Introduction

For my UC Berkeley Master of Engineering capstone, I am working as the controls/ hardware/ software/ mechanical engineer for Squishy Robotics' centrally actuated payload team.

Squishy Robotics is a startup based in Berkeley, California. Squishy's mission is to develop mobile robots which can be used to assist emergency responders. Squishy robots are tensegrities in structure, meaning that they are composed of members in pure tension and compression. Tensegrity structures are resilient to high impact (such as from being air-dropped via drone) and have many degrees of freedom.
Squishy Robotics Website: https://squishy-robotics.com/

The goal of the centrally actuated payload mobile robot team is to demonstrate the feasibility of a mobile robot using a quarter the amount of motors used in the current mobile robot prototype, Mobile Robot 2 (MR2) as seen in **Figure 1**.

Figure 1: MR2, Current Mobile Robot

Flounder 1.0: Punctuated Motion with 12 Motors

After designing a center payload, assembling hardware, building the structure, and troubleshooting, Flounder (**Figure 2**), Squishy's first centrally actuated mobile robot, achieved punctuated motion (**Figure 3**). See a video here. Flounder had bent rods to allow free motion of the center payload, bungee cords as the members in tension, and a center payload with 12 mounted motors. For motion, the motors would plug into the board, which would be held by an operator. The board would be incorporated into the center payload in a later iteration of Flounder. An operator would type in encoder values into a terminal and that signal would be wirelessly sent to the board, which would then operate the motors.

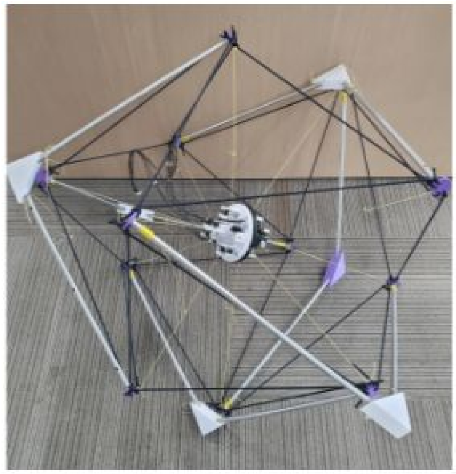
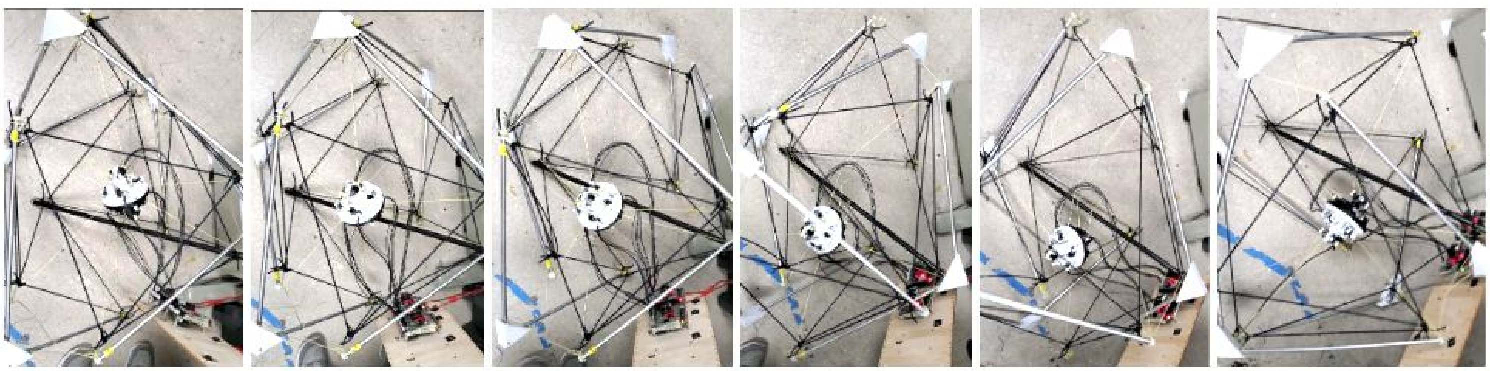


Figure 2: Flounder 1.0 - 12 Motors and Original Bent Rods



0.0 s 1.5 s 3.0 s 3.4 s 3.7 s 4.2 s

Figure 3: Flounder 1.0 Punctuated Motion

Flounder was iterated upon and became Flounder 1.1 (**Figure 4**), where the optimal angle for bent rods was found. It was found that in Flounder 1.0, the bent rods would get in the way of full punctuated motion.

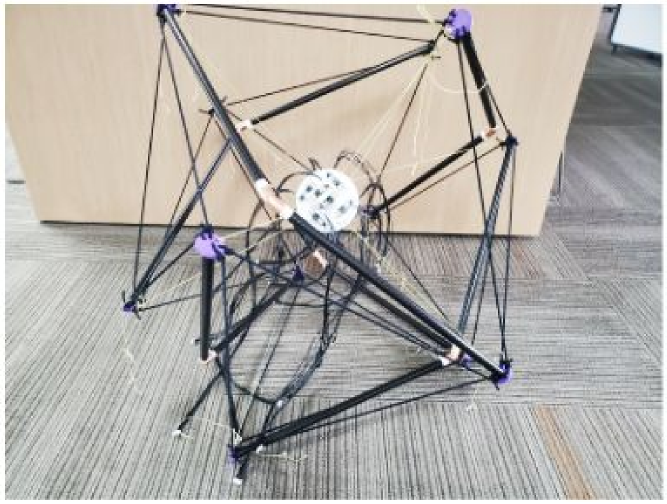
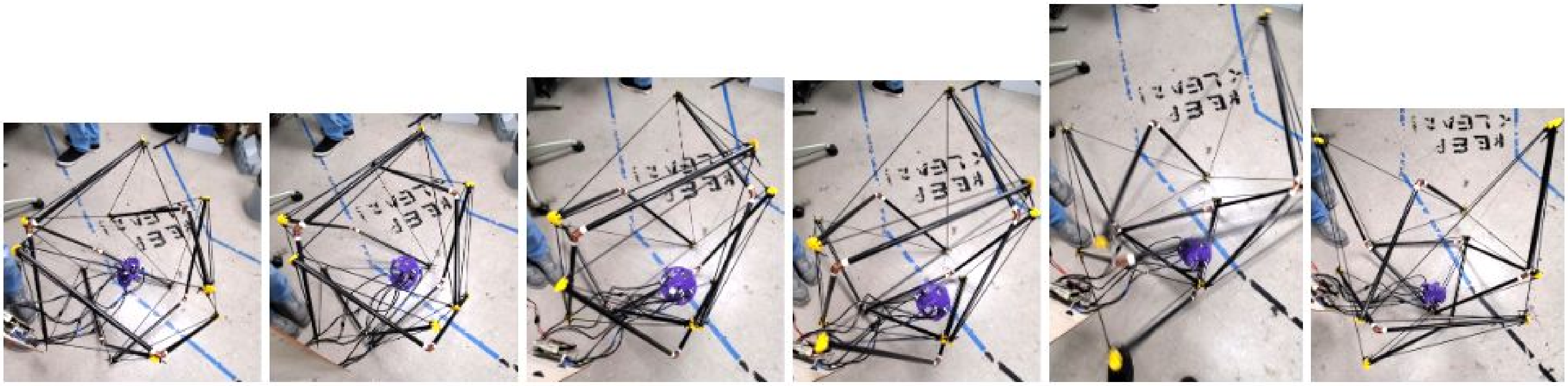


Figure 4: Flounder 1,1 - 12 Motors and Improved Bent Rods

Flounder 2.0: Punctuated Motion with 6 Motors

After the 12-motor Flounder was able to demonstrate punctuated motion in multiple directions, the second main iteration of Flounder was built, Flounder 2.0 (**Figure 5**). Flounder 2.0 incorporated a newly designed center payload for housing only 6 newly specced motors, and custom designed and machined double spools made of Aluminum. Flounder 2.0 achieved punctuated motion (**Figure 6**). See a video here. Flounder 2 achieved several instances of punctuated motion in all directions.

Figure 5: Flounder 2.0 - 6 Motors



0.0 s 8.7 s 14.0 s 14.2 s 14.3 s 14.5 s

Figure 6: Flounder 2.0 Punctuated Motion Before PID Fix

Flounder 2 experienced motor gear slippage, which was partially solved with a higher communication rate. The gear slippage caused inconsistencies is the needed encoder counts to achieve punctuated motion, and thus deemed the robot unreliable. It is believed that with the new code running, Flounder 2 achieved punctuated motion in less than 14 seconds. *A video will be uploaded after the shelter-in-place has been lifted.*

Flounder 3.0: Autonomous Punctuated Motion with 6 Motors

Flounder 3 (**Figure 7**) is being developed currently. The center payload will incorporate 6 motors as well as the board, which has not been incorporated in previous iterations of Flounder. An IMU, Adafruit's BNO055 will be used to attempt autonomous punctuated motion, that is, motion achieved without explicitly commanding particular motors as done in prior iterations of Flounder.

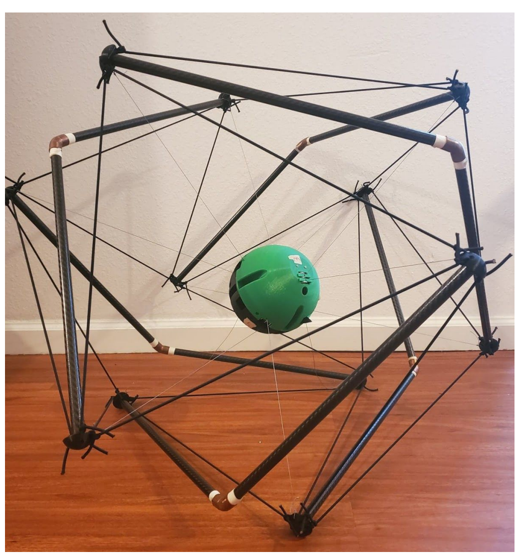


Figure 7: Flounder 3.0 - 6 Motors, Integrated Microcontroller

Centrally Actuated Mobile Robot Simulation

Squishy Robotics wants to employ advanced controls into its mobile robot, to most efficiently and effectively produce continuous motion. Simulations run based on Brian Cera's MATLAB code show that the robot can achieve continuous motion with model predictive control (MPC). It was found that a 0.01s simulated timestep is significantly better than a 0.1s simulated timestep, likely due to errors in the linearization of the dynamics. Note that all other variables between simulations were kept constant. **Figure 8** below shows the progression of the simulation when set at a 0.01s timestep.

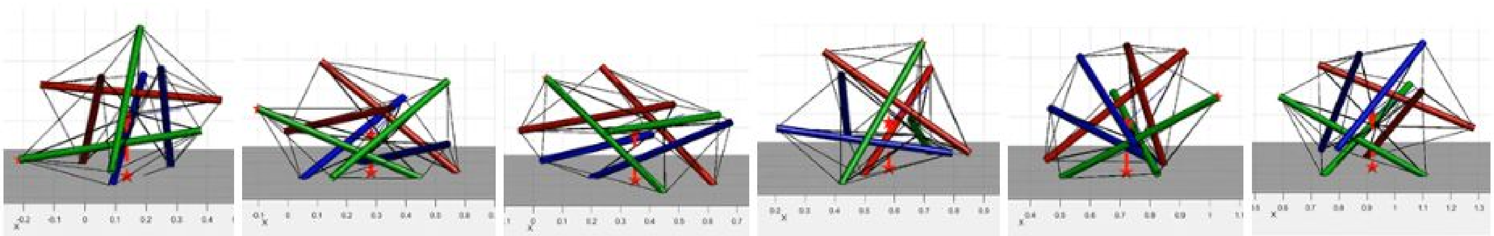


Figure 8: Central Payload Simulation with 0.01 s timestep