# 机械臂自动刷牙系统算法框架与流程设计

## 一、系统总览

本系统实现了六轴机械臂在结构光相机和六维力传感器辅助下，对牙模进行自动清洁作业的全过程。系统支持自动与人工协同更换牙刷/冲牙器、实时力控、安全防护机制及清洁效果评估等功能，具有高度自动化与安全性，适用于智能牙科护理研究与实验平台。

## 二、整体算法流程图

流程图使用 Mermaid 格式描述如下：

graph TD  
 A[系统启动] --> B[建立 WebSocket 通信：机械臂/相机/力传感器]  
 B --> C{设备状态是否正常?}  
 C -- 否 --> Z1[触发错误提示 & 停止流程]  
 C -- 是 --> D[机械臂回归安全位置（避障路径规划）]  
 D --> E{是否更换末端工具?}  
 E -- 是 --> F1[松开夹具/等待人工更换]  
 F1 --> F2[夹紧夹具并启动标定流程]  
 E -- 否 --> G[牙模扫描 & 点云重建]  
 F2 --> G  
 G --> H[计算牙模与末端装置空间关系]  
 H --> I[用户界面选择清洁区域与参数]  
 I --> J[轨迹规划模块生成刷牙路径]  
 J --> K[执行刷牙轨迹 & 力控调整]  
 K --> L{过载或异常力?}  
 L -- 是 --> Z2[立即停止，触发保护机制]  
 L -- 否 --> M[刷牙完成，采集染色情况图像]  
 M --> N[清洁评估模块分析清洁力指数]  
 N --> O[流程完成]

## 三、核心模块类结构设计

以下为系统核心模块的 Python 类结构：

class 状态机:  
 def \_\_init\_\_(self): ...  
 def 启动系统(self): ...  
 def 运行(self): ...  
  
class 设备管理:  
 def 连接机械臂(self): ...  
 def 连接相机(self): ...  
 def 连接力传感器(self): ...  
 def 检查设备状态(self): ...  
 def 处理通信异常(self): ...  
  
class 机械臂控制:  
 def 回到安全位置(self): ...  
 def 避障路径规划(self, 目标点): ...  
 def 松开夹具(self): ...  
 def 夹紧夹具(self): ...  
 def 执行轨迹(self, 轨迹): ...  
  
class 标定模块:  
 def 末端标定(self): ...  
 def 手眼标定(self): ...  
  
class 视觉扫描:  
 def 扫描牙模(self): ...  
 def 构建点云(self): ...  
 def 计算牙模坐标关系(self): ...  
  
class 轨迹规划:  
 def 生成刷洗轨迹(self, 区域, 参数): ...  
  
class 力控监测:  
 def 实时监测力(self): ...  
 def 检测过载(self): ...  
 def 动态调整(self): ...  
  
class 清洁评估:  
 def 计算清洁指数(self, 刷前图, 刷后图): ...  
  
class 图像采集模块:  
 def 拍摄图像(self, 名称): ...  
 def 配准图像(self, img1, img2): ...  
  
class 数字孪生同步:  
 def 更新位姿(self, pos): ...  
 def 推送数据(self, 力, 状态): ...

## 四、主函数（执行逻辑）

def 主函数():  
 初始化所有模块...  
  
 if not 所有设备连接成功:  
 错误处理()  
 return  
  
 if not 设备状态正常:  
 错误处理()  
 return  
  
 回归机械臂初始位姿  
  
 if 用户选择更换末端工具:  
 松开夹具 -> 人工更换 -> 夹紧夹具 -> 末端标定  
  
 扫描牙模 -> 构建点云 -> 计算空间关系  
  
 用户选择区域与参数 -> 生成轨迹 -> 执行刷洗  
  
 while 执行轨迹:  
 力 = 实时监测  
 if 过载或异常:  
 停止动作 -> 触发保护机制  
 break  
 动态调整轨迹  
  
 拍摄刷前/刷后图像 -> 清洁评估

## 五、系统状态图

状态图使用 Mermaid 格式描述如下：

stateDiagram-v2  
 [\*] --> 等待设备  
 等待设备 --> 设备异常: 通信失败/状态异常  
 等待设备 --> 待准备: 设备正常  
 待准备 --> 等待装配: 需更换末端工具  
 等待装配 --> 标定中: 装配完成  
 待准备 --> 标定中: 无需更换工具  
 标定中 --> 扫描中: 标定成功  
 扫描中 --> 执行中: 扫描 & 点云完成  
 执行中 --> 故障: 力过载/冲突  
 执行中 --> 清洁评估: 正常完成  
 故障 --> [\*]  
 清洁评估 --> 完成  
 完成 --> [\*]

## 六、数据流程图（简化）

数据流使用 Mermaid 格式描述如下：

graph LR  
 UI界面 -->|选择区域+参数| 轨迹规划  
 结构光 -->|牙模图像| 点云构建 --> 几何建模  
 六维力传感器 --> 力控模块 -->|调整反馈| 机械臂控制  
 几何建模 -->|牙模坐标| 轨迹规划 --> 机械臂控制  
 机械臂控制 -->|执行轨迹| 刷牙动作  
 图像采集模块 --> 清洁评估 --> UI界面

## 七、后续扩展建议

为增强系统适应性与可靠性，可在当前架构基础上进一步拓展以下模块：  
- UI系统对接：与上层用户控制平台进行接口集成，实现参数可视化配置与实时进度反馈；  
- 异步调度机制：引入任务队列与事件监听机制，提升模块间协同与响应效率；  
- 安全策略子系统：加强力控异常处理、软限位保护与误触容错机制的设计。