Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292333
Fax: +44 (0)1684 297929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Tel: +1 412.968.9744
Fax: +1 412.968.9746

Tomson Centre
118 Zhang Yang Rd., B1701
Pudong New Area, Shanghai,
Postal code: 200122
P. R. CHINA
Tel/Fax: +86-21-58797659

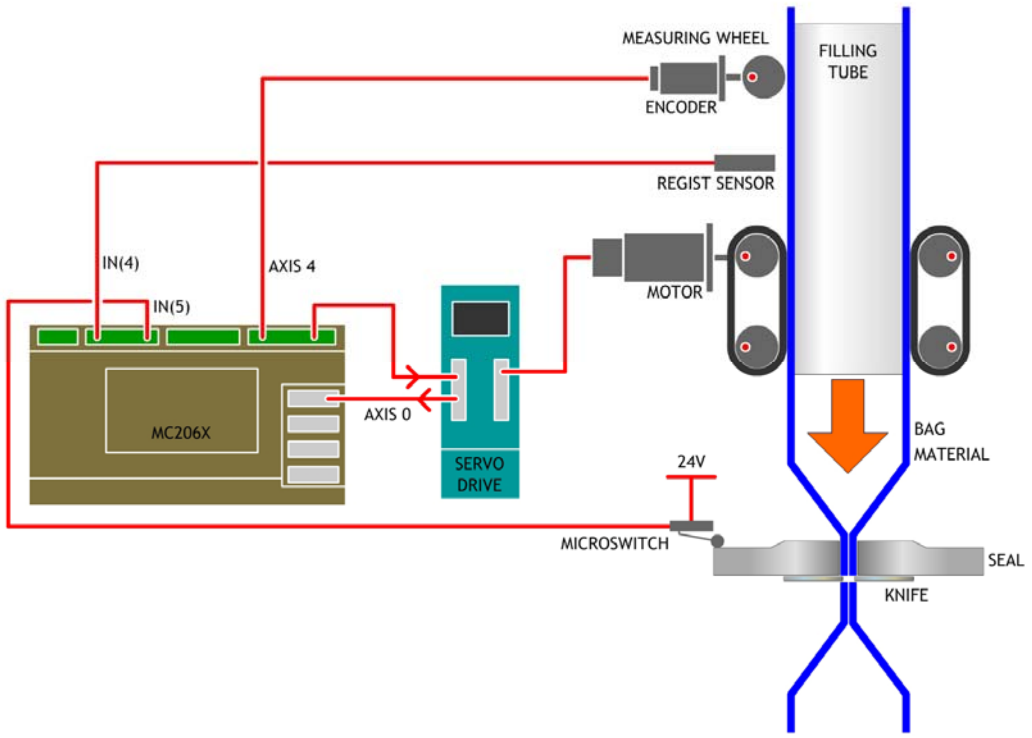
Doc No.: AN-221
Version: 1.0
Date: 05 November 2008
Subject: Vertical Bag Machine – Trio BASIC example

APPLICATION NOTE

www.triomotion.com

1. Introduction

This application note discusses the program structures needed to make a vertical bag machine. (form-fill-seal machine) There are various mechanical forms to the actuators on this type of machine. This example is for a system where the seal/cut jaws are stationary and provide a signal to show when the jaws are open. The mechanism that pulls the plastic tube down is driven by a servo motor that is controlled by the Motion Coordinator. It is possible that this mechanism can slip, so an external encoder is provided that more accurately measures the length of tube being pulled down. The mechanical layout and the connections to an MC206X are shown as an example.



Vertical Form-Fill-Seal layout

In this example the seal jaws and knife are driven by a motor that is external to the Motion Coordinator based servo system. The MC206X gets a signal to indicate that the jaws are open and that the bag material can be pulled down ready to make the next bag.

2. Initialisation

It is assumed that the servo axis is already set up, the WDOG is ON and that the following user parameters have been set:

- VR(10) is set to the bag length in mm.
- VR(11) is set with the position of the registration mark sensor. mm measured from the cut edge.
- VR(12) is the number of missed marks allowed before an alarm is sounded and the machine is stopped.
- VR(13) is the initial slip factor. This should be set to the approximate slip factor so that the slip compensation starts with a value near to the true slip. E.g. 1.05 is 5% slip.

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

```
bag_length = 10
mark_offset = 11
mark_alarm = 12
init_slip = 13
average_length = 20
slip_factor = 21
adjust = 22
draw_length =23
missed_marks = 24
```

In a similar way, names are given to the Inputs and Outputs.

```
' input definitions
jaws_open = 5

' output definitions
status = 8
led = 9
```

Some internal parameters are initialised at the start of the program:

```
' initialise slip factor and registration adjustment
VR(average_length)=VR(bag_length) * VR(init_slip)
VR(slip_factor)=0
VR(adjust)=0

OP(status,ON)
```

3. Homing

The first part of the program is a homing routine. It must set the zero position of axis 4. Axis 4 is the external measuring wheel encoder. The jaws must be open, so this is checked, then a move is started to pull down the plastic tube until it finds a printed mark. It then moves the material to align the mark with the programmed offset in VR(11) using MOVEMODIFY.

```
BASE(0)
' find first mark
WAIT UNTIL IN(jaws_open)=ON
DEFPOS(0)
DEFPOS(0) AXIS(4)
REGIST(3) AXIS(4)
MOVE(VR(bag_length)*1.5)
WAIT UNTIL MTYPE<>0 'make sure that move has started
WAIT UNTIL MARK AXIS(4) OR (MTYPE=0)
IF MARK AXIS(4) THEN
  MOVEMODIFY(REG_POS AXIS(4)+(VR(bag_length)-VR(mark_offset)))
ELSE
  GOTO alarm_routine
ENDIF
WAIT IDLE
WA(100)
DEFPOS(0) AXIS(4) ' bags in position and ready to start
```

Notice that if the move completes without the MARK being detected, the program goes to an alarm routine to show that there is a problem. If the MARK is seen then at the end of this routine, the bag is in position ready for the first cut.

4. Run cycle

The main run cycle consists of 3 parts.

1. The move to draw down the tube after the jaws open signal has been detected.
2. The registration mark detection and correction.
3. The slip detection and correction.

Operation of these 3 parts is described here separately.

4.1. *Main motion:*

```
REPEAT
  ' wait for rising edge of jaws open signal
  WAIT UNTIL IN(jaws_open)=OFF
  WAIT UNTIL IN(jaws_open)=ON
  . . .
  MOVE(VR(bag_length)+VR(slip_factor)+VR(adjust))
  WAIT IDLE
  . . .
  DEFPOS(0) AXIS(4) 'set the stopped position as 0
UNTIL FALSE
```

It can be seen that the cycle is very simple when registration and slip compensation are removed. The measuring encoder axis is zeroed after each move so that it is ready to measure the next bag length.

4.2. *Registration mark compensation:*

VR(adjust) in the above 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 move. 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
  ' wait for rising edge of jaws open signal
  WAIT UNTIL IN(jaws_open)=OFF
  WAIT UNTIL IN(jaws_open)=ON
  REGIST(3) AXIS(4)
  MOVE(VR(bag_length)+VR(slip_factor)+VR(adjust))
  WAIT IDLE
  . . .
  IF MARK AXIS(4) THEN
    VR(missed_marks)=0
    mark_pos=REG_POS AXIS(4)
    mark_error=mark_pos-VR(mark_offset)
    VR(adjust)=mark_error*0.4
  ELSE
    VR(missed_marks)=VR(missed_marks)+1
    IF VR(missed_marks)>VR(mark_alarm) THEN
      GOTO alarm_routine
    ENDIF
  ENDIF
  DEFPOS(0) AXIS(4) 'set the stopped position as 0
UNTIL FALSE
```

If the move is finished and the MARK did not go true, then a counter is incremented. When the count in VR(missed_marks) is greater than the limit set in VR(mark_alarm), the program goes to alarm_routine. Here the machine will be stopped and an alarm condition is displayed. See Alarm Routine below.

4.3. Slip correction:

Slip is measured by comparing the length measured by an accurate measuring wheel against the programmed bag length. As the slip can vary from one move to the next, it is best to filter this value by calculating a "rolling average". The following expression is a common rolling average filter:

```
filter_out = (1-c) * filter_out + c * filter_in
```

c is the filter time constant factor and must have a value between 0 and 1. The higher numbers give a faster response.

The amount of average slip is then added to the total amount to be drawn on the next cycle. Here is the complete calculation:

```
VR(draw_length)=MPOS AXIS(4)
VR(draw_length)=VR(draw_length)-VR(slip_factor)-VR(adjust)
VR(average_length)=(1-c)*VR(average_length) + c*VR(draw_length)
VR(slip_factor)=VR(bag_length)-VR(average_length)
```

At the end of each move, MPOS of the encoder axis has the total measured length that was drawn down. This must be adjusted by removing the extra lengths put in by the previous slip compensation and any registration adjust value used. After the average measured length is calculated, the slip factor for the next move is produced by subtracting the average slip from the programmed bag length.

The full motion cycle is therefore made up of these 3 parts. Together, they look like this:

```
REPEAT
  ' wait for rising edge of jaws open signal
  WAIT UNTIL IN(jaws_open)=OFF
  WAIT UNTIL IN(jaws_open)=ON
  REGIST(3) AXIS(4)
  MOVE(VR(bag_length)+VR(slip_factor)+VR(adjust))
  WAIT IDLE

  ' calculate measured draw length and calculate average
  VR(draw_length)=MPOS AXIS(4)
  VR(draw_length)=VR(draw_length)-VR(slip_factor)-VR(adjust)
  VR(average_length)=(1-c)*VR(average_length) + c*VR(draw_length)
  VR(slip_factor)=VR(bag_length)-VR(average_length)

  IF MARK AXIS(4) THEN
    VR(missed_marks)=0
    mark_pos=REG_POS AXIS(4)
    PRINT #5,mark_pos[2]
    mark_error=mark_pos-VR(mark_offset)
    VR(adjust)=mark_error*0.4
  ELSE
    VR(missed_marks)=VR(missed_marks)+1
    IF VR(missed_marks)>VR(mark_alarm) THEN
      GOTO alarm_routine
    ENDIF
  ENDIF

  DEFPOS(0) AXIS(4) 'set the stopped position as 0
UNTIL FALSE
```

5. Alarm routine

When the number of missed marks exceeds the set value in VR(mark_alarm), this routine is called.

```
alarm_routine:
  OP(status,OFF)
  ' flash a led
  REPEAT
    OP(led,ON)
    WA(250)
    OP(led,OFF)
    WA(250)
  UNTIL FALSE
STOP
```

The routine here is very simple and is shown only as a guide. Each system will have its own requirements for this routine, it probably would put a message on the operator's display and trigger a highly visible indicator lamp. Rather than just stopping the program, this routine should wait for the operator to reset the system and then go back to the start of the program so that the machine can be re-homed.