

知
识
点

敲黑板，本文需要学习的知识点有

路径规划 激光雷达

感知 定位

通信 行为决策

ROS是机器人学习和无人车学习最好Linux平台软件，资源丰富。无人车的规划、控制算法通常运行在Linux系统上，各个模块通常使用ROS进行连接。

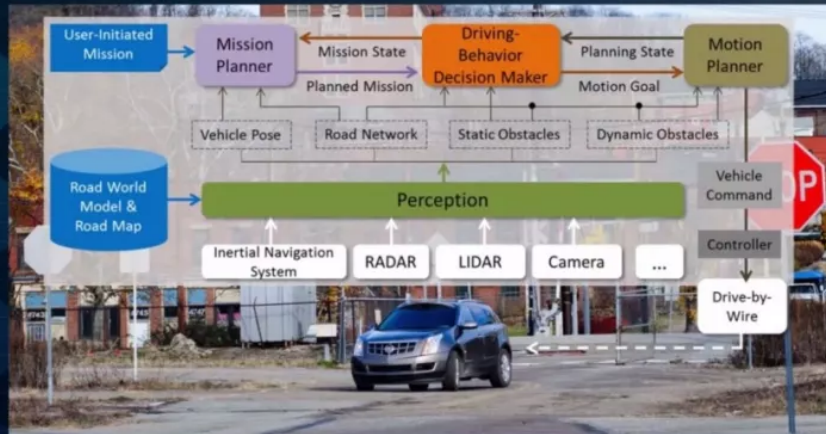
完整版视频学习 请戳『[阅读原文](#)』

以下，ENJOY



引入ROS的背景介绍

自动驾驶 – 一个复杂的系统性工程



▲ 自动驾驶系统组成

如上图所示：自动驾驶系统包括**障碍物检测**、**行为决策**、**路径规划**等一系列复杂的工程模块，同时还要支持激光雷达、相机、GPS等一系列传感器的实时数据收集和实时处理。如何将这功能模块相互独立又相互交互集成一起，构建成一个稳定的自动驾驶系统是一个巨大的挑战，也是自动驾驶计算框架所承载的基本功能。

首先自动驾驶系统还处在一个快速发展的阶段，**算法**和**整体方案**还在持续迭代优化。使用框架开发，可以将更多的精力放在算法模块的迭代上。开发者不需要关心具体配置管理、部署运行、底层通信等功能。

其次是**感知**、**定位**、**决策控制**等模块各自承载了一部分独立功能，相互之间有一定的数据依赖。框架能够在开发阶段减少各个模块之间的耦合，在运行阶段可以将各个模块串联起来。

最后是自动驾驶涉及大量的图像点云处理算法，对于可视化和调试工具有很大的需求。比如调试过程中，**障碍物检测**需要看Detection的那个框准不准；**路径规划**需要看自动驾驶算法规划的路径是否符合当前的一些决策行为；**定位**也需要看车辆当前是否在准确的车道内。



背景介绍

- 开发工具包

- 完整的包管理和工程结构
- 庞大的基础库
- 多语言接口支持

- 消息驱动的异步运行模型
- 抽象的P2P通信接口
- 自定义的消息格式

- 可视化的调试工具
- 消息查看、存储、回放工具



ROS主要工具包

▲ ROS的基本框架

上图是一个ROS的基本框架，自动驾驶底层通信框架选择ROS，主要有三个方面的原因：

1. ROS是一个比较强大、灵活的机器人编程框架。从软件架构的层面来说，它是一个基于消息传递的分布式多进程框架，很早就被机器人行业广泛使用。很多著名的机器人开发框架，如基于**四元数的坐标转换、3D点预处理驱动、定位算法、SLAM**等都是基于ROS开发的开源的现成方案。
2. ROS基于消息机制，开发者可以根据功能把软件拆分成独立的子模块，子模块通过不断的组合能够建立起比较复杂的系统来完成复杂的功能。
3. ROS是学术界广泛使用的一个框架，对实验各种新算法提供了一些支持。

基于以上三点，我们在初期选择ROS作为自动驾驶底层框架，用于快速验证顶层算法和技术方案。

