



智能无人系统





课件目录

- 第一章：认识无人驾驶系统
- 第二章：无人驾驶系统的定位方法以及传感器的应用
- 第三章：深度学习和无人驾驶视觉感知
- 第四章：迁移学习和强化学习在无人驾驶中的应用
- 第五章：无人驾驶的规划
- **第六章：车辆模型和高级控制**
- 第七章：无人驾驶的平台介绍和系统安全
- 第八章：多智能体无人系统

参考资料：

《第一本无人驾驶技术书》（刘少山等）

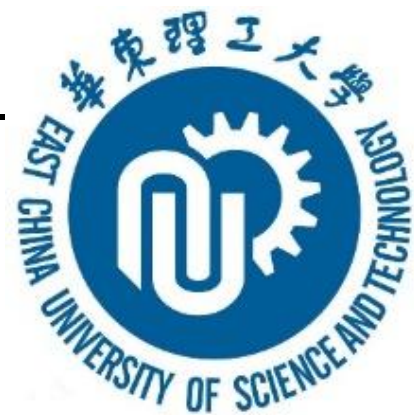
《无人驾驶原理与实践》（申邦泽等）

《无人系统基础》（杨光红、王俊生）



车辆模型和高级控制



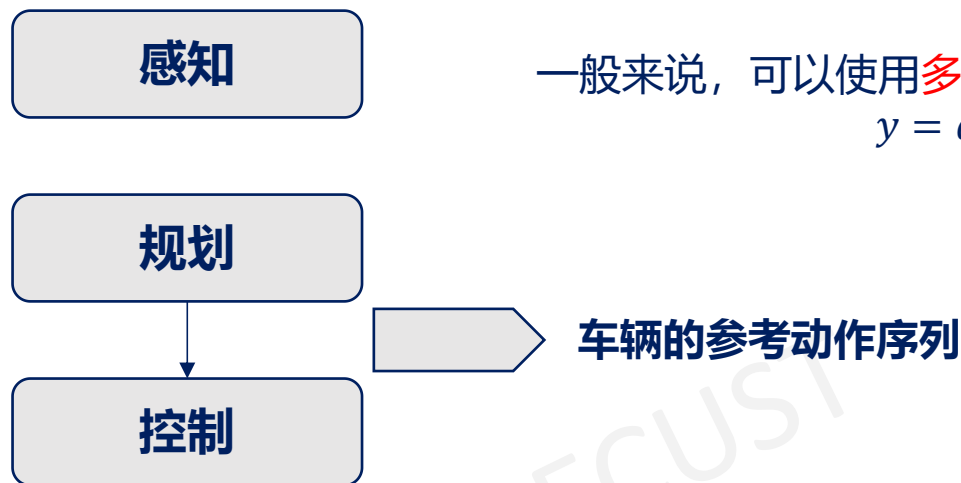


目录

CONTENT

- 1 运动学自行车模型和动力学自行车模型
- 2 无人车控制入门
- 3 基于运动学模型的模型预测控制
- 4 轨迹追踪
- 5 车辆运动学和动力学建模

6.1 运动学自行车模型和动力学自行车模型



一般来说，可以使用**多项式**来描述这个路径曲线：

$$y = ax^3 + bx^2 + cx + d$$

无人车的控制依赖于一项称为**模型预测控制**的技术，这种控制的方法是产生一系列可行的（也就是说实际车辆能够做到的）控制输入，并基于一定的算法（往往是带约束的非线性优化算法）来调整这一系列控制输入，使得损失函数最小化。损失函数的求解依赖于车辆运动学或者动力学模型的输出和参考路径、速度的差值。

6.1 运动学自行车模型和动力学自行车模型

6.1.1 自行车模型

自行车模型基于如下几个**假设**:

- 忽略车辆在垂直方向的运动，即假定车辆是一个二维平面上的运动物体；
- 假设车辆的结构就像自行车一样，即车辆的前面两个轮胎拥有一致的角度和转速等，后面的两个轮胎也是如此，那么前后轮胎就可以各用一个轮胎来描述；
- 假设车辆运动与自行车一样，意味着前面的轮胎控制车辆的转角。



6.1 运动学自行车模型和动力学自行车模型

6.1.1 自行车模型

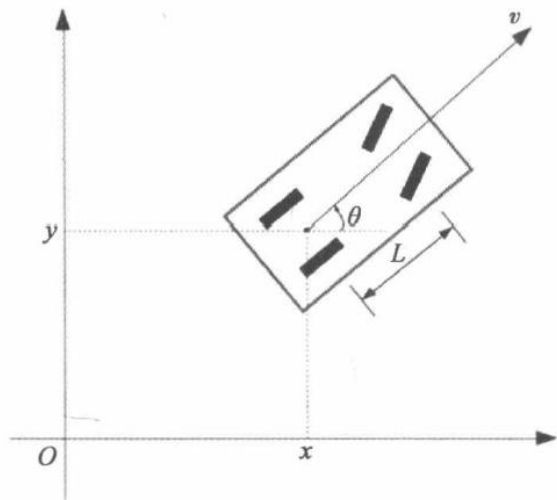


图1.1 二维平面上的车辆描述

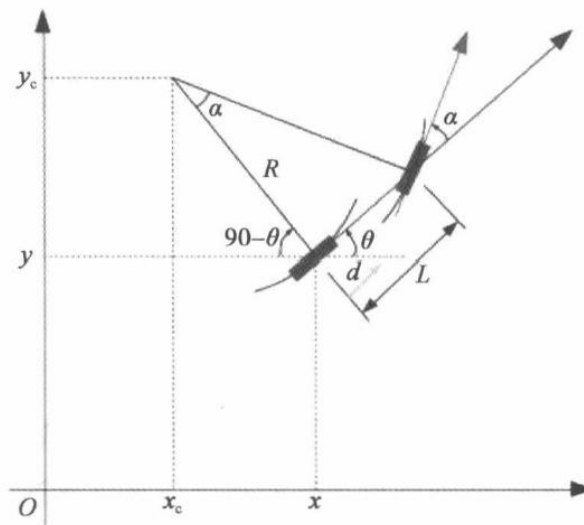


图1.2 车辆的自行车模型

θ : 偏航角 (yaw) (相对于 x 轴的逆时针方向的角度)

v : θ 方向的速度

L : 车辆的轴距 (前后轮胎的距离)

(x, y) : 车辆的坐标

(本例中, 车辆的坐标指车辆后轮的中心坐标, 车辆的速度指车辆后轮中心的速度)

6.1 运动学自行车模型和动力学自行车模型

6.1.2 运动学自行车模型

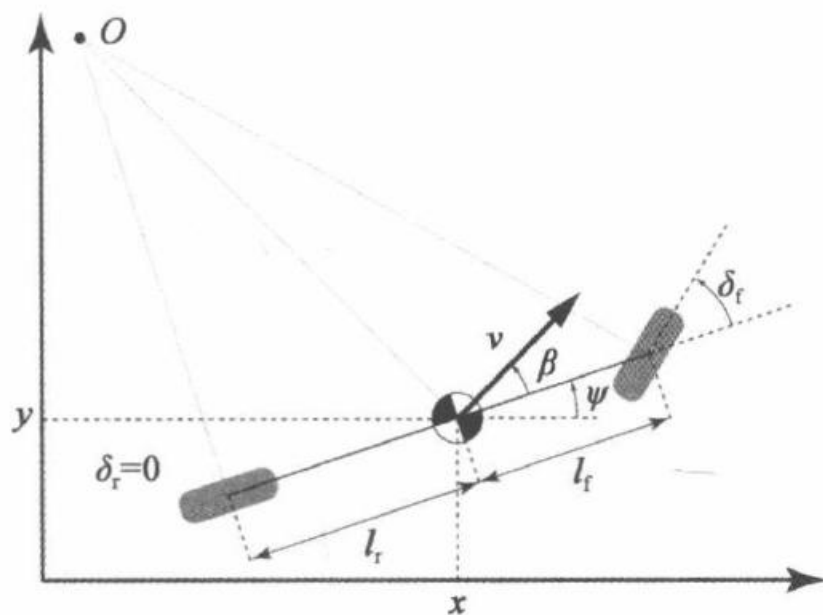


图1.3 一个简单的运动学自行车模型
本例中以车辆质心作为参考点

控制输入(a, δ_f)

- a 表示车辆的加速度，踩油门意味着正的加速度，踩刹车意味着负的加速度
- δ_f 表示方向盘的转角，假定方向盘转角就是前轮当前的转角

6.1 运动学自行车模型和动力学自行车模型

6.1.2 运动学自行车模型

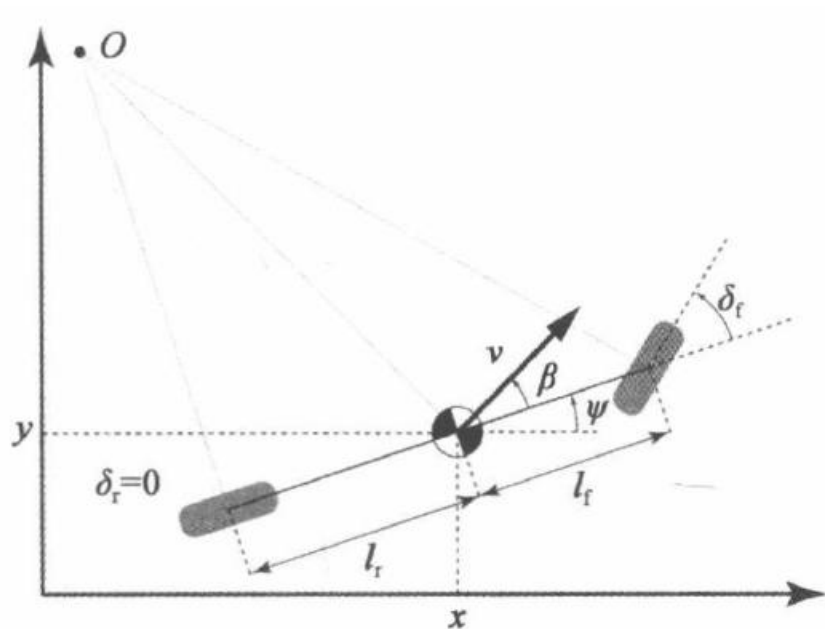


图1.3 一个简单的运动学自行车模型
以车辆质心作为参考点

使用四个状态量描述车辆当前的状态：

x ：车辆当前的 x 坐标

y ：车辆当前的 y 坐标

ψ ：车辆当前的偏航角（用弧度来描述，逆时针方向为正）

v ：车辆的速度

l_f ：前轮到车辆重心的距离

l_r ：后轮到车辆重心的距离

6.1 运动学自行车模型和动力学自行车模型

6.1.2 运动学自行车模型

根据运动学定理，运动学自行车模型中**各个状态量的更新公式**如下：

$$x_{t+1} = x_t + v_t \cos(\psi_t + \beta) \times dt$$

$$y_{t+1} = y_t + v_t \sin(\psi_t + \beta) \times dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{l_r} \sin(\beta) \times dt$$

$$v_{t+1} = v_t + a \times dt$$

其中， dt 表示间隔时间， β 又如下的公式计算可得：

$$\beta = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta_f) \right)$$

自行车模型假定后轮的转角控制输入 $\delta_r = 0$ 。

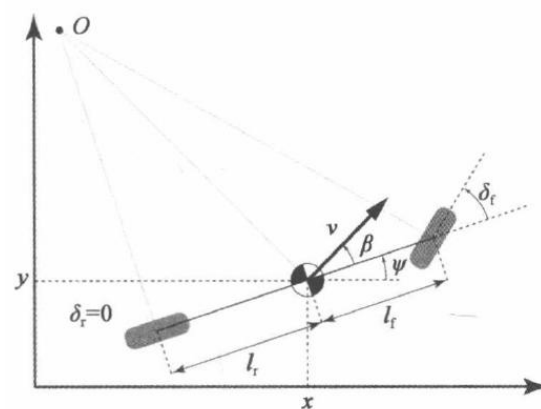


图1.3 一个简单的运动学自行车模型

6.1 运动学自行车模型和动力学自行车模型

6.1.2 动力学自行车模型

- 车辆运动学自行车模型假设：车前轮的方向是车辆当前的速度方向
- 而在车辆运动过程中，当车辆以相对高的速度行驶时，**车轮的方向并不一定是车辆当前的速度方向。**

考虑各种各样的力的作用，大致分为两类：

- **纵向力** (Longitudinal force)：使车辆前后移动的力量
- **侧向力** (Lateral force)：促使车辆横向移动

在力的相互作用过程中，轮胎起着决定性作用（根据物理规律，轮胎是车辆运动的一个重要的力来源）。

6.1 运动学自行车模型和动力学自行车模型

6.1.2 动力学自行车模型

模型的状态量为: $(\dot{x}, \dot{y}, \dot{\psi}, X, Y)$

\dot{x} : 车身的纵向速度

\dot{y} : 车身的侧向速度

$\dot{\psi}$: 偏航角速度

(X, Y) : 车身的当前坐标

$$\ddot{x} = \dot{\psi}\dot{y} + a_x$$

$$\ddot{y} = -\dot{\psi}\dot{x} + \frac{2}{m}(F_{c,f}\cos\delta_f + F_{c,r})$$

$$\ddot{\psi} = \frac{2}{I_z}(l_f F_{c,f} - l_r F_{c,r})$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi$$

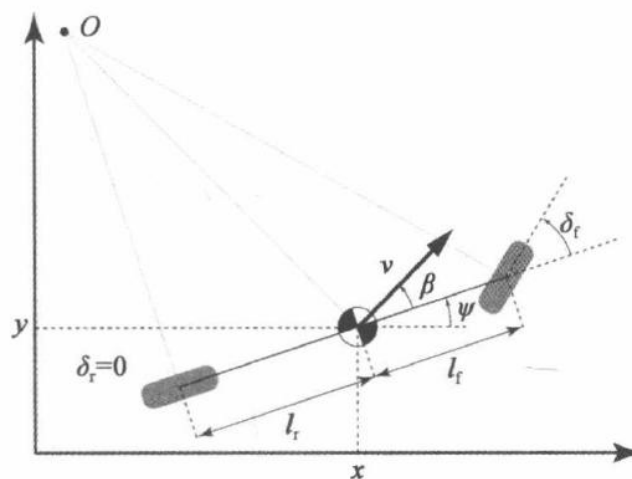


图1.3 一个简单的运动学自行车模型

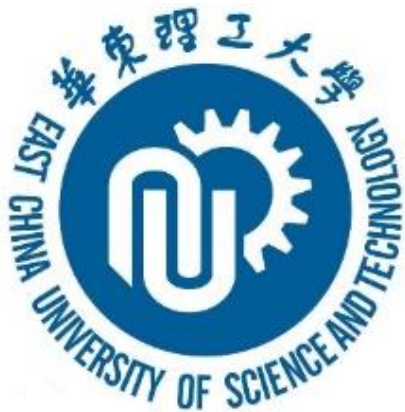
m 表示车辆的质量

I_z 表示车辆的偏航惯性

$F_{c,f}$ 和 $F_{c,r}$ 表示前后轮胎受到的侧向力

$$F_{c,i} = -C_{\alpha_i}\alpha_i$$

其中, α_i 表示轮胎的偏转角, 指轮胎当前的朝向和当前速度的夹角, C_{α_i} 表示轮胎偏转刚度。



目录

CONTENT

- 1 运动学自行车模型和动力学自行车模型
- 2 **无人车控制入门**
- 3 基于运动学模型的模型预测控制
- 4 轨迹追踪
- 5 车辆运动学和动力学建模

6.2 无人车控制入门

6.2.1 为什么需要控制理论

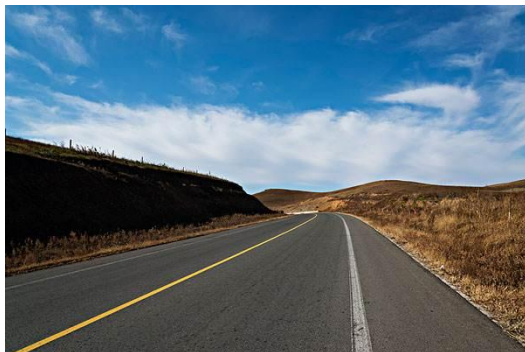


图2.1 弯道行驶真实场景¹

★ 反馈控制

基于环境反馈的控制方法

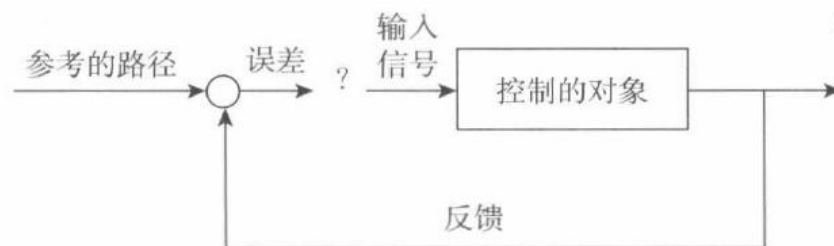


图2.2 反馈控制的一般思路

¹ <https://image.baidu.com/>

6.2 无人车控制入门

6.2.1 为什么需要控制理论

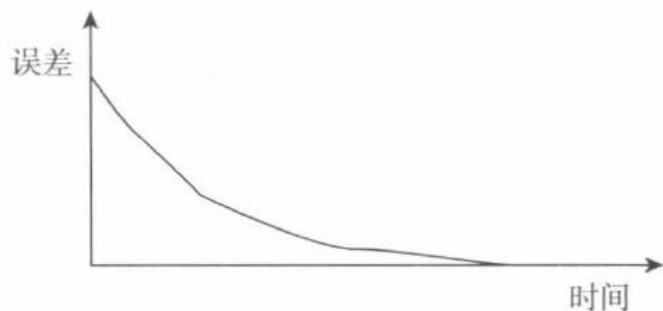


图2.3 误差和时间关系图

控制模块的目的是让无人车能够按照规划好的路径行驶，需要将环境当前的反馈和规划的参考量进行比较，得到当前偏离参考量的误差，并基于这个设计**一定的算法**来产生输出信号，使得误差不断变小，直到为0。

如：PID控制

6.2 无人车控制入门

6.2.2 PID控制

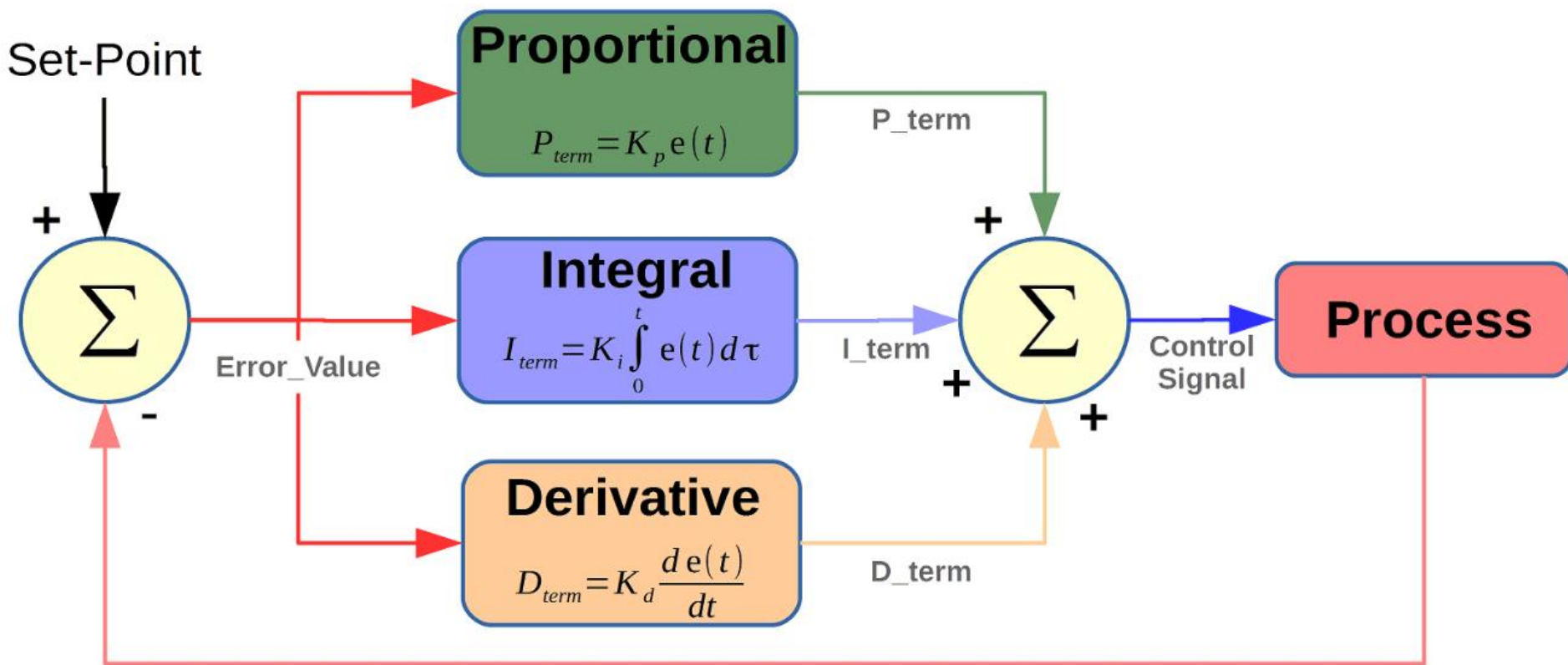


图2.4 用误差来产生控制指令的整个流程

6.2 无人车控制入门

6.2.2 PID控制——P控制



图2.5 车辆和线路相对位置

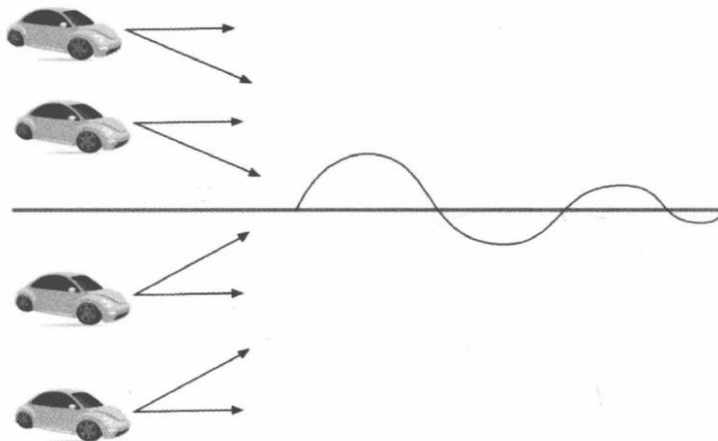


图2.6 按照固定的角度转向情况图示

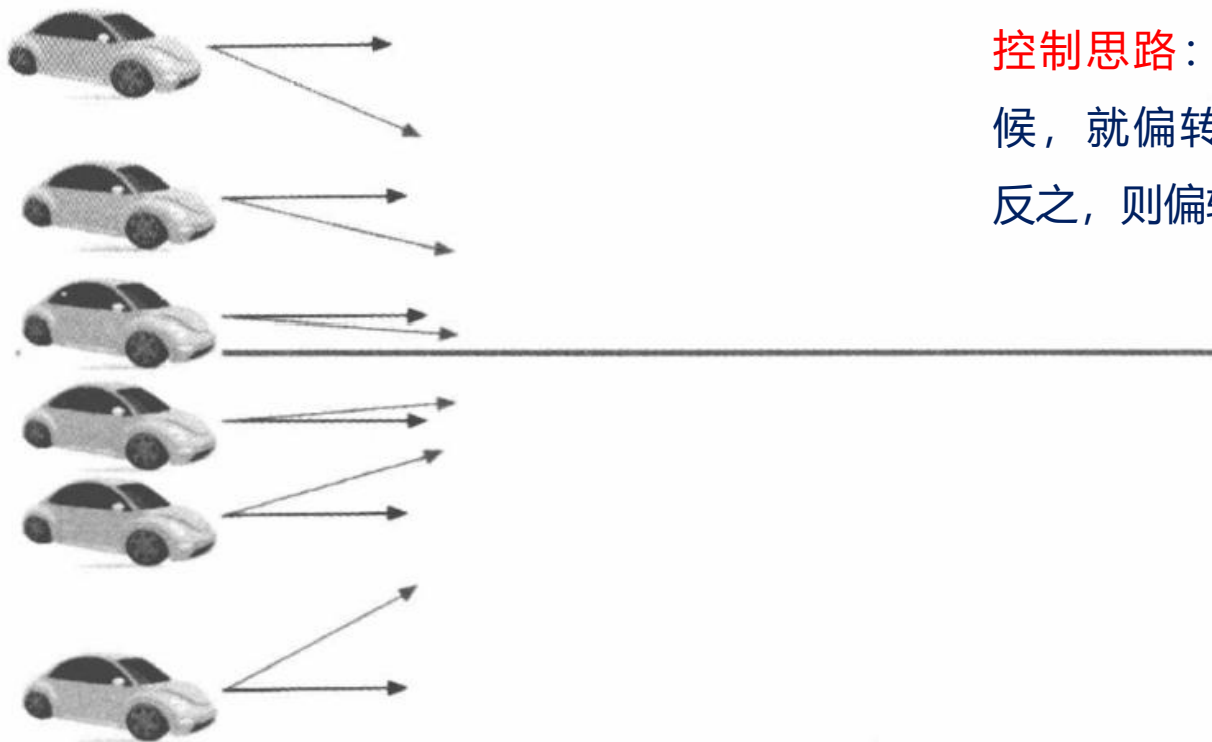
控制输出：车辆的转角

按照固定的角度调整车辆的转向，等值控制是震荡的

比例控制器

6.2 无人车控制入门

6.2.2 PID控制——P控制



控制思路：当偏差大的时候，就偏转更大的角度，反之，则偏转小一点

图2.7 使用比例控制情况图示

6.2 无人车控制入门

6.2.2 PID控制——P控制

误差度量： **偏离航迹误差** (Cross Track Error, CTE)

本例中，CTE是车辆到参考线的距离；

转角表示为：

$$\text{Steering angle} = K_p \cdot e(t)$$

其中， $e(t)$ 表示车辆在 t 时刻的CTE，在P控制器中 K_p 会直接影响实际控制效果。在合理的数值范围内越大，控制的效果越好（即更快地回到参考线附近），但是当本身位置和参考线相距很远且系数较大时，车辆会失去控制。

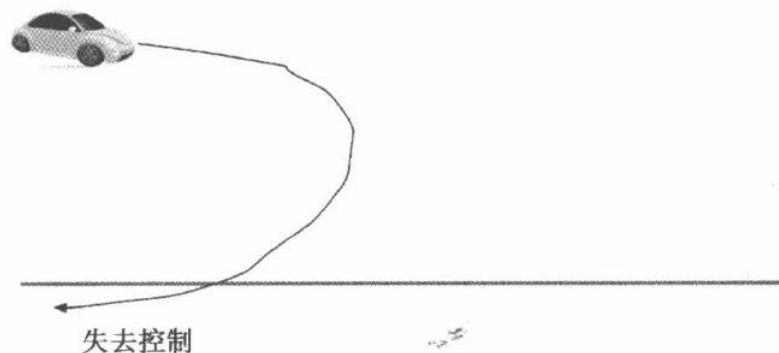


图2.8 K_p 系数较大，车辆失去控制

6.2 无人车控制入门

6.2.2 PID控制——P控制

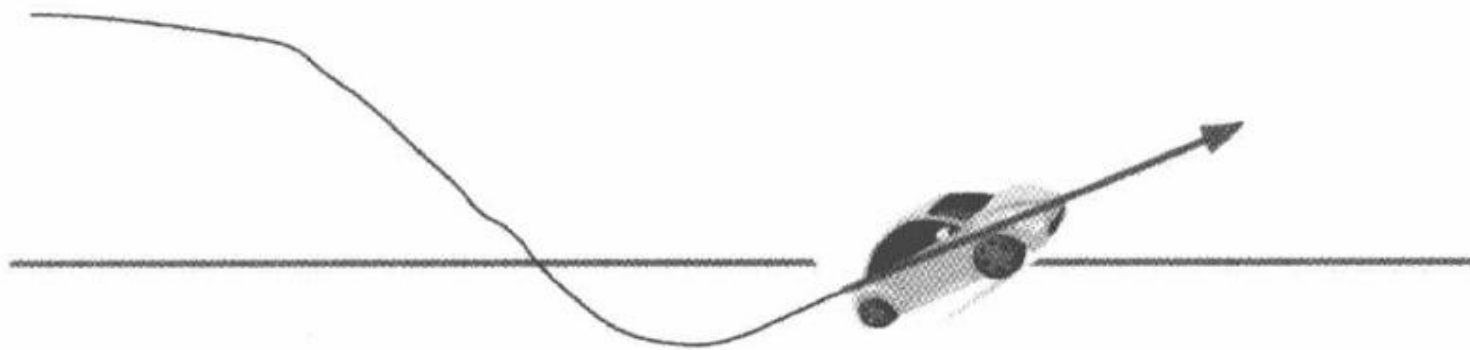


图2.9 车辆已经在参考线上

此时，车辆虽然在参考线上，但是并不是我们希望的状态（它在下一刻就会偏离），但是对P控制器而言，这是理想状态，此时控制转角为0，因此，P控制器会一次又一次超过参考线，为了校正这个问题，需要考虑一个额外的误差项——**CTE变化率**

6.2 无人车控制入门

6.2.2 PID控制——PD控制

CTE的变化率描述了无人车向着参考线方向移动多快。如果无人车一直都完美地在参考线上运动的话，CTE变化率为0，所以这一项可以用导数来表示，此时控制输出变为：

$$\text{Steering angle} = K_p \cdot e(t) + K_d \frac{de(t)}{dt}$$

其中， K_d 是导数项系数，它的大小决定了CTE变化率对于反馈控制的影响

增大**P系数**会增大无人车向着参考线方向运动的倾向

增大**D系数**会增大无人车转角快速变化的“阻力”，从而使得转向更加平缓

使用过大P系数，过小D系数的系统——**欠阻尼**
(无人车将沿着参考线震荡前进)

P系数过小，D系数过大——**过阻尼**
(使得无人车需要较长的时间才能纠正误差)

6.2 无人车控制入门

6.2.2 PID控制——PD控制

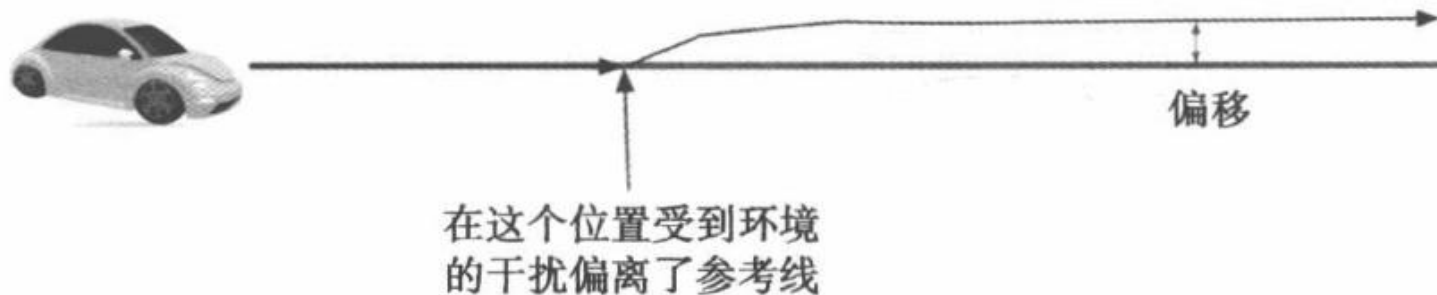


图2.10 环境存在扰动的情况

PD控制器可以保证正常的控制需求，但是当环境存在扰动时，车辆在受力发生轻微偏移后，由于PD控制器中的P项倾向于向参考线方向运动，而D项则尝试抵消这种倾向，造成无人车始终无法沿着参考线运动的问题——**稳态误差**

6.2 无人车控制入门

6.2.2 PID控制——PID控制

将积分项加入控制输出函数中，无人车的转角输出为：

$$\text{Steering angle} = K_p \cdot e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t) dt$$

其中， K_i 是积分项系数，**积分项本质上是车的实际路线到参考线的图形面积**，加入积分项后，控制器会使车辆路线的积分尽可能小，从而避免稳态误差的情况。

积分项系数的大小会影响整个控制**系统的稳定性**，过大的 K_i 会使系统“震荡”运行，过小的 K_i 又会使控制的车辆在遇到扰动后（处于稳态误差状态）需要很长时间才能回到参考线上，这在某些情况下会使车辆处于危险的境况。

6.2 无人车控制入门

6.2.2 PID控制——PID控制

PID控制器的参数整定：

1. 理论计算整定法

它主要是依据系统的数学模型，经过理论计算确定控制器参数。这种方法所得到的计算数据未必可以直接用，还必须通过工程实际进行调整和修改。

2. 工程整定方法

它主要依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。PID控制器参数的工程整定方法，**主要有临界比例法、反应曲线法和衰减法**。三种方法各有其特点，其共同点都是通过试验，然后按照工程经验公式对控制器参数进行整定。但无论采用哪一种方法所得到的控制器参数，都需要在实际运行中进行最后调整与完善。现在一般采用的是**临界比例法**。

6.2 无人车控制入门

6.2.2 PID控制——PID控制

临界比例法进行PID控制器参数的**整定步骤**如下：

(1)首先预选择一个足够短的采样周期让系统工作；

(2)仅加入比例控制环节，直到系统对输入的**阶跃响应**出现**临界振荡**，记下这时的比例放大系数和临界振荡周期；

(3)在一定的控制度下通过公式计算得到PID控制器的参数。

PID调试一般原则

- 在输出不振荡时，增大比例增益。
- 在输出不振荡时，减小积分时间常数。
- 在输出不振荡时，增大微分时间常数。

6.2 无人车控制入门

6.2.2 PID控制——PID控制

PID调试的一般步骤：

- (1)确定比例增益
- (2)确定积分时间常数
- (3)确定微分时间常数
- (4)系统空载、带载联调，再对PID参数进行微调，直至满足要求。



先P后I最后D

PID三个环节的总结

比例环节：

优点：调整系统的开环比例系数，提高系统的稳态精度，减低系统的惰性，加快响应速度。

缺点：仅用P控制器，过大的开环比例系数不仅会使系统的超调量增大，而且会使系统稳定裕度变小，甚至不稳定。

6.2 无人车控制入门

6.2.2 PID控制——PID控制

PID三个环节的总结

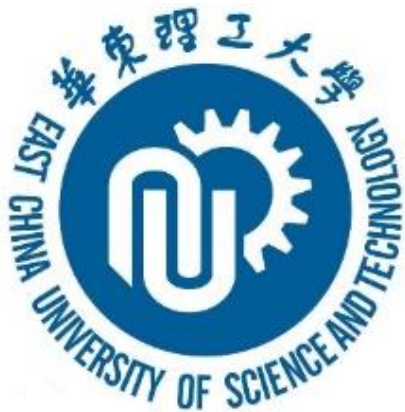
积分环节：

优点：消除稳态误差。

缺点：积分控制器的加入会影响系统的稳定性，使系统的稳定裕度减小。

微分环节：

优点：使系统的响应速度变快，超调减小，振荡减轻，对动态过程有“预测”作用。



目录

CONTENT

- 1 运动学自行车模型和动力学自行车模型
- 2 无人车控制入门
- 3 **基于运动学模型的模型预测控制**
- 4 轨迹追踪
- 5 车辆运动学和动力学建模

6.3 基于运动学模型的模型预测控制

传统的PID控制：易于实现，被广泛应用，具有**延迟性**，即控制指令会在“**未来的车辆状态下**”被执行，在实际无人车的横向控制（转向控制）中并不常使用。

模型预测控制（Model Predictive Control, MPC）：一类追求短时间间隔内最优控制理论，可以将**控制延迟**考虑到车辆模型中，从而避免PID控制器的延迟问题。

横向控制

6.3 基于运动学模型的模型预测控制

6.3.1 将PID控制应用于转向控制存在的问题

模型预测控制（MPC）是一种**将更长时间跨度，甚至于无穷时间的最优化控制问题分解为若干个更短时间跨度，或者有限时间跨度的最优化控制问题**，并且在一定程度上仍然追求最优解的理论。模型预测控制由如下三个因素组成：

- **预测模型**：预测模型能够在短时间内很好地预测系统状态的变化。
- **在线滚动优化**：通过某种最优化算法来优化未来一段时间的控制输入，使得在这种控制输入下预测模型的输出与参考值的差距最小。
- **反馈校正**：到下一个时间点根据新的状态重新进行预测和优化。

6.3 基于运动学模型的模型预测控制

6.3.1 将PID控制应用于转向控制存在的问题

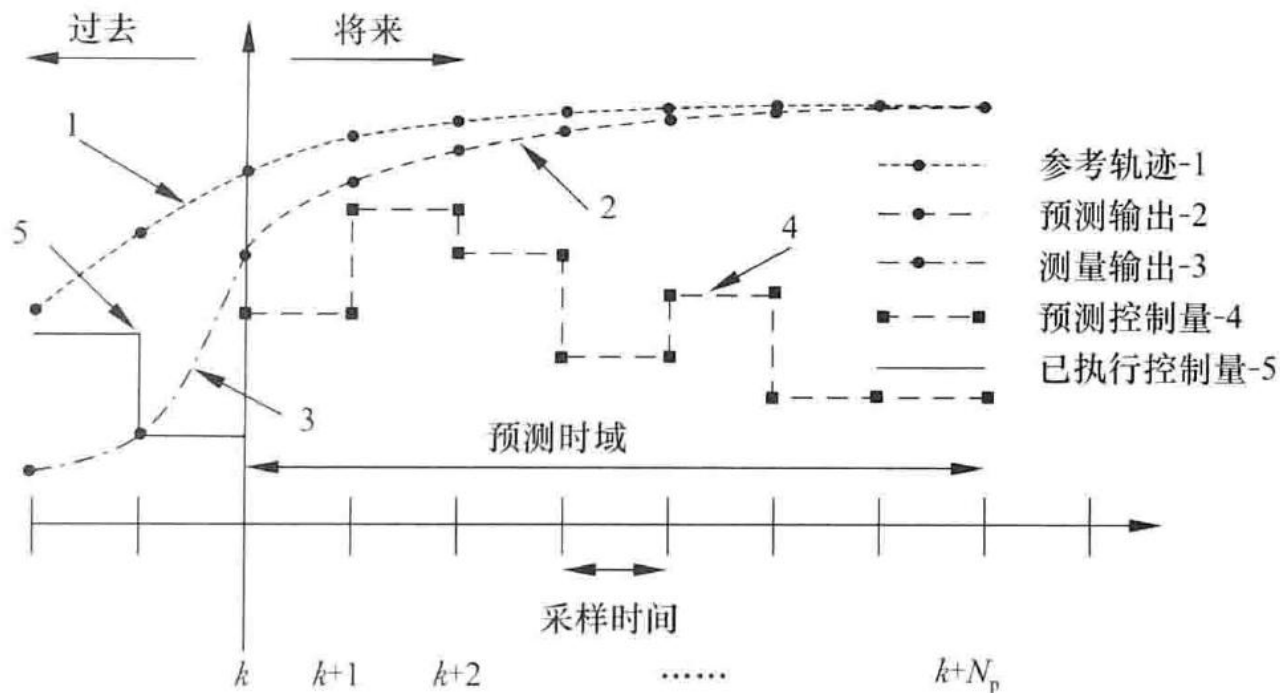


图3.1 模型预测控制原理示意图

6.3 基于运动学模型的模型预测控制

6.3.1 将PID控制应用于转向控制存在的问题

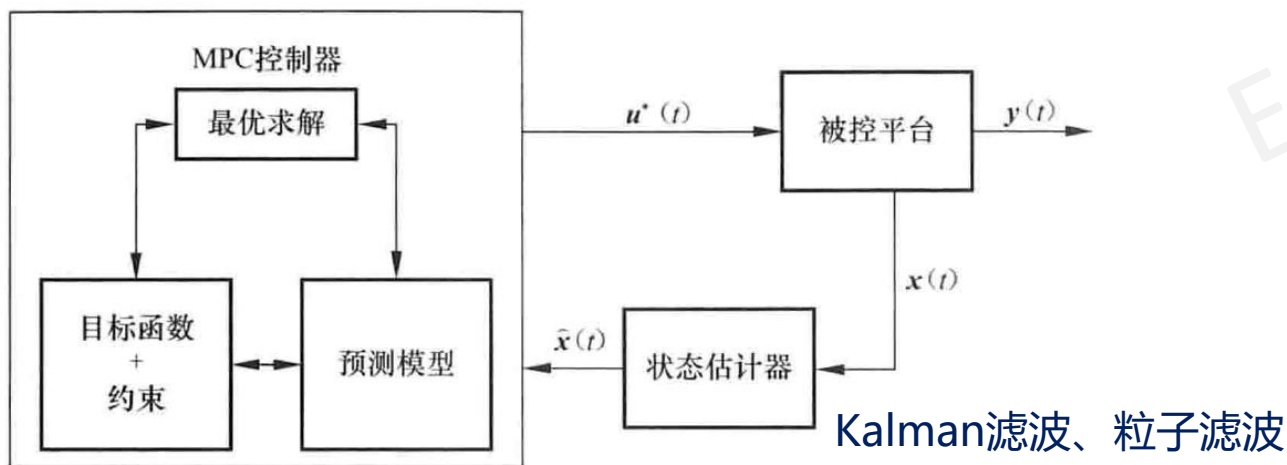


图3.2 MPC控制原理框图

根据所采用的**模型的不同**，MPC可分为

- 动态矩阵控制(Dynamic Matrix Control, DMC)
- 模型算法控制(Model Algorithm Control, MAC)
- 广义预测控制(Generalized Predictive Control, GPC)

6.3 基于运动学模型的模型预测控制

6.3.2 预测模型

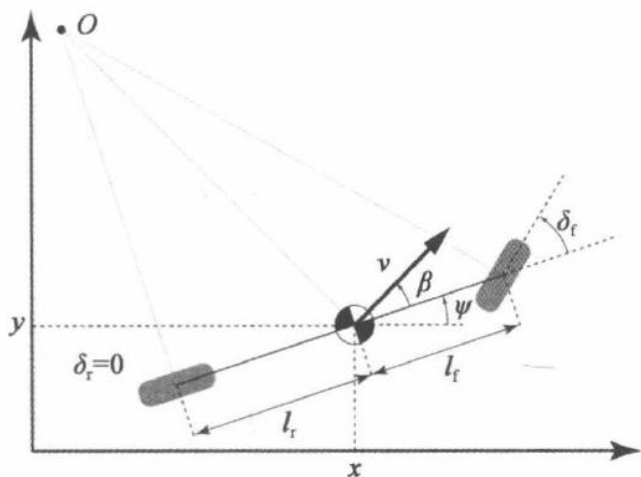


图3.3 一个简单的运动学自行车模型

各个状态量的更新方式：

$$x_{t+1} = x_t + v_t \cos(\psi_t + \beta) \times dt$$

$$y_{t+1} = y_t + v_t \sin(\psi_t + \beta) \times dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{l_r} \sin(\beta) \times dt$$

$$v_{t+1} = v_t + a \times dt$$

其中， β 可以由如下公式计算：

$$\beta = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta_f) \right)$$

基于以上公式，在给定一个控制指令的情况下，这个预测模型能够根据运动学的规律计算出 dt 时间以后车辆的状态 (x, y, ψ, v) 。由于自行车模型本身建立在一定的假设前提下的，所以计算出来的状态只是**理论上车辆的可能状态**。

6.3 基于运动学模型的模型预测控制

6.3.2 预测模型

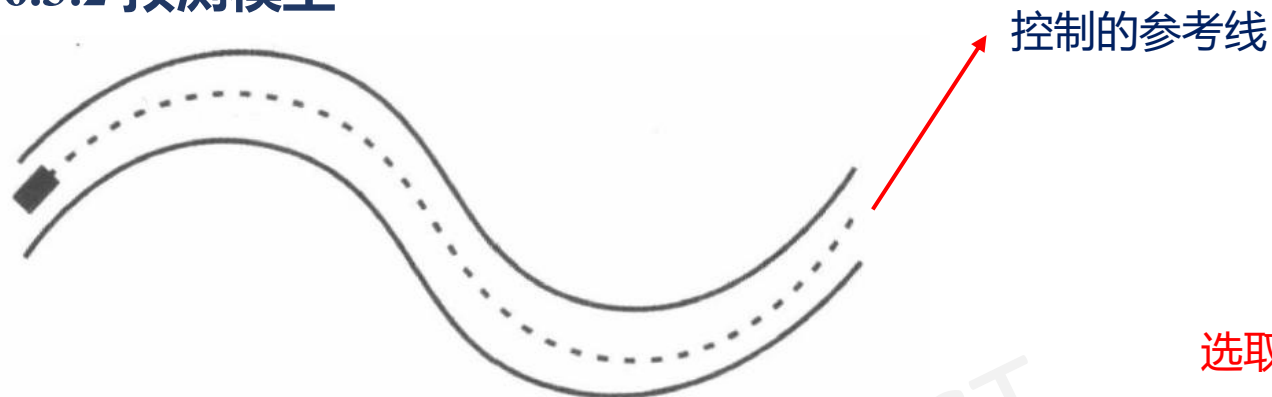


图3.4 带虚线的S形路

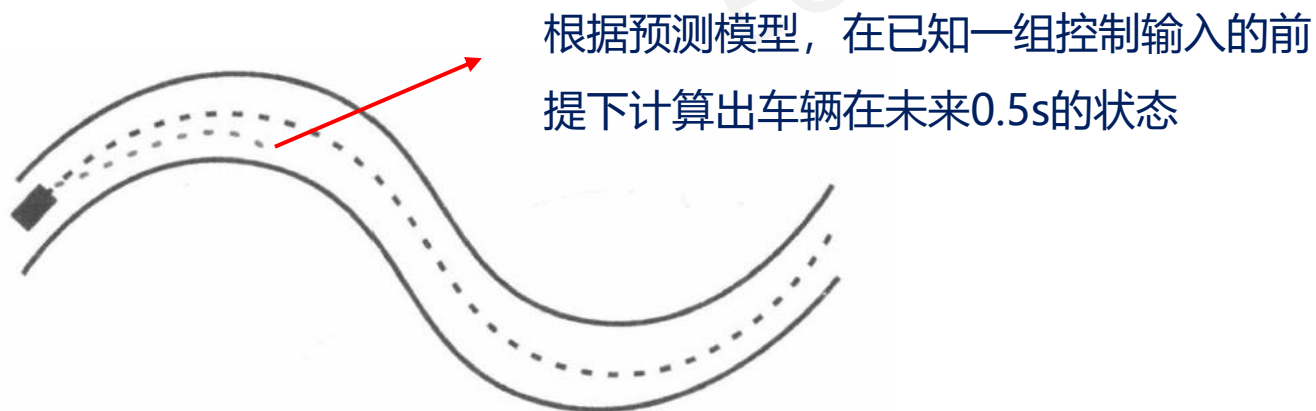


图3.5 车辆在未来0.5s的状态

6.3 基于运动学模型的模型预测控制

6.3.2 在线滚动优化

损失函数就是模型预测的轨迹和参考线之间的CTE，优化的目标就是找出一组控制量，即 (a, δ_f) 油门刹车系数和方向盘转角，使得损失函数最小化。

$$\text{Loss} = \text{CTE} = \sum_{i=1}^{10} (z_i - z_{\text{ref},i})^2$$

只考虑未来10个时间
间隔内的模型预测

其中， $z_i - z_{\text{ref},i}$ 是预测点到实际参考线的距离。

为了完善预测控制，可以在损失函数中添加一项**速度的平方差**，用以控制车辆在这个短时间范围内在每个点上的速度：

$$\text{Loss} = \text{CTE} = \sum_{i=1}^{10} [(z_i - z_{\text{ref},i})^2 + (v_i - v_{\text{ref},i})^2]$$

6.3 基于运动学模型的模型预测控制

6.3.2 在线滚动优化

进一步，如果希望油门系数不突变（即缓慢地踩油门和刹车），就可以将**前后两个油门系数的差值的平方**作为一项加入损失函数在

$$\text{Loss} = \text{CTE} = \sum_{i=1}^{10} [(z_i - z_{\text{ref},i})^2 + (v_i - v_{\text{ref},i})^2 + (a_{i+1} - a_i)^2]$$

变量带约束，如：

- 车辆的前轮转角 δ_f 的取值范围
- 车辆油门系数 a 的取值范围（-1到1之间，1意味着满油门，-1意味着满刹车）

6.3 基于运动学模型的模型预测控制

6.3.2 在线滚动优化

PID控制中遇到的**制动延迟问题**可以通过**添加约束**来解决。

假设车辆的制动延迟是100ms，本文中模型采用10步预测，每步间隔为50ms，模型预测的前两步实际上在制动延迟的时间内，即这两步中车辆仍然在执行上一个状态的制动指令，为了让模型更加贴近实际情况，可约束这两步的制动指令，即 $(a, \Delta a_f)$ **为上一状态的指令**，这样模型预测控制就将制动延迟考虑了。

6.3 基于运动学模型的模型预测控制

6.3.3 反馈校正

本质：反馈控制

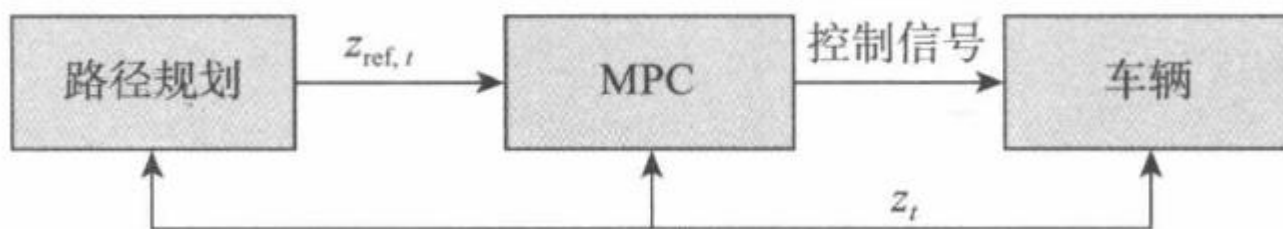


图3.6 MPC的简化示意图

当通过最优化方法得到一组控制输出以后，车辆执行控制指令并且继续以一定的频率反馈当前车辆的状态 z_t 。这个状态会被同时输入到路径规划模块以及MPC模块。路径规划模块会依据新的车辆状态，结合感知模块的信息以及地图信息重新作出规划，MPC模块则根据新的参考路径和车辆当前状态进行新一轮的预测控制。

6.3 基于运动学模型的模型预测控制

6.3.3 反馈校正

注意的是，车辆真实状态的反馈并不是在一个预测时间段的控制执行完以后才进行的，**反馈的时间间隔往往小于一个预测时间段**（在本例中，预测时间段长度为 $0.05 \times 10 = 0.5s$ ）

模型预测控制往往可以分解成如下几步：

- (1) 从 t 时刻开始，预测未来 n 步的输出信号。
- (2) 基于模型的控制信号以及相应的输出信号，构造损失函数，并且通过调整控制信号最优化损失函数。
- (3) 将控制信号输入系统。
- (4) 等到下一个时间点，在新的状态重复步骤1。

6.3 基于运动学模型的模型预测控制

6.3.3 反馈校正

模型预测控制有其天然的多模型约束处理优势，能够与规划控制、感知过程的传感器数据预处理算法很好地结合，是在无人驾驶车辆控制过程中体现车辆运动学与动力学约束的理想方法。

然而，在实际的开发过程中，车辆模型的复杂性受到车辆自由度的影响，当涉及大量自由度时，车辆模型会变得非常复杂。

近年来，**基于神经网络和深度学习的方法**被应用于控制领域，在具有高维数据的控制系统中，引入深度学习具有一定的意义，深度学习自动学习状态特征的能力使得其在无人驾驶系统的研究中具有先天的优势。



目录

CONTENT

- 1 运动学自行车模型和动力学自行车模型
- 2 无人车控制入门
- 3 基于运动学模型的模型预测控制
- 4 **轨迹追踪**
- 5 车辆运动学和动力学建模

6.4 轨迹追踪

对于无人车来说，规划好的路径通常由一系列路径点构成，这些路径点通常包含空间位置信息、姿态信息、速度和加速度等，这些路径点被称为**全局路径点** (Global Waypoint) 。

路径 (Path) 和轨迹 (Trajectory) 的区别在于，轨迹还包含了时间信息，轨迹点也是一种路径点，它在路径点的基础上加入了时间约束，通常将这些轨迹点称为**局部路径点** (Local Waypoint) 。

无人车追踪轨迹的两类方法：

- **基于几何追踪的方法**
- **基于模型预测的方法**



纯追踪

6.4 轨迹追踪

轨迹追踪控制的模型约束：

- **非完整动力学约束轨迹追踪。** 非完整（Nonholonomic）动力学约束就是假设车辆在追踪过程中没有滑移。如**纯追踪算法**。
- **摩擦力与滑移率非线性约束轨迹追踪。** 带有纵横向耦合滑移的非线性轮胎模型在轨迹追踪控制器中一般利用数值优化计算方法进行解算。
- **车辆动力学模型简化对模型预测控制的影响。** 使用复杂的模型能够更好地对车辆输出进行预测，但同时增大了控制器的计算量，实时性难以保证；而使用简单的模型有可能导致追踪失败。
 - 微分平坦方法在非线性运动控制模型简化众多的应用。
 - 侧倾约束对轨迹追踪的影响。
 - 模型预测控制算法实时性。

6.4 轨迹追踪

自行车模型实际上是对**阿克曼转向几何**的一个简化，自行车模型将4轮车辆简化为2轮模型，并且假定车辆只在平面上行驶。

采用自行车模型的一大好处在于它简化了前轮转向角与后轴将遵循的曲率之间的几何关系，其关系如下所示：

$$\tan(\delta) = \frac{L}{R}$$

其中， δ 表示前轮的转向角， L 为轴距， R 为在给定的转向角下后轴遵循着的圆的半径。

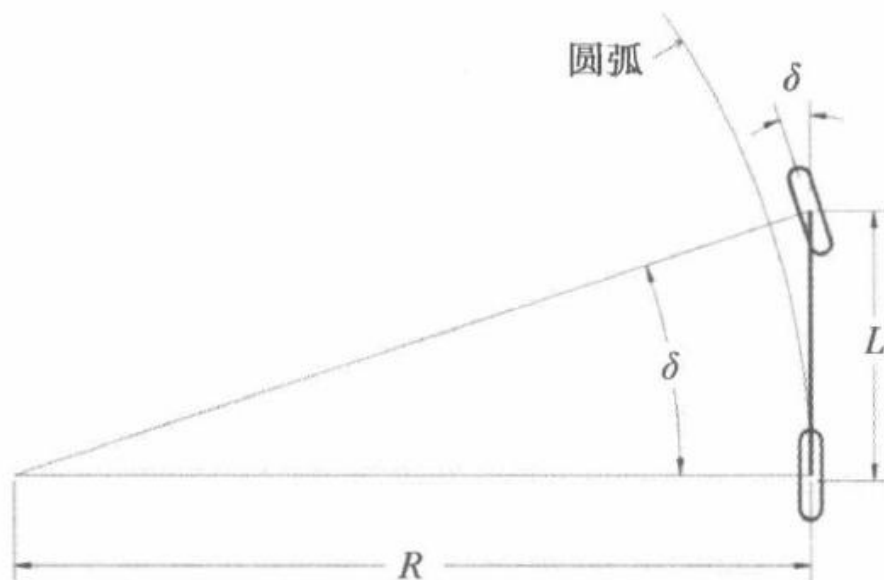


图4.1 自行车模型

6.4 轨迹追踪

从自行车模型出发，纯追踪算法以车后轴为切点、车辆纵向车身为切线，通过控制前轮转向角 δ ，使车辆沿着一条经过目标路径点的圆弧行驶。 (g_x, g_y) 是下一个要追踪的路径点，它位于已经规划好的全局路径中，现在需要控制车辆使得车辆的后轴经过该路径点， l_d 表示车辆当前位置（即后轴位置）到目标路径点 (g_x, g_y) 的距离， α 表示目前车身姿态和目标路径点的夹角，根据正弦定理可以推导出：

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)}$$

$$\frac{l_d}{2\sin\alpha\cos\alpha} = \frac{R}{\cos\alpha}$$

$$\frac{l_d}{\sin\alpha} = 2R$$

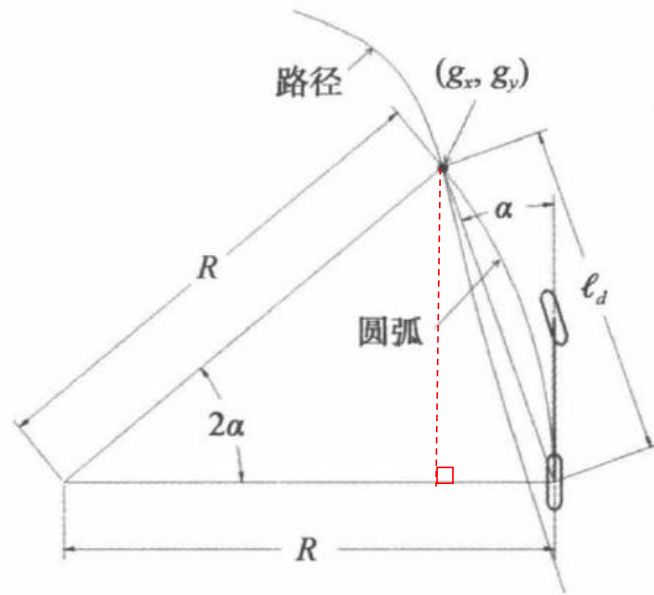


图4.2 纯追踪几何学

6.4 轨迹追踪

上式也可以表示为：

$$\kappa = \frac{2\sin \alpha}{l_d}$$

其中， κ 是计算出来的**圆弧的曲率**，那么

前轮的转向角 δ 的表达式为：

$$\delta = \tan^{-1}(\kappa L)$$

$$\tan(\delta) = \frac{L}{R}$$

结合以上两式，可以得出**纯追踪算法控制**
量 δ 的最终表达式：

$$\delta(t) = \tan^{-1}\left(\frac{2L\sin(\alpha(t))}{l_d}\right)$$

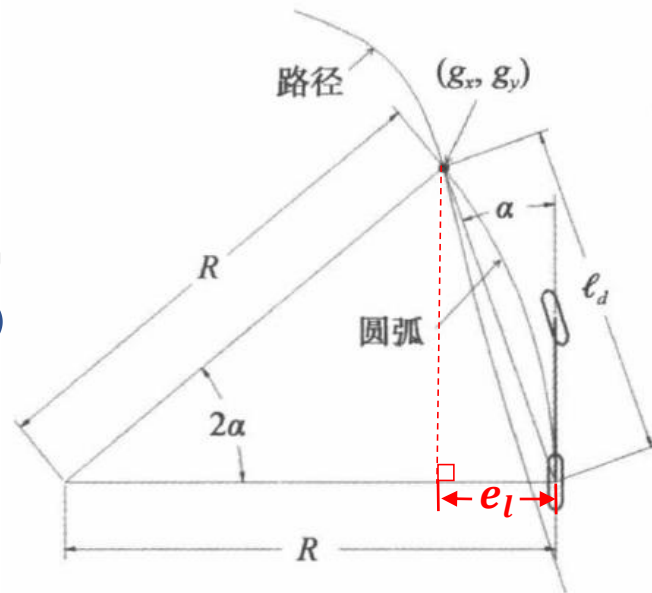


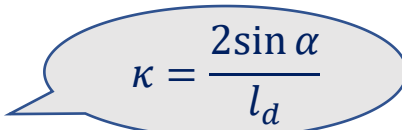
图4.2 纯追踪几何学

6.4 轨迹追踪

考虑时间 t ，在知道 t 时刻车身和目标路径点的夹角 $\alpha(t)$ 以及距离目标路径点的前视距离 l_d 的情况下，由于车辆轴距固定，可以利用上式估计出应该做的前轮转角 $\delta(t)$ ，为了更好地理解纯追踪控制器地原理，定义一个新的量，即车辆当前姿态和目标路径点在**横向上的误差** e_l ，由此可得夹角 α 正弦：

$$\sin(\alpha) = \frac{e_l}{l_d}$$

圆弧的弧度 κ 可重写为：

$$\kappa = \frac{2}{l_d^2} e_l$$


考虑到 e_l 本质是横向上的CTE，由上式可知纯追踪控制器其实是一个横向转角的P控制器，其P系数为 $\frac{2}{l_d^2}$ ，这个**P控制器受参数 l_d （即前视距离）的影响很大**。如何调整前视距离成为纯追踪算法的关键，通常来说，其被认为是车速的函数。

6.4 轨迹追踪

一种最常见的**调整前视距离的方法**就是将前视距离表示成车辆纵向速度的线性函数，即 $l_d = kv_x$ ，那么前轮的转角公式就变为：

$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha(t))}{kv_x(t)} \right)$$
$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha(t))}{l_d} \right)$$

于是，纯追踪器的调整就变成了调整系数 k 。通常来说，会使用**最大、最小前视距离**来约束前视距离，越大的前视距离意味着轨迹的追踪越平滑，越小的前视距离会使得追踪更加精确会带来控制的震荡。

6.4 轨迹追踪

使用Python实现一个简单的纯追踪器

```
import numpy as np
import math
import matplotlib.pyplot as plt

k = 0.1 # 前视距离系数
Lfc = 2.0 # 前视距离
Kp = 1.0 # 速度P控制器系数
dt = 0.1 # 时间间隔, 单位: s
L = 2.9 # 车辆轴距, 单位: m
```

6.4 轨迹追踪

```
class VehicleState:
```

定义车辆状态类

```
    def __init__(self, x=0.0, y=0.0, yaw=0.0, v=0.0):
```

```
        self.x = x
```

```
        self.y = y
```

```
        self.yaw = yaw
```

```
        self.v = v
```

```
def update(state, a, delta):
```

```
    state.x = state.x + state.v * math.cos(state.yaw) * dt
```

```
    state.y = state.y + state.v * math.sin(state.yaw) * dt
```

```
    state.yaw = state.yaw + state.v / L * math.tan(delta) * dt
```

```
    state.v = state.v + a * dt
```

```
    return state
```

模拟车辆的状态更新过程

在简单的自行车模型中，只考虑车辆的当前位置 (x, y) 、车辆的偏航角 yaw 以及车辆的速度 v 。

6.4 轨迹追踪

```
def PControl(target, current):  
    a = Kp * (target - current)  
  
    return a
```

```
def pure_pursuit_control(state, cx, cy, pind):  
  
    ind = calc_target_index(state, cx, cy)  
  
    if pind >= ind:  
        ind = pind  
  
    if ind < len(cx):  
        tx = cx[ind]  
        ty = cy[ind]  
    else:  
        tx = cx[-1]  
        ty = cy[-1]  
        ind = len(cx) - 1
```

```
    alpha = math.atan2(ty - state.y, tx - state.x) - state.yaw
```

计算目前车身姿态和目标路径点的夹角

```
    if state.v < 0: # back  
        alpha = math.pi - alpha
```

```
    Lf = k * state.v + Lfc
```

```
    delta = math.atan2(2.0 * L * math.sin(alpha) / Lf, 1.0)
```

$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha(t))}{kv_x(t)} \right)$$

纵向控制：P控制

横向控制（转角控制）：纯追踪控制

6.4 轨迹追踪

定义函数用于搜索最临近的路径点：

```
def calc_target_index(state, cx, cy):  
    # 搜索最临近的路点  
    dx = [state.x - icx for icx in cx]  
    dy = [state.y - icy for icy in cy]  
    d = [abs(math.sqrt(idx ** 2 + idy ** 2)) for (idx, idy) in zip(dx, dy)]  
    ind = d.index(min(d))  
    L = 0.0  
  
    Lf = k * state.v + Lfc  
  
    while Lf > L and (ind + 1) < len(cx):  
        dx = cx[ind + 1] - cx[ind]  
        dy = cy[ind + 1] - cy[ind]  
        L += math.sqrt(dx ** 2 + dy ** 2)  
        ind += 1  
  
    return ind
```

6.4 轨迹追踪

主函数：

```
def main():
    # 设置目标路线
    cx = np.arange(0, 50, 1)
    cy = [math.sin(ix / 5.0) * ix / 2.0 for ix in cx]

    target_speed = 10.0 / 3.6 # [m/s]

    T = 100.0 # 最大模拟时间

    # 设置车辆的出事状态
    state = VehicleState(x=-0.0, y=-3.0, yaw=0.0, v=0.0)

    lastIndex = len(cx) - 1
    time = 0.0
    x = [state.x]
    y = [state.y]
    yaw = [state.yaw]
    v = [state.v]
    t = [0.0]
    target_ind = calc_target_index(state, cx, cy)

    while T >= time and lastIndex > target_ind:
        ai = PControl(target_speed, state.v)
        di, target_ind = pure_pursuit_control(state, cx, cy, target_ind)
        state = update(state, ai, di)

        time = time + dt

        x.append(state.x)
        y.append(state.y)
        yaw.append(state.yaw)
        v.append(state.v)
        t.append(time)

    plt.cla()
    plt.plot(cx, cy, ".r", label="course")
    plt.plot(x, y, "-b", label="trajectory")
    plt.plot(cx[target_ind], cy[target_ind], "go", label="target")
    plt.axis("equal")
    plt.grid(True)
    plt.title("Speed[km/h]: " + str(state.v * 3.6)[:4])
    plt.pause(0.001)

if __name__ == '__main__':
    main()
```

实现两种控制

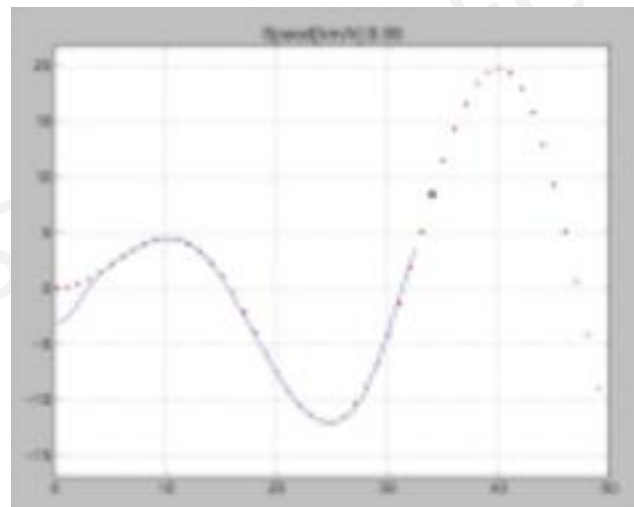


图4.3 纯追踪控制器效果

6.4 轨迹追踪

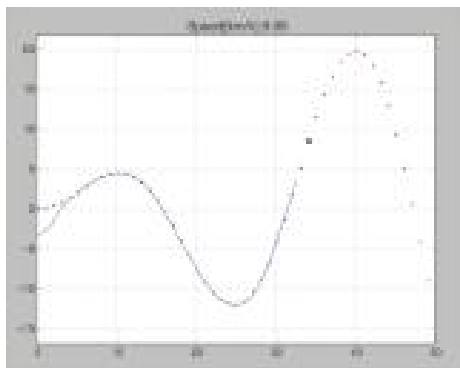


图4.3 纯追踪控制器效果

红点表示实现规划好的路点，蓝线则表示车辆实际运行的轨迹，前面的绿点表示当前前视距离，在这段代码中，设置了最小前视距离为2m，还可以进一步去实验，比如说将前视距离设置得更大些，那么纯追踪控制器就会表现得更加“平滑”，更加平滑的结果就是在某些急剧的转角处会存在转向不足的情况。

6.4 轨迹追踪

以纯追踪控制器为代表的几何路径跟踪器很容易理解和实现，达到了基本的路径跟踪性能，但当**存在显著的速度改变**时会遇到瓶颈，**纯追踪方法使用前视距离考虑路径信息**，这种方法在低速情况下几乎不受路径形状的影响。但是，**选择最佳的前视距离的方法并不明确**。将前视距离表示为速度的函数是一种常见的方法，但是，前视距离也可能是路径曲率的函数，甚至可能和纵向速度以外的CTE有关。所以纯追踪控制器的前视距离调整应该额外注意，**很短的前视距离会造成车辆控制的不稳定甚至震荡，为了确保车辆稳定而设置较长的前视距离又会出现车辆在大转角处转向不足的问题。**



目录

CONTENT

- 1 运动学自行车模型和动力学自行车模型
- 2 无人车控制入门
- 3 基于运动学模型的模型预测控制
- 4 轨迹追踪
- 5 **车辆运动学和动力学建模**

6.5 车辆运动学和动力学建模

6.5.1 车辆运动学建模

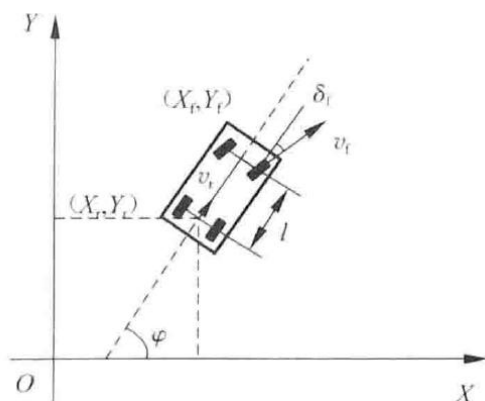


图5.1 车辆运动模型

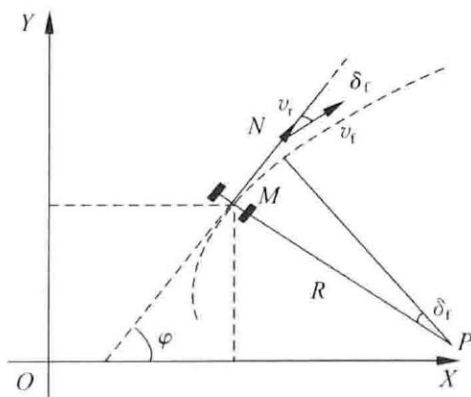


图5.2 车辆前轮转向示意图

(X_r, Y_r) 和 (X_f, Y_f) 分别表示车辆后轴和前轴轴心的坐标

φ : 车体的横摆角 (航向角)

δ_f : 前轮偏角

v_r : 车辆后轴中心速度

v_f : 车辆前轴中心速度

l : 轴距

R : 后轮转向半径

P : 车辆的瞬时转动中心

M : 车辆后轴轴心

N : 前轴轴心

此处, 假设转向过程中车辆质心侧偏角保持不变

6.5 车辆运动学和动力学建模

6.5.1 车辆运动学建模

在后轴行驶轴心(X_r, Y_r)处, 速度为:

$$v_r = \dot{X}_r \cos \varphi + \dot{Y}_r \sin \varphi \quad (1)$$

前后轴的运动学约束为:

$$\begin{cases} \dot{X}_f \sin(\varphi + \delta_f) - \dot{Y}_f \cos(\varphi + \delta_f) = 0 \\ \dot{X}_r \cos \varphi - \dot{Y}_r \sin \varphi = 0 \end{cases} \quad (2)$$

上两个式子联合, 可得:

$$\begin{cases} \dot{X}_r = v_r \cos \varphi \\ \dot{Y}_r = v_r \sin \varphi \end{cases} \quad (3)$$

根据前后轮的几何关系:

$$\begin{cases} X_f = X_r + l \cos \varphi \\ Y_f = Y_r + l \sin \varphi \end{cases} \quad (4)$$

6.5 车辆运动学和动力学建模

6.5.1 车辆运动学建模

将(3)和(4)代入(2)，可得横摆角速度为：

$$\omega = \frac{v_r}{l} \tan \delta_f \quad (5)$$

其中， ω 为车辆横摆角速度。同时，根据 ω 和车速 v_r 可得转向半径 R 和前轮偏角 δ_f ：

$$\begin{cases} R = v_r / \omega \\ \delta_f = \arctan (l / R) \end{cases} \quad (6)$$

由(3)和(5)可得车辆运动学模型为：

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ \tan \delta_f / l \end{bmatrix} v_r \quad (7)$$

该模型可以进一步表示为更一般的形式：

$$\dot{\xi}_{\text{kin}} = f_{\text{kin}}(\xi_{\text{kin}}, \mathbf{u}_{\text{kin}}) \quad (8)$$

其中，状态量 $\xi_{\text{kin}} = [X_r, Y_r, \varphi]^T$ ，控制量 $\mathbf{u}_{\text{kin}} = [v_r, \delta_f]^T$ 。

6.5 车辆运动学和动力学建模

6.5.1 车辆运动学建模

在无人车驾驶车辆的路径追踪控制过程中，往往希望以 $[v_r, \omega]$ 作为控制量，将(5)代入(7)中，该车辆运动学模型可以被转换为如下的形式：

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ 0 \end{bmatrix} v_r + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (9)$$

可以通过Matlab/Simulink环境进行运动学模型验证。

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模

在进行车辆动力学建模时，进行以下理想化的假设：

1. 假设无人驾驶车辆在平坦路面上行驶，忽略车辆垂向运动。
2. 悬架系统以及车辆是刚性的，忽略悬架运动及其对耦合关系的影响。
3. 只考虑纯侧偏轮胎特性，忽略轮胎里的纵横向耦合关系的影响。
4. 用单轨模型来描述车辆运动，不考虑载荷的左右转移。
5. 假设车辆行驶速度变化缓慢，忽略前后轴的载荷转移。
6. 忽略纵向和横向空气动力学

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——车辆单轨模型

基于以上6点理想假设，平面运动车辆只具有3个方向的运动，即纵向、横向和横摆运动。

设定车辆为前轮运动。

轮胎受力的定义如下：

F_{lf}, F_{lr} ：前后轮胎受到的纵向力

F_{cf}, F_{cr} ：前后轮胎受到的侧向力

F_{xf}, F_{xr} ：前后轮胎受到的x方向的力

F_{yf}, F_{yr} ：前后轮胎受到的y方向的力

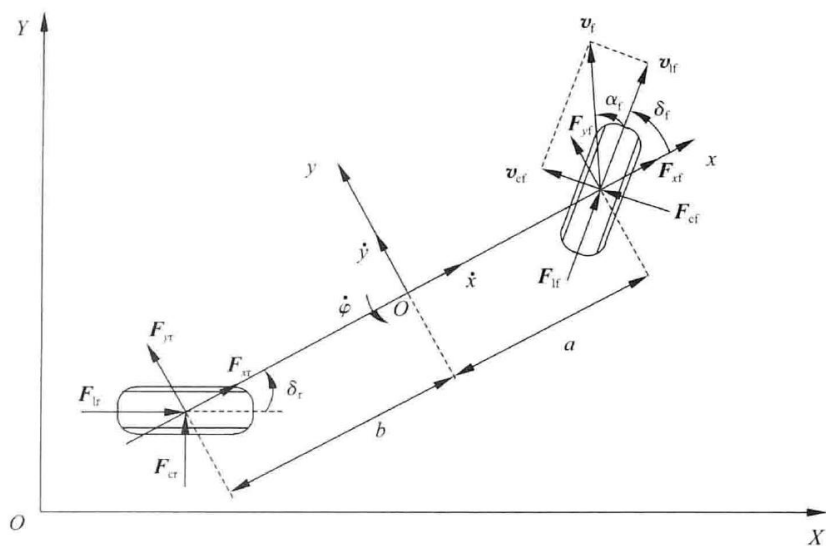


图5.3 车辆单轨模型

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——车辆单轨模型

根据牛顿第二定律，分别的沿 x 轴、 y 轴和 z 轴的受力平衡方程

$$m\ddot{x} = m\dot{y}\dot{\phi} + 2F_{xf} + F_{xr} \quad (10)$$

$$m\ddot{y} = -m\dot{x}\dot{\phi} + 2F_{yf} + F_{yr} \quad (11)$$

$$I_z\ddot{\phi} = 2aF_{yf} - 2bF_{yr} \quad (12)$$

式子中， a 、 b 分别为质心到前后轴的距离， m 为车辆整备质量， I_z 表示车辆绕 z 轴的转动惯量。

轮胎在 x 方向和 y 方向上受到的合力与纵、侧向力的转换关系如下：

$$F_{xf} = F_{lf}\cos\delta_f - F_{cf}\sin\delta_f \quad (13)$$

$$F_{xr} = F_{lr}\cos\delta_r - F_{cr}\sin\delta_r \quad (14)$$

$$F_{yf} = F_{lf}\sin\delta_f + F_{cf}\cos\delta_f \quad (15)$$

$$F_{yr} = F_{lr}\sin\delta_r + F_{cr}\cos\delta_r \quad (16)$$

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——车辆单轨模型

轮胎的纵向力、侧向力可以表示为轮胎侧偏角、滑移率、路面摩擦系数和垂向载荷等参数的复杂函数：

$$F_l = f_l(\alpha, s, \mu, F_z) \quad (17)$$

$$F_c = f_c(\alpha, s, \mu, F_z) \quad (18)$$

式中， α 为轮胎侧偏角， s 为滑移率， μ 为路面摩擦系数， F_z 为轮胎所受到的垂向载荷。

轮胎的偏转角 α 可以由几何关系计算得到：

$$\alpha = \tan^{-1} \frac{v_c}{v_l} \quad (19)$$

其中， v_c 和 v_l 为轮胎在侧向、纵向的速度，可以由坐标系方向的速度 v_x 和 v_y 表示， δ 为轮胎偏转角：

$$v_l = v_y \sin \delta + v_x \cos \delta \quad (20)$$

$$v_c = v_y \cos \delta - v_x \sin \delta \quad (21)$$

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——车辆单轨模型

轮胎的速度往往很难获取，一般可以通过车辆速度计算得到。根据速度关系可以推导出如下转换关系

$$v_{yf} = \dot{y} + a\dot{\phi} \quad v_{yr} = \dot{y} - b\dot{\phi} \quad (22)$$

$$v_{xf} = \dot{x} \quad v_{xr} = \dot{x} \quad (23)$$

轮胎在地面上的滑移率 s 可计算：

$$s = \begin{cases} \frac{rv\omega_t}{v} - 1 (v > r\omega_t, v \neq 0) \\ 1 - \frac{v}{r\omega_t} (v < r\omega_t, \omega_t \neq 0) \end{cases} \quad (24)$$

式中， r 为车轮半径， ω_t 为车轮旋转角速度。

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——车辆单轨模型

假设车辆行驶速度变化缓慢，忽略前后轴的载荷转移，可以通过以下算式计算得到车辆前、后轮胎所受到的垂向载荷：

$$F_{zf} = \frac{bmg}{2(a+b)} \quad (25)$$

$$F_{zr} = \frac{amg}{2(a+b)} \quad (26)$$

最后，考虑车身坐标系与惯性坐标系之间的转换关系，可得：

$$\dot{Y} = \dot{x} \sin \varphi + \dot{y} \cos \varphi \quad (27)$$

$$\dot{X} = \dot{x} \cos \varphi - \dot{y} \sin \varphi \quad (28)$$

结合式(10)~(28)，可以得到**车辆非线性动力学模型**。除了路面摩擦系数和滑移率，其他参数都可以由车辆状态信息计算得到。

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——车辆单轨模型

假设被控车辆具备良好的防抱死制动系统，滑移率始终保持在最佳工作点，将系统描述为以下状态空间表达式：

$$\begin{aligned}\dot{\xi}_{\text{dyn}} &= f_{\text{dyn}}(\xi_{\text{dyn}}, \mathbf{u}_{\text{dyn}}) \\ \boldsymbol{\eta}_{\text{dyn}} &= h_{\text{dyn}}(\xi_{\text{dyn}})\end{aligned}\tag{29}$$

在该系统中，状态量选取为 $\xi_{\text{dyn}} = [\dot{y}, \dot{x}, \varphi, \dot{\varphi}, Y, X]^T$ ，控制量选取为 $\mathbf{u}_{\text{dyn}} = \delta_f$ （仅考虑前轮转向车辆， δ_r 视为 0），输出量选取为 $\boldsymbol{\eta}_{\text{dyn}} = [\varphi, Y]^T$ 。

在实际的控制过程中，路面摩擦系数和滑移率视为已知量，该模型即模型预测控制器中预测模型的基础。

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

轮胎模型可以分为下列三种：

理论轮胎模型

经验轮胎模型

物理轮胎模型



Pacejka提出的以**魔术公式**（Magic Formula, MF）为基础的**经验轮胎模型**，运用**三角函数的组合公式**拟合轮胎试验室数据，描述轮胎的纵向力 F_l 、侧向力 F_c 、回正力矩 M_z 、翻转力矩 M_x 、阻力矩 M_y 与侧偏角 α 、滑移率 s 之间的定量关系，以及纵向力、侧向力的联合作用工况。

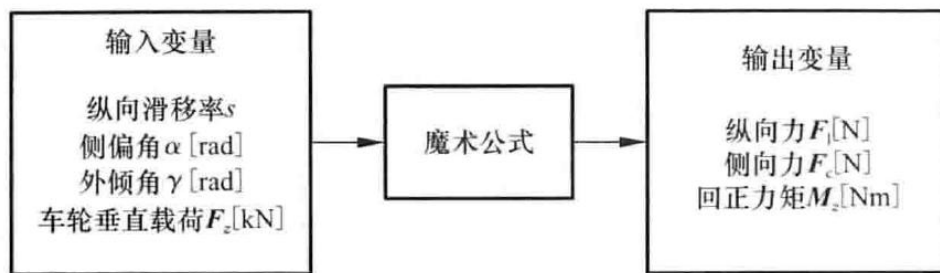
6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

魔术公式的一般表达式为：

$$Y(x) = D \sin \{ C \arctan [Bx - E(Bx - \arctan(BX))] \} \quad (30)$$

其中， B 、 C 、 D 依次由轮胎的垂直载荷和外倾角确定， Y 为输出变量。可以是纵向力或者侧向力或者回正力矩； x 为输入变量，在不同情况下分别表示轮胎的偏转角和纵向滑移率； B 为刚度因子， C 为形状因子， D 为峰值因子， E 为曲率因子。



实际应用中，由于帘布层转向效应、侧偏力作用或滚动阻力会引起偏移，通常还会引入垂直偏移和水平偏移

图5.4 基于魔术公式的轮胎模型的输入和输出变量

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

Pacejka轮胎模型认为，轮胎在垂直、侧面方向上是线性的，阻尼为常数。这在侧向加速度 $a_y \leq 0.4g$ 、轮胎侧偏角 $\alpha \leq 5^\circ$ 的情景下对常规轮胎具有很高的拟合精度。

魔术公式具有较好的健壮性，在极限值以外一定程度内使用，可以对有限工况进行外推且具有较好的置信度。



6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

利用魔术公式计算轮胎的纵向力及侧向力的关系如下：

(1) 轮胎纵向力计算方法

$$F_l = D \sin \left[C \arctan \left(Bx_l - E(Bx_l - \arctan(Bx_l)) \right) \right] + S_v$$

式中， $x = s + S_h$ ， s 为轮胎侧偏角， S_h 为曲线的水平方向漂移， S_v 为曲线的垂直方向漂移。

- C 为曲线的形状因子， $C = B_0$,
- D 为曲线颠因子，表示曲线的最大值， $D = B_1 F_z^2 + B_2 F_z$ ，其中， F_z 为轮胎受到的垂向载荷。
- B 为刚度因子， $B = (B_3 F_z^2 + B_4 F_z) \times e^{-B_5 F_z} / (C \times D)$
- E 为曲线的曲率因子， $E = B_6 F_z^2 + B_7 F_z + B_8$,
- $S_h = B_9 F_z + B_{10}$
- $S_v = 0$

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

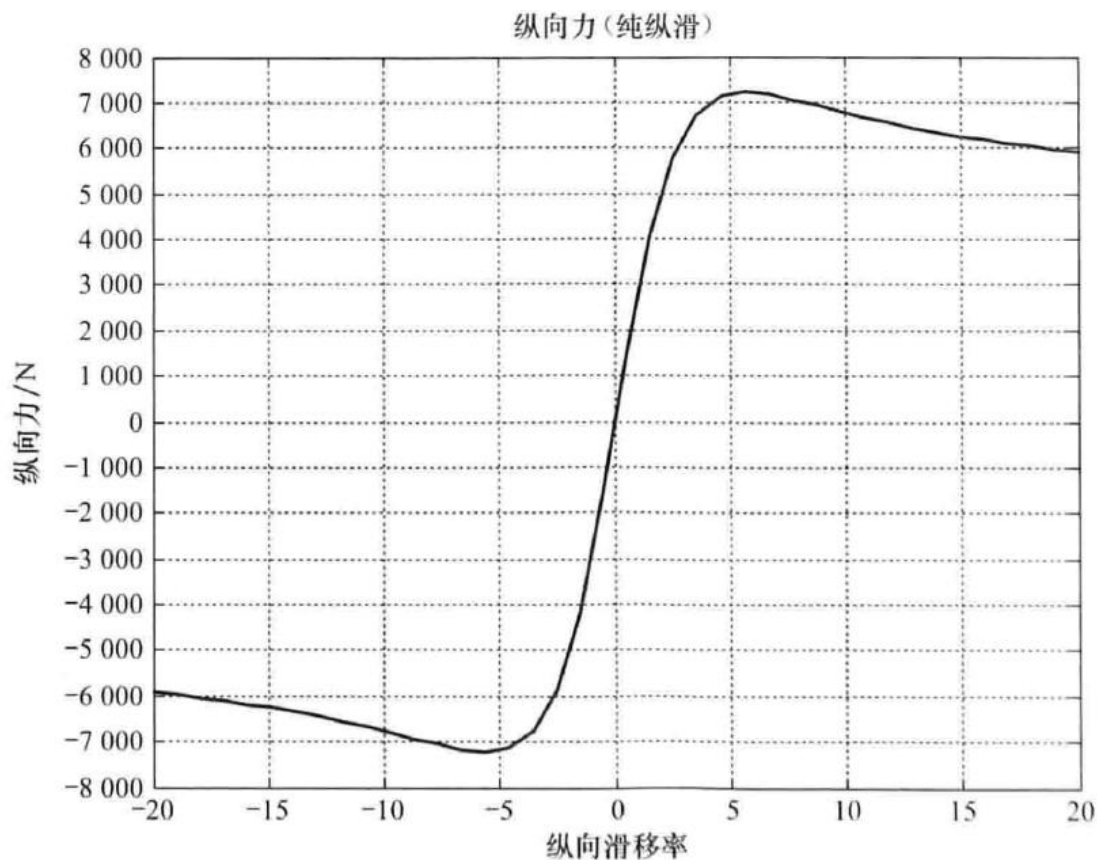


图5.5 Pacejka' 89轮胎纵向力示例

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

利用魔术公式计算轮胎的纵向力及侧向力的关系如下：

(2) 轮胎侧向力计算方法

$$F_c = D \sin [C \arctan (Bx - E(Bx - \arctan (Bx)))] + S_v$$

式中， $x = s + S_h$ ， s 为轮胎侧偏角， S_h 为曲线的水平方向漂移， S_v 为曲线的垂直方向漂移。

- C 为曲线的形状因子， $C = A_0$,
- D 为曲线颠因子，表示曲线的最大值， $D = A_1 F_z^2 + A_2 F_z$,
- B 为刚度因子， $B = A_3 \sin (2 \arctan \frac{F_z}{A_4}) \times (1 - A_5 |\gamma|) / (C \times D)$ ，其中， γ 为轮胎外倾角，
- E 为曲线的曲率因子， $E = A_6 F_z + A_7$,
- $S_h = A_9 F_z + A_{10} + A_8 \gamma$
- $S_v = A_{11} F_z \gamma + A_{12} F_z + A_{13}$

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

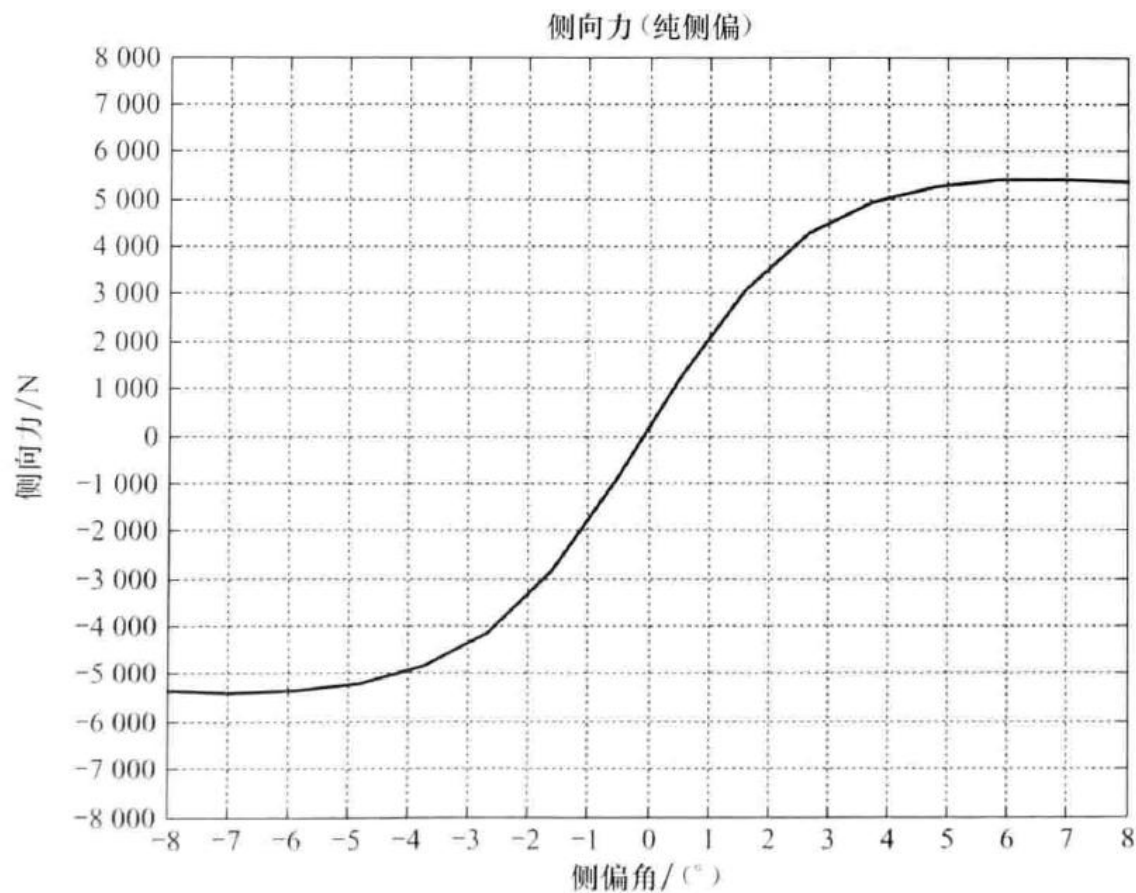


图5.6 Pacejka' 89轮胎侧向力示例

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

(3) 轮胎回正力矩计算方法

$$M_z = D \sin [C \arctan (B x_1 - E (B x_1 - \arctan (B x_1)))] + S_v$$

式中, x_1 为回正力矩计算组合自变量, $x_1 = (\alpha + S_h)$, α 为侧偏角。

- C 为曲线的形状因子, $C = C_0$,
- D 为巅因子, 表示曲线的最大值, $D = C_1 F_z^2 + C_2 F_z$
- BCD 为回正力矩零点处的扭转刚度: $BCD = (C_3 F_z^2 + C_4 F_z) \times (1 - C_6 |\gamma|) \times e^{-C_5 F_z}$
- B 为刚度因子, $B = BCD / (C \times D)$,
- E 为曲线的曲率因子, 表示曲线最大值附近的形状, $E = (C_7 F_z^2 + C_7 F_z + C_9) \times (1 - C_{10} |\gamma|)$
- $S_h = C_{11} \gamma + C_{12} F_z + C_{13}$
- $S_v = \gamma (C_{14} F_z^2 + C_{15} F_z) + C_{16} F_z + C_{17}$

6.5 车辆运动学和动力学建模

6.5.2 车辆动力建模——轮胎模型

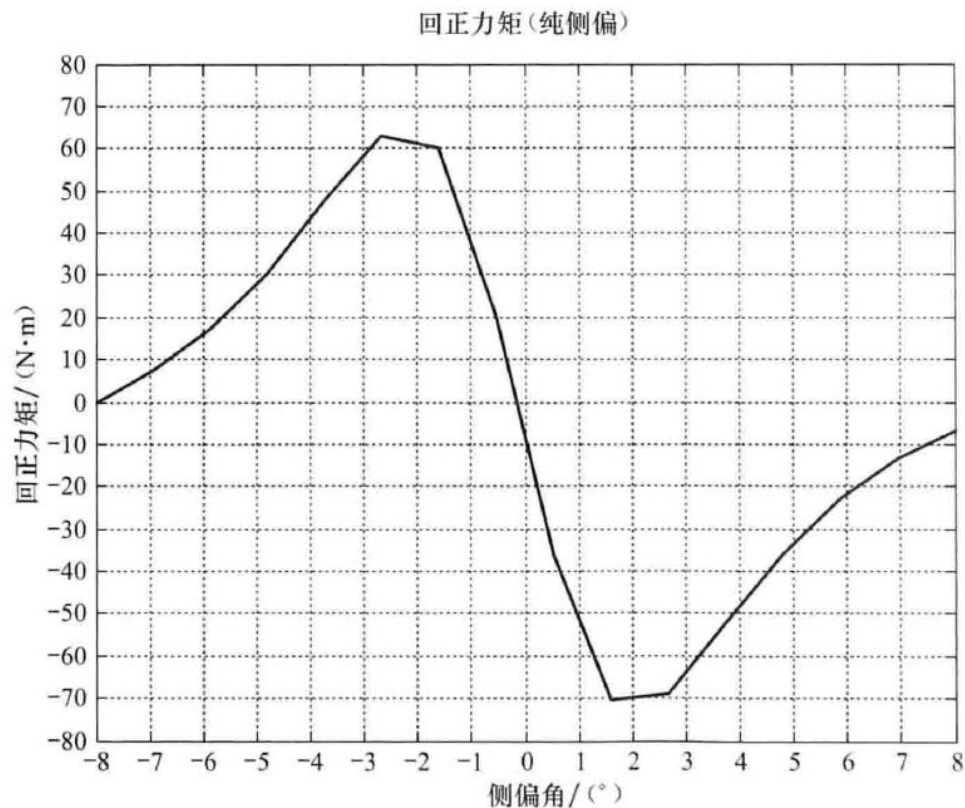
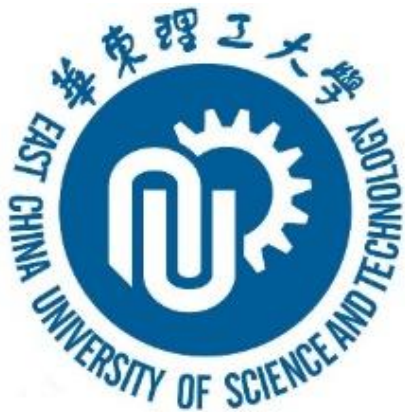


图5.7 Pacejka' 89轮胎回正力矩示例



谢谢！

THANK YOU FOR LISTENING