

四、项目建设内容和方法

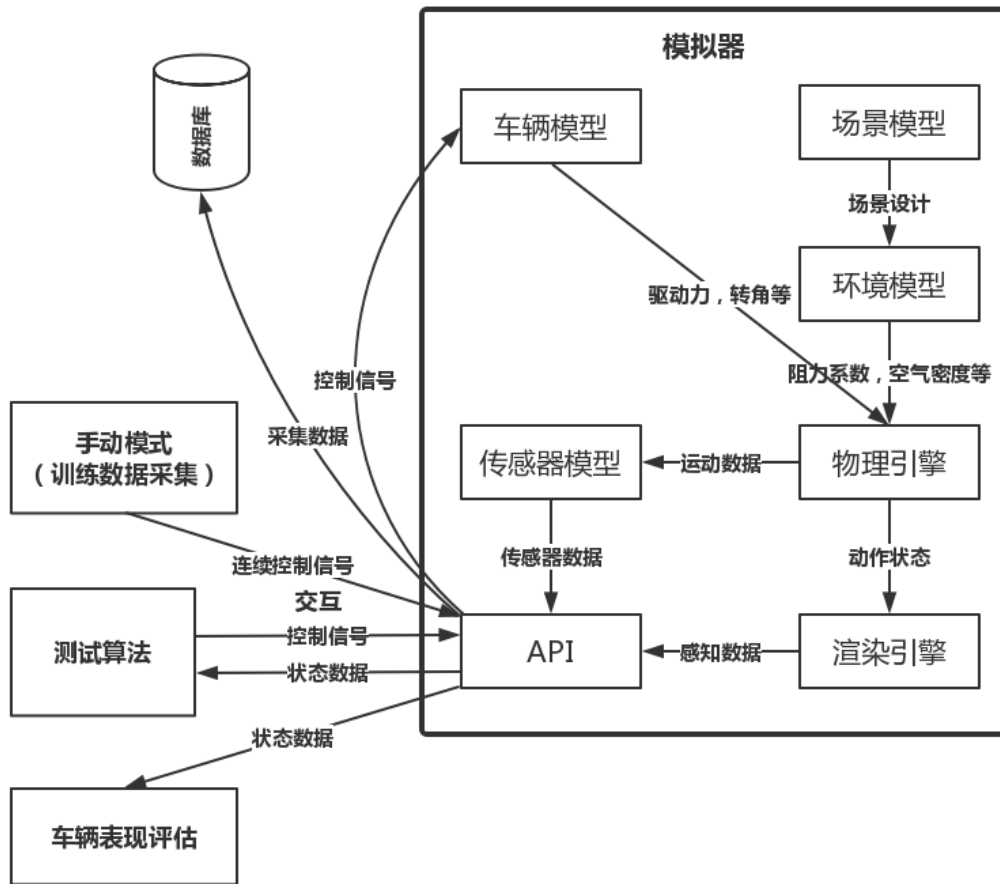


图 4.1 基于合成数据的场景模拟器开发结构示意图

4.1 基于合成数据的场景模拟器开发（汤，商，傅，甘）

基于合成数据的场景模拟器开发主要研究基于合成数据对环境、感知以及车辆的模拟技术，此技术下的模拟器主要用于控制与规划算法的初步开发中。模拟器接收外部算法产生的控制指令作为输入，渲染新的驾驶状态，提供场景模拟的运动学以及感知信息作为输出，从而完成模拟器与外部的交互。

基于合成数据场景的模拟器开发的结构示意图如图 4.1 所示，包括手动模式（训练数据采集）模块、测试算法模块、车辆表现评估模块以及右侧的场景模拟器内部结构等多个模块。

场景模拟器支持手动操作模式，用以生成训练数据。在手动模式中，通过连续的输入控制信号，手动驾驶记录行驶行为，在模拟环境中渲染出行驶数据，并用记录的运动信息和图像信息训练机器学习模型。**算法测试模块**用来对机器学习算法模型进行自动驾驶或测试。在自动驾驶模式中，**使用算法与模拟器实时的交互**

互控制汽车，算法输入车辆控制信号，得到模拟器输出的行驶信息产生新的控制信号。评估模块通过判断汽车是否为安全行驶状态检测一个机器学习模型的性能。

场景模拟器基于 Unity3D、Unreal Engine 等物理引擎搭建，包括场景设计模型、环境模型、车辆模型、物理引擎、传感器模型、渲染引擎以及模拟场景 API 接口等部分。场景模型中，基于不同的路型、不同的障碍物类型、不同的道路规划、不同的红绿灯信号等完成对典型/复杂的仿真场景的搭建。车辆模型中编辑汽车基础属性信息，并且对每一时刻接收的控制信号进行理解，解释为模拟场景中相应的物理信息。环境模型计算出生成场景的物理信息（如空气密度、场景地面摩擦系数等），与车辆模型的输出信息（如车体重量、转角、驱动力等）一同输入物理引擎进行运动信息的计算。物理引擎对模拟器中物理信息处理之后，渲染引擎将下一时刻场景中的动态渲染成画面，传感器模型接收处理后的运动学信息。最后渲染引擎将感知数据（如图像、画面等）、传感器模型将传感器数据（如车的位置、事实速度、是否在路口等）汇总到 API 接口，由 API 接口完成和模拟器外界的信息/控制交互。API 提供对模拟器内部传感器数据和感知信息的获取途径，同时也接受来自模拟器外部的控制信号，并将这些控制信号交由车辆模型进行解释。

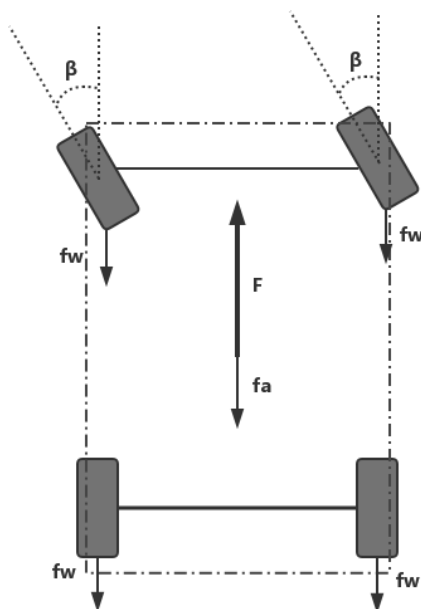


图 4.1.1 汽车运动模型示意图

4.1.1 车辆模型（汤）

模拟环境中的车辆模型遵循真实车辆的基本构造原理，提供接口以编辑

车辆基本属性配置信息，如汽车的长、宽、高等车体信息，车体中心（重心）位置以及车体距离中心点前、后、左、右的距离等位置信息。车辆模型还包括一些如质量，惯性，直线和角阻力系数，摩擦和转向系数的参数，从而物理引擎可以计算出车体的动态情况。

在模拟场景中，仿真汽车的运动情况，主要与汽车的驱动力和转向两个方面有关，前者控制子车辆行驶行进方向（纵向）的动态，后者描述子汽车在横向的变化情况。形式化地表示，模拟环境接在时刻 t 收到（来自算法的）一系列交互控制指令信号，如 $\{a^t, \beta^t, \dots\}$ ，表示模拟车辆的加速度信号、转角（转向）信号等。在模拟环境中，加速度信号 a^t 对应于汽车的进行方向的加速度，产生于车体的驱动力（负的加速代表刹车等驾驶动作施加于车体的阻力）；转角信号 β^t 映射车体行进方向的变化。

真实世界中的汽车动力主要来源于发动机燃烧汽油带动活塞上下做功，经由变速箱等传动装置，将扭矩放大到车轮，从而驱动汽车前进。汽车的发动机输出功率与车子的阻力功率始终相等，发动机的输出功率 \times 机械效率 = 行驶阻力（滚阻 + 风阻 + 坡度阻力 + 加速阻力等） \times 车速：

$$P_e \eta_T = F_f \mu_a$$

发动机输出功率是扭矩 \times 曲轴角速度，也就是曲轴每秒所做的功 $P_e = T\omega$ ，加速度可以视为来自于引擎的扭矩经由变速箱放大之后到达轮胎的扭力（驱动力） $F_{pow} = T'/r_{tile}$ ，即作用于轮胎的扭矩 T' 除以轮胎半径 r_{tile} 。

模拟场景中，汽车接受来自外界输入的控制信号加速度 a^t ，基于车辆本身的属性参数如质量 m ，得到驱动/制动车体动态的物理信息，行进方向合力 F ，同时合力 F 又应为驱动力和行驶阻力的共同作用：

$$F = ma^t \quad F = F_{pow} - F_f$$

转向描述了车体行进的方向，通过改变汽车的轮胎的转角完成横向的移动和曲线的行驶。模拟器输入的转向信号 β^t ，直接映射模拟汽车的转角变化。

通过对具体输入控制信号所产生的车体的驱动力和转向的描述，使得虚拟环境中的车辆能够模拟真实车辆的一些驾驶行为。这些物理信息连同车体本身的一些信息如车体重量 m 、车体框架中的横截面积 A 等，由物理引擎用来计算车体在模拟环境中受到的线性和角阻力等动态情况。

4.1.2 场景模型（商）

在无人驾驶模拟器开发中，需要首先进行仿真场景模型的建立，用来模拟

真实的现实行车环境。需要建立的场景模型主要包括：地形地貌等地理环境（如道路、房屋、障碍物等），道路模型（路段、栏杆、交通标志等），动态场景（道路上的车辆、行人等），及自然环境（如风向风力等）。

由于仿真环境场景中不同的模型所需要达到的精细程度各不相同，在建立场景模型之前，首先应对整体场景模型有全局考虑。对场景模型应该根据系统的要求进行合理的划分，具有一定的层次结构。不同的场景模型模块之间需要相互独立，以保证操纵动态模型的时候不会影响到静态模型，动态模型之间、静态模型之间也不会发生互相干涉。最后使用先进的建模技术将场景建模实现逼真的仿真效果，并且提供仿真的汽车行驶环境。仿真环境模拟器场景构成如图 4.1.2 所示。

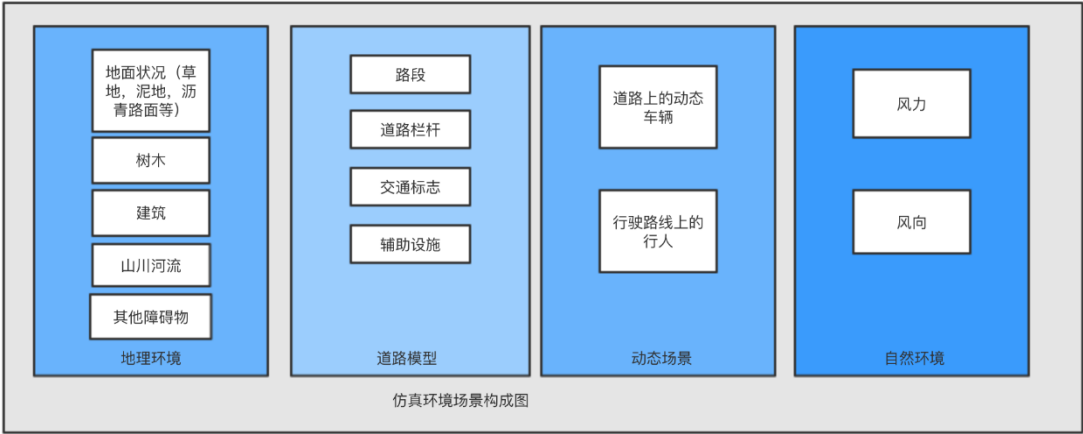


图 4.1.2 仿真环境模拟器场景构成图

4.1.3 环境模型（商）

车辆存在与环境中并且与环境进行交互。环境中的路况（路面平滑程度）、风力大小等自然环境会影响汽车与路面的摩擦力以及空气摩擦力等，进而影响车辆的行驶表现。同时真实环境中的房屋道路信息、红绿灯、路上的车辆行人等都应该在模拟环境中进行体现。

对于车辆行驶道路中的路面情况和自然环境情况，如路面摩擦系数，风的风向风速以及空气密度，会直接影响汽车的行驶行为。这部分主要通过输入的仿真环境场景，通过一定的监测和计算，得到需要的直接物理变量（地面的摩擦系数，风力，风向，空气密度）。

对路面摩擦系数的评估，通过监测道路材质，比对数据库中的地面材质和摩擦力系数对应表，这个对应表是预先通过收集数据存放在数据库中的，比如沥青道路的摩擦系数为 μ_1 ，草地的摩擦系数为 μ_2 ，沙漠地面的摩擦系数为

μ_3 等等。

对于风的评估，根据模拟场景得到风的仿真设置，如风速 x_i m/s，风向角度 θ_i 。用角度表示风向，是把圆周分成 360 度，比如北风(N)是 0 度(即 360 度)，东风(E)是 90 度，南风(S)是 180 度，西风(W)是 270 度，将其余的风向都用角度 θ_i 来表示出来。空气密度 ρ 根据场景模拟设置得到。

将这部分通过对路面和风的监测得到的相关数据，传到物理引擎部分再进行进一步模拟计算。流程图如图 4.1.3 所示。

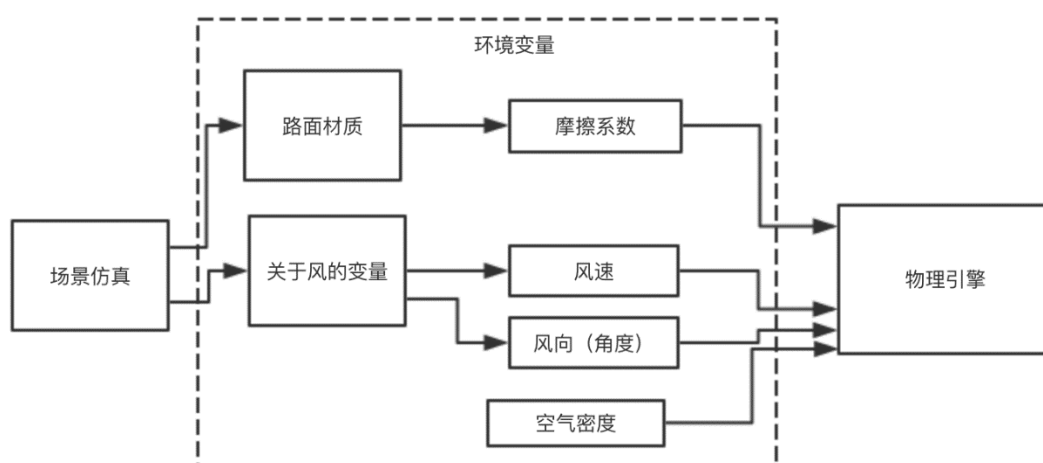


图 4.1.3 环境模拟流程图

4.1.4 物理引擎（傳）

动力学中使用四种属性描述车体的动力学状态：位置、方向、线速度和线加速度。物理引擎的作用是在给定物体作用力、扭矩和环境参数的情况下，计算出该物体的下一个动力学状态。物理引擎接收车辆模型和场景模型传输的多种参数，并将计算得到的物体动力学状态传输到渲染引擎进行渲染。构造能够以高频率（1000Hz）进行循环更新的物理引擎，可以有效地满足实时模拟的情景。重新设计的物理引擎避免了普通引擎带来的复杂性，也可以实现对模拟器性能的控制，最大程度满足模拟的需求。

（1）阻力：

由于汽车在路面上行驶，受到的阻力对车体运动有明显地影响。车体受到的阻力包括：轮胎对地面的滚动阻力和外部形状所影响的风阻两项构成。轮胎滚动阻力，

$$T_l = G_l \mu$$

其中， G_l 为法向载荷（主要来自车重）， μ 称为滚动阻力系数。法向载荷 G_l 由

车辆模型提供的车体重量获得，滚动阻力系数 μ 由场景模型提供。

车体在各种速度下所受的风阻，

$$F_2 = \frac{1}{2} C_d A \rho v^2$$

其中， C_d 为空气阻力系数， ρ 为空气密度(kg / m^3)， A 为迎风面积，即汽车行驶方向的投影面积(m^2)， v 为相对速度，在无风时即汽车的行驶速度(m / s)。空气阻力系数 C_d 和空气密度 ρ 由场景模型提供，迎风面积 A 由场景模型提供的风速、风向与汽车行驶方向夹角，以及车辆模型提供的车体尺寸数据获得。相对速度 v 由模拟器输入的加速度和当前速度获得。

因此，车体在运动过程中所受的阻力 F_d 可以表示如下，

$$F_d = \frac{1}{2} C_d A \rho v^2 + G_l \mu$$

(2) 加速度:

除阻力和扭矩之外，还需要考虑作用在车体 r_i 点上的作用力 F_i 。根据以下等式计算合力：

$$F_{net} = \sum_i F_i + F_d$$

其中， F_d 为阻力， F_i 由车辆模型提供。

使用牛顿第二定律获得线性加速度， $a = \frac{F_{net}}{m}$ 。

(3) 整合:

通过整合速率和初始位置 p_0 更新 $k+1$ 时刻车体的位置 p_{k+1} 。使用计算代价低且稳定的 Velocity Verlet 算法。定义如下，

$$v_{k+1} = v_k + \frac{a_k + a_{k+1}}{2} \cdot dt$$
$$p_{k+1} = p_k + v_k \cdot dt + \frac{1}{2} \cdot a_k \cdot dt^2$$

(4) 碰撞:

车体除行驶状态外，还有可能发生碰撞，需要对碰撞场景进行渲染。发生碰撞时的动力学状态较为复杂，因而使用 Unreal Engine 等 3D 引擎技术。Unreal Engine 提供了丰富的碰撞检测系统，针对不同类别的碰撞类型做以优化，~~可以选用这项技术。~~ 模拟器接收在渲染期间发生每次碰撞的冲击位置、法向冲击和穿透深度。物理引擎使用这些数据和库仑摩擦计算碰撞响应，进

而修改车体的动力学状态。

物理引擎是对真实物理世界的建模，反映汽车在当前场景下的运行方式。物理引擎将车辆模型和场景模型在物理场景中建模，输出车体下一时刻的动力学状态到渲染引擎，输出车体的运行数据到传感器模型。用于汽车下一状态的渲染和环境的感知。

4.1.5 传感器模型（甘）

这项工作的主要目标之一是通过收集数据来实现对显示系统的机器学习，因此，必须尽可能实际地对传感器采集数据进行建模。传感器可以模仿机器人如何感知环境，下文将介绍如何实现传感器，以便实际反映其特征。

（1）加速度计和陀螺仪

陀螺仪和加速度计构成惯性测量单元（IMU）的核心，可以利用加速度计和陀螺仪来测量车辆运动，可以通过加速度计来获得车辆转弯、加速或者制动时产生的冲击力，但是车辆是向前倾斜的，测量结果中就会有重力分量，倾斜传感器把重力方向当作参考方向。重力是一种加速度，并且不断变化。当进行倾斜测量时，我们只需要得到重力加速度；当进行车辆动力测量时，却又只想得到运动加速度。通过测量绕车辆重心的旋转，陀螺仪有助于纠正车辆向前倾斜带来的不利影响。可以结合角速度值和加速度值，计算出车辆动态运动时的精确数据。角速度和加速度的缺点可以相互弥补，就可以得到实时的加速度和角度的精确值。

（2）全球定位系统

模拟器中的 GPS 模型是一个简单的实现，具有基本的噪声分量和其他错误来源。GPS 传感器的关键问题之一是延迟。实现了 200 ms 的典型延迟（即目前由 GPS 指示的位置仅在 200 ms 前为真）。此外，与其他传感器相比，GPS 传感器通常在 20-50 ms 时具有更慢的更新速率。我们在模拟中实现这些关键方面，因为 GPS 通常用于纠正 IMU 中的漂移，因此 GPS 模型的不准确性可以级联到其他方面。此外，还模拟由 GPS 传感器本身计算的位置误差估计。具体来说，对 GPS 模块的模拟输出由传感器本身估计的水平和垂直位置误差的标准偏差。这两个输出的主要特点是随着 GPS 定位好随时间而衰减。使用简单的一阶低通滤波器来模拟该延迟。

（3） 车载传感器

车辆可以对周围的静态和动态环境进行 360 度的精确感知。在障碍物的检测识别中包括了传感器融合（整合多传感器的数据），物体检测（发现障碍物），物体分类（“障碍物是行人”），物体分割（“行人位于道路右侧”）和障碍物跟踪（“行人在向左移动”）。

在车辆的前后端装上传感器、雷达、摄像机等设备，能够自动探测出与前后车的距离，并于本车的制动、灯光等系统联动，当跟车距离低于安全距离时，系统会在零点几秒内启动，以强制拉大跟车距离。

4.1.6 渲染引擎（甘）

~~由于高级渲染和详细环境一直是模拟器环境的关键需求，所以选择了 Unreal Engine 4(UE4)作为渲染平台。UE4 的许多功能，使其成为一个有吸引力的选择，主要包括（1）图片逼真的渲染：UE4 带来了一些最突出的图形进步，包括实时的全球文化（2）开源和跨平台：UE4 是完全开源的，可以自由地用于仿真，可以在 Linux, Windows, iOS, Android 和 OS X 上无缝地运行（3）大型环境市场：大型市场，可以购买预先制作的精细环境，可以方便地从各种环境收集数据。~~

通过 UE4 将物理传感器、环境、车辆模型得到的动作状态在虚拟模拟器中渲染出来，把渲染的效果传给 API。

4.1.7 API 接口（甘）

API 接口是模拟器与外部算法、控制交互的模块，完成模拟场景内部信息和外部控制指令的交互。API 接口汇总来自传感器模型的感知器数据（运动学数据及判别数据等）和来自渲染引擎的描绘车辆、环境动态的感知数据，因此 API 的交互的模拟器信息包括如渲染后的图片、车辆的当前位置、车辆的当前速度、行车安全距离等等。模拟器通过 API 进行交互，交互环节有以下：

（1）外部算法（手动模式、测试算法等）获取 API 提供的模拟环境内部的各类信息，运动学信息如车辆当前位置，判别信息如红绿灯信号信息，感知信息如实时行车记录图像等。这些信息通过外部算法对 API 访问获取；

(2) 测试算法或手动模拟通过 API 接口输入控制信号，从而完成对车辆模型进行控制；

(3) API 接口把手动模式控制得到的模拟行驶信息，以及异常驾驶等特殊信息记录到数据库里。

4.1.8 车辆表现评估 (甘)

车辆表现评估模块用于对模拟场景中运行的汽车的行驶情况进行评估，对模拟车辆的行驶行为是否合理、安全、可靠进行评判。

主要考虑到 API 层的智能的场景判别系统，主要考虑以下 5 个判别标准：

- (1) 碰撞检测：判别车辆是否发生碰撞，碰撞距离为 0.1；
- (2) 闯红灯检测：检测是否闯红灯；
- (3) 限速检测：检测车速是否超过地图限速信息；
- (4) 在路检测：检测车辆是否行驶在路上；
- (5) 到达目的地检测等：是否到达目的地判断。