ENGINEERING

Project Three – There's A Recyclable Among Us:

Design a System for Sorting and Recycling Containers

ENGINEER 1P13 – Integrated Cornerstone Design Projects

Tutorial 5

Team Tues-33

Nolan Cross (crossn3)

Zareen Kabir (kabirz)

Fondson Lu (luh57)

Luigi Quattrociocchi (quattrl)

Sude Sayyan (sayyans)

Submitted: March 7, 2021

Table of Contents

Academic Integrity Statement

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Luigi Quattrociocchi 400318027

X

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Sude Sayyan 400316246

X

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Zareen Kabir 400315411

X

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Nolan Cross 400328232

X 

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Fondson Lu 400322526

X 

Executive Summary

Approximately 30% of items placed in recycling bins are not recyclable, because of improper sorting, material hybrid, etc. [1, pg 3]. Although recycling station have sorting facilities, with the huge influx of garbage in recent years, it is difficult for people to sort them by themselves. The purpose of this project, then, was to improve the current recycling process by automating a system with sorting capabilities [1, pg. 3].

The modelling sub-team's members were tasked with designing, modelling, and simulating a device which deposits containers into a recycling bin (i.e., a recycling hopper) [1, pg 8]. This device would be mounted to the top of a terrestrial drone (called the Q-bot) in the Q-Labs environment [2]. It would be responsible for holding the containers during the bot's movement as well as dumping the containers into the recycling bin once the bot is in the correct position. The design for the device would be proposed and revised by the sub-team, then modelled and simulated in Autodesk Inventor [3]. The agreed upon design utilized a rotary actuator (rather than a linear actuator) which was connected to the hopper through a gear train and a gear-rack, which converted the rotary motion of the actuator to vertical linear motion as well as increased the torque via gear reduction and a rack and pinion mechanism [1, pg. 12] [4]. After modelling the parts and assembling the mechanism in Inventor, the assembly was dynamically simulated by imposing rotational motion on the actuator [Fig. 1-5] [Fig. 6]. Through the simulation it was shown that our mechanism was capable of effectively dumping the assorted containers out of the hopper [Fig. 7-15].

The computing sub team was given the task to write a computer program in Python that controls the movement of robotic devices (via Q-Labs Library Documentation) for transferring containers to their appropriate bins in a recycling station [1, pg. 16] [5]. To perform this objective, we included a set of functions to be called in the main function to allow the robotic devices to execute its tasks in the correct order [Fig. 30]. Our first function was written to dispense the containers onto the servo table [Fig. 26]. Then, our load container function was responsible for loading the containers onto the hopper using the Q-Arm [2] [Fig. 27]. Next, our transfer container function was where we incorporated an ultrasonic sensor to sense the correct bin that the container(s) belonged in [Fig. 28]. Our deposit container and return home functions were used to drop off the containers into the designated bins and return the robotic devices to their home position [Fig. 29]. This process while loop until the user kills the code. The main objective we struggled with throughout this project was incorporating a sensor into our code. One

thing that helped us was writing pseudocode for each function before starting to hard code. From there, the Python code was written.

Our project deliverables (the recycling hopper mechanism paired with the python-controlled robotic devices in the Q-Labs environment) successfully demonstrate the usefulness of this kind of automated recycling sorting system and the practical applications of such a system.

Project Schedule

Preliminary Gantt Chart

PROJECT START 12-Jan

TASK	START	END
Milestone 0 - Determine and document administrative responsibilities for each team member.	12-Jan	18-Jan
Milestone 1 - Complete a series of design exercises to frame the given problem.	12-Jan	18-Jan
Milestone 2 - Conceptualize both how the containers will be transferred and how the containers will be dropped off. Evaluate and propose refinements.	19-Jan	25-Jan
Milestone 3 - Create preliminary models of design in Autodesk Inventor based on detailed sketches. Create preliminary program tasks.	26-Jan	01-Feb
Milestone 4 - Submit a portion of your design for TA feedback.	02-Feb	08-Feb
Dedicated Project Time - Student teams work towards finalizing their design.	09-Feb	22-Feb
Project Demonstration and Interview - Student teams demonstrate their design to an IAI. Students answer questions of their design in an individual interview.	23-Feb	01-Mar
Final Deliverables - Design Project Report. Independent materials research summary. Learning Portfolio Entry. Self- and Peer-evaluation.	02-Mar	07-Mar

Final Gantt Chart

PROJECT START 12-Jan

TASK	START	END
Milestone 0 - Determine and document administrative responsibilities for each team member.	12-Jan	13-Jan
Milestone 1 - Complete a series of design exercises to frame the given problem.	12-Jan	13-Jan
Milestone 2 - Conceptualize both how the containers will be transferred and how the containers will be dropped off. Evaluate and propose refinements.	19-Jan	20-Jan
Milestone 3 - Create preliminary models of design in Autodesk Inventor based on detailed sketches. Create preliminary program tasks.	26-Jan	27-Jan
Milestone 4 - Submit a portion of your design for TA feedback.	02-Feb	02-Feb
Dedicated Project Time - Student teams work towards finalizing their design.	09-Feb	22-Feb
Project Demonstration and Interview - Student teams demonstrate their design to an IAI. Students answer questions of their design in an	23-Feb	23-Feb
Final Deliverables - Design Project Report. Independent materials research summary. Learning Portfolio Entry. Self-and Peer-evaluation.	02-Mar	07-Mar

Logbook of Additional Meetings and Discussions

Jan. 31, 2021: 3 hours – *Working on model in CAD* (Luigi, Nolan, Fondson)

Feb. 1, 2021: 5 hours – *Started writing out code in Python* (Zareen, Sude)

Feb. 2, 2021: - 1 hour – *Write out code for the detect_container function* (Zareen, Sude)

Feb. 4, 2021: 2 hours – *Figure out how to incorporate sensor + switch sensor type* (Zareen, Sude)

Feb. 11, 2021: 2 hours – *Write out code for the dispense_container function* (Zareen, Sude)

Feb. 16, 2021: 5 hours – *Fix errors for the transfer_container function* (Zareen, Sude)

Feb. 17, 2021: 1.5 hours – *Fix errors for the load_container function* (Zareen, Sude)

Feb. 19, 2021: 1 hour – *Work on main function* (Zareen, Sude)

Feb. 20, 2021: 1.5 hours – *Finalize Python Code and Test Run Code* (Zareen, Sude)

Feb. 21, 2021: 2 hours – *Finishing model and simulation* (Luigi, Nolan, Fondson)

Feb. 22, 2021: 1 hour – *Final touches for model + submission* (Luigi, Fondson)

Feb. 23, 2021: 2:00PM – 2:30PM – *Final Project Interview* (Luigi, Nolan, Fondson, Zareen, Sude)

Mar. 5, 2021: 2 hours – *Worked on design report* (Zareen, Sude)

Mar. 5, 2021: 4 hours – *Worked on design report* (Luigi, Nolan, Fondson)

Mar. 7, 2021: 1 hour – *Worked on design report* (Luigi, Nolan, Fondson, Zareen, Sude)

*CAD model for modelling team was worked on individually by each sub-team member

Scheduled Weekly Meetings

Jan. 19 Agenda and Meeting Minutes

https://luigi.quattrociocchi.net/portfolio-files/P3%20Agenda%20and%20Meeting%20Minutes%20-%20Jan%2019.pdf

Jan. 26 Agenda and Meeting Minutes

https://luigi.quattrociocchi.net/portfolio-files/P3%20Agenda%20and%20Meeting%20Minutes%20-%20Jan%2026.pdf

Feb. 09 Agenda and Meeting Minutes

https://luigi.quattrociocchi.net/portfolio-files/P3%20Agenda%20and%20Meeting%20Minutes%20-%20Feb%2009.pdf

Design Studio Worksheets

Milestone 0

https://luigi.quattrociocchi.net/portfolio-files/Tues-33_P3_Milestone0.pdf

Milestone 1

https://luigi.quattrociocchi.net/portfolio-files/Tues-33_P3_Milestone1.pdf

Milestone 2

https://luigi.quattrociocchi.net/portfolio-files/Tues-33_P3_Milestone2.pdf

Milestone 3

https://luigi.quattrociocchi.net/portfolio-files/Tues-33_P3_Milestone3.pdf

Milestone 4

https://luigi.quattrociocchi.net/portfolio-files/Tues-33_P3_Milestone4.pdf

List of Sources

[1] – “There’s a Recyclable Among Us,” *P3 Project Module*, pp. 3–44, class notes for ENGINEER 1P13, Department of Engineering, McMaster University, Winter, 2021.

[2] – “Quanser Interactive Labs’’ Quanser Consulting Inc, Markham, ON, 2021. (https://www.quanser.com/)

[3] – “Autodesk Inventor Professional 2021.” Autodesk, San Rafael, CA, 2021. (https://www.autodesk.com/)

[4] – “Mechanisms: Types and Application,” class notes for ENGINEER 1P13, Department of Engineering, McMaster University, Winter, 2021.

[5] “P3 Python Library Documentation,” class notes for ENGINEER 1P13, Department of Engineering, McMaster University, Winter, 2021.

Additional Citations:

A. Help, “Insert rolling joints,” *Insert rolling joints* | *Inventor* | *Autodesk Knowledge Network*, February 13, 2014. [Online]. Available: https://knowledge.autodesk.com/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2014/ENU/Inventor/files/GUID-590E099F-F51A-4876-9495-AFB7F7334F5F-htm.html. [Accessed: March 7, 2021].

“Color sensors | SICK.” https://www.sick.com/ca/en/products-by-tasks/monitoring-and-controlling/quality/color-sensors/c/g113666. [Accessed: January 17, 2021].

“Custom Ultrasonic Sensors Detect Wide Range of Materials,” FierceElectronics, December 11, 2017. [Online]. Available: https://www.fierceelectronics.com/components/custom-ultrasonic-sensors-detect-wide-range-materials#:~:text=The%20ultrasonic%20sensors%20can%20detect,as%20mesh%20trays%20or%20springs. [Accessed: January 18, 2021].

“Detection based on ‘Ultrasonic Waves’ What is an ultrasonic / level sensor?,” KEYENCE [Online]. Available: https://www.keyence.ca/ss/products/sensor/sensorbasics/ultrasonic/info/. [Accessed: January 18, 2021].

Ebrahim6000, *YouTube*, 28-Jul-2017. [Online]. Available: https://www.youtube.com/watch?v=Xhe8lgi4-_4. [Accessed: March 7, 2021].

E. Hassan, "Inventor Assembly Design 202 - CAD Tutorial," MacVideo [Online]. Available: https://www.macvideo.ca/media/Inventor+Assembly+Design+202+-+CAD+Tutorial/1_j2bumroo?st=600&ed=1176. [Accessed: March 7, 2021].

"Graphics Lab #6 Slides," class notes for ENGINEER 1P13B, Department of Engineering, McMaster University, Winter, 2021.

"Graphics Lab #7 Slides," class notes for ENGINEER 1P13B, Department of Engineering, McMaster University, Winter, 2021.

"Hall Effect Sensor and How Magnets Make It Works," Basic Electronics Tutorials, February 9, 2021. [Online]. Available: https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html. [Accessed: January 18, 2021].

"Infrared sensors - Sensor Technology," Metropolia Confluence [Online]. Available: https://wiki.metropolia.fi/display/sensor/Infrared+sensors#:~:text=Active%20infrared%20sensors%20are%20the,photodiode%2C%20phototransistor%20or%20photoelectric%20cells. [Accessed: January 18, 2021].

"Light Dependent Resistor LDR, Photoresistor » Electronics Notes." https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php. [Accessed: January 17, 2021].

"Seven uses for photoelectric sensors." https://www.rs-online.com/designspark/seven-uses-for-photoelectric-sensors. [Accessed: January 17, 2021].

Appendix A – Screenshots of Solid Model

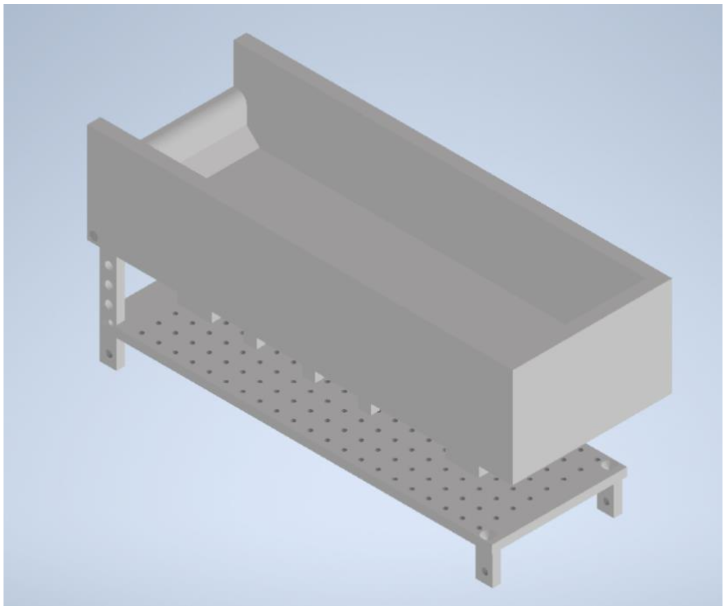

Figure 1: Final model with hopper

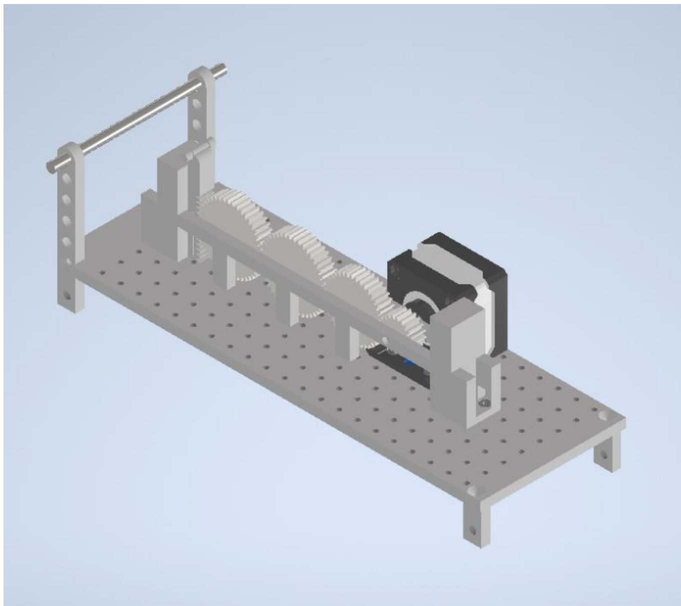

Figure 2: Final model gear train

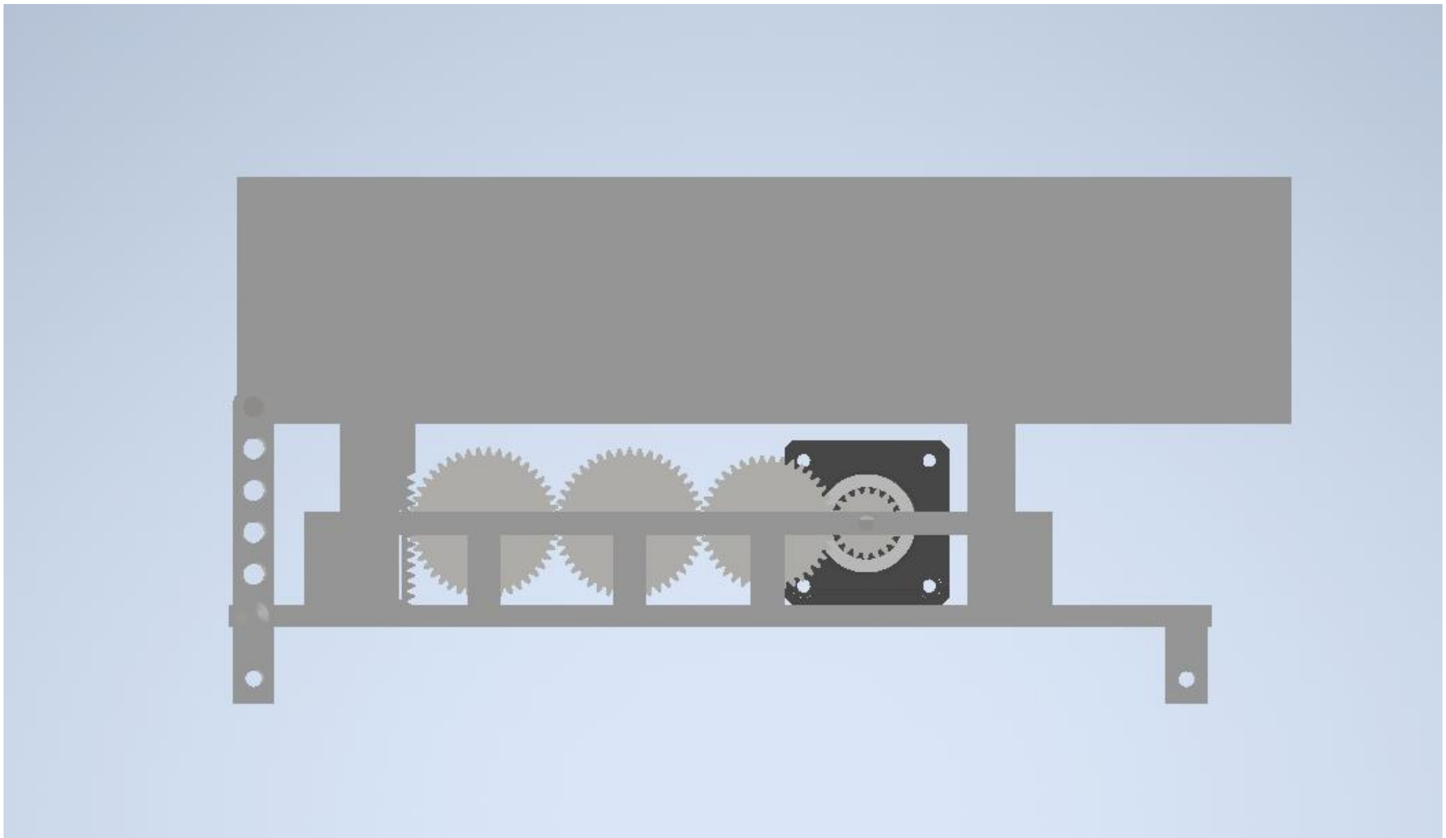

Figure 3: Final model side view

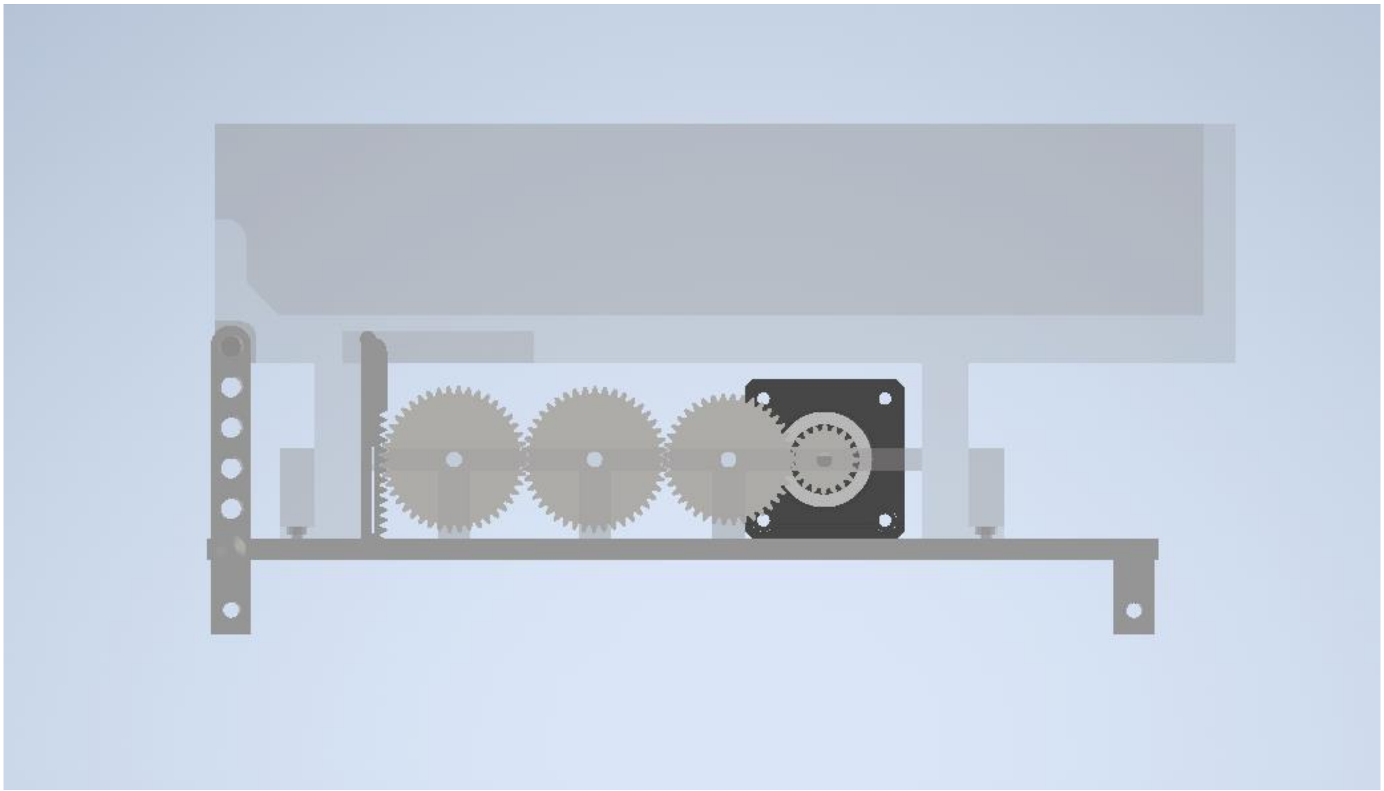

Figure 4: Final model side view (transluscent components)

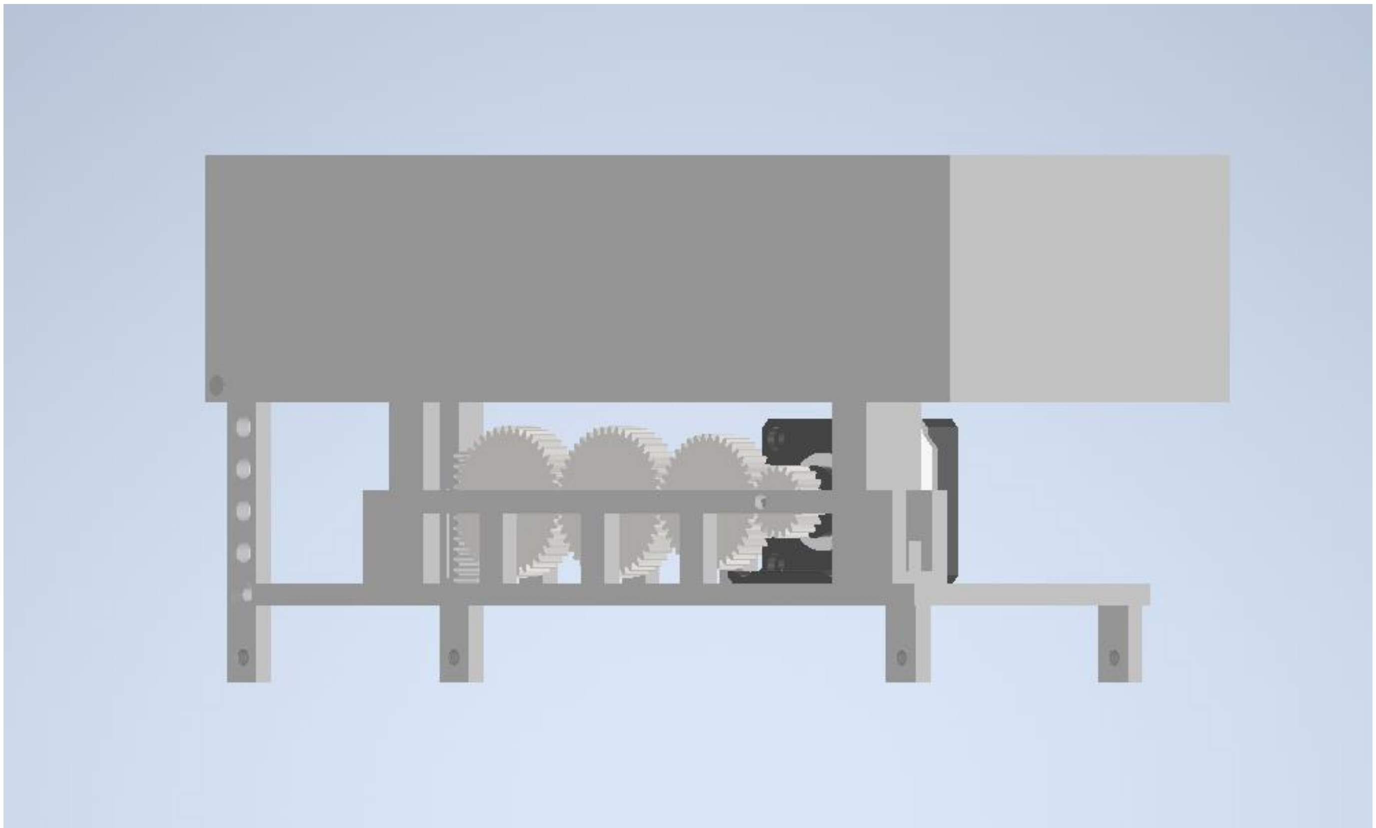

Figure 5: Final model corner view

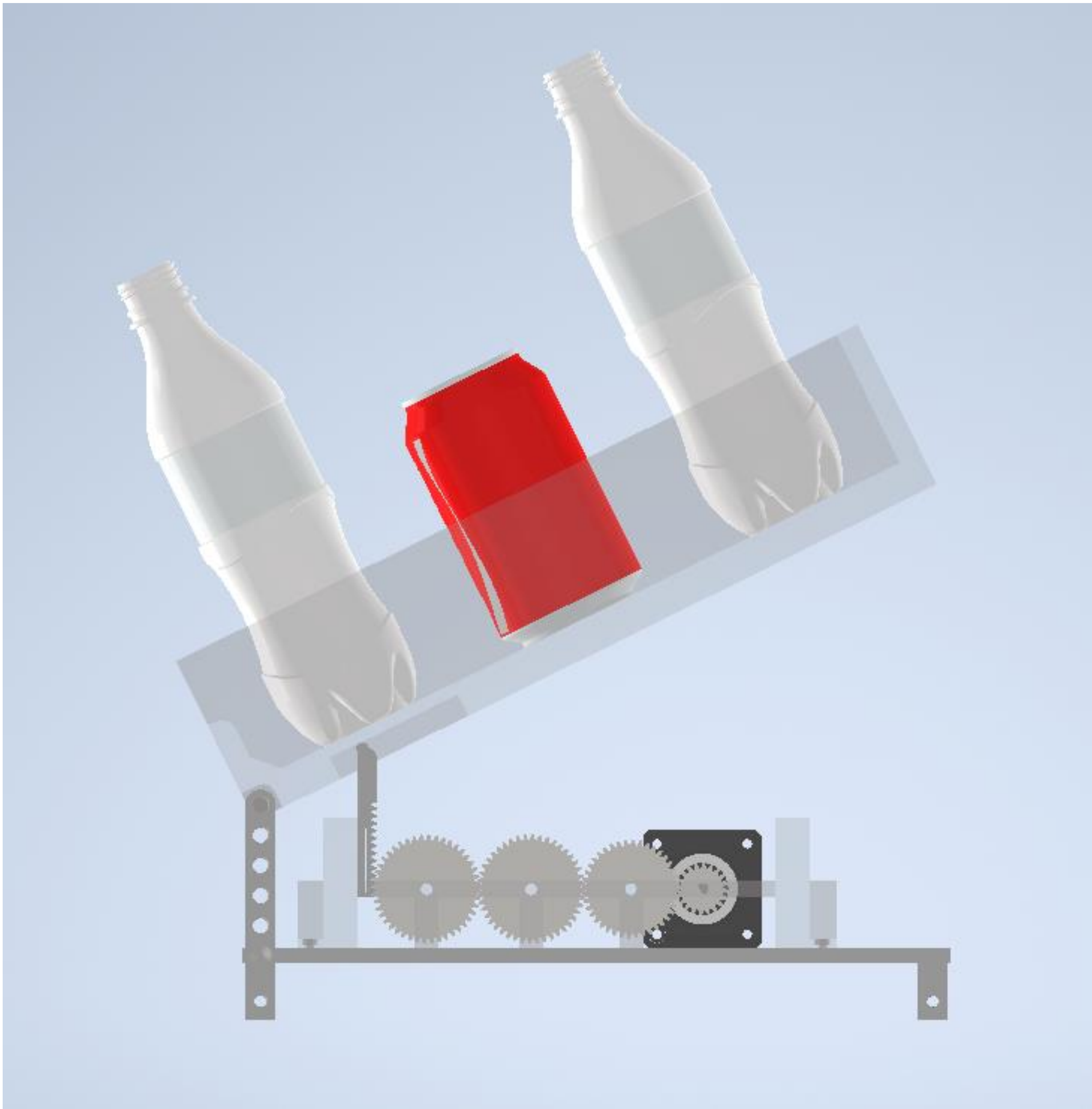

Figure 6: Final model in extended position with bottles

Figures 7 – 15: Motion simulation joint position output graphs

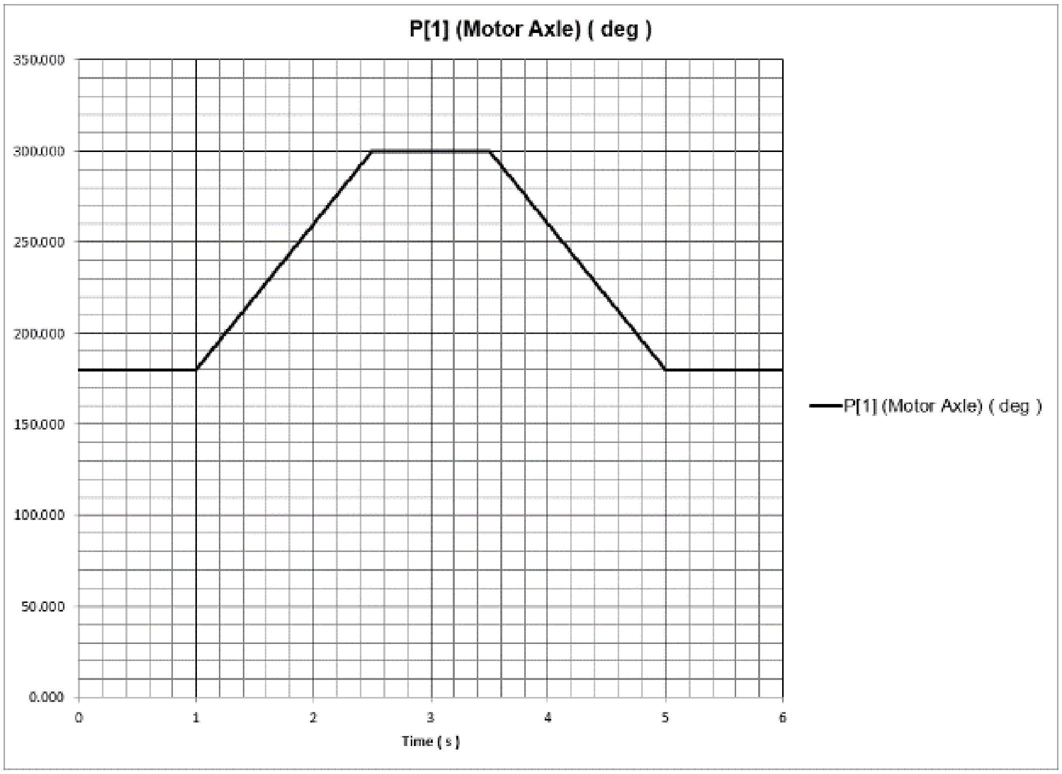

Figure 7

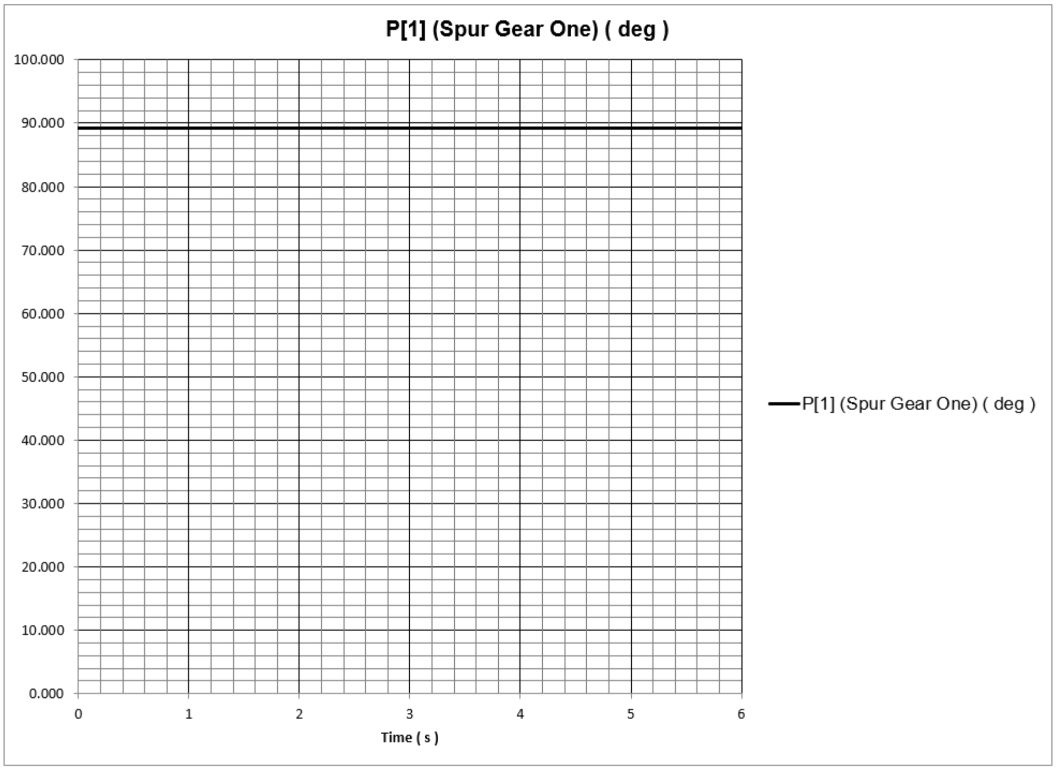

Figure 8

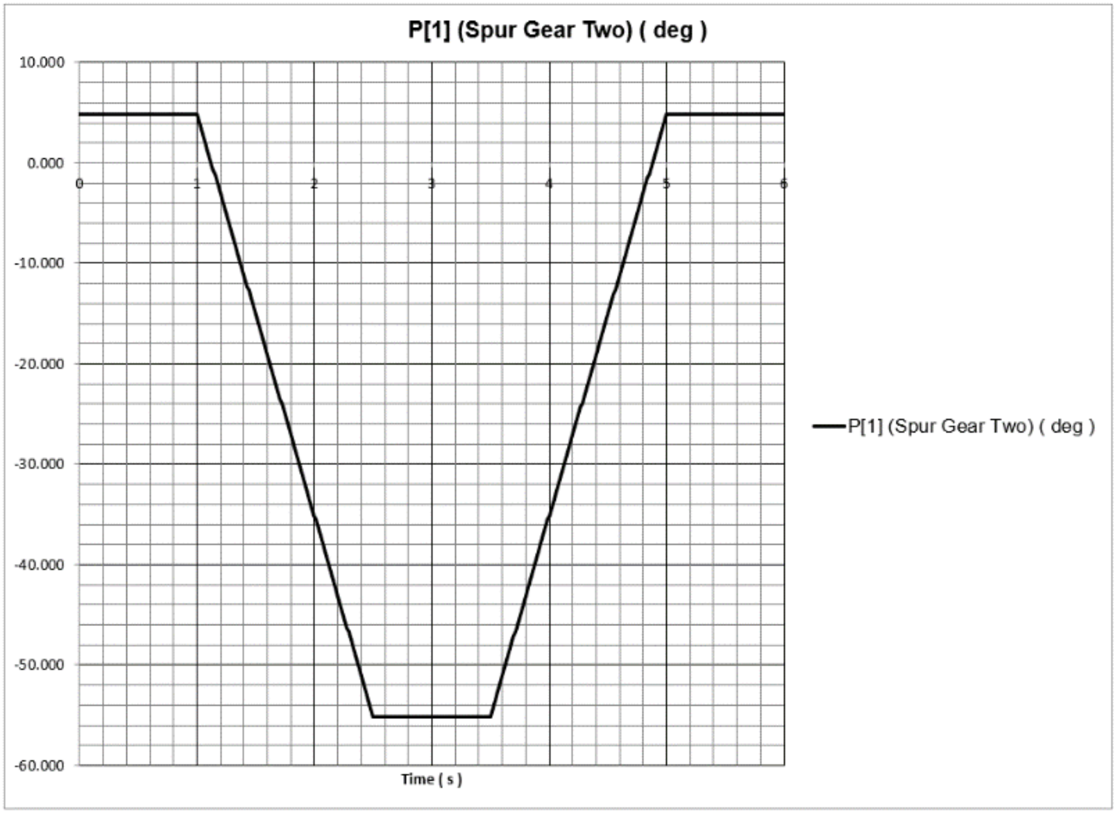

Figure 9

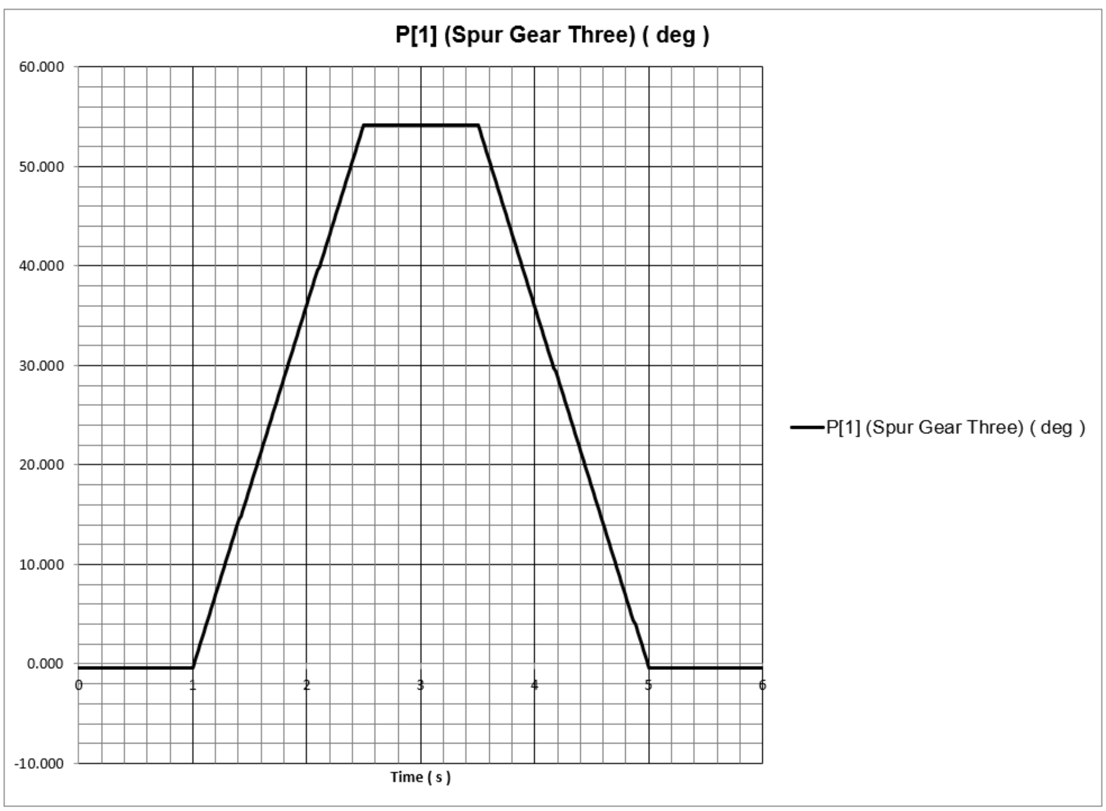

Figure 10

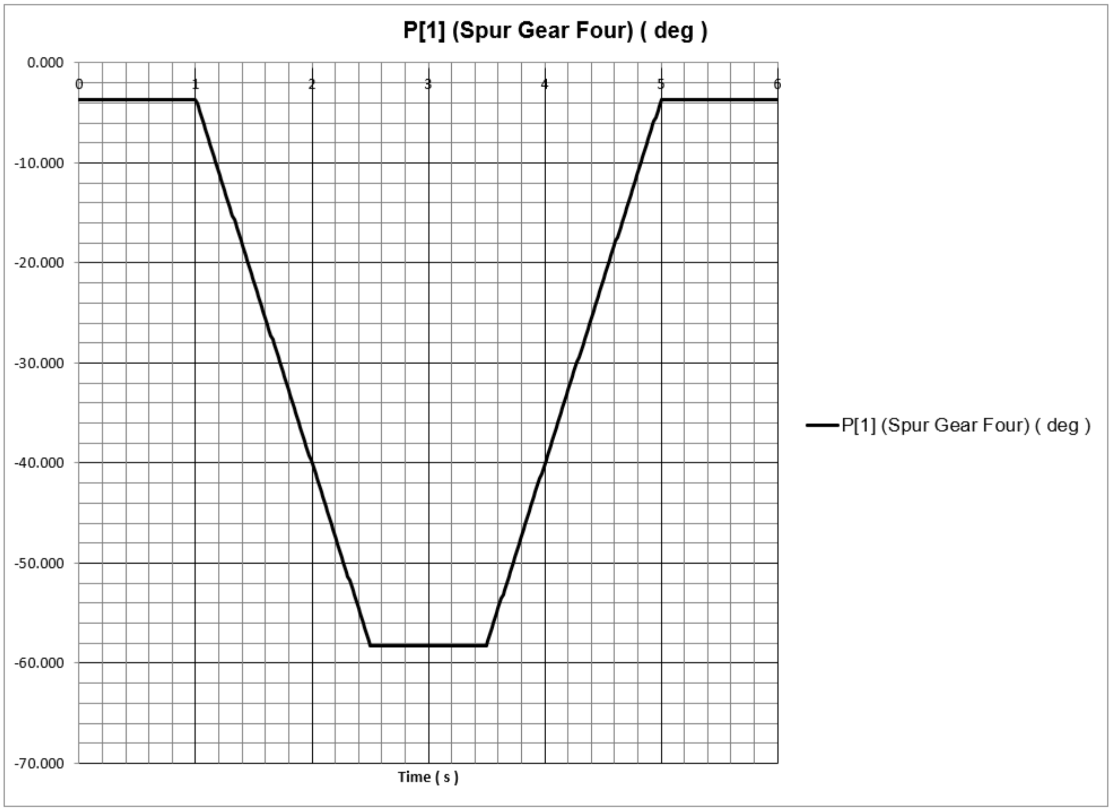

Figure 11

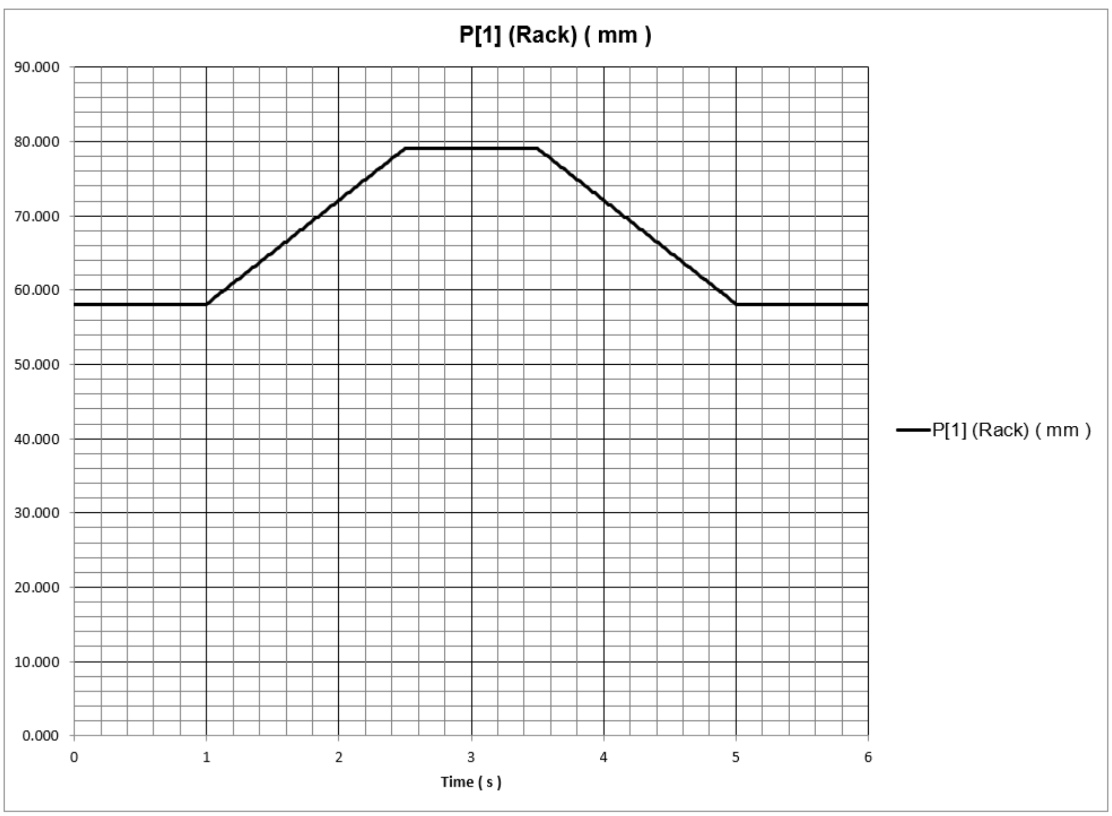

Figure 12

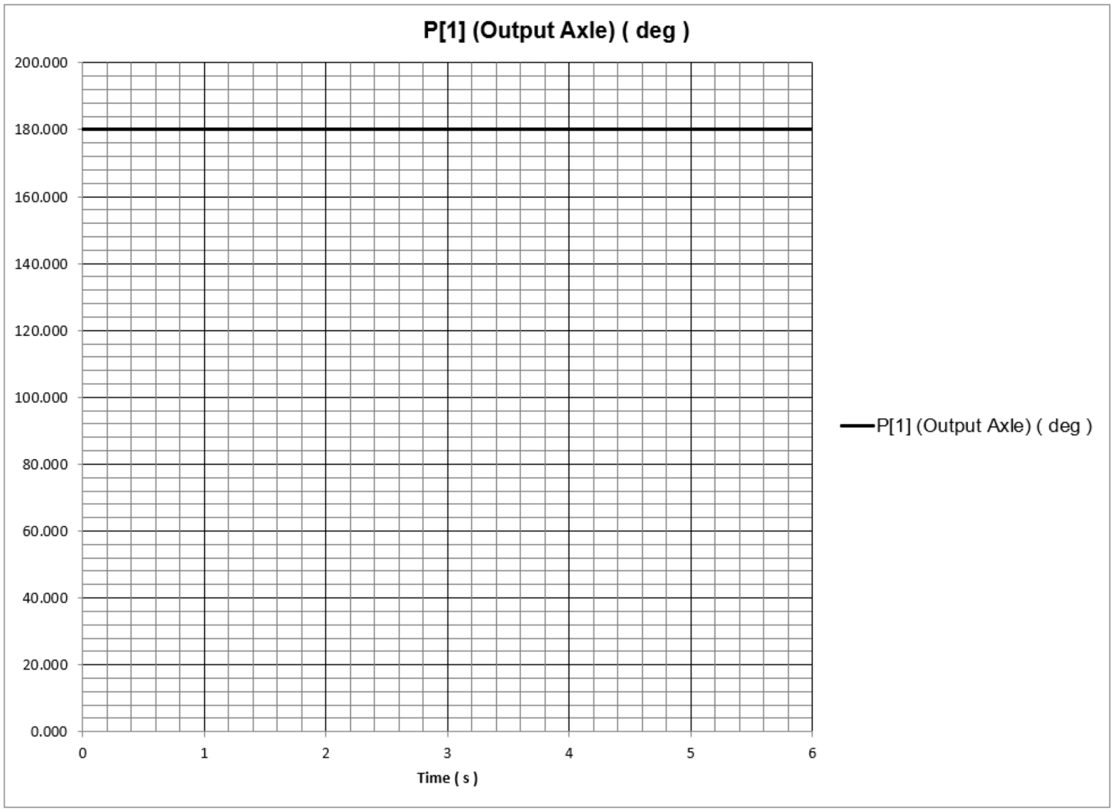

Figure 13

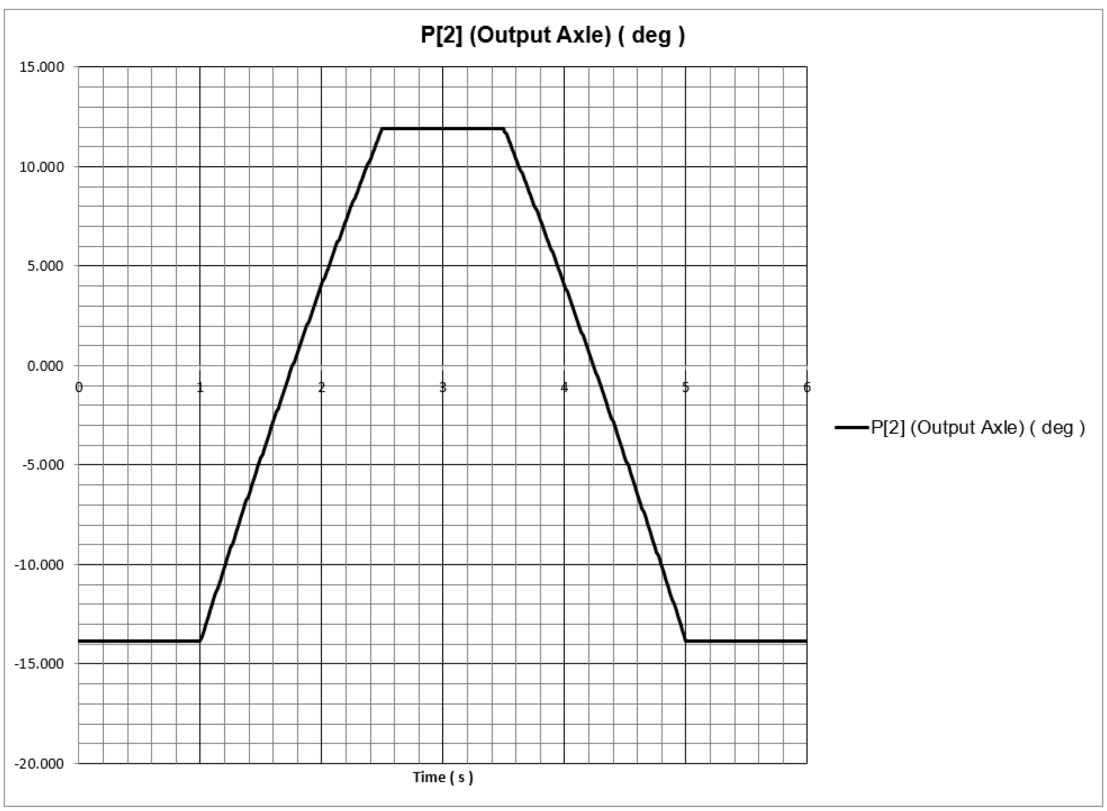

Figure 14

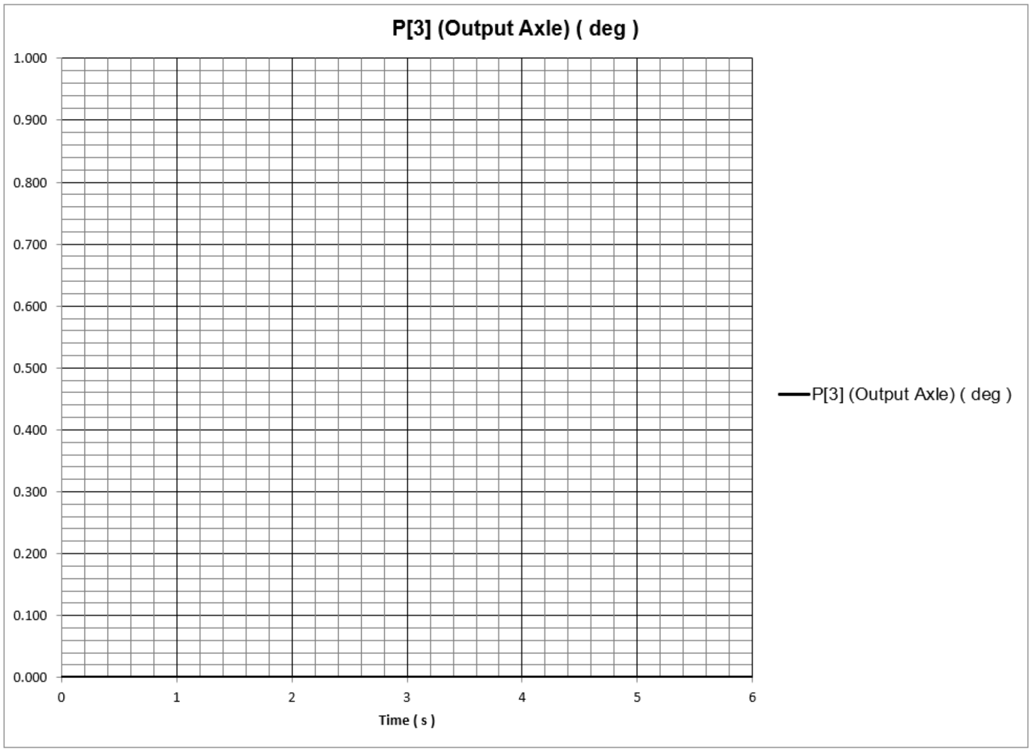
P[3] (Output Axle) (deg)
1.000
0.900
0.800
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.000
0
1
2
3
4
5
6
Time (s)
P[3] (Output Axle) (deg)

Figure 15

Appendix B – Fully-dimensioned Engineering Drawings

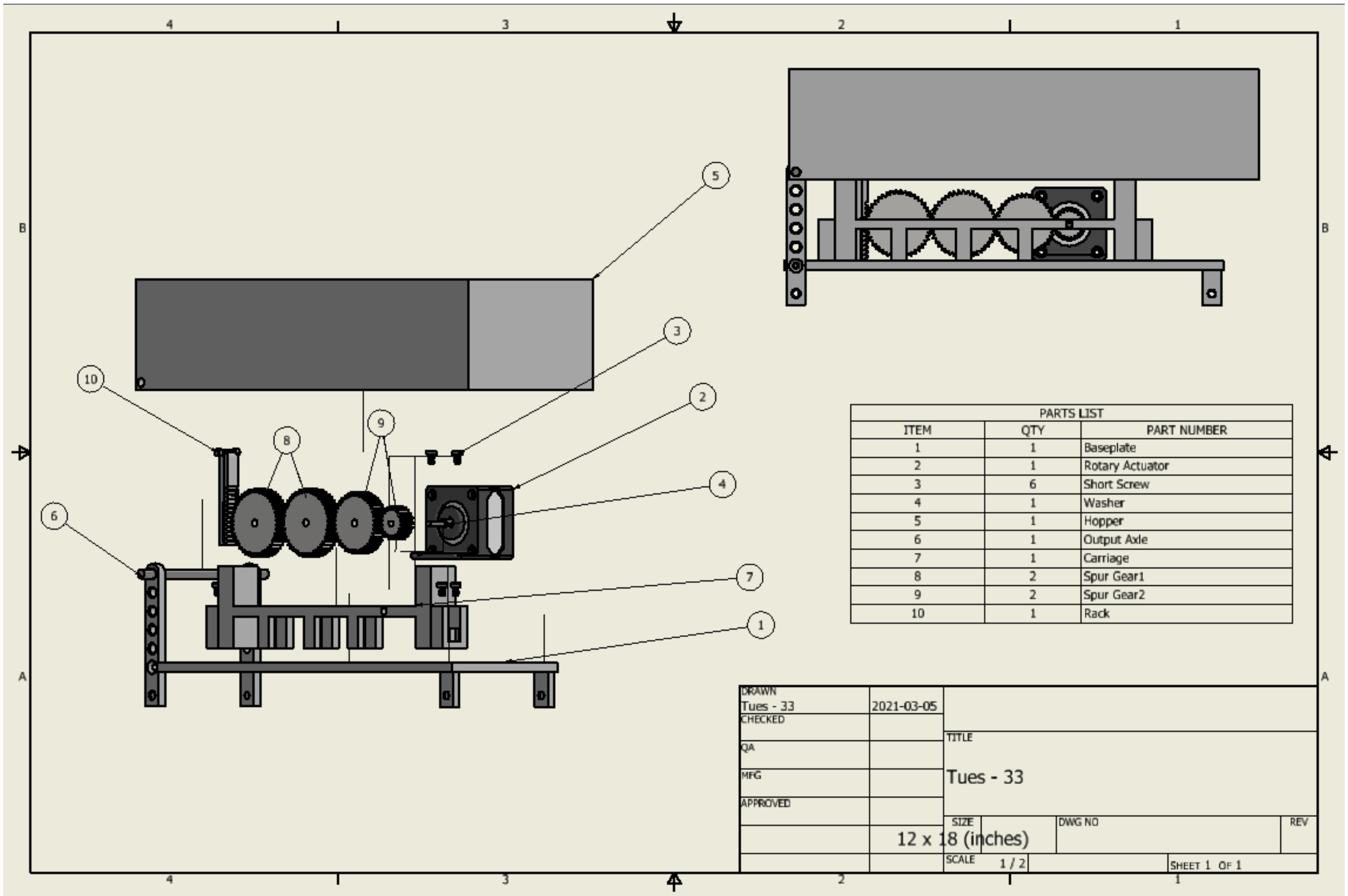

Figure 16: Exploded assembly of model

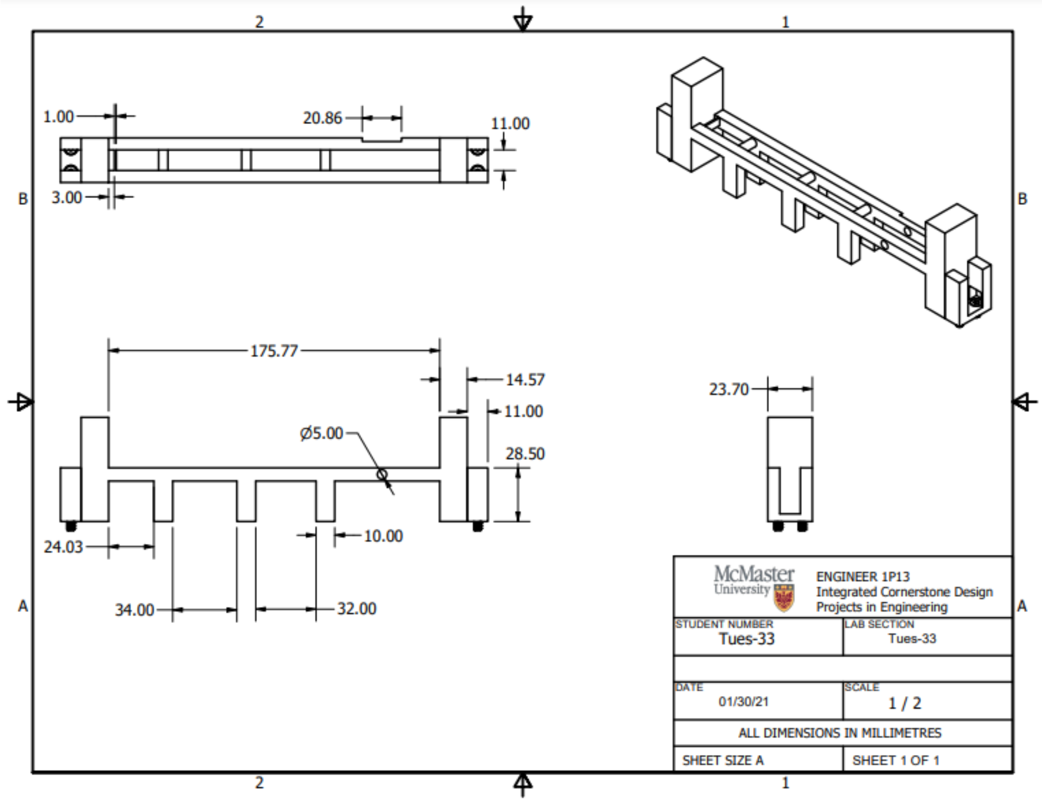

Figure 17: Gear train carriage annotated drawing

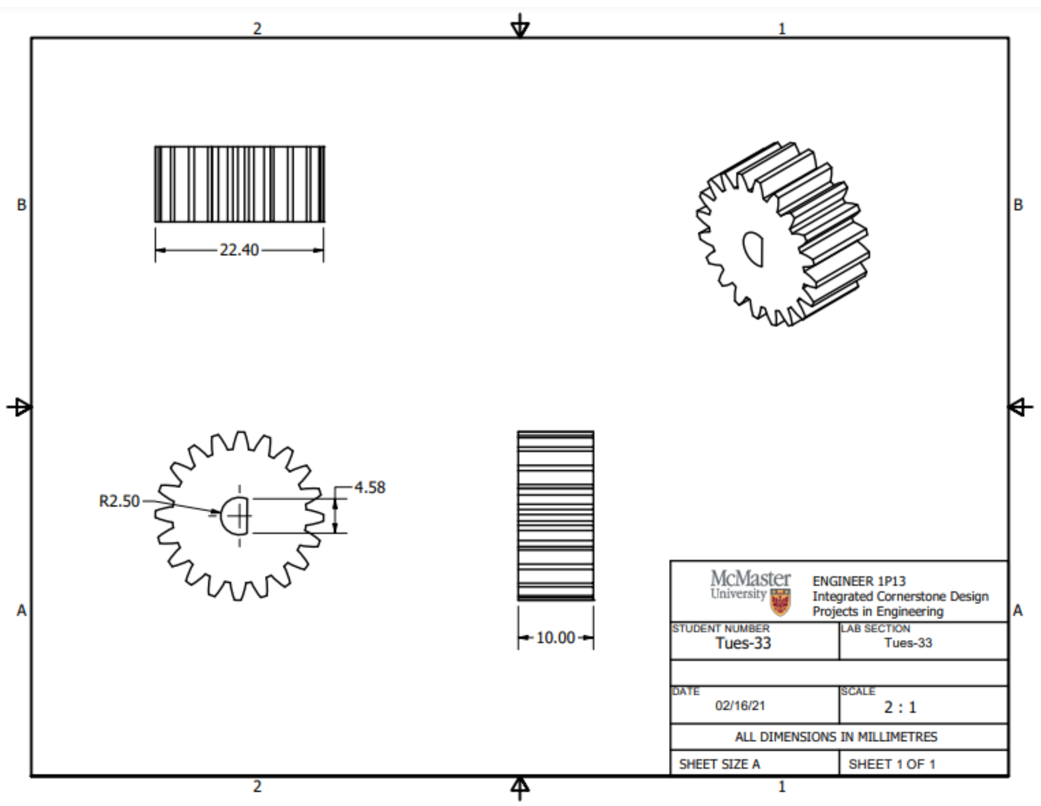

Figure 18: Spur gear one annotated drawing

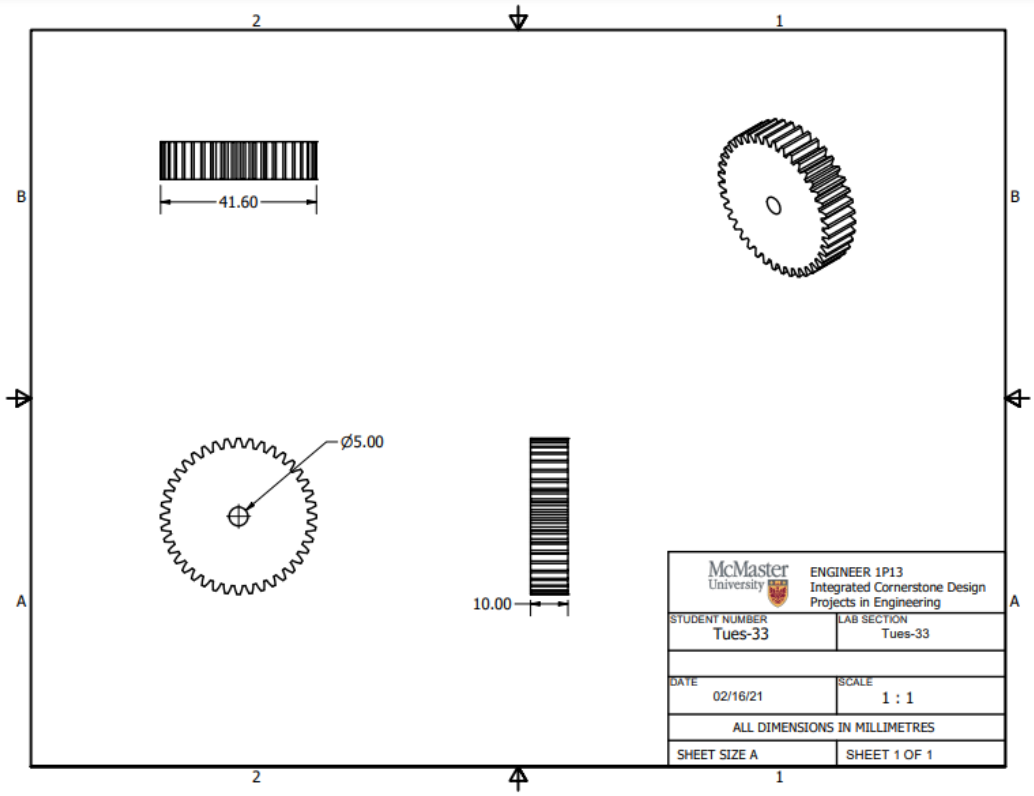

Figure 19: Spur gear two annotated drawing

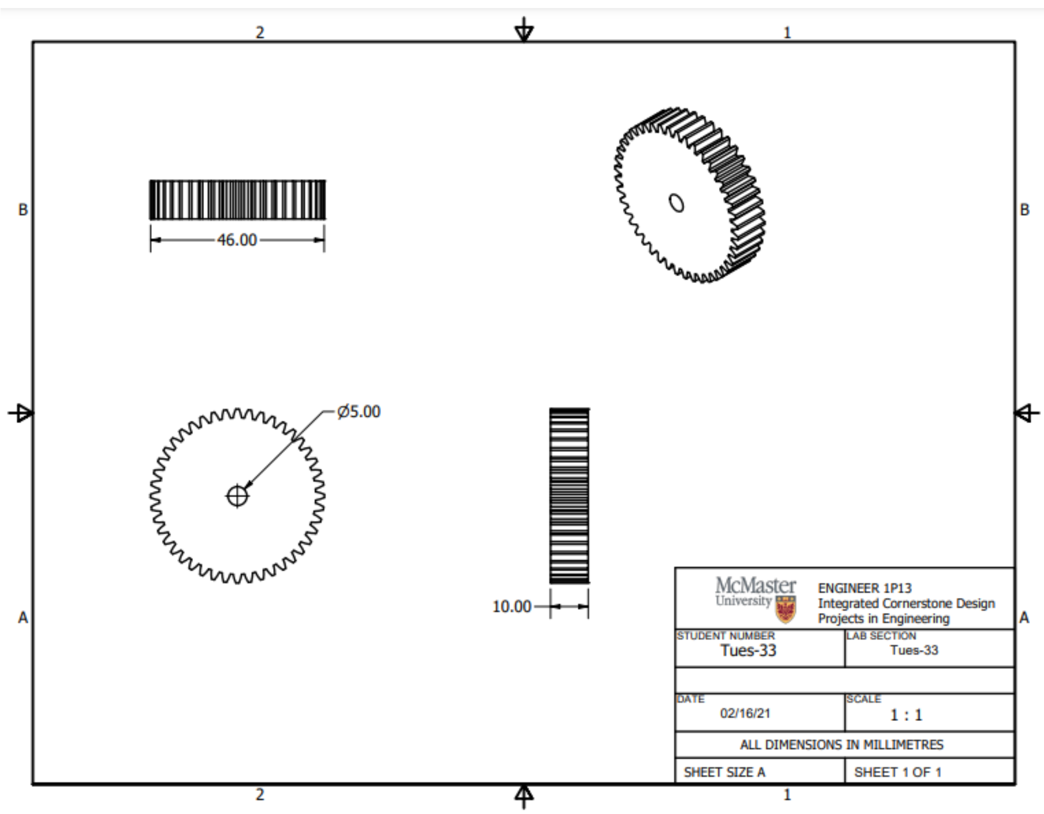

Figure 20: Spur gears three and four annotated drawing

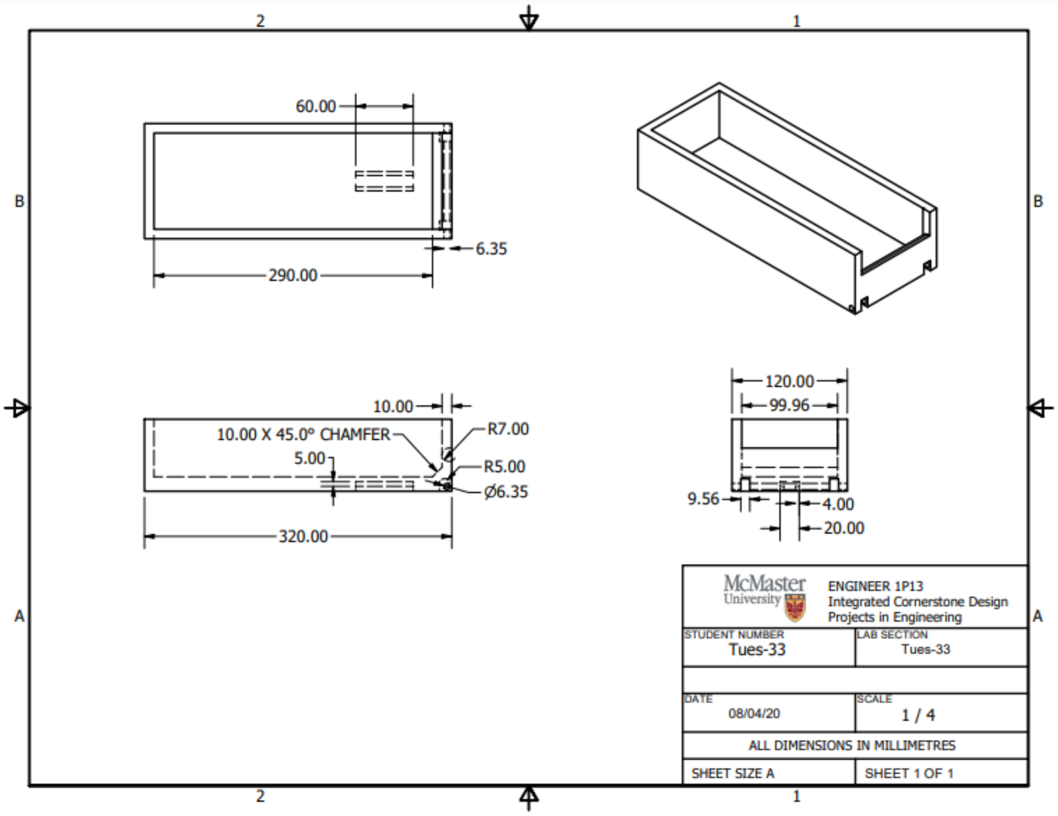

Figure 21: Modified recycling hopper annotated drawing

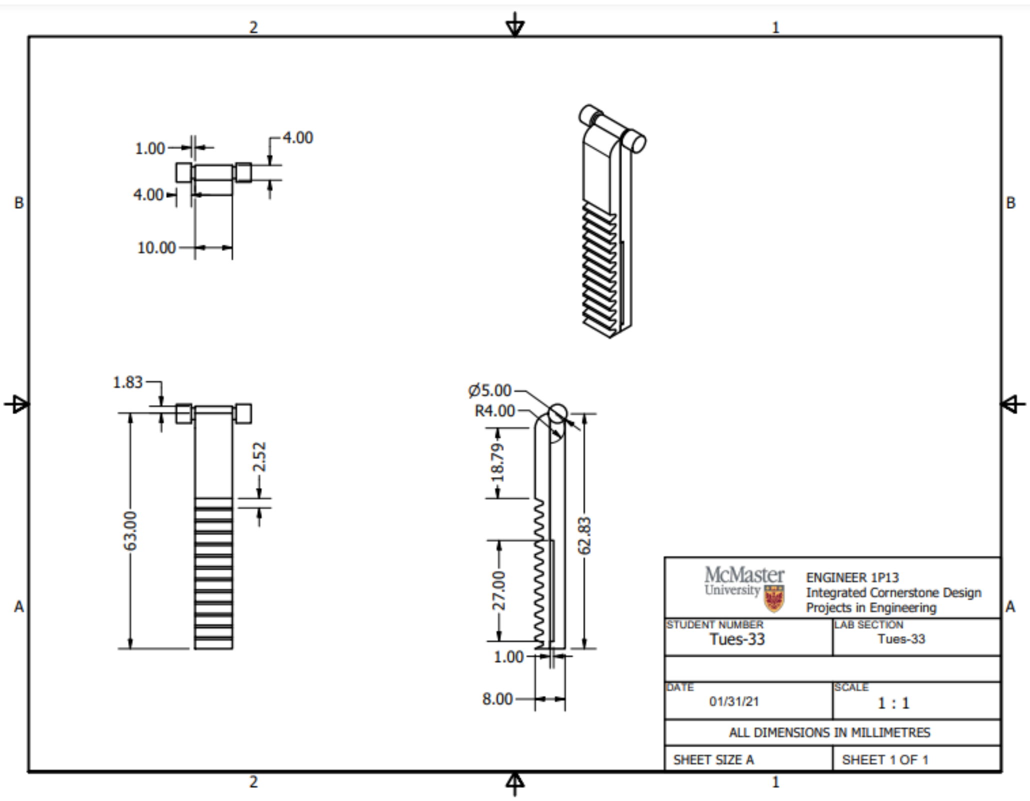

Figure 22: Gear rack annotated drawing

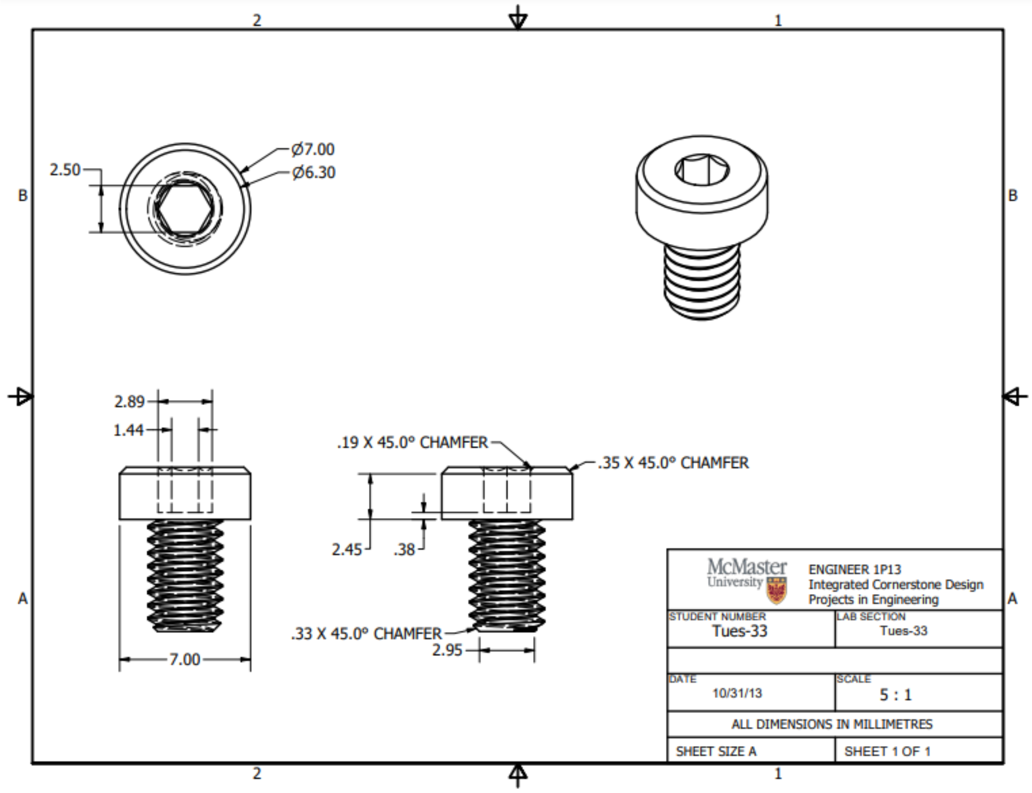

Figure 23: Short screw annotated drawing

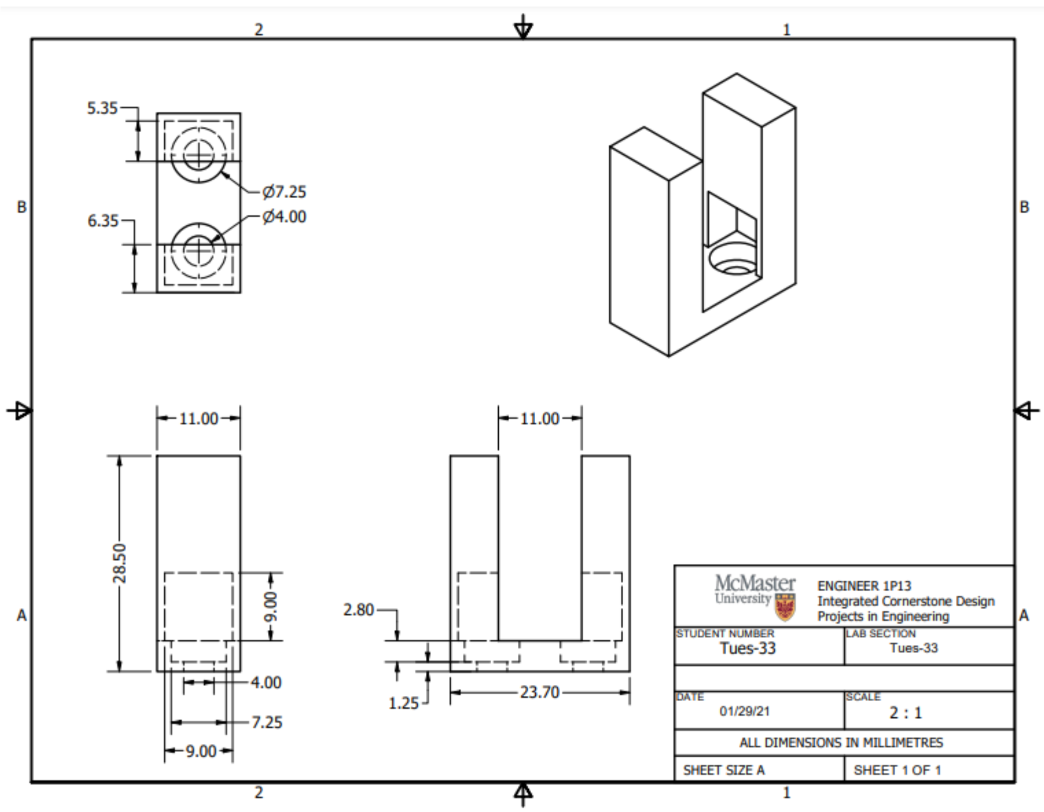

Figure 24: Carriage stand annotated drawing

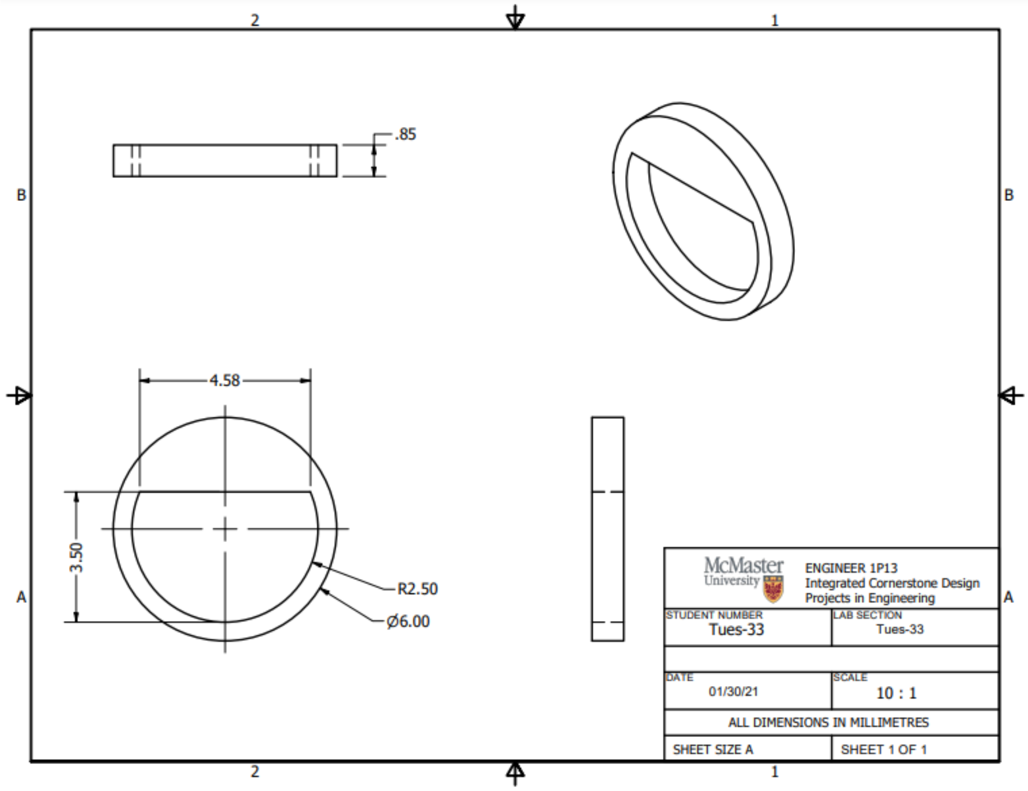

Figure 25: Motor axle alignment washer annotated drawing

Appendix C – Screenshots of Computer Program

```
#Zareen and Sude
#this function dispenses the bottles onto the servo table
def dispense_container():
    bottle = random.randint(1,6)                          #a random container through 1-6 is chosen
    c_type = my_table.container_properties(bottle)        #the container properties are determined
    my_table.dispense_container()                         #and dispensed (as well as returned)
    return c_type
```

Figure 26: dispense container() function

```
#Zareen and Sude
#this function loads the container onto the Q-bot using the Q-arm
def load_container(counter):
    net_weight = 0
    p_bin_num = 0
    bottle_num = 0
    if counter == 0:
        bot.forward_time(9.9)
    else:
        bot.forward_time(0.7)

    for bottle_num in range(1,4):                          #for loop that takes the container properties and
        c_properties = dispense_container()               #splits them into material, weight, and bin number
        print(c_properties)
        material, weight, bin_num = c_properties[0], c_properties[1], c_properties[2]
        net_weight += weight                              #calculates the total weight
        if bottle_num == 1:
            p_bin_num = bin_num                           #stores the first bin number, (meets the first constraint)
            pass
            print("Bottle number 1!")


        if p_bin_num != bin_num:                          #and if the bin number doesn't match the previous one,
            print("Different bin!")                       #the q-arm won't pick it up (meets the second constraint)
            break
        else:
            if net_weight > 90:                           #net weight can't surpass 90 grams (meets the last constraint)
                print("Too much weight")
                break

          else:
            arm.move_arm(0.644, 0.0, 0.2733)              #if it meets all of the previous conditions, the container is
            arm.control_gripper(45)                       #picked up and placed on the hopper
            arm.move_arm(0.4064, 0.0, 0.4826)
            if bottle_num == 1:
                arm.move_arm(-0.0956, -0.3567, 0.4107)
            elif bottle_num == 2:
                arm.move_arm(0.0, -0.3926, 0.3774)
            else:
                arm.move_arm(0.1016, -0.3792, 0.3774)
            arm.control_gripper(-25)
            arm.move_arm(0.0, -0.2874, 0.77)
            arm.home()
  return p_bin_num                                        #returns the first bin number
```

Figure 27: load container() function

```
#Zareen and Sude
#this function transfers containers using bin_num while approaching the appropriate bin whilst following the trajectory
#of a line
def transfer_container(bin_num):
    bot.rotate(180)
    time.sleep(2)
    MIN_READING = 0.1                                  #defined variable for sensor reading of a bin proximity
    bot.activate_ultrasonic_sensor()
    lost_line = 0
    reading = bot.read_ultrasonic_sensor(bin_num)      #bot activates and reads the sensor for the correct bin
    while reading > MIN_READING:                       #while the designated bin is out of reach
                                                       #keep reading until it is in reach
        while lost_line < 2:                           #checks if the bot is on the line, and makes sure it stays
            lost_line, velocity = bot.follow_line(0.1) #on the line
            bot.forward_velocity(velocity)
            reading = bot.read_ultrasonic_sensor(bin_num)
            if reading <= MIN_READING:                 #if the bot has approached the correct bin, stop the bot
                bot.stop()                             #and prepare for dumping
                bot.deactivate_ultrasonic_sensor()
                print("Coast is clear!!")
                return
            print(bin_num , reading)
```

Figure 28: transfer container() function

```
#Zareen and Sude
#this function deposits the container into the designated bin
def deposit_container():
    time.sleep(2)
    bot.activate_actuator()                            #activates the actuator to deposit the containers
    bot.dump()                                         #dumps the container

#Zareen and Sude
#this function returns both the Q-arm and Q-bot to their appropriate home positions
def return_home():
    arm.home()
    lost_line = 0
    while lost_line < 2:                               #follows the trajectory of the line around the loop
        lost_line, velocity = bot.follow_line(0.1)
        bot.forward_velocity(velocity)
        print(lost_line)
    bot.stop()                                         #stops at home position :)
```

Figure 29: deposit container() and return home() function

```
#Zareen and Sude
#this function is the main function (all the other functions are called within it)
def main():
    counter = 0                                        #the variable counts how many times the simulation runs
    while True:
        bin_num = load_container(counter)              #counter is sent into load_container
        transfer_container(bin_num)
        deposit_container()
        my_table.rotate_table_angle(45)                #servo table rotates 45 degrees after each iteration of the
        return_home()                                  #program

        counter += 1                                   #1 is added to the counter so the loop occurs another time


main()
```

Figure 30: main() function