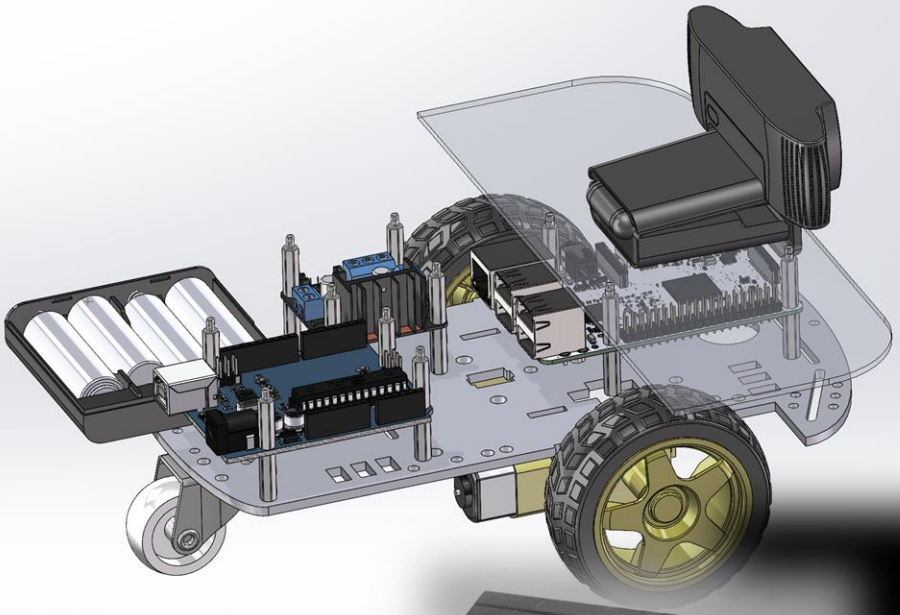
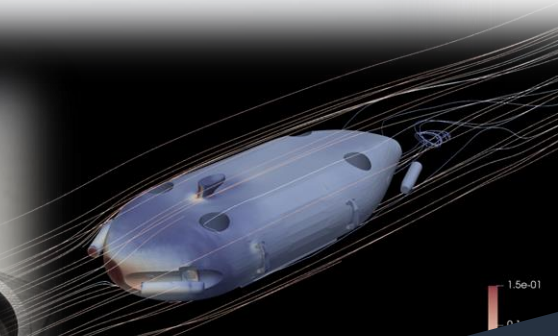
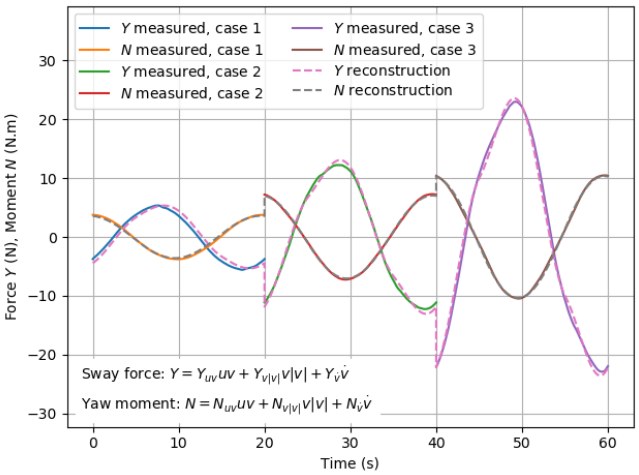
James “Wyatt” Cross
Engineering Portfolio

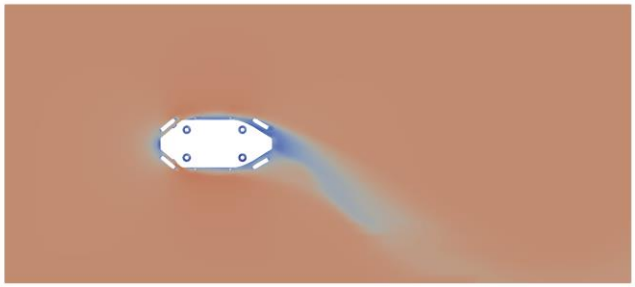


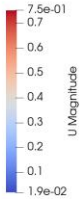


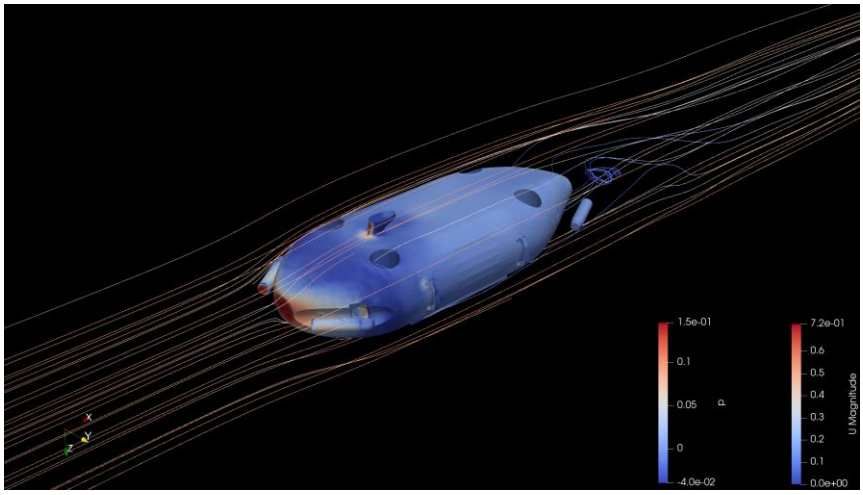
Contents

Internship – Hydrodynamic Coefficient Determination (Summer 2023)





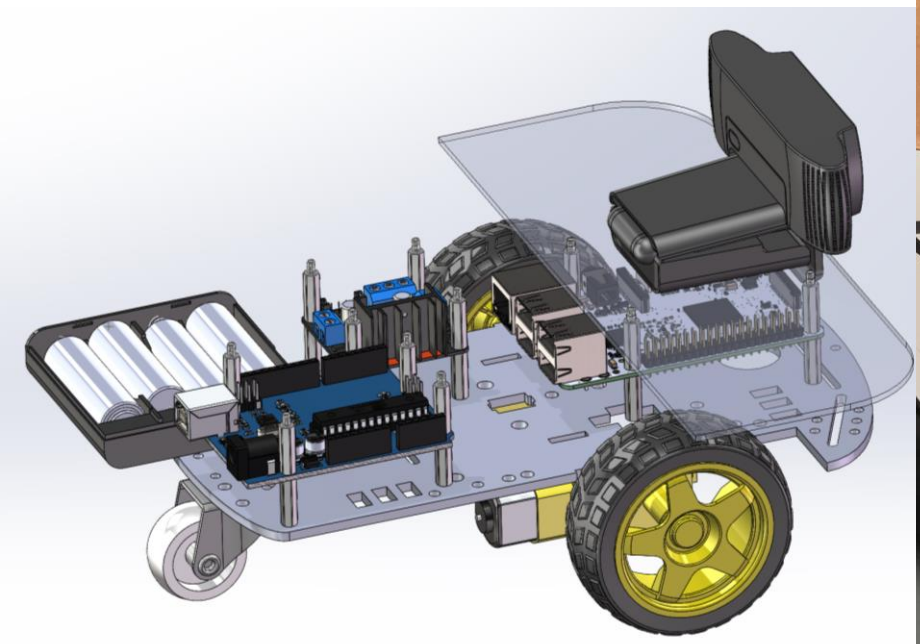


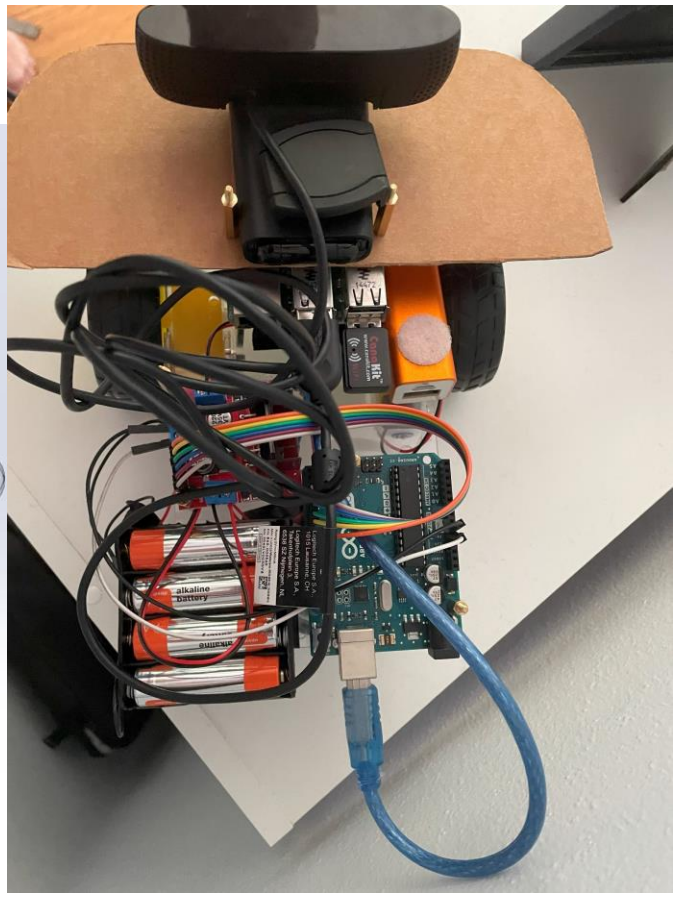


I worked primarily on developing a Computational Fluid Dynamics (CFD) model for an Uncrewed Underwater Vehicle (UUV). I worked on a team to estimate hydrodynamic coefficients that can be implemented into a model-based controller for a more robust vehicle control system.

Through mentorship and self study, I learned how to use OpenFOAM for CFD meshing and solving in 10 weeks. Meshing presented challenges as the included tool was highly automated but struggled with sharp corners in the vehicle geometry. I resolved this issue with additional meshing iterations and trying numerous meshing parameters to fix areas that would imapct the results.

Homemade TurtleBot (Summer 2023)



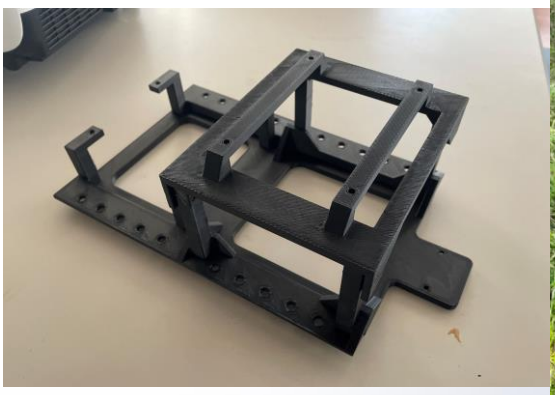


Background: I was interested in understanding how wheeled ground robots are built. I had previously worked with ground robots at a high level, receiving sensor data and sending velocity/heading commands.

Design: I started with the minimum to move; A L298N motor driver, 2 brushed motors and a battery. As I worked on my design, I added a microcontroller for motor control, and a computer for streaming images and commanding heading or velocity wirelessly.

Future Work: To add a GPS module to log position and work on filtering GPS signals along with an IMU to gain more control over velocity and heading as the current design can only regulate voltage.

Robot Backpacks (Fall 2023 – Present)



Background: For a research lab, additional computers and sensors needed to be onboard quadruped robots to support experiments. A compact “backpack” to mount the components was required.

Design: The final design combines sensor visibility requirements with port access. Space constraints meant components had to be stacked vertically to stay clear of the legs.

My design is the result of iterating over several months as additional sensors had to be added. It was very rewarding to assemble and watch the robot successfully complete waypoint missions with the new hardware onboard.

Robot Emergency Stop (Spring 2023)

estop_arduino_talker | Arduino 1.8.19

File Edit Sketch Tools Help

estop_arduino_talker

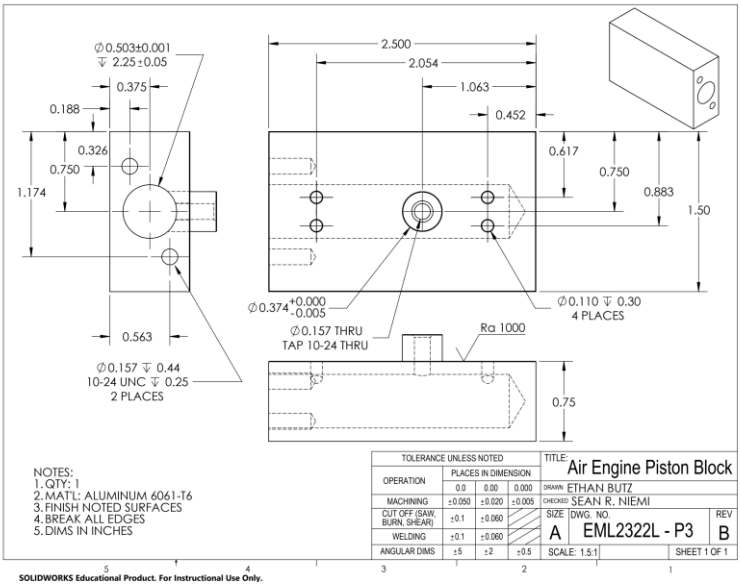
```
const int button_pin = 7;       //default 7 //changed to 4 - D2 on ESP8266
const int led_pin = 13; //default 13  //16 is nodemcu board, 2 is esp32 module

bool last_reading;
long last_debounce_time=0;
long debounce_delay=50;
bool published = true;

//------------------------------------------------------------------------

void setup()
{
  Serial.begin(115200); // use the same baud-rate as the python side

  //initialize an LED output pin
  //and a input pin for our push button
  pinMode(led_pin, OUTPUT);
  pinMode(button_pin, INPUT);

  //Enable the pullup resistor on the button
  digitalWrite(button_pin, HIGH);

  //The button is a normally button
  last_reading = ! digitalRead(button_pin);

}

//------------------------------------------------------------------------

void loop()
{

  bool reading = ! digitalRead(button_pin);

  if (last_reading!= reading){
      last_debounce_time = millis();
      published = false;
  }

  //if the button value has not changed during the debounce delay
```

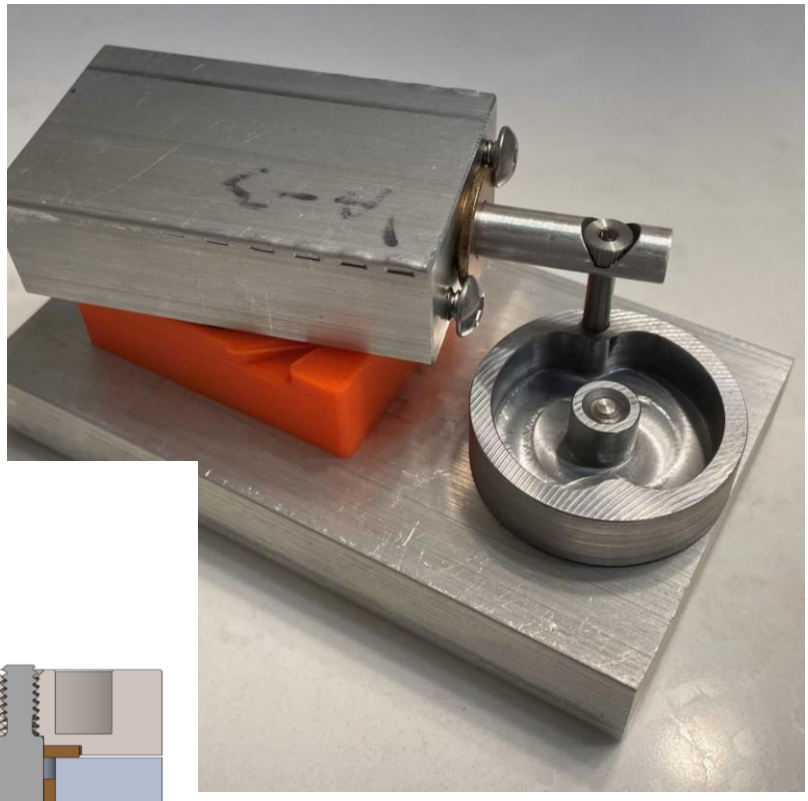
Goal: It can be difficult to find the correct terminal to kill when controlling a swarm of robots. Design a one-button system to stop many robots running different operating systems and middleware.

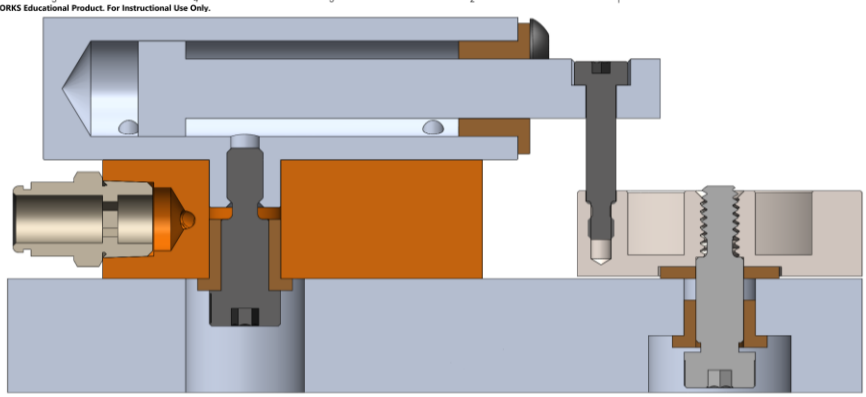
Description: I used a microcontroller to read button status and debounce the signal when activated. A connected ROS2 node publishes on each robot's estop topic to stop any connected robots.

Results: When tested, this system successfully stopped robot dogs, wheeled robots, and quadcopters. The Robot Emergency Stop is not intended to be a replacement for safety lock-out controls but used to stop everything when one robot is about to crash into something.

Design and Manufacturing Laboratory Air Engine (Fall 2023)





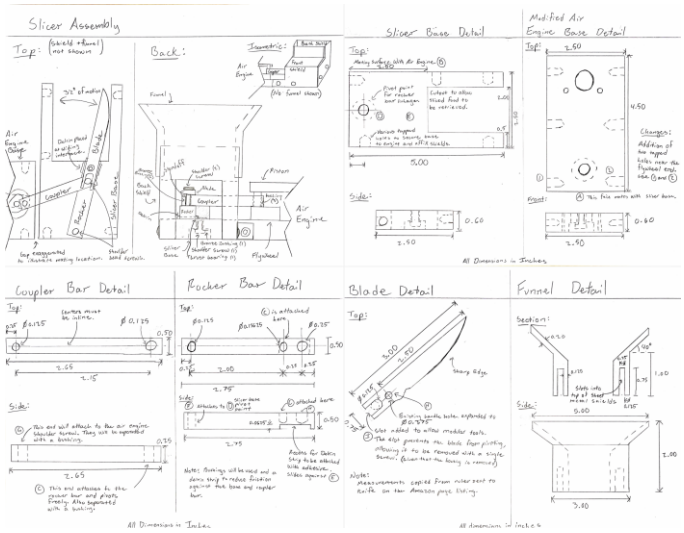


Objective: Manufacture 6 discrete components and assemble to create 4 functional air engines.

Process: I applied Geometric Dimensioning and Tolerancing (GD&T) knowledge to revise provided air engine drawings. Reducing noncritical tolerances allowed me to more easily achieve design requirements during manufacturing. I learned how to identify critical surfaces and got to see firsthand how high tolerances impacted manufacturing time.

I also learned how to preform manual mill and lathe work, and how to generate g-code in Autodesk Fusion for Haas VF-3 CNC mills.

Design and Manufacturing Laboratory Technical Reports (Fall 2023)



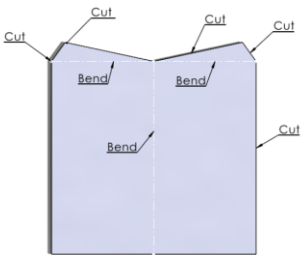
EML 2322L MANUFACTURING INSPECTION RECORD

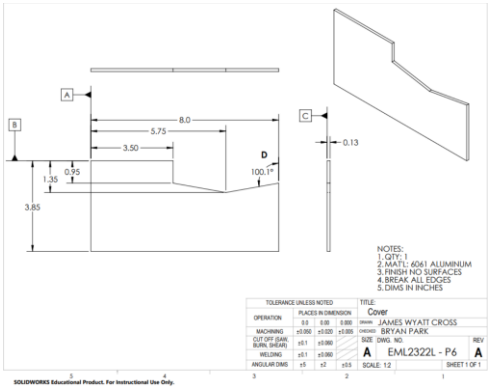
Part Number	Revision	Part Name	Team Number	Machinist:
P3	C	Piston Block	1A	Inspector:

No.	Ref. Location	Feature Type	Measurement Tool/Type	Requirement	Lower Bound	Upper Bound	Units	Results	Comments
1	A	Linear Dimension	Calipers	0.617	0.612	0.622	Inches		
2	A	Linear Dimension	Calipers	0.750	0.745	0.755	Inches		
3	A	Linear Dimension	Calipers	0.883	0.878	0.888	Inches		
4	A	Linear Dimension	Calipers	1.5	1.45	1.55	Inches		
5	B	Linear Dimension	Calipers	0.452	0.447	0.457	Inches		
6	B	Linear Dimension	Calipers	1.063	1.058	1.068	Inches		
7	B	Linear Dimension	Calipers	2.054	2.049	2.059	Inches		
8	B	Linear Dimension	Calipers	2.5	2.45	2.55	Inches		
9	E	Linear Dimension	Calipers	0.188	0.183	0.193	Inches		
10	E	Linear Dimension	Calipers	0.375	0.370	0.380	Inches		
11	E	Linear Dimension	Calipers	0.563	0.558	0.568	Inches		
12	C	Linear Dimension	Calipers	0.326	0.321	0.331	Inches		
13	C	Linear Dimension	Calipers	0.750	0.745	0.755	Inches		
14	C	Linear Dimension	Calipers	1.174	1.169	1.179	Inches		
15	D	Diameter	Calipers	0.110	0.105	0.115	Inches		
16	D	Thread	Thread Gauge or Fastener	10-24 UNC	-----	-----	Inches		
17	D	Diameter	Calipers	0.374[-0.005/+0.000]	0.369	0.374	Inches		
18	B	Diameter	Calipers	0.157	0.152	0.162	Inches		
19	B	Diameter	Calipers	0.503	0.502	0.504	Inches		
20	B	Thread	Thread Gauge or Fastener	10-24 UNC	-----	-----	Inches		
21	Notes	Notes	Visual	Finish Noted Surface	-----	-----	-----		
22	Notes	Notes	Visual	Break all edges	-----	-----	-----		

		RESULT:	PASS
Inspector Signature:		FAIL	☐ Remanufacture
Group Member 1:	Bryan Park		☐ Rework
Group Member 2:	Luna Garcia		☐ Engineering Change
Group Member 3	James Wyatt Cross		
Group Member 4	Lauren Barroso	TA Signature	

Angled Mouth

Manufacturing Processes	Process Setup
1.Find a .125" thick aluminum sheet 2. Overlay a 1:1 scale drawing of the angled mouth onto the sheet. Draw solid lines for cuts and dotted for bends. 2. Cut the shape using the manual shear. 4. Debur edges using deburring tool 5. Fold along the horizontal dotted line using the hand brake. 6. Fold the vertical line using the finger brake.	

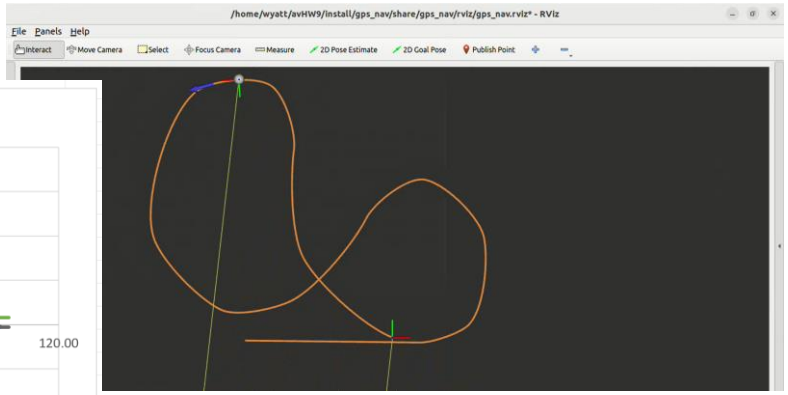


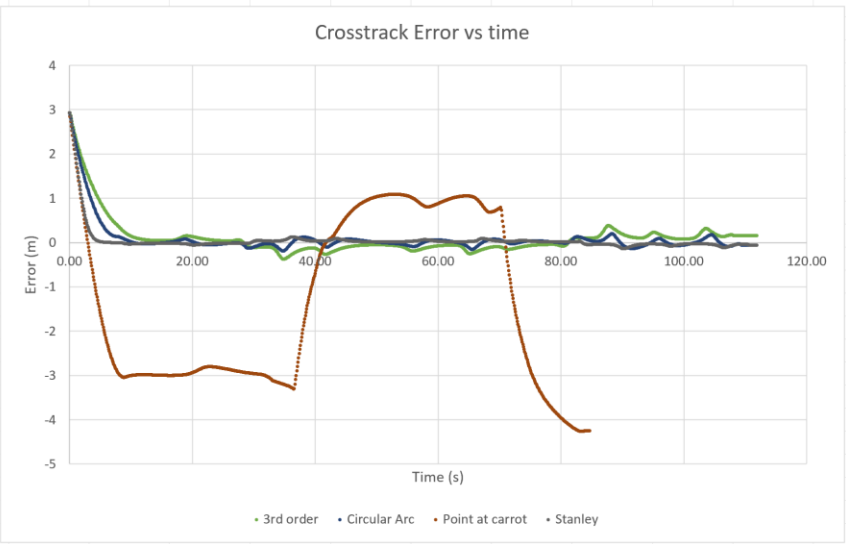
Objective: Design a novel mechanism that does not use gears or belts to transfer power from the Air Engine flywheel to a cutting blade.

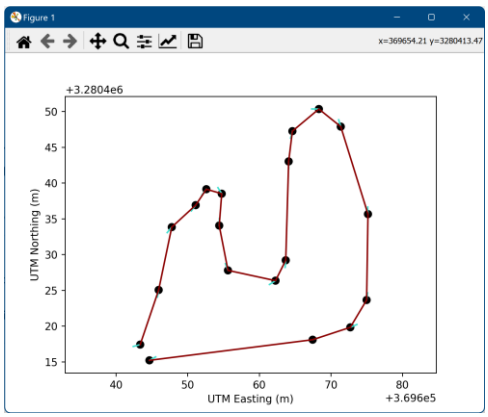
Process: I designed a reciprocating radial slicer that relied on a two-bar linkage and created dimensioned sketches for each functional part. Manufacturing Inspection Records and toleranced Engineering Drawings were created for each part. Finally, detailed manufacturing instructions were created for each part.

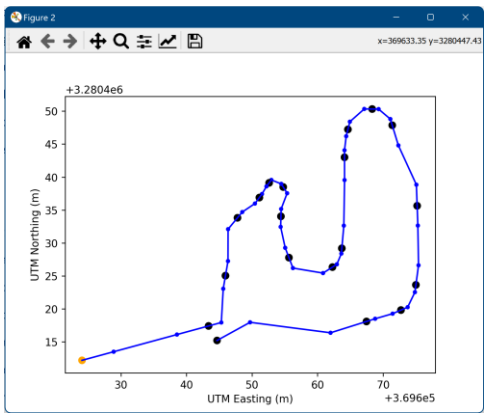
Results: I learned how engineering processes are documented and conveyed to machinists, quality control, and assembly. Cost was also a factor, and I learned to keep raw stock sizes in mind when designing.

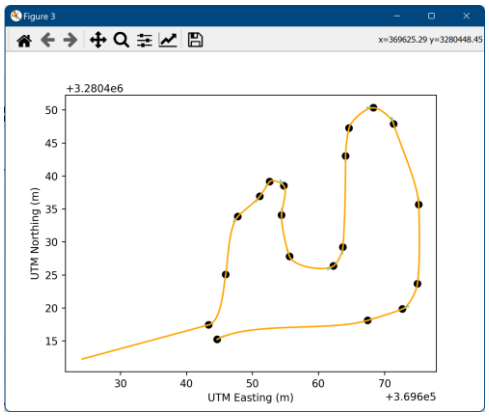
Autonomous Vehicles Class (Fall 2023)











Background: This was a technical elective I took that focused on learning how autonomous vehicles navigate in the real world. Shoebox-sized wheeled robots were used. They were equipped with GPS and LIDAR sensors.

One of the major assignments was getting the robot to travel along a predefined path. The top two images are results of testing my implementation of various waypoint following algorithms along a simulated path. The lower 3 images show route planning for use in a Stanley controller. I selected UTM points from Google Earth and learned how to turn a list of waypoints into a smooth path for the robot to follow.