

KV 脚本大辞典

使用编程语言的 PLC 程序

定位运算篇 3

四则运算

函数运算

控制语句

字符串处理

控制

精选！实用 KV 脚本函数的实践技巧

可按应用查找

可按函数查找

可按功能查找

扭矩控制

根据卷径，算出张力和卷取速度

整数转换
TOL
数据转换函数

(返回值) = TOL(转换对象软元件)

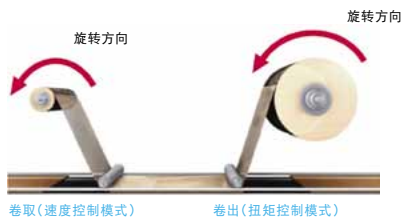
将数据的形式转换成 32 位数据(有符号)。

概要

在扭矩控制和速度控制过程中进行卷出、卷取控制时，扭矩指令值与速度指令值需根据卷径算出。

- 什么是卷出、卷取控制

如下图所示，将薄膜或纸张卷出，经印刷或涂层后再重新卷取，这种装置上使用的控制方法就是卷出、卷取控制。一般而言，卷出、卷取控制是通过控制卷出侧的扭矩和卷取侧的速度来实现的。



- 卷出、卷取控制的要点

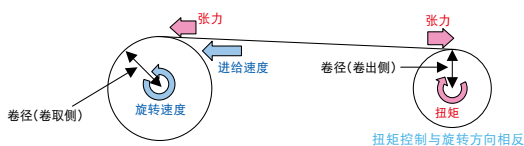
卷出、卷取控制的要点是，**应保持恒定的进给速度和张力**，以保证印刷或涂层的质量稳定。

- 进给速度、张力与旋转速度 - 扭矩的关系

进给速度和张力的控制实际通过马达的旋转速度和扭矩来进行，它们分别有如下关系。

$\text{进给速度} = \text{旋转速度} \times \text{卷径(卷取侧)}$

$\text{张力} = \text{扭矩} \times \text{卷径(卷出侧)}$



- 关于卷径的变化

进行卷出、卷取控制时，随着工序的进行，卷出侧的直径会变小，卷取侧则变大。这叫做卷径变细、卷径变粗。如果卷径变细或变粗，进给速度、张力会随之改变，因此需要根据卷径的变化来控制旋转速度和扭矩。

使用示例

1. 卫生纸卷筒的制造
2. 印刷工艺
3. 薄膜的卷取

KV 脚本示例

根据卷径计算扭矩指令值、速度指令值。

DM0: 扭矩控制指令值(卷出侧)

DM2: 速度控制指令值(卷取侧)

伺服电机每转的脉冲数 Pm[pls/rev]=36000[pls/rev]

DM10: 初始扭矩控制指令 DM12:初始速度控制指令

DM30: R1i[mm](卷出侧卷筒的初始直径)

DM32: P1[pls](卷出侧伺服电机的当前坐标)

DM40: R2i[mm](卷取侧卷筒的初始直径)

DM42: P2[pls](卷取侧伺服电机的当前坐标)

DM50: t[mm](卷取的工件的厚度)

扭矩控制指令值 = 初始扭矩控制指令 × R1i/(R1i - 2 × 厚度 × (P1/Pm))

DM0.L = TOL(DM10.L ★ DM30.F/(DM30.F - 2 ★ DM50.F ★ (DM32.L / 36000.0)))

└ P1/Pm<算出转数>

└ R1i - 2 × 厚度 × (P1/Pm)<算出卷径的变化>

└ 初始扭矩控制指令 × R1i / (R1i - 2 × 厚度 × (P1/Pm))<根据卷径算出扭矩指令>

└ 将运算结果转换成附 32 位数据符号

速度控制指令值 = 初始速度控制指令 × R2i/(R2i + 2 × 厚度 × (P2/Pm))

DM2.L = TOL(DM12.L ★ DM40.F/(DM40.F + 2 ★ DM50.F ★ (DM42.L / 36000.0)))

└ P2/Pm<算出转数>

└ R2i + 2 × 厚度 × (P2/Pm)<算出卷径的变化>

└ 初始速度指令值 × R2i / (R2i + 2 × 厚度 × (P2/Pm))<根据卷径算出速度指令>

└ 将运算结果转换成附 32 位数据符号

函数说明

TOL

将保存在转换对象软元件中的数据转换成 32 位数据(带符号)，再将结果保存为返回值。

例1) DM0.L = TOL(DM2.S)

DM2.S = -352 (16 位数据的转换) => DM0.L = -352 (附 32 位数据符号)

例2) DM0.L = TOL(DM2.F)

DM2.F = -1932.87 (浮点小数点实数数据的转换) => DM0.L = -1932 (附 32 位数据符号)

根据 X、Y 坐标，求出运输臂需要移动的长度和角度

反正切
ATAN
浮动小数点函数

(返回值) = ATAN(运算对象软元件)

求出保存在指定范围内的二进制数据的合计值。

根
ROOT
算术函数

(返回值) = ROOT(运算对象软元件)

求出指定软元件的平方根。

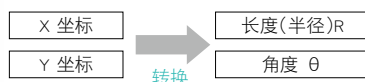
弧度
RAD
浮动小数点函数

(返回值) = RAD(运算对象软元件)

将以度(°) 为单位的角度转换成弧度单位。

概要

根据输入的目标坐标(X、Y 坐标)，计算运输臂需要移动的长度(半径) R 和角度 θ 。



- 根据 X 坐标、Y 坐标进行转换的公式如下。

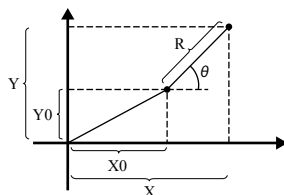
$$R = \sqrt{X^2 + Y^2}$$

$$\theta = \tan^{-1}(Y/X)$$

- 如果运输臂的旋转中心不在原点(0,0) 而是在另外一点(X0、Y0)，则算式如下。

$$R = \sqrt{(X - X_0)^2 + (Y - Y_0)^2}$$

$$\theta = \tan^{-1}\{(Y - Y_0)/(X - X_0)\}$$



KV 脚本示例

将 X, Y 坐标转换成长度 (半径) R 和角度 θ 。

EM0: X 坐标 EM2: Y 坐标 EM4: 长度 (半径) R EM6: 角度 θ (弧度)

```
EM4.D = ROOT(EM0 ^ 2 + EM2 ^ 2)
```

└─ X 坐标的平方 └─ Y 坐标的平方

└─ 计算平方根

根据 X 坐标、Y 坐标求出到原点的距离。

```
IF EM0 > 0 THEN
```

```
EM6.F = ATAN(FLOAT(EM2) / FLOAT(EM0))
```

└─ 将 Y 坐标转换成
浮点小数实数

└─ 将 X 坐标转换成
浮点小数实数

└─ 根据 X 坐标、Y 坐标的倾斜计算角度(弧度)

X 坐标为正时, 在 -90 度到 90 度范围内求出 θ 。

```
ELSE IF EM0 < 0 THEN
```

```
EM6.F = ATAN(FLOAT(EM2) / FLOAT(EM0)) + RAD(180)
```

X 坐标为负时, 在 90 度到 270 度范围内求出 θ 。

```
ELSE
```

```
IF EM2 > 0 THEN
```

```
EM6.F = RAD(90)
```

```
ELSE IF EM2 < 0 THEN
```

```
EM6.F = RAD(270)
```

```
ELSE
```

```
EM6.F = 0
```

```
END IF
```

```
END IF
```

X 坐标为 0 时,
如果 Y 坐标为正, 则 $\theta = 90$ 度
如果 Y 坐标为负, 则 $\theta = 270$ 度
如果 Y 坐标为 0, 则 $\theta = 0$ 度

函数说明

ATAN

执行条件为 ON 时, 根据浮点小数实数的正切(tan) 值求出角度(弧度)。
运算结果保存为浮点小数实数。

例1)

```
DM0.F = ATAN(DM4.F / DM2.F)
```

DM2.F = 10.0, DM4.F = 10.0 \Rightarrow DM0.F = 0.785 ($\approx \pi/4 = 45^\circ$)

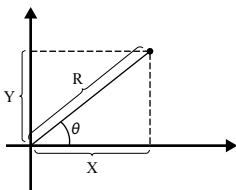
DM2.F = 10.0, DM4.F = 17.32 \Rightarrow DM0.F = 1.047 ($\approx \pi/3 = 60^\circ$)

例2)

```
DM0.F = DEG(ATAN(DM4.F / DM2.F))
```

DM2.F = 10.0, DM4.F = 10.0 \Rightarrow DM0.F = 45.0

DM2.F = 10.0, DM4.F = 17.32 \Rightarrow DM0.F = 60.0



根据 3 点求出圆心坐标及半径

根
ROOT
算术函数

(返回值) = ROOT(运算对象软元件)

求出平方根。(动作与 SQRT 函数一样。)

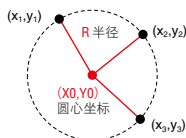
多次方
^
算术运算符

(返回值) = (运算对象软元件 D1) ^ (运算对象软元件 D2)

以运算对象软元件 D2 为指数, 求出运算对象软元件 D1 的多次方值。

概 要

根据 3 点的坐标求出圆心坐标和半径。



$$(X - X0)^2 + (Y - Y0)^2 = R^2$$

- 圆心坐标 (X0, Y0) 的算式

$$\begin{pmatrix} X0 \\ Y0 \end{pmatrix} = \frac{1}{(x_1 - x_2)(y_3 - y_2) - (y_1 - y_2)(x_3 - x_2)} \begin{pmatrix} y_3 - y_2 & -(y_1 - y_2) \\ -(x_3 - x_2) & x_1 - x_2 \end{pmatrix} \begin{pmatrix} (x_1^2 - x_2^2 + y_1^2 - y_2^2)/2 \\ (x_3^2 - x_2^2 + y_3^2 - y_2^2)/2 \end{pmatrix}$$

- 半径 R 的算式

$$R = \sqrt{(x_1 - X0)^2 + (y_1 - Y0)^2}$$

使用示例

1. 用图像传感器测量圆形工件的 3 点, 对圆心坐标的偏移进行调整
2. 根据球形工件的数据计算球的直径误差

KV 脚本示例

求出通过 3 点(x1,y1)、(x2,y2)、(x3,y3)坐标的圆的圆心坐标(X0,Y0)和半径 R。

'圆心坐标 : (X0,Y0) (全局标号*)

'圆的半径 : R (全局标号*)

'测量坐标 : (x1,y1), (x2,y2), (x3,y3) (全局标号*)

'运算中途结果 : X12, Y12, X13, Y13 (局部标号*)

*将软元件注册为标号使用。

X12=x1-x2

Y12=y1-y2

X32=x3-x2

Y32=y3-y2

将运算的中途结果代入局部标号。

$$X0 = \frac{(Y32 \star (X12 \star (x1+x2) + Y12 \star (y1+y2)) - Y12 \star (X32 \star (x3+x2) + Y32 \star (y3+y2)))}{2 \star X12 \star Y32 - 2 \star Y12 \star X32}$$

└─ 计算圆心坐标 X0

$$Y0 = \frac{(-X32 \star (X12 \star (x1+x2) + Y12 \star (y1+y2)) + X12 \star (X32 \star (x3+x2) + Y32 \star (y3+y2)))}{2 \star X12 \star Y32 - 2 \star Y12 \star X32}$$

└─ 计算圆心坐标 Y0

$$R = \text{ROOT}((x1-X0)^2 + (y1-Y0)^2)$$

└─ 计算半径 R

函数说明

ROOT (SQRT)

求出保存在运算对象软元件中的数据的平方根，再将结果保存为返回值。
(SQRT 和 ROOT 只是名称不同，动作完全一样。)

例1) DM0 = ROOT(DM100)

DM100 = 100 ⇒ DM0 = 10

DM100 = 400 ⇒ DM0 = 20

DM100 = 1000 ⇒ DM0 = 31

例2) DM0.F = ROOT(TOF(DM100))

DM100 = 100 ⇒ DM0.F = 10.0

DM100 = 400 ⇒ DM0.F = 20.0

DM100 = 1000 ⇒ DM0.F = 31.62278

例3) DM0.F = ROOT(-100) ⇒ DM0.F = -100.0*

* 如果将负数指定为运算对象软元件，运算错误继电器 CR2012 将开启，并将指定的负数保存为返回值。



www.keyence.com.cn

基恩士(中国)有限公司 最新发售情况，请咨询就近的基恩士

200120 上海市浦东新区世纪大道100号上海环球金融中心8楼

电话: +86-21-5058-6228 传真: +86-21-5058-7178

【关于产品的咨询,请致电】

电话: +86-21-3357-1001 传真: +86-21-6496-8711

咨询热线 4007-367-367

E-mail: info@keyence.com.cn

日本語ダイヤル +86-21-5058-7128



最新信息

扫一扫关注
基恩士微信



安全方面的注意事项

为了安全使用商品,请务必在
使用之前仔细阅读《使用说明书》。