轨迹优化

硬约束与软约束

硬约束是指优化过程中的硬性限制条件，包括等式与不等式，比如：

$$
\begin{gathered}
minf(x) \\
s.t. g_i(x) = c_i \\
h_j(x) \geq d_j
\end{gathered}
$$

软约束是指在代价函数中增加带权惩罚项，让求解结果趋于最小化该项，比如：

$$minf(x) + \lambda_1 \cdot g(x) + \lambda_2 \cdot h(x)$$

如果优化问题同时存在硬约束和软约束，应该可以把硬约束的部分施加一个比较大的惩罚加到代价函数中，以软约束的形式存在。

施加硬约束

如果想要对一段路径施加不等式约束，比如约束曲线的边界或者速度加速度的上下限，使用之前常规的多项式形式是无法办到的。一种可行的方法是将路径离散成很多点，然后对这些点施加约束，另一种更优的方法就是使用贝塞尔曲线作为路径的假设。

贝塞尔曲线

之前使用的常见的多项式成为monomial多项式，现在引入一个称为Bernstein多项式，也就是贝塞尔曲线，它的形式为：

$$B_j(t) = c_j^0 b_n^0(t) + c_j^1 b_n^1(t) + \cdots + c_j^n b_n^n(t) = \sum_{i=0}^{n} c_j^i b_n^i(t)$$

$$b_n^i(t) = \binom{n}{t} \cdot t^i \cdot (1-t)^{n-i}$$

其中每个cj称为一个控制点，他是具有物理意义的。

值得注意的是，如果将贝塞尔曲线展开，他其实就是monomial多项式的一种特殊形式，他们之间具有映射关系：

$$p = M \cdot c$$

因此之前在monomial多项式中推导的结果也一样可以迁移到贝塞尔曲线中。

性质

1. 贝塞尔曲线只会经过第一个控制点和最后一个控制点，不会经过其他控制点。
2. 凸包：贝塞尔曲线会包含在由所有控制点组成的一个凸多边形称为凸包中
3. 贝塞尔曲线的导数形式跟原形式有非常密切的关系：

$$c_i' = n(c_{i+1} - c_i)$$

那么二阶导的形式就为：

$$c_i'' = n(n-1)[(c_{i+2} - c_{i+1}) - (c_{i+1} - c_i)]$$

以此类推。

4. 贝塞尔曲线的时间t定义域为[0,1]。这意味着实际应用中需要将时间归一化

应用

1. 边界约束

$$
\begin{gathered}
c_0^0=p_0, \\
n(c_0^1-c_0^0)=v_0, n(c_0^n-c_0^{n-1})=c_{n-1}'=v_n \\
n(n-1)[(c_2-c_1)-(c_1-c_0)]=a_0
\end{gathered}
$$

2. 连续性约束

$$
c_j^n=c_{j+1}^0
$$

3. 安全性约束（曲线的限制区域）

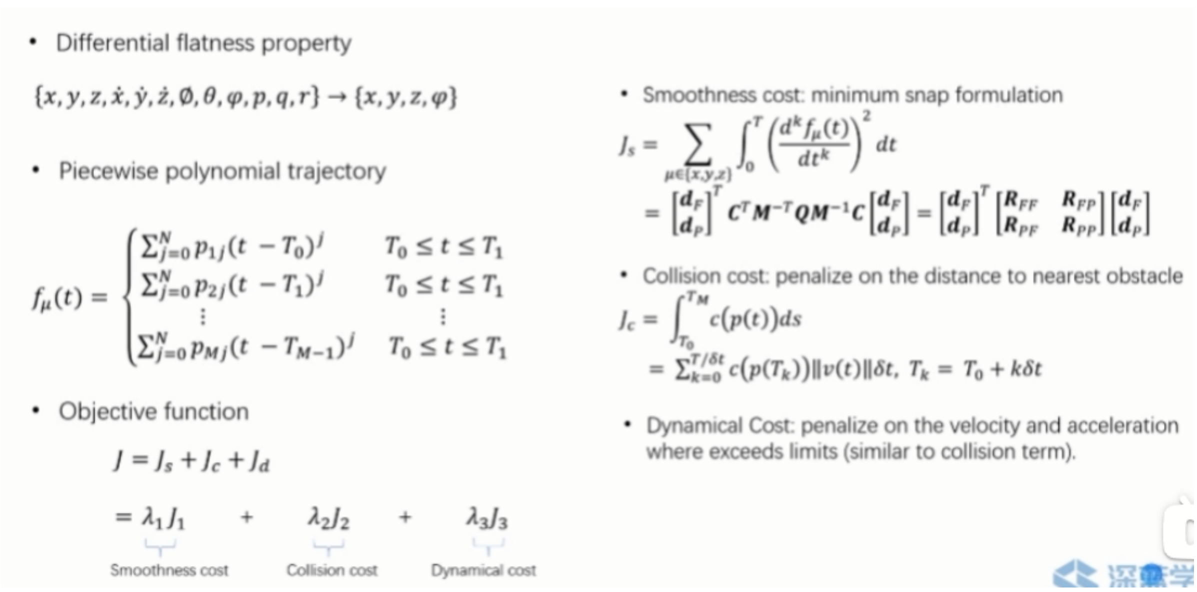
$$
a\leq c_j^i\leq b
$$

4. 动力学约束

$$
\begin{gathered}
v_m^-\leq n(c_j^i-c_j^{i-1})\leq v_m^+ \\
a_m^-\leq n(n-1)(c_j^i-2c_j^{i-1}+c_j^{i-2})
\end{gathered}
$$

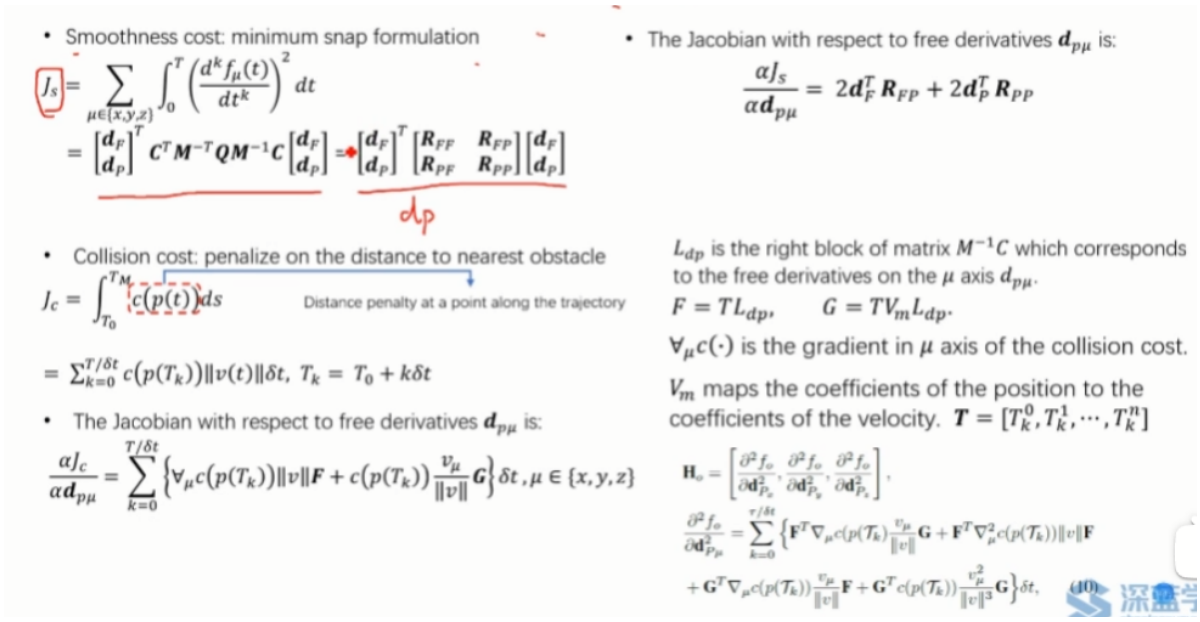
这样整个问题仍是一个典型的QP问题。

软约束



与之前不同的是代价函数中不止有平滑项，还加入了碰撞项和动力学项，其中碰撞项的代价是根据离障碍物的距离构建一个距离场计算的，也就是函数c。

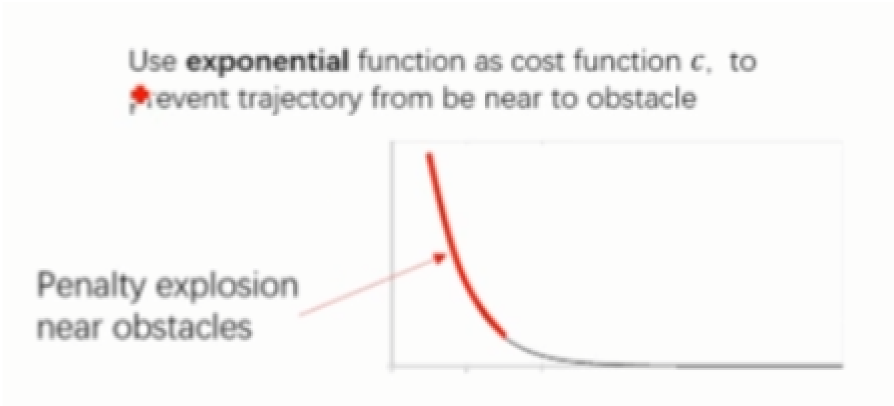
碰撞项和动力学项都是通过对ds进行积分计算的，原因在于如果对dt进行积分，可能会使结果偏向于让速度很大，时间很小来减小cost。这两项无法像平滑项一样可以化成一个很好的形式求解，只能通过离散很多点求和的形式进行数值近似。



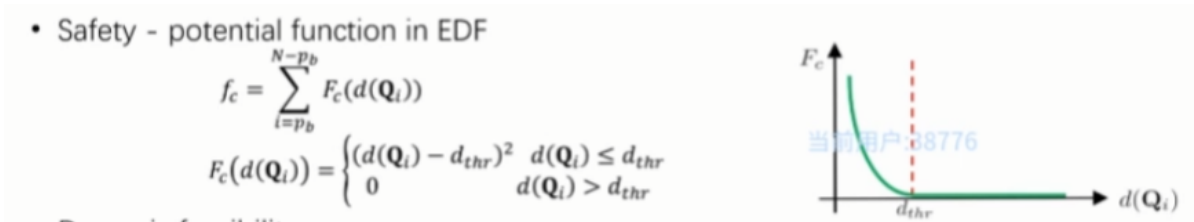
之后计算出各项梯度后，可以使用各种非线性优化的算法进行优化，代码上直接用ceres库就行。

距离场

距离场的构建有很多种办法，比如使用指数函数，当离障碍物很近的时候很大，远的时候很小。



还可以使用二次函数。



对于速度和加速度也可以用同样的方式进行场的构建。