

Minimum Snap轨迹规划详解

(2) corridor与时间分配

轨迹规划

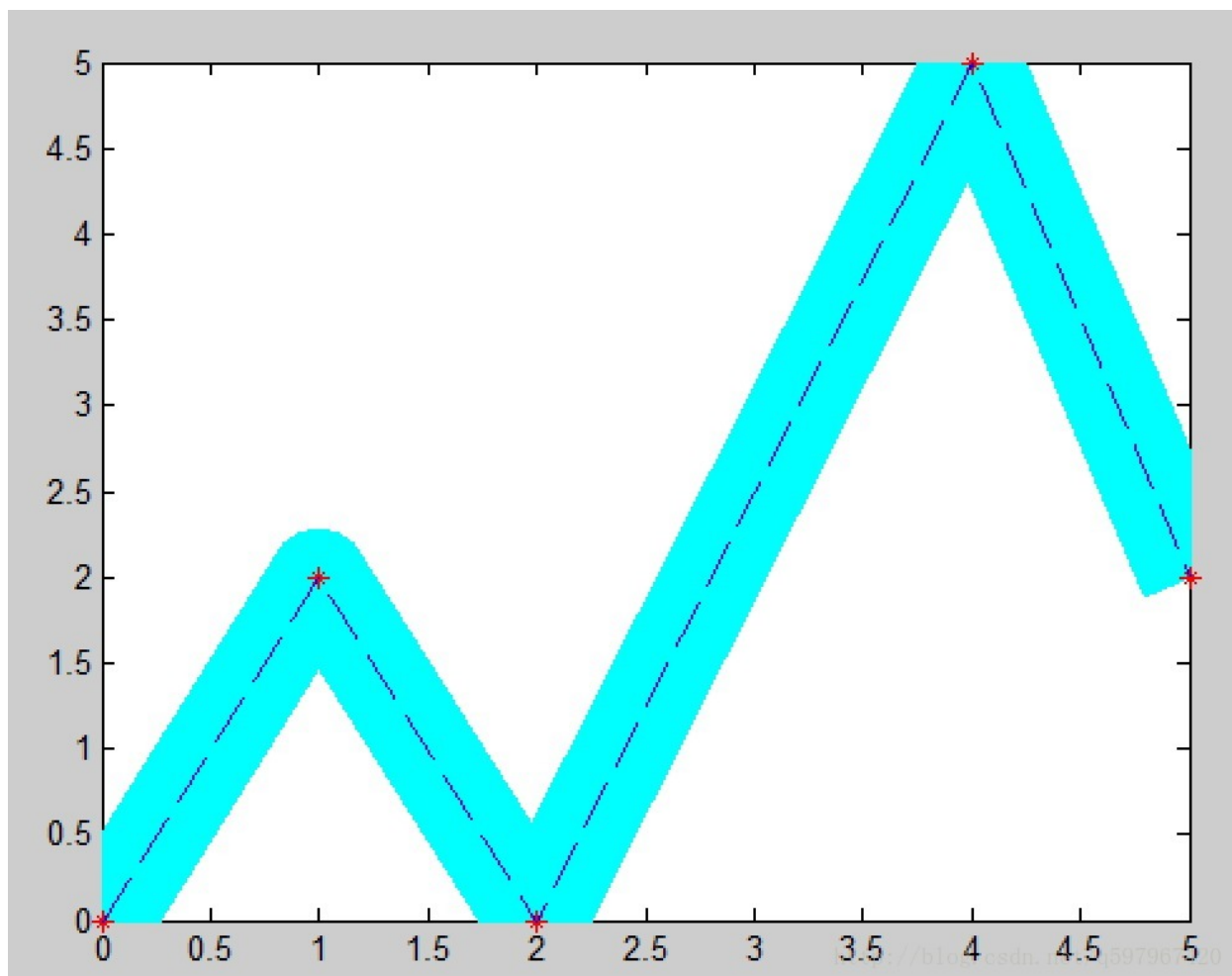
在上一篇文章中，我们得到的轨迹并不是很好，与路径差别有点大，我们期望规划出的轨迹跟路径大致重合，而且不希望有打结的现象，而且希望轨迹中的速度和加速度不超过最大限幅值。为了解决这些问题有两种思路：

- 思路一：把这些“期望”加入到优化问题中。
- 思路二：调整时间分配，来避免这些问题。

1.corridor

1.1 corridor是什么？

为了限制轨迹的形状，引入了corridor的概念，corridor可以理解为可行通道，如下图，规划出的轨迹必须在corridor内。直观的思路是：如果能把corridor当作约束加入到QP问题中，那么解得的轨迹自然就在corridor内了。



1.2 corridor约束怎么加？

很容易想到，把之前的等式约束 $Ap = d$ 改成不等式约束 $Ap \leq d_1$ 和 $Ap \geq d_2$ 。然而，不管是等式约束还是不等式约束，都是针对一个特定的时刻，而实际希望的是对所有时刻 $t \in [0, T]$ ，都需要在corridor中。

一个简单粗暴的思路：在路径上多采样一些中间点，每个中间点都加corridor约束。尽管这种方法理论上只能保证采样点在corridor中，但实际过程中，如果corridor大小和采样步长设置得合理，而且不fix waypoints，能work的比较好。这里不fix 中间点的位置，是为了去掉中间点的强约束，尽量避免轨迹打结，实际中，我们也是不一样要求轨迹完全经过中间点，只要在那附近（corridor）内就行。

1.3 构造corridor不等式约束

为了方便构造，对于每一个采样点 pt ，我们施加一个矩形的corridor，即

$$x_{min} \leq pt_x \leq x_{max}, y_{min} \leq pt_y \leq y_{max}$$

这里用矩形corridor是因为斜线corridor（平行四边形）不好构造，而且我们只需要大致在corridor里面就好，没有必要非弄一个严格的斜线corridor。对于每个采样点，设矩形corridor的边长为 $2r$ ，增加两个位置不等式约束：

$$\begin{aligned} [1, t_i, t_i^2, \dots, t_i^n] p &\leq p(t_i) + r \\ [-1, -t_i, -t_i^2, \dots, -t_i^n] p &\leq -(p(t_i) - r) \end{aligned} \quad (1)$$

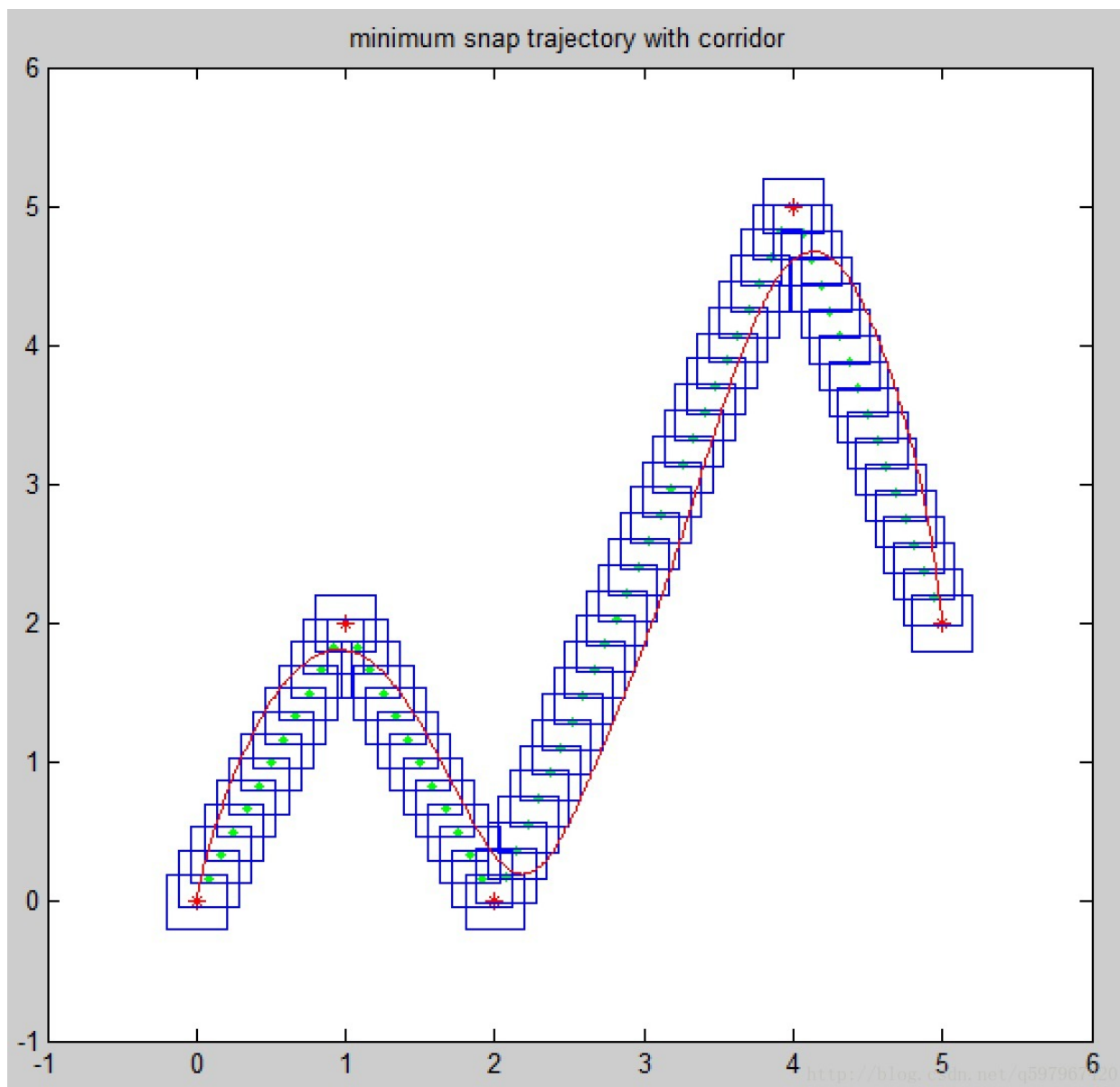
1.4 实验结果

[代码见这里](#)

优化列表

- **min**: snap
- 等式约束: 起点pva(3)+终点pva(3)+中间点pva连续($3*(n_{poly}-1)$)
- 不等式约束: 中间点位置corridor($2*(n_{poly}-1)$)

优化结果如下图所示, 绿色的点为按固定步长新采样的中间点, 每个中间点都有一个矩形corridor, 红色为优化后的轨迹。可以看到, 去掉中间点的位置强约束(等式约束)改成弱约束(corridor不等式约束)后, 轨迹比之前要好很多, 而且尽管轨迹有小部分超出了corridor (因为只加了**采样时刻**的位置corridor约束), 但基本不会超出太多。



但这带来的一个问题是, 为了保证轨迹不超出corridor, 需要采样比较密, 但这样轨迹的段数就上来的 (轨迹段数=waypoint数量-1), 优化参数变多了, 计算效率会下降。

- 一种解决思路是通过corridor合并来减少中间点\轨迹段数, 从而提高计算效率;
- 另一种解决思路是, 不采样中间点, 直接规划出轨迹以后, 找到轨迹的极值点 (波峰、波谷), 若极值点位置超出corridor, 在极值点上添加一个corridor或者位置强约束 (俗称“压点”, 把极值点压

到corridor以内)，以此迭代。实验证明：压有限个（不超过路径点总数）点就能把整段轨迹压到corridor以内。

1.5 广义corridor

上面实验介绍了位置corridor，也就是位置不等式约束，但同样也可以扩展到速度corridor、加速度corridor乃至更高阶。

诶~是不是有点灵感？上一篇文章中实验得到的轨迹的另一个问题是速度、加速度超过了最大限幅，所以同样的思路每个waypoint都增加速度、加速度corridor，是不是就能解决了？

然而，**并不能!!!** 为什么呢？举个简单的例子，假设一维空间规划一条轨迹从0到10，给定总时间 $T=1s$ ，要求最大限速 $2m/s$ ，满足要求的轨迹根本不存在，即使你加很多中间点并加加速度corridor，得到的轨迹在采样点中间肯定会速度蹭的巨高。

速度、加速度之所以很大，是因为时间给的不合理（时间太小），所以这种情况光靠加corridor是行不通的，还是要从时间分配的层面来解决。

2. 时间分配

时间分配是轨迹规划中很关键的一个问题，它直接影响规划结果的好坏。

2.1 初始时间分配

- 总时间给定：通常给一个平均速度，根据总路程计算得到总时间
- 各段时间分配：
 - 匀速分配：假设每段polynomial内速度恒定不变，这样各段poly的时间比就等于路程比。
 - 梯形分配：假设每段polynomial内速度曲线为：以恒定加速度 a 从0加速到最大速度 v_{max} ，再以 $-a$ 减速到0。 a 和 v_{max} 事先给定，就能得出每一段所需要的时间。

设各段时间分别为 t'_1, t'_2, \dots, t'_n ，求前k项和可以得到轨迹分段的时刻向量

$$t_0 = 0, t_1 = t'_1, t_2 = t_1 + t'_2, \dots, t_n = t_{n-1} + t'_n。$$

2.2 feasibility check

有了时间分配，就可以规划得到轨迹参数，对于轨迹参数，求速度和加速度曲线的极值，判断是否超过最大限幅，如果所有极值都小于最大限幅，则得到可行轨迹。如果不满足，则需要调整该段的时间。

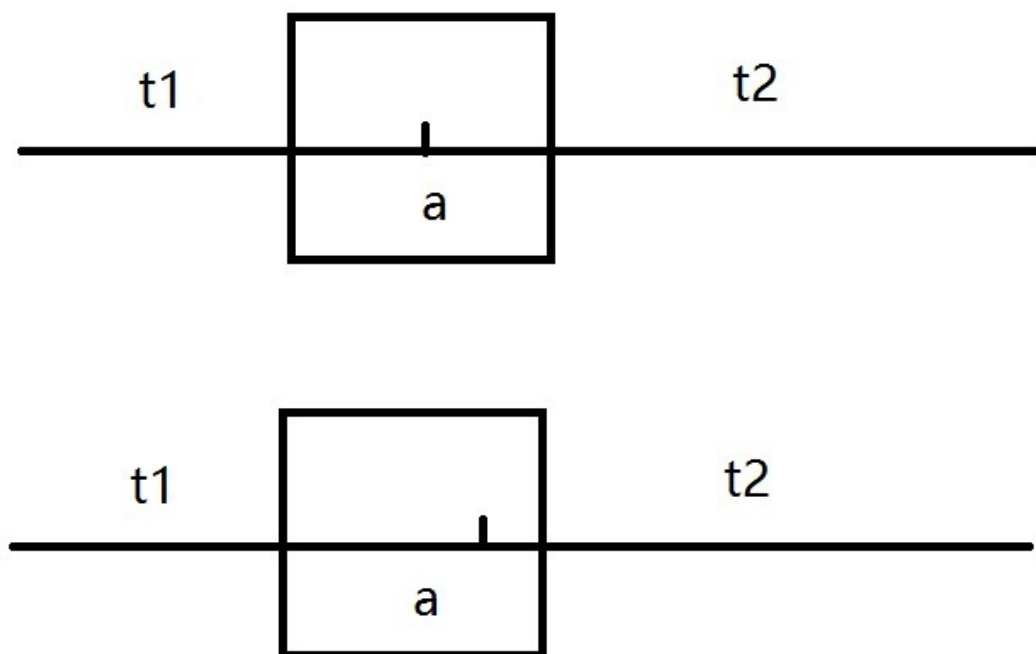
调整的方法是：增大unfeasible段poly的时间，显然，增大时间会使得速度和加速度都变小。通常以一个固定的比例来调整时间如（ $k=1.2\sim1.5$ ），并进行迭代，每调整一次，重新规划轨迹并check feasibility，如若不满足，再次加大时间，直至满足为止。

2.3 corridor与时间分配的关联

不等式corridor约束实际上也有时间分配的效果，可以这么想，polynomial的分段点（轨迹中间点）在corridor内移动，实际上就相当于分段点的时刻在移动，中间点在corridor内移动，也就变相调整了前后

两段poly分配的时间。

举个简单的例子，如下图所示，两段轨迹的分段点为 a ，初始时间分配为 t_1, t_2 ，比如说想减少前一段的时间 t_1 ，加大后一段的时间 t_2 ，其实就相当于把分段点 a 向右挪。也就是说，还是前段poly仍然分配 t_1 时间，但走的路程更多了（相当于原来路程所用的时间减少了），后一段poly在 t_2 时间走的路程更少了（相当于原来的路程所用时间增加了）。



<http://blog.csdn.net/q597967420>

所以如果corridor比较大的话，时间调整的空间就会更大，规划出的轨迹就会更平滑。

3. 归一化时间参数

轨迹规划中容易出现一些数值问题，比如需要求 t 的高阶指数，如果 t 比较大，计算数值就会很大，而且最终轨迹的参数会很小，特别是高阶项系数，比如很有可能小于float的精度，所以一般计算过程中最好用double。

但还有一种办法，是把轨迹参数做时间归一化，将时间归一化到 $[0, 1]$ 之间。设未归一化的轨迹为 $p(t) = p_0 + p_1 t + \dots + p_n t^n$ ，令时间 $\bar{t} = t/T$ ，则

$$\bar{p}(\bar{t}) = p(\bar{t}T) = p_0 + p_1 T \bar{t} + \dots + p_n T^n \bar{t}^n = \sum_{i=0}^n \bar{p}_i \bar{t}^i$$

其中， $\bar{p}_i = p_i * T^i$ ， T 为轨迹总时间， $[\bar{p}_i]_{i=0 \rightarrow n}$ 就是归一化时间的轨迹参数，它们不会像非归一化参数那么小，便于参数传递。

归一化参数如何求pva?

对于任一时刻 t ，先除以总时间 T 得到归一化时间 \bar{t} ，

$$\begin{aligned}
 p(t) &= \bar{p}(\bar{t}) = \sum_{i=0}^n \bar{p}_i \bar{t}^i \\
 v(t) &= \frac{d\bar{p}(\bar{t})}{dt} = \frac{d\bar{p}(\bar{t})}{d\bar{t}} \frac{d\bar{t}}{dt} = \bar{p}'(\bar{t})/T \\
 a(t) &= \bar{p}''(\bar{t})/T^2
 \end{aligned}
 \tag{2}$$

注意：归一化事件参数后，位置计算和原来相同，但速度计算需要除以 T ，加速度需要除以 T^2 ，即每求一阶导需要除以一个 T 。

实验过程中，用Matlab求解，如果一开始就用归一化时间参数，会解不出来，不知为何？所以，在求解过程中，仍然用的非归一化时间参数，而求解完以后，参数传递出去时，转换成归一化时间参数。

参考文献

1. Richter C, Bry A, Roy N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments[M]//Robotics Research. Springer International Publishing, 2016: 649-666.
2. Vijay Kumar的一系列论文：Mellinger D, Kumar V. Minimum snap trajectory generation and control for quadrotors[C]//Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011: 2520-2525.
3. HKUST 沈劲劼老师的工作：Chen J, Liu T, Shen S. Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments[C]//Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016: 1476-1483.