CSCI 212
FINAL EXAM

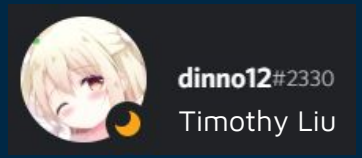
By
GROUP #3
Adam Amer, Timothy Liu,
Cody McKinney, and Nathan White

TABLE OF CONTENTS

Introduction and contributions

As well as discord contact information.

- Documentation and testing. Helped with code and logic for the leader bot. Mostly focused on Lab 6 material and understanding the svm model.



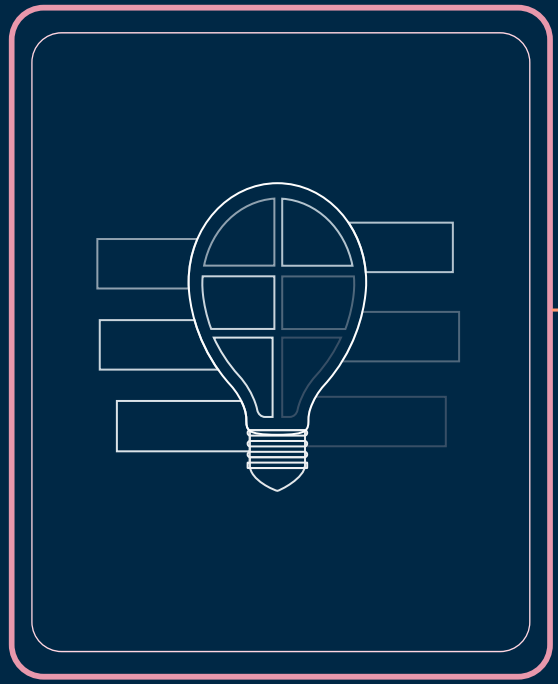
- Helped with the implementation and algorithm for the follow bot. Responsible for collecting results and testing the various methods.

- Drafted the original code for obstacle avoid and follow and wrote the increase and decrease speed methods in assembly. Wrote the ICD and ADD for the final report.

- Implemented the measure distance method and created a multithreading process to handle and receive input at the same time from the user. Responsible for design and structure of all methods for our final code.

OUR GOAL

Aside from the given leader and follower bot attributes we really strived to create a program that would allow the user to turn on obstacle avoid, or follow, and allow the bot to roam until the user wished to turn it off. This was achieved by Nathan creating threads to handle and receive input at the same time.



Interface Control Document and Algorithm Description Document

Interface Control Document (ICD)
for Palomar CSCI 212 FINAL
using osoyoo robot

Version 1.2

Adam Amer
Timothy Liu
Cody McKinney
Nathan White

GROUP #3
05.29.2021

Algorithm Description Document
(ADD)
for Palomar CSCI 212 FINAL
using osoyoo robot

Version 1.2

Adam Amer
Timothy Liu
Cody McKinney
Nathan White

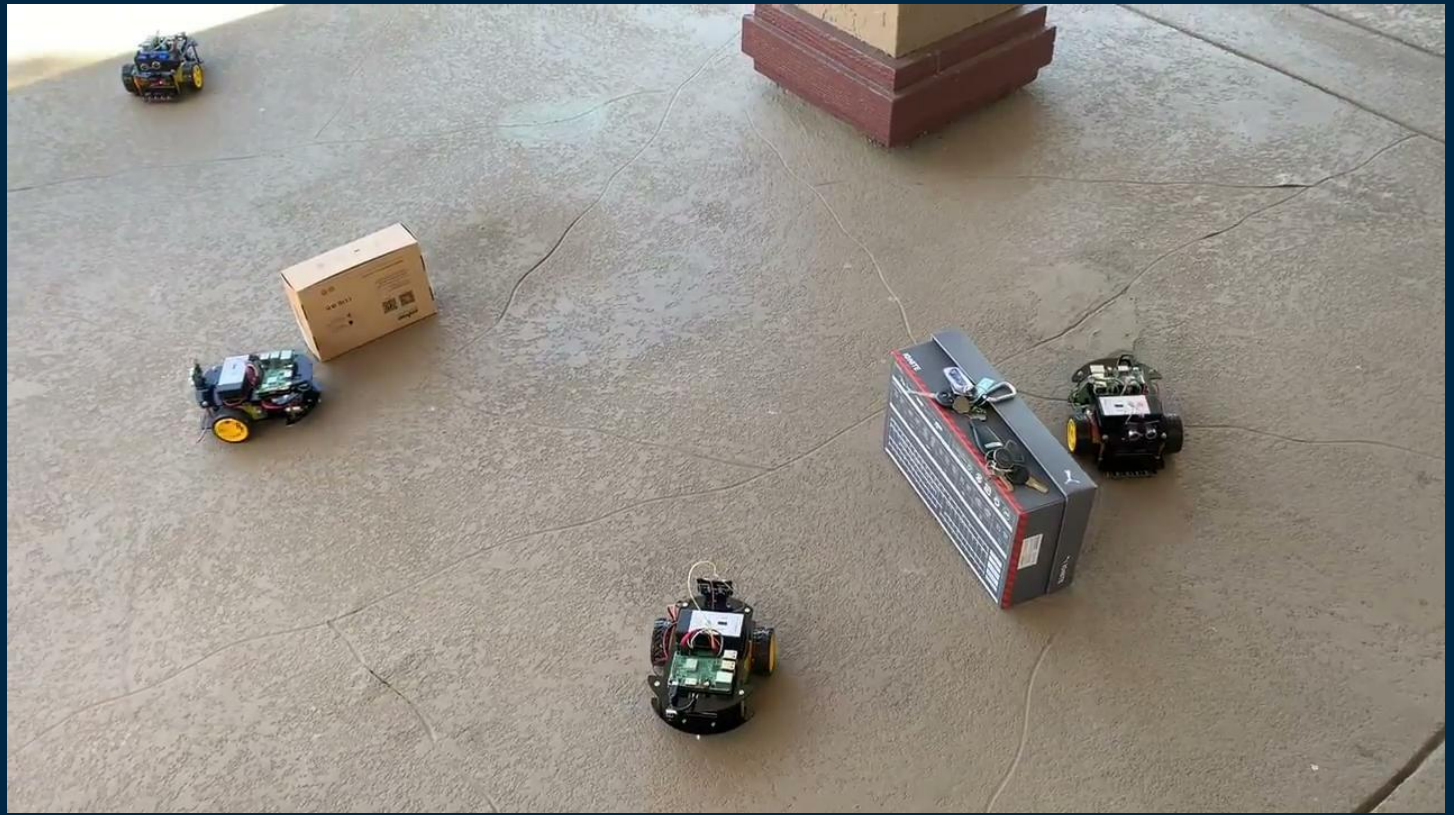
GROUP #3
05.29.2021

Results and Data

4 videos

1. All 4 Bots Avoiding an Obstacle
2. Bots Driving Different Speeds Showcasing Acceleration / Deceleration
3. Obstacle Avoid and Follower
4. Follower Bot behind Controlled Bot

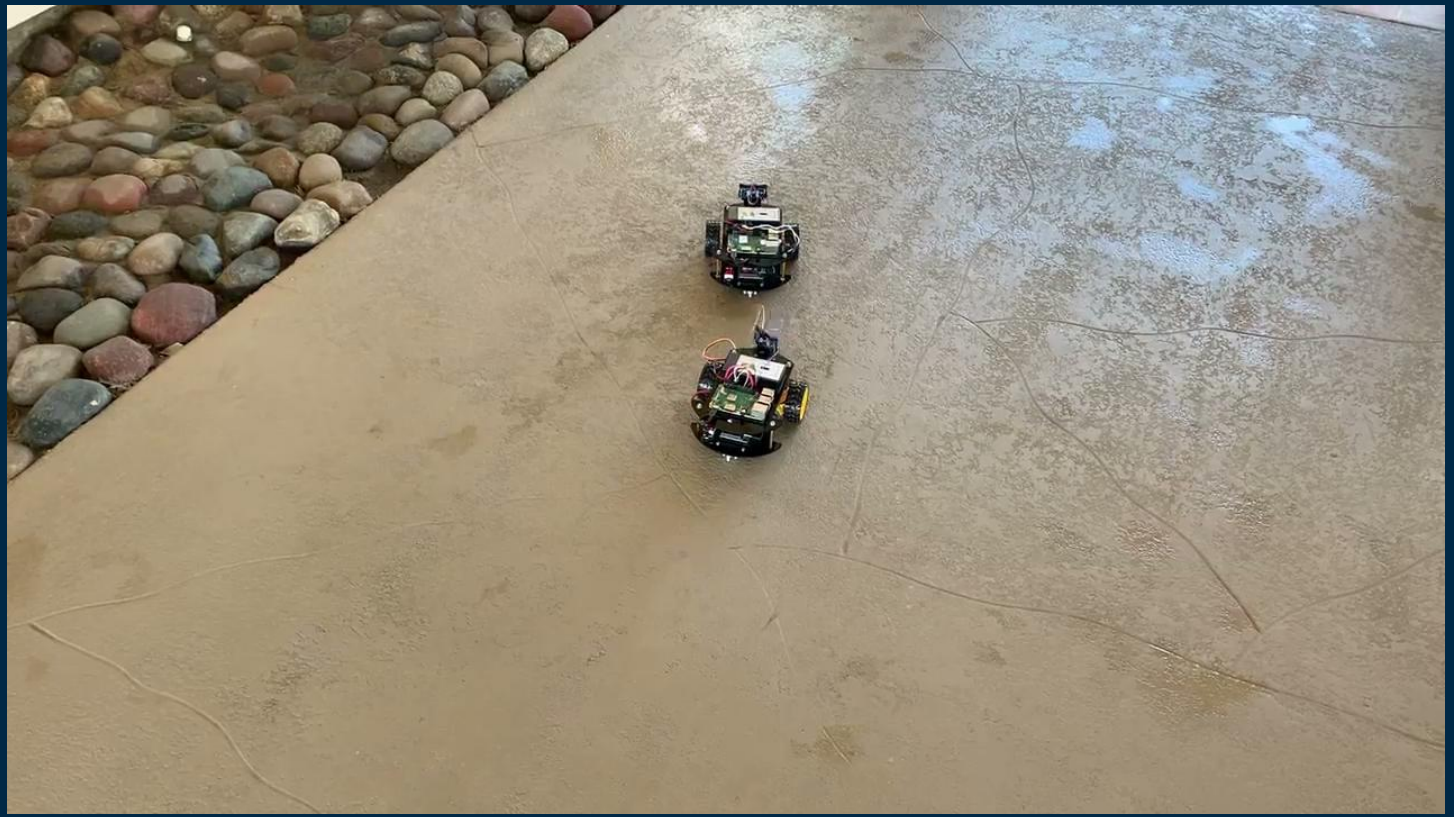
All 4 of our Robots Avoiding



Acceleration / Deceleration

Leader Avoid and Follower

Following a Controlled Robot



Source Code


```
float distance()
{
  struct timeval t0;
  struct timeval t1;
  float elapsed;
  float dis;
  digitalWrite(TRIGGER, 1);
  delay(0.00001);
  digitalWrite(TRIGGER, 0);
  gettimeofday(&t0, 0);
  while(digitalRead(ECHO) == 0)
  {
    gettimeofday(&t0, 0);
  }
  while(digitalRead(ECHO) == 1)
  {
    gettimeofday(&t1, 0);
  }
  elapsed = timedifference_msec(t0, t1);
  //printf("elapsed:         elapsed);
  dis   ((elapsed   34300) / 2) / 1000; // in centimeters
  printf("distance: %f\n"  dis);
  return dis;
}

/* check left sonic sensor value */
void check_left(void)
{
  printf("Left ");
  pca9685PWMWrite(fd, SERVO_PIN, 0, SERVO_LEFT);
  leftDistance   distance();
  if (leftDistance < OBSTACLE) { sts[0] = '1'; } else { sts[0]   '0'; }
  delay(long_delay);
}

/* check center sonic sensor */
void check_center(void)
{
  printf("Center ");
  pca9685PWMWrite(fd, SERVO_PIN, 0, SERVO_CENTER);
  centerDistance   distance();
  if (centerDistance < OBSTACLE) { sts[1] = '1'; } else { sts[1] = '0'; }
  delay(long_delay);
}
```

THANK YOU PROFESSOR & CLASS.