

Apollo自动驾驶入门课程第⑨讲 — 控制（上）



上周我们发布了无人驾驶技术的规划篇，车辆基于高精地图、感知和预测模块的数据来进行这一规划。

本期我们将介绍和控制相关的知识，具体讲解最常用的控制算法——**PID控制器**。自动驾驶的入门课程已经接近尾声，开发者学习的热情依然高涨。希望开发者们在完成入门课程后，继续深入学习无人驾驶技术，在无人驾驶领域越走越远。

第九课，控制（上）

1 | 简介

控制是驱使车辆前行的策略。对于汽车而言，最基本的控制输入为转向、加速和制动。通常，控制器使用一系列路径点来接收轨迹。控制器的任务是使用控制输入让车辆通过这些路径点。

首先，控制器必须**准确**，这意味着它应避免偏离目标轨迹。这对于安全来说，尤为重要。即使路面潮湿或者道路比较陡峭，控制器也需要准确地执行轨迹。其次，控制策略对汽车应该具备**可行性**。例如，如果你的汽车向北行驶，而你希望它立即向东转。你可以在游戏中做到这一点，但在现实中无法实现。最后，需要考虑的是**平稳度**。舒适的驾驶非常重要。如果车辆行驶得不规律，

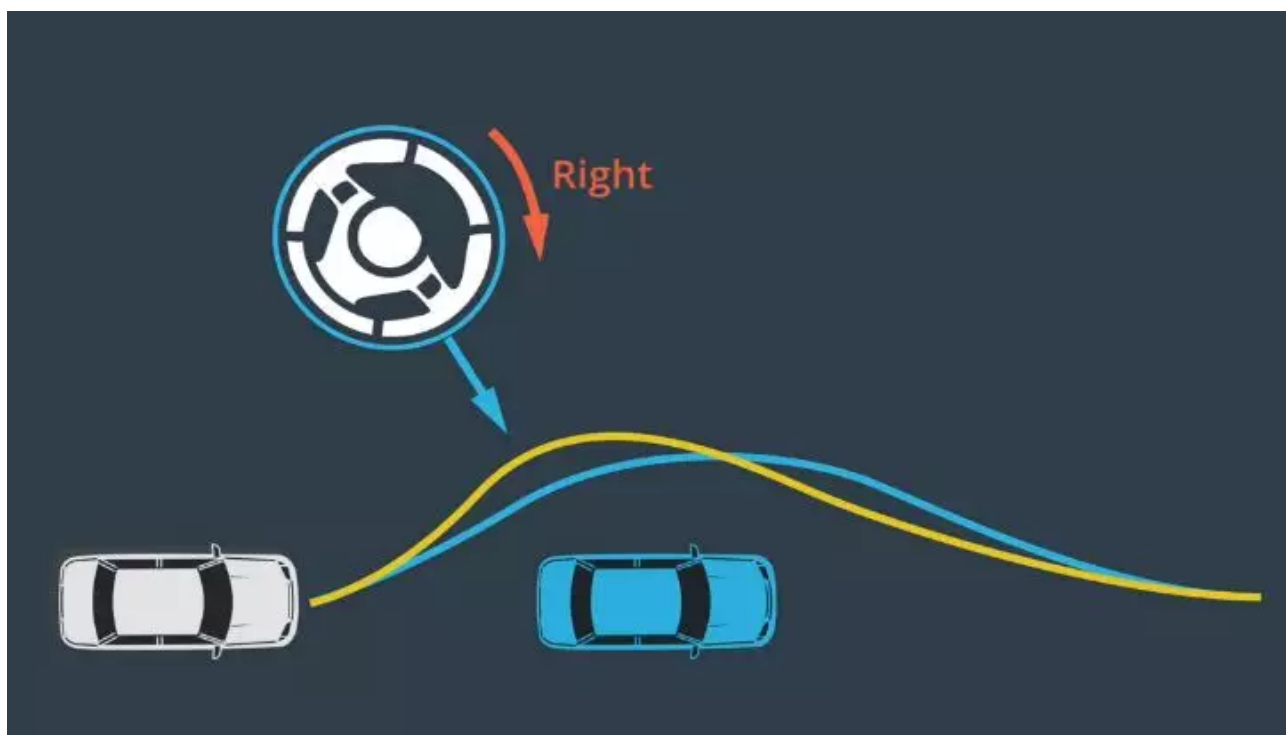
那乘客永远不会想再次乘坐它了。要使控制顺利进行，驱动必须是连续的。这意味着你应避免突然转向、加速或制动。

总之，我们的目标是使用可行的控制输入，最大限度地降低与目标轨迹的偏差、最大限度地提供乘客的舒适度。有三种可用于实现这些目标的控制策略：**比例积分微分控制（或PID）**、**线性二次调节器（或LQR）**、**模型预测控制（或MPC）**。

2 | 控制流程

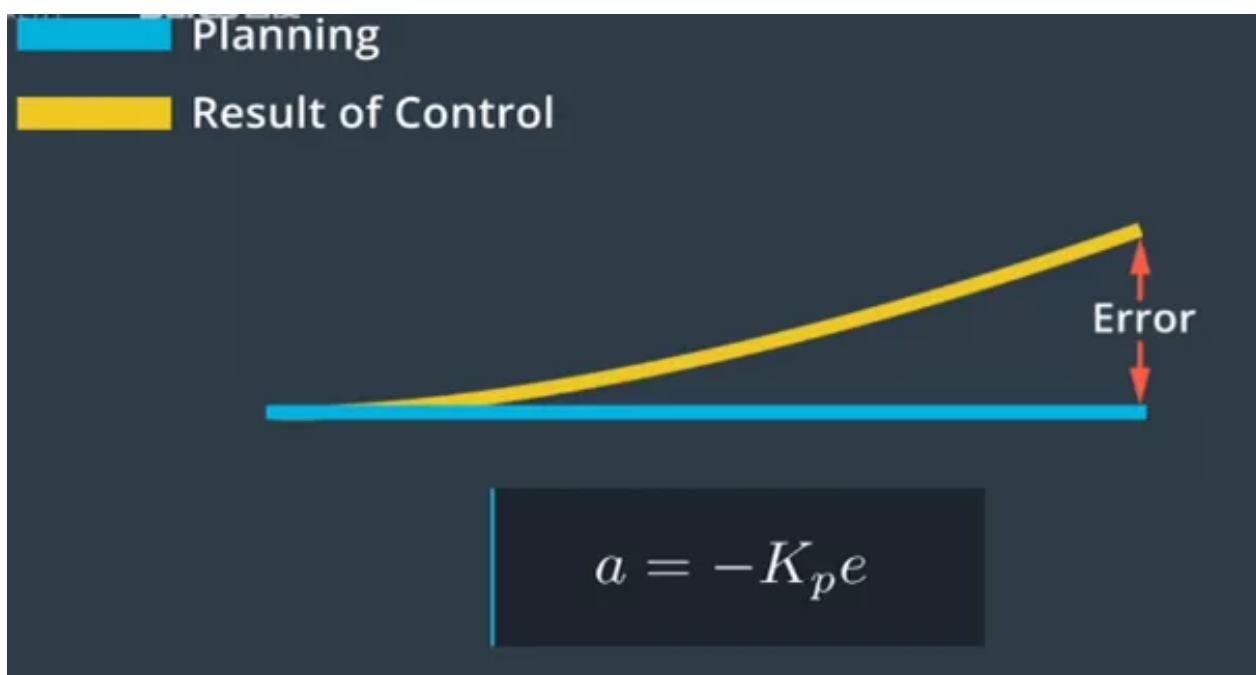
控制器预计有两种输入：**目标轨迹与车辆状态**。目标轨迹来自规划模块，在每个轨迹点，规划模块指定一个位置和参考速度。在每个时间戳都对轨迹进行更新。我们还需要了解车辆状态，车辆状态包括：通过本地化模块来计算的车辆位置、从车辆内部传感器获取的数据（如速度、转向和加速度）。我们使用这两个输入来计算目标轨迹与实际行进轨迹之间的偏差。

控制器的输出是控制输入（转向、加速和制动）的值。当偏离目标轨迹时，我们希望采取行动来纠正这种偏差。对于普通汽车，我们使用方向盘控制行驶方向（即转向）、使用油门加速、使用刹车减速（即制动）。这也是无人驾驶汽车所做的。一旦将这三个值传递给车辆，汽车实际上已经开始无人驾驶了。之后将介绍不同的控制算法，如何计算这三个输出-转向、加速和制动。

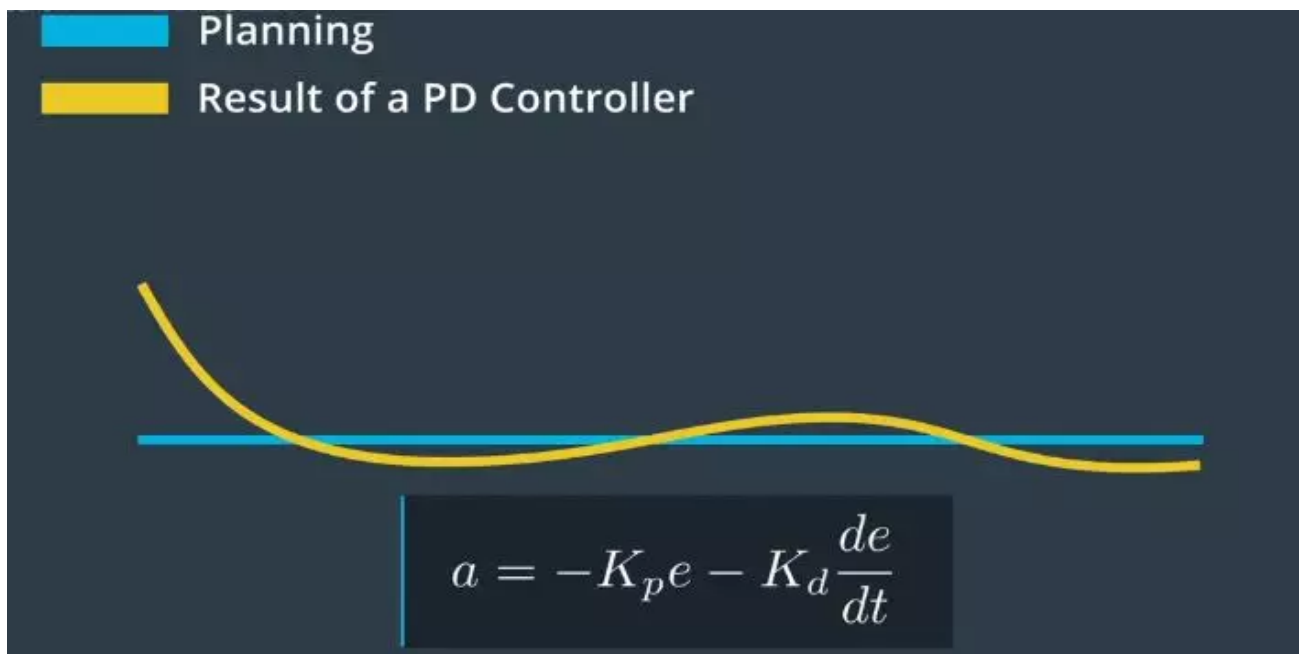


3 | PID 控制

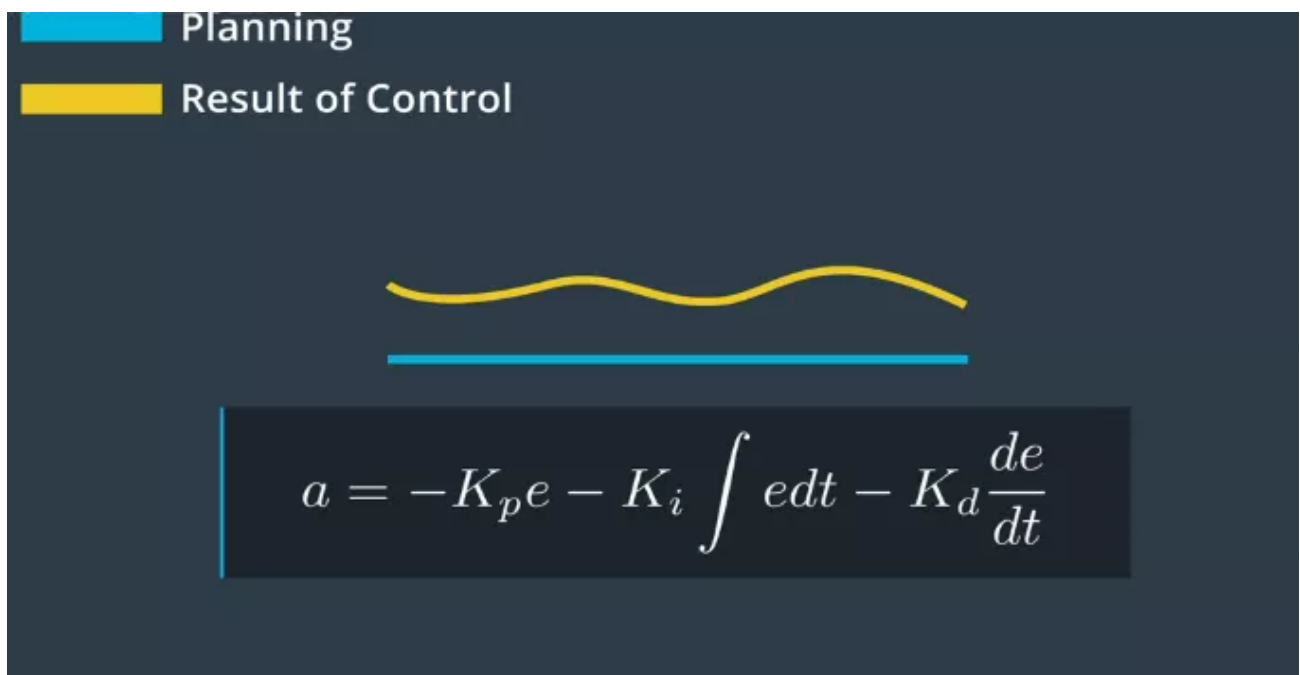
首先介绍的算法为**PID控制**，这个控制器的优点在于它非常简单，只需要知道与目标轨迹有多大的偏离。PID的第一组件为**P代表“比例”**(Proportional)。设想一辆车正试图遵循目标轨迹，P控制器在车辆开始偏离时立即将其拉回目标轨迹。比例控制意味着，车辆偏离越远，控制器越难将其拉回目标轨迹。



在实践中P控制器的一个问题在于，它很容易超出参考轨迹。当车辆越来越接近目标轨迹时，我们需要控制器更加稳定。PID控制器中的D项致力于使运动处于稳定状态，**D代表“微分”**（Derivative）。PD控制器类似于P控制器，它增加了一个阻尼项，可最大限度地减少控制器输出的变化速度。



PID控制器中的最后一项**I代表积分**（Integral），该项负责纠正车辆的任何系统性偏差。例如，转向可能失准，这可能造成恒定的转向偏移。在这种情况下，我们需要稍微向一侧转向以保持直行。为解决这一问题，控制器会对系统的累积误差进行惩罚。我们可以将P、I和D组件结合构成PID控制器。



PID控制器很简单，但它在很多情况下的效果很好。对于PID控制器，你只需要知道你的车辆与目标轨迹之间的偏差。但是PID控制器只是一种线性算法，对于非常复杂的系统而言，这是不够的。例如，为控制具有多个关节的四轴飞行器或机器人，我们需要建立机器人的物理模型。对无人驾驶而言，我们需要应用不同的PID控制器来控制转向和加速，这意味着很难将横向和纵向控制结合起来。另一个问题在于PID控制器依赖于实时误差测量，这意味着受到测量延迟限制时可能会失效。