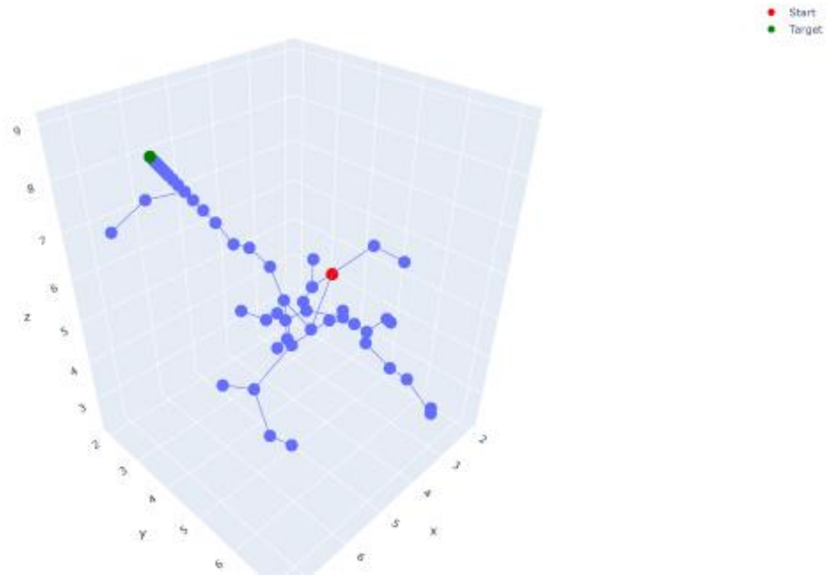
Project 6 - Report

CS 3630, Spring 22
mim30@gatech.edu
Moungsung Im

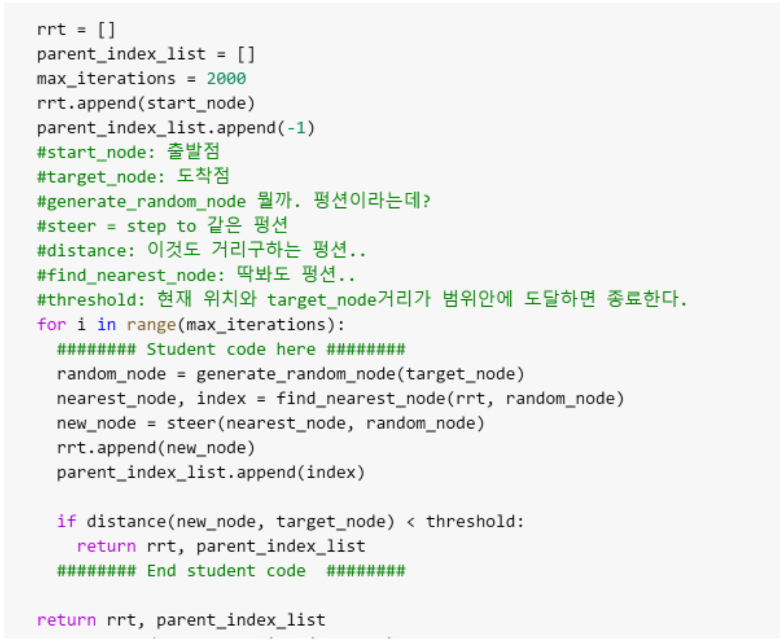
Q1.1 Visualization of the RRT tree (1 point)

Paste a screenshot of the visualization of the RRT tree in your report.



Q1.2 RRT Implementation (1 point)

Paste a screenshot of your implementation of the run_rrt function.



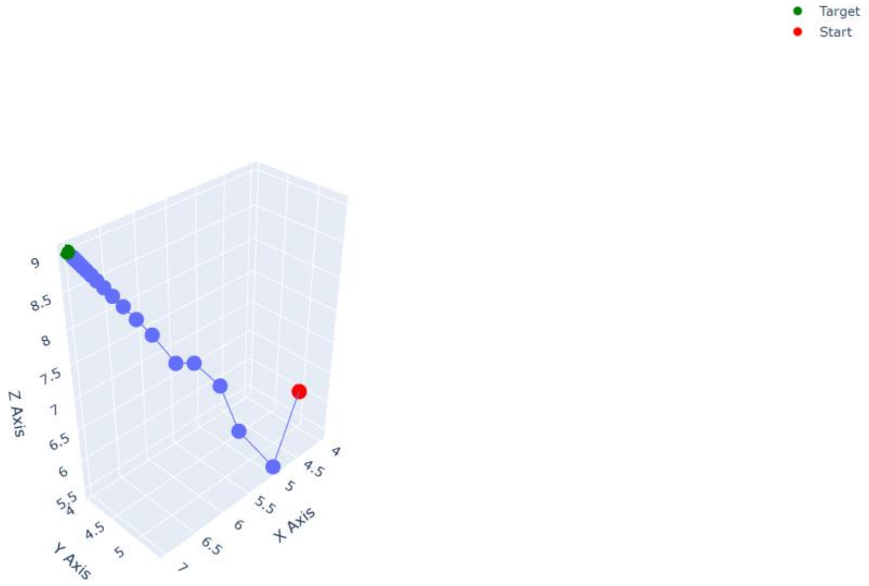
```
rrt = []
parent_index_list = []
max_iterations = 2000
rrt.append(start_node)
parent_index_list.append(-1)
#start_node: 출발점
#target_node: 도착점
#generate_random_node 뭘까. 펑션이라는데?
#steer = step to 같은 펑션
#distance: 이것도 거리구하는 펑션..
#find_nearest_node: 딱봐도 펑션..
#threshold: 현재 위치와 target_node거리가 범위안에 도달하면 종료한다.
for i in range(max_iterations):
  ######## Student code here ########
  random_node = generate_random_node(target_node)
  nearest_node, index = find_nearest_node(rrt, random_node)
  new_node = steer(nearest_node, random_node)
  rrt.append(new_node)
  parent_index_list.append(index)

  if distance(new_node, target_node) < threshold:
    return rrt, parent_index_list
  ######## End student code  ########

return rrt, parent_index_list
```

Q1.3 Visualization of the final path (1 point)

Paste a screenshot of the visualization of the final path obtained in your report.



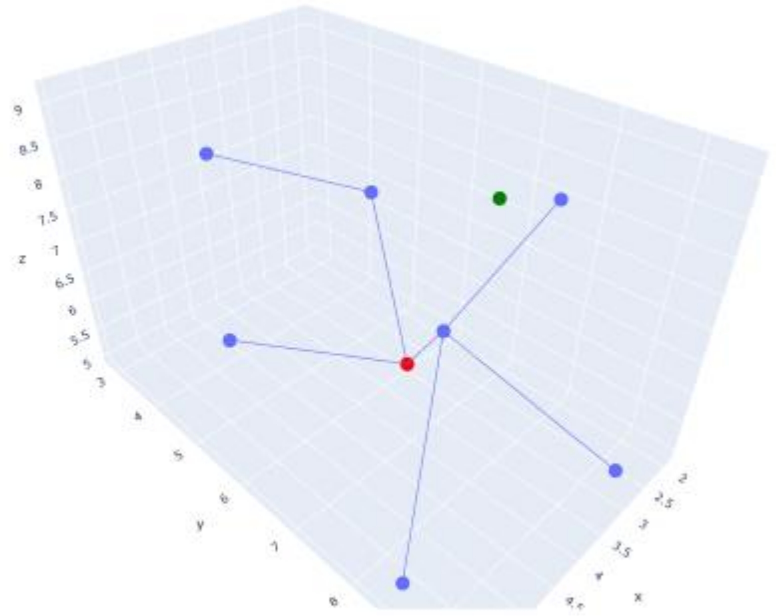
Q1.4 Nodes are close to one another! (2 points)

If you look closely at the tree and the path, you'll notice that as we reach close to the goal (the green node), the nodes are extremely close to one another! We begin to take really small steps and we converge very slowly towards it. Can you explain why this is happening?

As the drone approaches to the target, in the steer_navie, target-parent gets less. Therefore, it returns a value which is little different from the parent. Then in the rrt, there is little advance in the path of drone.

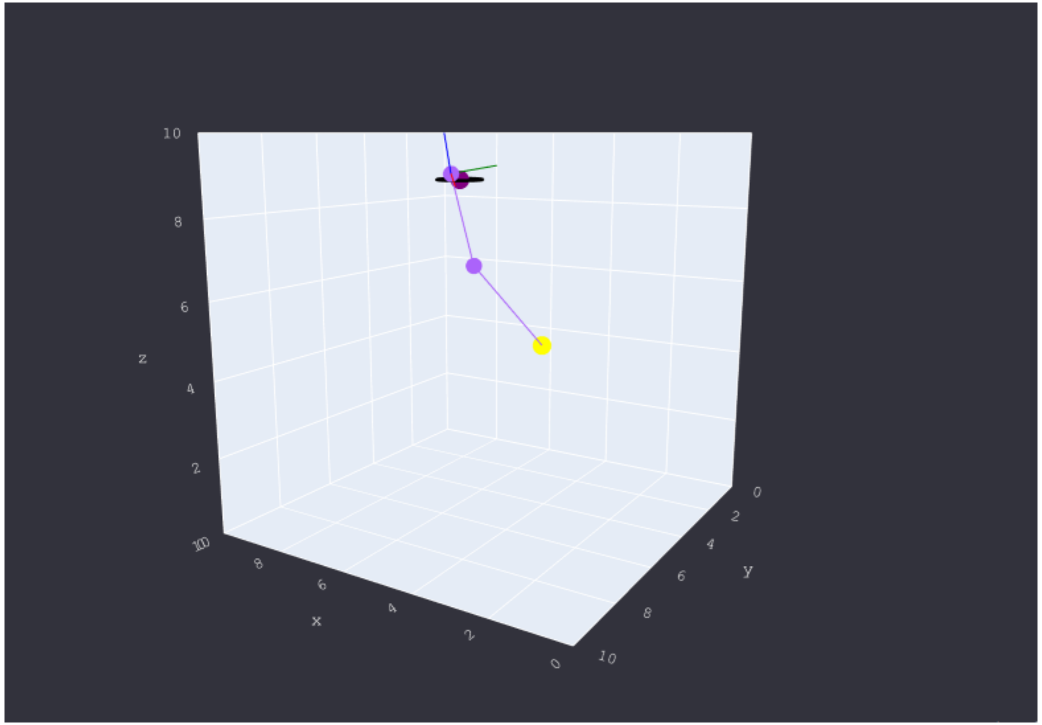
Q2.1 Visualization of the RRT tree (2 points)

Paste a screenshot of your RRT tree visualization obtained after steering with a terminal velocity.



Q2.2 Visualization of the final path (2 points)

Paste a screenshot of your final path obtained, after steering with a terminal velocity.



Q2.3 Is steer() realistic? (2 points)

The “steer” function we coded up does not seem realistic in nature. What do you think are the two major limitations of our approach?

We ignored the gravitational force affecting the direction and magnitude of our applied thrust.

By assuming we could apply the thrust in the direction we want to steer in, we allowed arbitrary and instantaneous attitude changes at the time of steering.

Q3.1 Configuration 1 (5 points)

Configuration:

- Yaw: 90 degrees
- Pitch: 0 degrees
- Roll: 0 degrees
- Thrust: 10 Newtons

1. In which direction is the drone flying in the navigation frame?

[0 0 0] because terminal velocity is 0 0 0..

2. Is the drone flying upwards, downwards or maintaining level flight?

Maintaining level flight

3. In what direction is the thrust applied in the navigation frame?

[0 0 1]

4. What is the speed of the drone?

0m/s

5. What direction is the front of the drone facing in the navigation frame?

[1 0 0]

Q3.2 Configuration 2 (5 points)

Configuration:

- Yaw: 0 degrees
- Pitch: 45 degrees
- Roll: 45 degrees
- Thrust: 20 Newtons

1. In which direction is the drone flying in the navigation frame?

[.64 -.77 0]

2. Is the drone flying upwards, downwards or maintaining level flight?

Maintaining level flight

3. In what direction is the thrust applied in the navigation frame?

[.5 -.71 .5]

4. What is the speed of the drone?

23.83 m/s

5. What direction is the front of the drone facing in the navigation frame?

[.71 0 -.71]

Q3.3 Configuration 3 (5 points)

Configuration:

- Yaw: 25 degrees
- Pitch: 0 degrees
- Roll: 45 degrees
- Thrust: 15 Newtons

1. In which direction is the drone flying in the navigation frame?

[.55 -.81 .2]

2. Is the drone flying upwards, downwards or maintaining level flight?

Upwards

3. In what direction is the thrust applied in the navigation frame?

[.3 -.64 .71]

4. What is the speed of the drone?

18.6 m/s

5. What direction is the front of the drone facing in the navigation frame?

[.91 .42 0]

Q3.4 Configuration 4 (5 points)

Configuration:

- Yaw: 10 degrees
- Pitch: 45 degrees
- Roll: 30 degrees
- Thrust: 0 Newtons

1. In which direction is the drone flying in the navigation frame?

[0 0 -1]

2. Is the drone flying upwards, downwards or maintaining level flight?

downward

3. In what direction is the thrust applied in the navigation frame?

[.69 -.39 .61]

4. What is the speed of the drone?

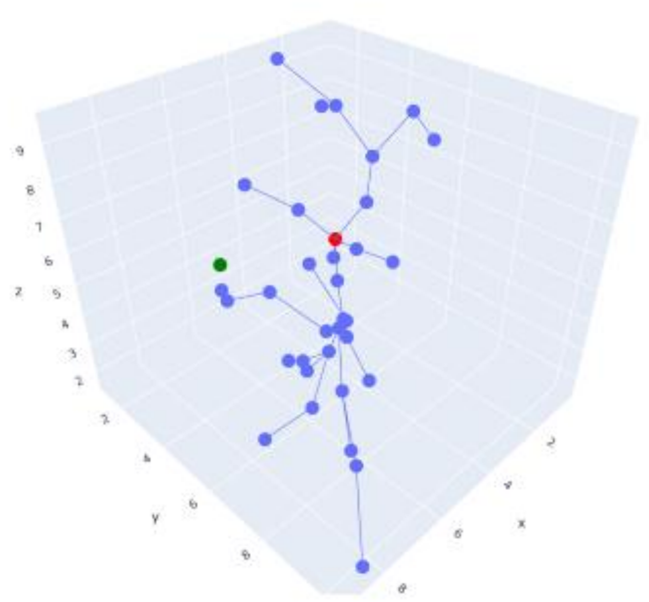
15.34 m/s

5. What direction is the front of the drone facing in the navigation frame?

[.7 .12 .71]

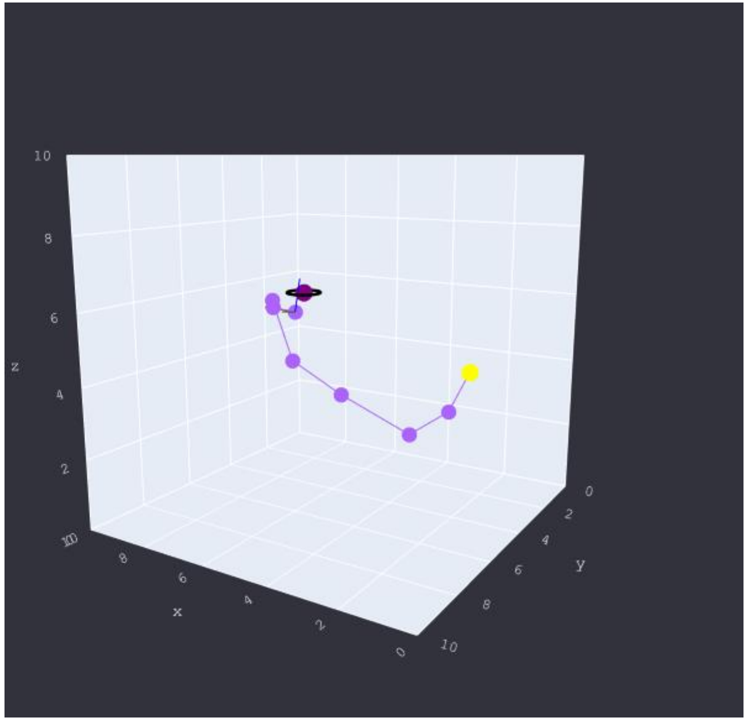
Q4.1 Visualization of the RRT tree (3 points)

Paste a screenshot of your RRT tree visualization obtained after implementing a more realistic version of the steer function.



Q4.2 Visualization of the final path (3 points)

Paste a screenshot of your final path obtained after implementing a more realistic version of the steer function.



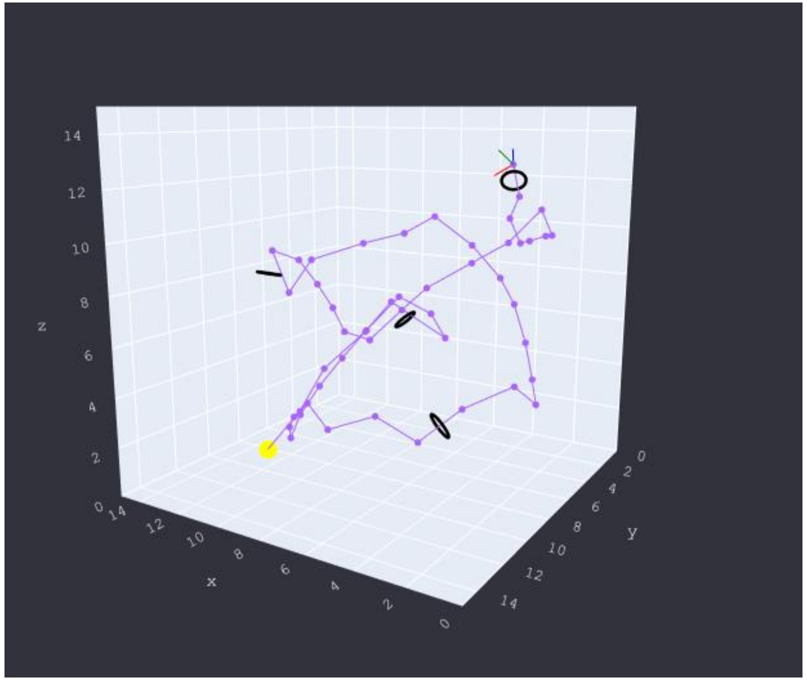
Q4.3 Fly faster? (3 points)

Open-ended: Do you think the drone can fly faster than our current implementation using these realistic dynamics? If yes, how? If no, why?

I think our drone obviously can move faster if we have realistic dynamics data with more data from the world, more accurate speed..etc. so that the drone doesn’t have to waste its power to go to unnecessary spot.

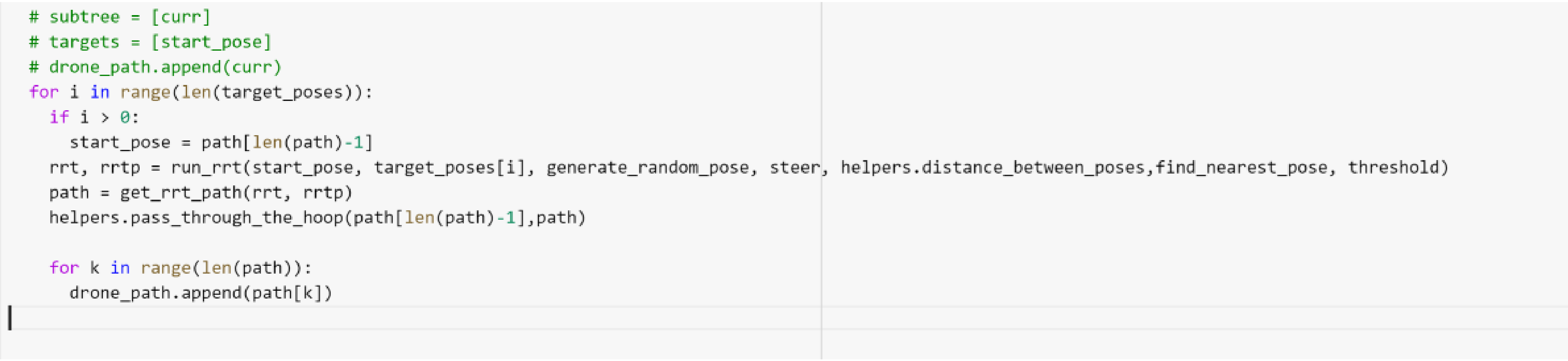
Q5.1 Drone trajectory visualization (3 points)

Paste a screenshot of your drone's trajectory at the GT 3630 Drone Racing Challenge, passing through all the hoops in the circuit.



Q5.2 Drone racing implementation (3 points)

Paste a screenshot of your implementation of the drone_racing_rrt function.



```
# subtree = [curr]
# targets = [start_pose]
# drone_path.append(curr)
for i in range(len(target_poses)):
  if i > 0:
    start_pose = path[len(path)-1]
  rrt, rrtp = run_rrt(start_pose, target_poses[i], generate_random_pose, steer, helpers.distance_between_poses,find_nearest_pose, threshold)
  path = get_rrt_path(rrt, rrtp)
  helpers.pass_through_the_hoop(path[len(path)-1],path)

  for k in range(len(path)):
    drone_path.append(path[k])
```

Q5.3 Drone trajectory optimization (3 points)

Open-ended: What do you think can be some ways to optimize the trajectory taken by the drone? Keep in mind that instantaneous rotation of the drone is limited to -10 to 10 degrees in yaw, pitch and roll.

We can calculate more nodes for rrt and calculate the best path from the huge rrt. When we make path with the huge rrt, we can find the most efficient and smooth path.

Q6 Feedback (1 point)

Please provide feedback on the coding portion of the project. How did it help your understanding of the material? Is there anything that you think could have been made more clear?

The coding materials are very helpful to understand the matrials (even more than the book) because as I struggle to find the right answer, I must try multiple attempts and figure out why a particular solution is necessary. As I always mention, I wish there are more test cases in the colab rather than in gradescope. Sometimes, althought I pass the test in colab, the answer isn’t right for the report.