

- 小问 2 求解算法：B 样条轨迹优化
  - 1. 算法一：遗传算法 (Genetic Algorithm, GA)
    - 1.1 原理概述
    - 1.2 适用场景
    - 1.3 优缺点分析
    - 1.4 在本题中的应用
  - 2. 算法二：序列二次规划法 (Sequential Quadratic Programming, SQP)
    - 2.1 原理概述
    - 2.2 适用场景
    - 2.3 优缺点分析
  - 3. 推荐求解方法
    - 推荐算法：遗传算法 (GA)

## 小问 2 求解算法：B 样条轨迹优化

---

针对小问 2 建立的 B 样条曲线 + 动力学约束轨迹优化模型，该问题是一个典型的 **带约束非线性规划 (NLP)** 问题。目标函数（平滑度积分）是二次型或非线性的，约束条件包含微分不等式（速度、力矩限制）。推荐以下两种求解算法。

---

# 1. 算法一：遗传算法 (Genetic Algorithm, GA)

---

## 1.1 原理概述

遗传算法是一种模拟自然选择和遗传机制的进化算法。它将决策变量（B 样条控制点  $P_i$  和总时间  $T$ ）编码为染色体，通过 **选择 (Selection)**、**交叉 (Crossover)** 和 **变异 (Mutation)** 操作迭代搜索最优解。在处理约束时，通常采用 **罚函数法 (Penalty Function)**，将违反约束的解的适应度大幅降低，从而引导种群向可行域收敛。

## 1.2 适用场景

- **目标/约束非凸、非线性：**当优化问题存在多个局部最优解，或导数难以计算时。
- **黑盒模型：**如果动力学约束  $T_{dyn}$  极其复杂，难以写出显式梯度。

- 离散/混合变量：虽然本题变量连续，但 GA 对混合变量支持良好。

## 1.3 优缺点分析

- 优点：
  - 全局搜索能力强：不易陷入局部最优，适合寻找轨迹规划的全局最优解。
  - 鲁棒性高：对初值不敏感，只需给定搜索范围。
  - 易于并行化：种群中的个体评估互不干扰。
- 缺点：
  - 收敛速度慢：需要大量的函数评估次数。
  - 解的精度有限：通常只能找到最优解的邻域，难以达到高精度的数值收敛。
  - 参数敏感：种群大小、变异率等参数对结果影响较大。

## 1.4 在本题中的应用

- 编码：实数编码，基因向量  $[\theta_1, \theta_2, T]$ 。
  - 适应度： $F = \frac{1}{J + \alpha \sum \max(0, g_i(x))}$ ，其中  $J$  为平滑度， $g_i(x)$  为速度和力矩越界量。
  - 流程：随机生成 100 个轨迹样本 -> 评估平滑度和约束 -> 进化 50-100 代 -> 输出最佳轨迹。
- 

## 2. 算法二：序列二次规划法 (Sequential Quadratic Programming, SQP)

### 2.1 原理概述

SQP 是解决中小规模非线性约束优化公认最有效的确定性算法之一。其基本思想是：在每一个迭代点  $x_k$  处，将原非线性问题近似为一个 **二次规划 (QP)** 子问题（目标函数二次近似，约束线性近似），求解该 QP 子问题得到搜索方向  $d_k$ ，从而更新迭代点。

### 2.2 适用场景

- 光滑非线性问题：目标函数和约束函数二阶连续可微。
- 精度要求高：需要精确满足约束条件（如  $Error < 10^{-6}$ ）。

- **已有较好初值：**如果能大致猜测出轨迹形状，SQP 能快速收敛到附近的局部最优。

## 2.3 优缺点分析

- **优点：**

- **收敛速度快：**具有超线性收敛速度，比 GA 快很多。
- **精度高：**能严格满足等式和不等式约束。
- **理论成熟：**Matlab 的 `fmincon` 默认算法之一。

- **缺点：**

- **局部最优：**非常依赖初值猜测，容易陷入局部极小值。
  - **需要梯度信息：**需要计算目标和约束的 Jacobian 和 Hessian 矩阵（或其近似），若模型不可导则失效。
- 

## 3. 推荐求解方法

### 推荐算法：遗传算法 (GA)

#### 推荐理由：

1. **全局寻优需求：**轨迹优化问题通常存在多个局部极值（例如时间  $T$  和控制点形状的耦合可能导致多峰地形），SQP 容易陷入局部解，而 GA 能探索更广的空间。
2. **导数计算困难：**虽然 B 样条本身可导，但引入复杂的动力学约束（涉及力矩计算，可能包含非光滑项如摩擦力）后，梯度的解析推导较为困难，数值梯度可能不稳定。GA 无需梯度信息，实现更简单。
3. **罚函数处理灵活：**本题中“平均速度”是一个软约束（约等于），“最大力矩”是硬约束。GA 通过设计灵活的罚函数可以很好地平衡这些目标。
4. **题目特性：**作为数学建模竞赛，GA 是常用的启发式算法，易于编写代码并生成展示迭代过程的图表（如适应度下降曲线），便于论文写作。

**改进策略：**为了克服 GA 精度低的问题，可以采用 **混合策略**：先用 GA 跑 50 代得到一个较好的初值，再以该初值启动 SQP（或 Matlab `fmincon`）进行局部精细寻优。