Overall Description

For my project, I modeled an passively adaptive underactuated fingertip that I use in my research. There fingertips feature 2 joints controlled by a single actuator via a tendon. Each joint has a torsion spring on them, where the stiffness of the proximal joint (the joint connected to the base) is 1/3-1/5 as much as the distal link (the joint connected directly to the fingertip). This creates a mechanism where the proximal joint moves much more than the distal joint to be similar to a human finger. To model an external contact, I had an angled plane move across the fingertip to push the finger back. Once the plane moves past the finger tip, the finger should spring back from the potential energy stored in the spring.

Modeling the system

Using measurements from the fingertip used on the Yale Model O Hand website https://www.eng.yale.edu/grablab/openhand/model_o.html), I first modeled the offsets between the finger joints and link lengths. I modeled the fingertip as an 8mm radius sphere. This roughly matches the fingertip size of the physical model. I also computed the masses from the fingertip CAD model assuming they were made from ABS plastic, and used spring stiffnesses from the expected range of spring stiffnesses. This resulted in a reasonably accurate model of the finger. The finger begins point upwards with all joints inline, and slowly gets pushed backwards.

Next, I worked on deriving the dynamics of the system. The kinetic energy in the system was modeled as 2 point masses. Initially, I did the point masses at the Center of Mass (COM) of the links. This caused the velocity to be far too high when the system was in free movement after contact with the plane. Instead, I model the mass at the end of the link, and that movement looked much more realistic. To model the potential energy (PE) of the system, I used the PE equation torsional spring, 1/2 k theta^2, and summed the PE in the spring. I then derived the Lagrangian using the equations from the notes.

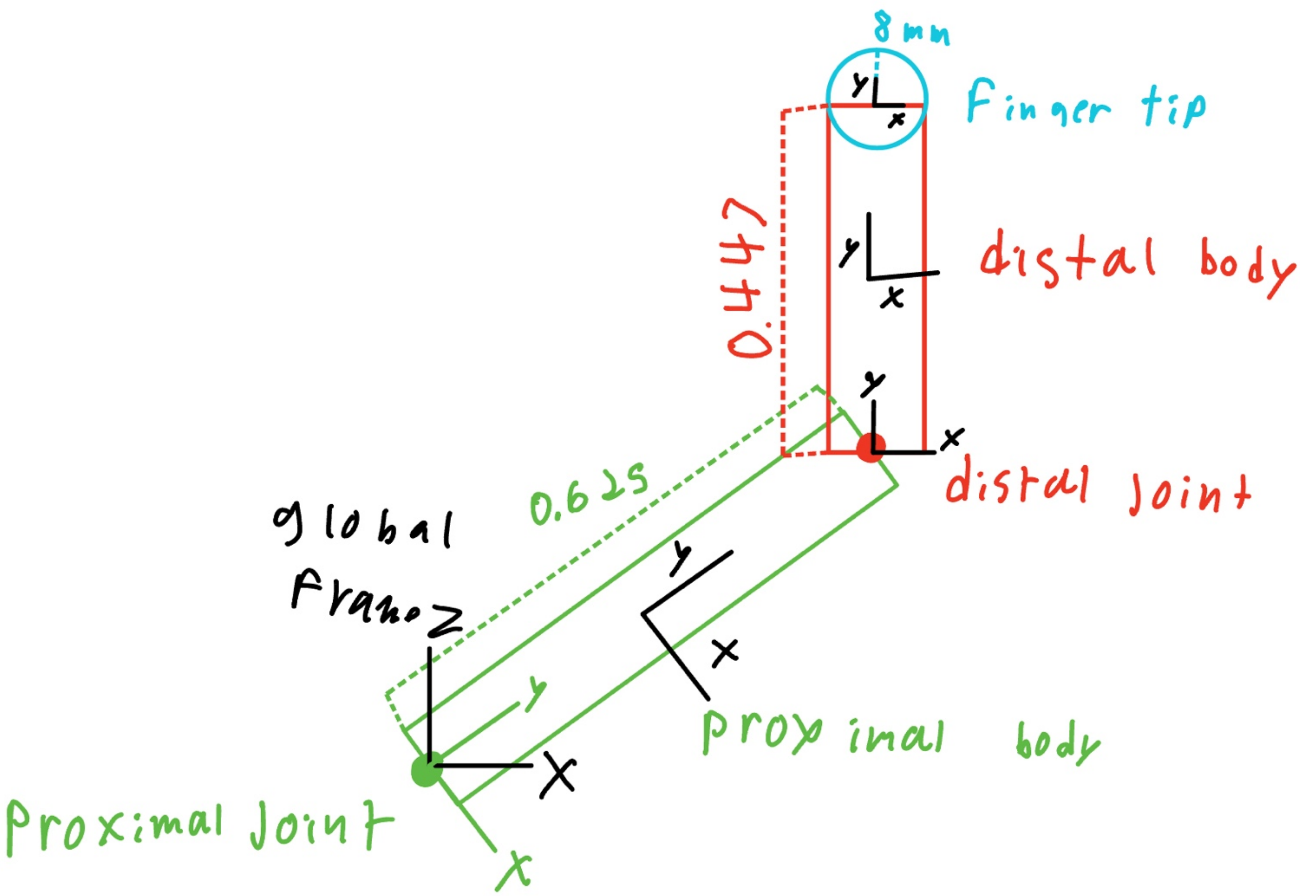
With all this implemented, I had a wonderful simulation with no energy in it, as the mechanism starts at rest. To add energy to the system, I had to compute the soft contact between the fingertip and the contact plane. First, I derived the phi function. This was just the distance from the center of the fingertip to the contact plane (as computed by the distance between the center point and the infinite place corresponding to the contact plane). I then subtracted the finger thickness from this value to get the distance from or penetration into the contact surface. I then computed the elastic contact between the fingertip and the contact surface, computing the x and y components of the contact, and multiplying this by an experimentally determined k. I selected k so that collision seemed realistic and looked right. Finally, I multiplied the contact Jacobian by these x and y contact forces to compute the energy added into the system.

The final modification I made was when to end contact. As I modeled the contact plane as an infinite plane, there wasn't technically an end to the contact. So, I made 2 version of my f function, one with contact and one without it. This was done by adding the force from contact, or removing the addition portion. To select which function to use, I checked if the edge of the finger was below or past the end of the contact plane. In both cases, the finger should move freely. So, the first time the finger breaks contact, I disable the contact portion of the simulation. I used disable contact checking after initial contact as there would be weird contact enable and disabling as the top of finger traversed close to the edge of the contact plane. So, I just disabled contact once contact was lost the first time. This looked roughly correct, so I left it there.

Future Work

Given more time, I'd model the tendon constraint of the finger. In the real fingers, the distal and proximal joint rotation is linked via a constant length tendon. I wasn't able to simulate this in time given the complexity of this constraint (it's contingent on the pulley size along with a few other factors), so I left simulating the tendon correctly as future work. Despite this limitation, I'm still pretty happy with how the simulation looks.

Note: While it looks like there's only 1 joint at the bottom of the finger, this is due to the stiffness of the distal joint. There is a minor deflection in this joint (notice the difference between the two y arrows at the top of the finger as the mechanism is pushed back) so, there are 2 joints modeled here, but 1 of the joints does most of the movement.


8mm
Finger tip
0.440
distal body
distal joint
0.625
global frame
Z
X
proximal body
Proximal Joint
x
y