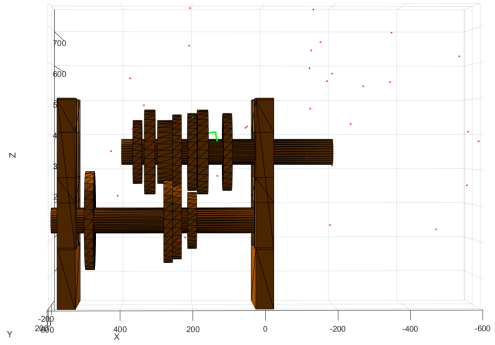
Compiled By:
Krutarth Trivedi (ktrivedi@wpi.edu)
MS Robotics,
Worcester Polytechnic Institute.

❖ Problem Statement:

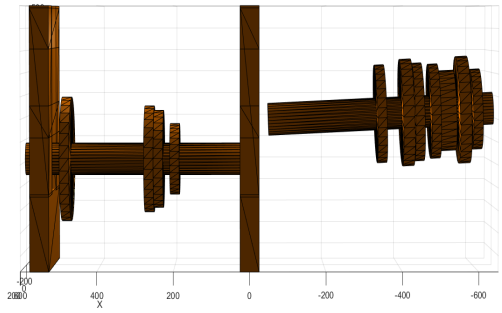
The goal at this step in the rebuild process is to remove the main shaft from the case. With the bearings uninstalled, they can be carefully lifted out. During removal, the main shaft must not make contact with the countershaft, or the case. The simulation should start with the transmission in the initial assembled state, and end with the main shaft completely clear of the case.

❖ Methodology:

- I used an RRT* algorithm to design a motion planner to transport the main shaft assembly in three-dimensional space.
- The planner takes the initial and goal states as inputs, creates a tree in full configuration space, and finds a solution path.



(Initial State)



(Final State)

❖ Environment:

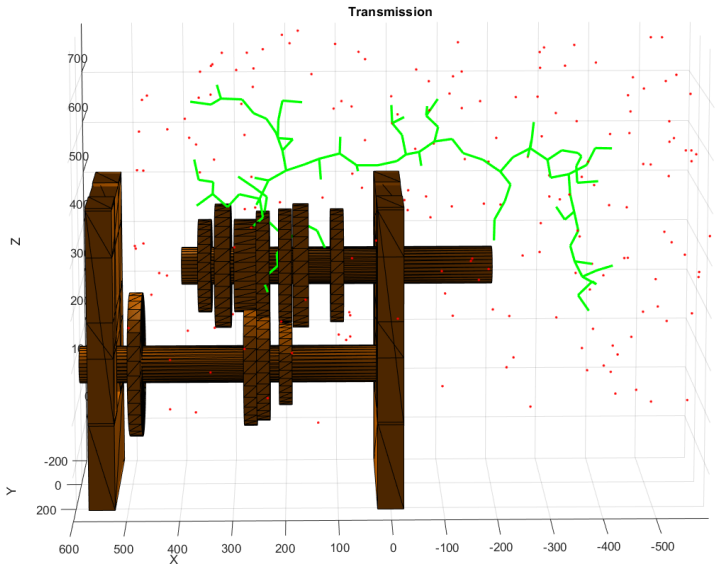
- I created the SM-465 transmission model along with the main shaft and countershaft based on the given dimensions in MATLAB.
- For designing the case, I used collisionBox and for both the shafts, I used collisionCylinder.

❖ RRT* Algorithm with the collision checker:

- Once the environment is designed, the next was to create a motion planner that can find a possible path to take the main shaft out of the transmission.
- For the same, I have used a variant of the Rapidly Exploring Random Tree(RRT) algorithm named RRT* - RRT* is an optimized version of RRT.
- First, RRT* records the distance each vertex has traveled relative to its parent vertex. This is referred to as the cost()of the vertex. After the closest node is found in the graph, a neighborhood of vertices in a fixed radius from the new node is examined. If a node with a cheaper cost() than the proximal node is found, the cheaper node replaces the proximal node.
- After a vertex has been connected to the cheapest neighbor, the neighbors are again examined. Neighbors are checked if being rewired to the newly added vertex will make their cost decrease. If the cost does indeed decrease, the neighbor is rewired to the newly added vertex while considering the collision checking.
- The collision checking is performed as a part of the obstacle avoidance algorithm. Obstacle avoidance is checked for the following tasks: when a node is placed, when a node is connected to its neighbor, and for each node that is to be rewired.
- This process continues till the planner finds an optimal path to reach the pre-defined goal state.

❖ RRT Figure:

- The figure below depicts the tree generated by the RRT* algorithm to reach the goal position.



❖ The path generated By the motion planner:

- The figure below depicts the path generated by the RRT* algorithm while considering obstacle avoidance.