CMU 16-264 Humanoids IK Optimization Writeup

Angel Macias

Writing the MATLAB Code

I first took a while just reading what I was given and trying to understand how initial parameters were set up and what they did and how they formed the Arm in its resting position. Then I worked on making a lot of the variables global for better accessibility, such as the p\_d and initial hand position along with some new globals like pre\_grasp\_d (MY desired pre\_grasp position). I just had the main program just run maincon and draw the resulting answer which led the hand straight to the objective, which was pretty simple.

Then I created a way to give a pre\_grasp position by aligning a vector to the x\_d angle and subtracting/adding(based on quadrant) a small offset so that the fingers don’t run into the object.

After the pregrasp I created a trajectory function that uses linear interpolation to give x\_d points for the hand to go to on the way to the pre\_grasp destination with a relatively smooth motion. After reaching the pregrasp position the hand waits a bit then moves forward to finish the objective. I’ll just briefly add here that my though process for this part was a bit odd in that I didn’t think of linear interpolation at first, but instead thought of having the joints move directly to pre\_grasp position at a given rate by computing each individual part’s distance to the goal then divide by an artificial rate to give how many increments to move in. That initial method was going to be rather tedious so I decided to use the linear interpolation method I learned in class since it’s so much more elegant.

The criterion function is basically the same as what was given to us for the previous example files, which makes sure that the hand is aligned and in the same are. However, I added a small bias of trying to keep the joint angles the same as the leading angle.

What worked rather well getting the arm to the pre-grasp position smoothly then to grasp. What doesn’t seem to work too well is trying to get the arm to other quadrants, though I suppose that’s due to the lb and ub constraints of the angles that force the arm to stay in a general area. What’s rather difficult to get to work is to check if the angle of the target is unreachable without trying to get there first.

1)

First ignoring the angle position, we know that the target can’t be reached when the target is further than the arm can reach fully extended. Factoring the angle in, if the arm has to bend too much to reach it, then clearly it can’t be reached. Meaning that as you reach the object, if each angle is reaching either an upper or lower bound, then the object can’t be reached. The optimizer being stuck at a local minimum means that the object can’t be reached, but we didn’t realize beforehand and didn’t stop the optimizer from trying to reach an unreachable object.

2)

We can first make a max circle with the radius being the lengths of the arms combined. The inner radius would be the joints all going to the max upper limit and then measuring how close the tip is to the origin. Then the distance from the origin is the radius for the inner circle, creating a donut type range for the (x, y) of the tip of the arm. The angle is a much more difficult thing to figure out. I suppose if we discretize the area bit, we can just have the arm try out segments of the area with various angles in that segment, and if the arm can’t reach it, just add it to a list of unreachable (x,y,angle) points. It seems a bit naïve, however I think it may be the way that Humans do things, as in, we try to reach an object and if we have to grab the object with our hands in a certain angle, we try to place our hands in that angle until it works or we just give up.

3)

The arm can reach a bit into the 4th quadrant now, though not completely. Objects that are closer to the +x-axis with an angle range of around .4< ᶿ < .6 (rads) would probably accessible now due to this range change. I’m not sure if I can rigorously prove the mathematics for this, considering how difficult it seems to be based on what I’ve seen and read online, but essentially it looks like a chunk of the 1st Quadrant has been made in accessible and placed into the 4th quadrant for accessibility.