Motion Control Products

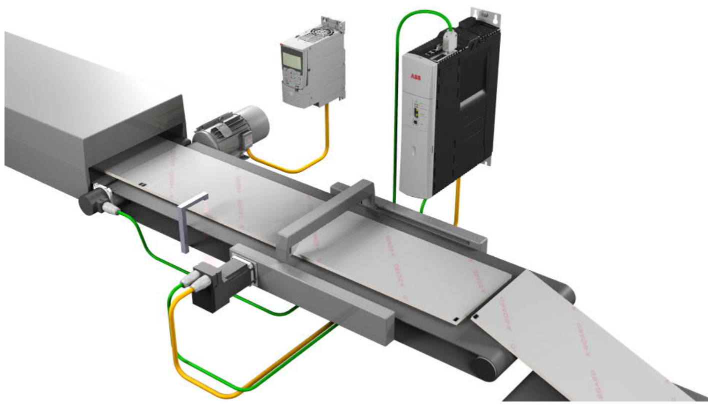
Application note
Flying shear

AN00116

Rev D (EN)

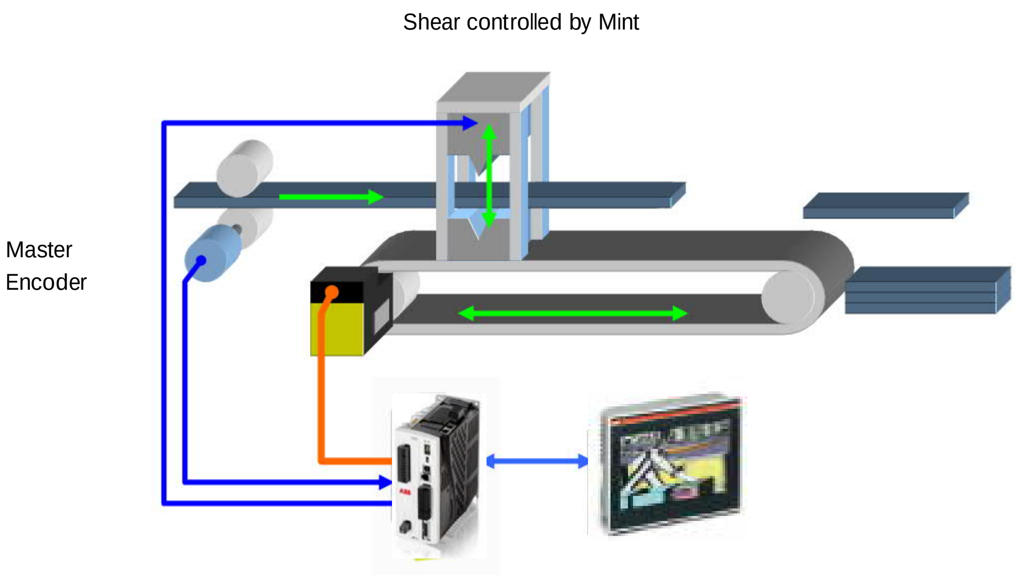
The Mint FLY command takes its name from its inherent ability to greatly simplify development of code to suit the typical "flying shear" application in comparison with the use of CAM profiles that may typically be used by other motion controllers and languages.

FLY allows the position lock of master and slave axes over defined distances with imposed accelerations and decelerations.



Introduction

The classic flying shear application, where a linear knife mechanism accelerates up to synchronous speed with an input stream of material before then cutting this at a predefined length and then returning to repeat the cycle, is illustrated by the figure below.



System example – MicroFlex e190 AC servo drive, with advanced motion control enabled via use of the Mint memory module (+N8020), combined with an ABB CP600 HMI for a complete single axis flying shear solution

Description

A set of nip rolls moves the material to be cut at a constant linear velocity through the machine. These nip rolls may be driven by an induction motor and variable frequency drive system or may even be controlled by a servo drive/motor. An encoder, often attached to the nip rolls, but occasionally driven by the material via a measuring wheel, detects the position and speed of the material as it passes through the machine. This is usually termed the '**master encoder'**.

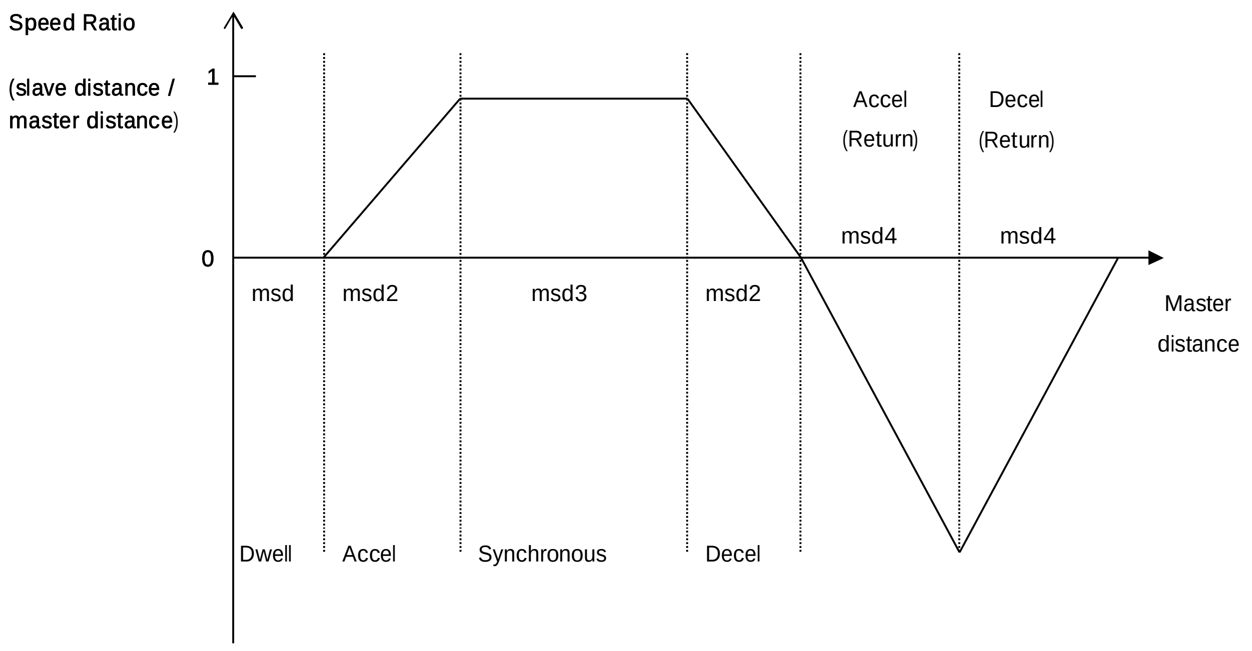
A timing belt drive, with a shear knife mechanism attached to it, is configured as a slave axis driven via a servomotor.

The motion controller (in the illustration above we have shown an integrated motion controller and servo drive – MicroFlex e190) monitors the master encoder (wired to encoder channel 2 on the drive) and uses it to determine when the shear should begin its motion.

The shear is accelerated to a speed that exactly matches the material speed and a cut is initiated (in this example by activating a solenoid for a fixed duration to actuate a blade). This all takes place over a defined distance traveled by the material (in Mint this is defined by the MASTERDISTANCE keyword). Once the cut has taken place the shear is decelerated to a stop and then returned to its start position. Again these actions take place over a specified material travel or master distance.

The total distance traveled by the material for during this whole process is fixed and corresponds to the shortest possible cut length. One commonly used method for varying (increasing) the cut length is to add a dwell to the beginning of the process (during which time the shear remains stationary whilst more material passes through the machine). This first method is the simplest in terms of application code. The other method is to "stretch" the distance over which the slave axis returns to its start position so the total master distance for the cycle is (or is just less than) the required cut length. Whilst this makes the application code slightly more complex it does mean the energy consumption is reduced, as is the mechanical stress.

The figure below illustrates the relationship between the speed ratio of the two axes and the distance traveled by the material referred to as the master distance):



Each element of this profile can be split into what are termed 'segments' (this has also been illustrated on the figure above and master distances [msd] are shown for each of these). For each segment the area under the curve corresponds to the distance traveled by the shear (slave) axis. In Mint programs this distance/area can be specified using the FLY keyword.

The distance traveled by the material for each segment can be specified in Mint by using the MASTERDISTANCE keyword.
If we examine a particular segment we can see that the segment area can be derived from:

Area = Master Distance * (Initial Speed Ratio + Final Speed Ratio) / 2

or put another way:

FLY = MASTERDISTANCE * (Initial Speed Ratio + Final Speed Ratio) / 2

This generic formula can be used to calculate FLY if only MASTERDISTANCE is known and vice-versa.

To return the shear axis to the start position the sum of the two return areas must equal exactly the distance traveled in the forward direction. It can be seen that each return segment is half the total distance traveled in the forward direction (for a triangular return profile).
By selecting appropriate values for MASTERDISTANCE (e.g. to ensure the total distance traveled by the shear in one direction does not exceed the physical limitations of the axis) the whole motion profile for the application can be written with just a few lines of Mint code.

Problems can arise if the synchronous speed segment results in a fractional value (at an encoder count level) for either MASTERDISTANCE or FLY. This can be overcome by splitting the synchronous section in half. Problems may also occur with floating point arithmetic errors when trying to sum the total travel of the axis in the forward direction and then splitting this in half for the return segments. These errors can be eliminated by making use of the Mint POSTARGETLAST keyword. This automatically returns the target position for any moves loaded into the Mint buffer, and so it can be used to determine the total forward travel for the shear axis instead of summing all of the FLY segments that have been used.

The Mint MOVEPULSEOUTX command allows us to use the motion profiler to trigger the cut action. This ensures the output is operated at the exact time the synchronous velocity is reached.

Using the basic methods outlined above we can now write some general purpose Mint code for this type of application:

```
  'Allow 8 moves to be loaded into the move buffer
MOVEBUFFERSIZE(0) = 8

Loop
   ' Calculate dwell distance (msd1) and wait
msd1 = ProductLength – ((2 * msd2) + msd3 + (2 * msd4))
MASTERDISTANCE(0) = msd1
FLY(0) = 0 :  GO.0

  ' Ramp from zero speed to 1:1 ratio
MASTERDISTANCE(0) = msd2
FLY(0) = msd2 / 2 : GO(0)

' Trigger the shear
MOVEPULSEOUTX(0) = 250
GO(0)

' Run at synchronous speed
MASTERDISTANCE(0) = msd3/2
FLY(0) = msd3/2 : GO(0)
MASTERDISTANCE(0) = msd3/2
FLY(0) = msd3/2 : GO(0)

' Decelerate to zero speed
MASTERDISTANCE(0) = msd2
FLY(0) = msd2 / 2 : GO(0)
```

```
'Return segments can use target position to calculate return distance
fPosTarget = POSTARGETLAST(0)

' Return segment 1
MASTERDISTANCE(0) = msd4
FLY(0) = - fPosTarget / 2 : GO(0)

' Return segment 2
MASTERDISTANCE(0) = msd4
FLY(0) = - fPosTarget / 2 : GO(0)
End Loop
```

In this example we have just re-used our segment names (msd2, msd3 etc..) for each MASTERDISTANCE, but in a real application we would either need to define values for these (e.g. we may decide to ramp the shear up to speed over a known material travel) or carry out further calculations to determine the size of these segments (e.g. the synchronous segment size will depend on how long it takes the knife to cut the material).

The FLY keyword is one of the most powerful commands available in the Mint programming language and it can be used in a wide variety of applications. Some of these include:

- Label feeders
- Replacement of mechanical clutches (with the added advantage of precision and repeatability)
- Press feeders (where the Press is the master axis)
- Replacing indexing gearboxes (packaging and food processing applications)

Look out for additional application notes in the Mint series where we will discuss further uses for the FLY keyword.

Examples

When using the classic flying shear the interpolation type is linear. Two examples are included with this application note showing how to do this application with both linear and s-ramped profiles.

Contact us

For more information please contact your
local ABB representative or one of the following:

new.abb.com/motion
new.abb.com/drives
new.abb.com/drives/drivespartners
new.abb.com/PLC