Online Safe Trajectory Generation For Quadrotors Using Fast Marching Method and Bernstein Basis Polynomial

Year	2018
Author	Fei Gao, William Wu, Yi Lin and Shaojie Shen
Published in	ICRA
Score /5	Empty
Keywords	Planning
Last Edited Time	Jul 17, 2020 4:59 AM

简介

本文提出了一种四旋翼运动规划方法，该方法可以在未知环境中实时地生成安全且可执行的轨迹。

相比于之前的工作，本文主要解决以下两个问题：

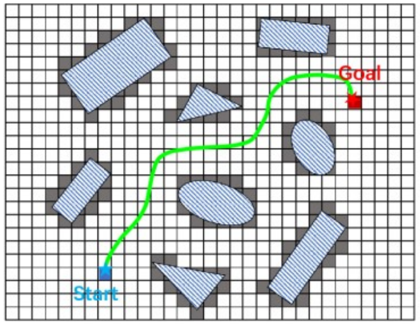
1. 时间分配问题，不恰当的时间分配会影响轨迹的质量。
2. 高效的可行解约束问题，即如何将可行解空间内（如存在障碍物时的 free space、速度、加速度等约束）的约束在优化问题中表示出来。

本文的主要贡献为:

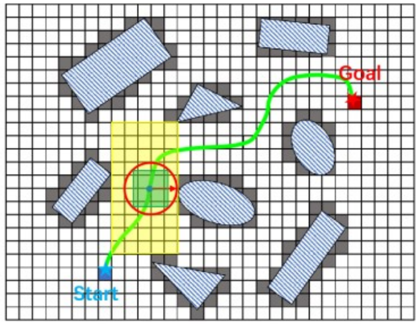
1. 采用基于欧氏距离场速度域（ESDF-based velocity field）上的 Fast marching 方法[4]，得到合理的时间分配。
2. 采用基于 Bernstein 的贝赛尔曲线用于轨迹优化。

Fast marching-based path searching

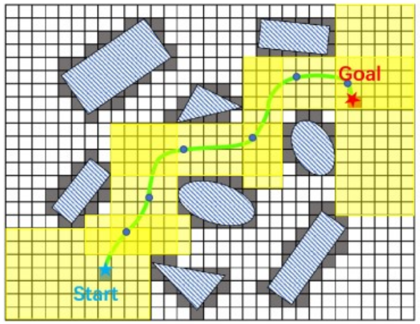
飞行走廊生成



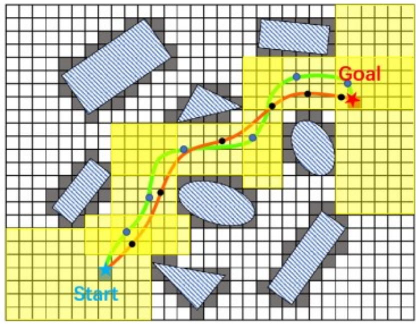
(a) Path found by the front-end.



(b) Free cube generated and inflated.



(c) Complete flight corridor.



(d) Trajectory generated.

Safe and dynamically feasible trajectory generation

伯恩斯坦多项式

这里用 伯恩斯坦多项式 来参数化轨迹，伯恩斯坦的多项式的基函数定义为：

$$b_n^i(t) = \left(\begin{array}{c} n \\ i \end{array} \right) \cdot t^i \cdot (1-t)^{n-i}, \quad t \in [0,1]$$

其中 n 为自由度，下面这个为二项式系数：

$$\left(\begin{array}{c} n \\ i \end{array} \right) = \frac{n!}{i!(n-i)!} \quad \text{for } 0 \leq i \leq n$$

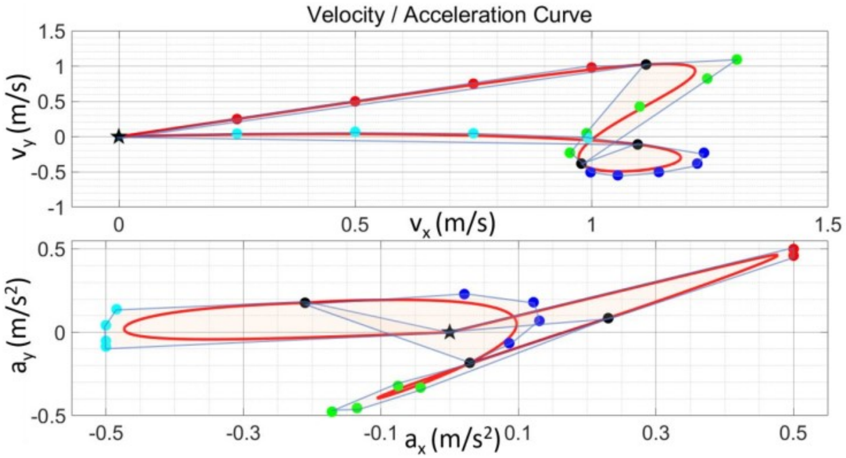
则贝塞尔曲线可以写为：

$$B_j(t) = c_j^0 b_n^0(t) + c_j^1 b_n^1(t) + \ldots + c_j^n b_n^n(t) = \sum_{i=0}^{n} c_j^i b_n^i(t)$$

其中 $[c_j^0, c_j^1, \ldots, c_j^n]$ 记为 $\mathbf{c}_j$，表示第 j 段贝塞尔曲线的控制点。

伯恩斯坦多项式的性质

- **Endpoint interpolation**，塞尔曲线的起点位置和第一个控制点重合，终点位置和最后一个控制点重合，其余控制点都不会经过。可以将 $t=0$ 和 $t=1$ 带入得出结论。
- **Convex hull**，贝塞尔曲线被限制在由所有控制点组成的凸多边形中。
- **Hodograph**，贝塞尔曲线求导仍然是贝塞尔曲线，新的控制点为 $c^{(1)i} = n \cdot (c^{i+1} - c^i)$。
- **Fixed time interval**，贝塞尔曲线的时间定义域为 $t \in [0,1]$。



1

多段贝塞尔曲线

对于实际的问题需要进行归一化。特别的，对于属于维度 μ 中 m 段轨迹的轨迹可以写为：

$$f_\mu(t)=\begin{cases} s_1\cdot\sum_{i=0}^{n}c_{\mu 1}^{i}b_n^i\left(\frac{t-T_0}{s_1}\right), & t\in[T_0,T_1]\\ s_2\cdot\sum_{i=0}^{n}c_{\mu 2}^{i}b_n^i\left(\frac{t-T_1}{s_2}\right), & t\in[T_1,T_2]\\ \vdots & \vdots\\ s_m\cdot\sum_{i=0}^{n}c_{\mu m}^{i}b_n^i\left(\frac{t-T_{m-1}}{s_m}\right), & t\in[T_{m-1},T_m]\end{cases}$$

其中 $s_1,\ldots,s_m$ 为归一化系数，将时间 t 转换到 $[0,1]$。注意每一段还乘上了归一化系数，这是为了获得更好的数值稳定性。

代价函数

这里采用 minimum jerk 作为代价函数，故下面式子中 $k=3$。若采用 minimun snap，则 $k=4$。

$$J=\sum_{\mu\in\{x,y,z\}}\int_0^T\left(\frac{d^k f_\mu(t)}{dt^k}\right)^2 dt$$

对于第 j 段的贝赛尔曲线 $f_{\mu j}(t)$，化成在时间 $[0,1]$ 内标准的贝塞尔曲线 $g_{\mu j}(\tau)$，$\tau=(t-T_{j-1}/s_j)$，则维度 μ 的第 j 段对应的代价函数 $J_{\mu j}$ 为：

$$\begin{aligned}J_{\mu j}&=\int_0^{s_j}\left(\frac{d^k f_{\mu j}(t)}{dt^k}\right)^2 dt=\int_0^1 s_j\left(\frac{s_j\cdot d^k\left(g_{\mu j}(\tau)\right)}{d\left(s_j\cdot\tau\right)^k}\right)^2 d\tau\\&=\frac{1}{s_j^{2k-3}}\cdot\int_0^1\left(\frac{d^k g_{\mu j}(\tau)}{d\tau^k}\right)^2 d\tau\end{aligned}$$

根据以上，可以构建二次规划问题 $\mathbf{p}^T\mathbf{Q}_0\mathbf{p}$，其中 $\mathbf{p}$ 包含了所有的控制点。其中 $\mathbf{Q}_0=\mathbf{M}'\mathbf{Q}\mathbf{M}$，Bernstein polynomial 和 monomial polynomial 的关系为 $p=\mathbf{M}\cdot c$，$\mathbf{Q}$ 的计算参考 Kumar 的论文[13]。

约束条件

对于分段轨迹生成问题，我们需要添加一定的约束保证轨迹的平滑性、安全性和可行性。对于伯恩斯坦的贝塞尔曲线，高阶控制点可以由低阶控制点的线性组合表示：

$$a_{\mu j}^{0,i}=c_{\mu j}^{i}, a_{\mu j}^{l,i}=\frac{n!}{(n-l)!}\cdot\left(a_{\mu j}^{l-1,i+1}-a_{\mu j}^{l-1,i}\right), l\geq 1$$

其中 l 为导数的次数。约束条件共 4 类，分别为：

1）Boundary Constraints：限制轨迹起始点（终点）的位置、速度和加速度等。在起始点的约束：

$$a_{\mu j}^{l,0}\cdot s_j^{(1-l)}=d_{\mu j}^{(l)}$$

2）Continuity Constraints：前一段贝塞尔曲线的终点和后一段贝塞尔曲线的起点（位置、速度、加速度等）必须保持一致。对于第 j 段和第 $j+1$ 段，我们有：

$$a_{\mu j}^{\phi,n}\cdot s_j^{(1-\phi)}=a_{\mu,j+1}^{\phi,0}\cdot s_{j+1}^{(1-\phi)}$$

3）Safety Constraints：贝塞尔曲线的控制点位置被约束在相应的可行区域内。（论文中公式有误）

$$\beta_{\mu j}^{-} \leq c_{\mu j}^{i} \cdot s_j \leq \beta_{\mu j}^{+}, \mu \in \{x, y, z\},\ i = 0, 1, 2, \ldots, n$$

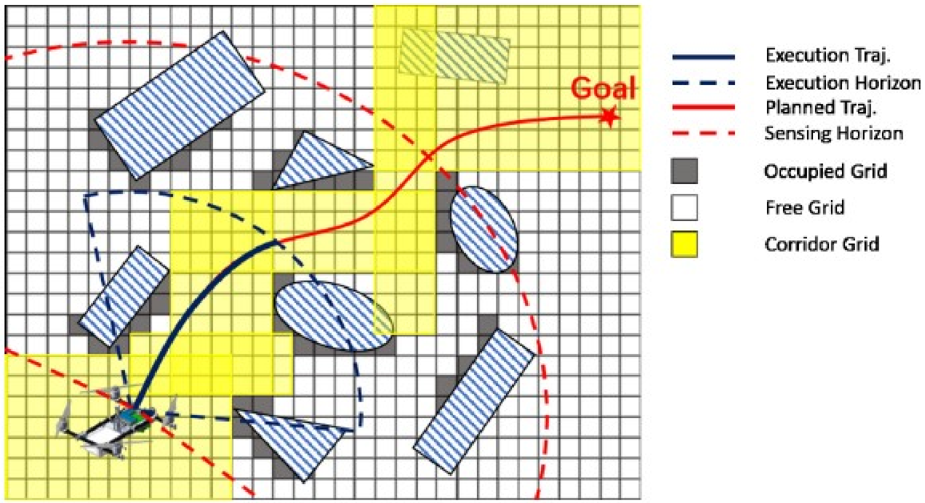
4）Dynamical Feasibility Constraints：贝塞尔曲线上每一点的速度、加速度等被约束在一定范围内。

$$\begin{aligned} & v_m^{-} \leq n \cdot \left(c_{\mu j}^{i} - c_{\mu j}^{i-1}\right) \leq v_m^{+} \\ & a_m^{-} \leq n \cdot (n-1) \cdot \left(c_{\mu j}^{i} - 2c_{\mu j}^{i-1} + c_{\mu j}^{i-2}\right) / s_j \leq a_m^{+} \end{aligned}$$

二次规划问题

将上述问题写成二次规划的形式：

$$\begin{array}{ll} \min & \mathbf{c}^T \mathbf{Q}_o \mathbf{c} \\ \text{s.t.} & \mathbf{A}_{eq} \mathbf{c} = \mathbf{b}_{eq} \\ & \mathbf{A}_{ie} \mathbf{c} \leq \mathbf{b}_{ie} \\ & \mathbf{c}_j \in \Omega_j, \quad j = 1, 2, \ldots, m \end{array}$$



点评

源码

https://github.com/HKUST-Aerial-Robotics/Btraj

References

1. 快速行进算法（Fast Marching）_人工智能_lusongno1的博客-CSDN博客
2. https://en.wikipedia.org/wiki/Bernstein_polynomial
3. Trajectory generation using Bernstein polynomial - 知乎

[4] J. A. Sethian, “Level set methods and fast marching methods,” Journal of Computing and Information Technology, vol. 11, pp. 1–2, 2003.

[13] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in Proc. of the IEEE Intl. Conf. on Robot. and Autom., Shanghai, China, May 2011, pp. 2520–2525.