



DEPARTMENT OF ENGINEERING CYBERNETICS

TTK4550 - SPECIALIZATION PROJECT

Design and Control of a Spring-actuated Jumping Quadruped in Earth Gravity

Author:

Johannes Ihle
Daniel Rosmæl Skauge

Supervisor:

Prof. Dr. Kostas Alexis

Co-supervisor:

Jørgen Anker Olsen

Date: 19th December 2024

Abstract

This specialization project presents the concept design of a small, lightweight jumping quadruped robot optimized for jumps in low gravity environments, which uses torsional springs mounted parallel to knee motors for energy storage and release. This work is motivated by the exploration of Martian lava tubes and other challenging extraterrestrial terrains where traditional wheeled rovers face mobility limitations. The robot is designed to be a platform for developing and testing Deep Reinforcement Learning (DRL) policies for jumping, aerial stabilization, and landing. The robot is designed to be lightweight and low-cost to reduce the risk and cost of damage during testing. Through Simulink Simscape simulations we demonstrate that pure motor actuation is insufficient for achieving significant jump heights, and that parallel torsional springs significantly increase the jump height. Grid search optimization reveals that equal-length leg segments maximize jump height, with maximum jump height of 1.18m in mars gravity and 0.5m in Earth gravity. The project delivers a detailed CAD model of the spring-actuated leg design, establishing the foundation for hardware implementation and control development in the subsequent master's thesis. While several challenges remain, including the lack of feedback control during jumps, the design shows promise for enabling agile locomotion in Martian lava tubes and other challenging extraterrestrial terrains.

Table of Contents

List of Figures	ii
List of Tables	ii
1 Introduction	2
1.1 Motivation	2
1.2 Scope	2
1.3 Related Work	2
2 Theory	4
2.1 BLDC Motor Model	4
2.2 Gear Transmission Friction Model	4
2.3 Spring Modelling	5
2.4 Kinematics	5
2.4.1 Robot Kinematics	5
2.4.2 Jacobian Matrix	6
2.4.3 Force/Torque Mapping	6
2.4.4 Inverse Kinematics	7
2.5 Contact Friction	7
2.6 Numerical Solvers	7
2.7 Linear Least Squares Regression	8
2.8 Mean Squared Error	8
2.9 Moving Average Filter	8
2.10 Centered Finite Differences	9
2.11 Inertia	9
2.11.1 Parallel Axis Theorem	9
2.11.2 Thin Rod	9
2.11.3 Disk	10
3 Modeling and Simulation	10
3.1 Simscape	10
3.2 Rigid Body Components of the Robot Model	10
3.3 Rigid Body Masses and Inertias	12
3.4 Elastic Components: Springs	15
3.5 Motor Modeling	16

3.6	Solver selection	16
4	Robot Design	18
4.1	The Single Vertical Manipulator Jump Model	18
4.2	Actuation Method Selection: Motors Only	19
4.3	Actuation Method Selection: Motors and Extension Spring	20
4.4	Actuation Method Selection: Motors and Torsional Spring	20
4.5	Hip Motor Strength Requirements	20
5	Motor Friction Estimation	21
5.1	Friction model	21
5.2	Pendulum Modeling	21
5.3	Data Acquisition	22
5.4	Linear Regression Derivation	22
6	Link Length Optimization	23
6.1	Impact of link lengths	23
6.2	Problem simplification	23
6.3	Initial Pose Calculation	24
6.4	Grid Search	26
7	Robot Hardware	26
7.1	Torsional Spring Leg	26
7.2	Extension Spring Leg Design	27
7.3	Motor Selection	28
7.4	Other Purchased Components	30
8	Results	30
8.1	Motor Friction Estimation	30
8.2	Link Length Optimization	31
8.3	Hip Motor Dimensioning Test	33
8.4	Motor Only Jumping Results	34
9	Discussion	35
9.1	Model Limitations and Uncertainties	35
9.1.1	Symmetric Jump Model Limitations	35
9.1.2	Unmodeled Dynamics	35
9.1.3	Motor Loading Uncertainty	35

9.2	Design Limitations	35
9.2.1	Landing Challenges	36
9.2.2	Hip Abduction/Adduction Design	36
9.3	Post-takeoff rotational velocity	36
9.3.1	Limited Jump Control	36
10	Future work	36
10.1	Jump Performance Analysis	36
10.2	Hardware Implementation	36
10.3	Control	37
11	Conclusion	37
Appendix		38
A	A06CLS V2 Website Information	38
B	A20BHM Website Information	38
C	A35CHM Motor Information	40
D	A80BHP-H Motor Information	40
E	Ultimaker Tough PLA Technical Data Sheet	42

List of Figures

1	Typical torque-speed characteristics of a Brushless DC (BLDC) motor [Microchip BLDC].	4
2	A 3-link planar robotic arm in \mathbb{R}^2	6
3	A typical Simscape block diagram.	11
4	A visualization of the model in figure 3.	11
5	Naming conventions for the parts of the robot.	13
6	Angle conventions for the robot body.	14
7	A figure displaying the conceptual difference between a parallel spring (right), as used in this work, and a series elastic spring (left) as used in the ANYmal robot [hutter anymal 2016].	15
8	A figure displaying the concept of a parallel extension spring, note that the spring generates torque directly in parallel with the knee joint just like for the parallel torsional spring seen in figure 7. Mathematically, the main difference is the non-linear relationship between knee angle and spring displacement, giving a non-linear torque-angle relationship.	16
9	Extension spring implementation in Simscape. As can be seen, the spring is connected between the appropriate points (output frames) of the thigh and shank spine, and in parallel to the knee motor model, which contains the knee joint.	17

10	Torque-speed characteristics for a given motor using our implemented motor model.	17
11	The manipulator corresponding to a vertical one leg jump.	18
12	Vertical Paw velocity as a function of knee angle.	19
13	Unadjusted paw placement in asymmetric configuration results in angled jump .	24
14	Adjusted paw placement in asymmetric configuration enables near vertical jump .	25
15	Paw placement adjustment for $L_2 > L_1$ configurations .	25
16	Comparison of normal and symmetric leg configurations .	26
17	Initial poses for different link length ratios .	26
18	Torsional spring leg CAD model.	27
19	Annotated exploded view of the CAD leg design.	27
20	Exploded view of the hip joint motor housings.	28
21	CAD model of components for in-house manufacture, with the motor shaft axle highlighted in red, which will lead directly into a ball bearing.	28
22	3D printed thigh prototype with torsional spring.	29
23	Extension spring configurations: outside (left) and inside (right) the leg.	29
24	Linear regression fit of the pendulum data for the knee motor. Derived theta dot dot is the double derivative of the pendulum angle.	31
25	Linear regression fit of the pendulum data for the hip motor.	31
26	Grid search results showing jump height performance across different link length configurations under Earth gravity.	32
27	Grid search results showing jump height performance across different link length configurations under Mars gravity.	33
28	Commanded and actual hip joint angle achieved during the hip motor strength test simulation.	33
29	Torque output of the hip motors during the hip motor strength test simulation.	34
30	Knee speed until takeoff with A80BHP-H motor.	34
31	A06CLS V2 Motor Information (Curt)	38
32	A35CHM Motor Information	40
33	A80BHP-H Motor Information	41
34	UM220509 Tough PLA TDS	42

List of Tables

2	Dimensions and density of the rigid bodies in the robot model. L1 and L2 have no set size, but are the variables we intend to optimize over.	12
3	Mass and inertia properties of the rigid bodies in the robot model. A list of the Eurepus robot's electronics can be found in [finn'tarek'master].	14
4	Masses and dimensions used in the main body mass calculation.	15

5	Selected Motors	29
6	Thigh mass comparison between simulation and actual design for different materials (calculated using Solidworks Mass Properties).	30
7	Additional purchased components.	30
8	Friction model parameters and Mean Squared Error (MSE) for both motors, comparing models with and without Coulomb friction.	30
9	Best performing link length configurations and their corresponding jump heights for Earth and Mars gravity.	32

Abbreviations

Abbreviation	Description
DRL	Deep Reinforcement Learning
BLDC	Brushless DC
DOF	Degrees of Freedom
CAD	Computer-Aided Design
CNC	Computer Numerical Control
MCU	Microcontroller Unit
MSE	Mean Squared Error
PWM	Pulse Width Modulation
I2C	Inter-Integrated Circuit
ADC	Analog-to-Digital Converter

1 Introduction

1.1 Motivation

Traditional wheeled rovers have successfully explored Mars, with six missions to date [**mars’rovers’x6**]. However, specialized robots like Axel [**Axel**] and Reachbot [**ReachBot**] are needed for more challenging terrain. One key target for exploration is Martian and Lunar lava tubes [**lavatubes**] - hollow caverns formed by ancient lava flows. These tubes interest scientists because they could shelter future missions from radiation and micrometeorites, maintain stable temperatures, and contain subsurface water ice [**lavatubes**].

Exploring lava tubes requires navigating rough terrain, sharp rocks, and steep slopes - challenges for wheeled rovers. Wheeled robots are also limited to ground movement and cannot take advantage of the lower gravity on Mars, the Moon, and asteroids. In contrast, jumping quadrupeds can utilize low gravity to potentially jump several meters high [**OLYMPUS2**], allowing them to cross obstacles that would stop wheeled rovers.

While quadruped robots have advanced significantly, they still struggle to jump effectively in Earth gravity. Testing hardware and control systems for low-gravity jumping is difficult since we cannot easily simulate low gravity on Earth. The high-speed impacts during jumping and landing also risk damaging expensive hardware. These challenges motivate our main goal: designing a small, lightweight, low-cost jumping quadruped robot. The low weight and cost reduce damage risk during testing and make the platform more accessible to researchers. We focus on achieving long jumps while maintaining the versatility of quadruped robots, such as walking on rough terrain, adjusting body pose, and carrying scientific equipment.

1.2 Scope

This report covers the design phase of a jumping quadruped robot as part of the TTK4550 Engineering Cybernetics Specialization Project at NTNU. While this project focuses on design, the work will continue in a master’s thesis that includes building, testing, and developing control algorithms for the robot.

The project scope includes:

- Creating a MATLAB/Simulink simulation to evaluate design choices
- Selecting an actuation method, either motors only, or a combination of motors and springs
- Choosing specific hardware components
- Developing a CAD model for one leg that:
 - Fits the chosen motors and springs
 - Can be manufactured using NTNU’s 3D printing and CNC facilities
 - Withstands jumping forces and impacts

1.3 Related Work

Several researchers have studied robotic jumping for Earth and low-gravity environments. NTNU’s Autonomous Robots Lab developed the Olympus robot [**OLYMPUS1**] [**OLYMPUS2**], which uses a spring-assisted 5-bar linkage leg for jumping. The robot weighs TODO kg and can jump TODO meters in Earth gravity, with testing done in simulated low gravity.

EPFL’s 600g RAVEN robot (Robotic Avian-inspired Vehicle for multiple ENvironments) [**RAVEN**] uses bird-inspired legs with two degrees of freedom. Similarly to our approach, it uses geared BLDC

motors to wind up embedded torsional springs for jumping. RAVEN can jump TODO cm in Earth gravity while also walking and hopping like a bird.

The 15g Grillo robot [**GRILLO**] demonstrates high-speed jumping, reaching takeoff velocities of 1.5 m/s (30 body lengths per second).

2 Theory

2.1 BLDC Motor Model

A brushless DC motor (BLDC) is a synchronous electric motor powered by direct current [PMSM BOOK]. The torque output of a BLDC motor decreases as its speed increases [Microchip BLDC]. This relationship is described by the motor's torque-speed curve, shown in Figure 1. The curve is defined by four key parameters: peak torque, maximum speed, rated speed, and rated torque. The rated torque is the maximum continuous torque output, while rated speed is the highest speed at which rated torque can be maintained. Above rated speed, torque output decreases. The motor can briefly exceed rated torque, for example during acceleration, within the limits of the torque-speed curve.

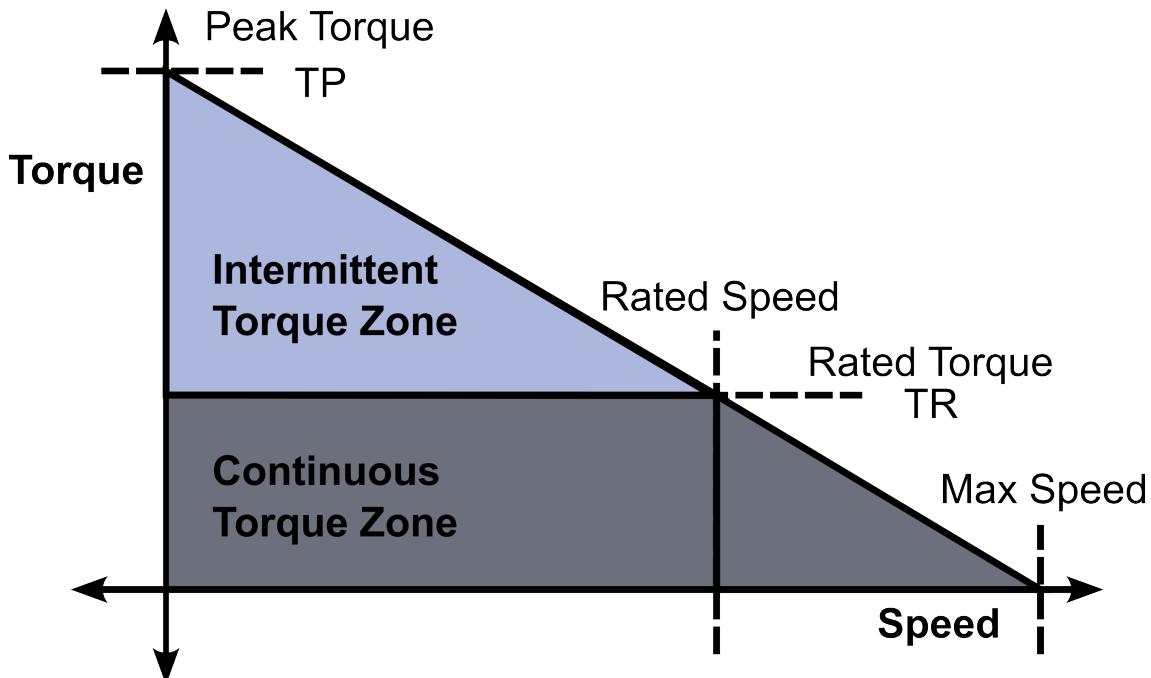


Figure 1: Typical torque-speed characteristics of a Brushless DC (BLDC) motor [Microchip BLDC].

When selecting a motor for an application, a 20% safety margin should be added to the required peak torque, after accounting for friction and other known factors [Microchip BLDC].

2.2 Gear Transmission Friction Model

Electric motors used in robotics often operate at speeds too high and torques too low for direct use. Gear transmissions are used to reduce speed and increase torque.

In an ideal geared transmission with no power loss, the output torque and velocity are given by equations 1 and 2, where N is the gear ratio, τ_{in} and τ_{out} are input and output torques, and w_{in} and w_{out} are input and output velocities [modern robotics book].

$$\tau_{out} = N\tau_{in} \quad (1)$$

$$w_{out} = \frac{w_{in}}{N} \quad (2)$$

Real transmissions experience power loss through friction. A common friction model combines viscous friction (proportional to velocity) and Coulomb friction (constant opposing force) [modern robotics book]. The total friction torque is the sum of these components, as shown in equation 3. Either term may be omitted depending on the application.

$$\tau_{friction} = b_{viscous}\dot{\theta} + b_{coulomb}\text{sign}(\dot{\theta}) \quad (3)$$

High gear ratios also increase the apparent rotor inertia. As shown in equation 4, apparent inertia increases with the square of the gear ratio [modern robotics book]. This can be problematic in robotic applications, particularly during contact, where high apparent inertia leads to stiff impacts [proprioceptive].

$$K = \frac{1}{2}I_{rotor}(G\dot{\theta})^2 = \frac{1}{2}I_{rotor}G^2(\dot{\theta})^2 = \frac{1}{2}I_{apparent}(\dot{\theta})^2 \quad (4)$$

2.3 Spring Modelling

Springs store potential energy through elastic deformation. The two main types relevant to this work are linear (extension/compression) springs and torsional springs.

Linear springs develop force proportional to displacement from their free length. For an ideal linear spring, this relationship follows Hooke's Law:

$$F = -kx \quad (5)$$

where F is the spring force, k is the spring constant, and x is displacement from equilibrium. The stored potential energy is:

$$U = \frac{1}{2}kx^2 \quad (6)$$

Torsional springs develop torque proportional to angular displacement from their free position:

$$\tau = -k\theta \quad (7)$$

where τ is the restoring torque, k is the torsional stiffness in Nm/rad, and θ is the angular displacement. The stored elastic potential energy is:

$$U = \frac{1}{2}k\theta^2 \quad (8)$$

Both spring types are used in mechanical systems for force/torque generation, shock absorption, and energy storage.

2.4 Kinematics

2.4.1 Robot Kinematics

Consider a planar robotic arm with n links, each with length l_i and joint angle θ_i . The end-effector position $\mathbf{x} = [x, y]^T$ in global coordinates is given by equation 9, derived from basic trigonometry. Figure 2 shows the coordinate system and joint angles.

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n l_i \cos\left(\sum_{j=1}^i \theta_j\right) \\ \sum_{i=1}^n l_i \sin\left(\sum_{j=1}^i \theta_j\right) \end{bmatrix} \quad (9)$$

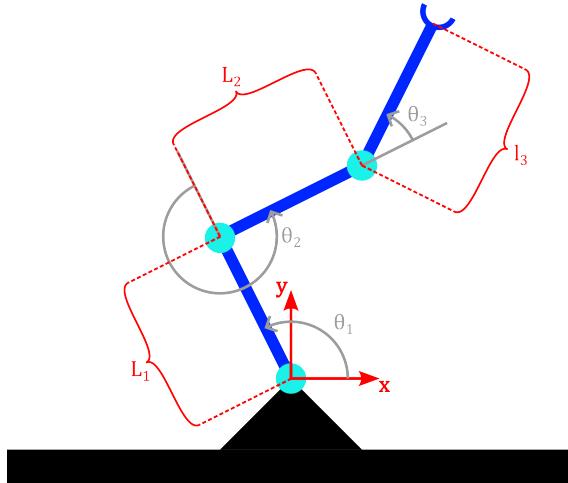


Figure 2: A 3-link planar robotic arm in \mathbb{R}^2 .

2.4.2 Jacobian Matrix

The Jacobian matrix relates small changes in joint angles to end-effector motion. As shown in equation 10, infinitesimal changes are described by partial derivatives [**modsim**]. The Jacobian matrix \mathbf{J} (equation 11) maps joint angle changes to end-effector position changes (equation 12). Using the chain rule on equation 9 yields equation 13, relating joint and end-effector velocities.

$$\delta y = \frac{\partial y}{\partial x} \delta x \quad (10)$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \dots & \frac{\partial y}{\partial q_n} \end{bmatrix} \quad (11)$$

$$\delta \mathbf{x} = \mathbf{J} \delta \mathbf{q} \quad (12)$$

$$\dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{q}} \quad (13)$$

2.4.3 Force/Torque Mapping

For a robotic manipulator with n joints (like figure 2), joint angles q_i and torques τ_i , the joint torques needed to support an end-effector force \mathbf{F} are given by equation 15 [**ASADA LECTURE NOTES**].

This relationship follows from the principle of virtual work. Consider virtual displacements δq_i of joint angles and $\delta \mathbf{x}$ of end-effector position. Virtual displacements need only satisfy geometric constraints, not dynamic laws [**ASADA LECTURE NOTES**].

For joint torques $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]$ and end-effector force $-\mathbf{F}$, the virtual work is:

$$\delta \mathbf{W} = \boldsymbol{\tau}^T \delta \mathbf{q} - \mathbf{F}^T \delta \mathbf{x} \quad (14)$$

At equilibrium, virtual work is zero for all valid displacements. Setting $\delta \mathbf{W} = 0$ and substituting $\delta \mathbf{x} = \mathbf{J} \delta \mathbf{q}$ from equation 12 gives:

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \quad (15)$$

2.4.4 Inverse Kinematics

Inverse kinematics calculates joint angles needed for a desired end-effector position. For a planar two-link manipulator with link lengths L_1 and L_2 , we can solve analytically for joint angles θ_1 and θ_2 given target position (x, y) .

The forward kinematics are described by the 2-link case of equation 9.

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \quad (16)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \quad (17)$$

Using the law of cosines:

$$\theta_2 = \pm \arccos \left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \right) \quad (18)$$

Then:

$$\theta_1 = \arctan 2(y, x) - \arctan 2(L_2 \sin(\theta_2), L_1 + L_2 \cos(\theta_2)) \quad (19)$$

The \pm in θ_2 indicates two possible solutions exist for most positions - one with elbow up and one with elbow down. When $L_1 = L_2$, the position $(0, 0)$ is a singularity with infinite solutions. At the workspace boundary where $x^2 + y^2 = (L_1 + L_2)^2$, only one solution exists.

2.5 Contact Friction

Contact friction is a force that resists relative motion between surfaces. The classical Coulomb friction model states that the friction force F_f is proportional to the normal force F_N :

$$F_f = \mu F_N \quad (20)$$

where μ is the coefficient of friction. This model distinguishes between static friction (surfaces at rest) and kinetic friction (surfaces in motion), with $\mu_s > \mu_k$. The friction force opposes motion and is limited by:

$$|F_f| \leq \mu_s F_N \text{ (static)} \quad (21)$$

When motion occurs, the friction force becomes:

$$F_f = -\mu_k F_N \text{sign}(v) \quad (22)$$

where v is the relative velocity. This discontinuity at $v = 0$ creates numerical challenges in simulation, often addressed by smoothing the transition between static and kinetic friction.

2.6 Numerical Solvers

Contact dynamics simulation requires specialized numerical solvers due to discontinuities and rapid force changes during impacts. These dynamics need stiff solvers to accurately model the fast transitions in forces and velocities [**stiff contact ODE 1**][**stiff contact ODE 2**]. Stiff solvers handle

problems with widely varying timescales, maintaining stability during contact events. Standard solvers can become unstable or fail to converge. MATLAB's `ode15s` and `ode23s` are examples of stiff solvers designed for these challenging differential equations [MATLAB'ODE].

2.7 Linear Least Squares Regression

Linear least squares regression is a method for finding the best-fitting line through a set of points by minimizing the sum of squared residuals. Given a set of observations (x_i, y_i) and a linear model $y = X\beta$, where X is the matrix of input variables and β contains the model parameters, the residual r_i for each observation is:

$$r_i = y_i - X_i\beta \quad (23)$$

The sum of squared residuals S is then:

$$S = \sum_{i=1}^n r_i^2 = (y - X\beta)^T(y - X\beta) \quad (24)$$

To minimize S , we take its derivative with respect to β and set it to zero:

$$\frac{\partial S}{\partial \beta} = -2X^T(y - X\beta) = 0 \quad (25)$$

Solving for β yields the normal equations:

$$X^T X \beta = X^T y \quad (26)$$

The solution is therefore:

$$\beta = (X^T X)^{-1} X^T y \quad (27)$$

This solution minimizes the sum of squared residuals and provides the optimal parameters β in the least squares sense.

2.8 Mean Squared Error

Mean Squared Error (MSE) measures the average squared difference between predicted values and actual values. For a set of n predictions \hat{y}_i and corresponding true values y_i , MSE is defined as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (28)$$

MSE penalizes larger errors more heavily due to the squared term, making it useful for evaluating model fit quality. A lower MSE indicates better model performance.

2.9 Moving Average Filter

A moving average filter smooths data by replacing each point with the average of a window of neighboring points. For a window size w , the filtered value y_i at index i is:

$$y_i = \frac{1}{w} \sum_{j=i-\lfloor w/2 \rfloor}^{i+\lfloor w/2 \rfloor} x_j \quad (29)$$

where x_j are the input values. The filter reduces high-frequency noise while preserving lower-frequency trends in the data. Larger window sizes provide more smoothing but can attenuate rapid changes in the signal.

2.10 Centered Finite Differences

Centered finite differences approximate derivatives using symmetric sampling around each point. For time series data with constant time step Δt , the first and second derivatives at index i are approximated as:

$$\dot{x}_i = \frac{x_{i+1} - x_{i-1}}{2\Delta t} \quad (30)$$

$$\ddot{x}_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2} \quad (31)$$

2.11 Inertia

The moment of inertia, I , quantifies the rotational inertia of an object, indicating how much torque is required for a desired angular acceleration.

2.11.1 Parallel Axis Theorem

The Parallel Axis Theorem allows the calculation of the moment of inertia of a body about any axis, given its moment of inertia about a parallel axis through its center of mass. It is mathematically expressed as:

$$I = I_{\text{cm}} + md^2 \quad (32)$$

where:

- I is the moment of inertia about the chosen axis,
- I_{cm} is the moment of inertia about the center of mass axis,
- m is the mass of the object,
- d is the distance between the two parallel axes.

2.11.2 Thin Rod

For a thin rod of length l and mass m , rotating about an axis perpendicular to the rod and passing through its end, the moment of inertia is derived using the Parallel Axis Theorem. The moment of inertia about the center of mass is:

$$I_{\text{cm, rod}} = \frac{1}{12}ml^2 \quad (33)$$

Applying the Parallel Axis Theorem with $d = \frac{l}{2}$:

$$I_{\text{rod}} = I_{\text{cm, rod}} + m \left(\frac{l}{2} \right)^2 = \frac{1}{12} ml^2 + \frac{1}{4} ml^2 = \frac{1}{3} ml^2 \quad (34)$$

2.11.3 Disk

For a solid disk of radius r and mass m , the moment of inertia about its center is:

$$I_{\text{cm, disk}} = \frac{1}{2} mr^2 \quad (35)$$

Using the Parallel Axis Theorem to find the moment of inertia about an axis parallel and a distance d away:

$$I_{\text{disk}} = I_{\text{cm, disk}} + md^2 = \frac{1}{2} mr^2 + md^2 \quad (36)$$

3 Modeling and Simulation

For the purpose of doing design verification and optimization, a simplified model of the robot was created. The model was created in Simscape, a physical modeling toolbox integrated with MATLAB/Simulink.

3.1 Simscape

Simscape is a simulation tool that allows you to rapidly create models of physical systems within Mathworks' MATLAB/Simulink environment. With Simscape, physical systems are built by interconnecting blocks representing physical components, such as rigid bodies, joints and springs in a block diagram. The blocks are parameterized by physical properties, such as mass, inertia, and damping. Simscape automatically generates the equations of motion for the system, which can be solved numerically to simulate the system's behavior. Like you can do with Simulink without Simscape, you can also add ordinary Simulink blocks, including Matlab Function blocks, to the model. Simscape is also compatible with Simulink's multiple numerical solvers, such as ode15s, ode45, and ode23s.

An example of a typical SimScape block diagram can be found in figure 3. A visualization of the corresponding model can be seen in figure 4. As one can see, each element in a block diagram is typically either a rigid body, or joints connecting the various rigid bodies. Since a given body has multiple possible locations that a joint could be connected to, as well as axes it can act on, blocks can export different frames, with different origins and orientations, depending on the desired position and orientation of the joint. For example, for a block representing the robotic equivalent of a thigh, natural output frames would be the ones with origins at the top and bottom of the thigh, with a select axis aligned with the desired knee or hip axis of rotation.

3.2 Rigid Body Components of the Robot Model

Since the purpose of the Simscape model is not to facilitate the development of a complicated full degree of freedom feedback controller, nor to optimize every small detail of the design, a simplified model was selected. This model consists of a main body with four legs, each of which with two degrees of freedom. A visualization of the model, as well as an overview of the body's naming conventions can be found in figure 5. An overview of the body's angle conventions can be found

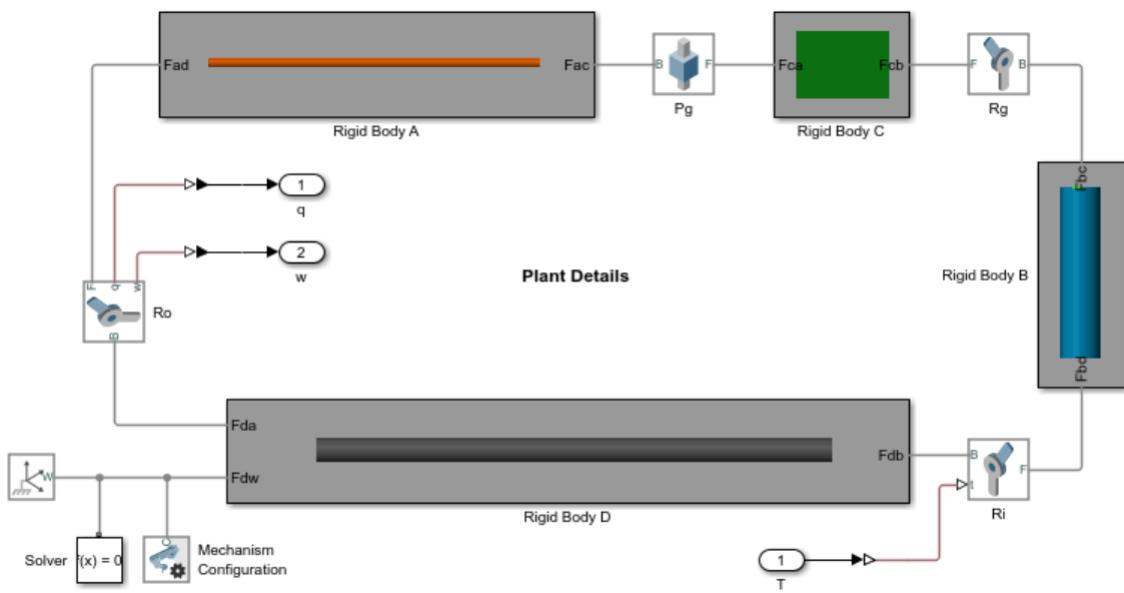


Figure 3: A typical Simscape block diagram.

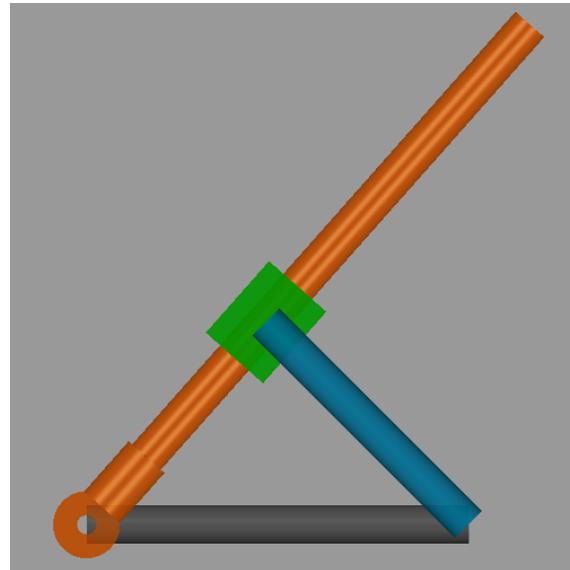


Figure 4: A visualization of the model in figure 3.

Component	Dimensions (cm)	Density (kg/m ³)	Mass (g)
Thigh	L1 x 0.5 x 0.25	2700	Calculated
Shank	L2 x 0.5 x 0.25	2700	Calculated
Hip Motor (A06CLS V2)	2 x 0.8 x 1.78	Calculated	7g
Knee Motor (A20BHM)	2.3 x 1.2 x 2.75	Calculated	20g
Knee Motor (A80BHP-H)	4.0 x 2.0 x 3.75	Calculated	79g
Thigh Spine	2 x 0.5 x 0.25	2700	Calculated
Shank Spine	1 x 0.5 x 0.25	2700	Calculated

Table 2: Dimensions and density of the rigid bodies in the robot model. L1 and L2 have no set size, but are the variables we intend to optimize over.

in figure 6. Note the absence of a hip abduction/adduction joint. This is because the model’s main purpose is to verify the design for jumping in the sagittal (forward-backward and upwards-downwards) plane, and the hip abduction/adduction joint is not necessary for this purpose.

Regarding the naming conventions presented in figure 5, note especially the naming of the different legs corresponding to location on the body, namely RH (Right Hind), RF (Right Front), LH (Left Hind), and LF (Left Front). Note also the naming of the joints hip (HIP) and knee (KNEE). If you see the angle conventions in figure 6, you can see that the angles of these joints correspond to the angles θ_1 and θ_2 respectively. Note that an orientation of zero degrees for the hip joint corresponds to the leg pointing straight downwards, and an orientation of zero degrees for the knee joint corresponds to the shank pointing in the same direction as the thigh. Note how each joint has its own coordinate system, with rotation being defined positive around the joints z axis. This is required for all Simscape Revolute joints. Note how the joints’ z axes correspond to body y axis. This is the case for all the robots legs, both the right and the left side legs.

As can be seen in both figure 5 and figure 6, in addition to the main body and legs colored in grey, the robot model also contains large purple blocks. These blocks represent motor masses, and their mass can be adjusted to represent different motors. In the current figures, the hip motors are A06CLS V2 motors, and the knee motors are A20BHM motors.

3.3 Rigid Body Masses and Inertias

In Simscape, the mass and inertia properties of a rigid body can be specified by the user or automatically calculated based on the body’s geometry and material properties. Hybrid solutions are also possible, where the user specifies some properties and Simscape calculates the rest [[simscape’tutorial](#)].

In the case of this model, a summary of the origin of the mass and inertia properties of the rigid bodies can be found in table 3. Exceptions are the properties of the main body and the paws, which will be specified in more detail in the two next paragraphs. For the parts whose mass and inertia are calculated by the geometry, a summary can be found in table 2.

The mass properties of the main body are based on the hardware components used by the Eurepus robot constructed by Maurer and El Agroudi [[finn’tarek’master](#)]. Since early in the design process a rough estimate was needed for the robot body mass, and it seemed likely that the electronic solution, apart from the motors, would be similar to the Eurepus robot, an approximate mass of the main body was calculated based on the mass of some of the Eurepus robot’s electronics plus an approximate amount of Nylon body material, four motors, and a random chosen microcontroller (MCU) mass. The formula used for the approximate of the main body mass can be found in equations 37 to 39. The masses corresponding to variables in equations 37 to 39 can be found in table 4. The motor mass chosen in table 4 corresponds to the mass of the AGF-RC A20BHM motor. This motor was chosen because, although the current plan is to use the A06CLS V2 motor, which is lighter, we would rather overestimate the mass of the main body than underestimate it, and a change of motor is one of the more likely changes to the robot design, as well as one of the changes that would increase the mass of the robot the most. TODO: Add method used for the inertia as

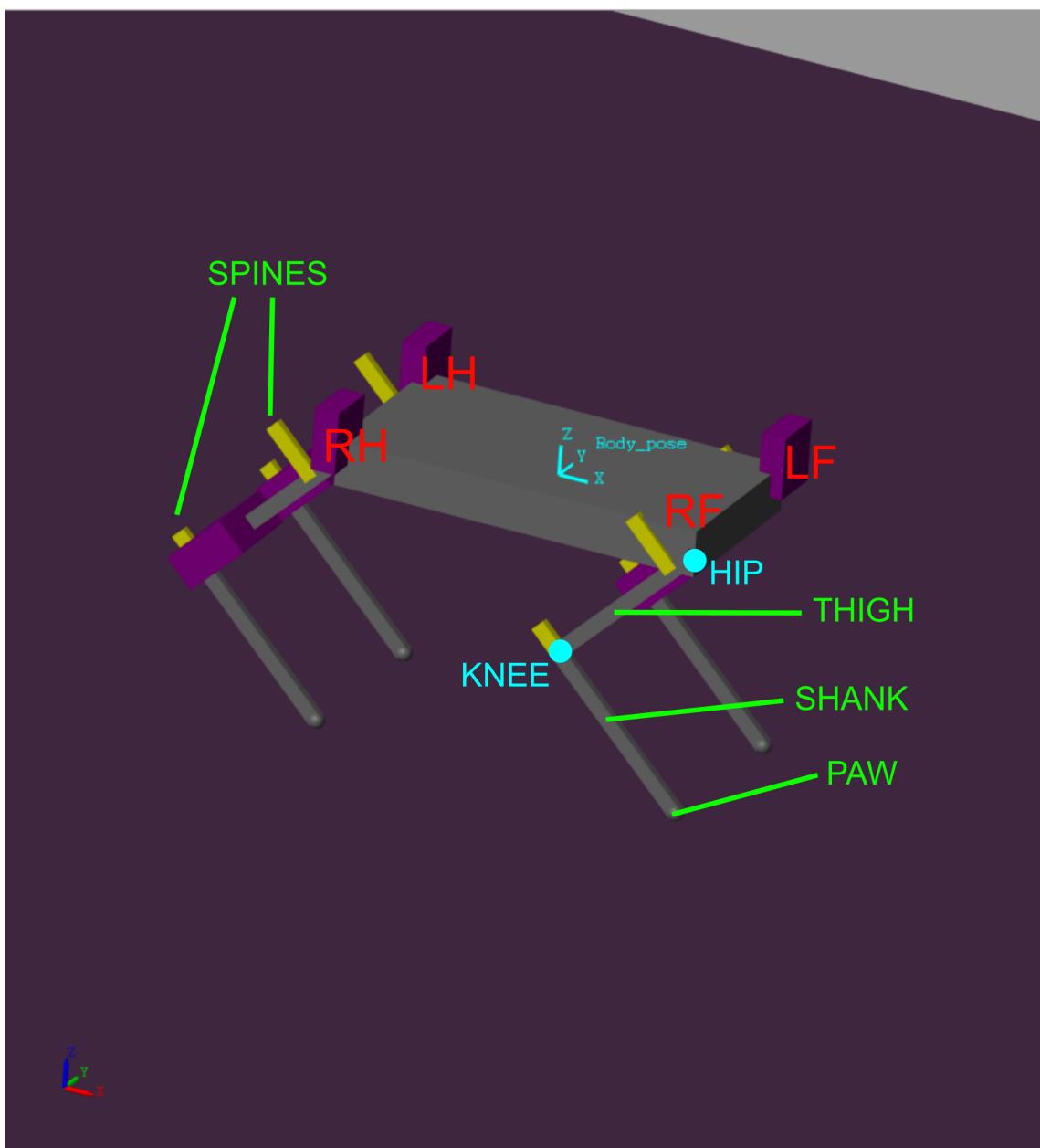


Figure 5: Naming conventions for the parts of the robot.

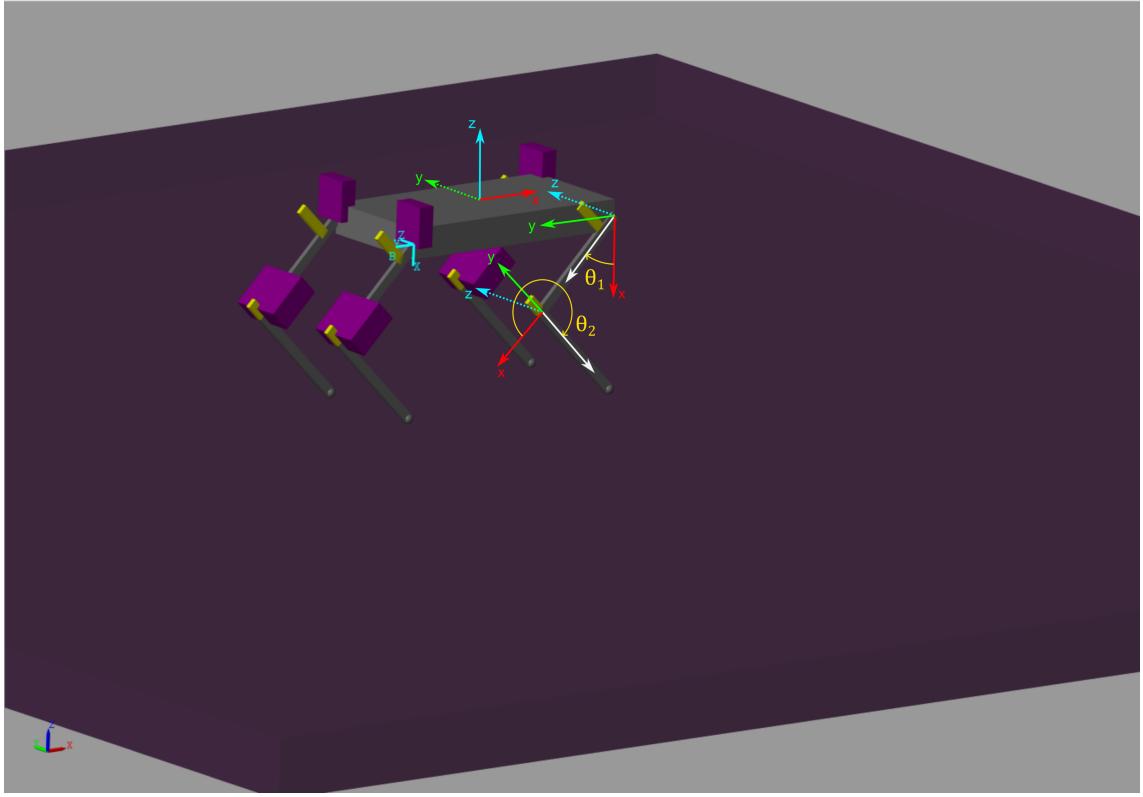


Figure 6: Angle conventions for the robot body.

well.

The geometrical, mass and inertia properties of the paws differ based on which of two scenarios we intend to simulate. The first scenario is the normal jumping scenario, in which the paw mass is simply set to 1000 kg/m^3 , though choosing the actual density of rubber would maybe be more appropriate. The dimensions of the paw are in this scenario simply chosen so that the diameter coincides exactly with $\max(\text{shank width}, \text{shank height})$. The second scenario is the one described in section 4.5, where the paw mass and volume is increased to estimate hip flexor/extensor motor requirements.

$$m_{plate} = \rho_{nylon} \cdot V_{plate} = \rho_{nylon} \cdot l_{plate} \cdot w_{plate} \cdot h_{plate} \quad (37)$$

$$m_{eurepus_electronics} = m_{battery} + m_{I2C} + 12 \cdot m_{ADC} + m_{PWM_driver} \quad (38)$$

Component	Mass	Density (kg/m^3)	Inertia	Geometry
Main Body	See section 3.3	See section 3.3	See section 3.3	Rectangular Prism
Thigh	From geometry	2700 (Aluminium 6061)	From geometry	Rectangular Prism
Shank	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Law	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Hip Motor	Actual motor mass	From Geometry	From Geometry	Rectangular Prism
Knee Motor	Actual motor mass	From Geometry	From Geometry	Rectangular Prism
Thigh Spine	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Shank Spine	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Paw	See section 3.3	See section 3.3	See section 3.3	Sphere

Table 3: Mass and inertia properties of the rigid bodies in the robot model. A list of the Eurepus robot's electronics can be found in [\[finn'tarek'master\]](#).

Variable	Description	Value
ρ_{nylon}	Density of Nylon	1520 kg/m ³
l_{plate}	Length of the plate	10 cm
w_{plate}	Width of the plate	6 cm
h_{plate}	Height of the plate	1.67 cm
$m_{battery}$	Mass of the battery	27 g
m_{motor}	Mass of one motor	20 g
m_{I2C}	Mass of the Inter-Integrated Circuit (I2C) module	5.1 g
m_{ADC}	Mass of one Analog-to-Digital Converter (ADC) module	2.4 g
m_{PWM_driver}	Mass of the Pulse Width Modulation (PWM) driver	8.5 g
m_{MCU}	Approximate mass of some microcontroller	30 g
m_{main_body}	Resultant mass of the main body	332 g

Table 4: Masses and dimensions used in the main body mass calculation.

$$m_{\text{main_body}} = m_{\text{plate}} + m_{\text{eurepus_electronics}} + m_{\text{MCU}} + 4 \cdot m_{\text{motor}} \quad (39)$$

3.4 Elastic Components: Springs

In addition to the model’s many rigid bodies, we also implemented two different forms of spring based passive actuation, namely:

- **A torsional spring** acting in parallel with the knee joint, as illustrated in figure 7. This spring is at zero extension when the knee joint is at zero degrees, and applies a torque that is proportional to the knee joint angle, as covered in section 2.3.
- **An extension spring** acting in parallel with the knee joint, attached to the shank and thigh spine, as illustrated in figure 8. The force generated by the extension spring is proportional to its displacement, as covered in section 2.3. The spring is intended to be unloaded when the knee joint is at zero degrees.

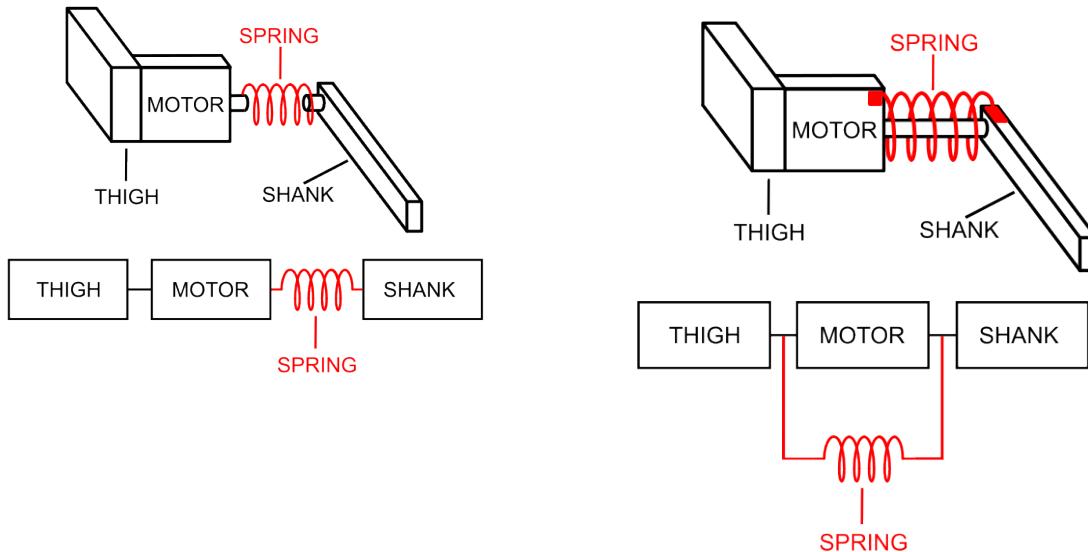


Figure 7: A figure displaying the conceptual difference between a parallel spring (right), as used in this work, and a series elastic spring (left) as used in the ANYmal robot [hutter'anymal'2016].

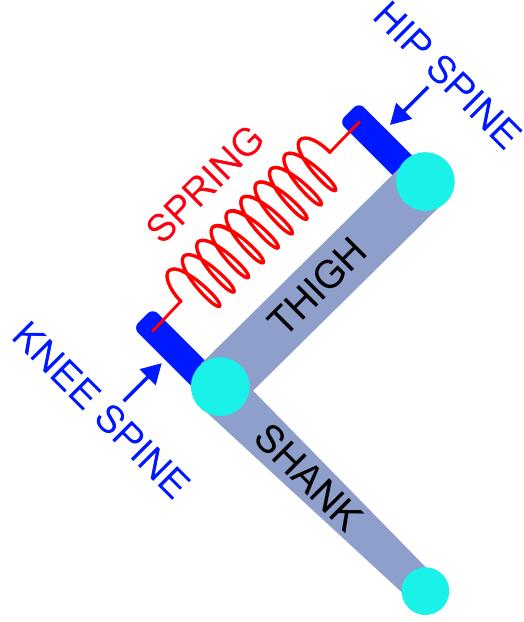


Figure 8: A figure displaying the concept of a parallel extension spring, note that the spring generates torque directly in parallel with the knee joint just like for the parallel torsional spring seen in figure 7. Mathematically, the main difference is the non-linear relationship between knee angle and spring displacement, giving a non-linear torque-angle relationship.

The torsion spring was implemented in the model using Simscape's prismatic joint option to add spring stiffness. The extension spring, on the other hand, was connected in parallel to the joint using Simscape's natural block diagram functionality, as seen in figure 9.

3.5 Motor Modeling

To be able to more accurately judge the jumping capability of different motors, a motor model in the form of a torque speed curve was implemented. The Brushless DC (BLDC) torque-speed curve presented in figure 1 is characterized by four parameters, namely the stall torque, the operating torque, the rated speed, and the maximum speed. Since motor suppliers contacted (primarily agf-rc and T-motor) were unable to provide most of the desired parameters, we chose a motor model with only two parameters, namely stall torque and maximum speed. The torque-speed model thus became a simple model where torque decreases linearly from stall torque, at speeds smaller than or equal to zero, to zero at speeds greater than or equal to maximum speed. The model is identical for negative velocities and torques, but with opposite signs. An example of the relation between desired torques and achieved torques for a given motor max speed can be seen in figure 10.

3.6 Solver selection

As described in section 2.3, the potential energy of a loaded spring can be easily calculated. Similarly, the potential energy due to gravity of a robot at the peak of a jump can also be determined. For the reasons discussed in section 2.6 we chose a stiff numerical solver, initially, we chose the `ode15s` solver. However, it was eventually observed that simulations using `ode15s` occasionally resulted in jump trajectories where the gravitational potential energy at the robot's peak height exceeded the combined potential energy of the four fully loaded springs by a factor of 2, even for jumps with only passive (spring) actuation. To address this inaccuracy, we experimented with different numerical solvers and ultimately selected the `ode23s` solver. This solver provided accurate simulations without artificially generating excess energy, and it performed efficiently.

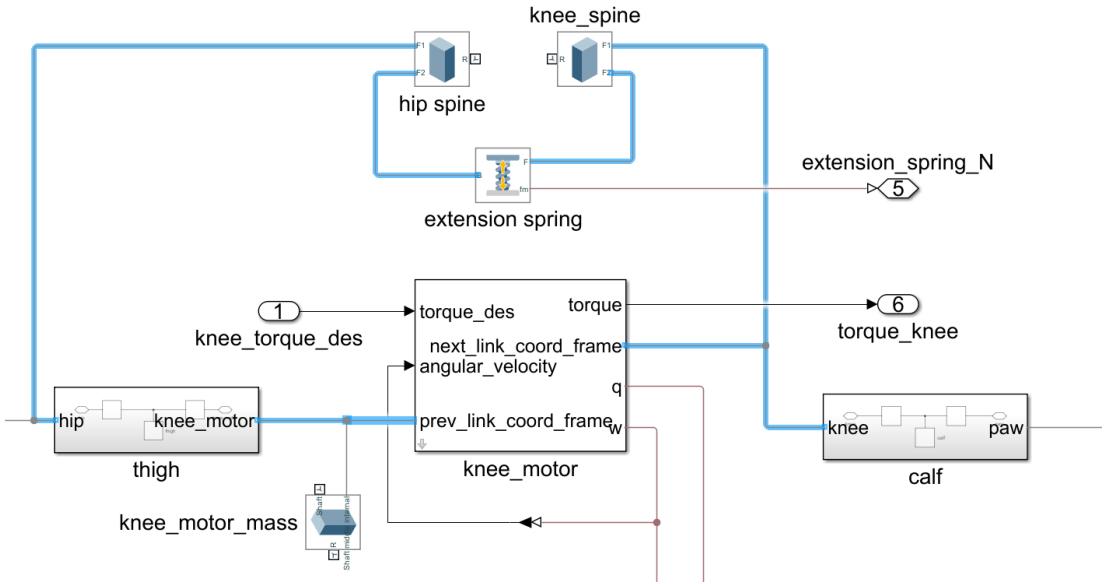


Figure 9: Extension spring implementation in Simscape. As can be seen, the spring is connected between the appropriate points (output frames) of the thigh and shank spine, and in parallel to the knee motor model, which contains the knee joint.

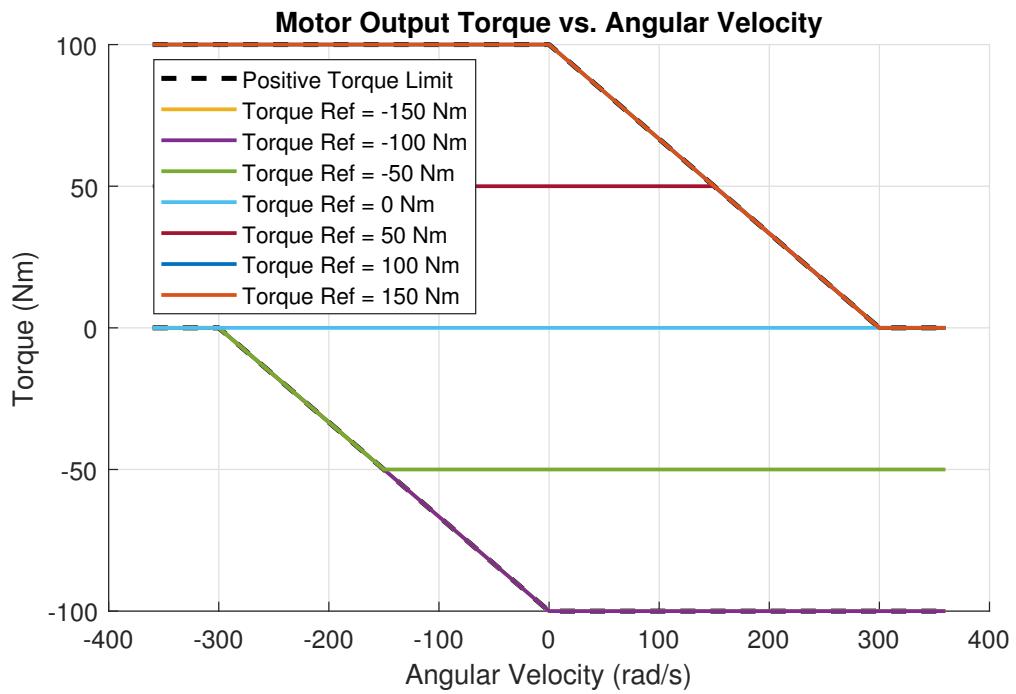


Figure 10: Torque-speed characteristics for a given motor using our implemented motor model.

4 Robot Design

4.1 The Single Vertical Manipulator Jump Model

For a jumping robot, takeoff velocity determines jump height when air resistance is negligible. Consider a simple case of maximizing vertical jump height for a single-leg robot with a 2-DOF leg of equal link lengths.

The paw position relative to the body follows standard 2-link manipulator kinematics (equation 40, section 2.4.1). For optimal vertical jumping, the body center of mass and paw move in opposite directions along the y-axis, as shown in figure 11. This constraint requires $\theta_1 = -\frac{(\theta_2 + \pi)}{2}$ and $\dot{\theta}_2 = -2\dot{\theta}_1$. Combining these with the end-effector Jacobian (equation 41), we can plot paw vertical velocity versus knee angle θ_2 (figure 12), using equation 13. TODO: does this make sense? body velocity as a function of theta2? Should the relationship between angular and linear velocity be explained instead?

Figure 12 shows that joint velocity translates more effectively to body velocity when the knee is crouched. This indicates joint acceleration capability is as important as maximum speed when selecting motors - a faster motor provides little benefit unless it can also accelerate the joints more quickly.

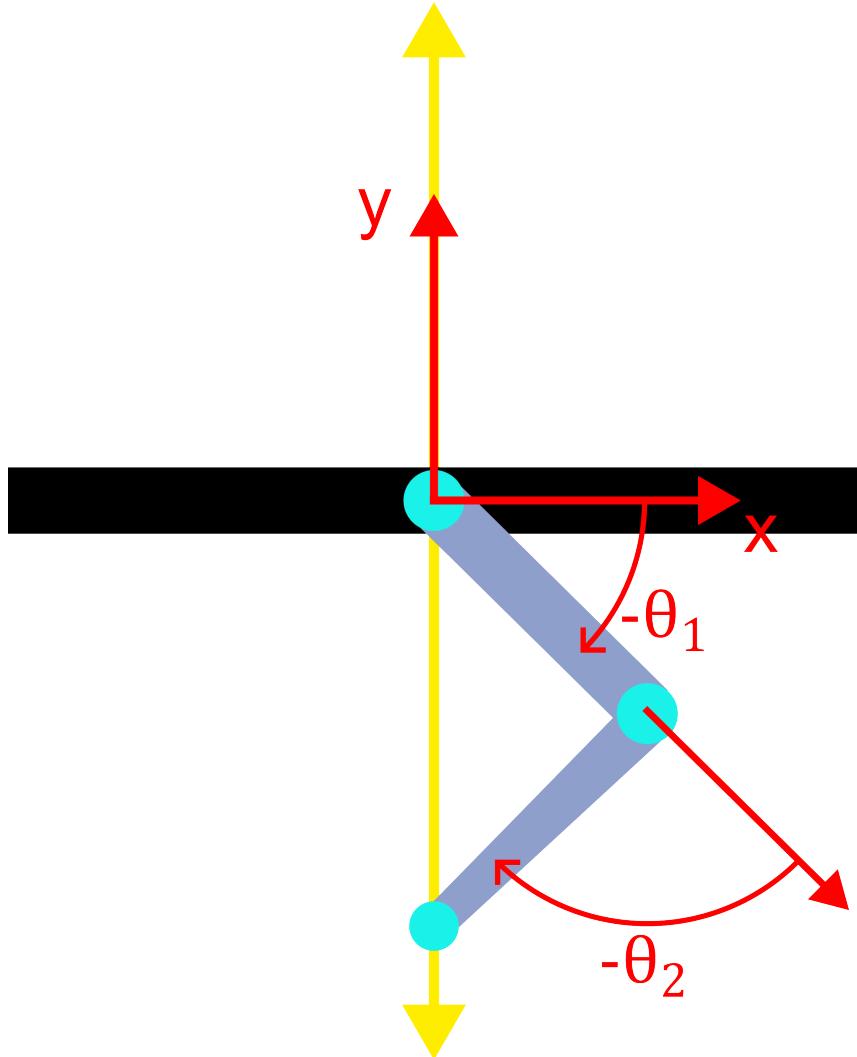


Figure 11: The manipulator corresponding to a vertical one leg jump.

$$\begin{aligned}x_{\text{end}} &= L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\y_{\text{end}} &= L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)\end{aligned}\quad (40)$$

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (41)$$

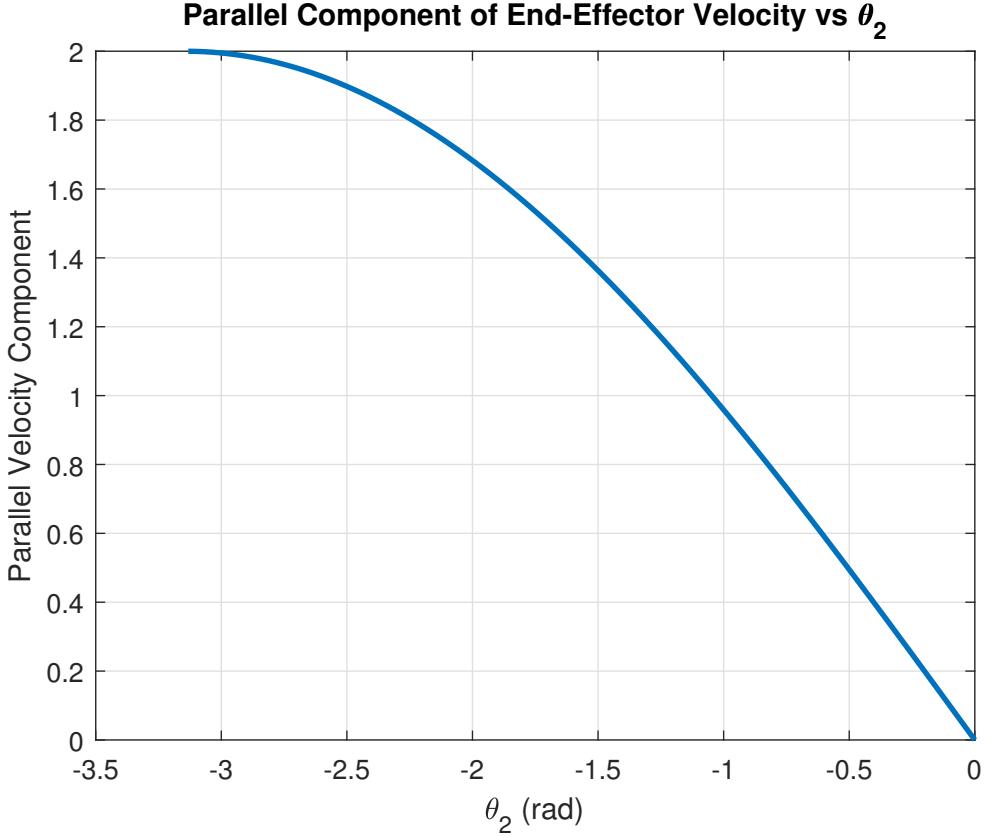


Figure 12: Vertical Paw velocity as a function of knee angle.

4.2 Actuation Method Selection: Motors Only

Initial experiments used motors alone. AGF-RC motors were selected for their high torque-to-weight ratio. The motor model used stall torque and maximum velocity parameters with linear torque decrease between them, as described in section 3.5. AGF-RC could not provide detailed performance data beyond these basic specifications. Motor-only actuation jumping performance was not satisfactory, thus the approach was abandoned. WHAT DOES THIS MEAN: "a motor model consisting of a stall torque and a max velocity, with a linear decrease in max torque between these two, was chosen."

The A80BHP-H motor from AGF-RC had the highest stall torque and operating speed in its weight class. No comparable alternatives were found, making tests with this motor an optimistic baseline. Details are in appendix D.

Tests applied maximum torque to knee motors in adherence with their torque-speed curve, while hip joints followed a control law (equation 45) to prevent slipping. This control uses the force-torque mapping from section 2.4.3 and friction theory from section 2.5. Though equation 15 assumes the system is in equilibrium, which is not valid for this dynamic jumping scenario, minimal slipping occurred in practice.

TODO: WHAT DOES THIS MEAN? Equation 15 is derived with an assumption of zero velocity/equilibria (TODO), an assumption that is obviously not valid for this dynamic jumping scenario. Despite this, very little slipping is observed in practice.

$$N = \text{Normal force} \quad (42)$$

$$\mu = \text{friction coefficient} = 0.8 \quad (43)$$

$$\tau_{\text{friction cone limit}} = J^T \begin{bmatrix} N\mu \\ 0 \end{bmatrix} \quad (44)$$

$$\max(|\tau_{knee}|) = \tau_{\text{friction cone limit}} \quad (45)$$

No leg configuration achieved satisfactory jumping height (defined as center of mass clearing ground by twice leg length). Results are detailed in section 8.4.

Later spring-motor experiments used weaker motors with more realistic friction models. The A80BHP-H was not used due to initially reported 90-degree travel limits. Later communication with AGF-RC revealed no such limitation existed. Potential A80BHP-H use with springs is discussed in section 10.

4.3 Actuation Method Selection: Motors and Extension Spring

The extension spring design (figure 23) was abandoned due to geometric constraints detailed in section 7.2. Extension springs require attachment points at both ends, adding four design parameters that significantly affect jumping performance. The spring torque varies non-monotonically with knee angle θ_2 , and optimal attachment points depend heavily on link lengths. Including these parameters in the link length optimization (section 6) would substantially increase complexity.

4.4 Actuation Method Selection: Motors and Torsional Spring

The final design combines motors with torsional springs in the knees (figure 18). Motors compress the knee springs until reaching a target angle. When released, the springs accelerate the joints for takeoff. Despite springs working against the friction of the motors as modelled in 5, this achieved better jumping performance than motors alone, as shown in section 8.2.

In contrast to extension springs, since the spring force from a torsional spring depends only on its angular displacement, it provides maximum torque at full knee flexion, most efficiently converting angular velocity into linear velocity at the paws.

4.5 Hip Motor Strength Requirements

Successful jumping inherently requires aerial attitude stabilization for proper landing. The use of torsional knee springs for jumping reduces the hip motors' role in takeoff, allowing them to be optimized for aerial control. To validate that the hip motors can provide sufficient control authority for landing orientation, we adopted the heuristic from [finn'tarek'master] of executing three 90-degree lateral swings per second. This benchmark provides a conservative bound on the torque and speed requirements needed to reorient the robot during flight.

We simulated this benchmark in Simulink Simscape TODO: citation???, with the robot body fixed in space and 1cm diameter iron spheres (32g) as paws. While lighter than the 80g paws in [finn'tarek'master], this scales appropriately with our 300g body mass compared to their 800g. The simulation results are presented in section 8.3.

5 Motor Friction Estimation

This section details the estimation of parameters for the motor friction models of the hip and knee motors. The motors serve as pivot points for pendulums consisting of aluminum rods attached to the motor shafts with ballast masses. The pendulums are released from a horizontal position, and their angular velocity and acceleration are measured. These measurements facilitate the estimation of motor friction coefficients using linear regression. The experimental setup for both motors is depicted in Figure ??.

5.1 Friction model

We adopt a linear friction model as described in section 2.2, incorporating both viscous and Coulomb friction, to accurately fit the observed data. To avoid numerical solver issues from the discontinuity at zero velocity introduced by the Coulomb friction, we used a tanh function to smooth the transition between positive and negative Coulomb friction, expressed as:

$$\tau_{\text{friction}} = b_v \dot{\theta} + b_c \tanh(\dot{\theta}/k) \quad (46)$$

where $k = 0.001$ controls the smoothness of the transitions, maintaining realism while avoiding numerical solver issues.

5.2 Pendulum Modeling

The pendulum used in the motor friction tests consists of an aluminum rod of length $l_{\text{arm}} = 0.21$ meters and a ballast mass $m_{\text{ballast}} = 0.301$ kg with radius $r_{\text{ballast}} = 0.03$ meters attached at a distance $d_{\text{ballast}} = 0.20$ meters from the pivot. The total mass of the arm is $m_{\text{arm}} = 0.034$ kg. The pendulum is modeled as a rigid body rotating about the motor shaft with a total moment of inertia I composed of the arm inertia I_{arm} and ballast inertia I_{ballast} . The arm is approximated as a thin rod, and the ballast as a disk, both with uniform density. Their inertias are calculated using the parallel axis theorem:

$$I_{\text{arm}} = \frac{1}{12} m_{\text{arm}} l_{\text{arm}}^2 + m_{\text{arm}} \left(\frac{l_{\text{arm}}}{2} \right)^2$$

$$I_{\text{ballast}} = \frac{1}{2} m_{\text{ballast}} r_{\text{ballast}}^2 + m_{\text{ballast}} d_{\text{ballast}}^2$$

$$I = I_{\text{arm}} + I_{\text{ballast}}$$

The torque due to gravity consists of contributions from both the arm and ballast mass:

$$\tau_{\text{gravity,arm}} = m_{\text{arm}} \frac{l_{\text{arm}}}{2} g \sin(\theta)$$

$$\tau_{\text{gravity,ballast}} = m_{\text{ballast}} d_{\text{ballast}} g \sin(\theta)$$

$$\tau_{\text{gravity}} = \tau_{\text{gravity,arm}} + \tau_{\text{gravity,ballast}}$$

The equations of motion, accounting for both viscous and Coulomb friction, are:

$$I\ddot{\theta} + b_v\dot{\theta} + b_c \text{sign}(\dot{\theta}) + \tau_{\text{gravity}} = 0$$

where:

- θ is the angular displacement (positive counterclockwise, zero at vertical down position)
- $\dot{\theta}$ and $\ddot{\theta}$ are the angular velocity and acceleration, respectively
- b_v is the viscous damping coefficient
- b_c is the Coulomb damping coefficient
- $g = 9.81 \text{ m/s}^2$ is the acceleration due to gravity

5.3 Data Acquisition

Pendulum angles θ were sampled by manually annotating video frames at 60 Hz using the Tracker program [**tracker**].

The angular velocity $\dot{\theta}$ and acceleration $\ddot{\theta}$ are computed using centered finite differences:

$$\dot{\theta}_i = \frac{\theta_{i+1} - \theta_{i-1}}{2\Delta t}$$

$$\ddot{\theta}_i = \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{(\Delta t)^2}$$

where Δt is the time step between measurements.

The angular position θ was smoothed using a moving average filter with a window size of 5 before computing $\dot{\theta}$. Both $\dot{\theta}$ and $\ddot{\theta}$ were smoothed using a moving average filter with a window size of 3.

5.4 Linear Regression Derivation

Rearranging the equations of motion for linear regression:

$$I\ddot{\theta} + \tau_{\text{gravity}} = -b_v\dot{\theta} - b_c \text{sign}(\dot{\theta})$$

This can be expressed in matrix form:

$$Y = X\beta$$

where:

- $Y = -I\ddot{\theta} - \tau_{\text{gravity}}$,
- $X = [\dot{\theta} \quad \text{sign}(\dot{\theta})]$,
- $\beta = \begin{bmatrix} b_v \\ b_c \end{bmatrix}$.

The linear least squares solution for β is:

$$\beta = (X^T X)^{-1} X^T Y$$

This yields both the viscous damping coefficient b_v and the Coulomb damping coefficient b_c .

To validate the model we forward simulate the pendulum motion using the estimated parameters with a runge-kutta solver and compare the results with the measured data. This and the results are shown in section 8.1.

6 Link Length Optimization

This section details the optimization of the robot's link lengths L1 and L2 to maximize jumping performance. The optimization ensures the robot can achieve sufficient jumping performance to clearly demonstrate future DRL control policies. A grid search approach using the Simscape robot model from section 3 systematically explores different link length configurations.

6.1 Impact of link lengths

Due to the lack of feedback control during the jumping maneuver, the jumping performance is determined fully by the link lengths and initial pose. By initial pose we mean the angles theta1 and theta2 of the knee and hip joints as the knee motor turns off and lets the loaded knee spring actuate the leg. Link lengths constrain the possible initial poses for any given jumping angle, while the initial pose determines spring compression through the knee angle theta2 and thus available potential spring energy. Additionally, link lengths affect the center of mass trajectory during jumps, influencing how gravitational and ground contact forces impact the robot's movement.

6.2 Problem simplification

First, we define the jump angle θ_J as the angle of the velocity vector of the center of mass of the robot body at the moment the robot leaves the ground. While the robot should be able to jump both vertically and at angles sufficient to overcome obstacles, optimizing for angled jumps presents challenges. We want to directly compare the jumping performance of different link lengths, but it is not obvious how to place the initial pose of the robot to achieve a θ_J across different link lengths.

To simplify the optimization, only the vertical jumping performance was considered, so that different link lengths can be compared directly. Vertical jumps are achieved across link lengths by flipping the front legs such that the legs are symmetric about the vertical axis, as shown in figure 16. This flipping transforms the asymmetric leg configuration in figure 16a where the front legs point forward into a symmetric configuration where both legs point backward. In this configuration, the movement of the legs during the jump is symmetric and the robot center of mass remains in the horizontal center of the robot, such that any horizontal component of the jump is canceled out. In practice, the robot will use the asymmetric configuration. The symmetric configuration simply provides an approximation for the jumping performance of a given link length configuration.

The metric used to evaluate the jumping performance is the maximum height reached by the center of the robot torso, minus the maximum standing height reached by the robot torso when the legs are fully extended vertically downwards and the paws are in contact with the ground. This metric will hereby be referred to as the "effective jump height".

The asymmetric leg configuration can also achieve vertical jumps by adjusting the angle between the hip-to-paw vector and vertical, as shown in figure ???. However, finding or calculating the optimal offset for arbitrary link lengths is complex. We therefore simplify by placing paws directly

beneath hips, except when $L_2 > L_1$. While this produces less realistic jump heights, it greatly simplifies the optimization.

For $L_2 > L_1$ configurations, placing paws directly under hips results in near-vertical legs that slip during jumps. Slipping does not occur to the same extent for asymmetric configurations with the same link lengths where the hip-to-paw angle is adjusted to give a vertical jump. In order to make the symmetric configuration a better approximation of the asymmetric case, we wish to avoid the slipping behavior. We do so by adjusting the hip-to-paw angle such that paws are slightly inward towards the body by a constant offset of 0.3 radians. This increases the vertical component of the paw force and thus the friction force (figure 15). This exact offset was chosen because experimentation showed that this makes asymmetric configuration jumps generally more vertical across a range of link lengths. This adjusted paw placement better approximates the practical asymmetric case that requires similar adjustment for vertical jumps, although the exact angle offset will vary depending on the link lengths. To handle the increased opposing horizontal forces between paws in the symmetric configuration, we double the friction coefficients from 1.0/0.8 to 2.0/1.6 (static/kinetic), which further reduces slipping.

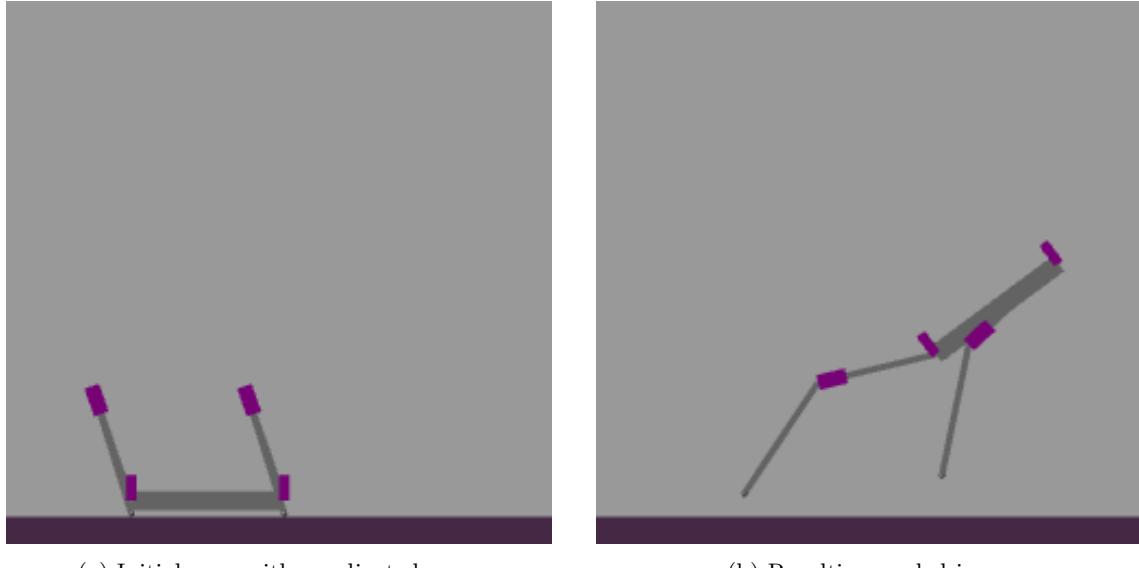


Figure 13: Unadjusted paw placement in asymmetric configuration results in angled jump

6.3 Initial Pose Calculation

For each set of link lengths, an initial pose must be calculated that satisfies several constraints:

- The paws must maintain ground contact
- Knee angle theta2 must be maximized to store maximum spring potential energy
- Knees cannot penetrate the ground
- Both knees bend outward rather than inward

The pose calculation considers three cases based on link length ratios:

1. $L_1 = L_2$ (equal lengths)
2. $L_2 < L_1$ (longer lower link)
3. $L_1 > L_2$ (longer upper link)

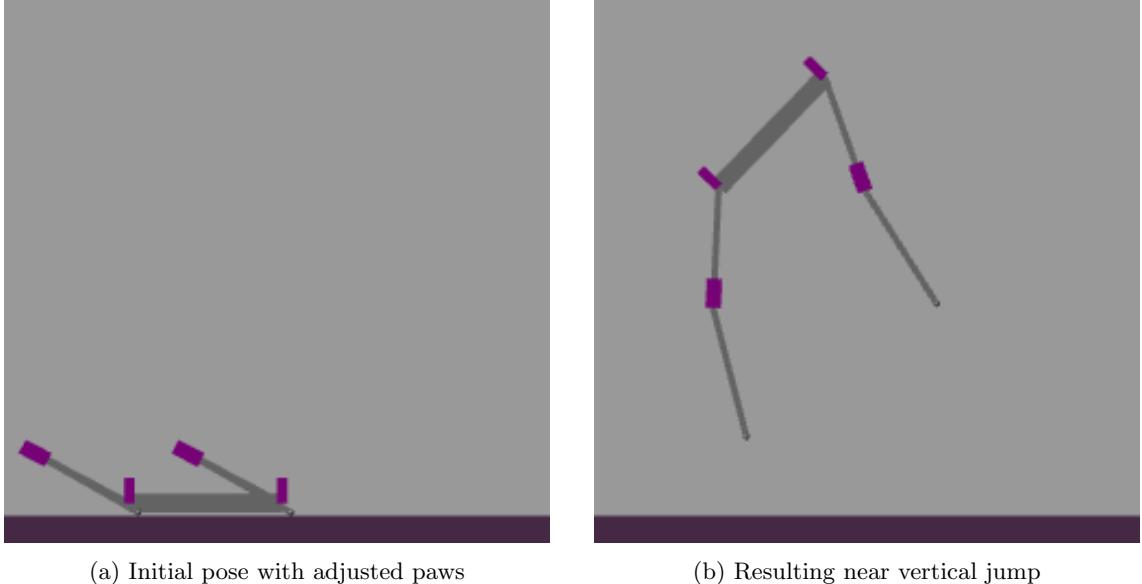


Figure 14: Adjusted paw placement in asymmetric configuration enables near vertical jump

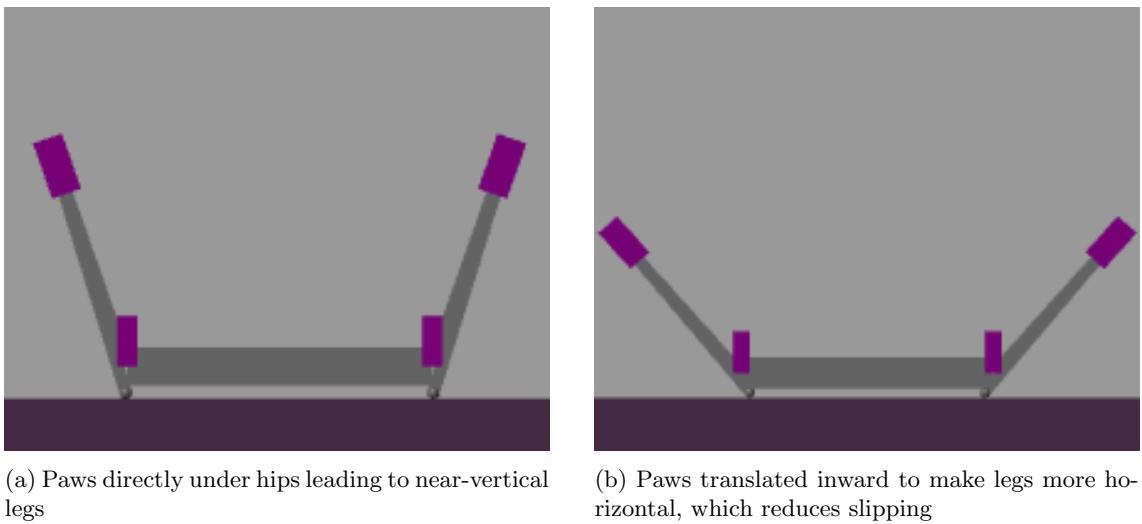


Figure 15: Paw placement adjustment for $L_2 > L_1$ configurations

For all cases, we calculate the distance d from hip joint to paw. For cases $L_1 = L_2$ and $L_1 > L_2$, where paw is placed directly under hip, d is vertical from the hip. For case $L_2 > L_1$, where paw is adjusted inwards towards the body, d is along the adjusted hip-to-paw vector.

d is minimized to maximize theta2 and thus spring potential energy while satisfying the pose constraints. Inverse kinematics then determines theta1 and theta2 , selecting the solution where knees bend outward. d is calculated as follows:

- For $L_1 = L_2$: $d = \epsilon$, where $\epsilon = 1\text{mm}$ ensures unique inverse kinematics solutions
- For $L_2 > L_1$: $d = L_2 - L_1 + \epsilon$, where $\epsilon = 1\text{mm}$ ensures the paw position is reachable given the numerical precision of the inverse kinematics solver
- For $L_1 > L_2$: $d = \sqrt{L_1^2 - L_2^2}$, derived when L_2 is horizontal (maximizing spring load) and forms a right triangle with L_1 and the hip-to-paw vector

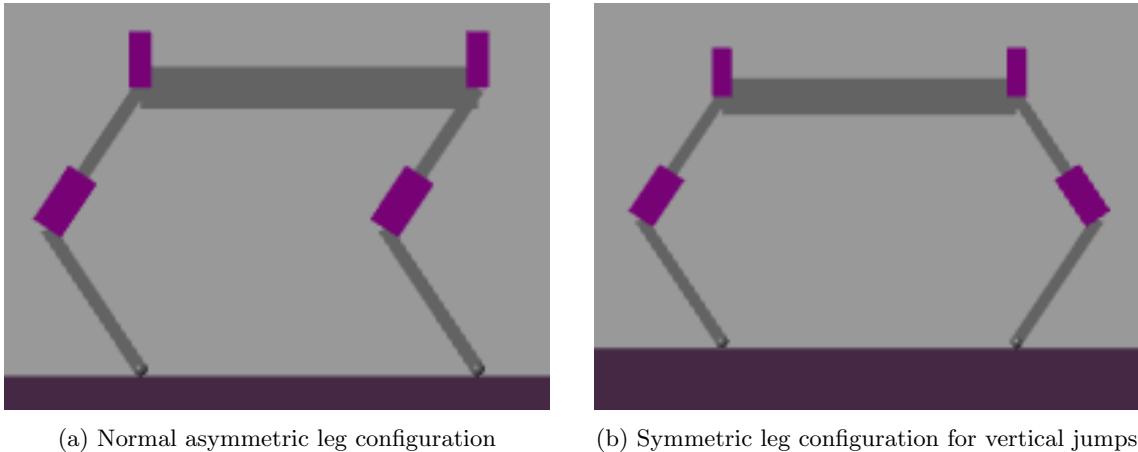


Figure 16: Comparison of normal and symmetric leg configurations

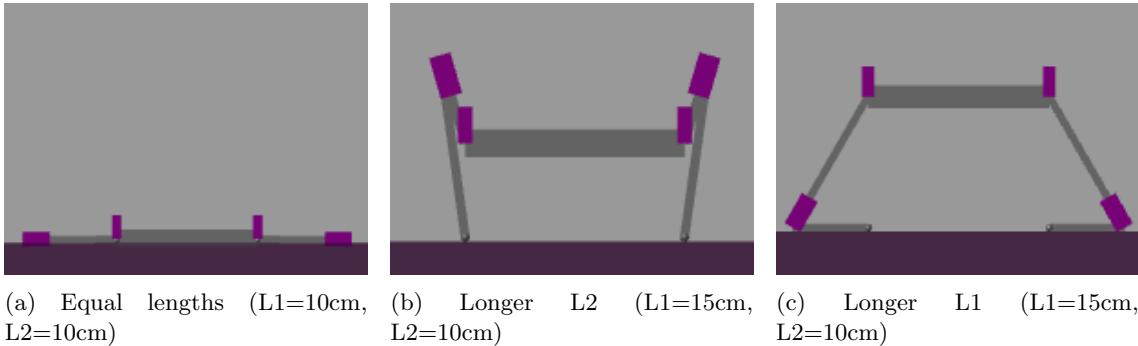


Figure 17: Initial poses for different link length ratios

6.4 Grid Search

The grid search varies two parameters: L_2/L_1 ratio, and total leg length L_1+L_2 , focusing around $L_2/L_1 \approx 1$ where preliminary tests indicated generally better performance. SimScape automatically updates the robot mass for each parameter set. Tests were run in both Earth (9.81 m/s^2) and Mars (3.72 m/s^2) gravity, with results in section 8.2.

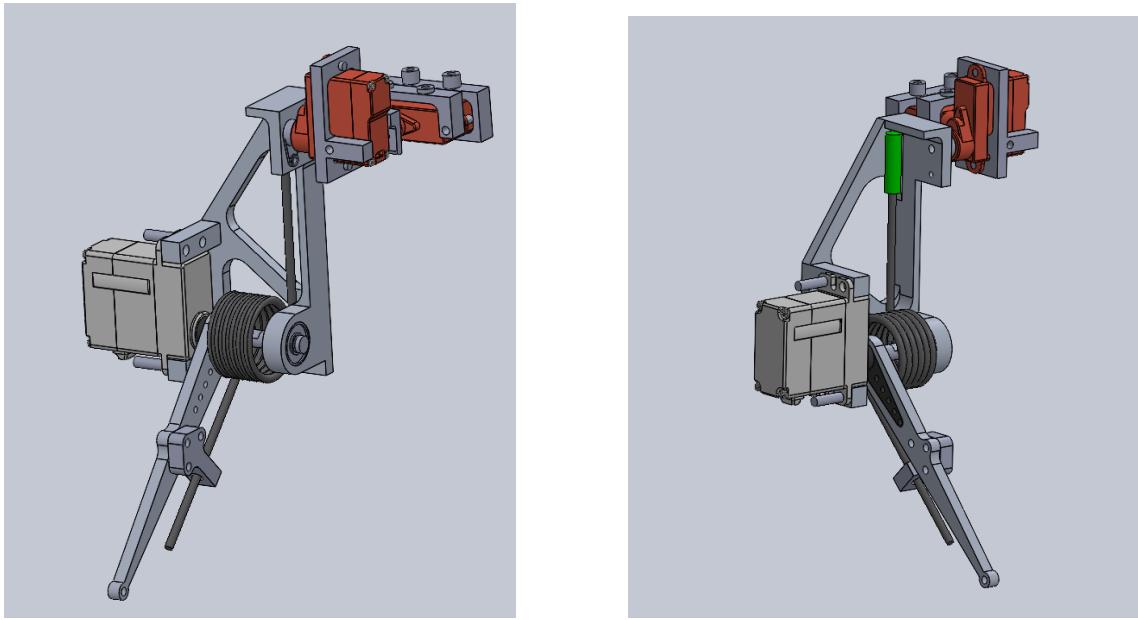
7 Robot Hardware

7.1 Torsional Spring Leg

Figure 18 shows the CAD model for the torsional spring leg design, including both knee and hip extension/flexion motor, as well as the hip adduction/abduction motor. The CAD model was created using the CAD software Solidworks. An annotated exploded view of the leg can be seen in figure 19, while an annotated exploded view for the hip motor housing can be seen in figure 20.

Components planned for in-house manufacturing are shown in figure 21. The axle that will be threaded and screwed directly into the motor shaft, and lead directly into a ball bearing, is emphasized in red. The leg is designed for aluminum manufacturing due to aluminum's high strength-to-weight ratio and machinability. While 3D-printable plastics are lighter, aluminum's superior strength allows for a lighter overall design when accounting for the required structural integrity [**Aluminum**] [**PLA**], barring complications from manufacturing.

Initial load testing of a 3D printed thigh prototype with the spring (figure 22) indicates sufficient strength for spring compression, suggesting 3D printing may be viable for the leg structure. The



(a) The bearing supporting the knee-joint shaft reduces off-axis loads on the motor shaft.

(b) A green plastic (PLA) holster reduces friction at the spring-leg contact point.

Figure 18: Torsional spring leg CAD model.

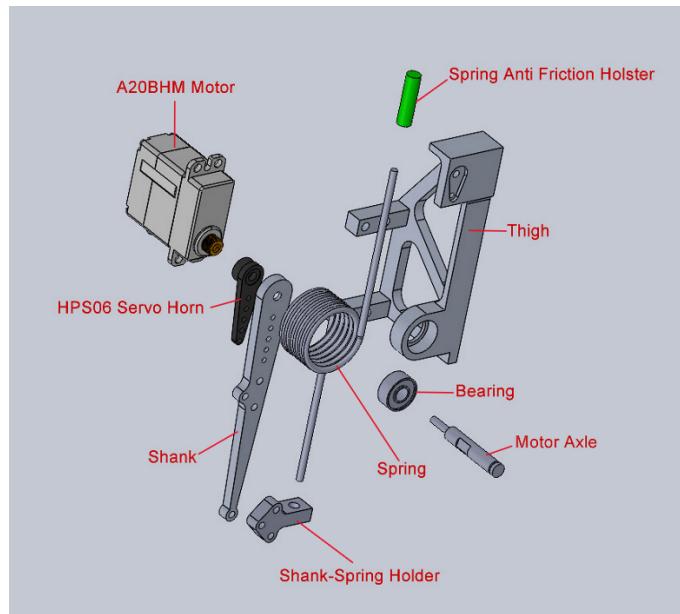


Figure 19: Annotated exploded view of the CAD leg design.

motor shaft axle will still be manufactured in aluminum.

7.2 Extension Spring Leg Design

The extension spring design (figure 23) was abandoned due to spring-shank collision at full knee flexion (-180 degrees). Avoiding this collision by moving the spring attachment point inward toward the body such that the spring could lie parallel to the leg would create an unacceptable moment arm on the motor shaft.

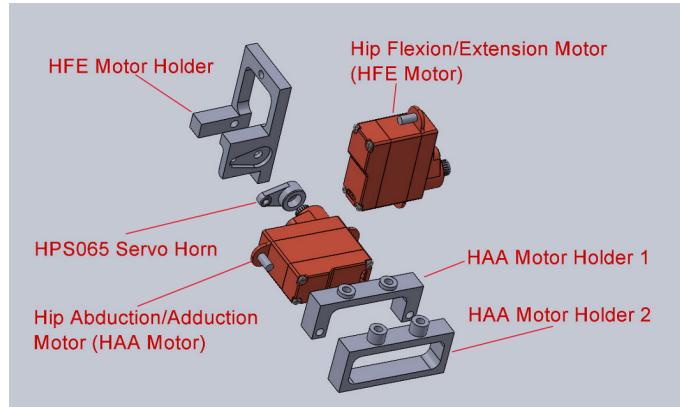


Figure 20: Exploded view of the hip joint motor housings.

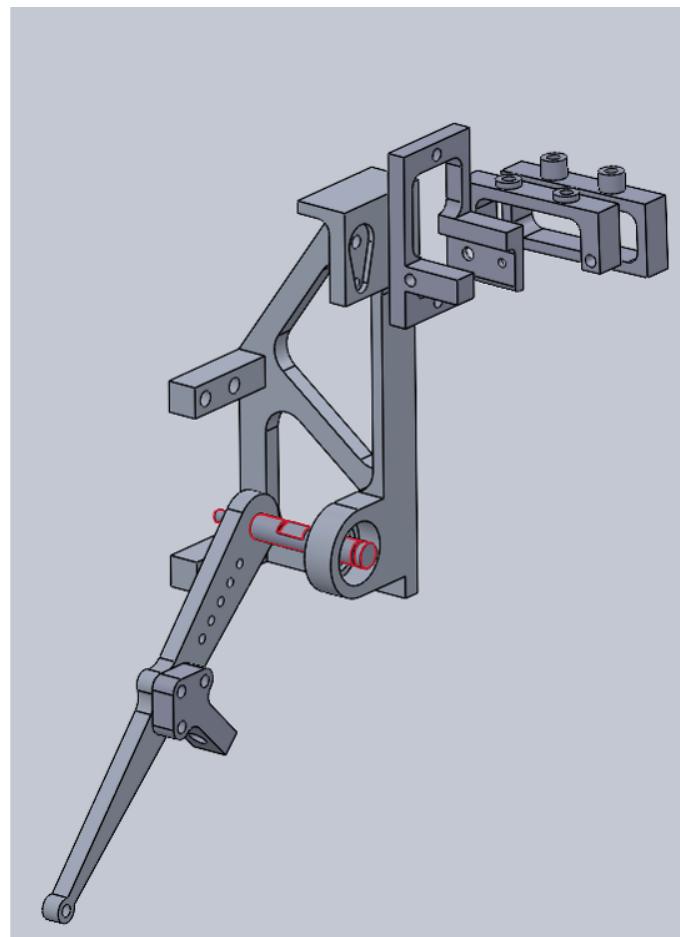


Figure 21: CAD model of components for in-house manufacture, with the motor shaft axle highlighted in red, which will lead directly into a ball bearing.

7.3 Motor Selection

AGF-RC motors were selected for their superior torque-to-weight ratio. Table 5 lists the specific motors chosen, with detailed specifications in appendices A to B.

The A20BHM was chosen over the heavier A35CHM (specifications in appendix C) which offers only marginally higher torque. We were unable to find other motors in a similar weight class that could provide comparable torque and speed. The A06CLS V2 selection reasoning is detailed in section 4.5.



Figure 22: 3D printed thigh prototype with torsional spring.

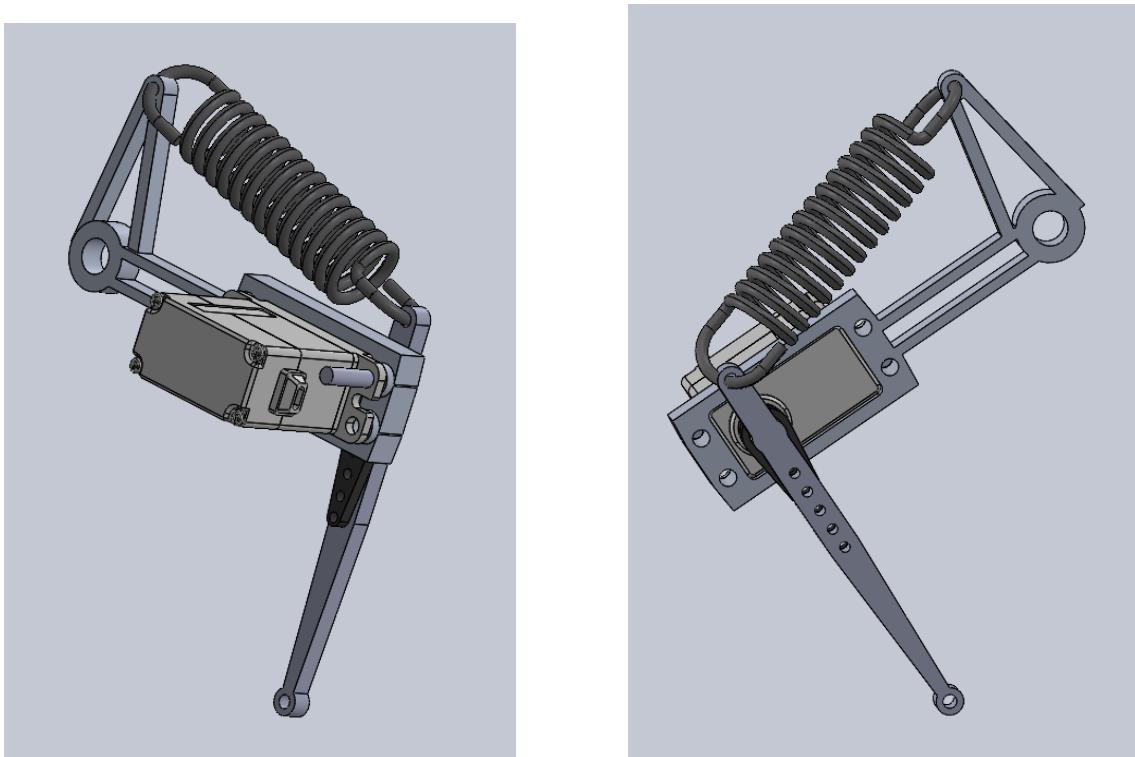


Figure 23: Extension spring configurations: outside (left) and inside (right) the leg.

Joint	Motor
Knee flexion/extension	A20BHM
Hip flexion/extension	A06CLS V2
Hip adduction/abduction	A06CLS V2

Table 5: Selected Motors

The physical thigh design is heavier than its simulation counterpart, which is simplified to a rectangular prism, due to additional width needed for motor and spring integration. However, this mass difference is negligible compared to the motor and body masses. Table 6 compares thigh masses for different materials.

Material Density	Simulation Thigh Mass	Actual Design Mass
1200 kg/m ³ (Tough PLA) E	N/A	6.45 g
2700 kg/m ³ (Aluminum 6061)	2.26 g	14.27 g

Table 6: Thigh mass comparison between simulation and actual design for different materials (calculated using Solidworks Mass Properties).

7.4 Other Purchased Components

Table 7 lists additional required components, excluding standard screws and nuts.

Component	Name/Part Number	Supplier
Bearing	RS: 612-5802	RS
Left leg spring	T075-180-484L	Sodemann
Right leg spring	T075-180-484R	Sodemann
Knee servo horn	HPS06	AGF-RC
Hip servo horns	HPS065	AGF-RC

Table 7: Additional purchased components.

8 Results

8.1 Motor Friction Estimation

Figures 24 and 25 compare the measured pendulum angles against simulations using two friction models: one with only viscous friction and one with both viscous and Coulomb friction. Table 8 presents the estimated friction parameters and Mean Squared Error (MSE) between simulated and measured angles for both motors.

The combined viscous-Coulomb model achieves better accuracy, particularly in the later phases of motion where angular velocities are lower. This aligns with expectations, as Coulomb friction becomes more dominant relative to viscous friction at lower speeds. The MSE values confirm this improved performance, as the MSE with Coulomb is about 1/3 of the MSE without Coulomb for both motors

Parameter	Hip Motor	Knee Motor
Viscous friction coefficient (without Coulomb) [Nm · s/rad]	0.003552	0.011612
Viscous friction coefficient (with Coulomb) [Nm · s/rad]	0.001703	0.006124
Coulomb friction [Nm]	0.010963	0.024127
MSE without Coulomb	0.003203	0.005989
MSE with Coulomb	0.000953	0.001967

Table 8: Friction model parameters and Mean Squared Error (MSE) for both motors, comparing models with and without Coulomb friction.

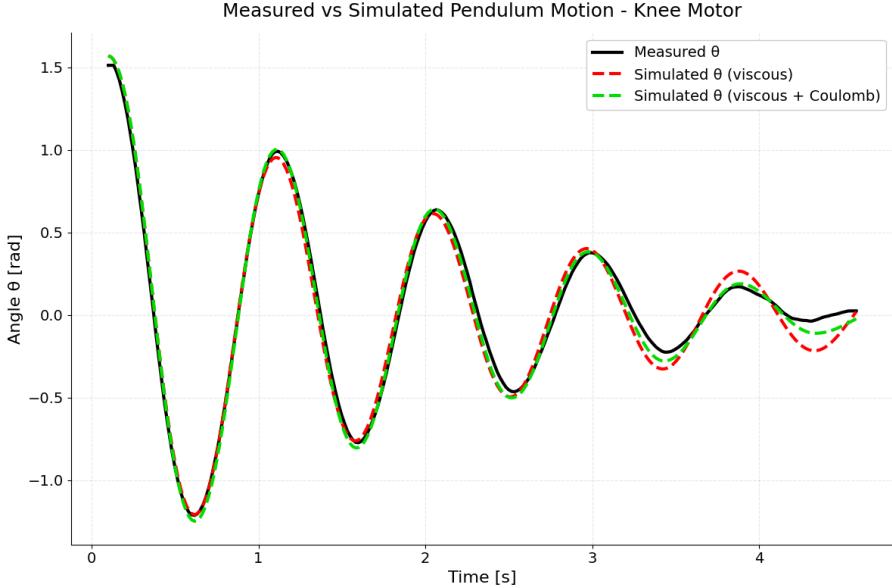


Figure 24: Linear regression fit of the pendulum data for the knee motor. Derived theta dot dot is the double derivative of the pendulum angle.

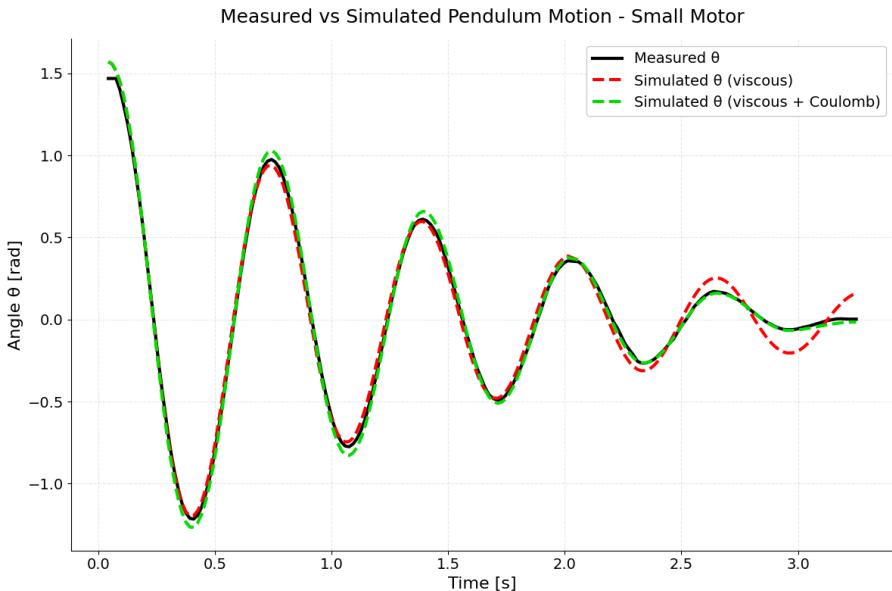


Figure 25: Linear regression fit of the pendulum data for the hip motor.

8.2 Link Length Optimization

Grid search results for Earth and Mars gravity are shown in figures 26 and 27. The search explored link length ratios $\frac{L_1}{L_2}$ from 0.8 to 1.6 and total lengths $L_1 + L_2$ from 10 cm to 36 cm, in increments of 0.1 and 1 cm respectively.

For both gravitational environments, a link length ratio of 1.0 yields optimal performance across all total lengths. The optimal total length in Mars gravity (28 cm) exceeds Earth gravity (20 cm) by 8 cm, achieving a 102.4 cm greater maximum jump height.

Jump performance degrades rapidly for ratios below 1.0, but declines more gradually above 1.0. The effective jump height is less steep in the total length dimension around the optimal point, allowing

flexibility in length selection with minimal performance impact. The performance landscape is non-convex, containing a secondary peak centered at $\frac{L_1}{L_2} \approx 1.3\text{-}1.4$ and total length ≈ 24 cm for Earth gravity, and at $\frac{L_1}{L_2} \approx 1.5$ and total length ≈ 35 cm for Mars gravity.

Table 9 summarizes the optimal parameters and their performance.

Gravity	Jump Height (cm)	Ratio	Total Length (cm)	L1 (cm)	L2 (cm)
Earth	45.39	1.0	20	10	10
Mars	149.79	1.0	28	14	14

Table 9: Best performing link length configurations and their corresponding jump heights for Earth and Mars gravity.

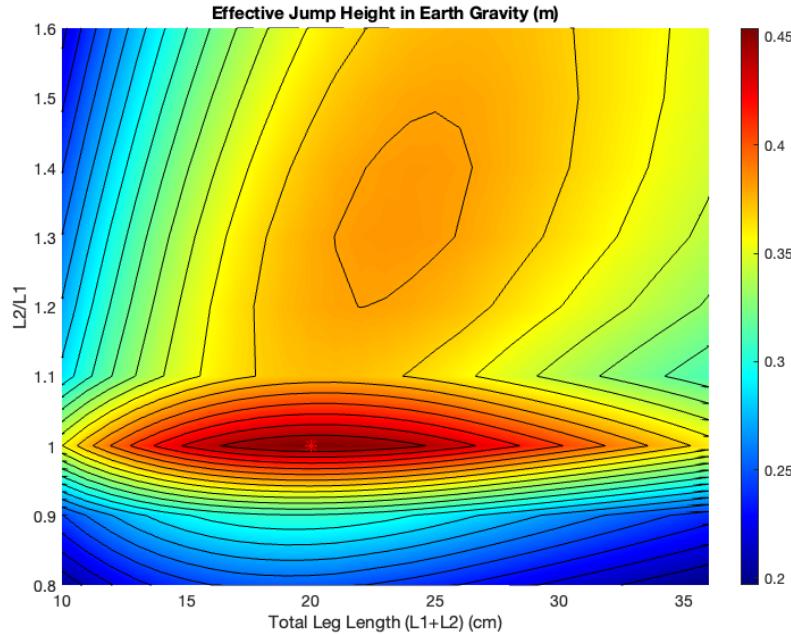


Figure 26: Grid search results showing jump height performance across different link length configurations under Earth gravity.

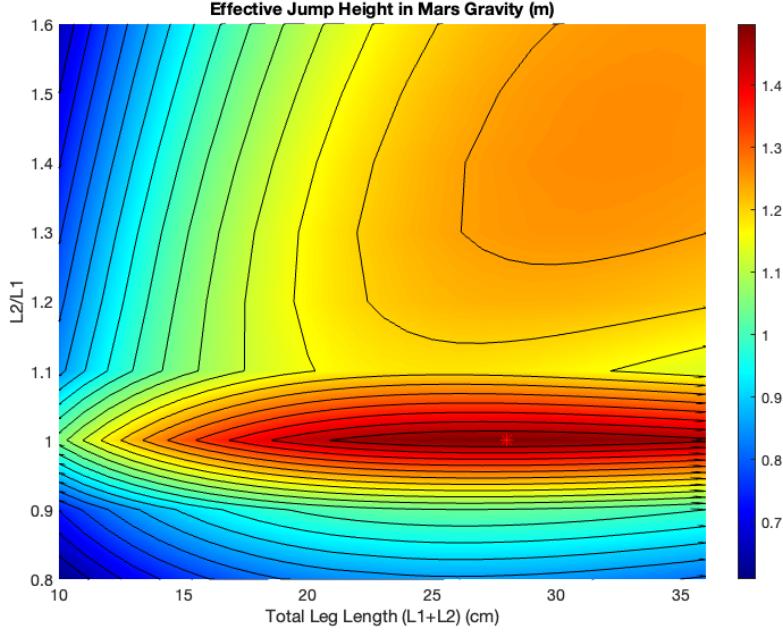


Figure 27: Grid search results showing jump height performance across different link length configurations under Mars gravity.

8.3 Hip Motor Dimensioning Test

As can be seen in figure 28, the hip motors follow the angle reference well, achieving three back and forth swings of 90 degrees over a one second period. The torque output of the motors during this maneuver can be seen in figure 29. In other words, the hip motors satisfy the heuristic as presented in [finn'tarek'master].

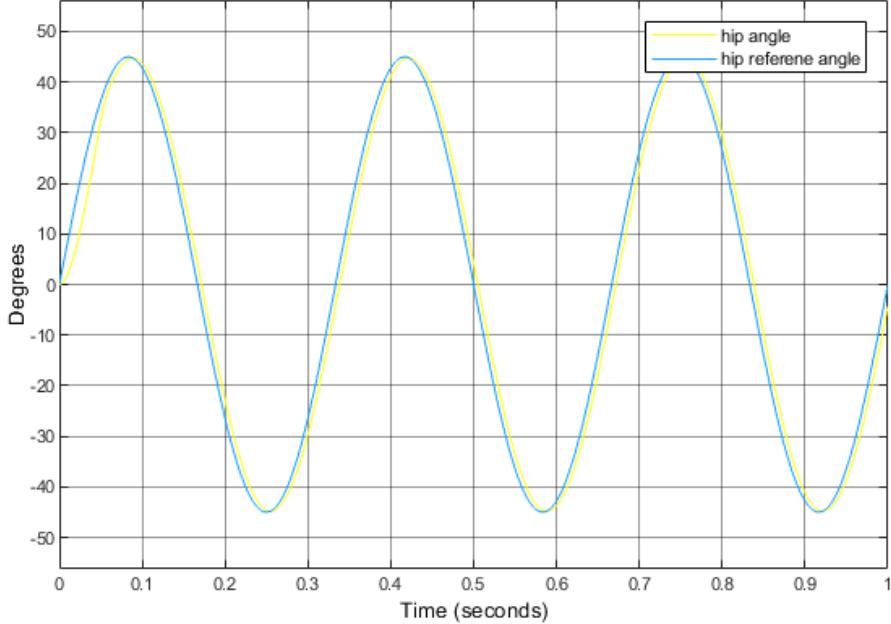


Figure 28: Commanded and actual hip joint angle achieved during the hip motor strength test simulation.

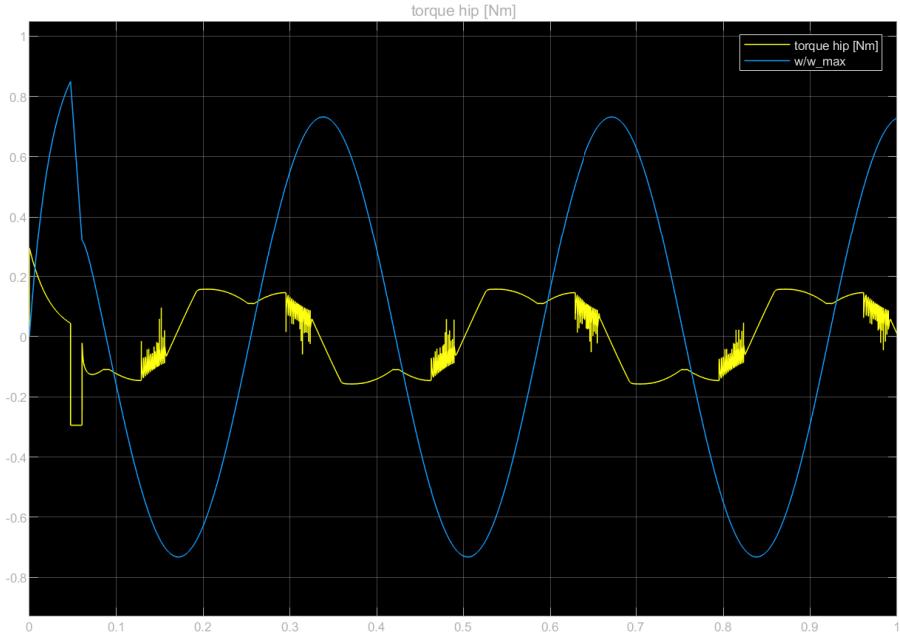


Figure 29: Torque output of the hip motors during the hip motor strength test simulation.

8.4 Motor Only Jumping Results

As described in section 4.2, motor-only simulations were done using the assumed characteristics of the A80BHP-H motor. Because the results were so far from acceptable, only the results of one experiment are presented here, but simulations with shorter leg and body lengths were performed. Figure 30 shows the knee joint actuation torque, knee joint velocity and knee joint angle during takeoff for the A80BHP-H motor. The robot is not able to reach a high enough speed to achieve a good jump, and the peak body center of mass height reached is 70cm, despite a total leg length of 40cm.

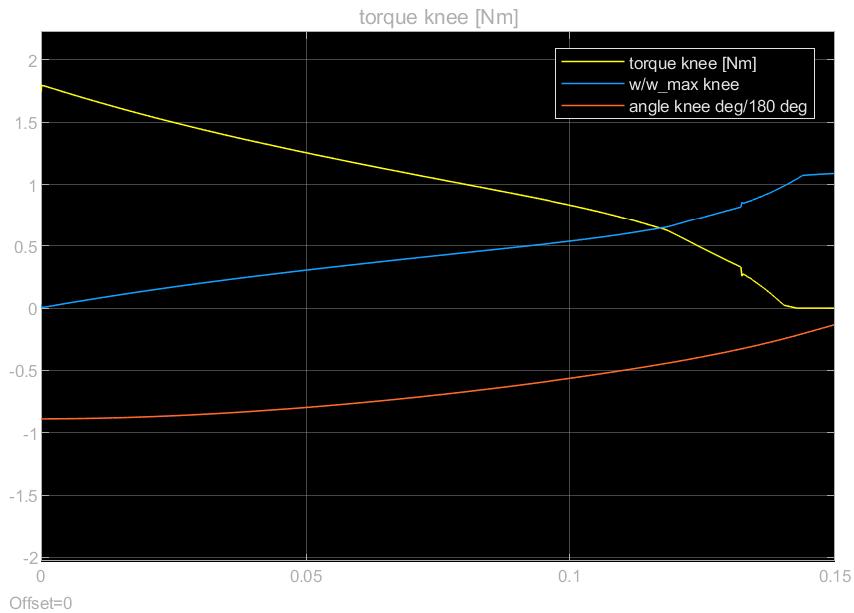


Figure 30: Knee speed until takeoff with A80BHP-H motor.

9 Discussion

9.1 Model Limitations and Uncertainties

9.1.1 Symmetric Jump Model Limitations

The symmetric jumping configuration approximates asymmetric jumping performance. For asymmetric legs, achieving vertical jumps requires careful paw placement relative to hip joints, varying with link lengths. While symmetric models place paws directly under hips for $L_1=L_2$ and $L_1 \neq L_2$, this produces angled rather than vertical jumps in asymmetric cases.

Experiments show vertical jumps are possible with asymmetric legs by adjusting hip-to-paw vector angle. However, the symmetric model only partially captures this through a constant angle offset when $L_2 > L_1$, failing to produce vertical jumps across all link lengths.

This introduces systematic biases:

- When $L_1 \neq L_2$: Underestimates jump height since asymmetric vertical jumps require initial poses that angle L_2 and L_1 more horizontally, better converting angular to vertical velocity with less slipping.
- When $L_1 = L_2$: Overestimates jump height since achieving vertical jumps requires L_2 to be less horizontal, where they are fully horizontal in the symmetric case 17a.

These biases explain the sharp performance peak at $L_2 = L_1$ in the grid search results (figure ??) for both Earth and Mars gravity.

9.1.2 Unmodeled Dynamics

While simulation results are promising, the lack of hardware testing introduces uncertainties about actual jumping performance. The simulated optimal link lengths may differ from real-world optima due to several unmodelled dynamics and approximations. There are unmodelled frictions like those between the springs and the legs. There are mass distribution differences from approximating legs and torso as uniform density rectangular prisms, and there are likely differences in the real and simulated masses of the robot.

9.1.3 Motor Loading Uncertainty

A critical uncertainty is the knee motors' ability to maintain stall torque while loading springs. As discussed in section 2.1, prolonged stall torque risks motor overheating. The time required to fully load springs at stall torque remains unknown, making it unclear whether motors can safely achieve maximum spring compression. Our 12% safety margin for stall torque falls below the recommended 20% 2.1, increasing risk that motors cannot load springs maximally.

9.2 Design Limitations

The optimal link lengths found for both Earth and Mars are too long for the current robot design. To avoid leg collision with bent knees, the longer lengths would require either elongating the robot body, or translating the hip joints outward. Body elongation would increase overall weight, while outward hip translation raises collision risks during aerial stabilization. Additionally, longer legs increase the inertia that motors must overcome during aerial maneuvers, potentially slowing aerial stabilization response.

Given these constraints, we opted for shorter link lengths without sacrificing significant performance. A potential fix would be to incorporate body length into the optimization process. Additionally, activating hip motors during jumps may yield shorter optimal link lengths.

9.2.1 Landing Challenges

The knee springs' equilibrium position at $\theta_2 = 0$ (straight leg) complicates landing. Achieving crouched landing poses requires knee motors to work against spring force, increasing response time. Further testing is needed to quantify motor response times across knee angles.

9.2.2 Hip Abduction/Adduction Design

As seen in figure 18, significant z-axis distance between shank end and hip abduction/adduction motor axis creates a significant and undesired motor loading. As the specialization project has primarily ignored abduction/adduction dynamics, adjusting the shank design to avoid this issue is considered an avenue of future work.

9.3 Post-takeoff rotational velocity

The common knee bending direction shifts the center of mass backward, causing the rear legs to support more weight while the front legs experience less resistance to acceleration. This imbalance generates backward rotation at takeoff, making aerial stabilization more challenging. Two potential solutions are to increase the spring stiffness of the rear knee springs or activate hip motors during jumps with greater torque applied to the back legs.

9.3.1 Limited Jump Control

Current design lacks motor actuation during jumps, preventing feedback control from compensating for parameter variations like spring stiffness. This could widen the sim-to-real gap. Adding hip motor actuation during jumps would enable feedback control - a focus for future masters thesis work.

10 Future work

10.1 Jump Performance Analysis

The upcoming master's thesis will investigate activating hip and knee motors during spring-powered jumps to improve performance and enable feedback control for handling sim-to-real gaps. This combined actuation may lead to different optimal link lengths. The thesis will also evaluate jumping performance in Simscape using the A80BHP-H motors, which provide twice the torque and speed of current motors, to determine if they should be incorporated into the design. We will also perform physical tests to see if the knee motors are able to provide stall torque for long enough to load springs maximally.

10.2 Hardware Implementation

The thesis will build and test the refined robot design. The shank design will be modified to minimize the moment arm between the shank end and hip abduction axis, eliminating undesired motor loads during jumps. We will also add elastic elements to absorb landing impacts.

10.3 Control

We will develop an accurate model in Nvidia Isaac Sim to train deep reinforcement learning controllers for jumping, stabilization, and landing. Domain randomization will help transfer policies from simulation to hardware, while curriculum learning will help the learning process. The policies will then be deployed and validated on the physical robot.

11 Conclusion

This project developed a small, lightweight and low-cost jumping quadruped that combines parallel torsional springs with BLDC motors. It is intended to be used as a platform for training DRL policies for jumping, aerial stabilization, and landing.

Simscape simulations showed motor-only actuation produced insufficient jump heights, leading to the addition of torsional springs. Grid search optimization revealed that equal-length links maximize jump height, though the optimal lengths were too long for a compact robot design.

Motor friction estimation through pendulum tests improved simulation accuracy, but hardware testing remains incomplete. We have not tested if motors can maintain stall torque during spring loading without overheating. Unmodeled dynamics like spring-leg interface friction and mass distribution differences will affect real performance. The symmetric jumping model differs from realistic asymmetric leg configurations, and simplified paw placement affects jump trajectory predictions.

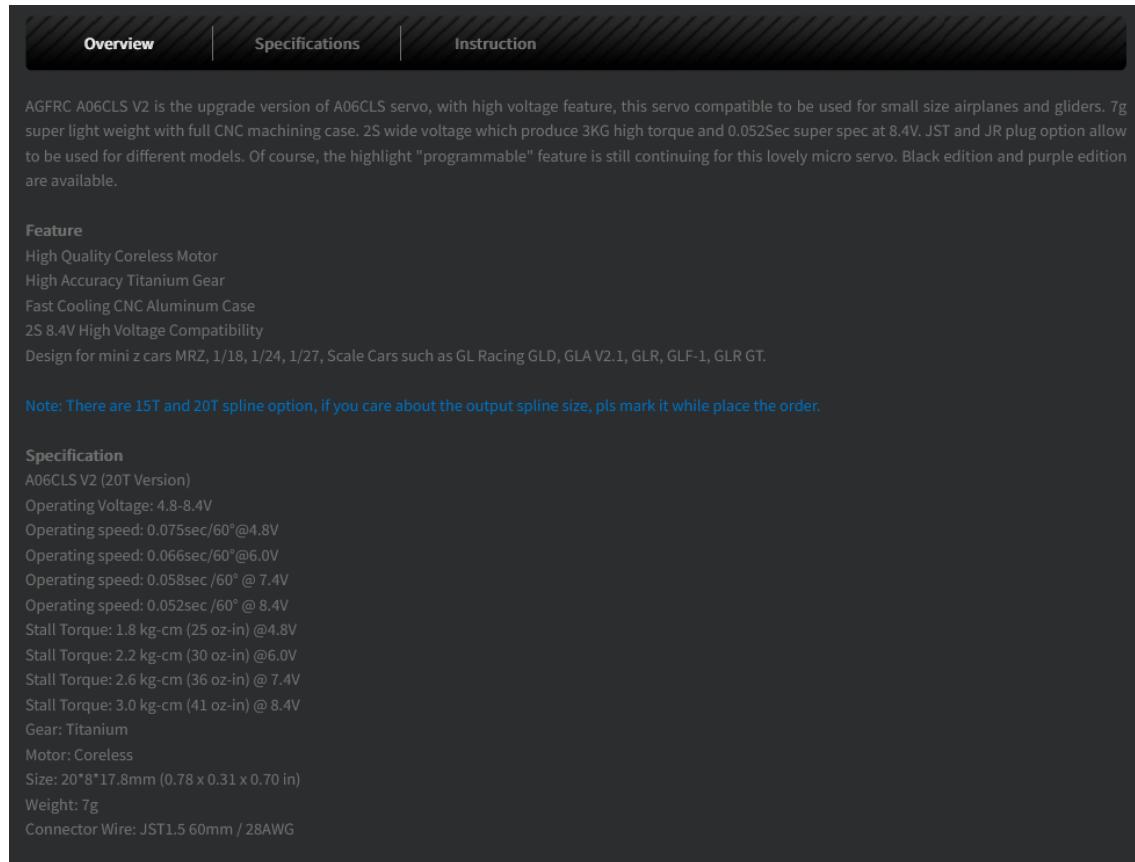
The current design has several limitations. Without motor actuation during jumps, potential energy remains untapped. Lack of feedback control during jumps prevents adjustment for unmodeled dynamics differences between simulation and real hardware. Spring equilibrium at straight leg slows landing response. Hip abduction/adduction motor placement creates undesired loading, and the common knee bending direction generates backward rotation at takeoff.

Despite these limitations, simulations indicate the design can achieve sufficient jump heights for demonstrating DRL controllers in both Earth and Mars gravity, setting the foundation for future hardware validation and control development in the upcoming master's thesis.

Appendix

A A06CLS V2 Website Information

This information is taken from the website: <https://www.agfrc.com/index.php?id=2666>



AGFRC A06CLS V2 is the upgrade version of A06CLS servo, with high voltage feature, this servo compatible to be used for small size airplanes and gliders. 7g super light weight with full CNC machining case. 2S wide voltage which produce 3KG high torque and 0.052Sec super spec at 8.4V. JST and JR plug option allow to be used for different models. Of course, the highlight "programmable" feature is still continuing for this lovely micro servo. Black edition and purple edition are available.

Feature

- High Quality Coreless Motor
- High Accuracy Titanium Gear
- Fast Cooling CNC Aluminum Case
- 2S 8.4V High Voltage Compatibility
- Design for mini z cars MRZ, 1/18, 1/24, 1/27, Scale Cars such as GL Racing GLD, GLA V2.1, GLR, GLF-1, GLR GT.

Note: There are 15T and 20T spline option, if you care about the output spline size, pls mark it while place the order.

Specification

- A06CLS V2 (20T Version)
- Operating Voltage: 4.8-8.4V
- Operating speed: 0.075sec/60°@4.8V
- Operating speed: 0.066sec/60°@6.0V
- Operating speed: 0.058sec /60° @ 7.4V
- Operating speed: 0.052sec /60° @ 8.4V
- Stall Torque: 1.8 kg-cm (25 oz-in) @4.8V
- Stall Torque: 2.2 kg-cm (30 oz-in) @6.0V
- Stall Torque: 2.6 kg-cm (36 oz-in) @ 7.4V
- Stall Torque: 3.0 kg-cm (41 oz-in) @ 8.4V
- Gear: Titanium
- Motor: Coreless
- Size: 20*8*17.8mm (0.78 x 0.31 x 0.70 in)
- Weight: 7g
- Connector Wire: JST1.5 60mm / 28AWG

Figure 31: A06CLS V2 Motor Information (Curt)

B A20BHM Website Information

This information is taken from the website:

<https://www.agf-rc.com/agfrc-a20bhm-21g-high-speed-0068sec-114kg-programmable-digital-brushless-48-84v-strength-steel-gear-micro-wing-servo-18010-for-airplane-aircraft-p4231199.html>

Description

Item specifics

Reviews (0)

AGFRC Aluminum Case HV High Torque Brushless Micro Digital Servo (A20BHM)

AGFrc A20BHM, is the 8.4V high voltage compatible performance digital mini servo, also the first launch brushless motor 12mm thickness RC servo. It is made in a robust CNC aluminum case with high accuracy metal gears, it particularly ideal for RC cars and other application.

Feature

- High Performance Digital Wing servo
- High Precision Metal Gears
- High Quality Brushless Motor
- Double Ball Bearings
- Full Aluminum Case

Specification

A20BHM

- Operating Voltage: 4.8-8.4 V
- Operating speed: 0.092sec/60° @ 6.0V
- Operating speed: 0.075sec/60° @ 7.4V
- Stall Torque: 8.5 kg-cm (118 oz-in) @ 6.0V
- Stall Torque: 10.2 kg-cm (142oz-in) @ 7.4V
- Gear: Strength Steel + Copper
- Motor: BLs
- Size: 23*12*27.5mm
- Weight: 21g
- Connector Wire: TYU 180 mm / 26AWG

C A35CHM Motor Information

PRODUCT SPECIFICATION A35CHM			
Control System	Pulse width modulation control		
Refresh Rate	333Hz		
Neutral Position	1520uS		
Signal Mode	Digital		
Dead band	2 uSec		
Operating Voltage	4.8V ~ 8.4V		
Operating Temperature	-15C°~ +70C°		
Bearing	Dual Ball Bearing		
Mechanical Limit Angle	220°		
Size	35.5*15*29.2mm		
Net Weight	41g		
Wire	JR 180mm /22AGW		
Operating Travel	180°±10°		
Signal Range	500 to 2500 uSec		
Stall Torque	9.5 kg-cm (132 oz-in) @ 4.8V 10.5 kg-cm (146 oz-in) @ 6.0V 11.5 kg-cm (160 oz-in) @ 7.4V 12.5 kg-cm (174 oz-in) @ 8.4V	Unload Current	230mA @ 4.8V 300mA @ 6.0V 380mA @ 7.4V 420mA @ 8.4V
Operating Speed	0.145sec/ 60°@ 4.0V 0.125sec/60° @ 6.0V 0.095sec/60° @ 7.4V 0.085sec/60° @ 8.4V	Loading Current	2000mA @ 4.8V 2400mA @ 6.0V 2600mA @ 7.4V 2800mA @ 8.4V
Direction	<input checked="" type="checkbox"/> CCW <input type="checkbox"/> CW		
Waterproof Level	<input type="checkbox"/> IP65 <input type="checkbox"/> IP67		
Angle Sensor	<input checked="" type="checkbox"/> Potentiometer <input type="checkbox"/> Magnet Angle Sensor		
Motor Type	<input type="checkbox"/> Brushless <input checked="" type="checkbox"/> Coreless <input type="checkbox"/> DC		
Motor Drive	<input checked="" type="checkbox"/> FET Drive <input type="checkbox"/> IC Drive <input type="checkbox"/> Transistor Drive		
Programmable	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No		
Gear Material	<input type="checkbox"/> Strength Steel <input checked="" type="checkbox"/> Titanium <input type="checkbox"/> Copper <input type="checkbox"/> Plastic		
Horn Gear Spline	<input checked="" type="checkbox"/> 25T-φ5.92mm <input type="checkbox"/> 25T-φ4.94mm <input type="checkbox"/> Other _____		
Case Material	<input checked="" type="checkbox"/> AL6061T6 <input type="checkbox"/> AL+Plastic <input type="checkbox"/> Plastic		
Bearing Material	<input checked="" type="checkbox"/> Metal <input type="checkbox"/> Plastic		
Horn Accessories	<input type="checkbox"/> AL6061T6 <input checked="" type="checkbox"/> Plastic		
Wire Color	Negative: <input checked="" type="checkbox"/> Black <input type="checkbox"/> Brown Positive: <input type="checkbox"/> Black <input checked="" type="checkbox"/> Red Signal: <input type="checkbox"/> Grey <input checked="" type="checkbox"/> White <input type="checkbox"/> Orange		

Figure 32: A35CHM Motor Information

D A80BHP-H Motor Information

Please note that this brochure gives an operating travel of 90 degrees, this was later confirmed to be false, the actual operating travel is 180 degrees.

SERVO SPECIFICATION

Operating Voltage	Gear Material	Operating Temperature	Case Material
4.8V ~ 8.4V	Steel	-15C°~ +70C°	Full Aluminum
Operating Travel	Mechanical Angle	Net Weight	Pulse Width
90°±5°	220°	79g	452 to 1072 uSec
Motor Type	Dead Band	Frequency	Waterproof Level
Brushless	2 uSec	760uS / 666Hz	/
Size	Bearing	Wire	Programmable
40*20*37.5mm	3BB	TYU 300mm / 22AWG	Yes

Operating Speed
0.030Sec / 60° @ 8.4V
0.034Sec / 60° @ 7.4V
0.039Sec / 60° @ 6.0V
0.045Sec / 60° @ 4.8V

Stall Torque
18.5KG-CM (257 oz-in) @ 8.4V
15.5KG-CM (215 oz-in) @ 7.4V
14.0KG-CM (195 oz-in) @ 6.0V
12.0KG-CM (167 oz-in) @ 4.8V



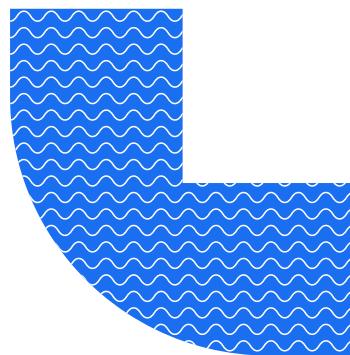
Figure 33: A80BHP-H Motor Information

E Ultimaker Tough PLA Technical Data Sheet

Ultimaker

Ultimaker Tough PLA

Technical data sheet



General overview

Chemical composition	See Tough PLA safety data sheet, section 3
Description	Ultimaker Tough PLA is a technical PLA filament with toughness comparable to Ultimaker ABS. Ideal for reliably printing technical models at large sizes, our Tough PLA offers the same safe and easy use as regular PLA
Key features	<p>With an impact strength similar as and higher stiffness compared to Ultimaker ABS, Tough PLA is less brittle than regular PLA and gives a more matte surface finish quality. Heat resistance is similar to standard PLA filaments, so printed parts should not be exposed to temperature above 58 °C.</p> <p>More reliable than ABS for larger prints, with no delamination or warping. Ultimaker Tough PLA is compatible with Ultimaker support materials (PVA and Breakaway), giving full geometric freedom when designing parts</p>
Applications	Functional prototyping, tooling, manufacturing aids
Non-suitable for	Food contact and in vivo applications. Long term outdoor usage or applications where the printed part is exposed to temperatures higher than 58 °C.

Filament specifications

	Method (standard)	Value
Diameter	-	2.85 ± 0.05 mm
Max roundness deviation	-	0.05 mm
Net filament weight	-	750 g
Filament length	-	~96 m

Color information

Color	Color code
Black	RAL 9017
White	RAL 9003
Green	RAL 6038
Red	RAL 3018
Gray	RAL 7000
Yellow	RAL 1018
Blue	RAL 5019

Figure 34: UM220509 Tough PLA TDS