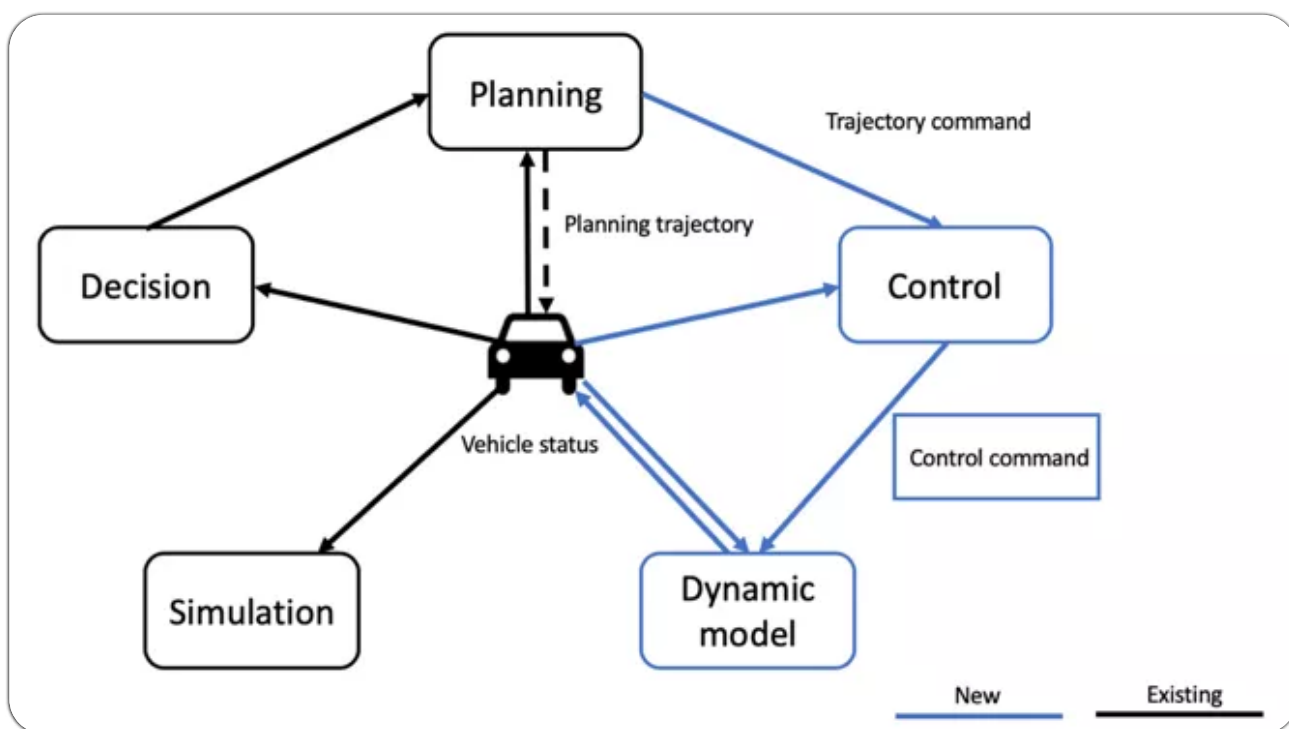


Apollo公开课 | 控制在环仿真技术分享

在Apollo 5.0中，我们升级了**Simulation**，增加了**Dynamic model**，使得控制可以入环仿真。由原来的决策-运动规划-完美控制闭环，升级到现在的**决策-运动规划-控制-动力学模型**。为什么要这么做呢？

首先，使得控制模块可以仿真入环，在**开发**和**调试效率**上可以得到很大的提升，进一步摆脱对上车调试的依赖。那么，现在Baidu Apollo开放的控制入环仿真具备什么样的能力呢？



▲决策-运动规划-控制-动力学模型图



对车辆动力学模型进行仿真

Apollo的动力学仿真是一个完全基于学习的模型，和传统车厂的开发模型相比，它具有**模型精度较好**，**开发成本较低**，**模型的灵活度和泛化程度相对较高**的特点。后两者对于Apollo平台和生态合作伙伴尤其重要。各种算法的对比如下图所示。

Vehicle Dynamic Models in the Simulation

Algorithms	Model Granularity	Model Generalization	Model Develop Cost	Model Flexibility
Perfect Control (No Dynamic model)	N/A	N/A	Low	High
Point-Mass Model	Low	High	Low	High
Manufactory Model (Rule based)	High	Low	High	Low
<u>Learning-Based Model</u>	Medium	High	Medium	High

▲车辆动力学模型仿真图



扩展到不同的车辆模型

下图是Apollo的一些合作伙伴，其中大概有四分之一是车企。他们每个都有不同的模型，不同的车辆。

最右边两张图是用来展示的Apollo支持的车辆。对于我们来说，能够很快的接入一辆各种模型参数新的模型车辆，**生成模型是以天为计算的**。而传统车企开发一个新的车辆模型的周期一般来说是以月为单位计算，通过仿真，我们的效率可以大大提高。

Easy to be extended to different vehicles



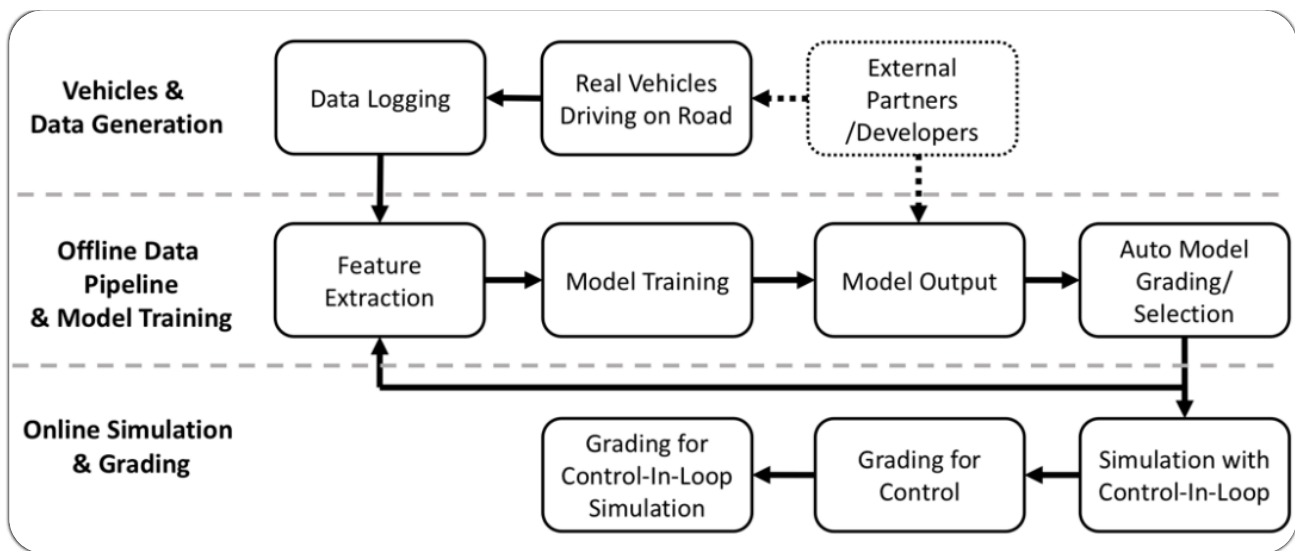
▲不同车辆模型合作伙伴



实现实车模型仿真平台

那么，在Apollo我们是怎么实现这样一个仿真平台的呢？我们把这个问题归成三类：

- 首先要解决**车端数据采集**的问题；
- 其次是用**Apollo fuel流水线**进行数据处理；
- 最后**跟后端进行结合**，把训练好的模型开放给大家，如下图所示。

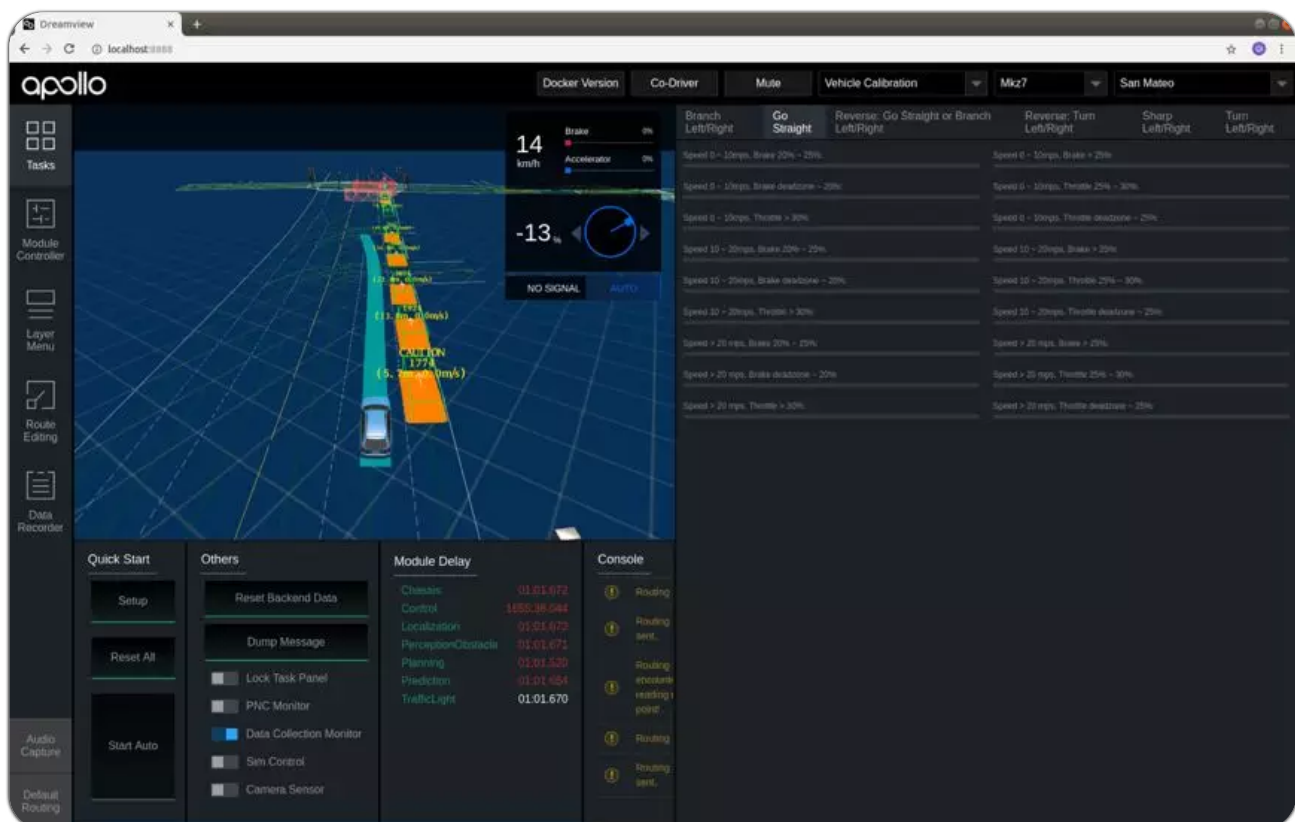


▲训练好的模型流程图

首先我们看第一步数据采集。

整体的流程是希望合作伙伴或者我们自己，在真正实车上路的时候，自动采集一些符合需求的数据，然后上传上来，这是如图所示的页面所需要解决的问题。

对于开发者来说非常简单，手动开这辆车，看一下对于油门刹车或者转弯时参数是什么样的量。在数据采集过程中，图示右边展示的数据会逐渐的增加，当所有的进条都在百分之百的时候，可以进行数据处理。



▲数据处理

第二步相对简单，是一个通用的**Data 流水线**，包括特征提取，模型的训练生成。同时我们支持外界开发者直接把他们的模型贡献在里面。对于我们整体生成的一个模型，不管是车厂直接生成的模型，或者自己学习训练的模型，**进行自动的模型打分**。

根据打分结果分类，我们把同样控制指令在仿真里面跑出来的结果跟真车跑出来的结果最相近的模型选为得分最高的模型。**根据得分最高模型进行模型升级**。

模型的集成效果

最后，选择最好的模型放在仿真的后端，在这个模型中**设置符合控制和车辆的一些参数**，同时**建立一个整体的评价标准**。我们再把这些我们觉得真正像实车的模型开放给大家使用。

下图是其中的两个例子，我们的模型在这Lincoln和Ford两辆车上都进行了训练，以及进行了一些效果总结。在Apollo 5.0 中，我们已经把**MKZ的模型**开放给大家。

Selective Learning Results



	Lincoln MKZ	Ford Transit
Power	Hybrid	Electric
Size(m^3)	$4.93 \times 1.86 \times 1.47$	$4.82 \times 1.84 \times 1.85$
Weight(kg)	1,769	2,449
Type	Sedan	Van

▲ Lincoln和Ford模型数据

首先展示了在纵向找出刹车跟油门跟加速度之间的**非线性关系**，这里面是一秒当中的加速度在不同的模型的结果的对比。

下面是一分钟的模型，这个对于模型转弯程度是一个很好的标准。视频画面的左边是我们的仿真效果，右边是流程图。**仿真里面将仿真结果跟实际录入的结果的相似程度来衡量仿真的优劣，结果一致才是好的。**

这里大家着重关注一下右边的模型，反映了我们的一些结果。**我们用故意劣化的控制参数来在仿真中控制车辆，发现仿真中不断出现replanning的情况，加速度的跳变非常频繁**，这点和同样的控制参数下面车辆测试的结果是一致的。

未来的技术规划

我们的模型跟车辆的模型相比优势是什么呢？首先对于车辆本身来说，数据是非常清晰的。例如延时分为很多小部分，比如我们发出的命令到车辆收到命令并进行相应，转弯的时候车辆在不同的速度下的相应时间是不同的。

- 下一步将在延时这一块做得更细；
- 另外是**车辆的怠速**；
- 第三是跟**定位**相关的问题。

在整个模型的数据采集训练中，减小加速和本地测量之间的差距。最后是怎么在整个动力学系统里面加入一些自然误差，来让我们的Apollo更像一个车。

What is next?

1. More detailed vehicle latency.
2. Vehicle Idling speed.
3. Close gaps between acceleration / location Measurement.
4. Modeling of measurement error.

▲控制在环仿真技术的四个优势