart Anderson. Learning tactile models for factor graph-based estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13686–13692, 2021. doi: 10.1109/ICRA48506.2021.9561011.

[89] Yunlong Song, HaoChih Lin, Elia Kaufmann, Peter Dürr, and Davide Scaramuzza. Autonomous overtaking in gran turismo sport using curriculum reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9403–9409, 2021. doi: 10.1109/ICRA48506.2021.9561049.

[90] Yunlong Song, Mats Steinweg, Elia Kaufmann, and Davide Scaramuzza. Autonomous drone racing with deep reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1205–1212, 2021. doi: 10.1109/IROS51168.2021.9636053.

[91] F. Stulp and G. Raiola. Dmpbbo: A versatile python/c++ library for function approximation, dynamical movement primitives, and black-box optimization. *Journal of Open Source Software*, 4(37):1225–1228, 2019.

[92] Zhe Su, Oliver Kroemer, Gerald E. Loeb, Gaurav S. Sukhatme, and Stefan Schaal. Learning manipulation graphs from demonstrations using multimodal sensory signals. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2758–2765, 2018. doi: 10.1109/ICRA.2018.8461121.

[93] B. Sundaralingam, A. S. Lambert, A. Handa, B. Boots, T. Hermans, S. Birchfield, N. Ratliff, and D. Fox. Robust learning of tactile force estimation through robot interaction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9035–9042, 2019.

[94] Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Jeffrey Ichnowski, Ellen R. Novoseller, Minho Hwang, Michael Laskey, Joseph Gonzalez, and Ken Goldberg. Untangling dense non-planar knots by learning manipulation features and recovery policies. In *Robotics: Science and Systems*, 2021.

[95] Neha Sunil, Shaoxiong Wang, Yu She, Edward Adelson, and Alberto Rodriguez Garcia. Visuotactile affordances for cloth manipulation with local control. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=s6NEzqZKaP-.

[96] G. Sutanto, Z. Su, S. Schaal, and F. Meier. Learning sensor feedback models from demonstrations via phase-modulated neural networks. In *IEEE Intl Conf on Robotics and Autom (ICRA)*, pages 1142–1149, 2018.

[97] G. Sutanto, N. Ratliff, B. Sundaralingam, Y. Chebotar, Z. Su, A. Handa, and D. Fox. Learning latent space dynamics for tactile servoing. In *IEEE Intl Conf on Robotics and Autom (ICRA)*, pages 3622–3628, 2019.

[98] V. Tangkaratt, H. V. Hoof, S. Parisi, G. Neumann, J. Peters, and M. Sugiyama. Policy search with high-dimensional context variables. In *Proc AAAI Conf on Artificial Intelligence*, page 2632–2638, 2017.

[99] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine. Manipulation by feel: Touch-based control with deep predictive models. In *IEEE Intl Conf on Robotics and Automation (ICRA)*, pages 818–824, 2019.

[100] 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. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 818–824, 2019. doi: 10.1109/ICRA.2019.8794219.

[101] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/titsias09a.html.

[102] Dezhong Tong, Andrew Choi, Longhui Qin, Weicheng Huang, Jungseock Joo, and M Khalid Jawed. Sim2real physically informed neural controllers for robotic deployment of deformable linear objects. *arXiv preprint arXiv:2303.02574*, 2023.

[103] Vassilios Tsounis, Mitja Alge, Joonho Lee, Farbod Farshidian, and Marco Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3699–3706, 2020. doi: 10.1109/LRA.2020.2979660.

[104] Lac Van Duong and Van Anh Ho. Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation. *IEEE Transactions on Robotics*, 37(2):390–403, 2021. doi: 10.1109/TRO.2020.3031251.

[105] H. van Hoof, T. Hermans, G. Neumann, and J. Peters. Learning robot in-hand manipulation with tactile features. In *IEEE-RAS Intl Conf on Humanoid Robots (Humanoids)*, pages 121–127, 2015.

[106] F. Veiga, J. Peters, and T. Hermans. Grip stabilization of novel objects using slip prediction. *IEEE Trans Haptics*, 11(4):531–542, 2018.

[107] F. Veiga, B. Edin, and J. Peters. Grip stabilization through independent finger tactile feedback control. *Sensors*, 20(6), 2020.

[108] F. F. Veiga. *Towards Dexterous In-Hand Manipulation through Tactile Sensing*. PhD thesis, Technische Univ., Darmstadt, July 2018.

[109] Vainavi Viswanath, Kaushik Shivakumar, Justin Kerr, Brijen Thananjeyan, Ellen R. Novoseller, Jeffrey Ichnowski, Alejandro Escontrela, Michael Laskey, Joseph E. Gonzalez, and Ken Goldberg. Autonomously untangling long cables. *ArXiv*, abs/2207.07813, 2022. URL https://api.semanticscholar.org/CorpusID:248942092.

[110] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[111] R. Wang, S. Wang, S. Du, E. Xiao, W. Yuan, and C. Feng. Real-time soft body 3d proprioception via deep vision-based sensing. *IEEE Robotics and Automation Letters*, 5(2): 3382–3389, 2020.

[112] Shaoxiong Wang, Yu She, Branden Romero, and Edward H Adelson. Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[113] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):3930–3937, 2022. ISSN 2377-3766. doi: 10.1109/LRA.2022.3146945. URL https://arxiv.org/abs/2012.08456.

[114] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft Robotics*, 5(2):216–227, 2018.

[115] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pages 2746–2754, 2015.

[116] N. Wettels, V. J. Santos, R. S. Johansson, and G. E. Loeb. Biomimetic tactile sensor array. *Adv Robot*, 22(8):829–849, 2008.

[117] Jie Xu, Sangwoon Kim, Tao Chen, Alberto Rodriguez Garcia, Pulkit Agrawal, Wojciech Matusik, and Shinjiro Sueda. Efficient tactile simulation with differentiability for robotic manipulation. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=6BIffCl6gsM.

[118] Mengyuan Yan, Yilin Zhu, Ning Jin, and Jeannette Bohg. Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE Robotics and Automation Letters*, PP:1–1, 01 2020. doi: 10.1109/LRA.2020.2969931.

[119] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54):eabd8803, 2021. doi: 10.1126/scirobotics.abd8803. URL https://www.science.org/doi/abs/10.1126/scirobotics.abd8803.

[120] H. Yousef, M. Boukallel, and K. Althoefer. Tactile sensing for dexterous in-hand manipulation in robotics—a review. *Sensors and Actuators A: Physical*, 167(2):171 – 187, 2011. Solid-State Sensors, Actuators and Microsystems Workshop.

[121] W. Yuan, M. A. Srinivasan, and E. H. Adelson. Estimating object hardness with a gelsight touch sensor. In *2016 IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, pages 208–215, 2016.

[122] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017.

[123] W. Yuan, Y. Mo, S. Wang, and E. H. Adelson. Active clothing material perception using tactile sensing and deep learning. In *2018 IEEE Intl Conf on Robotics and Automation (ICRA)*, pages 4842–4849, 2018.

[124] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H. Adelson. Active clothing material perception using tactile sensing and deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4842–4849, 2018. doi: 10.1109/ICRA.2018.8461164.

[125] Marvin Zhang, Sharad Vikram, Laura Smith, P. Abbeel, Matthew J. Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, 2018.

[126] Yunbo Zhang, Wenhao Yu, Karen Liu, Charlie Kemp, and Greg Turk. Learning to manipulate amorphous materials. *ACM Transactions on Graphics*, 39:1–11, 11 2020. doi: 10.1145/3414685.3417868.