EXPERIMENTS IN QUASI-STATIC MANIPULATION OF AN ELASTIC ROD

BY

DENNIS MATTHEWS

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Associate Professor Timothy Bretl, Chair
Professor Andrew Alleyne
Professor Seth Hutchinson
Professor Jose Schutt-Aine

ABSTRACT

The purpose of this dissertation is to experimentally validate a new approach to robotic manipulation of deformable objects. As a case study, it will focus on the manipulation of objects that can be modeled as Kirchhoff elastic rods, for example a metal wire that is held at each end by robotic grippers. Any curve traced by this wire when in static equilibrium can be described as the solution to an optimal control problem with boundary conditions that vary with the position and orientation of each gripper. Recent work has shown that the set of all local solutions to this problem over all possible boundary conditions is a smooth manifold of finite dimension that can be parameterized by a single chart, the coordinates for which have a direct interpretation as forces and torques. These coordinates—in principle—allow the problem of manipulation planning to be formulated as finding a path *of the wire* through its set of equilibrium configurations, something that was previously thought impossible and that has significant advantages. However, this approach has never before been applied to hardware experiments.

We begin by considering a metal wire that is confined to a planar workspace. We derive global coordinates for this wire and characterize the extent to which they accurately describe its shape during robotic manipulation. In particular, we show that differences between predicted and observed manipulation (which can be quite large) derive primarily from small errors in the position and orientation of each robotic gripper. We reduce these differences in two ways. First, we give an algorithm for manipulation planning that locally minimizes sensitivity to errors in gripper placement. Second, we give a feedback control policy (based on force sensor data as well as on position and orientation estimates) that locally minimizes the sum-squared error between planned and observed paths in our global coordinate chart for the wire.

We conclude by showing—again, with hardware experiments—that these results extend directly to enable robotic manipulation of a metal wire in a three-dimensional workspace.

To my family and friends.

ACKNOWLEDGMENTS

I thank my advisor, Timothy Bretl, for his guidance through the completing of my PhD degree. Thank you for never letting me settle for "good" but insisting I push until I achieve excellence. I would like to thank the remainder of my committee, which includes Andrew Alleyne, Seth Hutchinson, and Jose Schutte-Aine. You all have been great mentors. All of the members of my committee have been role models of innovation and technical leadership.

I thank the best parents I could ask for, Jackie and Dennis Matthews. Thank you for your unwavering support and love in all that I do. Thank you for all of the sacrifices so that I could have every opportunity available. Thank you for the company on those late nights in the lab, the fish dinners on Fridays, the steak dinners on Sundays, and so much more.

I thank my first teacher, Prudence Sanders, who encouraged me and strengthened my analytical capabilities at an early age. You have always supported, corrected, and protected me. Thank you to her family, Frank, Brooke, and Grant, for continued uplifting words.

I thank my grandparents, Tillis and Lucille Rendleman, for incessant prayers and my grandparents Molly and Lee for your guidance from above. I also thank countless cousins, aunts, and uncles from my Brooks, Davis, Matthews, and Rendleman families.

I thank my God parents, Polly and Eugene Washington. You have been parents, friends, and confidants. I also thank Pastor Lewis and all of my previous pastors of Bethel A.M.E. Church. In addition, I thank the rest of my Bethel family that has treated me as a brother, son, and grandson. You have opened your homes and hearts and have been present at every stage of my academic career. I thank all of the members that have supported me my entire life, which include the Arrington, Banks, Barnes, Barrow, Bland, Brown, Donaldson, El-Jamal, Gray, Hayes, Haywood-Benson, Hewitt, Hill, Hines, Hudson, Hursey, Latham, Lomax, Lotts, McGavock, Mc-

Mullen, Meares, Miller, Mitchell, Mooney, Nelson, Nichols, Pope, Ransom, Renfro, Rivers, Rollins, Session, Suggs, Sayles, Turner, Warner, Woodruff, and Young families.

I thank Suzanne Gray for all for your support through my graduate studies. I also thank the Gray and Lycan families for your continued words of encouragement.

I thank all of the staff at B.T. Washington Elementary School, Jefferson Middle School, and Centennial High School. You all set me up for success from my first day in kindergarten.

I thank the deans in the Morrill Engineering Program, including Paul Parker, N. Jonne Brown, and Ivan Favilla. I also thank Preston Ransom for the invitation to the Merge Program and also for the SURGE Fellowship. You all have continued to inspire me and so many others who have graduated with degrees in engineering at the University of Illinois.

I thank everyone in the ECE Department at the University of Illinois. I especially want to thank Dan Mast, Walthamuel Smith, Frank Dale, and the rest of the members of the ECE Parts Shop for your support as a student and an instructor. I also thank Patricia Franke, Janak Patel, and Donna Brown, for the teaching opportunities. In addition, I thank Steven Franke Jen Carlson, Laurie Fisher, all of the professors, and all of the support staff for helping to make my time in the department a home away from home.

I thank the members of my research group. Miles Johnson, we have definitely defied many odds over the years. Thank you Jamie Norton, Andy Borum, and Or Dantsker for your friendship and all of your time as I completed my final defense. I am here any time you need me. I thank Aadeel Akhtar, Aaron Becker, and Abdullah Akce, and Navid Aghasadeghi for being great examples of researchers and lab mates. I also thank Zoe McCarthy and Andy Borum for your expertise with regard to elastic rod theory and all of the past and current members of the Bretl Lab including Sharath, Joe, Dave, Mary, Xinke, Carlos, Sophie, Cem, Jessica, Patrick, Michael, and Edward

Last, but not least, I thank countless friends and family members who have helped to make me the person I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GPIO	General Purpose Input Output
HSV	Hue Saturation Value
PMP	Pontryagin Maximum Principle
PRM	Probabilistic Roadmap
PI	Proportional Integral
RRT	Rapidly-Exploring Random Tree
RGB	Red Green Blue
SCARA	Selective Compliance Assembly Robot Arm
TCL	Tool Command Language

LIST OF SYMBOLS

$\mathcal{A}$	A-Space
$\mathcal{B}$	B-Space
$J(1, a)$	Jacobian at End of Elastic Rod
Γ	Estimator Constant that Weights Force Sensor Measurements
K_p	Proportional Gain
K_i	Integral Gain

CHAPTER 1

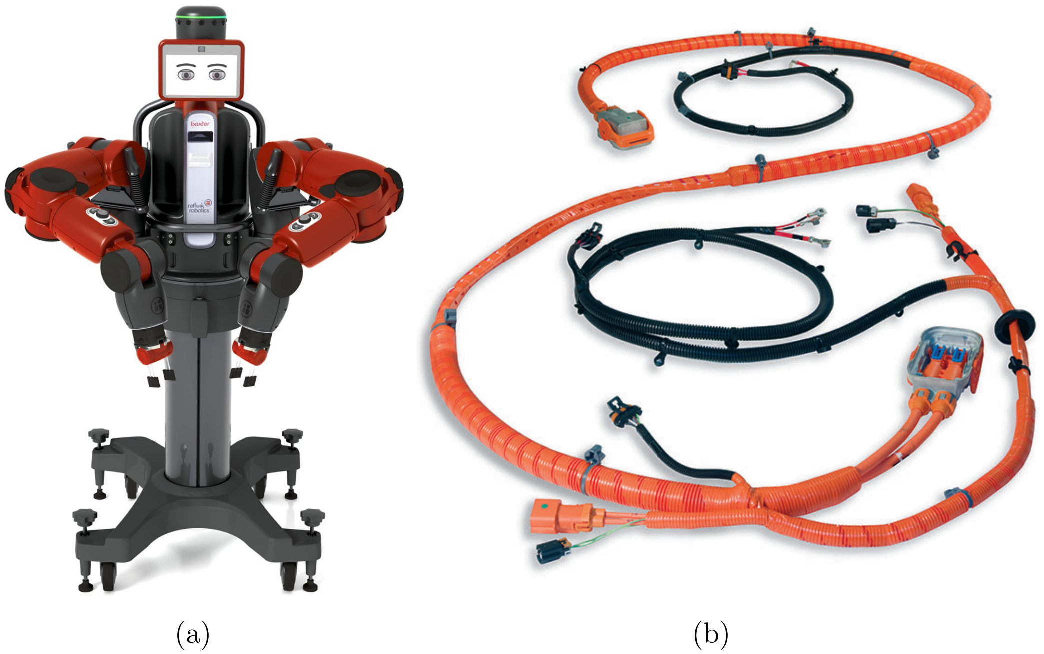
INTRODUCTION

This dissertation presents and experimentally validates a new approach to the robotic manipulation of flexible objects, specifically "linear deformable objects" that can be described by the shape of their centerline. In this introduction, we will present a motivating application (Section 1.1), state a canonical problem that must be solved in order to enable this application (Section 1.2), describe the challenges posed by this problem (Section 1.3), and discuss our approach to addressing these challenges (Section 1.4).

1.1 Motivation

Our work has been driven by the need of small-to-medium sized businesses in the United States to automate handling and assembly of compliant parts, a critical and widespread manufacturing task that is often done manually. These businesses have unique requirements that are not served by traditional industrial robots or approaches to automation. These requirements principally derive from low production volumes (e.g., hundreds rather than millions of items) and small capital budgets. Small-to-medium sized businesses cannot afford to buy different robots for different production lines, nor can they afford to hire engineers to retool and reprogram a single robot. By some estimates, there is a $16B market for a low-cost, easy-to-use, general-purpose robot that can do simple, repetitive tasks that involve light payloads (e.g., pick-and-place, machine operation, part sorting, packing, finishing).

"Baxter" is a robot that targets this market (Figure 1.1a). It has two arms, each with seven degrees of freedom. It has a payload capacity of 5 lb. It is human scale, standing 6'1" on a pedestal. Most importantly, it is low-cost ($27K, an order of magnitude less than traditional industrial robots) and can be "programmed" by physical demonstration—simply by taking hold of



(a) (b)

Figure 1.1: (a) Baxter, a low-cost industrial robot with two arms that each have seven degrees-of-freedom. (b) Delphi wire harness. The installation of this device in an automobile is a critical manufacturing task that is hard for Baxter.

an arm and moving it to a desired location. Compliant joints with series-elastic actuators enable the physical interaction and, more broadly, enable operation in close proximity to human partners without the need for "safety cages." Vision, touch, force, and position sensors—coupled with software tools like ROS (Robot Operating System) and OpenCV (Open-Source Computer Vision)—enable recognition and handling of rigid objects. At the time of writing this thesis, it remains to be seen if Baxter will succeed in the marketplace, but early results are exciting: promotional video shows line workers teaching this robot to perform simple tasks in under 10 minutes.

A common manufacturing task that, at the time of writing this thesis, Baxter cannot perform yet is the installation of a wire harness (Figure 1.1b). This harness is a collection of wires that are joined at various points and that terminate in electrical connectors. Installation means routing these wires through an existing structure (e.g., a hybrid electric vehicle or a medical device) and plugging in the connectors. The connectors are rigid objects, so plugging them in is equivalent to classical "peg-in-hole" manipulation, for which Baxter is already well suited. However, the wires are deformable objects, and reasoning about their deformation poses a significant challenge. This challenge has so far limited even special-purpose robotic installation systems—costing an order of magnitude more than Baxter and requiring engineers to program—to success rates of 50% in the laboratory [1].

1.2 Problem

In this dissertation, we will restrict our attention to quasi-static manipulation of a thin, flexible strip of metal with an industrial robot (Figure 1.2 and Figure 1.3). The metal strip can be modeled as a planar elastic rod such as a "Kirchhoff" elastic rod [2]. So, henceforth we will refer to it as a "planar elastic rod" (or more simply as an "elastic rod" or just a "rod"). The metal strip behaves like one piece of a wire harness (Section 1.1) — a single wire — that has been confined to a horizontal plane. One end of the metal strip is attached to a table and the other end of the metal strip is held by a Selective Compliance Assembly Robot Arm (SCARA), an industrial robot. By assuming that the manipulation is performed in a quasi-static manner [3], we can ignore the dynamics of the rod. This assumption entails that given the boundary conditions of the rod, the position and orientation of the table and the robot, the rod will be in a configuration that locally minimizes the stored potential energy. For instance, an example of a manipulation task could be to move the metal strip from between the two configurations shown in Figure 1.4. Throughout the duration of performing manipulation, the elastic rod must avoid configurations that are physically infeasible. This class of infeasible configurations includes shapes in which the rod experiences instabilities, configurations in which the rod experiences a self-collision, and configurations that are physically unrealizable by the robotic manipulator.

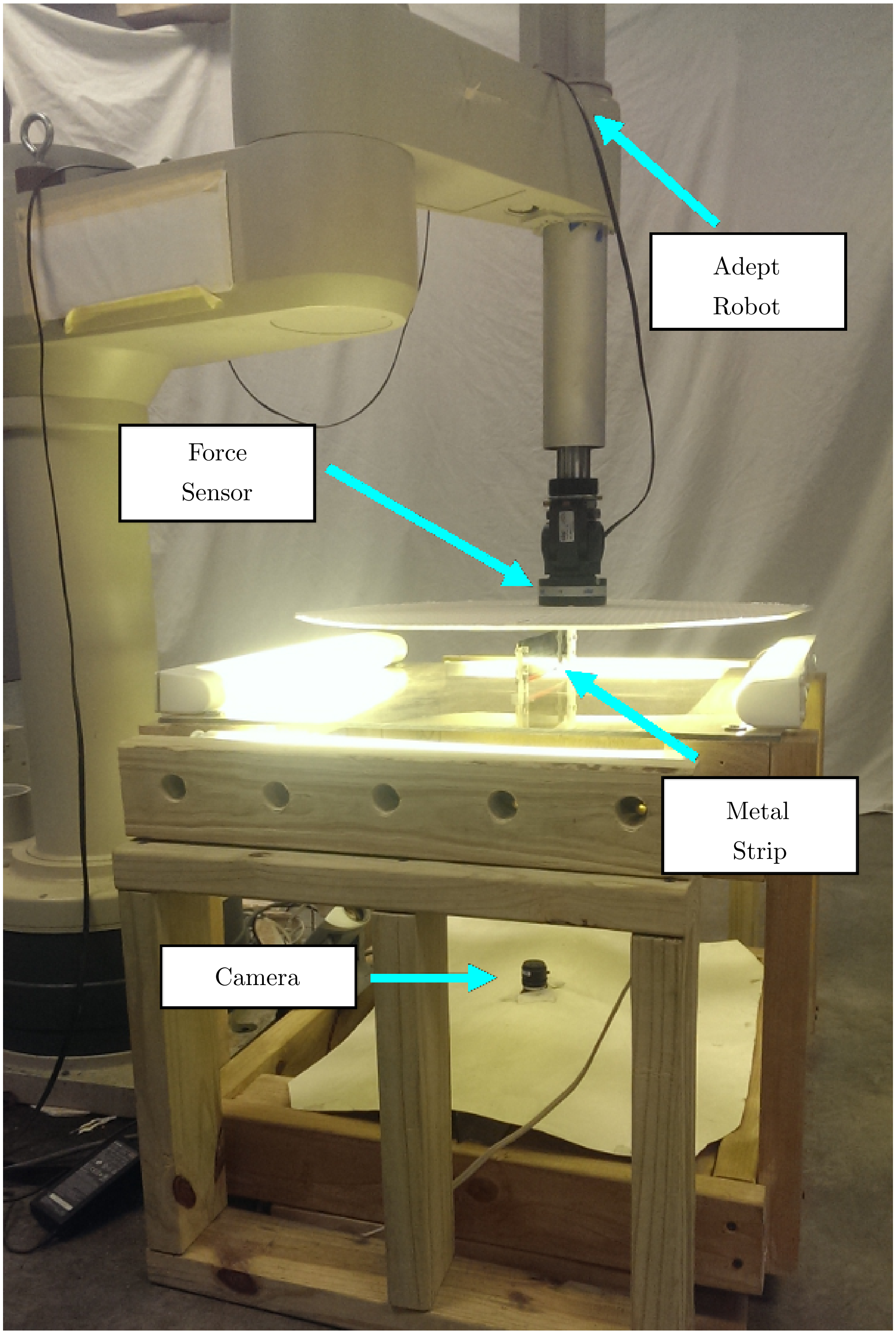


Figure 1.2: Adept XL-One robot attached to one end of a strip of blue spring steel. The other end of the strip is attached to a table with a clear, polyurethane surface. A digital camera is mounted below the surface to capture images of the strip as it is manipulated.

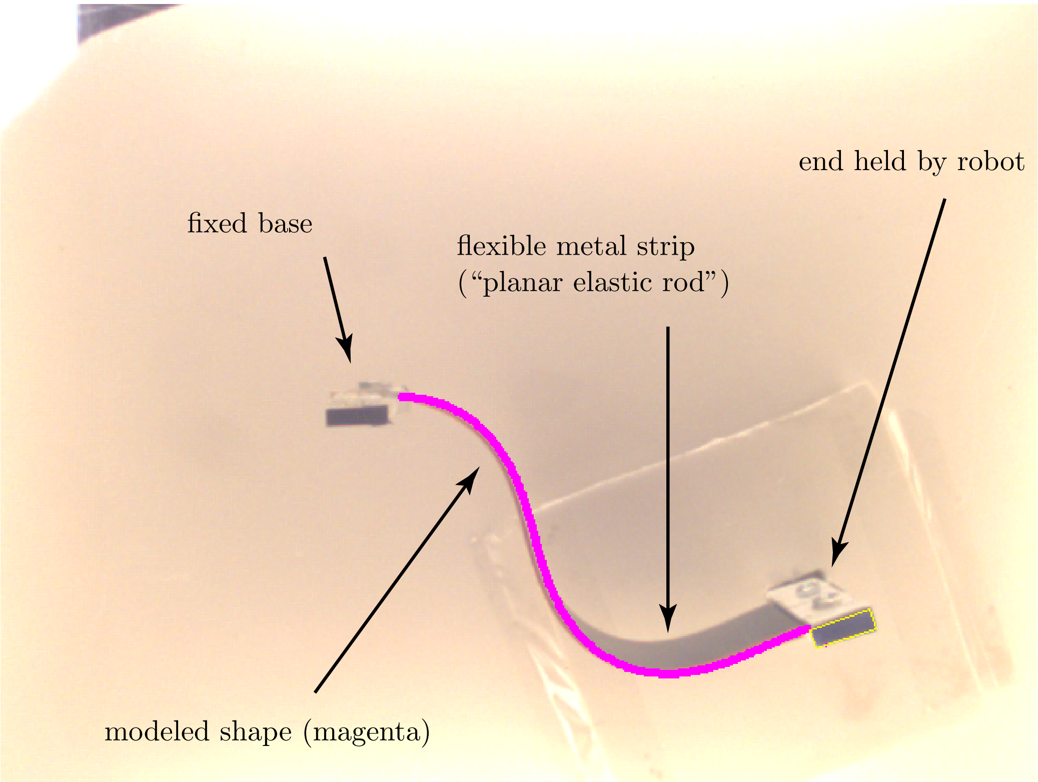


Figure 1.3: This photograph, taken using the camera below the surface of the table, depicts a thin, flexible metal strip being placed into a particular configuration. One end is secured to the table and the other end is attached to a robot. The magenta curve depicts the modeled shape of the strip.

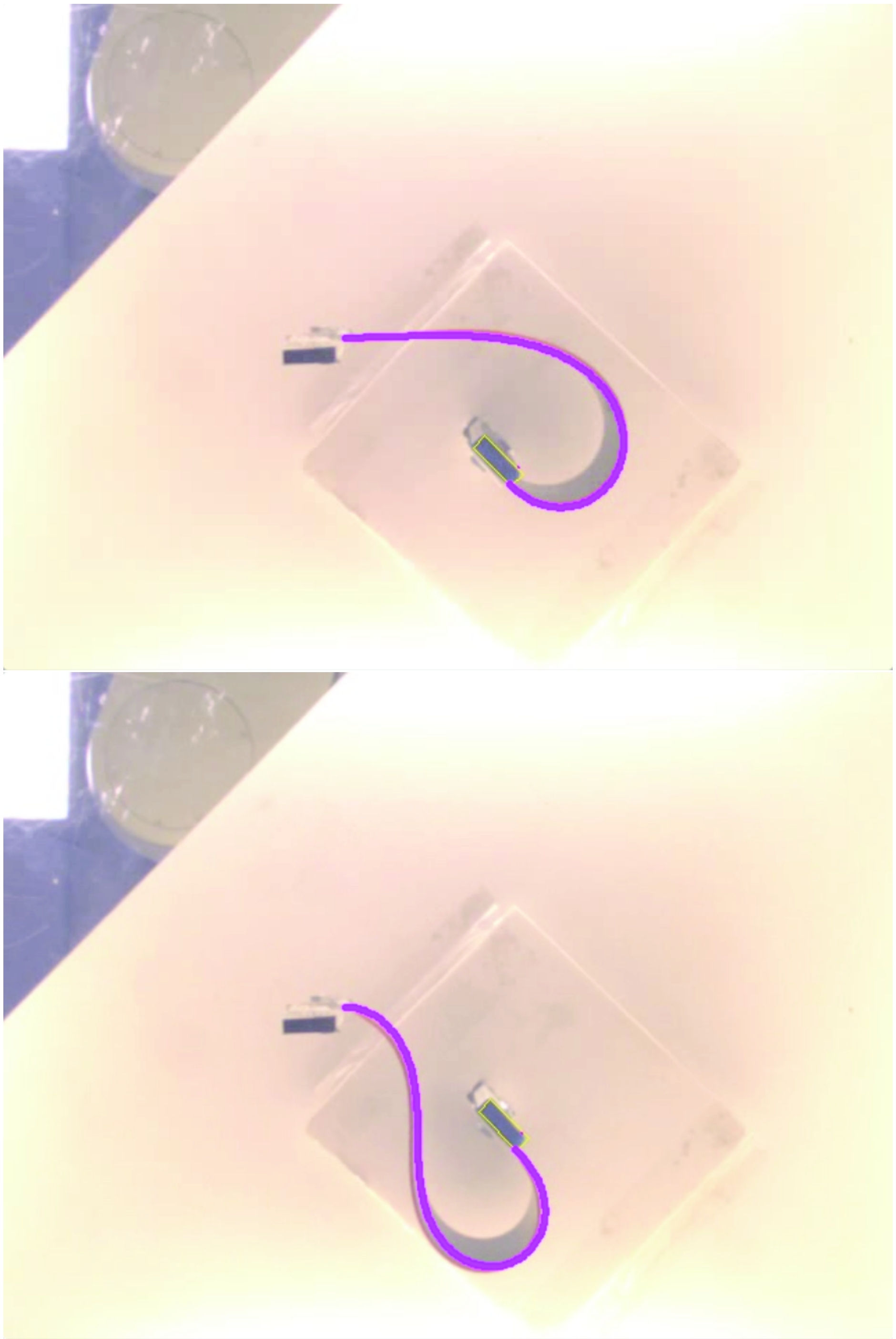


Figure 1.4: Example of an initial and goal configuration for a strip of spring steel. The magenta curve plots the modeled shape of the strip on top of the photo taken from below the surface of the experimental table. The two configurations of the planar elastic rod have the same boundary conditions. This means that the robot is placed in the exact same position and orientation. However, the metal strip has two different shapes.

1.3 Challenges

There are two reasons why quasi-static manipulation of a planar elastic rod is hard. First, the configuration space of the rod has infinite dimension. Elements of this space are continuous maps $q\colon [0,1] \to SE(2)$, the shape of which in general must be approximated. Second, a countable number of configurations may be in static equilibrium for a given placement of the robot, none of which (typically) can be computed in closed form. Because of these two challenges, techniques to perform manipulation struggle to not only describe the shape of the rod, but to also describe how the rod changes shape as the boundary conditions are altered. For these reasons, it is not clear how to plan a path *of the rod* through its set of equilibrium configurations, which is how standard methods of manipulation planning work for rigid objects.

An approach taken by Lamiraux and Kavraki [4] on manipulation of elastic objects has been applied by Moll and Kavraki [5] to "deformable linear objects" like the planar elastic rod we consider here. This previous work clearly states that the set of equilibrium configurations is the space through which one *should* construct a manipulation plan. However, it ultimately suggests exploring this set indirectly, by sampling displacements of the robot and using numerical simulation to approximate their effect on the rod.

1.4 Approach

In the remainder of this dissertation, we will provide three contributions.

- We present a concise model of the planar elastic rod. We have developed a chart that uniquely maps configurations to a point in $\mathbb{R}^3$. We show that this point in $\mathbb{R}^3$ is the force and torque at the base of the metal strip. To move from an initial to a goal configuration, we plan a path, using this global coordinate chart, that produces a sequence of configurations through which to move the robotic manipulator so the metal strip remains in static equilibrium. We show with hardware experiments that this model accurately predicts configurations that will cause the metal strip to experience an instability, configurations that will cause self-collisions, and configurations that will exceed joint limitations of the Adept robot. With hardware experiments, in addition to predicting infeasible configurations, our model can also determine when configurations of the metal strip are sensitive to small perturbations in robot placement. This material is the topic of Chapter 3.

- We present a "robust" planner that ensures that the metal strip avoids both infeasible configurations and configurations that are sensitive to small perturbations in robot placement. We have created a cost function that assesses a high cost to configurations that are sensitive to small perturbations in the boundary conditions and configurations of the metal strip that are infeasible due to instabilities, self-collisions, and joint limitations. We show with hardware experiments that planning minimum-cost paths using the total cost function avoids both configurations that exhibit large amounts of model error and that are close to being infeasible. This material is the topic of Chapter 4.

- We present a controller that compensates for any remaining differences between predicted and observed motion of the elastic rod. This controller utilizes an estimator that uses measurements of both the force at the end of the rod and the position. Both measurements are weighted based on the sensitivity of each sensor to error. Using a proportional-integral (PI) control law, we show with hardware experiments that using feedback reduces the error between the measured and observed configuration when performing manipulation. This material is the topic of Chapter 5.

The algorithms presented for manipulating planar elastic rods directly extend to the manipulation of spatial elastic rods that are not confined to the horizontal plane. We present two proof of concept prototypes for performing experiments with the Baxter robot. We provide preliminary results and provide a description for how these results can be improved by future work. This material is the topic of Chapter 6.

CHAPTER 2

RELATED WORK

In this chapter, we discuss previous work on manipulation *of* flexible objects that are modeled as elastic rods (Section 2.1) and on manipulation *by* flexible objects that are modeled as elastic rods (Section 2.2). In each case, we describe the methods used for modeling and for manipulation planning. We continue by discussing cost functions that have been used to plan optimal manipulation of flexible objects (Section 2.3), and conclude by discussing methods of incorporating closed-loop feedback (Section 2.4).

2.1 Manipulation of Flexible Objects Modeled as Elastic Rods

In this section, we describe four motion planning algorithms for manipulating flexible objects that are modeled as elastic rods. All of these methods describe the shape of a flexible object as a curve of minimum energy and present algorithms for moving from an initial shape to a goal shape. The methods proposed in Section 2.1.1, 2.1.2, and 2.1.3 are directly related to our research. These methods assume a flexible object is modeled as an elastic rod where both ends can both rotate and translate. The research presented in 2.1.4 performs manipulation with a flexible object that is treated as an elastic rod but both ends are only allowed to rotate about fixed axes that cannot translate in any direction.

While our work builds upon the motion planning algorithm presented by Moll et al., which is described in Section 2.1.1, the description that we present in this thesis provides some benefits over all of the methods discussed in this section. All of these methods, along with our method, describe the shape of a flexible object as a curve of minimum energy. However, we use three parameters to describe all of the feasible shapes of a planar flexible object. This contrasts with the methods discussed in this section that do not present a formulation for describing all feasible curves of minimum energy. Furthermore, these methods require more than three parameters to represent the shape of a flexible object.

2.1.1 Method of Moll et al.

Elastic Rod Description

Moll and Kavraki [5] describe flexible objects using a subdivision technique that involves splitting the object into smaller pieces and then describing each segment using the Frenet frame. Using this technique, the rod will be described by a $n \times 3$ matrix of parameters where n is the number of segments into which the rod has been divided. This method provides a trade-off between accuracy and representation size. The accuracy improvement comes at the expense of increasing the number of parameters required to describe the rod. The three parameters, tangent (τ), bi-normal (κ), and length (s), are used to describe each segment of a curve x. The energy of the rod is defined as

$$\text{energy}(q) = \sum_{i=1}^{n} \left({\kappa_i}^2 + {\tau_i}^2 \right) \cdot s_i. \tag{2.1}$$

Initially just three parameters are used to describe the entire rod. With each iteration, the subdivision scheme uses the difference formula

$$\left((\kappa_{i+1} - \kappa_i)^2 + (\tau_{i+1} - \tau_i)^2 \right) \cdot \max\left(s_i, s_{i+1}\right) \tag{2.2}$$

to determine adjacent segments with the largest difference in energy. One of these two segments is then divided in two and the parameters (τ and κ) are optimized to minimize the total energy of the elastic rod. Thus, each step reduces the energy by smoothing segments that vary the most. This procedure continues until the error at the endpoints and the difference between consecutive segments are both less than predetermined thresholds.

Manipulation Algorithm

Moll and Kavraki perform path planning with the elastic rod by searching a roadmap of minimal energy curves to determine a sequence of curves that move from an initial configuration to a goal configuration. The roadmap is constructed by sampling a small displacement of a robot connected to one end of the flexible object. For each displacement, the distance between a curve (q_0) and the curve that results from the robot displacement (q_1) is calculated by integrating

$$d(q_0 - q_1) = \sqrt{\int_0^1 \left((\kappa_0(s) - \kappa_1(s))^2 + (\tau_0(s) - \tau_1(s))^2\right) ds} \tag{2.3}$$

over the entire length of the curve. When the distance between the two curves is less than a parameter ϵ, the roadmap adds a connection between these two configurations. Performing motion planning to move from an initial shape to a goal shape simply requires finding a path in the roadmap between the two configurations. A local planner is then used to move between the connected configurations.

2.1.2 Method of Hermansson et al.

Recent work by Hermansson et al. [6] focuses on two problems related to the wiring of flexible cables and hoses within an automobile. This work examines not only predicting the stability of cables when the vehicle is subjected to vibrations, but also routing individual wires of a wire harness through obstacles. This manipulation problem corresponds directly with our motivating application of enabling Baxter to install a wire harness.

Elastic Rod Description

The approach taken by Hermansson et al. assumes that a cable or hose can be modeled as an elastic rod. The center line, ψ, comprises the centers of all of the cross sections along the entire length $[0, L]$. The elastic rod has a rotation vector (R(s)) composed of three vectors d_1, d_2, and d_3 where $d_3 = d_1 \times d_2$. The rod has the shearing and stretching strain vector ($\Gamma = (\Gamma_1, \Gamma_2, \Gamma_3)$) that is defined as

$$\Gamma(s) = R(s)^T \delta_s \varphi(s) - e_3. \tag{2.4}$$

The potential energy due to this type of strain is calculated by

$$w^\Gamma(s) = \frac{1}{2}\Gamma(s)^T K^\Gamma \Gamma(s). \tag{2.5}$$

The rod has the curvature/torsion strain vector ($\Omega = (\Omega_1, \Omega_2, \Omega_3)$) defined as

$$\Omega_s = R(s)^T \delta_s R(s). \tag{2.6}$$

The potential energy due to this type of strains is calculated by

$$w^\Omega(s) = \frac{1}{2}\Omega(s)^T K^\Omega \Omega(s). \tag{2.7}$$

The rod is assumed to be in equilibrium if the shape minimizes the potential energy, W, where

$$W = \int_{s=0}^{L} w^\Gamma(s) + w^\Omega(s) - K^\rho g^T(\varphi(s))ds. \tag{2.8}$$

Given a set of boundary conditions, the rod is first discretized using an adaptive finite difference algorithm. A Quasi-Newton method is then employed to minimize the potential energy function.

Manipulation Algorithm

Hermansson et al. use this formulation of the rod to perform motion planning using handles, or control points that are attached to points along cables, to install a wire harness [7]. First the constraints are relaxed by removing all handles along the cable and removing the gravity constraint. Once these two constraints are removed, the shape of the rod is determined by calculating its shape when it is in static equilibrium. After the shape of the harness in equilibrium is determined, this algorithm seeks to find a path to move it out of the environment without colliding with obstacles within the workspace. A potential function U assesses a cost to positions that are near the boundary of obstacles within the environment. Then, the A$*$ algorithm is used to determine a path to move the wire harness that maximizes the clearance for a ball (B_r) around the master handle that controls the junction of the wire harness.

Once the harness has been removed from the environment, cables in the harness are then unfolded or untangled. This is done by straightening each of the individual wires and ensuring they do not collide with the environment with the derived motion plan. Given that this procedure identifies a feasible path to remove the wire harness from the workspace without experiencing a collision, identifying a path to install the wire harness is trivial. Simply reversing the path of the wire harness yields a motion plan for installing the wire harness.

2.1.3 Method of Saha and Isto

In addition to modeling hoses and cables, Saha and Isto model a piece of rope as an elastic rod and perform experiments in which both ends of the rope are manipulated by robots [8]. This approach determines a sequence of motions that allows for moving a rope from an initial topology to a goal topology. One of the more common applications addressed by their work is tying knots, as with surgical suturing.

Elastic Rod Description

The description proposed by Saha and Isto requires a geometric model and a physical model. The rope is modeled as a curved cylinder of non-zero radius where the centerline of the cylinder is a smooth curve (c) of the form

$$c : s \in [0, L] \longrightarrow c(s) \in \mathbb{R}^3, \tag{2.9}$$

where s is the arc distance and L is the curvilinear length. The rope is discretized into a set of grasp points s_1, s_2, ... s_k which correspond to positions that can be manipulated by the robot. Saha and Isto utilize the physical model proposed by Wang et al. This procedure describes the rope as a group of massless springs that are connected by rigid links [9]. The position and orientation of each of the links is then determined by calculating the configuration that minimizes the energy stored in the springs. Their formulation ignores the dynamics of the system and only considers the rope when it is stable.

In addition to the geometric model and the physical model, Saha and Isto also present a topological model for describing the rope. The topological state of the rope requires the knowledge of the crossings. A crossing is simply a point of intersection when the shape of the rod is projected onto a plane. To determine the topology of a rope, "walk" is performed and a triplet of the form $(X_j^1, X_j^2, \epsilon_j)$ is generated for each intersection on the plane. For this representation, X_j^1 and X_j^2 indicate the crossing. The term ϵ_j is the sign of the crossing. The sign is determined by whether the crossing goes under or over a second portion of rope.

Manipulation Algorithm

The manipulation planning algorithm presented by Saha and Isto identifies a strategy to move control points along the rope so that the shape in transformed from the initial configuration (q_{init}) to the goal configuration (x_{goal}). Once the shape of the rod is determined and represented using the previously described topology, a probabilistic roadmap is constructed. The initial configuration, q_{init}, is treated as the root of the roadmap. Nodes with a bias towards x_{goal} are probabilistically sampled and are connected to the roadmap if there is a collision-free path to move between the two configurations. This algorithm then checks to determine if a crossing in the rope that indicates a new loop is formed. When a crossing is formed, a needle, or control point, is added to preserve the loop formed by this crossing. In addition to ensuring there is a feasible, collision-free path, paths must also be checked to ensure that proposed motions do not require the robot to achieve infeasible configurations. This process of probabilistically sampling nodes and checking to determine whether they can be connected to the roadmap continues until one of two conditions is met. Either a successful query is found to move from the initial configuration to the goal configuration or the maximum number of samples is reached and the algorithm exits and indicates that no viable paths have been found.

2.1.4 Methods of Hirose and Yamada

Yamada and Hirose have presented robots that perform locomotion on land and in water. The robots are composed of closed elastic rods that are connected to rotating actuators at both ends. These systems include robots that jump [10] as well as robots that maintain contact with the ground [11]. The crawling robots are inspired by snakes that are also capable of performing locomotion in water. In addition to snake robots, Yamada has shown swimming robots that use the elastic rod in a fashion similar to the fin of a fish [12].

Elastic Rod Description

The method presented by Yamada and Hirose does not seek to describe the entire shape of the metal strip, but does utilize elastic rod theory. Their work solely requires calculating the existence of an inflection point along the metal strip. The metal strip is discretized to n points where the static equilibrium equation at each point along the rod is described as

$$k\theta_i - {e_z}^T\{(p_n - p_{i-1}) \times \lambda_e + \tau_e\} = 0 \tag{2.10}$$

$$(i = 1....n), \tag{2.11}$$

where p_i is the position along the rod, p_n is the position at one end of the rod, $\lambda_e = [\lambda_1, \lambda_2, 0]^T$ is the force, and $\tau_e = [0, 0, \tau_3]^T$ is the joint torque. The existence of an inflection point along the elastic rod requires that

$${e_z}^T\{(p_n - p_c) \times \lambda_e + \tau_e\} = 0. \tag{2.12}$$

Manipulation Algorithm

The manipulation algorithm for these robotic systems determines a plan for rotating the actuators in a manner that induces “snap-through” buckling. Initially, the elastic rod is in the shape of an arch. By rotating one of the ends of the elastic rod, the rod is placed into an ‘S’ shape. Continuing to rotate this end forces the elastic rod to experience an instability and causes the shape of the rod to become an inverted ‘S’. Yamada et al. leverage this transition, or snap-through buckling, to provide locomotion. Essentially, this procedure aims to force the metal strip to have an inflection point. When there are no inflection points along the rod, $\lambda_2 = 0$. As one end of the rod is rotated, the rod gains one inflection point. Continued rotation produces a second inflection point along the metal strip when the rod is symmetric about the line crossing its center. The model predicts that snap-through buckling transpires approximately when the second inflection point appears. After this buckling completes, the rotation joint must be rotated in the opposite direction in order to induce another instance of snap-through buckling.

2.2 Manipulation by Devices Modeled as Elastic Rods

While in the previous section we discussed research that examined the manipulation of flexible objects, in this section we present work that focuses on how to use flexible devices to manipulate other objects. Both the modeling techniques and the motion planning algorithms described in the previous section can directly extend to performing manipulation with flexible devices. Though we do not conduct experiments in which the metal strip is used to manipulate other objects, our work can be directly applied to some of the applications described in this section.

2.2.1 Method of Tang et al.

The application of manipulating body tissues has created interest in the design and the motion planning algorithms for medical devices such as guidewires and catheters [13]. In addition to guidewires and catheters, flexible needles [14–17] and active cannula [18–22] have some advantages over rigid medical instruments. All of these devices provide the ability to "steer" around obstacles including sensitive tissue and vital organs. While rigid instruments can only manipulate tissues that have a straight-line-path to the insertion site, flexible medical instruments allow for manipulating tissue that is more difficult to access.

Elastic Rod Description

Tang et al. simulate elastic rods using a discretization approach that is based on previous work by Bergou et al. [23]. The guidewire is represented as a set of vertices x_0, x_1, ..., x_{n+1} and edges e_0, e_1, ..., e_n. The "dynamics" of the curve traced by the guidewire are described using the $\mathbb{R}^3$ vectors u_i^k

(control), x_i^k (position), v_i^k (velocity), and a_i^k (acceleration) for each point along the rod (i) at each time step (k). The position and velocity along the rod evolve with the discrete state equations where

$$x^{k+1} = x^k + \Delta t v^k + t^2((\frac{1}{2} - \beta)a^k + \beta a^{k+1}) \tag{2.13}$$

and

$$v^{k+1} = v^k + \Delta t((1-\gamma)a^k + \gamma a^{k+1}). \tag{2.14}$$

The parameters β and γ are both constants.

The acceleration is approximated using the relationship

$$F\left(x^k, v^k\right) + \frac{\delta F}{\delta x}\left(x^{k+1} - x^k\right) + \frac{\delta F}{\delta v}\left(v^{k+1} - v^k\right) \approx M a^{k+1}, \tag{2.15}$$

which is derived by applying one iteration of the Newton-Raphson method to the equation

$$F\left(x^{k+1}, v^{k+1}\right) = M a^{k+1}. \tag{2.16}$$

Once the acceleration is determined, the entire shape can be calculated by substituting the acceleration into Equation (2.13) and Equation (2.14).

The energy stored by an elastic rod must be a local minimum given the set of boundary conditions imposed by the position and orientation of both ends of the guidewire. The guidewire is capable of storing energy due to both bending and also twisting [23]. The total elastic energy for a configuration is defined as

$$E(\Gamma) = E_{bend}(\Gamma) + E_{twist}(\Gamma), \tag{2.17}$$

where the bending energy is defined as

$$E_{bend}(\Gamma) = \int \frac{1}{2}\alpha\kappa^2 ds \tag{2.18}$$

and the twisting energy is defined as

$$E_{twist}(\Gamma) = \int \frac{1}{2}\beta m^2 ds. \tag{2.19}$$

For these formulas, κ is the curvature (normal) vector of the centerline and m is the twist about the centerline. The constants α and β are based on

Young's modulus for the material. This is the ratio of the energy due to stress to the energy due to strain.

Manipulation Algorithm

The motion planning algorithm proposed by Tang et al. approximates a cubic spline of the centerline of a blood vessel and steers a guidewire in a manner that ensures the shape of the guidewire is approximately the shape of the centerline. The shape of the guidewire can be controlled by two motion primitives. The guidewire can only be controlled by inserting the rod and also by rotating the rod at the insertion point. As the guidewire is inserted, its shape is dictated by contacts with vascular walls. Given the boundary conditions that are imposed by contact between the guidewire and vascular walls, the guidewire forms a curve of minimal energy. Once the guidewire reaches a junction, additional actuation is required to steer into the desired blood vessel. A steering force is applied in a direction that is tangential to the desired vessel to explore and then insertion continues. As the guidewire is inserted into a vessel, the shape is continually simulated to compare the shape of the guidewire to the centerline of the vessel. This comparison is then used to determine the next action. In addition to the contrast in motion planning algorithms, the object representation differs from our approach. While we use a total of three parameters to describe all possible configurations of a metal strip, the method proposed by Tang et al. uses 12 parameters for each vertex. The representation of the guidewire is also of variable length where the accuracy comes at the expense of a larger representation. Furthermore, the algorithm does not produce global representation of feasible configurations of the guidewire.

2.2.2 Method of Hannan and Walker

As opposed to applications that perform manipulation with small objects, other research examines manipulation at a larger scale. Hannan and Walker propose experiments where they perform manipulation using an elephant trunk robot [24]. The robot they use to perform manipulation is similar to other variants of continuum robots [25–28]. These hyper-redundant robots are composed of multiple links that are connected by hinges that allow for rotating. The shape of these robots can be dictated by either springs, cables, or servo motors. Research that has examined continuum robots, such as the robot shown by Hannan and Walker, has provided another compelling application for our work. The relationship between these robots and flexible devices was pointed out by Tanner [29]. Additionally, the continuum robot designed by Gravagne [26] is composed of a spring steel spine. This is the material that we use to validate the theoretical framework proposed in this work.

Robot Description

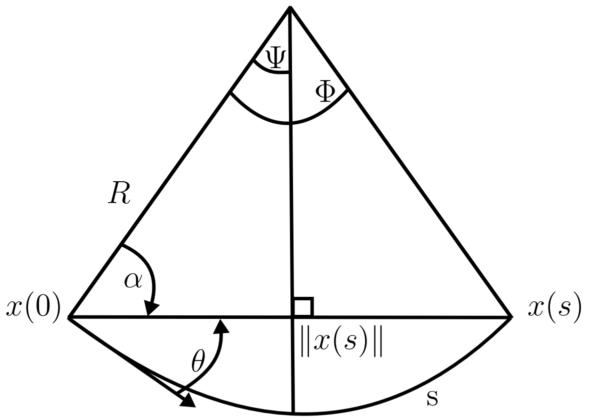


Figure 2.1: Sector which corresponds to one section of the hyper-redundant robot that Hannan and Walker use to perform manipulation experiments.

The approach taken by Hannan and Walker relies on describing each segment of the hyper-redundant robot using a constant curvature model and then using the Denavit-Hartenberg procedure to determine the kinematics of the entire robot. Each link is treated as a sector of a circle, as shown in Figure 2.1, with a chord that has points $x(0)$ and $x(s)$. The norm of this chord is $\|x(s)\|$ and the sector has the interior angle Φ. The angle Ψ is half of the interior angle. For this section of the robot the following relationships hold:

$$\Phi = \frac{s}{R} = \kappa s \tag{2.20}$$

$$\Psi = \frac{\Phi}{2} = \frac{\kappa s}{2}, \tag{2.21}$$

where R is the radius of the circle, s is the arc length, and κ is the curvature. This produces the result where

$$\|x\| = \frac{2}{\kappa} sin(\theta) = \frac{s}{\theta} sin(\theta). \tag{2.22}$$

Given the length of the chord, $\|x(s)\|$, Hannan and Walker use this expression to calculate θ, the tangent at the chord. This equation is then used to produce the transformation matrix which relates the position and angle of two adjoined links in the hyper-redundant robot.

Manipulation Algorithm

The manipulation algorithm implemented by Hannan and Walker utilize the direct relationship between the D-H kinematics analysis and the velocity kinematics. Planning is done by solving the general equation

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}} \tag{2.23}$$

where $\dot{\mathbf{x}}$ is the differentiation of the task space vector, J is the Jacobian, and $\dot{\mathbf{q}}$ is the differentiation of a vectoring consisting of the rotation angle and curvature of each section of the robot. Hannan and Walker show that the general solution to Equation (2.23) is

$$\dot{\mathbf{q}} = J(\mathbf{q})^{+}\dot{\mathbf{x}} + \{I - J(\mathbf{q})^{+}J(\mathbf{q})\}\epsilon. \tag{2.24}$$

The error, ϵ, is

$$\epsilon = k(\mathbf{q_r} - \mathbf{q}), \tag{2.25}$$

where k is a gain value, $\mathbf{q_r}$ is the desired configuration and $\mathbf{q}$ is the current configuration. Solving Equation (2.23) produces the solution in joint space to move towards the desired configuration.

2.2.3 Methods of Yamada and Mochiyama

Yamada and Mochiyama propose a method for manipulating devices that involve inducing instabilities or “snap-through” buckling. In addition to locomotion applications that were described in Section 2.1.4, they use planar metal strips to enable actuation. Instead of using the strip to move a robot between configurations, the metal strip is used to manipulate other objects within the environment. In addition to enabling swimming, the robotic swimmer presented by Yamada et al. also serves as a catapult system [10,30]. As with the robotic system, the catapult is composed of a metal strip that is connected to two rotating actuators. While the application greatly contrasts with the applications discussed in Section 2.1.4, the elastic rod description and manipulation algorithms are identical.

2.3 Manipulation Using a Cost Function

In this section, we will discuss the cost functions that have previously been applied to motion planning with flexible objects that are modeled as elastic rods. We then describe an algorithm for identifying the optimal motion plan given a particular cost function. While the goal of motion planning algorithms is to determine a feasible path to move from an initial configuration to a goal configuration, additional objectives can be taken into consideration in order to ensure “improved” manipulation. While common motion planning objectives include minimizing the amount of time or the amount of force required to complete a task, additional objectives can be used to ensure other assurances. First, we describe previous cost functions that have been applied to the problem of manipulating flexible objects and hyper-redundant robots to ensure that systems avoid self-collisions and joint limitations. Secondly, we discuss a cost function that is associated with the manipulability of the flexible object. This section concludes with a description of the motion planning algorithm we have implemented to identify paths that are optimal with respect to the cost functions we utilize.

2.3.1 Method of Khatib

Previous work conducted by Khatib [31] describes components of a cost function that directly apply to the task of installing a wire with a robot such as Baxter. This work presents a cost function in the form of a potential field that ensures collision-free manipulation. While preventing collisions is the primary concern, Khatib also presents a cost function, in the form of a potential field, that can be used to avoid configurations that approach the physical limitations of the robotic manipulator.

The first component of the cost function addresses avoiding configurations that experience self-collision. The potential field, U, is defined

$$U_{\text{collision}} = \frac{1}{2}\eta(\frac{1}{p} - \frac{1}{p_0})^2 \tag{2.26}$$

for $p < p_0$ where η is the scaling factor of this cost function, p is the shortest distance to a collision and p_0 is the limit distance or distance at which the cost function is 0. When the minimum distance to a collision is greater than or equal to p_0, the cost due to self-collisions is 0. When the minimum distance to a collision is less than p_0, the cost increases exponentially. When the rod experiences self-collision, $p = 0$, this component of the cost function is infinite.

An additional constraint to motion planning could be imposed by the joint limitations of the robotic manipulator. The cost component that Khatib et al. use to avoid joint limitations is the potential function

$$U_{\text{joint limitation}} = \eta(\frac{1}{\underline{p_i}} - \frac{1}{\underline{p_i(0)}})\frac{1}{\underline{p_i}^2}, \tag{2.27}$$

where the difference between the i^{th} joint angle q_i and the minimum possible value for that angle $\underline{q_i}$ is defined as

$$p_i = q_i - \underline{q_i}. \tag{2.28}$$

Again, the term η is used to scale this component of the cost function with respect to other components. When $q_i = \underline{q_i}$, or an angle is at its maximum value, the value of the cost function is infinite. The value of this component of the cost function decreases as the p_i increases.

While the cost functions derived by Khatib et al. are suited to working with rigid robots, this method does not directly extend to elastic rod functions. Mainly, determining the exact minimum distance to a self-collision can be computationally expensive. A variety of collision detection algorithms have been implemented to calculate the presence of a collision or proximity to a collision [32–35]. These techniques use bounding volume hierarchies to surround portions of deformable objects, such as elastic rods. Once the rod is discretized, these bounding volumes are checked for collisions or overlap in volumes. These volumes can also provide an estimate of the proximity to a collision.

2.3.2 Method of Yoshikawa

In addition to using potential fields to avoid configurations that experience self-collision and joint limitations, previous work by Yoshikawa has also described a cost function that is based on the manipulability of a flexible object or hyper-redundant robot [36]. This work is similar to the notion of the "shape Jacobian" that has been discussed by Mochiyama for kinematic chains [25]. The work by both Yoshikawa and Mochiyama points out that the Jacobian relationship between the shape of a kinematic chain and the position of the endpoints can be used to identify configurations that are sensitive to small changes in boundary conditions. The work by Yoshikawa is related to our work, whereas we also use a measure of the magnitude of the Jacobian matrix for performing motion planning.

Yoshikawa considers a hyper-redundant robot with joints that have angles θ_i, for i = 1,2,...,n and for the positions r_j for j=1,2,...,m. The positions of r are functions of the variable θ where

$$r = f(\theta) \tag{2.29}$$

for the joint vector

$$\theta = [\theta_1, \theta_2, ..., \theta_n,] \tag{2.30}$$

and the manipulation vector

$$r = [r_1, r_2, ..., r_m,] \,. \tag{2.31}$$

The Jacobian of the position with respect to the joint angle is calculated by differentiating Equation (2.29) with respect to time such that

$$\dot{r} = J(\theta)\dot{\Theta}. \tag{2.32}$$

Given the Jacobian for a set of joint angles, the manipulability (w) is calculated

$$w = \sqrt{det(J(\theta)J(\theta)^T)}. \tag{2.33}$$

Yoshikawa uses this measure of manipulability to avoid configurations where the robot experiences a singularity. This measure also allows for creating motion plans based on which joint has the most manipulability. When moving from an initial position vector, r, the manipulability for each of the joint angles θ_j is calculated. The joint angle with the largest manipulability is chosen for modification to move towards the goal configuration.

2.3.3 Motion Planning Algorithm

Once a cost function is derived, the task of identifying the path that minimizes the total cost over the course of a manipulation task must be identified. The optimal probabilistic roadmap (PRM$*$) [37–39] is the algorithm we use to identify an optimal path to move from an initial configuration to a goal configuration. This algorithm requires two phases. The first phase of the algorithm randomly samples states through the configuration space and calculates the cost to move between "nearby" configurations. A graph, or roadmap, that contains collision-free paths within the configuration space is maintained. This procedure can become computationally expensive, especially with configuration spaces of high dimension. After the configuration space has been sufficiently sampled, the second phase of the PRM$*$ proceeds. During this phase, a path is sought to move from the initial configuration to the goal configuration. A graph search is used to identify the sequence of states to move through that minimizes the total path cost. In section 4.3.1, we discuss our implementation of the PRM$*$ algorithm that is used to minimize a cost function that was derived to ensure robust manipulation of an elastic rod.

2.4 Application of Feedback to Manipulation

The integration of sensor information through the implementation of feedback is an additional tool that can be used to ensure robust manipulation of an elastic rod. In order to provide feedback, it is necessary to provide measurements of the rod's current state. It is also necessary to provide a control input based on the error message and control law that will move the rod towards the desired state. In this section we will describe existing methods for performing closed-loop manipulation with an elastic rod. This includes both the task of measuring the current state of the elastic rod and performing actuation in a manner to move the elastic rod to the goal configuration.

2.4.1 Measurement of Elastic Rod State

The most direct method for measuring the shape of an elastic rod is to use computer vision to extract the shape from captured images. This requires the implementation of image segmentation algorithms, such as that by Shi and Malik [40], to isolate the elastic rod from the background of the image. Once this is done, the shape can be measured. Borum et al. [41] also use computer vision to track the configuration of a metal strip that has been annotated with a small set of fiducial markers spaced along the length of the rod.

There are also some alternatives to direct measurements of the state of elastic rods through the use of computer vision. A common technique is to use force measurements, such as with whisker sensors [42–45]. These systems are comprised of thin elastic objects that extend directly from a force sensor. These sensors allow for locating and identifying nearby obstacles. The entire shape of the whisker can be determined by using force measurements at its base.

2.4.2 Application of Feedback

Once the state of the rod has been measured, it is necessary to determine the control input required to achieve the desired configuration. This involves determining the appropriate translation and rotation of an end point that transforms the shape of the rod to the goal shape. This procedure must take into consideration the entire shape of the rod and not just the position and orientation of the boundary conditions. In some elastic rod manipulation literature, this relationship is termed the "shape Jacobian" or simply Jacobian [25, 27]. The Jacobian provides the relationship between changes in the position and orientation at the end of the rod and the entire shape of the rod. This relationship provides a mechanism for transforming the shape from an initial configuration to a goal configuration.

In addition to using the shape Jacobian, the Jacobian of the force with respect to the position at the end of the rod can be used to apply feedback for closed-loop manipulation. The relationship between the force exerted at the end of an elastic rod and the position has been shown by Tang et al. [13]. In Chapter 3, we describe the relationship between the shape and the force and torque seen at the end of the rod. This relationship between force and positions along the rod provides an additional mechanism for applying feedback.

CHAPTER 3

MODEL

In this chapter we present a theoretical framework for manipulating a metal strip that is modeled as a planar elastic rod. First we show that all feasible shapes of the metal strip can be represented by a point in $\mathbb{R}^3$. A short version of this work was published in our IROS conference paper [46]. The coordinates used to represent the shape correspond to the force and torque at the base of the strip. After presenting the configuration space, we describe the free space and then describe an algorithm to move the strip from an initial configuration to a goal configuration. This chapter concludes with the presentation of two sets of experiments and a discussion of the results. We confirm the relationship between our coordinates and the force and torque at the end of the rod and validate the predicted boundary of the feasible configurations.

3.1 Theoretical Framework

We will see in Section 3.2 that the framed curve traced by a planar elastic rod in equilibrium is a local solution to a geometric optimal control problem. Here, we provide the framework to characterize this solution. Section 3.1.1 states necessary and sufficient conditions for optimality on manifolds. Section 3.1.2 derives coordinate formulae to test these conditions. We are citing work by Bretl and McCarthy [47].

In what follows, we denote the space of all smooth real-valued functions on a smooth manifold M by $C^\infty(M)$. We also recall that a smooth map $F\colon M \to N$ between smooth manifolds M and N is degenerate at $m \in M$ if the Jacobian matrix of any coordinate representation of F at m has zero determinant. We use T_mF and T_m^*F to denote the pushforward and pullback of F, respectively. The rest of our notation is standard [48, 49].

3.1.1 Optimal Control on Manifolds

Let M be a smooth manifold and let $U \subset \mathbb{R}^m$ for some $m > 0$. Assume $g\colon M \times U \to \mathbb{R}$ and $f\colon M \times U \to TM$ are smooth maps. Consider the optimal control problem

$$\begin{aligned} &\underset{q,u}{\text{minimize}} && \int_0^1 g\left(q(t), u(t)\right) dt \\ &\text{subject to} && \dot{q}(t) = f\left(q(t), u(t)\right) \text{ for all } t \in [0,1] \\ & && q(0) = q_0, \qquad q(1) = q_1, \end{aligned} \tag{3.1}$$

where $q_0, q_1 \in M$ and $(q,u)\colon [0,1] \to M \times U$. Define the parameterized Hamiltonian $\widehat{H}\colon T^*M \times \mathbb{R} \times U \to \mathbb{R}$ by

$$\widehat{H}(p,q,k,u) = \langle p, f(q,u)\rangle - kg(q,u),$$

where $p \in T^\star_q M$.

Theorem 1 (Necessary Conditions)**.** *Suppose*

$$(q_{opt}, u_{opt})\colon [0,1] \to M \times U$$

is a local optimum of (3.1)*. Then, there exists $k \geq 0$ and an integral curve $(p,q)\colon [0,1] \to T^*M$ of the time-varying Hamiltonian vector field X_H, where $H\colon T^*M \times \mathbb{R} \to \mathbb{R}$ is given by*

$$H(p,q,t) = \widehat{H}(p,q,k,u_{opt}(t)),$$

that satisfies $q(t) = q_{opt}(t)$ and

$$H(p(t),q(t),t) = \max_{u \in U} \widehat{H}(p(t),q(t),k,u) \tag{3.2}$$

for all $t \in [0,1]$. If $k = 0$, then $p(t) \neq 0$ for all $t \in [0,1]$.

Proof. See Theorem 12.10 of [50]. □

The integral curve (p,q) in Theorem 1 is an *abnormal extremal* when $k = 0$ and a *normal extremal* otherwise. As usual, when $k \neq 0$ we may assume $k = 1$. We call (q,u) abnormal if it is the projection of an abnormal extremal.

We call (q,u) normal if it is the projection of a normal extremal and it is not abnormal.

Theorem 2 (Sufficient Conditions)**.** *Suppose*

$$(p,q)\colon [0,1] \to T^*M$$

is a normal extremal of (3.1)*. Define $H \in C^\infty(T^*M)$ by*

$$H(p,q) = \max_{u \in U} \widehat{H}(p,q,1,u), \tag{3.3}$$

assuming that the maximum exists and that $\partial^2 \widehat{H}/\partial u^2 < 0$. Define $u\colon [0,1] \to U$ so that $u(t)$ is the unique maximizer of (3.3) *at $(p(t),q(t))$. Assume that X_H is a complete vector field and that there exists no other integral curve (p',q') of X_H satisfying $q(t) = q'(t)$ for all $t \in [0,1]$. Let $\varphi\colon \mathbb{R} \times T^*M \to T^*M$ be the flow of X_H and define the endpoint map $\phi_t\colon T^*_{q(0)}M \to M$ by $\phi_t(w) = \pi \circ \varphi(t,w,q(0))$. Then, (q,u) is a local optimum of* (3.1) *if and only if there exists no $t \in (0,1]$ for which ϕ_t is degenerate at $p(0)$.*

Proof. See Theorem 21.8 of [50]. □

3.1.2 Lie-Poisson Reduction

Let G be a Lie group with identity element $e \in G$. Let $\mathfrak{g} = T_eG$ and $\mathfrak{g}^* = T_e^*G$. Denote the functional derivative of any $h \in C^\infty(\mathfrak{g}^*)$ at $\mu \in \mathfrak{g}^*$ by $\delta h/\delta\mu \in \mathfrak{g}$, as in [49].

Theorem 3 (Reduction of Necessary Conditions)**.** *Let*

$$H\colon T^*G \times [0,1] \to \mathbb{R}$$

be both smooth and left-invariant for all $t \in [0,1]$. Denote the restriction of H to $\mathfrak{g}^*$ by $h = H|_{\mathfrak{g}^* \times [0,1]}$. Given $p_0 \in T^*_{q_0}G$, let $\mu\colon [0,1] \to \mathfrak{g}^*$ be the solution of

$$\dot{\mu} = \mathrm{ad}^*_{\delta h/\delta\mu}(\mu) \tag{3.4}$$

with initial condition $\mu(0) = T_e^*L_{q_0}(p_0)$. The integral curve $(p,q)\colon [0,1] \to T^*G$ of the time-varying Hamiltonian vector field X_H with initial condition

$p(0) = p_0$ *satisfies*

$$p(t) = T^*_{q(t)} L_{q(t)^{-1}} (\mu(t))$$

for all $t \in [0, 1]$, *where* q *is the solution of*

$$\dot{q} = X_{\delta h/\delta \mu}(q)$$

with initial condition $q(0) = q_0$.

Proof. See Theorem 13.4.4 of [49]. □

It is convenient for us to introduce coordinates on $\mathfrak{g}$ and $\mathfrak{g}^*$. Let $\{X_1, \ldots, X_n\}$ be a basis for $\mathfrak{g}$ and let $\{P_1, \ldots, P_n\}$ be the dual basis for $\mathfrak{g}^*$. We write ζ_i to denote the ith component of $\zeta \in \mathfrak{g}$ with respect to this basis, and so forth. Define the structure constants $C^k_{ij} \in \mathbb{R}$ by

$$[X_i, X_j] = \sum_{k=1}^{n} C^k_{ij} X_k \tag{3.5}$$

for $i, j \in \{1, \ldots, n\}$.

In addition to these three theorems, we also build upon Theorem 3 of Bretl and McCarthy [47] which shows a reduction of sufficient conditions.

3.2 Application to a Planar Elastic Rod

The previous section derived coordinate formulae to compute necessary and sufficient conditions for a particular class of optimal control problems on manifolds. Here, we apply these results to a planar elastic rod. Section 3.2.1 recalls that the framed curve traced by the rod in static equilibrium is a local solution to a geometric optimal control problem [51,52]. Section 3.2.2 proves that the set of all trajectories that are normal with respect to this problem is a smooth manifold of dimension three that can be parameterized by a single chart (Theorem 5). Section 3.2.3 proves that the set of all normal trajectories that are also local optima is an open subset of this smooth manifold, and provides a computational test for membership in this subset (Theorem 6). These two results suffice to describe all possible configurations of the elastic rod that can be achieved by quasi-static manipulation.

3.2.1 Model

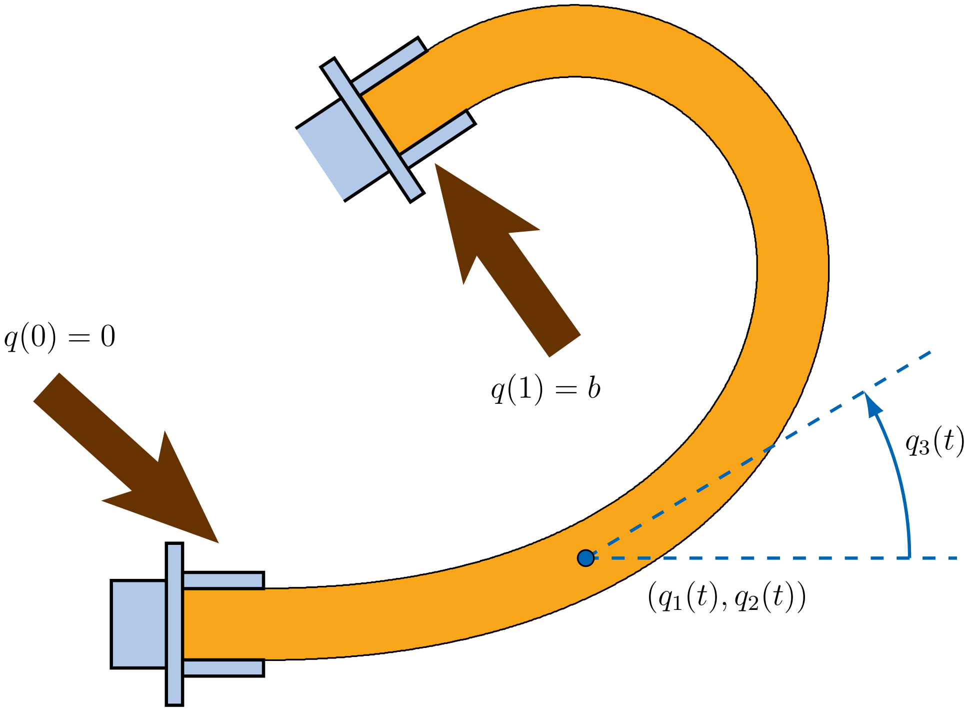


Figure 3.1: Schematic of an elastic rod. This graphic is annotated with labels indicating the start of the elastic rod where $q(0) = 0$ and the end of the rod where $q(1) = b$.

We model the object in Figure 1.3 as a planar elastic rod. Assuming it is thin, inextensible, and unit length, we describe the shape of this rod by a continuous map $q\colon [0,1] \to G$, where $G = SE(2)$. A schematic of this model is shown in Figure 3.1. We require this map to satisfy

$$\dot{q} = q(X_1 + uX_3) \tag{3.6}$$

for some $u\colon [0,1] \to U$, where $U = \mathbb{R}$ and

$$X_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad X_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

is a basis for $\mathfrak{g}$. Denote the dual basis for $\mathfrak{g}^*$ by $\{P_1, \ldots, P_3\}$. We refer to q and u together as $(q,u)\colon [0,1] \to G \times U$ or simply as (q,u). We assume the base of the rod is held fixed at the origin, so that $q(0) = e$. The other end is held by a robotic gripper, which we assume can impose arbitrary $q(1)$. Figure 3.1 shows a schematic of the model where both grippers are labeled. We denote the space of all $q(1)$ by $\mathcal{B} = G$. For fixed $q(1)$, the rod will remain motionless only if its shape locally minimizes total elastic energy. In particular, we say that (q, u) is in static equilibrium if it is a local optimum of

$$\begin{aligned} \underset{q,u}{\text{minimize}} \quad & \frac{1}{2}\int_0^1 u^2 dt \\ \text{subject to} \quad & \dot{q} = q(X_1 + uX_3) \\ & q(0) = e, \qquad q(1) = b \end{aligned} \tag{3.7}$$

for some $b \in \mathcal{B}$.

3.2.2 Necessary Conditions for Static Equilibrium

Theorem 4. *Define*

$$\mathcal{A} = \left\{a \in \mathbb{R}^3 \colon (a_2, a_3) \neq (0,0)\right\}.$$

A trajectory (q, u) is normal with respect to (3.7) *if and only if there exists $\mu\colon [0,1] \to \mathfrak{g}^*$ that satisfies*

$$\dot{\mu}_1 = \mu_2 u \qquad \dot{\mu}_2 = -\mu_1 u \qquad \dot{\mu}_3 = -\mu_2 \tag{3.8}$$

$$\dot{q} = q(X_1 + uX_3) \tag{3.9}$$

$$u = \mu_3 \tag{3.10}$$

with $q(0) = e$ and $\mu(0) = \sum_{i=1}^3 a_i P_i$ for $a \in \mathcal{A}$.

Proof. We begin by showing that (q, u) is abnormal if and only if $u = 0$. Theorem 1 tells us it is equivalent that (q, u) is the projection of an integral curve (p, q) of X_H that satisfies (3.2), where $H(p, q, t) = \widehat{H}(p, q, 0, u(t))$ and

$$\widehat{H}(p, q, 0, u) = \langle p, q(X_1 + uX_3)\rangle.$$

Since H is left-invariant, the existence of (p, q) satisfying Theorem 1 is equivalent to the existence of μ satisfying the conditions of Theorem 3, namely that

$$\dot{\mu} = \mathrm{ad}^*_{\delta h/\delta\mu}(\mu) \qquad \text{and} \qquad \dot{q} = q(\delta h/\delta\mu),$$

where $h = H|_{\mathfrak{g}^*}$. Application of the sufficient conditions produces (3.8)-(3.9), where we require $\mu_3 = 0$ to satisfy (3.2). We therefore have $\mu_2 = -\dot{\mu}_3 = 0$, hence also $\mu_1 u = -\dot{\mu}_2 = 0$. Since μ cannot vanish when $k = 0$, we must have $\mu_1 \neq 0$, hence $u = 0$.

Now, we return to the normal case. Theorem 1 tells us that (q, u) is normal if and only if it is not abnormal and it is the projection of an integral curve (p, q) of X_H that satisfies (3.2), where $H(p, q, t) = \widehat{H}(p, q, 1, u(t))$ and

$$\widehat{H}(p, q, 1, u) = \langle p, q(X_1 + uX_3)\rangle - (u^2/2).$$

As before, H is left-invariant. Application of the sufficient conditions to the conditions of Theorem 3 produces the same formulae (3.8)-(3.9), where (3.10) follows from (3.2) because $\widehat{H}$ is quadratic in u. It remains to show that trajectories produced by (3.8)-(3.10) are not abnormal if and only if $a \in \mathcal{A}$. We prove the converse. First, assume $a \in \mathbb{R}^3 \backslash \mathcal{A}$, so $(a_2, a_3) = (0, 0)$. From (3.8) and (3.10), we have $u = 0$, hence (q, u) is abnormal. Now, assume (q, u) is abnormal, so $u = 0$. From (3.10), we therefore have $\mu_3 = 0$, and in particular $a_3 = 0$. Plugging this result into (3.8), we see that $\mu_2 = -\dot{\mu}_3 = 0$, hence also that $a_2 = 0$. So, $a \in \mathbb{R}^3 \backslash \mathcal{A}$. Our result follows. □

Theorem 4 provides a set of candidates for local optima of (3.7), which we now characterize. Denote the set of all smooth maps $(q, u)\colon [0, 1] \to G \times U$ under the smooth topology by $C^\infty([0, 1], G \times U)$. Let $\mathcal{C} \subset C^\infty([0, 1], G \times U)$ be the subset of all (q, u) that satisfy Theorem 4. Any such $(q, u) \in \mathcal{C}$ is completely defined by the choice of $a \in \mathcal{A}$, as is the corresponding μ. Denote the resulting maps by $\Psi(a) = (q, u)$ and $\Gamma(a) = \mu$. We require three lemmas before our main result (Theorem 6).

Lemma 1. *If $\Psi(a) = \Psi(a')$ for $a, a' \in \mathcal{A}$, then $a = a'$.*

Proof. Suppose $(q, u) = \Psi(a)$ and $\mu = \Gamma(a)$ for some $a \in \mathcal{A}$. It suffices to show that a is uniquely defined by u (and its derivatives, since u is clearly smooth). From (3.8) and (3.10), we have

$$a_2 = -\dot{\mu}_3(0) = -\dot{u}(0) \qquad\qquad a_3 = u(0). \tag{3.11}$$

We differentiate (3.8) to compute

$$\ddot{u}(0) = a_3 a_1 \qquad \dddot{u}(0) = a_2 \left(a_3^2 - a_1\right). \tag{3.12}$$

At least one of these two equations allows us to compute a_1 unless $(a_2, a_3) = (0, 0)$, which would violate our assumption that $a \in \mathcal{A}$. Our result follows. □

Lemma 2. *The map $\Psi \colon \mathcal{A} \to \mathcal{C}$ is a homeomorphism.*

Proof. The map Ψ is a bijection—it is well-defined and onto by construction, and is one-to-one by Lemma 1. Continuity of Ψ follows from Theorem 4. It remains only to show that $\Psi^{-1} \colon \mathcal{C} \to \mathcal{A}$ is continuous—this result is an immediate consequence of (3.11)-(3.12). □

Lemma 3. *If the topological n-manifold M has an atlas consisting of the single chart (M, α), then $N = \alpha(M)$ is a topological n-manifold with an atlas consisting of the single chart (N, id_N), where id_N is the identity map. Furthermore, both M and N are smooth n-manifolds and $\alpha \colon M \to N$ is a diffeomorphism.*

Proof. Since (M, α) is a chart, then N is an open subset of $\mathbb{R}^n$ and α is a bijection. Hence, our first result is immediate and our second result requires only that both α and α^{-1} are smooth maps. For every $p \in M$, the charts (M, α) and (N, id_N) satisfy $\alpha(p) \in N$, $\alpha(M) = N$, and $\mathrm{id}_N \circ \alpha \circ \alpha^{-1} = \mathrm{id}_N$, so α is a smooth map. For every $q \in N$, the charts (N, id_N) and (M, α) again satisfy $\alpha^{-1}(q) \in M$, $\alpha^{-1}(N) = M$, and $\alpha \circ \alpha^{-1} \circ \mathrm{id}_N = \mathrm{id}_N$, so α^{-1} is also a smooth map. Our result follows. □

Theorem 5. *$\mathcal{C}$ is a smooth 3-manifold with smooth structure determined by an atlas with the single chart $(\mathcal{C}, \Psi^{-1})$.*

Proof. Since $\Psi \colon \mathcal{A} \to \mathcal{C}$ is a homeomorphism by Lemma 2 and $\mathcal{A} \subset \mathbb{R}^3$ is open, then $(\mathcal{C}, \Psi^{-1})$ is a chart whose domain is $\mathcal{C}$. Our result follows from Lemma 3. □

3.2.3 Sufficient Conditions for Static Equilibrium

Theorem 6. *Let* $(q,u)=\Psi(a)$ *and* $\mu=\Gamma(a)$ *for some* $a\in\mathcal{A}$*. Define*

$$\mathbf{F}=\begin{bmatrix}0 & \mu_3 & \mu_2\\ -\mu_3 & 0 & -\mu_1\\ 0 & -1 & 0\end{bmatrix}\qquad\qquad \mathbf{G}=\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}$$

$$\mathbf{H}=\begin{bmatrix}0 & \mu_3 & 0\\ -\mu_3 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}.$$

Solve the (linear, time-varying) matrix differential equations

$$\dot{\mathbf{M}}=\mathbf{FM}\qquad\qquad \dot{\mathbf{J}}=\mathbf{GM}+\mathbf{HJ}\tag{3.13}$$

with initial conditions $\mathbf{M}(0)=I$ *and* $\mathbf{J}(0)=0$*. Then,* (q,u) *is a local optimum of* (3.7) *for* $b=q(1)$ *if and only if* $\det(\mathbf{J}(t))\neq 0$ *for all* $t\in(0,1]$.

Proof. As we have already seen, normal extremals of (3.7) are derived from the parameterized Hamiltonian function

$$\widehat{H}(p,q,1,u)=\langle p,q(X_1+uX_3)\rangle-(u^2/2).$$

This function satisfies $\partial^2\widehat{H}/\partial u^2=-1<0$ and admits a unique maximum at $u=\langle p,qX_3\rangle$. The maximized Hamiltonian function is

$$H(p,q)=\langle p,qX_1\rangle+\langle p,qX_3\rangle^2/2.$$

It is clear that X_H is complete. By Lemma 1, the mapping from (q,u) to a and hence to $\mu=\Gamma(a)$ is unique. By Theorem 3, it is equivalent that the mapping from (q,u) to (p,q) is unique. As a consequence, we may apply Theorem 2 to establish sufficient conditions for optimality. Since H is left-invariant, we may apply the equivalent conditions of Theorem 4 of [47]. Noting that $h=H|_{\mathfrak{g}^*}\in C^\infty(\mathfrak{g}^*)$ is given by

$$h(\mu)=\mu_1+\mu_3^2/2,$$

it is easy to verify that $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ take the given form. □

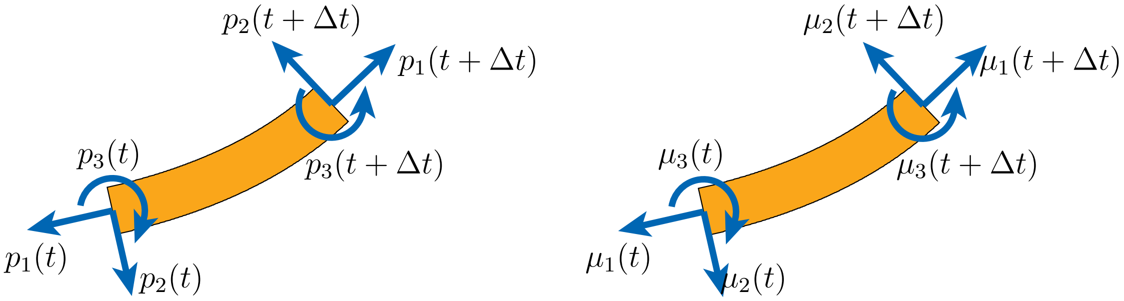


Figure 3.2: The forces and torques applied to a piece of the planar elastic rod that correspond to the balance equations in Equation (3.8). This representation provides a physical interpretation of the costate trajectory $\mu\colon [0,1] \to \mathfrak{g}^*$. Equilibrium configurations are uniquely defined by the choice of $a = \mu(0)$.

Theorem 6 provides a computational test of which points $a \in \mathcal{A}$ actually produce local optima $\Psi(a) \in \mathcal{C}$ of (3.7). Let $\mathcal{A}_{\text{stable}} \subset \mathcal{A}$ be the subset of all a for which the conditions of Theorem 6 are satisfied and let $\mathcal{C}_{\text{stable}} = \Psi(\mathcal{A}_{\text{stable}}) \subset \mathcal{C}$. An important consequence of membership in $\mathcal{A}_{\text{stable}}$ is smooth local dependence of (3.7) on variation in b. Define

$$\mathcal{B}_{\text{stable}} = \{q(1) \in \mathcal{B}\colon (q,u) \in \mathcal{C}_{\text{stable}}\}$$

and let $\Phi\colon \mathcal{C} \to \mathcal{B}$ be the map taking (q,u) to $q(1)$. Clearly $\mathcal{A}_{\text{stable}}$ is open, so

$$\Psi|_{\mathcal{A}_{\text{stable}}}\colon \mathcal{A}_{\text{stable}} \to \mathcal{C}_{\text{stable}}$$

is a diffeomorphism. We arrive at the following result:

Theorem 7. *The map $\Phi \circ \Psi|_{\mathcal{A}_{stable}}\colon \mathcal{A}_{stable} \to \mathcal{B}_{stable}$ is a local diffeomorphism.*

Proof. The map $\Phi \circ \Psi|_{\mathcal{A}_{\text{stable}}}$ is smooth and by Theorem 6 has non-singular Jacobian $\mathbf{J}(1)$. Our result follows from the Implicit Function Theorem [53, Theorem 7.9]. □

3.2.4 Modeled Force at the Base of the Rod $x(0)$

The coordinate chart $\mathcal{A}$ has a physical interpretation. To derive it, we will assume that $\mu(t)$ describes the force and torque acting on the rod at $t \in [0,1]$, and we will show that this assumption allows us to reconstruct (3.8) and (3.10). Consider a small piece of the rod (Figure 3.2). Choose (v_1, v_2, θ) so that:

$$\begin{bmatrix} \cos\theta & -\sin\theta & v_1 \\ \sin\theta & \cos\theta & v_2 \\ 0 & 0 & 1 \end{bmatrix} = q(t)^{-1} q(t+\Delta t).$$

In static equilibrium, a force and torque balance requires that:

$$\begin{aligned} 0 &= -\mu_1(t) + \mu_1(t+\Delta t)\cos\theta - \mu_2(t+\Delta t)\sin\theta \\ 0 &= -\mu_2(t) + \mu_1(t+\Delta t)\sin\theta + \mu_2(t+\Delta t)\cos\theta \\ 0 &= -\mu_3(t) + \mu_3(t+\Delta t) + \mu_1(t+\Delta t)\left(v_1\sin\theta\right. \\ &\qquad \left.-v_2\cos\theta\right) + \mu_2(t+\Delta t)\left(v_1\cos\theta + v_2\sin\theta\right). \end{aligned}$$

In the limit as $\Delta t \to 0$, we recover (3.8). Equation (3.10) then follows from the linear relationship between stress and strain. It is now clear that $\mathcal{A}$ is a space of forces and torques, and in particular that $\mu(0) = a \in \mathcal{A}$ describes the force and torque at the base of a planar elastic rod. The reader may also verify that abnormal (q, u) are exactly those configurations of the rod at which $\mu(0)$ is indeterminate.

3.3 Manipulation Planning

In this section, we present an algorithm for manipulation planning with a metal strip. First, the configuration space is defined and we describe how to sample a point uniformly at random in this space. We then define the free space, which corresponds to configurations of the metal strip that are feasible, and describe how to test whether a sampled point in configuration space is also in free space. Given this description of the free space, we present an algorithm for identifying a path the robot should follow so that the metal strip moves from an initial configuration to a goal configuration.

3.3.1 Configuration Space Sampling

In order to implement a sampling-based planner, it is necessary to define a configuration space and to also describe the task of sampling a point at random in this space. For the application of manipulating a metal strip, the configuration space represents all of the feasible curves that can be formed.

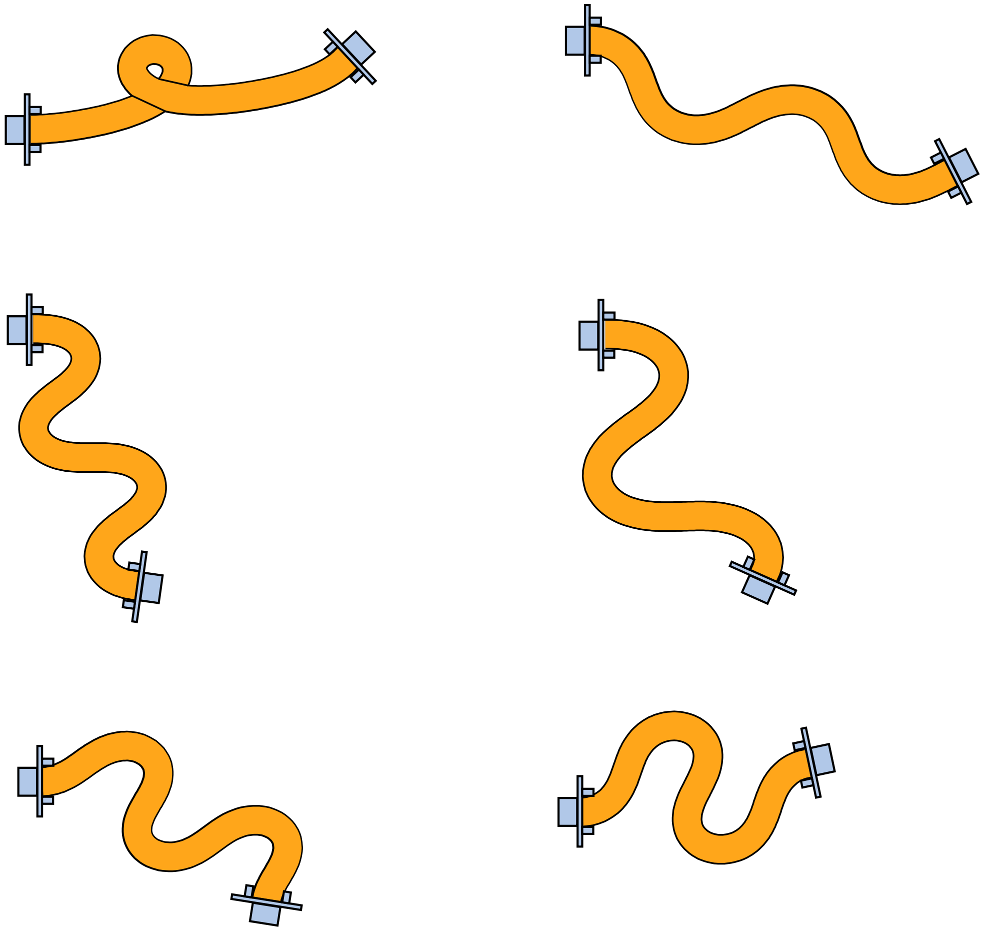


Figure 3.3: These images all show configurations that were generated by uniformly sampling the configurations space by finding a vector in $\mathbb{R}^3$ for each curve where $a_1 \in [-150, 150]$, $a_2 \in [-150, 150]$, and $a_3 \in [-20, 20]$.

In the previous two sections, we have shown that each configuration can be represented by a point in $\mathbb{R}^3$. Simply integrating Equations 6.1 for the initial costate values for a_1, a_2, and a_3 produces a unique minimal energy curve.

Sampling points in $\mathcal{A}$ refers to randomly sampling three rational numbers using a uniform distribution. Figure 3.3 shows six configurations that were generated using a random number generator to select three numbers with $a_1 \in [-150, 150]$, $a_2 \in [-150, 150]$, and $a_3 \in [-20, 20]$. All of the configurations in this image do not correspond with feasible shapes that can be realized using the robot gripper. For example, the curve in the top-left experiences a self-collision. It is necessary to ensure that sampled configurations are feasible, or $a \in \mathcal{A}_{\text{free}}$.

3.3.2 Definition of $\mathcal{A}_{\text{free}}$

While the configuration space, $\mathcal{A}$, allows for describing all configurations of the elastic rod using three coordinates, not all configurations can be achieved with an actual metal strip. In order to implement a sampling-based planner, it is necessary to describe obstacles within the configuration space. In this section we will describe the constraints on $\mathcal{A}_{\text{free}}$, feasible configurations of an elastic rod. With respect to sampling-based motion planning algorithms, these conditions constitute obstacles in the configuration space ($\mathcal{A}$) that must be avoided. The following sections will describe three types of constraints on $\mathcal{A}_{\text{free}}$ and show how these conditions can be included into our model to ensure that paths planned in $\mathcal{A}$ safely move from an initial configuration to a goal configuration.

Instabilities

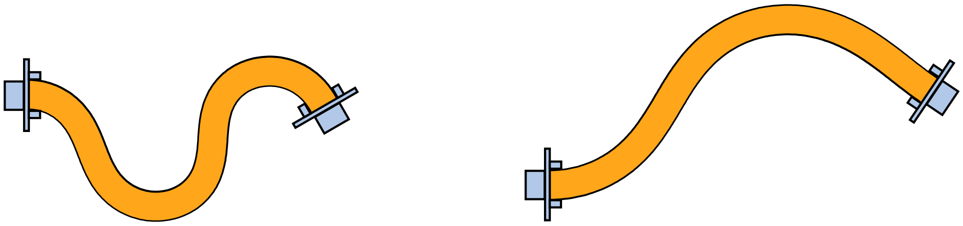


Figure 3.4: Schematics of what takes place before and after the rod experiences an instability. The image on the left shows the metal strip before it has experienced an instability. As a result of moving the robot gripper, the strip experiences an instability. The image on the right shows the shape after the gripper has moved and the rod has experienced an instability.

Instabilities are characterized by the deviation between the theoretical shape of an elastic rod from the physical shape. Some instabilities can result in the metal strip undergoing large deformations quickly as the strip moves from an unstable configuration to a stable configuration. Figure 3.4 shows the metal strip before and after it has experienced an instability. The placement of the robot has moved a small amount. However, this small perturbation results in a large deformation of the shape of the metal strip. As described in Section 3.2.3, the Jacobian of the system provides information about the stability of elastic rod configurations. This relationship provides an indication of the effect that small perturbations in the robot's placement have on the configuration in $\mathcal{A}$. We check points along the curve to determine whether the Jacobian vanishes for $t \in [0, 1]$. The presence of a singular Jacobian indicates that the configuration is unstable.

Collisions

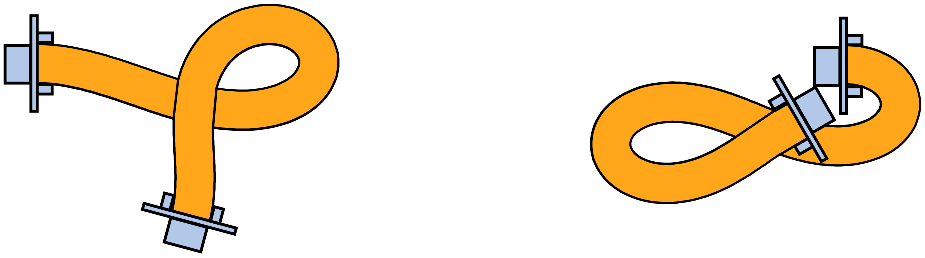


Figure 3.5: Two schematics in which the metal strip experiences a self-collision.

In addition to instabilities, there are two other constraints imposed on experiments as a result of using the Adept XL-One robot to manipulate the metal strip. The theoretical formulation allows for configurations that include self-collisions. Essentially, the rod is able to pierce itself without altering its shape. This does not happen with hardware experiments where collisions can result in permanent deformations of the metal strip. Figure 3.5 shows two configurations that experience self-collisions. In order to perform collision detection for the elastic rod application, it is necessary to check for collisions between

- the metal strip and itself
- the metal strip and end the gripper at table or on robot
- the gripper on the table and the robot gripper on the robot

Whether the metal strip experiences a self-collision can be determined using a variety of collision checking algorithms [33, 34, 54] . Planning paths that avoid configurations that cause self-collisions requires the addition of constraints to limit $\mathcal{A}_{\text{free}}$.

Robot Joint Limits

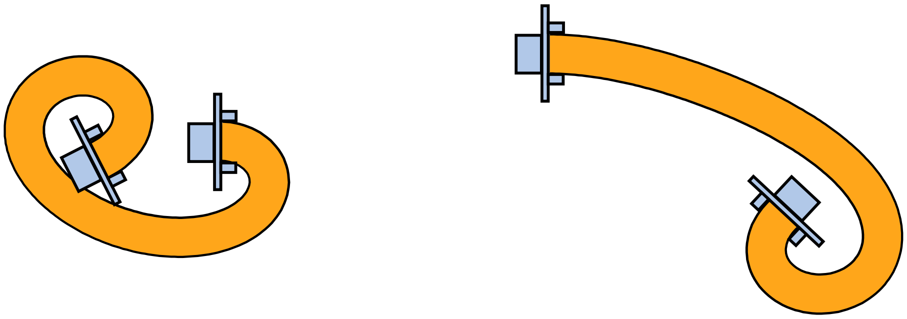


Figure 3.6: Two schematics that require the angle of the manipulator to exceed its rotation limit.

The Adept robot imposes a constraint due to its joint limitations. The joint controlling θ only provides a range of 540°. The Adept robot can only achieve configurations that fall within the range where $\theta \in [-\frac{3\pi}{2}, \frac{3\pi}{2}]$. To calculate the position and orientation, the differential equations from Equation (3.13) must be integrated using the values for $\mathbf{a}$. The orientation ($q_3(1)$) corresponds to the orientation of the robot and must be checked to ensure the value is within the limits of the Adept robot. Figure 3.6 shows two configurations that are infeasible to realize with the Adept robot due to the joint limitations.

Predicted Boundary of $\mathcal{A}_{\text{free}}$

By combining the previously described constraints, we can predict whether a configuration mapping to a point in $\mathcal{A}$ is feasible. Furthermore, we can construct a boundary of $\mathcal{A}_{\text{free}}$ such that all of the configurations inside the boundary are feasible. The flow chart in Figure 3.7 shows the algorithm to determine whether a point $a \in \mathcal{A}_{\text{free}}$. The input to the algorithm is a point in $\mathbb{R}^3$. This algorithm produces two outputs. First the algorithm produces a

Boolean value indicating whether the configuration is feasible. Second, the algorithm produces a $b \in \mathcal{B}$ which indicates the position and orientation to hold the robot gripper to achieve the minimal energy curve with an initial costate vector corresponding with the input value, a.

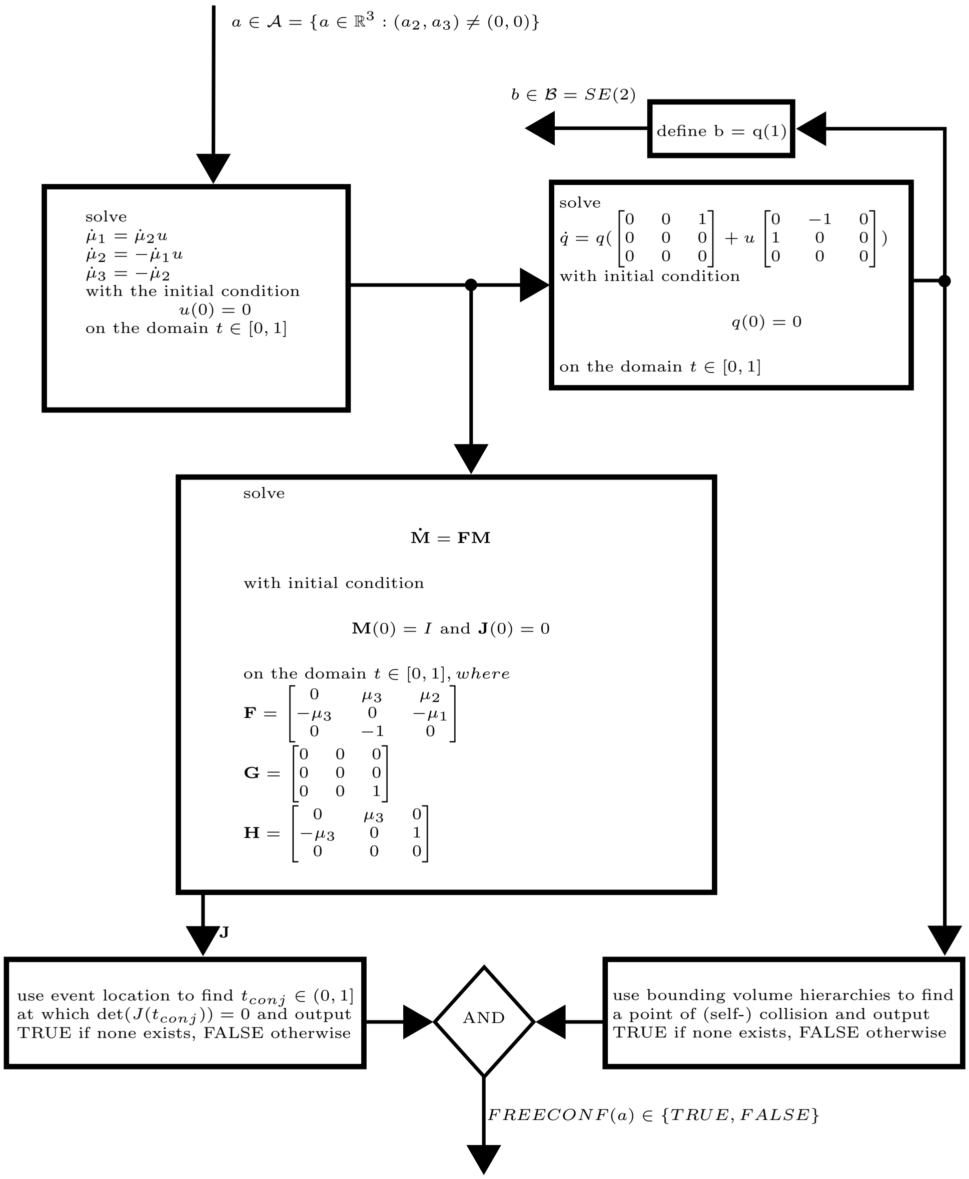

$a \in \mathcal{A} = \{a \in \mathbb{R}^3 : (a_2, a_3) \neq (0,0)\}$
$b \in \mathcal{B} = SE(2)$
define b = q(1)
solve
$\dot{\mu}_1 = \dot{\mu}_2 u$
$\dot{\mu}_2 = -\dot{\mu}_1 u$
$\dot{\mu}_3 = -\dot{\mu}_2$
with the initial condition
$u(0) = 0$
on the domain $t \in [0,1]$
solve
$\dot{q} = q(\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + u \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix})$
with initial condition
$q(0) = 0$
on the domain $t \in [0,1]$
solve
$\dot{\mathbf{M}} = \mathbf{F}\mathbf{M}$
with initial condition
$\mathbf{M}(0) = I$ and $\mathbf{J}(0) = 0$
on the domain $t \in [0,1]$, *where*
$\mathbf{F} = \begin{bmatrix} 0 & \mu_3 & \mu_2 \\ -\mu_3 & 0 & -\mu_1 \\ 0 & -1 & 0 \end{bmatrix}$
$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\mathbf{H} = \begin{bmatrix} 0 & \mu_3 & 0 \\ -\mu_3 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
$\mathbf{J}$
use event location to find $t_{conj} \in (0,1]$ at which $\det(J(t_{conj})) = 0$ and output TRUE if none exists, FALSE otherwise
AND
use bounding volume hierarchies to find a point of (self-) collision and output TRUE if none exists, FALSE otherwise
$FREECONF(a) \in \{TRUE, FALSE\}$

Figure 3.7: This flow chart shows the algorithm required to ensure that a configuration, indicated by a coordinate $a \in \mathcal{A} \subset \mathbb{R}^3$, is a feasible configuration. This algorithm also indicates the position to place a gripper at the end of the rod.

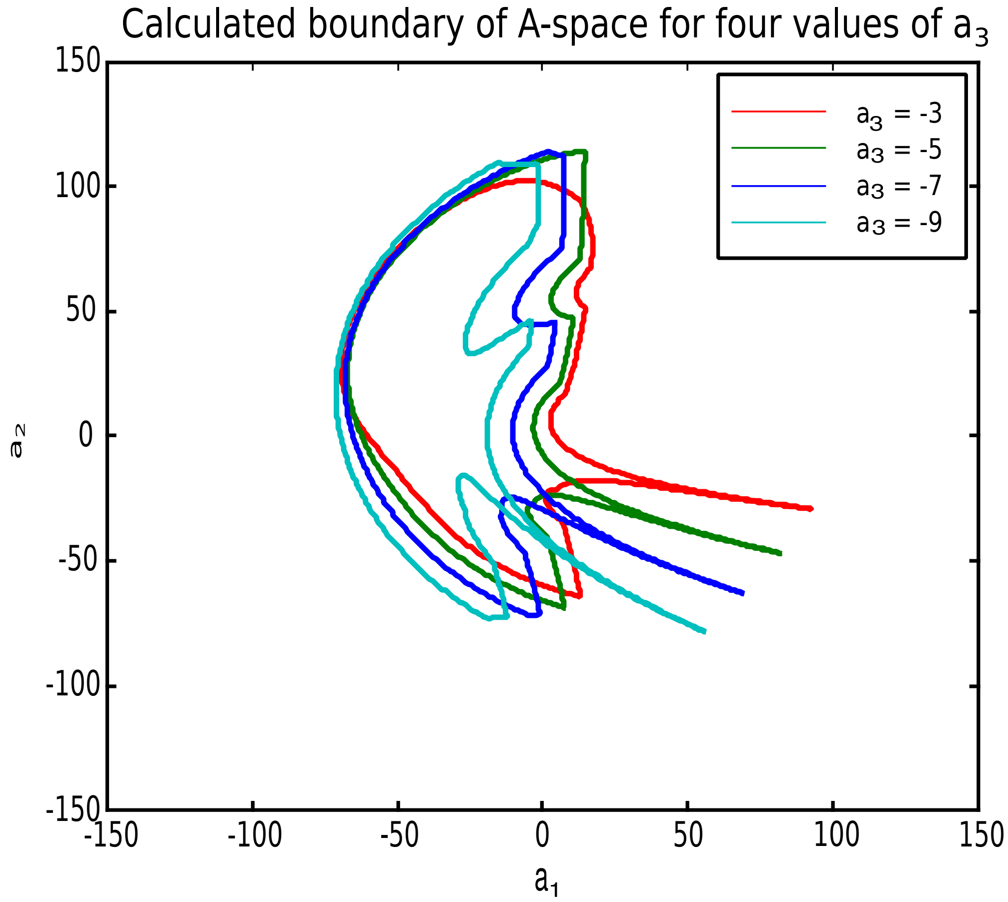


Figure 3.8: These plots show four individual boundaries or slices for $\mathcal{A}_{\text{free}}$. For each slice, the value a_3 is fixed. A continuation method is used to determine the boundary for the a_1 and a_2 coordinates.

Predicted boundary of $\mathcal{A}_{\text{free}}$

Though the elastic rod has an infinite number of configurations, the 3-dimensional boundary can be calculated, even with the additional constraints due to the limitations of the robotic manipulator. For each value of a_3, random values where $a_1 \in [-125, 125]$ and $a_2 \in [-125, 125]$ are selected. Each random point is tested to determine whether it is a feasible configuration in $\mathcal{A}_{\text{free}}$. When a feasible configuration is identified, points along a line that passes through this point are tested for feasibility until a point on the boundary of $\mathcal{A}_{\text{free}}$ is detected. Once a point on the boundary is identified, a continuation method identifies the remaining points on the boundary for the selected value of a_3. Figure 3.8 shows four individual slices of a_3.

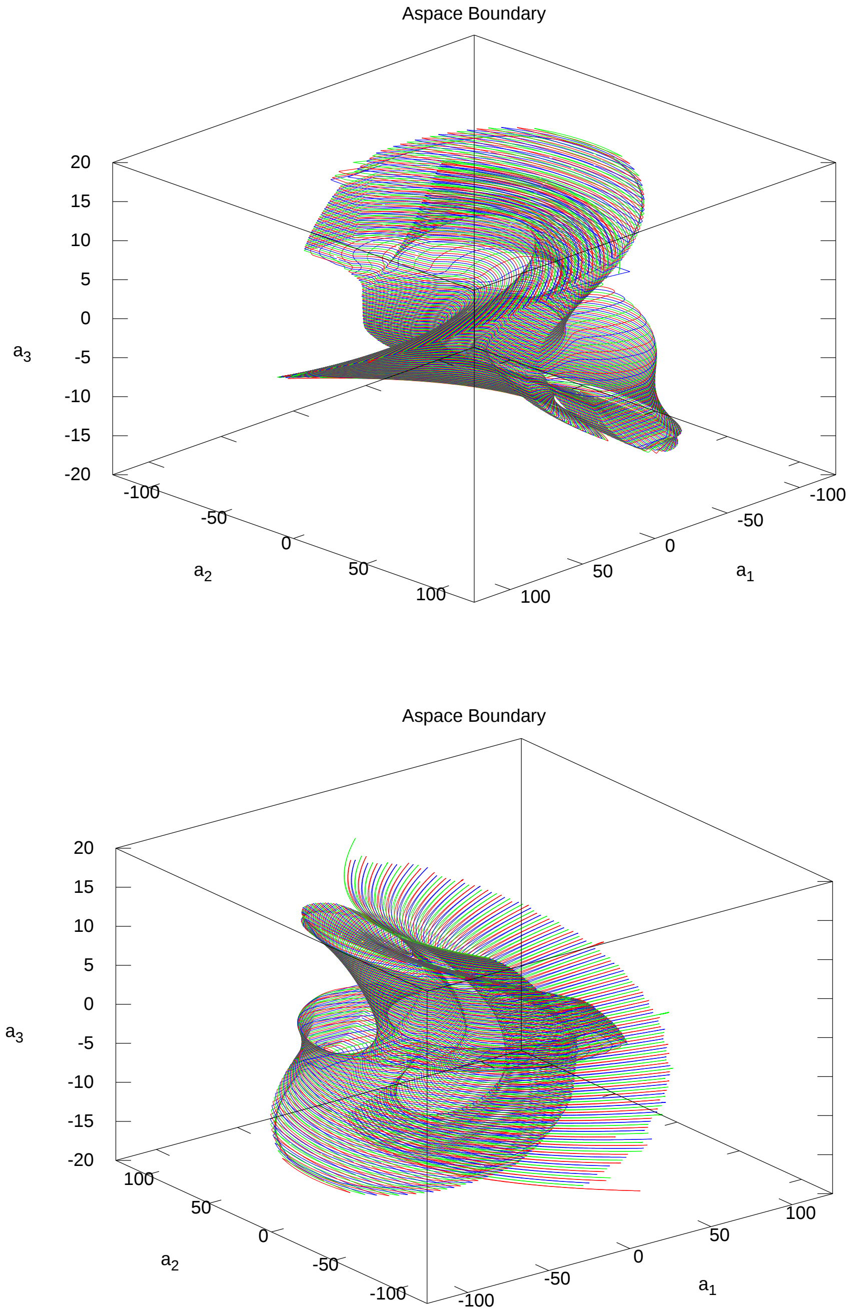


Figure 3.9: Calculation of the entire 3-dimension volume of $\mathcal{A}_{\text{free}}$. This region contained within the volume represents the coordinates of all of the feasible configurations for the elastic rod. This volume is constructed using the continuation method with a_3 increments of 0.2.

Figure 3.9 shows the calculation of the entire boundary of $\mathcal{A}_{\text{free}}$. This was calculated by starting with $a_3 = -16.0$, the minimum a_3 with feasible configurations. We then calculate the boundary for each value of a_3 up to the maximum value with feasible configurations at 0.2 intervals. While most slices consist of just one connected component, for values where $-14 < a_3 < -9.8$ and $9.8 < a_3 < 14$ there are two connected components. For these values of a_3 with multiple connected components, the continuation method is applied twice. In order to correctly identify both boundaries, the method is initialized with separate starting points that are within both of the components.

3.3.3 Motion Planning Algorithm

With a concise model and coordinate system for describing feasible configurations of the metal strip (Section 3.2) and a representation within this coordinate system of the constraints on $\mathcal{A}_{\text{free}}$ (Section 3.3.2), we can now perform motion planning to move the strip from an initial configuration to a goal configuration. This section describes the algorithm used to move the metal strip from an initial configuration to a goal configuration by following a path planned in the new configuration space, $\mathcal{A}$.

We now know that any equilibrium configuration of a planar elastic rod can be represented by a point in $\mathcal{A}_{\text{stable}} \subset \mathcal{A} \subset \mathbb{R}^3$ (Theorems 4-6) and that any path of the rod in $\mathcal{A}_{\text{stable}}$ can be realized by a path of the robotic gripper in $\mathcal{B}_{\text{stable}}$ (Theorem 7). These results allow us to apply a sampling-based algorithm for manipulation planning (here, we describe one based on PRM [37]):

- Sample points in $\mathcal{A}$, for example uniformly at random in $\{a \in \mathcal{A}\colon\ \|a\|_\infty \leq w\}$ for some $w > 0$. Note that it is possible to choose w by taking advantage of the correspondence between a and forces/torques at the base of the elastic rod (Section 3.2.4).

- Keep points that are in $\mathcal{A}_{\text{stable}}$ and add them as nodes in the roadmap. This test requires only solving the ordinary differential equations (3.8)-(3.10) in 3 variables and the matrix differential equations (3.13) in 18 variables.

- Try to connect each pair of nodes a and a' with a straight-line path in $\mathcal{A}$, adding this path as an edge in the roadmap if it lies entirely in $\mathcal{A}_{\text{stable}}$. This test can be approximated in the usual way by sampling points along the straight-line path at some resolution, again solving (3.8)-(3.10) and (3.13) for each point.

- Declare $a_{\text{start}}, a_{\text{goal}} \in \mathcal{A}_{\text{stable}}$ to be path-connected if they are connected by a sequence of nodes and edges in the roadmap. This sequence is a continuous and piecewise-smooth map

  $$\alpha \colon [0,1] \to \mathcal{A}_{\text{stable}},$$

  where $\alpha(0) = a_{\text{start}}$ and $\alpha(1) = a_{\text{goal}}$.

- Move the robotic gripper along the path

  $$\Phi \circ \Psi|_{\mathcal{A}_{\text{stable}}} \circ \alpha \colon [0,1] \to \mathcal{B}_{\text{stable}}.$$

  This path is again continuous and piecewise-smooth, and can be evaluated at waypoints $s \in [0,1]$ by solving the matrix differential equation (3.9) on $SE(2)$.

Each step is trivial to implement using modern numerical methods. It is also easy to include other constraints within this basic framework. For the experiments that we describe in the following section, we check for self-collision (using hierarchical bounding volumes) and enforce bounds on position and orientation of the robotic gripper.

The “initial” and “goal” for the manipulation planning problem must be points in $\mathcal{A}_{\text{stable}}$, or equivalently points in $\mathcal{C}_{\text{stable}}$ through the diffeomorphism Ψ. It is insufficient to specify start and goal by points in $\mathcal{B}_{\text{stable}}$ since these points do not uniquely define configurations of the rod.

Figure 3.10 shows a flow chart of the PRM algorithm implemented to move between two configurations. This algorithm takes the **a** values of the initial configuration and goal configuration as inputs. In addition, the algorithm requires stop criteria that determines when the PRM algorithm terminates execution if a viable path between the two points in $\mathcal{A}$ is not identified. If a feasible path is determined before the stop criteria are met, the algorithm terminates and returns true. When the PRM completes successfully, this algorithm also produces a sequence of points in $\mathcal{A}$ to move from the initial

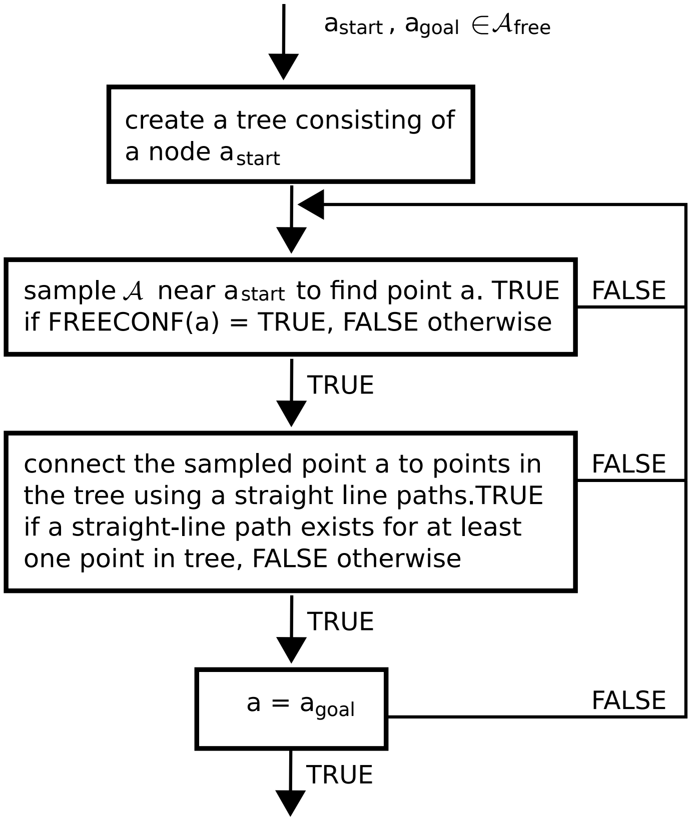

a_{start} , a_{goal} $\in \mathcal{A}_{free}$
create a tree consisting of a node a_{start}
sample $\mathcal{A}$ near a_{start} to find point a. TRUE if FREECONF(a) = TRUE, FALSE otherwise
FALSE
TRUE
connect the sampled point a to points in the tree using a straight line paths.TRUE if a straight-line path exists for at least one point in tree, FALSE otherwise
FALSE
TRUE
a = a_{goal}
FALSE
TRUE

Figure 3.10: Motion planning algorithm that moves from an initial configuration a_{start} to a goal configuration a_{goal}.

configuration to the goal configuration.

Figure 3.11 shows an example of quasi-static manipulation that was planned by our sampling-based algorithm. Notice that the start and goal configurations are both associated with the same boundary conditions, each one being a different local minimum of total elastic energy, i.e., a different local optima $a_{\text{start}}, a_{\text{goal}} \in \mathcal{A}_{\text{stable}}$ of (3.7) for the same choice of $b \in \mathcal{B}_{\text{stable}}$. The motion shown in Figure 3.11 therefore does not correspond to a single straight-line path in $\mathcal{B}_{\text{stable}}$ where planning has traditionally been done (e.g., [4,5]). However, this motion does indeed correspond to a single straight-line path in $\mathcal{A}_{\text{stable}}$ and was trivial to generate with our planning algorithm. We have not yet performed comprehensive experiments that compare our sampling-based algorithm to others in terms of running time, failure probability, etc.—these experiments are a topic of ongoing work. However, a number of planning heuristics like lazy collision-checking [55]—which bring huge speed-ups in practice—are easy to apply when planning in $\mathcal{A}_{\text{stable}}$ but hard to apply when planning in $\mathcal{B}_{\text{stable}}$. Also, should we still want to plan in $\mathcal{B}_{\text{stable}}$ (i.e., to connect nearby configurations by straight-line paths in $\mathcal{B}_{\text{stable}}$ rather than in $\mathcal{A}_{\text{stable}}$), it is now easy to do so by using the Jacobian matrix $\mathbf{J}(1)$, which is non-singular in $\mathcal{B}_{\text{stable}}$ by construction. In particular, there is the relationship $\delta b = \mathbf{J}(1)\delta a$, which can be inverted to move along straight lines in $\mathcal{B}_{\text{stable}}$. Without this relationship, it would be necessary to apply gradient descent in the infinite-dimensional space of inputs $u\colon [0,1] \to U$, prompting methods of approximation like the one described in [5]. In any case, the key insight was realizing that the set of equilibrium configurations has dimension three—at that point, nearly any planning algorithm will perform well.

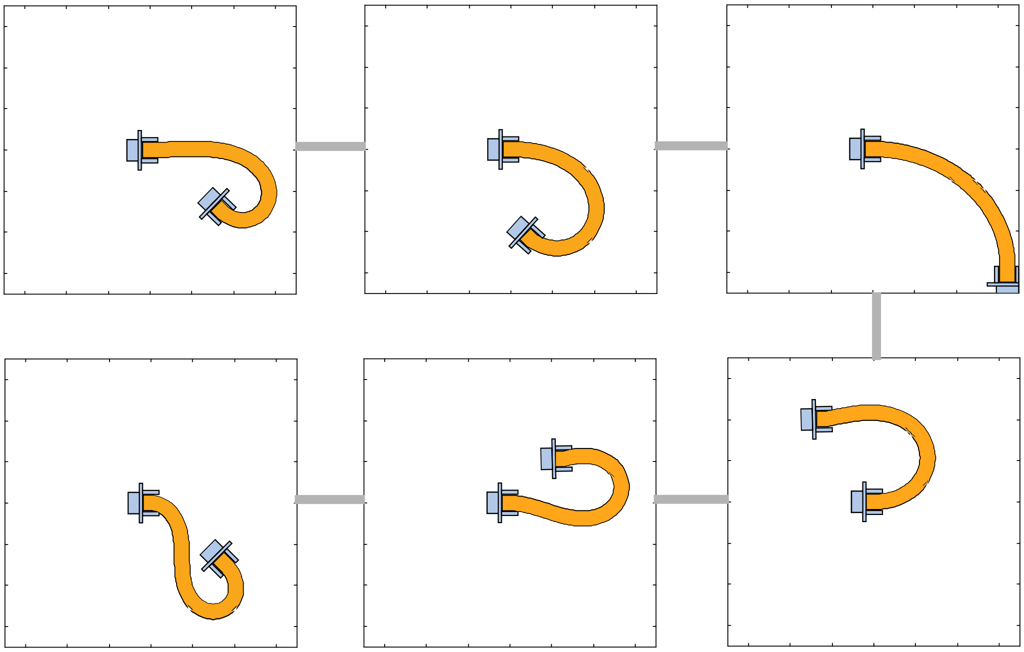


Figure 3.11: Sequence of frames depicting configurations of the metal strip while moving from an initial configuration to a goal configuration. Notice that the initial and goal configurations have identical boundary conditions.

3.4 Experimental Validation

In this section, we will present the design of experiments used to compare the theoretical model to physical devices. Without loss of generality, we are able to hold one end of the strip at a fixed location and manipulate the other end with a robotic manipulator that is capable of translating in the x-y plane and rotating about the z-axis. This framework is used to perform qualitative and quantitative comparisons between the modeled configuration and the measured configuration.

3.4.1 Hardware

Metal Strip

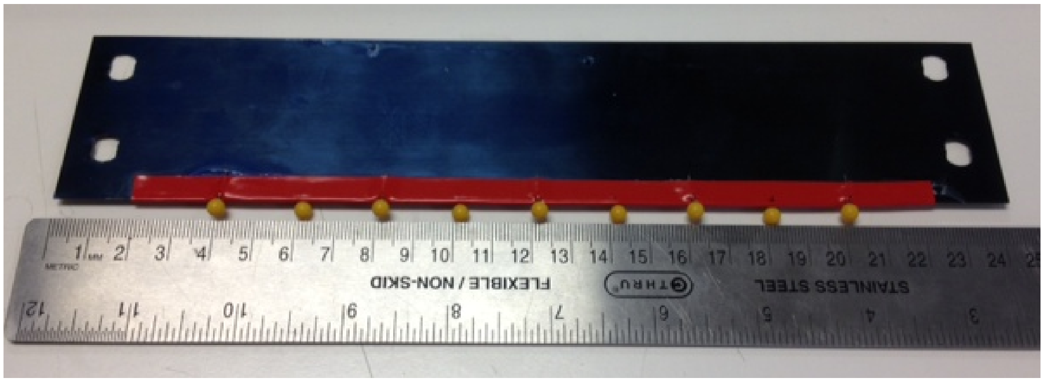


Figure 3.12: Strip of spring steel that is used for experiments. The strip has 2 holes on each side that are used for attaching to the Adept robot and the experiment table. This strip has red tape affixed to the bottom that allows for image segmentation. A ruler is placed blow the metal strip to provide the scale.

All of the experiments were conducted with stock spring steel that required minimal modifications. The shape formed by the rod is not a function of the length. The experiments we show were conducted using a strip of 5.08 cm wide 1095 spring steel. Each strip is approximately 23 cm. We model the metal strip of length 20 cm. At each end, 1.5 cm is used to attach the metal strip to the Adept robot and the table. Figure 3.12 shows an image of a strip of spring steel used to conduct experiments. A ruler is shown below the strip to provide the scale. To increase the visibility and improve segmentation, the metal strip is instrumented with red electrical tape along its bottom edge. In addition to this tape, we have also affixed yellow map tacks that are spaced at 2 cm intervals. The tape and the map tacks are assumed to have a negligible impact on the shape of the metal strip.

Experiments using metal strips with a thickness ranging from 0.1016 mm to 0.1778 mm all provided similar results. Thinner metal strips do not exhibit elastic characteristics and were also more likely to develop permanent folds. Thicker metal strips were too strong to be bent using the Adept robot. The release of potential energy when thicker strips experience instabilities has the potential to damage test equipment.

Experiment Table

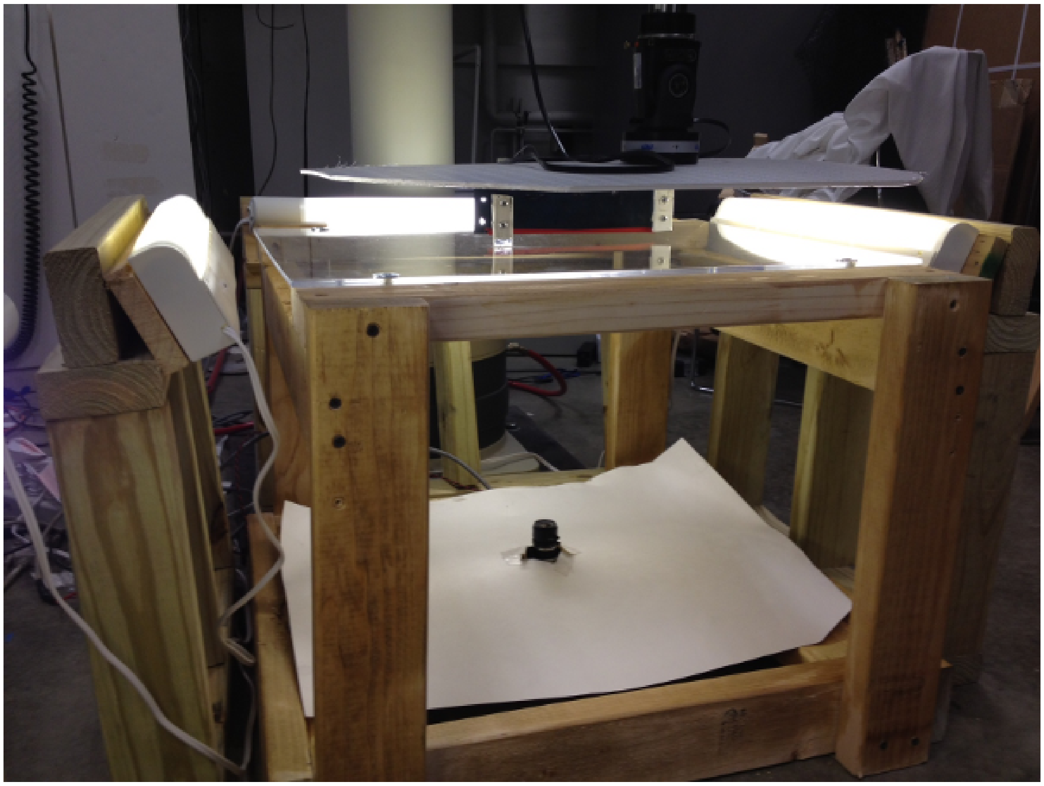


Figure 3.13: The system used for conducting experiments. This included the Adept robot, the table constructed, the metal strip, and the camera used to capture images.

We have designed and constructed a custom table and lighting solution that allows for capturing images using a digital camera below the surface. Figure 3.13 shows an image of the experiment table. The main structure is composed of 2x4 boards. The mass was sufficient to prevent the table from moving as a result of the force applied by the metal strip. The surface of the table consists of a .635 cm thick sheet of polyurethane. At the center of the sheet, two screws are used to attach a polyurethane end effector to the top of the surface. This end effector has two holes in addition to the holes drilled in the bottom, for attaching to the metal strip.

A metal plate is mounted below the surface of the table which has holes drilled to align with the mounting holes for the Pt Grey Flea 2 camera. A piece of white poster board is placed above the metal plate so that reflections of the table are not visible in images captured by the camera. To reduce the amount of variability in lighting within images captured by the digital camera, four lights are placed on all sides of the experiment table. The clear surface allows for capturing images of the metal strip from below the surface of the table. This is opposed to capturing images from the side or from above (mounted on the robot). Both of these alternatives have disadvantages which include occlusions for certain configurations and the need to adjust each frame to account for the motion of the camera.

Experiment Manipulator

Adept Robot and Robot Controller

One end of the metal strip is manipulated using an Adept XL-One robot. This robot provides five degrees of freedom. Only three are necessary for manipulating planar metal strips. Similar to the mount on the table, an end effector is also attached to the bottom of the Adept-XL One robot arm. This mount has two holes that allow the metal strip to be attached using two screws and two nuts. An Adept 6-axis force sensor is attached between the robot arm and the mount, which is attached to the metal strip. This sensor is aligned with an end effector so that the axes correspond with the orientation of the end of the metal strip. White foam board is attached to the robot below the force sensor to ensure that the background behind the metal strip is a consistent color.

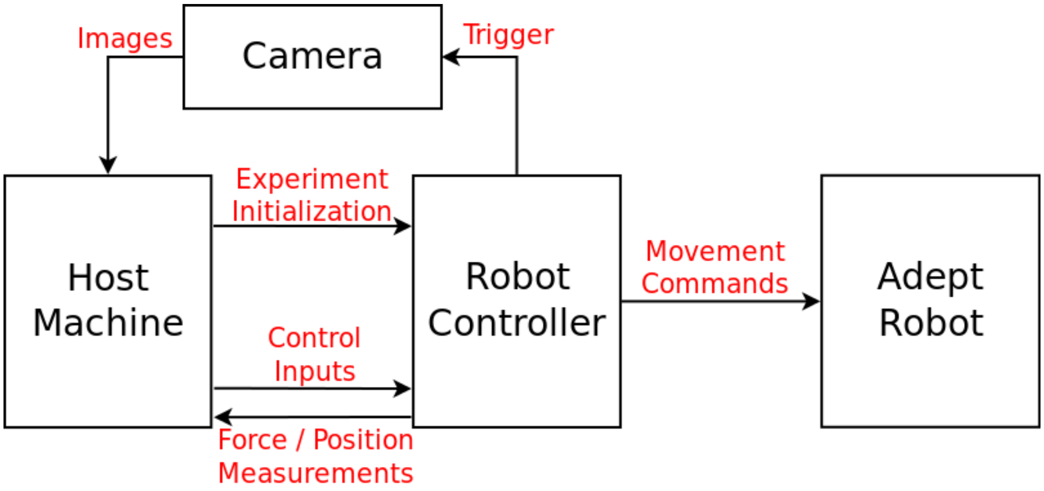


Figure 3.14: Block diagram of hardware interfaces and data that is transmitted by each component. This includes the Adept robot used for manipulation, the robot controller, and the host machine.

The Adept robot has three channels for providing information to the host machine and the digital camera. Figure 3.14 shows a schematic of the interconnects between all of the components. The Adept robot has a parallel port which allows for a GPIO interface. We use a custom cable that allows for one bit of this interface to provide a trigger to the digital camera. The Adept robot is also connected to the host machine via USB and Ethernet connections. The USB connection is required to make system calls. We have written scripts, using the Expect programming language, that automate interactions with the terminal interface to the USB port. The faster Ethernet connection is used when communicating force measurements and position measurements.

Camera

The camera is responsible for capturing images from below the surface of the table and passing these images to the host machine. The results presented in this thesis are taken using a Pt Grey Flea 2 camera. The camera has one input and one output. Pin 1 is the input trigger that comes from the Adept controller. A parallel cable was modified to connect a digital output from the Adept controller and a ground pin to a connector that is compatible with the Flea2 camera. The images are sent via an IEEE 1394 (Firewire) interface to the host machine. All images are captured in RGB format and have a resolution of 640x480 pixels. The camera is configured manually and the configuration settings for the experiments are provided in Table D.1. The manual configuration along with the lighting solution ensures more consistent RGB values for all of the images captured over the duration of experiments. The camera is configured to only capture images when the input trigger coming from the Adept controller detects a rising edge.

Host Machine

The host machine has three interfaces that allow for communicating with both the robot and the digital camera. Two interfaces connect the host machine to the Adept robot for communicating control commands and data. The third interface is the IEEE 1394 connection to the digital camera. Though the camera is triggered using a GPIO port on the Adept robot, the captured images are transferred to the host machine for storage and off-line processing.

Experimental Software

The details of our software implementation can be found in Appendix D. This includes low level software that controls the Adept robot. We have written robot control software that initializes the state of the robot and configures experiment parameters. Higher-level code has been implemented that controls the robot position and captures images and records data measurements from the robot. We have also developed post-processing software that parses the generated experiment files, generates annotated images of the elastic rod, and compares the results of experiments to the model we have presented. This software can be downloaded at http://www.matharlabs.com/research/MatthewsSourceCode.tar.gz.

3.4.2 Validation of Configuration Space

The first hardware experiment we present shows the relationship between the derived coordinates, $\mathcal{A}$, and the force and torque at the end of a planar metal strip. Our model predicts the parameters a_1, a_2, and a_3 are directly proportional to the force in the x direction (f_x), the force in the y direction

(f_y), and the moment about the z axis (m_z) at the base of the elastic rod. This was shown in Section 3.2.1. The scaling factor between these sets of quantities is a function of the stiffness and length of the object being manipulated. With this experiment, we validate this relationship. We also show how this relationship can allow for a quantitative measurement of the current configuration of the elastic rod using a force/torque sensor.

Procedure

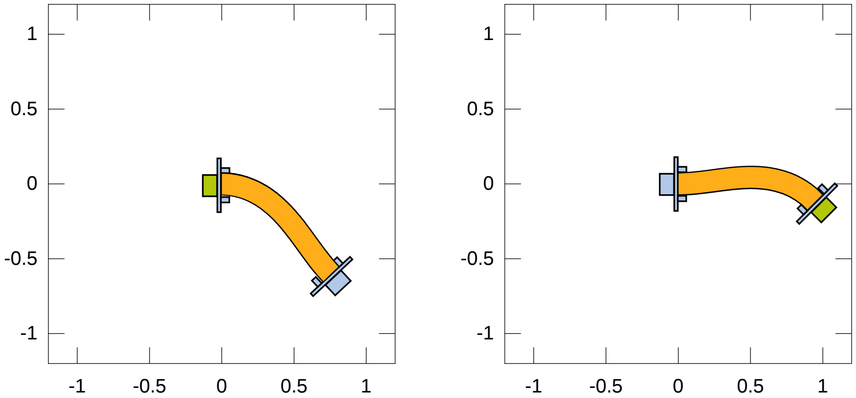


Figure 3.15: These schematics show the same configuration, but with two different origins. The green end effector indicates the $x(0)$. The schematic on the left shows the configuration with the force sensor is mounted on the table. The schematic on the right shows the same configuration but with the force sensor mounted to the robotic manipulator.

In order to validate the relationship between the configuration space $(\mathcal{A})$ and the force at the end of the rod, we perform experiments that determine the linear relationship between these two quantities. We have conducted an experiment where the metal strip moves through a sequence of configurations and the force at the end effector attached to the robot is measured for each configuration. The Adept robot is instrumented with a 6-dimensional force sensor that measures the force applied to the robot by the metal strip. The force sensor provides both force and torque measurements in each of the x, y, and z directions. We attach the force sensor to the robot and so the end of the metal strip that is connected to the robot is considered to be $x(0)$ and the end connected to the table is considered to be $x(1)$. This is simply done by calculating the difference between the original end of the strip, $x(1)$, and each of the points along the strip. After the offset has been calculated for each point, all of the points are rotated by the angle $x_3(1)$. Figure 3.15 shows an example of this transformation. These schematics show the identical configuration, but with different origins. The schematic on the left shows the configuration where $a = [-5, 0, -2]$ when the origin, $x(0)$, corresponds with the end connected to the table. The schematic on the right shows the same configuration, but with the origin connected to the robotic manipulator.

The goal of this experiment is to calculate a matrix, H, that transforms a point in $\mathcal{A}$ to the force-torque such that

$$\begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix} = H \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \tag{3.14}$$

If there is an invertible matrix H, there will also be the inverse relationship where

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = H^{-1} \begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix}. \tag{3.15}$$

Error Metric

The force/torque sensor provides a straight forward measurement of the error between the predicted and observed configuration. Once the H matrix has been determined, the configuration of the metal strip can be measured by applying Equation (3.15) to the force. For experiments, there is a predicted configuration (a_1,a_2,a_3) and the observed configuration (a_1',a_2',a_3'). We define the error as

$$error^2 = (a_1 - a_1')^2 + (a_2 - a_2')^2 + \lambda^2 (a_3 - a_3')^2. \tag{3.16}$$

Because the torque is not the same unit as the two force components, the error $(a_3 - a_3')^2$ is scaled by a constant value λ. For data shown in this thesis, $\lambda = 4$. This was calculated based on the measured ratio between the magnitude of the force measurements and the torque measurements.

For this experiment, we identified 32 straight-line paths between points in $\mathcal{A}_{\text{free}}$. Each path is equally divided into 50 equidistant points. For each point, the position to place the robot is calculated using the algorithm shown in Figure 3.7 and the robot is moved to the appropriate position and orientation. The robot remains in this configuration while 50 force/torque measurements are taken by the Adept robot. We retain the median measurement for each step of the experiment.

After the experiment is completed, two matrices are constructed. The first matrix is constructed by concatenating all of the predicted points such that

$$A = \begin{bmatrix} a_1(1) & a_1(2) & ... & a_1(i) \\ a_2(1) & a_2(2) & ... & a_2(i) \\ a_3(1) & a_3(2) & ... & a_3(i) \end{bmatrix}, \tag{3.17}$$

where $a_1(2)$ is the value of a_1 for step 2 of the experiment. The second matrix is constructed by concatenating all of the measurements such that

$$F = \begin{bmatrix} f_x(1) & f_x(2) & ... & f_x(i) \\ f_y(1) & f_y(2) & ... & f_y(i) \\ m_z(1) & m_z(2) & ... & m_z(i) \end{bmatrix}. \tag{3.18}$$

Once these two matrices have been constructed, linear least squares regression is used to calculate the matrix H where

$$H = F \setminus A. \tag{3.19}$$

Results

The results from this experiment validate the relationship between the configuration space and the force/torque at the end of the rod. The experiment produced the transformation

$$H = \begin{bmatrix} 0.0052 & -0.1267 & -0.0060 \\ 0.1275 & -0.0045 & 0.0213 \\ 0.0000 & 0.0008 & -0.0263 \end{bmatrix} \tag{3.20}$$

for a strip of spring steel strip with a thickness of .007". Figure 3.16 shows the comparison between the predicted components for each a_1, a_2, and a_3 and

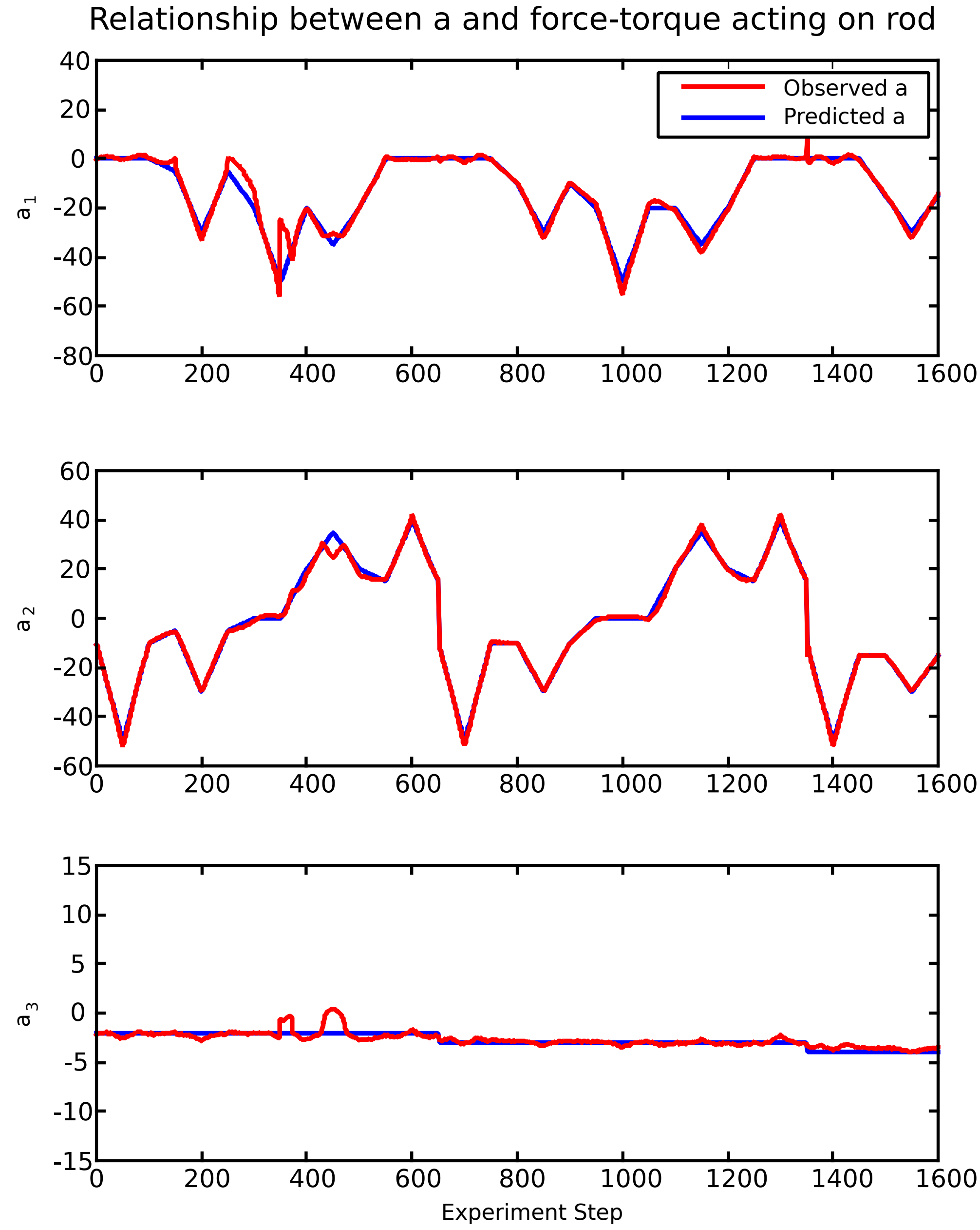


Figure 3.16: Comparison between the modeled configuration in $\mathcal{A}$ and measured configuration using a force-torque sensor.

the observed values of these quantities. We determined the measured value, **a'**, by applying the inverse of the H matrix to the force/torque measurements using Equation (3.15). The lines corresponding to the measured values for a and the modeled values are nearly identical for the majority of the path

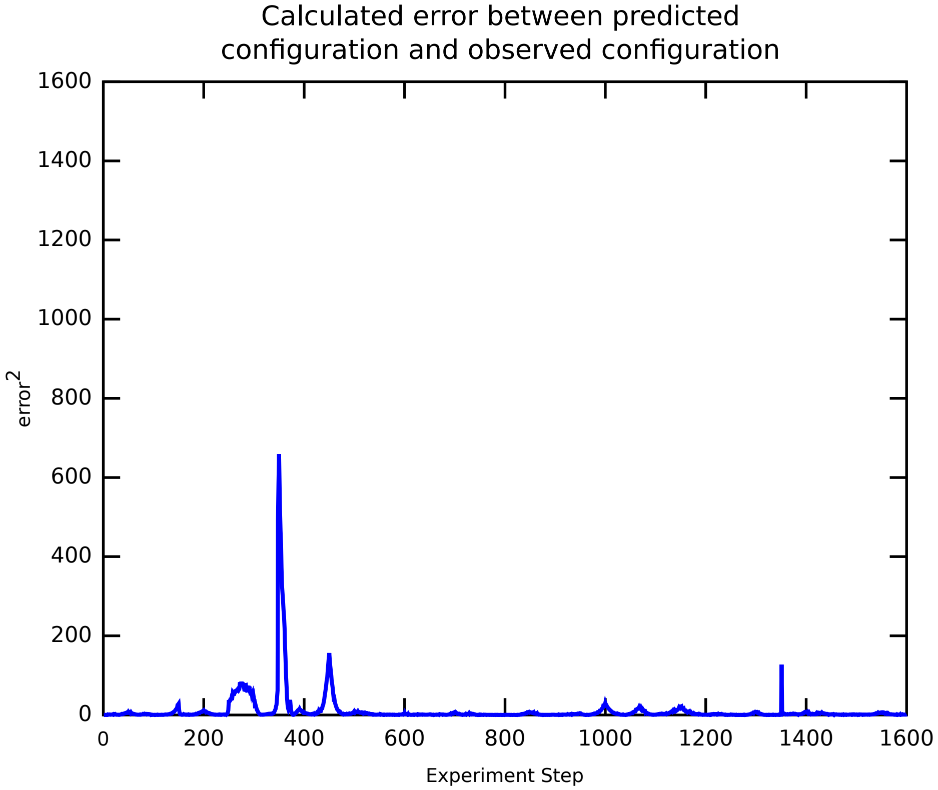


Figure 3.17: This plot shows the error between the predicted configuration and the observed configuration for the experiment shown in Figure 3.16 using Equation (3.16).

planned. Figure 3.17 shows the error as defined by Equation (3.16). This plot shows that a relatively small amount of error was observed with this experiment. With the exception of one "peak", the measured $error^2 < 200$. In Section 3.4.4, we will describe the cause of this peak where the highest error is seen.

Ideally, the transformation matrix would be a diagonal matrix where

$$H = I\gamma. \tag{3.21}$$

The parameter γ is correlated with the stiffness of the elastic rod. This is not a diagonal matrix due to a 90 degree rotation of the force sensor. Additionally, the third row refers to the torque exerted by the rod. Because this quantity is a product of force and distance measurements, the scaling incorporates the length.

The relationship between force and $\mathcal{A}$ provides the ability to describe the shape of the rod with only knowledge of the force at the base. Additionally, the inverse of this operation can be performed. Force measurements can be performed with a camera and computer vision algorithms that determine points along the rod and use this information to estimate the current configuration in $\mathcal{A}$.

3.4.3 Validation of Free Space

The second set of experiments serves to confirm that our model accurately predicts infeasible configurations of the metal strip. These experiments provide valuable insight for future manipulation of flexible objects. Certain events, such as instabilities and self-collisions, can damage either the metal strip or the robotic manipulator. By performing these experiments, we are able to further constrain $\mathcal{A}$ to better ensure that manipulation tasks do not produce undesired outcomes. This characterization provides insight for planning paths that approach the boundary of our model for configurations that can be achieved.

Procedure

The goal of the experimental procedure is to recreate the boundary seen in Figure 3.9 with hardware experiments. Experiments were performed for 9 different values for $a_3 \in -16, -14, -12, -10, -8, -6, -4, -2, 0$. For each slice, the calculated a_1 and a_2 boundary is sampled at 36 equidistant intervals. For each experiment performed on a slice in a_3, the initial configuration is the exact same configuration. Figure 3.18 shows the paths for the predicted boundary where $a_3 = -8.0$. All of the experiments for this value of a_3 start at the configuration where $\mathbf{a} = [-59.0, 48.0, -8.0]$. The green lines show the path followed for each experiment until the predicted boundary is reached. The red circles denote the point at which the path reaches the predicted boundary. Each path that is confined to the slice of a_3 is then planned to a point along the calculated boundary. This ensures that any variation from our model takes place within the plane of a_3 and identifies a boundary point for the selected slice. The robot then follows this path and the experiment continues until one of the three constraints on $\mathcal{A}$ is observed. For each

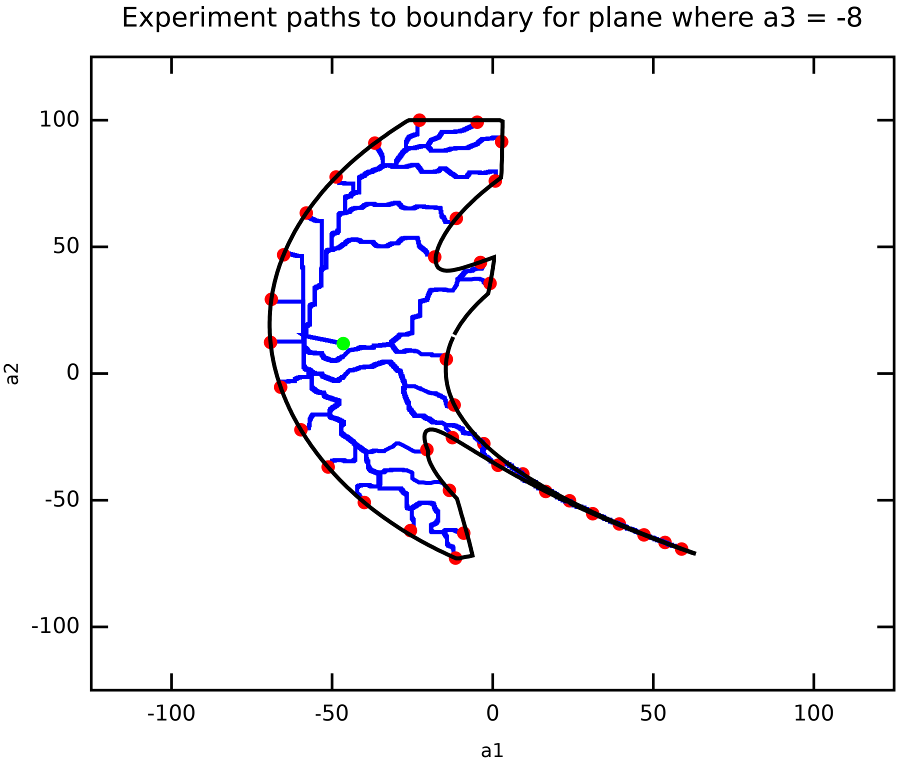


Figure 3.18: This plot shows paths that move from a point in $\mathcal{A}_{\text{free}}$ to points along the predicted boundary where $a_3 = -8.0$. Each blue line corresponds to a separate experiment. The green circle indicates the point in $\mathcal{A}$ at which all of the experiments start. The boundary point where the experiment is predicted to terminate is indicated by the red circle.

step of the experiment, an image of the metal strip is captured and then annotated with the predicted shape of the metal strip. The configuration is also measured using the force sensor for each configuration.

A human operator is responsible for terminating experiments once the observed configuration deviates from the predicted configuration. The three types of infeasible configurations are shown in Figures 3.19 - 3.21. Figure 3.19 shows the metal strip before and after it experiences an instability that resulted in a large deformation of its shape. Notice that the two end effectors are in approximately the same position and orientation. However, this small change has resulted in a large change in the shape of the elastic rod. Figure 3.20 shows three types of collisions that can be experienced when performing manipulation with the metal strip. When performing manipulation experiments, it is necessary to ensure there are no collisions between: the

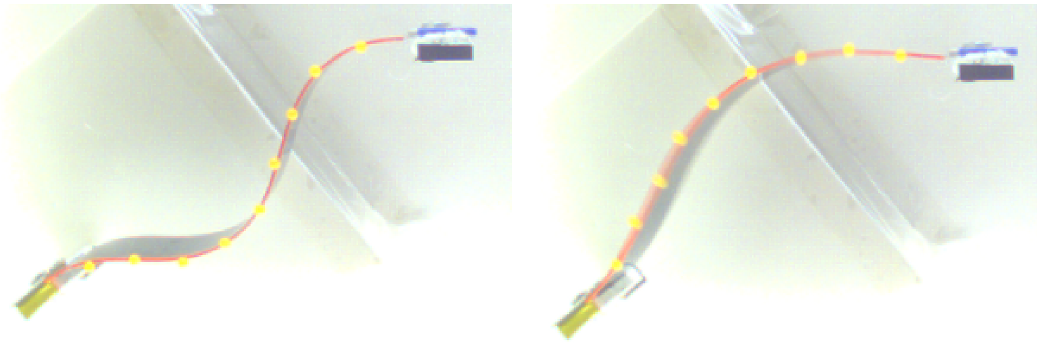


Figure 3.19: These two images show the shape of the rod immediately before and after it experiences an instability. The result of this instability is a large deformation in which the shapes of the metal strip differ significantly.

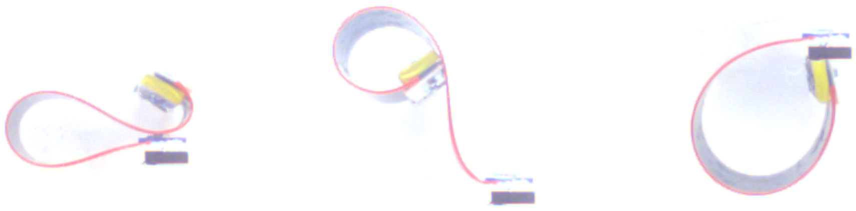


Figure 3.20: Three types of collisions that can be experienced. These situations include the metal strip colliding with itself, the metal strip colliding with the robot and the robot colliding with the table.

robot and the table, the robot and the metal strip, and the metal strip and itself. Figure 3.21 shows two examples of infeasible configurations that are due to limitations of the Adept robot. These two images show the rod when the robot has reached a joint limit. After the experiment is terminated, the metal strip is moved back to an initial configuration.

Error Metric

We define error in these experiments as the Euclidean distance between the observed boundary of $\mathcal{A}$ and the predicted boundary. For each of the 36 experiments for each value of a_3, the error is calculated by

$$error^2 = (a_1 - a_1')^2 + (a_2 - a_2')^2 \tag{3.22}$$

Figure 3.21: Two configurations for the metal strip in which the robot has reached its maximum rotation angle.

where a_1 and a_2 correspond to the predicted point on the boundary for this value of a_3. The values a_1' and a_2' correspond with the point in $\mathcal{A}$ where the metal strip is placed into an infeasible configuration. The component a_3 is not factored into the error calculation for these experiments because it remains constant for the entire path.

Results

We have shown with hardware experiments that the observed boundary adheres to the predicted boundary extremely well for some values of a_3. The results shown in Figures 3.22 - 3.37 both show the ability of our model to predict when the metal strip will be placed into an infeasible configuration. The plots in Figures 3.22 - 3.29 show a contrast between the predicted boundary for eight values of a_3. Each plot shows two curves. The red curve indicates the predicted boundary of $\mathcal{A}_{\text{free}}$ that was calculated using a continuation method. The plot also includes a blue curve along with blue circles. Each circle corresponds with the point in $\mathcal{A}$ at which the experiment was terminated because the configuration was determined to be infeasible. The blue curve connects adjacent circles.

The predicted boundary of $\mathcal{A}_{\text{free}}$ does not always correspond well with the observed boundary. For experiments where $a_3 > -10$ the predicted boundary contains a long, narrow region where a_1 increases. The observed boundary does not include this same long, narrow region. While the calculated error for points in this region can be rather large, this region has a rather small volume. In addition to this discrepancy, we also see other deviations between the predicted boundary and the observed boundary. For values where $a_3 > -6$, the observed boundary of $\mathcal{A}_{\text{free}}$ does not correspond well with the predicted boundary as a_2 increases. All of these experiments terminated significantly before the predicted boundary was reached due to the metal strip experiencing an instability.

The histograms shown in Figures 3.30 - 3.37 also illustrate the discrepancies that are shown in the plots in Figures 3.22 - 3.29. When $a_3 < -8$, we see that $error < 20$. With these experiments, the long, narrow region is disconnected from the portion of the boundary that is validated. With the histograms that correspond with when $a_3 \geq -6$, more experiments have $error > 20$. However, for all values of a_3 more than half of the experiments have $error < 17$.

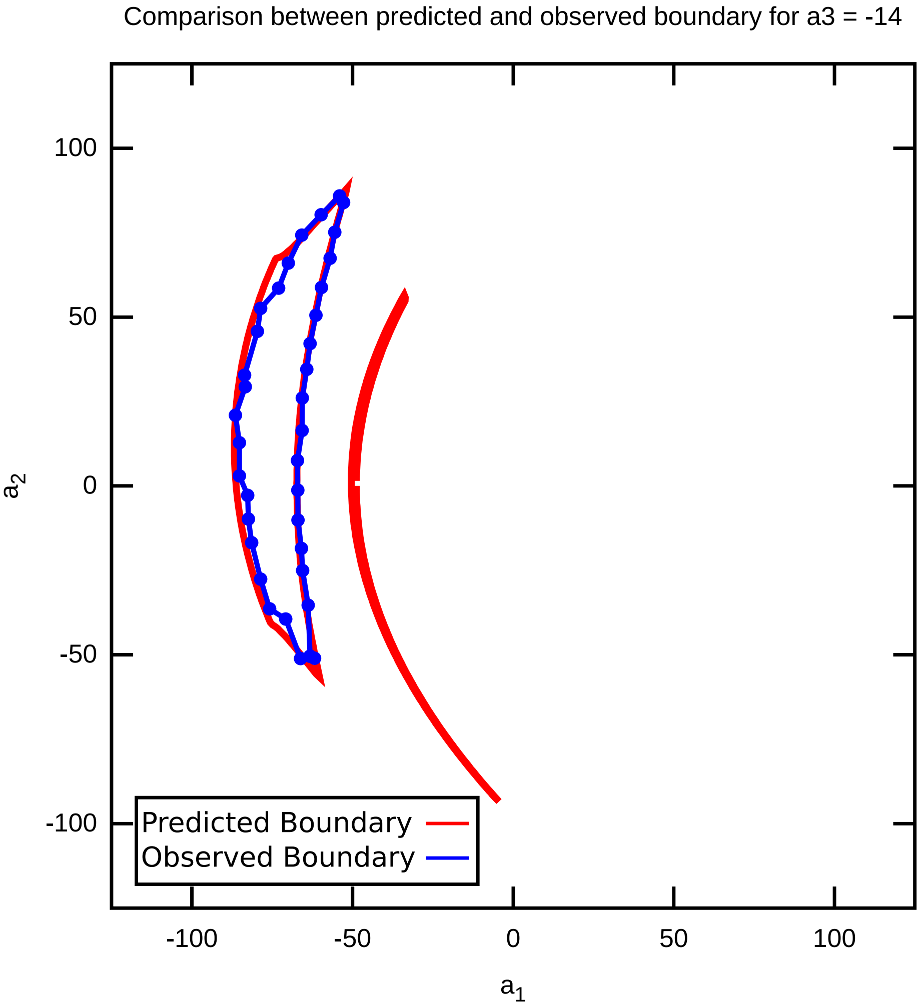


Figure 3.22: Comparison between modeled and measured boundary for $a_3 = -14.0$. The red line indicates the boundary found using the continuation method for $a_3 = -14.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary. The experiments conducted only validate the boundary of the larger of the two connected components.

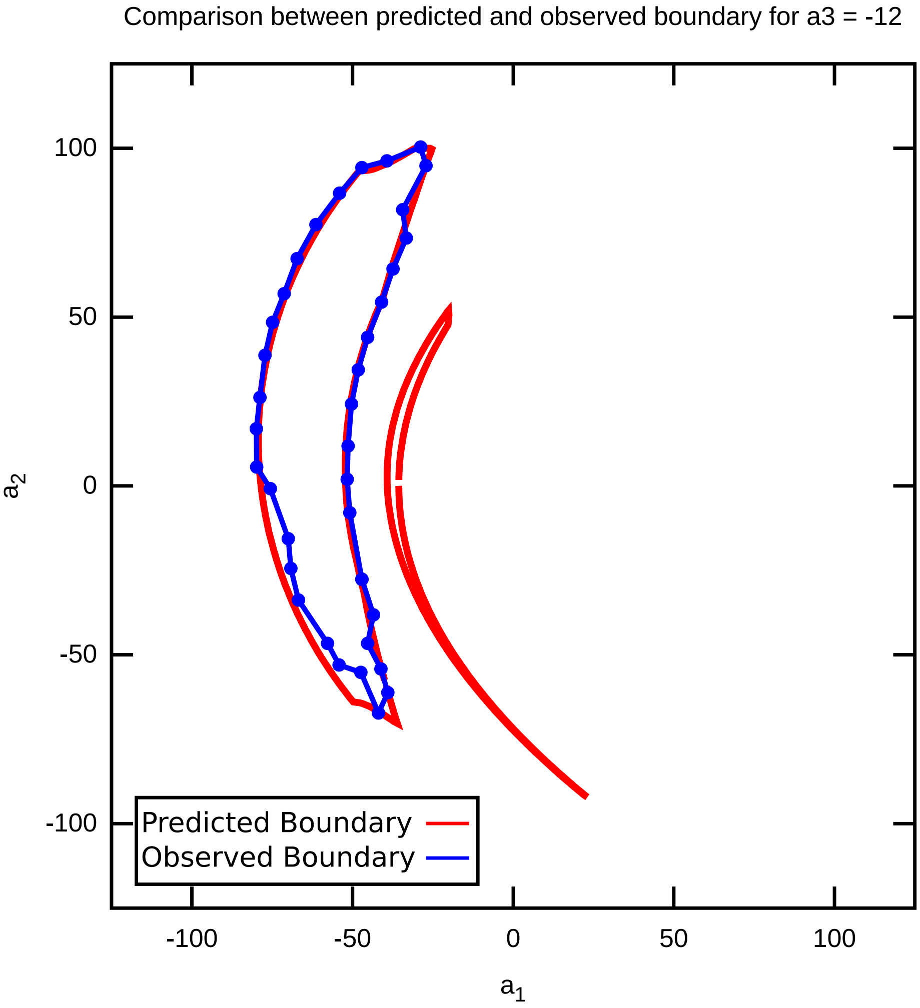


Figure 3.23: Comparison between modeled and measured boundary for $a_3 = -12.0$. The red line indicates the boundary found using the continuation method for $a_3 = -12.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary. The experiments conducted only validate the boundary of the larger of the two connected components.

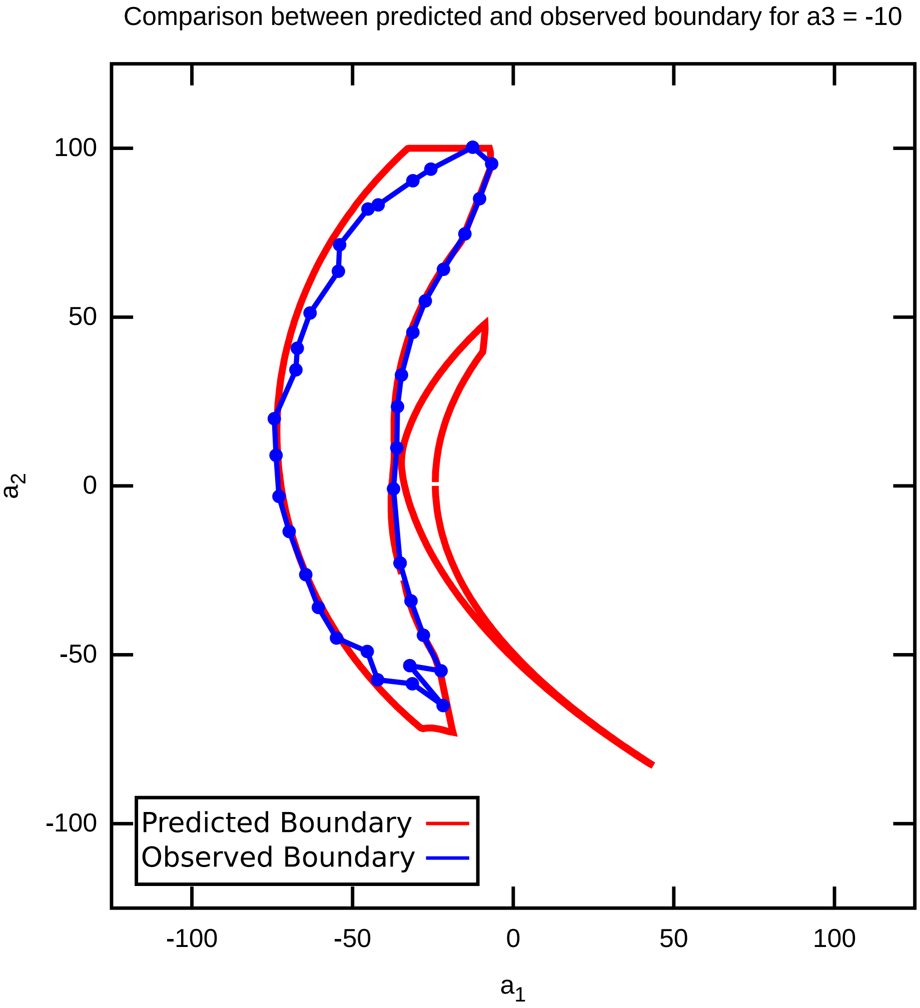


Figure 3.24: Comparison between modeled and measured boundary for $a_3 = -10.0$. The red line indicates the boundary found using the continuation method for $a_3 = -10.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary. The experiments conducted only validate the boundary of the larger of the two connected components.

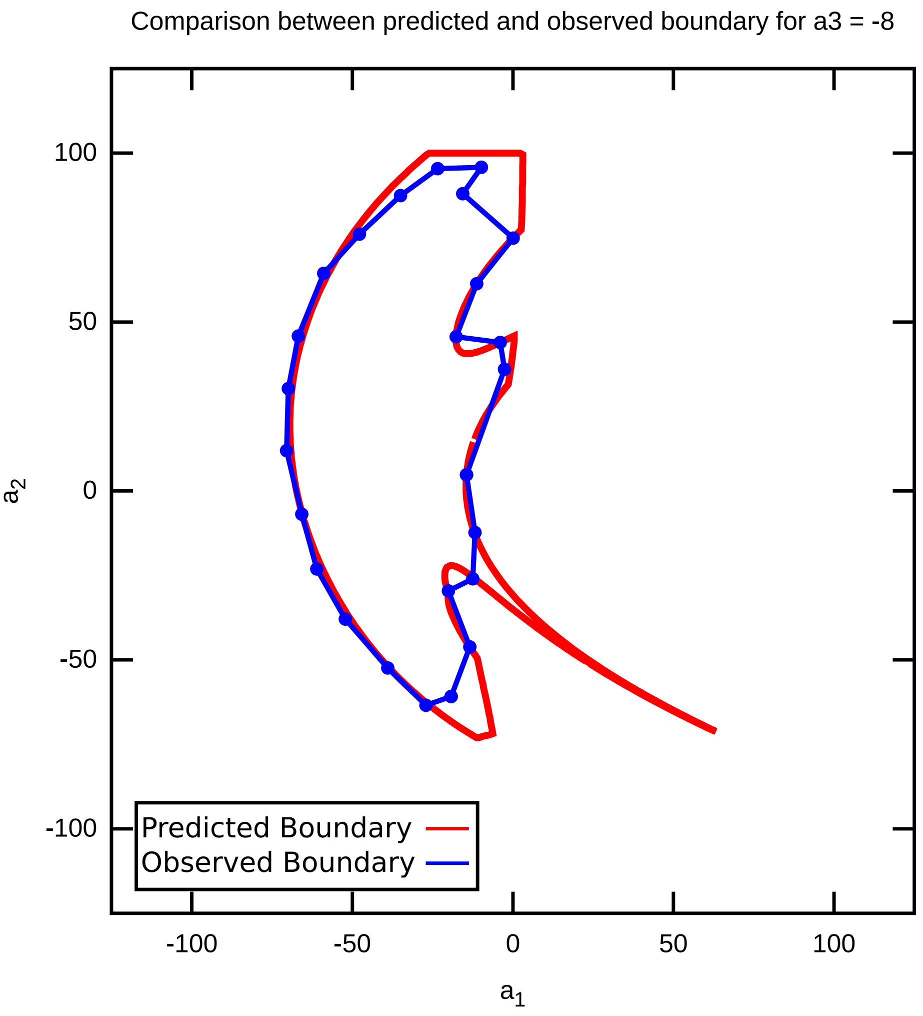


Figure 3.25: Comparison between modeled and measured boundary for $a_3 = -8.0$. The red line indicates the boundary found using the continuation method for $a_3 = -8.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary.

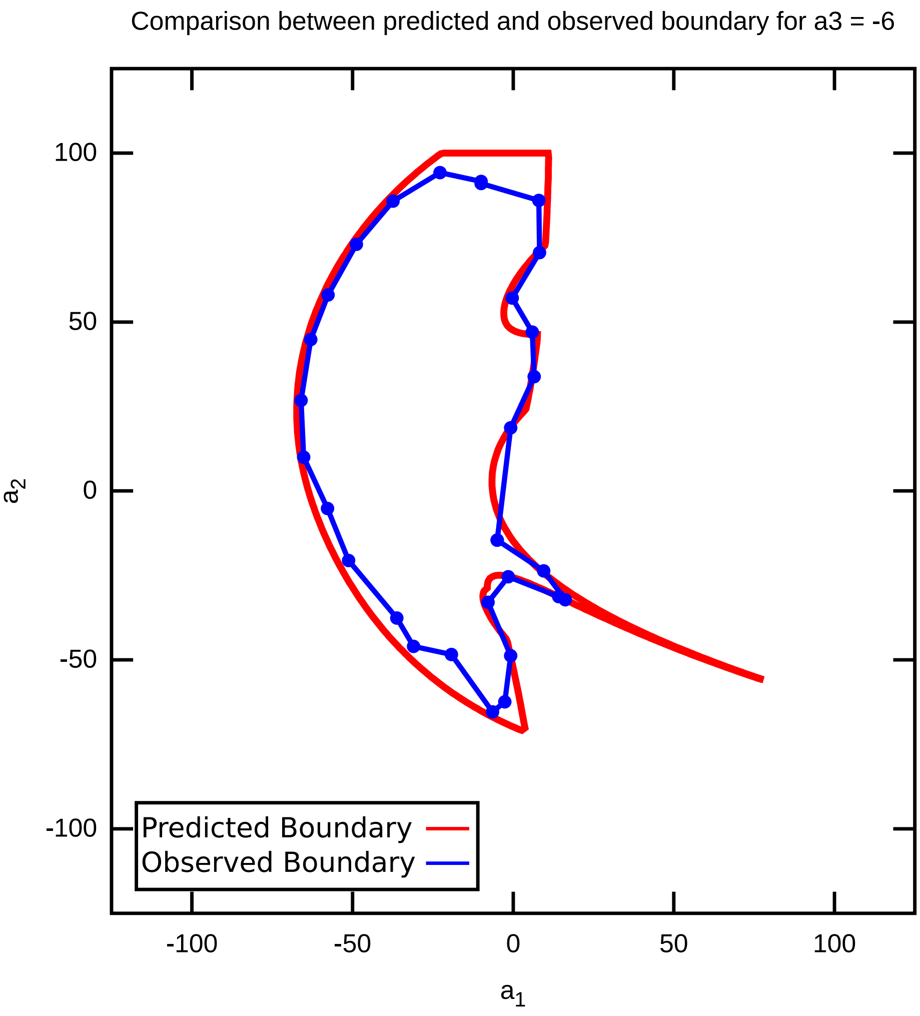


Figure 3.26: Comparison between modeled and measured boundary for $a_3 = -6.0$. The red line indicates the boundary found using the continuation method for $a_3 = -6.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary.

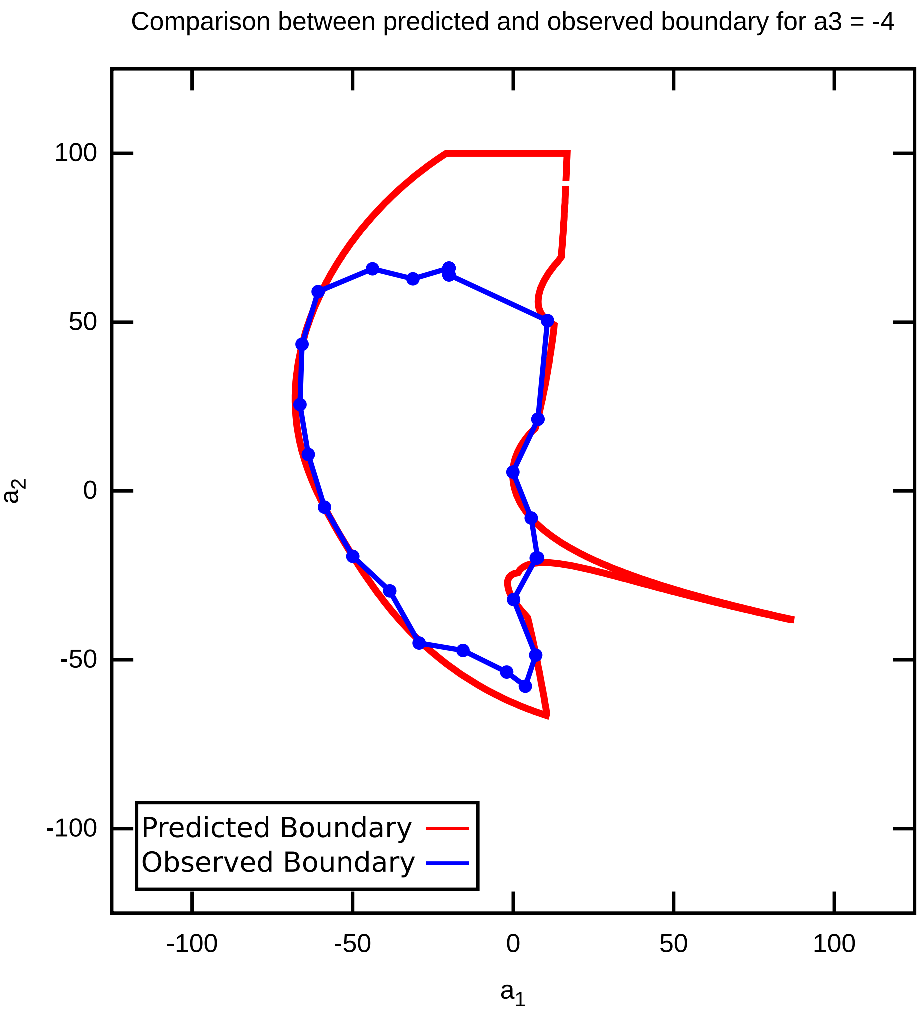


Figure 3.27: Comparison between modeled and measured boundary for $a_3 = -4.0$. The red line indicates the boundary found using the continuation method for $a_3 = -4.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary.

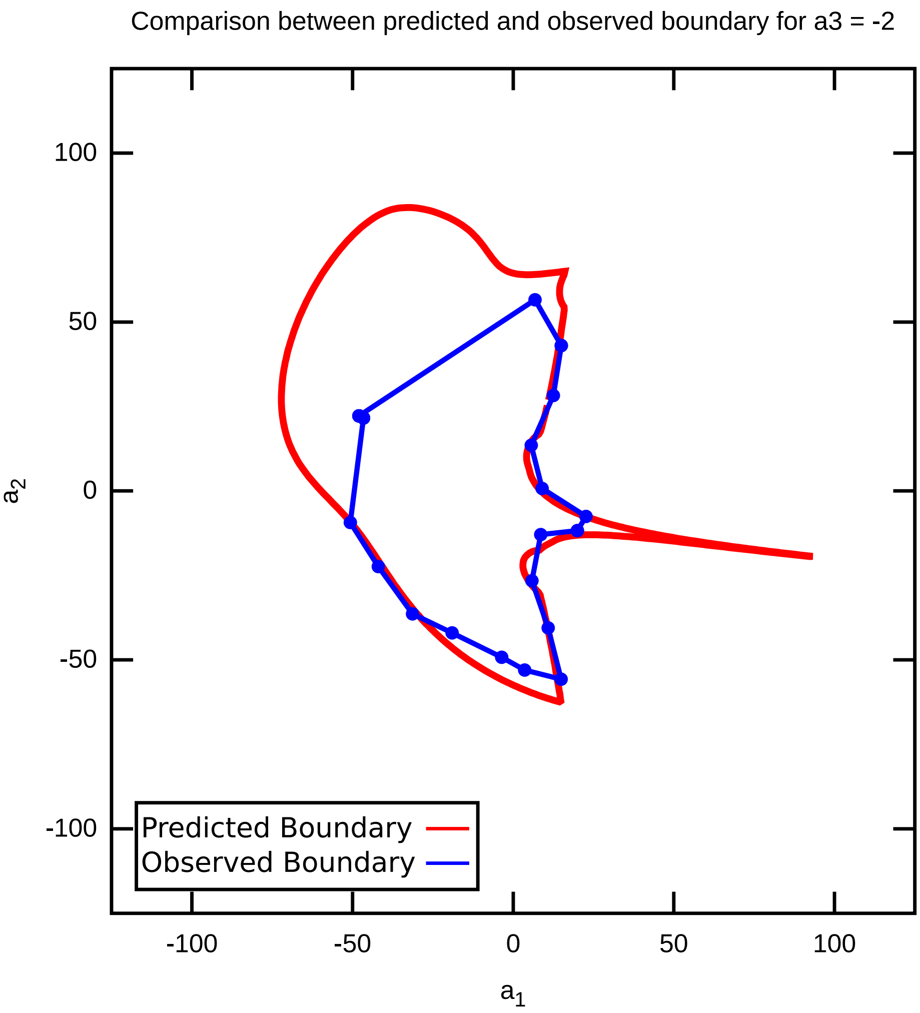


Figure 3.28: Comparison between modeled and measured boundary for $a_3 = -2.0$. The red line indicates the boundary found using the continuation method for $a_3 = -2.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary.

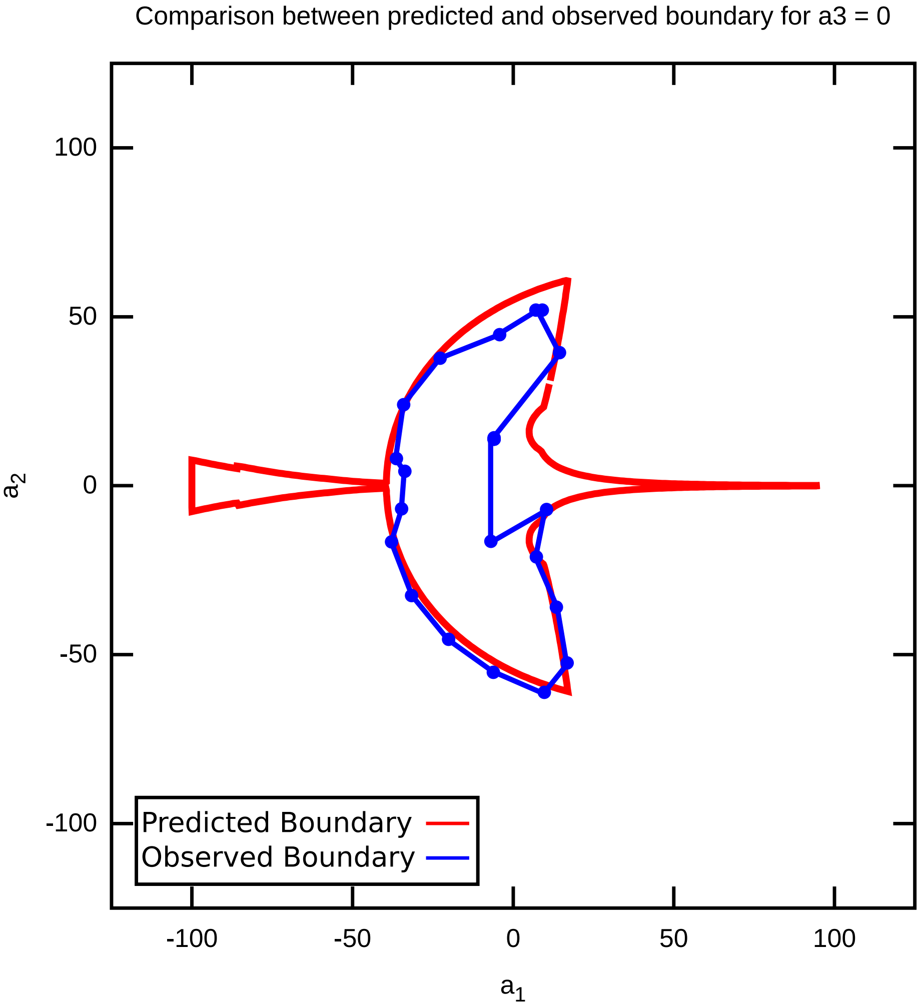


Figure 3.29: Comparison between modeled and measured boundary for $a_3 =$ 0.0. The red line indicates the boundary found using the continuation method for $a_3 = 0.0$. The blue points indicate the point at which the shape of the metal strip disagrees with the model. The blue line connects the points to allow for a comparison between the modeled and experimental boundary.

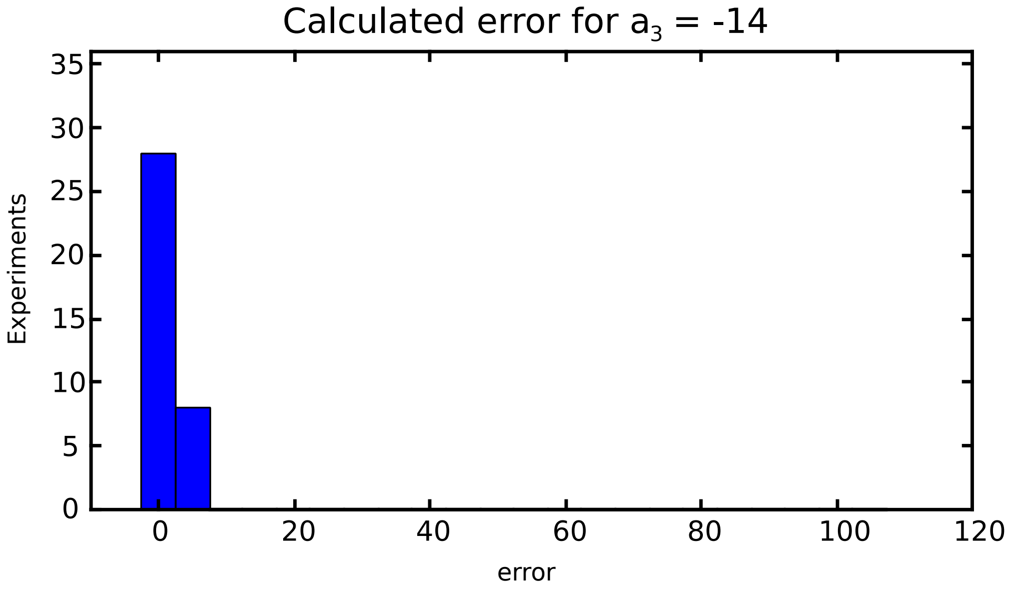


Figure 3.30: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -14$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

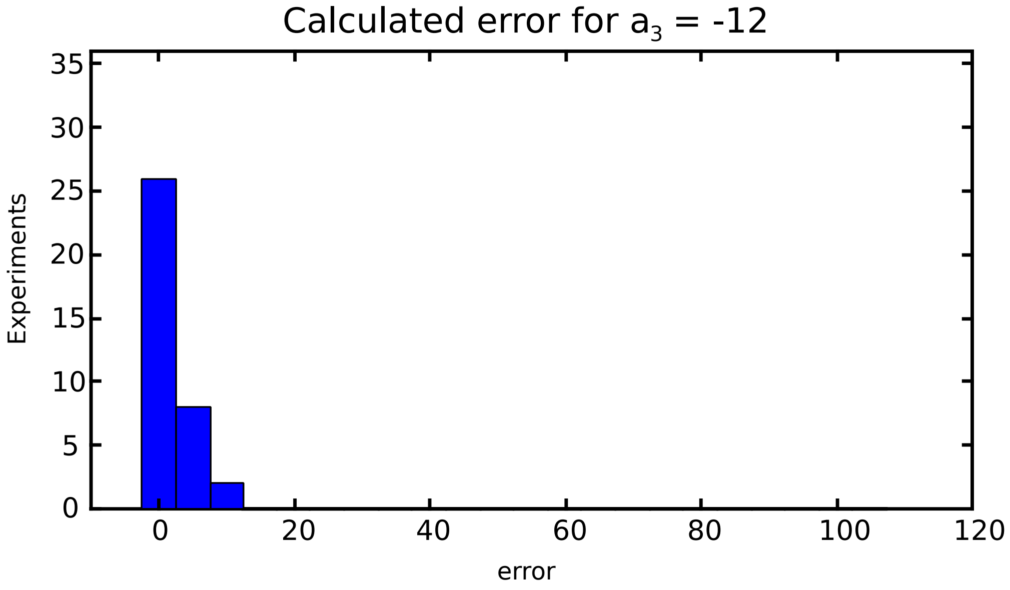


Figure 3.31: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -12$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

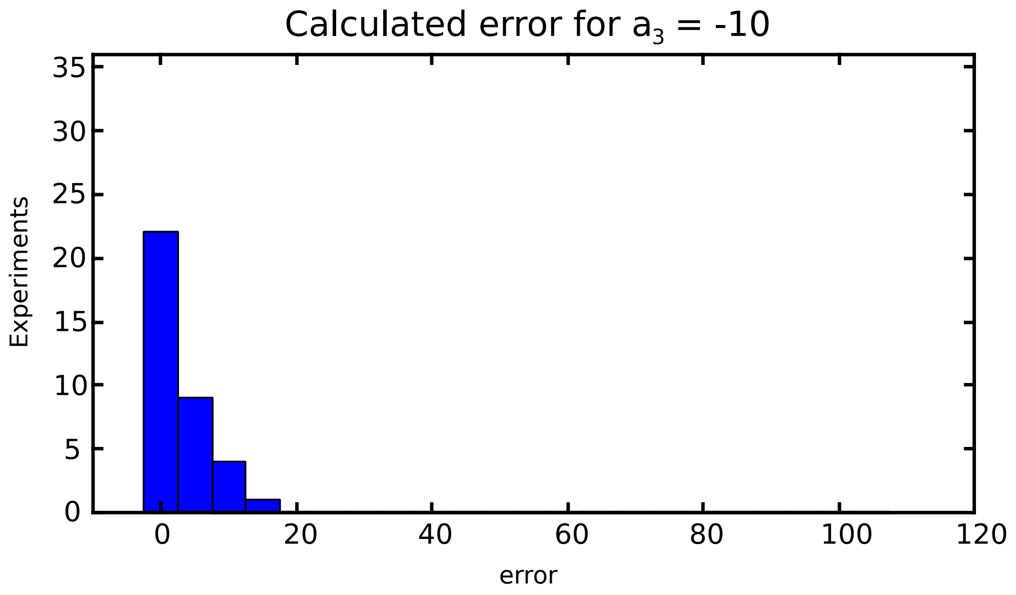


Figure 3.32: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -10$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

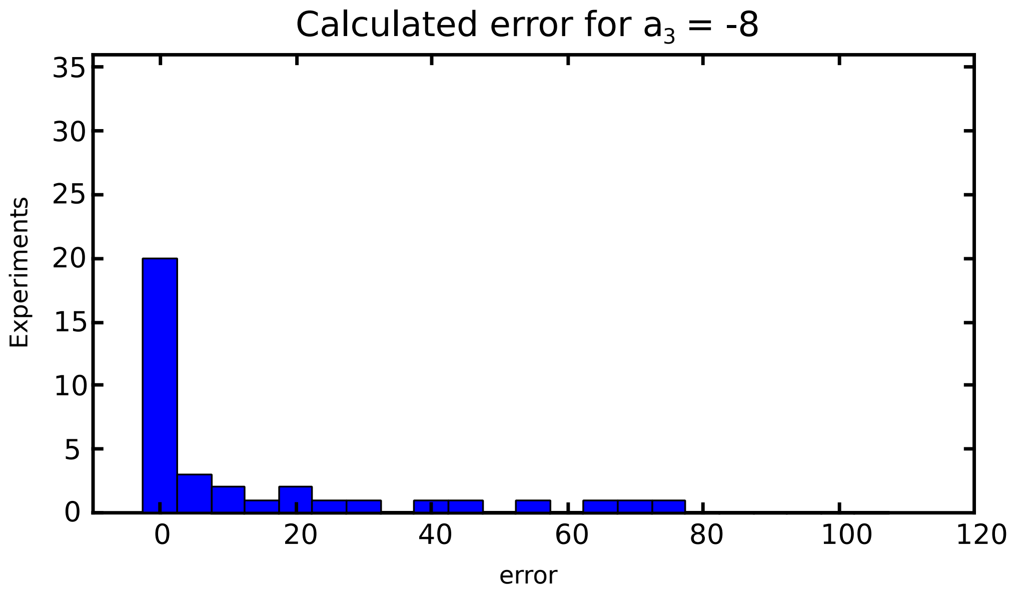


Figure 3.33: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -8$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

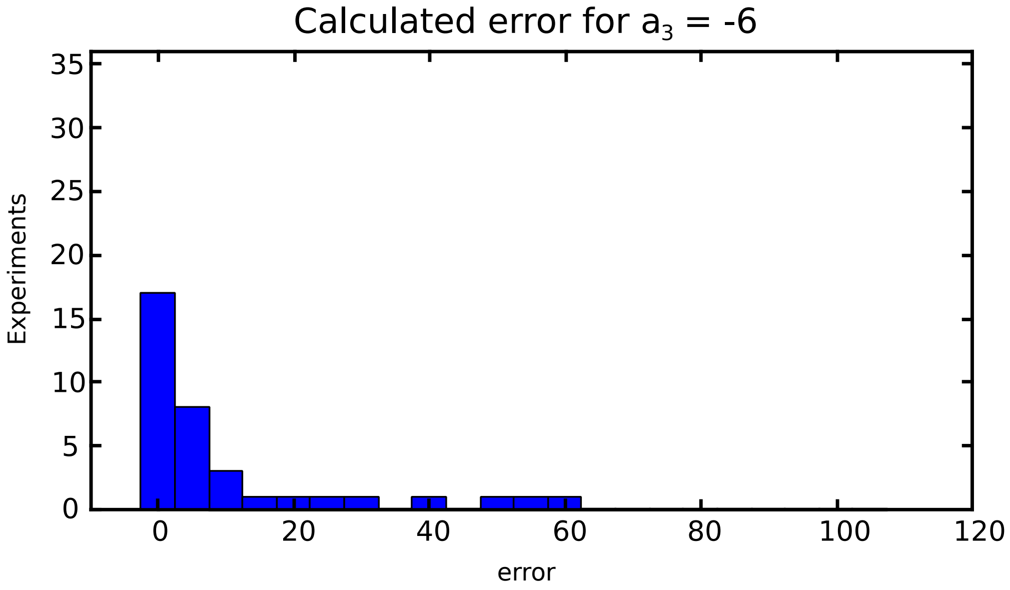


Figure 3.34: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -6$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

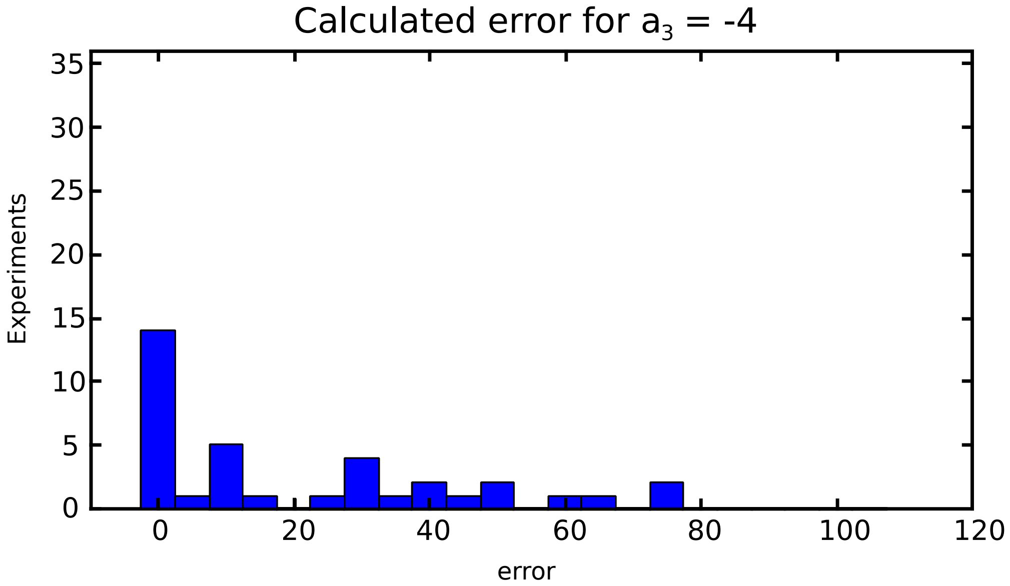


Figure 3.35: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -4$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

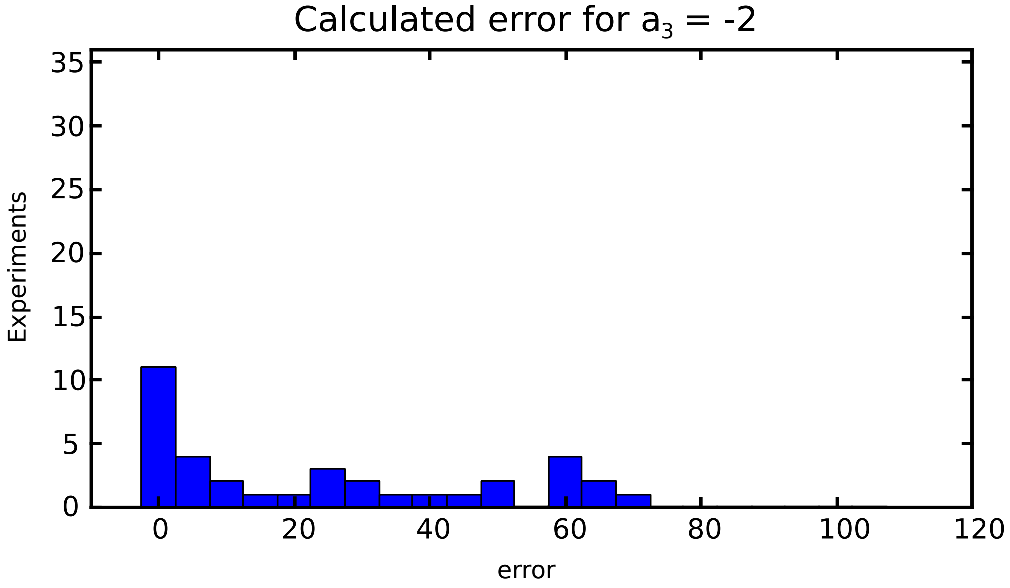


Figure 3.36: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = -2$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

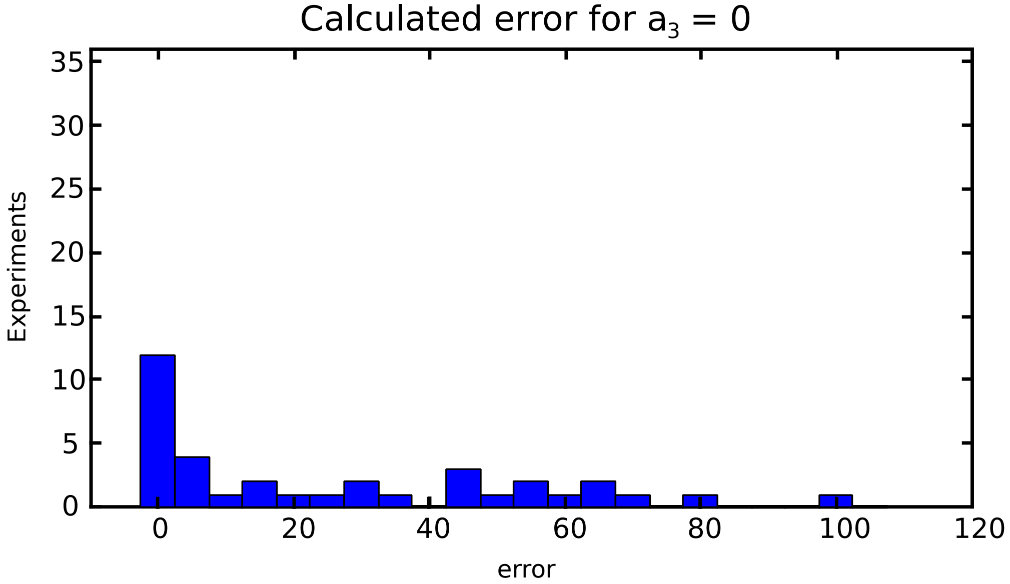


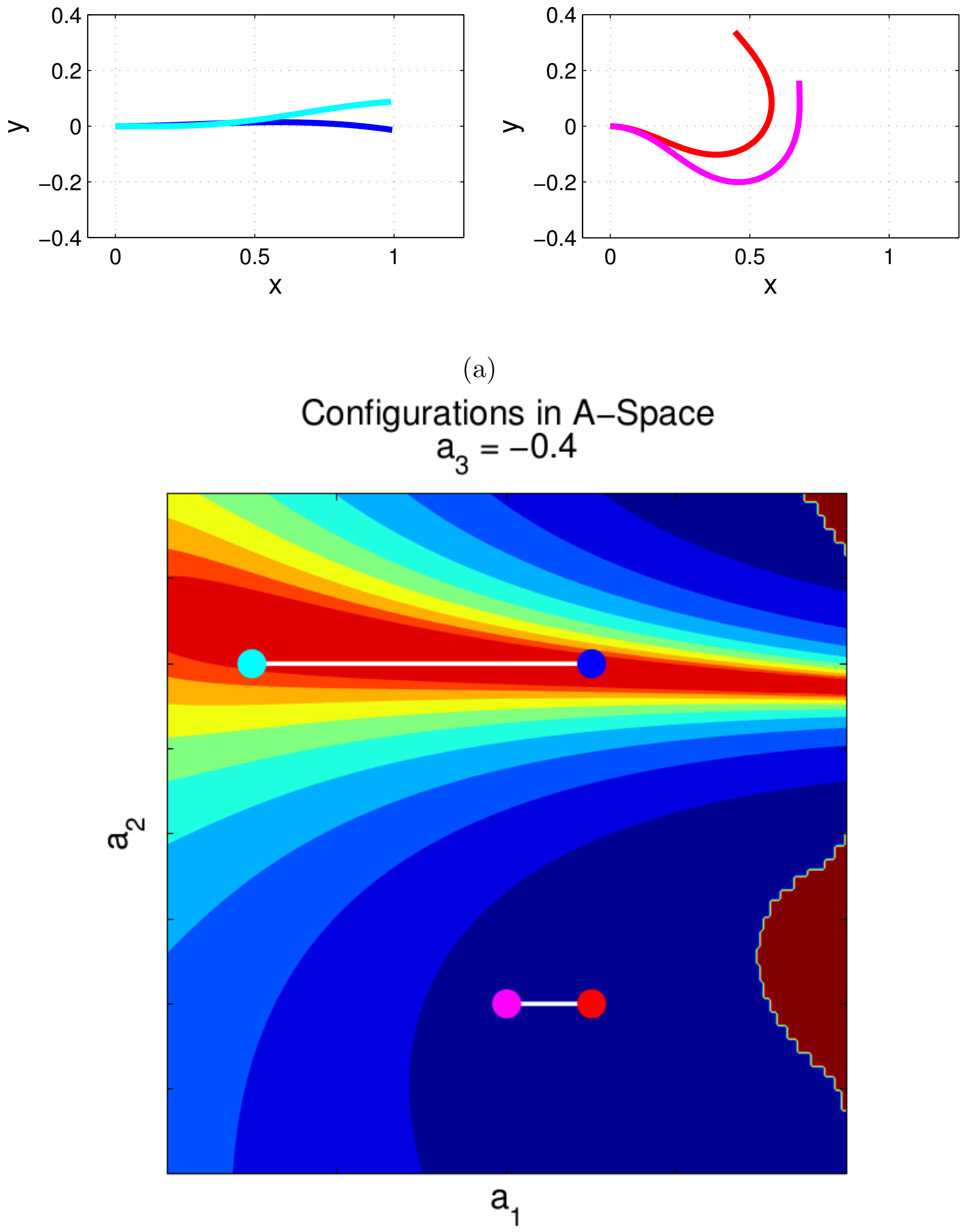
Figure 3.37: This histogram shows the error for each of the experiments that compare the predicted boundary where $a_3 = 0$ to the observed boundary. The error for each experiment was calculated using Equation (3.22).

3.4.4 Errors in Validation

While the results shown in Section 3.4.2 validate the linear relationship between the $\mathcal{A}$ and the measured force, this relationship does not always hold. For certain configurations, the measured value, a', does not correspond with the planned configuration. In fact, the discrepancy can be quite large. In this section, we present an experiment that shows that the Jacobian of our coordinate system ($\mathcal{A}$) with respect to the position of the end of the rod indicates configurations that exhibit large amounts of error. This error is seen both using the error metric from the previous section and also results in a discrepancy between the modeled shape and the physical shape of the metal strip. We conclude by presenting results from hardware experiments that validate that configurations with a "large" value for $J(1, a)^{-1}$, the inverse of the Jacobian at the end of the rod for a point $\mathbf{a}$ in $\mathcal{A}$, exhibit a "large" error.

The Jacobian, as described in Section 3.3.2, provides valuable insight about when the shape and force of the rod are sensitive to small changes in the boundary conditions. When the norm of the inverse of the Jacobian of a value in $\mathcal{A}$ with respect to the position ($\| J^{-1}(1, a) \|$) is "large", changes in the boundary conditions result in large changes in the force at the end of the rod. The presence of sub-millimeter error in robot placement can result in errors in force larger than 100%. On the other hand, when $\| J^{-1}(1, a) \|$ is "small", a large change in the boundary conditions only results in a small change in the force at the end of the rod. For the calculation of the norm for matrices in this thesis, we use the Frobenius norm.

Figure 3.38a shows two pairs of configurations that correspond to feasible shapes for the rod. The blue and cyan configurations are more similar in shape than the red and magenta curves. Despite the similarity in shape, the blue and cyan rods correspond to configurations that are separated by a larger distance in $\mathcal{A}$. Figure 3.38b shows all four points, corresponding to both pairs of configurations. These points, which all have $a_3 = -0.4$, are plotted onto a contour plot of the function of $\log(\| J^{-1}(1, a) \|)$. The red region, where both the blue and cyan points reside, indicates that this function where $\delta a \approx J^{-1}(1, a)\delta b$, a small δb yields a large δa. To summarize, when $J^{-1}(1, a)$ is large, even a small change (or error) in b produces a large change in a. In regions where $J^{-1}(1, a)$ is small, small changes in b have a smaller impact on the force seen at the end of the rod.



(a)

(b)

Figure 3.38: (top) This plot shows two different pairs of configurations. Both of these configurations appear to have "similar" shapes. (bottom) The four points show the points in $\mathcal{A}$ that correspond to the configurations shown in the top two images. This illustration shows that two configurations with approximately the same shape can have points in $\mathcal{A}$ with significant variation.

Procedure

To better validate these deviations from our model, we have conducted an experiment where the elastic rod intentionally moves through regions of $\mathcal{A}$ that have "large" values of $\parallel J(1,a)^{-1} \parallel$. We ensure this value is greater than 1000. For three values for $a_3 \in -1, -2, -3$, 12 distinct paths were planned. Only four paths were planned for when $a_3 = -4$. The same values for a_1 and a_2 were used for each of the values of a_3. This allows for the determination of how the sensitivity to error at the boundary conditions is impacted by the value for a_3 or the torque around the z axis. For this experiment, all 42 experiments were performed consecutively. Each path planned had a fixed length of 50 steps. For every step of the experiment, the measured value for $\mathcal{A}$ using the force sensor was compared with the planned value in $\mathcal{A}$.

Results

The experiment confirms that the predicted configuration deviates from the observed configuration when $\parallel (J(1,a))^{-1} \parallel$ is large. Figure 3.39 shows a comparison between a path that was planned in $\mathcal{A}$ and the measured configuration from the force sensor attached to the end effector. In this graph, the blue line corresponds with the measured $error^2$ between the planned configuration and the configuration measured using the force sensor. The red line shows the calculated quantity $\parallel (J(1,a))^{-1} \parallel$ for each of the planned configurations. Note that the results from the hardware experiment coincide well with the model, except for small regions where significant error is seen. The configurations that exhibit large amounts of error all have a large value for $\parallel (J(1,a))^{-1} \parallel$. These results clearly show that the measured value for $\mathcal{A}$ deviates from the model when the quantity $\parallel (J(1,a))^{-1} \parallel$ increases.

The sensitivity to small perturbations in the robot placement can manifest in both error between the predicted and observed shape of the metal strip and error, as defined in Section 3.4.2. Figure 3.40 shows a picture of the rod while executing a straight-line-path planned in $\mathcal{A}$. This particular path travels through a region where some configurations have $\parallel J^{-1}(1,a) \parallel > 3 \times 10^3$. These configurations are extremely sensitive to the robot placement. For the configuration shown, the shape of the rod deviates from the modeled shape. This deviation in shape transpires when we see large error. In the

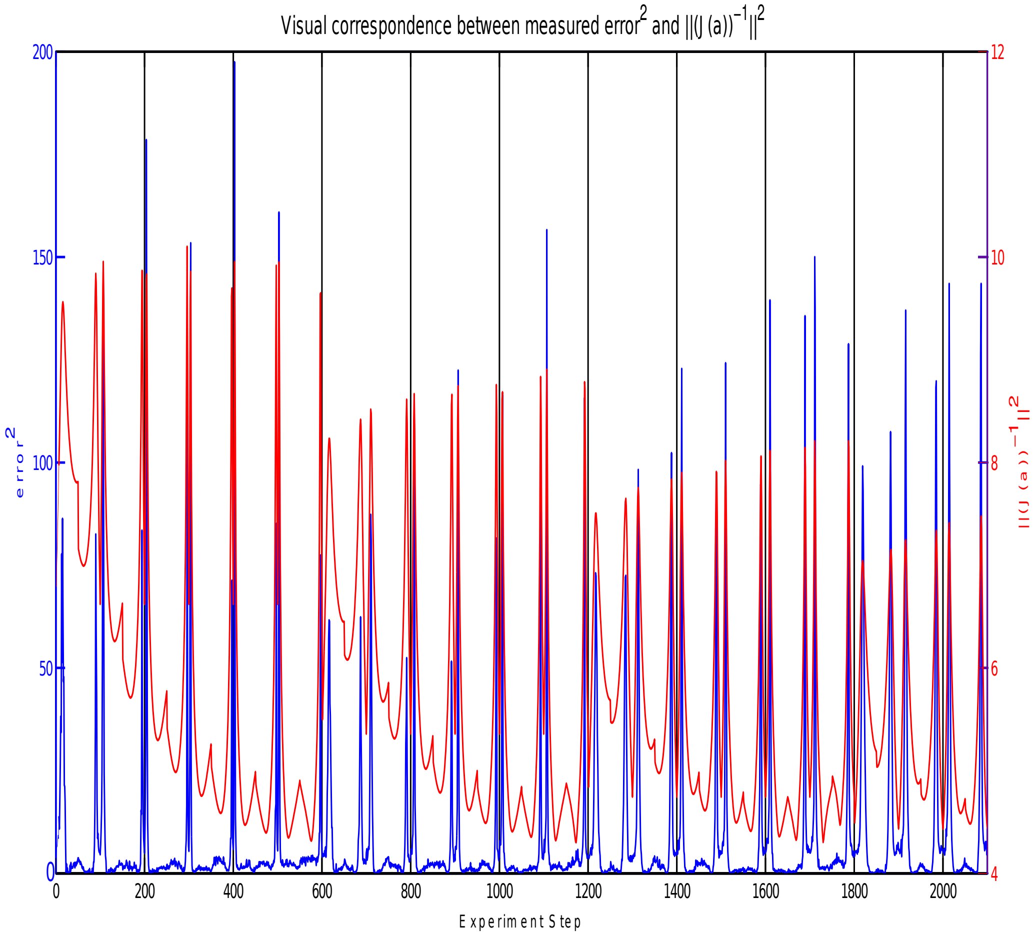


Figure 3.39: Calculated error between the measured configuration of the elastic rod using a force-torque sensor and the modeled configuration. This error plotted on the same horizontal axis as $\| J^{-1}(1, a) \|$ for a path planned in $\mathcal{A}$.

next chapter, we present a cost function and a motion planning algorithm to move between configurations in a manner that ensures the metal strip does not exhibit large error, but remains close to the predicted configuration.

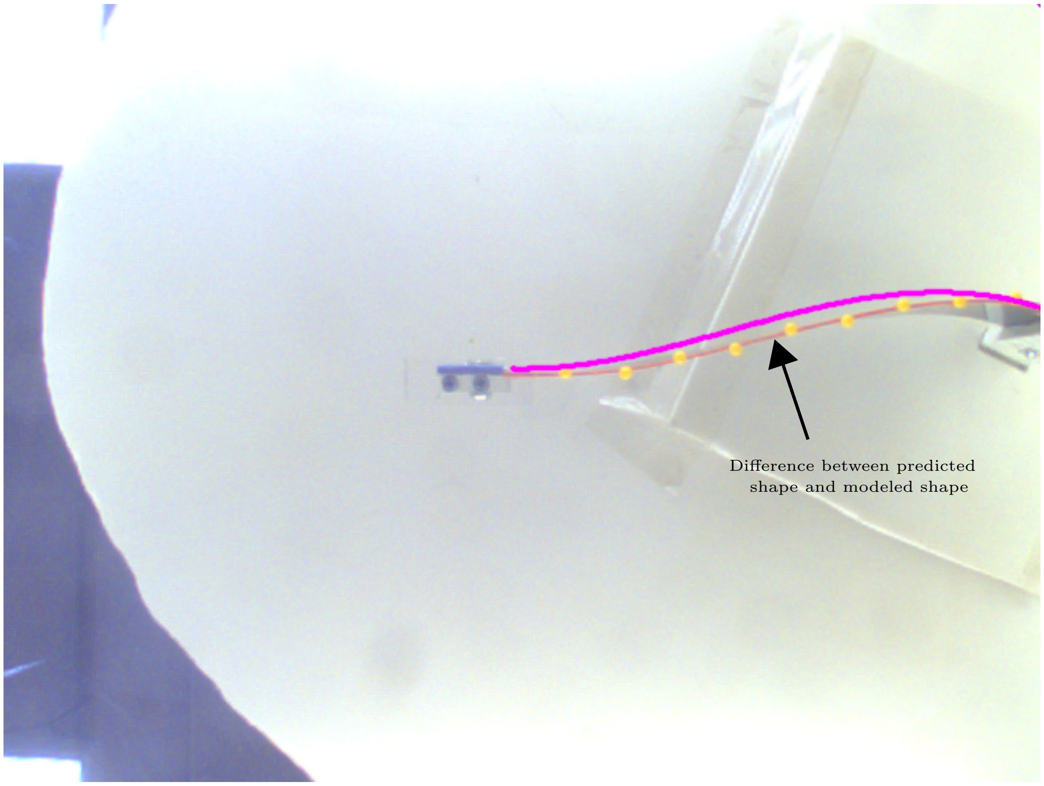


Figure 3.40: Comparison between the predicted and observed configuration of the metal strip. The two shapes have a noticeable deviation.

CHAPTER 4

PLANNER

In this chapter we will describe an extension to the path-planning algorithm presented in Section 3.3 that ensures that the metal strip will avoid configurations that could be infeasible with hardware experiments due to small amounts of error. While our hardware experiments are conducted with a planar metal strip, the cost function we present directly extends to other devices such as flexible cables. First, we provide motivation by explaining why simply identifying a feasible path can be problematic. We then introduce a cost function that not only avoids infeasible configurations, but also avoids configurations where the rod is sensitive to small error in robot placement. This function also incorporates costs that avoid configurations that are close to collisions and joint limitations of the robotics manipulator. We validate, through the use of hardware experiments, that using the cost function to plan paths reduces the error between the predicted and observed configuration.

4.1 Motivation

In this section, we will provide the motivation behind the cost function that we use to perform planning with a metal strip in a manner that avoids configurations where the predicted configuration deviates from the observed configuration. In fact, with a small amount of error, these configurations can actually be infeasible. For example, our model does not take into account the thickness of the metal strip used for experiments when checking for self-collisions. Due to this assumption, the metal strip may experience a self-collision though our model predicts that the configuration is feasible.

In addition to avoiding configurations that our model predicts are nearly infeasible, we designed our cost function to take into account the sensitivity of the force exerted by the metal strip to small errors in the placement of

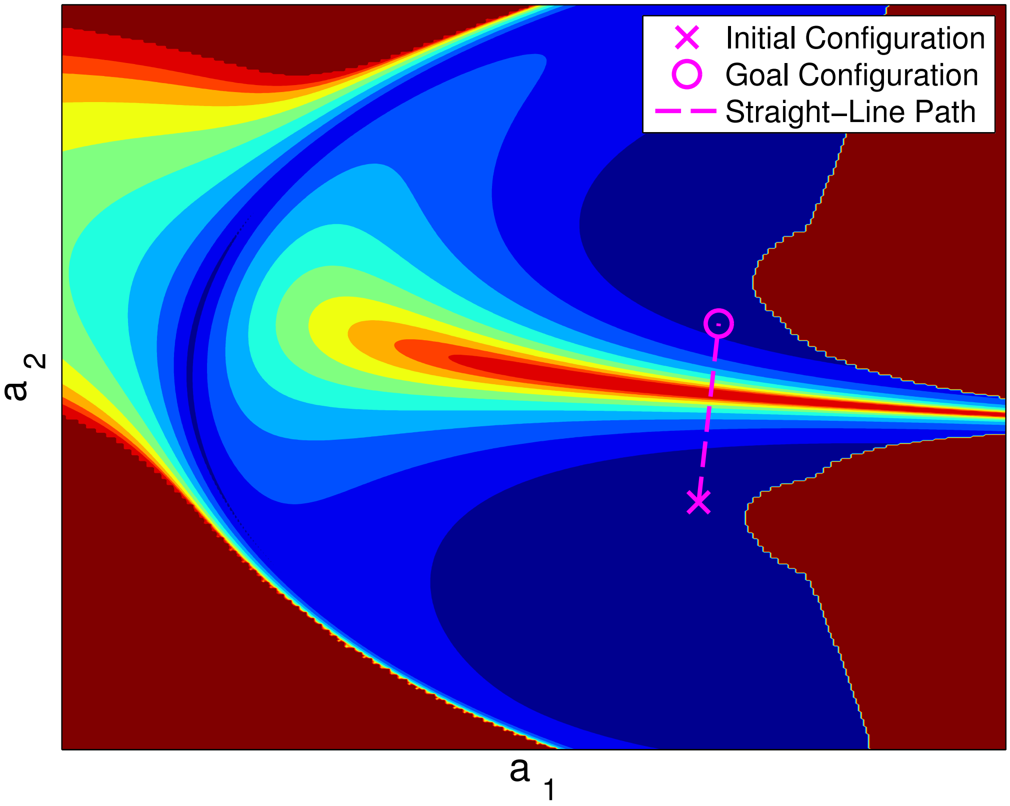


Figure 4.1: Contour of the function $\log(\| (J(1,a))^{-1} \|)$ for $a_3 = -1.0$ along with a line segment between an initial and goal point. This line segment corresponds to a performing manipulation through configurations that are sensitive to small error in the placement of the robot's end effector.

the robotic manipulator. As described in Section 3.3.2, our chart provides for a straight forward calculation of the Jacobian which provides information about the relationship between perturbations in $\mathcal{A}$ and the related change in $\mathcal{B}$ for a particular configuration. This relationship can be used to perform manipulation in a manner that prevents the rod from moving through shapes that are sensitive to the robot placement.

To illustrate the cost function, we show a concrete example where a straight-line path in $\mathcal{A}$ produces a path where the metal strip moves through configurations that are sensitive to error in the placement of the robotic manipulator. Figure 4.1 shows a contour plot of the quantity $\| (J(1,a))^{-1} \|$ for $a_3 = -6$. The blue areas of the plot correspond with when the quantity $\| (J(1,a))^{-1} \|$ is relatively small. On the other hand, the red areas of the plot correspond with when this quantity is large. This plot shows points in $\mathcal{A}$ that corre-

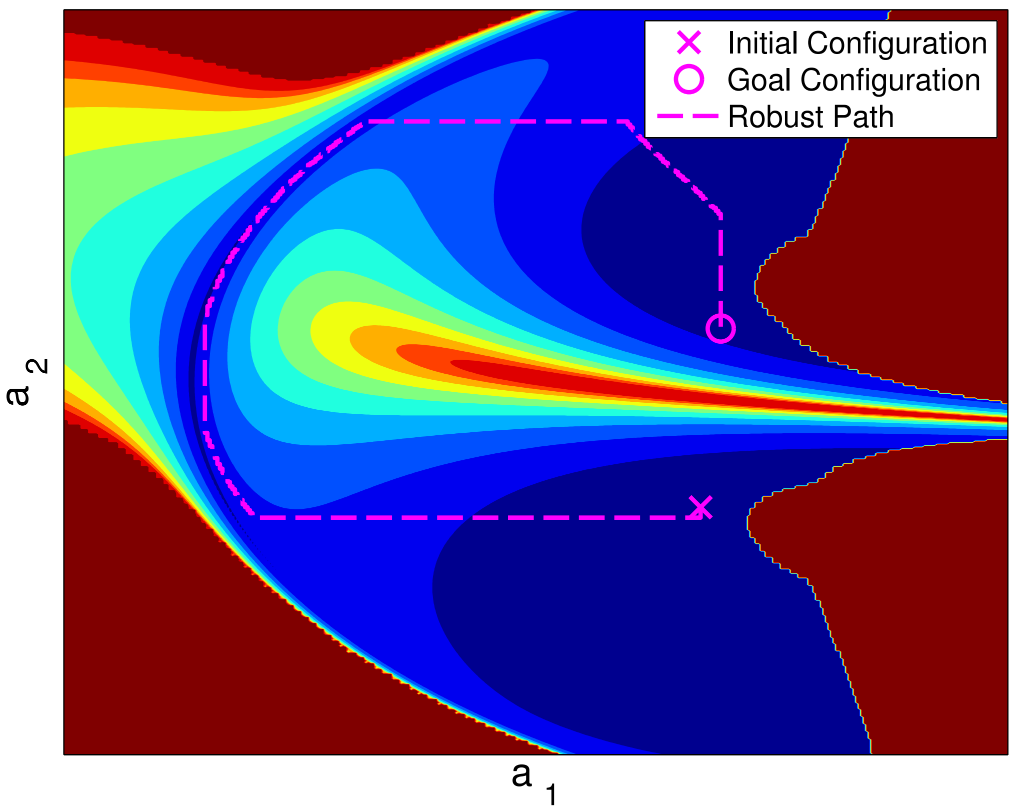


Figure 4.2: This plot shows a contour of the function $\log(\| (J(1,a))^{-1} \|)$ for $a_3 = -1.0$. The path that minimizes the total cost to move between an initial configuration and a goal configuration is displayed on top of the plot of the cost function.

spond with feasible configurations of the metal strip. These two points are connected by a straight-line path that could be produced by our motion planning algorithm. All of the points on this line are in $\mathcal{A}_{\text{free}}$. However, this path moves through a region with a high value of $\| (J(1,a))^{-1} \|$.

In contrast to the motion planning algorithm where two feasible configurations are connected with a straight line, we can perform motion planning in a manner that avoids regions that are undesirable. Figure 4.2 shows the same initial and goal configuration. However, the path between these two points is not a straight line, but consists of curves that move around this region where the value of $\| (J(1,a))^{-1} \|$ is large. While the length of this path, or the total Cartesian distance between these two points, is longer, the total cost is still significantly lower than moving through configurations that are sensitive to error in robot placement. Figure 3.40 shows how the shape of the metal strip deviates from the predicted shape when the configuration is sensitive to the robotic placement. By following a path that avoids these regions of high sensitivity, we can avoid similar deviations.

4.2 Cost Function to Ensure Robust Manipulation

In this section, we describe the three terms of the cost function that are used to plan robust paths to move between two configurations of the metal strip. The first term of the cost function is used to avoid configurations where the metal strip is sensitive to error in the placement of the robotic manipulator. The second two components are used to penalize configurations that are nearly infeasible due to the proximity to a self-collision or the proximity to robotic manipulator reaching a joint limitation.

4.2.1 Cost Due to Sensitivity to Robot Placement

The first term in the cost function avoids configurations that experience instabilities and configurations that are highly sensitive to robot placement. Both of these types of configurations can be identified using the Jacobian matrix, as described in Section 3.3.2. We define the cost due to sensitivity to boundary conditions as

$$\text{cost}_{\text{sensitivity}} = \| \left(J\left(1, a\right)\right)^{-1} \|^2 . \tag{4.1}$$

While this penalizes configurations where the metal strip is sensitive to small changes in the placement of the robotic manipulator, the error term, $\| \ (J(1,a))^{-1} \ \|^2$, naturally increases as the rod approaches an instability. Recall, to test for an instability, we check the determinant along the rod to ensure that $\det(J(1,a)) \neq 0$ along the whole length of the rod. As the determinant approaches 0, the term $\| \ (J(1,a))^{-1} \ \|^2$ approaches ∞. Figure 4.3 shows a contour plot of the cost due to this term. All of the cost plots show the log of the cost to provide more contrast. The infeasible configurations are indicated with red. The dark blue region of this contour plot indicates points in $\mathcal{A}_{\text{free}}$ where the metal strip is less sensitive to the robotic placement. This contour plot also shows a region in the center of $\mathcal{A}_{\text{free}}$ where the cost increases

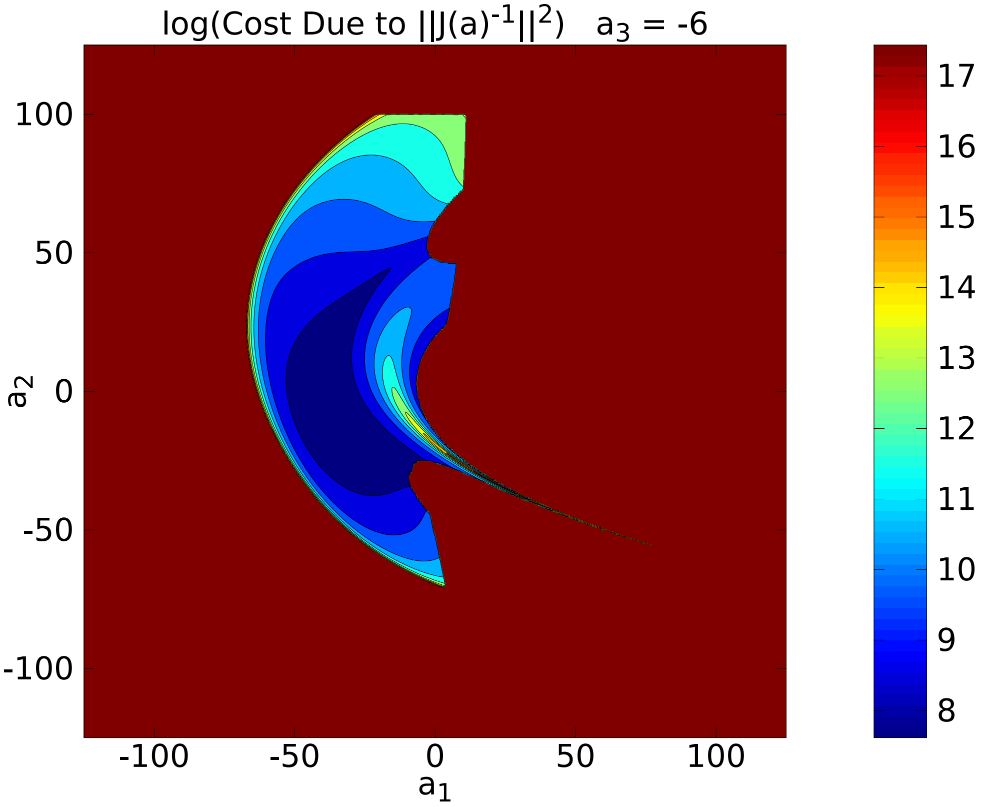


Figure 4.3: Cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -6.0$.

significantly. Within $\mathcal{A}_{\text{free}}$, the cost also increases towards the points along the boundary where our model predicts the rod will experience an instability.

4.2.2 Cost Due to Proximity to Collision

The second term of the cost function is due to the proximity to a configuration in which the metal strip experiences a self-collision. When detecting the proximity to self-collisions, we use a bounding volume hierarchy to detect when two points along the rod approach intersection with each other. To calculate the cost function due to collisions, we divide the rod into 20 segments. A circle of diameter $\frac{1}{20}$ is constructed around each of the selected points. The center of the circle is a point along the predicted shape of the

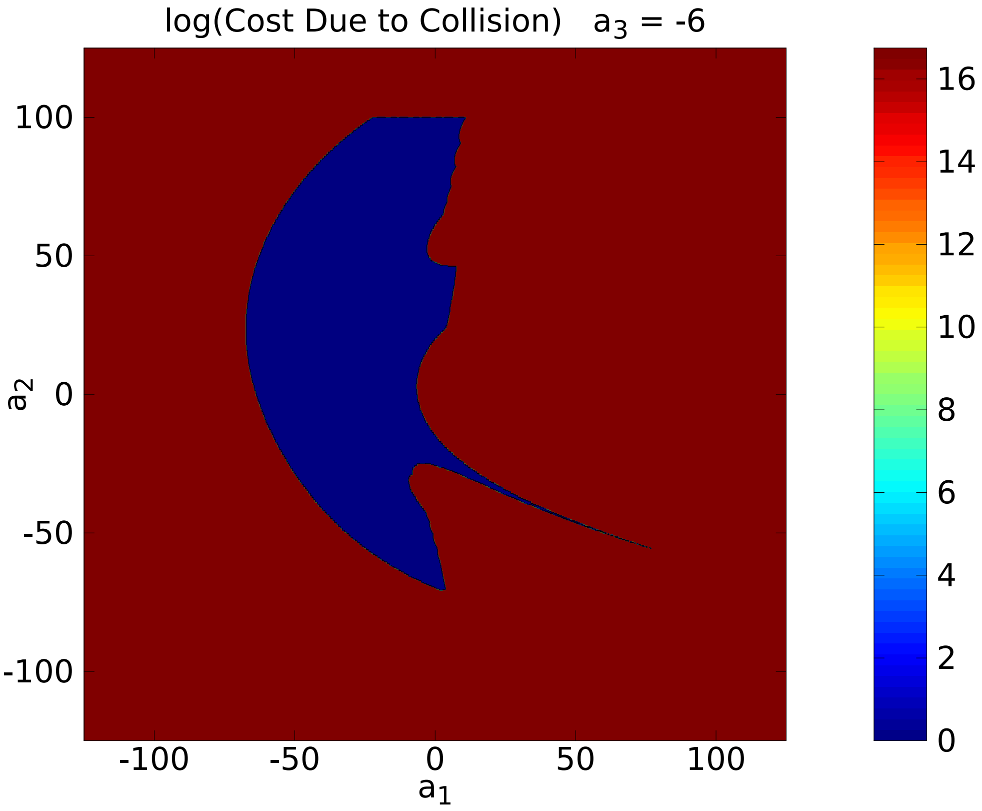


Figure 4.4: This plot shows the component of the total cost due to the proximity to self-collision as defined by Equation (4.2) and Equation (4.3) for the plane where $a_3 = -6.0$.

metal strip. An exhaustive search then determines which two non-adjacent volumes have the smallest distance between their centers. Adjacent portions of the metal strip are naturally in proximity so adjacent volumes are not checked for collisions.

When the distance between the two volumes is less than 0.11 we assess a cost due to proximity to collision using two calculations. The first calculation adjusts the minimum distance to take into account the radius of the bounding volumes. We consider the minimum distance to be

$$\text{adjusted\_min}_{\text{dist}} = max\left(.00001, \text{min}_{\text{dist}} - .05\right). \tag{4.2}$$

This adjustment ensures that the cost function is both positive and finite. When the distance between the center of the two nearest volumes is less than

0.05, or twice the radius of the bounding volumes, the metal strip is considered to experience a self-collision. The second calculation assesses a cost to configurations that are not determined to experience a self-collision, but have volumes with an adjusted minimum distance where adjusted_min$_{\text{dist}}$ $=<$ 0.06. The cost function associated with these configurations is

$$\text{cost}_{\text{collision}} = 5 \times 10^8 \left(\frac{1}{.06 - (.06 - \text{adjusted\_min}_{\text{dist}})} \right)^2 . \qquad (4.3)$$

Figure 4.4 shows a contour plot of the cost due to the proximity to self-collisions for the plane where $a_3 = -6.0$. This plot does not have as much contrast as the plot of the sensitivity to the placement of the robotic manipulator. For very few configurations towards the right boundary of $\mathcal{A}_{\text{free}}$, the plot has green points that indicate that the cost is non-zero and non-infinite.

4.2.3 Cost Due to Proximity to Robot Joint Limits

While the first two terms of the cost function are applicable to all manipulation tasks, the last term of the cost function is required due to constraints imposed by the robotic manipulator. In addition to the cost due to sensitivity to boundary conditions and instabilities, our cost function has a component that is related to the angle at the end of the rod, θ. The orientation of the Adept robot's gripper is limited to $\theta < |\frac{4\pi}{3}|$ by imposing an infinite cost to configurations with a larger magnitude. The cost due to the rotation angle is calculated for configurations with $\theta > \frac{7\pi}{6}$ using the formula:

$$\text{cost}_{\text{rotation}} = 5 \times 10^8 \left(\frac{1}{\frac{4\pi}{3} - \theta} - \frac{1}{\frac{4\pi}{3} - \frac{7\pi}{6}} \right)^2 . \qquad (4.4)$$

Only a small region of feasible configurations are impacted by the cost function for the rotation angle. Figure 4.5 shows the cost due to the rotation. The cost due to the orientation is either ∞ for infeasible configurations or mostly 0 for feasible configurations. A small region towards the right of the feasible configurations is green which indicates that these configurations are feasible, but have an orientation that exceeds $\frac{7\pi}{6}$. For this slice of a_3, and for other slices, only a small region of $\mathcal{A}_{\text{free}}$ correspond to feasible configurations with $\theta < \frac{4\pi}{3}$.

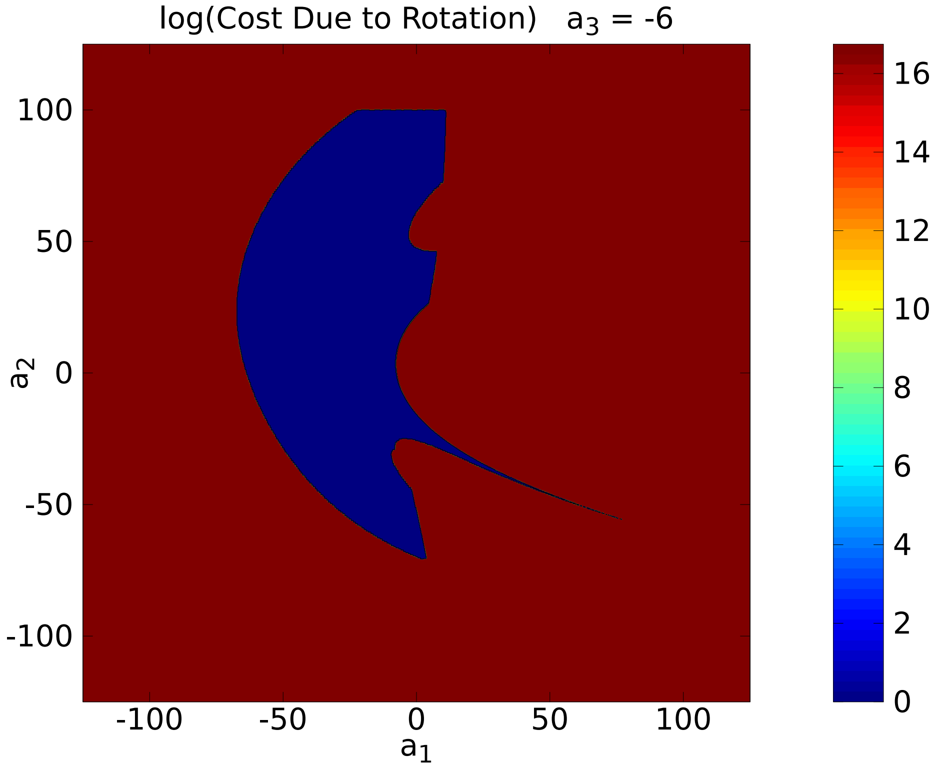


Figure 4.5: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -6.0$.

4.2.4 Total Cost Function

We calculate the total cost function as the sum of the three individual terms such that

$$\text{total}_{\text{cost}} = \text{cost}_{\text{sensitivity}} + \text{cost}_{\text{collision}} + \text{cost}_{\text{rotation}}. \tag{4.5}$$

The total cost for feasible configurations is truncated with a limit of 5×10^7. This is done to ensure that feasible configurations that are extremely close to the boundary do not receive an infinite cost. This could result in motion plans that extend outside of $\mathcal{A}_{free}$ instead of consisting completely of feasible configurations. Figure 4.6 shows a contour of the total cost function. This function largely resembles the cost due to the sensitivity of the configuration to error in robotic placement. As seen with the plots of the individual terms of the cost function, the second two terms of the cost function are either

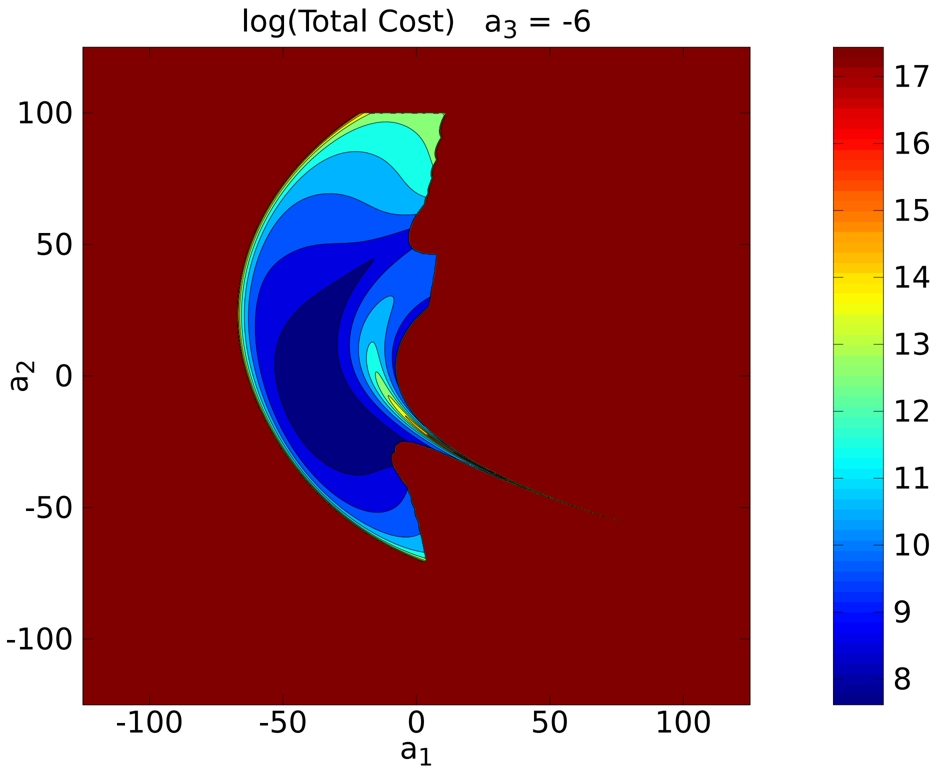


Figure 4.6: Total of all three components of the cost function Equation (4.1), Equation (4.3), and Equation (4.4) for the plane where $a_3 = -6.0$.

0 or infinite for the majority of configurations. The plots of the total cost function, along with plots for each of the components of the cost function, can be found in Appendix B.

4.3 Experimental Validation of Reduced Error Using Minimal Cost Paths

The purpose of this set of experiments is to contrast the error seen when performing motion planning using the cost function presented in this chapter to motion planning that only ensures that configurations are in $\mathcal{A}_{\text{free}}$. Hardware experiments in the previous chapter validated that our algorithm successfully plans a path to move from an initial configuration to a goal configuration. However, this planning algorithm was also shown to move the metal strip through configurations where the observed strip deviates from the our prediction. The hardware experiments reported in this chapter show the extent to which planning using the cost function reduces the error experienced when performing manipulation.

4.3.1 Procedure

Initial and Goal Configuration Selection

To validate the ability to plan paths that minimize sensitivity to small error in robot placement, 30 experiments were performed to compare straight-line paths to paths that minimized the total path cost using our cost metric. To best demonstrate the benefit of using the robust planner, 30 configurations were selected from within $\mathcal{A}_{\text{free}}$ where $\| \ (J\,(1,a))^{-1} \ \|^2 > 1 \times 10^6$, is larger than 1×10^6. Two configurations that are collinear with the initial high-cost point are then selected where $\| \ (J\,(1,a))^{-1} \ \|^2 < 1.5 \times 10^4$. All configurations that are on the straight-line path in $\mathcal{A}$ between the two configurations are checked to ensure feasibility. After this straight-line path is identified, the LRM* algorithm is used to identify a path that minimizes the total cost between these two configurations.

Planner Implementation

The ability to represent all feasible configurations for an elastic rod with just three parameters can be leveraged to decrease the amount of time needed to search the state space for a path of minimal cost. In addition to being of low state space, we have found that feasible values for each of the three dimensions (a_1, a_2, a_3) are bounded. The LRM* implementation requires time to sample the feasible configuration space and generate data structures with this sampled information. However, subsequent paths can be determined faster than single-query approaches. We have implemented a lattice roadmap [38] that creates a 3D lattice that samples the cost function at intervals of 0.2 in each dimension. This results in an array of size 1251x1251x81. Though the sampling of the state space is deterministic and not random, this representation provides a few key benefits. The lattice structure allows for a simple data structure that allows for constant data access time. Instead of requiring a structure that requires dynamically allocated memory, this lattice structure can be stored in a 3D array. Array cells that are within one cell of a particular cell can be accessed with a simple mathematical operation that requires only addition and multiplication. This regular structure also allows for scalable search time for determining optimal paths. Instead of checking the neighbors within the data structure ($[x-1, y, z][x+1, y, z][x, y+1, z][x, y-1, z][x, y, z-1][x, y, z-1]$), we can trivially increase the distance to "neighbor" cells ($[x-10, y, z][x+10, y, z][x, y+10, z][x, y-10, z][x, y, z-10][x, y, z-10]$). This property allows for developing more coarse motion plans that are not as optimal, but can be generated with less time. Once this data structure is constructed, searches to identify paths that minimize the total cost can be performed using Dijkstra's algorithm or A* algorithm.

While this implementation does require significant up-front calculation to construct (approximately 240 hours with Intel Core i5 processor), it provides some key benefits. Searches using Dijskstra's algorithm of this structure to find optimal paths through this graph can be conducted faster than other algorithms. The LRM* implementation completed searches for paths in less than 10 seconds when the interval size was 1 and less than 30 minutes when the interval size was set to .2.

Path Execution

After the LRM* algorithm has generated an optimal path to follow to move from the initial configuration to the goal configuration, a straight-line path is planned to move back to the initial configuration. For each point along the path which includes both the robust path and the straight-line path, the position and orientation are calculated using Equations (3.7) - (3.10). The Adept robot is initialized at the starting configuration and the robust path is used to move to the goal configuration. Figure 4.7 shows a plot of the predicted values for a_1, a_2, and a_3 for the entire experiments shown in Figures 4.1 and 4.2. The vertical black line in Figure 4.7 indicates when the goal configuration was reached using the robust planner. The remaining 400 steps show the path planned back to the initial configuration. The cost due to sensitivity is also provided in Figure 4.7 to show how the minimum

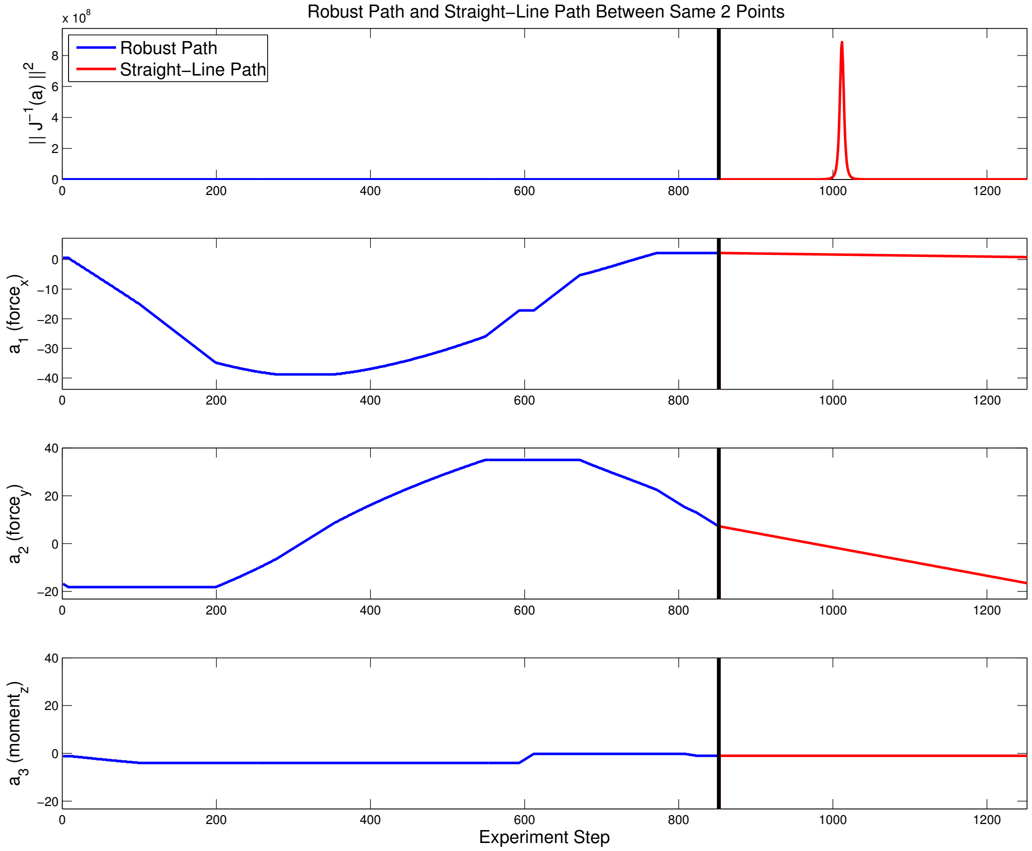


Figure 4.7: Results of experiments that compare straight-line-paths in $\mathcal{A}$ to paths that minimize the total cost to move from an initial configuration to a goal configuration. The first part of the path moves from starting configuration to goal configuration using the robust planner. A straight-line path is used to move back to the goal configuration. The sensitivity to boundary conditions is also shown for this entire experiment.

cost path is significantly less sensitive than the straight-line-path. Figure 4.8 shows a 3D plot of this same experiment to help contrast the two paths in $\mathcal{A}$ to move between the two configurations. As with Figure 4.7, the red line shows the straight-line path and the blue line shows the robust path that was calculated using the LRM* algorithm.

For these experiments, the force sensor is used to measure the force exerted by the metal strip against the robot. The camera is also used to capture images for each configuration after the robot has completed movement to the desired pose. We use the same error metric given in Equation (3.16) to compare the robust paths to the straight-line paths.

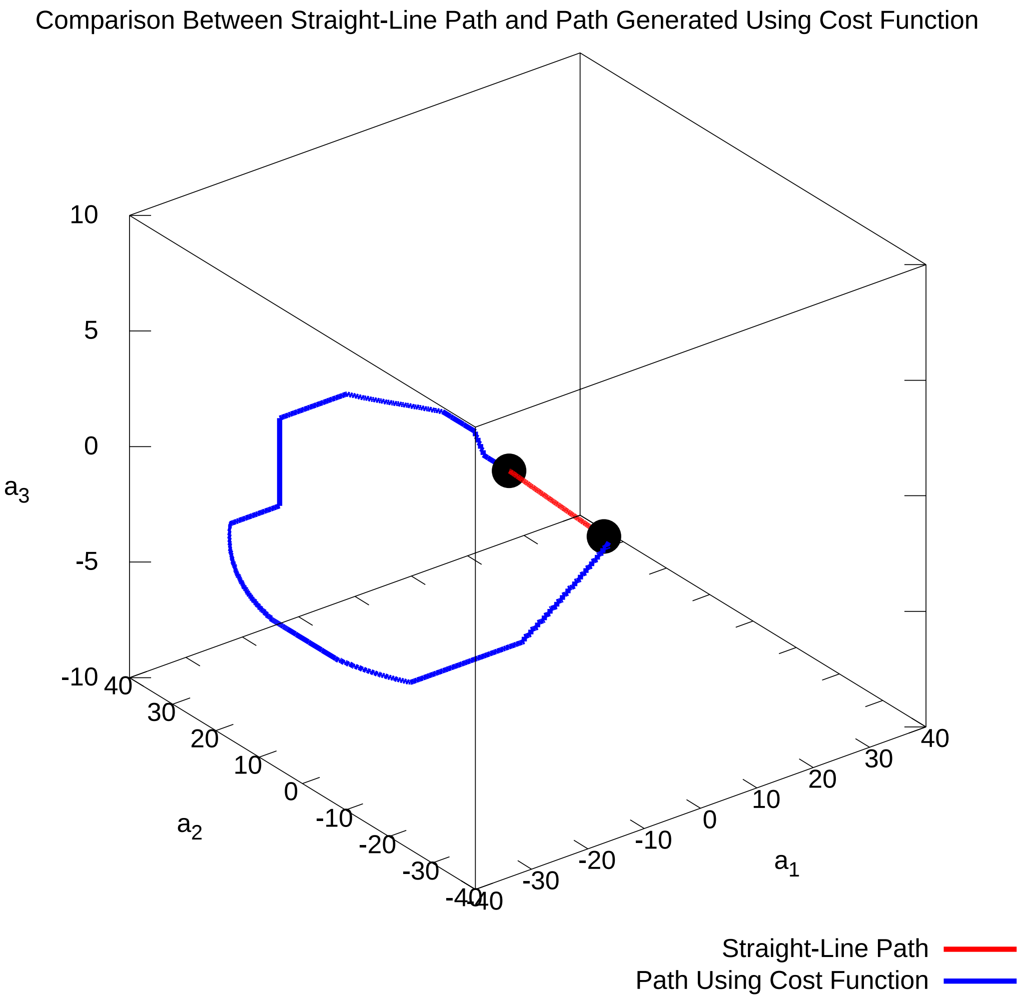


Figure 4.8: Complete path in $\mathcal{A}$ for an experiment. The two circles indicate the initial and goal configuration. The blue line shows the robust path for manipulating the metal strip. The straight red line shows the path followed to move back to the initial configuration.

4.4 Results

The hardware experiments conclusively show that using the cost function significantly reduced the error between the observed configuration and the predicted configuration. Similar results are seen for several of the experiments comparing straight-line-paths in $\mathcal{A}$ to minimum cost paths. Figures 4.9 - 4.18 show the results from ten experiments. Additional results can be found in Appendix C. These plots show a comparison between the predicted and observed configuration for each of the three dimensions of $\mathcal{A}$. These plots also plot the error and the value of our cost function for each step in the experiment. The results in Table 4.1 show both the median and maximum $error^2$ value seen during each experiment. Figure 4.19 graphically shows the data contained within this table. While some of the straight-line paths have maximum $error^2$ values less than 20, with some experiments, this quantity is larger than 6×10^4. These deviations correspond to large deviations between the predicted and observed force.

Path planning using minimum cost paths does come with some potential drawbacks. First, these paths are longer than the straight-line-paths. This is both in terms of distance traveled in $\mathcal{A}$ and also the total distance the robot moves. While the planned paths do involve a larger range in forces exerted by the robotic manipulator, these forces are smaller than the observed forces using straight-line-paths. These paths do sometimes move close to configurations that are near instabilities. This proximity to instabilities sometimes causes the shape of the rod to deviate from the predicted configuration. Despite this error in shape, the observed value of a is still smaller than when moving along a straight-line-path. Second, the determination of minimum-cost-paths does require a computation cost. Unlike the straight-line-paths where the robot placements can be calculated in a trivial amount of time, more time is necessary to calculate minimum-cost paths.

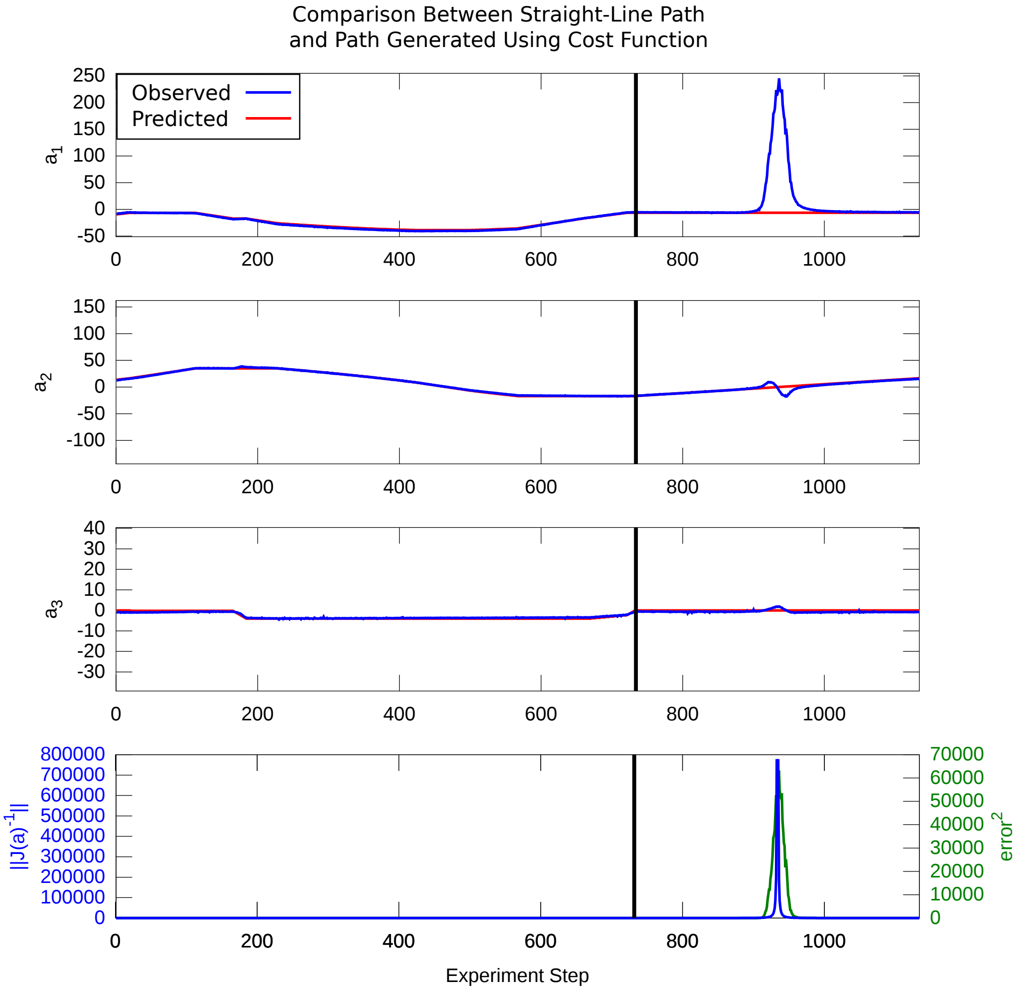


Figure 4.9: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-6.4, 16.6, 0.0]$ and $[-6.4, -16.6, 0.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

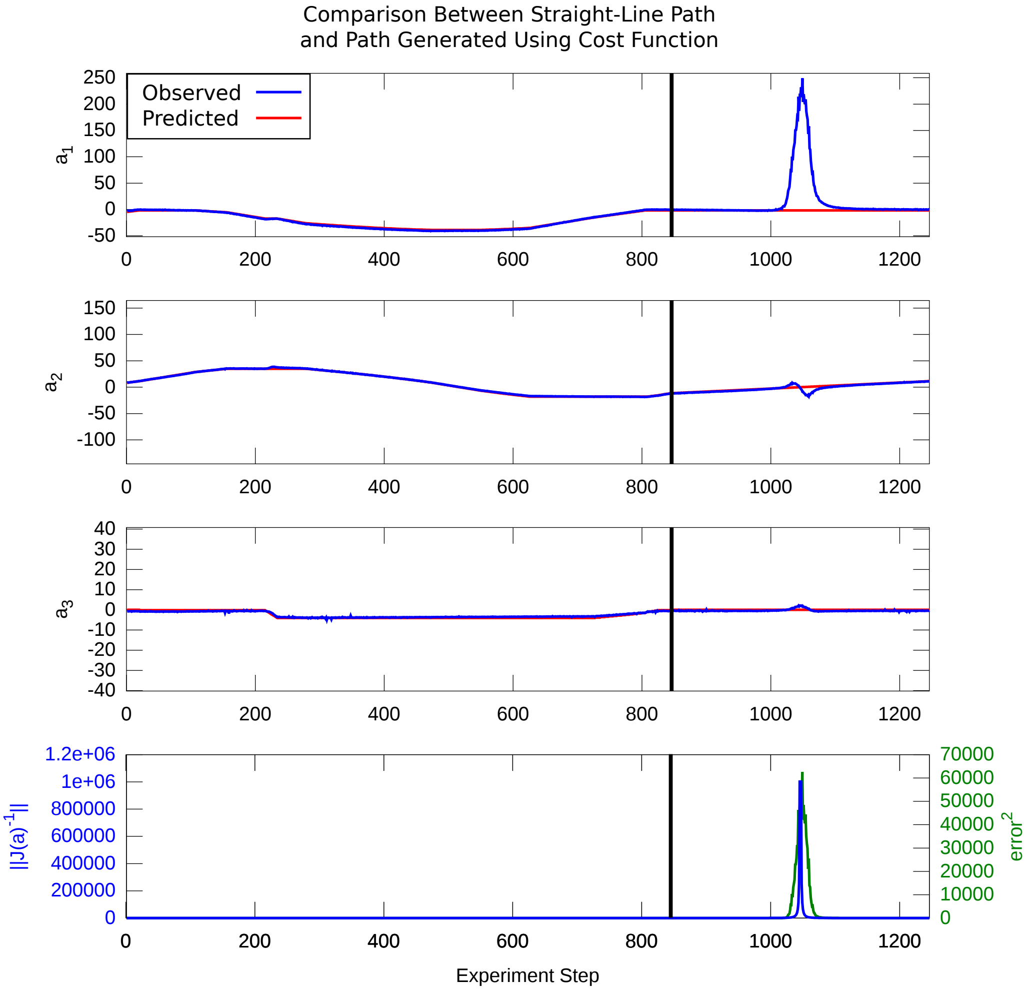


Figure 4.10: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-1.6, 11.4, 0.0]$ and $[-1.6, -11.4, 0.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

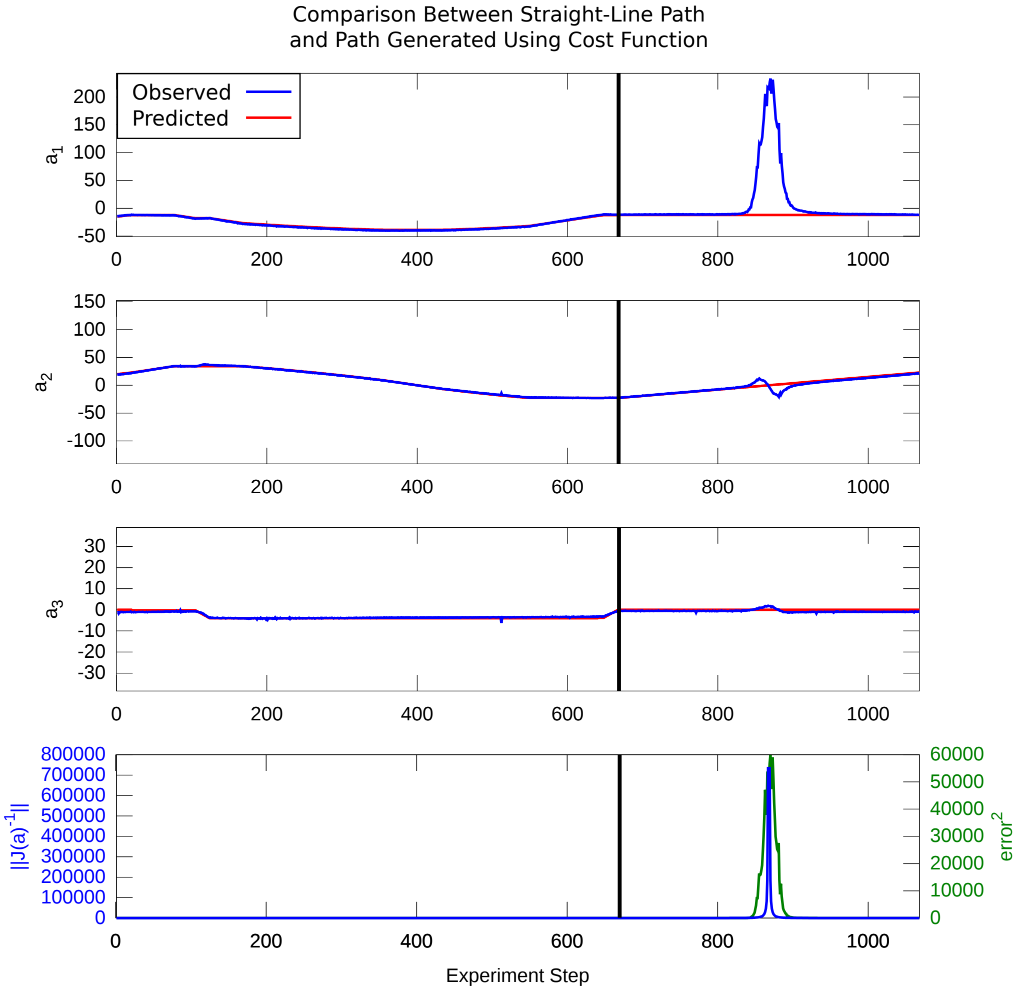


Figure 4.11: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-12.0, 22.8, 0.0]$ and $[-12.0, -22.8, 0.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

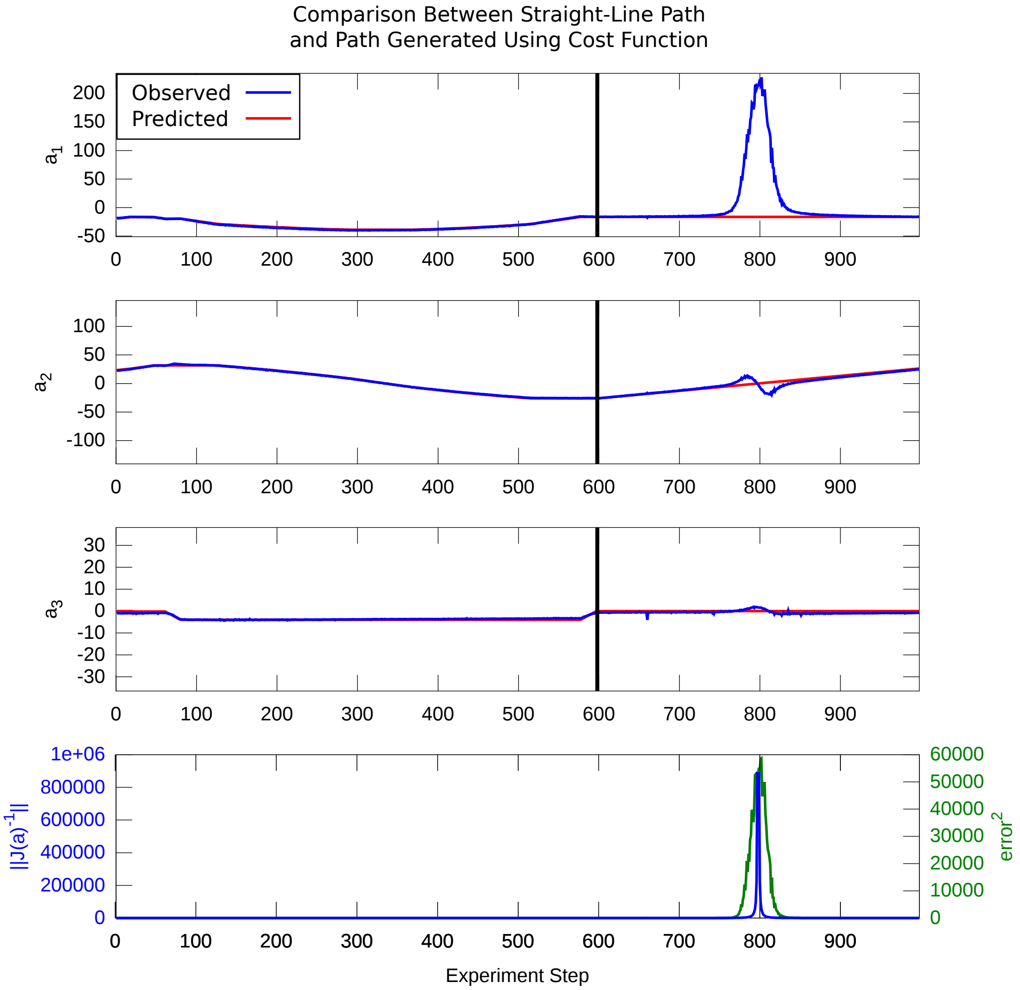


Figure 4.12: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-16.4, 26.2, 0.0]$ and $[-16.4, -26.2, 0.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

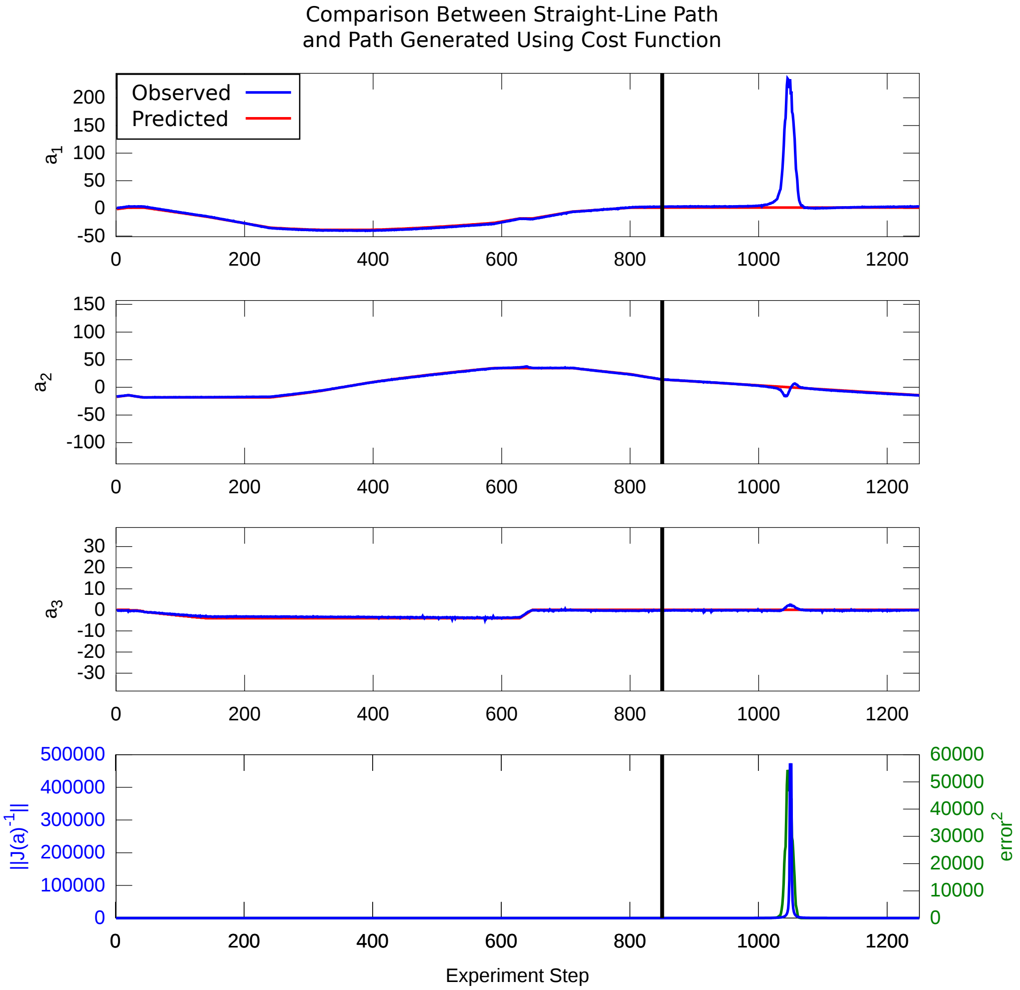


Figure 4.13: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[1.6, -14.2, 0.0]$ and $[1.6, 14.2, 0.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| (J(1,a))^{-1} \|^2$ (blue).

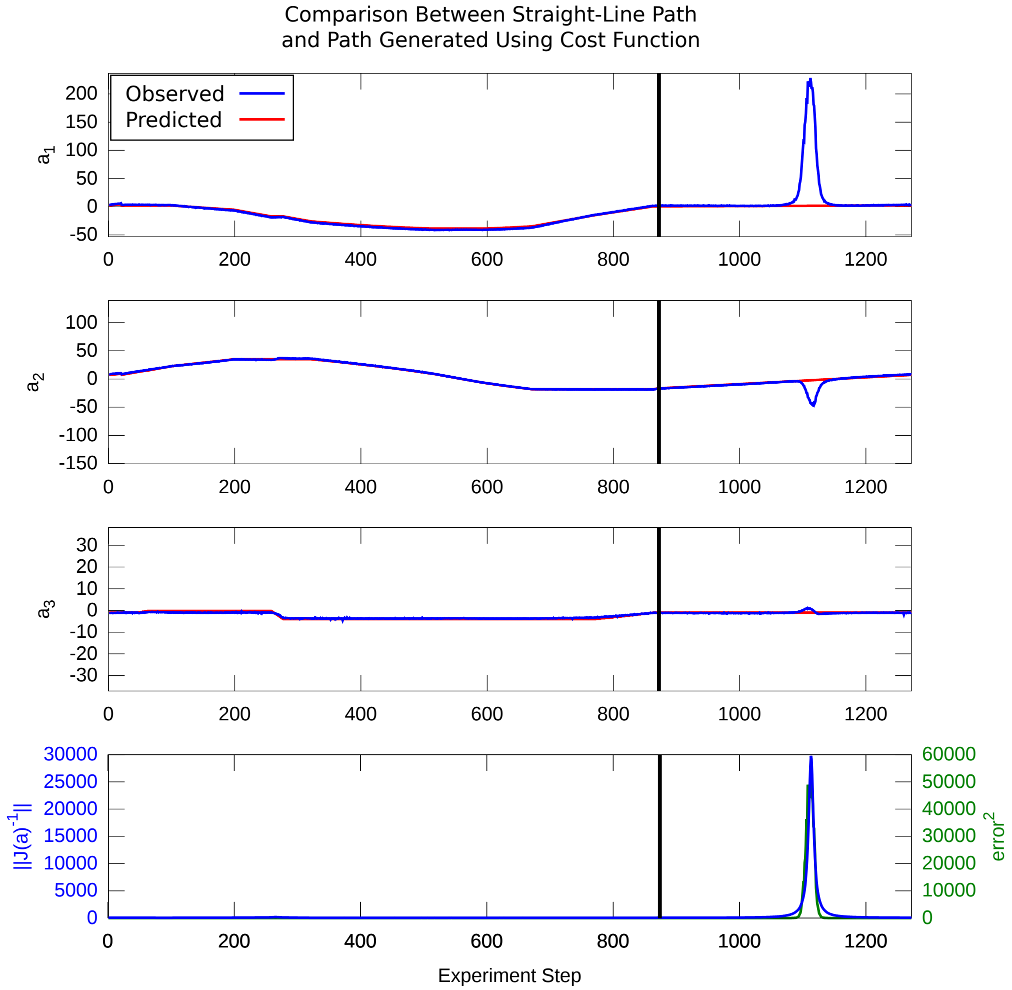


Figure 4.14: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[2.2, 7.2, -1.0]$ and $[0.8, -16.6, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

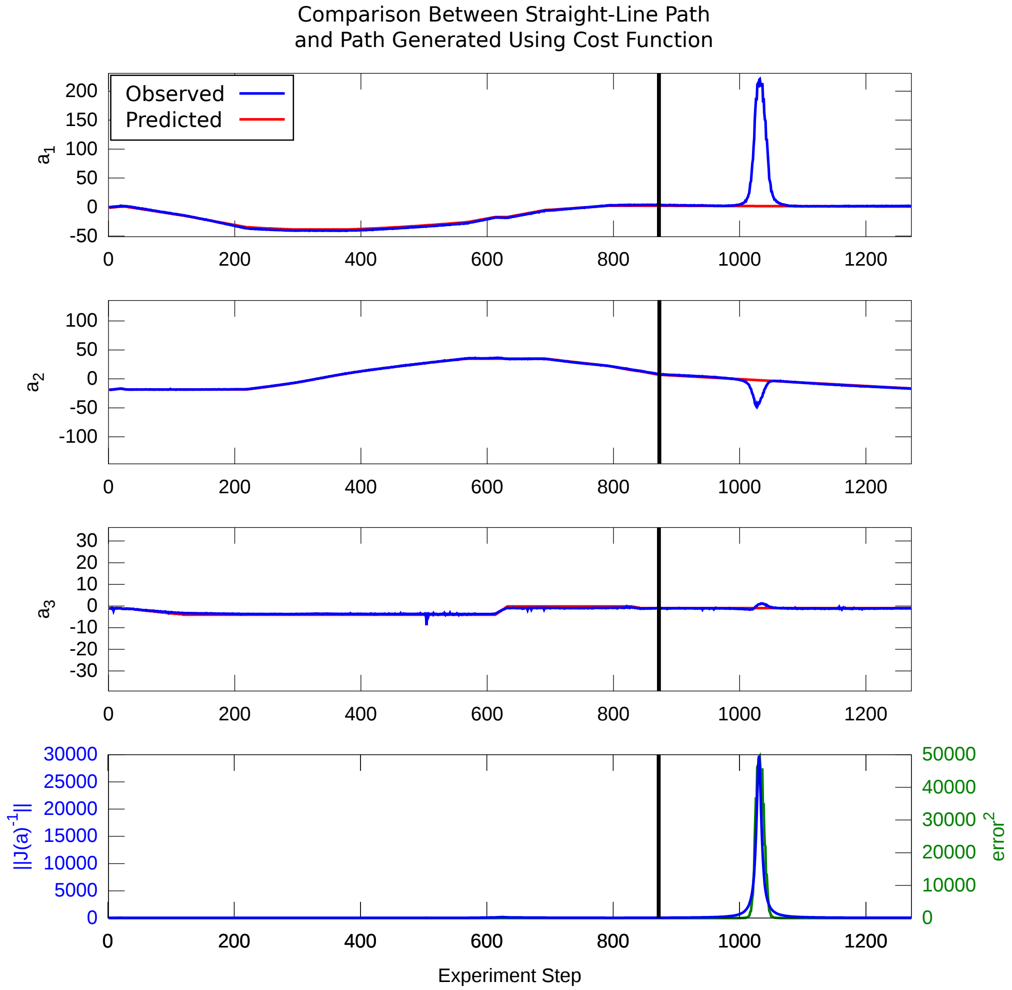


Figure 4.15: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[0.8, -16.6, -1.0]$ and $[2.2, 7.2, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

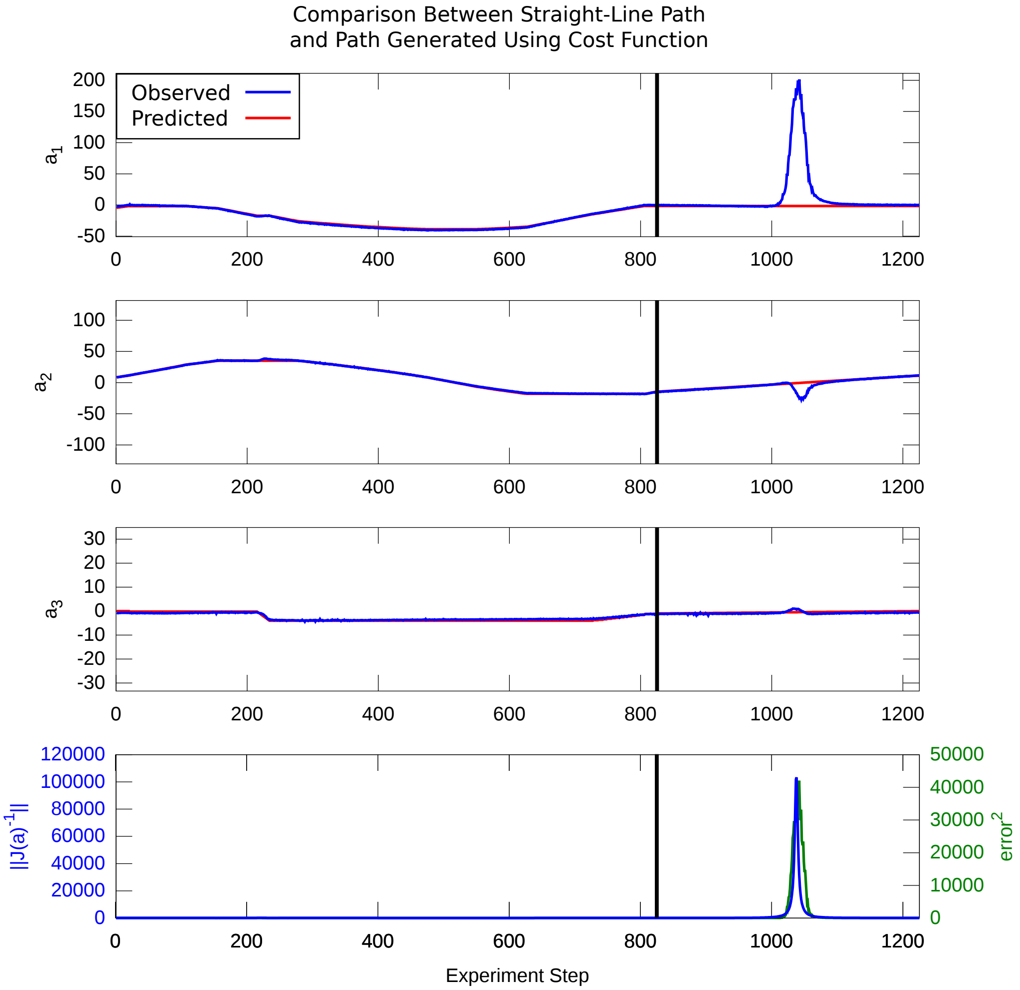


Figure 4.16: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-1.6, 11.4, 0.0]$ and $[-1.6, -14.6, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

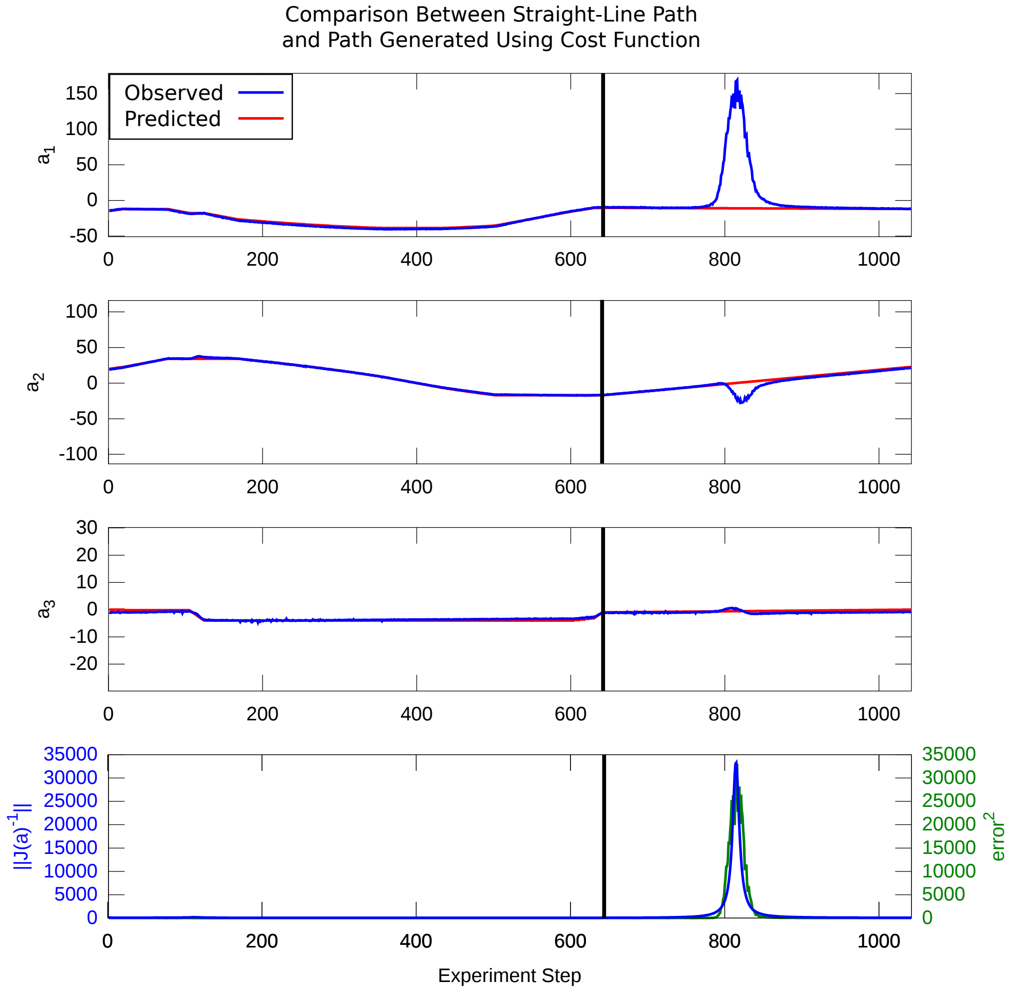


Figure 4.17: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-12.0, 22.8, 0.0]$ and $[-10.4, -17.0, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

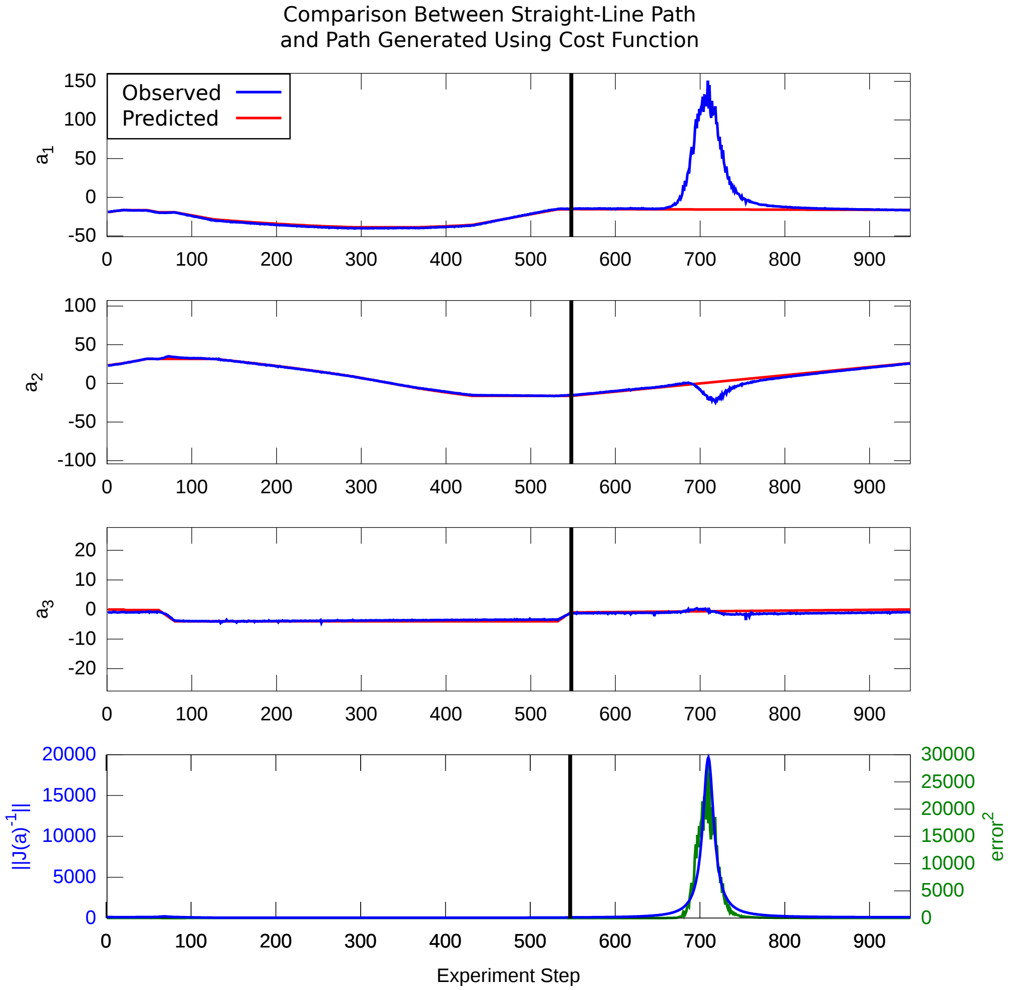


Figure 4.18: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-16.4, 26.2, 0.0]$ and $[-15.4, -16.2, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

Table 4.1: Comparison of the median and maximum $error^2$ value for 30 experiments that used both the robust planner and a straight-line path in $\mathcal{A}$ to move between two configurations.

	Robust Planner		Straight-Line Paths	
	Median$_{\text{err}^2}$	Max$_{\text{err}^2}$	Median$_{\text{err}^2}$	Max$_{\text{err}^2}$
1	5.13	12	5.84	63200
2	3.78	13.1	4.24	62600
3	7.31	21.2	12.9	59900
4	8.5 1	14.7	4.7	58600
5	3.27	9.85	3.87	54400
6	15.5	28.5	4.85	52300
7	3.76	106	1.61	49500
8	30.8	43.3	32.5	42000
9	20.9	30.8	30.3	33000
10	19.8	29.8	23.9	28300
11	8.28	17.12	4.18	14640
12	18.41	26.36	7.77	10860
13	8.5	19.17	2.68	8463.5
14	10.21	18.39	1.84	6318.6
15	12.32	19.59	3.36	5220.3
16	10.259	21.768	3.6837	5113.2
17	7.6119	22.642	3.9306	4932.9
18	12.237	27.538	3.4528	4034.2
19	17.183	27.537	15.26	3621.4
20	19.64	30.458	15.709	2548.5
21	10.378	18.676	2.6024	2418.7
22	12.517	23.225	1.9009	2035.5
23	11.725	25.087	2.7445	1808.2
24	11.785	35.273	2.7105	1128.8
25	11.044	19.732	1.4762	947.57
26	12.309	21.511	3.3374	586.9
27	21.261	30.05	20.156	454.03
28	10.751	20.08	3.2406	441.04
29	13.799	23.971	7.7549	420.91
30	25.043	36.089	26.871	122.03

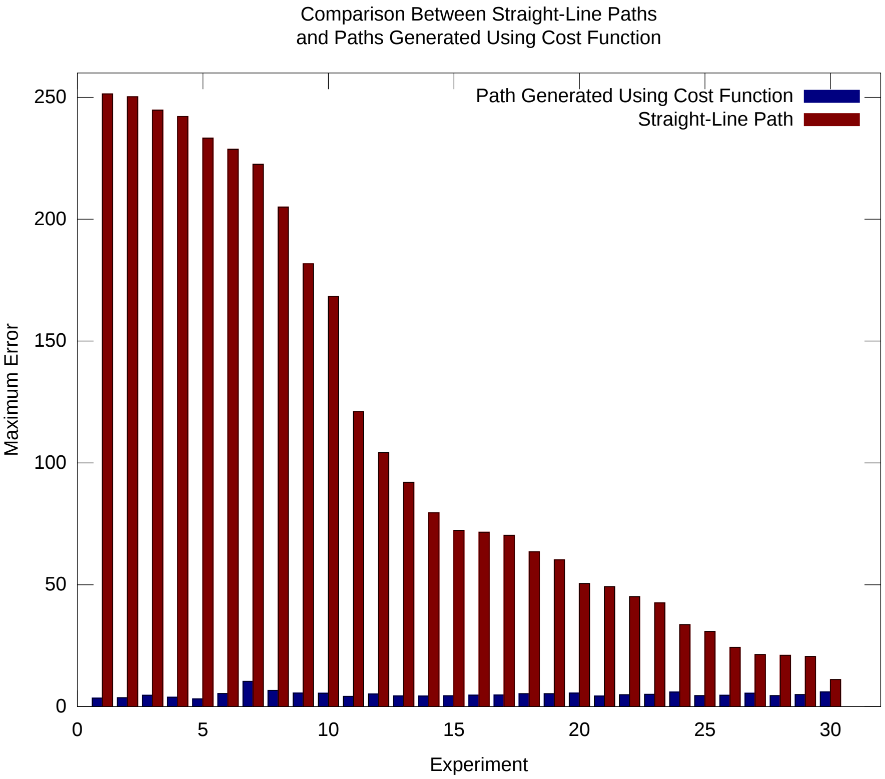


Figure 4.19: Comparison between the maximum error observed when following straight-line paths and the maximum error observed when following paths generated using our cost function.

Path planning using the cost function to identify minimum cost paths prevents the rod from experiencing large amounts of error. This error translates to a large deviation between the predicted and observed force exerted by the robotic manipulator. Recall from Section 3.2.4 that the observed value of a is the observed force applied at the end of the rod. When the rod experiences a large deviation from the planned value for $\mathcal{A}$, the rod is exerting a force that varies from the expected value. Figure 4.20 shows the force measured by the robotic manipulator for an experiment where the cost function was applied to identify a robust path between two configurations and a straight-line path was used to return to the original starting configuration. For this plot, only the magnitude of the f_x and f_y are shown and not the torque, m_z.

With this experiment, the graph comparing the observed and predicted forces shows that the planned force with the robust path is larger than the maximum that was planned for the straight-line path. However, the rod experiences a large deviation in force with the straight-line path. The maximum force experienced at the end of the rod when using a straight-line path exceeds 30 N. This result gives cause for concern. When performing manipulation tasks, the force exerted against the robot is more than 15 times larger than our model suggests. The maximum deviation between observed force and predicted force experienced when using the robust planner is less than 1 N compared with more than 30 N seen with the straight-line path.

Using the cost function to determine minimum cost paths does not remove all of the error. As shown in Table 4.1, both the median error^2 and maximum error^2 values for the robust planner often exceed 8. In fact, the median error for the robust planner is larger than the median error for the straight-line-paths. Though the error is greatly reduced using the cost function to plan paths, additional techniques can further reduce the observed error.

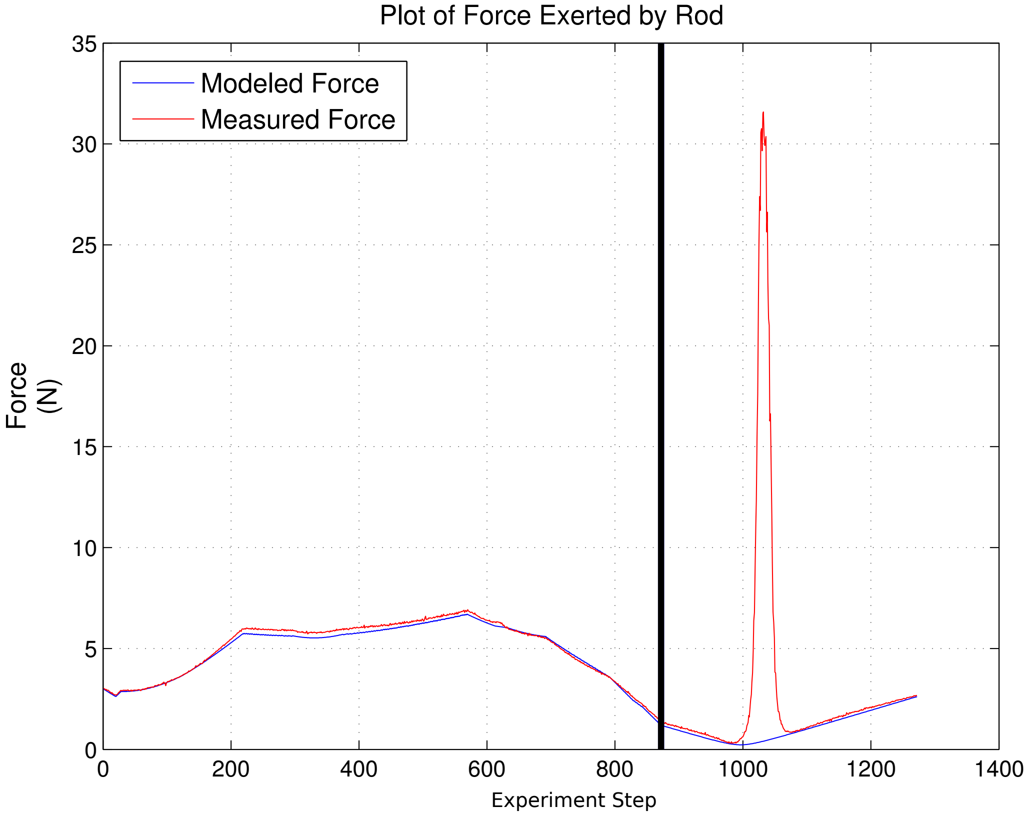


Figure 4.20: This plot shows a comparison between the predicted force and observed force when performing an experiment that uses the cost function to plan a path from a start configuration to a goal configuration and uses a straight-line path in $\mathcal{A}$ to return to original start configuration.

CHAPTER 5

CONTROLLER

In the previous chapter, we presented a cost function that can be used to plan paths for a robotic manipulator that ensure that the metal strip avoids configurations that are either nearly infeasible or sensitive to small amounts of error. We uniformly sampled the entire region of $\mathcal{A}_{\text{free}}$ to calculate the cost for each configuration and then completed experiments to contrast the paths that minimize the total cost to straight-line paths. The experiments conducted validate that performing motion planning using this cost function significantly reduced the error between observed configurations and predicted configurations.

In this chapter, we present the design of a closed-loop controller that utilizes feedback from sensor measurements to manipulate of a metal strip in a manner that reduces the amount of observed error. The controller serves as a compliment to the motion planning algorithm shown in the previous chapter. First we describe a method for measuring the configuration of the metal strip using a position sensor and then present an estimator that combines two different types of measurements of the elastic rod's configuration in. Next, we describe the implementation of the feedback controller that incorporates our estimator and provide experimental results that compare the error observed when performing open-loop manipulation to the observed error when utilizing the controller.

5.1 Estimator

In Chapter 3, we demonstrated the ability to estimate the configuration of the strip by measuring force and torque at one end. In this section, we demonstrate the ability to estimate the configuration of the strip by measuring the position of a finite number of points along its length. While the position sensor usually provides the best measurement of the metal strip's configuration, this is not the case when the configuration is sensitive to error in the robotic manipulator placement. The estimator we have designed combines both types of sensor measurements in a manner that weighs the force sensor more heavily when our model predicts the position is less accurate.

5.1.1 Measurement of a_{position}

We have previously shown a method for recovering the configuration from a the position of points along the metal strip [41]. We denote the position by $(\bar{X}, \bar{Y})$ and this orientation by $\bar{\theta}$, and let $\bar{x} = \bar{X}/L$ and $\bar{y} = \bar{Y}/L$. We assume that a camera has a line of sight. This camera is orthogonal to the plane in which the rod is deforming and is placed at a fixed distance from this plane. Let B be a linear transformation that takes points in the image plane of the camera and converts them to points in the world frame and then scales these points by L. After applying this transformation to the rod observed by the camera, the resulting rod will have unit length.

The metal strip is in some configuration (q, u) corresponding to some $a \in \mathcal{A}$. Thus determining the configuration (q, u) of the rod (which is an infinite-dimensional curve) is equivalent to determining the finite-dimensional value of a, and we denote this dependence by $(q(t, a), u(t, a))$.

We assume that fiducial markers are placed at known locations along the rod. Assume that n markers are located at distinct positions along the rod, given by $0 < L_1 < L_2 < ... < L_n < L$, and let $l_j = L_j/L$. Let $(\tilde{X}_j, \tilde{Y}_j)$ be the location of the jth marker observed in the image plane, and let $(\tilde{x_j}, \tilde{y_j}) = B((\tilde{X}_j \tilde{Y}_j))$, i.e. $(\tilde{x_j}, \tilde{y_j})$ is the position of the jth marker in the plane of the rod scaled by the length of the rod. For any $a \in \mathcal{A}$, let $x(t, a)$ and $y(t, a)$ be the position coordinates and let $\theta(t, a)$ be the orientation of point t along the configuration $(q(t, a), u(t, a))$.

Define $F(a)$ to be

$$F(a) = \begin{bmatrix} x(l_1, a) - \tilde{x}_1 \\ y(l_1, a) - \tilde{y}_1 \\ . \\ . \\ . \\ x(l_n, a) - \tilde{x}_n \\ y(l_n, a) - \tilde{y}_n \\ x(1, a) - \bar{x} \\ y(1, a) - \bar{y} \\ \theta(1, a) - \bar{\theta} \end{bmatrix} \tag{5.1}$$

If the configuration corresponding to a matches the observed configuration exactly, then we will have $F(a) = 0$. We now have a system of $2n + 3$ nonlinear equations. If we let $\hat{\mathbf{J}}(a)$ denote the Jacobian of $F(a)$, we can use the Newton-Raphson method to recursively approximate a using Equations (5.2) and (5.3). For the experiments we present in this thesis, a_{force} is the initial configuration for this method.

$$\Delta a^{(i)} = (\hat{\mathbf{J}}(a^{(i)})^T \hat{\mathbf{J}}(a^{(i)}))^{-1} \hat{\mathbf{J}}(a^{(i)})^T F(a^{(i)}) \tag{5.2}$$

$$a^{(i+1)} = a^{(i)} - \Delta a^{(i)} \tag{5.3}$$

All that remains is to find $\hat{\mathbf{J}}(a)$. The k^{th} column of $\mathbf{J}(t)$ is

$$\begin{bmatrix} \frac{\delta x(t,a)}{\delta a_k} & \frac{\delta y(t,a)}{\delta a_k} & \frac{\delta \theta(t,a)}{\delta a_k} \end{bmatrix}. \tag{5.4}$$

Denoting the dependence of $\mathbf{J}(t)$ on a explicitly and denoting the k^{th} row of $\mathbf{J}(t, a)$ by $\mathbf{J}_k(t, a)$, we have

$$\hat{\mathbf{J}}(a) = \begin{bmatrix} \mathbf{J}_1(l_1, a) \\ \mathbf{J}_2(l_1, a) \\ . \\ . \\ . \\ \mathbf{J}_1(l_n, a) \\ \mathbf{J}_2(l_n, a) \\ \mathbf{J}_1(1, a) \\ \mathbf{J}_2(1, a) \\ \mathbf{J}_3(1, a) \end{bmatrix}. \tag{5.5}$$

In [41] we use computer vision to measure the position of markers along the metal strip, but for the experiments we present in this thesis, we look at a special case of this formulation. We consider the case where the positions of markers are not used ($n = 0$). Only the position and orientation at the end of the rod are measured. The Adept robot provides a measurement of its current placement. This includes the planned position along with the addition of a small amount of noise due to the alignment of the table. The manufacturer reports that the Adept robot has a repeatability of $+/-0.025$ mm for the xy plane. The table alignment procedure that is utilized allows for repeatability of less than 1mm.

5.1.2 Estimator Design

Once the configuration of the metal strip has been measured using the force sensor and the position sensor, these two measurements must be combined to form an estimate. In previous sections, we have shown that the position sensor attached to the robot most often is sufficient for measuring the configuration. We have also shown that the position measurement becomes less accurate as the norm of the inverse of the Jacobian of a with respect to q at the end of the strip ($\| \ J(1, a)^{-1} \ \|$) increases. The Jacobian provides a relative measurement of the sensitivity of each sensor to error. We have designed an estimator $\hat{a}$ where:

$$\hat{a} = a_{\text{force}} \frac{\Gamma}{\Gamma + \| \ (J(a\,[n])^{-1} \ \|} + a_{\text{position}} \frac{\| \ (J(a\,[n])^{-1} \ \|}{\Gamma + \| \ (J(a\,[n])^{-1} \ \|}. \tag{5.6}$$

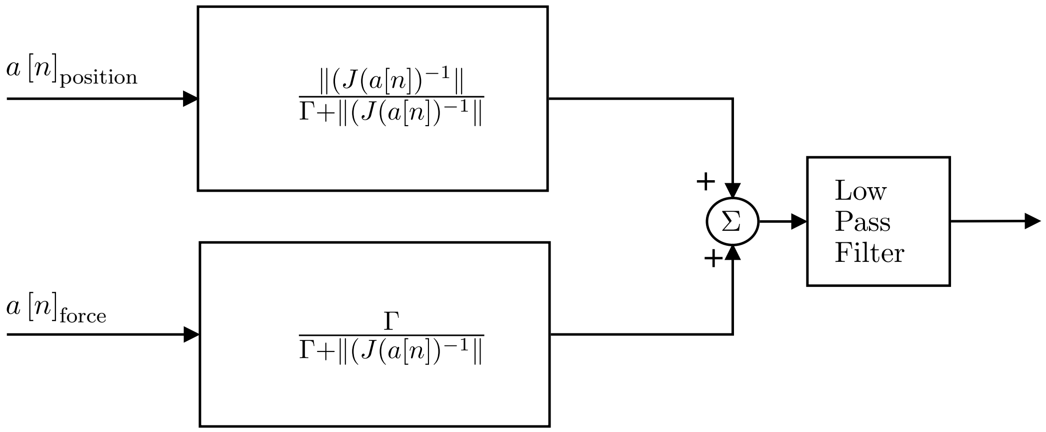

$a\,[n]_{\text{position}}$
$\frac{\|(J(a[n])^{-1}\|}{\Gamma+\|(J(a[n])^{-1}\|}$
$a\,[n]_{\text{force}}$
$\frac{\Gamma}{\Gamma+\|(J(a[n])^{-1}\|}$
+
+
Σ
Low Pass Filter

Figure 5.1: The design of the estimator that combines measurements from the position sensor and the force sensor.

This estimator requires Γ, a parameter that we select, and $\parallel (J(1,a)^{-1} \parallel$ which is calculated for the observed configuration (a_{force}). The block diagram of the estimator is shown in Figure 5.1. This weighting allows for the combined measurement to be biased towards the force sensor when the metal strip is determined to be in a configuration that is highly sensitive to small perturbations. When the rod is less sensitive to these perturbations, the combined measurement is biased towards the position measurement. After the two weighted measurements are combined, a low-pass filter is applied to remove high-frequency noise.

5.2 Controller Design

In this section, we describe the design of the proportional integral (PI) controller implemented that incorporates the observer presented in the previous section. Figure 5.2 shows the block diagram of the controller. The input to this system is the signal $a_{\text{ref}}[n]$, the reference configuration. For each estimate, the difference between the reference configuration and the observed configuration is calculated. An integrator calculates the sum of the error observed from each measurement. This cumulative sum is multiplied by the integral gain K_i. The current error measurement is scaled by the proportional gain K_p. The integral and proportional components are then combined to

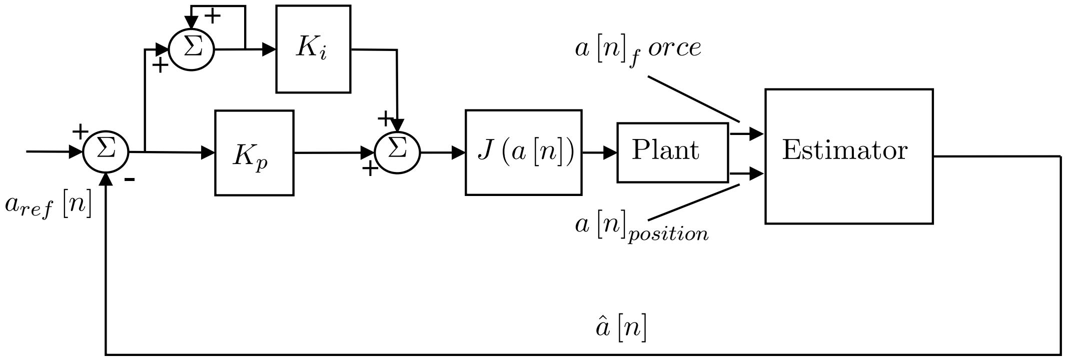


Figure 5.2: The feedback controller loop implemented for hardware experiments. This controller uses a force sensor and a position sensor to perform regulation at a point in $\mathcal{A}$ and tracking a straight-line path in $\mathcal{A}$

determine δa, the desired change in a. In order to determine the adjustment in robot placement required to achieve a desired configuration, the Jacobian of the system must be calculated. The Jacobian of $a_{\text{force}}[n]$ is calculated and then this matrix is multiplied by the vector obtained from combining the integral error and proportional error components such that

$$\delta b = J\left(1, a_{\text{force}}\left[n\right]\right)\delta a. \tag{5.7}$$

The position of the robot is adjusted by this calculated value for δb. After the robot is relocated to the new position and both sensors measure the current configuration of the metal strip, the estimator from Section 5.1.2 is applied, and another iteration of the feedback loop is executed.

Time-Varying a_{ref}

We use the feedback controller to track a time-varying point in $\mathcal{A}$. Instead of a constant value of $a_{\text{ref}}[n]$ for all n, the value can change over time. For the experiments presented, the paths tracked are the same straight-line paths used with the previous chapter. When tracking trajectories through $\mathcal{A}$, the a_{ref} is updated with each iteration of the feedback controller such that

$$a_{ref}\left[n+1\right] = a_{ref}\left[n\right] + \delta a. \tag{5.8}$$

5.3 Experiments

5.3.1 Purpose

The purpose of this set of experiments is to validate that using a feedback controller to manipulate a metal strip reduces the observed error below the error seen when performing open-loop manipulation. These experiments show that using sensor measurements of the current configuration to determine a feedback input ensures more robust manipulation. In addition to the planning paths that are minimally sensitive to the placement of the robotic manipulator, as shown in Chapter 4, we show that error can be reduced by using the sensor information.

5.3.2 Procedure

Experiment Paths

We performed a set of experiments where the feedback controller, presented in Section 5.2, is applied to paths that move through configurations that are sensitive to error in robot placement. The paths chosen are a subset of the paths used for experiments from Chapter 4. Each path moves through configurations that are extremely sensitive to the robot placement while the other path moves through configurations that are significantly less sensitive ($\| J(1,a)^{-1} \| > 3000$). Each path is sampled at an interval of $l\frac{1}{400}$ where l is the length of the path so that the trial will have the same number of steps as the open-loop manipulation experiment in Chapter 4.

Parameter Selection

The implementation of the feedback controller requires the selection of two sets of parameters. First, we must select Γ, the constant that is used in the estimator described in Section 5.1. For the experiments in this chapter, $\Gamma = 175$. This value was determined using a tuning procedure. The second set of parameters that we selected are the gains for the feedback controller. For the experiments in this chapter, $K_p = 1.000$ and $K_i = 0.005$. These values were also determined using a tuning procedure.

Error Metric

The error metric used for these experiments is the distance between the observed configuration and the predicted configuration

$$error^2 = (a_1 - a_1')^2 + (a_2 - a_2')^2 + (a_3 - a_3')^2. \tag{5.9}$$

We use the estimate which combines measurements of the configurations from the force sensor and measurements using the position sensor.

5.3.3 Results

The results demonstrate that the application of a PI feedback control law reduced the maximum error observed below the error observed when performing open-loop manipulation. The first set of graphs, Figure 5.3 and Figure 5.4, show the impact of the PI feedback controller. The black line shows a plot of the planned straight-line path but projected onto the a_3 plane. The blue lines in both plots show the observed path when the feedback controller is used along with the observed path using just open-loop control.

With both experiments, the observed configurations correspond well towards the ends of both line segments using an open-loop controller. The ends of both line segments correspond with configurations that are minimally sensitive to the robot placement. There is a large deviation from the planned paths for portions towards the middle of the line segment. Both sets of graphs show that the feedback controller follows the planned path better than the open-loop control strategy. However, the deviation from the planned path is less pronounced with the second trajectory that is less sensitive to robot placement. These results suggest that the application of the feedback controller to paths that minimize sensitivity to error in robot placement can better ensure robust manipulation when moving between configurations.

The second set of graphs, Figure 5.5 and 5.6, contrast the error observed when performing open-loop manipulation to the error observed when performing closed-loop manipulation. The trajectory in the first plot moves through configuration in $\mathcal{A}$ that are extremely sensitive to small perturbations in boundary conditions. The open-loop control has a maximum $error^2$ that is more than 10x the maximum experienced with the closed-loop controller. Though the closed-loop controller deviates from the planned path, the magnitude of error does not compare with the error seen with open-loop manipulation. With the second trajectory, the straight-line path does not move through configurations that are as sensitive to the placement of the robot as with the first trajectory. The maximum $error^2$ for both the open-loop and closed-loop controllers is significantly smaller than with the first trajectory. However, the error is still reduced by the application of the feedback controller.

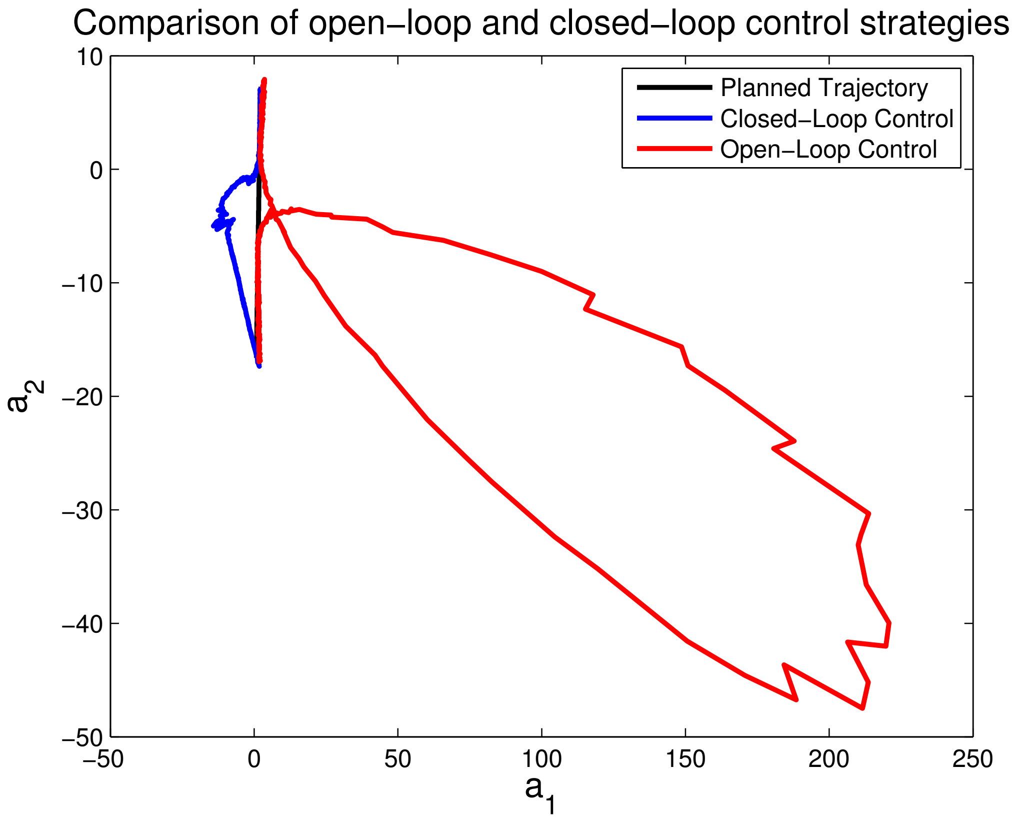


Figure 5.3: A straight-line path in $\mathcal{A}$, the observed configurations using an open-loop controller, and the observed configurations using a closed-loop controller.

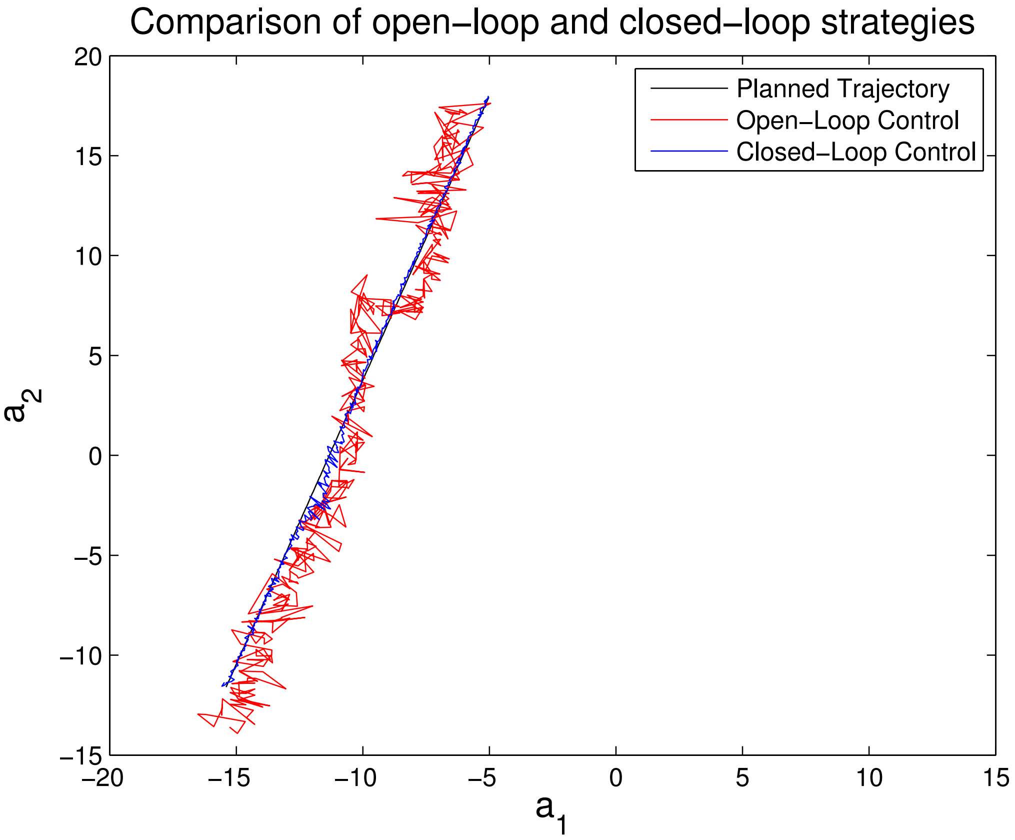


Figure 5.4: A straight-line path in $\mathcal{A}$, the observed configurations using an open-loop controller, and the observed configurations using a closed-loop controller.

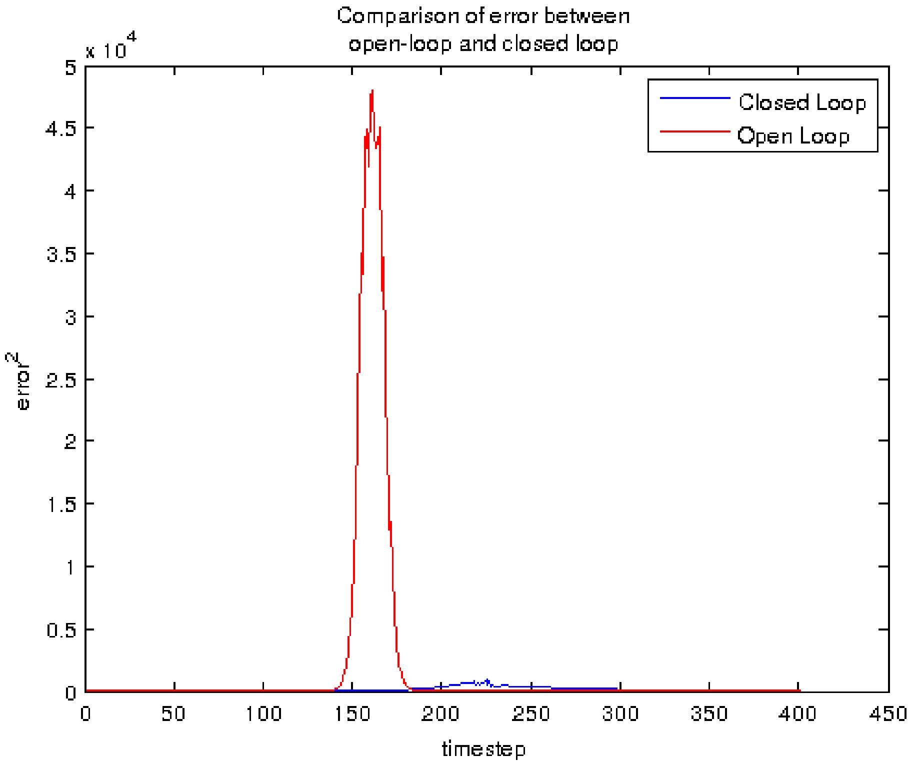


Figure 5.5: Reduction in error due to the implementation of the PI controller. For this experiment, the path planned in $\mathcal{A}$ is a straight-line path through a region in $\mathcal{A}$ that is extremely sensitive to error at the boundary conditions.

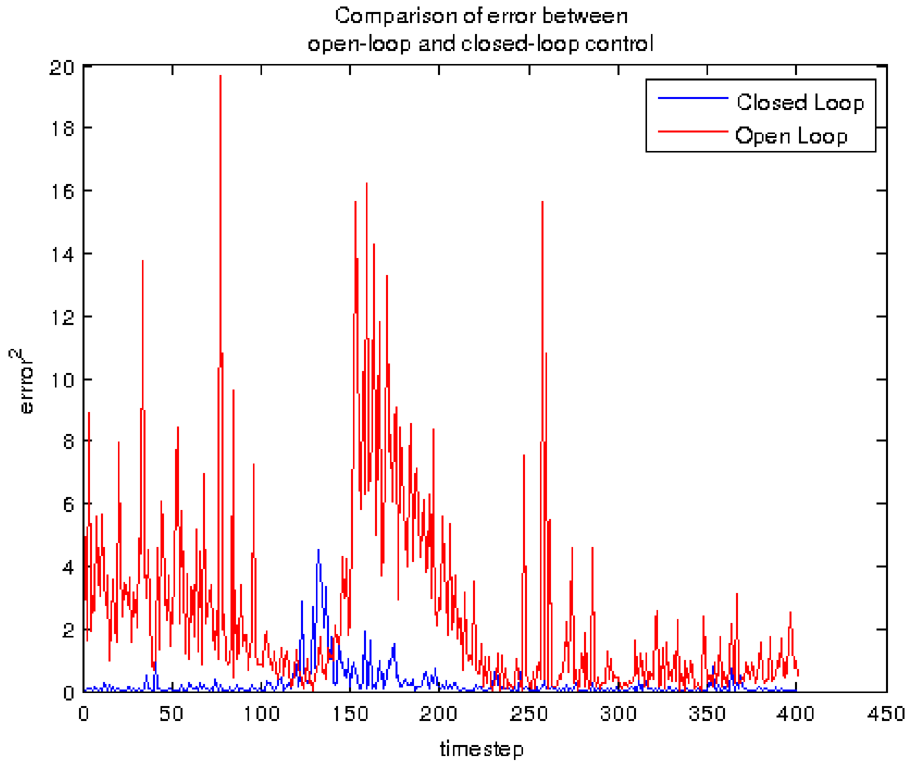


Figure 5.6: Reduction in error due to the implementation of the PI controller. For this experiment, the path planned in $\mathcal{A}$ is a straight-line path through a region in $\mathcal{A}$ that is less sensitive to error at the boundary conditions.

CHAPTER 6

FUTURE WORK AND CONCLUSION

In this chapter we discuss potential future work that is related to the research presented in this thesis. We begin by revisiting our motivating application and sharing how the problem formulation that we validated for the planar metal strip directly extends to spatial metal cables. The problem formulation is followed by the presentation of preliminary results that have been obtained with both the Adept robot and the Baxter robot. We close this chapter with remarks that conclude the work presented in this thesis.

6.1 Future Work

6.1.1 Motivation

The application of installing a wire harness using an industrial robot, such as Baxter, serves as motivation for the work we have conducted with a planar metal strip. The formulation applied to planar elastic rods can easily be extended to describe three-dimensional elastic rods as well. We can apply this theory to model thin, inextensible materials, such as thin rods and cables. The implementation of manipulation with spatial flexible objects, such as wires, has direct applications to industrial settings. Figure 6.1 shows Baxter, a new industrial robot, manipulating one end of a wire for a wire harness prototype with two cables. This prototype consists of two wires that are connected at one end. The other end of both cables is instrumented with a magnetic connector that enables connecting to the workspace. This prototype is representative of possible usages for the Baxter robot and the formulation that we present is capable of enabling the installation of cables using this environment or other industrial settings.

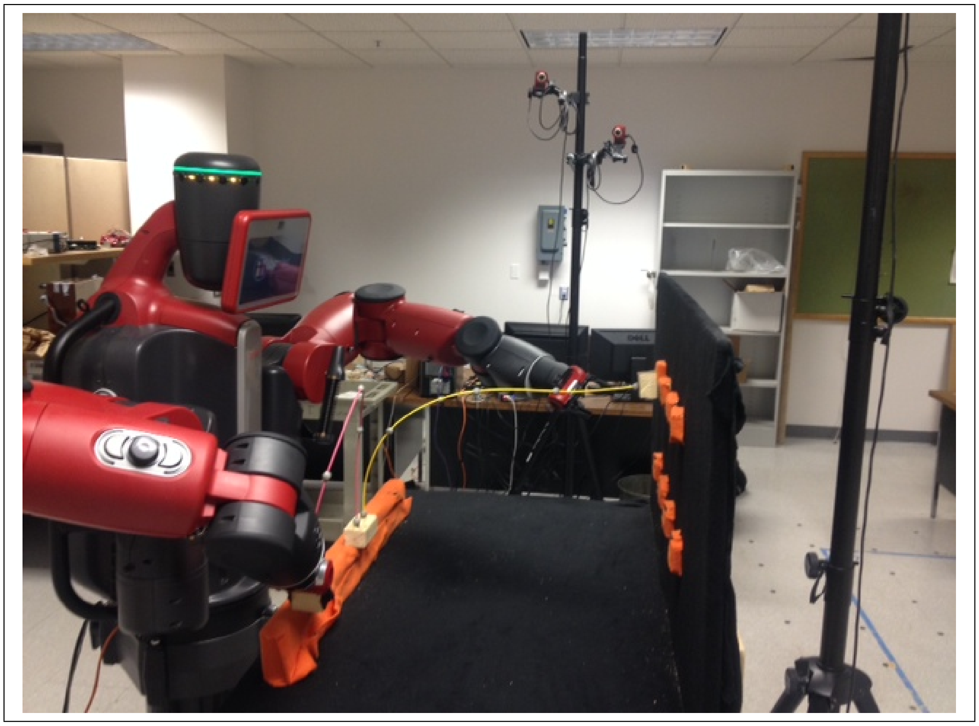


Figure 6.1: Baxter performing manipulation with a cable assembly prototype. This prototype contains two separate wires that are attached at one end. The Baxter robot is attaching the other end of one of the wires to a connector of a table surface.

6.1.2 Theoretical Framework for $\mathcal{A}$ with Spatial Elastic Rod

Our manipulation work builds upon previous work by Bretl and McCarthy [56] that describes the shape of kinematic chains and spatial elastic rods. We will focus on manipulating "massless" elastic rods, allowing us to neglect the impact of gravity on the shape formed as a result of end effector placement. Similar to the planar elastic rod, feasible shapes for the spatial elastic rod minimize the total energy given a set of boundary conditions. However, both endpoints are in SE(3) instead of SE(2).

We model the metal wire as a thin, inextensible, and unit length where the shape of this rod by a continuous map $q\colon [0,1] \to G$, as shown by Bretl and McCarthy, where $G = SE(3)$. Abbreviating $T_eL_q(\zeta) = q\zeta$, we require this map to satisfy

$$\dot{q} = q(u_1X_1 + u_2X_2 + u_3X_3 + X_4) \tag{6.1}$$

for some $u\colon [0,1] \to U$, where $U = \mathbb{R}^3$ and

$$\begin{aligned}&\{X_1, \ldots, X_6\} = \\ &\left\{ \begin{bmatrix} 0&0&0&0\\0&0&-1&0\\0&1&0&0\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&1&0\\0&0&0&0\\-1&0&0&0\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&-1&0&0\\1&0&0&0\\0&0&0&0\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&1\\0&0&0&0\\0&0&0&0\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0 \end{bmatrix} \right\}\end{aligned} \tag{6.2}$$

is a basis for $\mathfrak{g}$. Denote the dual basis for $\mathfrak{g}^*$ by $\{P_1, \ldots, P_6\}$. We refer to q and u together as $(q, u)\colon [0,1] \to G \times U$ or simply as (q, u). For our experiments, we assume that both ends of a wire are affixed to grippers at locations $q(0)$ and $q(1)$. We also assume that $q(0) = e$, however, this pose is free to change within the world frame such that q(1) is just the offset from the base of the rod. The other end of the rod, $q(1)$, is free to take any configurations in $\mathcal{B} = G$. Unlike the planar elastic rod that only could bend in one plane and had only one component for potential energy, the spatial has three potential energy components that must be minimized in order for the rod to be in static equilibrium. Similar to Bretl and McCarthy [56], we assume the rod has total elastic energy $\frac{1}{2}\int_0^1 (c_1u_1^2 + c_2u_2^2 + c_3u_3^2)\, dt$ for given constants $c_1, c_2, c_3 > 0$ [51]. These constants c_1, c_2, c_3 are weighting factors that relate the three sources of potential energy, u_i. This is essentially the Poisson's ratio.

In order for the rod to be in static equilibrium, the rod must be in the minimal energy configuration given the boundary conditions. This energy is:

$$\begin{aligned} \underset{q,u}{\text{minimize}} \quad & \frac{1}{2}\int_0^1 \left(c_1u_1^2 + c_2u_2^2 + c_3u_3^2\right) dt \\ \text{subject to} \quad & \dot{q} = q(u_1X_1 + u_2X_2 + u_3X_3 + X_4) \\ & q(0) = e, \qquad q(1) = b \end{aligned} \tag{6.3}$$

for some $b \in \mathcal{B}$.

Necessary Conditions for Static Equilibrium

In previous work by Bretl and McCarthy [56], Theorem 5 states:

Theorem 8. *A trajectory* (q, u) *is normal with respect to* (6.3) *if and only* *if there exists* $\mu\colon [0,1] \to \mathfrak{g}^*$ *that satisfies*

$$\begin{aligned} \dot{\mu}_1 &= u_3\mu_2 - u_2\mu_3 & \dot{\mu}_4 &= u_3\mu_5 - u_2\mu_6 \\ \dot{\mu}_2 &= \mu_6 + u_1\mu_3 - u_3\mu_1 & \dot{\mu}_5 &= u_1\mu_6 - u_3\mu_4 \qquad (6.4) \\ \dot{\mu}_3 &= -\mu_5 + u_2\mu_1 - u_1\mu_2 & \dot{\mu}_6 &= u_2\mu_4 - u_1\mu_5, \end{aligned}$$

$$\dot{q} = q(u_1X_1 + u_2X_2 + u_3X_3 + X_4), \tag{6.5}$$

$$u_i = c_i^{-1}\mu_i \quad \textit{for all } i \in \{1,2,3\}, \tag{6.6}$$

with initial conditions $q(0) = e$ *and* $\mu(0) = \sum_{i=1}^{6} a_iP_i$ *for some* $a \in \mathcal{A}$*, where*

$$\mathcal{A} = \left\{a \in \mathbb{R}^6\colon (a_2, a_3, a_5, a_6) \neq (0,0,0,0)\right\}.$$

These equations are the spatial analog to Equations (3.8) - (3.10). Using a differential equation solver, we can determine the shape of a 3D elastic rod given the value in $\mathcal{A}$.

6.1.3 Preliminary Experiments with Adept Robot

We have performed a preliminary set of experiments with the Adept robot to compare the shape formed by the metal cable to the predicted shape from the model described in Section 6.1.2. Figure 6.2 shows the testbed designed for these experiments. As with planar experiments, one end of the "rod" is anchored to a table and the other end is affixed to the Adept robot. The thin, stranded steel cable used for these experiments is sufficiently strong that gravity has only a small impact on the experienced shape. This allows the rod to be treated as a massless elastic rod. Because the Adept robot only provides five degrees of freedom, a servo motor has been integrated with the robot to provide a sixth degree of freedom.

To track a finite set of points along the rod, the cable has been instrumented with infrared markers that allow the position of these locations to be tracked using an Opti-Track motion capture system. In addition to the markers along the cable, markers have been attached to the table and the servo motor attachment to provide orientation information for both surfaces. Three markers are used to annotate the surface of the table where the elastic rod is anchored. Two markers are placed on the table and a marker is placed

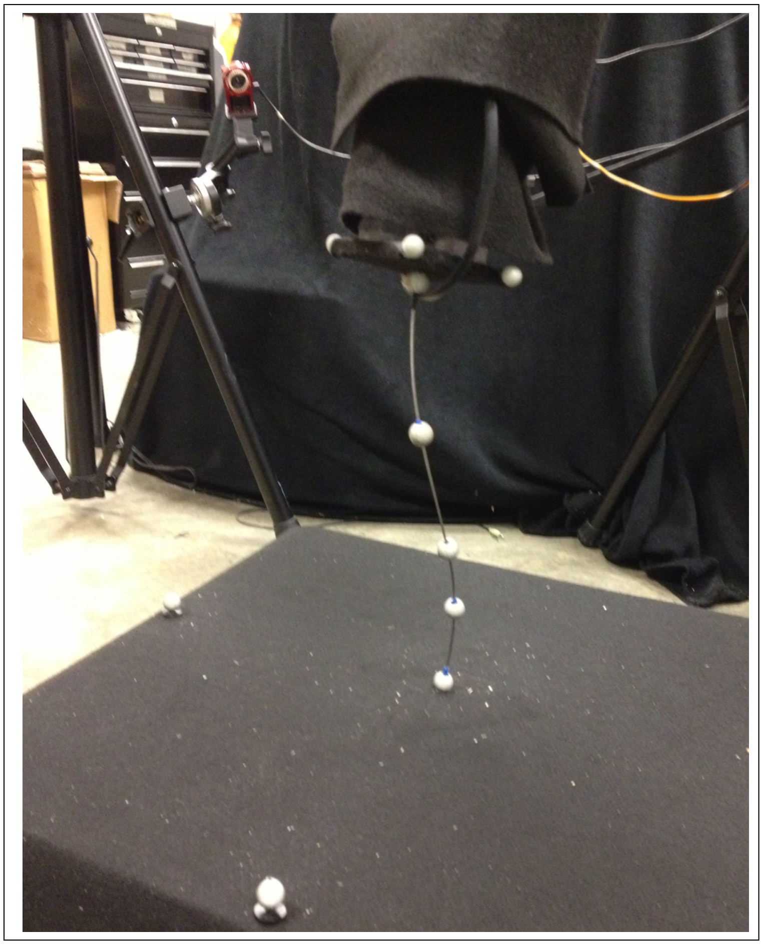


Figure 6.2: Steel cable that is attached at one end to an Adept robot. The other end of the cable is attached a table. This cable and the table have both been instrumented with infrared markers that are tracked by Opti-Track cameras.

at the base of the metal cable. These markers allow for the determination of the surface normal at the origin of the rod. Three markers on the servo attachment are used to determine the surface normal vector of the end effector. All four markers on the attachment are used to calculate the centroid of the servo surface. This centroid corresponds with the end of the elastic rod attached to the robot.

Figure 6.3 shows plots of two sample configurations. We chose two points in $\mathcal{A}$ and calculated the boundary conditions to determine the position and orientation for the robot and Servo motor. The Opti-Track motion capture system was used to determine the location of the IR markers. The yellow

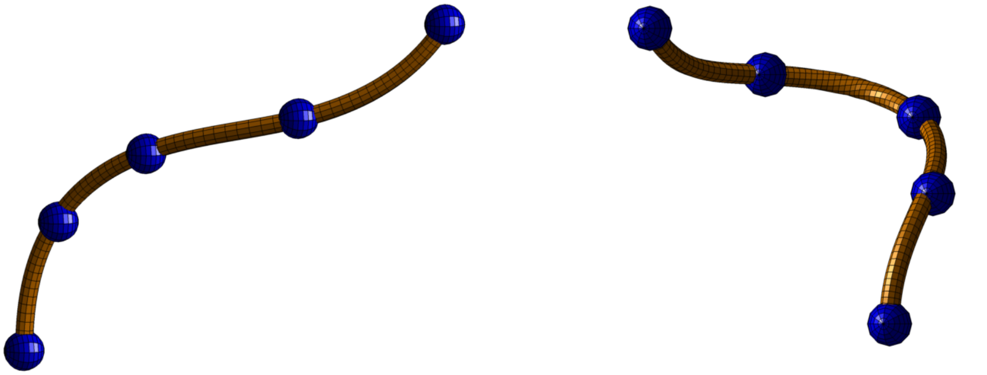


Figure 6.3: Two examples from hardware experiments with a spatial metal cable that help to confirm that our model predicts the shape of a metal wire. The spheres correspond with the detected position of the IR markers along the rod. The yellow curve corresponds with the configuration that best matches the detected marker positions.

rods in these images show the theoretical shape for the planned configuration. The blue spheres correspond to the IR markers that were detected with the experiment. The correlations between the markers and the planned shape serves to validate that the shape of the elastic rod matches the predicted configuration.

6.1.4 Preliminary Experiments with Baxter Robot

In addition to the preliminary experiments with the Adept robot, we also conducted experiments with the Baxter robot. Figure 6.4 shows a picture of an experimental setup where both ends of a metal, spatial cable are held by the Baxter robot. This configuration was planned in $\mathcal{A}$ and the position and orientation to place Baxter's gripper to realize this shape were calculated by integrating Equation (6.5). The motion capture system recorded the position of 5 points along the cable. Figure 6.5 shows a comparison between the detected position of IR markers along the metal cable and the corresponding points along the predicted shape of the curve. Though these two shapes do show some correlation, the results are not as accurate as the model for planar rods. Our current speculation is that the discrepancy in shape is due to imprecise realization of Cartesian coordinates using Baxter as well as the wire's natural curvature.

Future work is necessary to reduce the amount of error between the predicted and observed shape of the metal cable. The error could be reduced by implementing a planning strategy that avoids configurations where the cable is sensitive to the gripper placement. The cost function shown in Chapter 4 directly extends to the spatial metal cable. Furthermore, future work should also refine the placement of the grippers of the Baxter robot. Unlike the Adept robot that is precise to a fraction of 1 mm, the Baxter robot repeatability is orders of magnitude larger.

In addition to planning robust paths and improving the placement of the robot grippers, manipulation can be improved by planning in a configuration space of higher dimension. The ability of Baxter to perform manipulation with two grippers that each have 7 degrees-of-freedom can also be leveraged to improve manipulation of 3D rods. With all of our experiments, we have relied upon one end of the rod being fixed. With the Baxter robot, we can perform manipulation using both ends of the rod and remove the constraint on one end of the rod. Motion planning can be done using a space of dimension 20 (7 degrees of freedom for each arm and 6 degrees of freedom to describe the shape of the rod). Though this formulation would greatly increase the size of the configuration space, planning using this space produces plans that involve simultaneous motion with both arms and remove the constraint that one end of the cable remains fixed. Planning in this configuration space produces a motion plan that fully leverages Baxter's capabilities.

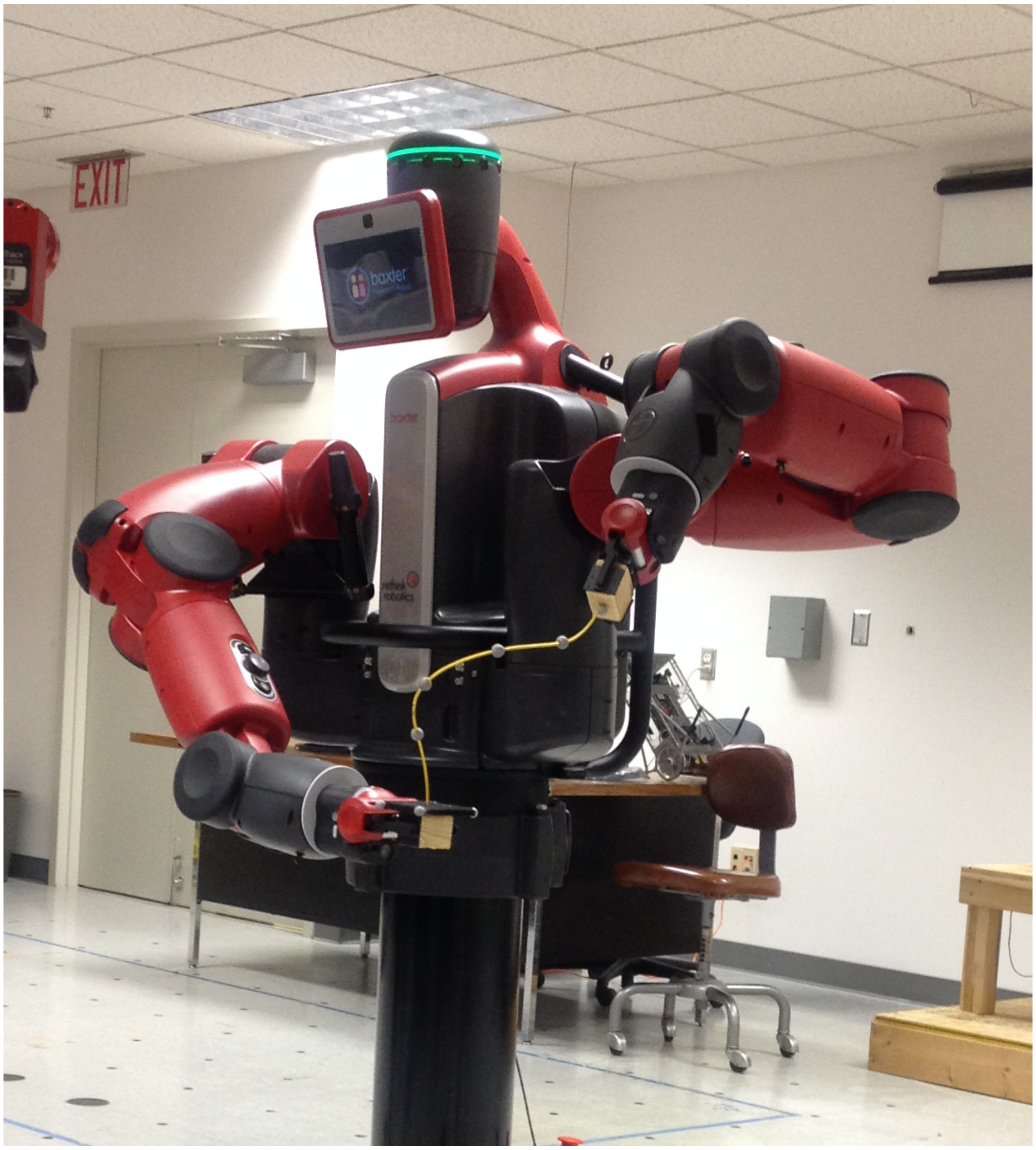


Figure 6.4: Baxter holding an elastic rod that has been instrumented with IR markers along its length. The bottom of the rod is affixed to a flat surface with IR markers that allow for determining the surface normal at the origin of the rod.

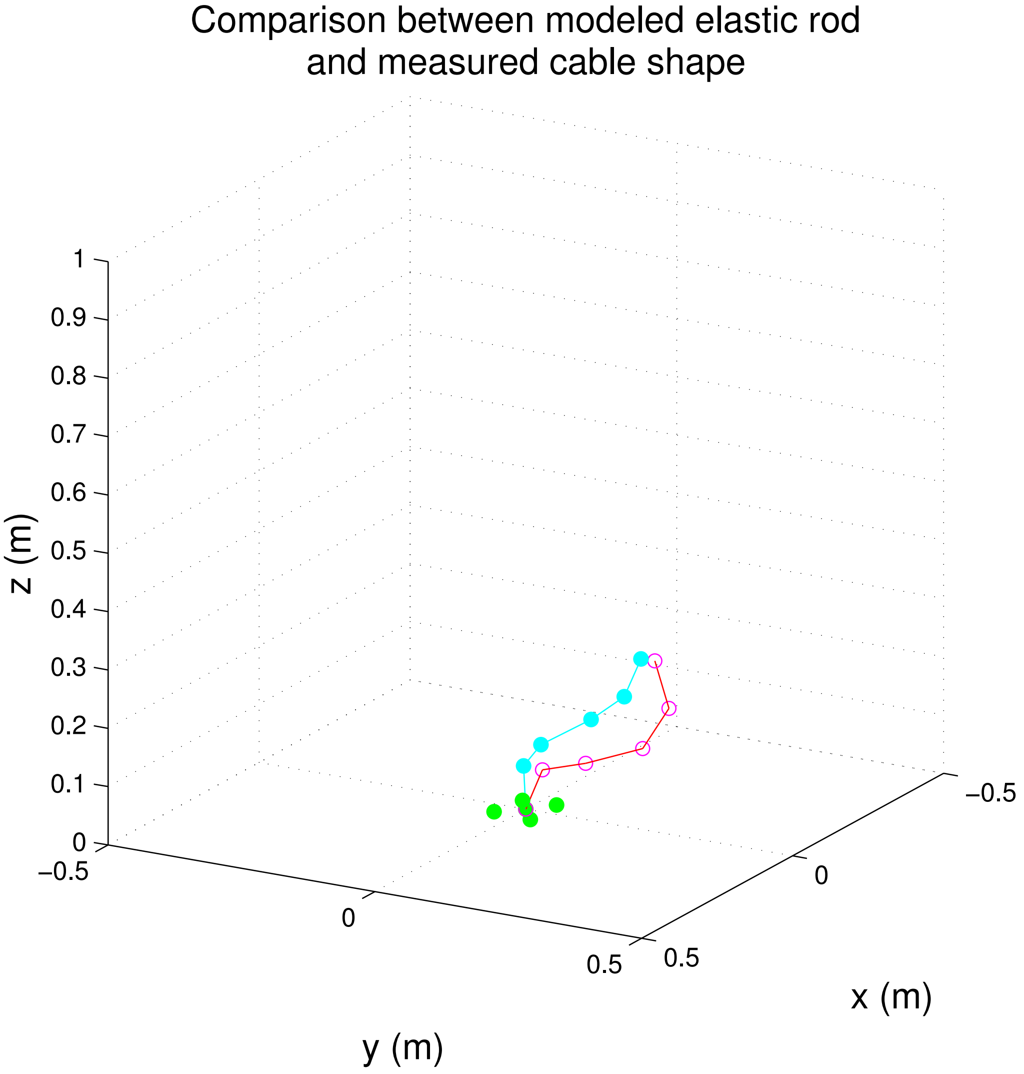


Figure 6.5: This plot shows a comparison between the predicted configuration of a wire and the shape seen in a hardware experiment using a Baxter robot. The turquoise curve and circles indicate the predicted shape of the metal cable and the position of the IR markers. The maroon curve and circles show the observed shape of the metal cable and position of the markers. The four green circles indicate the position of the IR markers used for determining the surface normal of the base of the metal cable.

6.2 Conclusion

In this dissertation, we have shown a new method for performing manipulation with a planar rod. We begin by modeling a thin metal strip as a Kirchhoff elastic rod whose movement is confined within a plane. This coordinate chart corresponds with the force and torque at one end of the rod. Using this description, that can uniquely describe all possible shapes for a planar rod, manipulation can be performed by identifying a sequence of feasible configurations with our chart. Each configuration represents a unique shape that can be determined using integration. Once the end of the rod is determined for each configuration, a robot is used to move the end of the rod to the correct position and orientation.

We provided evidence in the form of hardware experiments to validate that our approach is capable of performing manipulation in a manner that allows us to precisely transform the shape of a flexible object from a starting configuration to a goal configuration. In addition to showing proof-of concept experiments that display the ability to change the shape, we examine the ability of our model to accurately predict when the shape of the metal strip will deviate from our model.

We presented a cost function that ensures that a flexible object avoids configurations that are nearly infeasible. The Jacobian for our coordinate chart provides a measurement of the sensitivity of the force to small perturbations at the boundary conditions. Using the relationship where $\delta a = J^{-1} \delta b$, we derived a cost function that assesses a penalty to configurations that are sensitive to boundary conditions. In addition to the sensitivity to small perturbations, a cost is also assessed for configurations that have an orientation at the end of the rod that approaches the manipulator joint limitations and also configurations that are close to self-collision.

We showed that by planning paths that minimize the total cost to transition from one state to another state, manipulation can be performed that ensures the observed force remains consistent with the predicted force. In addition to showing the success of manipulation using this cost function, we also show experimental results that clearly show that motion planning strategies that do not take the sensitivity to boundary conditions can result in the rod exerting forces larger than 15x the intended force.

While planning robust paths using the cost function reduces the observed error, we also demonstrated that error can also be reduced by applying a feedback controller that utilizes measurements of the state using a force sensor and a position sensor. The implemented feedback policy locally minimizes the sum-squared error between the planned and observed paths in our global coordinate chart for the wire. We showed that this feedback controller reduces the error between the planned configuration and the observed configuration.

In addition to the wires whose movement is constrained to translation rotation in SE(2), we extend our formulation to apply to spatial cables. We presented some preliminary experiments where the wire is manipulated with an Adept robot and show that the shape of the rod can accurately be described using our formulation. To demonstrate the utility of our approach, we designed a prototype for a wire harness as well as a circuit board prototype. We also presented results from proof-of-concept manipulation experiments with the Baxter robot and the Adept robot.

APPENDIX A

REVIEW OF SAMPLING-BASED PLANNERS

A.1 Review of Sampling Based Planner

This section provides an overview of sampling based motion planners to provide a foundation for the algorithms that we implement for performing manipulation with both planar and spatial elastic rods. These planners generate a sequence of steps that move a system from a starting state to a destination state. Sampling based planners are described thoroughly in textbooks [57], [58], [59] and also survey papers [60], [39], [61].

A.1.1 Configuration and Configuration Space

For robotic systems, the state can be represented as a vector with a dimension that corresponds to the available degrees of freedom. For example, with a 2 dimensional Cartesian space, the values for x and y can be used to describe the position of an object within the environment. The vector representation, also called a configuration (c), is simply a point that corresponds to the system's state.

The configuration space, C-space, represents all possible configurations that the system may take. C-space consists of both feasible configurations ($\mathcal{C}_{free}$) and infeasible configurations ($\mathcal{C}_{obs}$). The infeasible configurations for the system correspond to obstacles without the configuration space that indicate imposed constraints. These infeasible configurations correspond to obstacles. For a robot within a 2D environment, these obstacles could consist of external boundaries or items that occupy space within the environment that prevent the robot from moving to locations. Figure A.1 shows a 2D configuration space. The three polygons indicate obstacles within the workspace ($\mathcal{C}_{obs}$). The remaining space indicates feasible configurations ($\mathcal{C}_{free}$).

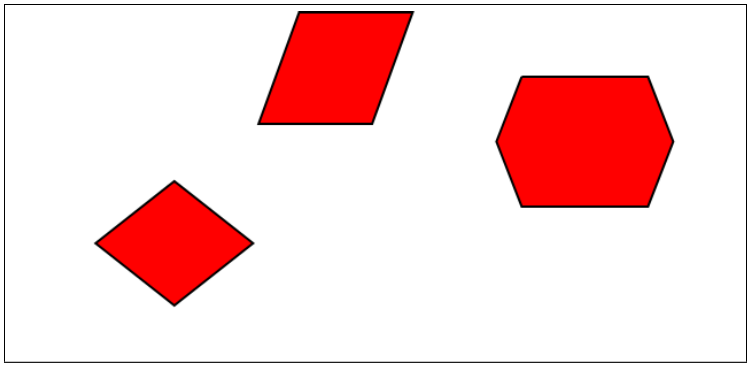


Figure A.1: Graphic of free space along with obstacles

While the shape of these three polygons could be explicitly described in C-space, describing ($\mathcal{C}_{obs}$) for some robotic systems can be significantly more complex.

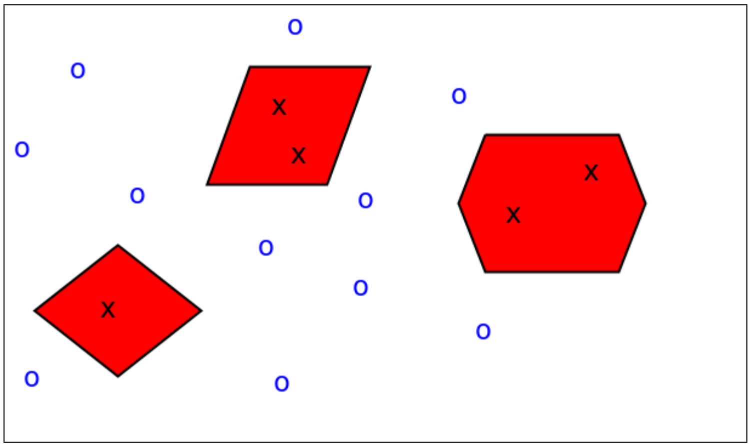


Figure A.2: Result of sampling configuration space to determine collision-free obstacles

Sampling based motion planners provide an alternative to having to represent $\mathcal{C}_{obs}$ explicitly. These planners simply require the determination of whether a selected configuration corresponds with a configuration that is within ($\mathcal{C}_{obs}$) or ($\mathcal{C}_{free}$). Figure A.2 depicts the result of sampling the 2D configuration space to determine configurations within $\mathcal{C}_{free}$. The o's correspond to sampled points within $\mathcal{C}$ that are obstacle-free. The x's were found to be within the boundaries of infeasible regions.

Once the space has been sampled, it is necessary to identify pairs of configurations that have a path between them that does not intersect with ob-

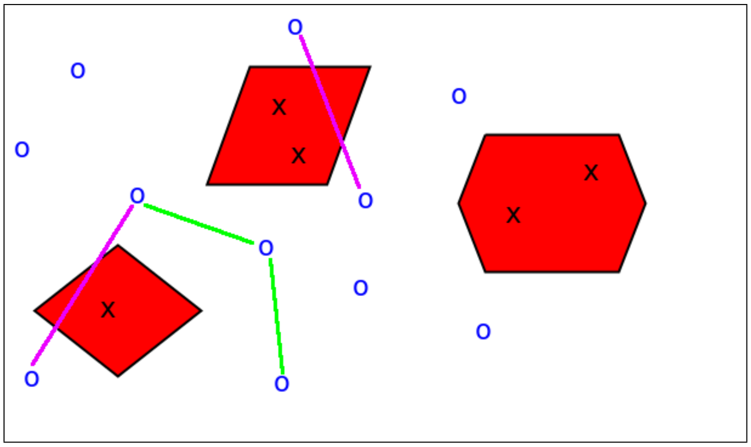


Figure A.3: Determination of whether paths between sampled configurations collide with obstacles in $\mathcal{C}$

stacles. To perform this task a straight-line path between two configurations is sampled at an interval small enough to ensure a collision will be detected if there is an intersection with an obstacle. Figure A.3 shows the result of checking four pairs of configurations for collisions with obstacles. The two green lines indicate that the path between the feasible configurations was collision-free. The two purple lines correspond with pairs of configurations that cannot be connected with a straight-line path.

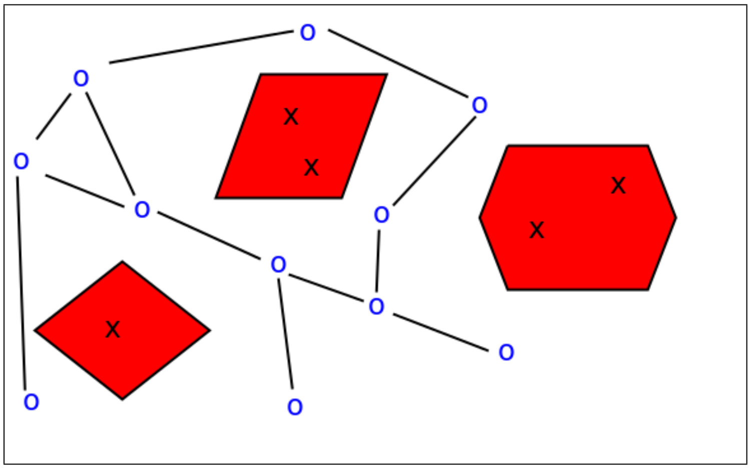


Figure A.4: Connection of configurations with collision-free paths

Figure A.4 shows a result of performing a search for nodes that can be connected with collision-free paths. The image shown does not indicate every pair of configurations that can be connected without passing through an obstacle is stored. To exhaustively check all paths between configurations requires $n(n-1)$ paths. This procedure could be computationally prohibitive. However, a distance metric could be used to reduce the number of paths that need to be checked for collisions. The collision-free paths are stored in a data structure such as an adjacency matrix.

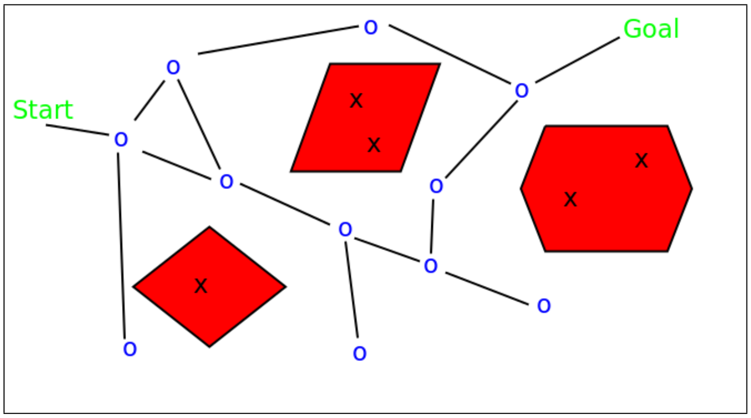


Figure A.5: Determination of paths to move to the start configuration and the goal configuration

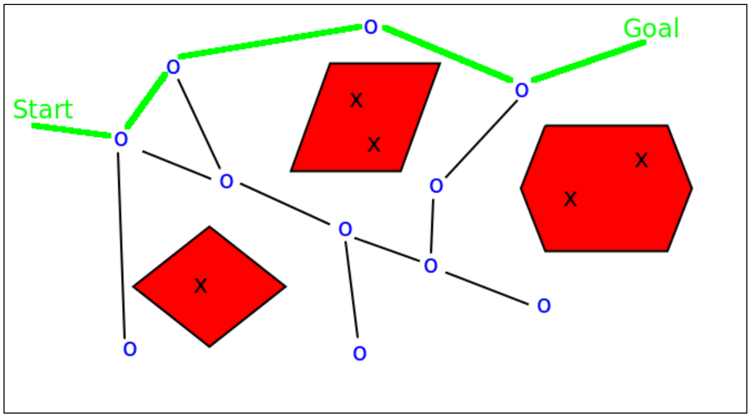


Figure A.6: Determination of path from start to goal

This structure that stores feasible paths between configurations is known as a roadmap. It can be used to identify a sequence of paths to take to move from a starting configuration to a goal configuration. The start configuration and the goal configuration are both added to the roadmap by finding collision-free paths from these configurations to the existing roadmap. Figure A.5 shows the roadmap after the start and goal configurations are connected.

Once these two configurations have been added, a graph search algorithm, such as breadth-first search, can be used to traverse the graph to find a sequence of states to visit in order to move between the two configurations. Figure A.6 shows a feasible path to move from the start configuration to the goal configurations. With the sample configuration space shown, there are multiple paths that could move from the start configuration to the goal. With some robotic systems, costs can be assessed to each path that could correspond to criteria such as proximity to obstacles or total distance traveled. This cost information can be used to determine the optimal path to take in the event that multiple feasible paths exist between the start and goal configuration.

APPENDIX B

CALCULATION OF COMPONENTS OF COST FUNCTION

In this appendix, we show the calculations of the three components of the cost function used for performing robust motion planning. The plots are shown for a_3 increments of 2. Only the planes where a_3 is negative are shown due to the symmetry in the cost function. For any configuration, $C(a_1, a_2, a_3) = C(a_1, -a_2, -a_3)$ where C is the cost at the point (a_1,a_2,a_3) in $\mathcal{A}$. Figures B.1 - B.6 show the cost due to the quantity $\log(\| (J(a))^{-1} \|)^2$ which indicates the sensitivity to the placement of the robotic manipulator. Figures B.7 - B.12 show the cost due to the proximity to self-collision. Figures B.13 - B.18 show the cost due to the orientation angle of the end of the metal strip. Figures B.19 - B.24 show the total cost, which is the sum of all three components.

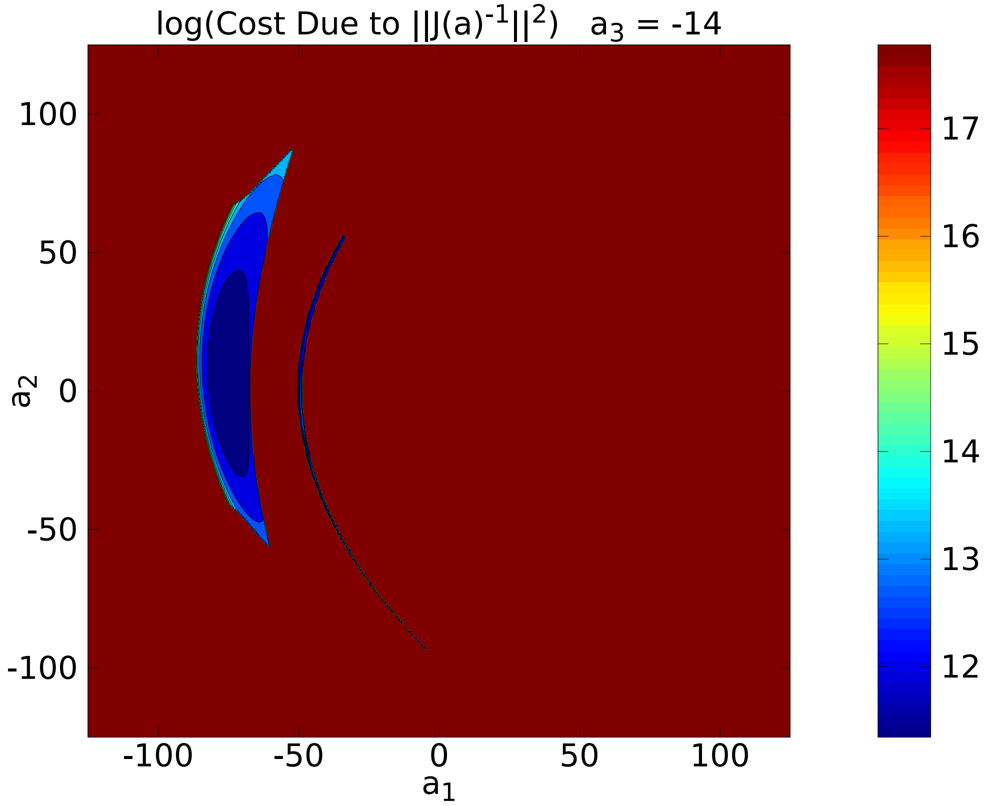


Figure B.1: The cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -14.0$.

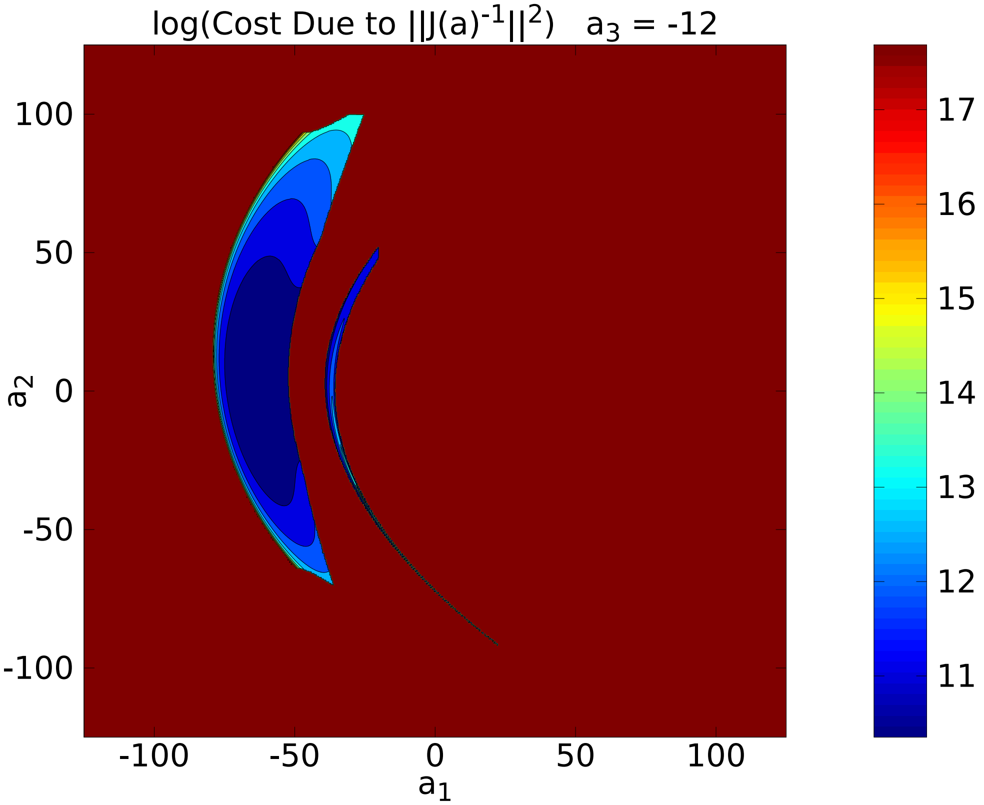


Figure B.2: The cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -12.0$.

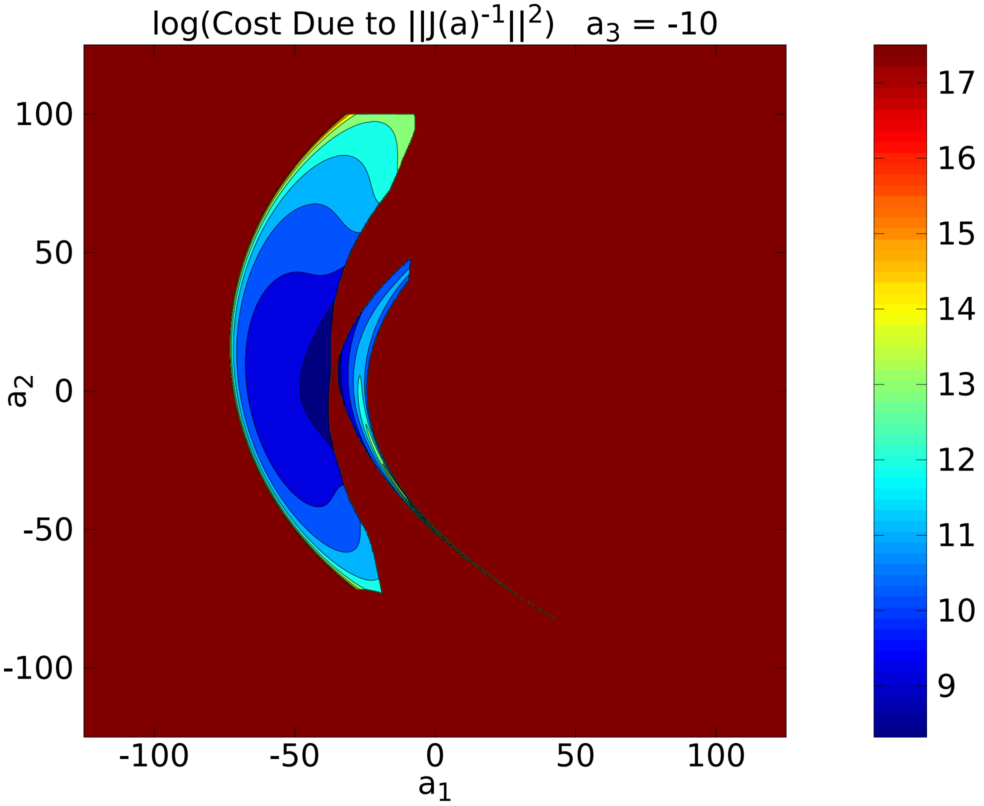


Figure B.3: The cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -10.0$.

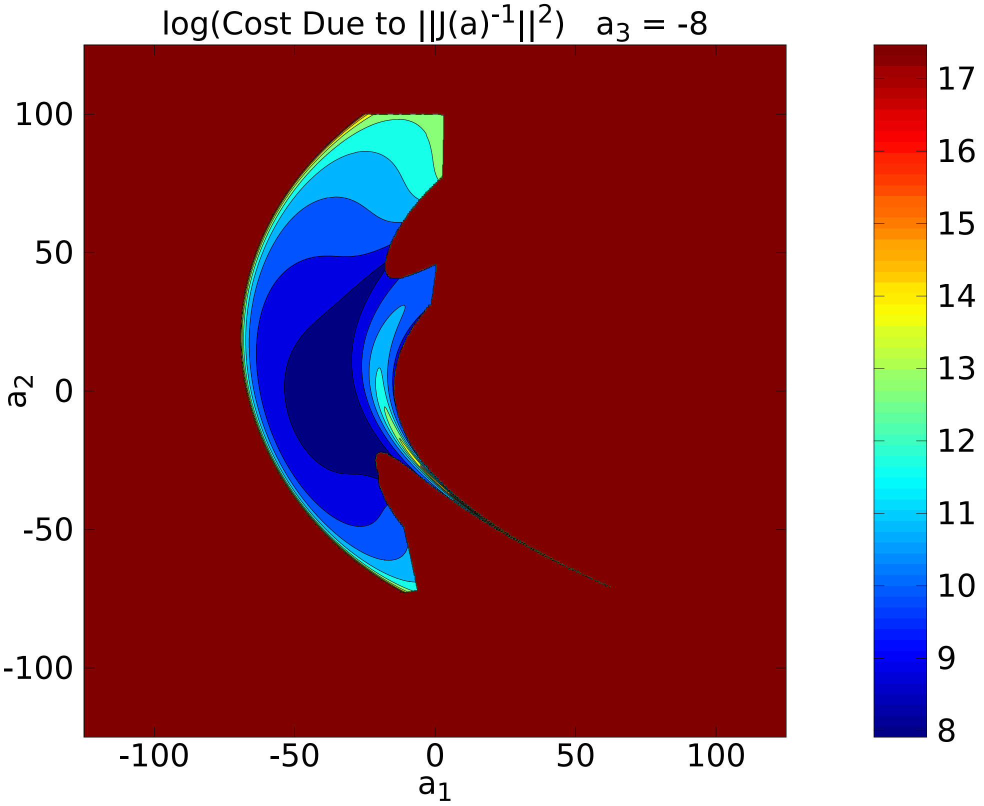


Figure B.4: The cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -8.0$.

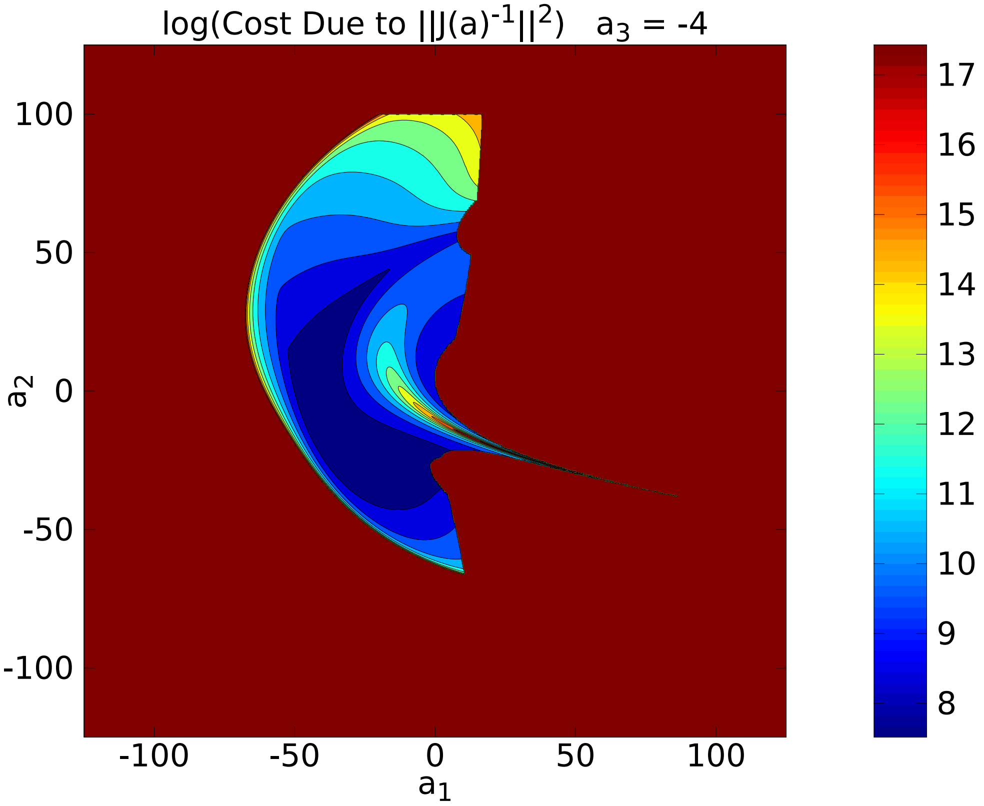


Figure B.5: The cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -4.0$.

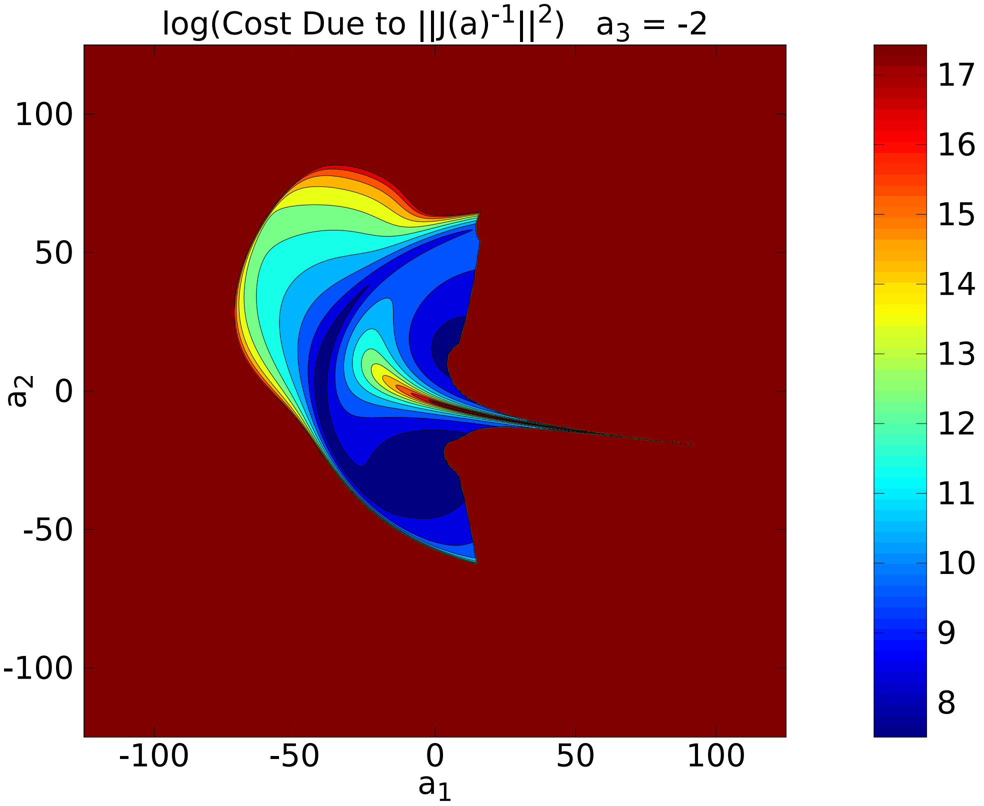


Figure B.6: The cost assessed to configurations due to the sensitivity of the configuration to error in the placement of the robotic manipulator for the plane where $a_3 = -2.0$.

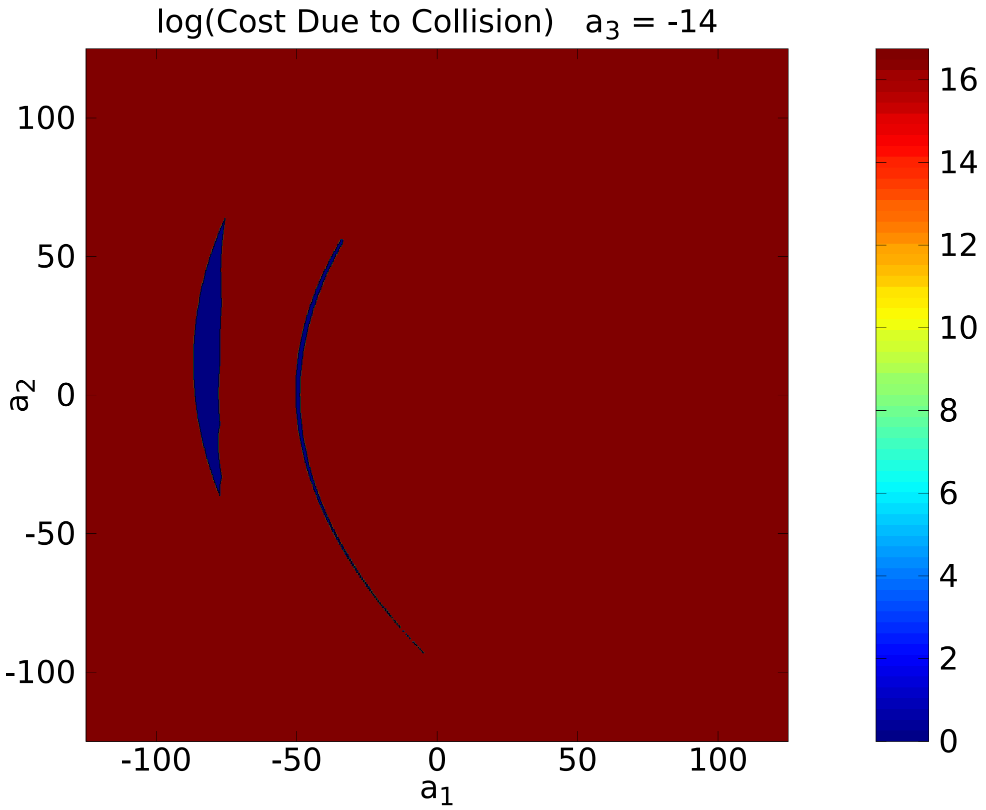


Figure B.7: Component of the total cost due to the proximity to self-collision as defined by Equations (4.2) and (4.3) for the plane where $a_3 = -14.0$.

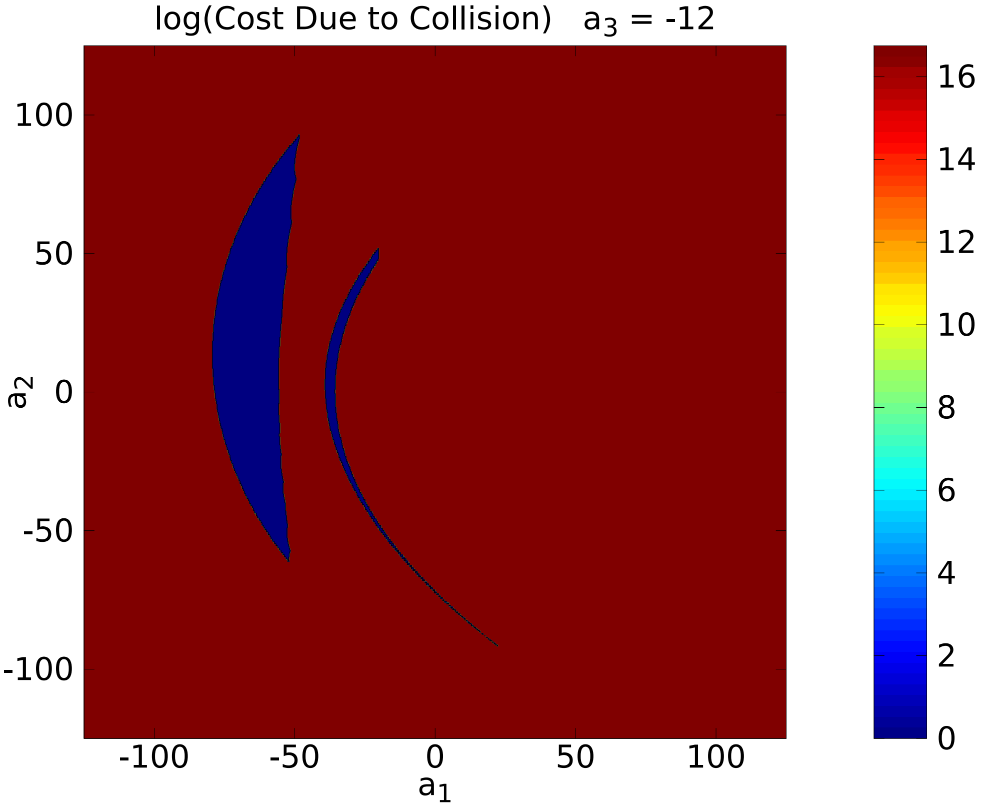


Figure B.8: Component of the total cost due to the proximity to self-collision as defined by Equations (4.2) and (4.3) for the plane where $a_3 = -12.0$.

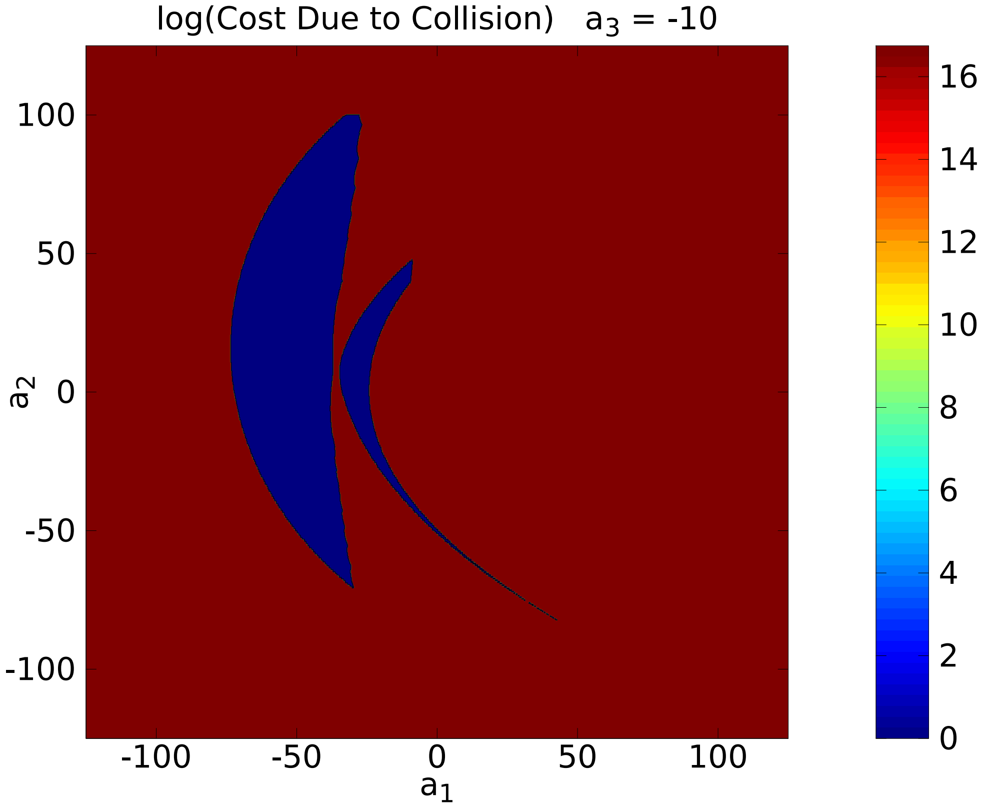


Figure B.9: Component of the total cost due to the proximity to self-collision as defined by Equations (4.2) and (4.3) for the plane where $a_3 = -10.0$.

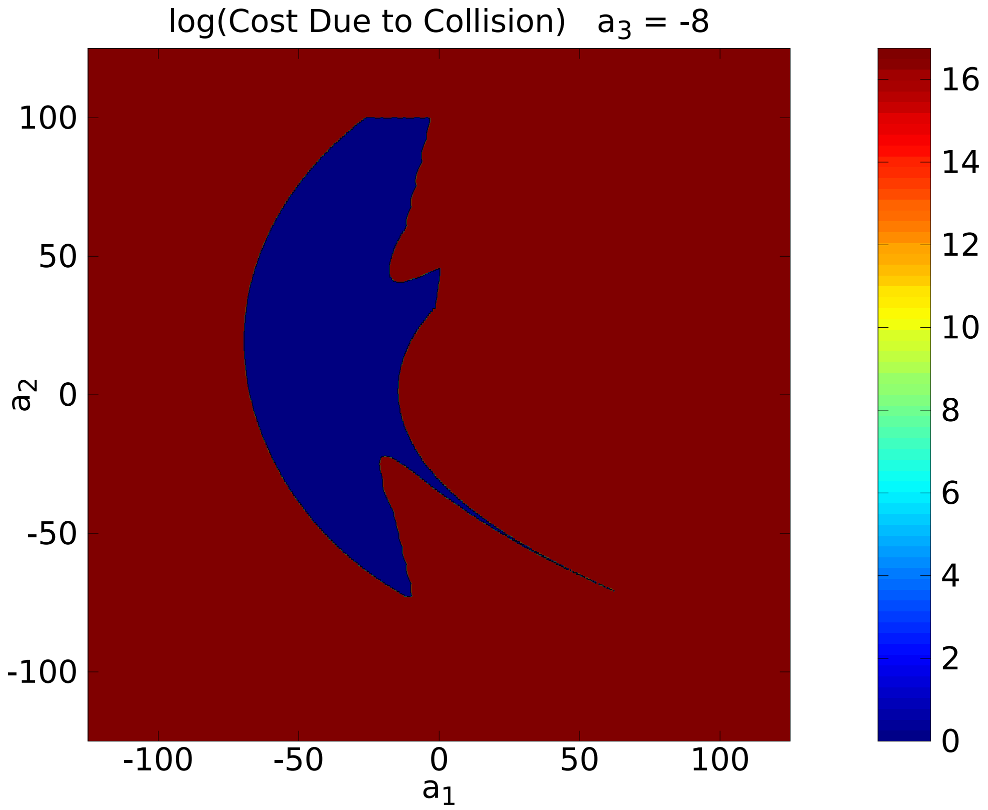


Figure B.10: Component of the total cost due to the proximity to self-collision as defined by Equations (4.2) and (4.3) for the plane where $a_3 = -8.0$.

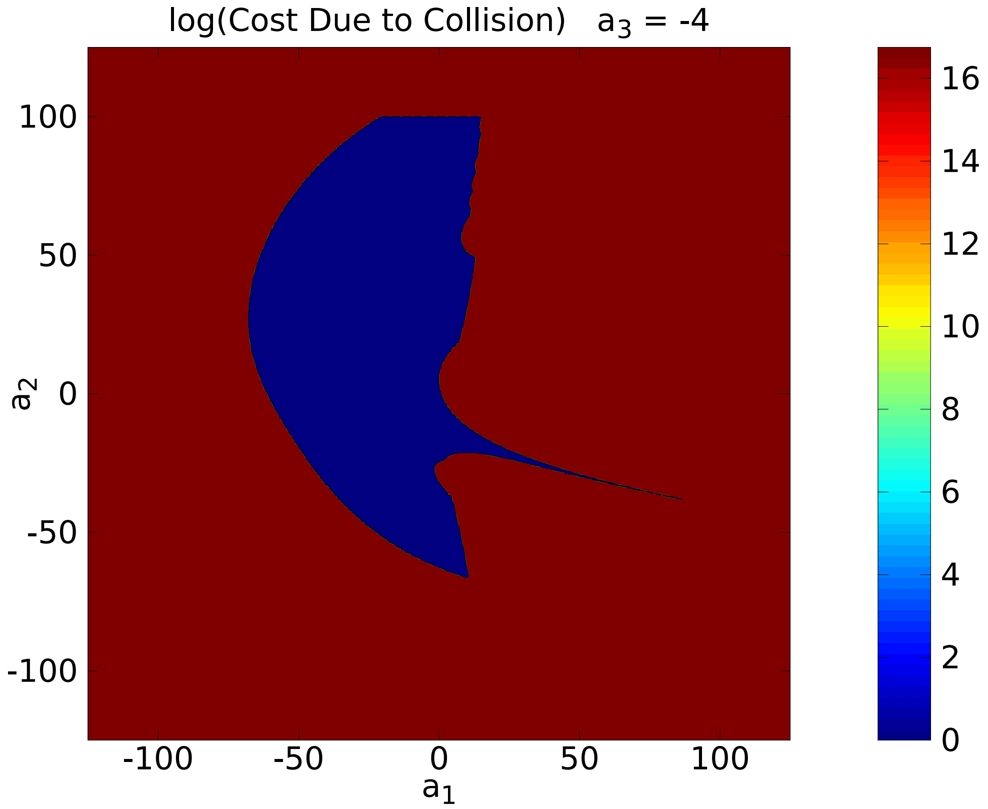


Figure B.11: Component of the total cost due to the proximity to self-collision as defined by Equations (4.2) and (4.3) for the plane where $a_3 = -4.0$.

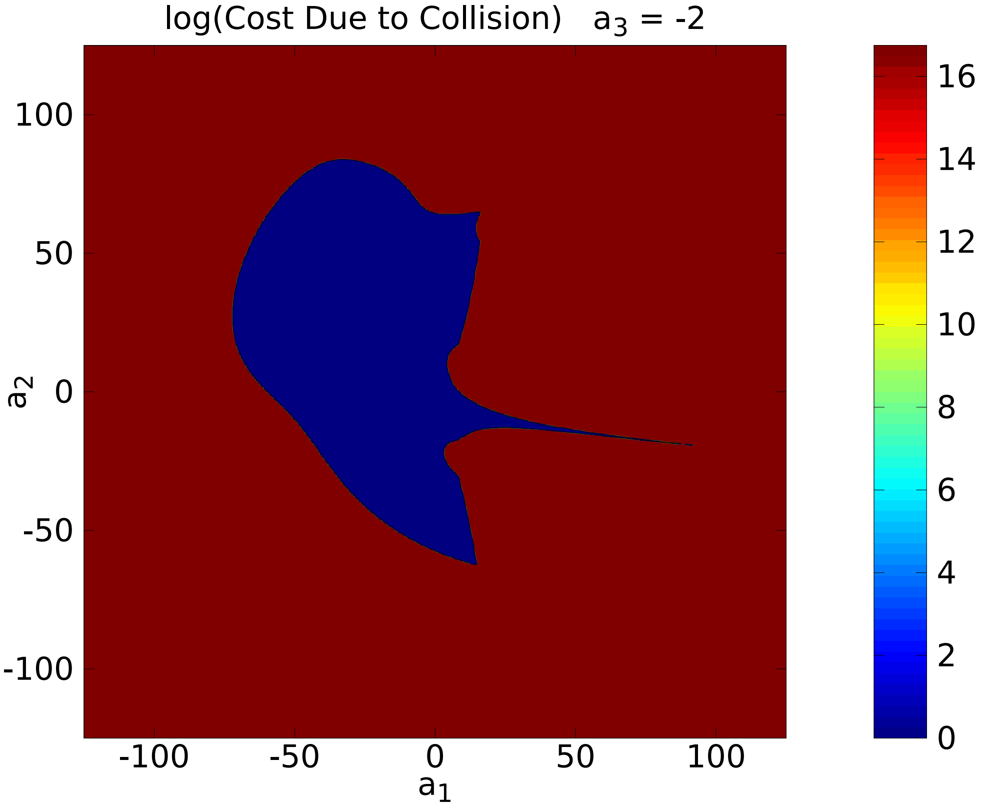


Figure B.12: Component of the total cost due to the proximity to self-collision as defined by Equations (4.2) and (4.3) for the plane where $a_3 = -2.0$.

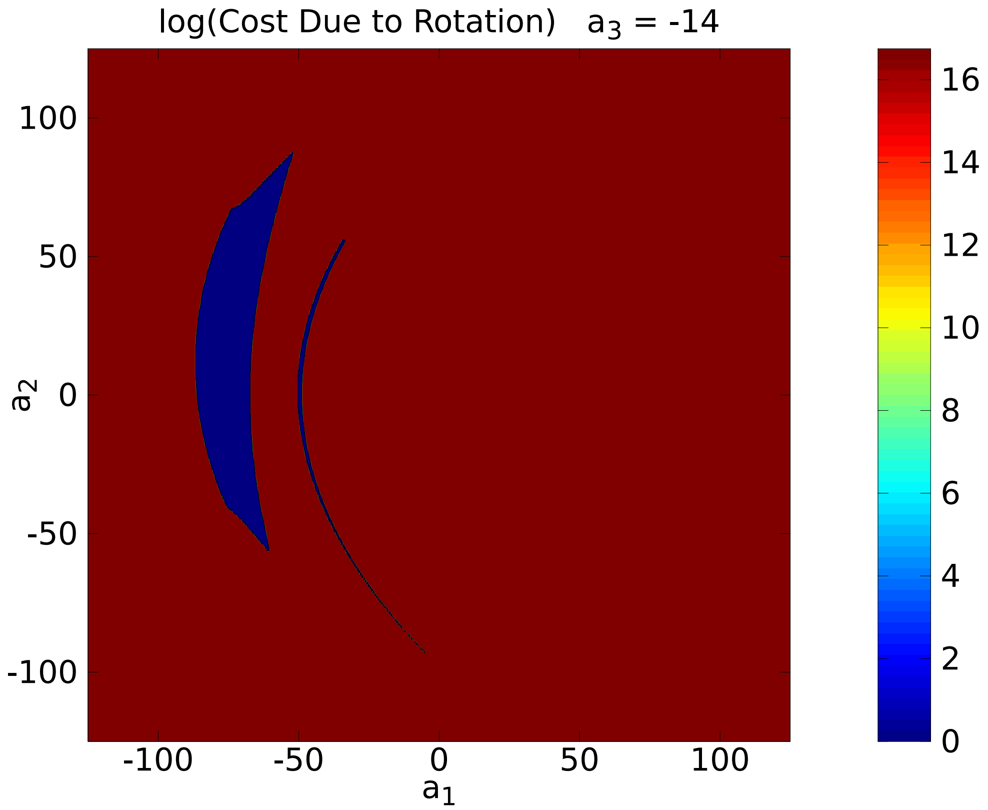


Figure B.13: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -14.0$.

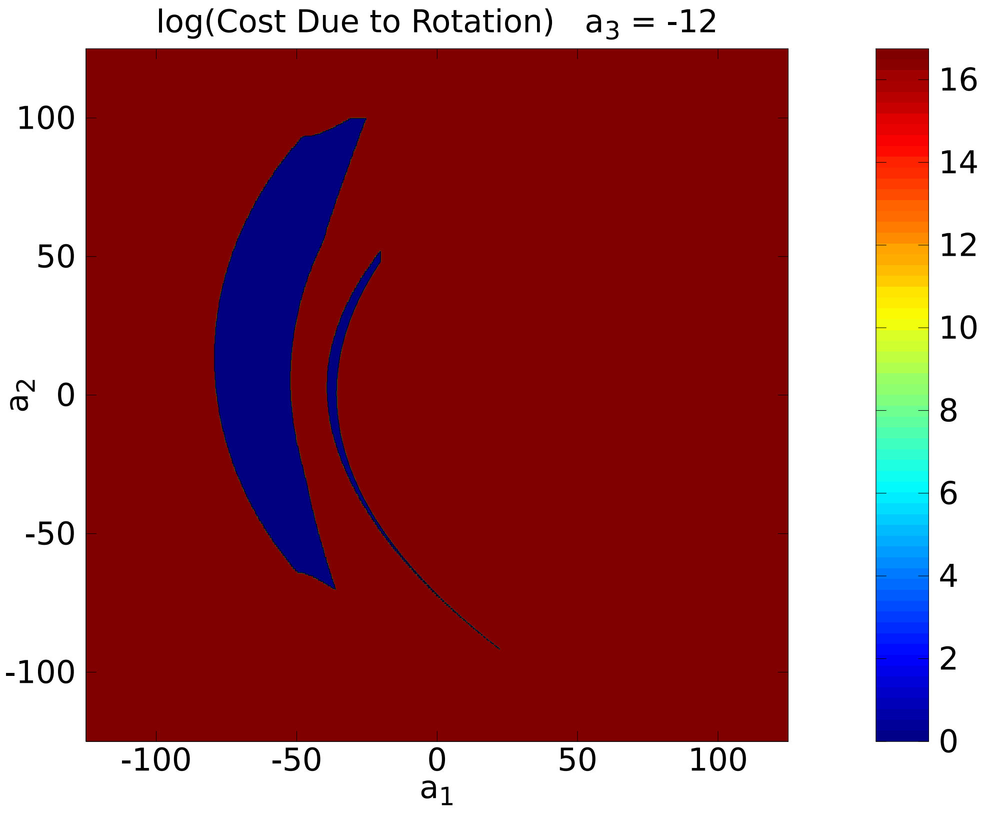


Figure B.14: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -12.0$.

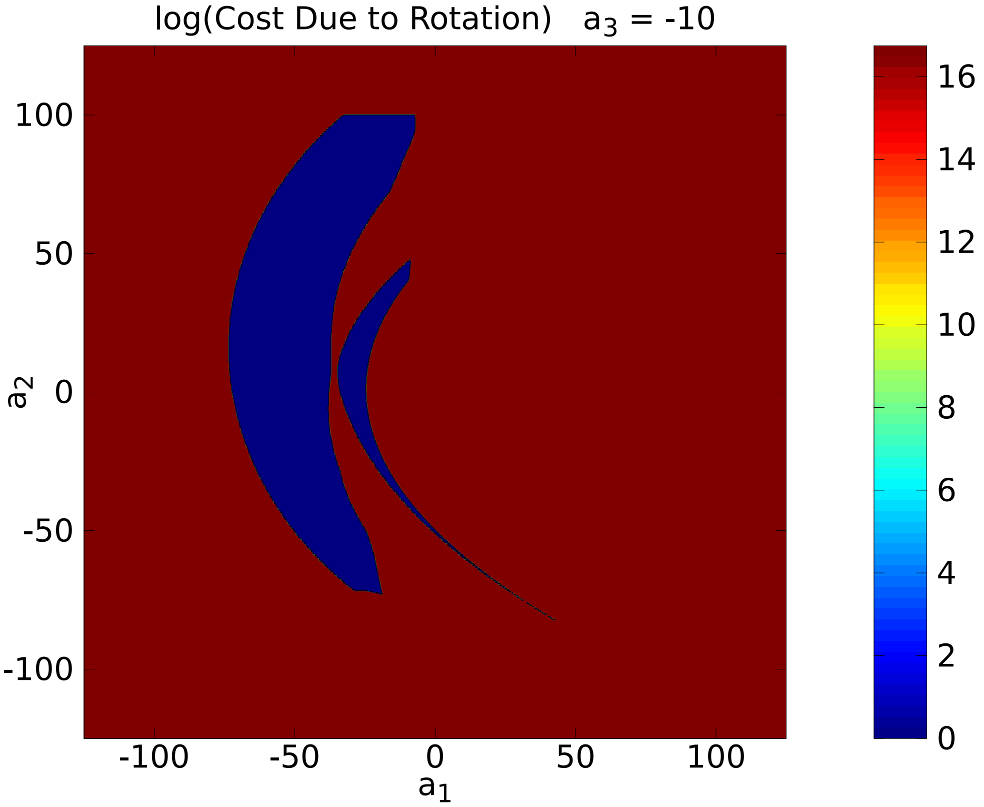


Figure B.15: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -10.0$.

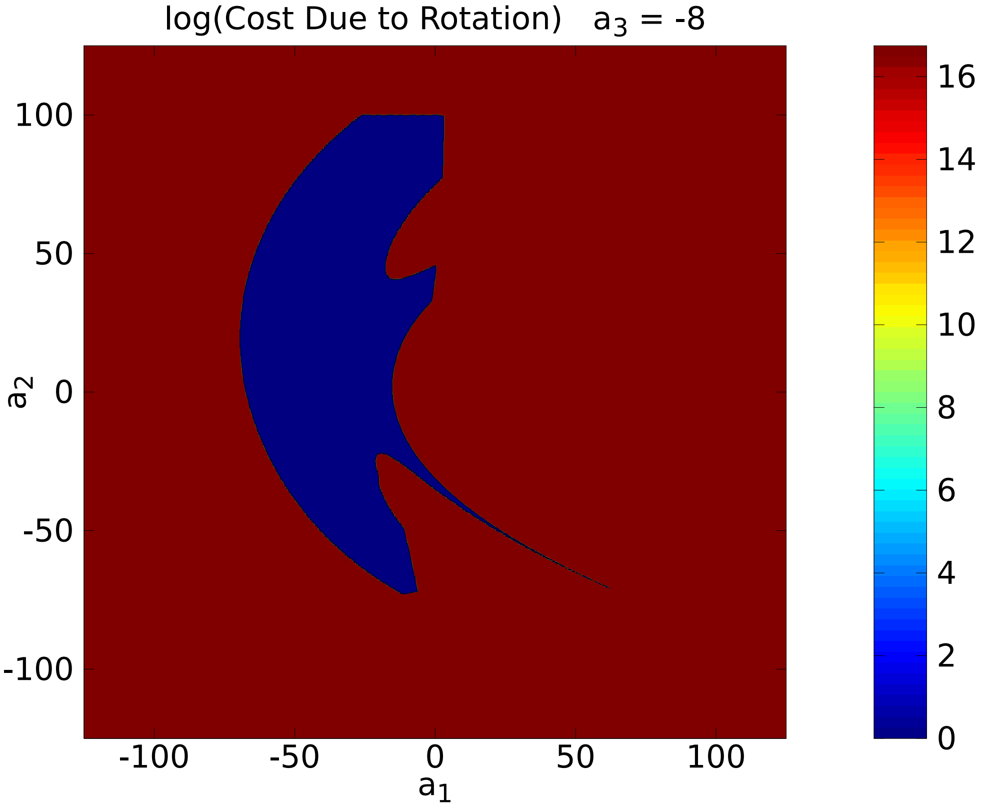


Figure B.16: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -8.0$.

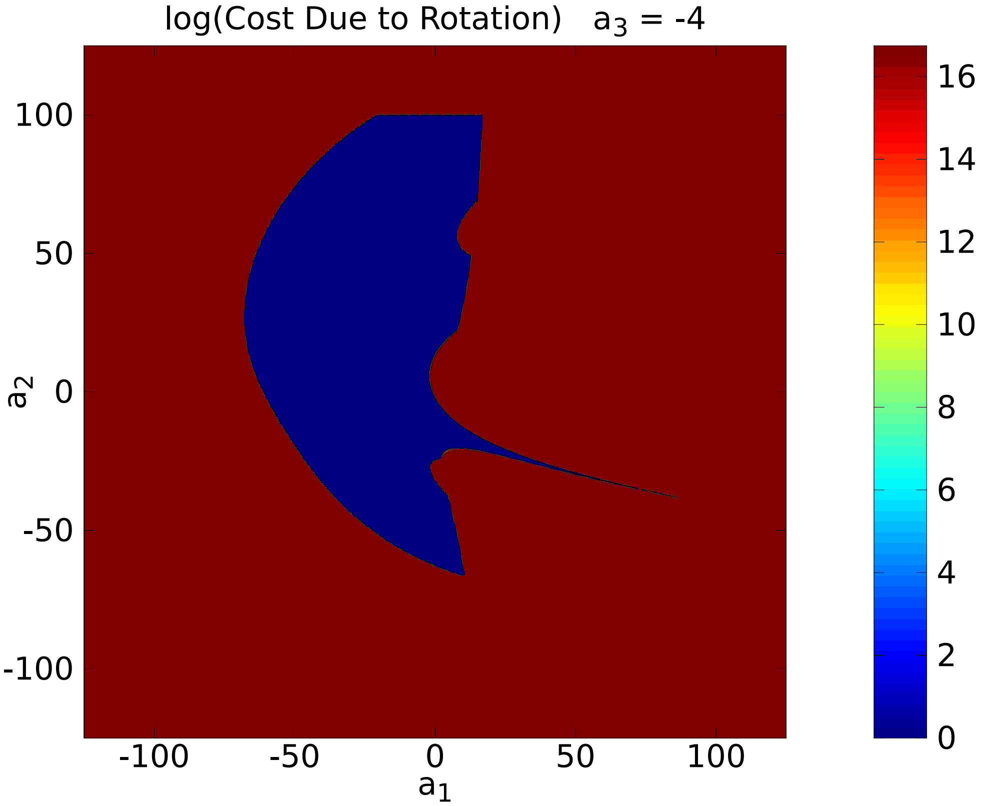


Figure B.17: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -4.0$.

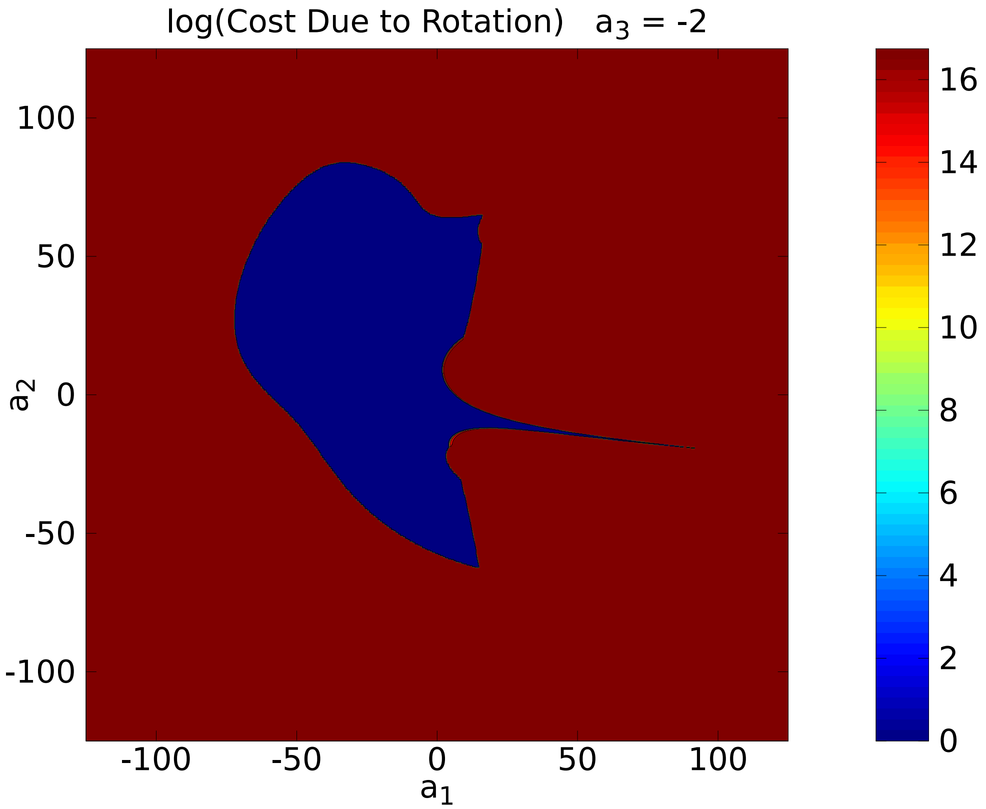


Figure B.18: Component of the total cost function due to the rotation angle as defined by Equation (4.4) for the plane where $a_3 = -2.0$.

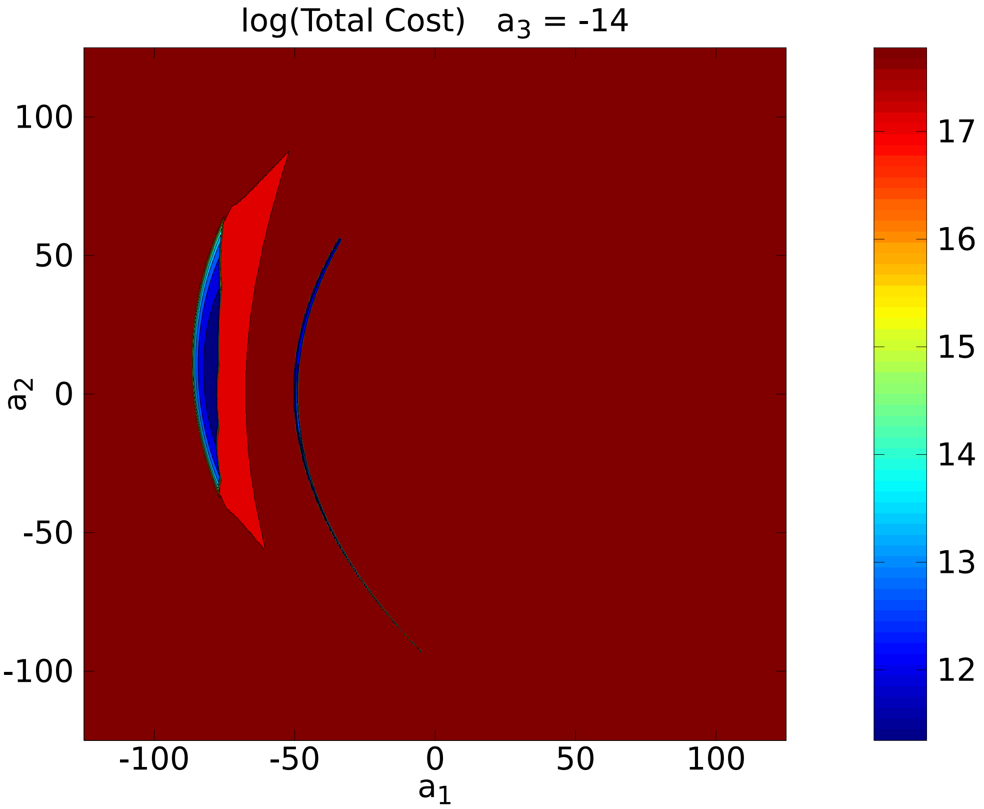


Figure B.19: Total of all three components of the cost function (Equations (4.1), (4.3), and (4.4)) for the plane where $a_3 = -14.0$.

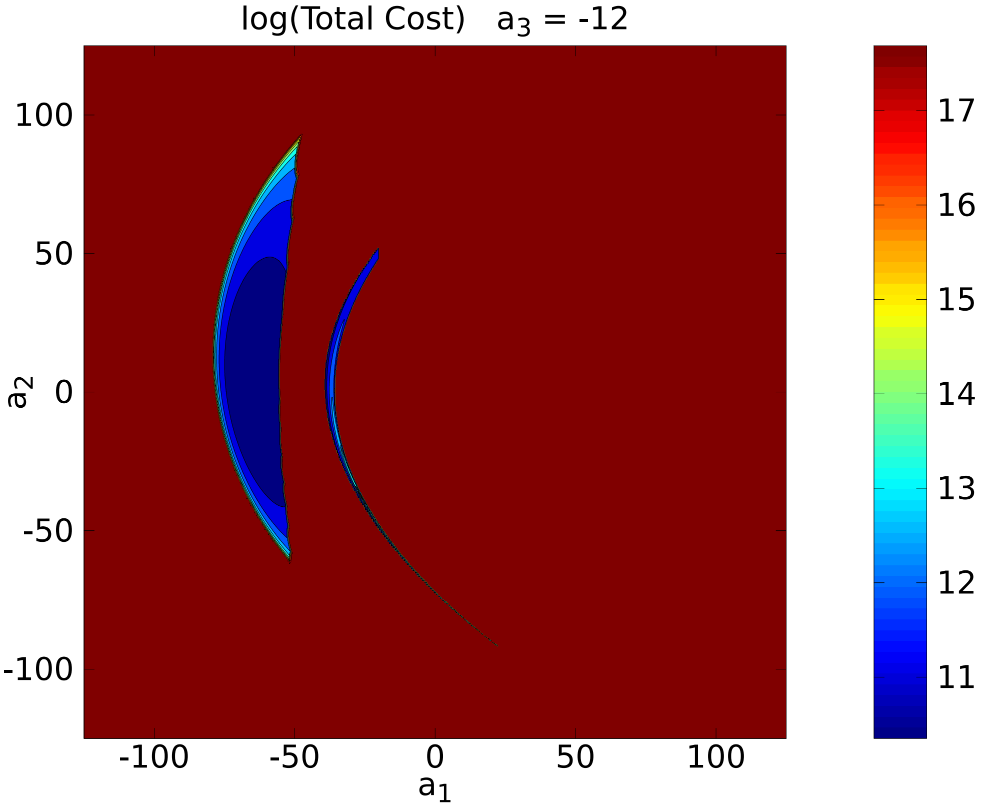


Figure B.20: Total of all three components of the cost function (Equations (4.1), (4.3), and (4.4)) for the plane where $a_3 = -12.0$.

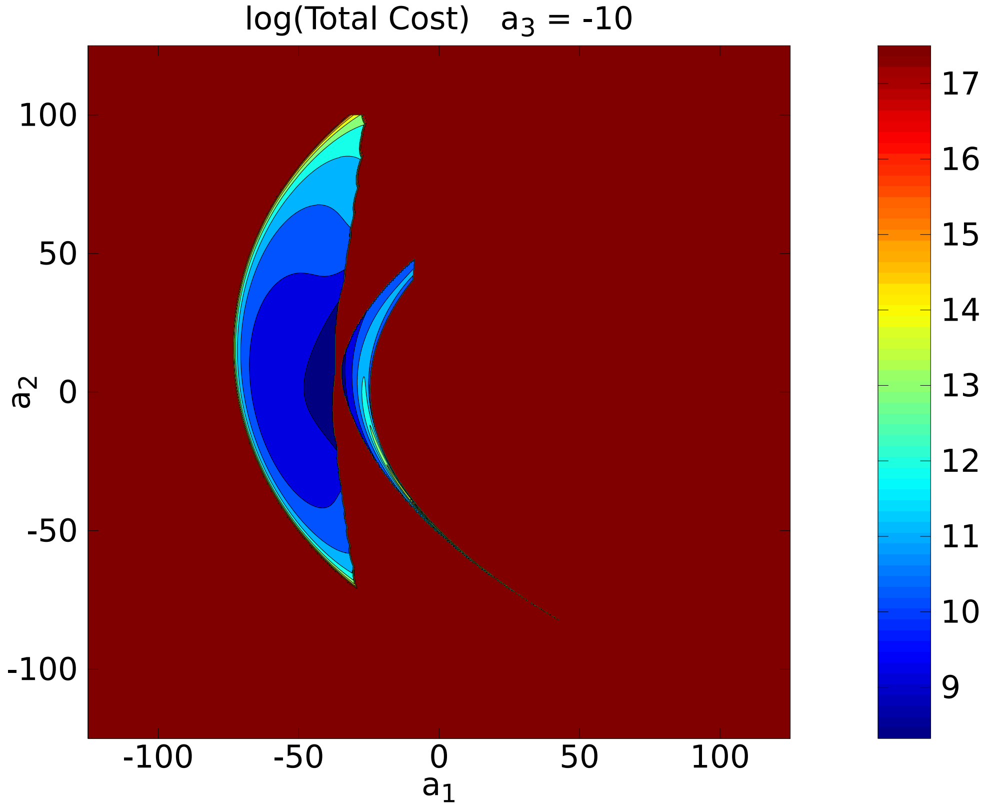


Figure B.21: Total of all three components of the cost function (Equations (4.1), (4.3), and (4.4)) for the plane where $a_3 = -10.0$.

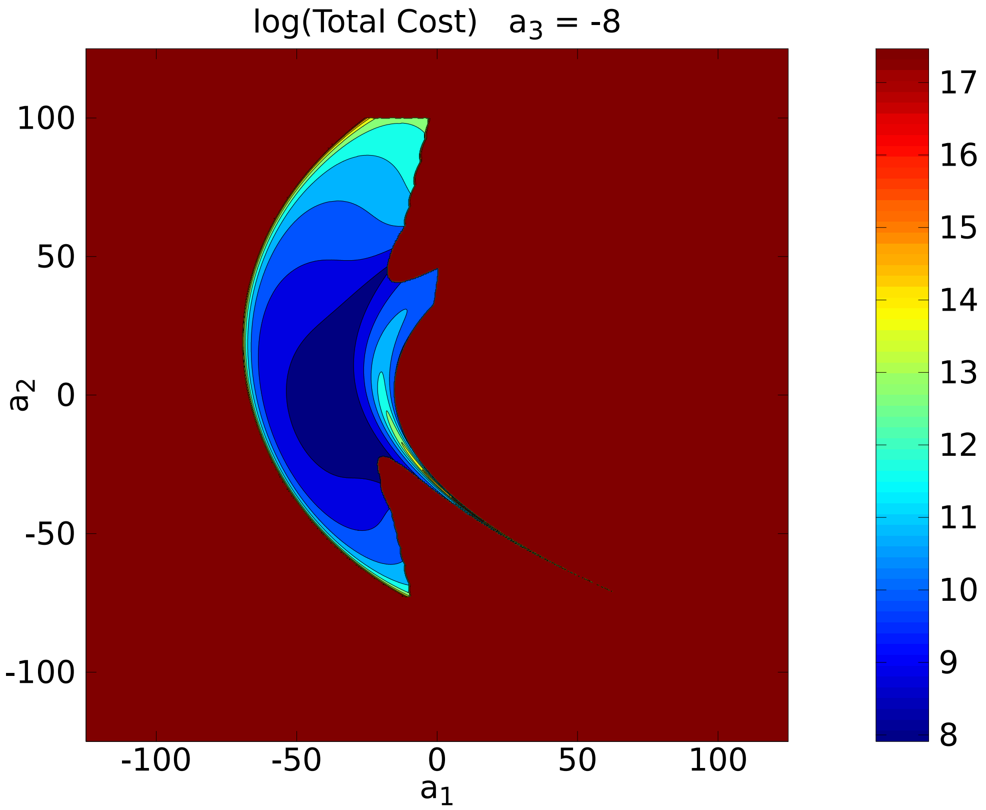


Figure B.22: Total of all three components of the cost function (Equations (4.1), (4.3), and (4.4)) for the plane where $a_3 = -8.0$.

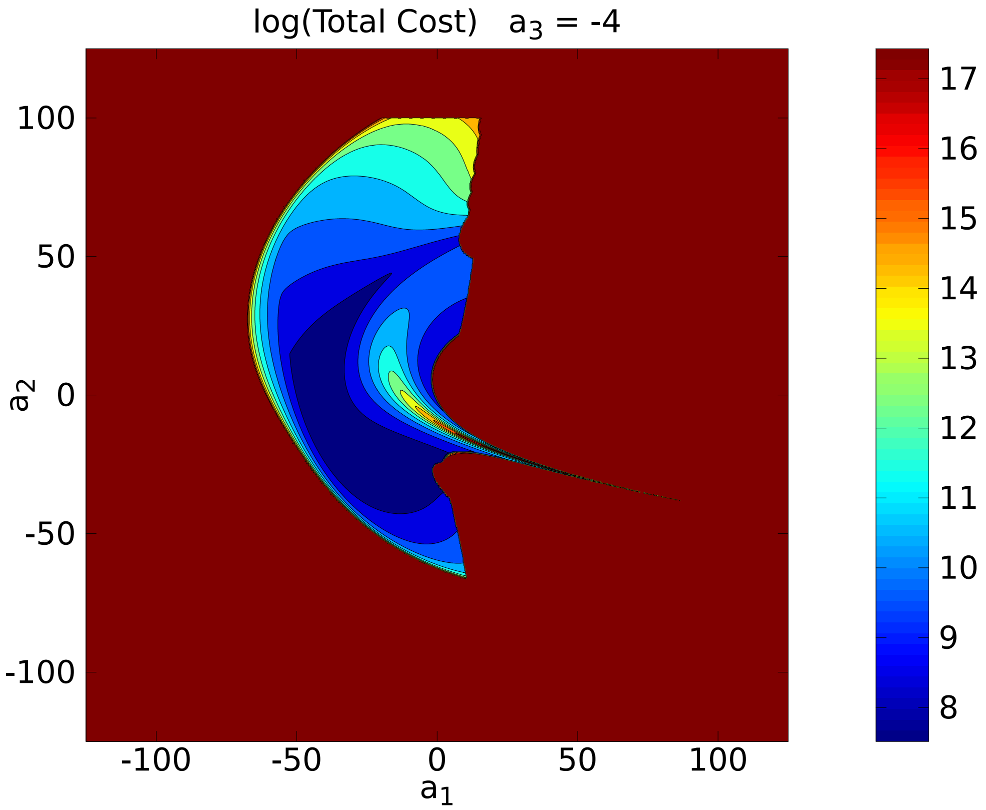


Figure B.23: Total of all three components of the cost function (Equations (4.1), (4.3), and (4.4)) for the plane where $a_3 = -4.0$.

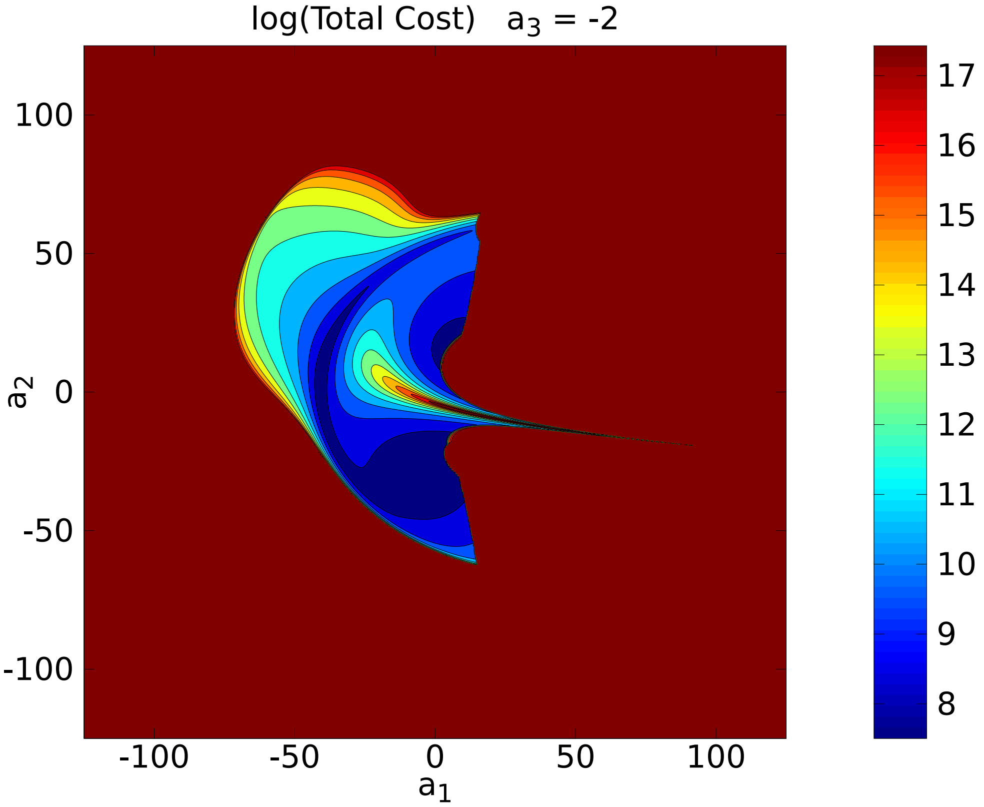


Figure B.24: Total of all three components of the cost function (Equations (4.1), (4.3), and (4.4)) for the plane where $a_3 = -2.0$.

APPENDIX C

ROBUST PATH PLANNING RESULTS

In this appendix, we show plots of results from experiments that compare paths planned using the cost function described in Chapter 4 to straight-line paths. The results from the first ten experiments of Table 4.1 were shown in Section 4.4. Figures C.1 - C.20 show the remainder of the results. These plots are sorted in decreasing order of the maximum error.

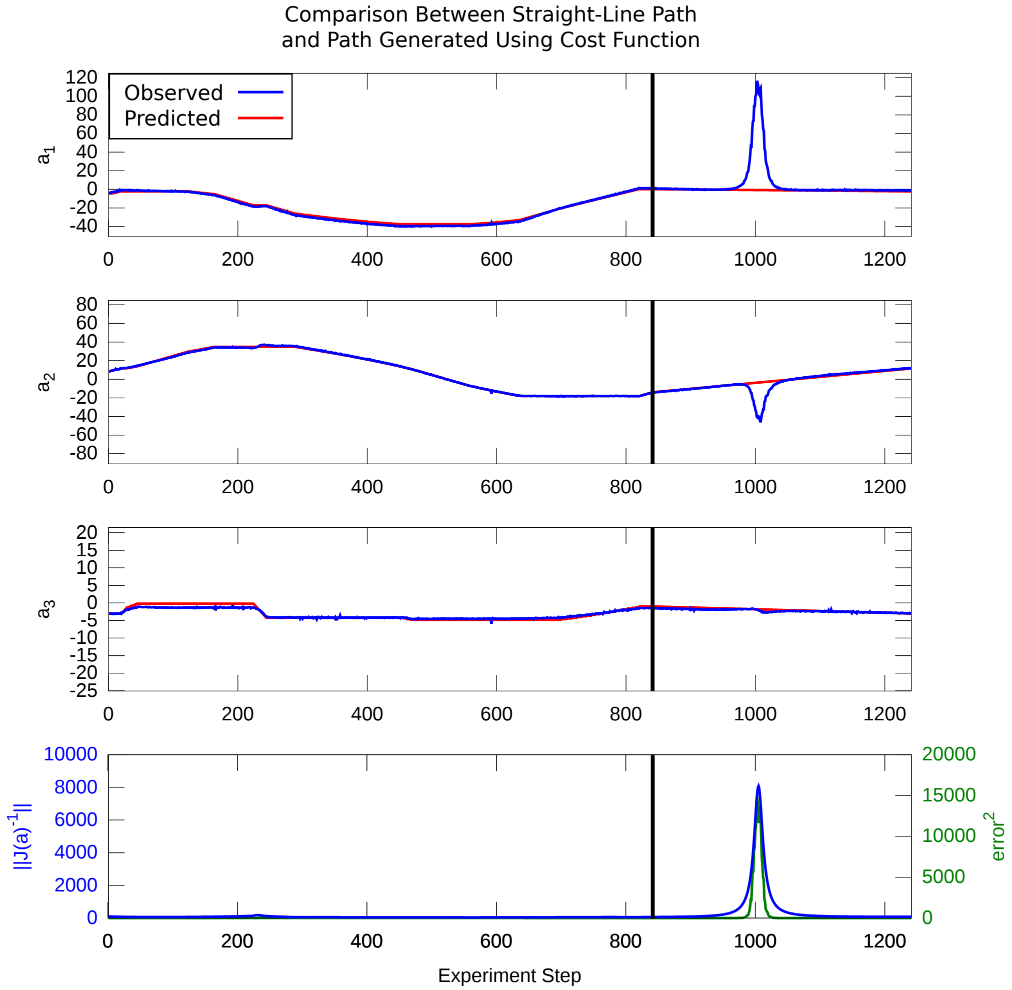


Figure C.1: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-2.2, 11.6, -3.0]$ and $[0.2, -14.0, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

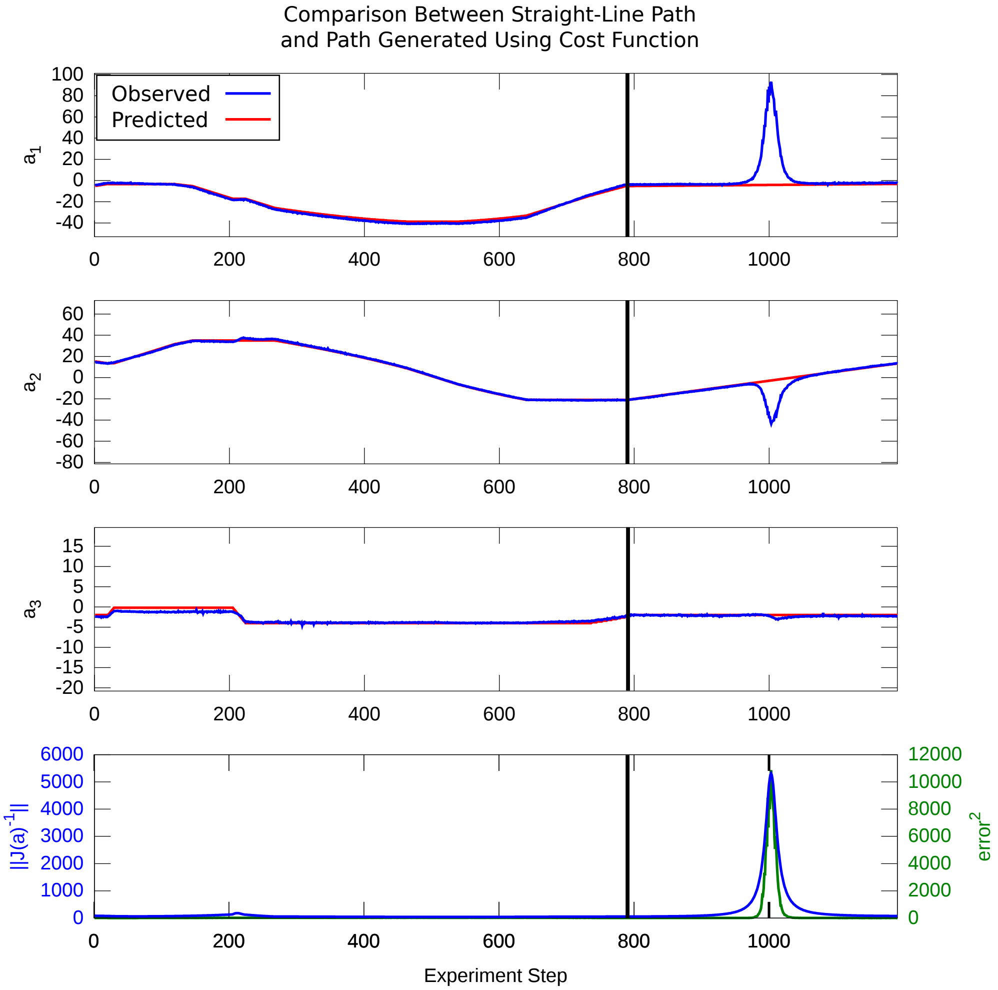


Figure C.2: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-3.2, 13.4, -2.0]$ and $[-5.2, -21.0, -2.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

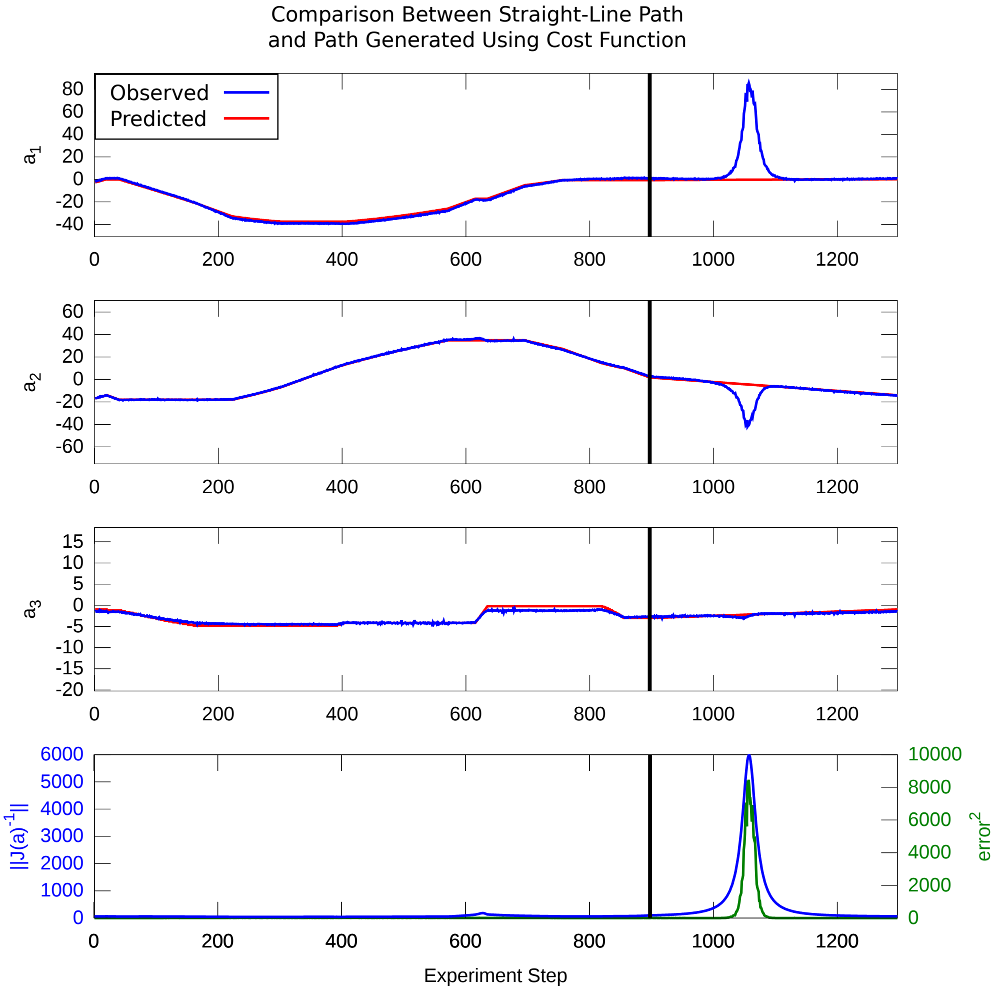


Figure C.3: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[0.2, -14.0, -1.0]$ and $[-0.6, 1.8, -3.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1, a))^{-1} \parallel^2$ (blue).

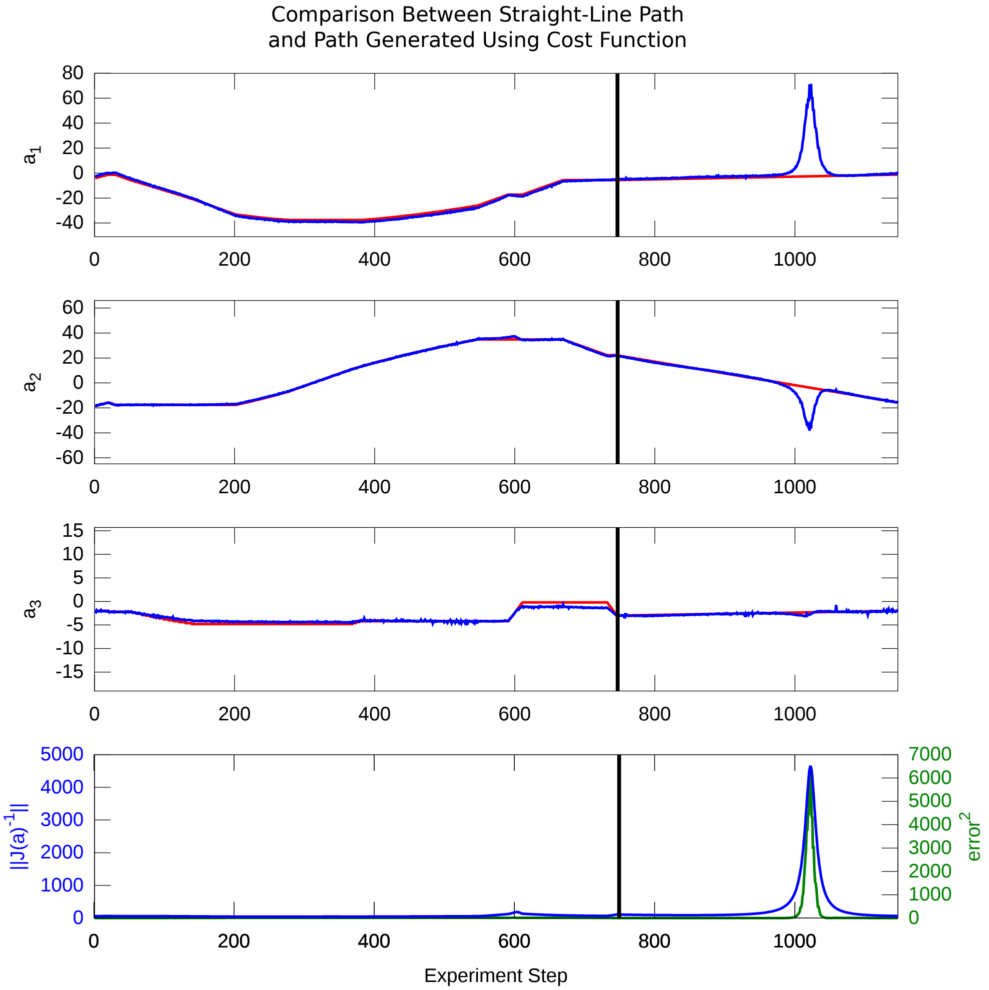


Figure C.4: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-1.2, -15.6, -2.0]$ and $[-5.6, 22.0, -3.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

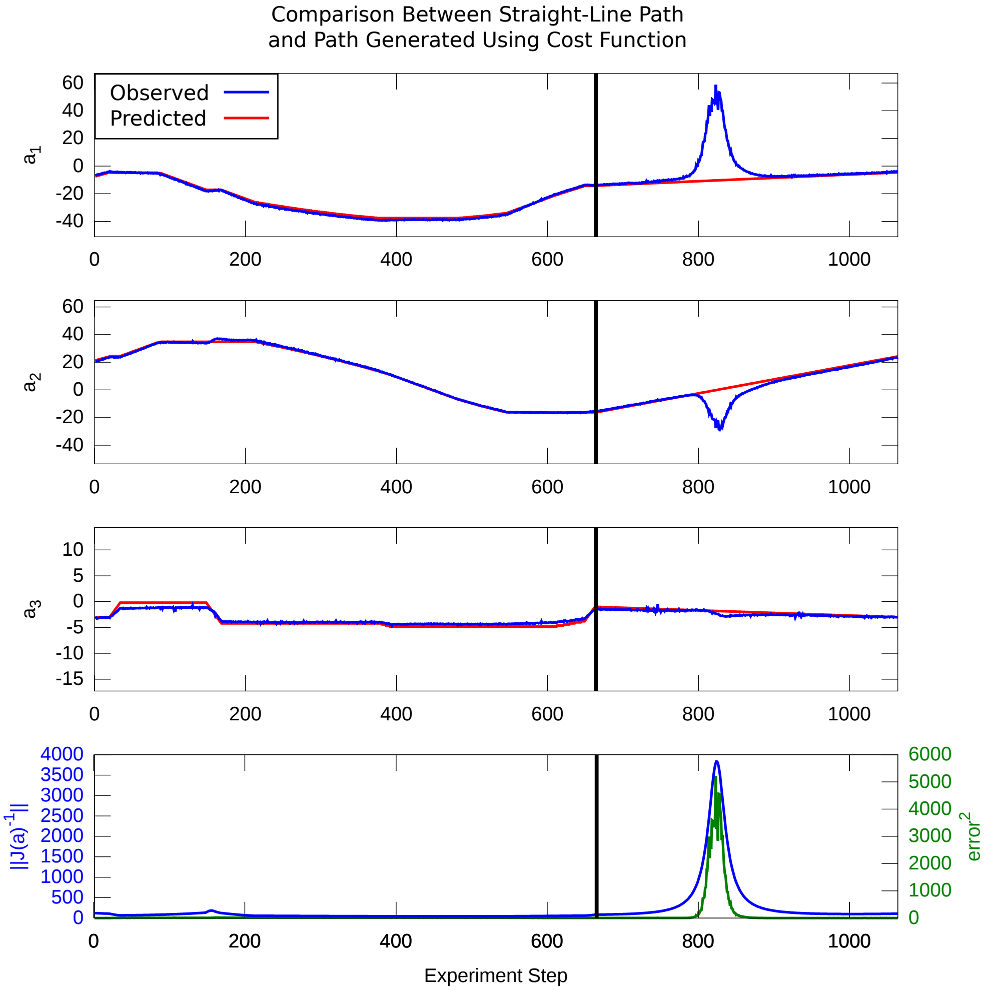


Figure C.5: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-4.6, 24.2, -3.0]$ and $[-14.2, -16.2, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

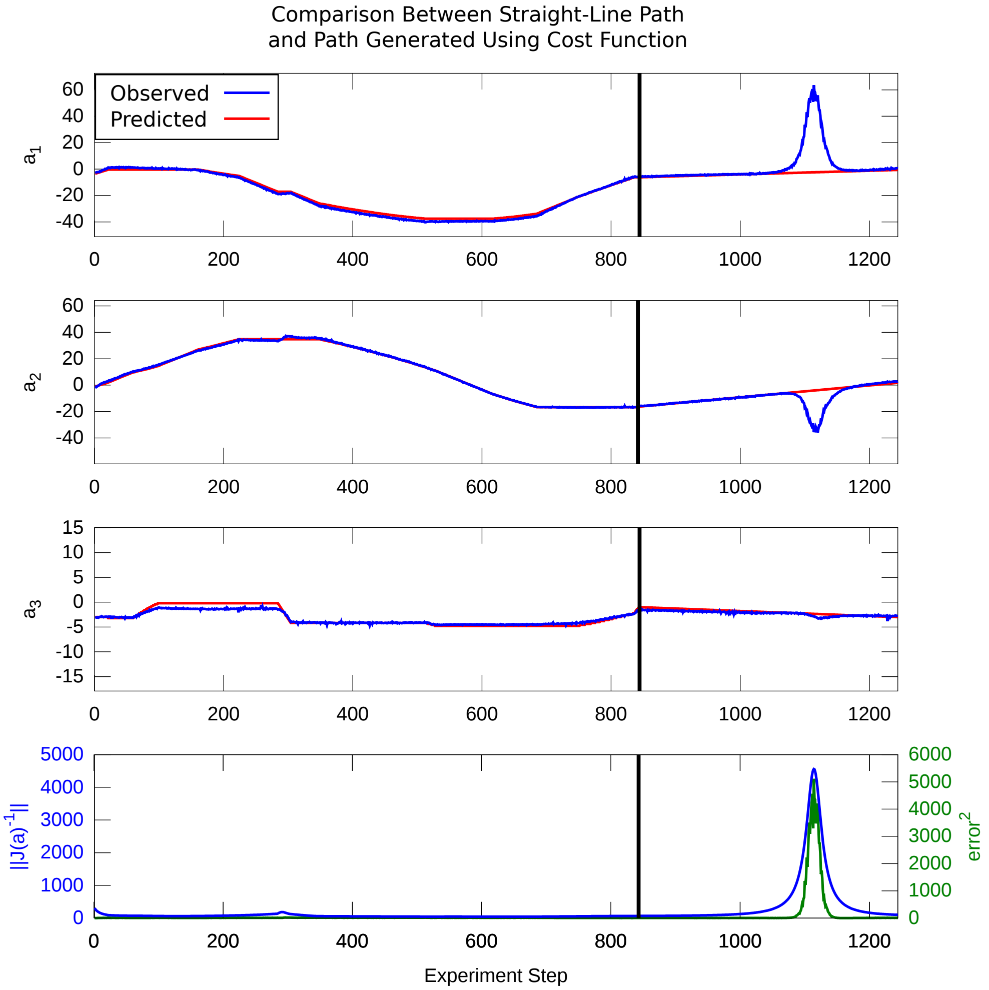


Figure C.6: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-0.6, 1.8, -3.0]$ and $[-6.2, -16.2, -1.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

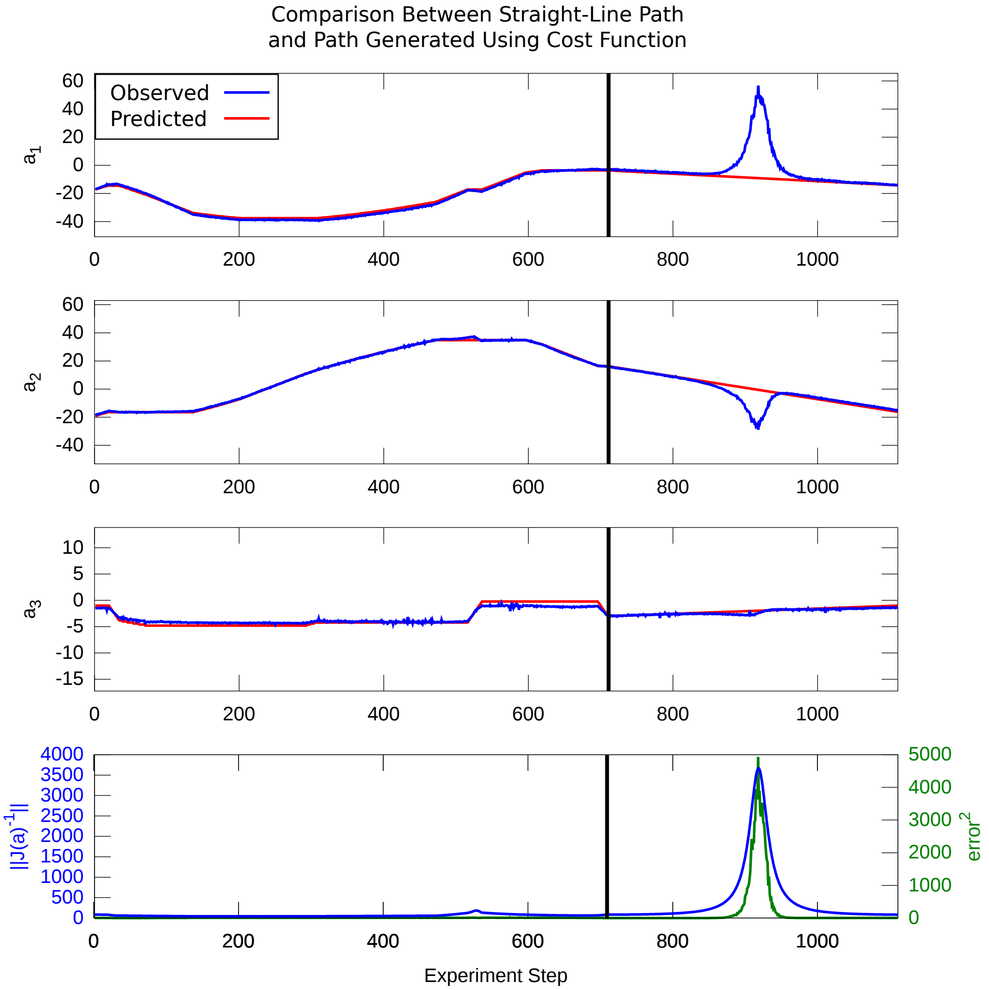


Figure C.7: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-14.2, -16.2, -1.0]$ and $[-3.6, 16.2, -3.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

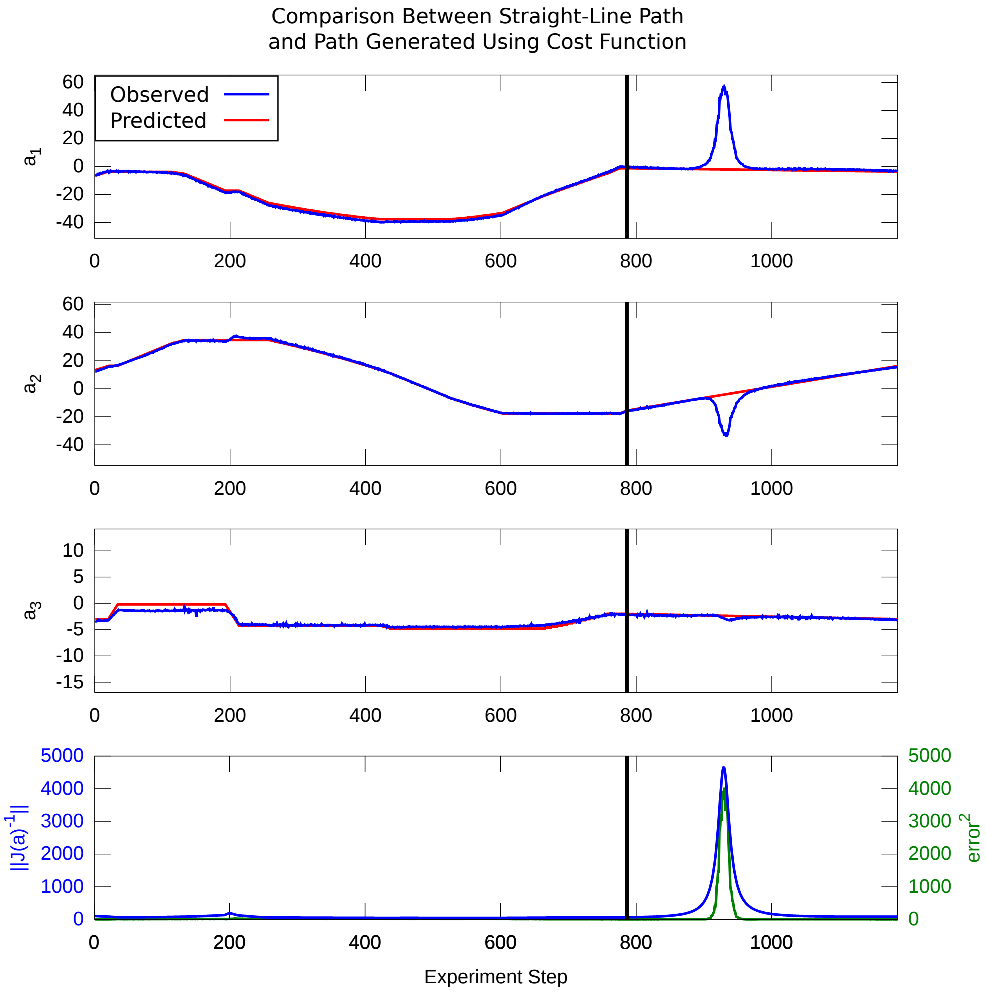


Figure C.8: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-3.6, 16.2, -3.0]$ and $[-1.2, -15.6, -2.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| (J(1,a))^{-1} \|^2$ (blue).

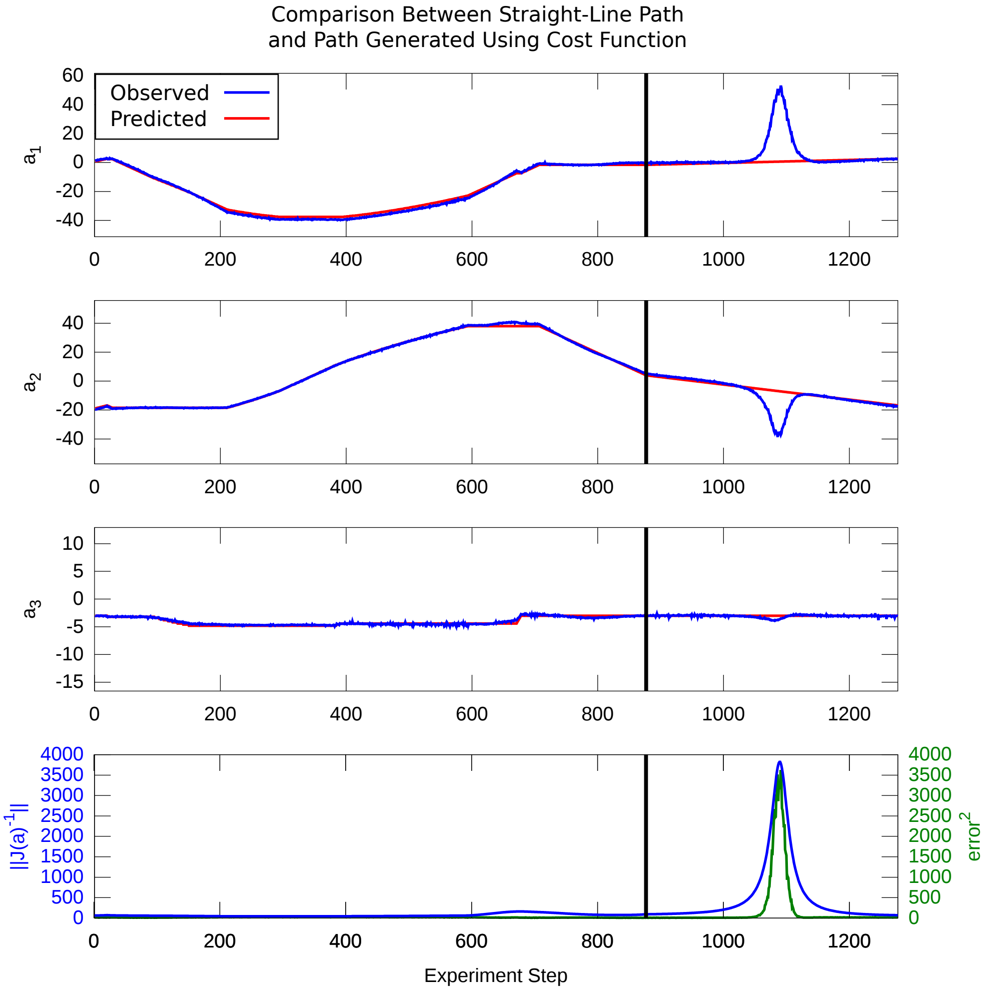


Figure C.9: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[2.6, -16.8, -3.0]$ and $[-1.6, 4.0, -3.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J\,(1, a))^{-1} \parallel^2$ (blue).

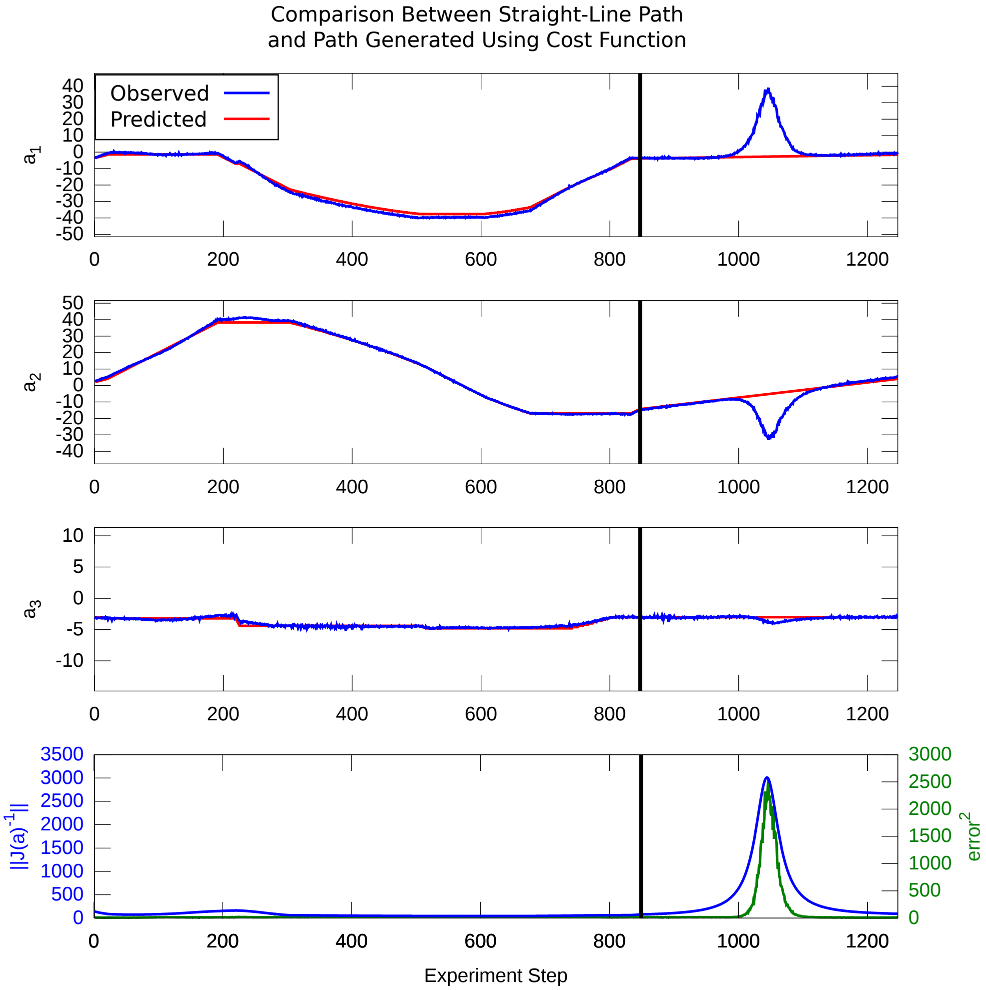


Figure C.10: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-1.6, 4.0, -3.0]$ and $[-3.8, -14.4, -3.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

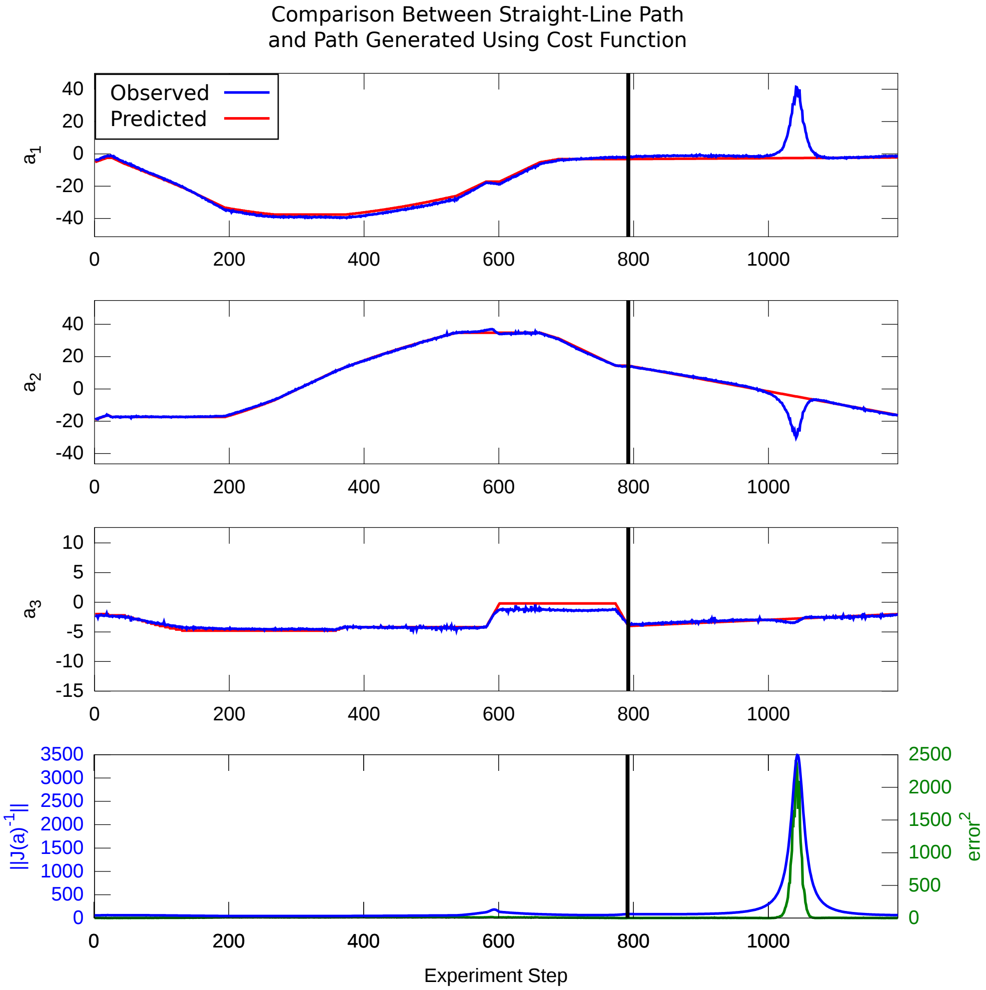


Figure C.11: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-2.2, -16.2, -2.0]$ and $[-3.2, 14.4, -4.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1, a))^{-1} \parallel^2$ (blue).

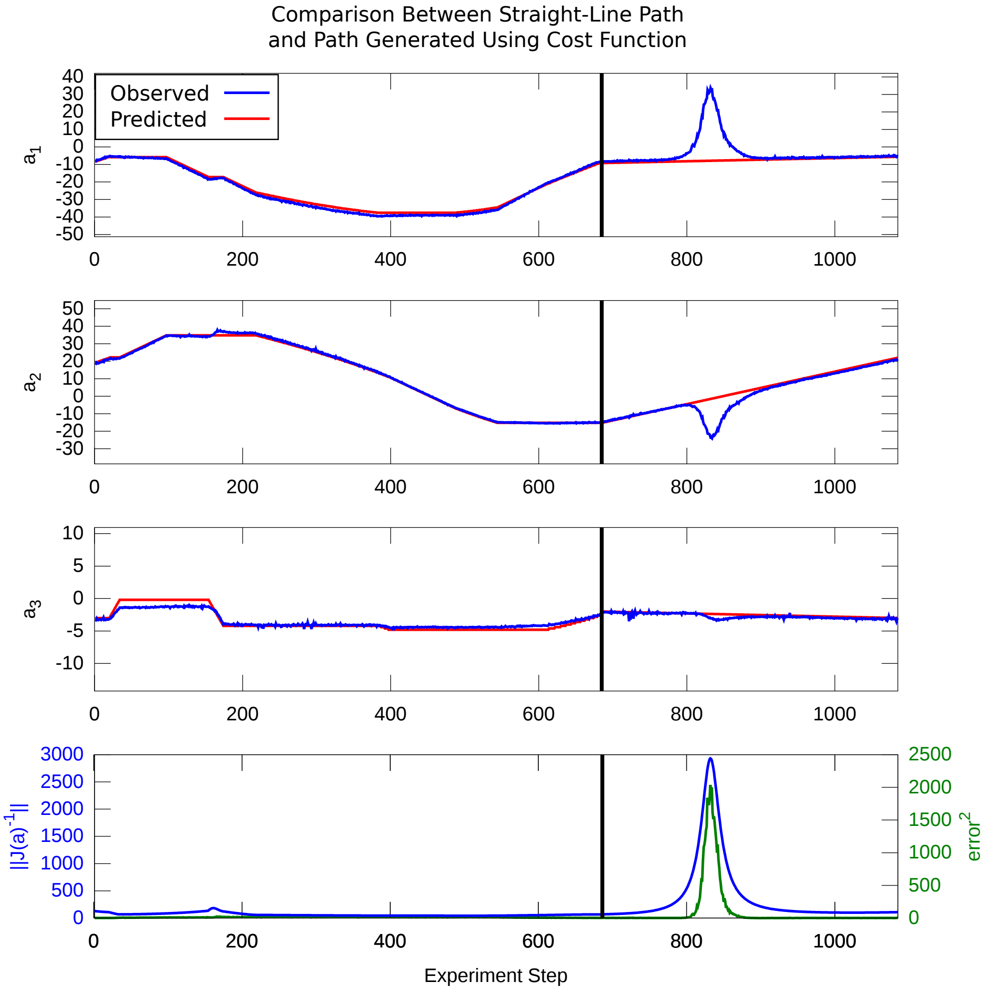


Figure C.12: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-5.6, 22.0, -3.0]$ and $[-9.2, -15.2, -2.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

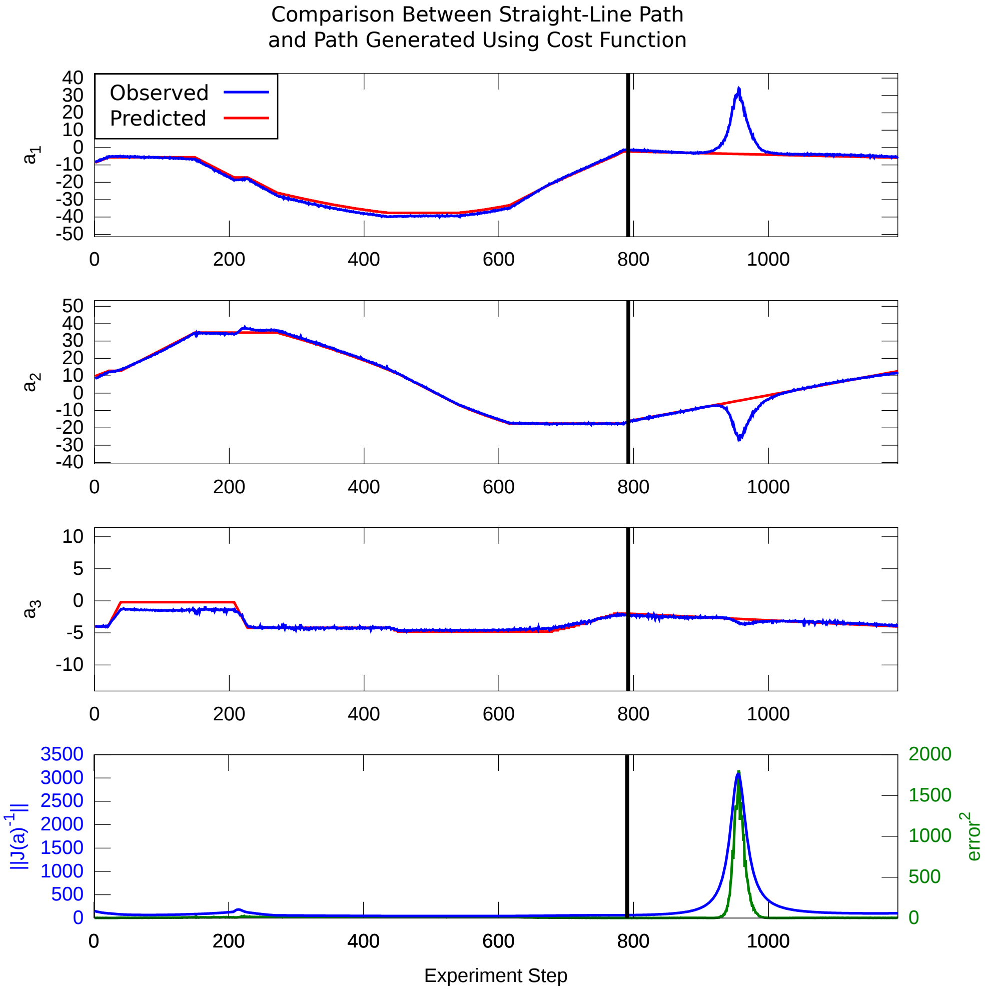


Figure C.13: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-5.8, 12.6, -4.0]$ and $[-2.2, -16.2, -2.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

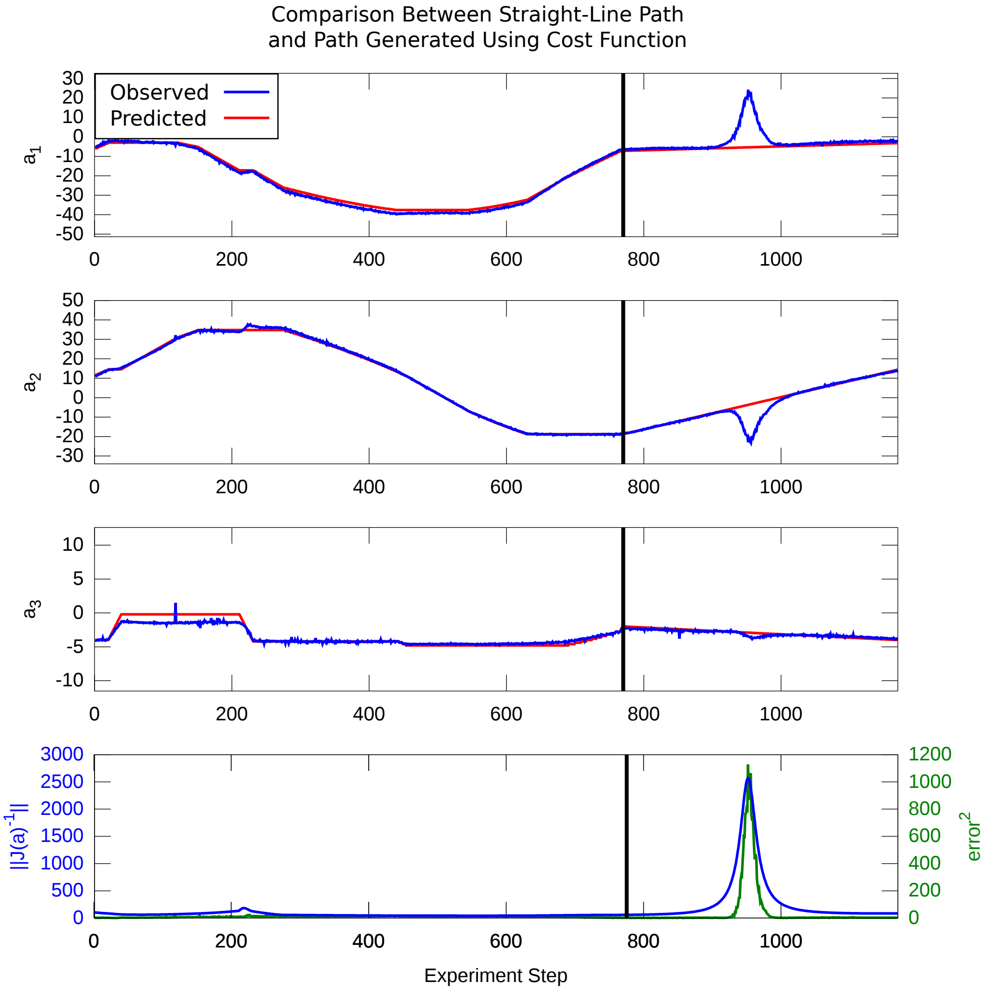


Figure C.14: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-3.2, 14.4, -4.0]$ and $[-7.2, -18.8, -2.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

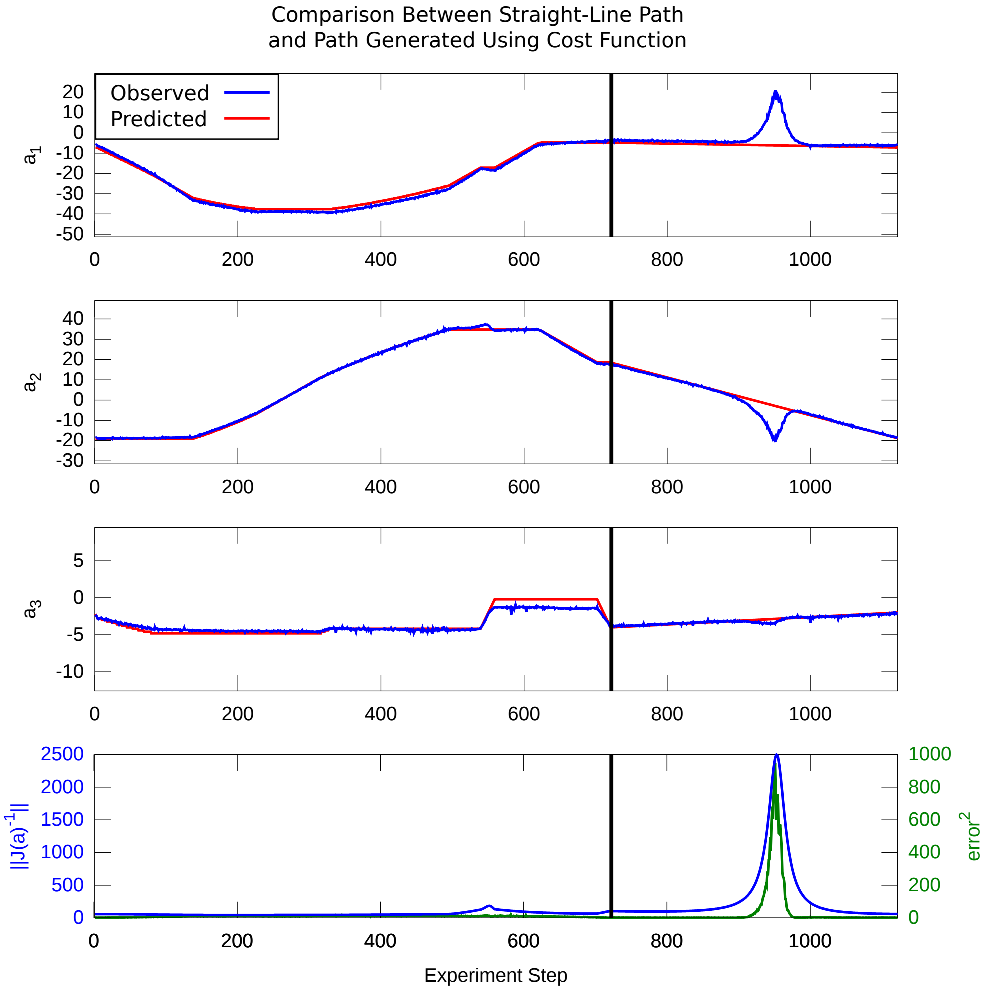


Figure C.15: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-7.2, -18.8, -2.0]$ and $[-4.8, -18.4, -4.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

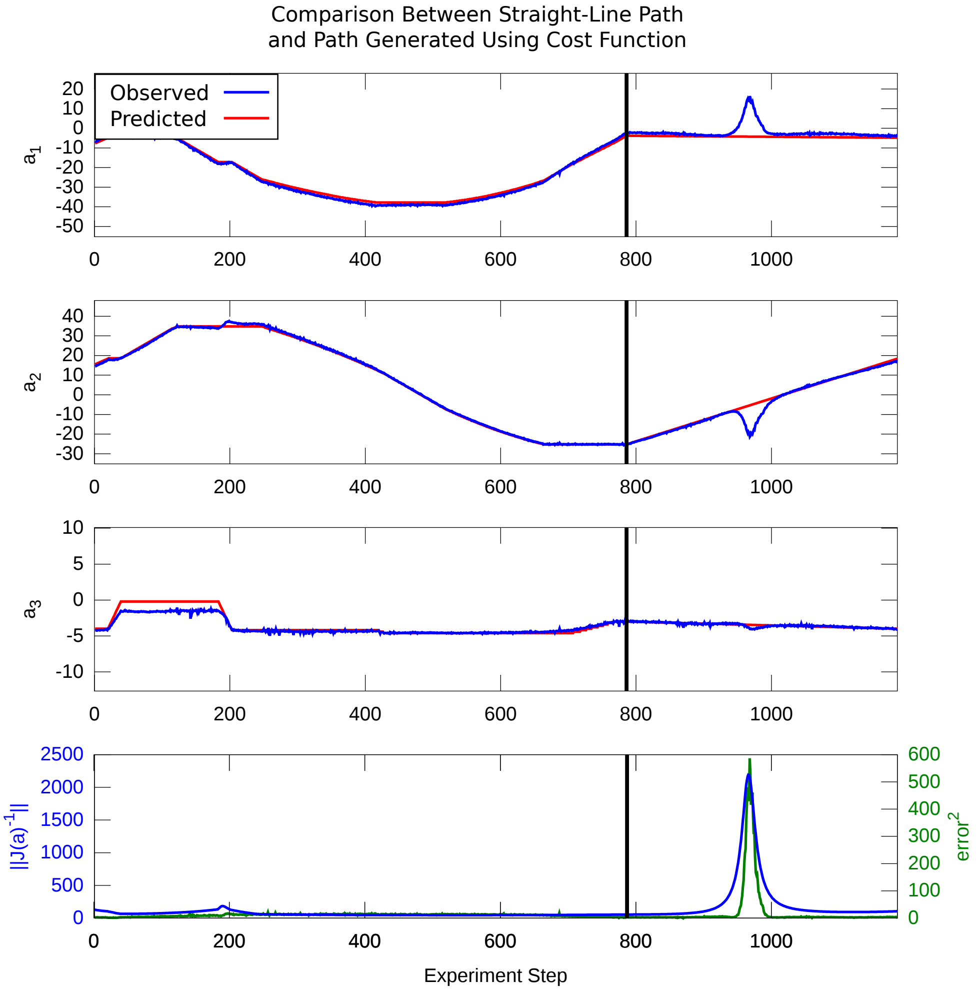


Figure C.16: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-4.8, 18.4, -4.0]$ and $[-3.8, -25.2, -3.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| (J(1,a))^{-1} \|^2$ (blue).

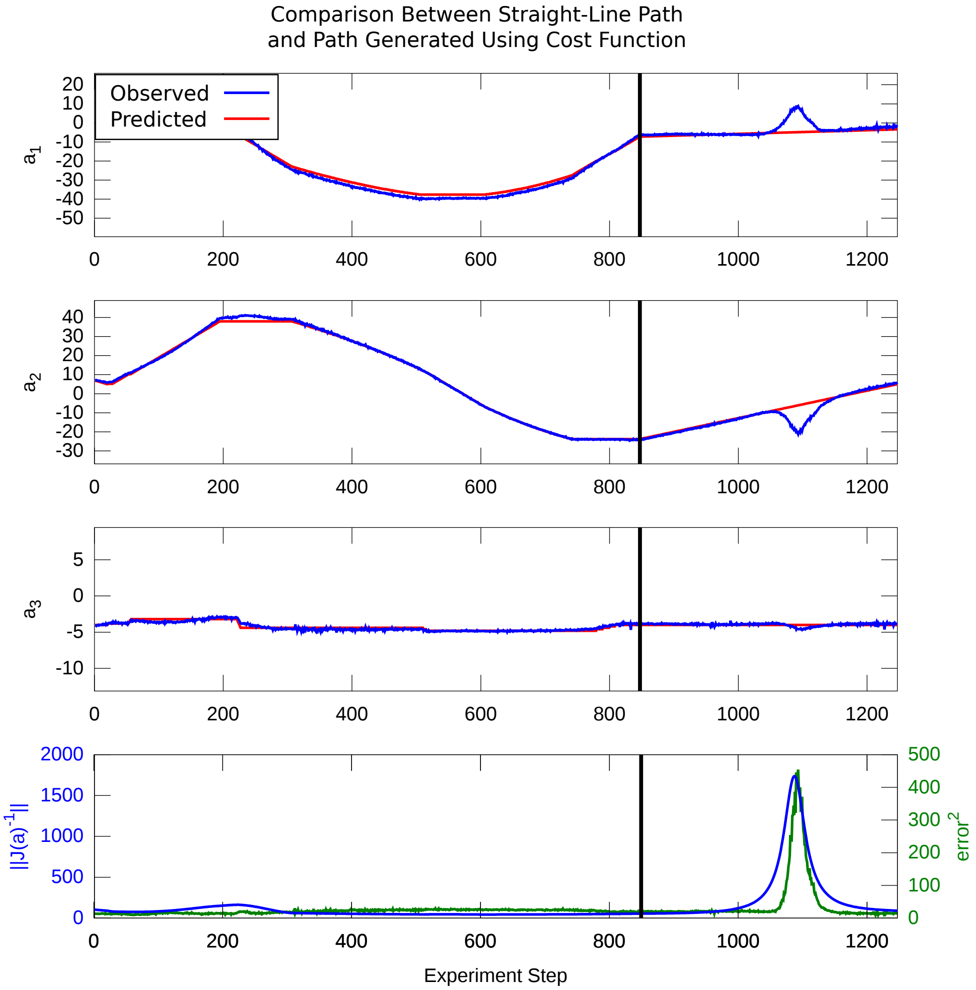


Figure C.17: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-3.4, 5.0, -4.0]$ and $[-7.2, -23.8, -4.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\parallel (J(1,a))^{-1} \parallel^2$ (blue).

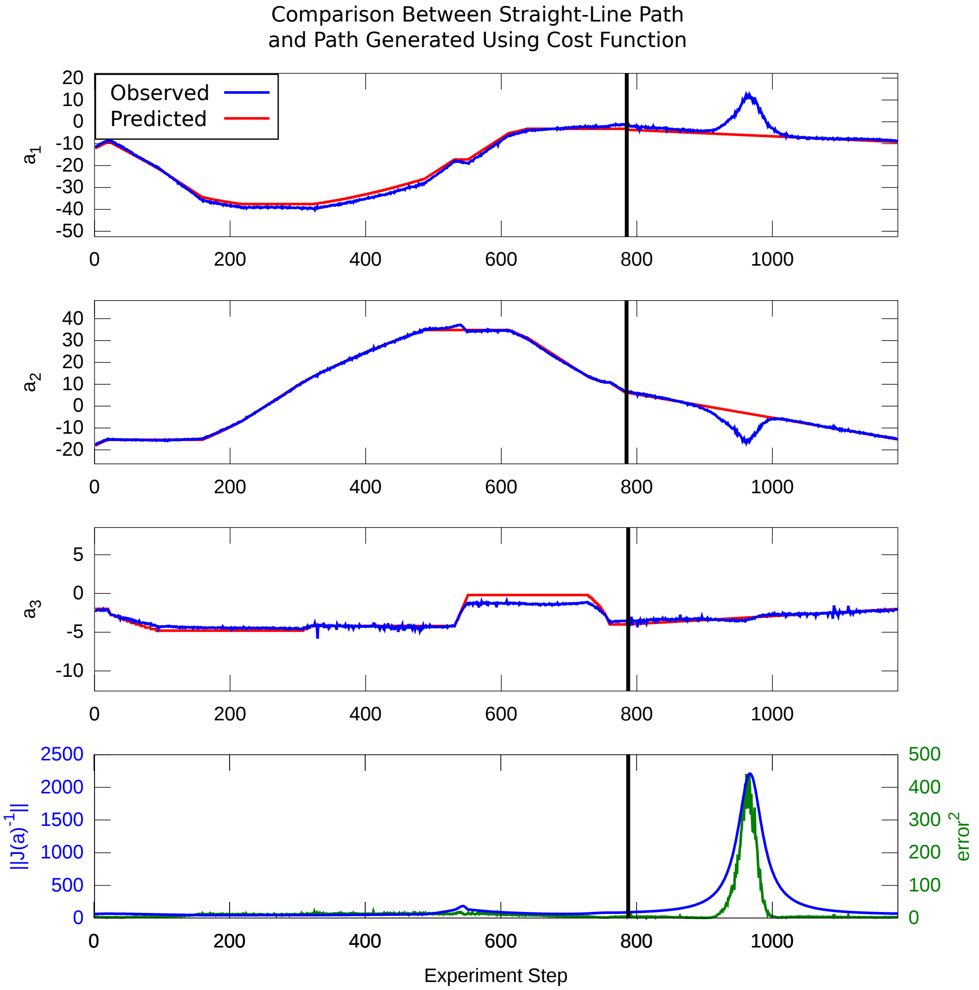


Figure C.18: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-9.2, -15.2, -2.0]$ and $[-3.6, 6.2, -4.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

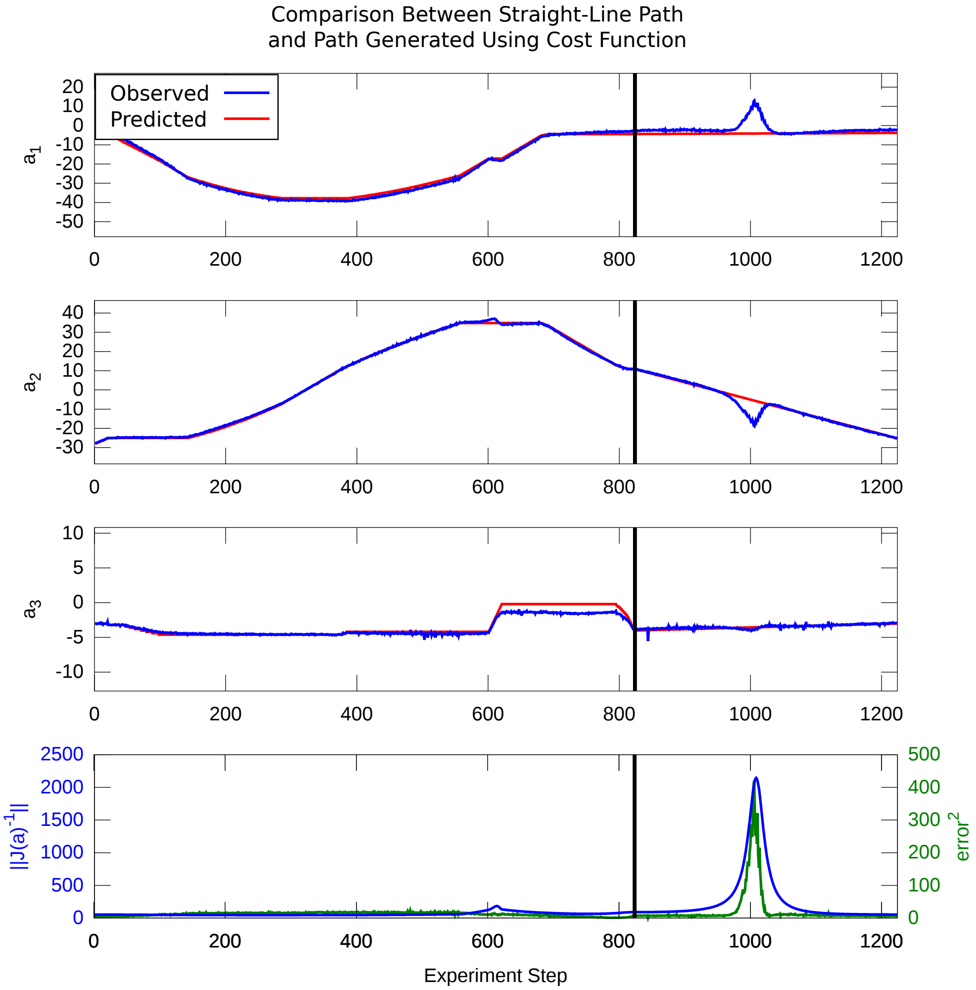


Figure C.19: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-3.8, -25.2, -3.0]$ and $[-4.4, 10.8, -4.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

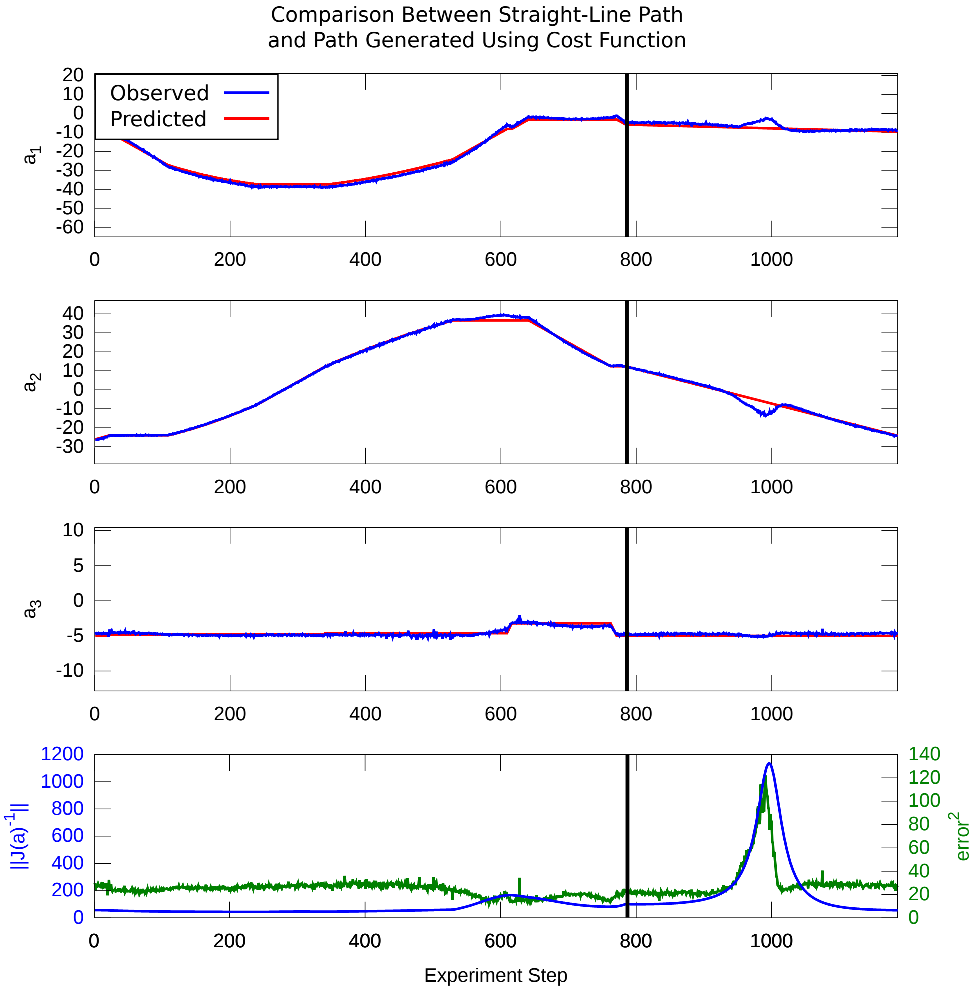


Figure C.20: The top three plots show a comparison between the predicted configuration (red) and observed configuration (blue) for each of the three components in $\mathcal{A}$ when we follow both a straight-line path and a path planned using our cost function between the points $[-9.6, -24.2, -5.0]$ and $[-5.9, 12.2, -5.0]$. The region to the left of the black line shows the robust path while the region to the right shows the straight-line path. The bottom plot shows the $error^2$ (green) and the $\| \left(J\left(1, a\right)\right)^{-1} \|^2$ (blue).

APPENDIX D

CAMERA CALIBRATION AND IMAGE PROCESSING

In this appendix, we describe the image processing and data analysis software that has been implemented to measure the shape of the metal strip used in experiments. First, we describe the procedure for calculating the extrinsic camera parameters which allow for converting between the world frame and the image plane. This requires the implementation of image segmentation procedures to detect markers placed along the metal strip. Once segmentation is performed, the detected markers are transformed into nondimensional coordinates and the position is compared against the theoretical positions to determine the difference between the predicted shape and the observed shape.

D.1 Camera Sensor Calibration

The intrinsic parameters for the camera are calculated using OpenCV's cv-CalibrateCamera2 function for a set of 70 checkerboard images. This produces a matrix with intrinsic parameters and the distortion parameters for the Flea2 camera. The calibration procedure produced the following values:

$$IntrinsicParameters = \begin{bmatrix} 5.21 \times 10^2 & 0 & 3.21 \times 10^2 \\ 0 & 5.21 \times 10^2 & 2.43 \times 10^2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DistortionParameters = \begin{bmatrix} -1.67 \times 10^{-1} \\ 1.55 \times 10^{-2} \\ 3.32 \times 10^{-3} \\ -3.97 \times 10^{-3} \\ 5.11 \times 10^{-2} \end{bmatrix}$$

Once the table is aligned, we calculate extrinsic camera parameters that allow the transformation between the world frame and the image plane. This

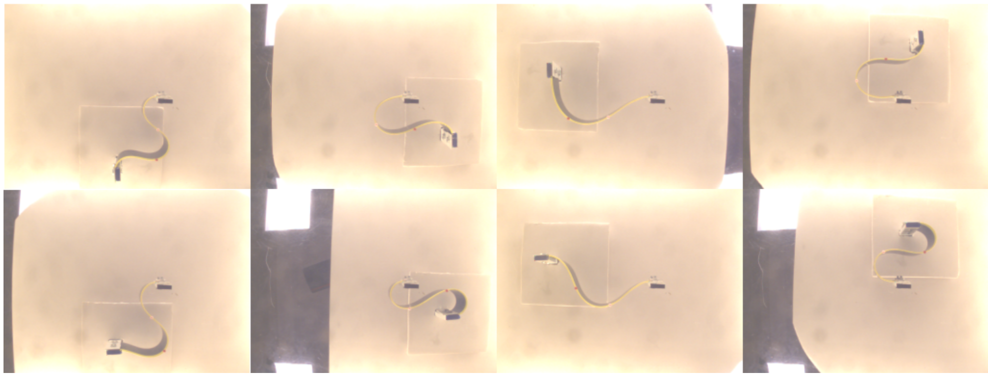


Figure D.1: Alignment images for calibrating extrinsic camera parameters

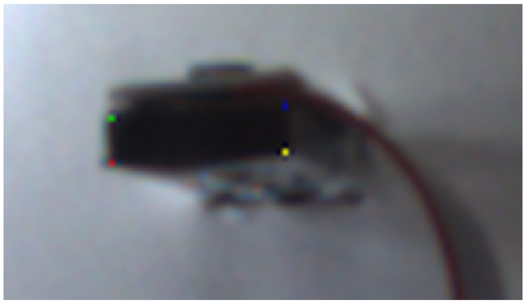


Figure D.2: Labeled end effector used for calibration

is done by capturing an image of the robot's end effector in 32 configurations. This consists of 16 different positions with two rotations at each location. Figure D.1 shows eight images of configurations used for calibration. For each image, a pixel at each of the four corners of the end effector is labeled using the colors red, green, blue, and yellow. This procedure produces 128 points in the image frame that correspond to 128 points calculated using the end effector's position that is read from the robot. Figure D.2 shows how the image is labeled. Once the points are gathered, OpenCV's function cvFindExtrinsicCameraParams2 and cvRodrigues are used to calculate the translation vector, rotation vector, and rotation matrix.

We calculated rotation matrix and translation vector to be

$$RotationMatrix = \begin{bmatrix} 2.38 \times 10^{-2} & -9.99 \times 10^{-1} & 6.31 \times 10^{-3} \\ 9.99 \times 10^{-1} & 2.40 \times 10^{-2} & 4.31 \times 10^{-2} \\ -4.32 \times 10^{-2} & 5.27 \times 10^{-3} & 9.99 \times 10^{-1} \end{bmatrix}$$

$$TranslationVector = \begin{bmatrix} -1.91 \times 10^1 \\ -6.16 \times 10^2 \\ -3.99 \times 10^1. \end{bmatrix}$$

D.2 Image Segmentation and Quantitative Comparison

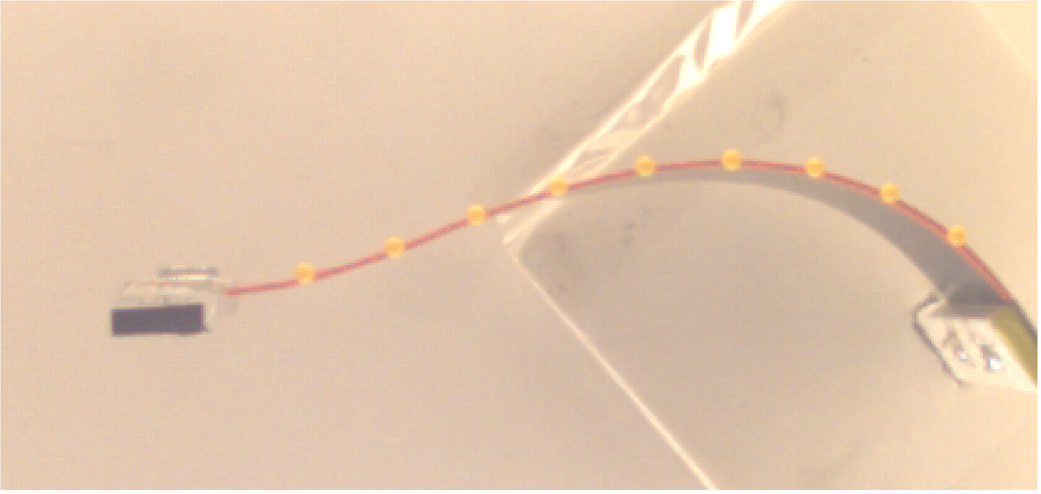


Figure D.3: This image shows the metal strip that has been instrumented with red tape and yellow map tacks that are spaced at 2cm.

By implementing image segmentation algorithms, we can identify the position of a discrete number of points along the rod and compare these positions to their predicted positions. The position in the image plane of the yellow map tacks is transformed to a position in the world frame. By comparing the positions of the markers in the world frame to the predicted position for these same intervals for the predicted strip, we have an intuitive error measurement. Figures D.3 and D.4 show the metal strip along with annotations of the observed and predicted position of the markers. The distance metric between these two sets of positions is a more objective error metric than simply using visual comparison.

To detect the markers along the rod, we have implemented an algorithm that first detects the metal strip and then identifies the yellow markers that are attached to the metal strip. First, each image of the elastic rod is converted to the Hue Saturation Value (HSV) color space. This color space has

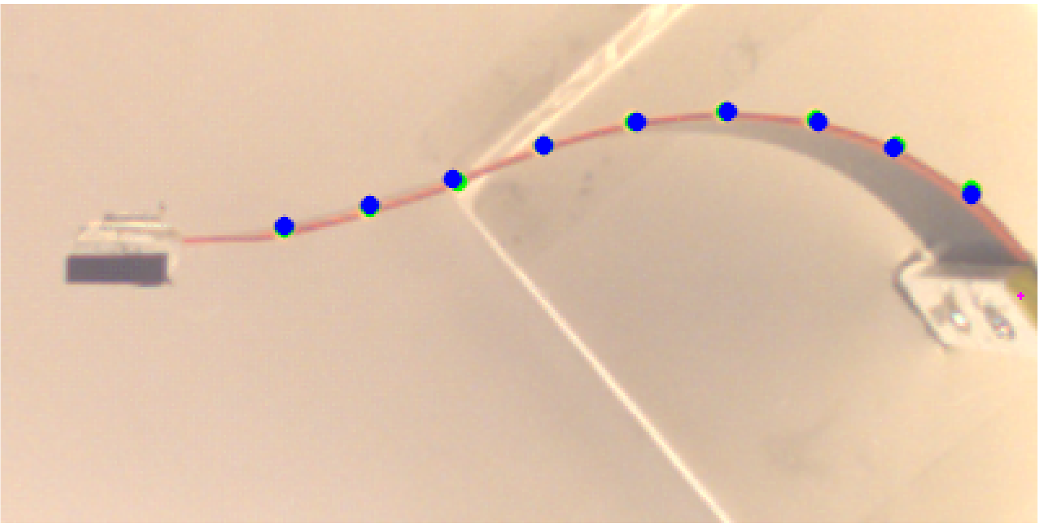
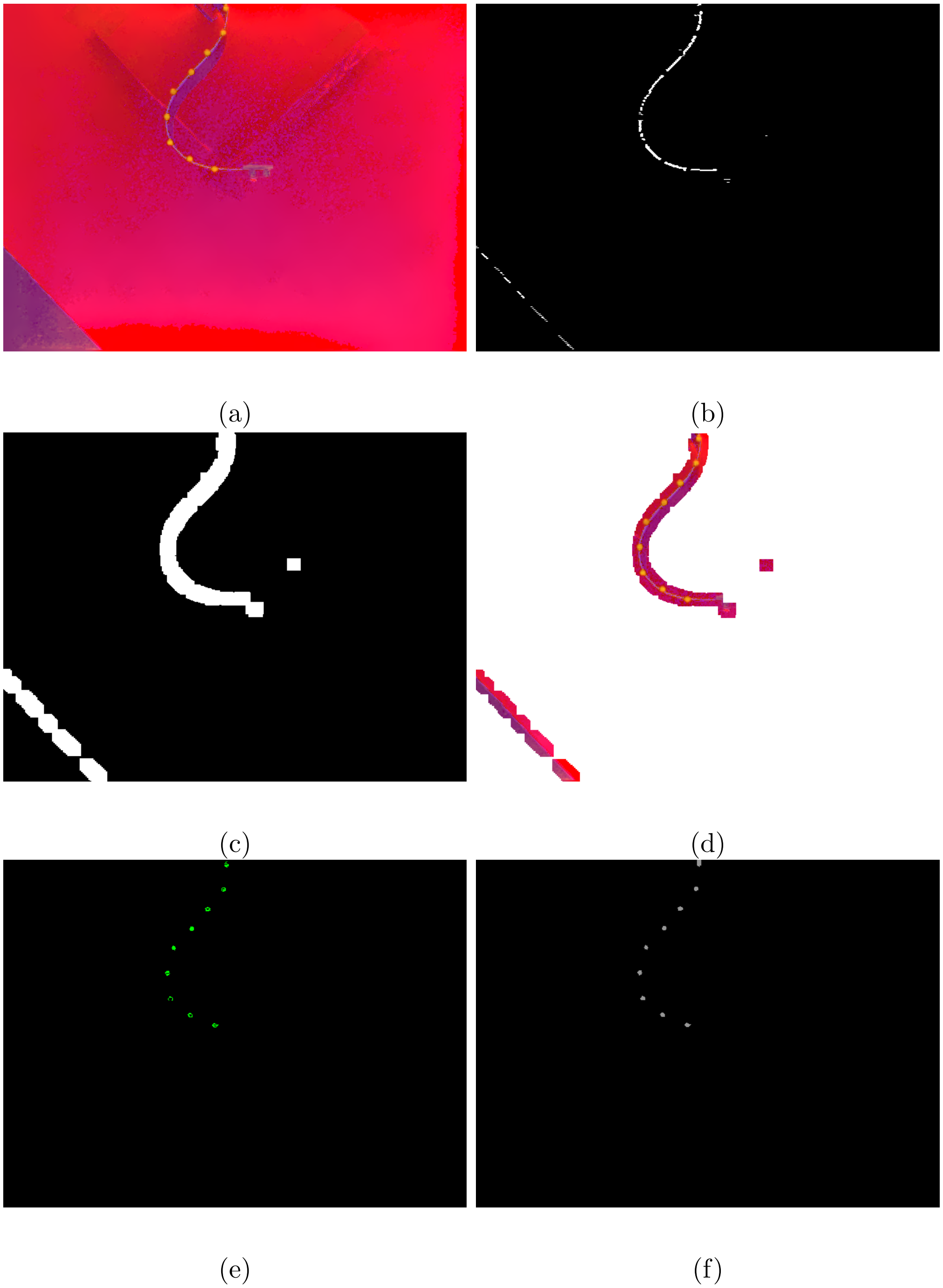


Figure D.4: This image shows the same metal strip, but it has been annotated with blue circles that show the predicted position of the map tacks and green circles that indicate the detected position of the map tacks using our segmentation algorithm.

been shown to improve segmentation of images over the RGB color space [62]. To cluster similar colors, the pyramid mean shift algorithm is used to cluster similar colors [63]. The result of these two operations is shown in Figure D.5a. Once the image has been converted and the pyramid mean shifting algorithm has been applied, a threshold is applied to the HSV values. The pixels that are within the HSV ranges of the threshold are retained. The result of this thresholding is shown in Figure D.5b. While this procedure detects the tape along the rod, these threshold values do not detect the yellow markers. We create a mask that consists of the region around the rod and only search this region for yellow markers. OpenCV's erosion function is applied to the thresholded image to increase the region that the mask accepts. The result of this operation is shown in Figure D.5c. We then apply this mask to the HSV image and only retain pixels from the original HSV image (Figure D.5a) that are within the masked region. The result of this operation is shown in Figure D.5d. We then apply the threshold algorithm gain to this masked region to identify the yellow map tacks. The result of this operation is shown in Figure D.5e where the pixels that fall within the ranges of the "yellow filter" are highlighted in green. After these pixels are detected, OpenCV's FindContours function is used to cluster the detected pixels and identify contours that surround them. This function also returns



(a)

(b)

(c)

(d)

(e)

(f)

Figure D.5: These images show the stages of the algorithm used for performing segmentation to detect the markers placed along the metal strip. This procedure first limits the search to regions near the red tape along the edge of the metal strip and then identifies contours of the yellow map tacks.

a centroid for each of the contours detected. The result of this operation is shown in Figure D.5f.

Once we have performed segmentation algorithms and determined the pixel coordinates of each marker in an image, we can determine its position in the world frame. Given a rotation matrix where

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} \tag{D.1}$$

and a translation matrix where

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3, \end{bmatrix}, \tag{D.2}$$

the equation from a point in the world frame (x,y,z) to a pixel coordinate (u,v) is

$$\begin{bmatrix} us \\ vs \\ s \end{bmatrix} = \begin{bmatrix} r_{1,1}x & r_{1,2}y & r_{1,3}z \\ r_{2,1}x & r_{2,2}y & r_{2,3}z \\ r_{3,1}x & r_{3,2}y & r_{3,3}z \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \tag{D.3}$$

where s is the distance from the camera to the image plane. Applying Gaussian elimination to reduce the rows and remove the variable s produces the two following equations:

$$u(r_{3,1}x + r_{3,2}y + r_{3,3}z + r_{3,4}) = r_{1,1}x + r_{1,2}y + r_{1,3}z + r_{1,4} \tag{D.4}$$

$$v(r_{3,1}x + r_{3,2}y + r_{3,3}z + r_{3,4}) = r_{2,1}x + r_{2,2}y + r_{2,3}z + r_{2,4}. \tag{D.5}$$

These two equations can be manipulated to yield the matrix equations

$$\begin{bmatrix} ur_{3,1} - r_{1,1} & ur_{3,2} - r_{1,2} \\ vr_{3,1} - r_{2,1} & vr_{3,2} - r_{2,2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_{1,3}z - ur_{3,3}z + r_{1,4} - ur_{3,4} \\ r_{2,3}z - vr_{3,3}z + r_{2,4} - vr_{3,4} \end{bmatrix}. \tag{D.6}$$

Solving this equation for x and y produces the position of the marker in the world frame.

D.3 Parameters Used for Performing Image Segmentation

We configure all of the camera settings to prevent automatic adjustments from dynamically modifying these settings for each image. Table D.1 shows the values for the camera settings used for the experiments we conducted. Tables D.2 and D.3 show the settings to detect the red tape and the yellow map tacks that are attached to the metal strip.

Table D.1: Camera settings for the Flea2 used when conducting hardware experiments. The camera was configured manually to ensure that automatic adjustments did not alter these parameters.

FPS	15
Brightness	200
Auto Exposure	400
Sharpness	2000
White Balance (Blue/U)	850
White Balance (Red/V)	450
Hue	Off
Saturation	2800
Gamma	2400
Shutter	700
Gain	20
Trigger Delay	0
Frame Rate	480

Table D.2: Minimum and maximum thresholds applied in the HSV color space to detect the red tape that is attached to the metal strip.

Parameter	Min	Max
H	0	15
S	65	120
V	170	255

Table D.3: Minimum and maximum thresholds applied in the HSV color space to detect the yellow map tacks that are attached to the metal strip.

Parameter	Min	Max
H	15	35
S	125	165
V	210	255

REFERENCES

[1] X. Jiang, K.-M. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," *Advanced Robotics*, vol. 25, no. 3-4, pp. 473–489, 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1163/016918610X551782

[2] S. Javdani, S. Tandon, J. Tang, J. F. O'Brien, and P. Abbeel, "Modeling and perception of deformable one-dimensional objects," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA) 2011*, May 2011, pp. 1–8. [Online]. Available: http://graphics.berkeley.edu/papers/Javdani-MPD-2011-05/

[3] M. Peshkin and A. Sanderson, "Minimization of energy in quasi-static manipulation," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 53–60, Feb 1989.

[4] F. Lamiraux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 188–208, 2001. [Online]. Available: http://ijr.sagepub.com/content/20/3/188.abstract

[5] M. Moll and L. E. Kavraki, "Path planning for deformable linear objects," *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, vol. 22, no. 4, pp. 625–636, Aug. 2006.

[6] T. Hermansson, J. S. Carlson, S. Bjrkenstam, and R. Sderberg, "Geometric variation simulation and robust design for flexible cables and hoses," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 227, no. 5, pp. 681–689, 2013. [Online]. Available: http://pib.sagepub.com/content/227/5/681.abstract

[7] T. Hermansson, R. Bohlin, J. S. Carlson, and R. Sderberg, "Automatic assembly path planning for wiring harness installations," *Journal of Manufacturing Systems*, vol. 32, no. 3, pp. 417 – 422, 2013, assembly Technologies and Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0278612513000393

[8] M. Saha and P. Isto, "Manipulation planning for deformable linear objects," *Robotics, IEEE Transactions on*, vol. 23, no. 6, pp. 1141–1150, Dec 2007.

[9] F. Wang, E. Burdet, R. Vuillemin, and H. Bleuler, "Knot-tying with visual and force feedback for vr laparoscopic training," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the.* IEEE, 2006, pp. 5778–5781.

[10] A. Yamada, M. Watari, H. Mochiyama, and H. Fujimoto, "A jumping robot based on the closed elastica," in *Micro-NanoMechatronics and Human Science, 2007. MHS '07. International Symposium on*, Nov. 2007, pp. 604 –609.

[11] S. Hirose and H. Yamada, "Snake-like robots [tutorial]," *Robotics Automation Magazine, IEEE*, vol. 16, no. 1, pp. 88–98, March 2009.

[12] A. Yamada, M. Watari, H. Mochiyama, and H. Fujimoto, "A robotic catapult based on the closed elastica with a high stiffness endpoint and its application to swimming tasks," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 1477 –1482.

[13] W. Tang, P. Lagadec, D. Gould, T. Wan, J. Zhai, and T. How, "A realistic elastic rod model for real-time simulation of minimally invasive vascular interventions," *The Visual Computer*, vol. 26, no. 9, pp. 1157–1165, 2010. [Online]. Available: http://dx.doi.org/10.1007/s00371-010-0442-1

[14] V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg, "Screw-based motion planning for bevel-tip flexible needles in 3d environments with obstacles," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 2483 –2488.

[15] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, and K. Goldberg, "3d motion planning algorithms for steerable needles using inverse kinematics," *Algorithmic Foundation of Robotics VIII*, pp. 535–549, 2009.

[16] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006. [Online]. Available: http://ijr.sagepub.com/content/25/5-6/509.abstract

[17] D. Minhas, J. Engh, M. Fenske, and C. Riviere, "Modeling of needle steering via duty-cycled spinning," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 2756 –2759.

[18] L. Lyons, R. Webster, and R. Alterovitz, “Motion planning for active cannulas,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 801 –806.

[19] L. Lyons, R. Webster, and R. Alterovitz, “Planning active cannula configurations through tubular anatomy,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 2082–2087.

[20] D. C. Rucker, R. J. Webster, G. S. Chirikjian, and N. J. Cowan, “Equilibrium conformations of concentric-tube continuum robots,” *The International Journal of Robotics Research*, vol. 29, no. 10, pp. 1263–1280, 2010. [Online]. Available: http://ijr.sagepub.com/content/29/10/1263.abstract

[21] J. van den Berg, S. Patil, R. Alterovitz, P. Abbeel, and K. Goldberg, “LQG-based planning, sensing, and control of steerable needles,” *Algorithmic Foundations of Robotics IX*, pp. 373–389, 2011.

[22] R. Webster, A. Okamura, and N. Cowan, “Toward active cannulas: Miniature snake-like surgical robots,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 2857–2863.

[23] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, “Discrete elastic rods,” *ACM Trans. Graph.*, vol. 27, no. 3, pp. 63:1–63:12, Aug. 2008. [Online]. Available: http://doi.acm.org/10.1145/1360612.1360662

[24] M. W. Hannan and I. D. Walker, “Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots,” *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003.

[25] H. Mochiyama and H. Kobayashi, “The shape Jacobian of a manipulator with hyper degrees of freedom,” in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 1999, pp. 2837–2842.

[26] I. Gravagne, C. Rahn, and I. Walker, “Large deflection dynamics and control for planar continuum robots,” *Mechatronics, IEEE/ASME Transactions on*, vol. 8, no. 2, pp. 299–307, June 2003.

[27] B. Jones and I. Walker, “Kinematics for multisection continuum robots,” *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 43–55, Feb 2006.

[28] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. Walker, B. Jones, M. Pritts, D. Dienno, M. Grissom, and C. Rahn, “Field trials and testing of the octarm continuum manipulator,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2336–2341.

[29] H. Tanner, “Mobile manipulation of flexible objects under deformation constraints,” *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 179–184, Feb 2006.

[30] H. Mochiyama, M. Watari, and H. Fujimoto, “A robotic catapult based on the closed elastica and its application to robotic tasks,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Nov. 2007, pp. 1508 –1513.

[31] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986. [Online]. Available: http://ijr.sagepub.com/content/5/1/90.abstract

[32] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino, “Collision detection for deformable objects,” *Computer Graphics Forum*, vol. 24, no. 1, pp. 61–81, 2005. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2005.00829.x

[33] N. K. Govindaraju, D. Knott, N. Jain, I. Kabul, R. Tamstorf, R. Gayle, M. C. Lin, and D. Manocha, “Interactive collision detection between deformable models using chromatic decomposition,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 991–999, July 2005. [Online]. Available: http://doi.acm.org.proxy2.library.illinois.edu/10.1145/1073204.1073301

[34] A. Nguyen, “Implicit bounding volumes and bounding volume hierarchies,” Ph.D. dissertation, Stanford University, 2006.

[35] N. Shapri, A. Bade, and D. Daman, “Hierarchy techniques in self-collision detection for cloth simulation,” in *Machine Vision, 2009. ICMV ’09. Second International Conference on*, dec. 2009, pp. 325 –329.

[36] T. Yoshikawa, “Analysis and control of robot manipulators with redundancy,” in *Robotics Research: The First International Symposium.* MIT press Cambridge, MA, 1984, pp. 735–747.

[37] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, 1996.

[38] M. Branicky, S. LaValle, K. Olson, and L. Yang, “Quasi-randomized path planning,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, pp. 1481–1487.

[39] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. [Online]. Available: http://ijr.sagepub.com/content/30/7/846.abstract

[40] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[41] A. Borum, D. Matthews, and T. Bretl, “State estimation and tracking of deforming planar elastic rods,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, June 2014.

[42] S. Hirose, S. Inoue, and K. Yoneda, “The whisker sensor and the transmission of multiple sensor signals,” *Advanced Robotics*, vol. 4, no. 2, pp. 105–117, 1989. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1163/156855390X00099

[43] T. Tsujimura and T. Yabuta, “Object detection by tactile sensing method employing force/torque information,” *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 4, pp. 444–450, Aug.

[44] G. Scholz and C. Rahn, “Profile sensing with an actuated whisker,” *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 1, pp. 124–127, Feb.

[45] A. Okamura, C. Simone, and M. O’Leary, “Force modeling for needle insertion into soft tissue,” *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 10, pp. 1707 –1716, Oct. 2004.

[46] D. Matthews and T. Bretl, “Experiments in quasi-static manipulation of a planar elastic rod,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 5420–5427.

[47] T. Bretl and Z. McCarthy, “Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 48–68, 2014. [Online]. Available: http://ijr.sagepub.com/content/33/1/48.abstract

[48] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC press, 1994.

[49] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry: a Basic Exposition of Classical Mechanical Systems.* Springer, 1999, vol. 17.

[50] A. A. Agrachev and Y. Sachkov, *Control Theory from the Geometric Viewpoint.* Springer, 2004, vol. 2.

[51] J. Biggs, W. Holderbaum, and V. Jurdjevic, "Singularities of optimal control problems on some 6-d Lie groups," *Automatic Control, IEEE Transactions on*, vol. 52, no. 6, pp. 1027–1038, June 2007.

[52] G. Walsh, R. Montgomery, and S. Sastry, "Optimal path planning on matrix Lie groups," in *Decision and Control, 1994, Proceedings of the 33rd IEEE Conference on*, vol. 2, Dec 1994, pp. 1258–1263 vol.2.

[53] J. M. Lee, *Introduction to smooth manifolds.* volume 218 of Graduate Texts in Mathematics, SpringerVerlag, New York, 2003.

[54] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, and M. Gross, "Optimized spatial hashing for collision detection of deformable objects," in *Proceedings of Vision, Modeling, Visualization VMV03*, 2003, pp. 47–54.

[55] G. Sánchez and J.-C. Latombe, "On delaying collision checking in PRM planning: Application to multi-robot coordination," *The International Journal of Robotics Research*, vol. 21, no. 1, pp. 5–26, 2002.

[56] T. Bretl and Z. McCarthy, "Equilibrium configurations of a Kirchhoff elastic rod under quasi-static manipulation," in *Algorithmic Foundations of Robotics X*, ser. Springer Tracts in Advanced Robotics, E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds. Springer Berlin Heidelberg, 2013, vol. 86, pp. 71–87.

[57] J.-C. Latombe, *Robot Motion Planning.* Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[58] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents series).* The MIT Press, June 2005. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262033275

[59] S. M. LaValle, *Planning Algorithms.* Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[60] S. Lindemann and S. LaValle, "Current issues in sampling-based motion planning," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, P. Dario and R. Chatila, Eds. Springer Berlin Heidelberg, 2005, vol. 15, pp. 36–54.

[61] C. Goerzen, Z. Kong, and B. Mettler, “A survey of motion planning algorithms from the perspective of autonomous UAV guidance,” *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 65–100, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10846-009-9383-1

[62] S. Sural, G. Qian, and S. Pramanik, “Segmentation and histogram generation using the HSV color space for image retrieval,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 2. IEEE, 2002, pp. II–589.

[63] L. An, X. Xue-mei, G. Qiaoyun, and M. Qin, “Robust segmentation using marked regular pyramid and mean shift,” in *Intelligence Information Processing and Trusted Computing (IPTC), 2010 International Symposium on*, Oct 2010, pp. 341–344.