**插补运动**

插补运动是以微小直线段精确地逼近轨迹，实现精准的轨迹控制的一种运动。插补运动有许多种类，如：直线插补、圆弧插补、螺旋线插补等。SC2-C系列PLC支持插补运动。SC2-C轴控制程序应放到HSIOTask任务下调用。

本文介绍SC2-C直线插补运动和圆弧插补运动的参数配置与运动实现方法。

1. 添加库

使用插补运动指令须先添加插补运动库。操作步骤如下：①双击打开库管理器；②点击库管理器；③点击添加库；④点击高级；⑤搜索LS\_IpoLib；⑥选择LS\_IpoLib库，完成添加。如图1所示。

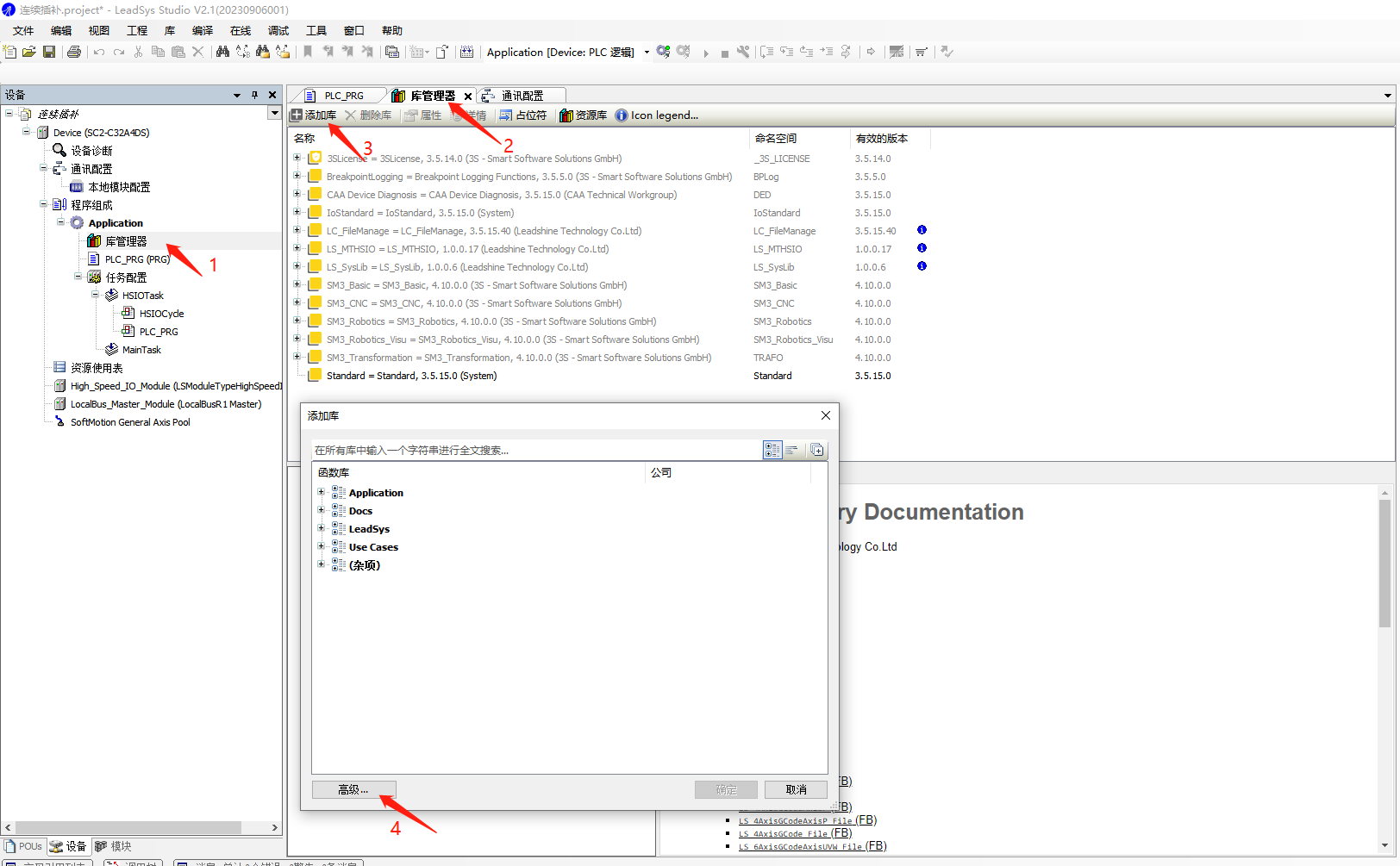
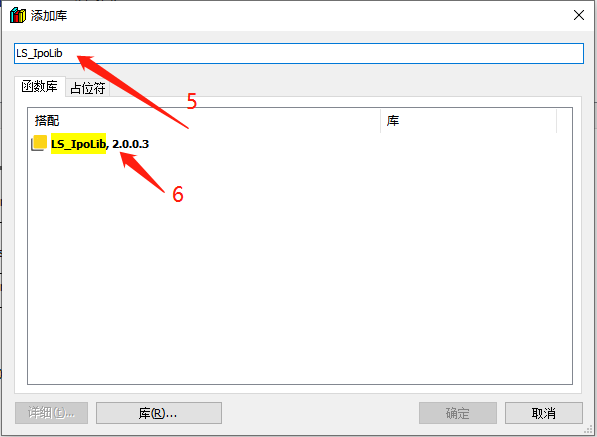


图1 添加插补运动库

1. 直线插补

2.1直线插补指令

直线插补指令有2~6轴的直线插补，其中第4~6轴为给随前三轴运动，下面以三轴直线插补指令介绍直线插补指令。

LS\_3AxisLine (Axis0:=轴号0, Axis1:=轴号1, Axis2:=轴号2, Exec:= 启动信号,

Halt:=暂停信号, Stop:=停止信号, IpoCycle:=插补周期,

PosMode:=坐标模式, EndPos:=终点坐标, VelocityMode:=速度曲线模式,

Velocity:=插补运动最大速度, Acceleration:=加速度, Deceleration:=减速度,

Jerk:=加加速度, Vel=>插补速度, Done=>插补运动结束, Busy=>插补运动中,

CommandAborted=>运动被终止, CommandHalt=>运动暂停,

Error=>指令出错, ErrorID=>错误代码 );

指令中：轴号0、轴号1、轴号2是参与三轴直线插补运动的轴号。

插补周期须设置为HSIO\_Task任务相等周期。

坐标模式为0时，终点坐标是绝对坐标，为1时是相对坐标。

速度曲线模式为0时，速度曲线为梯形速度曲线；为1时是S形速度曲线；为3时是二次速度曲线。

2.2直线插补例程

设置坐标模式为绝对坐标，即PosMode = 0，终点坐标为[1000,700,400]；速度曲线为梯形，即VelocityMode = 0 ；插补运动的最大速度为1200 unit/s，加速度为5000unit/s2，减速度为10000unit/s2。程序代码如下代码。

|  |
| --- |
| PROGRAM PLC\_PRG  VAR  Power: ARRAY [0..2] OF MC\_POWER;  LineXYZ: LS\_3AxisLine;  Linestart: BOOL;  Linedistance: ARRAY [0..2] OF LREAL:= [1000,700,400];  Linedone: BOOL;  Vel\_0: LREAL;  END\_VAR |

|  |
| --- |
| //轴使能  Power[0](Axis:= X, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= 1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  Power[1](Axis:= Y, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= 1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  Power[2](Axis:= Z, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= 1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  (\*直线插补  绝对坐标模式，终点坐标为[1000,700,400]，速度曲线为梯形；  插补运动的最大速度为1200 unit/s，加速度为5000unit/s2，减速度为10000unit/s2  \*)  LineXYZ(Axis0:=X, Axis1:=Y, Axis2:=Z, Exec:=Linestart, Halt:= , Stop:= , IpoCycle:= 4000,  PosMode:=0, EndPos:=Linedistance, VelocityMode:=0, Velocity:=1200, Acceleration:=5000,Deceleration:=10000, Jerk:=, Vel=>Vel\_0, Done=>Linedone, Busy=> ,  CommandAborted=> , CommandHalt=> , Error=> , ErrorID=> ); |



图2 三轴直线插补执行结果

从图2中可看出，在启动三轴直线插补指令之前，X、Y、Z轴的当前位置均为0；当变量startLine有FALSE变为TRUE时，开始执行直线插补；插补指令结束后，X、Y、Z轴的位置分别是1000、700、400 unit，与指定的终点坐标一致，过程中最大速度为1200 unit/s。

1. 圆弧插补

圆弧插补指令包括二轴圆弧插补和三轴圆弧插补。

3.1 平面两轴圆弧插补指令

两轴圆弧插补指令如下。

LS\_2AxisCircle( Axis0:=轴号0, Axis1:=轴号1, Exec:=启动信号, Halt:=暂停信号,

Stop:=停止信号, IpoCycle:=插补周期, PosMode:=坐标模式,

ArcIpo\_dir:=圆弧方向, Circle\_Mode:=圆弧模式, StopAngle:=圆心角,

CircleRadiual:=半径, AuxPos:=辅助坐标, EndPos:= 终点坐标,

RunCycle:=圆弧循环次数, VelocityMode:=速度曲线模式,

Velocity:=插补运动最大速度, Acceleration:=加速度, Deceleration:=减速度,

Jerk:=加加速度, Vel=>插补速度, Done=>插补运动结束, Busy=>插补运动中,

CommandAborted=>运动被终止, CommandHalt=>运动暂停,

Error=>指令出错, ErrorID=>错误代码 );

指令中的重要参数有：

坐标模式：为0时终点坐标是绝对坐标，为1时是相对坐标。

圆弧模式：0为三点圆弧模式；1为圆心终点模式；2为终点半径模式；3为圆心角模式；4为三点整圆模式。

圆弧方向：圆弧模式为1、2、3时需要设置该值。为0时为逆圆，为1时是顺圆。

圆心角：圆弧模式为3时需要设置该值。圆弧起点到终点对应的圆心角，单位为度，取值范围为0~360.0。

半径：圆弧模式2需要设置该值。该值为负表示圆心角小于180度，半径为正表示圆弧角大于180度。

辅助坐标：圆弧模式0、1、3需要设置该值。圆弧模式0设置圆弧轨迹上出起点和中点外的任意一点坐标。圆弧模式1、3设置圆心坐标。

终点坐标：存储终点坐标的数组。

圆弧循环次数：绝对坐标模式下，0为小于360度的圆弧；n为圆循环n次，从终点开始计算循环次数。

相对坐标模式下不支持圆循环，且该参数要设为1。

速度曲线模式：0为梯形速度曲线；1为S形速度曲线；2为二次速度曲线。

注：速度、加速度、减速度、加加速度均指插补运动的矢量速度。

3.2 空间三轴圆弧插补指令

三轴圆弧插补指令如下。

LS\_3AxisCircle( Axis0:=轴号0, Axis1:=轴号1, Axis2:= 轴号2, Exec:= 启动信号,

Halt:=暂停信号, Stop:=停止信号, IpoCycle:=插补周期, PosMode:=坐标模式,

Circle\_Mode:=圆弧模式, AuxPos:=辅助坐标, EndPos:= 终点坐标,

VelocityMode:=速度曲线模式, Velocity:=插补运动最大速度,

Acceleration:=加速度, Deceleration:=减速度, Jerk:=加加速度,

Vel=>插补速度, Done=>插补运动结束, Busy=>插补运动中,

CommandAborted=>运动被终止, CommandHalt=>运动暂停,

Error=>指令出错, ErrorID=>错误代码 );

三轴圆弧插补指令中的参数定义和两轴圆弧插补指令的基本相同，但圆弧模式只有三点圆弧模式。

3.3 圆弧插补例程

使用二轴圆弧插补指令三点圆弧模式XY轴走一段圆弧轨迹。设置坐标模式为相对坐标，终点坐标为[4000,-3300]，中间点坐标为[5800,4200]；速度曲线为梯形；插补运动的最大速度为2000 unit/s，加速度和减速度都为5000unit/s2。程序代码如下代码。

|  |
| --- |
| PROGRAM PLC\_PRG  VAR  Power: ARRAY [0..2] OF MC\_POWER;  circleXY: LS\_2AxisCircle;  EndPosition: ARRAY[0..1] OF LREAL := [4000,-3300]; // 终点坐标  AuxPosition: ARRAY[0..1] OF LREAL := [5800,4200]; // 中间点坐标  circleStart: BOOL;  circleDone: BOOL;  Vel\_1: LREAL;  END\_VAR |

|  |
| --- |
| //轴使能  Power[0](Axis:= X, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= 1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  Power[1](Axis:= Y, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= 1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  Power[2](Axis:= Z, Enable:= 1, bRegulatorOn:= 1, bDriveStart:= 1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  (\*圆弧插补  相对坐标模式，三点圆弧模式，终点坐标为[4000,-3300]，中间点坐标为[5800,4200]，速度曲线为梯形；  插补运动的最大速度为2000 unit/s，加速度为5000unit/s2，减速度为5000unit/s2  \*)  circleXY( Axis0:=X, Axis1:=Y, Exec:=circleStart, Halt:= , Stop:= , IpoCycle:= 4000, PosMode:=1,ArcIpo\_dir:=, Circle\_Mode:=0, StopAngle:= , CircleRadiual:= ,  AuxPos:=AuxPosition, EndPos:=EndPosition, RunCycle:=1, VelocityMode:=0,  Velocity:=2000, Acceleration:=5000, Deceleration:=5000, Jerk:= ,  Vel=>Vel\_1, Done=>circleDone, Busy=> , CommandAborted=> ,  CommandHalt=> , Error=> , ErrorID=> ); |

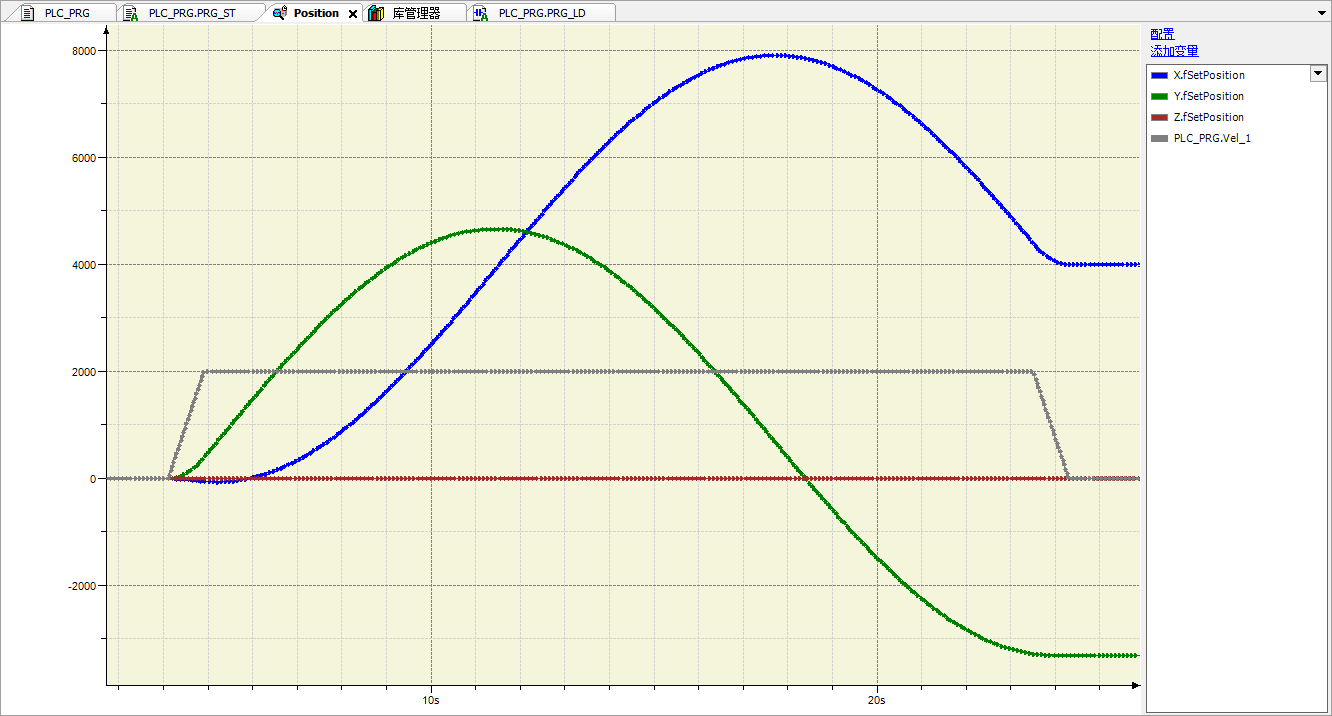


图3 二轴圆弧插补执行结果

从图3中可看出，在启动二轴圆弧插补指令之前，X、Y轴的当前位置均为0；当变量circleStart有FALSE变为TRUE时，开始执行圆弧插补；插补指令结束后，X、Y轴的位置分别是4000、-3000 unit，与指定的终点坐标一致，过程中最大速度为2000 unit/s。