Transmission

Implementation

Environment Modelling

Using MATLAB and collision meshes, I simulated the continuous space of the environment with all the components of the gearbox as shown in Fig. 1. All dimensions are taken from the CAD files given and are scaled down.

The primary shaft and the gearbox casing are modelled as collision cylinders and collision boxes, not to the exact geometry. Collision checking is performed between these cylinders and between the cylinder and the boxes. The side walls of the casing are not shown as they do not contribute to the planning algorithm and are removed to show the movement of the mainshaft.

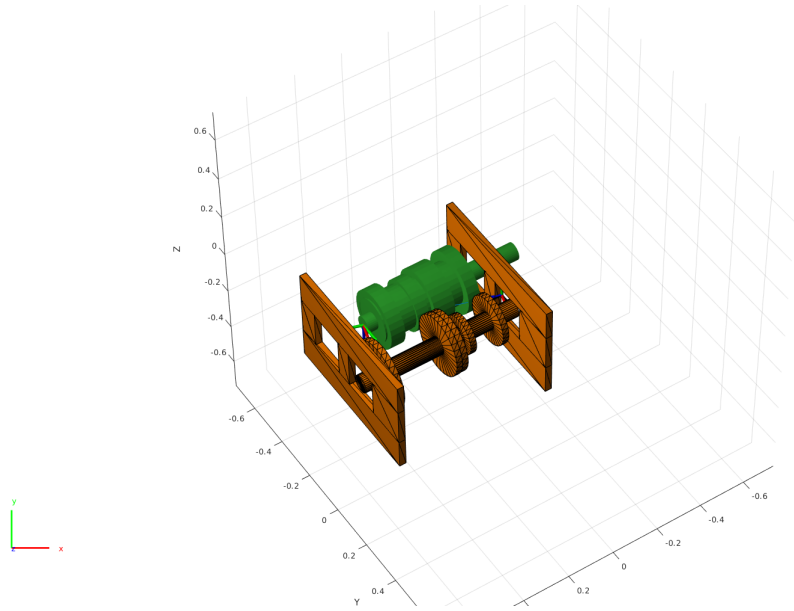

Figure 1: Environment made with collision meshes (boxes and cylinders). In green - mainshaft

Robot Modelling

Since sampling based planners like RRT are generally used for manipulators in their configuration space, I designed a 3-DOF (Revolute-Prismatic-Revolute) robot manipulator using DH parameters and created a rigidbodytree in MATLAB with the Robotics Toolkit [1]. The robot is modelled such that the mainshaft is the endeffector of the robot as shown in Fig 2.

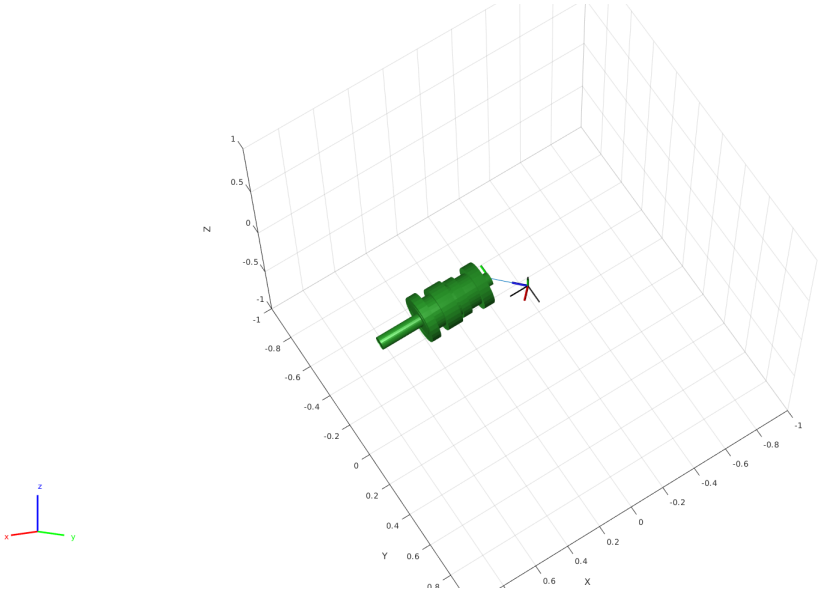

Figure 2: Final robot manipulator with collision bodies at the end effector

Planning Algorithms

RRT Algorithm

The RRT is a pretty straightforward algorithm. Points are randomly generated in the joints space of the manipulator and the it is checked to see if it is colliding with the environment. Pseudo code for the RRT algorithm is shown in Fig 3. Here the collision checker only checks for collisions between the end effector and the environment and ignores self collisions as the manipulator is only serving the purpose of finding a path for the mainshaft.

RRT Pseudo Code

```
Qgoal //region that identifies success
Counter = 0 //keeps track of iterations
lim = n //number of iterations algorithm should run for
G(V,E) //Graph containing edges and vertices, initialized as empty
While counter < lim:
    Xnew  = RandomPosition()
    if IsInObstacle(Xnew) == True:
        continue
    Xnearest = Nearest(G(V,E),Xnew) //find nearest vertex
    Link = Chain(Xnew,Xnearest)
    G.append(Link)
    if Xnew in Qgoal:
        Return G
Return G
```

Figure 3: Pseudo code for RRT algorithm

Results

The results for the planned path are shown in the figures 4 & 6. The video containing the animation is attached in the submission on canvas.

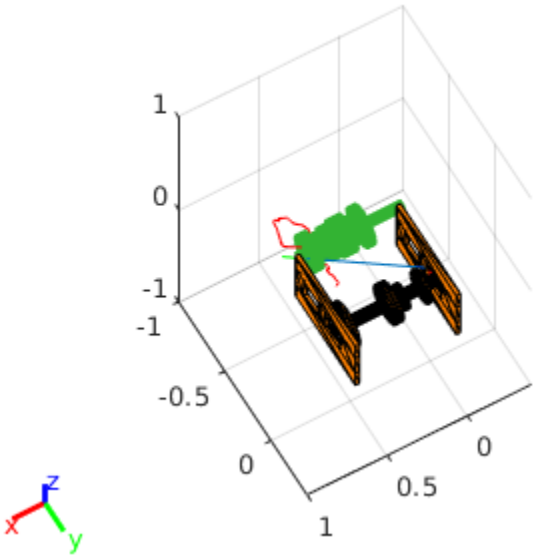

Figure 4: Final planned path of the mainshaft

The entire random search with end effector points is projected in 2D and shown in Fig 5

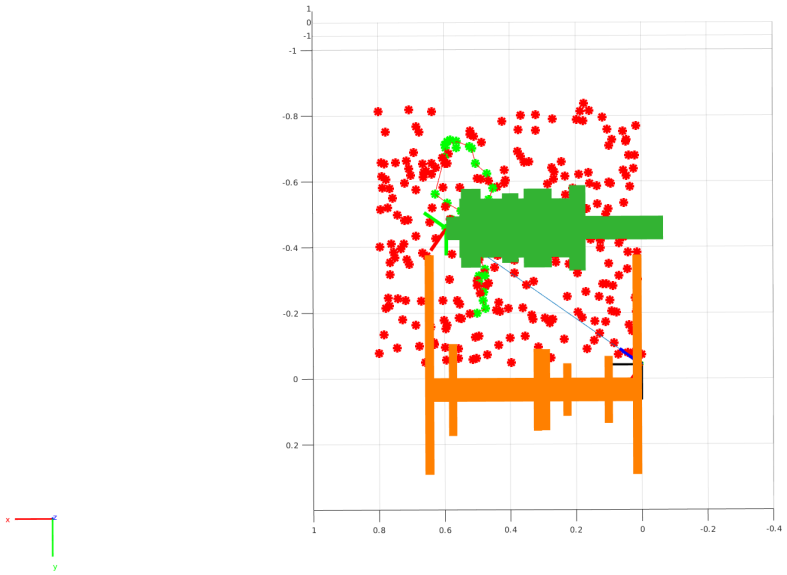

Figure 5: Complete Random end effector positions

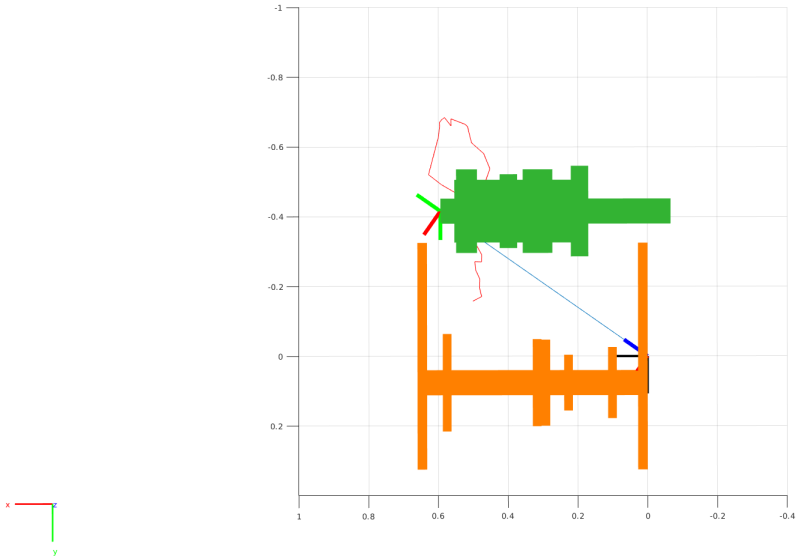

Figure 6: Final Planned Path

And finally the ugliest vehicle with SM-465 transmission in Fig 7

Figure 7: (very) Ugly vehicle with SM-465

Discussion

In this assignment, I implemented the RRT algorithm with collision checking to remove the mainshaft from the gearbox casing. Also, the mainshaft was modelled as the end effector of the robot. The environment is modelled as collision meshes and collision bodies. The manipulator serves the purpose of finding the path for the mainshaft only.

References

[1] MATLAB Robotics System Toolbox

[2] Robotic Path Planning: RRT and RRT*