

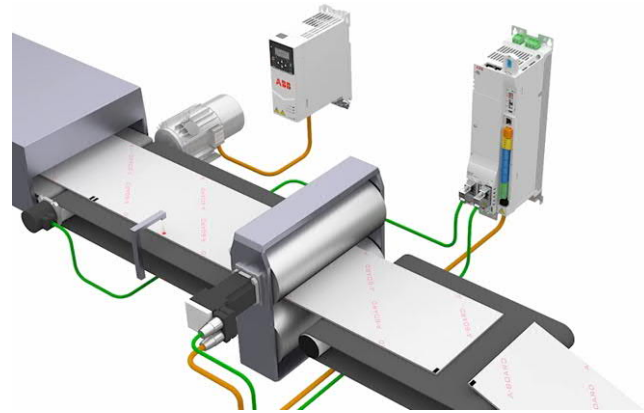
# 应用说明

## 以同步速度运行的旋转切割

AN00154

Rev H (CN)

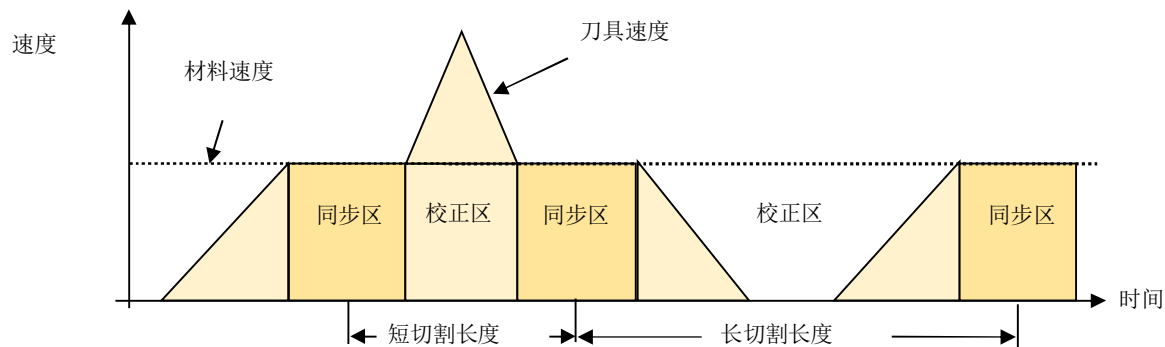
许多运动控制器和 PLC 将使用 CAM 功能来实现复杂的位置/速度跟随轨迹。Mint 通过 FLY 关键字提供了一种更简单，更灵活的方法



### 引言

在许多连续给料生产过程中，通常需要将材料切割成特定的长度。有时甚至可能需要相对于印刷对准标记一定的距离进行精确切割。在这些情况下，经常使用旋转刀筒。在最简单的形式中，在规定的旋转角度（切割区域）下，刀齿滚筒的速度与材料线性速度（即进给速度）的速度比为 1:1，并且通过调整刀具在周期其余部分的速度来对应切割长度。当刀片有角度（或在螺旋上）时经常出现这种情况。

该方法如下图所示：

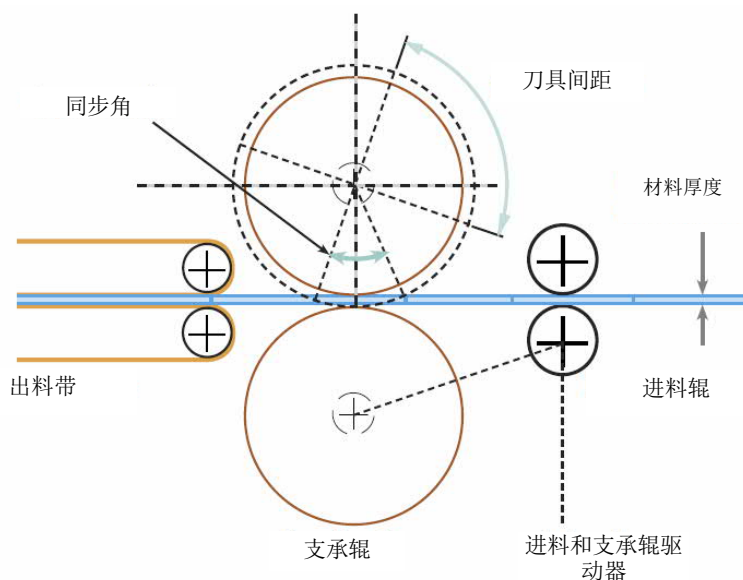


通过使用 MINT 提供的 **FLY**、**FOLLOW** 和 **OFFSET** 等关键字，可以大大简化运动轨迹的编程（有关更详细的说明和应用示例，请参阅应用说明 AN00122 和 AN00226）。虽然这种方法在大量应用中是足够的，但刀具的线性速度仅与材料在精确切割点处的线性速度同步。这可能导致材料处理问题和不完美的切割质量，特别是当产品厚度增加时。

本应用说明旨在说明如何在 Mint 中控制旋转刀筒的速度/位置，以使刀具的线性速度在整个同步区域内与材料的线性速度完全匹配，从而改善材料切割质量。

### 旋转刀具理论

旋转刀具利用伺服电机驱动刀筒（通常是通过机械传动系统，例如齿轮箱和/或皮带和皮带轮来驱动）。



安装在刀筒上的刀具数量可能根据要生产的产品长度而变化。在上面的例子中，滚筒上安装了四个刀具。

刀尖的水平速度分量（在被切割材料的方向上），则可以由下式得出瞬时速度：

$$v = \omega r \sin\theta \quad (\theta \text{以弧度表示, } r \text{是刀辊的半径})$$

在整个同步角期间，刀具将与材料接触。在该区域中，运动控制器必须使刀按照  $\sin\theta$  的倒数运动，以保持恒定的水平速度。请注意，在下死点， $\theta = \pi/2$ （90度）。Mint 控制器三角函数使用度数，因此我们将从现在开始以度为单位关联所有角位移。

可以根据材料厚度（m）和刀具半径（r）计算同步角：

$$\text{同步角} = 2 * (\text{Cos}^{-1}((r-m)/r))$$

如果我们将刀轴表示围绕圆周切线的线性单位（例如 mm 或英寸），并且我们还将材料行程表示为相同的线性单位，则两个轴之间的比率变为...

$$1 / \text{Sin}(\theta)$$

...其中， $\theta$ 在切割点处为90度（在上面的例子中，刀齿滚筒的下死点），即，当刀处于下死点时，两个轴之间的线性速度比为  $1/\text{Sin}(90) = 1$ 。随着角度偏离90度， $\text{Sin}(\theta)$ 减小。并且刀的速度比因此增加以保持线性速度分量与材料速度比率为1。

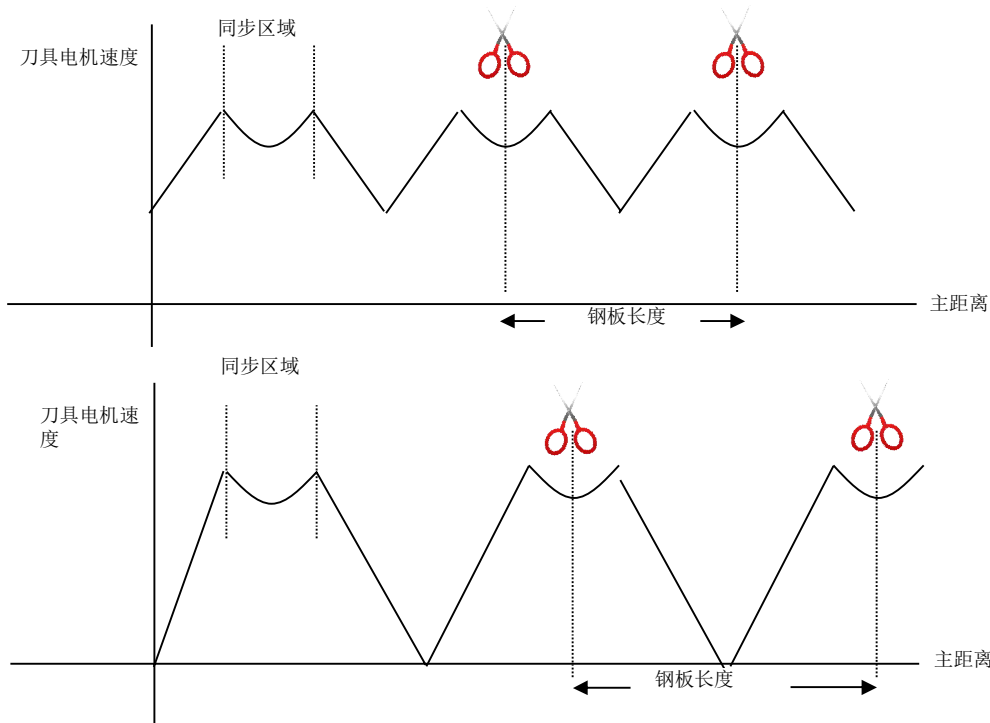
通过最初将 $\theta$ 设置为（90-同步角度/2），并将 $\theta$ 按固定增量递增（例如2度），我们可以迭代通过轨迹的同步部分，并计算出其中每一个角度下刀辊电机和每个材料之间的速度比。

正如应用说明 AN00122 所示，Mint FLY 关键字允许运动控制器在材料行进的特定距离（MASTERDISTANCE）上同步从轴（FLY）的位置。

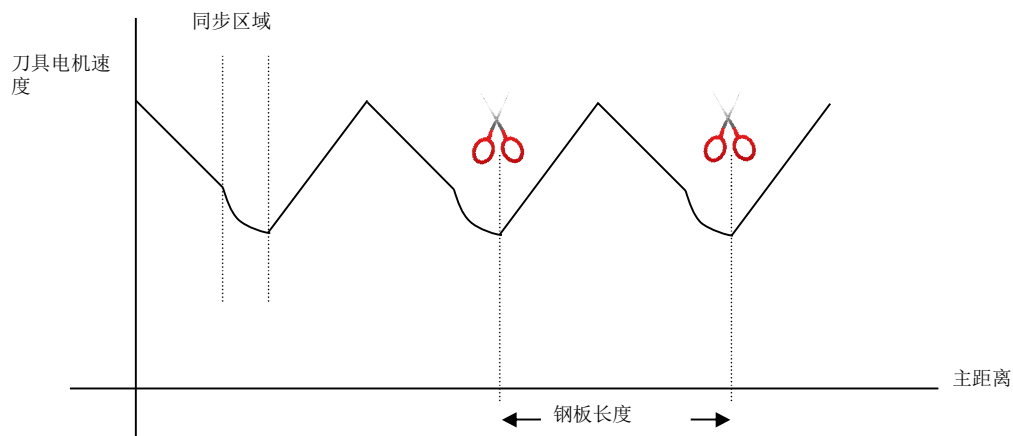
在这种情况下，FLY 是将之前使用的 $\theta$ 的增量（例如2度），转换为刀具周长的等效行程（例如  $\text{FLY} = 2 * \text{刀具周长} / 360$ ）。因此，使用 Mint 飞剪段的通用公式（ $\text{FLY} = \text{MSD} * (\text{IR} + \text{FR}) / 2$ ）- 其中 IR 是初始速度比，FR 是最终速度比 - 由此可以计算每个 FLY 段对应的 MASTERDISTANCE（材料行程）。

在计算出每个 FLY 段的 MASTERDISTANCE 后，可以看到材料在同步切割过程中行进的总距离是这些 MASTERDISTANCE 的总和（当然刀具的总行程是同步角）。现在剩下的就是由运动控制代码加载飞剪段，以使刀具在剩余的所需切割长度上完成剩余部分的刀具间距的旋转（此过程将在后面详述）。

根据切割长度，生成的运动轨迹将类似于下图。随着切割长度的增加，同步区域外的刀速将减小，直到刀停止并开始等待：



在某些应用中，一旦达到下死点（即，一旦切割完成），就不必再保持同步轨迹。如下所示，在这种情况下，轨迹变得不再对称。这在切割长度较短时存在优点，即刀轴需要达到的最大速度会降低...



本应用对应的示例代码包含一个参数。它允许用户在同步区域的这两种操作模式之间进行选择。

#### 计算同步区域的 FLY 段

正如我们已经看到的，这些计算取决于刀齿滚筒的半径 ( $r$ ) 和材料厚度 ( $m$ )。材料厚度和刀具半径共同决定了同步角。通常，允许操作员设置此同步角度（例如通过 HMI）。因此，在此示例中，我们假设同步角度是直接输入的（而不是根据  $2 * (\cos^{-1}((r-m)/r))$  计算），但两种方法都是可以接受的。

刀具半径和同步角被作为参数传递给可重复使用的 Mint 子程序。通过引用传递的数组允许子程序返回描述每个运动段的 MASTERDISTANCE、FLY 和速度比值的数据。然后，可以在应用程序中使用这些数据，把 FLY 和 FOLLOW 段加载到刀轴的移

动缓冲区中。子程序使用通过引用传递的其他参数来计算并回传调用例程常用的距离。该距离将在稍后的应用中用于确定刀具是否需要停止以产生所需的切割长度。

为了在同步区域内计算两个轴之间的速度比，代码在同步区域内使用两度的步长递增，即 `nInc` 默认指定为 2（因此同步角必须指定为偶数值）。

```
'Calculate speed ratio based on angular increments in slave axis
'Assumed minimum increment of 2 over max angle of 180...
i = 1
For nAngle = nSynchAngle To (90 + (nSyncMode * (90 - nSynchAngle))) Step nInc
    fSpeedRatio(i) = 1/Sin(Float(nAngle))
    i = i+1
Next nAngle
'Remember how many points were calculated...
j = i-1
```

传递给该子例程的参数“`nSyncMode`”确定是只应使用同步区域的前半部分（`nSyncMode = 0`），还是应同时使用两半部分（`nSyncMode = 1`）。

一旦知道了速度比（以及已计算的这些数值），子例程就会加载作为参数传递的数组，其中包含稍后可能由代码使用的 `FLY` 和 `MASTERDISTANCE` 值。

```
'At this point all of the speed ratios for the synch section are stored
'in an array. -1 indicates the end of the data...
'Using FLY = MSD * (IR + FR)/2 we can now calculate the FLY table
'for the synchronous section of the profile. The first speed ratio will
'be used during the accel up to speed or continuation from previous cycle
'so we need only worry about the second increment onwards...
For i = 2 To j
    fFLY(i) = _fKnifeCirc * Float(nInc) / 360
    fMSD(i) = (2*fFLY(i)) / (fSpeedRatio(i-1) + fSpeedRatio(i))
Next i
```

### 计算剩余切割长度的 FLY 段

用于计算同步切割区域的 `FLY` 和 `MASTERDISTANCE` 值的子例程也会计算其他几个参数。

- 主轴（材料）在同步切割阶段行进的距离（刀具必须在剩余的材料长度内完成一个刀具间距的剩余部分的行进。剩余材料长度是总切割长度 - 切割同步段的行程）。
- 如果刀具在完成同步区域后减速到停止，在刀具之间的半途停顿，则材料的行程是多少
- 如果刀具加速到同步区域的起点处，在刀具之间的半途启动，则材料的行程是多少
- 刀具从同步区域的末端到刀具之间的中点（即长切割长度的停止点）的行走距离
- 刀具从刀具之间的中间点到下一个同步区域的起点的行走距离

如果所需的切割长度大于（a）+（b）+（c）的总和，那么刀具必须完全停止（刀具离开切割点 - 即刀具之间的中间点），以在重新启动前允许更多的材料通过切割区域（这称为“`dwell`”）。在这种情况下，在 `MASTERDISTANCE` 上加载 `FLY=0`（刀具行程），用于补偿在刀辊的减速和加速期间所需的切割长度和材料行程之间的差异。

如果所需的切割长度小于（a）+（b）+（c）的总和，则刀具将在每个刀具之间略微减速或增加速度，并且不需要加载“`dwell`”段。

因为需要实现在半同步区域或完整同步区域之间切换的功能，这些计算中采用的公式被复杂化。但是，应用说明中的示例代码完成了这些计算，通过二次方程计算得出需要的数据。

```
'Carry out calculations relating to cut length which can be changed at any time...
m = (fCutLength - fTotalMSD)

B = fKnifeRatio(j) + fKnifeRatio(1) - ((2*fSynchToMid)/m) - ((2*fMidToSynch)/m)
C = (fKnifeRatio(j) * fKnifeRatio(1)) - ((2*((fKnifeRatio(1)*fSynchToMid)+(fKnifeRatio(j)*fMidToSynch)))/m)
x = (-B + Sqrt((B^2) - 4*C))/2

M1 = (2*fSynchToMid) / (fKnifeRatio(j)+x)
M2 = m-M1
```

本应用说明中包含的 Mint 示例程序详细说明了如何加载这些飞剪段。将刀具带到刀具之间的中点的段，被作为 doLoadSynch 子例程的一部分加载。将刀具带回下一个同步区域开头的段，被作为 doLoadRemainder 子例程的一部分加载。

### 补偿浮点计算的不准确性

刀辊围绕周线以“线性”单位换算。因为其中涉及到使用数学常数 Pi，所以总是会导致无法精确表示的比例因子。

例如：

电机每转分辨率 10,000，并配有 10:1 齿轮箱。刀具半径为 90mm。因此，围绕周线以 mm 为单位的线性行程的比例因子是...

$(10000 * 10) / (2 * \pi * 90) = 176.83882565766148418764862596946...$

随着应用继续使用分数比例因子加载飞剪段，刀轴将开始偏离位置，最终导致刀片切割材料时刀具速度与材料的线性速度的不同步。

为对其进行补偿，只要特定的刀片经过切割循环的中心点时，Mint 程序就会生成一个数字输出。示例使用数字输出 0。它必须是前三个输出之一以保证锁存正确运行。配置锁存事件以在此时捕获刀轴编码器值。如果刀具未偏离，则该编码器的值应在每个循环中保持一致（假设在刀具的一次旋转中编码器计数为整数 - 如果存在分数编码器计数，则必须采用在每次旋转时把刀具编码器的值设置为零的第二种机制，而不只是依靠 ENCODERWRAP 的编程）。

将理想的编码器值与通过数字输出锁存的编码器值进行比较，然后将两者之间的误差作为 **OFFSET** 加入。**OFFSET** 应在刀具同步区域外发生的一部分材料行程中应用。因为刀具已经完成切割，可以通过锁存事件启动此 **OFFSET**。校正非常小，因此对刀速影响很小...

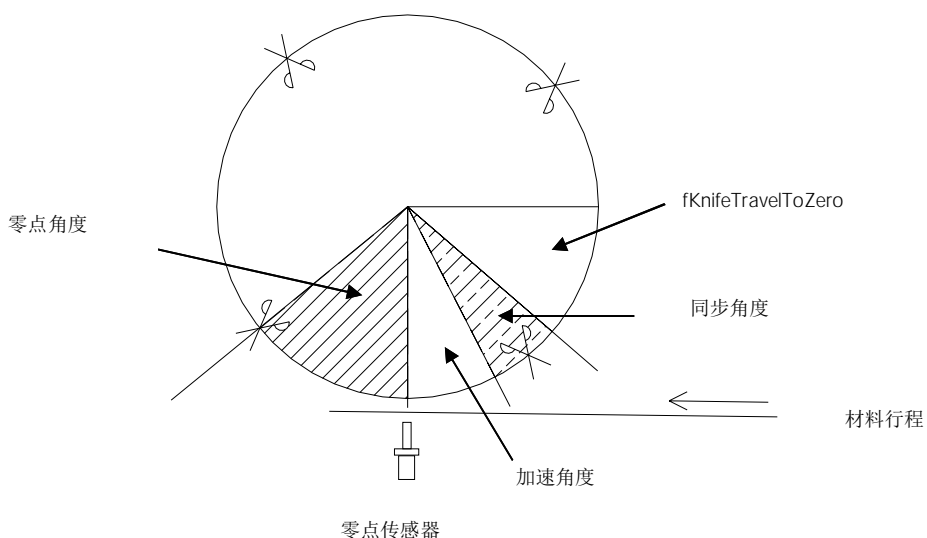
### 切割对准

为了确保刀具始终以同步线性速度切割，我们必须确保刀齿滚筒的物理位置不会相对于编程的速度轨迹发生移动（即，不允许移动刀具的位置）。因此，必须通过人工缩短或延长所产生的切割长度来进行对准校正。

### 启动刀具

在允许系统运行之前，刀具必须寻零（因此运动控制器知道其中一个刀片的位置 - 通常，零点标志的数量与安装的刀片一样多）。

零点例程包括一个“零点偏移”参数。在检测到零点传感器后，该参数允许操作者确保刀辊把刀片移离支承辊（这样，当机器首次启动时，材料可以从刀具下方通过）。该零点偏移通常是可调节的（例如通过 HMI），以便（与通常为固定值的“Accel Angle”参数一起）操作者有方法调整刀具的物理位置，以便刀具在刀具速度轨迹的同步部分进行切割（通常在轨迹这一部分的正中心）。



当在对准模式下启动系统时，通常希望刀具的启动方式能实现在对准时进行第一次切割。这可以在 Mint 中通过使用 TRIGGER 等一组关键字，使刀具在材料的已知位置上启动来轻松实现。TRIGGERVALUE 是在检测到第一个对准标记时计算出来的。它考虑了当刀具从静止加速到轨迹的同步区域的起点处时材料行进的距离。

如果您需要有关纳入对准控制的更多信息，请联系 [cn-motionsupport@cn.abb.com](mailto:cn-motionsupport@cn.abb.com)。

### Mint 应用代码示例

本应用说明中包含的 Mint 示例程序包含 Mint (mnt) 程序文件和用于 Nextmove e100 运动控制器的设备配置文件 (dcf)。代码使用条件编译编写，以便它也可以在 MicroFlex e190 和 MotiFlex e180 驱动器上运行。因此，代码适用于所有平台，并己为材料编码器输入选择了编码器通道 2。

如果使用 NextMove e100 运行该示例，则 dcf 文件要求通过以太网 Powerlink（设置为节点 3）连接 MicroFlex e190 驱动器。如果使用 NextMove e100，刀轴为轴 3。如果使用伺服驱动器运行该程序，则刀轴将变为轴 0。

务必检查编码器通道 2（材料编码器）的编码器计数方向是否为正（即应用要求材料正向行走）。

数字输入 4ON 时，将执行刀轴回零。（无论使用哪个驱动器，零点传感器应连接 DI0）。除非刀具已经寻零，否则机器模拟将不会运行。一旦刀轴已经寻零，刀轴将跟随材料编码器的速度/位置开始运行。

使用 Mint Workbench 中的数据监视窗口在运行时修改 NETFLOAT (13) 的值 - 这可以模拟操作员输入的所需切割长度的变化。NETINTEGER (1) 可用于修改刀具数量，NETINTEGER (2) 可用于调整同步角度，NETINTEGER (3) 可用于设置配置的同轴角度是半角 (0) 还是全角 (1)。务必按使用全角度的方式设置同步角度参数，在只使用前半段的情况下也是如此。

系统的默认操作参数（例如刀具半径，刀具数量）在 doInitKnife 子例程中设置，因此可以调整这些参数以适应应用。

示例中的 Onerror 代码非常简单，只能读取错误信息，通过 Workbench 显示有关错误的一些诊断信息，然后重新启动程序。如有必要，可以根据需要扩展/增强。

## 联系我们

要了解更多信息，请联系您的当地的 ABB 代表，或以下一种方式：

[new.abb.com/drives/low-voltage-ac/motion](http://new.abb.com/drives/low-voltage-ac/motion)

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

[new.abb.com/channel-partners](http://new.abb.com/channel-partners)

© ABB 公司，2019 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。