Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Tel: +1 412 968 9744
Fax: +1 412 968 9746

B1602 Tomson Centre
188 Zhang Yang Rd.,
Pudong New Area, Shanghai,
Postal code: 200122
CHINA
Tel/Fax: +86 21 587 97659

Doc No.: **AN-252 (Preliminary)**

Version: **0.2**

Date: **20 April 2010**

Subject: **3D Direction Calculations and MSPHERICALSP**

APPLICATION NOTE

1. Introduction

Motion Coordinators calculate 2D directions for the input and output directions of multi-axis movements. A new software enhancement allows the 3 dimensional directions of 3 axis moves to be calculated automatically. The software will also calculate the change in direction between movements which may be used to automatically slow down using the corner speed control features. A new move command MSPHERICALSP is provided to make use of these features.

The features described are implemented from version 1.6697 LA version only.

1.1. 3D Direction Vector representation

2D directions are easily defined using number +/-PI. For example, if you define a movement and read back its start and end directions:

```
>>MOVE(1000,-1000,0)

>>? END_DIR_LAST AXIS(0)

2.3562
```

The END_DIR_LAST of the 1000,1000 move is 3*PI/4 or 2.3562. The third axis movement has no effect.

3D directions are defined in a completely different way because the 3D direction cannot be described with a single number. The 3D direction is a vector (dx,dy,dz) where dx, dy, dz are numbers between plus and minus 1.0.

The Motion Coordinator must firstly be told to calculate in 3D as 2D directions from 3 axis moves remain valid. This is done by setting bit 2 of AXIS_MODE on the base axis of the group.

```
>>AXIS_MODE=4

>>MOVE(1000,-1000,0)

>>? END_DIR_LAST AXIS(0),END_DIR_LAST AXIS(1),END_DIR_LAST AXIS(2)

0.7071     -0.7071     0.0000

>>
```

This simple example doesn't use the 3rd dimension to make it easy to understand. A 3D simple example would be:

```
>>MOVE(1000,-1000,1000)
>>? END_DIR_LAST AXIS(0),END_DIR_LAST AXIS(1),END_DIR_LAST AXIS(2)
0.5774     -0.5774     0.5774
>>
```

Note that changes in 3D directions can be represented as a single number +/-PI and for ease of use this is done rather than representing the change as a 3D vector:

```
>>MOVE(0,1000,0)
>>? END_DIR_LAST AXIS(0),END_DIR_LAST AXIS(1),END_DIR_LAST AXIS(2)
0.0000     1.0000     0.0000
>>MOVE(1000,-1000,0)
>>? CHANGE_DIR_LAST
2.3562
>>
```

This CHANGE_DIR_LAST quantity can be used to automatically slow down using the same methods as 2D move changes using DECEL_ANGLE, STOP_ANGLE etc.

1.2. 3D MSPHERICALSP

To allow speeds to be embedded within a 3D arc the software supports a new move type MSPHERICALSP. This has the same parameters as MSPHERICAL but has FORCE_SPEED and ENDMOVE_SPEED embedded with the move parameters into the move buffer. Using the CORNER_MODE bits you can program the system to automatically slow down due to junction angle changes in the same way as using 2D moves. Use of FULL_SP_RADIUS is also supported.