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.: **TN20-92**
Version: **2.1**
Date: **26th June 2008**
Subject: **Tangential Axis Control**

Application Note

1. Tangential control:

In many X-Y plotting systems, there is a need to keep the stylus oriented in the direction of travel. Each combination of speeds in X and Y results in a Vector value which needs to be fed to a third axis controlling the rotation of the stylus. Examples of machines that require such a control are; glass cutting, knife blades, powered jig-saws and "pizza wheels".

This document describes a new motion command and axis parameter which when used together will enable the easy implementation of Tangential Axis Control.

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:

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 π 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. A tangential control routine is run in a separate process. The XY axis pair are axes 4 and 5. The tangential stylus axis is 2:

```
BASE(0,1)
WHILE TRUE
  angle=TANG_DIRECTION
  MOVETANG(angle) AXIS(2)
  WA(1)
WEND
```

Or:

Using the *link_axis* parameter load a MOVETANG which remains running to automatically update the stylus position. The XY axis pair are axes 4 and 5. The tangential stylus axis is 2:

```
MOVETANG(0,4) AXIS(2)
```

Type: Axis Parameter

Description: 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:

```
BASE(0,1)
angle_deg = 180 * TANG_DIRECTION / PI
```

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
REPEAT
  MOVETANG(TANG_DIRECTION) AXIS(2)
  WA(1)
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
```