Preliminary Design Report

Submitted to:
Dr. Phil Schlosser
GTA Melissa Hrivnak

Created by:
Team I

Holly van der Lans
Stephanie Smithson
Ryan Devine
Rob McEwan

Engineering 1182
The Ohio State University
Columbus, OH
27 March 2017

Executive Summary

The main purpose of this AEV lab was to construct a vehicle and program it to move across a track and respond to track cues accordingly to better protect the galactic empire. In addition to the actual creation of the vehicle, it was important that the team developing the design made sure to be as cost and energy efficient as possible due to power limitations on remote planets. To achieve this, the team developed code to upload to an Arduino that was attached to motors as well as sensors to keep track of vehicle movement. One of the team's main goals was to make the AEV as light as possible so that weight would not be an obstacle when programming the vehicle's power usage during movement. The main objective on this track was for the AEV designed by the team to pick up an R2D2 unit with a magnet and carry it back across the track.

In an attempt to make the vehicle lighter, the group swapped out the base piece from a large rectangle to a T-shaped part and assembled the Arduino, battery, and motors on it. To achieve an X-Wing design, the team cut small wing shaped pieces out of a folder and assembled them onto the vehicle. By taping some cardboard and adding a rubber band to the servo, a brake was installed to press up against the track and keep the vehicle in place. This installation has made the vehicle much more consistent with stopping, especially when it approaches the gate.

After the design process was completed for the AEV, the programming was the next main concern. With diagrams provided to the team in the lab manual, the team converted physical parameters to marks and programmed the vehicle to travel 345 marks (14 ft) before stopping at the gate. The vehicle then was programmed to brake the motors, cue the servo brake, wait for five seconds, release the brake, and then continue to the end of the track to retrieve the R2D2. After some trial and error, the team changed the code to reverse the motors when approaching the gate to avoid ramming into it, as well as braking the motors before calling the go to absolute position method for a second time. This program helped to better control the speed of the AEV when approaching the gate and made the stop much more smooth.

This lab is important to the scientific community because it practiced restraint on cost and energy. It also involved a lot of troubleshooting and changing parameters. The end result of this lab was a flawless run on the track. The AEV began traveling towards the gate, slowed down, then stopped at the sensor before heading to the end of the track. It then picked up the AEV, waited six seconds, then proceeded back towards the gate. After reaching the sensor and waiting for the gate arm to lower, it returned to the beginning of the track, successfully rescuing the R2D2 unit.

Some recommendations would be to start out building the vehicle with the least amount of material as possible to keep it lightweight. From there it's easy to add onto it some necessary components to actually improve the vehicle as it moves. Additionally, programming the vehicle in parts and breaking up the track into segments helped with organizing and implementing the actual movement of the AEV.

Table of Contents

List of Figures and Tables

Introduction

The objective of the labs was to determine an AEV design that could transport the R2D2 units across the land while also being energy and cost efficient. Another objective of the lab is that the AEV must complete the task within a time limit and follow all instructed rules. In order to come to a conclusion of which AEV would fit better for the task, some goals needed to be met by using design and efficiency. These goals included being able to stop at the gate for a certain amount of time in order to gain access into the cargo area. Another goal was to be able to travel back to the drop off area and safely deliver the cargo. In order to meet these goals AEV designs had to complete multiple tests that collected data about their performance. These tests were then compiled to reach an overall conclusion

The report was created to demonstrate major trends between design and efficiency. These trends helped to solve the problem of finding an energy and cost efficient AEV that could travel to pick up the R2D2 units in a timely manner. The following report includes an experimental methodology section where the experimental procedure of the labs is discussed, followed by a results section where the data from the labs is presented. Following the results is a discussion section where the analysis, potential error, and comparison to theory of the report are located. After the discussion section is the conclusions and recommendations that discusses which AEV design was chosen and why it is better for the task.

Experimental Methodology

The setup for the labs involved the following materials and equipment: the AEV track, the arduino, the motors, the propellers, the wheel sensors, the servo, Arduino Sketchbook, and MatLab. The AEV track consists of a half oval monorail like track that has a gate in the middle that can only be activated by tripping the first sensor. The following connections were made to the arduino board: the 2 motors, the wheel sensors, and the servo. The arduino board was used to store and transfer code to the other equipment parts, which then runs the AEV on the track.

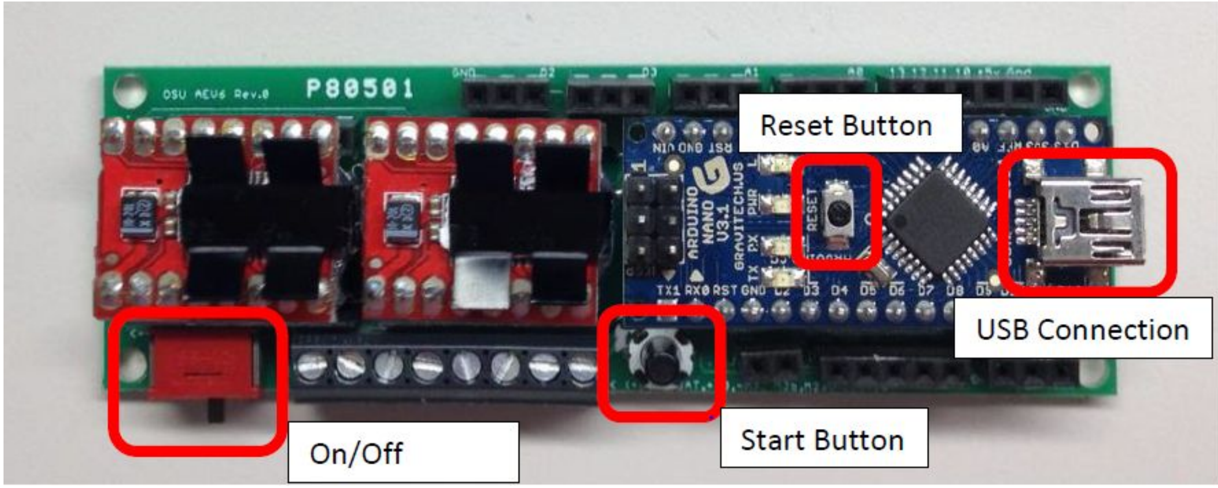


Figure 1: Arduino Board

The motors provided power to the AEV and which allowed the AEV to move forwards and backwards on the track. The propellers are attached to each motor and help the AEV move by either using the push or pull method.

Figure 2: Motor and Propellor Assembly

The wheel sensors were used to measure distances along the track that the AEV had travelled. The wheel sensors work by counting how many marks have been travelled by the AEV and then complete codes when it gets to a certain number of marks.

Figure 3: Wheel Sensors

The servo helps to provide a control of angular position. The Arduino Sketchbook was used to write code and then transfer it to the arduino using a USB cable.

The motors and wheel sensors were calibrated by creating and running a set of code to make sure each part was functioning properly. The calibration of the motors was completed in lab 1 before the AEV was even put on the track. In order to calibrate the wheel sensors a special code called the reflectanceSensorTest() was used, which helped to make sure the wheel sensors were measuring marks properly and that the wheels were travelling in the proper direction. This calibration took place in lab 2 but also occurred in other labs when the wheel sensors weren't working properly. After insuring that all parts of the equipment were functioning properly a code was created using the Arduino Sketchbook. The sketchbook uses the same coding process as a C or C++ program but has specific functions used for the AEV. Once the code was created the AEV was ready to be tested. Using a USB cable the code was transferred from the sketchbook into the arduino controller. Once the arduino was started the code would be executed making it run along the track and perform the tasks written in the code. After each run on the track the AEV was hooked back up to the computer using the USB cable in order to retrieve its data from the run. The AEV reads in the data using EEPROM which stands for Electrically Erasable Programmable Read-Only Memory. The AEV's EEPROM data was collected by downloading the aevDataRecorder file to MatLab. The MatLab file is used to convert the data into parameters that can be easily understood, such as time to seconds, current to amps, voltage to volts, and distance and position to meters. Converting the data into these parameters makes it easier to calculate the power of the AEV at any point in time along it's run. The power is then graphed along with time to calculate the energy used from the AEV, in order to find the energy a midpoint approximation of the Riemann Sum was used.

The process of running the AEV on the track was used amongst multiple different designs. These designs were all tested using the same code in order to be compared against each other. There were two different types of methods that were used to compare AEV designs against each other. The first method was screening which used +, -, and 0 to rank the AEV in different categories. After assigned a value in each category, the categories were totaled up giving the AEV an overall score. Based on the score given, it was decided whether or not the AEV was fitting for the tasks. Another method of test AEV design was the concept scoring, which was similar to concept screening but used numbers inside of +,-, and 0. Each AEV was tested and then ranked in each category. Both methods were using in determining the design of the AEV.

Results

The two AEV concepts currently being tested are the regular X-wing model and the X-wing model with a servo- brake included. Both models have distinctive dual wings on the rear of the model, as well as a lightweight plastic body that surrounds the components of the AEV. The wings and body are implemented to mimic the famous star wars space cruiser while also making the AEV aerodynamic. Dual propellor propulsion is implemented on the the rear of the model in a "push" system. In Both models the arduino was mounted on the top while the battery on the undercarriage. The difference between the two models was the proposed servo- brake, and would be mounted on the wheel arm of the model. A brake lever arm would be connected to the servo in order to come in contact with the rail the AEV travels on. The servo's actions would be applied easily in the existing code where additional braking is necessary. The brake would be implemented at points along the track where precise stopping is necessary, such as at the midpoint gate.

From the four designs proposed in lab 4 (X-wing, Y-wing, Double Propellor, Reference), the team decided to move forward with the X-wing model as it scored the highest from the screening and scoring charts, tables 1 & 2. In the success criteria: balanced, minimal blockage, center of gravity location, maintenance, durability, cost, environmental and aesthetics; the X-wing primarily scored higher than its competitors. The criteria where it did not perform well in comparison to others (maintenance, durability, and environmental), were improved through changes the team decided on. The majority of the proposed model was left unchanged, however, the main issue with the model was the excess weight from the wings and body. This issue was solved by using light weight plastic material for construction. This also made the AEV significantly more aerodynamic. Maintenance and durability were also simultaneously improved by using the rigid materials, and securing bolts on the AEV better. The next proposed change to the model (servo-brake), was an extension of the original X-wing, it would provide the model with a much more reliable and precise means of braking, to ensure its success on the track.

Table 1: Screening Chart from Lab 5

Success Criteria	Reference	Design X-Wing	Design Y-wing	Design Double Prop
Balanced	0	0	0	0
Minimal blockage	0	0	0	-
Center-of-gravity location	0	+	0	+

Maintenance	0	-	0	-
Durability	0	0	0	-
Cost	0	-	-	0
Environmental	0	+	+	-
Aesthetics	0	+	-	+
Sum +'s	0	3	1	2
Sum 0's	8	3	5	4
Sum -'s	0	2	2	2
Net Score	0	1	-1	0
Continue?	Combine	Yes	Combine	No

Table 2: Scoring Chart from Lab 5

		A Reference		Design X-wing		Design Y-wing		Design Double Prop	
Success Criteria	**Weight**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**
Balanced	15%	3	0.45	3.5	0.525	3.5	0.525	3	0.45
Minimal blockage	5%	3	0.15	4	0.60	4	0.20	0.5	0.025
Center-of-gravity location	15%	2	0.30	3	0.45	3	0.45	4	0.60

Maintenance	5%	3	0.15	1	0.05	3	0.15	2	0.10
Durability	10%	2	0.20	2	0.20	3	0.30	1	0.10
Cost	15%	3	0.45	2	0.30	2.5	0.375	3	0.45
Environmental	15%	3	0.45	3	0.45	4	0.60	2	0.30
Aesthetics	20%	1	0.20	5	1	2.5	0.50	3	0.60
Total Score	100%		2.35		3.575		3.10		2.625
Continue?			No		Develop		Combine		No

The testing of the X-wing with the servo-brake was highly effective yet also highly inconsistent. Regardless of having identical code structure, and not changing anything physically on the AEV, the AEV was observed to either stop too early with the brake or not stop at all. The team is currently dissecting code structure to ensure that no issues in it would result in the model “skipping” the instructions for it to perform. Considering the code was implemented using the “goToAbsolutePosition” function, with the same line of code, the AEV should perform the same actions on the same portions of the track with slight variation. Due to the drastic differences observed, conflicting code lines or hardware issues are likely to blame.

During the first few trials, the servo performed as expected and was implemented precisely before the midpoint stopping gate. The AEV was brought to a complete halt when the brake made contact with the rail and was observed to use less power to stop. This was opposed to using just a reverse motor function in order to stop the AEV at the midpoint, which would use more power over a longer period of time, and being less precise. The power vs time figure can be observed in figure 3.

In comparison to the regular X-wing, the code was primarily the same, with the only change being the implementation of the “rotateServo” function to come in contact with the rail when it was necessary for the AEV to stop. The original model had to have a longer period or reversing the motor in the opposite direction of its travel in order to stop. This uses a larger amount of power than when using the servo-brake. Also, as stated before, the stopping of the AEV before the midpoint gate was highly inconsistent and often triggered the gate, unless a large reverse motor percentage was implemented.

Based on the results of the test runs, although inconsistent due to code and / or hardware malfunctions, the servo-brake performed as expected and the team plans to continue with it in the final design. The servo brakes adds precise stopping that cannot be mimicked with propulsion alone.

The performance test was an integral part of the team’s decision making process for designs to be considered. It was especially helpful in making a choice between the two designs the team narrowed its scope to. What was found during performance test 1 and comparing two designs was that the servo motor design was the most effective at stopping the AEV where it was supposed to stop. Without the performance test, the team might not have been motivated to create the design with the servo motor brake, and thus would have a lower chance of completing the mission.

In figure 4 below, the propellor efficiency vs advance ratio graph is shown which was derived from a wind tunnel test. This graph, from system analysis 1 helped the team decide the correct propellor to use. What was learned from this test was that the 3030 propellor was much more efficient than the 2510 propellor this was essential in making the correct propellor choice. After the most efficient propellor model was determined, the most efficient motor speed percentage to run the AEV with was determined. According to figure 4, an advanced ratio or 0.4 yielded the most ideal efficiency percentage of approximately 50%. Using the advanced ratio formula it was determined that the ideal motor speed percentage was approximately 30. From this data, the team implemented a motor speed equal or near this value as much as possible within the structure of the code. Regarding the two different concepts tested in performance test 1, the advanced test ratio and wind tunnel test data was applied in the same way to both models because of how similar they were. The advanced ratio as well as any other wind tunnel data would not be affected by the implementation of a servo motor.

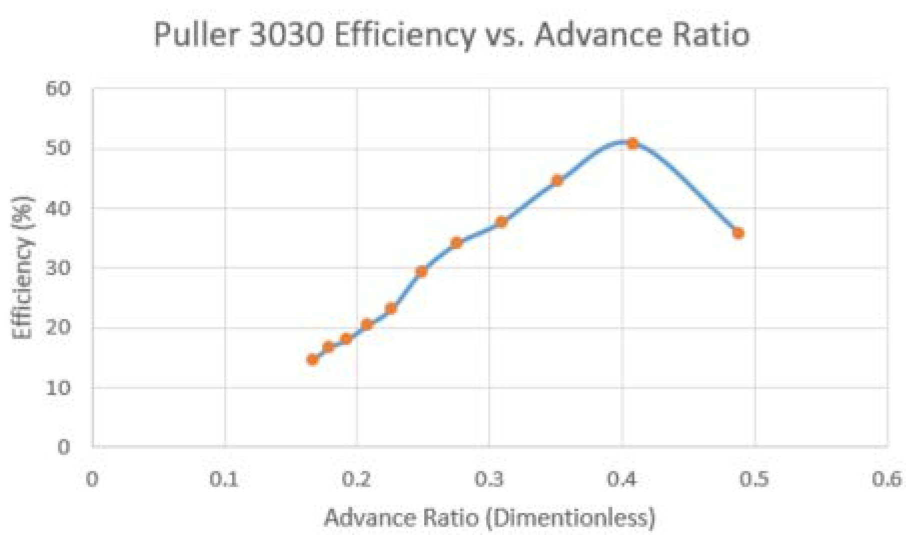


Figure 4: Propulsion Efficiency vs Advance Ratio

Figure 5 below, shows the supplied power vs time for the test run with the servo brake in performance test 1. What is shown is the specific phases of the program the team wrote to run the AEV. This data wass useful in making a final decision on whether or not to include the servo brake. What was evident to the team after analyzing this data, was that the servo brake used a lot less power than imagined. This then allowed the team to make the easy decision to keep the brake because by using it instead of the reverse motors, the AEV will use less power to get to the stopping position and complete the task. The breakdown of the different phases in the figure are explained in table 3.

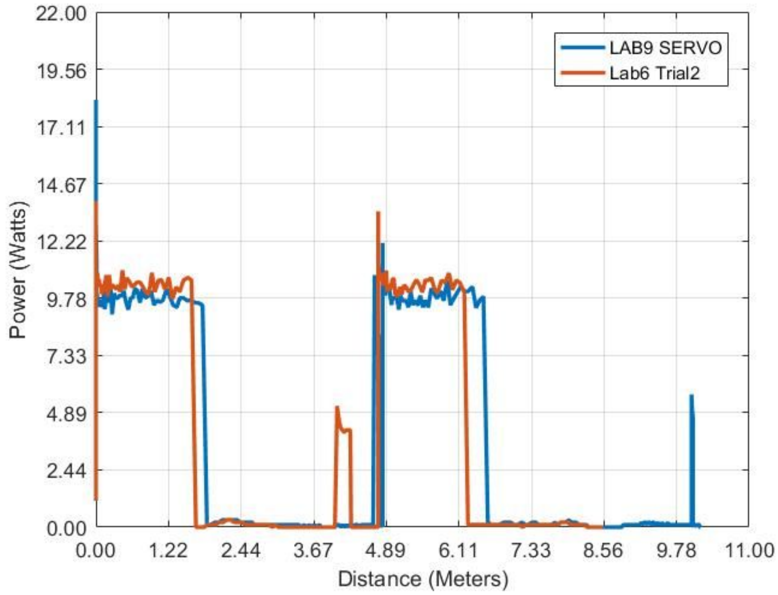


Figure 5: Supplied Power vs Time, Lab 8 Performance Test

In table 3 the breakdown of the different phases of the AEV test run are explained. It was important to understand which component of the graph correlates to what in the actual code. Without that understanding, the team would have a hard time correcting the errors moving forward because there would be a lack of data to base the decisions on.

Table 3: Phase break down of Figure 5 and the Arduino Code

Time(s)	Activity Consuming Energy	Description	Servo Code	Contained in non-servo code?
0-0.1	Servo Motor	Brake is rotating to start position	rotateServo(45);	N

0-3	Propellor Motors (Forward)	Initial thrust to move the AEV	reverse(4);
			motorSpeed(4,32);
			goToAbsolutePosition (145);
			brake(4);
				Y

7-7.5	Propellor Motors (reverse)	Thrust to slow the AEV down for the Gate	goToAbsolutePosition (378);
			reverse(4);
			motorSpeed(4,30);
			goFor(0.5);
			brake(4);
				Y

8.1-9.6	Servo Motor (Brake)	Stop the AEV within the Gate	goToAbsolutePosition (390);	N

		range (Includes rotating to stop position and back to go position)	rotateServo(10);
			goFor(3);
			rotateServo(45);
			goFor(7.5);	

18.6-21.4	Propellor Motors (Forward)	Thrust to move the AEV toward the pickup location	reverse(4);
			motorSpeed(4,32);
			goToAbsolutePosition (525);
			brake(4);
				Y

28.2-28.4	Propellor Motors (Reverse)	Slow the AEV down before it hits the cart	goToAbsolutePosition (810);
			reverse(4);
			motorSpeed(4,20);
			goFor(0.25);
			brake(4);
				Y

Discussion

It was inferred from the data that the X-wing model with the servo-brake included performed better than its competitor, the plain X-wing model. From figures observed, the servo-brake implication results in the AEV model using less energy to stop in comparison to a propulsion stop alone. This is because less power is applied over a period of time to stop the model by propulsion. Also, as stated before, the use of the servo-brake allows the AEV to stop precisely, without any drift that may result in triggering the midpoint gate to shut. Besides the obvious errors in code consistency, the servo brake proved to be a success and will be used in the future. Some areas of potential error in this project could come stem from a few places. The first being the wheel sensors. These were found to be very inaccurate compared to what was calculated based on the distance the AEV traveled. The only way to correct this, though, would be to either replace the sensors, or keep checking them and adjusting the code to match the inaccuracy. The next source of error would be the servo brake. What the team found of recent, was that the servo doesn’t always operate as planned. It sometimes doesn’t get triggered thus allowing the AEV to hit the gate. This can be resolved by adjusting the code because the speculation was that the code is the issue. The last source of error would be the bounce back problem of the end of the track. This could cause the AEV to either hit the gate on the way back or stop too short. The team worked to resolve this using the servo brake to stop the bounce back. The experimental results observed correlate with the theoretical predictions the group made about the addition of the servo-brake. The group predicted that if the servo-brake was applied the AEV would thus use less power in order to make a complete stop and would not vary its stopping point because of outside interference and unwanted drift. From physical observations and eeprom data, the AEV without the servo-brake was observed to stop at slightly different points due to drift. At points where precision is necessary, such as the midpoint gate, an abrupt stop is necessary to not trigger it to shut, ruining the run. With the servo brake, the "goToAbsolutePosition" function is used to stop the AEV at the exact point needed, 390 marks.

Conclusions and Recommendations

From the results obtained, the team elected to move forward with the implication of the servo-brake. After the first few successful trials the team experienced extreme inconsistency with where the AEV performed its functions along designated portions of the track. Because the "goToAbsolutePosition" function is only dependent on track position,and it was held constant this should not have occurred.. The only issue that may have arouse would be hardware malfunction, specifically the reflectance sensor or servo itself. Also, lines in the code that contradicted itself may have caused the inconsistency. Accordingly, during the trials that the servo and motors acted when in the correct positions on the track, the servo-brake performed as expected, abruptly stopping the AEV at 390 marks, precisely in the desired region. This was in comparison to the AEV without the servo brake which experienced slight variation due to drift from propulsion braking. Although the team dialed in the code to exact specifications, the AEV would occasionally be inconsistent and drift to trigger the gate to close. Accordingly, it was also observed that the use of the servo-brake supports the goals of the MCR, making the model more efficient. As predicted, the brake allows less power to be expended for propulsive braking then if it was not present. From the observations observed, the team was convinced that the servo-brake will aid in completing the goals of the MCR and thus would move forward with the new model. In future lab analysis the team has intended to finalize the arduino code used with the servo-brake implicated, ensuring no lines of code contradict themselves. Also, various hardware components (reflectance sensors & servo motor) will be tested to ensure hardware issues are not causing the inconsistency in the AEV performing actions at designated portions of the track. The team will also aim to complete the arduino code to move the AEV through its full path of travel, as only the trip to the R2 unit was performed before. Looking ahead to performance test 2, the team will aim to produce the most efficient version of this code possible.

Appendix

Gantt Chart	No.	Task	Start	Finish	Due Date	Est. Time	Ryan Devine	Rob McEwan	Holly van der Lans	Stephanie Smithson	% Complete
Week 2	1	Build Sample AEV	21-Jan	23-Jan	24-Jan	1 hr	0.25	0.25	0.25	0.25	100
	2	Week 2 Progress Report	21-Jan	23-Jan	24-Jan	4 hrs	0.25	0.25	0.25	0.25	100
	3	Portfolio Update	21-Jan	23-Jan	24-Jan	1 hr	0.25	0.25	0.25	0.25	100
	4	Lab 2 Final Arduino Code	21-Jan	23-Jan	24-Jan	1 hr	0	0	0	1	100
Week 3	5	Brainstorm Design	21-Jan	20-Jan	21-Jan	1 hr	0.25	0.25	0.25	0.25	100
	6	Week 3 Progress Report	31-Jan	6-Feb	7-Feb	4 hrs	0.25	0.25	0.25	0.25	100
	7	Portfolio Update	29-Jan	6-Feb	7-Feb	1 hr	0.25	0.25	0.25	0.25	100
	8	Lab 3 Final Arduino Code	31-Jan	31-Jan	7-Feb	1 hr	0	0	1	0	100
Week 4	9	AEV Design and Implementation	4-Feb	4-Feb	7-Feb	2 hrs	0.25	0.25	0.25	0.25	100
	10	Portfolio Update	4-Feb	8-Feb	7-Feb	3 hrs	0	1	0	0	100
	11	Sketch Final Design	4-Feb	4-Feb	7-Feb	2 hrs	0	1	0	0	100
	12	Lab 4 Final Arduino Code	4-Feb	4-Feb	7-Feb	1 hrs	0	0	0	1	100
	13	Build Final AEV Design	4-Feb	6-Feb	7-Feb	1 hr	0.5	0.5	0	0	100
	14	Week 4 Progress Report	4-Feb	6-Feb	7-Feb	4 hrs	0.25	0.25	0.25	0.25	100
Week 5	15	Fill out screening and scoring	14-Feb	20-Feb	21-Feb	1 hr	0.25	0.25	0.25	0.25	100
	16	Code the AEV for the comparison run	14-Feb	20-Feb	21-Feb	0.5 hrs	0	0	0	1	100
	17	Progress report 5	14-Feb	20-Feb	21-Feb	4 hrs	0.25	0.25	0.25	0.25	100
	18	Build AEV with new design	14-Feb	20-Feb	21-Feb	2 hrs	0.5	0.5	0	0	100
Week 6	19	Analyze scoring and screening sheets	18-Feb	24-Feb	28-Feb	0.5 hrs	0.25	0.25	0.25	0.25	0
	20	Run AEV on track for updated data	18-Feb	21-Feb	28-Feb	0.5 hrs	0.25	0.25	0.25	0.25	0
	21	Brainstorm 3d Design	18-Feb	21-Feb	28-Feb	0.5 hrs	0.25	0.25	0.25	0.25	0
	22	Redesign Arduino Code	18-Feb	27-Feb	28-Feb	1 hr	0.25	0.25	0.25	0.25	0
	23	Progress Report 6	26-Feb	27-Feb	28-Feb	5 hrs	0.25	0.25	0.25	0.25	0
Week 8	24	Progress Report 8	28-Feb	9-Mar	10-Mar	5 hrs	0.25	0.25	0.25	0.25	0
	25	Design full arduino code	28-Feb	9-Mar	10-Mar	2 hrs	0	0	0.5	0.5	0
	26	Design two functional deisgns	28-Feb	9-Mar	10-Mar	3 hrs	0.5	0.5	0	0	0
	27	Performace Tests 1	10-Mar	10-Mar	10-Mar	2 hrs	0.25	0.25	0.25	0.25	0
Week 9	28	Progress Report 9	11-Mar	23-Mar	24-Mar	5 hrs	0.25	0.25	0.25	0.25	0
	29	Design code to test in lab 9A	23-Mar	23-Mar	24-Mar	2 hrs					0
	30	Preliminary Design Report due 9B	1-Mar	26-Mar	27-Mar	8 hrs	0.25	0.25	0.25	0.25	0
	31	Performace Test 2	24-Mar	24-Mar	24-Mar	2 hrs	0.25	0.25	0.25	0.25	0
Week 10	32	Progress Report 10	1-Apr	2-Apr	3-Apr	4 hrs	0.25	0.25	0.25	0.25	0
	33	Performace Test 3	3-Apr	3-Apr	3-Apr	2 hrs	0.25	0.25	0.25	0.25	0
Week 11	34	Progress Report 11	7-Apr	10-Apr	10-Apr	3 hrs	0.25	0.25	0.25	0.25	0
	35	Performace Test 4	10-Apr	10-Apr	10-Apr	2 hrs	0.25	0.25	0.25	0.25	0
	36	Run final design	17-Apr	17-Apr	17-Apr	1 hrs	0.25	0.25	0.25	0.25	0
Week 12	37	Critical Design Review	17-Apr	20-Apr	21-Apr	10 hrs	0.25	0.25	0.25	0.25	0
	38	CDR Oral Presentation	10-Apr	20-Apr	21-Apr	6 hrs	0.25	0.25	0.25	0.25	0

Figure 6: Team Schedule

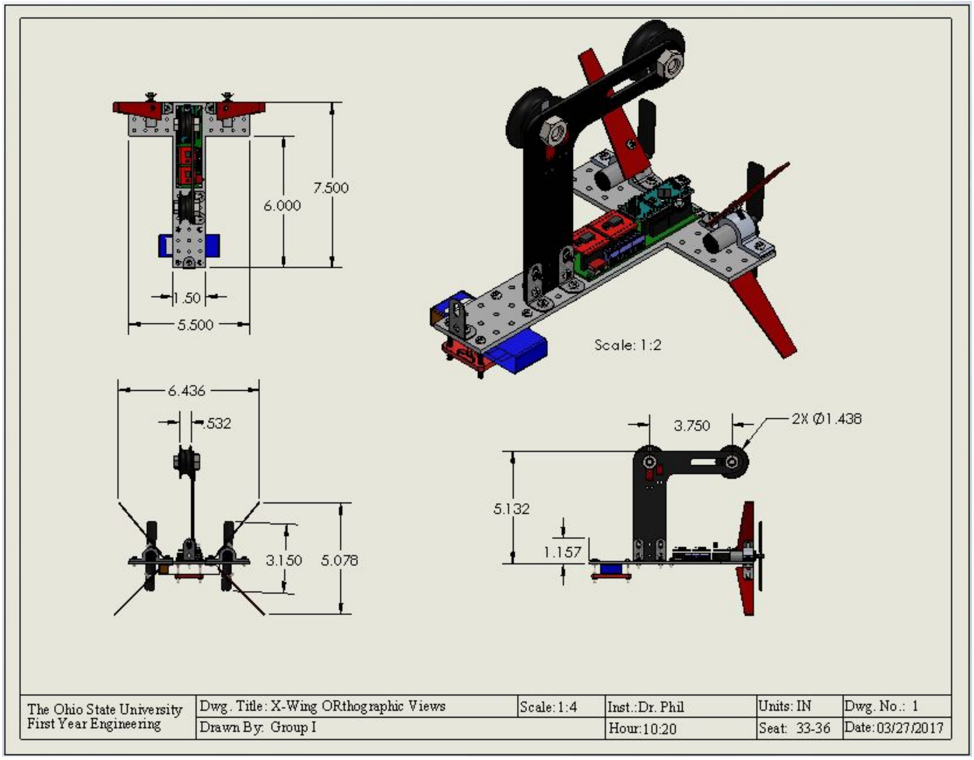
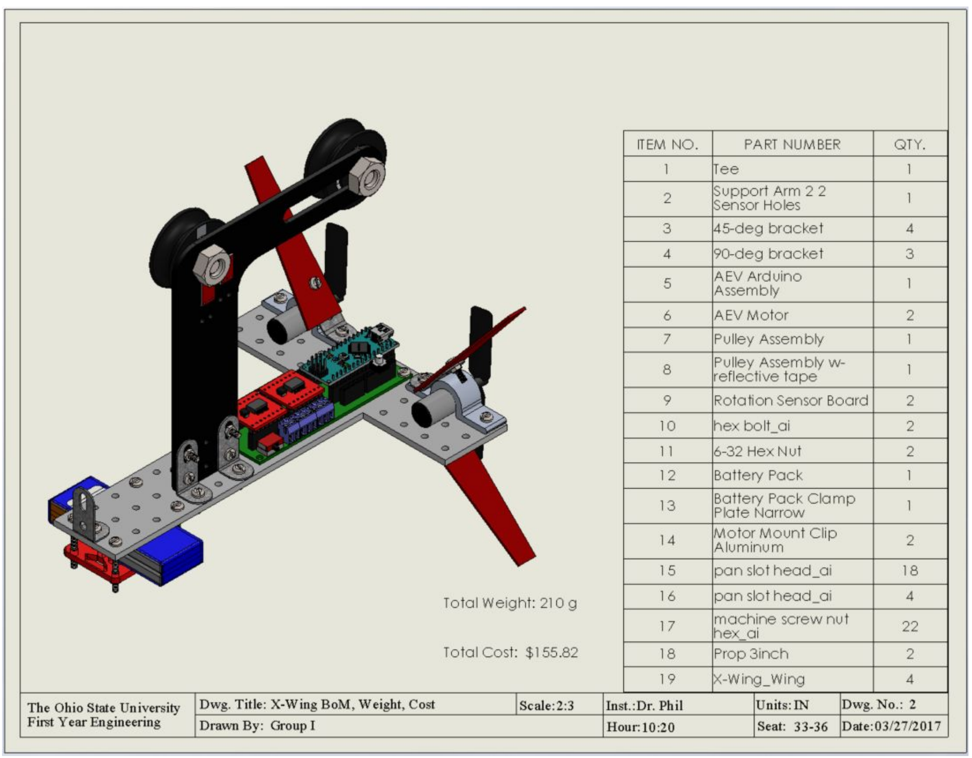


Figure 7: X-Wing Orthographic Views w/ Dimensions



ITEM NO.	PART NUMBER	QTY.
1	Tee	1
2	Support Arm 2 2 Sensor Holes	1
3	45-deg bracket	4
4	90-deg bracket	3
5	AEV Arduino Assembly	1
6	AEV Motor	2
7	Pulley Assembly	1
8	Pulley Assembly w-reflective tape	1
9	Rotation Sensor Board	2
10	hex bolt_ai	2
11	6-32 Hex Nut	2
12	Battery Pack	1
13	Battery Pack Clamp Plate Narrow	1
14	Motor Mount Clip Aluminum	2
15	pan slot head_ai	18
16	pan slot head_ai	4
17	machine screw nut hex_ai	22
18	Prop 3inch	2
19	X-Wing_Wing	4

Figure 8: X-Wing w/ Bill of Materials, Weight, and Cost

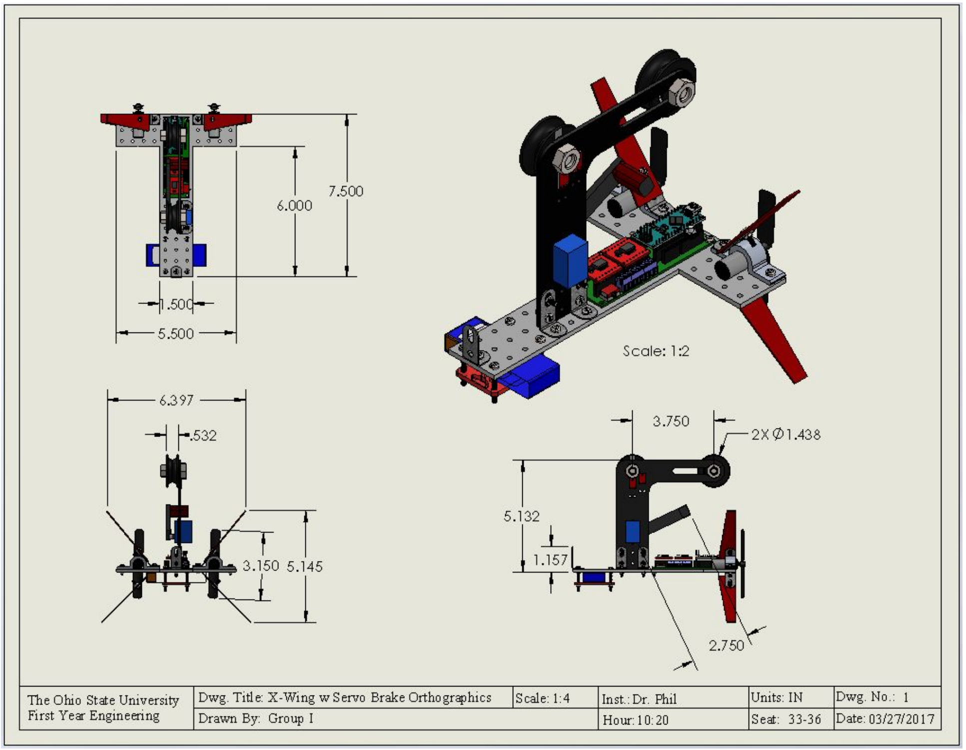
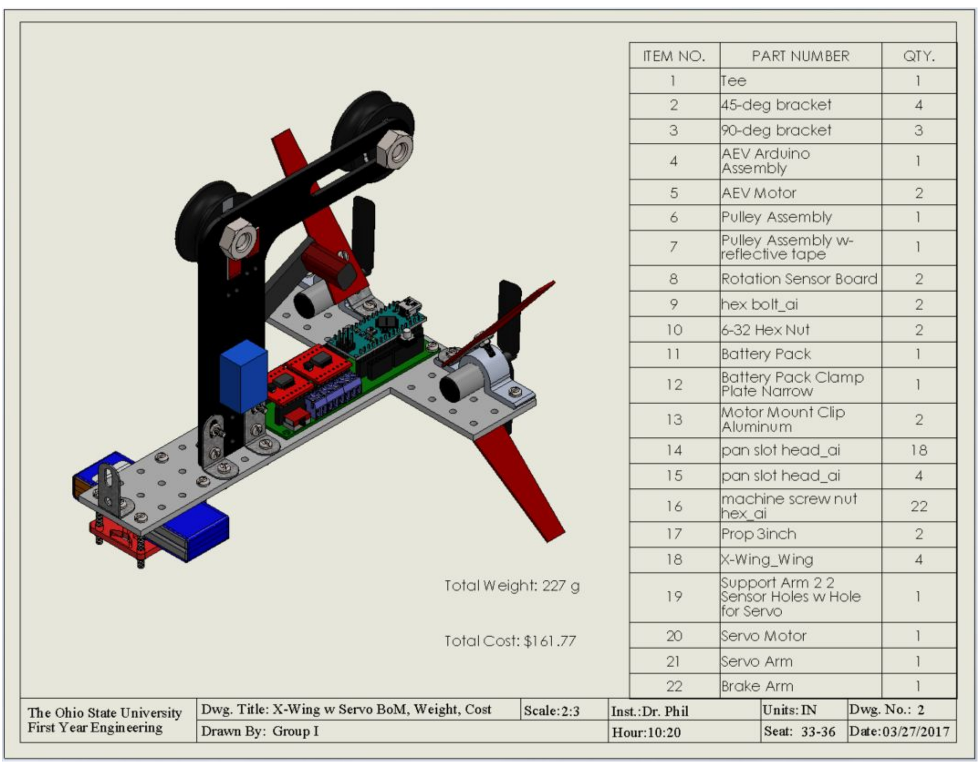


Figure 9: X-Wing w/ Servo Brake Orthographic Views w/ Dimensions



ITEM NO.	PART NUMBER	QTY.
1	Tee	1
2	45-deg bracket	4
3	90-deg bracket	3
4	AEV Arduino Assembly	1
5	AEV Motor	2
6	Pulley Assembly	1
7	Pulley Assembly w-reflective tape	1
8	Rotation Sensor Board	2
9	hex bolt_ai	2
10	6-32 Hex Nut	2
11	Battery Pack	1
12	Battery Pack Clamp Plate Narrow	1
13	Motor Mount Clip Aluminum	2
14	pan slot head_ai	18
15	pan slot head_ai	4
16	machine screw nut hex_ai	22
17	Prop 3inch	2
18	X-Wing_Wing	4
19	Support Arm 2 2 Sensor Holes w Hole for Servo	1
20	Servo Motor	1
21	Servo Arm	1
22	Brake Arm	1

Figure 10: X-Wing w/ Servo Brake w/ Bill of Materials, Weight, and Cost