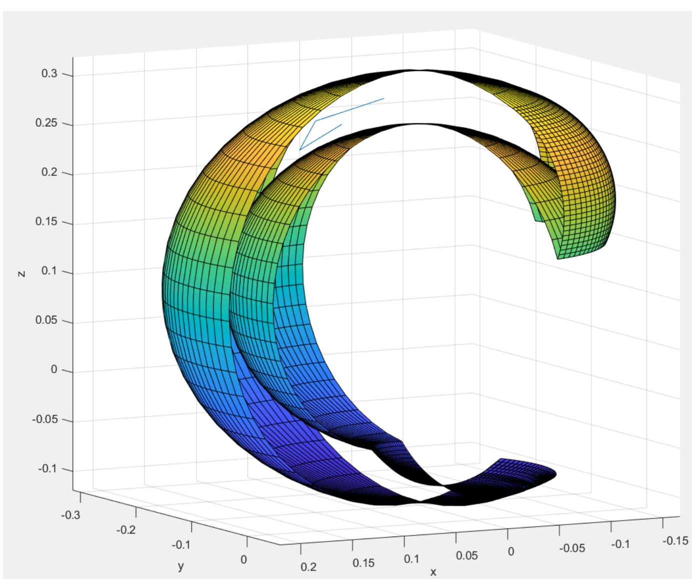
Andrew Zhang's Portfolio

NAO robot arm trajectory generation and motion control

For Robot Dynamics and Control course

Arm of NAO v6 robot



Limits of right hand end effector in 3D Cartesian space, and four points defining a tested trajectory

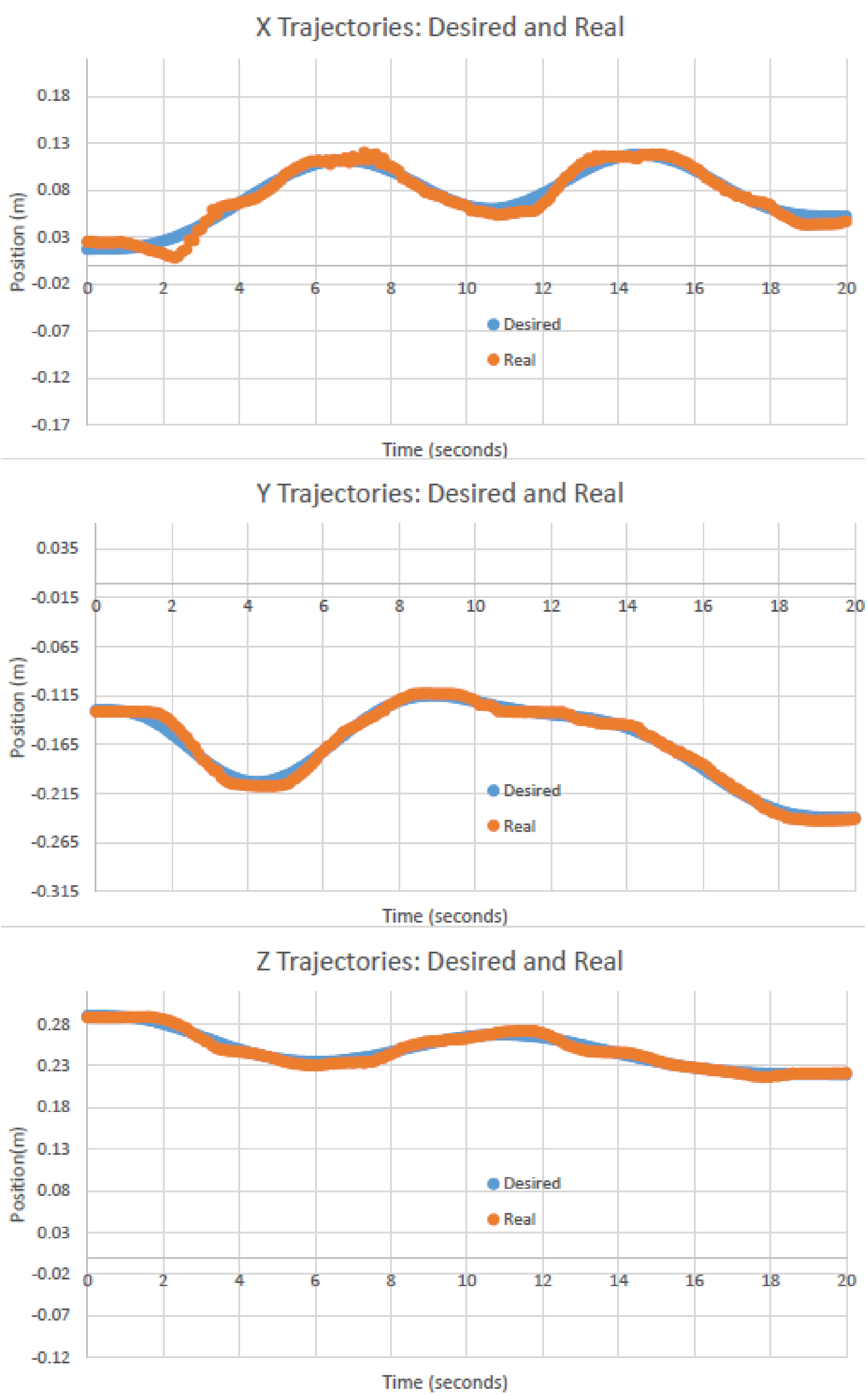
The aim was to make a non-trivial trajectory for the end effector of a NAO v6 robot arm and make the end-effector of the arm follow the trajectory as closely as possible. I worked on this project with a partner. This required managing only three degrees of freedom in the robot arm (two at the shoulder and one at the elbow) as only the position mattered, and not the orientation.

The limits of the right hand end effector were found and plotted in 3D Cartesian space using forward kinematics.

Points within the limits were selected and a minimum jerk trajectory was generated.

An inverse dynamics controller was used to determine the necessary joint torques, first for a simulated robot arm, then after tuning the gains, for a real robot arm.

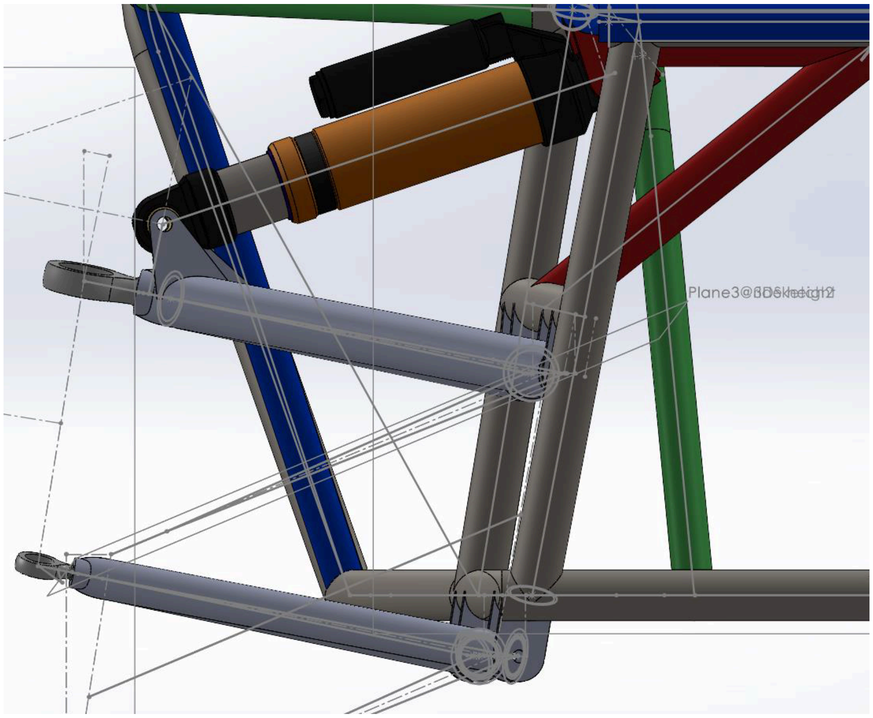
The arm was able to roughly follow the trajectory with measurable but small errors.



Desired and real trajectories of robot arm in x, y, and z dimensions

Front Suspension Design for Off-Road Vehicle

For University of Waterloo Baja SAE



View of suspension from the front

View of support structure

The University of Waterloo Baja SAE team worked to make a one seat all terrain vehicle with a structure containing the driver, for competitions with other university teams. I was responsible for designing the front suspension.

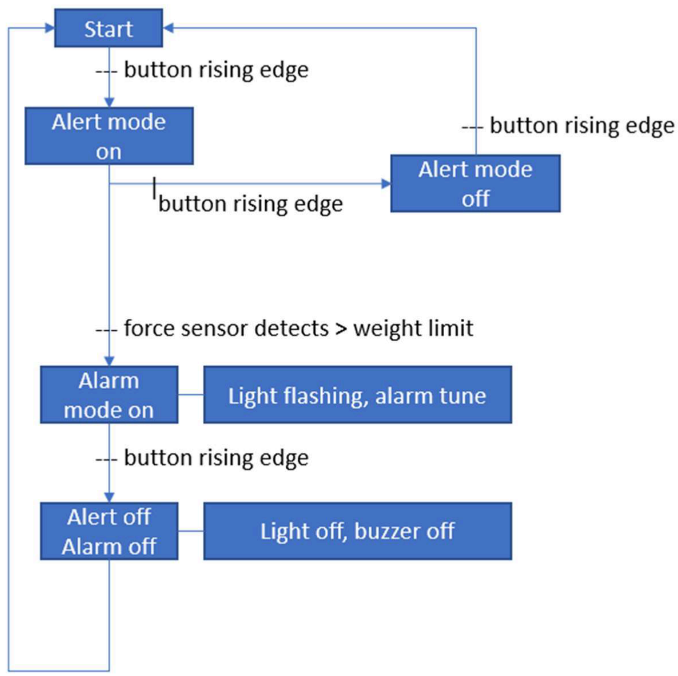
The criterion for the front suspension was to be able to withstand an instantaneous force of 3200 N per wheel without yielding with a safety factor of at least 1.2. The constraints were numerous, involving working with pre-existing parts of the vehicle, following competition rules, being able to accommodate parts that would be made later, and more.

After many preliminary designs were drawn in significant positions the suspension could be in, and tested in Solidworks Simulation for yielding, a near final design was reached. A support structure was added to the chassis to prevent buckling. A few adjustments had to be made after research was done on the feasibility of fabricating the design.

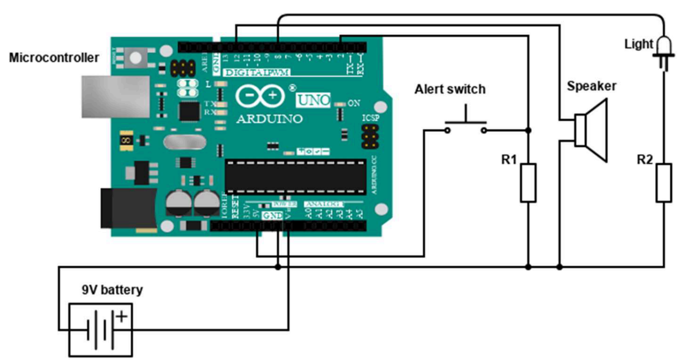
Ultimately, the final design could easily withstand an instantaneous force of 3200 N per wheel without yielding with a safety factor greater than 1.2 in the Solidworks Simulation, while also satisfying all constraints.

Kelp harvester alert system

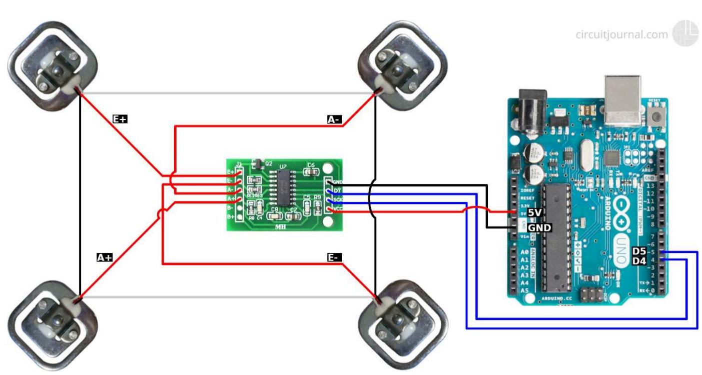
For undergraduate capstone design project



Simplified state diagram



Circuit diagram, excluding load cells



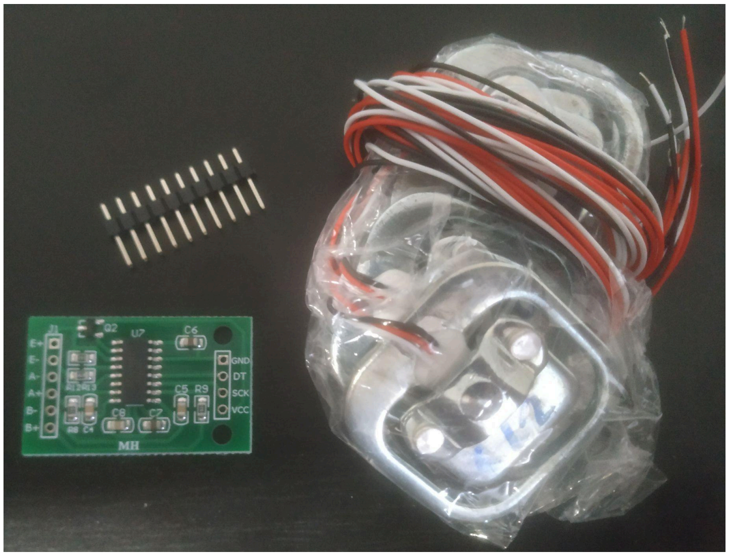
Load cells and amplifier layout and connections

The project was to create a system that could be installed on a kelp farmer's boat that would reduce the manual labor needed for harvesting. I was responsible for the alert system, which alerted the farmer that the kelp bin was full.

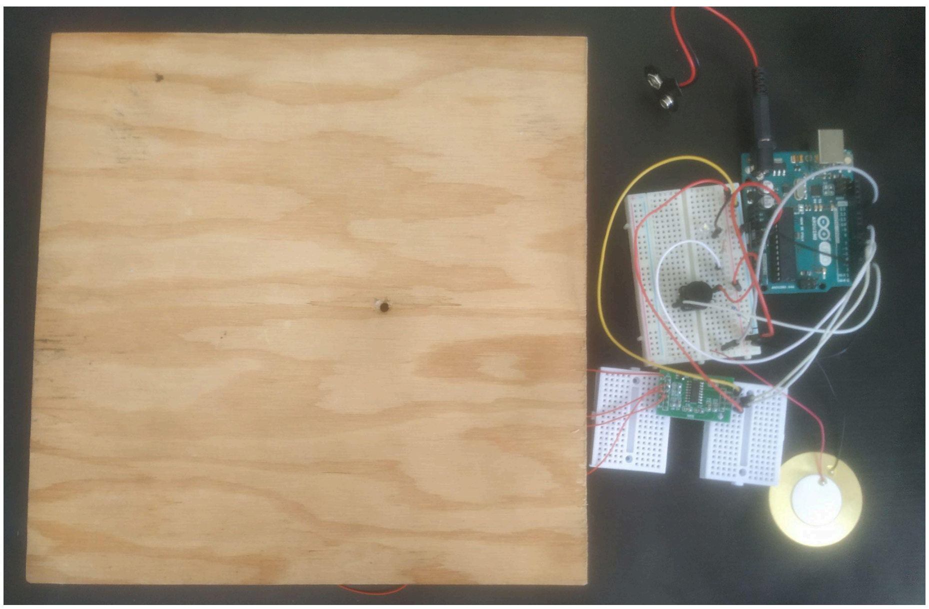
An Arduino UNO microcontroller was used to process sensor data and provide control signals. The microcontroller was powered by a 9V battery. It was programmed in C++.

Four load cells were attached to a wood board that was modified to accommodate them. They were wired together and the connections were soldered and covered with heat shrink tubes. They were soldered to a voltage signal amplifier. Pins were soldered to the amplifier so that it could be used with breadboards. Other connections were made via breadboard. The load cells were calibrated with known weights.

When the scale and alert system is turned on, a tone is briefly played and the LED is turned on and steady, indicating the system is in the ready state. When the threshold weight is detected, the LED starts flashing and the buzzer plays a melody on a loop. When the user presses and releases the button, the LED turns off and the buzzer stops playing. When the user presses and releases the button again, the system



Load cells, pins, and amplifier pre-assembly



Alert system, battery not included

Kelp harvester prototype. The alert system is covered by plastic for water resistance.

returns to the ready state, the LED is turned on and held steady, the scale is tared, and a tone is briefly played.

Implementing STOMP algorithm for robot arm obstacle avoidance

For Human Robot Interaction Course

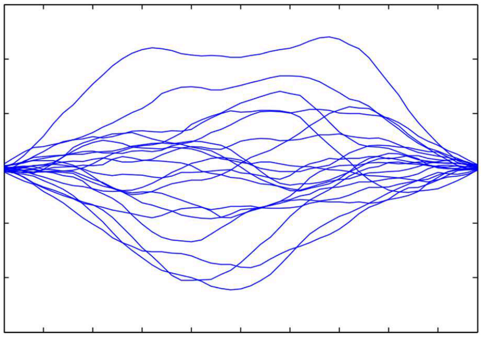
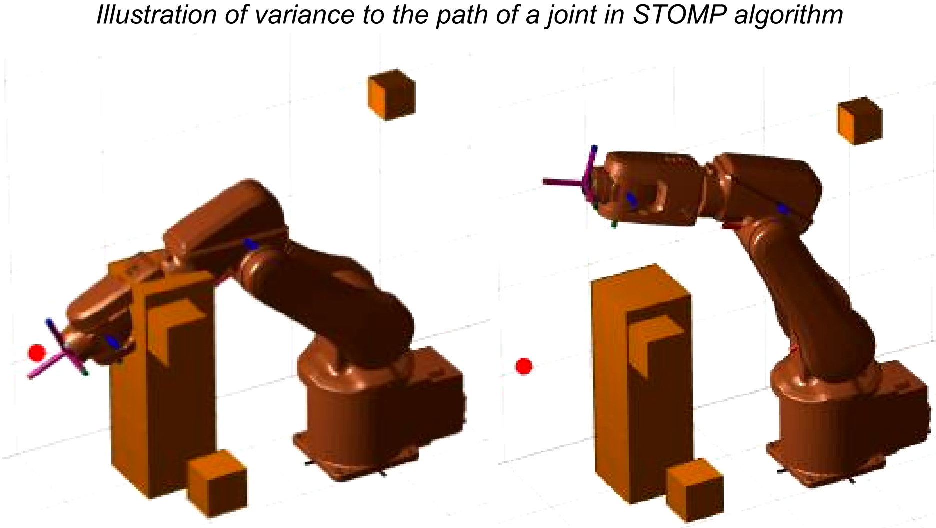
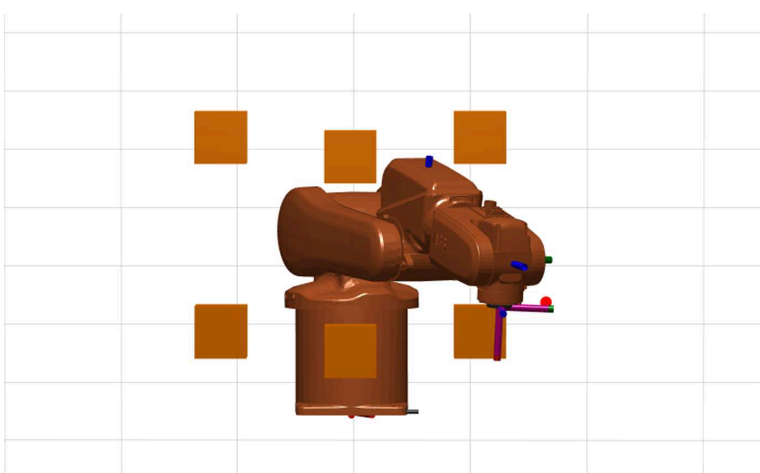


Illustration of variance to the path of a joint in STOMP algorithm



Left: Arm following partially optimized trajectory and colliding with object
Right: Arm following fully optimized trajectory



Robot arm completing second obstacle scenario

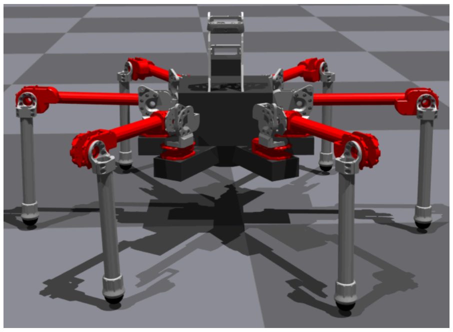
An ABB IRB 120 6 degree of freedom robot model in MATLAB was made to move the end effector from one position to another while the entire arm avoided colliding with obstacles. I worked on this project with two other team members.

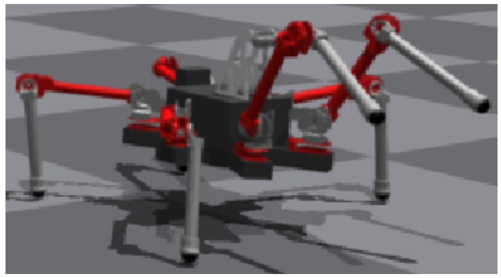
This was done by implementing the STOMP (Stochastic trajectory optimization for motion planning) algorithm.

Trajectories were found such that the arm was able to successfully complete two obstacle avoidance scenarios. It was also able to complete the second scenario with constraints on the end effector position and orientation.

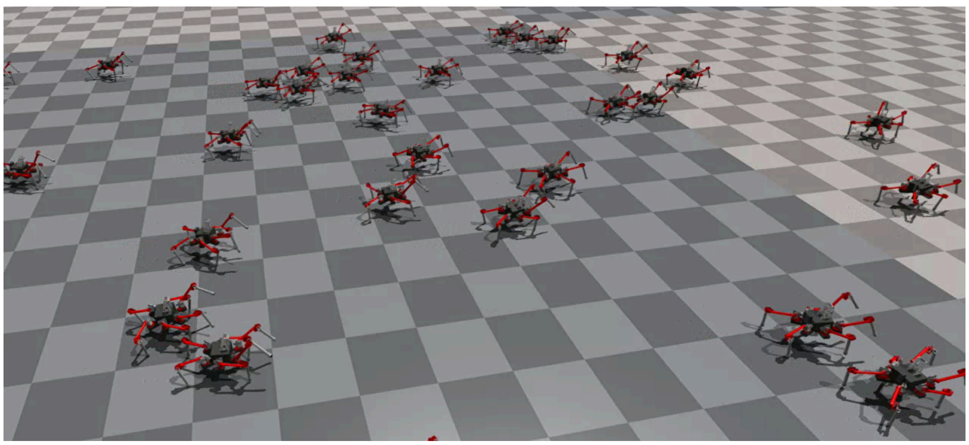
Training a hexapod robot

For Machine Learning in Robotics course





Left: hexapod robot standing in Isaac Gym physics simulation
Right: hexapod robot ready to push an object



Multiple hexapod robots training in parallel

A six-legged robot was made to learn to walk on four legs with two legs up and out such that it could push an object. I worked on this project with two other team members.

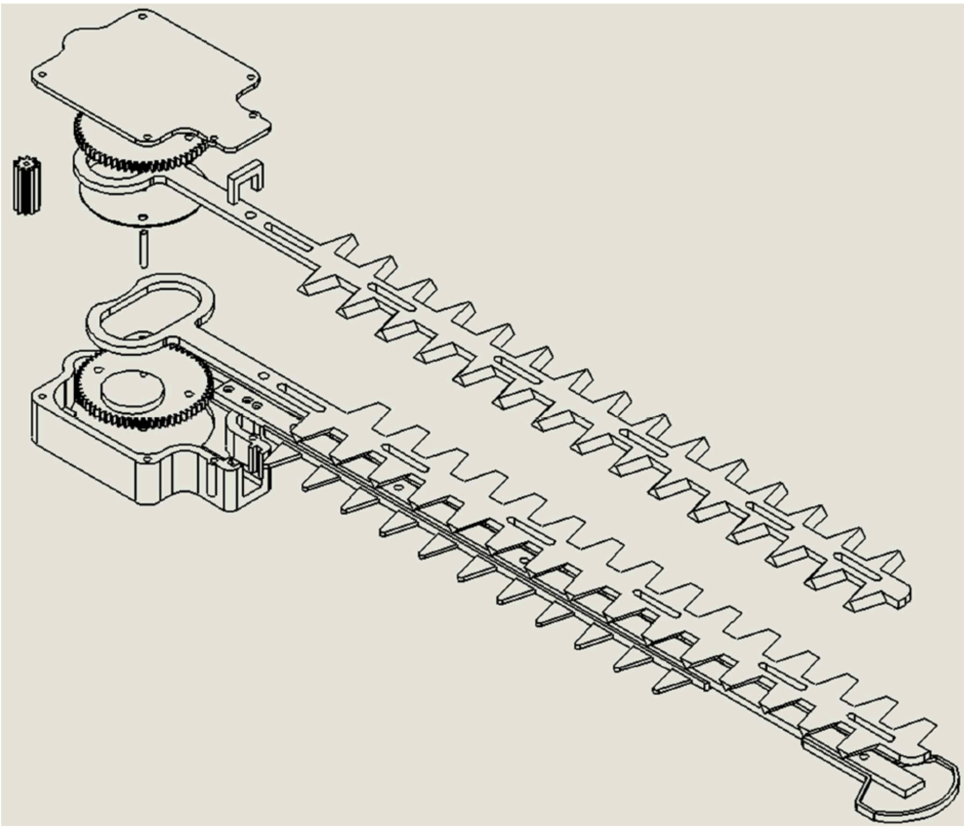
This was accomplished using reinforcement learning in the Isaac Gym physics simulation. Programming was done with Python, and the program was run on a Linux OS.

After testing, four layer structures were used for the actor and critic neural networks.

The robot was generally able to walk to the goal position with its two pushing limbs positioned correctly. Unfortunately we did not have the opportunity to apply the model to a physical robot.

Hedge Trimmer 3D CAD model

For Computer-Aided Design course



Hedge trimmer mechanism with case and guard

I modelled the mechanism in a hedge trimmer using Solidworks. As the smaller gear is turned, the larger gears are engaged, causing the trimmer blades to oscillate. This is due to a protrusion from the larger gears contacting the blades like a pin in a slot.

Flower Pollinator Robot

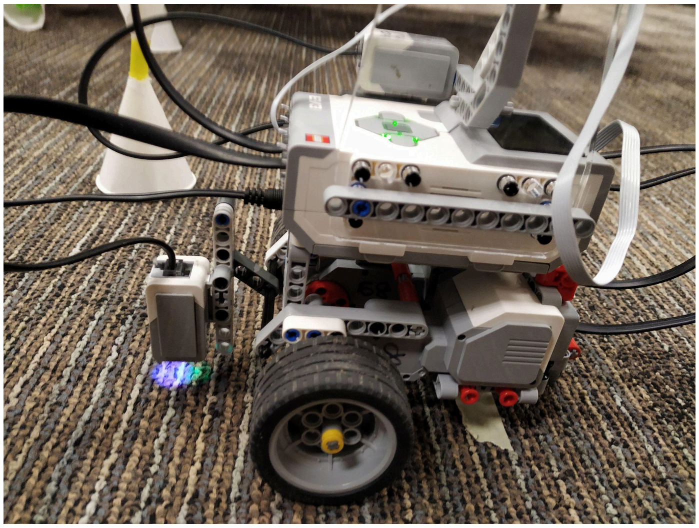
For Introduction to Mechanical Engineering Practice course

A robot was made to pollinate flowers. This project was completed with two other team members.

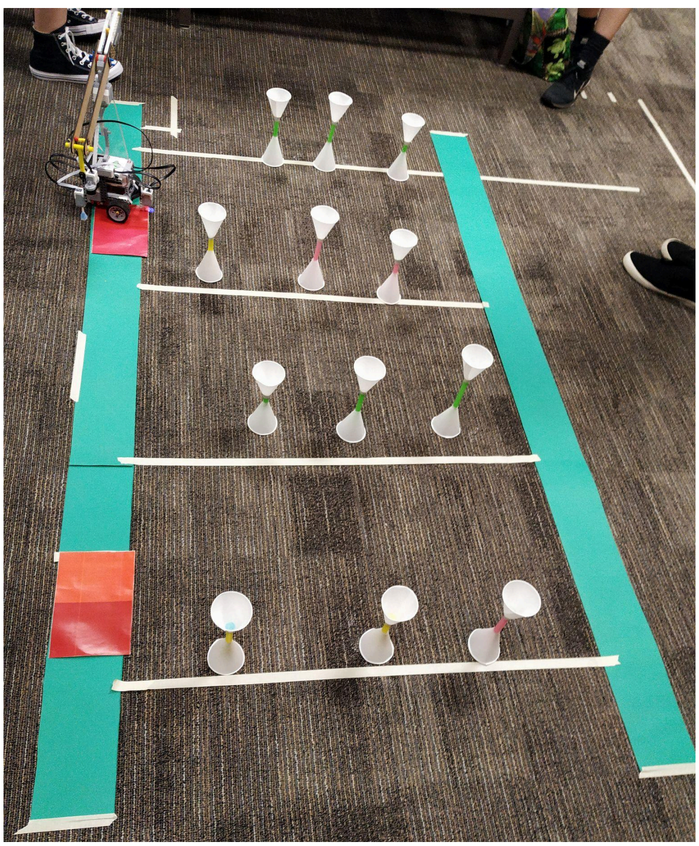
The robot was built using the Lego Mindstorm EV3 platform. It used 3 motors, color, ultrasonic, and touch sensors. Fiberboard was lasercut to make parts for the pollinating arm.

The robot moved forward with two of the motors until the ultrasonic sensor detected a flower. The arm on the robot would lower until the pollen attachment on the touch sensor made contact with the flower. The arm would then raise and the robot would move to the next flower. The color sensor was used so the robot could detect the bounds of the flower bed, which were marked out.

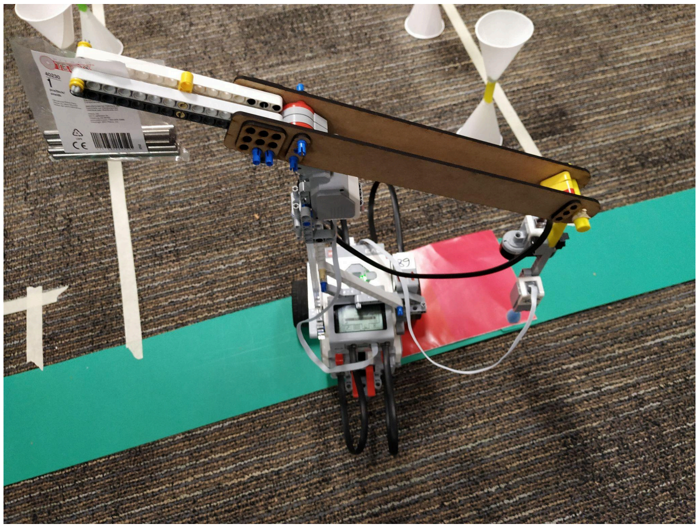
The robot was able to pollinate fake flowers of varying heights placed roughly in rows. A better motor for the arm and a more sensitive touch sensor would be needed before it could be allowed to try pollinating real flowers.



Side view of the bottom portion of the pollinator robot



Pollinator robot on a trial course flower bed, with fake flowers of varying heights



View of pollinator robot from above and behind