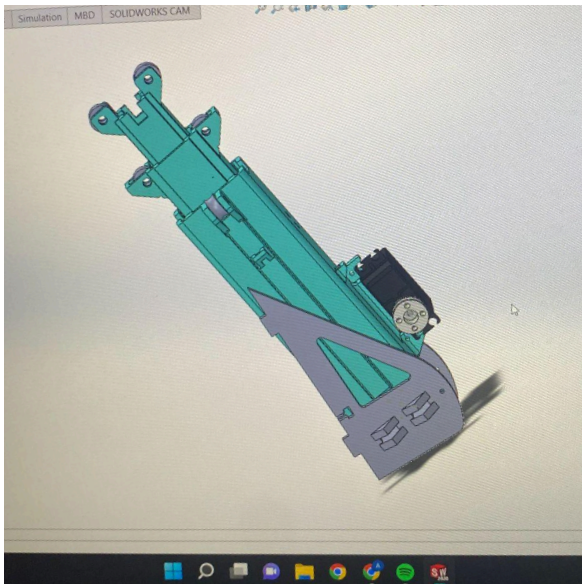
Anthony Marinov's Portfolio

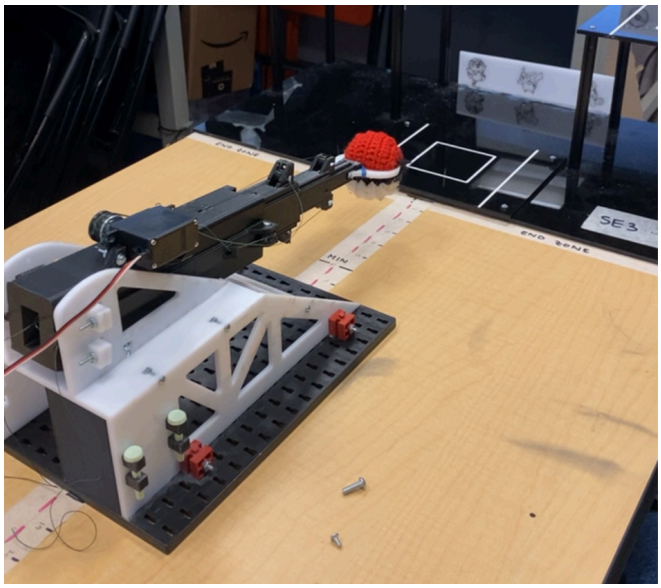
*****WORK IN PROGRESS: This document is being developed to provide an easily navigable catalog of my physical projects, and software projects that I am not able to post on GitHub. It is currently in the process of being built, and is continually developed as time permits. My coding projects can be found at my GitHub :)*****

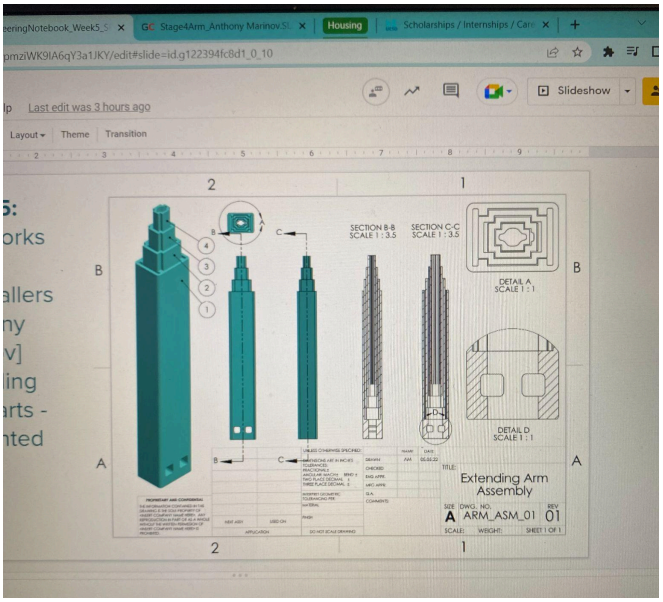
Physical Design Projects

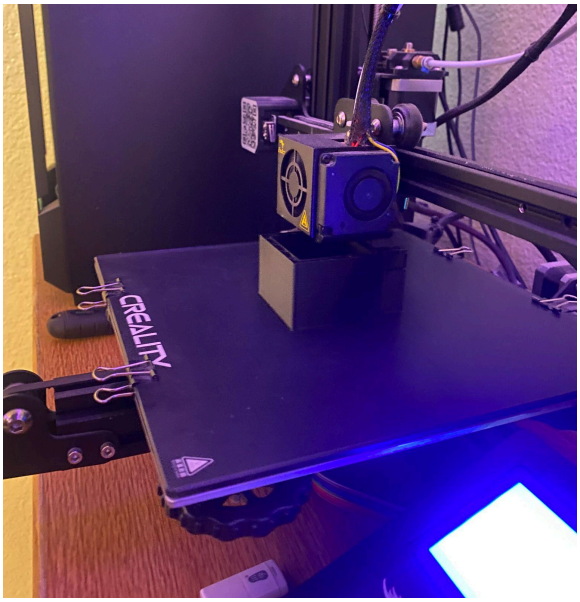
Extending Arm - Design Build Competition

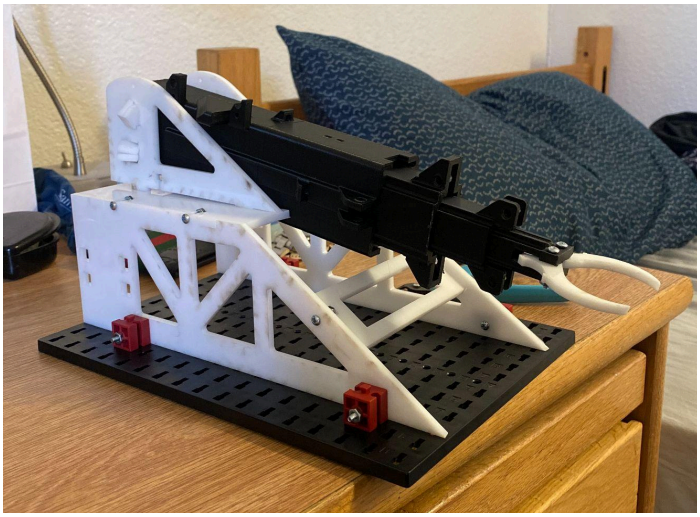
Design-build competition completed as a term project for SE3 at UC San Diego. Placed 1st out of 46 teams, with a 73% higher score than the 2nd place team.

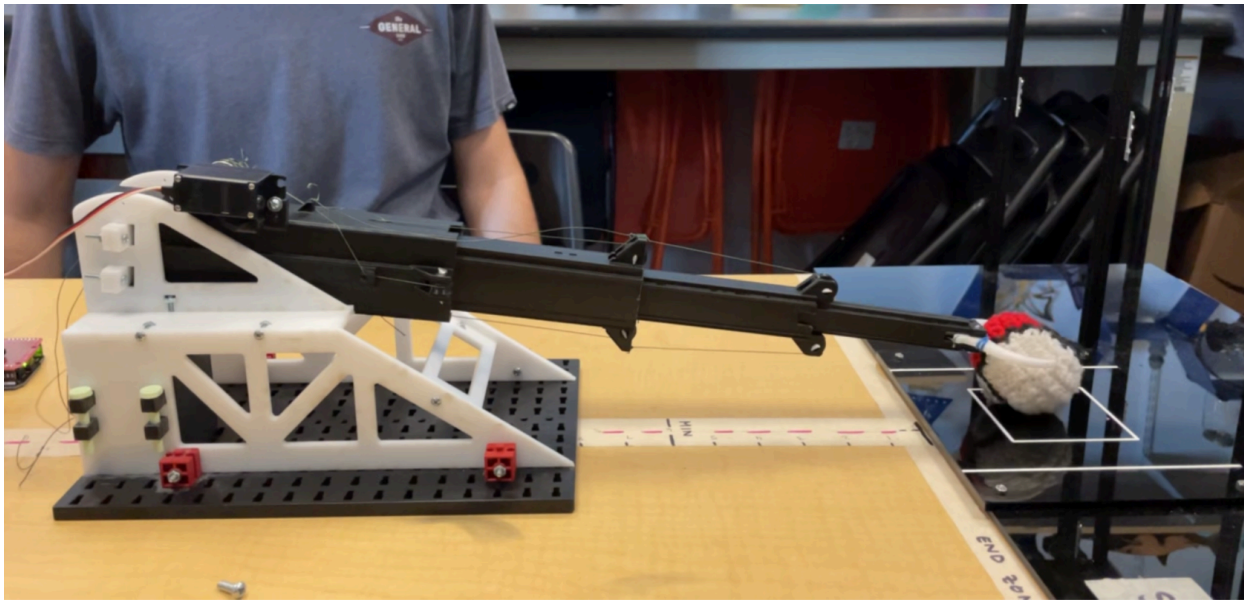





Housing
Slideshow
Last edit was 3 hours ago
Layout
Theme
Transition
SECTION B-B
SCALE 1 : 3.5
SECTION C-C
SCALE 1 : 3.5
DETAIL A
SCALE 1 : 1
DETAIL D
SCALE 1 : 1
Extending Arm
Assembly
SIZE
A
DWG. NO.
ARM_ASM_01
REV
01
SCALE:
WEIGHT:
SHEET 1 OF 1


CREALITY





Purpose

The goal of this project was to create a structure/mechanism that could take a ball and place it on scoring platforms at least 6 inches away. The mechanism had to be entirely independent of any human interaction and actuated by a servomotor. The project was required to be fully designed in Solidworks, with a complete set of drawings produced in AutoCAD. The mechanism also had to fit in a limited area, limiting the size of the contraption we could create.

Given

Available materials/resources included:

- 3D printing
- Laser-cut polycarbonate sheets
- Servomotor
- Rubber bands
- Prefabricated plastic beams and connectors
- Steel ball bearings
- Aluminum tube
- Braided thread
- Screws and nuts

Solution

My team and I had to first decide on the intent of our mechanism: would we develop a catapult to launch the ball onto the scoring platform like every other team was developing, or would we try to tackle

a design that fundamentally addresses the problem with reliable and consistent scoring? Weighing the options, the catapult would at best be extremely inconsistent, and more likely to not work at all due to the balls sliding off of the scoring platform. The catapult also faces many sources of inconsistency due to the use of rubber bands and sensitivity to friction between joints in determining the ball's launch path. This led us to brainstorming other solutions that venture “outside the box” in an attempt to gain consistent performance that would set us far apart from every other team.

In my time on my high school’s FIRST robotics team (696), we built many extending arms for a variety of purposes, which led me to suggest this as a solution to this problem. My team agreed it would likely be our best option, but were concerned about the feasibility of developing such a mechanism in the limited time we had for the competition. We came to the consensus that if we want to win the competition, getting this extending mechanism functional would assure us victory, and we went full steam ahead into developing it. Because of my familiarity with such mechanisms and experience with 3D CAD design, I took on the design role.

After doing some torque calculations and considering the very limited range of the servomotor, I decided that we should pursue a cascading lift mechanism, where every arm stage extends at the same rate, rather than a sequential arm where the arms extend one at a time. This cut the required range of the servomotor by 4 times, for our 4 stage arm. I also had to consider the limited torque output of the servo, as it would have to drive the arm extension. A 3D printed spool was attached to the servo which held the thread that ran along the pulley system of the arms. Using the servo specifications, I sized the spool with the largest diameter possible to balance long extension range with a torque that the servo could handle with some margin. I then chose the number of stages and sized the arm lengths to match the servo and spools range of possible extension.

To get a powerless claw system to release the ball on the scoring platform, I came up with an idea to link the claws to the base stage of the extending arm with thread, so that when the arm neared full extension, the thread would become fully taught and pull the claws back as the servo continues to drive the arm forward, releasing the ball. This system actually turned out to be extremely effective and consistent.

To get this mechanism in its final form, it required:

- ~ 20 hours of Solidworks design time
- 50 + hours of 3D print time (my printer ran nonstop for a week)
- Almost a full spool of PLA filament
- 2 + hours of laser cutting time

In the end, this design was extremely worth it as it absolutely blew every other team’s mechanism out of the water. Out of the 3 competition runs used for scoring, we scored in all 3, whereas the second place team had 1 scored ball. We also used 0 rubber bands, which was advantageous because they posed a significant penalty in the scoring index.

The attached pictures above are from various stages of the design process. Videos of this mechanism working are posted on my GitHub, and can be found here.

Achievements

- 1st Place out of 46 teams, with a 73% higher score than the 2nd place team.
- The only team with a unique design that avoided using a catapult mechanism and rubber bands.

Helmet Crash Detection Sensor

Purpose

The purpose of this project was to identify a problem (with some personal weight) and create a gadget that addressed it. The gadget was not going to be actually manufactured for this competition, but the design had to take into account manufacturability and was graded on it. This was done as a senior project for my high school's FIRST robotics team (696).

Solution

I do not have any pictures or records of this project because my school account was deleted after I graduated, before I could transfer all of my files to my computer :(.

I was an avid mountain bike rider at the time, and came up with the idea of creating a crash detection sensor that could be mounted to a helmet. This gadget would be small enough to be unnoticed by the rider, and would contact emergency services in the event of an accident where the rider is unresponsive or seriously injured. This is especially important as many mountain bikers ride alone in the mountains, away from much human traffic, leaving them helpless in the event of a serious incident.

I used Onshape for the 3D CAD models, as that is what our robotics teacher had newly standardized for that year, transitioning from Autodesk Inventor and Fusion 360 the prior year (eliminating the need for the entire server our teacher had built in the lab to order to share and store files).

The gadget I developed was a roughly 2" x 1" x ½" box, mounted to the top of a helmet connected to a small battery pack on the back of the helmet, that contained a chip with an integrated accelerometer, GPS, speaker, and bluetooth compatibility, with a small button and status light on the back of the box. I also developed sample code and a sample app interface to go along with this gadget. The gadget was designed to be mounted on the top of a helmet, rather than on the bike or rider's backpack, because it is the most reliable place where false impact detections won't commonly occur and the most critical area of the human body, where an impact is potentially debilitating.

The gadget automatically turns on whenever accelerations lasting more than 5 seconds take place (the rider puts the helmet on), and turns off whenever no accelerations are detected for 2 minutes. When the gadget senses an acceleration above a set threshold (that I calculated to be low enough to capture a decent impact, but high enough to not be exceeded under normal riding circumstances), it would set off the emergency service sequence. In this sequence, the gadget would emit loud beeps for 20 seconds to notify the user that the emergency sequence has begun, allowing the user to press the button on the gadget to cancel the sequence if triggered by false alarm or an incident that is not serious enough to warrant emergency services. If the sequence is not canceled, the gadget sends the rider's GPS coordinates to the user's phone, linked by bluetooth through the gadget's app, which sends a SOS request to emergency services with the rider's GPS coordinates.

There are limitations to this design that I was aware of, but was not prepared to address for the scope of the project. Namely, the biggest issue was the reliance on the user's phone to contact emergency services, as in some cases the user may be in a place with no cellular connection. This problem was not addressed because any possible solution would make the gadget way more bulky and not realistic to be worn by a user, and would have required a proprietary microchip, which makes the manufacturability not realistic for this project's scope. I also considered linking the device to a satellite communication device (known name brand ones), but this would require the satellite communication device to be bluetooth enabled.

Software/Coding Projects

AI / ML Engineering Design Program (developed for MiTek)

The information regarding this project is sensitive to MiTek, and therefore I am not able to share too much detail about it. I created a write up about this project and developed an animation with Python showing the program's functionality that management approved for me to post on LinkedIn, sharing as many details as I could. That post can be found here.

Some additional details/reflections are below:

I put together a file package containing all of the necessary dependencies, Python files, trained model files, training data sets, and 50 pages of documentation (yes 50, I spent a week on this). The documentation contains detailed instructions on how to use the design program, troubleshoot some possible faults, descriptions of the inner workings and logic, as well as detailed instructions on how to collect more data, process/engineer the data, and retrain the ML models. I tested the package extensively to get rid of as many bugs as possible, and allow for easy usage and future development. I also led a presentation to 15 of MiTek's engineers and management, many of which were presidents or vice-presidents of the global teams they managed, where I explained the functionality, benefits, and limitations of this design program, and led a demo where I taught the usage of the program.

I'm heavily obsessed with AI and ML, and I see endless potential in developing this tech for technical applications. We have barely scratched the surface of what AI can do, and I believe technical applications are a largely untapped application of AI. That's why I was so determined in developing this program at MiTek. The program is very much a prototype, and mostly serves as a proof of concept that AI design in the engineering field is possible and extremely beneficial. I was able to uncover a new design philosophy and new usage cases for some of MiTek's products with this program that none of the senior engineers that I worked with had ever even considered, producing much more time and cost efficient lateral system designs than previously achieved with these products. I was able to generate optimized lateral system designs for buildings containing almost a thousand separate structural walls in a matter of a couple of minutes, which is impossible to traditionally do (the program could achieve this in a couple of seconds or less, but is bottlenecked by having to transfer data in and out of Excel to work with MiTek's existing Excel tools). Manually designing such a lateral system yields very unoptimized/inefficient results and takes multiple days of tedious work.

A really interesting idea I have since thought about would be to try to develop and apply self-attention in a transformer architecture, similar to GPT-2, to this technical design generation. One of the major limitations of the program I built was the inability for each wall in the lateral system to essentially "communicate" with other walls. Changing a wall in the lateral system has implications on the other walls in the system; my program neglected that and designed each wall independently of other walls in the system. I knew this was a limitation at the time, but I did not have the time during my internship to really tackle this issue. I tried to work around it by engineering the features of the ML models to take in some building data parameters from sections of the building near the specific wall that was being designed, in order for the model to learn at least a tiny bit of some of the relationships that the walls in the lateral system have with each other. I used cross validations to help tune the hyperparameters of these models as well, within the range of training performance reasonable for the computer I was working on, as I didn't have access to any external computing systems. Limited data was a significant challenge as well, as I collected all of the training data by generating designs for the few sample projects I had with an algorithm that I developed, which was extremely slow due to repeated data transfer with Excel, hence my transition to ML models (ML models cut this design time down by over 20 times). I think implementing a transformer with a self-attention architecture characteristic of large-language models, along with more training data, could yield some seriously revolutionary results, and is something I would love to research/develop in the future.

Soil Water Retention ML Model

The write up for this is coming soon. I'm slowly adding stuff to this doc as time allows me :)