Computer Science And Engineering Portfolio

Tiernan Lindauer, Texas A&M University Brown Scholar

Contents

Preface

This portfolio is represented in reverse chronological order by their completion date, with the most recent entries appearing first. Unless otherwise stated, the projects described are my independent, unique work. To examine the codebases for these projects in depth, please visit https://t-lind.github.io/projects/. The first entry starts on the subsequent page.

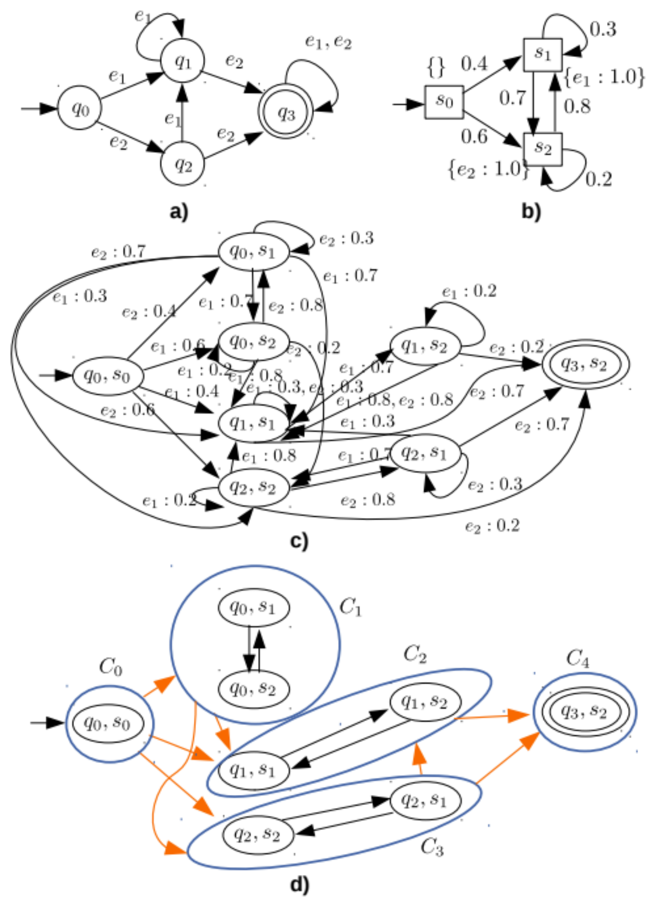
My personal favorites include **4D Projection**, **REPNET**, the **Differential Swerve Drive & Pathing Library**, and my **Kinect Terrain Tracker**.

Thank you very much for your time,

Tiernan Lindauer

EXPLORING ALTERNATIVE COST MECHANISMS FOR PROBABILISTIC PLANNING

H. Rahmani et al.'s paper at TAMU, "Planning to Chronicle: Optimal Policies for Narrative Observation of Unpredictable Events," introduced probabilistic planning to minimize expected steps in a POMDP framework, applicable to robot videography, such as recording a race. This research explored alternative cost metrics—like distance traveled—beyond mere step counts to enhance performance in various scenarios. The project refactored the original code to implement and evaluate new cost functions, allowing for soft constraints and cumulative cost minimization. This approach assessed when cost-based planning outperforms step minimization, expanding the applicability of probabilistic planning.

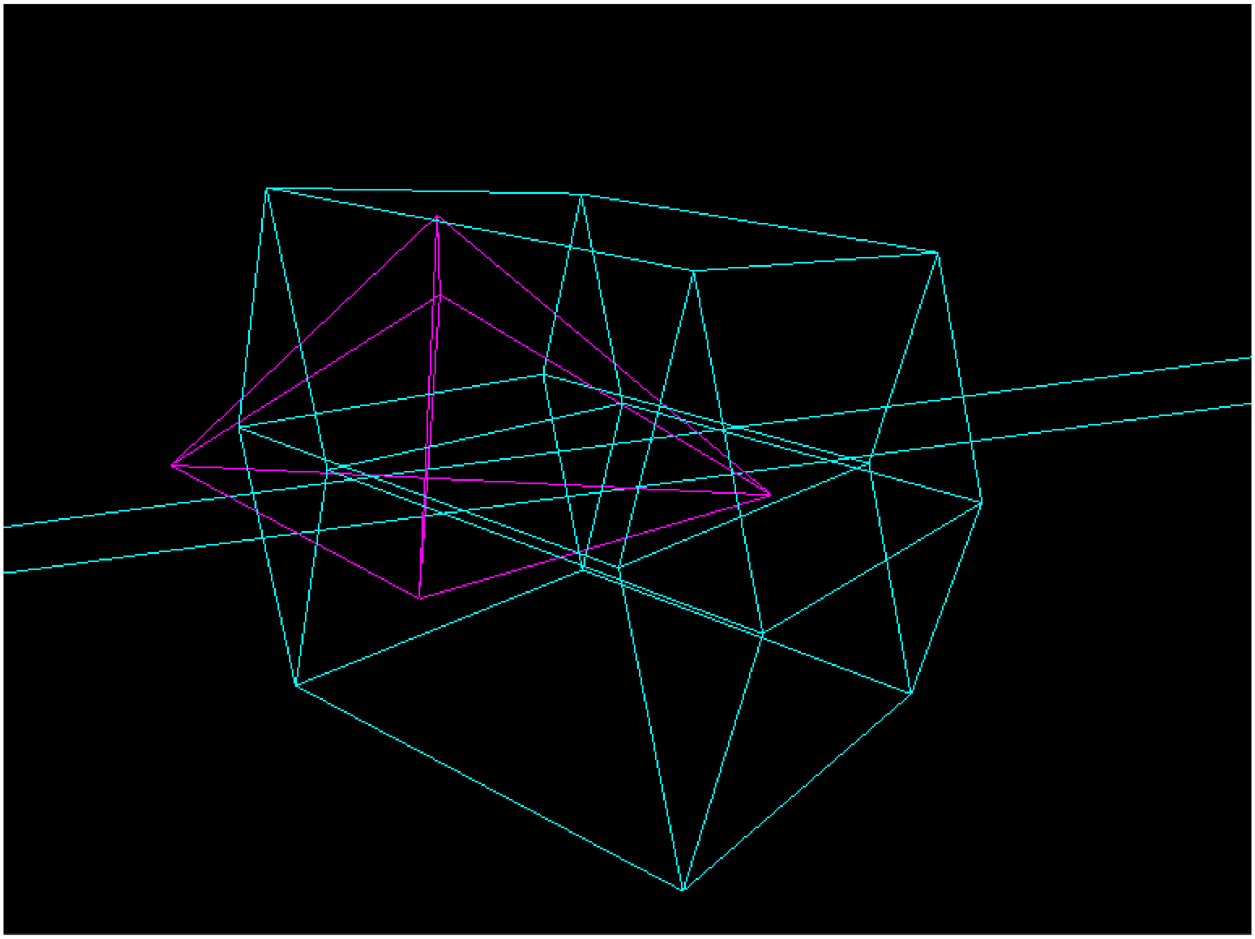


A LOOP-OMITTED ACYCLIC DFA. B) A SAMPLE EVENT MODEL. C) THE GOAL MDP OBTAINED FROM THE PRODUCT OF THE DFA AND THE EVENT MODEL IN PARTS (A) AND (B). D) IT SHOWS THE STRONGLY CONNECTED COMPONENTS OF GRAPH UNDERLYING THE MDP IN PART C. EACH BLUE CIRCLE IS A STRONGLY CONNECTED COMPONENT. THE SELF-LOOPS AND THE EDGES BETWEEN STATES OF DIFFERENT SCCS HAVE BEEN OMITTED TO REDUCE VISUAL

REFERENCES

1. Rahmani, H., Shell, D. A., & O'Kane, J. M. (2023). Planning to chronicle: Optimal policies for narrative observation of unpredictable events. *The International Journal of Robotics Research*, *42*(6), 412-432.

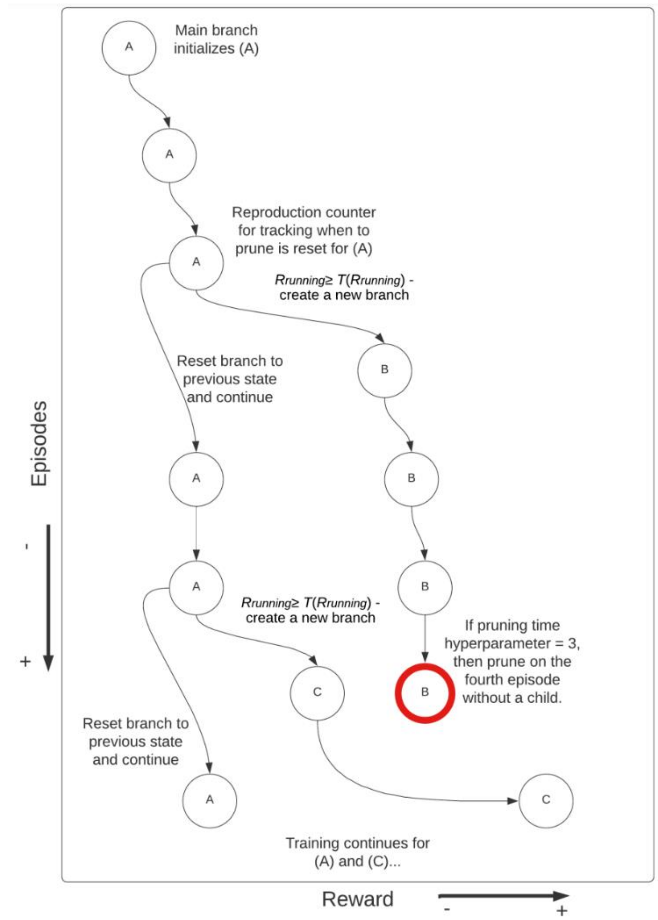
4D Projection: Rendering Non-Parallel 4D Mesh Slices in 3D



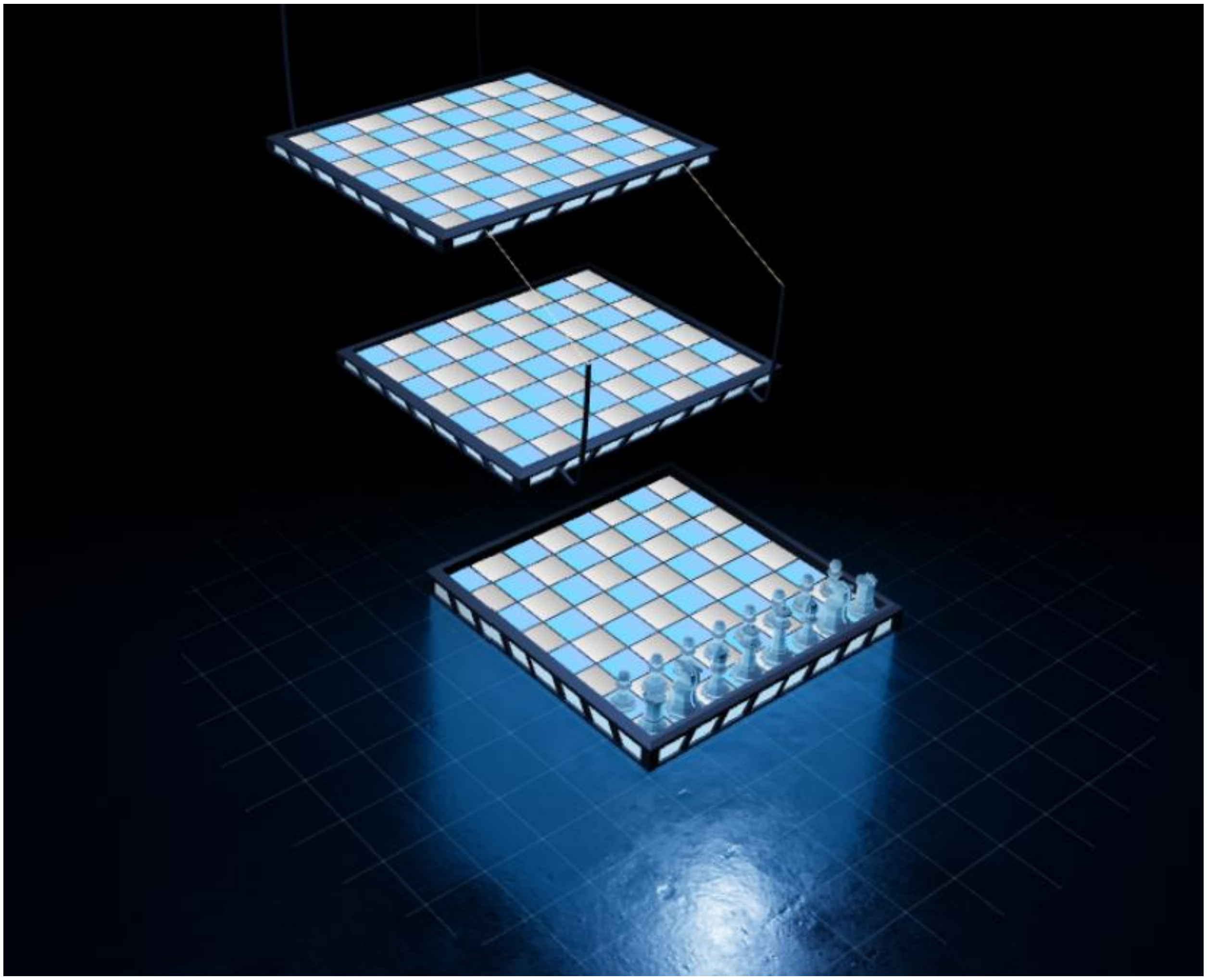
4D Projection enables a user to define 4D meshes and then renders the meshes in an interactive 3D program using Python. Using this tool, it is possible to not only analyze the 3D object using parallel hyperplanar slices (which current technologies implement) but skewed hyperplanar slices, enabling the new 3D projections that were previously not mainstream. This project used substantial linear algebra to calculate these projections.

REPRODUCTIVE TRAINING ARCHITECTURE FOR REINFORCEMENT LEARNING NEURAL NETWORKS (REPNET)

Since 2001, actor-critic algorithms have combined "the strong points of actor-only and critic-only methods" (Konda & Tsitsiklis, 1999) to reduce the time needed to train a reinforcement learning neural network. The REProductive NETwork architecture, REPNET, uses a custom data structure built on top of this idea to train networks in significantly less time. During training, the models in REPNET, called branches, can reproduce, rewarding models that achieve a high running reward change. Branches can also be pruned due to a lack of reproduction, penalizing stagnant branches. These branches are contained in the custom recursive data structure of REPNET. Through the use of automatic add-ons such as Adaptive Pruning Time Adjustment (APTA) and Auto-Generating Nonlinear Reproduction Threshold Functions, these hyperparameters can be tuned without human input. After training the neural network on OpenAI's "CartPole" environment, REPNET reduced the mean of the training episodes by 29.3% and the standard deviation of episodes by 72.62%. Overall, REPNET made reinforcement learning training significantly faster and less variable in comparison to its actor-critic counterpart. Decreasing the training times of reinforcement learning neural networks can accelerate progress in AI research, leading to faster development of technologies that can improve various aspects of the world, from healthcare to environmental sustainability. I currently have a patent pending with the USPTO.



3D CHESS



3D Chess is a true 3D chess game, extending the moves pieces can make into three dimensions. Other approaches like Star Trek's Tri-Dimensional Chess aren't truly 3D, and moreover, there are no good online options. I worked on this project with Kavin Gupta, implementing the backend and chess logic (and coming up with the idea) while he designed the frontend, or what you see above. Our code provides proper move hints and logic for playing a local game.

FIRST TECH CHALLENGE (FTC) STATE CHAMPION ROBOT

The FIRST Tech Challenge offers high school students a competitive robotics game each year, allowing teams to build their robots with considerable freedom and compete at various levels. My team, 9527 Rogue Resistance, set multiple Texas state records and achieved a top ten world ranking. We utilized a mecanum wheel drive system, a solid steel chassis for stability, and cascade strung linear rails within a compact 12”x12” footprint, enhancing our maneuverability.

We placed first in the challenging Central Texas Semi-Regional competition and became UIL 6A State Champions. As the software team lead, I worked with developers to ensure precise pathing during the 30-second autonomous phase using camera vision and localization techniques, and contributed to developing Kalman filters for our open-source library, Blacksmith: blacksmithftc.com.

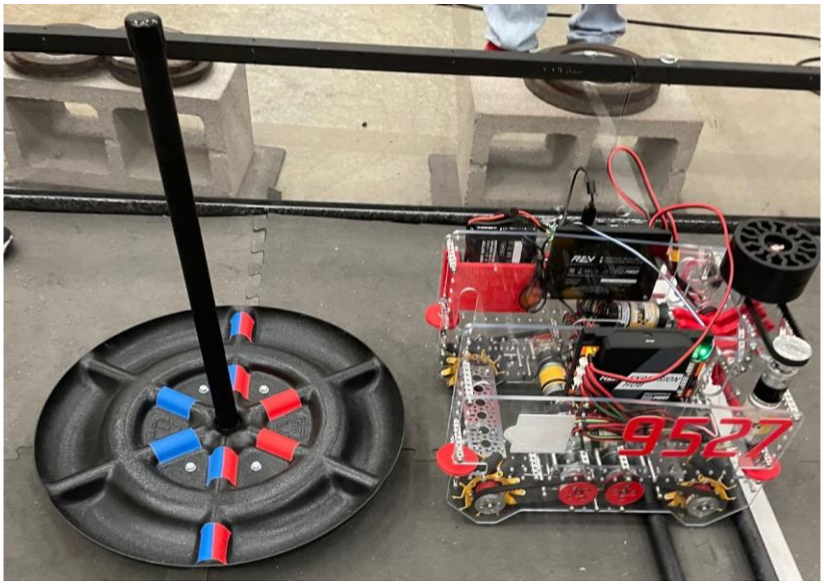
Our dedicated 15-member team faced challenges, including a critical failure during our league championship when an aluminum part on our lift shattered. Without a backup, we withdrew from that competition but advanced based on an award, ultimately fixing our issues and gaining recognition in the FTC community.

TURING MACHINE LANGUAGE

The Turing Machine Language uses a Java compiler in order to turn easy-to-understand code that controls a Turing machine into a system of pages and awareness through lambda expressions. It also allows quality of life functionality, such as descriptive exceptions, setting variables, defining macros, and allotting registers of numbers easier. To the right is an example of summing two 7-bit binary numbers. The full code can be found here. It also comes with a VSCode extension that performs code highlighting, and autocomplete for a better development experience. The installer (for Windows only) is found here.

I created this language because I wasn't satisfied with instructional materials for learning about Turing Machines – and, given that many people love learning hands-on, I decided to use my technical acumen to do something about it! I'm currently working with my previous high school to integrate it into the Computer Science 3 curriculum. My hope is that can help others learn more about this interesting thought experiment, and I'm excited to see what people do with it! Future development plans include the ability to import files, being able to partition a very large program out into logical chunks.

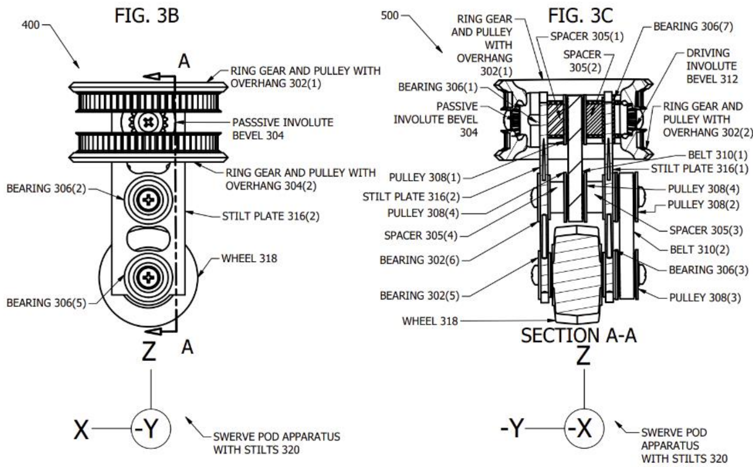
FIRST TECH CHALLENGE (FTC) STATE RUNNER-UP ROBOT



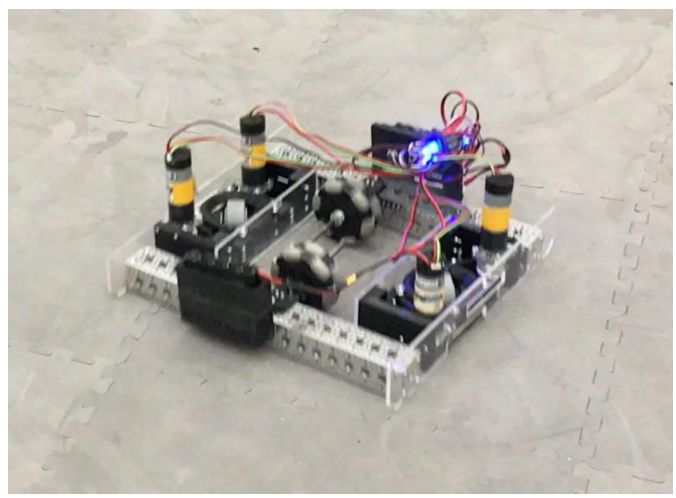
During the 2022 FTC season, my team first created a suspension mecanum drivetrain (seen on the right), used to navigate across the terrain on the field. We worked through many iterations to be able to create a smooth driving experience, but our large size was not an optimal strategy for that year. In four weeks before the UIL State 6A Competition, our team designed, manufactured, programmed, and tested the robot above on the left, which was much smaller and only placed 2nd as Runner-Up due to a technical fault in one of our alliance partners. It was an amazing test of dedication and skill, and bonded the team in a way like no other.

Over this season, I worked on the drivetrain design in Fusion 360 and manufacturing process. I also worked on using classical CV methods to identify certain game elements – cubes, wiffle balls, and ducks – and automate our intake and navigation processes.

DIFFERENTIAL SWERVE DRIVE & PATHING LIBRARY

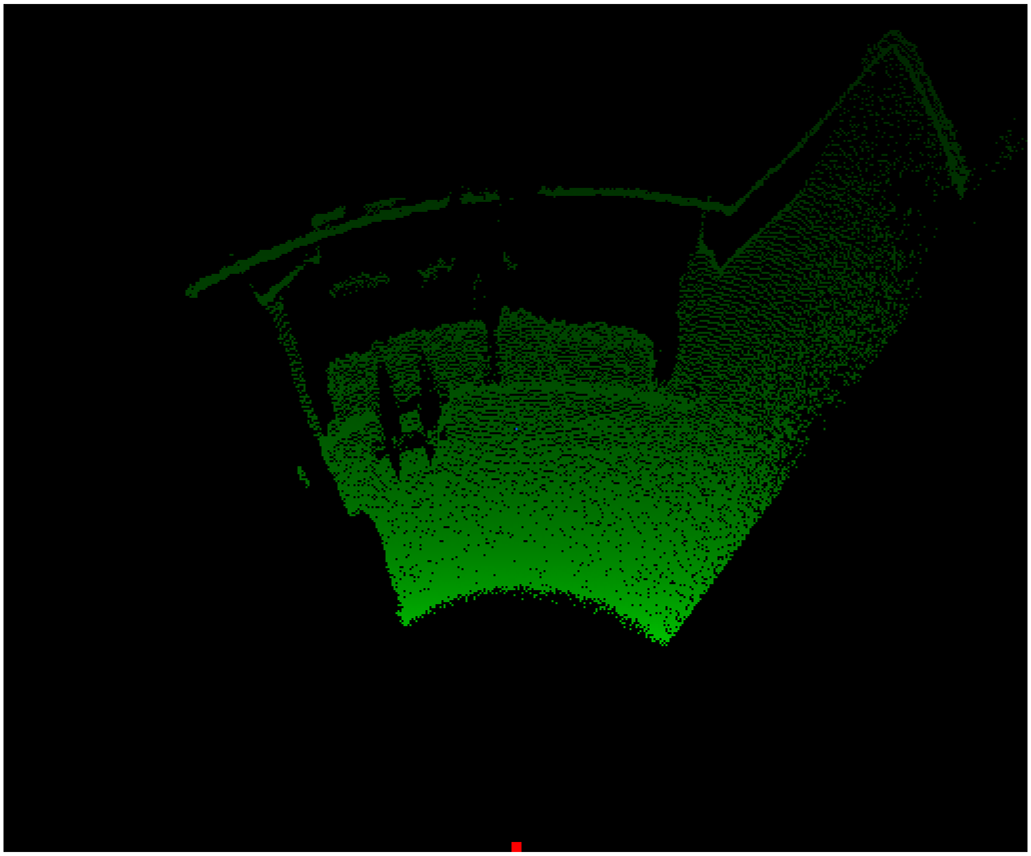


Coyote is a pathing and control library for two-wheel differential swerve drives. I used CAD to iteratively design the drivetrain in Fusion 360, and 3D printed / machined the different components of the swerve pods. Differential swerve drives are much more efficient given the constraints of FTC robotics, and provide for greater maneuverability.



Notably, I designed the pathing and teleoperated control library for following complex curves, like Bezier curves. Using three-wheel odometry and encoders in each drive pod, accumulated errors were addressed and reliability was maintained.

KINECT TERRAIN TRACKER



The Kinect Terrian Tracker is a fast and efficient floor detection system that uses raw depth data from an Xbox One Connect to project the detected floor into a bird's-eye view with dynamic calibration. The project was written in C++ and uses the Kinect SDK, which provides the raw depth data from the camera but nothing else. The Kinect Terrain Tracker uses this data with no additional dependencies to convert this raw data into positional information, and is able to detect the floor surface in a room, marking it from the camera's perspective or giving a birds-eye-view of the floor.