Tensegrity Full Body Control

An Inequality Constrained Optimization Approach

by

Kjartan Gudmundsson

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday February 28, 2019.

Student number:	4516028	
Thesis committee:	Ir. J. O. van der Weijde,	TU Delft, supervisor
	Prof. dr. ing. H. Vallery,	TU Delft
	Dr. L. Peternel,	TU Delft

Abstract

Tensegrity is a structural form that is defined as a set of rigid elements suspended in a net of continuous tension. This structure shows potential for compliance, impact tolerance and mechanical robustness. However, its non-linear coupled dynamics and often complex geometry require advanced control strategies. An actuator reference planning strategy to bring tensegrity robots closer to controlled full body movements was proposed by Guido Tournois [56] in 2017. This strategy, called the Full Body Reference Planner (FBRP), finds a sequence of equilibrium configurations for a tensegrity structure, predominantly to follow a trajectory in space. However, the method is incapable of incorporating inequality constraints while obtaining said equilibrium configurations. This is a problem when dealing with certain restrictions, e.g., actuator limitations and stability of the structure.

In this thesis we implemented a robust way to account for inequality constraints while utilizing the FBRP. That was done by means of optimization, i.e., an implementation of a Sequential Quadratic Programming method to ensure inequality constraints were respected for each configuration. The approach was validated in scenarios related to practical applications where inequality constraints were enforced. The results showed advancements towards practical feasibility. Furthermore, the robustness, efficiency and accuracy of the method were validated. The extended implementation depicted robustness to parameter variations and good results in terms of accuracy. However, given the iterative nature of the method, it was more computationally expensive than its precursor.

Preface

The end of this thesis project marks the finishing line of my path at the Delft University of Technology. It has been an incredible learning experience, not only academically but also culturally and socially.

The initial focus of my research project was to make a prototype of a tensegrity robot actuated by twisted and coiled polymer muscles (TCPM). The goal was to achieve controlled full body movements of the prototype utilizing prior work on the Full Body Reference Planner (FBRP) by Guido Tournois. I want to thank him for his help and contribution during the initial phase of the project. By learning first hand about the physical restrictions of the TCPM and the limitations of the FBRP, it became clear that the approach needed supplementary work in order to account for certain practical restrictions. This motivated the subject of this thesis project which focuses on enhancing the FBRP to account for practical restrictions.

This project has taken longer than I intended and there have been times where I have been close to giving up. I want to thank my supervisors Joost van der Weijde and Heike Vallery for their help and valuable guidance. Also, I want to thank Joost for not giving up on me during this rather long and cumbersome journey, it is much appreciated. Finally, to my dear partner Kristin, I thank you for always pushing me and supporting me in the hardest of times, I could never have done it without you.

Contents

1 Introduction

In the past decade, a special bioinspired structure has caught the eye of robotics researchers. This structural form is called tensegrity and can be found on many levels in nature and daily life, from molecular biology [24] to man-made architecture such as the Kurilpa Bridge seen in Figure 1.1.

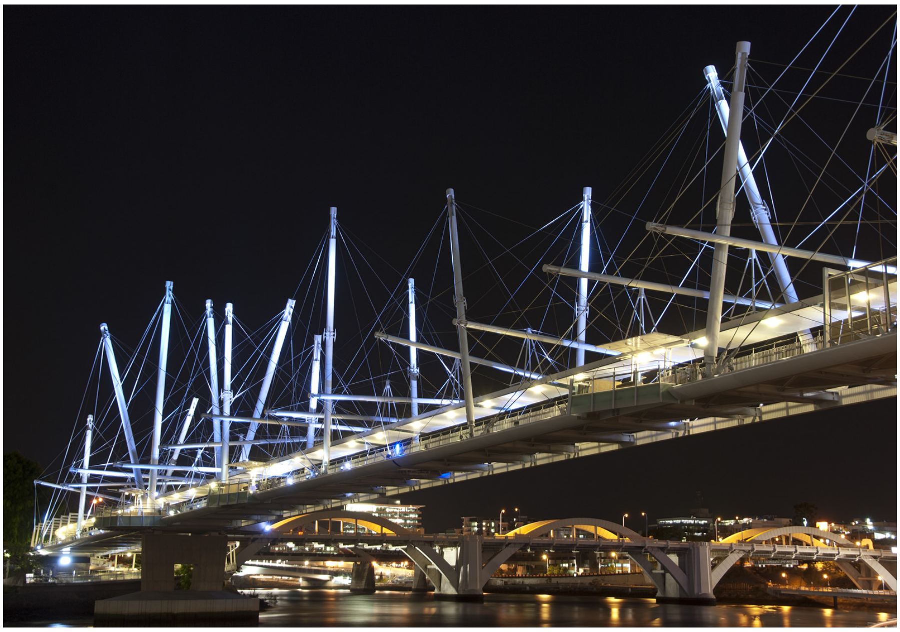
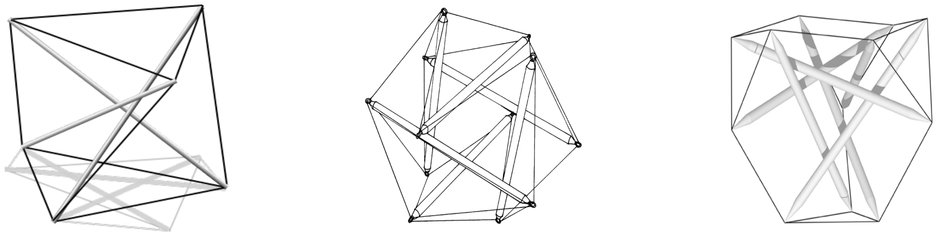


Figure 1.1. Example of tensegrity in Architecture. Source: [10].

The word tensegrity is a portmanteau of the term tensional integrity. In literature, Buckminister Fuller [15] is often considered to be the inventor of the structure as he was the first to coin the word itself. However, Kenneth Snelson published an article in 1996 [51] claiming he was the original inventor of the structure. Tensegrity is defined as a set of rigid elements suspended in a net of continuous tension. The rigid elements are subject to continuous compression which results in no bending moment acting on any structural members of the tensegrity.

Among the reasons why tensegrity structures have caught the eye of researchers is their potential for compliance, impact tolerance and mechanical robustness. However, this bioinspired structure poses a difficult problem when it comes to controls; its nonlinear coupled dynamics and often complex geometry require advanced control strategies. Moreover, tensegrities are often underactuated, i.e., they have more degrees of freedom than actuators which underpins the challenging nature of controlling such a structure.

Tensegrity structures can take various shapes and forms. For example, the simplest form, a 3-prism can be seen in Figure 1.2a. A further complex geometry in the form of the tetrahedron can be seen in Figure 1.2c. As the structures become more intricate, the more compression and tension elements are present which results in more complex dynamics.



a. 3-prism. Source: [2] **b.** Icosahedron. Source: [5] **c.** Tetrahedron. Source: [6]

Figure 1.2. Common tensegrity structures.

1.1. Tensegrity Robots

In the year 2000, Sultan et. al [52] proposed a tensegrity approach for a flight simulator, implicitly linking tensegrity to robotics. Three years later, Aldrich et. al [1] became first to explicitly link the structure to robotics by demonstrating control of a tethered tensegrity robotic arm. Since then many robots have been developed based on the principle of tensegrity. Examples include climbing, crawling and rolling robots [7, 13, 48, 55]. These types of robots could be used in near future space exploration where its feasibility has been addressed in a thesis by Hong in 2014 [23]. Space exploration by means of tensegrity robots has also motivated research by the NASA Ames Research Center's Intelligent Robotics Group [7, 47, 48] where Figure 1.3 shows one of their prototypes.

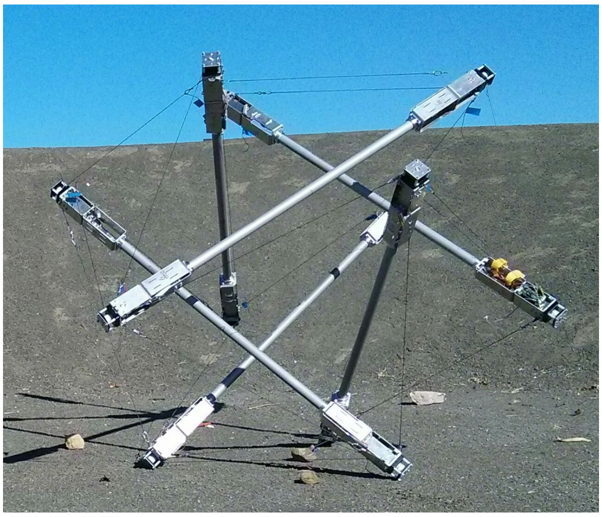


Figure 1.3. Fully assembled prototype of the robot SUPERBall, developed at DTRL, NASA. Source: [48], with author's consent.

Furthermore, with their compliant body properties, tensegrities open up opportunities for application in human-robot interaction [31]. Current robots that interact with humans are often heavy, rigid and powerful, which may result in hazardous interaction. Inspection of areas unreachable to humans is also a field of application that has been researched, e.g., climbing in tight areas for duct inspection [13, 14].

Several factors need to be taken into consideration when designing a robot, from mechanical design to control strategies. For example, a key component in any robot build are actuators and tensegrity robots are no exception there. Many different actuators have been utilized in tensegrity builds found in literature. For example, shape memory alloys [41, 50], pneumatic artifical muscles [9, 32], twisted and coiled polymer muscles (TCPM) [21, 59] and linear DC motors [30]. These actuators all have limitations which need to be taken into account when building and controlling the robot (e.g., stroke, torque and speed constraints). A specified practical example of the limitations of TCPMs can be found in Appendix B.

1.2. Tensegrity Control

Initially, control strategies for tensegrity structures were classical open loop control schemes [1, 37] but soon the field started leaning towards artificial intelligence as an answer to the structures' complexity. In 2013 Iscen et. al [26] compared two different machine learning control approaches to a hand coded solution, single-agent learning and multi-agent learning. Control of tensegrity via Central Pattern Generators (CPGs) has been explored in a tensegrity swimmer [3] and on the robots Tetraspine & ReCTeR [7, 55] which showed promising results in simulation.

Another area of control for tensegrity robots are vibration-driven tensegrity robots. Khazanov et. al [28, 29] combined morphological computation with vibration actuation to exploit the tensegrity's complexity to achieve robot locomotion. In contrast, Böhm et. al [4] explored locomotion by a single vibration driven actuator to simplify all controls.

The robot ReCTeR was tested with various control strategies such as coevolutionary control, bioinspired control and physical reservoir computing [7]. The open loop coevolutionary control was adapted from prior work by Panait [42] to icosahedral tensegrity robots by Iscen et. al [25]. This demonstrated that machine learning controls can elicit robust rolling locomotion of icosahedral robots without actual understanding of the robot's analytical dynamics. The bioinspired control approach was divided in three different methods: reactive controls, inverse kinematics controls and CPG-based controls, all of which being closed loop approaches. All these strategies had the same underlying principle where the robot's speed and heading were controlled by shifting its center of mass.

The spherical tensegrity robot developed by Böhm et. al was also tested with a closed loop control approach. In principle, the method shifts the robot's centre of mass (CoM) outside its base of support via internal mass movement. That was done to achieve tip over motion and if applied in sequence, rolling locomotion was realized.

By definition, the base of support (BoS) is the area encompassed by every point of contact an object makes with its supporting surface. Furthermore, a common way of defining stability in locomotion is: an object has to confine its CoM within the BoS so that while standing the body remains balanced [11, 34, 43]. This can be important in tensegrity locomotion as for example a traversing tensegrity icosahedron robot rolls by shifting its centre of mass outside its base of support. That results in a new supporting triangle i.e. three contact nodes defining a new base of support. By sequencing this operation in a controlled manner robot rolling can be realized.

Koizumi, Hirai & Shibata (2012-13) [22, 32] used their proposed principle from 2009 [50] to demonstrate open loop control of their icosahedron robot. The principle states that the gravitational potential energy of the robot system is at a minimum during a naturally stable state. By actuating the muscles of the structure a nonlinear force distribution occurs, resulting in a bending moment around the robot's contact base with the ground. This bending moment induces a tip over step from one contact triangle to another which is again the same principle mentioned above. Due to the complexity of the robot dynamics the authors did not attempt an analytical prediction of which muscles needed activation at each time to make a goal-directed step. Kim et. al (2014) [30] used a similar principle of open loop control on their rapid prototyping kit for icosahedron tensegrity robots. Wu et. al (2016) [59] also exhibited open loop control of their icosahedron robot in a similar fashion to Koizumi et. al, but directed focus more towards twisted and coiled polymer muscle actuation than contact modelling.

The robot DuCTT (v1-2) achieved climbing locomotion via a force density method for inverse kinematic control policy [13, 14]. This open loop control strategy has showed promising results and is built on the same principles as the Full Body Reference Planner described in the following section.

1.3. Full Body Reference Planner

In 2017, Guido Tournois proposed an actuator reference planning strategy to bring tensegrity robots closer to controlled full body movements [56]. He called his method the Full Body Reference Planner (FBRP). This strategy is a numerical reference planner that generates a sequence of equilibrium configurations for a given task. Firstly, static equilibrium equations are obtained via force density theory. Combined with kinematic equality constraints the method defines a system of equations which establishes an equilibrium manifold [44].

The FBRP is an efficient method of finding a sequence of equilibrium configurations for a tensegrity structure, namely to follow a trajectory in space. However, it can be augmented when it comes to respecting inequality constraints. As has been mentioned it can incorporate equality constraints in an efficient manner but fails with inequalities as they are not differentiable. This is a problem when dealing with certain restrictions, e.g., actuator limitations and stability concerns.

1.4. Contribution

The contribution of this thesis project is to implement a robust way to account for inequality constraints while utilizing the Full Body Reference Planner. That is done by means of implementing a Sequential Quadratic Programming method to ensure inequality constraints are respected. The extended implementation is meant to make the FBRP more feasible in practice. This approach is finally validated with various practical applications such as constraining a structure's cable force changes and centre of mass position in space. That can, for instance, ensure that physical limitations of actuators are respected. Furthermore, constraining the structure's CoM can aid with stability. That is done by confining the CoM within its base of support while tracking a trajectory.

2

Background - Full Body Reference Planner

In this chapter the FBRP for tensegrity trajectory tracking is derived as proposed by Guido Tournois in 2017 [56]. The input of the FBRP is a desired set of tensegrity states, namely a trajectory formulated in nodal coordinates. The planner then renders a change in state variables to achieve a set of equilibrium configurations while tracking this desired trajectory. The remainder of this thesis uses the following assumptions:

- Cables are massless
- Rods are assumed infinitely stiff relative to cables
- Influence of rod thickness ignored
- Quasi-static behaviour, i.e., velocities and accelerations are zero
- Member forces act only on nodes
- External forces other than gravitational force and their reactions are neglected
- Tension forces are regarded as positive and compression forces as negative

2.1. Equilibrium Force Equations

A tensegrity structure is modelled with points in space, denoted as nodes. They are connected by the rigid members under compression, denoted as rods, and elastic members under tension, denoted as cables. Let us define a Cartesian coordinate system with x, y, z as dimensions. In this system we model a tensegrity structure with N nodes and M members (rods and cables). Coordinates of i^{th} node are defined as (x_i, y_i, z_i). External forces acting on node i are defined as $F^{\text{ext}}_{i,x}$, $F^{\text{ext}}_{i,y}$ and $F^{\text{ext}}_{i,z}$ acting in their respective dimensions. As stated in force-density theory [12], the force equilibrium equations for node i can be written as

$$\sum_{j=1}^{m}\left(\left(x_j - x_i\right) q_{i,j}\right) + F_{i,x}^{\text{ext}} = 0 \tag{2.1a}$$

$$\sum_{j=1}^{m}\left(\left(y_j - y_i\right) q_{i,j}\right) + F_{i,y}^{\text{ext}} = 0 \tag{2.1b}$$

$$\sum_{j=1}^{m}\left(\left(z_j - z_i\right) q_{i,j}\right) + F_{i,z}^{\text{ext}} = 0\,, \tag{2.1c}$$

where according to tensegrity form-finding [12, 57, 61, 62] we define $q_{i,j} := f_{i,j}/l_{i,j}$ as the force-density of a member connecting node i to node j. Note that $f_{i,j}$ and $l_{i,j}$ are the force and length of the member between node i and a node j.

Next we define an incidence matrix $\mathbf{C} \in 2\mathbb{R}^{(M\times N)}$ [62] to compress the $3N$ force equilibrium equations. This matrix contains information about member connections to each node of the tensegrity and is constructed as

$$\mathbf{C}_{k,p} = \begin{cases} 1 & \text{for } p = i \\ -1 & \text{for } p = j \\ 0 & \text{for other cases}\,, \end{cases} \tag{2.2}$$

where k indexes a matrix row that represents connection information of k^{th} member of the tensegrity. Next, we define a set of nodal coordinate vectors in our Cartesian coordinate system. For N nodes the vectors are denoted

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T \in \mathbb{R}^N \tag{2.3a}$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_N)^T \in \mathbb{R}^N \tag{2.3b}$$

$$\boldsymbol{z} = (z_1, z_2, \ldots, z_N)^T \in \mathbb{R}^N\,. \tag{2.3c}$$

Furthermore, we define a force-density vector for M members as $\boldsymbol{q} = (q_1, q_2, ..., q_M)^T \in \mathbb{R}^M$ [12, 57, 61, 62]. Then equation (2.1) can be rewritten for N nodes as [54, 62]

$$\mathbf{C}^T\mathbf{Q}\mathbf{C}\boldsymbol{x} + \boldsymbol{F}_x^{\text{ext}} = 0 \tag{2.4a}$$

$$\mathbf{C}^T\mathbf{Q}\mathbf{C}\boldsymbol{y} + \boldsymbol{F}_y^{\text{ext}} = 0 \tag{2.4b}$$

$$\mathbf{C}^T\mathbf{Q}\mathbf{C}\boldsymbol{z} + \boldsymbol{F}_z^{\text{ext}} = 0\,, \tag{2.4c}$$

where $\mathbf{Q}$ is a diagonal force density matrix with $\boldsymbol{q}$ on the diagonal and $\boldsymbol{F}_x^{\text{ext}}$, $\boldsymbol{F}_y^{\text{ext}}$ and $\boldsymbol{F}_z^{\text{ext}}$ are external forces acting on each node in their respective dimensions.

2.2. Reference Planner

The configuration of a tensegrity structure is described by a set of internal states. These states are represented by the nodal coordinate and force-density vectors. Let us define vector $\boldsymbol{\xi}$ to represent the configuration and degrees of freedom of a tensegrity structure

$$\boldsymbol{\xi} = \left(\boldsymbol{x}^T, \boldsymbol{y}^T, \boldsymbol{z}^T, \boldsymbol{q}^T\right)^T .$$

Now we can rewrite 2.4 as a nonlinear function $f : \mathbb{R}^{3N+M} \to \mathbb{R}^{3N}$ of $\boldsymbol{\xi}$

$$f(\boldsymbol{\xi}) = 0 . \tag{2.5}$$

As stated in the previously listed assumptions, we set $F_x^{ext} = F_y^{ext} = 0$ and assume cables are massless. In the z-direction gravitational forces and ground reaction forces (GRF) are taken into account. We assume even mass distribution of rods such that gravity is equally distributed between two nodes on each rod

$$F_{i,z}^{\text{ext}} = F_{j,z}^{\text{ext}} = \frac{m_{\text{r}} g}{2} , \tag{2.6}$$

where m_{r} is the rod mass and g gravity.
The GRFs are modelled such that they support the full weight of the tensegrity structure. The equilibrium equations that correspond to a contact node n_{c} are summed up and equated to the negative gravitational force

$$\sum \left(f_{n_{\text{c}}}\right) = -m_{\text{r}} g . \tag{2.7}$$

Let $\left(g_1(\boldsymbol{\xi}) = 0, \ldots, g_h(\boldsymbol{\xi})\right)$ be h continuous equality constraint equations. Then a system of equations $\mathbf{J} : \mathbb{R}^{3N+M} \to \mathbb{R}^{3N+h}$ can be defined as

$$\mathbf{J}(\boldsymbol{\xi}) = \begin{bmatrix} f(\boldsymbol{\xi}) \\ g_1(\boldsymbol{\xi}) \\ \vdots \\ g_h(\boldsymbol{\xi}) \end{bmatrix} . \tag{2.8}$$

If a configuration satisfies $\mathbf{J} = 0$, the tensegrity is in a state of equilibrium. Therefore, $\mathbf{J}$ transforms the configuration space of the tensegrity into an implicitly defined equilibrium manifold.

To track desired nodal trajectories we need to determine S number of equilibrium configurations along that path. Let $\boldsymbol{\xi}_{\text{c}}$ denote the set of controlled state variables with a subset of desired nodal coordinates $\boldsymbol{\xi}_{\text{c,d}}$. Next the states are split into task variables and free variables $\boldsymbol{\xi} = \left(\boldsymbol{\xi}_{\text{c}}^T, \boldsymbol{\xi}_{\text{f}}^T\right)^T$ where the latter are not restrained to tracking trajectories.

For the assumption of quasi-static behavior to hold $\mathbf{J}$ needs to be zero at all times, including when changing between equilibrium configurations. The states of the tensegrity have a linear relation in their velocity as $\mathbf{J}$ is a second order system [45]. The change in $\mathbf{J}$ is then approximated by the first order derivative with respect to $\boldsymbol{\xi}$ times the discrete change in configuration states

$$\Delta \mathbf{J} \approx \frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}} \Delta \boldsymbol{\xi} = 0 \,, \tag{2.9}$$

where $\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}}$ is a Jacobian matrix. Given that $\Delta\boldsymbol{\xi}$ is small enough, this relationship can be exploited to determine the state changes necessary to follow a nodal trajectory while staying bounded on the equilibrium manifold. States $\boldsymbol{\xi}_{\mathrm{c,d}}$ for a desired trajectory are discretized in $S+1$ steps in order to achieve S number of desirable state changes. Note that to exploit this relation, inequality constraints can not be included in $\mathbf{J}$ as they are not differentiable. Next we refactor equation 2.9 with respect to task and free state variables

$$\Delta \mathbf{J} = \frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{c}}} \Delta \boldsymbol{\xi}_{\mathrm{c,d}} + \frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{f}}} \Delta \boldsymbol{\xi}_{\mathrm{f}} = 0 \,, \tag{2.10}$$

that allows us to connect the change in tracking variables to the change in free variables

$$\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{f}}} \Delta \boldsymbol{\xi}_{\mathrm{f}} = -\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{c}}} \Delta \boldsymbol{\xi}_{\mathrm{c,d}} \,. \tag{2.11}$$

The Jacobian matrices $\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{f}}}$ and $\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{c}}}$ have a column rank deficiency and are therefore singular. This also means that these equations are underdetermined and have an infinitude of solutions. The Moore-Penrose pseudoinverse gives a solution

$$\Delta \boldsymbol{\xi}_{\mathrm{f}} = \left(\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{f}}} \right)^{\dagger} \left(-\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{c}}} \Delta \boldsymbol{\xi}_{\mathrm{c,d}} \right) \,. \tag{2.12}$$

Now the change in all state variables from configuration ξ_s can be determined to ensure a consecutive equilibrium configuration

$$\boldsymbol{\xi}_{s+1} = \boldsymbol{\xi}_s + \begin{bmatrix} \Delta \boldsymbol{\xi}_{\mathrm{c,d}} \\ \Delta \boldsymbol{\xi}_{\mathrm{f}} \end{bmatrix}_{s \to s+1} \,. \tag{2.13}$$

Since the change in state is determined by a first order approximation, numerical errors are inevitable. These errors might lead to ill-conditioning of the Jacobian $\frac{\delta \mathbf{J}}{\delta \xi_{\mathrm{f}}}$. That can cause a very inaccurate approximation of $\boldsymbol{\xi}_{s+1}$ to be obtained. The pseudoinverse can be composed by use of singular value decomposition to mitigate these numerical errors where singular values below a set tolerance are set to zero [19].

2.3. Initial Conditions

To make use of the Reference Planner an initial equilibrium configuration $\boldsymbol{\xi}_0$ needs to be determined. This configuration can be determined by the following optimization problem

$$\min_{\boldsymbol{\xi}} \quad \left\{ f(\boldsymbol{\xi}), \|\boldsymbol{\xi}_{\mathrm{d}} - \boldsymbol{\xi}\| \right\} \tag{2.14a}$$

$$\text{s.t.} \quad g(\boldsymbol{\xi}) = 0, \tag{2.14b}$$

$$\boldsymbol{\xi}_{\mathrm{lb}} \leq \boldsymbol{\xi} \leq \boldsymbol{\xi}_{\mathrm{ub}}, \tag{2.14c}$$

where $\boldsymbol{\xi}_\mathrm{d}$ is the set of desired states, $g\left(\boldsymbol{\xi}\right)$ are constraints and ξ_lb and ξ_ub are the bounds of the nodal coordinates and member force densities.

3

Method

This chapter outlines the method for full body control of a tensegrity structure while respecting a set of inequality constraints. The structure control is realized by a full body reference planner which calculates a sequence of equilibrium configurations for a desired trajectory. This method is based on force density theory [12] and tensegrity form-finding [12, 57, 61, 62]. Due to the nature of this method it can not incorporate inequality constraints directly. The planner is extended to incorporate inequality constraints to deal with limitations, as described previously, in practical applications. That is accomplished by means of a Sequential Quadratic Programming method based on theory by Gill and Murray (1978) [16], Schmid and Biegler (1994) [49] and Nocedal and Wright (2006) [39]. It iteratively solves Karush-Kuhn-Tucker (KKT) conditions [33] of a Quadratic Program and then uses a line search method for a second order correction. Each step is solved by means of an active set method [16, 18, 49]. Note that the remainder of this thesis builds on the same assumptions as listed in Chapter 2.

3.1. Adding Inequality Constraints

In this section the method of incorporating inequality constraints into the reference planner is depicted. As stated before, inequality constraints are not differentiable and can therefore not be added in equation (2.8). Also, the set of equations in (2.12) is an underdetermined system with an infinite number of solutions. Let us rewrite equation (2.12) for shorthand convenience

$$\boldsymbol{d} = \mathbf{A}^{\dagger}\boldsymbol{r}\ , \tag{3.1}$$

where $\Delta\boldsymbol{\xi}_{\mathrm{f}} = \boldsymbol{d}$, $\left(\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{f}}}\right) = \mathbf{A}$ and $\left(-\frac{\delta \mathbf{J}}{\delta \boldsymbol{\xi}_{\mathrm{c}}}\Delta\boldsymbol{\xi}_{\mathrm{c,d}}\right) = \boldsymbol{r}$. In accordance to generalized inverse theory [27], if a linear system of equations has any solutions they are all given by

$$\boldsymbol{d} = \mathbf{A}^{\dagger}\boldsymbol{r} + \left(\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A}\right)\boldsymbol{w}\ , \tag{3.2}$$

where $\mathbf{I}$ is the identity matrix and $\boldsymbol{w}$ an arbitrary vector. Solution(s) exist if and only if $\mathbf{A}\mathbf{A}^{\dagger}\boldsymbol{r} = \boldsymbol{r}$. If a solution exists, it is unique if and only if $\mathbf{A}$ has full column rank. In that case

$\left(\mathbf{I}-\mathbf{A}^{\dagger}\mathbf{A}\right)$ is a null matrix. Here $\mathbf{A}$ has column rank deficiency and the system in (3.1) is, as mentioned before, underdetermined. Hence, all of the infinite number of solutions are given by equation (3.2). Next let us rewrite equation (3.2) with $\mathbf{M} = \left(\mathbf{I}-\mathbf{A}^{\dagger}\mathbf{A}\right)$ for shorthand convenience

$$\boldsymbol{d} = \mathbf{A}^{\dagger}\boldsymbol{r} + \mathbf{M}\boldsymbol{w} \ . \tag{3.3}$$

Now we can tailor arbitrary vector $\boldsymbol{w}$ to obtain a solution that meets certain criteria. We can determine $\boldsymbol{w}$ in an iterative manner by means of minimizing $\boldsymbol{d}^T\boldsymbol{d}$. This gives freedom to implement not differentiable inequality constraints. The optimization problem is defined

$$\begin{aligned} \min_{w} \quad & \left(\mathbf{A}^{\dagger}\boldsymbol{r} + \mathbf{M}\boldsymbol{w}\right)^T \left(\mathbf{A}^{\dagger}\boldsymbol{r} + \mathbf{M}\boldsymbol{w}\right) && \text{(3.4a)} \\ \text{s.t.} \quad & c_i \geq 0 \ , && \text{(3.4b)} \end{aligned}$$

where c_i can be n number of inequality constraints where $\{n \in \mathbb{Z} : n \geq 0\}$. Extending the objective function and dropping expressions without $\boldsymbol{w}$ the following is obtained

$$\begin{aligned} \min_{w} \quad & f(\boldsymbol{w}) = \boldsymbol{w}^T\mathbf{M}^T\mathbf{M}\boldsymbol{w} + 2\boldsymbol{w}^T\mathbf{M}^T\mathbf{A}^{\dagger}\boldsymbol{r} && \text{(3.5a)} \\ \text{s.t.} \quad & c_i(\boldsymbol{w}) \geq 0 \ . && \text{(3.5b)} \end{aligned}$$

This is now a Quadratic Programming (QP) problem and can be solved via optimization. With this method we can now obtain a set of equilibrium configurations for a desired trajectory whilst respecting inequality constraints. This also allows for the possibility to enforce various constraints, e.g., to respect actuator limitations and aid with stability. To solve this problem we implement a robust Sequential Quadratic Programming (SQP) method.

3.2. Sequential Quadratic Programming

Sequential Quadratic Programming is an effective method for optimization problems subject to nonlinear constraints. It is based on the principle of iteratively solving a Quadratic Programming subproblem until optimality is realized. In this implementation the QP subproblem is solved at each iteration by means of an active set method defined in Section 3.3. That solution is then corrected using a line search method to increase robustness and efficiency [58]. The approach can be viewed as a generalized version of Newton's method. A Newton step for a function g at step k is defined as

$$g(x_k) + \nabla g(x_k)^T \Delta x = 0 \ . \tag{3.6}$$

For this implementation, the function g would correspond to KKT conditions for a given Lagrangian function. Let us firstly define the Lagrangian for our problem in (3.5)

$$L(\boldsymbol{w}, \boldsymbol{\lambda}) = f(\boldsymbol{w}) - \boldsymbol{\lambda}^T c(\boldsymbol{w}) \ , \tag{3.7}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers. Optimality is obtained by solving its corresponding KKT conditions. They are defined in matrix form as

$$\begin{aligned}\mathbf{K}(\boldsymbol{w},\boldsymbol{\lambda}) &= \begin{bmatrix}\nabla_w L(\boldsymbol{w},\boldsymbol{\lambda})\\ \nabla_\lambda L(\boldsymbol{w},\boldsymbol{\lambda})\end{bmatrix}\\ &= \begin{bmatrix}\nabla f(\boldsymbol{w}) - \nabla c(\boldsymbol{w})\,\boldsymbol{\lambda}\\ -c(\boldsymbol{w})\end{bmatrix} .\end{aligned} \tag{3.8}$$

The corresponding gradient is

$$\nabla\mathbf{K}(\boldsymbol{w},\boldsymbol{\lambda}) = \begin{bmatrix}\nabla^2_{ww} L(\boldsymbol{w},\boldsymbol{\lambda}) & -\nabla c(\boldsymbol{w})\\ -\nabla c(\boldsymbol{w})^T & 0\end{bmatrix} , \tag{3.9}$$

where

$$\nabla^2_{ww} L(\boldsymbol{w},\boldsymbol{\lambda}) = \nabla^2 f(\boldsymbol{w}) - \sum_{i=1}^{n} \lambda_i \nabla^2 c_i(\boldsymbol{w}) \ . \tag{3.10}$$

Now we can write the Newton step as

$$\begin{bmatrix}\nabla^2_{ww} L(\boldsymbol{w},\boldsymbol{\lambda}) & -\nabla c(\boldsymbol{w})\\ -\nabla c(\boldsymbol{w})^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{w}\\ \Delta\boldsymbol{\lambda}\end{bmatrix} = -\begin{bmatrix}\nabla f(\boldsymbol{w}) - \nabla c(\boldsymbol{w})\,\boldsymbol{\lambda}\\ -c(\boldsymbol{w})\end{bmatrix} , \tag{3.11}$$

with a corresponding Quadratic Program

$$\min_{w} \quad \frac{1}{2}\boldsymbol{w}^T \nabla^2_{ww} L(\boldsymbol{w},\boldsymbol{\lambda})\,\boldsymbol{w} + \nabla_w L(\boldsymbol{w},\boldsymbol{\lambda})^T\,\boldsymbol{w} \tag{3.12a}$$

$$\text{s.t.} \quad \nabla c(\boldsymbol{w})^T\,\boldsymbol{w} \geq -c(\boldsymbol{w}) \ . \tag{3.12b}$$

According to Gill and Wong (2012) [17], we can now simplify the KKT system by inserting $\Delta\boldsymbol{\lambda} = \boldsymbol{\mu} - \boldsymbol{\lambda}$

$$\begin{bmatrix}\nabla^2_{ww} L(\boldsymbol{w},\boldsymbol{\lambda}) & -\nabla c(\boldsymbol{w})\\ -\nabla c(\boldsymbol{w})^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{\mu}\end{bmatrix} = -\begin{bmatrix}\nabla f(\boldsymbol{w})\\ -c(\boldsymbol{w})\end{bmatrix} . \tag{3.13}$$

This simplified system now corresponds to a Quadratic Program which is then solved at each iteration using an active set method

$$\min_{w} \quad \frac{1}{2}\boldsymbol{w}^T \nabla^2_{ww} L(\boldsymbol{w},\boldsymbol{\lambda})\,\boldsymbol{w} + \nabla f(\boldsymbol{w})^T\,\boldsymbol{w} \tag{3.14a}$$

$$\text{s.t.} \quad \nabla c(\boldsymbol{w})^T\,\boldsymbol{w} \geq -c(\boldsymbol{w}) \ . \tag{3.14b}$$

3.2.1. Line Search

Let us define a merit function Ψ and consider a step $\boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k$ using a correction α_k and step direction $\boldsymbol{p}_k$

$$\Psi\left(\boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k,\ \boldsymbol{\mu}\right) = f\left(\boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k\right) + \sum_{i=1}^{n} \mu_i \left| c_i\left(\boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k\right)\right| . \tag{3.15}$$

Next we calculate a directional derivative $\mathrm{D}_p\Psi$ in the direction p_k

$$\mathrm{D}_p\Psi\left(\boldsymbol{w}_k, \boldsymbol{\mu}\right) = \nabla f^T \boldsymbol{p_k} - \sum_{i=1}^{n} \mu_i \left| c_i\left(\boldsymbol{w_k} + \alpha_k \boldsymbol{p_k}\right)\right| . \tag{3.16}$$

Now we can define a second order correction step

$$\boldsymbol{h}_k = -\mathbf{C}_k^T \left(\mathbf{C}_k \mathbf{C}_k^T\right)^{-1} c\left(\boldsymbol{w}_k + \boldsymbol{p}_k\right) , \tag{3.17}$$

where $\mathbf{C} = \nabla c\left(\boldsymbol{w}\right)$ and follow the iterative procedure defined in Algorithm 1.

Algorithm 1: Line Search

```
// Choose parameters $\beta \in (0, 0.5)$ and $\tau_1$, $\tau_2$ with $0 < \tau_1 < \tau_2 < 1$;
// Evaluate $f_k, \nabla f_k, c_k, \mathbf{C}_k$;
// Obtain step direction $\boldsymbol{p}_k$ by means of active set method (3.3);
// Set $\alpha_k \leftarrow 1$
while NOT STOP do
    if $\Psi\left(\boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k\right) \leq \Psi\left(\boldsymbol{w}_k\right) + \beta \alpha_k \mathrm{D}_p\Psi\left(\boldsymbol{w}_k\right)$ then
        // Set $\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k$
        // STOP;
    else if $\alpha_k = 1$ then
        // Compute $\boldsymbol{h}_k$ from (3.17);
        if $\Psi\left(\boldsymbol{w}_k + \boldsymbol{p}_k + \boldsymbol{h}_k\right) \leq \Psi\left(\boldsymbol{w}_k\right) + \beta \mathrm{D}_p\Psi\left(\boldsymbol{w}_k\right)$ then
            // Set $\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k + \boldsymbol{p}_k + \boldsymbol{h}_k$;
            // STOP;
        end
        // Choose new $\alpha_k \in [\tau_1 \alpha_k, \tau_2 \alpha_k]$;
    else
        // Choose new $\alpha_k \in [\tau_1 \alpha_k, \tau_2 \alpha_k]$;
    end
end
```

3.2.2. Hessian Approximation

The Hessian of the Lagrangian $\nabla^2_{ww} L\left(\boldsymbol{w}, \boldsymbol{\lambda}\right)$ is computationally expensive to obtain and therefore an approximation is used. Estimation $\mathbf{G}_k$ of the Hessian of the Lagrangian can be obtained based on the secant condition where

$$\boldsymbol{s} = \boldsymbol{w}^{k+1} - \boldsymbol{w}^{k} \tag{3.18}$$

$$\mathbf{y} = \nabla L\left(\boldsymbol{w}^{k+1}, \boldsymbol{\lambda}^{k+1}\right) - \nabla L\left(\boldsymbol{w}^{k}, \boldsymbol{\lambda}^{k+1}\right) \tag{3.19}$$

$$\mathbf{G}_{k+1} = \mathbf{G}_k + \frac{\mathbf{y}\mathbf{y}^T}{\mathbf{y}^T\boldsymbol{s}} - \frac{\mathbf{G}_k\boldsymbol{s}\boldsymbol{s}^T\mathbf{G}_k}{\boldsymbol{s}^T\mathbf{G}_k\boldsymbol{s}} \ . \tag{3.20}$$

If $\mathbf{G}_k$ is positive definite and $\mathbf{y}^T\boldsymbol{s} > 0$ then $\mathbf{G}_{k+1}$ will be positive definite. By replacing $\mathbf{y}$ in the BFGS-update formula we employ the Powell-SQP update (damped BFGS-update) [46] with

$$\hat{\mathbf{y}} = \theta\mathbf{y} + (1-\theta)\,\mathbf{G}_k\boldsymbol{s} \ , \ \theta \in [0,1] \ , \tag{3.21}$$

which ensures $\mathbf{y}^T\boldsymbol{s} > 0$ and therefore positive definiteness of the approximated Hessian.

3.2.3. Algorithm - Sequential Quadratic Programming

Algorithm 2 defines the Sequential Quadratic Programming method in terms of an iterative approach.

Algorithm 2: SQP Method

```
// Evaluate $f_0, \nabla f, c_0, \mathbf{C}_0, \boldsymbol{\lambda}_0$;
while NOT STOP do
    // Obtain solution $[\boldsymbol{w}, \boldsymbol{\mu}]^T$ by means of active set method (3.3)
    if $(|\nabla f^T \boldsymbol{w}| + |\boldsymbol{\mu}^T c_0| = 0)$ then
        // STOP - Optimal solution found
    else
        // Obtain step size $\alpha$ by utilizing line search method (3.2.1)
        // Make step $\boldsymbol{w} \leftarrow \boldsymbol{w}(\alpha)$
        // Evaluate $f_k, \nabla f_k, c_k, \mathbf{C}_k$;
        // Compute
        $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \alpha(\boldsymbol{\mu} - \boldsymbol{\lambda}_0)$
        // Approximate Hessian $\mathbf{G}_{k+1}$ using Damped BFGS update (3.2.2)
    end
end
```

3.3. Active Set Method

This section details the active set method used to solve the QP at each iteration of the SQP method. This active set method is based on theory by [16, 18, 39, 49].

3.3.1. Predefinition

The corresponding QP is defined as

$$\begin{aligned} \min_{w} \quad & f(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\mathbf{G}\boldsymbol{w} + \boldsymbol{a}^T\boldsymbol{w} && && (3.22a)\\ \text{s.t.} \quad & s_i(\boldsymbol{w}) = c_i^T\boldsymbol{w} - b_i \geq 0 && i \in \mathcal{N}\ , && (3.22b) \end{aligned}$$

where $\mathbf{G}$ is our Hessian approximation and $\boldsymbol{a} = \nabla f(\boldsymbol{w})$. We define a Lagrangian function

$$L(\boldsymbol{w},\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{w}^T\mathbf{G}\boldsymbol{w} + \boldsymbol{a}^T\boldsymbol{w} - \sum_{i\in\mathcal{N}} \mu_i\left(c_i^T\boldsymbol{w} - b_i\right) \qquad (3.23)$$

and a coinciding maximization problem or dual program as

$$\begin{aligned} \max_{w,\mu} \quad & L(\boldsymbol{w},\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{w}^T\mathbf{G}\boldsymbol{w} + \boldsymbol{a}^T\boldsymbol{w} - \sum_{i\in\mathcal{N}} \mu_i\left(c_i^T\boldsymbol{w} - b_i\right) && && (3.24a)\\ \text{s.t.} \quad & \mathbf{G}\boldsymbol{w} + \boldsymbol{a} - \sum_{i\in\mathcal{N}} \mu_i c_i = 0 && && (3.24b)\\ & \mu_i \geq 0 && i \in \mathcal{N}\ . && (3.24c) \end{aligned}$$

The Karush-Kuhn-Tucker conditions of the dual program (3.24) can be defined

$$\begin{aligned} & \mathbf{G}\boldsymbol{w}^* + \boldsymbol{a} - \sum_{i\in\mathcal{N}} \mu_i^* c_i = 0 && && (3.25a)\\ & s_i\left(\boldsymbol{w}^*\right) = c_i^T\boldsymbol{w}^* - b_i = 0 && i \in \mathcal{A}\left(\boldsymbol{w}^*\right) && (3.25b)\\ & s_i\left(\boldsymbol{w}^*\right) = c_i^T\boldsymbol{w}^* - b_i \geq 0 && i \in \mathcal{N}\setminus\mathcal{A}\left(\boldsymbol{w}^*\right) && (3.25c)\\ & \mu_i \geq 0 && i \in \mathcal{N}\cap\mathcal{A}\left(\boldsymbol{w}^*\right)\ . && (3.25d) \end{aligned}$$

A key element of the active set method is to keep track of a working set $\mathcal{A}$ where constraints in this set satisfy the following conditions

$$\begin{aligned} & \mathbf{G}\boldsymbol{w} + \boldsymbol{a} - \sum_{i\in\mathcal{A}} \mu_i c_i = 0 && && (3.26a)\\ & s_i(\boldsymbol{w}) = c_i^T\boldsymbol{w} - b_i = 0 && i \in \mathcal{A}(\boldsymbol{w}) && (3.26b)\\ & \mu_i \geq 0 && i \in \mathcal{A}(\boldsymbol{w})\ . && (3.26c) \end{aligned}$$

In addition, Lagrange multipliers that correspond to constraints that are not in the current working set are defined as zero. The basic approach of the method can be briefly summarized in three steps. In Phase 0 we determine an initial guess $\boldsymbol{w}_0$ as defined in Section 3.3.2. In Phase 1 we check whether there are any violated constraints. If there are none violated, our solution is feasible and optimal. If there are any violated constraints we choose one and continue. Phase 2 is where we calculate a new solution by means of determining step direction (Section 3.3.3) and step length (Sec. 3.3.4). Finally, in this phase we determine which step length to use and whether to drop or add constraints to our working set (Section 3.3.5), all while maintaining feasibility. Phases 1 and 2 are repeated until an optimal solution is found.

3.3.2. Initial Guess

In order to obtain an initial guess for the active set method, we start with an empty working set and all Lagrange multipliers as zero

$$\mathscr{A} = \emptyset \tag{3.27}$$

$$\mu_i = 0\,. \tag{3.28}$$

With this we can easily determine a reasonable starting guess by solving

$$\boldsymbol{w}_0 = -\mathbf{G}^{-1}\boldsymbol{a}\,, \tag{3.29}$$

which is an unconstrained KKT system (3.25a).

3.3.3. Step Direction

If we have not obtained an optimal solution at iteration n a constraint s_r is violated. In order to satisfy the violated constraint we need to make a step while maintaining feasibility. The aim is to obtain a step direction $[\boldsymbol{p}, \boldsymbol{v}]^T$ by solving a system of equations on matrix form. That is done by utilizing a null space method which is described in Appendix A. Furthermore, for this section we presume the following:

- $\mathbf{C} = [c_i]_{i\in\mathscr{A}}$ has full column rank
- $\mathbf{G}$ is symmetric and positive definite
- $\bar{\boldsymbol{\mu}} = [\bar{\mu}_i]_{i\in\mathscr{A}}$
- c_r is the violated constraint and μ_r is the corresponding Lagrangian multiplier

Let us define a new step

$$\bar{\boldsymbol{w}} = \boldsymbol{w} + \boldsymbol{z} = \boldsymbol{w} + t\boldsymbol{p} \tag{3.30a}$$

$$\bar{\mu}_i = \mu_i + u_i = \mu_i + t v_i \qquad i \in \mathscr{A} \tag{3.30b}$$

$$\bar{\mu}_r = \mu_r + t\,, \tag{3.30c}$$

where t is the step length and

$$\begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix} t \,. \tag{3.31}$$

Let us write (3.24b) and (3.26b) in matrix form

$$\begin{bmatrix} \mathbf{G} & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{w}} \\ \bar{\boldsymbol{\mu}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} - \begin{bmatrix} c_r \\ 0 \end{bmatrix} \bar{\mu}_r = 0 \,, \tag{3.32}$$

and expand:

$$\begin{bmatrix} \mathbf{G} & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} - \begin{bmatrix} c_r \\ 0 \end{bmatrix} \mu_r + \begin{bmatrix} \mathbf{G} & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{u} \end{bmatrix} - \begin{bmatrix} c_r \\ 0 \end{bmatrix} t = 0 \,. \tag{3.33}$$

As per the conditions from (3.24b) and (3.26b), equation (3.33) reduces to

$$\begin{bmatrix} \mathbf{G} & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{u} \end{bmatrix} - \begin{bmatrix} c_r \\ 0 \end{bmatrix} t = 0 \,. \tag{3.34}$$

We then substitute (3.31) and obtain our desired system

$$\begin{bmatrix} \mathbf{G} & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} c_r \\ 0 \end{bmatrix} \,. \tag{3.35}$$

3.3.4. Step Length

Here we describe how to determine the step length t. First let us define what we call partial step length t_1

$$t_1 = \min\left(\infty, \min_{i;v_i<0} \frac{-\mu_i}{v_i}\right) \geq 0 \,, \tag{3.36}$$

which is the maximum step in dual space without violating dual feasibility. Next we need to look at a full step length t_2 which is the minimum step in primal space such that violated constraint s_r becomes feasible. To derive t_2 we need to look at the effect on the dual program. Focusing only our violated constraint, let us observe the Lagrangian at $\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\mu}}\right)$

$$L\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\mu}}\right) - L\left(\boldsymbol{w}, \boldsymbol{\mu}\right) = -\frac{1}{2} t^2 c_r^T \boldsymbol{p} - t s_r\left(\boldsymbol{w}\right) \,. \tag{3.37}$$

Next we determine the largest increment of the dual program with respect to t

$$\frac{\mathrm{d}L}{\mathrm{d}t} = -t c_r^T \boldsymbol{p} - s_r\left(\boldsymbol{w}\right) \,. \tag{3.38}$$

From this we can derive our full step length t_2

$$t_2 = \frac{-s_r\left(\boldsymbol{w}\right)}{c_r^T \boldsymbol{p}} \ . \tag{3.39}$$

Finally, in case of $c_r^T \boldsymbol{p} \neq 0$ then the step length t, chosen as the minimum of the two t_1 and t_2 is

$$t = \min\left(t_1, t_2\right) \ . \tag{3.40}$$

3.3.5. Adding and Dropping Constraints

In this section we examine when to add and drop constraints from the working set $\mathscr{A}$. It has been stated that we can only solve the system in (3.35) if the following holds:

- **G** is positive definite
- **C** has full column rank

Let us look at when a constraint needs to be dropped from working set $\mathscr{A}$ i.e. when the second condition does not hold. That occurs when the constraints in $\mathbf{C} = [c_i]_{i\in\mathscr{A}}$ and c_r are linearly dependent. This poses a problem when keeping track of the working set and constraint c_r needs to be added to **C**. This is dealt with by dropping a constraint j from $\mathscr{A}$ in order to maintain full column rank of **C** after adding c_r. When a constraint needs to be dropped, we identify it and obtain its index j by

$$j = \operatorname*{argmin}_{j;v_j<0} \frac{-\mu_j}{v_j} \ . \tag{3.41}$$

Now let us look at when a constraint is added to the working set $\mathscr{A}$. That occurs when a step $\left(\boldsymbol{w}, \boldsymbol{\mu}\right) \rightarrow \left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\mu}}\right)$ is infeasible i.e. $c_r^T \boldsymbol{p} \neq 0$. In case of this occurring we add index r to $\mathscr{A}$ and make the largest possible step t in dual space from $\left(\boldsymbol{w}, \boldsymbol{\mu}\right)$ until said constraint c_r blocks us.

3.3.6. Algorithm - Active Set Method

Algorithm 3 defines the active set method in terms of an iterative approach.

Algorithm 3: Active Set Method

```
// Obtain initial guess via unconstrained minimum $\boldsymbol{w}_0 = -\mathbf{G}^{-1}\boldsymbol{a}$. Set
   $\mathscr{A} = \emptyset$, $\mu_i = 0$.
while NOT STOP do
    if $s_i(\boldsymbol{w}) \geq 0, \quad i \in \mathscr{N}$ then
        // STOP - Optimal solution found
    else
        // Choose violated constraint $s_r(\boldsymbol{w}) < 0$
    end
    while $s_r(\boldsymbol{w}) < 0$ do
        // Determine step direction $[\boldsymbol{p}, \boldsymbol{v}]^T$
        if $c_r^T \boldsymbol{p} = 0$ then
            if $v_i \geq 0, \quad i \in \mathscr{A}$ then
                // STOP - Problem is infeasible
            else
                // Compute step length $t$ via (3.36) and identify constraint
                   $j$ via (3.41) and set:
                $\mu_i \leftarrow \mu_i + t v_i, \qquad i \in \mathscr{A}$
                $\mu_r \leftarrow \mu_r + t$
                // Remove constraint $j$ from $\mathscr{A}$
                $\mathscr{A} \leftarrow \mathscr{A} \setminus j$
            end
        else
            // Compute step length $t_1$ via (3.36) and $t_2$ (3.39) and identify
               constraint $j$ via (3.41)
            if $t_2 \leq t_1$ then
                // Take step
                $\boldsymbol{w} \leftarrow \boldsymbol{w} + t_2 \boldsymbol{p}$
                $\mu_i \leftarrow \mu_i + t_2 v_i$
                $\mu_r \leftarrow \mu_r + t_2$
                // Append constraint $r$ to $\mathscr{A}$
                $\mathscr{A} \leftarrow \mathscr{A} \cup r$
            else
                // Take step
                $\boldsymbol{w} \leftarrow \boldsymbol{w} + t_1 \boldsymbol{p}$
                $\mu_i \leftarrow \mu_i + t_1 v_i$
                $\mu_r \leftarrow \mu_r + t_1$
                // Remove constraint $j$ from $\mathscr{A}$
                $\mathscr{A} \leftarrow \mathscr{A} \setminus j$
            end
        end
    end
end
```

4
Validation

This chapter describes the validation procedures and results for the method implemented in Chapter 3.

4.1. Setup

In this validation process the extended reference planner was implemented numerically in MATLAB®, version 9.3 (2017b) [38]. MATLAB® was chosen for this implementation due to its availability and competency with algorithms and numerical analysis. It is validated on a 3-prism, which is the simplest form of a tensegrity structure. Figure 1.2a and Figure 4.1 illustrate this structure. Its simplicity allows for a clear presentation of information. As the number of elements in a tensegrity increase, it quickly becomes intricate. The 3-prism consists of a total of twelve elements of which three are rods and nine are massless cables. Furthermore, nodes 1, 2 and 3 in Figure 3.1 were constrained to their initial position. For this validation, nodes 4, 5 and 6 were unconstrained. Table 4.1 lists some general numerical values used for this implementation. These values are based on a desktop prototype made up of polylactic acid rods and thin string cables. Table 4.2 shows parameter settings used for the Full Body Reference Planner and its SQP extension. Note that if the algorithm converges, all prescribed trajectories are accurately tracked by the FBRP.

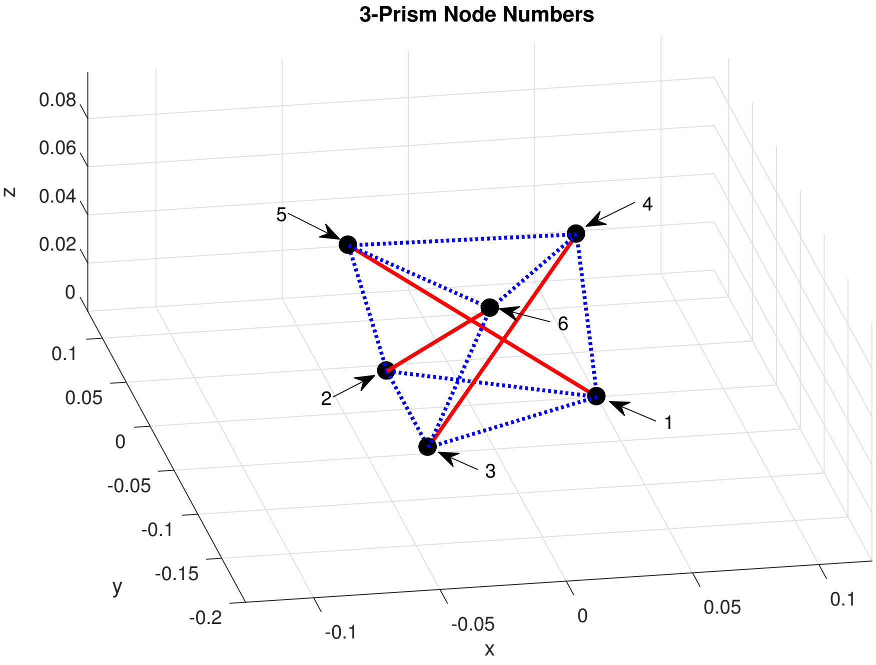


Figure 4.1. Node numbering for numerical implementation. In this graphical representation nodes are black, rods red and cables blue.

Table 4.1. Numerical values for implementation.

Component	***Value***
Gravity	$9.81\,\mathrm{m/s^2}$
Rod radius	1.00 cm
Rod length	15.00 cm
Rod mass	10.00 g
Initial top/bottom cable length	8.66 cm
Initial diagonal cable length	11.76 cm
Cable mass	Massless
Structure mass	30.00 g

Table 4.2. Algorithm input parameters, both for FBRP and SQP.

Method	***Parameter***	***Value***	***Description***
FBRP	n_s	250	No. of discrete steps for trajectory tracking.
	ϵ	$5\cdot10^{-5}$	Tolerance on force equilibrium equations for initial configuration.
SQP	τ	$5\cdot10^{-5}$	Tolerance on termination criteria.
	i_m	1000	Max iterations for SQP and active set methods.

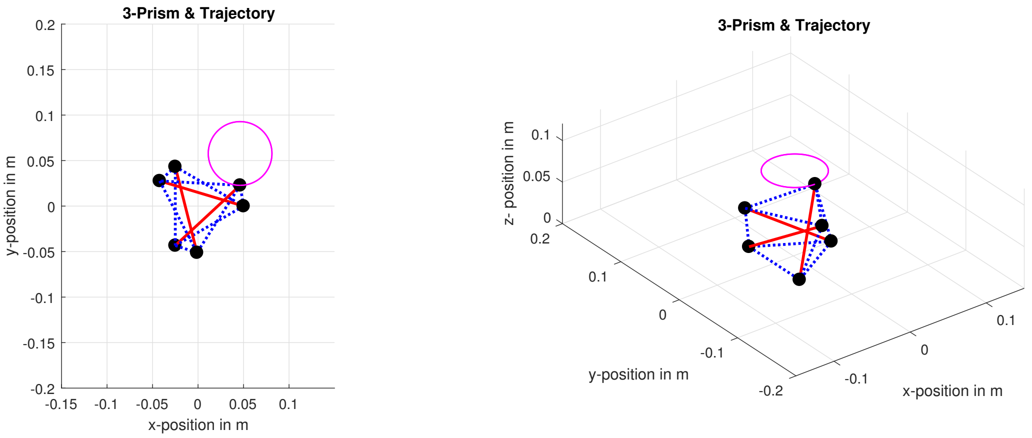
4.2. Functionality Part I

The first part of this validation demonstrates the method's functionality and practicality. It confirms that this method indeed enables the FBRP to deal with inequality constraints.

In this part, the planner had to track the circular trajectory shown in Figure 4.2. In this scenario the set of inequality constraints consisted of a maximum allowed change in cable forces between states. A total of nine constraints were defined according to convention in equation (3.5b); one for the force change of each of the structure's cables. This, for example, represents having to deal with actuator limitations. Maximum allowed force change was set to $\delta F_\mathrm{m} = 0.07\,\mathrm{N}$ based on examples in Appendix B and $F_{\mathrm{c},i}$ was defined as the force change for i^th cable. The constraints are formulated as

$$c_i \geq 0 \tag{4.1a}$$

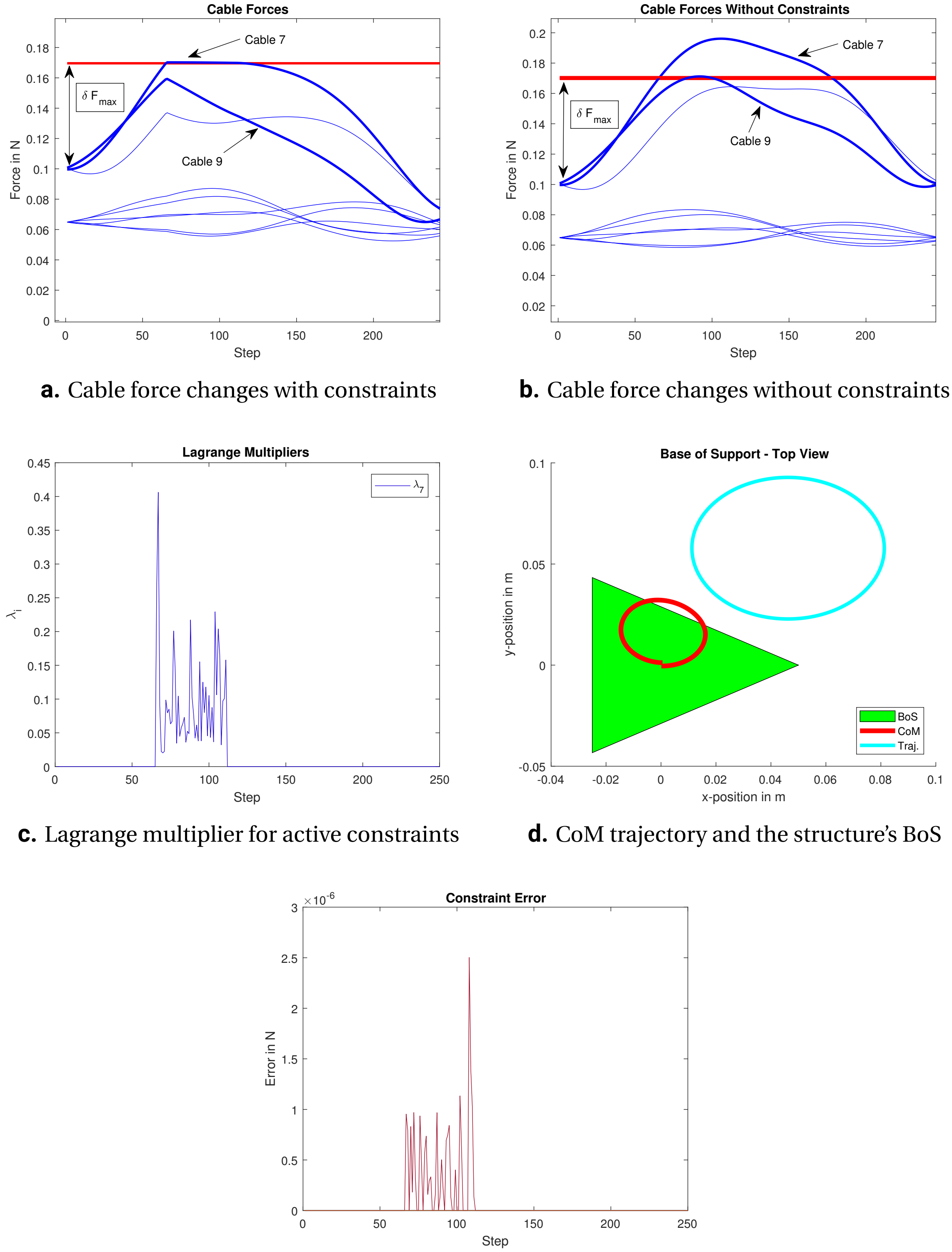
$$c_i = \delta F_\mathrm{m} - \delta F_{\mathrm{c},i}, \qquad i \in \{1,\ldots,9\}\,. \tag{4.1b}$$



a. Top view

b. Isometric view

Figure 4.2. Top and isometric views of the nodal trajectory (magenta) tracked in this validation section.



a. Cable force changes with constraints

b. Cable force changes without constraints

c. Lagrange multiplier for active constraints

d. CoM trajectory and the structure's BoS

e. Error for constraint c_7

Figure 4.3. Results where the tensegrity's cable force changes are constrained to not exceed δF_{m}. All while tracking a prescribed trajectory.

Figure 4.3a shows that only one cable was close to surpassing the force change limitations. This corresponds to one out of nine constraints being activated briefly during the trajectory tracking. In Figure 4.3b the change in force can be seen when no constraints are enforced. The Lagrange multipliers in Figure 4.3c confirm that one constraint was activated for this run. Note that on Figure 4.3d it can be seen that the tensegrity's centre of mass reaches outside the base of support.

4.3. Functionality Part II

This part of the validation was aimed at further depicting the functionality of the extended planner. This was done by enforcing a set of inequality constraints on the structure's centre of mass. Assuming that the structure's mass is distributed equally among its nodes, the centre of mass coordinates are defined as the average of the nodal coordinates in each direction

$$X_{\text{cm}} = \frac{\sum_{i=1}^{6} x_i}{6} \tag{4.2}$$

$$Y_{\text{cm}} = \frac{\sum_{i=1}^{6} y_i}{6} \tag{4.3}$$

$$Z_{\text{cm}} = \frac{\sum_{i=1}^{6} z_i}{6} \; . \tag{4.4}$$

As mentioned in Chapter 1, for a structure to maintain stability its centre of mass must stay within the confines of its base of support. The centre of mass was only constrained in the x-y plane. For this validation, a total of three constraints were defined according to convention in equation (3.5b). These are marked as c_1, c_2 & c_3 in Figure 4.4 along with the corner points of the base of support (x_1, y_1), (x_2, y_2) and (x_3, y_3). For this, the basic slope-intercept equation of a line $y = mx + b$ is utilized, where b is the y-intercept and m the line's slope. Let us define

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.5}$$

$$m_2 = \frac{y_3 - y_1}{x_3 - x_1} \tag{4.6}$$

$$b_1 = y_2 - m_1 x_2 \tag{4.7}$$

$$b_2 = y_3 - m_2 x_3 \; . \tag{4.8}$$

The three constraints can now be written as

$$c_1 \geq 0 \tag{4.9a}$$

$$c_1 = X_{\text{cm}} - x_2 \tag{4.9b}$$

$$c_2 \geq 0 \tag{4.9c}$$

$$c_2 = -Y_{\text{cm}} + b_1 + m_1 X_{\text{cm}} \tag{4.9d}$$

$$c_3 \geq 0 \tag{4.9e}$$

$$c_3 = Y_{\text{cm}} - b_2 - m_2 X_{\text{cm}} \; . \tag{4.9f}$$

These constraints were to enforce the CoM to stay within the BoS while tracking a trajectory that well reached outside that area. The trajectory can be seen in Figure 4.5.

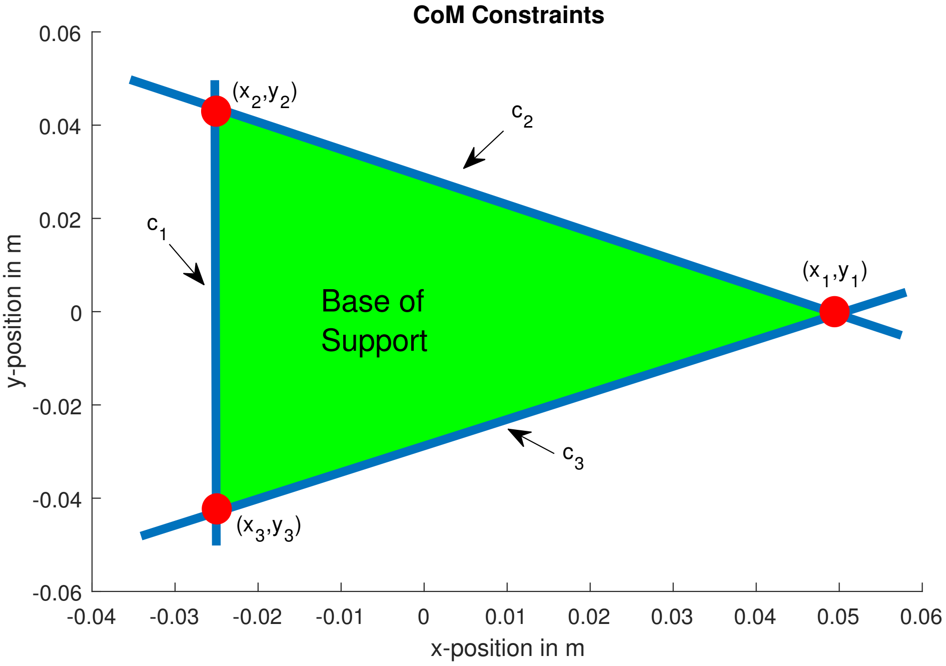
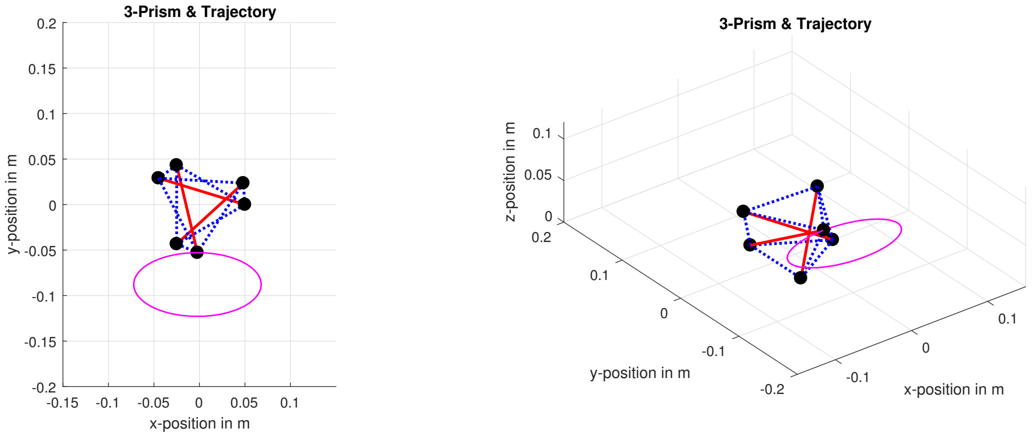


Figure 4.4. Graphical portrayal of constraints to contain the tensegrity's centre of mass within its base of support.

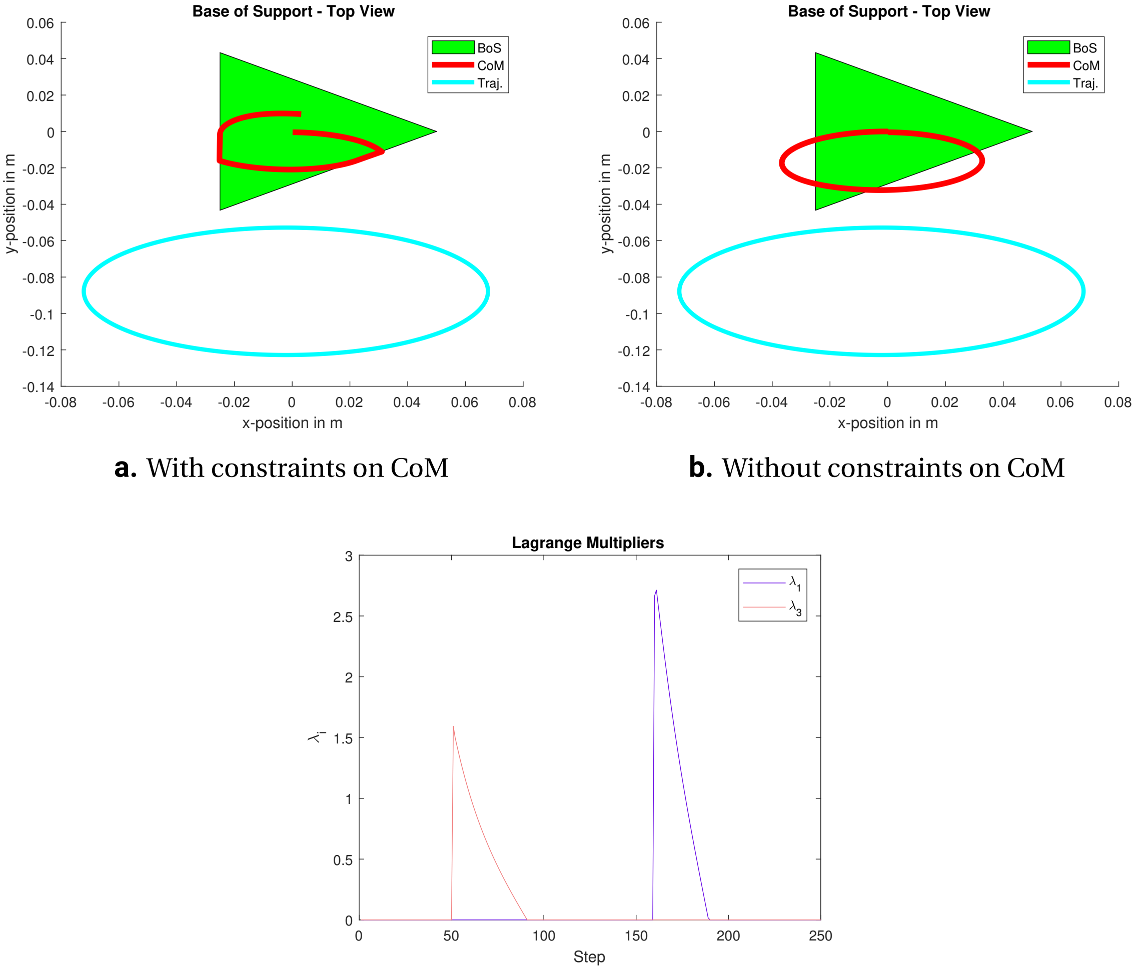


a. Top view

b. Isometric view

Figure 4.5. Top and isometric views of the nodal trajectory (magenta) tracked in this validation of the base of support constraints.

Figure 4.6a shows the results of the tracking where constraints (4.9) were enforced. The prescribed trajectory was tracked successfully while the centre of mass was confined within the base of support. As previously mentioned, if the algorithm converges then the prescribed trajectory is accurately tracked. In contrast, Figure 4.6b shows results where the trajectory is tracked without any constraints. Constraints c_1 and c_3 were active during the run and which indeed be confirmed by Lagrange multipliers in Figure 4.6c. In a case where the algorithm does not converge the problem is either deemed over constrained or the trajectory infeasible. Either way it results in high constraint errors, instability and state configurations that do not lie on the equilibrium manifold (force imbalance).



a. With constraints on CoM

b. Without constraints on CoM

c. Corresponding lagrange multipliers

Figure 4.6. Results showing the path of the strucure's centre of mass alongside the tracked trajectory. It also highlights the contrast between constraining the centre of mass and not.

4.4. Efficiency and Robustness

For the third part of the validation, a simple efficiency analysis was carried out. Furthermore, robustness of the method against slight parameter variations was tested. I did not include any repeatability validations in this chapter since the method is intrinsically deterministic i.e. the output results should be consistent given the same input parameters. However, sanity checks regarding repeatability can be found in Appendix C.

The efficiency analysis was performed by defining a measure of computational effort. This effort was quantified in terms of computational time T_c for tracking a prescribed trajectory. In other words, the sum of the time it takes to calculate a sequence of n equilibrium configurations. This measure was then compared between using the FBRP with inequality constraints and without them. This test was run on a machine utilizing Intel® Core™ i7-4710MQ @ 2.50GHz processor and 8GB RAM @ 1600Hz.

Firstly, I measured the computational time under the same conditions in Section 4.2 i.e. the trajectory on Figure 4.2. Both with and without the constraints in (4.1). Secondly, I ran the same measurements under the conditions detailed in Section 4.3. Computational time was collected 20 times and averaged across these 20 runs. This was to mitigate variations due to processor and memory availability at each time. The results from these measurements are listed in Table 4.3.

Table 4.3. Results for computational time.

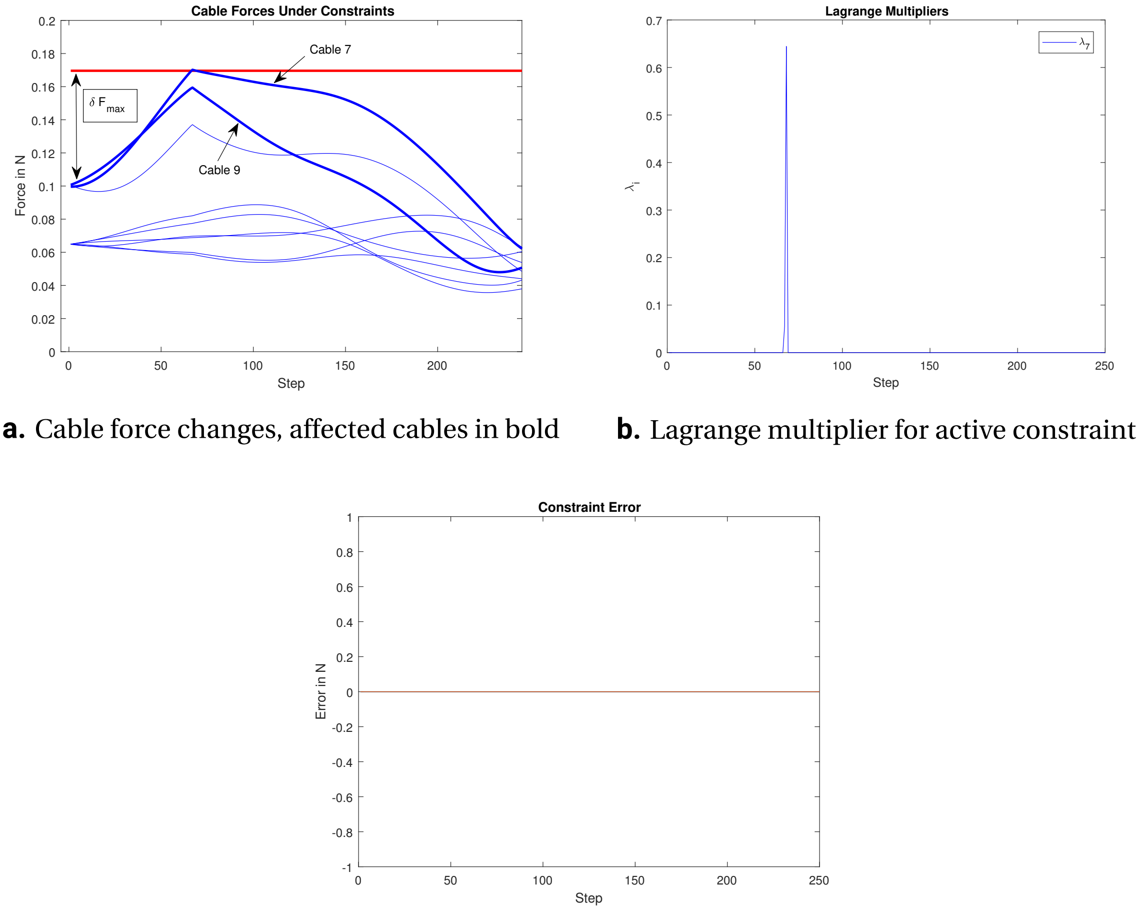
Section	***4.2***		***4.3***	
Constraints	No	Yes	No	Yes
T_c (in s)	9.38	991.97	8.67	507.36

Secondly, the robustness of the method was tested by slightly varying input parameters in Table 4.2. This validation was to get an indication of the method's reliability during typical usage. I used the same conditions as in Section 4.2 i.e. the trajectory on Figure 4.2 and the constraints in (4.1). The parameters n_s, τ and i_m were increased and decreased by 2% respectively. Table 4.4 shows the parameter values with slight deviation applied. The expectations were that the method would prove successful regardless of the slight variations in the optimization tuning parameters. The parameter n_s stands for the amount of discrete equilibrium configuration the prescribed trajectory is divided into and increasing it will cause each step to become smaller. This was expected to provide more accurate results due to the first order approximation in equation (2.9) of the FBRP. This equation is a linear approximation of a system of equations made up of force equilibrium equations and equality constraints. This estimate becomes more precise with smaller step sizes. With an increase in maximum number of iterations and constraint error termination criteria, the expectations were to obtain greater accuracy. That is mainly due to the stricter termination criteria. The algorithm generally does not require many iterations if the optimization problem is feasible. With a decrease in parameter values, lowered accuracy was expected. Fewer steps or greater step size will cause equation (2.9) to become less precise along with the termination criteria being more lenient.

Table 4.4. Parameter variation values.

	+2%	**−2%**
$\mathbf{n}_s$	255	245
τ	$5.1 \cdot 10^{-5}$	$4.9 \cdot 10^{-5}$
$\mathbf{i}_m$	1020	980

The results from increasing input parameters by 2% can be seen in Figure 4.7. They indeed confirm the method being successful regardless of the slight parameter deviation. They also show a decrease in constraint error. Figure 4.7c shows that there is no constraint error. Moreover, Figure 4.8 depicts results from decreasing input parameters by 2%. Again they confirm the method being successful regardless of the slight parameter deviation. An increase in constraint error can also be observed. This increase in error can be seen when Figure 4.8c is compared to Figures 4.3e and 4.7c.

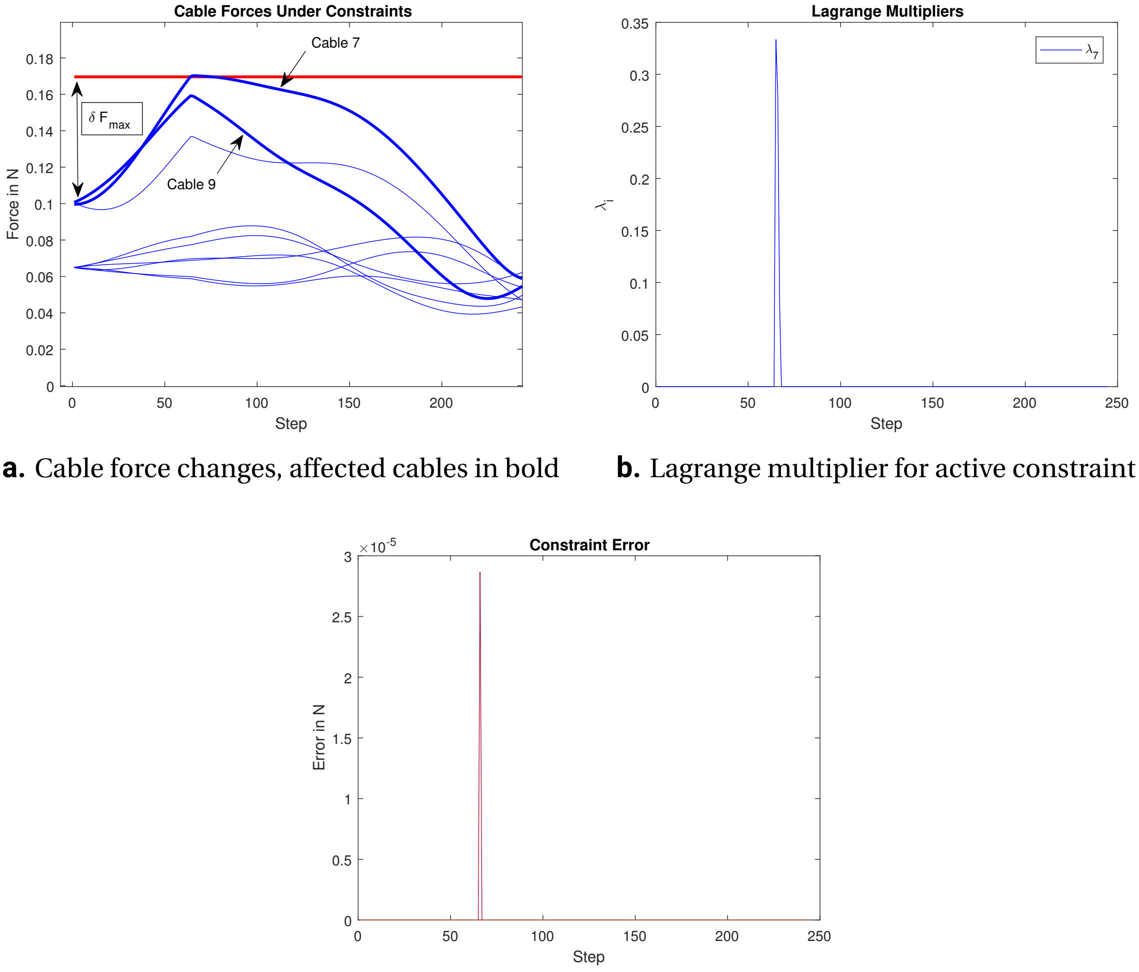


a. Cable force changes, affected cables in bold

b. Lagrange multiplier for active constraint

c. Error for constraint c_7

Figure 4.7. Results from executing the extended planner with 2% increase in input parameters, showing the cable force changes, Lagrange multipliers and constraint error.



a. Cable force changes, affected cables in bold

b. Lagrange multiplier for active constraint

c. Error for constraint c_7

Figure 4.8. Results from executing the extended planner with 2% decrease in input parameters, showing the cable force changes, Lagrange multipliers and constraint error.

4.5. Accuracy

For the fourth and final part of the validation the aim was to depict accuracy of the extended planner. The accuracy of the method was quantified in terms of the constraint error. This was observed for various nodal trajectories. Two trajectories with a radius of 0.035 m and 0.05 m respectively were chosen for each of the three top nodes (4, 5 and 6) to track. These trajectories can be observed in Figures 4.9 - 4.11. The constraints from equation (4.9) were utilized for this validation. The graphical representation can be seen as before in Figure 4.4.

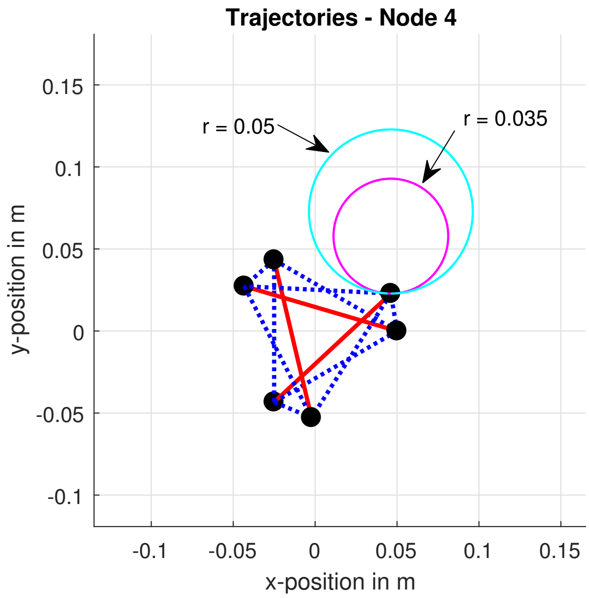


Figure 4.9. Top view of the two trajectories tracked on node 4 for validation of the method's accuracy.

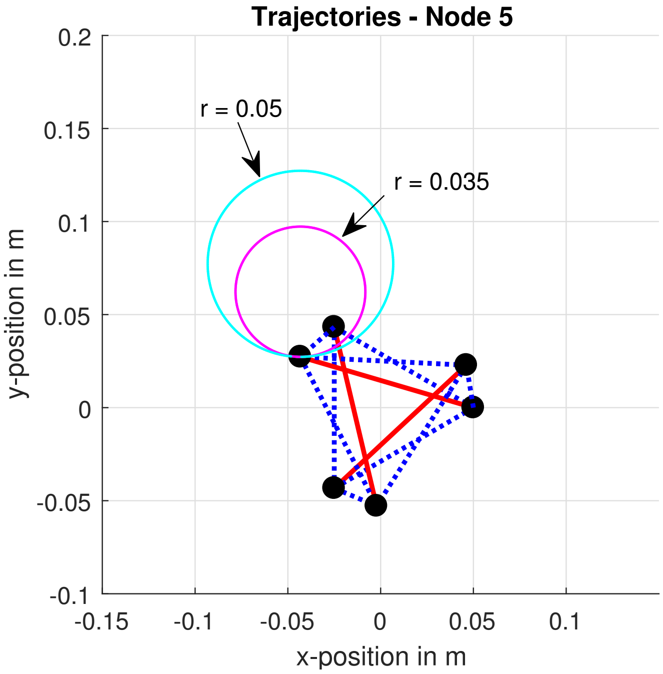


Figure 4.10. Top view of the two trajectories tracked on node 5 for validation of the method's accuracy.

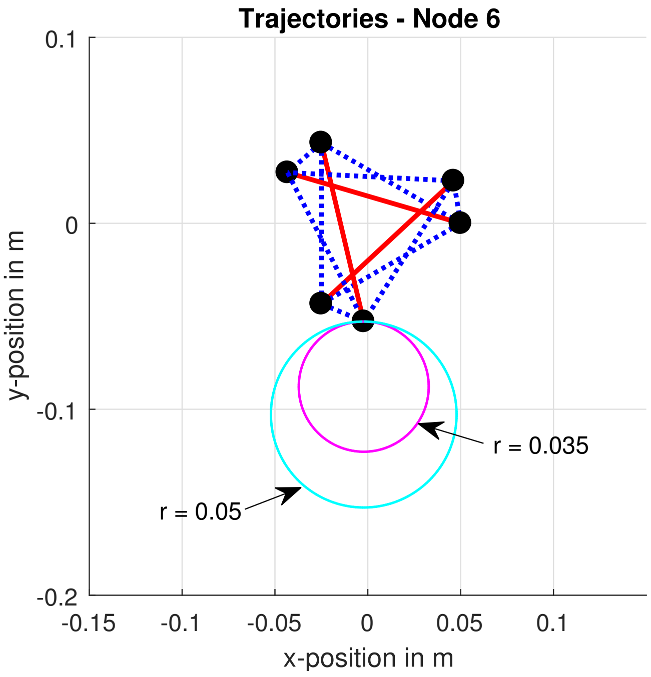
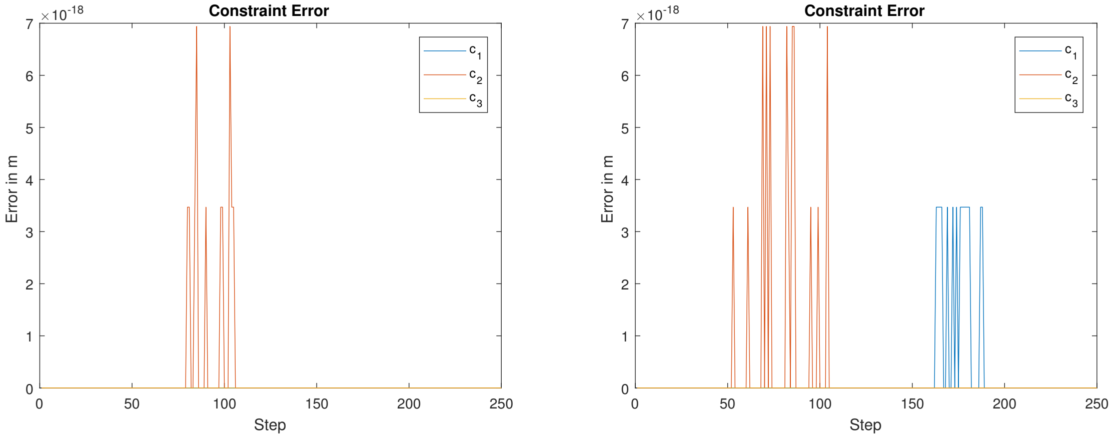


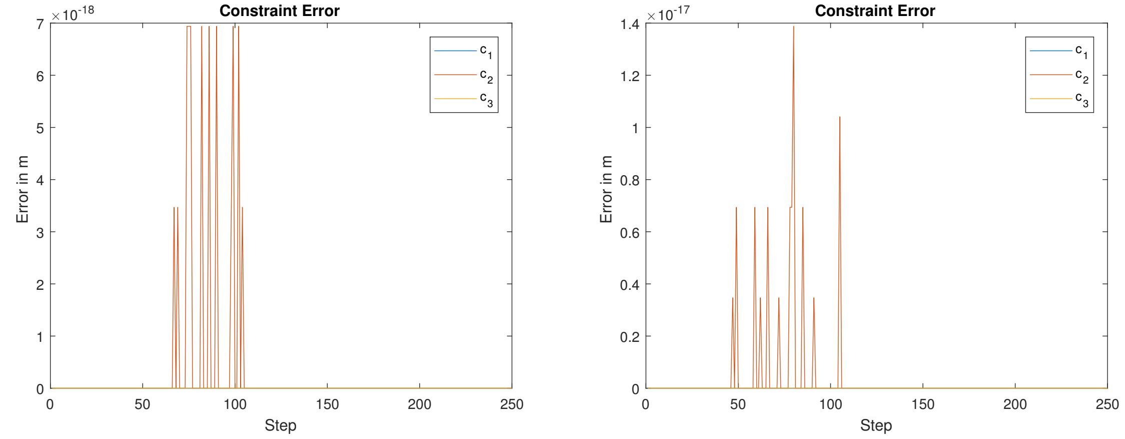
Figure 4.11. Top view of the two trajectories tracked on node 6 for validation of the method's accuracy.



a. Results with trajectory radius of 0.035 m

b. Results with trajectory radius of 0.05 m

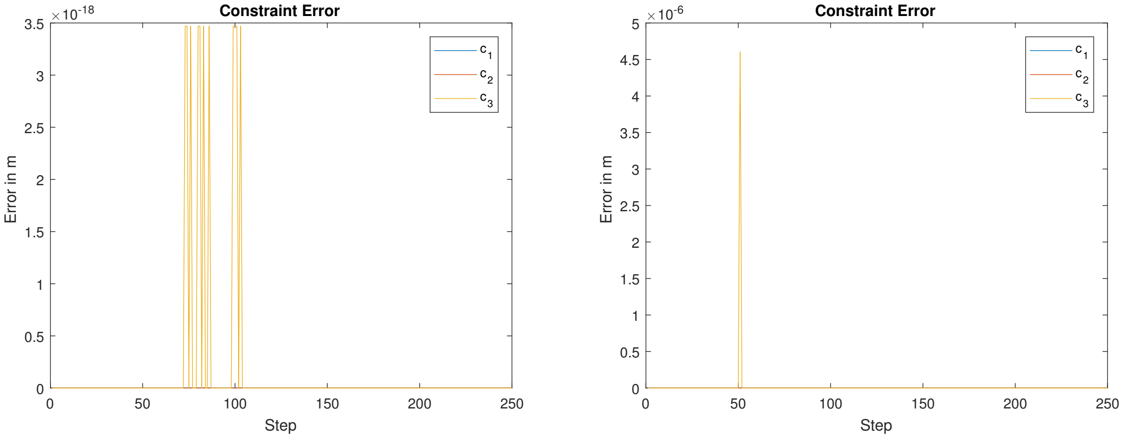
Figure 4.12. Constraint errors for trajectory tracking on node 4 while enforcing constraints on the tensegrity's centre of mass.



a. Results with trajectory radius of 0.035 m

b. Results with trajectory radius of 0.05 m

Figure 4.13. Constraint errors for trajectory tracking on node 5 while enforcing constraints on the tensegrity's centre of mass.



a. Results with trajectory radius of 0.035 m

b. Results with trajectory radius of 0.05 m

Figure 4.14. Constraint errors for trajectory tracking on node 6 while enforcing constraints on the tensegrity's centre of mass.

5 Discussion

This chapter will recapitulate each area of the validation process along with providing a discussion on the results. Furthermore, limitations of the method detailed in Chapter 3 are discussed and disputed.

5.1. Functionality Part I

In Section 4.2 the aim was to depict functionality of the extended planner in a practical application. The example used was setting limits on allowed force changes in the cable elements. The constraints were based on actuator limitations obtained in Appendix B. For the chosen trajectory 4.2 it can be seen that only one out of the nine cables is affected by the constraints. This particular cable is stretched the most and is the only one pushing to exceed the maximum force change. This is confirmed by looking at the Lagrange multiplier values throughout the sequence in Figure 4.3c. An interesting comparison is made between using constraints and no constraints in Figures 4.3a-4.3b. The latter shows force changes for a run without constraints. It is clearly visible from the graph that cable 7 would have exceeded the force change limitations. It can also be seen that constraint activation on that cable steered cable 9 away from approaching maximum force change. These results give us the confidence to plan a sequence of equilibrium configurations while respecting for example actuators limitations by enforcing inequality constraints.

5.2. Functionality Part II

In Section 4.3 the aim was to show a different type of practical functionality to that depicted in Section 4.2. This was done by setting constraints on the tensegrity's centre of mass which entailed keeping it within the structure's base of support. These constraints are visualized in Figure 4.4. Keeping the centre of mass within the base of support is a critical part in maintaining stability while generating a sequence of equilibrium configurations. The trajectory chosen can be observed in Figure 4.5. On Figure 4.6a, it can be seen that two constraints, c_1 and c_3, were activated at separate times. It is interesting to compare this to tracking the same trajectory without any constraints. It clearly shows in Figure 4.6b that the centre of mass ventures outside the base of support. In reality this would cause instability in the structure. The activation of the two constraints can indeed be confirmed by looking at the Lagrange multiplier values throughout the sequence in Figure 4.6c. These results show that enforcing inequality constraints on the centre of mass can aid in ensuring stability of the tensegrity structure. It also gives us confidence in utilizing the extended planner for practical applications in tensegrity robots.

5.3. Efficiency and Robustness

In Section 4.4 efficiency and robustness of the extended planner was validated. The efficiency was quantified in terms of computational time which was then measured both with and without constraints. This was done with conditions from both Section 4.2 and 4.3. The results showed that the computational cost of using constraints was far greater then not using them. For the conditions set in Section 4.2, the computational cost was approximately 100 times greater when enforcing constraints. As for the conditions set in Section 4.3, the cost was about 60 times higher when enforcing constraints. Some of this difference can be attributed to code implementation and time could be reduced by refactoring the code. However, these results do confirm that there is a substantial difference in computational time when enforcing constraints. This was expected to an extent since the SQP method is an iterative optimization procedure.

The robustness of the method to small parameter variations was validated by varying parameters n_s, τ and i_m. They were increased and decreased by 2% respectively. The results show that the method is robust enough to handle input parameter variations. It appears that decrease in parameters causes growth in constraint error whereas an increase in parameters reduces error. This can be explained by several factors. Firstly, if maximum number of iterations and constraint error termination criteria are increased it will force more iterations along with a stricter accuracy criteria. This results in lower constraint error. Decreasing those parameters will have an opposite effect or cause reduction in accuracy. Secondly, the FBRP uses a first order approximation in (2.9) which means that smaller steps result in better accuracy, i.e., the approximation is more accurate. Incrementing the parameter n_s means increasing the amount of steps for a trajectory which will cause each step to become smaller.

5.4. Accuracy

In section 4.5 the aim was to demonstrate the accuracy of the method. This was done by tracking two different trajectories for nodes 4, 5 and 6. Constraints were enforced on the structure's centre of mass and accuracy was quantified in terms of constraint error. It was expected to consistently obtain values less than the set SQP termination criteria $\tau = 5 \cdot 10^{-5}$. In Figures 4.12 - 4.14 it can be observed that indeed for all different trajectories the error is less than set tolerance. Based on these results we can be confident in the method's accuracy. It can be noted that for a trajectory with $r = 0.05\,\mathrm{m}$ on node 6 (Fig. 4.14b) the error is greater compared to other cases. This can be explained by the fact that this trajectory, compared to others, causes the most strain on the constraints. That is due to the amount of mass shifting over the precipice of the base of support. Possibly a larger trajectory would push the method to its limits and cause eventual failure. However, that would indicate an infeasible and impractical trajectory within the current set of constraints.

5.5. Method Limitations

5.5.1. Stability

A limitation of the proposed methodology is that a desired change in state variables may result in a configuration that does not lie on the equilibrium manifold. This occurs when a prescribed trajectory is infeasible. This phenomenon can be overcome by adaptive trajectory correction, i.e., adjusting the trajectory dynamically to avoid configurations that result in an unstable tensegrity. Another limitation of force density methods in general is that it can be difficult to predict the outcome for a prescribed set of force densities as they are not necessarily always intuitive quantities [20, 40, 53].

5.5.2. Efficiency

If the tensegrity structure that is being controlled has many elements the process of finding a solution becomes very intricate. This does not severely affect the full body reference planner as it is computationally effective. However, this is an issue for the SQP method, namely due to a computationally expensive constraint gradient matrix. In this code implementation of the SQP extension it can increase computational time a hundredfold. However, the FBRP assumes a quasi-static process and therefore large computation times are not of major concern.

5.5.3. Maratos Effect

A known limiting behaviour of the SQP method is often called the Maratos effect [36]. This behaviour is characterized by the SQP method repeatedly rejecting steps that would make significant advancements towards a solution. This occurs due to the curvature of the constraints not being sufficiently represented when linearized in the SQP model. This phenomenon is addressed in Section 3.2.1 by a second order correction technique applied to a line search method.

5.5.4. Gradient Rank Requirement

Another known limitation of SQP methods is the condition of a full rank constraint gradient matrix. However, this is addressed in Section 3.3.5 by adding and removing constraints from the QP subproblem.

6 Conclusion

The research goal of this thesis was to enhance and build upon a Full Body Reference Planner, a strategy aimed at bringing tensegrity robots closer to controlled full body movements. The enhancement was to implement a robust way to deal with inequality constraints within the planner, i.e., to compute an equilibrium configuration that satisfies certain constraints. Before, only equality constraints could be incorporated. The enhancement was realized by a Sequential Quadratic Programming optimization for each step detailed in Chapter 3.

6.1. Validation Conclusions

This approach was validated using a numerical implementation detailed in Chapter 4. For the purpose of clarity, the validation was applied to the simplest form of tensegrity, a 3-prism. The validation consisted of two different practical applications, efficiency and robustness tests and finally accuracy affirmation.

The first practical application was applying inequality constraints on the maximum force change allowed in the cable elements of the tensegrity robot. This was to represent a scenario where one could take into account actuator limitations when controlling a tensegrity robot. The second practical application was to constrain the centre of mass of a tensegrity structure within its base of support. This was to illustrate a scenario where stability of the structure would need to be ensured. Both of these validations produced results that support this method being applied in practice.

Efficiency of the method was considered by measuring computational time of the extended FBRP with and without inequality constraints. In its current form, the SQP method did cause a substantial increase in computational time. This was expected due to the iterative nature of the method and the current non-optimized code implementation.

Robustness of the method to parameter variations was tested by running previous tests with different input parameters. The method proved to be successful in enforcing inequality constraints with various input parameters. However, there were some notable but explainable changes in accuracy.

Lastly, accuracy of the method was quantified in terms of constraint error. Various nodal trajectories were tracked while enforcing constraints on the tensegrity structure's centre of mass. For all trajectories in question the constraint errors were within the set tolerance, affirming the accuracy of the approach.

6.2. General Conclusions

In accordance with the set goal of this thesis we now have means of applying the Full Body Reference Planner for tensegrity nodal control while enforcing inequality constraints. Although the proposed method has some limitations such as computational efficiency it does bring us a step closer to controlled full body movements for tensegrity robots in practice.

Bibliography

[1] JB Aldrich, RE Skelton, and K Kreutz-Delgado. Control synthesis for a class of light and agile robotic tensegrity structures. In *American Control Conference, 2003. Proceedings of the 2003*, volume 6, pages 5245–5251. IEEE, 2003.

[2] BenFrantzDale. An svg of a physically possible tensegrity structure in 3d, with a shadow., 2008. URL `https://commons.wikimedia.org/wiki/File:3-tensegrity.svg`. Under Creative Commons Attribution 3.0.

[3] Thomas Bliss, Tetsuya Iwasaki, and Hilary Bart-Smith. Central pattern generator control of a tensegrity swimmer. *IEEE/ASME Transactions on Mechatronics*, 18(2):586–597, 2013.

[4] Valter Böhm and Klaus Zimmermann. Vibration-driven mobile robots based on single actuated tensegrity structures. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5475–5480. IEEE, 2013.

[5] Bob Burkhardt. Tensegrity icosahedron, 1994. URL `https://commons.wikimedia.org/wiki/File:Tensegrity_Icosahedron.png`. Under Creative Commons Attribution 2.5.

[6] Bob Burkhardt. Tensegrity tetrahedron, 2004. URL `https://commons.wikimedia.org/wiki/File:Tensegrity_Tetrahedron.png`. Under Creative Commons Attribution 3.0.

[7] Ken Caluwaerts, Jérémie Despraz, Atıl Işçen, Andrew P Sabelhaus, Jonathan Bruce, Benjamin Schrauwen, and Vytas SunSpiral. Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface*, 11(98):20140520, 2014.

[8] Antonello Cherubini, Giacomo Moretti, Rocco Vertechy, and Marco Fontana. Experimental characterization of thermally-activated artificial muscles based on coiled nylon fishing lines. *AIP Advances*, 5(6):067158, 2015.

[9] Ching-Ping Chou and Blake Hannaford. Measurement and modeling of mckibben pneumatic artificial muscles. *IEEE Transactions on robotics and automation*, 12(1): 90–102, 1996.

[10] Steve Collis. Brisbane, 2011. URL `https://upload.wikimedia.org/wikipedia/commons/d/dd/Brisbane_%286868660143%29.jpg`. Under Creative Commons Attribution 2.0.

[11] Geoffrey HG Dyson, BD Woods, and PR Travers. *The mechanics of athletics*. Holmes & Meier Publishers, 1986.

[12] G Gomez Estrada, H-J Bungartz, and Camilla Mohrdieck. Numerical form-finding of tensegrity structures. *International Journal of Solids and Structures*, 43(22-23):6855–6868, 2006.

[13] 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)*, pages 4222–4228. IEEE, 2014.

[14] 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)*, pages 2123–2128. IEEE, 2016.

[15] R Buckminster Fuller. *Synergetics*. Pacific Tape Library, 1975.

[16] Philip E Gill and Walter Murray. Numerically stable methods for quadratic programming. *Mathematical programming*, 14(1):349–372, 1978.

[17] Philip E Gill and Elizabeth Wong. Sequential quadratic programming methods. In *Mixed integer nonlinear programming*, pages 147–224. Springer, 2012.

[18] Donald Goldfarb and Ashok Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

[19] Gregor Gregorcic. The singular value decomposition and the pseudoinverse. *matrix*, 2(1):2, 2001.

[20] RB Haber and JF Abel. Initial equilibrium solution methods for cable reinforced membranes part i—formulations. *Computer Methods in Applied Mechanics and Engineering*, 30(3):263–284, 1982.

[21] Carter S Haines, Márcio D Lima, Na Li, Geoffrey M Spinks, Javad Foroughi, John DW Madden, Shi Hyeong Kim, Shaoli Fang, Mônica Jung de Andrade, Fatma Göktepe, et al. Artificial muscles from fishing line and sewing thread. *science*, 343(6173):868–872, 2014.

[22] Shinichi Hirai, Yuusuke Koizumi, Mizuho Shibata, Minghui Wang, and Li Bin. Active shaping of a tensegrity robot via pre-pressure. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 19–25. IEEE, 2013.

[23] Cheng-yu Hong, Alice Agogino, and Edward A Lee. Viability of tensegrity robots in space exploration. 2014.

[24] Donald E Ingber. Tensegrity: the architectural basis of cellular mechanotransduction. *Annual review of physiology*, 59(1):575–599, 1997.

[25] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. Controlling tensegrity robots through evolution. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1293–1300. ACM, 2013.

[26] Atil Iscen, Adrian Agogino, Vytas SunSpiral, and Kagan Tumer. Robust distributed control of rolling tensegrity robot. In *The Autonomous Robots and Multirobot Systems (ARMS) Workshop at AAMAS*, volume 2013, 2013.

[27] M James. The generalised inverse. *The Mathematical Gazette*, 62(420):109–114, 1978.

[28] Mark Khazanov, Ben Humphreys, William Keat, and John Rieffel. Exploiting dynamical complexity in a physical tensegrity robot to achieve locomotion. *Advances in Artificial Life, ECAL*, 12:965–972, 2013.

[29] Mark Khazanov, Jules Jocque, and John Rieffel. Evolution of locomotion on a physical tensegrity robot. In *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, pages 232–238, 2014.

[30] Kyunam Kim, Adrian K Agogino, Deaho Moon, Laqshya Taneja, Aliakbar Toghyan, Borna Dehghani, Vytas SunSpiral, and Alice M Agogino. Rapid prototyping design and control of tensegrity soft robot for locomotion. In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pages 7–14. IEEE, 2014.

[31] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294, 2013.

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

[33] Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Traces and emergence of nonlinear programming*, pages 247–258. Springer, 2014.

[34] Arthur D Kuo. An optimal control model for analyzing human postural balance. *IEEE transactions on biomedical engineering*, 42(1):87–101, 1995.

[35] John DW Madden, Nathan A Vandesteeg, Patrick A Anquetil, Peter GA Madden, Arash Takshi, Rachel Z Pytel, Serge R Lafontaine, Paul A Wieringa, and Ian W Hunter. Artificial muscle technology: physical principles and naval prospects. *IEEE Journal of oceanic engineering*, 29(3):706–728, 2004.

[36] Nicholas Maratos. Exact penalty function algorithms for finite dimensional and control optimization problems. 1978.

[37] Milenko Masic and Robert E Skelton. Open-loop control of class-2 tensegrity towers. In *Smart Structures and Materials*, pages 298–308. International Society for Optics and Photonics, 2004.

[38] MATLAB. *version 9.3.0 (R2017b)*. The MathWorks Inc., Natick, Massachusetts, 2017.

[39] Jorge Nocedal and Stephen J Wright. Sequential quadratic programming. *Numerical optimization*, pages 529–562, 2006.

[40] Thouraya Nouri-Baranger. Computational methods for tension-loaded structures. *Archives of computational methods in Engineering*, 11(2):143, 2004.

[41] Kazuhiro Otsuka and Clarence Marvin Wayman. *Shape memory materials*. Cambridge university press, 1999.

[42] Liviu Panait. Theoretical convergence guarantees for cooperative coevolutionary algorithms. *Evolutionary computation*, 18(4):581–615, 2010.

[43] Aftab Patla, James Frank, and David Winter. Assessment of balance control in the elderly: major issues. *Physiotherapy Canada*, 42(2):89–97, 1990.

[44] Jean-Paul Pinaud, Soren Solari, and Robert E Skelton. Deployment of a class 2 tensegrity boom. In *Smart Structures and Materials 2004: Smart Structures and Integrated Systems*, volume 5390, pages 155–163. International Society for Optics and Photonics, 2004.

[45] Josep M Porta and Sergi Hernández-Juan. Path planning for active tensegrity structures. *International Journal of Solids and Structures*, 78:47–56, 2016.

[46] Michael JD Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis*, pages 144–157. Springer, 1978.

[47] Andrew P Sabelhaus, Jonathan Bruce, Ken Caluwaerts, Yangxin Chen, Dizhou Lu, Yuejia Liu, Adrian K Agogino, Vytas SunSpiral, and Alice M Agogino. Hardware design and testing of superball, a modular tensegrity robot. 2014.

[48] 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)*, pages 2867–2873. IEEE, 2015.

[49] Claudia Schmid and Lorenz T Biegler. Quadratic programming methods for reduced hessian sqp. *Computers & chemical engineering*, 18(9):817–832, 1994.

[50] Mizuho Shibata, Fumio Saijyo, and Shinichi Hirai. Crawling by body deformation of tensegrity structure robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4375–4380. IEEE, 2009.

[51] Kenneth Snelson. Snelson on the tensegrity invention. *International Journal of Space Structures*, 11(1-2):43–48, 1996.

[52] Cornel Sultan, Martin Corless, and Robert E Skelton. Tensegrity flight simulator. *Journal of Guidance, Control, and Dynamics*, 23(6):1055–1064, 2000.

[53] Kia Yick Tan. The computer design of tensile membrane structures. 1989.

[54] Gunnar Tibert. *Deployable tensegrity structures for space applications*. PhD thesis, KTH, 2002.

[55] 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*, pages 261–267. IEEE, 2013.

[56] G.A. Tournois. Tensegrity locomotion on rough terrain. diploma thesis, Delft University of Technology, June 2017.

[57] Hoang Chi Tran and Jaehong Lee. Form-finding of tensegrity structures with multiple states of self-stress. *Acta mechanica*, 222(1-2):131, 2011.

[58] Andreas Wächter and Lorenz T Biegler. Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization*, 16(1):32–48, 2005.

[59] Lianjun Wu, Monica Jung de Andrade, Tarang Brahme, Yonas Tadesse, and Ray H Baughman. A deformable robot with tensegrity structure using nylon artificial muscle. In *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pages 97993K–97993K. International Society for Optics and Photonics, 2016.

[60] Michael C Yip and Günter Niemeyer. On the control and properties of supercoiled polymer artificial muscles. *IEEE Transactions on Robotics*, 2017.

[61] Xingfei Yuan, Xiaotian Liang, and Along Li. Shape and force control of prestressed cable-strut structures based on nonlinear force method. *Advances in Structural Engineering*, 19(12):1917–1926, 2016.

[62] JY Zhang and M Ohsaki. Adaptive force density method for form-finding problem of tensegrity structures. *International Journal of Solids and Structures*, 43(18-19):5658–5673, 2006.

A

Null-Space Method

This appendix outlines the Null-Space procedure based on implementations by Gill et al. [16] and Nocedal & Wright [39]. It is used to solve the KKT system in every iteration of the active-set method. Given the following KKT system of a quadratic program

$$\begin{bmatrix} \mathbf{G} & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} , \tag{A.1}$$

where $\mathbf{C}$ is a $n \times m$ matrix with full column rank. By definition the QR decomposition of $\mathbf{C}$ is

$$\mathbf{C} = \mathbf{Q} \begin{bmatrix} \boldsymbol{R} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{Y} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \boldsymbol{R} \\ 0 \end{bmatrix} , \tag{A.2}$$

where $\mathbf{Z}$ is the $n \times (n-m)$ null space matrix of $\mathbf{C}$ and $\mathbf{Y}$ is any $n \times m$ matrix to ensure $\mathbf{Q}$ is nonsingular. This method does not require the Hessian matrix $\mathbf{G}$ to be nonsingular. Let us presume that the vector $\boldsymbol{p}$ can be partitioned

$$\boldsymbol{p} = \mathbf{Y}\boldsymbol{p}_y + \mathbf{Z}\boldsymbol{p}_z \, . \tag{A.3}$$

If we substitute this into (A.1) and use the fact that $\mathbf{C}^T\mathbf{Z} = 0$ as per $\mathbf{Z}$ being the null space of $\mathbf{C}$. Now the second equation becomes

$$\mathbf{C}^T\mathbf{Y}\boldsymbol{p}_y = \boldsymbol{b} \, , \tag{A.4}$$

where $\boldsymbol{p}_y$ can be obtained via back substitution due to the aforementioned full rank and nonsingular properties of $\mathbf{C}$ and $\mathbf{Y}$. The second equation of the KKT system is

$$\mathbf{G}\mathbf{Y}\boldsymbol{p}_y + \mathbf{G}\mathbf{Z}\boldsymbol{p}_z - \mathbf{C}\boldsymbol{v} = -\boldsymbol{a} \, . \tag{A.5}$$

If we multiply by $\mathbf{Z}^T$ and again use the fact that $\mathbf{C}^T\mathbf{Z} = 0$, we obtain

$$\left(\mathbf{Z}^T\mathbf{G}\mathbf{Z}\right)\boldsymbol{p}_z = -\mathbf{Z}^T\left(\mathbf{G}\mathbf{Y}\boldsymbol{p}_y + \boldsymbol{a}\right)\ . \tag{A.6}$$

now that we know $\boldsymbol{p}_y$, $\boldsymbol{p}_z$ can be solved for via Cholesky factorization of the reduced Hessian matrix $\mathbf{Z}^T\mathbf{G}\mathbf{Z}$. The reduced Hessian is positive definite which ensures local optima. Finally to obtain $\boldsymbol{v}$ we multiply the first equation of (A.1) by $\mathbf{Y}^T$:

$$\mathbf{Y}^T\mathbf{C}\boldsymbol{v} = \mathbf{Y}^T\mathbf{G}\mathbf{Y}\boldsymbol{p}_y + \mathbf{Y}^T\mathbf{G}Z\boldsymbol{p}_z + \mathbf{Y}^T\boldsymbol{a} \tag{A.7}$$

$$\Rightarrow \mathbf{Y}^T\mathbf{C}\boldsymbol{v} = \mathbf{Y}^T\left(\mathbf{G}\boldsymbol{p} + \boldsymbol{a}\right) \tag{A.8}$$

which again can be solved via back substitution. Now the vector $\begin{bmatrix}\boldsymbol{p} & \boldsymbol{v}\end{bmatrix}^T$ has been obtained which concludes this Null-Space procedure.

B
Actuators

This appendix describes an experiment to indicate physical limitations of actuators used in tensegrity robots. The twisted and coiled polymer muscle is used as an example. They have good applicability in tensegrities as they can serve not only as actuators but also as passive structural elements. TCPM have also shown to have low manufacturing cost, next to no hysteretic behaviour and a high cycle life [21]. Experiments have demonstrated that it can exceed the maximum tensile stroke of skeletal muscles in vivo by approximately 20% [21, 35]. However, TCPM are known to have low speed of contraction and low efficiency in terms of thermal actuation is [8]. Generally, there exist two types of twisted and coiled polymer muscles which are distinguished by the means of achieving coils in the precursor fibre (nylon fibre):

- Autocoiling: inserting twist in the precursor fibre to achieve coiling due to mechanical instability
- Mandrel coiling: inserting twist in the precursor fibre right until mechanical instability is reached, then wrapping the fibre around a mandrel

To have an idea of the limitations of TCPM in practical application let us look closer at mandrel coiled muscles. They are known to have certain limitations depending on production method, material selection, means of actuation and geometry. Mandrel coiled muscles have reduced load carrying capacity compared to autocoiled but can achieve larger stroke due to distance between adjacent coils [21]. An experiment was conducted to characterize the force increase for a mandrel coiled muscle by means of Joule-Heating. For a given power input the TCPM will reach a steady-state force when the internal rate of heating matches the external rate of cooling [60]. If a given steady state temperature surpasses the heat deflection temperature of the precursor fibre the TCPM is damaged. If one were to apply a mandrel coiled muscle in a tensegrity robot it would be useful to incorporate safety limits of the actuator in the control strategy. In this section the production method of a mandrel wrapped TCPM is described. Next, experimental details are recounted and finally results are interpreted.

B.1. Manufacturing of Twisted & Coiled Polymer Muscles

The TCPM is constructed of two materials, a 1 m long Nylon 6 fibre with a diameter of 0.6 mm and an iron resistance wire of same length and 0.2 mm diameter. Fiber and wire are tied together at both ends. One end is attached to a rotary mechanism driven by a drilling machine and the other to a 300 g weight which is suspended in air with a pulley system. This enforces, by gravitational force, a constant tension in the fibre and wire. The constant tension is to prevent entangling of the fibre and resistance wire. Next, the drilling machine is driven to insert twist. That is done until autocoiling occurs which is when maximum turns per unit length is reached.

The twisted fibre and wire are then wrapped around a mandrel while keeping the aforementioned tension applied. The direction of wrapping determines whether the muscle will be homochiral or heterochiral. When heated, homochiral muscles contract but heterochiral elongate [8]. For this project, homochiral muscles were produced. The coil diameter of the produced muscle is defined by the mandrel diameter. After the fibre and wire have been securely wrapped and attached to the mandrel they are thermally annealed. Annealing is performed in the following manner:

- One hour in a traditional oven at 155 °C.
- One hour in a traditional fridge at 6 °C.

B.2. Experiment

The equipment used for this experiment is as follows:

- Mark-10 Tensile Tester with a force gauge.
- CALTEK PSA30/3B power supply.
- One TCPM manufactured as described.
- PCB board for current control.

The muscle is actuated under isometric strain, i.e. the length is kept constant while it is Joule-Heated. The displacement d is varied from $20-55$ mm with 5 mm increments. First the muscle is placed in the tensile tester while slacked. Then it is stretched to the desired displacement and allowed to rest for 180 s to mitigate dynamic effects. Finally a 12 V direct current is fed through at 30% Duty Cycle for 120 s, this procedure is repeated for a 60% Duty Cycle. Force data is collected from the tensile tester force gauge.

B.3. Experimental Results

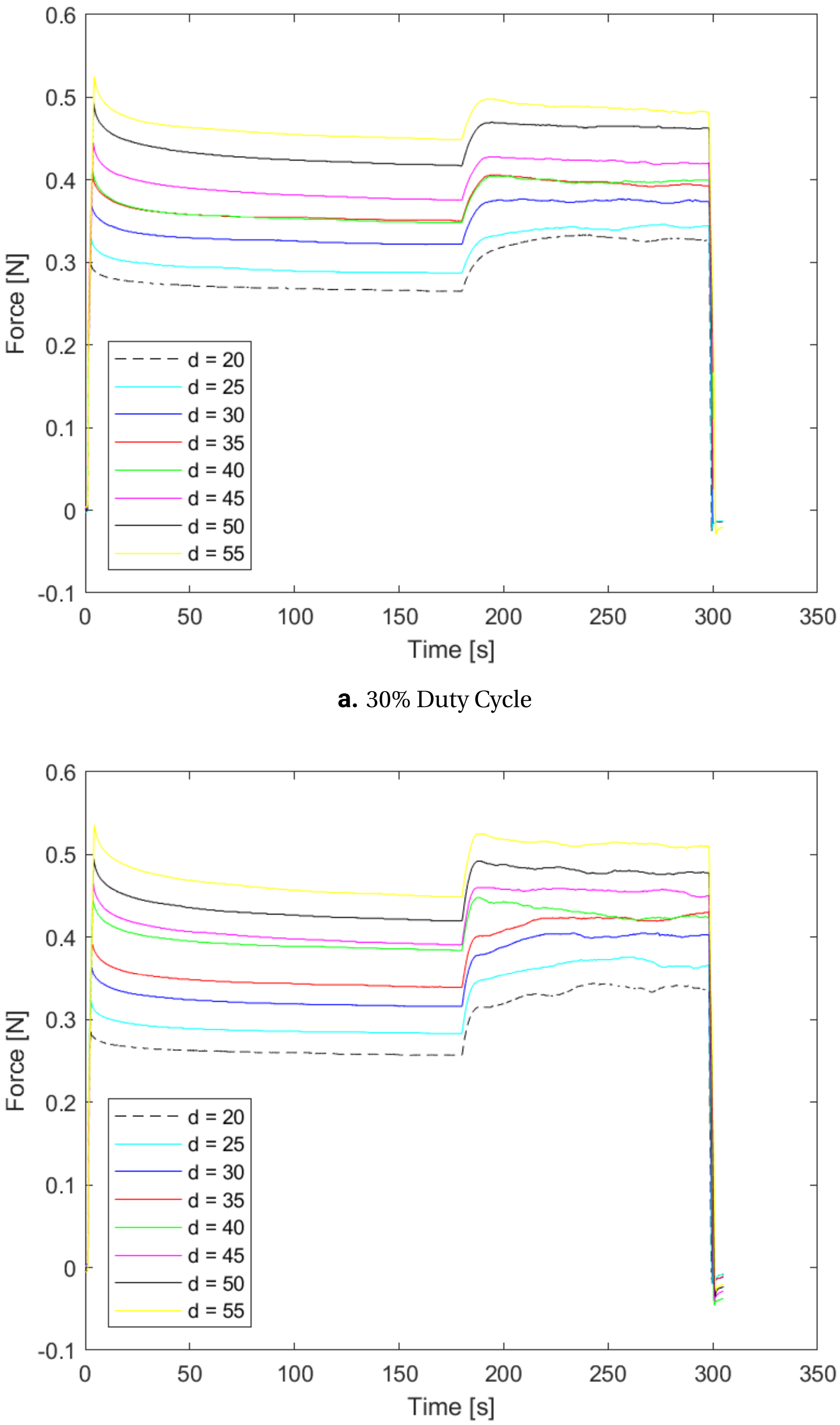
Figures B.1a-B.1b show force measurements for 30% Duty Cycle and 60% Duty Cycle respectively. As can be observed there is a saturation of force change after a certain Joule-Heating on-time. It can also be observed that for 30% Duty Cycle the force does reach steady-state whereas for 60% Duty Cycle the force curve does fluctuate substantially.

Table B.1 shows the maximum force change achieved by Joule-Heating at 30% Duty Cycle and 60% Duty Cycle at 12 V direct current. As can be seen a greater change in force is achieved at larger Duty Cycle. The maximum change at 30% Duty Cycle is 0.068 N and 0.093 N at 60% Duty Cycle.

Table B.1. Maximum force change achieved through Joule-Heating of the TCPM

30% Duty Cycle @ 12V								
d (in mm)	20	25	30	35	40	45	50	55
δF_{m} (in N)	0.068	0.059	0.056	0.056	0.057	0.053	0.053	0.049

60% Duty Cycle @ 12V								
d (in mm)	20	25	30	35	40	45	50	55
δF_{m} (in N)	0.087	0.093	0.089	0.085	0.064	0.069	0.072	0.077



a. 30% Duty Cycle

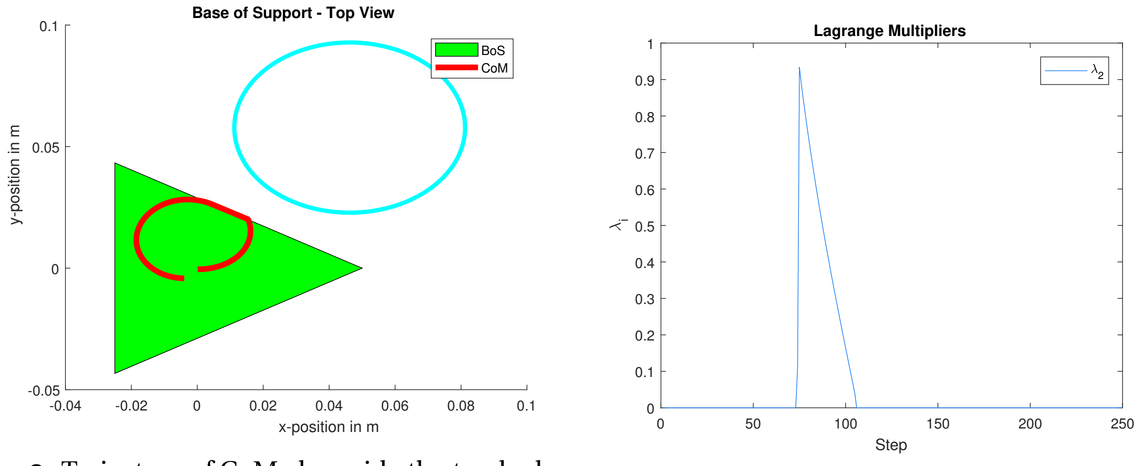
b. 60% Duty Cycle

Figure B.1. Force-Time relationship of Joule-Heated TCPM at different displacement

C Validation

This appendix details repeatability sanity checks. To determine repeatability the method was run ten times with the same input variables, i.e. a fixed set of parameter settings, starting configuration and trajectory to track. The goal was to achieve the same outcome every run to confirm the method's deterministic nature. Input variables were set as defined in Table 4.2 and the outcome variables monitored were no. of iterations, Lagrange multipliers, objective function value and constraint errors. The trajectory tracked was the one seen in Figure 4.2. The constraints used were defined to confine the structure's centre of mass within its base of support.

For each of the ten runs, the trajectory was divided in 250 steps. As can be seen in Figure C.1 only constraint 2 is active for this tracked trajectory. As was expected across ten runs with the same set of input variables the output results were indeed the same, underpinning the deterministic nature of the algorithm. A table comparing results for each step across all runs can be seen in Appendix C. These tables report values for steps where constraints were active.



a. Trajectory of CoM alongside the tracked trajectory. Showing constraints in effect.

b. Corresponding lagrange multiplier.

Figure C.1. Result of a single repeatability run. Showing tracked trajectory and activated constraint.

The following tables contain the variables; constraint error, objective function value, Lagrange multiplier value and iteration counts for both the SQP and active set method.

Table C.1. Constraint error across 10 runs. Only shown for operative constraint g_2 when active.

	Runs									
Steps	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18	-6.94E-18
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18
	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18	-3.47E-18

Table C.2. Objective function values across 10 runs. Only shown when constraint was active.

	Runs									
Steps	2.12E-06	2.12E-06	2.12E-06	2.12E-06	2.12E-06	2.12E-06	2.12E-06	2.12E-06	2.12E-06	2.12E-06
	1.52E-04	1.52E-04	1.52E-04	1.52E-04	1.52E-04	1.52E-04	1.52E-04	1.52E-04	1.52E-04	1.52E-04
	1.40E-04	1.40E-04	1.40E-04	1.40E-04	1.40E-04	1.40E-04	1.40E-04	1.40E-04	1.40E-04	1.40E-04
	1.30E-04	1.30E-04	1.30E-04	1.30E-04	1.30E-04	1.30E-04	1.30E-04	1.30E-04	1.30E-04	1.30E-04
	1.20E-04	1.20E-04	1.20E-04	1.20E-04	1.20E-04	1.20E-04	1.20E-04	1.20E-04	1.20E-04	1.20E-04
	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04
	1.02E-04	1.02E-04	1.02E-04	1.02E-04	1.02E-04	1.02E-04	1.02E-04	1.02E-04	1.02E-04	1.02E-04
	9.43E-05	9.43E-05	9.43E-05	9.43E-05	9.43E-05	9.43E-05	9.43E-05	9.43E-05	9.43E-05	9.43E-05
	8.65E-05	8.65E-05	8.65E-05	8.65E-05	8.65E-05	8.65E-05	8.65E-05	8.65E-05	8.65E-05	8.65E-05
	7.91E-05	7.91E-05	7.91E-05	7.91E-05	7.91E-05	7.91E-05	7.91E-05	7.91E-05	7.91E-05	7.91E-05
	7.21E-05	7.21E-05	7.21E-05	7.21E-05	7.21E-05	7.21E-05	7.21E-05	7.21E-05	7.21E-05	7.21E-05
	6.55E-05	6.55E-05	6.55E-05	6.55E-05	6.55E-05	6.55E-05	6.55E-05	6.55E-05	6.55E-05	6.55E-05
	5.93E-05	5.93E-05	5.93E-05	5.93E-05	5.93E-05	5.93E-05	5.93E-05	5.93E-05	5.93E-05	5.93E-05
	5.34E-05	5.34E-05	5.34E-05	5.34E-05	5.34E-05	5.34E-05	5.34E-05	5.34E-05	5.34E-05	5.34E-05
	4.79E-05	4.79E-05	4.79E-05	4.79E-05	4.79E-05	4.79E-05	4.79E-05	4.79E-05	4.79E-05	4.79E-05
	4.27E-05	4.27E-05	4.27E-05	4.27E-05	4.27E-05	4.27E-05	4.27E-05	4.27E-05	4.27E-05	4.27E-05
	3.78E-05	3.78E-05	3.78E-05	3.78E-05	3.78E-05	3.78E-05	3.78E-05	3.78E-05	3.78E-05	3.78E-05
	3.33E-05	3.33E-05	3.33E-05	3.33E-05	3.33E-05	3.33E-05	3.33E-05	3.33E-05	3.33E-05	3.33E-05
	2.91E-05	2.91E-05	2.91E-05	2.91E-05	2.91E-05	2.91E-05	2.91E-05	2.91E-05	2.91E-05	2.91E-05
	2.51E-05	2.51E-05	2.51E-05	2.51E-05	2.51E-05	2.51E-05	2.51E-05	2.51E-05	2.51E-05	2.51E-05
	2.15E-05	2.15E-05	2.15E-05	2.15E-05	2.15E-05	2.15E-05	2.15E-05	2.15E-05	2.15E-05	2.15E-05
	1.82E-05	1.82E-05	1.82E-05	1.82E-05	1.82E-05	1.82E-05	1.82E-05	1.82E-05	1.82E-05	1.82E-05
	1.52E-05	1.52E-05	1.52E-05	1.52E-05	1.52E-05	1.52E-05	1.52E-05	1.52E-05	1.52E-05	1.52E-05
	1.24E-05	1.24E-05	1.24E-05	1.24E-05	1.24E-05	1.24E-05	1.24E-05	1.24E-05	1.24E-05	1.24E-05
	9.96E-06	9.96E-06	9.96E-06	9.96E-06	9.96E-06	9.96E-06	9.96E-06	9.96E-06	9.96E-06	9.96E-06
	7.77E-06	7.77E-06	7.77E-06	7.77E-06	7.77E-06	7.77E-06	7.77E-06	7.77E-06	7.77E-06	7.77E-06
	5.85E-06	5.85E-06	5.85E-06	5.85E-06	5.85E-06	5.85E-06	5.85E-06	5.85E-06	5.85E-06	5.85E-06
	4.21E-06	4.21E-06	4.21E-06	4.21E-06	4.21E-06	4.21E-06	4.21E-06	4.21E-06	4.21E-06	4.21E-06
	2.84E-06	2.84E-06	2.84E-06	2.84E-06	2.84E-06	2.84E-06	2.84E-06	2.84E-06	2.84E-06	2.84E-06
	1.74E-06	1.74E-06	1.74E-06	1.74E-06	1.74E-06	1.74E-06	1.74E-06	1.74E-06	1.74E-06	1.74E-06
	9.08E-07	9.08E-07	9.08E-07	9.08E-07	9.08E-07	9.08E-07	9.08E-07	9.08E-07	9.08E-07	9.08E-07
	3.44E-07	3.44E-07	3.44E-07	3.44E-07	3.44E-07	3.44E-07	3.44E-07	3.44E-07	3.44E-07	3.44E-07

Table C.3. Iteration count for SQP across 10 runs. Only shown when constraint was active.

	Runs									
Steps	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000

Table C.4. Iteration count for active set across 10 runs. Only shown when constraint was active.

	Runs									
Steps	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000

Table C.5. Lagrange multiplier λ_2 values across 10 runs. Only shown when constraint was active.

	Runs									
Steps	0.1105	0.1105	0.1105	0.1105	0.1105	0.1105	0.1105	0.1105	0.1105	0.1105
	0.9339	0.9339	0.9339	0.9339	0.9339	0.9339	0.9339	0.9339	0.9339	0.9339
	0.8882	0.8882	0.8882	0.8882	0.8882	0.8882	0.8882	0.8882	0.8882	0.8882
	0.8493	0.8493	0.8493	0.8493	0.8493	0.8493	0.8493	0.8493	0.8493	0.8493
	0.8115	0.8115	0.8115	0.8115	0.8115	0.8115	0.8115	0.8115	0.8115	0.8115
	0.7749	0.7749	0.7749	0.7749	0.7749	0.7749	0.7749	0.7749	0.7749	0.7749
	0.7393	0.7393	0.7393	0.7393	0.7393	0.7393	0.7393	0.7393	0.7393	0.7393
	0.7046	0.7046	0.7046	0.7046	0.7046	0.7046	0.7046	0.7046	0.7046	0.7046
	0.6709	0.6709	0.6709	0.6709	0.6709	0.6709	0.6709	0.6709	0.6709	0.6709
	0.6381	0.6381	0.6381	0.6381	0.6381	0.6381	0.6381	0.6381	0.6381	0.6381
	0.6060	0.6060	0.6060	0.6060	0.6060	0.6060	0.6060	0.6060	0.6060	0.6060
	0.5748	0.5748	0.5748	0.5748	0.5748	0.5748	0.5748	0.5748	0.5748	0.5748
	0.5442	0.5442	0.5442	0.5442	0.5442	0.5442	0.5442	0.5442	0.5442	0.5442
	0.5142	0.5142	0.5142	0.5142	0.5142	0.5142	0.5142	0.5142	0.5142	0.5142
	0.4849	0.4849	0.4849	0.4849	0.4849	0.4849	0.4849	0.4849	0.4849	0.4849
	0.4561	0.4561	0.4561	0.4561	0.4561	0.4561	0.4561	0.4561	0.4561	0.4561
	0.4278	0.4278	0.4278	0.4278	0.4278	0.4278	0.4278	0.4278	0.4278	0.4278
	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
	0.3727	0.3727	0.3727	0.3727	0.3727	0.3727	0.3727	0.3727	0.3727	0.3727
	0.3457	0.3457	0.3457	0.3457	0.3457	0.3457	0.3457	0.3457	0.3457	0.3457
	0.3191	0.3191	0.3191	0.3191	0.3191	0.3191	0.3191	0.3191	0.3191	0.3191
	0.2928	0.2928	0.2928	0.2928	0.2928	0.2928	0.2928	0.2928	0.2928	0.2928
	0.2667	0.2667	0.2667	0.2667	0.2667	0.2667	0.2667	0.2667	0.2667	0.2667
	0.2409	0.2409	0.2409	0.2409	0.2409	0.2409	0.2409	0.2409	0.2409	0.2409
	0.2154	0.2154	0.2154	0.2154	0.2154	0.2154	0.2154	0.2154	0.2154	0.2154
	0.1900	0.1900	0.1900	0.1900	0.1900	0.1900	0.1900	0.1900	0.1900	0.1900
	0.1648	0.1648	0.1648	0.1648	0.1648	0.1648	0.1648	0.1648	0.1648	0.1648
	0.1397	0.1397	0.1397	0.1397	0.1397	0.1397	0.1397	0.1397	0.1397	0.1397
	0.1147	0.1147	0.1147	0.1147	0.1147	0.1147	0.1147	0.1147	0.1147	0.1147
	0.0897	0.0897	0.0897	0.0897	0.0897	0.0897	0.0897	0.0897	0.0897	0.0897
	0.0648	0.0648	0.0648	0.0648	0.0648	0.0648	0.0648	0.0648	0.0648	0.0648
	0.0399	0.0399	0.0399	0.0399	0.0399	0.0399	0.0399	0.0399	0.0399	0.0399