

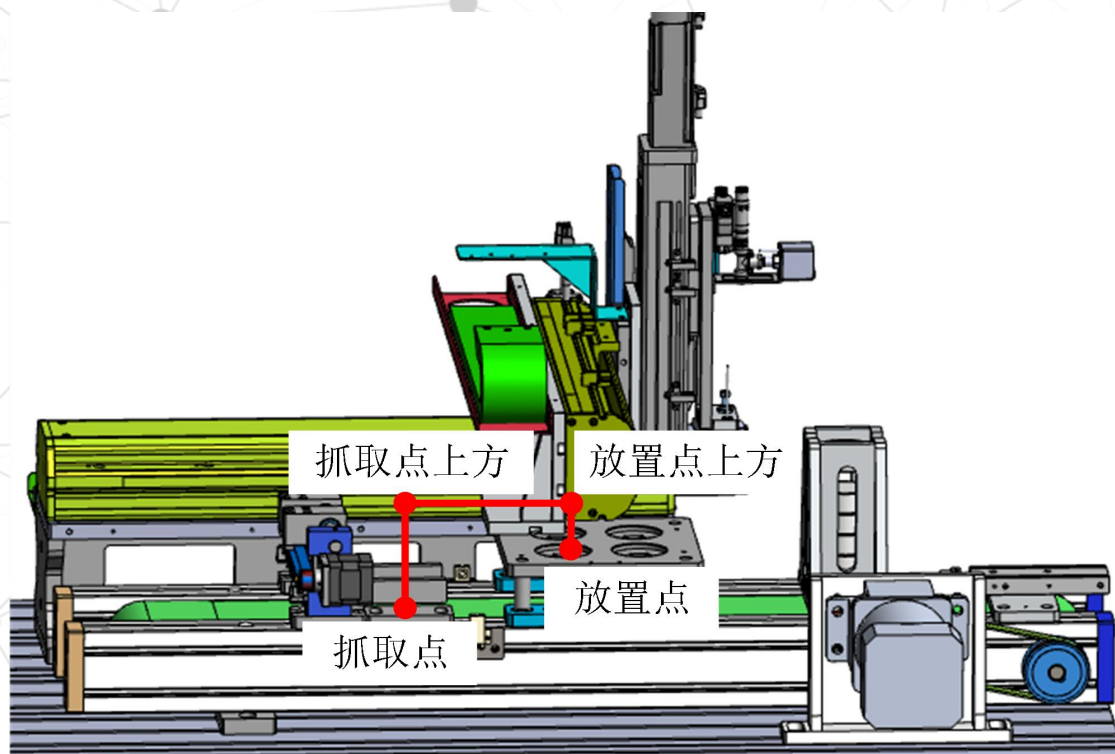
XYZ平台的物料搬 运编程





任务提出

了解综合供料系统的工艺流程，学会测定物料搬运过程中特征点的坐标，利用运动控制器编程实现指定轨迹的物料搬运。





一、XY平台回零位

1. 创建VS应用程序;
2. 调用运动控制器函数库;
3. 编写X轴和Y轴的回零运动程序;

注意：为防止回零过程中发生碰撞，保证回零过程的安全性，XYZ模组回零的顺序应为Z轴→Y轴→X轴。

4. 完成XY平台回零运动，注意回零运动结束，要对XY轴进行位置清零操作。





二、建立坐标系

首先建立三维坐标系，
将物理上的XYZ两个
电机轴与软件计算中的
XYZ轴对应起来，
获得一个以机械固定
位置为原点的坐标系。

坐标系建立后，
末端工具的运动位置
均可以由此坐标系表
示。

```
void CCoordinateDlg::OnButtonInitialCoordinate()
{
    // TODO: Add your control notification handler code here
    short sRtn;
    TCrdPrm crdPrm;

    memset(&crdPrm,0,sizeof(crdPrm)); // 将结构体变量初始化为0
    crdPrm.dimension = 3;               // 建立三维的坐标系
    crdPrm.synVelMax = 50;               // 坐标系的最大合成速度是: 50 pulse/ms
    crdPrm.synAccMax = 1;                // 坐标系的最大合成加速度是: 1pulse/ms^2
    crdPrm.evenTime = 50;                // 坐标系的最小匀速时间为50ms
    crdPrm.profile[0] = 1;               // 规划器1对应到X轴
    crdPrm.profile[1] = 2;               // 规划器2对应到Y轴
    crdPrm.profile[2] = 3;               // 规划器3对应到Z轴

    crdPrm.setOriginFlag = 1;            //表示需要指定坐标系的原点坐标的规划位置
    // 坐标系的原点坐标的规划位置为 (0, 0,0)
    crdPrm.originPos[0] = 0;
    crdPrm.originPos[1] = 0;
    crdPrm.originPos[2] = 0;
    //根据设定的坐标系参数，建立1号坐标系
    sRtn = GT_SetCrdPrm(1,&crdPrm);
}
```



三、物料搬运程序

建立坐标系后，可直接使用插补指令控制末端工具的移动，如果需要末端工具进行一个直线移动，可直接使用运动控制卡的直线插补指令。通过直线插补指令，可以使末端工具在当前位置直线运动到工作空间中的另一指定坐标位置。

注意：由于回零参数设置的差异和设备本身的机械误差，即使是同一种设备，特征点的坐标也会存在差异，因此不可直接使用程序中的特征点坐标。





三、物料搬运程序

```
void CCoordinateDlg::OnButtonLinearMotion()
{
    short sRtn; // 指令返回值变量
    short run; // 坐标系运动状态查询变量
    long segment; // 坐标系运动完成段查询变量

    sRtn = GT_CrdClear(1, 0); //首先清除坐标系1的FIFO0缓存区中的数据
    sRtn = GT_LnXYZ (          //直线插补，从回零位走到物料抓取点上方
        1,                    // 该插补段的坐标系是坐标系1
        -311951, -125134, 0, //该插补段的终点坐标(-311951, -125134,0)
        50,                   // 该插补段的目标速度：50pulse/ms
        0.1,                  // 插补段的加速度：0.1pulse/ms^2
        0,                    // 终点速度为0
        0);                   // 向坐标系1的FIFO0缓存区传递该直线插补数据

    // 走到物料抓取点
    sRtn = GT_LnXYZ(1, -311951, -125134, -160657, 50, 0.1, 0, 0);
    // 走到物料点上方
    sRtn = GT_LnXYZ(1, -311951, -125134, 0, 50, 0.1, 0, 0);
    // 走到物料放置点上方
    sRtn = GT_LnXYZ(1, -131521, -133400, 0, 50, 0.1, 0, 0);
}
```





三、物料搬运程序

```
// 走到物料放置点
sRtn = GT_LnXYZ(1, -131521, -133400, -34757, 50, 0.1, 0, 0);
// 走到物料放置点上方
sRtn = GT_LnXYZ(1, -131521, -133400, 0, 50, 0.1, 0, 0);
// 搬运完成, 回零位
sRtn = GT_LnXYZ(1, 0, 0, 0, 50, 0.1, 0, 0);
// 启动坐标系1的FIFO0的插补运动
sRtn = GT_CrdStart(1, 0);
// 查询坐标系1的 FIFO0 插补运动坐标系状态
sRtn = GT_CrdStatus(1, &run, &segment, 0);
do // 等待运动完成
{
    // 查询坐标系1的FIFO的插补运动状态
    sRtn = GT_CrdStatus(
        1, // 坐标系是坐标系1
        &run, // 读取插补运动状态
        &segment, // 读取当前已经完成的插补段数
        0); // 查询坐标系1的FIFO0缓存区
} while (run == 1); // 坐标系在运动, 查询到的run的值为1
}
```



谢谢观看

