

改进 RRT 在汽车避障局部 路径规划中的应用^{*}

宋晓琳[†], 周南, 黄正瑜, 曹昊天

(湖南大学 汽车车身先进设计与制造国家重点实验室, 湖南 长沙 410082)

摘要:根据传统快速搜索随机树算法(rapidly random-exploring trees, 简称 RRT)搜索速度快、所需时间短,但随机性大以及约束不足等特点,建立了直道和弯道的期望路径模型,采用高斯分布描述随机采样点,并引入启发式搜索机制,改进 RRT 算法.与原算法仿真对比,结果表明:改进算法所规划的路径质量显著提高,规划时间缩短一倍.同时,在 Prescan 软件中搭建直道和弯道仿真场景,跟随规划路径,结果表明:改进后 RRT 算法所得路径具有很好的跟随效果,且侧向加速度在车辆稳定性要求范围内,说明采用改进后的 RRT 算法进行汽车局部路径规划可行实用.

关键词:快速搜索随机树;汽车局部路径规划;高斯分布;路径跟随

中图分类号: TP242

文献标志码: A

An Improved RRT Algorithm of Local Path Planning for Vehicle Collision Avoidance

SONG Xiaolin[†], ZHOU Nan, HUANG Zhengyu, CAO Haotian

(Stake Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, China)

Abstract: The original Rapidly-exploring Random Trees(RRT) algorithm has a rapid exploring-speed and short required time in path planning though it has large randomness and lacks of constraints. Thus, an improved RRT was proposed where the expected paths were built in both straight and curved roads. The random points were accorded with normal distribution around the expected paths. Heuristic search method that led the random points to the goal with a certain probability was also used for improvement. Compared with the original RRT algorithm, it performs quite well in both time-efficient and path quality in the simulation. Meanwhile, the effectiveness of the improved RRT algorithm was verified in Prescan. The path can be followed well and the secure lateral acceleration was satisfied. In conclusion, the improved RRT is effective in the path planning for vehicle collision avoidance.

Key words: rapidly-exploring random trees; vehicle path planning; Gauss distribution; path following

近年来随着汽车向智能化的不断发展,无人驾驶汽车技术引得越来越多人开始关注.路径规划作

为其中一项关键技术,许多学者开展了有益的探索,并取得了一些研究成果.比如 A* 算法^[1]、D* 算法^[2]

^{*} 收稿日期:2016-04-15

基金项目:国家自然科学基金资助项目(51175159,51575169),National Natural Science Foundation of China(51175159,51575169)

作者简介:宋晓琳(1965—),女,湖南常德人,湖南大学教授,博士生导师

[†] 通讯联系人, E-mail:jqysxl@hnu.edu.cn

等启发式搜索算法,在复杂环境下有着很好的规划速度,但是路径并不理想;人工势场法^[3]是一种虚拟力算法,其优点是规划的路径平滑,但是容易陷入局部最优解;人工势场法与弹性绳模型结合^[4-6]在车辆局部路径规划时有理想的路径,但由于模型比较复杂,搜索效率低;此外蚁群算法、遗传算法、神经网络算法、水滴算法等智能仿生算法^[7-10],在处理复杂动态环境信息时有较好表现,但由于需迭代计算,时效性不够好。

快速搜索随机树算法(RRT)^[11]由Lavalle于1998年提出,是一种基于树结构的典型算法,在移动机器人路径规划中有着较早应用。由于算法在进行路径规划时是随机采样,不需要预处理,因此有着很快的搜索速度。路径点之间可以经过运动学、动力学仿真生成可执行曲线,因此该方法可用于解决含有运动学约束的路径规划问题。但RRT算法存在一些不足:1)重现性差,对同一环境进行多次路径规划结果不同;2)寻找最优或者次优路径能力较差;3)随机采样点在整个可行域内随机寻点的搜索方式,存在很多不必要的运算,影响算法速度。

随着RRT算法的应用和改进,一些学者也提出了不同的方法。偏向RRT^[12]以及双向RRT^[13]的提出使得算法的搜索效率得到了提高;DRRT^[14]与MP-RRT^[15]原算法在搜索路径的稳定性上做出了改进。但上述改进之后的结果并不适用于汽车行驶路径。针对以上不足,本文将**建立道路模型,考虑路径约束,改进RRT算法**,使其规划出的路径能够适用于汽车运动。

1 道路环境建模

在环境已知的情况下,建立道路环境模型。直道环境模型根据道路长度以及车道宽即可得到,弯道环境模型如图1所示,位于道路上的惯性坐标系的原点选取在道路中心线上,道路宽度为 W ,规划起始点即主车位置 A ,目标点 C ,障碍车位置为 B 。高速路直线之间的缓和曲线通常采用回旋线来实现平滑过渡,回旋线的特征是其曲率变化为常数。假定长度为 l 的回旋线线段起始曲率为 C_1 ,终止曲率为 C_f ,变化常数为 C_2 ,则有 $C(l) = C_1 + C_2 \times l$ 成立, $C_2 = (C_f - C_1)/l$ 。回旋线常采用多项式逼近的形式表示:

$$R(x) = R_0 + C_0 x + \frac{C_1}{2} x^2 + \frac{C_2}{6} x^3 \quad (1)$$

式中: R_0 为道路中心线初始横向偏移; C_0 为初始的方向角。根据图1,结合边界条件 $R(0) = 0, R'(0) = 0$ 可以将 R_0 和 C_0 消除掉。

为了保持车道宽度一致,弯道的左右边界是通过中心线上点向着其法线方向上下平移单个车道宽度来得到。在道路坐标系下中心线上的点可由式(2)表示。

$$p_{xy} = x e_x + y e_y \quad (2)$$

中心线上一点的切向量和法向量则可以表示成:

$$e_t = \frac{e_x + R'(x) e_y}{\sqrt{1 + R'(x)^2}}, e_n = \frac{-R'(x) e_x + e_y}{\sqrt{1 + R'(x)^2}} \quad (3)$$

因此道路左右边界则可以由(4)来表示

$$B_{xy} = p_{xy} \pm \frac{W}{2} e_n \quad (4)$$

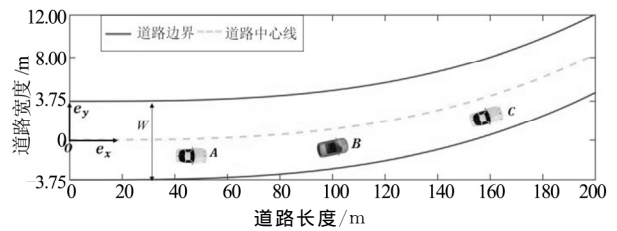


图1 弯道环境模型
Fig. 1 Curved road model

2 RRT 算法原理

基本的RRT算法如图2所示,RRT算法以给定的起始点为随机树根节点,根据当前环境快速有效地搜索可行域空间,通过随机采样点,将搜索导向空白区域并增加随机树的叶节点直至目标点区域,从而生成从起始点到目标点的路径。

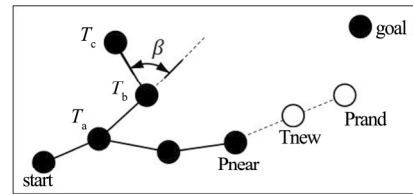


图2 RRT 算法扩展示意图
Fig. 2 Extension of the basic RRT

算法的一般步骤如下:

- 1) 首先通过 `environment()` 函数建立环境数据模型,初始化各项参数;
- 2) 通过 `while` 循环来判断树节点是否达到目标点 `goal` 范围内,若没有,则开始扩展点。先生成随机采样点 `Prand`,在树节点上通过函数 `Nearest()` 选择距离 `Prand` 最近的树节点作为扩展节点 `Pnear`,

通过扩展函数 New 得到新的树节点 Tnew, 并将其添加到随机树上, 直到循环终止.

3) 通过 getpath() 函数来得到生成的路径 path.

原算法主体程序如下:

RRT_pathplanning()

1) environment()

2) Tree(start)

3) While distance(tree(), goal) > δ

4) Prand ← Pick(Gaussrand(), goal)

5) Pnear ← (Prand, Tree)

6) Tnew ← New(Pnear, ΔL)

7) Tree ← add_tree(path_new)

8) endwhile

9) path = getPath()

如图 3 所示, 传统的 RRT 算法应用到车辆路径规划中存在以下问题:

1) 由于随机采样点随机性大, 导致搜索空间中有过多的冗余搜索, 表现为搜索树布满了道路环境空间;

2) 搜索出来的路径曲率变化过大, 甚至出现小范围内直角变化, 这样的路径并不能满足汽车行驶的正常状态;

3) 路径在靠近障碍时才开始避障, 对于运动中的汽车会造成失稳或者与障碍物发生碰撞.

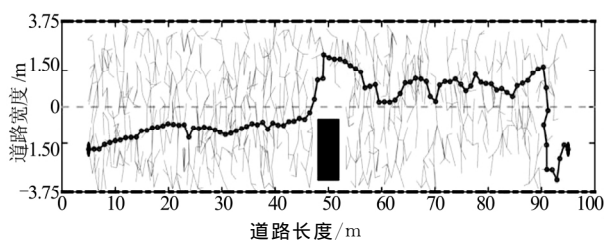


图 3 原 RRT 算法在直道路径规划的表现

Fig. 3 Performance of the basic RRT in straight road

3 RRT 算法的改进

3.1 期望路径模型

针对原 RRT 算法表现出来的问题, 提出了一种随机采样点高斯分布的改进 RRT 算法. 根据汽车行驶环境不同, 设计了两种期望路径模型.

3.1.1 直道模型

驾驶经验丰富的驾驶员期望的避障路径模型如图 4(a) 所示. 期望路径以函数 E_{pz} 表示, 其中各段均为直线段, start 为起始点, goal 为目标点.

提前避障在车辆避障路径规划中是必要的, 故

在模型中需要添加提前避障距离 S , 并根据车速调整大小. 设 V 为当前车速, t_c 为换道时间, 通常完成换道时间 t_c 为 1~2 s, ΔS 为自定义安全提前量.

$$E_{pz}(x) = \begin{cases} f_{z1} & x \in [X_s, X_{preL1}] \\ f_{z2} & x \in [X_{preL1}, X_{ob1}] \\ f_{z3} & x \in [X_{ob1}, X_{ob2}] \\ f_{z4} & x \in [X_{ob2}, X_{preL2}] \\ f_{z5} & x \in [X_{preL2}, X_g] \end{cases} \quad (5)$$

$$S_1 = V \times t_c + \Delta S \quad (6)$$

对于车速为 V 的汽车其刹车距离

$$S_2 = \frac{V^2}{2\mu g} \quad (7)$$

则提前避障距离

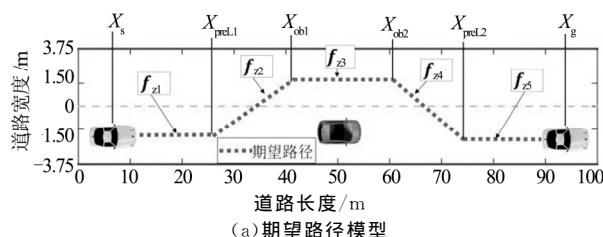
$$S = \max(S_1, S_2) \quad (8)$$

图 4(a) 中 f_{z2} 表示提前避障区域, 区间长度为 S , f_{z4} 区间长度也为 S .

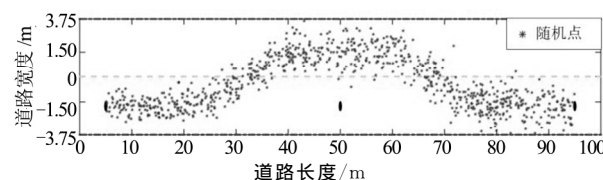
期望路径只是粗略的路径模型, 在此基础上进行平滑得出的路径并不能满足汽车安全稳定行驶的要求. 因此采用 RRT 算法进行路径寻优搜索. 为了使随机采样点分布在期望路径周围, 利用高斯分布函数生成的点集中在其均值周围的特点, 结合期望路径函数则可以实现这一目的. 在道路坐标系下随机采样点的高斯分布概率函数为:

$$f(x, y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \quad (9)$$

令 $\mu = E_{pz}(x)$, 则可以得到如图 4(b) 所示的随机采样点分布趋势图.



(a) 期望路径模型



(b) 随机采样点高斯分布

图 4 直道期望路径模型

Fig. 4 Expected path model of straight road

σ 的大小决定了随机点在 $E_{pz}(x)$ 周围的集中程度, σ 越小越靠近 $E_{pz}(x)$. 特别地, 对于 f_{z2} 与 f_{z4} 周围的随机采样点, 如图 4(a) 以 f_{z2} 为例, 通过相应水

平范围的随机点高斯分布旋转处理得到,即对 $x \in (X_{\text{preL1}}, X_{\text{ob1}})$, $\mu = X_o$, 旋转中心 (X_o, Y_o) :

$$X_o = \frac{X_{\text{ob1}} + X_{\text{preL1}}}{2} \quad (10)$$

$$Y_o = \frac{f_{z1}(X_{\text{ob1}}) + f_{z2}(X_{\text{preL1}})}{2} \quad (11)$$

旋转角度:

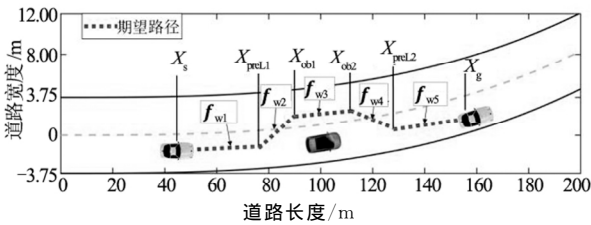
$$\theta = \arctan\left(\frac{Y_o}{X_o}\right) \quad (12)$$

3.1.2 弯道模型

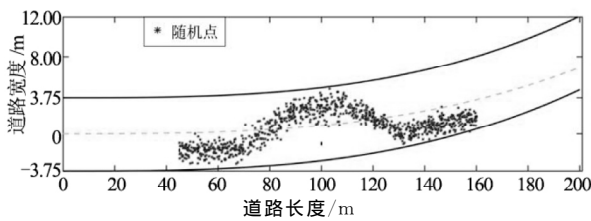
将弯道分为多段,采用直线代替弯道曲线的形式来完成期望路径模型的建立.如图5(a)弯道的期望路径以函数 E_{pw} 来表示.

$$E_{\text{pw}}(x) = \begin{cases} f_{w1} & x \in [X_s, X_{\text{preL1}}] \\ f_{w2} & x \in [X_{\text{preL1}}, X_{\text{ob1}}] \\ f_{w3} & x \in [X_{\text{ob1}}, X_{\text{ob2}}] \\ f_{w4} & x \in [X_{\text{ob2}}, X_{\text{preL2}}] \\ f_{w5} & x \in [X_{\text{preL2}}, X_g] \end{cases} \quad (13)$$

随机采样点在 f_w 各段函数区间的分布同直道的处理相同,从而可以得到如图5(b)所示的分布趋势图.



(a) 弯道期望路径模型



(b) 弯道随机采样点高斯分布

图5 弯道期望路径模型

Fig. 5 Expected path model of curved road

3.2 约束条件

要使一条规划出来的路径有效地应用于汽车运动,即路径可跟踪,则规划路径时必须满足道路环境约束.首先,随机树节点的生成要满足道路环境约束,设 $B_{\text{left}}, B_{\text{right}}$ 分别为道路的左右边界,则树节点位置坐标要满足:

$$B_{\text{right}} \leq \text{tree}(i)_y \leq B_{\text{left}}$$

$$\text{start} \leq \text{tree}(i)_x \leq \text{goal} \quad (14)$$

考虑到汽车是具有几何形体的,设其车宽为 D ,则上述 y 方向的约束变为:

$$B_{\text{right}} - \frac{D}{2} \leq \text{tree}(i)_y \leq B_{\text{left}} + \frac{D}{2} \quad (15)$$

假定汽车质心沿着规划的路径运动,为了保证行驶过程中的稳定性,规划出的路径的曲率变化不能过大.若在实际情况下前轮最大转角为 θ_{max} ,则路径中子节点与其父节点的连线和父节点与其父节点的连线之间的夹角 β 必须满足 $\beta < \theta_{\text{max}}$,通常不同车型的 θ_{max} 值在 $30^\circ \sim 40^\circ$ 之间.如图2中子节点 T_b 的父节点为 T_a , T_c 的父节点为 T_b ,那么夹角约束表现为:

$$\arctan\left(\frac{K_1 - K_2}{1 + K_1 K_2}\right) < \beta_T \leq \theta_{\text{max}} \quad (16)$$

其中: K_1 为直线 $T_a T_b$ 的斜率; K_2 为 $T_c T_b$ 的斜率. β_T 为夹角限制值.

为了保证所扩展的点不与障碍车有交集,即树节点不与障碍车碰撞的要求,采用安全椭圆包络障碍车,并适当放大安全椭圆以保证避障要求.若新节点与其父节点的连线不与安全椭圆相交,则所扩展的新点满足避障要求.取连线上的五等分点 $P_i(x, y)$,则约束方程表现为:

$$\frac{(x - x_{\text{ob}})^2}{(s \cdot a)^2} + \frac{(y - y_{\text{ob}})^2}{(s \cdot b)^2} > 1 \quad (17)$$

其中 $(x_{\text{ob}}, y_{\text{ob}})$ 为障碍车的位置,半车长 $a = 2$ m; 半车宽 $b = 1$ m; s 为安全椭圆放大系数,当 $s = \sqrt{2}$ 时,安全椭圆正好包络矩形的障碍车,因此从安全避障考虑, $s \geq \sqrt{2}$.

3.3 启发式搜索

原算法在扩展随机树时,由于缺乏导向机制,算法的收敛速度在一定程度上受到了影响,为了提高算法计算速度,引入启发式机制来对随机采样点以及随机树节点的选择进行处理.采样点 Prand 在随机生成过程中会以一定概率 ρ_0 选择目标点,从而将随机树节点向目标点引导,通常 $\rho_0 = 0.1$.

$$\text{Prand} = \begin{cases} \text{GaussRand}() & \rho > \rho_0 \\ \text{goal} & \rho < \rho_0 \end{cases} \quad (18)$$

其中, GaussRand() 为随机采样点生成函数.

另外,在选择 Pnear 时不再单独以距离 Prand 最近作为选择标准,而是以随机树节点的择优系数 C_h 来决定, Pnear 确定为 C_h 值最小所对应的树

节点.

$$C_h = \omega_i \cdot D_{pr} + \omega_j \cdot D_g \quad (19)$$

其中 ω_i, ω_j 为权重系数, 且 $\omega_i + \omega_j = 1$; D_{pr} 为树节点到 Prand 的距离, D_g 为树节点到目标点 goal 的距离. 当 $\omega_j > \omega_i$ 时, 选择出的 Pnear 具有向目标点靠近的趋势. 通过以上启发式的处理, 使得算法更快地收敛, 减少计算时间.

3.4 平滑处理

RRT 算法规划出来的路径通常会存在小范围内的曲折现象, 路径并不连续. 为了使得路径能够满足汽车在运动时的稳定性和安全性要求, 需要对规划出来的路径进行光滑处理. B 样条在处理路径光滑时能够不改变整个路径形状而进行局部调整^[16], 利用 B 样条这一特性, 来对算法所规划出来的路径进行插值拟合, 从而达到光滑路径的目的, 通常所采用的为 3 次 B 样条曲线.

当有 $m+1$ 个控制顶点 $P_i (i=0, 1, \dots, m)$ 时, 3 次 B 样条曲线表示为:

$$C_k(u) = \sum_{i=0}^3 P_{i+k} B_{i,3}(u), u \in [0, 1]$$

$$k = 0, 1, \dots, m-3 \quad (20)$$

其中基函数 $B_{i,3}$:

$$B_{0,3}(u) = (-u^3 + 3u^2 - 3u + 1)/3!$$

$$B_{1,3}(u) = (3u^3 - 6u^2 + 4)/3!$$

$$B_{2,3}(u) = (-3u^3 + 3u^2 + 3u + 1)/3!$$

$$B_{3,3}(u) = u^3/3!$$

3.5 算法流程

随机树节点的扩展受到随机采样点的影响, 通过以上方式的处理, 随机采样点不再是在可行域内随机分布, 而是具有一定的趋向性. 这样使得随机树节点的分布也具有趋向性, 算法的随机性得到了改善, 所规划出来的路径质量得到提高. 主体程序如下:

```
EXP-RRT_pathplanning()
1) environment
2) expect ()
3) Tree(start)
4) While distance(tree(), goal) > δ
5) Prand ← Pick(Gaussrand(), goal)
6) Tree.Ch ← Choice(Prand, goal, Tree)
7) Pnear ← Fitbest(Tree, Ch)
8) Tnew ← New(Pnear, ΔL)
9) if constraint(Tnew)
10) Tree ← addtree(Tnew)
```

```
11) return Tree
12) else
13) return Tree
14) end if
15) end while
16) path = getPath(tree)
17) road ← smoothing(path)
```

算法的实现过程如下:

1) 初始化阶段, 首先通过 environment() 函数建立道路环境模型, 根据现有的环境模型, 以 expect() 函数建立期望路径模型.

2) 路径求解阶段, 进入 while 循环来判断树节点是否达到目标点 goal 范围内, 若没有, 则开始扩展点. 随机采样点 Prand 通过 Pick() 函数在 goal 和 GaussRand() 函数所生成的点之间进行概率选择; 根据当前 Prand 计算树节点的择优系数 C_h , 并由 Fitbest() 函数得出 Pnear; 通过函数 New 在 Pnear 的基础上扩展出新节点 Tnew, 当新节点满足约束函数 constraint() 时, 新节点则添加到整个随机树 Tree 上, 否则, 返回循环重新寻点直到其终止.

3) 路径处理阶段, getPath() 函数从所得的 Tree 中获取最短路径, 最后通过函数 smoothing() 对所得路径 path 进行光滑处理, 得到一条平缓的路径.

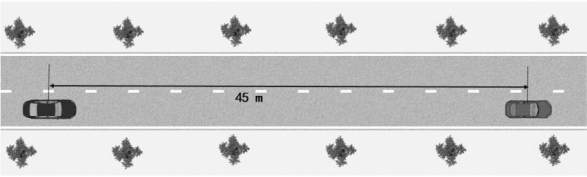
4 仿真实验验证

改进的 RRT 在应用于信息多变的不确定环境时会存在建模困难的缺陷, 本文主要对确定车道环境下的改进 RRT 开展研究. 由于条件有限, 不能在实际车辆中进行试验验证. 而 Prescan 软件能够对实际场景进行建模, 并且有根据实车建立的车辆模型数据库, 能够模拟实际车辆行驶过程. 因此, 为验证改进 RRT 算法的有效性, 在 Prescan 软件平台中搭建直道和弯道场景如图 6 所示, 仿真实验硬件平台 Win10 + Core-i7@3.6Hz + ARM@8G. 仿真参数如表 1~表 3 所示.

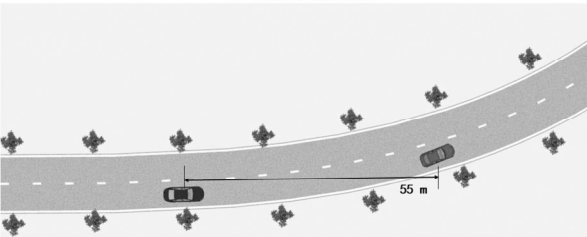
4.1 规划时间的影响参数分析

影响算法计算效率的重要参数主要包括搜索步长 ΔL 、约束夹角 β_T . 步长越小, 为了搜索到目标点所需要的节点数目也就越多, 计算时间越长; 约束夹角 β_T 越小, 规划路径也越平缓, 但计算时间也越长. 改变步长以及约束夹角并进行大量仿真实验, 统计表明: 在 $\Delta L = 3, \beta_T = 15^\circ$ 时, 改进 RRT 算法在规划

时间以及路径效果上的综合表现较好. 图 7 为 $\Delta L=3$ 时,不同角度下计算时间的统计,以 20 次计算时间平均值做一次统计.



(a) 直线道路场景示意图



(b) 曲线道路场景示意图

图 6 仿真场景示意图
Fig. 6 The scenarios setting

表 1 道路模型数据
Tab. 1 Data of roads

道路	横向长度/m	障碍车位置 B	起始点 A	目标点 C	单车道宽度/m
直道	100	50, -1.9	5, -1.9	80, -1.75	3.75
弯道	200	100, -0.9	45, -1.9	160, 1.8	3.75

表 2 高斯分布函数参数
Tab. 2 Value of σ

区间	直道 σ	弯道 σ
$[X_s, X_{preL1}]$	0.6	0.55
$[X_{preL1}, X_{ob1}]$	0.7	0.7
$[X_{ob1}, X_{ob2}]$	0.6	0.55
$[X_{ob2}, X_{preL2}]$	0.7	0.7
$[X_{preL2}, X_g]$	0.6	0.55

表 3 其他仿真参数
Tab. 3 Parameters of simulation

名称	数值
目标偏向概率 ρ_0	0.1
安全椭圆放大系数 s	1.7
搜索步长 $\Delta L/m$	3
约束夹角 $\beta_T/(^\circ)$	15
权重系数 ω_j	0.4
权重系数 ω_j	0.6
搜索终止长度 δ/m	2

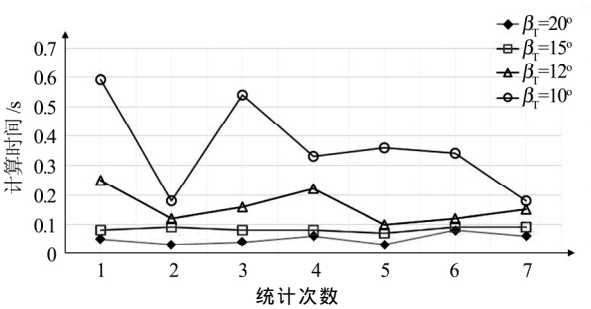


图 7 约束角度对计算时间的影响
Fig. 7 Constraint angle influence on time

4.2 规划路径对比

为说明改进 RRT 算法的效果,在直道场景中,采用了其余 2 种规划算法来进行对比. 一种是蚁群算法^[7],另一种是弹性绳算法^[5]. 直道仿真对比结果如图 8 所示,改进前后的算法规划效果在弯道中的对比如图 9 所示,路径效果参数对比如表 4 所示. 由图和表的结果对比可知:

1)相对于蚁群算法和原 RRT 算法,改进 RRT 算法与弹性绳算法规划的路径都更为平滑,不存在频繁的大曲率变化,且路径较短.

2)从直道的规划时间上看,传统的蚁群算法用时最长,而弹性绳算法平均计算时间为 1.42 s. 改进后的 RRT 算法平均计算时间 0.25 s,相对于原 RRT 计算时间略有增加,这是由于增加了模型处理过程以及各约束条件所导致的. 但与其余 2 种算法相比,改进 RRT 算法依然保持其在计算时间上的优势.

3)原 RRT 算法在靠近障碍时才开始避障,改进 RRT 算法能够实现提前避障,并且根据车速不同自动调节避障提前量,这对安全行驶很重要.

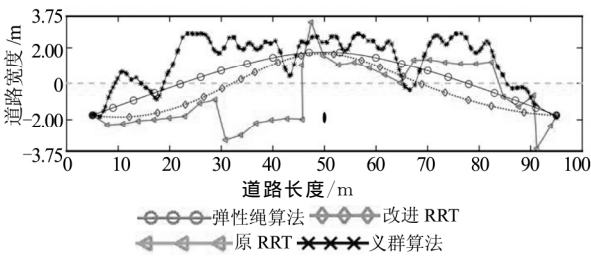


图 8 直道算法效果对比图
Fig. 8 Performance contrast in straight road

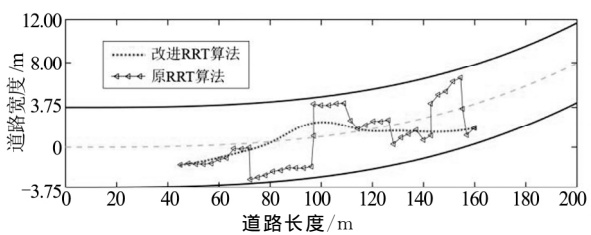


图 9 弯道算法效果对比图
Fig. 9 Performance contrast in curved road

表 4 路径效果参数对比
Tab. 4 Path data contrast

道路	算法	长度 /m	路径曲率 均值/ m^{-1}	路径曲率 均方差	平均计算 时间/s
直道	原 RRT	100.9	0.100 3	2.095	0.17
	改进 RRT	90.4	0.003 7	0.179	0.25
	蚁群算法	98.8	0.013 0	0.530	6.86
	弹性绳算法	89.6	0.002 3	0.023	1.42
弯道	原 RRT	125.8	0.010 2	1.382 9	0.35
	改进 RRT	115.3	0.000 14	0.009 7	0.45

4.3 路径跟随验证

对一条规划出来的路径,需要验证其有效性,即路径的跟随性,本文采用路径跟随时主车的侧向加速度曲线监测路径的稳定性,跟随速度 20 m/s. 直道和弯道仿真结果分别如图 10、图 11 所示.

由图 10(a)(b)可知,直道跟随路径基本和目标路径一致,跟随误差在 0.2 m 以内,误差较小;车辆保持稳定行驶时的最大侧向加速度由地面附着力决定,通常为 0.8g. 由图 10(c)可知,跟随时的侧向加速度在 0.4g 以内,说明整个过程中都能够保证主车按照路径行驶时的稳定性. 弯道仿真结果如图 11 所示,跟随误差都在 0.1 m 以内,跟随效果好;侧向加速度都在 0.5g 范围内,满足稳定性要求.

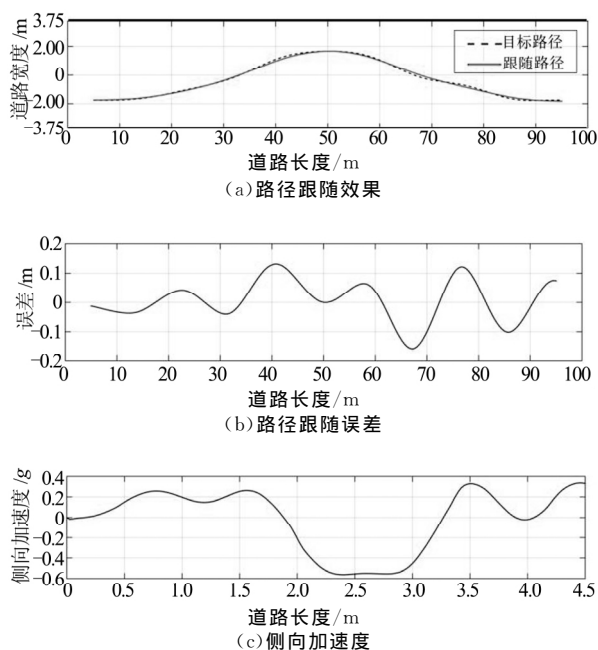


图 10 直道仿真结果

Fig. 10 Simulation results in straight road

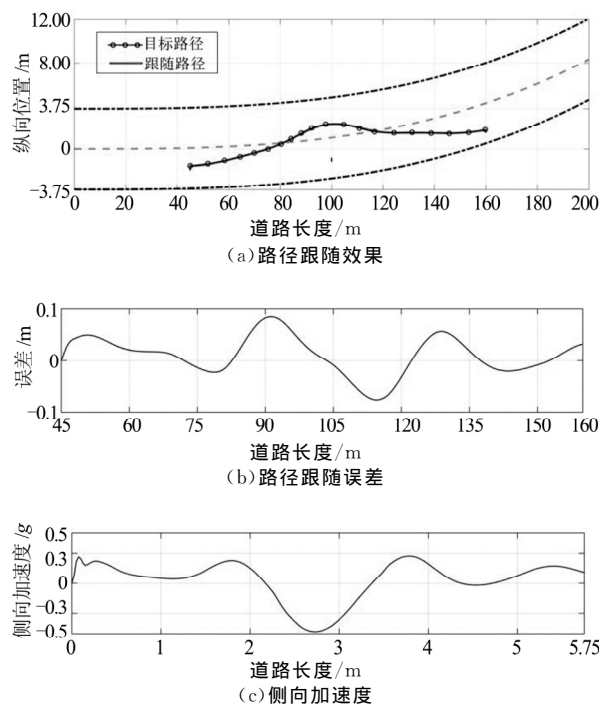


图 11 弯道仿真结果

Fig. 11 Simulation results in curved road

5 结 论

本文在将 RRT 算法应用到汽车避障局部路径规划时,针对原算法在随机性以及路径曲率变化上的不足,建立了直道和弯道的期望路径模型,使随机采样点在期望路径模型周围近似呈高斯分布,采用启发式搜索方式使采样点和随机树节点向目标点靠近,从而降低算法的随机性;并且在路径规划时加入约束,使得所规划出的路径更为合理. 仿真实验结果表明:改进 RRT 算法不仅规划出的路径质量高、实时好、跟随性强,而且车辆的稳定性也满足要求. 因此,改进 RRT 算法可以应用于实时汽车路径规划.

致谢:感谢天欧 TNO 公司(中国·上海)的 Prescan 软件支持.

参考文献

- [1] YAO J, LIN C, XIE X, *et al.* Path planning for virtual human motion using improved A* algorithm[C]//Proceedings of the Conference on Information Technology: New Generations (IT-NG). Washington, DC: IEEE Computer Society, 2010: 1154—1158.
- [2] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. *Numerische Mathematik*, 1959, 1(1): 269—271.
- [3] KHATIB O. Real-time obstacle avoidance for manipulators and mobile robots[J]. *The International Journal of Robotics*

- Research, 1986, 5(1): 90—98.
- [4] SONG X, CAO H, HUANG J. Vehicle path planning in various driving situations based on the elastic band theory for highway collision avoidance[J]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2013, 227(12): 1706—1722.
- [5] SONG X, CAO H, HUANG J. Influencing factors research on vehicle path planning based on elastic bands for collision avoidance[J]. SAE International Journal of Passenger Cars—Electronic and Electrical Systems, 2012, 5(2): 625—637.
- [6] 肖浩, 宋晓琳, 曹昊天. 基于危险斥力场的自动驾驶汽车主动避障局部路径规划[J]. 工程设计学报, 2012, 19(5): 379—384.
XIAO Hao, SONG Xiaolin, CAO Haotian. Local path planning for autonomous vehicle collision avoidance based on dangerous repellant fields[J]. Chinese Journal of Engineering Design, 2012, 19(5): 379—384. (In Chinese)
- [7] 康冰, 王曦辉, 刘富. 基于改进蚁群算法的搜索机器人路径规划[J]. 吉林大学学报:工学版, 2014, 44(4): 1062—1068.
KANG Bing, WANG Xihui, LIU Fu. Path planning of searching robot based on improved ant colony algorithm[J]. Journal of Jilin University: Engineering and Technology Edition, 2014, 44(4): 1062—1068. (In Chinese)
- [8] HU Y, YANG S X. A knowledge based genetic algorithm for path planning of a mobile robot[C]//Proceedings of the Conference on Robotics and Automation. Washington, DC: IEEE Computer Society, 2004: 4350—4355.
- [9] 吴乙万, 黄智. 基于动态虚拟障碍物的智能车辆局部路径规划方法[J]. 湖南大学学报:自然科学, 2013, 40(1): 33—37.
WU Yiwan, HUANG Zhi. Dynamic virtual obstacle based local path planning for intelligent vehicle[J]. Journal of Hunan University: Natural Sciences, 2013, 40(1): 33—37. (In Chinese)
- [10] SHAH-HOSSEINI H. Problem solving by intelligent water drops[C]// Proceedings of the Conference on Evolutionary Computation. Washington, DC: IEEE Computer Society, 2007: 3226—3231.
- [11] LAVALLE S M. Rapidly-exploring random trees: A new tool for path planning[J]. Algorithmic & Computational Robotics New Directions, 1998: 293—308.
- [12] LAVALLE S M, KUFFNER J J. Rapidly-exploring random trees: progress and prospects[J]. Algorithmic & Computational Robotics New Directions, 2000: 293—308.
- [13] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning[C]// Proceedings of International Conference on Robotics and Automation. Washington, DC: IEEE Computer Society, 2003: 995—1001.
- [14] FERGUSON D, KALRA N, STENTZ A. Replanning with rrt[C]//Proceedings of International Conference on Robotics and Automation. Washington, DC: IEEE Computer Society, 2006: 1243—1248.
- [15] ZUCKER M, KUFFNER J, BRANICKY M. Multipartite RRTs for rapid replanning in dynamic environments[C]//Proceedings of International Conference on Robotics and Automation. Washington, DC: IEEE Computer Society, 2007: 1603—1609.
- [16] 李红, 郭孔辉, 宋晓琳. 基于样条理论的自动垂直泊车轨迹规划[J]. 湖南大学学报:自然科学版, 2012, 39(7): 25—30.
LI Hong, GUO Konghui, SONG Xiaolin. Trajectory planning of automatic vertical parking based on spline theory[J]. Journal of Hunan University: Natural Sciences, 2012, 39(7): 25—30. (In Chinese)