

第三届“ScienceWord 杯”数学中国

数学建模网络挑战赛

承 诺 书

我们仔细阅读了第三届“ScienceWord 杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛报名号为：**1195**

参赛队员（签名）：

队员 1：汪海涛

队员 2：万敏

队员 3：潘海军

参赛队教练员（签名）： 邓磊

参赛队伍组别：**大学组**

第三届“ScienceWord 杯”数学中国

数学建模网络挑战赛

编 号 专 用 页

参赛队伍的参赛号码：（请各个参赛队提前填写好）：

1195

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

2010 年第三届“ScienceWord 杯”数学中国 数学建模网络挑战赛

题 目 聪明的汽车

关 键 词 逐步优化 分类讨论 抽象 平面几何 AutoCAD

摘 要：

在狭窄空间停车一直是困扰众多驾驶员的难题之一，本问题要求我们协助驾驶员解决停车问题，即在狭窄的空间里把车停放在合适的位置以及在短小的车位上侧位停车。

设计这样一个模型，其核心是路线选择的模型与算法。用数学的观点的把显示中的停车过程抽象成平面几何问题，从而使文中需要解决的问题都归结到纯数学的研究上。

对于问题一和问题二，实质上可以用同样的模型来解决。首先，根据停车位的形状将停车位抽象位平面几何图形并且分为三种类型，即‘一’字形、‘非’字形和侧位。由层层优化的思想又把‘非’字型分为四类。然后用极值分析、平面几何相关理论对每一类建立相应的理想的最优模型，得到最优路径的算法。并且运用 AutoCAD 等绘图软件绘出相应的轨迹图，用 VC++ 编写出相应的程序模块。

结合实际考虑，诸如驾驶员技术问题、情绪状态等人为因素和路况等客观因素的影响，往往使得驾驶员不能按照最优路径停车，甚至会出现较大偏差。于是，我们对此模型进行优化，在给定停车位的尺寸内设定偏差范围，判断实际偏差是否在给定偏差范围内。当实际偏差在允许范围内，用同样的工具软件再次找出在此情况下的最优路线，否则认为停车失败。最后，对此模型进行数据测试，从而评估该模型的优缺点。

参赛队号 1195

参赛密码 _____
(由组委会填写)

所选题目 A

一、 问题重述

随着经济的发展,人们的生活水平日益提高,拥有汽车人口的比例大幅增加,从 2009 年汽车上市公司的年报,以及汽车工业协会公布的 2010 年前两个月的数据来看,汽车数量高增长的趋势已经确立。因此,停车问题困扰着许多驾驶员,在狭窄的空间里把车停放在合适的位置,或者在短小的停车位上侧位停车,一直是考验驾驶员技术和信心的问题,调查报告显示:57%的驾驶员对自己的停车技术缺乏自信,这不仅影响驾驶体验,也使停车空间得不到充分利用。

本题的目的就是协助驾驶员解决侧位停车的问题,设计一个能协助驾驶员确定汽车是否能顺利停入,在此基础上,帮助驾驶员准确判断进入停车位的位置及角度。对此,我们将综合运用平面几何、极值分析等数学知识,并结合 MATLAB,AutoCAD,VC++,Windows 自带画图工具等应用软件,对进入停车位的位置以及角度做出合理的选择,并综合考虑实际可操作性、难度系数、驾驶员心理等各种因素,使驾驶员停车时更加方便、顺利。

要求: 1) 判断汽车是否可以在指定停车位停放。

2) 如果已判定汽车可以在该停车位停放,根据停车位、汽车参数等具体情况选择合理的位置和角度进入,并作出理想路线的示意图。

二、 问题假设

- 1、假设汽车的俯视图为一矩形,即假设车轮位于汽车的四角。
- 2、假设汽车是前轮驱动还是后轮驱动对停车造成的影响可以忽略,汽车本身性能(机动性、稳定性)较好,驾驶员状态良好。
- 3、各类汽车车长、车宽、转弯半径与转角的关系是已知的,即 $R_{\min}=L/\sin \beta_{\max}$, β 为转向角, R_{\min} 为汽车的转弯半径, L 为轴距。且各数据在车辆制造过程中造成的误差忽略不计。
- 4、车载显示仪、车载传感器都能准确的测出停车位的长度或宽度,车身到停车位的距离,车道的宽度等所需数据并且能显示出最优方案。
- 5、因为题目要求在狭窄的空间或短小的停车位上停车,则可以把所有汽车视为二轮转向系统。
- 6、汽车一般以倒车方式进入车位路线,并假设如何进入,就可如何退出(即路线一致)。
- 7、在停车过程中,为了达到预定的要求,在系统给出正确指令的前提下,倒车次数不得超过五次,超过 5 次则认为停车失败。且在停车过程中,若占据了其他车位,则也认为停车失败。
- 8、行车路线与预定路线有偏差时,只考虑第一次偏差,此后的行车路线与预定路线形状一致。

三、 符号说明

符号	含义
n	汽车车身的宽度
m	汽车车身的长度
$g1$	汽车车头离停车位的水平距离
$g2$	汽车侧面离停车位的距离

报名号#1195

$L(L_{min})$	停车位的(最小)宽度
α	汽车第一次转弯的偏转角度
R	汽车最小转弯半径
M	车道的宽度
$L_{实}$	停车位的实际宽度

*注：该符号说明为全局变量，在具体的分析过程中的符号，将随分析及其图形给出说明。

四、 问题的分析、模型的建立及求解

4.1 问题分析

已知：

利用车载传感器测出车身侧面距停车位的距离 g_2 、汽车距停车位的水平距离 g_1 ，汽车自身的长 m 与宽 n ，以及停车位的宽度 L 。

要求：

通过综合分析问题，建立合理的模型，由已知及测得的数据判断汽车是否可以在该停车位停车。并选出最佳停车路线。

分析：

停车位可以有多种形式，考虑到停车的各种情况，以及停车位的排布情况，可将停车位分为三种基本类型：‘非’字型停车位，‘一’字型停车位，侧位型停车位（即车位与道路两旁或停车场边沿呈一个锐角或钝角，而锐角与钝角虽然方向不同，但本质相同，可归为锐角一类），三种类型车位如图 4.1-1 所示。

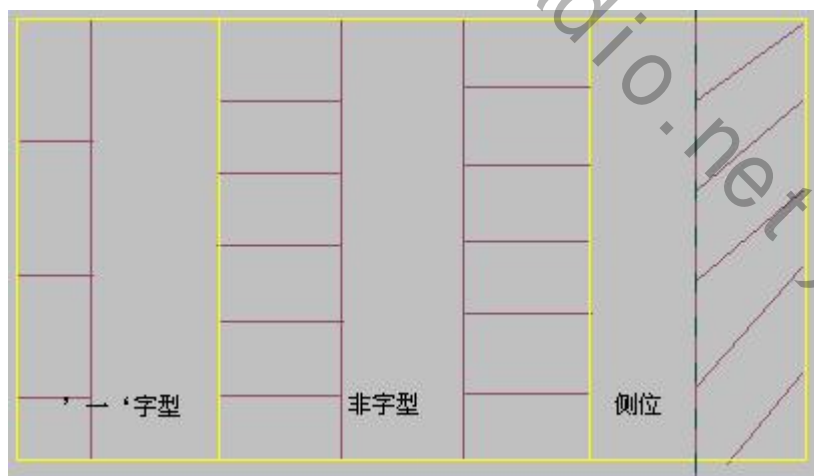


图 4.1-1

我们将汽车抽象成一个矩形，其车轮位于矩形的四角，以将这种实际的模型，抽象为平面解析几何的问题来求解。在可以停车的情况下，综合考虑停车路线长度、停车路线复杂度（以汽车转向次数度量）及停车位宽度（或长度）三个因素最优者为最优方案。

为判断停车位是否符合停车要求，我们对以上三种情况分别进行讨论。

首先，从理论上分析，只要车辆能按照预先设计好的恰能进入停车位的临界条件行驶，那么这个停车位就满足停车要求，而这个临界条件可由车身恰好从车位转角处擦过，以及车身贴近停车位等临界情况建立模型来确定。

然后，根据理论得出的临界条件，再加上实际问题中对停车位的要求，以及技术可行性，根据汽车的相关参数和停车位的情况，判断出该停车位是否满足停车要求。如果停车位满足停车要求，再根据模型所得算法，利用 VC++ 编程，输入已知量或可测量，输出一个数字对 (g_1, g_2) ——汽车进入坐标，及转弯角度值（由轨迹弧线所对应的圆心角度量），使驾驶员明确进入停车位的位置及角度，以成功进入停车位。

4.2 模型的建立与求解

由三类不同的车位类型分别建立不同的模型（相关程序见附录 1）

4.2.1 模型一：侧向停车模型

模型分析：

如图 4.2-1 所示，在该模型中， g_1, g_2 是可测的， n, α 是预先给定的。则，设进入停车位的最优路径是 S_1 ，此时的最小转弯半径为 $R(\alpha)$ ，由几何知识得：

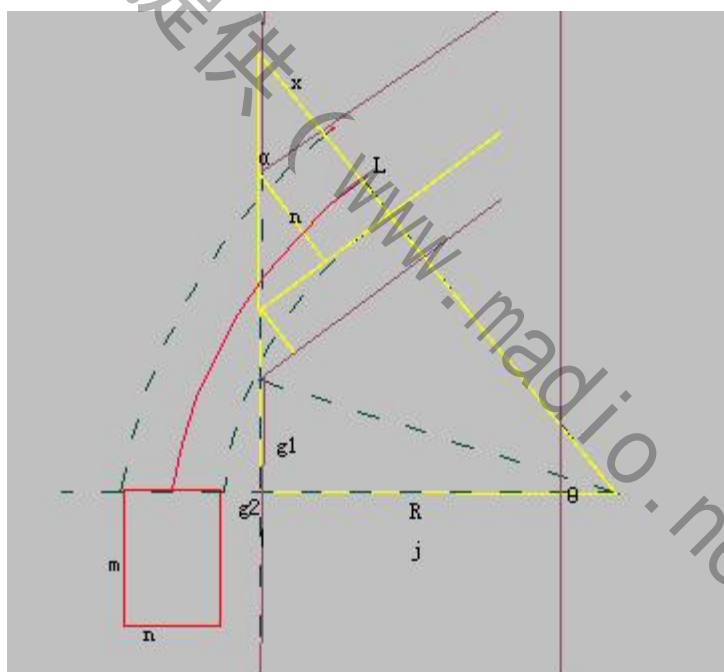


图 4.2-1

$$\text{停车位的倾斜角 } \alpha = \beta + \theta, \quad j = R - n/2 - g_2,$$

$$\text{解三角形知 } \cos \theta = j / (R - n/2), \quad \sin \theta = g_1 / (R - n/2),$$

则 $\cos \beta = \cos (\alpha - \theta)$,

停车位宽减去车宽后 $a = R - n/2 - (R - n/2) \cos \beta$

从而, $a = R - n/2 - \cos \alpha * (R - n/2 - g_2) - g_1 * \sin \alpha$

停车位宽 $L = a + n/2$

① 知转弯半径 R 求 a , 从而得出最小的停车位宽 L , $L_{\text{min}} < L$ 时, 不可进;

② 知 a 求 R , α 为倾斜角,

则转弯半径 $R = (a + n/2 + g_1 * \sin \alpha - \cos \alpha (n/2 + g_2)) / (1 - \cos \alpha)$

(由 a 求 R , 判断 $R \geq 4.5$ 米, 给定 a , 若对于任意的

$g_1, g_2 (\leq M)$, $R < 4.5$ 米, 则认为停车失败, $n + g_2 \leq M$ (车道宽, 可测)。

4.2.2 模型二：一字型停车模型

模型分析:

首先, 由实际情况可知, 汽车进入该类型停车位停放时车身必定与车位底线平行, 从而有两种进入的方式, 即从停车位前直接开入与开过停车位后倒车进入。但是两种方式的抽象出的数学模型相同, 均可由图 4.2-2 表示。可以由几何知识初步得出汽车在此情况下的轨迹图。

如图所示, 考虑汽车恰能进入停车位的临界情况。在这种情况下, 所需停车位的长度 L 是最小的, 从而得出能满足停放要求的最小停车位。

根据理论上得出的最小值, 再加上实际情况需要考虑的因素, 诸如技术要求, 停车规范等, 从而得出对实际情况的判断, 显然, 当车位比这种情况大时, 车位必定能满足停车要求。

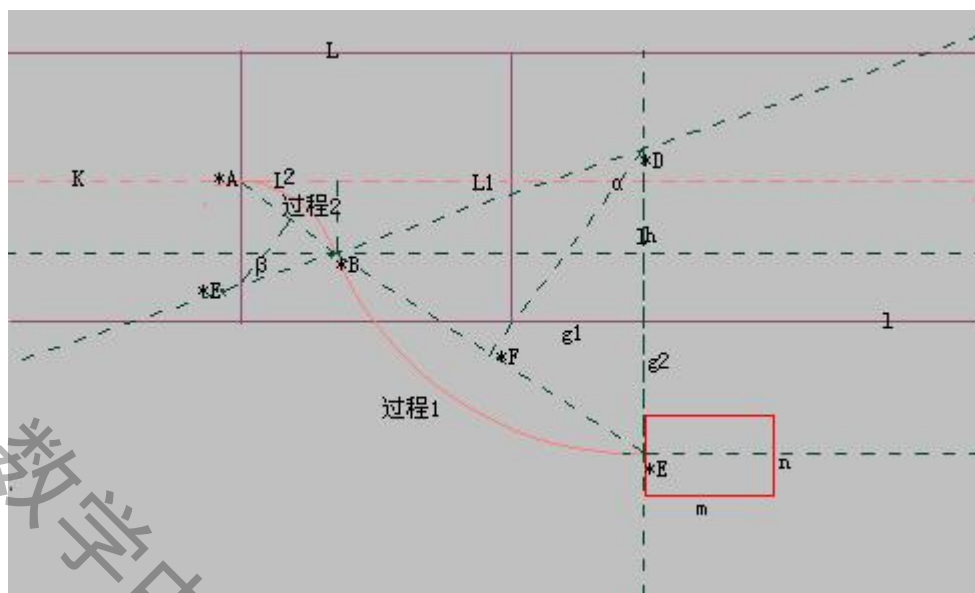


图 4.2-2

具体计算分析如下：

汽车第一轨迹（过程 1）在垂直于停车位方向上移动的距离：

$$l = K/2 - h + g_2 + n/2$$

汽车第一轨迹在水平方向上移动的距离（即车头前进的距离）：

$$L_1 = \sqrt{R^2 - (R - l)^2}$$

解三角形得 $\sin \alpha = L_1/R$

$$\cos \alpha = (R - l)/R$$

从而， $\sin(\alpha/2) = \sqrt{l/2R}$;

汽车第二轨迹在水平方向上移动的距离：

$$L_2 = 2R \cdot \sin(\alpha/2) \cdot \cos(\alpha/2) = R \cdot \sin \alpha;$$

汽车第一轨迹与第二轨迹交点与停车位中轴线的距离：

$$h = 2R \cdot \sin(\alpha/2) \cdot \sin(\alpha/2);$$

联立以上几式，得： $(K/2) - 2R \sin(\alpha/2) \cdot \sin(\alpha/2) + g_2 + (n/2) = R - R \cos \alpha,$

则 $(4R - K - 2 * g_2 - n) / (4R) = \cos \alpha$

汽车内侧行走圆弧半径： $r = R - (n/2)$, $j = r - g_2 = R - (n/2) - g_2$,

又由解三角形得 $\sin \theta = (j/r)$; $\cos \theta = (g_1/r)$;

直角三角形中 $g_1^2 + j^2 = r^2$;

则 $g_1^2 = [R - (n/2)]^2 - [R - (n/2) - g_2]^2 = (2R - n - g_2) * g_2$,

可求出 g_1

则汽车位置 (g_1, g_2) (g_2 可测), 转角为 α , 然后, 反转方向盘, 转角仍为 α , 即可顺利进入停车位。

4.2.3 模型三：非字型停车模型

模型分析：

在该种模型下, 有三种方式可以进入停车位, 分别为图 4.2-3, 图 4.2-4, 图 4.2-5 所示。且在这种情况下, 由停车规范可知, 车头必然朝外, 否则汽车退出停车位时会遇到许多问题。分别考虑三种方式下, 恰能进入停车位的极值问题, 这个极值则为理论上可以停车所需要的空间, 然后再根据实际情况适当调整这个空间的大小。

模型建立：

(1) 在非字型车位情况下, 如果行车道较宽, 则可以按照如图 4.2-3 所示路线倒车进入停车位中。可得到理论上所需的最小停车位。

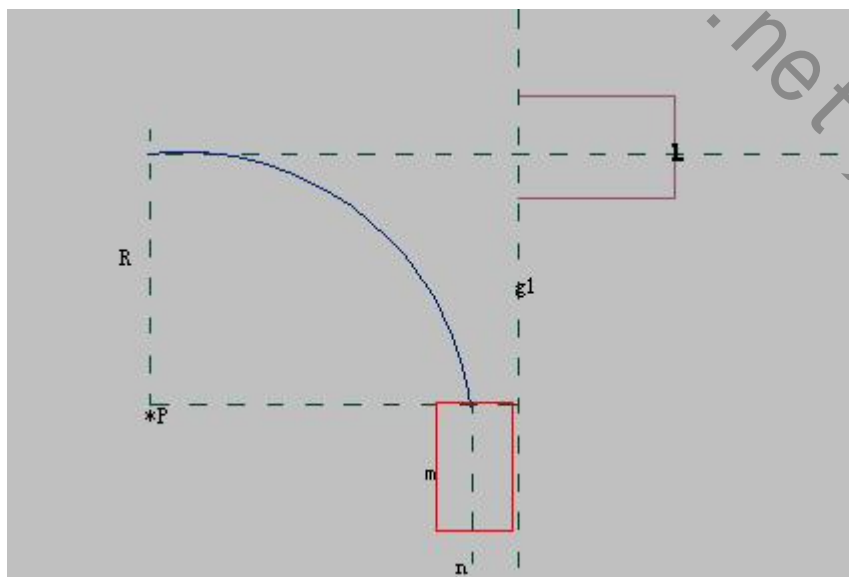


图 4.2-3

具体计算分析如下： $L = n$ ； $g_1 = R - n/2$ ； $g_2 = 0$ ；

(2) 如图，倒车进入停车位时，汽车内侧行走弧线正好与停车位一角相切， L 的大小与 g_2 、 α 有关。

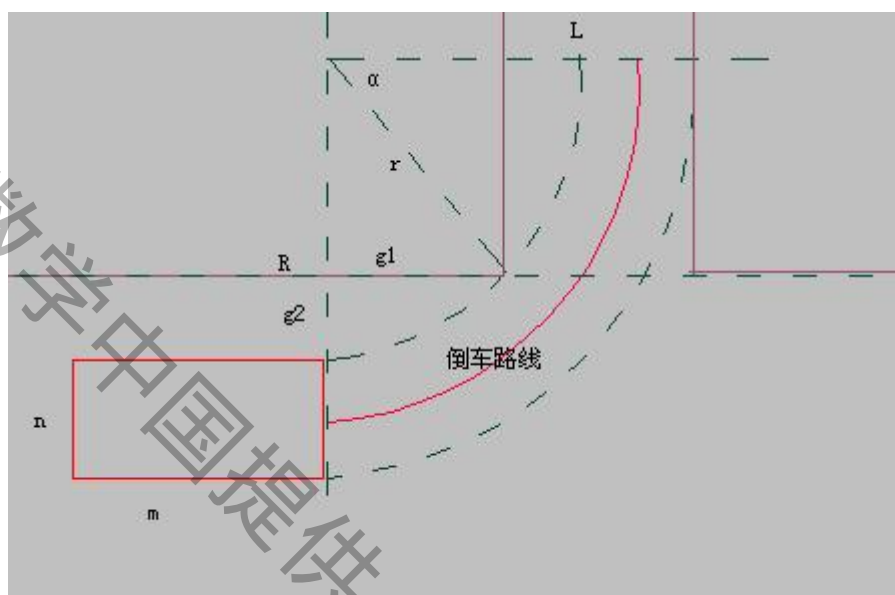


图 4.2-4

具体分析计算如下：

① $\alpha = 45^\circ$ 时，

汽车内侧行走圆弧半径 $r = R - n/2$ ；

$$g_1 = (\sqrt{2}/2)r;$$

$$g_2 = r - g_1 = [1 - (\sqrt{2}/2)][R - (n/2)];$$

$$\text{停车位宽 } L = [1 - (\sqrt{2}/2)][R - n/2] + n/2.$$

则汽车位置 (g_1, g_2) ，转角 α 为 90° 。

③ α 角不定时，有：

汽车内侧行走圆弧半径 $r = R - n/2$ ；

$$g_1 = r \cos \alpha; \quad g_2 = r - r \sin \alpha;$$

停车位宽 $L = r - r \cos \alpha + n = (R - n/2) * (1 - \cos \alpha) + n$.

$L_{1测} \geq L_{min} = R + n/2 - g1$, 其中, $L_{1测}$ 为车载显示器所给的停车位宽

$$g2 = (R - n/2) * (1 - \sqrt{1 - (g1)^2 / (R - n/2)^2}),$$

给定 $g1$, 求取 L , $g2$.

因 α 不定, 我们可以根据 $g1$ 的情况来随意选取一适当的 α 值, 并计算出 $g2$, 则汽车位置 $(g1, g2)$, 转角为 90 度。

(3) 在非字型停车位情况下, 由于行车道较窄, 不能满足情况 (1) 的要求, 则考虑如下图所示的进入停车位的方法。而这种进入方法又有两种可能, 分别为图 4.2-5、4.2-6 所示。

在图 4.2-5 的情况下, 理论上可以暂且忽略车身内侧与停车位之间的距离, 因为这个距离并不影响汽车转向。此时, 这种情况是一种特殊情况, 即汽车在第二轨迹起始处时, 以 30 度的角度倒车, 则可以恰好进入停车位中。

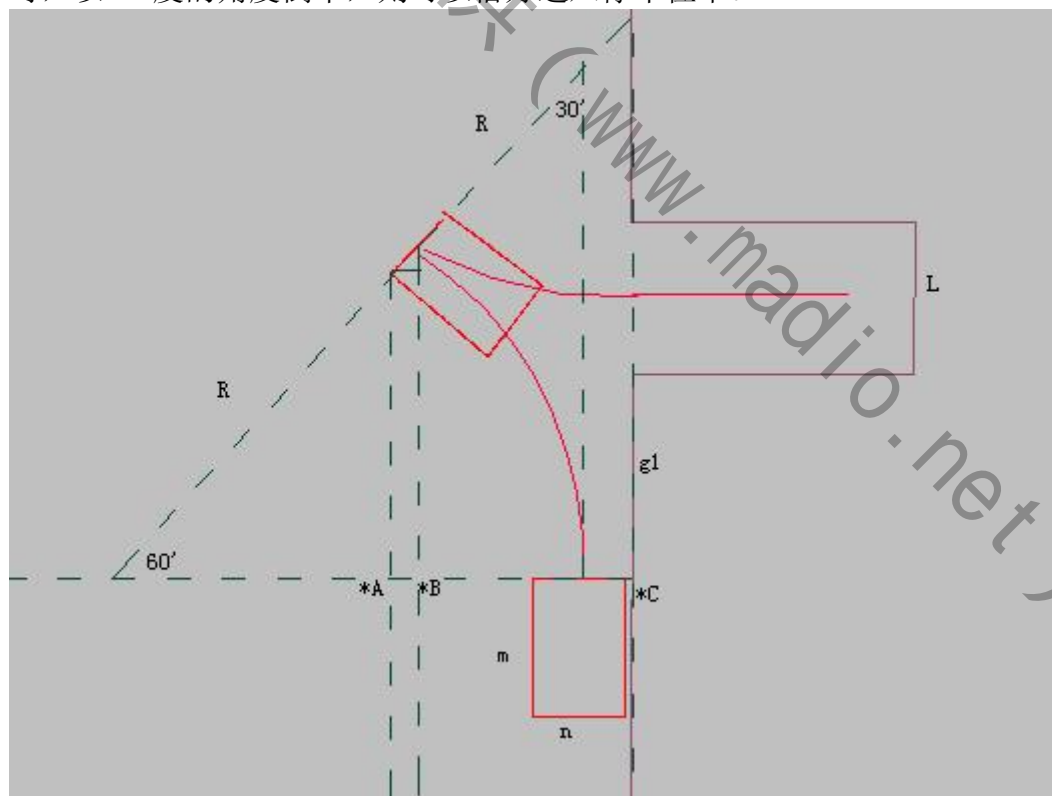


图 4.2-5

具体分析计算如下:

$$g1 = \sqrt{3}R - R - L/2; \quad L_{测} \geq L = n,$$

已知, n, R . 求 $g_1 = (\sqrt{3}-1)R - n/2$, $R/2 + 3 \cdot n/4 \leq M$,

$$\alpha_1 = \pi/3, \quad \alpha_2 = \pi/6$$

则汽车位置 $(g_1, 0)$, 第一转角为 $\alpha_1 = \pi/3$, 第二转角为 $\alpha_2 = \pi/6$

(4) 如图 4.2-6 情况, 则是在图 4.2-5 的情况下的一个深入分析, 考虑汽车倒车时不能垂直进入停车位, 而是车身内侧与停车位一角相切进入的情况。在这种情况下, 可以得到汽车第一次转弯半径与转角, 进而确定第二次转弯半径与转角, 从而得出模型的最优解。

由图易知, 汽车第二轨迹在延长后就是非字型停车模型 (2), 由此我们可知第二转弯半径就是 R 。

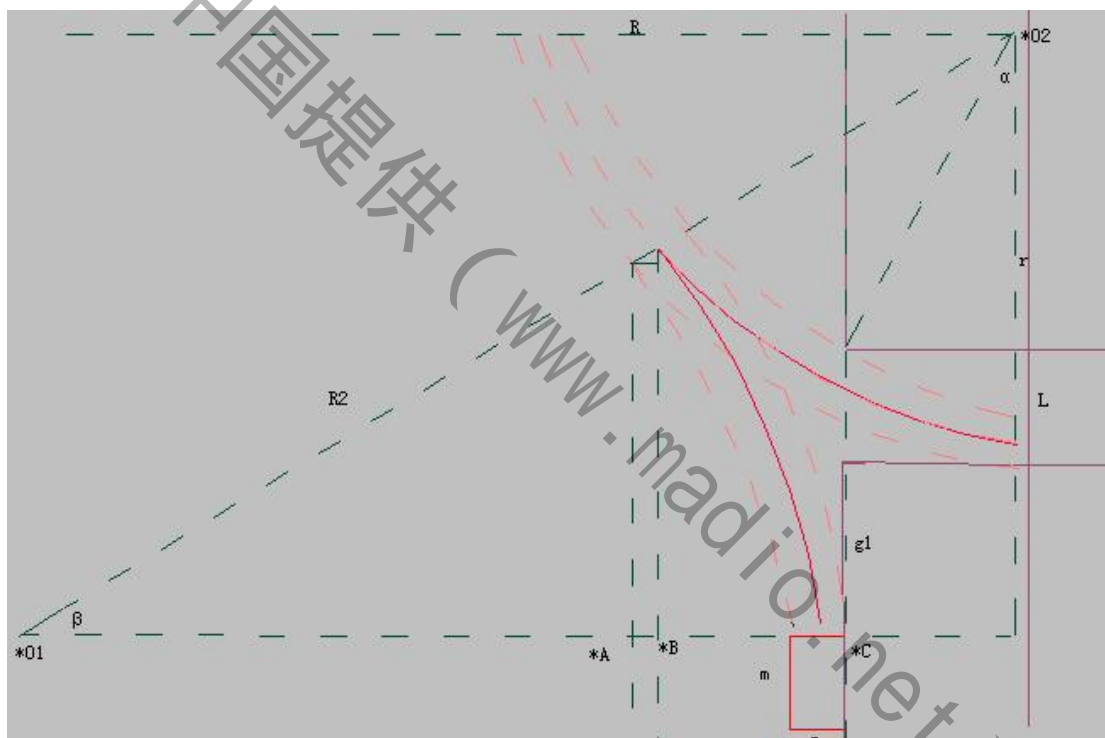


图 4.2-6

在大三角形中, 由勾股定理:

$$(R + R_2)^2 = (R + n/2 + g_1)^2 + (r \sin \alpha + n/2 + R)^2, \text{ 从而,}$$

$$R_2 = [(R + n/2 + g_1)^2 + (n/2 + r \sin \alpha)^2 - R^2] / (2R - n - 2r \sin \alpha)$$

其中, $r = R - n/2$

解三角形得 $\sin \beta = (R + n/2 + g_1)/(R + R_2)$, 从而,

$$\beta = \arcsin((R + n/2 + g_1)/(R + R_2));$$

固定 α (一般取 $\alpha = 45^\circ$), R_2 已知, 给定 R, n, α 和 g_1 ,

把 $r = R - n/2$ 代入上式, 可求得 R_2, β 。

L_1 是否大于等于 $L = (1 - \cos \alpha)(R - n/2) + n$;

$$g_2 = n/2 + R_2(1 - \cos \beta);$$

$$g_3 = g_2 + (n \cos \beta)/2 = n/2 + R_2(1 - \cos \beta) + n/2 \cos \beta \leq M;$$

一些限制:

$$|(R + n/2 + g_1)/(R + R_2)| \leq 1, 0 < \beta < \pi/2;$$

$$j = R + n/2 + g_1 = (R + R_2) \sin \beta$$

则

联立以上各式可得:

$$[R(1 + \cos \beta) - g_3 + R_2(1 - \cos \beta)](1 - \cos \alpha) = L(1 + \cos \beta)$$

因已知 m, n , 固定 L, g_1, g_3 ,

$$\text{可知: } R_2 = [R(1 - \sin \beta) + n/2 + g_1] / \sin \beta = (R + n/2 + g_1) / \sin \beta - R,$$

$L = (1 - \cos \alpha)(R - n/2) + n$. 固定 α , 则 L 确定。

$$\text{而 } \beta = \arcsin((R + n/2 + g_1)/(R + R_2))$$

则汽车位置 $(g_1, 0)$ ，第一转角为 β ，第二转角为 $\pi/2 - \beta$

五、模型优化与偏差分析

如图 5.1-1 所示，根据已建立的模型有最优路线如图中红色路线所示，结合实际（包括驾驶员技术、周围障碍，路况等因素）考虑，实际汽车行进轨迹可能不按照预先设定的最优路线行驶。例如，图中黄色实线所示的偏离后的路线，在实际系统中，我们必须对这种类似情况进行考虑，使得设计出的模型符合实际问题需求。

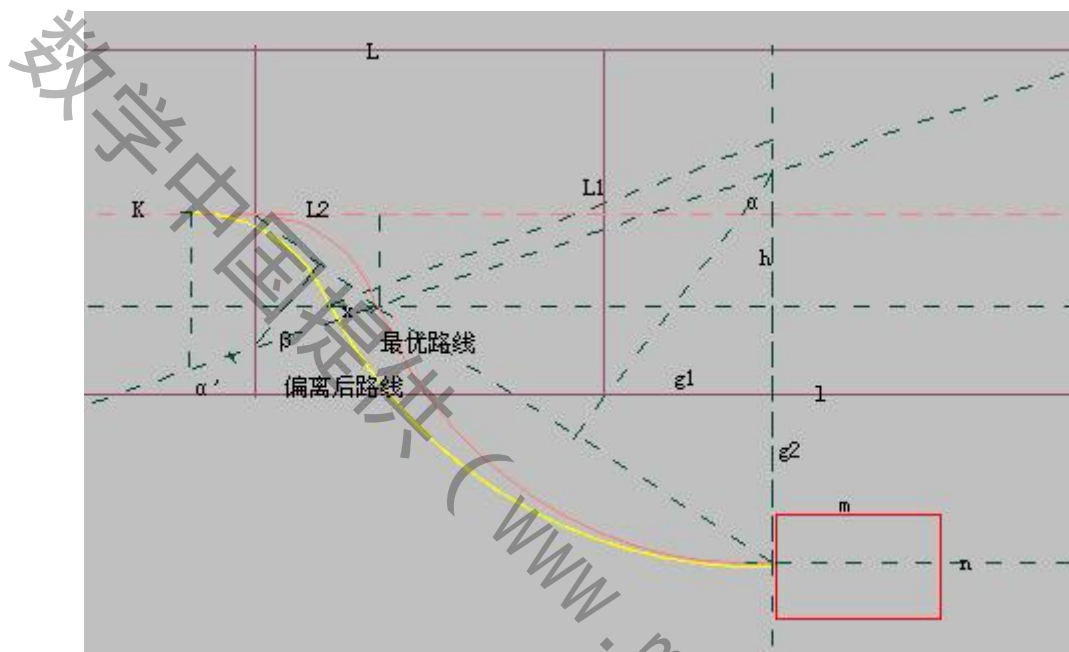


图 5.1-1

对图示偏差路线分析有：

$$l = K/2 - h + g_2 + n/2$$

$$\text{汽车行驶的水平距离 } L_1 = \sqrt{R^2 - (R-l)^2}$$

$$\text{解三角形得 } \sin \alpha = L_1/R$$

$$\cos \alpha = (R-l)/R$$

$$\text{从而, } \sin(\alpha/2) = \sqrt{l/2R};$$

$$\text{汽车行驶的垂直距离 } L_2 = 2R \cdot \sin(\alpha/2) \cdot \cos(\alpha/2) = R \cdot \sin \alpha;$$

$$h=2R*\sin(\alpha/2)*\sin(\alpha/2);$$

已知： $\cos\alpha$ 的值，则汽车内侧行走圆弧的半径： $r=R-(n/2)$ ， $j=r-g_2=R-(n/2)-g_2$ ，

$$\text{又 } \sin\theta=(j/r); \cos\theta=(g_1/r); g_1^2+j^2=r^2;$$

$$\text{则 } g_1^2=[R-(n/2)]^2-[R-(n/2)-g_2]^2=(2R-n-g_2)*g_2,$$

$$\text{求得 } g_1=\sqrt{g_2*(2R-n-g_2)}$$

$$(K/2)-2R\sin(\alpha/2)*\sin(\alpha/2)+g_2+(n/2)=R-R\cos\alpha,$$

$$\text{则 } (4R-K-2*g_2-n)/(4R)=\cos\alpha$$

$$\text{停车位宽 } L=L_1+L_2=2R*\sin\alpha \quad \text{第一次偏差 } x=h_1-R*\sin\alpha$$

$ch=2*x=2*(h_1-R*\sin\alpha)$ ，其中， h_1 为汽车在水平方向前进的距离。

则 L 可由 R, K, α, g_2, n 确定，偏差 ch 亦可由已知量表示出来（具体表达式见程序一的源代码）。

对于其他模型，即侧位停车模型与非字型停车模型，由于只考虑第一次偏差（见假设），且假设第二次轨迹形状与预设轨迹一致，则在汽车到达第一轨迹与第二轨迹交点处时，设定偏差为 x ，则最终偏差将恰好是 $2*x$ ，也就是说，在由上述模型求出 L_{min} 之后，只要 $L_{min} + 2*x \leq L$ ，汽车就可以顺利进入停车位。

另外，对于非字型停车模型（1）与（2），考虑到汽车在转弯时的最小半径只与车自身的参数有关，并且有相关资料表明，汽车前轮的最大转角大约是 45 度，最小转弯半径为 4.5 米，由此我们可以大胆假设，对于非字型停车模型（1）与（2），可以认为驾驶员在实际驾驶时不会出现使得汽车不能顺利驶入停车位的较大误差，即这两个模型与实际情况很相似。

对于需要多次倒车才能顺利进入停车位的情况，我们考虑至多倒车五次，超过五次视为无法顺利进入停车位。多次倒车看似复杂，实质上就是对汽车行进方向的调整，假设只有在产生偏差时我们才多次倒车，以调整方向，使之回归到预定轨迹上去。则我们可将此模型总结如下：多次倒车（大于等于 2 次），用以修正偏差，使汽车在产生偏差后走上一条预设轨迹，以便后续的停车工作。

六、模型的评价与改进

报名号#1195

6.1 模型的优点：

(1)在代入可测数据后，各模型能计算出在对应停车位停车的可实现性、偏差范围、最优路径。且能在偏差范围内，计算出新的最优路径。人性化理念强。

(2)规定了倒车次数最多为五次，从而排除了以不规则的混乱的轨迹进行倒车的情况。

(3)把汽车停车问题抽象为平面几何问题，便于讨论研究。

(4)该模型层层深入，分类明确，并运用程序输出数据，使驾驶员可以顺利地停车入位。

6.2 模型的缺点：

(1)经模型确定的最优路径，实际操作起来对驾驶员技术要求较高。

(2)该模型在误差估计方面做得不够好，没有详细考虑偏差对停车的影响。可在此方面加以改进。

七、 参考资料

[1]姜启源、谢金星、叶俊编，数学模型，北京：高等教育出版社，2003.8

[2]H.M.Deitel、P.J.Deitel 著，薛万鹏等译，C how to program，北京：机械工业出版社，2000.7

[3] 匿名，四轮驱动，<http://zhidao.baidu.com/question/47384029>，2010-4-24

[4]崔衍渠，停车入库秘诀，
<http://hi.baidu.com/teyqiu/blog/item/a286800a2a32093cb1351d90.html>，
2010-4-25

[5]sunlvhao, 标准停车位的尺寸，
<http://wenda.tianya.cn/wenda/thread?tid=50bee0489da7d889>，2010-4-26

八、 附录

附录 1：相关程序源代码

程序一

```
/*该程序中所有需要输入的值都是由车载设备测量而得，由于这个源程序未设计与设备的接口，故所有有设备测量而得的值暂且均由手动输入，以检验程序*/
```

```
/*汽车进入车位的最优路线选择程序*/
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
#define pi 3.1415926
```

```
void main() {
```

```
    int autochoicel, flag, signal;
```

```
    void parking_A();
```


报名号#1195

```
void parking_B();
void parking_C();
for(flag=0;flag<5;flag++){

printf("Input the kind of the parking space.\n");
printf("1 for type one , 2 for type fei ,3 for ceru.\n");
scanf("%d",&autochoicel);

switch(autochoicel){                                /*选择某种车位*/

case 1: parking_A();break;
case 2: parking_B();break;
case 3: parking_C();break;

}

printf("There will be a signal 0 or 1 from the sensor.\n");
scanf("%d",&signal);                                /*从外部设
备得到信号判断停车是否到位*/
if(signal==1);
else{
    flag=4;
    printf("The car has been parked well.\n");
}
}
}

//一字行车位停车选择
void parking_A(){

double R,k,n,a1,g1,L,g2,m,temp;

printf("You choosed type one to set the car.\n");
printf("\n");
printf("Input smallest turning radius, the width of the parking space, ");
printf("the width of the car,");
printf("the distance the car between the car and left/right side of the
parking space,");
printf("the length of the parking space,the length of the car.\n");
scanf("%lf %lf %lf %lf %lf %lf", &R, &k, &n, &g2, &L, &m);

g1=sqrt( g2*(2*R-n-g2) );
temp=(4*R-(k+n)-2*g2)/(4*R);
a1=acos( temp );
```

```
    a1=a1/pi*180;

    printf("Turn %lf at the position of (%lf,%lf). \n", a1, g1, g2);

}

//非字行停车选择
void parking_B() {

    void parking_B_forward();
    void parking_B_ren1();
    void parking_B_ren2();
    void parking_B_backward();

    int autochoice2;
    printf("choice the way of 'fei'.\n Input 1 for kind1,2 for kind2 ,3 for
kind3,4 for kind4.\n");
    scanf("%d",&autochoice2);

    switch(autochoice2) {

        case 1: parking_B_forward(); break;
        case 2: parking_B_ren1(); break;
        case 3: parking_B_ren2(); break;
        case 4: parking_B_backward(); break;
    }
}

//非字行停车选择子函数—直接转入'
void parking_B_forward() {

    double M, R, n, g1, g2=0;
    printf("Input the width of the road and the turning radius and the width of
the car:\n");
    scanf("%lf %lf %lf", &M, &R, &n);

    if(M>=R+n/2.000000) {

        printf("The way you choose is OK\n");
        g1=R-n/2.000000;
```

报名号#1195

```
printf("You should turn 90' at the position of (%lf,%lf)\n", g1, g2);
}
}
```

//非字形停车选择子函数--倒退转入'

```
void parking_B_backward() {
    double Lmin, L, R, n, g1, g2;

    printf("Input the turning radius, the width of the car ");
    printf("the width bettween the car and the parking space and the width of the space.\n");
    scanf("%lf %lf %lf %lf", &R, &n, &g1, &L);
    Lmin=R+n/2.000000-g1;

    if(L>=Lmin) {

        printf("The way you choose is OK.\n");

        g2=(R-n/2.000000)*(1-sqrt(1-(g1*g1/((R-n/2.000000)*(R-n/2.000000)))));
        printf("You should turn 90' at the position of(%lf,%lf)\n", g1, g2);

    }

    else
        printf("The way you choose is not allowed. Please try again.\n");

}
```

//非字形停车选择子函数--局部人字型

```
void parking_B_ren1() {
    double g1, g2=0, g3, R, n, a1, b1, R2, Lmin, L, r, M;

    printf("Input the distance between the car and the parking space,");
    printf("the smallest turing radius and the width of the car, and the known angle.\n");
    printf("the width of the road and the parking space.\n");
    scanf("%lf %lf %lf %lf %lf %lf", &g1, &R, &n, &a1, &M, &L);

    r=R-n/2.000000;
    a1=a1/180.000000*pi;
    R2=((R+n/2.000000+g1)*(R+n/2.000000+g1)
    +(n/2.000000+r*sin(a1))*(n/2.000000+r*sin(a1))-R*R)/(2*R-n-2*r*sin(a1));
```

报名号#1195

```
b1=asin((R+n/2.00000+g1)/(R+R2));
b1=b1/180.0000*pi;
g3=n/2.00000+R2*(1-cos(b1))+n/2.00000*cos(b1);
Lmin=(1-cos(a1))*(R-n/2.00000)+n;
if(L>=Lmin && g3<=M) {

    printf("The way you choose is OK.\n");
    printf("You should turn %lf at the position (%lf,%lf).", b1, g1, g2);
}
else
    printf("The way you choose is not allowed.\n");
}

//非字形选择子函数—局部人字形
void parking_B_ren2() {
    double n, R, Lmin, L, g1, g2=0, M, a1;
    printf("Input the width of the car and the road, the smallest of the turning radius, the real and smallest width of the parking space.\n");
    scanf("%lf %lf %lf %lf", &n, &M, &R, &L, &Lmin);
    g1=(sqrt(3)-1)*R-n/2.000000;
    if(Lmin<=L && R/2.00000+0.75*n<=M) {

        a1=60;
        printf("The way you choosed is OK.\n");
        printf("You should turn %lf at the position (%lf,%lf).", a1, g1, g2);
    }
    else
        printf("The way you choosed is not allowed.Please try again.\n");
}

//侧入停车方式
void parking_C() {
    double Lmin, L, a1, a, g1, g2, R, n, Rmin;

    printf("You choosed the type of ceru.\n");
    printf("Input the smallest turning radius, the width of the car, the angle of the space.\n");
    scanf("%lf %lf %lf", &Rmin, &n, &a1);
    printf("Input the width of the space, the data of g1,g2.\n");
    scanf("%lf %lf %lf", &L, &g1, &g2);
```

报名号#1195

```

a1=a1/180.000000*pi;
Lmin=Rmin+n/2.000000-(Rmin-n/2.000000-g2)*sin(a1)-g1*sin(a1);
a=L-n;

    if(L>=Lmin){
        R=(a+n/2+sin(a1)*g1-(n/2+g2)*cos(a1))/(1-cos(a1));
        printf("The way you choose is OK.\n");
        printf("You should turn %lf at the
position(%lf,%lf).\n",a1/pi*180.000,g1,g2);
    }
    else
        printf("The way you choose is not allowed. Please try again.\n");
}

```

程序二

```

// 一字形车位停车模型优化
#include <stdio.h>
#include <math.h>
#define pi 3.1415926
void main(){
    double R,k,n,a1,g1,L,g2,m,temp,h,ch,x,Lmin;

    printf("Input smallest turning radius, the width of the parking space, ");
    printf("the width of the car,");
    printf("the distance the car between the car and left/right side of the
parking space,");
    printf("the length of the parking space,the length of the car.\n");
    printf("the distance the car has moved.\n"); /*增加一个
水平前行的距离*/
    scanf("%lf %lf %lf %lf %lf %lf %lf", &R, &k, &n, &g2, &L, &m, &h);

    g1=sqrt(g2*(2*R-n-g2));
    temp=(4*R-(k+n)-2*g2)/(4*R);
    a1=acos(temp);
    a1=a1/pi*180;
    Lmin=2*R*sqrt(1-temp*temp);
    x=h-R*sqrt(1-temp*temp);
    ch=2*x;

    if(ch+Lmin<=L)

```

```
printf("The way you choose is OK!\n");

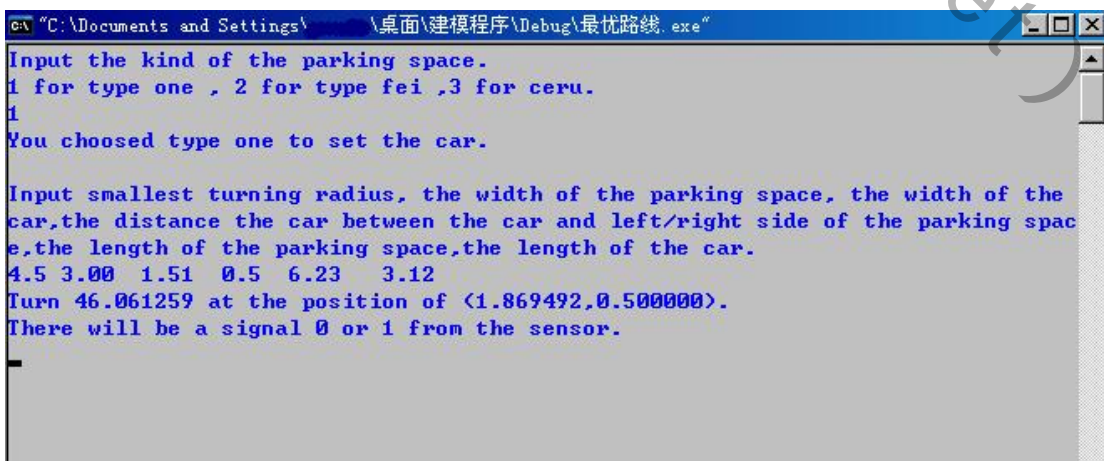
printf("Turn %lf at the position of (%lf,%lf).\n",a1,g1,g2);
}
```

附录 2

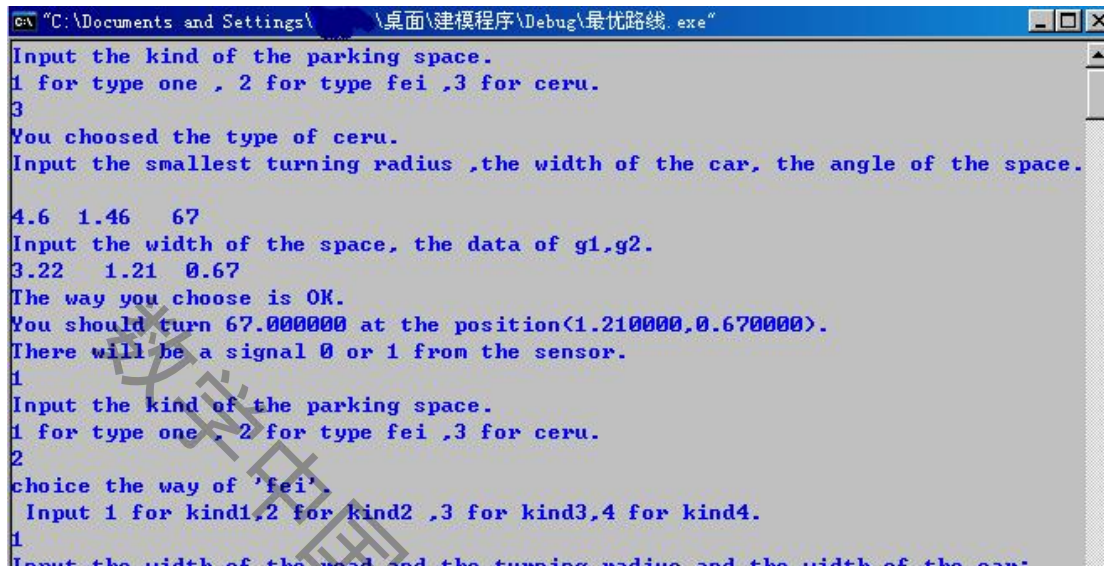
(1) 非字形停车模型程序检测结果截图



(2) 一字型停车模型程序检测结果截图



(3) 侧入停车模型程序检测结果截图



```
C:\Documents and Settings\...\桌面\建模程序\Debug\最优路线.exe
Input the kind of the parking space.
1 for type one , 2 for type fei ,3 for ceru.
3
You choosed the type of ceru.
Input the smallest turning radius ,the width of the car, the angle of the space.
4.6 1.46 67
Input the width of the space, the data of g1,g2.
3.22 1.21 0.67
The way you choose is OK.
You should turn 67.000000 at the position(1.210000,0.670000).
There will be a signal 0 or 1 from the sensor.
1
Input the kind of the parking space.
1 for type one , 2 for type fei ,3 for ceru.
2
choice the way of 'fei'
Input 1 for kind1,2 for kind2 ,3 for kind3,4 for kind4.
1
Input the width of the road and the turning radius and the width of the car:
```