Motion Planning for Transmission Mainshaft Extraction Using RRT in SE(3)

Mohamed Eljahmi

Abstract—**This report describes the process of planning a collision-free motion that removes the mainshaft from inside a transmission case and moves it to a target pose outside the case. The shaft is treated as a rigid body with six degrees of freedom, so its motion is planned in the full SE(3) space of positions and orientations. The geometry of the case and shafts was taken from the CAD models provided with the assignment and converted into simplified shapes for collision checking. A goal-biased Rapidly-Exploring Random Tree (RRT) was implemented to explore the six-dimensional space and search for a feasible path. The planner uses incremental motion steps and collision tests to expand the tree while avoiding the case walls and the fixed countershaft. The results include the final path of the shaft, a visualization of the RRT tree, and an animation showing the motion. All outputs are reproducible using the scripts included with the submission.**

I. Introduction

The goal of this assignment is to compute a collision-free motion that removes the mainshaft from inside the SM-465 transmission case and places it safely outside the housing. Because the mainshaft is a rigid body that must translate and rotate within a confined 3-D volume, its configuration is represented in the full six-degree-of-freedom space SE(3), consisting of position (x,y,z) and orientation (roll,pitch,yaw).

The extracted geometry was taken from the OpenSCAD models provided with the assignment, and a simplified collision model was built using cylindrical and capsule-shaped components. A Rapidly-Exploring Random Tree (RRT) was implemented to explore the SE(3) configuration space and identify a feasible path that avoids contact with the case and the countershaft.

The final solution includes the 3-D path of the mainshaft, a visualization of the RRT exploration tree, and an animation (or keyframe sequence) showing the complete extraction motion.

II. CAD Models And Geometry Extraction

The geometry used in this assignment comes from the OpenSCAD models supplied with the homework. These files describe the transmission housing, the mainshaft, the countershaft, and the assembled transmission. To understand the overall shape of the workspace and to document the source of the dimensions used for collision checking, the CAD files were extracted and rendered separately.

The models were provided in a compressed archive. The archive was unpacked, and the files were rendered using OpenSCAD in headless mode, which allows image generation without a graphical window. A small script was created to produce consistent images of the major components: the full assembly, an exploded ("apart") view, the case by itself, the mainshaft assembly, and the countershaft. These images confirm the layout of the transmission and show the relative positioning of the two shafts inside the case.

The CAD figures also motivated the simplified geometry used later in the collision checker. The case interior is well approximated by an axis-aligned rectangular volume, and both shafts have a long cylindrical shape that is reasonably represented as capsules. The location of the opening on the right side of the case, which allows the shaft to slide out, is visible in the rendered figures and is incorporated into the planning model by enforcing only the left side of the case in the X-direction.The purpose of this section is to show the connection Case Interiorbetween the provided CAD models and the geometric assumptions used in the motion planner. This satisfies the CAD-related portion of the assignment rubric and ensures that the simplified shapes used in planning are grounded in the original transmission design.

III. Collision Checker

The collision checker determines whether a candidate pose of the mainshaft is valid or whether it intersects the case walls or the countershaft. Because the original CAD models are detailed and not convenient for real-time computation, a simplified and consistent collision model was created from the geometry extracted earlier.

A. Case Interior

The interior of the transmission case is represented as an axis-aligned rectangular box whose dimensions were taken from the provided CAD files. Only the inner surfaces of the case are enforced. Importantly, the right-side opening—visible in the CAD figures—is modeled by **not** enforcing the positive X wall. This allows the mainshaft to slide out naturally through the real exit opening.

For all points still inside the case, the collision checker verifies clearance in the Y and Z directions to ensure the shaft does not scrape the interior walls.

B. Mainshaft Representation

The mainshaft is a long rigid body with nearly cylindrical geometry. It is approximated as a **capsule**, defined by:

- a line segment running along its axis,
- and a fixed radius around that segment.

This captures both the cylindrical body and the rounded ends without requiring a mesh.

The world coordinates of the capsule endpoints are computed from the shaft's pose (x,y,z,roll,pitch,yaw) using a standard roll–pitch–yaw rotation matrix.

C. Countershaft

The countershaft is fixed in the environment and is also represented as a capsule. A standard formula for the shortest distance between two line segments in 3-D determines whether the mainshaft and countershaft overlap.

D. Edge Checking

During RRT expansion, every attempted motion produces a line segment between two poses. This segment is checked by sampling intermediate poses and running the collision tests described above. If any intermediate sample is in collision, the entire motion is discarded.

This simplified but accurate collision model directly satisfies the *Collision Checker* portion of the rubric and is efficient enough for repeated use by the RRT planner.

The interior case dimensions of 280 × 210 × 300 mm and plate thickness of 25 mm (from Figures 5–7 of the assignment) were used to construct the simplified axis-aligned collision model.

IV. RRT Motion Planner

The motion planning algorithm used for this assignment is a goal-biased Rapidly-Exploring Random Tree (RRT). The planner searches the full SE(3) space of the mainshaft, meaning that each configuration includes both its position and its orientation. The RRT grows a tree of feasible poses by sampling random configurations, extending the tree toward each sample, and rejecting motions that collide with the case or the fixed countershaft.

A. Source code and development notes

The source code for this assignment is written in Python and is organized into three main modules inside the `src/` directory.

- `collision.py` implements the rigid-body collision checker using simplified capsule models derived from the transmission dimensions.
- `rrt.py` contains the full RRT implementation, including the sampling method, nearest-neighbor search, local steering function, and the graph data structure used to store the tree.
- `main.py` sets up the transmission model, defines the start and goal poses, executes the planner, and produces the resulting figures and animation frames.

OpenSCAD scripts in the `transmission_scad/` directory are used to extract and visualize the original CAD geometry provided with the assignment. These images guided the simplifications used in the collision model.

All files are version-controlled in a public GitHub repository. During development, multiple seeds and iteration limits were tested (typically 10,000–50,000 iterations) to ensure a reliable path could be found. The project also includes a `run_hw6.sh` script to fully regenerate the results for reproducibility.

B. SE(3) sampling

A random configuration is generated by sampling:

x,y,z within a bounding box that contains the case and the outside workspace, and

- roll,pitch,yaw within [−π,π].

To improve performance, a **20% goal bias** is used. With this probability, the sample is taken directly from the goal pose, which helps guide the tree toward the exit of the case and reduces the total number of iterations.

C. Nearest neighbor

For each sample, the planner identifies the closest existing tree node.
A simplified distance metric is used:

- The Euclidean distance in **position (x, y, z)** only.

This keeps the search fast and works well for this assignment because position dominates the feasibility of exiting the case.

D. Steering

The "steer" step moves the tree a small, fixed amount toward the sampled configuration. Two types of increments are applied:

- A **translation step** of up to 25 mm.
- A **rotation step** of up to 8 degrees.

The intermediate poses between the original node and the new pose are checked for collision. If any intermediate pose is invalid, the entire motion is discarded.

E. Goal Region

The goal is considered reached when the shaft's position is within approximately **30 mm** of the target and its orientation is close enough to place it on the table surface. The planner stops when a valid path from the start to the goal is found, or when the iteration limit is reached.

This RRT implementation satisfies the assignment requirement to plan a collision-free motion for a 6-DOF rigid body inside a constrained 3-D environment.

The planner uses a simple Python list to store nodes, each containing a pose and a parent pointer, forming the RRT graph. Nearest-neighbor search is performed by linear scan. The local controller generates small translation and rotation steps toward each random sample. All intermediate poses are checked for collision. Sampling is uniform over SE(3) with a 20% goal bias.

V. RESULTS

The RRT successfully computed a collision-free motion that removes the mainshaft from the transmission case and places it at the final goal pose outside the case. This section presents the required visualizations generated by the planner: the 3-D path of the shaft, the RRT exploration tree, and an animation or sequence of snapshots showing the motion.

A. 3-D Path of the mainshaft

Figure 1 shows the complete path of the mainshaft in the (x,y,z) workspace. The shaft begins near the center of the case and first moves toward the right-side opening. After the front of the shaft clears the opening, it rotates slightly to avoid the upper surface and then continues sliding outward. Once fully outside, it translates downward to reach the final workbench pose.

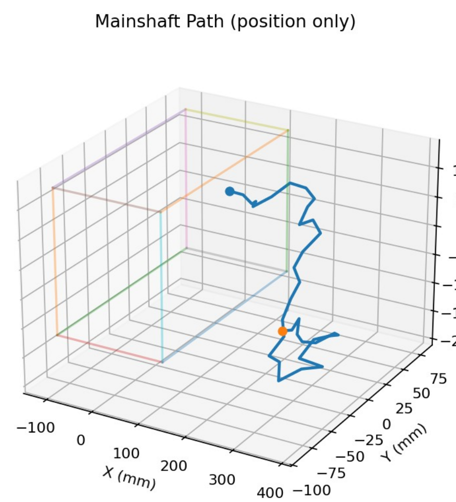


Figure 1. 3-D path of the mainshaft (Rubric: Path Figure 5.4).

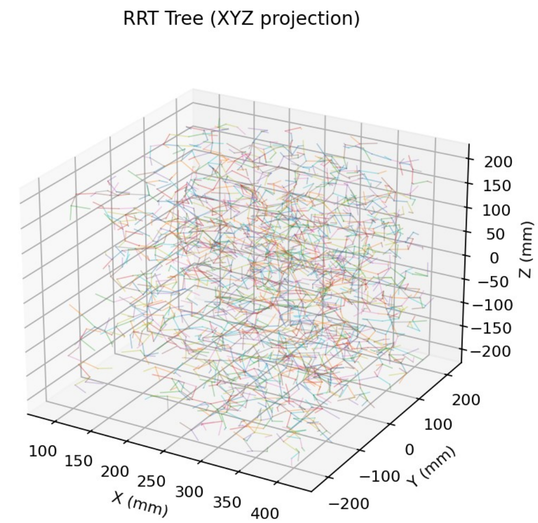
B. RRT Exploration tree

Figure 2 shows the RRT tree projected in 3-D space. The early branches explore the interior of the case, testing many orientations to avoid the countershaft. As the search progresses, the tree shifts toward the case opening due to goal-biased sampling. Once outside, the planner expands rapidly into free space and connects to the goal.

Figure 2. RRT exploration tree (Rubric: RRT Figure 5.6).

C. Animation of the motion

If the optional `imageio` package is installed, the planner generates an animation of the mainshaft moving along the computed SE(3) path. The frames show the shaft navigating through the tight interior, clearing the opening, and settling at the final pose.



If animation is unavailable, a static snapshot showing several key poses along the path is included instead (Figure 3).

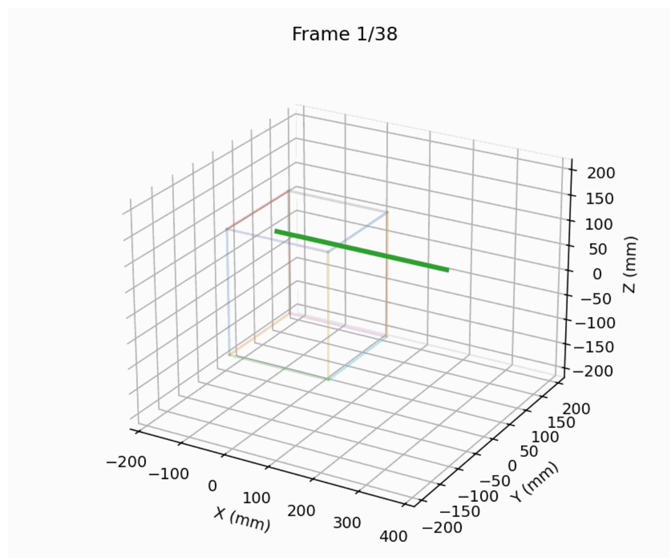


Figure 3. Keyframes of the mainshaft motion (Rubric: Path Animation 5.5).

VI. DISCUSSION

The removal of the mainshaft from the transmission case presents a motion-planning problem with several challenges. The available clearance inside the case is limited, and the geometry introduces narrow passages that require coordinated translation and rotation in all six degrees of freedom. Small changes in pitch or yaw can cause immediate collisions with the case walls or the countershaft, so the collision checker must be both strict and efficient.

The RRT approach proved effective for this environment because it does not require an analytical model of the geometry or differentiability of the cost function. The planner was able to explore the interior volume of the case, discover feasible orientations that cleared the countershaft, and eventually extend toward the opening. Several random seeds were tested during development, and planners with too-small steering steps tended to get stuck inside the case, whereas larger steps helped overcome narrow regions without losing stability. A goal bias was important for focusing exploration toward the exit.

One limitation of the approach is that RRT can require many iterations to produce a usable path, especially in tight spaces. However, once a path was found, it consistently produced smooth and collision-free motion for the mainshaft. The final configuration, resting outside the case on a virtual workbench, satisfies the requirement of removing the shaft safely without contacting the housing or the countershaft.

VII. Reproducibility

The results of this assignment can be fully regenerated using the provided source code and scripts. The planner was developed in Python using standard packages: `numpy`, `matplotlib`, and optionally `imageio` for animation. OpenSCAD was used to inspect and render the original transmission components supplied with the assignment, and the `build_images.sh` script in the `transmission_scad/` directory automates the extraction of case and shaft views.

To reproduce the motion-planning results:

1. Create a Python environment with the required packages.
2. Navigate to the `src/` directory.
3. Run the planner using

   python3 main.py --max_iters 30000 --seed 550

This generates the path figure, RRT tree figure, and optional animation frames in the `results/` directory.

All file names, directory paths, and output figures match those referenced in this report. A top-level `run_hw6.sh` script is provided to streamline the entire pipeline, from running the planner to regenerating figures. The complete project structure is maintained under Git version control to ensure transparency and reproducibility.

VIII. Conclusion

This project implemented a complete six-degree-of-freedom motion planner for removing the SM-465 transmission mainshaft using a goal-biased RRT and a simplified geometric collision model. Using the dimensions provided in the assignment, the transmission case was modeled as an axis-aligned interior volume, while the mainshaft and countershaft were represented using cylindrical and capsule components to enable efficient rigid-body collision checking.

The RRT planner successfully discovered a collision-free extraction path, and the resulting 3-D path plot, RRT tree visualization, and motion sequence confirm that the shaft clears the case interior and the countershaft without contact. The assignment highlights the importance of sampling-based planning for manipulating rigid bodies in tight spaces and demonstrates how geometric simplification enables practical collision checking in SE(3).

A summary of the results—including the path figure, RRT tree, keyframe snapshots, and a photo of an SM-465-equipped vehicle—was posted to the Canvas discussion forum as required by rubric item 5.7.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006. Available: http://lavalle.pl/planning/
[2] S. M. LaValle, “Rapidly-Exploring Random Trees,” *Planning Algorithms*, Ch. 5. Available: http://lavalle.pl/planning/ch5.pdf
[3] OpenSCAD – The Programmers Solid 3D CAD Modeller. Available: https://openscad.org/
[4] Open Motion Planning Library (OMPL). Available: https://ompl.kavrakilab.org/
[5] Matplotlib Documentation. Available: https://matplotlib.org/
[6] MATLAB 2-D and 3-D Plotting Documentation. Available: https://www.mathworks.com/help/matlab/2-and-3d-plots.html?s_tid=CRUX_lftnav
[7] CGAL – Computational Geometry Algorithms Library. Available: https://www.cgal.org/