Accounting for Directional Rigidity and Constraints in Control for Manipulation of Deformable Objects without Physical Simulation

Mengyao Ruan[1], Dale McConachie[1] and Dmitry Berenson[1]

Abstract— Deformable objects like cloth and rope are challenging to manipulate because it is difficult to predict the state of the object given a motion of the gripper(s) holding it. In much previous work, physical models (such as Mass-Spring or Finite-Element) have been used to model such affects. However, these models often require significant parameter tuning for each scenario and can be expensive to simulate inside a control loop. Furthermore, it is difficult to create a practical controller for deformable object manipulation that preserves constraints, especially avoiding overstretching the object. In this paper, we developed a more effective controller than previous work by 1) constructing a more accurate geometric model of how the direction of gripper motion and obstacles affect deformable objects; and 2) specifying a novel stretching avoidance constraint to prevent the object from being overstretched by the robot. Experiments comparing our new method to the previous method in simulation and on a physical robot suggest that our new model captures the behavior of the object more accurately. We also find that our controller is able to prevent tearing that would occur when using the previous method.

I. INTRODUCTION

Manipulation of deformable objects is essential for many applications, such as surgery, doing laundry, and industrial assembly. However, robotic manipulation of deformable objects remains a challenging problem. One of the central difficulties is modeling and predicting the state of the object. The compliant nature of objects like cloth and rope (we do not consider volume-preserving or elastic objects in this work) entails that their motion depend on a large set of physical properties. To model and predict the state of the deformable object, most previous work focuses on constructing a precise physical model, e.g., using Mass-Spring [1] or Finite-Element [2] models. We seek to create a controller for practical deformable object manipulation tasks that does not rely on detailed knowledge of physical properties, i.e. that does not have a highly-accurate physical model. Such models require parameters (both physical and numerical) that are difficult to obtain and are usually more time-consuming to compute, preventing their use in a fast control loop.

Controlling objects such as cloth or rope is made more difficult by the fact that they have an infinite number of degrees of freedom. The system, which consists of a set of grippers which hold the deformable object, is also extremely underactuated. On top of this, it is also challenging to design a practical controller for manipulation that preserves

Fig. 1: An illustrative example of directional rigidity. Left: The rope moves almost rigidly when dragging it by one end to the left. Right: The rope deforms when pulling it on the right in the opposite direction.

constraints, such as the avoidance of overstretching of the object, due to a lack of direct control of deformation.

The model in this paper will not involve precise physical simulation. Instead, we aim to construct an approximate model that predicts the object motion based on geometric information, such as the current poses of the grippers and the current configuration of the deformable object. Previous work on this kind of modeling [3] approached the problem by exploiting a property called *diminishing rigidity*, which assumes that the effect of gripper motion along the deformable object diminishes as the distance from the gripper increases. This method has shown that it is possible to do practical manipulation tasks with a deformable object without a precise physical model. However, we observe that this rigidity does not only diminish as the distance from the gripper increases. Instead, it is a function of a larger set of variables derived from the configuration of the object. The rigidity also depends on the direction of gripper motion. Fig.1 shows an example of an object's directional rigidity.

Thus the key contributions of this paper are: 1) A geometric model for deformable object manipulation that is computationally efficient to use in a control loop while capturing the directional rigidity effects more accurately than that in [3]; and 2) A controller that addresses the excessive stretching constraint by better approximating which directions of motion lead to more/less stretching. This new constraint is especially important when the object is caught on an obstacle, which the method in [3] does not address.

Since we build on the idea described in [3], we use this method as a benchmark in our experiments. We compared [3] with our proposed method in several simulation and physical robot experiments. We found that the proposed model is able to predict the object motion better than the previous model and that the new controller is more effective at preventing overstretch than the method in [3].

In the remainder of the paper, we discuss related work and present the problem statement. We then describe our new model formulation and how to formulate the constraints

for the controller. Finally, we describe the results of running this method in several test scenarios.

II. RELATED WORK

Deformable object manipulation has been studied in many contexts ranging across domestic, surgical, and industrial domains [4]. Much of this work relies on accurate modeling and simulation of deformable objects. Some of the most common simulation methods use Mass-Spring Models [1], [5] and Finite-Element Models (FEM) [2], However, such approaches usually require significant tuning and are very sensitive to discretization parameters. Also, we seek a model that can be evaluated very quickly inside an optimal control framework, and Finite-element models, while accurate, can be computationally-expensive to simulate. While methods have been developed to track objects using FEM in real-time [6], a controller may need to evaluate the model many times to find an appropriate command, requiring speeds faster than real-time. Specialized models have also been developed, e.g., [7] and [8] focus on elastic rods that are not in contact. We seek a model that works well with rope- and cloth-like materials that can deform as a result of contact. Finally, researchers have also investigated automatic modeling of deformable objects [9], [10]. However, these methods rely on a time-consuming training phase for each object to be modeled, which we would like to avoid.

Given a model such as those above, researchers have investigated various control methods to manipulate deformable objects. Hirai and Wada proposes an iterative visual-servoing controller in [11], which aligns interest points on the deformable object to targets. Its control law is based on modeling the deformable object as a lattice of interconnected springs. [12] uses a similar spring model to formulate a PID controller that aligns interest points to targets. While effective, these methods rely on models like those above and thus suffer from the above issues.

Our work is complementary to methods that adapt the model of the object during manipulation [13]–[15]. Our model can serve as an initial guess and a reference for such methods so that the online adaptation process does not diverge too far from a reasonable model as a result of perception or modeling error. Our paper builds on the idea of diminishing rigidity Jacobians [3] by improving the model and formulating appropriate constraints for deformable object manipulation. [16] extended [3] to use multiple Jacobian-based models when it is uncertain which model is appropriate for the object. The new model proposed here could be incorporated into the framework of [16].

The model and controller in this paper are built on the method for deformable object manipulation in [3] and task space control of robot manipulators [17]. However, unlike [3], we explore the directional diminishing rigidity of the deformable object. This maintains a low computational cost of prediction while improving accuracy.

Much previous work in this area focuses on completing a task but does not consider constraints on obstacle avoidance or compensation for excessive stretching [11], [12]. [3]

does consider these constraints but its stretching-avoidance method does not handle cases where the object is caught on an obstacle because it does not reason about the way strain propagates through the object as a result of gripper motion. We consider this issue in our paper.

III. PROBLEM STATEMENT

Let a set of P points $\mathcal{P} \subset \mathbb{R}^{3P}$ be the configuration of the deformable object, where these points are either in the curve of a 1D object (e.g., rope) or on the surface of the 2D object (e.g., cloth). This paper focuses on 1D and 2D objects like cloth and rope. We do not consider volume-preserving or elastic objects. Denote the object velocity as $\dot{\mathcal{P}} = [\dot{p}_1^T, ..., \dot{p}_P^T]^T \in \mathbb{R}^{3P}$. Let $q \in SE(3)^G$ be the configuration of a set of G grippers. Denote the velocity of the robot to be $\dot{q} = [\dot{q}_1^T, ...\dot{q}_G^T]^T$ and the velocity of the gth gripper $\dot{q}_g = [v_g^T \ \omega_g^T]^T \in \mathfrak{se}(3)^G$, where $\mathfrak{se}(3)$ is the tangent space of $SE(3)$ [18], v_g is the translation components, and ω_g is the rotation components.

Let ϕ be the mapping such that $\dot{\mathcal{P}} = \phi(q, \dot{q}, \mathcal{P}, \mathcal{O})$, where $\mathcal{O}$ denotes the configurations of obstacles in the environment and $\mathcal{P}$ is the object configuration at the current state. Let C_c denote the set of gripper motions $\dot{q}$ which preserve the gripper collision constraint and C_s denote the set of $\dot{q}$ which preserve the object stretching constraint.

In this paper we assume that the grippers are free-floating and we have a method of sensing $\mathcal{P}$. Inverse-kinematics can be used to track these gripper motions with robot arms. We do not consider stretching as a result of friction, as friction forces are negligible when compared with the stretching induced by the robot pulling on the object for the tasks we consider.

Overall, our goal is to move the G grippers such that the motion of points in $\mathcal{P}$ best matches commanded motions $\dot{\mathcal{P}}_d$, without violating constraints. To achieve this, we must address two fundamental problems: 1) Design an approximate model $\tilde{\phi}$ which is practical to be used by the robot controller with improved accuracy compared to the benchmark [3]. 2) Prevent robot collision and excessive stretching of the object by specifying a set of constraints $C = C_c \cap C_s$. We formalize this control problem as a constrained optimization problem:

$$
\begin{aligned}
\underset{\dot{q}}{\text{minimize}} \quad & ||\dot{\mathcal{P}}_d - \tilde{\phi}(q, \dot{q}, \mathcal{P}, \mathcal{O})|| \\
\text{subject to} \quad & \dot{q} \in C, \\
& ||\dot{q}||\Delta t < \Delta_{q,\max}.
\end{aligned}
\tag{1}
$$

IV. METHODS

To fully specify the problem in Eq. 1, we need to construct the approximate model $\tilde{\phi}$ and define the set of constraints C. A highly accurate model for object motion can require a high computational cost when using FEM simulation. To make a model practical to be used by a robot controller with lower computational cost, we build on the idea of constructing a geometry-based model [3].

To create a more accurate model, we need to improve the accuracy of the $\tilde{\phi}$ approximated in [3] by capturing

more accurate behavior of the object and addressing contact with the environment. To that end, we design a model accounting for directional rigidity and obstacle-penetration constraints (Section IV-A). We then define the collision avoidance constraint C_c and object over-stretching constraint C_s in the robot's task space (Section IV-B).

A. Geometric Model of Deformable Object Motion

1) Model Overview: The approximate model locally describes the object's motion when given a motion of the grippers. Below we describe our approach first by specifying the model and then enforcing collision constraints on the prediction this model makes.

The current state of the deformable object is a function of the current gripper pose, the history of gripper motions that have been applied, the object's initial configuration, and the obstacles in the environment:

$$\mathcal{P} = F(q, Q_{hist}, \mathcal{P}_0, \mathcal{O}) \tag{2}$$

To create a predictive model, we need to compute how $\mathcal{P}$ changes as a result of changing the gripper pose. Taking the time derivative of the above we obtain

$$\frac{\partial \mathcal{P}}{\partial t} = \frac{\partial F}{\partial q}\frac{\partial q}{\partial t} + \frac{\partial F}{\partial Q_{hist}}\frac{\partial Q_{hist}}{\partial t} + \frac{\partial F}{\partial \mathcal{P}_0}\frac{\partial \mathcal{P}_0}{\partial t} + \frac{\partial F}{\partial \mathcal{O}}\frac{\partial \mathcal{O}}{\partial t} \tag{3}$$

Only the first term is non-zero, thus

$$\dot{\mathcal{P}} = \frac{\partial F(q, Q_{hist}, \mathcal{P}_0, \mathcal{O})}{\partial q}\dot{q} \tag{4}$$

$\frac{\partial F}{\partial q}$ is a matrix we call J. In previous work [3], J is assumed to be independent of $\dot{q}$ and $\mathcal{O}$, yielding[1] $\frac{\partial F}{\partial q} = J(q, Q_{hist}, \mathcal{P}_0) = J(q, \mathcal{P})$, which is analogous to a rigid-body Jacobian. While these assumptions allow a linear relationship between $\dot{q}$ and $\dot{\mathcal{P}}$, and thus computational convenience, they are not accurate in many situations (see Figure 1 for an example). In this paper we change the definition of J to the following:

$$\dot{\mathcal{P}} = J(q, \dot{q}, \mathcal{P}, \mathcal{O})\dot{q} = \tilde{\phi}(q, \dot{q}, \mathcal{P}, \mathcal{O}) \tag{5}$$

We now describe how J is computed, focusing on how it accounts for directional rigidity (using $\dot{q}$) and how it enforces obstacle penetration constraints (using $\mathcal{O}$).

2) Directional Rigidity: We build on the idea proposed in [3], which approximates J based on the observation that the deformable object behaves rigidly near points grasped by the robot grippers. [3] encoded this effect through a simple function that only considered the distance of a point from the the nearest gripper. However, we find that we can exploit geometric information in the object's configuration to better predict the object's motion when we use a more complex model. We have observed that the key features of

the deformable object configuration for predicting its motion are its deformability (which is determined by its material properties) and where it is slack. The deformation influences the transmission of the force from the grippers, i.e. the more stretchable the object, the more it will stretch when force is applied. However, when a region of the object is taut, regardless of how stretchable it is, it will move as if it were rigidly connected to a gripper (e.g. imagine a rope held taut by two grippers). We also must take into account that points are not influenced equally by different grippers, i.e. grippers farther away contribute less to the motion of a point than those closer to it.

To incorporate the above effects into our model, we define the following variables, which can be derived from $q, \dot{q}$, and $\mathcal{P}$:

- ρ_{ij}: the geodesic distance (a scalar) between points p_i and p_j on the surface of the object.
- d_{ij}: the vector starting at a point p_i and ending at the point p_j, as shown in Fig. 2
- q_g: the configuration of the gth gripper
- $\dot{q}_g$: the velocity of the gth gripper.

Furthermore, let $c(i, g)$ be the index of the point with the minimal geodesic distance to p_i among the ones grasped by the gth gripper. We address the notion of rigidity in object motion by considering the slackness of the object and reformulating the rigidity as a function of $\rho_{ic(i,g)}$, $d_{ic(i,g)}$, and $\dot{q}_g$. For each point p_i we compute

$$J^{dr}(q_g, \dot{q}_g, p_i) =$$
$$\theta_{i,g}[w_{t,i,g}J_t^{dr}(q_g, \dot{q}_g, p_i), w_{r,i,g}J_r^{dr}(q_g, \dot{q}_g, p_i)]$$
$$w_{t,i,g} = w_t(\rho_{ic(i,g)}, d_{ic(i,g)}, \dot{q}_g)$$
$$w_{r,i,g} = w_r(\rho_{ic(i,g)})$$
$$J_t^{dr}(q_g, \dot{q}_g, p_i) = \mathcal{I}_{3\times3}$$
$$J_r^{dr}(q_g, \dot{q}_g, p_i) = [R_g[1] \times r, R_g[2] \times r, R_g[3] \times r] \tag{6}$$

where $R_g[n]$ is the nth column of the rotation matrix of the gth gripper at the current configuration, $r = p_{c(i,g)} - v_g$, with v_g being the translation of the gth gripper. $w_{t,i,g}$ and $w_{r,i,g}$ are the corresponding translational and rotational diminishing rigidity factors defined by the point p_i and gth gripper (discussed below).

Our goal is to encode the directional rigidity of the object motion into $w_{t,i,g}$ and $w_{r,i,g}$ and use $\theta_{i,g}$ to describe the *influence* of gripper g on p_i. Intuitively, $w_{r,i,g}$ should decrease with the increasing geodesic $\rho_{ic(i,g)}$ distance between p_i and $p_{c(i,g)}$. This is because the deformation of the region between p_i and $p_{c(i,g)}$ will attenuate the transmitted force of the gripper's motion unless the object is taut. Since the effects on $w_{r,i,g}$ from $\dot{q}_g$ and $d_{i,c(i,g)}$ are not as clear or significant as $\rho_{ic(i,g)}$, we keep the $w_{r,i,g}$ as a function of $\rho_{ic(i,g)}$, where

1) $w_{r,i,g}$ ranges between 0 and 1.
2) $w_{r,i,g}$ decreases as $\rho_{ic(i,g)}$ increases.

We give the definition of $w_{r,i,g}$ below.

---

[1]In this equation we replaced Q_{hist} and $\mathcal{P}_0$ with $\mathcal{P}$, the current configuration of the object. Q_{hist} and $\mathcal{P}_0$ are need in F to compute the current state of the object, but if we can sense $\mathcal{P}$ directly (as we assume), then Q_{hist} and $\mathcal{P}_0$ are not needed to compute J.

From observation, we find two key reasons related to the slackness of the object that induce the diminishing rigidity effect for translation motion, and we aim to encode these factors into $w_{t,i,g}$. The first case is that the moving direction of $\dot{q}_g$ makes the region on the object between p_i and $p_{c(i,g)}$ less taut. The second case is that this region is already slack. $w_{t,i,g}$ is thus a product of two terms:

$$w_{t,i,g} = \alpha_{i,g} \cdot \beta_{i,g} \tag{7}$$

where $\alpha_{i,g}$ addresses the effect in the first case (motion reducing tension), and $\beta_{i,g}$ addresses the effect in the second case (object slackness). Both $\alpha_{i,g}$ and $\beta_{i,g}$ are functions of some of $q_g, \dot{q}_g, p_i$, or variables derived from these.

For p_i on the object, we find $\alpha_{i,g}$ is greatly impacted by $d_{ic(i,g)}$ and $\dot{q}_{ic(i,g)}$. Decomposing $\dot{q}_g$ into $\dot{q}_{g,r}$, the radial component in the direction of $d_{ic(i,g)}$, and $\dot{q}_{g,t}$, the transverse component perpendicular to $d_{ic(i,g)}$. We observed that if $\dot{q}_{g,r}$ is in the opposite direction to $d_{ic(i,g)}$, then it is more likely to make the intervening region slacker and thus reduce the transmission of force from the gripper to p_i. Moreover, if $\dot{q}_{g,r}$ and $d_{ic(i,g)}$ are in the same direction when the object is not already slack, p_i can move almost rigidly with $\dot{q}_g$. Fig.1 shows an example of the impact of this alignment. Based on these observations, we design the function $\alpha_{i,g} = \alpha(d_{ic(i,g)}, \dot{q}_g)$ with the following properties:

1) $\alpha(d_{ic(i,g)}, \dot{q}_g)$ ranges between 0 and 1.
2) $\alpha(d_{ic(i,g)}, \dot{q}_g) > \alpha(d_{jc(j,g)}, \dot{q}_g)$ if $\langle d_{ic(i,g)}, \dot{q}_g \rangle > \langle d_{jc(j,g)}, \dot{q}_g \rangle$ and $\rho_{ic(i,g)} = \rho_{ic(i,k)}$.
3) $\alpha(d_{ic(i,g)}, \dot{q}_g) > \alpha(d_{jc(j,g)}, \dot{q}_g)$ if $\langle d_{ic(i,g)}, \dot{q}_g \rangle = \langle d_{jc(j,g)}, \dot{q}_g \rangle$ and $\rho_{ic(i,g)} > \rho_{ic(i,k)}$.

We give the definition of $\alpha(d_{ic(i,g)}, \dot{q}_g)$ below.

As mentioned above, $\beta_{i,g}$ depends on the current slackness of the intervening region. Without other external forces applied on the object, the pulling force applied by the robot will tend to unwind or unfold the object eventually (we do not consider cases where the object is tied into knots). For this reason, the part of the intervening region on the object that is not already spread out is less likely to move rigidly with the gth gripper. One indicator that can address this property is the ratio between the Euclidean distance between p_i and $p_{c(i,g)}$, and the geodesic distance $\rho_{ic(i,g)}$ between them. We denote $\mathfrak{r}_{i,g} = \frac{\|d_{ic(i,g)}\|}{\rho_{ic(i,g)}}$ to be this ratio. A larger $\mathfrak{r}_{i,g}$ indicates a tauter intervening region. A tauter intervening region is more likely to result in $\dot{p}_i$ moving more rigidly. Thus we can design the function $\beta_{i,g} = \beta(\mathfrak{r}_{i,g})$ with the following properties:

1) $\beta(\mathfrak{r}_{i,g})$ ranges between 0 and 1.
2) $\beta(\mathfrak{r}_{i,g}) = 1$ if $\mathfrak{r}_{i,g} = 1$.
3) $\beta(\mathfrak{r}_{i,g}) > \beta(\mathfrak{r}_{i,g})$ if $\mathfrak{r}_{i,g} > \mathfrak{r}_{i,g}$

Finally, $\theta_{i,g}$, which captures the influence of gripper g on point p_i should have the following property (where k is the index of a different gripper on the robot):

1) $\theta_{i,g}$ ranges between 0 and 1.
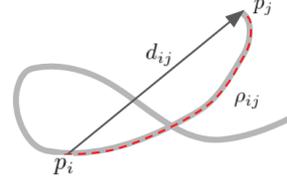2) $\theta_{i,g} < \theta_{i,k}$ if $\rho_{ic(i,g)} > \rho_{ic(i,k)}$.

Fig. 2: The length of the the red segment on the rope is the geodesic distance ρ_{ij}. d_{ij} is the vector showing the relative position of p_j with respect to p_i.

3) $\sum_{m=0}^{G} \theta_{i,m} = 1$.

Through experimentation, we obtained good results with the following functions:

$$\alpha = e^{k_g \rho_{ic(i,g)} (\cos \angle (d_{i,c(i,g)}, \dot{q}_g) - 1)}$$

$$\beta(\mathfrak{r}_{i,g}) = \left(\frac{\|d_{ic(i,g)}\|}{\rho_{ic(i,g)}} \right)^{k_D}$$

$$w_r(\rho_{ic(i,g)}) = e^{-k_r \rho_{ic(i,g)}} \tag{8}$$

$$\theta_{i,g} = \frac{x_g}{\sum_{m=0}^{G} x_m}$$

$$x_m = \frac{\min\{\rho_{ic(i,1)}, ..., \rho_{ic(i,G)}\}}{\rho_m}$$

where k_g, k_D, and k_r are non-negative parameters. Specifically, a larger k_g indicates a greater impact on the diminishing in the rigidity from the motion reducing tension. A larger k_D indicates a greater impact on the diminishing in the rigidity from the slackness of the object in the current state. A larger k_r indicates a faster decrease in the rigidity of J_r^{dr} as the distance from p_i to the gripper increases.

3) Obstacle Penetration Constraints: With the model developed above, we get a prediction of a point's movement from $\tilde{p}_i = J^{dr}(q_g, \dot{q}_g, p_i)\dot{q}$. However, at this stage, we haven't take into account the effect from the obstacles $\mathcal{O}$. Thus the predicted $\tilde{p}_i$ can move p_i into an obstacle.

When the prediction of p_i enters the obstacle, we project any penetration by the predicted $\tilde{p}_i$ into the tangent space of the obstacle surface. Let $d_{i,m} < \|\tilde{p}_i\|$ be the distance to collision in direction $\tilde{p}_i$ from point p_i; let $\lambda_i = \frac{d_{i,m}}{\|\tilde{p}_i\|}$; let $\mathbf{n}_i$ be the unit surface normal of the obstacle in contact; and let $N_i = (I_{3 \times 3} - \mathbf{n}_i \mathbf{n}_i^+)$. Then to account for obstacles we compute

$$J(q_g, \dot{q}_g, p_i, \mathcal{O}) =$$
$$\begin{cases} (\lambda_i + (1 - \lambda_i)N_i)J^{dr}(q_g, \dot{q}_g, p_i) & \text{if } p_i + \tilde{p}_i \text{ in collision} \\ J^{dr}(q_g, \dot{q}_g, p_i) & \text{otherwise} \end{cases}$$
$$\tag{9}$$

To generate J for all the points and grippers we compute $J(q_g, \dot{q}_g, p_i, \mathcal{O})$ for each p_i and then stack these matrices to obtain $J(q_g, \dot{q}_g, \mathcal{P}, \mathcal{O})$. We then compute $J(q_g, \dot{q}_g, \mathcal{P}, \mathcal{O})$ for every g and concatenate the results to obtain $J(q, \dot{q}, \mathcal{P}, \mathcal{O})$. Finally, we arrive at our approximate model: $\tilde{\phi}(q, \dot{q}, \mathcal{P}, \mathcal{O}) = J(q, \dot{q}, \mathcal{P}, \mathcal{O})\dot{q}$.

B. Collision and Stretching Constraints

Applying $\dot{q}$ without constraints in the controller can damage the robot due to collision or tear the deformable object

due to the excessive strain. We describe the constraints we impose to avoid these issues below.

1) Collision: Collision avoidance for the robot is addressed by the constraint C_c, which is the set of motions that keeps the grippers away from obstacles:

$$C_c = \left\{ \dot{q}_g \in \mathfrak{se}(3) \mid l_c(g) - l(q_g) - \frac{\mathfrak{n}(q_g) \cdot \dot{q}_g}{||\dot{q}_g||} \dot{q}_g \Delta t < 0 \right\} \tag{10}$$

where $l(q_g)$ is the function returning the distance from the gripper to its closest obstacle. $l_c(g)$ returns the minimal safe distance allowed between the gth gripper and an obstacle. $\mathfrak{n}(q_g)$ returns the unit surface normal of the obstacle closest to the gth gripper. The idea is to make each gripper keep at least the safe distance away from the closest obstacle. While we consider free-flying grippers in this paper, similar constraints can be imposed on the entire geometry of a robot arm (one constraint per link) to avoid collisions all along the arm.

2) Overstretch: The stretching avoidance of the deformable object is more difficult to formulate due to the compliant nature and the lack of control of the deformable object. In [3], a stretching correction term $\dot{\mathcal{P}}_s \in \mathbb{R}^{3P}$ is applied when the object becomes overstretched. However, this method cannot handle cases where the object is caught on an obstacle.

We detect the overstretching (i.e. excessive strain) of the object by examining the value of the stretching ratio γ, which denotes the maximum value among the ratio between the Euclidean distance $||d_{ij}||$ and the geodesic distance ρ_{ij} for every pair of points p_i and p_j:

$$\gamma = \max_{p_i, p_j \in \mathcal{P}, i \neq j} \frac{||d_{ij}||}{\rho_{ij}} \tag{11}$$

we denote γ_s as the maximum allowed stretching ratio. Our method initiates stretching-avoidance when $\gamma > \gamma_s$.

We assume that the object starts in an unstretched state, so the overstretch that arises is due to the motion of the grippers. Thus if we can constrain gripper motions to a set which does not overstretch the object further than a threshold, we can prevent or reduce the overstretch at the next time step. We know that the force causing the overstretch comes from the grippers, so if we reduce the length of geodesic paths through the object between grippers, the strain on the object should decrease. When overstretch is detected, we thus introduce a conical constraint for each gripper that shrinks the allowable $\dot{q}_g$ to reduce the length of the geodesics between the grippers.

A conical constraint is constructed for each gripper and points along the *stretching avoidance vector*, which is an estimation of the direction to move to decrease the strain. For a pair of grippers with index g and k, two stretching avoidance vectors are defined, one for each gripper. Let $\mathcal{I}_g(q_g, q_k)$ be the index of the point grasped by the gth gripper, which has the minimum geodesic distance to the set of points grasped by q_k. We define $\mathcal{I}_k(q_g, q_k)$ similarly. Let $\int_{gk}$ be the geodesic on the object from $p_{\mathcal{I}_g(q_g, q_k)}$ to $p_{\mathcal{I}_k(q_g, q_k)}$. We denote u_g^k and u_k^g as the pair of stretching
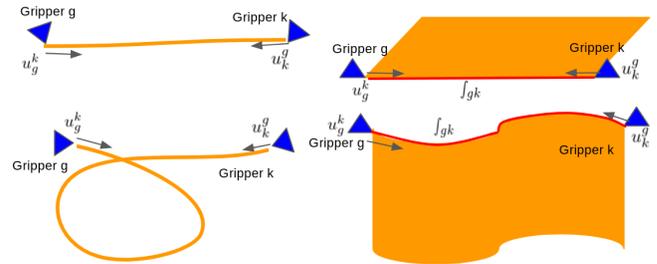

Fig. 3: The arrows in gray show the direction of each stretching vector at the corresponding gripper with respect to the gripper pair q_g and q_k. Left: stretching vectors on the rope when the rope is at rest (above) or is deformed (below). Right: stretching vectors on the cloth when the cloth is at rest (above) or is deformed (below). The red lines denote the geodesic connecting the corresponding $p_{\mathcal{I}_g(q_g, q_k)}$ and $p_{\mathcal{I}_k(q_g, q_k)}$ on the object.

avoidance vectors on the gth and kth grippers. Then u_g^k is the tangent vector of $\int_{gk}$ at $p_{\mathcal{I}_g(q_g, q_k)}$ and u_k^g is the tangent vector of $\int_{kg}$ at the point $p_{\mathcal{I}_k(q_g, q_k)}$ (as shown in Fig. 3).

To specify the stretching constraint, we first define the function $s(\dot{q}_g, q_g, q_k, \mathcal{P})$, which specifies the constraint on gripper g defined by the interaction of grippers g and k. Correspondingly, u_g^k is the stretching avoidance vector for the gth gripper, which is the tangent vector of $\int_{gk}$ at $p_{\mathcal{I}_g(q_g, q_k)}$. The larger the value of s, the more we expect geodesic path length between grippers will be reduced. Thus, s should increase as $\angle(\dot{q}_g, u_g^k)$ increases. Assume we wish to have a lower bound s_s on s, then C_s is a set of constraints $C_s = \{C_s^1, ... C_s^G\}$, where each constraint is:

$$C_s^g = \{\dot{q}_g \in \mathfrak{se}(3) \mid \forall k \neq g, \ s_s - s(\dot{q}_g, q_g, q_k, \mathcal{P}) < 0\} \tag{12}$$

Many functions can satisfy the requirements of s. In our work, we specify the function as:

$$s(\dot{q}_g, q_g, q_k, \mathcal{P}) = \cos \angle(\dot{q}_g, u_g^k) \tag{13}$$

C. Optimization Method

Because our objective function is not necessarily convex, we used a custom optimization method to solve the problem specified in Eq. 1. Our method is a type of numerical gradient descent with an additional projection step to enforce constraints.

Our method's outer loop computes the numerical gradient of the objective function. An inner loop then performs backtracking line search to find the gradient step size. However, the gradient step may cross a constraint boundary, thus after we compute the step size, we check if any constraint has been violated after taking the step. If it has, we project the step back to the feasible space. A simple projection to the boundary of a violated constraint may satisfy that constraint but violate others. Instead, to perform the projection, we solve a convex optimization problem (using the Gurobi optimizer [19]) to find the nearest feasible point. This is possible because all the constraints in our problem are convex. Once

such a point is found, the outer loop continues to iterate until convergence.

V. RESULTS

Our two goals are improving the accuracy of the deformable object motion model (for use in the controller), and formulating a set of constraints for the controller to mitigate collision and excessive stretching issues. As mentioned in previous sections, our benchmark model and controller is [3]. To evaluate our method we perform experiments in simulation and on a physical robot. The simulator used is Bullet physics [20], however, we emphasize that our method has no knowledge of the simulation parameters or simulation methods used therein. The simulator is used as a "black-box," mainly to stand in for a perception system and to allow us to do repeatable experiments. The physical robot consists of two KUKA iiwa 7DoF arms with Robotiq 3-finger hands.

We ran experiments with scenarios involving both cloth and rope. The parameters are set as $k_g = 4$, $k_D = 10$, $k_r = 20$ for the new model, $l_c = 0.023$, and $s_s = 0.4$ for the new controller. The parameters we used for the benchmark method are its default best value found in [16]. The stretching detection ratio is set as $\gamma_s = 1.667$ for the cloth and $\gamma_s = 1.1$ for the rope. The maximum gripper motion is set as $\Delta_{q,\max} = 0.03$, with time step $\Delta t = 0.01$s. For the experiment with a task goal defined, a precalculated Dijkstra field for the task will generate the $\Delta \mathcal{P}_d$ for each point at each state to move the object toward the goal given each point's current position in the space. All experiments were run on a i7-8700K 3.7 GHz CPU with 32 GB of RAM. A video showing the experiments is included with this paper.

A. Model Accuracy Results

We evaluated model accuracy by pulling the rope in a straight line along the direction of the rope, then turning the gripper and pulling back towards the rope as shown in Fig. 1. As shown in Fig. 4, our new model is a better approximation of the true motion when the gripper is pulling the rope. When the gripper is turning, both the baseline and the new model produce comparable error, but when the gripper starts pulling again (this time in the opposite direction), the new model is a significantly better approximation.

We also evaluated model accuracy by pulling the cloth in a similar fashion; pulling the cloth one way, turning the grippers, and then pulling in the opposite direction. As shown in Fig. 5, our new model is a better approximation of the true motion when the grippers are pulling the cloth. As in the rope test, when rotating the grippers both models produce comparable error. While the cloth is folded on itself both models produce noisy results, but when the cloth lies flat again, the new model achieves lower error.

B. Constraint Enforcement

Since the benchmark controller can already handle the collision constraint very well, and the new controller addresses the collision constraint in the similar way as the benchmark, there is not a significant difference in how the collision
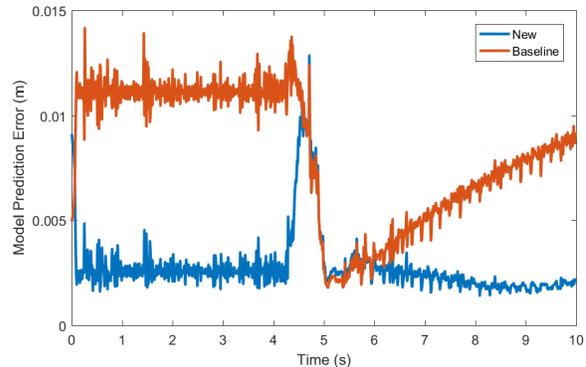

Fig. 4: RMS model prediction error. The gripper pulls the rope for the first 4.5 seconds, then turns for half a second, then moves in the opposite direction at the 5 second mark.
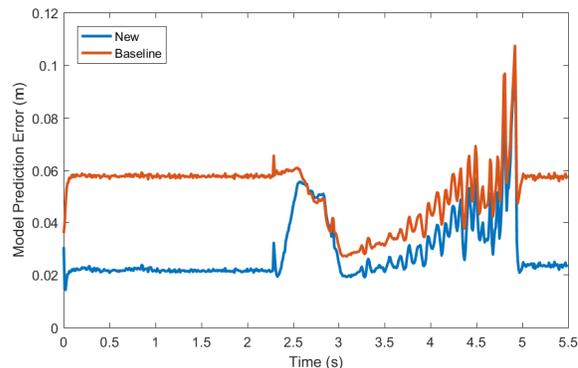

Fig. 5: RMS model prediction error. The grippers pull the cloth for the first 2.3 seconds, then turn for 0.63 seconds, then move in the opposite direction at the 2.93 second mark. At the 5 second mark the cloth is no longer folded.

constraint is enforced. However, the stretching constraint shows a very clear improvement.

The metrics of stretching avoidance is the stretching ratio γ defined in Section IV-B.2. A controller with good stretching avoidance should prevent γ from increasing beyong a certain threshold.

The two experiments we used for the stretching avoidance test are the rope-wrapping-cylinder and the cloth-passing-single-pole, shown in Fig.6a and Fig.6d. We ran each controller separately for a fixed amount of time for each task and show γ vs. time for both controllers in Fig. 7 . In both these two setups, the desired object motion $\dot{\mathcal{P}}_d$ generated by the Dijkstra field will tear the object unless overstretching is prevented.

Fig. 7 shows the new controller is able to prevent further stretching happening when the object is taut for both the rope and the cloth. In the rope test, the new controller can prevent overstretching with $s_s = 0.4$, as defined in Eq.13. We can see the γ of the benchmark methods keeps growing beyond this threshold, while the γ of the new method stays close to the threshold. In the cloth test, the benchmark method's γ (in purple) increases above the threshold $\gamma_s = 1.667$ for cloth, and a sudden drop in γ happens after running the test for 2 seconds. This drop is the "tearing" point in the simulator.

(a) Rope wrapping cylinder   (b) Rope matching zig-path

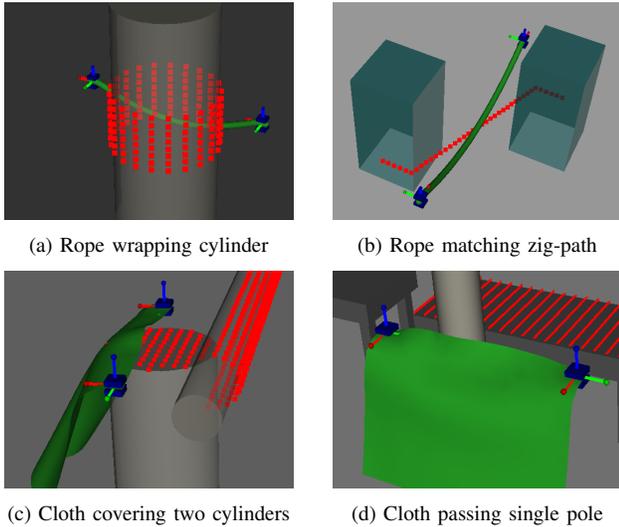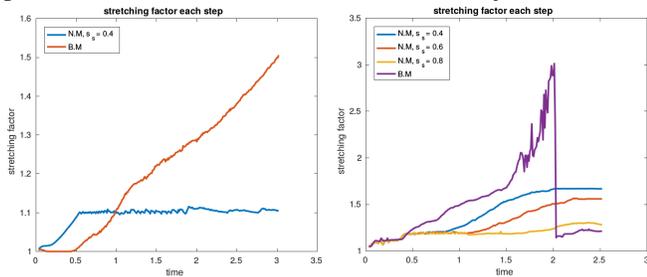(c) Cloth covering two cylinders   (d) Cloth passing single pole

Fig. 6: Initial state of the four experiments, where the red points act as attractors for the deformable object.

(a) Rope wrapping cylinder   (b) Cloth passing single pole

Fig. 7: (a) The red line shows the γ of the benchmark and the blue line shows the γ of the new controller with $s_s = 0.4$ throughout the simulation. (b) The purple line shows the γ of the benchmark, and the blue, red, and yellow lines each show the γ of the new controller with $s_s = 0.4$, $s_s = 0.6$, and $s_s = 0.8$, respectively.

Though we still see overstretching happened using the new method for some settings of s_s, in all cases the γ converged before tearing happened (instead of growing without bound).

C. Controller Task Performance

Besides the quantitative analysis of the model accuracy and stretching avoidance, we ran another two experiments, rope-matching-zig-path and cloth-covering-two-cylinder, one each with the rope or the cloth, as shown in Fig.6b and Fig.6c to see how the new method performed for some coverage tasks. Both the benchmark and the new controllers are able to perform these tasks with comparable performance; reaching approximately the same configurations when forward progress stops due to a local minimum (Fig. 8), and completing the task (Fig. 9). This result suggests that we have not lost functionality with respect to the benchmark despite changing the model and control method used.

D. Physical Robot Experiments

To evaluate our new model and controller on a physical system, we set up two experiments with cloth-like objects manipulated by two 7DoF KUKA iiwa arms (Fig. 10). To
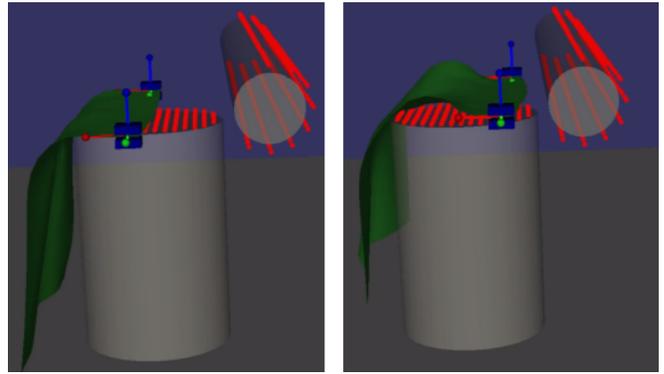

Fig. 8: Cloth-covering-two-cylinder task start and end configurations. Both controllers are unable to make progress due to a local minima.
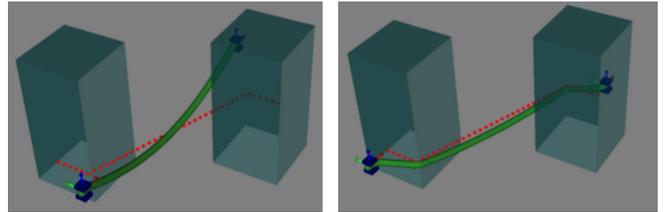

Fig. 9: Rope-matching-zig-path start and end configurations. Both controllers are able to succeed at the task, bringing the rope into alignment with the desired path.

sense the position of the cloth, we use the AprilTags [21] and IAI Kinect2 [22] libraries. The parameters are set as $k_g = 4$, $k_D = 10$, $k_r = 10$ for the new model, $l_c = 0.08$, and $s_s = 0.6$ for the new controller. The first test, which evaluates model accuracy, uses a motion profile similar to the simulation accuracy tests (Fig. 11). Similar to the simulation results, the new model improves performance when dragging the cloth (first and last sections of Fig. 11), and is comparable during rotational motion and when the cloth is resting on edge perpendicular to the table (see video). To test the controller, we set up a task similar to the cloth-passing-single-pole example using a paper towel. For this task, the baseline controller tears the paper towel while the new controller avoids excessive overstretch, instead wrapping around the pole to reach a local minimum.

E. Computation Time

To verify the practicality of our method, we gathered data comparing its computation time to the benchmark's and to using the Bullet simulator. Table I shows the average computation time of a call to the controller for the new method vs. the benchmark. As expected, the benchmark, which uses a linear model, is faster than the new method. However, the computation times for the new method are still reasonable to use in a control loop.

Table II shows a comparison between the average time needed to evaluate the new model and the time needed to simulate a gripper motion with the Bullet simulator. Note that the amount of time required for the simulator to converge to a stable estimate depends on many conditions, including what object is being simulated. Through experimentation
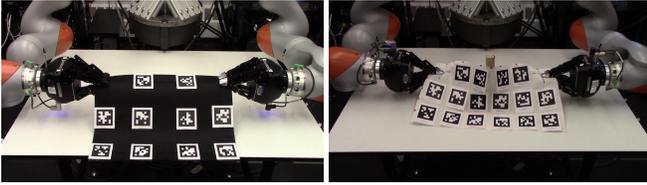
Fig. 10: Initial setup for the physical robot experiments. Left: model accuracy test. Right: stretching avoidance test.
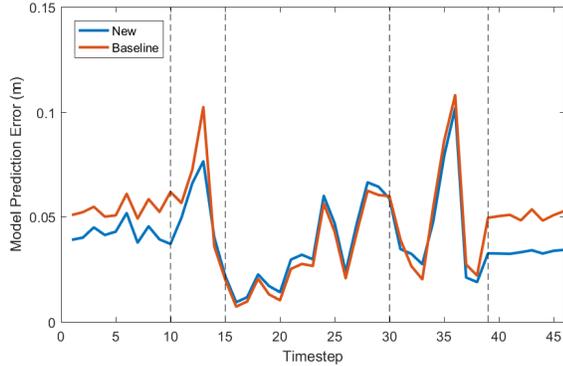

Fig. 11: RMS model prediction error. The grippers pull the cloth toward the robot for the first 10 timesteps, upward for 5 timesteps, rotate for 15 timesteps, diagonally down and away for 9 timesteps, then directly away from the robot.

we determined that 4 simulation steps were adequate for rope and 10 for cloth. Comparing the time needed to do this simulation to the time needed to evaluate our model, we see that the new model is indeed faster by at least an order of magnitude, in some cases by two orders of magnitude, confirming that, despite being slower than the benchmark, our method still outperforms the simulator in terms of computation time.

VI. CONCLUSION

In this paper we extended a geometric model for deformable object manipulation to include directional rigidity and obstacle penetration constraints. Considering these terms in the model allowed us to improve the model's prediction while maintaining low computation cost. We also addressed the excessive stretching issue by approximating the stretching constraints with conical constraints around stretching avoidance vectors. Our controller is able to complete the same tasks as the previous benchmark method but also correctly enforces stretching constraints when the object is caught on obstacles. In future work, we would like to investigate the sensitivity of the model with respect to its parameters and develop methods to automatically tune these parameters based on observed motion.

REFERENCES

[1] N. Essahbi, B. C. Bouzgarrou, and G. Gogu, "Soft Material Modeling for Robotic Manipulation," in *Applied Mechanics and Materials*, vol. 162, April 2012, pp. 184–193.
[2] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations," *SIGGRAPH*, pp. 49–54, 2002.
[3] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *IROS*, 2013, pp. 4525–4532.

	rope-wrapping -cylinder	rope-matching -zig-path	cloth-passing single-pole	cloth-wrapping -two-cylinders
BM	0.0055	0.0054	0.0153	0.0037
NM	0.0342	0.0834	0.2363	0.1008

TABLE I: Mean computation time (s) to compute the gripper motion for a given state. BM: benchmark method; NM: new method.

	rope-wrapping -cylinder	rope-matching -zig-path	cloth-passing single-pole	cloth-wrapping -two-cylinders
BT	0.686	0.571	19.29	3.680
NM	0.029	0.014	1.172	0.339
# evals	50.72	143.5	83.81	63.32

TABLE II: Top two rows: Mean computation time (ms) per model prediction for a given gripper motion. BT: Bullet simulator; NM: new model. Bottom row: Mean number of times the model was evaluated when executing the new method.

[4] P. Jiménez, "Survey on model-based manipulation planning of deformable objects," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 154–163, Apr. 2012.
[5] B. Maris, D. Botturi, and P. Fiorini, "Trajectory planning with task constraints in densely filled environments," in *IROS*, 2010.
[6] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Siciliano, "Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot," *Rob. Auton. Syst.*, vol. 88, no. C, pp. 187–201, Feb. 2017.
[7] A. Borum, D. Matthews, and T. Bretl, "State estimation and tracking of deforming planar elastic rods," in *ICRA*, May 2014, pp. 4127–4132.
[8] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *IJRR*, vol. 33, no. 1, pp. 48–68, 2014.
[9] J. Lang, D. Pai, and R. Woodham, "Acquisition of elastic models for interactive simulation," *IJRR*, vol. 21, no. 8, pp. 713–733, 2002.
[10] A. M. Cretu, P. Payeur, and E. M. Petriu, "Neural network mapping and clustering of elastic behavior from tactile and range imaging for virtualized reality applications," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 9, pp. 1918–1928, Sept 2008.
[11] S. Hirai and T. Wada, "Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model," *Robotica*, vol. 18, no. 1, pp. 3–11, Jan. 2000.
[12] T. Wada, S. Hirai, S. Kawarnura, and N. Karniji, "Robust manipulation of deformable objects by a simple PID feedback," in *ICRA*, 2001, pp. 85–90.
[13] D. Navarro-Alarcon, Y.-h. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *IJRR*, vol. 33, no. 11, pp. 1462–1480, jun 2014.
[14] D. Navarro-Alarcon and Y. H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours," *T-RO*, vol. 34, no. 1, pp. 272–279, Feb 2018.
[15] Z. Hu, P. Sun, and J. Pan, "Three-dimensional deformable object manipulation using fast online gaussian process regression," *RA-L*, vol. 3, no. 2, pp. 979–986, April 2018.
[16] D. McConachie and D. Berenson, "Estimating model utility for deformable object manipulation using multiarmed bandit methods," *T-ASE*, vol. 15, no. 3, pp. 967–979, July 2018.
[17] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *T-RO*, vol. 3, no. 1, pp. 43–53, 1987.
[18] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994, vol. 29.
[19] Gurobi, "Gurobi optimization library," *Proprietary: gurobi.com*, 2017.
[20] E. Coumans, "Bullet physics library," *Open source: bulletphysics.org*, 2010.
[21] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *ICRA*, May 2011, pp. 3400–3407.
[22] T. Wiedemeyer, "IAI Kinect2," https://github.com/code-iai/iai_kinect2, Institute for Artificial Intelligence, University Bremen, 2014 – 2015.