RBE 550: Motion Planning
Transmission

Environment Modeling:

For this assignment I have used MATLAB for the simulation in a continuous space of the environment with all the components of the gearbox. The dimensions are taken with the reference of the given CAD file in the assignment and then scaled down.

The primary shaft and the gearbox are modeled as collision cylinders and collision boxes, not to the exact geometry. Collision checking is between the cylinders and boxes. The side walls of the case are not involved in collision hence it is not required in the simulation.

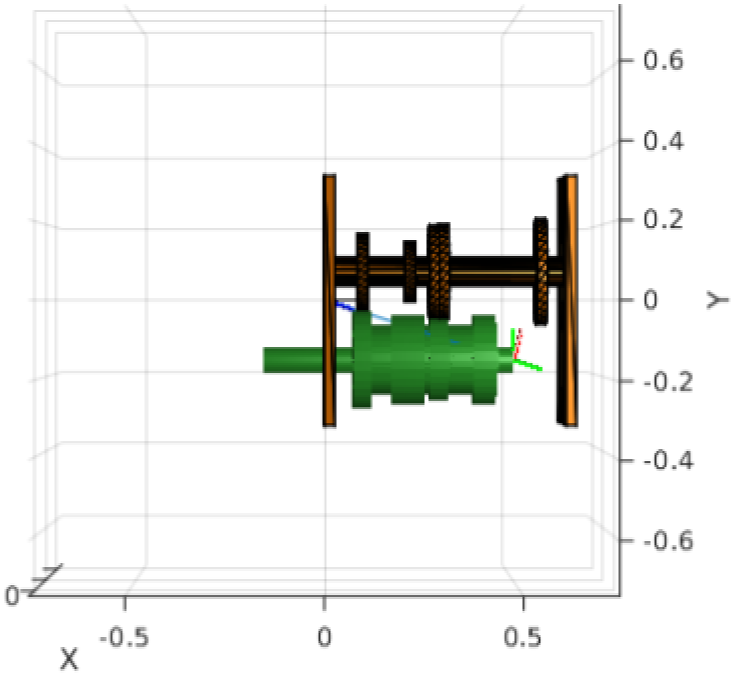


Fig: Environment made with collision. The green is the mainshaft.

Robot Modeling:

The sampling based planners like RRT are generally used for manipulators in their configuration space. I have designed a 3 DOF robot manipulator RPR (Revolute - Prismatic - Revolute). Revolute means a rotatory joint and Prismatic means a linear joint.
The DH parameters are created in MATLAB with the Robotics Toolkit. The robot is modeled such that the mainshaft is the end effector of the robot.

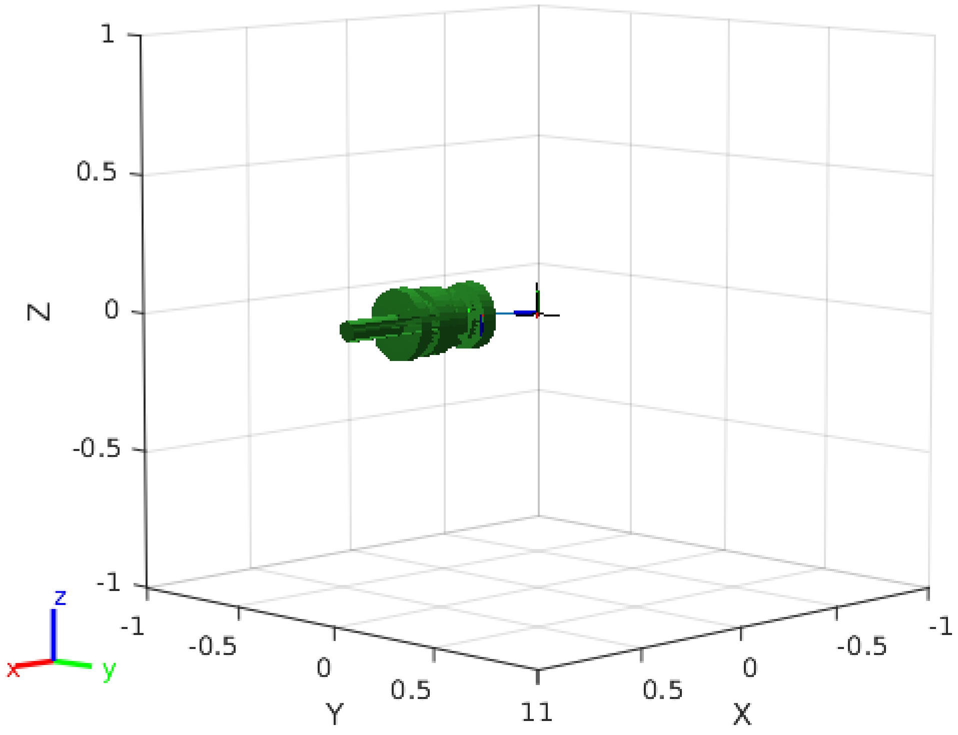
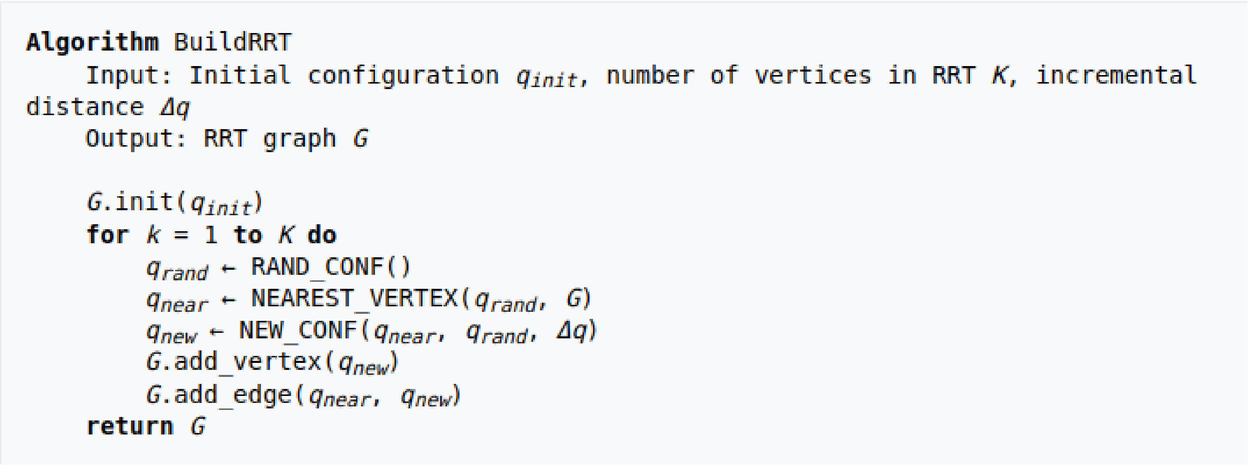


Fig: Robot manipulator with collision bodies at the end effector

Planning Algorithm:

Rapidly-exploring Random Tree Algorithm: RRT is an algorithm designed to efficiently search nonconvex, high - dimensional spaces by randomly building a space-filling tree. The tree is constructed incrementally from samples drawn randomly from the search and is inherently biased to grow towards a large unsearched area of the problem.

RRT’s can be viewed as a technique to generate open-loop trajectories for nonlinear systems with state constraints.



```
Algorithm BuildRRT
    Input: Initial configuration $q_{init}$, number of vertices in RRT $K$, incremental distance $\Delta q$
    Output: RRT graph $G$

    $G$.init($q_{init}$)
    for $k$ = 1 to $K$ do
        $q_{rand}$ ← RAND_CONF()
        $q_{near}$ ← NEAREST_VERTEX($q_{rand}$, $G$)
        $q_{new}$ ← NEW_CONF($q_{near}$, $q_{rand}$, $\Delta q$)
        $G$.add_vertex($q_{new}$)
        $G$.add_edge($q_{near}$, $q_{new}$)
    return $G$
```

Fig: Algo RRT

Results:

The results for the path planning are shown below. The video has been attached with the document.

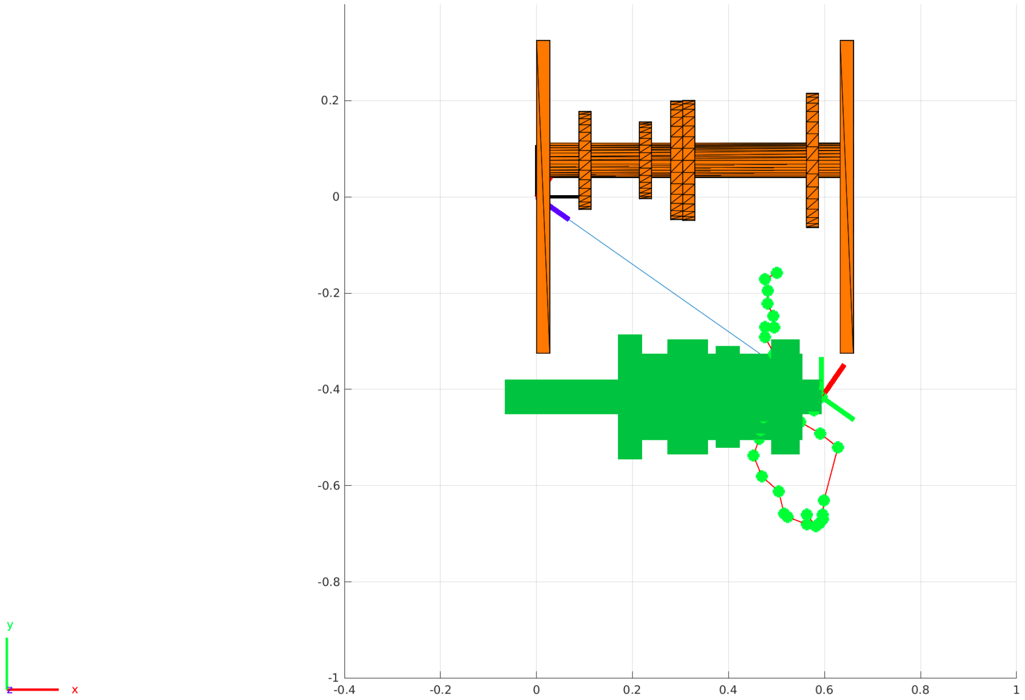


Fig: The path followed by the mainshaft

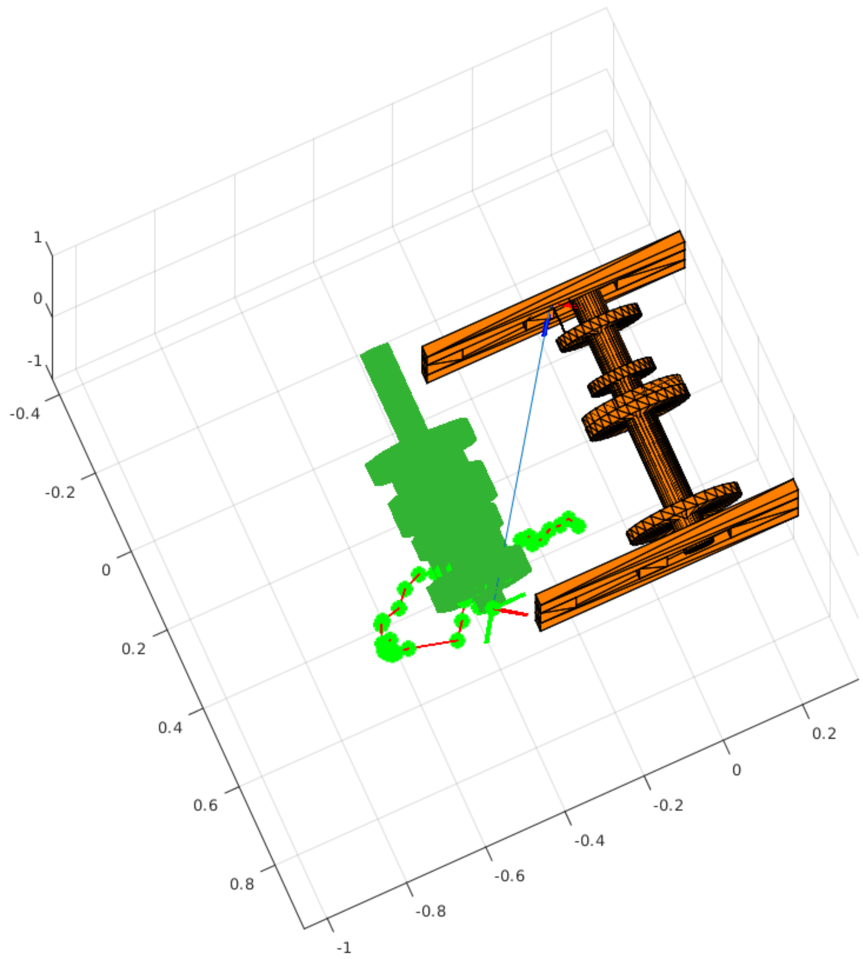


Fig: RRT path