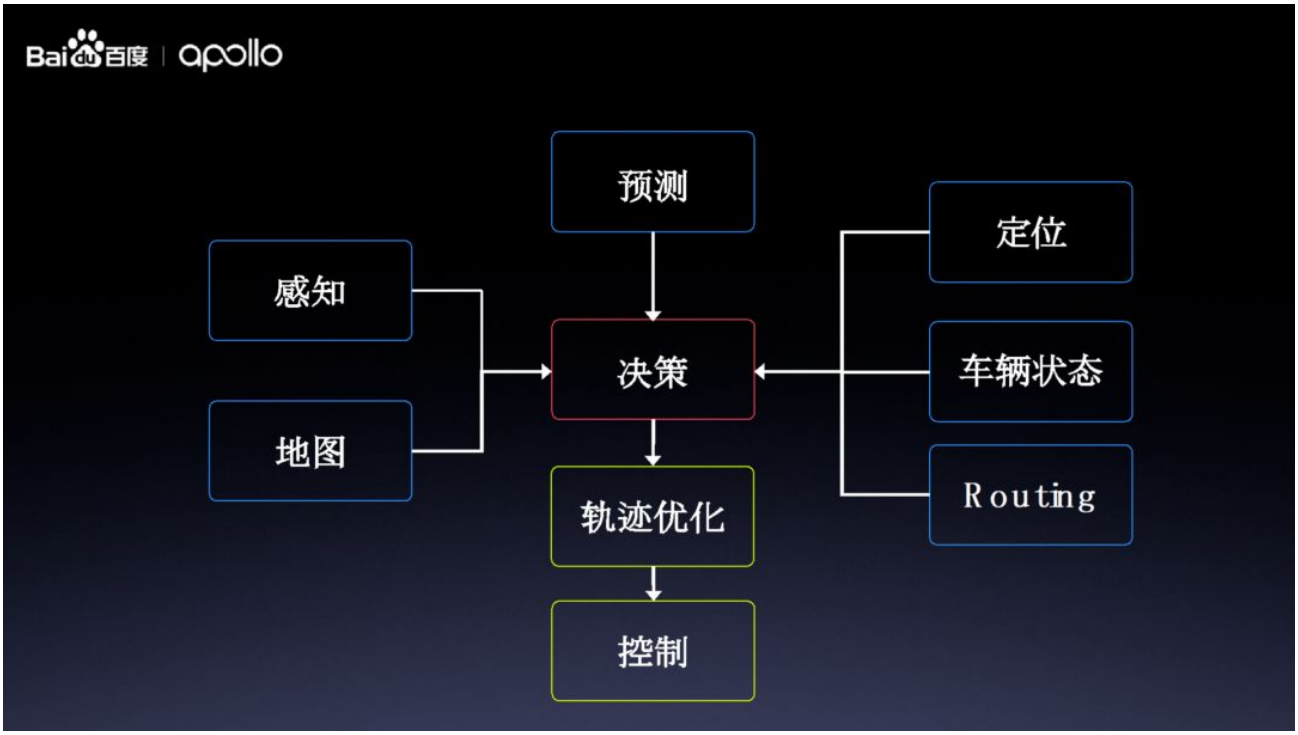
Apollo公开课丨Apollo决策技术分享

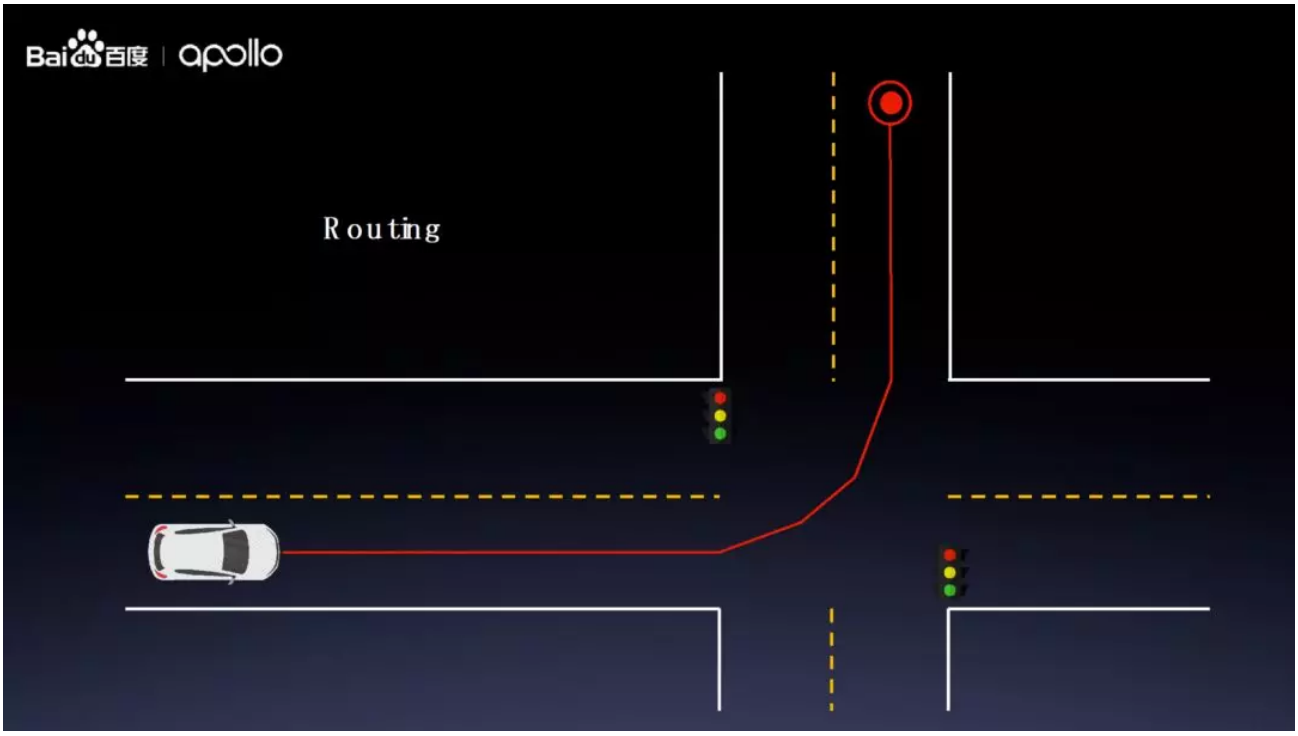
本次内容主要向大家介绍 Apollo 平台的决策技术。下图展示了决策在自动驾驶软件模块中的信息流。如图所示，决策的左侧是感知和地图两大模块，这两个模块是提供无人驾驶的环境信息，包括静态环境（主要来自地图）和动态环境（主要来自感知）。上方的预测模块用来预测动态障碍物的运动轨迹。右侧的定位、车辆状态模块提供自动驾驶车辆的位置信息，车辆本身状态（如速度、加速度等）；Routing模块提供目的地信息以及到达目的地的车道级别路线。所有这些信息都汇总到决策模块，决策模块根据所有信息来提供行车决策，产生的决策信息会发给下一步轨迹优化，轨迹优化会生成行车轨迹，包括速度和路径信息。控制模块根据行车轨迹控制车辆自动驾驶。

决策模块有两个特点，第一是相当于信息hub，它把所有信息集中起来；第二是经过决策模块传给轨迹优化之后，轨迹优化不用考虑上层所有复杂的信息，使抽象的问题变得相对简单，有助于替换新的算法。



总得来说，决策就是对无人车的行车进行各种限制，主要分为三类：第一类是交规限制，保证无人车遵守交规，比如信号灯、人行横道等交规；第二类是路径限制，例如，不能跨两条车道行车，不能驶出道路边界；第三类是速度限制，无人车不能超过道路限速，需要低速通过限速带。

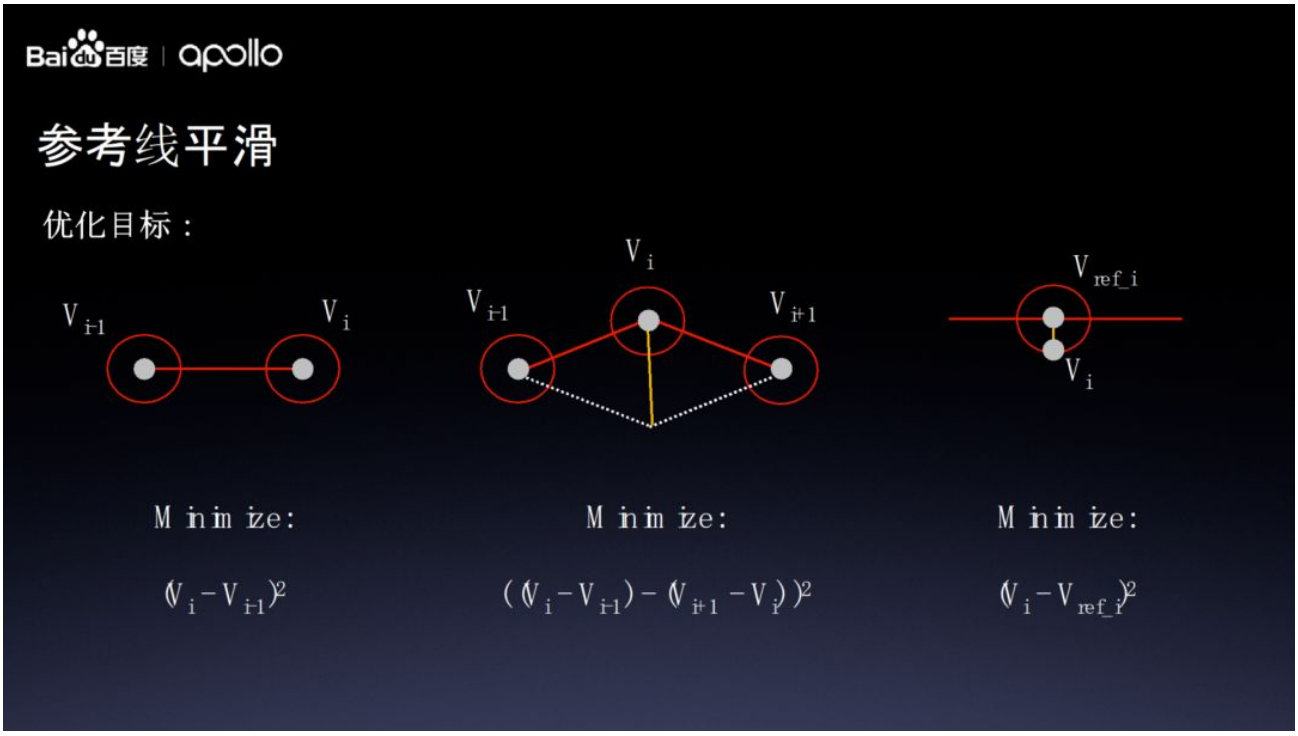
下面通过一个非常简单的例子讲解如何实现这三种限制。



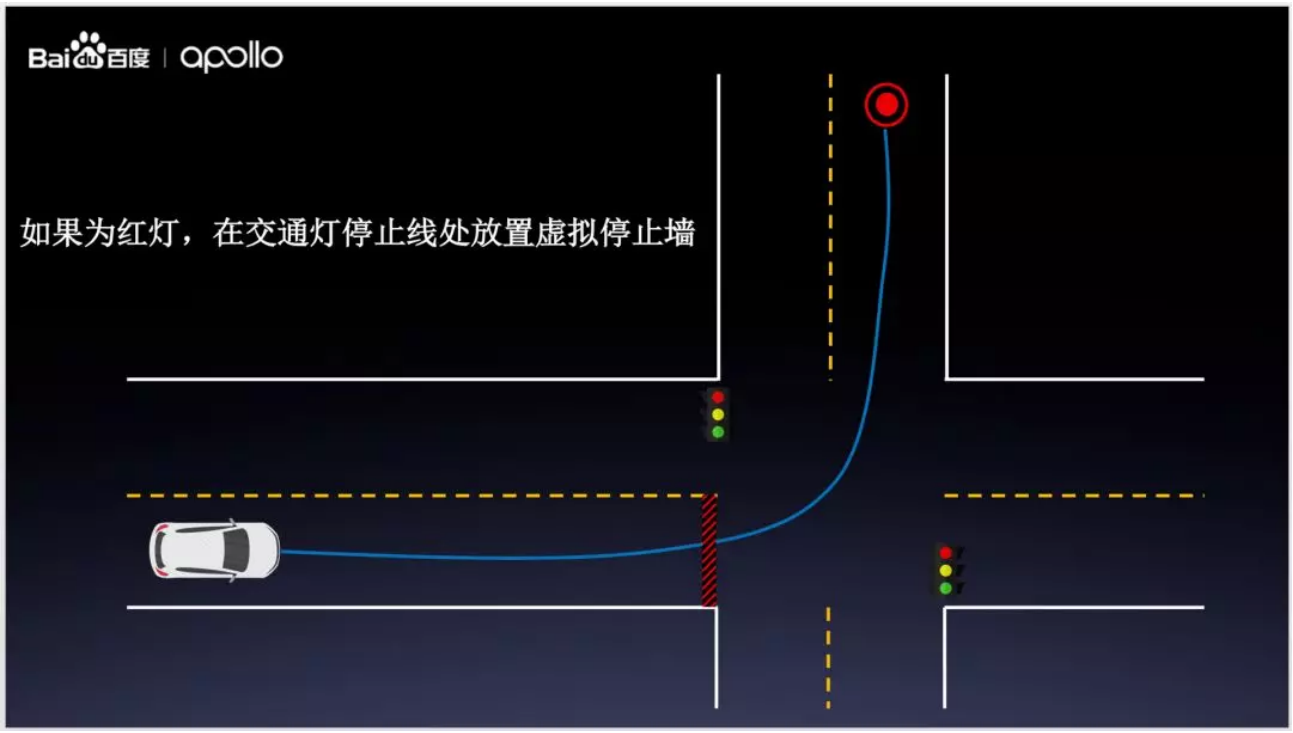
如上图所示，左下角白色车是自动驾驶汽车，目的地是右上角红色点。首先通过Routing，拿到规划线路，整个过程和在手机上使用百度地图导航差不多，唯一区别在于，Routing线路的精度是车道线级别而不是道路级别。有Routing线路之后，首先要对参考线做平滑处理。

Apollo5.0发布了新的参考线平滑算法，首先沿着Routing线每0.25米采样一个点，这个密度非常高，使得优化轨迹更细致。每个点用向量gx、y表示。对每一个采样点都可以根据高精地图定位信息和车道边界进行调整，得到一个非常光滑的曲线。

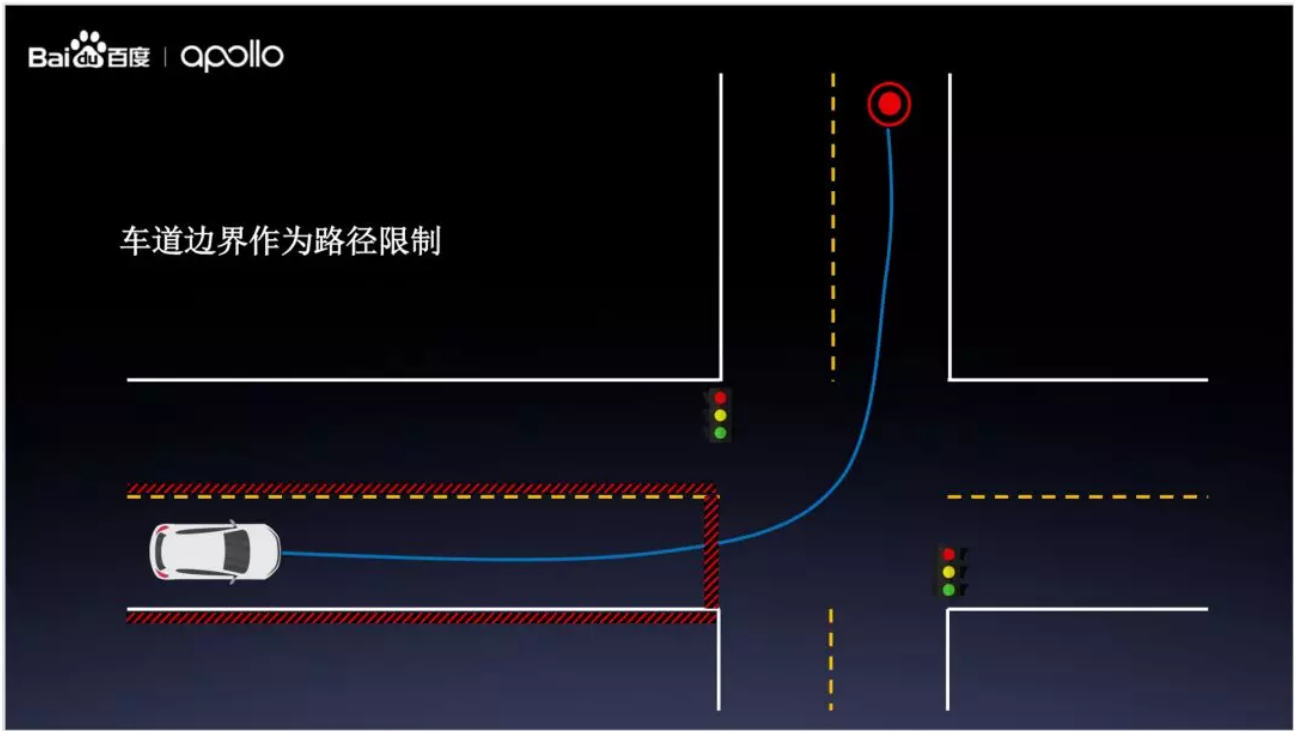
参考线平滑的优化目标有三个，第一个是每相邻两个点的距离要尽量近；第二是每三个相邻采样点的夹角要足够小；第三是优化目标尽量不要远离于初始点位置，调整尽量小。



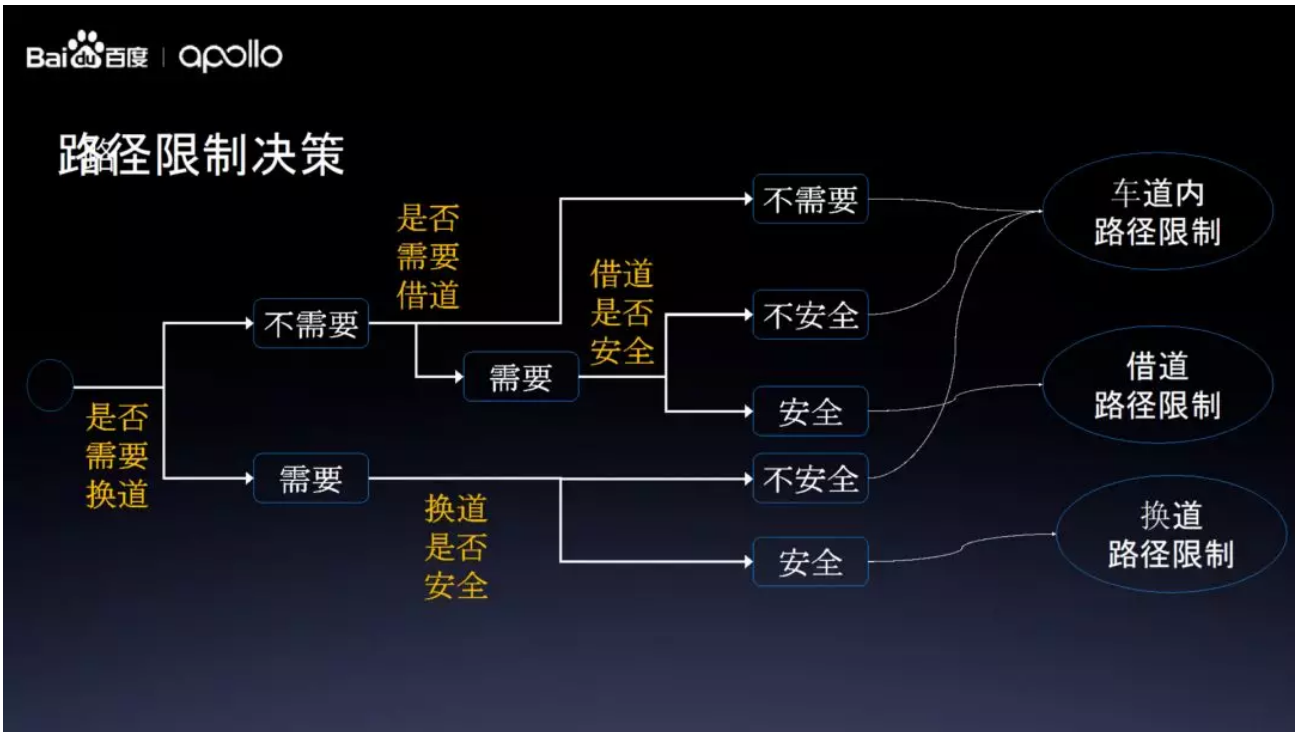
有了这三个优化目标之后，可以把优化问题转化成二次规划问题。求解此二次规划问题，即可得到平滑的参考线。得到参考线后，我们首先沿着参考线找到所有相关的交通标识。



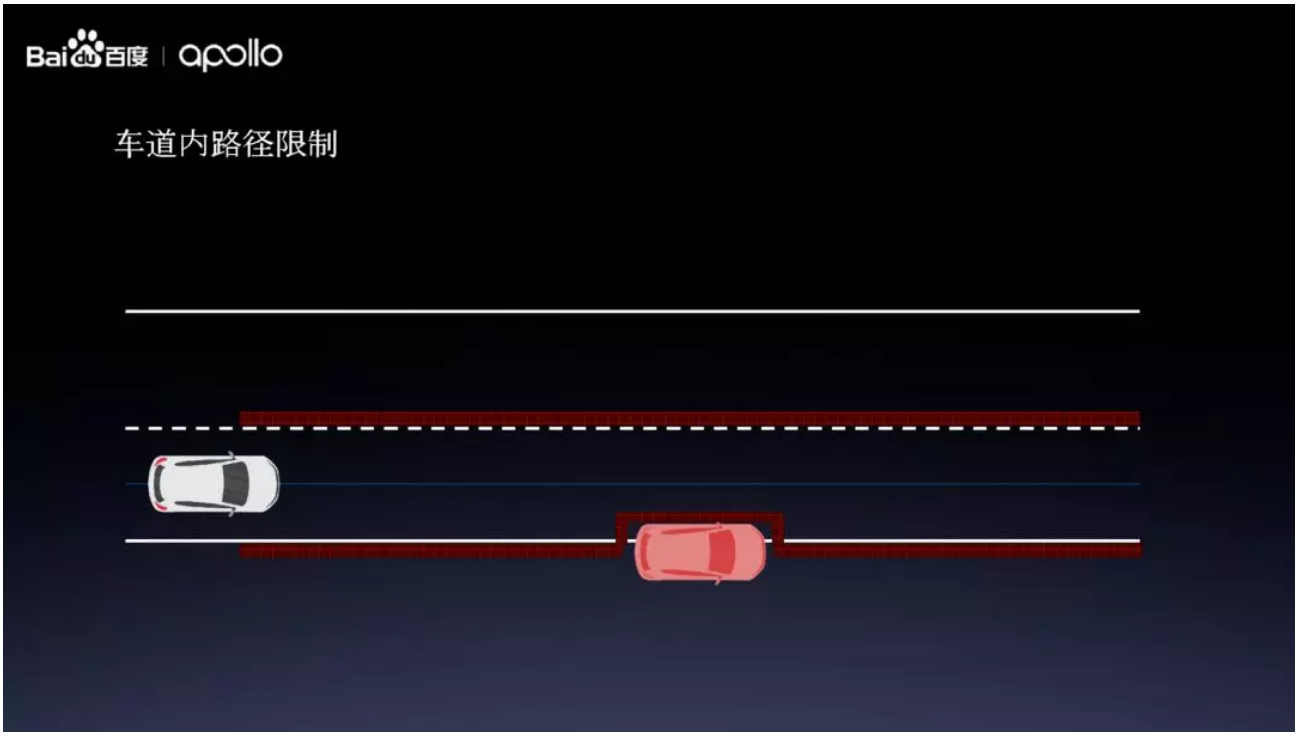
上图的例子是一个十字路口的道路，路口有交通灯。我们用这个例子来说明我们的交规决策是如何实现的。如果交通灯为红灯，我们在交通灯停止线处放置虚拟墙，达到交规限制目的。车必须在墙之前停下来。但只有交通限制并不充分，不能保证车辆按照既定路线行驶，比如车辆可能会绕过虚拟墙。所以我们还需要在交通限制的基础上加上路径限制，目的是不希望车开出道路外，如下图所示。



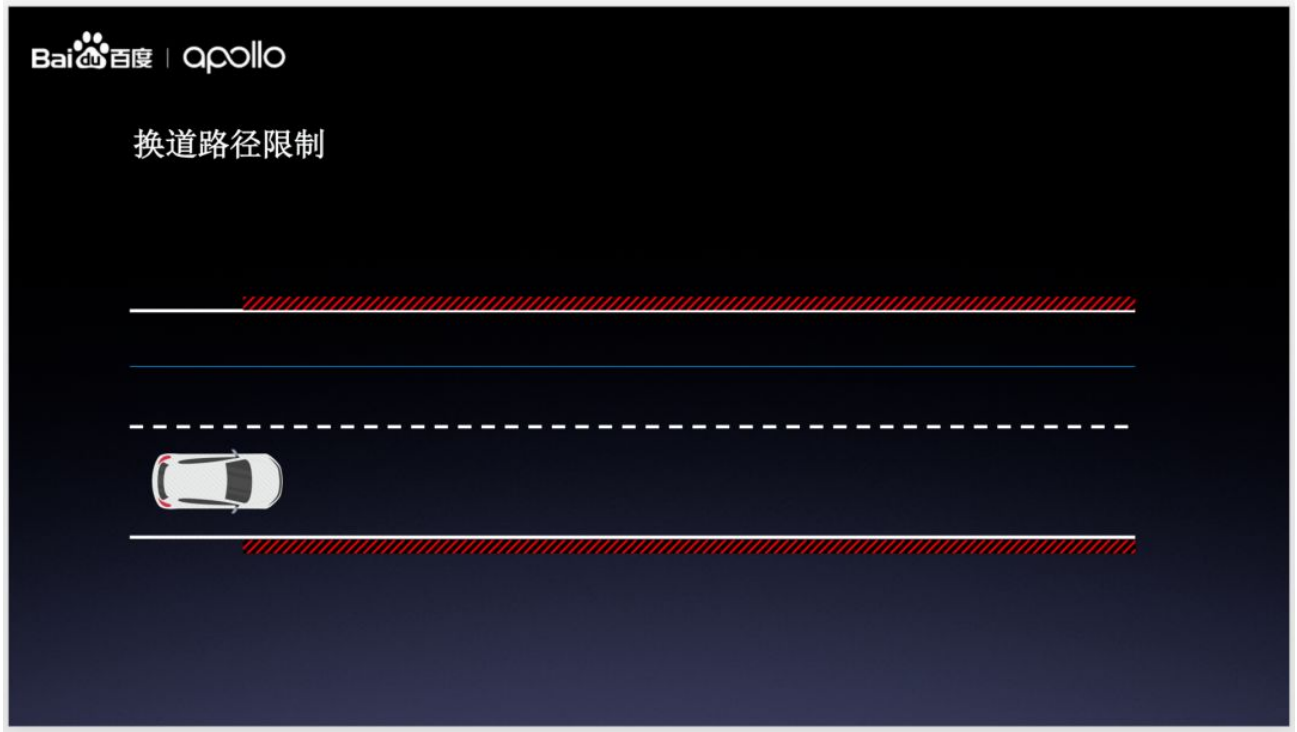
在Apollo中，有三种路径限制，一个是车道内，一个是借道，最后一个是换道。决定用哪种路径限制方法，相当于一个决策问题。首先看无人车需不需要换道，还要判断换道是否安全。只有在需要换道，而且换道安全的情况下才做换道路径限制，同时借道也是这样，需要借道而且是在安全的情况下，我们才作出借道路径限制。其他情况，做车道内路径限制，如下图所示。



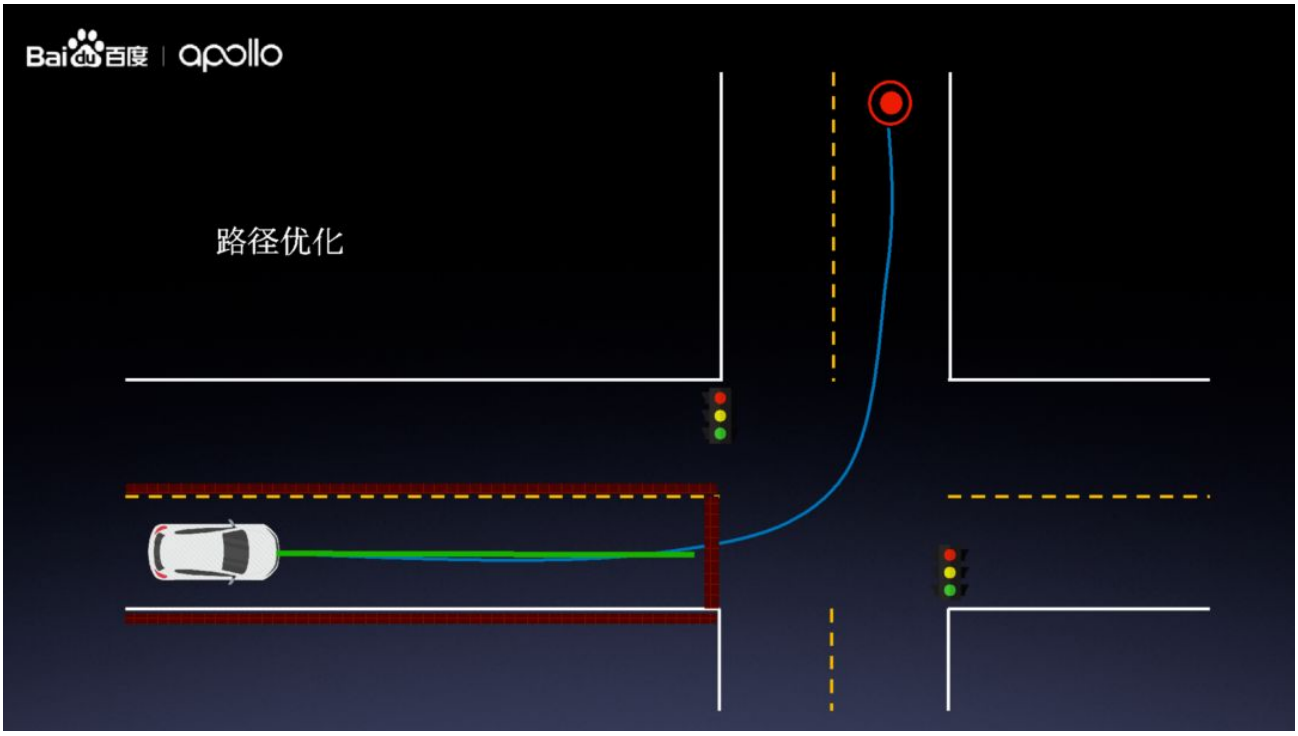
下面看如何做**车道内路径限制**，跟刚才十字路口左转例子相似，车道边界作为路径限制，如果考虑到障碍物还需要做一些变化。首先，障碍物在边界之外，为防止车辆与障碍物相撞，车子不能开到障碍物周围；第二，在保证与障碍车不发生碰撞，对这个边界做调整之后，宽度至少大于车本身宽度。如果宽度不够，需要做借道路径限制，根据实际情况并结合规则判断到底应该选择什么样的路径限制。



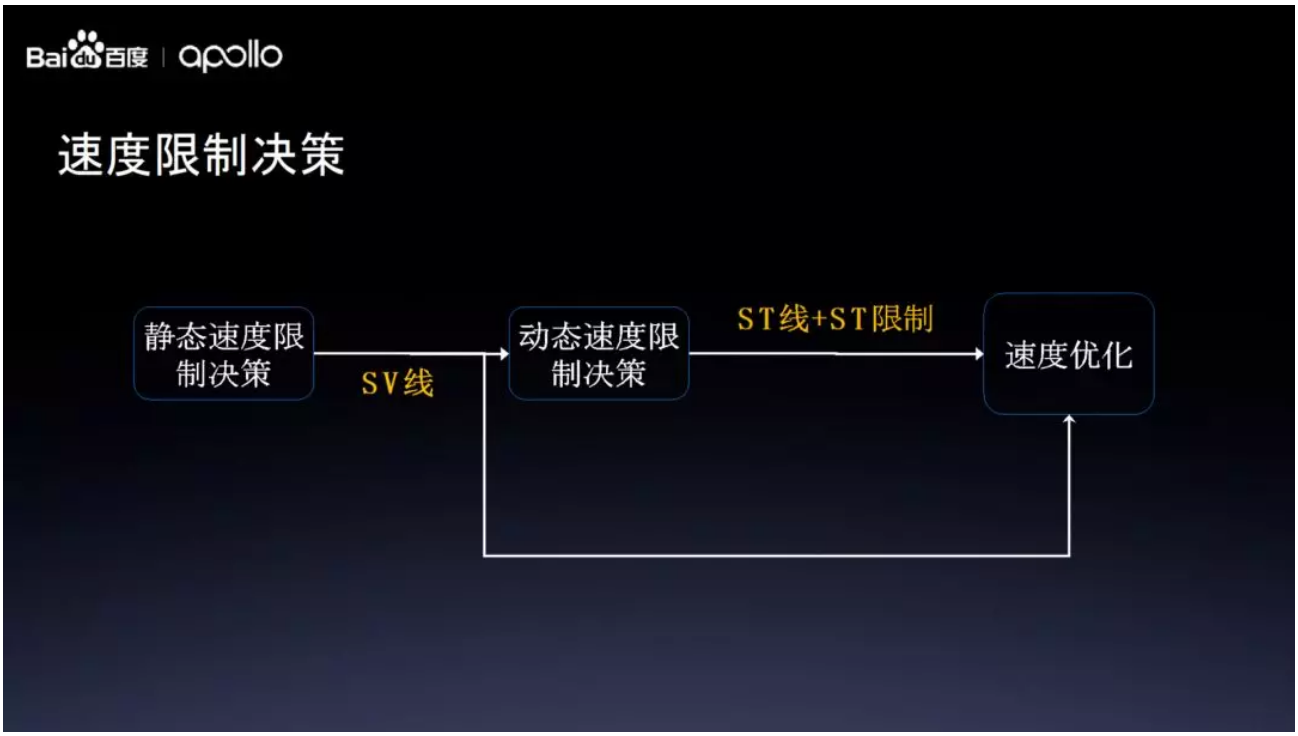
对于换道路径限制，主车的目标道是上方车道，这个时候边界限制要把上方目标车道和当前车道包含进来。



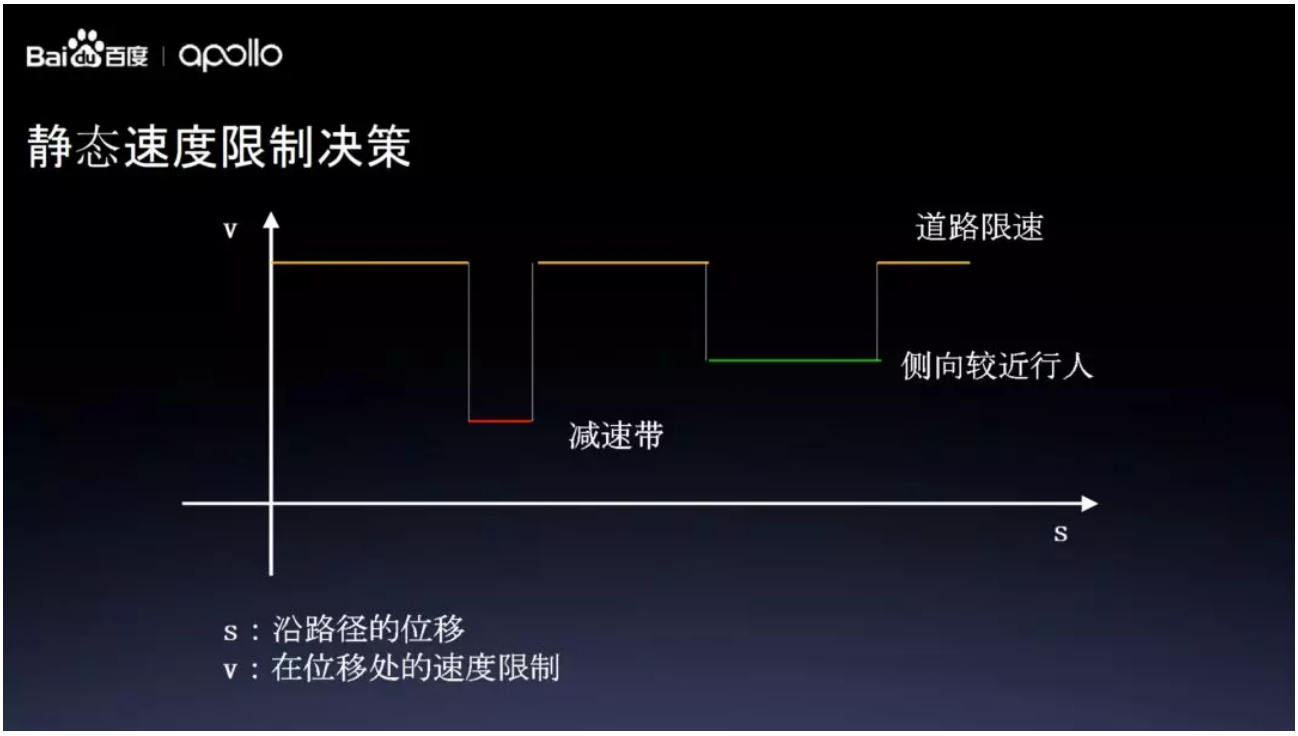
有了交规限制和路径限制之后，无人车可驾驶区域大大缩小，缩小到一个红框里头，红框将障碍物排除在外，只要保证红框安全，同时保证遵守交规，无人车在里面怎么开都可以。有红框之后进行路径优化和平滑处理，就可以得到这样一条平滑的路径（下图绿色的线）。



有了路径之后，还需要考虑速度限制，即以多大的速度沿着这条路径行驶。速度限制决策大概分为**静态速度限制**和**动态速度限制**。

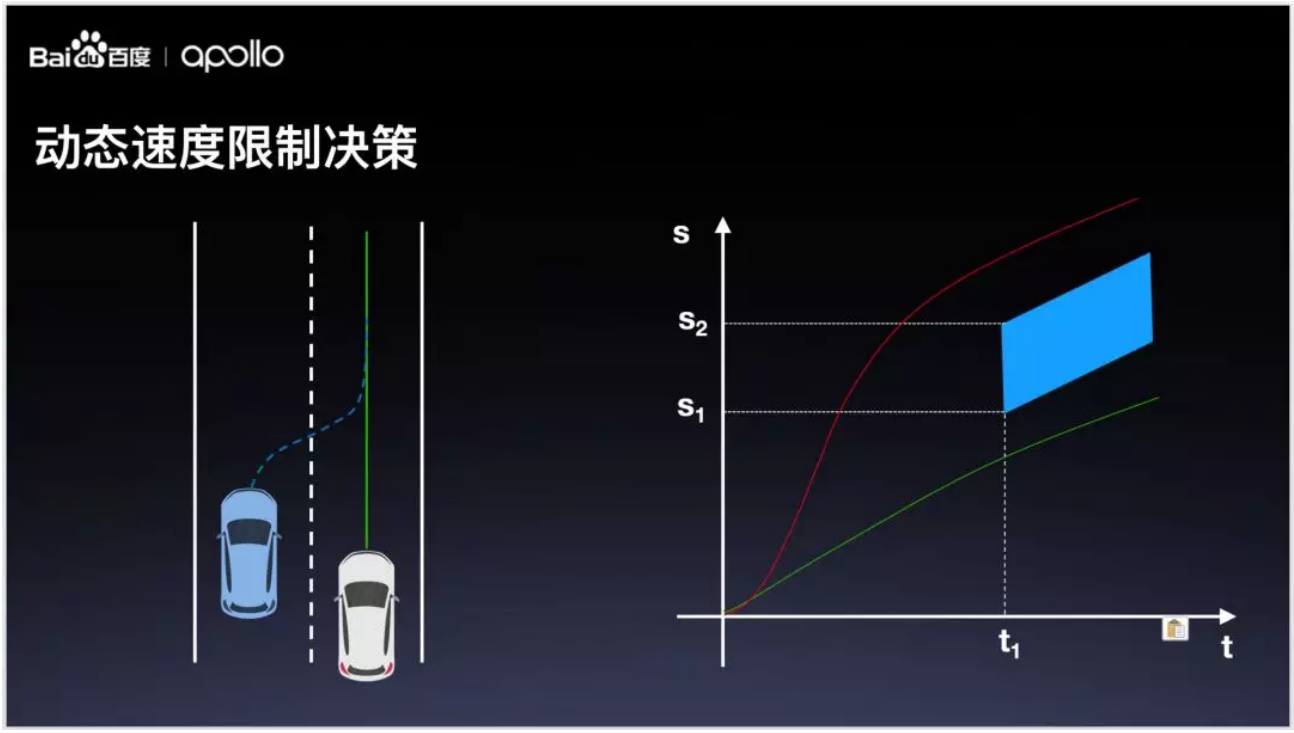


静态速度限制又叫SV线，横轴S指沿着路径的位移，纵轴V是在位移处的速度限制。这里列举三种限制，第一是道路的基本限速。第二是路上的减速带，我们需要把速度降到足够低，让乘客体感足够好。第三种情况是在离行驶路径较近处有行人或者自行车，为了安全，车速不能太快。在实践中静态速度限制有更多因素。

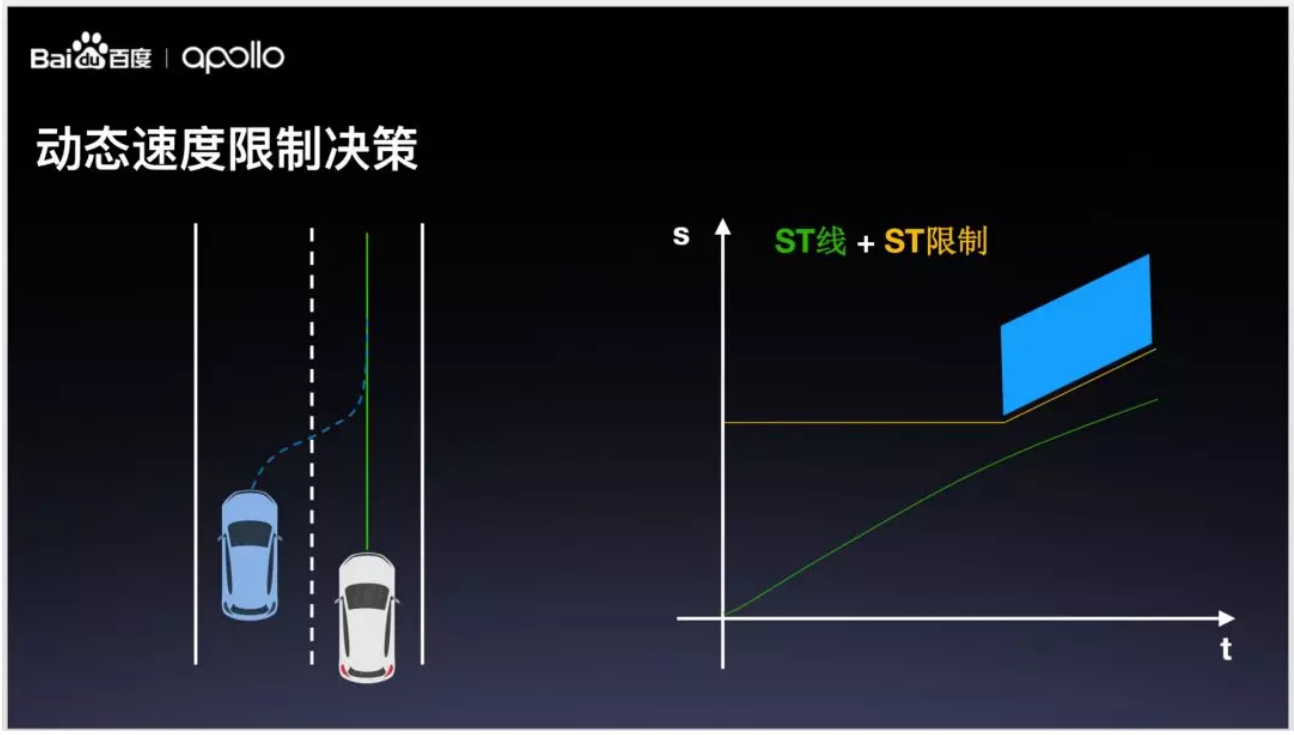


动态速度限制的工作原理我们用下图来解释。白色是无人驾驶车，绿色是规划好的行车路径，旁边有一个运动的蓝色障碍车，根据我们预测它想换道到我们这个车道，这个时候动态速度，有两种决策结果，一种是避让蓝色车，让它先换进来；另外一种是加速超过去，在蓝车换到车道之前就超过这辆车。这种决策过程在Apollo里是用st图来实现的，t是时间，s是沿着路径的位移，第一步把障碍车投到st轴，如图中蓝色平行四边形所示，t1代表t1时刻蓝色车切入到当前车道车，s2减s1是蓝车的长度，图中红色线代表主车角色超障碍车，在蓝色车换道之前超过这个车；绿色是减速让蓝色运动车。

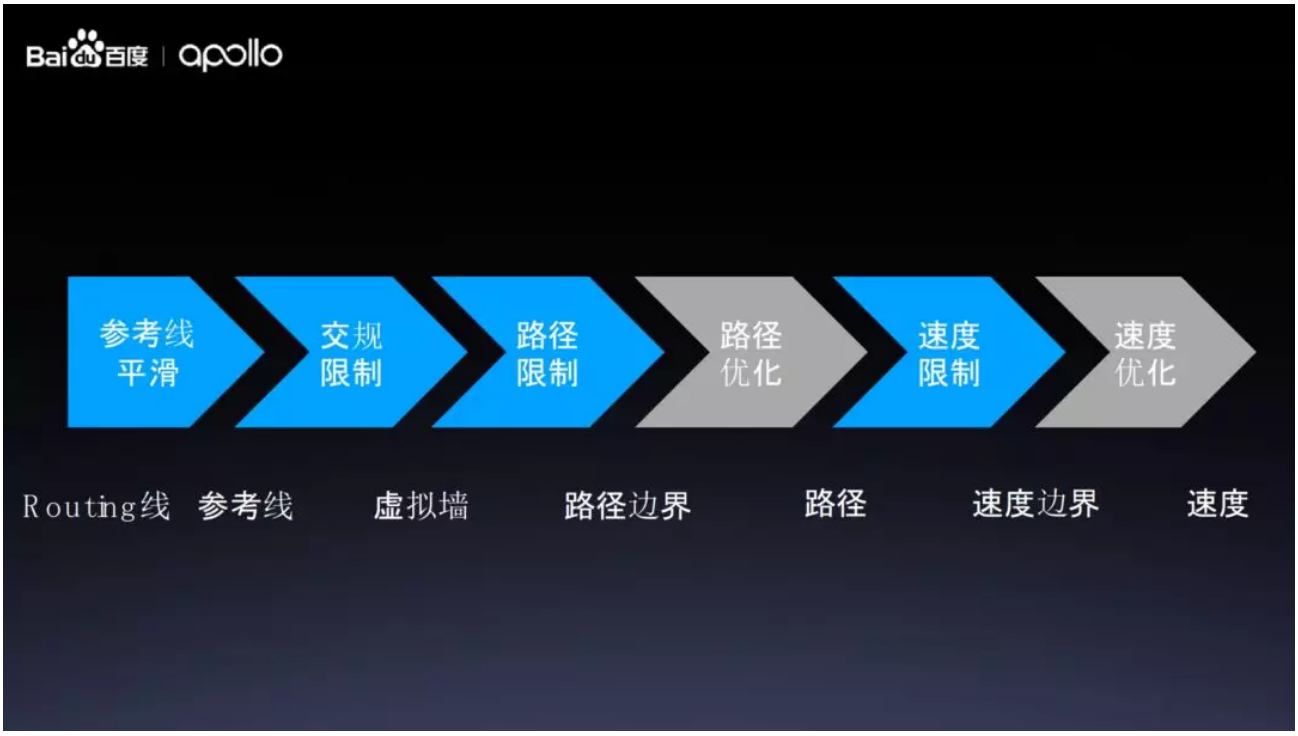
到底如何选规则需要考虑很多因素，主要有三点，第一点要考虑舒适度，不能够加速度太快，或者加速变化太快；第二点要考虑车本身物理特性，比如车最高加速度能达到多少；第三点考虑SV道路限速。



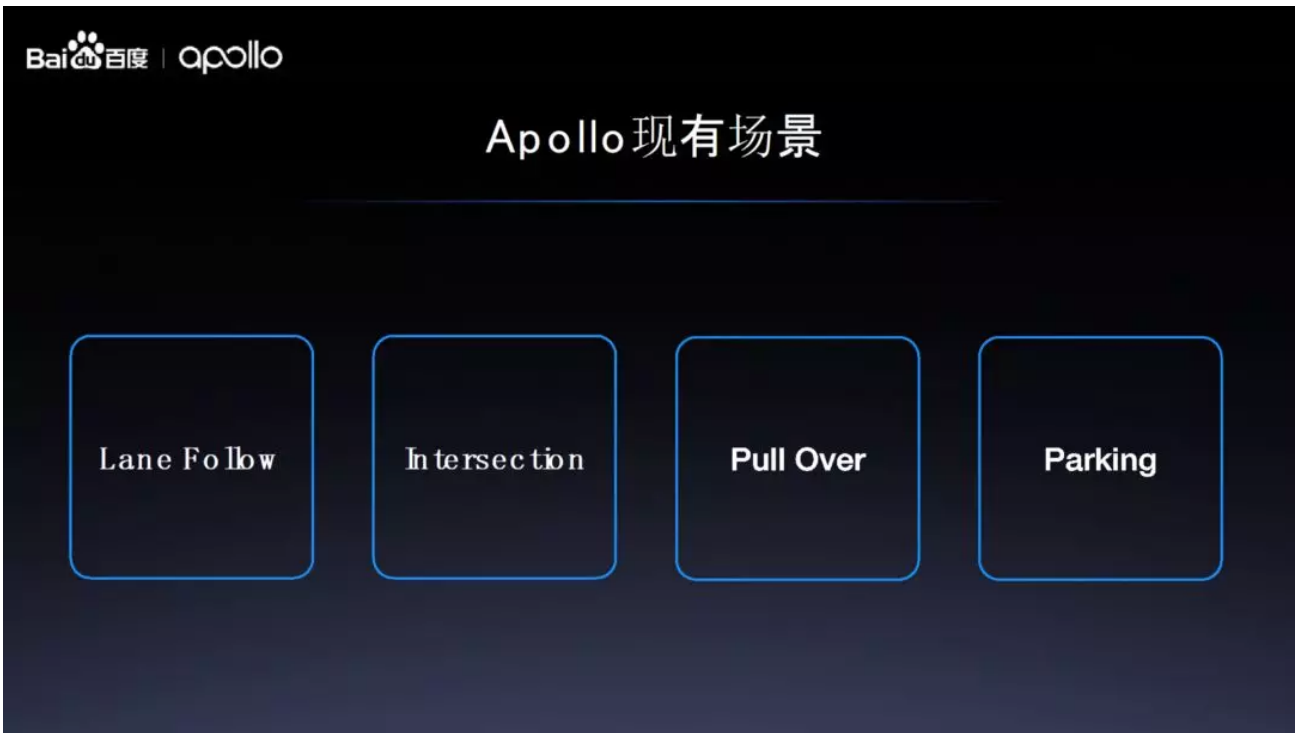
得到一条选定的st线后，我们就可以得到st的限制（如下图所示）。我们把得到的st线和st限制作为速度决策的输出传到速度优化模块。



总体决策流程如下图所示。首先有一个参考线平滑过程，将Routing线转成参考线，有了参考线之后做交规限制，第一步是沿着参考线找到交通标识，如果需要停在参考线上则建虚拟墙，保证车辆能够遵守交规。之后是路径限制，得到路径边界，有了路径边界传给路径优化，得到最终路径。最后做速度限制，得到一个速度边界，这三个信息传给速度优化得到最终速度。有速度有路径之后，将两个信息结合在一起实现Planning。



下面简单介绍一下**决策场景**。决策场景是在Apollo3.5中提出的，在最新5.0版本进行了一定升级和改进。场景是地图中有一定特征的路段，或者是无人车想要完成的一系列复杂的动作。比如5.0推出的靠边停车功能。我们为什么要建场景，其实场景给我们带来两点好处。第一，场景之间互不干扰，可以进行并行开发和调参；第二，可以实现一系列的有时序或者依赖关系的复杂任务，就像状态机一样进行任务和任务之间的切换。



在Apollo5.0里有四个大的场景。第一个是lane follow，属于默认场景；第二是十字路口；第三个是5.0发布路边停车，路边停车是实现无人驾驶过程中让客人上车或者下车功能；第四是停车场停车的功

能。不同的场景之间的转化，我们是采用中央的场景管理器，统筹安排车辆应该处于哪一个场景。另外每个场景里面有多个阶段，每个阶段之间关系可以互相切换。

以红绿灯右转场景为例。首先确认场景进入条件，明确什么情况下才可以进入这个场景，红绿灯右转场景有两个条件，第一接近路口；第二路口是右转而不是左转。进入红绿灯右转场景后，如果当时状态是红灯，第一阶段是Stop阶段，车辆停下来，Stop阶段就完成了这个阶段的事情。之后系统进入到Creep阶段，这个阶段在路口慢速探头，目的是扩大感知视野，看到对象车，或者在车道线、停止线之后看不到的场景，同时确认安全。如果右转安全则进入下一个阶段Cruise阶段，完成右转通过十字路口。

以上就是本次关于**Apollo决策技术**