Chapter 4: Prototype Performance and Final Design

4.1 Frame

The frame had several requirements it needed to complete. First off the frame needed to be safe. A modal analysis was performed on the frame to identify the resonant frequencies at failure. The lowest mode identified was along the depth measurement (distance from the front of the frame to the back). It was found that the mode is at 8 Hz. Leg's generally oscillate at 2 Hz, which is four times lower than the lowest mode.

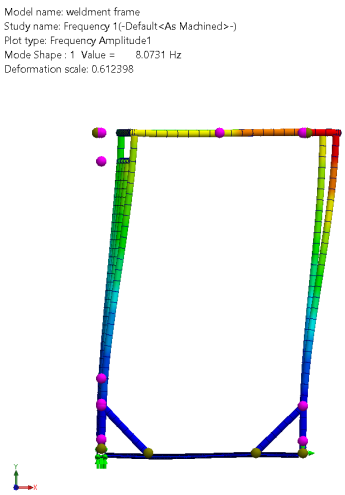


Figure 4.1.1: Modal Analysis of the frame

Mode	Natural Frequency (Hz)
1	8.0731
2	11.112
3	16.632
4	35.091
5	37.387

Table 4.1.1: Table of Modal Values

Qualitative checks were also performed to ensure the safety and stability of the frame. Within the manufacturing process, rebuilding and further addition of cross bracing took place. Rebuilding and loosening of each bar took place in order to relieve any internalized stress of the T-slotted bars. Once the frame was built and conformed to those requirements, further cross

bracing was added. This included extra corner frames and bars and the addition of two long aluminum bars to dampen lateral motion and shaking.

4.2 Force Mechanism

The force mechanism needs to satisfy the design requirements stated in Table 4.2.1. This includes the maximum displacement, weight, accuracy and cost. The finalized design can be seen in Figure 4.2.1.



Figure 4.2.1: Final Force Mechanism Prototype

The first requirement that was tested was the displacement. This was tested by measuring the distance the weight could travel from its zero position to its max position. The total displacement was just under (1m -1ft), which is around 70 cm. This is more than 10 percent above the ideal value we set out to achieve in the beginning of the project. Although this displacement is well above what we set out to achieve, it still requires tuning to get a patient in the correct center position so that they do not hit either the top or bottom endstops.

The second requirement that was tested was the ability of the weight system to hold 100N of weight (10 kg or 22 lb). This was validated by running testing while this max weight was on the device. No significant signs of wear or stress were present during the test.

The third verification to be done on the force mechanism was the validation of the accuracy of the system. We set out to have 10 percent accuracy of force application to the patient's leg, no matter where their leg was in their ROM. As seen in Fig 4.2.2, the force mechanism did not achieve this goal. It had a 30 percent error in the shank and a 20 percent error in the thigh. Although this error is higher than we want, we have an understanding of what is leading to this error and can therefore decrease this error by adjusting the spring to mass ratio. By decreasing the spring force and increasing the weight on the shanks, we can reduce the force gradient and allow for a much lower error.

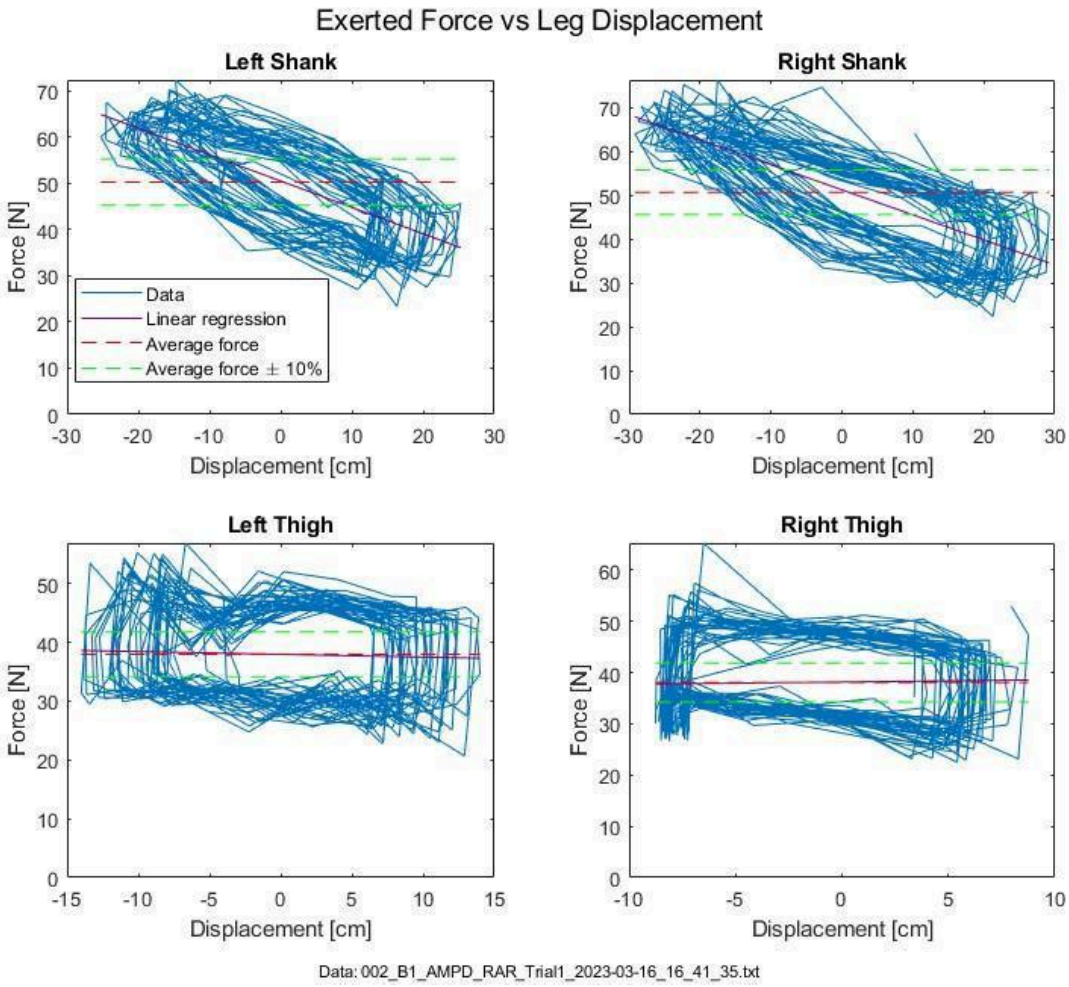


Figure 4.2.2: Force vs Displacement

Target Specifications				
No.	**Requirement Metric**	**Units**	**Marginal Value**	**Ideal Value**
1	Maximum Displacement	cm	>40	60
2	Equilibrium Force Exerted on Leg	N	30-100	20-110
3	Accuracy of Force Across ROM (% of Equilibrium Force)	%	15	10
4	Cost	$	1000	400
Compliance				
1.1	Displacement of the Weights	cm		70
2.1	Max Weight	N		100
3.1	Accuracy	%		15-30
4.1	Cost	$		805

Table 4.2.1: Compliance Table for Force Mechanism

4.3 Sensing System

Verification

Under the same set of rotary encoders (which means that the rotary encoders are going to be installed to the prototype in the constant location and constant connections), we try to do two pieces of tests about how rotary encoders will perform within the system to see if we need to adjust them: the first test is about the accuracy of the data recorded by the rotary encoders and the actual data we measured with some other tools, like rulers; the second one is about judging whether there will be errors taking place or not when the rotary encoders do work within the continuous motion, such as the error generating because of delay or insufficient time of making respond. The figure and table will be listed as follows:

ruler measure (cm)	rotary encoder (cm)	percentage error
10	7.6	24
13	13.5	38
16	17.7	10.6
20	20.7	3.5
21	18.2	13.3
25	26.1	4.4
27	28.3	4.8
30	30.5	1.7

Table 4.3.1: Accuracy test of sense system

According to the data revealed above, the percentage error generated by the rotary encoder is relatively low so that it definitely follows the initial requirements that the rotary encoder should be able to measure the data precisely.

Calibration - Method of Filtering

Due to the fact that there are noises generated to influence the reading, we see the existence of a high degree of oscillations shown on the graph, which is declared in the figure 4.3.1.

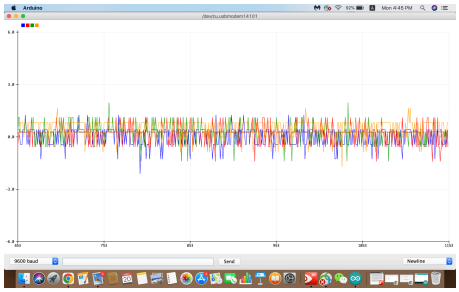


Figure 4.3.1: The Graph of Load Cell before Filt

Because such a high degree of oscillations would influence the reading deeply, leading to inaccuracy of the final results, we incorporated the method of filt to decrease the noise. To achieve this goal, we set up a number to filt parameter to be included in the codes to decrease the amplitude of oscillations. To determine the optimal choice of filt parameter, we have done a series of experiments.

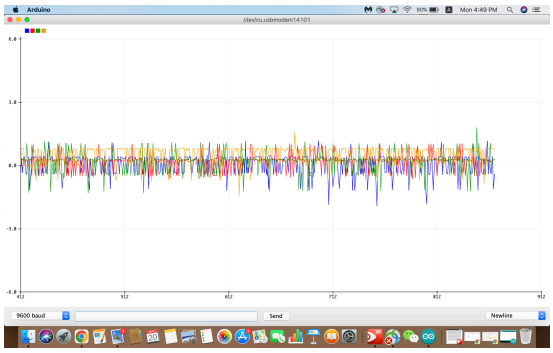


Figure 4.3.2: Filt Parameter = 0.9

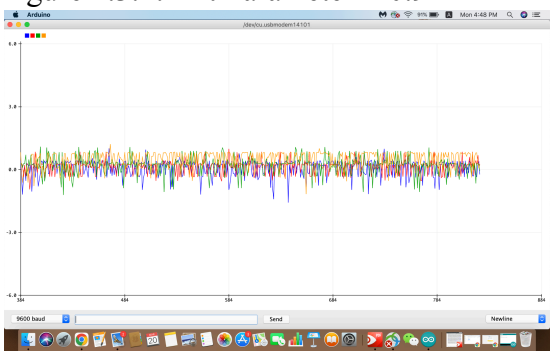


Figure 4.3.3: Filt Parameter = 0.7

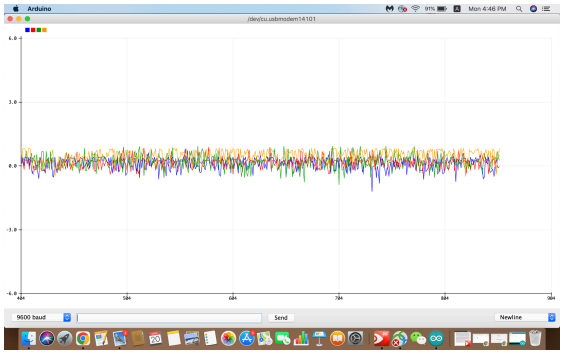


Figure 4.3.4: Filt Parameter = 0.5

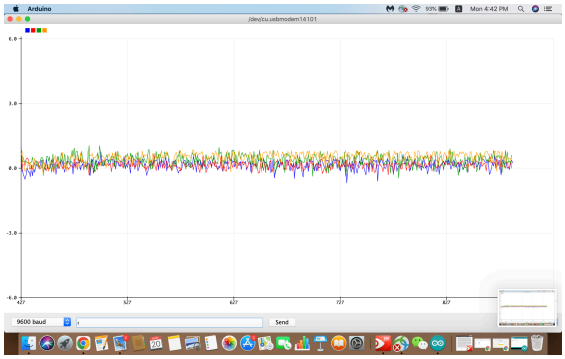


Figure 4.3.5: Filt Parameter = 0.3

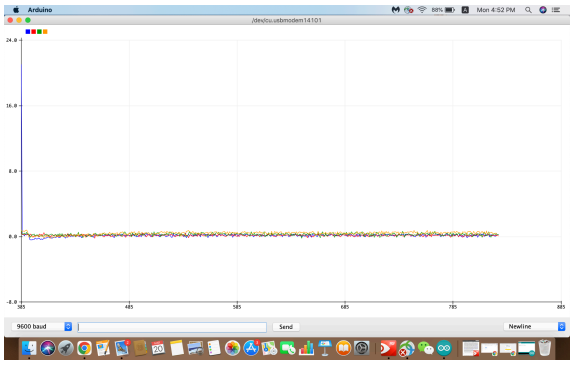


Figure 4.3.6: Filt Parameter = 0.1

Just as what five figures have revealed, as the filt parameter decreases regularly from 0.9 to 0.1, the graph approaches to be much more stable. However, after doing experiments when the parameter is 0.1 or even lower, there is a huge error appearing between the reading and actual weight, so that we determine 0.3 to be the optimal choice.

Moreover, throughout the whole experiment, we also found out that no matter what the filt parameter is, when there is no extra force exerted on the load cell, the graph is relatively stable around 0. As the initial condition, it successfully met the requirements. Next, we further test when we put weight on the load cell. With the data shown in the Serial Monitor, we found that the change of data is close to 10 in Newtons when we put the weight of 1 kg; the change of data is close to 5 in Newtons when we put the weight of 500 g. Under this circumstance, we verified our design in sense system on its accuracy.

Finalized BOM:

Item	Product #	Qt.	Manufacturer/Company	Price	Total
Grand Total			**Subtotal:**	**$ 2,610.43**	**$ 2,936.73**
Leg Brace					**$ 234.92**
M MANUEKLEAR Ankle Straps pair		1	Amazon	$ 15.99	$ 15.99
The x Bands - Cable Attachments (pair)		1	Amazon	$ 18.99	$ 18.99
Orthomen Hinged ROM Knee Brace		2	Amazon	$ 99.97	$ 199.94
Sense				**TOTAL**	**$ 290.50**
rotary encoder		2	Amazon	$ 18.99	$ 37.98
		0	Amazon		$ -
S-type load cell		0	Amazon		$ -
S-type load cell		4	Amazon	$ 30.76	$ 123.04
		0	Amazon	$ 38.82	$ -
Arduino mega		1	Amazon	$ 48.37	$ 48.37
HX711 amplifier		0	Amazon		$ -
Breadboard and wire		1	Amazon	$ 48.91	$ 48.91
INA125P amplifier		4	Mouser Electronics	$ 8.05	$ 32.20
Cable Grommets		1	Amazon	$ 11.89	$ 11.89
Box for arduino board		1	Amazon	$ 59.49	$ 59.49
Frame 1 (Single)				**TOTAL**	**$ 1,338.79**
1x1 bar	47065T101	0	McMaster	$ 27.44	$ -
1.5x1.5 bar 5ft	47065T102-47065T603	1	McMaster	$ 43.06	$ 43.06
2x1 bar 10ft	47065T107	3	McMaster	$ 76.88	$ 230.64
2x1 bar 10ft	47065T107	2	McMaster	$ 76.88	$ 153.76
2x18 diagonal brace	47065T705	2	McMaster	$ 31.96	$ 63.92
2x6 diagonal brace	47065T187	2	McMaster	$ 23.04	$ 46.08
90deg 1" 3" brace	47065T267	8	McMaster	$ 11.60	$ 92.80
1x3 Tee brace	47065T278	2	McMaster	$ 12.10	$ 24.20
Retraction Pedal on Both Sides with Pivoting Base	2728T21	2	McMaster	$ 66.28	$ 132.56
straight 2x4 brace	47065T262	0	McMaster	$ 18.10	$ -
Closed Corner Gusset	47065T216	10	McMaster	$ 6.11	$ 61.10
Linear bearing	47065T781	2	McMaster	$ 75.03	$ 150.06

1x18 diagonal brace	47065T701	6	McMaster	$ 30.08	$ 180.48
Linear Rail Breaks	6812N306	2	McMaster	$ 22.08	$ 44.16
2x1 Gussets	47065T799	4	McMaster	$ 15.20	$ 60.80
t-nuts 1/4-20	98001A130	2	McMaster	$ 7.83	$ 15.66
aluminum bar 2inx1/2inx4ft	9246K491	1	McMaster	$ 39.51	$ 39.51
Frame 2				TOTAL	1338
1.5x1.5 bar hollow 10ft			McMaster	$ 84.33	$ -
2x1 bar 10ft		8	McMaster	$ 76.88	$ 615.04
2x18 diagonal brace		2	McMaster	$ 31.96	$ 63.92
2x6 diagonal brace		2	McMaster	$ 23.04	$ 46.08
90deg 1" 3" brace		16	McMaster	$ 11.60	$ 185.60
1x3 Tee brace		2	McMaster	$ 12.10	$ 24.20
Retraction Pedal on Both Sides with Pivoting Base		4	McMaster	$ 109.09	$ 436.36
straight 2x4 brace		8	McMaster	$ 18.10	$ 144.80
Closed Corner Gusset		8	McMaster	$ 6.11	$ 48.88
Linear bearing		2	McMaster	$ 75.03	$ 150.06
Solid Casters		0	McMaster	$ 95.87	$ -
Rotating Casters		6	McMaster	$ 37.42	$ 224.52
1x18 diagonal brace		16	McMaster	$ 30.08	$ 481.28
Spring Mass System				**TOTAL**	**$ 746.22**
5 lb Weight set (x2)		2	Amazon	$ 27.99	$ 55.98
2 lb Weight set (x2)		2	Amazon	$ 18.99	$ 37.98
1 lb weight set (x4)		2	Amazon	$ 23.99	$ 47.98
Linear Rails 1m	HGH20 CA	4	Amazon	$ 83.99	$ 335.96
Springs		2	The Spring Store	$ 18.03	$ 36.06
Springs		2	The Spring Store	$ 9.25	$ 18.50
Springs		1	Amazon	$ 18.99	$ 18.99
Bearings For Encoders		1	Amazon	$ 10.99	$ 10.99
Aluminum Tubing		4	Amazon		$ -
Plate Holders		1	Amazon	$ 19.97	$ 19.97
Aluminum Bar		2	Amazon	$ 19.75	$ 39.50
Plate Holders		2	Amazon	$ 25.15	$ 50.30
Mounted D rings		2	Amazon	$ 13.99	$ 27.98
Cable Clamps		1	Amazon	$ 8.39	$ 8.39
M12x50mm hex bolt		1	Amazon	$ 9.99	$ 9.99
M12 Nuts		1	Amazon	$ 8.82	$ 8.82
M12 Washers		1	Amazon	$ 6.34	$ 6.34

M12 Eye Bolt		1	Amazon	$ 12.49	$ 12.49
Lowe's				**TOTAL**	**$ 155.37**
Bolts for linear bearing to Al plate(M5x12mm 100Pc)	91290A228	1	McMaster	$ 15.38	$ 15.38
M5 Washer 4Pc		8	Home Depot		$ -
Angle Bar for Spring Mount (1"x36"x1/8")		1	Home Depot		$ -
3/8 in. x 16 x 1 in. Zinc-Plated Grade-8 Cap Screw		8	Home Depot		$ -
3/8 in.-16 Zinc Plated Jam Nut (6-Pack)		2	Home Depot		$ -
Bolts for Linear Rails to frame (m5x16mm 100Pc)	91290A232	1	McMaster	$ 16.23	$ 16.23
T-nuts for rails (m5)	94579A860	4	McMaster	$ 5.94	$ 23.76
Solid Casters	Frame	2	Home Depot	$ 25.00	$ 50.00
Rotating Casters	Frame	2	Home Depot	$ 25.00	$ 50.00

Budget:

Accounts:	**Total**	**Spent**	**Remaining**	
UROP	$ 800.00	$ 794.03	$ 5.97	
Senior Design	$ 900.00	$ 900.00	$ -	Edited
Research Fund	$ 1,500.00	$ 1,073.16	$ 426.84	
Total	**$ 3,200.00**	**$ 2,767.19**	**$ 432.81**	
Grand Total:		**$ 2,813.57**		

Chapter 5: Design Recommendations and Conclusions

Summary of Accomplishments

Throughout the journey of the horizontal stepping robot project, several accomplishments were achieved as a consequence of the collective efforts of the team.
The rehabilitation device was improved as several design and manufacturing iterations were conducted during the time of two quarters.

From the beginning prototype version to the final version, several improvements were made for device mechanics; the structure of the frame is constructed using aluminum extrusions for better stability; the components e.g. pulley, leg braces were expanded for rehabilitation of one leg to two legs; a spring-mass system was incorporated for force adjustability.

Progresses in the sense team are achieved as well during the time of two quarters. The data of distance traveled and force exerted by the patients' leg segments, i.e. shank and thigh can be measured by rotary encoders and load cells, respectively; the circuitry of sensing components was created as well, in an organized manner; an user interface for more convenient data collection and post-processing was conducted.

Design Recommendations for the Future

On the software side, the user interface could be improved by incorporating more features, for example, recording the participant's biometric data, having real-time plotting of data in addition to the saving of data. On the hardware side, the device could be improved in regards to safety and set-up time optimization.

Lessons Learned

Having good engineering ideas are critical for the success of the project; tests are essential as well to verify the feasibility of the ideas. For example, the purpose of using a linear actuator is to support the weight of a participant's leg; however, it was discovered that the speed of participant's movement is much faster than that of linear actuator. As a result of the testing/verification, the solution of utilizing counterweights was utilized for participant's leg support.

Conclusions

In conclusion, the project of rehabilitation devices has improved to a great extent as a result of iterations of designs, with regards to stability, convenience and comfortability. Future plans shall be conducted for the purpose of improvement of the project. Credit shall be given to Dr. Walter and Dr. Copp, Chris Johnson, Piyashi Biswas, Dr. David Reinkensmeyer and Dr. Reggie Edgerton for their genuine support for the success of this project.

Appendix

List of Suppliers

1. McMaster Carr
2. Amazon
3. The Spring Store

Bill Of Materials:

*Part listings, suppliers and purchased parts are all included in this BOM

Item	Product #	Qt.	Manufacturer/Company	Price	Total
Grand Total			**Subtotal:**	**$ 2,610.43**	**$ 2,936.73**
Leg Brace					**$ 234.92**
M MANUEKLEAR Ankle Straps pair		1	Amazon	$ 15.99	$ 15.99
The x Bands - Cable Attachments (pair)		1	Amazon	$ 18.99	$ 18.99
Orthomen Hinged ROM Knee Brace		2	Amazon	$ 99.97	$ 199.94
Sense				**TOTAL**	**$ 290.50**
rotary encoder		2	Amazon	$ 18.99	$ 37.98
		0	Amazon		$ -
S-type load cell		0	Amazon		$ -
S-type load cell		4	Amazon	$ 30.76	$ 123.04
		0	Amazon	$ 38.82	$ -
Arduino mega		1	Amazon	$ 48.37	$ 48.37
HX711 amplifier		0	Amazon		$ -
Breadboard and wire		1	Amazon	$ 48.91	$ 48.91
INA125P amplifier		4	Mouser Electronics	$ 8.05	$ 32.20
Cable Grommets		1	Amazon	$ 11.89	$ 11.89
Box for arduino board		1	Amazon	$ 59.49	$ 59.49
Frame 1 (Single)				**TOTAL**	**$ 1,338.79**
1x1 bar	47065T101	0	McMaster	$ 27.44	$ -
1.5x1.5 bar 5ft	47065T102-47065T603	1	McMaster	$ 43.06	$ 43.06

2x1 bar 10ft	47065T107	3	McMaster	$ 76.88	$ 230.64
2x1 bar 10ft	47065T107	2	McMaster	$ 76.88	$ 153.76
2x18 diagonal brace	47065T705	2	McMaster	$ 31.96	$ 63.92
2x6 diagonal brace	47065T187	2	McMaster	$ 23.04	$ 46.08
90deg 1" 3" brace	47065T267	8	McMaster	$ 11.60	$ 92.80
1x3 Tee brace	47065T278	2	McMaster	$ 12.10	$ 24.20
Retraction Pedal on Both Sides with Pivoting Base	2728T21	2	McMaster	$ 66.28	$ 132.56
straight 2x4 brace	47065T262	0	McMaster	$ 18.10	$ -
Closed Corner Gusset	47065T216	10	McMaster	$ 6.11	$ 61.10
Linear bearing	47065T781	2	McMaster	$ 75.03	$ 150.06
1x18 diagonal brace	47065T701	6	McMaster	$ 30.08	$ 180.48
Linear Rail Breaks	6812N306	2	McMaster	$ 22.08	$ 44.16
2x1 Gussets	47065T799	4	McMaster	$ 15.20	$ 60.80
t-nuts 1/4-20	98001A130	2	McMaster	$ 7.83	$ 15.66
aluminum bar 2inx1/2inx4ft	9246K491	1	McMaster	$ 39.51	$ 39.51
Frame 2				TOTAL	1338
1.5x1.5 bar hollow 10ft			McMaster	$ 84.33	$ -
2x1 bar 10ft		8	McMaster	$ 76.88	$ 615.04
2x18 diagonal brace		2	McMaster	$ 31.96	$ 63.92
2x6 diagonal brace		2	McMaster	$ 23.04	$ 46.08
90deg 1" 3" brace		16	McMaster	$ 11.60	$ 185.60
1x3 Tee brace		2	McMaster	$ 12.10	$ 24.20
Retraction Pedal on Both Sides with Pivoting Base		4	McMaster	$ 109.09	$ 436.36
straight 2x4 brace		8	McMaster	$ 18.10	$ 144.80
Closed Corner Gusset		8	McMaster	$ 6.11	$ 48.88
Linear bearing		2	McMaster	$ 75.03	$ 150.06
Solid Casters		0	McMaster	$ 95.87	$ -
Rotating Casters		6	McMaster	$ 37.42	$ 224.52
1x18 diagonal brace		16	McMaster	$ 30.08	$ 481.28
Spring Mass System				**TOTAL**	**$ 746.22**
5 lb Weight set (x2)		2	Amazon	$ 27.99	$ 55.98
2 lb Weight set (x2)		2	Amazon	$ 18.99	$ 37.98
1 lb weight set (x4)		2	Amazon	$ 23.99	$ 47.98
Linear Rails 1m	HGH20 CA	4	Amazon	$ 83.99	$ 335.96
Springs		2	The Spring Store	$ 18.03	$ 36.06
Springs		2	The Spring Store	$ 9.25	$ 18.50
Springs		1	Amazon	$ 18.99	$ 18.99

Bearings For Encoders		1	Amazon	$ 10.99	$ 10.99
Aluminum Tubing		4	Amazon		$ -
Plate Holders		1	Amazon	$ 19.97	$ 19.97
Aluminum Bar		2	Amazon	$ 19.75	$ 39.50
Plate Holders		2	Amazon	$ 25.15	$ 50.30
Mounted D rings		2	Amazon	$ 13.99	$ 27.98
Cable Clamps		1	Amazon	$ 8.39	$ 8.39
M12x50mm hex bolt		1	Amazon	$ 9.99	$ 9.99
M12 Nuts		1	Amazon	$ 8.82	$ 8.82
M12 Washers		1	Amazon	$ 6.34	$ 6.34
M12 Eye Bolt		1	Amazon	$ 12.49	$ 12.49
Lowe's				**TOTAL**	**$ 155.37**
Bolts for linear bearing to Al plate(M5x12mm 100Pc)	91290A228	1	McMaster	$ 15.38	$ 15.38
M5 Washer 4Pc		8	Home Depot		$ -
Angle Bar for Spring Mount (1"x36"x1/8")		1	Home Depot		$ -
3/8 in. x 16 x 1 in. Zinc-Plated Grade-8 Cap Screw		8	Home Depot		$ -
3/8 in.-16 Zinc Plated Jam Nut (6-Pack)		2	Home Depot		$ -
Bolts for Linear Rails to frame (m5x16mm 100Pc)	91290A232	1	McMaster	$ 16.23	$ 16.23
T-nuts for rails (m5)	94579A860	4	McMaster	$ 5.94	$ 23.76
Solid Casters	Frame	2	Home Depot	$ 25.00	$ 50.00
Rotating Casters	Frame	2	Home Depot	$ 25.00	$ 50.00

Budget:

Accounts:	**Total**	**Spent**	**Remaining**	
UROP	$ 800.00	$ 794.03	$ 5.97	
Senior Design	$ 900.00	$ 900.00	$ -	Edited
Research Fund	$ 1,500.00	$ 1,073.16	$ 426.84	
Total	**$ 3,200.00**	**$ 2,767.19**	**$ 432.81**	
Grand Total:		**$ 2,813.57**		

MATLAB Post-processing code for angle calculation:

```
clear;

clc;

clear;

clc;

%% Data input

%Input of encoder data from txt file

total_data = fileread('006_fw.txt');

% encoder_angles = splitlines(encoder_angles);

total_data = str2double(split(total_data));

total_data(length(total_data))= [];

a = length(total_data)/4;

encoder_angles = zeros(a,4);

counter = 1;

for y = 1:a

   for x = 1:4

       encoder_angles(x,y) = total_data(counter);

       counter = counter + 1;

   end

end

encoder_angles = encoder_angles';

encoder_angles(:,5) = [];

encoder_angles = lta(encoder_angles);

encoder_angles = mod(encoder_angles,90);

%% Angle derivation and calculation

l1 = 0.4572; % Lee's thigh length, in m

l2 = 0.3937; % Lee's cakf length, in m

r1 = 1.15; r2 = 1.15; % Initial lengths of ropes connected to thigh and calf, in m

r3 = 1.15; r4 = 1.15; % Initial lengths of ropes connected to thigh and calf, in m

r1_enca = encoder_angles(:,1);

r2_enca = encoder_angles(:,2); % enca would stand for encoder angles

r1_m = r1+atl(r1_enca); % length of rope1 in motion, subject to change by measuring rotations of
pulley using rotary encoder, it is theoretical here

r1_m = fliplr(r1_m);

r2_m = r2+atl(r2_enca); % length of rope2 in motion, subject to change by measuring rotations of
pulley using rotary encoder, it is theoretical here

r2_m = fliplr(r2_m);

r3_enca = encoder_angles(:,3);

r4_enca = encoder_angles(:,4); % enca would stand for encoder angles

r3_m = r3+atl(r3_enca); % length of rope1 in motion, subject to change by measuring rotations of
pulley using rotary encoder, it is theoretical here

r3_m = fliplr(r3_m);

r4_m = r4+atl(r4_enca); % length of rope2 in motion, subject to change by measuring rotations of
pulley using rotary encoder, it is theoretical here
```

```
r4_m = fliplr(r4_m);
px1 = 0.3;
py1 = 1.15;% contact coordinates with pulley 1
px2 = px1 + 0.51;
py2 = 1.15;% contact coordinates with pulley 2
b1 = 0.55*l1; % distance from origin to thigh connection point
c1 = 0.45*l2; % distance from origin to calf connection point
d = px1^2+py1^2+b1^2-r1_m.^2; % newly introduced variable d, helps with theta angle calculation
theta = atan(py1/px1)+acos(d/(2*b1*sqrt(px1^2+py1^2))); % knee flexion calculation
theta = theta/pi*180;
theta = abs(theta-90);
% shift of coordinate system, then apply the same algorithm for knee
% flexion angle calculation
px2_new = px2 - sin(theta)*l1;
py2_new = py2 - cos(theta)*l1; % contact coordinates with pulley 2 under new system
d2 = r2_m.^2-(px2_new.^2 + py2_new.^2+c1^2);% newly introduced variable d2, helps with theta
angle calculation
% knee flexion calculation
gamma = zeros(length(d2),1);
for m = 1: length(d2)
   gamma(m)= atan(py2_new(m)/px2_new(m))+acos(d2(m)/(2*c1*sqrt(px2_new(m)^2+py2_new(m)^2)));
end
% gamma = atan(py2_new./px2_new)+acos(d2/(2*c1*sqrt(px2_new.^2+py2_new.^2)))*0;
gamma = real(gamma);
alpha = theta + gamma;
px3 = 0.3;
py3 = 1.15;% contact coordinates with pulley 1
px4 = px3 + 0.51;
py4 = 1.15;% contact coordinates with pulley 2
b2 = 0.55*l1; % distance from origin to thigh connection point
c2 = 0.45*l2; % distance from origin to calf connection point
d3 = px3^2+py3^2+b2^2-r3_m.^2; % newly introduced variable d, helps with theta angle calculation
theta2 = atan(py3/px3)+acos(d3/(2*b2*sqrt(px3^2+py3^2))); % knee flexion calculation
theta2 = theta2/pi*180;
theta2 = abs(theta2-90);
% shift of coordinate system, then apply the same algorithm for knee
% flexion angle calculation
px4_new = px4 - sin(theta2)*l1;
py4_new = py4 - cos(theta2)*l1; % contact coordinates with pulley 2 under new system
d4 = r4_m.^2-(px4_new.^2 + py4_new.^2+c2^2);% newly introduced variable d2, helps with theta
angle calculation
% knee flexion calculation
gamma2 = zeros(length(d4),1);
for m = 1: length(d4)
```

```
    gamma2(m)= atan(py4_new(m)/px4_new(m))+acos(d4(m)/(2*c2*sqrt(px4_new(m)^2+py4_new(m)^2)));
end
% gamma = atan(py2_new./px2_new)+acos(d2/(2*c1*sqrt(px2_new.^2+py2_new.^2)))*0;
gamma2 = real(gamma2);
alpha2 = theta2 + gamma2;
t = [1:length(r1_enca)];
t = t';
%% Save of flexion angle data
flexion_angle_data = zeros(length(theta2),4);
% theta = mod(theta,45);
% alpha = mod(alpha,45);
% theta2 = mod(theta2,45);
% alpha2 = mod(alpha2,45);
flexion_angle_data(:,1)= theta;
flexion_angle_data(:,2)= alpha;
flexion_angle_data(:,3)= theta2;
flexion_angle_data(:,4)= alpha2;
%% Plot of data
subplot(2,2,1);
plot(t,theta);
title('Left leg: thigh support and thigh flexion test 1');
xlabel('time (ms)');
ylabel('thigh flexion angle (degrees)');
subplot(2,2,2);
plot(t,alpha);
title('Left leg: shank support and shank flexion test 1');
xlabel('time (ms)');
ylabel('knee flexion angle (degrees)');
subplot(2,2,3);
plot(t,theta2);
title('Right leg: thigh support and thigh flexion test 1');
xlabel('time (ms)');
ylabel('thigh flexion angle (degrees)');
subplot(2,2,4);
plot(t,alpha2);
title('Right leg: shank support and shank flexion test 1');
xlabel('time (ms)');
ylabel('knee flexion angle (degrees)');
%% Functions for distance and angle conversion
function l = atl(theta) % The atl stands for angle to length that will translate
% the change in rotational angle to change in rope length
r = 0.27; % radius of shaft of rotary encoder
theta = theta/180*pi;
```

```
l = r * theta;
end
function theta = lta(length) % The atl stands for angle to length that will translate
% the change in rotational angle to change in rope length
r = 2; % radius of shaft of rotary encoder, in cm
theta = length/(r);
end
```

Arduino Code for Sensors

```
volatile unsigned int temp, counter1, counter2, counter3, counter4=0;
float maxRev=600; float maxAngle=360;
float angle1=0; float angle2=0; float angle3=0; float angle4=0;


//rotary encoder
float distance1;
float distance2;
float distance3;
float distance4;


//first load cell
float load_cell1_val_old;
float load_cell1_val_filt;
float calibration1 = 2.695;
float load_cell1_val1;
float load_cell1_val2;
float load_cell1_val;
float force1;

//second load cell
float load_cell2_val_old;
float load_cell2_val_filt;
float calibration2 = -14.166;
float load_cell2_val1;
float load_cell2_val2;
float load_cell2_val;
float force2;

//third
float load_cell3_val_old;
float load_cell3_val_filt;
float calibration3 = -5.019;
float load_cell3_val1;
float load_cell3_val2;
float load_cell3_val;
float force3;

//fourth
float load_cell4_val_old;
float load_cell4_val_filt;
float calibration4 = -5.548;
float load_cell4_val1;
float load_cell4_val2;
float load_cell4_val;
float force4;


float filt_strength = 0.3;
```

```
/// <summary>
/// Start of the test to calculate time elapsed
/// </summary>
unsigned long time_start = 0;


/// <summary>
/// Time elapsed since start of test
/// </summary>
unsigned long time_current = 0;

/// <summary>
/// Current time elapsed in millis() since the start of code
/// </summary>
unsigned long time_now = 0;

/// <summary>
/// time in millis() last time since data was saved in a loop
/// </summary>
unsigned long time_last = 0;


void save_data();

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  //for encoder 1
  pinMode(2,INPUT_PULLUP);
  pinMode(4,INPUT_PULLUP);
  //encoder 2
  pinMode(3,INPUT_PULLUP);
  pinMode(10,INPUT_PULLUP);
  //encoder 3
  pinMode(21,INPUT_PULLUP);
  pinMode(17,INPUT_PULLUP);
  //encoder 4
  pinMode(20,INPUT_PULLUP);
  pinMode(16,INPUT_PULLUP);

  //set up interrupt
  //encoder 1
  attachInterrupt(digitalPinToInterrupt(2), ai0, RISING);
  attachInterrupt(digitalPinToInterrupt(4), ai1, RISING);
  //encoder 2
  attachInterrupt(digitalPinToInterrupt(3), ai2, RISING);
  attachInterrupt(digitalPinToInterrupt(10), ai3, RISING);
  //encoder 3
  attachInterrupt(digitalPinToInterrupt(21), ai4, RISING);
  attachInterrupt(digitalPinToInterrupt(17), ai5, RISING);
  //encoder 4
  attachInterrupt(digitalPinToInterrupt(20), ai6, RISING);
  attachInterrupt(digitalPinToInterrupt(16), ai7, RISING);
}

void loop() {
  // put your main code here, to run repeatedly:

  //ai8();  //reset switch

  //angle measurement for encoder 1
  angle1=counter1 / maxRev * maxAngle;
  if (angle1 > 15000) {
    angle1 = -1 * (39321 - angle1);
  }
  angle1 = angle1 * -1;

  //angle measurment for encoder 2
  angle2 = counter2 / maxRev * maxAngle;
  if (angle2 > 15000) {
```

```
  angle2 = -1 * (39321 - angle2);
}
//angle2 = -1 * angle2;

//angle measurement for encoder 3
angle3 = counter3 / maxRev * maxAngle;
if (angle3 > 15000) {
  angle3 = -1 * (39321 - angle3);
}
angle3 = angle3 * -1;

//angle measurement for encoder 4
angle4 = counter4 / maxRev * maxAngle;
if (angle4 > 15000) {
  angle4 = -1 * (39321 - angle4);
}
//angle4 = -1 * angle4;

//distance converting
distance1 = (1.3462+0.2)*2*3.1415926*angle1/360;
distance2 = (1.3462+0.2)*2*3.1415926*angle2/360;
distance3 = (1.3462+0.2)*2*3.1415926*angle3/360;
distance4 = (1.3462+0.2)*2*3.1415926*angle4/360;


//first load cell reading
load_cell1_val1=analogRead(A1);
load_cell1_val2=analogRead(A0);
//second load cell reading
load_cell2_val1=analogRead(A3);
load_cell2_val2=analogRead(A2);
//third load cell reading
load_cell3_val1=analogRead(A9);
load_cell3_val2=analogRead(A6);
//fourth load cell reading
load_cell4_val1=analogRead(A12);
load_cell4_val2=analogRead(A11);

load_cell1_val = load_cell1_val2 - load_cell1_val1;
load_cell2_val = load_cell2_val2 - load_cell2_val1;
load_cell3_val = load_cell3_val2 - load_cell3_val1;
load_cell4_val = load_cell4_val2 - load_cell4_val1;

//filt
load_cell1_val_filt = filt_strength * load_cell1_val + (1 - filt_strength) * load_cell1_val_old;
load_cell1_val_old = load_cell1_val_filt;

load_cell2_val_filt = filt_strength * load_cell2_val + (1 - filt_strength) * load_cell2_val_old;
load_cell2_val_old = load_cell2_val_filt;

load_cell3_val_filt = filt_strength * load_cell3_val + (1 - filt_strength) * load_cell3_val_old;
load_cell3_val_old = load_cell3_val_filt;

load_cell4_val_filt = filt_strength * load_cell4_val + (1 - filt_strength) * load_cell4_val_old;
load_cell4_val_old = load_cell4_val_filt;


force1 = (load_cell1_val_filt * 85.1 + 466) * -9.8/1000 - calibration1;
force2 = (load_cell2_val_filt * 85.1 + 466) * -9.8/1000 - calibration2;
force3 = (load_cell3_val_filt * 85.1 + 466) * -9.8/1000 - calibration3;
force4 = (load_cell4_val_filt * 85.1 + 466) * -9.8/1000 - calibration4;

//output in the Serial monitor
//Serial.print(distance1);
//Serial.print("\t");
//Serial.print(distance2);
//Serial.print("\t");
//Serial.print(distance3);
//Serial.print("\t");
//Serial.print(distance4); //distance in cm
```

```
  //Serial.print("\t");
//  Serial.print(force1);
//  Serial.print("\t");
//  Serial.print(force2);
//  Serial.print("\t");
//  Serial.print(force3);
//  Serial.print("\t");
//  Serial.println(force4); //force in N
//  delay(10);
  save_data();
}

//ai0 to ai7 for encoders and ai8 for reset switch
void ai0() {
  //ai0 is activated if reading on 2 changes from LOW to HIGH
  //check pin 4 reading
  if(digitalRead(4)==LOW) {
    counter1++;
  }else {
    counter1--;
  }
}

void ai1() {
  //ai1 is activated if reading on 4 changes from LOW to HIGH
  //check pin 2 reading
  if(digitalRead(2)==LOW) {
    counter1--;
  }else {
    counter1++;
  }
}

void ai2() {
  //ai2 is activated if reading on 3 changes from LOW to HIGH
  //check pin 10 reading
  if(digitalRead(10)==LOW) {
    counter2++;
  }else {
    counter2--;
  }
}

void ai3() {
  //ai2 is activated if reading on 5 changes from LOW to HIGH
  //check pin 3 reading
  if(digitalRead(3)==LOW) {
    counter2--;
  }else {
    counter2++;
  }
}

void ai4() {
  //ai2 is activated if reading on 21 changes from LOW to HIGH
  //check pin 17 reading
  if(digitalRead(17)==LOW) {
    counter3++;
  }else {
    counter3--;
  }
}

void ai5() {
  //ai2 is activated if reading on 17 changes from LOW to HIGH
  //check pin 21 reading
  if(digitalRead(21)==LOW) {
    counter3--;
  }else {
    counter3++;
```

```
  }
}

void ai6() {
  //ai2 is activated if reading on 20 changes from LOW to HIGH
  //check pin 16 reading
  if(digitalRead(16)==LOW) {
    counter4++;
  }else {
    counter4--;
  }
}

void ai7() {
  //ai2 is activated if reading on 16 changes from LOW to HIGH
  //check pin 20 reading
  if(digitalRead(20)==LOW) {
    counter4--;
  }else {
    counter4++;
  }
}

//void ai8() {
  //if(digitalRead(6)==1) {
    //counter1=0;
    //counter2=0;
    //counter3=0;
    //counter4=0;
  //}
//}


void save_data() {
  time_now = millis();

  if ((time_now - time_last) >= 100) {
    time_current = time_now - time_start;
    time_last = time_now;



    Serial.print(distance1);
    Serial.print("\t");
    Serial.print(distance2);
    Serial.print("\t");
    Serial.print(distance3);
    Serial.print("\t");
    Serial.print(distance4);
    Serial.print("\t");
    Serial.print(force1);
    Serial.print("\t");
    Serial.print(force2);
    Serial.print("\t");
    Serial.print(force3);
    Serial.print("\t");
    Serial.print(force4);
    Serial.print("\t");
    Serial.print("\n");


  }
}
```

Data and Results

Matlab code for Force vs Displacement Graphs:

```
% Read the txt file
```

```
filename = '004_B1_AMPD_RAR_Trial1_2023-03-20_13_23_37.txt';
data = load(filename);
% Separate displacement and force data into their respective matrices
displacement = data(:, 1:4);
force = data(:, 5:8);
% Create a 2x2 subplot for the four graphs
figure;
% Graph labels
labels = {'Left Shank', 'Left Thigh', 'Right Thigh','Right Shank'};
subplot_idx = [1, 3, 4, 2];
% Plot the force vs displacement data for each pair and label the graphs
for i = 1:4
   subplot(2, 2, subplot_idx(i));

   % Shift the x-axis so the zero is at the center of the min and max
displacements
   x_centered = displacement(:, i) - mean([min(displacement(:, i)),
max(displacement(:, i))]);

   % Calculate a linear regression for the data and plot the line in red
   p = polyfit(x_centered, force(:, i), 1);
   x_fit = linspace(min(x_centered), max(x_centered), 100);
   y_fit = polyval(p, x_fit);

   % Plot the original data as a line in standard MATLAB plot blue
   plot(x_centered, force(:, i), 'Color', [0, 0.4470, 0.7410], 'DisplayName',
'Data');
   hold on;

% Plot the linear regression
plot(x_fit, y_fit, 'Color', [0.5, 0, 0.5], 'DisplayName', 'Linear regression');

   xlabel(['Displacement [cm]']);
   ylabel(['Force [N]']);
   title(labels{i});
   ylim([0, max(force(:, i))]);

   % Calculate the average force and draw the horizontal lines
   avg_force = mean(force(:, i));
   x_range = [min(x_centered), max(x_centered)];
   draw_horizontal_lines(x_range, avg_force);

   % Add a legend for the first subplot
   if i == 1
       legend('Location', 'best');
   end
end
% Add a title "Exerted Force vs Leg Displacement"
sgtitle('Exerted Force vs Leg Displacement');
% Add a note at the bottom of the graph with the filename used for the data,
ensuring underscores are not interpreted as subscripts
annotation('textbox', [0.5, 0.03, 0, 0], 'String', ['Data: ', strrep(filename,
'_', '\_')], 'FitBoxToText', 'on', 'EdgeColor', 'none', 'HorizontalAlignment',
'center', 'FontSize', 0.5 * get(gca, 'FontSize'));
% Define a function to draw horizontal lines for average force and 10%
deviation
function draw_horizontal_lines(x_range, avg_force)
```

```
    line(x_range, [avg_force, avg_force], 'Color', 'red', 'LineStyle', '--',
'DisplayName', 'Average force');
    line(x_range, [avg_force * 1.1, avg_force * 1.1], 'Color', 'green',
'LineStyle', '--', 'DisplayName', 'Average force \pm 10%');
    line(x_range, [avg_force * 0.9, avg_force * 0.9], 'Color', 'green',
'LineStyle', '--', 'HandleVisibility', 'off');
end
```

Raw Data

Left Encoder	Right Encoder	Left Force(mV)	Left Force Sensitivity	Right Force	Right Force Sensitivity	Left PWM	Right PWM
41.03	38.46	32.12	17.99	40.01	44.54	51.17	63.83
39.17	36.77	33.29	22.91	40.93	43.7	56.98	63.25
35.61	34.8	35.65	30.33	41.15	41.26	58.98	61.55
30.83	32.17	39.46	39.28	40.43	39.79	62.15	61.62
24.85	28.93	44.12	49	42.22	41.9	62.74	58.2
18.69	26.26	48.58	58.11	46.14	42.57	61.14	57.04
13.42	24.79	52.07	65.75	45.34	41.75	61.73	53.26
8.91	24.51	53.81	70.63	48.08	42.12	61.04	47.42
6.65	24.58	53.71	71.55	51.22	52.75	38.08	47.24
6.61	25.39	51.64	69.64	56.26	51.74	49.4	40.15
8.06	26.85	48.45	64.35	59.97	50.7	43.11	42.07
11.82	28.77	43.98	56.4	58.49	55.55	45.34	39.16
15.41	31.72	39.43	46.86	62.05	55.81	45.13	41.46
21.13	34.99	35.31	36.58	57.53	56.41	45.55	46.29
26.99	38.2	32.16	26.94	58.82	56.16	44.91	48.16
32.56	40.3	30.67	19.83	51.13	53.63	42.51	56.4
37	40.63	30.29	16.5	46.51	48.44	47.38	59.28

38.89	40.29	30.59	16.94	53.02	39.63	57	61.73
38.93	38.34	32.16	19.72	39.88	44.42	52.01	63.25
36.87	35.88	34.25	25.37	40.49	42.52	57.12	66.21
33.05	32.95	37.31	33.47	38.05	43.13	59.15	61.35
28.17	29.99	41.03	42.75	41.43	40.86	63.28	57.58
22.47	27.33	45.21	52.23	45.67	40.76	62.49	57.1
16.97	25.18	49.14	61.09	43.65	42.93	61.06	52.81
12.53	24.21	52.45	68.04	45.14	42.22	61.37	53.55
9.18	24.21	54.21	72.26	46.61	49.81	62.9	48.49
7.46	24.72	54.32	72.78	48.43	51.66	43.29	48.93
7.35	25.76	53.3	71.02	54.5	50.47	49.42	45.11
8.55	26.99	51.15	66.87	58.6	51.1	45.43	40.81
11.58	28.76	47.72	60.51	59.58	56.44	42.85	42.82
15.25	31.54	43.36	52.41	57.7	54.86	44.56	41.44
20.71	34.47	39.05	43.13	56.36	54.82	45.14	46.48
25.89	37.03	35.46	34.23	56.16	56.77	47.07	47.81
31.27	38.96	33.26	26.31	57.06	54.4	45.91	52.43
35.59	39.39	32.69	20.84	52.09	51.6	41.47	54.88
38.03	39.23	32.68	18.41	52.41	39.91	43.36	55.7
38.47	37.45	32.71	18.86	43.88	39.84	52.56	61.55
37.05	34.21	32.97	21.66	35.29	43.83	56.34	62.41
32.92	30.81	34.29	27.01	40.01	39.44	54.46	61.24
27.53	27.43	36.64	34.33	40.19	41.32	58.03	60.85
21.63	24.79	40.04	42.68	44.1	40.84	60.15	58.03
15.56	23.19	43.8	51.05	50.15	42.19	61.71	54.38

11.67	22.91	47.28	58.5	51.15	44.37	60.54	54.35
8.57	23.12	49.66	64.05	49.06	54.18	60.03	50.05
7.63	24.55	50.28	66.45	50.31	49.93	54.74	47.17
8.37	26.3	49.69	65.93	60.9	49.43	38.72	43.65
11.3	27.91	47.15	62.87	60.71	53.5	44.29	35.76
14.51	30.36	43.7	56.41	63.58	58.06	43.22	41.49
20.16	33.91	39.57	48.09	61.31	57.83	44	40.74
25.37	37.71	35.82	38.83	57.6	57.72	45.51	46.76
32.14	41.08	32.93	29.91	55.92	56.05	44.5	49.75
36.93	43.07	31.41	22.73	53.87	55.36	43.47	53
40.98	43.41	31.17	18.7	48.23	52.42	45.3	55.08
42.28	42.94	31.31	18.22	53.49	39.21	57.57	58.48
41.9	40.35	32.74	20.19	36.35	44.41	51.19	62.54
38.81	37.24	34.55	24.9	42.4	39.93	56.62	64.77
34.89	33.13	37.1	32.04	40.25	44.63	58.33	62.69
29.26	29.03	40.33	40.69	41.34	42.66	64.13	59.86
22.86	25.4	44.25	49.9	46.74	40.38	62.86	59.77
16.52	22.6	47.9	58.89	45.8	42.78	62.73	55.01
11.61	21.44	50.6	65.98	54.4	42.41	61.3	51.01
8.01	21.44	51.39	70.11	49.56	49.9	55.57	46.12
6.99	22.62	50.81	70.48	48.33	55.32	36.88	48.79
7.55	25.6	48.01	68.54	63.36	49.64	45.5	37.51
10.59	28.25	44.09	62.92	59.44	51.88	42.68	40.82
14.07	30.67	39.27	54.53	61.63	56.8	44.26	41.38
19.43	33.52	35.12	44.72	60.42	57.34	46.28	45.24

24.81	36.43	32.35	34.55	55.35	56.97	44.88	49.42
30.62	38.97	30.99	25.76	58.13	54.43	42.01	53.15
35.54	39.86	30.49	19.8	53.52	54.81	40.12	56.46
39.14	39.83	30.51	17.89	52.6	43.54	55.28	61.59
40.92	39.27	32.37	19.11	53.13	42.94	51.69	53.15
41	37.69	34.49	22.13	40.23	41.12	49.57	67.3
38.96	35.31	35.64	28.35	37.53	43.05	57.51	61.05
34.91	32.61	37.95	36.42	39.91	40.07	61.95	62.23
30.2	29.39	41.58	45.52	37.55	42.67	64.86	62.44
23.83	26.52	45.94	55.08	47.63	39.77	62.98	57.83
18.43	24.14	50.19	63.91	42.68	43.6	61.7	56.37
13.67	23.3	53.43	70.82	50.56	40.49	59.76	49.91
9.97	23.3	54.44	74.45	44.68	45.85	54.2	45.59
8.26	23.72	53.5	74.26	51.2	52.58	41.9	46.64
8.39	25.26	50.36	71.45	59.36	51.01	44.23	38.44
10.14	27.19	45.87	65.33	58.21	51.15	45.21	44.74
13.23	29.29	40.66	57.11	56.47	56.4	44.41	39.86
17.1	32.01	35.86	47.17	57.71	53.41	47.38	44.49
22.13	34.5	32.61	37.13	54.64	54.59	44.88	50.26
26.8	36.32	31.15	29.03	54.5	54.07	42.67	52.71
31.54	37.05	30.83	23.66	56.24	54.16	39.21	52.16
35.04	37.03	30.83	21.29	54.6	45.67	42.94	56.26
37.81	36.9	31.9	22.2	54.91	40.94	63.42	56.58
39.35	36.29	35.12	24.68	50.89	41.69	49.42	66.17
39.27	34.72	37.03	29.39	38.49	40.4	57.61	63.83

Team organization

Major Task	Sub-Task	Task	Assigned To	Priority	Status	Deadline
SEA Redesign					In Progress	
		Define Requirements	**All**	1	In Progress	2/3/23
		Order Components	**All**	2	Not Started	2/8/23
		Build Prototype	**All**	2	Not Started	2/17/23
		Validate Prototype	**All**	2	Not Started	2/24/23
	Linear Actuator				In Progress	
		Research Actuator Speed Needed	**Kohl**		Complete	1/20/23
		Research Possible solutions	**Kohl**	1	In Progress	1/27/23
		Detailed Design	**Kohl**	1	Not Started	2/3/23
		Order Linear Motion System Components	**Kohl**	1	Not Started	2/3/23
	Frame Redesign				Not Started	
		Analyze forces on rear frame	**Lee**	1	Not Started	2/3/23
	Pulley Redesign				Not Started	2/8/23
		Look for off the shelf pulleys	**Lee**	3	Not Started	2/8/23
		Redesign Pulleys?	**Lee**	3	Not Started	2/8/23
	Electrical Redesign				Not Started	2/8/23
		Spec Microcontroller	**Andrew**	1	Not Started	2/4/23
		Purchase Microcontroller	**Andrew**	2	Not Started	2/8/23
		Spec Force Sensors	**Andrew**	1	Not Started	2/4/23
		Purchase Force Sensors	**Andrew**	2	Not Started	2/8/23
		Assemble Prototype	**Andrew**	2	Not Started	2/17/23
		Validate	**Andrew**	2	Not Started	3/10/23
	Code Redesign				Not Started	
		Connect Arduino to Matlab	**Conghao and Thomas**	1	Not Started	2/20/23
		Integrate PID control	**Conghao**	2	Not Started	2/20/23
		Generalize code to work with multiple actuators	**Conghao**	1	Not Started	3/3/23
		Make code run at a specified loop speed	**Conghao**	1	Not Started	3/3/23
		Test code	**Conghao**	2	Not Started	3/10/23
Leg Brace					Not Started	
		Define Requirments	**Leo**	1	Not Started	2/1/23
		Conceptual Design	**Leo**	1	Not Started	2/1/23
		Detailed Design	**Leo**	2	Not Started	2/8/23
		Materials List Prepared	**Leo**	2	Not Started	2/8/23
		Test Prototype	**Leo**	3	Not Started	2/20/23
		Purchase Further Materials For full Desgin	**Leo**	3	Not Started	3/1/23
Frame Redesign					Not Started	
		Material Analysis	**Lee**	2	Not Started	2/17/23
		Design Frame and Subsystems (CAD)	**Lee**	2	Not Started	2/17/23
		Material Analysis (Forces)	**Lee**	2	Not Started	2/17/23
		Purchase Order for Frame Materials	**Lee**	2	Not Started	3/1/23
		Assemble Frame	**Lee**	2	Not Started	3/8/23
		Test Frame	**Lee**	2	Not Started	3/8/23
Post Processing Algo					In Progress	
		Calculate Eqn to relate distance to angle of leg	**Thomas**	1	In Progress	
		Integrate eqn into matlab	**Thomas**	2	Not Started	
		Create matlab GUI	**Thomas**	3	In Progress	
		Integrate user input into matlab GUI?	**Thomas**	3	In Progress	

Midterm and Final Presentations