Victor Lorenzo B. Tirona	23 October 2019
154292	ELC 151.2

Laboratory Activity 5 Results

1. What have you observed on the behavior of the Mobot's motors?

Inputting the same value for Arduino's analogWrite() function may not necessarily lead to the same rotational velocity for each motor. This implies that the Mobots' motors need to be calibrated first to determine the relation of the Mobot's first motor and the second motor.

Not only this, even after calibrating the motors, there were times that our motors had to be calibrated again as it would start skewing the other way.

2. What are the difficulties you encountered on the experiment and how did you solve it?

As mentioned in the previous question, the Mobot's motors needed to be calibrated first. We solved this through trial and error and determining the analog inputs that would drive the motors to output the same rotational velocity.

Another difficulty was that, some Mobot's wheels were not horizontally fixed. As the wheel rotated, the wheel would move side to side with small displacement. As the mobot would move forward, this would cause its direction to change. The change in direction was minute but at long distances, would translate into the mobot leaving the line guide. Our solution for this was to have the Mobot move forward with a small skew to the right and then with a skew to the left. This made the change in direction caused by the faulty wheel to be insignificant.

It is important to note that while our solution did work, it did not work all the time. Reaching the 3m and 5m marks was partially relied on luck and trial-and-error. T his is significant to note as it expresses the difficulty of using an open-loop system. With no use of sensors, the change of direction caused by the faulty wheel could not be corrected.

3. What other ways can you think of to make your open-loop system work more smoothly and efficiently?

Our system would probably have worked better if we replaced our Mobot with one with more sturdy wheels. At the same time, finding a way to plot the calibration line to relate the rotational velocity of the two motors would allow for a more reliable method than just trial-and-error. This would allow us to decrease and increase the speed without having to determine the calibrated values via trial-and-error.

Appendix

Code for 3-meters

```
nt i;
int run1 = 9; // 0 to 255, 255 = max, 0 = stop
int run2 = 10; // 0 to 255, 255 = max, 0 = stop
int dir1 = 8;
int dir2 = 11;
int LS1, LS2, LS3, col1, col2, col3;
void setup() {
 Serial.begin(9600);
 for (i = 2; i <= 7; i = i + 1)
 pinMode(i, INPUT); //sensors
 for (i = 8; i <= 11; i = i + 1) {
 pinMode(i, OUTPUT); //motor control
 }
 delay(10);
 //stop motors
 for (i = 8; i <= 11; i = i + 1)
 digitalWrite(i, LOW);
}
void loop() { //main routine here
 // motor1
 digitalWrite(dir1, HIGH); //direction control of motor1, 1 = forward
 analogWrite(run1, 170); //speed control of motor1, 0 =stop, 255 =fastest RIGHT
 // motor2
 digitalWrite(dir2, HIGH); //direction control of motor2, 1 = forward
 analogWrite(run2, 180); //speed control of motor2, 0 =stop, 255 =fastest LEFT
 delay(10000);

  digitalWrite(dir1, LOW); //direction control of motor1, 1 = forward
 analogWrite(run1,0); //speed control of motor1, 0 =stop, 255 =fastest
 digitalWrite(dir2, LOW); //direction control of motor1, 1 = forward
 analogWrite(run2, 0); //speed control of motor1, 0 =stop, 255 =fastest
 delay(2000);


 while (1);
}
```

CODE for 5-meters

```
int i;
int run1 = 9; // 0 to 255, 255 = max, 0 = stop
int run2 = 10; // 0 to 255, 255 = max, 0 = stop
int dir1 = 8;
int dir2 = 11;
int LS1, LS2, LS3, col1, col2, col3;
void setup() {
 Serial.begin(9600);
 for (i = 2; i <= 7; i = i + 1)
 pinMode(i, INPUT); //sensors
 for (i = 8; i <= 11; i = i + 1) {
 pinMode(i, OUTPUT); //motor control
 }
 delay(10);
 //stop motors
 for (i = 8; i <= 11; i = i + 1)
 digitalWrite(i, LOW);
}
void loop() { //main routine here
 // motor1
 digitalWrite(dir1, HIGH); //direction control of motor1, 1 = forward
 analogWrite(run1, 170); //speed control of motor1, 0 =stop, 255 =fastest RIGHT
 // motor2
 digitalWrite(dir2, HIGH); //direction control of motor2, 1 = forward
 analogWrite(run2, 180); //speed control of motor2, 0 =stop, 255 =fastest LEFT
 delay(11000);

  digitalWrite(dir1, LOW); //direction control of motor1, 1 = forward
 analogWrite(run1,0); //speed control of motor1, 0 =stop, 255 =fastest
 digitalWrite(dir2, LOW); //direction control of motor1, 1 = forward
 analogWrite(run2, 0); //speed control of motor1, 0 =stop, 255 =fastest
 delay(2000);
  digitalWrite(dir1, HIGH); //direction control of motor1, 1 = forward
 analogWrite(run1, 190); //speed control of motor1, 0 =stop, 255 =fastest RIGHT
 // motor2
 digitalWrite(dir2, HIGH); //direction control of motor2, 1 = forward
 analogWrite(run2, 170); //speed control of motor2, 0 =stop, 255 =fastest LEFT
 delay(5500);

  digitalWrite(dir1, LOW); //direction control of motor1, 1 = forward
 analogWrite(run1,0); //speed control of motor1, 0 =stop, 255 =fastest
 digitalWrite(dir2, LOW); //direction control of motor1, 1 = forward
 analogWrite(run2, 0); //speed control of motor1, 0 =stop, 255 =fastest
 delay(2000);



 while (1);
}
```