ME112: Mechanical Systems Design

Professors Mark Cutkosky and Steve Collins

Team Froggy McFrogger

Holly Francis, Kathleen Miller, Kalvin Parker, Heidi Peterson

March 21, 2018

Contents

Table of Contents

1. Executive Summary

For the final project in ME112: Mechanical Systems Design, each team faced the challenge to create a small, battery-powered jumping robot, able to leap 1 meter vertically and attach itself via velcro to the underside of a surface within 5 seconds of being turned on. The robots were to be inspired by Mynocks, fictional creatures from the Star Wars universe, whose jump loosely resembles that of a frog. The team's additional challenge was to take a frog's jump -- a natural example of what we were tasked to do mechanically -- as inspiration for a biomimetic design.

Our team's final robot featured three major design choices: a lightweight frame, two legs stabilized by interlocking gears, and a hook and ring release mechanism. We decided to emphasize minimizing weight as the best way to maximize the jump height of our robot. With this in mind, we built the legs out of *1/16*" plywood with thin steel rods as shafts, a purchased 9.5-gram motor, and two 9-gram batteries, we were able to build a robot that weighed only 60 grams. This lightweight system allowed the miniature motor to wind up the system and store potential energy in six rubber bands. Once the robot crouched far enough to trigger the release mechanism, ring slipped off the hook and the robot propelled itself 1 meter into the air to connect solidly with the underside of the "spaceship" above.

The final robot was powered by a Polulu 6V Low Power Micro Metal Gearmotor with a 210.59:1 reduction running at 7.6V from two Lithium Ion batteries, each rated to output 3.7V at 400mAh. During the loading phase, the motor operated with an energy efficiency of approximately 42% and produced a maximum mechanical power output of 0.7W, and the system was able to store a total of 1.65J in the rubber bands. During the jump phase, the robot was able to reach a maximum height of 1.25m, at which point the system had 0.74J of potential energy.

Overall, our team was very happy with the performance of our robot. However, if allowed the opportunity for further improvements, we would change the material type of certain parts of our robot and consider a four-legged design. Our main points of failure were the structural integrity of the release mechanism and the ability of the knee joints to stand up against bending moments applied by the off center pull of the motor. This unbalanced pull was greatly offset by the addition of a low-friction pulley, but the structure was still in need of more rigidity to oppose bending.

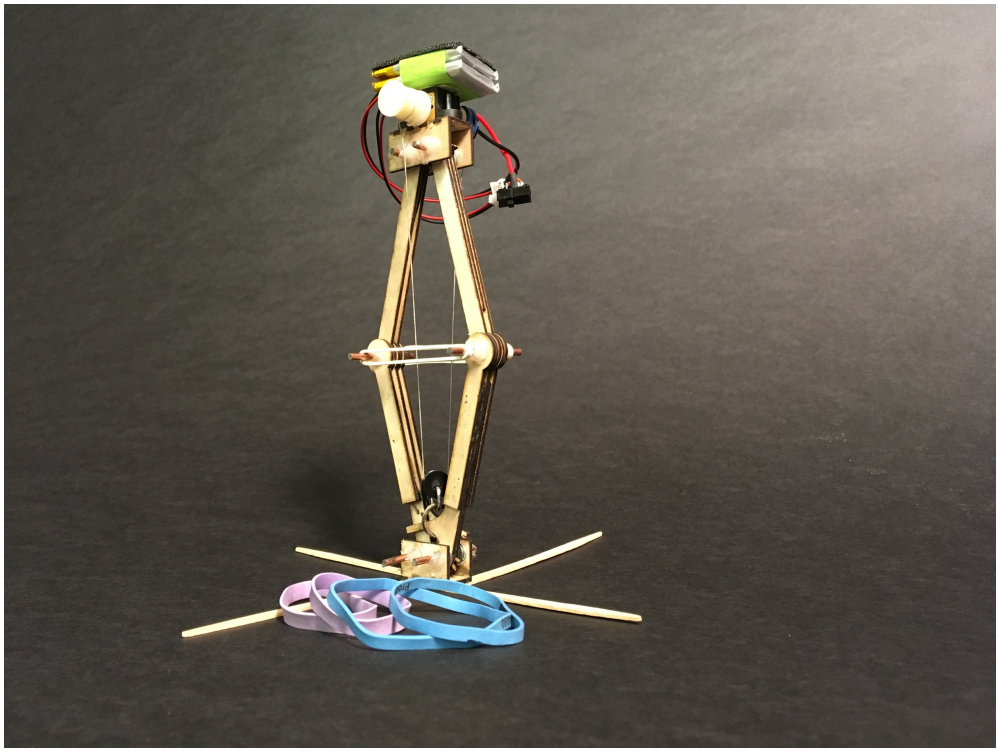


Figure 1. Final Mynock jumper design. Ready for wind-up..

2. Background

Inspired by the space-barnacle creatures, Mynocks, from Star Wars Episode IV, the goal of this project was to create a jumping robot that mimics the mechanics of these space parasites. Although shown in the movie and extended universe to function as a large winged creature that uses its tail as a spring to propel itself onto incoming spaceships, we took inspiration from real life jumping creatures, such as the frog, the locust, and the flea to develop a system that could accurately and effectively mimic the mynock's main functionality of vertically jumping and sticking to the horizontal surfaces of spaceships effectively and reliably.

When presented with the initial prompt, we set a goal for our team to create a creature that not only completed the challenge, but was also reliable and robust. This meant that we wanted to avoid any solutions that would require extensive setups or repairs following a jump and wanted to push towards a solution that utilized interchangeable parts in order to limit downtime if any part of the robot were to break.

Three existing robot design reviews helped guide us in our decision-making as we attempted to design a mechanical system that would jump ten times its own height. IOPScience's locust inspired jumping robot utilizes a winding device to load its torsional springs -- a system that our final drum and string mechanism resembled. UC Berkeley's Salto is a much more

sophisticated jumper, but the lightweight system demonstrated the importance of decreasing mass; a positive cycle as the robot requires less energy to propel itself upward, smaller springs, a smaller motor, and smaller batteries which in turn all continue to decrease the mass. A seven gram robot from the Swiss Laboratory of Intelligent Systems employed an interesting cam torsional spring load and release mechanism that had the benefit of structural simplicity -- no strings or bending moments to manage. We ultimately settled on the string and drum as a simpler system to design correctly, but this seven gram robot also proved the essential point of lightness. We differed noticeably from these three designs, however, in the construction of our legs. Rather than follow these examples of a one legged system, we settled on two legs to take advantage of symmetry. Since a vertical jump was essential to sticking to the Millenium Falcon, we wanted to avoid any tumbling if the center of mass was misaligned. These studies helped us simplify the areas of focus in our design to three major aspects: the motor and drum mechanism, the release mechanism, and the legs, as discussed in the Design Description section.

3. Design Description

This section takes a deeper look into our three design areas and provides reasoning for our decisions within each area. Figure 2 provides a general overview of each mechanism described below.

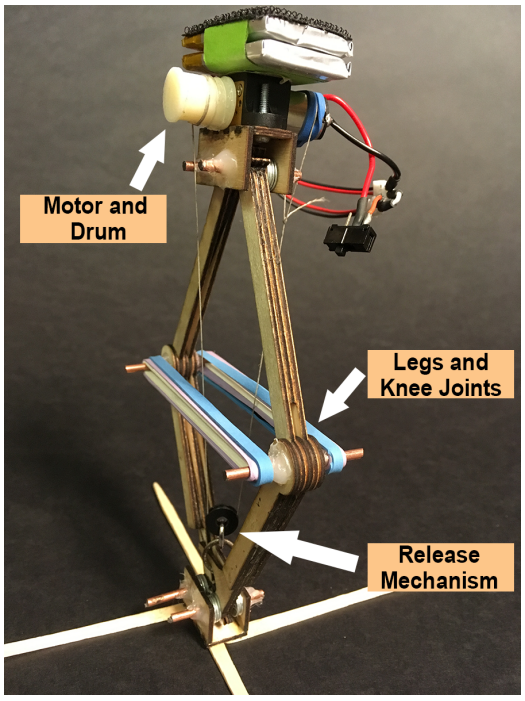


Figure 2. Our jumping robot in its unwound position.

3.1 Motor and Drum Mechanism

The motor and drum was one of the more straightforward aspects of our final design. Our motor and timing mechanism used a Pololu Low Power Micrometal Gear Motor, with a gear ratio of 210.59:1, a no load speed of 60 rpm, and a stall torque of 0.19 Nm, to stretch the rubber bands and lower the robot into a crouching position. Before purchasing a motor, we performed initial calculations using estimated values to determine what the requirements for our motor were (Appendix A). From those calculations, we determined that we needed a motor that could produce at least 0.16 Nm at a high enough speed to wind up the robot in under 5 seconds. The Polulu motor was the lightest motor we could find that was able to satisfy these requirements. Indeed, this motor was substantially smaller and lighter than the gearbox we were supplied with in class, weighing just over 9.5 grams as compared to the Tamiya 6-speed gearbox, which weighed over 75 grams fully assembled. To power the motor, we attached two lightweight Li-ion batteries, each rated at 3.7 volts and 400mAh, that we connected in series. When purchasing batteries, we had to ensure that the batteries we chose would supply enough voltage to run the motor and be able to withstand the maximum current that the motor might draw. Specifically, we needed the batteries to supply at least 6V and withstand a current of 360mA, which was the stall current of the Polulu motor. The batteries we ultimately purchased were the lightest option (each battery weighed ~9 grams) that satisfied these requirements (when placed in series) and had the added advantage of being rechargeable.

Figure 3. Pololu 6V motor with attached 210:1 gearbox.[3]

The timing drum we used was one of a selection of 3D-printed drums of various diameters that would allow us to increase or decrease the loading time and torque required for the loading phase. Though we used a smaller diameter drum in our initial iterations, we decided to use the 10mm diameter drum in our final design, which decreased the loading time without exceeding the allowable torque of the motor.

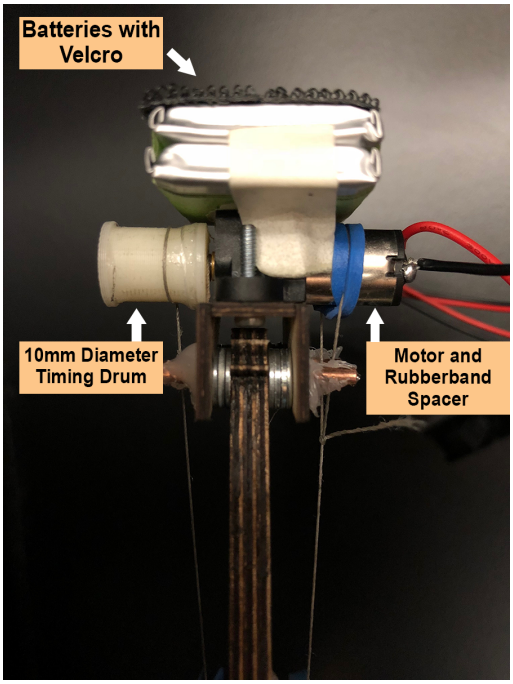


Figure 4. Image shows the motor and the timing drum. The spacer was created using a rubber band to easily adjust the distance of the string attachment point from the central axis of the robot. The gearbox attached to the motor is hidden by the motor mount.

3.2 Release Mechanism

When initially presented with the prompt for this jumper, we focused on designing a release mechanism that would allow us to fully load the robot and that would also stand up to the internal forces of the rubber bands that we would be using. In order to minimize energy usage, we decided on a passive release that would be attached to both legs and would deploy automatically. Our hook and ring system was simple and easy to make.

This mechanism was the most crucial aspect of our design and also the most fragile, as the thin plywood would be subject to the forces required to stretch the rubber bands. This led to a lot of breakage in the hooks during initial testing as the ring got caught on one of the hooks rather than releasing. By altering the shape of our hooks, we were able to reduce friction in the mechanism which allowed the ring to slide more freely. By changing the shape from hooked at the end to a linear slope, the ring was able to more freely slide along the path and released more consistently at the correct crouch point.

Early on in the build process, we directly connected the motor-driven drum to the release mechanism at the bottom section of the jumper, but this introduced a problem that we encountered in later iterations as well: the tension in the string would bow and bend the legs of

the jumper. To mitigate this issue, we had to equalize the force around central axis of the jumper to limit internal torques pulling it apart. To do this in our final iteration, we introduced a low friction pulley to the release mechanism and mounted the string to the opposing side of the motor. The equal and opposite torques reduced bending and stabilized the robot throughout the trials, and the low-friction pulley vastly reduced friction in the system as compared to earlier iterations that simply used a metal ring.

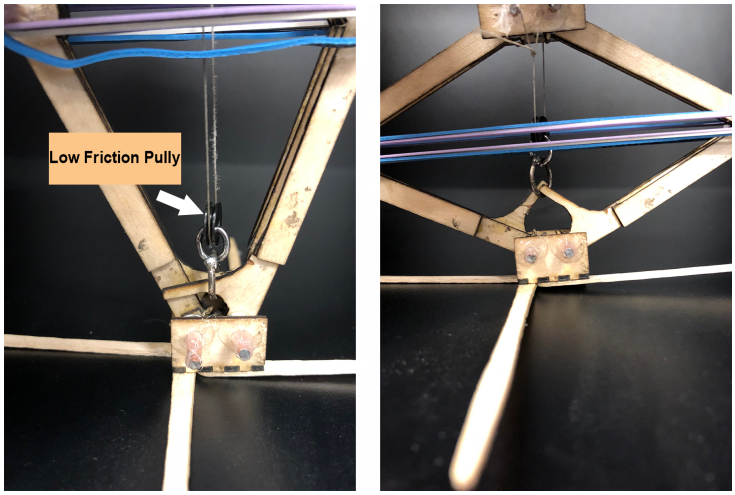


Figure 5. The release mechanism in its closed (left) and nearly released (right). The hook construction was initially longer, but was adjusted by removing material to improve jump timing.

3.3 Legs

In the literature that we reviewed, most designs could be classified as one of three basic shapes; they either consisted of a single knee joint, two knee joints that traveled in opposing directions, or four knee joints that were arranged on two perpendicular planes. We ruled out a single knee design almost immediately, as the requirement to keep the center of mass evenly distributed directly around the central axis of the robot introduced issues of balance, center of mass, and tumbling. By narrowing our design to double knee and quadruple knee designs, we were able to limit issues relating to stability and gained more freedom for the placement of our center of mass. We decided that in order to limit extraneous weight, a double knee design could provide enough stability at nearly half the weight of the quadruple leg design.

In early tests, however, the double knee design did present some issues. Due to the forces present at the knee joints that was created by the rubberband, our robot had a tendency to bend if the release mechanism was not set up at the correct angle. After adding supportive spacers around the points of rotation, our next step to solve the problem was to add two supports on either side of the lower knee joint. This fortified the simple support joint, prevented the joints' shafts from moving freely inside of the knee, and created a much more stable joint that could

hold up to more extreme forces. The final addition of the frictionless pulley solved the bending issue.

As seen in Figure 6, the support legs also have gears on the connecting ends to ensure one leg could not move independently of the other. This feature vastly improved the lateral stability of the legs and ensured that our Mynock jumper jumped straight up everytime. In our first iteration, we did not discern a difference between the left and right leg gears, but as issues emerged with stability and jump consistency, we realized that in order to ensure the legs moved at the same rate and that the top surface remained level, the gearing needed to be offset between the left and right legs.

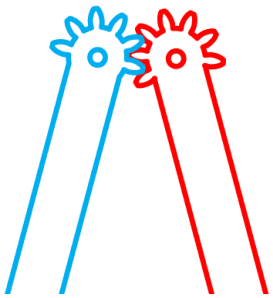


Figure 6. Illustrator rendering of robots internal gear system. Note that this view is hidden in Figure 5.

5. Analysis of Performance

This section reviews the final performance of our jumper, including energy flow calculations, motor characteristics, spring characteristics, and the effect of geometry on the forces required. Our design process began with approximate calculations that motivated choice of materials and general structure (Appendix A). Our assumptions of energy loss and inefficiencies allowed us to have a target motor torque and rpm to buy a new, lighter motor. Once we had materials to work with, we began to iterate on our first design – ironing out unforeseen issues (such as the bending moment on the legs from the motor drum) and improving the existing aspects that were already working (such as the timing gears that make the legs move together). Once the final design was working well, we recalculated our initial approximations to analyze what our jumper is actually capable of.

5.1 Motor Analysis

Table 1 shows the specifications of the Pololu motor, which we powered at 7.6V.

Specification	Value	Units
Rated voltage	6	V
Gear ratio	210.59:1	
No load speed	60	rpm
No load current	40	mA
Stall current	360	mA
Stall torque	27	oz-in

Table 1. Given motor characteristics. See figures below for graphs of motor characteristics at our operating voltage of 7.6V.

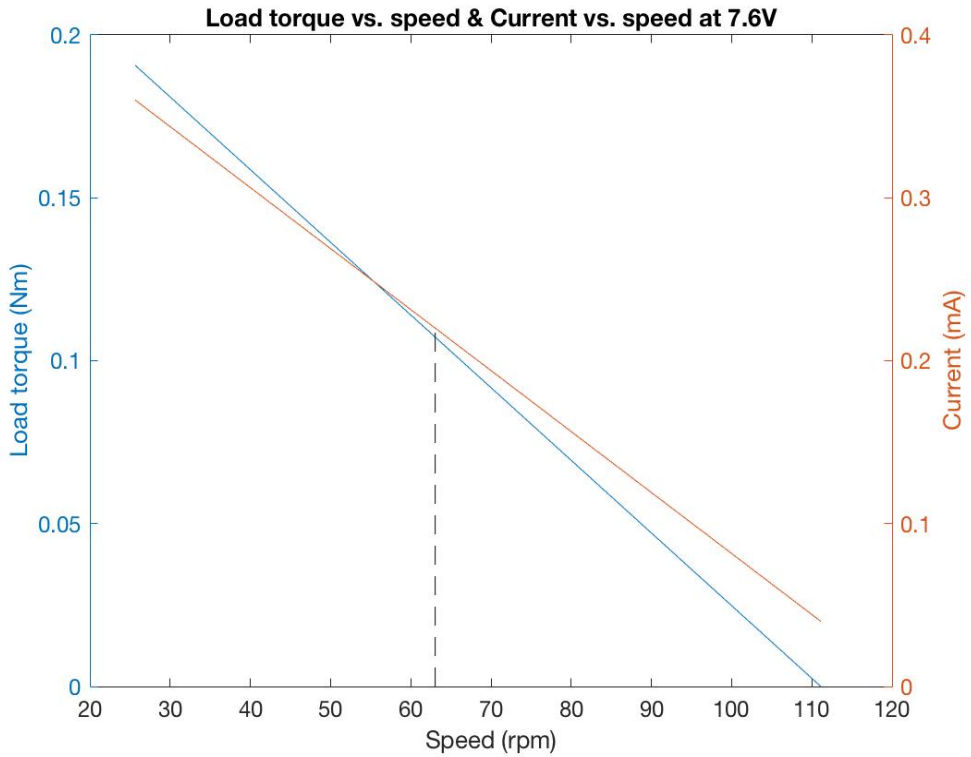


Figure 7. Motor characteristic curves. Load torque and current draw vs. speed of the motor. The dotted line marks the speed at which our motor was operating while pulling maximum current. See MATLAB source code in Appendix C.2.

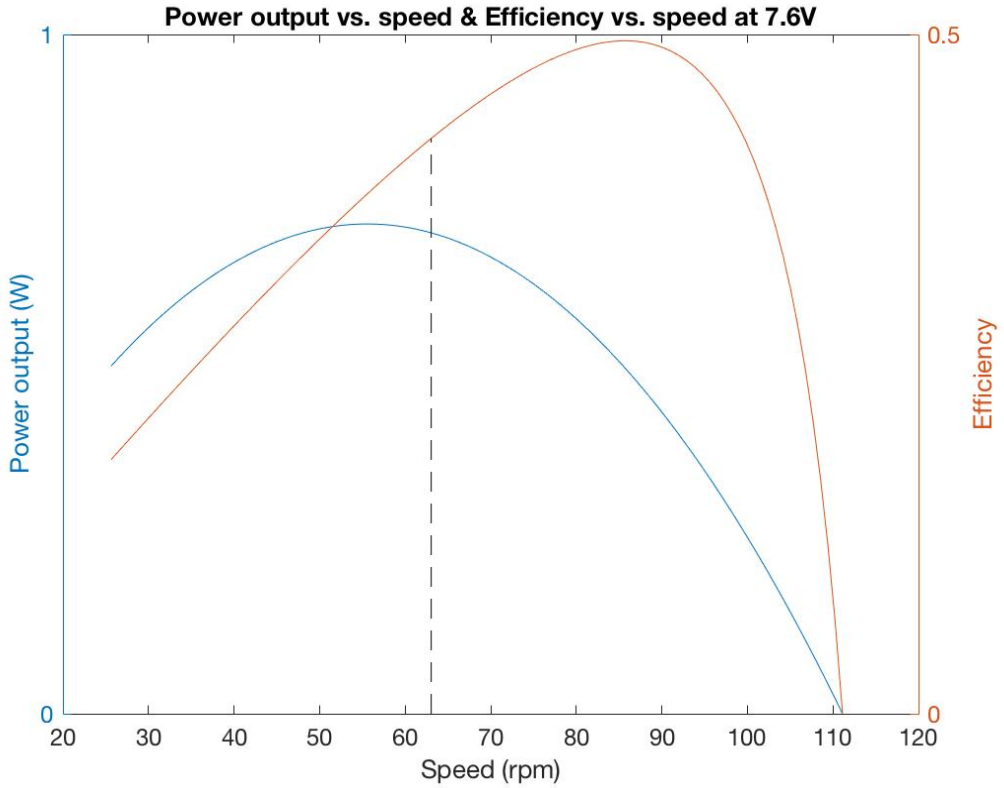


Figure 8. Motor characteristic curves. Mechanical power output and efficiency vs speed. The dotted line marks the speed at which our motor was operating while pulling maximum current. See MATLAB source code in Appendix C.2.

5.2 Rubber Band Analysis

Rubber bands were a particularly good choice for this challenge -- a prototype of the Mynock-inspired jumper. They are lightweight for the amount of energy they can store, and they are easily exchanged for other sizes to get the perfect K constant for our motor capacity, leg length, and time constraint. The final rubber band configuration was symmetrically attached to the shaft on either side of the knee joint. There was one blue, one white, and one purple on either side for a total of six. The differing slack lengths allowed the motor to overcome the first few centimeters without engaging all three of the rubber bands. Once the jumper was crouched enough to allow the geometry to provide more of a lever arm to the motor, the blue rubber band engaged to maximize energy storage.

Color	Slack Length (m)	Avg k (N/m)
blue 1	0.075	39.35405245
blue 2	0.065	53.3678744
purple 1	0.052	100.0361842
purple 2	0.048	117.473871
white 1	0.046	57.11174757
white 2	0.053	55.20603211

Table 2: Rubber band slack length and corresponding k coefficients. The coefficients were calculated using different weights and length measurements.

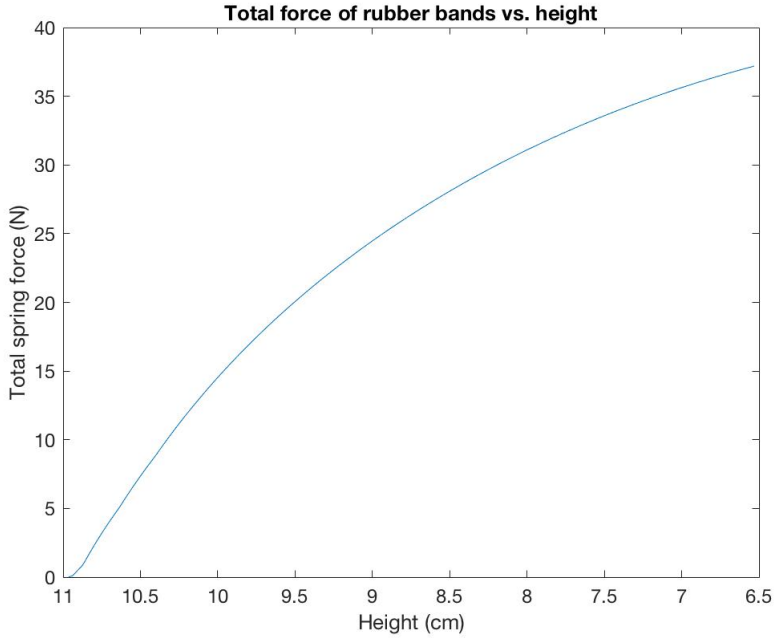


Figure 9. Force of the rubber bands as the jumper crouched. The force was calculated using a function that summed the force of each rubber band ($F_s = kx$) after it reached its slack length. See MATLAB source code in Appendix C.3.

5.3 Jump Phase

This phase details the energy required by the robot to reach its maximum height due to the change in potential energy.

Symbol	Value	Units	Description	Source
g	9.81	m/s^2	Gravitational constant	Constant
m	0.06	kg	Mass of system	Measured
h	1.25	m	Height of jump	Measured
PE_g	0.74	J	Gravitational potential energy at top of jump	$PE_g = mgh$

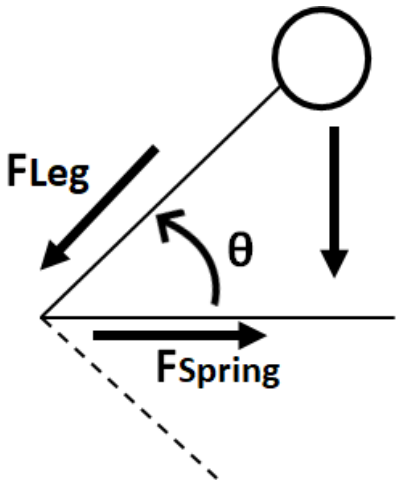
Table 3. Change in gravitational potential energy as the robot reaches maximum height.

5.4 Loading Phase

The loading phase details the change that the robot undergoes from its extended position with unstretched springs to completely crouched with energized springs.

Symbol	Value	Units	Description	Source
L_{min}	0.046	m	Minimum length of spring	Measured
L_{max}	0.142	m	Maximum length of spring	Measured
PE_{spring}	1.65	J	Stored potential energy in spring at launch	MATLAB (See Appendix C.3)
$\eta_{PE,\ spring}$	0.45		Efficiency between change from potential energy in the spring to gravitational potential energy	$\eta_{PE,\ spring} = PE_g/PE_{spring}$
k	Varying	N/m	Spring constant	MATLAB (See Appendix C.3)
$F_{spring,\ max}$	37.19	N	Maximum spring force	MATLAB (See Appendix C.3)
r	0.005	m	Drum radius	Measured
L_{leg}	0.07	m	Length of leg (between joints)	Measured
$T_{load,\ max}$	0.12	Nm	Maximum load torque supplied by motor	MATLAB (See Appendix C.1)

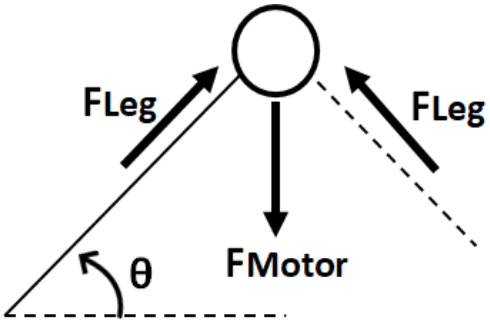
Table 4: The loading phase of the motor was dependant upon how the geometry of the system changed. As the jumper crouched deeper, the motor did not need to stretch the rubber bands as quickly. See the diagram below for an analysis of the geometry.



$$\sum F_x = 0 = 2F_{LegX} - F_{Spring}$$

$$F_{Leg} = \frac{F_{Spring}}{2\cos(\theta)}$$

Figure 10, Equation 1. Free body diagram of how the spring force projects onto the legs of the jumper. These are the forces acting on one knee joint. The structure is symmetrical.



$$\sum F_y = 0 = -F_{Motor} + 2F_{Leg}\sin(\theta)$$

Figure 11, Equation 2. Free body diagram of how the leg force projects in the direction of the motor force (tension in the string).

$$F_{Spring} = -k(x_2 - x_1)$$

$$x_2 = 2Lcos(\theta)$$

$$F_{Spring} = -k(2Lcos(\theta) - x_1)$$

Equations 3,4 and 5. Details of relationship between spring force and stretched length. x1 is the unstretched length of the spring.

$$F_{Motor} = ktan(\theta)(2L\,cos(\theta) - x_1)$$

Equation 6. Final equation detailing the relationship between the angle of crouch and the force output required by the motor.

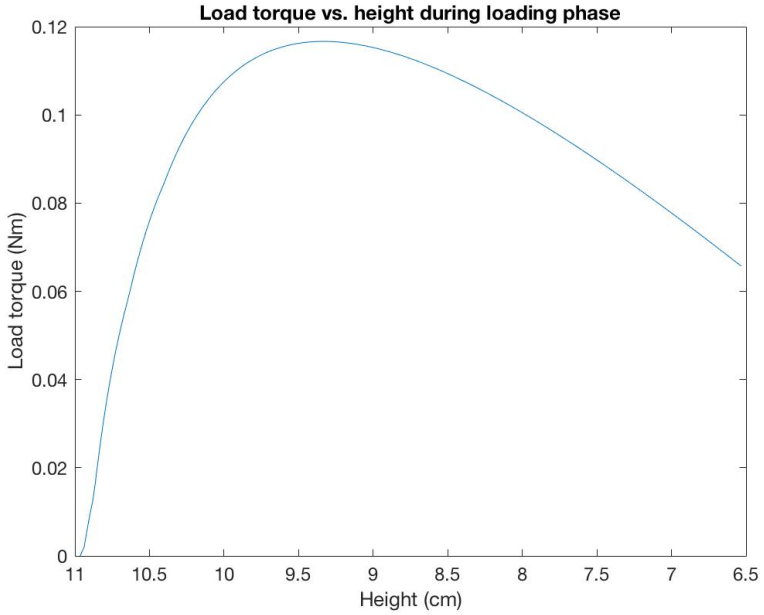


Figure 12. Because the geometry changed with crouch height, the torque that the motor was required to produce also changed with height. The maximum torque required occurred early in the crouch, but after all six rubber bands began to stretch -- the point at which the geometry did not provide a large lever arm for the tension in the string.

6. Redesign for Improved Performance and Reliability

In the extreme engineering task of making a robotic Mynock that can function fully and reliably in space, the robustness of the individual parts is what could really take our Mynock from successful in a controlled environment, such as Cubberley Auditorium, to successful in all environments. Many improvements could be made in the material selection alone. By using different materials, we would be able to improve structural strength, as well as mechanism function in the long run.

Throughout our testing, our largest issue was the strength of the plywood that we used. Because weight was a major concern, we decided that, under the stresses in the system, plywood would be an accessible option that we could easily alter, laser-cut, and glue throughout multiple iterations. For an object that would be consistently used in space, a material like Titanium or even reinforced carbon fiber would allow us to build legs that would stand up to and resist repeated bending moments without snapping, which happened often with plywood, without vastly increasing the weight of our jumper. Titanium and carbon fiber would also hold up to the vast fluctuations in temperature on an asteroid better than plywood would, and thin pieces, such as those used on the release mechanism, would not break off as easily under the same forces, which would improve the longevity of our robot.

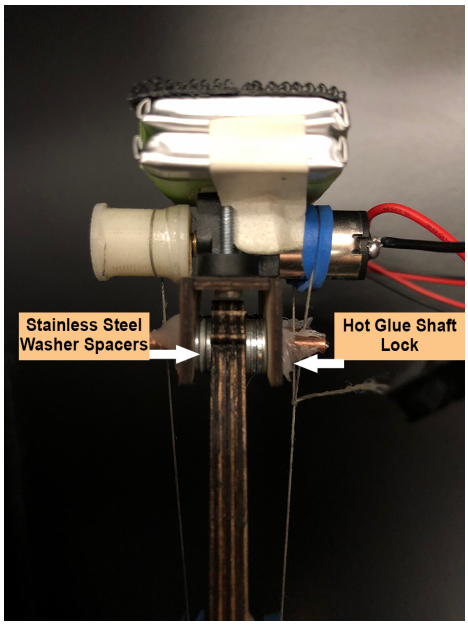


Figure 13. A close up of our spacers and the hot glue used to hold the shaft in the joints. Changing these two points could vastly improve our robots stability.

By improving the stability of our robot, would also be able to maximize its useful life, and better asses its function. Using polymer spacers, as opposed to the steel washers in Figure 13, would allow us to more accurately place the legs inside the body of the robot, which would improve gear meshing, and reduce extraneous forces on the gear teeth, while also reducing weight. Focusing on the joints of the robo-Mynock specifically, we would add more points of adjustment and improve lateral movement inside the joints by using threaded rods rather than hot glue to hold the upper and lower legs together. We would also be able to sandwich the rubber bands between the wingnut and the legs on the threaded rod, which would prevent the robot from losing its rubber bands in space, as well as allow us to add or remove springs as the gravity requirements for the planets change.

7. Conclusions

During final tests, our Mynock inspired robot was able to successfully launch itself at the spaceship, stick, and be used again after the jump. Our robot's success is largely due to the massive weight reduction from earlier versions, the high torque-weight ratio of our motor, and the consistency and reliability of our release mechanism. By keeping our robot's weight under 70 grams, we were able to minimize much of the strain on our internal systems by reducing the number of rubber bands that would be required to produce sufficient thrust. At the same time, our motor's relatively high torque output, coupled with a precision gearbox that reduced losses due to friction, allowed us to transfer much of the energy into vertical movement. The simplicity of our release mechanism allowed us to focus on improving other parts of our design early on, as the decision to go with a passive, rather than active mechanism, reduced the number of variables in our robot's jumps.

The main issue that arose during design and construction, was the strength of many of our parts. Although we chose a material with a fairly high specific strength, the size of our parts, along with the relatively large forces, proved to be our design's weakest point. In future iterations, improving our final material choice and removing points of failure in our joints and connections would considerably increase robustness, reduce losses to friction and bending stresses, and also increase the life of some of our more fragile parts.

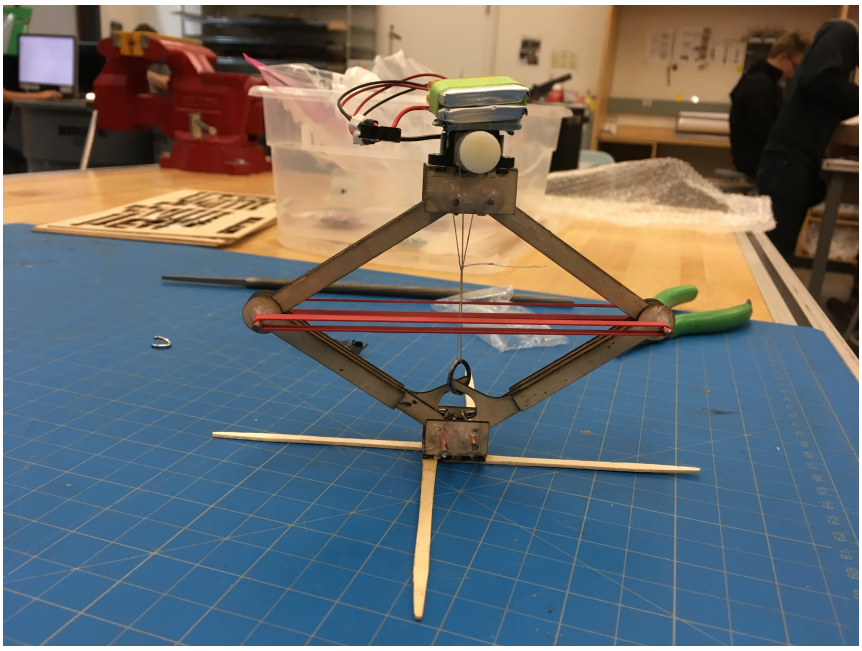


Figure 14. Our final jumper, crouched and ready to jump onto another spaceship.

8. References

[1] V. Z. and O. G. and U. B. H. and A. W. and A. A. and G. Kosa, "A locust-inspired miniature jumping robot," *Bioinspir. Biomim.*, vol. 10, no. 6, p. 66012, 2015.

[2] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Sci. Robot.*, vol. 1, no. 1, 2016.

[3] M. Kovac, M. Fuchs, A. Guignard, J. C. Zufferey, and D. Floreano, "A miniature 7g jumping robot," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 373–378.

[4] *210:1 Micro Metal Gearmotor LP 6V*. Pololu Corporation. 2016.

9. Appendices

A. Initial Calculations

Symbol	Value	Units	Description	Source
g	9.81	m/s^2	Gravitational constant	Constant
m_{est}	0.067	kg	Estimated mass of system	Estimate
h	1	m	Height of required jump	Requirement
PE_g	0.66	J	Gravitational potential energy at top of jump	$PE_g = mgh$
η_{KE}	0.7		Estimated efficiency of transfer from kinetic energy to gravitational potential energy	Estimate
KE	0.94	J	Required kinetic energy at launch	$\eta_{KE} = PE_g/KE$
v	5.3	m/s	Required velocity at launch	$KE = mv^2/2$
L_{min}	0.025	m	Estimated minimum length of spring	Estimate
L_{max}	0.15	m	Estimated maximum length of spring	Estimate
$\eta_{PE, spring}$	0.33		Estimated efficiency of transfer from potential energy in the spring to gravitational potential energy	Estimate
PE_{spring}	1.99	J	Required stored potential energy in spring at launch	$\eta_{PE, spring} = PE_g/PE_{spring}$
k	182	N/m	Required spring constant	$K = 2PE_{spring}/(L_{max}^2 - L_{min}^2)$
$F_{spring, max}$	22.8	N	Maximum spring force	$F_{spring} = k(L_{max} - L_{min})$
r	0.01	m	Estimated drum radius	Estimate
L_{leg}	0.076	m	Estimated length of leg (between joints)	Estimate
$\theta_{max\ torque}$	57	°	Angle at which maximum torque is applied	MATLAB (See Appendix C.1)
$T_{load, max}$	0.16	Nm	Maximum load torque supplied by motor	MATLAB (See Appendix C.1)

B. Additional Images

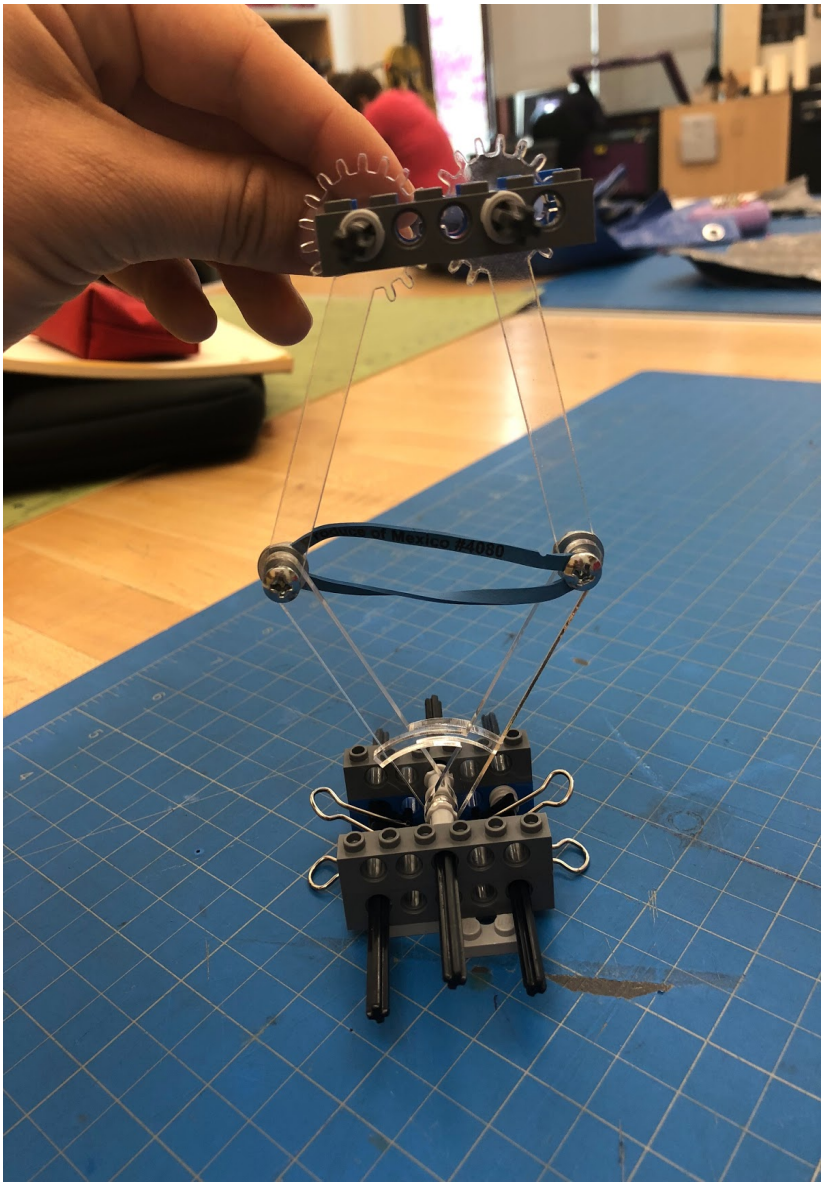


Figure 15. Our initial jumper prototype.

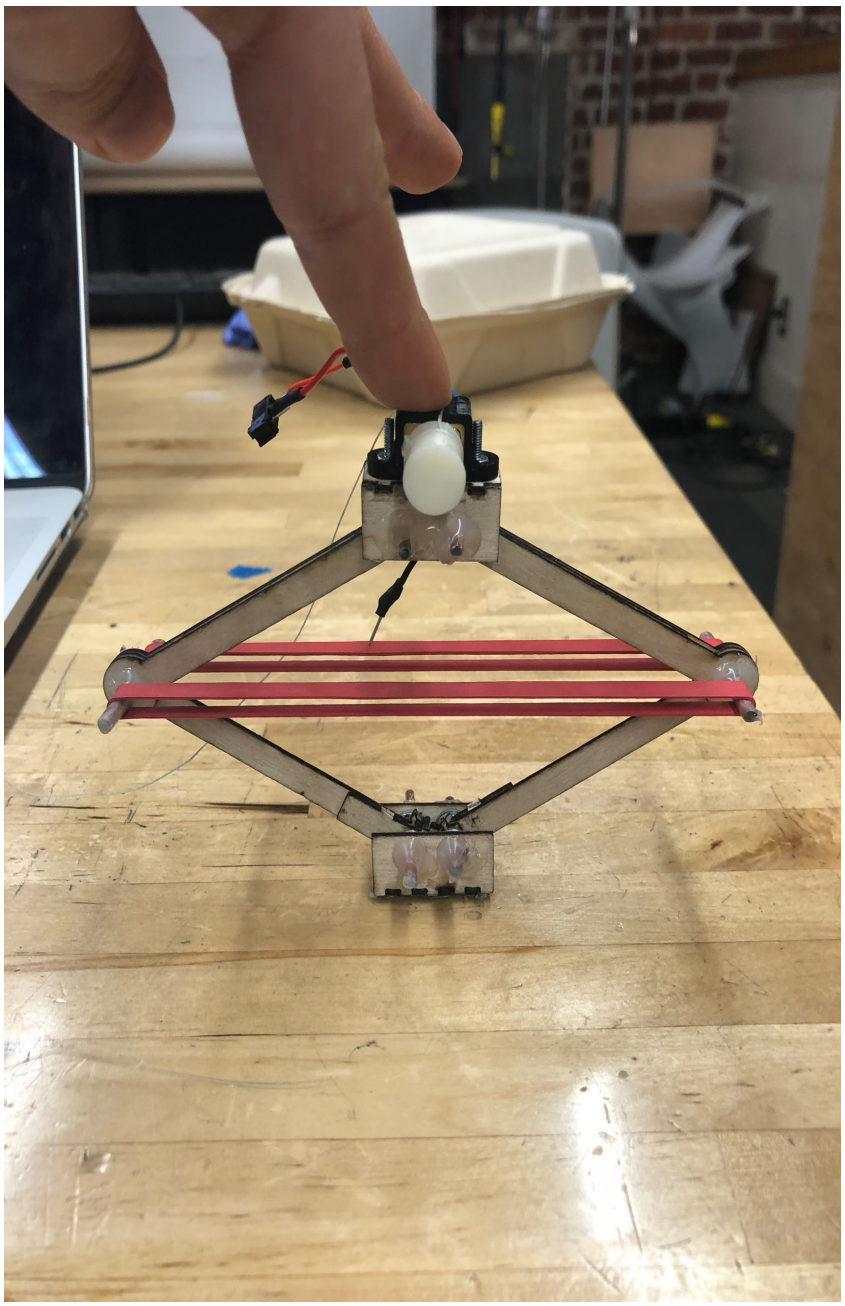


Figure 16. Our secondary jumper prototype.

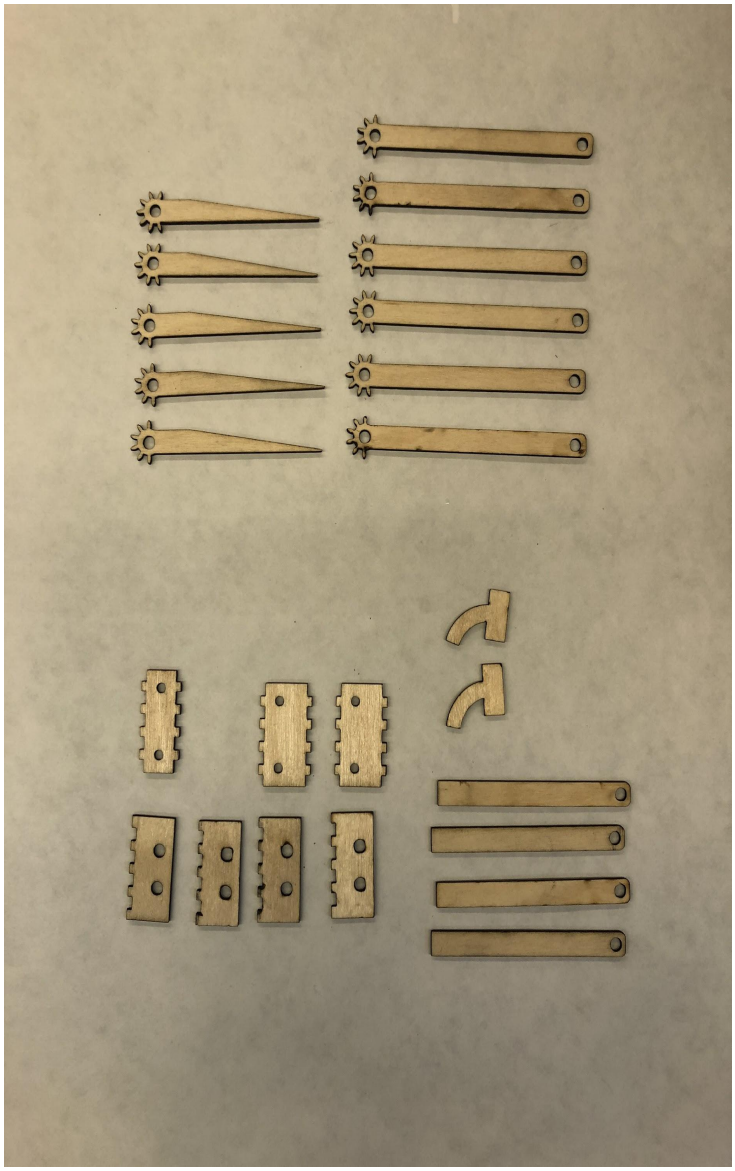


Figure 17. All of the laser cut parts, prior to assembly.

C. MATLAB Source Code

C.1 Initial Load Torque Analysis

```
close all; clear variables; clc;
k = 182.1; %N/m Estimated k constant
L = 3*.0254; %m Estimated leg length
R = 0.01; %m Estimated drum radius
x = .025; %m Estimated mimumum spring length
```

```
theta = linspace(10,acosd(x/(2*L)),100);
T_L = zeros(size(theta));

for i = 1:100
T_L(i) = R*tand(theta(i))*(k*(2*L*cosd(theta(i))-x)); %Load torque on motor
end

plot(theta,T_L);
set(gca, 'XDir','reverse')
xlabel('theta'); ylabel('Torque required oz-in');
title('Angle of compression vs motor torque required');
set(gcf, 'color', 'w')
set(gca,'FontSize',12)

thetamax = theta(find(T_L==max(T_L))) %Angle at which motor produces maximum torque
maxT = max(T_L) %Maximum load torque (Nm)
```

C.2 Motor Characteristics

```
close all; clear variables; clc;

%Characteristics
V_rated = 6; %V
i_s = 0.36; %A
R = V_rated/i_s; %Ohms
i_nl = 0.04; %A
T_s_ozin = 27; %oz-in
T_s = T_s_ozin*0.0070615518333333; %Nm
rpm_nl = 60; %rpm
w_nl = rpm_nl.*pi./30; %rad/s
k = T_s/(i_s - i_nl);
T_f = k*i_nl; %W-s or Nm

%Voltage we used
V = 7.6;

%Different currents (corresponding to different loads)
I = linspace(i_nl,i_s,100);

%Calculate omega, torque, power, and efficiency for each current
w_L = (V-I.*R)./k; %rad/s
T_m = I.*k; %W-s or Nm
```

```
P_in = I.*V; %W
T_L = T_m-T_f; %W-s or Nm
P_out = T_L.*w_L; %W
Efficiency = P_out./P_in;
rpm = w_L.*30./pi; %rpm

%Motor efficiency
Peak_output_power_6V = max(P_out);
Peak_power_w_L = w_L(P_out == Peak_output_power_6V);

%Peak efficiency calculations
Peak_eff = max(Efficiency);
Peak_eff_w_L = w_L(Efficiency == Peak_eff);

%Peak values for our robot
I_peak = 0.22; %A
w_L_peak = (V-I_peak.*R)./k; %rad/s
T_m_peak = I_peak.*k; %W-s or Nm
P_in_peak = I_peak.*V; %W
T_L_peak = T_m_peak-T_f; %W-s or Nm
P_out_peak = T_L_peak.*w_L_peak; %W
Efficiency_peak = P_out_peak./P_in_peak;
rpm_peak = w_L_peak.*30./pi; %rpm

figure(3)
[hAx, ~, ~] = plotyy(rpm, T_L, rpm, I);
line(hAx(2),[rpm_peak rpm_peak], [0 I_peak],'Color', 'black', 'LineStyle', '--');
title('Load torque vs. speed & Current vs. speed at 7.6V');
xlabel('Speed (rpm)','FontSize', 12);
ylabel(hAx(1), 'Load torque (Nm)','FontSize', 12);
ylabel(hAx(2), 'Current (mA)','FontSize', 12);
set(gcf, 'color', 'w')
set(hAx,'FontSize',12)

figure(4)
[hAx, hLine1, hLine2] = plotyy(rpm, P_out, rpm, Efficiency);
line(hAx(2), [rpm_peak rpm_peak], [0 Efficiency_peak],'Color', 'black', 'LineStyle', '--');
title('Power output vs. speed & Efficiency vs. speed at 7.6V');
xlabel('Speed (rpm)','FontSize', 12);
ylabel(hAx(1), 'Power output (W)','FontSize', 12);
ylabel(hAx(2), 'Efficiency','FontSize', 12);
set(gcf, 'color', 'w')
set(hAx,'FontSize',12)
```

```
%{
figure(5);
hold on
plot(w_L, T_L,  w_L, P_out, w_L, Efficiency)
%line([700 700], [0 1.4965],'Color', 'black', 'LineStyle', '--');
title('Motor characterization curves: Load Torque, Power, and Efficiency vs. \omega');
xlabel('\omega (rad/s)','FontSize', 12); ylabel('Motor characteristic','FontSize', 12);
legend('T_L (Nm)', 'P_{out} (W)', 'Efficiency');
lgd1.FontSize = 12;
set(gcf, 'color', 'w')
set(gca,'FontSize',12)
%}
```

C.3 Rubber Band Analysis

```
clear variables; close all; clc;

slack_L = [0.075, 0.065, 0.052, 0.048, 0.046, 0.053]; %m
k = [39.4, 53.4, 100, 117.5, 57.1, 55.2]; %N/m

min_L = min(slack_L); %m
max_L = 0.142; %m
L = linspace(min_L, max_L, 100); %m

%Geometry of robot
leg_L = 0.07; %m
top_h = .031; %m
bottom_h = .011; %m
space = 0.01; %m
adjacent = (L - space)/2; %m
opposite = sqrt(leg_L^2 - adjacent.^2); %m
theta = acosd(adjacent./leg_L); %degrees
h = top_h + opposite + bottom_h; %m

F = zeros(size(L,2), 6);
PE = zeros(size(L,2), 6);
F_total = zeros(size(L,2),1);
PE_total = zeros(size(L,2),1);
T_L = zeros(size(L));

r = 0.005;
```

```
for i = 1:100
    for j = 1:6
        if(L(i) >= slack_L(j))
            F(i,j) = (L(i) - slack_L(j)).*k(j); %N
            PE(i,j) = 0.5*(L(i) - slack_L(j))^2.*k(j); %J
        end
    end
    F_total(i) = sum(F(i, :)); %N
    PE_total(i) = sum(PE(i, :)); %J
    T_L(i) = r*tand(theta(i))*F_total(i); %Nm
end

thetamax = theta(T_L==max(T_L)); %degrees
PE_max = max(PE_total); %J
T_L_ozin = T_L/0.0070615518333333; %oz-in
T_L_max = max(T_L); %Nm

figure(1)
plot(h*100, F_total)
xlabel('Height (cm)','FontSize',12)
ylabel('Total spring force (N)','FontSize',12)
title('Total spring force vs. height during loading phase')
set(gcf, 'color', 'w')
set(gca,'xdir', 'reverse','FontSize',12)

figure(2)
plot(h*100, PE_total)
xlabel('Height (cm)','FontSize',12)
ylabel('PE (J)','FontSize',12)
title('PE vs. height during loading phase')
set(gcf, 'color', 'w')
set(gca,'xdir', 'reverse','FontSize',12)

figure(3)
plot(h*100, T_L)
xlabel('Height (cm)','FontSize',12)
ylabel('Load torque (Nm)','FontSize',12)
title('Load torque vs. height during loading phase')
set(gcf, 'color', 'w')
set(gca,'xdir', 'reverse','FontSize',12)
```