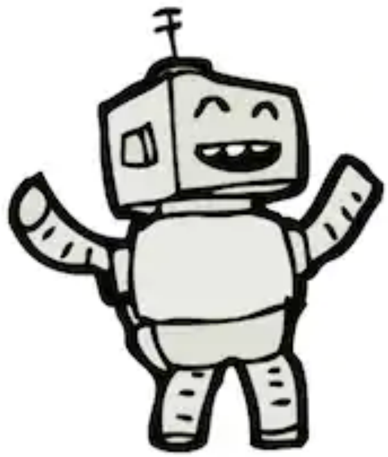
RBE 1001, Introduction to Robotics

Final Project Report

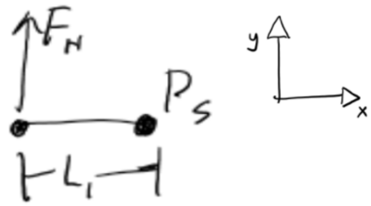


Team 2

Member	Signature	Contribution (%)
Dilce Oliveira	Dilce Oliveira	33%
Owen Rouse	[illegible]	33%
Jenna Tripoli	Jenna Tripoli	33%

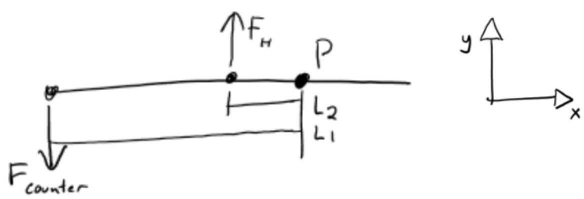
Table of Contents

List of Figures



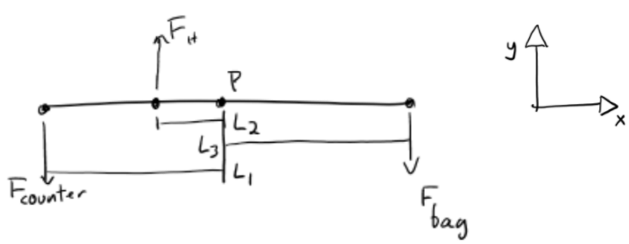
L_1 = 5 cm
P_S = servo pivot point
F_H = force of servo horn

Figure 1: This is the Free Body Diagram of the servo horn (side view). The unknown force is F_H, which is acting upwards on the left end of the servo horn.



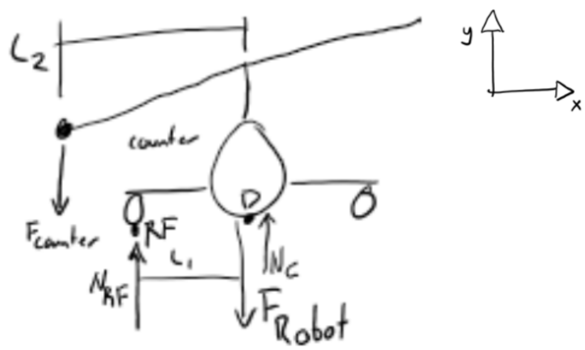
L_1 = 12.6 cm
P = arm pivot point
F_H = (2.04 N) force of servo horn
$F_{counter}$ = force of counterweight

Figure 2: This is the Free Body Diagram of the servo arm (side view). The unknown force is $F_{counter}$, which is acting downwards on the left end of the servo arm.



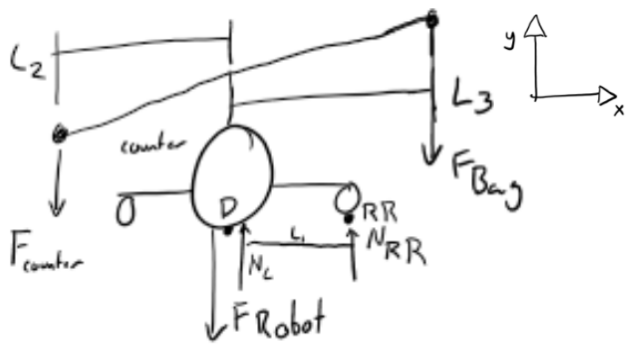
L_1 = 12.6 cm
L_2 = 5 cm
L_3 = 14.8 cm
P = servo arm pivot point
F_H = 2.04 N (force of servo horn)
$F_{counter}$ = 0.810 N (force of counterweight)
F_{bag} = force of bag

Figure 3: This is the Free Body Diagram of the servo arm (side view). The unknown force is F_{bag}, which is acting downwards on the right end of the servo arm.



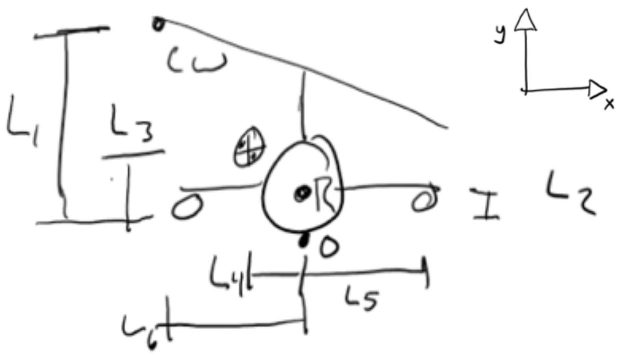
Before picking up a bag:
L_1 = 4.45 cm
L_2 = 11.5 cm
D = point on drive wheel
RF = point on front roller
N_C = normal force on center/drive wheel
N_{RF} = normal force on front roller
F_{robot} = 513 g (mass of robot)
$F_{counter}$ = 82.5 g (mass of counterweight)

Figure 4: This is the Free Body Diagram of the robot (side view) before the robot picks up a bag. The unknown forces are N_C and N_{RF}, which are acting upwards on the center/drive wheel and front roller, respectively.



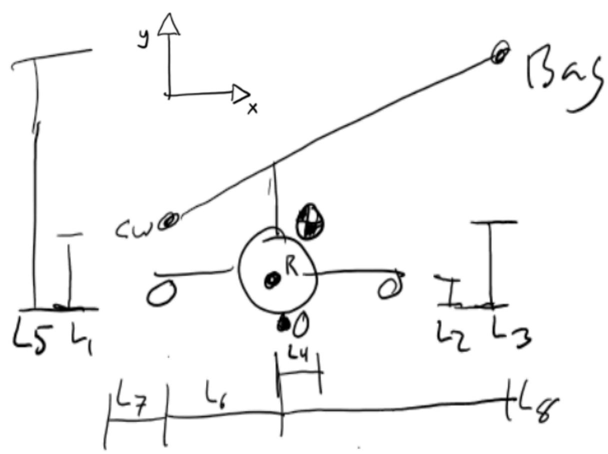
After *picking up a bag:*
L_1 = 6.35 cm
L_2 = 11.5 cm
D = point on drive wheel
RR = point on rear roller
N_C = normal force on center/drive wheel
N_{RR} = normal force on rear roller
F_{robot} = 513 g (mass of robot)
$F_{counter}$ = 82.5 g (mass of counterweight)
F_{bag} = 120 g (mass of bag)

Figure 5: This is the Free Body Diagram of the robot (side view) after the robot picks up a bag. The unknown forces are N_C and N_{RF}, which are acting upwards on the center/drive wheel and rear roller, respectively.



Before picking up a bag:
O = origin
B = bag (120 g)
CW = counterweight (82.5 g)
R = robot weight (513 g)
L_1 = 16.5 cm
L_2 = 5.5 cm
L_3 = vertical distance to Center of Mass
L_4 = horizontal distance to Center of Mass
L_5 = 6.35 cm
L_6 = 11.5 cm

Figure 6: This is the diagram of dimensions and point-masses on the robot (side view) before the robot picks up a bag. The unknown distances are L_3 and L_4, which are the y- and x-distances between the origin and Center of Mass, respectively.



After *picking up a bag:*
O = origin
B = bag (120 g)
CW = counterweight (82.5 g)
R = robot weight (513 g)
L_1 = 10 cm
L_2 = 5.5 cm
L_3 = vertical distance to Center of Mass
L_4 = horizontal distance to Center of Mass
L_5 = 21 cm
L_6 = 11.5 cm
L_7 = 1 cm
L_8 = 16 cm

Figure 7: This is the diagram of dimensions and point-masses on the robot (side view) after the robot picks up a bag. The unknown distances are L_3 and L_4, which are the y- and x-distances between the origin and Center of Mass, respectively.

Introduction

The overall objective of this project was to use a small driving robot with an arm to autonomously pick up and deliver three bags to drop-off zones of varying heights on a field. The robot is allowed to do two runs, with three bags each run. The main components of the robot are the drive wheels, servo arm, ultrasonic rangefinder, drive motor encoders, and the line sensor.

The drive wheels allow the robot to traverse the field, and the servo arm is used to allow the robot to interact with the bags. The servo arm picks up bags from the pick-up zone and the free-range area and places bags in the three drop-off zones on the field. The robot has several sensors that allow for more precise control. One such sensor is the ultrasonic rangefinder, which calculates how far away the sensor is from an object. The ultrasonic rangefinder is used in this project to determine the robot's distance from the bags to drive up to the bags with closed-loop control. The encoders on the drive motors provide a rotational position of the drive wheels. The change in wheel rotation is used in combination with the ultrasonic rangefinder to make the robot center itself on a bag. The encoders are useful for other precise movements with the drive wheels, such as making a turn of a certain number of degrees or driving in a straight line for a set distance. The third sensor on the robot is a line sensor, which allows for the robot to follow the black line on the field to easily get to the pick-up and drop-off zones.

Preliminary Discussion

When programming the robot, using proportional control is advantageous over not using proportional control. The kP allows for far more precise adjustments to the speed of the wheels and thus the speed of the robot. This allows for more precise error corrections while using the line sensor. The kP will allow for a less jittery line following. This will create less wear on the robot and also allow for the robot to travel faster when the kP is correctly adjusted. Although non-proportional control is simple, proportional controls will be far superior.

The speed of the robot should be set to the fastest speed possible while remaining in reliable control. This can be adjusted with the kP value to find the ideal setting with few oscillations and a consistent speed. It is important to not overshoot targets such as the bags and the positions where the robot must make a 90 degree turn.

Each bag will be of the same weight. There is very little advantage to having bags of different weights. This creates a problem where some bags will stress the servo more than others. Any bag movement where the servo needs less weight should result in a reduction of weight in all bags as the servo would be having the same problems with the other bags, which would damage the servo. On the other hand, having lighter bags reduces the maximum potential of weight to achieve the goal.

The robot should pick up the two bags in the pickup area first as this will be easiest and reliable. Then the robot will travel to the free-range area and pick up the bag. The free range bag is worth the most points, however, this is the least reliable bag and has the highest chance of disrupting the flow of the rest of the program. For this reason, the two bags from the pick-up zone will be picked up before the free-range bag.

Problem Statement

There are many goals that the robot must be able to achieve with its design. At a baseline level, the robot needs to be able to drive around the field, pick up and drop off bags with the servo arm, and have enough lifting range with the servo to be able to reach all three heights (0 inches, 1.5 inches, and 3 inches) of the drop-off platforms.

From there, the robot should be able to use its line sensor to follow the black line on the field. This makes it so that the robot moves on the same path every time it moves and can easily line up with the pick-up and drop-off zones. Additionally, the robot should use its ultrasonic rangefinder to calculate the distance between the sensor and a bag. This is used so that the robot can center itself on the bags to have a better chance of picking up the bags from the center rather than the side. The ultrasonic rangefinder is also essential in having the robot locate the bag in the free-range area, since that bag is in a random location. The ultrasonic rangefinder is also used to enable proportional control so that the robot will slow down as it gets closer to a bag, in an effort to reduce overshoot.

Finally, it is most ideal for the robot to be able to deliver a total of at least 600 g in the two runs (which is with six bags). This is achieved by having a maximum weight of the bags that the robot is able to carry being above 100 g. It is preferable that the calculated maximum bag weight is large enough so that there will not be a great amount of stress put on the servo arm while still meeting the 600 g threshold. Another factor that comes into play is the amount of time that it takes for the robot to complete one run. Ideally, the robot should try to complete each run as fast as possible and under 60 seconds. However, this a very difficult goal to achieve, so it is more logical to aim for the robot to deliver two bags in under 90 seconds.

Selection of Final Design

The rangefinder was mounted to the front of the robot. In this position, the robot can have a clear view for the rangefinder as well as have it centered along the middle of the robot. This allows for a more precise location of objects. In addition, with this position, the rangefinder can easily aid with picking up the bags by finding the distance from the bag as the robot approaches.

The arm is positioned to pick up the robot in the rear. This positioning is not ideal for ease of use, however, with the rangefinder in front, the arm is best placed in the rear as the arm and bags do not interfere with the rangefinder. In addition, in this position, the counterweight is over the suspension roller, which means when the robot picks up a bag, less stress is put onto the suspension roller.

The servo bracket and the lifting arm bracket were both placed as close to being between the center of the drive wheels as possible. This directs most of the weight between the two drive wheels. This allows for far superior traction, which in turn, allows the robot to turn far more accurately as well as maintain stability. Without this stability, too much weight is placed on the rollers so when the robot turns, the drive wheels could lose contact with the field.

The pivot location was based on what range of motion was needed to achieve the necessary lifting height. At this pivot point, the robot can achieve the necessary lifting height with precision as well as achieve the weight required to receive maximum points with an adequate counterweight.

Final Design Analysis

The max torque of the servo at 5V is 12.25 N-cm. The position where this occurs is when the servo horn is at a 90° angle with the connecting rod. At this point, the force on the servo horn only has y-components with the connecting arm. The allowable torque at 5V, which is 25% of the stall torque at 5V, is 3.06 N-cm.

The maximum weight for how much the robot can lift is 141 g. However, we used 120 g bags in order to reduce unnecessary stress on the servo. The robot can lift a bag 10.2 cm off the ground (this is the arm's vertical range of motion).

The magnitude of the counterweight is 82.5 g and is positioned 12.6 from the pivot point on the servo arm. This weight and position work well because they allow the weight to counteract the torque of the servo, which in turn counteracts the weight of the bag being lifted.

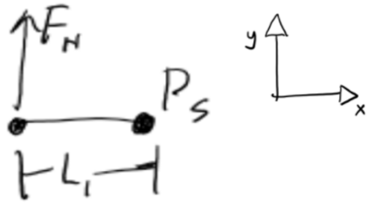
Before a bag is picked up, the center of mass is located 1.6 cm to the left of the center of the drive wheels in the x-axis, 7.9 cm above the bottom of the drive wheels, and halfway between the drive wheels in the z-axis (the robot is symmetrical with weight distribution in the x-axis). 64.2% of the mass is on the drive wheels prior to a bag being picked up.

After a bag is picked up, the center of mass is located 1.4 cm to the right of the center of the drive wheels in the x-axis, 9.4 cm above the bottom of the drive wheels, and halfway between the drive wheels in the z-axis (the robot is symmetrical with weight distribution in the x-axis). 78.6% of the mass is on the drive wheels after a bag is picked up.

Servo Stall Torque at 5V: $\frac{6\text{ V}}{5\text{ V}} = \frac{14.7\ N-cm}{T_{stall}} \rightarrow T_{stall} = 12.25\ N-cm$

Note: The data sheet for the servo indicates that the servo has a stall torque of 14.7 N-cm at 6V. The servo will be operating at a supply voltage of 5V.

Allowable Torque (25% of T_{stall} at 5V): $T_{allowable} = (12.25\ N-cm) * 0.25 = 3.06\ N-cm$

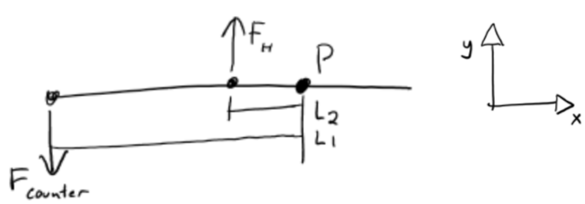


L_1 = 5 cm
P_S = servo pivot point
F_H = force of servo horn

Figure 1: This is the Free Body Diagram of the servo horn (side view). The unknown force is F_H, which is acting upwards on the left end of the servo horn.

$$\boldsymbol{\Sigma M_P = 0 = (F_H * L_1) - T_{allowable}}$$

$$= (F_H * 1.5\ cm) - (3.06\ N-cm) \rightarrow F_H = 2.04\ N$$

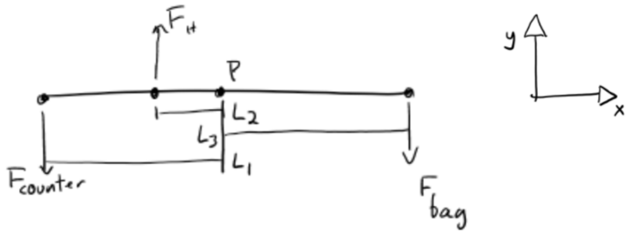


L_1 = 12.6 cm
P = arm pivot point
F_H = (2.04 N) force of servo horn
$F_{counter}$ = force of counterweight

Figure 2: This is the Free Body Diagram of the servo arm (side view). The unknown force is $F_{counter}$, which is acting downwards on the left end of the servo arm.

$$\boldsymbol{\Sigma M_P = 0 = (F_H * L_2) - (F_{counter} * L_1)}$$

$$= (2.04\ N * 5\ cm) - (F_{counter} * 12.6\ cm) \rightarrow F_{counter} = 0.810\ N = 82.5\ g$$

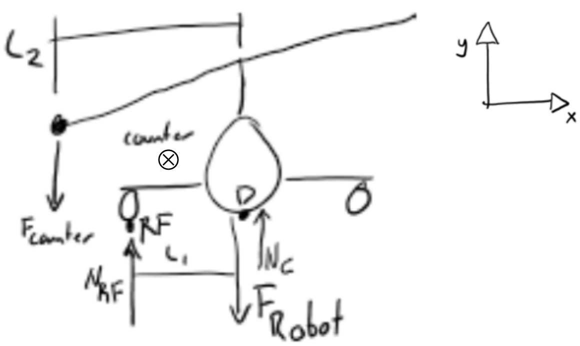


L_1 = 12.6 cm
L_2 = 5 cm
L_3 = 14.8 cm
P = servo arm pivot point
F_H = 2.04 N (force of servo horn)
$F_{counter}$ = 0.810 N (force of counterweight)
F_{bag} = force of bag

Figure 3: This is the Free Body Diagram of the servo arm (side view). The unknown force is F_{bag}, which is acting downwards on the right end of the servo arm.

$$\boldsymbol{\Sigma M_P = 0 = (F_H * L_2) + (F_{counter} * L_1) - \left(F_{bag} * L_3\right)}$$

$$= (2.04\ N * 5\ cm) + (0.810\ N * 12.6\ cm) - \left(F_{bag} * 14.8\ cm\right)$$
$$\rightarrow F_{bag} = 1.38\ N = 141\ g$$



Before picking up a bag:
L_1 = 4.45 cm
L_2 = 11.5 cm
D = point on drive wheel
RF = point on front roller
N_C = normal force on center/drive wheel
N_{RF} = normal force on front roller
F_{robot} = 513 g (mass of robot)
$F_{counter}$ = 82.5 g (mass of counterweight)
⊗ = Center of Mass in x-y plane

Figure 4: This is the Free Body Diagram of the robot (side view) before the robot picks up a bag. The unknown forces are N_C and N_{RF}, which are acting upwards on the center/drive wheel and front roller, respectively.

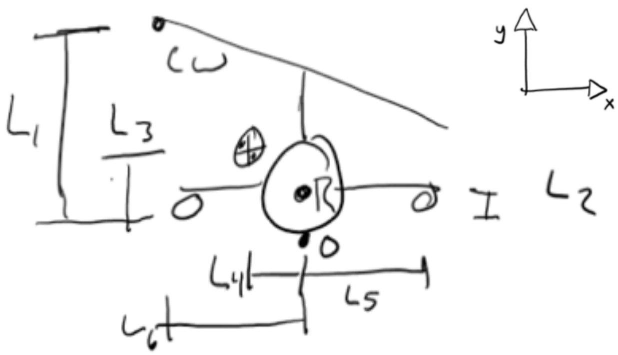
$$\boldsymbol{\Sigma M_R = 0 = (F_{robot} * L_1) - \left(F_{counter} * (L_2 - L_1)\right) - (N_C * L_1)}$$

$$= (513\ g * 4.45\ cm) - \left(82.5\ g * (11.5\ cm - 4.45\ cm)\right) - (N_C * 4.45\ cm) \rightarrow N_C = 382\ g$$

$$\boldsymbol{\Sigma M_D = 0 = (N_{RF} * L_1) - \left(F_{counter} * (L_2)\right)}$$

$$= (N_{RF} * 4.45\ cm) - (82.5\ g * 11.5\ cm) \rightarrow N_{RF} = 213\ g$$

Percent of mass on drive wheel before bag pickup: $\frac{\mathbf{N_c}}{\mathbf{N_C + N_{RF}}} * \mathbf{100} = \frac{382\ \mathrm{g}}{382\ \mathrm{g} + 213\ \mathrm{g}} * 100 = 64.2\%$



Before picking up a bag:
O = origin
B = bag (120 g)
CW = counterweight (82.5 g)
R = robot weight (513 g)
L_1 = 16.5 cm
L_2 = 5.5 cm
L_3 = vertical distance to Center of Mass
L_4 = horizontal distance to Center of Mass
L_5 = 6.35 cm
L_6 = 11.5 cm

Figure 5: This is the diagram of dimensions and point-masses on the robot (side view) before the robot picks up a bag. The unknown distances are L_3 and L_4, which are the y- and x-distances between the origin and Center of Mass, respectively.

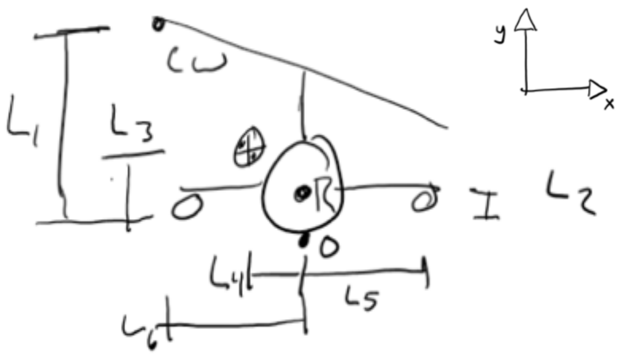
$$\boldsymbol{\Sigma M_R = 0 = \left(F_{bag} * (L_3 - L_1)\right) - (F_{robot} * L_1) - \left(F_{counter} * (L_1 + L_2)\right) + (N_C * L_1)}$$

$$= \left(120\ g * (16\ cm\ -\ 6.35\ cm)\right) - (513\ g * 6.35\ cm)$$
$$- \left(82.5\ g * (6.35\ cm + 11.5\ cm)\right) + (N_C * 6.35\ cm) \rightarrow N_C = 563\ g$$

$$\boldsymbol{\Sigma M_D = 0 = \left(F_{bag} * L_3\right) - (F_{counter} * L_2) - (N_{RR} * L_1)}$$

$$= (120\ g * 16\ cm) - (82.5\ g - 11.5\ cm) - (N_{RR} * 6.35\ cm) \rightarrow N_{RR} = 153\ g$$

Percent of mass on drive wheels after bag pickup: $\frac{\mathbf{N_c}}{\mathbf{N_C + N_{RR}}} * \mathbf{100} = \frac{563\ g}{563\ g + 153\ g} * 100 = 78.6\%$



Before picking up a bag:
O = origin
B = bag (120 g)
CW = counterweight (82.5 g)
R = robot weight (513 g)
L_1 = 16.5 cm
L_2 = 5.5 cm
L_3 = vertical distance to Center of Mass
L_4 = horizontal distance to Center of Mass
L_5 = 6.35 cm
L_6 = 11.5 cm

Figure 6: This is the diagram of dimensions and point-masses on the robot (side view) before the robot picks up a bag. The unknown distances are L_3 and L_4, which are the y- and x-distances between the origin and Center of Mass, respectively.

$$\boldsymbol{L_3} = \frac{(\boldsymbol{R} * \boldsymbol{L_2}) + (\boldsymbol{CW} * \boldsymbol{L_1})}{(\boldsymbol{R} + \boldsymbol{CW})}$$

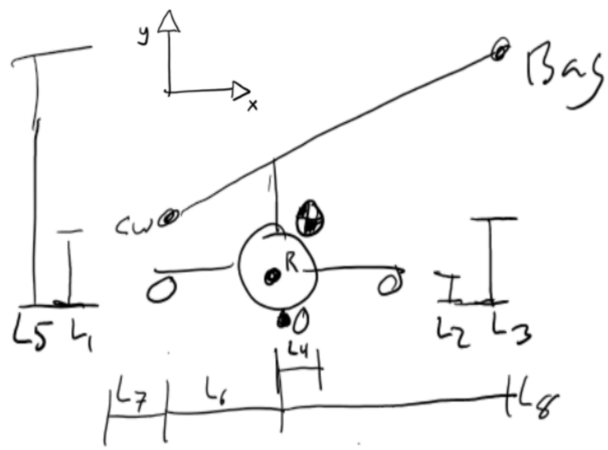
$$= \frac{(513\ g * 6.35\ cm) + (82.5\ g * 16.5\ cm)}{513\ g + 82.5\ g} = 7.0\ cm$$

$$\boldsymbol{L_4} = \frac{(\boldsymbol{R} * \boldsymbol{L_2}) + (\boldsymbol{CW} * (\boldsymbol{L_6} + \boldsymbol{L_5}))}{(\boldsymbol{R} + \boldsymbol{CW})} - \boldsymbol{L_5}$$

$$= \frac{(513\ g * 6.35\ cm) + (82.5\ g * (11.5\ cm\ + 6.35\ cm))}{513\ g + 82.5\ g} - 6.35\ cm = 1.6\ cm$$

Note: Since the robot is symmetrical along the z-axis and the origin is directly in-between of the bottom of the drive wheels in the z-axis, the z-coordinate of the Center of Mass is 0.

The Center of Mass before bag pickup is at the x-y-z coordinate (-1.6, 7.9, 0). This means that the Center of Mass is 1.6 cm to the left of the origin in the x-axis, 7.9 cm above the origin in the y-axis, and at the origin in the z-axis.



After *picking up a bag:*
O = origin
B = bag (120 g)
CW = counterweight (82.5 g)
R = robot weight (513 g)
L_1 = 10 cm
L_2 = 5.5 cm
L_3 = vertical distance to Center of Mass
L_4 = horizontal distance to Center of Mass
L_5 = 21 cm
L_6 = 11.5 cm
L_7 = 1 cm
L_8 = 16 cm

Figure 7: This is the diagram of dimensions and point-masses on the robot (side view) after the robot picks up a bag. The unknown distances are L_3 and L_4, which are the y- and x-distances between the origin and Center of Mass, respectively.

$$\boldsymbol{L_3} = \frac{(\boldsymbol{R} * \boldsymbol{L_2}) + (\boldsymbol{CW} * \boldsymbol{L_1}) + (\boldsymbol{B} * \boldsymbol{L_5})}{(\boldsymbol{R} + \boldsymbol{CW} + \boldsymbol{B})}$$

$$= \frac{(513\ g * 5.55\ cm) + (82.5\ g * 16.5\ cm) + (120\ g * 21\ cm)}{513\ g + 82.5\ g + 120\ g} = 9.4\ cm$$

$$\boldsymbol{L_4} = \frac{(\boldsymbol{R} * (\boldsymbol{L_7} + \boldsymbol{L_6})) + (\boldsymbol{CW} * \boldsymbol{L_7}) + (\boldsymbol{B} * (\boldsymbol{L_7} + \boldsymbol{L_6} + \boldsymbol{L_8}))}{(\boldsymbol{R} + \boldsymbol{CW})} - \boldsymbol{L_5}$$

$$= \frac{(513\ g * (1\ cm + 11.5\ cm)) + (82.5\ g * 1\ cm) + (120\ g * (1\ cm\ + 11.5\ cm + 16\ cm))}{513\ g + 82.5\ g + 120\ g} - 11.5\ cm = 1.4\ cm$$

Note: Since the robot is symmetrical along the z-axis and the origin is directly in-between of the bottom of the drive wheels in the z-axis, the z-coordinate of the Center of Mass is 0.

The Center of Mass after bag pickup is at the x-y-z coordinate (-1.4, 9.4, 0). This means that the Center of Mass is 1.4 cm to the left of the origin in the x-axis, 9.4 cm above the origin in the y-axis, and at the origin in the z-axis.

Pseudocode

```
// rangefinder
Rangefinder rangefinder;
double distanceToBag = 5;    // how far away the robot should be from the bag when picking it up
int bagThreshold = 30;    // how far away the robot can be to detect a bag

// driving
Motor leftMotor, rightMotor;
double diam = 2.75;    // diameter of drive wheels, in inches
double track = 5.875;    // distance between drive wheels, in inches
double defaultSpeed = 200;    // default speed for drive wheels, in degrees/sec
double kP = 0.1;    // proportional constant for drive wheels

// line following
const int reflectancePin1 = 39, reflectancePin2 = 36;
int reflectance1, reflectance2;    // output of the two line sensor pins
Int lineThreshold = 1200;    // minimum line sensor output for the line to be seen

// lifting arm
Servo armServo;
int deliverA = 0, deliverB = 75, deliverC = 130;    // angles to deliver bags on zones A, B, and C

// state machine
enum ROBOT_STATES { LINE_FOLLOW_OUT, APPROACH_BAG, STREET_1, STREET_2, STREET_3,
        STREET_4, STREET_5, LINE_FOLLOW_CRUTCH, end };
int robotState = LINE_FOLLOW_OUT;
int bagState = 0;    // 0 = Bag 1 (zone A), 1 = Bag 2 (zone B), 2 = Bag 3 (zone C)

// line following
void lineFollow(int reflectance1, int reflectance2) {
        double error = reflectance1 - reflectance2;
        double effort = kP * effort;
        rightMotor set speed to (defaultSpeed + effort);
        leftMotor set speed to (defaultSpeed - effort);
}

// drive straight
void straight(double distance) {
        double degreesToMove = (360 * distance) / (diam * PI);
        leftMotor start move for (degreesToMove at defaultSpeed);
        rightMotor move for (degreesToMove at defaultSpeed);
}
```

```
// turn in place
void hardTurn(double angle) {
	stop drive motors;
	double degreesToMove = (angle * track) / diam;
	leftMotor start move for (degreesToMove at 100 degress/sec);
	rightMotor move for (-degreesToMove at 100 degrees/sec);
	stop drive motors;
}

// turn corners
void softTurn(double angle) {
	stop drive motors;
	double degreesToMove = (2 * angle * track) / diam;
	if (angle >= 0) leftMotor move for (degreesToMove at 150 degrees/sec);    // turn right
	else rightMotor move for (-degreesToMove at 150 degrees/sec);    // turn left
	stop drive motors;
}

// deliver bag on its correct zone
void dropOffBag(void) {
	turn clockwise in place 180 degrees;
	bagState++;
	if (bagState == 1) armServo move to (deliverA);    // first bag delivered to zone A
	else if (bagState == 2) armServo move to (deliverB);    // first bag delivered to zone B
	} else if (bagState == 3) {
		armServo move to 180 degrees;
		armServo move to (deliverC);    // first bag delivered to zone C
	}
}
```

```
// center robot on bag using rangefinder, approach bag, and pick up bag
void pickUpBag(void) {
	double leftEdge = 0, rightEdge = 0, error;
	armServo move to 0 degrees;

	// while bag is out of range
	while (rangefinder distance > bagThreshold) turn clockwise in place;
	leftEdge = rightMotor current degrees;
	rightEdge = leftEdge;    // fail safe so that the default is not turning
	while (rangefinder distance < bagThreshold) { }    // wait for bag to be out of range
	if (rangefinder distance > maxRangeDistance) rightEdge = rightMotor current degrees;
	turn counterclockwise in place (rightEdge - leftEdge) / 4);    // to the center of the bag

	while (rangefinder distance > distanceToBag) {
		error = rangefinder distance;
		rightMotor and leftMotor set effort to (error * kP / 4);    // approach bag
	}
	armServo move to 0 degrees;
	stop drive motors;
	drive backward 2 inches;
	turn in place 180 degrees;
	drive backward 3.25 inches;
	armServo move to 180 degrees;    // pick up bag
}
```

```
// switch statement to decide next robot movement
void updateRobotState(void) {
	switch (robotState) {
		// going down STREET_2 heading towards pick-up zone
		case LINE_FOLLOW_OUT:
			if (line sensor sees street of pick-up zone) {
				turn to the left 85 degrees;
				robotState = APPROACH_BAG;
			} else lineFollow(reflectance1, reflectance2);
			break;

		//  after bag drop-off, deciding how to turn to get back onto STREET_2
		case LINE_FOLLOW_CRUTCH:
			if (line sensor sees delivery zone intersection) {
				if (bagState == 1) {
					turn to the left 85 degrees;
					robotState = LINE_FOLLOW_OUT;
				} else if (bagState == 2) {
					drive forward 2 inches;
					robotState = STREET_3;
				} else if (bagState == 3) {
					stop drive motors;
					robotState = end;
				}
			} else lineFollow(reflectance1, reflectance2);
			break;

		case APPROACH_BAG:    // approaching to pick up a bag
			if (line sensor sees T at pick-up zone) {
				drive backward 5 inches;
				turn in place counterclockwise 30 degrees;
				armServo move to 0 degrees;
				pickUpBag();
				drive forward 3 inches;
				robotState = STREET_1;
			} else lineFollow(reflectance1, reflectance2);
			break;

		case STREET_1:    // leaving pick-up zone
			if (line sensor sees end of pick-up street) {
				stop drive motors;
				turn to the right 85 degrees;
				robotState = STREET_2;
			} else lineFollow(reflectance1, reflectance2);
			break;
```

```
case STREET_2:   // returning from STREET_1, deciding which street to deliver bag
        if (line sensor sees delivery zone intersection) {
                stop drive motors;
                if (bagState == 0) {
                        turn to the right 85 degrees;
                        robotState = STREET_3;
                } else if (bagState == 1) {
                        turn to the left 85 degrees;
                        robotState = STREET_4;
                } else if (bagState == 2) {
                        drive forward 2 inches;
                        robotState = STREET_5;
                }
        } else lineFollow(reflectance1, reflectance2);
        break;

case STREET_3:    // first bag drop-off on zone A
        if (line sensor sees T at zone A) {
                stop drive motors;
                if (bagState == 2) {
                        turn clockwise in place 90 degrees;
                        drive forward 5 inches;
                        turn counterclockwise in place 30 degrees;
                        servoArm move to 0 degrees;
                        pickUpBag();    // pick up free-range bag
                        turn counterclockwise in place 35 degrees;
                        robotState = STREET_2;
                } else {
                        dropOffBag();
                        robotState = LINE_FOLLOW_CRUTCH;
                }
        } else lineFollow(reflectance1, reflectance2);
        break;

case STREET_4:    // second bag drop-off on zone B
        if (line sensor sees T at zone B) {
                stop drive motors;
                dropOffBag();
                robotState = LINE_FOLLOW_CRUTCH;
        } else lineFollow(reflectance1, reflectance2);
        break;
```

```
        case STREET_5:    // third bag (free-rage bag) drop-off on zone C
            if (line sensor sees T at zone C) {
                stop drive motors;
                dropOffBag();
                robotState = LINE_FOLLOW_CRUTCH;
            } else lineFollow(reflectance1, reflectance2);
            break;

        case end:    // exit program
            break;
    }
}

void loop() {
    while (true) {
        reflectance1 = ADC value of pin 1;    // update line sensor values
        reflectance2 = ADC value of pin2;    // update line sensor values
        updateRobotState();    // enter switch statement
    }
}
```

Project Summary

Throughout this project, our team had great internal communication, asked for help when needed, and was creative with problem-solving. We spent many hours in the Foisie Lab completing activities and getting TA input when necessary. Due to a series of unfortunate circumstances related to COVID, our team had to adapt to the situation to get all three of our runs done. Our timeline was thrown off-course, but we did not hesitate to reach out to our professor to ask for an extension.

The main issue with our robots was consistency. In many runs, the robots would stop suddenly and would require a gentle tap to keep moving. Our team was also impacted by many unfortunate situations as two team members were moved into isolation before the demos were recorded, and the third member was across the country. Regardless, we achieved our goals.

In our final demos, two of the three robots worked almost perfectly. For these two robots, we were able to pick up and deliver both of the bags in the pick-up zone the free-range bag. In both runs the robots stopped moving and needed a slight tap to continue, but otherwise operated as expected. The third robot had both a broken line sensor rangefinder, so it was unable to do line following or find the free-range bag. However, this robot was still able to pick up and deliver two bags. With more time, our team would make the robots more consistent. We would spend more time diagnosing the issue stopping/tapping issue and hopefully eliminate that problem. We would also work on creating a softer approach to picking up the bags when using the rangefinder. The robots would sometimes approach the bag at a very high speed which would cause overshoot and correction, and this could have been avoided with more tuning. Overall, our team worked very hard and are proud of the work we presented.