DESIGN AND CONTROL OF A CARDBOARD SELF-BALANCING CUBE

Thomas Godden - 2021

Summary

One of the "holy grails" of control engineering is the concept of a self balancing cube. More specifically, a cube that can keep itself upright by exerting torques through reaction wheels or another mechanism. For my Gizmo project, I built a cardboard cube that is able to self stabilize and even reject minor disturbances in 3dof using 3 independent reaction wheels. To fulfill the interaction requirement, I created a remote "tuning station" that gives live feedback about the cube's balance, and allows for on-the-fly adjustment of the gain constants within in the control algorithm.

Motivation

When approaching my project, I not only wanted to create something cool out of cardboard, but I wanted to make something that actually benefited from the material properties of cardboard. I considered objects that needed to be light and rigid, but would not be subjected to large loads. After some ideation, the concept of a self-balancing device came to mind. In general, you want the structure of these devices to be as light as possible, so you can have as much control over the center of mass location and devote the largest portion of it's mass to the actuators and power systems.

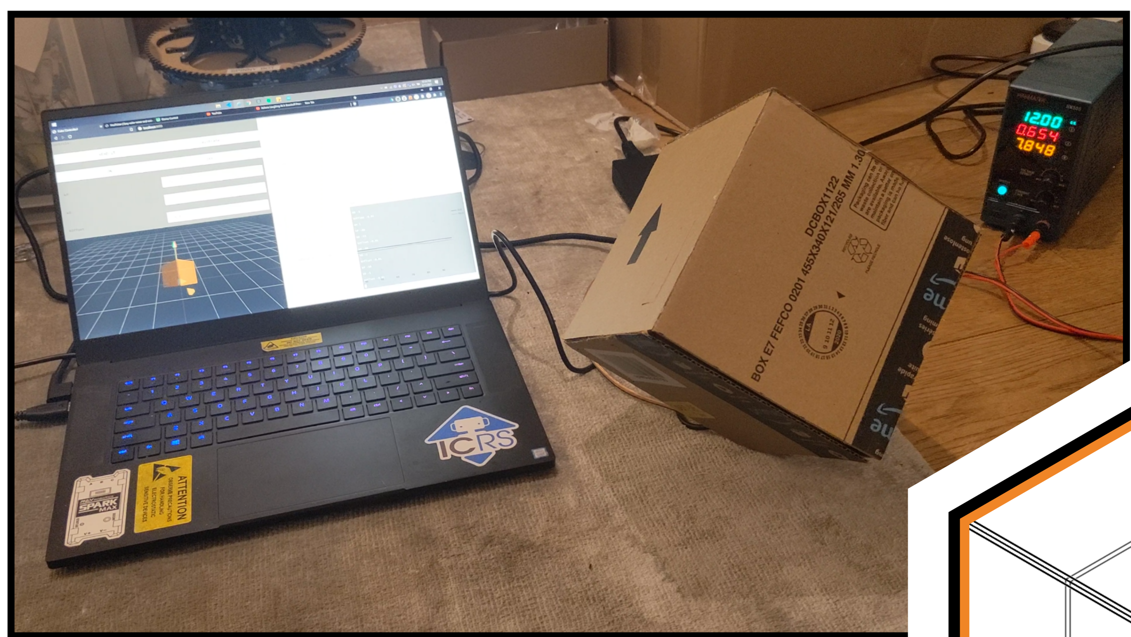
The Goal

The project I settled on attempting, was building a self balancing cube. While this was an ambitions goal, in a classic "how hard could it be" attitude: I reasoned I could break it down into some small, achievable chunks and work through them step by step. While it took me quite a while to go through, I greatly appreciated the smaller milestones to complete. These smaller chunks also gave me good "abort points" that I could re-evaluate my approach at, to make sure I wasn't getting sucked into something that would never work.

The Final Cube

The control strategy for the 3 dimensional cube was fundamentally the same as the 1 dimensional prototypes. Angles were treated in terms of roll, pitch and yaw. At the balance point (0 angle), roll and pitch are orthogonal, meaning controlling the cube can be treated like controlling two, independant 1D cubes simultaneously. Yaw is not affected by gravity, so as long as the cube do esnot induce yaw torques with its control strategy, it will remain stable. The final controller duplicated the 1D cube controller for pitch and roll, the converted those commands into individual wheel torques. After another long period of tuning, testing and debugging, a stabilizing controller was achieved.

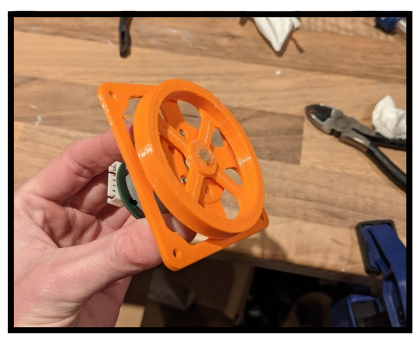
I am incredibly satisfied of the result, this was something I was never certain could work. It is an almost comical fusion of complex control theory with hardware built from stuff I had laying around my house. I thought it would be fun to post a quick video of an initial prototype to twitter, and to my suprise it ended gaining over 20K views and likes from people who worked at SpaceX, NASA and Blue Origin; apparently the internet likes crazy stuff built out of cardboard as well.



Viability Tests

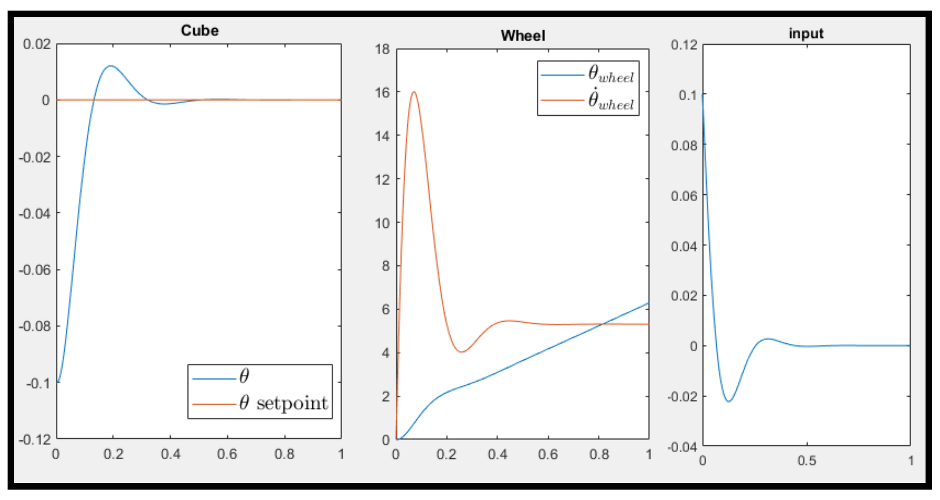
Test 1 - Hardware

Before moving onto anything more complex, a simple question had to be answered, could the DC motors produce enough torque to move a cube of reasonable size and weight. A scaled down frame and reaction wheel were built to test this, and the results were very promising, the system had more then enough power to twist itself around.



Test 2 - Software

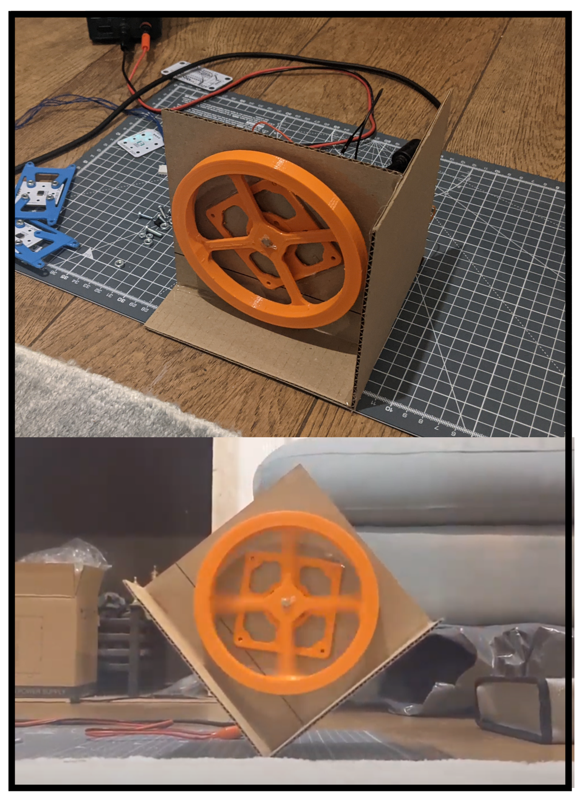
Unfortunately, a more complex, non-linear control strategy such as MPC was not feesable to realize on the hardware (Arduino) and the timeframe given. So, before attempting to control the actual hardware, a simulation was created to demonstrate that you could stabilize this system with state-space linear feedback and get a feel for the dynamics. A MATLAB model was written and estimated constants were pulled from the CAD model and verified with experimental data.

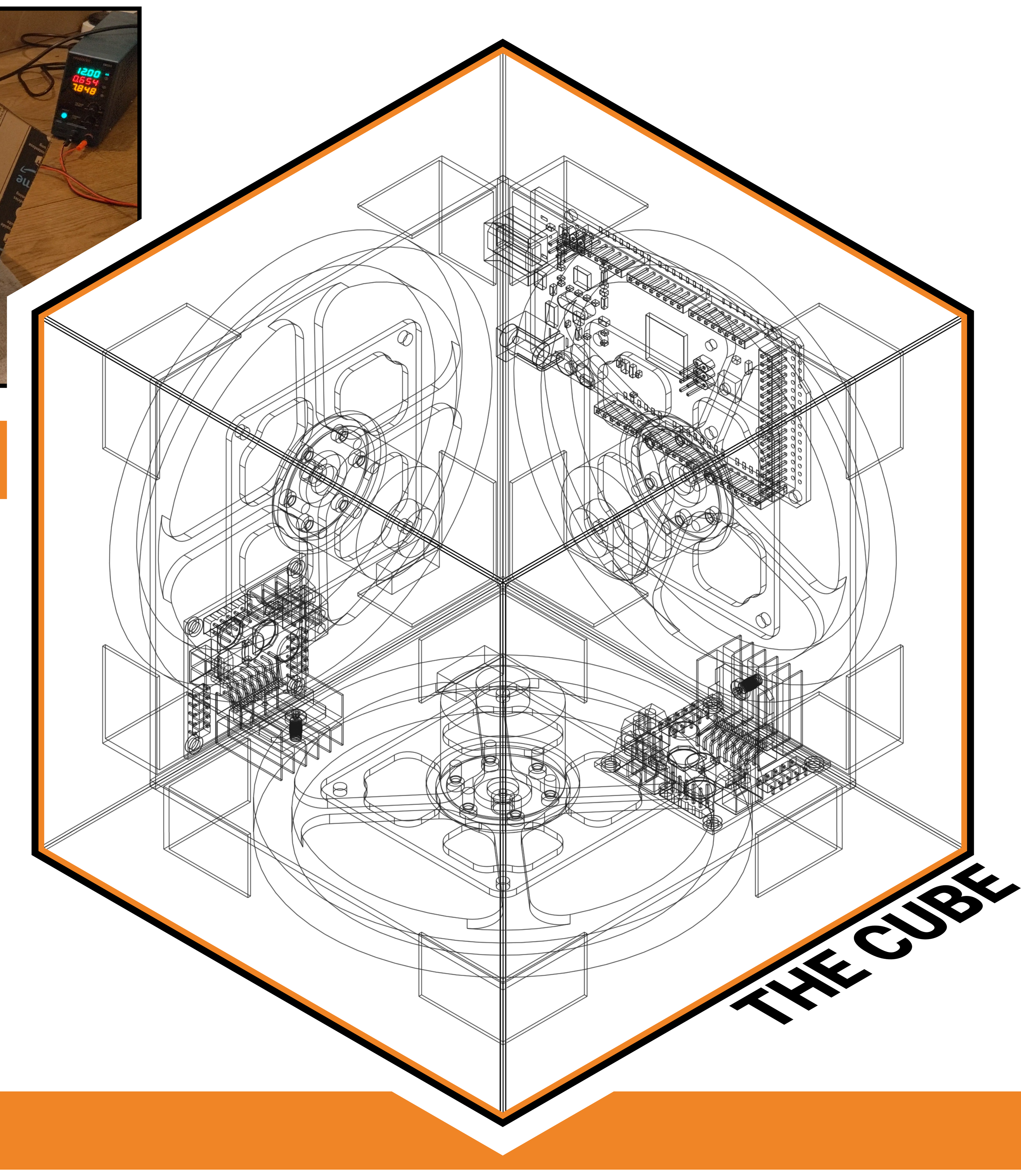


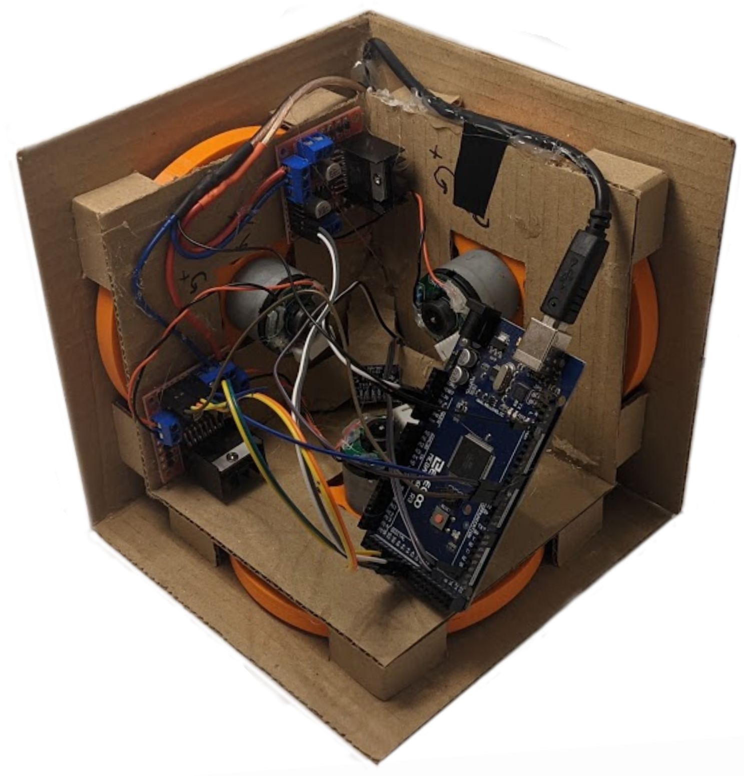
Test 3 - 1 Degree of Freedom

After proving that the hardware was capable of providing the torques to stabilize itself, and that a software controller on an Arduino had the ability to stabilize a simulated model, the last step before upgrading to 3 dimensions was to successfully implement a stabilizing controller on a 1 dimensional version of the full cube.

It quickly became apparent that an incorrect assumption had been made during simulation. The cube could momentarily stabilize itself, however the reaction wheel would quickly spin up, causing the motor to saturate and the cube to fall over. After some investigation, an explanation was found: if the 0-angle setpoint was not perfectly at the balance point, a constant torque was required to hold angle, resulting in constantly increasing wheel speed. This was combatted by integrating control output and adjusting the setpoint to result in no net output. This worked well and was even able to recover from live center of mass changes.







Stats

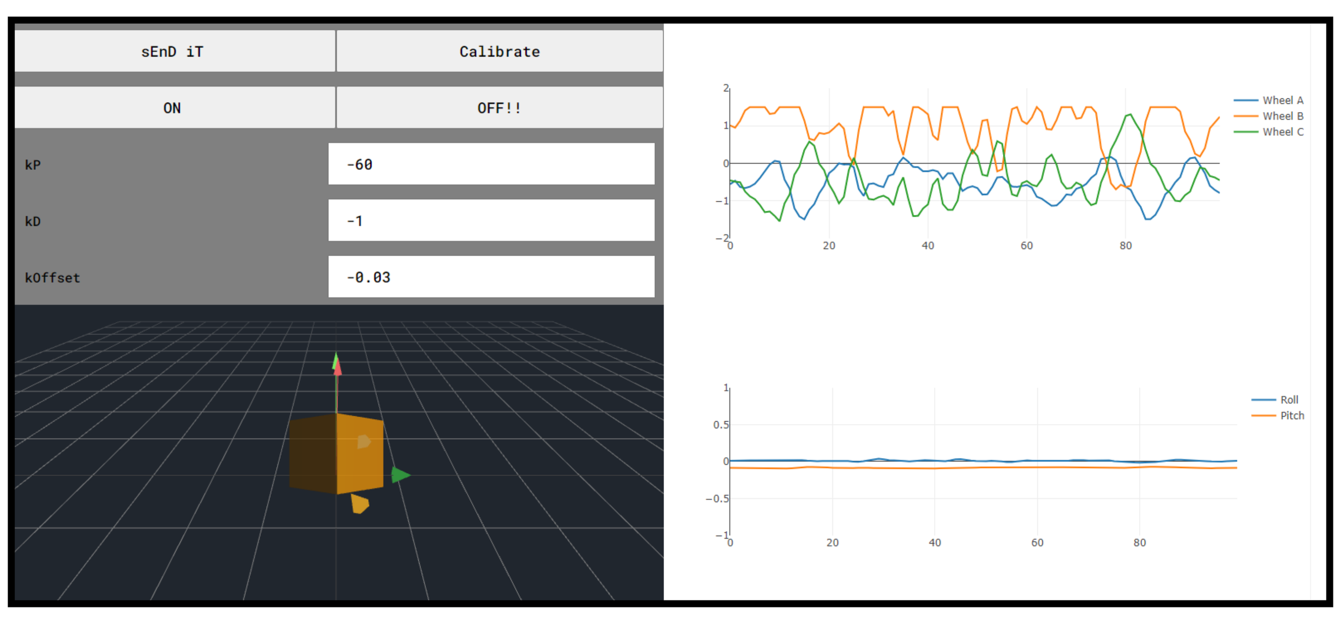
1098 lines of code
- viz: 556
- cube: 219
- controller: 158
- custom libs: 165

33 pieces of cardboard
6 3D printed components

3 RS-550 scale motors
2 Arduino Mega 2560s
2 LN298 motor drivers
1 MPU6050 IMU

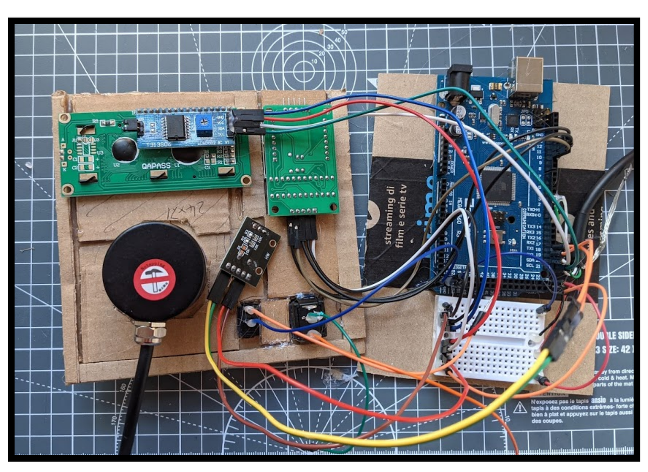
Interaction

Visulization Software



To aid in tuning, I created an accompanying visualization software using node, plot.ly and three.js. This receives data streamed from the cube over serial and live plots graphs as well as a 3D representation of the cube's angle data. It can also send data back, allowing constants to be tuned and for it to be used as a soft stop.

Control Box



The control box houses various sensors and displays to aid in the tuning of the cube's control algorithm. Specifically: Two rotary encoders: a small one for constant selection and a large one for value adjustment. A push switch for setpoint calibration, an on/off rocker switch to enable/disable the motors. It also has a dot matrix screen to show roll/pitch values and a 16x2 LCD that is used as a text display.