Feasibility analysis of a jumping tensegrity robot using form finding & kinematic models

Master Research Project: Final Report
Tim de Vries – S2974177

April 20, 2023

Rijksuniversiteit Groningen

Abstract

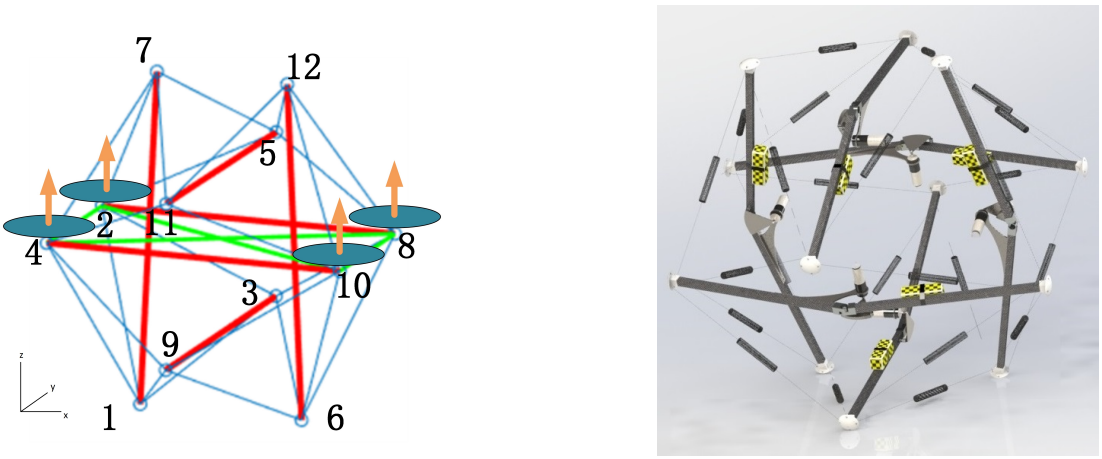
In this research it is theoretically investigated how well an icosahedron tensegrity structure with 6 bars and 24 cables is capable of jumping. The bars of the conventional icosahedron tensegrity are replaced by legs with a torsion spring joint at the halfway point, and the legs are actuated by a gear-reduced motor. A form finding model of the jumping tensegrity structure has been constructed in a Matlab/Simulink environment, which is the main result of this research. This model is used to examine a static state of the robot, both at rest and during application of torque at the joint. After that two other models are proposed: a kinematic model which directly calculates acceleration from reaction forces, and another more complicated Euler-Lagrange description which utilizes kinetic and potential energies. By varying physical parameters of the robot an estimate can be made of the performance of the robot. It is determined that theoretically the robot should be able to jump, and the set of relevant parameters can be tuned to increase performance and stability. A Simscape simulation which also shows behaviour in real time, including rolling and repositioning of the robot. The Simscape model is used as a validation tool for the found parameters. The Simulink model is a good use for optimizing parameter values, and the Simscape model gives a better prediction of real life behaviour of the robot. More work is needed on real-time kinematic simulations like the Euler-Lagrange model to do more accurate analysis regarding the performance of the robot.

Contents

Introduction

The term tensegrity was coined by architect Buckminster Fuller in 1960, and it is a combination of the words 'tension' and 'structural integrity'. He came up with the term after a former student of his named Kenneth Snelson showed him a structure of pipes and wires which seemed to float in the air. At first it started out as art installations and purely experimental setups, not as load bearing structures. In the space race of the 1960's there was need for lightweight structures which could be used for space construction. The tensegrity structures came up as an advantageous applications due to their flexibility and lightness [1]. Since then a wide variety of applications have been seen in for instance civil engineering [2], and in biological research ranging from the cellular scale to skeletal systems [3].

There is a wide variety of jumping robots that have been built and researched, like a soft steerable jumping robot consisting of a flexible cage as a body [4]. Burdick and Fiorini [5] designed a hopping robot which should be equipped for exploration of celestial bodies. Haldane et al. researched a robot which emulated the movement of the galago (a small primate from Sub-Saharan Africa), such that the power input was amplified to increase jump height [6]. A famous example of a tensegrity structure used in locomotion of a robot is the NASA SUPERball, a crawling tensgrity robot [7]. Movement of this structure is generated by actuation of the tension members, which can shift the center of gravity of the structure with respect to its supporting bars, which in turn causes the robot to roll. A different design by Hirai et al. uses McKibben actuators which shrink as air pressure is applied also produces a crawling motion [8].



(a) (b)

Figure 1.1: Schematic view (a) and a rendering from a CAD model (b) of the concept.

The tensegrity structure that is studied in this report is that of an icosahedron with 6 bars and 24 cables. This structure is also investigated by Zha et al. [9] as a protective exoskeleton for drones. When the structure experiences a collision with a surface, the load of the impact is distributed among its members. The concept of a jumping 6 bar tensegrity robot is based on the work of Mintchev et al. [10], who uses the same tensegrity configuration for a soft robot for random exploration.

The big difference is in the way the robot is propelled. In the paper by Mintchev the locomotion strategy is to use periodically activated propellers to make the robot jump. Since we are interested in random exploration on celestial bodies, a propeller has one big drawback

because of its dependency on the properties of the atmosphere to work, if there is an atmosphere at all. Therefore a good alternative would be to make the structure able to use its legs to jump.

Concept

The idea is to replace the straight rigid bars with bars that have a joint in the middle, and this member we call a *leg*. The joint contains a torsion spring in the bending axis, which is wound up as the legs of the robot compress by use of a motor and a gearbox to minimize the required torque on the motor. The potential energy of the torsion springs increases, and when the legs are released this will be transferred into kinetic energy. Cable members in the structure are all equivalent and do not differ from a regular elastic string member in a tensegrity structure.

The robot can jump in two different ways, with one of the modes being the 2-legged jumping mode when 2 parallel legs are touching the ground, and the other mode is activated when 3 non-parallel legs touch the ground and the robot needs to readjust. In this case only one of the legs is contracted and used to jump. An example would be that nodes 1, 6 and 9 would be the supporting nodes (see figure 1.1a), which means that bars 1–7 and 6–12 are used in this particular case. If for instance nodes 1, 4 and 9 are touching the ground, there is no set of parallel bars which have contact to the ground. This means the structure should reposition to the first configuration.

Figure 1.2: CAD model of two parallel bars. The cam is the triangular shaped object which is actuated by the circular white motor, which also contains a small gearbox. The torsion spring is embedded in the joint.

In preceding work a CAD model of the structure has been visualized (figure 1.1b), and in figure 1.2 two parallel legs are displayed. The joints of the legs consist of a cam, a rotational motor, a gearbox and a torsion spring. The cam is shaped in a way that the applied torque remains constant throughout the actuation of a leg. The design of the actuator and the legs in general is beyond the scope of this report. Later on a few assumptions are made about the legs, which are simplifications to the real world example.

Structure of this report

There are multiple stages of research in this report. The first and main stage is using a form finding algorithm in order to find appropriate parameters for a given motor torque τ_m. Parameters will be chosen such that jump height can be maximized, all the while the prestress stability of the tensegrity structure is not compromised. Factors that can be adjusted are the leg lengths, stiffness coefficients of both the cables and the torsion springs, and the rest length of the cables. A Simscape Multibody simulation has been constructed in order to get an idea of the behaviour of a jumping tensegrity structure. This model is used as a comparison to the

estimated performance from several Simulink simulations.

There are two stages following up the results found in the form finding model. In a second stage an attempt is made to generate a kinematic model of the jumping tensegrity structure. In practice the form finding model is adapted to a mechanical simulation in Simulink by double integration of calculated reaction forces. Then as a third stage of the project a more accurate model is proposed: an Euler-Lagrange model. The model is constructed according to the Euler-Lagrange equations and is described in detail in a paper by Hsu et al. [11]. There are certain facets and generalizations for which our model needs an adaption, but in general it is a good example to start from.

The reader should be aware that the main results are found in chapters 2 and 3, which discuss stage 1 and 2. Before continuing to the 3rd stage, the multibody simulation in Simscape is discussed in chapter 4. The kinematic model does not approach the Simscape simulation very well, but can still give us a different insight in the influence of varying parameters on the effectiveness of the structure. The third stage is primarily a proposal of a kinematic model based on the Euler-Lagrange equations. This part requires more work and its Matlab code is not in a working state at the moment this report is finished. Yet the mathematical description and the adaptions that are made w.r.t. the work by Hsu et al. [11] are laid out in chapter 5 of this report.

Chapters 2 and 3 have a conventional structure of explanation of theory, presentation of results and discussion of these results. Chapter 5 is structured differently, only the theory has been described. There are no results to be present because the model was not complete at the moment this report was written. Therefore a section in chapter 6 is dedicated to giving recommendations for future continuation of this topic.

Assumptions & Simplifications

In the Simulink models discussed in this report there some important assumptions and simplifications:

- Firstly the mass of the structure only stems from the mass of the bars, the batteries and the motors. The cable members of the tensegrity have no mass. Furthermore, the calculations consider a bar to have their mass situated at the center of mass, and not along the length of the entire length of the bar. There are also no localized masses for placed batteries or motors.
- The bar sections are rigid.
- The linear springs only act as tension members, and do not exert an outward force when compressed.
- The linear springs are all attached to a bar end node at exactly the point in space where the node is situated, whereas in reality the connection points would have some space between them. This offset prevents the bar from rotating around its axis. In the simulation the orientation of the leg is defined by its parallel counterpart, so that the joints of parallel legs lie in the same plane as the parallel legs themselves.
- The forces on the end nodes stemming from applied torque at the joint can only point in two directions. The direction toward the other end node of the bar, or exactly the opposite direction away from the other end node.
- All nodes have the condition that they cannot move to a z-coordinate value below zero, so in essence there is a rigid plane on which the robot can take off and land again.

- The joints of the bars are not treated as nodes of the tensegrity. A choice is made to scope the project down to the simplification that the bars are essentially soft elastic members. It is possible that some time during the motion a joint is located 'beneath ground' as it is not considered a node but it is plotted in the animation. For a more accurate simulation it should be defined as a node, but the constraints on its motion make the two joint-connected bar members a special case which does not exactly fit in the general description of tensegrity dynamics discussed in this report.

General Theory

A tensegrity can be defined as a graph G with vertices $1, 2, ..., n$ in d-dimensional space E^d. Assigned to these vertices are nodes $p = (p_1, ..., p_n)$ in 2- or 3-dimensional Euclidean space. Edges are defined by the unordered pair $\{i, j\}$, where $i \neq j$. In the most simple description, every edge is either a *cable*, a *bar* or a *strut*. Nodes connected by a cable can come closer together, but can't get further apart than the cable length. Struts have the opposite effect, where the nodes can't come closer together than the minimum length of the strut. Nodes connected by a bar stay the same distance apart at all times. The configuration constituted by a graph is then a tensegrity $G(p)$ [12].

Any continuous motion of nodes is called a *flex*, where a *trivial flex* is a subset of flexes. A trivial flex occurs under translation or rotation of the tensegrity, i.e. the configuration is moved as a whole without node distances changing and we say the configuration is *rigid*. A non-trivial flex occurs when the node distances are changing, and thus the configuration is *flexible*. When the tensegrity structure is the 6 bars and 24 cable isocahedron, it is rigid. However, in this research neither bars nor cables are rigid. We allow the nodes to get closer together by bending of the joint, where the outward force on end nodes get larger due to joint spring compression. The cables act as linear springs, so the force on the end nodes from a cable is a function of the node distance with respect to the cable rest length l_r. For $d > l_r$ the force is given by Hooke's Law, $F = k(d - l_r)$, and when $d \leq l_r$ then $F = 0$. The joints in the legs have torsion springs in them, so the torque is given by $\tau_t = -\kappa\alpha$.

Prestress & Energy

The general prestressability condition is that all tendons are in tension all the while the structure is in a stable configuration without any external forces or torques acting on the structure. The configuration in which the structure is in a state of prestress need not be unique [13].

Because of the two types of springs, we can define the stress energy contained in the tensegrity as the sum of energy of individual members. The stresses are defined by $\omega = \{..., \omega_{ij}, ...\}$. If nodes are not connected the entry of the stress vector is zero. The stress energy of graph a G is defined as:

$$E_\omega(p) = \sum_{i<j} \omega_{ij} ||p_i - p_j||^2 \tag{1.1}$$

For tension members, the stress energy of a single member is defined by the energy from Hooke's Law: $E_c = \frac{1}{2}k(d - l_r)^2$. The energy gained from the winding of the torsion spring in the legs is in radial form: $E_j = \frac{1}{2}\kappa(\alpha_r - \alpha)^2$ with the angle in radians.

Lagrangian description

A 2020 paper from Hsu et al. [11] presents a Lagrangian description of a general tensegrity structure. This paper has been an important guide for constructing every kind of model discussed in this report. A seemingly simple equation denoting the kinetic and potential energy in the system, $\mathcal{L} = T - V$, is utilized to derive the equations of motion of the entire system. As the reader proceeds through the report, more and more of the Euler-Lagrange method will come

to light and it becomes increasingly more important. At first only the energies are calculated in order to get an idea of the performance of the robot in chapter 2. In chapter 3 there is not much added with regard to the Euler-Lagrange description, but the form finding model is converted to an iterative kinematic model. Normal force and friction with the ground is added to include landing and rolling of the tensegrity structure, which also affects the performance of the robot. In chapter 5 the model gets a complete overhaul and the E-L description as laid out in the paper by Hsu et al. [11] is adapted and implemented in an attempt to get a more accurate simulation.

Form Finding Model

The goal of this chapter is to check the feasibility of the six bar 24 cable jumping tensegrity robot and to find an optimal set of parameters of various parts of the robot so it might be realised as a 1:1 size prototype in the future. In Simulink a form finding model is created to see how a static structure will react to the application of a set of parameter settings. The model has its initial conditions and constants set up in a script within Matlab [14]. The Simulink model discussed in this report is a continuation of an earlier assembled model of form finding. In this previous model the bars were modelled as linear springs with high stiffness, and the cables were also linear springs. The 12 end nodes were given initial conditions and the structure would converge towards a stable prestressed configuration. The new model is an extension of this model, with the bars being completely remodelled. The joints are implemented and the reactive force from the torsion spring is also modelled, together with an addition of mass to the bars and gravity affecting the whole structure. As a consequence of this adaption, the existing model had to be reviewed from top to bottom.

2.1 Theory & Methods

The form finding model is initialized with user input constants as follows: the legs have user defined constants κ, l_b and α_j which are the torsion spring stiffness, the total length of the whole leg in its stretched position and the rest angle of the joint respectively. The cables have similar inputs k and l_c which are the linear spring stiffness and rest length. The user can also define the total mass of a leg m_b and the torque applied by a motor τ_m.

2.1.1 Gradient Descent

Gradient descent is a popular algorithm in optimization. It is a way to minimize an objective function $J(\theta)$ with model parameters $\theta \in \mathbb{R}^d$ by means of updating the parameters in the opposite direction of the gradient $J_\theta J(\theta)$. So at every time step we calculate the gradient of the position of an individual node. We build the model around this principle and find the different forces acting on individual nodes. Forces that arise from the prestressed state (forces exerted by the linear- and torsion springs) and other forces like gravity and motor torque are added at time t_i. The sum of all forces on a single node are integrated to find the position of node x_i at new time step t_{i+1}.

2.1.2 Cable modelling

The simplest part of the model is the cable part. Every node is connected to four other nodes by a tendon, and these tendons are modelled as linear springs which exert a force when $|x_i - x_j| > l_c$. However when the string is slack (i.e. $|x_i - x_j| \leq l_c$), there is no force exerted by this string. The force on an end node is calculated by:

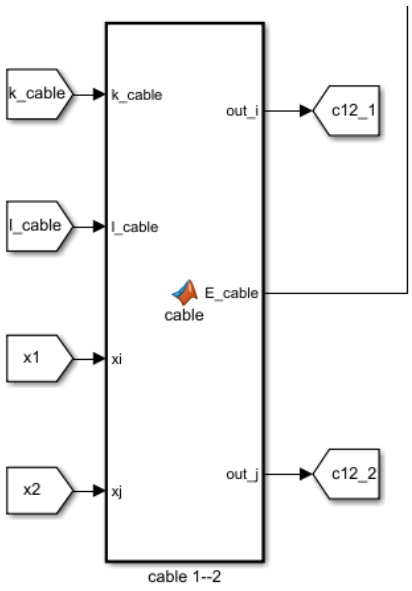


Figure 2.1: Function block of the cable between nodes 1 and 2 from the Simulink model.

$$\begin{aligned}\mathbf{F_c} &= k(l_c - |x_i - x_j|)\frac{\mathbf{x_j} - \mathbf{x_i}}{|x_i - x_j|} \\ &= k(\frac{l_c}{|x_i - x_j|} - 1)(\mathbf{x_j} - \mathbf{x_i})\end{aligned} \tag{2.1}$$

The force on the other end node is the exact opposite to this force. The individually modelled cable has these two forces as outputs, and a third output is the elastic energy stored in the linear spring due to contraction:

$$E_c = \frac{1}{2}k(d - l_c)^2 \tag{2.2}$$

This is done 24 times, and all energies are added together so we can calculate the total elastic energy change in the system. A block as seen in Simulink can be found in figure 2.1.

The distance between cable connected nodes has been geometrically determined. From figure 2.2 we can see that

$$\begin{aligned}c^2 &= (L_b/4)^2 + (L_b/4)^2 = L_b^2/8 \\ L_c^2 &= c^2 + (L_b/2)^2 = L_b^2/8 + 2L_b^2/8 = \frac{3}{8}L_b^2 \\ L_c &= \sqrt{\frac{3}{8}}L_b = \frac{\sqrt{6}}{4}L_b\end{aligned} \tag{2.3}$$

Since we want the structure to be in prestress, the cable rest length of the tendons in the structure should be shortened. For this we define a parameter δ which is the cable rest length deviation percentage. The cable rest length constant is then defined as in equation 2.4.

$$l_c = \frac{\sqrt{6}}{4}l_l(1 - \delta) \tag{2.4}$$

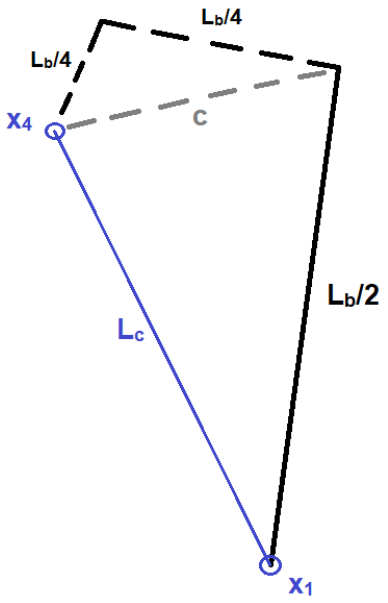


Figure 2.2: Schematic drawing of the cable link between node x_1 and x_4 from figure 1.1a.

2.1.3 Legs modelling

We now know the forces caused by the linear springs, and next we want to calculate how the torsion spring in the individual legs will exert their force on the end nodes. The calculation of the force is divided into two parts, where the angle of the joint is calculated as first thing and this output is also used to calculate the forces on both end nodes in the second block (see figure 2.3). There is also one instance of an `jumping angle` function which is an in between step in calculating the maximum jump height of the robot. The information available to us for calculation of the joint angle is the location of the nodes $\mathbf{x_i}, \mathbf{x_j}$, the location of one of the nodes of a parallel leg $\mathbf{x_j}$, and the leg length in stretched position l_l. The outputs of the `angle` function block include the angle of the leg joint and the position of the joint.

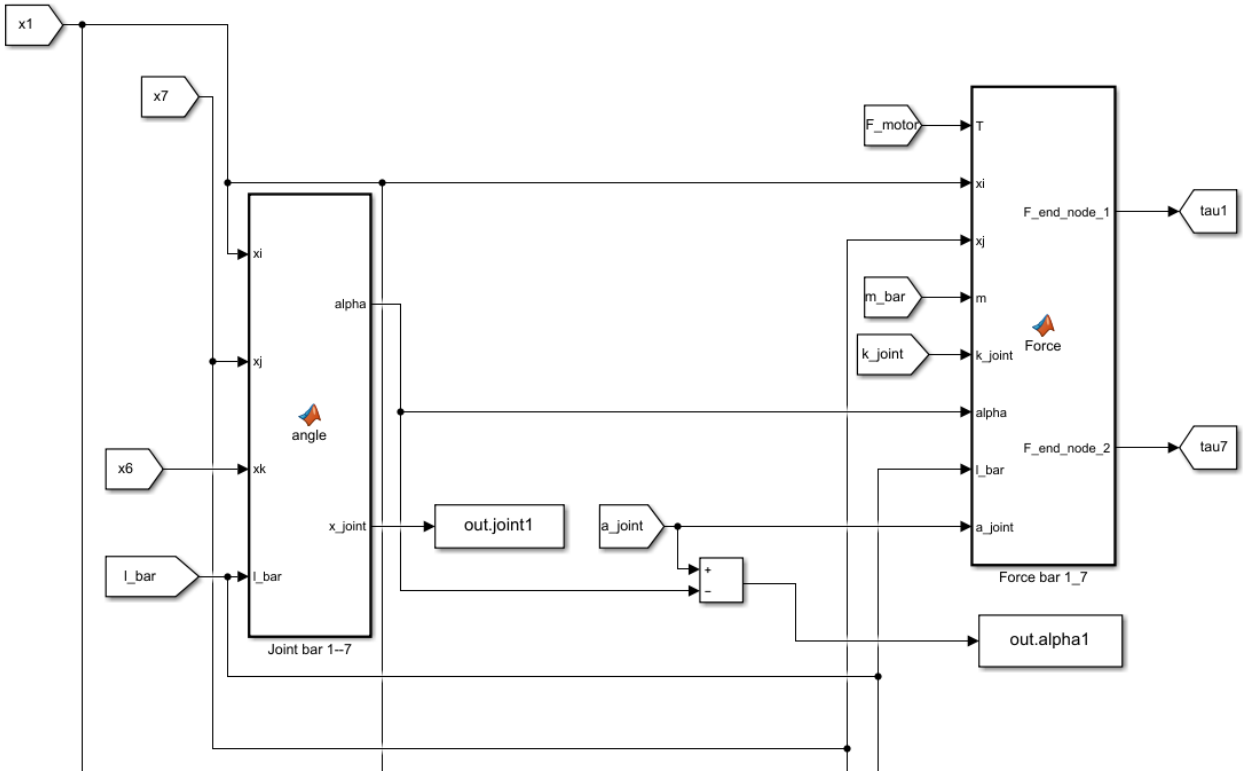


Figure 2.3: Block diagram from Simulink showing the function that are used to calculate the reaction force from the joint of leg 1.

The law of the cosines is used to find the angle of the joint, and when we use it in its standard form with information we get from the node position we get:

$$d^2 = l_b^2 + l_b^2 - 2l_b^2 cos(\alpha) \tag{2.5}$$

We rewrite this to find the angle as follows:

$$\begin{aligned} d^2 &= 2l_b^2 - 2l_b^2 cos(\alpha) \\ \frac{d^2}{2l_b} &= (1 - cos(\alpha)) \\ cos(\alpha) = 1 - \frac{d^2}{2l_b^2} &= \frac{2l_b^2}{2l_b^2} - \frac{d^2}{2l_b^2} = \frac{2l_b^2 - d^2}{2l_b^2} \\ \alpha &= cos^{-1}(\frac{2l_b^2 - d^2}{2l_b^2}) \\ &= cos^{-1}(\frac{l_l^2 - 2d^2}{l_l^2}) \end{aligned} \tag{2.6}$$

The output α is given in radians. The other output from the `angle` function are the coordinates of the joint which is used for plotting and apart from that does not have any influence on the behaviour of the structure.

The force calculation function calculates and adds up the force from torsion, from gravity and from the motor torque (if applicable) to complete the total force addition on the end nodes.

$$\mathbf{F_{node}} = (F_{tor} - F_{motor})\frac{\mathbf{x_i} - \mathbf{x_j}}{|\mathbf{x_i} - \mathbf{x_j}|} + \mathbf{F_g} \tag{2.7}$$

The one big assumption in equation 2.7 is that the reaction force from the torsion spring only acts along the line that can be drawn between the two end nodes of the leg, which is seen in the

equation by the multiplication of the normalized vector direction of the leg. Because of this, the force can be simplified quite a lot and we can work with a lever equation.

Torsion spring reaction force
The force from the torsion spring is dependent on the angle at time t and the deviation from the rest angle of the joint. This expression is then halved because the force is distributed among two end nodes: $F_{spring} = \frac{1}{2}\kappa(\alpha_r - \alpha)$. This should be equated to the force experienced at one of the end nodes, which means that the length of the bar from the joint needs to be accounted for as well. The torsion force on the end node is multiplied by l_l. The force can only act in the direction of $\mathbf{x_i} - \mathbf{x_j}$ and is thus dependent on the bending angle α.

$$F_{tor}\frac{l_l}{2}cos(\frac{\alpha}{2}) = \frac{1}{2}\kappa(\alpha_r - \alpha)$$

Rewriting this we obtain an expression for F_{tor}:

$$F_{tor} = \frac{\kappa(\alpha_r - \alpha)}{l_l cos(\frac{\alpha}{2})} \tag{2.8}$$

Note that this is a scalar quantity and the direction is determined in equation 2.7. There is also an energy associated with winding up of the torsion spring (eq. 2.9).

$$E_{tor} = \frac{1}{2}\kappa(\alpha_r - \alpha)^2 \tag{2.9}$$

Motor torque
Similar to the lever effect in the previous part, the motor torque should exactly counteract the torsion spring force. From the applied torque the force on the end node is given by;

$$F_{motor} = \frac{\tau_m}{l_l cos(\frac{\alpha}{2})} \tag{2.10}$$

As seen in equation 2.7 this is subtracted from the torsion spring force.

2.1.4 Gravity
Gravitational acceleration $\mathbf{g} = [0\ 0\ -9.81]^T$ is multiplied by the mass of a bar to find the force exerted by the bar. However in the one cases the force on the bar is slightly different because of the orientation of the structure. In the case of the jumping legs 1–7 and 6–12 jumping and leg 3–9 being the support it has to be taken into account that the joint rotation is in the same plane as the gravitational force direction.

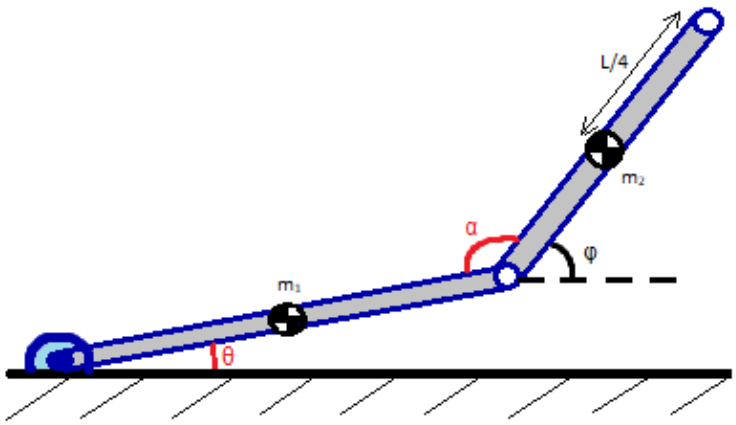


Figure 2.4: Free body diagram of leg 3–9.

The gravitational force on the end node is a sum of the force stemming from the center of mass of the left bar and that from the right bar. From a lever equation we find the force on the joint node to be:

$$\begin{aligned} \frac{l_l}{2}\mathbf{F_{39,1}} &= \frac{ml_l}{4}cos(\theta)\mathbf{g} \\ \mathbf{F_{39,1}} &= \frac{m}{2}cos(\theta)\mathbf{g} \end{aligned} \tag{2.11}$$

The right part of the leg in figure 2.4 has a contribution of half of the total gravitational force, $\mathbf{F_{39,2}} = \frac{m}{2}\mathbf{g}$. The end node on the right part is also influenced by the left part, as the angle under which the bar 'hangs' influences the pull on the end node. If the rightmost bar is horizontal, there is no instantaneous transfer of forces and there is no contribution of the left part. As such we get the following relation for the end node:

$$\begin{aligned} \mathbf{F_g} &= sin(\phi)\mathbf{F_{39,1}} + \mathbf{F_{39,2}} \\ \mathbf{F_g} &= \frac{m}{2}(cos(\theta)sin(\phi) + 1)\mathbf{g} \end{aligned} \tag{2.12}$$

2.1.5 Maximum jump height

In the Simulink file there is one more function block, which calculates the jumping angle of the robot. This is the angle of one of the actuated legs (1–7) with respect to the ground. Together with the total energy stored the maximum theoretical jumping height can be derived by doing multiple simulations of form finding. This model is not fit for simulating real time jumping of the robot, but we can approximate the maximum performance by comparing the energy stored in the structure when no force is applied versus when there is a torque τ_m applied to legs 1–7 and 6–12.

We need the information of 4 nodes to do the calculation: all three nodes that touch the ground are needed (x_1, x_6 and x_3) and also one of the nodes of the other end of the actuated bars, for which we choose x_7. In figure 2.5 a schematic drawing of a part of the tensegrity structure can be seen. The angle between the two normal vectors in the figure is the same as the angle of the actuated legs with the ground. We can find the normal vectors by performing the following cross products:

$$\begin{aligned} \mathbf{n_1} &= (\mathbf{x_1} - \mathbf{x_7}) \times (\mathbf{x_1} - \mathbf{x_6}) \\ \mathbf{n_2} &= (\mathbf{x_1} - \mathbf{x_3}) \times (\mathbf{x_1} - \mathbf{x_6}) \end{aligned} \tag{2.13}$$

Then we can find the angle between these two rewriting the definition of the dot product:

$$\theta = cos^{-1}(\frac{\mathbf{n_2} \cdot \mathbf{n_1}}{|n_2||n_1|}) \tag{2.14}$$

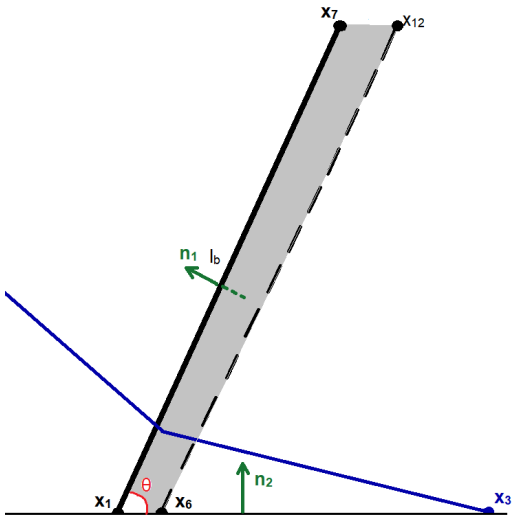


Figure 2.5: Schematic side view of legs 1–7, 3–9 and 6–12. Two virtual planes can be seen in this picture, one being the ground level and the other being the greyed out area between the actuated legs. Their respective normal vectors can be seen in green.

Other information needed for the calculation of the maximum jump height is the energy difference between the relaxed and the compressed state. In the relaxed state, all springs are in a state of prestress and there is already some stress energy present in the system. As the two motors are activated, the torsion springs of these legs are wound up, and there are also changes in the length of linear spring and in the angle of other joints which are not being actuated. Following the earlier discussed Euler-Lagrange principle, these elastic energies can be added up as total potential energy U in the system.

$$U = E_{comp} - E_{relax} \tag{2.15}$$

There are two runs of the simulation needed to obtain this information, first without applied torque. The form finding algorithm will find a stable configuration and at final time t_f the stored energy will be measured. This is done again but now with 50 Nm torque applied at the joints of bars 1–7 and 6–12.

$$\begin{aligned} E_j = &\frac{1}{2}\sum_{i=1}^{6} \kappa(\alpha_r - \alpha_{comp})^2 \\ &-\frac{1}{2}\sum_{i=1}^{6} \kappa(\alpha_r - \alpha_{relax})^2 \end{aligned} \tag{2.16}$$

Similar as the energy from springs of equation 2.2:

$$\begin{aligned} E_c = &\frac{1}{2}\sum_{i<j} k(d_{ij,relax} - l_r)^2 \\ &-\frac{1}{2}\sum_{i<j} k(d_{ij,comp} - l_r)^2 \end{aligned} \tag{2.17}$$

Here d_{ij} is the node distance between node $\mathbf{x_i}$ and $\mathbf{x_j}$ if the nodes are cable connected, and $d_{ij} = 0$ when not connected by a cable.

By a simple calculation of conservation of energy we can find the maximum possible jumping height. Starting with the potential energy of the springs and height energy. With

$$U_{jump} = U \cdot sin(\theta) \tag{2.18}$$

the robot jumps at angle, so some of the energy is also converted into a lateral translation instead of height. The height is simply calculated as follows:

$$\begin{aligned} U_{jump} &= mgh \\ h = \frac{U_{jump}}{mg} &= \frac{U \cdot sin(\theta)}{mg} \end{aligned} \tag{2.19}$$

The calculation of this jump height is an approximation of the maximum possible performance, and does not accurately represent a real situation. For instance it assumes no losses of energy during the extension of the legs, there is no rotation of the robot created by jumping. This jump height calculator is a tool for finding stable configurations and trying out different stiffness values, cable lengths, leg lengths and more.

2.1.6 Initialization of Simulink model

The modelling approach has been described in full, the only thing left to do is initialize the model. There are a few parameters which are fixed from the beginning, which include the the rest angle of the joint ($\alpha_r = 160°$), the density of the bar material is chosen to be the density of carbon fiber (1900 kg/m^3) and the radius of the bars is set at 0.5 cm. User defined parameters include but are not excluded to k, κ, l_l, m_{motor}, $m_{battery}$, τ_m and l_c. The set of standard initial parameters can be seen in table 2.1. For all following analysis the joint rest angle, the bar radius, applied torque and the weight of attached masses remain constant.

	Parameter	Value	Unit
$\mathbf{l_l}$	Leg length	30.0	[cm]
$\mathbf{r_b}$	Bar radius	0.5	[cm]
$\mathbf{m}$	Attached mass	0.618	[kg]
$\mathbf{m_{tot}}$	Total mass	3.98	[kg]
$\boldsymbol{\kappa}$	Joint stiffness	50	[N/rad]
$\mathbf{k}$	Cable stiffness	2500	[N/m]
$\boldsymbol{\alpha_r}$	Joint rest angle	160°	[deg]
$\boldsymbol{\delta}$	Deviation	20	[%]
$\boldsymbol{\tau_m}$	Applied torque	50	[N/rad]

Table 2.1: Standard parameter values used in analysis. When the influence of a parameter is investigated, the value will be varied and all other parameters remain at the value stated in this table.

2.2 Results

The tensegrity structure was plotted in 3D to see how the structure finds its optimal configuration under different circumstances. If the structure fails under a set of parameters, we can try to find out why it happens through the form finding plot which makes it a useful tool (see figure 2.6).

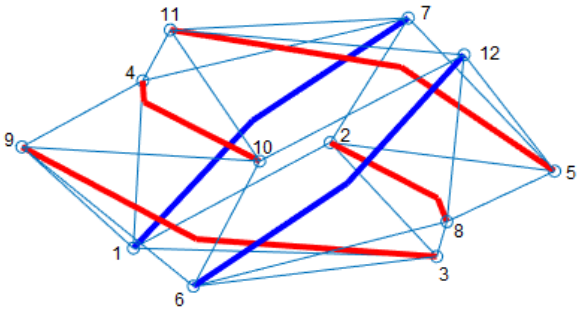


Figure 2.6: Example of a 3D plot of the tensegrity structure when at rest, with all nodes labelled. Nodes 1, 3 and 6 are touching the ground in this case.

2.2.1 Leg length

The first thing we investigate is how different values of the leg length l_l affect the structure during actuation. In figure 2.7 the compression angle of the structure can be seen with an applied torque of 50 Nm. With increasing leg length, the winding up of the torsion spring at the joint also gets more effective. This probably has to do with 2 factors: at first with the fact that all the cables have a compressing effect and all the legs have an extending effect. Since the stiffness of the torsion springs is kept constant, the reaction force at the end nodes from the joint of the legs is lower for longer leg lengths. Therefore the structure is also compressed more in its relaxed state for longer leg lengths. For this reason the difference in compression angle between the relaxed and the actuated state should be observed, and in figure 2.7 we can see

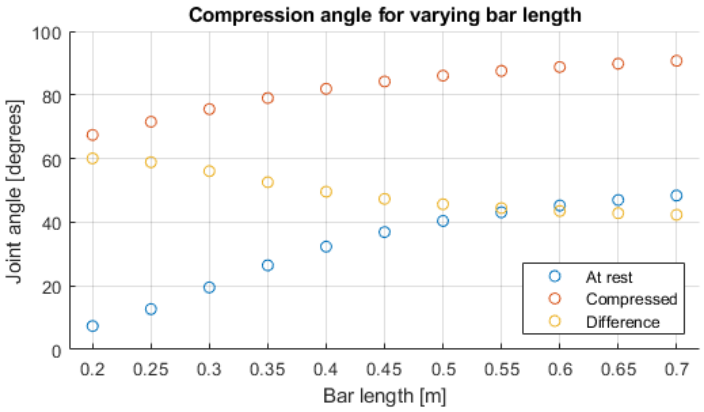
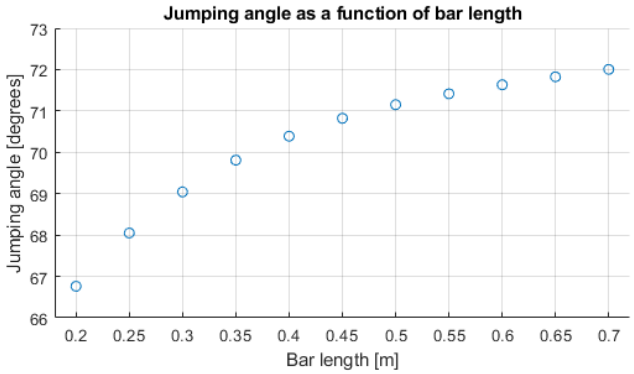
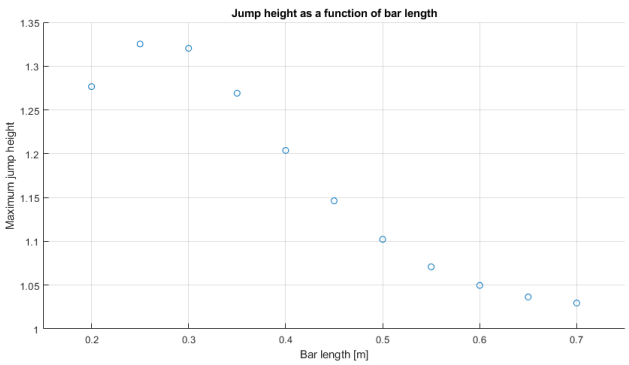


Figure 2.7: Plot of the compression angle of the jumping joints as a function of leg length l_l, with one of the plots showing the relaxed state and the other the compressed state.



(a)



(b)

Figure 2.8: **(a)**: Angle of actuated legs with respect to the ground for 50 Nm applied torque. **(b)**: Maximum jump height of the robot as a function of bar length.

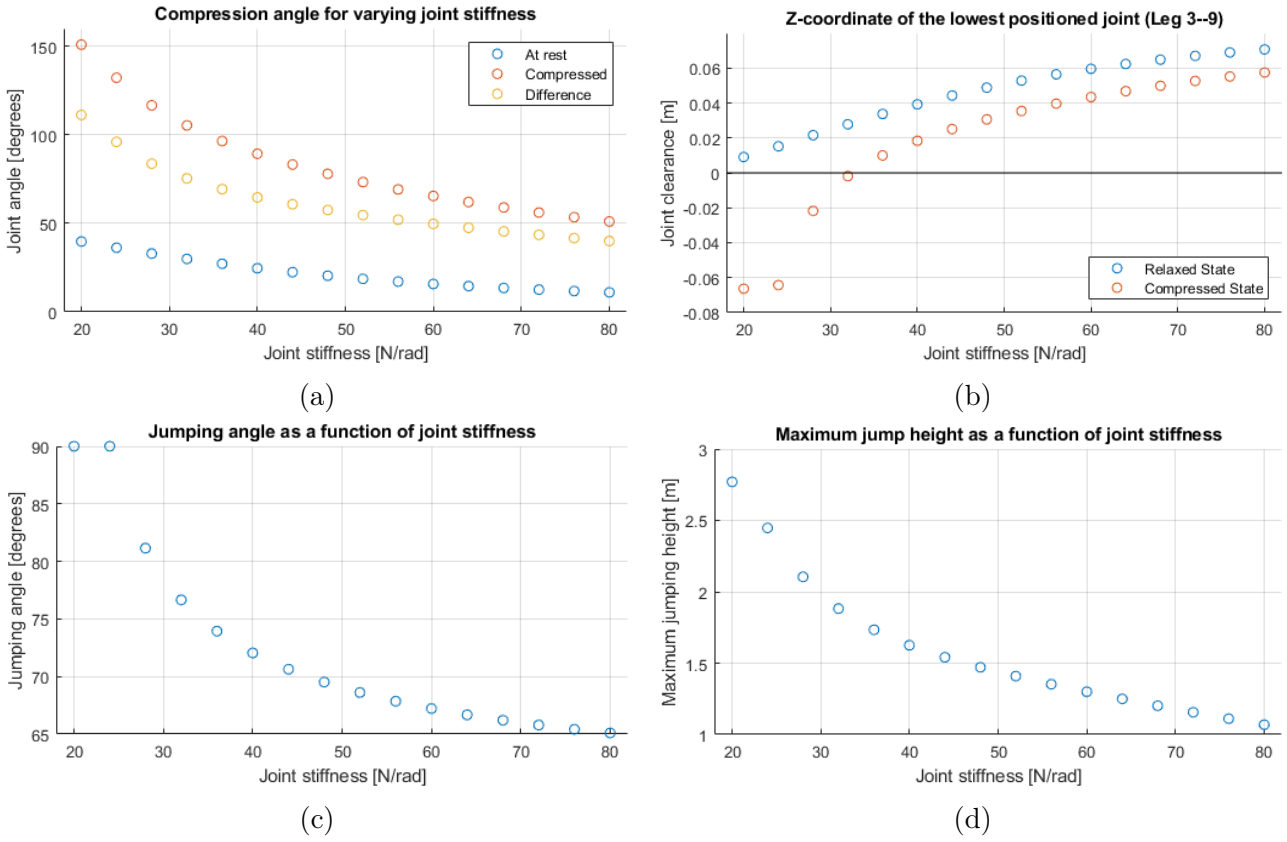


Figure 2.9: All of the above graphs are plotted with respect to the torsion spring stiffness, and compressed state has an applied torque of 50 Nm.

that there is a larger difference in angles for shorter leg length.

Another important factor is the angle under which the actuated legs will release, which will also be the general direction in which the robot will jump. Figure 2.8a shows us that the value for the leg angle gets higher as the leg length increases. This also affects the jump height in general, and in figure 2.8b the jump height is plotted. There is a peak around 30 centimeter leg length, from which we can extract the information that the jump angle difference has a smaller influence than the actuation leg compression angle. It is also beneficial to have a smaller robot because the weight is a big factor in the jumping performance of the robot.

2.2.2 Torsion spring stiffness

From figure 2.9a we can make an obvious observation, which is that the leg joint angle value gets higher for lower values of κ. There is a lower force resisting the compression by the torque of the motor. This is not necessarily a good thing, because when we look at figure 2.9b we see that one of the plotted joints goes below ground in the simulation. This means that in reality there would be another contact point between the two jumping legs. This would most probably destabilize the robot and reduce the efficiency of jumping. As such we have to make sure that the joint clearance remains above zero for both the relaxed and the compressed state. It might be desirable to have a low value of joint clearance when at rest, because this might improve the ability of the robot to keep rolling after it has landed. The rolling efficiency is not something we can evaluate in the Simulink model, but more so in the Simscape simulation.

In figure 2.9c we can see that the jump angle approaches 90 degrees as κ gets to 20 N/rad. It seems like there needs to be a stiffness of at least 30 N/rad to obtain a structure which would withstand gravity during compression, and this more or less agrees with the value for the stiffness we find in the joint clearance plot. A hint of discontinuity in the jumping angle plot

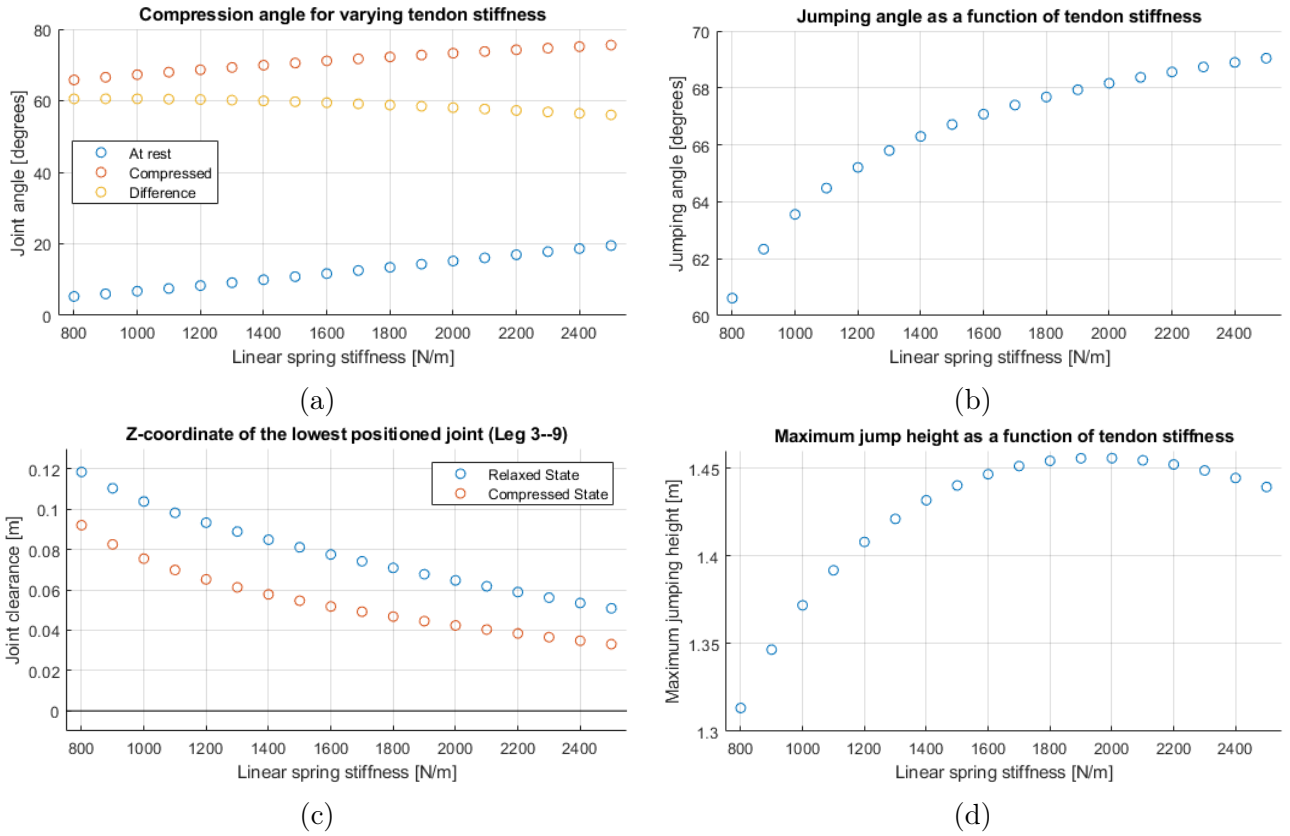


Figure 2.10: All of the above graphs are plotted with respect to the linear spring stiffness of the cables, and compressed state has an applied torque of 50 Nm.

is an indication of the form finding algorithm collapsing, which can be caused by the stiffness of one of the components getting too low. So when we decide on the parameters of the final design we have to decide by comparing the jump angle and joint clearance to make sure the robot will function. In figure 2.9d the jump height is plotted, and the theoretical maximum value gets highest when we minimize the torsion spring stiffness. So in the end there is going to be a trade off between the stability of the structure and the potential jumping performance.

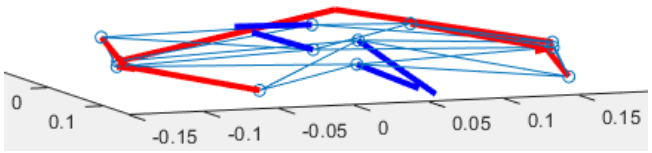


Figure 2.11: Example of a structure collapse in the form finding model.

2.2.3 Cable stiffness

The first way to vary the properties of the cable is to change the stiffness coefficient k of the linear springs. Among the graphs of figure 2.10 there is a quite striking feature that can be immediately noticed. In graph 2.10d there is a maximum around 2000 N/m. The reason for the maximum in this plot can be partially derived from graphs 2.10a and 2.10b since these are two big contributors to the jump height result.

The plot for the leg compression angle of figure 2.10a does not tell us the whole story, and thus we need the information of the other joint angles to know why this is happening. Therefore we plot the angles of other joints which are not actuated as well, which is seen in figure 2.10c.

Angle 1 in this plot is equivalent to the differenced values of the actuated joint angles of plot 2.10a. There are two other angles plotted, and each is an angle of one of the joints in a pair of parallel legs. It is not necessary to plot the other three angles due to symmetry of the structure. We see that the non-actuated joints actually stretched more as the cable stiffness increases. This means that during the jumping motion, part of the energy goes into attaining the equilibrium angles of 4 other joints. On the flip side, the jumping angle increases as the stiffness is raised. This trade off is the main reason for the local maximum in the jump height.

It is important to note that the optimal value of the cable stiffness is not fixed, but can also depend on other factors. From figure 2.10a we can also see that the collapse of the robot starts to happen below values of 900 N/m, because the jumping leg's angle has quite low values. The collapse is different to that of the structure when the torsion spring stiffness was too low. In this case the structure starts to tip over, with its center of gravity shifting towards node 3. It is suspected that the tendency of the robot to land in a position for 2-legged actuation is also lowered for lower values of the linear springs stiffness k.
The relation between the linear- and torsion spring stiffness is investigated further, with a plot seen in figure 2.12. Here we see multiple scatter plots with varying κ, and the torsion spring stiffness is also varied. We can also see that the greatest impact stems from the torsion spring, so we want to minimize this value in our final design without compromising the integrity of the structure.

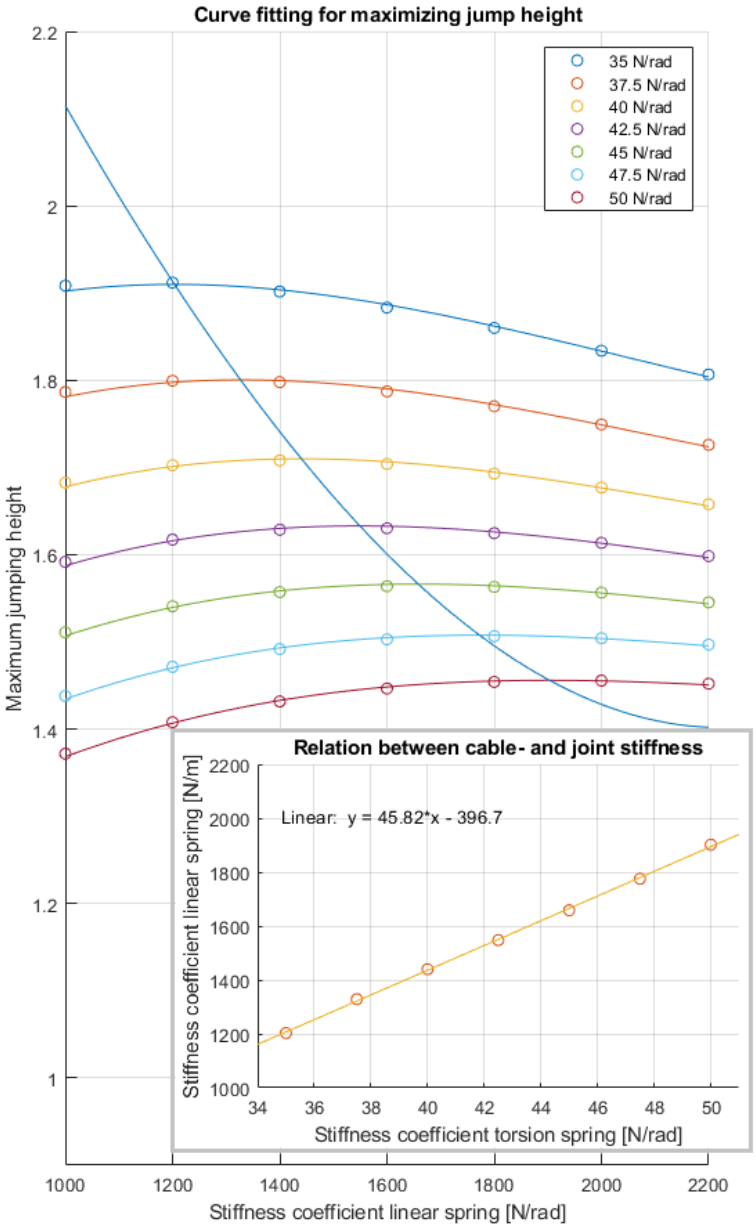


Figure 2.12: A fitted curve for optimizing the combination of joint- and cable stiffness. The local maxima found in the larger plot are shown as data points in the inset.

Thus we want to know the ideal value of the linear springs for a given value of the torsion spring stiffness. Thus the curves from figure 2.12 are fitted by a 3rd degree polynomial, and by solving the root finding problem of the differentiated polynomial we can find the local extremum for which the jump height is maximized. Now we know the optimal value of k for multiple instance of κ, we investigate what kind of relationship the two parameters have by fitting a line to that data (seen in the inset of figure 2.12).

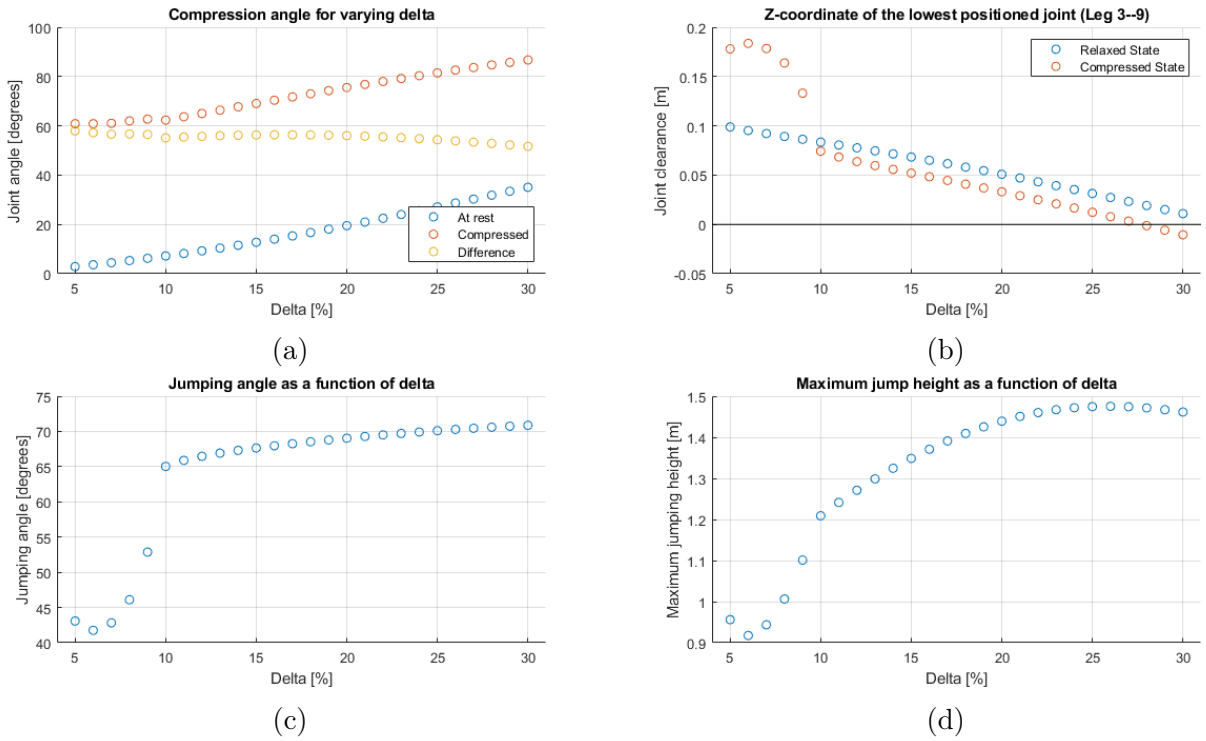


Figure 2.13: The graphs above are sweeps for the cable rest length deviation δ

Since it is the better choice to let the user decide what the value of joint stiffness should be, we fit for k. For this particular set of parameters, the linear equation is

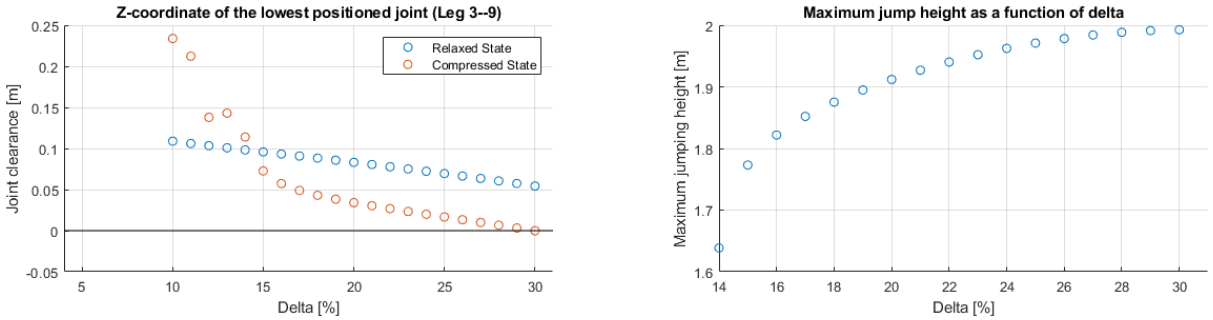
$$k = 45.82\kappa + 395.7 \tag{2.20}$$

For a joint stiffness of $\kappa = 35$ N/rad we then find a value of $k = 1208$ N/m.

2.2.4 Cable rest length

Looking at equations 2.3 and 2.4 we can see that the cable rest length can be varied by a percentage δ, with respect to the theoretical separation of the nodes with zero external load and the legs being in a complete stretched position. The definition of δ is chosen this way because this parameter can be kept constant as we change the leg length of the robot, and the cable rest length scales accordingly. Various plots of the result of δ variation can be seen in figure 2.13. The difference in angle compression see in figure 2.13a does not change much as δ increases, and thus the cable rest length consequently decreases. In figures 2.13b and 2.13c a drastic change in joint clearance and jumping angle respectively for values lower than 10% is seen. This indicates that the structure starts to tip over due to longer cable rest length, but only for the actuated state of the robot. The reason that it only happens during the actuated state is because the distance between some nodes get smaller than the cable rest length and the cable between these nodes goes slack, and therefore can not exert a force. This causes the robot to lose its integrity partially. Around $\delta = 6\%$ there occurs a local extremum in the jumping angle and the joint clearance. This happens because node 5 also touches the ground when the structure is compressed for low values of δ.

The most important result is that a higher value of δ in plot 2.13d corresponds to better jumping performance. This makes it doubly beneficial, because the tension forces caused by the prestressed cables are larger for large δ and thus the integrity of the robot is more easily preserved under impact. However it must be noted that a higher value of δ means that assembly of the robot is more difficult.



(a) (b)

Figure 2.14: Plots of the joint clearance and maximum jump height respectively for varying the δ value with a joint stiffness of $\kappa = 35$ [N/rad.]

2.2.5 Final parameter value selection

In determining the final parameters we look at the outcome of analysis of the cable rest length deviation δ with all the other parameters set at the right values. For $\kappa = 35$ N/rad we have calculated the linear spring stiffness using equation 2.20 and set it at $k = 1200$ N/m. Then we look at the plots in figure 2.14 and see that for values higher than 30% we have the problem of the joint in between the jumping legs touching the ground. Therefore we choose a slightly lower value, which is then 26% because this gives us a little bit of clearance.

The parameters shown in table 2.2 are used in the Simscape environment. The average jump height observed is around 50 centimeters, which is a little below the goal of having the robot reach twice its own height. It should be noted that the weight estimation is not very accurate, so the jump height of the prototype might be very different depending on the weight of the real motors and batteries. If they are a bit lighter it is very beneficial to the final result and the stiffness coefficient of the torsion spring might be reduced as well, improving performance even further.

	Parameter	Value	Unit
$\mathbf{l_l}$	Leg length	30.0	[cm]
$\mathbf{r_b}$	Bar radius	0.5	[cm]
$\mathbf{m}$	Attached mass	0.618	[kg]
$\mathbf{m_{tot}}$	Total mass	3.98	[kg]
$\boldsymbol{\kappa}$	Joint stiffness	35	[N/rad]
$\mathbf{k}$	Cable stiffness	1200	[N/m]
$\boldsymbol{\alpha_r}$	Joint rest angle	160°	[deg]
$\boldsymbol{\delta}$	Deviation	26	[%]
$\mathbf{l_c}$	Cable rest length	13.39	[cm]
$\boldsymbol{\tau_m}$	Applied torque	50	[Nm]

Table 2.2: Parameter values deemed to be optimal for a leg length of 30 centimeters.

2.3 Discussion

The details in the Simulink model can be increased a lot, especially when one looks at the assumptions and simplifications that are made in section 1. The model could be made more complicated by assigning the mass of the bars differently and also assign masses to the tendons. The battery and motor have a more localized mass situated somewhere next to the bar in reality, which would make the robot behave a lot differently. The mass of the motor is quite substantial and for now it has been evenly spread along the length of the bar.

One could also try to mathematically determine the angular momentum transferred to the entire body of the robot during extension of the legs. One could probably realize this by calculating the center of mass of the entire structure and determine its position with respect to the contact points of the two jumping legs.

It is quite common for the robot to get into the orientation where three non-parallel legs are touching the ground, and the robot chooses to jump with only one leg. This is more of a re-orientation jump so the robot can get into a position where it can use two parallel legs. The robot will not jump very high in this mode, and will also not cover a long distance. This is also the reason it was not investigated further, because the importance of form finding in one-legged jumping is not very prominent.

In the mathematical model the decision was made to define the position of the joint in the same plane as the two parallel legs. This method is chosen because the cable are all attached at the same infinitesimally small point at the end of a bar in the simulation. In reality a small offset is used and the tension of the cables also contribute to counteraction of rotation of the legs. This does not mean that the legs cannot rotate around their longitudinal axis at all, but when a leg rotates it will meet some resistance. This is a factor that can play quite a big role in the stability of the robot during compression, jumping and landing. Parameters that could be varied would be offset distance from the bar end node and slight asymmetry of attachment points.

During release of the robot from the compressed state there are also damping losses involved in the jumping motion, during the time the robot extends itself and is still touching the ground. The choice was made not to model this because it adds an unnecessary complexity to the model. We can evaluate the jumping motion in Simscape anyway, so there was no big motivation to bother including this in Simulink.

The advantage of making models in both Simulink and Simscape is that the Simulink model gives us a more exact calculation based on modelling equations, whereas in Simscape a more realistic depiction of the jumping motion can be observed. The optimization by curve fitting is one of the more valuable assets of the Simulink model. The optimal settings could then be tested in the Simscape model to observe how the robot would behave. One of characteristics that could not be seen in the form finding model was the tendency of the robot to roll into a position for two-legged jumping versus one-legged jumping. Since two-legged jumping is generally more desirable, a set of parameters that encourages this would have to be found. A mathematical optimum of two-legged jumping performance might have a tendency to roll into a one-legged jumping mode, and then the potential of the robot would not be realised.

The weights of the motors are given from the start at 600g, and this is coupled to the applicable torque of 50 N/m. Choosing a different motor might reduce the weight and also reduce the available torque. This is not investigated further because the motors have been decided on from the start so there is nothing to be changed about this part.

2.4 Conclusion

The form finding algorithm can give a good indication of the optimal parameters of the tensegrity structure. A smaller structure is more preferable for a couple of reasons including the reduced weight and shorter distance from joint to end node. The most important parameters to tune by the user are the joint stiffness κ and the cable rest length deviation δ, which need to be tuned according to plots for the jumping angle or the joint clearance. When these parameter values are decided, the cable stiffness k can be chosen by interpolation. The set of parameters that have been chosen for a stretched leg length of 30 cm can be seen in table 2.2. Although this report cannot give us a final conclusion on the best set of parameters, it certainly gives one a direction to look into for prototyping of the tensegrity robot.

Kinematic Model

3.1 Theory & Methods

The first adjustment that is made w.r.t. the form finding model is that the gradient descent method is replaced. Instead we calculate both velocities and positions from the calculated reaction forces occurring in the model. At first there do not need to be drastic changes to the model. The most important additions to the model are the addition of damping in both linear- and torsion springs, and the addition of a normal force.

On top of the assumptions mentioned in the main introduction, there are a couple more assumptions that apply to the real time simulation:

- There is no air resistance.
- The ground is flat, rigid and we assume dry contact (no viscous forces).
- Friction with a surface is calculated according to an extension of Coulomb friction equation, with the Stribeck effect incorporated (see section 3.1.3).

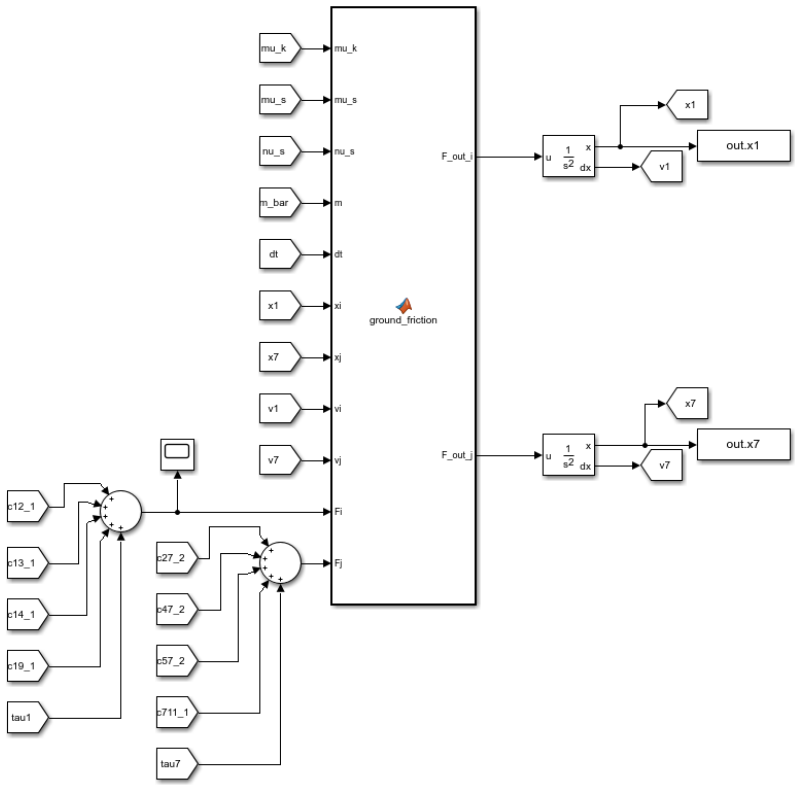


Figure 3.1: Function in Simulink of leg 1–7, in which the forces are added and it is checked whether one of the nodes is touching the ground.

3.1.1 Simulink Structure

The damping forces can easily be added to the existing `Force` functions as earlier shown in figure 2.3, because the force of damping acts in the same direction as the torsion spring force when viewed from the end nodes of the legs. For convenience, the integration of the acceleration

of the nodes is not done in individual node functions, but the nodes that are connected by a leg are pulled together in the same function (see figure 3.1).

3.1.2 Spring Damping

Damping losses are a central part in the behaviour of the tensegrity system. In both the torsion- and linear springs damping is present, although the behaviour and the damping coefficients are not be the same. For linear springs the damping magnitude is proportional to the relative velocity of the two joints that are connected by the cable member. For torsion springs the damping force is proportional to the angular velocity of the joint.

From Hsu et al. [11] we can find that the damping term of linear springs would look like:

$$\mathbf{f}_{s,k} = -c_s \frac{d||\mathbf{s}_k||}{dt} \frac{\mathbf{s}_k}{||\mathbf{s}_k||}$$

This translates to

$$\mathbf{f}_{s,k} = -c_s \frac{\dot{\mathbf{s}}_k^T \mathbf{s}_k}{||\mathbf{s}_k||} \frac{\mathbf{s}_k}{||\mathbf{s}_k||} = -c_s \frac{(\dot{\mathbf{s}}_k^T \mathbf{s}_k)\mathbf{s}_k}{||\mathbf{s}_k||^2} \tag{3.1}$$

This equation tells us that the magnitude of the velocity component (velocity dotted with $\mathbf{s}_k$) along the cable between the two nodes is multiplied by the normalized direction of the cable.

Similarly, the expression for damping in the legs

$$\mathbf{f}_{b,k} = -c_t \frac{d\alpha}{dt} \frac{\mathbf{b}_k}{||\mathbf{b}_k||}$$

$$\frac{d}{dt}\alpha(t) = \frac{d}{dt}(cos^{-1}(f))$$

with

$$f = \frac{l_l^2 - 2d(t)^2}{l_l^2}$$

Then according to the chain rule

$$\begin{aligned}
\frac{d}{dt}\alpha(t) &= \frac{df(t)}{dt}\left(\frac{-1}{\sqrt{(1-f^2)}}\right) \\
&= \frac{-4\dot{d}d}{l_l^2}\left(\frac{-1}{\sqrt{(1-f^2)}}\right) \\
&= \frac{4\mathbf{x}\cdot\mathbf{v}}{l_l^2\sqrt{1-\left(\frac{l_l^2-2d(t)^2}{l_l^2}\right)^2}}
\end{aligned}$$

Thus the damping term for legs becomes

$$\begin{aligned}
\mathbf{f}_{b,k} &= -c_t \frac{4\mathbf{x}\cdot\mathbf{v}}{l_l^2\sqrt{1-\left(\frac{l_l^2-2d(t)^2}{l_l^2}\right)^2}} \frac{\mathbf{x}}{||\mathbf{x}||} \\
&= -c_t \frac{2\mathbf{x}\cdot\mathbf{v}}{\sqrt{||\mathbf{x}||^2(l_l^2-||\mathbf{x}||^2)}} \frac{\mathbf{x}}{||\mathbf{x}||}
\end{aligned} \tag{3.2}$$

The damping coefficients c_s or c_t of a member can make a single mass spring damper system underdamped, critically damped or overdamped [15]:

- **Underdamping** (c $<$ 1): In this case the system will come to a standstill through an oscillatory motion, with the amplitude of oscillation of a member decreasing over time.
- **Critical damping** (c $=$ 1): There is no oscillatory motion, and the mass spring damper system comes to a standstill in the quickest way possible
- **Overdamping** (c $>$ 1): Again, no oscillation and in this case the system comes to a standstill in a slower fashion.

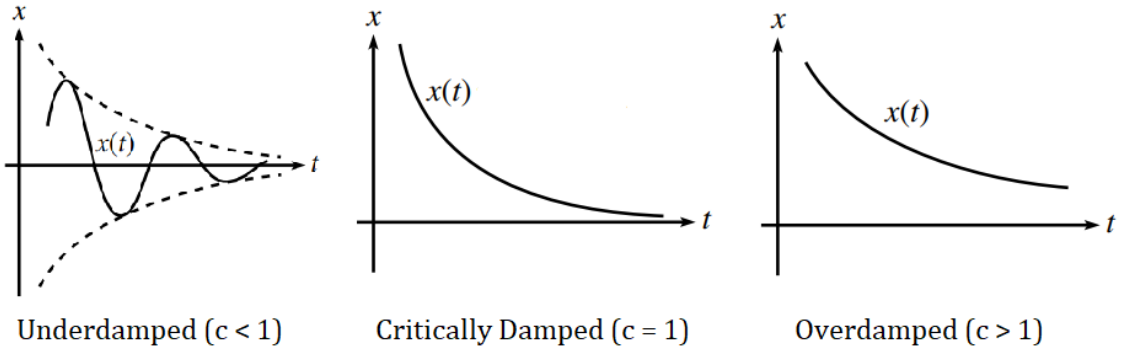


Figure 3.2: Types of damped oscillations as a function of time (image adapted from Morin [15])

An important thing to remember is that these general rules apply to a single mass spring damper system. In our case there is interaction of 30 coupled oscillators which makes the problem quite complex. The interplay between members can cause oscillatory motion of the entire structure, even though the individual coefficients of damping suggest otherwise.

3.1.3 Normal Force and Friction

There needs to be a definition of the normal force on nodes in order for the structure to take off, land and roll on the ground. This is why we incorporate both a normal force and friction in the model. In the form finding model the nodes which touched the ground (nodes 1, 3 and 6) were given the constraint that they were always situated on the xy-plane. Now this is not sufficient anymore, since all nodes need to be constrained to remain above ground.

The normal force on a node can be described as the combination of the sum of all z-directed components of the forces acting on a node (linear springs, torsion spring and gravity) in the exact opposite direction, and the z-direction momentum of the leg as it drops on the ground. The normal force should then exactly counteract this and the node should become stationary in the z-direction in one time step.

Friction is a result of the normal force and scales with its magnitude. In a review paper about friction in multibody dynamics by Marques et al. [16] the friction on a node is described as the combination of Coulomb Friction with the Stribeck effect. Coulomb friction magnitude is given by

$$F_C = \mu_k ||\mathbf{F}_N|| \tag{3.3}$$

in which $\mathbf{F}_N$ is the normal force and μ_k is the kinetic coefficient of friction. The Stribeck effect incorporates a dependence on velocity in the friction model. Static friction is larger than kinetic friction, and the kinetic friction decreases continuously as the velocity of the object w.r.t. the ground increases. The equation for the friction force looks like

$$\begin{aligned}\mathbf{F}_f = (F_C + (F_S - F_C)e^{-(\frac{||\mathbf{v}_T||}{\nu_S})^{\delta_\sigma}})\text{sgn}(\mathbf{v}_T) \\ +F_V\mathbf{v}_T\end{aligned} \tag{3.4}$$

with

$$F_S = \mu_s ||\mathbf{F}_N|| \tag{3.5}$$

and the signum function being defined as follows:

$$\text{sgn}(\mathbf{v}_T) = \begin{cases} \frac{\mathbf{v}_T}{||\mathbf{v}_T||}, & \text{if } ||\mathbf{v}_T|| \neq 0 \\ \mathbf{0}, & \text{if } ||\mathbf{v}_T|| = 0 \end{cases} \tag{3.6}$$

Similar to the equation for Coulomb friction, the static friction magnitude F_S is calculated by multiplying the coefficient of static friction with the magnitude of the normal contact force. A signum function as seen in equation 3.6 is a conditional equation which states that the term on the left is non-zero when the object has a non-zero velocity. The rightmost term in equation 3.4 gives us the viscous friction, but this is neglected since we assume dry contact ($F_V = 0$). Parameter ν_s denotes the Stribeck velocity, and δ_σ is an exponent that is dependent on the geometry of the contact surface (considered to be equal to 2). Now it can be seen that the friction force increases with increasing normal force, and higher relative tangential velocities result in lower values of since $\lim_{v\to\infty} e^{||\mathbf{v}_T||} \frac{\mathbf{v}_T}{||\mathbf{v}_T||}$.

In a paper on friction coefficients [17] we find that there is not much consensus on the coefficients of material interface contacts. However we can stake that the static coefficient should be larger than the kinetic coefficient of friction, and the static coefficient has common values between 0.3 and 0.6. Thus we set these coefficient to $\mu_s = 0.45$ and $\mu_s = 0.35$. The Stribeck velocity has a common value of around $\nu_s = 0.2$ [18].

The integrator function blocks in figure 3.1 have a built in limiting of the position and velocity. So this way we can constrain the structure only to be able to attain z-coordinate values equal or greater than zero. Within the `leg` function there still is a conditional statement which ensures that friction is applied the moment a node touches the ground. The normal force and the friction force on the node are added to the existing sum of forces from springs and gravity.

3.2 Results

The main result from this analysis would be the jump height and horizontal jump distance, and these are easily obtained from Simulink. Facets of the general performance can be the rolling or bouncing efficiency, which can be stated as the distance that can be covered from the point in time the structure touches the ground again after taking off in the jumping motion.

By varying the parameters in the same way as in section 2.2 we can see how much the parameter selection agrees between form finding and the kinematic model. Every time the kinematic model is activated with a different set of parameters, the form finding algorithm has to be used as well to ensure that initial positions of all nodes are consistent. The analysis of results can be extended a little bit as well, by varying the damping coefficients of the strings and of the torsion springs. In the next chapter the Simscape model is discussed as well, including a comparison to the kinematic model.

3.2.1 Leg length

There is one problem with sweeping over the bar length with the parameters found in the form finding model. Due to the tendon stiffness k and cable rest length deviation δ have been chosen such that the joint of leg 3–9 barely touches the ground, the joint will actually touch the ground if we run the kinematic model with longer bar lengths. Therefore we need to change parameters k and δ to different values so we can make a representative analysis. We therefore set $k = 1000$ N/m and $\delta = 15\%$.

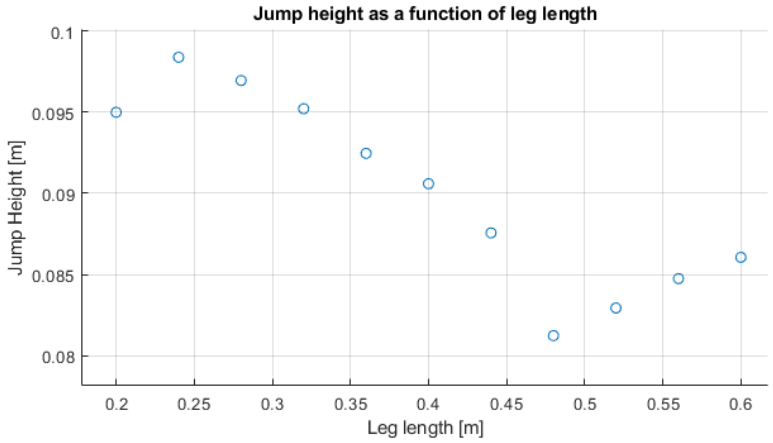


Figure 3.3: Jump height for varying leg length, with $k = 1000$ N/m and $\delta = 15\%$

In the figure above it is once again illustrated that for shorter leg length the jump height becomes higher, even though the weight of the motors and batteries remains the same as before. The same peak as before occurs around 25 centimeters leg length.

It is probably wise to remain with the 30 centimeter leg length. Extending the leg length has some implications for other parameters as well. The main drawback is that the stiffness and rest length of the cables have to be adjusted such that the joint displacement does not destabilise the structure during compression. This is such a big drawback that the leg length is kept at 30 cm.

3.2.2 Joint stiffness

The expectation from section 2.2.2 is that the coefficient of elasticity for the torsion spring should be minimized, with compromising the ability to jump. The fitted curve from figure 2.12 where a relation between the joint- and cable stiffness was found is not considered when analysing the influence of κ on the jump height. By also varying cable stiffness the influence on result may be confounded by this change. Thus we sweep from 35 to 65 N/rad stiffness, and the result is displayed in figure 3.4

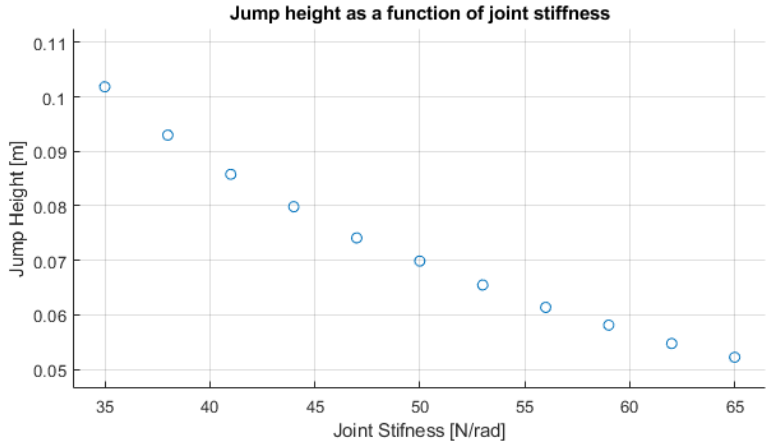


Figure 3.4: Jump height in the kinematic model as a function of joint stiffness.

This graph is very similar to the curve seen in figure 2.9d, which means that it still holds up that in the kinematic model the lowest possible joint stiffness should be used. Therefore the joint stiffness is still kept at 35 N/rad.

3.2.3 Cable stiffness

In the form finding analysis we found that there was an optimal value of jump height according to the sweep over cable stiffness (see figure 2.10d). But in figure 3.5 there is more of a linear relationship seen between the jump height and cable rest length. So it would be best to maximize the cable stiffness, while still considering the configuration of the structure with a joint stiffness of 35 N/rad.

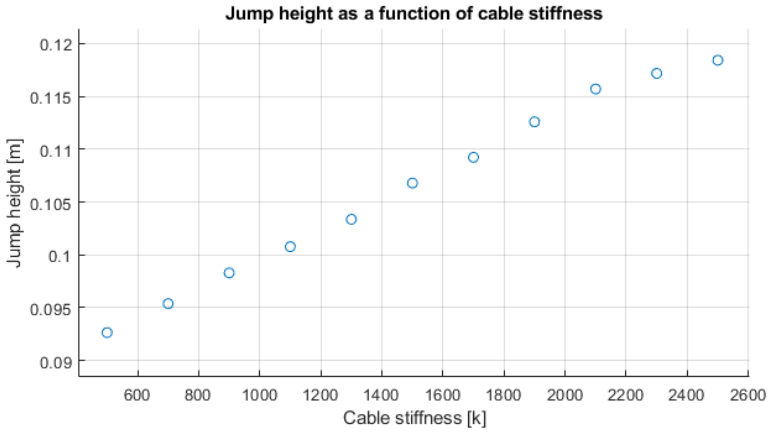


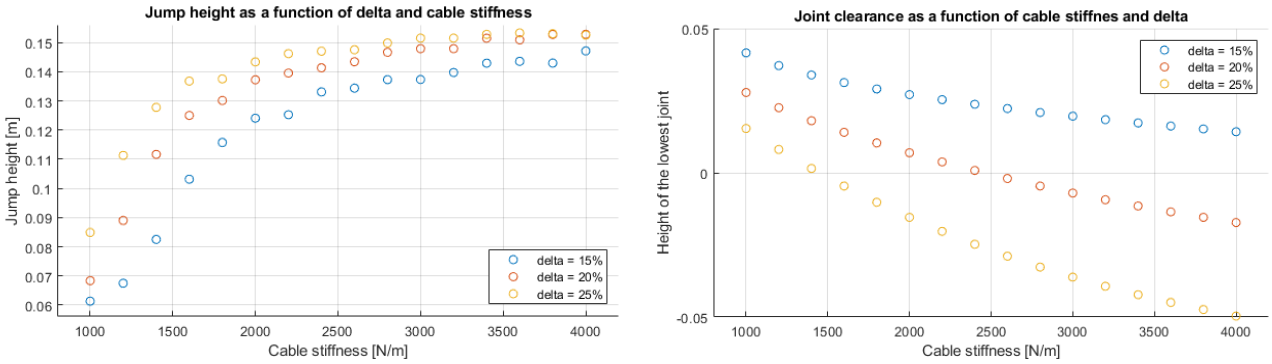
Figure 3.5: Jump height in the kinematic model as a function of cable stiffness.

3.2.4 Cable rest length deviation

From the analysis of the cable rest length deviation in section 2.2.4 we have learnt that there is a minimum of cable rest length deviation for which the robot will still jump effectively, and this threshold is around 10 % deviation for a leg length of 30 centimeters. This is also the point at which we start analysing the behaviour of the kinematic model.

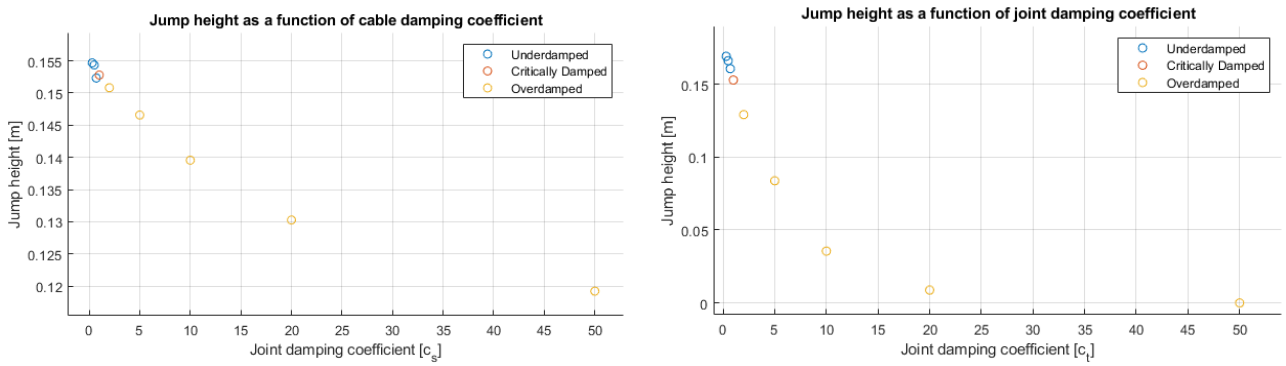
The graph of figure 3.7 is quite interesting compared to the graphs obtained from the form finding model (figure 2.14). There is a similar 'drop off' at low deviation, but the form finding graph (figure 2.14b) shows more of a maximum as δ goes beyond 30% deviation, whereas figure 3.7 shows a much more flattened curve from 15% deviation and higher. This makes the cable rest length deviation a factor for which we can optimize regarding rolling efficiency.

It now becomes clear that the impact of raising the value of the cable stiffness k has a positive impact on the performance of the robot in this kinematic model. Lowering the value of δ does not impact the performance as much as expected from the form finding analysis.



(a) (b)

Figure 3.6: Plots of the joint clearance and maximum jump height respectively for varying the δ and k values, with a joint stiffness of [35 N/rad].



(a) (b)

Figure 3.8: Plots of the jump height for varying the damping coefficients c_s and c_t respectively. The values of the dimensionless damping coefficients range from 0.3 to 50, with the blue dots signifying a data point in the underdamping regime, the orange dot shows the critical damping case and the yellow ones gives us the result for overdamping.

Therefore it is probably a good choice to look into optimization of the combination of these two parameters. We sweep over the cable rest length deviation and the cable stiffness, and these plots can be observed in figure 3.6. A general rule of thumb would be that a lower

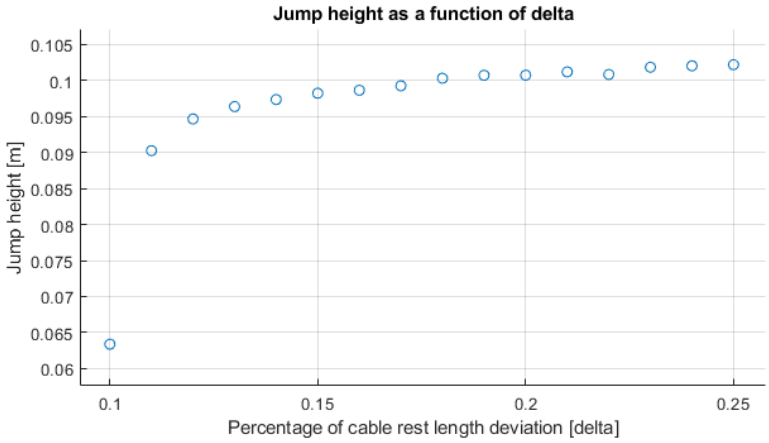


Figure 3.7: Jump height in the kinematic model as a function of cable rest length deviation.

value of δ allows for higher values of cable stiffness regarding the joint clearance, which is seen in figure 3.6b. The maximum cable stiffness for a δ of 20% would be around 2200 [N/m], but if we choose the rest length deviation to be 15% then we can go beyond a stiffness of 4000 [N/m]. The combination of these values can be evaluated in the graph of 3.6a and there we see that the lowest δ-value together with a stiffness of 4000 [N/m] corresponds to a slightly higher recorded jump height than the parameter combination at $\delta = 20\%$, albeit that the difference is minimal. This leaves the user with a design choice, where the trade off between these two parameters can be balanced at the will of the user.

3.2.5 Damping Coefficients

The two spring damping coefficients (c_s and c_t) can now be tested as well, which was not the case for the form finding model. The static nature of that analysis was not the environment for damping coefficients to be varied. The range between which the damping is evaluated goes from 0.3 in the underdamping regime, to 50 far into the overdamping range. When a sweep is done over one of the damping coefficients, the other one is set to critical damping values ($= 1$).

First thing to note is that when we look at the case where both damping coefficients are at the critical damping value ($c_s = c_t = 1$) individual members that are coupled to other members do not show the behaviour that is shown in the theory section (see figure 3.2). The structure experiences multiple oscillations instead of approaching the equilibrium position in the critical damping fashion. Only if the damping coefficients are raised into the overdamping regime this behaviour could be obtained.

For this part of the analysis the cable rest length deviation δ is set at 20% and the cable stiffness $k = 2200$ N/m, as a consequence of the additional analysis in the previous section. The results from varying the damping coefficients independently from each other can be found in figure 3.8.

As can be expected, the largest jump height is attained at low damping coefficients. Although the performance seems better, the low damping coefficient also means that the robot behaves somewhat erratic. The whole structure keeps vibrating for a couple of seconds which makes the behaviour during landing unpredictable. In figure 3.8a we can see the results from sweeping over the cable damping coefficient c_s. The underdamping regime gives us the greatest value for jump height. The difference are not big between the critical damping case and the lower values of the overdamping regime ($c_s < 10$) are not very significant, so strings in the overdamping range are not yet excluded. Moreover, the structure seems a bit more stable when we look at the animation for overdamped strings. It is thus probably wise to choose strings that are in the lower overdamping regime.

The damping in the joints is a different story though, which is mainly because the effectiveness of jumping greatly depends on the torsion springs. Any energy dissipated in the torsion springs during extension is a direct reduction of the potential energy, and thus a reduction of the maximum jump height. This can be clearly seen in the graph of figure 3.8b. In the underdamping regime the values of maximum jump height lie very close together, and the critically damped torsion spring only generates a slightly lower value. But when there is a transition to the overdamping region, the jump height keeps dropping. At some point, the structure is not even able to take off anymore. The overdamping causes slow extension of the legs and there is less momentum generated as c_t becomes larger.

With this newly acquired knowledge the obvious choice can be made that the damping coefficient of the torsion springs c_t should be minimized, while the cable damping coefficient can best be chosen in the overdamping regime. How far one would want to go into the overdamping regime depends on how stable the designer would desire the jumping tensegrity, but a value of string damping higher than $c_s = 100$ might be detrimental to the performance of the robot.

3.3 Discussion

Of course we can not fully rely on the conclusions taken from this chapter, as there is no close agreement between the established Simscape model and the kinematic model. We can still say something about the process of parameter optimization.

There is not much investigation done regarding the friction with the ground. We assume a completely flat area with constant coefficient of friction between the contacting node and the ground. Furthermore we don't know how big the contacting surface area would be, so the magnitude of friction forces in the model come with a large error margin. It could be that the jumping legs slip during extension, which will significantly reduce the effectiveness of the jumping motion. Not only will the jump height decrease in that case, but the covered distance will also be reduced in that case.

We have not included elasticity of bars, so the individual bar sections of the legs are rigid. But even going further, we have not investigated yet how the solid sections of the tensegrity structure react to bending forces. After all, we want the robot to be robust and to be able to

take a hit. If the forces on a bar section are too great during extension, or even during the compression phase of actuation, the bar sections might break. In a regular tensegrity structure, bending forces are not in play because the compression force on a bar is always along the length of the bar. The nodes are placed exactly at the ends of a bar, so it is not possible for the bar to experience bending as the force will always act along the length of the bar. But in the case of a tensegrity structure with legs that have a joint in them, there can be forces experienced by a bar section that are not along the length of the individual bar section. The moment at which the bars are mostly likely to fail is when the joint reaches its maximum angle and bounces back from the 160° stretched position.

The cable rest length deviation does not seem to be such a limiting factor as in chapter 2. There we would just maximize the cable rest length deviation such that the lowest joint barely touched the ground in compressed state, and that was the end of it. In this chapter we can look at a trade off between the cable rest length and the cable stiffness, where multiple combinations of values of these parameters yield similar result regarding jump height. The actual selection of parameters then becomes more of a design choice than purely a maximization of jump height. Things like rolling efficiency, robustness and availability of materials are things that one could consider.

Instead of using the exact same spring coefficients, rest lengths- and angles we could implement a randomized perturbation of these parameters. Small deviations from the desired value are realistic, and it could give us a good inside in the variation of jumping motion if the simulation is done a lot with slightly different parameters. This is already done in the Simscape environment and it gives a nice insight into the motion of a slightly imperfect tensegrity structure during the jumping motion.

3.4 Conclusion

In this chapter a kinematic model was described and the results of its simulations were analysed. A comparison was made to the form finding model of chapter 2, where some parameters like the bar length and the joint stiffness yielded similar results, and analysis of other parameters like the cable stiffness- and rest length deviation have given us new insights in the parameter selection process.

The results from this section should be taken with a grain of salt however, because there is no great similarity to the motion of the tensegrity structure in the Simscape environment. In order to improve on this model, a next iteration of the tensegrity simulation is the implementation of the Euler-Lagrange description, which should yield far more accurate results if implemented correctly.

Simscape Model

In this chapter the Simscape model of the jumping icosahedron tensegrity robot is introduced and explained. This model is used as more of a validation tool for the Simulink models to be compared to. For that reason this is mostly a theoretical chapter where the working mechanisms of the model are explained.

Within Matlab one can enter the Simscape environment where physical elements can be modelled in block diagrams in order to create a multibody system [14]. The kinematics are all based on the user defined geometries, forces and multibody interactions. The Simscape model is used as a verification tool to see how a structure would react in an environment which resembles reality. The optimal parameters found in the Simulink environment need not be the best set of parameters in the Simscape environment, because the Simulink model only is a simplification. Maybe there is a factor not included in the mathematical model which is of great influence in the real situation. On of the advantages is that we can see how the robot would behave in a more realistic environment and we can also test the decision making of the robot.

4.1 General Framework

There are 4 block components in Simscape which lie at the base of the Simscape model, and these give an overarching definition of how physical components behave in the environment. These are:

World Frame

The world frame is a Cartesian coordinate system with respect to which the position of the physical elements can be defined. Say that you want to define the initial position of a bar. Then you can place this bar anywhere in the world frame if you define a displacement from the origin of the world frame and you define its orientation by rotational transforms. These fall under the label of *Rigid Transforms*.

Solver Configuration

Another function which is standard in every Simscape model. Here some parameters regarding the solver can be adjusted, like the way the matrices are stored. This function mostly has to do with the accuracy and speed of the solver.

Mechanism Configuration

This is a function in which the user can apply model-wide parameters which are either constant or time-varying. In the case of this model the gravitational constant is defined in this block.

Infinite Plane

The infinite plane defines the ground in this model. The robot will fall onto the plane at the start of the simulation, from which it will experience a normal force. For this normal force to exist there should be a defined *Spatial Contact Force* block that applies a contact condition the plane on one side, and on the other side a leg of the robot. These blocks can be seen in part A of figure 4.1.

4.2 Strings

The strings are defined in an equivalent way to the strings in the form finding model in the next chapter. A function block *Spring Damper Force* defines the spring with parameters of natural length, stiffness and damping coefficient. The ports are attached to the frames of the corresponding nodes, and the forces acts on the line between the origins of the two attached frames.

A difference with respect of the Simulink model is that in this case a random perturbation of cable rest length is added. A random value between -0.3 and 0.3 centimeter is added to the existing cable rest length. The problem that occurs when there is no perturbation is that the robot behaves 'too perfect'. When the robot is dropped at the start of the simulation, it does not tip over. Instead it is standing balanced on two legs which is quite an unrealistic situation.

4.3 Legs

This part of the model is split up into six equivalent subsystems. The block diagram of one of the legs can be seen in figure 4.1. A *6-DOF Joint* is defined in the middle to signify that the objects attached to it are free to move and rotate in space.

The legs of the robots are defined as a combination of two rigid bars. At the point the bars touch, a *Revolute Joint* is placed which allows for rotation in a plane. Inside this block the rest angle and the spring stiffness of the joint can be defined. Two of the ports are connected to the frames associated with the contact point of the two bars. A third port to the *Revolute Joint* is the applied torque, which we will get back to on a later note. The batteries and motors are placed close to the joint, and for a pair of bars they are both placed at the same side of their respective joints (see figure 4.2). Rigid transforms are applied and by a *Weld Joint* block the rigid objects are welded to the bars of the legs at that particular position. The motors are quite a bit heavier than the batteries and thus contribute to the motion of the robot more. The legs, batteries and motors are shown in the animation but not the linear springs, because it is hard to observe the behaviour with 24 cables in view.

The attachment points of the linear springs are not exactly at the same point at the end of a bar. There is an offset of the order of a centimeter, which makes the calculation in the form finding model very complicated. But in Simscape the multibody simulation can pick this up more easily. In parts C of figure 4.1 there are rigid transforms which define the offset relative to the position of the end of a bar. In this model there is no constraint on the rotation of the leg, whereas in the form finding model it was a necessity to have no rotation. The legs rotate around their body axis during the compression motion. In turn this means that some potential energy gets lost in the rotational motion of the legs, which worsens the performance of the robot w.r.t. the Simulink model where the legs are set not to rotate.

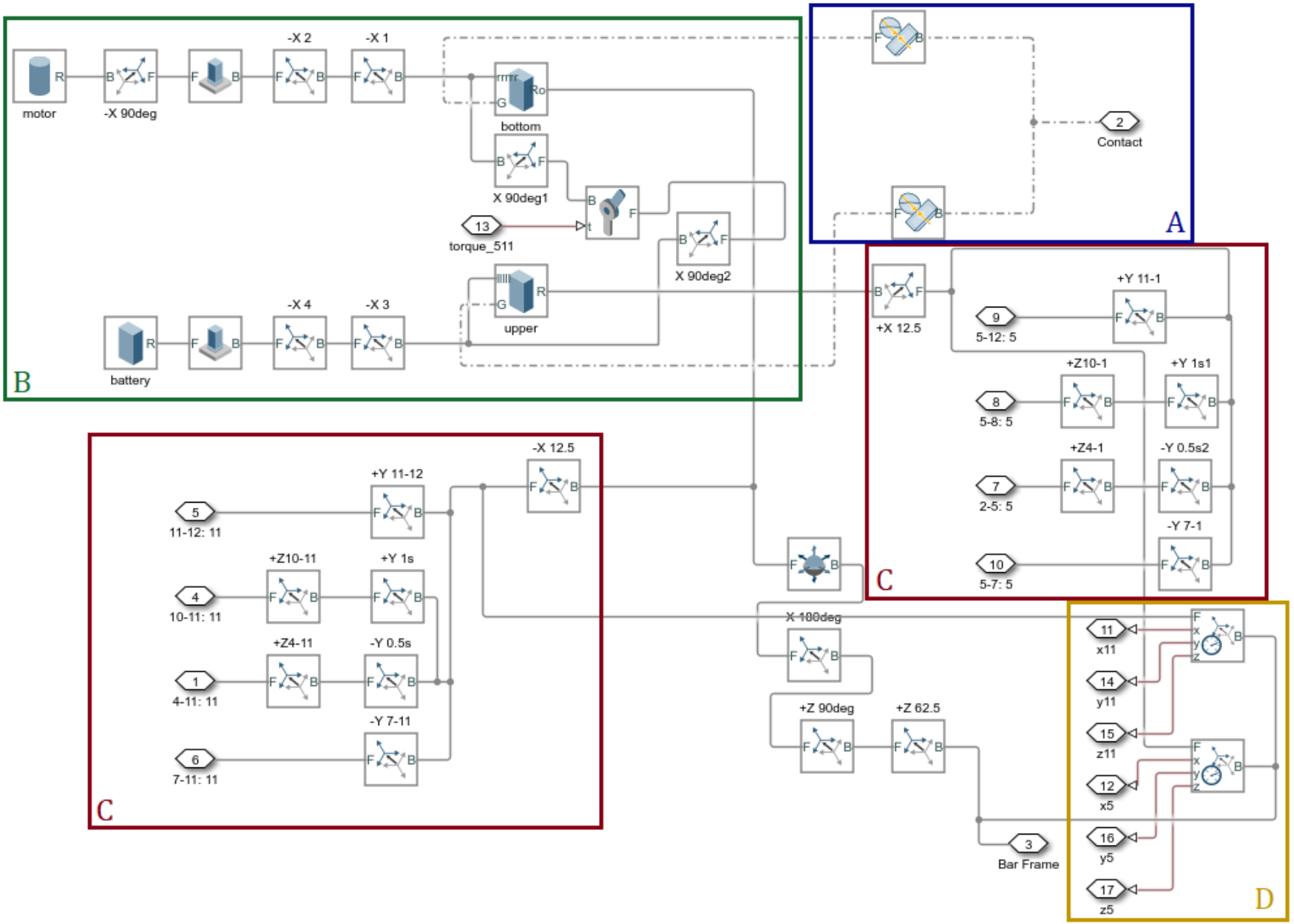


Figure 4.1: Block diagram for a single bar in Simscape. **A)**: The contact forces that are defined by the infinite plane. **B)**: Block defining the leg, together with the welded on battery and motor. **C)**: Attachment points of the strings at both ends of the leg. **D)**: Measurement of coordinates of the end nodes of the leg.

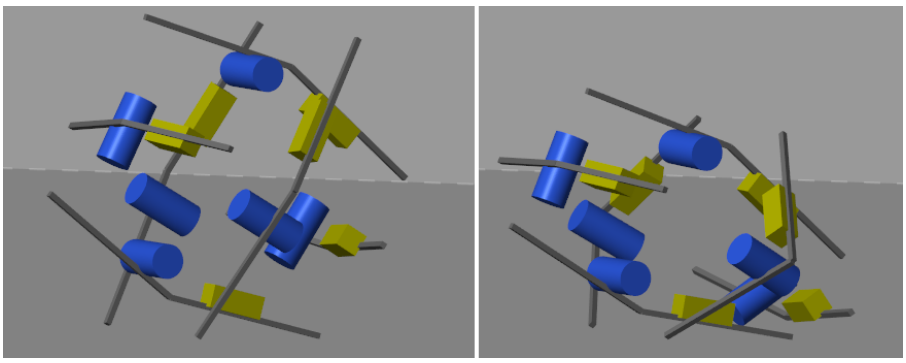


Figure 4.2: Cutouts of pictures from the animation produced by the Simscape simulation. The left picture is the tensegrity structure at rest, and the right one is at its most compressed state. The blue cylinders are the motors, the yellow rectangles are the batteries.

4.4 Decision & Actuation

The Cartesian coordinates of the end of every leg are measured in part D in figure 4.1 and is routed to a decision function. This function has the parameter *jump* as its main output, which is a vector of six entries that contain either a 1 or a 0. At every time step it is checked if the robots movement is below the threshold value (i.e. the robot is static). If so, the function continues and looks at the z-coordinate value of each end node. The 3 nodes with lowest value are selected and the angles between the legs are compared. If two legs are parallel (up to some threshold) then the function decides to make these two legs jump, as it recognizes that the robot is in 2-legged jumping mode. If all three legs are not parallel then on of the legs is arbitrarily chosen to jump.

Next, the vector *jump* is routed to the actuation function which has got all six torques as outputs. There are also outputs that check if the robot is idle, that define a time parameter used during compression motion and a parameter which renews the *jump* vector if the jumping motion is finished. These logical values or vectors are stored in memory blocks each time step and reused in the next one. Inside the function block conditions are defined which can change the outcome of these logical parameters.

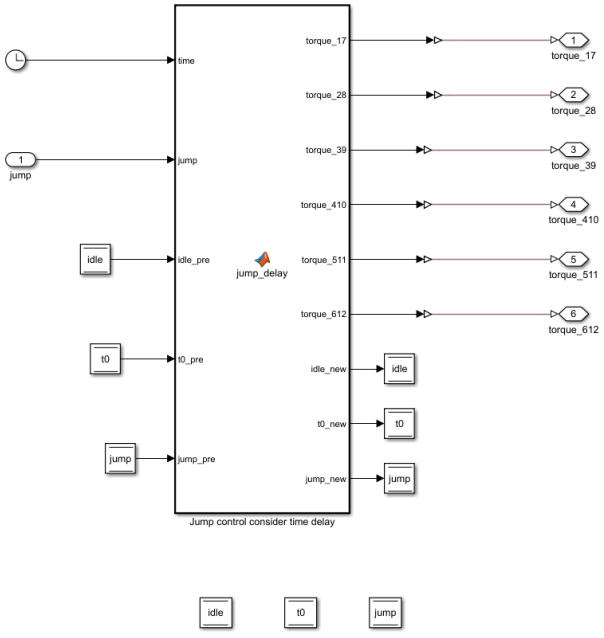


Figure 4.3: Jump control function block in Simscape

The `idle` parameter is 1 when the robot is ready to jump, and it is 0 when the jump is in progress. We will first look at the case that the robot is ready to jump. There are two possibilities in this case:

- There is a non-zero element in the `jump` vector, in which case the `idle` parameter switches from a 1 to a 0. At this particular timestep the torques are set to zero, since the torque can only be applied when the jump is in progress.
- It is also possible that `jump` exists of only zeros, in which case nothing changes in the next time step.

If the `idle` parameter indicates that the jump is in progress, three possibilities rise to the occasion.

- The actuated legs are still busy compressing. A time span `dt` is set inside the function, and this makes sure that the torque is gradually applied to the joint in the span of half a second.
- If there is a time delay set by the user, one of the actuated bars will release later than the other. this can b done to steer the robot in a direction when jumping using 2 legs. For all analysis in this report the time delay is set to zero so we can skip this part of the function
- The last option is that after the final time step of compression, the torque is set to zero again. This releases the torsion springs inside the legs to extend again, and the actual jump of the robot will take place. The `idle` parameter is set to 1 again, which means that in following time steps the robot will not jump unless the movement threshold is reached again.

The torques are routed to the joint of their respective leg. Inside the model the jump height, airtime, distance covered, etc. can be evaluated and compared to results from the form finding and kinematic models. In the case of the form finding model it is used to see whether the set of parameters found in the results section actually yields good performance.

4.4.1 Comparison to Kinematic Model

In figure 4.4 it can be clearly seen that the motion of the two models differs greatly. Both the upwards and downwards motion of the structure has a greater acceleration in the Simscape simulation then in the Simulink kinematic model, even though the gravitational acceleration constant is equivalent in both model. As such it is now unclear where the different outcomes originate from, but for now it is assumed that the Simscape multibody simulation is the more accurate one.

Although this is not the result one expects, there are still things we can learn from the model. The results that are discussed in chapter 3 are still relevant amongst each other, but can not be directly compared to the results from the Simscape model or to those of the form finding model. The best way to look at the Simulink results is that we can choose the set of parameters more easily. Then the Simscape model is more of a demonstration tool.

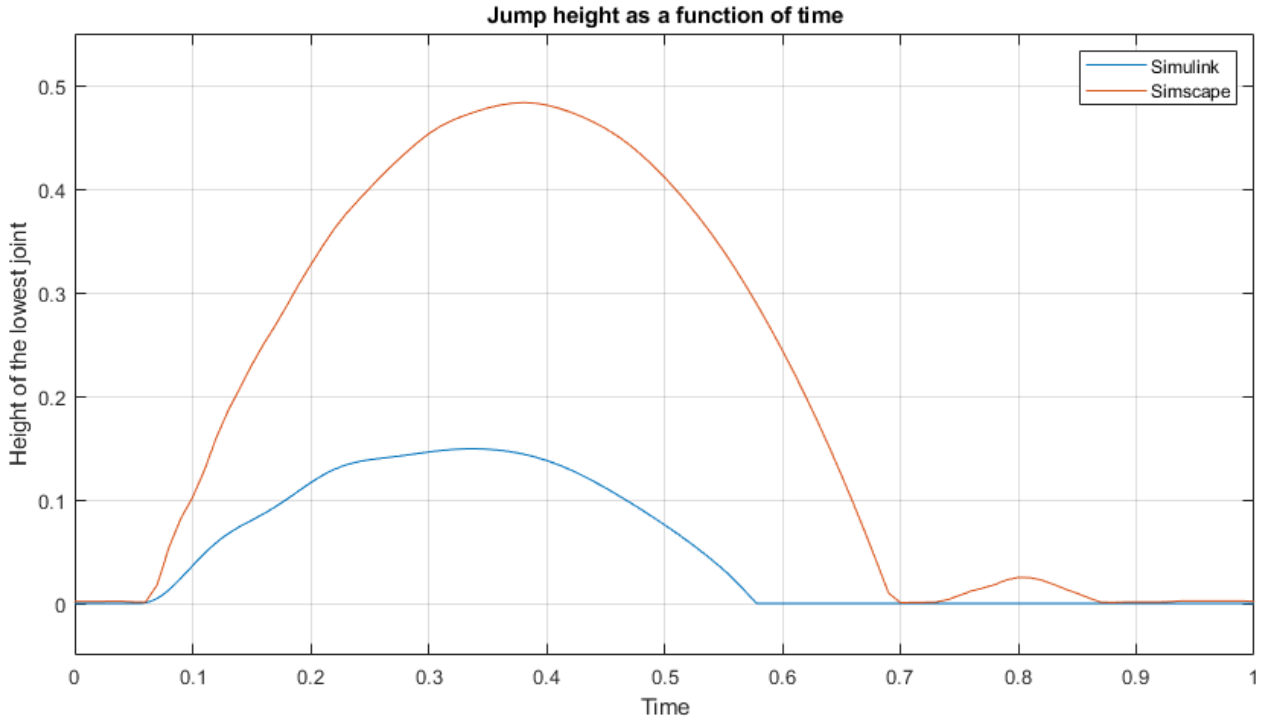


Figure 4.4: Jump height of the tensegrity structure in Simscape and Simulink respectively

Euler-Lagrange Model

The first motivation to adapt the model to the Euler Lagrange description was that it contains a useful way of implementing the angular momentum and inertia of the rigid bodies in the tensegrity system.

Conversion to the E-L model is different endeavor than the conversion to the kinematic model of the previous chapter. Graph theory is used in a clever way to describe the position coordinates of every node in a single column vector $\mathbf{q}$ of size $3n \times 1 = 36 \times 1$, and the velocities are shown in vector $\dot{\mathbf{q}}$ of the same size.

This chapter is largely based on the framework written by Hsu et al. in their paper on the Lagrangian dynamics of tensegrity system. Parts about potential gravitational energy and potential energy contained in the strings are equivalent to the theory written in [11]. There are also some instances where a mathematical derivation will be too long to be included in this work. If that is the case, the reader is referred to the work by Hsu [11].

5.1 Definitions

We define two connectivity matrices $C_l \in \mathbb{R}^{n_l \times n}$ and $C_s \in \mathbb{R}^{n_c \times n}$ for respectively the legs and the cables, with $n_l = 6$ and $n_c = 24$. The bars, centers of mass of the bars and the strings can be defined as:

$$\begin{aligned} \mathbf{b}_k &:= \left((\boldsymbol{\theta}_k^T \mathbf{C}_l) \otimes \mathbf{I}_3\right) = \mathbf{X}_k \mathbf{q} \\ \bar{\mathbf{b}}_k &:= \frac{1}{2}\left((\boldsymbol{\theta}_k^T |\mathbf{C}_l|) \otimes \mathbf{I}_3\right) = \bar{\mathbf{X}}_k \mathbf{q} \\ \mathbf{s}_k &:= \left((\boldsymbol{\eta}_k^T \mathbf{C}_s) \otimes \mathbf{I}_3\right) = \mathbf{Y}_k \mathbf{q} \end{aligned} \tag{5.1}$$

In this notation $\boldsymbol{\theta}_k^T$ and $\boldsymbol{\eta}_k^T$ are vectors in $\mathbb{R}^{n_l}$ and $\mathbb{R}^{n_s}$. The k-th elements of these vectors are one and the rest is zeros. The Simulink model is initialized using $\mathbf{X}_k$,$\bar{\mathbf{X}}_k$ and $\mathbf{Y}_k$ which are 3×36 matrices.

In this section we are only concerned with the legs of the structure, since we use the simplification that the strings carry no mass. The relative velocity of a bar is defined as

$$\dot{\mathbf{b}}_k := \dot{\mathbf{n}}_{j_k} - \dot{\mathbf{n}}_{i_k} = \boldsymbol{\omega} \times \mathbf{b}_k$$

This can be rewritten to describe $\boldsymbol{\omega}_k$ in terms of the bar coordinates and velocities, under the assumption that the bars cannot rotate around their longitudinal body axis ($\boldsymbol{\omega} \cdot \mathbf{b}_k = 0$).

$$\boldsymbol{\omega}_k = \frac{\mathbf{b}_k \times \dot{\mathbf{b}}_k}{d_{b_k}} \tag{5.2}$$

The parameter d_{b_k} is the node distance of the k-th leg, and then we can say $\mathbf{b}_k^T \mathbf{b}_k = d_{b_k}^2$. When we define a bar frame with the body axis defined as $(\hat{\mathbf{b}}_k, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3)$, the angular velocity can then be written in terms of only $\hat{\mathbf{b}}_2$ and $\hat{\mathbf{b}}_3$ as $\boldsymbol{\omega}_k = \omega_2 \hat{\mathbf{b}}_2 + \omega_3 \hat{\mathbf{b}}_3$. The moment of inertia of a bar in this body-fixed frame is:

$$I_{\mathbf{b_k}} = \begin{bmatrix} \frac{m \cdot r^2}{2} & 0 & 0 \\ 0 & \frac{m}{12}(3 \cdot r^2 + d_{b_k}^2) & 0 \\ 0 & 0 & \frac{m}{12}(3 \cdot r^2 + d_{b_k}^2) \end{bmatrix}$$

It might be obvious but it should be stressed that this is a simplification of the real situation, because the legs of the jumping tensegrity structure have a joint and are not straight rigid bars. In order to make a more accurate representation this simplification should be dealt with, but it will not be done for now. The angular momentum of a leg can now be written as:

$$\begin{aligned}\mathbf{h}_k &= \mathbf{I}_{b_k}[0 \quad \omega_2 \quad \omega_3] \\ &= \frac{m(3r_k^2 + d_{b_k}^2)}{12d_{b_k}^2}\mathbf{b}_k \times \dot{\mathbf{b}}_k\end{aligned} \tag{5.3}$$

For further explanation on the derivation the reader is referred to [11]

5.2 Lagrangian Dynamics

The Lagrangian is defined as $\mathcal{L} := T - V$. The general equation of motion has the form

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \lambda^T(t)\frac{\partial \mathbf{R}(\mathbf{q})}{\partial q_i} = \mathbf{f}^T\frac{\partial \mathbf{q}}{\partial q_i}$$

Because the term $\frac{\partial \mathbf{q}}{\partial q_i}$ is the ith column of an identity matrix $\mathbf{I}_{3n}$, and we substitute $\mathcal{L} = T - V$, then we can rewrite the above equation in vector form as follows:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) + \frac{\partial V}{\partial \mathbf{q}} - \lambda^T\frac{\partial \mathbf{R}(\mathbf{q})}{\partial \mathbf{q}} = \mathbf{f}^T \tag{5.4}$$

Therefore it is valuable to know the kinetic and potential energies in the entire system to get a description of the state of a tensegrity structure. The equations for dampers are equivalent to the equations in the previous chapter for linear springs 3.1 and for torsion springs 3.2.

5.2.1 Potential Energies

The potential energies of the system are contained in different parts, which are constituted by gravity and energy contained in all springs. The total gravitational potential energy is written as

$$V_g := -\mathbf{g}^T\left(\sum_{k=1}^{n_b} m\bar{\mathbf{X}}_k\right)\mathbf{q} = -\mathbf{G}^T\mathbf{q} \tag{5.5}$$

with $\mathbf{q} = [00 - 9.81]^T$.

The cables are modelled as linear springs according to Hooke's Law, just like before in equation 1.1. In different notation the total potential energy stored in linear springs is written as

$$V_s = \frac{1}{2}\sum_{k=1}^{n_s} k(||\mathbf{s}_k|| - l_s)^2.$$

There is still the condition that there can be no energy in strings with $||\mathbf{s}_k|| \leq l_s$. Therefore the parameter force density is defined as

$$\sigma_k := k\left(1 - \frac{l_s}{||\mathbf{s}_k||}\right) \tag{5.6}$$

Converting the equation to be written in terms of $\mathbf{q}$ we get (by skipping a few steps)

$$V_s = \frac{1}{2}\mathbf{q}^T\mathbf{Y}\left(\frac{\boldsymbol{\sigma}^2}{k} \otimes \mathbf{I}_{3n}\right)\mathbf{q} \tag{5.7}$$

Here $\boldsymbol{\sigma}^2 = [\sigma_1^2 \cdots \sigma_{n_s}^2]^T$ is a 36×1 vector and $\mathbf{Y} = [\mathbf{Y}_1^T\mathbf{Y}_1 \cdots \mathbf{Y}_{n_s}^T\mathbf{Y}_{n_s}]$ a 36×864 size matrix.

There is also potential energy stored in the torsion springs, which is the same as equation 2.9. If we rewrite it in terms of $\mathbf{q}$ we get

$$\begin{aligned} V_l &= \frac{1}{2}\sum_{k=1}^{n_l} \kappa \left(\cos\left(\frac{2l_l^2 - d_{b_k}^2}{2l_l^2} \right) - \alpha_r \right)^2 \\ &= \frac{1}{2}\sum_{k=1}^{n_l} \kappa \left(\cos\left(\frac{l_l^2 - 2\mathbf{q}^T\mathbf{X}_k^T\mathbf{X}_k\mathbf{q}}{l_l^2} \right) - \alpha_r \right)^2 \end{aligned} \tag{5.8}$$

5.2.2 Derivatives in the Lagrangian

In order to obtain the equations of motion we take the derivatives written in equation 5.4. From the gravitational potential energy equation of 5.5 we get

$$\frac{\partial V_g}{\partial \mathbf{q}} = -\mathbf{G}^T \tag{5.9}$$

The partial derivative of the potential energy in the strings looks like

$$\begin{aligned} \frac{\partial V_s}{\partial \mathbf{q}} &= \sum_{i=1}^{n_s} \sigma_k \mathbf{q}^T \mathbf{Y}_k^T \mathbf{Y}_k \\ &= \mathbf{q}^T \mathbf{Y}(\boldsymbol{\sigma} \otimes \mathbf{I}_{3n}) \end{aligned} \tag{5.10}$$

For the full derivation the reader is referred to [11].

The potential energy stored in the torsion springs is written in terms of α, but this should be converted to be written in terms of $\mathbf{q}$ so the spatial derivative can be taken. Using the chain rule the first step is

$$\begin{aligned} \frac{\partial V_l}{\partial \mathbf{q}} &= \frac{1}{2}\kappa \sum_{k=1}^{6} \frac{\partial}{\partial \mathbf{q}} (\alpha_k(q) - \alpha_r)^2 \\ &= \kappa \sum_{k=1}^{6} (\alpha_k(q) - \alpha_r) \frac{\partial \alpha_k(q)}{\partial \mathbf{q}} \end{aligned}$$

We can define α in terms of $\mathbf{q}$ or similarly in terms of $\mathbf{b_k}$.

$$\begin{aligned} \frac{\partial \alpha_k(\mathbf{q})}{\partial \mathbf{q}} &= \frac{\partial}{\partial \mathbf{q}} \left(cos^{-1} \left(\frac{l_l^2 - 2\mathbf{b_k^T b_k}}{l_l^2} \right) \right) \\ &= \frac{\partial}{\partial \mathbf{q}} (cos^{-1}(f_k)) \\ &= \frac{\partial f_k}{\partial \mathbf{q}} \left(-\frac{1}{\sqrt{(1 - f_k^2)}} \right) \end{aligned}$$

Here

$$\begin{aligned} f_k &= \left(\frac{l_l^2 - 2\mathbf{b_k^T b_k}}{l_l^2} \right) \\ \frac{\partial f_k}{\partial \mathbf{q}} &= -4\mathbf{b_k^T} \frac{\partial \mathbf{b_k}}{\partial \mathbf{q}} \end{aligned}$$

Then

$$\begin{aligned}\frac{\partial V_l}{\partial \mathbf{q}} &= \sum_{k=1}^{6} 4\kappa \left(\frac{cos^{-1}(f_k) - \alpha_r}{\sqrt{(1-f_k^2)}} \right) \mathbf{b}_k^T \frac{\partial \mathbf{b}_k}{\partial \mathbf{q}} \\ &= \sum_{k=1}^{6} 4\kappa \left(\frac{cos^{-1}(f_k) - \alpha_r}{\sqrt{(1-f_k^2)}} \right) \mathbf{b_k^T X_k} \\ &= \sum_{k=1}^{6} \gamma_k \mathbf{q^T X_k^T X_k} \\ &= \mathbf{q}^T \mathbf{X} (\boldsymbol{\gamma} \otimes \mathbf{I}_{3n})\end{aligned} \tag{5.11}$$

with $\mathbf{X} = [\mathbf{X}_1^T \mathbf{X}_1 \cdots \mathbf{X}_6^T \mathbf{X}_6]$ and $\boldsymbol{\gamma} = [\gamma_1 \cdots \gamma_6]$.
The kinetic energy of the k-th bar is written as

$$T_{\mathbf{b}_k} = \frac{1}{2} \left(m \dot{\mathbf{b}}_k^T \dot{\mathbf{b}}_k + \frac{I_{\mathbf{b}_k}}{d_{b_k}^4} (\mathbf{b}_k \times \dot{\mathbf{b}}_k) \cdot (\mathbf{b}_k \times \dot{\mathbf{b}}_k) \right)$$

Here the term with the cross products can be rewritten as

$$(\mathbf{b}_k \times \dot{\mathbf{b}}_k) \cdot (\mathbf{b}_k \times \dot{\mathbf{b}}_k) = d_{b_k}^2 (\dot{\mathbf{b}}_k^T \dot{\mathbf{b}}_k) - (d_{b_k} \dot{d}_{b_k})^2$$

Rewriting the terms with the cross products we get

$$T_{\mathbf{b}_k} = \frac{1}{2} \left(m \dot{\mathbf{b}}_k^T \dot{\mathbf{b}}_k + \frac{I_{\mathbf{b}_k}}{d_{b_k}^2} \dot{\mathbf{b}}_k \cdot \dot{\mathbf{b}}_k \right) - \frac{1}{2} \left(\frac{I_{b_k}}{d_{b_k}^2} \dot{d}_{b_k}^2 \right)$$

By using the equations from 5.1 this equations can be written in terms of q:

$$\begin{aligned} T_{\mathbf{b}_k} &= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}_{\mathbf{b}_k} \dot{\mathbf{q}} - T_{f_k}, \quad \text{with} \\ \mathbf{M}_{\mathbf{b}_k} &= m \bar{\mathbf{X}}_k^T \bar{\mathbf{X}}_k + \frac{I_{\mathbf{b}_k}}{d_{b_k}^2} \mathbf{X}_k^T \mathbf{X}_k, \quad \text{and} \\ T_{f_k} &= \frac{1}{2} \left(\frac{I_{b_k}}{d_{b_k}^2} \dot{d}_{b_k}^2 \right) \end{aligned} \tag{5.12}$$

Here $\mathbf{M}_{\mathbf{b}_k}$ is the mass matrix associated with one particular bar, so the kinetic energy of that same bar can be calculated. When all individual mass matrices are summed we get a mass matrix $\mathbf{M}$, we can calculate the total kinetic energy in the structure.

The derivative of the kinetic energy makes everything a little more complicated because the leg length is not constant, and the inertia of the geometry is variable.

$$\begin{aligned}\frac{\partial T}{\partial \dot{\mathbf{q}}} &= \dot{\mathbf{q}}^T \mathbf{M} - \frac{\partial T_f}{\partial \dot{\mathbf{q}}} \\ &= \dot{\mathbf{q}}^T \mathbf{M} - \frac{1}{2} \sum_{k=1}^{6} \frac{I_{b_k}}{d_{b_k}^2} (2 \dot{d}_{b_k} \frac{\partial \dot{d}_{b_k}}{\partial \dot{\mathbf{q}}}) \\ &= \dot{\mathbf{q}}^T \mathbf{M} - \mathbf{q}^T \sum_{k=1}^{6} \frac{I_{b_k}}{d_{b_k}^2} \frac{\mathbf{X}_k^T \mathbf{X}_k}{d_{b_k}} \dot{d}_{b_k} \\ &= \dot{\mathbf{q}}^T \mathbf{M} - \mathbf{q}^T \mathbf{M}_f\end{aligned}$$

Then to get the time derivative from the Euler Lagrange equation we take the time derivative of $\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right)$:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) = \ddot{\mathbf{q}}^T \mathbf{M}_{\ddot{\mathbf{q}}} + \dot{\mathbf{q}}^T \mathbf{M}_{\dot{q}} - \dot{\mathbf{q}}^T \mathbf{M}_{\mathbf{q}} \tag{5.13}$$

with

$$\begin{aligned}
\mathbf{M}_{\ddot{\mathbf{q}}} &= \mathbf{M} - \sum_{k=1}^{6} \frac{\mathbf{X}_k^T \mathbf{X}_k \mathbf{q}}{d_{\mathbf{b}_k}} \frac{I_{b_k} \mathbf{X}_k^T \mathbf{X}_k}{d_{\mathbf{b}_k}^3} \\
\mathbf{M}_{\dot{\mathbf{q}}} &= \dot{\mathbf{M}} - \mathbf{M}_f \\
\mathbf{M}_{\mathbf{q}} &= \sum_{k=1}^{6} \left[-\frac{dI_{\mathbf{b}_k}}{dt} \frac{\mathbf{X}_k^T \mathbf{X}_k \dot{l}_{\mathbf{b}_k}}{d_{\mathbf{b}_k}^3} + 3\left(\frac{I_{\mathbf{b}_f} \mathbf{X}_k^T \mathbf{X}_k}{d_{\mathbf{b}_k}^4}\right) \dot{d}_{\mathbf{b}_k}^2 \right. \\
&\quad \left. - \left(\frac{\dot{\mathbf{q}}^T \mathbf{X}_k^T \mathbf{X}_k \dot{\mathbf{q}}}{d_{\mathbf{b}_k}} \frac{\mathbf{q}^T \mathbf{X}_k^T \mathbf{X}_k \dot{\mathbf{q}}}{d_{\mathbf{b}_k}^2} \dot{d}_{\mathbf{b}_k} \right) \frac{I_{\mathbf{b}_k} \mathbf{X}_k^T \mathbf{X}_k}{d_{\mathbf{b}_k}^3} \right]
\end{aligned}$$

As before, the full derivation can be found in [11].

5.2.3 Equations of Motion

From all preceding derivatives of potential energies found in equations 5.9, 5.10, 5.11 and the equations for damping forces in equations 3.1 and 3.2 we can construct the equations of motion:

$$\begin{aligned}
\mathbf{M}_{\ddot{\mathbf{q}}} \ddot{\mathbf{q}} - \frac{\partial \mathbf{R}^T(\mathbf{q})}{\partial \mathbf{q}} \boldsymbol{\lambda} = &- \left[\mathbf{M}_{\mathbf{q}}{}^T + \mathbf{q}^T \mathbf{Y} (\boldsymbol{\sigma} \otimes \mathbf{I}_{3n}) \right. \\
&\left. + \mathbf{q}^T \mathbf{X} (\boldsymbol{\gamma} \otimes \mathbf{I}_{3n}) \right] \mathbf{q} - \mathbf{M}_{\dot{\mathbf{q}}}^{\mathbf{T}} \dot{\mathbf{q}} + \mathbf{G} + \mathbf{f} := \boldsymbol{\xi}_3
\end{aligned} \tag{5.14}$$

Through a cumbersome derivation of a linearized form a linear system can be obtained. First we define a parameter $\boldsymbol{\xi}_4$:

$$\boldsymbol{\xi}_4 = \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{q}} \partial \mathbf{q} + \frac{\partial \boldsymbol{\xi}_3}{\partial \dot{\mathbf{q}}} \partial \dot{\mathbf{q}} + \frac{\partial \boldsymbol{\xi}_3}{\partial \boldsymbol{\sigma}} \partial \boldsymbol{\sigma} + \frac{\partial \boldsymbol{\xi}_3}{\partial \boldsymbol{\gamma}} \partial \boldsymbol{\gamma} + \frac{\partial \boldsymbol{\xi}_3}{\partial \mathbf{f}} \partial \mathbf{f}$$

$$\begin{bmatrix} \delta \ddot{\mathbf{q}} \\ \delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{\ddot{\mathbf{q}}} & -\mathbf{R}_{\mathbf{q}}{}^T \\ -\mathbf{R}_{\mathbf{q}}{}^T & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\xi}_4 \\ \mathbf{0} \end{bmatrix} \tag{5.15}$$

with

$$\begin{aligned}
\mathbf{M}_\alpha &= \begin{bmatrix} \mathbf{M}_{\ddot{\mathbf{q}}} & -\mathbf{R}_{\mathbf{q}}{}^T \\ -\mathbf{R}_{\mathbf{q}}{}^T & \mathbf{0} \end{bmatrix}^{-1} \\
\mathbf{M}_\beta &= \mathbf{M}_\alpha(1:36, 1:36)
\end{aligned}$$

As stated before, the full derivation is available in [11]

5.2.4 Leg Length Constraints

In practice the end-nodes of legs of the tensegrity structure are constrained to be able to move a finite distance apart. The maximum separation is reached when the angle of the joint reaches 160 degrees. In the paper by Hsu et al., ideal constraints are used to ensure that bar lengths remain constant in a model with fixed bar lengths. In equation 5.4 the $\mathbf{R}_{\mathbf{q}}{}^T \lambda$ term constitute the leg length constraints. Thus it is described as $\mathbf{q}^T \mathbf{X}_k^T \mathbf{X}_k \mathbf{q} - l_l \sin(\alpha_k/2) = 0$ for a rigid bar.

A 30×1 constraints matrix is defined as

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} \mathbf{Aq} - \mathbf{b} \\ \mathbf{q}^T \mathbf{X}_1^T \mathbf{X}_1 \mathbf{q} - l_l \sin(\alpha_1/2) \\ \vdots \\ \mathbf{q}^T \mathbf{X}_6^T \mathbf{X}_6 \mathbf{q} - l_l \sin(\alpha_6/2) \end{bmatrix} = 0 \tag{5.16}$$

with $\mathbf{Aq} - \mathbf{b}$ being a 24 entry vector describing constraints on string length. The spatial derivative is then a 30×36 matrix:

$$\mathbf{R}_\mathbf{q} := \frac{\partial \mathbf{R}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{q}^T \mathbf{X}_1^T \mathbf{X}_1 \\ \vdots \\ \mathbf{q}^T \mathbf{X}_6^T \mathbf{X}_6 \end{bmatrix} \tag{5.17}$$

And if we take it further we get

$$\begin{bmatrix} \dot{\mathbf{q}}^T \frac{\partial^2 R_1}{\partial \mathbf{q}^2} \dot{\mathbf{q}} \\ \vdots \\ \dot{\mathbf{q}}^T \frac{\partial^2 R_{30}}{\partial \mathbf{q}^2} \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 2\dot{\mathbf{q}}^T \mathbf{X}_1^T \mathbf{X}_1 \dot{\mathbf{q}} \\ \vdots \\ 2\dot{\mathbf{q}}^T \mathbf{X}_6^T \mathbf{X}_6 \dot{\mathbf{q}} \end{bmatrix} \tag{5.18}$$

General Discussion

After the analysis of three different models there is quite a lot to unpack with regard to the usefulness of every model. Each model has its own upsides and downsides. While the form finding model is useful for parameter selection, a more lively representation of the motion of the structure can be seen in the kinematic model. If the right improvements are made to the E-L model, the accuracy of the model might improve a lot when comparing to the Simscape model.

It should be mentioned once again that the best validation tool would be to build a prototype. It is advisable to go with the parameter values of the form finding model, found in table 2.2. When the prototype is built the researcher might still want to see if changing some parameters improves on the performance in the way it does in the form finding model. The bar length is not as easily varied, so this should definitely be chosen at 30 centimeters total length. The embedded torsion spring is probably easy and cheap to replace, so trying multiple stiffness coefficient might be feasible for the prototype. Ordering different sets of elastic strings might also be a good way to go about it. Then the cable rest length can also be varied easily by changing the attachment point of the stretched string.

6.1 Continuation of E-L model

Because there is no working program for now, future continuation could be done after this project to complete the Euler-Lagrange description in Simulink. A number of adjustments could be made:

The moment of inertia of the legs has not been redefined as of now. The description of the moment of inertia is still based on the deformable bar example from Hsu [11], which makes it a simplification of the bending leg case of the jumping tensegrity robot. The straight bar example makes the simplification that the moment of inertia along the axis of the bar is zero, and thus the equation can be defined only in the directions of ω_2 and ω_3 (see 5.3). A bent bar will give us a different matrix for the moment of inertia, and this matrix is probably not diagonal anymore. It will make the calculation of the mass matrix a bit more difficult.

There are some difficulties occurring with the mass matrix, which may- or may not originate from the ill-defined moment of inertia. Roughly stated it means that the transfer of potential energy to kinetic energy is not going as it should, and in the future the definition of the mass-matrix should be looked into.

The leg length constraints of section 5.2.4 are not directly applicable to the bending leg case, since an inequality as in equation 5.16 is hard to realize. Instead one could use a steep potential to constrain the motion of leg-connected nodes, which is also used in an example for Euler-Lagrange mechanics in a book by Morin [15]. In the example a particle slides of a sphere, and the sphere has a very high elastic coefficient. The motion of the particle is thus constrained by a very steep potential, as the particle can only slightly indent the surface of the sphere. The sphere pushes back with a normal force which originates from the constraining potential. In the example the constraining force is found by taking the spatial derivative of the potential, and the equation for the normal force is found.

In the bent leg case, the rotation of the joint is locked at 160 degrees in its most stretched position. We have no reference of the elasticity of the bar sections, so for now we construct an equation for constraining potential $\mathbf{R_q}$. The main requirement for this equation is that it is zero at the point where the leg is stretched, i.e. the nodal distance is equal to d_2. We can then

use an exponential function like

$$\begin{aligned}\mathbf{R}(\mathbf{q}) =& e^{p\left(d_{b_k}-d_2\right)} \\ =& e^{p\left(\mathbf{b}_k^T\mathbf{b}_k-d_2\right)},\end{aligned}$$

where p is a scalar value which determines the steepness of the potential. If we also want to impose a minimum angle of the joint, then we can add another term which forms a potential barrier at nodal separation d_1.

$$\mathbf{R}(\mathbf{q}) = e^{-p\left(\mathbf{b}_k^T\mathbf{b}_k-d_1\right)} + e^{p\left(\mathbf{b}_k^T\mathbf{b}_k-d_2\right)} \tag{6.1}$$

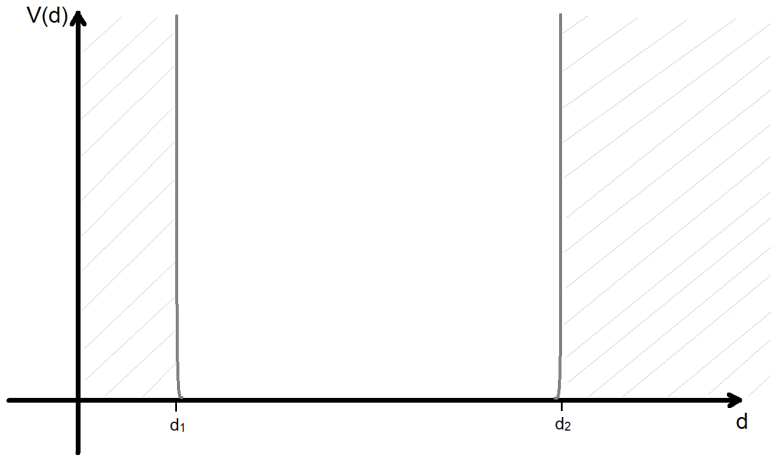


Figure 6.1: Potential well with the boundaries at d_1 and d_2.

Then we can also find $\frac{\partial \mathbf{R}}{\partial \mathbf{q}}$:

$$\begin{aligned}\frac{\partial \mathbf{R}}{\partial \mathbf{q}} &= \frac{\partial}{\partial \mathbf{q}}\left[e^{-p\left(\mathbf{b}_k^T\mathbf{b}_k-d_1\right)} + e^{p\left(\mathbf{b}_k^T\mathbf{b}_k-d_2\right)}\right] \\ &= \frac{\partial}{\partial \mathbf{q}}[-p(\mathbf{b}_k^T\mathbf{b}_k - d_1)]e^{-p\left(\mathbf{b}_k^T\mathbf{b}_k-d_1\right)} \\ &\quad + \frac{\partial}{\partial \mathbf{q}}[p(\mathbf{b}_k^T\mathbf{b}_k - d_2)]e^{p\left(\mathbf{b}_k^T\mathbf{b}_k-d_2\right)} \\ &= 2p\mathbf{q}^T\mathbf{X}_k^T\mathbf{X}_k\left[e^{-p\left(\mathbf{b}_k^T\mathbf{b}_k-d_1\right)} + e^{p\left(\mathbf{b}_k^T\mathbf{b}_k-d_2\right)}\right]\end{aligned}$$

This is the constraint equation that can be applied in the Lagrange multiplier term in equation 5.4. When this constraint was applied to the model, it did not seem to have an great effect on the behaviour of the simulation. In fact, the limit of 160° leg angle can still be exceeded. This indicates that there might be a mistake in the mass matrix $\mathbf{M}_{\ddot{\mathbf{q}}}$ of the equations of motion of 5.14. Where the mistake might be originating from is not clear at this point unfortunately.

6.2 Future Perspective

Unfortunately it was not possible to realize a physical prototype of any form, which will be paramount in validation of any of the models. Although the three Simulink models can be compared to one another and to the Simscape model, the only real validation can be achieved by building a prototype. If the theoretical work on this project is continued, I believe that the realization of a working Euler Lagrange model in Simulink will be very helpful. In that case a

more accurate mathematical analysis is available to simulate in, and this will reduce the work and testing on a prototype. Some of the steps that need to be taken are laid out in the previous section.

Figure 6.2: Rendering from Solidworks of two parallel legs, with a linear spring used at the joint of the legs.

The initial idea of the design of the joints included a torsion spring embedded within a joint. Thus, the force in the end nodes was determined by the angular offset from the rest angle of the spring. In a newer tentative design, a linear spring was introduced instead of a torsion spring as in figure 6.2. This would definitely make a difference in the way the energy stored in the springs is calculated. For now there was the angular version of Hooke's Law in the Simulink model. Placing a linear spring close to the joint would mean that the energy stored in the spring depends differently on the relative position of end nodes of the leg. Furthermore it would result in a lot more new analysis parameters to be looked into, such as the placement of the attachment point of the linear spring and the spring rest length. And the big question would of course be: would this make the robot jump more effectively?

With regard to the legs there are more complicated design which might yield more efficient jumping, but there is always the limitation of structure weight and it is most desirable to have a lightweight robot. An example of a different leg design would be by bio mimicry, as in the paper by Wang et al. [19]. In this paper the efficiency of a frog-like leg is described. The idea is that the transfer of forces is more efficient when using multiple joints, because the lever arms are shorter. The manner of actuation is also different, because a cable is wound up. This would make the entire structure a lot more complicated, but the jumping efficiency at equivalent input torque might be higher.

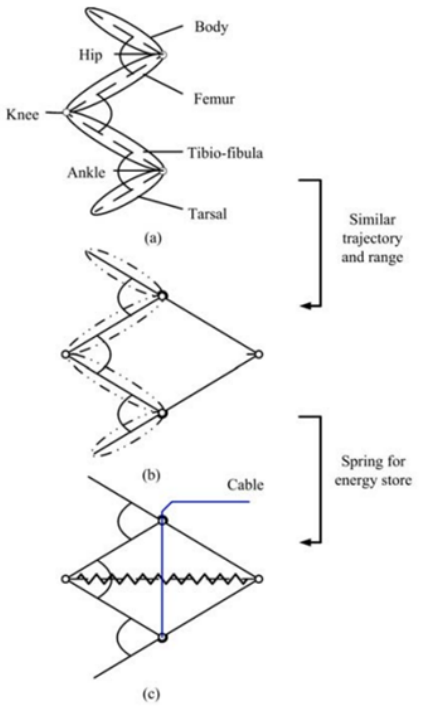


Figure 6.3: Schematic drawings of the mechanical frog leg from [19].

Another bio mimicry paper shows a design which is taken from a galago leg [6]. Since the galago has the highest power to weight ratio regarding jumping in the animal kingdom it seems like a good starting point. This time a series of rigid bar links that are connected to the actuating motor wind up the leg (see figure 6.4). This design might be adapted to fit in a tensegrity structure, but it greatly increases the complexity and cost of building a prototype, so the practicality of this design is in question.

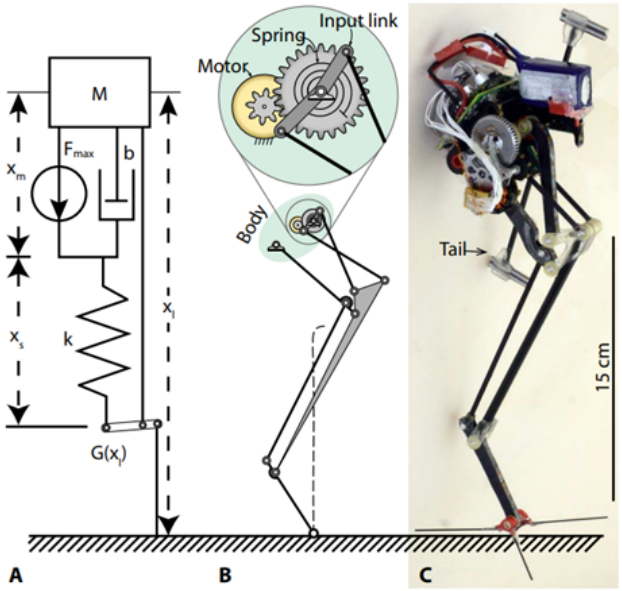


Figure 6.4: Leg design which is derived from the leg of a galago (image taken from [6])

A crawling mode of a leg-actuated type of tensegrity robot is probably an interesting outlook. The NASA SUPERball has already shown that a icosahedron tensegrity is able to crawl on irregular surfaces [7]. This robot makes use of cable contraction for the center of mas to shift and for the robot to roll. It might be feasible to create a mechanism which can release the compression of the bars more gradually than during jumping, and thus shifting the center of mass so that the robot will start to crawl. The concept of the mechanism that makes sure that both the jumping- and crawling mode can be accessed without human interference has to be thought up. Then it would be best for performance if the weight increase is not all too great. The crawling mode would be an upgrade because the repositioning jumping mode would not be necessary anymore. The crawling mechanism is probably more precise and the structure is more easily pointed in the direction that it should jump to.

If this crawling can be realised, there are other considerations to be made. In the current design the stiffness and length of each leg is equivalent, but this is not necessarily a design requirement. The design was chosen in this way because the symmetry promotes rolling of the structure after landing. When a crawling mode is realised as well, the robot can be more accurately positioned for when it needs to jump. The designer can then for instance choose to have torsion springs with different stiffness in different legs. There can then be two parallel legs with equal properties which are designated for jumping, and the other four legs have different properties than the jumping legs. Not only could the increase in performance be investigated, but it might also be wise to implement motors and gearboxes with lower mass on the four legs which are not used for jumping.

6.3 Final Conclusion

An art installation idea from the sixties of last century has grown out to be a more and more common topic in the multibody dynamics, control and robotics field. The popular icosahedron tensegrity already knows a range of applications in the public domain, but not one where the six bars are compliant and can be actuated for jumping. A set of models has been created in Simulink and Simscape, which are versatile tools within the Matlab environment.

The form finding model yielded quite a clear set of parameters which optimized performance of the robot. In the analysis of the kinematic model a wider range of possible 'good' parameters were found, which would give more flexibility to the user. However, the usefulness of the output data of the kinematic model is debatable.

The Euler-Lagrange description is a lot more accurate, but a lot more complex as well. For this research it was too time consuming to solve the problems that occurred during model building. In spite of that, quite some recommendations for future work on the E-L description could be given.

In a path to building the prototype, the analysis in this report is very useful in the selection of appropriate materials for the robot prototype. The constructed models are robust to alterations in the design requirements, as the program only needs adjustment in base parameters for the analysis to be re-done if needed.

Acknowledgements

I would like to dearly thank Zhiyuan Liu for providing a starting point in both the Simulink and the Simscape models. His work on previous versions has made my life easier and it has given me a great start to an otherwise very tough model building process. Furthermore he has advised me on the directions to take during the project which helped me a lot.

Bibliography

[1] Gunnar Tibert. *Deployable tensegrity structures for space applications*. 2002.

[2] Wojciech Gilewski, Joanna Kłosowska, and Paulina Obara. "Applications of tensegrity structures in civil engineering". In: *Procedia Engineering* 111 (2015), pp. 242–248.

[3] Christopher S. Chen and Donald E. Ingber. "Tensegrity and mechanoregulation: from skeleton to cytoskeleton". In: *Osteoarthritis and Cartilage* 7.1 (1999), pp. 81–94.

[4] Mirko Kovač et al. "Steerable miniature jumping robot". In: *Autonomous Robots* 28.3 (2010), pp. 295–306.

[5] Joel Burdick and Paolo Fiorini. "Minimalist jumping robots for celestial exploration". In: *The International Journal of Robotics Research* 22.7-8 (2003), pp. 653–674.

[6] Duncan W. Haldane et al. "Robotic vertical jumping agility via series-elastic power modulation". In: *Science Robotics* 1.1 (2016), eaag2048.

[7] Andrew P. Sabelhaus et al. "System design and locomotion of SUPERball, an untethered tensegrity robot". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2867–2873.

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

[9] Jiaming Zha et al. "A collision-resilient aerial vehicle with icosahedron tensegrity structure". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1407–1412.

[10] Stefano Mintchev et al. "A soft robot for random exploration of terrestrial environments". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7492–7497.

[11] Shao-Chen Hsu, Vaishnav Tadiparthi, and Raktim Bhattacharya. "A Lagrangian method for constrained dynamics in tensegrity systems with compressible bars". In: *Computational Mechanics* 67.1 (2021), pp. 139–165.

[12] Robert Connelly. "Tensegrities and global rigidity". In: Shaping space. Springer, 2013, pp. 267–278.

[13] Cornel Sultan, Martin Corless, and Robert E. Skelton. "The prestressability problem of tensegrity structures: some analytical solutions". In: *International Journal of Solids and Structures* 38.30-31 (2001), pp. 5223–5252.

[14] MATLAB. *version 9.9 (R2020b)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[15] David Morin. *Introduction to classical mechanics: with problems and solutions*. Cambridge University Press, 2008.

[16] Filipe Marques et al. "Modeling and analysis of friction including rolling effects in multibody dynamics: a review". In: *Multibody System Dynamics* 45 (2019), pp. 223–244.

[17] Peter J Blau. "The significance and use of the friction coefficient". In: *Tribology International* 34.9 (2001), pp. 585–591.

[18] Michael A Michaux, Aldo A Ferri, and Kenneth A Cunefare. “Effect of tangential dither signal on friction induced oscillations in an SDOF model”. In: (2007).

[19] Meng Wang et al. “Biological jumping mechanism analysis and modeling for frog robot”. In: *Journal of Bionic Engineering* 5.3 (2008), pp. 181–188.