AERO7970 - Trajectory Optimiztion

Project 03

Matt Boler

October 23, 2022

Abstract

This report details the implementation of a fuel-optimal close-proximity rendezvous maneuver with obstacle avoidance. The solution was implemented in `Matlab` usingthe built-in `solve` function. Included are performance comparisons and the resulting trajectories and control inputs.

1 Problem Modeling

We are given the optimization problem from Project 1 with the added goal of avoiding the obstacle defined by

- $-0.6km \leq x \leq -0.5km$
- $0.45km \leq y \leq 0.55km$
- $-0.01km \leq z \leq 0.01km$

We use the same techniques as before (introducing a variable to bound the L1 norm, discretizing the dynamics, etc). In addition, we use the Big-M technique to handle the nonfeasible region of the obstacle by introducing variables Z with constraints:

$$\begin{gathered}
-X(t) \leq -X_{max} + M * Z_{x1}(t) \\
-Y(t) \leq -Y_{max} + M * Z_{y1}(t) \\
-Z(t) \leq -Z_{max} + M * Z_{z1}(t) \\
X(t) \leq X_{min} + M * Z_{x2}(t) \\
Y(t) \leq Y_{min} + M * Z_{y2}(t) \\
Z(t) \leq Z_{min} + M * Z_{z2}(t)
\end{gathered}$$

2 Results

The problem was solved in `Matlab` using the `intlinprog` solver. The resulting trajectory is presented below alongside the original trajectory for comparison. It is worth noting that in problems of this form, while the individual discretization points may avoid the obstacle, actually following the solved trajectory may cause the chaser to collide with the obstacle. This is due to the discretization of the trajectory, which causes constraints to only be evaluated at select points. The trajectory between the discretization points can intersect the obstacle without violating constraints.

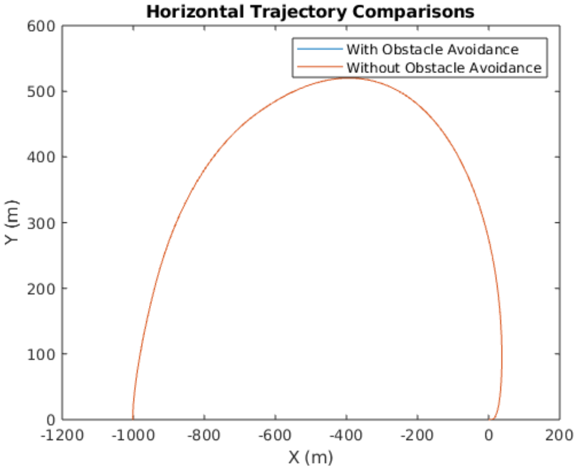

Figure 1: Overhead view of generated trajectories

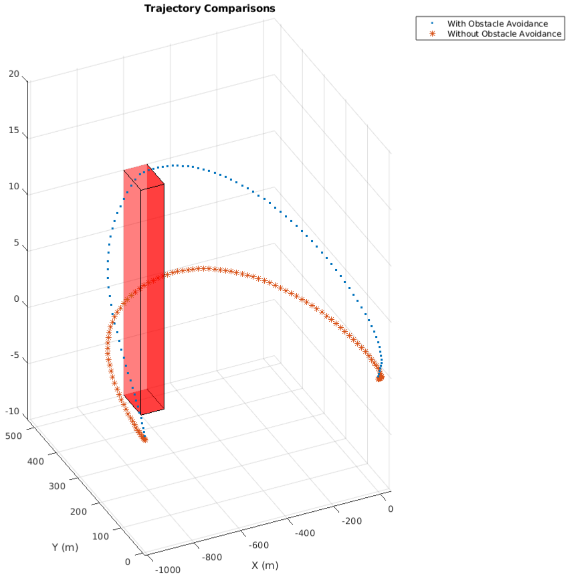

Figure 2: 3d view of generated trajectories

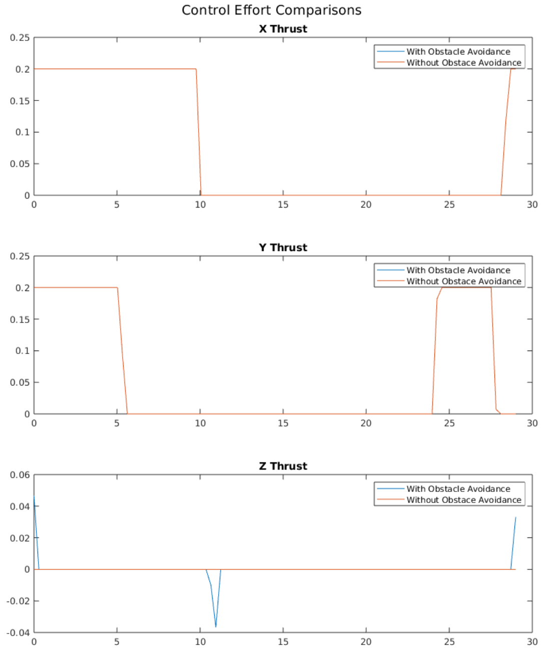

Figure 3: Optimal control efforts

Afterwards, the same problem was solved with the modificaion of $Y_{min} = 0.35km$. This resulted in a very similar optimal trajectory, except that the chaser now flies under the obstacle instead of over it. The results are shown below.

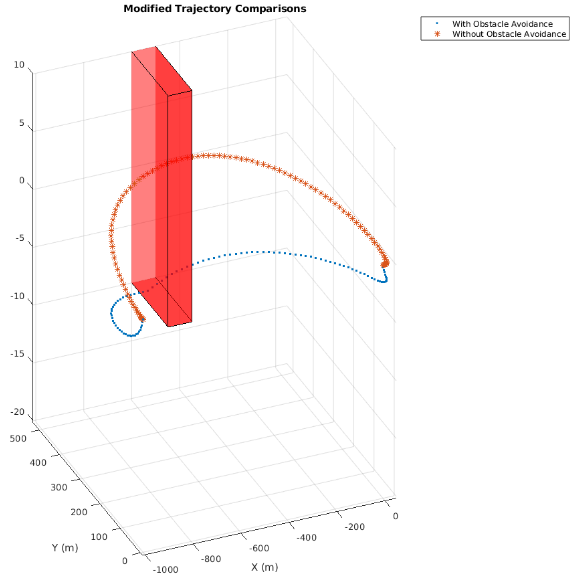

Figure 4: 3d view of generated trajectories

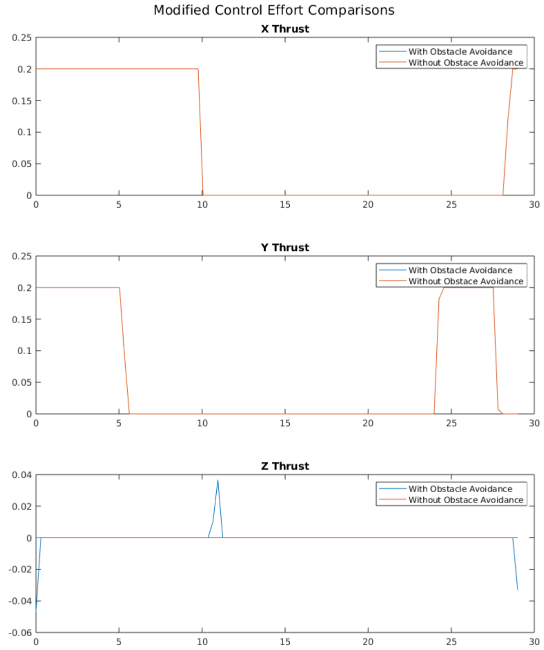

Figure 5: Optimal control efforts

Conclusion

This problem did not have any major challenges, as it was a simple modification of an earlier problem. The main thing I learned is that integrating square (or generally convex?) state constraints is simple and fast, which should help me down the road in my research. I spent approximately 3 hours on this project.