ENGINEERING PORTFOLIO

Eric Weissman

eweissm1@asu.edu
480.993.5118

Contents

Introduction

Enclosed is my engineering portfolio, a collection of academic, professional, and personal projects that encapsulate my expertise and passion for engineering. Through a blend of academic knowledge, professional experience, and personal initiatives, this portfolio showcases my ability to innovate, problem-solve, and deliver tangible results in the engineering realm. I invite you to explore the diverse range of projects and solutions, each representing a unique facet of my skills and dedication to the field. For more details about the project or the work I have done, please reach out to me at 480.993.5118.

Thank you for your consideration,

Eric Weissman

M1X: Induction-Based Additive Manufacturing

During my time at Rosotics inc., a startup looking to develop induction-based additive manufacturing for use in aerospace superstructures, I was tasked to work on the Mantis project, later renamed M1X. The goals of the project were (1) to develop an induction based additive manufacturing (AM) process for high volume and safe manufacturing of aluminum components, (2) to develop a machine which could carry this print head, and (3) to convert these developments into a marketable product. Below you will find that my team and I were able to successfully meet these first two goals, while the work towards the third goal is currently in progress.

Print Head Technology

By far the most challenging aspect of the project was converting the sparse academic information that existed for induction-based metal AM into a marketable technology ready for aerospace applications. The work required for this endeavor is far too extensive to fully detail in this portfolio (it was a multi-year endeavor), however as a quick summary this process included:

- The determination of aluminum alloys compatible with our process
- The optimization of our induction parameters (current, frequency, and coil geometry) using surface optimization in COMSOL Multiphysics
- A half factorial exploration of process parameters (operating temperature, feed stock diameter, mass flux, substrate material, nozzle material, shielding gas, and auxiliary heat sources)
- Continual refinement of the experimental setup and of process reliability

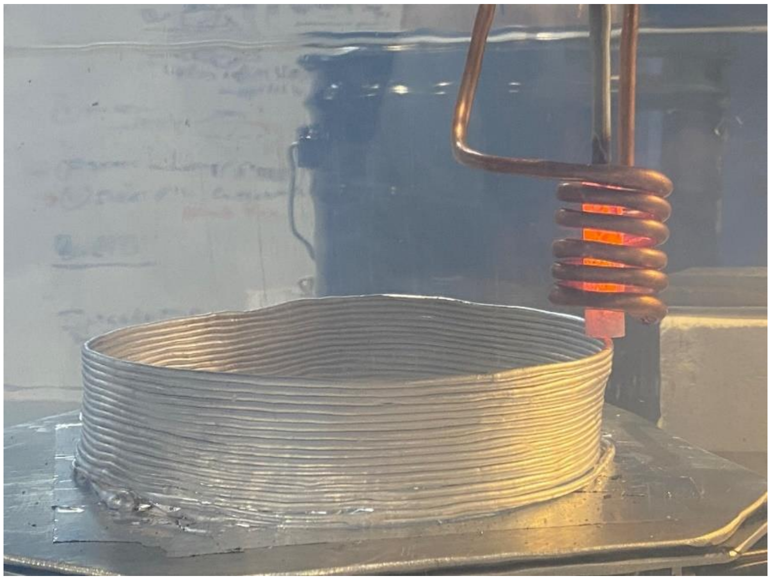

Figure 1: Primitive demonstration of printing technique.

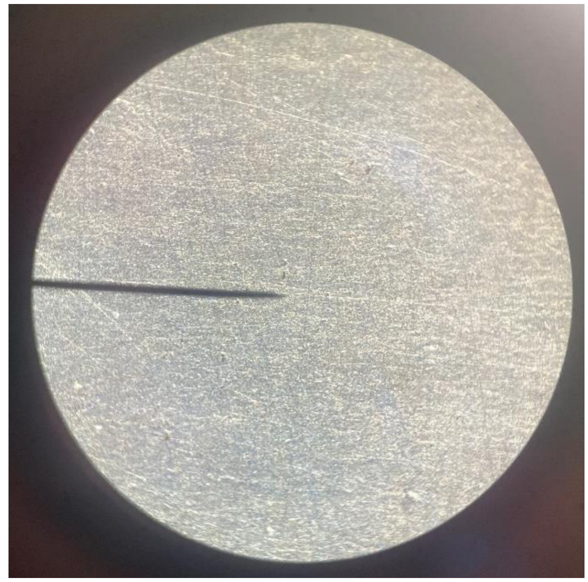

Figure 2: Etched sample of printed material under low magnification. Note lack of voids.

Figure 3: Early successful demonstration of our printing process.

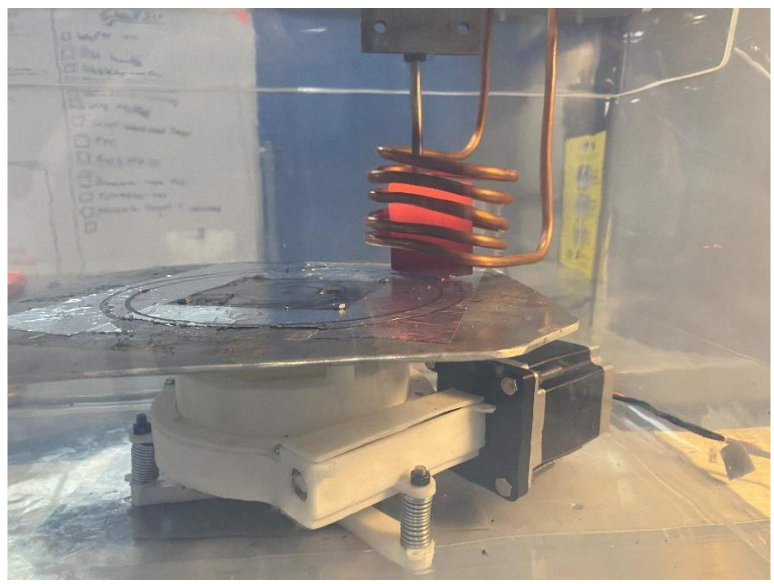

Figure 4: Testing the use of a graphite nozzle.

The pictures above demonstrate some of the early successes of this development, showing an early variant of the print head successfully depositing aluminum. Unfortunately, I am not at liberty to show any later successes nor discuss the mechanics of the above systems in detail to protect Rosotics.

2 DOF Cylindrical Printer

While the print head development was unequivocally a team effort, the development of the structure and controls for our full-scale prototype system fell on my shoulders as I was Rosotics' structural and controls engineer. Thus, I was responsible for modeling the machine in SolidWorks, using FEA and first principles to validate the designs, and coordinating with machine shops to order components. I was then responsible for implementing and coding the control systems (a Parker ACR controller and drivers using a Nabtesco gearbox). The result was a machine which could locate our 40 lb. print head with 0.1 mm repeatability over a 12 ft diameter, 6 ft high cylindrical table. Additionally, the table could support an 800 lb. distributed load without the code needing to accommodate for the deflection of the print bed.

Figure 5: Print head integrated onto the printer, actively printing a second layer of aluminum.

Figure 6: M1X "mule" printer, used to refine printing technique and demonstrate scalability of our technology.

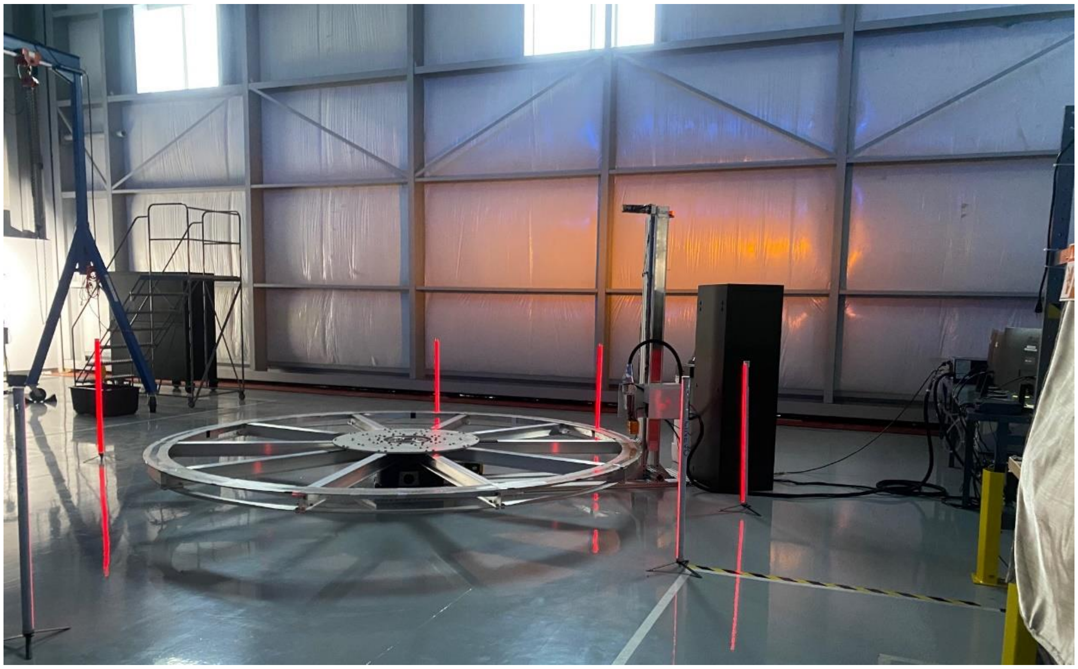

Figure 7: M1X "mule" one month after figure 6.

Personal Projects

Harmonic Drive

Shown below is a 3D printed harmonic gearbox capable of a 25:1 reduction for a Nema-17 stepper motor. The image on the right is a cross section showing the elliptical spline (orange), the flex-spline (pink), and the circular spline which doubles as the casing (green).

This project was deceptively challenging as the flex-spline was printed using a relatively cheap FDM printer. Thus, the low tolerance and interference required to make a reliable harmonic drive necessitated a substantial amount of testing and refinement.

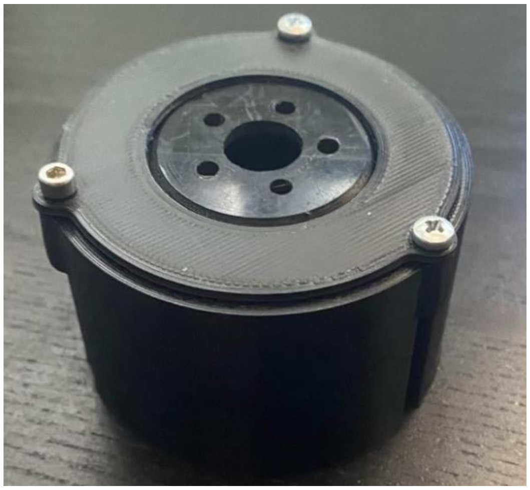

Figure 8: 3D Printed harmonic drive.

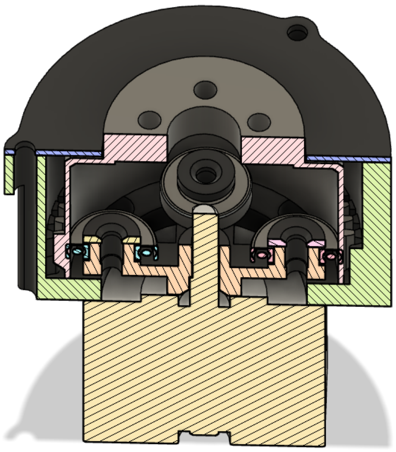

Figure 9: Cross section of harmonic drive.

Slew Drive

This is a 3D printed slew drive with a 132:1 gear reduction meant to be driven by a Nema-17. As can be seen above under P*rint Head Technology*, the slew drive was co-opted at Rosotics into the test stand we used for the print head development. The model used at Rosotics has some additions to allow it to be leveled and to better survive the high temperature environment.

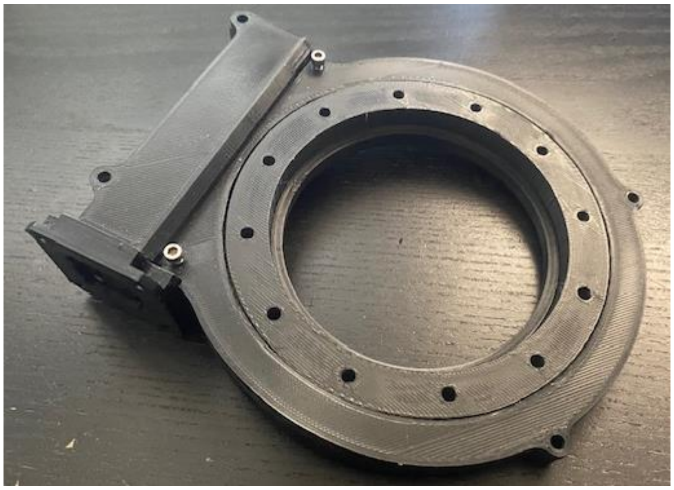

Figure 10: 3D printed slew drive.

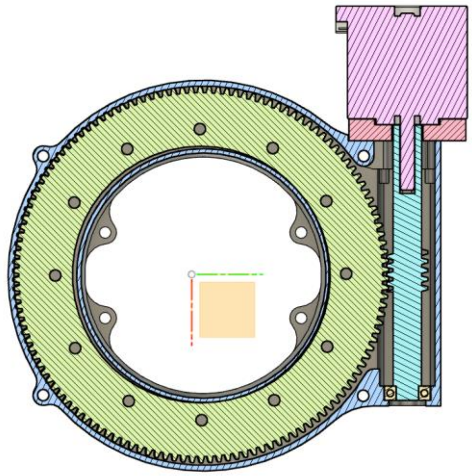

Figure 11: Cross section of slew drive.

Compact Planetary Gearbox

This is a single stage planetary gear box designed for use on a 6 DOF robotic arm. The gearbox offers an inline integrated 5:1 reduction that could be used on the wrist or forearm joints of a 6 DOF robot arm in a compact package. The design is optimized for FDM printing to minimize the supports needed.

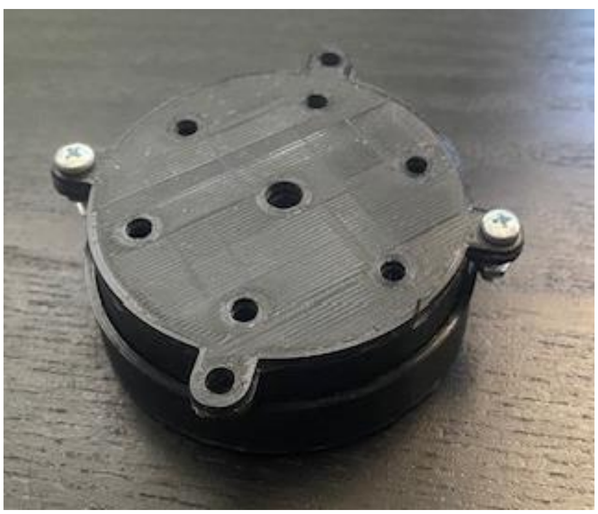

Figure 12: 3D printed compact planetary gearbox.

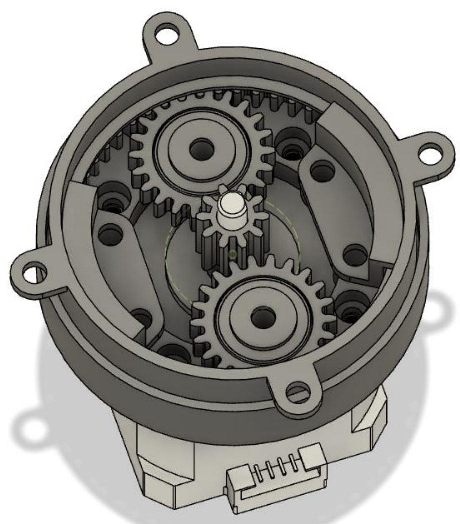

Figure 13: Rendering of gearbox with top mounting plate removed.

Stewart Platform

The Stewart platform is a 6 DOF kinematic manipulator capable of supporting tremendous loads over many design modes. Shown below is a Stewart platform (made from 3D printed and laser cut parts) controlled by an Arduino connected to a custom designed breakout PCB. The servo angles are computed on Python (see custom GUI below) and then communicated to the Arduino through serial communication.

See https://github.com/eweissm/StewartPlatfromKinematicsSolver for Python code for the inverse kinematics solver.

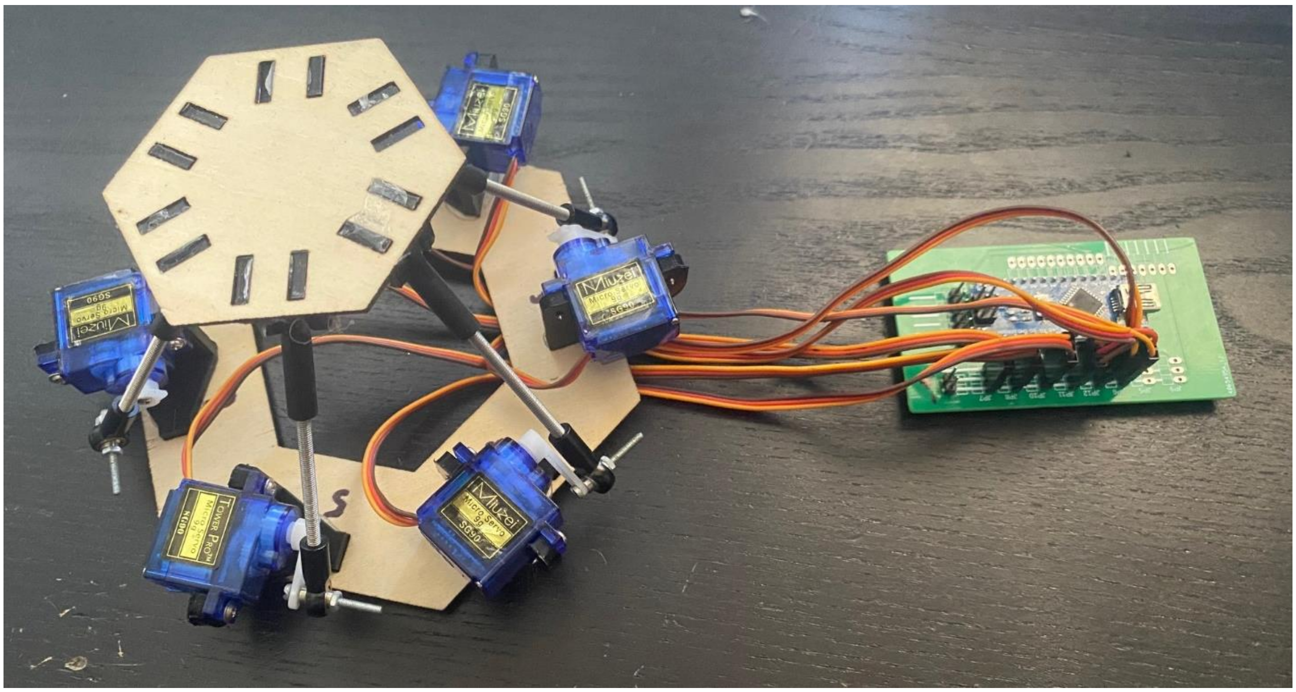

Figure 14: Stewart platform (left) with Arduino and custom PCB breakout board (right).

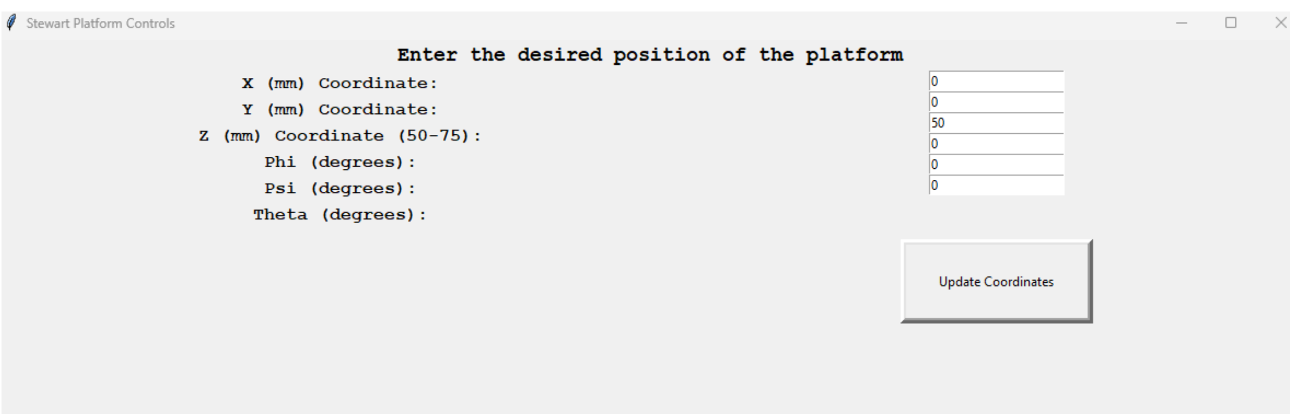

Figure 15: Screenshot of UI. Users can input the desired angles and coordinates (given from base origin).

Automatic Zen Garden

Shown below is an automatic Zen Garden, which uses a 2 DOF SCARA robot arm to position a ball bearing moving through the sand. This project consisted of numerous elements, including:

- The design of a continuous rotation SCARA arm and its integration into the wooden housing
    - This involved a vibration analysis to prevent the stepper motor from exciting an unlucky natural frequency of the box, causing an obnoxious noise.
- A hall effect homing sensor
- Wiring the microcontrollers and various servos or stepper motors
- Coding for the Arduino
    - This involved a homing sequence which would occur on system power up.
    - Additionally, this involved two-way serial communication with the controlling computer.
        - The Arduino would receive the desired joint angles from the Python, estimate the time for the move, send this estimation to the python, complete the joint moves (both joints arriving at the same time), and finally communicate to the python that the moves were complete, and the next joint angles could be sent.
- Coding the controller in Python
    - Used a root finding algorithm to find the optimal path and joint angles for the robotic arm.
    - A custom GUI was developed, which would take user input and use serial communication to talk with an Arduino to control the joint angles. Shown below is the UI, which outlines the workspace of the robot (black dotted line) and the selected preset path the arm can follow when the "follow path" button is pressed (blue dotted line). The user can scroll through various presets by selecting the "Change Path" button. Finally, if the user inputs custom X and Y coordinates, once the "Update Coordinates" button is pressed, the arm will move to that location, and the arm joint angles and position will be shown.
    - The preset paths can be passed through a parametric equation or through a G-code file via a custom regex interpreter I wrote for basic G-code commands.

See https://github.com/eweissm/Autonomous-Zen-Garden for code (note the code will not run unless an Arduino is plugged into the computer or if all lines beginning with "ser" are commented out). Additionally, a video of the operation can be found on the GitHub page.

Figure 16: Picture of completed Automated Zen Garden

Figure 17: Pattern drawn in sand by bearing.

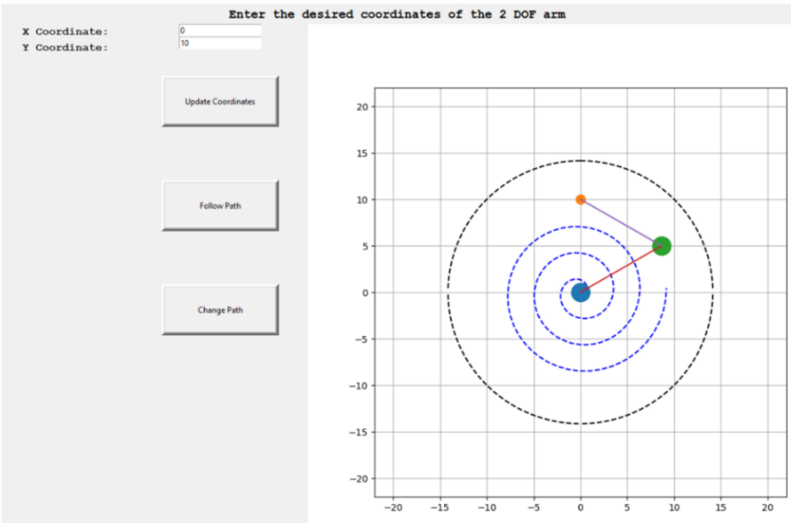

Figure 18: Screen Shot showing UI. Note the workspace (black), the selected path (blue) and the arm with its computed joint angles.

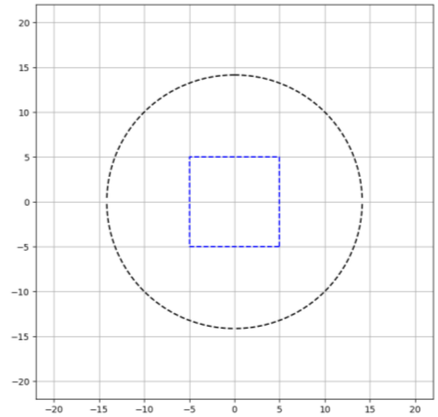

Figure 19: Screenshot showing an alternative pre-generated path (rectangle).

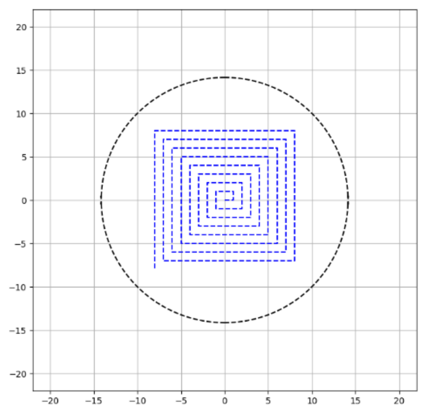

Figure 20:Screenshot showing an alternative G-code generated path

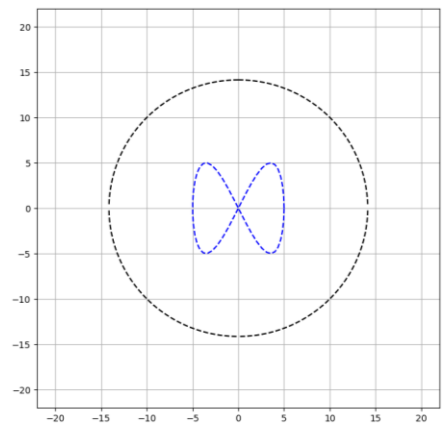

Figure 21:Screenshot showing an alternative pre-generated path (lemniscate).

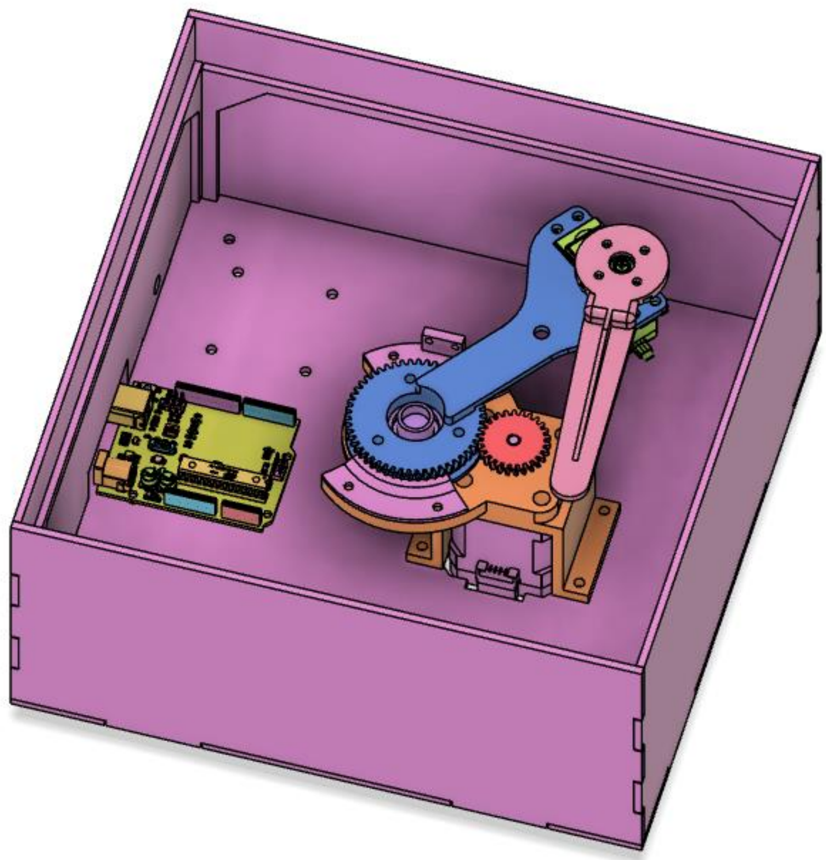

Figure 22: Partial CAD model showing how SCARA arm functions.

Machine Vision Maze Solver

This project involved writing a Python script which would analyze the live feed from a web cam to (1) find the start and end of a maze, (2) determine the walls of the maze, (3) determine where the corners of the maze were, (4) use the found corners of the maze to apply a parallelogram correction to rectangularize the image, and (5) solve the maze. This was achieved by detecting volumes in the image of certain colors, allowing for the corners (yellow), walls (pink), the start, and the finish (both green) to be detected in the image. Then it was a simple matter of applying the inverse parallelogram transformations and using an A-star pathfinding algorithm to find the shortest path through the maze. In the images below you will see (top left) the original image with the various elements detected, (top right) the rectangularized image, and (figure 24) the shortest path denoted by the blue line.

This project was inspired by the IEEE Micromouse competition which requires a robot to solve a maze as quickly as possible. Given enough development time, an algorithm like this could be written in which a robot could quickly evaluate the maze, use the rectangularization to convert the maze into a usable bitmap, and then quickly complete the maze.

See https://github.com/eweissm/Machine-Vision-Maze-Detection-and-path-finder for the Python script.

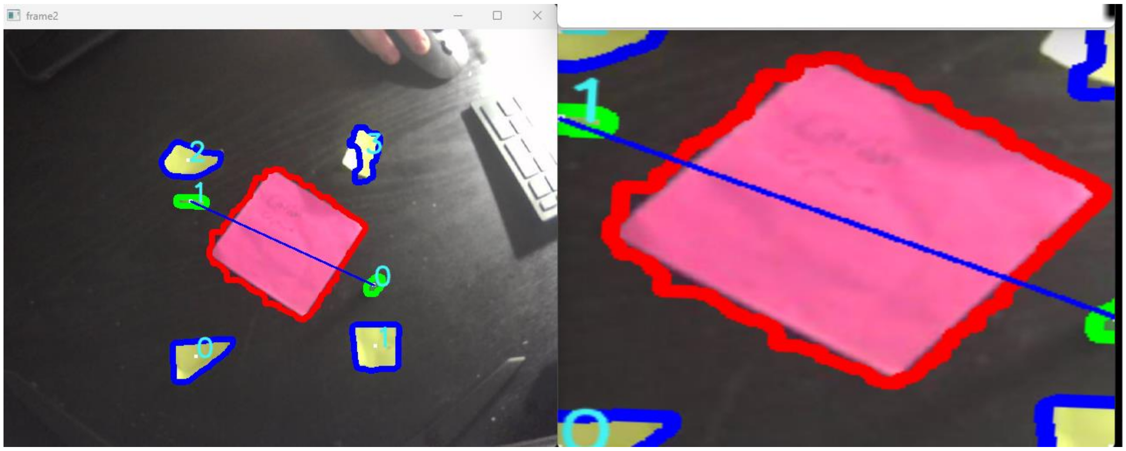

Figure 23: Screenshot showing the original image (left) and the rectangularized image (right). Note the outlined volumes denoting different elements.

Figure 24: Bitmap interpretation of the "maze". Note that the pink paper became an obstacle for the maze to find a path around.

Topology Optimized Model Aircraft Motor Mount

After a particularly hard landing the motor mount on the RC aircraft shown below was damaged. It was decided that a 3D printed replacement would be the best solution to replace the damaged balsa wood. Using the topology optimization package in Ansys, a new motor mount was developed with a factor of safety of 3 (this high F.S. was selected due to the unreliability of the FDM parts). The new mount was ultimately 12% lighter than the older mount and is still flying today.

Figure 25: RC Aircraft with new motor mount installed.

Academic Projects

Ramjet Optimization Tool

For this project a first order design tool was developed for the optimization of design parameters for a ram/scramjet engine using MATLAB. To achieve this a first order model of a ram/scramjet was coded. Then using (1) a full factorial design exploration, (2) an SQP optimization, and (3) an fsolve optimization, the design space could be explored, and an optimal multivariate solution or solutions could be easily determined given a desired mission profile.

For a full discussion of this project see https://github.com/eweissm/RamjetOptimizationTool.

Neural Network Optimized Rocket Controller

This project involved training a neural network to control a 3 DOF (2 translational and 1 rotational) rocket coming in for a landing. The network needed to land the rocket (have a speed equal to zero at the landing pad) before the fuel ran out. The model was trained by generating a rocket at a random coordinate with a random velocity at t=0s and weighing the performance of the controller to land the rocket. Two different algorithms were investigated (AdaMax and LBFGS), with the AdaMax converging slower but with fewer residuals. It is important to note that the rockets could be spawned in unrecoverable initial conditions, so the residuals never reached zero.

For a full discussion of this project see https://github.com/eweissm/Rocket-Neural-Network-Optimized-Control.