Advanced Energy Vehicle Design Project: CRITICAL DESIGN REVIEW

Submitted to:
Inst. Dr. Busick
GTA Chris Chang

Created by:
Blake Harriman, Computer Science & Engineering
Kyle Kottyan, Computer Science & Engineering
Kyle Pellikan, Computer Science & Engineering
Joe Sudar, Chemical Engineering

Fundamentals of Engineering 1182
The Ohio State University
Columbus, OH
13 April 2017

Executive Summary

The purpose of the Advanced Energy Vehicle (AEV) Design Project was for the team to create the most energy and time-efficient, blade-propelled vehicle possible while also fulfilling the project's overall requirements. The requirements for the design project, as outlined in the mission concept review (MCR), were to successfully navigate the rail system installed on the ceiling of the lab rooms. Additionally, the vehicle was to start at one end of the track, travel to a gate equipped with a motion sensor, travel to the end, and safely retrieve the cargo there. A successful return trip was then needed to fully complete the mission.

As previously described, the two quantities of concern were time and power consumption, which were both measured using the MATLAB technical programming language. More specifically, the team employed a design tool in MATLAB (created by university faculty) that allowed it to download and plot the data collected by the AEV during each test run. Aside from the software, the physical hardware, which included the Arduino microcontroller and reflectance sensors, collected the data that was to be imported into MATLAB for analysis. Through the useful combination of the hardware and software, the team could successfully measure the time and power consumption of various design and coding implementations.

For most the project's duration, the team was tasked with the gradual development of the AEV's software and physical design. The time spent during these phases of the project enabled the team to become familiar with the hardware and Arduino software. Through this familiarization, team members could more easily work through the remainder of the assignment, which was more focused on the refinement of the AEV's final design.

Performance Test 01 involved the testing of two different variations of the team's final design. The alterations in the design were made to determine which variation was more energy efficient. To complete this performance test, the team altered the orientation of the trapezoidal wings where the motors and propellers were mounted, either being positioned up at a 45-degree angle (Up-Wing) or down at a 45-degree angle (Down-Wing). To observe the different energy consumption, the team ran both designs on the same coding scenario. The conclusion of this performance test was that the Up-Wing configuration was slightly more energy efficient than its counterpart, using about 14 less watts. As a result, the team chose to maintain the Up-Wing design for the remainder of the project. Determining this aspect of the vehicle led into the second performance test.

Much like the design-efficiency analysis in Performance Test 01, Performance Test 02 was similar, but through the testing of two different coding scenarios that the team created. The two main coding structures were a coasting scenario and a hybrid that utilized both self-correction and coasting techniques. To gather data for this test, the team created a coding scenario that solely used coasting and one that was the hybrid previously described. The conclusion of this performance test demonstrated that although coasting was overall more energy efficient, the hybrid coding scenario was more precise in its movement along the track. This only sacrificed a small amount of energy, which was a trade off the team was willing to take.

The final performance test was concerned with refining the near-finished design. The duration of the testing was to be conducting test runs on the rails in preparation of the final examinations. Gathering data for this performance test consisted of multiple trials. Each of which varied slightly from its predecessor as the team developed a coding scenario that was more energy efficient and that fulfilled project requirements.

Table of Contents

Introduction

The goal for the Advanced Energy Vehicle was to complete the MCR by traveling past a gate, retrieve precious cargo, and return to its original starting point, passing through the gate again. The team was to design an AEV that would save the most energy, as well as complete the mission's objective in under two and a half minutes. Over the course of a semester, the team tested their AEV to improve its performance as well as discover other ways to save more energy or time. This lab report was created to analyze the performance of the AEV during its final test run, as well as what improvements could be made if the team had more time to test.

Experimental Methodology

In the creation of an Advanced Energy Vehicle (AEV), physical and code design were key factors. The first step involved brainstorming a physical design for the AEV from a given set of parts including, and Arduino board, two motors, propellers, different bases, wings and rail-mounts, and a battery mount. Using these parts, each team member developed their own design, resulting in 4 designs. The team proceeded to score each design using concept screening and scoring matrices to eliminate ideas and settle on one final design to improve and finalize. The concept screening and scoring matrices took important qualities within an AEV such as weight, balance, durability, center of gravity, and required maintenance. Based on the results of the scoring matrices the team determined that the third design scored the best and therefore would be improved upon. The final AEV design used a medium 15x5 cm rectangular base, the L-shaped rail-mount, and trapezoidal wings mounted to the base with the 45-degree brackets in the upward position (Figure 11). The motors were mounted on the topside of the wings with the 3-inch propellers attached to topside of the wings (Figure 7). The propellers were attached in such a way that they can run forward and reverse. The rail-mount was mounted on the back of the base with the Arduino board mounted on the front-right of the base. The off-center position of the Arduino was to correct the leftward shift in weight caused by the position of the wheels on the rail. The battery was mounted on the underside of the base directly beneath the rail-mount, causing the center of gravity to rest under the strongest connection point on the base greatly improving overall balance. The magnetic connection was created through connecting multiple straight brackets and a 90-degree bracket together. This entire apparatus was then positioned on the front and center of the base and then zip tied in place, ensuring a perfectly centered connection site. The reflectance sensors were then attached to the side of the rail-mount with countersunk screws and zip ties (Figure 6). The front wheel of the AEV was marked with an alternating pattern of reflective and non-reflective surfaces to interact with the reflectance sensors. Each time a reflective portion changed to a non-reflective portion it was recorded by the sensors as a mark. The Arduino board acted as a communicator between all the electrical components, telling the motors when to run, which direction to turn, and how long, and using the information collected in the reflectance sensors to dictate how far the vehicle has travelled in relation to a specific point (Figure 5). With these parts working together in the completed AEV, the vehicle was prepared to carry out a specific task.

The AEV was tested in many different trials using multiple code scenarios. Each trial was carried out to determine what changes needed to be made within each version of code, eventually leading to a finalized code which would consistently carry out a specific task. The specific task was presented to the team as the Mission Concept Review (MCR). The MCR tasked each team with creating an AEV and code that successfully travelled from a starting point to an activation sensor. This sensor required the AEV to idle for 7 seconds to activate and open a gate. Once open, the AEV was to proceed through the gate and retrieve a cargo waiting on the opposite side of the track. Once connected to the cargo via magnet, the AEV was to idle for 5 seconds before traveling back to the gate. Again, the AEV was required to idle for 7 seconds, activate the sensor, and open the gate. Once open, the AEV was to travel through the gate and return to the starting point with the cargo. This entire scenario was to be completed under 2 minutes and 30 seconds, and with minimal power usage. Based on the AEV's ability to complete this task the code was altered and then tested until performing the task consistently.

The completion of the MCR was not the only data used in optimizing the code. During each trial, the Arduino board recorded data based on the energy consumption over distance and time. Using MATLAB, this data was extracted and synthesized to generate graphs which displayed energy efficiency trends. To optimize the energy efficiency within each run of the AEV, this data was analyzed, and the code was altered to decrease the total energy consumption. Through the optimization of energy usage and the consistency in completion of the MCR, the most energy and time efficient code implementation was created, resulting in successful final testing of the AEV.

Results

Design Analysis and Evaluation

The group's design choices throughout this project depended upon a conclusive analysis of energy consumption and balance as the designs traveled along the track during test runs. The team first got to a general, tentative, final design through analysis and observations of various ideas during labs 4 and 5. During labs 4 and 5 the team came up with individual ideas for the AEV design and then used a concept scoring and screening matrix to determine which design was the best. As seen in Table 1 below, design C scored the highest which is what the team decided to go with for the general and tentative final design.

Table 1: Scoring matrix for all considered design concepts

		Design A - Kyle K.		Design B - Joe S.		Design C - Team	
Success Criteria	**Weight (%)**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**
Balance	5	3	0.15	1	0.05	3	0.15
Minimal Blockage	15	3	0.45	3	0.45	4	0.60
Center of Gravity	10	2	0.20	2	0.20	3	0.30

Location							
Maintenance	25	4	1.0	2	0.50	4	1.0
Durability	15	2	0.30	1	0.15	4	0.60
Cost	20	3	0.60	3	0.60	3	0.60
Environment	10	3	0.30	3	0.30	3	0.30
Total Score		3		2.25		3.55	
Continue		No		No		Refine	

A main requirement stated in the mission concept review was that the AEV was to be as energy efficient as the team could make it. A significant contributor to the overall power usage of the AEV was the type and configuration of the propellers that would move the vehicle through its entire mission.

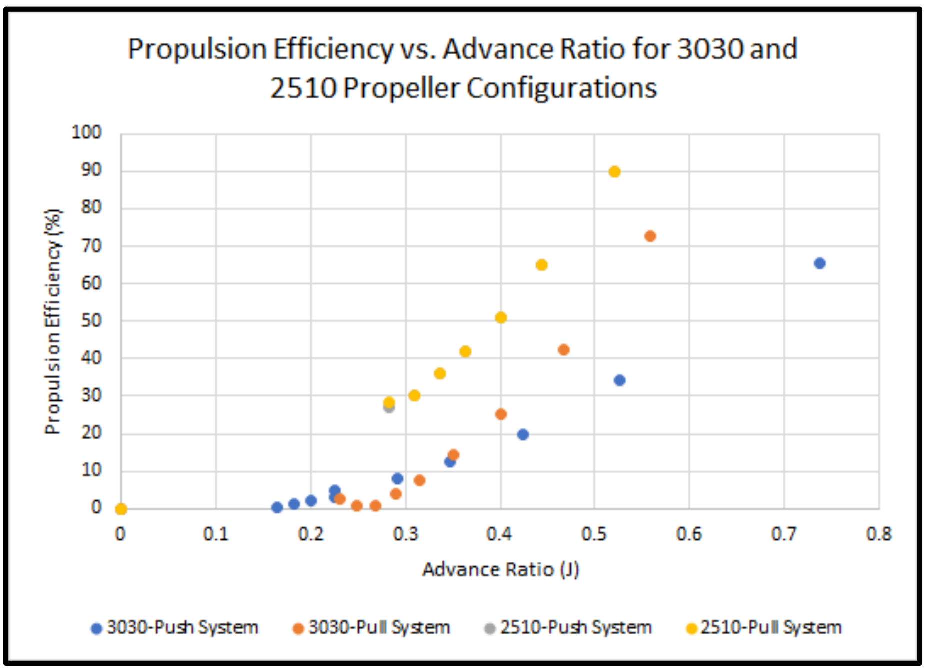
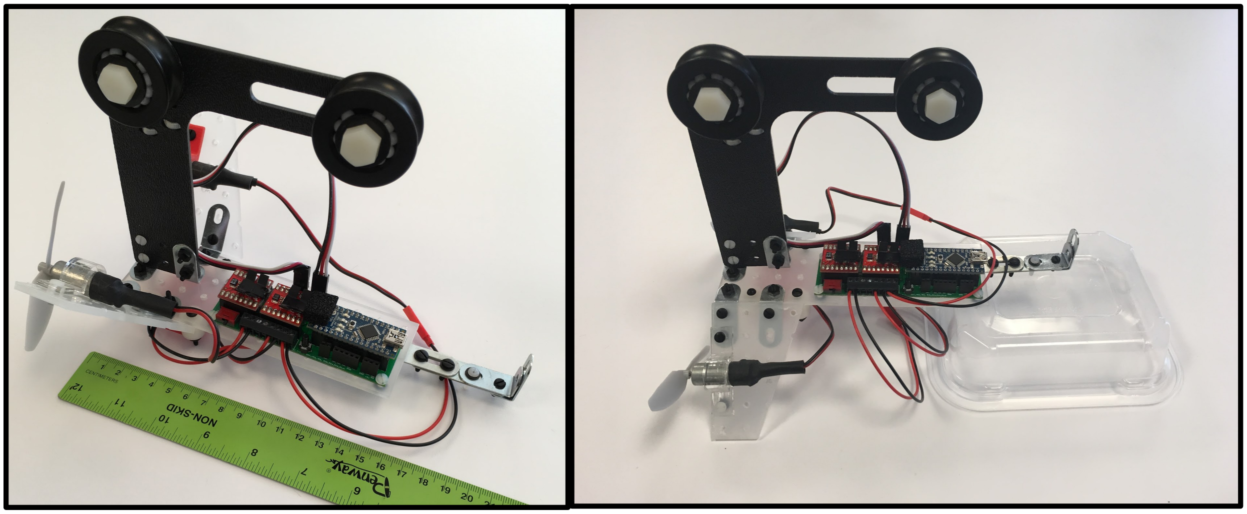


Figure 1: Propulsion efficiency vs. advance ratio for each propeller configuration

Figure 1 above is a graph that displays the propulsion efficiency of each blade and system configuration versus the measured advance ratio (wind speed to propeller speed) at each efficiency point. This figure aided the team in deciding the blade type and configuration that would be the best-performing and most-efficient. From the figure, the 3-inch (7.62 cm.) configurations (3030-Push and Pull systems) were the best performing systems analyzed by the group. The group employed both the push and pull systems using these 3-inch blades to minimize the power that would be used.

After figuring out the tentative final design, the team decided to go with two possible final designs during Performance Test 01. The two designs (seen in Figures 2 and 3 below) the team went with for testing included one with the wings at an upwards angle of 45 degrees and one with the wings at a downwards angle of 45 degrees.



Figures 2 and 3: Up-Wing (left) and Down-Wing (right) designs

Of the two tested designs shown in the figures, the Up-Wing AEV was observed to be slightly more energy efficient than the Down-Wing AEV design. Due to these results, the team decided the Up-Wing design would be the final AEV design to be used for the remainder of the project. The results from this test is shown in Table 2 below.

Table 2: Power consumption of Up-Wing and Down-Wing designs

Up-Wing Design Power Consumption (watts)	Down-Wing Design Power Consumption (watts)
538.65	550.84

The next performance tests involved measuring the power consumption of two different coding implementations. Two strategies that the team tested was a coding implementation solely focused on coasting to the various checkpoints on the track and the other being a hybrid of self-correction and coasting. The use of loops in the code enabled the AEV to self-adjust based on its location.

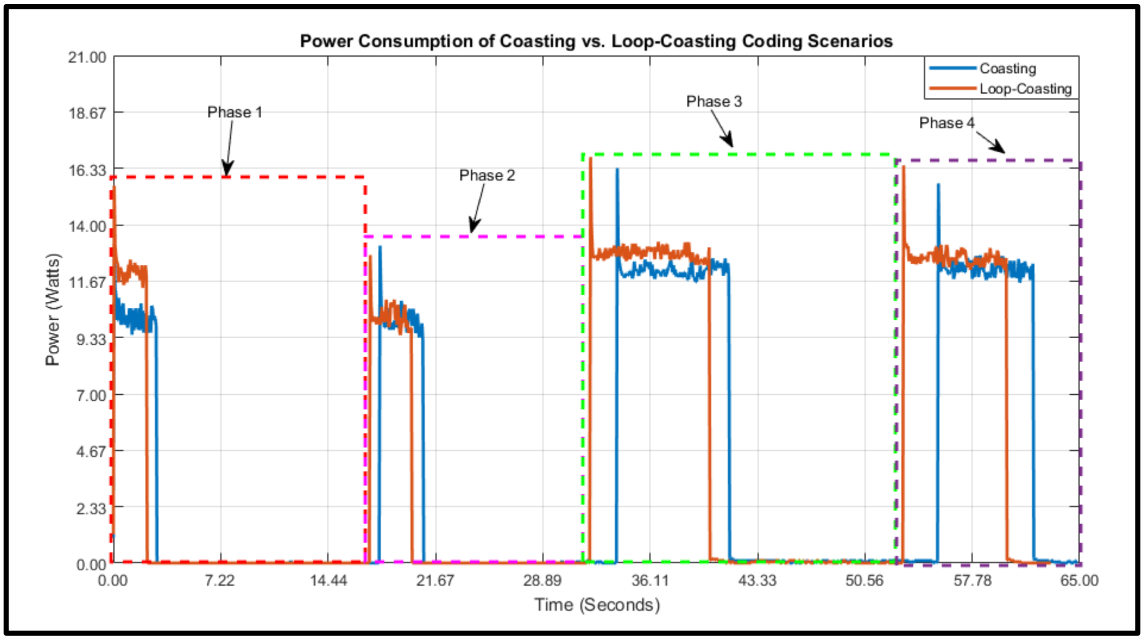


Figure 4: Supplied power vs. time for coasting and loop-coasting coding scenarios

Figure 4 above is a plot of the supplied power versus time of the two coding implementations tested. The corresponding measurements of each phase indicated in the figure is shown in Table 3 below.

Table 3: Power consumption of coasting and loop-coast coding scenarios by phase

Phase:	Coasting Scenario Power Usage (watts)	Loop-Coasting Scenario Power Usage (watts)
1	489.65	450.31
2	494.97	479.54
3	1547.50	1729.20
4	1309.30	1469.40
Total:	**3841.42**	**4128.45**

The team concluded from the collected data that the coasting scenario was slightly more efficient than the hybrid of coasting and loop statements. Although this was the case, the team decided to choose the loop-coasting scenario for its running consistency while testing the AEV on the rails.

Discussion

During Performance Test 1, the team's goal was to test two different variations of the tentative, final design of the AEV to improve the consistency and energy efficiency of the AEV. The two variations focused on the wing orientations of the AEV. The two designs included one with an upward wing orientation (Up-Wing) and the other design had a downward wing orientation (Down-Wing). The team focused on the behaviors of these two designs with respect to both balance and energy efficiency when deciding which design to ultimately go with.

The two AEV designs that were tested during Performance Test 1 were strongly based on the results of the Concept Scoring and Screening Matrices. The Up-Wing design was based primarily on design C with the idea that the upward oriented wings would provide more balance to the AEV. The Down-Wing design was made with the idea in mind that the downwards orientations would lower the center of mass and improve stability on the track especially during the turns of the track. The data showed that the Up-Wing design was more energy efficient than the Down-Wing design by a small margin. The team observed that both designs were relatively the same with respect to balance while traveling along the track with the Up-Wing design being slightly more stable so the team decided to go with the Up-Wing design as the final design choice for the AEV.

After making the decision to go with the Up-Wing design, the team focused on getting the AEV to do the run as quickly as possible while still being the most energy efficient. During the first test run, the AEV was not able to properly complete the task due to a few flaws in the code and used an abundance of energy to go along with this error. The team also found that the AEV took a much longer time than expected during the run which then became an area of concern as the team prepared for the second final test run.

The team decided that the best way to go about fixing these errors and inefficiencies was to adjust the code until the AEV was consistently completing the task while taking as little time as possible and using as little energy as possible. At this point, changing the final design would be too time consuming so adjusting the code was the best route to take. After adjusting the initial coasting distances as well as making the loops used to check the position of the AEV more accurate, the team was satisfied with the final implementation of code. During the second final test run, the AEV was not able to get to the end of the track due to a miscalculation in the number of marks the AEV needed to travel. After being pushed along by one of the team members to get to the end, the AEV could successfully complete the rest of the task. The team found that with the new and improved code the AEV used much less energy while being significantly more time efficient. The mass to energy ratio for this second run also turned out to be a large improvement from the first run. Although the data the team recorded from this run indicated that the AEV was energy and time efficient, there were still potential errors that prevented the AEV from performing the most efficiently.

Potential errors that occurred during the design and testing process included uncontrollable variables as well as human errors in the code the group implemented. The uncontrollable errors included: having to test on different tracks, air flow due to other AEVs running at the same time, and inconsistent batteries. The team observed that these factors forced the team to have to constantly change numbers around for how far the AEV needed to travel before the motors stopped running to allow it to coast. This was problematic because the team had to waste time correcting this error to get accurate test results.

On the other hand, the team did not take into consideration potential problems the written code would cause. For example, the AEV did not consistently stop at the same place at the gate for every test run due to the uncontrollable variables discussed previously. Because the team used the command goToRelativePosition() that meant that the AEV would travel to a different part of the track during the 2nd and 4th phases of the run which caused the runs to be inconsistent. Looking back, if the team had decided to use goToAbsolutePosition(), then the 2nd and 4th phases of the runs would have been more consistent than what the team ended up having for the final test runs.

Conclusion

The overall objective of the AEV was to complete the MCR, getting the cargo from the end of the track to the beginning within a short time frame. The group decided to use the Up-wing design for further testing due to its slightly lower power output compared to the Down-wing design. The team used a code implementation that utilized coasting to minimize energy usage. With correctional loops added to the code, the AEV would perform more consistently and save time at the cost of slightly more energy. With a final design for both the AEV and the code, the team could test on the track two final times before the end of experimentation.

The AEV could complete the MCR within the allotted time, and with minimal error. The AEV's first final test run didn't go as planned, as it overshot the gate on the return trip and caused the gate to cancel its opening sequence. Thus, the AEV's first test was completely inefficient, wasting time and energy. The second test run proved to be more successful, leading to a strong energy/mass ratio as well as shaving off a few seconds on the overall run. Though the AEV successfully completed the mission, there were still some errors that prevented the AEV from performing to its full capabilities.

Errors naturally occur during the design and test process which is always an issue when making decisions that affect the energy and time efficiency of the AEV. Human error is always an issue, but there are additional variables that are out of the team's control. These variables include things such as: having to test in different settings, additional air flow caused by multiple AEVs running at the same time, and inconsistent batteries. There is no way to get completely rid of these errors so the team had to adjust the code accordingly for each lab and each time an additional design decision was made.

At the same time, human error is something that falls on the team to fix and is important to take into consideration whenever a decision is being made that could affect the AEV's time and energy efficiency. Examples of human error include: putting on screws too tight or too loose, inconsistent starting positions, and mistakes in the code itself. The team can solve these issues by being prepared for common mistakes like this and learning how to adjust to these common errors that most teams will inevitably run into at one point or another.

Appendix

Arduino Coding Implementations

```
// Final Test Code with Coasting and Conditional Statements
// TRIP TO END
//-------------------------------------------

reverse(4);

// Start to gate
motorSpeed(4,34);
goToRelativePosition(151);

motorSpeed(4,0);
goFor(8);

// Correct if short at first gate
while (getVehiclePostion() < 465) {
motorSpeed(4, 20);
goFor(0.5);
}

// Wait for gate to open
motorSpeed(4,0);
goFor(8);

// Gate glide to end
motorSpeed(4,35);
goToRelativePosition(157);

motorSpeed(4,0);
goFor(8);

// If short, correct to end
while (getVehiclePostion() < 950) {
  motorSpeed(4, 20);
  goFor(0.5);
}

motorSpeed(4,0);
goFor(5);

// TRIP BACK TO START
//-------------------------------------------
// Prepare for trip back by reversing motors
```

```
reverse(4);

// Glide back from end to gate
motorSpeed(4,45);
goToRelativePosition(-274);

motorSpeed(4,0);
goFor(6);

// If short, correct to gate on the way back
while (getVehiclePostion() > 536) {
  motorSpeed(4, 41);
  goFor(0.5);
}

motorSpeed(4,0);
goFor(9);

// After gate opening, glide to end
motorSpeed(4,45);
goToRelativePosition(-295);

motorSpeed(4,0);
goFor(8);

// If short, correct to end.
while (getVehiclePostion() > 10) {
  motorSpeed(4, 35);
  goFor(0.5);
}

brake(4);
```

```
// Coasting Scenario (Start to Cargo Retrieval)
// Used to test for implementation efficiency
// Orient direction correctly
reverse(4);

// Start to shutdown point for gliding
motorSpeed(4,35);
goToRelativePosition(70);

// Glide from shutdown position to gate sensor and wait for 7 seconds
// start to gate takes about 8 seconds, 7 additional for gate opening
motorSpeed(4,0);
goFor(15);
```

```
// Run motors at 35% power until relativePosition of 70 marks
motorSpeed(4,35);
goToRelativePosition(70);

// Glide to cargo and wait 5 seconds for secure connection
motorSpeed(4,0);
goFor(13);

// Reverse motors for trip back
reverse(4);

// Cargo retrieval area to gate sensor
motorSpeed(4,45);
goToRelativePosition(132);

// Glide to gate and wait 7 seconds
motorSpeed(4,0);
goFor(14);

// 2nd gate glide to end
motorSpeed(4,45);
goToRelativePosition(140);
```

Equipment Figures

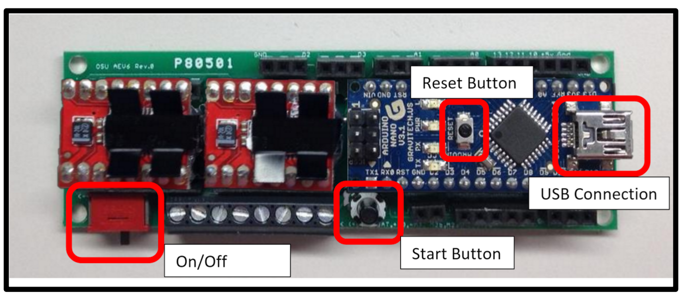


Figure 5: Arduino board

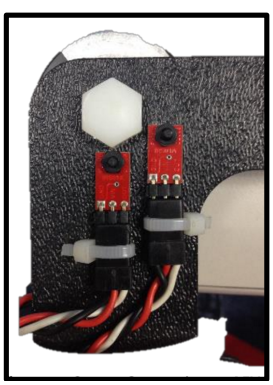


Figure 6: Reflectance sensors (left) and Figure 7: Motors with 7.62 centimeter (3-inch) propellers (right)

Vehicle Designs & Constructions

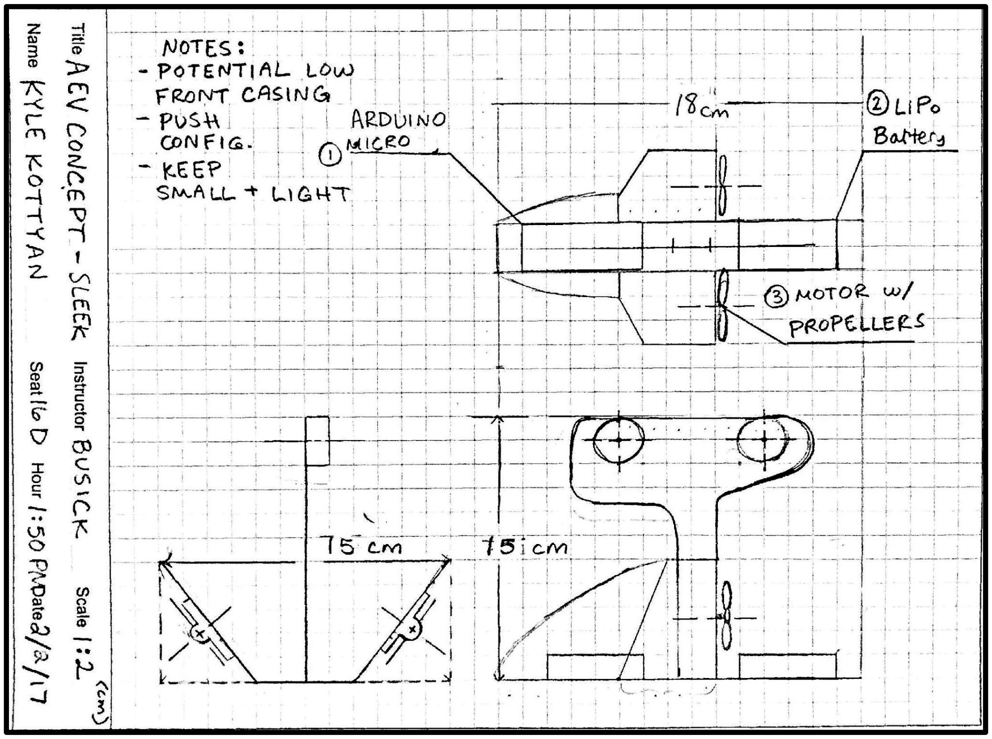


Figure 8: Design A - Kyle K.

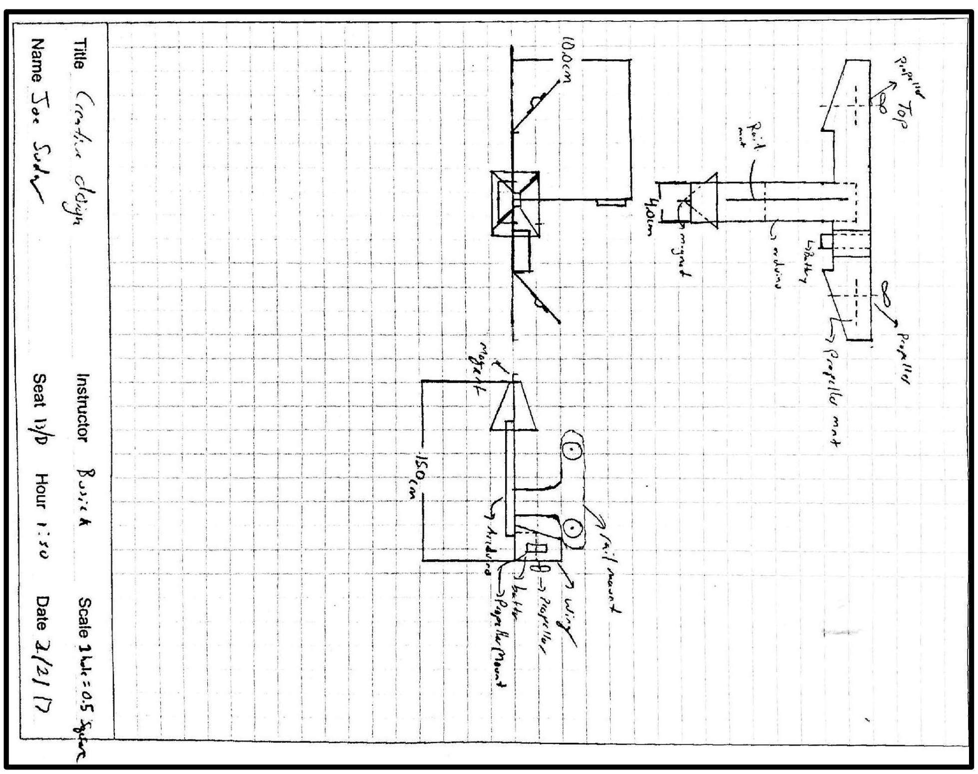


Figure 9: Design B - Joe S

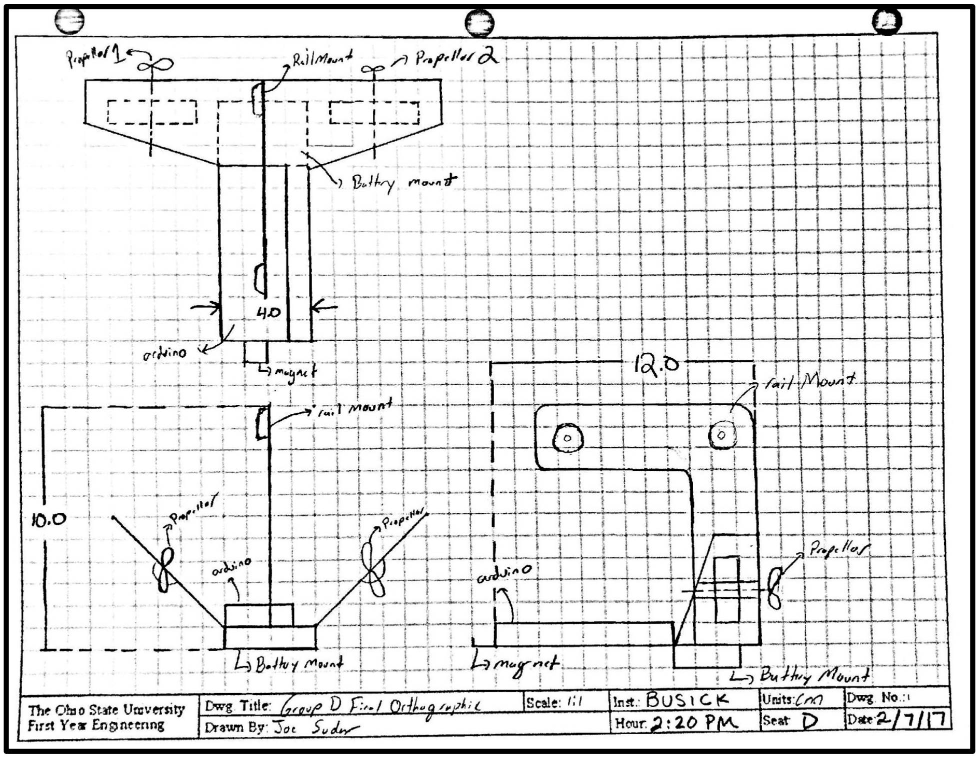


Figure 10: Design C - Team

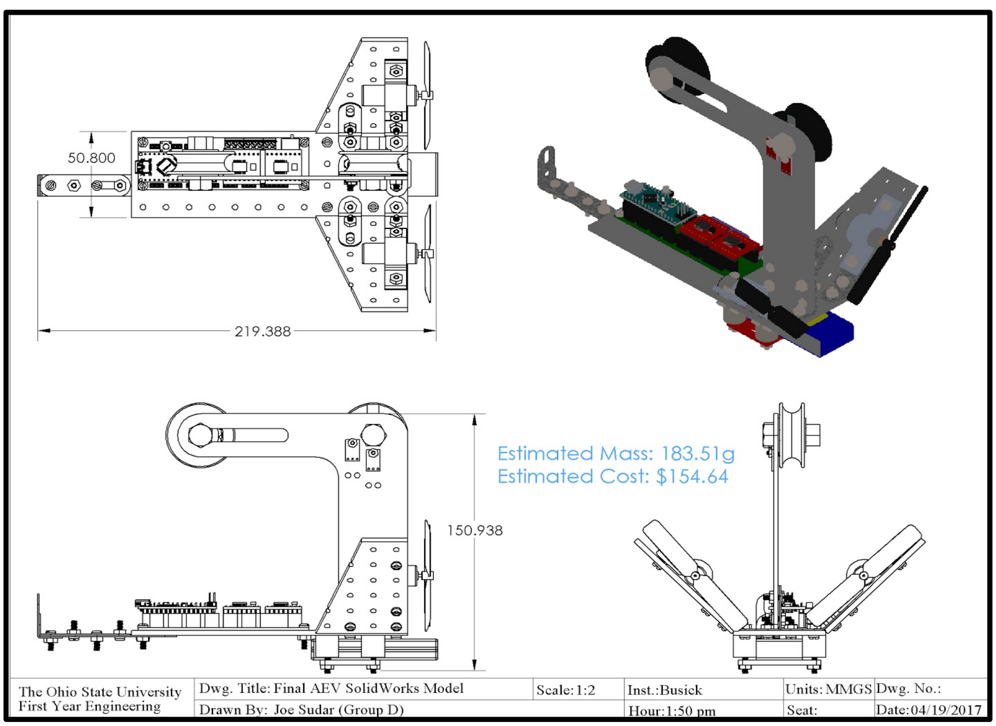


Figure 11: Final AEV SolidWorks Model

Production Schedule

Table 4: Group D Production Schedule

Tasks:	Start:	Finish:	Due Date:	Kyle Kottyan	Joe Sudar	Kyle Pellikan	Blake Harriman	Time:	Percentage Complete
Lab 2 Progress Report	01/19	01/25	01/26	X Forward Looking	X Goal Creation	X Schedule Creation	X Backward Looking	4 Hours	100%
Lab 3 Progress Report	01/26	02/08	02/09	X Report Reviser	X Schedule Creator	X Forward Looking	X Backward Looking	4 Hours	100%
Lab 4 Progress Report	02/04	02/08	02/09	X Report Reviser	X Schedule Creator	X Forward Looking	X Backward Looking	3 Hours	100%
Lab 5 Progress Report	02/09	02/15	02/16	X Report Reviser	X Schedule Creator	X Forward Looking	X Backward Looking	3 Hours	100%
Lab 6 Progress Report	02/16	02/22	02/23	X Writer/ Editor	X Forward Looking	X Schedule Creator	X Backward Looking	3 Hours	100%
Oral Presentation - PDR	02/23	03/01	03/02	X Writer/ Presenter	X Writer/ Presenter	X Writer/ Presenter	X Writer/ Presenter	4 Hours	100%
Lab 8 Progress Report	03/02	03/08	03/09	X Writer/ Editor	X Forward Looking	X Schedule Creator	X Backward Looking	2 Hours	100%
Lab 9 Progress Report	03/19	03/22	03/23	X Writer/ Editor	X Forward Looking	X Schedule Creator	X Backward Looking	2 Hours	100%
Preliminary Design Report	03/20	03/26	03/27	X Executive Summary/ Reviser	X Results/ Conclusion	X Table of Contents/ Appendix	X Introduction /Discussion	4 Hours	100%
SolidWorks Model	03/09	03/25	04/20		X Creator			5.5 Hours	100%
Lab 10	03/30	04/02	04/03	X	X	X	X	2	100%

Progress Report				Writer/ Editor	Forward Looking	Schedule Creator	Backward Looking	Hours	
Oral Presentation Draft - CDR	04/02	04/05	04/06	X Writer			X Editor	1 Hour	100%
Lab 11 Progress Report	04/06	04/09	04/10	X Writer/ Editor	X Forward Looking	X Schedule Creator	X Backward Looking	2 Hours	100%
Critical Design Review	04/15	04/19	04/20	X Executive Summary/ Reviser	X Results/ Conclusion	X Table of Contents/ Appendix	X Introduction /Discussion	4 Hours	100%
Oral Presentation	04/02	04/19	04/20	X Presenter	X Presenter	X Presenter	X Presenter	2 Hours	100%
Project Portfolio	03/09	04/20	04/20			X Creator		5 Hours	95%