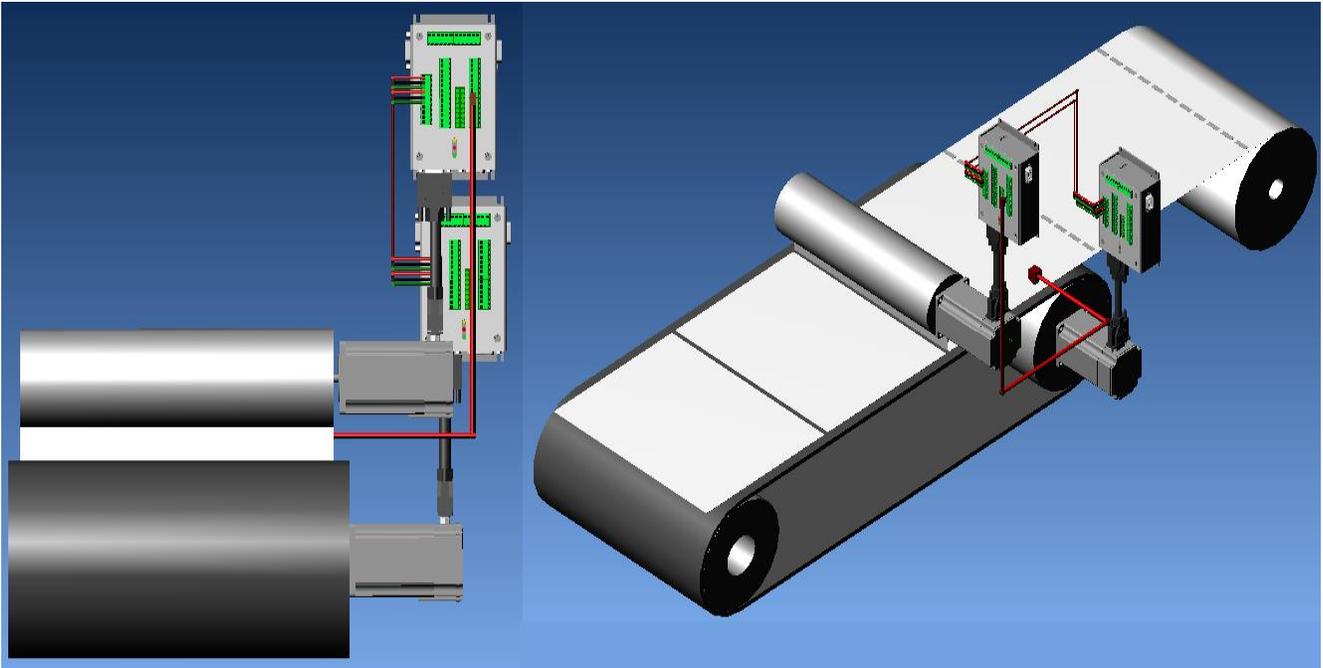# Rotary Knife

Included files:

- QCI-AN034 Rotary Knife.pdf: This document
- Rotary Knife – Simple - Fixed.qcp
- Rotary Knife – Simple - Dynamic.qcp
- Rotary Knife - Registration.qcp
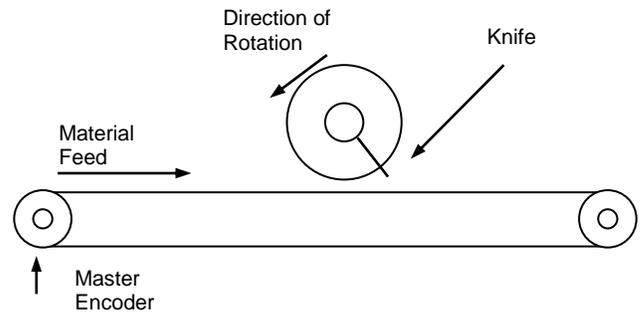- Rotary Knife - Simple SN.qcp

Introduction

Rotary knife systems are widely used to cut, seal or perforate material moving along a conveyor. Rotary knife system consists of a cylinder with a knife blade fastened along the longitudinal axis. As the cylinder rotates, it cuts the material passing underneath it. During the cut, the tangential knife velocity must match the linear speed of the material. During the rest of the revolution, the knife must either speed up or slow down to be ready for the next cut.
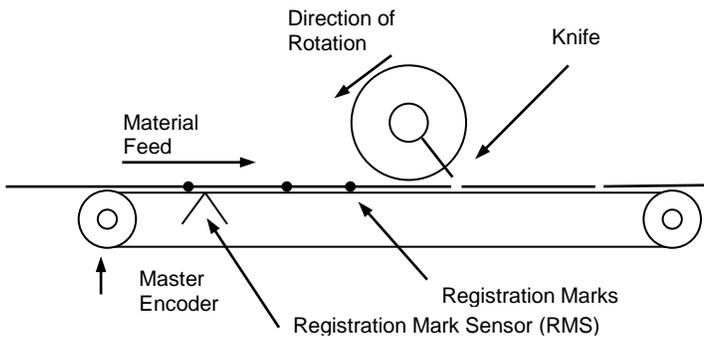
Most rotary knife applications are continuous operation with material continuously moving along the conveyer belt. The knife axis (slave) is geared electronically to the feed axis (master) so that the material cut lengths are independent of the material feed speed.

This document will discuss two types of rotary knife applications, Simple and Registration.

In the Simple application, the material length is fixed in the program or selected by an operator and the cuts are made continuously.

Direction of Rotation

Knife

Material Feed

Master Encoder

Rotary Knife - Simple

Direction of Rotation

Knife

Material Feed

Master Encoder

Registration Marks

Registration Mark Sensor (RMS)

Rotary Knife - Registration

The Registration application adds a Registration Mark Sensor (RMS) that detects a registration mark on the material where the cut is to be made. The material length is variable and the RMS may be several material lengths from the knife.

Rotary Knife - Simple

The material feed is continuous. The knife speed matches the material speed for the Cut Segment(CS) and then, depending on Material Length (Mn), speeds up or slows down during the Return Segment(RS) to get into position for the next cut.

Mn is set by the operator or hard coded in the program.

Knife Return Segment (Krs)

Knife Cut Segment (Kcs)

Knife Radius (R)

Mrs

Mcs

Mn
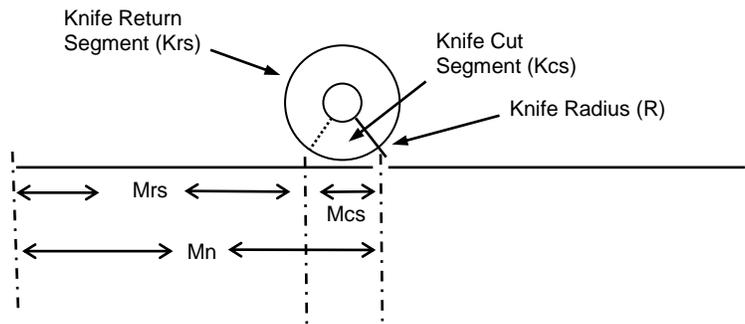
Application Specific Parameters

- Knife Counts/Rev (Kcpr): Kcpr is the number of knife encoder counts for 1 revolution of the Knife.

   Kcpr = <user defined>

- Knife Cut Segment (Kcs): Number of knife encoder counts for Cut Segment (CS). This is somewhat arbitrary. It needs to be long enough to clear the material at both ends. It also needs to be long enough to allow the knife to settle into matching speed prior to engaging the material. The following is a good default number:

   Kcs = Kcpr/4

- Knife Return Segment (Krs): Knife encoder counts for Return Segment (RS). This is just the rest of the knife's rotation.

   Krs = Kcpr - Kcs

- Cut Segment Scale Factor (SFcs): Scale Factor (SF) for CS.
  The Rotary Knife is an electronic gearing application, therefore a key calculation that needs to be made is the EGM command's Scale Factor (SF). SF is basically a gear ratio between the knife (slave) and master. See Application Note "QCI-AN019 Electronic Gearing" for more information. The general equation of SF is:

   SF = SF1 * GR

  where SF1 is SF for a 1:1 ratio. In other words if SF=SF1, the knife axis would move the same number of encoder counts as the master. SF1 is not user settable and is automatically calculated in the program.

Gear Ratio (GR) is the slave axis movement (knife encoder counts) movement over the given master axis movement (master encoder counts) that synchronizes the knife to material passing beneath it. GR should be less than 1.0 for smooth operation. GRs of 1.0 and greater will require servo tuning. Therefore, the Cut Segment Scale Factor (SFcs) is as follows:

$$SFcs = SF1 * GR$$
$$SFcs = SF1 * Kcs/Mcs$$

$$GR = Kcs/Mcs <= 1.0$$

- Material Length (Mn): Material Length in Master Encoder Counts. In the "Rotary Knife-Simple-Fixed.qcp" example, Mn is fixed at the beginning of the program file. In the "Rotary Knife-Simple-Dynamic.qcp" example, Mn can changed from a host (i.e. HMI) in real time.

$$Mn = <user defined>$$

- Master Cut Segment (Mcs): Master encoder counts for CS. Mcs is calculated from Kcs using GR.

$$Mcs = Kcs/GR$$
$$Mcs = Kcs * SF1/SFcs$$

- Master Return Segment (Mrs): Master encoder counts for RS.

$$Mrs = Mn - Mcs$$

Setup

1) Initialize the servo as described in the User Manual chapter 1. The application assumes the knife rotates in the servo's "positive" direction.   You may need to edit the DIR command in the Initialization Wizard to change the servo's positive direction.

Servo
Positive
Rotation
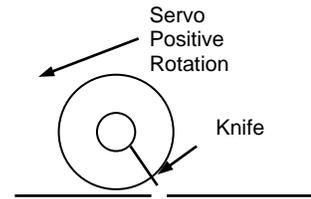
Knife

2) Attach the servo to the knife axis, aligning the encoder index with the knife blade at the end of the Cut Segment. This can be done by either physically aligning the index mark on the motor's shaft or by running a program to make the motor move to its index. See the example file included with QuickControl:

QCI Examples\Homing Routines\Home To Encoder Index.qcp

3) Wire the servo for electronic gearing by connecting the master encoder A/B signals to the controller's I/O #6 and #7. See Application Note "QCI-AN019 Electronic Gearing" for more information. It is assumed the master encoder rotates in the positive direction. Reverse the A/B signals to reverse your master encoder.

Rotary Knife-Simple-Fixed.qcp

Requires:

- QuickControl 4.64
- SilverDust Firmware 37
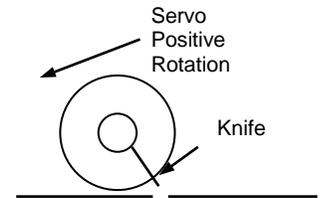
The following is an overview of "Rotary Knife-Simple-Fixed.qcp". This QCP is well documented and should be easy to follow. The following overview in combination with the rest of this app note and the QCP's source code remarks should allow the user to modify the QCP for their specific needs.

This program file is composed of the following programs:

Init
Initialize application specific and derived parameters (registers) and setup the servo for an electronic gearing application. On completion, "Init" runs the "Home to Index" program. The only registers you should have to edit for your application are Kcpr, SFcs, and Mn.

Home to Index
Homes the servo to its internal index pulse. Note how electronic gearing is used to rotate the knife at the Cut Segment's matching speed. This ensures the knife will not bind during the homing process. Once the index is found, the encoder is zeroed to the index location and the "Rotary Knife" program is run.

Rotary Knife

This program uses the EGM command to run in the rotary knife function in the background.

Knife Return Segment Geared (Krsg) is the distance the knife would move during the RS if it stay geared at the synchronous scale factor of SFcs. This is used as a "base" distance to calculate how much more or less the EGM command's Trapezoid Move feature needs to move during the RS to be ready for the next cut.

$$Krsg = Mrs * GR = Mrs * SFcs/SF1$$

Note that we have to use the 64 bit multiple and divide to not overflow the 32 bit registers.

The EGM Trapezoid Move Registers define the distance the knife will move during the RS.

- Start Position (SP)= 0: Start immediately.
- Master Distance (MD)=Mrs
- Move Ramp Distance (MRD)=MD/100: Nice easy acc/dec.
- Trapezoid Move Distance (TMD)=Krs-Krsg: The distance TMD is the "extra" distance the knife will travel during the RS. Note this is in addition to the distance traveled as normal electronic gearing (Krsg). Negative values of TMD mean the knife will rotate slower than during the CS.
- Master Modulo Distance (MMD)=Mn: Modulo every Mn.

Line# Oper	Label	Command
1:REM		Rotary Knife Operation
2:REM		Knife Return Segment Geared (Krsg) Distance Knife will move during Mrs due to GR Krsg = Mrs * GR Krsg = Mrs * SFcs/10,000,000
3:CLX		T64L[38] (U64) = Mrs[23] * SFcs[18]
4:CLD		Krsg[30] = (T64L[38] (U64))/10000000
5:REM		EGM Params (see app note)
6:WRP		Write 0 to "SP[42]" Register
7:CLX		MD[43] = Mrs[23]
8:CLD		MRD[44] = MD[43]/LO(100)
9:CLD		MRD[44] = Max of MRD[44] or 1
10:CLX		TMD[45] = Krs[24] - Krsg[30]
11:CLX		MMD[46] = Mn[19]
12:REM		EGM Run EGM in modulo mode to automatically repeat every Mn.
13:EGM		Electronic Gearing Mode: Acc Factor in "AF[40]" Scale Factor in "SF[41]" Trapezoid Move Enabled Start Position (SP) in Register 42 Master Distance (MD) in Register 43 Master Ramp Distance (MRD) in Register 44 Trapezoid Move Distance (TMD) in Register 45 Modulo Master Distance (MMD) in Register 46 Override Existing Move Relative Master Position Modulo Enabled

EGM is launched and runs in the background while the "Main Loop" is executed for user specific code.

Rotary Knife-Simple-Dynamic.qcp
Requires:
- QuickControl 4.64
- SilverDust Firmware 37
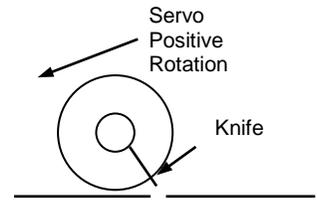
The following is an overview of "Rotary Knife-Simple-Dynamic.qcp".
This is very similar to "Rotary Knife-Simple-Fixed.qcp" except the EGM
command is not run in modulo mode, but, instead, is executed every Mn to allow for Mn to
change on the fly (i.e. using an HMI).

This program file is composed of the following programs:

Init
Same as "Rotary Knife-Simple-Fixed.qcp", except Mn can be thought of as an initial value.

Home to Index
Same as "Rotary Knife-Simple-Fixed.qcp".

Rotary Knife
EGM is executed for each Mn or, in other words, for each
knife revolution. Each pass through "Main Loop" is one
revolution.

Mrs is re-calculated in case Mn has changed.

The EGM Trapezoid Move Registers define the distance the
knife will move during the RS. All the parameters are the
same "Rotary Knife-Simple-Fixed.qcp" except there is no
MMD and EGM is configured execute only one Trap Move.

While EGM is executing, the program waits for one
complete revolution of the knife. After which, the knife's
position is modulo down by one revolution and we jump
back to the start for the next pass.

Line# Oper	Label	Command
1:REM		Rotary Knife Operation
2:REM	LOOP	=============== Main Loop ===============
3:REM		Re-calculate Mrs in case Mn has changed Mrs=Mn-Mcs
4:CLX		Mrs[23] = Mn[19] - Mcs[25]
5:REM		Knife Return Segment Geared (Krsg) Distance Knife will move during Mrs due to GR Krsg = Mrs * GR Krsg = Mrs * SFcs/10,000,000
6:CLX		T64L[38] (U64) = Mrs[23] * SFcs[18]
7:CLD		Krsg[30] = (T64L[38] (U64))/10000000
8:REM		EGM Params (see app note)
9:WRP		Write 0 to "SP[42]" Register
10:CLX		MD[43] = Mrs[23]
11:CLD		MRD[44] = MD[43]/LO(100)
12:CLD		MRD[44] = Max of MRD[44] or 1
13:CLX		TMD[45] = Krs[24] - Krsg[30]

14:EGM		Electronic Gearing Mode:
		Acc Factor in "AF[40]"
		Scale Factor in "SF[41]"
		Trapezoid Move Enabled
		  Start Position (SP) in Register 42
		  Master Distance (MD) in Register 43
		  Master Ramp Distance (MRD) in Register 44
		  Trapezoid Move Distance (TMD) in Register 45
		  Modulo Master Distance (MMD) in Register 46
		
		Override Existing Move
		Relative Master Position
		One Trapezoid Move Only

15:REM		Wait for one full rotation
16:CLX		Accumulator[10] = Target Position[0] - Kcpr[13]
17:JMP		Jump to "-1" If Last Calc Was Negative TRUE
18:REM		Modulo Knife
19:CLC		Targ - Kcpr[13], Pos - Kcpr[13]
20:REM		Loop forever
21:JMP		Jump to "LOOP"

Rotary Knife - Registration

Requires:

- QuickControl 4.64
- SilverDust Firmware 37

This version of the Rotary Knife uses a sensor (RMS) to read registration marks on the material to calculate the Material Length (Mn). Since there could be several registration marks between the knife and the RMS, the program needs to queue up several lengths.
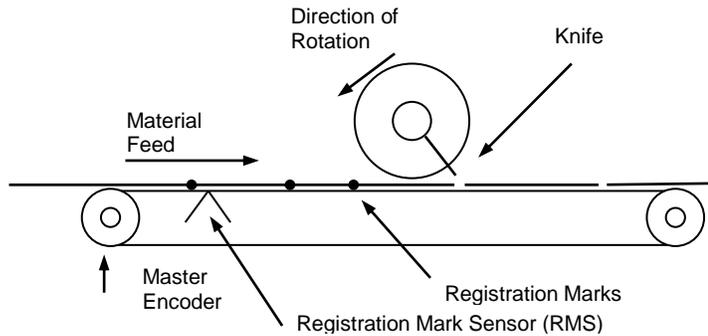
In the case where the RMS stops detecting registration marks (i.e. material is not present), the knife comes to a stop after the Cut Segment. Once a mark is read, the knife starts back up again to get into position for the next cut.

Additional Application Specific Parameters

- Maximum Material Length in Master Encoder Counts (Mmax): Mmax is also the distance from the RMS to end of Mcs.
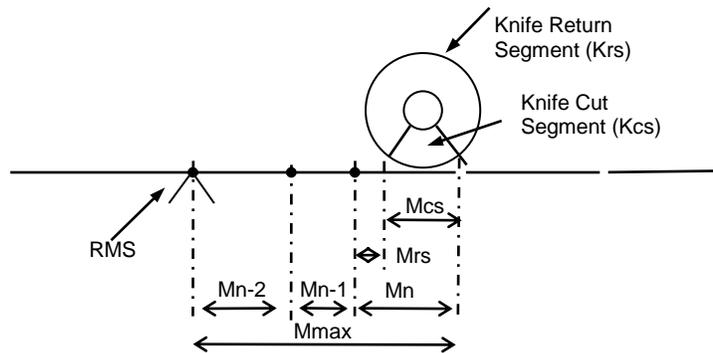
  Mmax = <user defined>

- Minimum Material Length (Mmin): You may change Mmin to an Application Specific Parameter if you have a particular Mmin you want to enforce. The smaller Mmin the faster the knife axis will have to go. Be careful not to exceed the knife's maximum speed. By default Mmin is:

  Mmin = Mcs
  Mmin < Mmax
  Mmin = <user defined>

  Material Length (Mn): Mn is automatically calculated by reading the difference between two registration marks.

  NOTE: Because the difference between two readings is being used to calculate Mn, the RMS delay or de-bounce time is cancelled out. Therefore, a high-speed RMS is not required.
  NOTE: The 2's complement math makes the equation work even through master encoder rollover.

Setup

1) Do Simple version steps 1-3.

2) Wire RMS to I/O#1 with an HIGH meaning registration mark detected. The only place this input is used is in the WBE command at the beginning of "T2: Queue Mn". Modify this WBE command if your RMS is not active HIGH on I/O#1.

3) Edit the "Application Specific Parameters" section of the "Rotary Knife - Registration.qcp" "Init" program. Edit the registers using information from your application.

Rotary Knife – Registration.qcp
The following is an overview of the Rotary Knife - Registration.qcp file. This program file is composed of the following programs.

Init
Same as "Rotary Knife-Simple-Fixed.qcp" except:
- Moved Mn and Mrs to "Rotary Knife" program.
- Added Mnmax/Mnmin which defines the longest and shortest material lengths.

Home to Index
Same as "Rotary Knife-Simple-Fixed.qcp" except:
- Thread 2 (Queue Mn) is started after index found (see below for details).

Rotary Knife
This program is similar to the same program in "Rotary Knife-Simple-Dynamic.qcp" except that Mn is retrieved (popped) from the queue "Q" (see below for details on Q).

If Q is empty, SF is set to zero to stop the knife and we jump back to the start.   Once a valid Mn is received, the EGM command is used the same as "Rotary Knife-Simple-Dynamic.qcp".

Note, the first Mn after an empty Q, causes the knife to start from a stop. The EGM first ramps up to speed using AF and SF and then executes the Trap Move to move the knife to the correct place for the cut. You will see the knife start up quickly as it ramps up to speed and then slow down as the Trap Move is executed. This is normal.

Line# Oper	Label	Command
2:REM	LOOP	=============== Main Loop ===============
3:REM		Add user specific code here
4:REM		Pop Mn from Queue
5:CLX		Mn Q[50] Tail -> Mn[19]
6:REM		Hold motion if Q empty This is done by setting SF=0
7:JMP		Jump to "VALID MN" If Last Calc Was Zero TRUE
8:WRP		Write 0 to "SF[41]" Register
9:REM		Loop back
10:JMP		Jump to "LOOP"
11:REM	VALID MN	Valid Mn popped from Q
12:REM		Master Return Segment (Mrs) Mrs=Mn-Mcs
13:CLX		Mrs[23] = Mn[19] - Mcs[25]
14:REM		Knife Return Segment Geared (Krsg) Distance Knife will move during Mrs due to GR Krsg = Mrs * GR Krsg = Mrs * SFcs/10,000,000
15:CLX		T64L[38] (U64) = Mrs[23] * SFcs[18]
16:CLD		Krsg[30] = (T64L[38] (U64))/10000000
17:REM		EGM Params (see app note)
18:CLX		SF[41] = SFcs[18]
19:CLD		AF[40] = SF[41]/LO(1000)
20:WRP		Write 0 to "SP[42]" Register
21:CLX		MD[43] = Mrs[23]
22:CLD		MRD[44] = MD[43]/LO(100)
23:CLD		MRD[44] = Max of MRD[44] or 1
24:CLX		TMD[45] = Krs[24] - Krsg[30]

25:EGM		Electronic Gearing Mode:
		Acc Factor in "AF[40]"
		Scale Factor in "SF[41]"
		Trapezoid Move Enabled
		  Start Position (SP) in Register 42
		  Master Distance (MD) in Register 43
		  Master Ramp Distance (MRD) in Register 44
		  Trapezoid Move Distance (TMD) in Register 45
		  Modulo Master Distance (MMD) in Register 46
		
		Override Existing Move
		Relative Master Position
		One Trapezoid Move Only

26:REM		Wait for one full rotation
27:CLX		Accumulator[10] = Target Position[0] - Kcpr[13]
28:JMP		Jump to "-1" If Last Calc Was Negative TRUE
29:REM		Modulo Knife
30:CLC		Targ - Kcpr[13], Pos - Kcpr[13]
31:REM		Loop forever
32:JMP		Jump to "LOOP"

T2: Queue Mn

Reading the RMS and queuing up the Mn's is done in the controller's 2nd program thread. This is actually a separate program running in parallel to the thread 1 programs (Init, Home to Index, and Rotary Knife). See Multi-Thread in User Manual for details.

This program calculates the Mn's and "pushes" them onto a queue to be later "popped" off and used by the Rotary Knife program.

The queue is first set to hold a maximum of 50 Mn's.

The Main Loop waits for RMS to go HIGH and then saves the master encoder position (reg 200) into a user register, "Reg Mark".

Mn is calculated by comparing the current "Reg Mark" with the "Last Reg Mark".

Mn = Reg Mark – Last Reg

Mn is then forced to be within the user defined min/max.

Mn is pushed onto Q. If more than 50 Mn's are pushed onto the queue, thread 1 is forced to load and run the Fault program and thread 2 halts.

| Line# Oper	Label	Command
1:REM		Thread 2: Queue Up Material Lengths (Mn)
2:REM		Initialize Mn Queue
3:CLD		Init Queue Mn Q[50] Size = 50 Accumulator[10] = Size
4:CLC		Clear Last Reg Mark[49]
5:REM	LOOP	Main Loop
6:REM		Wait for Reg Mark (RMS)
7:WBE		Wait On Bit Edge Until "I/O #1" is HIGH/TRUE
8:CLX		Reg Mark[48] = External Encoder Position[200]
9:REM		Mn=This RM - Last RMS
10:CLX		Accumulator[10] = Reg Mark[48] - Last Reg Mark[49]
11:REM		Clip Between Mmin and Mmax
12:CLX		Accumulator[10] = Min of Accumulator[10] or Mmax[32]
13:CLX		Accumulator[10] = Max of Accumulator[10] or Mmin[31]
14:REM		Push onto Queue Fault if push fails
15:CLX		Mn Q[50] Head <- Accumulator[10] Accumulator[10] = Size
16:JMP		Jump to "PUSH GOOD" If Last Calc Was Zero TRUE
17:REM		Fault Force Thread 1 to fault and halt program here.
18:T1F		Force Load and Run in Thread 1: Program = "Fault"
19:REM		Wait Forever This allows somebody to use Real-Time Trace to detect where the program is at
20:JMP		Jump to "0"
21:REM	PUSH GOOD	Push Good
22:CLX		Last Reg Mark[49] = Reg Mark[48]
23:JMP		Jump to "LOOP"
24:REM		

Fault

Fault stops the servo and enters an infinite loop. The infinite loop keeps the program running which is nice when using QuickControl's Real-Time Trace feature to identify what the program is doing.

Line# Oper	Label	Command
1:REM		Fault Stop Motion End Program
2:VMP		Velocity Mode: acc=40000 cps/s, vel=0 cps
3:WBS		Wait On Bit State Until "Trajectory Active" is LOW/FALSE
4:DMT		Disable Multi-Tasking
5:REM		Wait Forever This allows somebody to use Real-Time Trace to detect where the program is at
6:JMP		Jump to "0"

Rotary Knife - Simple SN.qcp

The following is an overview of "Rotary Knife-Simple SN.qcp". This is very similar to "Rotary Knife-Simple-Dynamic.qcp" except it is written for the SilverNugget. Instead of the EGM command, the SilverNugget uses the RSD command. The SilverNugget does not support the extended calculation commands CLD/CLX so the calculations usually require a little more code.
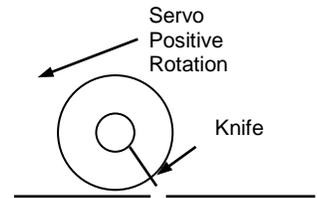
Servo
Positive
Rotation

Knife

This program file is composed of the following programs:

Init
Same as "Rotary Knife-Simple-Dynamic.qcp" except as noted above.

Home to Index
Same as "Rotary Knife-Simple-Dynamic.qcp" except as noted above.

Rotary Knife
This program switches the Registered Step and Direction (RSD) command's Scale Factor (SF) between the Cut Segment Scale Factor (SFcs) and the Return Segment Scale Factor (SFrs).

Return Segment Scale Factor (SFrs): Scale Factor (SF) for RS.   The desired knife movement is Krs (the rest of the knife rotation).   While the knife goes the distance Krs, the master must move the rest of material pass the knife. The material distance Mcs has already passed, therefore the master just has to move Mrs=Mn-Mcs. SFrs is therefore,

SFrs = SF1 Krs/Mrs

Line# Oper	Label	Command
1:REM		Rotary Knife Operation
2:REM	LOOP	================ Main Loop ================
3:REM		Re-calculate Mrs in case Mn has changed Mrs=Mn-Mcs
4:CLC		Accumulator[10] = Mn[19]
5:CLC		Accumulator[10] = Accumulator[10] - Mcs[25]
6:CLC		Mrs[23] = Accumulator[10]
7:REM		Start Return Segment SF = SFrs = SF1 * Krs ----------------- Mrs
8:CLC		Accumulator[10] = SF1[29]
9:CLC		Accumulator[10] = Accumulator[10] * Krs[24]
10:CLC		Accumulator[10] = Accumulator[10]/Mrs[23]
11:CLC		SF[28] = Accumulator[10]
12:REM		Wait for Return Segment to pass
13:CLC		Accumulator[10] = Target Position[0]
14:CLC		Accumulator[10] = Accumulator[10] - Krs[24]
15:JMP		Jump to "-2" If Last Calc Was Negative TRUE
16:REM		Start Cut Segment
17:CLC		Accumulator[10] = SFcs[18]
18:CLC		SF[28] = Accumulator[10]
19:REM		Wait for Cut Segment to pass
20:CLC		Accumulator[10] = Target Position[0]
21:CLC		Accumulator[10] = Accumulator[10] - Kcpr[13]
22:JMP		Jump to "-2" If Last Calc Was Negative TRUE
23:REM		Modulo Knife
24:CLC		Targ - Kcpr[13], Pos - Kcpr[13]
25:REM		Loop forever
26:JMP		Jump to "LOOP"

Other Considerations

Programmable Limit Switch (PLS)

A PLS could be used to signal "End of Cut" to allow an outfeed stage to process the newly cut piece. The PLS would be added to the knife program and configured to trigger sometime after the Knife Cut Segment (Kcs). See Application Note QCI-AN050 Programmable Limit Switch for details.

Control Constants

In an Electronic Gearing application, the slave has to determine its target position from incoming master encoder counts. Because there is no pre-planed trajectory to follow, the servo loop does not have acceleration feedforward information. This is less of an issue as the line is typically brought up to speed slowly (low acceleration). The various Step and Direction routines estimate velocity feedforward from incoming master encoder counts.

CTC

CTC start with factory defaults and make the following change:

Ki=0 for increased stability. Increase Ki if your application requires the servo to firmly hold its position while stopped.

NOTE: Kv2 may need to be increased for larger inertial mismatches (10+). See Technical Document "QCI-TD054 Servo Tuning".

Missing Register Marks

"Rotary Knife – Registration.qcp" assumes the products can change length (Mn) often. It cannot, therefore, distinguish between a missing registration mark and an extra long product. If your application only has only small changes in Mn, a missing mark can be identified and Mn for the missing mark can be calculated using the past Mn's.