Critical Design Report

Submitted to:
Inst. Dr. Phil Schlosser
GTA Rahel Beyene

Created by:
Team P

Kenny Roback
Sam Stegman
Caleb Woy
Matt Heffinger

Engineering 1182
The Ohio State University
Columbus, OH
21 April 2017

Executive Summary

The AEV project was needed to assist the rebel alliance in their war effort against the galactic empire. The rebel alliance was hiding out on remote planets to keep their operations secret from the empire. AEVs were used to transport R2D2 units, which were being constructed on one side of the planet, to another part of the planet where interceptor aircrafts were being constructed. Since the alliance was hiding out on a remote planet power was very limited. This mean't that along with being operationally consistent the AEVs also had to be energy efficient. Essentially the team was trying to decrease the energy/mass ratio as much as possible without compromising its operational consistency. Due to the planet having shifting faults the track developed variations over time so the final design did not depend on a certain track. That was taken into account in the code by referencing the AEVs absolute position.

Many methods of research were used to create the most efficient vehicle possible. Research was done on propellers by placing different types in wind tunnels to determine which was most efficient. Energy graphs was a method of research that the team used to help determine what structural and coding practices were most efficient. Some educated guess work was also used in the begining to help determine the power levels needed for the AEV to reach the gate and package consistently.

In performance test one the team focused on creating two potential final concepts of their AEV. This was done in order to test energy usage and compare it to the distance the AEV was able to travel. The two designs used were a double pull and a push/pull system. The double pull system was more efficient but when it had to break or reverse direction it lost its advantage. To keep the double pull systems advantage, a servo was attached to the underside of the AEV to keep the motors rotated in the pull orientation. In performance test two, the team focused on using the code to increase the efficiency of the AEV. The team first tested the command celerate() vs the command motorSpeed(), and compared their energy graphs. Since distance was a constant with both codes stopping at the first gate the team quickly determined that motorSpeed() was more efficient since it used less power to travel the same distance. A lot of inconstancy was noted while the team was working on breaking the AEV, and it was eventually noted that if the servo rotates while still on a turn, the AEV would shake and lose a lot of momentum. The code was altered so that the servo would never rotate during a turn. In the third performance test the team worked on increasing the consistency of the runs. Two different checks were made using the AEVs absolute position to determine if the marks were increasing, decreasing, or stopped. One check looked at if it had stopped too soon, and would advance it until it tripped the sensor. The other check was used to increase the braking power in case the AEV was going too fast and going to trip the second sensor.

In future projects Team P recommends that stricter enforcement of ladder use for taking the AEV off the track is applied. When testing if the AEV were to get stuck in the middle of the track and or short or average height person tried to remove the AEV it sometimes wouldn't make it and the person would end up pulling down on the AEV causing major warping on the track. Some people noticed that from run to run their AEVs would go from being inches off to being over a foot off and this is what Team P believed caused it.

Table of Contents

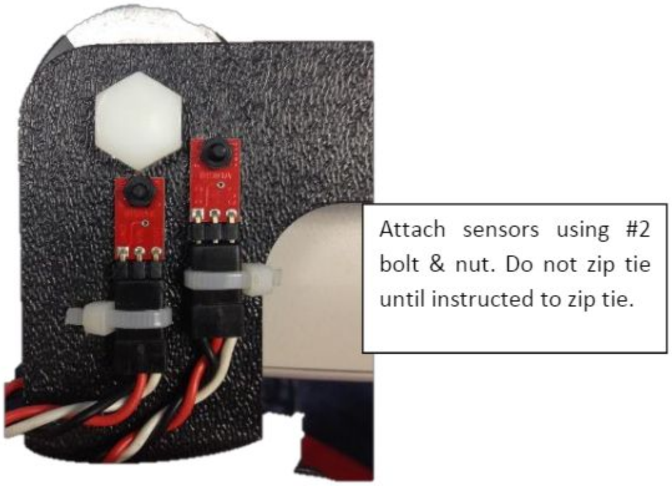
Introduction

In this semester long lab, Team P set out to efficiently transport the Galactic Empire's R2-D2 droid around the planet's surface using an AEV design centered around cost and energy efficiency. This included the careful planning and testing of multiple design options that Team P considered worthy of completing the task on hand. As Team P discovered more and more about efficiency and design details, a final prototype was eventually developed and used to successfully transport R2 safely. This report details aspects of multiple designs considered throughout the 12 weeks of lab, and the results of testing two final prototypes. It also explains what aspects of design the team emphasized and how they have been executed to create the best possible product.

Experimental Methodology

In Week 1, the team familiarized themselves with the AEV coding software by reviewing a brief list of functions that would be involved in coding. They then went on to write two simple programs that ran the motors for specific durations at specific speeds that sounded similar to the song March of the Empire.

In Week 2, the team installed the reflectance sensors on the base AEV as show in the picture below and ran the reflectance sensor test. This procedure involved using a provided program that reports the relative and absolute marks, and simply spinning the attached wheel to see that the sensors are recording.



Attach sensors using #2 bolt & nut. Do not zip tie until instructed to zip tie.

In Week 3, the team tested propulsion efficiency of multiple propellor orientations using the lab's wind tunnels. The propellers tested include the 3030 push/pull and the 2510 push/pull, the numbers in their names referring to the diameter and pitch of the propellers and push/pull being their thrust orientation. A push orientation faces opposite the direction of travel where a pull orientation faces into the direction of travel. The propellers were all placed in identical wind tunnels as shown below, and their thrust per increment of increased voltage was recorded. The team then calculated the propulsion efficiency, advance ratio, and power input using the equations below and the gathered data.

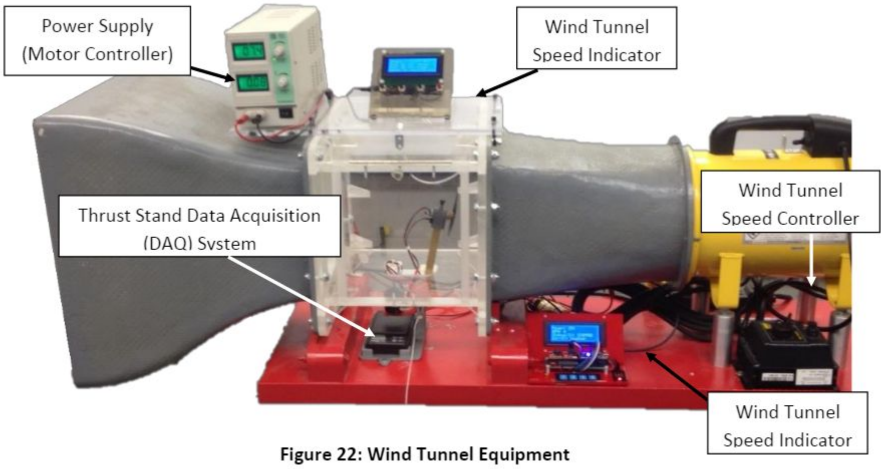


Figure 22: Wind Tunnel Equipment

$$Propulsion\ Efficiency:\ n_{sys} = (P_{out}/P_{in})\ x\ 100\%$$
$$Propeller\ Advance\ Ratio:\ J = v/((RPM/60) * D)$$
$$Power\ Input: P_{in} = V * I * (P_{\%}/100)$$

In Week 4, the team brainstormed ideas for the project and created four separate drawings of individual AEVs. The drawings included orthographic projections, dimensions, and a bill of materials. Three of such drawings are inserted below in the appendix (Figures 8, 9 & 10). The team then went on to create one final drawing combining the best aspects of each individual drawing for their AEV. This eventually lead to the creation of prototype 2 on page 9 .

In Week 5, the team ran comprehensive tests on the base AEV and variations of prototype 2 that included using only either the front or back motor. This included the use of the matLab data recorder data analysis tool to analyze runs down the straight track for each AEV with the same code. Once the run was completed, graphs for power vs time and power vs distance were created. One of which is provided in Figure 4 below. The graphs were then integrated using a matLab script similar to a loop that functions as a riemann sum to calculate the total energy for the run, and the total energy used during phases of the code. The equation this script was modelled from is produced below.

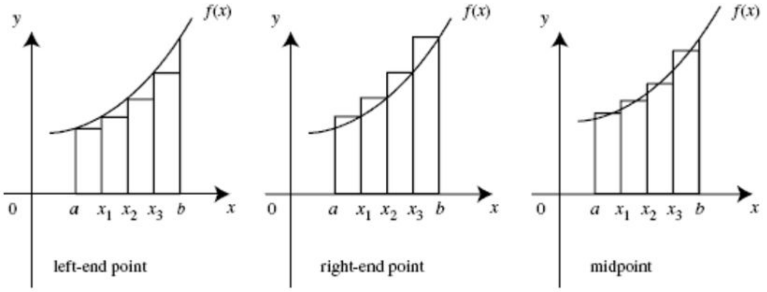


Figure 33: Rectangular Approximate Method

$$Incremental\ Energy : E_j = ((P_j + P_{j+1})/2) * (t_{j+1} - t_j)$$

$$Total\ Energy :\ E\ =\ \sum_{N=1}^{N-1} E_N$$

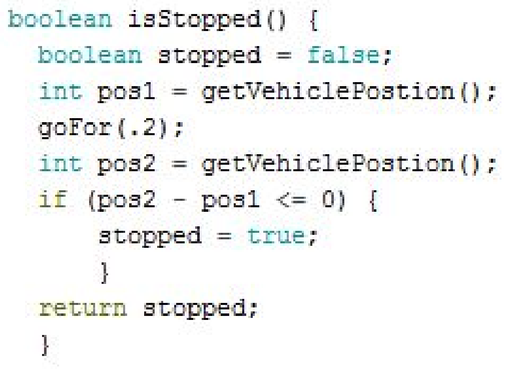
In Week 6, the team created a concept scoring chart, which is used to compare the different designs' strength and flaws to each other, and a screening matrix, which compares each design to the base AEV. It assigns it a score that is then weighted based on what aspects of design are most important to the team using the data from the energy analysis as a reference. Other aspects such as center of gravity, mass, blockage, maintenance, durability, and cost were also taken into account in order to give the team the best general sense of which design was best to continue testing. Examples of the final tables are found in the appendix, Tables 1 & 2.

In Week 7, the team tested using prototype 2 to ensure that the reflectance sensors were functioning properly. To do this they created a standardized code that simple started the motors at 30 percent power and ran them for 4 seconds, then compared the recorded marks to a prediction calculated in an excel worksheet. The results of this test, which can be seen in Tables 3 and 4 in the appendix below, made the team decided to change their design to allow for both motors to fit the pull configuration rather than 1 push and 1 pull. This was the genesis of prototype 1.

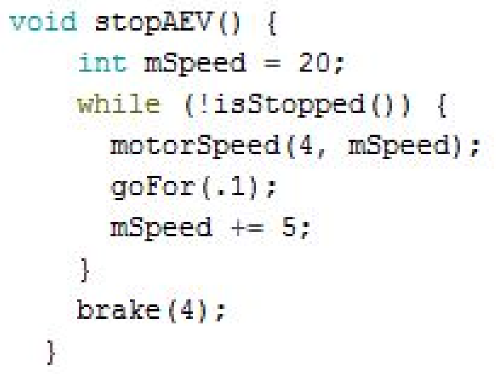
In Week 8, the team began their independent performance testing routine using prototype 1. This was somewhat delayed for Team P due to the team having to troubleshoot their reflectance sensors, and ultimately resulted in them having to be replaced after they spontaneously stopped reporting data. Once the new sensors were installed the team began testing. Their plan included slowly accelerating the AEV and coasting as much as possible to get to the first gate of the track. This took the team multiple iterations of coding to achieve; however, the end result was an acceleration in the beginning, coasting to an absolute position and then reversing the motors and applying backward thrust for approximately 1 second. This stopped the AEV at the gate sensor over multiple tests afterwards. The team then went on to estimate the time it took from their small thrust to slow down the AEV to the time the gate would be open. The next step for the team was to get the AEV from the gate to the cargo using as little energy as possible, which they accomplished by not braking after rotating the servo and lightly colliding with the cargo instead.

In week 9, the team focused on mirroring their strategy for the first two stages of the track to return the AEV back to the start. The marks used to time the rotation of the servo, and the brake were again found by downloading the Arduino data after a run and finding the distances. The AEV was again slowed down with a slight brake before the gate and wasn't slowed down when traveling back to the starting position. Most of the lab time this week was spent on making slight changes in the motor speed to find the lowest possible setting that still provided consistent results in order to conserve the most energy.

In the final weeks of lab, the team settled on the final design, which can be seen in Figure 12 in the appendix below. This design featured a servo that could rotate the motors 180 degrees to maintain the more efficient pull position. The team used the rotateServo() command while the AEV was coasting to use a pull direction during braking and rotated it back before the next stage. The servo also helped by allowing the AEV to stay in a pull orientation when travelling back from the R2D2 unit. The team was able to complete several successful runs with their AEV, but noted that as the charge level in the vehicle's battery decreased the AEV's performance did as well. In the final labs, the team decided to incorporate a check system using a while loop and an if statement after the initial planned brake that would ensure the AEV would stop if it overshot its destination and also begin moving again if it fell short. The team wrote two methods to help with this task, isStopped() and stopAEV(). The isStopped() method read one AEV position and another position after a short delay. If the difference of the positions was less than or equal to zero, the method returned true, indicating that the vehicle was stopped, or, more precisely, not actively moving forwards. The stopAEV() method worked by increasing the motor speed of the AEV by 5 with each loop iteration until isStopped() returned true, which provided a very precise and quick brake while only using as much motor speed as the vehicle needed to stop. The full code for the two methods is included below.



```
boolean isStopped() {
  boolean stopped = false;
  int pos1 = getVehiclePostion();
  goFor(.2);
  int pos2 = getVehiclePostion();
  if (pos2 - pos1 <= 0) {
      stopped = true;
      }
  return stopped;
  }
```



```
void stopAEV() {
    int mSpeed = 20;
    while (!isStopped()) {
      motorSpeed(4, mSpeed);
      goFor(.1);
      mSpeed += 5;
    }
    brake(4);
  }
```

Results

In weeks one and two of lab, Team P didn't collect any data that was all that important to the design process. The lab activities in weeks one and two were focused more on familiarizing the team with the lab equipment and objectives for the future.

The figure below displays data found in Week 3 of lab that Team P used to determine which propeller design/orientation was the most efficient. Based on the results of figure 1, Team P concluded that the EP-3030 propeller in the pull orientation was the most efficient since it created more thrust while at the same amount of power as the rest of the design/orientation combos. The 3030 propeller in the push orientation also produced more thrust than the 2510 propeller in both the push and pull orientation, so the 3030 push was also left on the drawing board to experiment with. In the Appendix on page 14, Figure 7 depicting the correlation of advance ratio to propulsion efficiency displays more data Team P used to determine which propeller to use moving forward.

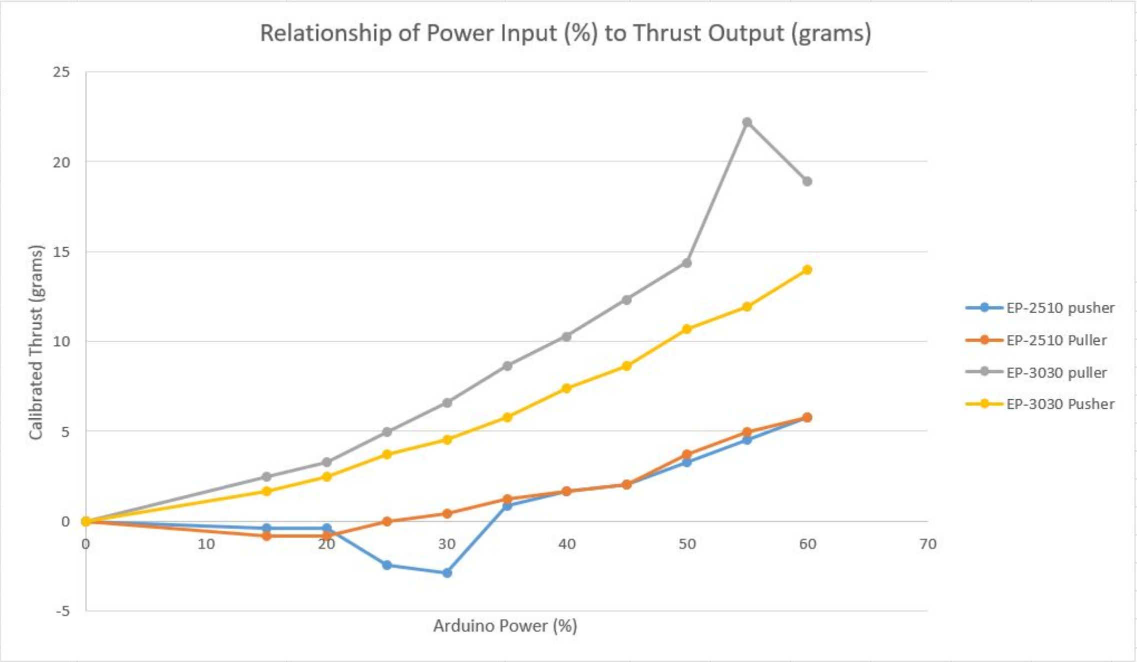


Figure 1:Power Input vs. Thrust Output

In week 4 of lab, Team P create four designs to potentially use in the final test which can be seen in Figures 8 & 9 of the Appendix below. These designs were then experimented with and changed a bit into two prototypes used in lab 8's performance test. Seen below in figures two and three are the two prototype concepts used in lab 8. Figure two below shows the double push or pull system and figure three depicts the push/pull combo system. Orthographic views with more details about each of the prototype designs can be found in the Appendix below (Figures 11 & 12).

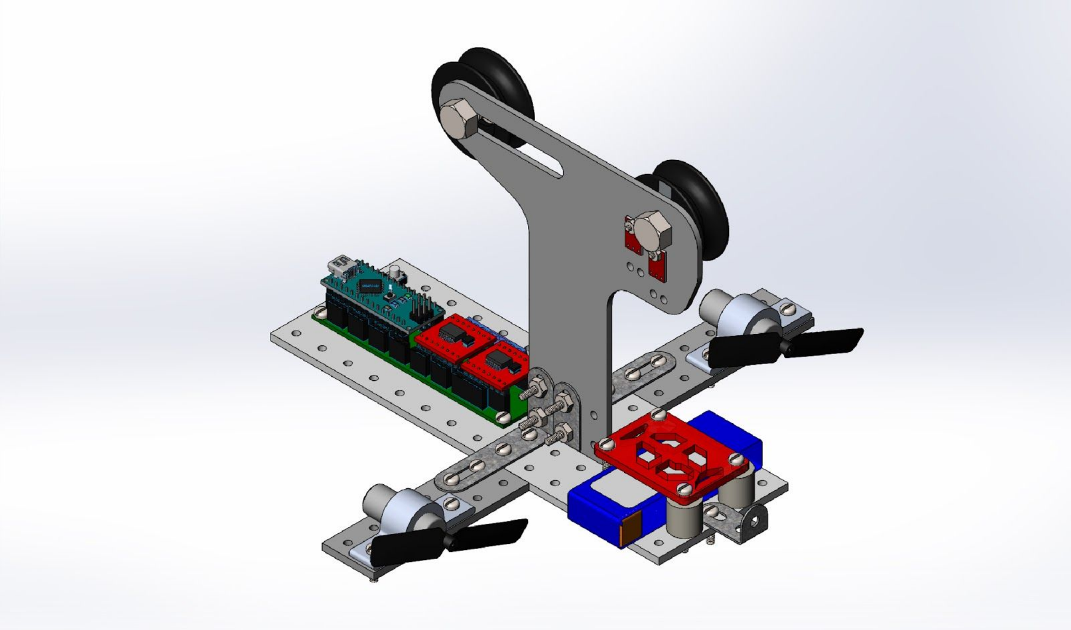


Figure 2: AEV Prototype 1

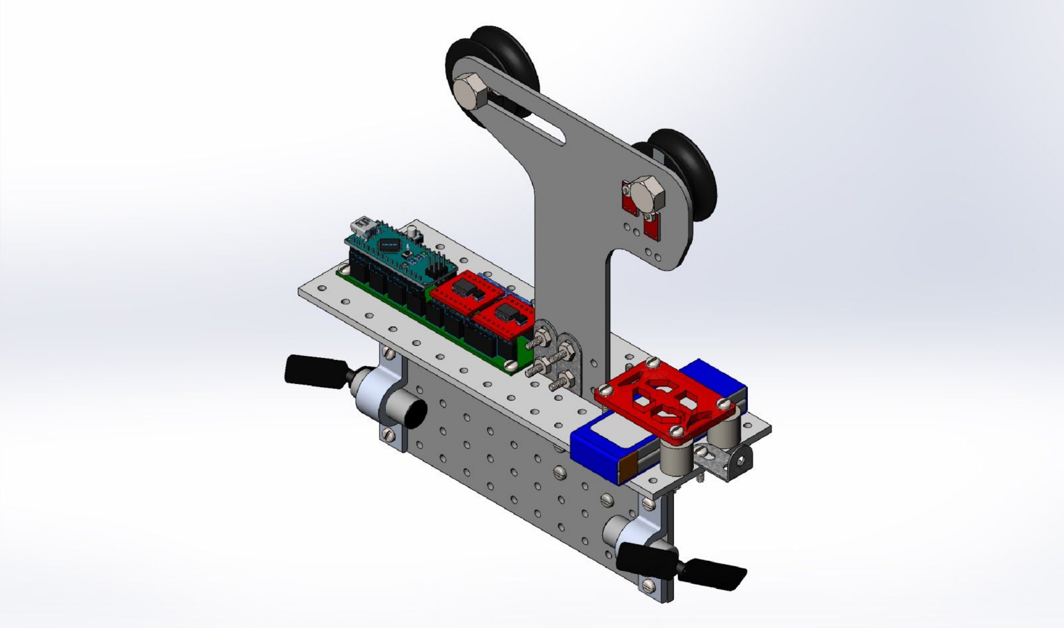
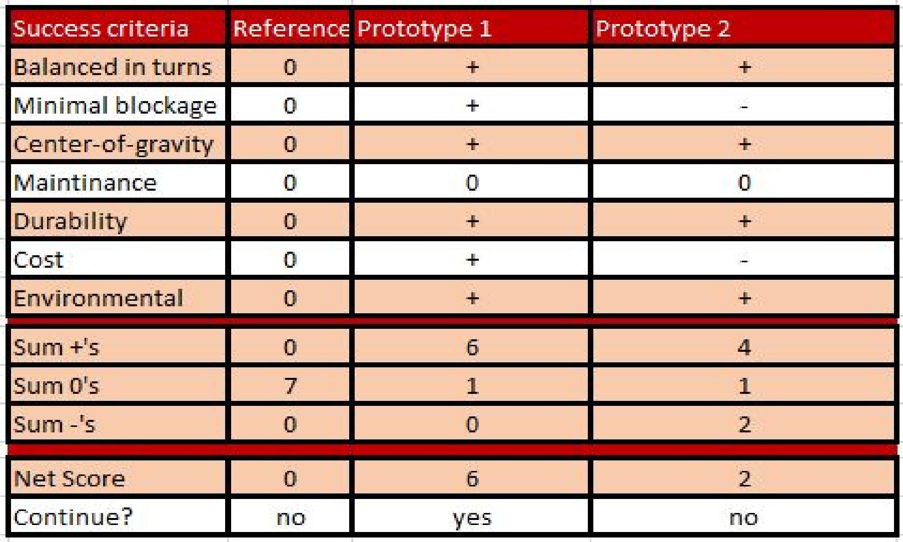


Figure 3: AEV Prototype 2

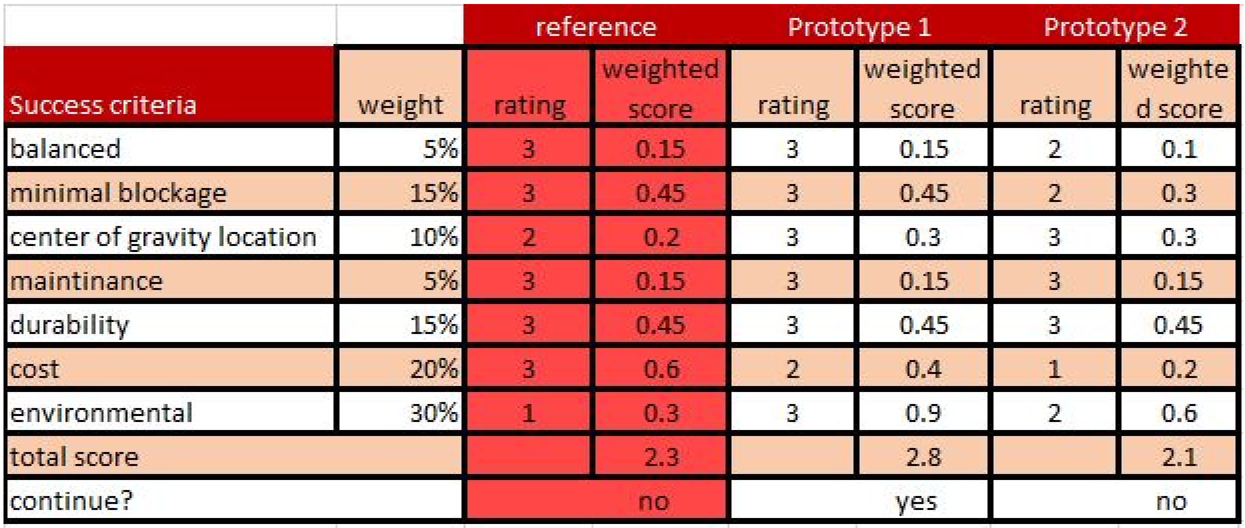
In Week 6 of lab, Team P created a screening score sheet and scoring matrix to compare different possible designs. The results from week 6 can be seen in tables 6 & 7 in the Appendix below. In lab 8 a new screening scoresheet and scoring matrix was created based on the prototypes from lab 8. They can be seen below as tables 1 and 2. The score sheet and scoring matrix in Week 6 favored the push/pull combo system compared to the other designs from then. From lab 8, the results of the new score sheet and scoring matrix below show the double push or pull system out scoring the push/pull combo system in both scenarios. On the scoresheet the double push or pull system received a net score of 6 compared to the push/pull combo system which totaled a 2. The scoring matrix showed the double push or pull scoring a 2.8 compared to the push/pull combo at a 2.1 and the reference AEV at a 2.3. Some of the improvements of Prototype 1 compared to Prototype 2 include less blockage of the motors with the double push or pull, and the design is cheaper than the push/pull combo.

Table 1: Concept screening Prototypes 1 & 2



Success criteria	Reference	Prototype 1	Prototype 2
Balanced in turns	0	+	+
Minimal blockage	0	+	-
Center-of-gravity	0	+	+
Maintinance	0	0	0
Durability	0	+	+
Cost	0	+	-
Environmental	0	+	+
Sum +'s	0	6	4
Sum 0's	7	1	1
Sum -'s	0	0	2
Net Score	0	6	2
Continue?	no	yes	no

Table 2: Concept scoring matrix Prototypes 1 & 2



		reference		Prototype 1		Prototype 2	
Success criteria	weight	rating	weighted score	rating	weighted score	rating	weighted score
balanced	5%	3	0.15	3	0.15	2	0.1
minimal blockage	15%	3	0.45	3	0.45	2	0.3
center of gravity location	10%	2	0.2	3	0.3	3	0.3
maintinance	5%	3	0.15	3	0.15	3	0.15
durability	15%	3	0.45	3	0.45	3	0.45
cost	20%	3	0.6	2	0.4	1	0.2
environmental	30%	1	0.3	3	0.9	2	0.6
total score			2.3		2.8		2.1
continue?		no		yes		no	

The results of the data collected in Week 8 by the EEPROM during the Performance Test can be seen below in Figure 4. It can be observed from the figure that the double push or pull system was more efficient than the Push/Pull combo. The two designs used about the same amount of power (about 7.5 W), but the double push or pull system traveled to around 3.2 meters while the motors were running which was almost a whole meter farther than the push/pull combo system in the same amount of time.

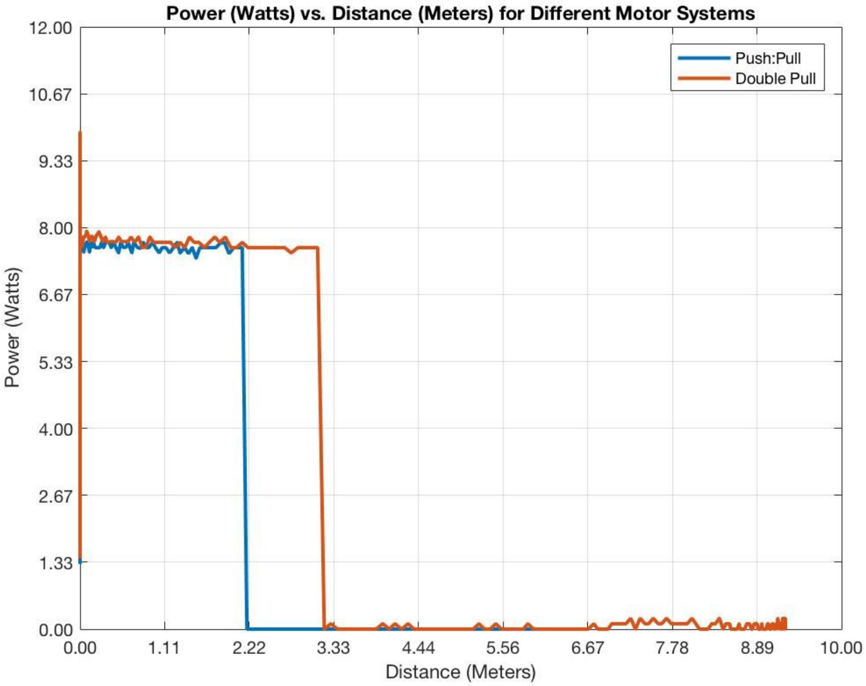


Figure 4: Power (Watts) vs. Distance (Meters) for Different Motor Systems

Tables 3 and 4 below show a breakdown of how much energy was used during each line of code for the Performance Test. It can be observed from the tables below that the double push or pull system required less energy to start the motors; however, the push/pull combo system required less overall energy. The double pull system did however travel a total distance of 7.97 meters which is much further than the push/pull combo system which only traveled 5.15 meters.

Table 3: Energy Breakdown Data for Double Pull System

Arduino Code	Total Energy (Joules)	Distance (m)
motorspeed(4,30)	1.84	0
goFor(4)	29.38	3.29
brake(4)	0	4.68
Totals	31.22	7.97

Table 4: Energy Breakdown Data for Push/Pull Combo System

Arduino Code	Total Energy (Joules)	Distance (m)
motorspeed(4,30)	2.25	0
goFor(4)	28.58	2.25
brake(4)	0	2.9
Totals	30.83	5.15

After seeing this data the team redesigned the AEV to include the servo used to rotate the propellers as an enhanced pull system. Before testing this concept on the track the team made calculations using the marks conversion and the diagram of the track provided in the lab manual. The conversion and table of marks estimates are provided below.

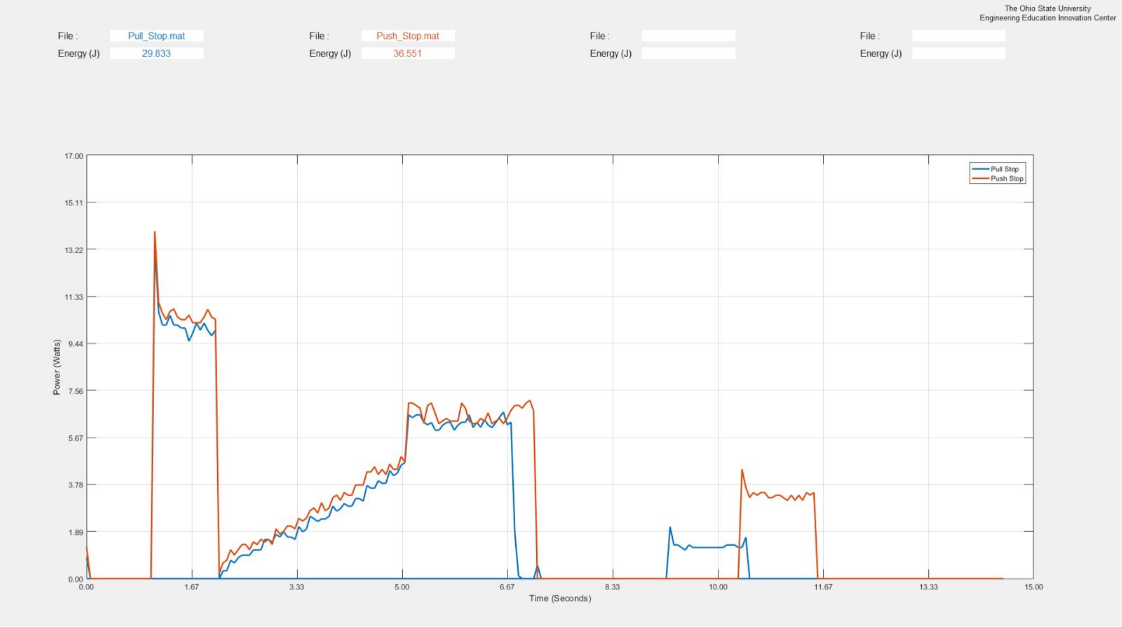
$$0.4875\ inches\ =\ 1\ mark$$

Table 5: Breakdown Distances

Track Stage	**Marks (Absolute)**	**Wait Time**
Start to gate	245 - brake motors and rotate servo 415 - engage motors to slow AEV to a stop	11 seconds at gate to wait for it to open and rotate servo
Gate to R2D2	724 - brake motors and rotate servo 872 - engage motors and slow AEV to a stop	5 seconds at end to rotate servo
R2D2 to gate	744 - brake motors and rotate servo 574 - engage motors and slow AEV to a stop	11 seconds at gate to wait for it to open and rotate servo
Gate to start	415 - engage motors to slow AEV to a stop 245 - brake motors and rotate servo	N/A

These calculations were used in the next phase to estimate when the rotations and braking procedures should be called. The team then went on to test whether a push brake or pull brake was more efficient.

Team P ran the AEV from the start to the first gate and tested the different codes. The energy output is pictured below on graph.



The Ohio State University
Engineering Education Innovation Center

File : Pull_Stop.mat
Energy (J) 29.833

File : Push_Stop.mat
Energy (J) 36.551

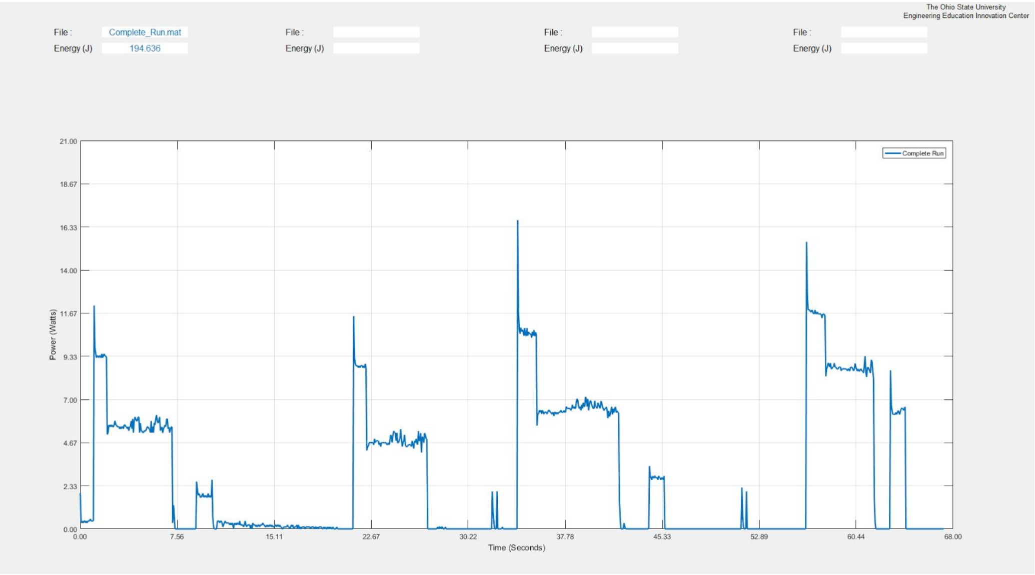
File :
Energy (J)

File :
Energy (J)

Figure 5: Power Use vs. Time for Push-Based Stop and Pull-Based Stop

This design was run on the track using the celerate() command to start the AEV moving and then coast it to the gate. The propellers were then rotated and a pull thrust was ran in the direction opposite of the travel to brake the AEV. This iteration was repeated for each section of the track. However the AEV did not function well using celerate() due to a dip in the beginning of the track. The team then decided to use motorspeed() instead for thrust. This ultimately used slightly more energy; however, it lead to improved function, and a quicker run than before.

During the next lab day the team focused on whether the braking procedure worked better with a non rotation and push brake or a rotation and pull brake. This variation was meant to reduce confusion in the runs caused by the rotation throwing off the center of balance. After multiple successful runs, the team found the rotation and pull thrust to be a better procedure. The balance problem caused by the rotation only was significant when the rotation occurred along the curve, and the pull orientation was already proven to be more effective in previous tests. The team also tested an iteration that eliminated the braking procedure before connecting to the cargo, and simply reduced the power input for the acceleration of the cargo. This can be seen on the energy input graph below, where the energy to break the AEV is absent following the second spike (about 30 seconds in).



The Ohio State University
Engineering Education Innovation Center

File : Complete_Run.mat
Energy (J) 194.636

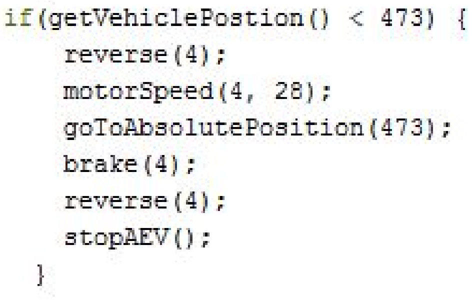
File :
Energy (J)

File :
Energy (J)

File :
Energy (J)

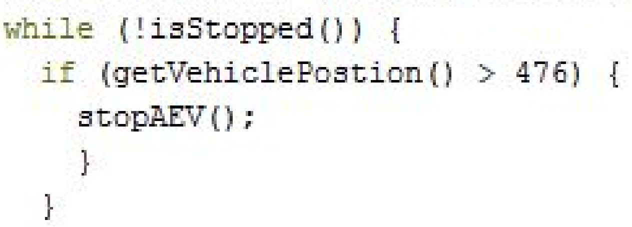
Figure 6: Power Usage vs. Time For a Complete Run

After this run was completed the team went on to improve the code’s accuracy when it came to a stopping procedure. Team P developed two checks. One that keeps the AEV moving if it stops short from a certain position. This code is included below:



```
if(getVehiclePostion() < 473) {
    reverse(4);
    motorSpeed(4, 28);
    goToAbsolutePosition(473);
    brake(4);
    reverse(4);
    stopAEV();
  }
```

Team P also developed a loop that keeps the AEV from travelling too far and tripping the second sensor. This was done by creating a method that checks if the AEV is still moving by repeatedly taking the position and subtracting it from a previous data point. If the result did not come out to zero then the AEV would be considered moving.



```
while (!isStopped()) {
  if (getVehiclePostion() > 476) {
    stopAEV();
    }
  }
```

Each of these segments were placed at the end of the sections that brought the AEV to the gate.

Discussion

The choice in propeller use was a very important part of the lab as a whole, because of how much more efficient the one is over the other. It was very important that good class data was taken during the lab in Week 3, and based on the results it was very easy to see that the 3030 propeller was more efficient all around. Team P had suspected that prior to the lab, and was already using the 3030 propeller in the very first runs. Another very important observation from Week 3 was determining whether using the propellers in a push or pull orientation was more efficient. Based on the results, Team P determined the pull orientation to be more efficient, and decided to focus on testing pull systems as much as possible.

Over the course of the lab, the design of Team P's AEV has been fairly similar to the two designs that Team P initially brainstormed in lab 4. The two designs developed in lab 4 can be seen in figures 6 and 7 below in the Appendix. Figure 6 displays the double push or pull system, and figure 7 displays the push/pull combo system. Other than these features that Team P has kept constant, the overall look of the designs from lab 4 have changed. Since lab 4, the double push or pull system (Figure 6 in the Appendix) was transformed into prototype 1 by moving the motors from the back of the vehicle more towards the front in order to create a more stable balanced design. There is also not a nose piece in the front which would increase the aerodynamics of the design, because the team has not received their 3D printed parts yet. Prototype 2 did not change much from the design in figure 7 of the Appendix. The team did however determine that it would be too hard to construct using a tube as a wind tunnel, so Team P decided to just use a platform already supplied to them to create the configuration.

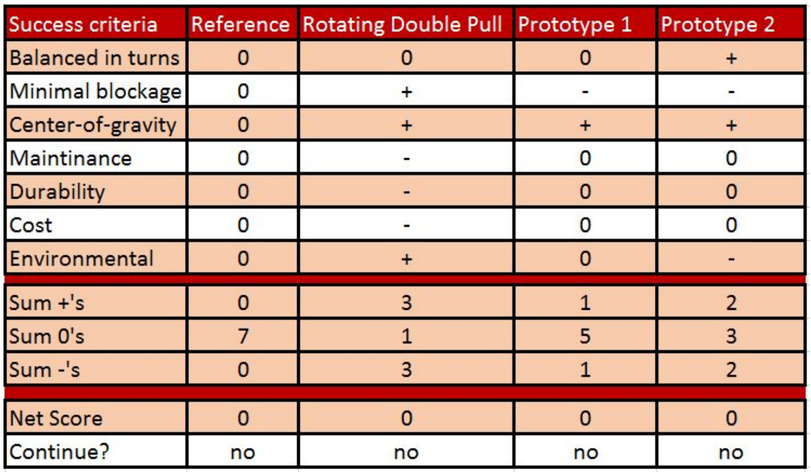
In week 6 of lab, Team P was able to conclude that two motors needed to be used. The score sheet and scoring matrix really helped with the decision. The designs with one motor did not receive a high enough score to really be considered anymore. After that, the push/pull combo system looked to be the favorite for the final design. In week 8 when the team decided to try a double pull for the performance test, it was a big turning point.

Based on test results from lab 8, Team P decided to focus more on the double push or pull system. When comparing the prototypes in the new scoresheet and scoring matrix the double push or pull outscores the push/pull combo. Based on observations of the data from lab 8, Team P was happy with the way the double push or pull system performed in the pull orientation. With the same code, the double pull traveled almost three meters further. However, the double pull system did use about 0.5 more joules of energy. Although, Team P was not too worried about that. The team believed that it had to do with a systematic error within the arduino or data recorder. In theory, they should have used the same amount of energy, because they were ran with the same code and each had two motors. Based on lab 8, Team P's design process was affected, because with the completion of the lab the team was able to move closer to choosing one final design. By observing how much further the vehicle was able to go in a double pull system specifically over the push/pull combo system, the team focused more of improving the double push or pull system.

During the weeks proceeding final testing the team continuously timed the AEV runs. Timing was of concern due to the coasting procedure taking longer than a continuously powered run. This downside was sacrificed though for the sake of less energy use. In theory, less power means less thrust, therefore less velocity and greater overall time. However, even using the coasting technique Team P was well below the time limit for the procedure on consecutive runs. This lead Team P to be able to focus mostly on stopping the AEV more accurately. This was also the team's most significant problems since the AEV was lighter than previous prototypes, it had less momentum and was therefore, more difficult to stop accurately.

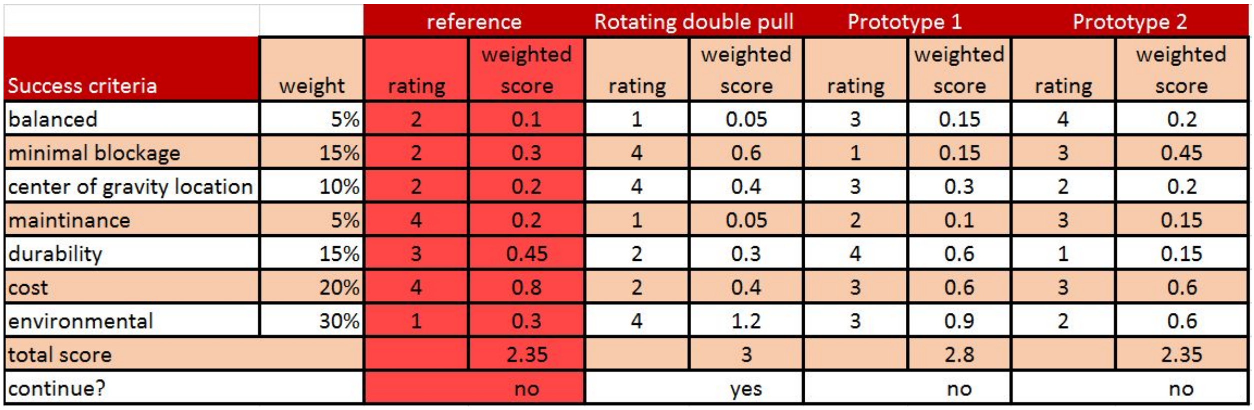
On the initial approach the AEV properly stopped, however, on the way back to the gate the AEV would not stop properly due to the isStopped() method checking if the marks were greater than or equal to zero. When the AEV is travelling backward to the start the marks are decreasing so a subtraction outputs a negative value when changing, throwing an error. Thus, resulting in the AEV having to be manually stopped a single time during final testing, and a loss of four points. All other AEV functions performed normally. This error was realized the day after final testing and corrected by creating a seperate isStopped method for the return journey. The final run lasted 63 seconds, used 178 Joules and had a Joules to Kilograms ratio of 635.

The final AEV was designed combining previous knowledge of the propeller orientation efficiency, variability, and the reduction of mass leading to lower energy use. The pull orientation being used in both directions is what ultimately lead to the energy usage being well below the class average. The team attempted to reduce the cost to manufacture the AEV by eliminating extra burdensome 3D printed parts and using tape instead of screws in some cases. The AEV's final cost was $178, which is higher than the previous prototypes tested but well worth the payoff when considered that the final energy usage was nearly half of the class average. The final scoring table and matrix are provided below, comparing the final design to the two prototypes that came before it and the reference AEV given in the beginning.



Success criteria	Reference	Rotating Double Pull	Prototype 1	Prototype 2
Balanced in turns	0	0	0	+
Minimal blockage	0	+	-	-
Center-of-gravity	0	+	+	+
Maintinance	0	-	0	0
Durability	0	-	0	0
Cost	0	-	0	0
Environmental	0	+	0	-
Sum +'s	0	3	1	2
Sum 0's	7	1	5	3
Sum -'s	0	3	1	2
Net Score	0	0	0	0
Continue?	no	no	no	no

Final Scoring Table



		reference		Rotating double pull		Prototype 1		Prototype 2	
Success criteria	weight	rating	weighted score	rating	weighted score	rating	weighted score	rating	weighted score
balanced	5%	2	0.1	1	0.05	3	0.15	4	0.2
minimal blockage	15%	2	0.3	4	0.6	1	0.15	3	0.45
center of gravity location	10%	2	0.2	4	0.4	3	0.3	2	0.2
maintinance	5%	4	0.2	1	0.05	2	0.1	3	0.15
durability	15%	3	0.45	2	0.3	4	0.6	1	0.15
cost	20%	4	0.8	2	0.4	3	0.6	3	0.6
environmental	30%	1	0.3	4	1.2	3	0.9	2	0.6
total score			2.35		3		2.8		2.35
continue?			no		yes		no		no

Final Scoring Matrix

The scoring table for this comparison can be misleading, highlighting the need for different methods of evaluation. Where all designs appear on even footing in the table, the matrix points out their benefits with the weighted system. Items such as balance in turns was accounted for by the team in the final design by rotating the prop after the curve. The drawbacks of maintenance and durability were simply an annoyance and mostly resulted from poor wiring. This didn't affect the performance of the AEV significantly. This could be accounted for in further improved designs by simply using more zip ties, reducing the problem entirely.

Conclusion and Recommendations

With the completion of the AEV lab, Team P gained valuable knowledge and experience that is needed to become a successful engineer. Team P learned the importance of attention to detail, and strengthened problem solving skills through lab procedures and trial and error practices. By working in a team the members of Team P were also able to understand the value of teamwork. The coding of the AEV to consistently transport R2-D2 on the track successfully really gave Team P a taste of how patient engineers must be. As Team P worked toward creating a vehicle that was as efficient as possible, many designs were tested. Throughout the lab, there were many times when problems arose. Team P resolved the reflectance sensor issue by eventually receiving new ones. Team P also resolved the stopping inconsistency issue by developing two checks in the code. One that kept the AEV moving if it stopped short of a certain position, and a loop that kept the AEV from travelling too far and tripping the second sensor at the gate. As Team P learned more and more through lab practices while using the design process and worked to constantly improve results, a final design eventually lead to the best energy/mass ratio in the class of 636 J/kg. In the future of the AEV lab, Team P recommends that the use of ladders around the track is strictly enforced at all times. This is because Team P observed that at times students that were not quite tall enough to reach their AEV on the track would try anyway causing them to pull down on the track as they would try to remove their vehicle. This allowed for unneeded potential damage/inconsistency of the track and skewed data. Overall, the AEV lab was beneficial to Team P in many ways, and many important engineering aspects were developed/strengthened by each member of the team with the completion of the lab.

Appendix

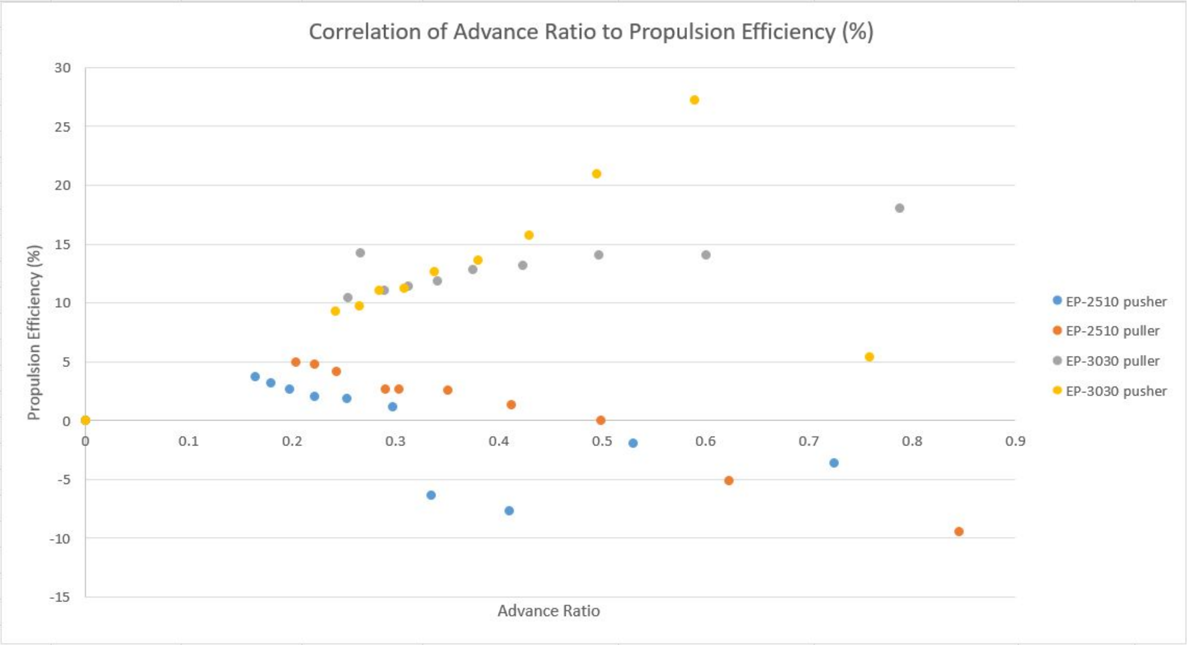


Figure 7:Correlation of Advance Ratio to Propulsion Efficiency

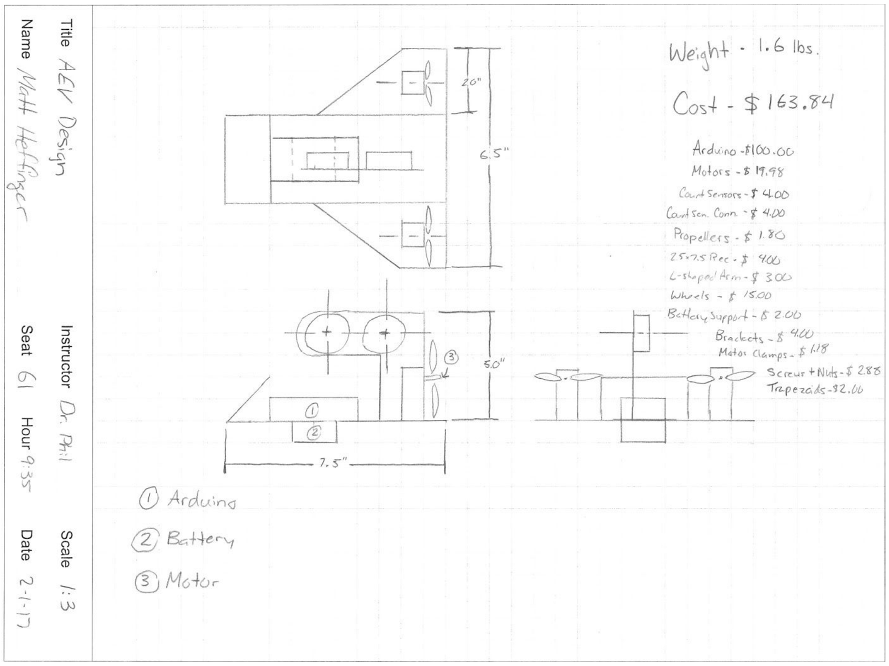


Figure 8: Double Push or Pull System

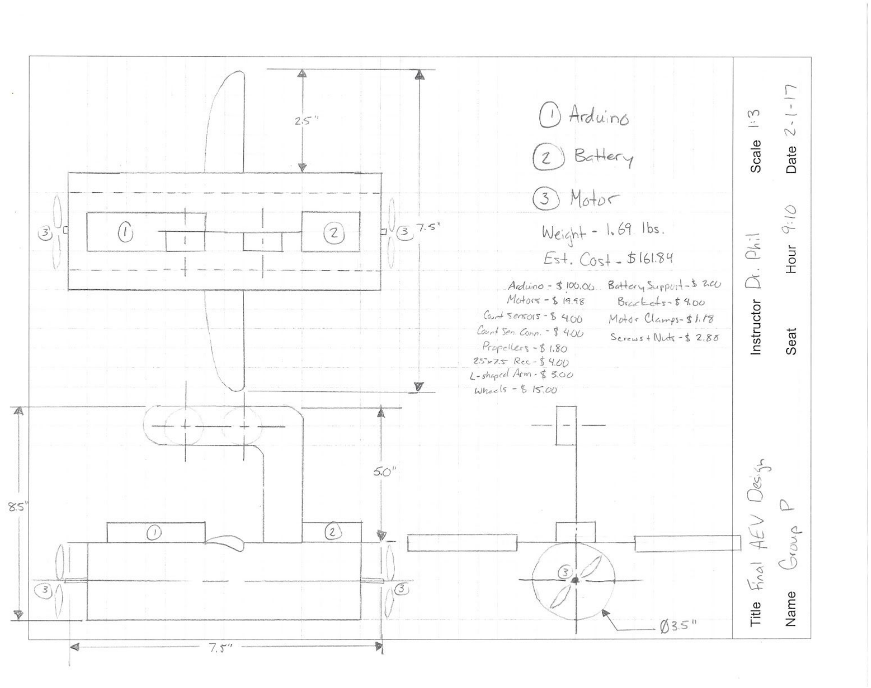


Figure 9: Push/Pull Combo System

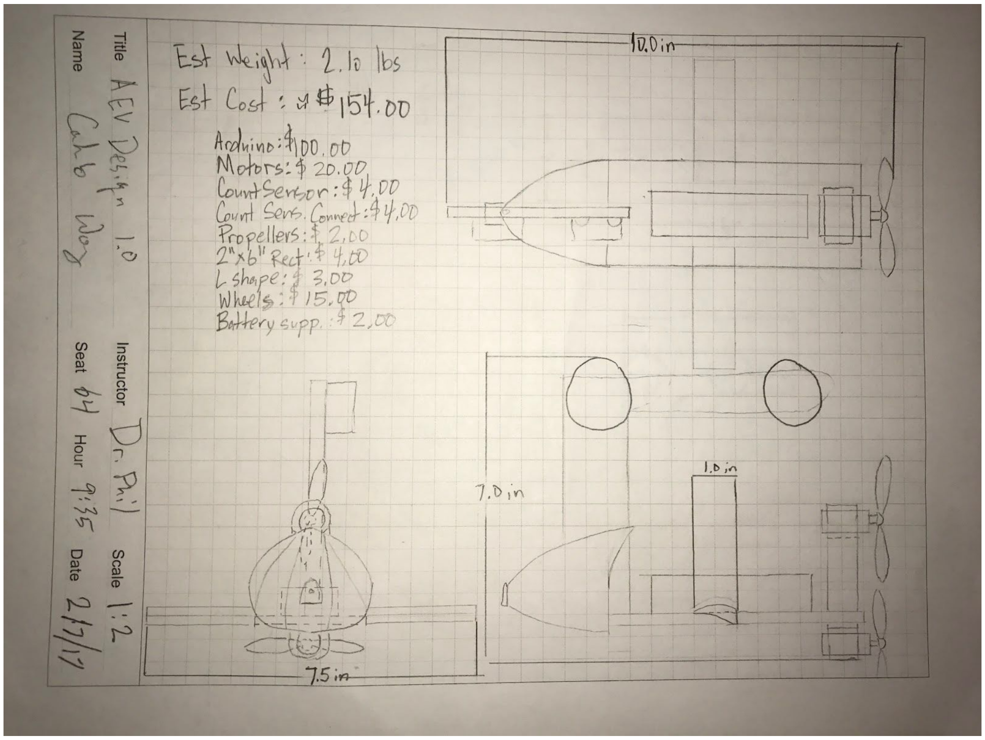
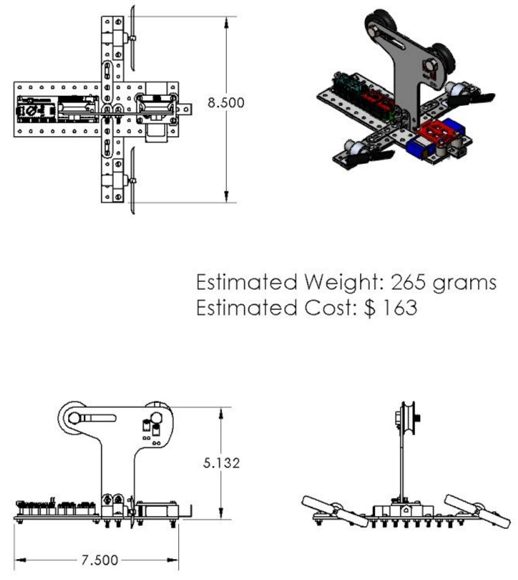


Figure 10:Design with Potential Wing

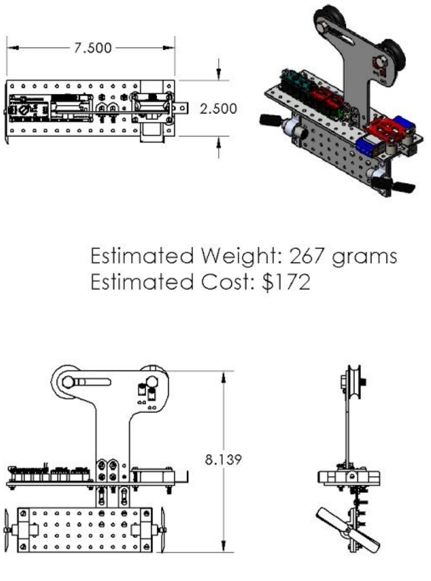


Estimated Weight: 265 grams
Estimated Cost: $ 163

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	5-16 - 18 Hex Nut		1
2	AEV Arduino Assembly		1
3	Tiny Rectangle		2
4	flat bracket		2
5	Motor Mount Clip Aluminum		2
6	AEV Motor		2
7	Prop 3inch		2
8	Battery Spacer		3
9	Battery Pack Clamp Plate Narrow		1
10	90-deg bracket		3
11	Support Arm 1 2 Sensor Holes V2		1
12	Housing w-reflective tape		1
13	Housing		1
14	hex bolt_ai		2
15	Rotation Sensor Board		2
16	pan slot head_ai		4
17	pan slot head_ai		19
18	flat head 100 screw_ai		2
19	pan slot head_ai		4
20	machine screw nut hex_ai		27
21	machine screw nut hex_ai		2
22	Battery Sleeve		1
23	Kapton Tape		1
24	machine screw nut hex_ai		2

The Ohio State University First Year Engineering	Dwg. Title: AEV Design 1	Scale: 1:4	Inst.: Dr. Phil	Units: inch	Dwg. No.: 1
	Drawn By: Team P		Hour: 9:10	Seat: P	Date: 3/27/17

Figure 11: AEV Prototype 1 Orthographic Views/Details



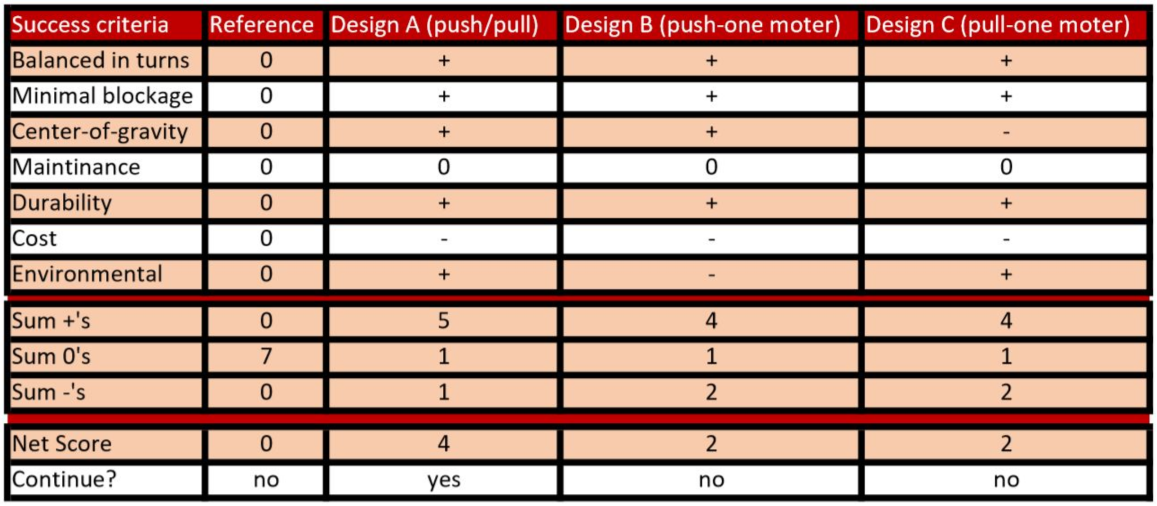
Estimated Weight: 267 grams
Estimated Cost: $172

BOM Table			
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	5-16 - 18 Hex Nut		1
2	AEV Arduino Assembly		1
3	Motor Mount Clip Aluminum		2
4	AEV Motor		2
5	Prop 3inch		2
6	Battery Spacer		3
7	Battery Pack Clamp Plate Narrow		1
8	90-deg bracket		5
9	Support Arm 1 2 Sensor Holes V2		1
10	Housing w-reflective tape		1
11	Housing		1
12	hex bolt_ai		2
13	Rotation Sensor Board		2
14	pan slot head_ai		4
15	pan slot head_ai		11
16	flat head 100 screw_ai		2
17	pan slot head_ai		4
18	machine screw nut hex_ai		15
19	machine screw nut hex_ai		2
20	Battery Sleeve		1
21	Battery		2
22	Kapton Tape		1
23	machine screw nut hex_ai		2
24	Medium Rectangle		1
25	flat bracket		2
26	SL-PHMS 0.112-40x0.5x0.5-S		10
27	MSHXNUT 0.112-40-S-S		14

The Ohio State University First Year Engineering	Dwg. Title: AEV Design 2	Scale: 1:4	Inst.: Dr. Phil	Units: inch	Dwg. No.: 2
	Drawn By: Team P		Hour: 9:10	Seat: P	Date: 3/27/17

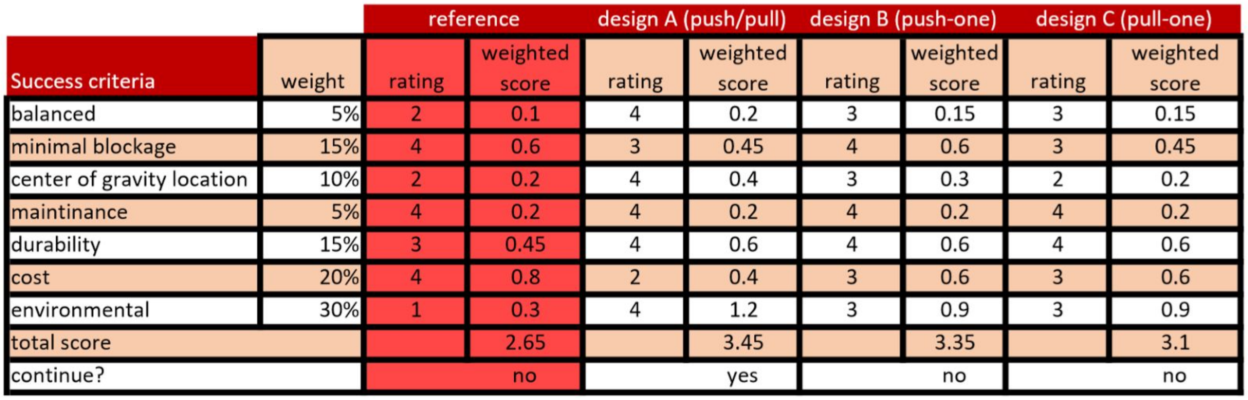
Figure 12: AEV Prototype 2 Orthographic Views/Details

Table 6: Concept Screening Scoresheet Week 6

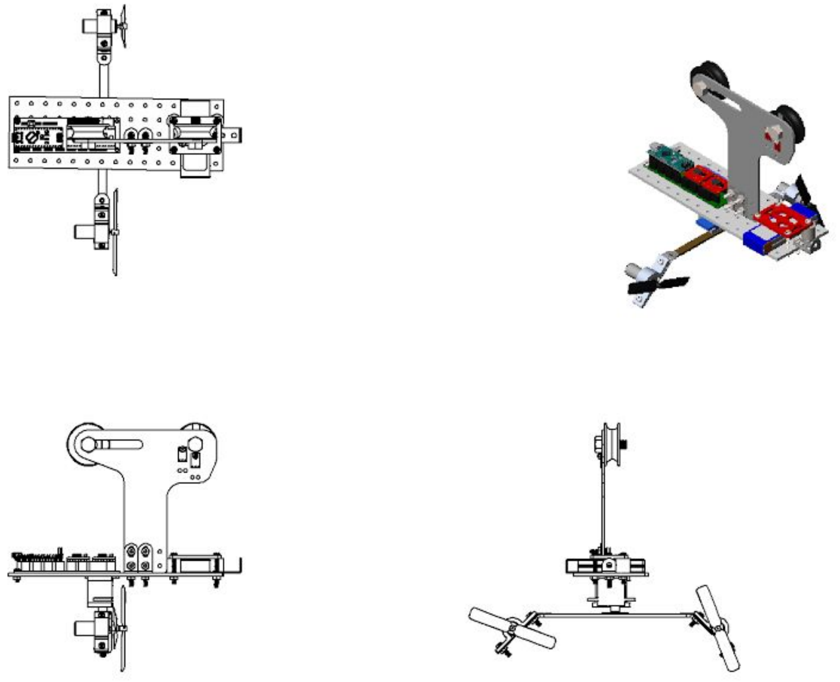


Success criteria	Reference	Design A (push/pull)	Design B (push-one moter)	Design C (pull-one moter)
Balanced in turns	0	+	+	+
Minimal blockage	0	+	+	+
Center-of-gravity	0	+	+	-
Maintinance	0	0	0	0
Durability	0	+	+	+
Cost	0	-	-	-
Environmental	0	+	-	+
Sum +'s	0	5	4	4
Sum 0's	7	1	1	1
Sum -'s	0	1	2	2
Net Score	0	4	2	2
Continue?	no	yes	no	no

Table 7: Concept Scoring Matrix Week 6

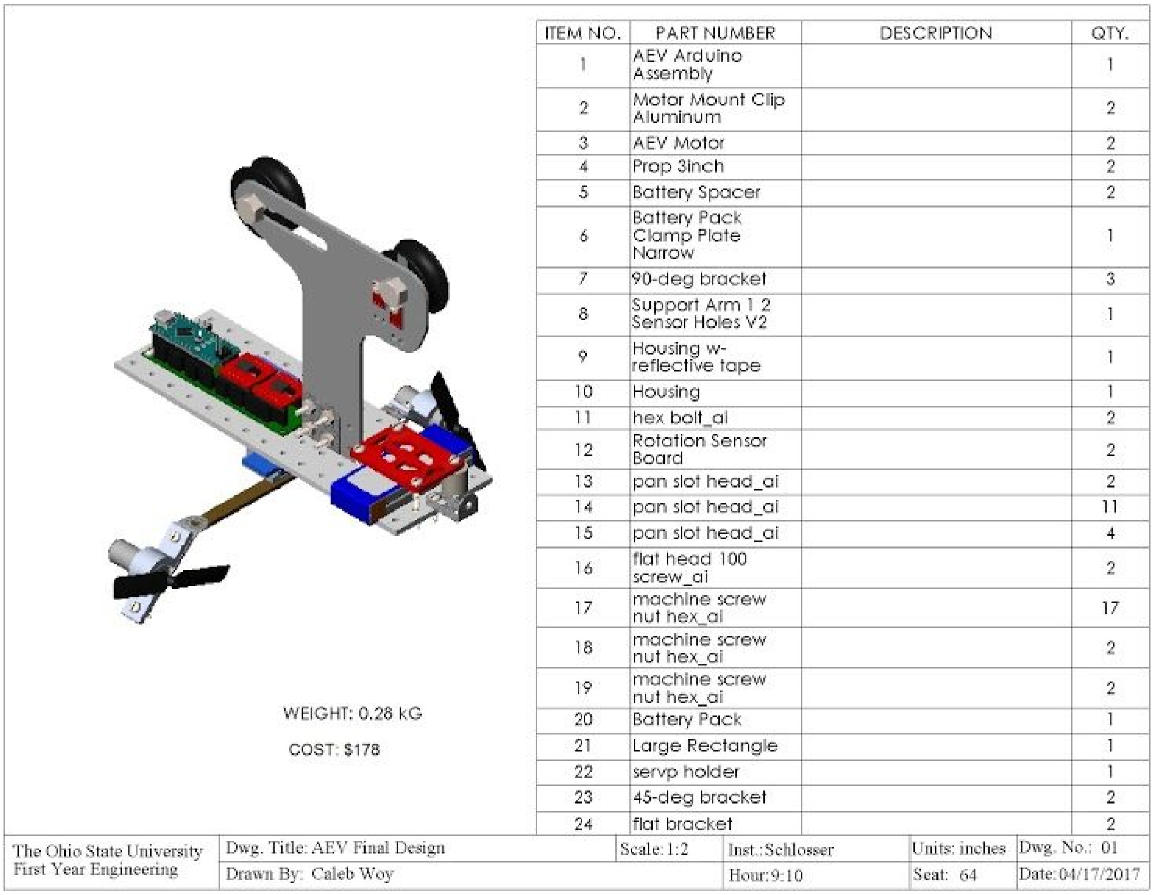


		reference		design A (push/pull)		design B (push-one)		design C (pull-one)	
Success criteria	weight	rating	weighted score	rating	weighted score	rating	weighted score	rating	weighted score
balanced	5%	2	0.1	4	0.2	3	0.15	3	0.15
minimal blockage	15%	4	0.6	3	0.45	4	0.6	3	0.45
center of gravity location	10%	2	0.2	4	0.4	3	0.3	2	0.2
maintinance	5%	4	0.2	4	0.2	4	0.2	4	0.2
durability	15%	3	0.45	4	0.6	4	0.6	4	0.6
cost	20%	4	0.8	2	0.4	3	0.6	3	0.6
environmental	30%	1	0.3	4	1.2	3	0.9	3	0.9
total score			2.65		3.45		3.35		3.1
continue?			no		yes		no		no



The Ohio State University First Year Engineering	Dwg. Title: AEV Final Design	Scale: 1:4	Inst.:Schlosser	Units: inches	Dwg. No.: 01
	Drawn By: Caleb Woy		Hour:9:10	Seat: 64	Date:04/17/2017

Figure 12: Final Design



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	AEV Arduino Assembly		1
2	Motor Mount Clip Aluminum		2
3	AEV Motor		2
4	Prop 3inch		2
5	Battery Spacer		2
6	Battery Pack Clamp Plate Narrow		1
7	90-deg bracket		3
8	Support Arm 1 2 Sensor Holes V2		1
9	Housing w-reflective tape		1
10	Housing		1
11	hex bolt_ai		2
12	Rotation Sensor Board		2
13	pan slot head_ai		2
14	pan slot head_ai		11
15	pan slot head_ai		4
16	flat head 100 screw_ai		2
17	machine screw nut hex_ai		17
18	machine screw nut hex_ai		2
19	machine screw nut hex_ai		2
20	Battery Pack		1
21	Large Rectangle		1
22	servp holder		1
23	45-deg bracket		2
24	flat bracket		2

WEIGHT: 0.28 kG

COST: $178

The Ohio State University First Year Engineering	Dwg. Title: AEV Final Design	Scale: 1:2	Inst.:Schlosser	Units: inches	Dwg. No.: 01
	Drawn By: Caleb Woy		Hour:9:10	Seat: 64	Date:04/17/2017

Figure 13: Final Bill of Materials

Table 7: Project Schedule

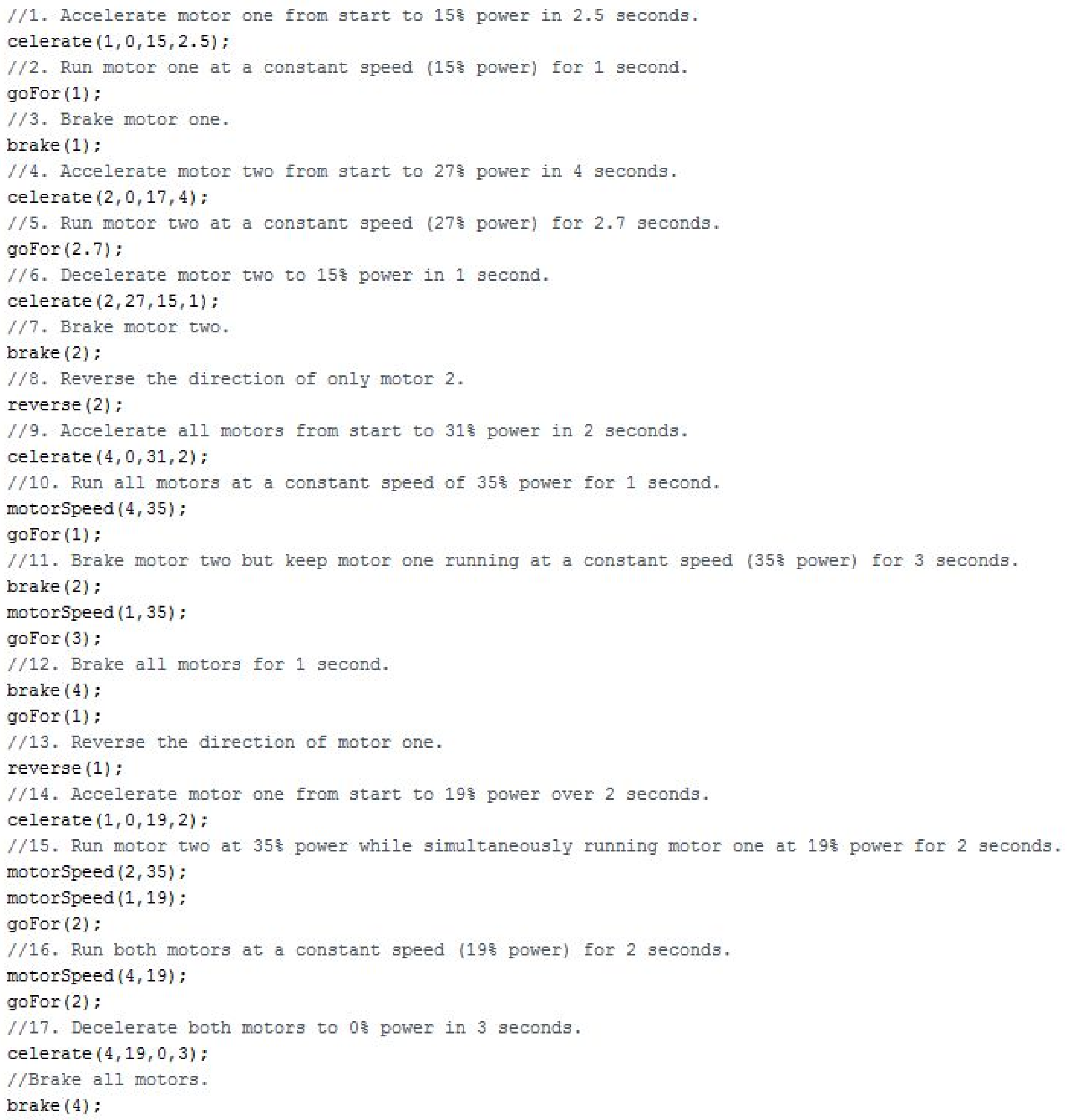
Task	Start Date	Finish Date	Team Members	Est. Hrs
Lab 1	1/18/17	1/18/17	All	1 hr
Progress report Week 2	1/22/17	1/22/17	All	3 hrs
Lab 2	1/25/17	1/25/17	All	1 hr
Progress report Week 3	1/27/17	1/29/17	All	3 hrs
Lab 3	2/1/17	2/1/17	All	1 hr
Revise Progress Week 2	2/4/17	2/4/17	Caleb	1 hr
Update Portfolio	2/5/17	2/5/17	All	2 hrs

Designed First Protoype	2/5/17	2/5/17	All	1 hr
Progress Report Week 4	2/5/17	2/7/17	All	3 hrs
Lab 4	2/8/17	2/8/17	All	1 hr
EEPROM Data Analyzed	2/12/17	2/12/17	All	1 hr
Progress Report Week 5	2/12/17	2/14/17	All	3 hrs
Lab 5	2/15/17	2/15/17	All	1 hr
Progress Report Week 6	2/19/17	2/21/17	All	3 hrs
Update Portfolio	2/19/17	2/19/17	Matt	.5 hr
Draw Wings to be 3D Printed	2/19/17	2/19/17	Kenny	1 hr
Lab 6	2/22/17	2/22/17	All	1 hr
Draw Motor Supp. to be 3D Printed	2/22/17	2/23/17	Caleb	2 hrs
Draw Nose Piece to be 3D Printed	2/22/17	2/22/17	Matt	1 hr
Draw Servo Supp. to be 3D printed	2/22/17	2/22/17	Sam	1 hr
Lab 7	3/1/17	3/1/17	All	1 hr
Lab 7 Exec. Summary	3/3/17	3/7/17	All	3 hrs.
Lab 8 Report	3/8/17	3/9/17	All	4 hrs
Lab 8a	3/10/17	3/10/17	All	1 hr
Lab 8b	3/20/17	3/20/17	All	1 hr
Update Portfolio	3/20/17	3/20/17	Sam	.5 hr
PDR	3/21/17	3/26/17	All	6 hrs

Lab 8c	3/22/17	3/22/17	All	1 hr
Lab 9a	3/24/17	3/24/17	All	1 hr
Lab 9b	3/27/17	3/27/17	All	1 hr

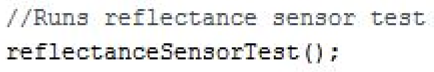
Arduino Code

Week 1



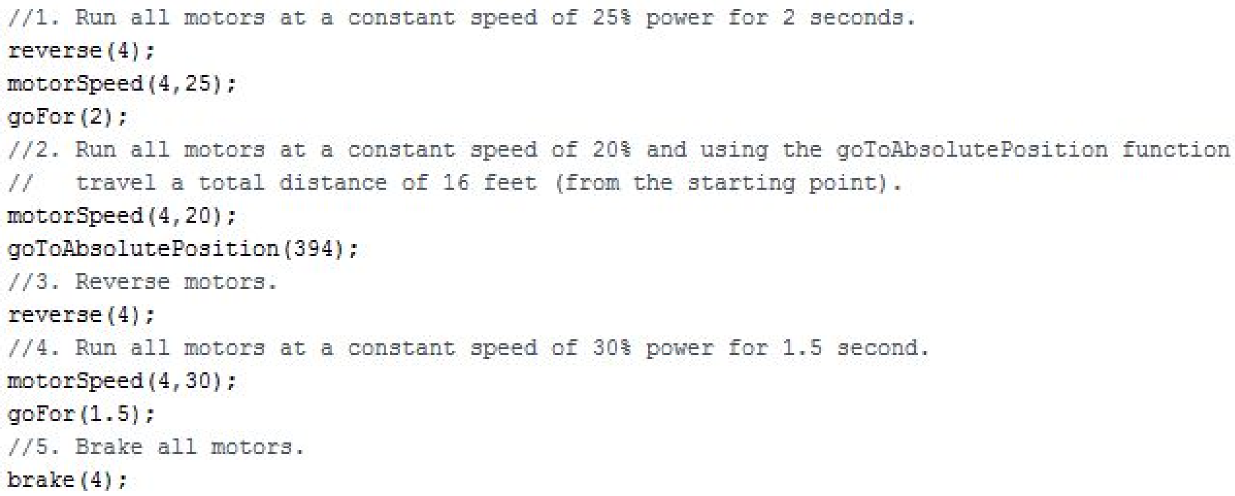
```
//1. Accelerate motor one from start to 15% power in 2.5 seconds.
celerate(1,0,15,2.5);
//2. Run motor one at a constant speed (15% power) for 1 second.
goFor(1);
//3. Brake motor one.
brake(1);
//4. Accelerate motor two from start to 27% power in 4 seconds.
celerate(2,0,17,4);
//5. Run motor two at a constant speed (27% power) for 2.7 seconds.
goFor(2.7);
//6. Decelerate motor two to 15% power in 1 second.
celerate(2,27,15,1);
//7. Brake motor two.
brake(2);
//8. Reverse the direction of only motor 2.
reverse(2);
//9. Accelerate all motors from start to 31% power in 2 seconds.
celerate(4,0,31,2);
//10. Run all motors at a constant speed of 35% power for 1 second.
motorSpeed(4,35);
goFor(1);
//11. Brake motor two but keep motor one running at a constant speed (35% power) for 3 seconds.
brake(2);
motorSpeed(1,35);
goFor(3);
//12. Brake all motors for 1 second.
brake(4);
goFor(1);
//13. Reverse the direction of motor one.
reverse(1);
//14. Accelerate motor one from start to 19% power over 2 seconds.
celerate(1,0,19,2);
//15. Run motor two at 35% power while simultaneously running motor one at 19% power for 2 seconds.
motorSpeed(2,35);
motorSpeed(1,19);
goFor(2);
//16. Run both motors at a constant speed (19% power) for 2 seconds.
motorSpeed(4,19);
goFor(2);
//17. Decelerate both motors to 0% power in 3 seconds.
celerate(4,19,0,3);
//Brake all motors.
brake(4);
```

Week 2



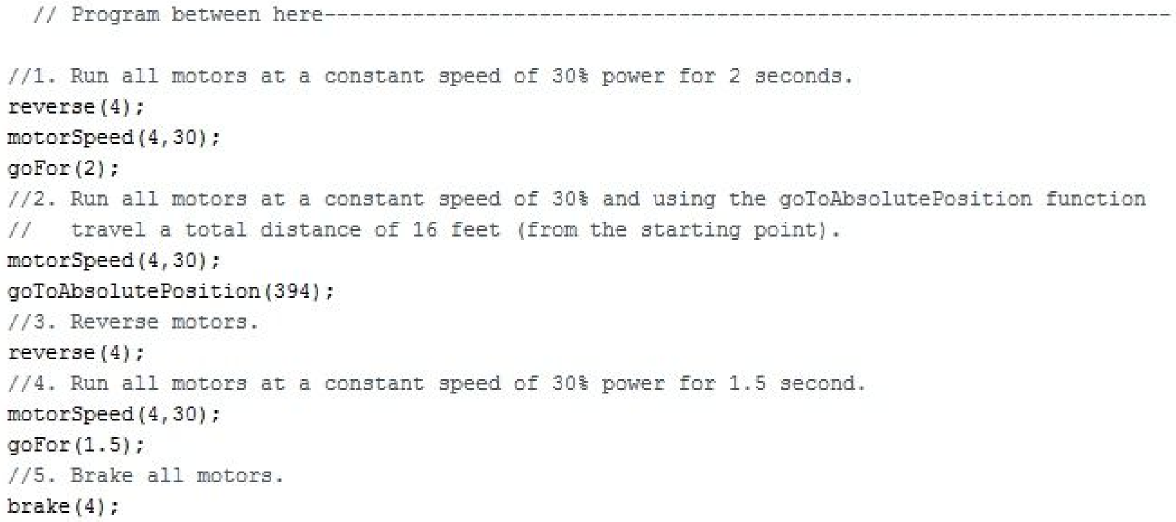
```
//Runs reflectance sensor test
reflectanceSensorTest();
```

Week 3 and Week 4



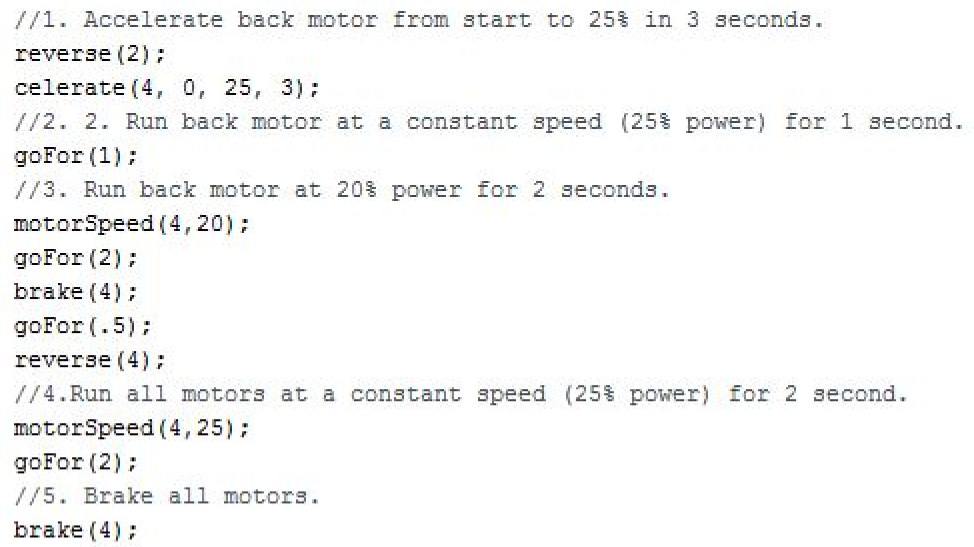
```
//1. Run all motors at a constant speed of 25% power for 2 seconds.
reverse(4);
motorSpeed(4,25);
goFor(2);
//2. Run all motors at a constant speed of 20% and using the goToAbsolutePosition function
//   travel a total distance of 16 feet (from the starting point).
motorSpeed(4,20);
goToAbsolutePosition(394);
//3. Reverse motors.
reverse(4);
//4. Run all motors at a constant speed of 30% power for 1.5 second.
motorSpeed(4,30);
goFor(1.5);
//5. Brake all motors.
brake(4);
```

Week 5



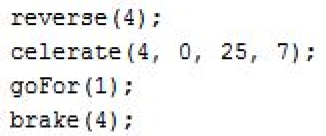
```
  // Program between here-----------------------------------------------------------------

//1. Run all motors at a constant speed of 30% power for 2 seconds.
reverse(4);
motorSpeed(4,30);
goFor(2);
//2. Run all motors at a constant speed of 30% and using the goToAbsolutePosition function
//   travel a total distance of 16 feet (from the starting point).
motorSpeed(4,30);
goToAbsolutePosition(394);
//3. Reverse motors.
reverse(4);
//4. Run all motors at a constant speed of 30% power for 1.5 second.
motorSpeed(4,30);
goFor(1.5);
//5. Brake all motors.
brake(4);
```

Week 6



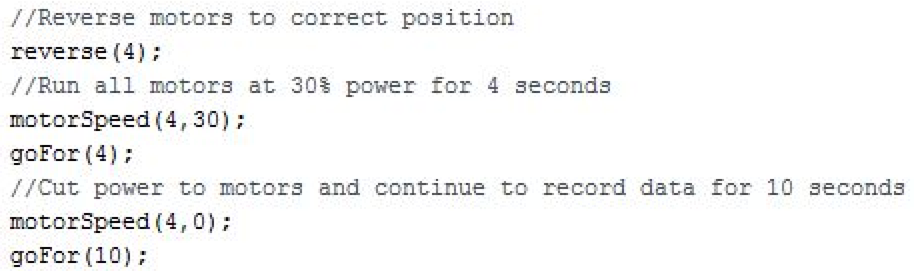
```
//1. Accelerate back motor from start to 25% in 3 seconds.
reverse(2);
celerate(4, 0, 25, 3);
//2. 2. Run back motor at a constant speed (25% power) for 1 second.
goFor(1);
//3. Run back motor at 20% power for 2 seconds.
motorSpeed(4,20);
goFor(2);
brake(4);
goFor(.5);
reverse(4);
//4.Run all motors at a constant speed (25% power) for 2 second.
motorSpeed(4,25);
goFor(2);
//5. Brake all motors.
brake(4);
```

Week 7



```
reverse(4);
celerate(4, 0, 25, 7);
goFor(1);
brake(4);
```

Week 8



```
//Reverse motors to correct position
reverse(4);
//Run all motors at 30% power for 4 seconds
motorSpeed(4,30);
goFor(4);
//Cut power to motors and continue to record data for 10 seconds
motorSpeed(4,0);
goFor(10);
```

Week 9

```
//Push Stop
// Start to gate
reverse(4);
rotateServo(0);
brake(4);
goFor(1);
```

```
motorSpeed(4, 40);
goFor(1);
celerate(4, 7.5, 28.5, 3);
goToAbsolutePosition(245);
brake(4);
reverse(4);
goToAbsolutePosition(415);
motorSpeed(4, 15);
goFor(1.2);
brake(4);

// Gate to R2D2
reverse(4);
goFor(11);
motorSpeed(4, 30);
goFor(1);
celerate(4, 7.5, 28.5, 3);
goToAbsolutePosition(724);
brake(4);
reverse(4);
goToAbsolutePosition(872);
motorSpeed(4, 15);
goFor(1.2);

//Backwards to gate
rotateServo(180);
reverse(4);
brake(4);
goFor(5);
motorSpeed(4, 45);
goFor(1.5);
celerate(4, 20, 38, 3);
goToAbsolutePosition(744);
brake(4);
reverse(4);
goToAbsolutePosition(574);
motorSpeed(4, 15);
goFor(1.2);
brake(4);

//Back to start
goFor(11);
reverse(4);
motorSpeed(4, 45);
goFor(1.5);
celerate(4, 20, 38, 3);
goToAbsolutePosition(415);
brake(4);
```

```
reverse(4);
goToAbsolutePosition(245);
motorSpeed(4, 15);
goFor(1.2);
brake(4);


//Pull Stop
// Start to gate
reverse(4);
rotateServo(0);
brake(4);
goFor(1);
motorSpeed(4, 40);
goFor(1);
celerate(4, 7.5, 28.5, 3);
goToAbsolutePosition(245);
brake(4);
rotateServo(180);
goToAbsolutePosition(415);
motorSpeed(4, 15);
goFor(1.2);
brake(4);

// Reaches gate
rotateServo(0);
goFor(11);
motorSpeed(4, 30);
goFor(1);
celerate(4, 7.5, 28.5, 3);
goToAbsolutePosition(724);
brake(4);
rotateServo(180);
goToAbsolutePosition(872);
motorSpeed(4, 15);
goFor(1.2);

//Backwards to gate
brake(4);
goFor(5);
motorSpeed(4, 45);
goFor(1.5);
celerate(4, 20, 38, 3);
goToAbsolutePosition(744);
brake(4);
rotateServo(0);
goToAbsolutePosition(574);
motorSpeed(4, 15);
```

```
goFor(1.2);
brake(4);

//Back to start
goFor(11);
motorSpeed(4, 45);
goFor(1.5);
celerate(4, 20, 38, 3);
goToAbsolutePosition(415);
brake(4);
rotateServo(0);
goToAbsolutePosition(245);
motorSpeed(4, 15);
goFor(1.2);
brake(4);
```

Week 10

```
//Start to gate
reverse(4);
rotateServo(0);
brake(4);
goFor(1);
motorSpeed(4, 34.6999987);
goFor(1);
motorSpeed(4, 23.15);
goFor(1.75);
goToAbsolutePosition(300);
brake(4);
rotateServo(180);
goToAbsolutePosition(415);
motorSpeed(4, 15);
goFor(1.2);
brake(4);

// Gate to R2D2
rotateServo(0);
goFor(11);
motorSpeed(4, 34);
goFor(1);
motorSpeed(4, 22.3);
goFor(1.85);
goToAbsolutePosition(770);
brake(4);
goFor(5);
rotateServo(180);
```

```
//Backwards to gate
brake(4);
goFor(2);
motorSpeed(4, 42);
goFor(1.5);
motorSpeed(4, 31);
goFor(2.5);
goToAbsolutePosition(680);
brake(4);
rotateServo(0);
goToAbsolutePosition(574);
motorSpeed(4, 15);
goFor(1.2);
brake(4);

//Back to start
goFor(6);
rotateServo(180);
brake(4);
goFor(5);
motorSpeed(4, 45);
goFor(1.5);
motorSpeed(4, 33);
goFor(3);
goToAbsolutePosition(250);
brake(4);
```

Final Code

```
//Start to gate
reverse(4);
rotateServo(0);
brake(4);
goFor(1);
motorSpeed(4, 35);
goFor(1);
motorSpeed(4, 23.15);
goFor(1.75);
goToAbsolutePosition(300);
brake(4);
rotateServo(180);
```

```
goToAbsolutePosition(415);
motorSpeed(4, 15);
goFor(1.2);
brake(4);
while (!isStopped()) {
  if (getVehiclePostion() > 476) {
    stopAEV();
    }
  }
if(getVehiclePostion() < 473) {
    reverse(4);
    motorSpeed(4, 28);
    goToAbsolutePosition(473);
    brake(4);
    reverse(4);
    stopAEV();
  }

// Gate to R2D2
rotateServo(0);
goFor(7);
motorSpeed(4, 34);
goFor(1);
motorSpeed(4, 23);
goFor(1.85);
goToAbsolutePosition(770);
brake(4);
goFor(10);
rotateServo(180);

//Backwards to gate
brake(4);
goFor(2);
motorSpeed(4, 42);
goFor(1.5);
motorSpeed(4, 30);
goFor(2.5);
goToAbsolutePosition(680);
brake(4);
rotateServo(0);
goToAbsolutePosition(590);
motorSpeed(4, 15);
goFor(1.2);
brake(4);
while (!isStopped()) {
  if (getVehiclePostion() < 521) {
    stopAEV();
    }
  }
if(getVehiclePostion() > 540) {
  reverse(4);
  motorSpeed(4, 32);
  goToAbsolutePosition(540);
  brake(4);
  reverse(4);
  stopAEV();
  }


//Back to start
goFor(5);
rotateServo(180);
brake(4);
goFor(3);
motorSpeed(4, 45);
goFor(1.5);
motorSpeed(4, 37);
goFor(3);
goToAbsolutePosition(250);
brake(4);
rotateServo(0);
goToAbsolutePosition(150);
motorSpeed(4, 27);
goFor(1.2);
brake(4);
while (!isStopped()) {
  if (getVehiclePostion() < 30) {
    stopAEV();
    }
  }
if(getVehiclePostion() > 50) {
  reverse(4);
  motorSpeed(4, 20);
  goToAbsolutePosition(50);
  brake(4);
  reverse(4);
  stopAEV();
  }
```

Methods

```
boolean isStopped() {
  boolean stopped = false;
  int pos1 = getVehiclePostion();
  goFor(.2);
  int pos2 = getVehiclePostion();
  if (pos2 - pos1 <= 0) {
      stopped = true;
      }
  return stopped;
  }

void stopAEV() {
    int mSpeed = 20;
    while (!isStopped()) {
      motorSpeed(4, mSpeed);
      goFor(.1);
      mSpeed += 5;
    }
    brake(4);
  }
```

10413

AEV Final Testing Scoresheet

Team/Team Name: P Instructor: Schlosser Class Time: 10413

This sheet must be filled out and signed by a member of the Instructional Staff by the end of Lab. The Instructor/TA must watch the AEV complete the operational objectives and will record the results below.

		Run 1			Run 2		
Procedure		Yes	No	PTS Earned	Yes	No	PTS Earned
Team shows proper testing procedure (up to 10 points)		✓		/10	✓		/10
AEV starts and travels to first gate		✓		/4	✓		/4
Gate Routine	Stops before gate	✓		/4	✓		/4
	Waits 7 seconds	✓		/4	✓		/4
	Travels through gate	✓		/4	✓		/4
AEV starts and travels to loading zone and waits for 5 seconds		✓		/4	✓		/4
AEV connects to cargo & travels to gate (crashes into cargo-deduct <= 2)		✓		/4	✓		/4
Gate Routine	Stops before gate		✓	/4		✓	/4
	Waits 7 seconds	✓		/4	✓		/4
	Travels through gate	✓		/4	✓		/4
AEV starts and travels to starting point		✓		/4	✓		/4
Total Points Earned				46/50	46		/50
Total Score = Total Pts Earned * Δt					Max Total Score		72.7

Track Layout: outside
(Inside or Outside)

Mass of AEV: .280
(in kilograms)

Total Energy: 198 / 178
(Joules)

Total Time Run1: 69
(seconds)

Total Time Run2: 63
(seconds)

Delta Time Run 1:

$$\Delta t1 = 1 + \frac{150 - \text{total time}}{150}$$

= ________

Delta Time Run 2:

$$\Delta t2 = 1 + \frac{150 - \text{total time}}{150}$$

= ________

Energy/Mass: 636
(Joules per kilogram)

Your final score will be based on the **Energy/Mass ratio** (how efficient is the team's AEV) and the **Total Score** (time and distance requirements).

Instructor / TA Signature: ________ Date: 4/12/17