# MonoLaneMapping

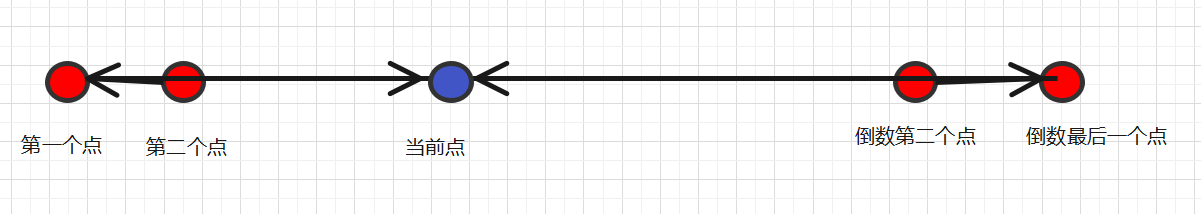
## 1、当前帧车道线每个点的噪声计算

其中，x和y分别是当前帧某车道线的坐标，剩下的其余参数都是人为设定的。

## 2、当前帧车道线计算每个点的阈值（get\_dist\_thd函数）

其中，除了xyz变量是根据当前帧拿到的，其余的变量全部是人为设定的。

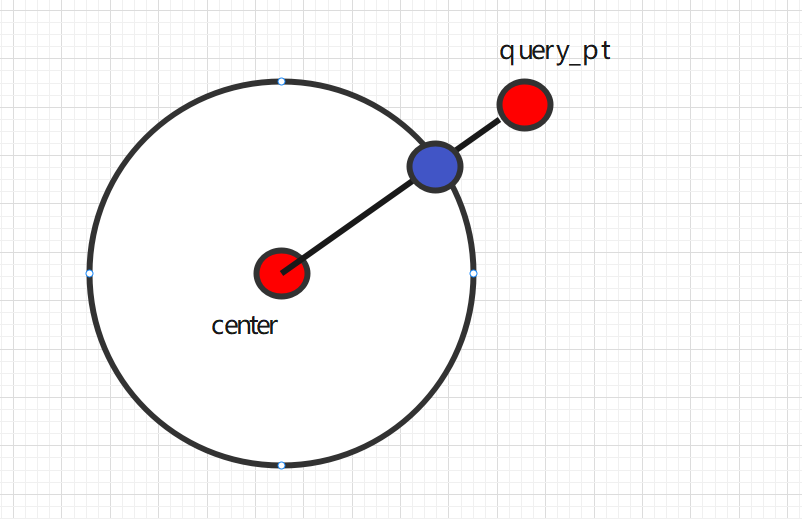
## 3、get\_pts\_to\_add



如果当前点在收尾控制点之间，那么这个点不会被加入，只有点不在两个控制点之间才会返回。上图的点就不会返回。

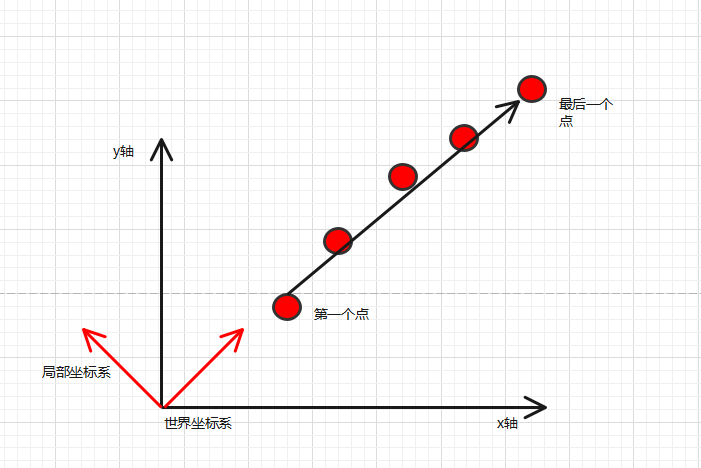
## 4、get\_nearest\_on\_circle

这个函数的根本目的是给定 圆的center和半径R和一个查询点，在圆上找到一个距离这个查询点最近：



基本方法是计算得到center到query\_pt构成的向量，并将这个向量单位化乘以半径R后加上center即可得到结果蓝色点。

**5、fitting**



先和地图中匹配上的当前帧车道线变换到世界坐标系下，如上图中的红色点，然后根据第一个点和最后一个点构成的向量与世界坐标系下的x轴计算角度，并构建局部坐标系。并将所有在世界坐标系下的点变换到局部坐标系下。

使用局部坐标系下所有点的x和y拟合得到x-y三次多项式线，x和z拟合得到x-z三次多项式线。

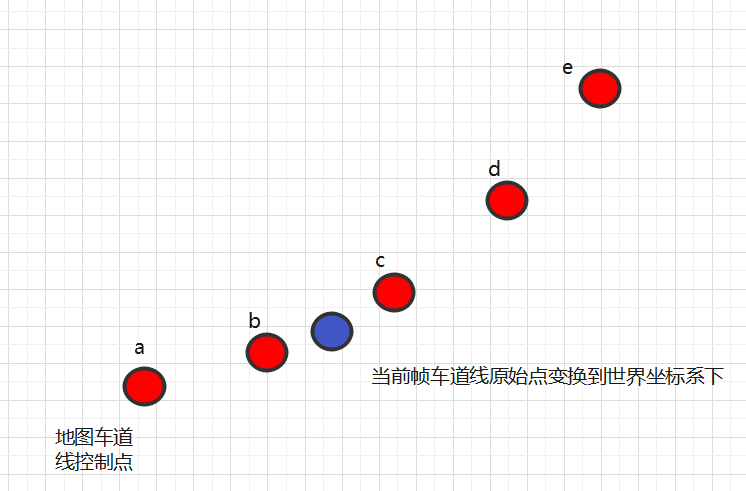
## 6、get\_skeleton

输入的是当前帧某条车道线的原始点和拟合过的多项式曲线表达式，以及地图中的最后一个控制点：

## 8、find\_footpoint

当前帧感知结果的车道线和地图中的车道线已经匹配，如何找到当前帧车道线每个点对应的样条曲线的u。

作者是这么实现的：假设当前帧车道线和地图中的车道线已经匹配；然后找到哪两个地图中车道线的控制点和当前帧的点最近。



如上图中，蓝色点为当前帧点，红色为地图车道线控制点。找到最近的两个控制点为b和c。使用a-d四个控制点构建CatmullRomSpline曲线，这个样条曲线的特点是a-d构建的样条曲线，作用域只在b到c。

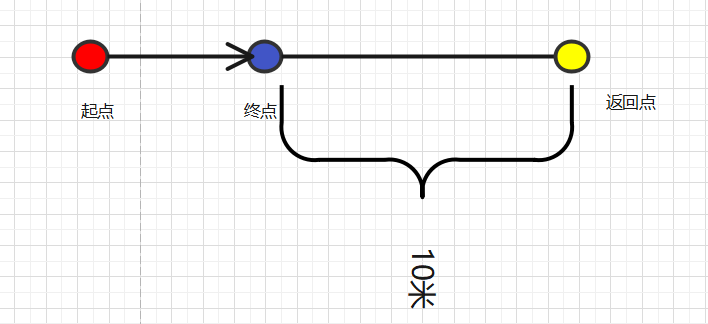
下一步是计算b-c作用域对应的u参数。我们知道作用域的起点是b，结束点是c。则：

则u = 0.25 + ratio \* (0.5 – 0.25)

## 9、get\_query

给定两个世界坐标系下控制点，和当前帧对应的多项式曲线参数。

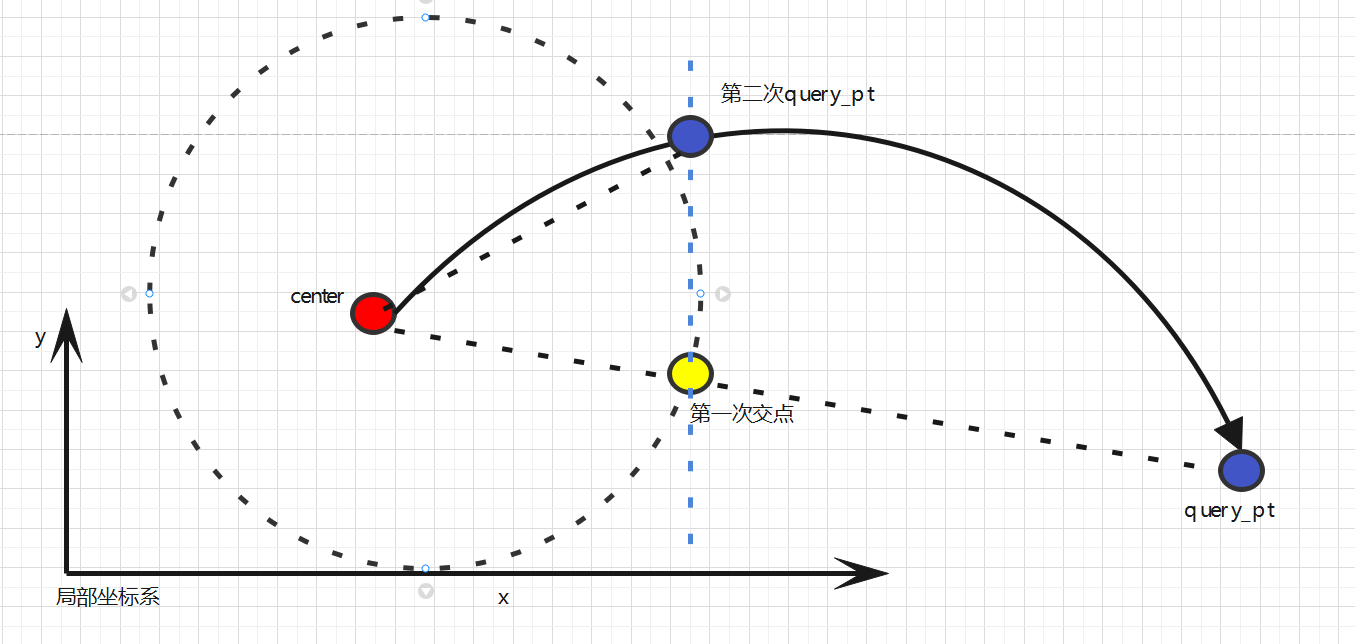
现将两个控制点变换到当前帧多项式曲线的局部坐标系。然后进行如下图的计算：



根据起点和终点构建向量，得到返回点（如上图黄色点所示）的x坐标，将此x坐标带入到当前帧曲线多项式中，然后将计算得到的局部坐标系点再转换到世界坐标系并返回。

## 10、get\_next\_node

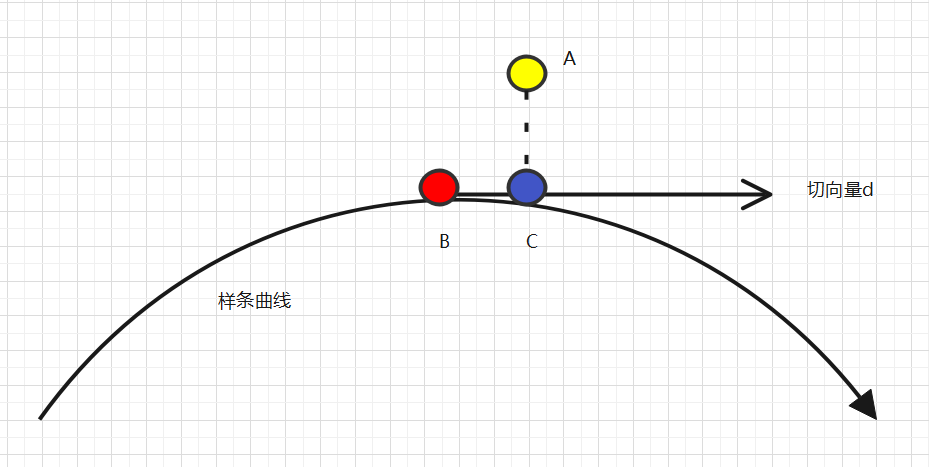
这个函数的本质功能是计算多项式曲线曲线和圆的交点，作者为了实现这个功能使用了数值分析的方法：



先计算center和query\_pt与指定半径圆的交点得到第一次交点，然后根据第一次交点的x坐标，带入到多项式曲线的x轴中得到第二次queery\_pt点的坐标，依次类推，直到两次相邻的query\_pt的距离小于0.02米立刻停止。

## 11、PosePointTangentFactor

点到切向量的距离计算方法：



其中A点为当前帧变换到世界坐标系下的点，含有待估计当前帧状态量。B为与之匹配的地图中的样条曲线上的点。d向量(3\*1向量)为B点对应的切向量，则点A到切向量的距离为：

上面的公式展开：

= -

= -

其中，为BC向量的距离。由此可知。作者之所以这么做是因为pA中含有待估计变量，必须将观测量和待估计量分离。