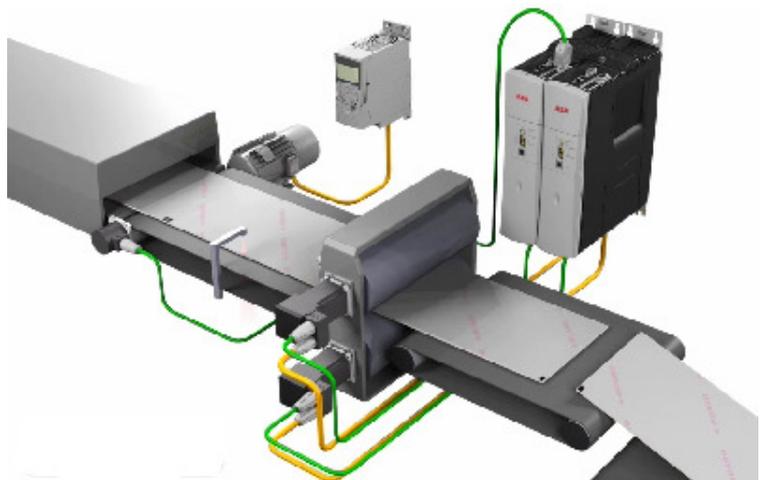# Application note
Rotary axis flying shear

AN00122-003

Flying shears allow the position lock of master and slave axes over defined distances with imposed accelerations and decelerations.

Usually there is a requirement to synchronize the speed of these axes over a specific distance traveled by one of the axes

Traditionally this might be achieved via a CAM, Mint however provides a much simpler way to do this – using the Mint FLY keyword.

Introduction

Synchronization between axes may occur with respect to linear output motion on both the master and slave axes (refer to application note AN00116 for an example of this) or alternatively the surface speed of a rotary slave axis may be synchronized to the linear surface speed of the master axis.

This application note details one of many possible rotary applications, an embossing wheel, and illustrates how the Mint FLY keyword eases application development.
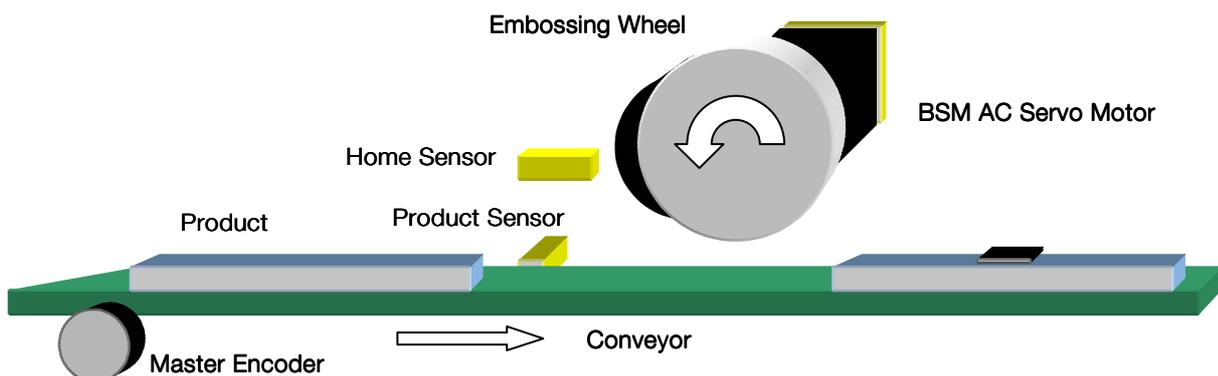
A conveyor moves the products to be embossed at a constant linear velocity through the machine. These products are randomly spaced (but with a minimum gap). The conveyor may be driven by an induction motor and variable frequency drive system or may even be controlled by a second servo drive/motor. An encoder detects the position and speed of the conveyor as it runs. This is usually termed the 'master' or 'auxiliary' encoder.

An embossing wheel, directly coupled to the motor shaft, is configured as a slave axis driven via a servomotor.

The motion controller (or an integrated motion controller and servo drive such as MicroFlex e150) monitors the master encoder (wired to the auxiliary encoder input) and uses it to determine when the embossing wheel should begin its motion.

When an incoming product is detected, the wheel is accelerated to a speed that exactly matches the product linear speed and the product is embossed. This all takes place over a defined distance traveled by the conveyor (master distance). Once the embossing has taken place the slave is accelerated and decelerated to complete one revolution of the wheel before the next product has arrived. Again these actions take place over a specified conveyor travel or master distance.

The total distance traveled by the conveyor during this whole process is fixed and corresponds to the shortest possible distance between consecutive products. The wheel dwells at the beginning of the process (the wheel remains stationary whilst the conveyor continues to run) waiting for an interrupt from the product detection sensor to start the next cycle.

The figure below illustrates the relationship between the speed ratio of the two axes and the distance traveled by the conveyor (master distance). The shaded area indicates when the surface speed of the rotary axis exactly matches the linear speed of the conveyor:
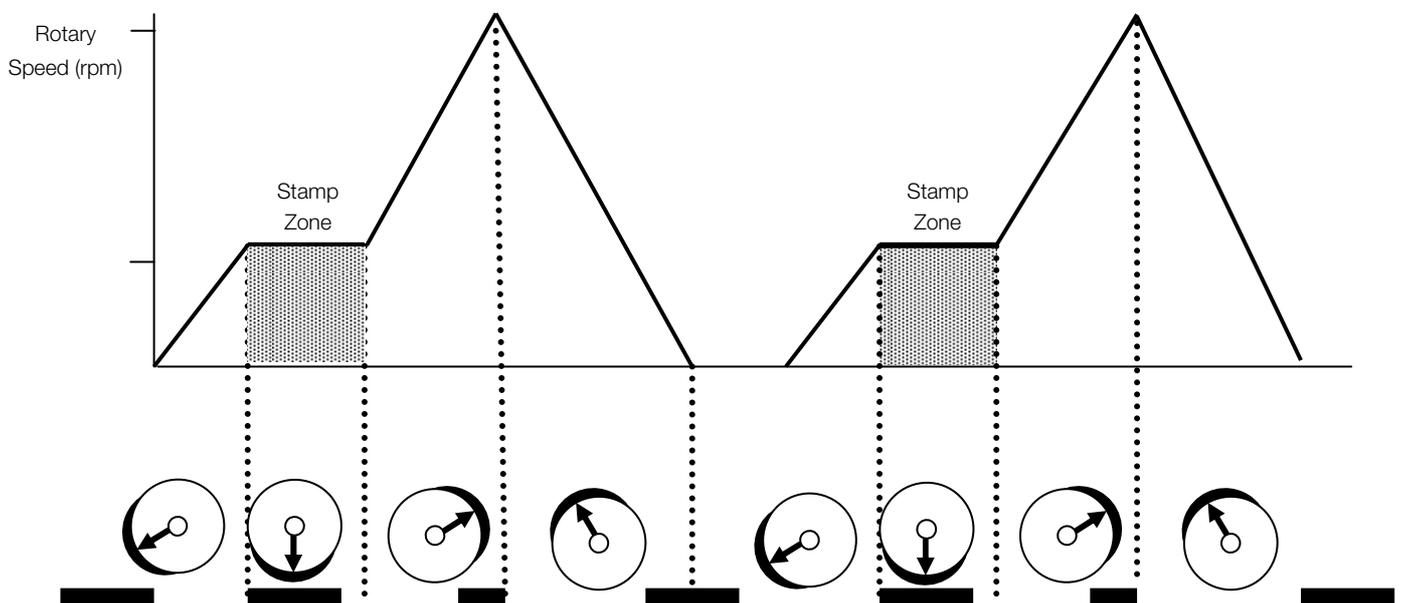
Each element of this profile can be split into what are termed 'segments' (this has also been illustrated on the figure above). The distance traveled by the conveyor for each segment can be specified in Mint by using the MASTERDISTANCE keyword.

For each segment the area under the curve corresponds to the distance traveled by the wheel (slave) axis. In Mint programs this distance/area can be specified using the FLY keyword.

If we examine a particular segment we can see that the segment area can be derived from:
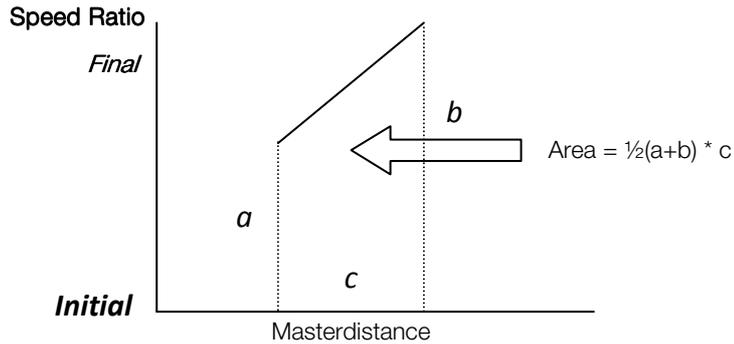
Area = Master Distance * (Initial Speed Ratio + Final Speed Ratio) /2

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 slave axis to the start position the sum of all of the flying shear segments must equal exactly the distance traveled in one revolution of the actuator (which in this case also equate to one revolution of the motor). By selecting appropriate values for MASTERDISTANCE the whole motion profile for the application can be written with just a few lines of Mint code.

Problems can arise if the values for any segment result in a fractional value (at an encoder count level) for FLY. This can be overcome by totaling the fly segments after initially calculating all of the segment values, comparing this with the required total, and then adding the difference back into one of the fly segments.

Important
Careful consideration needs to be given if a scale factor other than 1 is used for either the slave or master axes (e.g. the user may decide to scale the master axis to suit a user unit of linear mm). There are limits to the resolution that Mint products can represent floating point numbers. If a scale factor is found to be a recurring fraction for example, then the axes will drift out of phase over time. For these reasons it is common to use a scale factor of 1 for high accuracy applications.

Example Working
Consider our machine to have the following characteristics:

Conveyor Information:
341.333mm circumference wheel with 1024 ppr encoder (4096 in quadrature)
12 pulses per mm
1 pulse = 0.0833mm

Embossing Wheel Information:
Wheel Diameter = 100mm (314.159mm circumference)
Encoder Resolution = 16384 ppr
52.152 pulses per mm of embossing wheel travel
1 pulse = 0.0192mm

Product Information:
Product Length = 190mm
Emboss Length = 50mm

Distance from sensor to wheel = 60mm
Minimum Product Gap = 30mm

We can start to calculate our MASTERDISTANCE and FLY segment values from these details using the Generic formula

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

Segment 1 (ramp up to synchronous speed) ;
Initial ratio = 0 ; Final Ratio = 1 ; Masterdistance1 = 60mm
Fly1 = 60 * (0 + 1)/2 = 30mm = 30 * 52.152 counts = 1564.56 (1564) counts

Segment 2 (emboss at synchronous speed)
Initial Ratio = 1 ; Final Ratio = 1 ; Masterdistance2 = 50mm
Fly2 = 50 * (1 + 1)/2 = 50mm = 50 * 52.152 counts = 2607.6 (2607) counts

Segments 3 and 4 (remainder of cycle)

To calculate these two segments we need to consider the remaining distance to be traveled by the wheel, the initial and final ratios for each segment and the distance we can allow the master (conveyor) to travel.

So far the conveyor has traveled: 60 + 50 = 110mm.

The minimum master travel between product lead edges is 190 + 30 = 220mm.

We therefore have 110mm worth of conveyor travel to complete our embossing wheel cycle. We will round this down to 100mm (leaving 10mm worth of travel in hand).

We will split this remaining 100mm into two equal halves (50mm each) – we'll call this Masterdistance3. The speed ratio the slave axis will reach in completing the embossing cycle is unknown, we'll call this ratio R.

Using the generic fly formula:

Fly3 = Masterdistance3 * (Initial Ratio + Final Ratio)/2 = Masterdistance3 * (1 + R)/2
Fly4 = Masterdistance3 * (Initial Ratio + Final Ratio)/2 = Masterdistance3 * (R + 0)/2

We also know that Fly3 + Fly4 must total the remainder of our embossing cycle;

Fly3 + Fly4 = Complete cycle – Fly1 – Fly2 = 16384 – 1564 – 2607 = 12213 counts

Substituting for Fly3 and Fly4 gives;

(Masterdistance3 * (1+R)/2) + (Masterdistance3 * (R/2)) = 12213 counts

 Or

 50 * (1/2 + R) = 12213/52.152 = 234.1808mm

Therefore R = (234.1808/50) - ½ = 4.1836

We can now substitute back into our formulas for Fly3 and Fly4;

Fly3 = 50 * (1 + 4.1836)/2 = 129.5904mm = 6758.398 (6758) counts
Fly4 = 50 * (4.1836/2) = 104.59mm = 5454.577 (5454) counts

Power and productivity
for a better world™

ABB

For this application we have decided to work with a scale factor of 1 (our user unit is encoder counts). We also decided to round all our FLY segments to the nearest integer value so we need to check what will be 'left over' from a complete cycle when we total all of our calculated flying shear segments;

Remainder= Cycle – Fly1 – Fly2 – Fly3 – Fly4 = 16384 – 1564 – 2607 – 6758 – 5454 = 1

We therefore need to add this 1 count back into one of the segments to ensure that our embossing wheel always completes 1 exact revolution (we'll add it to the synchronous section).

The application may also require that the embossing position be adjusted. We cannot use a time delay after detection of the product to adjust this position, as this time will vary according to the speed of the conveyor. What we need to do is wait for a specified conveyor travel (master distance) before starting the embossing cycle.

This can be achieved with another flying shear segment, this time with a slave travel (FLY) of 0 over a variable master distance (MASTERDISTANCE) – this master distance is adjusted to produce the required embossing position. Our program would store this setting in a program variable (we'll call it nStartDelay – specified in encoder counts as our auxiliary encoder scale factor is set to 1).

If all distances are specified in encoder counts then our Mint code for the embossing wheel cycle therefore becomes...

```
'Embossing cycle…
MASTERDISTANCE(0) = nStartDelay
FLY(0) = 0
MASTERDISTANCE(0) = 60 * 12
FLY(0) = 1564
MASTERDISTANCE(0) = 50 * 12
FLY(0) = 2607 + 1  'Remainder to complete cycle added here
MASTERDISTANCE(0) = 50 * 12
FLY(0) = 6758
MASTERDISTANCE(0) = 50 * 12
FLY(0) = 5454
```

By sizing the move buffer slightly greater than the number of segments needed to complete an embossing cycle we can monitor the free space available in this buffer and ensure that the next cycle is pre-loaded whilst the previous cycle is still in progress. In this way, when the product detection interrupt occurs there is no need to delay the start of the cycle.

We'll also place the code to load a complete cycle into a subroutine so that it can be called from anywhere within our application.

If we add a homing routine to the start of our program (using Input 1 as the Home Input) and configure Input 0 (our Product detection input) as rising edge triggered then we have written the majority of the application with very little effort…

```
  'Setup initial dwell (this could be adjusted via an HMI)
Dim nStartDelay As Integer = 500

  'Setup scale factors
SCALEFACTOR(0) = 1
AUXENCODERSCALE(0) = 1  'Or ENCODERSCALE(x) = 1 depending on Mint product used

'Configure master axis
MASTERSOURCE(0) = _msAUXENCODER 'Or _msENCODER depending on Mint product used
MASTERCHANNEL(0) = 0  'Or encoder channel being used depending on Mint product used

'Configure Inputs
'Input 0 edge triggered, Input 1 level triggered
INPUTMODE(0) = 01
'Inputs 0 and 1 active high
```

```
INPUTACTIVELEVEL(0) = 011
'Input 0 rising edge triggered
INPUTPOSTRIGGER(0) = 01
'Neither input triggered on a falling edge
INPUTNEGTRIGGER(0) = 00

'Setup Home Input
HOMEINPUT(0) = 1

'Enable drive
CANCEL(0) : DRIVEENABLE(0) = 1

'Home in positive direction to sensor
HOME(0) = _hmPOSITIVE_SWITCH
PAUSE IDLE(0)

  'Allow 7 moves to be loaded into the move buffer
MOVEBUFFERSIZE(0) = 7
  'Load a complete embossing cycle
doLoad

  'Main Program Loop…
Loop
  If MOVEBUFFERFREE(0) >= 5 Then doLoad
End Loop

Sub doLoad ( )
  'Embossing cycle…
  MASTERDISTANCE(0) = nStartDelay
  FLY(0) = 0
  MASTERDISTANCE(0) = 60 * 12
  FLY(0) = 1564
  MASTERDISTANCE(0) = 50 * 12
  FLY(0) = 2607 + 1  ' Remainder to complete cycle added here
  MASTERDISTANCE(0) = 50 * 12
  FLY(0) = 6758
  MASTERDISTANCE(0) = 50 * 12
  FLY(0) = 5454
End Sub

Event In0
  GO(0)
End Event
```

Contact us

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

www.abbmotion.com
www.abb.com/drives
www.abb.com/drivespartners
www.abb.com/PLC

Power and productivity
for a better world™

ABB