Lab 10 Progress Report

Group F – Abbey Hamilton, Merveille Kavota, McKenzie Kennelly, and Xinjie Li

Instructor-Professor John Schrock, GTA-Rahel Beyene 03/31//2017

Lab 9 - Performance Test 2:

Situation:

In performance test 2 of lab 9, the team created two different codes that met the requirements outlined in the Mission Concept Review. The MCR simply laid out a scenario that the AEV must complete on the overhead track. Essentially, the AEV would start at the drop-off area, travel to the gate, traverse through the gate, travel to the cargo, pick up the cargo, travel back through the gate to the drop off area, and stop. Both code solutions created by the team should successfully complete this scenario. However, by creating two different codes, the team can choose certain criteria that they value most to focus on and analyze. This criteria may include energy efficiency, difference between similar command calls, flexibility to various tracks, and so on. Performance test 2 allowed for the team to choose the best of two codes to use in performance test 4, the final testing of the AEV on the overhead track.

Results and analysis:

The team chose to create two codes that focused on the functioning of the propellers. This further would be analyzed to identify which code minimized the AEV's energy the most. Going into performance test 2, the team had created two distinguishable codes. The first code ran both 3030 propellers, in the pusher configuration, as the AEV went forward on the track to the cargo, and then ran both 3030 propellers, in the puller configuration, as the AEV reversed and traversed back to the drop off area with the cargo. The second code ran the top 3030 propeller, in the puller configuration, as the AEV traveled forward, and then ran the bottom 3030 propeller, also in the puller configuration, as the AEV traveled back to the drop-off area. However, the team, due to time constraints and a reflectance sensor malfunction, was never able to run the AEV successfully around the track.

When creating the two codes, the team decided to use certain command calls in both of the code solutions. To allow the AEV to travel to the desired distance needed to travel, such as to the gate or the cargo area, the *goToAbsolutePosition()* command call would be used. This would allow for the team to have control of exactly where the AEV traveled to. To stop the AEV when needed, the team decided to set *motorSpeed()* to 0 at approximately 50 marks before the actual stopping mark so the AEV could slow to a stop instead of using the *brake()* command call. By slowly coasting to a stop rather than abruptly stopping with *brake(),* the AEV will run smoother on the track. To control how long the AEV stops for, the team used the *goFor()* command call.

Working with the *goToAbsolutePosition()* command call was a challenge for the team. To allow for the AEV to travel exactly the distance desired, the team had to calculate the estimated number of marks that they concluded it would travel. To do this, the team utilized Figure 1, shown below.

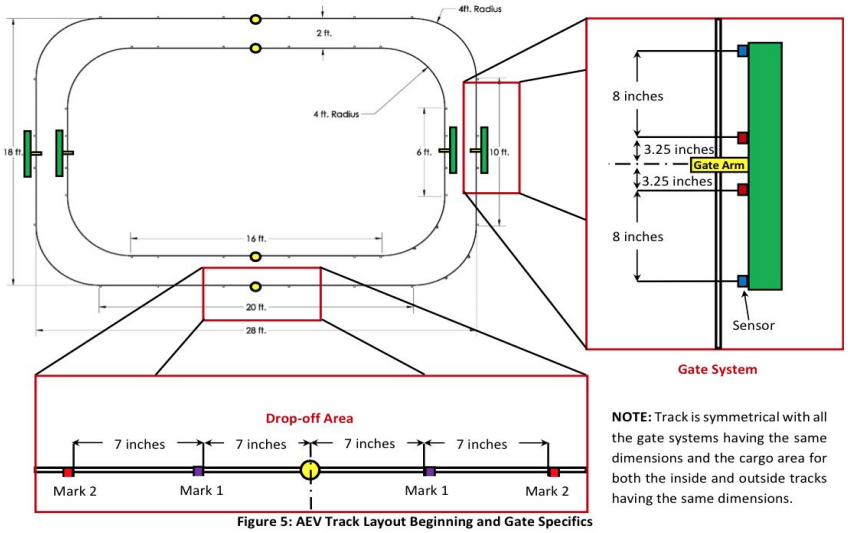

Figure 1: AEV Track Layout

The team also utilized their knowledge that 8 marks are recorded for one full wheel revolution. This further allowed the team to use the conversion factors shown below.

$$0.4875\ inches\ =\ 1\ mark$$
$$0.040625\ feet\ =\ 1\ mark$$

The team could then calculate the amount of feet that the AEV would need to travel to each stop, such as to the gate and the cargo, and then use the conversion factors to calculate the number of marks that that distance equalled. However, the team was aware that Figure 1 was not going to be a completely accurate distance measurement of the track layout as the track has often been mishandled, bent, and altered. Therefore, going into performance test 2, the team had planned to make minor changes to their code regarding the marks calculated, whether they had to decrease or increase them. The first code the team constructed is displayed in Table 1, along with descriptions as to how each command meets the MCR requirements.

Table 1: Code 1 MCR Breakdown

Arduino Code	MCR Requirements
reverse(4);	AEV will move forward.
motorSpeed(4, 30);	AEV starts at drop off area and begins to move.
goToAbsolutePosition(309);	Goes to mark 260 (≈ 10 feet 6 inches).
motorSpeed(4, 0);	Propellers turn off.
goToAbsolutePosition(359);	Slowly coasts to a stop right before the gate at mark 359 (≈ 14 feet 6 inches).

goFor(15);	Stops for 7 seconds plus an extra 8 seconds for the coasting to allow for the gate to open.
motorSpeed(4, 30);	Moves at speed of 30.
goToAbsolutePosition(653);	Goes to mark 653 (≈ 26 feet 6 inches) .
motorSpeed(4, 0);	Propellers turn off.
goToAbsolutePosition(708);	Slowly coasts to a stop right before the cargo at mark 708 (≈ 28 feet 9 inches).
goFor(10);	Stops for 5 seconds plus an extra 5 seconds for the coasting to verify that the cargo is attached.
reverse(4);	Reverses the AEV's direction.
motorSpeed(4, 35);	Moves at speed of 35. (Increase in speed to account for cargo's weight)
goToAbsolutePosition(458);	Goes to mark 458 (≈ 18 feet 7 inches).
motorSpeed(4, 0);	Propellers turn off.
goToAbsolutePosition(408);	Slowly coasts to a stop right before the gate at mark 408 (≈ 16 feet 6 inches).
goFor(15);	Stops for 7 seconds plus an extra 8 seconds for the coasting to allow for the gate to open.
motorSpeed(4, 35);	Moves at speed of 35.
goToAbsolutePosition(50);	Goes to mark 50 (≈ 2 feet).
motorSpeed(4, 0);	Propellers turn off.
goToAbsolutePosition(0);	Slowly coasts to a stop at drop-off area at mark 0 (≈ 0 feet).

The team's first run of their first code on the track did not go as planned. When the team ran their AEV on the overhead track, the AEV smashed into the first gate and continued to run for about 15 seconds after it hit the gate before stopping. This is not as the team had thought the AEV would run. The mark the AEV was set to travel to using the *goToAbsolutePosition()* call was 359. To hopefully get the AEV to stop at the gate on the next run, the team decreased the mark to 300. The AEV still ran into the gate. Numerous times after these attempts, the team lowered the mark hoping the AEV would stop. It was not until the team set the mark to 112 that the AEV came even close to stopping. This was obviously incorrect as the team had calculated the distance the AEV would need to travel to correctly. After discussing with class administrators, the team was told that their reflectance sensors were sticking out too far on the arm and

that their reflective tape on their wheel was old. Therefore, the team pushed their sensors in tighter into the wheel, and put new tape on their wheels. Then, the team decided to see if this solved the problem by running a reflectance sensor test. With this test, the team could assure that a single rotation of the wheel yields 8 marks. After running the test, it was confirmed this was true and the reflectance sensors now worked properly.

Essentially, going back to the original code, the team ran *goToAbsolutePosition()* with the mark set at 359. The AEV still ran too long; however, it functioned correctly as a mark in the 300s was accurate. After lowering the mark a number of times again, the team found that the 310 mark worked the best to allow for the AEV to stop at the gate. As mentioned before, the team stopped the AEV's propellers at mark 260 so the AEV could slowly coast to a stop.

Because of the problem with the reflectance sensors, the team was never able to test the entire code, nevertheless both codes. This meant the team did not do any testing in performance test 2 regarding the propellers and their energy efficiency. However, the team plans to continue to test both codes on March 31.

Takeaways from Lab 9:

1. Arduino - Team created 2 different codes that met most of the MCR requirements.
2. Reflectance sensors - Team identified the importance of positioning the sensors correctly in regards to the command call *goToAbsolutePosition()*.
3. Wheel - Team recognized that wheel reflective tape was old and preventing the sensors from working correctly.
4. Time management - Team saw importance of time management as they did not complete all of their goals set for performance test 2.

Lab 10 - Performance Test 3:

Situation:

Due to the team's inability to test the two Arduino codes in performance test 2, the team will begin testing of the two codes during performance test 3. This may affect the AEV's overall energy efficiency, as testing during performance test 3 will focus on completing a full run on the track and attaching to the cargo caboose rather than perfecting energy efficiency. To complete the full run, the team aims to perfect the number of marks the AEV needs to travel to stop when desired at required stops such as the gate, the cargo, and the drop-off area. Although the team will focus on operational consistency and efficiency for a majority of performance test 3, the team will still focus on energy efficiency by working with the propellers.

In order to decide which of the two codes will be chosen as the final code that will be improved upon and used during performance test 4, the team will test both modified codes from performance test 2. As stated before, the first of these two codes utilizes both propellers as the AEV travels from the start to the cargo area and then back to the starting point. By using both propellers, the configurations of the propellers change dependent on the direction of the AEV. With this code, as the AEV travels towards the cargo area,

both propellers function as pusher configurations. However, as the AEV travels away from the cargo area and back towards the starting area, both propellers function as puller configurations.

The second of the two codes will focus on using the puller configuration as the AEV travels in both directions. In order to do so, only one propeller will be used when the AEV is travelling towards the cargo area. As the AEV travels away from the cargo area and towards the starting point, the other propeller will be used. Although this method will only allow for one propeller to be used, the propeller used will be in puller configuration at all points on the track, which the system analysis of propulsion efficiency proved to be the configuration that was the most energy efficient.

Examining the propellers allows the team to see which code runs smoother on the track and as well make analyzations about the AEV's energy efficiency on its run for each code. To see which code runs smoother, the team will simply visually inspect each run and note any differences. These differences may include difficulty in pulling the cargo, stopping at the gate, or moving at the set power. To test energy efficiency, the EEPROM data will be collected for each code. The data can then be imported into the Design Analysis Tool to create the power versus time graph for each code. By analyzing the graphs, the team can conclude which code minimizes more energy by identifying which graph has a smaller area underneath the curve. A smaller area means the run was more energy efficient. The team specifically looking at the propellers and their energy efficiency, whilst developing a fully working code with all commands moving and stopping the gate at desired distances will ensure that the AEV meets all MCR requirements when it is run in performance test 4. Overall, by the end of performance test 3, the team will have an energy efficient, fully functioning code.

Weekly goals:

1.) Decide which code will be chosen to further develop and finalize for performance test 4 (final testing)
2.) Update the Project Portfolio
3.) Work on the extra credit video

Weekly schedule:

When it is time to write the progress report for lab 11, the entire team will meet on Wednesday, April 5 from 8 P.M. to 10 P.M. in the Thompson Library. They will complete the progress report and prepare for performance test 4.

Lab 10 Progress Report

Performance Test 3

No.	Task	Start	Finish	Due Date	Est Time	Abbey Hamilton	Merveille Kavota	McKenzie Kennelly	Xinjie Li	% Complete
1	AEV Design 1 Construction	1-February	21-March	21-March	8 h	0.5 h	1 h	0.5 h	1 h	100
2	Lab 10 Progress Report	21-March	30-March	31-March	5h	3h	3h	3h	3h	100
3	Performance Test 2	22-March	28-March	28-March	8h	3h	3h	3h	3h	100
4	Preliminary Design Report	21-February	24-March	24-March	10h	5h	1h	5h	3h	100
5	Performance Test 3	31-March	Before 5-April	5-April	8h					0
6	Oral Presentation Draft - Final Project Report	24-March	Before 5-April	5-April	5h					0
7	Lab 11 (PT4) Progress Report	31-March	Before 7-April	7-April	5h					0
8	Oral Presentation - Final Project Report	14-April	Before 19-April	19-April	8h					0
9	Final Project Report (CDR)	14-April	Before 19-April	19-April	10h					0
10	Project Portfolio	18-Januaty	Before 21-April	21-April	20 h	0h	0h	8h	0h	40
11	Extra Credit Video	15-February	Before 28-February	21-April	20 h	0h	3h	0h	3h	25

Appendix:

Team Meeting Notes

Date: 03/30/2017
Time: 8PM (Face-to-Face)
Members Present: Abbey Hamilton, Merveille Kavota, McKenzie Kennelly, and Xinjie Li
Topics Discussed: Lab 9 Performance Test 2 and Lab 10 Performance Test 3

Objective:
Today's main focus was to work on the progress report for lab 10 and update the project portfolio..

To do/Action Items:
- Decide which of the two codes will be the final code used for the AEV by the end of Performance Test 3 (Abbey H, Merveille K, McKenzie K, Xinjie L)
-Continue updating project portfolio (Abbey H, McKenzie K)

Decisions:
- The final code will be based off of which of the two codes is more likely to complete a run on the track while fulfilling MCR requirements, even if it is less energy efficient than the other code.

Reflections:
-Before PT3, the team must be capable of using the Design Analysis Tool quickly.
-Before PT3, the team must have a good understanding of how the marks work to successfully create the final functioning code.

Arduino Code

Code 1

```
reverse(4); // both motors in forward direction
motorSpeed(4, 30); // both motors move at speed of 30%
goToAbsolutePosition(309); // move to mark 309
motorSpeed(4, 0); // both motors set to speed of 0% and coast to stop at gate
goToAbsolutePosition(359); // move to mark 359 and stop
goFor(15); // stop for 15 seconds
motorSpeed(4, 30); // both motors move at speed of 30%
goToAbsolutePosition(653);  // move to mark 653
motorSpeed(4, 0); // both motors set to speed of 0% and coast to stop
goToAbsolutePosition(708); // move to mark 708 and stop to pick up cargo
goFor(10);  // stop for 10 seconds
reverse(4); // both motors reverse to move backwards direction
motorSpeed(4, 35); // both motors move at speed of 35%
goToAbsolutePosition(458); // move to mark 458
motorSpeed(4, 0); // both motors set to speed of 0% and coast to stop
```

```
goToAbsolutePosition(408); // move to mark 408 and to stop at gate
goFor(15);  // stop for 15 seconds
motorSpeed(4, 35); // both motors move at speed of 35%
goToAbsolutePosition(50); // move to mark 50
motorSpeed(4, 0); // both motors set to speed of 0% and coast to stop
goToAbsolutePosition(0); // move to mark 0 and stop at drop-off area
```

Code 2

```
reverse(4); // both motors in forward direction
motorSpeed(1, 30); // top motor moves at speed of 30%
goToAbsolutePosition(309); // move to mark 309
motorSpeed(1, 0); // top motor set to speed of 0% and coast to stop at gate
goToAbsolutePosition(359); // move to mark 359 and stop
goFor(15); // stop for 15 seconds
motorSpeed(1, 30); // top motor moves at speed of 30%
goToAbsolutePosition(653);  // move to mark 653
motorSpeed(1, 0); // top motor set to speed of 0% and coast to stop
goToAbsolutePosition(708); // move to mark 708 and stop to pick up cargo
goFor(10);  // stop for 10 seconds
reverse(4); // both motors reverse to move backwards direction
motorSpeed(2, 35); // bottom motor moves at speed of 35%
goToAbsolutePosition(458); // move to mark 458
motorSpeed(2, 0); // bottom motor set to speed of 0% and coast to stop
goToAbsolutePosition(408); // move to mark 408 and to stop at gate
goFor(15);  // stop for 15 seconds
motorSpeed(2, 35); // bottom motor moves at speed of 35%
goToAbsolutePosition(50); // move to mark 50
motorSpeed(2, 0); // bottom motor set to speed of 0% and coast to stop
goToAbsolutePosition(0); // move to mark 0 and stop at drop-off area
```