Using the Engineering Design Process

Box Mover Project

Graham Kroll

In fulfillment of

ENGR 3414 Project Management Design and Entrepreneurship

12/12/24

Presented To

Dr. Owen Schipplein

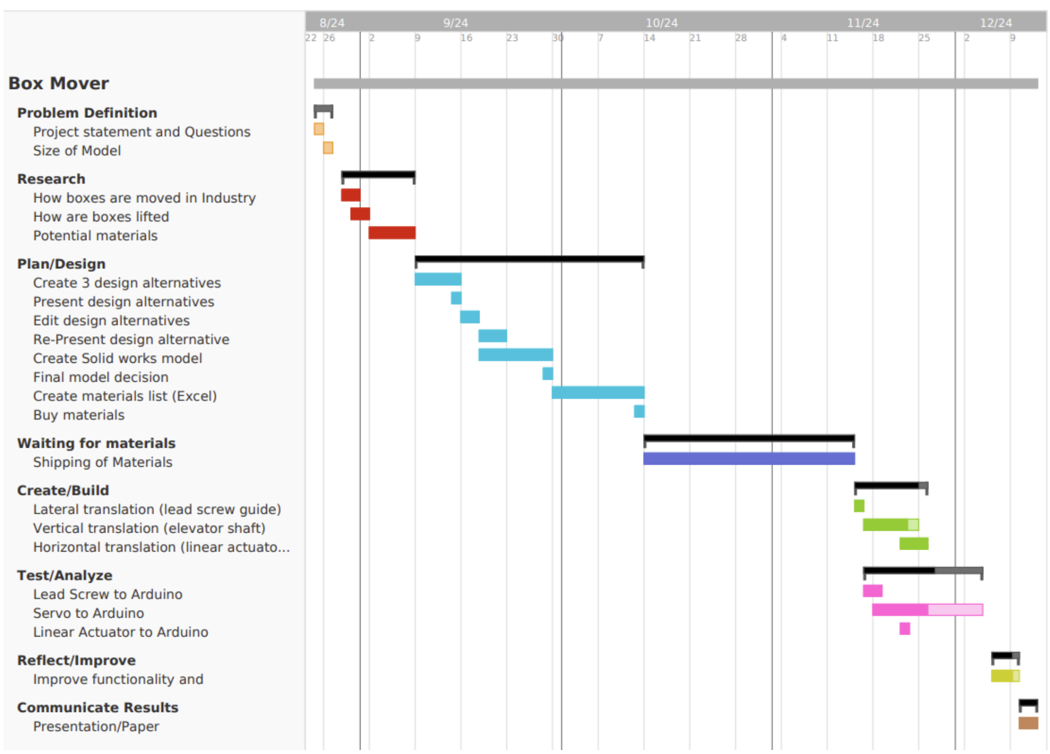
Table of Contents

Problem Statement

The members of ENGR 3414 have embarked upon a project to minimize injuries to their employees of their newly started business. The business is involved with sorting products by weight so that they can be transported to different vehicles for the purpose of shipping to their customers. To minimize injuries, the team will design a machine or device and build the prototype for a feasibility study. The machine will sort packages both vertically and horizontally placing the heaviest packages on the bottom and the lightest packages on the top. When the forklift driver places the package on the machine the package will have a color code indicating a weight. From the color code the machine will place the package in the appropriate bin. This build was to be accomplished using the seven steps of the Engineering Design Process[1].

Constraints

The project's constraints included time, budget, and resources (both in tools and engineers working on the project). The first constraint was trying to design, purchase materials, and then build the project in Fall of 2024 Semester. The timeline of this process is described in detail in the Gantt chart below. Another constraint was budget as we had to make sure that the materials we were choosing would fit within the engineering budget for the class this semester. The largest constraint was in the tools available in the lab and the people working on the project. Since I was the only student in this class, all facets of this project were handled by either myself or Dr. Schipplein.



[2]

[1] Resources – Steps of the Engineering Design Process
[2] Resources – Gantt Chart

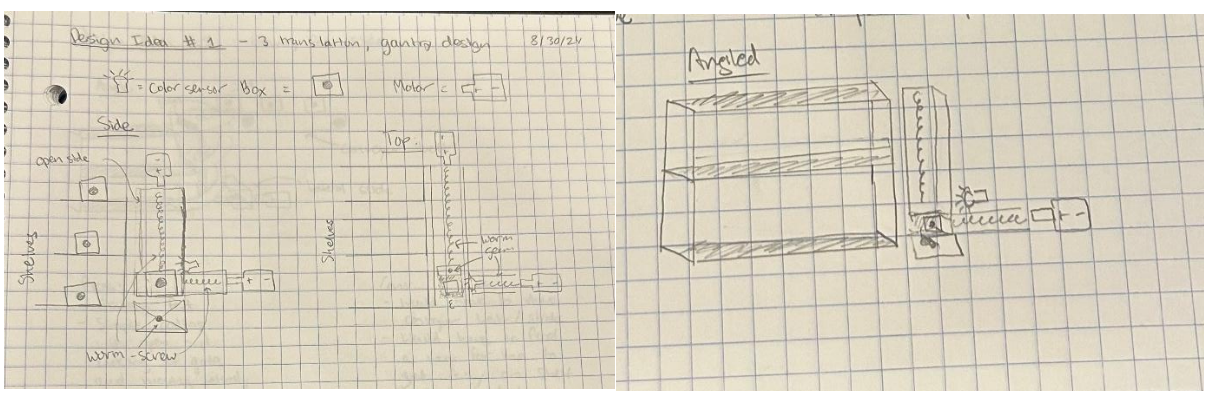
Research

Hours of research were done on how items were moved in an industrial setting. From Amazon warehouse to car manufacturing to rail cars, inspiration was pulled from the internet and past experiences to try to find a solution to this problem. This search was also guided by beginning to search for materials that were both readily available and within the budget for the semester. One of the most prominent websites was Servo City which provides small scale metal and motor product perfect for our prototype, which made it a great source of inspiration. Research was continually done throughout this project as problems arose, but the majority was conducted at the onset and in the design stage.

Design Space

In the planning phase of this project, there were numerous design ideas we thought through. Below are the design ideas created that made up our design space as well as the pros and cons for each design alternative.

Design #1: The gantry design had the idea of using 3 worm gears of varying lengths to account of the three degrees of motion we would need in the box mover (horizontal, vertical, and pushing translations). Since all three degrees of motion would be performed by the same mechanism, the integration with the Arduino board and assembly would be easy to integrate since each section would be modular. Because of the modular design, it would also be relatively easy to scale and worm gear would provide great precision of movement in all three directions. The determent of this type of design is that it would involve three motors which would mean including more electronics (batteries, Arduino boards, motor controls ext.) and the shelving would have to be lifted, or the horizontal worm-gear dug into the ground of the bottom of the vertical translation would line up with the bottom shelf.

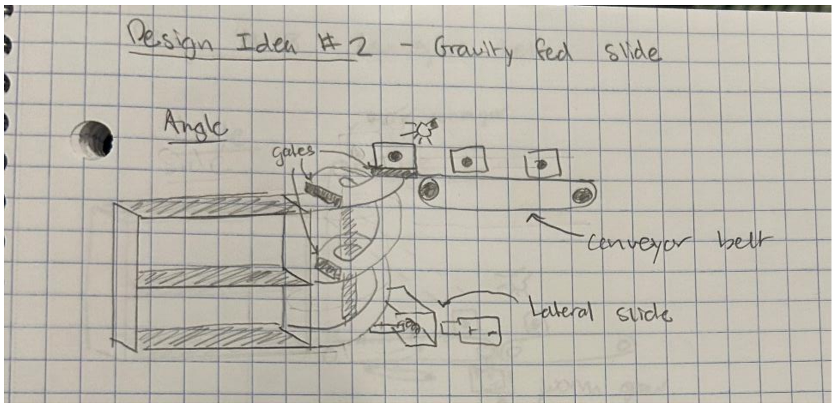


[3]

Design #2: The second design revolved around the idea of a gravity fed slide system where boxes would be on an elevated conveyor belt system and then slide down to the appropriate shelf using a series of gates to control the sorting of the boxes. The benefits of this kind of design were that it would eliminate the need for a motor in vertical translation and the coding of the project would be relatively simple with the opening and closing of gates to control the delivery of boxes.

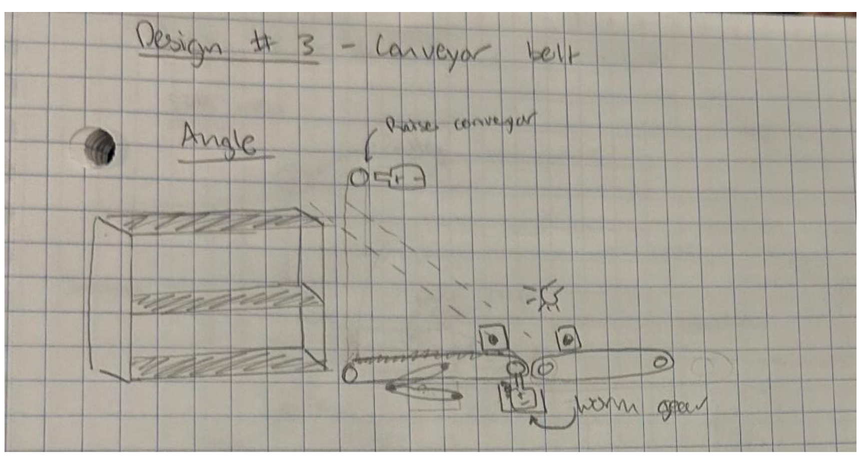
[3] Hand drawn gantry – Design #1

The cons of this design would be this design would not be modular, and the translation of the shelving to another row would be difficult. Controlling the speed of the box decent and exact potential energy required to slide fully into the shelf would also be tedious to regulate and would require the manipulation of the slide system. Another issue would be what material to build the slide out of as 3D printing was not an option and cutting sheet metal in a spiral shape would need to be precise, which would be very difficult with the tools at our disposal.



[4]

Design #3: The conveyor belt system was created with the idea of having a series of conveyor belts for with the side of the conveyer belt closest the shelving unit having a pulley system to raise and lower the leading edge of the conveyer to the desired shelf based upon the weight of the box. The benefits of this method are that the delivery system would all be controlled by conveyor making feeding onto shelving very controlled. This design would also be easily scalable and all that would need to be changed is the size of the conveyor belts. Similar problems arose with this idea to Design #2 as it would be difficult to translate the assembly to other shelfing, which would most likely require a pivoting conveyor system or a worm gear under the entire assembly to slide it horizontally. The system would require at least two motors, and the pulley system used to raise it would likely have to be on both sides of the conveyor belt so that the weight of the boxes could be distributed evenly.



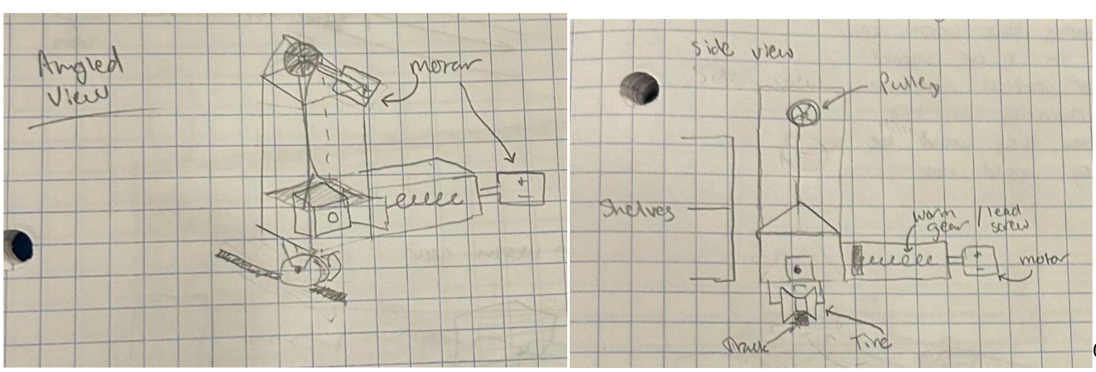
[5]

[4] Hand drawn gravity fed slide – Design #2
[5] Hand drawn conveyor belt – Design #3

Decision Matrix: After talking through the alternative solutions, the following decision matrix was made for important criteria on a scale of 1-10 and given a graded weight.

Criteria	**Weight (%)**	**Gantry**	**Gravity-Fed Slide**	**Conveyor Belt**
Cost	20%	5	6	5
Complexity	10%	7	8	5
Speed	15%	6	8	7
Scalability	15%	5	3	7
Safety	20%	9	6	7
Energy Efficiency	5%	4	9	6
Durability	15%	8	7	5
Total	100%	6.55	6.35	6.05

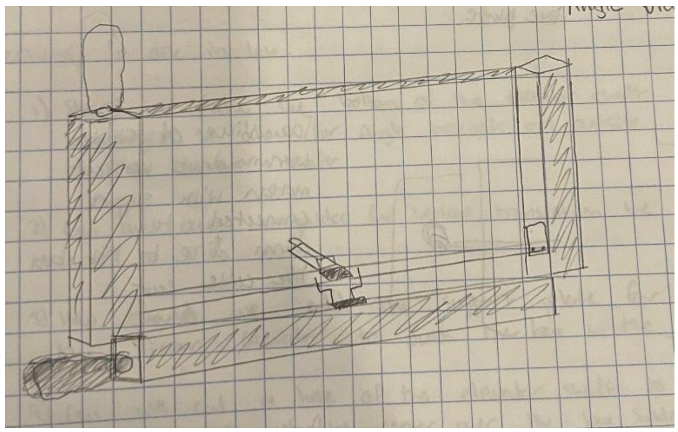
After deciding that the gantry method was the best design to pursue, the model was restructured several times to better fix the functionality and material options available. In taking over the design, the idea was created of making the horizonal motion a wheel with a tract along the front of the shelving units and the vertical motion being controlled by a system like an elevator shaft. The pushing motion of the box onto the shelf would still be accomplished through a worm gear and the whole assembly would move laterally as the tire rotated. Some of the issue foreseen in this design were how to make the rotation of the tire precise enough to stop were needed, how to attach the motor to the wheel underneath the elevator and worm gear assembly, and how unstable the whole assembly would be requiring another wheel or a track system running along the top of the shelves to hold the box move assembly in position. This design refactor can be seen in the image below.



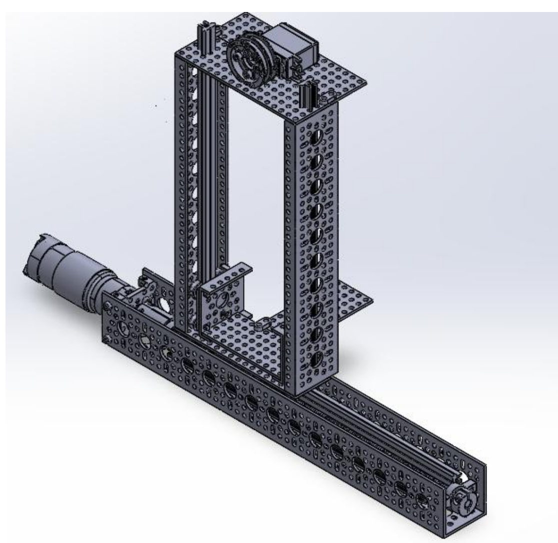
[6]

[6] Hand drawn gantry refactor 1 – Design #1

The second time the gantry design was refactored entailed making the vertical translation out of two worm gears with a fork catch along the horizonal worm gear that would only be engaged with the box assembly when the box was on the lowest level. Once the box was lifted by the two-worm gear vertical system it would disengage the horizontal fork and then eventually be pushed in by a linear actuator as shown in the drawing below. After talking through the idea, it was decided that the two worm gear vertical translation was going to be difficult to control with one motor or would require the additional cost another motor for the two worm gears to be synchronized when lifting.


[7]

After going through all pros and cons of the initial design and the refactoring, it was decided to have a worm gear for the horizonal motion, an elevator shaft controlled by a servo motor for the vertical motion, and a linear actuator for the pushing motion onto the shelving. This design provided the greatest amount of control over the exact positioning of the box, was scalable for larger sizes, and best fit the function of the materials and budget available. Once this final design was decided, a SolidWorks model was created of the design for as much of the material as


[8]

[7] Hand drawn gantry refactor 2 – Design #1
[8] SolidWorks model of final design

possible. The linear actuator was not able to be modeled and the length of the support rails on the elevator shaft unable to be shortened in the model (they stick out of the top of the elevator shaft), but otherwise the model reflects the model almost perfectly. Several design features such as the exact length of the support rails on the elevator shaft, the positioning of all the screws, string used to lift the elevator cart, attachment of this string to the elevator cart were to be decided upon construction because they would take excessive amounts of line to model in SolidWorks.

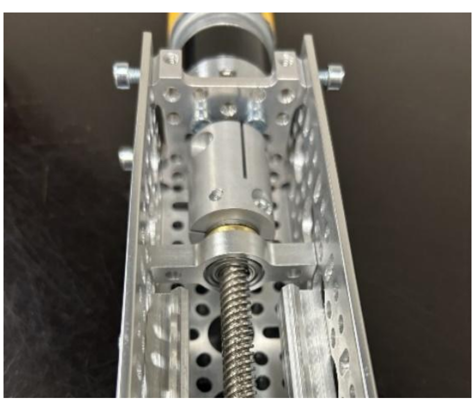
Materials List

After creating the model for the final design, a materials list[9] was made of all the parts needed to create the model as summarized in the excel tables. The tables are separated by the vendors materials were purchased from. After purchasing we realized there were several oversights including that our jumper wires were male to female rather than the needed male to male, no M4 nuts were purchased to attach to the M4 screws, and purchasing a power distribution board and volt regulator for the servo motor would have made the assembly easier.

Assembly

After receiving the parts from the vendors, the process of assembling all the materials and problem solving began.

Day 1: The first day of assembly was very straightforward, following the design of the worm gear assembly on Servo City's website. When attempting to fit the V-groove bearing onto the guide tracks, it was discovered that the M4 screws used in the V-guide lead screw dolly needed to be loosened to provide enough tolerance to slide the bearings over the guide tracks. Another issue encountered was that the hyper coupler used to attach the motor to the worm gear was loose, allowing the worm gear to slide out of the assembly when the dolly hit the pillow block support at the end of the worm gear. Both issues were resolved, and assembly then began on the elevator shaft. The elevator shaft, used for the vertical assembly, proved challenging to configure due to the initial design. Our model included an angled component between the short U-channel


10

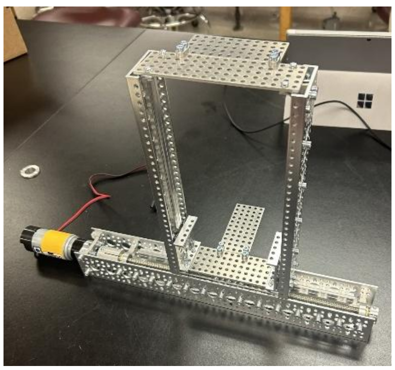

11

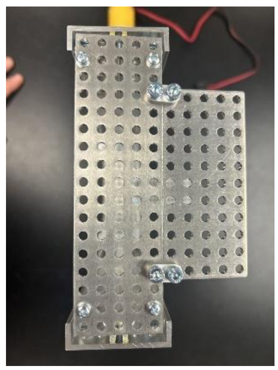

12

[9] Table of contents - Materials list Servo City and Amazon
[10] Hyper couple and pillow block
[11] Angle mount
[12] Elevator shaft and cart

and the guide tracks, which were deemed unnecessary and subsequently removed from the elevator cart. As a result of this change, the width of the bottom of the cart increased, requiring us to replace the originally planned 11-hole support plate with a 17-hole support plate for the servo motor on top of the model. The longer support plate was then cut down to fit the exact dimensions of the bottom of the elevator cart. Additionally, the guide tracks were cut to match the length of the shaft.

Day 2: On day two, more time was spent reconfiguring the elevator shaft to allow it to slide easier on the guide tracks. There was excessive friction in the assembly because of parts out of square and excessive tightness on the bearing in the sides of the cart which was the exact same problem that was fixed in the horizonal assembly. The support plates for the linear actuator and servo motor were added to the assembly. The wench attachment was then added to the servo and an attempt was made to control the servo motor with the dual motor programmer. Originally, the idea was to use an Arduino to control the servo, but it was found that the motor programmer may have to be used in the assembly and the Arduino board wired into the programmer.


[13]

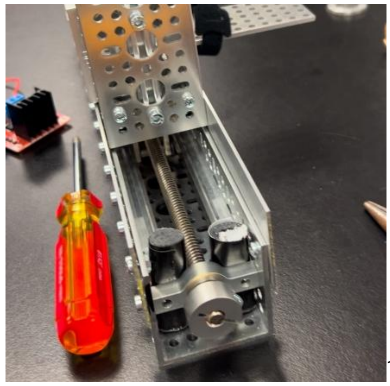

[14]

[15]

Day 3: The third day was spent trouble shooting how make the assembly slide more regularly with the elevator shaft attached as changing direction would cause the assembly to rock and decrease the speed of the dolly as the weight shifted. Changing the code and once again loosening the bearing on the dolly allowed for a smoother translation of the dolly. It was noted that the bearing could be very loose if they didn’t fall out of the dolly assembly. All screws were changed to be a uniform M4 since several M3 screws were used on the elevator shaft for a lack of materials on day 2. The linear actuator was also tested and proved to be functional and easy to code.

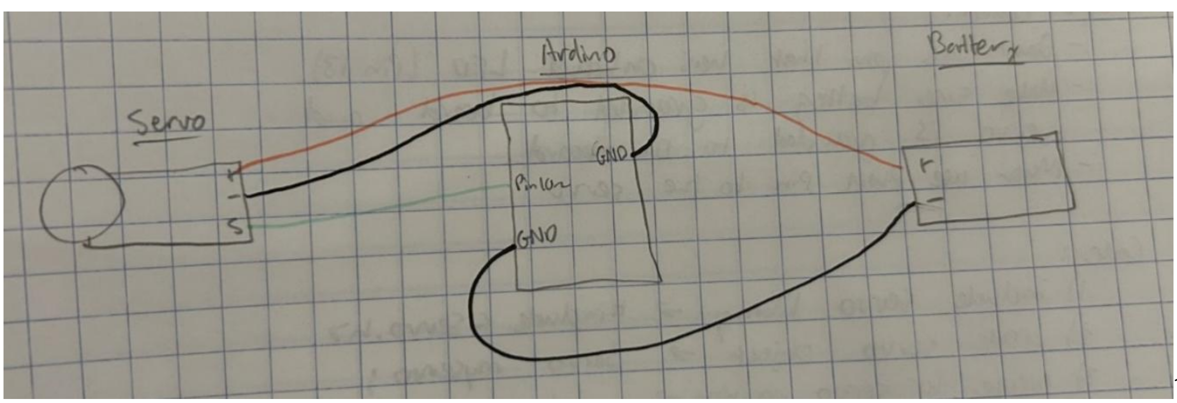
Day 4: Lab started by adding rolled up padding at the end of the worm gear and guide plates also proved to be helpful in smoother translation of the dolly. The guide rails were loosened so that friction between the elevator shaft and guide rail did not slow the speed of lateral translation. Afterwards, attempts were made with the Arduino Uno servo motor to make it function in continuous rotation mode. It was proved that the Arduino could put out

[13] Assembly with cart and servo supports
[14] Servo support
[15] Wench attachment to servo


[16]

the appropriate signal to control the servo in continuous mode and then we attempted to add a motor controller to the assembly. The motor controller was removed to attempt to get the Arduino to rotate the motor in both directions continuously. The issue was unable to be resolved tonight as more research needed to be done on the servo library code and output of the motor controller.

Day 5: After researching and refactoring the code for the servo motor, the servo would run in both directions as coded by the Arduino when the servo was powered by a battery. A key discovery here was that the power source and the servo both needed to be grounded to the servo motor. However, the motor controller could not replace the battery in the assembly. Several theories on this complication were proposed such as the code was not supplying the appropriate signal or voltage to the servo, the power source was not grounded correctly to the Arduino board, and the timing of the signal to rotate the motor may have been out of sync with the power being sent to the servo. These theories were tested and while some could be fixed, the system proved not to be operational with the motor controller in it. Later it was attempted to return to the system with the battery as the power source and this assembly proved not to work as before as well. It was determined that more research was needed to solve this problem.

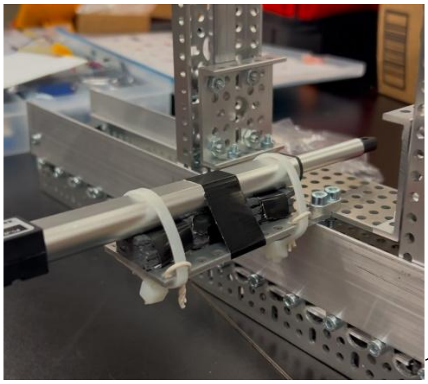


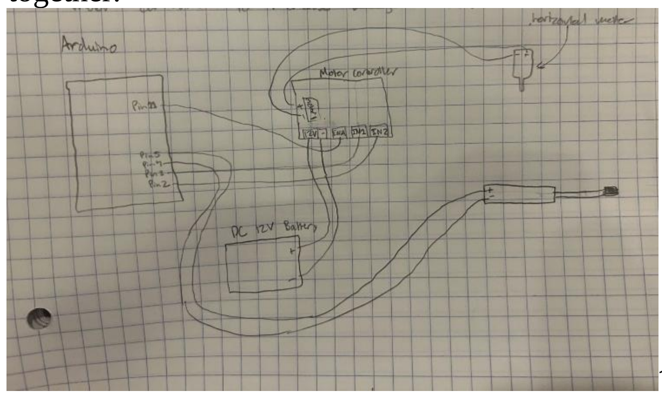
[17]

Day 6: Further research and communication with engineers are Servo City proved to be helpful for eliminating potential issues but when the battery assembly from day 5 was unable to be recreated successfully. Some of the issues checked were that a PWM pin was being used in the Arduino board and that the PWM signal range for continuous mode was correct at 900-2100 microseconds and that the frequency was fixed at 50 Hz which is equivalated to the desired 20

[16] Rolled padding and guide plates
[17] Servo system with battery as power source

millisecond desired. Therefore, it was determined that either the servo motor or the motor programmer was malfunctioning and that one would need to be replaced. The rest of the lab was spent attaching the linear actuator to the elevator assembly. Padding was added underneath the linear actuator so that it would push through the center of the box. Test code was then created to drive the worm gear and linear actuator together.


[18]



[19]

Discussion

Currently the assembly is not operational so it is not solving the desired problem but once it is assembled, it will successfully recognize and sort boxes into the appropriate shelves as requested to decrease back injuries in the company. This product will impact all material handlers, which can be most employees' at large warehouses. Back pain resulted in 25% of the work force missing 10 or more days for work amongst employees ages 18-44 last year in America ("Chronic Back Pain"). It was also found that nearly 40% of adults had back pain in 2019 (Lucas, Jacqueline W., et al). These are just a few statics why this product is crucially important to the workforce. By removing the need for the lifting of heavy boxes, employees can maintain health allowing them to work more regularly and longer in their lifetime.

Specification

At this time, the box mover assembly is not completed it can not be tested or measured for the limits and necessary features. Once the box mover assembly is completed, the specifications can be acquired for the product and added to this section.

Conclusion

This project was an example of the engineering design process from start to finish. It required a great deal of research, planning, building, testing, and problem solving. The project is currently unfinished but is very close to completion pending the restoration of the servo motor and the final integration of the three degrees of translation. In addition to individual research, there was a heavy use of AI in this project especially when trouble shooting and creating the code for the Arduino due to my limited knowledge of Arduino. ChatGPT was the main source of artificial intelligence in this project as it allowed me to express my ideas for the code without having to know the syntax to make it functional. Many of the problems faced during this project were

[18] Linear actuator attached with padding
[19] Pin up linear actuator and horizontal motor

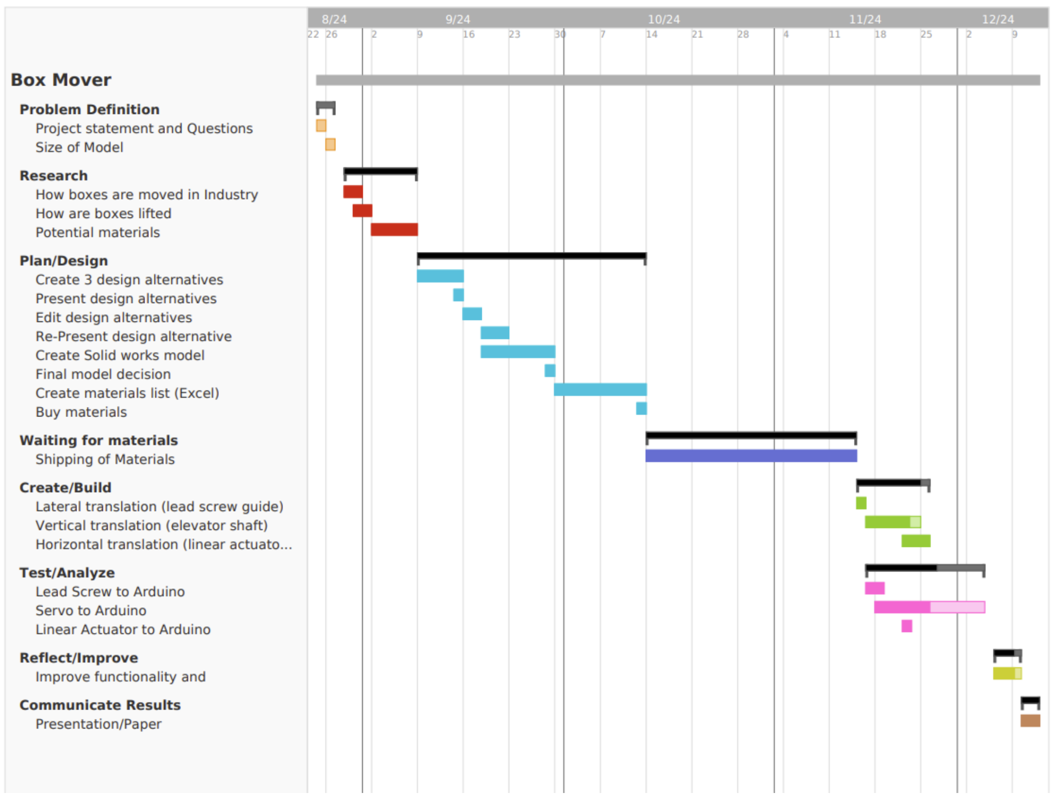
regarding the integration of the three translations both physically and coding wise. The servo motor proved to be the most difficult and what I would have researched the most beforehand so that it would have been easier to integrate and get to function correctly. I am excited to see the project completed and functioning, which will entail getting the servo motor operational, attaching the color sensor, and integrating the systems for all three translations and the color sensor.

<u>Resources</u>

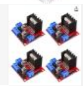
Steps of the Engineering Design Process:

1. Define problem/Ask questions
2. Research/Brainstorm
3. Plan/ Design
4. Build/Create
5. Test/Analyze
6. Reflect/Improve
7. Communicate

Gantt Chart:



Materials list - Amazon

Box Moving Machine Materials List (Amazon)					
SKU	**Quantity**	**Description**	**Image**	**Price**	**Total**
Linear Acuator for push (z)					
Linear Acutator	1	DC HOUSE Mini Electric Linear Actuator Stroke 4"-Force 4.5 lbs-12V | High-Speed		$31.99	$ 31.99
Electronics					
Battery	1	HRB 4S Lipo Battery 3000mAh 14.8V 60C RC Lipo Battery Compatible with RC Car Truck		$33.99	$ 33.99
Ardunio Board	3	Arduino Uno REV3 [A000066]		$27.60	$ 82.80
Motor Controller	1	BOJACK L298N Motor DC Dual H-Bridge Motor Driver Controller Board Module		$ 9.99	$ 9.99
Jumper Wires	1	40 PCS 20 CM (8 inch) Breadboard Jumper Wires Length Optional Dupont wire		$ 3.99	$ 3.99
			Total		$ 162.76
Color Sensor	1	Teyleten Robot GY-31 TCS3200 TCS230 Color Sensor Module Color Recognition Sensor		$ 9.99	$ 9.99

Materials list – ServoCity

Box Moving Machine Materials List (Servo City)						
SKU	**Quantity**	**Description**	**Image**	**Price**	**Total**	Notes
V-Guide Lead Screw Dolly Kit (x and y direction)						
3205-0003-0001	2	V-Guide Lead Screw Dolly Kit		$ 34.99	$ 69.98	One for vertical translation and one for horizontal
Structure of Lead Screw Dolly						
1201-0043-0002	1	Quad Block Pattern Mount		$ 6.99	$ 6.99	
2800-0004-0008	2	M4 x 0.7mm Zinc-Plated Socket Head Screw (8mm Length) - pack of 25		$ 3.19	$ 6.38	
2800-0004-0011	2	M4 x 0.7mm Zinc-Plated Socket Head Screw (11mm Length) - pack of 25		$ 3.49	$ 6.98	
2800-0004-0014	1	M4 x 0.7mm Zinc-Plated Socket Head Screw (14mm Length) - pack of 25		$ 3.69	$ 3.69	
4006-0008-1006	1	Hyper Coupler 8mm Round Bore to 6mm D-bore		$ 7.99	$ 7.99	
1613-0516-0008	2	Thrust Ball Bearing 8mm ID		$ 3.99	$ 7.98	
1606-0043-0008	2	2-side, 1 Post Pillow Block 8mm Bore		$ 6.99	$ 13.98	
3504-0804-1808	1	Lead Screw Clamping Collar 8 mm Lead Bore		$ 6.99	$ 6.99	
3501-0804-0350	1	3501 Series Lead Screw (8mm Lead, 4 Start, 350mm Length)		$ 10.99	$ 10.99	
1120-0015-0384	1	1120 Series U-Channel (15 Hole, 384mm Length)		$ 17.99	$ 17.99	
3700-0145-0288	4	3700 Series 14.5mm V-Guide (288mm Length)		$ 3.99	$ 15.96	two for vertical translation two and for horizontal
Motor for horizontal (x) translation						
5202-0002-0001	1	5202 Series Yellow Jacket Planetary Gear Motor		$ 44.99	$ 44.99	
Structure for the vertical translation (y)						
1116-0040-0088	2	1116 Series Grid Plate (5 x 11 Hole, 40 x 88mm)		$ 2.79	$ 5.58	One for bottom of elevator cart and one for support of linear acuator
1116-0040-0136	3	1116 Series Grid Plate (5 x 17 Hole, 40 x 136mm)		$ 3.49	$ 10.47	One for the bottom of elevator and two for top platform
1121-0001-0048	2	1121 Series Low-Side U-Channel (1 Hole, 48mm Length)		$ 2.99	$ 5.98	
1121-0010-0264	2	1121 Series Low-Side U-Channel (10 Hole, 264mm Length)		$ 11.99	$ 23.98	
1216-0001-0001	3	Steel 90 Degree Angle Bracket (1-1) - 4 Pack		$ 3.99	$ 11.97	
Motor for vertial translation (y)						
2000-0025-0002	1	2000 Series Dual Mode Servo (25-2, Torque)		$ 33.99	$ 33.99	
3102-0001-0001	1	3102 Series Dual Mode Servo Programmer (1-1)		$ 9.99	$ 9.99	
Wench Pulley for servo motor (y)						
3410-0025-0112	1	3410 Series Servo-Mount Winch Pulley (25T Spline, Dual Spool, 112mm Circumference)		$ 6.99	$ 6.99	
Support for the linear acutator						
1102-0002-0016	8	1102 Series Flat Beam (2 Hole, 16mm Length) - 4 Packs		$ 1.99	$ 15.92	
				Total Price(Before Tax)	$ 345.76	

References

Arduino Forum. "Hook up Servo to a L298N." *Arduino Forum*, 30 Sept. 2015, https://forum.arduino.cc/t/hook-up-servo-to-a-l298n/336030.

Arduino Documentation. "Servo Library." *Arduino Docs*, https://docs.arduino.cc/libraries/servo/.

"Chronic Back Pain." *Health Policy Institute*, 13 Feb. 2019, hpi.georgetown.edu/backpain/.

Circuito.io. "Arduino Uno Pinout." *Circuito.io Blog*, https://www.circuito.io/blog/arduino-uno-pinout/#:~:text=Pins%200%2D13%20of%20the,10%2C11%20have%20PWM%20capability..

Dhanushka. *Controlling a Servo Motor with Arduino and L298N Motor Driver*. YouTube, 6 Aug. 2019, https://www.youtube.com/watch?v=kUHmYKWwuWs&t=1074s.

DroneBot Workshop. *Using L298N Motor Driver with Arduino - DC Motor Control*. YouTube, 16 Sept. 2020, https://www.youtube.com/watch?v=NV6YHZ2RAqc.

Lucas, Jacqueline W., et al. "NCHS Data Brief ■ No. 415 ■ July 2021." *National Center for Health Statistics*, July 2021, www.cdc.gov/nchs/data/databriefs/db415-h.pdf.

HowToMechatronics. "Arduino DC Motor Control Tutorial – L298N PWM H-Bridge." *HowToMechatronics*, https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/.

Arduino Code

Servo Motor Test:

```
#include <Servo.h>

Servo myservo; // Create a Servo object

int pos = 0;

void setup() {
    myservo.attach(9); // Attach the servo signal wire to pin 10
}

void loop() {
    for(pos = 0; pos <= 180; pos += 1){
      myservo.write(pos);
      delay(15);
    }
    for(pos = 180; pos >= 0; pos  -= 1 ){
      myservo.write(pos);
      delay(15);
    }
}
```

Linear Actuator and Horizontal Motors together:

```
// This is a test to see if I can get the linear acuator and horizontal assembly
#include <Wire.h>

// Motor control pin definitions for the X-axis
const int motorIn1 = 2;  // Motor direction control pin 1
const int motorIn2 = 3;  // Motor direction control pin 2
const int motorPWM = 11;  // Reassigned PWM pin for speed control

// Pin definitions for the linear actuator
const int actuatorPosPin = 4; // Positive output
const int actuatorNegPin = 5; // Negative output

// Time for X-axis motor to move to the next column (to be defined later)
```

```
int X_MOVE_TIME = 1000;  // Placeholder value (in milliseconds)

void setup() {

  // Initialize motor pins
  pinMode(motorIn1, OUTPUT);
  pinMode(motorIn2, OUTPUT);
  pinMode(motorPWM, OUTPUT);  // New PWM pin for motor speed control

  // Initialize the pins as outputs
  pinMode(actuatorPosPin, OUTPUT);
  pinMode(actuatorNegPin, OUTPUT);

  // Initialize Serial for debugging
  Serial.begin(9600);

  //Print to confirm the code is starting
  Serial.println("Linear actuator test started.");

}

void loop() {
  for(int i=0;i<10;i++){
    // Turn the motor forward for 2 seconds
    Serial.println("Horizontal Forward...");
    moveX(-1, 255);  // Forward direction at full speed (255)
    delay(1000);    // Wait for 2 seconds
    moveX(0, 0);    // Stop the motor
    delay(1000);     // Short delay before switching direction

    // Extending the linear actuator
    Serial.println("Extending actuator...");
    digitalWrite(actuatorPosPin, LOW);    // Set positive pin LOW
    digitalWrite(actuatorNegPin, HIGH);  // Set negative pin HIGH
    delay(5000);                          // Wait for 5 seconds

    // Retracting the linear actuator
    Serial.println("Retracting actuator...");
    digitalWrite(actuatorPosPin, HIGH);  // Set positive pin HIGH
    digitalWrite(actuatorNegPin, LOW);    // Set negative pin LOW
    delay(5000);                          // Wait for 5 seconds

    // Stop the actuator
    Serial.println("Stopping actuator...");
    digitalWrite(actuatorPosPin, LOW);    // Stop both pins
```

```
    digitalWrite(actuatorNegPin, LOW);
    delay(1000);                         // Wait 5 seconds before repeating

    // Turn the motor backward for 2 seconds
    Serial.println("Horizontal Backward...");
    moveX(1, 255); // Backward direction at full speed (255)
    delay(1000);    // Wait for 2 seconds
    moveX(0, 0);    // Stop the motor
    delay(1000);     // Short delay before repeating

    // Stop the actuator
    Serial.println("Stopping actuator...");
    digitalWrite(actuatorPosPin, LOW);   // Stop both pins
    digitalWrite(actuatorNegPin, LOW);
    delay(1000);                         // Wait 5 seconds before repeating
  }

}


// X-axis motor control function (direction: 1 = forward, -1 = backward, 0 =
stop)
void moveX(int direction, int speed) {
  if (direction == 1) {
    digitalWrite(motorIn1, HIGH);  // Move forward
    digitalWrite(motorIn2, LOW);
  } else if (direction == -1) {
    digitalWrite(motorIn1, LOW);   // Move backward
    digitalWrite(motorIn2, HIGH);
  } else {
    digitalWrite(motorIn1, LOW);   // Stop the motor
    digitalWrite(motorIn2, LOW);
  }
  analogWrite(motorPWM, speed);  // Control the motor speed (new pin 11)
}
```