www.triomotion.com

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
Ph: +1 412.968.9744
Fx: +1 412.968.9746

Doc No.: TN20-101
Version: 2.1
Date: 1st Nov 2007
Subject: Corner Speed Control

Application Note

1. Corner Speed Control:

In many X-Y plotting systems, there is a need to adjust the speed of movement to allow for changes in direction around corners. Functions in the Motion Coordinator are now available to assist with programming such applications:

The features allow for:

- Allowing merged movement to decelerate to a fraction of the programmed speed depending on changes in direction.
- Allow for interaction with a knife raising subroutine when angle changes exceed a preset value.
- Slowing down around arcs automatically depending on the radius of the arc.
- Tangential rotary axes for knives, saws and "pizza wheels"

See also the application notes on Tangential Axis Control TN20_92 and Lookahead Move Buffer TN20_55

2. System requirements:

Tangential control requires the use of a "Look-Ahead" version of Motion Coordinator system software. The feature is supported in version 1.66 Dev 32 or later. This version can be built for Euro205x, MC206X, PCI208 and MC224 only. It is not available for older generation products due to memory size limitations.

3. Command reference:

DECEL_ANGLE

Type: Axis Parameter

Description: Used to define the angle change in radians above which an X-Y system should start to decelerate. Operation of DECEL_ANGLE applies to MOVESP/MOVEABSSP/MOVECIRCSP and depends on setting of STOP_ANGLE and CORNER_MODE.

Example1:
```
' Decelerate above 15 deg (0.2618 rad)

DECEL_ANGLE = 0.2618
```

STOP_ANGLE

Type: Axis Parameter

Description: Used to define the angle change in radians above which an X-Y system should decelerate to zero speed. Operation of STOP_ANGLE applies to MOVESP/MOVEABSSP/MOVECIRCSP and depends on setting of DECEL_ANGLE and CORNER_MODE. If the X-Y system turns through an angle between STOP_ANGLE and DECEL_ANGLE the system will slow down to an intermediate speed.

Example1:
```
' Stop above 25 deg (0.4363 rad)

STOP_ANGLE = 0.4363
```

RAISE_ANGLE

Type: Axis Parameter

Description: Used to define the angle change in radians above which an X-Y system will interact with a BASIC program to allow for a large change in direction on a tangential axis. RAISE_ANGLE does not control speed directly so should be greater than or equal to STOP_ANGLE. When a change in direction in a sequence of MOVESP/MOVEABSSP/MOVECIRCSP moves exceeds RAISE_ANGLE the system will set axis parameter CORNER_STATE according to the following sequence:

1 – System sets CORNER_STATE to 1 to indicate move ready to be loaded with large angle change.

2 – BASIC program raises knife.

3 – BASIC program sets CORNER_STATE to 3.

4 – System will load following move but with speed overridden to zero. This allows the direction to be obtained from TANG_DIRECTION.

5 – BASIC program orients knife possibly using MOVE_TANG.

6 – BASIC program clears CORNER_STATE to 0.

7 – System will ramp up speed to perform the next move.

Example1:

```
‘ Raise knife above 25 deg (0.4363 rad)

RAISE_ANGLE = 0.4363
```

CORNER_MODE

Type: Axis Parameter

Description: Allows the program to control the cornering action:

Bit 0 – Setting bit 0 allows the manual setting of STARTMOVE_SPEED

Bit 1 – Setting bit 1 allows system to calculate STARTMOVE_SPEED using DECEL_ANGLE, STOP_ANGLE and the change in angle. Note that STARTMOVE_SP is a speed value assigned to individual MOVESP/MOVEABSSP/MOVECIRCSP moves.

Example1:

```
CORNER_MODE=2 ‘ Automatic speed control
```

CORNER_STATE

Type: Axis Parameter

Description: Allows a BASIC program to interact with the move loading process to facilitate knife rotation at sharp corners. See RAISE_ANGLE command.

Example:

```
MOVEABSSP(x,y)
IF CHANGE_DIR_LAST>RAISE_ANGLE THEN
    WAIT UNTIL CORNER_STATE>0
    GOSUB raise
    CORNER_STATE=3
    WA(10)
    WAIT UNTIL VP_SPEED AXIS(2)=0
    GOSUB lower
    CORNER_STATE=0
ENDIF
```

START_DIR_LAST

Type: Axis Parameter (Read Only)

Description: Allows the program to examine the start direction in radians of the last programmed MOVESP/MOVEABSSP/MOVECIRCSP move of 2 or more axes.

Example1:

```
>>MOVESP(10000,10000)
>>? START_DIR_LAST
0.7854

>>MOVESP(0,10000)
>>? START_DIR_LAST
0.0000
>>
```

END_DIR_LAST

Type: Axis Parameter (Read / Write)

Description: Allows the program to examine the end direction in radians of the last programmed MOVESP/MOVEABSSP/MOVECIRCSP move of 2 or more axes. END_DIR_LAST will be the same as START_DIR_LAST except in the case of circular moves.

END_DIR_LAST is used by the system to calculate the change in angle when a new move is loaded. END_DIR_LAST can be written to. This is often required when initialising a system or after a sequence of moves which are not MOVESP /MOVEABSSP /MOVECIRCSP moves of 2 or more axes.

Example1:
```
>>MOVESP(10000,-10000)
>>? END_DIR_LAST
2.3562
>>
```

CHANGE_DIR_LAST

Type:	Axis Parameter (Read)
Description:	Allows the program to examine the CHANGE in direction in radians of the last programmed MOVESP/MOVEABSSP/MOVECIRCSP move of 2 or more axes. CHANGE_DIR_LAST is always positive.

Example1:	`>>MOVESP(10000,-10000)`
	`>>? CHANGE_DIR_LAST`
	`1.4150`
	`>>`

TANG_DIRECTION

Type: Axis Parameter (Read only)

Description: Allows the program to examine the current move direction in radians of the executing MOVESP/MOVEABSSP/MOVECIRCSP move of 2 or more axes. In the case of circular moves it will change during the move.

Example1:
```
>>? TANG_DIRECTION
2.3562
>>
```

MOVETANG

Type:	Motion Command
Syntax:	MOVETANG(absolute_position<,link_axis>)
Description:	Moves the axis to the required position using the programmed SPEED, ACCEL and DECEL for the axis. The direction of movement is determined by a calculation of the shortest path to the position assuming that the axis is rotating and that REP_DIST has been set to ∏ radians (180 degrees) and that REP_OPTION=0. **IMPORTANT:** *The REP_DIST value will depend on the UNITS value and the number of steps representing PI radians. For example if the rotary axis has 4000 pulses/turn and UNITS=1 the REP_DIST value would be 2000.* If a MOVETANG command is running and another MOVETANG is executed for the same axis, the original command will not stop, but the endpoint will become the new absolute position.
Parameters:	*absolute_position*: The absolute position to be set as the endpoint of the move. Value must be within the range –PI to +PI in the units of the rotary axis. For example if the rotary axis has 4000 pulses/turn, the UNITS value=1 and the angle required is PI/2 (90 deg) the position value would be 1000 *link_axis*: An optional link axis may be specified. When a link_axis is specified the system software calculates the absolute position required each servo cycle based on the link axis TANG_DIRECTION. The TANG_DIRECTION is multiplied by the REP_DIST/PI to calculate the required position. Note that when using a *link_axis* the *absolute_position* parameter becomes unused. The position is copied every servo cycle until the MOVETANG is CANCELled.
Example:	An X-Y positioning system has a stylus which must be turned so that it is facing in the same direction as it is travelling at all times. The XY axis pair are axes 4 and 5. The tangential stylus axis is 2: MOVETANG(0,4) AXIS(2)

TANG_DIRECTION

Type:	Axis Parameter

Description: Allows the program to examine the current move direction in radians of the executing MOVESP/MOVEABSSP/MOVECIRCSP move of 2 or more axes. In the case of circular moves it will change during the move.

When used with a 2 axis X-Y system, this parameter returns the angle in radians that represents the vector direction of the interpolated axes. The value returned is between –PI and +PI and is determined by the directions of the interpolated axes as follows:

X	Y	value	
0	1	0	
1	0	PI/2	
0	-1	PI	(+PI or –PI)
-1	0	-PI/2	

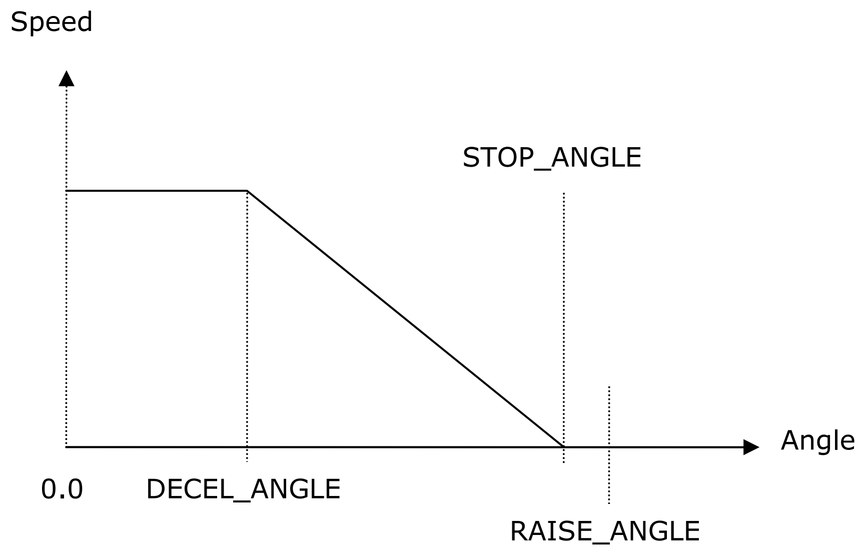
Example1:

```
‘ Note scale_factor_x MUST be the same as scale_factor_y
UNITS AXIS(4)=scale_factor_x
UNITS AXIS(5)=scale_factor_y

BASE(4,5)
MOVE(100,50)
angle = TANG_DIRECTION
```

Example2:

```
>>? TANG_DIRECTION
2.3562
>>
```



4. Example:

An X-Y cutting table has a “pizza wheel” cutter which must be steered so that it is always aligned with the direction of travel. The main X and Y axes are controlled by Motion Coordinator axes 0 and 1, and the pizza wheel is turned by axis 2.

Control of the Pizza Wheel is done in a separate program from the main X-Y motion program. In this example the steering program also does the axis initialisation.

Program TC_SETUP.BAS:

```
' Set up 3 axes for Tangential Control

WDOG=OFF

BASE(0)
P_GAIN=0.9
VFF_GAIN=12.85
UNITS=50 ' set units for mm
SERVO=ON

BASE(1)
P_GAIN=0.9
```

```
VFF_GAIN=12.30
UNITS=50 ‘ units must be the same for both axes
SERVO=ON

BASE(2)
UNITS=1 ' make units 1 for the setting of rep_dist
REP_DIST=2000 ‘ encoder has 4000 edges per rev.
REP_OPTION=0
UNITS=4000/(2*PI) ‘ set units for Radians
SERVO=ON

WDOG=ON

' Home the 3rd axis to its Z mark
DATUM(1) AXIS(2)
WAIT IDLE
WA(10)

‘ start the tangential control routine
BASE(0,1) ‘ define the pair of axes which are for X and Y
old_angle=TANG_DIRECTION ‘ store the first position
REPEAT
  angle=TANG_DIRECTION
  IF angle<>old_angle THEN
    MOVETANG(angle) AXIS(2)
    old_angle=angle
  ENDIF
UNTIL FALSE
```

Program MOTION.BAS:

```
‘ program to cut a square shape with rounded corners
MERGE=ON
SPEED=300

nobuf=FALSE ‘ when true, the moves are not buffered
size=120 ‘ size of each side of the square
c=30 ‘ size (radius) of quarter circles on each corner

DEFPOS(0,0)
WAIT UNTIL OFFPOS=0
WA(10)

MOVEABS(10,10+c)
REPEAT
  MOVE(0,size)
  MOVECIRC(c,c,c,0,1)
  IF nobuf THEN WAIT IDLE:WA(2)
  MOVE(size,0)
  MOVECIRC(c,-c,0,-c,1)
  IF nobuf THEN WAIT IDLE:WA(2)
  MOVE(0,-size)
  MOVECIRC(-c,-c,-c,0,1)
  IF nobuf THEN WAIT IDLE:WA(2)
  MOVE(-size,0)
  MOVECIRC(-c,c,0,c,1)
  IF nobuf THEN WAIT IDLE:WA(2)
UNTIL FALSE
```