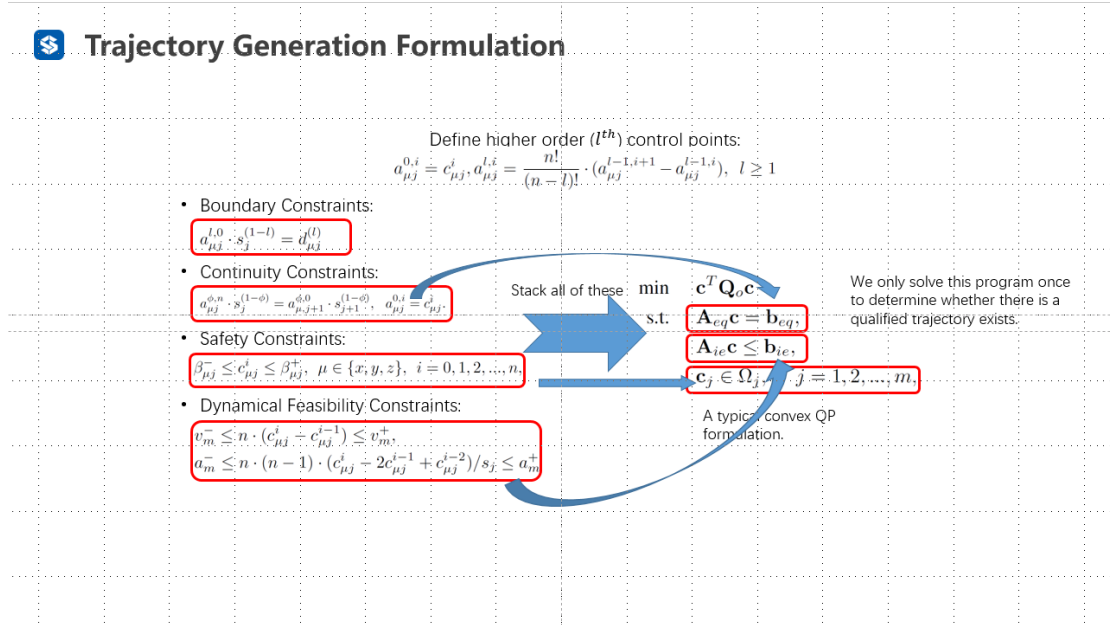


作业说明：

1. 本次作业是 Corridor-based and Bezier-based Minimum Snap Trajectory Optimization, 该问题可被形式化的描述为下图所示的公式：



2. 本次作业的代码流程和 hw5 十分类似，详细的说明可以观看课程后面的讲解说明。

对视频中没说清楚的点进行补充：

- a) 在计算 \mathbf{A}_{eq} , \mathbf{b}_{eq} 的时候，本次作业和 hw5 有区别，本次作业中没有对中间点的位置等式约束；
- b) Bezier curve 的时间是 [0,1]，作业中简单处理为全为 1，如果是自定义时间，需要对时间做一个比例缩放。

• 第6章: 软约束和硬约束下的轨迹优化(SOFT AND HARD CONSTRAINED TRAJECTORY OPTIMIZATION)	^
• 第1节: 轨迹优化方法(Trajectory Optimization)	
○ 任务21-1: 【课件】轨迹优化方法.pdf	↓
○ 任务21-2: 【视频】轨迹优化	▶
○ 任务21-3: 【视频】作业讲解说明	▶

3. 本次作业共需完成 4 个步骤:

- 步骤 1: 计算 Q_0 矩阵, 需要完成函数 `getQM()`;
- 步骤 2: 计算 A_{eq}, b_{eq} , 需要完成函数 `getAbeq()`;
- 步骤 3: 计算 A_{ieq}, b_{ieq} , 需要完成函数 `getAbieq()`;
- 步骤 4: 可视化轨迹

4. 本次作业中使用到的一些函数说明

- `M=getM(n)`: 返回 n 阶贝塞尔曲线系数和展开后的多项式系数的映射矩阵 M ;
- `Q' = NearestSPD(Q)`: 返回和 Q 矩阵距离最近的一个对称正定 (Symmetric Positive Definite) 矩阵 Q' 。目的是在把目标函数微调为一个凸函数, 保证得到的解为全局最优解。
- `plot_rect()`、`plot_line()`: 绘制矩形和线段的函数。