Jacobian-based Control of Soft Robots for Manipulation Using Implicit Surface Models

by

Geronimo Mirano

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Author .....
Department of Electrical Engineering and Computer Science
May 26, 2017

Certified by .....
Russ Tedrake
Professor
Thesis Supervisor

Accepted by .....
Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Jacobian-based Control of Soft Robots for Manipulation Using Implicit Surface Models

by

Geronimo Mirano

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 2017, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Soft robot hands offer numerous advantages over rigid ones for manipulation, including robustness and safety. Yet, compared to rigid robots, soft robots are characterized by continuous mechanics, and finite-element approximations with many degrees of freedom present a significant obstacle for modern control approaches. The central question my thesis explores is whether we can capture the benefits of soft robot hands with relatively simple dynamical models. Specifically, we demonstrate a very simple model of a 2D soft manipulator that uses pulleys and cables to model deformable surfaces. This model captures much of the qualitative behavior of soft membranes, while also proving amenable to modern control techniques. We validate this model physically using a hardware set-up. We then demonstrate a simple quasi-static Jacobian controller which solves a second-order cone program to achieve the task of in-hand object repositioning.

Thesis Supervisor: Russ Tedrake
Title: Professor

Acknowledgments

I would like to thank my advisor and fellow members of the Robot Locomotion Group for all their help and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of soft robotics offers the promise of biologically-inspired manipulators whose performance could meet or exceed the manipulators found in nature [7]. Due to the compliant interface which they present to the manipuland, soft manipulators can achieve robust, high-friction grasps with a much higher chance of success than rigid-body manipulators. There have been numerous successes involving manipulator designs which exploit this robustness [2][8]. Indeed, the power of soft hands is such that basic force-closure grasps can be achieved with very high success rate even when driven by open-loop control.

However, there are some applications which require more than just force-closure grasps. One of the most pure such tasks is that of in-hand manipulation, the task of repositioning an already-grasped object, which might be necessary in order to place an object in a configuration that is beyond the reach of the arm itself. Another class of fine manipulation tasks is manipulating in the face of external constraints, such as when turning a doorknob or screwing a nut onto a bolt, where the axis of rotation is a constraint imposed by the environment. Although it might be possible to accomplish such tasks by means of a force-closure grasp and the kinematic repositioning of the manipulator, this will likely require large kinematic repositioning on the part of the rest of the robot's body. In contrast, when a human is screwing in a nut, rather than rotate their entire arm, the person will use fine control of their fingers at the contact interface in order to achieve the motion at a greatly reduced energy and greatly

improved speed. A related motivating example comes from the field of upper-limb prosthetic devices. Presently in such systems, the pinch grasps executed by the fingers have one controllable degree of freedom. As a result, the angle that the fingers make relative to the wrist is critical, as it determines which local reorientation maneuvers the expert operator can perform without having to grossly reposition their entire arm [1]. In this case, the most effective solution would be to have a hand that is capable of fine in-hand manipulation techniques to circumvent this problem at all wrist angles, and though compliant and actuated wrists have been introduced, this is still a field which could see much benefit from improved robotic manipulation control.

For the above reasons, people have been exploring ways to apply the tools of control theory to the task of manipulation via soft actuators. This involves tackling the two difficult problems of controlling through contact and controlling potentially-continuous soft robots.

One of the most daunting properties of soft materials from a modeling perspective is their continuously-deformable nature. For example, imagine we are impressing rigid, complicated objects into the surface of a silicone block. For every such object we could produce, the rubber contains the state degrees of freedom necessary to conform to that shape exactly. This seems to stand in stark contrast to rigid body models, which require only a handful of state variables to fully-characterize their state-space. In addressing this high dimensionality, there are several approaches that have been taken. Finite-element approximations to the continuously-deformable surfaces have offered a means to approximate these surfaces. By modeling a material as a grid of discrete interacting pieces and solving the constitutive equations throughout the soft material, one can capture the intricacies of the deformations at a specific granularity [4]. Using these complicated models in tandem with higher-rate linearizations of those models, Jacobian-control type methods have been demonstrated [3] [6]. Although finite-element methods are fast enough for Jacobian control, the high dimensionality of these models precludes the ability to perform higher-level verification or optimization-based feedback, techniques which scale poorly in state dimensions. A different approach, known as constant curvature continuum robotics, preserves the

continuously-deformable nature of the system but captures these continuous interfaces via a reduced set of state variables thus allowing for feasible computations while preserving the flexibility offered by these soft robots [10]. In this case, the dimensionality remains low, but the dynamics become more complicated, posing a challenge. More importantly, these systems are not applied to modeling the contact surface between the soft body and other bodies, a major deficiency when controlling for the task of manipulation.

In contrast to both of the above approaches, we propose a model that captures the continuous kinematics and dynamics of soft-membrane interaction by encoding it as a simple kinematic constraint on rigid-body systems. Specifically, we demonstrate a model of a 2D soft manipulator that uses cables and springs to model deformable surfaces. This model is low-dimensional, like the constant curve continuum models, but it captures the full continuous dynamics of soft contact. We demonstrate via simulation and experimental validation that it captures much of the qualitative behavior of several soft membrane systems. By capturing the dynamics of soft interaction within the framework of constrained rigid-body dynamics, we hope to present a path toward applying sophisticated control-theoretical tools to soft systems.

Chapter 2

2D Cable Model for Soft Dynamics

2.1 Modeling Thin Elastic Membranes

Consider a thin, one-dimensional membrane embedded in two dimensions. We can model this as a continuum of elements, each constrained relative to its neighbors to capture tension stiffness and bending stiffness (Figure 2-1).

Consider the sorts of soft, thin membranes encountered in the real world. This might encompass fabrics or elastic materials like rubber bands or latex. In such systems, tension stiffness dominates their dynamics, whereas bending stiffness is often negligible. If we allow bending stiffness to go to zero, and also assume our membrane to be mass-less and friction-less, then we will have a membrane which uses only tensile forces to affect the world. The major contribution of this paper is to evaluate a broadly-applicable special case of thin-membrane models implemented entirely in terms of constrained rigid-body dynamics.

We implement a model of soft dynamical systems using cables, pulleys, and springs. Cables and pulleys are implemented as kinematic constraints which constrain the evolution of the rigid body system, and these cables terminate in springs to incorporate tension stiffness and damping.

It might seem strange to model a soft membrane as stateless. However, there are many materials which behave in a way that is qualitatively analogous to this. For example, a rubber band will move so quickly in response to changes in its wrapping

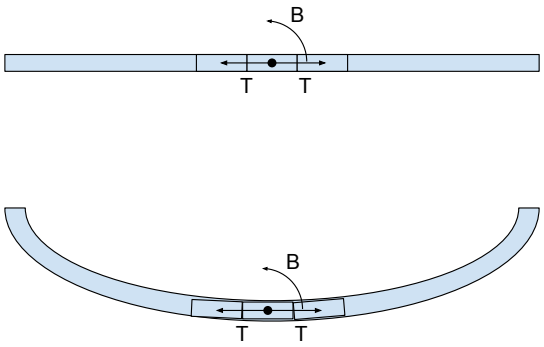


Figure 2-1: Examples of coarse finite-element models of soft membranes. Neighboring elements exert both tension and bending forces on one another. The lower figure would represent a higher-energy configuration, both in tension and in bending.

points that it can be difficult even to see the motion. The settling time of the rubber band is much shorter than that of the surrounding system, and so a quasi-static description of its motion will be a very close fit.

First, we give a general overview of constrained rigid body dynamics. Then, we show how our model of a friction-less, mass-less, implicit surface membrane is implemented as a constrained rigid body system.

2.2 Bilateral Constraints in Rigid-Body Dynamics

First, we will give a brief overview of rigid body dynamics. The major reference for this overview is the online textbook Underactuated Robotics by Professor Russ Tedrake [9].

Consider a dynamical system. It will consist of a position vector $\vec{q} \in \mathbb{R}^N$, which gives the poses of objects. Moreover, physical systems include a velocity vector $\vec{v} \in \mathbb{R}^{N_v}$ as part of their state. For many systems, we will simply have $\vec{v} := \dot{\vec{q}}$, but having the ability to choose a more general $\vec{v}$ can be very useful for things such as

3D rotations. We thus denote the full state of a physical system as $\vec{x} = \begin{bmatrix} \vec{q} \\ \vec{v} \end{bmatrix}$.

From Lagrangian dynamics, one can derive that any physical second-order dynamical system is subject to the "manipulator equations":

$$\mathbf{H}(\vec{q})\dot{\vec{v}} + \vec{c}(\vec{q}, \vec{v}) = \mathbf{B}\vec{u}$$

One convenient way to represent a kinematic constraint is as a scalar function which is 0 when the constraint is satisfied and nonzero otherwise, and whose signed magnitude is proportional to the error in the constraint:

$$\phi(\vec{q}) = 0$$

Constraints remain satisfied by exerting forces on the system. Setting the constraint and its derivatives equal to zero, one can solve for this force:

$$\phi(\vec{q}) = \dot{\phi}(\vec{q}) = \ddot{\phi}(\vec{q}) = 0 \tag{2.1}$$

$$\mathbf{J}(\vec{q}) := \frac{\partial \phi(\vec{q})}{\partial \vec{q}}$$

$$\mathbf{H}(\vec{q})\dot{\vec{v}} + \vec{c}(\vec{q}, \vec{v}) = \mathbf{B}\vec{u} + \mathbf{J}(\vec{q})^\top \lambda \tag{2.2}$$

$$\lambda = -(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^\top)^\dagger(\mathbf{J}\mathbf{H}^{-1}(\mathbf{B}\vec{u} - \vec{c}) + \dot{\mathbf{J}}\vec{v}) \tag{2.3}$$

The generalized force term $\mathbf{J}(\vec{q})^\top \lambda$ captures the intuitive fact that a constraint should only apply forces in the direction in which it is constraining a configuration. This shows how bilateral position constraints can be implemented in rigid body systems.

2.3 Cable Equations of Motion

We will implement cables which wrap through a number of radius-zero pulleys whose locations are determined by the configuration $\vec{q}$ of our system. We restrict our analysis to radius-zero point pulleys in order to keep the mathematics simple, but for a

treatment involving circular pulleys see Katzschmann, et al. [5].

The cable constraint says that sum of the lengths of cable between the wrap points in our system should remain constant. Consider a cable which wraps through M radius-zero pulleys, each of which is at a location $\vec{p}_i(\vec{q})$. Let L be the nominal length of our cable. We can then define our constraint function ϕ as:

$$\phi(\vec{q}) = -L + \sum_{i=1}^{M-1} ||\vec{p}_{i+1}(\vec{q}) - \vec{p}_i(\vec{q})||_2 = 0$$

With this in hand, equation 2.3 can be solved to simulate the system dynamics.

As an implementation detail, it will be useful to compute the quantities $\mathbf{J}(\vec{q})$, $\mathbf{J}_v(\vec{q})$ and $\dot{\mathbf{J}}_v(\vec{q})\vec{v}$. The Jacobian $\mathbf{J}(\vec{q})$ of this constraint can be computed as follows:

$$\mathbf{J}(\vec{q}) = \frac{\partial \phi(\vec{q})}{\partial \vec{q}} = \sum_{i}^{M-1} \frac{\vec{p}_{i+1}(\vec{q}) - \vec{p}_i(\vec{q})}{||\vec{p}_{i+1}(\vec{q}) - \vec{p}_i(\vec{q})||_2} (\mathbf{J}_{\text{kin},i+1} - \mathbf{J}_{\text{kin},i})$$

where $\mathbf{J}_{\text{kin},i} = \frac{\partial \vec{p}_i(\vec{q})}{\partial \vec{q}}$, the gradient of the point $\vec{p}_i$ with respect to $\vec{q}$. It is convenient to express our Jacobian in terms of this forward-kinematics Jacobian, as this allows us to offload this calculation to a forward-kinematics computation pipeline.

As we consider higher-order terms, we will also need to define a new term, the "velocity Jacobian" $\mathbf{J}_v(\vec{q})$. This type of Jacobian will be necessary due to the fact that $\vec{q}$ and $\vec{v}$ might potentially be expressed in different coordinate frames. Whereas the typical Jacobian is capable of mapping $\dot{\vec{q}}$ into some output $\dot{\vec{\phi}}(\vec{q})$ by means of multiplication and the chain rule, the velocity Jacobian will produce the same result when applied to $\vec{v}$ (thus, $\mathbf{J}\dot{\vec{q}} = \mathbf{J}_v\vec{v}(\dot{\vec{q}})$, $\forall \dot{\vec{q}}$). Due to the nice properties of the chain rule, our constraint's velocity Jacobian $\mathbf{J}_v(\vec{q})$ can again be written in terms of the forward kinematics Jacobian:

$$\mathbf{J}_v(\vec{q}) = \sum_{i}^{M-1} \frac{\vec{p}_{i+1}(\vec{q}) - \vec{p}_i(\vec{q})}{||\vec{p}_{i+1}(\vec{q}) - \vec{p}_i(\vec{q})||_2} (\mathbf{J}_{v,\text{kin},i+1} - \mathbf{J}_{v,\text{kin},i})$$

where $\mathbf{J}_{v,\text{kin},i}$ is the velocity Jacobian of the point $\vec{p}_i(q)$. Finally, a little calculus

suffices to show that $\dot{\mathbf{J}}_v(\vec{q})\vec{v}$ can be computed as:

$$\dot{\mathbf{J}}_v(\vec{q})\vec{v} = \frac{(\vec{p}_{i+1} - \vec{p}_i)^\top}{|\vec{p}_{i+1} - \vec{p}_i|}(\dot{\mathbf{J}}_{v,\text{kin},i+1}\vec{v} - \dot{\mathbf{J}}_{v,\text{kin},i}\vec{v}) + \vec{w}^\top \left[\frac{1}{|\vec{p}_{i+1} - \vec{p}_i|} - \frac{(\vec{p}_{i+1} - \vec{p}_i)(\vec{p}_{i+1} - \vec{p}_i)^\top}{|\vec{p}_{i+1} - \vec{p}_i|^3}\right]\vec{w}$$

with

$$\vec{w} := \mathbf{J}_{v,\text{kin},i+1}\vec{v} - \mathbf{J}_{v,\text{kin},i}\vec{v}$$

Conveniently, we have computed this in terms of $\dot{\mathbf{J}}_{v,\text{kin},i}\vec{v}$, so we only require that $\dot{\mathbf{J}}_{v,\text{kin},i}\vec{v}$ be computed by our forward kinematics pipeline in order to compute $\dot{\mathbf{J}}_v\vec{v}$ for our constraint.

2.4 Dynamic Collisions Between Pulleys and Cables

We now have an implementation of the cable constraint for any fixed set of points $\{\vec{p}_i\}$. However, we will find that it will be convenient to allow for this set of points to change over time. For example, consider the case of a hexagonal pulley (Figure 2-2). In this case, the locations of the terminal wrap points determines which vertices of the hexagon will be engaged.

2.4.1 General Proof of Non-Impulsive Pulley Collisions

Unlike colliding rigid-body systems, we will show that the velocity v of this system will not jump discontinuously as points are added to or removed from the cable constraint.

Theorem 1. For any configuration of points $\{\vec{p}_i(\vec{q})\}$, consider the addition of a point $\vec{p}(\vec{q})$ at time $t = 0$ between points $\vec{p}_j(\vec{q})$, $\vec{p}_{j+1}(\vec{q})$ which satisfies

$$\vec{p}(\vec{q}_0) = \alpha \cdot \vec{p}_j(\vec{q}_0) + (1 - \alpha) \cdot \vec{p}_{j+1}(\vec{q}_0) \quad ; 0 < \alpha < 1.$$

This addition will be non-impulsive provided that $\mathbf{J}(\vec{q}_0)$ has full row-rank and $\mathbf{H}(\vec{q}_0)$ is positive definite.

Proof. We will demonstrate that, so long as $\mathbf{J}(\vec{q}_0)$ has full row-rank, then there is a consistent non-impulsive solution to the manipulator equations and the system will not undergo an impulse. In order to reason about this formally, let terms subscripted with a minus sign ("$-$") represent represent the values of these variables before the point is added, and let the terms subscripted with a plus sign ("$+$") represent the values after the point is added. In this case, the terms to consider are:

$$\dot{\vec{q}}, \ddot{\vec{q}}, \phi, \dot{\phi}, \ddot{\phi}, \mathbf{J}, \dot{\mathbf{J}}$$

It is straightforward to observe that the constraint value $\phi(\vec{q}_0)$ is unchanged under the addition of a wrap point to an already-extant segment. Furthermore, the Jacobian $\mathbf{J}(\vec{q}_0)$ with respect to $\vec{q}$ must also be unchanged, as the point which was added lies precisely on the line segment between its predecessor and successor, meaning that it occupies a local minimum for the length function that the constraint evaluates and thus satisfies $\frac{\partial \phi}{\partial \vec{p}} = 0$. Formally, we can write

$$\phi(\vec{q}_0) := \phi_-(\vec{q}_0) = \phi_+(\vec{q}_0)$$
$$\mathbf{J}(\vec{q}_0) := \mathbf{J}_-(\vec{q}_0) = \mathbf{J}_+(\vec{q}_0).$$

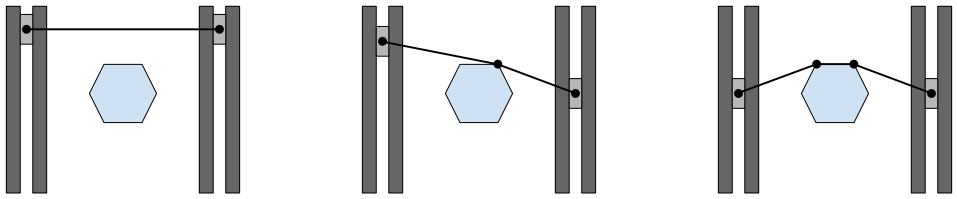


Figure 2-2: This figure shows an example where we may wish to allow for dynamic changing of the wrap point set. Comparing the three figures, one can see that the pose $\vec{q}$ of the objects not only determines the continuous locations $\vec{p}_i$ of the wrap points, but also determines which wrap points are used and in what order.

Before adding the point $\vec{p}$ to our cable, the system satisfied the following dynamics equations:

$$\mathbf{H}\ddot{\vec{q}}_- + c(\vec{q}, \dot{\vec{q}}_-) = \mathbf{J}^\top \lambda_- \tag{2.4}$$

$$\phi = 0 \tag{2.5}$$

$$\dot{\phi}_- = \mathbf{J}\dot{\vec{q}}_- = 0 \tag{2.6}$$

$$\ddot{\phi}_- = \mathbf{J}\ddot{\vec{q}}_- + \dot{\mathbf{J}}_-\dot{\vec{q}}_- = 0 \tag{2.7}$$

After adding the point $\vec{p}$ to our cable, the system must satisfy the following dynamics equations:

$$\mathbf{H}\ddot{\vec{q}}_+ + c(\vec{q}, \dot{\vec{q}}_+) = \mathbf{J}^\top \lambda_+ \tag{2.8}$$

$$\phi = 0 \tag{2.9}$$

$$\dot{\phi}_+ = \mathbf{J}\dot{\vec{q}}_+ = 0 \tag{2.10}$$

$$\ddot{\phi}_+ = \mathbf{J}\ddot{\vec{q}}_+ + \dot{\mathbf{J}}_+\dot{\vec{q}}_+ = 0 \tag{2.11}$$

Note that equation 2.9 is satisfied already. If there is not to be an impulse, then we will have $\dot{\vec{q}}_+ = \dot{\vec{q}}_-$. In this case, equation 2.10 will automatically be satisfied, following from equation 2.6. The remaining equations are:

$$\mathbf{H}\ddot{\vec{q}}_+ + c(\vec{q}, \dot{\vec{q}}) = \mathbf{J}^\top \lambda_+ \tag{2.12}$$

$$\mathbf{J}\ddot{\vec{q}}_+ + \dot{\mathbf{J}}_+\dot{\vec{q}} = 0 \tag{2.13}$$

The scalar term $\dot{\mathbf{J}}_+\dot{\vec{q}}$ may differ from its predecessor $\dot{\mathbf{J}}_-\dot{\vec{q}}$ under the addition of the new point, because $\frac{d}{dt}\left[\frac{\partial\phi}{\partial\vec{p}}\right]$ is not necessarily zero. We can solve this new system of equations for λ_+ [1]:

$$\lambda_+ = -(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^\top)^\dagger(\dot{\mathbf{J}}_+\dot{\vec{q}} - \mathbf{J}\mathbf{H}^{-1}c(\vec{q}, \dot{\vec{q}}))$$

[1] This is very close to the derivation for generic bilateral constraints found in *Underactuated Robotics: Appendix A* [9].

As long as $\mathbf{J}$ has full row-rank and $\mathbf{H}$ is positive-definite, the above shall be solvable and there shall be a consistent assignment to $\lambda_+, \ddot{q}_+$ which preserves the dynamics without impulse. $\square$

We have demonstrated that these constraints will be non-impulsive under the addition of a point along a pre-existing length of the cable. In the next section, we will see an example of this non-impulsivity.

2.4.2 Example: Cable, Point, and Ballast

Let us consider an example of Theorem 1. Consider the case of a point $\vec{p}(\vec{q})$ which is about to make contact with a cable system. This system consists of a cable which is taut between two fixed pulleys and which terminates in a ballast mass initially at rest (Figure 2-3). In this system, two of the cable wrap points are affixed to the environment, and the free-moving pieces are the point and the ballast mass:

$$\vec{q} = \begin{bmatrix} p_x \\ p_y \\ b_x \end{bmatrix}, \vec{v} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{b}_x \end{bmatrix}$$

The initial velocity of this point can be written as $\vec{v}_0 := \frac{\partial \vec{p}}{\partial \vec{q}} \dot{\vec{q}}$. We shall leave v_0 a free parameter to show that this example is non-impulsive under any velocity. In this example, we will start with the object positioned at the center of the cable and the ballast a distance l to the right (we'll define this pose to be $\vec{q}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$). At time $t = 0_-$, the point $\vec{p}$ was moving with velocity $\dot{\vec{p}} = \vec{v}_0$ and the ballast was at rest with $\dot{b}_x = 0$ (as it had to be for the cable to have $\dot{\phi}(\vec{q}) = 0$). We will show that these same velocities are consistent at time 0_+ after adding the object as part of the cable.

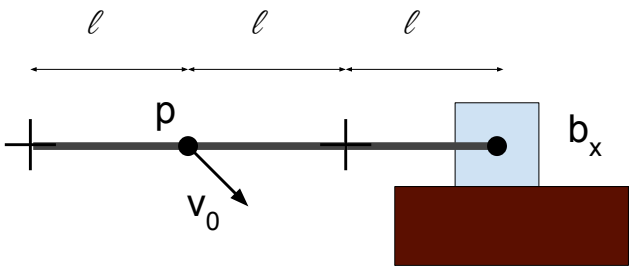


Figure 2-3: The set-up for the non-impulsivity example. There are 2 fixed-in-place wrap points represented by black cross-hairs. Connected to the right-hand cross-hair is the ballast mass, which is free to slide frictionlessly along the table in the x-direction. Between the two anchors is a point mass which is being instantaneously added to the cable system at the current timestep.

The manipulator equations for this system are very simple:

$$\begin{aligned}
\mathbf{H}\dot{\vec{v}} &= \mathbf{J}^\top \lambda \\
\phi(\vec{q}) &= 0 \\
\dot{\phi}(\vec{q}) &= \mathbf{J}\vec{v} = 0 \\
\ddot{\phi}(\vec{q}) &= \mathbf{J}\dot{\vec{v}} + \dot{\mathbf{J}}\vec{v} = 0
\end{aligned}$$

We can find $\mathbf{J}$ by considering the constraint equations

$$\begin{aligned}
\phi(\vec{q}) &= -3l + \sqrt{p_y^2 + (p_x - l)^2} + \sqrt{p_y^2 + (p_x + l)^2} + b_x \\
\mathbf{J} = \frac{\partial \phi}{\partial \vec{q}} &= \begin{bmatrix} \left(\frac{p_x - l}{\sqrt{p_y^2 + (p_x - l)^2}} + \frac{p_x + l}{\sqrt{p_y^2 + (p_x + l)^2}}\right) & \left(\frac{p_y}{\sqrt{p_y^2 + (p_x - l)^2}} + \frac{p_y}{\sqrt{p_y^2 + (p_x + l)^2}}\right) & 1 \end{bmatrix} \\
\dot{\mathbf{J}} = \frac{d}{dt}\frac{\partial \phi}{\partial \vec{q}} &= \begin{bmatrix} \left(\frac{p_y^2 \dot{p}_x}{(p_y^2 + (p_x - l)^2)^{\frac{3}{2}}} + \frac{p_y^2 \dot{p}_y}{(p_y^2 + (p_x + l)^2)^{\frac{3}{2}}}\right) & \left(\frac{(p_x - l)^2 \dot{p}_y}{(p_y^2 + (p_x - l)^2)^{\frac{3}{2}}} + \frac{(p_x + l)^2 \dot{p}_y}{(p_y^2 + (p_x + l)^2)^{\frac{3}{2}}}\right) & 0 \end{bmatrix}
\end{aligned}$$

The manipulator equations are therefore:

$$\begin{bmatrix} m_p & 0 & 0 \\ 0 & m_p & 0 \\ 0 & 0 & m_b \end{bmatrix} \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{b}_x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \lambda \tag{2.14}$$

$$\dot{\phi}(\vec{q}) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{0,x} \\ v_{0,y} \\ 0 \end{bmatrix} = 0 \tag{2.15}$$

$$\ddot{\phi}(\vec{q}) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^\top \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{b}_x \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2v_{0,y}}{l} \\ 0 \end{bmatrix}^\top \begin{bmatrix} v_{0,x} \\ v_{0,y} \\ 0 \end{bmatrix} = 0 \tag{2.16}$$

And we can solve them straightforwardly to yield consistent accelerations:

$$\ddot{b}_x = -\frac{2v_{0,y}^2}{l}$$

$$(m_x b_x = \lambda)$$

$$\ddot{p}_x = \ddot{p}_y = 0$$

Thus, although we observe a jump discontinuity in $\ddot{b}_x$ (from 0 to $-\frac{2v_{0,y}^2}{l}$), the velocities of the system did not undergo any impulsive changes, irrespective of the velocity $\vec{v}_0$ of the newly-contacting body.

2.4.3 Choosing Cable Wrap Points

Theorem 1 assures us that our dynamics are non-impulsive across pulley collisions, the types of cable wrap point changes which we wish to support.

Next, we must address the issue of choosing wrap points given the state $\vec{q}$ of the system. Specifically, we will seek to handle the case of regular n-gonal pulleys, which are versatile enough to be useful and which have convex shapes which make it easier to reason about their windings.

Given the winding direction for a pulley, we can ascertain which wrap points the

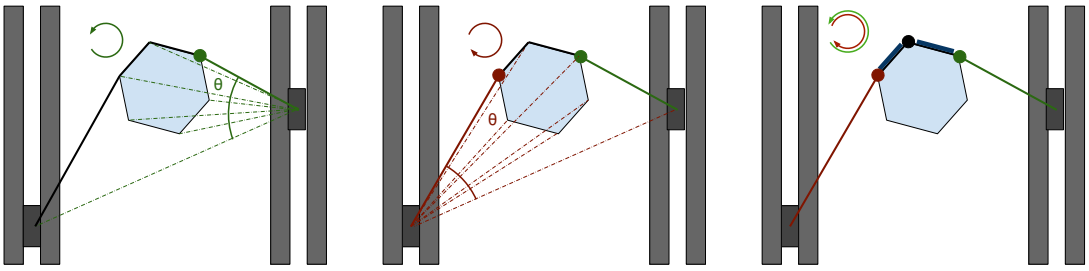


Figure 2-4: Dynamic wrap point selection for n-gonal pulleys is determined by which point makes the greatest angle to the line between the two end points.

cable will pass through. Consider the example of the hexagonal pulley. The critical decision to make is which two vertices (including the terminal points themselves) are directly connected to the terminal points $\vec{p}_A$, $\vec{p}_B$.

Because this is a wrap point selection rule of the form shown in Theorem 1 (ie. one where pre-/post- removal points were already on some segment of the cable), we do not need to use guards to implement these dynamics. Instead, for any given configuration, we can evaluate the wrap points as a pure function of $\vec{q}$ (though, see later discussion on exceptional configurations, where guards may prove useful).

The relevant parameter is the angle each vertex makes with respect to the line from $\vec{p}_A$ to $\vec{p}_B$. Whichever vertex makes the greatest angle, lower-bounded at 0 (by the choice of the opposite terminal point itself), that point will be the wrap point. Once the proximal wrap points have been determined for each terminal point, the cable can be straightforwardly wrapped along the perimeter of the figure to connect the two points. Figure 2-4 illustrates this procedure graphically.

There are some exceptional circumstances where these wrapping rules do not supply qualitatively correct behavior. One exceptional case is when the neighboring pulley to an n-gonal pulley enters within the hull of that pulley. Another is when the n-gonal pulley lies on the side of the wrapping half-plane of the system, but not between points $\vec{p}_A$ and $\vec{p}_B$. One solution to the halfspace issue would be to employ guards to transition the system upon breaking contact from a system which dynamically wraps the pulley to one which does not include that pulley, switching back to the wrapping case when that pulley again crosses that half-plane between $\vec{p}_A$

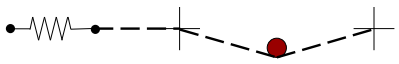


Figure 2-5: A soft membrane is modeled as a cable which is stretched between two radius-zero pulleys and which terminates in a Hooke's law spring. The manipuland interacting with this membrane can be treated as another pulley, situated between these pulleys.

and $\vec{p}_B$. For our purposes, we will not define the behavior for these exceptional areas of operation. Instead, we will ensure that they never arise in the systems we create.

2.5 Modeling Deformable Surfaces Using Cables and Springs

The simplest soft robot model that we investigate is a soft membrane stretched taught between two points, with a linear spring at the end (Figure 2-5). The system behaves similarly to an elastic membrane, even though all of the constraints are rigid at every point.

Furthermore, we allow for dynamically engaging and disengaging pulley wrap locations, allowing for objects to briefly break and make contact with our soft membrane. This relatively simple model captures a lot of the qualitative behaviors of a soft membrane, include restorative forces directed toward the center of operation, and a force that grows with the degree of deformation within the surface. We look more closely at the energetics of this system in section 3.2, where we validate our model experimentally, and in section 4.2, where we will reason about its steady-state behavior to perform an in-hand manipulation task.

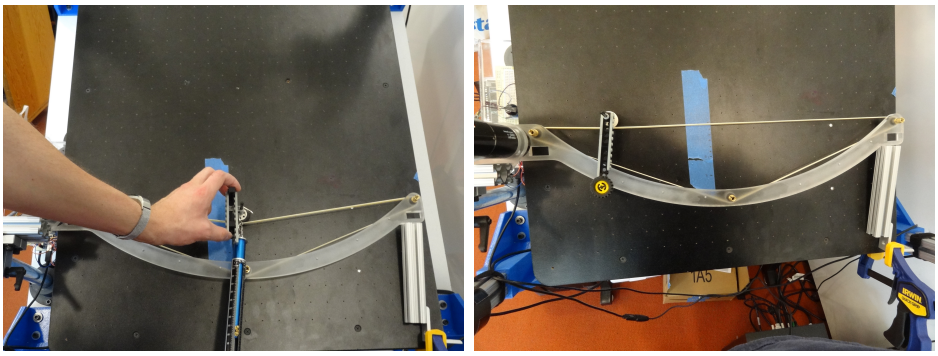
Chapter 3

Physical Experiment

It is important that we validate our model as it pertains to physically realizable 2D soft grip systems. In order to do this, we measured the static force exerted at a number of deflections in a physical realization of a cable membrane. We took advantage of the hardware setup which had been constructed by the authors of Katzschmann, et al. for their work on controlling a restitutive elastic interaction [5]. In this case, a rubber-band is used as the cable, which allows it to provide a restorative spring force while also deforming in the same manner as the cable. Our results show a tight match between our model and the empirical measurements.

3.1 Experiment Setup and Data Collection

Our experimental setup consists of a rubber band wrapped taut around three pulleys (Figure 3-1). Two of these pulleys are spaced widely apart from one another, constituting our soft membrane. The rubber band is pre-loaded with a good amount of tension. Measurements were taken with a roller wheel as the interface between the measuring device and the rubber band surface, essentially eliminating the effects of longitudinal friction or shear in the system. Two types of measurement were made at each location. The first measurement employed a strain gauge sensor to determine the vertical component of the force at every point. The second measurement was made using a special device to ascertain the direction of the force at every point.



(a) (b)

Figure 3-1: The experimental setup used to validate our model. A rubber band stretched taught between two pulleys serves as our soft membrane. Force is measured at a number of deformation points of the cable, stretching the cable down to each such point. The force was characterized via two types of measurement: (a) the vertical component of the force was measured using a linearly-constrained strain gauge apparatus with a roller wheel at the interface to the band, and (b) the angle of the force at every point was measured using a swiveling apparatus which was free to rotate to a minimal energy orientation, also with a rolling interface. For this second device, the measurement was the angle made by the device when the roller was made to settle at $\vec{p}$.

3.2 Model and Results

We fit of our spring tension model to this system. Our rubber band has a length of 35cm when loose, compared to a length of 115cm in its stretched configuration. Thus, for this system, the spring is "pre-loaded" with a certain amount of tension, even before our object penetrates the membrane surface. We thus fit our model using two parameters: the spring constant k, and the amount of pre-loading L_0 in the spring (ie. how far away the spring has been stretched from its minimum-energy configuration).

In our experimental setup, we measured the static force $\vec{F}$ exerted by the rubber band for all point-deformations $\vec{p}$ of the membrane. We can find this force in our model using the energy of the system:

$$E(\vec{p}) = \frac{1}{2}k(\Delta L)^2 = \frac{1}{2}k(|\vec{p} - \vec{p_1}| + |\vec{p} - \vec{p_2}| - |\vec{p_1} - \vec{p_2}| - L_0)^2$$

$$\vec{F} = -\frac{\partial E(\vec{p})}{\partial \vec{p}}^\top = \frac{1}{2}k(|\vec{p} - \vec{p_1}| + |\vec{p} - \vec{p_2}| - |\vec{p_1} - \vec{p_2}| - L_0)\left(\frac{\vec{p} - \vec{p_1}}{|\vec{p} - \vec{p_1}|} + \frac{\vec{p} - \vec{p_2}}{|\vec{p} - \vec{p_2}|}\right)$$

With this equation in hand, we can compare our model against the real measured forces (Figures 3-2 and 3-3) and we find that it is a very close fit.

3.3 Discussion

We have validated our model using a stretched rubber band as our membrane. However, a more accurate physical representation of our system would be a fixed-length cable, stretched between two pulleys and terminating in a spring. By fitting our model to the rubber band setup, we demonstrate a degree of generalization to a different class of soft membrane.

Friction is still being ignored in this experiment, and that should be addressed. One can envision a scenario in which the friction of the interaction between the rubber band and the manipuland is reduced—for example, if the manipuland is wet, or if it is made of a very smooth plastic, or if it is itself covered in rollers—but, in most cases, the rubber band will exert extreme frictional forces against any gripped object. In such cases, particularly when the object being grasped has lots of mass, a major effect of this will be an asymmetry in the tension on either side of the manipuland. These shear forces could be useful for tasks such as screwing in a nut, where the frictional interface can exert strong torques on the manipuland. In the future, extending our low-dimensional model in such a way as to capture longitudinal friction forces could be powerful way to expand its usefulness.

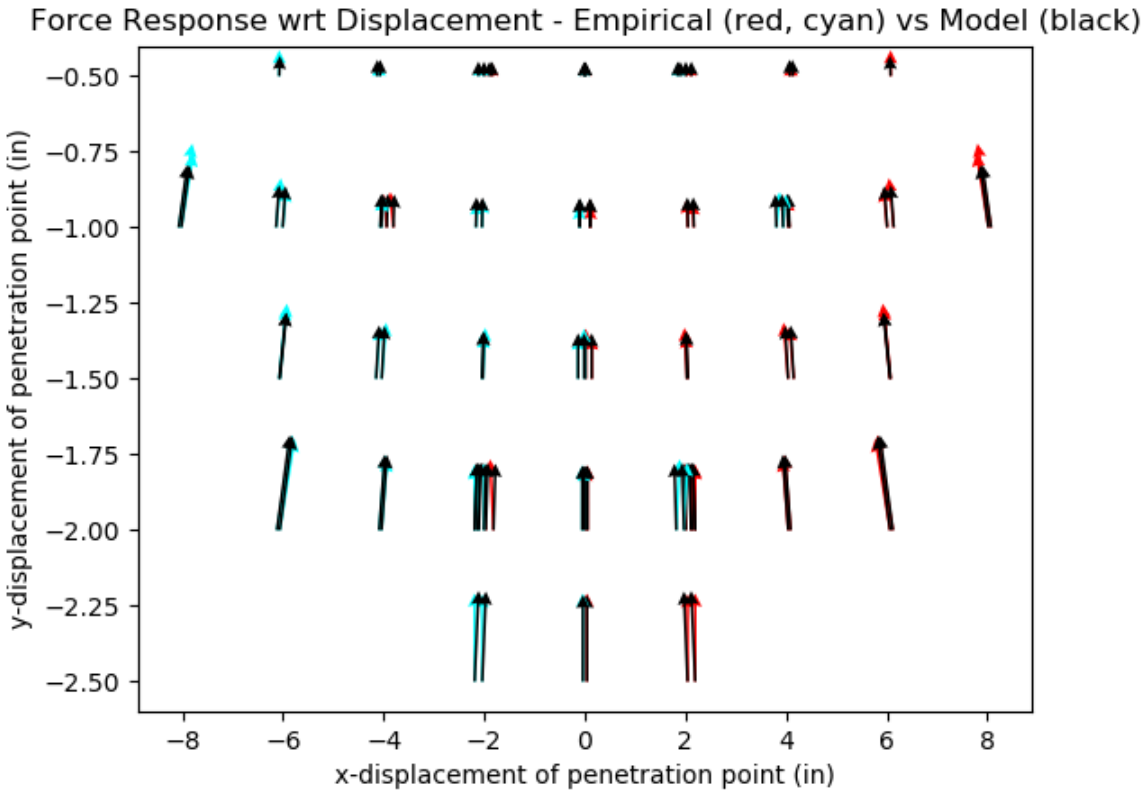


Figure 3-2: Plot of empirically measured forces (red) matched to the predicted model force (black) at every displacement point $\vec{p}$. Each arrow corresponds to a single measurement. For intuition, the horizontally mirrored forces (predicted to be symmetric under our model) are included as well (cyan).

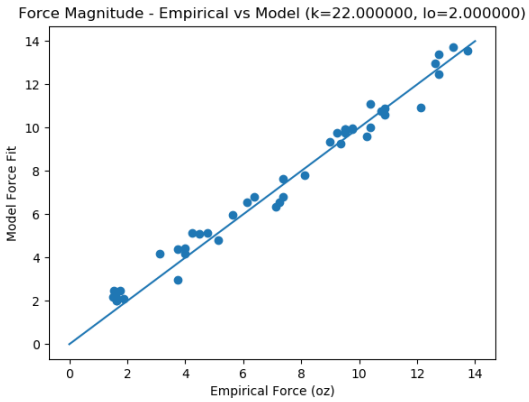


Figure 3-3: Plot of empirically measured forces (x-axis) versus model forces (y-axis).

Chapter 4

Statics-Based Jacobian Control of Soft Hand

We consider the in-hand repositioning of a rigid manipuland grasped by soft-tipped fingers (Figure 4-1).

4.1 Hand Model

The hand model has the following state vector:

$$\vec{q}_{\text{manip}} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}, \vec{q}_{\text{robot}} = \begin{bmatrix} \theta_{p1} \\ \theta_{l1} \\ \theta_{d1} \\ \theta_{s1} \\ \theta_{p2} \\ \theta_{l2} \\ \theta_{d2} \\ \theta_{s2} \end{bmatrix}, \vec{q} = \begin{bmatrix} \vec{q}_{\text{manip}} \\ \vec{q}_{\text{robot}} \end{bmatrix}$$

For each of the top and bottom phalanges, there are four parameters: θ_p, the proximal angle, which represents the angle of the left-most joint; θ_l, the linear displacement along the horizontal linear degree of freedom; θ_d, the distal angle, which

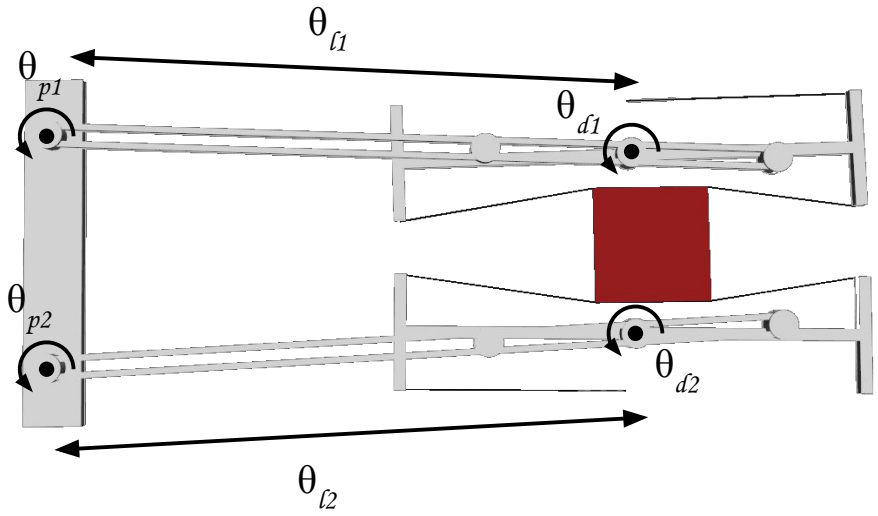


Figure 4-1: A soft-fingered hand, implemented using the cable model. The hand contains two soft membrane models, which are both engaged in gripping a red, four-sided manipuland.

represents the angle of the right-most joint with respect to the linear actuator; and θ_s, a parameter specifying the displacement of the linear spring. Each phalanx is actuated along θ_p,θ_l, and θ_d. The cables each terminate in a linear spring, also rigidly affixed to the paddle, allowing for the membrane to deform according to a Hooke's law spring energy equation $U = \frac{1}{2}k\theta_s^2$. This spring will also have a damping term, to model the inelastic behavior of the soft membrane which we are trying to model.

The cables in the model are strung between the two corners of each paddle, making contact with the manipuland in between. This manipuland is modeled as an n-gonal pulley, and so our dynamics allow for it to make and break contact with the paddles. For the remainder of this section, we will consider the case of square pulley which does not reorient in theta. The values b_x, b_y specify the box-shaped manipuland's x- and y- position.

4.2 SOCP to find Static Equilibrium

We are hoping to find the steady-state static equilibrium for our cable dynamical system. The manipulator equations for this constrained dynamical system are:

$$\mathbf{H}(q)\dot{\vec{v}} + \vec{c}(\vec{q}, \vec{v}) = \mathbf{B}\vec{u} + \mathbf{J}^\top \vec{\lambda}$$
$$\vec{\phi}(q) = \dot{\vec{\phi}}(q) = \ddot{\vec{\phi}}(q) = 0$$

We can find the steady-state static equilibrium by setting $\vec{v} = \dot{\vec{v}} = 0$. Doing this in the above reduces our manipulator equations down to

$$\vec{c}(\vec{q}, 0) = \mathbf{B}\vec{u} + \mathbf{J}^\top \vec{\lambda} \tag{4.1}$$
$$\vec{\phi}(q) = 0. \tag{4.2}$$

We have $\dim(q) + \dim(\phi)$ equations in $\dim(q) + \dim(\phi)$ unknowns. Thus, the above equations should encompass all the static equilibria of our system.

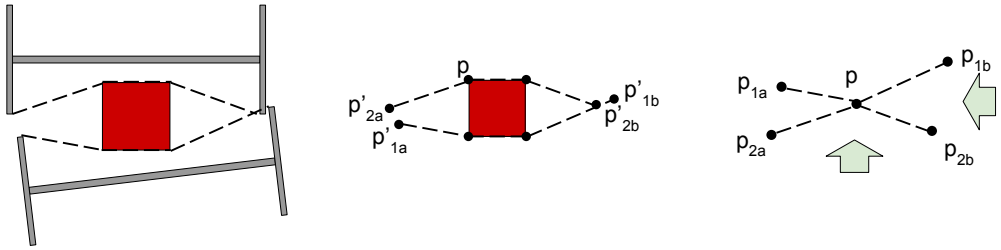


Figure 4-2: Reduction from finding box steady-state to finding point steady-state for fixed fingertip locations. We model the finger paddles which are making contact with the manipuland via taut cables (left). Mathematically, we need only consider the cable wrap points (middle). Because we assume the box does not undergo rotations, we can shift the finger points by constant offsets in order to reduce the manipuland to a point (right).

For our purposes, consider the task of using our soft hand to manipulate a box which is able to translate but which cannot rotate. Let us assume that the top paddle makes contact with both corners of the top surface of this box and that the bottom paddle makes contact with the bottom surface. Our goal is to find the steady-state equilibrium location of the manipuland $\vec{p}$ given that the fingers are fixed in place. In this case, we can reduce the problem to that of grasping a solitary point (Figure 4-2).

If we fix the finger paddles in place, then the state variables of our system consist only of the point-manipuland $\vec{p}$ and the spring offsets x_1, x_2:

$$\vec{q} = \begin{bmatrix} p_x \\ p_y \\ x_1 \\ x_2 \end{bmatrix}, \vec{v} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

The manipulator equations for this system have the form:

$$\mathbf{H}\dot{\vec{v}} + \mathbf{G}\vec{q} = \mathbf{J}^\top \vec{\lambda}$$

$$\vec{\phi}(\vec{q}) = \begin{bmatrix} \phi_1(\vec{q}) \\ \phi_2(\vec{q}) \end{bmatrix} = 0$$

Thus, the steady-state can be found as:

$$\mathbf{G}\vec{q} = \mathbf{J}^\top \vec{\lambda}$$
$$\vec{\phi}(\vec{q}) = 0$$

The potential force matrix G encodes the spring forces exerted on x_1 and x_2.

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_1 & 0 \\ 0 & 0 & 0 & k_2 \end{bmatrix}$$

And the Jacobian can be derived in a straightforward manner:

$$\vec{\phi}(\vec{q}) = \begin{bmatrix} -L_1 + ||\vec{p} - \vec{p}_{1a}||_2 + ||\vec{p}_{1b} - \vec{p}||_2 + x_1 \\ -L_2 + ||\vec{p} - \vec{p}_{2a}||_2 + ||\vec{p}_{2b} - \vec{p}||_2 + x_2 \end{bmatrix}$$
$$\mathbf{J} = \frac{\partial \vec{\phi}}{\partial \vec{q}} = \begin{bmatrix} (\frac{\vec{p} - \vec{p}_{1a}}{||\vec{p} - \vec{p}_{1a}||_2} + \frac{\vec{p}_{1b} - \vec{p}}{||\vec{p}_{1b} - \vec{p}||_2})^\top & 1 & 0 \\ (\frac{\vec{p} - \vec{p}_{2a}}{||\vec{p} - \vec{p}_{2a}||_2} + \frac{\vec{p}_{2b} - \vec{p}}{||\vec{p}_{2b} - \vec{p}||_2})^\top & 0 & 1 \end{bmatrix}$$

This gives us the following equations:

$$0 = \lambda_1(\frac{\vec{p} - \vec{p}_{1a}}{||\vec{p} - \vec{p}_{1a}||_2} + \frac{\vec{p}_{1b} - \vec{p}}{||\vec{p}_{1b} - \vec{p}||_2}) + \lambda_2(\frac{\vec{p} - \vec{p}_{2a}}{||\vec{p} - \vec{p}_{2a}||_2} + \frac{\vec{p}_{2b} - \vec{p}}{||\vec{p}_{2b} - \vec{p}||_2})$$
$$k_1 x_1 = \lambda_1$$
$$k_2 x_2 = \lambda_2$$

And finally we arrive at:

$$\begin{aligned} 0 = {} & k_1 \left(L_1 - ||\vec{p} - \vec{p}_{1a}||_2 - ||\vec{p}_{1b} - \vec{p}||_2\right) \frac{\partial \vec{\phi_1}}{\partial \vec{q}} \\ & + k_2 \left(L_2 - ||\vec{p} - \vec{p}_{2a}||_2 - ||\vec{p}_{2b} - \vec{p}||_2\right) \frac{\partial \vec{\phi_2}}{\partial \vec{q}} \end{aligned} \tag{4.3}$$

The problem of solving for $\vec{p}$ given $\vec{q}$ can be posed as a second-order cone program

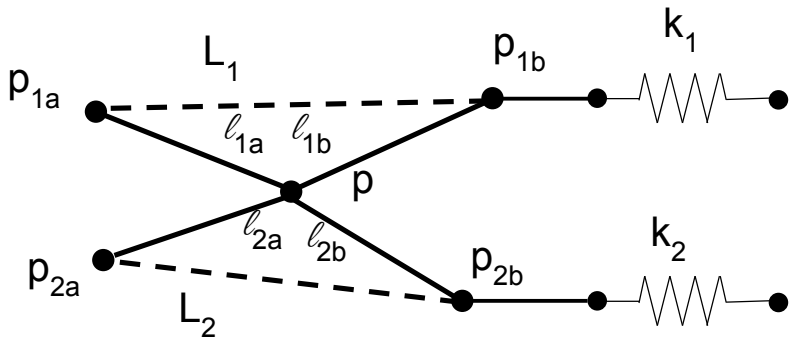


Figure 4-3: A pictorial representation of our optimization problem. The fingertip points $\vec{p}_{ij}$ are fixed in place. The decision variable $\vec{p}$ is free to move around, and the decision variables l_{ij} should correspond to the norm lengths of the different cable segments. The amount that each cable is deflected corresponds to the amount of energy stored in the each spring. Our SOCP seeks an energy minimum, where the forces on the at-rest manipuland are zero.

with Lorentz cone constraints. Solving equation 4.3 is equivalent to minimizing the energy equation:

$$\min_{p} \frac{1}{2}k_1 \left(||\vec{p} - \vec{p}_{1a}||_2 + ||\vec{p}_{1b} - \vec{p}||_2 - L_1\right)^2 + \frac{1}{2}k_2 \left(||\vec{p} - \vec{p}_{2a}||_2 + ||\vec{p}_{2b} - \vec{p}||_2 - L_2\right)^2$$

By introducing decision variables l_{ij} to represent the norms, we can solve the above optimization as follows:

$$\min_{\vec{p}, l_{ij}} k_1(l_{1a} + l_{1b} - L_1)^2 + k_2(l_{2a} + l_{2b} - L_2)^2$$
$$l_{ij} \geq ||\vec{p} - \vec{p}_{ij}||_2 \ , \forall (i,j) \in \{1,2\} \times \{a,b\}$$

In this case, we rely on the fact that our objective is pushing down on the decision variables l_{ij} in order to drive them to equality on their constraints. This relies on the important fact that the terms $(l_{1a} + l_{1b} - L_1)$ and $(l_{2a} + l_{2b} - L_2)$ will always be nonnegative, which we can see by applying the triangle inequality to the Lorentz cone constraints lower-bounding the variables l_{ij}. Thus, a solved version of this problem should satisfy $l_{ij} = ||\vec{p} - \vec{p}_{ij}||_2$. Thus, our objective will represent the true energy minimum (Figure 4-3).

The above does not provide a general solution to equations 4.1 and 4.2. In particular, the requirement that the objective always be pushing down on the norm decision variables prevents adding in multiple moving points or other geometric transforms. To solve these more general equations, local gradient-descent methods would likely work well, as the search could be initiated at every time-step with the result from the previous time-step.

As for which problems this second-order cone program could be made to solve to global optimality, there are a few extensions which could be handled. As long as there is still only one position decision variable $\vec{p}$ that all cable segments connect to (before terminating in springs), then this optimization should be able to handle any number of additional cables. Furthermore, we discuss in section 3.2 that we might

want to handle the case of springs with some pre-tensioning offsets $L_{0,i}$. As long as these offsets increase the tension of the springs (not decrease), the above formulation could handle this by simply adding these offsets into the L_i terms.

4.3 In-Hand Manipulation Task: Positioning a 2D Object

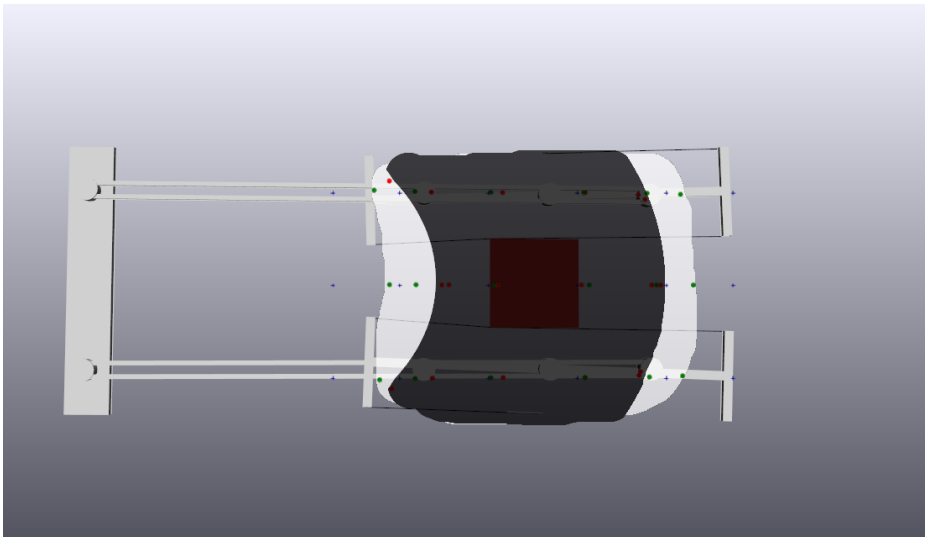


Figure 4-4: Results are shown for two different controllers that were made to move the manipuland to specific locations (dots). The steady-state Jacobian controller achieves a greater range of in-hand positionings (light region) than the range of motion of the fingers themselves (dark region).

With the static equilibrium solver described in section 4.2, we can create a controller that reasons about how our fingers influence the position of the manipuland.

Let $\vec{b}$ be the position of the manipuland in our state vector, and let $\vec{q}|_u$ be those elements of our state vector which are actuated. For a given positioning of the fingertips $\vec{p}_{ij}$, our SOCP allows us to ascertain the predicted steady-state pose of the manipuland. We can take the Jacobian $\frac{\partial \vec{b}}{\partial \vec{p}_{ij}}$ of this pose to approximate the kinematic relationship between the box and the fingertips. Furthermore, we can use forward-kinematics to obtain the Jacobian of our fingertips with respect to the actuated state

$\frac{\partial \vec{p}_{ij}}{\partial q|_u}$. We have thus approximated the relevant Jacobian:

$$\frac{\partial \vec{b}}{\partial q|_u} = \frac{\partial \vec{b}}{\partial \vec{p}_{ij}} \cdot \frac{\partial \vec{p}_{ij}}{\partial q|_u}$$

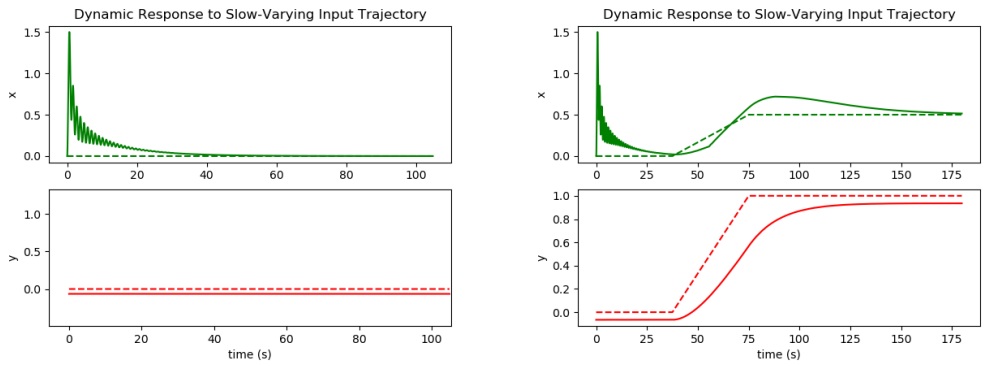
We can then use feedback control to bring the manipuland to its target pose as:

$$\dot{q}|_{u,\text{des}} = -\frac{\partial \vec{b}}{\partial q|_u} \left[k_p(\vec{b} - \vec{b}_{\text{des}}) + k_d \dot{\vec{b}} \right] \tag{4.4}$$

We implemented the cable constraint model and controller in the Drake software library for robot simulation. Simulated results were obtained using a slower-than-life simulation. We employed a two-stage controller: this included an inner-loop position controller, which was keeping the hand in a specified pose, and an outer-loop PD controller, which used the SOCP-derived Jacobian to change the desired hand pose based on the error in the manipuland location (equation 4.4). In this case, we used the parameters: $k_s = 150.0$, $b_s = 550.0$ (springs' stiffness and damping), $k_{p,\text{outer}} = 0.1$, $k_{b,\text{outer}} = 0.58$, $\Delta t = 0.053$, $k_{p,\text{inner}} = 500.0$, $k_{d,\text{inner}} = -2\sqrt{k_{p,\text{inner}}}$. The first experiment involved ascertaining the set of reachable manipuland positions under the Jacobian controller. As shown in Figure 4-4, the steady-state kinematic controller was able to increase the in-hand manipulation range of the hand. It does this by leveraging the cable mechanics, taking advantage of skew orientations of the fingers to stably hold the manipuland at distances beyond the joint limits.

The second experiment concerned the dynamics of the system. Figure 4-5 shows that the system is stable, and converges to the proper values. One issue is the persistent offset that seems to appear for certain desired positions. Using a PID controller rather than a PD controller might solve this problem.

Thus, for this relatively simple case, we have demonstrated the ability to apply classical control techniques to a soft robotic system for grasping.



(a) (b)

Figure 4-5: Some sample trajectories of the controller given a time-varying trajectory $\vec{b}_{\text{des}}$. Desired location of the manipuland shown with dashed lines, actual system response shown with solid lines. Figure (a) shows a constant desired location setting of $[0\ 0]^\top$, whereas the desired location in Figure (b) changes linearly to $[0.5\ 1.0]^\top$. For both of these runs, the hand was started in a non-grasping position, then was brought swiftly by our controller into a pinching grasp, and then was smoothly controlled thereafter. This startup activity accounts for the oscillatory settling period observed at the beginning of each plot, before the extra dynamic energy is swiftly dissipated by spring damping forces and the controller's influence.

Chapter 5

Discussion and Conclusion

We have presented a simple dynamical system which captures many of the qualitative behaviors of real soft systems. The model does this while maintaining the simplicity of constrained rigid body dynamics. Although the control techniques presented in this paper were relatively simple, the next steps would be to move on to more sophisticated validation techniques, such as trajectory optimization, model-predictive control, and Lyapunov verification. These techniques require that the dimensionality of a system remain relatively small, and thus our model is well-suited to them.

One limitation of the current form of the model is the lack of all friction at the interface between the manipuland and the manipulator. Adding friction has the potential to introduce a few more state variables, and leads to some of the intricacies encountered in modeling rigid-body contact. However, whether a penetrative friction model or a stateless friction model or third intermediate model were to be employed, all such extensions should only add one or two new state variables to the space, and so we would not expect to lose the power of this model for sophisticated analysis.

Bibliography

[1] Joseph T Belter, Jacob L Segil, and BS SM. Mechanical design and performance specifications of anthropomorphic prosthetic hands: a review. *Journal of rehabilitation research and development*, 50(5):599, 2013.

[2] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.

[3] Christian Duriez. Control of elastic soft robots based on real-time finite element method. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3982–3987. IEEE, 2013.

[4] Christian Duriez and Thor Bieze. Soft robot modeling, simulation and control in real-time. In *Soft Robotics: Trends, Applications and Challenges*, pages 103–109. Springer, 2017.

[5] Robert K. Katzschmann, Aykut C. Satici, Daniela Rus, and Russ Tedrake. The Soft Juggler: Model-based Control of a Dynamically Dexterous Soft Robot under Large Deformations. In *Soft Robotics Journal (under review)*, 2017.

[6] Frederick Largilliere, Valerian Verona, Eulalie Coevoet, Mario Sanz-Lopez, Jeremie Dequidt, and Christian Duriez. Real-time control of soft-robots using asynchronous finite element modeling. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2550–2555. IEEE, 2015.

[7] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[8] Koichi Suzumori, Shoichi Iikura, and Hiroshisa Tanaka. Applying a flexible microactuator to robotic mechanisms. *IEEE control systems*, 12(1):21–27, 1992.

[9] Russ Tedrake. Underactuated robotics: Learning, planning and control for efficient and agile machines. *Course notes for MIT*, 6:832, 2009.

[10] Robert J Webster III and Bryan A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.