Entry-level project

Songhao Huang

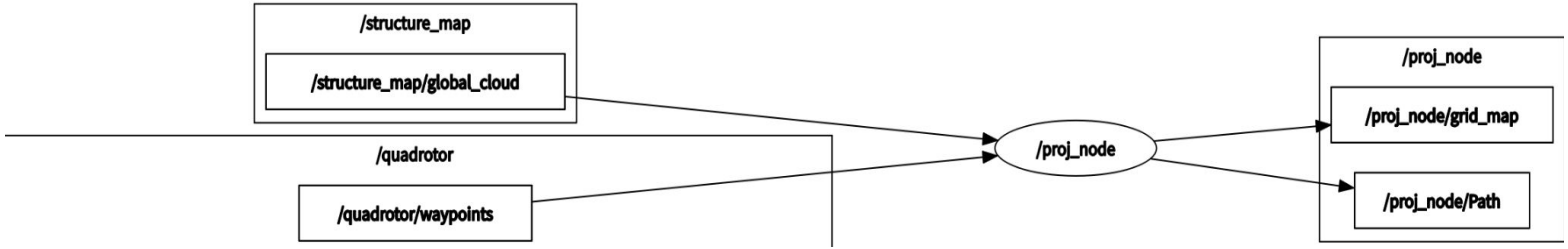
Reference:

1. Minimum Snap Trajectory Generation and Control for Quadrotors.

2. Planning_Dynamically_Feasible_Trajectories_for_Quadrotors_Using_Safe_Flight_Corridors_in_3D_Complex_Environments

3. Github kumar lab

Receive map



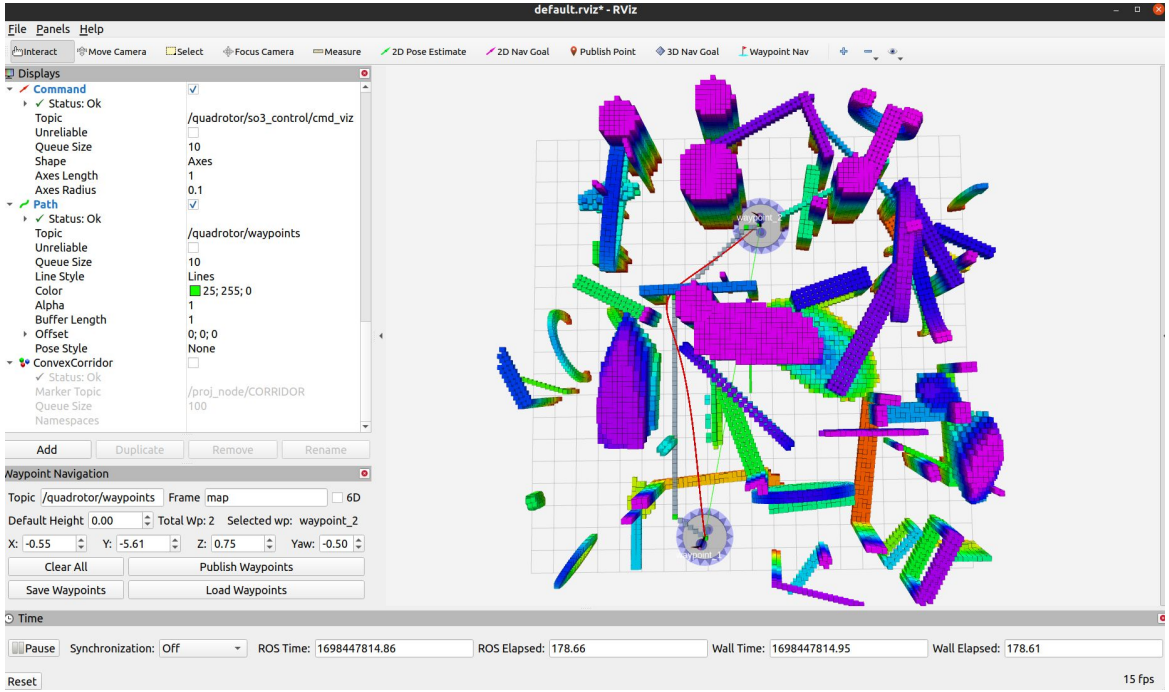
/proj_node:

Subscribe PCL map and

start point & end point

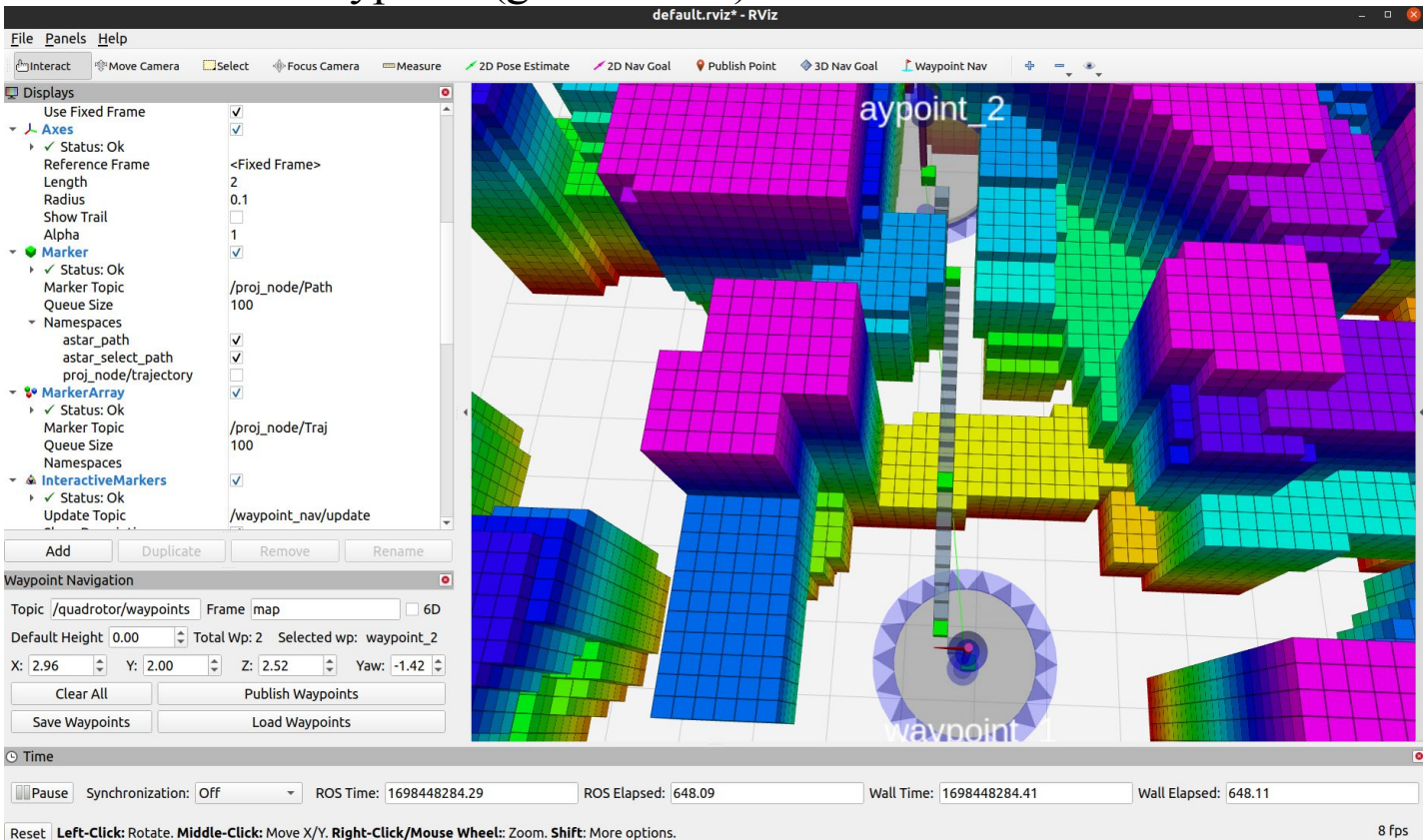
Publish 3D grid map and

path & trajectory visualization



Front-end path planning

A* search in 3D inflated grid map(transparent cubes) + Ramer–Douglas–Peucker algorithm to reduce the number of waypoints(green cubes)



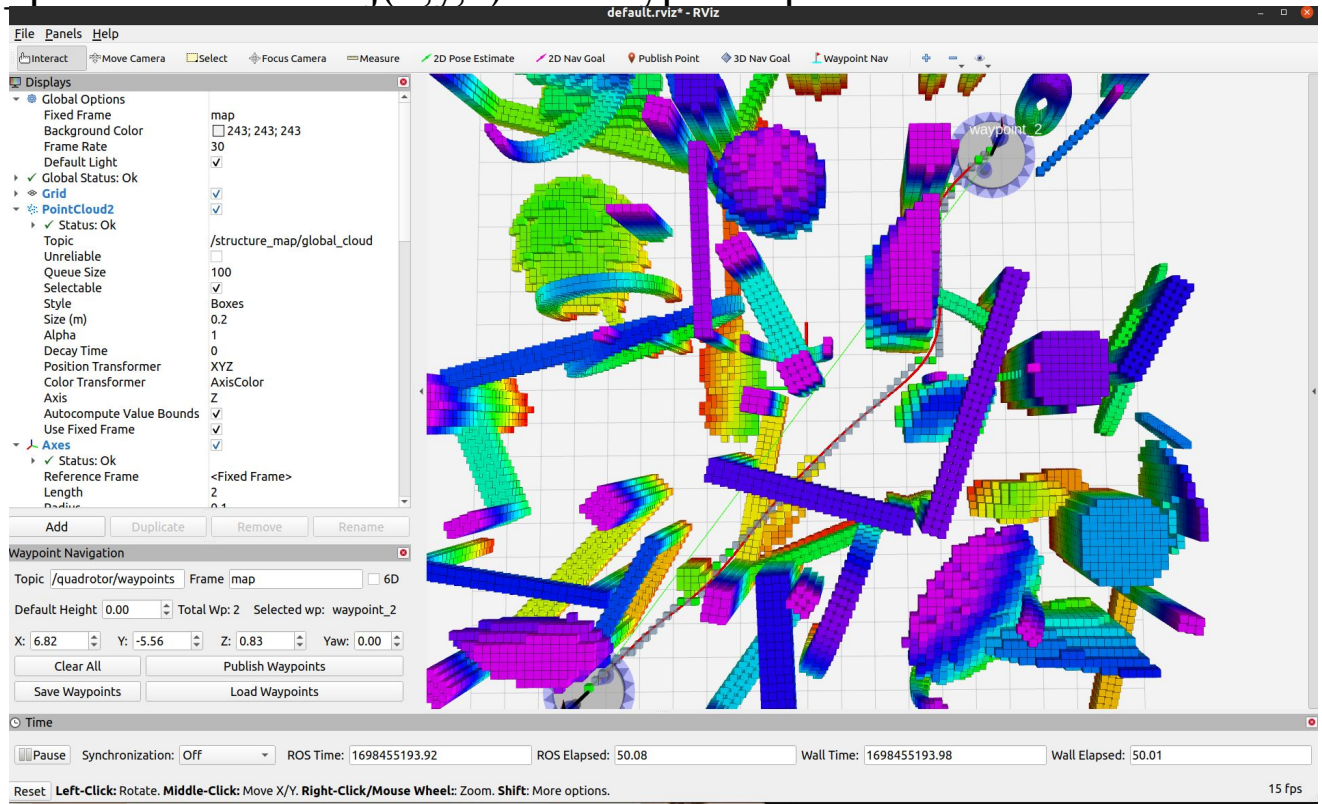
Back-end trajectory optimization

1. Trajectory: 7th order polynomial
2. Objective function: minimum the snap
3. Constraints: waypoints position constraints + continuous constraints + corridor position constraints
4. Time allocation: trapezoid time allocation for whole trajectory + optimize time allocation iteratively to limit max v,a,j.
5. Corridor: dense corridor / convex corridor

Back-end trajectory optimization-dense corridor

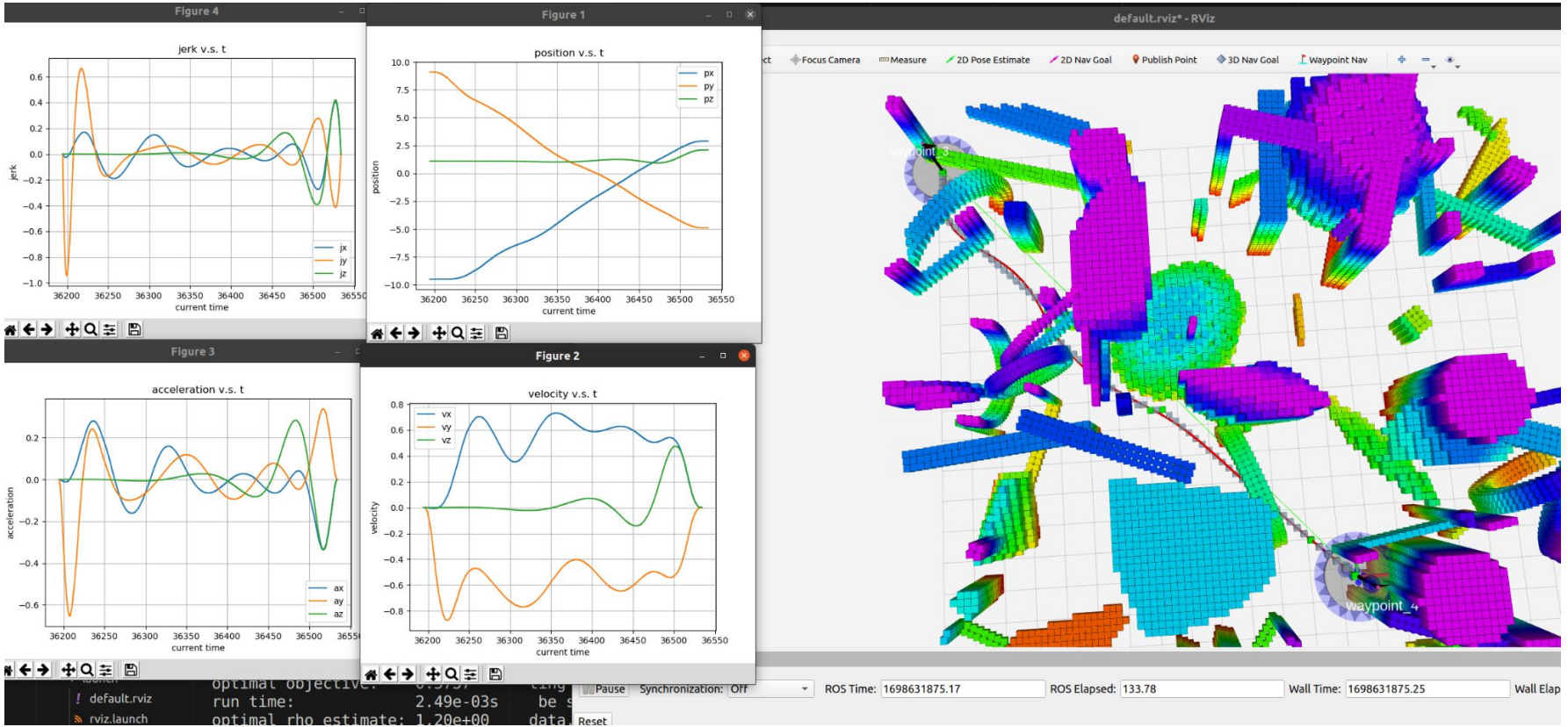
Sample more waypoints between any two waypoints.

waypoint_position - r <= traj(x,y,z) <= waypoint_position + r



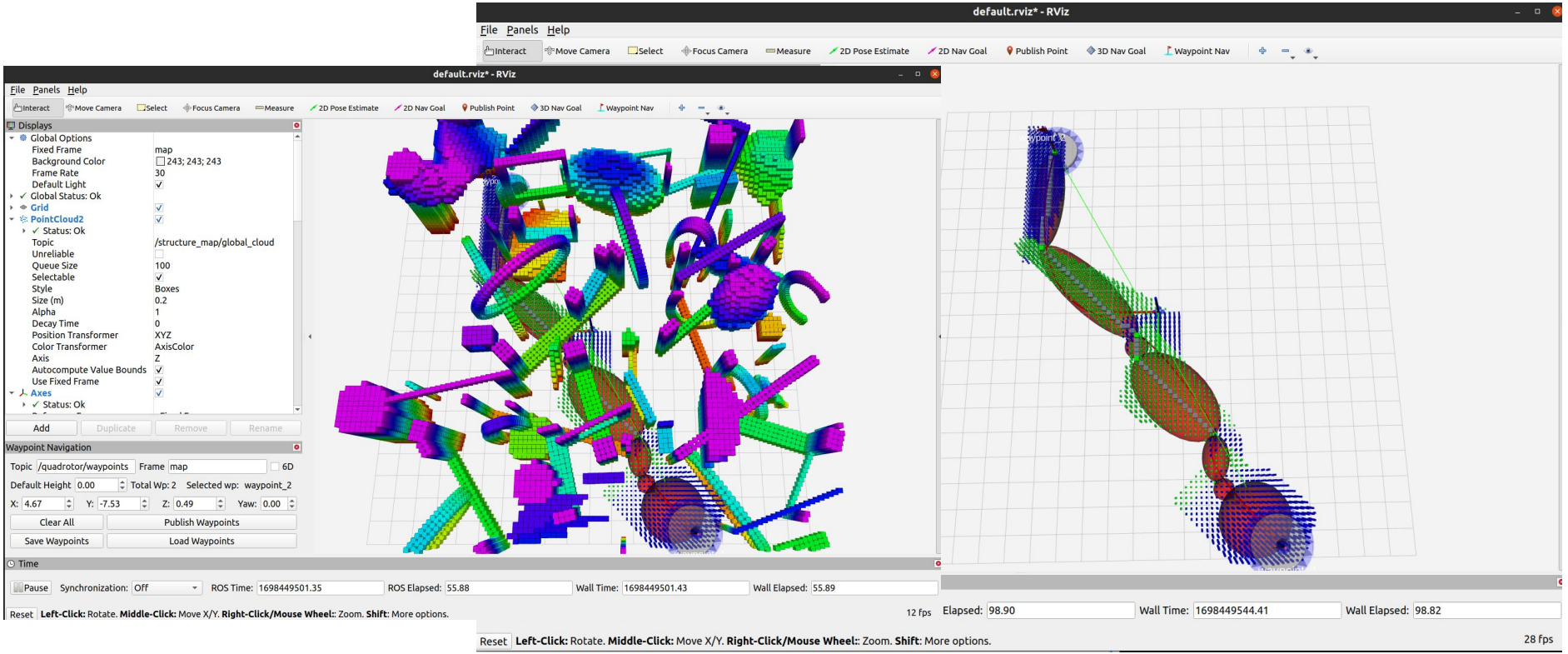
Back-end trajectory optimization-dense corridor

Performance(slow v, slow a, slow jerk): Average time: frontend: ~30ms, backend: ~80ms



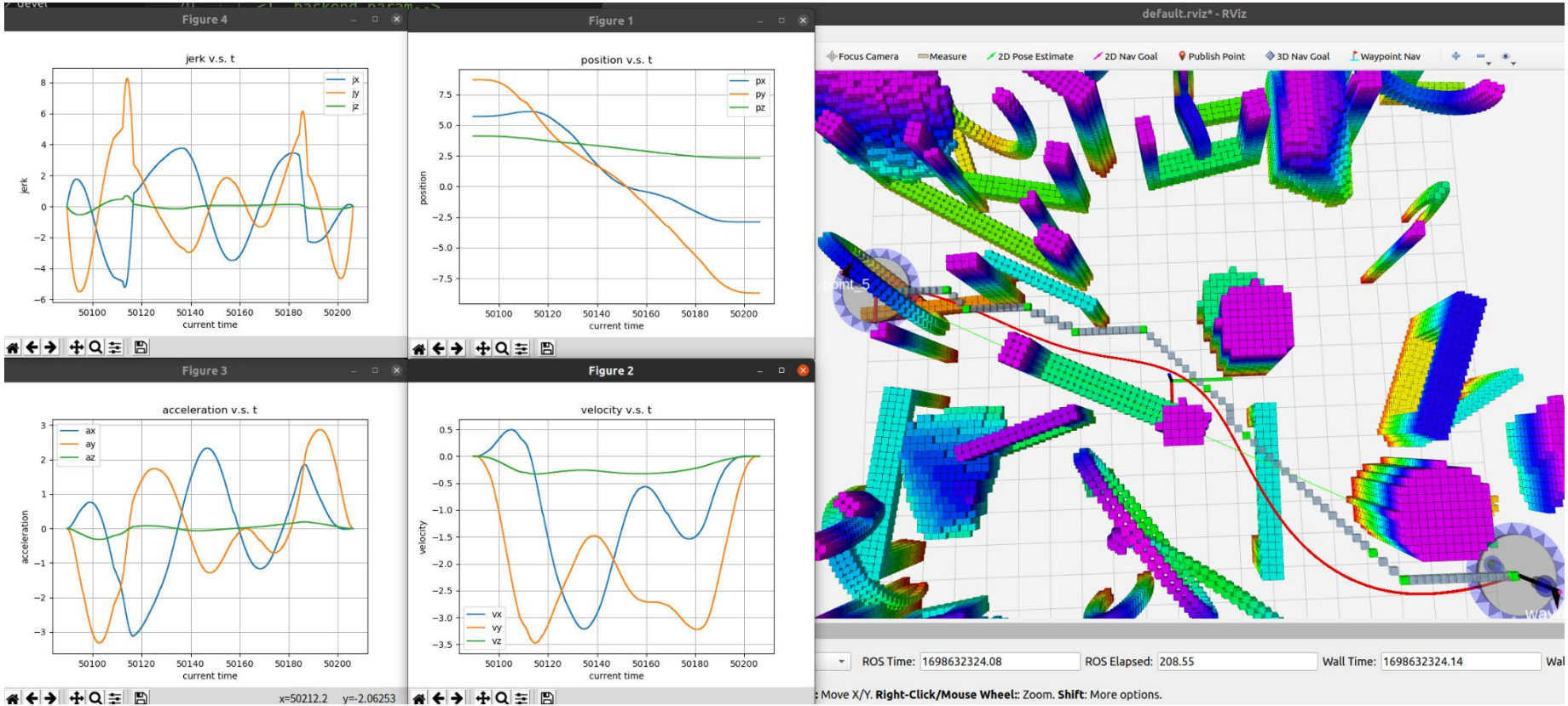
Back-end trajectory optimization-convex corridor

Construct 3D convex corridor in 3D space to make sure fully use of available space.



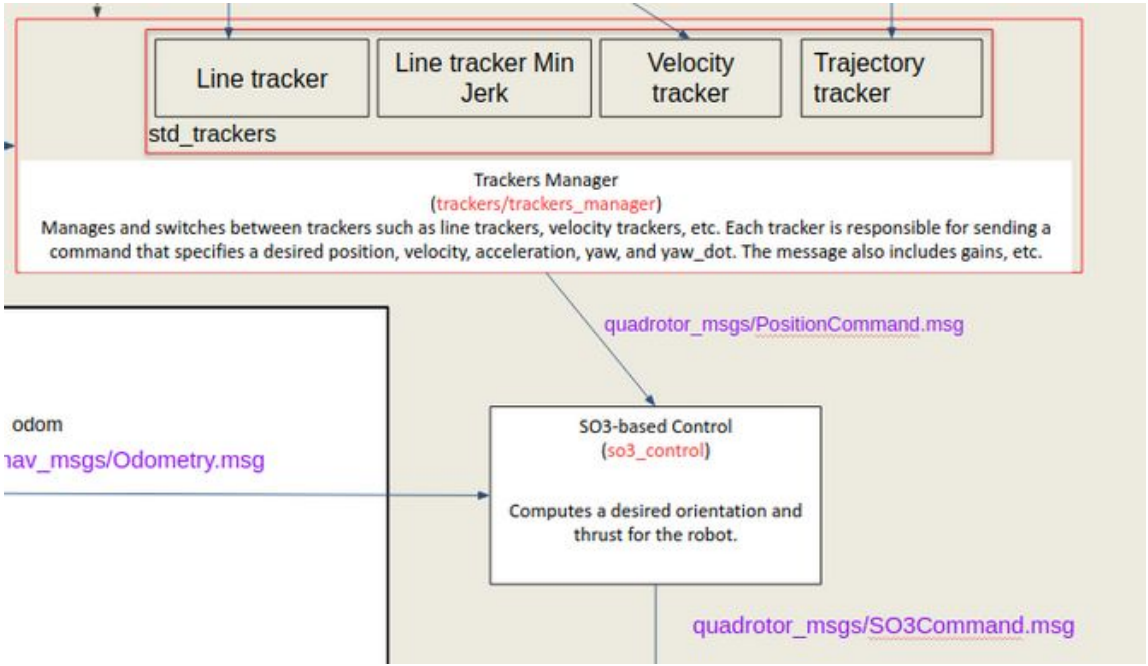
Back-end trajectory optimization-convex corridor

Performance(faster v, faster a, faster jerk): Average time: frontend: ~30ms, backend: ~300ms

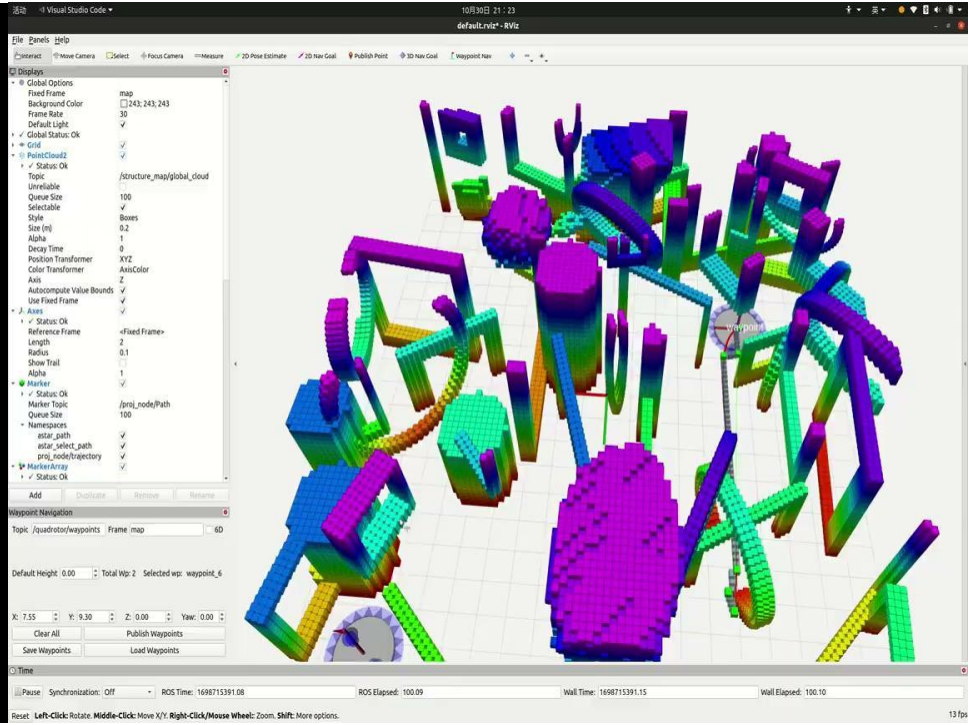


so3_controller

Use “Trajectory tracker”, modify the “update” function with my own trajectory calculated by my coefficients.



Run the result



Dense Corridor demo

Convex Corridor demo