Fundamental Robotics
a comprehensive summary of robotics

Nils Funk

Contents

1 Trajectory Planning

1.1 Liu - Planning Dynamically Feasible Trajectories for Quadrotors Using Safe Flight Corridiors in 3-D Complex Environments

1.1.1 Contribution

A method to formulate trajectory generation as a quadratic program (QP) using the concept of a Safe Flight Corridor (SFC)

1.1.2 Summary

The SFC is a collection of convex overlapping polyhedra that models free space and provides a connected path from the robot to the goal position.

Derive an efficient convex decomposition method that builds the SFC from a piece-wise linear skeleton obtained using a fast graph search technique.

The SFC provides a set of linear inequality constraints in the QP allowing real-time motion planning.

Develop a framework of Receding Horizon Planning, which plans trajectories within a finite footprint in the localmap, continuously updating the trajectory through a re-planning process.

Pipeline:

A. Path Planning

Map representation: Occupancy grid
Graph search algorithm: Jump Point Search (JPS)

B. Safe Flight Corridor Construction

Path $P = \langle \mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_n, \rangle$
Line segement $L_i = \langle \mathbf{p}_i \rightarrow \mathbf{p}_{i+1} \rangle$
Flight corridor C_i

Generate a convex polyhedron around each line segement L_i in Path $\mathbf{P}$ to construct a valid $SFC(\mathbf{P})$.

One criterion for the construciton of the SFC is that two consecutive polyhedra need to intersect in a non-empty set containing $\mathbf{p}_{i+}$.

- Use bounding box to confine the space around L_i and to reduce computation time
- Use shrinking process to guarantee that a non-point robot is collison-free

Steps:
1. Find ellipsoid that does not include any obstacle points O
Ellipsoid ξ axis a, b, c
- Ellipsoid is a sphere centered at the mid point of L_i and with a diameter equal to the length of L_i
- Assume the length of ellipsoid's $\tilde{x}$-axis is fixed and aligned with L
- Reduce lenth of the other two axes until the spheroid contains no obstacles

1.1 Shrink an initial sphere to derive the maximal spheroid (an ellipsoid with two axes of qual length
Search for closest obstacle $\mathbf{p}*$ in O form the center of the the ellipsoid ξ
The line segment L_i, defines the plane of $\tilde{x} - \tilde{y}$ axes of the spheroid

1.2 Stretch this spheroid along the third axis to obtain the final ellipsoid
Strech the length of the $\tilde{z}$-axis of the spheroid to make it equal to a to form a new initial ellipsoid.

The actual value of c can be determined through finding another closest point using the similar process.

2. Find polyhedron
Construct the polzhedron C_i from tangent plances to a sequence of dilated ellipsoids.

Iteratively dilate the ellipsoid until it is in contact with the next obstacle in point $\mathbf{p}_i$ and creat a half space $H_i = \{\mathbf{p}|\mathbf{a}_i^T\mathbf{p} < b_i\}$ and remove the obstacles that lie outside of H_i.

3. Bounding box
The bounding box for L is composed of 6 rectangles such that the axis of the bouding box is aligned with L and the minimum distance from each face to L is r_s.

$r_s =\geq \frac{v_{max}^2}{2a_{max}^2}$

4. Shrink
Use the original map M to generate the SFC and shrink the SFC by the robot radius r_r in order to guarantee safety. The shrinking process is applied by pushing every support hyperplane by r_r.

If the minimum distance $d(L_i, H'_{i,j}) < r_r$ (i.e. the line segment L_i is outside the shrunk polyhedron H') the normal of the hyperplane gets adjustet s.t. $d(L_i, H'_{i,j}) = r_r$.

C. Trajectory Optimiazation

Convex optimization for minimum snap trajectories is formed as a quadratic program with constraints for each trajectory segement Φ

$\frac{d^k}{dt^k}\Phi_i(\Delta t_i) = \frac{d^k}{dt^k}\Phi_{i+1}(0), k = 0...4$

$\mathbf{A}_i^T \Phi_i(t) < \mathbf{b}$

here the matrices $\mathbf{A}_i, \mathbf{b}_i$ correspond to the i-th polyhedron C_i

Time allocation:
Similar to [Richter]

$\Delta t'_i = max\{1, (\frac{v_{max}}{\bar{v}_{max}}), (\frac{a_{max}}{\bar{a}_{max}})^{\frac{1}{2}}, (\frac{j_{max}}{\bar{j}_{max}})^{\frac{1}{3}}\}\Delta t_i$

D. Receing Horiyon Planning (RHP)

Solve optimal control problem over a fixed future time interval with a planning horizon of distance d_r. The generation of the trajectory for the next epoch is started when the robot is executing the trajectorz at the current epoch. Choose execution time T_e such that the time to generte a new trajectory is guaranteed to be less. Since the execution time T_e is bigger than the time it takes for generating a trajectory, the robot is always able to transit to track a new trajectory when it finishes executing the current one.

1.1.3 Results

The re-planning process takes between 50 to 300 ms for a large and cluttered map. Approach is feasible, complete, with applications to high-speed flight in both simulated and physical experiments using quadrotors.

1.1.4 Flaws

In our optimization process, we use a sample-based method to confine each polynomial, the details for which can be found in [2]

Note: The optimization process seems still to just sample the trajectory at certain points, rather than to check if the trajectory segement is within the flight corridor at all times.

1.1.5 Highly Related Work

- Trajectory planning: Derivation of UAV differential flatness and optimization based trajectory gernation of minimum snap trajectories with corridor constraints
  Mellinger - Minimum snap trajectory gerneation and control of quadrotors
- Trajectory planning:
  Richter - Polynomial trajyectory planning for aggressive quadrotor flight in dense indoor environements
- Trajectory planning:
  Chen - Online generation of collison-free trajectories for quadrotor flgit in unknown cluttered environments
- Trajectory planning:
  Liu - High speed navigation for quadrotors with limited onboard sensing
- Trajectory planning:
  Watterson - Safe receding horiyon control for aggressive MAV flight with limited range sensing
- Trajectory planning:
  Mellinger - Trajectory genration and control for quadrotors
- Graph Search Algorithm: Jump Point Search - Fast search strategy in uniform-cost grids that can speed up A* by an order of magnitude
  Harabor - Online graph pruning for path finding on grid maps

1.1.6 Covered Topics

Quadratic program, convex polyhedra, convex decompositon method, linear inequality constraints, receding horizon planning (RHP), trajectory parameterization, mixed integer methods, convex free space, generic, parallelopipeds, A*, Dijkstra, jump point search (JPS), natural/purned neighbors, symmetric positive definit, ellipsoid, half space, non-linear control

1.2 Chen - Online Generation of Collision-Free Trajectories for Quadrotor Flight in Unknown Cluttered Environments

1.3 Mellinger - Minimum snap trajectory gerneation and control of quadrotors

1.4 Richter - Polynomial trajyectory planning for aggressive quadrotor flight in dense indoor environements

1.5 Chen - Online generation of collison-free trajectories for quadrotor flgit in unknown cluttered environments

1.6 Liu - High speed navigation for quadrotors with limited onboard sensing

1.7 Watterson - Safe receding horiyon control for aggressive MAV flight with limited range sensing

1.8 Mellinger - Trajectory genration and control for quadrotors

1.9 Harabor - Online graph pruning for path finding on grid maps

1.9.1 Contribution

Jump Point Search - Fast search strategy in uniform-cost grids that can speed up A* by an order of magnitude