

无人驾驶技术入门（十二）

无人驾驶中的坐标转换

陈光

在第十一期的分享中,我以CAN总线的解析为例,介绍了如何通过解析CAN总线的消息,获取传感器的数据。这一期的分享将会集中在如何将传感器坐标系下的数据转换到自车坐标系下。

无人车上拥有各种各样的传感器,每个传感器的安装位置和角度又不尽相同。对于传感器的提供商,开始并不知道传感器会以什么角度,安装在什么位置,因此只能根据传感器自身建立坐标系。无人驾驶系统是一个多传感器整合的系统,需要将不同位置的传感器数据统一到一个固定的坐标系——自车坐标系下,才能分析当前无人车所在的道路场景。

1 无人车的自车坐标系

不同的无人驾驶团队对于坐标系的定义可能不同,但这并不影响开发,只要团队内部达成一致即可。

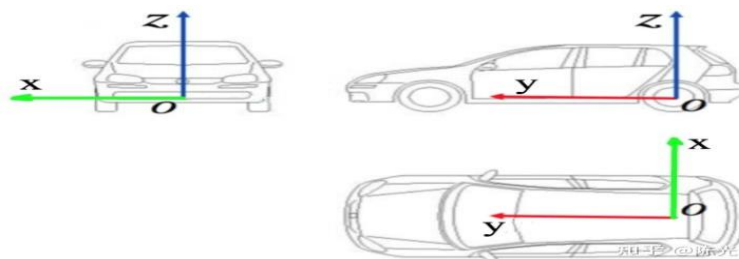
以百度 Apollo 提供的自车坐标系为例,自车坐标系的定义为:

z 轴 – 通过车顶垂直于地面指向上方

y 轴 – 在行驶的方向上指向车辆前方

x 轴 – 自车面向前方时,指向车辆右侧

车辆坐标系的原点在车辆后轮轴的中心,如下图所示。

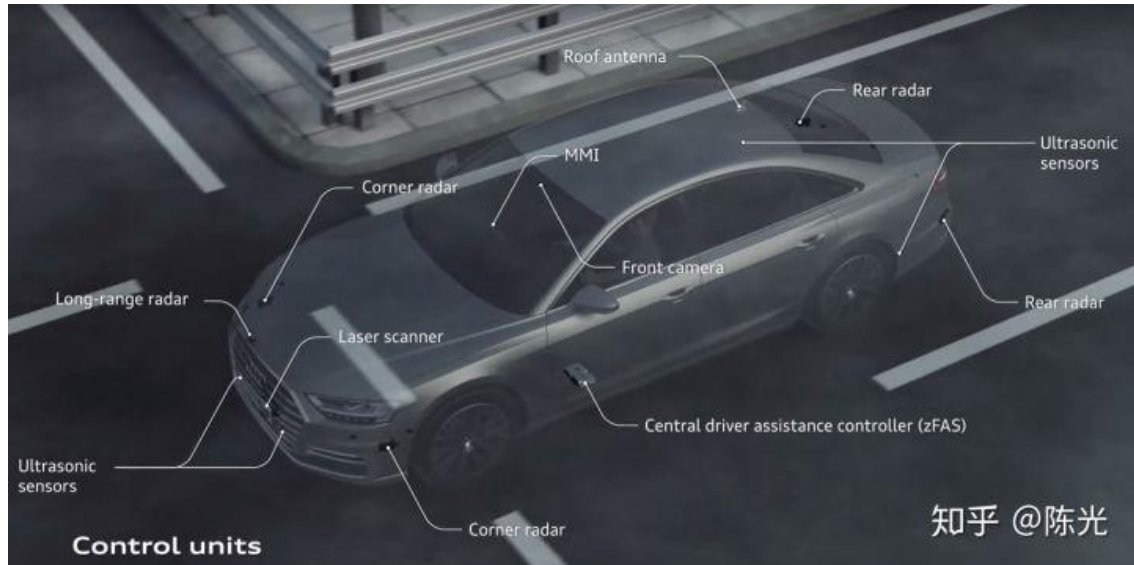


图片出处:

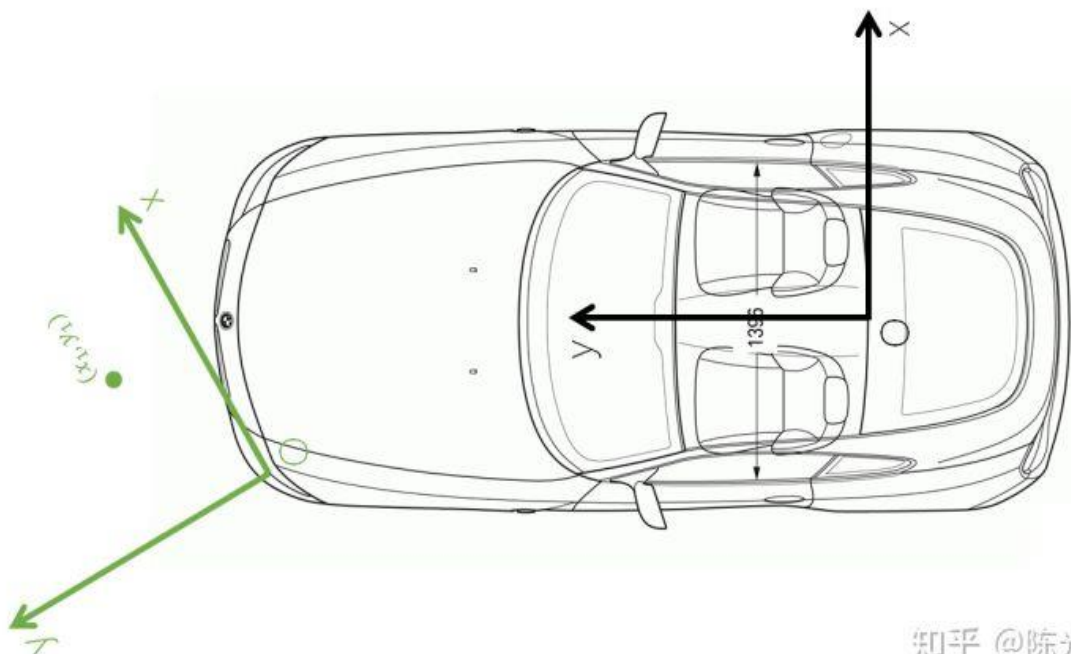
https://github.com/ApolloAuto/apollo/blob/master/docs/specs/coordination_cn.md

2 传感器坐标系与自车坐标系的转换

对于无人车的传感器安装位置，业内大同小异，比如奥迪 A8 的传感器配置如下图所示。



以安装在无人车左前方的角雷达（Corner radar）为例，进行后面的介绍，叫雷达的安装位置和坐标系如下图绿线所示。

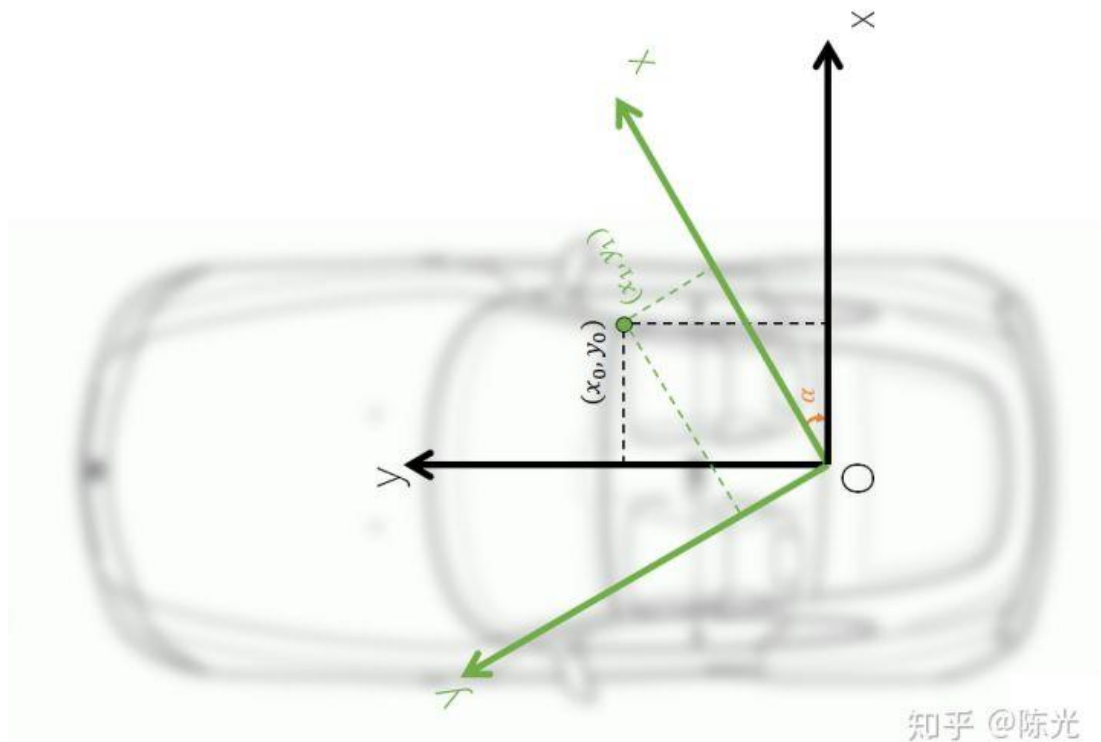


角雷达检测到的障碍物如图中的绿点所示，绿点在雷达坐标系下的坐标为 $(x1,y1)$ ，为了便于理解暂不加入 z 方向的坐标。

绿点转换到自车坐标系下需要经过一定的数学运算。基本思路是这样的：

平移

先将角雷达坐标系的 O 点平移到与自车坐标系的 O 点重合，此时 (x_1, y_1) 需要减去两个坐标系在 x 和 y 方向的距离。如下图所示。



旋转

在两个坐标系的 O 点重合后，将角雷达坐标系沿着 z 轴进行一定角度的旋转，这样 (x_1, y_1) 就转到了自车坐标系上。这个过程在数学上称为欧拉旋转。

坐标系的平移和旋转是两件相互独立的事情，先平移再旋转和先旋转再平移并不会影响最终的结果。

以上是感性的分析过程，下面我们将这整个过程在数学上实现。

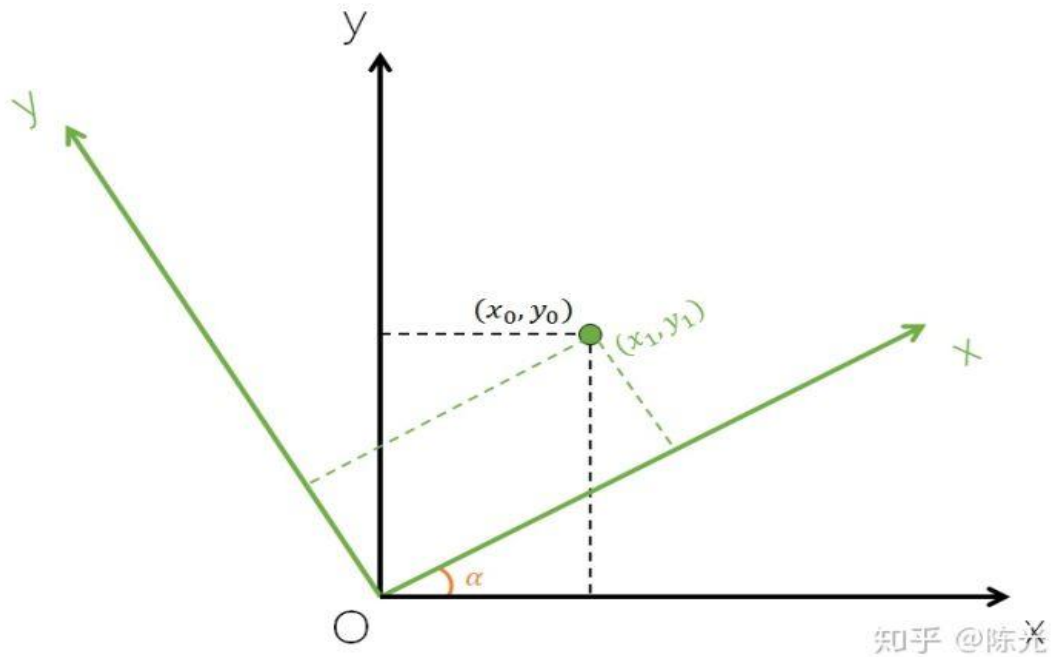
平移

平移步骤根据传感器安装位置和自车后轴的距离进行计算，仅仅是 XYZ 三个方向加减运算。

旋转

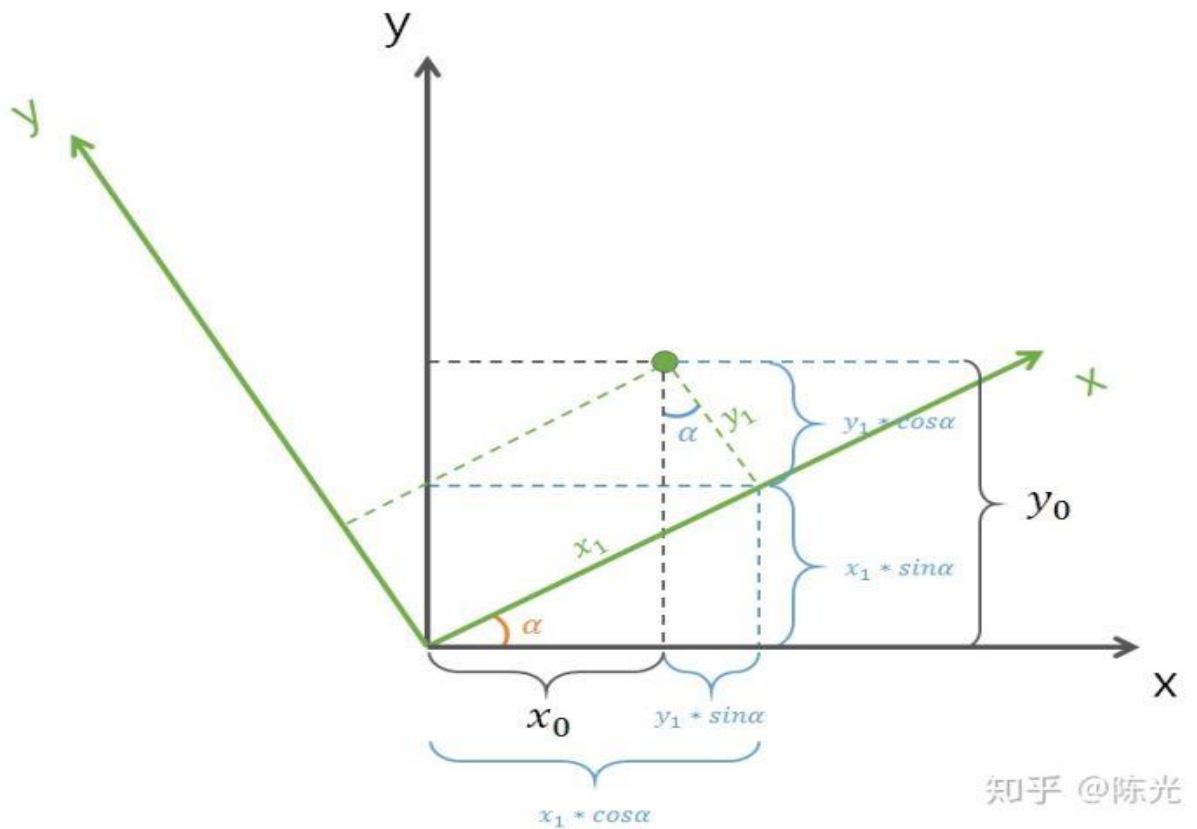
绕轴旋转需要引入角度，不是简单的加减运算，所以我们通过图示来推导一下。

先将两个坐标系变换到正常的视角，如下图所示：



障碍物在角雷达坐标系下的坐标为 (x_1, y_1) ，假设障碍物在自车坐标系下的坐标为 (x_0, y_0) ，需要根据安装角度 α （可测量），用 x_1, y_1, α 这三个已知量表示 x_0, y_0 ，求得他们的数学关系。

通过做辅助线进行计算，如下图蓝线所示所示：



几何关系可用以下两个等式表示：

$$x_0 = x_1 \cos \alpha - y_1 \sin \alpha$$

$$y_0 = x_1 \sin \alpha + y_1 \cos \alpha$$

使用矩阵表示，可以简化表达，用一个等式代替两个等式，是这样的。

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} * \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

由于这次旋转是绕 z 轴旋转，因此旋转前和旋转后的 z 值是保持不变的。

$$z_0 = z_1$$

将 z 方向的值也放到上面的等式中，即可得到。

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R_Z * \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

那就意味着，只要把角雷达采集到的障碍物坐标值与上面这个矩阵进行矩阵乘法运算，即可完成沿 Z 轴的旋转。在这里我们把这个矩阵叫做 Z 轴旋转矩阵 RZ，那必然还有沿着 X 轴和 Y 轴的旋转矩阵 RX 和 RY。

角雷达目标的坐标依次右乘这三个矩阵，就完成了沿着 Z 轴，Y 轴，X 轴的旋转，得到的结果就是自车坐标系下的坐标值了。即：

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = R_X * R_Y * R_Z * \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

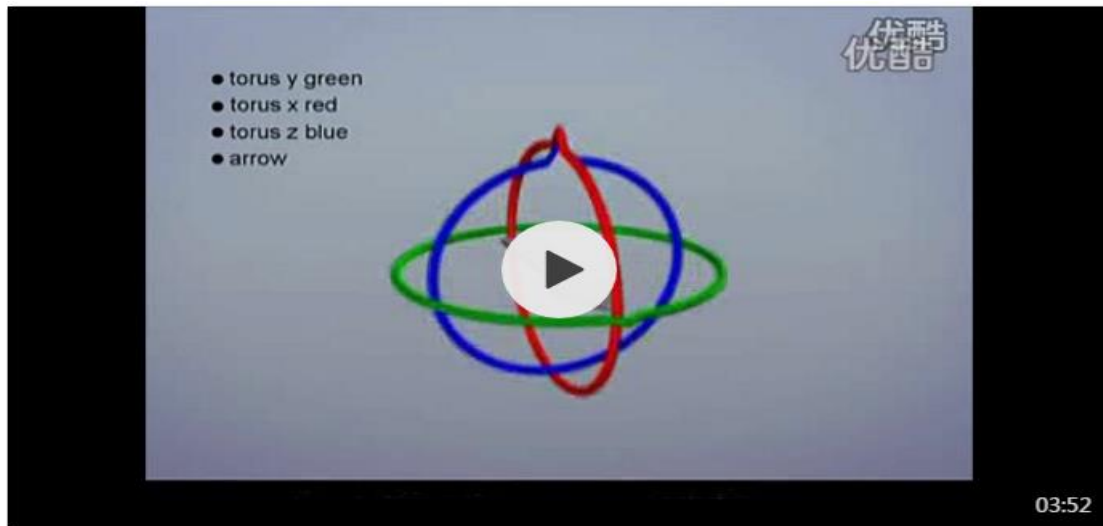
再加上一个平移的矩阵，就能够完整描述整个坐标转换的关系了。

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = R_X * R_Y * R_Z * \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + T_{XYZ}$$

不同的坐标系定义，会有不同的 RX，RY 和 RZ，因此需要根据实际情况计算旋转矩阵和平移矩阵。

3 欧拉旋转所存在的问题

以上过程也称为欧拉旋转，但是欧拉旋转会有一个不可避免的问题——万向锁。欧拉旋转的过程中在某些特殊情况时，会导致少一个维度。万向锁通过文字解释起来会有点困难，可以看一个讲解万向锁的小视频进行了解。



视频地址见原文：<https://zhuanlan.zhihu.com/p/41263701>

欧拉旋转所存在的问题——万向锁

为了解决欧拉旋转所带来的万向锁问题，业界引入了四元数。四元数除了解决万向锁的问题外，还能在一定程度上简化计算。因而百度 Apollo 也选择了四元数作为各个传感器安装位置和角度的存储介质。有关四元数的定义和使用方法这里不做详细讨论，可参看 Apollo 的 Calibration 模块。

4 结语

实际参与到开发中你才会发现，自己在大学学习的高数和线性代数等都是很重要的数学工具，而不是仅仅用来考试的。

以上就是有关无人驾驶技术中传感器到车体坐标系的坐标转换原理及数学推导，该原理不仅用在无人车领域，同时在机器人、无人机、三维建模等领域也得到了广泛使用。

好了(^o^)/~，这篇分享就到这啦。如果对坐标转换还有什么疑问，可以在评论区与我互动。

本文原载：知乎号“陈光”，作者授权转载。



临菲信息技术港



临菲信息技术港公众号



临菲学堂