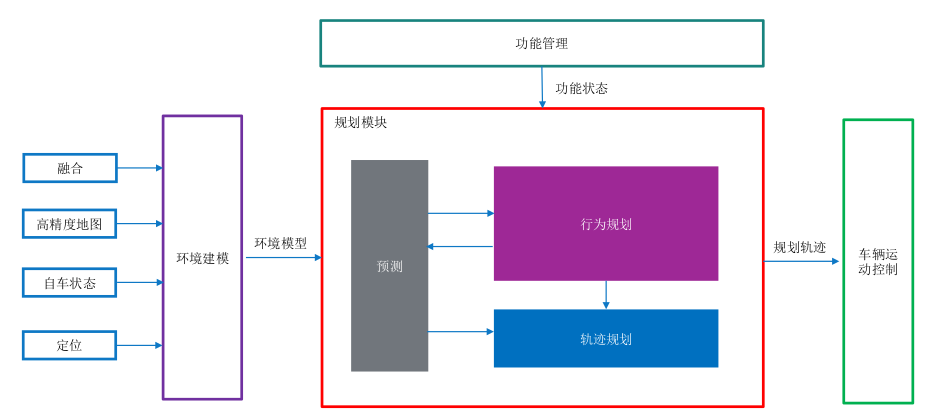
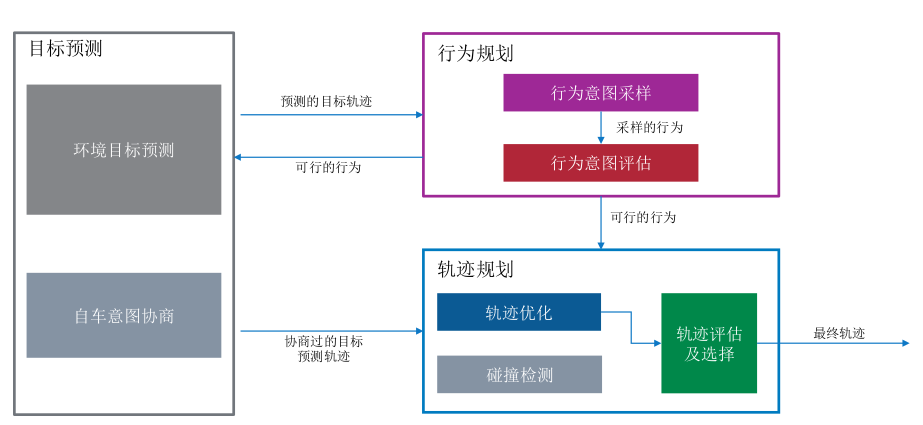
# 1 决策规划模块的整体架构

自动驾驶系统由感知模块、地图模块、定位和自车状态模块、规划模块、用户功能模块和运动控制模块组成。 感知模块提供动态障碍物、车道线等环境信息；定位模块提供自车绝对定位和车辆运动状态信息，如经纬度、车速；地图模块提供道路信息，如高速、高架的车道拓扑。 当高级导航自动驾驶功能启用时，用户功能模块将向规划模块发送功能激活请求。规划模块根据地图和当前自车定位规划出到达目的地的最佳路线，同时结合实时感知信息，评估最佳运动行为并规划出最佳轨迹，然后将其发送给运动控制模块执行运动。



规划模块内部架构如下图所示。该模块内部划分为环境建模、预测、行为规划和轨迹规划。在环境建模模块之后，规划过程按照分层方式进行。行为规划模块接收定义为导航的路线和当前环境模型，并与预测模块迭代，探索自车可能行为的树形结构。行为规划过程的结果是一组候选行为和相关的行为成本。然后，每个候选行为在并行运动规划实例中进行评估，产生该行为的最佳轨迹以及运动规划成本和可行性确定。行为候选的行为成本、运动规划成本和运动规划器的可行性信息然后被发送到决策模块，决策模块选择最佳动作。与所选行为相关的最佳轨迹然后被发送到控制和执行系统。



# 2 决策规划模块子系统

## 2.1 环境建模

### 2.1.1 障碍物感知

感知信息主要包括：

* 动态世界
* 静态世界
* 自由空间，表示车辆可通过的空间
* 可见网格地图，网格值表示是否可以被任何传感器检测到

### 2.1.2 道路拓扑感知

环境模型应提供以下道路拓扑信息：

* 全局路由信息
* 道路段列表
* 车道网络列表
* 道路边缘列表
* 停车线列表
* 人行横道列表
* 交通标志列表
* 红绿灯列表
* 停车位列表
* 参考线
* 畅通区域列表
* 十字路口

### 2.1.3 高精度地图

环境模型应提供搜索和提取地图信息的功能，包括：

* 车道网络
* 车道关系
* 人行横道
* 停车点（或停车线）

### 2.1.4 高精度定位

环境模型应提供以下定位信息：

* 时间戳（UTC）
* 在WGS84坐标系中的全局位置
* 在本地坐标系中的位置（x、y、z）及其方差
* 在自车坐标系中的线性速度（vx、vy、vz）及其方差
* 在自车坐标系中的线性加速度（ax、ay、az）及其方差
* 在本地坐标系中的欧拉角（roll、pitch、yaw）及其方差
* 在自车坐标系中的角速度（rollrate、pitchrate、yawrate）

## 2.2 预测模块

### 2.2.1 预测的概念

自动驾驶需要理解交通场景并做出决策，其预测目标是预测所有交通参与者的运动。然而，由于感知和定位的不确定性、复杂的车辆动态和车辆间的相互依赖性以及人类驾驶风格的变化，这是一个具有挑战性的任务。

自动驾驶需要对高速公路和城市道路的环境进行预测。对高速公路的处理主要涉及车辆和平行车道，而城市自动驾驶则更为复杂。一方面，城市道路既包括平行车道，又包括交叉口和人行横道。另一方面，城市道路上有多种类型的动态障碍物，包括车辆、自行车和行人，每种类型都有其独特的特点。例如，自行车通常靠近路边，并且速度范围低于车辆。此外，行人的外观（例如形状、大小）和运动模式各不相同。行人保护系统一直是一个活跃的研究课题，重点是行人检测和行为建模以避免碰撞。对于城市自动驾驶中不同类型的动态障碍物的预测仍然是一个开放的研究课题。预测交通参与者的意图和轨迹对于避免碰撞和优化自车行为规划至关重要。

现有的预测方法通常通过对以下两个方面的某些假设来解决这个问题。

**预测时间范围**：即可以预测多远的未来。短期预测侧重于预测诸如车辆加速度和转向速率等物理量，并根据运动学和动力学计算未来的轨迹，准确度通常在一秒以内。长期预测旨在预测未来几秒钟的情况。在这个长期的时间范围内，预测将会发生什么需要了解当前情况。因此，长期预测有两个任务：意图识别和轨迹预测。意图，也称为行为，如保持车道或变道，是对物理运动的高层抽象，假设未来的轨迹是由驾驶员打算做什么来决定的。例如，如果观察到一个障碍物与当前车道中心存在轻微偏差，意图识别将有助于确定它是开始变道还是仍然保持车道但具有意外的转向误差。虽然意图是离散的，但轨迹是连续的。轨迹预测是预测未来一段时间内的连续低级物理状态。

**交互建模**：独立预测是最简单的模型。它将每辆车都视为独立的，仅基于障碍物本身的属性来预测其意图或轨迹。然而，车辆与其他车辆共享道路，一个车辆的动作必然会影响其他车辆的动作。例如，如果前方车辆速度太慢，车辆可能会变换车道。虽然将每辆车独立对待简单而快速，但可靠的预测应考虑交通参与者之间的依赖关系，以更好地理解交通情况。由于计算复杂性，很少有现有工作明确建模了相互作用。一个更可行的方法是以不对称的方式建模依赖关系，即周围的交通影响目标车辆，而不是相反。在这个假设下，一个车辆的局部环境，包括与周围车辆的相对位置、速度等，常被用来推断未来的运动。

### 2.2.2 预测的方法

现有的运动预测方法可以分为三大类：基于物理的、基于机动的、以及基于交互的模型。这些方法各有优缺点，适用于不同的预测场景和时间范围。

**1. 基于物理的方法**

基于物理的模型主要依赖于运动的物理定律来预测车辆的运动。这些模型重点关注跟踪控制输入，如转向角和加速度，并可能考虑轮胎与路面之间的相互作用。尽管最简单的方法使用恒定的控制输入，如恒定的转向速率或加速度，但它们无法捕捉车辆运动的变化，并忽略了状态估计中的噪声。为了更准确地处理不确定性，一些方法使用卡尔曼滤波器或其变种，如扩展卡尔曼滤波器（EKF），来预测未来的轨迹。基于物理的方法适用于短期预测，因为它们只考虑低级的物理特性，例如转向和加速度，而忽略了高级机动或意图。

**2. 基于机动的方法**

基于机动的方法主要用于长期预测，因为它们对高级行为进行建模。这些方法通常将每个驾驶员视为独立的机动实体，并根据估计的机动或意图进行预测。其中一种方法是模式匹配，它将轨迹模式与预先学习的模式进行匹配，以预测车辆的行为。另一种方法是意图识别，通过机器学习技术对驾驶员的意图进行分类，并根据预测的意图来计算轨迹。虽然这些方法允许相对长期的预测，但随着抽象程度的提高，复杂性和计算成本也会增加。

**3. 基于交互的模型**

基于交互的模型考虑了车辆之间的依赖关系，并通过使用本地上下文信息来隐式地将交互纳入考虑。这些模型更好地理解交通情况，但由于交互受到多种因素的影响，包括车辆的运动学和动力学、交通规则等，因此它们的实现具有挑战性。一些方法假设交互的一个重要目标是避免碰撞，并根据这一假设来调整预测的轨迹。为了减少计算复杂度，一些方法采用不对称建模，即周围的交通影响目标车辆，而不是相反。这些方法使用本地上下文信息来推断车辆的意图和行为。

基于物理的方法简单且流行，但只适用于短期预测。基于机动的方法允许相对长期的预测，但随着抽象程度的提高，复杂性和计算成本也会增加。基于交互的模型可以更好地理解交通情况。

**意图识别模块**负责预测交通参与者可能会执行的行为，包括以下几种类型：

* 保持车道：交通参与者将继续沿着当前车道行驶，不进行车道变更。
* 向左变道：交通参与者打算从当前车道向左变更到相邻的左侧车道。
* 向右变道：交通参与者打算从当前车道向右变更到相邻的右侧车道。
* 左转：交通参与者打算在交叉口或转弯处向左转弯。
* 右转：交通参与者打算在交叉口或转弯处向右转弯。
* 掉头：交通参与者打算在允许的地方进行掉头操作。

**轨迹预测模块**会根据意图分类的结果，预测交通参与者未来8秒内的轨迹。轨迹的信息包括位置、速度和方向，以便在给定时间内更好地理解交通参与者的运动状态和行为意图。这种轨迹信息对于自动驾驶系统的决策和规划过程非常重要，因为它们提供了关于周围环境中其他车辆可能的动作的关键线索，从而帮助自动驾驶车辆做出适当的反应和行驶决策。

## 2.3 任务规划(Mission Module)

### 2.3.1任务规划的概念

在自动驾驶中，路线规划器扮演着关键的角色。它负责将车辆从起点导航到目的地，并在行驶过程中遵循交通规则、安全驾驶，并优化驾驶行为以提高行驶效率。

首先，路线规划器会基于车辆当前位置、目的地以及环境信息（如地图数据、障碍物、交通状况等）计算出一条最佳路径。这个路径考虑了不同的因素，比如最短路径、最快到达时间、安全性等，以确保车辆可以安全、高效地到达目的地。

其次，路线规划器会将这条路径转化为车辆可以执行的一系列动作，比如转向、加减速等。这些动作需要考虑车辆的动力学特性、道路规则、交通信号等因素，以保证车辆在执行过程中能够安全驾驶并遵守交通规则。

在实际行驶过程中，路线规划器会不断地更新路径和动作，以应对环境的变化。例如，当出现交通堵塞、道路封闭或者突发事件时，路线规划器会重新计算路径，并相应地调整车辆的行驶策略，以选择更合适的路线或采取避障措施。

### 2.3.2任务规划的方法

任务规划模块是为后续的决策和动作等子模块提供可用的道路路径和所选的候选项（通过路径规划或人工命令）。由本地路径生成器和车道变换决策器这两部分组成。

本地路径生成器：利用给定的定位和语义地图框架生成可用的道路路径（定义为本地路径）。

车道变换决策器：选择最佳的候选项，以确保自动驾驶车辆能够取得最佳进展，遵循路径规划并超越缓慢的车辆。

## 2.4 行为规划(Decision Module)

### 2.4.1所有动态驾驶任务

#### 定速巡航

在自动车道保持工作模式下，定速巡航需满足以下功能要求：

1）支持直道、弯道、上下坡；

2）直道、弯道情况下，速度控制精度不得超过±1km/h；

3）上下坡情况下，速度控制精度不得超过±2km/h

4）当安全员调整目标巡航速度后，系统应控制车速平稳过渡到安

全员设定车速；

5）激活自动车道保持时，初始巡航速度判定逻辑：

若当前速度≤20km/h，则设置 20km/h 作为目标巡航速度；

若当前速度＞20km/h，则设置当前速度作为目标巡航速度；

1. 允许安全员设定的目标巡航速度范围为【20km/h，80km/h】

#### 智能限速控制

——基于限速标识的智能限速：

1）需识别 GB 5768 法规要求的各类限速标识；

2）限速信息来源需支持：视觉、导航、高精地图

3）限速信息设定为目标巡航速度需支持：安全员确认、自动更新

安全员确认：安全员可通过操作将限速设置为目标巡航速度；

自动更新：系统自动将限速设置为目标巡航速度，跟随限速更新

4）允许安全员基于限速作偏移量调整，调整方式分为：固定和百

分比

固定：允许安全员在-10km/h 至+10km/h 之间进行调整

百分比：允许安全员在-20%至 20%之间进行调整

5）激活自动车道保持时，初始巡航速度为当前速度，仅当限速信

息发生变化时，系统将自动同步限速+偏移值作为目标巡航速度

#### 基于天气的智能限速

1）系统需识别雨、雪、雾霾等天气

2）系统根据识别到的天气，进行限速提醒

#### 基于道路结构的智能限速

1）系统需识别道路曲率，进入弯道前，提前减速至合适车速，完

成舒适入弯

2）过弯期间，可根据道路曲率变化实时调整车速，车速变化应保

证舒适性，如在大曲率车道行驶

3）出弯之后，系统需舒适加速恢复至原有设定速度

4）弯道对应最大过弯加速度不得大于 3m/s 2 ，加速度的平均变化

率不得大于 5m/s 3

#### 车道居中控制

车道居中控制需满足以下功能要求：

1）支持直道、弯道、上下坡等道路结构；

2）系统需识别满足 GB 5768 规定的各类车道标线；

3）当一侧车道线消失，车辆与有车道线侧保持一定距离（可标定）

行驶；

1. 当两侧车道线短时消失时仍可以维持车辆当前方向行驶；
2. Passing through truck driving on neighbor lane; 有相邻车道有大车，自车应远离大货车在自车道内行驶。

5）对于横向控制精度的要求：

直道：控制精度保持在±15cm 之间；

弯道：控制精度保持在±25cm 之间；

6）系统在横向控制过程中，需保证横向运动的舒适性，横向加速

度最大不得超过 3m/s 2 ；

横向加速度的平均变化率不得大于 5m/s 3 ；

1. 若车道线长时间缺失或模糊导致无法识别时，横向功能可退出；

#### 本车道内刹停

本车道内刹停需满足以下功能要求：

1）系统需控制车辆功能运行范围内舒适性减速；

2）系统应控制车辆于碰撞目标前 2m[可标定]处刹停；

#### 车道内避障

车道内避障需满足以下要求：

1）前方目标干扰自车行驶轨迹，但仍存在可行驶区间，系统执行

车道内横向偏移避障,如邻近车道有停放车辆、车辆停放在外面、打开车门的停放车辆的情况；

2）待完成避障后，恢复至车道中间行驶，同时恢复之前设定巡航

速度；

#### 自动跟行

自动跟行功能需满足以下要求：

1）跟行时距分为 5 挡，分别为：1s，1.4s，1.8s，2.2s，2.6s；

2）允许安全员调整跟车时距，首次默认为 3 挡，后续记忆；

3）系统支持直道、弯道、上下坡跟行，且跟行性能不应受道路结

构的影响；

4）前方车辆一旦侵占本车道即作为跟行目标；

5）前车匀速行驶时，自车根据安全员设定的车距保持恒定车距行

驶；

6）前车出现急减速时，系统应及时感知前车意图，及时减速，保

证安全；

7）前车出现急加速时，自车不应跟随前车急加速，维持自车舒适

加速；

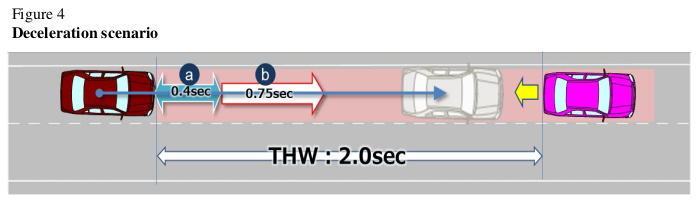
8）在跟行状态下，自车横向维持在车道中心行驶，纵向依据安全

员设定时距行驶，不应随前车的横向移动自车发生偏移；

7）当前车判定为跟随目标时，应颜色突出显示，易于安全员区分；

8）最小跟行距离不得低于 2m；

9）系统不得将临侧车道目标作为跟行目标；



#### 智慧跟车

智慧跟车需满足以下功能要求：

1）前车是大车，增加跟车距离，大型车辆识别包括但不限于：公交车、货车、卡车

2）调整跟车时距=设定跟车时距\*1.2；

#### 停走功能

停走功能需满足以下功能要求：

1）自车跟停后，开始计时，若跟停时间＜5min，前车起步，自车

自动跟随起步；1min＜跟停时间＜5min，车辆起步时，需给予安全员

起步提醒；

2）若跟停时间≥5min，则通过 HMI 告知安全员自车处于 Hold 状

态，需要安全员的确认【踩油门踏板】，才能继续自动起步；

3）若跟停时间在 0-10min 内，收到安全员的确认【踩油门踏板】，

车辆再次恢复到静止后，重新计时；

4）若跟停时间＞10min，自动车道保持功能自动退出，

拉起 EPB；

1. 前车起步，自车跟随起步的延迟不得大于 2s【可标定】；

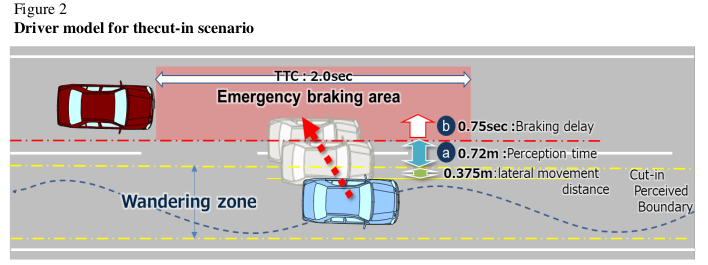
#### 切入 Cut-in

切入 Cut-in 需满足以下功能要求：

1）前方出现目标切入后，系统应及时完成跟车目标切换，控制车

辆速度，保持与前车安全距离；当超出系统控车能力时，需及时发送

接管提醒，通知安全员；



#### 切出 Cut-out

切出 cut-out 需满足以下功能要求：

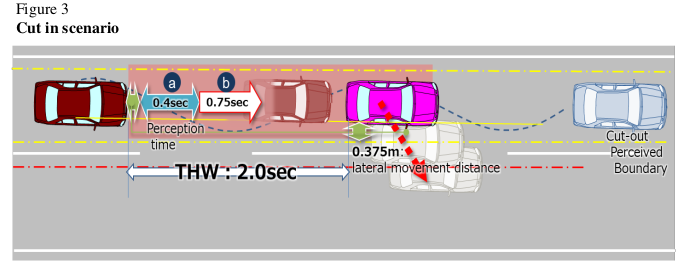
1）前车切出后，若前方无目标物，自车进入巡航状态；

2）前车切出后，若前方出现新的跟车目标，应及时进行目标切换

且与之保持设定跟车时距；

3）前车切出后，若前方出现静止目标物，则根据当时情况，判断

是车道内刹停还是车道内避障。



### 2.4.2 行为规划的方法

自动驾驶技术的发展正日益成为汽车行业的焦点。其中，行为规划是自动驾驶系统中至关重要的一部分，它决定了车辆如何在不同的交通环境中行驶，以确保安全、高效和舒适的驾驶体验。

行为规划可分为两种主要方法：基于策略的方法（Policy Based Approaches）和搜索与规划的方法（Search and Planning Approaches）。

1. 基于策略的方法（Policy Based Approaches）：这种方法通过直接学习或构建出行为策略（policy），即从给定的状态选择最优动作的函数。策略可以是确定性的，也可以是概率性的。在基于策略的方法中，主要的挑战是设计或学习一个高效且鲁棒的策略，使得系统能够在不同的环境下做出合适的决策。典型的算法包括深度强化学习中的策略梯度方法（Policy Gradient Methods）和Actor-Critic方法。

2. 搜索与规划的方法（Search and Planning Approaches）：这种方法通过搜索和规划技术来寻找最优的行动序列。通常会将问题建模为搜索问题，利用搜索算法（如A\*、蒙特卡罗树搜索等）或规划算法（如经典的规划算法、动态规划等）来寻找最优策略。搜索与规划的方法通常需要对问题的状态空间和动作空间进行建模，并需要解决状态空间爆炸和搜索效率等问题。典型的算法包括A\*算法、Dijkstra算法、蒙特卡罗树搜索（MCTS）等。

这两种方法在自动驾驶系统中都有广泛的应用。基于策略的方法通常用于端到端的学习或者在线学习，适用于连续决策任务。而搜索与规划的方法则更适用于对环境建模较为精确、需要长期规划和考虑的场景，例如路径规划、场景分析等。在实际应用中，这两种方法也可以相互结合，以充分利用各自的优势来解决复杂的行为规划问题。

#### 2.4.2.1. 状态机

基于状态机的设计流程如下：

1.定义状态集合： 首先，定义一组状态，这些状态描述了系统可能处于的不同情况或行为模式。每个状态代表系统在某一时刻的特定状态或行为。

2.确定状态转移条件： 对于每个状态，确定状态转移条件。这些条件描述了系统何时从一个状态转移到另一个状态。状态转移条件可以基于输入信息、环境感知或系统内部状态的变化。

3. 设计状态转移规则： 根据状态和状态转移条件，设计状态转移规则。这些规则定义了系统如何根据当前状态和条件选择下一个状态。

4. 定义行为策略： 对于每个状态，定义相应的行为策略。这些策略描述了系统在特定状态下应该采取的行动或决策。

5. 实现状态机控制器： 将状态集合、状态转移条件、状态转移规则和行为策略组合在一起，实现状态机控制器。该控制器根据当前状态和输入信息，选择适当的行为并管理状态转移。

6. 执行状态机控制： 在实际应用中，将状态机控制器与系统集成，执行行为规划任务。状态机控制器根据系统状态和环境条件，动态调整系统的行为，以达到规划的目标。

#### 2.4.2.2. 蒙特卡洛树搜索

蒙特卡洛树搜索（Monte Carlo Tree Search，MCTS）是一种用于决策问题的搜索算法，常应用于行为规划中。下面是MCTS算法在行为规划中的应用过程：

1. 定义问题和状态空间： 首先，将问题形式化为一个状态空间，在该状态空间中，每个状态代表系统在某一时刻的状态，而每个动作代表系统可能采取的行为。

2. 构建搜索树：从起始状态开始，利用MCTS算法逐步构建搜索树。在每一步中，通过对当前状态进行模拟并使用蒙特卡洛方法进行采样，从而选择最有价值的动作来扩展搜索树。

3. 模拟和选择：在搜索树中，通过一系列模拟操作来评估候选动作的价值。这些模拟可以是随机的或者基于某种策略，以获取对动作的预期效果。然后，根据这些评估结果选择一个动作来扩展搜索树。

4. 扩展和更新：选择的动作会导致系统转移到一个新的状态。将该状态添加到搜索树中，并更新搜索树的节点信息，包括节点访问次数和累计奖励值。

5. 迭代搜索过程：重复执行模拟、选择、扩展和更新步骤，直到达到一定的搜索时间或者搜索次数。这样可以逐步完善搜索树，提高对系统状态空间的探索。

选择最优动作： 在搜索树构建完成后，根据搜索结果和节点信息，选择具有最高价值的动作作为最终的行为规划结果。

#### 2.4.2.3. POMDP

部分可观察马尔可夫决策过程（Partially Observable Markov Decision Process，POMDP）是一种用于解决决策问题的框架，特别适用于行为规划中需要考虑不完全观察和不确定性的情况。下面是POMDP算法在行为规划中的应用过程：

1. 定义问题模型： 首先，将问题建模为一个POMDP。定义状态空间、观察空间、动作空间和奖励函数。状态空间包括系统可能处于的所有状态，而观察空间包括能够观察到的所有信息。动作空间包括系统可采取的所有行为。奖励函数定义了在每个状态下采取动作的奖励或成本。

2. 确定观察模型和转移模型：定义观察模型和转移模型，用于描述系统状态和观察之间的关系，以及动作对状态转移的影响。观察模型表示给定状态下观察的概率分布，转移模型表示给定动作后状态转移的概率分布。

3. 求解策略：通过求解POMDP，得到最优策略。由于POMDP通常是一个高度复杂的问题，难以直接求解，因此通常采用近似方法。常见的方法包括值迭代、策略迭代、蒙特卡洛方法等。

4. 执行策略： 将得到的最优策略应用到实际系统中。根据当前观察和状态，选择最优的行动，并根据策略执行相应的动作。

POMDP算法在行为规划中的应用可以帮助系统在面对不完全观察和不确定性的情况下，制定出最优的决策策略。通过考虑观察和状态之间的关系，以及未来可能的不确定性，POMDP能够更好地适应复杂的实际环境，提高系统的决策性能。

## 2.5 **运动规划 (Motion Module)**

### 2.5.1运动规划的概念

运动规划是指针对L3自动驾驶车辆在结构化车道驾驶场景中的技术，其发展已经趋于一致。这些方法针对自动驾驶车辆的独特特征进行了优化，主要包括以下几点：

1. 动力学约束的重要性： 动力学约束（如有限的加速度、颠簸和转向率）是规划问题的关键因素之一。

2. 舒适性是首要考虑因素：与其他机器人系统不同，自动驾驶汽车必须优先考虑乘客的舒适性。因此，最小化纵向和横向颠簸至关重要。

3. 平滑性和一致性是必需的：自动驾驶汽车的行为应该是平滑且一致的，以保证乘客的舒适和安全，同时确保与其他车辆的安全、可预测的交互。一致、可预测的驾驶风格对其他人类驾驶员准确预测自动驾驶汽车的未来运动至关重要。

4. 高度结构化的环境：车道驾驶任务中的固有结构降低了自动驾驶车辆的可能配置空间，使得可以使用小型、定义良好的搜索区域和离散采样等方法。

5. 动态重新规划是必不可少的：自动驾驶车辆必须与其他车辆并行行驶，并与其协作，但其他车辆的意图是未知的。因此，自动驾驶汽车必须能够快速重新规划其计划的轨迹，以适应动态环境中频繁变化的情况。

这些特点共同构成了针对自动驾驶车辆的运动规划技术的基础，使其能够在复杂的城市环境中实现高效、安全、舒适的驾驶。

### 2.5.2运动规划的方法

#### 2.5.2.1 路径和速度解耦规划

路径和速度解耦规划是一种在路径跟踪问题中常用的方法，特别是在机器人运动控制和自动驾驶领域。这种方法的核心思想是将路径规划和速度控制分开处理，从而简化问题并提高控制效率。

在路径和速度解耦规划中，路径规划负责生成机器人或车辆应该遵循的理想轨迹，而速度控制则负责沿着这条路径以适当的速度移动。这样做的好处是可以分别优化路径和速度控制，使系统更加灵活和鲁棒。

#### 2.5.2.2 基于采样的运动规划

基于采样的运动规划方法，其流程可以分为以下几个步骤：

1. 将车辆当前位姿信息转换到Frenet坐标系下，确定车辆在该坐标系中的初始状态。同时，根据当前速度计算前瞻距离，并据此确定前瞻点的位置，以获取车辆在前瞻点处的目标状态。

2. 对轨迹状态进行采样，包括轨迹运行时间 t、目标速度 v、以及到参考线的横向位移 d 。这三个规划参数组成了采样状态。

3. 构建横向位移 d(s) 和纵向位移 s(t) 的多项式规划函数。通过这些规划函数，可以获得参考线在Frenet坐标系下的轨迹点。最后，进行时间插值，将轨迹点从Frenet坐标系转换到笛卡尔坐标系，得到物理世界中的采样轨迹。由于横向和纵向位移都是通过高次多项式插值得到的，因此得到的笛卡尔坐标系下的轨迹也是光滑的。

4. 对采样轨迹进行碰撞检测、曲率约束和最优轨迹打分。采样轨迹是一系列满足速度约束的光滑轨迹，但还需要满足无碰撞和车辆运动学曲率约束等强制约束，以及远离障碍物和靠近参考线等成本约束。轨迹打分的目的是为了获得一条最优的、满足各种约束条件的、无碰撞的光滑轨迹。

#### 2.5.2.3碰撞检测方法

三种碰撞检测算法——GJK、AABB和OBB

1. GJK (Gilbert-Johnson-Keerthi) 算法：

- GJK算法是一种用于检测两个凸形状是否相交的算法。

- 它通过迭代寻找最近点对来判断两个凸形状是否相交。

- GJK算法的核心是通过迭代寻找包络球（Minkowski差集的包围球）来逼近凸形状的交点，直到找到交点或确定两个凸形状不相交。

2. AABB (Axis-Aligned Bounding Box) 算法：

- AABB算法是一种用于快速检测碰撞的简单而有效的算法。

- 它将物体用轴对齐的最小边界框（AABB）来表示，使得边框的轴与坐标轴对齐。

- AABB算法通过比较两个AABB是否相交来判断物体是否相交。虽然不如OBB精确，但计算简单且高效，特别适用于快速排除不可能的碰撞情况。

3. OBB (Oriented Bounding Box) 算法：

- OBB算法是一种用于检测物体碰撞的更精确的算法，相对于AABB，它允许包围盒的旋转。

- OBB算法通常由包围盒的中心点、方向向量和尺寸组成，这些信息可以描述盒子的方向和形状。

- OBB算法在需要更准确的碰撞检测时特别有用，因为它能够更好地适应物体的形状，但与之相关的计算复杂度相对较高。

在实际应用中，这些算法可能会结合使用，根据具体情况选择合适的算法来进行碰撞检测。例如，可以使用AABB快速排除大部分不可能的碰撞，然后再使用更精确的GJK或OBB算法来进行更细致的检测。

## **2.6运动控制(Control Module)**

### 2.6.1运动控制的概念

在自动驾驶车辆的软件堆栈中，运动规划生成描述车辆期望运动的轨迹。使车辆按照给定轨迹行驶的任务是跟踪控制器的任务，该控制器由纵向控制器和横向控制器组成。纵向控制器通过制动系统和动力传动系统对车辆进行控制，而横向控制器利用转向系统。

### 2.6.2运动控制的方法

#### 2.6.2.1 MPC控制

MPC在车辆运动控制的一般求解流程：

1. 问题建模：

- 系统建模：对车辆系统进行建模，包括车辆的动力学模型、约束条件以及性能指标等。

- 控制目标：明确控制的目标，例如跟踪一条参考轨迹。

2. 预测模型：

- 将车辆的动力学模型离散化，以便在有限的预测时段内进行预测。

- 建立状态空间模型，包括状态变量、控制输入、系统约束等。

3. 目标函数构建：

- 设计目标函数，通常由两部分组成：跟踪误差项和控制输入项。

- 跟踪误差项用于衡量车辆状态与参考轨迹之间的差异。

- 控制输入项用于衡量控制输入的大小和变化量，以保持控制的平滑性。

4. 约束条件添加：

- 添加约束条件，包括状态约束（如速度、加速度约束）、控制输入约束（如转向角、加速度约束）、系统动力学约束等。

5. 优化求解：

- 将MPC问题转化为一个优化问题，一般是二次规划（QP）问题。

- 使用优化求解算法（如内点法、梯度下降法等）求解优化问题，得到最优的控制输入序列。