

无人机路径规划的HJB-RRT算法

张清江¹,王东辉²,左玲²,王苗苗³,吕宾¹,孙小礼¹,郑华¹,李俨^{1†}

1. 西北工业大学, 西安 710072

E-mail: liyan@nwpu.edu.cn

2. 飞行器控制一体化技术重点实验室, 航空工业自控所, 西安 710065

3. 西安华为技术有限公司, 西安 710075

摘要: 近年来, 基于快速扩展随机树(rapidly-exploring random tree, RRT)算法的无人机路径规划方法得到了广泛的关注和研究。其中,运动方程约束下扩展点之间的快速连接问题是目前RRT算法相关研究中的难点。针对此问题,论文提出了一种基于Hamilton-Jacobi-Bellman(HJB)方程的RRT 算法,实现了运动方程约束下RRT算法扩展点之间的快速连接。论文首先根据最优控制理论, 将运动方程约束下任意两扩展点之间的连接问题直接转化为HJB方程的求解问题。然后使用semi-Lagrangian 数值方法, 求解了HJB方程的值函数, 并以此实现了任意扩展点之间的快速连接。最后,通过讨论连接点之间的位置关系, 利用坐标变换进一步缩小了HJB方程的求解区间, 降低了HJB 方程的离线计算负担。通过对比实验, 验证了所提算法的高效性。

关键词: RRT算法; 两点边值问题; HJB方程; semi-Lagrangian数值方法

A HJB-RRT Algorithm for UAV Path Planning

Qingjiang Zhang¹, Donghui Wang², Ling Zuo², Miaomiao Wang³, Bin Lv¹, Xiaoli Sun¹, Hua Zheng¹, Yan Li^{1†}

1. Northwestern Polytechnical University, Xi'an 710072, P. R. China

E-mail: liyan@nwpu.edu.cn

2. Science and Tchenology on Aircraft Control Laboratory, FACRI, Xi'an 710065, P. R. China

3. Xi 'an Huawei Technology Co. LTD, Xi'an 710075, P. R. China

Abstract: The UAV path planning method based on rapidly-exploring random tree (RRT) has been widely studied in recent years. Realizing fast connection between extended points under kinematic constraints is the main difficulty in the current research of RRT algorithm. In this paper, a new RRT algorithm based on Hamilton-Jacobi-Bellman (HJB) equation is proposed to realize fast connections between the extended points under kinematic constraints. Firstly, based on the optimal control theory, connecting any two expansion points under the kinematic constraint is directly transformed into the problem of solving the corresponding HJB equation. Then the value function of HJB equation is solved using the semi-Lagrangian method, and thus the fast connection between arbitrary extension points is achieved. Finally, by analyzing the relationship between the connecting points, the solving domain of the HJB equation is further reduced by using coordinate transformation. The efficiency of the proposed algorithm is verified using numerical simulations.

Key Words: RRT algorithm; TPBVP; HJB equation; semi-Lagrangian numerical method

引言

传统的路径规划算法如遗传算法、人工势场法、蚁群优化算法等, 一般都不能很好的解决运动学和动力学约束下的路径规划问题。而对于无人机来说, 规划的路径必须满足无人机自身的运动学和动力学约束, 这就限制了这些算法在无人机路径规划中的应用^[1, 2]。

基于快速扩展随机树^[3](rapidly-exploring random tree, RRT)的路径规划算法, 通过对状态空间中的采样点进行碰撞检测, 避免了对空间的建模, 能够有效地解决高维空间中和复杂约束下的路径规划问题^[1-4]。运动方程约束下的RRT算法, 一般需要求解复杂的两点边值问题(two-point boundary value problem, TPBVP), 以实现扩展点之间的精确连接^[5, 6]。在这些算法中, 计算资源主要消耗在TPBVP的求解中, 这也大大影响了RRT算法的效率。

为了解决RRT算法扩展过程中的TPBVP, 国内外学者对RRT算法进行了不断的改进, 主要分为两类: 一类是直接求解RRT算法扩展过程中的TPBVP, 另一类是通过间接方法, 避免求解TPBVP来完成RRT算法中的扩展问题。

对于第一类改进, 文献^[7]将打靶法用在RRT算法中, 用于求解TPBVP。但打靶法的基本原理是将边值问题转化为初值问题来求解, 初值的选取过于依赖经验, 计算量大, 同时收敛性较差, 不利于实际问题中使用。文献^[8-10]都将微分平滑(Differential Flatness)用于RRT算法扩展过程中TPBVP的求解。但微分平滑的关键在于求解一个平滑的输出来表达状态和输入的轨迹, 该平滑输出的求解比较困难, 目前没有通用的求解办法。文献^[11]基于Clothoid 曲线实现了运动方程约束下的路径规划, 但是这种方法只能针对特定的运动方程。

对于第二类改进, 文献^[12]采用随机输入驱动的方法求解RRT算法中扩展点之间的连接问题, 虽然没有

†此项工作得到中国航空科学基金资助, 项目编号:ASFC-20165853040。

求解TPBVP问题,但随机驱动导致规划路径的随机性,一般得不到最优路径。文献^[6]利用一种简单的离线学习回归模型来近似距离测度,从而得到了RRT算法的扩展路径。文献^[12]将两个状态之间的最优代价函数通过它们之间的A*路径来近似,得到了RRT算法的扩展路径。这两种方法虽然避免了RRT算法中TPBVP的求解,但都对RRT算法中代价函数进行了近似,不能保证路径的最优性。

当存在运动方程约束时,RRT算法中一般要求解复杂的TPBVP。针对此问题,论文提出了一种基于Hamilton-Jacobi-Bellman(HJB)方程的RRT算法(以下简称HJB-RRT算法)。论文的主要创新点如下:

- 1) 使用HJB方程解决了运动方程约束下的RRT算法的快速扩展问题,避免了复杂的TPBVP的求解过程,在保证扩展路径最优性的前提下降低了RRT算法扩展过程的计算负担。
- 2) 通过讨论连接点之间的位置关系,利用坐标变换缩小了HJB方程的求解区间,降低了HJB方程的离线计算负担。

论文进一步进行了仿真对比实验,通过与随机扩展RRT算法的对比,验证了该算法的高效性。

1 无人机运动学模型

无人机的运动方程可以表述为^[13, 14]

$$\begin{cases} \dot{x}(t) = v(t) \cos \mu(t) \cos \psi(t) \\ \dot{y}(t) = v(t) \cos \mu(t) \sin \psi(t) \\ \dot{z}(t) = v(t) \sin \mu(t) \\ \dot{v}(t) = g(n_x - \sin \mu(t)) \\ \dot{\mu}(t) = g(n_z \cos \varphi(t) - \cos \mu(t)) / v(t) \\ \dot{\psi}(t) = g n_z \sin \varphi(t) / (v(t) \cos \mu(t)) \end{cases} \quad (1)$$

在这里, (x, y, z) 是无人机惯性坐标系下的坐标, v 是真空速, μ 是无人机的航迹倾斜角, ψ 是无人机的偏航角, φ 是滚转角, g 是重力加速度, n_x 是纵向负载因子, n_z 是法向负载因子。控制输入为 n_x, n_z 与滚转角 φ 。假设飞行中可以消除侧滑角,因此在模型中忽略了侧滑影响。

假设无人机在执行任务过程中定高飞行,那么飞行的航迹角 $\mu = 0$, 而且 $\dot{\mu} = 0$ 。因此可以得到

$$n_z \cos \varphi(t) = 1 \quad (2)$$

对于式(1)系统,可以得到

$$\dot{\psi}(t) = \frac{g \tan \varphi}{v} \quad (3)$$

假设无人机速度的上限和下限为 $v^{\min} \leq v(t) \leq v^{\max}$, 而且滚转角输入界限为 $|\varphi| \leq \varphi^{\max}$ 。那么在无人机航迹上处处可以取到的转弯角速率为

$$-\frac{g \tan \varphi^{\max}}{v^{\max}} \leq \dot{\psi}(t) \leq \frac{g \tan \varphi^{\max}}{v^{\max}} \quad (4)$$

无人机速度和转弯角速率的限制可以简化为

$$\begin{cases} g n_x^{\min} \leq \dot{v}(t) \leq g n_x^{\max} \\ |\dot{\psi}(t)| \leq \frac{g \tan \varphi^{\max}}{v^{\max}} \end{cases} \quad (5)$$

由于 $\dot{z}(t) = 0, \dot{\mu}(t) = 0$, 则无人机的运动方程可以简化为

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos \psi(t) \\ v(t) \sin \psi(t) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (6)$$

在这里, u_1 和 u_2 分别是加速度和角速率输入,且输入的范围为 $g n_x^{\min} \leq u_1(t) \leq g n_x^{\max}$ 和 $|u_2(t)| \leq \frac{g \tan \varphi^{\max}}{v^{\max}}$ 。

注解:虽然对于无人机模型做出了简化,但是针对简化模型规划出的航迹对于原模型所代表的无人机而言任然是可行的,因为简化假设中并没有超出无人机的输入限制和可达到的状态。换言之,上述过程是对无人机进行了保守性的简化,简化模型的飞行状态都在原状态的包线内。

2 RRT算法

RRT算法是一种高效且可扩展性强的路径规划算法。RRT算法的主要步骤为,以规划空间起始点 \mathbf{x}_{init} 为根节点,通过随机采样逐渐增加子节点来生成随机扩展树,当随机树的子节点包含目标点 \mathbf{x}_{goal} 或处于目标点所在的子区域中时,随机扩展树构建结束。RRT算法如图1所示

Algorithm 1: RRT_Expansion

Input: Initial and final points $\mathbf{x}_{init}, \mathbf{x}_{goal}$, obstacle parameter

Output: RRT tree T

```

1  $T.Root \leftarrow \mathbf{x}_{init}$ ;
2 while  $\mathbf{x}_{new} \neq \mathbf{x}_{goal} \vee \mathbf{x}_{new} \notin \text{Area}(\mathbf{x}_{goal})$  do
3    $\mathbf{x}_{rand} \leftarrow \text{Rand}(T, \mathbf{X}_{free})$ ;
4    $\mathbf{x}_{near} \leftarrow \text{Near}(T, \mathbf{x}_{rand})$ ;
5    $\mathbf{x}_{temp} \leftarrow \text{Newnode}(\mathbf{x}_{near}, \mathbf{x}_{rand}, \mathbf{u})$ ;
6   if  $\text{collision}(\mathbf{x}_{temp}) == \text{false}$  then
7      $\mathbf{x}_{new} \leftarrow \mathbf{x}_{temp}$ ;
8      $T.Addnode(\mathbf{x}_{new})$ ;
9      $T.Addedge(\mathbf{x}_{near}, \mathbf{x}_{new})$ ;
10  end
11 end
12 return  $T$ ;
```

图1 RRT算法

算法中, T 为待扩展的随机树, \mathbf{x}_{init} 为路径规划的起始状态, \mathbf{X}_{free} 为规划空间中无障碍物的区域, $\text{Area}(\mathbf{x}_{goal})$ 为以 \mathbf{x}_{goal} 为圆心的子区域。

在随机树的生长过程中,初始时树上只有一个节点:根节点 \mathbf{x}_{init} 。首先Rand函数在状态空间中随机采样一个点 \mathbf{x}_{rand} ,然后Near函数找到树上距离 \mathbf{x}_{rand} 最近的点 \mathbf{x}_{near} ,最后Newnode函数通过随机驱动 \mathbf{u} 使 \mathbf{x}_{near} 向 \mathbf{x}_{rand} 扩展,得到最接近 \mathbf{x}_{rand} 的临时节点 \mathbf{x}_{temp} 。如果 \mathbf{x}_{temp} 没有与障碍物发生碰撞,将 \mathbf{x}_{temp} 加入到树中,否则放弃此次生长,继续进行下一次的生长,直到满足 $\mathbf{x}_{new} = \mathbf{x}_{goal}$ 或 $\mathbf{x}_{new} \in \text{Area}(\mathbf{x}_{goal})$ 。然后从目标点出发,找到其父亲节点,依次向上进行寻找,直到到达根节点。这样就规划出一条从起始状态点到目标状态点的路径,以及在每一时刻的控制输入参数^[1]。

3 无人机路径规划的HJB-RRT算法

在RRT树的构建过程中, \mathbf{x}_{near} 向 \mathbf{x}_{rand} 的扩展是通过随机驱动 \mathbf{u} 实现的^[2],得到最接近 \mathbf{x}_{rand} 的临时节

点 \mathbf{x}_{temp} , 这样就可能导致 \mathbf{x}_{temp} 不能精确的到达随机采样点 \mathbf{x}_{rand} , 而要精确到达 \mathbf{x}_{rand} , \mathbf{x}_{near} 向 \mathbf{x}_{rand} 的扩展就需要求解TPBVP^[5, 6]。

扩展点之间的连接问题是影响RRT算法性能的关键。随机输入连接方法速度快, 但忽略了路径的最优性; 基于TPBVP的连接方法一般计算精度高, 然而计算负担繁重。此外, TPBVP求解的数值稳定性较差, 甚至会出现难以收敛的情形。为了简化扩展点的连接问题并保证数值稳定性, 论文首先根据最优控制理论, 将运动方程约束下任意两扩展点之间的连接问题, 直接转化为HJB方程的求解问题。然后使用semi-Lagrangian数值方法, 求解了HJB方程的值函数, 并以此实现了任意扩展点之间的快速连接。最后, 通过讨论连接点之间的位置关系, 利用坐标变换进一步缩小了HJB方程的求解区间, 降低了HJB方程的离线计算负担。

3.1 HJB方程

HJB(Hamilton-Jacobi-Bellman)方程是一个偏微分方程, 近年来, HJB方程的数值解法得到了广泛的研究。不失一般性地, 可以将非线性系统(6)表述为

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t) \quad (7)$$

$\mathbf{x}(t_0) = \mathbf{x}_0$ 为系统的初始状态。

对于时间最优的路径规划问题, 值函数 $V(\mathbf{x}, t)$ 可以定义为

$$V(\mathbf{x}(t), t) = \inf_{\mathbf{u}_i \in \mathbf{u}} \{t | \mathbf{x}(t) = \mathbf{x}_f\} \quad (8)$$

这里 \mathbf{x}_f 是无人机的终点状态。

根据动态规划理论, 可得

$$V(\mathbf{x}(t), t) = \inf_{\mathbf{u}_i \in \mathbf{u}} \{\Delta t + V(\mathbf{x}(t + \Delta t), t + \Delta t)\} \quad (9)$$

将 $V(\mathbf{x}(t), t)$ 进行泰勒级数展开, 除以 Δt , 然后求 $\Delta t \rightarrow 0$ 时的极限就可以得到路径规划问题的HJB方程

$$-1 = \min_{\mathbf{u}_i \in \mathbf{u}} \left\{ \left[\frac{\partial V(\mathbf{x}, t)}{\partial \mathbf{x}} \right]^T f(\mathbf{x}, \mathbf{u}, t) \right\} \quad (10)$$

这里, 偏导数 $\frac{\partial V(\mathbf{x}, t)}{\partial \mathbf{x}}$ 可以使用中心差分法来进行近似。

3.2 semi-Lagrangian方法

相比于有限差分法等数值解法, semi-Lagrangian方法在数值稳定性和收敛性上具有一定优势。对于值函数不连续的HJB方程, semi-Lagrangian方法可以保证收敛到HJB方程的粘性解^[15, 16]。

具体来讲, semi-Lagrangian方法的思想是在结构化的网格上对动态规划理论进行离散化处理^[17, 18]。其中, semi-Lagrangian 离散定义为

$$V(\mathbf{x}_i) \approx \hat{V}(\mathbf{x}_i) = \min_{\mathbf{u}_i \in \mathbf{u}} \{ \hat{V}(\mathbf{x}_i + \Delta \mathbf{x}) + \Delta t \} \quad (11)$$

在这里 $V(\mathbf{x}_i)$ 是网格点 \mathbf{x}_i 处的值函数的真实值, $\hat{V}(\mathbf{x}_i)$ 是网格点处的迭代值。其中, $\hat{V}(\mathbf{x}_i + \Delta \mathbf{x})$ 可以通过插值得

到, 而 $\Delta \mathbf{x}$ 通过在时间段 Δt 内使用控制输入 \mathbf{u} 驱动动态系统得到。

在semi-Lagrangian方法中, HJB方程的值函数可以通过下式进行更新

$$\hat{V}^n(\mathbf{x}_i) = \min_{\mathbf{u}_i} \{ \hat{V}^c(\mathbf{x}_i), \hat{V}(\mathbf{x}_i + \Delta \mathbf{x}) + \Delta t \} \quad (12)$$

这里 $\hat{V}^c(\mathbf{x}_i)$ 是状态 \mathbf{x}_i 处的当前值, 而 $\hat{V}^n(\mathbf{x}_i)$ 是迭代后的新值。

在解算过程中, 给定一个迭代结束的标准 ξ , 则迭代结束的条件为

$$\sum |\hat{V}^c(\mathbf{x}_i) - \hat{V}^n(\mathbf{x}_i)| < \xi \quad (13)$$

在semi-Lagrangian求解HJB方程的值函数时, 首先确定求解区域大小, 确定各个方向的网格宽度。然后根据选定的值函数对网格进行初始化, 终点坐标的一个小区域内的所有点的值函数都设置为0, 其他点的值函数设置为一个较大的数。最后利用式(12)进行值函数的迭代更新。文献^[19]给出了数值迭代的详细过程。

在解算出HJB方程的值函数后, 可以进行最优路径的求解。通过欧拉方法来计算起始状态 \mathbf{x}_s 到终止状态 \mathbf{x}_f 的最优路径

$$\mathbf{x}^n = \mathbf{x}^c + \dot{\mathbf{x}}^c \Delta t_s \quad (14)$$

这里 \mathbf{x}^c 是当前状态, Δt_s 是时间步长, \mathbf{x}^n 是移动一个时间步长之后的状态。最优路径的计算可以通过式(14)运算, 直到抵达终点。

3.3 值函数求解区域的缩小

影响RRT算法性能的因素主要有: 产生随机叶点的数量、随机树扩展的步长、产生随机点的规律分布以及树的数量^[20]。

本文中, 在HJB-RRT算法每一步扩展中求出 \mathbf{x}_{rand} 和 \mathbf{x}_{near} 之后, 首先要计算这两点之间的距离, 如果距离太大, 以一定的步长 δ_1 来截断 \mathbf{x}_{near} 到 \mathbf{x}_{rand} 之间的距离, 如果距离太小, 以步长 δ_2 来扩展 \mathbf{x}_{near} 到 \mathbf{x}_{rand} 之间的距离。HJB-RRT算法扩展和步长的选择如图2所示

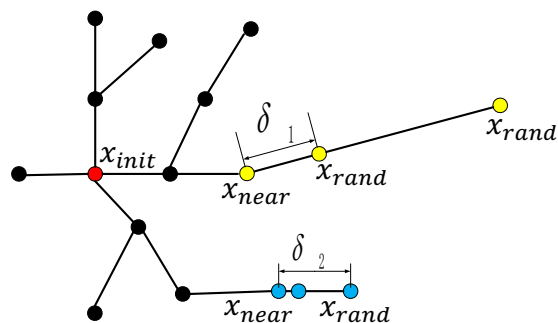


图2 HJB-RRT算法扩展和步长的选择

在RRT算法中, 采样点可以位于规划区域的任何地方, 这就要求使用semi-Lagrangian方法求解HJB方程的值函数时, 终点位置必须选在区域的中心, 这样会导致求解值函数的区间过大。本文根据规划路径的对称性, 将路径翻转用在HJB-RRT算法的扩展中, 就可以缩小HJB方程值函数的求解区间。

设随机采样点 \mathbf{x}_{rand} 的位置坐标为 (x_t, y_t) , RRT树上距离随机采样点最近的点 \mathbf{x}_{near} 的位置坐标为 (x_o, y_o) 。在二维空间中, \mathbf{x}_{rand} 和 \mathbf{x}_{near} 的相对位置有四种情况, 分别为 \mathbf{x}_{rand} 在 \mathbf{x}_{near} 的左上方($x_o > x_t, y_o < y_t$), 左下方($x_o \geq x_t, y_o \geq y_t$), 右上方($x_o \leq x_t, y_o \leq y_t$)和右下方($x_o < x_t, y_o > y_t$)。

以下分析都假设semi-Lagrangian求解值函数时终点选在区域的右上方位置。路径翻转的基本思想是:在RRT算法的每一步扩展中, 如果 \mathbf{x}_{rand} 在 \mathbf{x}_{near} 的右上方, 不需要翻转,直接积分求解扩展路径; 如果 \mathbf{x}_{rand} 不在 \mathbf{x}_{near} 的右上方, 利用路径对称性将 \mathbf{x}_{rand} 翻转到 \mathbf{x}_{near} 的右上方的位置, 分别对应于 \mathbf{x}_{rand1} 和 \mathbf{x}_{near1} , 然后积分求解 \mathbf{x}_{near1} 到 \mathbf{x}_{rand1} 的扩展路径, 最后再将这些路径点反变换回对应于初始状态 \mathbf{x}_{rand} 和 \mathbf{x}_{near} 的路径点。这样,一步扩展就完成了。RRT算法的每一步扩展都采用上述的方法,直到到达目标点。

翻转过程中x坐标和y坐标的变换可以利用简单的几何关系得到, 对称轴和角度的变换分别如表 1 和表 2 所示。翻转的基本思路如上所述, 具体的做法可以根据semi-Lagrangian求解值函数时终点的选取位置不同进行简单的调整。图 3 给出了 \mathbf{x}_{rand} 在 \mathbf{x}_{near} 的左下方的路径翻转示意图, 左上方和右下方的图可以类比得到。

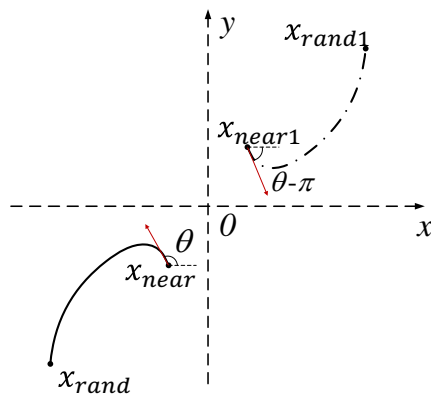


图 3 路径翻转

综上所述, HJB-RRT算法的主要步骤如下:

Step 1 根据最优控制理论,将运动方程约束下的RRT算法中的扩展问题转换为HJB方程的求解;

Step 2 通过semi-Lagrangian数值方法求解HJB方程的值函数并存储;

Step 3 在值函数的基础上求解RRT算法每一步的扩展路径;

Step 4 判断目标状态有没有在RRT扩展树的节点中, 如果有, 算法结束; 如果没有, 重复Step3直到RRT树的节点中包含目标状态或在目标状态的子区域中。

4 仿真实验

实验在一台3.0GHz,8G内存的计算机上运行, 编程环境为Matlab 2015b, 使用Fortran 6求解值函数。由于不同扩展方法之间计算方式差异较大, 论文选取运行时间为基准, 对比了在相同运行时间下不同算法的效果。

表 1: 对称轴

\mathbf{x}_{rand} 、 \mathbf{x}_{near} 位置关系	对称轴/点
\mathbf{x}_{rand} 在 \mathbf{x}_{near} 右下方	X
\mathbf{x}_{rand} 在 \mathbf{x}_{near} 左上方	Y
\mathbf{x}_{rand} 在 \mathbf{x}_{near} 左下方	原点

表 2: 角度变换

\mathbf{x}_{rand} 、 \mathbf{x}_{near} 位置关系	\mathbf{x}_{near} 处角度	\mathbf{x}_{near1} 处角度
\mathbf{x}_{rand} 在 \mathbf{x}_{near} 右下方	θ	$-\theta$
\mathbf{x}_{rand} 在 \mathbf{x}_{near} 左上方	θ	$\pi - \theta$
\mathbf{x}_{rand} 在 \mathbf{x}_{near} 左下方	θ	$\theta - \pi$

使用如式(6)所示的非线性系统, 本文选取 $|u_1| \leq 30m/s^2, |u_2| \leq 1rad/s$ 。

选取1000m×1000m的区域来迭代求解值函数, x和y的坐标范围均为[0,1000m], ψ 的坐标范围为[0,2 π], v的变化范围为[60m/s,90m/s], x 和y 方向的网格宽度均为10m, ψ 方向的网格宽度为 $\pi/60$, v方向的网格宽度为2m/s, 这样值函数就离散为 $100 \times 100 \times 120 \times 15$ 的四维张量V, 终止点坐标选为(800m,800m)。

由于TPBVP数值解法不稳定, 会出现不能收敛的情况, 所以实验对比了随机驱动RRT算法, 随机驱动RRT算法采用随机控制输入驱动来完成每一步的扩展^[2]。

通过三个不同的仿真环境进行了两算法的对比。HJB-RRT算法中步长最小为300m, 最大为600m, 程序均运行60s。

仿真环境一: 初始点坐标为(200m, 200m), 初始角度为 πrad , 初始速度为80m/s, 终止点坐标为(1550m,1550m)。

仿真环境二: 初始点坐标为(200m, 200m), 初始角度为 πrad , 初始速度为80m/s, 终止点坐标为(1700m,1400m)。

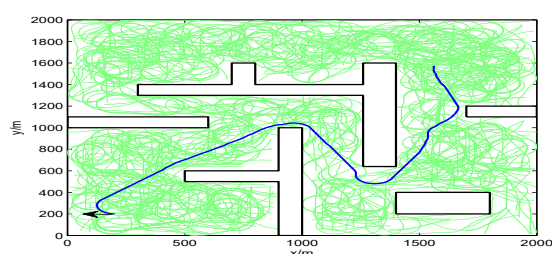
仿真环境三: 初始点坐标为(1700m, 200m), 初始角度为0rad, 初始速度为80m/s, 终止点坐标为(400m,1800m)。

HJB-RRT算法和随机驱动RRT算法规划的路径如图 4 所示, 路径长度对比如表 3 所示

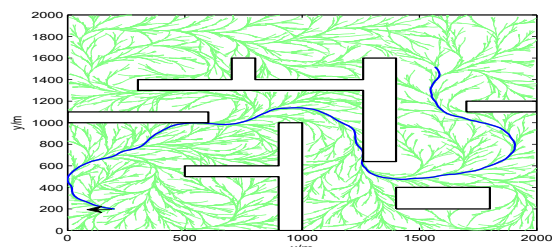
表 3 路径长度对比

	HJB-RRT	RRT
仿真环境一	3204m	4080m
仿真环境二	2530m	2568m
仿真环境三	2424m	4800m

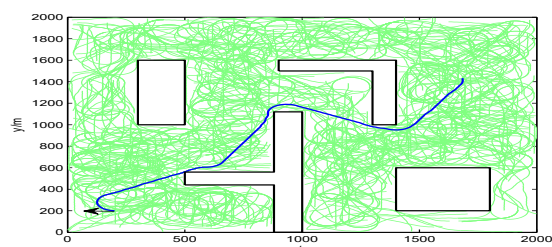
通过图 4 和表 3 的数据以及对随机驱动RRT算法和HJB-RRT算法的分析, 我们可以得到以下结论:



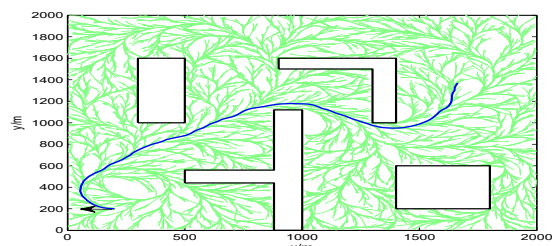
(a) 仿真环境一 HJB-RRT



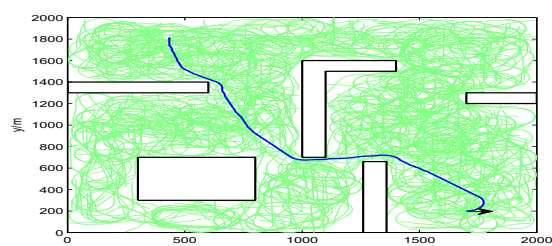
(b) 仿真环境一 RRT



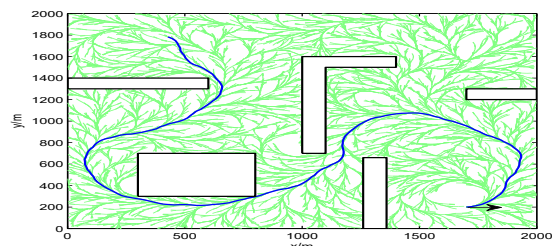
(c) 仿真环境二 HJB-RRT



(d) 仿真环境二 RRT



(e) 仿真环境三 HJB-RRT



(f) 仿真环境三 RRT

图4 简化固定翼无人机模型的规划路径

1) 对于简化的无人机模型,在保证规划时间相同的情况下, HJB-RRT算法可以获得更短的规划路径。

2) HJB-RRT算法的扩展路径是平滑的, 有助于实际研究中的路径跟踪。而随机驱动RRT算法, 两个扩展点之间的路径是通过直线连接的, 不能完全保证扩展路径的平滑性。

3) 通过HJB-RRT算法的分析可知, 当扩展的两个状态点确定后, 采用HJB-RRT算法可以得到最优的扩展路径。而随机驱动RRT算法采用随机输入驱动的方式来完成扩展, 当随机输入不同的时候, 虽然扩展的两个状态点相同, 扩展的路径也不相同, 存在很大的不确定性。也就是说HJB-RRT算法的两扩展点之间扩展路径是随机驱动RRT算法在最优控制输入驱动下的路径。

4) HJB-RRT算法将扩展过程中的TPBVP转化为HJB方程的求解问题, 可以保证计算精度和数值稳定性。而随机驱动RRT算法, 如果采用随机输入驱动的方式来完成扩展, 不能准确的到达随机采样点, 如果要精确到达随机采样点又需要求解TPBVP, 但是会增加扩展过程的计算负担。

综上所述, HJB-RRT算法在路径长度、路径平滑和路径精度方面都优于随机驱动RRT算法。

5 结论

论文提出了一种基于HJB方程的RRT算法, 实现了运动方程约束下RRT算法扩展点之间的快速连接。首先根据最优控制理论, 将运动方程约束下任意两扩展点之间的连接问题, 直接转化为HJB方程的求解问题。然后通过semi-Lagrangian数值方法求解了HJB方程的值函数, 并以此完成了RRT算法扩展点之间的快速连接。最后, 通过讨论连接点之间的位置关系, 利用坐标变换缩小了HJB方程的求解区间, 降低了HJB方程的离线计算负担。

通过简化无人机模型的HJB-RRT算法和随机驱动RRT算法的仿真实验, 验证了HJB-RRT算法的高效性。通过实验结果的分析可以看出: HJB-RRT算法避免了扩展过程中TPBVP的求解, 并且可以保证数值稳定性以及两扩展点之间路径的最优性。

参考文献

- [1] 冯林, 贾菁辉. 基于对比优化的RRT路径规划改进算法 [J]. 计算机工程与应用, 2011, 47(3): 210-213.
(Feng Lin, Jia Jinghui. Improved algorithm of RRT path planning based on comparison optimization[J]. Computer Engineering and Applications, 2011, 47(3): 210-213.)
- [2] 徐娜, 陈雄, 孔庆生, 等. 非完整约束下的机器人运动规划算法 [J]. 机器人, 2011, 33(6): 666-672.
(Xu Na, Chen Xiong, Kong Qingsheng, et al. Motion planning for robot with nonholonomic constraints[J]. Robot, 2011, 33(6): 666-672.)
- [3] LaValle S M. Rapidly-exploring random trees: A new tool for path planning, Technical Report No. 98-11[R].

- [4] 宋金泽, 戴斌, 单恩忠, 等. 一种改进的RRT路径规划算法[J]. 电子学报, 2010, 38(B02): 225 - 228.
(Song Jinze, Dai Bin, Shan Enzhong, et al. An improved RRT path planning algorithm[J]. Acta Electron Sin, 2010, 38(B02): 225 - 228.)
- [5] Palmieri L, Arras K O. A novel RRT extend function for efficient and smooth mobile robot motion planning[C]. //Intelligent Robots and Systems (IROS). Chicago, USA, 2014.
- [6] Palmieri L, Arras K O. Distance metric learning for rrt-based motion planning with constant-time inference[C]. //IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 637-643.
- [7] Altaher M, Nomir O. Euler-Lagrange as Pseudometric of the RRT algorithm for optimal-time trajectory of flight simulation model in high-density obstacle environment[J]. Robotica, 1-19.
- [8] Seemann M, Janschek K. Exact and efficient local planning for orbitally flat systems within the RRT* framework[C]. //Control Automation Robotics & Vision (ICARCV). 2014 13th International Conference on. IEEE, 2014: 1631-1636.
- [9] hwan Jeon J, Karaman S, Frazzoli E. Optimal sampling-based feedback motion trees among obstacles for controllable linear systems with linear constraints[C]. //2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 4195-4201.
- [10] Pivtoraiko M, Mellinger D, Kumar V. Incremental micro-UAV motion replanning for exploring unknown environments[C]. //Robotics and Automation (ICRA). 2013 IEEE International Conference on. IEEE, 2013: 2452-2458.
- [11] Shan Y, Li B, Guo X. A considering lane information and obstacle-avoidance motion planning approach[C]. //17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014: 16-21.
- [12] Li Y, Bekris K E. Learning approximate cost-to-go metrics to improve sampling-based motion planning[C]. //Robotics and Automation (ICRA). 2011 IEEE International Conference on. IEEE, 2011: 4196-4201.
- [13] Yu Z, Jing C, Shen L. Real-time trajectory planning for UCAV air-to-surface attack using inverse dynamics optimization method and receding horizon control[J]. //Chinese Journal of Aeronautics, 2013, 26(4):1038-1056.
- [14] Whalley M S. A Piloted Simulation Investigation of the Normal Load Factor and Longitudinal Thrust Required for Air-to-Air Acquisition and Tracking[J]. 1993.
- [15] M.Bardi, I.Capuzzo-Dolcetta. Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations[M]. //Springer Science & Business Media, 2008.
- [16] M.Bardi, T.Raghavan, T.Parthasarathy. Stochastic and Differential Games:Theory and Numerical Methods[M]. //Springer Science & Business Media, 2012.
- [17] M. Falcone, R. Ferretti. Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi equations[M]. //Philadelphia: SIAM, 2013.
- [18] Cristian E, Falcone M. Fast semi-Lagrangian schemes for the eikonal equation and applications[J]. Journal on Numerical Analysis, Philadelphia: SIAM, 2007, 45(5): 1979-2011.
- [19] Ryo Takei, Richard Tsai. Optimal Trajectories of Curvature Constrained Motion in the Hamilton-Jacobi Formulation [J]. Journal of Scientific Computing, 2013, 54: 622-644
- [20] 吕伟新, 赵立军, 王珂, 等. 基于边界约束RRT的未知环境探索方法 [J]. 华中科技大学学报: 自然科学版, 2011, 39(2): 366-370.
(Lu Weixin, Zhao Lijun, Wang Ke, et al. Efficient exploration of unknown environments with RRT-boundary constraints [J]. Huazhong Univ. of Sci. & Tech:Natural Science Edition, 2011, 39(2): 366-370.)