

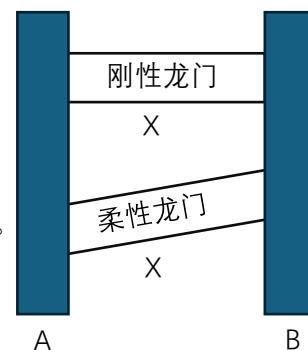
730W 龙门功能介绍 V1.0

一. 730W 龙门主要功能简介

- 主从轴协调工作，系统仅需给主轴（A 轴）发送位置指令。
- 协调停止功能（防止发生警报时的机械损坏）。其中任何轴发生故障，所有轴立即停止
- 轴间位置偏差过大检出功能（检测机械的扭转，防止机械损坏，有利于预防维护）。
- 位置补偿表功能（抑制机械误差、无效转矩等）。
- 增量式编码器和绝对值编码器都可适用，增量模式使用龙门功能必须先执行回原

二. 龙门介绍

- 刚性龙门：龙门轴之间刚性连接，机械耦合没有弹性。
不允许龙门轴有较大的龙门同步偏差。
- 柔性龙门：龙门轴之间柔性连接，机械耦合具有一定弹性。
允许龙门轴存在一定的同步偏差。



三. 使用条件

调试软件 1.24 版本，驱动器主机固件 5117 版本

- 轴：该功能固定用于 A B 轴，固定 A 轴为主轴，B 轴为从轴不可更改。
- 控制方式：位置控制。

四 . 相关参数

参数编号	参数名称	单位	默认值	范围值
PN6C0.0	龙门功能开关	-	0	0-1
PN6C1.0	位置补偿表功能	-	0	0-1
PN6C1.1	同步补偿功能	-	0	0-1
PN6C2	龙门协调停止模式	-	0	0-1
PN6C3	龙门轴间位置偏差过大警告值	%	100	10-100
PN6C4	龙门轴间位置偏差过大警报值	指令单位	1048576	1048576-1073741823
PN6C7	龙门同步位置环比例增益	0.1%S	400	10-2000
PN6C8	龙门同步位置环积分时间	0.1ms	2000	0-50000
PN6C9	龙门同步速度限幅	0.1%	20	0-1000
PN6CB	龙门协调停止完成速度阈值	rpm	15	0-1000

PN6C1.1: 开启同步补偿功能，设置 PN6C7,PN6C8 可减小主轴和从轴的跟随误差

PN6C2: 设置协调停止功能

- 协调停止功能模式 1: 协调停止轴将警报发生轴的反馈位置作为目标位置进行位置控制。
- 协调停止功能模式 2: 当 A 轴或 B 轴发生警报时，协调停止轴也会变为伺服 OFF 状态。

Pn6CB: 龙门协调停完成速度阈值，到达此速度退出协调停的功能。

Pn6C4: 设置轴间位置偏差过大警报值，单位指令单位。

Pn6C3: 设置轴间位置偏差过大警告值，Pn6C4 的百分比。

五 . 使用示例

只需要给主轴 A 轴发送命令即可。执行回原以后，执行定位控制。

1，驱动器电机信息

驱动器信息	
名称	值
设备唯一序列号	123456789AB...
型号	SV730WBN-C...
功率[W]	400
驱动器类型	-
主机程序(core1)版本	51170000 H
从机程序(core2)版本	071D H
Boot版本	4718 H
Prm版本	5117 H
Ecat Xml版本	0000 H
Ecat ESC版本	0007 H

电机与编码器	
名称	轴A
电机型号	X6MH040A-N2LD
电机序列号	31424000661
电机功率[W]	400
电机额定电流[A]	2.9
电机额定转速[rpm]	3000
电机额定扭矩[N·m]	1.27
编码器型号	HROA23D35M6T2C
编码器分辨率	23
	轴B
	X6MH075A-N2LD
	51723209981
	750
	5.3
	3000
	2.39
	HROA23D35M6T2C
	23
	轴C
	-
	-
	-
	-
	-
	-
	-
	轴D
	-
	-
	-
	-
	-
	-
	-

2，驱动器参数设置

▼ Pn6C0	龙门功能开关	-	0001	0001
0	龙门功能总开关		1H :使能	1H :使能
2~3	测试		0	0
▼ Pn6C1	龙门从轴功能开关	-	0000	0010
0	位置补偿表功能		0H :不使能	0H :不使能
1	同步补偿功能		0H :不使能	1H :使能
Pn6C2	龙门协调停止模式	-	0000	0000
Pn6C3	龙门轴间位置偏差过大警告值	%	100	100
Pn6C4	龙门轴间位置偏差过大警报值	指令单位	1048576	1048576
Pn6C5	龙门轴间差分转矩过大警报值	%	100	100
Pn6C6	龙门协调停止速度FF前馈	%	0	0
Pn6C7	龙门同步位置环比例增益	0.1/s	800	400
Pn6C8	龙门同步位置环积分时间	0.1ms	2000	2000
Pn6C9	龙门同步速度限幅	0.1%	20	20
Pn6CA	龙门协调停止轴间差分速度限制	rpm	99	99
Pn6CB	龙门协调停止完成速度阈值	rpm	15	15

3.回原设置

- 目前只支持双光电回原对齐, 需**保证原点的光电开关安装是一致的, 且物理位置相同, 否则可能造成机械损坏**。若实际只有单侧光电开关, 需将硬件信号同时接入 A、B 轴或将原点信号分配至所有轴。
- 测试示例里面使用了原点信号和正向超程开关相关的回原方式, 因此把所有轴的正向超程开关信号和原点开关都定义到 DI1,DI5 上面。只接入一个传感器信号即可
- 龙门回原, 给 A 轴 (主轴) 发送回原命令即可, **回原模式中有关 Z 相回原的不可用**。龙门回原完成以后, Un049 龙门回原标志置 1, Un04E 龙门 Apos 反馈位置清 0。
- 除 35 号回原方式外, 其它回原方式完成以后, UN04E 的值受到 PN522 的影响, 如果回原以后 UN04E 数据比较大, 可减小 PN522 的值 (PN522 默认 5872)

DI1输入信号设定	所有轴	正向超程开关(/POT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI2输入信号设定	A轴	负向超程开关(/NOT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI3输入信号设定	A轴	探针0(/LATCH_0)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI4输入信号设定	A轴	探针1(/LATCH_1)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI5输入信号设定	所有轴	原点开关(/DEC)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI6输入信号设定	不使用	正向超程开关(/POT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI7输入信号设定	B轴	负向超程开关(/NOT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI8输入信号设定	B轴	探针0(/LATCH_0)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI9输入信号设定	B轴	探针1(/LATCH_1)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI10输入信号设定	不使用	原点开关(/DEC)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI11输入信号设定	不使用	正向超程开关(/POT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI12输入信号设定	C轴	负向超程开关(/NOT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI13输入信号设定	C轴	探针0(/LATCH_0)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI14输入信号设定	C轴	探针1(/LATCH_1)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI15输入信号设定	不使用	原点开关(/DEC)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI16输入信号设定	不使用	正向超程开关(/POT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI17输入信号设定	D轴	负向超程开关(/NOT)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI18输入信号设定	D轴	探针0(/LATCH_0)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI19输入信号设定	D轴	探针1(/LATCH_1)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI20输入信号设定	不使用	原点开关(/DEC)	信号不取反(极性相同)	<input checked="" type="radio"/>
DI21输入信号设定	所有轴	紧急停机(/FSTP)	信号不取反(极性相同)	<input checked="" type="radio"/>

4, PLC 程序示例

```

//*****使能*****/
MC_Power_A(Axis:=A, Enable:= TRUE, bRegulatorOn:= 回原使能, bDriveStart:= TRUE, Status=> , ErrorID=> );
//*****执行回原*****/
MC_HomeA(Axis:= A, Execute:=执行回原 , Position:= HomePos, Done=> , Busy=> , CommandAborted=> , Error=>回原ERR , ErrorID=> );

//回原后循环运动
CASE STAND OF
  0:
    IF MC_HomeA.Done THEN
      TON_3(IN:=TRUE, PT:= T#20S, Q=> , ET=> );
      IF TON_3.Q THEN
        STAND :=STAND+1;TON_3(IN:=FALSE);
        END_IF
      END_IF

    1: MC_MoveAbsolute_0(Axis:=A, Execute:=TRUE , Position:= 500, Velocity:= 300, Acceleration:= 1000, Deceleration:= 1000 );
    2: MC_MoveAbsolute_1(Axis:=A, Execute:=TRUE , Position:=0 , Velocity:=300 , Acceleration:= 1000, Deceleration:= 1000, ErrorID=> );

  END_CASE

  IF MC_MoveAbsolute_0.Done THEN
    TON_0(IN:=TRUE, PT:= T#5S, Q=> , ET=> );
    IF TON_0.Q THEN
      MC_MoveAbsolute_0(Execute:=FALSE,Axix:=A);STAND :=STAND+1;TON_0(IN:=FALSE);
    END_IF
  END_IF

  IF MC_MoveAbsolute_1.Done THEN
    TON_1(IN:=TRUE, PT:= T#5S, Q=> , ET=> );
    IF TON_1.Q THEN
      MC_MoveAbsolute_1(Execute:=FALSE,Axix:=A);STAND :=1;TON_1(IN:=FALSE);
    END_IF
  END_IF
END_IF

```

5, 运行波形

