Fast Planner 学习记录

二零二三年三月

目录

[1 前端路径搜索 1](#_Toc132362160)

[1.1 A\*算法 1](#_Toc132362161)

[1.2 混合A\*算法 1](#_Toc132362162)

[1.3 Reeds-Shepp 曲线 2](#_Toc132362163)

[1.4 Hybrid A\*的计算流程 2](#_Toc132362164)

[2 B样条曲线 5](#_Toc132362165)

[2.1 公式原理 5](#_Toc132362166)

[2.1.1 形式一 5](#_Toc132362167)

[2.1.2 形式二（高飞讲解） 6](#_Toc132362168)

[2.1.3 用Toeplitz矩阵表示de Boor-Cox公式 6](#_Toc132362169)

[2.2 几何意义 6](#_Toc132362170)

[2.3 B样条分类 6](#_Toc132362171)

[2.3.1 均匀B样条曲线 6](#_Toc132362172)

[2.3.2 准均匀B样条曲线 7](#_Toc132362173)

[2.3.3 分段B样条曲线 7](#_Toc132362174)

[2.3.4 其他分类形式 7](#_Toc132362175)

[2.4 重要性质 8](#_Toc132362176)

[2.5 用矩阵形式表示德布尔-考克斯递推式 8](#_Toc132362177)

[3 控制点获得-前端路径拟合 9](#_Toc132362178)

[4 Lattice Planner 10](#_Toc132362179)

[4.1 Control Lattice Planning 10](#_Toc132362180)

[4.2 state lattice planner 11](#_Toc132362181)

[4.2.1 逆路径生成器 11](#_Toc132362182)

[4.2.2 样条插值法 11](#_Toc132362183)

[4.2.3 最优边界求解OBVP 12](#_Toc132362184)

[5 庞特里亚金原理求解两点边值问题 13](#_Toc132362185)

前端路径搜索

A\*算法

混合A\*算法

前端运动动力学路径搜索模块为混合A\*算法。该方法能够在栅格地图当中规划出一条安全的并且是符合动力学约束的一条可行轨迹。混合A\*算法就是将 A\*与Lattice Graph相关联。混合A\*算法继承了A\*算法启发式的特性。

A\*算法是一种在静态环境中求解最短路径的比较直接、有效的启发式搜索算法，即在整个状态空间中对每一个搜索节点按照制定的规则进行综合评估，通过当前节点与目标点之间的启发式剔除大量无关紧要的搜索，高效解算出一系列最优的扩展节点，同时A\*算法广泛应用于栅格地图中，具有较强的泛化能力。但是传统的 A\* 算法在搜索过程中通常是以栅格节点的中心为扩展点，生成的路径一般是分段式的折线，导致运动状态会发生突变，不具有连续性，因此不能满足车辆的非完整性约束，需要在相关限制条件下进行后续的路径优化。

Hybrid A\* 算法是 A\* 算法的变体，前者在节点的扩展细节上对后者进行了改进，在二维空间上的规划，A\*算法一般只考虑平面上的位置坐标关系，扩展时将当前节点的上、下、左、右、左上、右上、左下、右下八个节点作为候选节点，而 Hybrid A\* 算法会重点关注航向上的变化，通过加入车辆非完整性约束，根据限定的车辆转角去搜索周围节点，将自由度约束为车辆前进或倒退、左转或右转的组合情况，把节点的扩展限制在符合运动学特性的方向上，保证规划出的路径可供车辆行驶。

Reeds-Shepp 曲线

Reeds-Shepp 曲线是一段由曲率相同的圆弧与直线组成的曲线。有关研究已经证明，对于位姿空间内任意的两个状态，都存在一条Reeds-Shepp 曲线可以将两者连接起来，同时满足路径最短。Reeds-Shepp 曲线允许车辆进行前进和后退的运动。

Hybrid A\*的计算流程

Hybrid A\*算法的计算流程与传统A\*算法相似，该算法用Open Set维护已访问的待扩展节点，用Close Set维护筛选出的已扩展节点。从起点开始搜索，在每个节点扩展周期中，通过设计的评价函数计算出当前节点每个邻居节点的代价值，选择代价值最小从起点开始搜索，在每个节点扩 展周期中，通过设计的评价函数计算出当前节点每个邻居节点的代价值，选择代价值最小。

Hybrid A\*算法的基本思想与传统A\*相同,都提出了一种评价函数,在规划路径的过程中,比较各个待扩展节点的评价函数,优先选择函数值最小的节点作为下一步要扩展的路径节点,提高搜索效率。Hybrid A\* 算法的评价函数可以表示为:

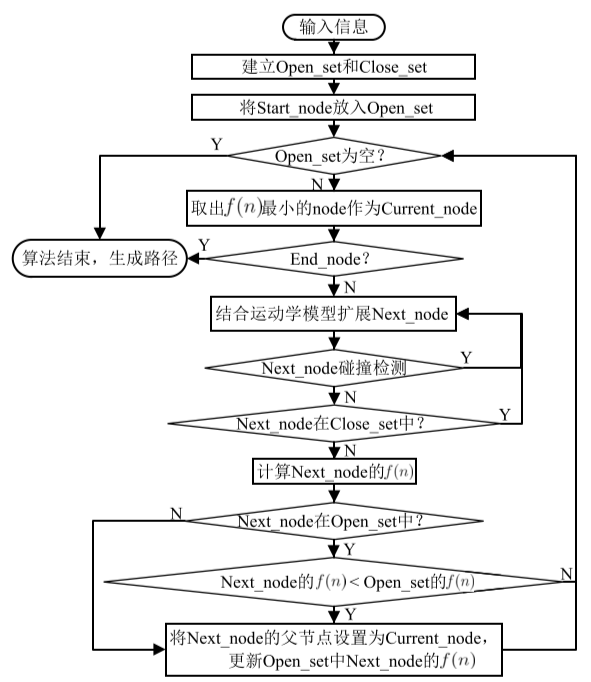


其中, 是节点n的评价函数, 是代价函数,其值为节点n与起点之间的实际代价, 是启发函数,其值为节点n与终点之间的估计代价。

与传统的A\*算法不同的是,Hybrid A\*算法的启发函数是由两种启发信息计算得到的，即



其中, 只考虑移动机器人的运动学约束,而不满足环境障碍约束。在计算其数值时,首先要假设移动机器人的行驶环境是无障碍的,接着利Reeds-Shepp曲线获取一条从当前节点到目标节点的最短路径,这条路径的长度就是该节点的启发函数值。与的意义恰恰相反,它只考虑环境障碍约束,而不考虑机器人的运动学约束。等于在障碍图中利用传统A\*算法搜索到的当前节点与目标节点的最短路径长度,这种启发信息的好处是可以发现所有的U形障碍和死胡同,并且引导机器人远离这些区域,避免陷入局部最优解。因此,该算法规划出的路径可以同时满足移动机器人的运动学约束与环境障碍约束.



最终，基于 Hybrid A\* 的算法流程如下:

1) 初始化 OpenList、CloseList，获得起始点位姿、目标点位姿、车辆运动学模型；

2) 从 OpenList 中选取具有最小花费的节点，并计算该节点到目标点的 Reeds-Shepp 曲线；

3) 若Reeds-Shepp 曲线不予任何障碍物相交，即满足障碍物约束，则将该曲线作为当前节点与目标点之间的路径，输出连接起始点与目标点的整条路径，算法结束；否则，继续；

4) 对 𝑣𝑙，𝑣𝑟 进行采样，得到该节点不在 CloseList 中且满足障碍物约束的子节点。

5) 计算子节点的花费 𝑓 = 𝑔 + ℎ，并调整节点间的连接顺序。

6) 若子结点中包含目标点，则输出路径，结束算法；否则，继续。

7) 将该节点加入 CloseList 中，将所有子节点加入 OpenList，回到步骤 2。

至此，基于Hybrid A\*算法的路径规划算法已经给出。利用该算法可以获得相邻任务点之间满足障碍物约束的粗糙路径。

B样条曲线

样条是通过一组指定的点集而生成平滑曲线的柔性带。 简单地说，B 样条曲线就是通过控制点局部控制形状的曲线。B样条曲线是B样条基函数（给定区间上的所有样条函数组成一个线性空间）的线性组合。

3阶B样条每一段轨迹需要4个控制点，第一段轨迹需要控制点以及时间为，每增加一段轨迹就需要增加一个控制点和时间，且改变和只改变最后一段轨迹。

公式原理

形式一

设有 一共个控制点用来生成段轨迹，这些控制点用于定义样条曲线的走向、界限范围，则具有个控制点的 阶B样条曲线的定义为:



式中，是自变量，是第 个阶B样条基函数，与控制点相对应， 。

基函数为德布尔-考克斯递推式:



式中，如果遇到分母为 0且分子也为0的情况，约定这一项整体为0；如果此时分子不为0，则约定分母为1 。​是一组被称为节点矢量的非递减序列的连续变化值，首末值一般定义为 0 和 1 ，该序列如下:



阶B样条是关于的 次曲线，即基函数的次数为；

段数为控制点个数减去次数即。

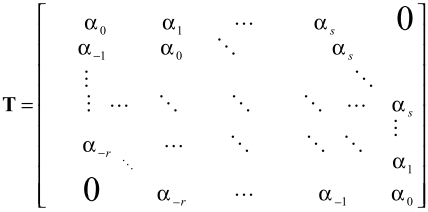
涉及到的节点为共个节点，个区间，因此从到共涉及个节点。

形式二（高飞讲解）

B样条曲线是一条多项式曲线，阶数、个控制点以及节点向量，并且定义，一个B样条曲线的时间为，而不是内。

用Toeplitz矩阵表示de Boor-Cox公式

Toeplitz矩阵是其在平行于主对角线的任何线上的元素全部相等的矩阵。Toeplitz矩阵定义如下：

几何意义

B样条分类

B样条分类可以根据节点 的取值分为均匀B样条曲线、准均匀B样条曲线和分段B样条曲线。

均匀B样条曲线

当节点沿参数轴均匀等距分布， 为均匀B样条曲线，如。当和一定时，均匀B样条的基函数呈周期性，所有基函数有相同形状，每个后续基函数仅仅是前面基函数在新位置上的重复。

准均匀B样条曲线

其节点矢量中两端节点具有重复度（即样条的阶数），所有的内节点均匀分布，具有重复度1，如。

准均匀B样条曲线保留了贝塞尔曲线在两个端点处的性质：样条曲线在端点处的切线即为倒数两个端点的连线。 准均匀B样条曲线用途最为广泛。

一般来说，次数越高，则曲线的导数次数也会较高，那么将会有很多零点存在，较多的导数零点就导致原曲线存在较多的极值，使曲线出现较多的峰谷值；次数越低，样条曲线逼近控制点效果越好。

另一方面，三次B样条曲线能够实现二阶导数连续，故最终选择准均匀三次B样条曲线作为轨迹规划的曲线比较合适。

分段B样条曲线

其节点矢量中两端节点的重复度与准均匀B样条曲线相同，为。不同的是内节点（即除去两端节点后的剩余中间节点）重复度为，该类型有限制条件，控制顶点点数减1等于次数的正整数倍，即为正整数。

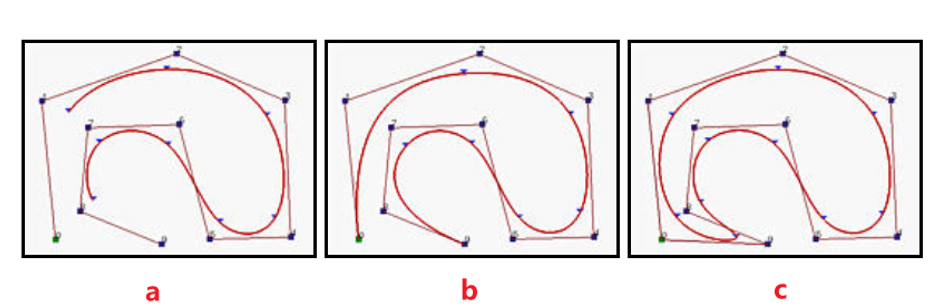
其他分类形式

许多论文中的分类是open 、clamped、closed。

如果节点向量没有任何特别的结构，那么产生的曲线不会与控制曲线的第一边和最后一边接触，曲线也不会分别与第一个控制点和最后一个控制点的第一边和最后一边相切。如下面图a所示。这种类型的B样条曲线称为开(open)B样条曲线。对于开(open)B样条曲线， u的定义域是。

clamped B-样条曲线即准均匀B样条曲线，如下图b。

通过重复某些节点和控制点，产生的曲线会是 闭(closed)曲线。 这种情况，产生的曲线的开始和结尾连接在一起形成了一个闭环如下边图c所示。



重要性质

一个有个控制点的 ​次B样条曲线，则其一共有时间节点，即 。同样的，假定希望设计一条 ​次B样条曲线，且具备个时间节点，则相应的控制点数量应为：。

这样一条B样条的定义域为。

每一个控制点​的作用域为 

该B样条曲线的重的一段曲线 只被这个控制点影响。

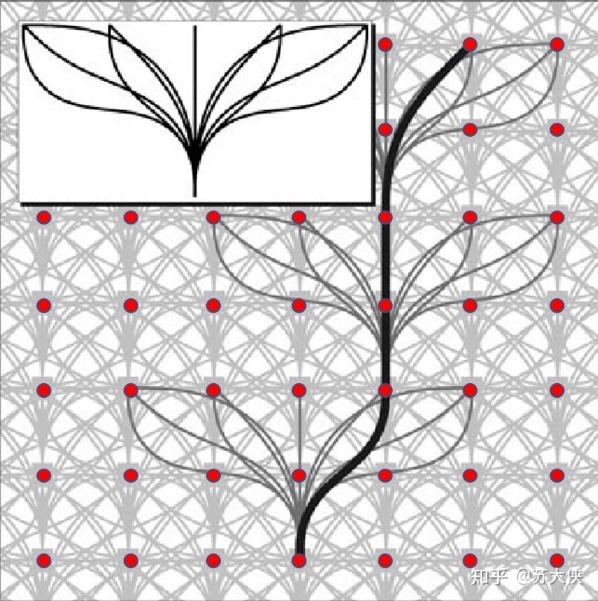
用矩阵形式表示德布尔-考克斯递推式

控制点获得-前端路径拟合

首先假设获得的离散轨迹点一共有K个，则有K-1段轨迹，根据B样条性质，这K-1段3次B样条曲线的定义域是。则一共有K+5个knot vector，即 ，所以应该有即个控制点。

Lattice Planner

使用晶格(lattice) 的目的就是为了简化多个网格(grid)，晶格是网格的一般化，使得原来的网格图演化成一个新的晶格图，如Fig.1[2]所示，红色点标出来的就是晶格，而连接他们之间的边则表示一个状态到另一个状态的转移，Fig.2中最浅的灰色部分就是我们建立完毕的 Lattice 地图。黑色实线和黑色加粗线代表什么，以及这个地图是如何生成的，详看后文，这里只需要记住 Lattice planning 就是在自己建立的新的 Lattice 地图上进行路径规划。Lattice Planner主要分为在控制空间上进行晶格生成和在状态空间上进行晶格生成，称为 Control Lattice Planner(CLP), State Lattice Planner(SLP)。



Control Lattice Planning

由于控制空间的 Lattice Plan 比较简单，本文就简要带过，主要讲述 State Lattice Planning.

设机器人状态方程如下：



其中， 为状态变量，为输入，矩阵是状态转移矩阵，矩阵是控制矩阵。我们固定输入量和时间，然后我们进行前向积分，就可以知道后的机器人状态，如Fig.2 所示。尤其是当机器人是线性系统的时候。在知道时刻机器人的末状态之后，就可以通过状态转移方程：



来还原出过程之中的各个状态。假设，此时我们想要求解，但的求解是一件很麻烦的事，我们可以利用泰勒展开



而且此时我们知道，矩阵有一个性质，就是nilpotent，也就是说它在多次方之后会变为0，那高次项就都被消去了，这使得在次矩阵多项式中都具有封闭的表达式。

但这种状态建图法有一点缺陷，和Dijkstra一样，比较盲目，你不知道自己是要往左边还是往右边，只是不断的尝试各种输入量的积分，所以导致在规划过程中的效率不够高。所以这时我们就提到我们这篇论文的核心部分，state lattice planner.

state lattice planner

我们可以选择在状态空间采样，我直接在空间中采样一个机器人状态（包括位置与姿态），然后反解出路径，即机器人如何选定一个在T上积分，刚好可以达到我们采样的这个状态。优点显而易见，这使得算法具有启发性和贪心性，难点在于难以实现。

逆路径生成器

在 State Lattice中，我们需要根据采样点与当前点的状态在他们之间生成符合运动学和车辆约束的路径。这个问题被称为Boundary Value Problem(BVP)，它是状态抽样格规划的基础。由于没有通解，它往往演变成一个复杂的数值优化问题。

样条插值法

样条插值是一种以可变样条来作出一条经过一系列点的光滑曲线的数学方法。插值样条是由一些多项式组成的，每一个多项式都是由相邻的两个数据点决定的，这样任意的两个相邻的多项式以及它们的导数在连接点处都是连续的。

最优边界求解OBVP

在BVP问题中加入一个优化目标，确保我们生成的状态曲线更加符合我们的工程实践，比如我们定一个Objective，要使得加速度的平方和最小，而且状态转移的过程要尽可能的快，如下所示：



状态量，输入量和系统方程表征如下：



边界条件表征如下



庞特里亚金原理求解两点边值问题