

Doc No.: AN-244

Version: 1.0

Date: 19 June 2009

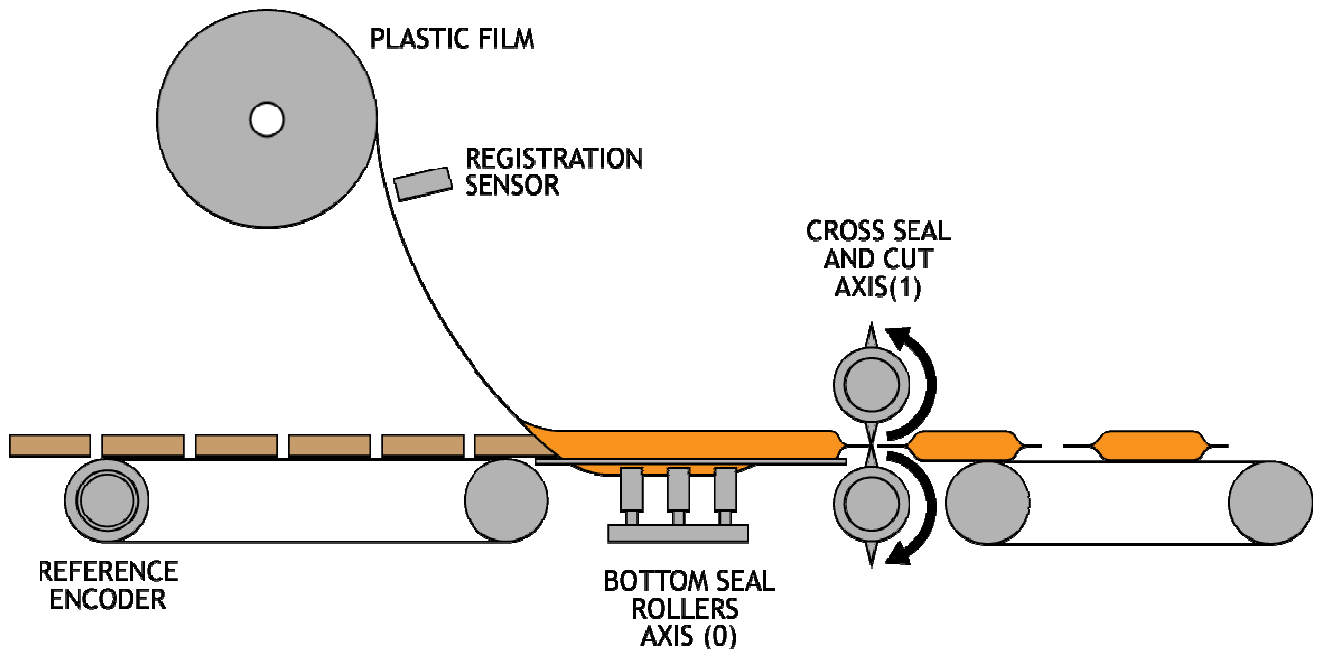
Subject: Horizontal Flow Wrapper – Trio BASIC example

APPLICATION NOTE

www.triomotion.com

1. Introduction

This application note discusses the program structures needed to make a horizontal flow wrapper packaging machine. The description and programs are provided as an example to show how this type of application might work. Trio does not guarantee that this solution will work on every flow wrapping machine or that any defined accuracy will be achieved. The mechanical layout is shown in the diagram below.



Horizontal Packaging Machine

In this example the bottom seal rollers and knife are driven by servo motors that are controlled by the Motion Coordinator. The main in-feed conveyor has a reference encoder which provides the information for the other 2 axes to follow. The MC206X gets a signal from a registration sensor to indicate the position where the printed marks on the material pass it. This is used to adjust the position of the rollers and knife so that the packets are cut at the correct place.

2. Initialisation

The STARTUP program sets up the servo axes, and some further settings for axis calibration are done in the ROLLERS program. In the example the WDOG is turned ON straight away, but this should be adapted to suit the machine safety arrangements required for a specific application.

The following user parameters must be set before running the programs:

- VR(10) is the start position for the rollers in mm. Set this to a suitable value so that the product feeds into the packet at the correct place when the machine is started.
- VR(11) is the pitch of the incoming product, that is spaced evenly on the feeder conveyor.
- VR(12) is the pack length. Set to the measured length of the packaging film material.

Two calibration parameters must also be set in the programs:

- Feeder UNITS must be set to counts per mm for the feeder conveyor. [ROLLERS]
- Roller UNITS must be set to counter per mm for the rollers. [ROLLERS]
- `expected` must be set to the offset value corresponding to the registration position sensor.
- `knife_ppr` must be set to the number of counts per revolution of the cutter. [CUTTER]
- `blades` must be set to the number of blades per revolution of the knife. [CUTTER]
- `d` must be set to the knife diameter in mm. [CUTTER]

Note that names are used for VRs instead of the VR number. This is to make the program easier to read.

```
start=1
knife_ready=2
start_pos=10
pitch=11
pack_length=12
reg_val=13
```

3. Startup

The startup program first checks to see if any axis errors exist and if there is an error, it runs the DATUM(0) procedure. This clears the error flags so that the axes can be restarted. RAPIDSTOP is used to make sure that all the motion buffers are empty before the program continues.

Two servo axes are set up. The correct position loop gains must be entered in this program. Loop gains shown in the example are for a specific Trio demonstration rig and will not work on a real machine. Note that the axis calibration figures for UNITS and REP_DIST are not the final numbers to be used in the system. The correct values are calculated and set in the ROLLERS and CUTTER programs.

The last part of the Startup program RUNs the two main programs, ROLLERS and CUTTER. When ROLLERS starts, it sets VR(start) to 0 and this is the signal for STARTUP to run the CUTTER program.

4. Homing

The first part the CUTTER program is a homing routine. (Subroutine `knife_home`) It must set the zero position of axis 1. For this example, the axis 1 motor is directly connected to the knife, so the Z mark of the encoder can be used. If there is a gearbox between the motor and the knife, then an external sensor must be fitted and DATUM(3) used to home the axis.

```
knife_home:
  BASE(knife)
```

```

CREEP=1000 ' set the homing speed
SPEED=100 ' set the speed for the MOVEABS
ACCEL=SPEED*10
DECEL=SPEED*100
DATUM(1) ' home on to the Z mark
WAIT IDLE
MOVEABS(0)
WAIT IDLE
RETURN

```

Notice the MOVEABS(0) which is there because DATUM(1) does not stop the motor exactly on the Z position. MOVEABS(0) assumes that the Z mark is in the park position of the knife. If the knife must park away from the Z mark, then use a sequence like this:

```

MOVEABS(park_offset)
WAIT IDLE
DEFPOS(0)
WAIT UNTIL OFFPOS=0

```

In many systems, the in-feed conveyor also needs to be homed and this is done in ROLLERS.

```

feeder_homing:
BASE(feeder)
REP_DIST=VR(pitch)
REP_OPTION=ON
REGIST(3)
WAIT UNTIL MARK
OFFPOS=-REG_POS
RETURN

```

The registration feature is used in this example to set the in-feed home position. If a conveyor with separators is used for the in-feed, then a sensor should be placed so that it detects one of the separator flights and this is then set as position 0. The homing routine requires that the in-feed conveyor is started as this is powered from an external motor. Notice that the Repeat Distance of the in-feed axis is set to the distance between separators.

5. Run cycle (ROLLERS)

VR(start) and VR(knife_ready) must be 1 before the ROLLERS program can continue.

The main run cycle consists of 3 parts.

1. The MOVELINK to accelerate the rollers and begin to follow the conveyor.
2. A continuous MOVELINK to make the rollers move at the same speed as the conveyor. During this phase, there is also a registration function to keep the wrapper material in phase with the printed marks on the wrapper.
3. When a stop is commanded, there is an “exit” MOVELINK to decelerate the rollers and stop.

Operation of these 3 parts is described here separately.

Acceleration motion:

```

' acceleration movelink to start the rollers:
MOVELINK(length/2,ld1,ad1,0,feeder,2,VR(start_pos))

```

The acceleration is done over half the product length. It has a delayed start and is set to start at the position of the master (feeder conveyor) in VR(start_pos).

Main motion and registration mark compensation:

A registration sensor is placed to pick up the printed marks on the wrapper material. REGIST(3) arms the registration and when the mark is detected, a correction move is then calculated and run on a virtual axis. The virtual axis is linked to the rollers axis by ADDAX which means that it makes a small adjusting move on the rollers “on the fly” while the main movelink is also running.

The adjust value `a_d` in the routine is calculated by recording the position where the printed mark was detected and comparing it with a pre-set expected position. 40% of that position error is then applied. The registration works like a feedback loop with the error from the feed being used to correct the length of the next movelink. Like any feedback loop, if the gain is too high then the control is unstable. Experience has shown that 40% gain is a good value for these types of systems. Other values can be tried to see if they produce a more or less stable registration position.

```

REPEAT
  REGIST(3)
  WAIT UNTIL MARK OR (VR(start)=0)
  IF MARK THEN
    mark_error=REG_POS-expected
    a_d=0.4*mark_error ' adjust distance
    IF a_d>max_adjust THEN a_d=max_adjust
    IF a_d<min_adjust THEN a_d=min_adjust
    MOVELINK(a_d,a_d*2,a_d,a_d,feeder) AXIS(virt1)
    VR(reg_val)=-a_d ' value for cutter to use
    OFFPOS=-a_d ' adjust the absolute value of rollers
    PRINT #5, VR(13)
    WAIT UNTIL (VR(reg_val)=0) OR (VR(start)=0)
  ENDIF
UNTIL VR(start)=0

```

Motion stop:

VR(start)=0 is the signal to stop the rollers. It also signals the knife to stop and park, so the rollers continue for another cycle until the knife program signals that the knife is parked. It does this by setting VR(knife_ready) to 0.

```

WAIT UNTIL VR(knife_ready)=0
REP_OPTION=3 ' force continuous movelink to stop (at end next length)
' deceleration MOVELINK
MOVELINK(length/2,ld1,0,ddl,feeder)

```

Setting REP_OPTION=3 causes the continuous (mode 4) MOVELINK to stop at the end of its next cycle. The motion is then taken over by the deceleration movelink which decelerates over half a product length.

6. Run cycle (CUTTER)

Before the cutter operation starts, there is a subroutine to calculate the knife parameters. The knife is calibrated in encoder edges, so it is necessary to enter the number of edges (or counts) per revolution of the knife. This is the number of counts to make the knife go 360 degrees and so must take into account any gearbox that is fitted.

```

' User set values
knife_ppr=4000 ' counts per rev for knife axis
blades=2 ' number of blades fitted

```

Also enter the number knife blades in one 360 degree turn of the knife axis.

Enter the knife diameter 'd' and the number of degrees of knife axis movement to allow for accel and decel of the knife. The routine now calculates the circumference, start position for the length adjustment and the MOVELINK accdist and decdist in feeder units.

Once the knife calculations are completed, the main routine starts. It waits for VR(start) to be 1, then starts a continuous MOVELINK (mode 4) to run the knife at the same speed as the rollers, then runs this loop:

```

REPEAT
  ' Product length in mm is in VR(pack_length)
  ' make pitch adjustment on virtual axis
  adjust=(VR(pack_length)-knife_c+VR(reg_val))*knife_ppr/c
  VR(reg_val)=0 ' signal to rollers to say "value taken"
  ald=c*120/360
  PRINT #5,"adjust ";adjust
  PRINT #5,"ald ";ald
  VR(knife_ready)=1
  WAIT UNTIL MPOS<adjust_start ' wait for start of cycle
  WAIT UNTIL MPOS>adjust_start ' wait for knife clear
  MOVELINK(adjust,ald,ald/4,ald/4,roller) AXIS(virt2)
  WA(2)
  WAIT UNTIL (MTYPE AXIS(virt2)=0) OR (VR(1)=0)
UNTIL VR(1)=0

```

The purpose of this loop is to put in an additional MOVELINK on a virtual axis, ADDAXed to the knife axis, which adds or subtracts a distance equal to the difference between the knife blade pitch and the product pack length. The adjustment distance is in the variable adjust and is calculated in the program like this:

$$\text{adjust} = (\text{VR}(\text{pack_length}) - \text{knife_c} + \text{VR}(\text{reg_val})) * \text{knife_ppr} / \text{c}$$

knife_ppr/c is to convert the result into encoder edges (or counts). The other parts of the expression are therefore in user units. (ie mm)

$$\text{adjust} = \text{VR}(\text{pack_length}) - \text{knife_c} + \text{VR}(\text{reg_val})$$

pack_length - knife_c gives the required difference between knife blades and product length. The addition of VR(reg_val) is so that when the rollers axis makes a registration adjustment, the cutters make the reverse adjustment so that they stay in the same position with the feeder axis.