Surface Constraints for 3-D Path Planning

Mike Sutherland

University of California, Irvine

Nov 4, 2021

We have a problem with planning in 3D.

2.5D approaches:

- "plane sandwich" approach
- when to hop planes?
- need to assess connectivity of each plane
- not optimal

Full 3D?

- Very expensive: A* on WxHxZ matrix?
- Other approaches (RRT, PRM, etc) can cut compute time...
- ...but come with own downsides
- *we want to spend most of the time close to the ground anyway*

...we want to spend most of the time close to the ground anyway...

Specifically:

- at least some minimum distance off the ground
- at least some minimum distance from an obstacle
- not climbing too quickly
- not changing climb rate too quickly

We can solve a convex problem

$$
\begin{aligned}
\underset{h}{\text{minimize}} \quad & g(h) = h^2 && \text{(1a)} \\
\text{subject to} \quad & \frac{\partial h}{\partial \bar{x}} \leq h_{accel}, && \text{(1b)} \\
& \frac{\partial^2 h}{\partial \bar{x}^2} \leq h_{climb}, && \text{(1c)} \\
& h \geq h_{min}, && \text{(1d)} \\
& h \geq h_{obs} + h_{barrier}, && \text{(1e)} \\
& \bar{x} \in (\mathcal{X}, \mathcal{Y}) && \text{(1f)}
\end{aligned}
$$

This optimization produces a mapping, h, for every $\bar{x} \in (\mathcal{X}, \mathcal{Y})$, which corresponds to the "sheet" of possible positions for the aircraft.

What does this look like?

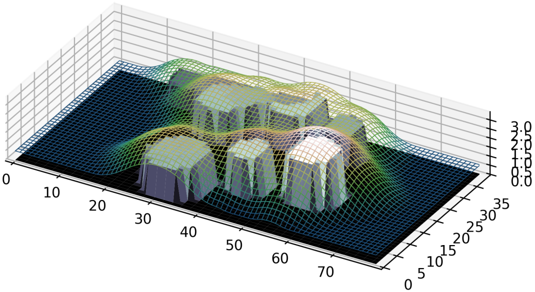

Figure: The mesh created by solving the optimization problem. We have an area of obstacles, each with a height. The solution is a mapping from $(\mathcal{X}, \mathcal{Y}) \to H$, which is shown here in 3-d by the wireframe. The surface is smooth and as low as possible without hitting the obstacles.

Runtime

We are solving an optimization problem over a mesh. So, runtime is roughly quadratic.

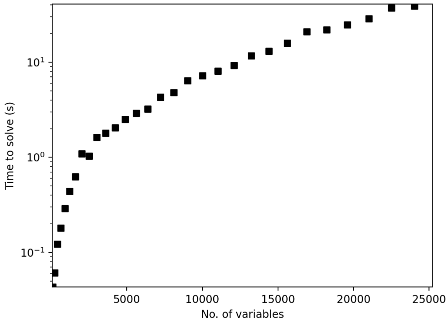

Figure: Runtime of the algorithm. Each grid point counts as a variable, and grid ponts scale roughly linearly with area, assuming an equal distance between points.

Now, how to plan?

- Planners such as A* and derivatives, RRT, PRM, and so on can be used in $\mathbb{R}^2$ as normal
- If desired, cost heuristics that use the mapping can be used to weight the path to be lower or higher.
- We lose 1 degree of freedom when re-planning around dynamic obstacles; for example, we cannot fly below or above the sheet to move around obstacles.
- Collisions with dynamic objects can be cheaply calculated in $\mathbb{R}^2$ if necessary.
- For dynamic objects, spherical collisions are cheap for the sheet.

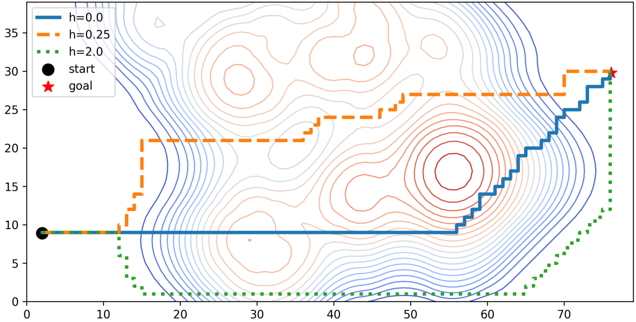

Figure: A standard A* planner moves from one point to another through a field of obstacles (the same field as fig. 1). A different cost weight is applied on h. When h is higher, the planner strongly penalizes h, whereas when h is lower, the planner does not care about h as much. All paths, however, are tractable within the vehicle dynamics and collision constraints.

Next Steps

- Dynamic sampling density: more points sampled near obstacles
- Explore cost heuristics in algorithms (and make some proofs?)
- Boundary conditions with $h_{obs} = \infty$ obstacles
- Speed up optimization?