

基于视觉传感器的智能车控制算法设计

王建¹, 张晓炜², 杨锦³, 咎鑫⁴, 刘小勇⁴

(1. 中国科学技术大学 自动化系, 安徽 合肥 230027;

2. 西安交通大学 人工智能与机器人研究所, 陕西 西安 710049;

3. 西安交通大学 金禾经济研究中心, 陕西 西安 710049;

4. 西安交通大学 电子与信息工程学院, 陕西 西安 710049)

摘要: 以第四届“飞思卡尔”杯全国大学生智能汽车大赛为背景, 介绍了基于视觉传感器的智能车控制算法, 包括方向控制和速度控制。在 PID 算法或模糊控制算法几乎为所有参赛队伍所采用的背景下, 提出了“最优曲率法”, 并使用与之配合的“贪婪路径规划”算法。该小车在复杂赛道上的平均速度达 3.3 m/s, 其控制算法设计对智能车设计有借鉴意义。

关键词: 智能车; 飞思卡尔; 最优曲率算法; 路径规划

中图分类号: TP242.6

文献标识码: A

文章编号: 1674-7720(2010)20-0074-04

Design of control algorithms for smart car based on vision

WANG Jian¹, ZHANG Xiao Wei², YANG Jin³, ZAN Xin⁴, LIU Xiao Yong⁴

(1. Department of Automation, University of Science and Technology of China, Hefei 230027, China;

2. Institute of Artificial Intelligence and Robots, Xi'an Jiaotong University, Xi'an 710049, China;

3. Jinhe Center for Economic Research, Xi'an Jiaotong University, Xi'an 710049, China;

4. The School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: It is under the background of the 4th “Freescale cup national undergraduates smart car contest”. It describes the control algorithms including direction control algorithms and speed control algorithms which were used in the author's team's car, and proposed “optimal curvature algorithm” and developed a co-ordinated algorithm that was called “greedy path planning” when almost all the other teams used PID algorithm or fuzzy control algorithm. The car's average speed was up to 3.3 m/s in a complex runway. The control algorithms design can give directions for further design.

Key words: smart car; Freescale; optimal curvature algorithm; path planning

以飞思卡尔智能车大赛为背景, 制作了一个能巡线快速行驶的小车。

1 巡线控制算法

1.1 运动模型

智能车的运动控制是通过控制驱动马达和伺服舵机实现的。在理想情况下, 可以认为 4 个轮胎绕着同一个圆心做纯滚动, 其离散运动模型公式为:

$$x(t+1) = x(t) + V(t) \times \Delta t \times \sin[\text{angle}(t) + (W(t) \times \Delta t) / 2] \quad (1)$$

$$y(t+1) = y(t) + V(t) \times \Delta t \times \cos[\text{angle}(t) + (W(t) \times \Delta t) / 2] \quad (2)$$

$$\text{angle}(t+1) = \text{angle}(t) + W(t) \times \Delta t \quad (3)$$

其中 V 是线速度, W 为角速度, $x(t)$ 、 $y(t)$ 为小车的坐标, $\text{angle}(t)$ 为智能车的方向角 (以小车前进方向为起始线,

逆时针为正), Δt 为采样周期。模型如图 1 所示。

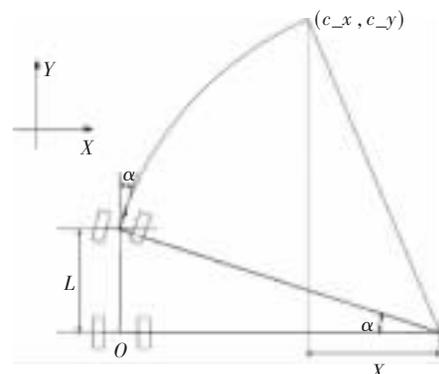


图 1 小车运动模型

技术与方法 Technique and Method

由于模型车的运动速度较高,在行驶过程中轮胎与赛道会发生一定侧滑,因此还需要引进侧滑修正。

1.2 舵机控制

方向控制是控制策略的难点。传统的PID算法^[1,2]不需要对控制对象做细致分析,只需根据控制效果对参数进行调整^[3]。但由于控制器的输入量均对输出量产生影响,且高度耦合,很难根据控制效果准确地对参数进行修正,因此参数的整定工作十分繁重。此外,当车的状态不同时(如低速和高速时),最佳参数也有很大不同,这给进一步调试带来了困难。

本文采用“最优曲率法”。控制思想是:模型车在一个控制周期内的运动轨迹可以近似为一段圆弧,整个运动轨迹可以看做由小段圆弧连接成的曲线。在假设线速度恒定的前提下,给定路径上的一个点就能确定出智能车到达该点所应具有舵机角度^[4]。在每个控制周期内选择路径上的一点(或若干点),根据预先建立的圆弧运动模型计算出模型车在接下来一个(或若干个)运动周期内要到达选定点的理想状态,此时该控制周期的输出量就是在未来几个控制周期内使模型车的实际状态尽可能逼近理想状态的控制量。

以车的后桥中点为原点建立坐标系,车的中轴为y轴,其正向为车的前方。设 (c_x, c_y) 为路径上的点在该坐标系中的坐标。则控制量的计算公式为^[5]:

$$r_b = (c_x^2 + c_y^2 - L^2) / (2 \times c_x) \quad (4)$$

$$\text{wheel_angle} = \arctan(L / r_b) \quad (5)$$

$$\text{rudder} = \text{wheel_angle} \times m \quad (6)$$

其中 r_b 为后轮的转弯半径, L 为车的前桥与后桥的距离, wheel_angle 为车前轮的理想角度, rudder 为舵机的PWM输出值, m 为将 wheel_angle 转换为 rudder 的系数(由实验测得)。

公式(4)证明:如图1所示, $r_b^2 + L^2 = c_x^2 + c_y^2$; $r_b = c_x + x$; 以上两式联立即得公式(4)。

这种方法在车速较低时可以取得很好的效果,当车速较高时,有很多因素对控制效果的影响会显著增大,需要作出适当的补偿。其中主要因素是控制量的实际输出时刻与理想输出时刻的延时,这个延时是由图像处理和控制量的计算都需要一定时间造成的。对此,本文采取的方法是根据车的当前状态估算出控制量的输出时刻的路径信息,用这个路径信息计算控制量。

设延时期间车的角度变化量为 Δw ,则横纵坐标变化量分别为 Δx 、 Δy ,则当前时刻的前瞻点 (c_x, c_y) 在经过延时 Δt 后的坐标为 (p_x, p_y) ,用 p_x 、 p_y 计算控制量。设车速为 v ,当前车前轮转角为 β ,车长为 L ,则计算公式为:

$$\Delta w = v \times \Delta t \times \tan(\beta) / L \quad (7)$$

$$\Delta x = v \times \Delta t \times \sin(\Delta w / 2) \quad (8)$$

$$\Delta y = v \times \Delta t \times \cos(\Delta w / 2) \quad (9)$$

$$p_x = (c_x - \Delta x) \times \cos(\Delta w) - (c_y - \Delta y) \times \sin(\Delta w) \quad (10)$$

$$p_y = (c_x - \Delta x) \times \sin(\Delta w) + (c_y - \Delta y) \times \cos(\Delta w) \quad (11)$$

证明公式(7)~公式(11):

$$v \times \Delta t = R \times \Delta w;$$

$$R = L / \tan \beta, \text{得公式(7);}$$

在直角三角形ABC中, $\angle ABC = (\pi - \Delta w) / 2$, $\Delta x = AB \times \cos \angle ABC = AB \times \sin(\Delta w / 2)$, $\Delta y = AB \times \sin \angle ABC = AB \times \cos(\Delta w / 2)$,而AB即为 $v \times \Delta t$,即得公式(8)和公式(9)。

$$p_x = AF = AE \times \cos(\Delta w) - GE \times \sin(\Delta w)$$

$p_y = GF = AE \times \sin(\Delta w) + GE \times \cos(\Delta w)$; 而 $AE = c_x - \Delta x$, $GE = c_y - \Delta y$, 由此得公式(10)和公式(11),如图2所示。

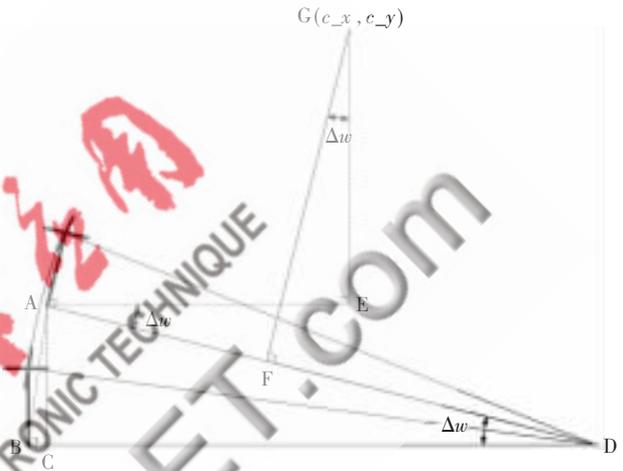


图2 公式(7)~(11)证明用图

另一个重要因素是侧滑。智能车在行驶过程中存在侧滑,可以通过侧滑角作补偿。设车轮的角度为 θ ,加入侧滑后的等效车轮角度为 θ' ,本文假设二者符合如下关系:

$$\theta' = \theta \times (1 - \text{speed} / \text{max_speed}) \quad (12)$$

在计算出车轮应当具有的实际角度 θ' 后,用上式求出 θ ,进而求出PWM的值。这样就可以补偿侧滑的影响。

还有一个重要因素是,当车速较高时,舵机的响应速度相对较慢。要克服这个问题,应使舵机提前做出反应。因此需要在控制器的输出量中引入路径的趋势。本文用前瞻点处的斜率 k 表示路径的趋势,在原控制量 θ 的基础上增加一个与路径趋势相关的分量 $\theta'' = \text{para} \times v \times k$,其中 para 通过实验结果自动调整,如果需要,可以将 para 设为和其他变量相关的函数。

经过上述改进之后,最优曲率法在车速较高的时候依然能够取得很好的控制效果。此部分程序流程图如图3所示。

该算法的优点是参变量较少,且相互间的耦合很

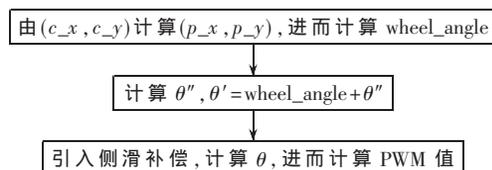


图3 舵机控制流程图

技术与方法 Technique and Method

小,对控制效果的影响较为独立,可以根据实验结果快速确定调整量。只要对参数进行适当调整,就能取得较好的控制效果。

1.3 速度控制

在速度决策方面,首先尝试了基于有效前瞻的速度控制^[6],即 $v = \text{基准值} + \text{有效前瞻量} \times \text{系数}$ 。这个策略的好处是简单有效。但事实上智能车通过路线的最佳速度和前瞻量并不呈简单的线性关系。如果处于弯道中,尤其是“十”字弯中时,由于车的阻力增大,车速也大大下降,同时由于车体运行较稳定,车轮不会发生大幅度转向,侧滑的风险很小,这时可以提速;当出弯时,如果检测或预测到前方较平缓,可以放心加速。由于局部快不一定整体快,因此还要做好衔接,使得整个跑的过程中各速度间过渡圆滑。基于这些想法,本文在前面的速度决策上做了改进,在保证安全性的前提下提高了整体速度。

1.4 仿真调试

为了便于分析控制效果,本文搭建了一个 Matlab 仿真环境。在这个仿真环境中,车模的参数和动态响应特性都是参照实验结果设计的,并引入了侧滑、延时、舵角死区等很多干扰因素,可以在很大程度上反映算法的真实情况。仿真界面如图 4 所示。

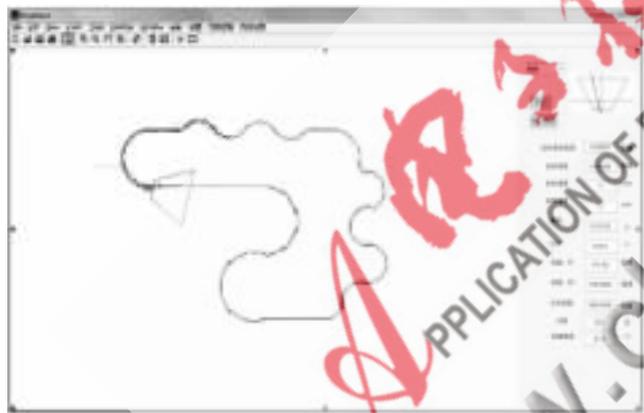


图 4 仿真界面

在实际的调试过程中,本文首先在仿真环境下设计算法,并分析模型车通过各种跑道时所需要的理想的舵机角度,测试通过后将算法移植到模型车上。如果实际控制效果与仿真结果差别较大,则将控制算法的执行情况与理想情况下的期望值进行比较,针对差别有目的地作出改进。这在很大程度上提高了调试效率。

2 贪婪路径规划

2.1 优化问题

优化问题中有两个约束条件:

(1)车不出边界。

(2)小车行驶路径最短。主要是弯道切弯行驶,大小 S 弯走最优路径。

对于一般问题,全局最优解是最好的结果,但是智能小车行驶的跑道是随机的,并且小车传感器获得的道

路信息有限,要得到全局最优解很困难,所以根据贪婪算法思想,在局部的每一点获得最优解来逼近全局最优解。贪婪路径规划就是给出一些小车行驶路径点,保证小车行驶时最大程度地满足以上两个优化条件。

2.2 最优解分析求解

比赛用的路径只有两种:直道和圆弧形。关键在圆弧形,分单圆弧形和多圆弧形,单圆弧形指衔接两个直道的一个弯道,多圆弧形指 S 形弯道。

首先,分析单圆弧形。单圆弧形示意图如图 5 所示,已知路径宽为 60 cm, T 点为小车传感器识别出的最远前瞻路径点。



图 5 单圆弧形示意图

如果车向着 T 点行驶去,车有可能超出跑道。所以需要计算出最远前瞻点和车头连线(OT)与路径圆弧形(弧 OT)之间的最远距离 D,如果 D 超出 20 cm(车模宽 20 cm,当车模恰好压跑道边时,车模中心与路径线的距离是 20 cm),就不满足约束条件(1)。

当 D 超出 20 cm 时,可以将前瞻点 T 后移,计算出使 $D = 20 \text{ cm}$ 的 T' 点,如图 5 所示。在给定的直线上根据小车的速度可以选取一点作为贪婪规划的解路径点。速度快时,选点前移,反之则后移。

其次,对于多圆弧形 S 形道,需要找到两个圆形的连接点,称为转折点。图 6 中 T 点为转折点。S 形道路与一般弯道路径不同,不能用单圆弧形的方法求出最优路径点,因为最远前瞻点可能跨两个圆弧形,求取距离 D 困难,最重要的是用单圆弧形的方法利用最远前瞻点求出的解不是最优的解。可以发现在圆弧形转折点前面的圆弧形(弧 OT)是一个单圆弧形,用这段单圆弧形来求解才是合理的。利用最小二乘法可以对每一个点求出一个斜率,其中转折点的特点是转折点是斜率曲线的极值点。可以发

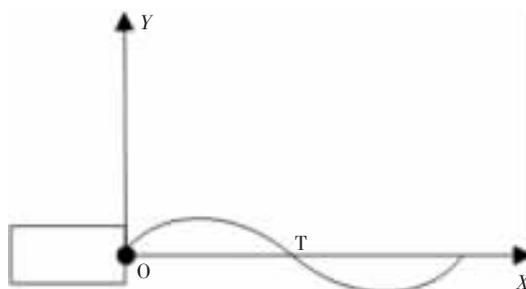


图 6 多圆弧形 S 形道

技术与方法 Technique and Method

现,对于 90° 小 S 弯, D 等于 14.65 cm, 小车直线通过, 而 120° 中等 S 弯, D 等于 25 cm。

需要注意, 由于传感器的视野和图像处理的限制, 有时最远前瞻点只能取到 N 点, 如图 5 所示。这时可以使用广角镜头(实验中选取焦距 $f=2.8$ mm)扩大视野, 提高图像处理算法的精度来使前瞻点尽量靠前, 到达 T 附近。

本文根据转向模型, 提出了“最优曲率”算法及“贪婪路径规划”算法。实验表明, 这两种算法结合起来能使小车更安全、快速地行驶, 可作为除 PID 算法和模糊控制算法以外的第三种选择。

参考文献

- [1] 卓晴, 黄开胜, 邵贝贝, 等. 学做智能车[M]. 北京: 北京航空航天大学出版社, 2007.
- [2] 马艳, 徐淑华, 周建春. 基于摄像头寻迹的智能电动车的设计[J]. 机械与电子, 2009, 27(2): 21-24.
- [3] 艾宁, 瞿少成, 刘冬, 等. 基于 CCD 摄像头的智能车路径识别及跟踪研究[J]. 电子测量技术, 2009, 32(8): 77-80.
- [4] 尹念东, 余群. 基于横向预瞄偏差的驾驶员前视轨迹控制模型[J]. 汽车工程, 2002, 24(4): 287-289.
- [5] 魏玉虎, 石琛宇, 姜建钊, 等. 基于视觉的智能车转向控制策略[J]. 电子技术应用, 2009, 35(1): 130-134.
- [6] 蔡庆楠, 蔡兴旺, 潘锦洲. 上海大学 S.U.L. 挑战者队技术报告[R], 2009.

(收稿日期: 2010-04-18)

作者简介:

王建, 男, 1987年生, 在读硕士研究生, 主要研究方向: 模式识别与智能系统。

电子技术应用
APPLICATION OF ELECTRONIC TECHNIQUE
www.chinaAET.com