Mobile Backpack Carrier

Project Duration Dates

September 3rd 2013 - December 15th 2013

Final Report Submission Date

December 10th 2013

Prepared by Team 64

Fangzhou Xia

Nolan Carbeck

Xingjian Lai

Zachary Zimmerman

Undergraduate Engineers, University of Michigan College of Engineering

Prepared For

Prof. Jeffery L. Stein

ME 350 Instructor Department of Mechanical Engineering

University of Michigan College of Engineering

Dr. Mike Umbriac

ME 350 Instructor Department of Mechanical Engineering

University of Michigan College of Engineering

Executive Summary

The goal for the project is to design and prototype a mobile backpack carrier that allows the occupant of a wheelchair to have access to their backpack while being able to transport easily. Specifications regarding the ideal design parameters were given at the beginning of project and the design was adjusted to best meet these goals. A variety of software was used to model and simulate the design for analysis and optimization before the manufacturing process. After finishing the design, a prototype was machined and then programmed with a microcontroller to meet the specific motion requirement.

The initial focus of the design was on the motion generation of the device. Among the four design proposal from all team members, the selected linkage mechanism was designed to have a compact starting position at the back of the wheelchair as well as a final position that was as close to the ideal parameter value as stated in the project description. A software called Lincages was utilized for the motion generation of the four-bar linkage; this provided a starting point for the mechanism design. The rough linkage system with only relative link length was then modeled in Solidworks using the real dimension and material. The force response on the device due to the weight of itself and the backpack were analyzed using a program called ADAMS. The design procedure went into another iteration to optimize the parameters obtained from the previous analysis. The focus was to reduce friction torque and force in the joints of the device without causing significant parameter deviation from the ideal parameter given. The mobile backpack carrier was also designed for easy attachment onto the wheel chair. By using only four screws threaded in to the adapter, it took less than three minutes to attach the mechanism.

In order to meet the motion requirement, a transmission was utilized to increase the torque output of the motor while decreasing the speed. The gear ratio was chosen after analyzing the torque needed to move the device along with angular velocity needed to move the backpack within the time limit. Power consumption was also calculated and minimized by changing the duty cycle of the control signal.

The mobile backpack carrier was controlled by an Arduino microcontroller that receives signals from sensors and output command signals to an H-bridge for motor control. Proximity sensors were used to prevent injury during the movement of device. The device would stop its motion as soon as the sensors detect obstacles. At the initial and final positions of the device, two limit switches were installed to stop the device and prevent it from damaging itself. The angular speed of the linkage was obtained from an encoder and converted to the real speed of the mechanism. A rocker switch was used for operator command input. The Arduino microcontroller monitored all signals obtained from the sensors along with the operator input and gave corresponding motor command using a state machine. A remote control and a liquid crystal display were employed for better user interface.

For safety concern, the device needed to be securely attached to the wheelchair so that there would be no significant deflection or movement of the device while in use. A half inch thick aluminum plate was used as the ground link so that the plate deflection was negligible. The device was programmed to slow down as it approached its final position to prevent injury or damage to the wheel chair.

The mobile backpack carrier designed and manufactured in this project was tested after assembly. All tested parameters fell within the range specified at the beginning and were very close to the ideal value. Further improvement can be done in terms of the appearance and power reduction.

Table of Contents

List of Figures

Main Content

Appendix A

Appendix B

Appendix C

Appendix D

Appendix E

List of Tables

Main Content

Appendix A

Appendix D

List of Equations

Main Content

1 Introduction & Specifications

1.1 Design Objective

The goal of this project is to design, build and test a mobile backpack carrier that allows the occupant of a wheelchair to be able to carry his backpack on his wheel chair and also have access to it. The occupant flips a switch and the arm moves from the back of the wheel chair to the side. The design must have a compact state to ensure that it does not interfere with the normal operation of the wheelchair. Besides the basic function, the device must operate efficiently in terms of its moving speed and power consumption while ensuring the safety of its user. Besides realizing the function, safety issues need to be taken into consideration. When using the system, wearing long sleeves, jewelry or having long hair down should be avoided. While the motor is operating, touching any moving part of the design or getting into the trajectory of the motion should be avoided. Upon setting up the power supply connection, all switches should be shut off. If powered off accidently, the device would resume to original state once powered on.

1.2 Objects in Contact

The mobile back carrier should only be in contact with the wheel chair and the backpack. The wheel chair used for the design is manufactured by QUICKIE and the model is QUICKIE 2. The wheels would be locked during testing and a 50 lbm weight would be used to balance and keep the wheelchair from tipping or sliding. The backpack used for testing is a standard backpack which weighs approximately 12 lbm with regular size. The backpack would be hanged on a standard backpack holder attached to the device.

1.3 Mechanism Requirement

The backpack should start from the back of the wheel chair and move around to the right side of the wheel chair and then return to the back using the same trajectory as shown in *Figure 1-1*. While the backpack carrier is at the final operating stage and ready for the user to pick up things from the backpack on his right hand side, the backpack holder is designed to be parallel to the armrest. The vertical offset of the backpack holder should also be at least 6 in. and no more than 17 in. from the centerline of the wheelchair for the convenience of the user. In addition, the right edge of the back pack holder should be at least 2 in. and no more than 10 in. from the right edge of the arm rest in its final position. The goal is to minimize the offset B to 2 in. The backpack cannot touch the wheel except at the end of its motion when it is at the right hand side of the wheel chair as shown in *Figure 1-1*. Detailed design target and parameters can be found in the **3.1 Design Selection** section in *Table 3-2*.

The moving mechanism should come to a hard stop at the beginning and end of its motion. The operating time is ideally 9 sec but can range from 6 sec to 12 sec. The rotation speed of the input link should be less than 10 degree per second in the last 30 seconds of the motion. In addition, the volume is measured by the smallest cubic box that the device can fit into. The volume of the design should be as small as possible. For practical purpose, the upper bound of the volume is set to be 3000 $in.^3$. The selected mechanism used to carry out the motion is a four-bar linkage system powered by one motor. The ground link of the mechanism should be connected to the standard wheel chair adapter without causing any damage. The detailed analysis for the mechanism requirement can be found in the **3 Motion Generation** section.

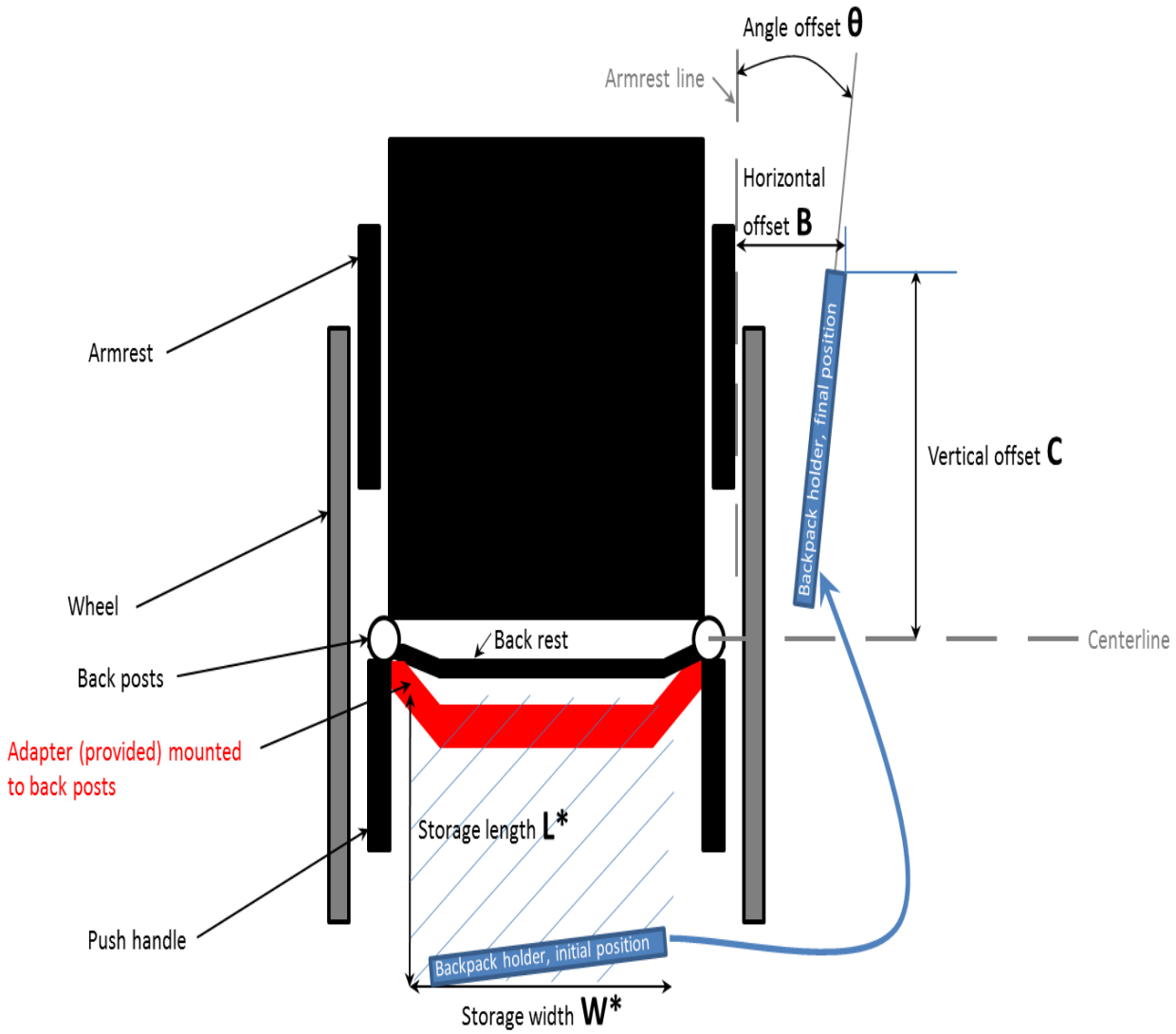


Figure 1-1: Design variable definition (Retrieved from ME 350 project description)

1.4 Actuator and Transmission

The motor used in the device is a Pololu (model number 1447) motor. It is a DC permanent magnet gear motor with a 131:1 gear ratio. For safety concerns, the maximum voltage supplied to the motor set to be 9 V. As the standard operating voltage of the DC motor is 12 V, by using linear interpolation, the estimated no-load speed is 60 RPM. For the mobile backpack carrier, a transmission is needed to allow the motor to provide suitable load to the linkage input. The transmission would be composed of timing belts, pulleys, and gears. Suitable speed and minimizing friction is essential for a successful transmission. The detailed analysis for the mechanism requirement can be found in the **4 Energy Conversion and Transmission** section.

1.5 Controller and Sensors

The mechanism is controlled by an Arduino Uno microcontroller board and is powered on by a rocker switch by the user. Sensors should be implemented in place to stop the motion of the mechanism if any object is detected between the moving linkage arms.

1.6 Craftsmanship and Testing

The mechanism needs to be durable for repeated use. All parts should be rigid and well designed with fillets. All unnecessary material and parts for testing should be removed. For the final test, the mechanism should be able to be attached or removed in less than five minutes. The performance of the device is determined by the volume in the starting position, the angle of the backpack holder in the final position relative to the armrest of the wheelchair, the horizontal distance from the armrest outside edge to the farthest point on the backpack holder, the perpendicular distance from the centerline to the farthest point on the backpack holder, the total travel time of the motion and the power consumption.

2 Functional Decomposition

Functional decomposition is an important and useful technique for processing knowledge. By dividing complicated real problem into smaller pieces, engineers would be able to design subsystems in detail to satisfy the individual requirement and later combine subsystems to realize the functionality of the whole design. In this project, the technique of functional decomposition was employed to divide the mobile backpack carrier design into several subsystems as shown in *Figure 2-1* with variable declaration included in *Table 2-1*.

Notation	Variable
V	Voltage
I	Current
T	Torque
ω	Angular Velocity
E	Energy
S	Signal

Table 2-1: Variable declaration for functional decomposition chart

*Subscript stands for the element that the variable belongs to.

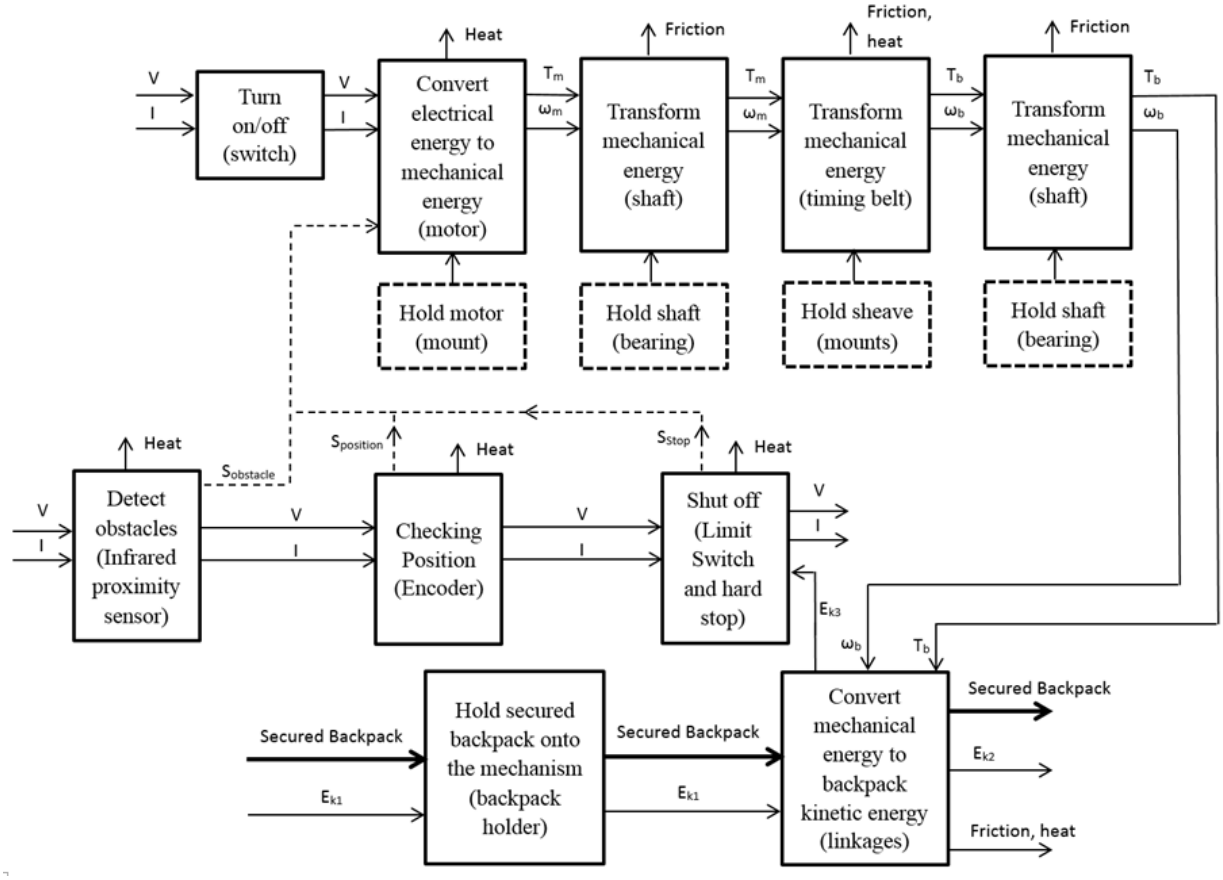


Figure 2-1: Basic functional decomposition chart

3 Motion Generation

3.1 Design Selection

The design process started with a group meeting by looking at videos of designs by previous teams to get an idea of what successful projects should be like. The entire team went through the project description and talked about the key parameters for the design. Then the software Lincages was used to get a rough design of the link length and optimization was conducted to improve the performance of the design. Upon obtaining the general length of the links, modeling in Solidworks and analysis in ADAMS of each design was carried out to obtain further engineering parameters of interest. The analysis for each design can be found in Appendix A.

After completing individual design, another team meeting was held to share the design parameters as summarized in *Table 3-1* and a Pugh chart was created to compare all four designs. The important factors for the design were categorized by associating a number to each parameter to represent its corresponding priority. The associated numbers were used as weight for the design selection purpose. The most important parameters were the angle of the backpack holder in the final position relative to the armrest of the wheelchair, the horizontal distance from the armrest outside edge to the farthest point on the backpack holder and the torque at the input joint. The first two parameters were listed in the project requirement and the torque input joint was important since it would affect the friction force exerted on the motor shaft and therefore affect the power consumption. A relative scale for the parameters of interest was used and at the end of comparison, Fangzhou Xia's design scored the highest in the Pugh chart and was chosen as the final design as shown in *Table 3-2*. For the detailed definition of the parameter, please refer to *Figure 1-1*.

Name	Fangzhou	Nolan	Xingjian	Zach
Volume (in.^{2}) (Assume h=10 in.)	1367	1700	2210	1435
Angle Offset (°)	0	0	0	0
Horizontal Offset (in.)	2.078	3.6	5	3.27
Forward Offset (in.)	12.58	9.36	8.7	12.12
Input Joint Torque (lbf*in.)	189	175	209	400
Input Joint Force (lbf)	16	15	86	86

Table 3-1: Raw data from each design

Parameter	Range	Target	Weight	Fangzhou	Nolan	Xingjian	Zach
Volume (h=10 in.)	<=3000 in.^{3}	3000 in.^{3}	1	0.54	0.43	0.26	0.52
Angle Offset	±15°	0°	3	1	1	1	1
Horizontal Offset	2-10 in.	2 in.	3	0.79	0.64	0.5	0.67
Forward Offset	6-17 in.	17 in.	3	0.60	0.31	0.25	0.56
Input Joint Torque	-	0 lbf*in.	2	0.04	0.07	0.95	0.12
Input Joint Force	-	0 lbf*in.	1	0.68	0.71	-0.69	-0.69
Total	0-13	13	0-3	8.47	7.13	6.72	6.76

Table 3-2: Pugh chart for final design selection

3.2 Linkage Design

As can be seen in *Figure 3-1*, the final design used a double crank four-bar linkage mechanism. This design started in a compact position at the back of the wheel chair and finishes parallel to the armrest. The transmission angle was between 15 degrees and 100 degrees with useful range from 25 degrees to 100 degrees as shown in *Figure 3-2* and *Figure 3-3*. The converted link lengths were shown in *Table 3-3*.

Link name	Input	Output	Coupler	Ground
Actual length (in.)	11.41	11.83	2.84	3.51

Table 3-3: Link lengths of selected design

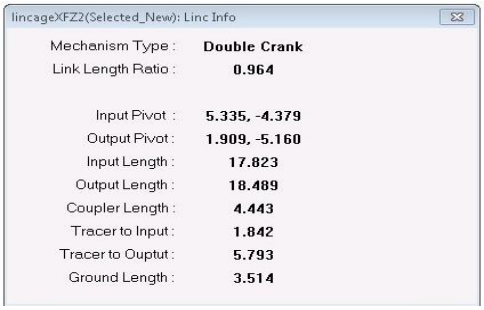


Figure 3-1: Link lengths from Lincages

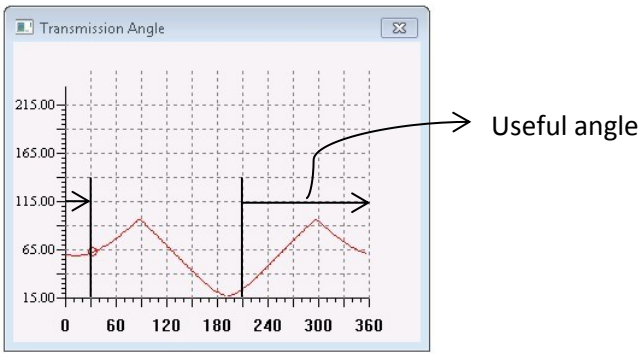


Figure 3-2: Lincages transmission angle

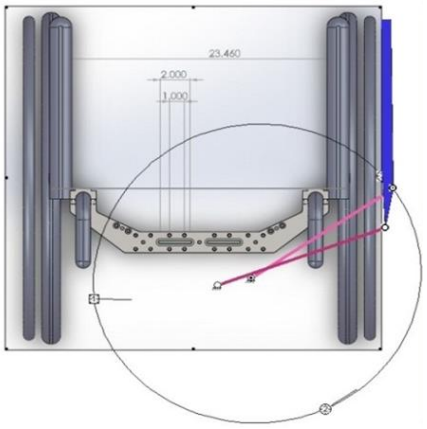


Figure 3-3: Design overview in Lincages

3.3 CAD Model

By using the link length generated in linkages, the detailed CAD models in Solidworks are generated and shown from *Figure 3-4* to Figure *3-11* for different views respectively.

Figure 3-4: Isometric view of the design with backpack

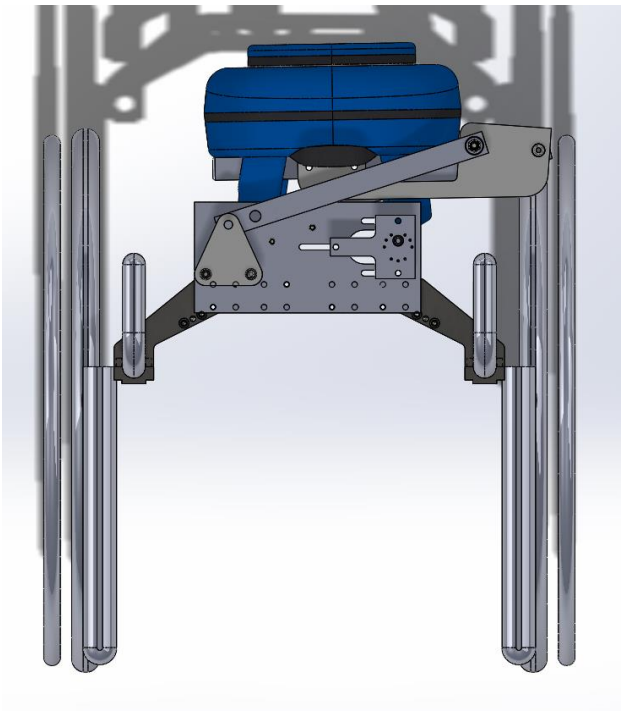


Figure 3-5: Top view of the design at the initial position

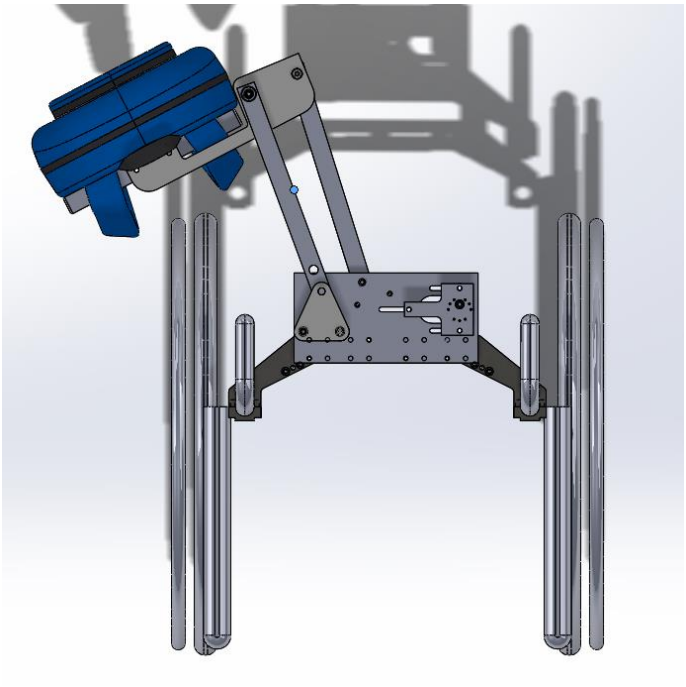


Figure 3-6: Top view of design at transition position

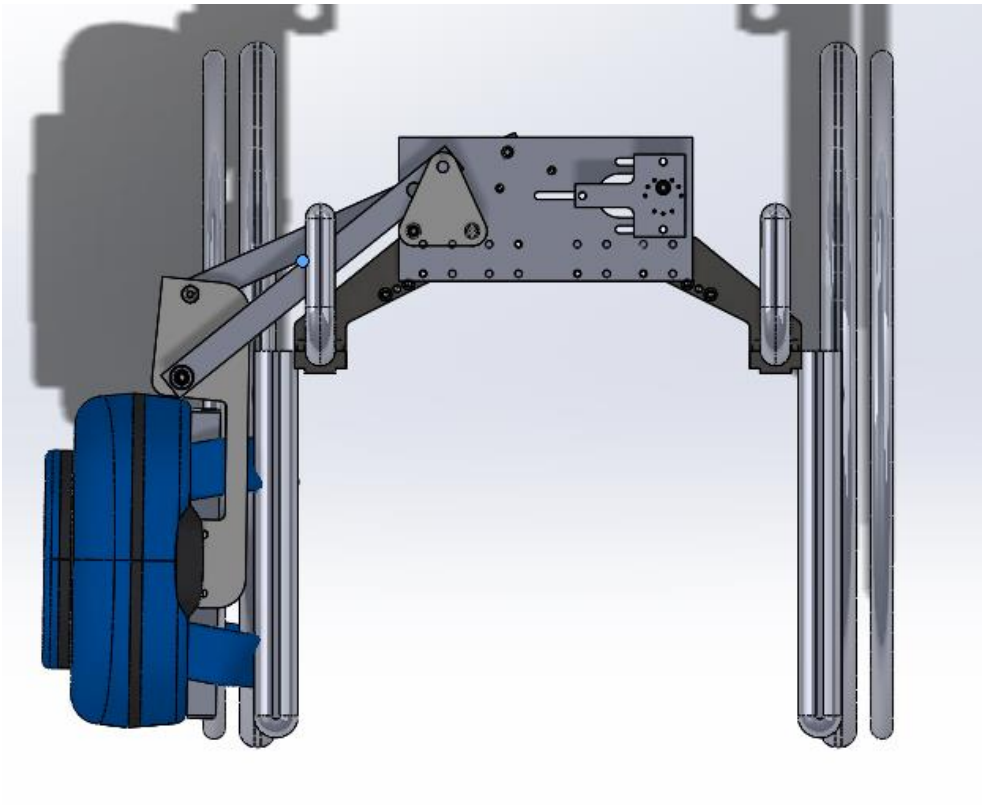


Figure 3-7: Top view of the design at final position

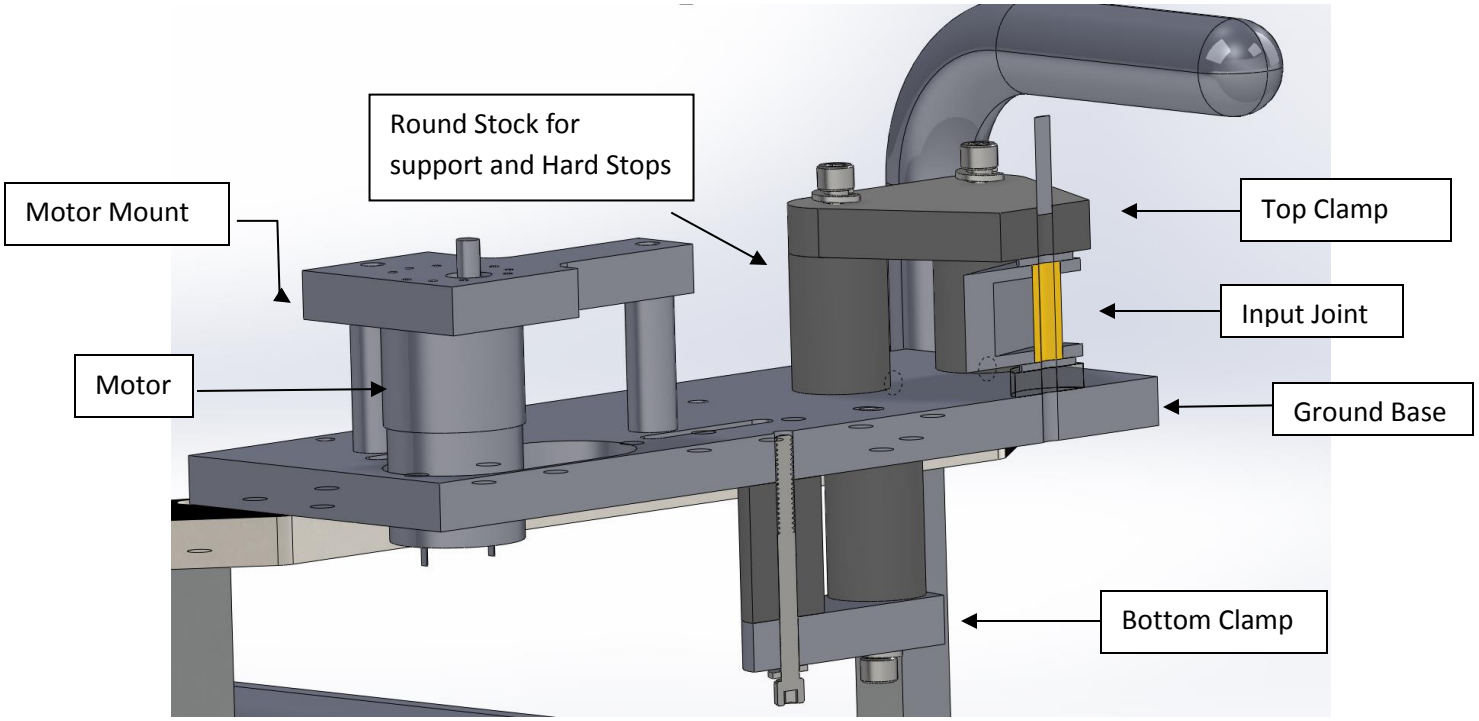


Figure 3-8: Section view for joint of input link and ground

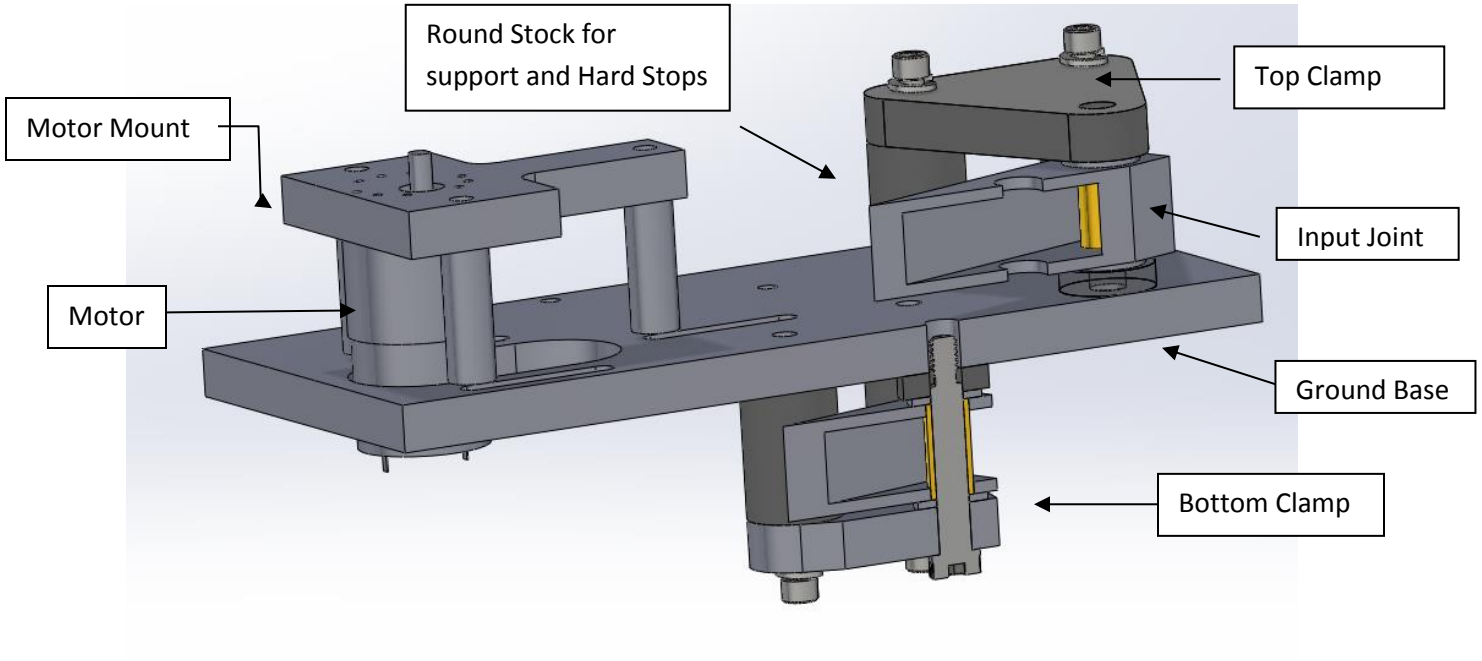


Figure 3-9: Section view for joint of output link and ground

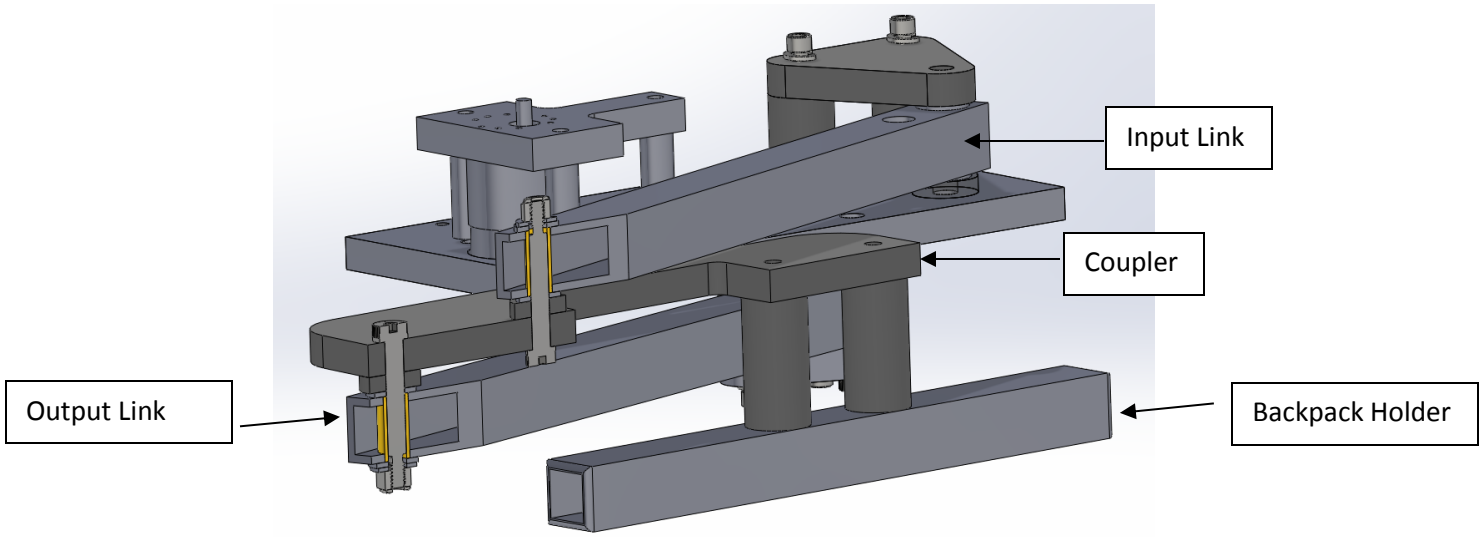


Figure 3-10: Section view for joint of input and output with coupler

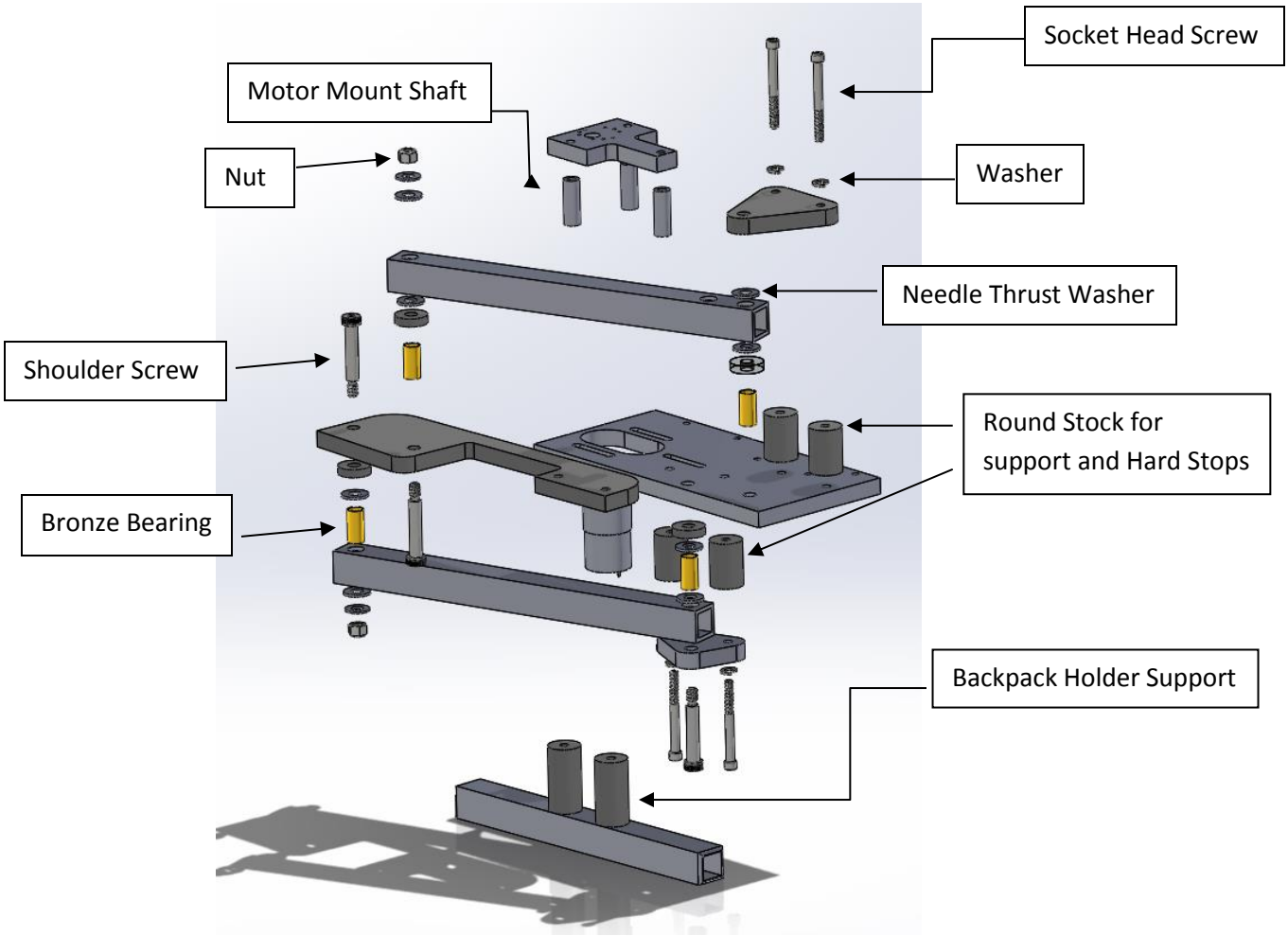


Figure 3-11: Explosion view of the design

3.4 Loading Analysis

In this section, detailed analysis based on the free body diagram of the coupler, input and output links respectively would be derived in the form of analytic equations for the loading. Note that the geometry is simplified for each part and the moment at the two joints on the coupler have been neglected for the purpose of calculation. For detailed analysis and numeric values of the forces, please refer to **Appendix A.4 Detailed Calculations for Deflections**.

3.4.1 Coupler

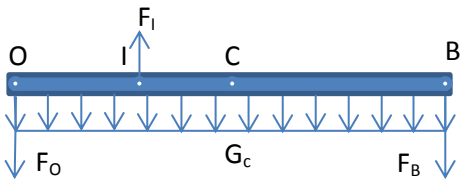


Figure 3-12: Coupler free body diagram (simplified & not to scale)

Definitions:

Point O: Joint for the output link and the coupler

Point I: Joint for the input link and the coupler

Point C: Center of gravity (mid-point after approximation)

Point B: Acting point of the gravity of the backpack and backpack holder (end point of the coupler after approximation)

Force F_O: Vertical downward force exerted on the coupler by the output link at Point O

Force F_I: Vertical upward force exerted on the coupler by the input link at Point I

Force F_B: Vertical downward force exerted on the coupler by the backpack holder at Point B

Load G_c: Vertical downward gravity of the coupler acting uniformly along the coupler (uniformly distributed with equivalent force equal to G_c)

Calculation:

Using the free body diagram, the unknown force F_I and F_O can be obtained based on the measureable dimension and known force F_B and calculated load G_c.

$$\begin{cases} F_I = \dfrac{l_{OB}F_B + l_{OC}G_c}{l_{IO}} \\ F_O = \dfrac{l_{IB}F_B + l_{IC}G_c}{l_{IO}} \end{cases} \qquad \text{Equation 1}$$

3.4.2 Input

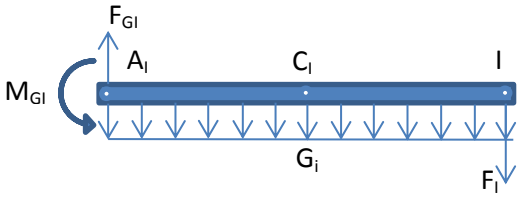


Figure 3-13: Input link free body diagram (simplified & not to scale)

Definitions:

Point A_I: Joint for the input link and the ground

Point I: Joint for the input link and the coupler

Point C_I: Center of gravity (mid-point)

Force F_{GI}: Vertical upward force exerted on the input link by the ground at Point A_I

Force F_I: Vertical downward force exerted on the input link by the coupler at Point I

Load G_i: Vertical downward gravity of the input acting uniformly along the coupler (uniformly distributed with equivalent force equal to G_i)

Torque M_{GI}: Counterclockwise torque at point A_I exerted by the ground on the input link

Calculation:

Using the free body diagram, the unknown force F_{GI} and torque M_{GI} can be obtained based on the measureable dimension calculated as well as force F_I and load G_o.

$$\begin{cases} F_{GI} = F_I + G_i \\ M_{GI} = l_{A_I I} F_I + l_{A_I C_I} G_i \end{cases} \quad \text{Equation 2}$$

3.4.3 Output

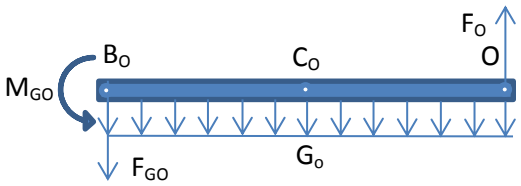


Figure 3-14: Output link free body diagram (simplified & not to scale)

Definitions:

Point B_O: Joint for the output link and the ground

Point O: Joint for the output link and the coupler

Point C_O: Center of gravity (mid-point)

Force F_{GO}: Vertical downward force exerted on the output link by the ground at Point B_O

Force F_O: Vertical downward force exerted on the output link by the coupler at Point O

Load G_o: Vertical downward gravity of the input acting uniformly along the coupler (uniformly distributed with equivalent force equal to G_o)

Torque M_{GO}: Counterclockwise torque at point B_O exerted by the ground on the output link

Calculation:

Using the free body diagram, the unknown force F_{GO} and torque M_{GO} can be obtained based on the measureable dimension as well as calculated force F_O and load G_o.

$$\begin{cases} F_{GO} = F_O - G_o \\ M_{GO} = l_{B_O C_O} G_o - l_{B_O O} F_O \end{cases} \quad \text{Equation 3}$$

3.5 ADAMS Simulation

By using the CAD model generated in Solidworks, the model was then imported into ADAMS for detailed analysis as shown in *Figure 3-15*. With constrains and material type defined, the following analysis result have been obtained. *Figure 3-16* and *Figure 3-17* shows the variation of force and torque with respect to time for each joint. The maximum and minimum force and torque has been summarized in *Table 3-4*.

	Min Force (lbf)	Max Force (lbf)	Min Torque (lbf*in.)	Max Torque (lbf*in.)
Input-coupler	2	16	5	181.5
Input	2	16.5	0	189
output-coupler	1.5	19.5	0	152.97
output	1.5	20	0	252

Table 3-4: Table of ADAMS results for final design

Based on the analysis, the maximum torque of the input link with the ground link was 190 lbf*in. In order to ensure a smooth motion of the mechanism, the input joint should be double supported in the final design in order to provide larger torque tolerance with smaller deflection for the input joint to reduce the extra friction generated in the transmission due to large torque.

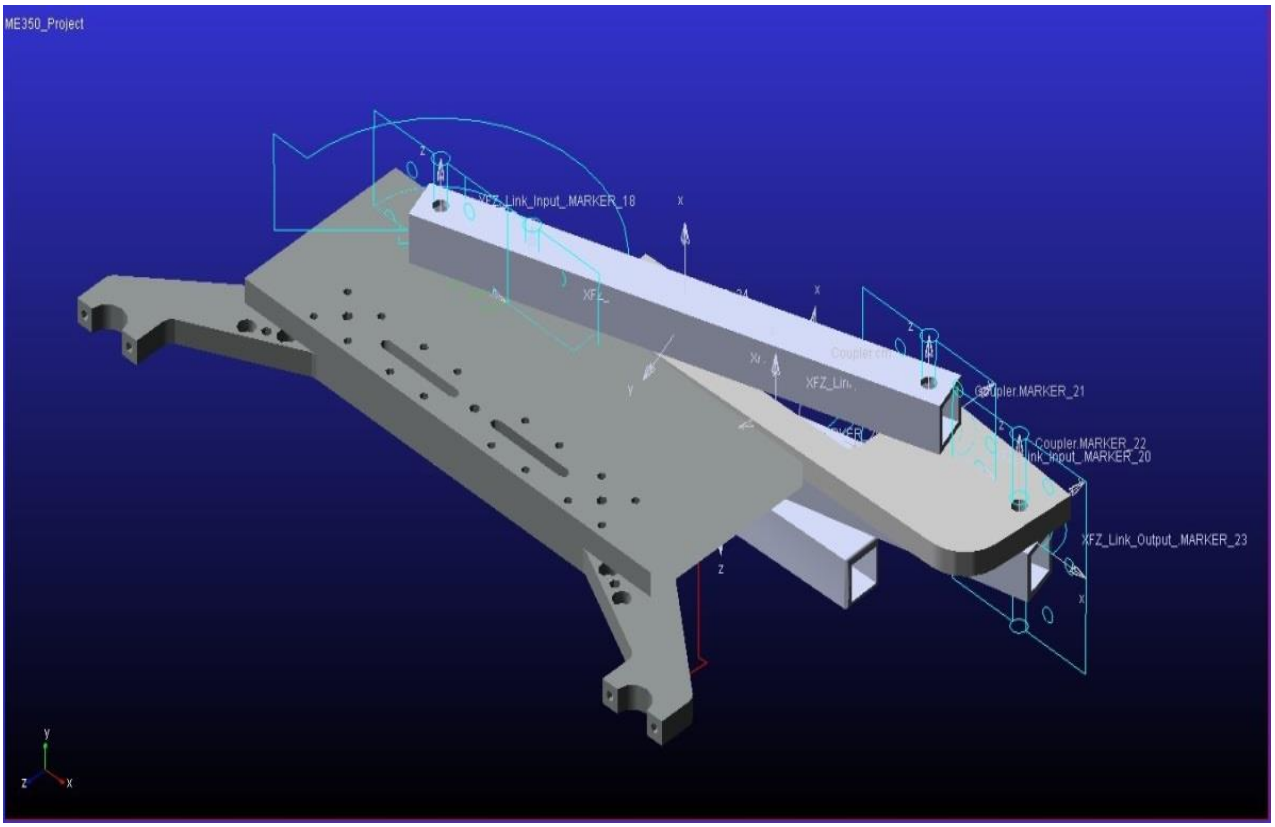


Figure 3-15: ADAMS model of final design

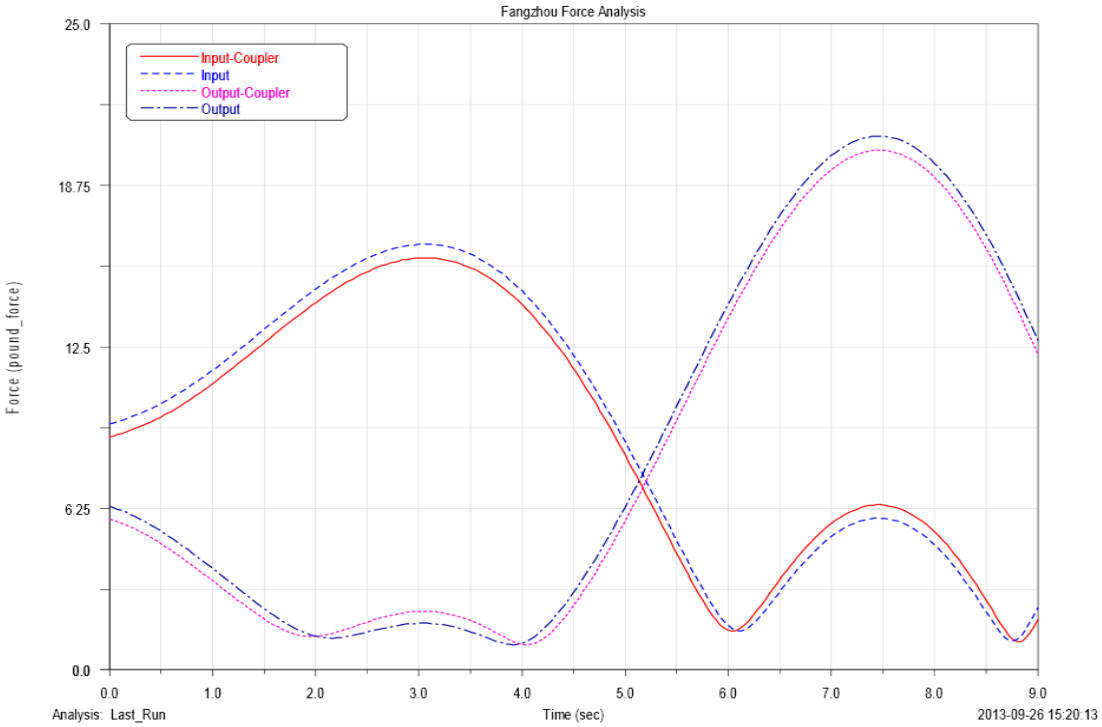


Figure 3-16: ADAMS force analysis result

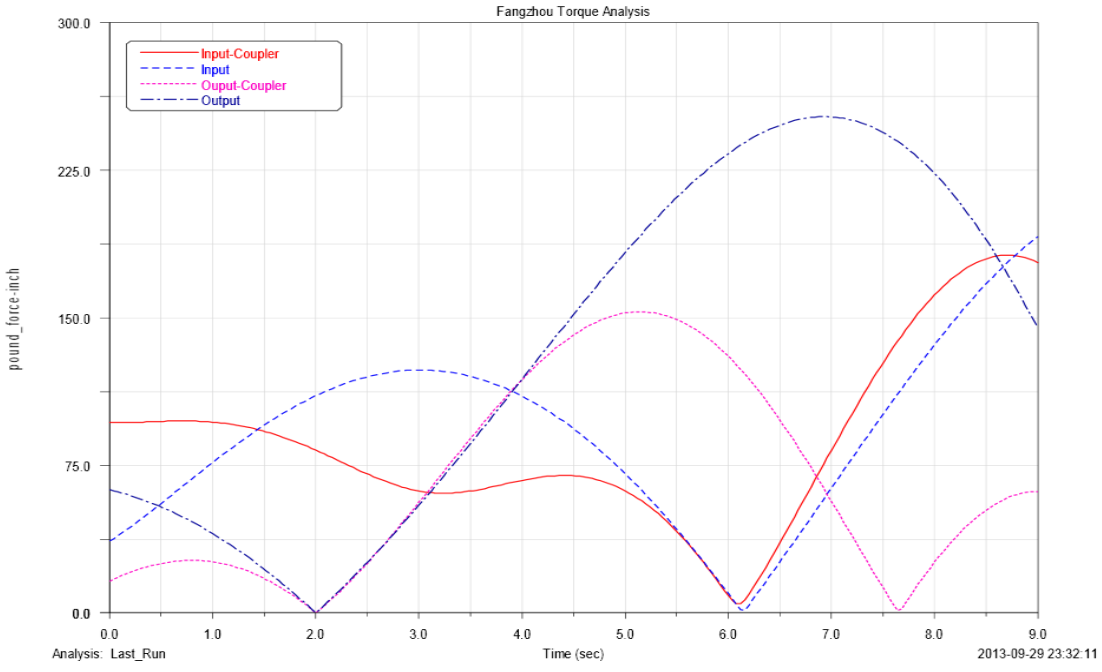


Figure 3-17: ADAMS torque analysis result

3.6 Preliminary Testing Result

During the design process, testing is an integral part of the design as it brings up the unexpected problem and helps to obtain data that are too complicated to calculate. As for the design of the mobile backpack carrier preliminary testing was conducted right after the completion of the linkage manufacturing and assembling. The result of the test was directly used for the modification of the original design as an optimization. Detailed values and changes can be found in section **3.7 Design Modification.**

3.7 Design Modification

With the link length properly designed and simulated, no significant changes were required for the four-bar linkage mechanism. However, during the process of manufacturing, minor changes were made on the geometry of some parts to cater to the raw materials given. The list of modified design can be found in *Table 3-5* and the corresponding drawings were shown below from *Figure 3-18* to *Figure 3-22*.

Part Name	Description of Change
Top Clamp	Outer contour for finding datum line
Bottom Clamp	Outer contour for finding datum line
Motor Mount	Outer contour for stiffness and sink hole for transmission
Aluminum Spacing	Outer contour for easy manufacturing
Ground Link	Number of mounting holes reduced for easy manufacturing Hole position for the adapter for travel torque reduction

Table 3-5: List of parts with changed design

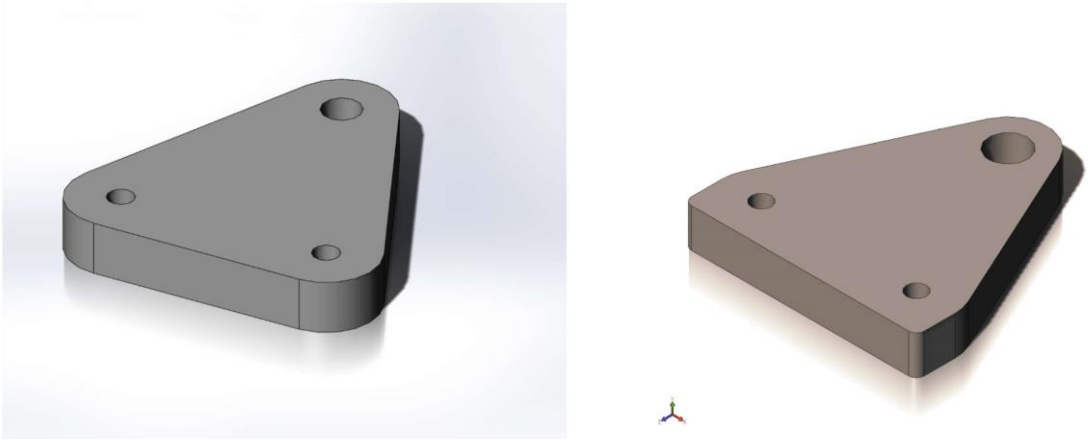


Figure 3-18: Top Clamp old design (left) and new design (right)

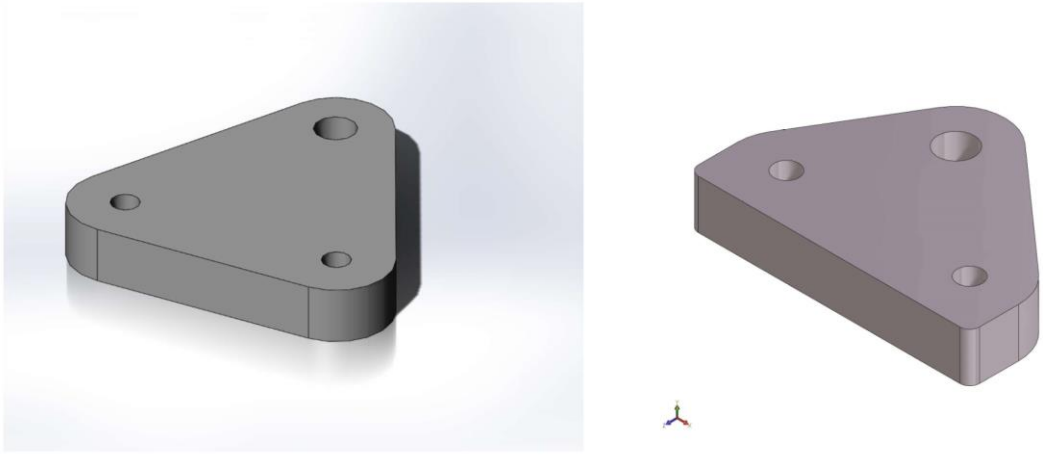


Figure 3-19: Bottom Clamp old design (left) and new design (right)

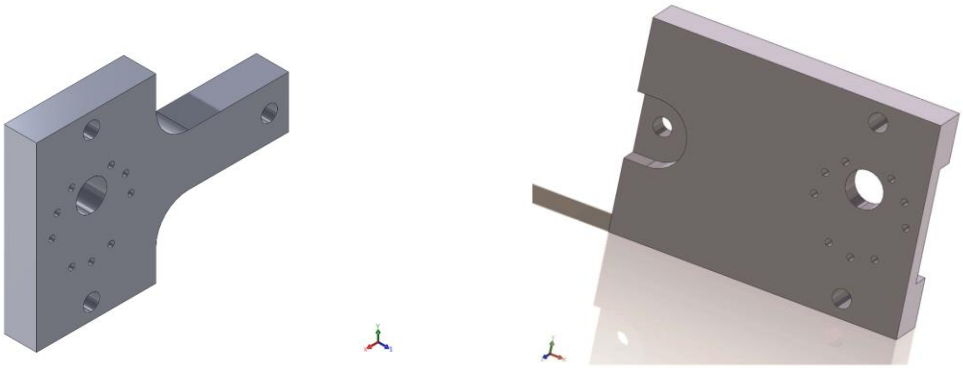


Figure 3-20: Motor Mount old design (left) and new design (right)

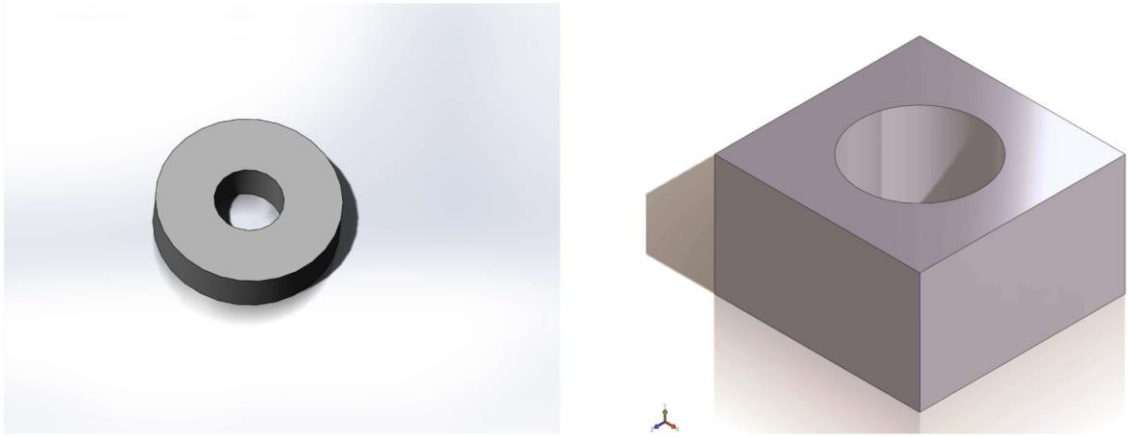


Figure 3-21: Aluminum Spacing old design (left) and new design (right)

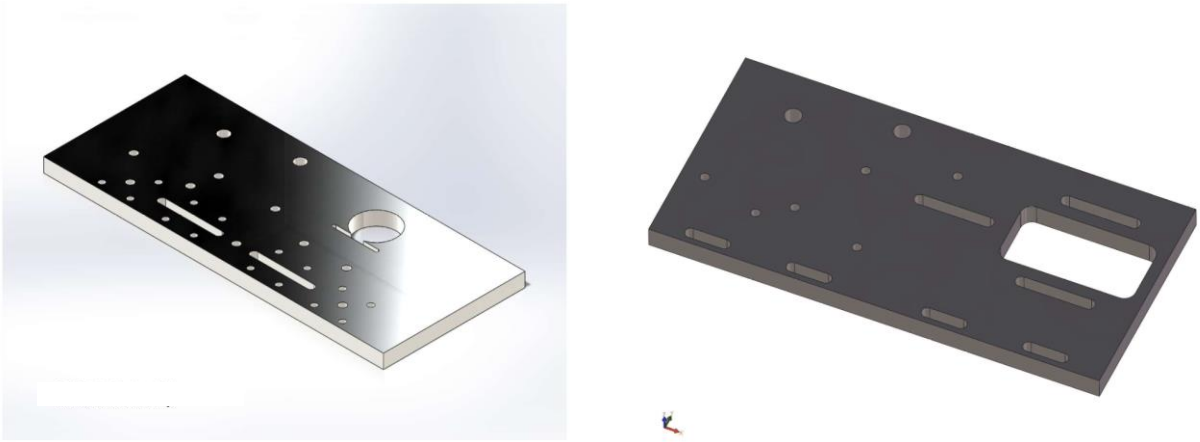


Figure 3-22: Ground Link old design (left) and new design (right)

During the process of torque testing, several engineering tradeoffs were made to compensate for the extra spacing needed for the backpack which had not been adequately estimated before. Also due to the extreme torque value applied on the input link when the horizontal offset was around 2 in., the tradeoff was made to reduce the torque greatly with the slight cost of horizontal offset value. The tradeoff data sheet can be found in *Table 3-6*. Note that the parameter value change due to adding transmission system is not included here.

Parameter name	Value before tradeoff	Value After tradeoff
Maximum Input Torque (oz-in.)	1289.33	804.41
Maximum Input Power (W)	32.17	20.07
Horizontal Offset (in.)	1.378	2.078
Vertical Offset (in.)	12.58	11.58

Table 3-6: Parameter value change due to tradeoff

As can be seen from the table, the offset value was slightly increased while the input torque and power were greatly decreased. The purpose of the trade of is to minimize the max torque and therefore the max power input of the mechanism with little cost of the ideal offset value. During the preliminary testing of the mechanism, we found that the torque required for the linkage to achieve its final position was very large due to the force exerted on the bag by the wheel as they make contact. The tradeoffs were made to reduce the force and therefore the torque.

4 Energy Conversion and Transmission

4.1 Introduction

Transmission is a mechanism used for controlled application of the power source. The transmission usually involves torque and speed conversion where a gearbox can serve as a typical example. In many applications, the given power sources have enough power supplying capability but do not match the requirement of the desired output in terms of torque and force or angular velocity and linear velocity. In this case, a transmission system is usually employed to transmit the power provided by the power source into the ideal value for the specific application. Although the transmission itself does not produce power and actually has a efficiency less than one which means it consumes some of the power supplied by the power source, it is still necessary as it provides a means to obtain desired output value by transmitting power from standard power source to different torque and angular velocity output.

For the mobile backpack carrier design, the standard power source is a permanent magnet gear motor with a 131:1 gear ratio as shown in *Figure 4-1*. At standard operating voltage 12 V, the motor specification is shown in *Table 4-1*. For safety concern the voltage applied to the motor would be 9 V.

Figure 4-1: 37D mm metal gearmotor with 64 CPR encoder and Pololu 90×10mm wheel (retrieved from 1447-Pololu data sheet)

Status	Free run		Stall	
Parameter Name	Angular velocity	Current	Torque	Current
Value	80 RPM	300 mA	250 oz-in	5 A

Table 4-1: 1447-Pololu motor specification

As mentioned in the motion generation section, the operating time for the mechanism is ideally 9 sec but can range from 6 sec to 12 sec which is much slower than the output velocity of the motor even if 9 V is applied instead of 12 V. On the other hand, the direct output torque of the motor 250 oz-in is relatively small compared to the friction torque based on precious designs where the low value is 290 oz-in. Hence, for the purpose of the design, a transmission system is necessary.

4.2 Transmission Design

In order to design a transmission ratio to meet the requirement of converting the motor output to large enough torque under suitable angular velocity, the motor characteristic, maximum torque and required

angular velocity need to be determine. The variable declaration for this section is shown in *Table 4-2*.

Variable Name	Notation	Variable Name	Notation
Motor Torque	T_m	Motor Constant	K_m
Stall Torque	T_s	Torque Constant	K_t
Angular Velocity Of Motor	ω_m	Current	I_m
Supplied Voltage	V_S	First 6 Sec Output Angular Velocity	ω_1
Last 3 Sec Output Angular Velocity	ω_2	Total Time Of Movement	t
First 6 Sec Rotation Angle	θ_1	Last 3 Sec Rotation Angle	θ_2
Angular Speed Load Effect Constant	LEC_{AS}	Transmission Ratio	N
High Input Torque	T_h	Maximum Actual Input Torque	T_{MA}
Maximum Output Power	P_{MO}	Maximum Input Power	P_{MI}

Table 4-2: Variable declaration

4.2.1 Motor Characteristic at 9V

For the motor used in the design, 9 V is used instead of 12 V as the voltage input, which shifts the torque versus angular speed curve downward. By using the DC motor character equation and the parameters given the revised DC motor curve at 9V instead of 12 V can be generated. Note that the motor given is not ideal as the free run current is 300 mA rather than 0 A but is close enough for the equation used here to be valid.

$$\begin{cases} T_m = T_s - K_m\omega_m = \dfrac{V_S K_t}{R} - \dfrac{K_t K_b}{R}\omega_m \\ V_{Sstandard} = 12\,V, V_{Sreal} = 9\,V \end{cases} \qquad \text{Equation 4}$$

As can be seem from the DC motor character equation, the stall torque is proportional to the voltage supplied to the DC motor. The modified motor curve and operating parameters can be found in *Figure 4-2* and *Table 4-3*.

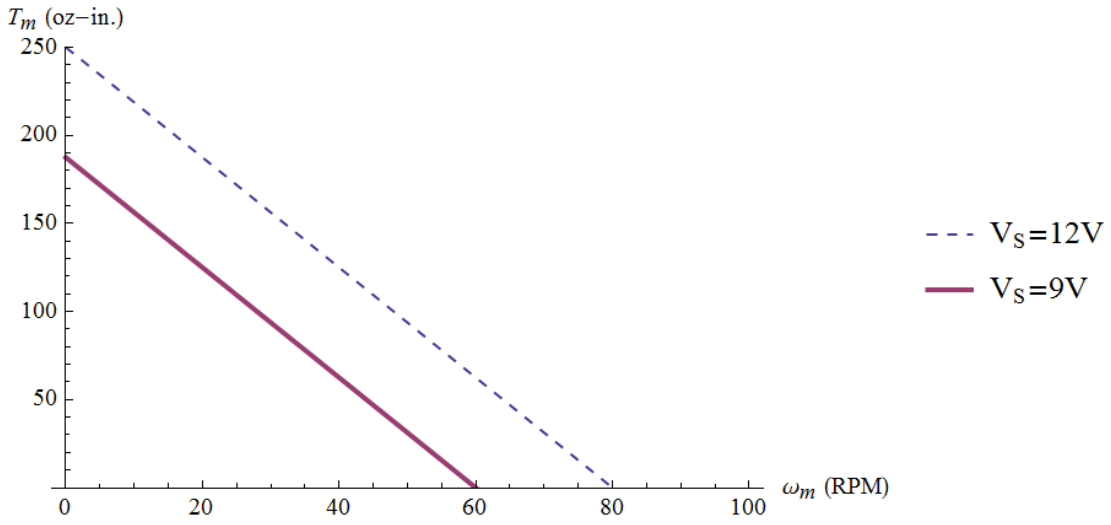


Figure 4-2: Motor Characteristic Curve

Status	Free run	Stall

Parameter Name	Angular velocity	Current	Torque	Current
Value	60 RPM	300 mA	187.5 oz-in	3.75 A

Table 4-3: Revised Pololu motor operating parameters at 9 V

The motor constant can then be calculated as:

$$K_t = \frac{T_{m@noload}}{\mathrm{I}_{m@stall}} = 50\,\mathrm{oz}\cdot\mathrm{in}./A \qquad \text{Equation 5}$$

$$K_m = \frac{T_{m@stall}}{\omega_{m@noload}} = 29.84\,\mathrm{oz}\cdot\mathrm{in}.\cdot\mathrm{sec} \qquad \text{Equation 6}$$

4.2.2 Angular Velocity Requirement

In order to determine the transmission ratio required for the design, the angular velocity should be determined first according to the time and angles specified previously. The total time of motion for the mechanism should be 9 sec with the angular velocity less than 10 deg/sec at the last 30 degrees of motion. Based on the design selected, the total rotated angle and the time of travel are given by

$$\theta_1 + \theta_2 = 200 \qquad \text{Equation 7}$$

$$\mathrm{t} = 9\,\mathrm{sec} = \frac{200\,\mathrm{deg} - \omega_2 * 3\mathrm{sec}}{\omega_1} + 3\,\mathrm{sec} \qquad \text{Equation 8}$$

In this formula, the term ω_2 can be taken by any value below 10 deg/sec. For the purpose of calculation $\omega_2 = 8\,\mathrm{deg/sec}$ are used which then gives

$$\omega_1 = 29.33\frac{\mathrm{deg}}{\mathrm{sec}} = 4.89\mathrm{RPM} \qquad \text{Equation 9}$$

In order to take the acceleration part into account, which would result in longer travel time, a safety factor of $\mathrm{LEC_{AS}} = 2$ is utilized to yield

$$\mathrm{N} = \frac{\omega_{\mathrm{m}@noload}}{\mathrm{LEC_{AS}}\cdot\omega_1} = \frac{60\mathrm{RPM}}{2\times 4.89\mathrm{RPM}} = 6.13{:}\,1 \qquad \text{Equation 10}$$

Based on the availability of common transmission, the transmission ratio $\mathrm{N} = 6{:}\,1$ was selected

4.2.3 Torque Requirement Verification

With the transmission ratio $\mathrm{N} = 6{:}\,1$ determined and satisfying the angular velocity requirement, the verification for torque can be done by taking comparing the high mechanism torque from previous year design with the stall torque of the transmission times the transmission ratio. In this case, it yields

$$\mathrm{N}\cdot\mathrm{T}_{s@stall} = 6\times 187.5\,\mathrm{oz}\cdot\mathrm{in}. = 1125\,\mathrm{oz}\cdot\mathrm{in}. > 1050\,\mathrm{oz}\cdot\mathrm{in}. \qquad \text{Equation 11}$$

Since the output torque for the mechanism is larger than the high torque from previous year data, the torque verification condition is satisfied.

After taking actual measurement of the torque required using a force meter, the forces applied perpendicular to the input link at its joint with the coupler required for the mechanism to complete its motion both forward and backward are shown respectively in *Table 4-4* and *Table 4-5* respectively.

Measurement	1	2	3	4	5	6	7	8	9	10
Force measured (oz)	0	13.0	35.0	36.5	46.0	53.5	66.5	70.5	110.0	113.0

Table 4-4: Forward motion torque

Measurement	1	2	3	4	5	6	7	8	9	10
Force measured (oz)	0	0	0	0	0	33	30.5	36.0	41	48.5

Table 4-5: Backward motion torque

The extreme force load and therefore torque load at the end of the forward motion is due to the backpack holder being leaning on the wheel, producing extra torque. This problem was resolved by moving the entire ground link 0.75 in. to the right and therefore the largest force of the measurement was taken to be up to 70.5 oz as the 110.0 oz and 120.0 oz measurements were reduced. View the Preliminary Testing Result

During the design process, testing is an integral part of the design as it brings up the unexpected problem and helps to obtain data that are too complicated to calculate. As for the design of the mobile backpack carrier preliminary testing was conducted right after the completion of the linkage manufacturing and assembling. The result of the test was directly used for the modification of the original design as an optimization. Detailed values and changes can be found in section **3.7 Design Modification.**

Design section for detailed explanation. For the purpose of torque analysis, the length of the input link l_{input} and the maximum actual torque measured T_{MA} after design modification were used to give

$$\mathrm{T}_{MA} = l_{input} \cdot T_{MA} = 804.405 \text{ oz} \cdot \text{in.} < 1353.75 \text{ oz} \cdot \text{in.} \qquad \text{Equation 12}$$

This proved that the stall torque was larger than maximum input torque and the mechanism was able to reach its final position.

For the operating torque at $\omega_1 = 29.33 \text{ deg/sec}$ the operating

$$\begin{cases} T_m = \dfrac{T_{m@stall} \cdot N}{FS_{AV}} = \dfrac{1350 \text{ oz} \cdot \text{in.}}{1.7} = 740.12 \text{ oz} \cdot \text{in.} \\ \mathrm{T}_{MA@\omega_1} = 53.5 \text{ oz} \times 11.41 \text{ in.} = 610.435 \text{ oz} \cdot \text{in.} \end{cases} \xRightarrow[comparison]{} T_m > \mathrm{T}_{MA@\omega_1} \qquad \text{Equation 13}$$

The torque analysis verified that the transmission ratio of $\mathrm{N} = 6{:}\,1$ met the requirement of both the angular velocity and the torque.

4.2.4 Maximum Power Analysis

During the process of motion, the input torque of the mechanism would increase from 0 to half of ω_m at the no load condition. Therefore, the maximum output power can be obtained at the point where $\omega_m = \omega_{m@noload}/2$ and can be calculated as

$$\mathrm{P}_{MO} = \frac{\mathrm{T}_{m@stall} \cdot \omega_{m@noload}}{4} = \frac{187.5 \text{ oz} \cdot \text{in.} \cdot 60\text{RPM}}{4} = 2.08\ W \qquad \text{Equation 14}$$

For the input power, as the input voltage was fixed to be 9 V, the maximum input power was obtained when the current was maximized. Since the current is proportional to the output torque, the maximum input power is obtained at the maximum input torque. In this case, the maximum measured torque was utilized to yield

$$\mathrm{I}_m = \frac{\mathrm{T}_{MA}}{N \cdot K_t} = \frac{804.405 \text{ oz} \cdot \text{in.}}{6 \times 50 \text{ oz} \cdot \text{in.}/A} = 2.68\text{A} \qquad \text{Equation 15}$$

$$P_{MI} = V_S I_m = 9\ \text{V} \times 2.68\ \text{A} = 24.132\text{W} \quad \text{Equation 16}$$

Notice that the maximum input power and maximum output power were not obtained at the same time. This was true since that the maximum output power was obtained at $\omega_m = \omega_{m@noload}/2$ while the maximum input power when voltage was fixed was obtained at $\omega_m = \omega_{m@min}$ with the input current I_m maximized under this condition. For the operating power, it depends on the PWM signal applied for speed control.

4.2.5 Design Parameters for Control

In the process of design, the parameters that can be changed are listed in *Table 4-6*.

Variable Name	Notation	Variable Name	Notation
First 6 sec output angular velocity	ω_1	Last 3 sec rotation angle	θ_2
Angular speed load effect constant	LEC_{AS}	Transmission ratio	N

Table 4-6: Variable parameters

Based on the procedure used to generate the transmission ratio, the direct independent variable for the determination of the input and output power was the transmission ration N and the angular first 6 sec output angular velocity ω_1. They were independent mainly because the value of angular velocity safety factor FS_{AV} can be chosen by the designer, as can be seen from the formula

$$\begin{cases} P_{MO} = \dfrac{T_{m@stall} \cdot \omega_{m@noload}}{4} \quad (\text{when } \omega_m = \dfrac{\omega_{m@noload}}{2} \text{ achieved}) \\ P_{MI} = V_S I = \dfrac{V_S \cdot T_{MA}}{N \cdot K_t} \end{cases} \quad \text{Equation 17}$$

The maximum output power can be minimized by reducing the transmission ratio N so that the operating angular velocity ω_1 was less than half of the no load angular velocity $\omega_{m@noload}$. Note that the torque for the external load T_L should be smaller than the operating torque so that the motion can be completed smoothly which served as a lower bound of the transmission ratio N. This made reducing the load torque ideal to reduce power.

In order to minimize the input power with fixed voltage input, the maximum torque of the load was needed to be minimized. This was done by increasing the transmission ratio to reduce the current at the maximum input power condition. Note that the operating angular velocity condition ω_1 should also be satisfied which set an upper bound to the largest transmission ratio that can be obtained. In addition, minimizing the load torque T_l is also an ideal approach to reduce the input power.

4.2.6 Minimizing Power by Link Design

As mentioned in the previous subsection, the input and output power are both related to the load torque T_l which should be minimized. In the process of link design for the four-bar linkage mechanism, the friction force of joints between links should be minimized. The steps of the design can be decomposed into the following three steps:

1. Use ADAMS simulation result to modify design.
2. Include needle thrust bearing for friction reduction
3. Double support joints under large torque to reduce the normal force for friction

4.2.7 Transmission Selection

In order to choose between different types of transmission, a Pugh chart was created for the purpose of transmission selection. With the key parameters taken into consideration, the weighted sum was produced to compare between different transmissions. As can be seen in *Table 4-7*, timing belt scored the highest due to its comparative advantage in different aspects compared with gears and chains and was therefore chosen for the transmission of the design.

Property	Relative Performance			Score		
Type of transmission	Gears	Timing Belts	Chains	Gears	Timing Belts	Chains
Torque transmission	High	Medium	Medium	3	2	2
Accuracy under large load	High	Medium	Medium	3	2	2
Load on shaft	High	Low	Low	1	3	3
Clearance requirement	High	Low	Low	1	3	3
Price	Medium	Low	Medium	2	3	1
Weight	High	Low	Medium	1	3	2
Noise	Low	Low	Medium	3	3	2
Sum	-	-	-	14	19	15

Table 4-7: Pugh chart for final transmission selection

4.2.8 Advantage and Disadvantage

The advantage of using a timing belt lies in the fact that timing belts are cheap, light in weight and exert almost no extra load on the motor shaft as using them does not require initial tension between the pulleys and the net force exerted on the pulleys by the belts is close to zero. In addition, the center distance of pulleys for timing belts is relatively flexible compared to gears. With larger clearance, the manufacturing process for the mountings becomes easier.

The disadvantage of using timing belt is that there would be small amount off backward sliding due to the change in tension side when the applied torque is changed under high large torque. However, the mobile backpack carrier design does not require very large torque, which should not be a big problem in this case.

4.2.9 Manufacturer and Part Number

The transmission components which consist of two timing pulleys and a timing belt were purchased at Stock Drive Products/Sterling Instruments (SDP/SI). The company SDP/SI was recommended by instructors and they are the leader in the design and manufacture of precision mechanical components, such as timing belts and timing belt pulleys. For the input shaft of the mechanism a 10 tooth plastic timing belt pulley would be purchased from SDP/SI that is listed as part A6Z3-10SF03708. The output shaft would be connected to a larger pulley of 60 teeth which is part A6K360NF093712. These two pulleys were driven by a timing belt listed as part A6R3-114037. These three components are all compatible with each other with a XL pitch (0.2 in.).

4.3 Mounting Design

4.3.1 Discussion of Effects on Gate One Values

In comparison to previous calculations, the changes that were made to the mechanism would increase the total volume, horizontal offset and forward offset of backpack holder from the armrest. The addition of the transmission would increase the height of the mechanism. The output pulley would be located above of top clamp of the input linkage decreasing the height from an approximated value of 10 in. to 6.819 in. which decreases the estimated volume of 1367 $in.^3$ to 896.09 $in.^3$. After testing and torque calculations it was considered beneficial to for performance to move the ground link backwards 1 inch and 0.75 left if looking at a top view relative to the mounting brace of the wheel chair as shown in *Figure 4-3*.The link was adjusted because of the bottom of the backpack coming into contact with wheelchair consequently increasing the effective torque at the end of the path. With moving the link to this new location torque decreased from 1289.33 oz-in. to 804.405 oz-in. during that last few degrees of motion. The mechanism after adjustment was still well within the design constraints of the project description.

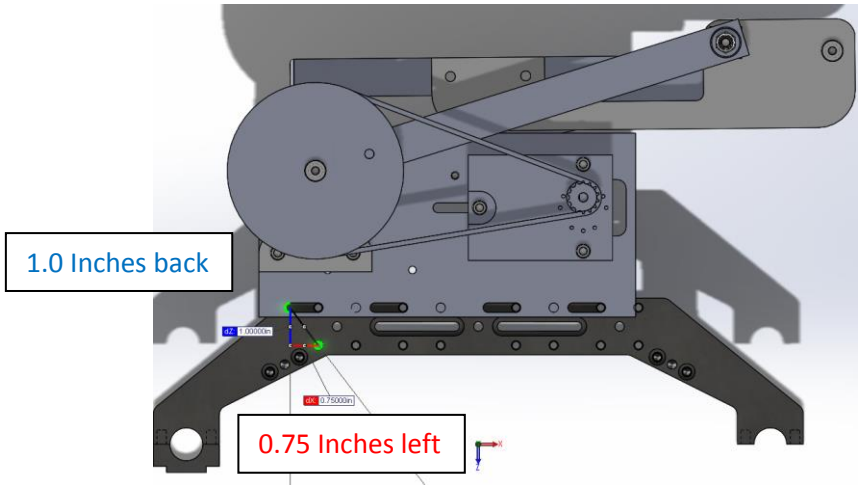


Figure 4-3: Top view of translation of ground link

4.3.2 Discussion, Explanation and Cross-sections of Transmission Design

In the cross section, it shows all the components of the transmission. The motor mount ensures that the motor remains vertical and fixed relative to the ground link during operation. For ease of installation and tensioning of the timing belt the motor mount was designed with slotted holes which were easily shown in *Figure 4-5*. The motor would transmit power via set screw in an aluminum insert of the pulley. Pulleys would each have a hub to resist the timing belt from sliding off. These hubs would be located on opposite sides to ensure the position of the belt. Although the output pulley and in the input link are revolving around the same shaft the power is transferred from the pulley to the link through the steel drive shaft shown in yellow within *Figure 4-4*.

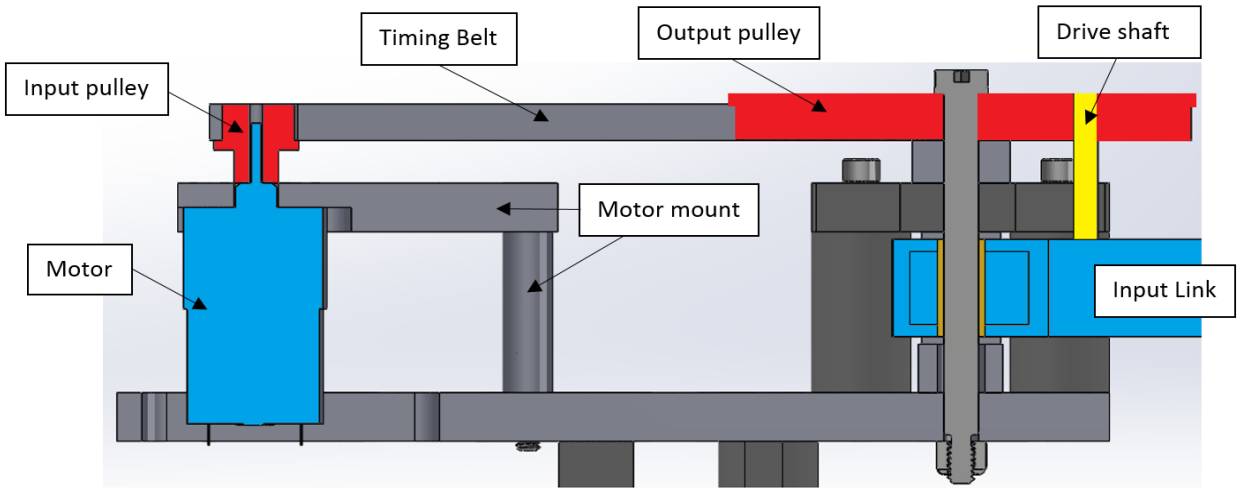


Figure 4-4: Cross section view of transmission

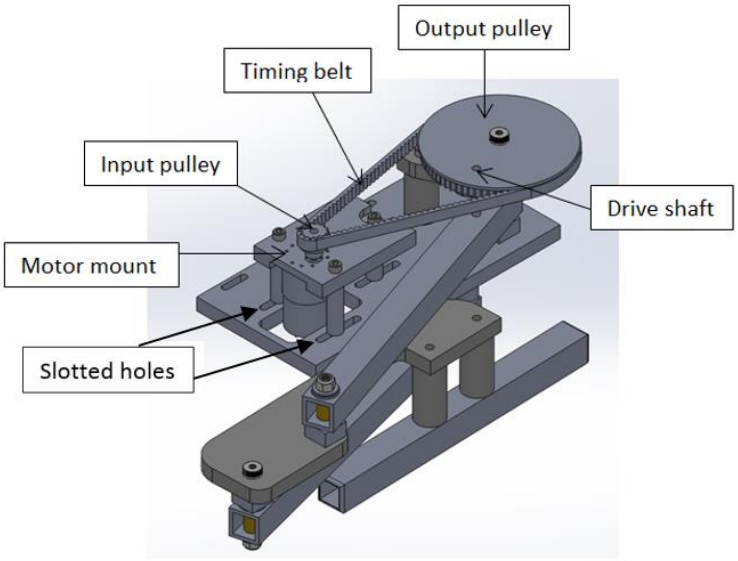


Figure 4-5: Motor mount slotted holes

4.4 Transmission Loading Analysis

4.4.1 Transmission joint stress analysis

a) Off-axis pin on the input link

The input link was mounted onto the pulley through an off-axis bolt with diameter $r_{axis} = 0.125$ in. as shown in *Figure 4-6*.

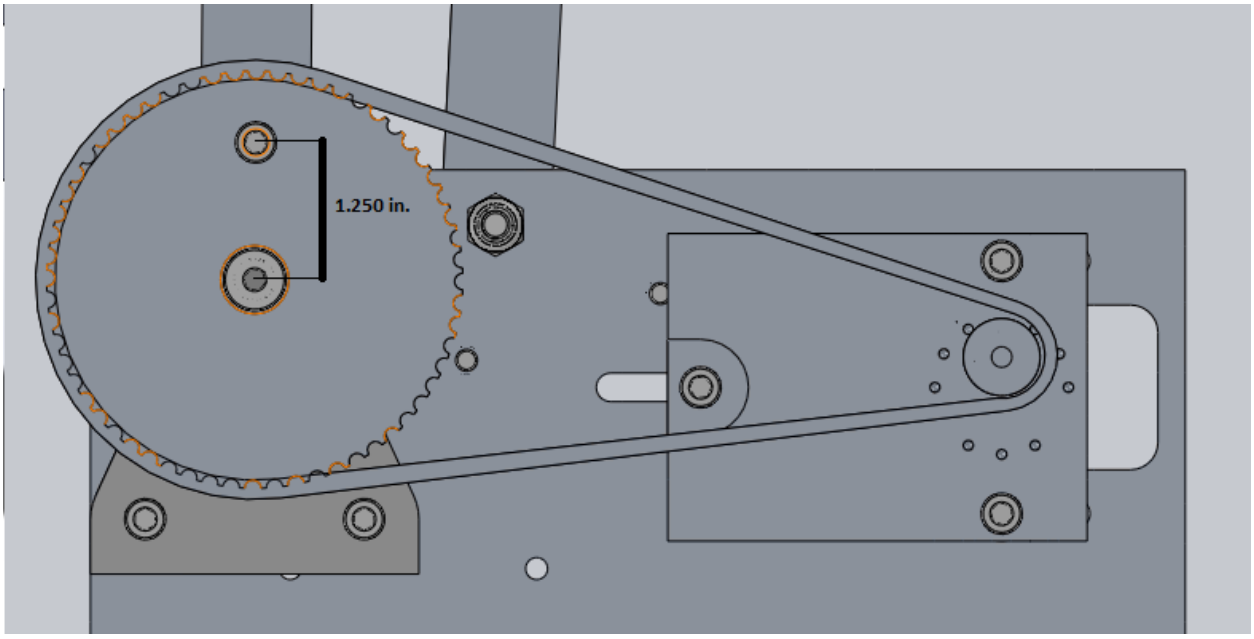


Figure 4-6: Off-axis pin location

The distance of the center of the off-axis pin to the center of the shaft is d_{offset} =1.25 in. The maximum average stress on the off-axis pin is given as

$$S_{axis} = \frac{T_{MA}}{d_{offset} \cdot \pi r_{axis}^2} = \frac{804.405 \text{ oz} \cdot \text{in.}}{1.25 \text{ in.} \times \pi \, (0.125 \text{ in.})^2} = 81.936 \text{ psi} \qquad \text{Equation 18}$$

The stress on the pin was very small compared with the ultimate tensile strength of steel shaft used as the off-axis pin whose magnitude is around 50 ksi to 100 ksi.

b) Set screw for the hub on the motor shaft

The hub was used to mount the pulley on the motor shaft. The diameter of the set screw on the shaft is given by $r_{set} = 2$ mm $= 0.0787$ in. n. The radius of the motor shaft is $R_{shaft} = 2.4$ mm $= 0.0945$ in.

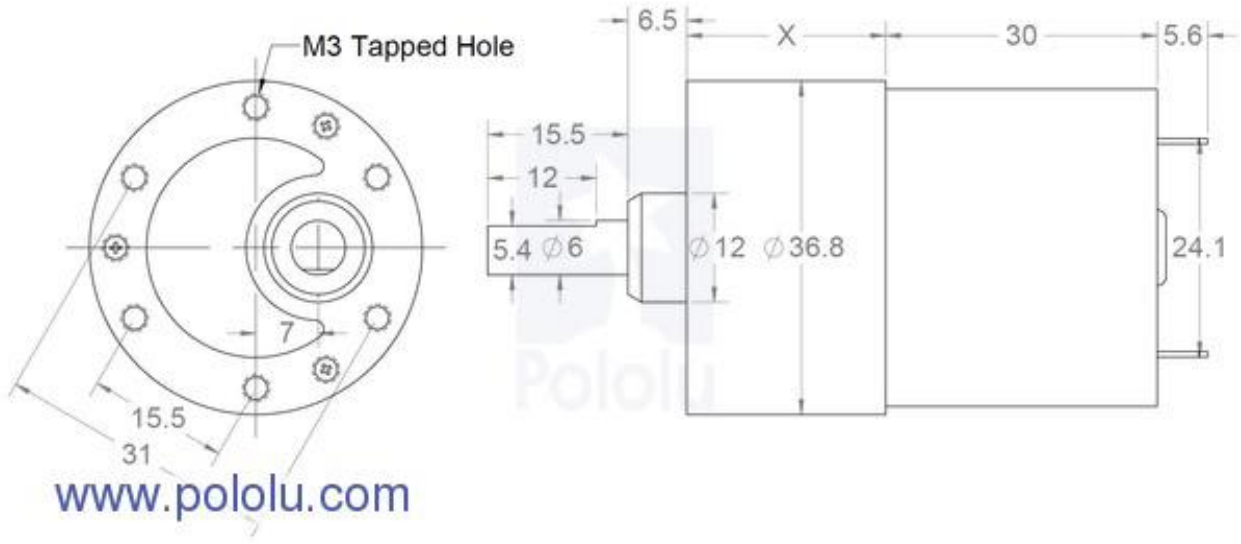


Figure 4-7: 37D mm metal gearmotor dimensions (units in mm)
(retrieved from 1447-Pololu data sheet)

The maximum average stress on the set screw is given as

$$S_{set} = \frac{T_{MA}}{N \cdot R_{shaft} \cdot \pi r_{set}^2} = \frac{804.405 \text{ oz} \cdot \text{in.}}{6 \times 0.0945 \text{ in.} \times \pi \times (0.0787 \text{ in.})^2} = 4.556\ ksi \qquad \text{Equation 19}$$

The stress on the pin was small compared with the ultimate tensile strength of steel shaft used as the off-axis pin whose magnitude is around 50 ksi to 100 ksi.

As for the holding power analysis, the set screw holding power for #4 set screw used in this design can be found to be 160 lbf. Since there is no axial force for the small pulley,

$$F_{axil} = 0 \text{ lbf} < 160 \text{ lbf} \qquad \text{Equation 20}$$

$$F_{circular} = \frac{T_{MA}}{N \cdot R_{shaft}} = \frac{804.405 \text{ oz} \cdot \text{in.}}{6 \times 0.0945 \text{ in.}} = 87.6 \text{ lbf} \qquad \text{Equation 21}$$

This has shown that a #4 screw would provide large enough holding power for the mounting the small pulley onto the shaft due to the negligible small axil load and D shape motor shaft for circular load.

4.4.2 Deflection Analysis on Motor Mounts

In order to solve for the deflection of the motor mount, the operating tension force on the timing belt needs to be determined first. Due to the teeth of the timing belt, the initial tension is not proportional to the friction required to synchronize the motion of the pulley and the timing belt. For the purpose of analysis, the tension of the slack side of the belt in operating state is assumed to be zero. Since the belt cannot support compression load, the initial tension under the previous assumption should be half of the torque required divided by the radius $r_{MotorPulley} = 0.3185$ in. of the input pulley. In this case, the tension on the tight side F_t of the belt is given by

$$F_t = \frac{T_{MA}}{N \cdot r_{MotorPulley}} = \frac{804.405 \text{ oz} \cdot \text{in.}}{6 \times 0.3185 \text{ in.}} = 26.304 \text{ lbf.} \qquad \text{Equation 22}$$

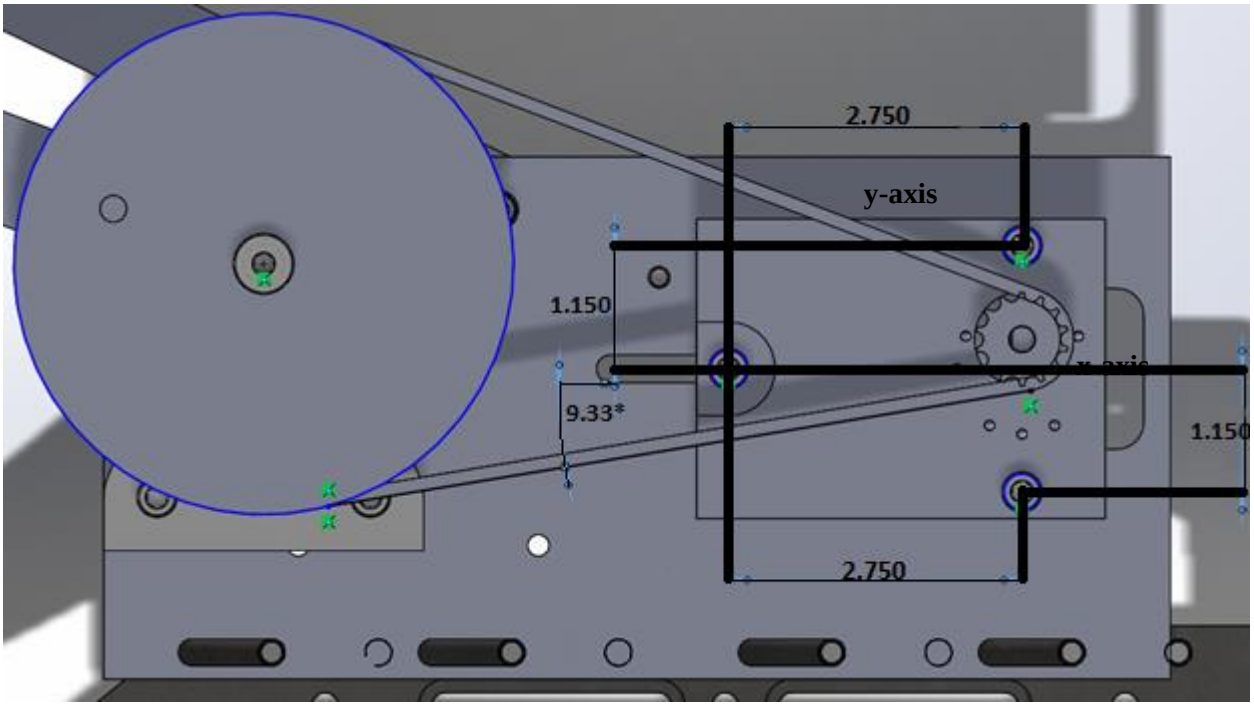


Figure 4-8: Deflection analysis geometry

The inclined angle of the applied force as shown in *Figure 4-8* was given by $\theta = 9.33$ deg.

Considering the small area of the support compared with the motor mount itself, which is a plate and the orientation of the force, the deflection of the motor mount can be viewed as a multi-axil bending problem with respect to the motor mount support. The area of moment of inertia with the coordinate system established at the neutral axis as shown in *Figure 4-8* can be calculated by using parallel axis theorem. For the neutral axis, by using the geometry property of triangle, the parameters are given as $d_1 = 1.150$ in., $d_2 = 2.750$ in.$\times 2/3$ and $d_3 = 2.750$ in.$\times 1/3$. In addition, the radius of the supporting stock is $r_{support} = 0.25$ in. .The area moment of inertia is given by

$$\begin{cases} I_x = 3\dfrac{\pi r_{support}^4}{4} + 2d_1\pi r_{support}^2 = 0.461 \text{ in.}^4 \\ I_y = 3\dfrac{\pi r_{support}^4}{4} + (d_2 + 2d_3)\pi r_{support}^2 = 0.729 \text{ in.}^4 \end{cases}$$

Equation 23

View the vertically upward motor mount as a cantilever beam with length $h = 3$ in., the deflection with respect to x-axis D_x and y-axis D_y can then be calculated as

$$\begin{cases} D_x = \left|\dfrac{-F_t \cdot h^3 \cdot \sin\theta}{3EI_x}\right| = 6.94 \times 10^{-6} \text{ in.} \\ D_y = \left|\dfrac{-F_t \cdot h^3 \cdot \cos\theta}{3EI_y}\right| = 2.67 \times 10^{-5} \text{ in.} \end{cases}$$

Equation 24

Based on the previous analysis, the deflection of the motor mount in this case is very small compared to the dimension of the mobile backpack holder and was therefore negligible. The motor mount design is stiff enough.

5 Safety and Motor Controls

5.1 Introduction

This section gives detailed analysis on the safety, and motor control used in the mobile backpack holder system for the purpose of control. Safety is an important criterion for good designs. To guarantee the safety for the users, all the wires should have no metal exposure at the surface, and the mechanism should have solid assembly so that no part will break off during operation. Besides, the mechanism should always be in control by users when the device is functioning. It requires good motor control technique by correctly applying Arduino code, and setting up sensors properly. In this project, an open loop control method was used to control the motor with the right speed. Two infrared proximity sensors were attached on the input link and they were constantly providing the Arduino with distance feedback, which was compared with a preset threshold value, in order to tell whether an obstacle was detected.

5.2 General Structure

For a small scale embedded system such as the mobile backpack carrier, the control system involves a central controller, several sensors for the measurement and actuators. The materials used for the control system are listed in *Table 5-1*.

Item	Model and Suplier	Quantity
Arduino Uno microcontroller board	DEV-09950-Sparkfun	1
400-point Breadboard	351-Polulu	1
H-bridge (L298 Motor Driver -- preassembled)	K CMD- Sparkfun	1
Rocker switch, Single-pole, Double-throw	35-695-Jameco	1
Infrared Proximity Sensor, short range	GP2D120XJ00F -Sparkfun	2
Snap Action Switch (limit switch)	187733-Jameco	2
Arduino Sensor Shield	V5.0-Taobao (Brought from China)	1
Arduino IIC Liquid Crystal Display	1602-Taobao (Brought from China)	1
Remote Controller	2262-Taobao (Brought from China)	1

Table 5-1: Materials used for the control system

5.3 Controller and Amplifier

5.3.1 Arduino Uno Microcontroller

The Arduino Uno microcontroller board shown in Figure *5-1* was chosen as the controller for the system.

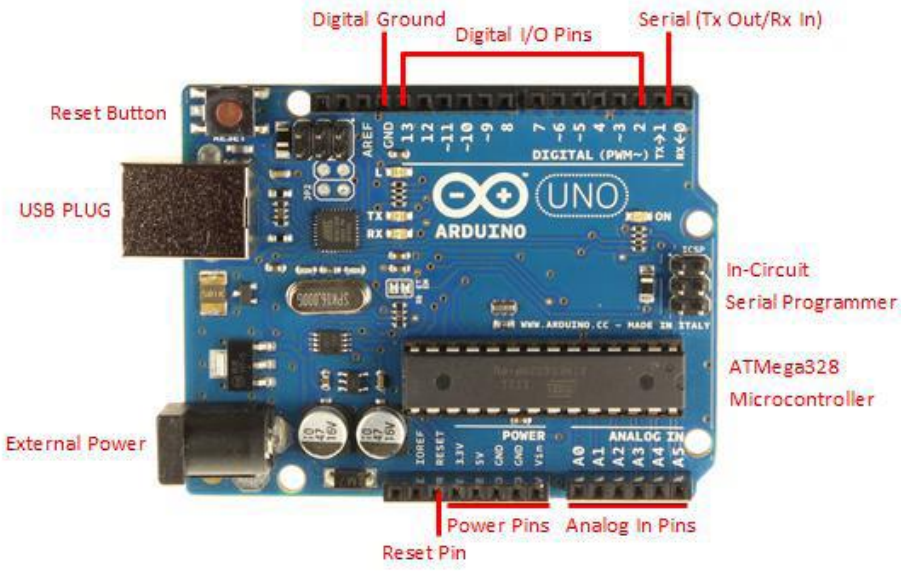


Figure 5-1: Arduino Uno Microcontroller

The Ardunio Uno microcontroller board contains 14 digital pins and 6 analog pins. All 20 pins have the capability of reading and writing digital signal (i.e. 0V and 5V) using the Arduino command **digitalRead(pinNo)** and **digitalWrite(pinNo,status)**. The difference between digital pins 0 to 13 and analog pins A0 to A5 lies in the fact that analog pins have the capability of reading analog signal from sensor input as a value ranging from 0V to 5V and map the value corresponding to a discreet set of values ranging from 0 to 1023 using the Arduino command **analogRead(pinNo)** while this command is not applicable to the digital pins. Moreover, digital pins with a tilde before pin number (i.e.3,5,6,9,10,11 on the Uno board) can also be used for the Arduino command analogWrite(pinNo,value) which outputs signal in the form of pulse width modulation that can help to control the output power through adjustment of duty cycle. The pulse width modulation value can be specified using a set of discrete integer number ranging from 0 to 255.

The Ardunio Uno microcontroller is programed using the classic C language in the Arduino integrated development environment which uses serial communication for compiled code upload and data transmitting between the board and the computer. Besides serial communication, the Arduino Uno microcontroller also supports SPI and IIC communication in which the IIC communication has been used to control a liquid crystal display for showing the status of the mechanism.

5.3.2 Sensor Shield

The functional diagram of Arduino sensor shield and the real component used in this design are shown in Figure *5-2* and Figure *5-3* respectively. Despite the seemingly complicated functional diagram of the sensor shield, its main function is to give more VCC and GND pins to provide more 5V power source to sensors since there are only 2 GND and 1 5V VCC and 1 3.3V VCC on the Arduino Uno board despite the plug for power input for the Arduino Uno board itself. For the control system, the sensor shield is used to ease the process of wiring on the bread board.

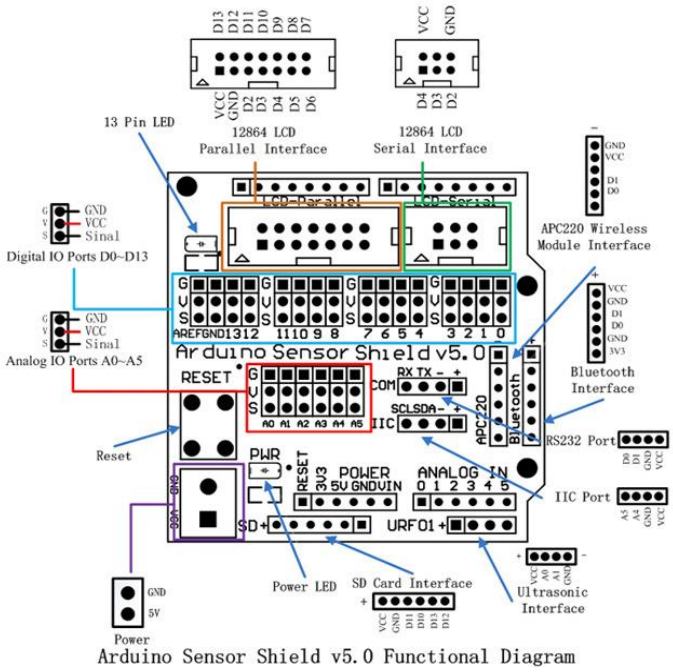


Figure 5-2: Arduino Sensor Shield Functional Diagram

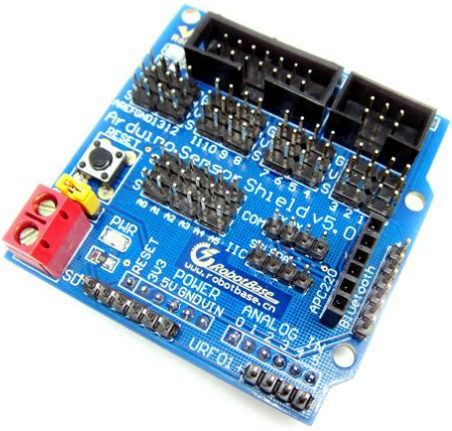


Figure 5-3: Arduino Sensor Shield

5.3.3 H-bridge and Breadboard

The H-bridge and bread board used in the control system are shown in *Figure 5-4* and *Figure 5-5*. These two parts were used to connect the Arduino Uno board to actuators, sensors and switches.

Figure 5-4: H-bridge

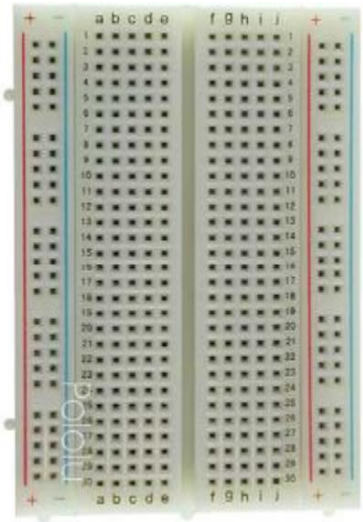


Figure 5-5: Breadboard

The purpose of the H-bridge is to amplify the control signal provided by the Arduino Uno microcontroller and provide power to the motor to carry out the motion. Since the voltage output of the Arduino Uno board is no more than 5V, which is not enough to drive the motor of which the desired operating voltage is 9V for safety concern. More importantly, the output current for each pin of the Arduino Uno board is limited to 40mA, which puts a limitation on its maximum output power. Therefore, the H-bridge is necessary as it converts the control signal to actuator voltage input.

The purpose of the breadboard is to connect the Arduino Uno board to other sensors. Holes on the + and – bus on left and right side of the board are connected vertically to each other respectively to serve as power source. Holes that lie on the same horizontal line a-e and f-j are connected to each other respectively for the wiring of Arduino Uno pins to sensor and H-bridge pins.

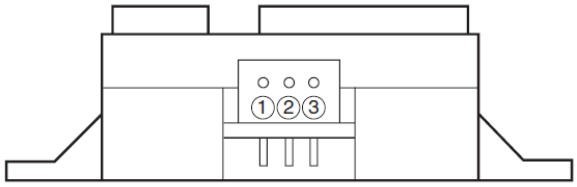
5.4 Sensor Capabilities and Limitations

In order to control the motion of the backpack carrier and increase the level of safety as much as possible, several sensors were used for information feedback to the controller. Sensor characters are discussed in this section to reflect its corresponding usage.

5.4.1 Infrared Proximity Sensor

The real part and pin arrangement of the infrared proximity sensor are shown in *Figure 5-6* and *Figure 5-7* respectively. The infrared remote sensor sends out a beam of infrared light and collects the amount of infrared remote light that was reflected and convert it to analog voltage on the signal pin. If there is an obstacle close to the sensor, the reflected light would increase and the output voltage on the signal pin would increase correspondingly. Due to the infrared light, this sensor is able to determine whether there are objects approaching before making actual contact. This capability of remote sensing makes the infrared proximity sensor an ideal candidate for a safety switch that can prevent objects to jam the path of the mechanism.

Figure 5-6: Infrared proximity sensor



PIN	SIGNAL NAME
①	V_O
②	GND
③	V_{CC}

Figure 5-7: Infrared proximity sensor pin

The limitations of the infrared proximity is that the output analog voltage signal does not exactly reflect the distance of the object to the sensor itself accurately due to the fact that the absorption rate of the infrared light for different objects. This means the threshold of the sensor for object detection has to be determined by testing. Also, the measuring range of the infrared remote sensor is limited to the direction it is pointing at.

On the other hand, as there is no sudden change for the infrared proximity sensor voltage signal, the interrupt mechanism on the Arduino Uno controller which works for rising and falling edges of digital signal does not react to small voltage changes provided by this sensor. This character results in the obstacle detection process being carried out by polling through looping instead of using the interrupt mechanism which is more efficient.

5.4.2 Limit Switch

The limit switch used in this project is shown in Figure *5-8*. The switch is a mechanism that can close the loop of the circuit whenever pressed. The advantage of the limit switch is that it has a spring mechanism that opens the switch whenever there is no applied force. This makes the limit switch an ideal candidate to tell whether the mechanism was in contact with the hard stop or not.

Figure 5-8: Limit switch

The disadvantage of the limit switch is that since it requires contact with the mechanism, which would reduce the life of the sensor as wear happens during physical contact. On the other hand, the voltage signal of the sensor can be pulled up by the 5V voltage but cannot be pulled down to 0V directly without

a proper connection to the ground. This means a circuit consisting of a resistor connecting the ground and the sensor is necessary for the purpose of pulling down the voltage to 0V when the switch is open.

5.4.3 Rocker Switch

The rocker switch used is shown in Figure *5-9*. The rocker switch is very similar to the limit switch in terms of its advantage and disadvantage. The difference is that the rocker switch consists of three pins for that can serve as two switches. Whenever the right side is pressed, the right pin and the middle pins are connected. If the left side is pressed, the left pin and the middle pins are connected. Combined with the shape of the switch, the rocker switch is ideal for the purpose of user control.

Figure 5-9: Rocker switch

The disadvantage of the rocker switch is also the voltage pull down issue. In this case, this was resolved by adding an additional parallel remote control for both remote control and voltage pull down when the switch was not pressed.

5.4.4 Digital Encoder

The digital encoder attached to the motor is shown in Figure *5-10*. The encoder has two channels giving out 16 high voltage impulses for each rotation of the motor respectively with 90 degrees of phase offset for rotation direction determination. The output is shown in Figure *5-11*.

Figure 5-10: Encoder on motor

Figure 5-11: Encoder voltage output

The capability of the encoder is to measure the direction and rotational speed by output digital signal corresponding to the angle of the motor armature. The two phase measuring mechanism increases the reliability of the measurement and enabled the determination of the direction of rotation.

The limitation of the encoder is its resolution of the armature. Since only 16 digital impulse in two phase are given for each rotation of the armature, there is no way to determine more accurately about which angle the motor is at within a range of 5.625 degree, which is obtained by the full rotation divided by 64. This would result in measurement error. Although the measurement uncertainty seems a large number in this case, it is divided by the gear ratio which is 786 due to the design of the transmission. The uncertainty is negligible and can be calculated as below.

$$\text{Uncertainty} = \frac{360 \text{ deg}}{16 \times 4 \times 131 \times 6} = 7.156 \times 10^{-3} \text{ deg}$$

5.5 Additional Components

5.5.1 Liquid Crystal Display

The liquid crystal display used for user interface is shown in *Figure 5-12* and *Figure 5-13*. This display uses IIC communication to acquire information from Arduino Uno board and display them on the screen.

Figure 5-12: Liquid crystal display front

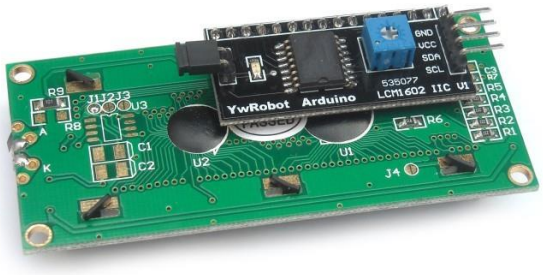


Figure 5-13: Liquid crystal display back

5.5.2 Remote Control

The liquid crystal display used for user control is shown in *Figure 5-14*. The remote control device uses radio communication to send out four different signals corresponding to four different buttons. Two buttons are usedparallel with the rocker switch to control the mechanism.

Figure 5-14: Infrared remote control

5.6 Threshold Value Determination

5.6.1 Infrared Proximity Sensor

The threshold value for the infrared proximity sensor was determined experimentally. Using five volts input for the infrared proximity sensor, the voltage output of the signal pin was measured using the Arduino command **analogRead(pinNo)** with output range 0 to 1023. In order to stop the mechanism whenever there were obstacles exist on the path of the mechanism, the threshold value was set to 100 for the infrared proximity sensor. When the measured voltage value on the signal pin of the infrared proximity sensor was larger than the value threshold value of 100, obstacle was approximately 20 centimeters away from the infrared proximity sensor. The 20 centimeters distance threshold value was determined for safety concern while the threshold value of 100 was obtained by putting the hand 20 centimeters away from the sensor and taking the measurement.

5.6.2 Encoder Sensor

The threshold value for the encoder was obtained by multiplying the gear ratio and the sampling mode selected. In order to increase the efficiency as well as the accuracy of the measurement, the interrupt method was used. Taking the voltage impulse measurement of encoder line A to the interrupt trigger pin (digital pin 2 for Arduino Uno board), the mode was set to **CHANGE** to get 32 interrupts for each rotation of the motor armature. For the design, the rotation angle of the output link was set to be 200.6 degree and the slow-down angle requirement was 30 degree. The encoder count was then found as the following.

$$\text{Forward} = \text{Angle} \times \text{GearRatio} \times \text{Rate} = \text{Ceiling}\left(170.6\ \text{deg} \times 6 \times 131 \times \frac{32\ \text{Count}}{360\ \text{deg}}\right) = 11920\ \text{Count}$$

$$\text{Backrward} = \text{Angle} \times \text{GearRatio} \times \text{Rate} = \text{Ceiling}\left(30\ \text{deg} \times 6 \times 131 \times \frac{32\ \text{Count}}{360\ \text{deg}}\right) = 2096\ \text{Count}$$

This gives the value of the expected encoder count for the mechanism to finish its full rotation. Due to the presence of the limit switch, the encoder threshold was not necessary for the entire rotation.

5.7 Arduino Code Change

5.7.1 General Structure

Although the sample code for the backpack holder was provided, the code used for controlling the mechanism was recreated from scratch for customization. The detailed code with comments can be found in **Appendix D.2**. The general structure of the Arduino code is introduced in this section.

The aim of the code is to control the movement of the mechanism accurately. The mechanism should be able to trigger an emergency stop upon sudden presence of obstacle and restore to its normal operating condition with the rocker sensor of remote control input. To realize the function mentioned above, a state machine is an ideal structure for keeping track of the current status and move on to other status based on external input.

In order to increase the efficiency of the code, an interrupt mechanism was used to keep track of the encoder. Meanwhile, polling was used for obstacle detection due to the analog signal output of the infrared proximity sensor. The motor control algorithm was also carried out using polling as the Arduino Uno microcontroller continues looping through its main code.

The state machine used in the main loop of the function consists of 5 states in total enumerated from 0 to 4. State 0 is waiting mode where the mechanism is in steady state waiting for user input or stopped due to the presence of external objects presenting in the path of motion. The state would be changed to 1 or 2 depending on the user input when one of the limit switches indicates that the linkage is at one final state. The state would be changed to 3 or 4 depending on the user input when both of the limit switches indicate that the linkage is at neither of the final state. This means that the linkage is stopped during the motion due to the presence of external objects or sudden loss of power input.

State 1 and state 2 are full turn mode in which the mechanism makes a full turn with user input direction. These two states would return to state 0 under the condition when the mechanism was in contact with one of the limit switch or external obstacles are detected by the infrared proximity sensors.

State 3 and state 4 are fix turn mode in which the mechanism turns slowly to its final position with user input direction. These two states would return to state 0 under the condition when the mechanism is in contact with one of the limit switch or external obstacles are detected by the infrared proximity sensors.

The algorithm of the state machine is shown in Figure *5-15*.

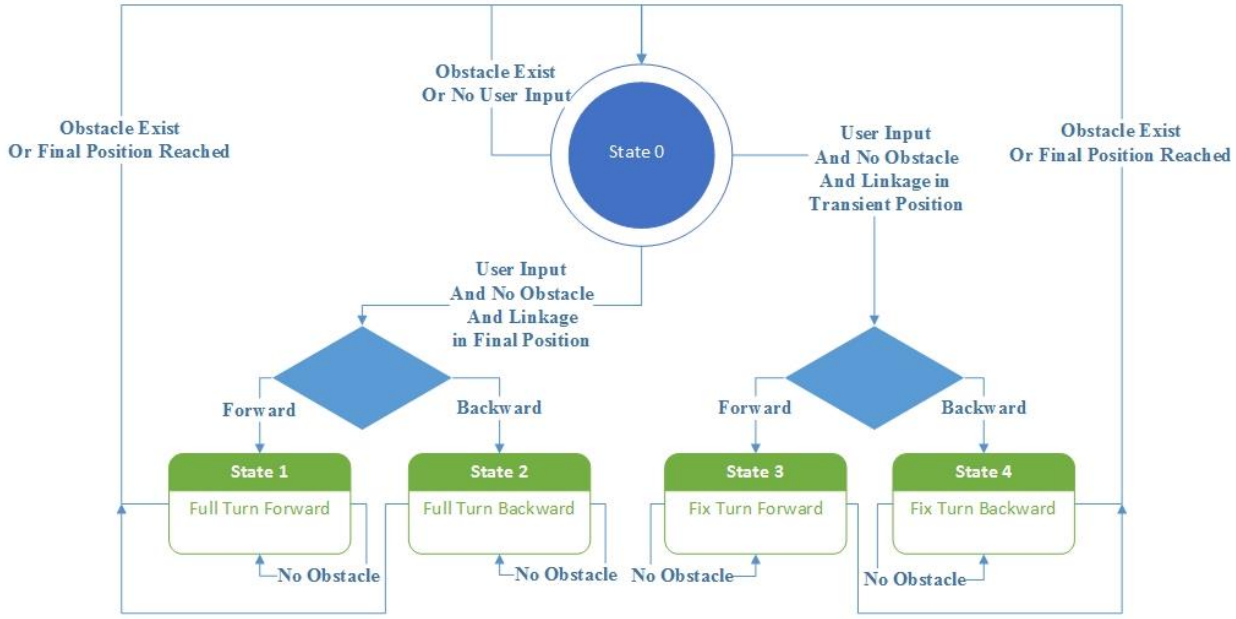


Figure 5-15: State machine algorithm diagram

5.7.2 Pin Definition

All 14 digital pins from pin 0 to pin 13 and 4 analog pins including pin A0, A1, A4 and A5 were used for some specific purpose with some redundancy for the controlling of the mobile backpack carrier. The detailed pin assignment is shown in Table *5-2*.

Pin Number	Pin Alias in Code	Pin Connection Function
0	None	Left unconnected for using serial portal to debug
1	None	Left unconnected for using serial portal to debug
2	emergenceStop	Used for emergency stop button
3	encoderA	Used for encoder counting measurement
4	encoderB	Used for encoder counting measurement
5	motorP	Used for H-bridge positive pin signal control
6	motorN	Used for H-bridge negative pin signal control
7	hardStopFront	Used for measurement of front hard stop limit switch
8	hardStopBack	Used for measurement of back hard stop limit switch
9	rockerFront/remoteA	Used for signal collection of rocker switch and remote pin A
10	rockerBack/remoteB	Used for signal collection of rocker switch and remote pin B
11	remoteC	Connected to remote pin C (Not used in this case)
12	remoteD	Connected to remote pin D (Not used in this case)
13	buzzer	Used for buzzer signal control
A0	proximityA	Used for signal collection from infrared proximity sensor A
A1	proximityB	Used for signal collection from infrared proximity sensor B

A2	None	Not used
A3	None	Not used
A4	sdaPin	Used for data line signal trasmiting
A5	sclPin	Used for clock line signal trasmiting

Table 5-2: Pin number definition

5.7.3 Function Definition

In order to simplify the code by grouping similar structure that appears multiple times in different part of the main code, customized functions were defined to carry out standard operation so that it can be executed with one line of code. The functions defined in the application to control the mobile backpack holder are listed in table.

Function Definition	Functionality
void setup()	Arduino default initialization
void loop()	Arduino default main loop function
void stopAll()	Interrupt and stop all linkage movement
void recordEncoder()	Record the current encoder measurement
float encoderSpeedRead()	Calculate the speed of the encoder
float convertToMotorSpeed(float encoderSpeed)	Convert encoder speed to motor speed
void motorControl(int com)	Control the motor speed and direction
int piControl(float targetSpeed, float currentEncoderSpeed)	Use PI control algorithm for speed control
int obtainPosition()	Check the current position of the linkage
float obtainAngle()	Obtain the mechanism angle
int checkObstacleExist()	Check the infrared proximity sensors
void fullTurn(int dir)	Make full turn based on direction command
void fixTurn(int dir)	Make fix turn based on direction command
void lcdAnglePrint(float angle)	Print angle message on LCD screen
void printMessage(String msg, int ifPrintAngle, int times)	Print message on LCD screen
void saveAngle()	Save the current angle to ROM

Table 5-3: Arduino function definition

The detailed realization of each function can be found in the source code attached in **Appendix D.2**.

5.7.4 Important Variable Value

In order to realize the ideal motion of the linkage, several pre-calculated variables were used in the code for the purpose of control. The value of the variables can be found in Table *5-4*.

Variable Name	Device	Calculated Value	Used Value
safeRange	Infrared Proximity Sensor	N/A	100
slowDownAngle	Motor (Linkage Position)	30	65
maxAngle	Motor (Linkage Position)	200.6	205
highSpeed	Motor (PWM Duty Cycle)	255	240
lowSpeed	Motor (PWM Duty Cycle)	N/A	165

Table 5-4: Variable value

5.7.5 Controller Tuning

The mobile backpack carrier was controlled by an open loop control mechanism in which the high speed and low speed mode were realized by changing the duty cycle of the pulse width modulation (PWM)

signal. The value used by the PWM pin on the Arduino control board ranges from 0 to 255 and was determined experimentally to be 240 for high speed and 165 for low speed to realize the ideal motion.

In order to control the mechanism more accurately, a PI controller was implemented to control the mechanism. Due to lack of knowledge for variable values such as the inductance of the motor and reflected moment of inertia of the mechanism, it was very difficult to tune the PI controller to obtain the ideal motion and therefore the controller was not used for the testing.

5.8 Sensor Mount

The mounting positions of the infrared proximity sensors are critical to their functionality. The basic criterion is that sensors should not sense any part of mechanism itself, or any part of the wheelchair, when the design is in operation. The locations of the two sensors in this project were determined to be fixed on the top and back side of the input link. They were firmly attached to the link by double-sided tapes. The mounting of the two infrared proximity sensors did not change the original volume or offset.

Two limit switches were utilized for feedback of the current position of the linkage system. The switches were mounted to the cylinder by using double sided tape. The switches were mounted right beside hard stop for the purpose of getting the direct feedback.

6 Design Critique

6.1 Final Testing Data

With the completion of the mechanism, final testing was conducted to evaluate the performance of the mechanism. The test was conducted using a LabVIEW data acquisition device. Screen shot of the test program front panel is shown in Figure *6-1*.

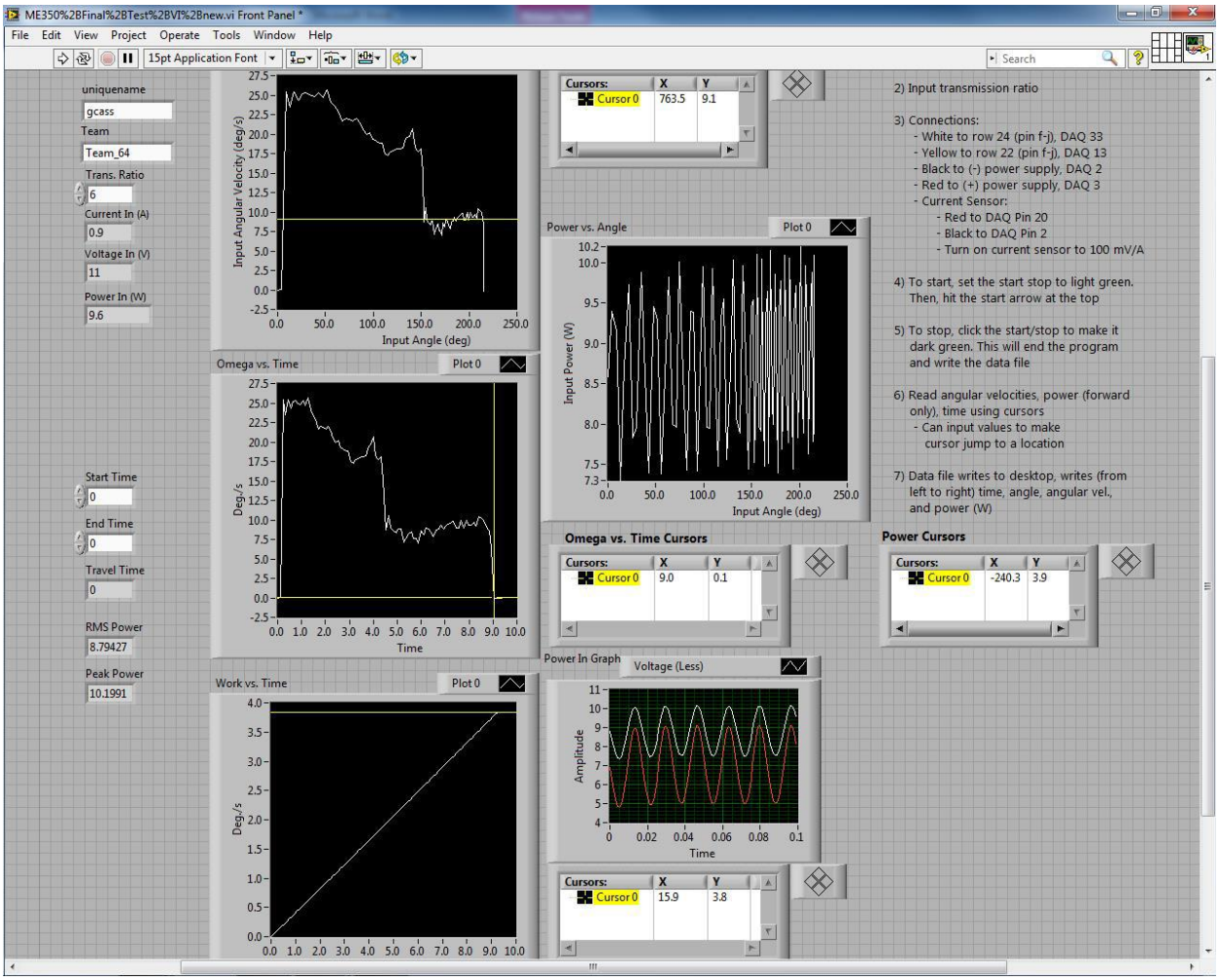


Figure 6-1:Test data on LabVIEW front pannel digram

The result of the test is summarized in Table *6-1*. As can be seen from the table, final testing values all fall within the required range. More importantly, design parameters such as the horizontal offset, travel time and angle offset were within 5 percent of the range specified. These parameters were further optimized for convenience and safety concern.

Design Parameter	Target	Min	Max	Expected Value (with SF)	Final Testing Value	Final Value within range of Min/Max
Volume	Minimize	N/A	3000 in3	1367.2in^3	877 in^3	Yes

Forward Offset	17 in	6 in	17 in	12.58 in	11.125 in	Yes
Horizontal Offset	2in	2 in	10 in	2.078 in	2 in	Yes
Angle Offset	0 degrees	-15 degrees	15 degrees	0 degrees	0.5 degree	Yes
Travel Time	9	6 sec	12 sec	9 sec	8.9 sec	Yes
Decelerated Speed	Minimize	3 degrees per sec	10 degrees per sec	10 degrees per sec	9.1 degrees per sec	Yes
Peak Power	Minimize	N/A	N/A	24.132 Watts	0.4270 Watt	Yes

Table 6-1: Final testing data summary

6.2 Problems Encountered

During the assembly process, two major issues that had not been taken into consideration previously emerged.

The first issue is related to the front hard stop of the mechanism. Due to the fact that the front hard stop was very close to the joint between the ground link and the input link, the inaccuracy due to the manufacture of the hard stop cylinder diameter was magnified and resulted in an approximately 2 degree angle offset. In order to solve the problem, one thick layer of tape was employed to reduce the angle offset from 2 degree to under 0.5 degree. The problem was almost resolved but the appearance of the mobile backpack carrier was degraded.

The second problem found during prototyping was the hardness in tuning the parameter for the PI controller. Due to lack of knowledge for the moment of inertia of the structure and all the values of the electric components, the tuning process went onto the track of try and error. It took more than two weeks for the tuning process but the result for the closed loop PI control mechanism was still not ideal. In the end, an open loop control mechanism was chosen instead of the PI controller due to the relatively easy tuning procedure of the first one. As can be seen from the result, the travel time of the mechanism is very close to the ideal value of 9 sec.

As for a recommendation, the outer diameter of the cylinder for the hard stop of the mechanism should be enlarged. More measurement should be taken regarding the value for different crucial variables of the control structure.

6.3 Design Critique

After completing the manufacturing of the mobile backpack carrier, some possible improvements were realized. The designed device was able to meet all the design specifications as most of the parts went well as expected. However, small adjustment could be made to raise the quality of the design.

The device was made adjustable in order to allow for error and to optimize it. The motor mount was adjustable so that the timing belt can be easily attached and tightened. There are also slots on the ground link for attaching to the wheel chair. By doing so, the position of the backpack became adjustable so that the backpack will not touch the wheel chair during motion without costing to much offset. Another thing

that worked well was the use of joint supports as hard stops. The placement of the supports allowed the final position of the backpack holder to be parallel to the armrest.

When the device was modeled in Solidworks and ADAMS the torque on the input link, the deflection or our linkage, the total volume of the backpack carrier, and the degree offset were all simulated. In ADAMS the input torque was modeled at 189 Nm, this was higher than the actual torque on the device, which was around 100 Nm. This happened due to reductions in friction by the additional components that were added. The deflection of the device was negligibly small and had no effect on the performance of our mechanism. The input and output links were double supported in order to reduce deflection when designed. Calculations showed that the deflection should be minimal. The design parameters proved to be ideal during the testing.

Loading the device with the 12 lb-m backpack induced a moment on the linkage. This slowed the speed of linkage movement since it needed a higher torque to move the backpack, which caused the device to draw more power. The additional moment from the backpack increased the friction in the joints which was part of the reason the linkage slowed down. More importantly, since the whole device had a three degree angle offset from horizontal, the linkage had to overcome the added 12 lb-m weight force as it moved into the final position.

One of the goals was to minimize friction on the device. To do this, eight needle thrust bearings were used at each joint of the input link and the output link. Bronze bushings were used at the joints of the input link and output link to reduce friction between the shoulder bolt and the link. There was also potential friction between the driven gear and the attachment point, to reduce this, the shoulder bolt that connected the gear to the link was kept slightly lose to allow free movement.

The control algorithm for the device was improved to make the device more user friendly. A remote control was added so that the operator could easily access their backpack. An liquid crystal display was also added to the device to display important information about the use of the device. If the device stops moving, the screen would output the reason such as an obstacle preventing the movement of the linkage. An EEPROM library was also added to the code to store the position of the linkage at all times. This allows the device to remember its position in the case of loss of power.

To better use the provided infrared proximity sensor, the threshold should be adjusted at multiple points in order to prevent false alerts and also increase safety. This would take time and is not necessarily needed in the prototype, but could be added in the final version of the device.

The horizontal offset of the design's final moving position is strictly followed by the specification given by professor, two inches. During real application, the offset value turns out to be too small because the backpack tends to lean backwards at the bottom. As a result, the backpack will touch the wheel during the final moving period, which produces a strong resisting force for the linkage to reach the final position. In a final version of the device the horizontal offset would be increased to 5 in to reduce the resisting force

On a scale of 1-5, the safety of the design would be rated as a 5. The device is solid and robust; it was designed to limit any unwanted movements and weak points. It rotates smoothly and never breaks apart during all the testing. The arrangement of all the wires were designed to make sure that they are put in good order and no metal exposure can be found. The added remote control system helps the wheelchair

occupant control the design alone easily. The provided sensors are adequate to make the mechanism function well.

The device is ready to be put to use by any wheelchair-bound student in need of it. Multiple tests have been conducted with ideal result as mentioned previously. With the additional user interface implemented on the device, this product must be attractive and useful to its users.

Among all the possible components, a box to contain the control system would be helpful to protect the circuit and the Arduino microcontroller.

6.4 Design Summary

Overall the main focus of the project was to transport a backpack from an initial position of behind the wheelchair to the side of the wheelchair which allows the backpack to be more accessible to the wheelchair user. When the backpack is stored in the initial position the wheelchairs mobility will be unhindered. The transportation of the backpack from initial to final position consist of smooth motion which lasts approximately 9 seconds while slowing down before ending in a final position that is most desirable to the user.

Appendix A Individual Design & Analysis

Appendix A.1 Nolan's Design

Appendix A.1.1 Linkage Design

The linkage design was a double crank four-bar linkage that moves in a circular motion from the initial to the final position. The design also has no change point which may cause problems in the motion of the links. The Lincages result and the actual link lengths are shown in *Figure A - 1* and *Table A - 1*.

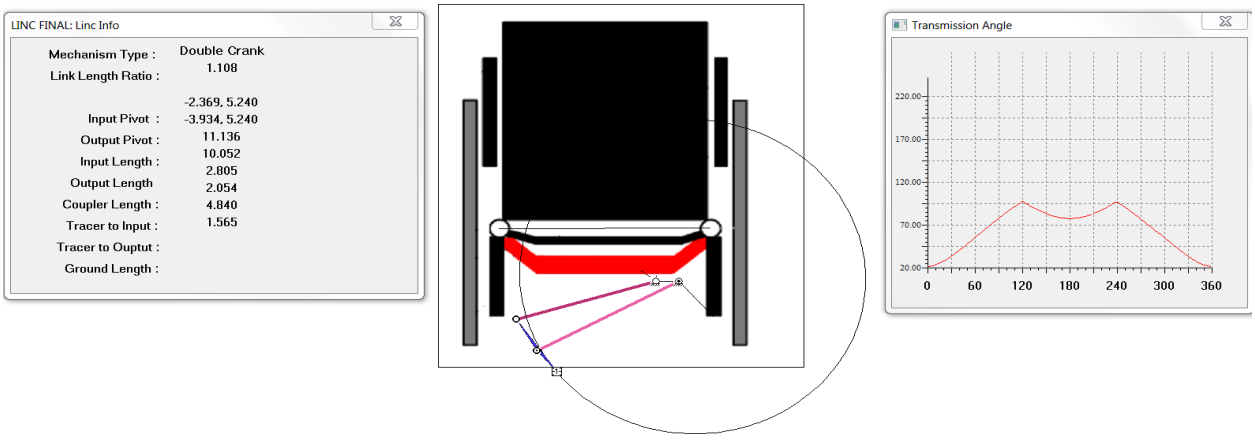


Figure A - 1: Nolan's Lincages design with link information and transmission angle

Link name	Input	Output	Coupler	Ground
Actual length (in.)	10.75	11.9	5.23	-

Table A - 1: Actual link lengths for Nolan's Design

Appendix A.1.2 CAD Model

The mechanism has a large ground link that attaches to the wheel chair, which allows room for the motor and the timing belt to drive the linkage. The input and output lengths were relatively short, this allows for a compact initial state which reduces the overall volume of the mechanism. The shorter links also decrease the torque that was imposed by the backpack. Different views of the final design in Solidworks can be found from *Figure A - 2* to *Figure A - 4*.

Figure A - 2: Nolan's Solidworks CAD model in its initial position

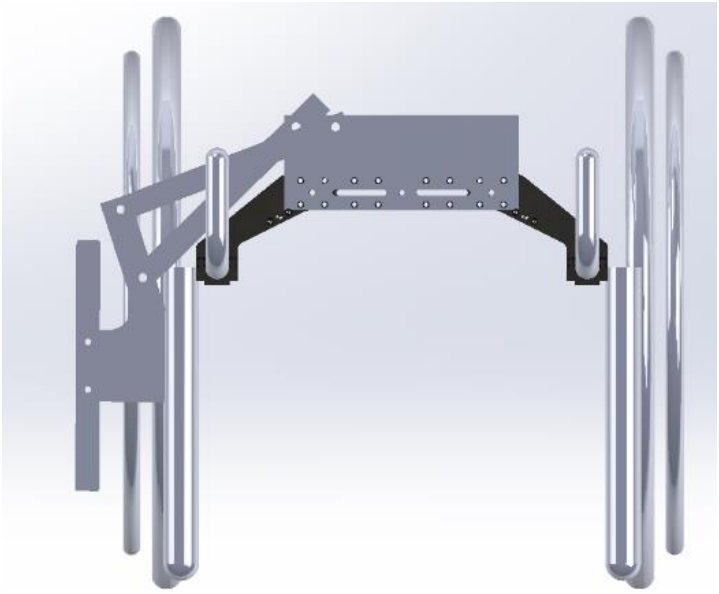


Figure A - 3: Nolan's Solidworks CAD model in its final position

Figure A - 4: Isometric view of Nolan's Solidworks CAD model

Appendix A.1.3 Loading Analysis

The ADAMS model analysis of Nolan's design showed that the force from the backpack was supported by the output link joints for roughly the first 5.5 seconds then the force was supported almost completely by the input link joints as it swings around. The model in ADAMS can be found in *Figure A - 5* and the result of simulation can be found in *Figure* A - *5*, *Figure A - 7* and summarized in *Table A - 2*.

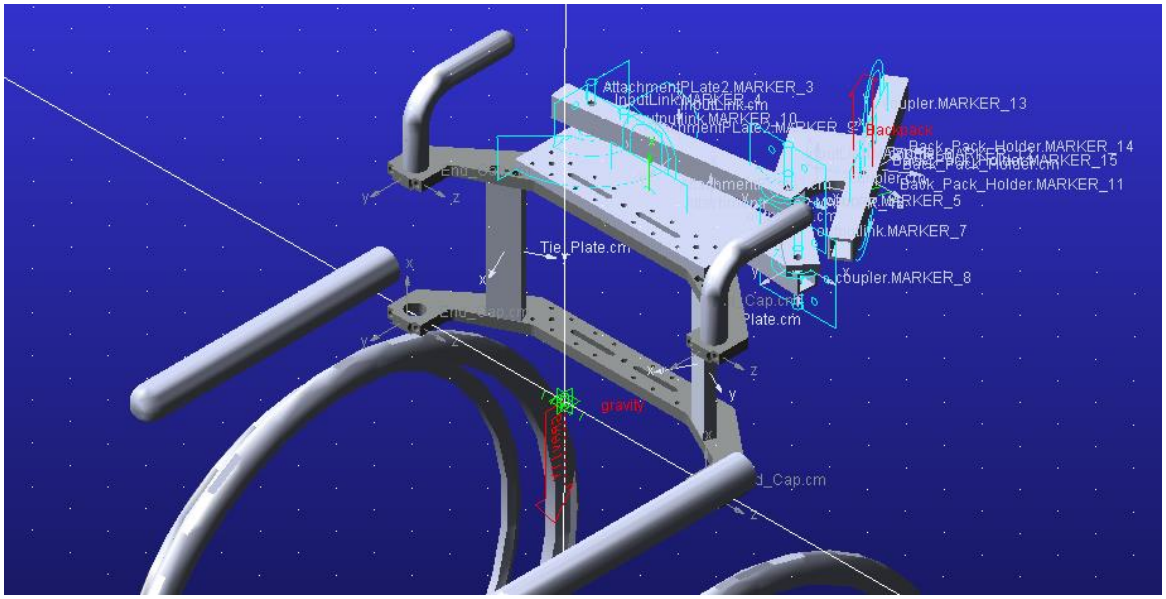


Figure A - 5: Nolan's ADAMS design in an isometric view

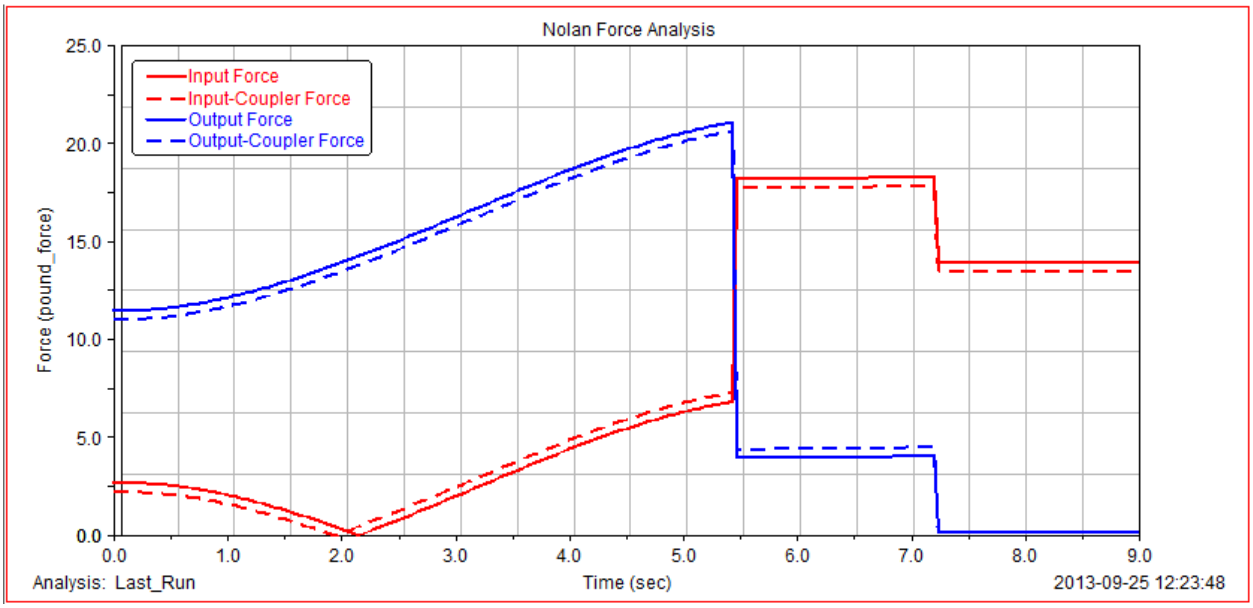


Figure A - 6: Force analysis of Nolan's design from ADAMS

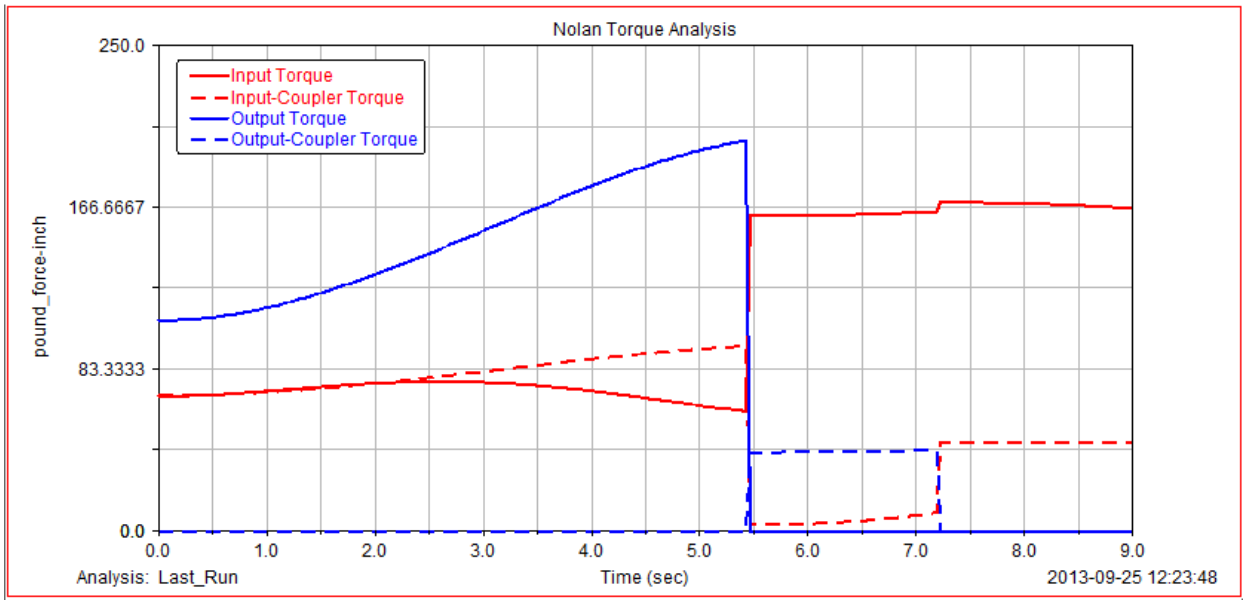


Figure A - 7: Torque analysis of Nolan's design from ADAMS

Joint	Force (lbf)	Torque (lbf*in.)
Input	18.4	169.6
Input-Coupler	17.9	95.4
Output	21.1	201.2
Output-Coupler	20.7	41.8

Table A - 2: Maximum forces and torques for Nolan's design from ADAMS

Appendix A.2 Zach's Design

Appendix A.2.1 Linkage Design

This four-bar linkage design was a Grashof double crank with no change point within the motion of the mechanism. The ground points for this linkage lie on the mounting plate to decrease the initial starting volume. The coupler goes through all the points on the Lincage model while dodging the wheel of the wheelchair. The Lincage results and the actual link lengths are shown in *Figure A - 8* and *Table A - 3*

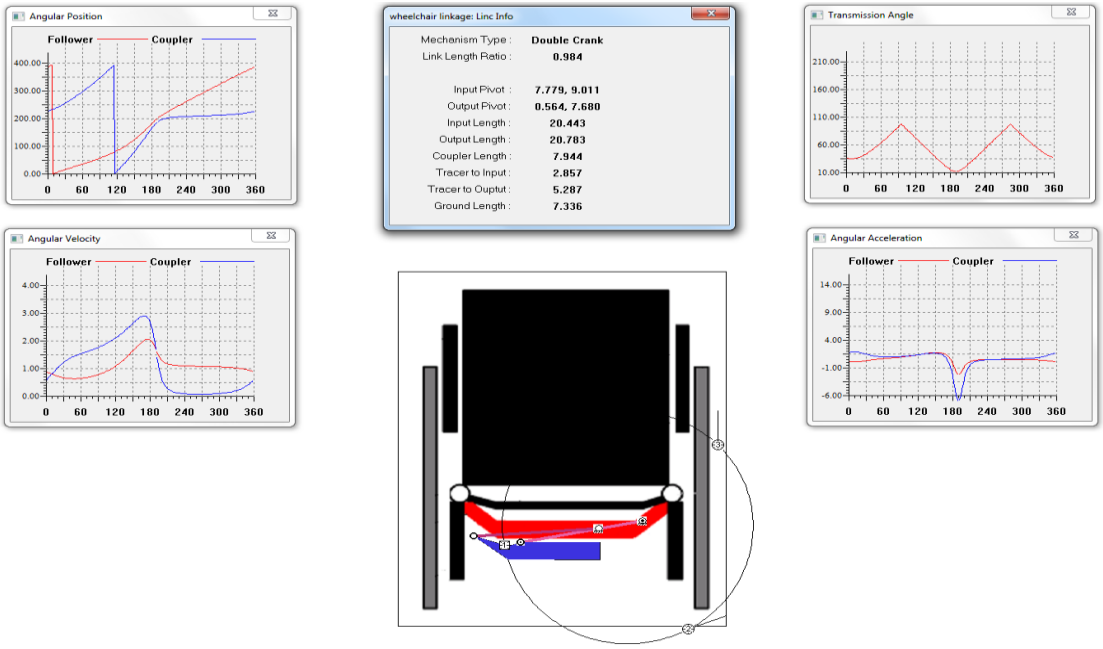


Figure A - 8: Zach's Lincages design with link information and transmission angle

Link name	Input	Output	Coupler	Ground
Actual length (in.)	8.88	9.03	3.45	3.19

Table A - 3: Actual link lengths for Zach's Design

Appendix A.2.2 CAD Model

The ground link of the mechanism was designed with extra room to allow room for the transmission to be designed and installed later. The design of the coupler minimized the initial starting volume while still providing full range of motion. At the end of the motion the backpack holder was parallel to the armrest. SolidWorks drawing of the mechanism that show the initial position and the final position of the backpack holder are shown in *Figure A - 9* to *Figure A - 11*.

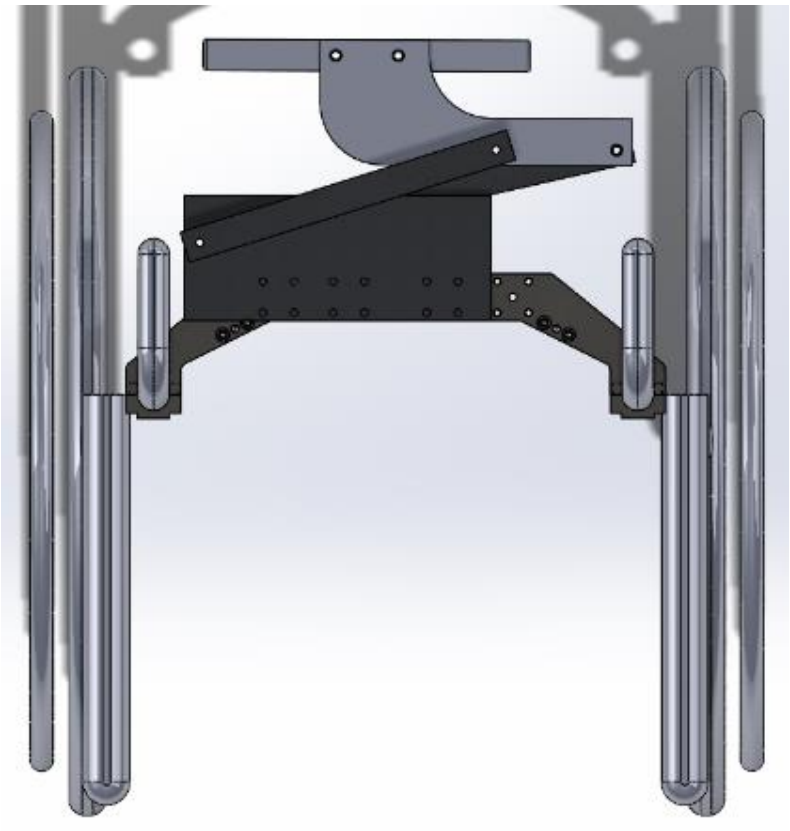


Figure A - 9: Zach's Solidworks CAD model in its initial position

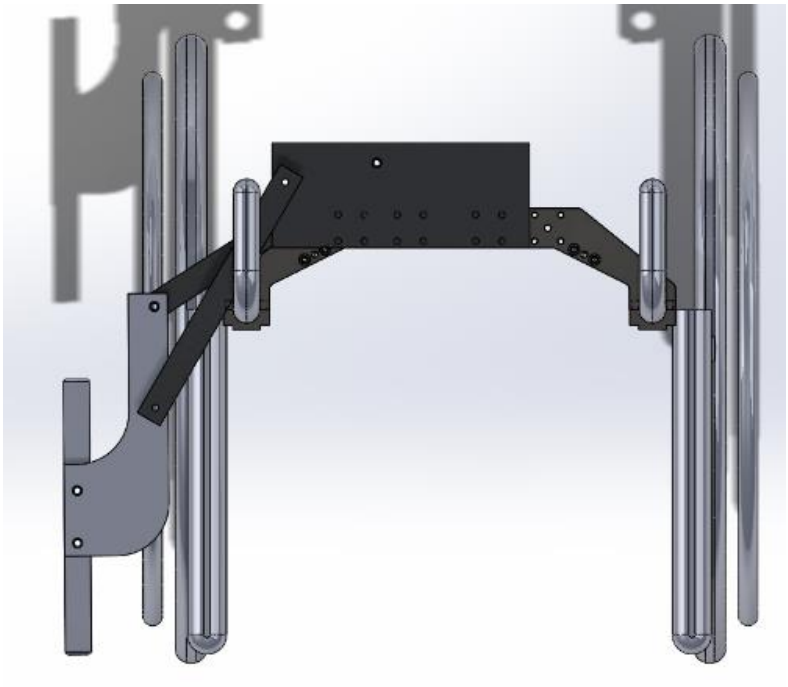


Figure A - 10: Zach's Solidworks CAD model in its final position

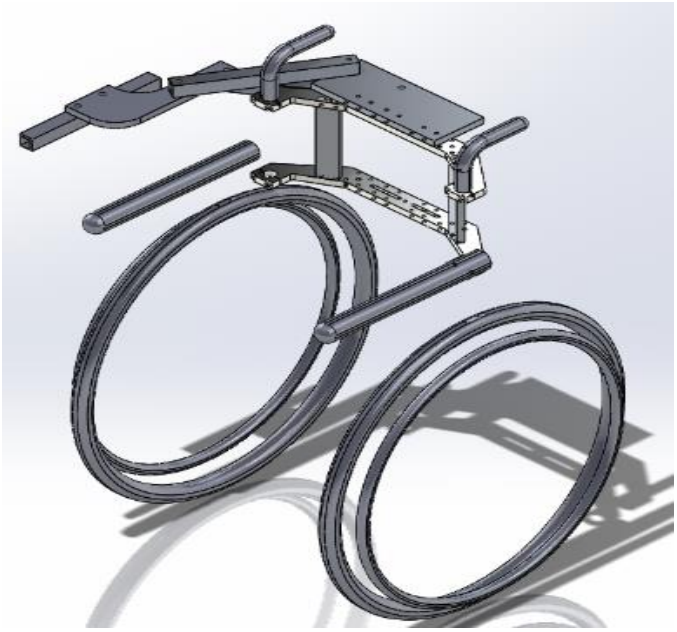


Figure A - 11: Isometric view of Zach's Solidworks CAD model

Appendix A.2.3 Loading Analysis

ADAMS model of Zach's four-bar linkage shows force and torque analysis for all of the points connecting the links. ADAMS provided for both force and torque analysis smooth curves which means that there was no sudden changes or discontinuity in the resulting forces. From such diagrams it can be expected that they motion of the mechanism would be smooth. Diagram for ADAMS model can be found at *Figure A - 12*. The force and torque analysis graphs from *Figure A - 13* and *Figure A - 14* with maximum and minimum value summarized in *Table A - 4*.

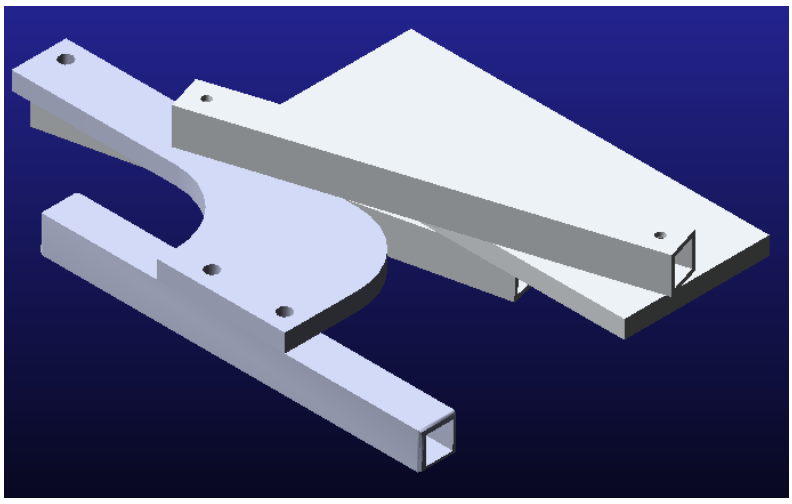


Figure A - 12: Zach's ADAMS design in an isometric view

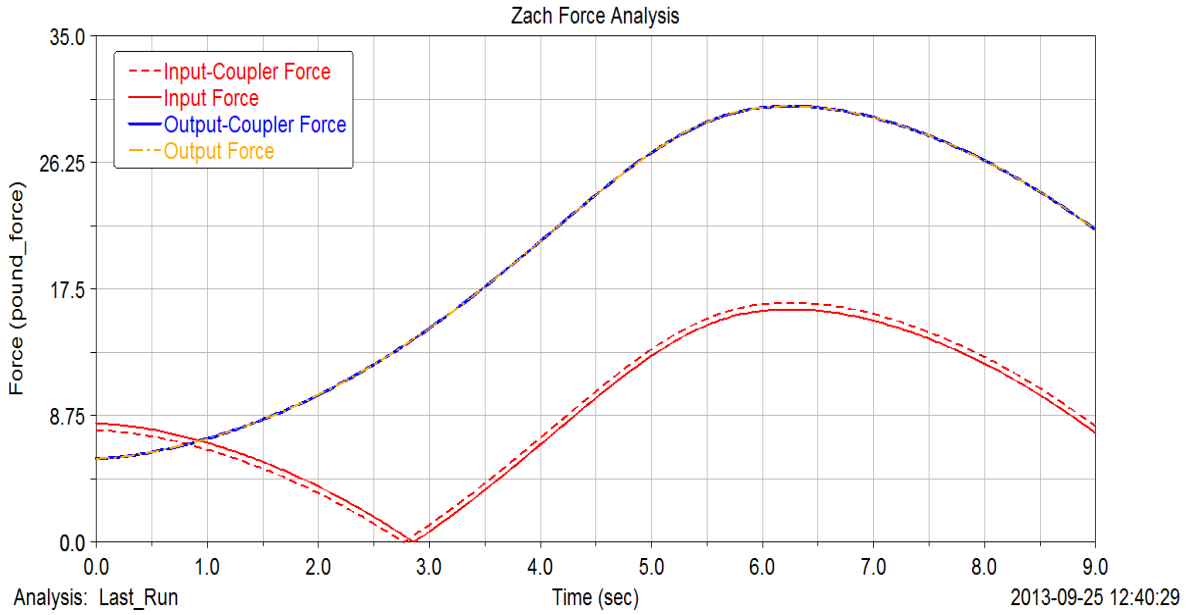


Figure A - 13: Force analysis of Zach's design from ADAMS

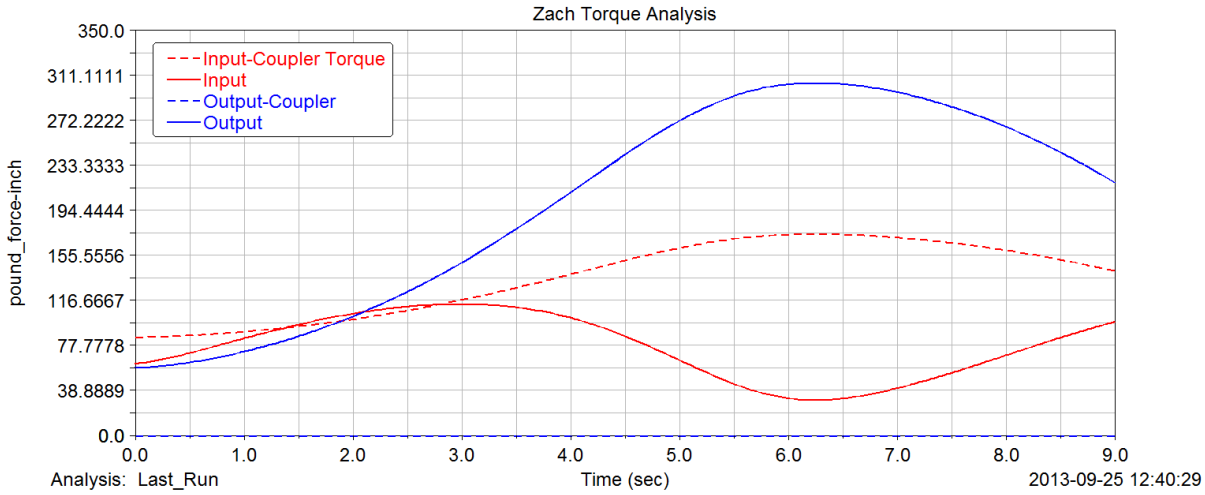


Figure A - 14: Torque analysis of Zach's design from ADAMS

Zach used a double crank system to design his four-bar linkages. Due to his suitable arrangement of the linkage system, the force exerted on the input link was greatly reduced, which was ideal for the purpose of friction reduction.

Joint	Force (lbf)	Torque (lbf*in.)
Input	16.1	113.8
Input-Coupler	16.6	174.2
Output	30.2	304.8
Output-Coupler	30.2	0

Table A - 4: Maximum forces and torques for Zach's design from ADAMS

Appendix A.3 Xingjian's Design

Appendix A.3.1 Linkage Design

Xingjian used a double crank system to design his four-bar linkages. There was no change point condition in his design as shown in *Figure A - 15*. From the Lincages simulation, it can be seen that the linkage would move in a circle with relatively large radius and end up in a position with good horizontal offset. The converted real length of the linkage is given in

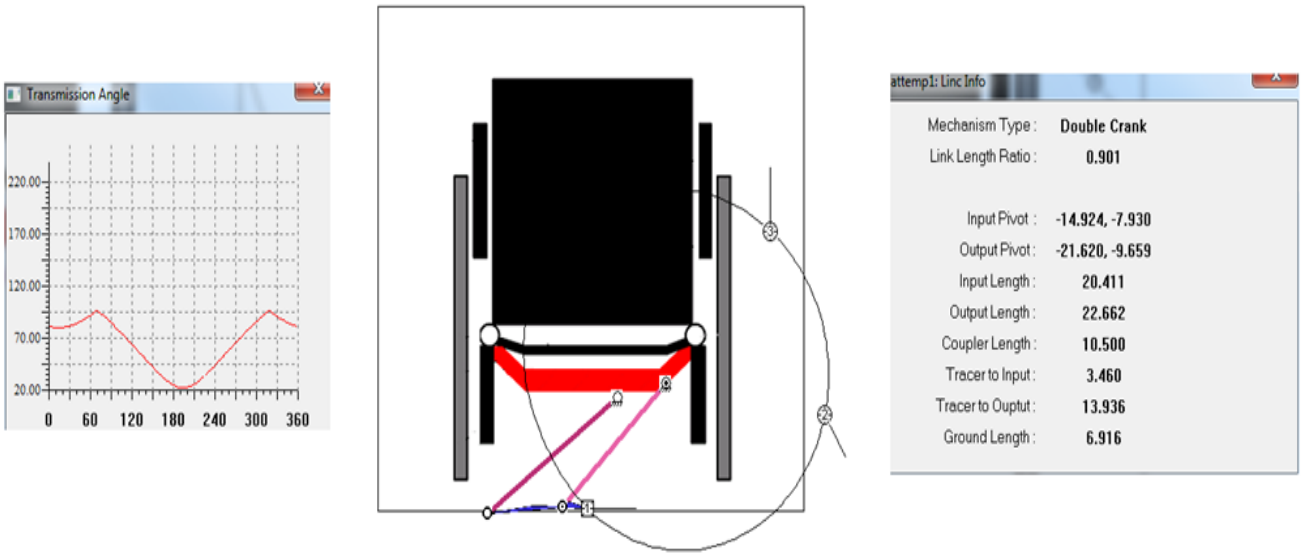


Figure A - 15: Xingjian's Lincages design with transmission angle and link information

Link name	Input	Output	Coupler	Ground
Actual length (in.)	10.00	11.31	5.59	3.5

Table A - 5: Actual link lengths for Xingjian's design

Appendix A.3.2 CAD Model

The coupler of Xingjian's design was big because he wants to avoid contact between backpack and wheels during motion. The shape and connections of his design were highly symmetric as shown below from *Figure A - 16* to *Figure A - 18*.

Figure A - 16: Xingjian's Solidworks CAD models at initial position

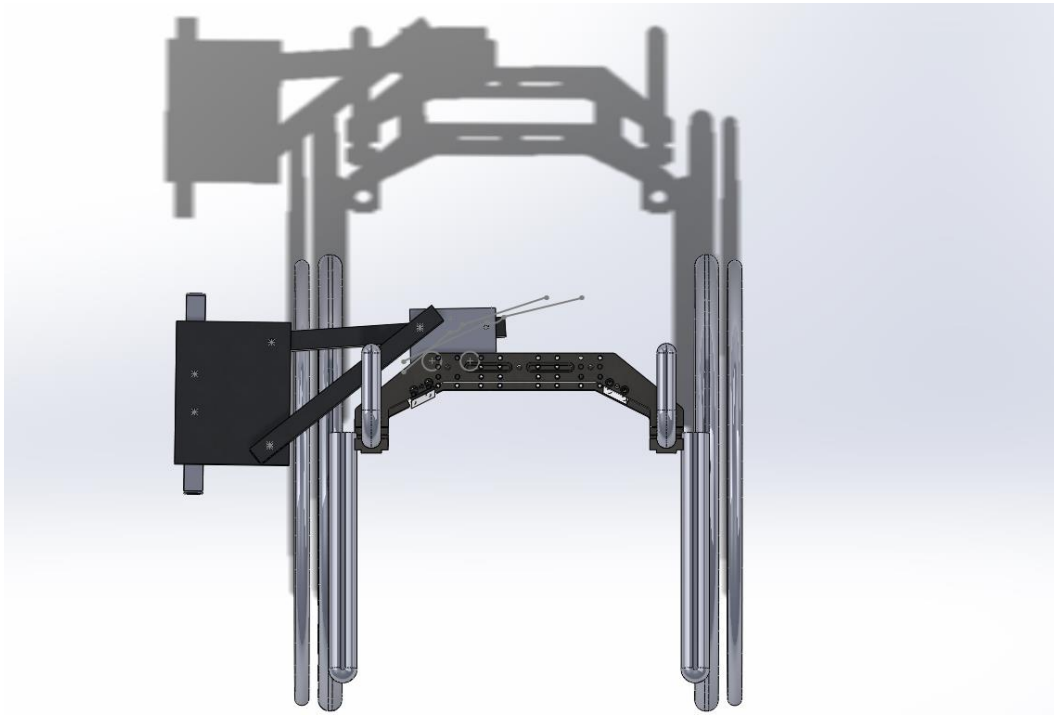


Figure A - 17: Xingjian's Solidworks CAD models at final position

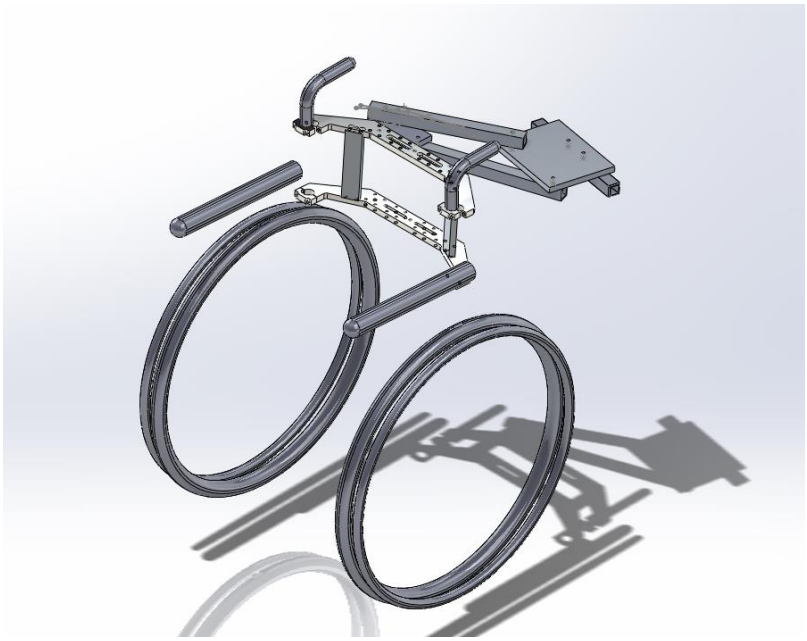


Figure A - 18: Xingjian's Solidworks CAD models at isometric view

Appendix A.3.3 Loading Analysis

Xingjian used a double crank system to design his four-bar linkages. There was no change point condition in his design and it was proved to be well functioning. The shape and connections of his design were highly symmetric, and it makes the force almost constant with time. The model in ADAMS can be found in *Figure A - 19* and the result of simulation can be found in *Figure A - 20*, *Figure A - 21* and summarized in *Table A - 6.*

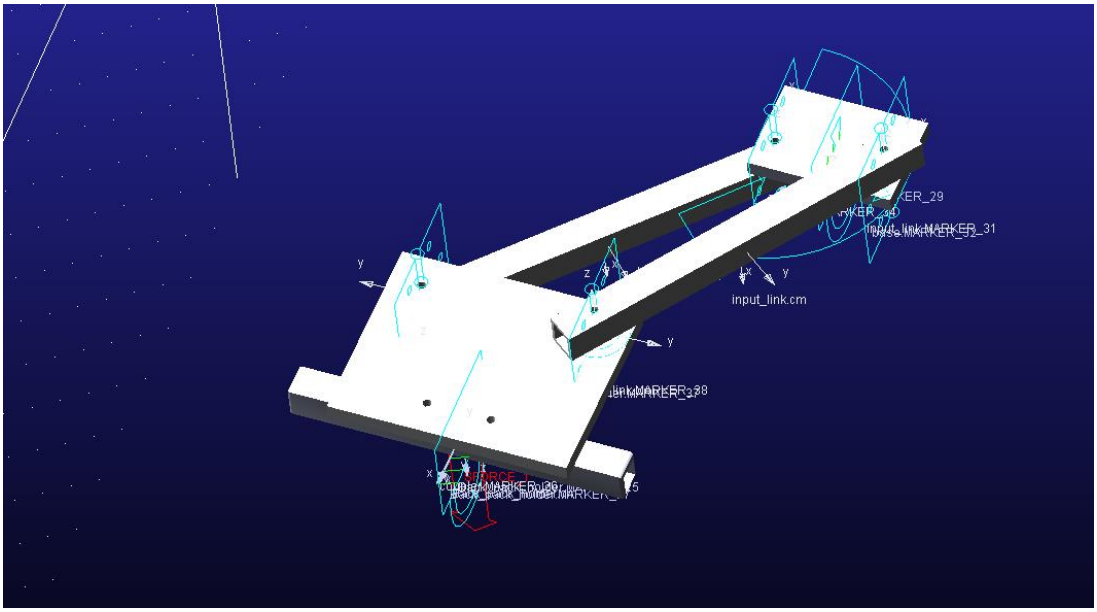


Figure A - 19: Xingjian's ADAMS design in an isometric view

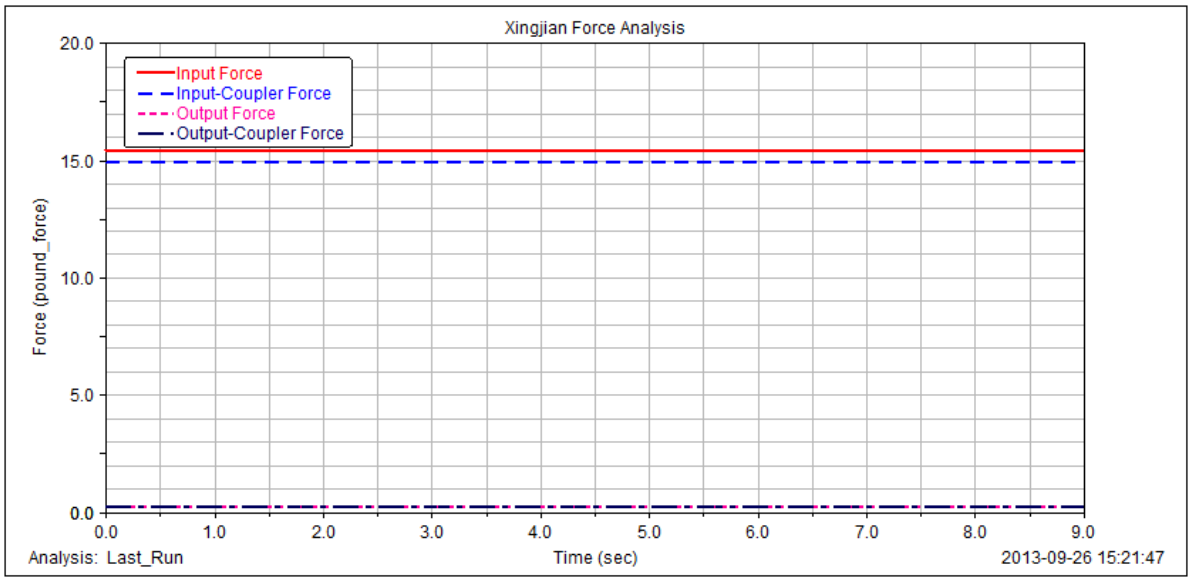


Figure A - 20: Force analysis of Xingjian's from ADAMS

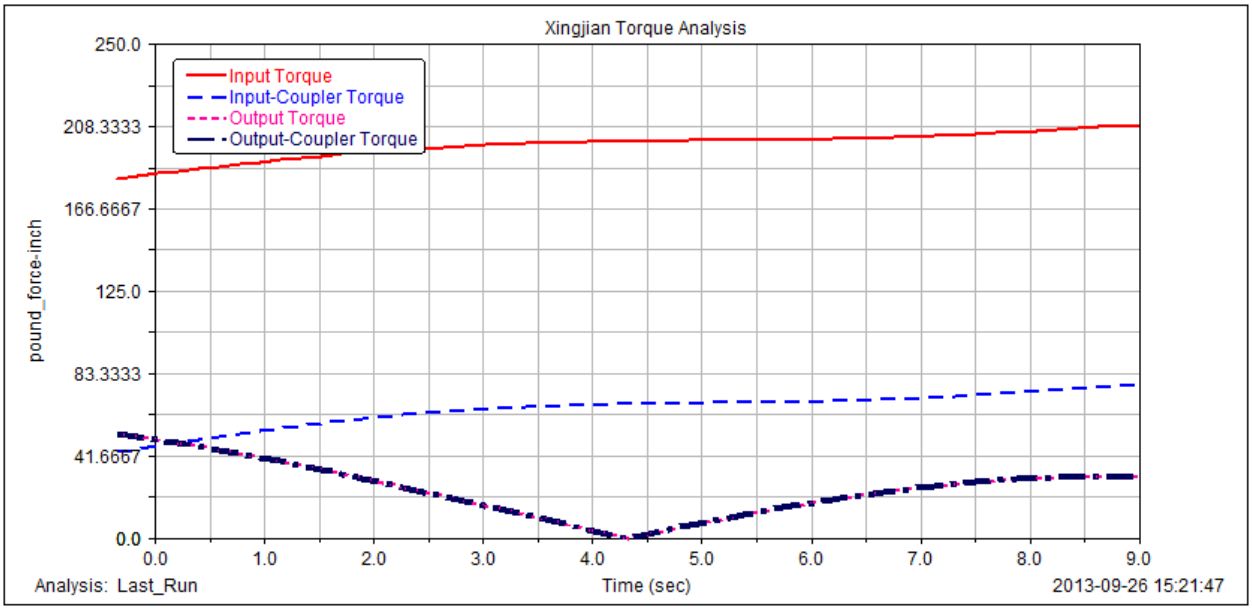


Figure A - 21: Torque analysis of Xingjian's from ADAMS

Joint	Force (lbf)	Torque (lbf*in.)
Input	15.5	209.5
Input-Coupler	15.0	78.4
Output	0.3	52.8
Output-Coupler	0.3	52.8

Table A - 6: Maximum forces and torques for Xingjian's design from ADAMS

Appendix A.4 Detailed Calculations for Deflections

Appendix A.4.1 Symbolic Calculation

a) Coupler

For the purpose of calculation, the following free body diagram was reproduced for different components of the structure with simplified geometry. Note that he bending moments at Joint I and Joint O have been neglected since the pin radius was small compared to the coupler length.

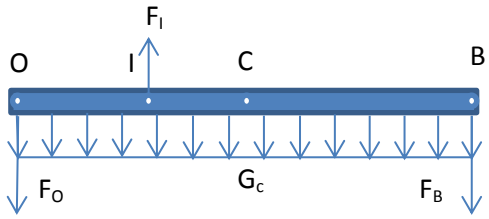


Figure A - 22: Coupler free body diagram
(simplified & not to scale)

From the free body diagram, the moment balance equation gives

$$\begin{cases} F_I = \dfrac{l_{OB}F_B + l_{OC}G_c}{l_{IO}} \\ F_O = \dfrac{l_{IB}F_B + l_{IC}G_c}{l_{IO}} \end{cases}$$

In this case, take the joint of the input link and the coupler as the fixed end and calculate maximum deflections with respect to fixed Point I. With the given geometry, assume the change of geometry was small and look into the appendix of textbooks to obtain the desired formula and use superposition to calculate the maximum deflection. For the coupler, the maximum deflections with respect to Point I were at Point B and Point O respectively. By denoting V_{ABL_C} as the magnitude of the maximum vertical deflection caused by load L_C at Point B with respect to fixed Point A, the equations go as the following

$$\begin{cases} V_{IBF_B} = \left|\dfrac{-F_B l_{BI}^3}{3EI_c}\right| \\ V_{IBG_c} = \left|\dfrac{-G_c l_{BI}^4}{8l_{OB}EI_c}\right| \end{cases} \qquad \begin{cases} V_{IOF_O} = \left|\dfrac{-F_O l_{OI}^3}{3EI_c}\right| \\ V_{IOG_c} = \left|\dfrac{-G_c l_{OI}^4}{8l_{OB}EI_c}\right| \end{cases} \qquad \begin{cases} V_{IBTotal} = V_{IBF_B} + V_{IBG_c} \\ V_{IOTotal} = V_{IOF_O} + V_{IOG_c} \end{cases}$$

b) Input Link

Similar to the calculation for the coupler, the free body diagram of the input link was reproduced below. Note that the bending moment at Joint I was neglected since the pin radius was small compared to the coupler length.

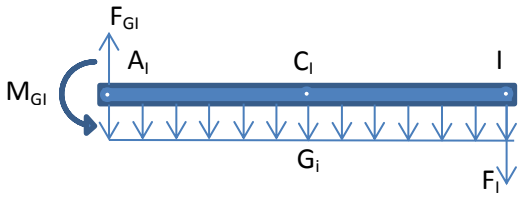


Figure A - 23: Input link free body diagram
(simplified & not to scale)

Considering the A_I point on the link as fixed with similar notation as discussed in the previous section, the magnitude of the downward deflection at Point I of the input link was given by

$$\begin{cases} V_{A_I IF_I} = \left|\frac{-F_I l_{IA_I}^3}{3EI_i}\right| \\ V_{A_I IG_i} = \left|\frac{-G_i l_{A_I C_I}^4}{8 l_{IA_I} EI_i}\right| \end{cases} \qquad V_{A_I I Total} = V_{A_I IF_I} + V_{A_I IG_i}$$

c) Output Link

Again the free body diagram of the output link was reproduced below. Note that neglected the bending moment at Joint I has been neglected since the pin radius was small compared to the coupler length.

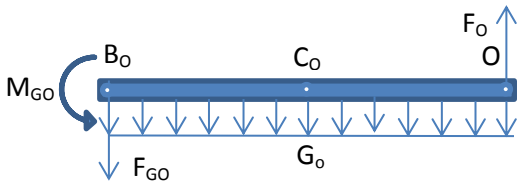


Figure A - 24: Output link free body
diagram (simplified & not to scale)

Considering the A_I point on the link as fixed with similar notation as discussed in the previous section, the magnitude of the upward deflection at Point O of the output link was given by

$$\begin{cases} V_{B_O OF_O} = \left|\frac{-F_O l_{OB_O}^3}{3EI_o}\right| \\ V_{A_O OG_o} = \left|\frac{-G_o l_{B_O C_O}^4}{8 l_{OB_O} EI_o}\right| \end{cases} \qquad V_{B_O O Total} = V_{B_O OF_I} - V_{B_O OG_o}$$

Appendix A.4.2 Dimensions

a) Direct Measurement

Based on the design measurement, the following value of parameters can be obtained.
From the design obtained in Lincages, the lengths of interest:

$$l_{IA_I} = l_{Input} = 11.41\ in. \quad l_{IO} = l_{Coupler} = 2.84\ in. \quad l_{OB_O} = l_{Output} = 11.83\ in.$$

Based on measurement taken in Solidworks the lengths:

$$l_{BO} = 9.16\ in. \quad l_{IB} = 6.37\ in.$$

Area moment of inertia for input, output link and backpack holder:

$$I_I = I_O = \frac{h_{out}^4}{12} - \frac{h_{in}^4}{12} = \frac{(1\ in.)^4 - (1\ in. - 2 \times 0.125\ in.)^4}{12} = \frac{175\ in.^4}{3072} = 5.6966 \times 10^{-2}\ in.^4$$

Area moment of inertia for coupler:

$$I_c = I_{xCoupler} = \frac{\mathrm{w} \cdot h^3}{12} - \frac{h_{in}^4}{12} = \frac{0.75\ in. \times (0.5\ in.)^3}{12} = \frac{1\ in.^4}{128} = 7.8125\ \times 10^{-3} in.^4$$

Young's modulus of aluminum:

$$E = E_{Al6061} = 10^7\ psi$$

Force of the backpack and the backpack holder:

$$F_B = G_{Backpack} + G_{Holder} = 12\ lb + 0.44\ lb = 12.44\ lb$$

b) Approximations

1) Coupler Length

Although the three points B, I and O were not on the same line, they were very close and can be approximated to be on the same line using the dimension of OB and IO. The percentage of distance error was calculated as the following to prove that the difference was negligible.

$$Difference\ \% = \left|\frac{(l_{BO} - l_{IB}) - l_{Coupler}}{l_{Coupler}}\right| \times 100\% = \left|\frac{(9.16 - 2.84) - 6.37}{6.37}\right| \times 100\% = 0.785\%$$

Based on the previous analysis:

$$l_{BO} = 9.16\ in. \quad l_{IB} = 6.32\ in. \quad l_{IO} = 2.84\ in.$$

2) Gravity Center

In the process of determining the mass center of the links, the horizontal displacement of the coupler has been neglected since its gravity center has a very small offset.

	Gravity (lbf)	Acting Position
Coupler	1.01	Center of $l_{BO} = 9.16\ in.$ (Approximated)
Input	0.52	Center of l_{Input} (Approximated)
Output	0.55	Center of l_{Output} (Accurate)

Table A - 7: Gravity and approximated gravity center position

3) Linkage length

The force acting point at each joint modeled in the calculation was at the calculated joint. Due to the fact that some material around the pivots for fixing the bearings was necessary, the actual length of the design was 1 in. longer than the calculated link length in total symmetrically. However, in calculation, the

uniformly distributed gravity of the additional material at both ends of the links does not contribute much to the deflection of the beam and was therefore not included in the calculation.

Appendix A.4.3 Numeric Result

By using the symbolic equations, dimensions and approximations listed above, plug in the equations into Mathematica to obtain the deflection of the beam as the following.

a) Input and output joint force:

$$\begin{cases} F_I = \dfrac{l_{OB}F_B + l_{OC}G_c}{l_{IO}} = 41.75\ lb \\ F_O = \dfrac{l_{IB}F_B + l_{IC}G_c}{l_{IO}} = 27.30\ lb \end{cases}$$

b) Input joint and ground force and torque:

$$\begin{cases} F_{GI} = F_I + G_i = 42.27\ lb \\ M_{GI} = l_{A_II}F_I + l_{A_IC_I}G_i = 479.35\ lb \cdot in. \end{cases}$$

c) Output joint and ground force and torque:

$$\begin{cases} F_{GO} = F_O - G_o = 28.85\ lb \\ M_{GO} = l_{B_OC_O}G_o - l_{B_OO}F_O = 338.09\ lb \cdot in. \end{cases}$$

d) Two end deflection with input link joint as fixed:

$$\begin{cases} V_{IBTotal} = V_{IBF_B} + V_{IBG_c} = 0.01340\ in. + 0.0002815\ in. = 1.368\ \times 10^{-2}\ in.\ (downward) \\ V_{IOTotal} = V_{IOF_O} + V_{IOG_c} = 0.002766\ in. + 1.148 \times 10^{-4}\ in. = 2.778 \times\ 10^{-3}\ in.\ (upward) \end{cases}$$

e) Input link deflection:

$$V_{A_IITotal} = V_{A_IIF_I} + V_{A_IIG_i} = 0.03629\ \ in. + 1.059 \times 10^{-5}\ in. = 3.630 \times 10^{-2}\ in.\ (downward)$$

f) Output link deflection:

$$V_{B_OOTotal} = V_{B_OOF_I} - V_{B_OOG_o} = 0.02742\ \ in. - 2.229 \times 10^{-5}\ in. = 2.744 \times 10^{-2}\ in.\ (upward)$$

From the above calculations, the deflections of the links were very small, with sum of all magnitude of deflection at the magnitude level of 10^{-2} in.

Appendix A.4.4 Mathematica code for calculation

```
In[1]:= ForceDeflection[force_, length_, modulus_, inertia_] :=
          Abs[(-force * length^3)/(3 * modulus * inertia)];
        LoadDeflection[force_, distributionlength_, length_, modulus_, inertia_] :=
          Abs[(-force * length^4)/(8 * distributionlength * modulus * inertia)];
        l_OB = 9.16;
        l_IO = 2.84;
        l_IB = l_OB - l_IO;
        l_OC = l_OB/2;
        l_IC = l_OC - l_IO;
        l_input = 11.41;
        l_output = 11.83;
        l_A_i C_i = l_input/2;
        l_B_o C_o = l_output/2;
        modulusE = 10^7;
        inertia_c = 7.8125 * 10^-3;
        inertia_link = 5.6966 * 10^-2;
        F_B = 12.44;
        G_c = 1.01;
        G_input = 0.52;
        G_output = 0.55;
        F_input = (l_OB * F_B + l_OC * G_c)/l_IO;
        F_output = (l_IB * F_B + l_IC * G_c)/l_IO;
        Print["F_input is ", F_input, " lb"];
        Print["F_output is ", F_output, " lb"];
        F_GI = F_input + G_input;
        M_GI = l_input * F_input + l_A_i C_i * G_input;
        Print["F_GI is ", F_GI, " lb"];
        Print["M_GI is ", M_GI, " lb*in."];
        F_GO = F_output + G_output;
        M_GO = l_output * F_output + l_B_o C_o * G_output;
        Print["F_GO is ", F_GO, " lb"];
        Print["M_GO is ", M_GO, " lb*in."];
        V_IBF_a = ForceDeflection[F_B, l_IB, modulusE, inertia_c];
        V_IBG_c = LoadDeflection[G_c, l_OB, l_IB, modulusE, inertia_c];
        V_IBTotal = V_IBF_a + V_IBG_c;
        Print["V_IBF_a is ", V_IBF_a, " in."];
        Print["V_IBG_c is ", V_IBG_c, " in."];
        Print["V_IBTotal is ", V_IBTotal, " in."];
        V_IOF_o = ForceDeflection[F_output, l_IO, modulusE, inertia_c];
        V_IOG_c = LoadDeflection[G_c, l_OB, l_IO, modulusE, inertia_c];
        V_IOTotal = V_IOF_o + V_IOG_c;
        Print["V_IOF_o is ", V_IOF_o, " in."];
```

```
Print["V_IOG_2 is ", V_IOG_2, " in."];
Print["V_IOTotal is ", V_IOTotal, " in."];
V_A_2 IF_2 = ForceDeflection[F_input, l_input, modulusE, inertia_link];
V_A_2 IG_2 = LoadDeflection[G_input, l_input, l_A_2 C_2, modulusE, inertia_link];
V_A_2 ITotal = V_A_2 IF_2 + V_A_2 IG_2;
Print["V_A_2 IF_2 is ", V_A_2 IF_2, " in."];
Print["V_A_2 IG_2 is ", V_A_2 IG_2, " in."];
Print["V_A_2 ITotal is ", V_A_2 ITotal, " in."];
V_B_0 OF_0 = ForceDeflection[F_output, l_output, modulusE, inertia_link];
V_A_0 OG_0 = LoadDeflection[G_C, l_output, l_B_0 C_0, modulusE, inertia_link];
V_B_0 OTotal = V_B_0 OF_0 + V_A_0 OG_0;
Print["V_B_0 OF_0 is ", V_B_0 OF_0, " in."];
Print["V_A_0 OG_0 is ", V_A_0 OG_0, " in."];
Print["V_B_0 OTotal is ", V_B_0 OTotal, " in."];
```

F_{input} is 41.7522 lb

F_{output} is 28.3022 lb

F_{BI} is 42.2722 lb

M_{BI} is 479.359 lb-in.

F_{BO} is 28.8522 lb

M_{BO} is 338.068 lb-in.

V_{IIF_2} is 0.0133986 in.

V_{IIG_2} is 0.000281458 in.

$V_{IITotal}$ is 0.0136801 in.

V_{IOF_0} is 0.00276607 in.

V_{IOG_2} is 0.0000114768 in.

$V_{IOTotal}$ is 0.00277755 in.

$V_{A_2 IF_2}$ is 0.036291 in.

$V_{A_2 IG_2}$ is 0.0000105934 in.

$V_{A_2 ITotal}$ is 0.0363016 in.

$V_{B_0 OF_0}$ is 0.0274181 in.

$V_{A_0 OG_0}$ is 0.0000229324 in.

$V_{B_0 OTotal}$ is 0.027441 in.

Appendix B Motion Parts Manufacturing & Assembly

Appendix B.1 Engineering Drawings

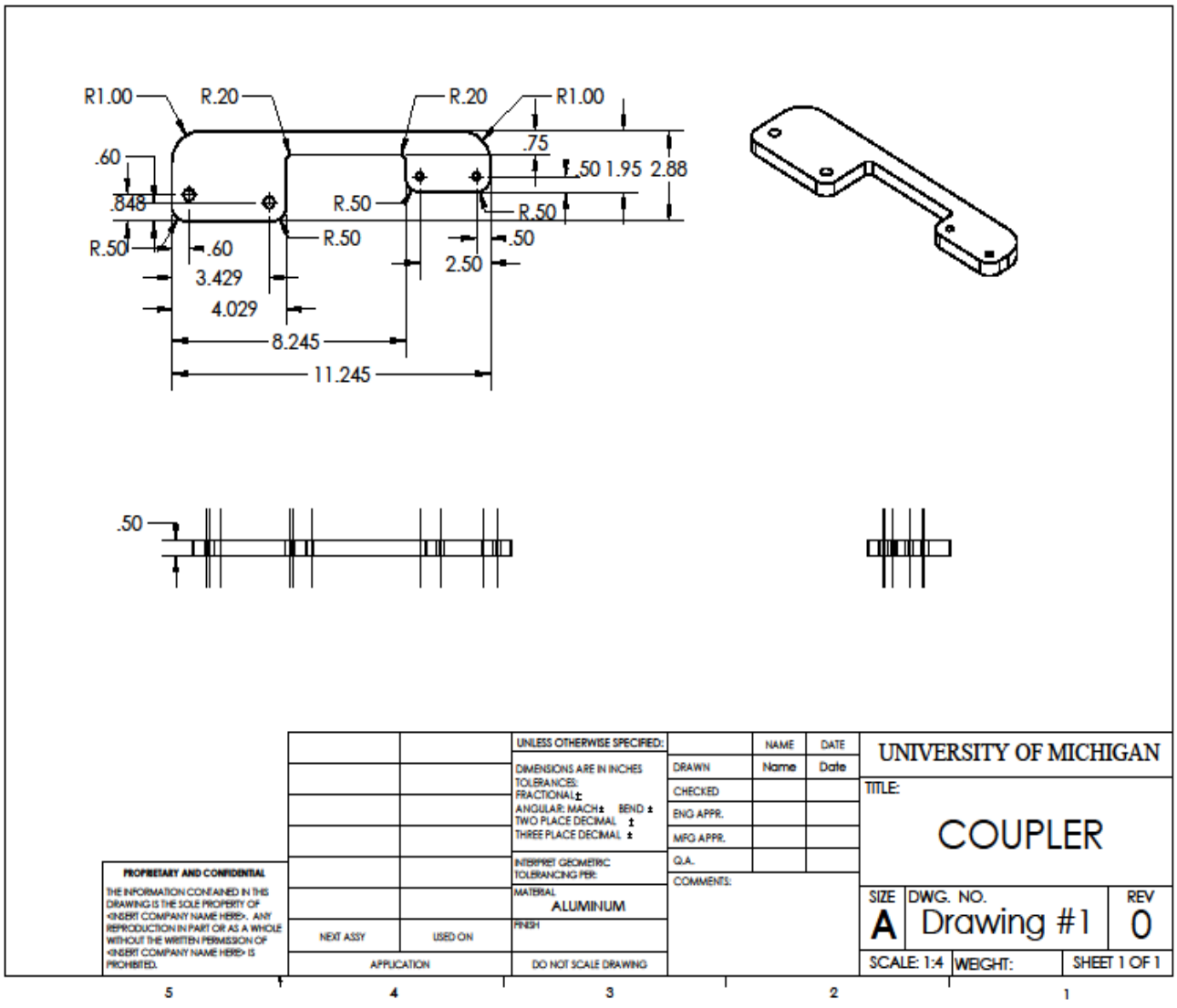


Figure B - 1: Engineering drawing for coupler

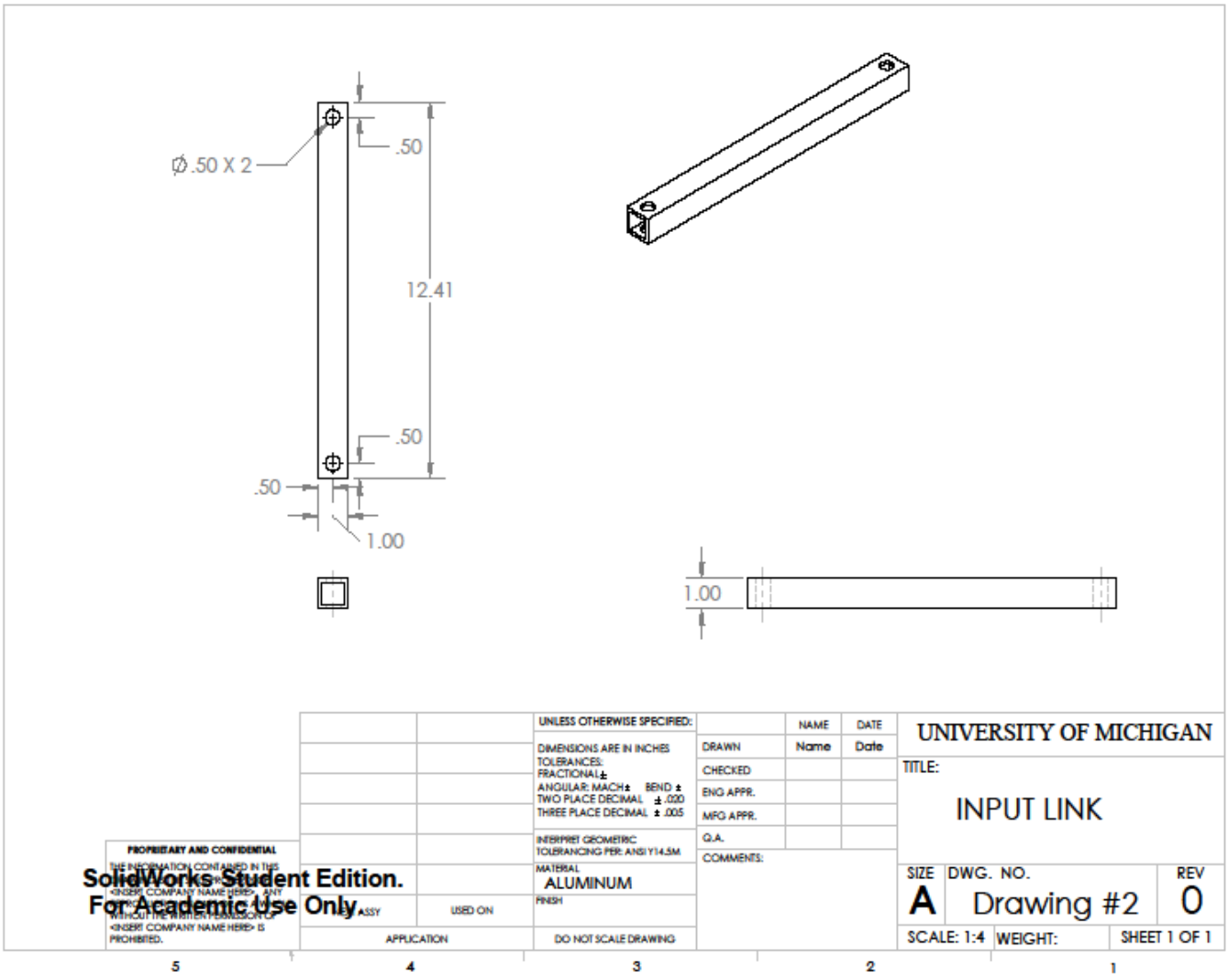


Figure B - 2: Engineering drawing for input link

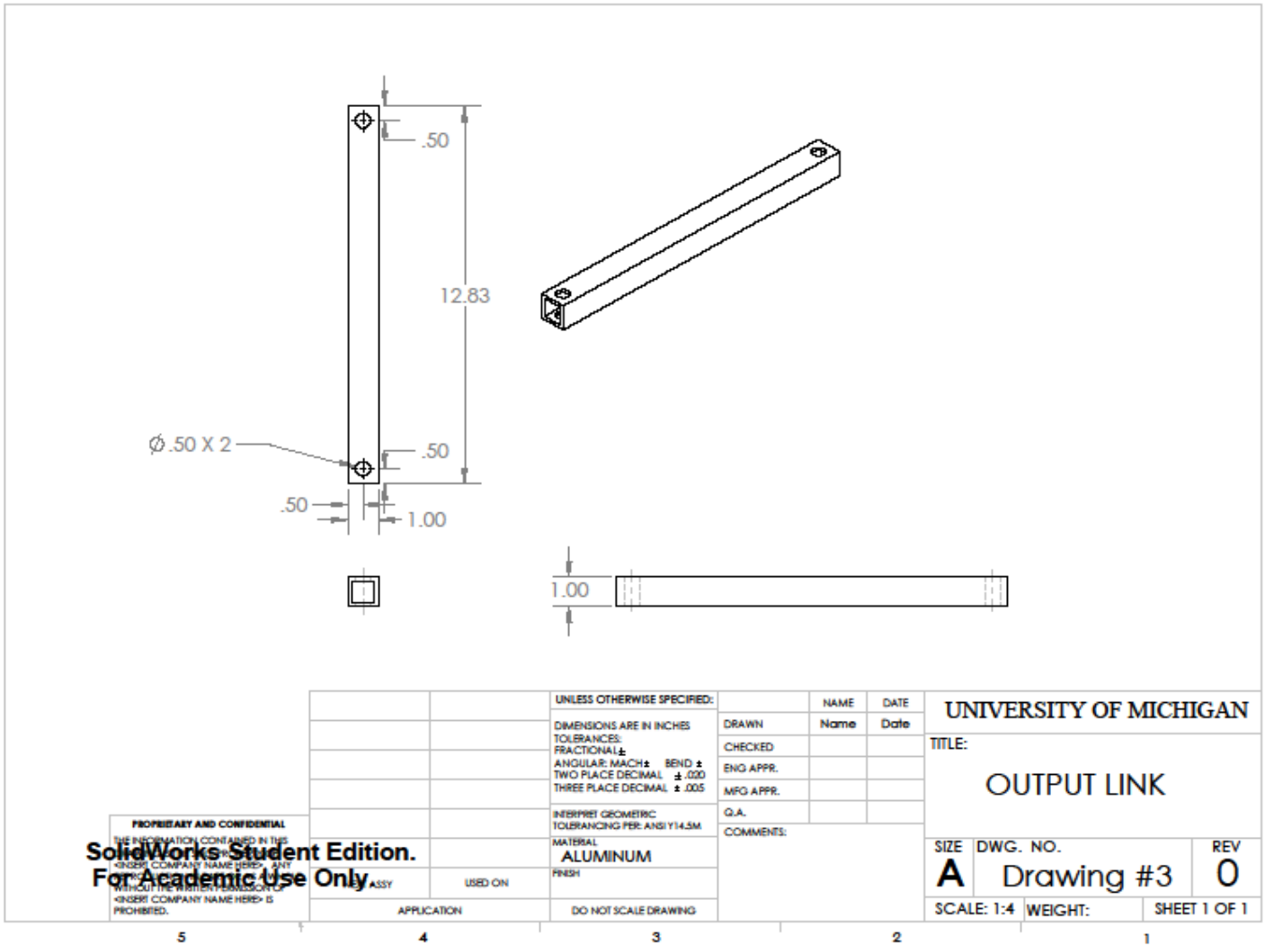


Figure B - 3: Engineering drawing for output link

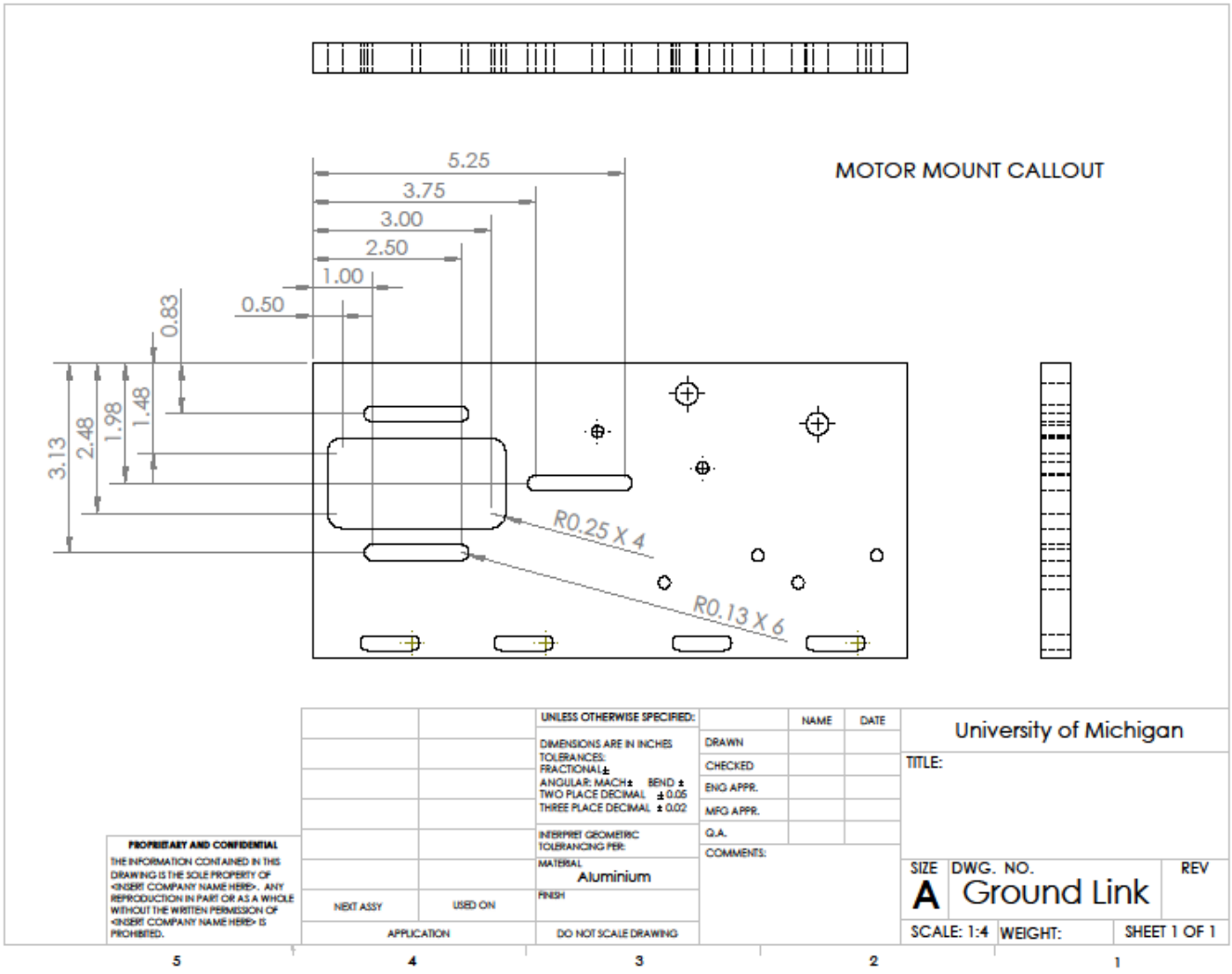


Figure B - 4: Engineering drawing for ground link

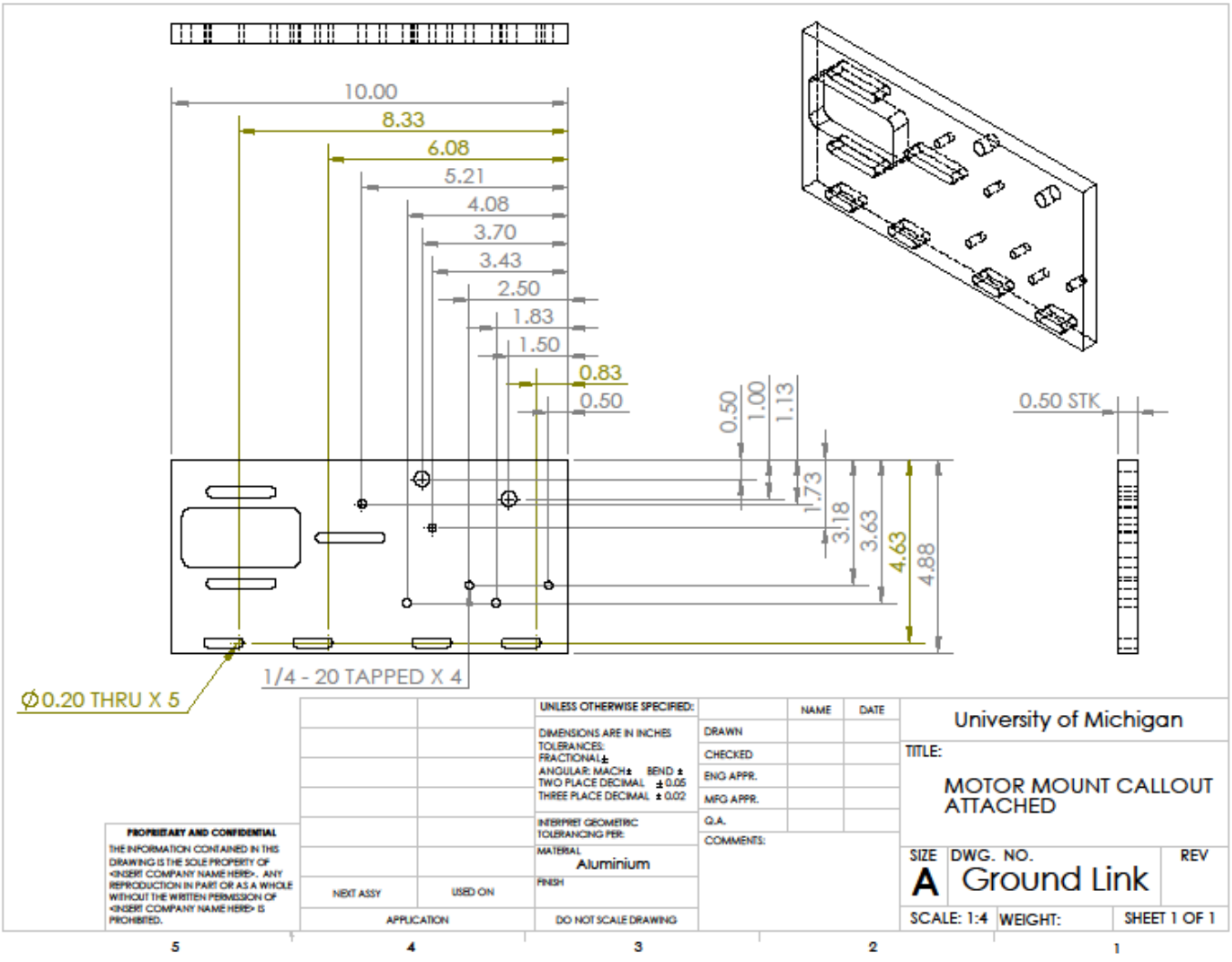


Figure B - 5: Engineering drawing for ground link

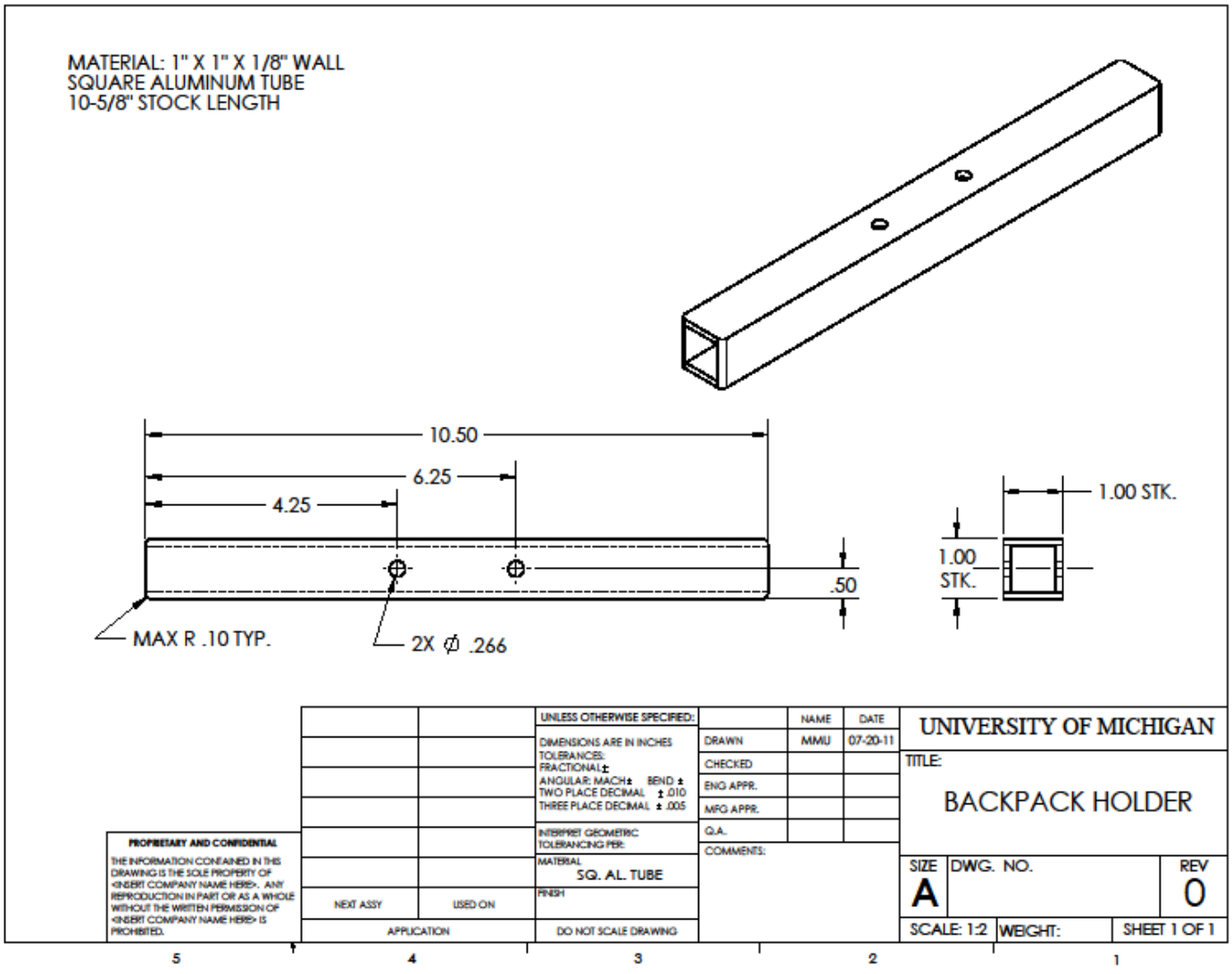


Figure B - 6: Engineering drawing for backpack holder

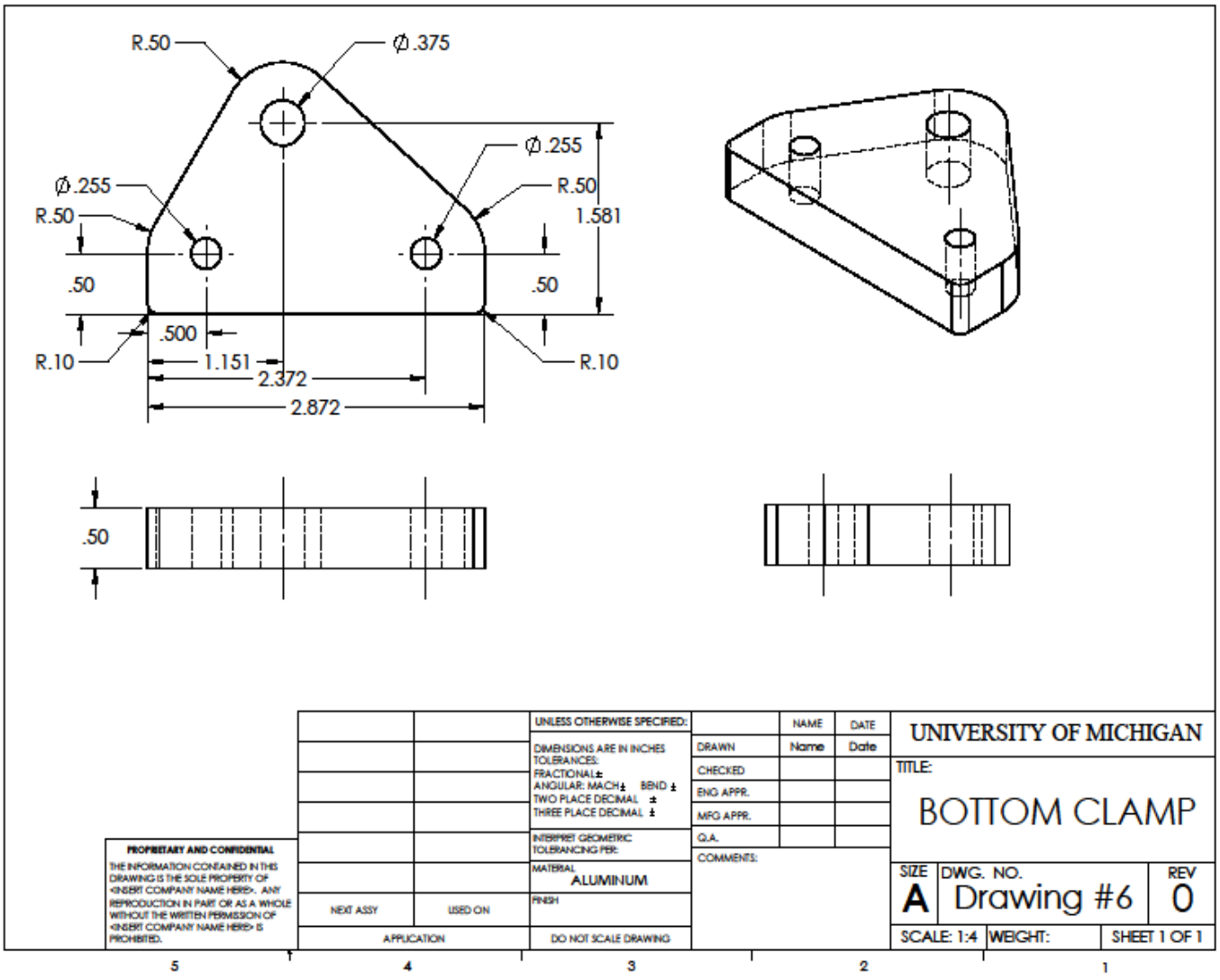


Figure B - 7: Engineering drawing for bottom clamp

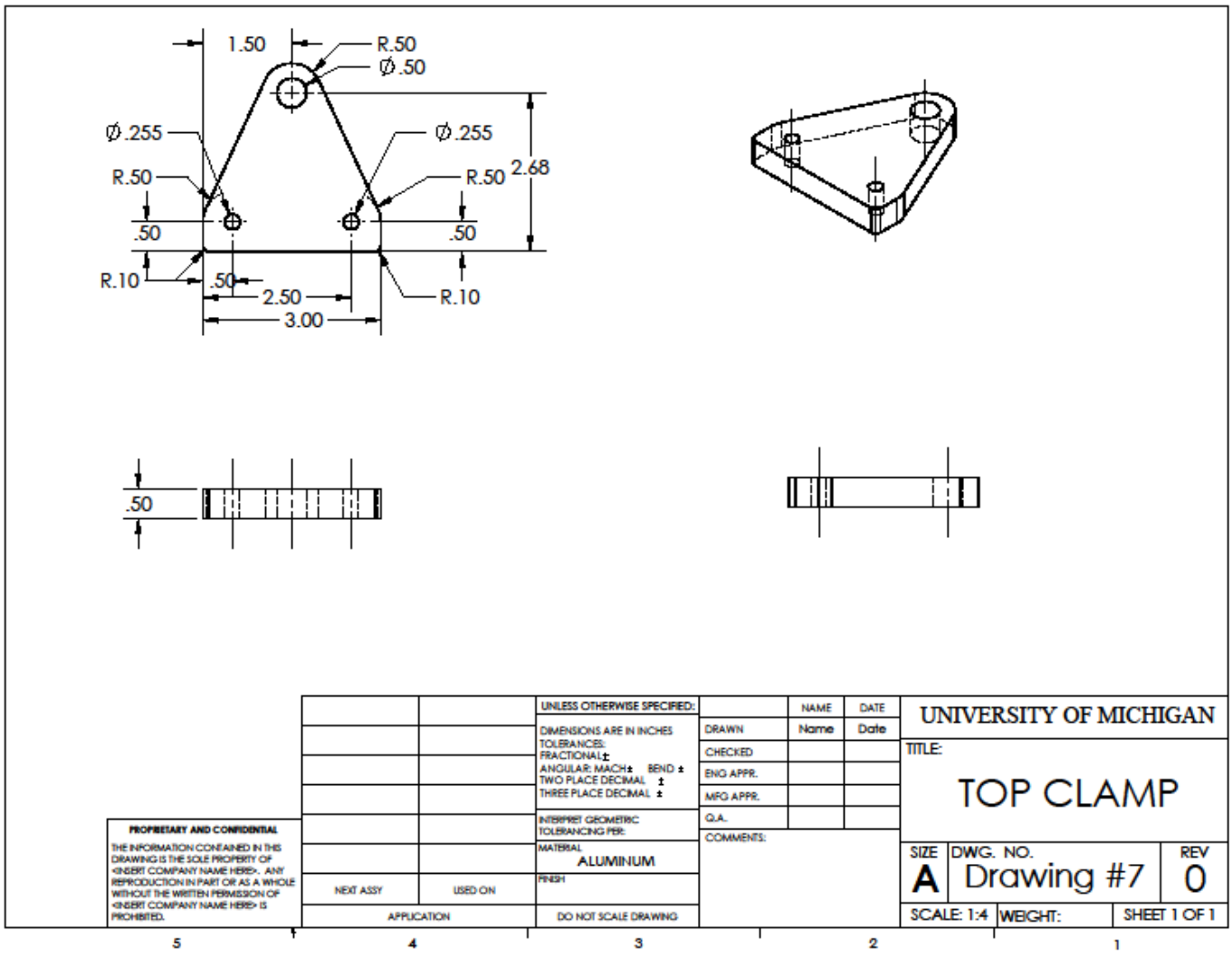


Figure B - 8: Engineering drawing for top clamp

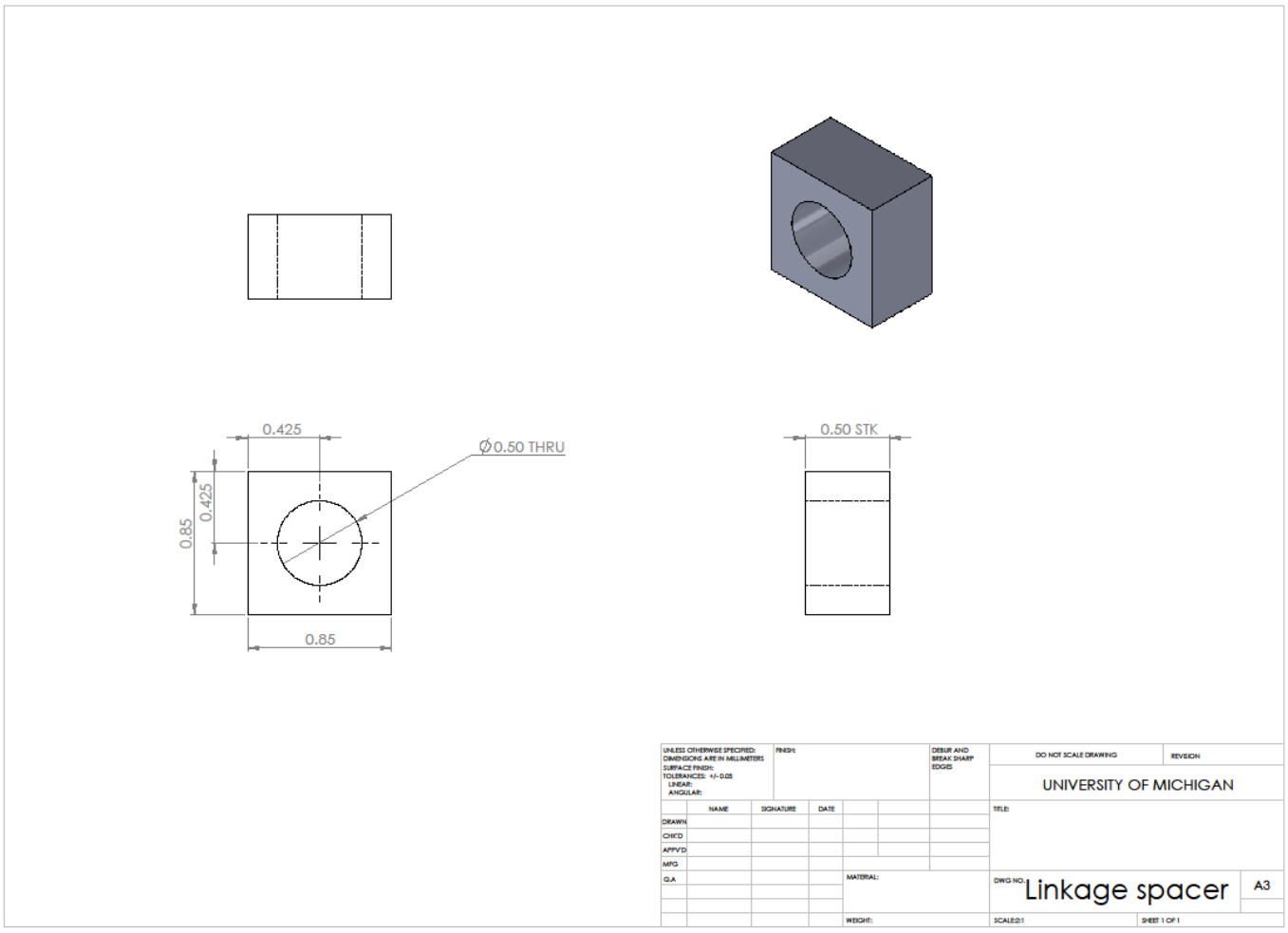


Figure B - 9: Engineering drawing for linkage spacer

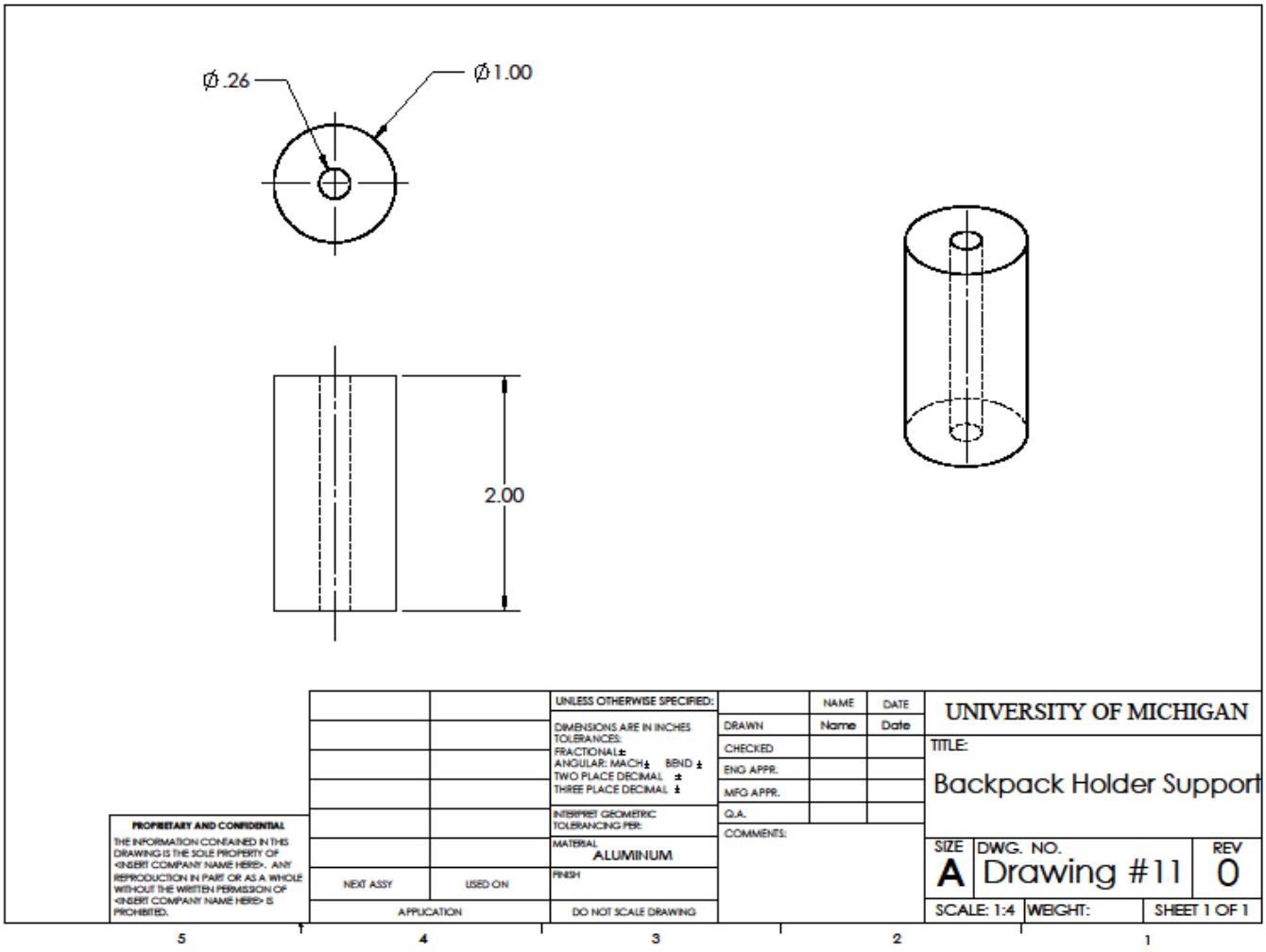


Figure B - 10: Engineering drawing for backpack holder support

Appendix B.2 Manufacturing Plans

Revision Date:
10/8/2013

Part Number: ME350-001

Part Name: Coupler

Team Name: Fouro

Raw Material Stock:

6061-T6 Aluminum Plate, 1/2" x 12" x 12"

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Water jet the plate	Water jet cutting machine			
2	Install drill chuck.	Mill	vise	drill chuck	
3	Find datum lines for X and Y of the two bigger holes	Mill	vise	edge finder, drill chuck	900
4	Centerdrill and drill the holes.	Mill	vise	Center drill, V drill bit, drill chuck	3500
5	Find datum lines for X and Y of the two smaller holes	Mill	vise	edge finder, drill chuck	900
6	Centerdrill and drill the holes.	Mill	vise	Center drill, H drill bit, drill chuck	3600

Part Number: ME350-002 Revision Date: 9/29/2013

Part Name: Input Link

Team Name: Fouro

Raw Material Stock:

6061-T6 Square Aluminum Tubing (1" x 1" x 6' long, 1/8" wall)

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Bandsaw the input link into a length of 12.50 inches			band saw	
2	Hold part in vise.	Mill	vise		
3	Mill one end of part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
4	Remove part from vise. Break all edges by hand.			file	
5	Place part in vise to machine other end of part. Mill the part to 12.41 inches length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840
6	Remove part from vise. Break all edges by hand.			file	
7	Remove cutter and collet. Install drill chuck. Return part to vise.	Mill	vise	drill chuck	
8	Find datum lines for X and Y of the frist hole.	Mill	vise	edge finder, drill chuck	900
9	Centerdrill and pre drill the hole under 1/2"	Mill	vise	Center drill, 31/64 in. drill bit, drill chuck	2865

10	Ream the hole to 1/2"	Mill	vise	drill chuck, 1/2 in. reamer	100
11	Find datum lines for X and Y of the second hole.	Mill	vise	edge finder, drill chuck	900
12	Centerdrill and pre drill the hole under 1/2"	Mill	vise	Center drill, 31/64 in. drill bit, drill chuck	2865
13	Ream the hole to 1/2"	Mill	vise	drill chuck, 1/2 in. reamer	100

Part Number: ME350-003 Revision Date: 9/29/2013

Part Name: output Link

Team Name: Fouro

Raw Material Stock:

6061-T6 Square Aluminum Tubing (1" x 1" x 6' long, 1/8" wall)

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Bandsaw the tubing into a length of 13 inches length			band saw	
2	Hold part in vise.	Mill	vise		
3	Mill one end of part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
4	Remove part from vise. Break all edges by hand.			file	
5	Place part in vise to machine other end of part. Mill the part to 12.83 inches length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840
6	Remove part from vise. Break all edges by hand.			file	
7	Remove cutter and collet. Install drill chuck. Return part to vise.	Mill	vise	drill chuck	
8	Find datum lines for X and Y of the first hole.	Mill	vise	edge finder, drill chuck	900
9	Centerdrill and drill the hole.	Mill	vise	Center drill, 1/2 in drill bit, drill chuck	2865

10	Find datum lines for X and Y of the second hole.	Mill	vise	edge finder, drill chuck	900
11	Centerdrill and drill the hole.	Mill	vise	Center drill, 1/2 in drill bit, drill chuck	2865

Part Number: ME350-004 Revision Date: 15/10/2013

Part Name: ground base

Team Name: Fouro

Raw Material Stock: 6061-T6 Aluminum Plate, 1/2" x 12" x 12"

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Bandsaw the plate into 5.1" x 12"			Band saw	
2	Hold part in vise.	Mill	vise		
3	Mill the plate to 5.00 inches length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840
4	Remove part from vise. Break all edges by hand.			file	
5	Mark the part to remove the space to produce the 0.125" wide slot of the plate			Height Gauge, Surface Plate	
6	Place part in vise and use endmill to remove material. Turn off the spindle, and measure part with calipers.	Mill	vise	1/8 inch 2-flute endmill, collet	3600
7	Mark the part to remove the space to produce the two 0.25" wide slots of the plate	Mill	vise	Height Gauge, Surface Plate	
8	Place part in vise and use endmill to remove material. Turn off the spindle, and measure part with calipers.	Mill	vise	1/4 inch 2-flute endmill, collet	3500
9	Remove cutter and collet. Install drill chuck. Return part to vise.	Mill	vise	drill chuck	

10	Find datum lines for X and Y of holes with diameter 0.375 inches.	Mill	vise	edge finder, drill chuck	900
11	Center drill and predrill under 3/8"	Mill	vise	Center drill, U drill bit, drill chuck	3400
12	Ream hole to 3/8"	Mill	vise	3/8 inch Reamer	900
13	Find datum lines for X and Y of holes with diameter 0.255 inches.	Mill	vise	edge finder, drill chuck	900
14	Centerdrill and drill the hole.	Mill	vise	Center drill, F drill bit, drill chuck	3500
15	Find datum lines for X and Y of holes with diameter 0.201 inches.	Mill	vise	edge finder, drill chuck	900
16	Centerdrill and drill the hole.	Mill	vise	Center drill, #7 drill bit, drill chuck	3600

Part Number: ME350-005 Revision Date: 9/29/2013

Part Name: backpack holder

Team Name: Fouro

Raw Material Stock:

6061-T6 Square Aluminum Tubing (1" x 1" x 6' long, 1/8" wall)

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Bandsaw the the tubing into a length of 10.50 inches			band saw	
2	Hold part in vise.	Mill	vise		
3	Mill one end of part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
4	Remove part from vise. Break all edges by hand.			file	
5	Place part in vise to machine other end of part. Mill the part to 10.30 inches length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840
6	Remove part from vise. Break all edges by hand.			file	
7	Remove cutter and collet. Install drill chuck. Return part to vise.	Mill	vise	drill chuck	
8	Find datum lines for X and Y of the frist hole.	Mill	vise	edge finder, drill chuck	900
9	Centerdrill and drill the hole.	Mill	vise	Center drill, I drill bit, drill chuck	3600

10	Find datum lines for X and Y of the second hole.	Mill	vise	edge finder, drill chuck	900
11	Centerdrill and drill the hole.	Mill	vise	Center drill, I drill bit, drill chuck	3600

Part Name: bottom clamp-006 Revision Date: 8/10/2013

Team Name: Fouro

Raw Material Stock:

6061-T6 Aluminum Plate, 1/2" x 12" x 12"

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Water jet the plate	Water jet cutting machine			
2	Install drill chuck.	Mill	vise	drill chuck	
3	Find datum lines for X and Y of the bigger hole	Mill	vise	edge finder, drill chuck	900
4	Centerdrill and drill the hole.	Mill	vise	Center drill, V drill bit, drill chuck	3500
5	Find datum lines for X and Y of the two smaller holes	Mill	vise	edge finder, drill chuck	900
6	Centerdrill and drill the holes.	Mill	vise	Center drill, H drill bit, drill chuck	3600

Revision Date:
10/8/2013

Part Number: ME350-007

Part Name: top clamp

Team Name: Fouro

Raw Material Stock:

6061-T6 Aluminum Plate, 1/2" x 12" x 12"

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Water jet the plate	Water jet cutting machine			
2	Install drill chuck.	Mill	vise	drill chuck	
3	Find datum lines for X and Y of the bigger hole	Mill	vise	edge finder, drill chuck	900
4	Centerdrill and drill the hole.	Mill	vise	Center drill, V drill bit, drill chuck	3500
5	Find datum lines for X and Y of the two smaller holes	Mill	vise	edge finder, drill chuck	900
6	Centerdrill and drill the holes.	Mill	vise	Center drill, H drill bit, drill chuck	3600

Part Number: ME350-009 Revision Date: 15/10/2013

Part Name: Linkage spacer

Team Name: Fouro

Raw Material Stock: 6061-T6 Aluminum plate, 2" x10" ,0.5" thick.

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Cut the piece into 1 in. x 1 in.	Bandsaw			
2	Hold the part in vise	Mill	vise		
3	Mill one end of part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
4	Remove part from vise. Break all edges by hand.			file	
5	Place part in vise to machine other end of part. Mill the part to 0.85 in. length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840
6	Remove part from vise. Break all edges by hand.			file	
7	Mill the side of the part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
8	Remove part from vise. Break all edges by hand.			file	
9	Place part in vise to machine other side of part. Mill the part to 0.85 in. length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840

10	Remove part from vise. Break all edges by hand.			file	
11	Remove cutter and collet. Install drill chuck. Return part to vise.	Mill	vise	drill chuck	
12	Find datum lines for X and Y.	Mill	vise	edge finder, drill chuck	900
13	Centerdrill and drill the holes.	Mill	vise	Center drill, 1/2" drill bit drill chuck	1600

Part Number: ME350-011 Revision Date: 15/10/2013

Part Name: Backpack holder support

Team Name: Fouro

Raw Material Stock: 6061-T6 Aluminum round stock, 1 in Diameter, 7in Long

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Use band saw to cut the part into the length of 2.1 in.			Bandsaw	
2	Hold the round stock in the clamp of lathe machine	Lathe	3 Jaw Chuck		
3	Turning the through hole in the center of the round stock	Lathe	3 Jaw Chuck	F turning drill	2000
4	Remove part from clamp.				
5	Face both surfaces of the spacer to make its length 2 in.	Lathe	3 Jaw Chuck	Facetool	1000

Appendix B.3 Bill of Materials

Bill of Materials of Motion Generation						
#	**Description**	**Use**	**Dimensions**	**Supplier**	**Part #**	**Price**
1	Square Aluminum Tubing	To build input and output link	1" x 1" x 6' long, 1/8" wall	kit	-	0
2	Aluminum Plate	To bulid ground plate, input link support, and coupler	1/2" x 12" x 12"	kit	-	0
3	Aluminum round stock	To attach backpack holder to round stock, supports for input link attachment to ground plate	1in Diameter, 6in Long	kit	-	0
4	SAE 841 Bronze Sleeve Bearing(x4)	To reduce friction in joints	3/8" dia. Shaft, 1" long	kit	-	0
5	Shoulder Screw(x2)	To attach links to coupler and to ground plate	3/8" Shoul. Dia, 2-3/4" L Shoul., 5/16"-18 Thread	X50	91259A630	0
6	SAE 841 Bronze Thrust Washers(x3)	To reduce surface axial load	3/8" dia. shaft, 1/16" thick	X50	5906K511	0
7	Nuts(x8)	To secure shoulder screws and socket head cap screws	(1/4) - 20	X50	97149A100	0
8	Socket Head Cap Screws(x7)	To attach input link support to ground	1/4-20 x 2-1/2" long	X50	91251A552	0
9	Washers(x4)	To disperse axial load	1/4"	X50	91083A029	0
10	Needle Thrust Bearing(x4)	To reduce friction in axial direction	.375" bore	Grainger	4XFN1	3.02
11	High collar washers(x10)	To disperse axial load	1/4"	X50	91083A029	0
12	Shoulder Screw	To attach links to coupler and to ground plate	3/8" Shoul. Dia, 3-1/2" L Shoul., 5/16"-18 Thread	Master Carr	91259A638	2.67
13	Bolt	To attach backpack holder to coupler	1/4-20 4" long	Jack's Hardware		

Appendix B.4 Assembly Instructions

Assembly Manual: Motion Generation

The motion generation part needs to be assembled with carefulness. The general idea is to first install those attachments to their links, and then assemble those links together. Inserting bushings, washers, and spacers in correct order is needed to reduce friction and run the linkage.

Step 1. Press fit bearings into input, output, coupler, and spacers.

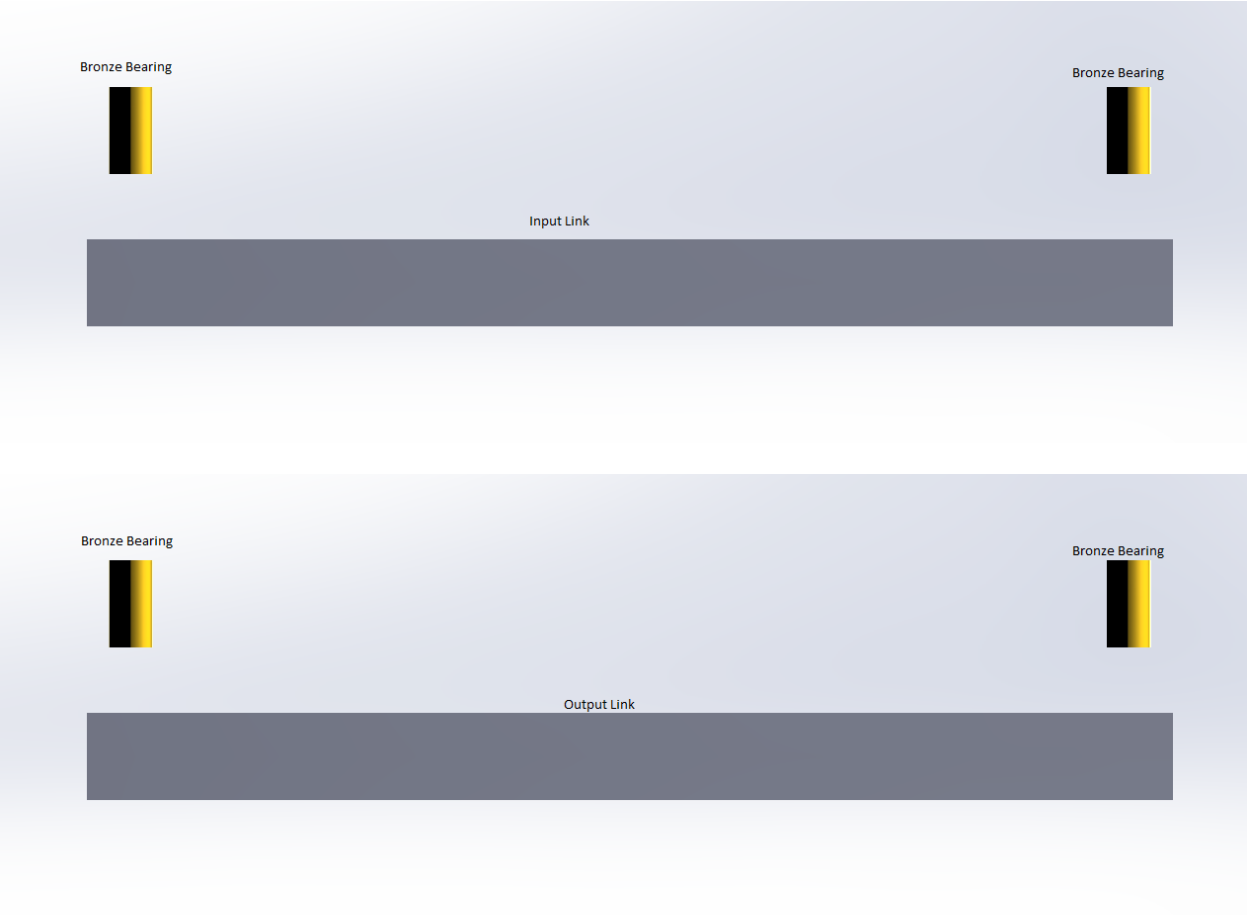


Figure B - 11: Step 1 Exploded View of Pressing fit bearings

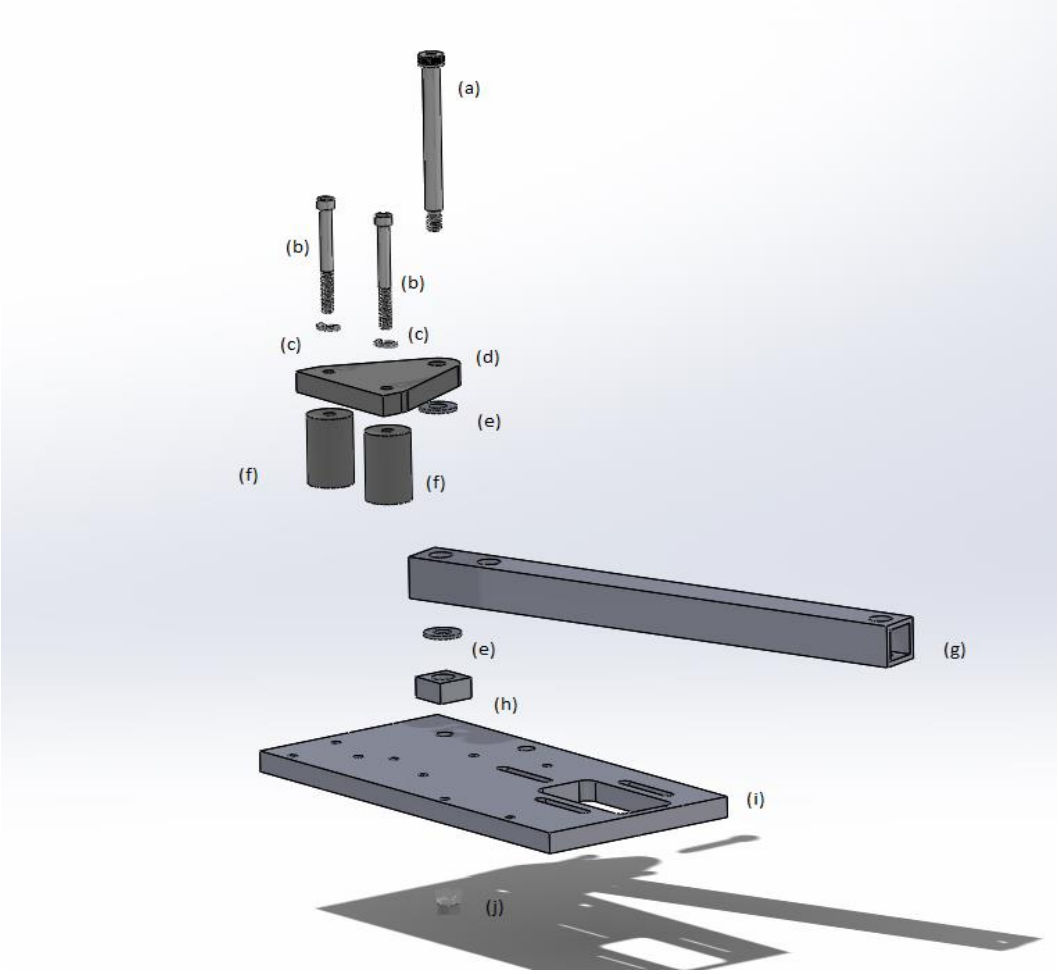


Figure B - 12: Step 2 & 3 exploded view of input link attachment

Step 2. Attach input link(g) to ground plate(i), put shoulder screw(a) through:

1. Top clamp(d)
2. Needle thrust washer(e)
3. Input link(g)
4. Needle thrust washer(e)
5. Spacer(h)
6. Ground plate(i)
7. Nut(j)

Step3. Put socket head screw(b) through:

1. Washer(spring lock)(c)
2. Top clamp(d)
3. Supports(f)

Turn the screw all the way down to ground plate(i).

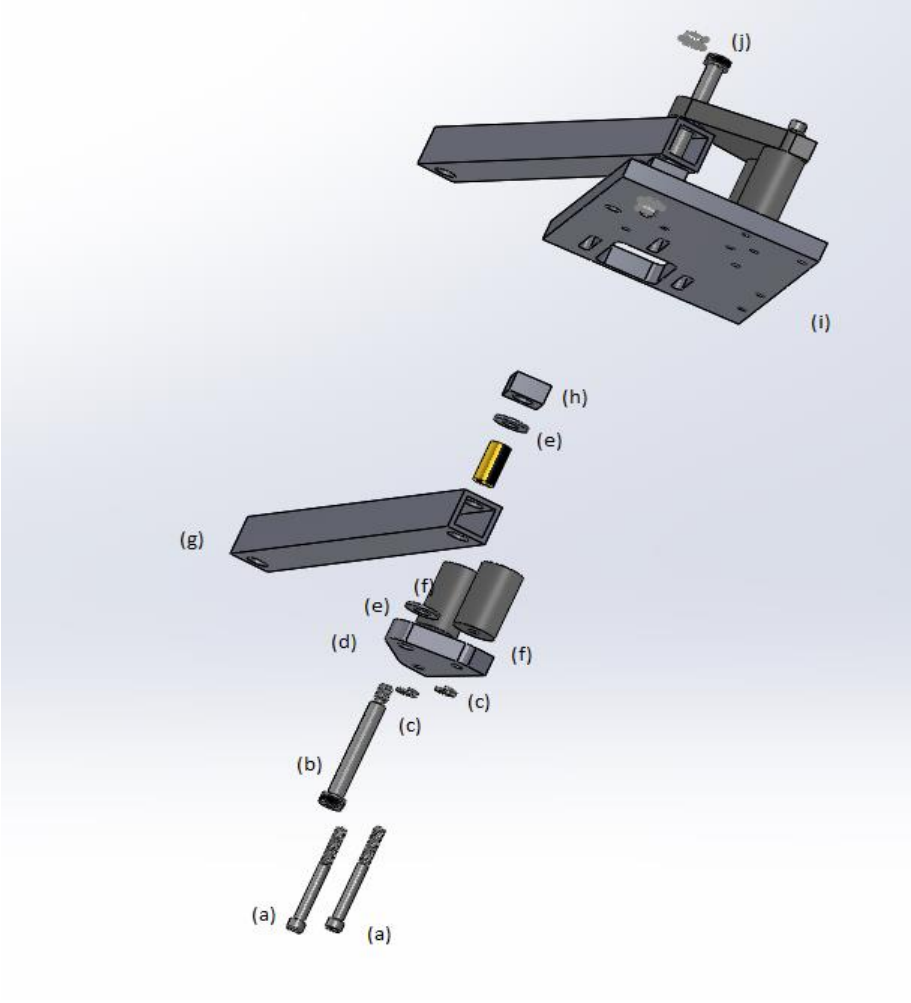


Figure B - 13: Step 4 &5 exploded view of output link attachment

Step 4. Attach output link(g) to ground plate(i), put shoulder screw(b) through:

1. Bottom clamp(d)
2. Needle thrust washer(e)
3. Output link(g)
4. Needle thrust washer(e)
5. Spacer(h)
6. Ground plate(i)
7. Nut(j)

Step 5. Put socket head screw(a) through:

1. Washers (spring lock)(c)
2. Bottom clamp(d)
3. Supports(f)

Screw into ground plate(i).

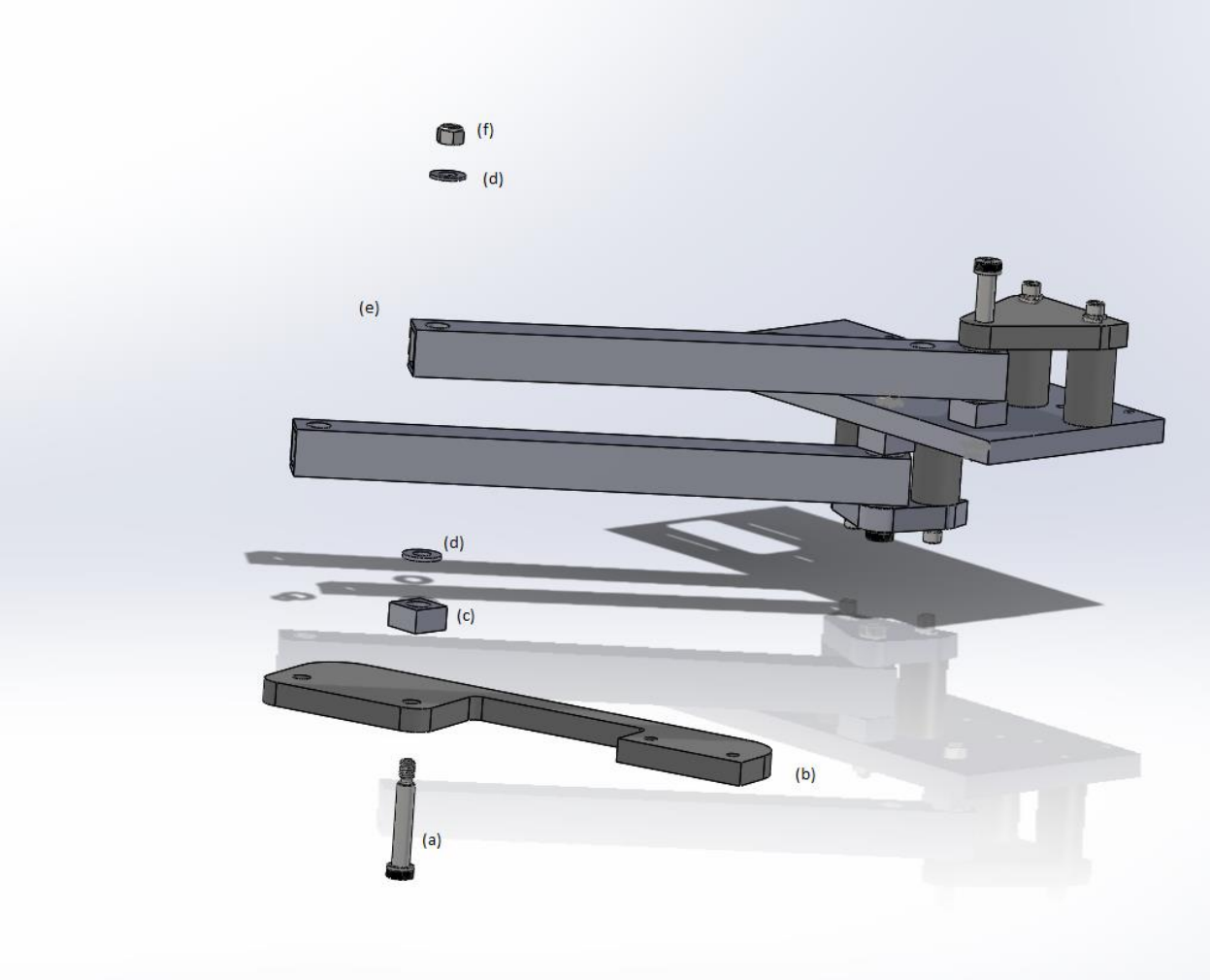


Figure B - 14: Step 6 exploded view of attaching input link to coupler

Step 6. Attach input link to coupler(b). Put shoulder screw(a) through:

1. Coupler(b)
2. Spacer(c)
3. Needle thrust washer(d)
4. Input link(e)
5. Needle thrust washer(d)
6. Nut(f)

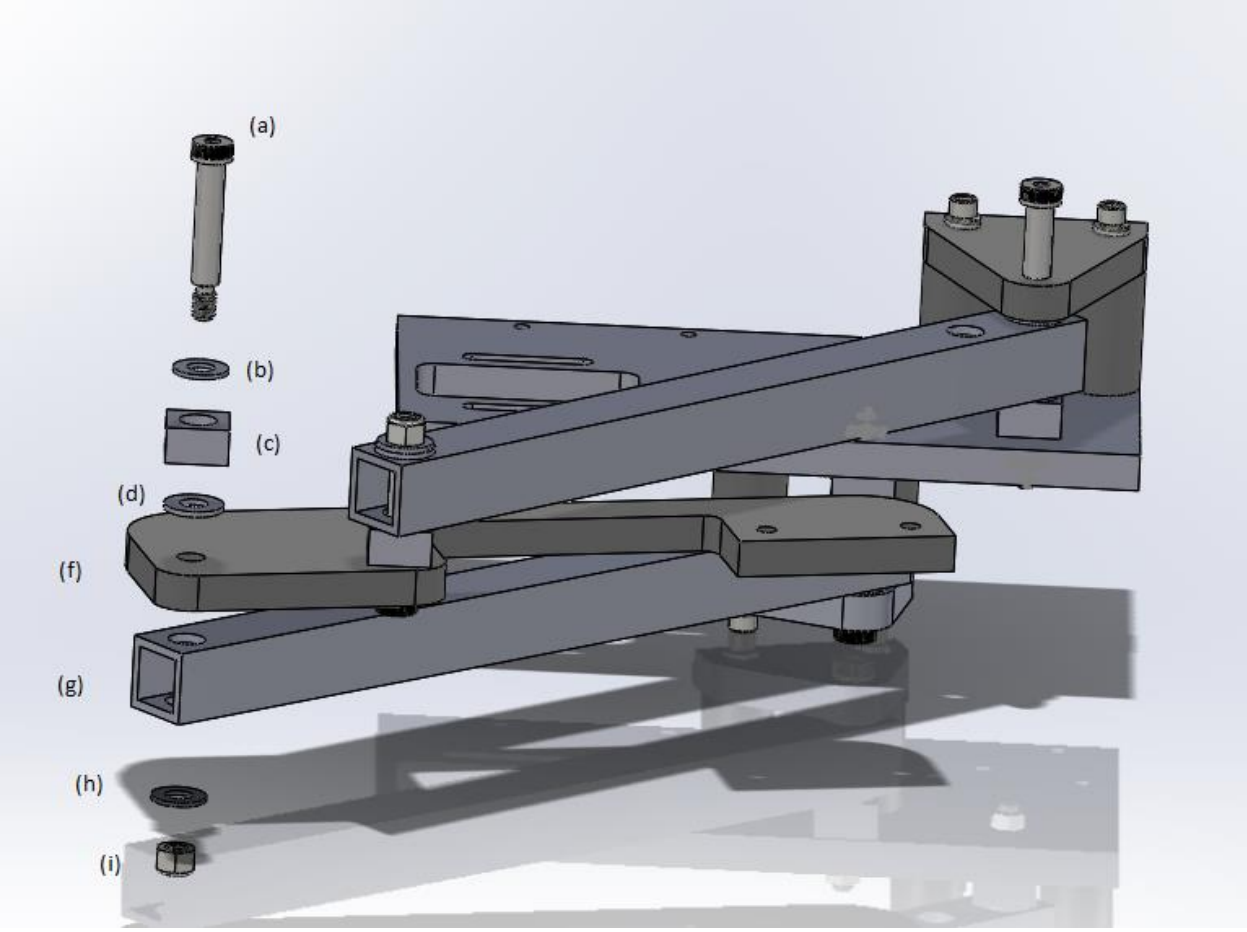


Figure B - 15: Step 7. exploded view of attaching output link to coupler

Step 7. Attach output link(g) to coupler(f), put shoulder screw(a) through:

1. Washer(b)
2. Coupler(f)
3. Spacer(c)
4. Needle thrust washer(d)
5. Output link(g)
6. Needle thrust washer(h)
7. Nut(i)

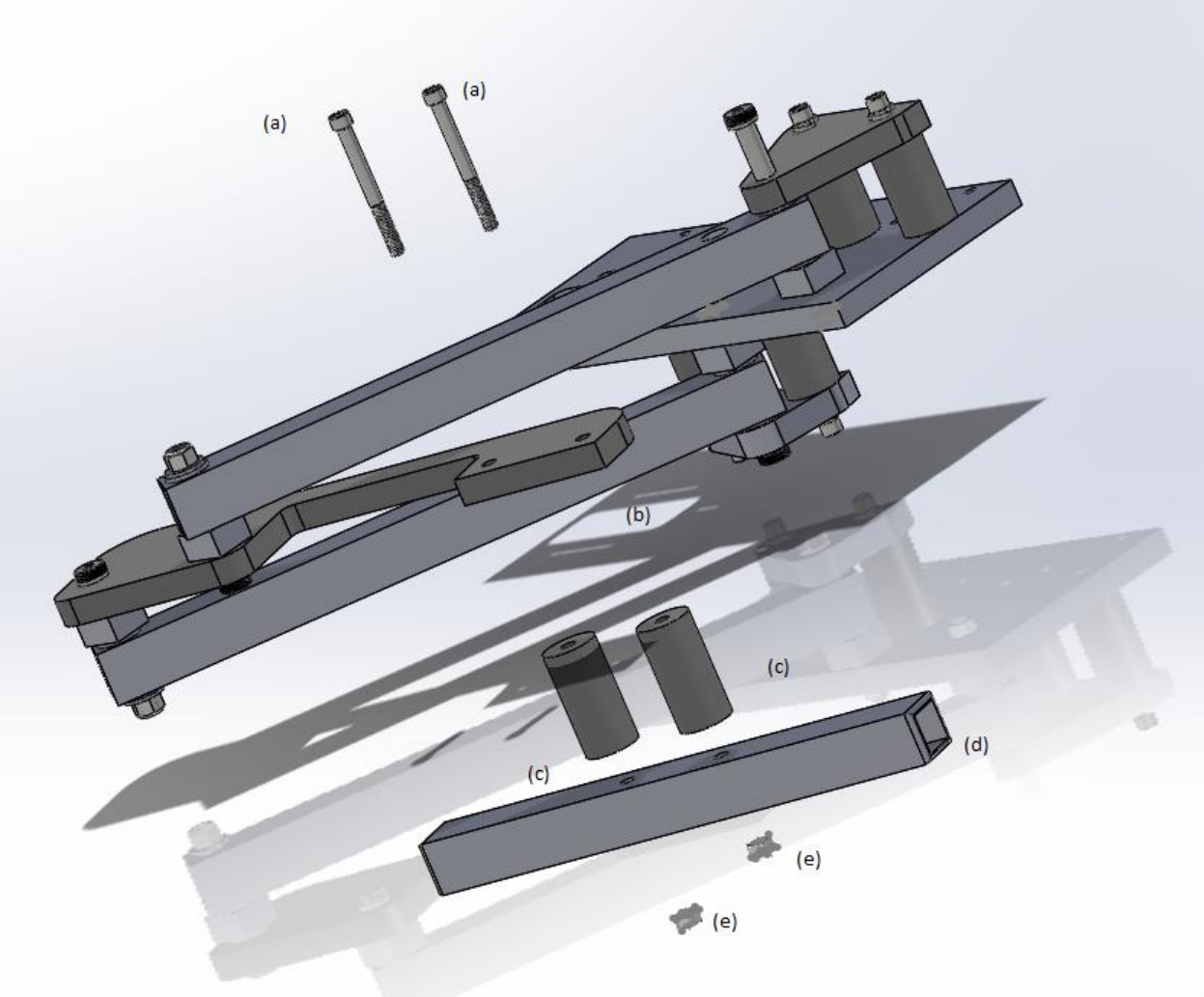


Figure B - 16: Step 8 exploded view of backpack holder attachment

Step 8. Attach backpack holder(d) to coupler(b), put socket head screws(a) through:

1. Supports(c)
2. Backpack holder(d)
3. Attach nuts(e)

Appendix C Transmission Manufacturing & Assembly

Appendix C.1 Design Drawings

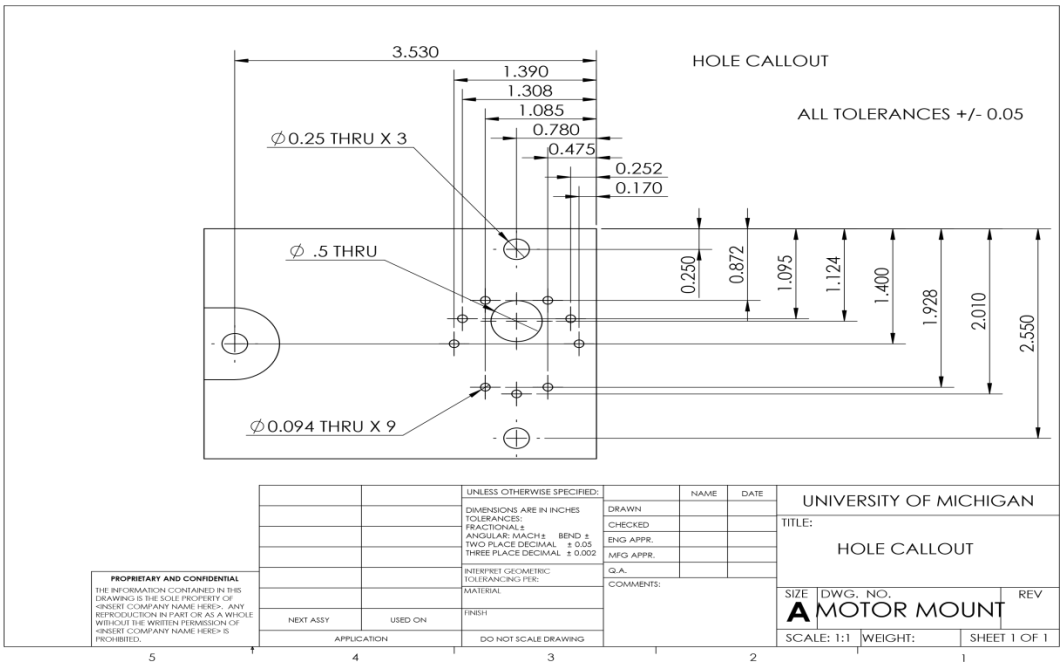


Figure C - 1: Engineering drawing for motor mount

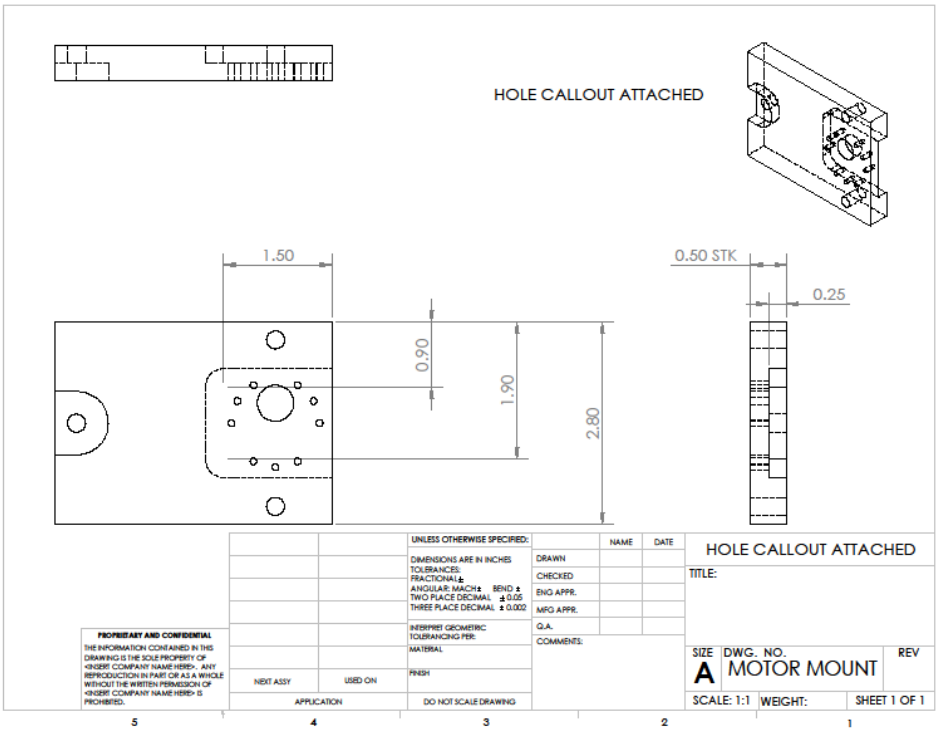


Figure C - 2: Engineering drawing for motor mount

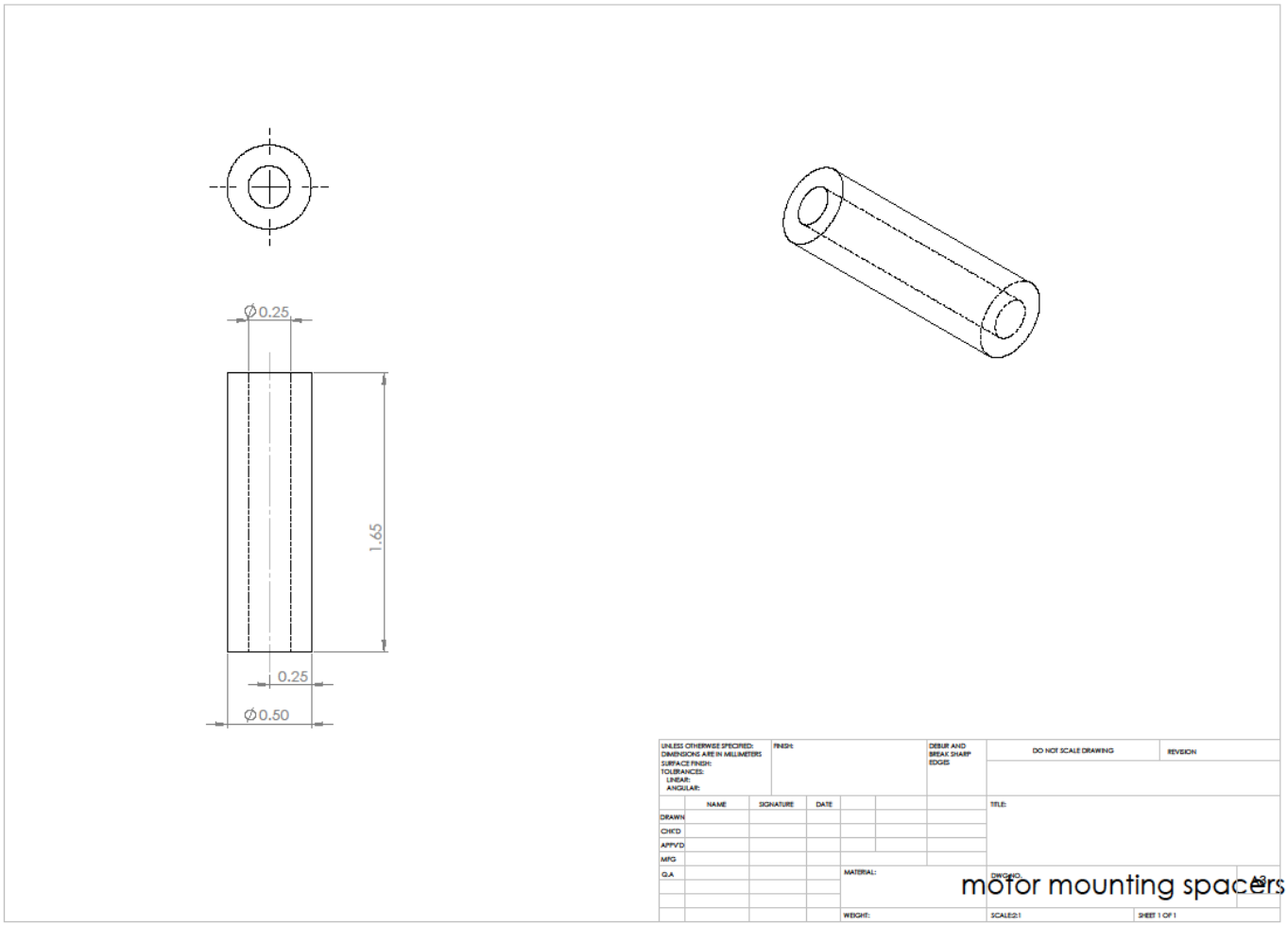


Figure C - 3: Engineering Drawing for Motor Mount Spacer

Appendix C.2 Manufacturing Plan

Part Number: ME350-008 Revision Date: 15/10/2013

Part Name: Motor Mount

Team Name: Four0

Raw Material Stock:

6061-T6 Aluminum plate, 12"x12",0.5" thick.

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Cut piece to 3" by 4.1"	Bandsaw	-	-	300
2	Hold part in vise.	Mill	vise		
3	Mill one end of part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
4	Remove part from vise. Break all edges by hand.			file	
5	Place part in vise to machine other end of part. Mill the part to 3.90 length, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840
6	Remove part from vise. Break all edges by hand.			file	
7	Mill one end of part, just enough to provide a fully machined surface.	Mill	vise	3/4 inch 2-flute endmill, collet	840
8	Place part in vise to machine other end of part. Mill the part to 2.80 width, taking several passes at .05 inches per pass. Turn off the spindle, and measure part with calipers.	Mill	vise	3/4 inch 2-flute endmill, collet	840

9	Remove part from vise. Break all edges by hand.			file	
10	Remove cutter and collet. Install drill chuck. Return part to vise.	Mill	vise	drill chuck	
11	Find datum lines for X and Y.	Mill	vise	edge finder, drill chuck	900
12	Centerdrill and drill the holes.	Mill	vise	Center drill, M(.09375) drill bit, F bit, 1/2" bit drill chuck	1600

Part Number: ME350-010 Revision Date: 15/10/2013

Part Name: Motor Mounting Spacer

Team Name: Fouro

Raw Material Stock:

6061-T6 Aluminum round stock, 0.5 in Diameter, 7in Long

Step #	***Process Description***	***Machine***	***Fixtures***	***Tool(s)***	***Speed (RPM)***
1	Use band saw to cut the part into the length of 2 in.			Bandsaw	
2	Hold the round stock in the clamp of lathe machine	Lathe	3 Jaw chuck		
3	Turning the through hole in the center of the round stock	Lathe	3 Jaw chuck	1/4 inch turning drill	2000
4	Remove part from clamp.				
5	Face both surfaces of the spacer to make its length 1.65 in.	Lathe	3 Jaw chuck	Facetool	1000

Appendix C.3 Bill of Materials

Bill of Materials of Transmission						
#	**Description**	**Use**	**Dimensions**	**Supplier**	**Part #**	**price**
1	Aluminum Plate	To build motor mount, linkage spacer	1/2" x 12" x 12"	kit	-	0
2	Aluminum round stock	To build motor mounting spacers	1in Diameter, 6in Long	kit	-	0
3	Socket Head Cap Screws(x3)	To attach motor mount to ground	1/4-20 x 2-3/4" long	X50	91251A553	0
4	Pulley A	To transmission use, attached with motor	# of grooves: 10, Pitch Diameter: 0.637", Outside Diameter 0.617"	SPDSI	A 6Z 3-10SF03708	5.53
5	Pulley B	To transmission use, attached with input link	# of grooves: 72, Pitch Diameter: 4.585", Outside Diameter 4.564"	SPDSI	A 6K 3-72NF03712	11.47
6	Timing belt	To transmission use, connecting Pulley A with Pulley B	# of grooves: 114, Belt Width: 0.375", Pitch: 0.200"(XL), Pitch Length: 22.8"	SPDSI	A 6R 3-114037	6.07

Appendix C.4 Assembly Instructions

Assembly Manual: Transmission

In transmission system assembly, the main goal is to install motor and pulleys into the original linkage system. To fix them, the motor mounts should be first assembled. After installing motors and pulleys, a timing belt is applied to connect them.

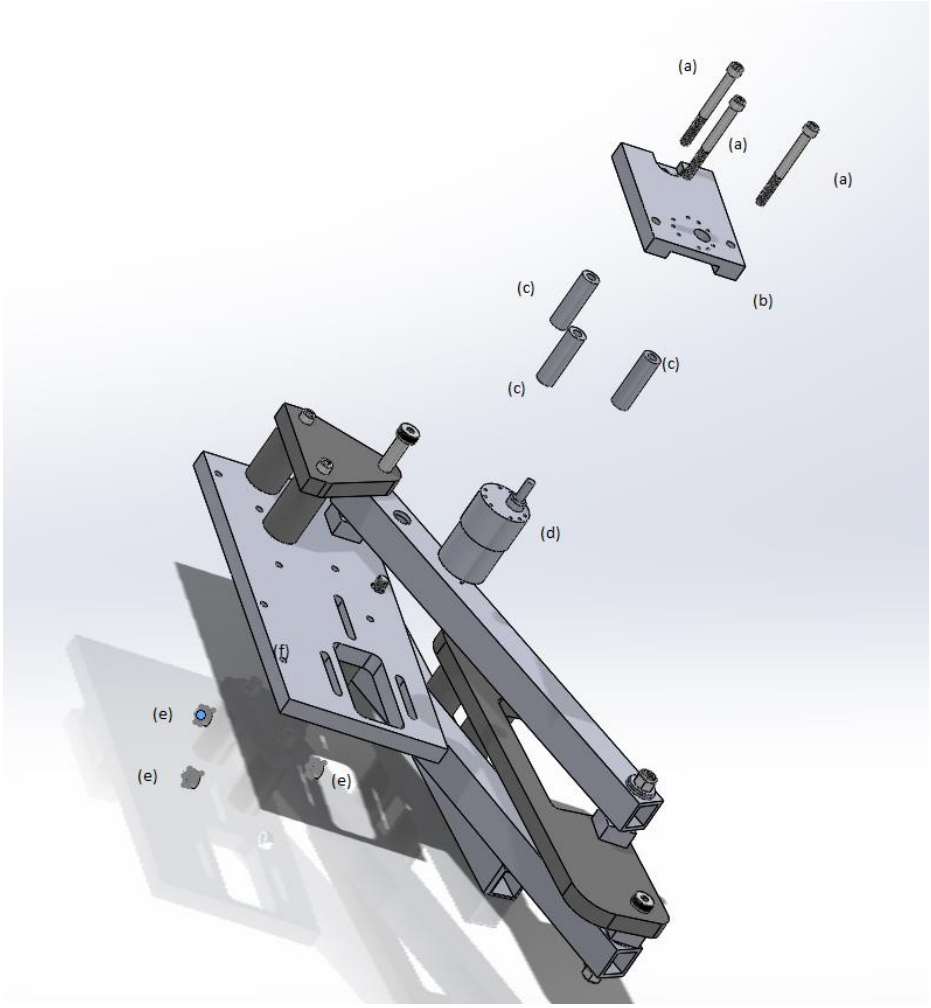


Figure C - 4: Step 1&2 exploded view of motor mount attachment

Step 1. Attach motor mount to ground plate(f), put bolts(a) through:

1. Motor Mount(b)
2. Supports(c)
3. Ground Plate(f)
4. Attach nuts(e)

Step 2. Attach motor(d) to motor mount(b), put the motor shaft through the hole in the motor mount, then screw the motor into the mount.

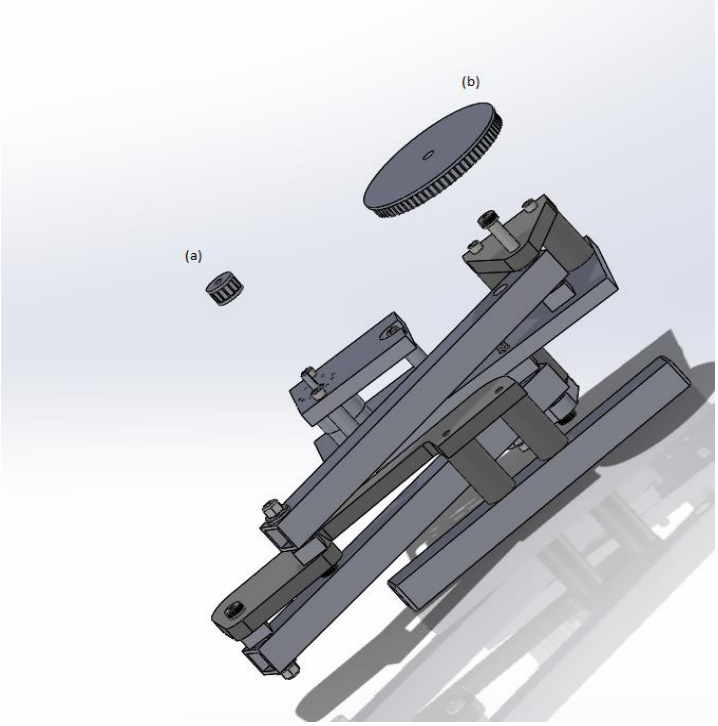


Figure C - 5: Step 3 exploded view of pulley attachment

Step 3. Attach small pulley(a) to the motor shaft with a set screw and the large pulley(b) to the shoulder screw that connects the input link to the ground plate.

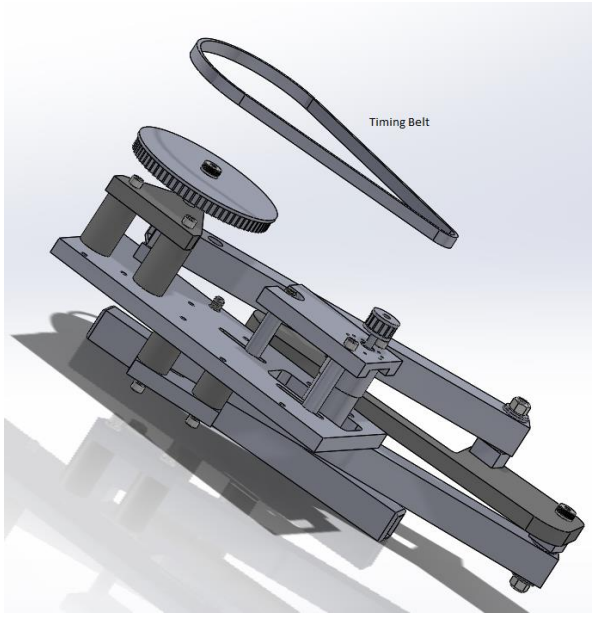


Figure C - 6: Step 4 exploded view of timing belt attachment

Step 4. Put the timing belt over the two pulleys and then tighten by sliding the motor mount and then securing the bolts on the motor mount.

Appendix D Wiring and Controller Code

Appendix D.1 Wiring Diagram

Appendix D.1.1 Wiring Diagram

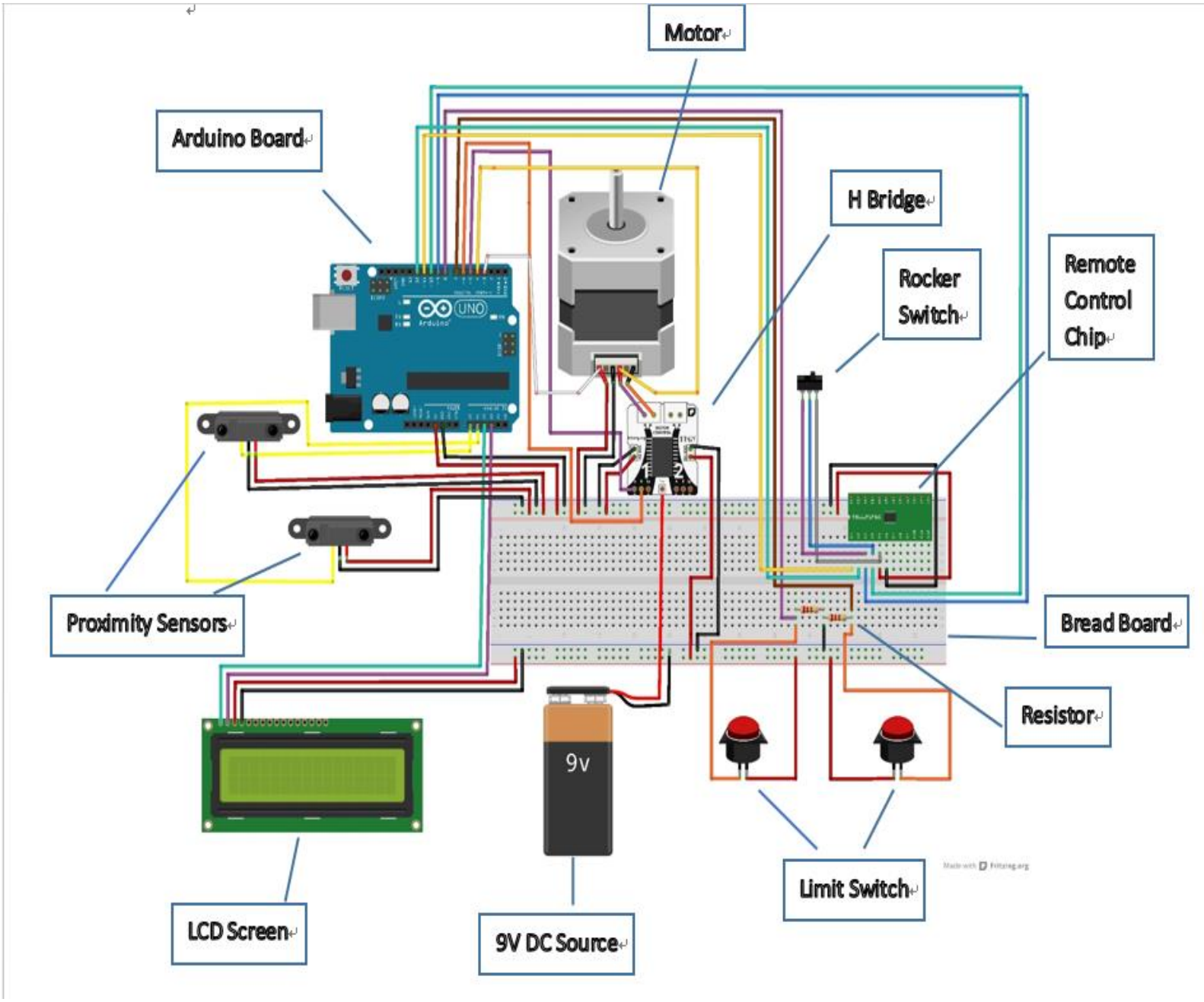


Figure D - 1: Overall wiring diagram

The detailed pin definition of the circuit can be found in Table *5-2*.

Appendix D.1.2 Limit Switch Circuit Discussion

The usage of two resistors in limit switch connections is also our design highlight. The resistor connects in series, one side with signal pin, limit switch and source, and the other side with ground. It makes sure that when the switch turns on, the signal pin will get the 5V from source, and when it turns off, the signal pin will connects to ground directly. There will be no unstable condition to disturb our control.

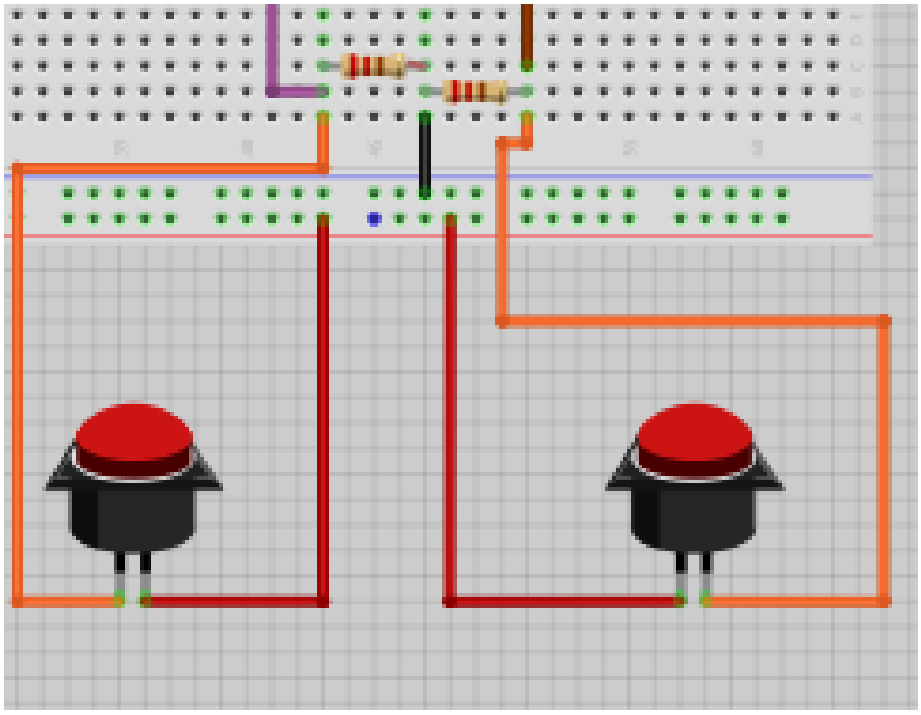


Figure D - 2: Limit switches circuit diagram

Appendix D.2 Controller Code

The Arduino microcomputer code used for controlling the mechanism is shown below.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <stdlib.h>
#include <EEPROM.h>

#define proximityA A0
#define proximityB A1
#define emergenceStop 2
#define encoderA 3
#define encoderB 4
#define motorP 5
#define motorN 6
#define hardStopFront 7
#define hardStopBack 8
#define rockerFront 9
#define rockerBack 10
#define remoteA 9  //Also used for the rocker switch
#define remoteB 10  //Also used for the rocker switch
#define remoteC 11
#define remoteD 12
#define buzzer 13
#define sdaPin A4
#define sclPin A5

//variable declaration
LiquidCrystal_I2C lcd(0x27,16,2);
char a[10]={
  '0','1','2','3','4','5','6','7','8','9'};
float currentSpeed=0;
float encoderCount;
float lastEncoderCount=0;
float currentEncoderCount=0;
float encoderRatio=32;
float gearRatio=6*131;
float toDeg=360;
int stateMachine=0;
float highSpeed=240;
float lowSpeed=165;
int safeRange=100;
int slowDownAngle=65;
int maxAngle=205;
int threshold=150;
unsigned long lastTime=0;
unsigned long currentTime=0;
unsigned long timeStamp=0;
float integrate=0;
float Kp=1;
```

```
float Ki=1;
float Kd=0;
int baseCommand=0;// Base Torque

//initialize system
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  attachInterrupt(1,recordEncoder,CHANGE);
  attachInterrupt(0,stopAll,RISING);// Comment this line to disable obstacle check
  pinMode(proximityA,INPUT);
  pinMode(proximityB,INPUT);
  pinMode(emergenceStop,INPUT);
  pinMode(encoderA,INPUT);
  pinMode(encoderB,INPUT);
  pinMode(motorP,OUTPUT);
  pinMode(motorN,OUTPUT);
  pinMode(hardStopFront,INPUT);
  pinMode(hardStopBack,INPUT);
  pinMode(rockerFront,INPUT);
  pinMode(rockerBack,INPUT);
  pinMode(remoteA,INPUT);
  pinMode(remoteB,INPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(remoteC,INPUT);
  pinMode(remoteD,INPUT);
  pinMode(sdaPin,OUTPUT);
  pinMode(sclPin,OUTPUT);
  motorControl(0);
  digitalWrite(buzzer,HIGH);
  printMessage("Go Blue!",1,1);
  encoderCount=EEPROM.read(0)*gearRatio*encoderRatio;
  delay(500);
  digitalWrite(buzzer,LOW);
}

/*main loop tate machine definition
 0: Stop & Wait 1: Forward Full 2: Backward Full 3: Backward Fix 4: Forward Fix
 */
void loop() {
  saveAngle();
  lcdAnglePrint(obtainAngle());
  if(checkObstacleExist()){
    stateMachine=0;
  }
  switch(stateMachine){
  case 0:
    {
      motorControl(0);
      printMessage("Waiting!",1,0);
```

```
    while(stateMachine==0)
    {
     if(checkObstacleExist()){
       digitalWrite(buzzer,HIGH);
     }
     else
     {
      if(obtainPosition()==-1 && (digitalRead(remoteA)==1)) {
        encoderCount=0;
        integrate=0;
        timeStamp=millis();
        stateMachine=1;
      }
      else if(obtainPosition()==1 && (digitalRead(remoteB)==1)) {
        encoderCount=maxAngle*encoderRatio*gearRatio/toDeg;
        integrate=0;
        timeStamp=millis();
        stateMachine=2;
      }
      else if(obtainPosition()==0 && (digitalRead(remoteA)==1)) stateMachine=3;
      else if(obtainPosition()==0 && (digitalRead(remoteB)==1)) stateMachine=4;
      else stateMachine=0;
     }
    }
    break;
   }
  case 1:
   {
    fullTurn(1);
    break;
   }
  case 2:
   {
    fullTurn(-1);
    break;
   }
  case 3:
   {
    fixTurn(1);
    break;
   }
  case 4:
   {
    fixTurn(-1);
    break;
   }
  }
}

//interrupt for safety
void stopAll(){
```

```
  motorControl(0);
  printMessage("Obstacle!",1,0);
  stateMachine=0;
}

//interrupt for angle recording
void recordEncoder(){
  if( digitalRead(encoderA) == digitalRead(encoderB) ){
    encoderCount--;
  }
  else{
    encoderCount++;
  }
}

//measure angular speed of rotation and output velocity in deg/s
float encoderSpeedRead(){
  float currentEncoderSpeed;
  lastEncoderCount=currentEncoderCount;
  currentEncoderCount=encoderCount;
  lastTime=currentTime;
  currentTime=millis();
  currentEncoderSpeed=(currentEncoderCount-lastEncoderCount)/((float(currentTime-lastTime))/1000);
  return currentEncoderSpeed;
}

//convert encoder speed measurement to link rotation angluar speed
float convertToMotorSpeed(float encoderSpeed) {
  return encoderSpeed/(gearRatio*encoderRatio)*toDeg;
}

//control motor speed and direction
void motorControl(int com) {
  if(com>0) {
    analogWrite(motorP,com);
    analogWrite(motorN,0);
  }
  else if(com<0) {
    analogWrite(motorP,0);
    analogWrite(motorN,-com);
  }
  else if(com==0) {
    analogWrite(motorP,0);
    analogWrite(motorN,0);
  }
}

//implement proportional and integral controller
int piControl(float targetSpeed, float currentEncoderSpeed){
  int speedCommand=0;
  float startTime=0;
```

```
  if(lowSpeedTimeStamp==0) startTime=timeStamp;
  else startTime=lowSpeedTimeStamp;
  float speedError=targetSpeed-currentEncoderSpeed/(gearRatio*encoderRatio)*toDeg;
  float measuredAngle=float(abs(currentEncoderCount-encoderStamp))/(gearRatio*encoderRatio)*toDeg;
  float desiredAngle=float((millis()-startTime))/1000*targetSpeed;
  float angleError=desiredAngle-measuredAngle;
  speedCommand=int(float((Kp*speedError+Ki*angleError+baseCommand+baseCommand))*speedToV
oltage);
  if (speedCommand>255) speedCommand=255;
  else if (speedCommand<0) speedCommand=0;
  return speedCommand;
}

//Obtain current position of lincage: front=1, middle=0, back=-1
int obtainPosition() {
  if(digitalRead(hardStopFront)==0 && digitalRead(hardStopBack)==0) return 0;
  else if(digitalRead(hardStopFront)==1 && digitalRead(hardStopBack)==0) return 1;
  else if(digitalRead(hardStopFront)==0 && digitalRead(hardStopBack)==1) return -1;
  else return 100;
}

//Convert encoder reading into real angle
float obtainAngle() {
  float convertedAngle=((float)encoderCount)/encoderRatio/gearRatio*toDeg;
  //Serial.println(convertedAngle);
  return convertedAngle;
}

//check the infrared proximity sensor to see if obstacle exist
int checkObstacleExist(){
  if((analogRead(proximityA)<threshold) && (analogRead(proximityB)<threshold))
  {
    //Serial.println("No Obstacle!");
    return 0;
  }
  else {
    delay(100);
    if(!((analogRead(proximityA)<threshold) && (analogRead(proximityB)<threshold))) {
      printMessage("Obstacle!",1,0);
      Serial.print("A=");
      Serial.println(analogRead(proximityA));
      Serial.print("B=");
      Serial.println(analogRead(proximityB));
      return 1;// Set to 0 to disable Obstacle check
    }
    else return 0;
  }
}

//make a full turn of the mechanism with speed reduction at final 30 degrees
void fullTurn(int dir) {
```

```
  float encoderSpeed=encoderSpeedRead();
  if(dir==1) {
    if(obtainPosition()<=0) {
      printMessage("Going Forward!",1,0);
      if(((int)obtainAngle())<(maxAngle-slowDownAngle))  {
        motorControl(highSpeed);
      }
      else {
        motorControl(lowSpeed);
      }
    }
    else {
      printMessage("Bag at Side!",1,0);
      Serial.print("Total Time Used = ");
      Serial.print(millis()-timeStamp);
      Serial.println("ms");
      motorControl(255);
      motorControl(0);
      digitalWrite(buzzer,HIGH);
      delay(1000);
      digitalWrite(buzzer,LOW);
      stateMachine=0;
    }
  }
  else if(dir==-1) {
    if(obtainPosition()>=0) {
      printMessage("Going Back!!",1,0);
      if(((int)obtainAngle())>slowDownAngle) {
        motorControl(-highSpeed);
      }
      else {
        motorControl(-lowSpeed);
      }
    }
    else {
      printMessage("Bag at Back!",1,0);
      Serial.print("Total Time Used = ");
      Serial.print(millis()-timeStamp);
      Serial.println("ms");
      motorControl(0);
      digitalWrite(buzzer,HIGH);
      delay(1000);
      digitalWrite(buzzer,LOW);
      stateMachine=0;
    }
  }
  else digitalWrite(buzzer,HIGH);
}

//turn the mechanism slowly to initial or final position
void fixTurn(int dir) {
```

```
  if(dir==1) {
    if(obtainPosition()<=0) {
      printMessage("Going Forward!",1,0);
      motorControl(150);
    }
    else {
      motorControl(0);
      printMessage("Bag at Side!",1,0);
      digitalWrite(buzzer,HIGH);
      delay(1000);
      digitalWrite(buzzer,LOW);
      stateMachine=0;
    }
  }
  else if(dir==-1) {
    if(obtainPosition()>=0) {
      printMessage("Going Back!",1,0);
      motorControl(-150);
    }
    else {
      printMessage("Bag at Back",1,0);
      motorControl(0);
      digitalWrite(buzzer,HIGH);
      delay(1000);
      digitalWrite(buzzer,LOW);
      stateMachine=0;
    }
  }
  else digitalWrite(buzzer,HIGH);
}

//print number on lcd
void lcdAnglePrint(float angle) .
{
  String angleConverted=String((int)angle);
  String angleString="Angle: ";
  String degString=" deg";
  String angleMsg=angleString+angleConverted+degString;
  int i=1;
  lcd.setCursor(1,1);
  while(i<=angleMsg.length()){
    lcd.write(angleMsg.charAt(i-1));
    i++;
  }
}

// print text message on the screen
void printMessage(String msg, int ifPrintAngle, int times) {
  lcd.clear();
  if(times==0) {
    lcd.setCursor(((int)(14-msg.length())/2+1),0);
```

```
    int i=1;
    while(i<=msg.length()){
      lcd.write(msg.charAt(i-1));
      i++;
    }
    if(ifPrintAngle) lcdAnglePrint(obtainAngle());
    return;
  }
  else {
    int roller=0;
    while(roller<=times*14) {
      lcd.clear();
      lcd.setCursor((((int)(14-msg.length())/2+1+14*times-roller)%14),0);
      int i=1;
      while(i<=msg.length()){
        lcd.write(msg.charAt(i-1));
        i++;
      }
      roller++;
      delay(100);
    }
    if(ifPrintAngle) lcdAnglePrint(obtainAngle());
  }
}

//save the current calculated angle of the link
void saveAngle() {
  int value = (int)(obtainAngle());
  EEPROM.write(0,value);
}
```

Appendix D.3 Drawing for Motor Mount

As discussed in section 5.8 Sensor Mount, the sensors are mounted to the mechanism using double sided tape. No manufacturing plan is necessary here. The drawing for the position of the sensors are shown in Figure D - *3* and Figure D - *4*.

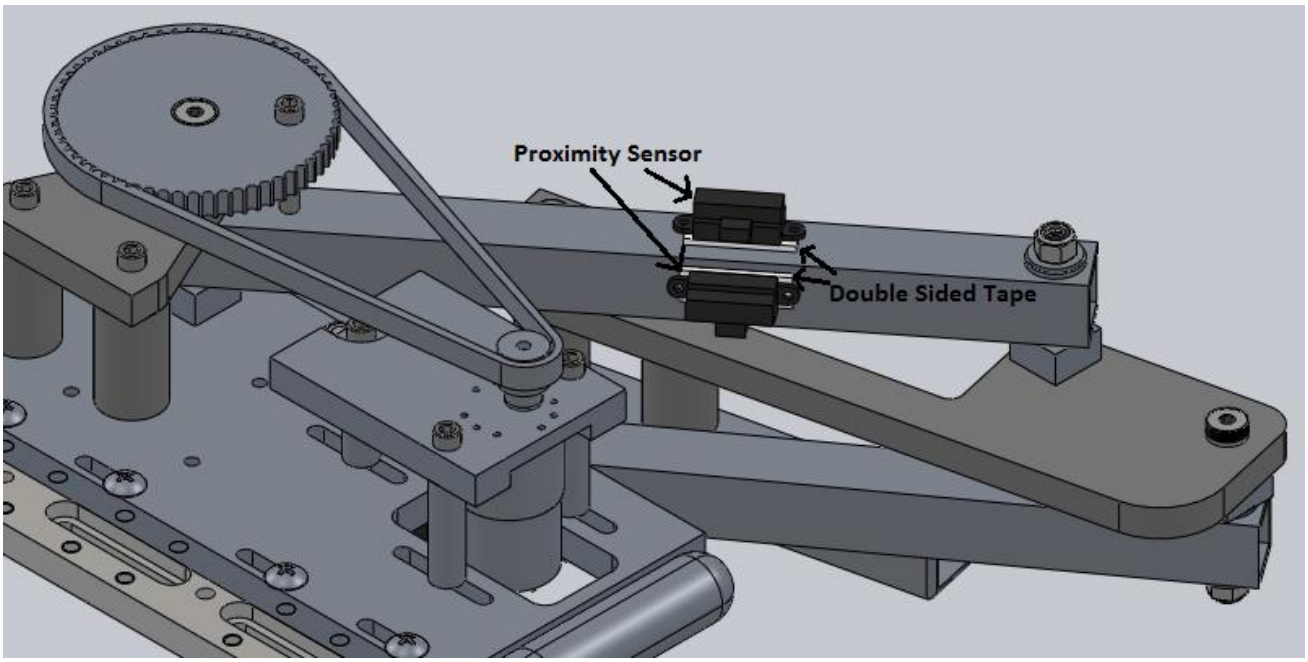


Figure D - 3: Proximity sensor position

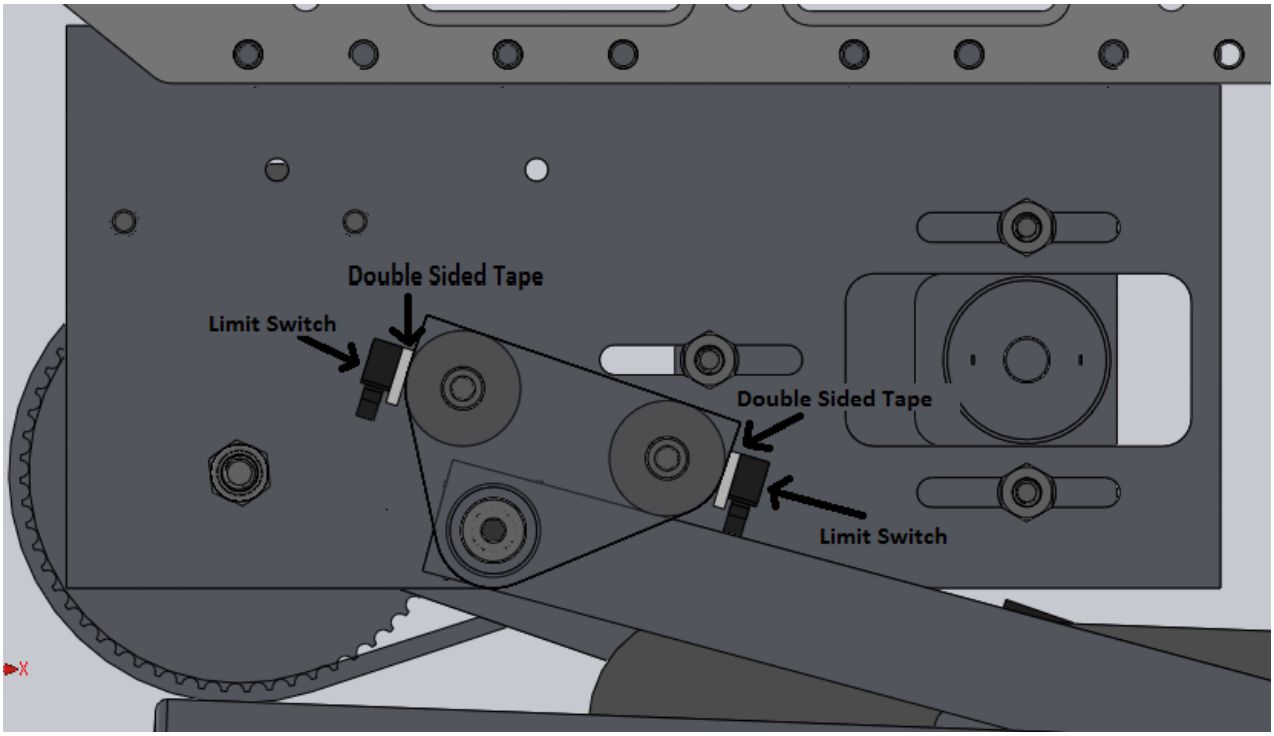


Figure D - 4: Limit switch position

Appendix D.4 Calculation for Encoder Count

Similar to the calculation done in section **5.6.2 Encoder Sensor**, the value of the encoder count as a function of the actual angle of rotation for the output link was given by the equation shown below.

$$\mathrm{EncoderCount(Angle)} = \mathrm{Angle} \times \mathrm{GearRatio} \times \mathrm{SmaplingRate}$$

The value of the **GearRatio** term in the equation was $6 \times 131 = 786$ and the sampling rate can take the value of 16, 32 or 64 depending on the mode of sampling. For the purpose of efficiency and accuracy, the channel A voltage change was used for the interruption pin on Arduino to keep track of the encoder count. This method requires the variable **SamplingRate** to be set equal to 32.

Appendix D.5 Bill of Materials for Controller Circuit

The bill of materials for the controller circuit is shown in *Table D - 1*.

No.	Description	Use	Supplier	Part No	Price
1	Remote sensor, remote controller	Applied circuit so that we add the function of remote control the motion of device	Xinde Eletronics	2262/2272	2.13
2	LCD screen for Arduino	Applied to display the status of operation	Xinde Eletronics	Arduino IIC/I2C 1602	3.28
3	Arduino sensor shield	Combined with Arduino board to make the circuits easier and clear to assemble.	Taobao	sensor shield v5.0	2.46
4	Double sided tape	Install sensors and limit switches to the mechanism	3M		0

Table D - 1: Bill of materials for controller circuit

Appendix E Additional Documents

Appendix E.1 Updated Functional Decomposition

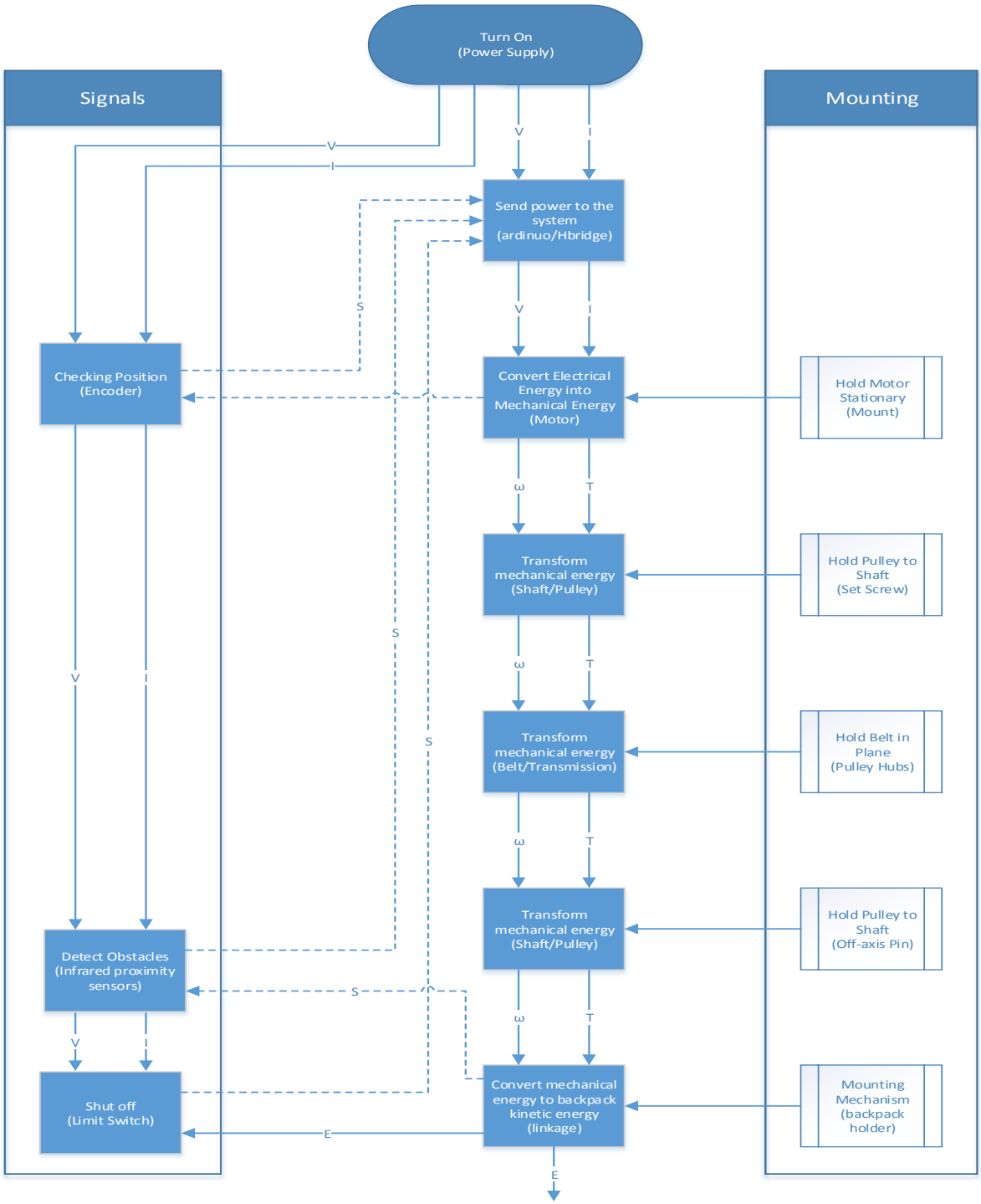


Figure 6-2: Updated functional decomposition

Appendix E.2 Figure for Assembled Mechanism

Pictures for the manufactured part are shown in Figure E - *1*,Figure E - *2* and Figure E - *3*.

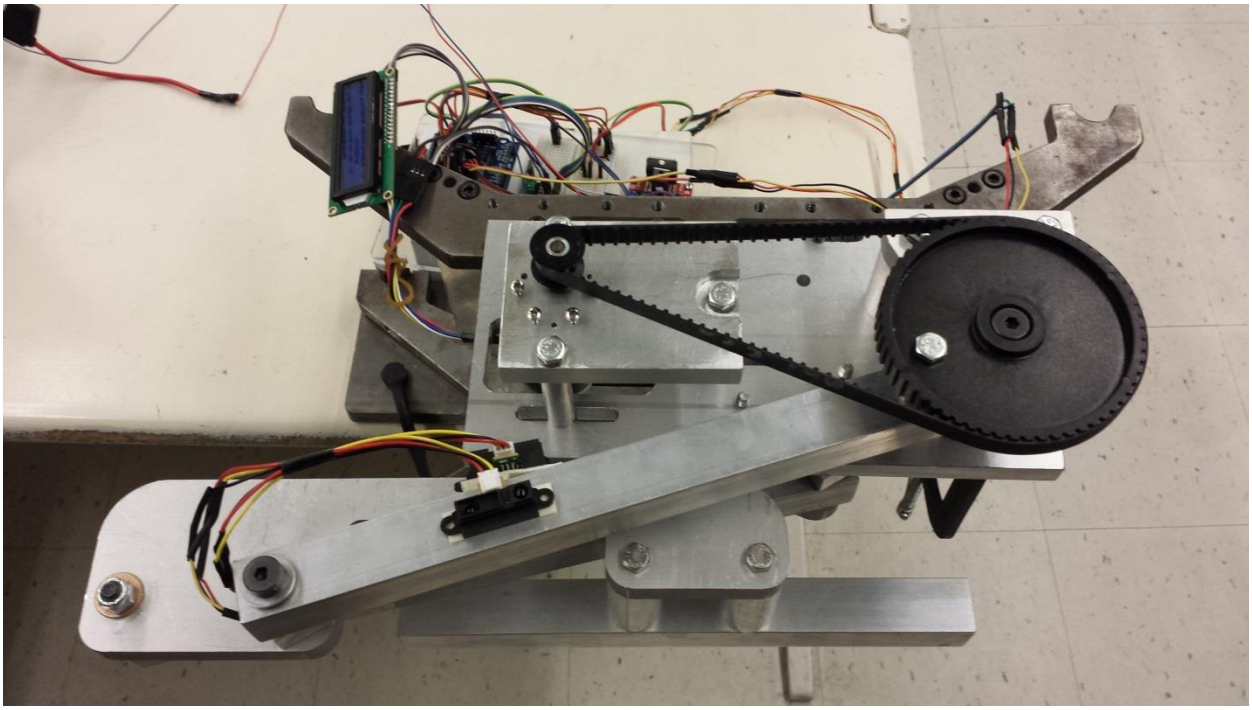


Figure E - 1: Starting position of the assembled mobile backpack carrier

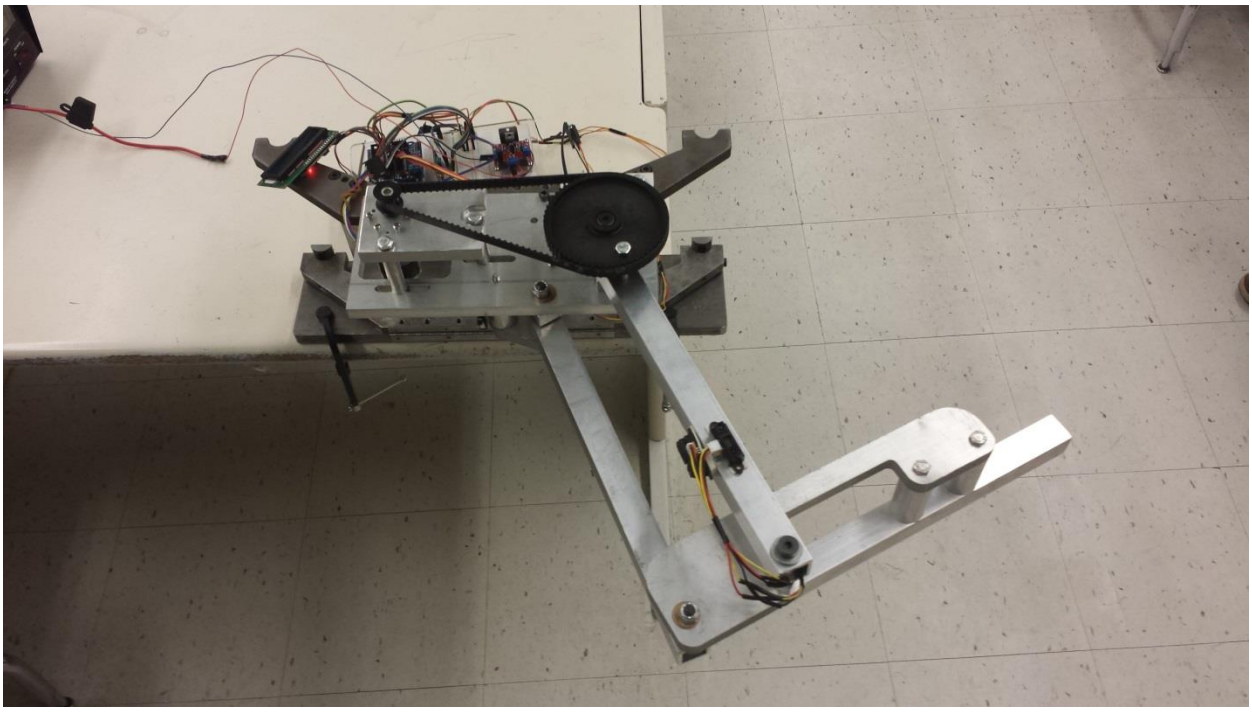


Figure E - 2: Transition position of the assembled mobile backpack carrier

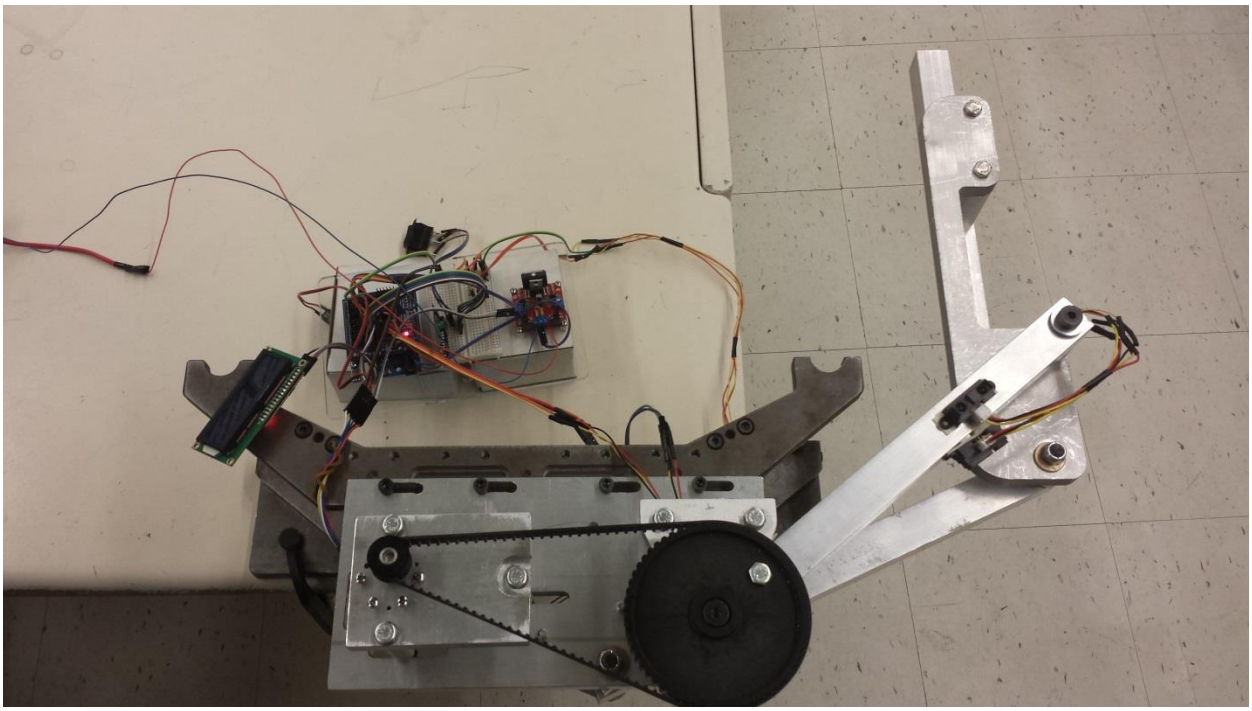


Figure E - 3: Final position of the assembled mobile backpack carrier

Appendix E.3 Project Gantt Chart