Design, Control, and Motion Planning of Cable-Driven Flexible Tensegrity Robots

by

Angelo Brian Micubo Cera

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alice M. Agogino, Chair
Professor Francesco Borrelli
Professor Mark Mueller
Dr. Adrian Agogino, NASA

Summer 2020

Design, Control, and Motion Planning of Cable-Driven Flexible Tensegrity Robots

Abstract

Design, Control, and Motion Planning of Cable-Driven Flexible Tensegrity Robots

by

Angelo Brian Micubo Cera

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor Alice M. Agogino, Chair

Tensegrity structures are an emergent type of soft-robotics that are compliant, lightweight, and impact-resilient. In collaboration with NASA Ames Research Center, research in the Berkeley Emergent Space Tensegrities Lab at UC Berkeley has largely focused on the design and control of these novel structures as potential surface exploration robots which could act as both landers and rovers. More recently, tensegrity robots have also been proposed for applications closer to home – working as disaster response and emergency co-robots to help first responders obtain situational awareness faster and safer. Constructed using isolated rigid bodies suspended in a tension network of elastic elements, tensegrity structures exhibit unique and advantageous mechanical properties for applications in uncertain and potentially hazardous environments, albeit at the cost of increased complexity for dynamic feedback control.

In addressing these challenges, this work explores possible approaches for feedback control and state estimation for ground-based rolling locomotion with six-bar spherical tensegrities. In this dissertation, we explore problems pertaining to practical implementation – state estimation, modeling, motion planning, and optimal control of tensegrity robots under uncertainty. Leveraging the well-structured dynamics of Class-1 tensegrity robots, we implement and evaluate model-based Model Predictive Control and iterative local quadratic methods for tensegrity motion planning. Additionally, we consider alternative tensegrity topologies and actuator schema which may enable improved performance for task-specific objectives. Due to the many degrees of freedom and compliant nature of tensegrity structures, however, excessive state estimate errors may propagate catastrophically. To evaluate these effects, Bayesian state estimators are applied to tensegrity ground mobility in simulation, evaluating their performance under the additional constraints of low-cost sensors and potentially scarce and noisy sensor data. An imitation learning approach is introduced to achieve directed rolling motion using a contextual neural network policy, combining deep learning and optimal control for real-time feedback control of highly nonlinear tensegrity systems. Finally, a robust minimax control approach is proposed in order to address challenges which arise at the intersection and interaction of state estimation and trajectory optimization for flexible tensegrity robotics. Combined, these pragmatic research developments help advance the progression of this novel technology towards becoming a viable and more widely adopted robotics paradigm.

To my mother, Elsie Micubo Cera

Your unwavering support, abundant love, and endless encouragement has sustained me throughout my academic journey. As a strong, independent mother who worked tirelessly and sacrificed plenty to provide the opportunities that made this dissertation possible, you inspire me to find joy in the little things and to always reach for the stars.

Contents

List of Figures

List of Tables

Acknowledgments

This dissertation and doctoral research would not have been possible without the guidance and support of so many around me. To them, I offer my heartfelt gratitude for providing such a rich and fulfilling graduate experience.

First and foremost, I'd like to thank my advisor, Professor Alice M. Agogino. Without her, my doctoral journey at UC Berkeley would have never taken root. Her enthusiasm to provide the intellectual freedom to explore my passions and interests has been truly invaluable throughout my doctoral studies. Always available, she helped guide my curiosity and creativity towards deeper insights in my own doctoral research, and her emphasis on people, not just technology, inspires me to always think on a broader scale. Her wisdom and perspective in all matters have ingrained in me a deep appreciation for lessons that transcend technical concepts taught in the classroom; I owe a great deal of personal growth and my holistic graduate education to the opportunities, challenges, and rewarding lessons granted to me through Alice's overwhelming support.

I would also like to thank the rest of my dissertation committee: Professor Francesco Borrelli and Professor Mark Mueller, for their guidance and feedback in writing this dissertation and for inspiring my research through lessons learned throughout their courses. I also thank Dr. Adrian Agogino, whose work at NASA motivated much of the original research at UC Berkeley as well as several key results presented in this dissertation.

To my colleagues and fellow labmates at the Berkeley Emergent Space Tensegrities Laboratory, I'm grateful for all of the moments of laughter and happiness that have been interspersed throughout the years. Though many folks have come and gone, I'd like to especially acknowledge the other graduate students in the lab for all of their support: Lee-Huang Chen, Kyunam Kim, Andrew Sabelhaus, Mallory Daly, Edward Zhu, Jianlan Luo, Andrew Barkan, and Alan Zhang, and all of the current and former undergraduate and master's students who have been a part of the B.E.S.T. Lab family.

I'd also like to thank the team at Squishy Robotics, especially Alice M. Agogino, Lee-Huang Chen, Deniz Dogruer, Douglas Hutchings, Anthony Thompson, and Katie Calvert. Co-founding the startup and working alongside you on the vision and mission of using technology to help firefighters and first responders has been a truly remarkable opportunity. Travelling with you across the country to meet first responders, venture capitalists, and technology enthusiasts has given me insights and perspectives I would have sorely missed, given any other graduate study experience.

Finally, thank you to all my friends and family who have supported me throughout the years, especially to my mother and brother. Your support, sacrifices, and encouragement have instilled in me a love of learning and a drive to succeed. This dissertation and the accomplishments it represents are a testament to hard work and big dreams.

Chapter 1

Introduction

This dissertation studies design, control, and motion planning of novel robotic systems based on *tensegrity* concepts. Tensile-integrity tensegrity robots are an emergent technology of lightweight, shape-shifting flexible robotics with cable-driven designs that are bio-inspired and mechanically efficient. Tensegrity designs are characterized by rigid bodies that are suspended in a balanced tension network of elastic elements. Consequently, they feature passive flexible compliance that enables these robots to detect and absorb large external forces, and their deformable bodies help naturally mitigate damage resulting from impact forces from their surrounding environment. The rigid rods and elastic cables which form these structures experience only compressive or tensile axial forces while in equilibrium, and because there are little to no bending moments or stress concentrators, tensegrity systems are inherently resistant to failure [87]. Additionally, these structures' natural compliance facilitates their potent ability to quickly distribute external forces throughout their entire tension network. This key mechanical property provides intrinsic shock protection from impact and makes the flexible structure a robust robotic platform for manipulation and mobility. Thus, tensegrity robots are mechanically robust and promising candidates for exploratory tasks in uncertain environments.

Through joint research efforts from NASA Ames Intelligent Robotics Group (IRG) and the Berkeley Emergent Space Tensegrities research lab, we have demonstrated that tensegrity robots are an elegant form-driven solution to critical problems that can arise during operation in rugged, uncertain environments. Namely, they possess valuable properties such as impact-resilience, passive compliance, and mechanical robustness, which are useful traits for complex co-robotic applications. These structural properties of compliant and lightweight tensegrities have proven to be advantageous in applications that involve high-impact loads and co-robotic cooperation with humans.

Motion planning and optimal control for rolling mobility has been a major driving force for tensegrity robotics research in recent years. Tensegrity robotics' high-dimensional highly-coupled nonlinear dynamics in contact-rich environments present a natural challenge for conventional control and motion planning techniques. As a result, innovative approaches utilizing evolutionary algorithms, data-driven methods, and model-based optimal control have all been developed to control these novel complex robots. In particular, great emphasis has been placed on optimal performance with respect to rolling speed under non-ideal conditions and rough terrain, but less consideration has been made for practical implementation challenges such as energy efficiency, controllability, and directional trajectory-tracking accuracy. Notably, the hardware performance and energy efficiencies of mobile tensegrity robots have yet to be thoroughly evaluated for practical use cases in realistic scenarios.

The Berkeley Emergent Space Tensegrities[1] (B.E.S.T.) Lab, in collaboration with the Intelligent Robotics Group at NASA Ames Research Center, had explored the use of tensegrity robotic systems for the purpose of space exploration. Due to their unique impact-resilient properties and lightweight designs, tensegrity structures have excellent potential as easily scalable, highly mobile, low-cost landers and surface exploration rovers. More recently, the B.E.S.T. Lab and spin-off startup *Squishy Robotics, Inc.* has been evaluating their effectiveness in earthly applications such as large-scale disaster response, emergency response, and rapid search-and-rescue. In order for these robots to be applicable in such high-impact and dynamic situations, however, their mobility and manipulation capabilities must be reliable, robust, and efficient even in uncertain conditions. This need is the primary driving motivation behind the work presented here - a summary of research towards further developing robust mobility for these unique and relatively complex tensegrity robots. In presenting an alternative to the open-loop control policies that have been used (surprisingly effectively) by the B.E.S.T. Lab in the past, this work attempts to motivate the use of optimal feedback controllers for robust dynamic control of these complex robots to enable faster and more energy-efficient mobility.

1.1 Research Objectives

At a high level, this dissertation addresses the following key research goals to further tensegrity robotics as a whole:

1. **Application of existing controllers, estimators, and motion planning approaches to novel tensegrity robotics.** Through implementing a variety of common model-based optimal control techniques, this work investigates the potential of using deterministic control approaches with nominal system dynamic models. Methods using locally-linearized dynamics as well as iterative nonlinear control techniques are compared and evaluated through key metrics for tensegrity mobility such as speed, trajectory tracking, and energy efficiency. Notably, tensegrity robots present a greater challenge for actuator and sensor selection than conventional robotics due to having significantly more degrees of freedom, natural structural compliance, and a greater number of moving parts and actuators. In this work, different actuation and sensor configurations are evaluated to motivate reliable and robust tensegrity hardware designs.

[1] Website: http://best.berkeley.edu/

2. **Robust Nonlinear Control.** Due to their intrinsic compliance and flexible design, reliable performance under uncertainty is a major concern for tensegrity robotics. The exact state or pose of the robot is often unattainable - sensor fusion and state observers can only provide probabilistic beliefs of the current state of the robot at any given time. To this end, this work explores applications of nominal controllers for uncertain systems using certainty-equivalent control assumptions and investigates other robust control approaches such as minimax dynamic programming.

3. **Data Driven Motion Planning and Control.** Data-driven approaches such as imitation learning are explored through practical applications with tensegrity robots. Namely, tensegrity systems are utilized as a useful platform for investigating hybrid approaches combining controls and machine learning for control and motion planning with high-dimensional nonlinear dynamic systems. In this work, we investigate how imitation learning with expert optimal control policies can be utilized for real-time feedback control.

1.2 Outline of Dissertation

This dissertation is structured as follows:

- **Chapter 2** provides background and motivation on tensegrity systems and discusses their advantageous mechanical properties as well as standing challenges. Additionally, this discussion helps elucidate why dynamic control of these structures can be difficult, relative to traditional rigid robots.
- **Chapter 3** presents some tensegrity dynamics representations for the six-bar spherical tensegrity which helps set the context for the numerical optimization approaches we discuss later in the work. The well-structured dynamics that are outlined in this chapter provide the mathematical backbone upon which constrained optimization problems can be rapidly formulated and applied to various different tensegrity topologies.
- Next, **Chapter 4** demonstrates two approaches for generating dynamic rolling locomotion for ground mobility using Model Predictive Control and iterative local quadratic methods.
- **Chapter 5** discusses the dual problem of state estimation, and unique challenges to tensegrity robotics which must contend with contact-rich ground mobility.
- **Chapter 6** combines and extends the work in Chapters 4 and 5 to address practical considerations of computational time complexity and uncertainties which arise from imperfect models and noisy sensor measurements.
- Finally, **Chapter 7** concludes by summarizing the work throughout the dissertation. We discuss potential future work in further developing and refining these approaches to new and uncertain environments, delayed-reward learning tasks, and full tensegrity autonomy.

Chapter 2

Background and Motivation

2.1 Tensegrities - A Tale of Tensional Integrity

> *"All the categories of creatures act individually as special-case and may be linearly analyzed; retrospectively, it is discoverable that inadvertently they are all interaffecting one another synergetically as a spherical, interprecessionally regenerative, tensegrity spherical integrity..." -R. Buckminister Fuller*

Tensegrity structures never fail to capture the imagination. Whether you're seeing them for the first time or you're putting together your hundredth tensegrity model, these unique structures always have a fascinating allure which sparks the creativity and wonder within. Time and time again through outreach demos, conferences, and technology expos, I've seen firsthand the sense of wonder as children and adult alike hold these odd and interesting robots in their hands for the first time. Often, they'll relate it to reminiscent egg-drop contests from high school or suggest some interesting new geometry or topology that no one had ever considered. They'll promote a seemingly endless list of new technical features or potential applications in their field, and each time I'm reminded of why tensegrity-related research called to me in the first place – a boundless sea of possibilities for such a unique, unproven robotic technology.

The tensegrity concept's origin is more humble than futuristic robots, however. Whereas today some research groups focus on enabling dynamic movement of tensegrity robots, tensegrity structures were originally rooted in the flowing flexibility of static artistic architecture. Buckminister Fuller – an eccentric 20^{th} century inventor, quack, and/or visionary (depending on who you ask) best known for his invention of geodesic domes[1] - coined the term 'tensegrity' to describe "*self-tensioning structures composed of rigid structures and cables, with forces of traction and compression, which form an integrated whole.*" For him and

[1] This is an often contentious point of debate, with some evidence of earlier European architecture dating back 40 years prior.

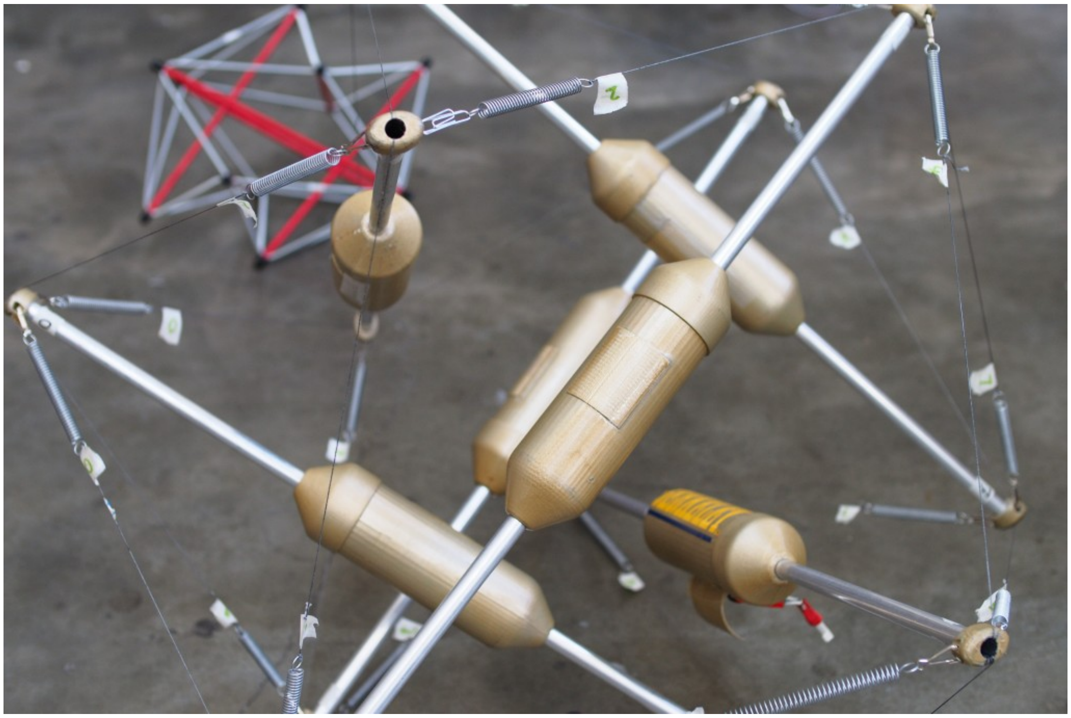
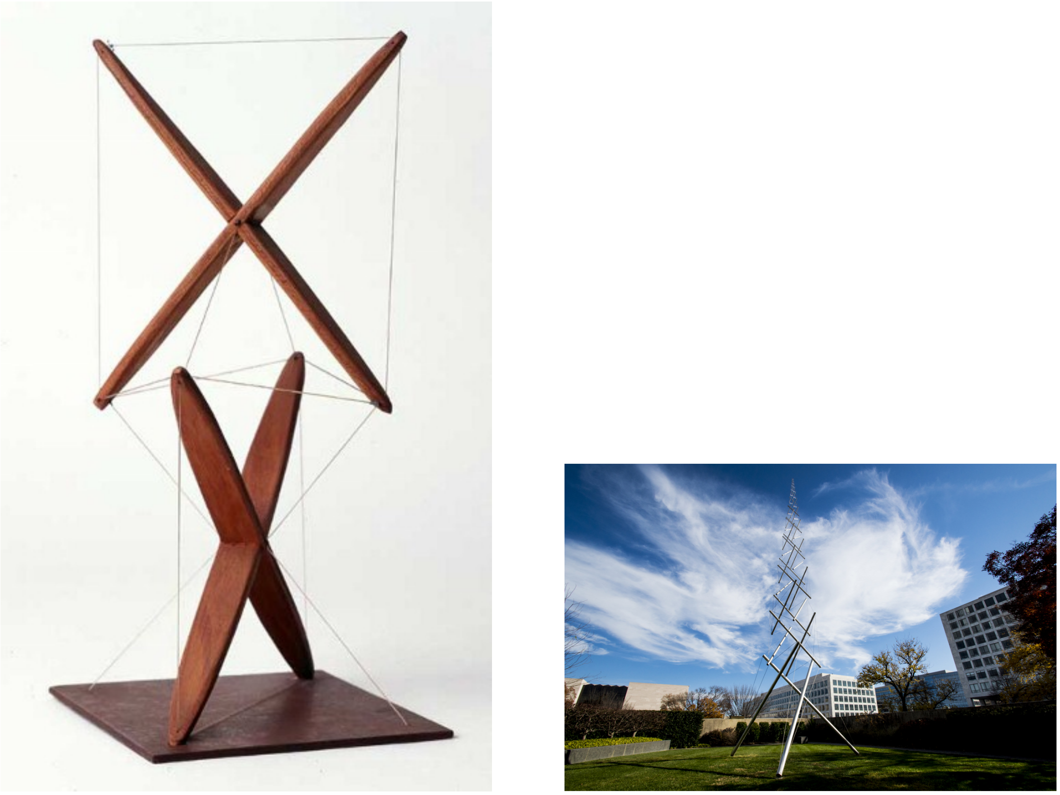


Figure 2.1: Early prototypes of spherical tensegrity robots, courtesy of the Berkeley Emergent Space Tensegrities Research Lab.

many others, tensegrities represented a harmonious balance or equilibrium that existed not only in structures, but in organizations, systems, and living beings as well.

The often-debated 'original tensegrity' structure is usually cited as the *X-Piece*, which was created by artist Kenneth Snelson in 1948 at Black Mountain College where Buckminister Fuller was lecturing at the time. The structure, shown in Figure 2.2, was modular and demonstrated the versatility of tensegrities by allowing for combining the three-bar tensegrity design into long tensegrity 'chains', such as in Snelson's iconic 18-meter tall *Needle Tower* of 1968. In 1949, one year after the first tensegrity piece was created by Snelson, he introduced the now canonical six-bar or six-rod icosahedral tensegrity (also known colloquially as the *six-bar spherical tensegrity*). This structure, example shown in Figure 2.1, is the most ubiquitous form of tensegrity robots, in part thanks to the popularity of NASA Ames' research as well as original children's toys which were pervasive in the 1980's (see 'Skwish' toys). This canonical spherical six-bar tensegrity robot topology, which enables rolling mobility, is the primary focus of this work.



(a) The *X-Piece* tensegrity structure created by Kenneth Snelson in 1948.

(b) The *Needle Tower* located outside of the Hirshhorn Museum and Sculpture Garden in Washington, D.C.

Figure 2.2: Early examples of tensegrity structures.

2.2 Tensegrity Applications

Prior to robotics, tensegrity structures have been recognized as exceptionally strong and lightweight mechanical designs, making structural bridges and domes a natural application [34]. Similarly, they've been used towards unique lightweight structures such as deployable masts [30], antennas [51], and even space structures [99].

Although a more nascent technology compared to traditional applications in structural architecture, tensegrity robotics have also already seen some experimental applications to date. In most cases, tensegrity robots excel in situations which must deal with unique design constraints, uncertainty, and/or close proximity with humans. In these situations, tensegrity robots' natural compliance ends up being a significant advantage for some interesting and impactful applications.

One interesting example - and the initial push for research in the lab - is NASA's interest in researching and developing a tensegrity platform for an innovative surface exploration rover as part of their *NASA Innovative Advanced Concepts Program* [2, 1, 93]. Initiated in 2012, the project aims to address the challenges of increasingly important small, lightweight, and low-cost missions. In their vision, compliant tensegrity robots may be compactly squeezed into a small launching platform and spring away upon landing without any need for extensive entry, descent, and landing technology. Their tensegrity form enables these robots to reorient themselves and survive significant falls without the need for additional specialized hardware. Additionally, characteristics of reusability, redundancy, and reliability make tensegrity platforms a compelling option. Their modular design allows for assemblies to be reused for many different applications, while their light weight and low cost mean that multiple robots can be assigned to collaboratively achieve a given task.

Other applications utilize tensegrity for biomechanical or co-robotics applications in close proximity to people. As an example, [4] from UC Santa Cruz presents an exploration into biomimetic tensegrity robotic manipulators inspired by the human shoulder. Tensegrities naturally pose as an analog to biological musculoskeletal systems of vertebrates, with rigid bodies and tension elements being analogous to the structural skeletal scaffolding and tendons/muscles in the system.

A research group out of UC San Diego utilizes tensegrities in [28, 27] to traverse complex duct systems, an application which requires small, agile robots and complex control with a wide variety of actions and large uncertain workspace. In this work, a simple design using two tetrahedral tensegrity frames is manipulated using a linear actuator and actuated cables. Periodic controllers control an alternating motion which allows the robot to make incremental progress in a inchworm-like crawling motion.

Finally, other research groups have also developed spherical tensegrity robots to explore their potential for ground mobility. Research groups in Japan first demonstrated tensegrities' capability for ground locomotion through 'crawling' in [84, 83]. Through deformation of the structure, gravitational potential energy is used to force the robot to transition from a stable neutral state to an unstable state, resulting in a punctuated rolling locomotion.

2.3 Motivations for the Tensegrity Robot Paradigm

Tensegrity robots are often unfairly characterized as a flexible robot paradigm that is needlessly complex or unnecessarily redundant, with potentially more actuators, sensors, and on-board computation than conventional robots. Admittedly, while there are kernels of truth to this sentiment (largely due to it being a new underdeveloped technology), there are strong arguments for the enticing advantages that novel tensegrity robot designs can provide.

Passive Compliance. Due to their flexible robot design, tensegrities exhibit structural compliance in the presence of external disturbances. Additionally, when combined with intrinsically applied internal forces, external disturbance forces from the environment can potentially enable a larger configuration space (i.e., the space of possible positions the robot may attain) than the robot acting solely on its own. Consider a flexible tensegrity manipulator working cooperatively with a human operator in order to move an object. Because of the robot's natural compliance, simpler impedance-based controllers and fewer cheaper sensors (e.g., IMUs rather than vision/LIDAR) can be sufficient when using a soft, flexible robot design. Furthermore, when the robot is finished with its operational tasks, the robot can be compactly packed away through externally applied forces from the operator.

Variable Stiffness. Tensegrity robots also have the unique ability to control their overall effective structural stiffness. This variable stiffness is an advantage that enables tensegrities to fully capture the wide range of compliance inherent to biological systems. In contrast to traditional robotic systems which utilize rigid and precise mechanical linkages that need to be compensated with complex software control, tensegrities have the ability to mechanically represent several orders of magnitude of stiffness, dynamically responding to changing design requirements and the environment.

Intrinsic safety. Tensegrity structures feature low mass, high strength-to-weight ratio, and passive compliance. Combined, these intrinsic characteristics of tensegrities mean that they naturally mitigate any potential damage to their surroundings. While tensegrity robots have the capability to increase their stiffness for more demanding tasks with greater loads and higher precision, their passive compliance can further provide system-level information for rich high-fidelity impedance control. While unintentional physical interactions may occur during complicated dexterous tasks, their low mass and inertia help mitigate any unforeseen risks from the robot.

Redundant sensors/actuators. Redundant sensors and actuators that are common to tensegrities can be viewed negatively, necessitating higher costs and complexity. In some cases, however, such as working in adversarial environments, this redundancy can enable the robot to continue its tasks under sub-optimal operating conditions. Rolling mobility for the six-bar spherical tensegrity, for example, can continue even in the presence of broken actuators. The degree of degradation in performance is dependent on the design and which specific actuators are non-functioning, but a sub-optimal robot is in many practical cases infinitely better than an entirely obsolete robot (e.g., remote space exploration rovers millions of miles from earth).

Modular design. Tensegrities may seem intricately complex at first glance – they have dozens of passive and active cables interconnecting floating rigid bodies in a seemingly random manner. In reality, however, they feature only a few repeated modules of individual parts representing two key components - actuated rods and cables. This modularized design lends itself to scalable manufacturing, rapid repairs, and plug-and-play substitution of system components. Unfortunately, tensegrity hardware is still nascent, and challenges inherent to cable-driven robotics still remain a significant challenge. Despite this, research and development along useability and reliability of tensegrity hardware may provide increasing returns, as the individual modular components themselves can be minimally modified for use in a huge variety of tensegrity topologies.

2.4 Challenges for Tensegrity Robotics

Despite their compelling and unique advantages, tensegrity robots must address many real challenges before they may be considered a mature technology viable for widespread use. As intrinsically flexible robots, tensegrities face numerous technical obstacles regarding dynamic modelling, distributed control/sensing, and control of compliant systems in the face of uncertainty. In particular, dynamic modeling is a standing challenge for all flexible robotics, with various approaches to tackling the issue including lumped-parameter models [110][92], finite element methods [72][6], and assumed modes models [96]. Notably, tensegrity robots naturally lend themselves towards lumped-parameter models due to their natural discretization between rigid rods and compliant cable elements. A few lumped-mass dynamics models are presented in Chapter 3.

More interestingly, tensegrity robots must also deal with the challenges of controlling high-dimensional, coupled nonlinear dynamics with uncertainty, all of which result from their soft-bodied designs. In the past, many previous works leveraged heuristic based approaches such as Monte Carlo simulation, evolutionary algorithms, biologically-inspired central pattern generators, and/or relied on quasi-static assumptions. Currently, there exists very little to no existing work on controlling dynamic tensegrities under uncertainty, though some examples of robust vibration control of *static* tensegrity structures are available [57, 22], as a result of tensegrity's historic roots in structural architecture. This work strives to elucidate and address many of the practical challenges surrounding optimal control of dynamic mobile tensegrity robots.

Chapter 3

Tensegrity Dynamics and Simulation

3.1 Introduction

This dissertation primarily investigates the canonical six-bar spherical topology; in this chapter we explore in greater detail the dynamics model representations of this tensegrity topology which are useful for controls and state estimation in later chapters. Notably, this tensegrity design is a *Class-1* tensegrity structure, meaning that in an idealized model, each node is attached to (at most) one rod and, equivalently, that the rods and cables act only as two-force members under pure compression or pure tension, respectively. In particular, we assume that the rigid bodies are rod-shaped and have negligible inertia about their longitudinal axis.

Class-k tensegrity systems are defined by the maximum number k of rigid bodies that are connected to a single node. Class-1 tensegrities, the main topology of interest in this work, are specific tensegrity structures constructed with compressive bodies that bear no rigid joints and which are interconnected solely through series-elastic tensile elements. For this reason, the highly-coupled nonlinear tensegrity dynamic equations of motion are well-structured and can be procedurally obtained when given a fixed set of model parameters. Forming the rigid-body dynamics of the tensegrity system can therefore be simply achieved in a straightforward manner once the physical parameters (i.e., rod mass, rod length, cable stiffness) and rod and cable connections are known, thus making tensegrity structures well-suited for model-based optimal control.

3.2 Prior Work

Tensegrity dynamical systems in existing literature have universally been represented as rigid body rods connected by spring-cable assemblies, often taking the assumption of discrete idealized two-force members. Nevertheless, numerous different lumped-parameter representations have been proposed over time as the field has developed. Due to the dynamic complexities that arise from such interconnected systems, it can be argued that a lack of universal, simple equations of motion have been a limiting factor for the adoption of tensegrity robotic systems [88].

Skelton et al. present some of the earliest formulations of Class-1 and Class-2 tensegrity systems in [89, 91]. This representation is summarized later in this work in Section 3.5. In the prior work, they represent tensegrity rigid body dynamics as a matrix differential equation rather than a traditional state space representation, with special emphasis on quasi-static analyses of tensegrity systems. In particular, the authors introduce the importance of a *connectivity matrix* describing cable-rod interconnections and introduce some of the earliest notions of tensegrity control problems with *string force density* as control variables.

Unfortunately, while this early formulation has the benefit of removing trigonometric nonlinearities due to the unique configuration space representation, control analyses of the system is limited, as it becomes difficult to reason about desired behavior through the unique tensegrity *configuration matrix* in comparison to a traditional Euclidean state space vector form. Nevertheless, the authors demonstrate applicability of these early representations for locally stabilizing control problems about linearized equilibrium states.

Later results in [88] by the same group represent the rigid body rods using a minimal representation state that keeps track of each rod's center of mass and orientation, specified by Euclidean coordinates and a unit vector's spherical coordinates, respectively. The details of this vectorized representation can be found in Section 3.5. More recent results in [25] introduce a numerical corrective algorithm for dynamic constraint violations in the minimal representation that arise due to machine round-off error.

While traditional tensegrity models have neglected string mass, very recent results in [33] present a new approach incorporating rigid bars with massive strings. The paper presents a matrix-second-order nonlinear differential equation that permits the distribution of cable mass among arbitrary point masses along the length of the string, while still maintaining exact rigid body dynamics for the rods. To reduce complexity of the model, the authors adopt a non-minimal representation for each rod, with each rod having six states, and cite that this non-minimal representation 'enables the model [to be] simplified and exploited during numerical computations'.

In a similar vein, [77, 76] represents rigid rods using point masses distributed along the longitudinal axis of each rod. Primarily concerned with transient behavior upon dynamic impacts, the authors adopt this representation to allow bending in each rod, which they argue is a critical component of tensegrities' unique impact resilient characteristics. In [75], Rimoli et al. present experimental results of tensegrities upon impact and argue that more efficient structural design of tensegrities could be achieved by *exploiting* rather than *avoiding* buckling behavior of compression members.

This work continues the trend in recent tensegrity research to adopt a conventional state space point mass representation that is more familiar to those coming from traditional control theory literature. The advancements presented later in this chapter are motivated by pragmatic considerations such as numerical stability, machine precision errors, and computation speed that arise in practical application of modern control techniques towards tensegrity robotic systems.

3.3 Contributions of this Chapter

This chapter outlines various lumped-mass dynamics models which are useful for controlling tensegrity robotics. The *minimal* and *point mass* representations introduced here guide us to consider tensegrities in an idealized manner, with discrete rigid bodies and elastic cables treated as simple two-force members. Furthermore, these simplifications allow us to reason about how to accurately simulate dynamic tensegrities under contact-rich rolling locomotion, as well as identify key areas where accuracy may be sacrificed for computation speed, a key trade-off for receding horizon control. Practical considerations for: 1) wide applicability to various tensegrity topologies and 2) computation speed, directly lead to the novel tensegrity point mass dynamics formulation presented at the end of this chapter which is utilized in the most recent results presented later in this work.

Furthermore, this chapter introduces the key open-source simulation framework which was utilized to obtain many of the results presented in this paper. Critically, the creation of this simulation framework vastly improved the tensegrity research workflow, which previously was hampered by numerous hardware reliability issues, as robust off-the-shelf tensegrity robots currently do not exist. The chapter concludes by demonstrating how these tools can be used to evaluate tensegrity structures for a variety of interesting use cases such as simulating high-altitude drop tests and evaluating center-payload protection through tensegrities' natural impact resilient structures.

3.4 Six-Bar Spherical Tensegrity Overview

The defining characteristic of the six-bar topology is the presence of six separate rigid rods in compression, the minimal number of rigid bodies necessary for a symmetric spherical tensegrity form and thus the simplest geometry to explore for ball-like rolling locomotion. For this specific topology, the rigid bodies are interconnected through a tension network of 24 elastic elements, such as compliant cables, with the two cable-ends connected to two independent rods.

As an active robotic system, the six-bar spherical tensegrity robot has the ability to control the rod lengths and/or the tensions of the actuated cables. By convention, as in [39], [24], [109], it is common for individual cable tensions to be controlled by using motors to spool in (retract) or extend (release) a cable's restlength. Using this principle, cable tensions can be readily manipulated, with special attention to slack cables which, by their nature, can produce no compressive force.

For the six-bar topology, the *neutral pose*, defined as the geometry when rest length is equal for each of the 24 elastic cables, has a convex hull enclosing the 12 endpoints of the rods in the shape of an icosahedron with 20 triangular-shaped faces. Through manipulation of the individual cable tensions, the robot deforms this geometry to achieve dynamic motion. This flexible shapeshifting capability is key to the design's versatility and enables the tensegrity

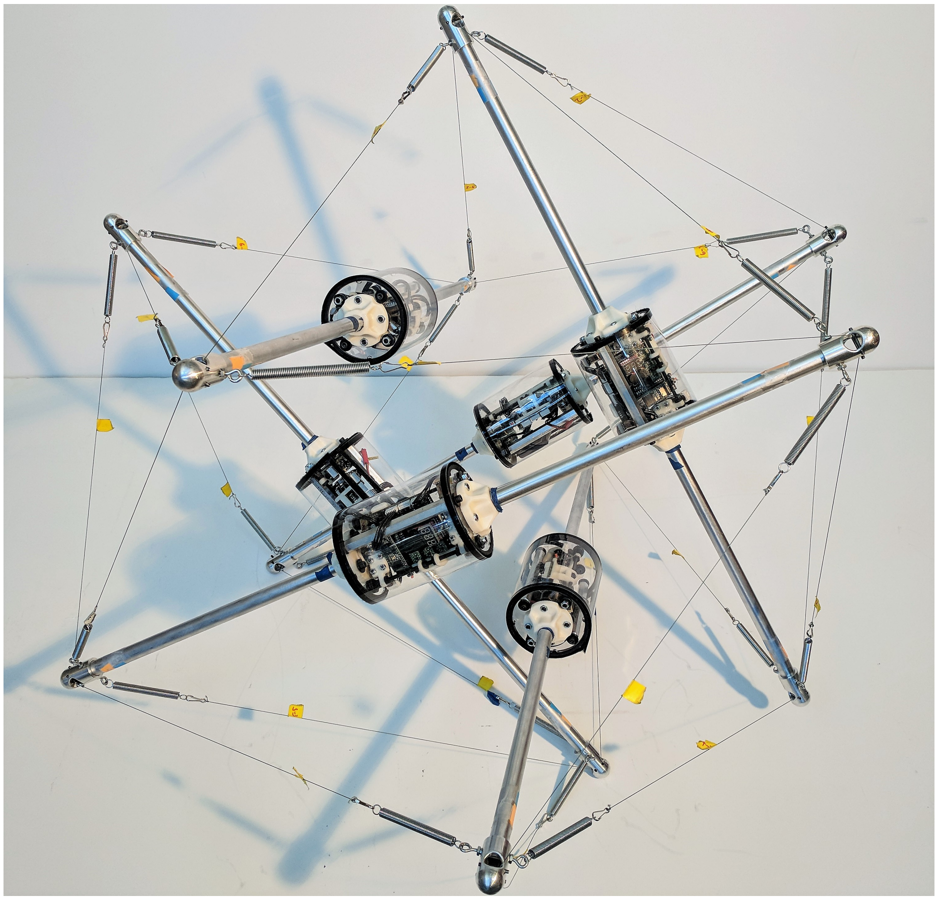


Figure 3.1: A robot hardware example of the canonical six-bar spherical tensegrity topology, TT-5 from the Berkeley Emergent Space Tensegrities research lab.

robot to perform rolling locomotion by shifting the center of mass of the robot in a desired manner, as described in [48].

3.5 Minimal Six-Bar Rigid Body Dynamics

In the following sections, we introduce the dynamics model representation used in earlier research, first presented by Wroldsen and Skelton [107], [88]. Although the previous work is included here for completeness, for this research, the dynamic formulation was ultimately abandoned in favor of the simpler but slightly less accurate point mass formulation introduced in Section 3.6. The rigid body formulation introduced here has the advantage of more accurately representing rotational inertia and being a minimal representation (i.e., having the fewest number of states necessary to represent the full dynamical system). However, as discussed towards the end of the section in Subsection 3.5.4, this minimal representation suffers from dynamic singularities and stiff ordinary differential equations which make model-

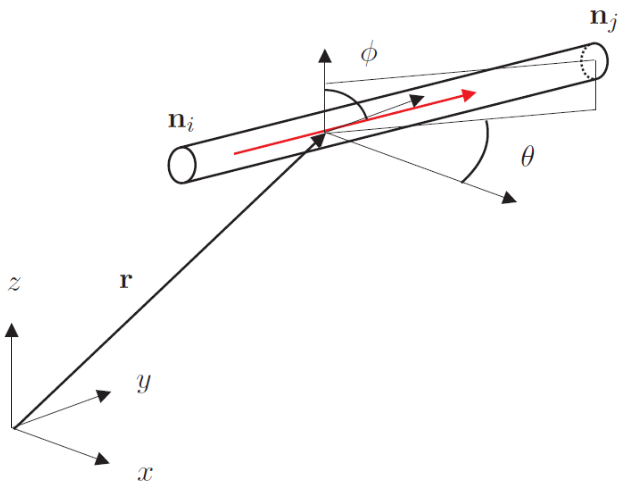


Figure 3.2: Minimal state representation of a single rod rigid body, using XYZ-coordinates of the rod's center of mass and spherical angles representing the directional unit vector that runs along the rod's axial direction. Figure by Anders Wroldsen presented in [107].

based receding horizon control difficult or impossible outside of short time-horizons or well-conditioned cases. Nevertheless, this minimal representation may be useful for static analyses or evaluating quasi-static interactions.

3.5.1 Rod and Cable Dynamics

In the minimal representation motivated by [88][107] and presented in this section, each rod is represented by five states: x, y, z position of its center of mass and the azimuth and polar angles of a unit vector which lies along the axial direction of the rod (see Fig. 3.2). Traditionally, rotation around the rod's longitudinal axis is less important and often unconstrained in the physical construction of the robot and is thus consequently neglected in this model. The equations of motion of rod i with respect to the minimal q-coordinate representation can therefore be represented as:

$$\ddot{\mathbf{q}}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) = \mathbf{M}_i^{-1}\Big(\mathbf{H}_i(\mathbf{q}_i)\big(\mathbf{g}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{f}_i\big)\Big) \tag{3.1}$$

$$\mathbf{q}_i = \begin{bmatrix} x \\ y \\ z \\ \theta \\ \phi \end{bmatrix}_i \in \mathbb{R}^5 \qquad \mathbf{g}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) = \begin{bmatrix} 0 \\ 0 \\ -mg \\ -2J\dot{\phi}\dot{\theta}sin\phi cos\phi \\ J\dot{\theta}^2 sin\phi cos\phi \end{bmatrix}_i \in \mathbb{R}^5$$

$$\mathbf{M}_i = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_i \in \mathbb{R}^{5\times 5} \qquad \mathbf{H}_i(\mathbf{q}_i) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Jsin^2\phi} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{J} \end{bmatrix}_i \in \mathbb{R}^{5\times 5}$$

where $\mathbf{M}_i$ is the mass matrix for rod i, $\mathbf{H}_i$ accounts for rotational inertia, and $\mathbf{g}_i$ incorporates gravitational effects and additional terms that arise as a consequence of the spherical coordinate representation. $\mathbf{f}_i$ represents the generalized forces which act on rod i (e.g., the cable tensions, collision forces between rods, and contact forces with the ground) with respect to the q-coordinate frame.

Individual cable tensions, t_i, can be modeled using a simple linear series-elastic representation of Hooke's Law, with the added 'slackness' constraint that tensile forces are strictly non-negative:

$$t_i = max\,\{\ 0\ ,\ k(s_i - \ell_i)\ \} \tag{3.2}$$

here t_i is the individual cable tension, s_i is the separation distance between the endpoints of the rods which cable i is connected to, and ℓ_i is the cable restlength which can be controlled (extended or retracted) using actuators. Collision forces between two rods and contact forces between rods and the ground are calculated using a similar piecewise-linear Hookean contact model in compression.

3.5.2 Complete Tensegrity Robot Dynamics

With the dynamic model for a single-rod given above, generating the complete equations of motion for the entire tensegrity structure is a straightforward procedure, given that rods are independent and interact primarily through the known cable connections of the robot. As alluded to previously, this principle means that modeling any Class-1 tensegrity topology is unambiguous once the cable and rod interconnections of the robot are properly defined.

To demonstrate this process for the six-bar spherical topology, we first designate each rod endpoint as a 'node' with index $j \in \{1, ..., 12\}$, which permits a beginning or end of a cable or rod connection to be defined. In convention with other existing simulations in [23, 49] from the Berkeley Emergent Space Tensegrities lab, the six rods are defined in this section with respect to the defined nodes as follows:

Rod	1	2	3	4	5	6
Nodes	1,2	4,3	5,6	8,7	9,12	10,11

Rod connections are represented through an analogous rod connectivity matrix, **R**:

$$\mathbf{R} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix} \in \mathbb{R}^{6\times 12} \tag{3.3}$$

A cable connectivity matrix, $\mathbf{C} \in \mathbb{R}^{24\times 12}$, is subsequently defined (an example for TT-5 robot is provided below), where each row corresponds to an individual cable and each column corresponds to one of the 12 nodes previously defined above. Cable-node connections are defined on each row by assigning a value of 1 or -1 (there must only be one of each; sign determines axial unit vector direction) for each nodal endpoint and 0 elsewhere.

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{24\times 12} \tag{3.4}$$

The full system robot dynamics can then be represented by a configuration vector by simply concatenating each of the independent rod dynamics:

$$\ddot{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{M^{-1}}(\mathbf{q})\Big(\mathbf{H}(\mathbf{q})\big(\mathbf{g}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{F}(\mathbf{q},\dot{\mathbf{q}})\big)\Big) \tag{3.5}$$

$$\mathbf{q} = \left[\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T, \mathbf{q}_4^T, \mathbf{q}_5^T, \mathbf{q}_6^T\right]^{\mathbf{T}} \in \mathbb{R}^{30}$$

$$\mathbf{g}(\mathbf{q},\dot{\mathbf{q}}) = \left[\mathbf{g}_1^T, \mathbf{g}_2^T, \mathbf{g}_3^T, \mathbf{g}_4^T, \mathbf{g}_5^T, \mathbf{g}_6^T\right]^{\mathbf{T}} \in \mathbb{R}^{30}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{M}_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{M}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{M}_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{M}_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{M}_6 \end{bmatrix} \in \mathbb{R}^{30\times 30}$$

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} \mathbf{H}_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{H}_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{H}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{H}_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{H}_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{H}_6 \end{bmatrix} \in \mathbb{R}^{30\times 30}$$

The vector $\mathbf{F}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{30}$ is the summation of cable, rod collision, and ground contact generalized forces in the q-coordinate frame. Generalized forces utilize the principle of virtual work to reframe applied forces in a new desired reference frame (e.g., transforming from Cartesian coordinates to the generalized q-coordinates used in this section). In general, when transforming from an initial n-dimensional reference frame $\boldsymbol{x}$ to a new m-dimensional reference frame $\boldsymbol{y}$, the infinitesimal work done on the dynamical system with respect to a small shift in coordinates $\partial \boldsymbol{y_i}$ is simply:

$$\partial W = \mathbf{F}_1 \cdot \sum_{i=1}^{m} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{y}_i}\partial \boldsymbol{y}_i + \mathbf{F}_2 \cdot \sum_{i=1}^{m} \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{y}_i}\partial \boldsymbol{y}_i + ... + \mathbf{F}_n \cdot \sum_{i=1}^{m} \frac{\partial \boldsymbol{x}_n}{\partial \boldsymbol{y}_i}\partial \boldsymbol{y}_i \tag{3.6}$$

$$= \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{y}_1}\partial \boldsymbol{y}_1 + \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{y}_2}\partial \boldsymbol{y}_2 + ... + \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{y}_m}\partial \boldsymbol{y}_m \tag{3.7}$$

By definition, the virtual work of the system is a summation of products of generalized forces and displacements:

$$\partial W = Q_1 \partial \boldsymbol{y}_1 + Q_2 \partial \boldsymbol{y}_i + ... + Q_m \partial \boldsymbol{y}_m \tag{3.8}$$

Thus, by identifying like terms, the generalized forces Q_i associated with the generalized coordinates $\boldsymbol{y}$ are defined as:

$$Q_i = \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x_j}}{\partial \boldsymbol{y_i}} \qquad i = \{1, ..., m\} \tag{3.9}$$

In an analogous manner, generalized cable forces and torques for the tensegerity system in the q-coordinate frame, $\mathbf{F}_{cables} \in \mathbb{R}^{30}$, can be obtained using the equation below:

$$\mathbf{F}_{cables} = -\frac{\partial \boldsymbol{\gamma}(\mathbf{q})^{\mathbf{T}}}{\partial \mathbf{q}} \sum_{i=1}^{24} (\mathbf{C}_i^T \mathbf{C}_i \otimes \mathbf{I_3}) \boldsymbol{\gamma}(\mathbf{q}) t_i = \mathbf{B}(\mathbf{q})\mathbf{T} \tag{3.10}$$

$$\boldsymbol{\gamma}(\mathbf{q}) = \begin{bmatrix} x_1 & + & \frac{L}{2} cos\theta_1 sin\phi_1 \\ y_1 & + & \frac{L}{2} sin\theta_1 sin\phi_1 \\ z_1 & + & \frac{L}{2} cos\phi_1 \\ x_1 & - & \frac{L}{2} cos\theta_1 sin\phi_1 \\ y_1 & - & \frac{L}{2} sin\theta_1 sin\phi_1 \\ z_1 & - & \frac{L}{2} cos\phi_1 \\ & \vdots & \\ x_6 & - & \frac{L}{2} cos\theta_1 sin\phi_6 \\ y_6 & - & \frac{L}{2} sin\theta_1 sin\phi_6 \\ z_6 & - & \frac{L}{2} cos\phi_6 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ X_2 \\ Y_2 \\ Z_2 \\ \vdots \\ X_{12} \\ Y_{12} \\ Z_{12} \end{bmatrix} \in \mathbb{R}^{36} \tag{3.11}$$

where $\mathbf{C_i}$ is the i^{th} row of the connectivity matrix $\mathbf{C}$, $\boldsymbol{\gamma}$ is the concatenated XYZ coordinates of all 12 nodal endpoints in the global XYZ-frame, and t_i is the individual cable tension calculated using Eq. 3.2. Thus, $\mathbf{B} \in \mathbb{R}^{30 \times 24}$ is the matrix that linearly maps the column vector $\mathbf{T} \in \mathbb{R}^{24}$ of tensile forces from each cable into the generalized q-coordinate reference frame.

The resulting complete acceleration dynamics can be compactly expressed as:

$$\ddot{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{M^{-1}} \Big(\mathbf{H}(\mathbf{q}) \big(\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{T} + \mathbf{F}_{collision} + \mathbf{F}_{contact} \big) \Big) \tag{3.12}$$

where $\mathbf{F}_{collision}$ and $\mathbf{F}_{contact}$ are the summation of rod collision forces and ground contact reaction forces, respectively, in the q-coordinate frame.

3.5.3 Ground Contact Dynamics and Rod Collision

Individual force elements are calculated at each timestep of the simulation using linear and nonlinear spring models for the ground contact and collision forces, respectively, using the equations below:

$$\mathbf{F}_{contact,i} = c \cdot \mathbf{\Delta}_{ground,i} \qquad or \qquad \mathbf{F}_{collision,j} = c \cdot (e^{\mathbf{\Delta}_{collision,j}} - 1) \tag{3.13}$$

where c is a constant scaling factor (in practice, 5×10^4 works well) and $\mathbf{\Delta}_{ground} \in \mathbb{R}^{12}$ and $\mathbf{\Delta}_{rods} \in \mathbb{R}^6$ are measures of interference, either between a nodal endpoint and the ground or between two different rods. For ground contact forces, $\mathbf{\Delta}_{ground}$ is measured by how far each node's Z-coordinate sits below the floor:

$$\mathbf{\Delta}_{ground,i} = max\{0,\ Z_{floor} - Z_{node,i}\} \tag{3.14}$$

Piecewise-affine rod collision forces are calculated in a similar manner by pairwise-checking of whether the rod axis separation distances are less than a given threshold (e.g., the radius of the rod).

3.5.4 Dynamic Singularities

The minimal representation used in the rigid body dynamics presented above lessens computational complexity by reducing the number of symbolic variables from 36 down to 30. Unfortunately, however, this model suffers from singularities at $\boldsymbol{\phi}_i = 0$ and $\boldsymbol{\phi}_i = \pi$, where the rod is nearly vertical. In these cases, azimuthal rotational velocities ($\dot{\boldsymbol{\theta}}$) and accelerations ($\ddot{\boldsymbol{\theta}}$) according to Eq. 3.5.1 can become excessively large. As a result, matrix norms of the Jacobian used for the linearized dynamics become excessively large and can lead to non-negligible linearization errors even for very small timesteps.

To address this, ϕ and θ values substituted into the pre-calculated symbolic Jacobian matrix were artificially bounded to provide a buffer around these singularity points. In practice, a $\pm 8 \times 10^{-2}$ rad (4.6 deg) for ϕ around the vertical orientation and a 1 $\frac{rad}{s}$ bound on the angular velocity $\dot{\theta}$ works well, without sacrificing too much accuracy. A smaller buffer lends to dynamics which are susceptible to poor ill-conditioned numerical behavior, while a larger buffer threshold causes inaccurate representations of the state of the system. In practice, this hyperparameter is tuned through trial and error.

Nevertheless, the dynamic singularities clearly highlighted the need for a different representation that is simple, versatile, and applicable for a wide range of tensegrity geometries.

3.6 Simplified Point Mass Dynamics

In the previous section, a minimal representation of the 3D rigid body dynamics for tensegrity systems is presented. While the reduced state-dimension of this minimal representation is advantageous, this particular formulation is susceptible to two major drawbacks: (1) dynamic singularities exist due to the spherical coordinates representation which can make robust and reliable motion planning and optimal control difficult; and (2) sharp discontinuities due to contact collision with the environment can lead to stiff ODEs and drastic changes in dynamics from one timestep to the next, presenting a challenge for finite-horizon optimal controllers. In this section, we present a novel simplified point mass tensegrity dynamics representation originally published and presented at the *International Conference on Intelligent Robots and Systems (IROS) Conference 2019* [20] which addresses these shortcomings and can be easily formulated for any Class-1 tensegrity structure.

For this point mass formulation, rather than representing true rigid body dynamics, we assume that the entirety of each rod's mass can be distributed between two point masses located at the ends of the rod, hereafter also refered to as nodes. Admittedly, this assumption's validity is largely dependent on the actual hardware design of the tensegrity robot; as an example, consider the tensegrity SUPERball [93] designed by the Intelligent Robotics Group at NASA Ames, which bears the majority of each rod's mass closer to the ends of the rod where heavier motor assemblies and electronics are housed. This point mass assumption can often be relatively accurate and greatly simplifies the formulation of tensegrity dynamic equations of motion, enabling rapid design and prototyping of new innovative topologies in simulation.

With these simplifying assumptions, we now consider only the positions, velocities, and accelerations of all point masses. Comparatively, this state vector has 36 states, as opposed the minimal q-coordinate frame with 30 states introduced in the previous section. We define vectors $\mathbf{p}$ and $\dot{\mathbf{p}} \in \mathbb{R}^{3N}$ containing the individual XYZ positions and velocities of the N nodes as:

$$\mathbf{p} = [X_1, Y_1, Z_1, X_2, Y_2, Z_2, ..., X_N, Y_N, Z_N]^{\mathbf{T}} \tag{3.15}$$

$$\dot{\mathbf{p}} = [\dot{X}_1, \dot{Y}_1, \dot{Z}_1, \dot{X}_2, \dot{Y}_2, \dot{Z}_2, ..., \dot{X}_N, \dot{Y}_N, \dot{Z}_N]^{\mathbf{T}} \tag{3.16}$$

Importantly, this dynamics representation assumes that forces are imparted on each node purely through idealized two-force members (i.e., the rods and cables in pure compression/tension) or from the external environment (e.g., contact forces with the ground). This simplifying assumption enables this representation to be versatile enough to quickly represent a broad range of tensegrity topologies.

3.6.1 Series-Elastic Cable Forces

Forces which act on the nodes due to the spring-cables are calculated in a similar manner as in Section 3.5.1, again with special consideration that no compressive forces can be applied through the cables:

$$\mathbf{F}_{cables,j} = max\{\ 0\ ,\ k_j(s_j - \ell_j)\ \} \tag{3.17}$$

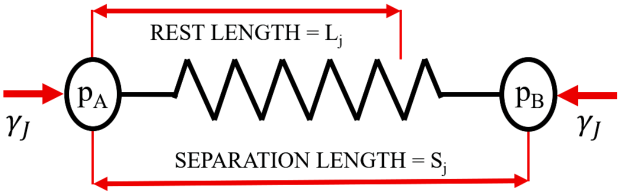


Figure 3.3: Hookean linear-elastic model between two point masses.

Here k_j is the stiffness of the series-elastic cable j, s_j is the separation distance between the two end nodes attached to cable j, and ℓ_j is the spring-cable assembly rest length (see Fig. 3.3).

Given a cable connectivity matrix $\mathbf{C} \in \mathbb{R}^{J \times N}$ such as Eq. 4.35, with rows $\mathbf{C}_j$ that encode cable interconnections between pairs of nodes, we represent the vector of Cartesian forces due to tension in cable j, $\boldsymbol{\gamma}_j \in \mathbb{R}^{3N}$, as:

$$\boldsymbol{\gamma}_j = \sqrt{2} k_j \cdot softPlus\Big(\boldsymbol{\alpha}_j, \beta\Big) \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \quad \forall j \in \{1, ..., J\} \tag{3.18}$$

where variables $\mathbf{z}_j$ and $\boldsymbol{\alpha}_j$ are defined as follows:

$$\begin{aligned} \mathbf{z}_j &= - \left[\mathbf{C}_j^T \mathbf{C}_j \otimes \mathbf{I}_3\right] \mathbf{p} \\ \boldsymbol{\alpha}_j &= \ (s_j - \ell_j) \ = \sqrt{\frac{\mathbf{z}_j^T \mathbf{z}_j}{2}} - \ell_j \end{aligned}$$

$\mathbf{z}_j \in \mathbb{R}^J$ is a structured sparse vector that contains the directional vector lying along the direction of the cable j. The *softPlus* function above is a smooth approximation to the non-differentiable rectifier function (used in Eq. 3.17), approximating $max\{0, \boldsymbol{\alpha}_i\}$ with tunable smoothness parameter β (see Fig. 3.4 for a comparison):

$$softPlus(\alpha, \beta) := \frac{\sqrt{\alpha^2 + \beta^2} + \alpha}{2} \ \geq \ 0 \tag{3.19}$$

In practice, this Lipschitz smooth approximation demonstrates better numerical stability in simulation and its continuously differentiable property is well-suited for the locally-linearized dynamic models used in the receding horizon control methods discussed in later chapters.

3.6.2 Numerically Robust Constraint Forces

The other set of essential forces in tensegrity structures are the rigid body constraints, which constrain the rod-defined nodal positions relative to each other and the environment.

Rather than model the rods using a linear-elastic model as with cable forces in the prior section, we instead adopt a constrained dynamics approach. The motivation behind this is

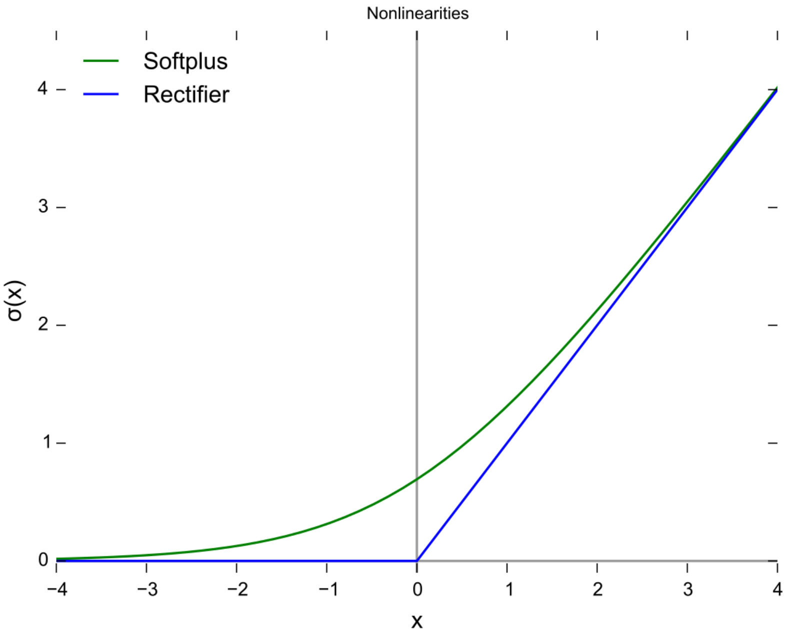


Figure 3.4: Comparison of the linear rectifier and the smooth softplus approximation.

that *penalty* or *energy barrier methods* [108][50], which rely on restorative forces to maintain rigid connections, necessitate large stiffness parameters and lead to stiff differential equations. That is, numerical methods may be unstable for solving the ODEs unless step sizes are chosen to be extremely small. Instead, the constraint forces we describe here analytically cancel out the components of the applied forces that violate rigid constraints at each timestep, creating accurate and numerically tractable dynamic simulations.

In our work [21], we adopt a similar approach to [106] and define constraint vectors, $\mathbf{G}(\mathbf{p})$ and $\dot{\mathbf{G}}(\mathbf{p}) \in \mathbb{R}^m$, to represent the implicit constraint functions and their time derivatives which must be satisfied at all times, where m is the number of active dynamic constraints. Each scalar-valued function $\mathbf{G}_i(\mathbf{p})$ is a single implicit constraint function that is satisfied when equal to zero. If we assume that initial positions and velocities of the system satisfy dynamic constraints (i.e., $\mathbf{G}(\mathbf{p}) = 0$ and $\dot{\mathbf{G}}(\mathbf{p}) = 0$), then any forces which maintain legal accelerations (i.e., $\ddot{\mathbf{G}}(\mathbf{p}) = 0$) will be valid forces satisfying all dynamic constraints.

We decompose the vector of *legal* forces $\mathbf{F}_{legal}$ which are ultimately applied to the particle masses into two components: 1) $\mathbf{F}_{total}$, the total forces originally applied to the particle, and 2) $\hat{\mathbf{F}}$, which are resultant constraint forces that strictly cancel out any illegal accelerations. From the principle of *virtual work* (i.e., that the virtual constraint force never add or remove energy from the system), it is required that:

$$\hat{\mathbf{F}} \cdot \dot{\mathbf{p}} = 0, \quad \forall\ \dot{\mathbf{p}} \ \mid\ \dot{\mathbf{G}} = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{p}}\right)\dot{\mathbf{p}} = 0 \tag{3.20}$$

All force vectors which satisfy this constraint can thus be written in the form:

$$\hat{\mathbf{F}} = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{p}}\right)^T \boldsymbol{\lambda} \tag{3.21}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ can be seen as a vector of Lagrange multipliers. Specifically, the virtual constraint force $\hat{\mathbf{F}}$ must lie in the *null space complement* of $\frac{\partial \mathbf{G}}{\partial \mathbf{p}}$. The rows of this matrix are the gradients of each scalar constraint function with respect to particle positions and thus are a collection of normal vectors to the permissible constraint hypersurfaces where the particles are permitted to move without violating any explicit constraints. By restricting the virtual constraint forces to the linear combination of normal constraint gradients, we ensure that any dot products of the virtual force with legal particle displacements will be strictly zero, satisfying the conditions of the principle of virtual work.

We also introduce the inverse-mass matrix $\mathbf{W} \doteq \mathbf{M}^{-1}$, which has the reciprocal of each particle's mass as elements along the diagonal. By applying the chain rule and substituting for $\ddot{\mathbf{p}}$, the legal acceleration condition can be written as:

$$\ddot{\mathbf{G}}(\mathbf{p}) = \left[\frac{\partial \dot{\mathbf{G}}(\mathbf{p})}{\partial \mathbf{p}}\right] \dot{\mathbf{p}} + \left[\frac{\partial \mathbf{G}(\mathbf{p})}{\partial \mathbf{p}}\right] \ddot{\mathbf{p}} \tag{3.22}$$

$$= \left[\frac{\partial \dot{\mathbf{G}}(\mathbf{p})}{\partial \mathbf{p}}\right] \dot{\mathbf{p}} + \left[\frac{\partial \mathbf{G}(\mathbf{p})}{\partial \mathbf{p}}\right] \mathbf{W}(\mathbf{F} + \hat{\mathbf{F}}) \;=\; \mathbf{0} \tag{3.23}$$

Simplifying notation of $\frac{\partial \mathbf{G}(\mathbf{p})}{\partial \mathbf{p}}$ and $\frac{\partial \dot{\mathbf{G}}(\mathbf{p})}{\partial \mathbf{p}}$ as matrices $\mathbf{J}(\mathbf{p})$ and $\dot{\mathbf{J}}(\mathbf{p})$, respectively, and dropping the matrices' explicit dependencies on $\mathbf{p}$, we manipulate and rewrite Eq. 3.23 as:

$$\mathbf{JW}\hat{\mathbf{F}} \;=\; -\dot{\mathbf{J}}\dot{\mathbf{p}} - \mathbf{JWF} \tag{3.24}$$

$$\mathbf{JW}(\mathbf{J^T}\boldsymbol{\lambda}) \;=\; -\dot{\mathbf{J}}\dot{\mathbf{p}} - \mathbf{JWF} \tag{3.25}$$

where Eq. 3.25 is a result of applying the principle of virtual work and substituting in Eq. 3.21. The vector $\boldsymbol{\lambda}$ of Lagrange multipliers determines how much of each constraint is applied. Each value represents the magnitude of the reaction force applied due to the corresponding constraint. To counteract the accumulation of numerical drift from floating point errors, corrective stiffness and damping terms are appended to (Eq. 3.25) as proportional and derivative error correction factors:

$$\mathbf{JW}(\mathbf{J^T}\boldsymbol{\lambda}) = -\dot{\mathbf{J}}\dot{\mathbf{p}} - \mathbf{JWF} - \mathbf{k_s G} - \mathbf{k_d}\dot{\mathbf{G}} \tag{3.26}$$

As a concrete example, let us consider the constraint forces imposed by the rigid body connection between two endpoint nodes of a rod. Given nodal positions and velocities, rod length $\boldsymbol{\ell}_{rod,q}$, and a rod connectivity matrix $\mathbf{R} \in \mathbb{R}^{Q \times N}$ with rows $\mathbf{R}_q$ that encode rod interconnections, we write an implicit rod constraint function constraining relative distance between nodal positions $\mathbf{p}_A$ and $\mathbf{p}_B$, the constraint function's respective time derivative, and their associated Jacobian matrices as:

$$\begin{aligned}
\mathbf{G}_i(\mathbf{p}) &= \|\mathbf{p_B} - \mathbf{p_A}\|_\mathbf{2}^\mathbf{2} - \ell_{rod,q}^2 && (3.27)\\
&= \frac{\mathbf{1}}{\mathbf{2}}\mathbf{p^T}([\mathbf{R_q^T R_q}] \otimes \mathbf{I_3})^\mathbf{2}\,\mathbf{p} - \ell_{rod,q}^2 && (3.28)\\
\dot{\mathbf{G}}_i(\mathbf{p},\dot{\mathbf{p}}) &= \mathbf{p^T}([\mathbf{R_q^T R_q}] \otimes \mathbf{I_3})^\mathbf{2}\,\dot{\mathbf{p}} && (3.29)\\
\mathbf{J}_i(\mathbf{p}) &= \mathbf{p^T}([\mathbf{R_q^T R_q}] \otimes \mathbf{I_3})^\mathbf{2} && (3.30)\\
\dot{\mathbf{J}}_i(\mathbf{p},\dot{\mathbf{p}}) &= \dot{\mathbf{p}}^\mathbf{T}([\mathbf{R_q^T R_q}] \otimes \mathbf{I_3})^\mathbf{2} && (3.31)
\end{aligned}$$

Given these implicit constraint functions which are obtained for each rod, we combine these results with the formulas in Section 3.6.1 to guarantee that nodal accelerations are accurately simulated, with no pair-relative acceleration components lying along the longitudinal axis of the rigid rods.

3.6.3 Ground Contact and External Forces

Ground contact forces are calculated in a similar manner to the approach introduced in Section 3.5.3, with a few additions to help improve numerical stability during simulation: (1) viscous damping is added in the upward direction to prevent excessive clipping (i.e., the robot geometry entering into the ground due to high downward velocities), and (2) horizontal viscous forces were added to provide realistic traction for rolling mobility.

The vertical ground contact forces are applied to each node in the Z-direction and uses the smooth approximation to the linear rectifier (Eq. 3.19) as well as its derivative, which acts a smooth approximation (one of many possible choices) to the unit step function:

$$softStep(\alpha,\beta) \doteq \frac{\partial}{\partial\alpha} softPlus(\alpha,\beta) = \frac{1}{2}\left(\frac{\alpha}{\sqrt{\alpha^2+\beta^2}}+1\right) \qquad (3.32)$$

The upwards vertical force as a result of nodal ground contact is therefore a function of floor penetration in the Z-direction, downward Z-velocity, and whether or not the node is in contact with the ground in the first place:

$$\mathbf{F}_{z,i} = softPlus(Z_{floor} - Z_i, \beta) - c \cdot softStep(Z_{floor} - Z_i, \beta)\dot{Z}_i \qquad (3.33)$$

where c is a damping coefficient and β is a smoothness parameter (in most of the simulations in this work, c and β are 3×10^2 and 1×10^{-3} respectively).

Finally, viscous floor friction is applied in a similar manner to all nodes which are in contact with the ground:

$$\begin{aligned}
\mathbf{F}_{x,i} &= -c_{vf} \cdot softStep(Z_{floor} - Z_i, \beta)\dot{X}_i && (3.34)\\
\mathbf{F}_{y,i} &= -c_{vf} \cdot softStep(Z_{floor} - Z_i, \beta)\dot{Y}_i && (3.35)
\end{aligned}$$

3.6.4 Complete Tensegrity Point Mass Accelerations

Using a first principles approach to obtain the particle accelerations, we form $\ddot{\mathbf{p}}$ as follows:

$$\begin{aligned}
\ddot{\mathbf{p}} &= \mathbf{W} \cdot \mathbf{F}_{legal} && (3.36)\\
&= \mathbf{W} \cdot \big(\mathbf{F}_{total} + \hat{\mathbf{F}}\big) && (3.37)\\
&= \mathbf{W} \cdot \big(\mathbf{F}_{total} + \mathbf{J}^{\mathbf{T}}\boldsymbol{\lambda}\big) && (3.38)\\
&= \mathbf{W} \cdot \big(\mathbf{F}_{total} + \mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathbf{T}})^{-1}[-\dot{\mathbf{J}}\dot{\mathbf{p}} - \mathbf{J}\mathbf{W}\mathbf{F}_{total} - k_s\mathbf{G} - k_d\dot{\mathbf{G}}]\big) && (3.39)\\
&= \mathbf{W} \cdot \Big(- \mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathbf{T}})^{\mathbf{-1}}(\dot{\mathbf{J}}\dot{\mathbf{p}} - k_s\mathbf{G} - k_d\dot{\mathbf{G}}) + \\
&\qquad (\mathbf{I}_{3N} - \mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathbf{T}})^{\mathbf{-1}}\mathbf{J}\mathbf{W})(\sum_j \boldsymbol{\gamma}_j + \mathbf{F}_{ext})\Big) && (3.40)
\end{aligned}$$

Eq. 3.39 is a direct result of substituting in Eq. 3.26, and the last line (Eq. 5.16) is obtained by combining the results of Sections 3.6.1 to 3.6.3. Note, $\mathbf{F}_{ext}$ is the vector of the total forces applied to the tensegrity robot which are external to the system (e.g., ground contact reaction forces) and are calculated using damped linear-elastic collisions previously presented above in Section 3.6.3.

The structured dynamics that result for the overall system are highly-coupled, nonlinear equations of motion. From these equations, one may deduce that it can quickly become difficult to intuitively reason about how local changes in an individual cable's tension may affect the overall geometry of the global system. Nevertheless, recent work such as [23] utilized hand-engineered open-loop actuation policies with two cables at a time that demonstrated good results for fast uphill climbing. This becomes increasingly difficult, however, as more cables are considered, with complete actuation of all 24 cables presenting a daunting combinatorial problem. For this reason, later chapters present iterative locally optimal control techniques and model predictive control approaches which can take advantage of the easily obtained mathematical models of the system to generate optimal actuation policies.

3.7 Linearization and Discretization

The continuous dynamics of the robot (specifically the nonlinear accelerations) can be linearized about the robot's current state and discretized using a first-order Euler approximation:

$$\begin{bmatrix} \mathbf{p}_{k+1} \\ \dot{\mathbf{p}}_{k+1} \\ \boldsymbol{\ell}_{cable,k+1} \\ \boldsymbol{\ell}_{rod,k+1} \end{bmatrix} \approx \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cable,k} \\ \boldsymbol{\ell}_{rod,k} \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\mathbf{p}}_{\mathbf{k}} \\ \ddot{\mathbf{p}}_{\mathbf{0}} + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}}\tilde{\mathbf{x}}_{\mathbf{k}} \\ \dot{\boldsymbol{\ell}}_{cable,k+1} \\ \dot{\boldsymbol{\ell}}_{rod,k+1} \end{bmatrix} \qquad (3.41)$$

or, for greater accuracy at the cost of more computation, a trapezoidal approximation:

$$\begin{bmatrix} \mathbf{p}_{k+1} \\ \dot{\mathbf{p}}_{k+1} \\ \boldsymbol{\ell}_{cable,k+1} \\ \boldsymbol{\ell}_{rod,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cable,k} \\ \boldsymbol{\ell}_{rod,k} \end{bmatrix} + \frac{\Delta t}{2} \left(\begin{bmatrix} \dot{\mathbf{p}}_k + \dot{\mathbf{p}}_{k+1} \\ 2\ddot{\mathbf{p}}_0 + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}} \tilde{\mathbf{x}}_k + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}} \tilde{\mathbf{x}}_{k+1} \\ \dot{\boldsymbol{\ell}}_{cable,k} + \dot{\boldsymbol{\ell}}_{cable,k+1} \\ \dot{\boldsymbol{\ell}}_{rod,k} + \dot{\boldsymbol{\ell}}_{rod,k+1} \end{bmatrix} \right) \tag{3.42}$$

where Δt is the simulation timestep, the state $\mathbf{x} \in \mathbb{R}^{102}$ is a concatenated vector of cable lengths, rod lengths, and nodal position/velocity states, $\tilde{\mathbf{x}}_{\mathbf{k}}$ is the deviation about the linearization point $\mathbf{x_0}$, and $\ddot{\mathbf{p}}_{\mathbf{0}}$ is the vector of exact nodal accelerations calculated at the linearization point $\mathbf{x_0}$. Depending on the hardware sensors, actuators, and exact controller implementation, the inputs to the system may be desired cable velocities, cable accelerations, and/or rod velocities and accelerations. Traditionally, hardware robots that we work with in the B.E.S.T. lab are controlled through position control of cable rest lengths, although cable velocities and acceleration inputs may be controlled for smoother control input trajectories. Approaches which modify the dynamics to incorporate an augmented input of cable/rod length velocities and/or accelerations are discussed in later chapters.

3.8 Custom Tensegrity MATLAB Simulation Framework

A custom-designed simulation environment was developed in MATLAB for simulation of the tensegrity robot and easy interfacing with existing optimization toolboxes (see example graphic in Fig. 3.5). Using the dynamics described in the previous section, a framework for simulation of the robot was created. With ease-of-use and usability in mind, the intent of the simulation environment is to be a general-purpose tool for simulation and motion planning of any Class-1 tensegrity system. Due to the structured nature of the tensegrity dynamics, as discussed in section previous sections, construction of idealized models for Class-1 topologies is relatively straightforward. However, during development, several key insights were made evident, and additional features in the simulation were incrementally added as needs progressed. What began as a simple simulation of a free-floating tensegrity in the absence of gravity has evolved into a relatively comprehensive simulation environment for tensegrities in contact-rich dynamics. The work-in-progress Github repository can be found at: `github.com/brianmcera/Tensegrity-Dynamics-and-Optimal-Control`.

3.8.1 Modular Object-oriented Approach

The primary motivation for creating this simulation environment is to enable rapid prototyping of different controllers, estimators, and motion planning techniques without the need to operate on experimental hardware. Additionally, this simulation environment allows for iterative testing of different hardware designs and tensegrity topologies, such as adjusting

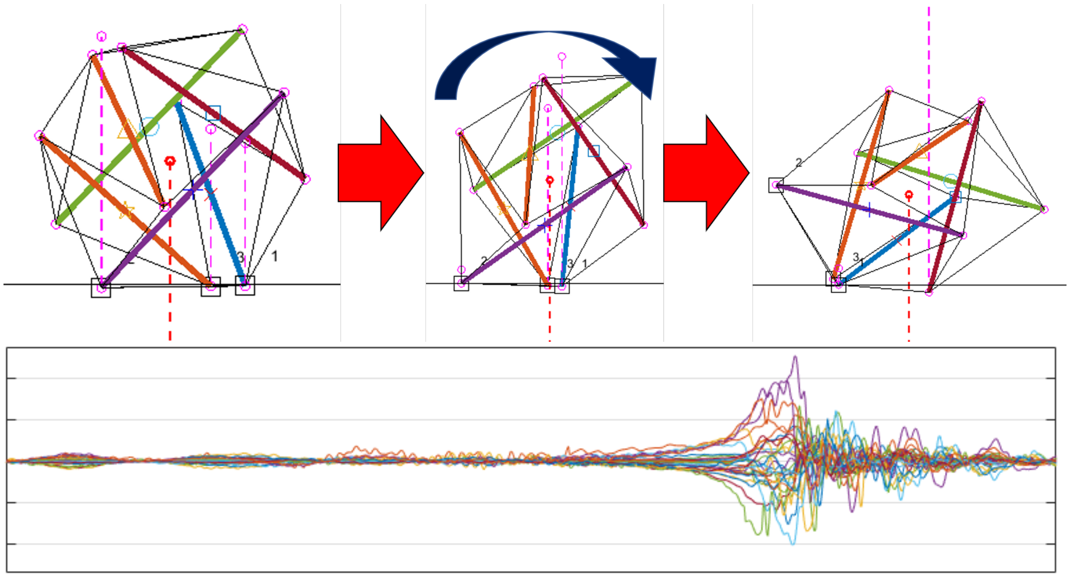


Figure 3.5: Top: MATLAB simulation of six-bar spherical tensegrity rolling using 24-cable actuation, at rest (left), initial tipping instability (middle), and impact on the next base (right). Bottom: Corresponding linear velocities [m/s] and angular velocities [rad/s] of all rods (30-dimensional velocity vector).

motor capabilities, sensor placement, component specifications, etc. Tensegrity hardware, with their many actuators and complex mechanical designs, are currently a challenging platform to work with. To improve research and development with tensegrity hardware in the loop, many advances must be achieved in terms of reliability and ease-of-use. In the meantime, this simulation framework has proven to be an invaluable software tool for evaluating novel approaches and exploring different research topics in tensegrity robotics.

To facilitate the evaluation of different interconnected components, careful consideration was given to ensure that a modular design paradigm was consistent throughout the simulation framework. It became evident early on that having a monolithic coding design which combined all aspects of the robot's dynamics, controls, and state estimation would not enable quick evaluation of different design trade-offs (e.g., using a specific controller approach such as MPC for a specific tensegrity design would require completely separate code than a tensegrity spine robot using differential dynamic programming). Instead, we adopt a object-oriented approach which isolates individual components of the simulation loop outlined in Figure 3.6. A high-level API guarantees that these components can interact with one another in a universal manner, meaning that different controllers can be easily combined with different observers to control any variety of tensegrity dynamics models.

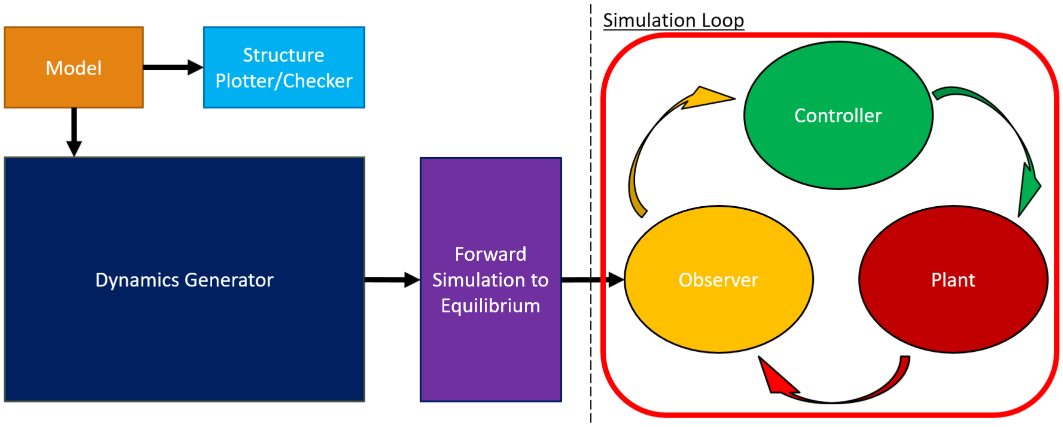


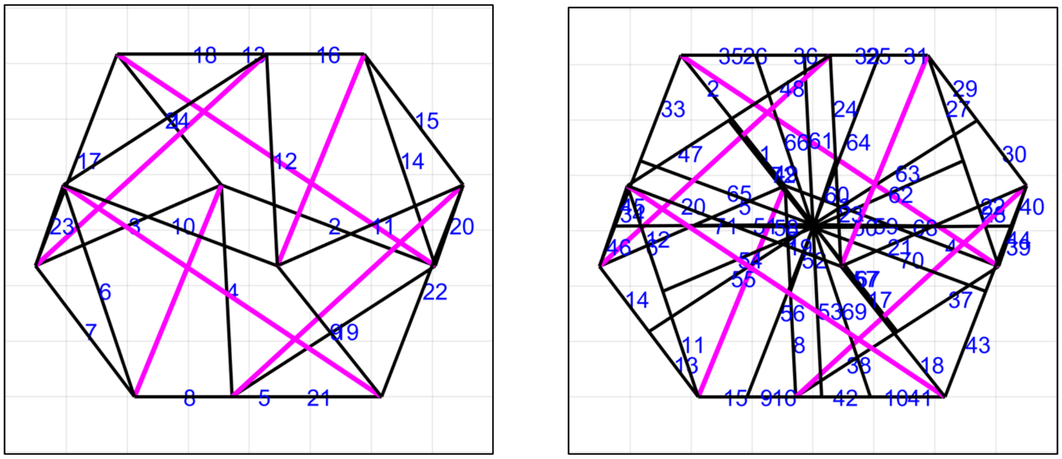
Figure 3.6: Overview of custom MATLAB simulation framework.

3.8.2 Automatic Generation of Dynamics Equations

The primary powerhouse of this simulation environment is the automated symbolic dynamics generator (represented in navy blue in Figure 3.6) which enables quick calculation of the dynamic equations of motion for any Class-1 tensegrity system. In contrast to previous work in the lab, which required hand-calculation or numerical approximation of linearized dynamics for model-based approaches, this framework produces differentially smooth dynamics equations (see Figure 3.7 and Figure 3.8) and analytically-obtained symbolic Jacobian matrices. Due to the continuously differentiable equations of motion and simple-to-obtain analytical Jacobian matrices, this dynamics formulation enables us to quickly prototype different controllers and state estimators which rely on smooth assumptions and linearized approximations.

3.8.3 State Initialization with Kinetic Energy Damping

The need for proper initialization of the robot state was apparent early on in the development of the simulation environment. In the earliest naive approaches, the 6-bar tensegrity was initialized using a regular icosahedron assumption (i.e., rods and nodal positions were defined assuming the outer surface of the spherical tensegrity had equilateral triangles for each of its 20 faces). In practice, this initialization did not work for two reasons - early implementations of the code either: 1) did not take initial desired pretension into account (so that the robot would immediately collapse) or 2) didn't account for drastic transient effects that occurred once the robot was initialized and released from the initial state. Specifically, if cable rest lengths were immediately set by a desired pretension rather than incrementally updated, excessively large accelerations caused the simulation to be unstable, regardless of timestep size. Additionally, these large accelerations made it difficult for the model-based controllers to initially find optimal input sequences, as the early dynamics of the robots were dominated by the erroneous effects described above.



(a) The canonical 6-bar spherical tensegrity.

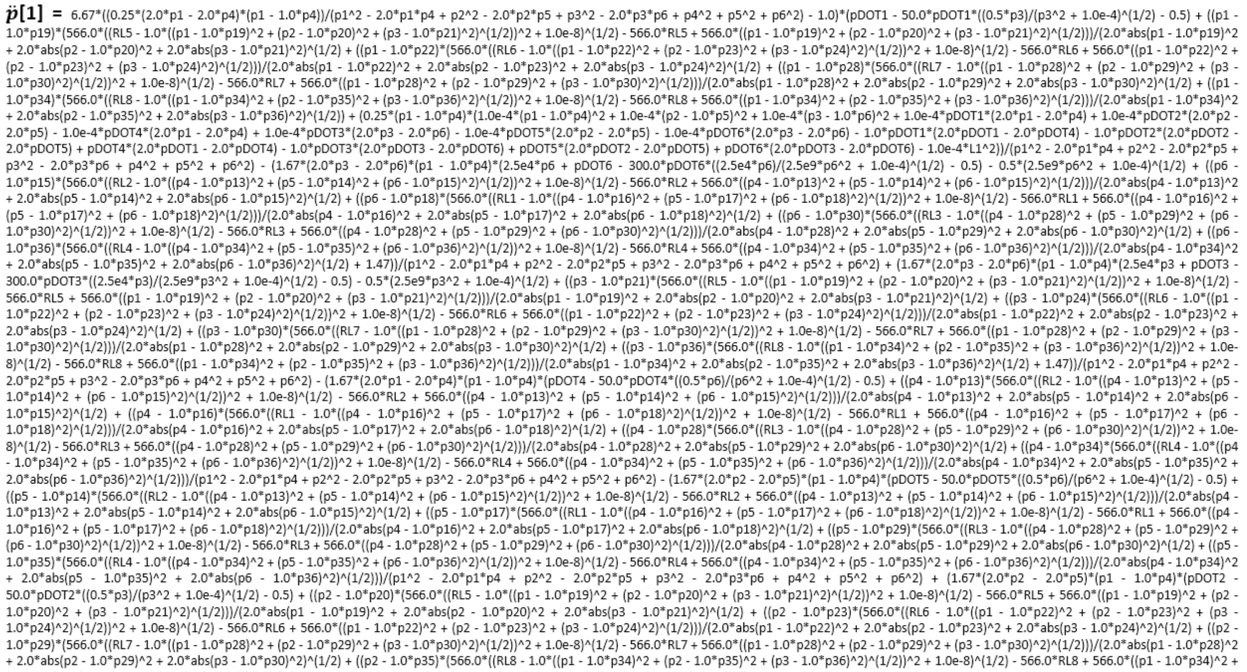
(b) A spherical tensegrity with central-payload suspended by an additional 24 internal cables.

Figure 3.7: Two different types of Class-1 spherical tensegrities. Rods are depicted in magenta; cables are shown in black and numbered in blue. Despite the difference in number of cables and connection locations, we can generate contact-rich dynamic equations of motions for both on the fly.

To address this, simulations now begin by first finding nominal states, using the regular-icosahedron assumption described above, and then forward simulating the dynamics until equilibrium (depicted by the purple block in Figure 3.6). Nominal nodal separation distances are first calculated, then a desired pretension determines the appropriate cable rest length set points relative to these initial distances. In contrast to the naive approach above, cables are gradually controlled to these rest lengths rather than immediately set, to mitigate excessively large forces and accelerations. To speed up this initialization, energy of the system is observed at each time step and *kinetic energy damping* is employed (i.e., nodal velocities are set to zero each time kinetic energy of the system peaks). This artificial damping is not representative of the true dynamics of the system but offers a drastic reduction in the number of computation iterations necessary during initialization to reach equilibrium.

3.8.4 Example Applications for Design Analyses

The simulation framework above allows us to rapidly prototype various experimental tensegrity designs and topologies, through faster brainstorming and ideation workflows than is possible with empirical hardware tests. While the primary focus of the simulations is to enable the comprehensive evaluation of different software controllers and observers, the simulation environment is also suitable for simple, rudimentary dynamic tests which can provide valuable information for quick mechanical design and trade-off analysis.



```
p̈[1] = 6.67*((0.25*(2.0*p1 - 2.0*p4)*(p1 - 1.0*p4))/(p1^2 - 2.0*p1*p4 + p2^2 - 2.0*p2*p5 + p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2) - 1.0)*(pDOT1 - 50.0*pDOT1*((0.5*p3)/(p3^2 + 1.0e-4)^(1/2) - 0.5) + ((p1 -
1.0*p19)*(566.0*((RL5 - 1.0*((p1 - 1.0*p19)^2 + (p2 - 1.0*p20)^2 + (p3 - 1.0*p21)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL5 + 566.0*((p1 - 1.0*p19)^2 + (p2 - 1.0*p20)^2 + (p3 - 1.0*p21)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p19)^2
+ 2.0*abs(p2 - 1.0*p20)^2 + 2.0*abs(p3 - 1.0*p21)^2)^(1/2) + ((p1 - 1.0*p22)*(566.0*((RL6 - 1.0*((p1 - 1.0*p22)^2 + (p2 - 1.0*p23)^2 + (p3 - 1.0*p24)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL6 + 566.0*((p1 - 1.0*p22)^2 +
(p2 - 1.0*p23)^2 + (p3 - 1.0*p24)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p22)^2 + 2.0*abs(p2 - 1.0*p23)^2 + 2.0*abs(p3 - 1.0*p24)^2)^(1/2) + ((p1 - 1.0*p28)*(566.0*((RL7 - 1.0*((p1 - 1.0*p28)^2 + (p2 - 1.0*p29)^2 + (p3 -
1.0*p30)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL7 + 566.0*((p1 - 1.0*p28)^2 + (p2 - 1.0*p29)^2 + (p3 - 1.0*p30)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p28)^2 + 2.0*abs(p2 - 1.0*p29)^2 + 2.0*abs(p3 - 1.0*p30)^2)^(1/2) + ((p1 -
1.0*p34)*(566.0*((RL8 - 1.0*((p1 - 1.0*p34)^2 + (p2 - 1.0*p35)^2 + (p3 - 1.0*p36)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL8 + 566.0*((p1 - 1.0*p34)^2 + (p2 - 1.0*p35)^2 + (p3 - 1.0*p36)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p34)^2
+ 2.0*abs(p2 - 1.0*p35)^2 + 2.0*abs(p3 - 1.0*p36)^2)^(1/2)) + (0.25*(p1 - 1.0*p4)*(1.0e-4*(p1 - 1.0*p4)^2 + 1.0e-4*(p2 - 1.0*p5)^2 + 1.0e-4*(p3 - 1.0*p6)^2 + 1.0e-4*pDOT1*(2.0*p1 - 2.0*p4) + 1.0e-4*pDOT2*(2.0*p2 -
2.0*p5) - 1.0e-4*pDOT4*(2.0*p1 - 2.0*p4) + 1.0e-4*pDOT3*(2.0*p3 - 2.0*p6) - 1.0e-4*pDOT5*(2.0*p2 - 2.0*p5) - 1.0e-4*pDOT6*(2.0*p3 - 2.0*p6) - 1.0*pDOT1*(2.0*pDOT1 - 2.0*pDOT4) - 1.0*pDOT2*(2.0*pDOT2 -
2.0*pDOT5) + pDOT4*(2.0*pDOT1 - 2.0*pDOT4) - 1.0*pDOT3*(2.0*pDOT3 - 2.0*pDOT6) + pDOT5*(2.0*pDOT2 - 2.0*pDOT5) + pDOT6*(2.0*pDOT3 - 2.0*pDOT6) - 1.0e-4*L1^2))/(p1^2 - 2.0*p1*p4 + p2^2 - 2.0*p2*p5 +
p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2) - (1.67*(2.0*p3 - 2.0*p6)*(p1 - 1.0*p4)*(2.5e4*p6 + pDOT6 - 300.0*pDOT6*((2.5e4*p6)/(2.5e9*p6^2 + 1.0e-4)^(1/2) - 0.5) - 0.5*(2.5e9*p6^2 + 1.0e-4)^(1/2) + ((p6 -
1.0*p15)*(566.0*((RL2 - 1.0*((p4 - 1.0*p13)^2 + (p5 - 1.0*p14)^2 + (p6 - 1.0*p15)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL2 + 566.0*((p4 - 1.0*p13)^2 + (p5 - 1.0*p14)^2 + (p6 - 1.0*p15)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p13)^2
+ 2.0*abs(p5 - 1.0*p14)^2 + 2.0*abs(p6 - 1.0*p15)^2)^(1/2) + ((p6 - 1.0*p18)*(566.0*((RL1 - 1.0*((p4 - 1.0*p16)^2 + (p5 - 1.0*p17)^2 + (p6 - 1.0*p18)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL1 + 566.0*((p4 - 1.0*p16)^2 +
(p5 - 1.0*p17)^2 + (p6 - 1.0*p18)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p16)^2 + 2.0*abs(p5 - 1.0*p17)^2 + 2.0*abs(p6 - 1.0*p18)^2)^(1/2) + ((p6 - 1.0*p30)*(566.0*((RL3 - 1.0*((p4 - 1.0*p28)^2 + (p5 - 1.0*p29)^2 + (p6 -
1.0*p30)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL3 + 566.0*((p4 - 1.0*p28)^2 + (p5 - 1.0*p29)^2 + (p6 - 1.0*p30)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p28)^2 + 2.0*abs(p5 - 1.0*p29)^2 + 2.0*abs(p6 - 1.0*p30)^2)^(1/2) + ((p6 -
1.0*p36)*(566.0*((RL4 - 1.0*((p4 - 1.0*p34)^2 + (p5 - 1.0*p35)^2 + (p6 - 1.0*p36)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL4 + 566.0*((p4 - 1.0*p34)^2 + (p5 - 1.0*p35)^2 + (p6 - 1.0*p36)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p34)^2
+ 2.0*abs(p5 - 1.0*p35)^2 + 2.0*abs(p6 - 1.0*p36)^2)^(1/2) + 1.47))/(p1^2 - 2.0*p1*p4 + p2^2 - 2.0*p2*p5 + p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2) + (1.67*(2.0*p3 - 2.0*p6)*(p1 - 1.0*p4)*(2.5e4*p3 + pDOT3 -
300.0*pDOT3*((2.5e4*p3)/(2.5e9*p3^2 + 1.0e-4)^(1/2) - 0.5) - 0.5*(2.5e9*p3^2 + 1.0e-4)^(1/2) + ((p3 - 1.0*p21)*(566.0*((RL5 - 1.0*((p1 - 1.0*p19)^2 + (p2 - 1.0*p20)^2 + (p3 - 1.0*p21)^2)^(1/2))^2 + 1.0e-8)^(1/2) -
566.0*RL5 + 566.0*((p1 - 1.0*p19)^2 + (p2 - 1.0*p20)^2 + (p3 - 1.0*p21)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p19)^2 + 2.0*abs(p2 - 1.0*p20)^2 + 2.0*abs(p3 - 1.0*p21)^2)^(1/2) + ((p3 - 1.0*p24)*(566.0*((RL6 - 1.0*((p1 -
1.0*p22)^2 + (p2 - 1.0*p23)^2 + (p3 - 1.0*p24)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL6 + 566.0*((p1 - 1.0*p22)^2 + (p2 - 1.0*p23)^2 + (p3 - 1.0*p24)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p22)^2 + 2.0*abs(p2 - 1.0*p23)^2 +
2.0*abs(p3 - 1.0*p24)^2)^(1/2) + ((p3 - 1.0*p30)*(566.0*((RL7 - 1.0*((p1 - 1.0*p28)^2 + (p2 - 1.0*p29)^2 + (p3 - 1.0*p30)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL7 + 566.0*((p1 - 1.0*p28)^2 + (p2 - 1.0*p29)^2 + (p3 -
1.0*p30)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p28)^2 + 2.0*abs(p2 - 1.0*p29)^2 + 2.0*abs(p3 - 1.0*p30)^2)^(1/2) + ((p3 - 1.0*p36)*(566.0*((RL8 - 1.0*((p1 - 1.0*p34)^2 + (p2 - 1.0*p35)^2 + (p3 - 1.0*p36)^2)^(1/2))^2 + 1.0e-
8)^(1/2) - 566.0*RL8 + 566.0*((p1 - 1.0*p34)^2 + (p2 - 1.0*p35)^2 + (p3 - 1.0*p36)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p34)^2 + 2.0*abs(p2 - 1.0*p35)^2 + 2.0*abs(p3 - 1.0*p36)^2)^(1/2) + 1.47))/(p1^2 - 2.0*p1*p4 + p2^2 -
2.0*p2*p5 + p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2) - (1.67*(2.0*p1 - 2.0*p4)*(p1 - 1.0*p4)*(pDOT4 - 50.0*pDOT4*((0.5*p6)/(p6^2 + 1.0e-4)^(1/2) - 0.5) + ((p4 - 1.0*p13)*(566.0*((RL2 - 1.0*((p4 - 1.0*p13)^2 + (p5 -
1.0*p14)^2 + (p6 - 1.0*p15)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL2 + 566.0*((p4 - 1.0*p13)^2 + (p5 - 1.0*p14)^2 + (p6 - 1.0*p15)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p13)^2 + 2.0*abs(p5 - 1.0*p14)^2 + 2.0*abs(p6 -
1.0*p15)^2)^(1/2) + ((p4 - 1.0*p16)*(566.0*((RL1 - 1.0*((p4 - 1.0*p16)^2 + (p5 - 1.0*p17)^2 + (p6 - 1.0*p18)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL1 + 566.0*((p4 - 1.0*p16)^2 + (p5 - 1.0*p17)^2 + (p6 -
1.0*p18)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p16)^2 + 2.0*abs(p5 - 1.0*p17)^2 + 2.0*abs(p6 - 1.0*p18)^2)^(1/2) + ((p4 - 1.0*p28)*(566.0*((RL3 - 1.0*((p4 - 1.0*p28)^2 + (p5 - 1.0*p29)^2 + (p6 - 1.0*p30)^2)^(1/2))^2 + 1.0e-
8)^(1/2) - 566.0*RL3 + 566.0*((p4 - 1.0*p28)^2 + (p5 - 1.0*p29)^2 + (p6 - 1.0*p30)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p28)^2 + 2.0*abs(p5 - 1.0*p29)^2 + 2.0*abs(p6 - 1.0*p30)^2)^(1/2) + ((p4 - 1.0*p34)*(566.0*((RL4 - 1.0*((p4
- 1.0*p34)^2 + (p5 - 1.0*p35)^2 + (p6 - 1.0*p36)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL4 + 566.0*((p4 - 1.0*p34)^2 + (p5 - 1.0*p35)^2 + (p6 - 1.0*p36)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p34)^2 + 2.0*abs(p5 - 1.0*p35)^2 +
2.0*abs(p6 - 1.0*p36)^2)^(1/2)))/(p1^2 - 2.0*p1*p4 + p2^2 - 2.0*p2*p5 + p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2) - (1.67*(2.0*p2 - 2.0*p5)*(p1 - 1.0*p4)*(pDOT5 - 50.0*pDOT5*((0.5*p6)/(p6^2 + 1.0e-4)^(1/2) - 0.5) +
((p5 - 1.0*p14)*(566.0*((RL2 - 1.0*((p4 - 1.0*p13)^2 + (p5 - 1.0*p14)^2 + (p6 - 1.0*p15)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL2 + 566.0*((p4 - 1.0*p13)^2 + (p5 - 1.0*p14)^2 + (p6 - 1.0*p15)^2)^(1/2)))/(2.0*abs(p4 -
1.0*p13)^2 + 2.0*abs(p5 - 1.0*p14)^2 + 2.0*abs(p6 - 1.0*p15)^2)^(1/2) + ((p5 - 1.0*p17)*(566.0*((RL1 - 1.0*((p4 - 1.0*p16)^2 + (p5 - 1.0*p17)^2 + (p6 - 1.0*p18)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL1 + 566.0*((p4 -
1.0*p16)^2 + (p5 - 1.0*p17)^2 + (p6 - 1.0*p18)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p16)^2 + 2.0*abs(p5 - 1.0*p17)^2 + 2.0*abs(p6 - 1.0*p18)^2)^(1/2) + ((p5 - 1.0*p29)*(566.0*((RL3 - 1.0*((p4 - 1.0*p28)^2 + (p5 - 1.0*p29)^2 +
(p6 - 1.0*p30)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL3 + 566.0*((p4 - 1.0*p28)^2 + (p5 - 1.0*p29)^2 + (p6 - 1.0*p30)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p28)^2 + 2.0*abs(p5 - 1.0*p29)^2 + 2.0*abs(p6 - 1.0*p30)^2)^(1/2) + ((p5 -
1.0*p35)*(566.0*((RL4 - 1.0*((p4 - 1.0*p34)^2 + (p5 - 1.0*p35)^2 + (p6 - 1.0*p36)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL4 + 566.0*((p4 - 1.0*p34)^2 + (p5 - 1.0*p35)^2 + (p6 - 1.0*p36)^2)^(1/2)))/(2.0*abs(p4 - 1.0*p34)^2
+ 2.0*abs(p5 - 1.0*p35)^2 + 2.0*abs(p6 - 1.0*p36)^2)^(1/2)))/(p1^2 - 2.0*p1*p4 + p2^2 - 2.0*p2*p5 + p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2) + (1.67*(2.0*p2 - 2.0*p5)*(p1 - 1.0*p4)*(pDOT2 -
50.0*pDOT2*((0.5*p3)/(p3^2 + 1.0e-4)^(1/2) - 0.5) + ((p2 - 1.0*p20)*(566.0*((RL5 - 1.0*((p1 - 1.0*p19)^2 + (p2 - 1.0*p20)^2 + (p3 - 1.0*p21)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL5 + 566.0*((p1 - 1.0*p19)^2 + (p2 -
1.0*p20)^2 + (p3 - 1.0*p21)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p19)^2 + 2.0*abs(p2 - 1.0*p20)^2 + 2.0*abs(p3 - 1.0*p21)^2)^(1/2) + ((p2 - 1.0*p23)*(566.0*((RL6 - 1.0*((p1 - 1.0*p22)^2 + (p2 - 1.0*p23)^2 + (p3 -
1.0*p24)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL6 + 566.0*((p1 - 1.0*p22)^2 + (p2 - 1.0*p23)^2 + (p3 - 1.0*p24)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p22)^2 + 2.0*abs(p2 - 1.0*p23)^2 + 2.0*abs(p3 - 1.0*p24)^2)^(1/2) + ((p2 -
1.0*p29)*(566.0*((RL7 - 1.0*((p1 - 1.0*p28)^2 + (p2 - 1.0*p29)^2 + (p3 - 1.0*p30)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL7 + 566.0*((p1 - 1.0*p28)^2 + (p2 - 1.0*p29)^2 + (p3 - 1.0*p30)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p28)^2
+ 2.0*abs(p2 - 1.0*p29)^2 + 2.0*abs(p3 - 1.0*p30)^2)^(1/2) + ((p2 - 1.0*p35)*(566.0*((RL8 - 1.0*((p1 - 1.0*p34)^2 + (p2 - 1.0*p35)^2 + (p3 - 1.0*p36)^2)^(1/2))^2 + 1.0e-8)^(1/2) - 566.0*RL8 + 566.0*((p1 - 1.0*p34)^2 +
(p2 - 1.0*p35)^2 + (p3 - 1.0*p36)^2)^(1/2)))/(2.0*abs(p1 - 1.0*p34)^2 + 2.0*abs(p2 - 1.0*p35)^2 + 2.0*abs(p3 - 1.0*p36)^2)^(1/2)))/(p1^2 - 2.0*p1*p4 + p2^2 - 2.0*p2*p5 + p3^2 - 2.0*p3*p6 + p4^2 + p5^2 + p6^2)
```

Figure 3.8: Example of a single symbolic equation of motion representing the acceleration of a single node in the positive X-direction. Note the complex dependencies on other states in the system, namely other nodal positions (variables prefixed by '*p*-'), nodal velocities (variables prefixed by '*pDOT*-'), and cable rest lengths (variables prefixed by '*RL*-').

Asymmetric Drop Testing

One example application which leverages the symbolic dynamics generator is the evaluation of dynamic effects of asymmetric tensegrity weighting. Namely, through a research grant supported by the US Army, Squishy Robotics would like to investigate the efficacy of using an asymmetric weight distribution on the rod ends to have the tensegrity robot reliably land in a specific orientation and/or location (see Figure 3.9).

Running through different modifications of the tensegrity models would typically require an additional recompilation step if using the C++ NASA NTRT simulation toolkit [14] or extensive computation times with long workflow cycles if using high-fidelity FEA software such as ANSYS. Instead, this low-fidelity framework provides preliminary results to help analyze broad, general trends - taking only a few minutes per trial experiment - which can help motivate design decisions and narrow down an overabundance of design variables to a select few which may warrant further examination using more complex and precise FEA simulation tools.

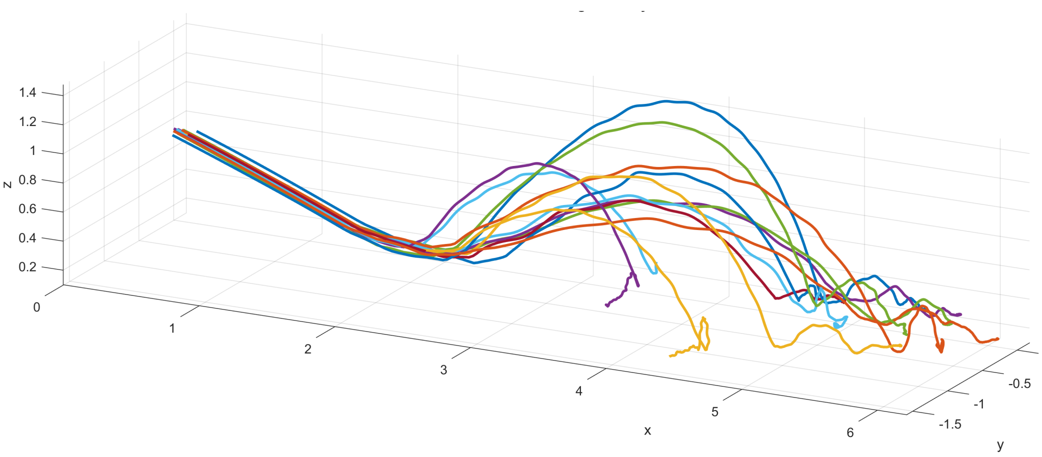


Figure 3.9: Visualization depicting Monte Carlo simulations of robot center-of-mass trajectory during bouncing, starting from the same position but different initial robot pose rotations. All trajectories start with robot center-of-mass roughly at (0,0,1) with an applied horizontal velocity. Results obtained using the simulation framework described in this chapter demonstrate the dynamics' stochastic nature and dependence on robot orientation upon impact.

Center Payload Acceleration Mitigation

Tensegrity structures boast the unique ability to mitigate impact forces using such a lightweight, compact compliant design. As such, many applications of interest are concerned with the tensegrity structure's ability to dissipate forces quickly thoroughout the entire structure, in order to protect a shock-sensitive payload. High-end FEA tools can provide detailed analysis of the stresses and strains propagated throughout the structure during impact, but a high-fidelity simulation make take dozens hours to complete. Instead, designers may be interested in investigating broad trends for a simple trade-off analysis, comparing competing mechanical design considerations with application goals and design criteria. This simulation framework enables rapid prototyping in simulation to evaluate different tensegrity topologies with various mechanical properties in order to quickly motivate design decisions moving forward and to identify potential modes of failure to analyze in greater detail.

Chapter 4

Motion Planning for Multi-cable Ground Mobility with Tensegrities

4.1 Introduction

The primary goal of tensegrity mobility research is to understand how to optimally control individual actuators of tensegrity robots in order to collectively and synchronously achieve a desired dynamic behavior. Tensegrity motion planning and trajectory optimization is challenging due to the high-dimensional, highly-coupled, nonlinear dynamics inherent to tensegrity robots. Specifically, generating optimal state-action trajectories (i.e., the control and time evolution of actuated cable rest lengths and the resulting dynamic states) for the six-bar topology can be a difficult task when considering the entirety of the 96-dimensional state-space and up to 24-dimensional action-space. Fortunately, we are able to leverage the well-structured dynamics of Class-1 tensegrities by importing the dynamic equations of motion derived in the previous chapter as optimization constraints for model-based receding horizon control.

In the case of the six-bar spherical tensegrity topology, we are primarily concerned with generating a performant rolling locomotion policy with respect to speed and energy efficiency. In the recent past, successful results have demonstrated good rolling performance with spherical tensegrities through a variety of sample-based methods, but due to the curse of dimensionality, these sample-based approaches scale poorly as more complex tensegrity topologies are considered. In later chapters, we discuss a few data-driven implementations for tensegrity mobility such as imitation and reinforcement learning. As an alternative, this chapter investigates model-based approaches that leverage the structured dynamics presented in the previous chapter to generate optimal actuation policies and state-action reference trajectories.

4.2 Motion Planning Background

Robotics motion planning and trajectory optimization has a rich history, with many existing algorithms and approaches for handling the important task of determining optimal sequences of actions a robot should take. In the context of tensegrity robotics, the primary challenge lies in identifying those approaches which are able to adequately handle high-dimensional nonlinear dynamics in a scalable manner and obtain good performance under complex differential dynamic constraints.

Sampling-based motion planning approaches such as A* [35], RRT/RRT* [53, 64], or Probablistic Roadmap Methods [32] consider the problem of constructing a motion plan that transfers the robot from some initial state to some terminal goal state. When represented in *configuration space* [60], the motion planning problem transforms from understanding how to manipulate a robot in a complex 3D environment, subject to dynamic/kinematic constraints, into that of understanding how to generate a collision-free path for a point in the *free configuration space.* As an example, a 7-DOF robot manipulator can be represented in a 7-D configuration space where each dimension represents the joint angle of a specific degree of freedom. Obstacles can thus be transformed from Euclidean space to the configuration space by identifying combinations of joint angles which result in collision. Mobile robots, however, often present an even more difficult challenge of *nonholonomic* motion planning, as each degree of freedom is not necessarily independent of others. For example, a car's position is dependent and constrained upon the steering angle and forward velocity, and care must be taken to ensure that generated trajectories are dynamically feasible.

While powerful, sampling-based techniques scale exponentially with both the time horizon and the dimension of the search-space. Additionally, identifying a desired terminal state and determining whether or not it is feasible can be a nontrivial challenge. For tensegrity robots specifically, where we may be interested in finding robot poses which induce an instability that allows punctuated rolling motion, the *form-finding* problem [98, 62] (i.e., that of determining a tensegrity structure's geometric configuration at equilibrium) is an open challenge with various proposed *kinematical* and *statical* methods. A good review of various form-finding approaches for tensegrities can be found in [69]. In short, statical methods determine equilibrium states of the tensegrity structure with a given topology, i.e., the number of nodes, rigid bodies, and cable interconnections between them, while kinematical methods determine the geometry of a tensegrity at equilibrium by maximizing rigid body lengths while keeping cable lengths constant or vice versa. As an example, kinematical dynamic relaxation was employed in [48] for a rod-actuated tensegrity robot, utilizing an evolutionary Monte Carlo approach to find quasi-static poses of the tensegrity that promote rolling. Nevertheless, the challenges of identifying dynamically feasible terminal states for tensegrity robots and poor scalability with state-dimensions discourages sampling-based approaches for more complex tensegrity structures.

Optimization-based approaches instead formulate motion planning as a trajectory optimization problem which seeks to minimize/maximize a cost/reward function subject to optimization constraints such as dynamics constraints and state/input safe sets. In particular, trajectory optimization is concerned with finding the optimal set of actions and resulting states *over time* and is thus well-suited for highly dynamic tasks such as rolling locomotion. In [10], Betts et al. surveys both indirect and direct collation numerical methods for trajectory optimization and presents several approaches for constrained nonlinear programming and related optimal control problems. Variational methods based on the *calculus of variations* [11] can be used to identify functional extrema of a non-convex constrained two-point boundary value problem according to Pontryagin's maximum principle [3, 95] or to obtain a *value function* according to the Hamilton-Jacobi-Bellman equation [70, 71, 7]. The techniques in this chapter rely primarily on these optimization-based approaches and seek to leverage the dynamics formulations presented in the previous chapter.

4.3 Prior Work

Although tensegrity robotics is a relatively new field, successful past examples of tensegrity motion planning and control inspire new and exciting advances. In 2002, Kanchanasaratool et al. demonstrate LQR trajectory tracking on a robotic tensegrity platform in [47, 46]. Although concerned primarily with trajectory tracking of an already given reference trajectory, this work demonstrates some of the earliest attempts at low-level tensegrity control for dynamic motion. In practice, however, a significant challenge of tensegrity mobility lies not in the fine motor input controls, but in identifying how the topology and cable tension distribution should evolve over time in order to achieve a desired overall behavior.

One of the earliest works focusing on spherical tensegrity rolling, [82] demonstrates rolling locomotion of a deformable tensegrity structure and first highlights the ability for tensegrities to utilize body deformation in order to augment gravitational and tensile potential energy towards achieving mobility. Interestingly, the authors uniquely discretize the rolling gait using the crystallography Miller Index to label the twenty faces of the six-bar spherical tensegrity's outer icosahedron structure. In these early results, they utilize a quasi-static approach to identify robot poses which promote discrete unstable transitions from one face to the next, where 'face' here refers to one of the 20 triangular shaped polygons which form the outer shell of the spherical tensegrity structure. They later extend this work towards a pneumatically-actuated tensegrity in [52]. Hirai et al. similarly utilize a quasi-static approach for a robot with pneumatic actuators replacing cables rather than rods in [36], controlling a fully-actuated robot with 24 pneumatic linear actuators. These works exemplify a common motif in spherical tensegrity mobility research that emphasizes the discretization of rolling gaits into finite states, such as face-to-face transitions which arise naturally due to the stable-unstable transitions inherent to the robot geometry.

In [49], Kim et al. utilize a similar finite state approach by identifying cable inputs which induce face-to-face transitions and combining them into long sequences for directional punctuated rolling. In this work, single-cable actuation (i.e., only one of the 24 cables is actuated at any given time) is utilized to great extent, demonstrating that complex punctuated rolling behavior is possible with only a subset of cables active at any given time. A similar approach is used in [24] to control a cable-driven rod-centered tensegrity design which houses actuators and electronics at the center of each rigid body. These works are later extended in [23], where uphill climbing on 24-degree inclines by an underactuated 6-cable spherical tensegrity is demonstrated using simultaneous actuation of only two cables at a time. Kim later demonstrates a Monte Carlo sampling-based approach using dynamic relaxation in [48] to find quasi-static poses of a rod-actuated tensegrity robot that promote face-to-face rolling.

Friesen et al. utilize a hand-engineered policy in combination with a *force density* method for inverse kinematics in [28] in order to achieve an inchworm-like motion for a tensegrity robot designed to traverse complex duct systems. Specifically, given a desired pose of the tensegrity robot which they design to achieve periodic crawling mobility, their approach solves for the required cable lengths by minimizing elastic potential energy in the tensegrity system.

Early examples of advanced higher level motion planning for tensegrities are primarily centered around evolutionary controllers and periodic gaits produced through central pattern generators. In [69, 68], the authors propose an approach for automatic design of controllers for locomotion of tensegrity prisms of three and four struts using genetic evolutionary algorithms. Here, Paul et al. forego precise trajectory tracking in favor of rhythmic periodic gaits which produce nonzero movement of the robot's center of mass.

In a similar manner, researchers at NASA applied a multi-agent evolutionary approach towards finding periodic gaits for six-bar spherical tensegrities [38]. Leveraging the unique distributed nature of tensegrities, [37] presents a robust, distributed, and directional rolling algorithm using coevolution approaches. Notably, the work utilizes the natural symmetry of tensegrity structures and is one of the first papers to demonstrate tensegrity rolling mobility on difficult terrain such as hills and inclines. Related work from the same lab, [100] uses a similar evolution strategy for distributed impedence controllers coupled with central pattern generators in order to control a spine-like tensegrity robot with rigid tetrahedron-shaped segments and actuated cable strings.

Lastly, deep reinforcement learning techniques have been used to great success for tensegrity motion planning in recent years. In [109], Zhang et. al utilize model-based reinforcement learning using Mirror Descent Guided Policy Search (MDGPS). Specifically, their approach learns local time-varying linear-Gaussian dynamics and neural network policies using sampled trajectories. This MDGPS approach trains a high-level global policy using supervised learning on local controllers and introduces some necessary techniques to promote stable periodic locomotion gaits, such as training local policies from many different initial conditions to combat compounding modelling errors due to contact-rich dynamics. Related work [61] builds on these results and demonstrates the reinforcement learning-based approach's ability to handle partial observability and limited sensory information.

4.4 Contributions of this Chapter

The results presented here highlight different optimization-based approaches which are capable of handling motion planning and trajectory optimization with complex nonlinear dynamics that are inherent to tensegrity robots. In particular, Model Predictive Control and iterative local quadratic methods are introduced in the context of rolling ground mobility with tensegrity robots. In contrast to sample-based methods presented in works such as [109][61], these model-based techniques circumvent the *curse of dimensionality* by foregoing sampling in high-dimensional subspaces and instead leverage the structured dynamics presented in the previous chapter to obtain sample-efficient locomotion policies.

In addition, the experimental results shown here help validate and evaluate spherical tensegrity robots as a viable mobility platform. Benchmark results compare how tensegrity locomotion performance may be impacted as varying mechanical parameters are modified (such as cable stiffness, pretension, mass distribution, etc.). Furthermore, this chapter demonstrates the efficacy of using these optimization-based approaches as an effective tool for designing radically different hardware topologies with varying hardware and degrees of control authority. Later sections present novel cable-driven actuation schema uniquely identified through the use of the numerical methods presented in this chapter.

4.5 Tensegrity Actuation and Hardware

To frame the discussion of control policies and motion planning approaches introduced in this chapter, we first outline the basic design principles of tensegrity hardware. Traditionally, tensegrity robots are cable-driven flexible robots with independent actuators such as brushed DC motors driving each individual elastic cable. Each compliant cable is a series-elastic element typically constructed using a high-strength stiff cable connected in series with a compliant extension or compression spring (see yellow cables in Figure 4.1).

For most hardware implementations, position control of the motor is carefully maintained to adjust the cable rest length and overall cable tension throughout the robot. Positional encoders are used in most of the designs constructed in the B.E.S.T. lab, and some tensegrity designs also feature motor current and tension sensing for individual cables, providing additional sensing for more precise control. Other hardware implementations feature a continuously compliant material such as Bungee cord or elastic tubing rather than a spring-cable assembly, providing the added benefit of greater system compliance at the cost of increased complexity due to nonlinear effects between the elastic cable and wrapping contact around the motor spindle. In all of the designs from the B.E.S.T. Lab and Squishy Robotics, the spring-cable assembly is used for modularity, ease of rapid prototyping, and quick identification of compromised spring parts.

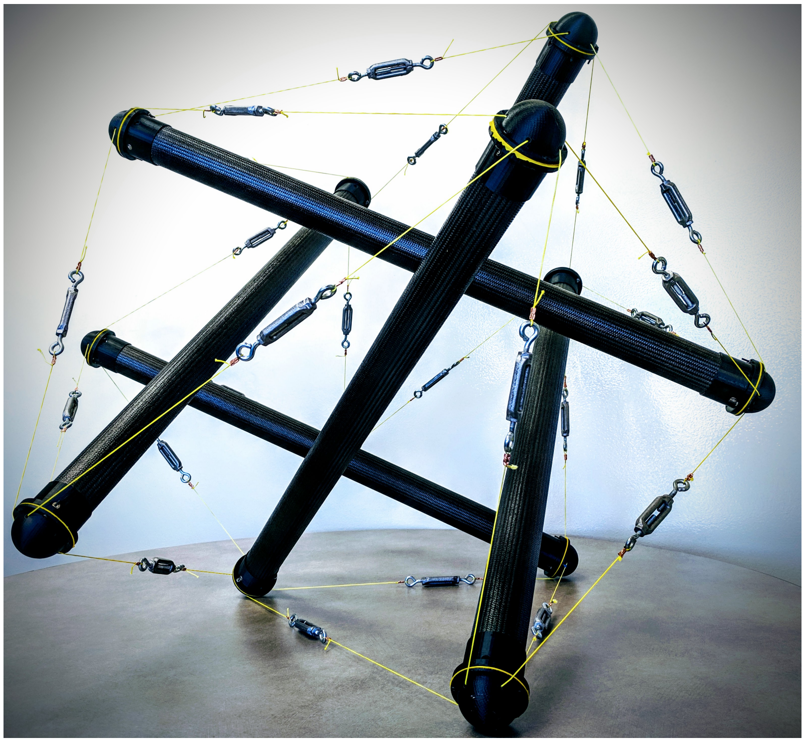


Figure 4.1: Example of six-bar spherical tensegrity robot from Squishy Robotics. Individual cables (in yellow) are controlled by adjusting their length using motors housed inside the carbon fiber rods.

4.5.1 Problem Formulation

The receding horizon control problem that we aim to solve at each timestep t is defined generally as the following minimization problem:

$$\min_{\mathbf{u}_{0\to N-1}} \sum_{k=0}^{N-1} \ell(\mathbf{x}_{k|t}, \mathbf{u}_{k|t}) + \ell_N(\mathbf{x}_{N|t}) \tag{4.1a}$$

$$s.t. \;\; \mathbf{x}_{k+1|t} = f(\mathbf{x}_{k|t}, \mathbf{u}_{k|t}) \tag{4.1b}$$

$$\mathbf{x}_{0|t} = \mathbf{x}_{init} \tag{4.1c}$$

$$\mathbf{x}_{k|t} \in \mathcal{X}, \quad \mathbf{u}_{k|t} \in \mathcal{U}, \quad \forall k \in \mathbb{N}^{N-1} \tag{4.1d}$$

where $\mathbf{x}_{k|t}$ and $\mathbf{u}_{k|t}$ are the state and input variables at time $t+k$, respectively, f are the problem-specific dynamics constraints, and $\mathcal{X}$ and $\mathcal{U}$ represent the feasible sets that satisfy state and input constraints.

The overall goal of trajectory optimization for mobile tensegrity robots is typically to maximize velocity while minimizing expended energy. The cost function for rolling locomotion typically rewards rigid body velocity or momentum that is aligned with a specified desired direction, with penalties for excessive robot deformation or motor speed. Depending on the techniques described below, the cost function we minimize may be a convex quadratic function, a quadratic approximation of a non-convex nonlinear problem, or it may be the true non-convex objective function itself.

The state of the system $\mathbf{x}_{k|t}$ is typically comprised of the positions/orientations of the rods and the cable rest lengths. The inputs to the system $\mathbf{u}_{k|t}$ are typically the rate of change of each cable's rest length.

4.6 Approach 1: Model Predictive Control

Model Predictive Control (MPC) is a control approach which iteratively solves a constrained optimization problem and implements only the first control input at the each timestep. This section provides an overview of MPC applied in the context of tensegrity rolling mobility, but for more in-depth surveys we refer the reader to [31, 74]. Some examples of MPC being applied successfully in robotics and autonomous systems can be found in [12, 26, 8, 85, 54]. In short, the primary benefit of this control scheme is the ability to leverage dynamic models to optimize future behavior over finite time-horizons while simultaneously complying with state and input constraints, such as those defined by realistic safety and actuator limitations. Additionally, because MPC is an iterative algorithm, the approach is inherently robust to unforeseen disturbances as they occur.

Model-based approaches such as MPC can be used to automatically design and evaluate tensegrity locomotion actuation policies (i.e., how to optimally actuate cable rest lengths). To form a tractable optimization problem, the continuous dynamics of the robot are first linearized about the robot's current state and discretized using a trapezoidal approximation:

$$\begin{bmatrix}\mathbf{p_{k+1}}\\ \dot{\mathbf{p}}_{\mathbf{k+1}}\end{bmatrix} = \begin{bmatrix}\mathbf{p_k}\\ \dot{\mathbf{p}}_{\mathbf{k}}\end{bmatrix} + \frac{\Delta t}{2}\begin{bmatrix}\dot{\mathbf{p}}_{\mathbf{k}} + \dot{\mathbf{p}}_{\mathbf{k+1}}\\ 2\ddot{\mathbf{p}}_{\mathbf{0}} + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}}\tilde{\mathbf{x}}_{\mathbf{k}} + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}}\tilde{\mathbf{x}}_{\mathbf{k+1}}\end{bmatrix} \tag{4.2}$$

where Δt is the simulation timestep, $\mathbf{x} \in \mathbb{R}^{96}$ is a concatenated vector of cable lengths and nodal position/velocity states, $\tilde{\mathbf{x}}_{\mathbf{k}}$ is the deviation about the linearization point $\mathbf{x_0}$, N is the number of nodal end points, and $\ddot{\mathbf{p}}_{\mathbf{0}} \in \mathbb{R}^{3N}$ is the current state acceleration at $\mathbf{x} = \mathbf{x_0}$, where $\ddot{\mathbf{p}}$ is calculated as follows:

$$\begin{aligned}\ddot{\mathbf{p}} = \mathbf{W}\Big(&- \mathbf{J}^T(\mathbf{J}\mathbf{W}\mathbf{J}^T)^{-1}(\dot{\mathbf{J}}\dot{\mathbf{p}} - k_s\mathbf{G} - k_d\dot{\mathbf{G}}) + \\ &(\mathbf{I} - \mathbf{J}^T(\mathbf{J}\mathbf{W}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{W})(\sum_i^J \boldsymbol{\gamma}_i + \mathbf{F}_{ext})\Big)\end{aligned} \tag{4.3}$$

Equation 5.16 is obtained by combining the results of Sections 3.6.1 and 3.6.2 from the previous chapter. Here, the vector $\mathbf{F}_{ext}$ contains the total forces applied to the tensegrity robot which are external to the system (e.g., ground contact reaction forces) and are calculated using the damped linear-elastic collisions described in Section 3.6.3.

Using the derived linearized and discretized dynamics as optimization constraints, we minimize the following cost function subject to the dynamics constraints above:

$$\min_{\mathbf{u}_{0\rightarrow N-1}} J(\mathbf{x}_0, \mathbf{u}_{0:N-1}) = \sum_{k=1}^{N} \boldsymbol{\psi}^{k-1} \left\{ -\theta_1 \sum_{i=1}^{N} \dot{\mathbf{p}}_i^T \mathbf{D}_{des} + \theta_2 \|\hat{\mathbf{L}}_{\mathbf{k}}\|_1 + \theta_3 \|\tilde{\mathbf{L}}_{\mathbf{k}}\|_1 \right\} \tag{4.4a}$$

$$s.t. \;\; \mathbf{x}_{k+1|t} = f(\mathbf{x}_{k|t}, \mathbf{u}_{k|t}) \tag{4.4b}$$

$$\mathbf{x}_{0|t} = \mathbf{x}_{init} \tag{4.4c}$$

$$\mathbf{x}_{k|t} \in \mathcal{X}, \quad \mathbf{u}_{k|t} \in \mathcal{U}, \quad \forall k \in \mathbb{N}^{N-1} \tag{4.4d}$$

$$\mathbf{x}_{N|t} \in \mathcal{X}_N, \quad \mathbf{u}_{N|t} \in \mathcal{U}_N \tag{4.4e}$$

Here $\mathbf{D}_{des} \in \mathbb{R}^3$ is the desired direction of travel, T is the MPC finite-time horizon, and $\boldsymbol{\psi} < 1$ is a discount factor placing less weight on later states to account for linearization errors. The optimization time horizon can range anywhere between 0.5 to 0.01 seconds, depending on desired accuracy and computation time, with longer time horizons being more susceptible to inaccuracies due to linearization approximations. Finally, $\theta_1, \theta_2, \theta_3$ are weighting parameters, and $\hat{\mathbf{L}}_{\mathbf{k}}$ and $\tilde{\mathbf{L}}_{\mathbf{k}} \in \mathbb{R}^J$ contain deviations of the k^{th}-step cable rest lengths about the neutral pretensioned lengths and initial lengths used for linearization, respectively. The cost function rewards rolling velocity in a desired direction while simultaneously penalizing cable rest length deviations from both initial pretensioned lengths and current rest lengths (i.e., $k = 0$). Combined, the additional penalty terms prevent the robot from excessive deformations and generate sparse motor actuation due to the Lasso-regularization.

The convex cost function above and linear equality and inequality constraints from the dynamics, state/actuator limits, and initial conditions thus form a linear program which is easily minimized using any convex optimization solver. For this work, Gurobi Optimizer and YALMIP [59] were used in MATLAB to solve the optimization problem at each timestep iteration.

4.6.1 Example MPC Results

Using the simulation parameters defined in Table 4.2, MPC was used to generate 150+ rollouts over the span of a few hours, utilizing full 24-cable actuation for dynamic rolling in varying directions sampled from a uniform distribution. For each trial, the robot was initialized with perturbed initial conditions of different robot orientations so that a wider variety of states were explored.

From the results, it is clear that the geometry and orientation of the topology plays a significant role in the trajectory tracking capabilities of the robot while rolling. Observe that the triangular faces of the icosahedron-shaped design require the robot to move in a zig-zag pattern (as described in [48]) even when moving in a straight line (see Fig. 4.2). However, whereas the path is well-defined in the case of single-cable actuation - where the robot must always return to its neutral pretensioned state before taking its next step - multi-cable

parameter	value	physical parameters	value
timesteps	10000	rod length	1.0 m
dt	0.01	rod mass (nominal)	1.1 kg
N	10	cable stiffness	515 N/m
γ	0.95	pretension	22 N
α	1.0	max motor velocity	0.02 m/s
β	3000	gravity	9.81 m/s^2

Table 4.1: Simulation parameters for MPC trial runs.

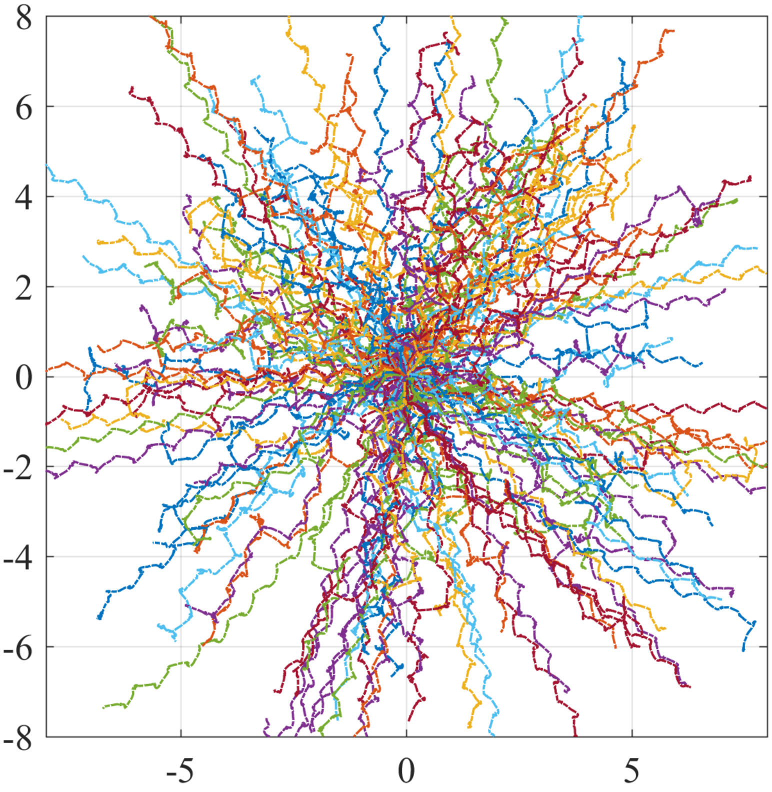
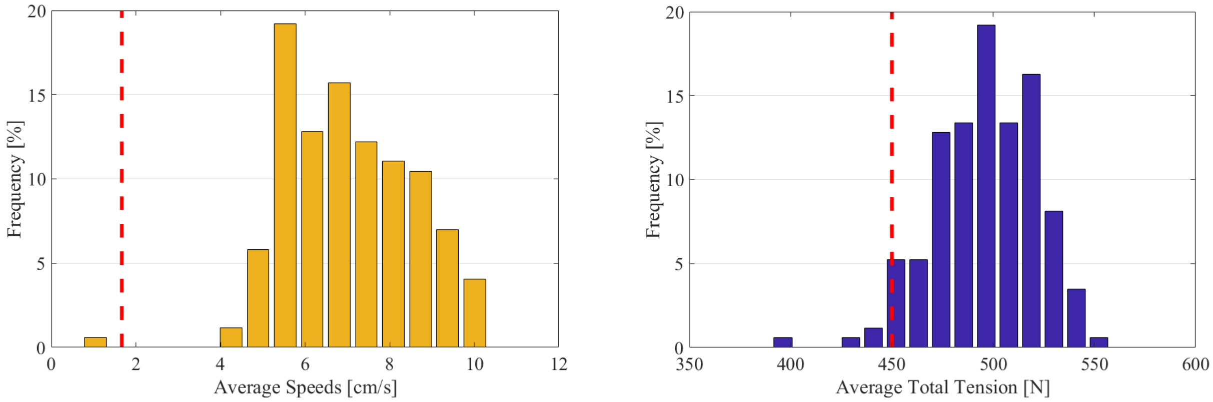


Figure 4.2: Overhead view of robot CoM during rolling trajectories generated using MPC. All trajectories begin at the origin and move radially outwards in uniformly random directions. Note the characteristic 'zig-zag' pattern for straight-line reference trajectories. Axis units are in meters.

actuation need not return to the neutral pose, allowing for increased speed and continuous rolling but making the robot's traversed path more complex and stochastic.

Interestingly, as the penalty weight for cable restlength deviation (θ_2 in Equation 4.4a) is relaxed – meaning the robot is rewarded less for keeping a spherical shape – the robot will locomote in a manner closer to amoeboid movement rather than pure rolling, with lower center of gravity and larger significant deformations, and tracking errors are reduced at the



(a) Distribution of average speeds for all MPC trial runs. Dotted red line represents average speed using single-cable actuation, obtained from prior hardware experiments.

(b) Distribution of average total tension in the system for all MPC trials. Dotted red line represents average total tension for single-cable policies, obtained through simulation.

Figure 4.3: Performance histograms compared to baseline single-cable actuation.

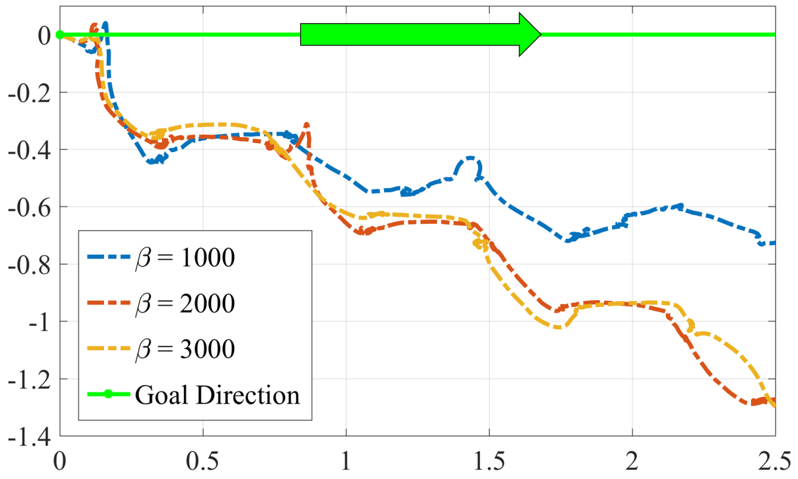


Figure 4.4: Zoomed-in comparison of trajectories with different cable restlength deviation penalty weights, β, in the worst-case direction due to robot geometry. Axis units are in meters.

cost of slower dynamic rolling speeds (see Fig. 4.4). This penalty term weight is thus one of many important hyperparameters for reward shaping and highlights the ability for this reward function to easily adjust the performance trade-offs of higher tracking accuracy versus faster speeds.

Although demonstrating improved speed, multi-cable actuation generated through MPC in this example is slightly outperformed by single-cable actuation (i.e., where only one cable can have a non-neutral rest length at a time) in terms of energy-efficiency. From Fig. 4.3, we can see that cable tensions (which are proportional to current-draw of the motors) are, on average, slightly higher when actuating all 24 cables. Nevertheless, for a 10% increase in average total tension, this multi-cable actuation policy provides approximately 400% increase in average speed. As well, the user may augment the MPC cost function to incorporate a reward for energy-efficiency, or any other prioritized metric, if desired.

4.7 Approach 2: Iterative Linear Quadratic Regulator

In the previous section, Model Predictive Control was introduced as a powerful approach which leveraged the structured dynamics of tensegrities for constrained model-based optimal control. In formulating the previous receding horizon control problem, we presented the optimization problem as a linear program (LP) using the approximated local linear dynamics. This approximation results in a tractable convex problem easily solved using off-the-shelf LP solvers.

The primary advantage of using the simplified linearized dynamics and linear cost is that the problem is computationally cost-effective and faster to solve compared to the true non-convex nonlinear minimization problem. The trade-off, however, is that the greedy approach over short horizons solves only for locally-optimal solutions in the linearized approximation, with longer time horizons giving increasingly inaccurate estimates of future states and stage costs. Notably, using open-loop state estimates to inform the next iteration's linearized dynamics was initially investigated but did not improve results in practice, partly due to inaccuracies resulting from the highly nonlinear discontinuous contact dynamics which are present while rolling. In this section, we present an approach which instead uses the true nonlinear dynamics to find a control input that achieves a local minimum cost using the well-known Iterative Linear Quadratic Regulator (iLQR) algorithm and then present some successful applications in tensegrity motion planning.

4.7.1 Differential Dynamic Programming and iLQR

From a high level, the approach of iterative linear quadratic regulation, originally introduced in [58], obtains locally-optimal solutions to the problem introduced in Section 4.5.1 by finding a sequence of proposal trajectories which incrementally improve the solution over several iterations. A simplified variant of *Differential Dynamic Programming* (DDP) [63], iLQR uses locally-linear dynamics in place of the quadratic local models found in DDP and applies a quasi-Newton method to quadratically converge to an optimal solution. In this form, the problem simplifies to a form identical to the Linear Quadratic Regulator problem, hence its familiar name. For completeness, this section outlines the DDP formulation originally presented in [63] before highlighting the key innovation of iLQR and its advantages.

Given a system with state $\mathbf{x}$ subject to the general nonlinear dynamics:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$$

DDP is a trajectory optimization algorithm that aims to find the optimal action sequence $\{\mathbf{u}_0, ..., \mathbf{u}_{N-1}\}$ that minimizes a total cost $J(\mathbf{x}_0, \mathbf{u}_{0\to N-1})$:

$$J^*(\mathbf{x}_0, \mathbf{u}_{0\to N-1}) = \min_{\mathbf{u}_{0\to N-1}} \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, \mathbf{u}_k) + \ell_N(\mathbf{x}_N)$$

where ℓ and ℓ_N are the running and terminal cost, respectively. If we define the cost-to-go $\mathbf{J}_i$ as the partial sum of costs from timesteps i to N:

$$J_i(\mathbf{x}_0, \mathbf{u}_{i\to N-1}) = \min_{\mathbf{u}_{i\to N-1}} \sum_{k=i}^{N-1} \ell(\mathbf{x}_k, \mathbf{u}_k) + \ell_N(\mathbf{x}_N)$$

then we can define the *value function* which represents the optimal cost-to-go given the minimizing control sequence:

$$V(\mathbf{x}, i) = J_i^*(\mathbf{x}_0, \mathbf{u}_{i\to N-1}) = \min_{\mathbf{u}_{i\to N-1}} J_i(\mathbf{x}_0, \mathbf{u}_{i\to N-1}) \tag{4.5}$$

Using the principle of dynamic programming and Bellman Optimality, we can reduce the problem of solving for the entire control sequence to obtaining the optimal input for just one timestep i:

$$V(\mathbf{x}, i) = \min_{\mathbf{u}} \big(\ell(\mathbf{x}, \mathbf{u}) + V(f(\mathbf{x}, \mathbf{u}), i+1)\big) \tag{4.6}$$

We define the function Q with dependent variables $\delta\mathbf{x}$, $\delta\mathbf{u}$ to represent the *variation* of the value about some nominal $\mathbf{x}$, $\mathbf{u}$ and form its second-order local approximation:

$$\begin{aligned} Q(\delta\mathbf{x}, \delta\mathbf{u}) \equiv{}& [\ell(\mathbf{x}+\delta\mathbf{x}, \mathbf{u}+\delta\mathbf{u}) - \ell(\mathbf{x}, \mathbf{u})] + \\ & [V(f(\mathbf{x}+\delta\mathbf{x}, \mathbf{u}+\delta\mathbf{u}), i+1) - V(f(\mathbf{x}, \mathbf{u}), i+1)] \qquad (4.7a) \\ ={}& [\ell(\mathbf{x}+\delta\mathbf{x}, \mathbf{u}+\delta\mathbf{u}) - \ell(\mathbf{x}, \mathbf{u})] + \\ & [V'(f(\mathbf{x}+\delta\mathbf{x}, \mathbf{u}+\delta\mathbf{u})) - V'(f(\mathbf{x}, \mathbf{u}))] \qquad (4.7b) \\ \approx{}& \frac{1}{2}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\\ 1\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q}_{xx} & \mathbf{Q}_{xu} & \mathbf{Q}_x\\ \mathbf{Q}_{ux} & \mathbf{Q}_{uu} & \mathbf{Q}_u\\ \mathbf{Q}_x^\top & \mathbf{Q}_u^\top & 0\end{bmatrix} \begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\\ 1\end{bmatrix} \qquad (4.7c) \end{aligned}$$

where the subscripts in the last line represent partial derivatives with respect to the state/input variables, as in [65], and are calculated as follows:

$$\mathbf{Q}_x = \boldsymbol{\ell}_x + \mathbf{f}_x^\top V_x' \tag{4.8a}$$

$$\mathbf{Q}_u = \boldsymbol{\ell}_u + \mathbf{f}_x^\top V_x' \tag{4.8b}$$

$$\mathbf{Q}_{xx} = \boldsymbol{\ell}_{xx} + \mathbf{f}_x^\top V_{xx}' \mathbf{f}_x + V_x' \mathbf{f}_{xx} \tag{4.8c}$$

$$\mathbf{Q}_{uu} = \boldsymbol{\ell}_{uu} + \mathbf{f}_u^\top V_{xx}' \mathbf{f}_u + V_x' \mathbf{f}_{uu} \tag{4.8d}$$

$$\mathbf{Q}_{ux} = \boldsymbol{\ell}_{ux} + \mathbf{f}_u^\top V_{xx}' \mathbf{f}_x + V_x' \mathbf{f}_{ux} \tag{4.8e}$$

$\mathbf{f}_{[\,\cdot\,]}$ represents the gradient and Jacobian partial derivatives of the dynamics taken with respect to the state/input. Similarly, V_x and V_{xx} represent the Jacobian and Hessian of the value function, respectively. With the assumption that the second-order approximation $Q(\delta\mathbf{x}, \delta\mathbf{u})$ is positive definite and convex, the unique optimal input perturbation $\delta\mathbf{u}^*$ which minimizes the local cost is obtained by setting the gradient equal to zero:

$$\nabla_{\partial\mathbf{u}} Q(\partial\mathbf{x}, \partial\mathbf{u}) = \mathbf{Q}_{ux}\partial\mathbf{x} + \mathbf{Q}_{uu}\partial\mathbf{u} + \mathbf{Q}_u \tag{4.9}$$

$$\Rightarrow \delta\mathbf{u}^* = \underset{\delta\mathbf{u}}{\operatorname{argmin}}\ Q(\delta\mathbf{x}, \delta\mathbf{u}) = -\mathbf{Q}_{uu}^{-1}(\mathbf{Q}_{ux}\delta\mathbf{x} + \mathbf{Q}_u) = -\mathbf{K}\partial\mathbf{x} + \mathbf{k} \tag{4.10}$$

Substituting the optimal input back to the original quadratic approximation, the recursive quadratic model of value functions at each time step can be computed:

$$\Delta V(i) = -\frac{1}{2}\mathbf{Q}_u^\top \mathbf{Q}_{uu}^{-1}\mathbf{Q}_u \tag{4.11}$$

$$V_x(i) = \mathbf{Q}_x - \mathbf{Q}_{xu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_u \tag{4.12}$$

$$V_{xx}(i) = \mathbf{Q}_{xx} - \mathbf{Q}_{xu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ux} \tag{4.13}$$

Completing a backwards pass, we calculate the affine offset and feedback gains at each timestep, $\mathbf{k}_i$ and $\mathbf{K}_i$, to iteratively compute new trajectories until convergence (i.e., negligible improvements in value function):

$$\begin{aligned}\hat{\mathbf{x}}_0 &= \mathbf{x}_0\\ \hat{\mathbf{u}}_i &= \mathbf{u}_i + \mathbf{k}_i + \mathbf{K}_i(\hat{\mathbf{x}}_i - \mathbf{x}_i)\\ \hat{\mathbf{x}}_{i+1} &= f(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i)\end{aligned}$$

Note that this backwards propagation procedure and resulting feedback policy is similar to concepts for the Linear Quadratic Regulator and is in fact *exactly* identical when second-order effects of the dynamics are neglected:

$$\mathbf{Q}_{xx} = \boldsymbol{\ell}_{xx} + \mathbf{f}_x^\top V'_{xx}\mathbf{f}_x + V'_x\mathbf{f}_{xx} \Rightarrow \mathbf{Q}_{xx} \approx \boldsymbol{\ell}_{xx} + \mathbf{f}_x^\top V'_{xx}\mathbf{f}_x \tag{4.14}$$

$$\mathbf{Q}_{uu} = \boldsymbol{\ell}_{uu} + \mathbf{f}_u^\top V'_{uu}\mathbf{f}_u + V'_u\mathbf{f}_{uu} \Rightarrow \mathbf{Q}_{uu} \approx \boldsymbol{\ell}_{uu} + \mathbf{f}_u^\top V'_{uu}\mathbf{f}_u \tag{4.15}$$

$$\mathbf{Q}_{ux} = \boldsymbol{\ell}_{ux} + \mathbf{f}_u^\top V'_{ux}\mathbf{f}_x + V'_x\mathbf{f}_{ux} \Rightarrow \mathbf{Q}_{ux} \approx \boldsymbol{\ell}_{ux} + \mathbf{f}_u^\top V'_{ux}\mathbf{f}_x \tag{4.16}$$

The critical insight that utilizing only linear dynamics information leads to faster convergence (wall-clock time) without needing to calculate the expensive Hessian of the dynamics is the key innovation of iLQR [58]. For receding horizon control methods where optimization is repeatedly solved at each time step, speed is often more important than fine improvements in precision. In this simplified formulation, the local quadratic approximation and linearized dynamics are used to iteratively apply the LQR algorithm and is guaranteed to converge to a local optima of the true non-convex nonlinear cost function subject to nonlinear dynamic constraints.

4.7.2 Tensegrity Problem Formulation with iLQR

The receding horizon control problem that iLQR solves at each timestep t is defined as

$$\min_{\mathbf{u}_{0\rightarrow N-1}} \sum_{k=0}^{N-1} \ell(\mathbf{x}_{k|t}, \mathbf{u}_{k|t}) + \ell_N(\mathbf{x}_{N|t}) \tag{4.17a}$$

$$s.t. \;\; \mathbf{x}_{k+1|t} = f(\mathbf{x}_{k|t}, \mathbf{u}_{k|t}) \tag{4.17b}$$

$$\mathbf{x}_{0|t} = \mathbf{x}_{init} \tag{4.17c}$$

$$\mathbf{x}_{k|t} \in \mathcal{X}, \quad \mathbf{u}_{k|t} \in \mathcal{U}, \quad \forall k \in \mathbb{N}^{N-1} \tag{4.17d}$$

$$\mathbf{x}_{N|t} \in \mathcal{X}_N, \quad \mathbf{u}_{N|t} \in \mathcal{U}_N \tag{4.17e}$$

where $\mathbf{x}_{k|t}$ and $\mathbf{u}_{k|t}$ are the state and input variables at time $t+k$, respectively, and $\mathcal{X}$ and $\mathcal{U}$ represent the feasible sets that satisfy state and input constraints.

The cost function we aim to optimize for the spherical tensegrity is a quadratic function which maximizes linear velocity of the system in the desired direction and minimizes large control inputs (which correlates with motor torque) and dynamic state deviations from the *neutral pose* (i.e., equal rest length for all cables) to prevent excessively large pose deformations which could be detrimental to the long-term continuous rolling gait of the robot:

$$J(\mathbf{x}_{aug,0}, \mathbf{u}_{0:T}) = \sum_{k=0}^{T} \mathbf{x}_{aug,k}^T \mathbf{Q}_{aug} \mathbf{x}_{aug,k} + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \tag{4.18}$$

$$\mathbf{x}_{aug,k} \equiv \left[\begin{array}{c} (\mathbf{x}_k - \mathbf{x}_{ref}) \\ \hline 1 \end{array}\right] = \left[\begin{array}{c} \mathbf{p}_k - \mathbf{p}_{ref} \\ \dot{\mathbf{p}}_k - \dot{\mathbf{p}}_{ref} \\ \boldsymbol{\ell}_{cables,k} - \boldsymbol{\ell}_{cables,ref} \\ \boldsymbol{\ell}_{rods,k} - \boldsymbol{\ell}_{rods,ref} \\ \hline 1 \end{array}\right] \tag{4.19}$$

$$\mathbf{u}_k \equiv \begin{bmatrix} \dot{\boldsymbol{\ell}}_{cables,k} \\ \dot{\boldsymbol{\ell}}_{rods,k} \end{bmatrix} \tag{4.20}$$

$$\mathbf{Q}_{aug} = \left[\begin{array}{c|c} \mathbf{Q} & -\mathbf{d}_{des} \\ \hline -\mathbf{d}_{des}{}^T & 0 \end{array}\right] \qquad \mathbf{Q} = \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & w_1\mathbf{I}_{nc} & \\ & & & w_2\mathbf{I}_{nr} \end{bmatrix} \qquad \mathbf{R} = \left[\begin{array}{c|c} w_3\mathbf{I}_{nc} & 0 \\ \hline 0 & w_4\mathbf{I}_{nr} \end{array}\right]$$

where $\mathbf{d}_{des} \in \mathbb{R}^{36}$ is a vector representing the desired rolling direction in the nodal position $\mathbf{p}$ state-frame, nr and nc are the number of rods and cables, and w_1, w_2, w_3 and w_4 are weight parameters that trade off penalties for cable/rod length deviations from the neutral position and cable/rod velocity magnitudes, respectively. To find the optimal perturbations of the input according to iLQR, we define the necessary matrices about the nominal trajectory using the linearized and discretized dynamics:

$$\mathbf{f}_{x,k} = \left[\begin{array}{c|c} (\mathbf{I} + \Delta t \bar{\mathbf{A}}(\mathbf{x}_k, \mathbf{u}_k)) & 0 \\ \hline 0 & 1 \end{array}\right] \qquad \mathbf{f}_{u,k} = \left[\begin{array}{c} \Delta t \bar{\mathbf{B}}(\mathbf{x}_k, \mathbf{u}_k) \\ \hline 0 \end{array}\right]$$

$$\bar{\mathbf{A}}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 \\ \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \mathbf{p}} & \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \dot{\mathbf{p}}} & \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \boldsymbol{\ell}_{cables}} & \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \boldsymbol{\ell}_{rods}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}_k} \qquad \bar{\mathbf{B}}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \mathbf{I}_{nc} & 0 \\ 0 & \mathbf{I}_{nr} \end{bmatrix} \tag{4.21}$$

where $\ddot{\mathbf{p}}(\mathbf{x})$ is calculated using the dynamics equations described in Section 3.6.4 and $\bar{\mathbf{A}}(\mathbf{x}_k, \mathbf{u}_k)$ and $\bar{\mathbf{B}}(\mathbf{x}_k, \mathbf{u}_k)$ are matrices with a block structure that coincide with the block structure of $\mathbf{x}_{aug}$.

Due to the quadratic objective and linear approximation of the dynamic constraints, we easily can solve the approximate sub-problems using LQR techniques. Importantly, we obtain locally-optimal feedback *policies* rather than open-loop inputs, which may be useful if the specific application requires optimal input over multiple timesteps (e.g., if computation time is too slow or expensive to re-solve) and is thus more robust to external disturbances. With the necessary items defined, iterative Linear Quadratic Regulation is applied to optimize actions over a predefined time horizon typically ranging between 0.5 to 0.01 seconds using the steps outlined in Algorithm 1.

Algorithm 1: Iterative Linear Quadratic Regulator

```
Input: General penalty function $\ell_t(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t)$
       Dynamics function $f(\mathbf{x}_t, \mathbf{u}_t)$
       Trajectory horizon $T$
       Initial state $\mathbf{x}_{init}$
       Initial control sequence $\hat{\mathbf{U}} = \{\hat{\mathbf{u}}_0, ..., \hat{\mathbf{u}}_{T-1}\}$
Output: Control sequence $\mathbf{U} = \{\mathbf{u}_0, ..., \mathbf{u}_{T-1}\}$
        Feedback terms $\mathbf{K}_0, ..., \mathbf{K}_{T-1}$
Obtain $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_T\}$ by evolving according to nonlinear dynamics:
 $\hat{\mathbf{x}}_{t+1} = f(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t)$ with initial condition: $\hat{\mathbf{x}}_0 = \mathbf{x}_{init}$
while Not Converged do
    $\delta_{\mathbf{x}_t} = \mathbf{x}_t - \hat{\mathbf{x}}_t,\ \delta_{\mathbf{u}_t} = \mathbf{u}_t - \hat{\mathbf{u}}_t$
    $\mathbf{A}_t = \frac{\partial f}{\partial \mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t),\ \mathbf{B}_t = \frac{\partial f}{\partial \mathbf{u}} f(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t)$
    $\mathbf{Q}_t = \frac{\partial^2 f}{\partial \mathbf{x}^2} \ell_t(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t),\ \mathbf{R}_t = \frac{\partial^2 f}{\partial \mathbf{u}^2} \ell_t(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t)$
    $\mathbf{q}_t = \mathbf{Q}\hat{\mathbf{x}}_t,\ \mathbf{r}_t = \mathbf{R}\hat{\mathbf{u}}_t$
    Solve for $\mathbf{K}_0, ..., \mathbf{K}_{T-1}$, $\mathbf{k}_0, ..., \mathbf{k}_{T-1}$ according to LQR($\mathbf{Q}, \mathbf{R}, \mathbf{q}, \mathbf{r}, \mathbf{A}, \mathbf{B}$)
    while cost is not improving do
        /* backtracking line search                                    */
        Obtain sequence $\mathbf{U}$ by propagating $\mathbf{u} = \hat{\mathbf{u}}_t + \mathbf{K}_t \delta_{\mathbf{x}_t} + \alpha \mathbf{k}_t$, according to
         $\hat{\mathbf{x}}_{t+1} = f(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t)$ and calculating $\delta_{\mathbf{x}_t}$ along the way
        Set $\hat{\mathbf{U}} = \mathbf{U}$
        Calculate new cost function using $\hat{\mathbf{X}}, \hat{\mathbf{U}}$
        $\alpha \leftarrow \alpha/2$
    end
end
```

4.7.3 Input Change Penalization

In practice, large changes in motor current are undesirable due to inductive loads to the motor. To ensure that cable velocity set points are relatively smooth over time and thus minimize large changes in motor current, a modified version of the original problem statement above is used instead, where the *rate of change* of the input is controlled rather than the input itself (i.e., the cable accelerations rather than cable rest length velocities). With this minor modification, the input to the system is included in the overall state of the problem and cable *acceleration* is the new input of interest, which we are now able to easily penalize. The complete modified Linear Quadratic Regulator problem for tensegrity rolling mobility used in practice is thus described using the following equations:

$$J(\mathbf{x}_{aug,0}, \Delta\mathbf{u}_{0:T}) = \sum_{k=0}^{T} \mathbf{z}_k^T \mathbf{Q}'_k \mathbf{z}_k + (\boldsymbol{\Delta}\mathbf{u}_k)^T \mathbf{R}'_k (\boldsymbol{\Delta}\mathbf{u}_k) \tag{4.22}$$

$$\mathbf{z}_k \equiv \left[\begin{array}{c} (\mathbf{x}_k - \mathbf{x}_{ref}) \\ \hline 1 \\ \hline \mathbf{u}_k \end{array}\right] = \left[\begin{array}{c} \mathbf{p}_k - \mathbf{p}_{ref} \\ \dot{\mathbf{p}}_k - \dot{\mathbf{p}}_{ref} \\ \boldsymbol{\ell}_{cables,k} - \boldsymbol{\ell}_{cables,ref} \\ \boldsymbol{\ell}_{rods,k} - \boldsymbol{\ell}_{rods,ref} \\ \hline 1 \\ \hline \dot{\boldsymbol{\ell}}_{cables,k} \\ \dot{\boldsymbol{\ell}}_{rods,k} \end{array}\right] \qquad \Delta\mathbf{u}_k \equiv \begin{bmatrix} \Delta\dot{\boldsymbol{\ell}}_{cables,k} \\ \Delta\dot{\boldsymbol{\ell}}_{rods,k} \end{bmatrix} \tag{4.23}$$

$$\mathbf{Q}' = \left[\begin{array}{c|c} \mathbf{Q}_{aug} & 0 \\ \hline 0 & \mathbf{R} \end{array}\right] \qquad \mathbf{R}' = \left[\begin{array}{c|c} w_1\mathbf{I}_{nc} & 0 \\ \hline 0 & w_2\mathbf{I}_{nr} \end{array}\right] \tag{4.24}$$

where w_1 and w_2 are new weighting constants which penalize change in cable inputs and change in rod inputs, respectively. The modified linearized dynamics with change of inputs are represented with the following augmented matrices:

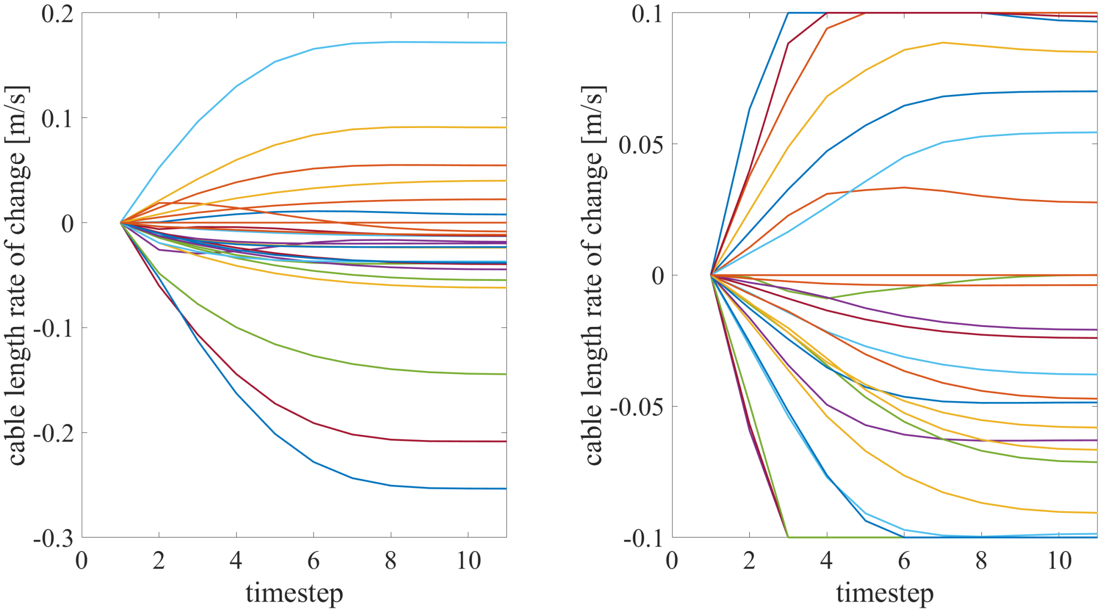
$$\mathbf{f}_{x,k} = \left[\begin{array}{c|c|c} (\mathbf{I} + \Delta t\bar{\mathbf{A}}(\mathbf{x}_k, \mathbf{u}_k)) & 0 & \Delta t\bar{\mathbf{B}}(\mathbf{x}_k, \mathbf{u}_k) \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & \mathbf{I}_{nc+nr} \end{array}\right] \qquad \mathbf{f}_{u,k} = \left[\begin{array}{c} \Delta t\bar{\mathbf{B}}(\mathbf{x}_k, \mathbf{u}_k) \\ \hline 0 \\ \hline \mathbf{I}_{nc+nr} \end{array}\right]$$

where, as before, the linearized particle dynamics are obtained according to the equations presented in Section 3.6:

$$\bar{\mathbf{A}}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 \\ \frac{\partial\ddot{\mathbf{p}}(\mathbf{x})}{\partial\mathbf{p}} & \frac{\partial\ddot{\mathbf{p}}(\mathbf{x})}{\partial\dot{\mathbf{p}}} & \frac{\partial\ddot{\mathbf{p}}(\mathbf{x})}{\partial\boldsymbol{\ell}_{cables}} & \frac{\partial\ddot{\mathbf{p}}(\mathbf{x})}{\partial\boldsymbol{\ell}_{rods}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}_k} \qquad \bar{\mathbf{B}}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \mathbf{I}_{nc} & 0 \\ 0 & \mathbf{I}_{nr} \end{bmatrix} \tag{4.25}$$

4.7.4 Handling State and Input Box Constraints

One advantage of using MPC, as outlined in approach 1, is that not only are dynamic constraints observed but also state and input constraints are locally satisfied as well. Addition-



(a) Optimal cable actuation over a 10-step horizon with unconstrained inputs.

(b) Optimal cable actuation over a 10-step horizon with constrained inputs bounded between -10 and 10 cm/s.

Figure 4.5: Comparison of iLQR solutions for optimal cable actuation without (left) and with (right) input box constraints, given identical robot state. Independent cables are indicated by a different color; 24 cables total.

ally, if the terminal set and cost function are properly defined, we can guarantee persistent feasibility and stability throughout the process [18][40]. In contrast, LQR and iLQR, in general, cannot handle arbitrary nonlinear state or input constraints. Fortunately, the relevant state/input box constraints we are typically concerned with for tensegrity systems (that of maximum/minimum cable rest lengths and cable velocities) are easily accounted for in the above formulation.

In practice, we adopt a similar approach as in [101], where we identify active constraints by observing where optimal perturbations push inputs outside of the constraint boundaries. In the cases where the time-varying affine term $\mathbf{k}(k)$ from Equation 4.10 forces control inputs outside of the designated safe set, the gain matrix $\mathbf{K}(k)$ is modified such that the rows of the feedback matrix corresponding to the inputs which have reached their limits are set to zero.

As the cable dynamics are simply represented by an integrator, we can extend this method to apply to cables which have also reached their state limits, by identifying which cables have reached their maximal extension or retraction rest length. In such cases, the state box constraints are also handled by clipping any cable velocity inputs which may violate the active set of constraints.

parameter	value	physical parameters	value
timesteps	5000	rod length	60 cm
dt	0.005	rod mass (nominal)	600 g
N	10	cable stiffness	800 N/m
		pretension	30 N
		max motor velocity	0.10 m/s
		gravity	9.81 m/s^2

Table 4.2: Simulation parameters for iLQR trial runs.

4.7.5 Cable and Rod Constraint Remapping Matrices

Another set of constraints that are useful to define are any restrictions or couplings between individual cable length (or rod length) actuators. Examples of this may be observed in cases where a single actuator can be used to control multiple cables in the structure and may be motivated by a desire to reduce complexity and/or redundancy in the tensegrity robot in order to save on cost and weight.

One way to achieve this for iLQR with linearly-related input couplings is to use a reduced-order input to the model that is linearly transformed through a *cable remapping* matrix. This simple modification results in a dynamic model with reduced degrees of freedom in the input which is then extrapolated to all 24 cables of the spherical tensegrity system. Without making major changes to the equations presented in earlier chapters, we are able to control the same dynamical system with 24 cables using only 6 or 12 input variables and reduced control authority. For further details, refer to Section 4.9.3 where remapping was used to great effect in implementing a novel paired-cable actuation schema.

4.8 Nonlinear Trajectory-Tracking with LQR

Feedback control of these systems can be achieved using traditional Linear Quadratic Regulator (LQR) control for trajectory tracking of the nonlinear system, linearized at each timestep about the current state. The reference trajectory $\mathcal{T} = \{\bar{\mathbf{x}}_0, \bar{\mathbf{u}}_0, ...\bar{\mathbf{x}}_{N-1}, \bar{\mathbf{u}}_{N-1}, \bar{\mathbf{x}}_N\}$ we would like to track can be obtained using one of the approaches described above. Alternatively, a greedy approach which uses the linearized dynamics of the initial state and maximizes local behavior over a short time-horizon can be used but often performs poorly due to the discontinuous nonlinear dynamics. Despite this, for simple tasks this approach will sometimes lead to useful albeit sub-optimal policies that are easy and quick to calculate online for real-time feedback control. In a similar manner as previously presented in Sec. 3.7, we can approximate the nonlinear dynamics using a first-order linear approximation:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{p}_{k+1} \\ \dot{\mathbf{p}}_{k+1} \\ \boldsymbol{\ell}_{cables,k+1} \\ \boldsymbol{\ell}_{rods,k+1} \end{bmatrix} \approx \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cables,k} \\ \boldsymbol{\ell}_{rods,k} \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\mathbf{p}}_{k} \\ \ddot{\mathbf{p}}_{k} \\ \dot{\boldsymbol{\ell}}_{cables,k} \\ \dot{\boldsymbol{\ell}}_{rods,k} \end{bmatrix} \tag{4.26}$$

$$\approx \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cables,k} \\ \boldsymbol{\ell}_{rods,k} \end{bmatrix} + \Delta t \Bigg(\begin{bmatrix} \bar{\dot{\mathbf{p}}}_{k} \\ \bar{\ddot{\mathbf{p}}}_{k} \\ \bar{\dot{\boldsymbol{\ell}}}_{cables,k} \\ \bar{\dot{\boldsymbol{\ell}}}_{rods,k} \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 \\ \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \mathbf{p}} & \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \dot{\mathbf{p}}} & \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \boldsymbol{\ell}_{cables}} & \frac{\partial \ddot{\mathbf{p}}(\mathbf{x})}{\partial \boldsymbol{\ell}_{rods}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Bigg|_{\mathbf{x}=\overline{\mathbf{x}}_k} \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\dot{\mathbf{p}}} \\ \tilde{\boldsymbol{\ell}}_{cables} \\ \tilde{\boldsymbol{\ell}}_{rods} \end{bmatrix} \tag{4.27}$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\dot{\boldsymbol{\ell}}}_{cables} \\ \tilde{\dot{\boldsymbol{\ell}}}_{rods} \end{bmatrix} \Bigg)$$

$$= \Big(\mathbf{I} + \Delta t \bar{\mathbf{A}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) \Big) \mathbf{x}_k + \Delta t \ \bar{\mathbf{B}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) \mathbf{u}_k \tag{4.28}$$

$$+ \Delta t \Bigg(\begin{bmatrix} \bar{\dot{\mathbf{p}}}_{k} \\ \bar{\ddot{\mathbf{p}}}_{k} \\ \bar{\dot{\boldsymbol{\ell}}}_{cables,k} \\ \bar{\dot{\boldsymbol{\ell}}}_{rods,k} \end{bmatrix} - \bar{\mathbf{A}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) \overline{\mathbf{x}}_k - \bar{\mathbf{B}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) \overline{\mathbf{u}}_k \Bigg)$$

In the equations above, $\bar{\mathbf{A}} \in \mathbb{R}^{n \times n}$ and $\bar{\mathbf{B}} \in \mathbb{R}^{n \times m}$ are the Jacobians of the nonlinear dynamics with respect to the state and input, respectively, linearized about the current state $\mathbf{x}_0$. While the example matrices above are independent of the current input, the dynamics may generally depend on either, both, or neither of the current state and/or input. The state and input variables, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$, contain the nodal positions and velocities and cable/rod length and rate of change (the latter, as input to the system). In terms of the described input and an augmented state variable, the general quadratic penalty to promote dynamic rolling motion is defined as follows:

$$J(\mathbf{x}_{aug,0}, \mathbf{u}_{0:T}) = p(\mathbf{x}_N) + \sum_{k=0}^{N-1} q(\mathbf{x}_k, \mathbf{u}_{\mathbf{k}}) \tag{4.29}$$

$$= \sum_{k=0}^{N} \mathbf{x}_{aug,k}^T \mathbf{Q}_{aug} \mathbf{x}_{aug,k} + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k + 2\mathbf{x}_{aug,k}^T \mathbf{N} \mathbf{u}_k \tag{4.30}$$

$$\mathbf{x}_{aug,k} = \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cables,k} \\ \boldsymbol{\ell}_{rods,k} \\ 1 \end{bmatrix} \qquad \mathbf{u}_{\mathbf{k}} = \begin{bmatrix} \dot{\boldsymbol{\ell}}_{cables,k} \\ \dot{\boldsymbol{\ell}}_{rods,k} \end{bmatrix} \tag{4.31}$$

where N is the time horizon and $\mathbf{x}_{k+1}$ is obtained by applying the system model forward dynamics.

In practice, cross-terms between state and inputs are neglected, and the reference state is incorporated to be able to penalize deviations from the desired nominal state trajectory of the robot, $\overline{\mathbf{x}}$. The overall optimal cost and associated terms are thus written as:

$$\min_{\mathbf{u}_{0:T}} J^*_{0\to N}(\mathbf{x}_{aug,0}, \mathbf{u}_{0:T}) = \sum_{k=0}^{N} \mathbf{x}^T_{aug,k} \mathbf{Q}_{aug} \mathbf{x}_{aug,k} + \mathbf{u}^T_k \mathbf{R} \mathbf{u}_k \tag{4.32}$$

$$\tag{4.33}$$

where:

$$\mathbf{x}_{aug,k+1} \equiv \left[\begin{array}{c} (\mathbf{x}_{k+1} - \overline{\mathbf{x}}_{k+1}) \\ \hline 1 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{I} + \Delta t \bar{\mathbf{A}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) & \mathbf{z}_k \\ \hline 0 & 1 \end{array}\right] \mathbf{x}_{aug,k} + \left[\begin{array}{c} \Delta t \bar{\mathbf{B}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) \\ \hline 0 \end{array}\right] \mathbf{u}_k$$

$$= \mathbf{A}' \mathbf{x}_{aug,k} + \mathbf{B}' \mathbf{u}_k \tag{4.34}$$

$$\mathbf{z}_k := \Delta t \left(\begin{bmatrix} \overline{\dot{\mathbf{p}}}_k \\ \overline{\ddot{\mathbf{p}}}_k \\ \overline{\dot{\boldsymbol{\ell}}}_{cables,k} \\ \overline{\dot{\boldsymbol{\ell}}}_{rods,k} \end{bmatrix} - \bar{\mathbf{B}}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k) \overline{\mathbf{u}}_k \right) - \left(\overline{\mathbf{x}}_{k+1} - \overline{\mathbf{x}}_k \right)$$

$\mathbf{Q}_{aug}$ and $\mathbf{R}$ are positive-definite matrices which penalize state (minus the uncontrolled augmented constant) and input deviations from the nominal trajectory, respectively.

4.9 Novel Tensegrity Actuation Schema

The model-based approaches outlined in this chapter enable rapid prototyping of different experimental controllers and actuation schema in simulation. In particular, varying degrees of control authority, number of actuators, and different types of actuation expand the capabilities of tensegrities beyond what traditional six-bar spherical tensegrities have done in existing hardware.

In the next sections, we present some interesting novel prototypes which were explored and evaluated in simulation through the use of the numerical approaches previously described. Driven by practical needs such as reducing the number of actuators, minimizing robot weight, or experimenting with rod-driven actuation rather than cable-driven actuation for ease of assembly, these experimental actuation schema explore different approaches we may pursue to push the current boundaries of tensegrity robotics.

4.9.1 Rod-Actuated Tensegrity Robots

One of the primary challenges of working with tensegrity robot hardware is the complexity and redundancy introduced through the numerous cable actuators. Cable-driven robotics, in general, is challenging due to issues regarding meticulous assembly, intricate mechanisms, cable tangling, cable slackness, etc. that are inherent to their design. In the fully-actuated nominal six-bar spherical tensegrity design specifically, individual motor actuators are needed to control all 24 cables, a complex mechanical assembly that can quickly get out of hand. Due to this, we wanted to explore using rod-actuated spherical tensegrities, reducing the number of required actuators from 24 down to 6, with passive elastic cables and controllable rod lengths. Motivated from previous works which used actuated rigid bodies such as [52, 48], the focus of this work is to identify if continuous dynamic rolling locomotion is attainable using solely rod actuation.

To do so, the same dynamics equations and motion planning approaches introduced in the previous sections are used, with minor modifications relaxing constraint boundaries on the rods to allow them to extend/retract and restricting the cables to be passive (i.e., no possibility for cable rest length change). Prior research work for six-bar tensegrity rolling mobility has often highlighted a finite state approach with discrete steps or state transitions; while dynamic locomotion with a continuous rolling gait for cable-driven tensegrities has been an open question, dynamic rod-actuated ground rolling has been explored even less.

Simulation results demonstrate the feasibility of using a rod actuation policy to produce fast rolling mobility with a maximum 10 cm/s linear rate of change for individual rod lengths. The rod travel distance was bounded between 0.75 m to 1.15 cm for a spherical tensegrity with nominal 1 m rod length at neutral position. It should be noted that while this approach is interesting and inspires future actuation schema, current existing off-the-shelf hardware such as solenoids or linear actuators capable of achieving both the necessary speeds and travel distances for fast dynamic rolling tend to be heavy and bulky, limiting practical application due to current hardware constraints.

4.9.2 Center-Payload Spherical Tensegrity

One exciting approach to simplify the six-bar spherical tensegrity design is to use a single centralized mechatronic payload of electronics which includes all of the actuation, sensors, and on-board compute in one compact package. As opposed to the current paradigm of using six distributed rods with independent microcontrollers which work in tandem through wireless communication, this center-payload spherical tensegrity topology seeks to reduce redundancy, weight, and costs while still having the capability to roll in any direction.

The center-payload variant of the six-bar spherical tensegrity (see Figure 4.6) houses all active components of the robot in a spherical payload which is suspended in the interior space of the robot using 12 inner-cables. In a similar manner of operation, this variant manipulates the controllable tensions of the 12 inner-cables in order to move the payload within the interior space of the outer shell, changing the robot's center of mass to cause an unstable transition from one face to another. The simplified design of this center-payload topology reduces redundancy in the system, minimizing the number of necessary actuators from 24 down to 12 or even 6 motors, and the robot prototype shown in the figure has an

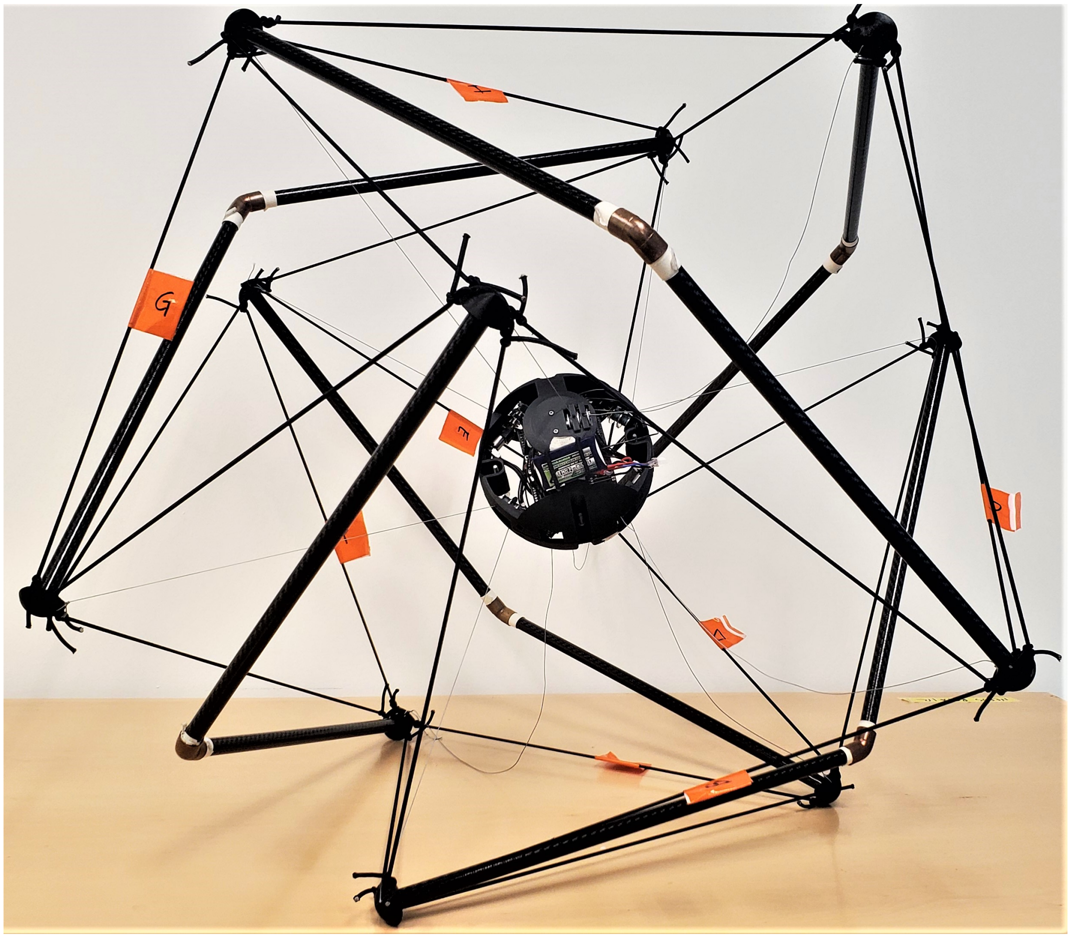


Figure 4.6: Proof-of-concept prototype of a center-payload spherical tensegrity robot, image courtesy of Squishy Robotics. While the external cables are passive and unactuated, the 12 inner cables are controllable, allowing the center payload, which houses all of the electronics and motors, to move freely within the internal volume of the robot's external convex hull.

approximately 70% reduced weight of 1.5 kg compared to the current six-bar robot's weight of 5.2 kg.

While seemingly intuitive, early work [1] had presented numerous challenges for the idea of a centrally-actuated tensegrity due to an inability to move the payload sufficiently far as to reliably induce a rolling motion and the need for accurate sensing for reactive control. From simulation results with realistic hardware components, we determined that in order achieve ground mobility, an expanded internal volume was necessary in order for there to be enough travel for the center payload to cause a rolling motion. To achieve this, the traditionally straight rigid bodies common to most six-bar designs were replaced with bent rods which can range between 127 to 140 degrees in order to provide the payload with a greater workspace. With this update geometry, motion planning simulations demonstrated successful results using all 12 inner-cables with cable rest lengths constrained between 1 cm to 15 cm for a robot of 60 cm diameter.

4.9.3 Energy Efficient Paired-Cable Actuation

In this section we use introduce in greater detail a new actuation schema initially presented at IROS 2019 [20] – *paired-cable actuation* – and use MPC as a tool to evaluate the performance of the the different hardware topologies. Similar to the 24-motor actuation scheme we introduced in previous sections, the 12-motor paired-cable actuation scheme also controls all 24 cables in a spherical tensegrity; however, for the 12-motor scheme, two cables are coupled by a single motor. For this actuation scheme, a pair of cables meet at a single node – the retraction of one cable means the extension of the other cable in that pair. Thus, while all 24 cables are actuated, only 12 degrees of freedom exist in the system:

$$\mathbf{u} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_{reduced,1} \\
\mathbf{u}_{reduced,2} \\
\vdots \\
\mathbf{u}_{reduced,12}
\end{bmatrix}$$

In practice, this cable coupling has been achieved in hardware through the use of clever mechanisms and/or pulleys, the design of which is outside of the scope of this work. Interestingly, this new paired-cable schema has some practical advantages over its 24-motor schema

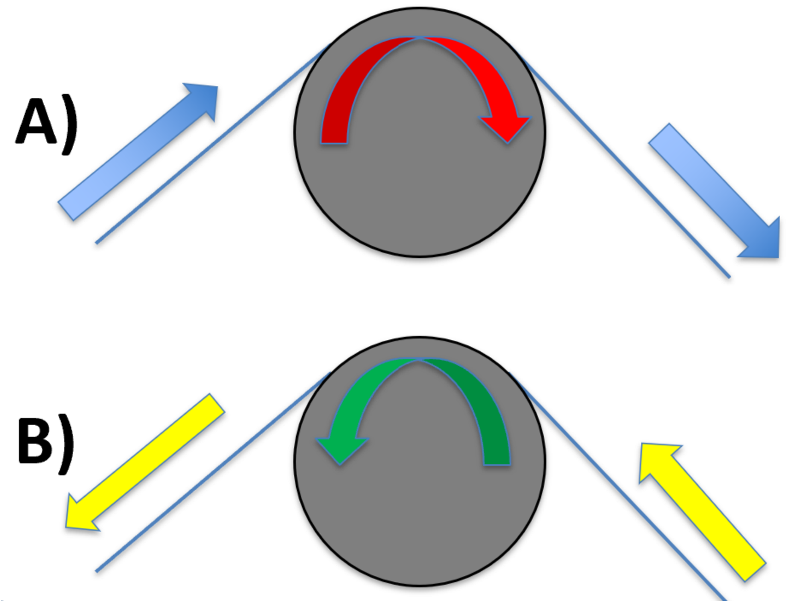


Figure 4.7: Paired-cable actuation enables control of 24 cables using only 12 motors (represented in gray), at the cost of dynamic coupling and reduced control authority. Extensions (retractions) of one cable's rest length mean that the complement cable's rest length must retract (extend).

counterpart. The most notable advantages are that fewer parts are necessary, and consequently, that the tensegrity robot weighs significantly less and is less prone to mechanical failures.

Comparisons of Rolling Locomotion Strategies

In this section, we discuss tensegrity rolling locomotion in detail and compare three cable-actuation schemes, each with varying degrees of control authority: 6-motor (underactuated), 12-motor (paired-actuation), and 24-motor (full-actuation) schema. In particular, these actuation schema vary the number of cables that are driven by motor actuators and consequently which cables remain as passive tensile elements. As a result, we demonstrate that greater control authority can provide improved performance at the cost of additional hardware and controller complexity.

In the results that follow, we utilize MPC with the dynamic constraints introduced earlier to generate optimal state-action trajectories for evaluation. Notably, the simulation model parameters used in these experiments are based on actual hardware parameters of the tensegrity robot shown in Fig. 4.1 (see Table 4.3).

For the remaining subsections, we discuss various characteristic properties of tensegrity locomotion. First, we use the 24-motor fully-actuated scheme to illustrate hardware and

Physical Parameters	**Value**
Rod Length	60 cm
Rod Mass	500 g
Cable Stiffness	400 N/m
Pretension	50 N
Max. Cable Linear Velocity	20 cm/s

Table 4.3: Model parameters used to compare different actuation schema, based on existing Mobile Robot hardware.

controller design considerations that are unique to compliant tensegrity rolling locomotion; namely, we assert that tensegrity stiffness and initial pretension are important hardware and controller design hyperparameters. We conclude this section by evaluating the nominal performance of the three common actuation schema introduced above with respect to relevant performance metrics such as speed, directional trajectory-tracking, and energy efficiency.

Effective Tensegrity Stiffness and Pretension

The inherent compliance of tensegrity structures serves as a benefit with regards to mechanical robustness, particularly in the structure's impact-resilience, natural force distribution, and lack of mechanical stress concentrators. On the other hand, the resulting oscillatory dynamic behavior complicates optimal control policy design. In this section, we present results obtained when adjusting overall tensegrity stiffness and pretension (i.e., the stiffness and initial pretension of all individual series-elastic tension elements) and assess its broader effects on rolling locomotion. For this evaluation, we examine the relative dynamic behavior of the robot as we adjust the overall stiffness and pretension of the fully-actuated robot between the range of 50% and 200% of the nominal values of 400 N/m and 50 N. The simulation parameters we use for MPC are as follows:

As we can see from the results in Fig. 4.8, greater overall stiffness in the robot leads to better rolling performance with the receding horizon controller. Similarly, performance improves as initial pretension of the robot increases, before dropping off. Intuitively, these results match expectations, as greater stiffness creates less oscillatory dynamics which the controller is unable to account for, due to the low controller timestep of 0.01 seconds. Greater stiffness and pretension ensures that the robot is less likely to drastically deform in a detrimental manner by giving immediate and precise control over the state of the robot with less actuation input. That said, the results illustrate that excessively large pretensions can also negatively affect tensegrity locomotion. One possible explanation for this observed behavior is that excessively large pretensions during the initial state require more actuation before the robot can sufficiently change its shape and enter a stable dynamic rolling gait.

Effectively, these results support that tensegrity robots become easier to control using inexpensive and computationally-limited microcontrollers as the tensegrity dynamics approach rigid-like behavior; lower update frequencies are less of an issue as state uncertainty due to

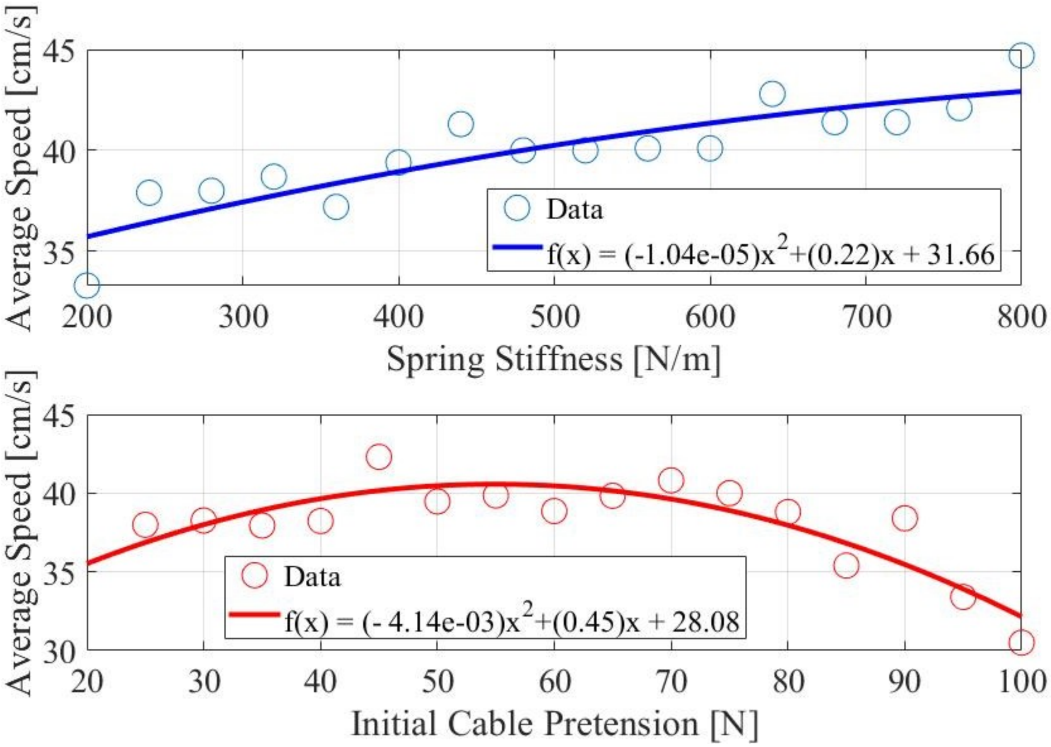


Figure 4.8: A comparison of average rolling speeds vs. overall stiffness and initial cable pretension, with 24-motor scheme. Near-linear fitting curves highlight the overall trends.

compliance in the robot decreases. Unfortunately, greater stiffness in the overall robot also leads to prohibitively high torque and power requirements on the motor actuators, an issue we discuss in greater detail later in Section 4.9.3.

Speed and Directional Control

Next, we compare the performance of the three actuation schema presented earlier (i.e., 6-motor underactuated, 24-motor fully-actuated, and 12-motor paired-actuated) with respect to average rolling speed and directional trajectory control. In this section, we discuss the advantages that are provided through the use of additional cable actuators.

As an illustrative example, we examine the average rolling speed of the robot under each of the three actuation policies, maximizing rolling velocity in a specified direction. To get a more representative average speed, the total number of timesteps we simulate is doubled from the previous section.

From the trials shown in Fig. 4.10, it is clear that the greater control authority, afforded by more actuated cables for 24-cable policies, enables the robot to accomplish locomotion tasks that the 6-motor variant simply cannot complete. In this simple illustrative example, we see that, given the limited degrees of freedom, the 6-motor actuation scheme is simply unable to roll in the desired direction and becomes stuck as it attempts to do so. That

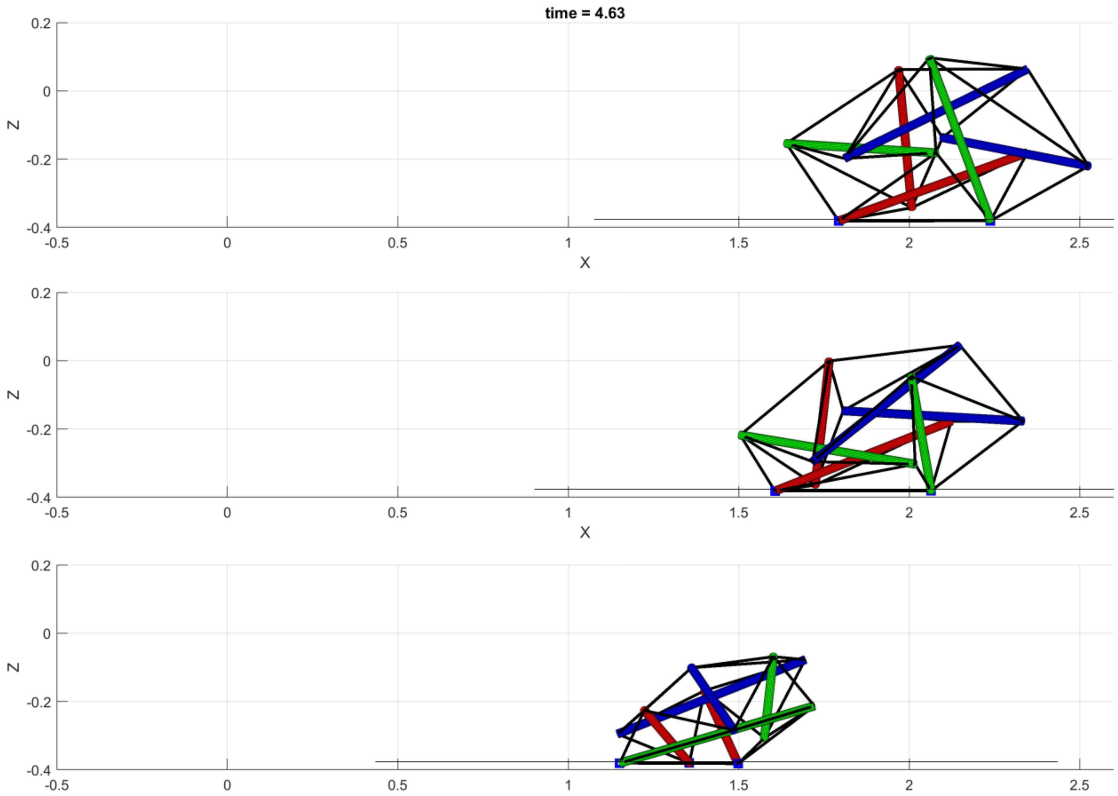


Figure 4.9: Visual depiction of difference in rolling speed for identical tensegrity topology but varying overall tensegrity stiffness. Starting at the same initial pose for all three cases, the stiffer spherical robot rolls farther, faster in the same amount of time. Reduced compliance and oscillatory behavior enables stiffer tensegrity robots (800 N/m stiffness, top) outperform more compliant tensegrities (400 N/m stiffness, middle; 200 N/m stiffness, bottom) at the cost of increased cable tensions and energy costs.

said, both the fully-actuated and paired-actuated 24-cable policies were able to perform reasonably well. Naturally, however, the greater controllability of the fully-actuated system allows for greater directional trajectory-tracking accuracy along the +X-axis. In Fig. 4.11, this improved directional controllability is highlighted as we evaluate the robot's performance in any-direction rolling. In this figure, average rolling speeds are normalized by the fastest experimental trial, so that all arrows lie within the unit circle. In particular, it is shown that while both the 24-motor and 12-motor schemes can achieve reasonable top speeds, the lesser degree of freedom provided by the paired-cable schema (i.e., 12 motor actuators rather than 24), precludes good performance in *all* directions.

To conclude, we summarize the relative performance of each actuation policy, stating the average speeds of each scheme followed by a normalized value – average speed divided by the product of the rod length of the spherical tensegrity design and the maximum linear velocity of the motors. Importantly, we believe that this normalization will simplify comparisons of

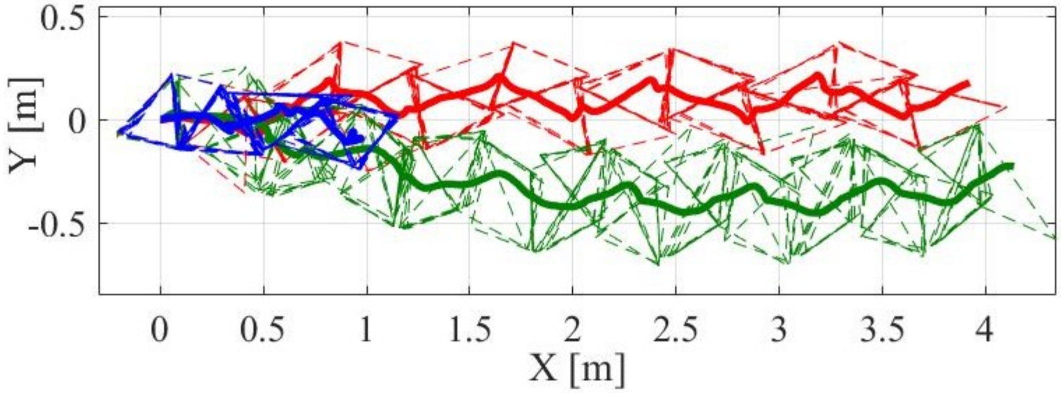


Figure 4.10: Footprint trails of the robot starting at the origin and rolling in the +X-direction for 10 seconds using 24-motor fully-actuated (red), 12-motor paired-actuated (green), and 6-motor actuation policies (blue). Solid lines indicate the robot's center of mass and dotted lines indicate supporting polygons in contact with the ground.

Schema	6-motor	12-motor	24-motor
Avg. Speed	18.1 cm/s	37.9 cm/s	38.2 cm/s
Norm. Speed	1.51e-02	3.16e-02	3.18e-02

Table 4.4: Comparison of rolling speeds with different tensegrity actuation schema.

performance across different hardware configurations and tensegrity topologies.

Energy Efficiency

Next, we briefly discuss the energy efficiency of both the 12-motor paired-cable and 24-motor actuation schema. As shown in the previous section, comparable performance for maximum rolling speeds is achieved for both 12-motor and 24-motor actuation schemes. However, as demonstrated, 12-motor actuation does sacrifice some degree of maneuverability due to the lesser control authority granted. Nevertheless, in this section, we provide compelling motivation for the 12-motor variant on the basis of energy efficiency.

We consider a simplified model of energy-costs which relates tension in the robot to power consumed by the motors. Specifically, cable tensions (see Fig. 4.13) are directly related to load torques on the motor and correlates to current draw and power consumption. If we assume 65% efficient motors, the average Cost of Transport (i.e., energy divided by mass times distance traveled) over 100 trials are 159.9 $\frac{J}{m \cdot kg}$ and 30.7 $\frac{J}{m \cdot kg}$ for 24-motor and 12-motor actuation schema, respectively. Notably, the exceptional Cost of Transport is on par with the locomotive efficiency of many animals found in nature [29] (see Figure 4.12), an aspiration for many mobile robotic systems. Specifically, the unique paired-cable mechanism is able

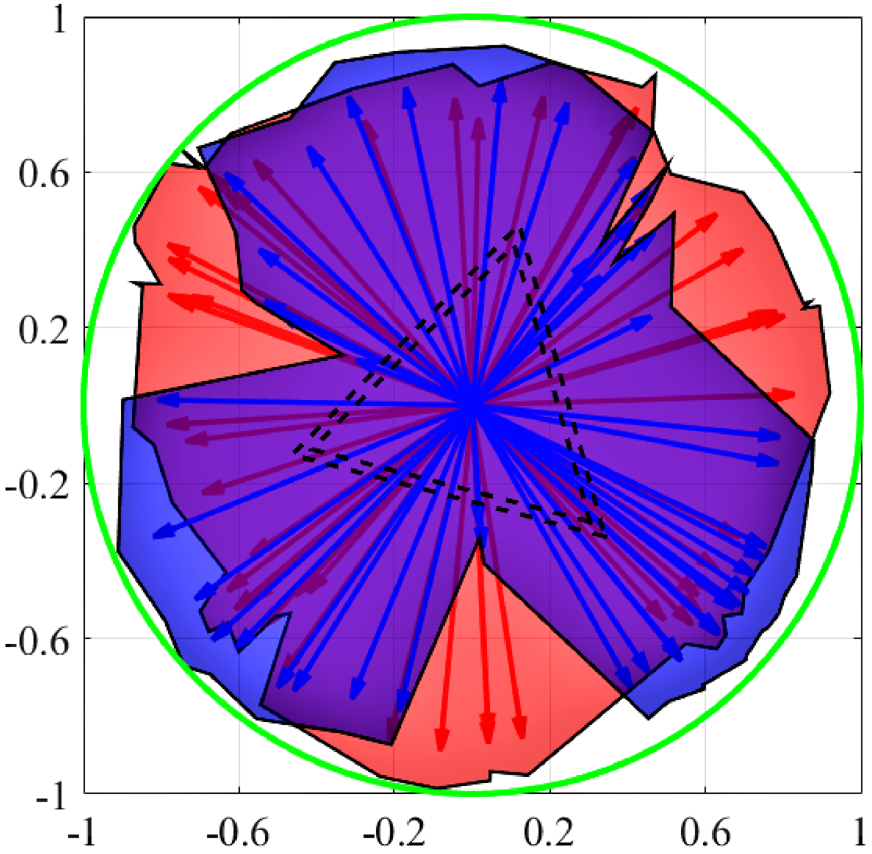


Figure 4.11: Plot showing normalized average speeds in various directions, starting from an identical initial state (base polygon outlined in dashed black lines). Red represents results for fully-actuated 24-motor scheme, and blue represents 12-motor paired-cable actuation scheme.

to leverage complementary tensions of each cable-pair such that the motor does minimal work under normal operating conditions. Thus, while some degree of maneuverability is lost, the energy efficiency gains makes a 12-motor paired-cable actuation schema an enticing candidate tensegrity robot design.

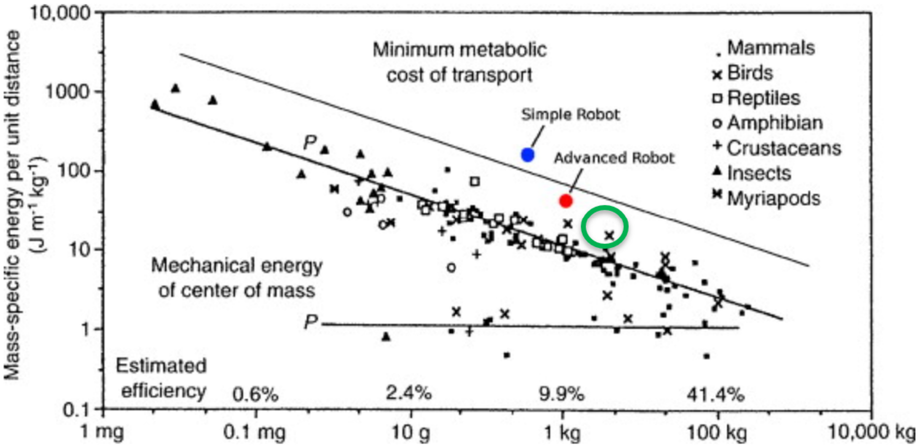


Figure 4.12: Scatter plot of biological systems' energy efficiency versus overall mass, with representative robot examples. Green circle signifies estimated energy efficiency of rolling tensegrity robots. Original plot from [29].

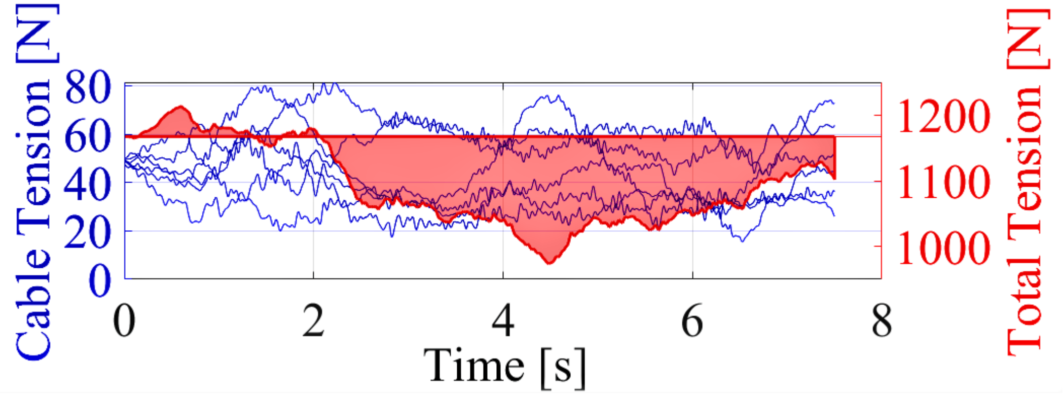


Figure 4.13: Cable tensions for 6 of 24 cables (left) and summed total tension of all 24 cables (right) for 24-motor rolling locomotion. Note, in attaining maximum rolling speed, total tension in the robot remains below the initial pretensioned state after some time (2 seconds).

Chapter 5

Tensegrity State Estimation and Control

5.1 Introduction

Ground contact-rich rolling interactions combined with the many degrees of freedom intrinsic to tensegrity robots can lead to excessive errors between state estimates and reality. To address these challenges of tensegrity locomotion, we evaluate several Bayesian state estimators for sensor fusion which intelligently combine multiple sensor measurements while accounting for sensor and actuator uncertainty (e.g., Unscented/Extended Kalman Filters and Particle Filters.). While the optimal controllers and motion planning trajectories introduced in the last chapter assume perfect state estimation, practical robotics must deal with the dual problem of understanding the robot's current state and pose from imperfect models and noisy sensor measurements. In particular, tensegrity robots have the added challenges of compliance and highly coupled dynamics in contact-rich ground rolling. In this chapter, we explore the effects of different sensor schema and their practical application to tensegrity ground mobility.

5.2 Prior Work

State estimation is an established field with areas of active ongoing research built on foundational classical theory; autonomous robotics, especially, has benefited extensively from advances in state estimation due to the natural affinity of robotics applications towards highly dynamic and uncertain scenarios. In particular, uncertainty naturally arises out of imperfect world models and noisy sensor measurements. Nevertheless, robotic systems and intelligent agents must grapple with understanding their own state relative to their (potentially unknown) environment before there can be any chance for a notion of optimal behavior.

The relatively recent movement towards a probabilistic framework in the last few decades has enabled great progress by providing a common language – that of representing information and uncertainty probabilistically and enabling us to reason about conditional dependencies using Bayesian approaches [97]. However, Bayesian Tracking with true belief probability distributions is generally intractable for real-world problems with continuous state-spaces or discrete but infinitely many states.

Given certain assumptions, however, such as linearity in the dynamics and measurement models along with independent Gaussian noise, tractable formulations can be computed which leverage the assumed structure of the posterior distribution (i.e., affine and linear combinations of jointly Gaussian random variables are also Gaussian). Seminal works written by Swerling [94] and Kalman [45] make use of the linear-Gaussian assumption to present the Kalman Filter as the optimal state estimator from the perspectives of least-squares and posterior belief distributions, respectively. Nonlinear extensions such as the Extended Kalman Filter [45, 44] and the Unscented Kalman Filter [42] enable optimal state estimation for more complex nonlinear systems often found in the real world. For a concise and elucidating comparison between the Extended and Unscented variants, please refer to [103].

Nonparametric filters, which do not rely on a predetermined form of the posterior distribution, instead capture Bayesian statistical information from sampled points or decompose the state space into partitions where calculating the posterior distribution is tractable. The advantages of such approaches entail no parametric assumptions on the representation of posterior densities and the ability to easily handle multimodal beliefs, a weakness of the aforementioned Gaussian filters. The widely popular Particle Filter variants [90, 102] use sampled and re-sampled points which are weighted depending on real-time sensor measurements in order to approximate meaningful measures of the posterior distribution. Histogram Filters [81, 97] in contrast, decompose the continuous state space into finitely many regions in order to represent the true cumulative posterior.

With regards to tensegrity robots, very little work has explored optimal state estimation in highly dynamic contexts. Previous work that featured control and sensing for dynamic tensegrities such as [79, 80] highlight possible hardware combinations of sensors and actuators which are typically found on tensegrity robots. In particular, common motifs for cable-driven robotics such as DC motors with pulley mechanisms, cable tension sensing, orientation measurements, etc. are ingrained throughout existing literature. More recent work such as [17, 15] have directly explored the challenges of state estimation for spherical tensegrities using an Unscented Kalman Filter approach to combine on-board IMU data, actuator state information, and ranging measurements for robot pose estimation and localization. In this case, however, a relatively simplistic actuation policy was used only to induce a single step. The primary focus of the work presented in this chapter is to explore state estimation of tensegrity robotics in the context of optimization-based control for continuous dynamic rolling mobility.

5.3 Contributions of this Chapter

In this chapter, we tackle the dual problem of optimal control – state estimation given uncertain sensor measurements and imperfect dynamics models – applied to tensegrity robotics. In particular, we propose a common set of tensegrity sensor measurements which enable state estimation using traditional nonlinear Kalman Filter approaches. Through comparing and evaluating the Extended Kalman Filter and Unscented Kalman Filter variants, we discuss discrepancies in estimator performance and highlight some key considerations which suggest new challenges that arise in the interaction between tensegrity motion planning and state estimation. To the best of my knowledge, no such work investigating state estimation in the context of highly dynamic tensegrity motion currently exists.

5.4 Tensegrity Sensors and Hardware

As in the previous chapter, we begin by first understanding sensor capabilities in conventional tensegrity hardware designs. Tensegrity sensor hardware often varies across different designs, but over the years, a commonality of necessary sensors has precipitated from practical tensegrity rolling mobility applications. Tensegrity sensor suites primarily focus on sensing capabilities necessary for low-level control of manipulable inputs such as cable length and/or rod length, but recent works have explored sensor combinations which facilitate state estimation and sensor fusion, which is the focus of this chapter. Despite the endeavor to develop spherical tensegrities as a mobile robotics platform, however, little to no work has explored other critical components which are integral to autonomous systems such as perception, localization, and mapping, which we leave to future work. As such, the sensors discussed in this chapter are evaluated under the lens of control and motion planning.

In common designs found in the B.E.S.T Lab, tensegrity robots have a 9-axis inertial measurement unit (IMUs) in each rod which obtains fused information from individual sensors such as three-axis accelerometers, gyroscopes, and magnetometers in order to measure the rods' accelerations, angular velocities, and orientation with respect to some inertial frame. Additionally, sensor information from a motor's angular position can be used to measure the cable rest length for each associated cable, and approximate tension sensing is possible through sensing the current draw from each DC motor. Prior work such as [17] also explored using wideband time-of-flight range sensors for position measurements. Using the Bayesian sensor fusion techniques and state observers presented in this chapter, these disparate sensor measurements of orientation, cable length, nodal velocity, etc. are combined for more accurate state and pose estimation than can be achieved solely through independent, isolated sensor measurements.

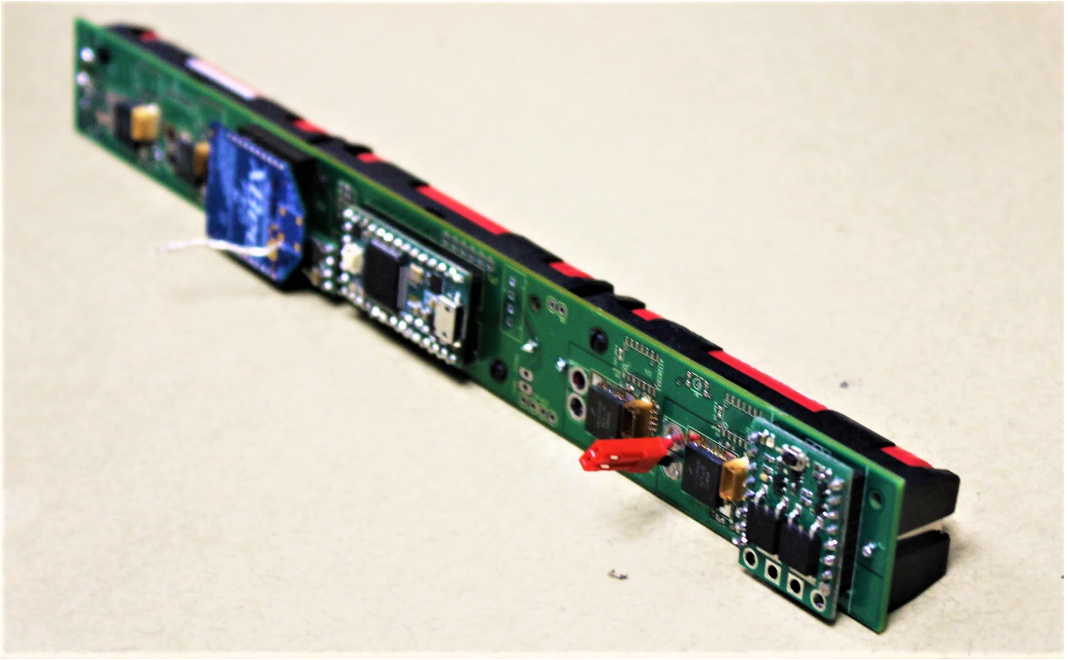


Figure 5.1: Current Mobile Robot hardware PCB featuring an embedded microcontroller, IMU, motor drivers, encoder integrated circuits, wireless radio communication, and battery management system.

5.5 State Estimation

All of the controls techniques and motion planning approaches presented in the last chapter assume exact knowledge of the nominal state of the robot. In reality, the problem we aim to solve is a *Partially Observable Markov Decision Process* (POMDP), as perfect information is never accessible either due to a limited number of sensors, inability to sense specific aspects of the state (*partial observability*), and/or due to measurement noise from the sensors. The next few sections present and evaluate how nonlinear Kalman Filter approaches and Particle Filters perform for various possible tensegrity sensor schema and furthermore quantify the extent to which sensor quality and the coupled, interconnected dynamics of tensegrity robots improves or degrades the accuracy of traditional state estimation techniques.

5.5.1 Background: Bayesian State Estimation

In practice, we measure and keep track of a probablistic *belief* of the robot's state at any given time through the use of a probability distribution of possible states, conditioned on previous state estimates and measurement data from the robot. The overall system we measure is thus a coupled dynamic system where the robot manipulates itself and the environment through control actions and simultaneously perceives its environment through sensors. The coupled dynamics of the robot and environment are represented by two impor-

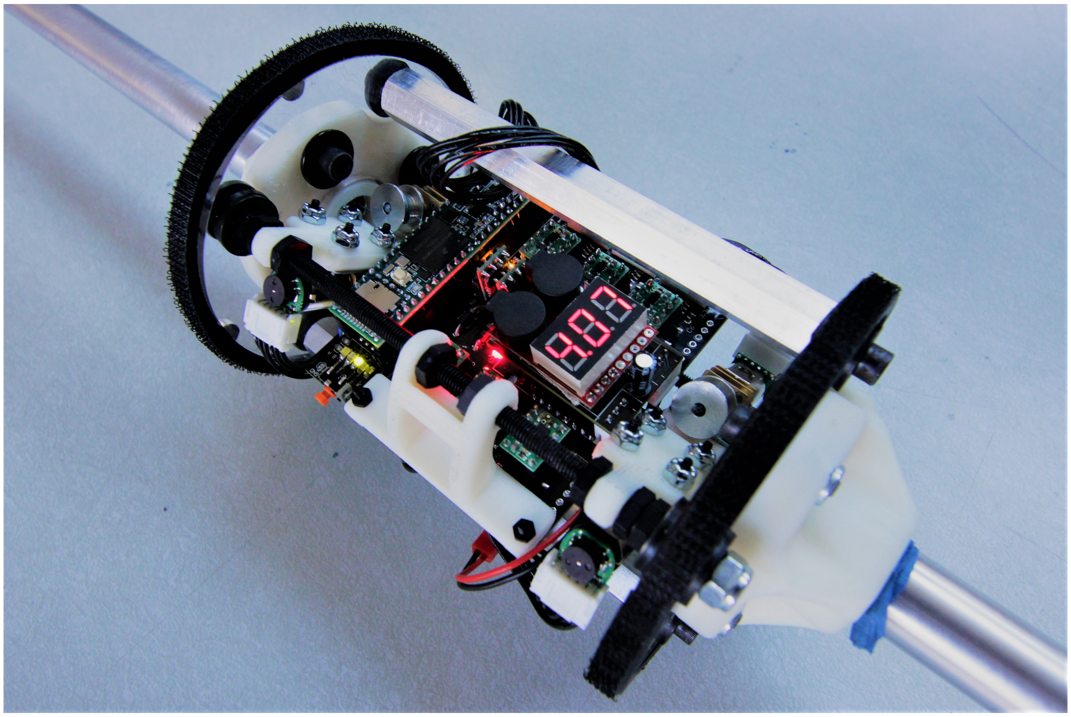


Figure 5.2: Electronics found on the older TT5 hardware constructed in the B.E.S.T. Lab. Pictured are the battery voltage sensor, IMU, Teensy microcontroller, custom PCB, and four small brushed DC motors with magnetic encoders.

tant probablistic equations: the stochastic state transition distributions and measurement distributions which predict 1) how the robot state dynamically evolves over time, and 2) how the measurements relate to the given state of the robot, respectively.

The belief of the robot's state is represented by the conditional probability distribution $f(\mathbf{x}(k)|\mathbf{z}(1:k))$, where $\mathbf{x}(k)$ is the state of the robot at time k and $\mathbf{z}(1:k)$ is the sequence of previous sensor measurements from the first timestep up to time k. In short, this probability distribution quantifies the likelihood of a specific state of the robot, given information collected from the sensors, by leveraging the known dynamics and measurement models of the system.

In practice, Bayesian Tracking and similar approaches are implemented in a recursive fashion to speed up computation and to prevent requiring the entire history of measurements to be stored for future calculations. The well-known recursive approach presented next is split into two distinct stages: the *prior update* and the *measurement update* which incorporate the known dynamics model and the known measurement model, respectively.

Prior Update

The prior update simply increments the state estimate forward given the dynamics and a known input to the system:

$$f(\mathbf{x}(k)|\,\mathbf{z}(1:k\text{-}1)) = \int\limits_{\mathbf{x}(k-1)\in\mathcal{X}} f(\mathbf{x}(k)|\,\mathbf{x}(k\text{-}1),\mathbf{z}(1:k\text{-}1))\; f(\mathbf{x}(k\text{-}1)|\,\mathbf{z}(1:k\text{-}1))\,\mathrm{d}\mathbf{x}(k\text{-}1) \tag{5.1}$$

$$= \int\limits_{\mathbf{x}(k-1)\in\mathcal{X}} f(\mathbf{x}(k)|\,\mathbf{x}(k\text{-}1))\; f(\mathbf{x}(k\text{-}1)|\,\mathbf{z}(1:k\text{-}1))\,\mathrm{d}\mathbf{x}(k\text{-}1) \tag{5.2}$$

In obtaining this equation, we used the total probability theorem in the first line and utilized the assumption that the state $\mathbf{x}(k)$ is conditionally independent from past measurements, given the previous state $\mathbf{x}(k\text{-}1)$. The conditional probability $f(\mathbf{x}(k)|\,\mathbf{x}(k\text{-}1))$ is obtained using the known dynamics of the system and the known control input to the system (if any). Notice here that we rely on the probability distribution $f(\mathbf{x}(k\text{-}1)|\,\mathbf{z}(1:k\text{-}1))$ which is calculated using the measurement update step (presented next) from the previous timestep.

Measurement Update

The measurement update improves upon the prior estimate by incorporating sensor information from the system. Using Bayes Theorem, we can rewrite the *posterior* belief (i.e., the state estimate conditioned on all measurements up to and including the current timestep) in terms of other key related quantities which are more readily obtained:

$$f(\mathbf{x}(k)|\,\mathbf{z}(1\!:\!k)) = f(\mathbf{x}(k)|\,\mathbf{z}(k),\mathbf{z}(1\!:\!k\text{-}1)) \tag{5.3}$$

$$= \frac{f(\mathbf{z}(k)|\,x(k),\mathbf{z}(1\!:\!k\text{-}1))\; f(\mathbf{x}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))}{f(\mathbf{z}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))} \tag{5.4}$$

$$= \frac{f(\mathbf{z}(k)|\,x(k),\mathbf{z}(1\!:\!k\text{-}1))\; f(\mathbf{x}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))}{\int_{\bar{\mathbf{x}}(k)\in\mathcal{X}} f(\mathbf{z}(k)|\,\bar{\mathbf{x}}(k))\; f(\bar{\mathbf{x}}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))\,\mathrm{d}\bar{\mathbf{x}}(k)} \tag{5.5}$$

$$= \frac{f(\mathbf{z}(k)|\,x(k))\; f(\mathbf{x}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))}{\int_{\bar{\mathbf{x}}(k)\in\mathcal{X}} f(\mathbf{z}(k)|\,\bar{\mathbf{x}}(k))\; f(\bar{\mathbf{x}}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))\,\mathrm{d}\bar{\mathbf{x}}(k)} \tag{5.6}$$

where $f(\mathbf{x}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))$ is the *prior* belief of the state obtained from the previous prior update, $f(\mathbf{z}(k)|\,\mathbf{x}(k))$ is calculated using the known observation model (i.e., how likely are the sensor measurements given a specific state), and $f(\mathbf{z}(k)|\,\mathbf{z}(1\!:\!k\text{-}1))$ is the probability of a measurement $\mathbf{z}(k)$ given past measurements – a normalization constant that is independent of the robot state. Equation 5.4 is a direct result of applying Bayes Theorem, and Equation 5.5 uses the Total Probability Theorem. The last line is obtained by simplifying the numerator through assuming that measurements $\mathbf{z}(k)$ and $\mathbf{z}(1:k\text{-}1)$ are conditionally independent, given the current state $\mathbf{x}(k)$.

5.5.2 Background: Kalman Filter

In general, analytically solving the general Bayesian formulation described in the previous section is intractable when the state space $\mathcal{X}$ is continuous. When the states are discrete and finite, the problem can be solved using *Bayesian Tracking*, as the integral is instead replaced with a sum over a finite number of states. However, one case involving continuous state and action spaces in which the true posterior distribution is obtainable is in the specific scenario when we are dealing with linear dynamic systems with Gaussian initial state distributions and independent, uncorrelated Gaussian noise. Consider the case when we have a linear time-varying system:

$$\mathbf{x}(k) = \mathbf{A}(k\text{-}1)\mathbf{x}(k\text{-}1) + \mathbf{u}(k\text{-}1) + \mathbf{v}(k\text{-}1) \tag{5.7}$$

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \tag{5.8}$$

where $\mathbf{A}(k)$ and $\mathbf{H}(k)$ are the linear mappings from the state $\mathbf{x}(k)$ to the next state and to the current measurement, respectively. The input to the system is $\mathbf{u}(k)$, where the traditional linear matrix $\mathbf{B}(k)$ is implicit in this formulation. Finally, the additive process noise and measurement noise at time k are represented with the vectors $\mathbf{v}(k) \sim \mathcal{N}(\boldsymbol{\mu}_v, \mathbf{Q})$ and $\mathbf{w}(k) \sim \mathcal{N}(\boldsymbol{\mu}_w, \mathbf{R})$.

If the initial state $\mathbf{x}(0)$, $\mathbf{v}(k)$, and $\mathbf{w}(k)$ are Gaussian, independent, and mutually distributed, then the prior and measurement update stages described above are summarized below, where $\hat{\mathbf{x}}_p$ and $\hat{\mathbf{x}}_m$ are the state estimates (i.e., the means of the Gaussian distributions) after the prior update and measurement update, and $\mathbf{P}_p(k)$ and $\mathbf{P}_m(k)$ are the respective variances:

Kalman Filter: Prior Update

$$\hat{\mathbf{x}}_p(k) = \mathbf{A}(k\text{-}1)\hat{\mathbf{x}}_m(k\text{-}1) + \mathbf{u}(k\text{-}1) \tag{5.9}$$

$$\mathbf{P}_p(k) = \mathbf{A}(k\text{-}1)^T\mathbf{P}_m(k-1)\mathbf{A}(k\text{-}1) + \mathbf{Q}(k-1) \tag{5.10}$$

Kalman Filter: Measurement Update

$$\mathbf{K}(k) = \mathbf{P}_p(k)\mathbf{H}^T(k)\big(\mathbf{H}(k)\mathbf{P}_p(k)\mathbf{H}(k) + \mathbf{R}(k)\big)^{-1} \tag{5.11}$$

$$\hat{\mathbf{x}}_m(k) = \hat{\mathbf{x}}_p + \mathbf{K}(k)\big(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}_p(k)\big) \tag{5.12}$$

$$\mathbf{P}_m(k) = \big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)\mathbf{P}_p(k) \tag{5.13}$$

$$= \big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)\mathbf{P}_p(k)\big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)^T + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}(k)^T \tag{5.14}$$

The prior update advances the state and error covariance variables forward according to the process dynamics. The measurement update calculates the Kalman Filter Gain and updates the error covariance and state estimate by incorporating the most recent measurements. $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ above are the time-dependent covariance matrices of the Gaussian-distributed process and sensor noise, as described previously. The equations are initialized with $\hat{\mathbf{x}}_m(0)$ and $\mathbf{P}_m(0)$ set to the belief distribution corresponding to the assumed known initial state of the system. The Kalman Filter Gain, $\mathbf{K}(k)$, scales the corrective linear term in Equation 5.50 of the measurement update in an optimal manner according to statistical measures of variance/uncertainty due to both the stochastic process noise as well as measurement noise. Note, while the equations above solve the Bayesian state estimation problem outlined in Equation 5.2 exactly (given linear Gaussian assumptions), the Kalman Filter is also guaranteed to be the best observer of all *linear* estimators which minimizes the variance of the estimation error for any system [86, 105].

5.5.3 Kalman Filter for Nonlinear Tensegrity Systems

As is evident from earlier chapters, tensegrity equations of motion model a complex dynamic system, with nonlinear dynamics and measurement models. For tensegrity robots, the original Kalman Filter updates don't translate directly, as the linear Gaussian assumptions are violated. Instead, in this section, we summarize two variants of the Kalman Filter which can be used to handle nonlinear dynamic systems with varying degrees of accuracy and computational complexity.

The nonlinear tensegrity dynamics were previously outlined in Chapter 3; in this section, we present the remaining nonlinear equations governing the measurement models. As described at the beginning of this chapter, common tensegrity sensors include inertial measurement units (IMU's) which measure acceleration, angular velocity, and orientation, as well as positional encoders which provide information on the cable rest lengths.

Tensegrity Sensor Measurements

Tensegrity robots must deal with many uncertainties that arise due to their compliant, contact-rich high-dimensional dynamics; as such, these robots must leverage multi-modal sensor fusion in order to improve optimal state estimation. As mentioned in previous sections, positional encoders are key sensors which provide information on each of the cable's rest lengths, that – in conjunction with external forces that arise from interactions with the ground and surrounding environment – play a major role in understanding the tension distribution of the elastic cable network of the tensegrity structure. IMUs provide critical measurement data on individual rod orientations to help localize nodal positions of the ends of each rod relative to one another. Finally, gyroscopic angular velocity measurements can also be transformed into nodal velocities, given orientation information and rod geometries.

As a reminder, the state vector of the robot which we wish to estimate concisely aggregates information about rod ends' nodal positions and velocities as well as cable and rod lengths, where input to the system are the rate of change of rod/cable lengths:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k \\ \dot{\mathbf{p}}_k \\ \boldsymbol{\ell}_{cables,k} \\ \boldsymbol{\ell}_{rods,k} \end{bmatrix} \qquad \mathbf{u_k} = \begin{bmatrix} \dot{\boldsymbol{\ell}}_{cables,k} \\ \dot{\boldsymbol{\ell}}_{rods,k} \end{bmatrix} \tag{5.15}$$

and the interesting nonlinear dynamics effects are captured in the following equation for nodal accelerations (see Chapter 3 for more details):

$$\begin{aligned} \ddot{\mathbf{p}} \quad = \quad & \mathbf{W} \cdot \Big(- \mathbf{J^T}(\mathbf{JWJ^T})^{\mathbf{-1}}(\dot{\mathbf{J}}\dot{\mathbf{p}} - k_s\mathbf{G} - k_d\dot{\mathbf{G}}) + \\ & (\mathbf{I}_{3N} - \mathbf{J^T}(\mathbf{JWJ^T})^{\mathbf{-1}}\mathbf{JW})(\sum_j \boldsymbol{\gamma}_j + \mathbf{F}_{external})\Big) \end{aligned} \tag{5.16}$$

where $\mathbf{W}$ is the inverse mass matrix, $\mathbf{J}$ and $\dot{\mathbf{J}}$ are the Jacobians of the constraint functions and their time derivatives with respect to the nodal positions, respectively, and $\mathbf{G}$ and $\dot{\mathbf{G}}$ are the vector-valued constraint functions and respective time derivatives.

In order to leverage the state estimation and sensor fusion techniques described in this chapter, we define the nonlinear measurement models mapping current robot state to the sensor measurements from the IMUs, encoders, etc. First, rod orientation information is mapped according to the following equations:

$$\mathbf{u}_i = -(\mathbf{R}_i \otimes \mathbf{I}_3) \cdot \mathbf{p}_k \qquad \mathbf{z}_{orient,i} = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|} + \mathbf{w}_i \tag{5.17}$$

$$\mathbf{z}_{orient} = \begin{bmatrix} \mathbf{z}_{orient,1} \\ \mathbf{z}_{orient,2} \\ \mathbf{z}_{orient,3} \\ \mathbf{z}_{orient,4} \\ \mathbf{z}_{orient,5} \\ \mathbf{z}_{orient,6} \end{bmatrix} \in \mathbb{R}^{18} \tag{5.18}$$

In short, the noisy measurement vector $\mathbf{z}_{orient}$ stacks the X,Y,Z coordinates of directional unit vectors which lie along the longitudinal axis of each rod, where $\mathbf{R}_i$ is the i^{th} row of the rod connectivity matrix defined as follows:

$$\mathbf{R} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \in \mathbb{R}^{6\times 12} \tag{5.19}$$

Positional encoders directly attached to the motor actuators give us direct measurements of the cable rest lengths. We assume a linear relationship between revolutions and rest length according to the following simple equation:

$$\mathbf{z}_{cable,i} = \eta_i \cdot \ell_{cable,i} + \mathbf{w}_i \qquad\qquad \mathbf{z}_{cable} = \begin{bmatrix} \mathbf{z}_{cable,1} \\ \mathbf{z}_{cable,2} \\ \vdots \\ \mathbf{z}_{cable,24} \end{bmatrix} \in \mathbb{R}^{24} \tag{5.20}$$

where η_i is a constant ratio relating the number of motor i's revolutions to cable i's rest length, often found empirically as a function of motor pulley diameter and cable thickness.

Finally, nodal velocities, $\mathbf{z}_{velocity}$ are also sensed using noisy gyrosopic data using fused sensor information from the IMUs. The final overall sensor measurement vector combining accelerometer, gyroscope, and encoder sensor information is thus:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_{orient} \\ \mathbf{z}_{cable} \\ \mathbf{z}_{velocity} \end{bmatrix} \in \mathbb{R}^{78} \tag{5.21}$$

The represented set of sensors described above is only one of many possible sensor schema for tensegrity autonomy. In the simulation experiments that follow, the results demonstrate that the above sensors constitute a sufficient but not necessary set of measurements, with lesser or alternative sensor information leading to acceptable but less accurate state estimates. Other considerations for sensor measurement data such as rod accelerations, contact forces, and cable tensions can all be utilized for improved sensor fusion and have been proposed for future tensegrity designs. In practice, the combination of sensors described in this work provide a good foundation for understanding important insights for tensegrity state estimation; we leave exciting investigations into different tensegrity sensor hardware configurations for future work.

Extended Kalman Filter

The first nonlinear extension to the vanilla Kalman Filter which we discuss is the *Extended Kalman Filter* (EKF), which linearizes the nonlinear dynamics about the latest state estimate. For the EKF update, the nonlinear process and measurement equations are linearized about the current state estimates, and the standard Kalman Filter update equations (Equations 5.9 - 5.33) are straightforwardly applied to calculate the Process and Measurement Updates using the linearized equations.

Consider the nonlinear discrete-time system:

$$\mathbf{x}(k) = q_{k-1}(\mathbf{x}(k\text{-}1), \mathbf{u}(k\text{-}1), \mathbf{v}(k\text{-}1)) \tag{5.22}$$

$$\mathbf{z}(k) = h_k(\mathbf{x}(k), \mathbf{w}(k)) \tag{5.23}$$

where q_{k-1} and h_k are the nonlinear process and measurement functions, respectively, and $\mathbf{v}_k$ and $\mathbf{w}_k$ are Gaussian, mutually independent noise vectors. Statistical measures of all relevant random variables are as follows:

$$\mathbb{E}[\mathbf{x}(0)] = \mathbf{x}_{init}, \qquad Var[\mathbf{x}(0)] = \mathbf{P}_{init} \tag{5.24}$$

$$\mathbb{E}[\mathbf{v}(k\text{-}1)] = 0, \qquad Var[\mathbf{v}(k\text{-}1)] = \mathbf{Q}(k\text{-}1) \tag{5.25}$$

$$\mathbb{E}[\mathbf{w}(k)] = 0, \qquad Var[\mathbf{w}(k)] = \mathbf{R}(k) \tag{5.26}$$

If we assume that we have access to the previous state estimate mean and variance, $\mathbf{x}_m(k-1)$ and $\mathbf{P}_m(k-1)$, then the modified Process and Measurement Updates are summarized below:

Extended Kalman Filter: Prior Update

$$\hat{\mathbf{x}}_p(k) = q_{k-1}(\hat{\mathbf{x}}_m(k\text{-}1), \mathbf{u}(k\text{-}1), 0) \tag{5.27}$$

$$\mathbf{P}_p(k) = \mathbf{A}(k\text{-}1)^\top \mathbf{P}_m(k\text{-}1)\mathbf{A}(k\text{-}1) + \mathbf{L}(k\text{-}1)^\top \mathbf{Q}(k\text{-}1)\mathbf{L}(k\text{-}1) \tag{5.28}$$

where we define the linearized $\mathbf{A}(k\text{-}1)$ and $\mathbf{L}(k)$ matrices as:

$$\mathbf{A}(k\text{-}1) := \frac{q_{k-1}(\hat{\mathbf{x}}_m(k\text{-}1), \mathbf{u}(k\text{-}1), 0)}{\partial \mathbf{x}} \qquad \mathbf{L}(k\text{-}1) := \frac{q_{k-1}(\hat{\mathbf{x}}_m(k\text{-}1), \mathbf{u}(k\text{-}1), 0)}{\partial \mathbf{v}} \tag{5.29}$$

Extended Kalman Filter: Measurement Update

$$\hat{\mathbf{x}}_m(k) = \hat{\mathbf{x}}_p + \mathbf{K}(k)\big(\mathbf{z}(k) - h_k(\hat{\mathbf{x}}_p(k), 0)\big) \tag{5.30}$$

$$\mathbf{K}(k) = \mathbf{P}_p(k)\mathbf{H}^\top(k)\big(\mathbf{H}(k)\mathbf{P}_p(k)\mathbf{H}(k) + \mathbf{M}(k)^\top \mathbf{R}(k)\mathbf{M}(k)\big)^{-1} \tag{5.31}$$

$$\begin{aligned} \mathbf{P}_m(k) &= \big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)\mathbf{P}_p(k) && (5.32) \\ &= \big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)\mathbf{P}_p(k)\big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)^\top + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}(k)^\top && (5.33) \end{aligned}$$

where the linearized matrices are defined as:

$$\mathbf{H}(k) := \frac{h_k(\hat{\mathbf{x}}_p(k), 0)}{\partial \mathbf{x}} \qquad \mathbf{M}(k) := \frac{h_k(\hat{\mathbf{x}}_p(k), 0)}{\partial \mathbf{w}} \tag{5.34}$$

The main intuition for the EKF process update is to propagate the state forward using the true nonlinear dynamics and to modify the variance according to the linearized dynamics. Similarly, the state estimation is improved during the EKF measurement step by taking into account the actual measurement and the predicted measurement as expected by the nonlinear measurement model. As opposed to the original Kalman Filter, the EKF cannot be computed offline as linearization matrices must be calculated at current state estimates. Additionally, it is important to keep in mind that the EKF only approximates the true mean and variance of the random variables, with highly nonlinear discontinuous dynamics leading to poor approximations. In the context of tensegrity ground mobility, the assumptions of mild nonlinearities and unimodal distributions are violated, which leads to excessively large estimation errors, as will see in later sections.

Unscented Kalman Filter

In contrast to the EKF, which approximates the nonlinear dynamics and measurement models, the *Unscented Kalman Filter* (UKF) seeks to directly approximate the probability distribution that arises as a result of the nonlinear equations instead. This alternative approach utilizes a set of specifically selected points called *sigma-points* which are transformed through the full nonlinear functions and uses the sampled points to obtain the relevant statistics of the random variables. The intuition behind the UKF is that it can be a simpler task to approximate a probability distribution than to approximate a nonlinear function. In this subsection, we summarize the simpler nonlinear UKF for *additive* noise which is applicable for tensegrity systems where independent Gaussian noise affects the process dynamics and measurements affinely, with no coupling or co-dependent effects. We thus assume normally distributed noise which is often valid for sensor noise that may arise due to many independent random effects (i.e., manufacturing tolerances, environmental conditions, temperature, humidity, etc.), according the the central limit theorem [78]. For details in the derivation of UKF and extensions for nonlinear noise, we refer the reader to [41, 43, 86].

We now consider systems of the following form, with additive noise:

$$\mathbf{x}(k) = q_{k-1}(\mathbf{x}(k\text{-}1), \mathbf{u}(k\text{-}1)) + \mathbf{v}(k\text{-}1) \tag{5.35}$$

$$\mathbf{z}(k) = h_k(\mathbf{x}(k)) + \mathbf{w}(k) \tag{5.36}$$

where q_{k-1} and h_k are the nonlinear process and measurement functions, respectively, and $\mathbf{v}_k$ and $\mathbf{w}_k$ are Gaussian, mutually independent noise vectors. As before, statistical measures of all relevant random variables are:

$$\mathbb{E}[\mathbf{x}(0)] = \mathbf{x}_{init}, \qquad Var[\mathbf{x}(0)] = \mathbf{P}_{init} \tag{5.37}$$

$$\mathbb{E}[\mathbf{v}(k\text{-}1)] = 0, \qquad Var[\mathbf{v}(k\text{-}1)] = \mathbf{Q}(k\text{-}1) \tag{5.38}$$

$$\mathbb{E}[\mathbf{w}(k)] = 0, \qquad Var[\mathbf{w}(k)] = \mathbf{R}(k) \tag{5.39}$$

Unscented Kalman Filter: Prior Update

We first generate the $2n$ sigma-points, where n is the dimension of the state vector. Afterwards, the prior sigma-points are evolved through the nonlinear dynamics assuming zero process noise:

$$\mathbf{s}_{\mathbf{x}_m(k-1),i} = \hat{\mathbf{x}}_m(k\text{-}1) + \left(\sqrt{n\mathbf{P}_m(k\text{-}1)}\right)_i \tag{5.40}$$

$$\mathbf{s}_{\mathbf{x}_m(k-1),n+i} = \hat{\mathbf{x}}_m(k\text{-}1) - \left(\sqrt{n\mathbf{P}_m(k\text{-}1)}\right)_i \tag{5.41}$$

$$\mathbf{s}_{\mathbf{x}_p(k),i} = q_{k-1}\Big(\mathbf{s}_{\mathbf{x}_m(k-1),i}\Big) \quad \forall \quad i \in \{0, 1, ..., 2n-1\} \tag{5.42}$$

where the subscript i refers to the i^{th} column of the scaled matrix square root (calculated for example using the Cholesky decomposition) of the measurement covariance matrix. Using these sampled sigma-points, the prior statistics are straightforwardly calculated:

$$\hat{\mathbf{x}}_p(k) = \sum_{i=0}^{2n-1} \frac{1}{2n}\mathbf{s}_{\mathbf{x}_p(k),i} \tag{5.43}$$

$$\mathbf{P}_p(k) = \sum_{i=0}^{2n-1} \frac{1}{2n}\Big(\mathbf{s}_{\mathbf{x}_p(k),i} - \hat{\mathbf{x}}_p(k)\Big)\Big(\mathbf{s}_{\mathbf{x}_p(k),i} - \hat{\mathbf{x}}_p(k)\Big)^\top + \ \mathbf{Q}(k\text{-}1) \tag{5.44}$$

Unscented Kalman Filter: Measurement Update

The sigma-points calculated in the Prior Update step are passed through the nonlinear measurement model and the measurement covariance matrix $\mathbf{P}_{zz}(k)$ and cross covariance matrix $\mathbf{P}_{xz}(k)$ are calculated:

$$\mathbf{s}_{\mathbf{z}(k),i} = h_k\Big(\mathbf{s}_{\mathbf{x}_p(k),i}\Big) \quad \forall \quad i \in \{0, 1, ..., 2n-1\} \tag{5.45}$$

$$\hat{\mathbf{z}}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n}\mathbf{s}_{\mathbf{z}(k),i} \tag{5.46}$$

$$\mathbf{P}_{zz}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n}\Big(\mathbf{s}_{\mathbf{z}(k),i} - \hat{\mathbf{z}}(k)\Big)\Big(\mathbf{s}_{\mathbf{z}(k),i} - \hat{\mathbf{z}}(k)\Big)^\top + \ \mathbf{R}(k) \tag{5.47}$$

$$\mathbf{P}_{xz}(k) = \sum_{i=0}^{2n-1} \frac{1}{2n}\Big(\mathbf{s}_{\mathbf{x}_p(k),i} - \hat{\mathbf{x}}_p(k)\Big)\Big(\mathbf{s}_{\mathbf{z}(k),i} - \hat{\mathbf{z}}(k)\Big)^\top \tag{5.48}$$

Using these values, the Kalman Filter gain and linear corrective terms are subsequently obtained and are used to calculate the posterior estimate:

$$\mathbf{K}(k) = \mathbf{P}_{xz}(k)\mathbf{P}_{zz}(k)^{-1} \tag{5.49}$$

$$\hat{\mathbf{x}}_m(k) = \hat{\mathbf{x}}_p + \mathbf{K}(k)\big(\mathbf{z}(k) - \hat{\mathbf{z}}(k)\big) \tag{5.50}$$

$$\mathbf{P}_m(k) = \mathbf{P}_p(k) - \mathbf{K}(k)\mathbf{P}_{zz}(k)\mathbf{K}(k)^\top \tag{5.51}$$

In summary, the UKF is appealing because it does not require calculating time-derivatives throughout the system dynamics and measurement model. This is useful when Jacobians may be difficult to obtain from complex hybrid dynamics. As well, the unscented transform utilized in the UKF approximates the mean correctly up to second-order, as compared to the EKF which is correct only up to first order, and partially incorporates some approximation of higher order terms [41]. Of course, this increased accuracy comes at the cost of computation time, as multiple dynamic simulations must now be completed at ever iteration.

5.5.4 Simulation Results and Discussion

In this section, we discuss simulation results which compare state estimation performance and accuracy for both the nonlinear Extended Kalman Filter and Unscented Kalman Filter for the canonical six-bar spherical tensegrity. In short, the UKF outperforms the EKF in terms of accuracy, as expected, at the expense of greater computational cost. For this experiment, we simulate the six-bar spherical tensegrity at rest, with no cable actuation. A noisy, perturbed state estimate is initially provided, and recursive filter updates using both nonlinear filters are performed over 500 time steps of 5e -3 seconds. Given the physical model parameters and noise characteristics specified in Table 5.1 and Table 5.2, we find that on average, the UKF performed 32% better than the EKF counterpart, with respect to nodal position absolute estimation error. As alluded to in [17], the EKF does not perform well on highly nonlinear systems such as tensegrities, where first-order approximations do not accurately represent the true propagation of uncertainties.

In particular, the EKF tends to perform poorly with regards to nodal position and velocity error along the Z-axis (see Figure 5.6 and Figure 5.8). One probable reason for the discrepancy in estimation accuracy for the Z-direction is the complexity of contact interactions between the ground surface and contact nodes which constitute the supporting base polygon of the robot. Namely, the discontinuous dynamics which are modeled are difficult to capture through first-order approximations. Qualitatively, we observe that a ‘popcorn’ behavior arises, where the base nodes continuously alternate between touching the ground and ‘popping’ back up just slightly above the surface level on the next recursive state estimate. This is likely due to high floor stiffness and atypical collision interference that is an artifact of inaccurate state estimates rather than reality. Notably, this inaccuracy in ground contact interactions is pivotal in understanding the challenges of coupled interdependent state estimation and controls for tensegrity robots; intricate static friction and viscous friction

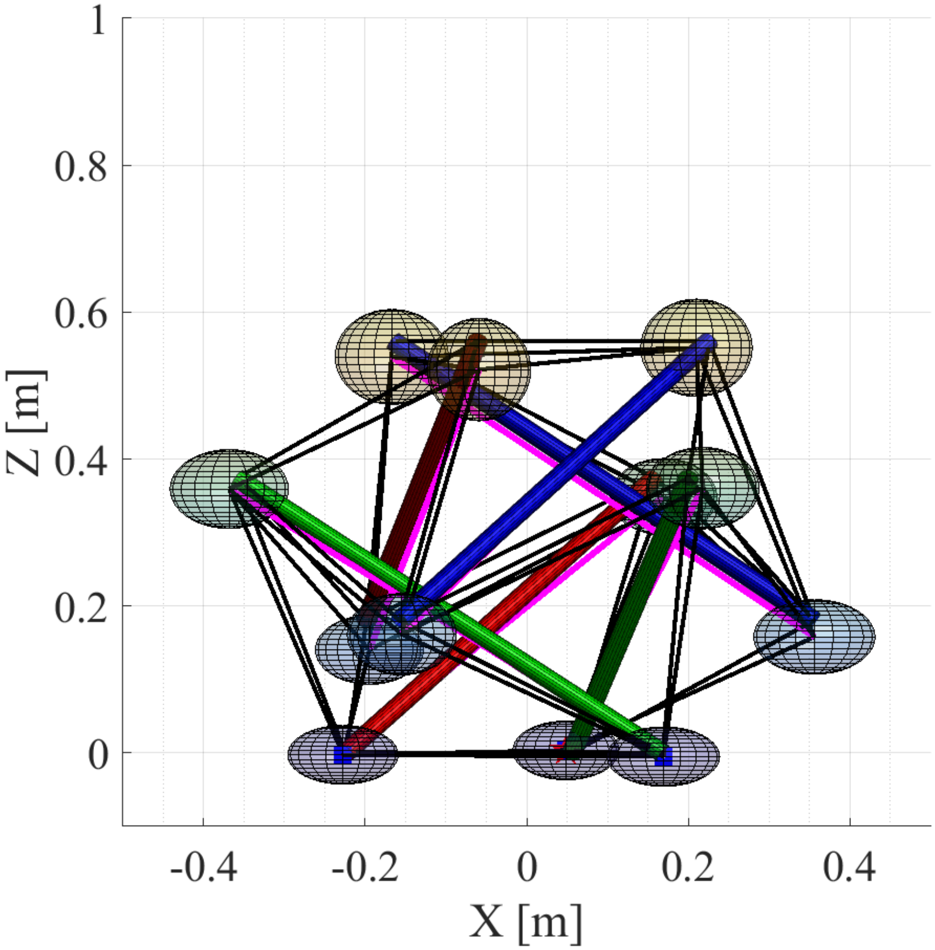


Figure 5.3: Side view of the six-bar tensegrity model and UKF state estimate. Estimated rod positions and orientations are depicted in magenta, offset from the closest ground truth values whose rods are shown in red/green/blue cylinders. Predicted standard deviations of nodal positions are depicted as uncertainty ellipsoids centered at each node estimate.

interactions play a dominant role in determining how the robot will deform and thus how the center of mass evolves over time in order to achieve a rolling motion. A summary of state estimation errors for this experiment are presented in Table 5.3 and Table 5.7 for the EKF and UKF, respectively.

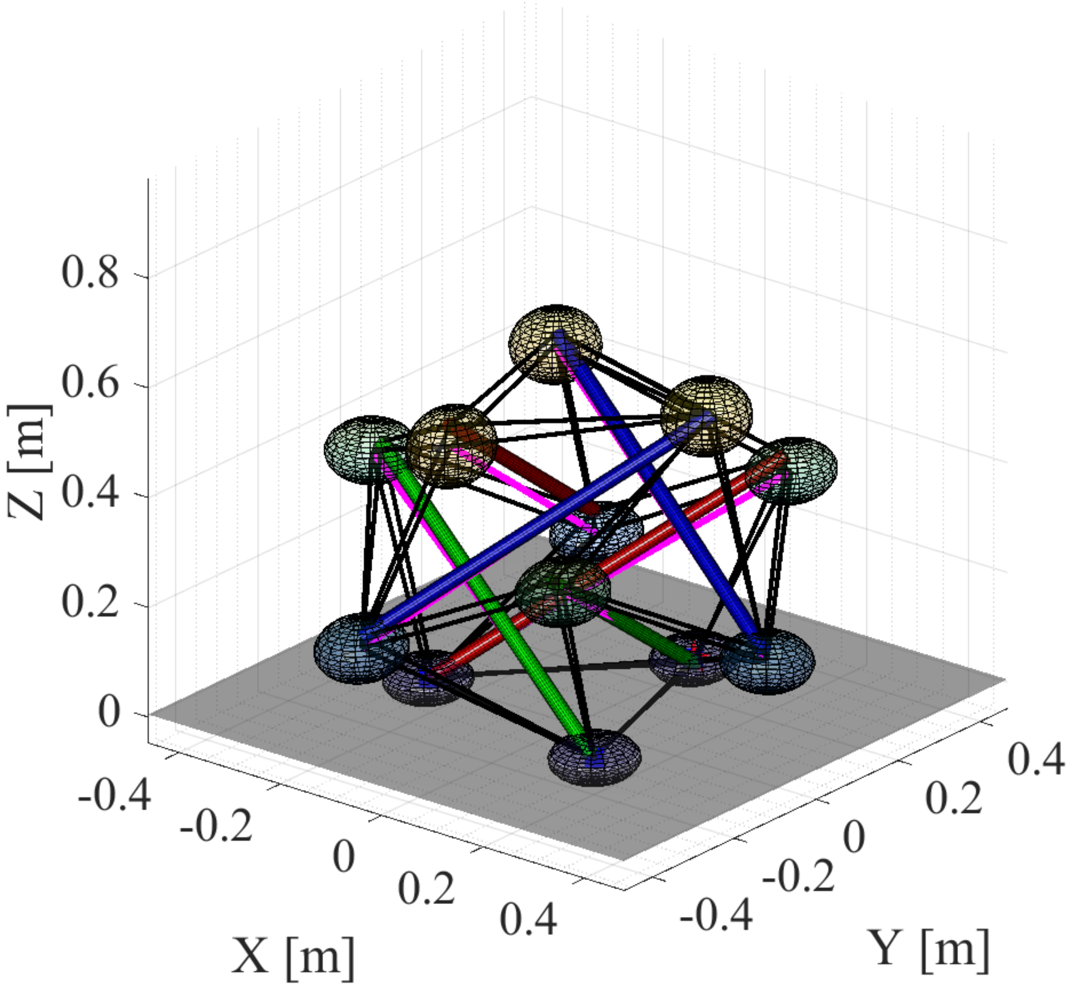


Figure 5.4: Isometric view of the simulated robot and UKF estimate of the rod positions and cable lengths. Estimated rod positions and orientations are depicted in magenta, offset from the closest ground truth values whose rods are shown in red/green/blue cylinders. Predicted standard deviations of nodal positions are depicted as uncertainty ellipsoids centered at each node estimate. Note here that rods are estimated to be slightly lower than ground truth values, likely due to a combination of ground contact inaccuracies and overestimation of cable lengths which result in a less stiff robot than in reality. Nodal positions which are higher (greater Z-value) tend to have larger deviations in the Z-direction, as errors in rod orientations and cable tensions aggregate when moving farther away from the ground-truth surface, having no direct sensor measurements of node height.

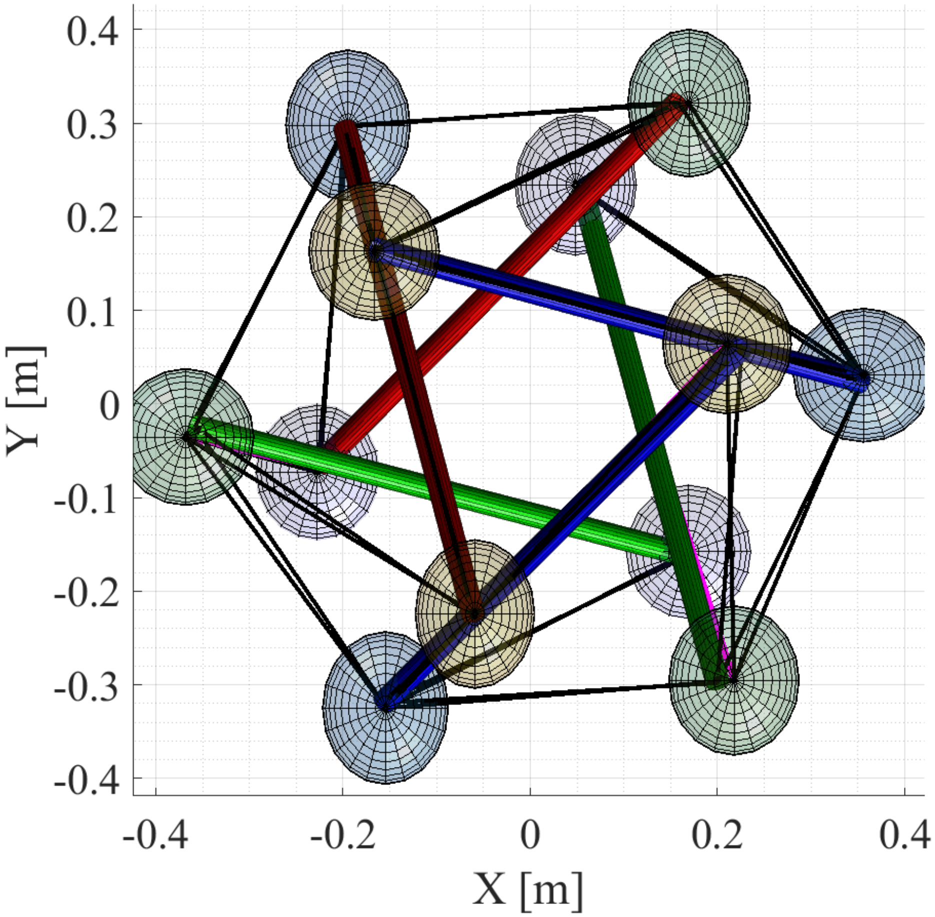


Figure 5.5: Top-down view of the simulated robot and UKF estimate of the rod positions and cable lengths. Estimated rod positions and orientations are depicted in magenta, offset from the closest ground truth values whose rods are shown in red/green/blue cylinders. Predicted standard deviations of nodal positions are depicted as uncertainty ellipsoids centered at each node estimate. State estimates in the XY-plane tend to be fairly accurate, as rod orientation sensors are combined for sensor fusion. Coupling between rods through imperfect cable estimates mean that sensor measurements must corroborate neighboring rods.

Physical Parameters	**Value**
Rod Length	60 cm
Rod Mass	300 g
Cable Stiffness	800 N/m
Pretension	30 N
Max. Cable Linear Velocity	10 cm/s

Table 5.1: Model parameters used to simulate and compare different state estimation approaches, based on existing Mobile Robot hardware.

Sensor Measurement	**Standard Deviation**
Rod Orientation	± 5 degrees
Cable Rest Length	± 5 cm
Nodal Velocity	± 1 m/s

Table 5.2: Sensor noise characteristics used to simulate and compare EKF and UKF state estimation approaches.

State	**EKF Error Quantiles**				
	.025	.25	.50	.75	.975
Node X-Position [m]	0.0259	-0.0090	-0.0001	0.0088	0.0268
Node Y-Position [m]	-0.0270	-0.0088	-0.0002	0.0091	0.0275
Node Z-Position [m]	-0.0399	-0.0114	-0.0023	0.0099	0.0337
Node X-Velocity [m/s]	-2.1107	-0.6596	0.0669	0.8314	2.2275
Node Y-Velocity [m/s]	-2.1757	-0.7662	0.0021	0.8187	2.2903
Node Z-Velocity [m/s]	0.5444	2.5021	3.3084	4.0654	5.4781
Cable Length [m]	-0.0123	-0.0042	0.0009	0.0065	0.0252

Table 5.3: Simulation quantile statistics for state estimation error obtained using the **Extended Kalman Filter**.

State	**UKF Error Quantiles**				
	.025	.25	.50	.75	.975
Node X-Position [m]	-0.0180	-0.0059	0.0000	0.0061	0.0172
Node Y-Position [m]	-0.0178	-0.0062	-0.0002	0.0062	0.0182
Node Z-Position [m]	-0.0344	-0.0172	-0.0078	-0.0010	0.0100
Node X-Velocity [m/s]	-0.0933	-0.0302	-0.0003	0.0316	0.0943
Node Y-Velocity [m/s]	-0.0900	-0.0307	0.0007	0.0326	0.0940
Node Z-Velocity [m/s]	-0.0927	-0.0315	0.0030	0.0345	0.0962
Cable Length [m]	-0.0285	-0.0104	-0.0009	0.008	0.0255

Table 5.4: Simulation quantile statistics for state estimation error obtained using the **Unscented Kalman Filter**.

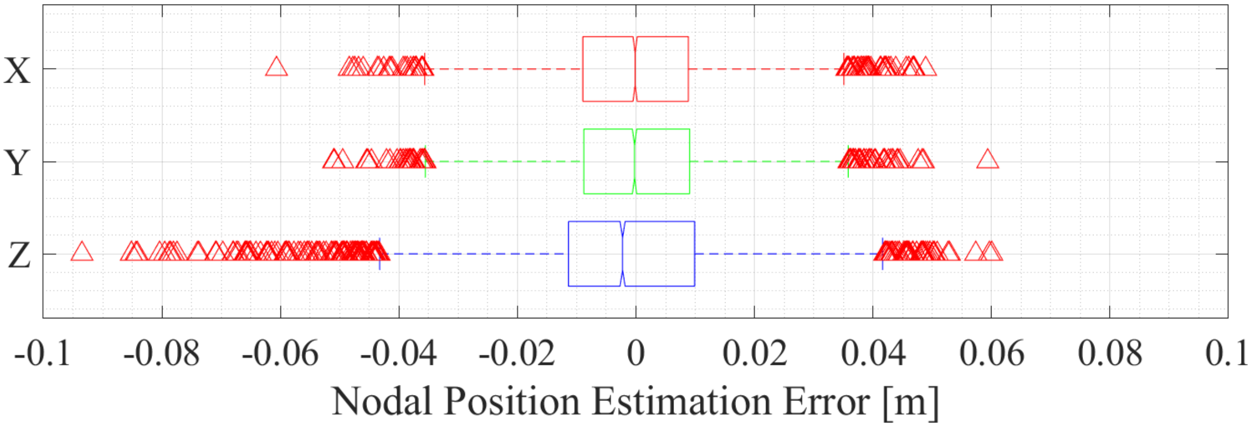


Figure 5.6: XYZ Nodal position estimation errors obtained when using the **Extended Kalman Filter**. Boxes indicate range of 25% and 75% quantiles, median depicted as notched centerline. Whiskers represent 1.5 times the interquantile range beyond the 25% and 75% quantiles. Outliers are represented by red triangles.

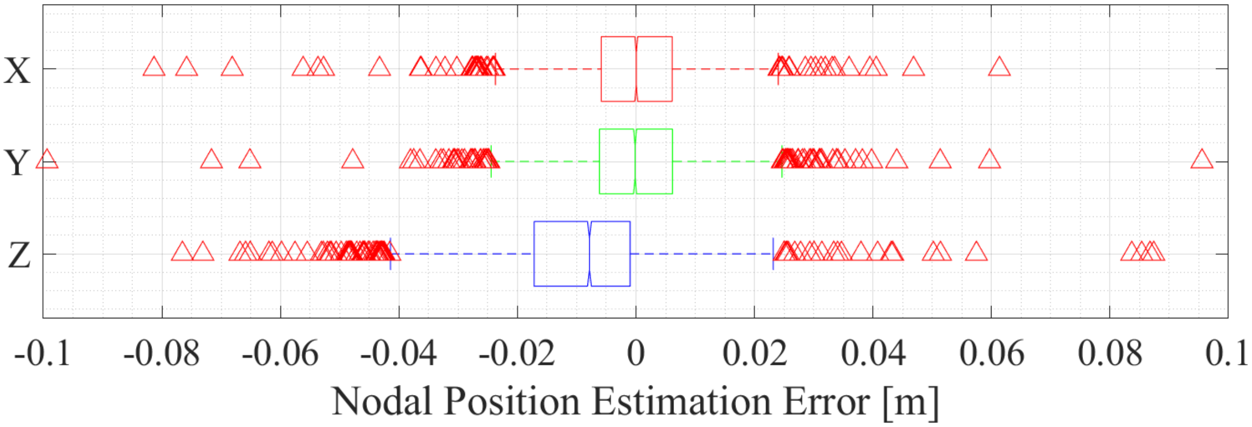


Figure 5.7: XYZ Nodal position estimation errors obtained when using the **Unscented Kalman Filter**. Boxes indicate range of 25% and 75% quantiles, median depicted as notched centerline. Whiskers represent 1.5 times the interquantile range beyond the 25% and 75% quantiles. Outliers are represented by red triangles. Note that Z-values tend to be consistently underestimated.

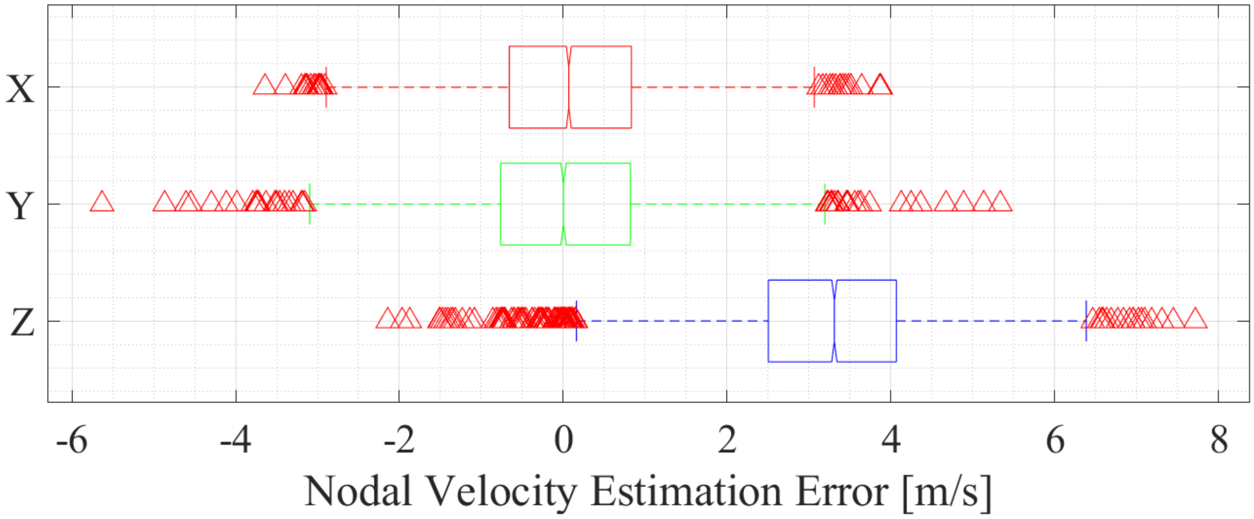


Figure 5.8: XYZ Nodal velocity estimation errors obtained when using the **Extended Kalman Filter**. Boxes indicate range of 25% and 75% quantiles, median depicted as notched centerline. Whiskers represent 1.5 times the interquantile range beyond the 25% and 75% quantiles. Outliers are represented by red triangles. Observe the extreme errors in the positive Z-direction, likely caused due to erroneous contact interactions and collisions between inaccurate state estimates and the surface plane.

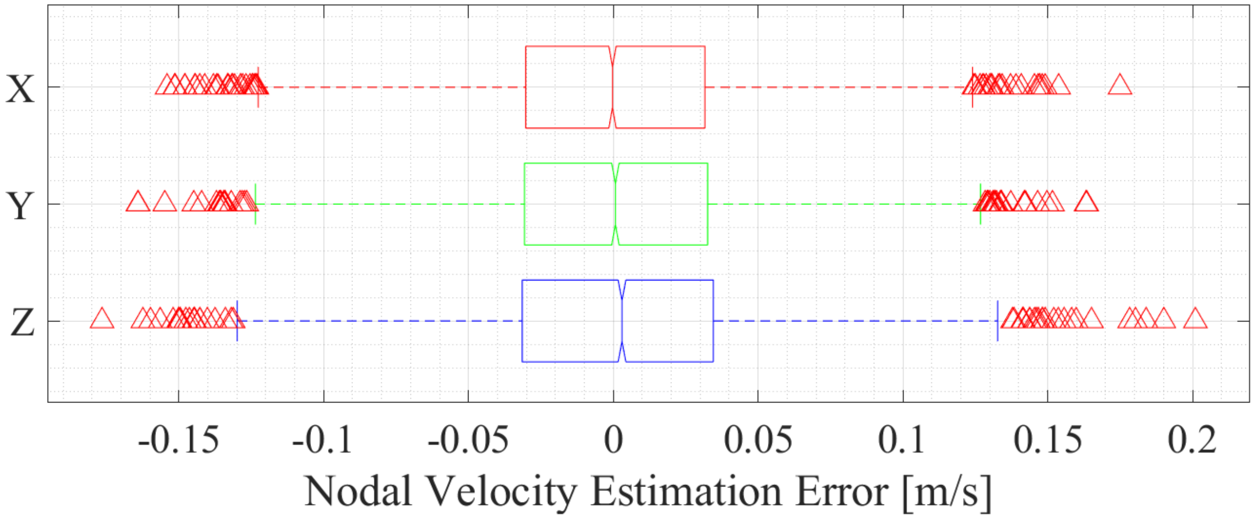


Figure 5.9: XYZ Nodal velocity estimation errors obtained when using the **Unscented Kalman Filter**. Boxes indicate range of 25% and 75% quantiles, median depicted as notched centerline. Whiskers represent 1.5 times the interquantile range beyond the 25% and 75% quantiles. Outliers are represented by red triangles. Note here the difference in scale for estimation error, as compared to results obtained using EKF.

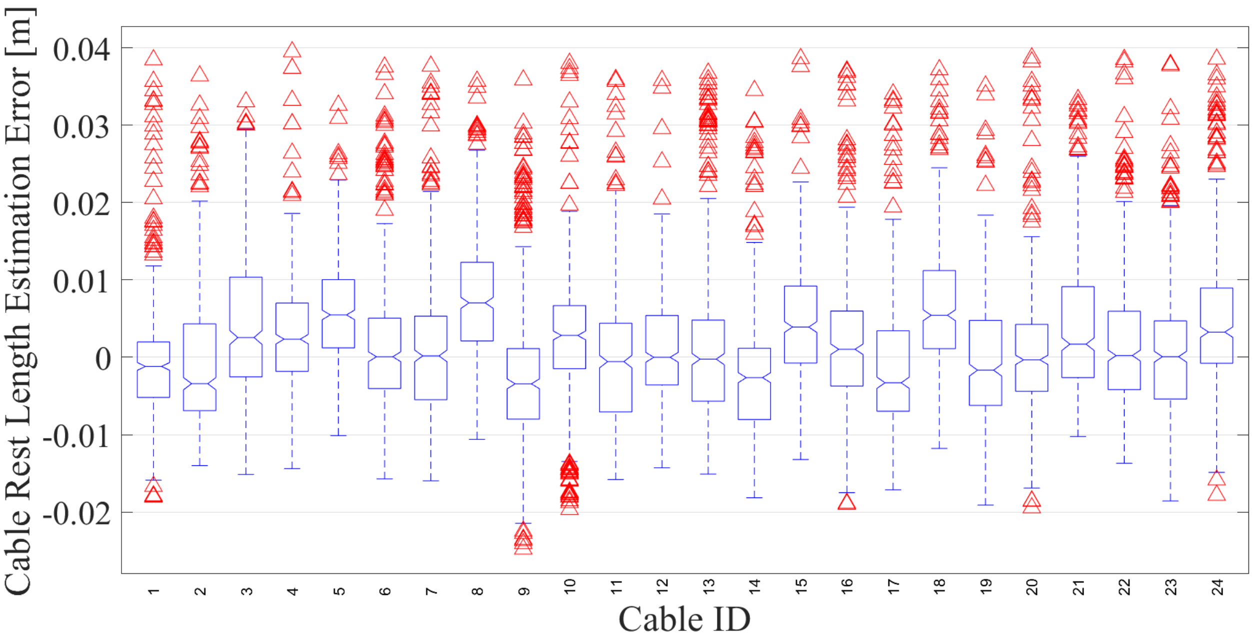


Figure 5.10: Cable length estimation errors obtained when using the **Extended Kalman Filter**. Boxes indicate range of 25% and 75% quantiles, median depicted as notched centerline. Outliers are represented by red triangles. Cable length estimates from EKF tend to be overestimated rather than underestimated.

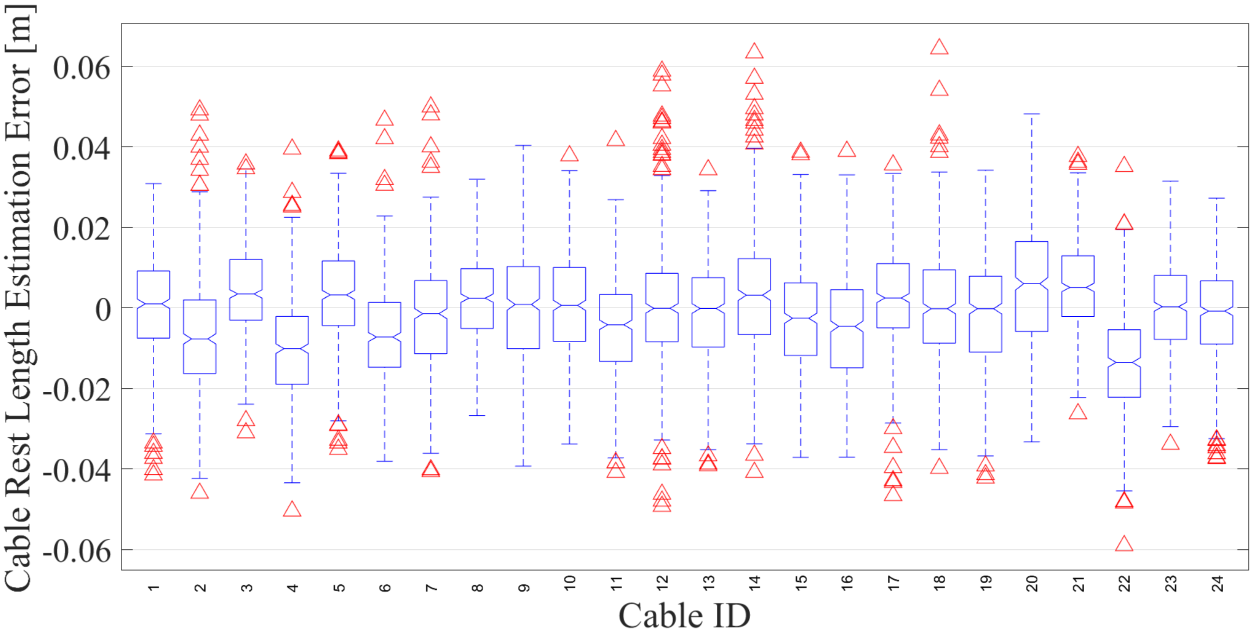


Figure 5.11: Cable length estimation errors obtained when using the **Unscented Kalman Filter**. Boxes indicate range of 25% and 75% quantiles, median depicted as notched centerline. Outliers are represented by red triangles. Interestingly, UKF provides no significant advantage here over EKF. One reason for this could be due to the direct (linear) correspondence between sensor measurement (from motor encoders) and cable length.

Chapter 6

Tensegrity Motion Planning Extensions - Imitation Learning and Robust Control

6.1 Introduction

In the previous chapters, we introduced several model-based approaches that leverage known dynamics of tensegrity systems for motion planning and control. These methods utilize the well-structured equations of motion for tensegrity dynamics to find optimal action inputs in a numerically rigorous way that contrasts with conventional heuristic approaches for tensegrity control. In many practical cases, however, the dynamics of the system may not be entirely known and only partial information - such as the measured performance of the control policy or specific examples of successful input sequences - are provided. Similarly, autonomous control often requires real-time feedback control in order to immediately handle unforeseen disturbances as they occur. In these interesting scenarios where tensegrity systems are tasked to behave optimally under uncertainty and with limited information, we explore some extensions to the approaches presented in prior chapters which may help to handle critical computation speed requirements and/or uncertainty in the system.

6.2 Contributions of this Chapter

In this chapter, we explore how imitation learning may be utilized in combination with optimal control to obtain control policies for tensegrities which can produce optimal actions in real-time. Afterwards, a new minimax robust control formulation is introduced which we can use to guide the robot towards optimal actions in the face of adversarial disturbances. In these scenarios, these approaches must contend with partial or imperfect information about the robot and its interactions with the environment. While the focus of these results is primarily in their application towards intelligent control of tensegrity robotics, this research touches more broadly on important research endeavors involving optimal control of complex dynamical systems with high-dimensional state, observation, and action spaces in the context of partial information and real-world uncertainty.

6.3 Imitation Learning with Optimal Control Experts

Solving the necessary constrained optimization problems, even with the convex formulations described in Chapter 4, requires significant computation time of up to one-hundredth of a second for every control update. More efficient software implementations could further improve computation time, but still would likely not allow for MPC to be run on a less powerful embedded microcontroller in real-time with readily available hardware. Thus, the compliance of these tensegrities is both a benefit and a burden – as can be seen from the results in previous chapters, the robust physical attributes of these unique systems come hand-in-hand with controller complexity.

6.3.1 Input Remapping and Online Control using Deep Learning

Our approach to addressing the intractability of real-time MPC with tensegrities is to utilize MPC and iLQR methods not as an online control policy but as an offline 'expert' reference for supervised *imitation learning.* In short, imitation learning is a machine learning approach that leverages expert-labeled data of optimal input-action pairs in order to train a policy to mimic expert behavior. Using an imitation learning approach, rather than solve a constrained optimization problem for MPC in real-time, we use a feedforward neural network trained on optimal state-action trajectories to allow for real-time implementation of feedback control. Furthermore, we leverage *input remapping* to allow us to generate a reduced-order mapping by providing full-state information during optimal trajectory generation while only using a subset of the states (e.g., observation information readily available through hardware sensors) for the learning process [56]. Utilizing this approach, these results demonstrate real-time directed spherical tensegrity rolling with multi-cable actuation – i.e., using simultaneous actuation of all 24 cables to achieve rolling locomotion in any arbitrary user-defined direction.

For this work, 150+ trajectories are generated using the MPC approach described in Chapter 4 and are used as demonstration trajectories to train a contextual neural network policy which accepts state estimates and a user-defined desired rolling direction. To ensure that the expert trajectories generate a wide support for all user-defined directions, not only is the overall desired direction of the straight-line path randomized (as in Figure 4.2), but intermediate corrective directions – provided whenever the robot deviated from the straight-line path – are also recorded for each timestep, greatly increasing the variation of observed commanded directions. Additionally, some data pre-processing was also utilized to add Gaussian noise to the input and to exploit rotational symmetry for the contextual policy.

To help improve the robustness of the policy to model perturbations, we also ensure that the state-action trajectories that are generated have rod masses which are randomly

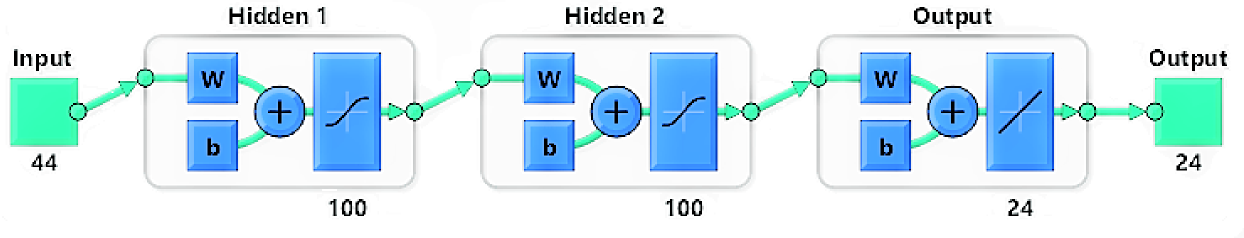


Figure 6.1: Feedforward neural network used for reduced-order contextual policy. The policy takes partial state information as input and outputs an action for each cable; has nearly 11 million individual weight parameters.

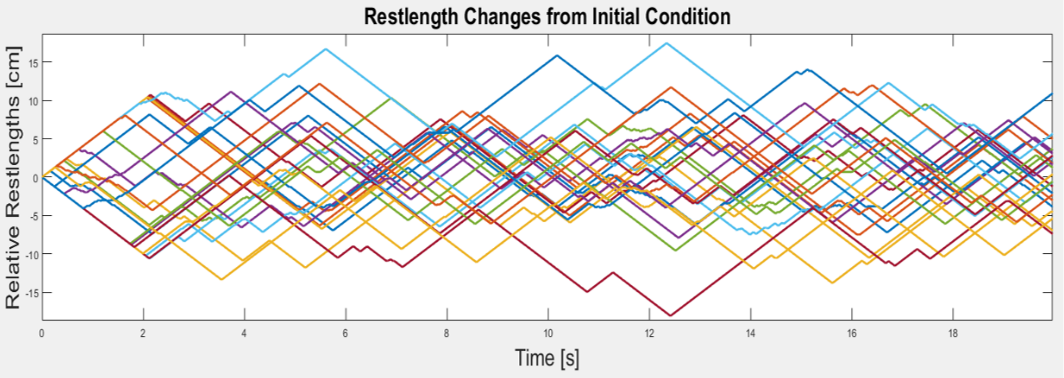


Figure 6.2: Controlled cable rest lengths using Model Predictive Control.

drawn from a uniform distribution with nominal mean of 1.1 kg and spread of +/- 10%. Based on recent results in [73] for robust policies through learning, this parameter variation approach helps the policy better generalize and reduces the effects of discrepancies between the simulation source domain and hardware target domain.

A contextual deep neural network policy (Fig. 6.1) with two hidden layers of 100 nodes each and hyperbolic tangent sigmoid activation function is trained to map a reduced-order input vector, $X \in \mathbb{R}^{44}$, to a discrete action vector, $U \in \mathbb{R}^{24}$.

Specifically, the input is a concatenation of: 1) rod orientations (represented through functions of spherical coordinates), 2) cable restlength deviations from the neutral pose, and 3) a contextual input of x, y coordinates corresponding to a desired rolling direction. Each action $U_i \in \{-1, 0, 1\}$ corresponds to a max velocity retraction, hold and maintain restlength, or max velocity release command, respectively, to one of the 24 actuated cables. In practice, this discretization of the continuous action space helped speed up learning time and improve sample-efficiency. To ensure that cable restlengths remained in a reasonable range, hard limits for minimum and maximum allowable lengths were set, helping prevent the robot from unreasonably de-tensioning and collapsing or tensioning to dangerous levels.

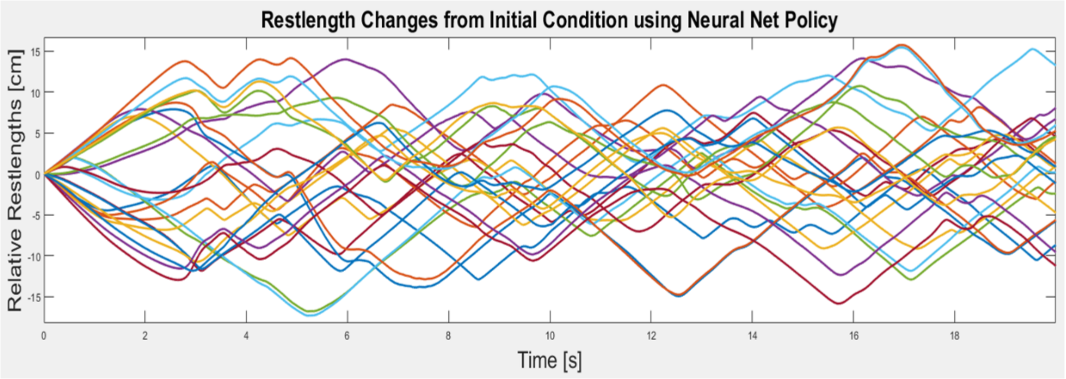


Figure 6.3: Controlled cable rest lengths from a contextual neural network policy trained using imitation learning. Note the smoother transitions for cable actuation as compared to that of trajectories generated using MPC.

From the results, we find that the neural network policy trained using simple supervised learning successfully captures the general patterns of optimal MPC state-action trajectories to allow the robot to locomote when starting from arbitrary poses, even with only partial observability. Although cable rest length commands often reached the upper and lower permissible limits (likely due to the discrete action space with maximum velocity extension or retraction), the simulated robot was able to perform dynamic rolling motions in directions that could be updated in real-time, starting from different initial conditions.

As a simple example, desired directions were given to command the robot to roll in a square trajectory from different initial conditions (see Fig. 6.4). Though optimality guarantees are no longer provided, using a deep neural network trained with supervised machine learning on simulated 'expert' MPC trajectories is a relatively simple, sample-efficient approach for effective real-time feedback control of complex, compliant tensegrity systems.

6.4 Robust Nonlinear Trajectories with Minimax Iterative Dynamic Games

The optimization-based techniques described in the previous chapter enable us to quickly generate optimal actuation policies for any desired tensegrity behavior which we can define a representative reward/cost function for. One weakness of the model-based approaches presented previously, however, is their dependency on accurate dynamics models in generating optimal motion planning trajectories. Any disparity, uncertainty, or model mismatch leads to inaccuracies when using an incorrect nominal model to predict behavior over longer time horizons. This brittleness limits their application in practical use cases where nominal models may be difficult to obtain or subject to uncertainty. In the context of flexible mobile robotics such as tensegrities, this uncertainty is further compounded with discontinuous nonlinearities that are present in contact-rich rolling locomotion.

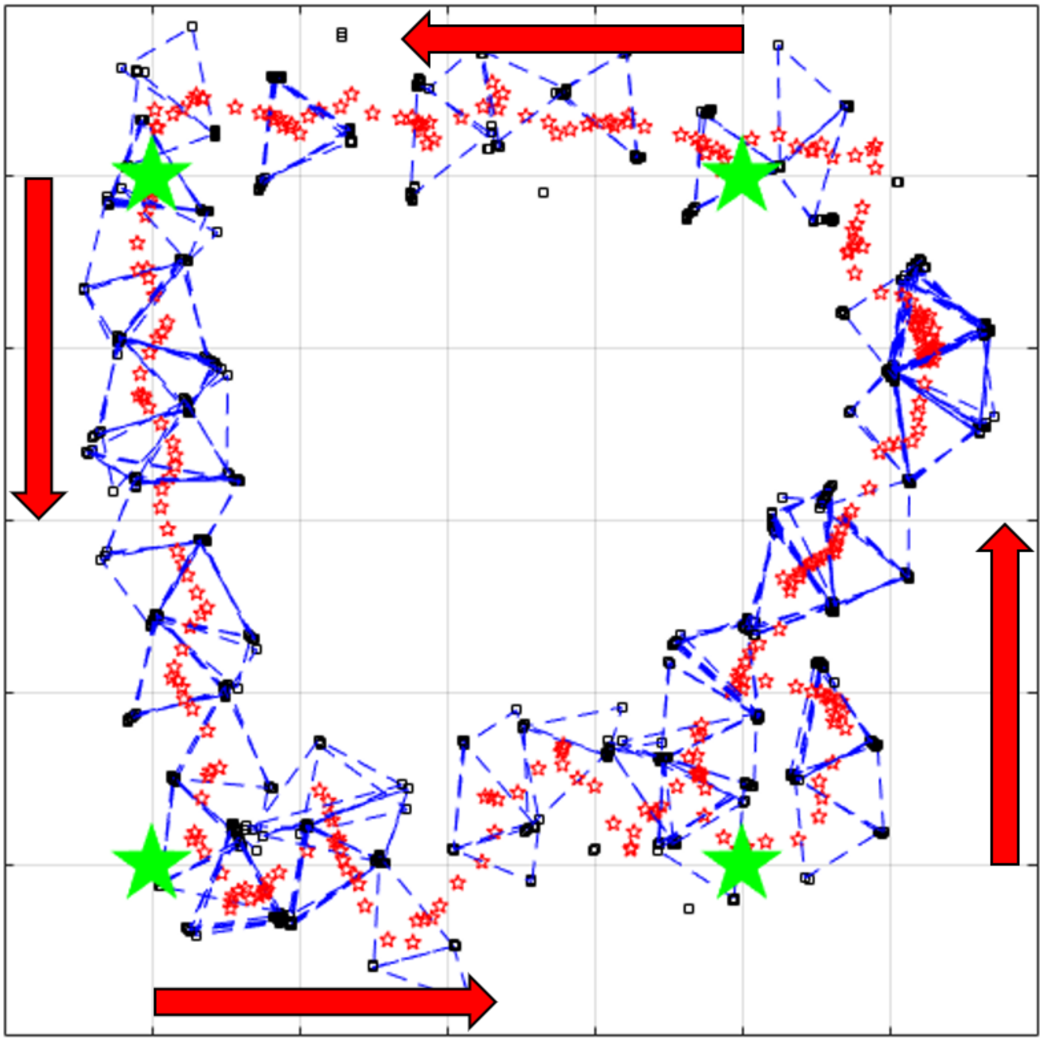


Figure 6.4: Rolling locomotion in a square trajectory controlled using a contextual policy trained through supervised learning. Green stars are waypoint destinations. Red points are the CoM over time. Blue outlines are the supporting base polygons of the robot and black squares signify nodal contact with the ground. For a video, please visit `youtu.be/lXWUPFmPY04`

To address this, techniques such as Robust Model Predictive Control [9, 19, 55, 5], H^{∞} [16] Control, and other parametric approaches using Linear Fractional Transformations for bounded disturbances such as [67, 13] have been utilized in application to linear systems, leading to important results in the field of robust control. In contrast, existing approaches for nonlinear systems are less common, with works such as [104] relying on heuristic or sampling-based Monte Carlo approaches.

Prior work such as [65, 66] frames the robust optimization problem as a *minimax* dynamic game, commonly found in robust control literature, which here aims to find the (locally) best input control to the system, subject to an adversarial disturbance. In this work, we use a minimax approach to extend the techniques described in the previous section and expand upon prior work by investigating this framework in the context of receding horizon control.

Formally, two-player minimax trajectory optimization seeks to solve an optimization problem with respect to both an optimal input *and* adversarial disturbance. We define the cost-to-go J_i as the partial sum of costs from timesteps i to N:

$$J_i(\mathbf{x}_0, \pi, \psi) = \min_{\mathbf{u}_{i \to N-1}} \sum_{k=i}^{N-1} \ell(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) + \ell_N(\mathbf{x}_N) \tag{6.1}$$

where the nominal agent chooses its action $\mathbf{u}_k$ under a policy $\{\pi = \pi_0, ...\pi_T\}$, and the uncertainty's actions $\mathbf{w}_k$ are governed by a policy $\{\psi = \psi_0, ...\psi_T\}$. As before, the stage cost is a function of the states, inputs, and disturbance, and ℓ_N represents the terminal cost of the final state.

The goal of minimax optimization is to find a pair of *saddle point equilibrium* policies (π^*, ψ^*) that satisfy:

$$J_0(\mathbf{x}_0,\, \pi^*,\, \psi) \leq J_0(\mathbf{x}_0,\, \pi^*,\, \psi^*) \leq J_0(\mathbf{x}_0,\, \pi,\, \psi^*) \quad \forall\ \pi \in \Pi, \psi \in \Psi, \mathbf{x}_0 \in \mathcal{X}$$

More generally, to find a policy at timestep t, the optimal control sequence pair $\{\mathbf{u}_t^*, \mathbf{w}_t^*\}$ are the *argmin* and *argmax* sequences, respectively, of the following optimization problem:

$$J_t^*(\mathbf{x}_t) = \min_{\pi \in \Pi} \max_{\psi \in \Psi} J_t(\mathbf{x}_t, \pi, \psi) \tag{6.2}$$

$$= \min_{\pi \in \Pi} \max_{\psi \in \Psi} \left[\ell_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) + J_{t+1}^*(x_{t+1})\right] \tag{6.3}$$

$$= \min_{\pi \in \Pi} \max_{\psi \in \Psi} \left[\ell_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) + J_{t+1}^*(f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t))\right] \tag{6.4}$$

The input policy is made robust by modifying the nominal stage cost ℓ_t with a norm on the adversarial input:

$$\bar{\ell}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = \ell(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) - \gamma g_t(\mathbf{w}_t) \tag{6.5}$$

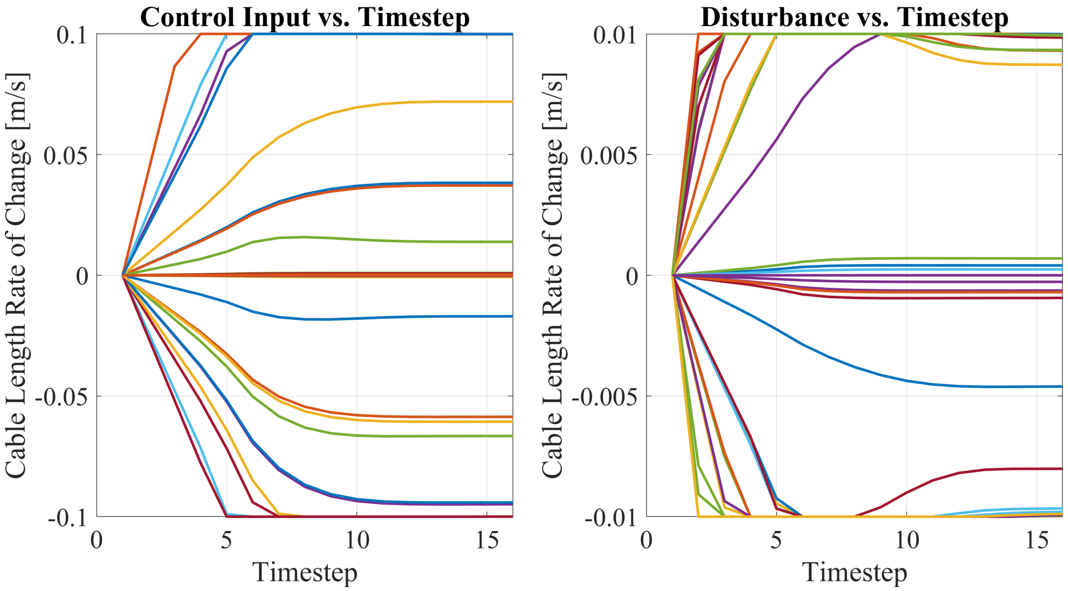


Figure 6.5: Example inputs for the control policy (left) and adversarial disturbance (right).

where $g_t(\cdot)$ is a non-negative penalty norm on the adversarial disturbance, decreasing the cost as disturbance magnitude increases. As gamma decreases, the adversary is able to take larger actions with less penalty. Thus, gamma is a hyperparameter we can tune to modify the robustness of a given policy, with lower gamma values corresponding to more robust policies that are able to handle larger magnitude perturbations. As gamma goes to infinity, the optimal action for the adversarial disturbance is to do nothing; alternatively, the lowest γ-value for which we can achieve acceptable performance can serve as a measure of robustness.

The adversary thus faces the complement optimization problem of *maximizing* the modified cost function, and balances maximizing the cost function with the minimal magnitude disturbance:

$$\max_{\psi \in \Psi} \sum_{k=0}^{N} \ell(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) - \gamma g_t(\mathbf{w}_t) = \max_{\psi \in \Psi} \sum_{k=0}^{N} \bar{\ell}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{6.6}$$

To obtain the optimal pair of policies (π^*, ψ^*), we leverage an approach similar to that of Section 4.7 and apply iLQR to converge to a saddle point equilibrium. In contrast to Equation 4.6, we define the minimax value function over both control and the additional disturbance inputs:

$$V(\mathbf{x}, i) = \min_{\mathbf{u}} \max_{\mathbf{w}} \big(\ell(\mathbf{x}, \mathbf{u}, \mathbf{w}) + V(f(\mathbf{x}, \mathbf{u}, \mathbf{w}), i+1)\big) \tag{6.7}$$

We similarly define a function Q to represent variations about the nominal sequence $\{\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t\}$ and find its second-order expansion, dropping second-order dynamics Hessians if using iLQR rather than DDP. Note, in the equations that follow, time dependence is not explicitly notated for conciseness in notation:

$$Q(\delta\mathbf{x},\delta\mathbf{u},\delta\mathbf{w}) \equiv [\ell(\mathbf{x}+\delta\mathbf{x},\mathbf{u}+\delta\mathbf{u},\mathbf{w}+\delta\mathbf{w}) - \ell(\mathbf{x},\mathbf{u},\mathbf{w})] + \\ [V(f(\mathbf{x}+\delta\mathbf{x},\mathbf{u}+\delta\mathbf{u},\mathbf{w}+\delta\mathbf{w}),i+1) - V(f(\mathbf{x},\mathbf{u},\mathbf{w}),i+1)] \tag{6.8}$$

$$\approx \frac{1}{2}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\\ \delta\mathbf{w}\\ 1\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q}_{xx} & \mathbf{Q}_{xu} & \mathbf{Q}_{xw} & \mathbf{Q}_{x}\\ \mathbf{Q}_{ux} & \mathbf{Q}_{uu} & \mathbf{Q}_{uw} & \mathbf{Q}_{u}\\ \mathbf{Q}_{wx} & \mathbf{Q}_{wu} & \mathbf{Q}_{ww} & \mathbf{Q}_{w}\\ \mathbf{Q}_x^\top & \mathbf{Q}_u^\top & \mathbf{Q}_w^\top & 0\end{bmatrix} \begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\\ \delta\mathbf{w}\\ 1\end{bmatrix} \tag{6.9}$$

where the first- and second-order coefficient matrices are defined as follows:

$$\begin{aligned} \mathbf{Q}_x &= \boldsymbol{\ell}_x + \mathbf{f}_x^\top V'_x & \mathbf{Q}_{xx} &= \boldsymbol{\ell}_{xx} + \mathbf{f}_x^\top V'_{xx}\mathbf{f}_x + V'_x\mathbf{f}_{xx}\\ \mathbf{Q}_u &= \boldsymbol{\ell}_u + \mathbf{f}_u^\top V'_x & \mathbf{Q}_{uu} &= \boldsymbol{\ell}_{uu} + \mathbf{f}_u^\top V'_{xx}\mathbf{f}_u + V'_x\mathbf{f}_{uu}\\ \mathbf{Q}_w &= \boldsymbol{\ell}_w + \mathbf{f}_w^\top V'_x & \mathbf{Q}_{ww} &= \boldsymbol{\ell}_{ww} + \mathbf{f}_w^\top V'_{xx}\mathbf{f}_w + V'_x\mathbf{f}_{ww}\\ \mathbf{Q}_{uw} &= \boldsymbol{\ell}_{uw} + \mathbf{f}_u^\top V'_{xx}\mathbf{f}_w + V'_x\mathbf{f}_{uw} & \mathbf{Q}_{xu} &= \boldsymbol{\ell}_{xu} + \mathbf{f}_x^\top V'_{xx}\mathbf{f}_u + V'_x\mathbf{f}_{xu}\\ \mathbf{Q}_{xw} &= \boldsymbol{\ell}_{xw} + \mathbf{f}_x^\top V'_{xx}\mathbf{f}_w + V'_x\mathbf{f}_{xw} \end{aligned}$$

A second-order local model of the value function used in the equations above is obtained through iterative updates. The terminal value function is first initialized by setting $V(\mathbf{x},N) = \ell_N(\mathbf{x}_N)$. The value function V and its first- and second-order derivatives V_x and V_{xx} are then calculated iteratively at each update using the following update formulas derived in [65]:

$$V(k) = V(k+1) - \mathbf{Q}_u(k)\mathbf{Q}_{uu}^{-1}(k)\mathbf{Q}_u(k) - \mathbf{Q}_w(k)\mathbf{Q}_{ww}^{-1}(k)\mathbf{Q}_w(k) \tag{6.10}$$

$$V_x(k) = \mathbf{Q}_x(k) - \mathbf{Q}_u(k)\mathbf{Q}_{uu}^{-1}(k)\mathbf{Q}_{ux}(k) - \mathbf{Q}_w(k)\mathbf{Q}_{ww}^{-1}(k)\mathbf{Q}_{wx}(k) \tag{6.11}$$

$$V_{xx}(i) = \mathbf{Q}_{xx}(k) - \mathbf{Q}_{xu}(k)\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ux}(k) - \mathbf{Q}_{xw}(k)\mathbf{Q}_{ww}^{-1}(k)\mathbf{Q}_{wx}(k) \tag{6.12}$$

By setting the gradients to zero with respect to the input and disturbance deviations, respectively, we obtain the optimal trajectory perturbations at each iteration in order to minimize and maximize the second-order expansion:

$$\delta\mathbf{u}^* = \underset{\delta\mathbf{u}}{\operatorname{argmin}}\ Q(\delta\mathbf{x},\delta\mathbf{u},\delta\mathbf{w}) = -\mathbf{Q}_{uu}^{-1}(\mathbf{Q}_{ux}\delta\mathbf{x} + \mathbf{Q}_{uw}\delta\mathbf{w} + \mathbf{Q}_u) \tag{6.13}$$

$$\delta\mathbf{w}^* = \underset{\delta\mathbf{w}}{\operatorname{argmax}}\ Q(\delta\mathbf{x},\delta\mathbf{u},\delta\mathbf{w}) = -\mathbf{Q}_{ww}^{-1}(\mathbf{Q}_{wx}\delta\mathbf{x} + \mathbf{Q}_{wu}\delta\mathbf{u} + \mathbf{Q}_w) \tag{6.14}$$

In this minimax iterative dynamic game (iDG) formulation, it is clear from the above equations that the optimal input of the policy agent is dependent on the adversarial disturbance, and vice-versa. Substituting and solving the system of equations above, we derive the final control/disturbance perturbations as a function of the state perturbations $\delta\mathbf{x}$:

$$\delta \mathbf{u}^* = -\mathbf{Q}_{uu}^{-1}(\mathbf{Q}_{ux}\delta\mathbf{x} + \mathbf{Q}_{uw}\delta\mathbf{w} + \mathbf{Q}_u) \tag{6.15}$$

$$= (\mathbf{I} - \mathbf{Q}_{uu}^{-1}\mathbf{Q}_{uw}\mathbf{Q}_{ww}^{-1}\mathbf{Q}_{wu})^{-1} \cdot \tag{6.16}$$

$$\left[- \mathbf{Q}_{uu}^{-1}(\mathbf{Q}_u - \mathbf{Q}_{uw}\mathbf{Q}_{ww}^{-1}\mathbf{Q}_w) - \mathbf{Q}_{uu}^{-1}(\mathbf{Q}_{ux} - \mathbf{Q}_{uw}\mathbf{Q}_{ww}^{-1}\mathbf{Q}_{wx})\delta\mathbf{x}\right]$$

$$\delta \mathbf{w}^* = -\mathbf{Q}_{ww}^{-1}(\mathbf{Q}_{wx}\delta\mathbf{x} + \mathbf{Q}_{wu}\delta\mathbf{u} + \mathbf{Q}_w) \tag{6.17}$$

$$= (\mathbf{I} - \mathbf{Q}_{ww}^{-1}\mathbf{Q}_{wu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{uw})^{-1} \cdot \tag{6.18}$$

$$\left[- \mathbf{Q}_{ww}^{-1}(\mathbf{Q}_w - \mathbf{Q}_{wu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_u) - \mathbf{Q}_{ww}^{-1}(\mathbf{Q}_{wx} - \mathbf{Q}_{wu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ux})\delta\mathbf{x}\right]$$

If we define the following coefficient terms for convenience:

$$\mathbf{K}_u = (\mathbf{I} - \mathbf{Q}_{uu}^{-1}\mathbf{Q}_{uw}\mathbf{Q}_{ww}^{-1}\mathbf{Q}_{wu})^{-1}\mathbf{Q}_{uu}^{-1} \tag{6.19}$$

$$\mathbf{K}_w = (\mathbf{I} - \mathbf{Q}_{ww}^{-1}\mathbf{Q}_{wu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{uw})^{-1}\mathbf{Q}_{ww}^{-1} \tag{6.20}$$

$$\mathbf{g}_u = \mathbf{K}_u(\mathbf{Q}_{uw}\mathbf{Q}_{ww}^{-1}\mathbf{Q}_w - \mathbf{Q}_u) \tag{6.21}$$

$$\mathbf{g}_w = \mathbf{K}_w(\mathbf{Q}_{wu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_u - \mathbf{Q}_w) \tag{6.22}$$

$$\mathbf{G}_u = \mathbf{K}_u(\mathbf{Q}_{uw}\mathbf{Q}_{ww}^{-1}\mathbf{Q}_{wx} - \mathbf{Q}_{ux}) \tag{6.23}$$

$$\mathbf{G}_w = \mathbf{K}_w(\mathbf{Q}_{wu}\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ux} - \mathbf{Q}_{wx}) \tag{6.24}$$

then we may compactly write the optimal input and disturbance perturbations as a constant affine term plus linear feedback (reintroducing the time dependency explicitly here for clarity):

$$\delta\mathbf{u}^*(k) = \mathbf{G}_u(k)\delta\mathbf{x}(k) + \mathbf{g}_u(k) \tag{6.25}$$

$$\delta\mathbf{w}^*(k) = \mathbf{G}_w(k)\delta\mathbf{x}(k) + \mathbf{g}_w(k) \tag{6.26}$$

Importantly, the formulas above (namely Equation 6.19 and Equation 6.20) include critical corrections to account for original errata in [66] and combine techniques from [66],[65] towards application in a receding horizon control fashion.

In summary, we follow a similar iterative approach as described in Section 4.7:

- We begin by first selecting a nominal initial guess for the input/disturbance sequences $\{\hat{\mathbf{u}}_k\}$ and $\{\hat{\mathbf{w}}_k\}$ and forward propagate through the dynamics to obtain a nominal sequence of dynamic states $\{\hat{\mathbf{x}}_k\}$.
- Afterwards, local linear/quadratic models are obtained about the nominal trajectory for the nonlinear dynamics/cost function respectively, which we utilize in a backwards pass to update the local second-order model of the cost function.

- Optimal perturbations about the current iterations nominal input/disturbance sequences are obtained using Equations 6.25 and 6.26.
- Lastly, a backtracking linesearch scales the affine term in the resulting optimal deviations, as larger deviations may violate local linear model assumptions.

This iterative cycle of selecting a modified nominal input/disturbance trajectory is repeated, with incremental improvements obtained from the equations above, until the value function converges to a local saddle point and negligible cost improvements are made (see Algorithm 2).

Algorithm 2: Minimax Iterative Dynamic Games

```
Input: General penalty function $\ell_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k))$
       Dynamics function $f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k))$
       Trajectory horizon $N$
       Initial state $\mathbf{x}_{init}$
       Initial control sequence $\hat{\mathbf{U}} = \{\hat{\mathbf{u}}(0), ..., \hat{\mathbf{u}}(N-1)\}$
       Initial disturbance sequence $\hat{\mathbf{W}} = \{\hat{\mathbf{w}}(0), ..., \hat{\mathbf{w}}(N-1)\}$
Output: Control sequence $\mathbf{U} = \{\mathbf{u}(0), ..., \mathbf{u}(N-1)\}$
        Control sequence $\mathbf{W} = \{\mathbf{w}(0), ..., \mathbf{w}(N-1)\}$
        Input Feedback gains $\mathbf{G}_u(0), ..., \mathbf{G}_u(N-1)$
        Disturbance Feedback gains $\mathbf{G}_w(0), ..., \mathbf{G}_w(N-1)$
Initialize Value function and derivatives $\mathbf{V}(k), \mathbf{V}_{\mathbf{x}}(k), \mathbf{V}_{\mathbf{xx}}(k)$, with $\mathbf{V}(N) = \ell_N$
Obtain $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}(1), ..., \hat{\mathbf{x}}(N)\}$ by evolving according to nonlinear dynamics:
 $\hat{\mathbf{x}}(k+1) = f(\hat{\mathbf{x}}(k), \hat{\mathbf{u}}(k), \hat{\mathbf{w}}(k))$ with initial condition: $\hat{\mathbf{x}}(0) = \mathbf{x}_{init}$
while Not Converged do
    $\mathbf{F}_x(k) = \frac{\partial f}{\partial \mathbf{x}} f(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{w}}),\ \mathbf{F}_u(k) = \frac{\partial f}{\partial \mathbf{u}} f(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{w}}),\ \mathbf{F}_w(k) = \frac{\partial f}{\partial \mathbf{w}} f(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{w}})$
    Calculate derivatives of quadratic approximation $\mathbf{Q}(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ using Equation 6.9
    Solve for $\mathbf{G}_u(0), ..., \mathbf{G}_u(N-1)$, $\mathbf{g}_u(0), ..., \mathbf{g}_u(N-1)$ according to Equation 6.23
    Solve for $\mathbf{G}_w(0), ..., \mathbf{G}_w(N-1)$, $\mathbf{g}_w(0), ..., \mathbf{g}_w(N-1)$ according to Equation 6.24
    Update Value Function and derivatives according to Equations 6.10, 6.11, 6.12
    while cost is not improving do
        /* backtracking line search                                    */
        Obtain sequence $\mathbf{U}$ by propagating $\mathbf{u}(k) = \hat{\mathbf{u}}(k) + \mathbf{G}_u(k)\delta_{\mathbf{x}}(k) + \alpha \mathbf{g}_u(k)$
        Obtain sequence $\mathbf{W}$ by propagating $\mathbf{w}(k) = \hat{\mathbf{w}}(k) + \mathbf{G}_w(k)\delta_{\mathbf{x}}(k) + \alpha \mathbf{g}_w(k)$
        Find $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}(1), ..., \hat{\mathbf{x}}(N)\}$ by executing $\hat{\mathbf{x}}(k+1) = f(\hat{\mathbf{x}}(k), \hat{\mathbf{u}}(k), \hat{\mathbf{w}}(k))$
        Set $\hat{\mathbf{U}} = \mathbf{U}$
        Set $\hat{\mathbf{W}} = \mathbf{W}$
        Calculate new cost function using $\hat{\mathbf{X}}, \hat{\mathbf{U}}, \hat{\mathbf{W}}$
        $\alpha \leftarrow \alpha/2$
    end
end
```

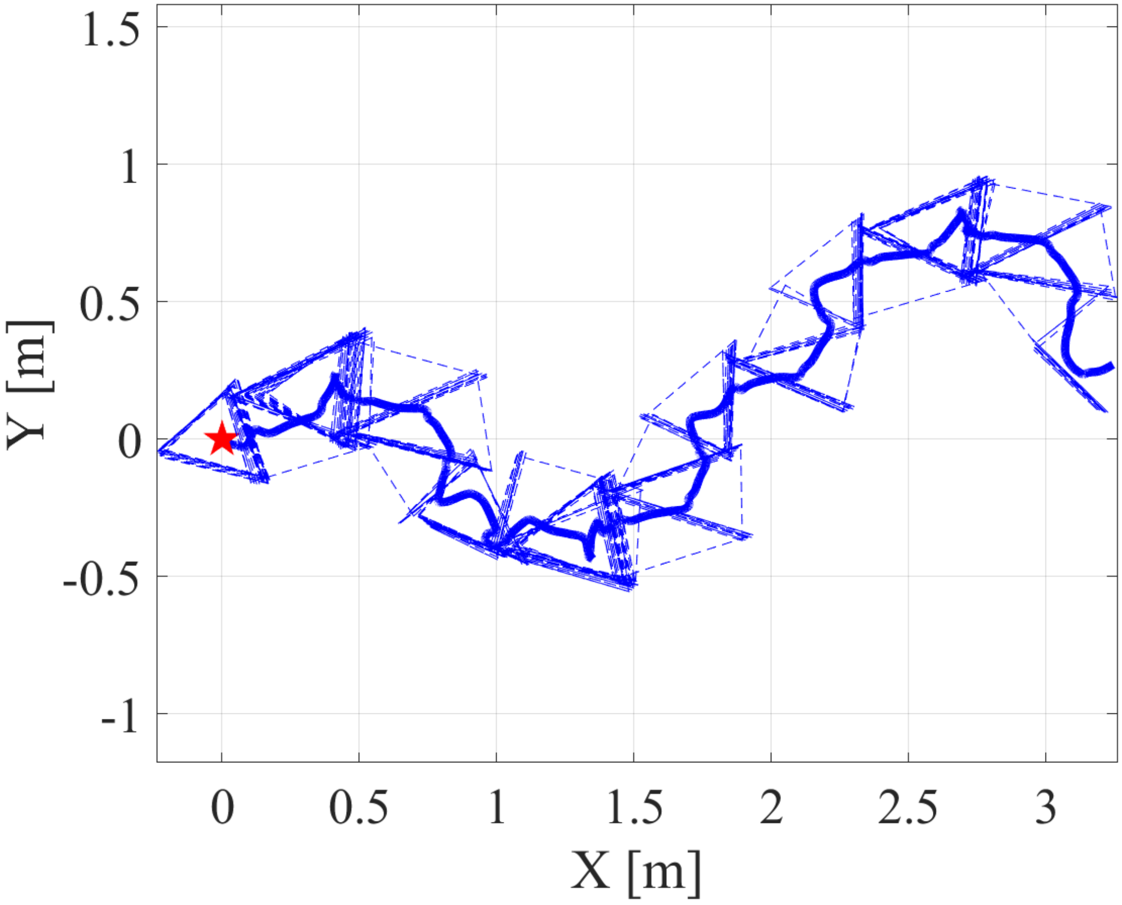


Figure 6.6: Trajectory tracking in the positive X-direction with a nominal controller and imperfect state estimates. Notice that while the general overall movement is in the correct direction, the robot missteps along the way.

6.4.1 Robust Minimax Control Combined with State Estimation

As alluded to in past sections, tensegrity robotics face an especially difficult challenge for motion planning and trajectory optimization due to the uncertainties which naturally arise as a result of compliance throughout the system in addition to noisy sensor measurements and imperfect models. Physical characteristics and material properties of the robot may vary significantly from model parameters solely due to manufacturing and assembly tolerances. Additionally, sensor fusion and optimal estimation approaches address sensor noise to a degree, but pragmatic applications of these control policies must be able to handle imperfect state estimates. In this section, we assess the performance of the previously described minimax controller in combination with state estimation in order to better understand performance capabilities of the robot in a more challenging context with imperfect information.

To this end, we simulated experiments applying the aforementioned robust minimax iDG method that expands upon the original motion planning approaches described in Chapter

4 and have demonstrated successful results. The results show that this robust control formulation enables successful rolling locomotion with adversarial disturbances/noise of up to 10% of the maximum control input magnitude (i.e., 1 cm/s control authority for the adversarial agent). More importantly, this ability to handle adversarial perturbations directly leads to being capable of continuous dynamic rolling even when given noisy imperfect state estimates. Using the same physical model parameters and sensor noise characteristics as outlined in Table 5.1 and Table 5.2, the robust minimax iDG approach was applied to a simulated robot with state estimation implemented using UKF. From the results shown in Figure 6.7, we illustrate that the the robust controller is significantly better at handling noisy state estimates, nearly matching the nominal performance of the iLQR controller with perfect full-state information, albeit at a slower conservative speed. In contrast, the nominal controller struggles to contend with noisy measurements (shown in green) and occasionally suffers from unexpected transitions in an undesired direction. One potential reason for this type of failure is the unique method of moving the tensegrity robot center of mass to force an unstable transition from one face to another. Inaccuracies in the shape of the supporting base polygon, the robot contact interactions with the ground, the pose of the robot, or any other multitude of factors may lead the robot into taking an action which it falsely believes is beneficial. In reality, inaccuracies cause the robot behavior to evolve in an unanticipated manner, and once the robot is placed in a precarious unstable state, the punctuated rolling motion continues through, with no means of control authority to stop (see Figure 6.6). Thus, while the minimax formulation is suboptimal in terms of purely rolling speed, this approach may be one way to address real-world uncertainty with rolling tensegrity robots.

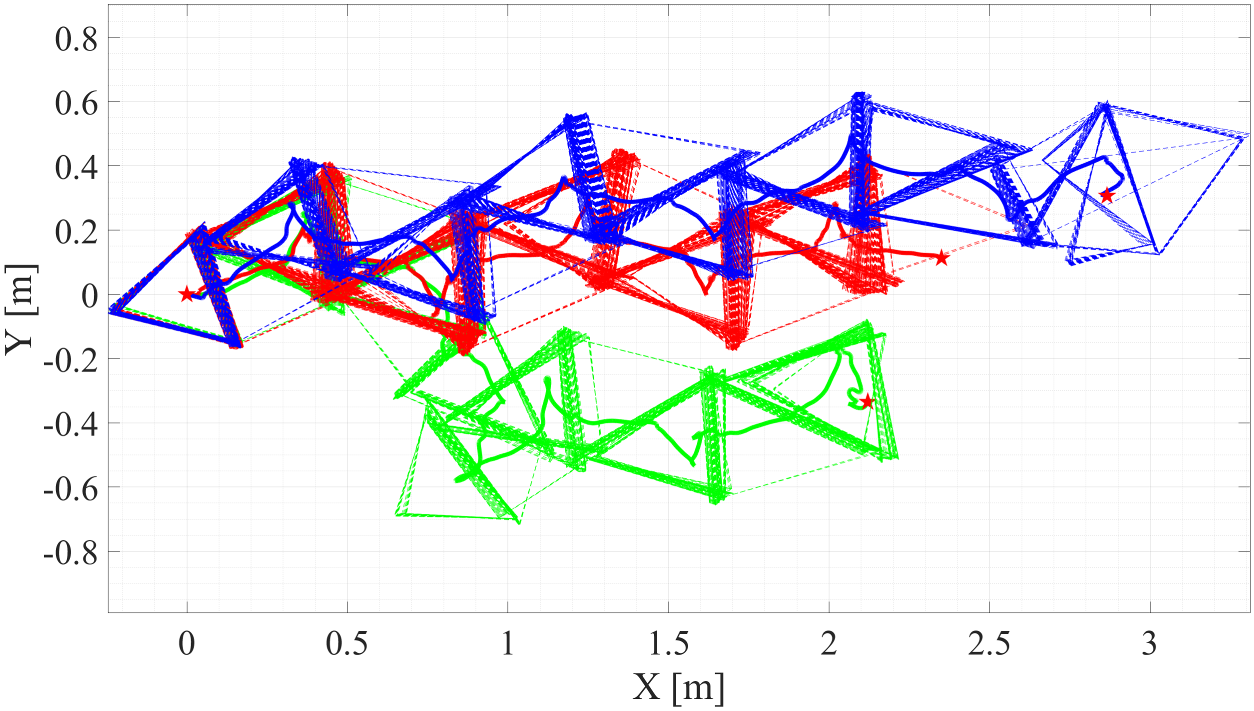


Figure 6.7: Comparison of rolling performance with/without state estimation and robust minimax motion planning. The three cases depicted are: perfect full state information with nominal iLQR controller (blue); noisy state estimates with UKF and nominal iLQR controller (green); noisy state estimates with UKF and robust minimax iDG controller (red). Note that the minimax controller does a better job at tracking the desired trajectory in the positive X direction, as compared to the nominal controller.

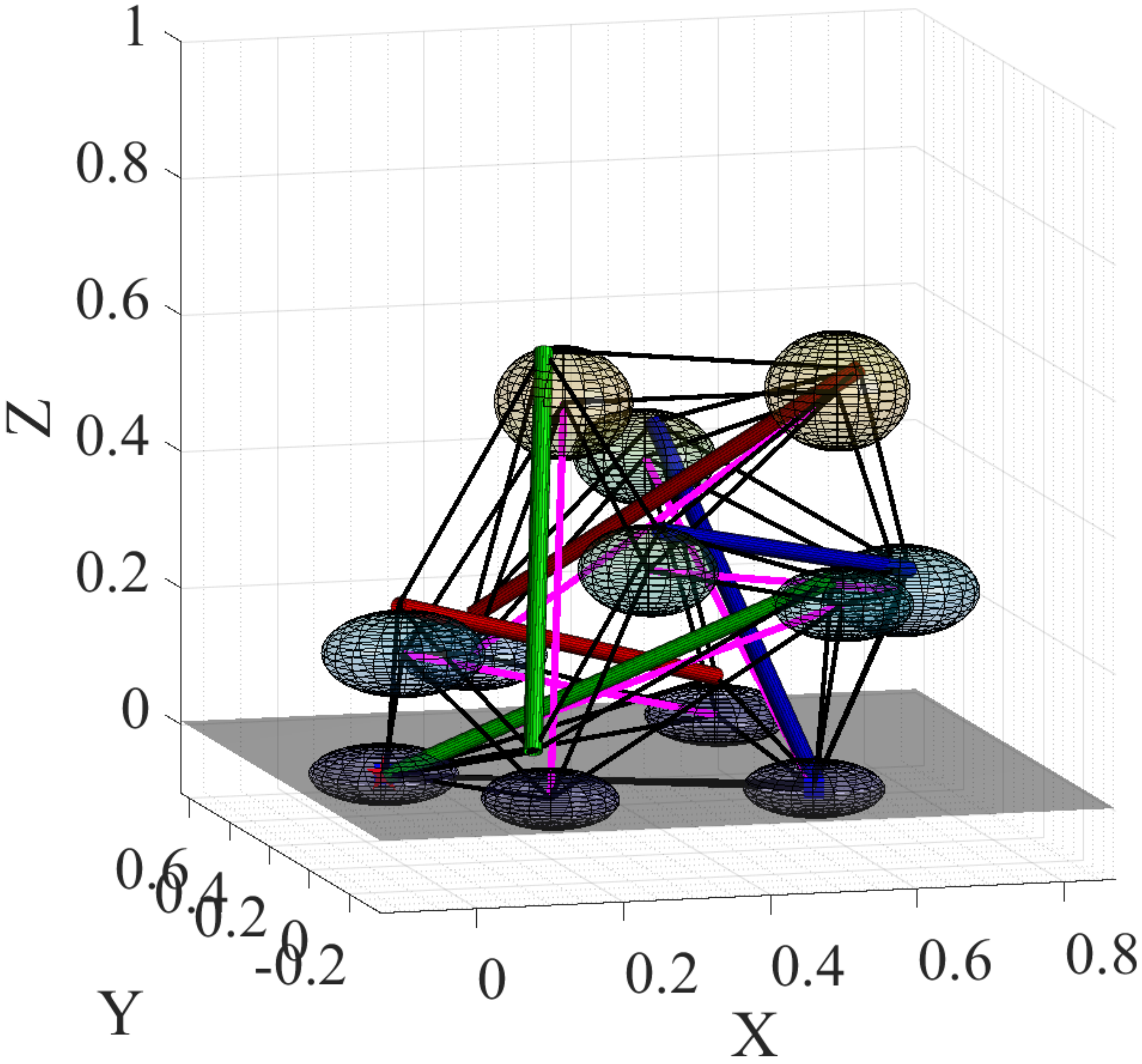


Figure 6.8: Accuracy of the state estimates for the robot pose can vary widely over time during dynamic rolling. In particular, during the moments immediately after a rolling motion impact (shown here), the rod position estimates (magenta) deviate significantly from the ground-truth, momentarily causing the robot to take control actions it falsely assumes to be optimal.

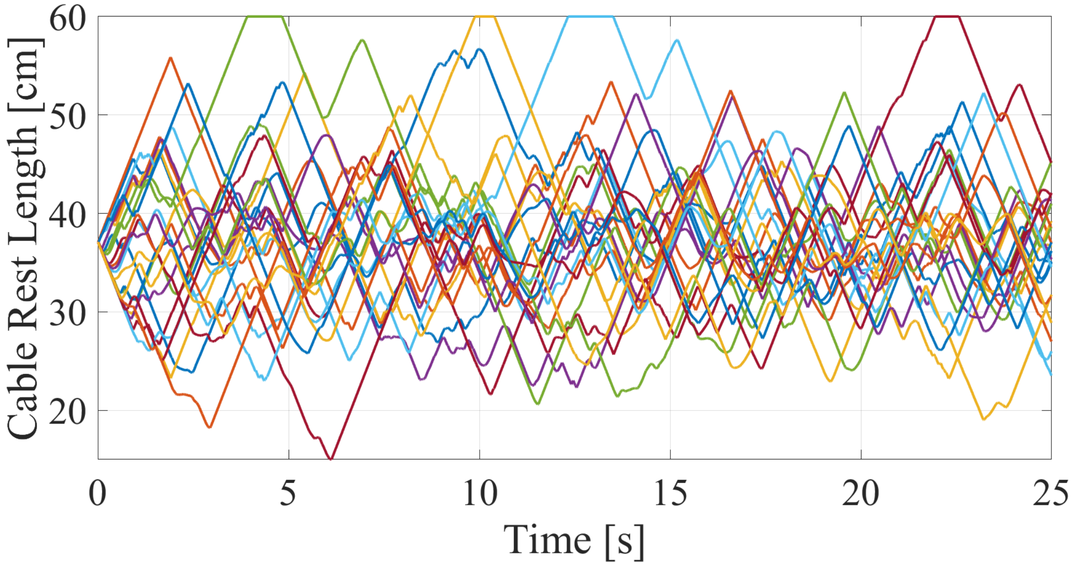


Figure 6.9: Example of cable rest length evolution over time, for a robot using nominal iLQR as a control policy and state estimation through UKF.

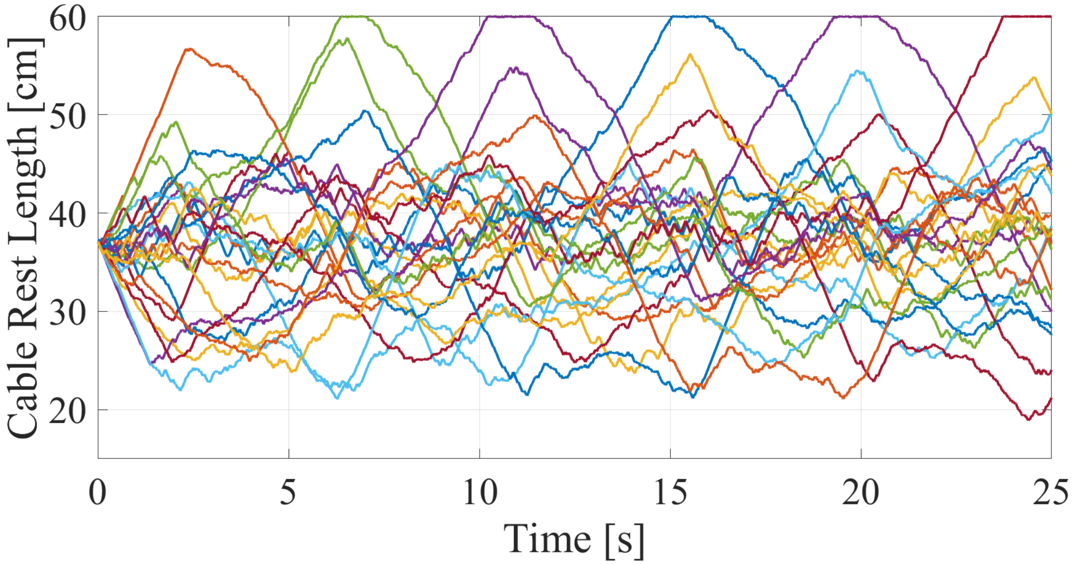


Figure 6.10: Example of cable rest length evolution over time, for a robot using robust minimax iDG as a control policy and state estimation through UKF. Note the less vertical slopes (unsaturated control inputs) and rounded transitions in comparison to the nominal controller inputs depicted in Figure 6.9.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

Tensegrity robots are a novel and emergent type of soft-robotics that have uniquely advantageous mechanical properties of being lightweight and passively compliant, giving them great potential for applications in uncertain environments or co-robotic applications where there may be unexpected contact with their surroundings. Before they can be considered as robust hardware platforms, however, their autonomous capabilities must be further developed for fast and reliable mobility. Unfortunately, the same system-level compliance that gives tensegrities such great mechanical properties also introduces many complexities when dealing with optimal feedback control of these systems. This work strives to elucidate and address many of the practical challenges surrounding optimal control and motion planning for dynamic mobile tensegrity robots.

This dissertation presented minimal and point mass dynamics model representations that allow for consideration of tensegrities in an idealized manner and for reasoning about approaches that may accurately simulate dynamic tensegrities under contact-rich rolling locomotion. Building on these dynamics, this work introduced an open-source modular simulation framework created specifically for investigating and evaluating novel tensegrity control and state estimation. Furthermore, this paper outlined two optimization-based approaches that are able to generate optimal state-action trajectories for complex nonlinear tensegrities. The given techniques leverage the well-structured dynamics of Class-1 tensegrities and may be extended to new topologies beyond the six-bar spherical tensegrity highlighted in this paper. Several novel tensegrity topologies with varying hardware configurations and degrees of control authority are described, and preliminary analyses demonstrate their potential for future hardware iterations. Moreover, this paper explored the complementary problem of state estimation. Given a commonality of proposed tensegrity sensors, nonlinear Kalman Filter methods are evaluated in the context of tensegrity dynamic rolling mobility. Finally, this work introduced how deep learning can be utilized with optimal MPC trajectories to create highly-expressive end-to-end contextual policies for real-time directed rolling, even for partially observable systems. A minimax robust control approach is also outlined, with simulation results that provide some insights into how uncertainty in tensegrity systems may be addressed. Combined, the motion planning and state estimation strategies introduced in this research help advance the progression of this novel technology towards becoming fully viable mobile robotic platforms.

7.2 Future Work

7.2.1 Hardware Testing

Squishy Robotics plans to implement the techniques described in this report with their latest *MR3* mobile robot hardware iteration. The current hardware prototype allows for real-time control and state estimation using the actuators and sensors described in this paper, with new electronics for improved reliability during hardware experiments. Furthermore, the hardware was designed with paired-cable actuation (described in Section 4.9.3) in mind, so we look forward to evaluating energy-efficient tensegrity locomotion and state estimation in hardware.

7.2.2 Transfer Learning to New Environments

With the framework for tensegrity trajectory optimization set in place, it is now possible to generate locally optimal state-action trajectories for spherical tensegrities with varying physical parameters using optimization-based approaches. An interesting next step would be to use policy-gradient and reinforcement learning methods (e.g., natural policy gradients, trust region policy optimization, and proximal policy optimization) for transfer learning to new environments (e.g., rough or inclined terrain rather than flat ground). This would allow for policies to be first generated offline with the approaches in this paper using nominal hardware parameters in simulation. Afterwards, learning approaches can enable the neural network policy to later learn from new online hardware sensor observation data through reinforcement learning methods, correcting for model mismatch and other disturbances to improve in real-time.

7.2.3 Tensegrity Perception, Localization, and Mapping

The topics presented in this work touch on a key area of autonomous systems, that of motion planning and control. Beyond this, however, autonomous tensegrity robots must also be able to tackle critical challenges in perception, localization, and mapping. Currently, some tensegrity hardware designs from Squishy Robotics feature sensor payloads with camera vision capabilities, which we may leverage in future work for visual-inertial odometry. Notably, new techniques and methods for tensegrity perception must be able to handle the intrinsic compliance and oscillatory behavior of the system and 360-degree requirements for rolling ground mobility. Such advances could also be beneficial in improving the state estimation techniques for tensegrities described in this work. Beyond this, simultaneous localization and mapping offers a compelling value proposition for the rapidly-deployable air-dropped tensegrity sensor robots that are currently being developed.

Bibliography

[1] Adrian K Agogino, Vytas SunSpiral, and David Atkinson. “Super Ball Bot-structures for planetary landing and exploration”. In: (2018).

[2] Adrian Agogino, Vytas SunSpiral, and David Atkinson. “Super Ball Bot-structures for planetary landing and exploration”. In: *NASA Innovative Advanced Concepts (NIAC) Program, Final Report* (2013), pp. 5–13.

[3] Sergei M Aseev and Arkady V Kryazhimskiy. “The Pontryagin maximum principle and transversality conditions for a class of optimal control problems with infinite time horizons”. In: *SIAM Journal on Control and Optimization* 43.3 (2004), pp. 1094–1119.

[4] Leya Breanna Baltaxe-Admony, Ash S Robbins, Erik A Jung, Steven Lessard, Mircea Teodorescu, Vytas SunSpiral, and Adrian Agogino. “Simulating the human shoulder through active tensegrity structures”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 50183. American Society of Mechanical Engineers. 2016, V006T09A027.

[5] Tamer Başar and Pierre Bernhard. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

[6] Eduardo Bayo. “A finite-element approach to control the end-point motion of a single-link flexible robot”. In: *Journal of Robotic systems* 4.1 (1987), pp. 63–75.

[7] Randal W Beard, George N Saridis, and John T Wen. “Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation”. In: *Automatica* 33.12 (1997), pp. 2159–2177.

[8] Alberto Bemporad, Francesco Borrelli, Manfred Morari, et al. “Model predictive control based on linear programming˜ the explicit solution”. In: *IEEE transactions on automatic control* 47.12 (2002), pp. 1974–1985.

[9] Alberto Bemporad and Manfred Morari. “Robust model predictive control: A survey”. In: *Robustness in identification and control*. Springer, 1999, pp. 207–226.

[10] John T Betts. “Survey of numerical methods for trajectory optimization”. In: *Journal of guidance, control, and dynamics* 21.2 (1998), pp. 193–207.

[11] Gilbert A Bliss. “Lectures on the Calculus of Variations”. In: (1946).

[12] Francesco Borrelli, Paolo Falcone, Tamas Keviczky, Jahan Asgari, and Davor Hrovat. "MPC-based approach to active steering for autonomous vehicle systems". In: *International journal of vehicle autonomous systems* 3.2-4 (2005), pp. 265–291.

[13] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory.* SIAM, 1994.

[14] J. Bruce, K. Caluwaerts, A. Iscen, A. P. Sabelhaus, and V. SunSpiral. "Design and evolution of a modular tensegrity robot platform". In: *2014 IEEE International Conference on Robotics and Automation (ICRA).* May 2014, pp. 3483–3489. DOI: `10.1109/ICRA.2014.6907361`.

[15] Jonathan Bruce. "Design, building, testing, and control of superball: A tensegrity robot to enable new forms of planetary exploration". PhD thesis. UC Santa Cruz, 2016.

[16] Jeff B Burl. *Linear optimal control: H (2) and H (Infinity) methods.* Addison-Wesley Longman Publishing Co., Inc., 1998.

[17] Ken Caluwaerts, Jonathan Bruce, Jeffrey M Friesen, and Vytas SunSpiral. "State estimation for tensegrity robots". In: *2016 IEEE International Conference on Robotics and Automation (ICRA).* IEEE. 2016, pp. 1860–1865.

[18] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control.* Springer Science & Business Media, 2013.

[19] Peter J Campo and Manfred Morari. "Robust model predictive control". In: *1987 American control conference.* IEEE. 1987, pp. 1021–1026.

[20] B Cera, Anthony A Thompson, and Alice M Agogino. "Energy-Efficient Locomotion Strategies and Performance Benchmarks using Point Mass Tensegrity Dynamics". In: *IROS 2019* (2019).

[21] Brian Cera and Alice M Agogino. "Multi-Cable Rolling Locomotion with Spherical Tensegrities Using Model Predictive Control and Deep Learning". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2018, pp. 1–9.

[22] Wai Leung Chan, Diego Arbelaez, Frederic Bossens, and Robert E Skelton. "Active vibration control of a three-stage tensegrity structure". In: *Smart Structures and Materials 2004: Damping and Isolation.* Vol. 5386. International Society for Optics and Photonics. 2004, pp. 340–346.

[23] L. H. Chen, B. Cera, E. L. Zhu, R. Edmunds, F. Rice, A. Bronars, E. Tang, S. R. Malekshahi, O. Romero, A. K. Agogino, and A. M. Agogino. "Inclined surface locomotion strategies for spherical tensegrity robots". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Sept. 2017, pp. 4976–4981. DOI: `10.1109/IROS.2017.8206380`.

[24] Lee-Huang Chen, Kyunam Kim, Ellande Tang, Kevin Li, Richard House, Edward Liu Zhu, Kimberley Fountain, Alice M Agogino, Adrian Agogino, Vytas Sunspiral, et al. "Soft spherical tensegrity robot design using rod-centered actuation and control". In: *Journal of Mechanisms and Robotics* 9.2 (2017), p. 025001.

[25] Joono Cheong, Robert E Skelton, and Youngsu Cho. "A numerical algorithm for tensegrity dynamics with non-minimal coordinates". In: *Mechanics Research Communications* 58 (2014), pp. 46–52.

[26] Paolo Falcone, H Eric Tseng, Jahan Asgari, Francesco Borrelli, and Davor Hrovat. "Integrated braking and steering model predictive control approach in autonomous vehicles". In: *IFAC Proceedings Volumes* 40.10 (2007), pp. 273–278.

[27] Jeffrey M Friesen, Paul Glick, Michael Fanton, Pavlo Manovi, Alexander Xydes, Thomas Bewley, and Vytas Sunspiral. "The second generation prototype of a duct climbing tensegrity robot, DuCTTv2". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 2123–2128.

[28] Jeffrey Friesen, Alexandra Pogue, Thomas Bewley, Mauricio de Oliveira, Robert Skelton, and Vytas Sunspiral. "DuCTT: A tensegrity robot for exploring duct systems". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 4222–4228.

[29] Robert J Full and Michael S Tu. "Mechanics of a rapid running insect: two-, four-and six-legged locomotion". In: *Journal of Experimental Biology* 156.1 (1991), pp. 215–231.

[30] Hiroshi Furuya. "Concept of deployable tensegrity structures in space application". In: *International Journal of Space Structures* 7.2 (1992), pp. 143–151.

[31] Carlos E Garcia, David M Prett, and Manfred Morari. "Model predictive control: theory and practice—a survey". In: *Automatica* 25.3 (1989), pp. 335–348.

[32] Roland Geraerts and Mark H Overmars. "A comparative study of probabilistic roadmap planners". In: *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 43–57.

[33] Raman Goyal and Robert E Skelton. "Tensegrity system dynamics with rigid bars and massive strings". In: *Multibody System Dynamics* 46.3 (2019), pp. 203–228.

[34] Ariel Hanaor. "Aspects of design of double-layer tensegrity domes". In: *International Journal of Space Structures* 7.2 (1992), pp. 101–113.

[35] Peter E Hart, Nils J Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.

[36] Shinichi Hirai and Ryo Imuta. "Dynamic simulation of six-strut tensegrity robot rolling". In: *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2012, pp. 198–204.

[37] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. “Flop and roll: Learning robust goal-directed locomotion for a tensegrity robot”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 2236–2243.

[38] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. “Learning to control complex tensegrity robots”. In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. 2013, pp. 1193–1194.

[39] Atil Iscen, Ken Caluwaerts, Jonathan Bruce, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. “Learning tensegrity locomotion using open-loop control signals and coevolutionary algorithms”. In: *Artificial life* 21.2 (2015), pp. 119–140.

[40] Mehdi Jalalmaab, Bariş Fidan, Soo Jeon, and Paolo Falcone. “Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 843–848.

[41] Simon J Julier and Jeffrey K Uhlmann. “A new extension of the Kalman filter to nonlinear systems”. In: *Int. symp. aerospace/defense sensing, simul. and controls*. Vol. 3. 26. Orlando, FL. 1997, pp. 182–193.

[42] Simon J Julier and Jeffrey K Uhlmann. “New extension of the Kalman filter to nonlinear systems”. In: *Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. International Society for Optics and Photonics. 1997, pp. 182–193.

[43] Simon J Julier and Jeffrey K Uhlmann. “Unscented filtering and nonlinear estimation”. In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422.

[44] Rudolph E Kalman and Richard S Bucy. “New results in linear filtering and prediction theory”. In: (1961).

[45] Rudolph Emil Kalman. “A new approach to linear filtering and prediction problems”. In: (1960).

[46] Narongsak Kanchanasaratool and Darrell Williamson. “Modelling and control of class NSP tensegrity structures”. In: *International Journal of Control* 75.2 (2002), pp. 123–139.

[47] Narongsak Kanchanasaratool and Darrell Williamson. “Motion control of a tensegrity platform”. In: *Communications in Information and systems* 2.3 (2002), pp. 299–324.

[48] Kyunam Kim, Adrian K Agogino, Aliakbar Toghyan, Deaho Moon, Laqshya Taneja, and Alice M Agogino. “Robust learning of tensegrity robot control for locomotion through form-finding”. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE. 2015, pp. 5824–5831.

[49] Kyunam Kim, Lee-Huang Chen, Brian Cera, Mallory Daly, Edward Zhu, Julien Despois, Adrian K Agogino, Vytas SunSpiral, and Alice M Agogino. "Hopping and rolling locomotion with spherical tensegrity robots". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE. 2016, pp. 4369–4376.

[50] G Kloosterman, Rudolf MJ van Damme, Antonius H van den Boogaard, and J Huetink. "A geometrical-based contact algorithm using a barrier method". In: *International Journal for Numerical Methods in Engineering* 51.7 (2001), pp. 865–882.

[51] Byron Franklin Knight. "Deployable antenna kinematics using tensegrity structure design". PhD thesis. University of Florida Gainesville, FL, 2000.

[52] Yuusuke Koizumi, Mizuho Shibata, and Shinichi Hirai. "Rolling tensegrity driven by pneumatic soft actuators". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 1988–1993.

[53] James J Kuffner and Steven M LaValle. "RRT-connect: An efficient approach to single-query path planning". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 2. IEEE. 2000, pp. 995–1001.

[54] Felipe Kuhne, Walter Fetter Lages, and J Gomes da Silva Jr. "Model predictive control of a mobile robot using linearization". In: *Proceedings of mechatronics and robotics*. Citeseer. 2004, pp. 525–530.

[55] Wilbur Langson, Ioannis Chryssochoos, SV Raković, and David Q Mayne. "Robust model predictive control using tubes". In: *Automatica* 40.1 (2004), pp. 125–133.

[56] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. "End-to-end training of deep visuomotor policies". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373.

[57] Tuanjie Li and Yujuan Ma. "Robust vibration control of flexible tensegrity structure via μ synthesis". In: *Structural control and health monitoring* 20.2 (2013), pp. 173–186.

[58] Weiwei Li and Emanuel Todorov. "Iterative linear quadratic regulator design for nonlinear biological movement systems." In: *ICINCO (1)*. 2004, pp. 222–229.

[59] Johan Lofberg. "YALMIP: A toolbox for modeling and optimization in MATLAB". In: *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE. 2004, pp. 284–289.

[60] Tomas Lozano-Perez. "Spatial planning: A configuration space approach". In: *Autonomous robot vehicles*. Springer, 1990, pp. 259–271.

[61] Jianlan Luo, Riley Edmunds, Franklin Rice, and Alice M Agogino. "Tensegrity robot locomotion under limited sensory inputs via deep reinforcement learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6260–6267.

[62] Milenko Masic, Robert E Skelton, and Philip E Gill. "Algebraic tensegrity form-finding". In: *International Journal of Solids and Structures* 42.16-17 (2005), pp. 4833–4858.

[63] David Mayne. "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems". In: *International Journal of Control* 3.1 (1966), pp. 85–95.

[64] Nik A Melchior and Reid Simmons. "Particle RRT for path planning with uncertainty". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 1617–1624.

[65] Jun Morimoto and Christopher G Atkeson. "Minimax differential dynamic programming: An application to robust biped walking". In: *Advances in neural information processing systems*. 2003, pp. 1563–1570.

[66] Olalekan Ogunmolu, Nicholas Gans, and Tyler Summers. "Minimax iterative dynamic game: Application to nonlinear robot control tasks". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 6919–6925.

[67] Andrew Packard and John Doyle. "The complex structured singular value". In: *Automatica* 29.1 (1993), pp. 71–109.

[68] Chandana Paul, John William Roberts, Hod Lipson, and FJ Valero Cuevas. "Gait production in a tensegrity based robot". In: *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005.* IEEE. 2005, pp. 216–222.

[69] Chandana Paul, Francisco J Valero-Cuevas, and Hod Lipson. "Design and control of tensegrity robots for locomotion". In: *IEEE Transactions on Robotics* 22.5 (2006), pp. 944–957.

[70] Shige Peng. "A generalized dynamic programming principle and Hamilton-Jacobi-Bellman equation". In: *Stochastics: An International Journal of Probability and Stochastic Processes* 38.2 (1992), pp. 119–134.

[71] Shige Peng. "Stochastic hamilton–jacobi–bellman equations". In: *SIAM Journal on Control and Optimization* 30.2 (1992), pp. 284–304.

[72] Gabriel Piras, WL Cleghorn, and JK Mills. "Dynamic finite-element analysis of a planar high-speed, high-precision parallel manipulator with flexible links". In: *Mechanism and machine theory* 40.7 (2005), pp. 849–862.

[73] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. "Epopt: Learning robust neural network policies using model ensembles". In: *arXiv preprint arXiv:1610.01283* (2016).

[74] James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design*. Nob Hill Pub. Madison, Wisconsin, 2009.

[75] Julian J Rimoli. "A reduced-order model for the dynamic and post-buckling behavior of tensegrity structures". In: *Mechanics of Materials* 116 (2018), pp. 146–157.

[76] Julian J Rimoli. "On the impact tolerance of tensegrity-based planetary landers". In: *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2016, p. 1511.

[77] Julian J Rimoli and Raj Kumar Pal. "Mechanical response of 3-dimensional tensegrity lattices". In: *Composites Part B: Engineering* 115 (2017), pp. 30–42.

[78] Murray Rosenblatt. "A central limit theorem and a strong mixing condition". In: *Proceedings of the National Academy of Sciences of the United States of America* 42.1 (1956), p. 43.

[79] Andrew P Sabelhaus, Jonathan Bruce, Ken Caluwaerts, Pavlo Manovi, Roya Fallah Firoozi, Sarah Dobi, Alice M Agogino, and Vytas SunSpiral. "System design and locomotion of SUPERball, an untethered tensegrity robot". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 2867–2873.

[80] Andrew P Sabelhaus, Hao Ji, Patrick Hylton, Yakshu Madaan, ChanWoo Yang, Alice M Agogino, Jeffrey Friesen, and Vytas SunSpiral. "Mechanism design and simulation of the ULTRA spine: a tensegrity robot". In: *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. 2015, V05AT08A059–V05AT08A059.

[81] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[82] M Shibata and S Hirai. "Rolling locomotion of deformable tensegrity structure". In: *Mobile Robotics: Solutions and Challenges*. World Scientific, 2010, pp. 479–486.

[83] Mizuho Shibata and Shinichi Hirai. "Moving strategy of tensegrity robots with semiregular polyhedral body". In: *Emerging Trends In Mobile Robotics*. World Scientific, 2010, pp. 359–366.

[84] Mizuho Shibata, Fumio Saijyo, and Shinichi Hirai. "Crawling by body deformation of tensegrity structure robots". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 4375–4380.

[85] David H Shim, H Jin Kim, and Shankar Sastry. "Decentralized nonlinear model predictive control of multiple flying robots". In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475).* Vol. 4. IEEE. 2003, pp. 3621–3626.

[86] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons, 2006.

[87] Robert E Skelton, Rajesh Adhikari, J-P Pinaud, Waileung Chan, and JW Helton. "An introduction to the mechanics of tensegrity structures". In: *Proceedings of the 40th IEEE conference on decision and control (Cat. No. 01CH37228).* Vol. 5. IEEE. 2001, pp. 4254–4259.

[88] Robert E Skelton and Mauricio C de Oliveira. *Tensegrity systems.* Vol. 1. Springer, 2009.

[89] Robert E Skelton, Jean Paul Pinaud, and D Lewis Mingori. "Dynamics of the shell class of tensegrity structures". In: *Journal of the Franklin Institute* 338.2-3 (2001), pp. 255–320.

[90] Adrian FM Smith and Alan E Gelfand. "Bayesian statistics without tears: a sampling–resampling perspective". In: *The American Statistician* 46.2 (1992), pp. 84–88.

[91] Cornel Sultan, Martin Corless, and Robert E Skelton. "Linear dynamics of tensegrity structures". In: *Engineering Structures* 24.6 (2002), pp. 671–685.

[92] Changyin Sun, Wei He, and Jie Hong. "Neural network control of a flexible robotic manipulator using the lumped spring-mass model". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.8 (2016), pp. 1863–1874.

[93] Vytas SunSpiral, Adrian Agogino, and David Atkinson. "Super Ball Bot-Structures for Planetary Landing and Exploration, NIAC Phase 2 Final Report". In: (2015).

[94] Peter Swerling. "Modern state estimation methods from the viewpoint of the method of least squares". In: *IEEE Transactions on Automatic Control* 16.6 (1971), pp. 707–719.

[95] Nico Tauchnitz. "The Pontryagin maximum principle for nonlinear optimal control problems with infinite horizon". In: *Journal of Optimization Theory and Applications* 167.1 (2015), pp. 27–48.

[96] Rex J Theodore and Ashitava Ghosal. "Comparison of the assumed modes and finite element models for flexible multilink manipulators". In: *The International journal of robotics research* 14.2 (1995), pp. 91–111.

[97] Sebastian Thrun. "Probabilistic robotics". In: *Communications of the ACM* 45.3 (2002), pp. 52–57.

[98] AG Tibert and Sergio Pellegrino. "Review of form-finding methods for tensegrity structures". In: *International Journal of Space Structures* 18.4 (2003), pp. 209–223.

[99] Gunnar Tibert. “Deployable tensegrity structures for space applications”. PhD thesis. KTH, 2002.

[100] Brian R Tietz, Ross W Carnahan, Richard J Bachmann, Roger D Quinn, and Vytas SunSpiral. “Tetraspine: Robust terrain handling on a tensegrity robot using central pattern generators”. In: *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE. 2013, pp. 261–267.

[101] Emanuel Todorov and Weiwei Li. “A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems”. In: *Proceedings of the 2005, American Control Conference, 2005.* IEEE. 2005, pp. 300–306.

[102] Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric A Wan. “The unscented particle filter”. In: *Advances in neural information processing systems.* 2001, pp. 584–590.

[103] Rudolph Van Der Merwe and Eric A Wan. “The square-root unscented Kalman filter for state and parameter-estimation”. In: *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221).* Vol. 6. IEEE. 2001, pp. 3461–3464.

[104] Qian Wang and Robert F Stengel. “Robust control of nonlinear systems with parametric uncertainty”. In: *Automatica* 38.9 (2002), pp. 1591–1599.

[105] Greg Welch, Gary Bishop, et al. “An introduction to the Kalman filter”. In: (1995).

[106] Andrew Witkin. “Physically Based Modeling–Constraint Dynamics”. In: *ACM SIGGRAPH 2001 Course Notes* (2001).

[107] Anders Sunde Wroldsen. “Modelling and control of tensegrity structures”. In: *Diss. Norwegian University of Science and Technology* (2007).

[108] G Zavarise, P Wriggers, and BA Schrefler. “A method for solving contact problems”. In: *International Journal for numerical methods in engineering* 42.3 (1998), pp. 473–498.

[109] Marvin Zhang, Xinyang Geng, Jonathan Bruce, Ken Caluwaerts, Massimo Vespignani, Vytas SunSpiral, Pieter Abbeel, and Sergey Levine. “Deep reinforcement learning for tensegrity robot locomotion”. In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on.* IEEE. 2017, pp. 634–641.

[110] G Zhu, Shuzhi Sam Ge, and Tong Heng Lee. “Simulation studies of tip tracking control of a single-link flexible robot based on a lumped model”. In: *Robotica* 17.1 (1999), pp. 71–78.