

智能无人系统

[本页PDF](#)

复习PPT东西这么多是要把我们撑死吗？

第一章和第七章

自动驾驶分级标准

- (1) **Level 0：无自动化。**
- (2) **Level 1：单一功能级自动化。**无法做到手和脚同时不操控。
- (3) **Level 2：部分自动化。**在某些预设环境下可以不操作汽车，即手脚同时离开控制。
- (4) **Level 3：有条件自动化。**特定条件部分任务。
- (4) **Level 4：高度自动化。**特定条件全部任务。
- (5) **Level 5：完全自动化。**全部条件全部任务。

传感器平台

主流以**激光雷达**和**车载摄像头**为主，包括多传感器融合

多传感器融合用了**卡尔曼滤波**

定位技术对比

- GPS+惯导：**开阔首选**，绝对定位能力强
- 激光SLAM：精确但成本高，**怕恶劣天气**，**适合有结构的场景**
- 视觉SLAM：成本低，**怕弱纹理、光照变化**
- 纯惯导：任意环境均可工作但误差累积

传感方案选择

传感方案的选择：

环境场景	主传感器	辅助传感器	为什么这样配
无GPS+有结构	激光雷达	IMU	激光提供绝对位置，IMU填补扫描间隔
有GPS+开阔地	GPS	IMU	GPS提供全局定位，IMU提高更新频率
有GPS+有纹理	视觉+GPS	IMU	视觉提供局部精度，GPS防止漂移
恶劣天气	GPS+IMU	-	激光/视觉受影响，回归最可靠组合

各种传感器工作原理

激光雷达

利用可见和近红外光波**发射、反射和接收**来探测物体

计算平台

GPS

特点：浮点运算、并行计算强

优势：占用数据中心基础设施少，吞吐量高

DSP

特点：采用**哈佛设计**，以数字信号处理大量数据，允许在程序空间和数据空间之间进行传输

优势：运算能力强，速度快，体积小，灵活性高

FPGA

特点：硬件配置灵活、低能耗、高性能及可编程

优势：硬件可升级、可迭代

ASIC

特点：有异构的全编程加速器，对专有算法优化

优势：节省计算时间，提高计算效率

控制平台

包括两大部分：

- 电子控制单元ECU：实现控制算法
- 通信总线：实现ECU以及机械部件间的通信功能
 - A类：低速网络
 - B类：中速网络
 - C类：多路传输高速网络
 - D类：高性能网络

通信网络协议

局部互联协议LIN

低速网络

数据传输方式：单线传输（低成本）

工作机制：主从模式、单主多从

适用场景：

控制器局域网CAN

中速网络

CAN总线采用**差分信号**传输

数据传输方式：单线传输（低成本）

工作机制：仲裁机制、节点并联

适用场景：

高速容错网络协议FlexRay

高速安全可靠网络

数据传输方式：单线传输（低成本）

工作机制：主从模式、单主多从

适用场景：

无人驾驶的安全性

传感器的安全

攻击方法：

- **惯性传感器IMU**：利用强磁场干扰
- **GPS**：设置大功率假GPS信号
- **轮测距技术**：黑客破坏轮子
- **激光雷达**：强反光物、激光照射、地图调包
- **计算机视觉**：在路上放置假的红绿灯和行人

解决方法：利用多传感器融合技术互相纠正

操作系统的安全

攻击方法

- 劫持其中一个ROS节点，无限分配资源导致内存溢出
 - 解决方法：使用**容器技术**限制每一个节点可使用的资源，并使用**沙盒方式**保证节点运行独立
- 劫持ROS的服务或者话题，导致ROS之间传递的信息被伪造
 - 解决方法：在ROS节点之间进行加密通信（影响性能）

控制系统的安全

CAN总线采用**差分信号**传输

攻击方式：

- OBD-II入侵
- 电动车充电器入侵
- 车载CD入侵
- 蓝牙入侵
- TPMS入侵

解决办法：进行加密验证，使用**对称或非对称密码**

车联网通信系统的安全

解决办法：

- 确认消息来源：验证安全证书

- 保证消息未被篡改：计算信息完整性
- PRESERVE三层保护
 - 硬件
 - 软件
 - 安全证书

第二章

自主运动的两大基本问题：

- 我在什么地方：**定位**
- 周围环境是什么样的：**建图**

SLAM：同步定位与地图构建

- 移动过程中根据位置估计和地图进行自身定位，构建**增量式地图**
- SLAM技术包括激光SLAM和视觉SLAM

下面介绍两个算法（ICP NDT），可以根据点云的匹配获取自身的位置，是局部、相对定位

迭代最近点算法（ICP）

算法原理：让数据点云的每个点在参考点云中找一个最近的匹配（欧氏距离），通过这个匹配计算均方误差，通过最小化误差计算刚体变换参数（**平移向量和旋转矩阵**），从而得到无人车的位置坐标

算法思想：启发式思想

求解方式：线性代数SVD，非线性优化方式求解

特点：对点云噪声和缺失敏感，易陷入局部最优，耗时长，计算量大且依赖较好的初值，匹配效果精准，初值好时收敛好

SVD求解步骤

点云的匹配过程就是寻找一个最优的转换矩阵 T ，由一个旋转矩阵 R 和平移向量 t 组成

P 为目标点集， Q 为样本点集

1. 将点集 P 、 Q 中的点减去自身的质心

$$\begin{cases} \mu_P = \frac{1}{n} \sum_{i=1}^n p_i \\ \mu_Q = \frac{1}{n} \sum_{i=1}^n q_i \end{cases}$$

去掉质心后坐标为

$$\begin{cases} P' = \{p_i - \mu_P\} = \{p'_i\} \\ Q' = \{q_i - \mu_Q\} = \{q'_i\} \end{cases}$$

2. 根据以下优化问题计算旋转矩阵

$$R^* = \argmin \frac{1}{2} \sum_{i=1}^n \|p'_i - Rq'_i\|^2$$

求得去中心后两点集的协方差矩阵

$$W = \sum_{i=1}^n q_i'(p_i')^T$$

3. 计算旋转矩阵 R

对协方差矩阵进行SVD分解

$$W = U \textcolor{red}{\Sigma} V^T$$

当 W 满秩的时候有唯一解

$$R = UV^T$$

4. 计算平移矢量

$$t = \mu_Q - R\mu_P$$

①	计算Q中的每一个点在P点集中的对应近点
②	求得使上述对应点对平均距离最小的刚体变换，求得平移参数和旋转参数
③	对Q使用上一步求得的平移和旋转参数，得到新的变换点集
④	如果新的变换点集与参考点集满足两点集的平均距离小于某一给定阈值，则停止迭代计算，否则新的变换点集作为新的Q继续迭代，直到达到目标函数的要求

正态分布变换（NDT）

基于标准正态分布的配准算法，应用了**三维点的统计模型**

特点：因为其在配准过程中不利用对应点的特征计算和匹配，所以计算速度比其他方法快。对噪声和点云缺失具有更强鲁棒性，计算效率高且对初值不敏感

基本思想：

基于GPS+惯性组合导航的定位系统

- GPU更新频率低，在1-20Hz
- INS设备更新频率高，在50-200Hz

几种坐标系

- 地心惯性坐标系（ECI）：不参与地球自转，**惯性**
- 地心地固坐标系（ECEF）：与地球保持同步旋转，**非惯性**
- 导航坐标系：指北导航坐标系，X指东，Y指北，Z沿垂线方向指向天向
- 车体坐标系：与车辆固连的坐标系，原点O位于车辆质心，X轴沿车辆纵轴指向前方，Y轴沿车辆横轴指向左侧，Z轴符合右手笛卡尔坐标系

坐标转换的方法：**三次旋转**

GPS输出的信号：经度、纬度、高度等

INS输出的信号：包含x、y、z方向的加速度，以及对应旋转角速度信息

影响GPS定位准确度的因素: 大气层影响、卫星星历误差、卫星钟差、多路径效应

GPS定位误差

接收机的公有误差	卫星钟误差、星历误差、电离层误差、对流程误差	可以通过差分技术完全予以消除
接收机自身存在的误差	内部噪声、通道延迟、多路径效应	硬件设备的内在误差，无法消除
基准站接收机与流动站接收机之间	传播延迟误差	取决于移动站与基站之间的距离

GNSS

基于GPS信号的导航系统

由三个主要部分组成：

- 空间段：卫星或航天器组成，传输导航电文
- 控制段：地面监控站和主控中心，提供导航信息，进行卫星控制
- 用户段：GNSS接收机

使用了三角定位法，即三个卫星组成一个三角形，另外一个卫星提供对时校准

惯性导航系统（INS）

惯性测量单元（IMU）的检测原理： 惯性测量单元（IMU） = 3轴加速度计 + 3轴陀螺

- 频率高，不易受干扰
- 但用积分确定导航信息，定位误差会累加

惯性传感器： 加速度计

- 使用比力作为单位，输出读数是比力

- 惯导比力方程， a 是相对于惯性空间的运动加速度， g 是万有引力加速度

$$f = a - g$$

惯性导航原理：递推式行位推算

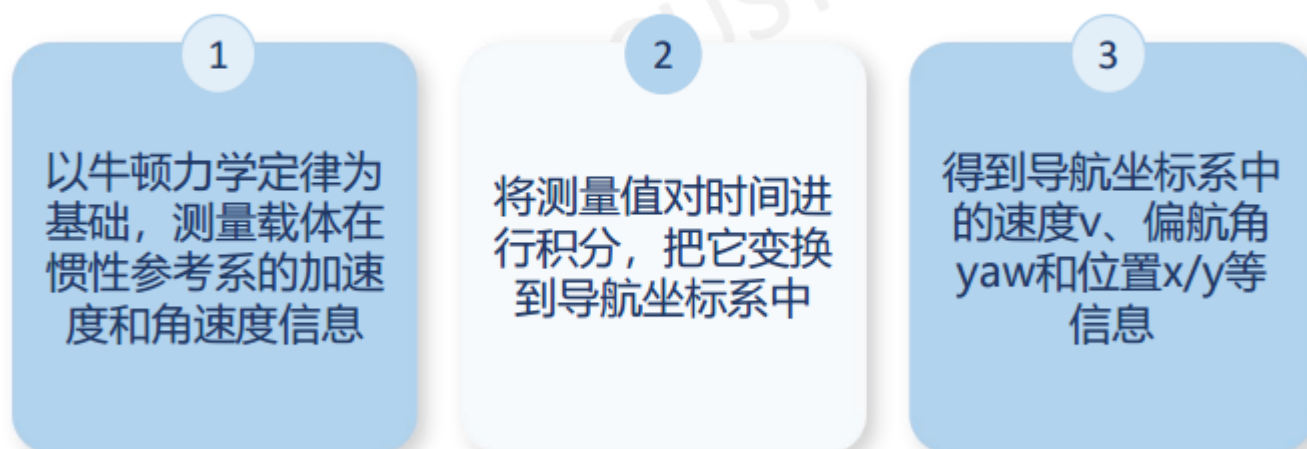
惯性导航系统从1维到2/3维需要考虑的因素：**需要补偿地球万有引力加速度；坐标系转换**

二维平面惯性导航

- 平台式系统：物理平台，加速度计敏感轴始终指向东向和北向
- 捷联式系统：数学平台，加速度计和陀螺与载体固联

惯性导航系统基本工作原理

基本工作原理



惯性导航系统优缺点

- 优点：不依赖于外部信息，可全天候工作，数据更新率高，精度、稳定性好，隐蔽性好
- 缺点：误差随时间累积，需要初始信息，成本高，笨重

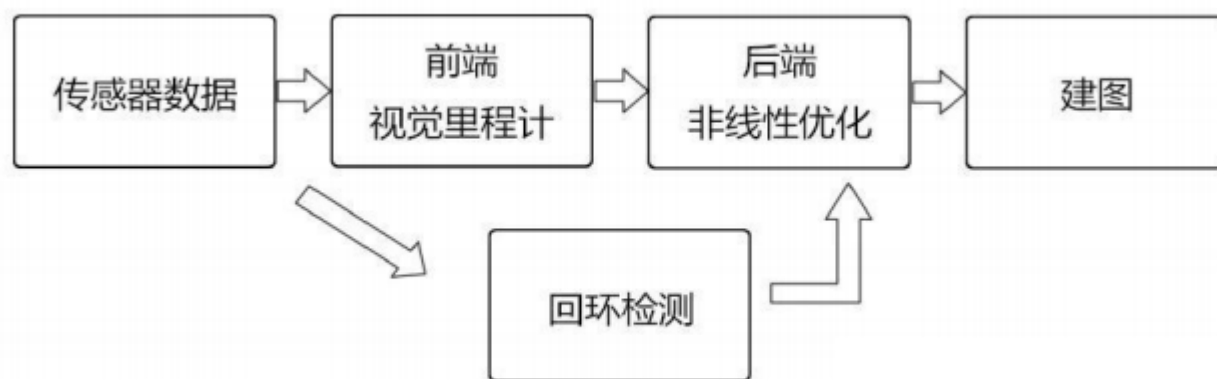
陀螺零偏是核心指标

相对测量能力 = 量程/零偏

不同传感器的定位融合使用**扩展卡尔曼滤波**

视觉SLAM

SLAM框架



SLAM框架

- 传感器数据：对图像信息进行读取和预处理
- 前端（视觉里程计）：根据相邻图像信息计算相机运动，构建局部地图，使用的方法有**特征点法和直接法**
- 后端（非线性优化）：接受不同时刻前端的测量信息和回环检测信息，图像信息具有噪声和误差，对他们进行优化，得到全局一致的轨迹地图
- 回环检测：判断机器人是否到达过先前位置，检测到回环就把信息交给后端处理
- 建图：根据轨迹信息建立地图

机器人移动的不确定性会导致**漂移问题**

如何计算同一个向量在不同坐标系里的坐标？**旋转加平移**

$$a' = Ra + t$$

前端里程计

主要内容：

- 点云预处理（地面点有三种情况）
- 特征点提取
- 特征点匹配
 - 通过特征点构建约束方程
 - 迭代求解约束方程得到位姿

特征点法

利用图像特征点作为SLAM中的**路标**

由**关键点和描述子**组成，是图像当中有代表性的部分（角点、区块、边缘），具有**可重复性、可区别性、高效性、本地性**

- 关键点：特征点在图像中的位置、大小、方向等
- 描述子：特征点周围的图像信息

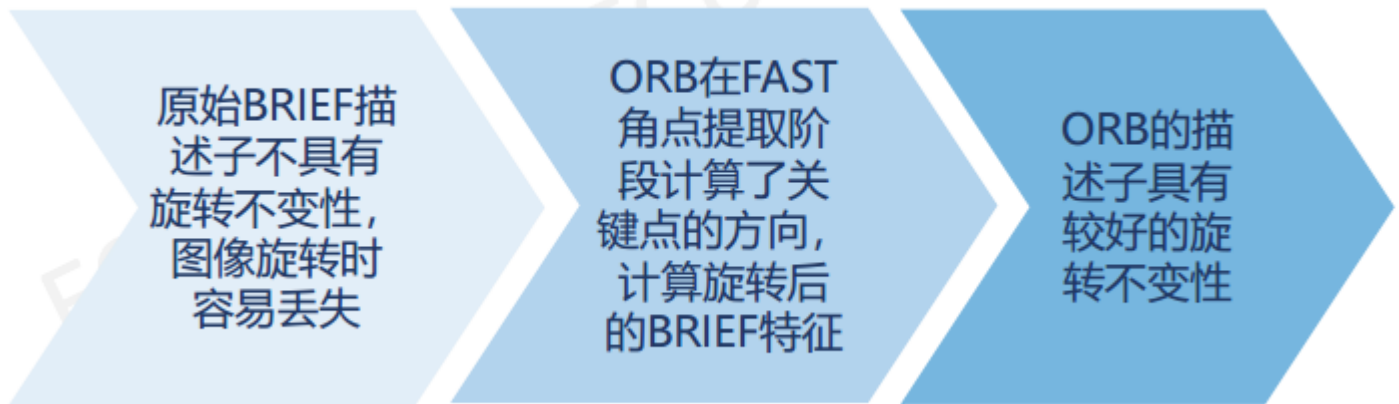
1. SIFT特征：考虑光照、尺度、旋转，精确，计算量大

2. FAST关键点：无描述子，不具有方向信息，有尺度问题，精度和鲁棒性低，计算速度快

3. ORB特征

- 关键点Oriented FAST（改进的FAST角点）：FAST连续N个点的灰度有明显差异，在FAST基础上计算旋转

- 描述子BRIEF：对前一步提取出来的关键点的周围区域进行描述
- 增添了尺度和旋转的描述：尺度不变性由构建**图像金字塔**实现，旋转不变性由**灰度质心法**实现



缺点：

- 关键点的提取和描述子的计算耗时
- 忽略了特征点以外的信息
- 弱纹理环境易失效

因此保留特征点，**只算关键点，不算描述子**，同时用**光流法**

特征匹配

- 暴力匹配：计算量大
- 快速近似最近邻FLANN：适合匹配点数量极多的情况

光流法

包括：

- LK光流：稀疏光流
- HK光流：稠密光流

LK光流的三个假设

- 灰度不变假设：同一空间点的像素灰度值在各个图像不变
- 时间持续性假设
- 空间一致性假设：窗口内像素具有相同运动

LK光流结果依赖于**图像梯度**

- 梯度不平滑，引起剧烈变化
- 局部梯度不能预测长期图像走向
- 若运动过快易达到局部最小值

解决方法：多层光流，引入**图像金字塔**

后端

处理方法：

- 批量式Batch：用过去和未来的信息

- 渐进式Incremental：用过去的信息，甚至是前一刻

回环检测

- 前端给出的是局部位姿约束，误差会累积
- 回环检测模块可以给出时刻更久的约束

前端 + 局部后端 = 视觉里程计

回环检测 + 全局后端 = SLAM

主流方法：基于外观的几何关系

核心问题：如何计算图像间的相似性

词袋模型

词袋模型 = 单词 + 字典

把单词（在字典出现过）出现的情况用一个向量表示，进而可以用这个向量表示整幅图像

相似度计算

- **TF-IDF（频率-逆文档频率）**：单词在图像中频率越高，则区分度越高；在字典中出现频率越低，则区分度越高

基于词袋的回环检测依赖外观，外观相似易被误认为回环，通常需进行**时间上的一致性检测**和**空间上的一致性检测**提升鲁棒性

状态估计与传感器融合

状态估计

卡尔曼滤波

使用条件：**线性系统、噪声是高斯白噪声、初始状态是高斯分布**

基本思想：利用前一刻的估计值和现时刻的观测值来更新对状态变量的估计，求出现在时刻的估计值（**预测、实测、修正**）

五个步骤：

1. 状态预测：加入噪声

$$x'_k = \alpha x_{k-1} + b\mu_k$$

2. 计算预测误差

$$P'_k = \alpha P_{k-1} \alpha^T$$

并且得出观测值 z_k

$$z_k = x_k + v_k$$

3. 计算卡尔曼增益

r 为测量噪声的均值

$$K_k = \frac{P'_k}{P'_k + r}$$


4. 计算最优估计值

$$\hat{x}_k = x'_k + K_k z_k - x'_k$$

5. 计算最优估计值误差：是为了求出下一时刻的预测误差

$$P_k = (1 - K_k)P'_k$$

总结：

预测	{	$x'_k = \alpha \hat{x}_{k-1} + bu_k$		$x'_k = A \hat{x}_{k-1} + B u_k$
		$P'_k = \alpha P_{k-1} \alpha^T$		$P'_k = A P_{k-1} A^T$
更新	{	$K_k = P'_k / (P'_k + r)$		$K_k = P'_k H^T (H P'_k H^T + R)^{-1}$
		$\hat{x}_k = x'_k + K_k (z_k - x'_k)$		$\hat{x}_k = x'_k + K_k (z_k - H x'_k)$
		$P_k = (1 - K_k) P'_k$		$P_k = (I - K_k H) P'_k$

其中 A 为状态转移矩阵， Q 为过程噪声的协方差矩阵， H 为状态域到观测域的转换矩阵， R 为观测噪声的协方差矩阵

扩展卡尔曼滤波

基本思想：使用线性变换来近似非线性变换，用雅可比矩阵代替转移矩阵和观测矩阵

无损卡尔曼滤波

核心：对非线性函数的概率分布进行近似，逼近状态的后验概率分布

传感器融合

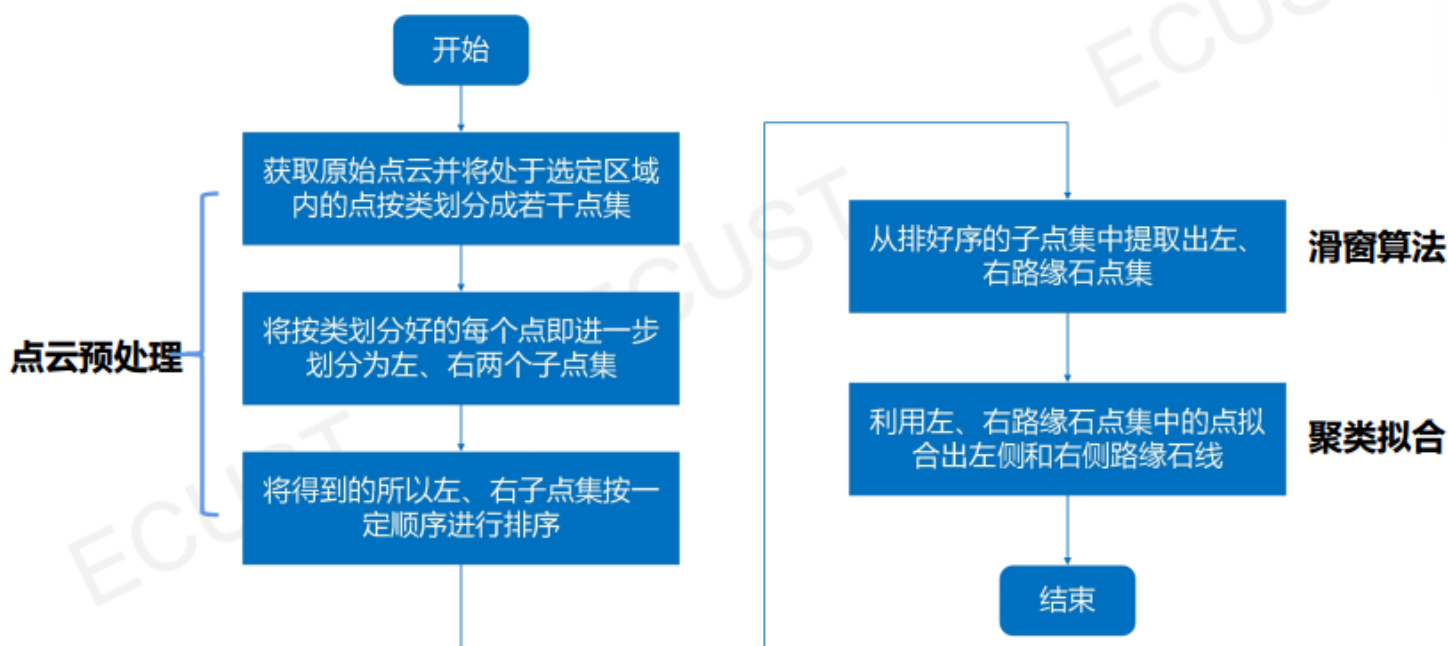
多传感器融合用了卡尔曼滤波

第三章

环境感知

基于激光雷达的路缘石检测

步骤：点云预处理、滑窗算法、聚类拟合



点集左右划分

- 按x轴划分点集
- 按参考线划分点集

点集左右划分

- 1、通过高度信息筛选，获得**最远点集**中投射在路面上的点
- 2、对最远点集中投射在路面点的坐标进行求和取平均，获得**参考点坐标**
- 3、将参考点与雷达中心进行连线得到参考线对所有点集进行左右划分，得到**左、右子点集**

路缘石点提取

1. 利用雷达投射在路缘石垂面上相邻两点高度之差较大这个特性提取路缘石点（与邻点高度差最大就是路缘石点）
2. 用滑动窗口方法对排序后的左右子点集进行筛查

路缘石线拟合

1. 对每个路缘石点进行两两连线，将得到的所有直线按斜率进行聚类
2. 对聚类结果中最大类内的直线取平均，即得到拟合的路缘石线

基于摄像机的车道线检测

基本步骤

1. 保留ROI、去畸变
2. 形态学处理（中值滤波、颜色空间转换、车道线提取）
 - 用中值滤波去除噪声

- 用颜色空间转换从RGB空间转到HSV空间
- 在HSV空间下给定颜色阈值范围，提取车道线

3. 车道线拟合

4. 坐标系转换

腐蚀：求局部最小值 **膨胀**：求局部最大值

机器学习

机器学习分类

- 监督学习
- 无监督学习
- 强化学习

经验风险最小化

风险：模型在数据真实分布上的平均损失

经验风险：模型在已知数据集上的平均损失

$$\overline{R(f)} = \frac{1}{n} \sum_{i=1}^n L(f(x_i), y_i)$$

经验风险最小化：选择能最小化经验风险的函数 f

欠拟合：模型学到的函数并没有满足经验风险最小化，训练误差太高

过拟合：训练误差低，测试误差高，模型学到了训练集上一些多余的规律（噪声）

梯度下降：向导数的反方向移动一小步来最小化目标函数（损失函数）

学习率

- 太大：震荡，无法收敛
- 太小：收敛缓慢

神经网络

感知机

单个感知机无法学习到异或的关系

$$z = \sum_{i=1}^d w_i x_i + b = w^T x + b$$

$$f(z) = \begin{cases} 1, & h > 0 \\ 0, & h \leq 0 \end{cases}$$

激活函数

常见激活函数：Sigmoid、ReLU、tanh、GeLU

性质：

- 连续可导非线性
- 导函数尽可能简单
- 导函数值域在合适区间

作用：

- 计算上更加高效
- 生物学合理性
- 单侧抑制、宽兴奋边界
- 缓解梯度消失

神经网络的优势：**拥有足够多神经元节点时，三层神经网络可以拟合任意函数**

前馈神经网络

One-Hot编码：单点编码，即对应类别的位置激活为1，其他位置为0

SoftMax分类：将神经元的输出得分转换为概率

前向传播和反向传播

前向传播：计算输出值，计算损失，存储中间值

反向传播：计算梯度，更新权重

深度神经网络

常见的正则化方法：

- 数据增强
- 早停
- 正则化（参数范数惩罚）
- Dropout
- 权重衰减
- SGD（随机梯度下降）

卷积神经网络的优势：

- 稀疏交互
- 权重共享
- 等变表示

卷积：使用卷积核在图片上滑动提取特征

池化：用相邻位置的总体统计特征来替换该位置的值，包括最大池化、平均池化

卷积的输出尺寸

$$\text{输出尺寸} = \frac{\text{输入图片尺寸} - \text{卷积核尺寸}}{\text{步长} + 1}$$

多通道卷积输出层的第三个维度是**卷积核的个数**

YOLO

是目标检测的模型

传统目标检测RCNN：采用**两阶段**的方法，生成候选区域，然后对这些候选区域进行分类和回归

YOLO：采用**单阶段**的方法，通过单个神经网络模型一次性完成目标检测和边界框回归

YOLOv3

- 采用3种不同尺度的特征图来进行目标检测
- 输入图像分辨率为416*416*3时，三种尺度特征图的大小分别为
 - 13*13*255：大感受野检测大目标
 - 26*26*255：中等感受野检测中等目标
 - 52*52*255：小感受野检测小目标
- 感受野越大，图像尺寸越小
- 使用卷积层代替池化层进行降采样
- 引入残差块，解决网络退化问题，防止过拟合

特征图深度的计算方式：

深度 = 一个cell与n个锚框匹配 × (中心偏移量+cell内是否含有物体 + 预测边界框的准确性 (含有物体时) + 物体类别概率

第四章

迁移学习

定义：机器学习领域用于解决标记数据难获取这一问题的手段

迁移学习的必要性

迁移学习的必要性

矛盾	传统机器学习	迁移学习
大数据与少标注	增加人工标注，但是昂贵且耗时	数据的迁移标注
大数据与弱计算	只能依赖强大计算能力，但是受众少	模型迁移
普适化模型与个性化需求	通用模型无法满足个性化需求	模型自适应调整
特定应用	冷启动问题无法解决	数据迁移

与传统机器学习的对比

比较项目	传统机器学习	迁移学习
数据分布	训练和测试数据服从相同的分布	训练和测试数据服从不同的分布
数据标注	需要足够的数据标注来训练模型	不需要足够的数据标注
模型	每个任务分别建模	模型可以在不同任务之间迁移

常见的迁移学习方法分类

- 基于实例的迁移
- 基于特征的迁移
- 基于模型的迁移
- 基于关系的迁移

移植模型

取决于两个因素：新应用数据量大小、新应用和原始模型的相似度

数据集数量	应用相似度	模型训练方法
较大	较高	微调模型
较大	较低	微调或者重新训练
较小	较高	对全连接分类层进行修改训练
较小	较低	重新设计、重新训练模型

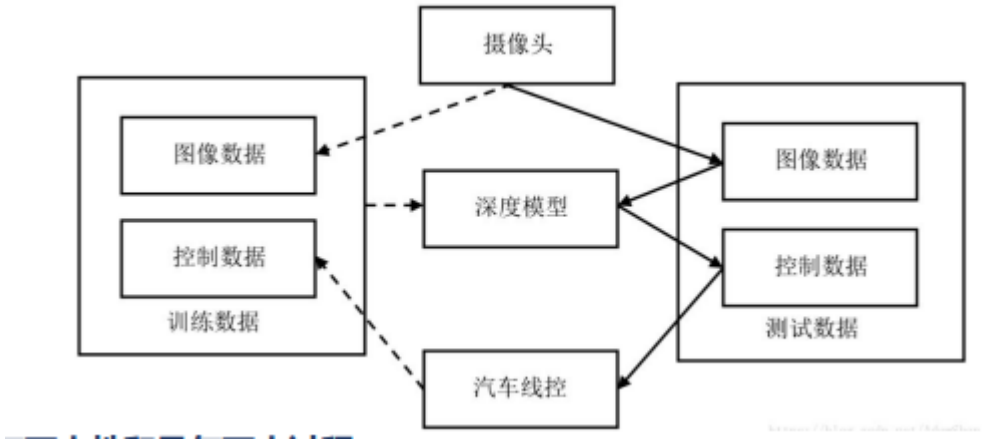
强化学习

端到端无人驾驶

基本思路

- 采集数据
- 网络训练
- 测试

模型框架



马尔可夫性和马尔可夫过程

马尔可夫性：系统的下一个状态 S_{t+1} 和当前状态 S_t 有关

马尔可夫过程：它是一个二元组 (S, P) ，其中 S 为有限状态机， P 为状态转移概率矩阵，该矩阵为

$$P = \begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \cdots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{pmatrix}$$

强化学习问题就是一个马尔可夫过程，由 $\langle S, A, P, R \rangle$ 表示

- S 为有限状态空间集合
- A 为有限动作空间集合
- P 为状态转移概率
- R 为奖惩函数

非确定性策略和确定性策略

非确定性策略：相同状态，输出动作不唯一，满足一定的概率分布

确定性策略：相同状态，输出动作确定

及时奖励和累计期望奖励

及时奖励：实时反馈给智能体的奖励

累计期望奖励：指一个过程的总奖励期望， γ 为折扣因子

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0} \gamma^k R_{t+k+1}$$

后面的强化学习知识移步 [LLM](#)

DQN与Q-Learning

Q-Learning（基于价值的RL算法）

学习并使用**状态-动作值函数**，但其状态空间离散，需要维护一个Q-table

使用了 ϵ -greedy 算法

- 动作选择：非贪心
 - 价值更新：贪心
1. 探索环境，尝试不同动作获得最佳策略（小概率探索）
 2. 不尝试新动作，利用已知可带来很大奖励的动作（大概率利用）

DQN

深度Q网络，基于深度学习的Q-learning算法，状态空间连续，使用一个函数

$$Q(s, a, w)$$

来近似动作价值 $Q(s, a)$ ， w 是网络的参数

两大重要思想

- 经验回放：将智能体的交互经验存储在一个经验池中，并在智能体进行更新时从中随机采样一小批经验来进行训练。
 - 作用：有利于提高智能体的学习效率，降低采样成本，减少数据相关性，提升收敛速度
- 目标网络：和评估网络的结构一样，只是参数和参数的更新频率不同。评估网络的参数随着训练不断更新，目标网络的参数是定期更新的。

- 评估网络作用：估计Q值函数，决定智能体的动作选择
- 目标网络作用：提高稳定的目标Q值，减少估计偏差，降低了当前Q值和目标Q值的相关性

策略梯度

学习一个参数化的策略函数，输入状态 s ，输出各个动作的执行概率

优化目标：增大产生高奖励动作的概率，降低产生低奖励动作的概率

通过最大似然和蒙特卡洛法计算梯度得到最终梯度公式：

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=1}^T \left[\sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t, t | s_t, t) \left(\sum_{t=0}^T r(s_t, t, a_t, t) \right) \right]$$

当前第n回玩游戏时，t时刻机器所看到的画面情况下做出a行为的概率

第n回完游戏时总的奖励

如果换成t时刻的奖励会发生什么后果？

会导致只注重当前做的动作的回报，而不会注重整个游戏的过程中的奖励。容易陷入局部最优，无法更新到最优解。比如在射击游戏里，只有开火能得到奖励，那么就会导致机器只会开火。

注意：

- 在一轮的学习中使用同一个策略
- 使用梯度上升的优化策略

DDPG算法

对DQN的改进，无模型

深度确定性策略梯度算法：使用演员-评论家（Actor-Critic）算法作为其基本框架

对于策略函数和价值函数均使用了双重神经网络模型架构，并引入了经验回放机制

- Critic目标网络用于近似估计下一时刻的状态动作的Q值函数，训练网络输出当前时刻的状态动作的Q值函数
 - 最小化损失函数更新参数
- Actor目标网络用于提供下一个状态的策略，训练网络则提高当前状态的策略
 - 策略梯度更新参数

第五章

深搜和广搜

深度优先搜索：沿着一条路径不断向下搜索直到不能继续为止，然后折返搜索下一跳

- 栈，先进后出

广度优先搜索：根据离起点的距离，按照从近到远的顺序对各节点进行搜索

- 队列，先进先出

Dijkstra算法

基本思想：由近到远把所有点的最短路径算出来

伪代码：

```

获取起点坐标、目标点坐标，把起点放进openlist
while (openlist不为空列表)
{
    取openlist中路程最小的点为中心点，从openlist剔除，放入closedlist
    if (中心点是目标点)
        {从目标点开始通过父节点反推，提取出路径，break跳出循环}
    else
        {遍历相邻的八个点
            {if(点已存在closedlist中 or 遇到障碍物)
                {跳过，不计算这个点}
            else
                {累计路径
                    if(点已存在openlist中)
                        if(累计路程<原来记录的路程)
                            {替换数据}
                        else {加入openlist}
                }
            }
        }
    openlist从小到大排序
}

```

缺点：

- 需要遍历整个离散空间，搜索效率低
- 计算开销大

A*算法

基本思想：以从起点到该点的距离(G)加上该点到终点的估计距离(H^*)之和作为该点的优先级(F^*)

$$F^* = G + H^*$$

混合A*算法

加入了 θ 表示朝向，改变了节点扩展方式，扩展了节点字段，完善了碰撞检测方式

比较A*和混合A*算法在节点字段定义、节点扩展方式、碰撞检测方式及距离定义方面的异同点

答：

- **节点字段定义：**经典A*包含的字段包括：F, G, H, 父节点。混合A*新增的字段包括：对应的位姿 $[x, y, \theta]$ ；节点是由何种 v, φ 扩展而来；由父节点到本节点的轨迹。
- **节点扩展方式：**经典A*扩展节点的方式就是找相邻节点。混合A*的扩展需要**考虑车辆动力学**。要求车辆在单位时间内，以恒定的速度和转角做匀速运动的，轨迹末端落入的三维网格就是要扩展的子节点。
- **碰撞检测方式：**经典A*算法中，判断是否碰撞只是检查对应网格是否被占据。混合A*需要将整条轨迹均匀采样，检测各采样时刻车辆位姿是否与障碍物碰撞。
- **距离定义：丰富了距离的含义。**混合A*从当前节点扩展到子节点的距离可以直接由 v *单位时间得到，此外，加入了对控制量 v, φ 频繁切换的惩罚。从当前节点到终点的估算距离综合考虑了车辆动力学和碰撞检测距离。

局部路径规划

TEB算法

将轨迹表示为一连串的时空位姿

局部优化：它是一个多目标优化问题，大多数目标是局部的，只有一小部分参数相关

- 每个目标函数只与elasticband中的某几个连续状态有关

约束目标

- 跟随路径+避障
- 速度/加速度约束
- 运动学限制
- 最快路径约束

优化算法：使用开源框架g2o（图优化法）

适合动态避障

行为规划

位于全局任务规划和底层的动作规划层之间

设计准则：合理性、实时性

有限状态机（FSM）

基本原理：对外界的输入产生数量有限的响应

分类：确定型自动机、非确定型自动机

存在的问题：可维护性差、可扩展性差、复用性差

分层有限状态机（HFSM）

新增了一个超级状态，将同一类型的一组状态合为一个集合

斯坦福大学Junior将顶层的驾驶行为分成了13个超级状态，无人车正常行驶中，状态机几乎处在普通驾驶模式

优点：

- 分层结构
- 分层继承
- 分层封装

路径生成

使用三次样条插值算法，拟合出来的曲线更符合车辆运动学规律

性质：

- 衔接点函数值连续
- 一阶导和二阶导连续可导
- 自由边界三次样条的边界二阶导数连续

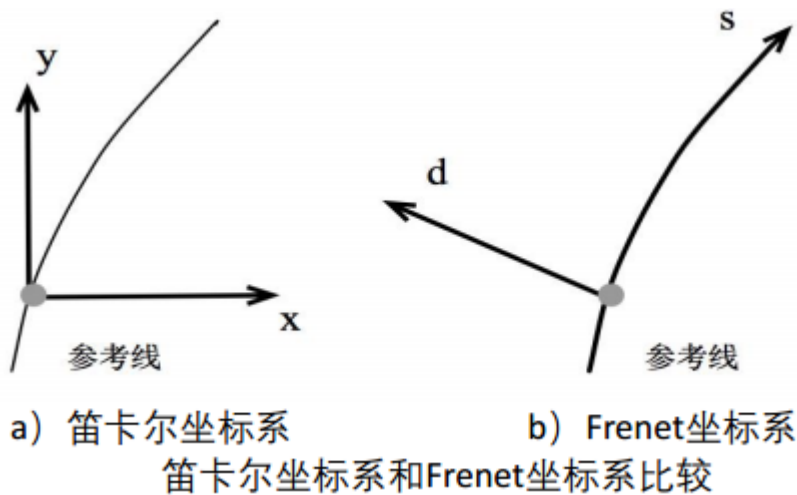
动作规划

包括轨迹规划和速度规划

Jerk：加速度的变化率，称为加加速度，过高会引起不适

Frenet坐标系：使用参考线的切线向量（s方向，纵向）和法线向量（d方向，横向）建立一个坐标系

- 好处：简化了道路曲线拟合的问题，且两个方向的速度计算相对简单



横向轨迹优化

损失函数的设计：

$$C_d = k_j J_t(d(t)) + k_t T + k_d d_1^2$$

包含三个惩罚项：

- $k_j J_t(d(t))$ ：舒适度惩罚项，惩罚Jerk大的轨迹
- $k_t T$ ：效率惩罚项，制动应当迅速，时间短
- $k_d d_1^2$ ：准确性惩罚项，目标状态不能离道路中心线太远

k_j 、 k_t 、 k_d 是惩罚项的权重值，决定损失函数更加偏向哪一方面的优化

纵向轨迹优化

优化场景：

- 跟车
- 汇流和停车
- 车速保持

车速保持的损失函数：

$$C_s = k_j J_t(s(t)) + k_t T + k_s (\dot{s}_1 - \dot{s}_c)^2$$

- $k_j J_t(s(t))$ ：舒适度惩罚项，惩罚Jerk大的轨迹
- $k_t T$ ：效率惩罚项，制动应当迅速，时间短
- $k_s (\dot{s}_1 - \dot{s}_c)^2$ ：让目标配置的纵向速度尽可能接近设定速度

合并两个方向的损失函数

$$C_{total} = k_{lat} C_d + k_{lon} C_s$$

⬆

碰撞避免

检查内容包括：

在完成所有备选轨迹的损失计算后进行一次轨迹检查，检查内容包括：

- 检查s方向上的速度是否超过设定的**最大限速**
- 检查s方向的加速度是否超过设定的**最大加速度**
- 检查轨迹的曲率是否超过**最大曲率**
- 检查轨迹是否会引起**碰撞**（事故）

第六章

自行车模型

几个假设

- 忽略车辆在垂直方向的运动
- 假设车辆结构和自行车一样，前后轮胎**各用一个轮胎**描述
- 假设车辆运动与自行车一样，前面轮胎可以控制车辆转角

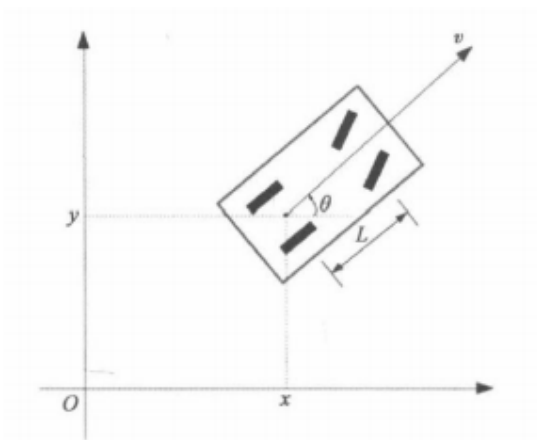


图1.1 二维平面上的车辆描述

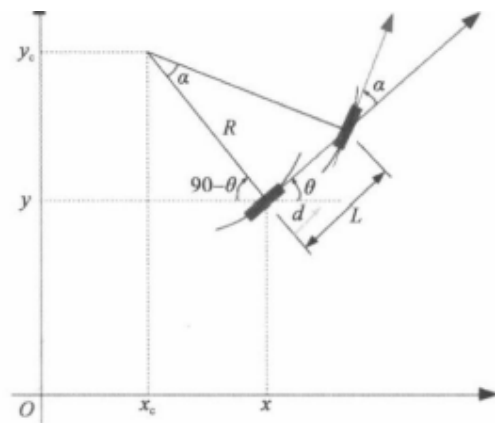


图1.2 车辆的自行车模型

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

运动学自行车模型

使用四个变量描述车辆状态：

- 车辆的 x 坐标
- 车辆的 y 坐标
- 车辆的偏航角
- 车辆的速度

两个控制输入：

- 车辆的加速度
- 方向盘的转角：前轮当前的转角

各个状态量的更新公式：

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 $\delta_r = 0$ 。

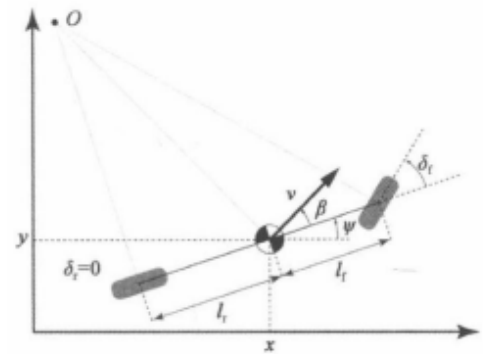


图1.3 一个简单的运动学自行车模型

动力学自行车模型

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

考虑各种力的作用，包括：

- 侧向力：促使车辆横向移动
- 纵向力：促使车辆前后移动

轮胎是车辆运动的一个重要的力来源

体现了由frenet坐标系下车身的纵向和侧向速度来推导笛卡尔坐标系下X和Y两个方向上的车辆行驶速度

PID控制

P（比例）

作用：使输出和误差成比例变化，纠正当前误差

过大易产生振荡或超调，过小反应迟钝，稳态误差大

I（积分）

作用：消除静态误差

过大反应滞后，超调严重，过小则无法消除静态误差

D（微分）

作用：预测未来误差趋势，抑制系统变化过快，使系统响应速度变快

过大对噪声敏感，引起抖动，过小易增大超调

参数整定步骤：先P再I后D

大P小D：欠阻尼 小P大D：过阻尼

模型预测控制（MPC）

一般用于车辆的**横向控制**，相较于PID控制，解决了PID的**延迟问题**

三个因素：

- 预测模型：使用预测模型对未来一个有限的时间段对车辆行为进行预测
- 在线滚动优化：求解一个优化问题，找出一组最优的**未来控制输入序列**，最小化代价函数
- 反馈校正：下一个时间步根据新的状态优化下次预测

轨迹追踪

追踪轨迹的两类方法：

- 基于几何追踪：**纯追踪**
- 基于模型预测