

# 无人驾驶车辆的避障、路径规划和控制

**编者按：**自动驾驶汽车的行驶安全性是一个具有挑战的问题，目前研究热点在于车辆自主避障。论文针对这一问题，提出了一个全局的避障控制框架，包括感知、规划和控制三个模块。在感知模块采用基于信度网格占用的方法识别障碍物，轨迹规划模块利用参数化的Sigmoid函数曲线设计避障轨迹，最后设计前馈加鲁棒反馈的横向控制器完成车辆的轨迹跟踪，在常用的方法基础上，作者还采用滚动地平线以及调节特定点横向和纵向误差的方法减少计算量，最后在实验平台ARTEMIPS上验证了所提出框架以及各方法的有效性。



**摘要：**在自动驾驶车辆中，避障需要三个主要层次，即感知、路径规划和制导控制。在这篇论文中，通过考虑这三个层次之间的联系，提出了一个全局架构。在环境感知层面，提出了一种基于信度网格占用的动态障碍物检测方法。因此，在轨迹生成中需要考虑目标的姿态。后者是基于平滑轨迹的s形函数。最后，控制制导利用该避障轨迹生成适当的转向角。整个策略在实验车上得到了验证。实验结果表明了该方法的有效性。



## 1 介绍

智能交通系统解决了复杂环境中具有挑战性的自主性和安全性问题，因此它吸引了研究人员特别的关注。自动驾驶汽车概念的主要模块是感知、规划和控制。

实际上，感知是由环境建模和本地化组成的。它们分别依赖于外界和本体的传感器。接下来，规划旨在基于感知结果传递的信息来生成最佳轨迹，以便到达给定的目的地。最后，控制模块专用于通过命令车辆的执行器来跟踪生成的轨迹。

本文将针对避免障碍的具体情况介绍该过程的每个模块。这些任务在全局体系结构中的集成是本文的主要贡献。感知模块确保根据准确的网格表示来描述环境。占用栅格地图（OGM）的使用对于避免障碍特别方便，因为它可以识别可行驶空间并在场景中定位静态和动态对象。然后在路径规划级别使用要避免的物体的姿态，该路径规划根据[1]中所示的 S 型参数化函数和滚动水平线生成轨迹和速度曲线。所获得的曲率轮廓被认为是引导控制模块的参考路径。根据使用撞击中心（CoP）而非经典重心的横向引导控制器，此水平为车辆提供了适当的转向角。所提出的控制器基于前馈和鲁棒的状态反馈动作，以分别减少干扰对横向误差的影响并保证横向稳定性[2]。

该文件的组织结构如下：第二部分介绍了全局方法，其中包含为避免避障而将要实施的不同模块。第三部分介绍了基于信度网格占用的动态物体检测方法。第四部分说明了基于参数化S型函数和滚动地平线

的避障算法。第五节详细介绍了基于前馈耦合到鲁棒状态反馈的控制器设计。第六部分说明了该实验方法的实验平台和结果。最后，第七节总结了论文。

## 2 避障策略

本节介绍基于三个模块的全局避障策略的标题，如图1所示。本节将简要介绍每个层次。

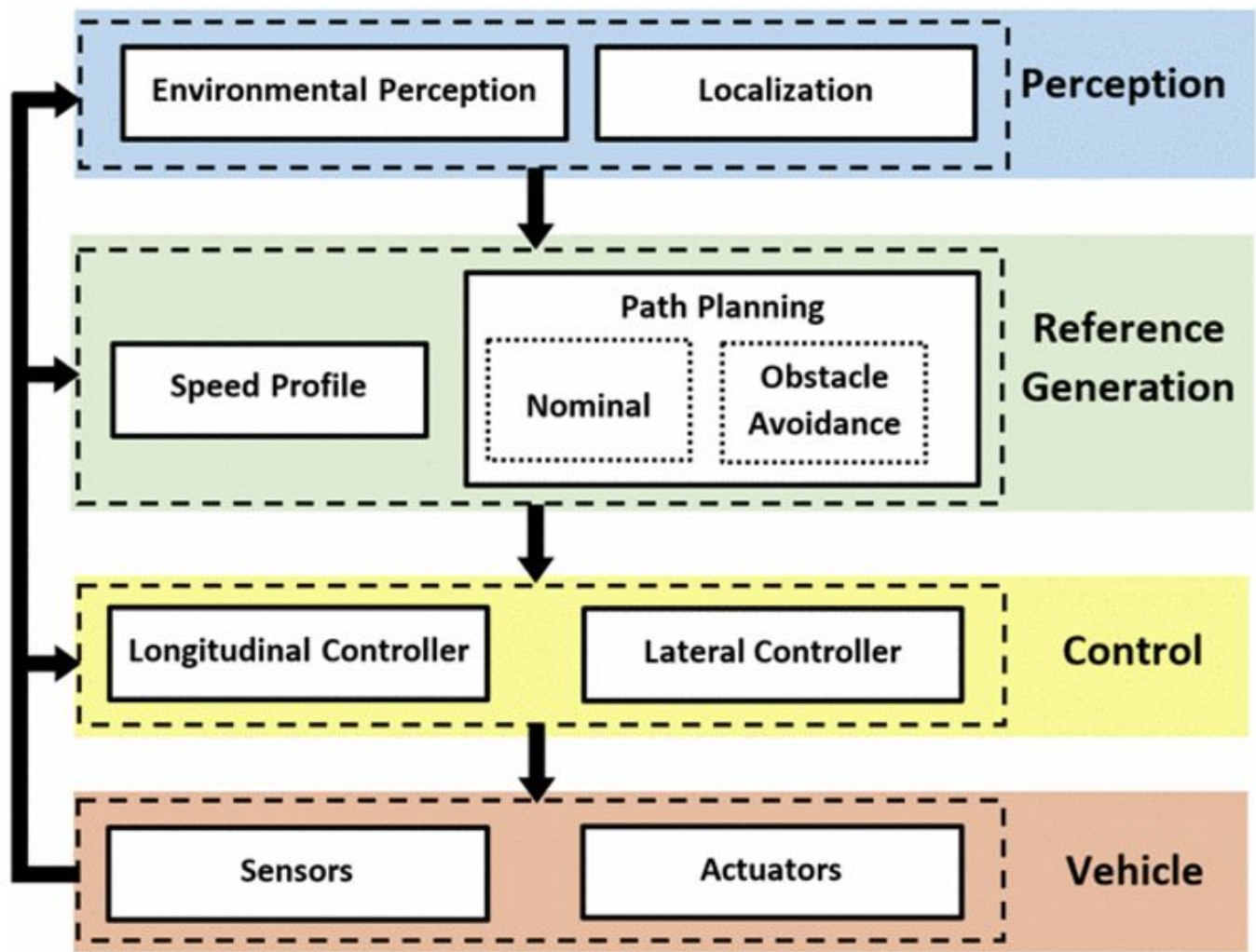


图1 避障策略

### A. 感知模块

正确而有效地感知环境对于自动驾驶汽车是强制性的。这项研究主要关注环境感知，以提取静态/动态物体的位置以及基于外部感知传感器的可驾驶路径。定位部分未被视为车辆的位置被认为是已知和可靠的。提取道路和周围物体信息的最常用方法之一是“占用网格”（OG）。它可用于多种应用，例如避免碰撞，传感器融合，目标跟踪以及同时定位和映射（SLAM）[3]。OG的基本思想是将环境图表示为二进制随机变量的均匀间隔字段，每个变量均表示环境中该位置处是否存在障碍物[4]。可以根据许多形式来生成它，以在已知车辆姿态的前提下处理嘈杂和不确定的传感器测量数据。在本文中，OG是由 Dempster和Shafer [5] [6]提出的信念理论定义的，因为它对不确定性，不精确性和未知部分进行建模，还允许管理数据融合中的冲突。第三部分给出了更多细节。

### B. 参考轨迹生成模块

该模块专用于定义轨迹以及车辆要遵循的相应速度曲线。规划者从感知模块接收可驾驶区域和障碍物位置。根据这些信息，可以生成几何轨迹以及速度曲线。本文着重于路径规划策略。本部分旨在基于感知的可驾驶区域，提供从起点到终点的名义轨迹。

当检测到障碍物时，计算第二轨迹（避障轨迹）以确保自动驾驶车辆乘客的安全性和舒适性，并在避开后加入标称轨迹。这种避免轨迹可以通过局部规划获得，因为它只涉及标称轨迹的一小部分。为了降低轨迹生成算法的计算成本，采用了滚动地平线方法，如[1]所述，其工作在本文中进行了扩展，请参见第IV节。然后，可以将这些轨迹（标称和避障）视为控制模块（主要是横向控制器）的参考。

### C. 控制模块

控制模块由两个主要部分组成：纵向和横向控制器，确保自动驾驶控制。这里主要关注侧向控制器以处理避障。实际上，适当的转向角由侧向控制器提供，以遵循参考生成模块给定的期望路径。可以通过减少两个跟踪误差，即横向误差和航向误差，来实现对所需路径的跟踪。在文献[7]中存在的几何和动态横向引导策略中，此处采用了基于碰撞中心（CoP）的动态方法[8]。该选择取决于此控制方法的性能。CoP是位于车辆重心（CoG）前面的几何点，可以预测横向位置误差。然后可以期望更好的轨迹跟踪。另一方面，由于CoP的运动与后轮胎侧向力[9]分离，如第V节所示，横向动力学方程变得不太复杂。

## 3 基于动态网格的动态障碍物检测

OG是一种将空间多维细分为单元的表示形式，每个单元都存储其占用状态的知识[4]。如今，由于可以使用更强大的资源来处理OG的计算复杂性，因此OG大量使用。网格的构造已应用于多个维度（2D，2.5D和3D）[10]，其中每个单元状态均根据选定的形式描述。最常见的是贝叶斯框架，该框架首先被Elferes [4]所采用，随后又进行了许多扩展，成为了著名的贝叶斯占用过滤器（BOF）[11]。其他作品则提出了基于Dempster-Shafer理论的形式主义，该理论也被称为证据论，随后进行描述。

### A. 使用信念理论

归纳为概率论，信念理论提供了数据和源缺陷的充分表示，因此适合于ITS中的感知。它提供了广泛的融合算子，可以根据应用程序处理这些属性。一些使用Belief框架构建OG的研究可以在[12]，[13]中找到。这项工作源自[13]的研究，该研究提出了一种基于由此产生的冲突的移动物体检测和可行行驶空间确定方法。为此，将识别帧定义为包括认为其为空闲（F）或已占用（O）的单元的状态。识别框是 $\Omega = \{F, O\}$ 。参考功效集框架包含以下假设的所有可能组合： $2^\Omega = \{\emptyset, F, O, \{F, O\}\}$ 。为了表达对每个状态的信念，质量函数 $m(\cdot)$ 分别表示冲突 $m(\emptyset)$ ，自由状态 $m(F)$ ，占领状态 $m(O)$ 和未知状态 $m(\{F, O\})$ 。

### B. 传感器模型

基本上，传感器模型是如何计算与度量对应的状态的质量函数。在我们的应用中，要使用的传感器是3D多回声LIDAR（请参见第VI节）。输入数据将包括根据一点 $p_i$ 范围 $r_i$ 和一个角度 $\theta_i$ 。根据此数据集，构造极坐标中的扫描网格（SG）。SG的每一行对应定义在 $R \times \Theta$ 的一个角扇区 $\Theta = [\theta^-, \theta^+]$ 。单元格的范

围是 $R=[r, r^+]$  这意味着每个像元由一对质量定义为 $m\{\Theta, R\}$ 。每个命题对应的质量 $A \in \Omega$ 在这里找到 [13]:

$$m\{\Theta, R\}(\emptyset) = 0 \quad (1)$$

$$m\{\Theta, R\}(F) = \begin{cases} 1 - \mu_F & r_i \in R \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$m\{\Theta, R\}(F) = \begin{cases} 1 - \mu_O & \text{if } r^+ < \min(r_i) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$m\{\Theta, R\}(O) = \begin{cases} \mu_F & r_i \in R \\ \mu_O & \text{if } r^+ < \min(r_i) \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

其中 $\mu_F$ 和 $\mu_O$ 分别对应于传感器的误报警和漏检的概率。为简单起见，将说明这些质量函数， $m(O)$ ， $m(F)$ 和  $m(\Theta)$ 。

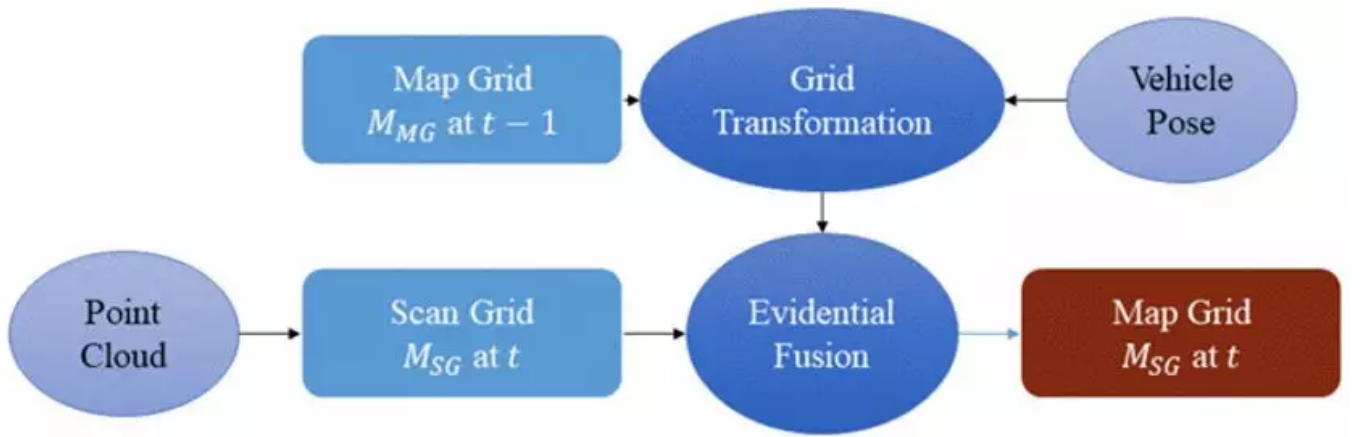


图2 地图网格(MG)构造

图2示出了使用在时间 $t$ 处提供的传感器点云来建立和更新MG的过程。此更新是根据多网格证据融合完成的。这是该过程中最有趣的部分，因为它允许时间更新地图网格并评估单元状态。在信念理论的各种运算符中，使用了Dempster-Shafer组合规则：

$$m_{MG,t} = m_{MG,t-1} \oplus m_{SG,t} \quad (5)$$

其中  $m_{MG,t}$ 和 $m_{MG,t-1}$ 分别表示在时间 $t$ 的地图网格和扫描网格的质量函数。运算符定义为：

$$(m_1 \oplus m_2)(A) = K \sum_{\forall B, C \in 2^\Theta, B \cap C = A, A \neq \emptyset} m_1(B).m_2(C). \quad (6)$$

其中，

$$K^{-1} = 1 - \sum_{\forall B, C \in 2^\Theta, B \cap C = \emptyset} m_1(B).m_2(C) \quad (7)$$

产生的结果  $m_{MG,t}(A)$  定义每个单元的状态，该状态取决于先前的状态和新的度量。发现根据每种状态产生的质量如下[13]：

$$\begin{aligned} m_{MG,t}(O) &= m_{SG,t}(O).m_{MG,t-1}(O) + m_{SG,t}(\Omega). \\ &\quad m_{MG,t-1}(O) + m_{SG,t}(O).m_{MG,t-1}(\Omega) \\ m_{MG,t}(F) &= m_{SG,t}(F).m_{MG,t-1}(F) + m_{SG,t}(\Omega). \\ &\quad m_{MG,t-1}(F) + m_{SG,t}(\Omega).m_{MG,t-1}(F) \quad (8) \\ m_{MG,t}(\Omega) &= m_{SG,t}(\Omega).m_{MG,t-1}(\Omega) \\ m_{MG,t}(\emptyset) &= m_{SG,t}(O).m_{MG,t-1}(F) + m_{SG,t}(F). \\ &\quad m_{MG,t-1}(O) \end{aligned}$$

基本上，此属性显示了在  $t-1$  和  $t$  之间的不协调。不协调的出现在单元从自由状态（Free）改变到被占用状态（Occupied）或者相反。因此，冲突的检测可以导致对动态单元格的评估。冲突允许标记占用的单元格，它们根据两种冲突类型改变其状态：

$$\begin{aligned} C_1 &= m_{SG,t}(O).m_{MG,t-1}(F) \quad \text{from } F \text{ to } O \\ C_2 &= m_{SG,t}(F).m_{MG,t-1}(O) \quad \text{from } O \text{ to } F \end{aligned} \quad (9)$$

其中，

$$m_{MG,t}(\emptyset) = C_1 + C_2$$

融合过程通过总冲突将状态质量归一化，但是考虑使用此信息来标记定义动态对象的移动单元。然后，将每个检测的姿势用作下一部分中轨迹生成的输入。

#### 4 轨迹生成



本节专门用于路径规划，即创建几何轨迹（跟随坐标点） $A_i(x_i, y_i)$ 。由于本文旨在验证所提出的回避体系结构的可行性，因此未考虑速度曲线以及相关的纵向控制。如第二部分所述，路径规划模块具有两个目标：根据起点和到达点生成全局标称轨迹，并生成局部轨迹以避免检测到障碍物。在此，焦点集中在回避轨迹的产生上。该回避轨迹必须遵守安全标准，尤其是与障碍物之间的纵向和横向距离。这些距离可以相等，如[1]中最近建议的那样，在障碍物周围创建了一个圆形安全区。本文通过考虑横向和纵向安全性标准不同的整体情况，提出该方法的一般化。

为了获得轨迹的几何形式，基于回旋曲线，贝塞尔曲线或样条曲线等函数，存在几种数学方法[14]，[15]。[16]中对所有这些几何方法进行了详尽的回顾。这些方法具有有趣的特征（平滑度，在候选集合中选择最佳轨迹等），但是它们的计算成本可能很高。其中，sigmoid函数代表了平滑度和计算成本之间的公平取舍。考虑的方法建议将此数学方法与本地Horizon计划程序结合使用，以减少计算成本。这种计划方法的优点在[1]中进行了很大的讨论。该本地计划者考虑从占用栅格中检测到的障碍物的信息，以定义适当的平稳回避操作，并返回到标称轨迹。

### A. 几何规避

图3给出了不同的轨迹：标称轨迹，安全区和最终的平滑回避轨迹。安全区定义 $R$ 是检测到障碍物之后的第一步。该区域不可导航，以免由于与障碍物接近而发生碰撞。 $L_x$  和  $L_y$ 椭圆的半长轴和半短轴分别是定义区域的安全标准。定义后，就可以设计回避轨迹。为了确保乘客的舒适性，选择了基于S形的功能。在图3中，A 指的是起点（即自我车辆的重心），B是sigmoid的屈曲点，而WP是要达到的起点。平滑度 $C$ （）可以进行调整，因此避免轨迹可以定义为

$$Y_m(n) = \frac{L_y}{1 + e^{C(L_y - L_x)}} \quad (10)$$

为了获得在障碍物移动时具有鲁棒性的算法，对每个水平矢量样本重复整个过程（确定安全区域并计算基于S形函数的路点）。

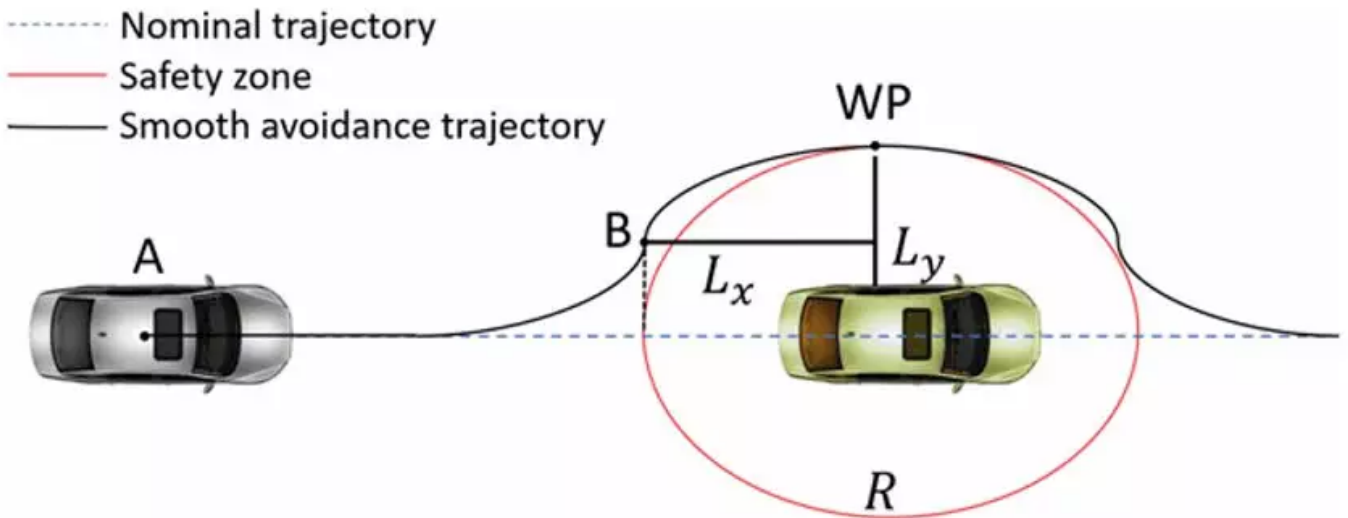


图3 轨迹规划

### B. 水平规划

为了减少算法的计算成本，使用了局部规划器。它没有遵循整个避障轨迹，而是分为几个部分。在此离散水平的每个样本处都计算局部轨迹，从而降低了计算成本并使算法对动态障碍物具有鲁棒性。可以参数化两个参数：样本大小和水平长度。最后一个取决于配备的感知传感器（硬件约束）和车速（滚动水平线）。样本步骤代表轨迹在局部段中的细分。整个原理总结在图4中。

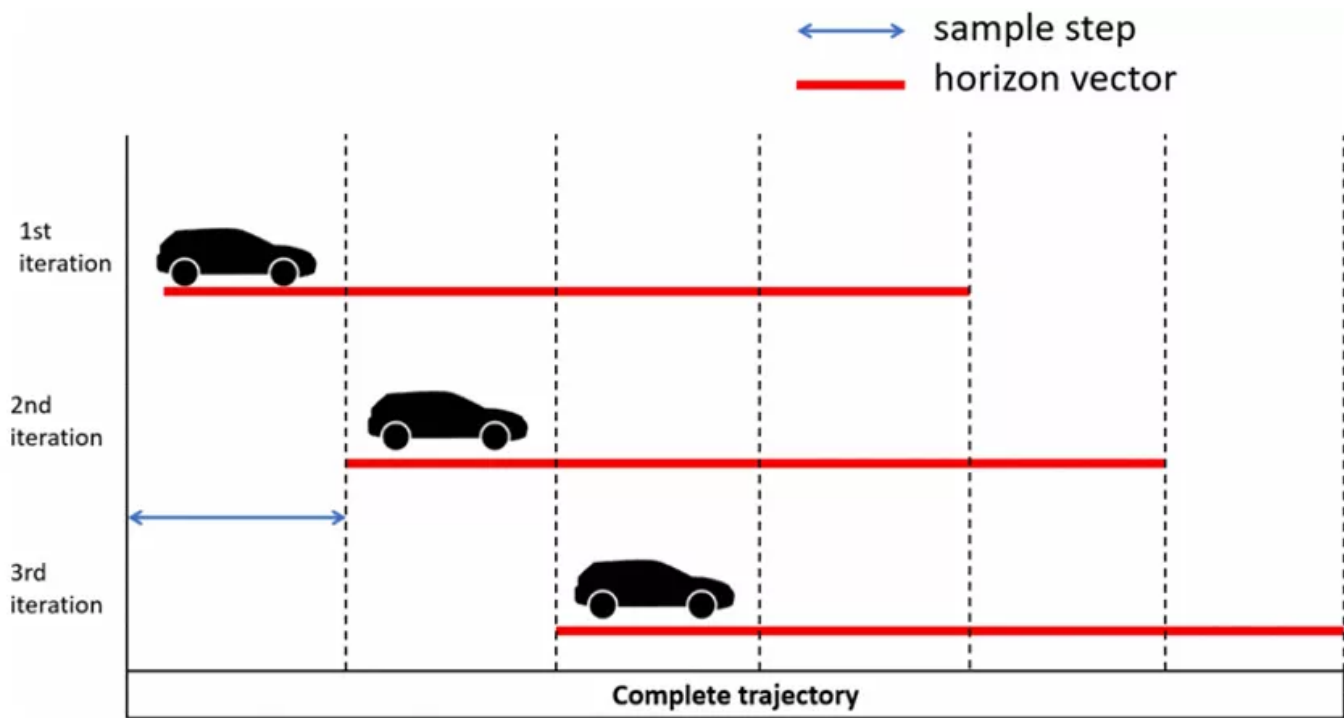


图4 地平线规划

当车辆到达下一个采样步骤时，将再次计算局部水平线。可以看到，两次迭代之间有一个共同的部分，允许算法处理动态障碍。与离散时域一样，样本大小的选择需要在准确性和计算成本之间进行权衡。该算法进行迭代，直到地平线矢量到达完整轨迹的末尾，即，当感知传感器覆盖轨迹的所有细分时。该几何轨迹是制导控制级的输入。

## 5 控制器设计

本节介绍了在图1所示的控制模块中使用的横向控制器设计。横向制导旨在减少两种误差，即横向误差，车辆的CoG与参考轨迹之间的距离以及航向误差，车辆纵轴与参考轨迹之间的距离，如图5所示：

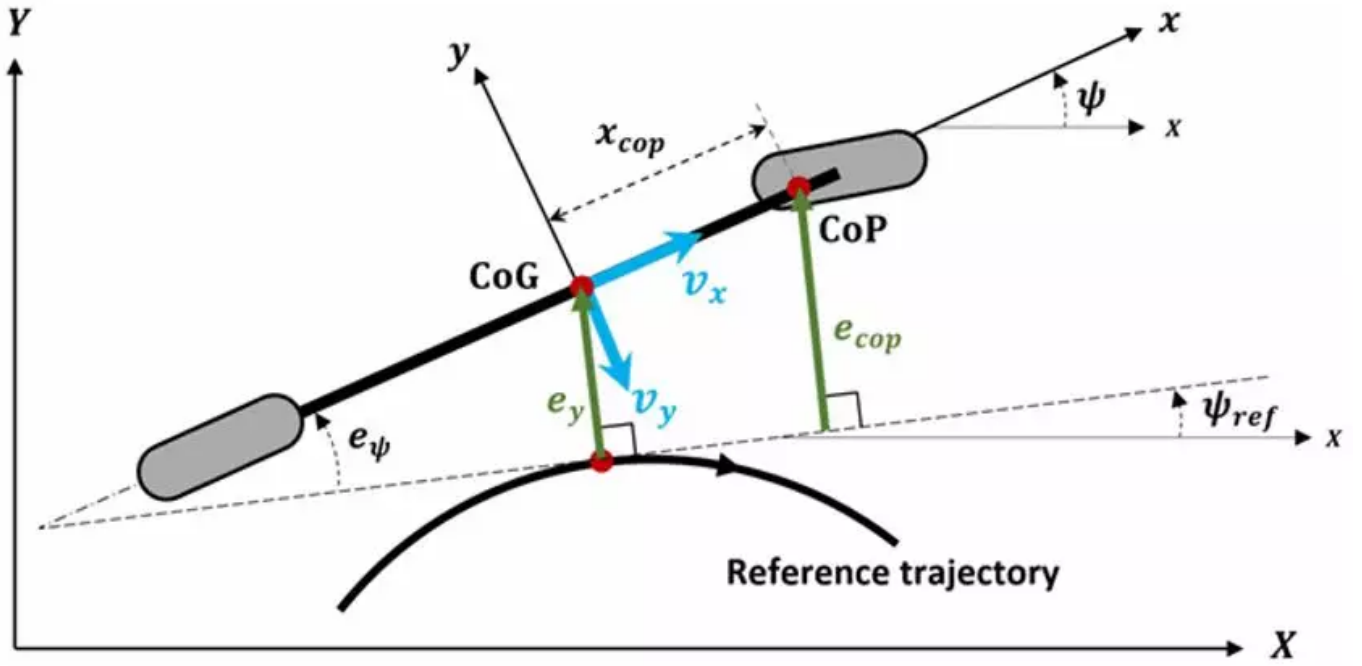


图5 横向和航向误差

这里，代替经典的CoG横向误差  $e_y$ ，建议在CoP定义为[9]时使用横向误差：

$$e_{cop} = e_y + x_{cop}e_{\psi} \quad (12)$$

CoP和CoG之间的距离 $x_{cop}$ 仅取决于车辆配置：

$$x_{cop} = I_z / (mL_f) \quad (13)$$

其中 $m$ 和  $I_z$  车辆质量和偏航惯性，以及  $L_f$ 是CoG与前轴之间的距离。从（12）可以看出CoP横向误差  $e_{cop}$ 高于图5侧向误差 $e_y$ 。这样，可以预期横向位置误差，并且可以预期更好的轨迹跟踪。与基于CoG（重心）的经典控制器相比，此处的撞击中心（CoP）被用作车辆上的几何点。CoP的主要优点是降低了横向动力学方程式的复杂度，因为后轮胎横向力不会影响CoP的运动[9]。

基于平面自行车模型[17]，并使用跟踪误差（11）和（12），用于设计CoP横向导航控制器的跟踪误差模型为：

$$\dot{\xi}(t) = A_c \xi(t) + B_c \delta_f(t) + D_c w_{ref}(t) \quad (14)$$

其中状态向量  $\xi = [e_{cop}, \dot{e}_{cop}, e_{\psi}, \dot{e}_{\psi}]^T$ ， $\delta_f$ 是前轮转角， $w_{ref} = [\dot{\psi}_{ref}, \ddot{\psi}_{ref}]^T$  作为扰动项包含期望横摆角速度和横摆角加速度。



$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-2R_l C_f}{mv_x} & \frac{2R_l C_f}{m} & \frac{2R_l C_f (x_{cop} - L_f)}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-b}{I_z v_x} & \frac{b}{I_z} & \frac{bx_{cop} - c}{I_z v_x} \end{bmatrix},$$

$$B_c = \begin{bmatrix} 0 \\ \frac{2R_l C_f}{m} \\ 0 \\ \frac{2C_f L_f}{I_z} \end{bmatrix}, D_c = \begin{bmatrix} 0 & 0 \\ \frac{-2R_l C_f L_f}{mv_x} - v_x & -x_{cop} \\ 0 & 0 \\ \frac{-c}{I_z v_x} & -1 \end{bmatrix}$$

$$a = 2(C_f + C_r), \quad b = 2(C_f L_f - C_r L_r)$$

$$c = 2(C_f L_f^2 + C_r L_r^2), \quad R_l = (L_f + L_r)/L_r$$

$L_r$ 是CoG与后轴之间的距离， $C_f$ 和  $C_r$ 是前和后轮胎侧偏刚度。注意后 $C_r$ 不在 $A_c$ 第二行 因此，使用CoP可以减少不确定参数的数量。

横向控制器计算适当的 $\delta_f$ 为了保证状态向量误差的收敛接近零。此外，由于跟踪误差模型的动力学特性受到 $w_{ref}$ 的影响，控制器还必须确保其影响的衰减水平。为了实现这些目标，提出了一种由前馈耦合到鲁棒状态反馈的横向控制器[2]：

$$\delta_f(t) = L_{FF}w_{ref}(t) - K_{FB}\xi(t) \quad (16)$$

$L_{FF}$ 和  $K_{FB}$ 分别是前馈和鲁棒的反馈增益。前馈动作旨在部分消除向量的影响 $w_{ref}$ 。CoP的好处在于，获得的前馈不需要了解 $C_r$ 。状态反馈动作可确保误差向量趋于零的指数收敛并减弱矢量 $w_{ref}$ 的影响。可以使用线性矩阵不等式（LMI）来表示这种鲁棒的控制问题，如[2]所示。

## 6 实验结果

### A. 实验设置

实验平台ARTEMIPS是配备有多个传感器的自动测试车：采用DGPS技术的高精度IMU（惯性测量单元）RT-3002、2台IBEO LUX 2D 4层激光扫描仪，2台VLP-16 Velodyne3D激光扫描仪和高范围摄像机MANTA-G125（参见图6））。RT-3002用作位置，速度，加速度和方向测量的参考传感器。LUX扫描仪用于在汽车的前部和后部提供远程检测（以4层点云的形式）。VLP-16用于完成对汽车两侧环境的检测（它们提供16层点云，并具有360°环绕视野）。ARTEMIPS还配备了3个执行器和2个集成的伺服电动机MAC-141，用于控制方向盘和制动踏板，以及用于导航汽

车发动机的多功能NI-daq系统。所有传感器和执行器都链接到运行Intempora的RTMaps软件解决方案的嵌入式计算机。它是专用于多传感器和多执行器系统的平台。

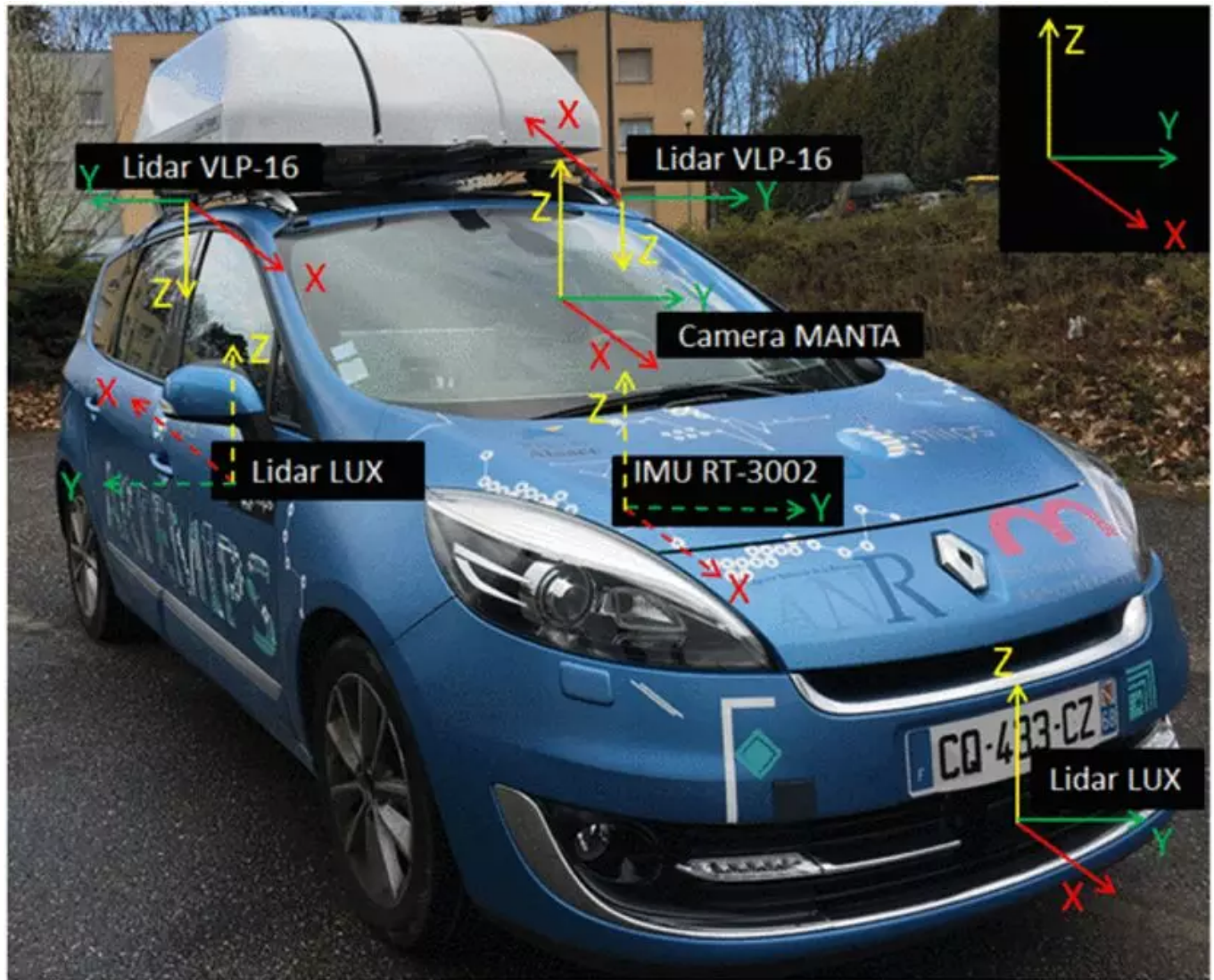


图6 实验平台ARTEMIPS及其参考框架

## B. 实验结果

出于可读性目的，仅通过考虑障碍物避开情况的一种实验方案来评估所提出体系结构的性能。此测试以恒定速度执行 $v_x=10\text{km/h}$ 。

根据来自四个激光扫描仪的记录数据集，根据第三部分中所述的方法，将点云用于OG的构造。图7显示了周围的场景和要避免的对象。OG的时间融合突显了描述动态单元的冲突。应用了层次聚类算法（来自MATLAB中的统计和机器学习工具箱）来构造动态对象。它们如图8所示。通过3D边界框。显示的坐标对应于基于GPS数据的车辆姿态。要避免的对象是红色的对象。可以注意到，由于方法对定位误差的敏感性，可以发现一些错误的检测结果。



图7 用于避障测试的顺序

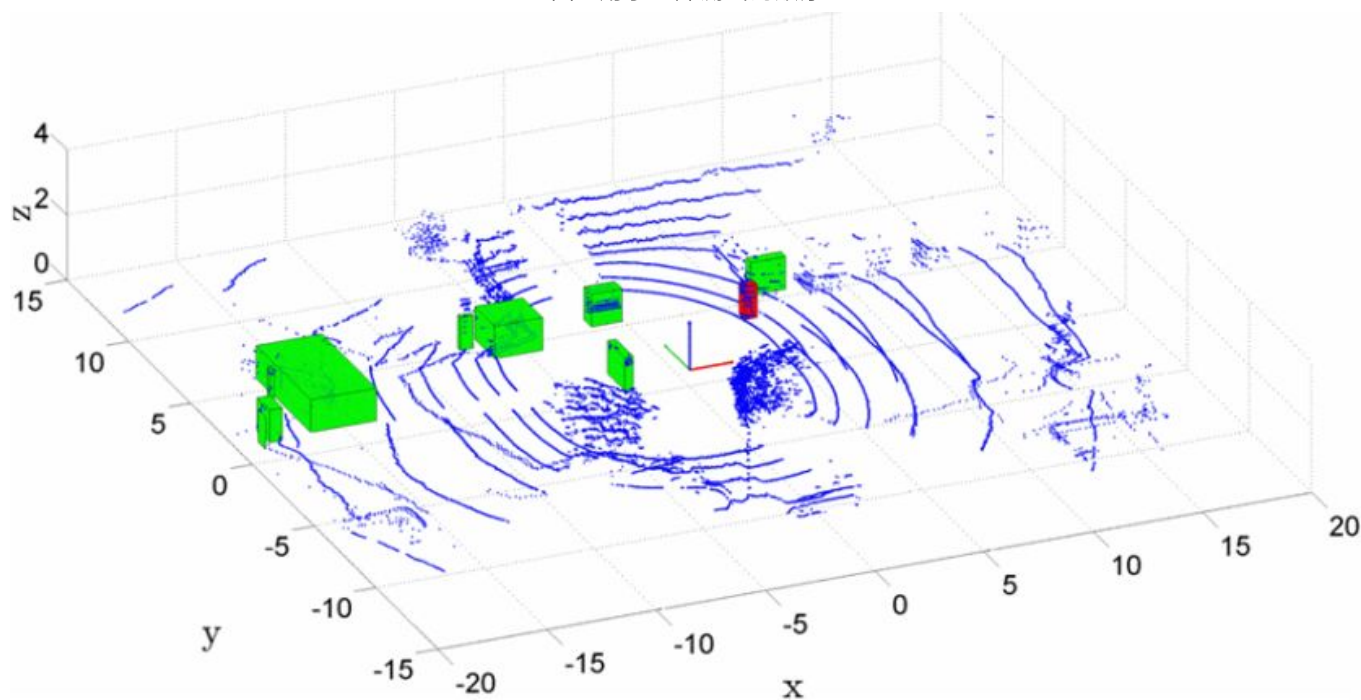


图8 点云，原点坐标和障碍物检测

实验结果如图9所示。从左上图可以看出，标称轨迹线与物体的位置相交，而生成的红色路径避免了障碍物。还可以观察到，横向控制器确保了良好的轨迹跟踪，并可以避免在13 s和20 s之间检测到障碍物。在此时间间隔内，控制器会生成从正值变为负值的转向角，以避免障碍物并确保较小的跟踪误差和。



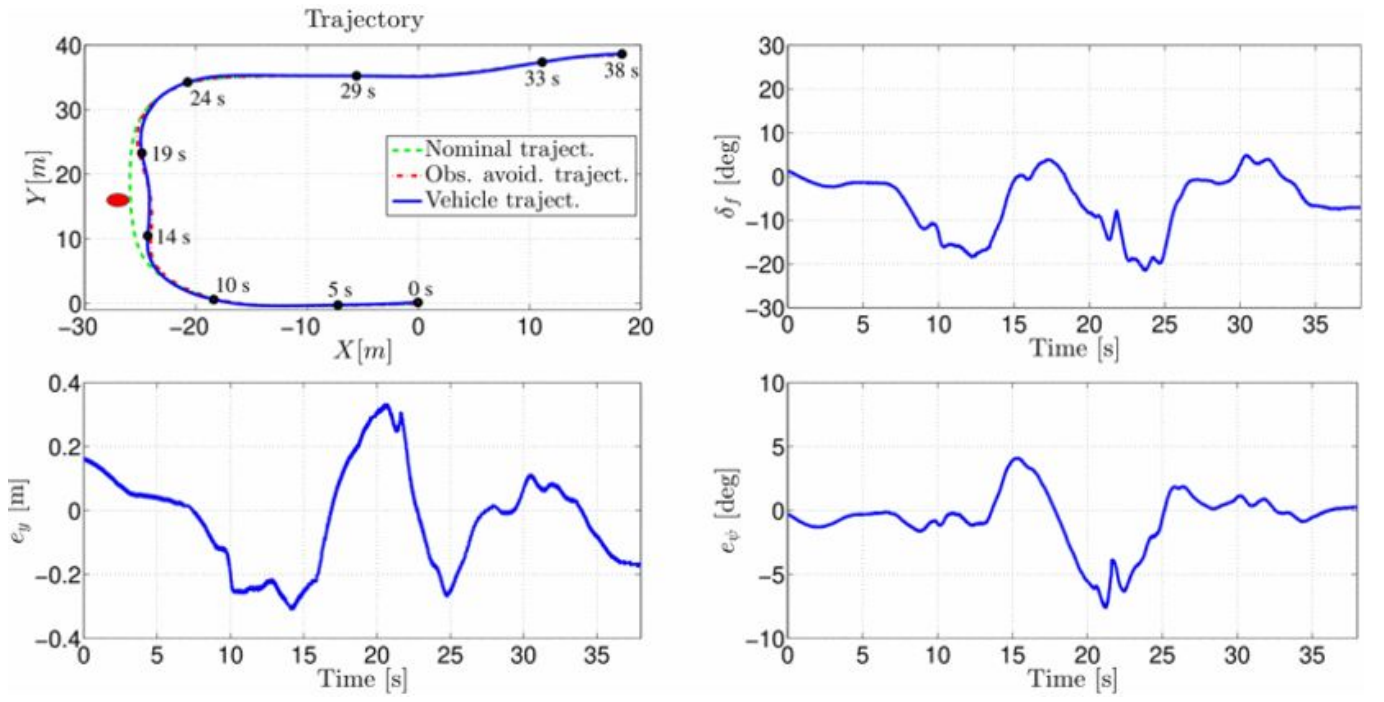


图9 转向控制器结果

## 7 结论

本文提出了一种基于感知，路径规划和控制指导三个层次的动态避障方案。动态障碍物检测是根据证据占用网格进行的。路径规划基于sigmoid函数生成平滑的轨迹，以避免检测到障碍物。最终，车辆通过碰撞中心的基于横向控制的策略跟随车辆生成的参考轨迹。在我们的测试车辆上的实验结果表明，该方法可有效避开障碍物。未来的工作将包括考虑定位策略以及在更复杂的情况下对该方法的评估。