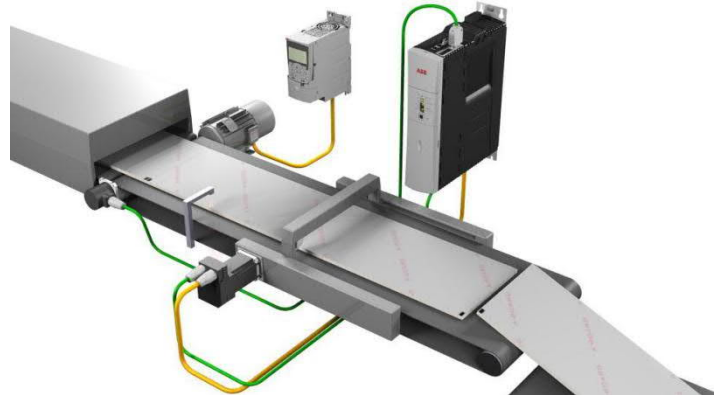


Mint FLY命令的名字来源于其固有的能力，即与其他运动控制器和语言通常使用的CAM轨迹相比，在适应典型的“飞剪”应用方面，它可以大大简化代码开发。

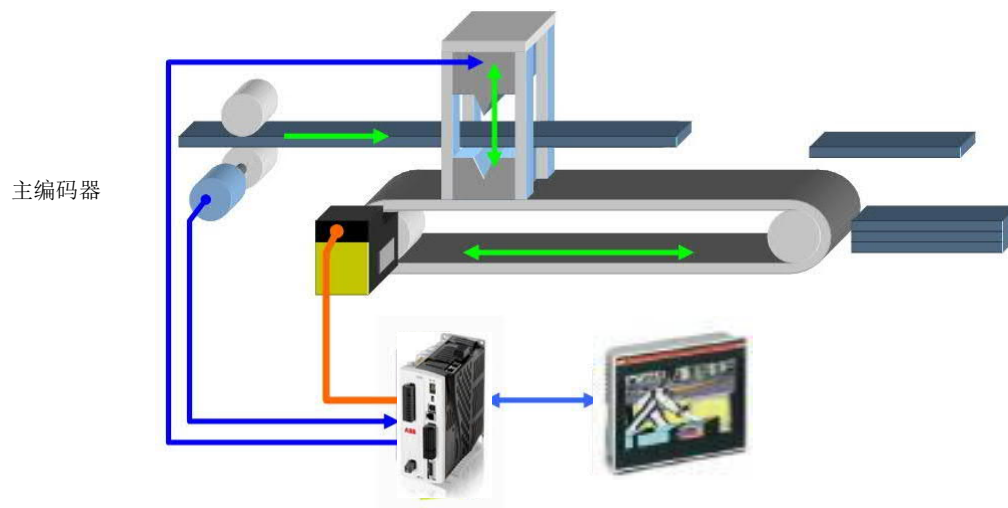
FLY允许主轴和从轴在规定距离内通过强制加速和减速锁定位置。



### 引言

下图所示为一个典型的飞剪应用。其中，线性刀具机构加速到与输入的材料流同步的速度，然后按预定的长度进行切割，再返回重复整个循环。

由Mint控制的剪切机



系统示例-MicroFlex e190交流伺服驱动器通过使用Mint存储器模块 (+N8020) 实现先进的运动控制，并结合ABB CP600 HMI构成完整的单轴飞剪解决方案。

### 说明

一组压送辊以恒定的线性速度使要切割的材料通过机器。这些压送辊可以由感应电机和变频传动系统驱动，甚至可以由伺服驱动器/电机控制。由编码器（通常与压送辊相连，但有时也由材料通过测量辊驱动）检测材料通过机器时的位置和速度。该编码器通常被称为“主编码器”。

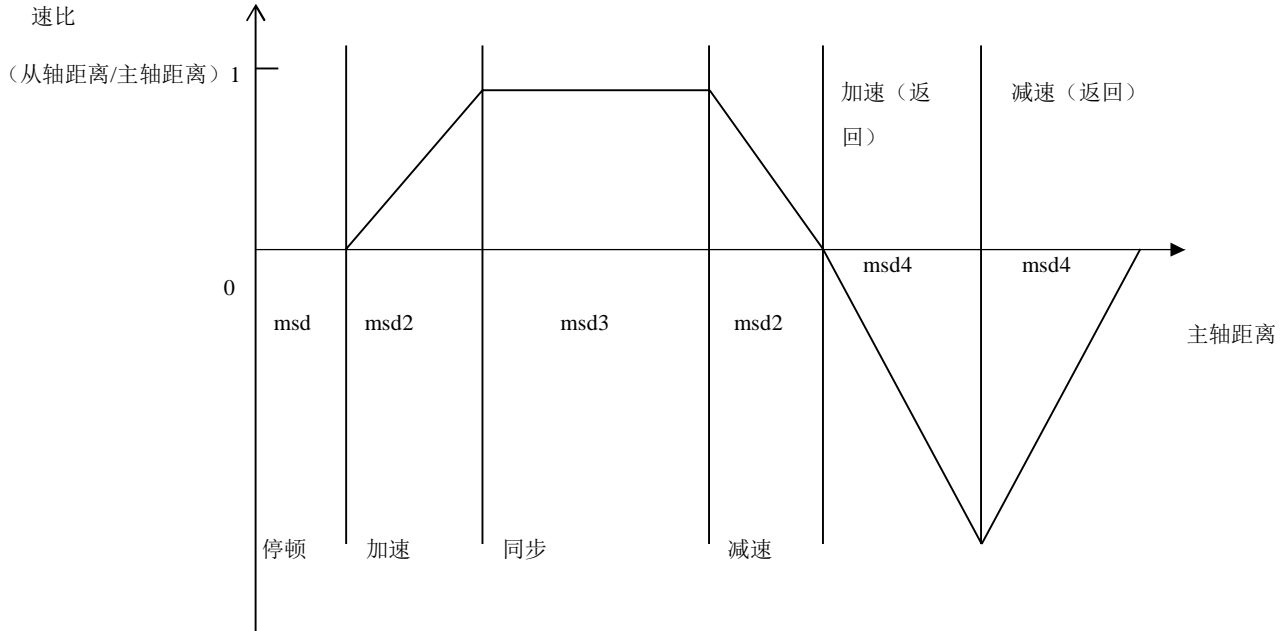
安装有剪切刀具机构的定时皮带传动通过伺服电机驱动，被配置为从轴。

运动控制器（在上图中，我们展示了一个集成的运动控制器和伺服驱动器——MicroFlex e190）监控主编码器（连接到驱动器上的编码器通道2），并使用它来确定剪切机开始运动的时间。

剪切机被加速到与材料速度完全相同的速度，然后开始切割（在本例中，通过激活螺线管一段固定的时间来启动刀具）。这些动作都在一个定义好的材料行进距离内完成（在Mint中，由MASTERDISTANCE关键字定义）。在切割完成后，剪切机减速至停止，然后返回其开始位置。同样，这些动作是在指定的材料行程或主距离上完成。

在整个过程中，材料的总行程是固定的，与最短的切割长度一致。改变（增加）切割长度的一种常用方法，是在过程开始时增加一个停顿动作（在此期间，当更多材料通过机器时，剪切机保持静止）。就应用程序代码而言，第一种方法最简单。另一种方法是“拉伸”从轴返回到其起始位置的距离，此时整个周期的主距离等于（或小于）所需的切割长度。虽然这样会稍微增加应用程序代码的复杂程度，但它确实降低了能量消耗和机械应力。

下图说明了两个轴的速比与被称为主轴距离的材料行进距离之间的关系：



此轨迹的每个部分都可以被分为所谓的“段”（上图也说明了这一点。图中显示了每个段的主轴距离[msd]）。对每个段，曲线下面的区域对应剪切机（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。

可以使用**MASTERDISTANCE**关键字在Mint中指定材料在每个段的行进距离。查看某一具体的段，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：

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

如果仅知道**MASTERDISTANCE**，则此通用公式可用于计算**FLY**，反之亦然。

为了使剪切轴返回到起始位置，两个返回段的面积的总和必须等于前进方向上行进的距离。可以看出，每个返回段是向前行进的总距离的一半（对三角形返回轨迹而言）。

通过为**MASTERDISTANCE**选择合适的值（例如，确保剪切机在一个方向上行进的总距离不超过轴的物理限制），可以用几行Mint代码来编写应用程序的整个运动轨迹。

如果同步速度段导致**MASTERDISTANCE**或**FLY**为分数值（在编码器计数级别），则可能会出现这个问题。可以通过将同步段分成两半来解决这一问题。如果试图求出前进方向上轴的总行程，然后将其分割为两个相等的返回段，也会出现浮点算术误差的问题。使用Mint **POSTARGETLAST**关键字可以消除这些错误。它会自动返回加载到Mint缓冲器中的任何运动的目标位置。因此，可以用它来确定剪切轴的向前总行程，而不是求出已经使用的所有**FLY**段的总和。

Mint **MOVEPULSEOUTX**命令允许我们使用运动轨迹触发切割动作。这样，就能确保在达到同步速度的准确时间上进行输出。

现在，我们可以使用上面的基本方法为这类应用编写一些通用的Mint代码：

‘ 允许将8个移动加载到移动缓冲区中

**MOVEBUFFERSIZE(0) = 8**

Loop

‘ 计算停顿距离（msd1）并等待

**msd1 = ProductLength - ((2 \* msd2) + msd3 + (2 \* msd4))**

**MASTERDISTANCE(0) = msd1**

**FLY(0) = 0 : GO.0**

‘ 从零速度斜坡上升到1:1比率

**MASTERDISTANCE(0) = msd2**

**FLY(0) = msd2 / 2 : GO(0)**

‘ 触发剪切

**MOVEPULSEOUTX(0) = 250**

**GO(0)**

‘ 以同步速度运行

**MASTERDISTANCE(0) = msd3/2**

**FLY(0) = msd3/2 : GO(0)**

**MASTERDISTANCE(0) = msd3/2**

**FLY(0) = msd3/2 : GO(0)**

‘ 减速至零速

**MASTERDISTANCE(0) = msd2**

**FLY(0) = msd2 / 2 : GO(0)**

‘ 返回段可以使用目标位置来计算返回距离

fPosTarget = POSTARGETLAST(0)

‘ 返回段1

MASTERDISTANCE(0) = msd4

FLY(0) = - fPosTarget / 2 : GO(0)

‘ 返回段2

MASTERDISTANCE(0) = msd4

FLY(0) = - fPosTarget / 2 : GO(0)

End Loop

在本例中，我们为每个MASTERDISTANCE使用了段名称（msd2、msd3等），但在实际应用中，我们需要为它们定义值（例如，我们可能决定在已知的材料行程中使剪切机加速），或进一步执行计算以确定这些段的大小（例如，同步段的大小取决于刀具切割材料所需的时间）。

FLY关键字是Mint编程语言中最强大的命令之一，它可以用于各种应用。其中包括：

- 标签供料器
- 机械离合器的更换（增加了精度和重复性的优势）
- 印刷机给纸装置（印刷机为主轴）
- 更换分度齿轮箱（包装和食品加工应用）

请注意Mint系列中的其他应用说明。我们将在这些应用说明中讨论FLY关键字的更多使用方法。

### 示例

使用经典飞剪时，插值是线性的。本应用说明中包含两个示例，它们说明了如何使用线性和s形轨迹执行此应用。

### 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  
new.abb.com/drives  
new.abb.com/drives/drivespartners  
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。