Group I – Carlos Perez-Oviedo, James Pfeifer, Albert Hsu, Zhanhao Wang Progress Report Week 2

Instructor – Professor Shrock, GTA - Sheena Marston 2/8/17

# Week 2

## Situation

In Lab 2, the group conducted two different tests in preparation for creating our AEV. For part A, three members of the group tested the reflectance sensors while the fourth member conducted part B, thrust tests for each propeller configuration. In part A, the AEV was fully assembled and all components attached. The group then ran a test program on the Arduino and determined that the reflectance sensors were functional as the serial monitor in the IDE registered marks when the reflecting wheel was spun. For part B, the two different diameter propellers were tested in a push and pull configuration at different power levels. The amount of thrust for each power level and configuration was recorded to identify the most efficient propeller setup. These tests will help in designing and constructing the AEV by ensuring all components are functional and allowing the use of the most efficient propeller configuration.

The group was mired by a persistent technical problem while conducting the reflectance sensor test. The serial monitor would not read data from the sensors. Nearly every component, including the reflectance sensors and reflecting wheel, was replaced. Ultimately, we determined that the start button was never pressed, which is why the marks were not recorded. Because of the great difficulty, the group learned to thoroughly troubleshoot the AEV, which will ensure an easier time in future labs.

## Results and Analysis

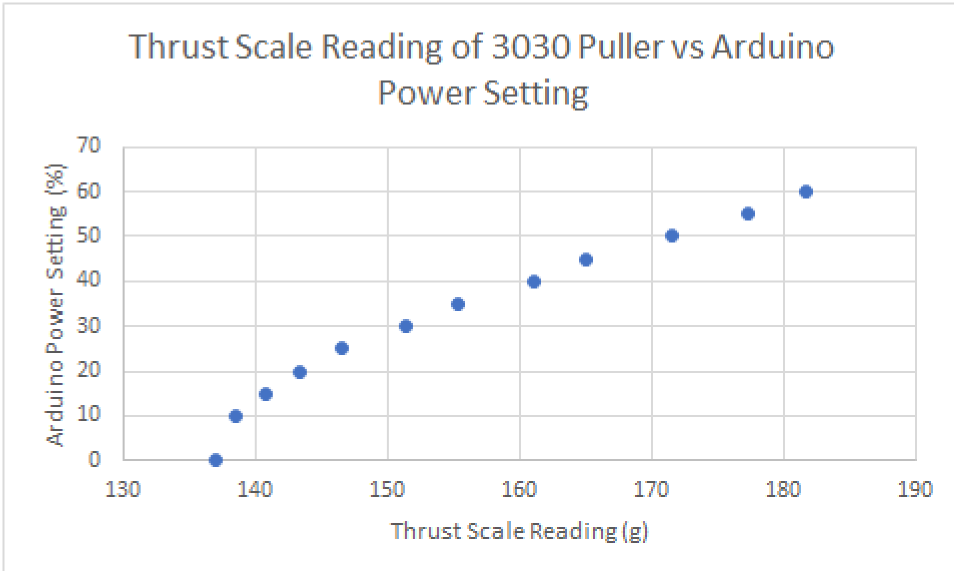
Due to the technical problem the group encountered, the AEV was never tested on the track. While we determined the reflectance sensors functioned in a stationary test environment, the group is still unsure about the behavior and performance of the AEV on the track. This prevents the team from optimizing the AEV design or code to ensure it reaches the gate and stops.

*Table 1: 3030 Propeller in Puller Configuration*

| Current (amps) | Thrust Scale Reading (grams) | RPM (RPM) | Arduino Power Setting (%) |
|---|---|---|---|
| **0.16** | 137 | 0 | 0 |
| **0.27** | 138.4 | 2155 | 10 |
| **0.35** | 140.7 | 3080 | 15 |
| **0.45** | 143.3 | 4000 | 20 |
| **0.54** | 146.5 | 4700 | 25 |
| **0.64** | 151.3 | 5500 | 30 |
| **0.73** | 155.3 | 6200 | 35 |
| **0.82** | 161 | 6886 | 40 |
| **0.9** | 165 | 7400 | 45 |
| **0.99** | 171.5 | 8000 | 50 |
| **1.07** | 177.2 | 8622 | 55 |
| **1.17** | 181.6 | 9221 | 60 |

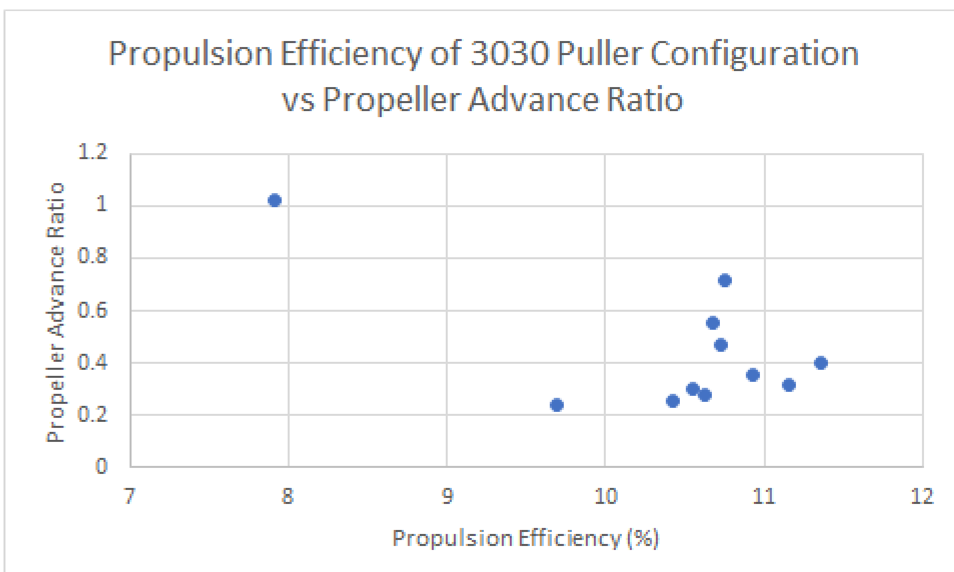
The collected data for the 3030 propeller in the puller configuration indicated the highest average thrust scale reading, making it the most powerful of the four configurations.

*Figure 1: Thrust Scale Reading of 3030 Puller vs Arduino Power Setting*




The figure shows how for the 3030 puller configuration has the thrust scale reading increase as the Arduino power setting increases. It initially increases with a logarithmic pattern before transitioning to a more linear pattern.

*Figure 2: Propulsion Efficiency of a 3030 Propeller Puller Configuration vs Propeller Advance Ratio*




There isn't an obvious pattern to the data and it seems to be random, not providing a definitive pattern.

This data helped the team determine that the optimal propeller configuration to use would be the 3030 puller because it offers the most thrust in the low to mid power range, requiring less run time to achieve the same distance due to greater acceleration and using less power. This will help the team create the code for the MCR by giving a constraint to stay in lower power settings as relatively large amounts of thrust can still be generated.

Takeaways

A significant takeaway from this lab was to ensure that all parts of the procedure to run the Arduino is checked to ensure that it runs properly. This would reduce delays due to human error. As there was no on-track testing done, the group cannot offer any insight into changes that should be made to the AEV design or code.

The group was also introduced to new program functions, such as goToRelativePosition and goToAbsolutePosition. Important conversion rates are also given, like .4875 inches/mark. Several formulas to calculate things such as the power input/output and propeller efficiency were also introduced.

## Week 3

### Situation

For week 3, the team will study how to design AEV Sample Vehicle. The main purpose of this lab is teach and guide the team to be creative, to bring something new and produce through imaginative skill. There are several ways for team how to become creative including read more, ask more questions and exchange ideas. Two tasks will be finished in lab 3. First, teammates have to complete MCR with their own design ideas. The ideas which come up by teammates should not be the same. In addition, the teammates have to consider how to deal with materials. The second task is to review the Mission Concept Review. The team also have to think if the AEV design fit the criteria and mission objectives.

### Weekly Goals

1. Brainstorm and make three designs for AEV Sample Vehicle
2. Learn several methods to improve creativity, get familiar with difficulty to creativity and the AEV kit
3. Study some basic skills for drawing.
4. Test the AEV om the track

### Weekly Schedule

*Table 1: Week 3 Schedule*

| Task | Teammate(s) | Start Date | Due Date | Time Needed |
|---|---|---|---|---|
| Make Design | Carlos<br>James Tyler | 2/1/17 | 2/8/17 | 1 hr |
| Assemble AEV | Albert | 2/1/17 | 2/2/17 | 1 hr |
| Progress Report Summary | Carlos,<br>James<br>Tyler<br>Albert | 2/1/17 | 2/8/17 | 2 hr |

Appendix: A

Team Meeting Notes

**Date**: 27-Jan-2017

**Time**: 6:00 pm (Face to Face)

**Members Present:** Albert Hsu, Tyler Wang, Carlos Perez-Oviedo, James Pfeifer

**Topics Discussed**: Post Lab

---

**Objective:**

Today's focus was on evaluating our troubleshooting and beginning the progress report.

---

**To do/ Action Items:**

**-**Looking back portion.

-Assign roles.

-List out effective problem solving techniques.

---

**Decisions:**

-The group assigned roles and determined who would write each portion.

-The group defined different techniques to ensure success for future labs. Despite facing a great deal of problems from the different dated components, the group ultimately came up with a system for in depth trouble shooting.

---

**Reflections:**

-Meeting on a Friday evening was difficult since it was a more relaxed time, but was still overall productive.

-Future labs should run more smoothly due to the coherent understanding of how to troubleshoot every component in the system.

Appendix B

Sample Calculations:

**Carlos Perez-Oviedo**

**Sample Calculations**

**3030 Puller 0.27 amps 138.4 grams [Thrust] 2155 RPM 10% [Arduino]**

Power Input:

$P_{in}$=V*I*($P_{\%}$/100)

=7.4*0.27*(10/100)

=0.1998 watts

Power Output:

$P_{out}$=$T_c$*v

=(56.88)*2.8

=159.3/3.28*0.002205/550*745.7 = 0.145 Watts

Calibrated Thrust:

$T_c$=0.411*(T-$T_0$)

=56.88 g

Advance Ratio:

J=v/(RPM/60)*D

=2.8/(2155/60)*.072

=1.08

Propulsions Efficiency:

$\Pi_{sys}$=($P_{out}$/$P_{in}$)*100%

=(0.145/0.1998)*100%

=72.57%

## Sample Calculations: Tyler

Current: 0.82 amps

Trust Scale Reading: 161 grams

RPM: 6886 RPM

Arduino Power Setting: 40%

Find:

Power input

Power output

Calibrated thrust

Advance ratio

Propulsion efficiency

Solve:

Calibrated Trust = 0.411*(161-137) = 9.864 grams

Power input = 7.4*0.82*(40/100) = 2.4272 Watts

Power output = 9.864*10^-3*9.81*2.8=0.270668 Watts

Advance ratio = 2.8/((40/60)*0.0762) = 55.1181

Propulsion Efficiency = (0.270668/2.4272)*100% = 11.1515%

## Sample Calculations: James

Given:

| Current (amps) | Thrust Scale Reading (grams) | RPM (RPM) | Arduino Power Setting (%) |
|---|---|---|---|
| 0.99 | 171.5 | 8000 | 50 |

Calibrated Thrust= .411*(171.5-161)=4.3155 grams

Power input= voltage*current*power setting/100=7.4*.99*.5=3.663 Watts

Power output=calibrated thrust * wind tunnel velocity=4.3155*.28=1.28 Watts

Advance ratio= velocity * 100 *60/ (rpm*diameter)=2.8*100*60/(8000*.07620)=27.5

Propulsion Efficiency= (output power/input power)*100%= 100%*1.28/3.663=35%

## Sample Calculation - Albert

3030 Propeller – Puller configuration

$I = 1.07$  $T = 177.2$  $RPM = 8622$  $P_{\%} = 55$  $T_0 = 137$  $V = 7.4$  $v = 2.8$  $D = .0762$

$P_{in} = V * I * (P_{\%} / 100)$
$P_{in} = 7.4 * 1.07 * (55 / 100)$
$P_{in} = 7.918 * .55$
$P_{in} = 4.3549$ watts

$T_c = 0.411 * (T – T_0)$
$T_c = 0.411 * (177.2 – 137)$
$T_c = 0.411 * 40.2$
$T_c = 16.5222$ grams

$P_{out} = T_c * v$
$P_{out} = 2.8 * (T_c * .002205) / .224809$
$P_{out} = 2.8 * (16.5222 * .002205) / .224809$
$P_{out} = 2.8 * .03643 / .224809$
$P_{out} = 2.8 * .16206$
$P_{out} = .45375$ watts

$n_{sys} = 100 * P_{out} / P_{in}$
$n_{sys} = 100 * .45375 / 4.3549$
$n_{sys} = 100 * .10419$
$n_{sys} = 10.4194$ %

$J = v / (D * RPM / 60)$
$J = 2.8 / (.0762 * 8622 / 60)$
$J = 2.8 / (656.9964 / 60)$
$J = 2.8 / 10.95$
$J = 0.25571$ m/s

## Sample Code

Arduino Code

```
//Run all motors at 25% power for 2 seconds
motorSpeed(4,25);
goFor(2);

//Run all motors at 20% power for a total distance of 13.5 feet
motorSpeed(4,20);
goToAbsolutePosition(332);

//Reverse all motors
reverse(4);

Run all motors at a constant speed at 30% power for 1 second
motorSpeed(4,30);
goFor(1);

//Brake all motors
brake(4);
```

Table 1 - 3030 Propeller Puller Configuration

| Current (amps) | Thrust Scale Reading (grams) | RPM (RPM) | Arduino Power Setting (%) |
|---|---|---|---|
| **0.16** | 137 | 0 | 0 |
| **0.27** | 138.4 | 2155 | 10 |
| **0.35** | 140.7 | 3080 | 15 |
| **0.45** | 143.3 | 4000 | 20 |
| **0.54** | 146.5 | 4700 | 25 |
| **0.64** | 151.3 | 5500 | 30 |
| **0.73** | 155.3 | 6200 | 35 |
| **0.82** | 161 | 6886 | 40 |
| **0.9** | 165 | 7400 | 45 |
| **0.99** | 171.5 | 8000 | 50 |
| **1.07** | 177.2 | 8622 | 55 |
| **1.17** | 181.6 | 9221 | 60 |

Table 2 - 3030 Propeller Pusher Configuration

| Current (amps) | Thrust Scale Reading (grams) | RPM (RPM) | Arduino Power Setting (%) |
|---|---|---|---|
| **-0.11** | 146 | 0 | 0 |
| **0** | 142 | 1616 | 10 |

| | | | |
|---|---|---|---|
| **0.08** | 138 | 2514 | 15 |
| **0.18** | 134 | 3353 | 20 |
| **0.28** | 130 | 4191 | 25 |
| **0.38** | 125 | 4970 | 30 |
| **0.48** | 120 | 5868 | 35 |
| **0.57** | 115 | 6646 | 40 |
| **0.66** | 111 | 7365 | 45 |
| **0.75** | 106 | 8203 | 50 |
| **0.85** | 99 | 8982 | 55 |
| **0.93** | 93 | 9700 | 60 |

Table 3 - 3020 Propeller Puller Configuration

| Current (amps) | Thrust Scale Reading (grams) | RPM (RPM) | Arduino Power Setting (%) |
|---|---|---|---|
| **0.04** | 160 | 0 | 0 |
| **0.16** | 159 | 1916 | 10 |
| **0.25** | 156 | 2934 | 15 |
| **0.34** | 155 | 4071 | 20 |
| **0.43** | 154 | 5089 | 25 |
| **0.51** | 152 | 6227 | 30 |
| **0.59** | 152 | 7365 | 35 |
| **0.65** | 151 | 8622 | 40 |
| **0.68** | 150 | 9700 | 45 |
| **0.74** | 149 | 10898 | 50 |
| **0.78** | 147 | 12155 | 55 |
| **0.79** | 147 | 13293 | 60 |

Table 4 - 3020 Propeller Pusher Configutation

| Current (amps) | Thrust Scale Reading (grams) | RPM (RPM) | Arduino Power Setting (%) |
|---|---|---|---|
| **0** | 16.55 | 0 | 0 |
| **0.13** | 162 | 1800 | 10 |
| **0.22** | 162 | 3000 | 15 |
| **0.32** | 162 | 4200 | 20 |
| **0.4** | 163 | 5200 | 25 |
| **0.49** | 164 | 6400 | 30 |
| **0.57** | 165 | 7600 | 35 |
| **0.51** | 168 | 9800 | 40 |
| **0.66** | 186 | 9800 | 45 |
| **0.72** | 171 | 11100 | 50 |
| **0.75** | 173 | 12300 | 55 |
| **0.8** | 175 | 13300 | 60 |

Figure 1

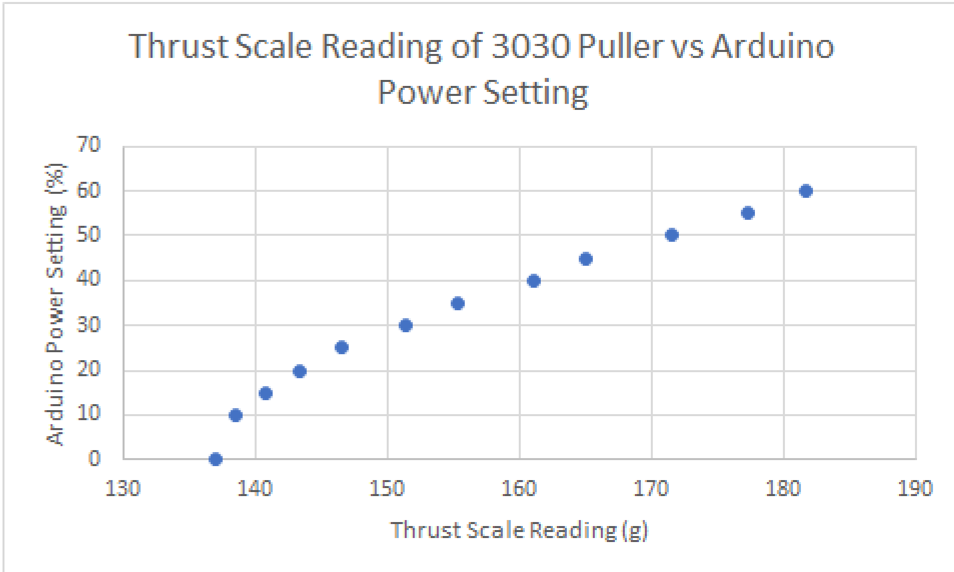
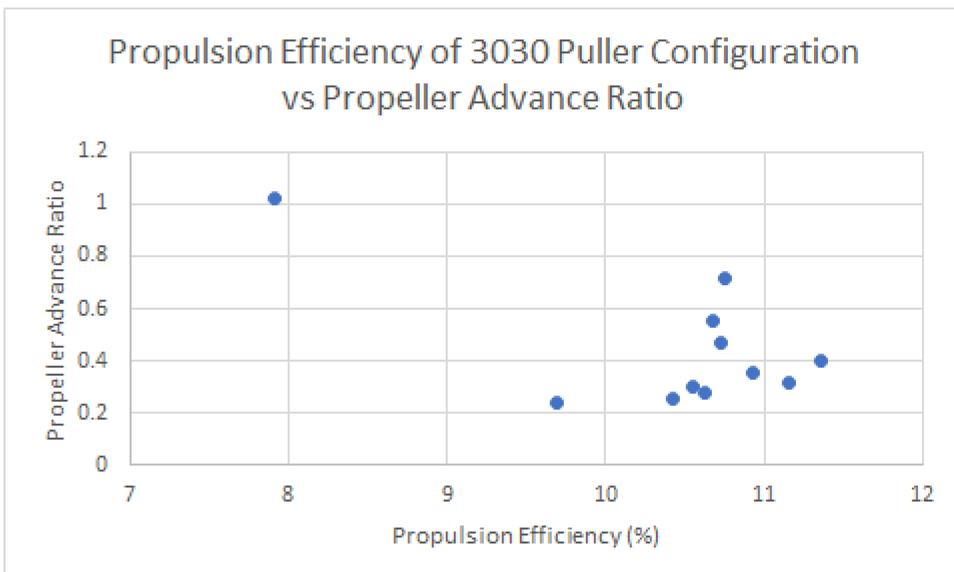

Thrust Scale Reading of 3030 Puller vs Arduino Power Setting
Arduino Power Setting (%)
0
10
20
30
40
50
60
70
130
140
150
160
170
180
190
Thrust Scale Reading (g)

Figure 2


Propulsion Efficiency of 3030 Puller Configuration vs Propeller Advance Ratio
Propeller Advance Ratio
0
0.2
0.4
0.6
0.8
1
1.2
7
8
9
10
11
12
Propulsion Efficiency (%)