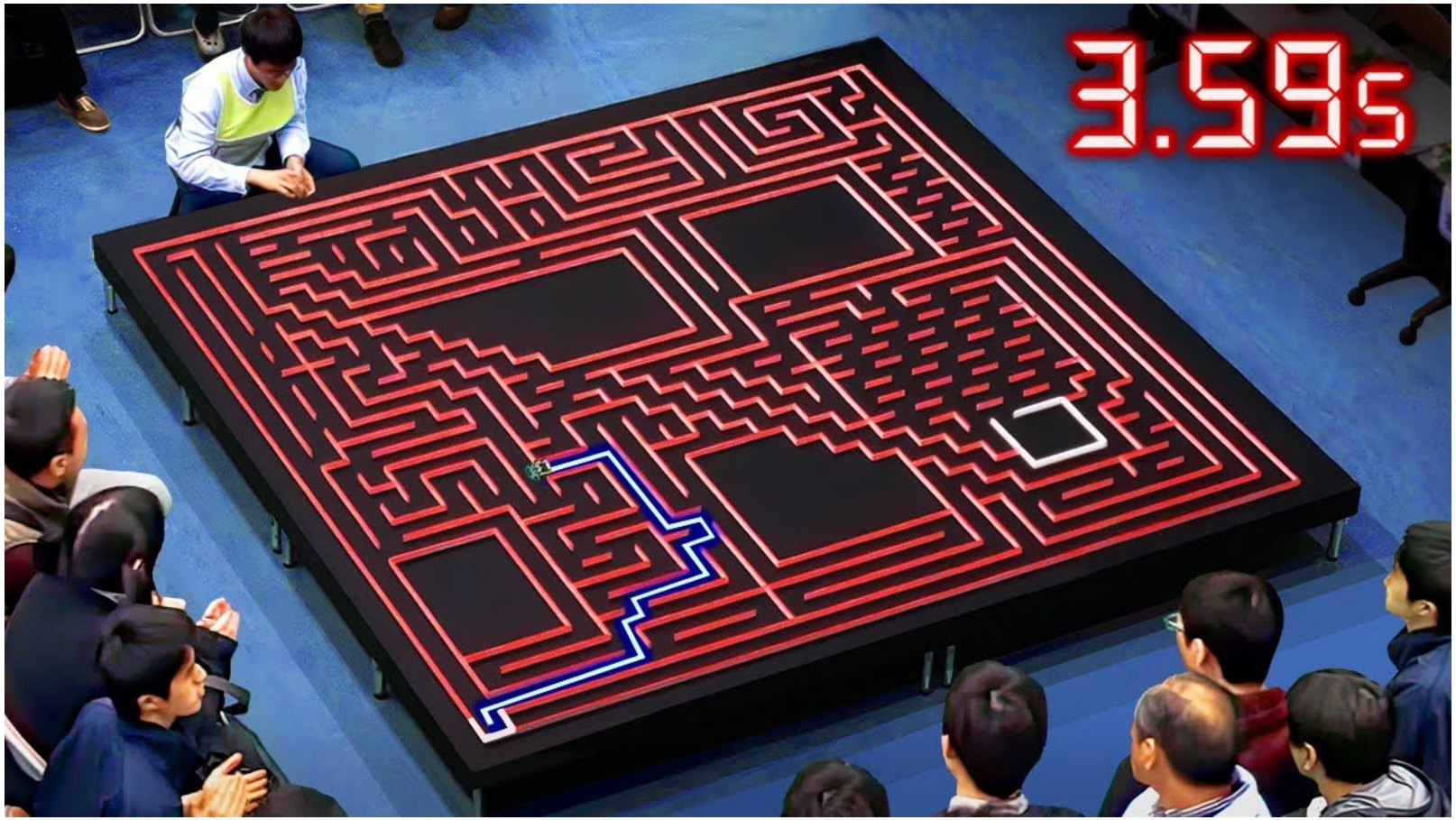
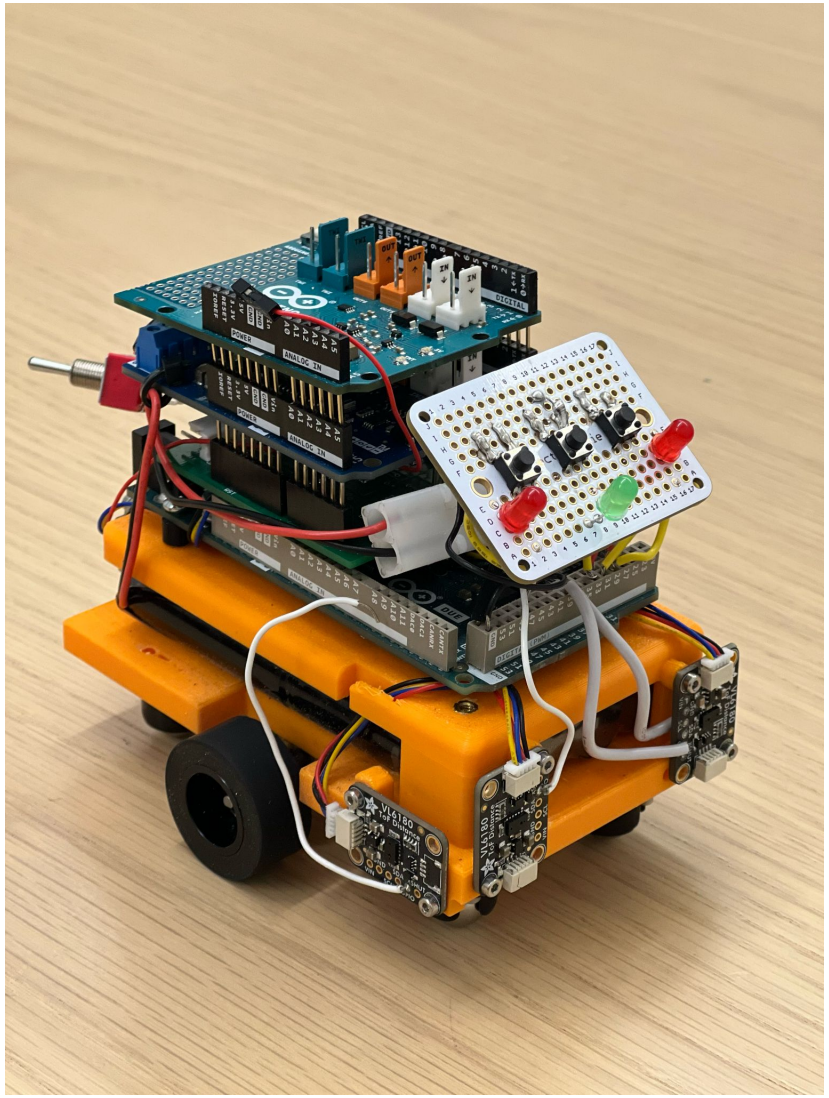
ENGS 147 - Mechatronics

Chingpheng Phoun, Justin Sapun, Vuthy Vey

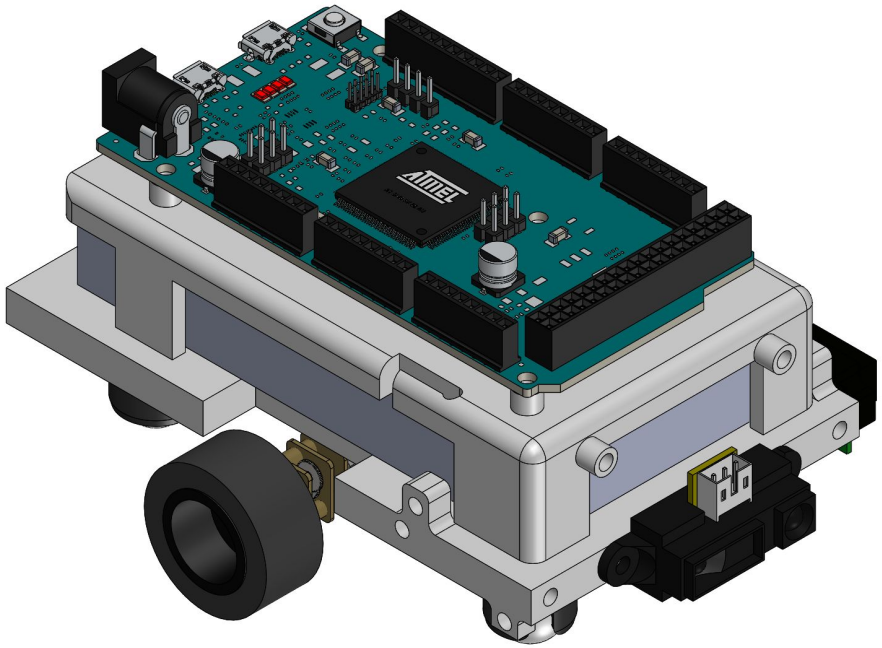
Introduction

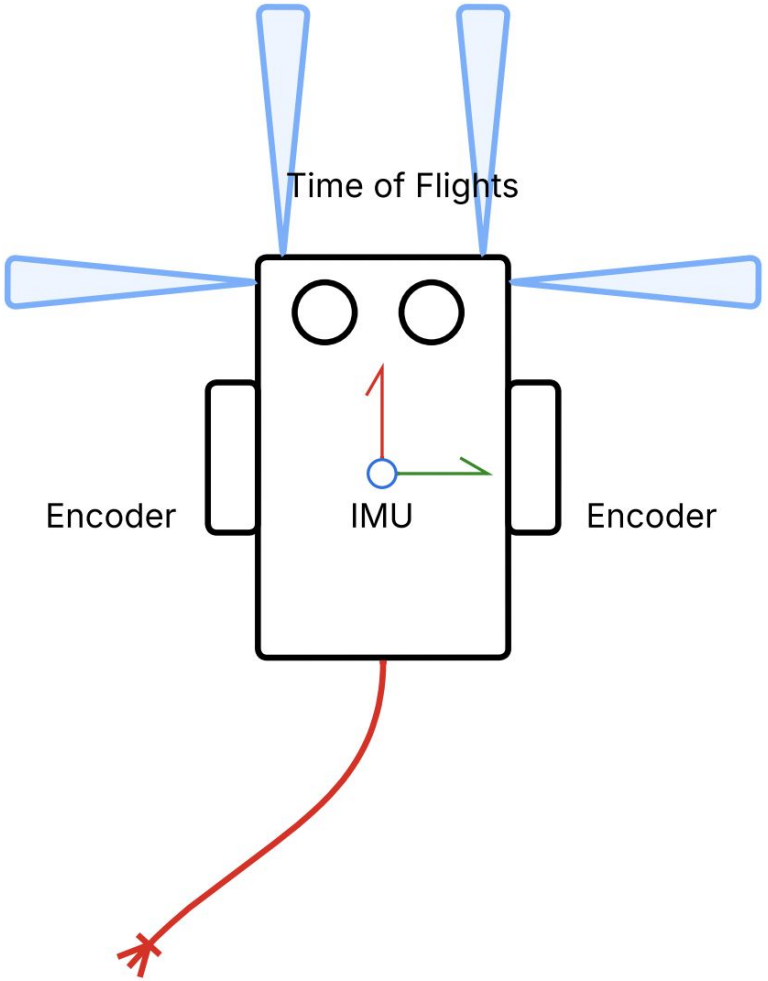


Introducing Remy!

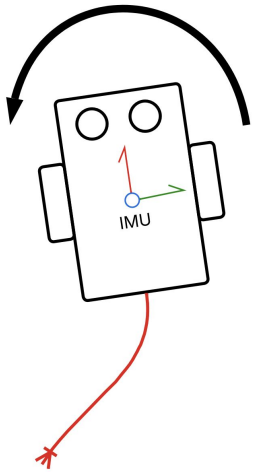


Hardware & Sensors





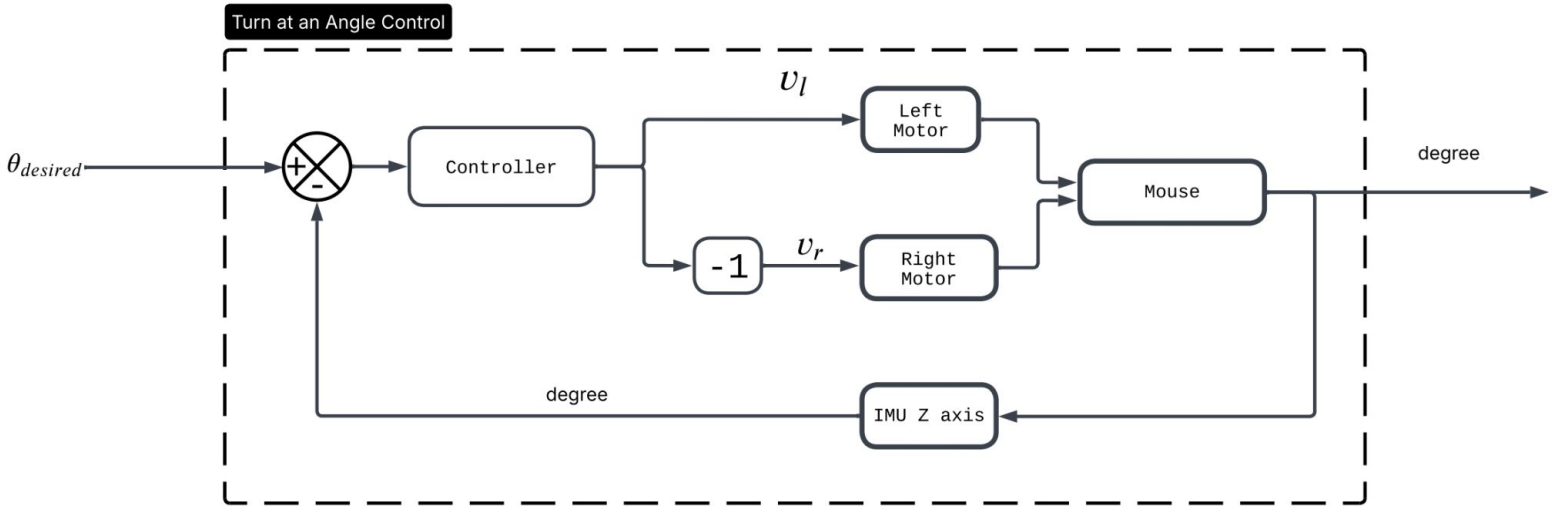
Turning Control



Plant:

```
G =

          68
--------------------
0.49 s^2 + 1.68 s + 1
```



Compensator for 90:

```
Gc =

   0.49 s^2 + 1.68 s + 1
--------------------
        s^2 + 8 s
```

with an overall gain of `K = 0.25`
and Sampling Time `Ts = 0.05s`.

Compensator for 180:

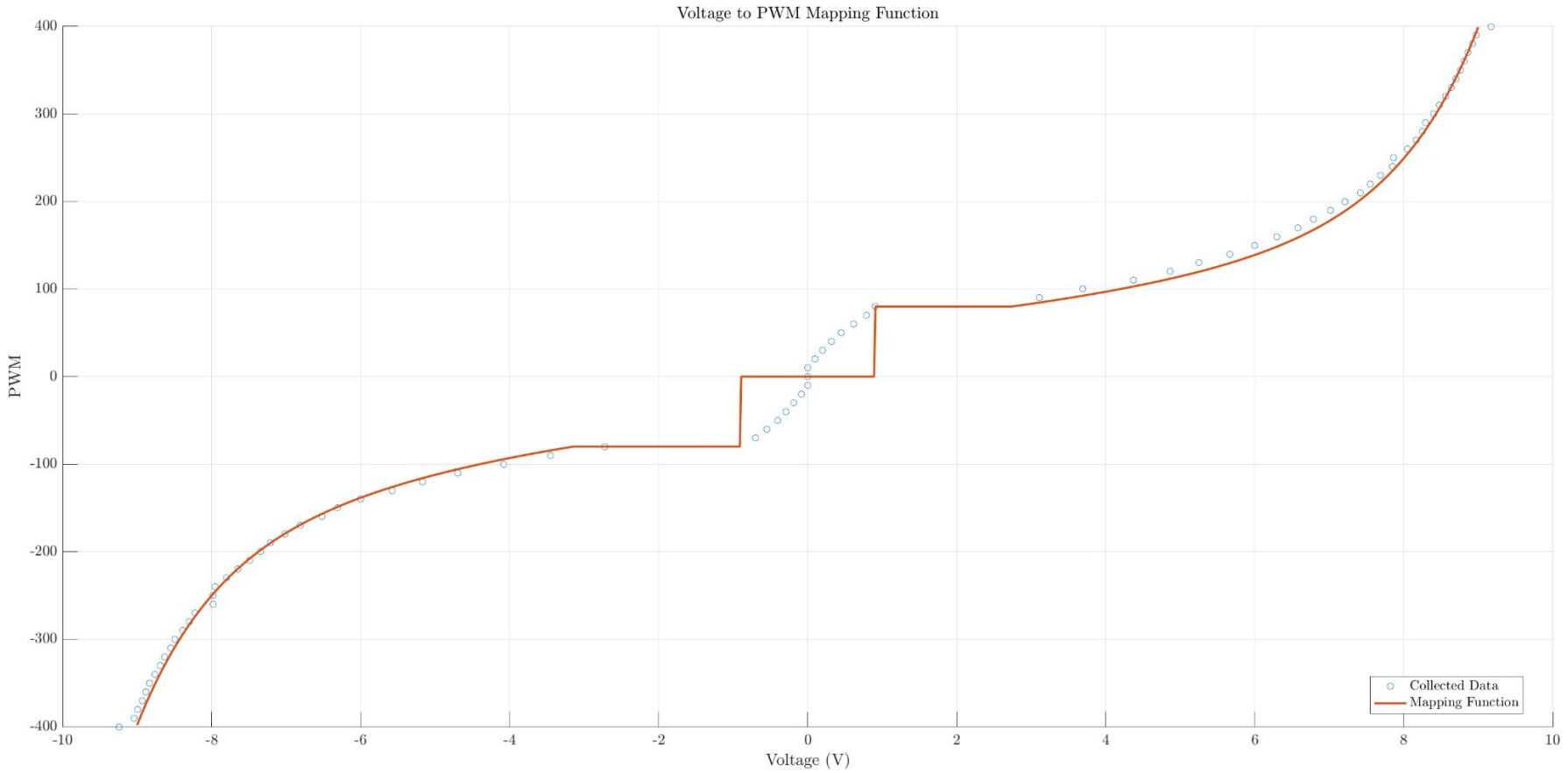
```
Gc =

   0.49 s^2 + 1.68 s + 1
--------------------
        s^2 + 6.3 s
```

with an overall gain of `K = 0.10`
and Sampling Time `Ts = 0.05s`.

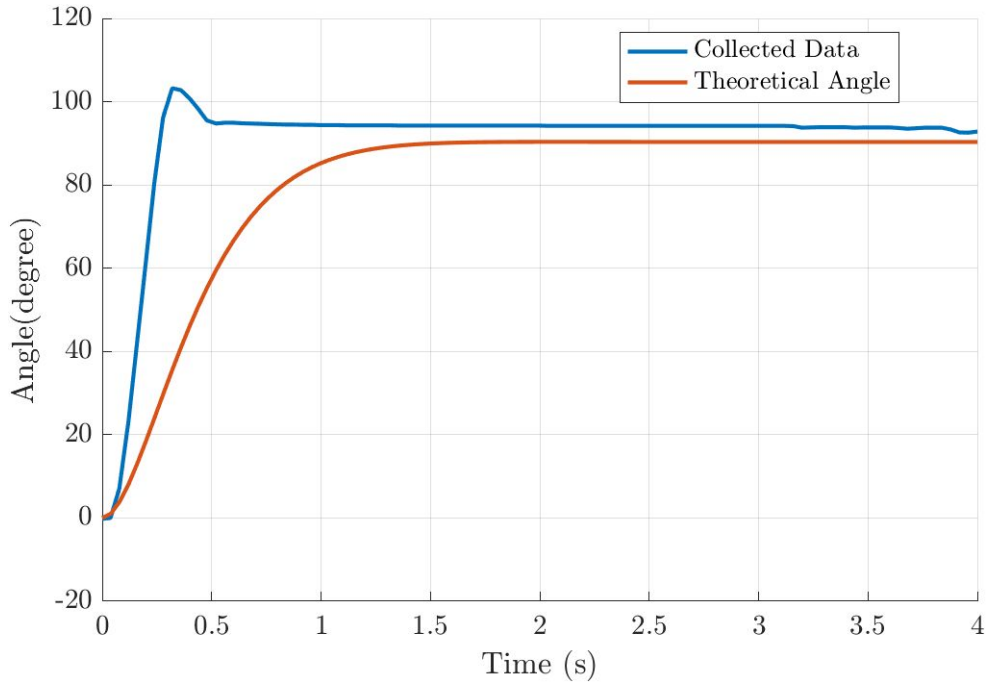
Stiction is Our FRIEND!

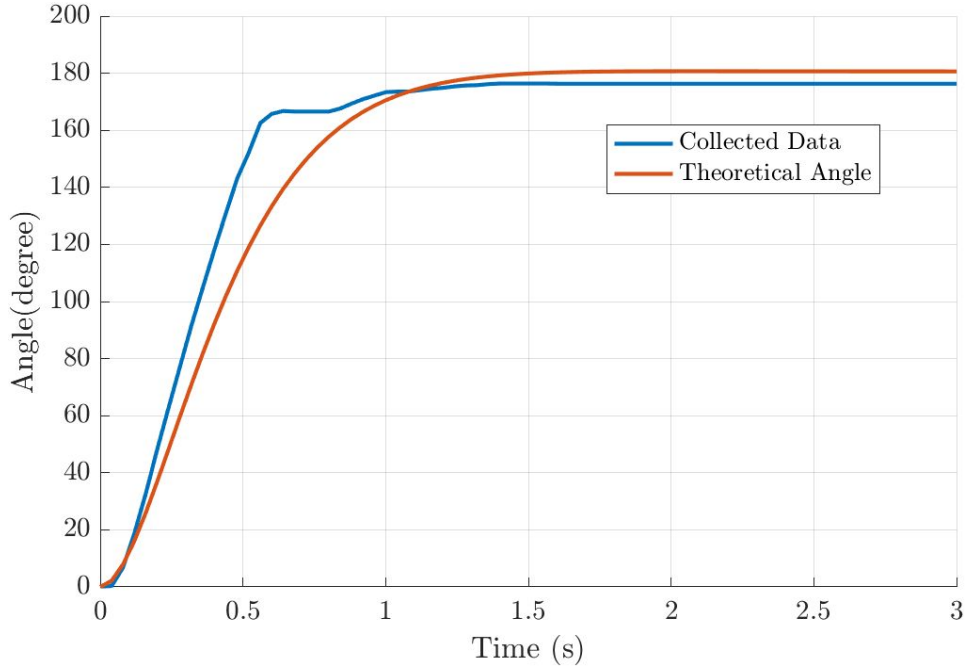
Linearizing Voltage To PWM



- No Rotation between PWM signal is between -80 to 80.
- There is a deadzone in the control system between -0.9 to 0.9 in error values.

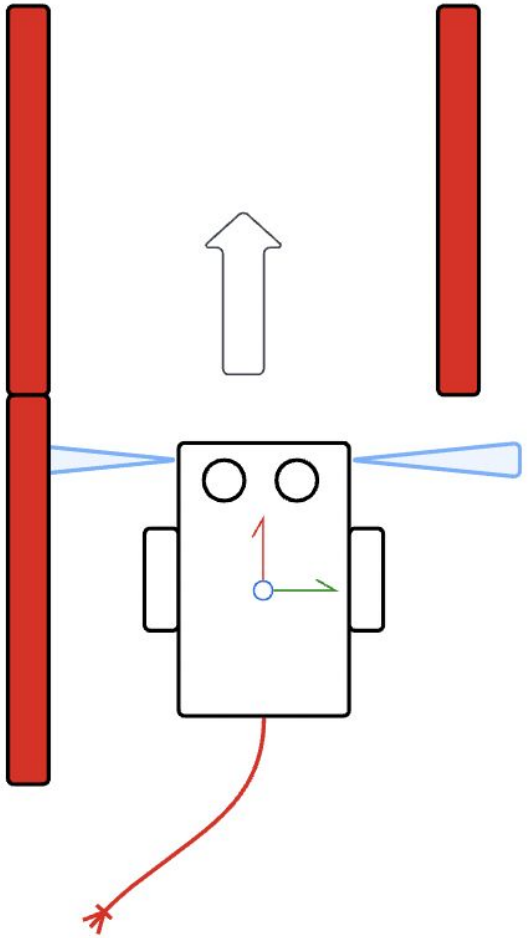
Turning Control

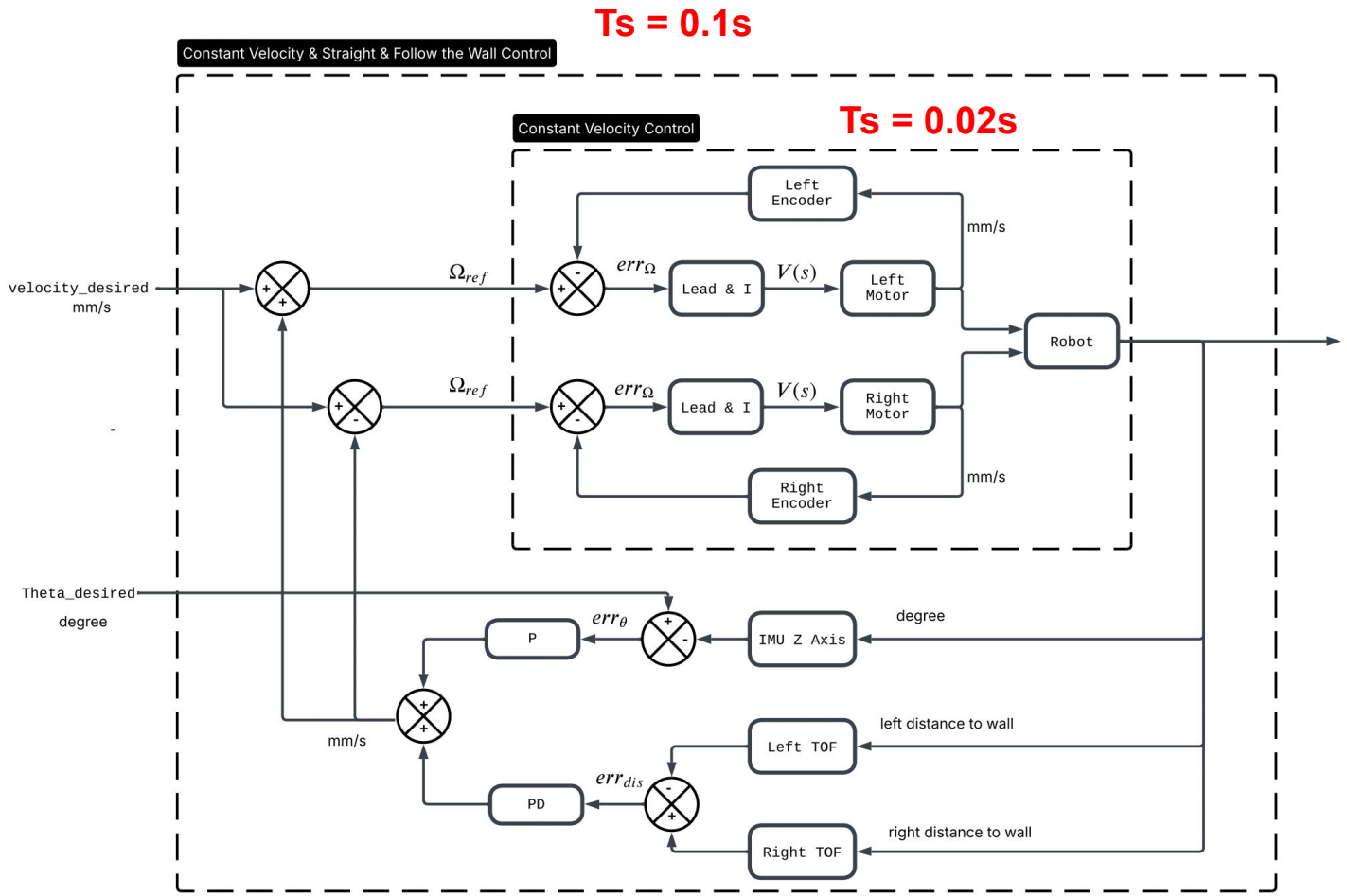





```
if (abs(angle_err_0) < 3 and abs(angle_err_1) < 3 and abs(angle_err_2) < 3) {
    done = true;
    stop_motors();
}
```

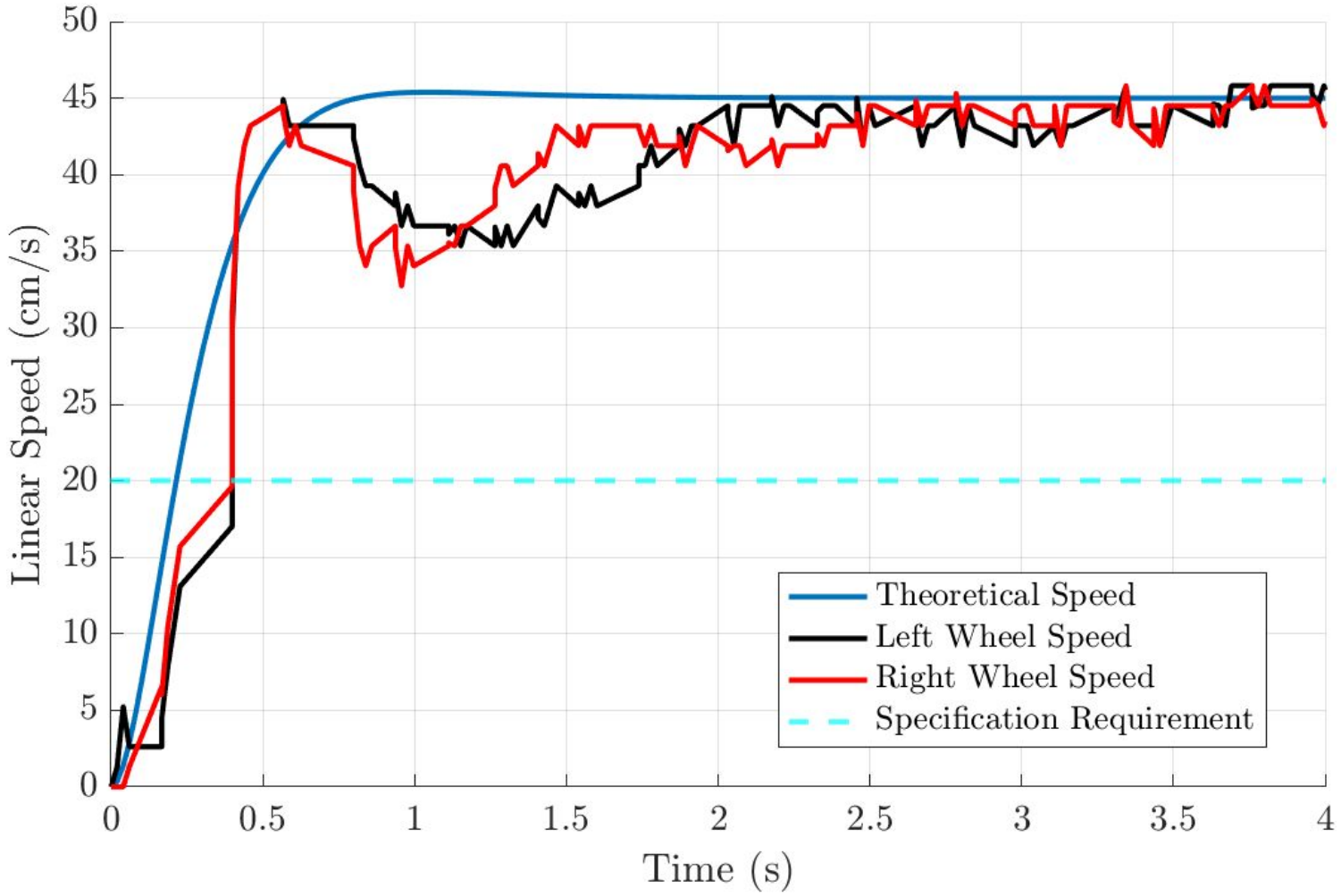
Constant-Velocity Control



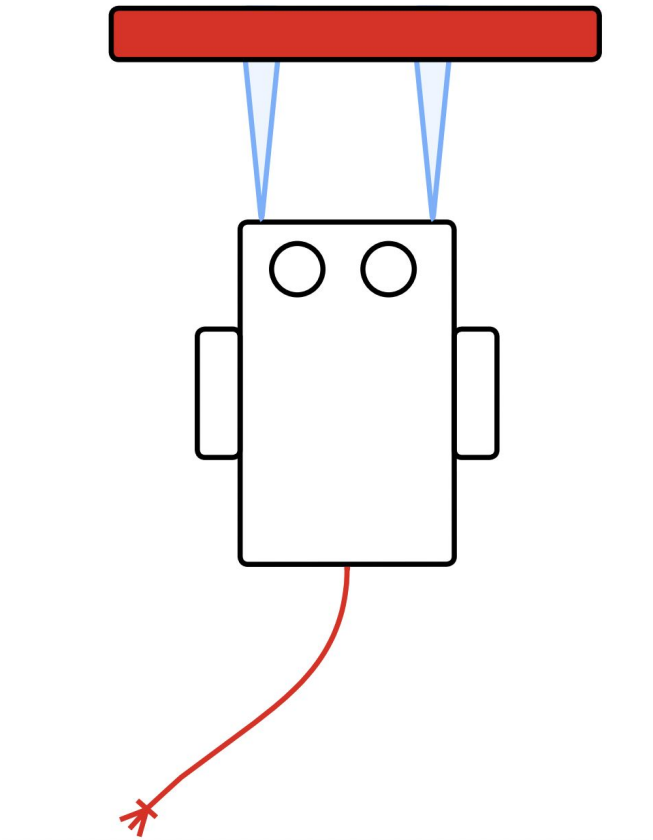


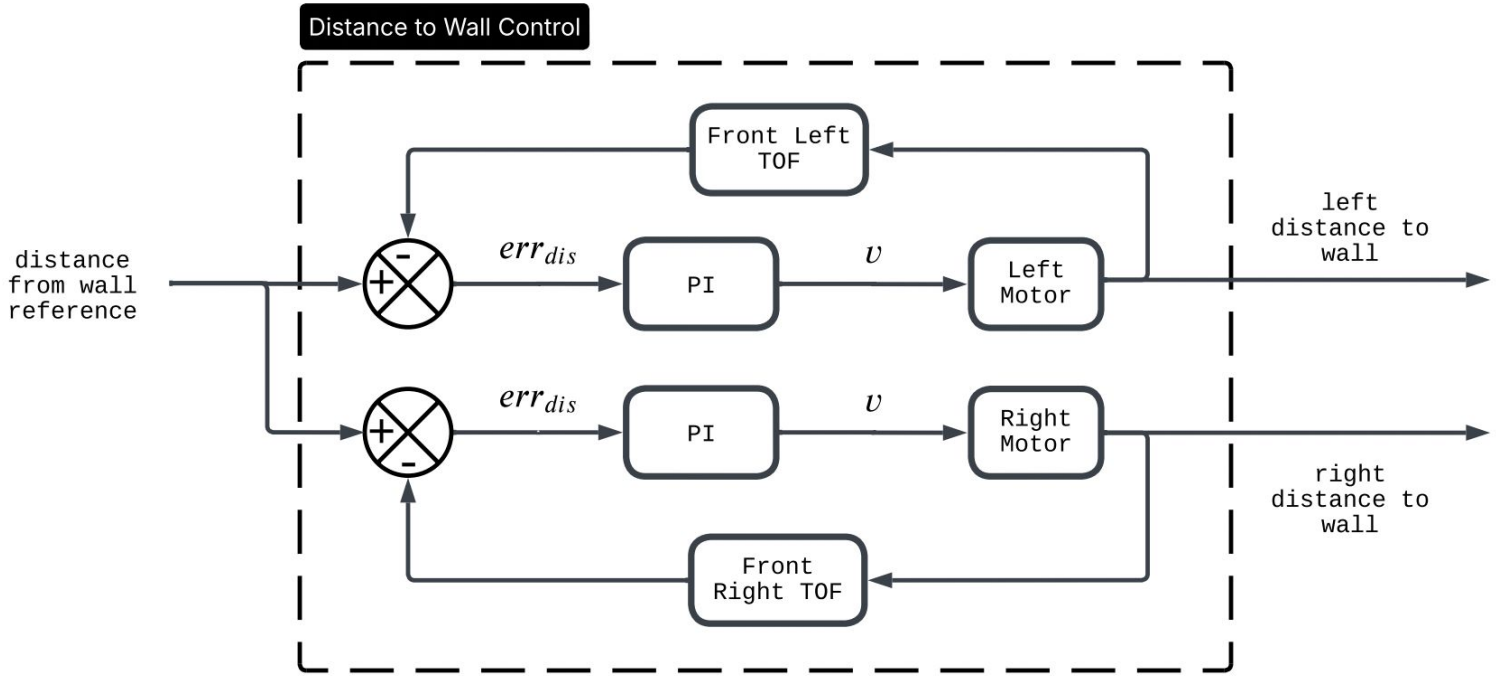
Constant-Velocity Control

Reference Velocity:
4.50 cm/s



Distance to Wall Control



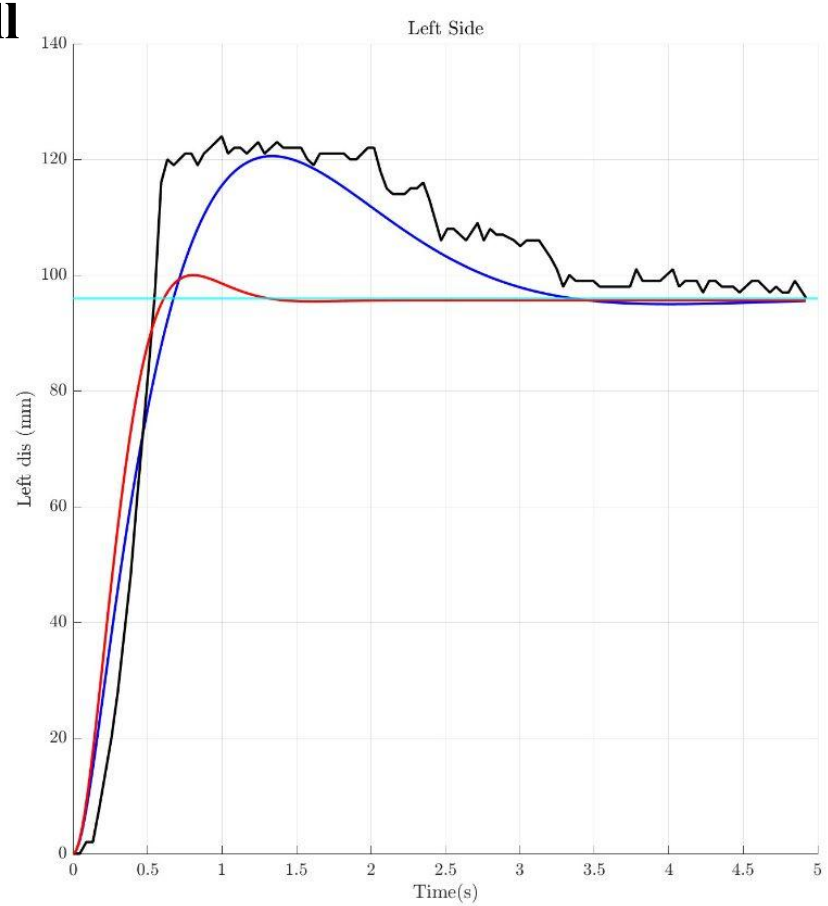


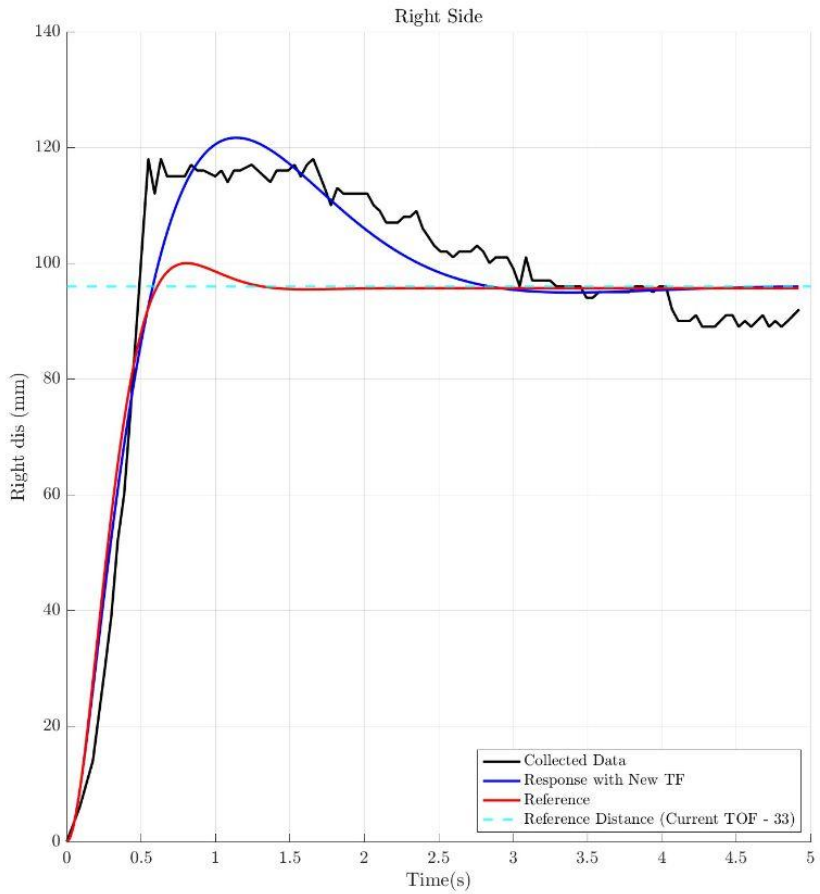
Distance to Wall Control

Target Distance to Wall
35 mm

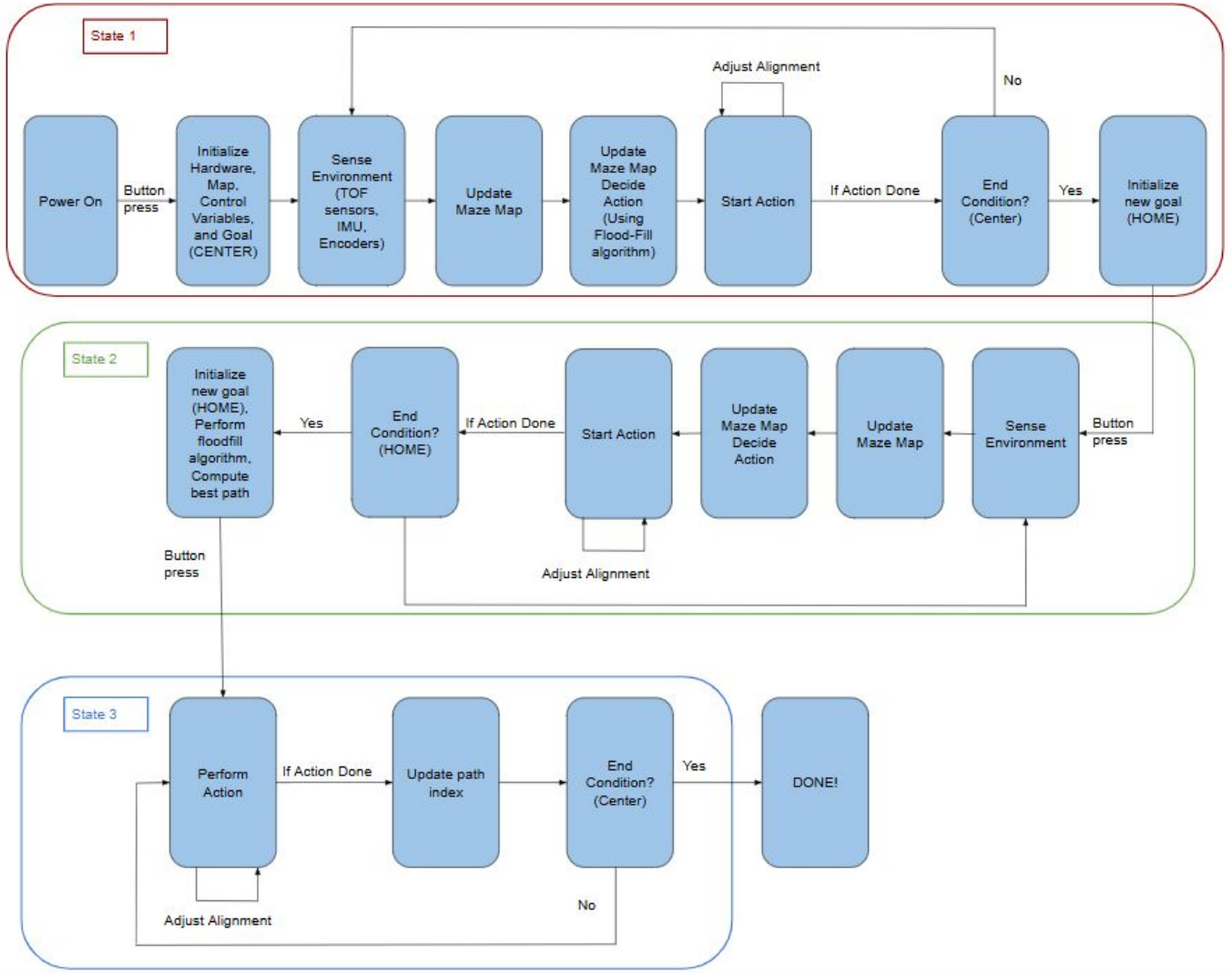
Start at from Wall
120 mm

Need to Travel
95 mm

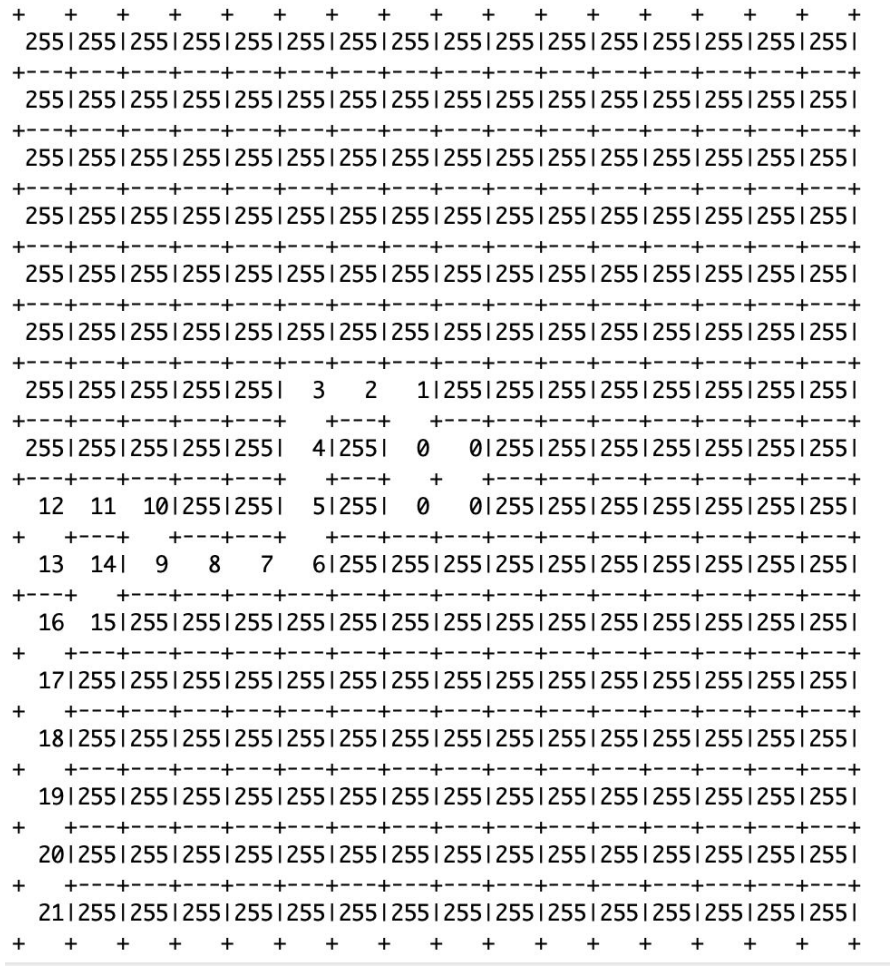




High Level Task Planning



Flood-Fill Algorithm



How it works

- Starts from goal cell and expands outward assigning increasing values until it reaches end cell
- Favors lower valued neighboring cells when exploring
- “Floods” path during exploration
- Next move is chosen by comparing adjacent flood values

Path Planning Integration

- Function that generates move codes (0=Left, 1=Right, 3=Forward) with index to give our robot a known path it can optimize for speed

Thank you!

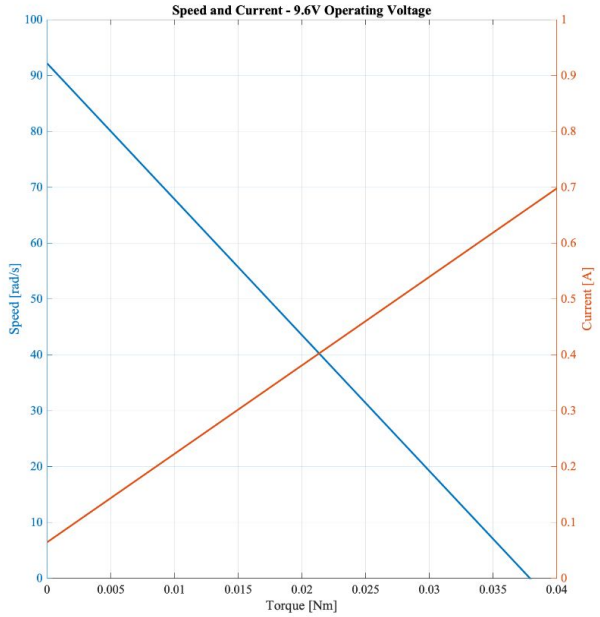
Design Specification

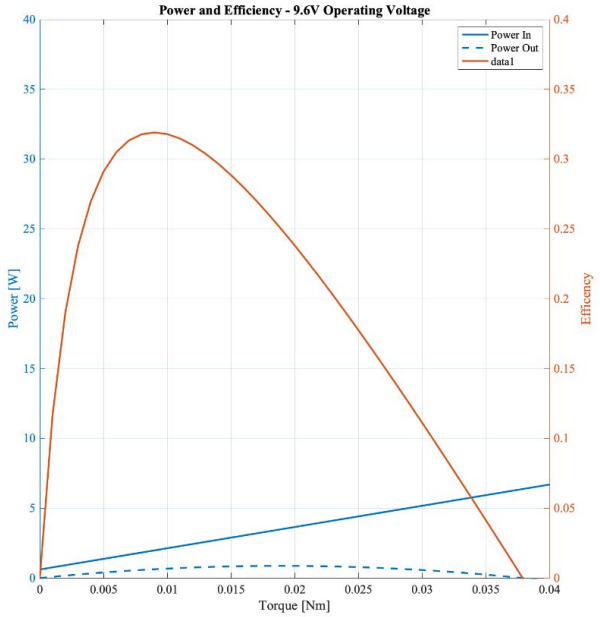
Requirement	MoSCoW	Method	Metric	Criteria	Justification	Complete or NOT?
Complete the Maze	M	Full Maze Test	Success rate	100%	Core functionality	Yes
Move straight, and don't hit the side walls	M	Path tracking test	Distance from wall	>2.54cm	Obstacle avoidance	Yes
IR sensors wall detection	M	4–30 cm Sensor test	Distance measurement error	±1 cm	Obstacle avoidance	Sometimes, we do still hit the front wall

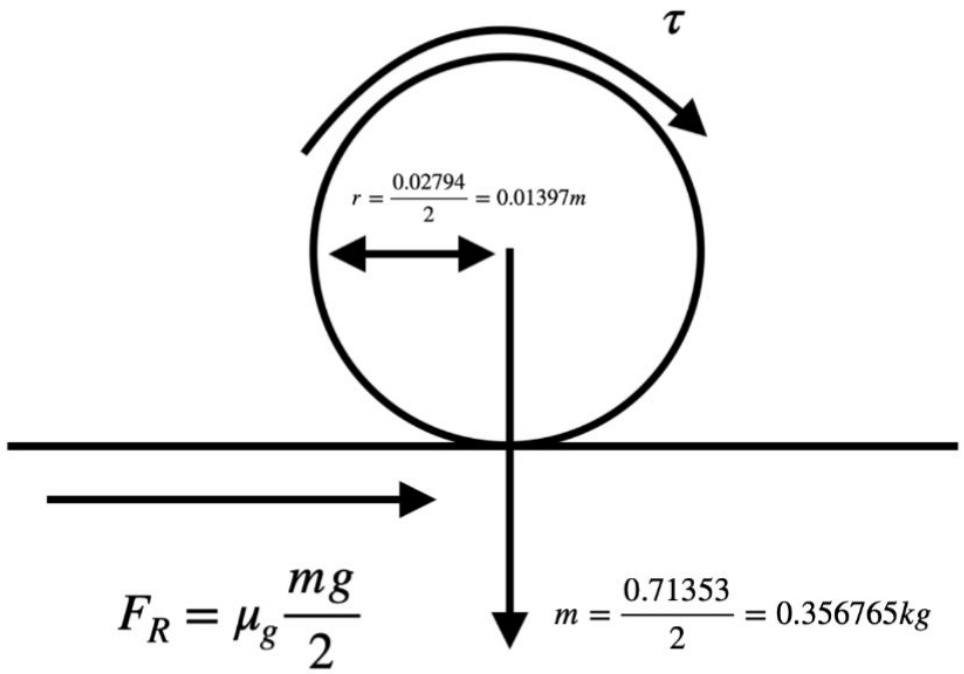
Design Specification

Requirement	MoSCoW	Method	Metric	Criteria	Justification	Complete or NOT?
Constant linear Velocity on the straight	S	Video Capture	Steady state velocity	±5%	Minimize the time to solve the maze	Yes, *except that we actually double our initial specification*
Distance Control	M	Motion test	Start to Final distance	±5%	Minimize position error	Yes
Turn 90 degrees accurately	M	Motion test	Turn angle error	±5°	Kinematic turning	Mostly, *about 80% of the time*
Flood-fill algorithm for maze-solving	M	Simulation	Accuracy	100%	Optimized pathfinding efficiency	Yes

Motor Characterization

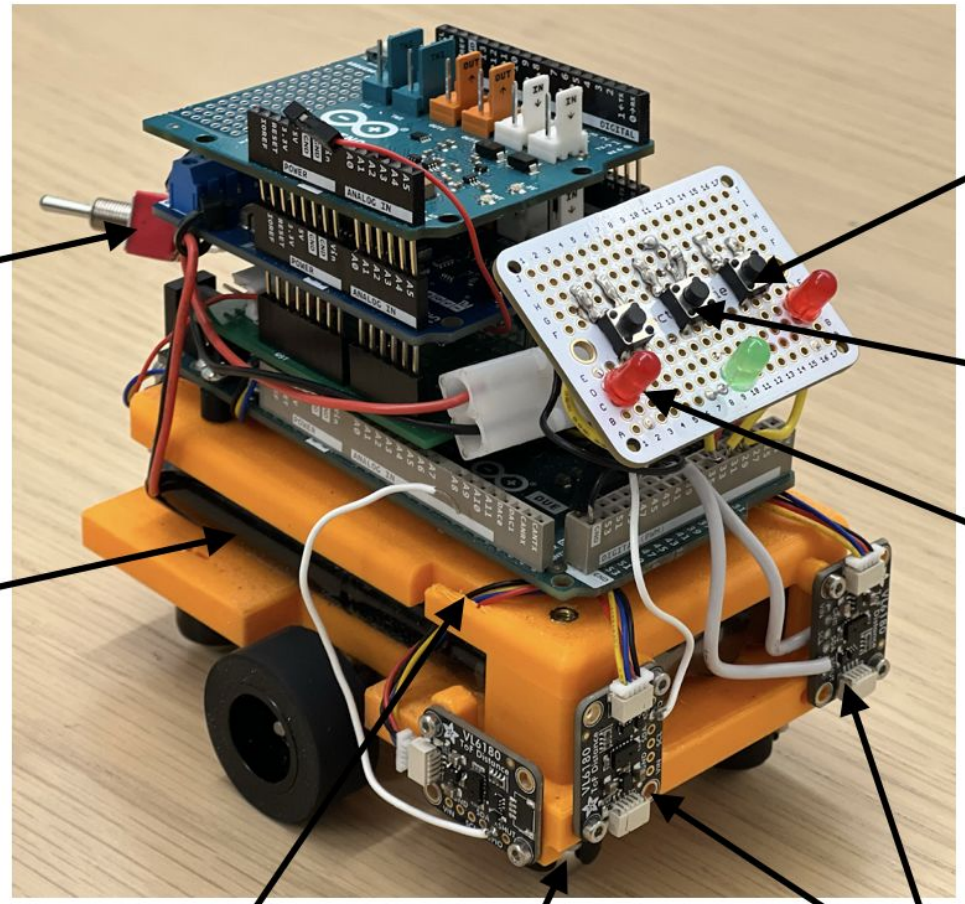






$$\tau = 0.002444655458025N \cdot m$$

$$\omega = 86.2098 rads^{-1} \Rightarrow v = 1.2069 ms^{-1} \simeq 4.34484 \frac{Km}{h} \simeq 2.698 mph$$



ON/OFF switch

Push to Start Switch (Instead of letting the code run straight away)

For logging data/switching to race mode after flood-fill algorithm

All the LED lights allows us to see if the robot see the walls during the flood-fill

Battery as low as possible to keep the center of gravity low and allow for more stability

Tiny slots for wire management

Fours Ball Casters

Two TOFs to make sure that the robot is perfectly straight before turning.

DARTMOUTH
ENGINEERING
THAYER SCHOOL

engineering.dartmouth.edu