Chapter 3

Tensegrity Dynamics and Simulation

3.1 Introduction

This work primarily investigates the canonical six-bar spherical topology; in this chapter we explore in greater detail the dynamics model representations of this topology which are useful for controls and state estimation in later chapters. Notably, this tensegrity design is a *Class-1* tensegrity structure, meaning that in an idealized model, each node is attached to (at most) one rod and, equivalently, that the rods and cables act only as two-force members under pure compression or pure tension respectively. Class-1 tensegrities are special tensegrity structures constructed with compressive bodies which bear no rigid joints and which are interconnected solely through series-elastic tensile elements. For this reason, Class-1 tensegrity dynamic equations of motion are well-structured and can be procedurally obtained when given a fixed set of model parameters. Forming the rigid-body dynamics of the tensegrity system can therefore be simply achieved in a procedural manner once the physical parameters (i.e., rod mass, rod length, cable stiffness) and rod and cable connections are known, thus making tensegrity structures well-suited for model-based optimal control.

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

As an active robotic system, the six-bar spherical tensegrity robot has the ability to control the rod lengths and/or the tensions of the actuated cables. By convention, as in [4, 3, 11], it is common for individual cable tensions to be controlled by using motors to spool in (retract) or extend (release) a cable's restlength. Using this principal and a simple application of Hooke's law, cable tensions can be readily manipulated, with special attention to slack cables which, by their nature, can produce no compressive force.

For the six-bar topology, the *neutral pose* (i.e., the geometry when tension is equal for

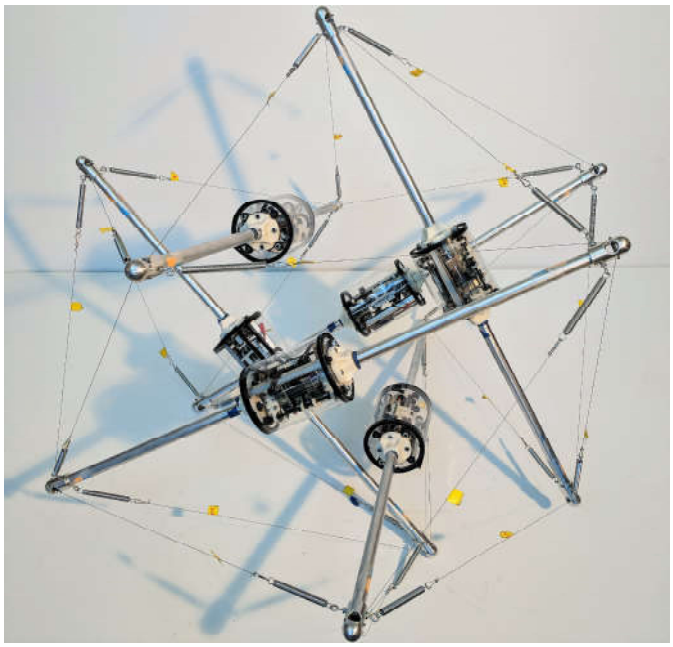

Figure 3.1: A robot hardware example of the canonical six-bar spherical tensegrity topology, TT-5 from the Berkeley Emergent Space Tensegrities research lab.

each of the 24 elastic cables) has a convex hull enclosing the 12 endpoints of the rods in the shape of an icosahedron with 20 triangular-shaped faces. Through manipulation of the tensions of each of the individual cables, the robot can deform its geometry to achieve dynamic motion. This flexible shapeshifting capability is key to the design's versatility and enables the tensegrity robot to perform rolling locomotion by shifting the center of mass of the robot in a desired manner, as described in [6].

3.2 Contributions of this Chapter

TO-DO: complete after writing sections.

3.3 Six-Bar Rigid Body Dynamics

In this section, I introduce a dynamics model representation used in my earlier research, first presented by Wroldsen and Skelton (cite). Although, I include the previous work here for completeness, this dynamic formulation was eventually abandoned in favor of the simpler but slightly less accurate point mass formulation introduced in Section 3.4. The rigid body formulation introduced here has the advantage of accurately representing rotational inertia and being a minimal representation (i.e., having the fewest number of states necessary to

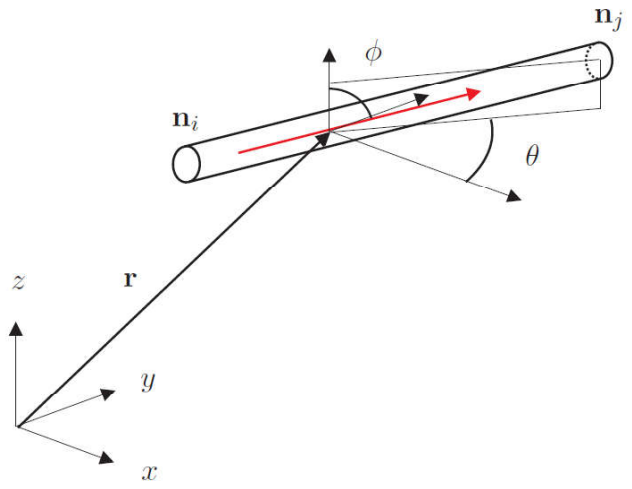

Figure 3.2: Minimal state representation of a single rod rigid body, using XYZ-coordinates of the rod's center of mass and spherical angles representing the directional unit vector that runs along the rod's axial direction. Figure by Anders Wroldsen presented in [10].

represent the full dynamical system). However, as I discuss towards the end of the section in Subsection 3.3.4, this minimal representation suffers from dynamic singularities and stiff ordinary differential equations which make model-based receding horizon control difficult or impossible outside of short time-horizons or well-conditioned cases.

3.3.1 Rod and Cable Dynamics

In the minimal representation motivated by [7, 10] and presented in this section, each rod is represented by five states: x, y, z position of its center of mass and the azimuth and polar angles of a unit vector which lies along the axial direction of the rod (see Fig. 3.2). Traditionally, rotation around the rod's longitudinal axis is less important and often unconstrained in the physical construction of the robot and is thus consequently neglected in this model. The equations of motion of rod i with respect to the minimal q-coordinate representation can therefore be represented as:

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

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

$$\mathbf{M}_i = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_i \in \mathbb{R}^{5\times5} \qquad \mathbf{H}_i(\mathbf{q}_i) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Jsin^2\phi} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{J} \end{bmatrix}_i \in \mathbb{R}^{5\times5}$$

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

Individual cable tensions, t_i, can be modeled using a simple linear series-elastic representation of Hooke's Law, with the added 'slackness' constraint that tensile forces are strictly non-negative (Eq. 3.2). Collision forces between two rods and contact forces between rods and the ground are calculated using a similar piecewise linear Hooke contact model in compression.

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

here t_i is the individual cable tension, s_i is the separation distance between the endpoints of the rods which cable i is connected to, and ℓ_i is the cable restlength which can be controlled (extended or retracted) using actuators.

3.3.2 Complete Tensegrity Robot Dynamics

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

To demonstrate this process for the six-bar spherical topology, we first designate each rod endpoint as a 'node' with index $j \in \{1, ..., 12\}$, which permits a beginning or end of a cable or rod connection to be defined. In convention with other existing simulations in [2, 5] from the Berkeley Emergent Space Tensegrities lab, the six rods are defined in this work with respect to the defined nodes as follows:

rod	1	2	3	4	5	6
node	1,2	4,3	5,6	8,7	9,12	10,11

A connectivity matrix, $\mathbf{C} \in \mathbb{R}^{24\times12}$, is consequently defined (example for TT-5 robot provided below), where each row corresponds to a cable and each column corresponds to one of the 12 nodes previously defined above. Cable connections are defined on each row by assigning a value of 1 or -1 (there must only be one of each; order determines axial unit vector direction) for each nodal endpoint and 0 elsewhere.

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{24\times\ 12} \tag{3.3}$$

The full system robot dynamics can then be formed by simply concatenating each of the independent rod dynamics:

$$\ddot{\mathbf{q}}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) = \mathbf{M}^{-1}(\mathbf{q})\Big(\mathbf{H}(\mathbf{q})\big(\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})\big)\Big) \tag{3.4}$$

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

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

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

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

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

$$\partial W = \mathbf{F}_1 \cdot \sum_{i=1}^{m} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{y}_i} \partial \boldsymbol{y}_i + \mathbf{F}_2 \cdot \sum_{i=1}^{m} \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{y}_i} \partial \boldsymbol{y}_i + ... + \mathbf{F}_n \cdot \sum_{i=1}^{m} \frac{\partial \boldsymbol{x}_n}{\partial \boldsymbol{y}_i} \partial \boldsymbol{y}_i \tag{3.5}$$

$$= \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{y}_1} \partial \boldsymbol{y}_1 + \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{y}_2} \partial \boldsymbol{y}_2 + ... + \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{y}_m} \partial \boldsymbol{y}_m \tag{3.6}$$

By definition, the virtual work of the system can also be equivalently written as:

$$\partial W = Q_1 \partial \boldsymbol{y}_1 + Q_2 \partial \boldsymbol{y}_i + ... + Q_m \partial \boldsymbol{y}_m \tag{3.7}$$

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

$$Q_i = \sum_{j=1}^{n} \mathbf{F}_j \cdot \frac{\partial \boldsymbol{x_j}}{\partial \boldsymbol{y_i}} \qquad i = \{1, ..., m\} \tag{3.8}$$

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

$$\mathbf{F}_{cables} = -\frac{\partial \boldsymbol{\gamma}(\mathbf{q})^{\mathbf{T}}}{\partial \mathbf{q}} \sum_{i=1}^{24} (\mathbf{C}_i^T \mathbf{C}_i \otimes \mathbf{I_3}) \boldsymbol{\gamma}(\mathbf{q}) t_i = \mathbf{B}(\mathbf{q})\mathbf{T} \tag{3.9}$$

$$\boldsymbol{\gamma}(\mathbf{q}) = \begin{bmatrix} x_1 & + & \frac{L}{2} cos\theta_1 sin\phi_1 \\ y_1 & + & \frac{L}{2} sin\theta_1 sin\phi_1 \\ z_1 & + & \frac{L}{2} cos\phi_1 \\ x_1 & - & \frac{L}{2} cos\theta_1 sin\phi_1 \\ y_1 & - & \frac{L}{2} sin\theta_1 sin\phi_1 \\ z_1 & - & \frac{L}{2} cos\phi_1 \\ & \vdots & \\ x_6 & - & \frac{L}{2} cos\theta_1 sin\phi_6 \\ y_6 & - & \frac{L}{2} sin\theta_1 sin\phi_6 \\ z_6 & - & \frac{L}{2} cos\phi_6 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ X_2 \\ Y_2 \\ Z_2 \\ \vdots \\ X_{12} \\ Y_{12} \\ Z_{12} \end{bmatrix} \in \mathbb{R}^{36} \tag{3.10}$$

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

The resulting complete acceleration dynamics can be compactly expressed as:

$$\ddot{\mathbf{q}}_\mathbf{i}(\mathbf{q_i}, \dot{\mathbf{q}}_\mathbf{i}) = \mathbf{M}^{-1}\Big(\mathbf{H}(\mathbf{q})\big(\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{T} + \mathbf{F}_{collision} + \mathbf{F}_{contact}\big)\Big) \tag{3.11}$$

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

3.3.3 Ground Contact Dynamics and Rod Collision

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

$$\mathbf{F}_{contact,i} = c \cdot \Delta_i \qquad or \qquad \mathbf{F}_{collision,j} = c \cdot (e^{\Delta_j} - 1) \tag{3.12}$$

where c is a constant scaling factor (in practice, 1e−3 works well) and Δ is a measure of interference, either between a nodal endpoint and the ground or between two different rods. For ground contact forces, Δ is measured by how far each node's Z-coordinate sits below the floor:

$$\Delta_i = max\{0,\ Z_{floor} - Z_{node,i}\} \tag{3.13}$$

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

3.3.4 Dynamic Singularities

It is important to note that while the minimal representation used in the rigid body dynamics presented in this section simplifies computational complexity by reducing the number of symbolic variables, this model suffers from singularities at $\phi = 0$ and $\phi = \pi$, where the rod is nearly vertical. In these cases, azimuthal rotational velocities ($\dot{\theta}$) and accelerations ($\ddot{\theta}$) can become excessively large. As a result, matrix norms of the Jacobian used for the linearized dynamics become excessively large and can lead to large linearization errors even for very small timesteps.

To address this, ϕ and θ values substituted into the pre-calculated symbolic Jacobian matrix were bounded to provide a buffer around these singularity points. In practice, a $\pm 8 \times 10^{-2}$ rad (4.6 deg) for ϕ around the vertical orientation and a 1 $\frac{rad}{s}$ bound on the angular velocity $\dot{\theta}$ works well, without sacrificing too much accuracy. Nevertheless, the dynamic singularities motivate a different representation that is simple, versatile, and applicable for a wide range of tensegrity geometries. (todo: add graph of Jacobian norm and condition number vs. phi angle)

3.4 Simplified Point Mass Dynamics

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

For this point mass formulation, rather than representing true rigid body dynamics, we assume that the entirety of each rod's mass can be distributed between two point masses located at the ends of the rod, hereafter also refered to as nodes. Admittedly, this assumption's validity is largely dependent on the actual hardware design of the tensegrity robot; as an

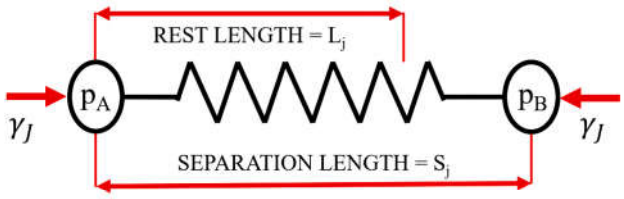

Figure 3.3: Hookean linear-elastic model between two point masses.

example, consider the tensegrity SUPERball [8] designed by the Intelligent Robotics Group at NASA Ames, which carries most of its mass closer to the ends of the rod, where heavier motor assemblies and electronics are housed. Thus, this point mass assumption can often be relatively accurate and greatly simplifies the formulation of tensegrity dynamic equations of motion, enabling rapid design and prototyping of new innovative topologies in simulation.

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

$$\mathbf{p} = [X_1, Y_1, Z_1, X_2, Y_2, Z_2, ..., X_N, Y_N, Z_N]^{\mathbf{T}} \tag{3.14}$$

$$\dot{\mathbf{p}} = [\dot{X}_1, \dot{Y}_1, \dot{Z}_1, \dot{X}_2, \dot{Y}_2, \dot{Z}_2, ..., \dot{X}_N, \dot{Y}_N, \dot{Z}_N]^{\mathbf{T}} \tag{3.15}$$

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

3.4.1 Series-Elastic Cable Forces

Forces which act on the nodes due to the spring-cables are calculated similarly to the approach presented in Section 3.3.1 simply using Hookean approximations, with special consideration that no compressive forces can be applied through the cables:

$$\mathbf{F}_{cables,j} = max\{\ 0\ ,\ k_j(s_j - \ell_j)\ \} \tag{3.16}$$

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

Given a cable connectivity matrix $\mathbf{C} \in \mathbb{R}^{J \times N}$ such as Eq. 3.3, with rows $\mathbf{C}_j$ that encode cable interconnections between pairs of nodes, we represent the vector of Cartesian forces due to tension in cable j, $\boldsymbol{\gamma}_j \in \mathbb{R}^{3N}$, as:

$$\boldsymbol{\gamma}_j = \sqrt{2}k_j \cdot softPlus\Big(\boldsymbol{\alpha}_j, \beta\Big)\frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \quad \forall j \in \{1, ..., J\} \tag{3.17}$$

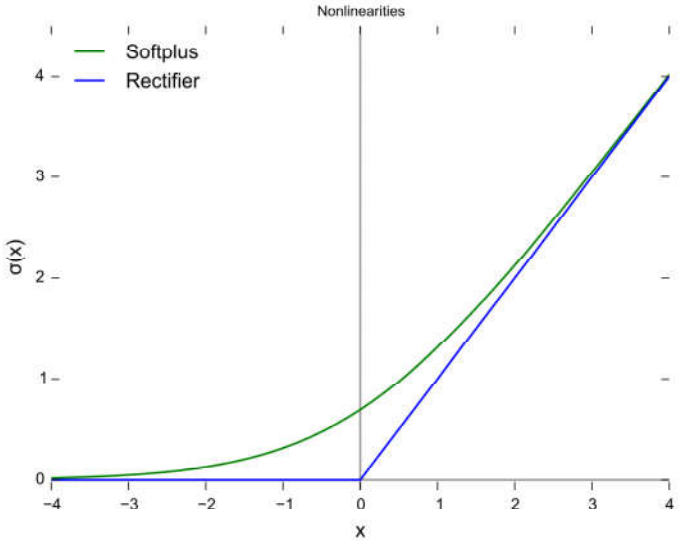

Figure 3.4: Comparison of the linear rectifier and the smooth softplus approximation. include different beta values here

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

$$\mathbf{z}_j = -\left[\mathbf{C}_j^T \mathbf{C}_j \otimes \mathbf{I}_3\right] \mathbf{p}$$

$$\boldsymbol{\alpha}_j = (s_j - \ell_j) = \sqrt{\frac{\mathbf{z}_j^T \mathbf{z}_j}{2}} - \ell_j$$

$\mathbf{z}_j \in \mathbb{R}^J$ is a sparse vector that contains the directional vector lying along the direction of the cable j. The *softPlus* function above is a smooth approximation to the non-differentiable rectifier function (used in Eq. 3.16), approximating $max\{0, \boldsymbol{\alpha}_i\}$ with tunable smoothness parameter β (see Fig. 3.4 for a comparison):

$$softPlus(\alpha, \beta) := \frac{\sqrt{\alpha^2 + \beta^2} + \alpha}{2} \geq 0 \tag{3.18}$$

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

3.4.2 Rigid Body Constraint Forces

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

Rather than model the rods using a linear-elastic model as with cable forces in the prior section, we instead adopt a constrained dynamics approach. The motivation behind this is that *penalty* or *energy barrier methods* (todo:add citation), which rely on restorative forces to maintain rigid connections, necessitate large stiffness parameters and lead to stiff differential equations. That is, numerical methods may be unstable for solving the ODEs unless step sizes are chosen to be extremely small. Instead, the constraint forces we describe here analytically cancel out the components of the applied forces that violate rigid constraints at each timestep, creating accurate and numerically tractable dynamic simulations.

In our work [1], we adopt a similar approach to [9] and define constraint vectors, $\mathbf{G}(\mathbf{p})$ and $\dot{\mathbf{G}}(\mathbf{p}) \in \mathbb{R}^m$, to represent the implicit constraint functions and their time derivatives which must be satisfied at all times, where m is the number of active dynamic constraints. Each scalar-valued function $\mathbf{G}_i(\mathbf{p})$ is a single implicit constraint function that is satisfied when equal to zero. If we assume that initial positions and velocities of the system satisfy dynamic constraints (i.e., $\mathbf{G}(\mathbf{p}) = 0$ and $\dot{\mathbf{G}}(\mathbf{p}) = 0$), then any forces which maintain legal accelerations (i.e., $\ddot{\mathbf{G}}(\mathbf{p}) = 0$) will be valid forces which satisfy all dynamic constraints.

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

$$\hat{\mathbf{F}} \cdot \dot{\mathbf{p}} = 0, \quad \forall \dot{\mathbf{p}} \;\mid\; \dot{\mathbf{G}} = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{p}}\right)\dot{\mathbf{p}} = 0 \tag{3.19}$$

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

$$\hat{\mathbf{F}} = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{p}}\right)^T \boldsymbol{\lambda} \tag{3.20}$$

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

We also introduce the inverse-mass matrix $\mathbf{W} = \mathbf{M}^{-1}$, which has the reciprocal of each particle's mass as elements along the diagonal. By applying the chain rule and substituting

for $\ddot{\mathbf{p}}$, the legal acceleration condition can be written as:

$$\ddot{\mathbf{G}}(\mathbf{p}) = \left[\frac{\partial \dot{\mathbf{G}}(\mathbf{p})}{\partial \mathbf{p}}\right] \dot{\mathbf{p}} + \left[\frac{\partial \mathbf{G}(\mathbf{p})}{\partial \mathbf{p}}\right] \ddot{\mathbf{p}} \tag{3.21}$$

$$= \left[\frac{\partial \dot{\mathbf{G}}(\mathbf{p})}{\partial \mathbf{p}}\right] \dot{\mathbf{p}} + \left[\frac{\partial \mathbf{G}(\mathbf{p})}{\partial \mathbf{p}}\right] \mathbf{W}(\mathbf{F} + \hat{\mathbf{F}}) \;=\; \mathbf{0} \tag{3.22}$$

Simplifying notation of $\frac{\partial \mathbf{G}(\mathbf{p})}{\partial \mathbf{p}}$ and $\frac{\partial \dot{\mathbf{G}}(\mathbf{p})}{\partial \mathbf{p}}$ as matrices $\mathbf{J}(\mathbf{p})$ and $\dot{\mathbf{J}}(\mathbf{p})$, respectively, and dropping the matrices' explicit dependencies on $\mathbf{p}$, we manipulate and rewrite Eq. 3.22 as:

$$\mathbf{JW}\hat{\mathbf{F}} \;=\; -\dot{\mathbf{J}}\dot{\mathbf{p}} - \mathbf{JWF} \tag{3.23}$$

$$\mathbf{JW}(\mathbf{J^T}\boldsymbol{\lambda}) \;=\; -\dot{\mathbf{J}}\dot{\mathbf{p}} - \mathbf{JWF} \tag{3.24}$$

where Eq. 3.24 is a result of applying the principle of virtual work and substituting in Eq. 3.20. The vector $\boldsymbol{\lambda}$ of Lagrange multipliers determines how much of each constraint is applied. Each value provides a measure proportional to the magnitude of the reaction force applied due to the corresponding constraint. To counteract the accumulation of numerical drift over time, corrective stiffness and damping terms are appended to (Eq. 3.24) as proportional and derivative error correction factors:

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

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

$$\mathbf{G}_i(\mathbf{p}) \;=\; \|\mathbf{p_B} - \mathbf{p_A}\|_\mathbf{2}^\mathbf{2} - L_{rod,q}^2 \tag{3.26}$$

$$\;=\; \frac{\mathbf{1}}{\mathbf{2}}\mathbf{p}^\mathbf{T}\left(\left[\mathbf{R_q^T R_q}\right] \otimes \mathbf{I_3}\right)^\mathbf{2} \mathbf{p} - L_{rod,q}^2 \tag{3.27}$$

$$\dot{\mathbf{G}}_i(\mathbf{p}, \dot{\mathbf{p}}) \;=\; \mathbf{p}^\mathbf{T}\left(\left[\mathbf{R_q^T R_q}\right] \otimes \mathbf{I_3}\right)^\mathbf{2} \dot{\mathbf{p}} \tag{3.28}$$

$$\mathbf{J}_i(\mathbf{p}) \;=\; \mathbf{p}^\mathbf{T}\left(\left[\mathbf{R_q^T R_q}\right] \otimes \mathbf{I_3}\right)^\mathbf{2} \tag{3.29}$$

$$\dot{\mathbf{J}}_i(\mathbf{p}, \dot{\mathbf{p}}) \;=\; \dot{\mathbf{p}}^\mathbf{T}\left(\left[\mathbf{R_q^T R_q}\right] \otimes \mathbf{I_3}\right)^\mathbf{2} \tag{3.30}$$

Given these implicit constraint functions which are obtained for each rod, we combine these results with the formulas in Section 3.4.1 to guarantee that nodal accelerations are accurately simulated, with no pair-relative acceleration components lying along the longitudinal axis of the rigid rods.

3.4.3 Ground Contact and External Forces

Ground contact forces are calculated in a similar manner to the approach introduced in Section 3.3.3, with a few additions to help improve numerical stability during simulation: (1) viscous damping is added in the upward direction to prevent excessive clipping (i.e., the robot geometry entering into the ground due to high downward velocities), and (2) horizontal viscous forces were added to improve realistic simulations of rolling mobility.

The vertical ground contact forces are applied to each node in the Z-direction and uses the smooth approximation to the linear rectifier (Eq. 3.18) as well as its derivative, which acts a smooth approximation (one of many possible choices) to the unit step function:

$$softStep(\alpha,\beta) \doteq \frac{\partial}{\partial\alpha} softPlus(\alpha,\beta) = \frac{1}{2}\left(\frac{\alpha}{\sqrt{\alpha^2+\beta^2}}+1\right) \tag{3.31}$$

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

$$\mathbf{F}_{z,i} = softPlus(Z_{floor}-Z_i,\beta) - c\cdot softStep(Z_{floor}-Z_i,\beta)\dot{Z}_i \tag{3.32}$$

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

(todo: add horizontal viscous forces)

3.4.4 Complete Tensegrity Point Mass Accelerations

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

$$\begin{aligned}
\ddot{\mathbf{p}} &= \mathbf{W}\cdot\mathbf{F}_{legal} && (3.33)\\
&= \mathbf{W}\cdot\left(\mathbf{F}_{total}+\hat{\mathbf{F}}\right) && (3.34)\\
&= \mathbf{W}\cdot\left(\mathbf{F}_{total}+\mathbf{J}^{\mathbf{T}}\boldsymbol{\lambda}\right) && (3.35)\\
&= \mathbf{W}\cdot\left(\mathbf{F}_{total}+\mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathbf{T}})^{-\mathbf{1}}[-\dot{\mathbf{J}}\dot{\mathbf{p}}-\mathbf{J}\mathbf{W}\mathbf{F}_{total}-k_s\mathbf{G}-k_d\dot{\mathbf{G}}]\right) && (3.36)\\
&= \mathbf{W}\cdot\Bigg(-\mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathbf{T}})^{-\mathbf{1}}(\dot{\mathbf{J}}\dot{\mathbf{p}}-k_s\mathbf{G}-k_d\dot{\mathbf{G}})+ \\
&\qquad (\mathbf{I}-\mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathbf{T}})^{-\mathbf{1}}\mathbf{J}\mathbf{W})(\sum_j\boldsymbol{\gamma}_j+\mathbf{F}_{external})\Bigg) && (3.37)
\end{aligned}$$

Eq. 3.36 is a direct result of substituting in Eq. 3.25, and the last line (Eq. 3.37) is obtained by combining the results of Sections 3.4.1 to 3.4.3. Note, $\mathbf{F}_{external}$ is the vector of the total forces applied to the tensegrity robot which are external to the system (e.g., ground contact reaction forces) and are calculated using damped linear-elastic collisions previously presented above in Section 3.4.3.

The structured dynamics that result for the overall system are highly-coupled, nonlinear equations of motion. From this, it is readily apparent that it can quickly become difficult to intuitively reason about how local changes in an individual cable's tension, t_i, may affect the overall geometry of the global system. Nevertheless, recent work such as [2] utilized hand-engineered two-cable actuation policies which demonstrated good results for fast uphill climbing. This becomes increasingly difficult, however, as more cables are considered, with full actuation of all 24 cables presenting a daunting combinatorial problem. For this reason, later chapters present iterative locally optimal control techniques and model predictive control approaches which can take advantage of the procedurally-obtained mathematical models of the system to find optimal actuation policies.

3.5 Linearization and Discretization

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

$$\begin{bmatrix} \mathbf{p}_{k+1} \\ \dot{\mathbf{p}}_{k+1} \\ \boldsymbol{\ell}_{cable,k+1} \\ \boldsymbol{\ell}_{rod,k+1} \end{bmatrix} \approx \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cable,k} \\ \boldsymbol{\ell}_{rod,k} \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\mathbf{p}}_{\mathbf{k}} \\ \ddot{\mathbf{p}}_{\mathbf{0}} + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}} \tilde{\mathbf{x}}_{\mathbf{k}} \\ \dot{\boldsymbol{\ell}}_{cable,k+1} \\ \dot{\boldsymbol{\ell}}_{rod,k+1} \end{bmatrix} \tag{3.38}$$

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

$$\begin{bmatrix} \mathbf{p}_{k+1} \\ \dot{\mathbf{p}}_{k+1} \\ \boldsymbol{\ell}_{cable,k+1} \\ \boldsymbol{\ell}_{rod,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{k} \\ \dot{\mathbf{p}}_{k} \\ \boldsymbol{\ell}_{cable,k} \\ \boldsymbol{\ell}_{rod,k} \end{bmatrix} + \frac{\Delta t}{2} \left(\begin{bmatrix} \dot{\mathbf{p}}_{k} + \dot{\mathbf{p}}_{k+1} \\ 2\ddot{\mathbf{p}}_{0} + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}} \tilde{\mathbf{x}}_{k} + \frac{\partial \ddot{\mathbf{p}}}{\partial \mathbf{x}} \tilde{\mathbf{x}}_{k+1} \\ \dot{\boldsymbol{\ell}}_{cable,k} + \dot{\boldsymbol{\ell}}_{cable,k+1} \\ \dot{\boldsymbol{\ell}}_{rod,k} + \dot{\boldsymbol{\ell}}_{rod,k+1} \end{bmatrix} \right) \tag{3.39}$$

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

3.5.1 ODE Solvers Discussion

TO-DO: Discuss the various discretization approaches and discretization approaches here. Some simple experiments maybe?

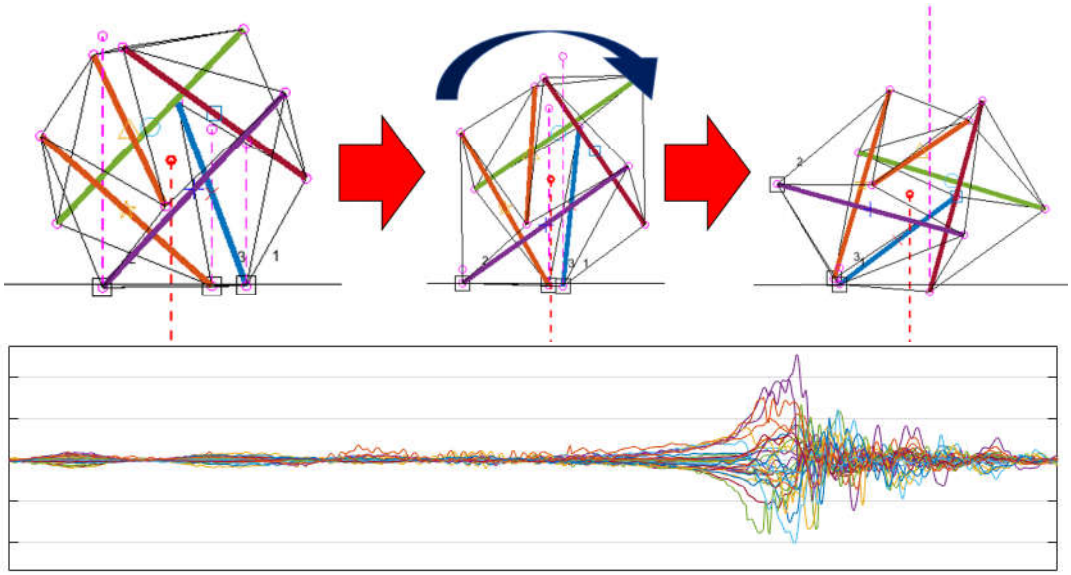

Figure 3.5: Top: MATLAB simulation of six-bar spherical tensegrity rolling using 24-cable actuation, at rest (left), initial tipping instability (middle), and impact on the next base (right). Bottom: Corresponding linear [m/s] and angular [rad/s] velocities of all rods (30-dimensional velocity).

3.6 Custom Tensegrity MATLAB Simulation Framework

A custom-designed simulation environment was developed in MATLAB for simulation of the tensegrity robot and easy interfacing with existing optimization toolboxes (see example graphic in Fig. 3.5). Using the dynamics described in the previous section, a framework for initialization and forward-simulation of the robot was created. With ease-of-use and usability in mind, the intent of the simulation environment is to be a general-purpose tool for simulation and control of any Class-1 tensegrity system. Due to the structured nature of the tensegrity dynamics, as discussed in section previous sections, construction of idealized models for Class-1 topologies is relatively straightforward. However, during development, some practical issues were made evident and additional features in the simulation were incrementally added as needs progressed. What began as a simple simulation of a free-floating tensegrity in the absence of gravity has evolved into a relatively comprehensive simulation environment for tensegrities in contact-rich dynamics.

3.6.1 Object-oriented Approach

3.6.2 Automatic Generation of Dynamics Equations

3.6.3 State Initialization using Kinetic Energy Damping

The first issue that became immediately apparent early on in the development of the simulation environment was the need for proper initialization of the robot state. In the earliest naive approach, the 6-bar tensegrity was initialized using a regular icosahedron assumption (i.e., rods and nodal positions were defined assuming the outer surface of the spherical tensegrity had equilateral triangles for each of its 20 faces).

In practice, this initialization did not work for two reasons - early implementations of the code i) did not take into a desired pretension into account (so that the robot could immediately collapse) and ii) didn't account for drastic transient effects that occurred once the robot was initialized and released from the initial state. Specifically, if cable restlengths were immediately set by a desired pretension rather than incrementally updated, excessively large accelerations caused the simulation to be unstable, regardless of timestep size.

To address this, simulations begin by first finding nominal states using the regular-icosahedron assumption described above. Nodal separation distances are then calculated, and a desired pretension determines cable restlengths setpoints relative to these initial distances. Cables are then incrementally controlled to these restlengths rather than immediately set, to mitigate any large forces and accelerations. To speed up this initialization, energy of the system is observed at each timestep and *kinetic energy damping* is employed (i.e., nodal velocities are set to zero each time kinetic energy of the system peaks). This artificial damping is not a real effect of the dynamics of the system but offers a drastic reduction in the number of iterations necessary to reach equilibrium at initialization of the simulation environment.