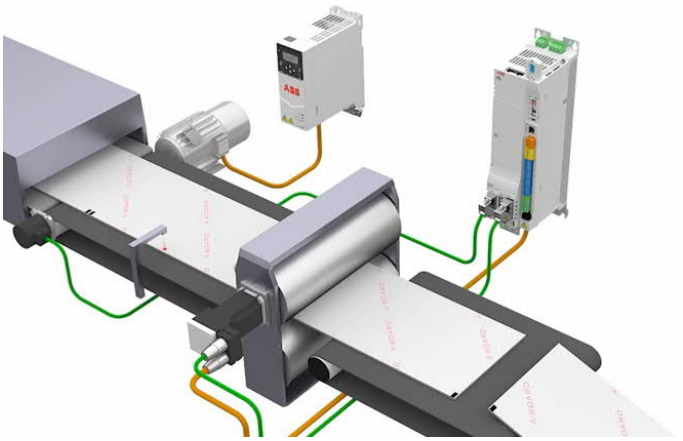
Motion Control Products

Application note
Rotary cutting at synchronous speed

AN00154

Rev H (EN)

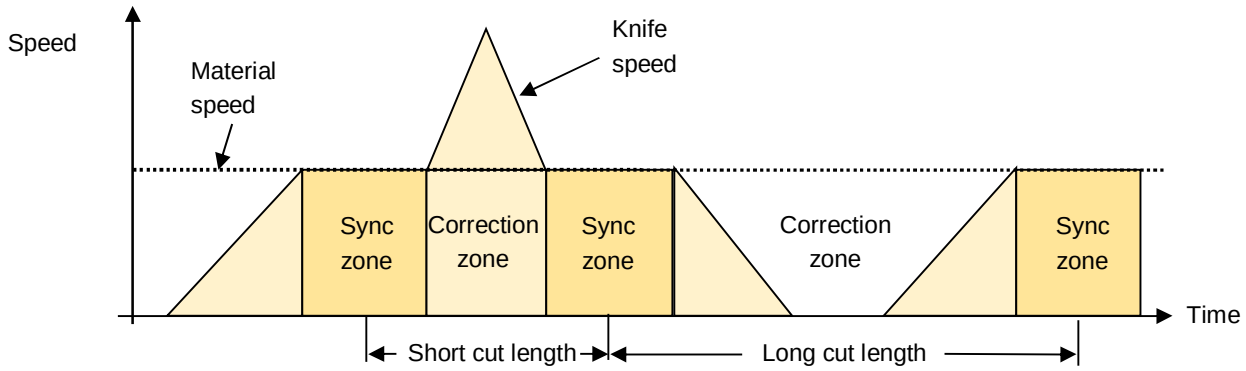
Many motion controllers and PLCs will use a CAM function to achieve a complex position/velocity profile such as that required to control the synchronous linear component of the velocity of a rotating knife. Mint provides a much simpler and more flexible method of achieving this via the FLY keyword



Introduction

In many continuous feed production processes there is often a requirement for the material to be cut to a specific length. Sometimes this may even include cutting the material in a precise position relative to a print registration mark. In these cases a rotating knife cylinder is often used. In its simplest form the knife cylinder periphery speed is simply geared 1:1 to the material linear speed (i.e. the infeed speed) over a defined rotary angle (the cut region) and the speed of the knife over the remainder of the cycle is adjusted to produce the required cut length. This is very often the case when the knife blade is angled (or on a helix).

This method is illustratred by the following diagram:

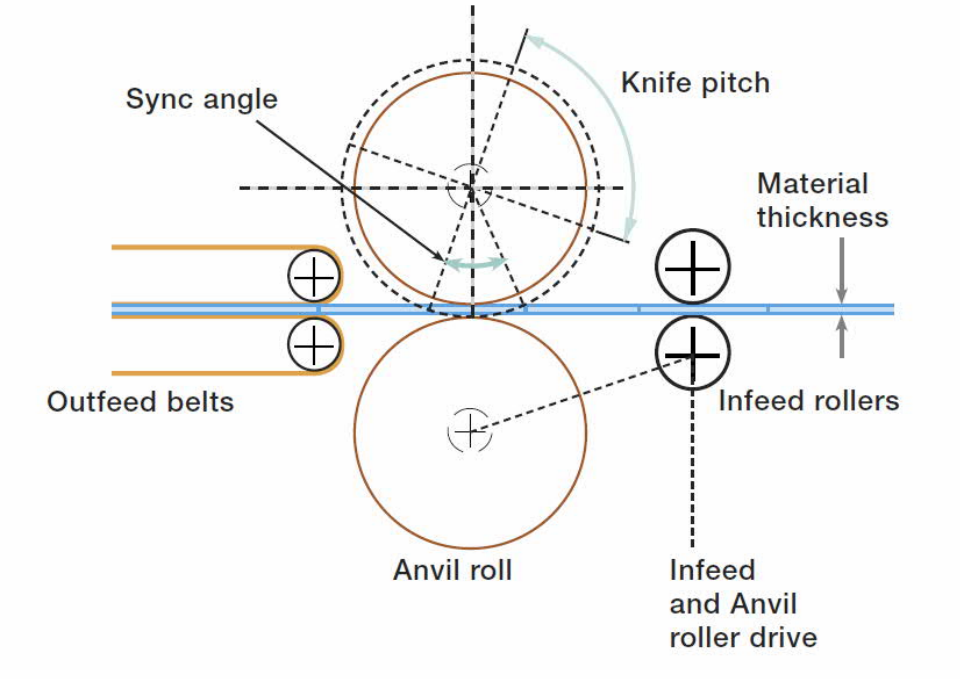


Programming a motion profile like this is made simple in Mint via use of the **FLY**, **FOLLOW** and **OFFSET** keywords (see Application Note AN00122 and AN00226 for a more detailed description and an application example). Whilst this method is adequate in a large number of applications the knife linear speed is only synchronous with the linear speed of the material at the exact point of cut. This can lead to material handling issues and less than perfect cut quality, particularly as the thickness of the product increases.

This application note sets out to illustrate how Mint can be used to control the speed/position of a rotary knife so that the linear speed of the knife is exactly matched to the linear speed of the material over the entire synchronous region, thereby improving material handling and cut quality.

Rotary Knife Theory

The Rotary Knife consists of a servo motor driving a knife cylinder (usually via a mechanical transmission system such as a gearbox and/or belts and pulleys).



The number of knives fitted to the knife cylinder may vary depending on the product lengths to be produced. In the example above four knifes have been fitted into the cylinder.

If we consider the linear speed of the knife roll surface (in the direction of the material being cut) the instantaneous velocity is given by the formula:

$v = \omega r \sin\theta$ (where θ is in radians and r is the radius of the knife roll)

The knife will be in contact with the material throughout the sync angle period. In this region the motion controller must force the knife to follow the inverse of the $\sin\theta$ function to maintain constant linear velocity. Note that at the bottom dead centre, $\theta = \pi/2$ (90 degrees. Mint controller trigonometric functions use degrees so we will relate all angular displacements in degrees from now on.

The synch angle can be calculated according to the material thickness (m) and the knife radius (r):

Synch Angle = $2 * (\cos^{-1}((r-m)/r))$

If we scale our knife axis into units representing linear travel around the circumference (e.g. mm or inches) and we also scale the material travel into the same linear units then the ratio between the two axes becomes…

$1 / \sin(\theta)$

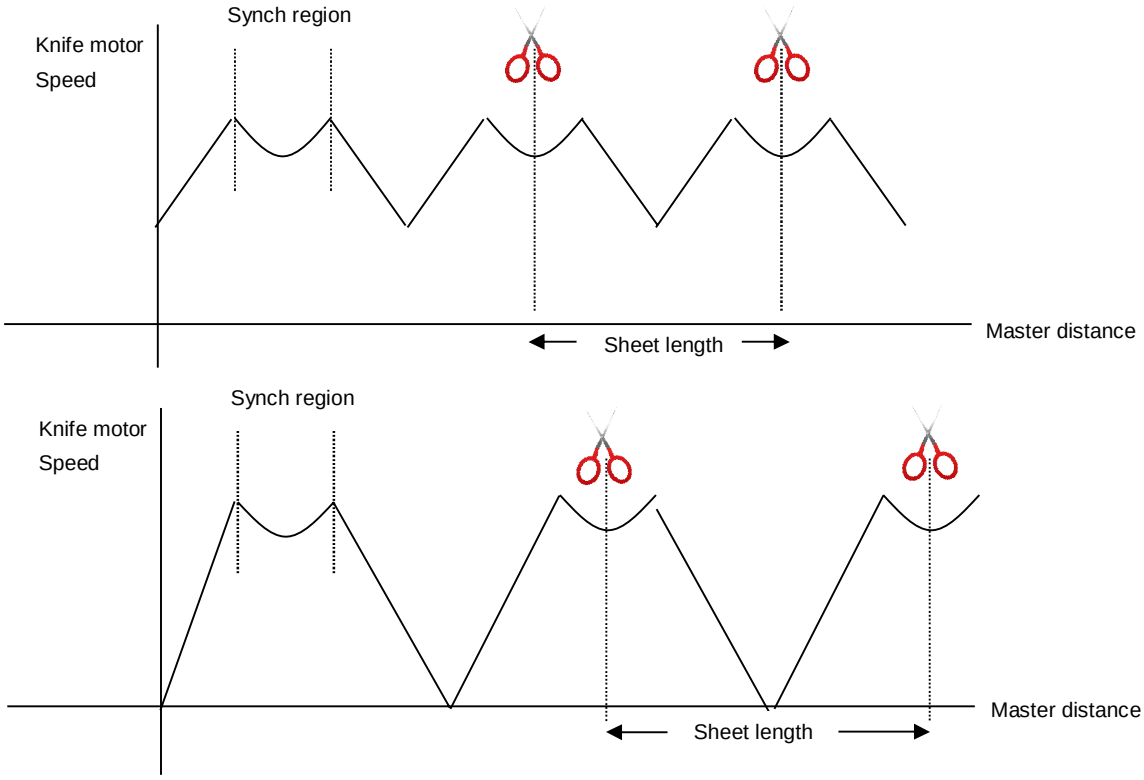
…where θ is 90 degrees at the cut point (bottom dead centre of the knife cylinder in our example above). i.e. When the knife is at the bottom dead centre the ratio in linear speed between the two axes is 1 / Sin(90) = 1. As the angle approaches or departs from 90 degrees $\sin(\theta)$ decreases and hence the speed ratio of the knife increases to keep the linear velocity component matched to the material speed.

By initially setting θ to (90- Synch Angle/2) and incrementing θ by a fixed increment (e.g. 2 degrees) we can therefore iterate through the synch section of the profile and calculate the speed ratio between the knife roll motor and material at each of these angles.

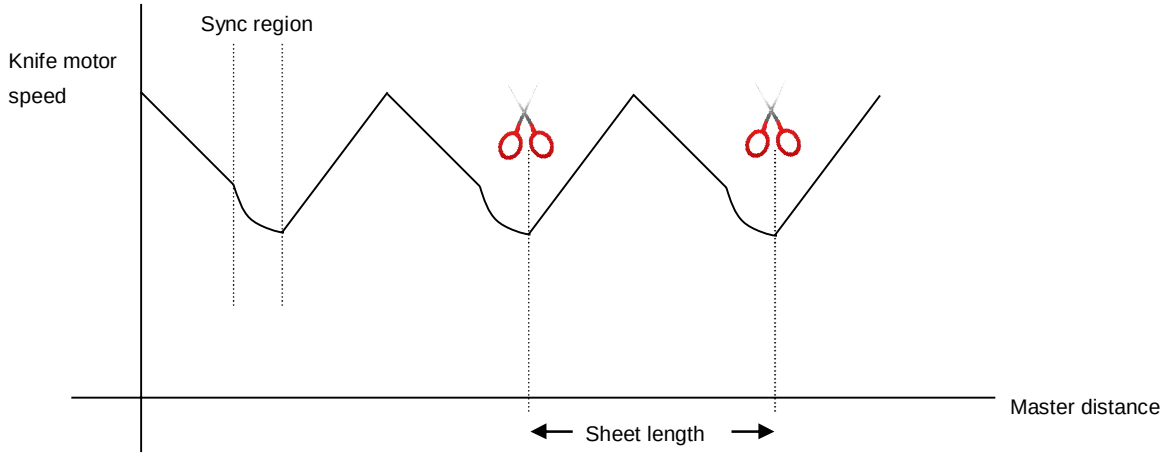
As was iillustrated by Application Note AN00122, the Mint FLY keyword allows the motion controller to synchronize the position of the slave axis (FLY) over a specific distance traveled by the material (MASTERDISTANCE).

In this case FLY is the increment in θ we used earlier (e.g. 2 degrees) converted into the equivalent travel around the knife circumference (e.g. FLY = 2 * Knife Circumference / 360) so using the general formula for Mint Flying Shear segments (FLY = MSD * (IR + FR) / 2) – where IR is the Initial Speed Ration and FR is the Final Speed Ratio – we can calculate the MASTERDISTANCE (material travel) to be used with each FLY segment.

Having calculated a MASTERDISTANCE for each FLY segment we can see that the total distance travelled by the material during the synchronous cut is the sum of these MASTERDISTANCEs (and of course the total travel of the knife is the synch angle). All that remains now is for the motion control code to load flying shear segments to rotate the knife the remainder of one knife pitch over the remainder of the required cut length (this process is detailed later).
The resulting motion profile will look something like the following diagrams depending on cut lengths. As the cut length increases the knife speed outside of the synch region will decrease until it reaches a point where the knife stops and starts to dwell:



In some applications it is not necessary to maintain the synchronous profile once the bottom dead centre has been reached (i.e. once the cut has taken place). In this case the profile becomes non-symmetrical as shown below, but with the advantage on shorter cut lengths that the maximum speed the axis needs to achieve is reduced…



The sample code included with this application note includes a parameter to allow the user to select between these two modes of operation for the sync region.

Calculating FLY segments for the synchronous region

As we've already seen, these calculations are dependant on the radius of the knife cylinder (r) and the material thickness (m). The material thickness and knife radius together determine the synch angle. It is common to allow the operator to adjust this synch angle (e.g. via an HMI) so in this example we'll assume the code operates on an entered synch angle directly (rather than calculating this from 2 * (Cos^{-1}((r-m)/r)) but either method is acceptable.

The knife radius and synch angle are passed as parameters to a re-usable Mint subroutine. Arrays passed by reference allow the subroutine to return data that describes the MASTERDISTANCE**,** FLY and speed ratio values for each segment of motion. These arrays can then be used later in the application to load FLY and FOLLOW segments into the knife axis' move buffer. Other parameters passed by reference are used by the subroutine to calculate and pass back to the calling routine commonly used distances that will be used later in the application to determine whether the knife needs to stop or not to produce the required cut length.
To calculate the speed ratio between the two axes during the sync region the code increments through the synch region in two degree steps, i.e. nInc is specified as 2 by default (therefore the synch angle must be specified as an even value)…

```
'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
```

The parameter 'nSyncMode' that is passed to this subroutine determines whether only the first half of the sync region should be used (nSyncMode = 0) or whether both halves should be used (nSyncMode = 1).

Once the speed ratios (and number of these that have been calculated) are known the subroutine than then load the arrays passed as parameters with calculated data for the FLY and MASTERDISTANCE values that may be used by the code later…

```
'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
```

Calculating FLY segments for the remainder of cut length

The subroutine used to calculate the FLY and MASTERDISTANCE values for the synchronous cutting region also calculates a number of other parameters…

a) Distance travelled by the master (material) during the synchronous cutting phase (The knife must travel the remainder of one knife pitch in the remaining material length. The remaining material length is the required cut length – the travel during the cut)
b) How far the material would travel if the knife decelerated to a stop after completing the synchronolus region, coming to a halt half way between knives
c) How far the material would travel if the knife accelerated up to the start of the synch region, starting from half way between knives
d) How far the knife travels from the end of the synch region to the midpoint between knives (i.e. the stop point for long cut lengths)
e) How far the knife travels from the midpoint between knives to the start of the next synch region

If the required cut length is greater than the sum of (a) + (b) + (c) then the knife must stop completely (with the knife away from the cut point – i.e. half way between knives) to allow more material through the cutting area before restarting (this is known as a 'dwell'). In this case a FLY (knife travel) of zero degrees is loaded over a MASTERDISTANCE which makes up the difference between the required cut length and the material travel during deceleration and acceleration of the knife roll.

If the required cut length is less than the sum of (a) + (b) + (c) then the knife will either slow down slightly or increase in speed between each knife and there is no need to load the 'dwell' segment.

The formulae for these calculations are complicated by the ability to switch between either a half or full sync region, but this is covered by the sample code in the application note that solves the quadratic equation necessary to derive all of the required data….

```
'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
```

The sample Mint program included with this application note details how to load these flying shear segments. The segments to take the knife to the midpoint between knives are loaded as part of the doLoadSynch subroutine. The segments to take the knife back to the start of the next synchronous region are loaded as part of the doLoadRemainder subroutine.

Compensating for floating point inaccuracies

The knife roll is scaled in 'linear' units around the circumference. Because this involves the use of the mathermatical constant Pi this will always result in a fractional scale factor that cannot be represented exactly.

Example:
Motor provides 10,000 encoder counts per rev and is fitted with a 10:1 gearbox. The knife radius is 90mm. Scalefactor for linear travel in mm around the circumference is therefore…

(10000 * 10) / (2 * _Pi * 90) = 176.83882565766148418764862596946...

As the application continues to load flying shear segments using a fractional scale factor the knife axis will start to drift out of position, eventually causing the knife blades to cut the material whilst the knife velocity is not synchronous with the linear speed of the material.
To compensate for this the sample Mint program generates a digital output whenever a specific knife blade should be halfway through a cut cycle. Digital output 0 is used in the sample code, it must be one of the first three outputs in order for the latching to operate correctly. A latch event is configured that latches the knife axis encoder value at this point in time. If the knife has not drifted this encoder value should be consistent on every cycle (providing there are a whole number of encoder counts in one revolution of the knife – if there are a fractional number of encoder counts then a second mechanism must be included to zero the knife encoder every revolution rather than just relying on the programmed ENCODERWRAP).

The ideal encoder value is compared against the encoder value latched via the digital output and the error between the two is then added in as an **OFFSET** which is applied over a portion of the material travel that occurs outside of the knife sync region. This offset can be started by the latch event as the knife has finished cutting at this point in time. The corrections are very small so have very little effect on the knife velocity profile…

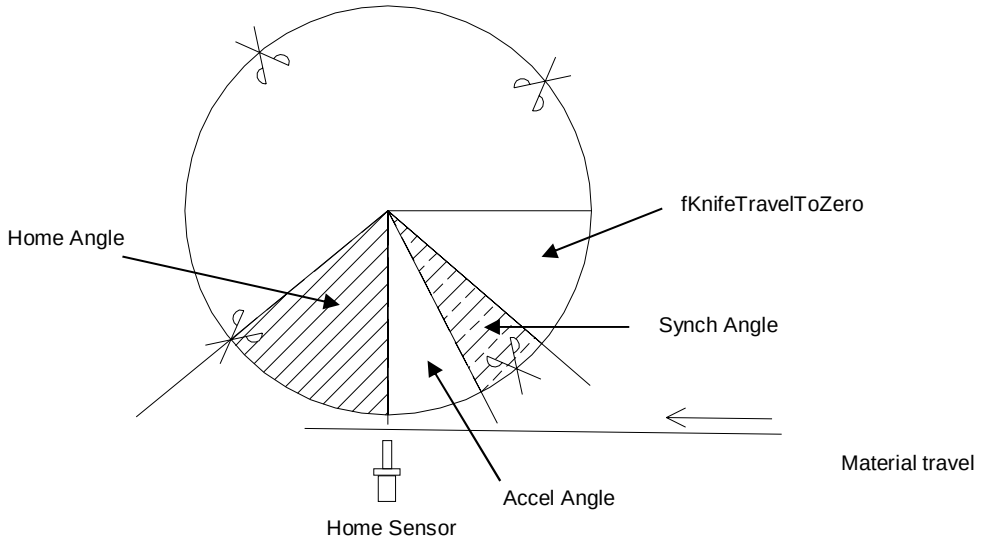
Cutting in register

To ensure the knives always cut at a synchronous linear velocity we must ensure that the physical position of the knife cylinder is not moved with respect to the programmed velocity profile (i.e. it is not permissible to simply offset the position of the knife). Therefore registration corrections must be made by artificially shortening or lengthening, as required, the cut length being produced.

Starting the knife

Before the system can operate the knife must be homed (so the motion controller knows where one of the knife blades are – usually there are as many home flags fitted as there are knife blades).

The home routine includes a 'home offset' parameter that allows the operator to ensure that after detecting the home sensor the knife roll moves the knife blade away from the anvil roll (so that material can pass under the knife when the machine is first started). This home offset is usually adjustable (e.g. from an HMI) so that (together with a parameter known as the 'Accel Angle' that is often fixed) the operator has a method to adjust the physical knife position so that it cuts during the synchronous section of the knife velocity profile (usually in the exact centre of this part of the profile).



When starting the system in registration mode it is usually desirable for the knife to startup so that the first cut occurs in registration. This is easily achieved in Mint using the TRIGGER**...**. set of keywords to start the knife at a known position of the material. The TRIGGERVALUE is calculated when the first registration mark is detected and takes into account how far the material will travel whilst the knife is accelerating from stationary up to the start of the synchronous region of the profile.

Please contact cn-motionsupport@cn.abb.com if you require further information regarding incorporating registration control.

Sample Mint application code

The sample Mint program included with this application note comprises a Mint (mnt) program file and a Device Configuration File (dcf) for use with a Nextmove e100 motion controller. The code is written using conditional compilation to allow it to also work on MicroFlex e190 and MotiFlex e180 drives. So that the code works on all platforms encoder channel 2 has been selected for the material encoder input.

If using a NextMove e100 to run the example the dcf file expects a MicroFlex e190 drive to be connected via Ethernet Powerlink (setup as node 3). If using NextMove e100 the knife axis is axis 3, if using one of the intelligent drive products the knife axis becomes axis 0.

Be sure to check that the encoder count direction for encoder channel 2 (the material encoder) is positive (i.e. the application expects the material to be travelling in a positive direction).

Actuating digital input 4 will cause the Knife axis to home (a home sensor should be wired to digital input 0 on whichever drive is being used). The machine simulation will not run unless the knife has been homed. Once the knife axis has homed the knife axis will start to run based on the speed/position of the material encoder.

Use the Data Watch Window in Mint Workbench to modify the value of NETFLOAT(13) at runtime – this simulates changes in desired cut length being entered by the operator.
NETINTEGER(1) can be used to modify the number of knives, NETINTEGER(2) can be used to adjust the sync angle and NETINTEGER(3) can be used to set whether a half (0) or full (1) sync angle is profiled. Always set the sync angle parameter as if the full angle is being used, even when only using the first half.

Default operating parameters for the system (e.g. knife radius, number of knives) are set in the doInitKnife subroutine so these can be adjusted to suit the application also.

The example Onerror code is very simple and just reads the error information, displays some diagnostics via Workbench about the error and then restarts the program. This can be expanded/enhanced as required if necessary.

Contact us

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

new.abb.com/drives/low-voltage-ac/motion
new.abb.com/drives
new.abb.com/channel-partners