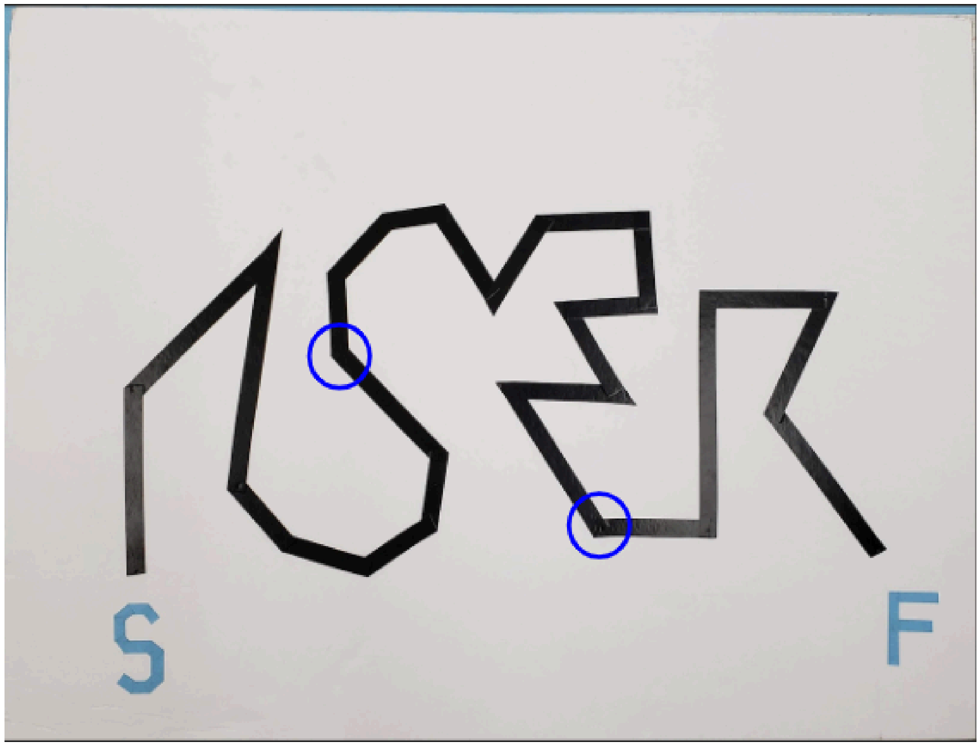
16-311 Lab 3 Writeup

Goals:

1. Get familiar with the MotorGo kits
   - You need to be confident in the use of these tools for future labs
2. Implement a basic driving program to enable smooth motion with two motors
3. Use sensors (a light sensor in this case) to gather real-time environment data that the robot can use
4. Tweak parameters to fine-tune your line-following algorithm
   - Trade-off between reliability and speed
5. Use odometry to estimate the position of the robot.

Part 1: Line Following:

- Build a robot that can follow a **dark line** on a **light plane**.
- Your robot will be placed at the beginning of the line and **must follow** that line to the end within **65 seconds**. This time starts when you put your robot on the course and let go of it. Having a calibration routine is encouraged.
- You can choose to follow either the **left** or **right** edge of the line (or both).
- **TAs** will place your robot at the start with anywhere between **0º** and **30º** from the direction of travel. (You pick which direction the 30˚ is.) Your robot should be able to find the line from this starting position and proceed.
- Your robot does not need to know when it has completed the course. You can manually stop it.
- There are two boards with the same course. During demo sign ups you will sign up for a slot on one of the boards. You get **1 trial only**.
- You have 4 minutes to complete the line-following part of the demo. Anything not shown in that time gets a 0.



Checkpoints are at the blue circles shown in the image above.

Task	Item	Points	Trial 1
Line Following	Checkpoint 1	+10	
	Checkpoint 2	+15	
	Finish	+15	
	Time Penalty	-1 per sec (> 65 sec)	
	Total	**40**	

- Optional Video Submission (Line Following ONLY)
    - Applicable if you get less than a 20/40 on Line Following during demo day
    - Submit a video of a ***successful*** line following run to get **half of the points you would have gotten during demo**
    - Show **stopwatch** in the screen for verification
    - Deadline: **midnight before demo day** (submission link will be posted on Piazza)

Part 2: Odometry:

> # Odometry
>
> From Wikipedia, the free encyclopedia
>
> **Odometry** is the use of data from motion sensors to estimate change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location. This method is sensitive to errors due to the integration of velocity measurements over time to give position estimates. Rapid and accurate data collection, instrument calibration, and processing are required in most cases for odometry to be used effectively.

- Students will place the robot at its starting point during the demo. This starting point is considered x = y = theta = 0, with the robot facing along the x-axis.
- You will be given **3 left-right motor power pairs** that your robot must execute in order for **3 seconds each.** Ex: (0.3, 0.75), (-0.5, 0.5), (-0.3, 0). Pausing between commands is allowed. Each command will be between -1 and 1, inclusive.
- Your robot must **print an estimate of its final position** relative to its starting point (print the x and y coordinates, in inches). You will tell us your FINAL LOCATION ((x, y) pair) AFTER you run all 3 motor pairs. You will need to be connected to your robot the entire time you run the odometry trials. This is because we would like you to print the FINAL LOCATION on the command window or terminal on your computer while connected to the Raspi. We will sanity-check your robot's movement during the demo and reserve the right to check your code if your robot appears to not be printing the final location properly.
- You will be graded based on how close this is to the robot's actual position in **L2**. Unless told otherwise before the demo, we will measure the position of the center of the wheel base by placing a piece of tape at each wheel contact point, using a ruler between the tape, and marking the center between these two.
- You are allowed to ramp up/down your power as long as you reach the power-pairs that the TAs give to you and you maintain the rule of "When moving in a straight line **at power (1, 1), your robot must go at least 1 foot in 3 seconds.**"
- The demo area will be at least 50% larger than is needed for a robot going at minimum speed. Going too much faster may result in hitting an obstacle.
- You will get **5 minutes** to demo **three** trials. Your score will be **the average of the 3 trials.**

Task	Item	Points	Trial 1	Trial 2	Trial 3
Odometry	Predicted x,y Position	+60			
	Actual x,y Position				
	Precision Penalty	-2 per 0.5 in. L2 (rounds in students' favor)			

<u>Lab Rules and Restrictions</u>:

- You must build your robot **from scratch**.
- Your robot must be no longer than **1 foot in length** and **1 foot in width**.
- **You may only use the LEGO components in the given kits and MotorGo kits.** You may use **only 1 light sensor** and **exactly two motors**.
- You are **not allowed to use anything that will leave residue** on the LEGOs or the Raspi cases. Such items include but are not limited to:
    - Super glue, Hot glue, Duct tape, Masking tape, Clay, etc
- Unless we're told otherwise before the odometry portion of the demo, the **center of the wheelbase** is the reference point from where measurements will be made
- You will have **2 minutes to modify** your robot between the line-following and odometry parts of the demo (optional).
- The line-following boards will be placed in a fixed position in the REL, and that's where they will be for the demos (we will mark those positions on the floor). We recommend you **don't move the boards** as differences in lighting conditions may impact your robot's performance.

<u>Tips on Line Following</u>:

- Line-following sounds like an easy algorithm, BUT it can take some time to tune if you want to go very fast.
- Don't attempt to bypass the line-following, as you will not receive any credit if your robot doesn't attempt to follow the line.
- Consider how the position of the light sensor on your robot affects the precision of its readings.
- Consider how the design of your robot (wheel size, weight, width) affects the accuracy of your odometry.

- To combat the difference in lighting conditions, consider a calibration method.

Tips on Odometry:

- See the equations here as a starting point
- See these Runge-Kutta slides for more intuition regarding numerical integration.
- Basic euler integration should be fine given a sufficiently low Δt if Runge-Kutta is difficult.
- For quick distance estimation: The floor tiles in the REL are 1 foot by 1 foot
- **Save your Lab 3 code**, you will want to expand on it for Labs 5 and 6.