

# 应用说明

## 旋转轴飞剪

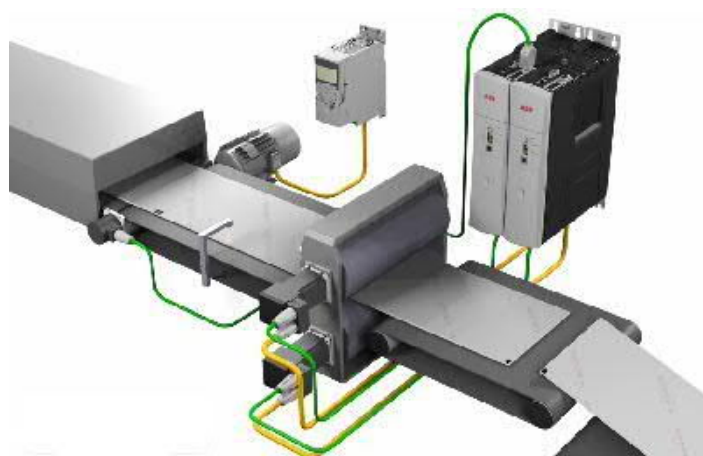
AN00122

版本C (中文)

飞剪允许通过施加加速和减速在规定距离内对主、从轴进行位置锁定。

通常，需要在这些轴的其中一条的特定行进距离上同步这些轴的速度。

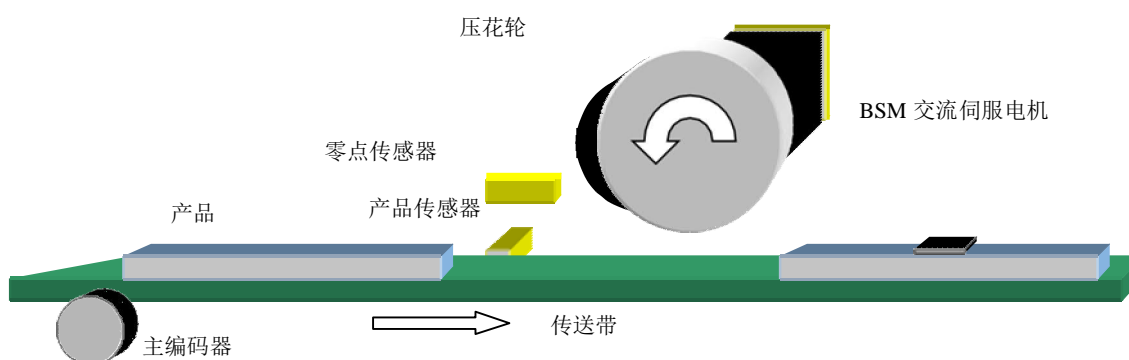
通常，可以通过CAM来实现这一目的，但Mint提供了一个更简单的方法来做到这一点，即使用Mint FLY关键字。



### 引言

轴之间的同步可能相对于主轴和从轴上的线性输出运动发生（参考应用说明AN00116中的例子）。另外，也可以使旋转从轴的表面速度与主轴的线性表面速度同步。

本应用说明详细说明了诸多可能的旋转应用中的一种，即压花轮，并说明了如何使用Mint FLY关键字来简化应用开发。



传送带以恒定的线性速度移动待压花的产品，使其通过机器。这些产品的间距是随机的（但间隔最小）。传送带可以由感应电机和变频传动系统驱动，甚至可以由另一台伺服驱动器/电机控制。由编码器检测输送机在运行时的位置和速度。它通常被称为“主”或“辅助”编码器。

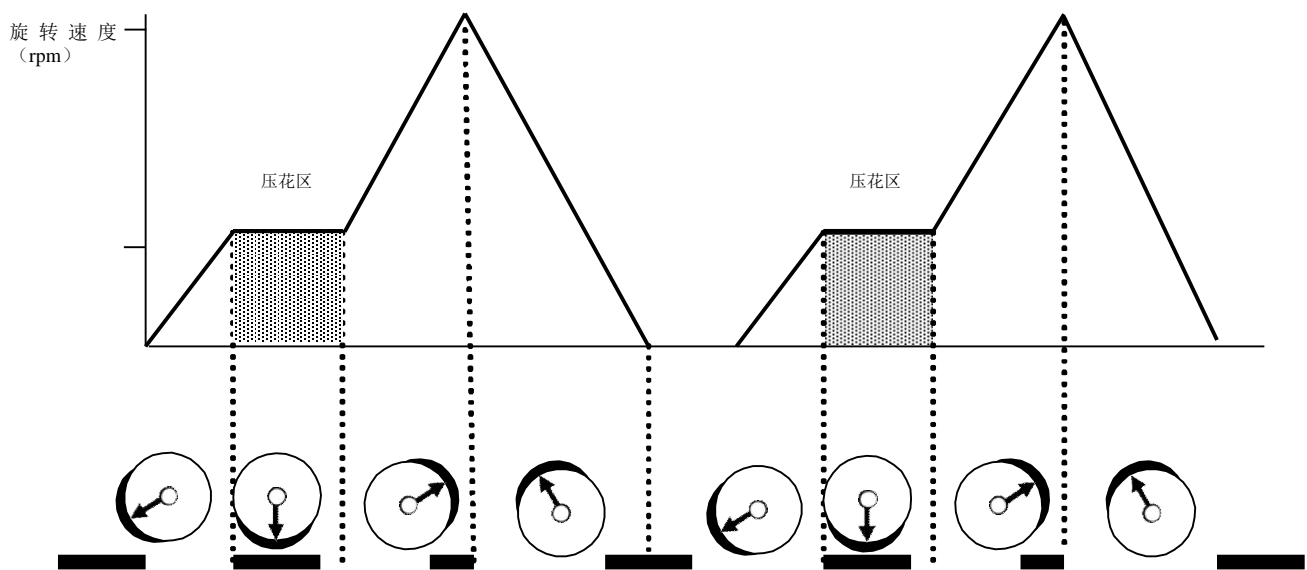
直接连接到电机轴的压花轮被配置为通过伺服电机驱动的从轴。

由运动控制器（或一个集成的运动控制器和伺服驱动器，比如MicroFlex e150）监控主编码器（连接到辅助编码器输入），并使用它来确定压花轮何时开始运动。

在检测到产品进入时，压花轮加速到与产品线性速度完全匹配的速度，并压印产品。所有这些动作都在传送带所行进的指定距离（主轴距离）内完成。在压花完成后，从轴进行加速和减速，以在下一个产品到达之前完成压花轮的一圈旋转。同样，这些动作也是在指定的传送带行程或主轴距离内进行。

传送带在整个过程中的总行程是固定的，等于连续两个产品之间可能的最短距离。在过程开始时，压花轮停顿（在传送带继续运行时，压花轮保持静止），等待产品检测传感器提供一个中断信号以开始下一个周期。

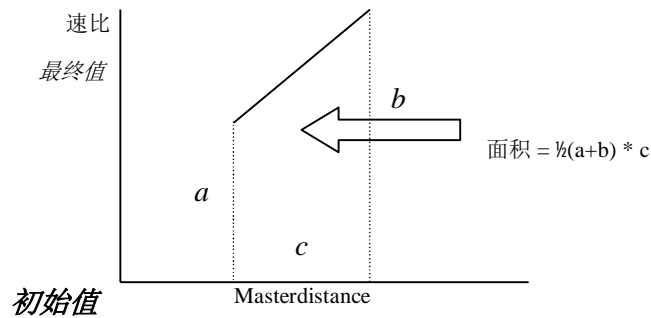
下图说明了两个轴的速比和传送带行进的距离之间的关系（主轴距离）：阴影区域表示旋转轴的表面速度与传送带的线性速度完全匹配的时间段：



本轨迹的每个部分可以分成所谓的“段”（上图中也有说明）。可以使用**MASTERDISTANCE**关键字在Mint中指定传送带在每个段的行进距离。

对每个段来说，曲线下面的区域对应压花轮（从）轴的行进距离。在Mint程序中，可以使用**FLY**关键字指定此距离/区域。

检查具体的段，我们看到可以根据以下方式推算出段的面积：



面积 = 主轴距离 \* (初始速比 + 最终速比) / 2, 或换一种方式表示:

$$FLY = MASTERDISTANCE * (初始速比 + 最终速比) / 2$$

如果只知道MASTERDISTANCE, 则可以使用通用公式来计算FLY, 反之亦然。

为了使从轴返回到起始位置, 所有的飞剪切段的总和必须等于执行机构旋转一次所行进的距离 (在这种情况下, 也相当于电机旋转一周)。通过为MASTERDISTANCE选择适当的值, 只用几行Mint代码就能编写应用程序的整个运动轨迹。

如果任何段导致值为分数值 (在编码器计数级别), 则可能会出现误差。要解决这个问题, 可以在最初计算完所有段的值后对飞剪段进行合计, 将其与所需的合计值进行比较, 然后将差值重新加入其中一个飞剪段。

#### 重要提示

如果对从轴或主轴使用了除1以外的换算系数, 则需要仔细考虑 (例如, 用户可以决定换算主轴, 以与线性毫米的用户单元保持一致)。Mint产品能够表示的浮点数分辨率是有限的。例如, 如果发现换算系数是一个循环分数, 则轴将随时间漂移出相。出于这些原因, 在高精度应用中通常使用1作为换算系数。

#### 实例

假定我们的机器具有以下特点:

传送带信息:

轮周为341.333mm, 带1024 ppr编码器 (4096正交)

12脉冲/mm

1个脉冲= 0.0833mm

压花轮信息:

轮直径 = 100mm (周长314.159mm)

编码器分辨率 = 16384 ppr

每mm压花轮行程52.152个脉冲

1个脉冲= 0.0192mm

产品信息:

产品长度 = 190mm

压花长度 = 50mm

传感器到压花轮的距离=60mm

最小产品间隙=30mm

根据这些详细信息，我们开始计算我们的MASTERDISTANCE和FLY段的值时，可以使用通用公式

$$FLY = MASTERDISTANCE * (\text{初始速比} + \text{最终速比}) / 2$$

段1（加速至同步速度）；

初始比=0；最终比=1；Masterdistance1=60mm

$$Fly1 = 60 * (0 + 1) / 2 = 30\text{mm} = 30 * 52.152 \text{ 个计数} = 1564.56 (1564) \text{ 个计数}$$

段2（在同步速度下压花）

初始比=1；最终比=1；Masterdistance2 = 50mm

$$Fly2 = 50 * (1 + 1) / 2 = 50\text{mm} = 50 * 52.152 \text{ 个计数} = 2607.6 (2607) \text{ 个计数}$$

段3和4（周期的剩余部分）

为了计算这两个段的值，我们需要考虑压花轮行进的剩余距离、每个段的初始和最终速比以及我们允许主轴（传送带）行进的距離。

到目前为止，传送带的行程为：60 + 50 = 110mm。

产品前缘之间的最小主轴行程为190+30=220mm。

因此，我们用110mm的传送带行程，来完成我们的压花轮周期。我们将把这个数字四舍五入到100mm（保留10mm的行程值）。

我们将剩余的100mm分成两等分（各50mm）- 我们将其称之为Masterdistance3。在完成压花周期时，从轴将达到的速比是未知的，我们将称之为R比。

使用通用飞剪公式：

$$Fly3 = Masterdistance3 * (\text{初始比} + \text{最终比}) / 2 = Masterdistance3 * (1 + R) / 2$$

$$Fly4 = Masterdistance3 * (\text{初始比} + \text{最终比}) / 2 = Masterdistance3 * (R + 0) / 2$$

我们还知道Fly3+Fly4必须是压花周期的剩余部分的总和；

$$Fly3 + Fly4 = \text{整个周期} - Fly1 - Fly2 = 16384 - 1564 - 2607 = 12213 \text{ 个计数。}$$

替换Fly3和Fly4，可得出：

$$(Masterdistance3 * (1 + R) / 2) + (Masterdistance3 * (R / 2)) = 12213 \text{ 个计数， 或}$$

$$50 * (1/2 + R) = 12213 / 52.152 = 234.1808\text{mm}$$

$$\text{因此， } R = (234.1808 / 50) - 1/2 = 4.1836$$

我们现在可以把Fly3和Fly4重新代入公式中；

$$Fly3 = 50 * (1 + 4.1836) / 2 = 129.5904\text{mm} = 6758.398 (6758) \text{ 个计数}$$

$$Fly4 = 50 * (4.1836 / 2) = 104.59\text{mm} = 5454.577 (5454) \text{ 个计数}$$

在本应用中，我们决定使用1作为换算系数（我们的用户单位是编码器计数）。我们还决定将所有FLY段四舍五入到最近的整数值。这样，当我们计算出所有飞剪段的总和时，我们需要检查整个周期的“剩余”部分：

$$\text{剩余部分} = \text{周期值} - \text{Fly1} - \text{Fly2} - \text{Fly3} - \text{Fly4} = 16384 - 1564 - 2607 - 6758 - 5454 = 1$$

因此，我们需要将这个1重新计入其中一个段中，以确保压花轮始终能完成一次精确的旋转（我们将把它添加到同步段中）。

应用还可能要求调整压花位置。我们不能在检测到产品后使用延时来调整这个位置，因为这个时间会随着传送带的速度而变化。我们需要做的是等待指定的传送带行程（主轴距离）才开始压花周期。

这可以通过另一个飞剪段来实现。这一次，是通过可变主轴距离（MASTERDISTANCE）上为0的从轴行程（FLY）来实现的——而调整主轴距离的目的是达到所需的压花位置。我们的程序将把这个设置存储在一个程序变量中（我们将它称为nStartDelay——因为我们的辅助编码器换算系数被设置为1，所以我们以编码器计数指定它的值）。

如果编码器计数中指定了所有距离，则压花轮周期的Mint代码变为：

```
' 压花周期...
MASTERDISTANCE(0) = nStartDelay
FLY(0) = 0
MASTERDISTANCE(0) = 60 * 12
FLY(0) = 1564
MASTERDISTANCE(0) = 50 * 12
FLY(0) = 2607 + 1 ' 在此处加上完整周期的剩余部分
MASTERDISTANCE(0) = 50 * 12
FLY(0) = 6758
MASTERDISTANCE(0) = 50 * 12
FLY(0) = 5454
```

通过调整移动缓冲区的大小，使其略大于完成压花周期所需的段数，我们可以监视此缓冲区中的可用空间，并确保在上一个周期仍在进行时预加载下一个周期。这样，当发生产品检测中断时，无需延迟启动一个周期。

我们还将把代码加载到子程序中，以便在应用程序的任何位置调用它。

如果我们在程序开始时添加一个寻零程序（使用输入1作为主输入），并将输入0（我们的产品检测输入）配置为上升沿触发，那么我们就已经轻松地完成了大部分应用程序：

```
' 设置初始停顿时间（可通过HMI进行调整）
Dim nStartDelay As Integer = 500

' 设置换算系数
SCALEFACTOR(0) = 1
AUXENCODERSCALE(0) = 1 ' 或ENCODERSCALE(x) = 1，具体取决于使用的Mint产品

' 配置主轴
MASTERSOURCE(0) = _msAUXENCODER ' 或_msENCODER 根据所使用的Mint产品而定
MASTERCHANNEL(0) = 0 ' 或由使用的Mint产品决定的编码器通道

' 配置输入
' 输入0边缘触发，输入1电平触发
INPUTMODE(0) = 01
' 输入0和1高激活
```

```

INPUTACTIVELEVEL(0)= 011
' 输入0上升沿触发
INPUTPOSTRIGGER(0)= 01
' 两个输入均不在下降沿触发
INPUTNEGTRIGGER(0)= 00

' 设置零位输入
HOMEINPUT(0) = 1

' 启用驱动器
CANCEL(0) : DRIVEENABLE(0) = 1

' 传感器正方向零位
HOME(0) = _hmPOSITIVE_SWITCH
PAUSE IDLE(0)

' 允许将7个移动加载到移动缓冲区中
MOVEBUFFERSIZE(0) = 7
' 加载一个完整的压花周期
doLoad

' 主程序循环...
Loop
  If MOVEBUFFERFREE(0) >= 5 Then doLoad
End Loop

Sub doLoad ( )
' 压花周期...
  MASTERDISTANCE(0) = nStartDelay
  FLY(0) = 0
  MASTERDISTANCE(0) = 60 * 12
  FLY(0) = 1564
  MASTERDISTANCE(0) = 50 * 12
  FLY(0) = 2607 + 1 ' 在此处加上完整周期的剩余部分
  MASTERDISTANCE(0) = 50 * 12
  FLY(0) = 6758
  MASTERDISTANCE(0) = 50 * 12
  FLY(0) = 5454
End Sub

Event In0 GO(0)
End Event

```

### 联系我们

要了解更多信息，请联系您当地的ABB代表，或使用以下一种方式：

[new.abb.com/motion](http://new.abb.com/motion)  
[new.abb.com/drives](http://new.abb.com/drives)  
[new.abb.com/drives/drivespartners](http://new.abb.com/drives/drivespartners)  
[new.abb.com/PLC](http://new.abb.com/PLC)

© ABB公司，2012年，版权所有。保留所有权利。技术规格如有变更，恕不另行通知。