Using the A3200 ROTATION Command for Five-Axis Systems

The ROTATION command in the Automation 3200 (A3200) implements a rotation matrix operation. The function can be used to transform coordinates on mechanical actuators that include rotary motion. Multiple ROTATION commands can be configured to work with systems that have more than one rotary axis. A five-axis system consisting of two rotary axes and three linear axes is an example of such a system (Figure 1). This application note will discuss the process of setting-up the ROTATION command for this type of application.

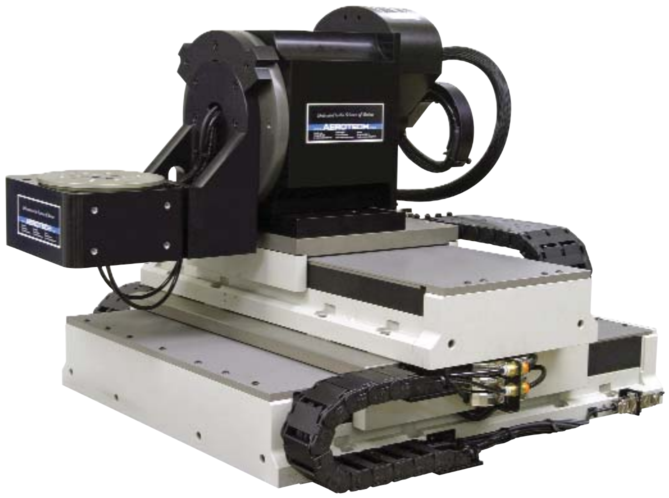

Figure 1. An example five-axis assembly (Z axis not shown).

Theory of Operation

Five-axis systems are capable of accessing any point on a hemisphere with a tool normal to the surface at the given point. Most 3D CAD tools can output tool paths that include the location on the surface in x/y/z coordinates along with the normal to the surface expressed as two angles (A and B in Figure 2).

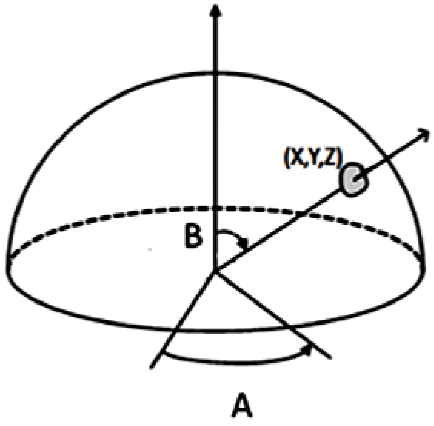

Figure 2. Five-axis systems can access any point on a hemisphere with a tool normal to the surface expressed as two angular positions.

Aerotech's ROTATION function can take these part x/y/z positions and rotation angles and transform this information in real time into position commands for the servo axes. Since the part positions are different than the servo system positions, two sets of axes must be used – one to define the part positions and the other the servo axes' positions. The part-axis system consists of three virtual axes and for this example we have named these axes x/y/z. The physical axes will be mapped to linear servo axes X/Y/Z and rotary axes A/B.

Setting-Up the ROTATION Command

The ROTATION command will transform data from one three-axis Cartesian system (InputAxes) into another three-axis system (OutputAxes). A separate rotation command is required for each rotary axis in the system. Up to 10 ROTATION commands can be active at one time. This example will require two ROTATION commands, one for each rotary axis.

The rotation command is shown below:

```
ROTATION 1 InputAxes x y z OutputAxes X Y Z
```

The order of the input and output axes is very important. The first two axes define a plane and the third axis is the normal to the plane. The third axis is also the axis about which the rotation occurs. The order of the axes implies positive move directions for each of the axes, the positive direction of rotation and the relationship between input axes and output axes when no rotation is active. Each of these items will be examined separately.

Right-Hand Rule

The orientation of the axes is defined by the right-hand rule, where the first axis of the three points along the index finger, the second axis points along the middle finger and the third axis points along the thumb (Figure 3A).

The rotation for this system occurs about Axis 3. The positive direction of rotation of the axis is also defined by the right hand rule where the thumb of the right hand points along the positive direction of Axis 3 and the curled fingers of the right hand indicate the positive angle of rotation (Figure 3B).

These relationships must be observed when defining the positive move directions for both the virtual and physical axes. The mapping from input to output axes starts with the virtual coordinate system and works back to the physical. At each step of the process the positive move directions of the mapped axes must be consistent with each other AND the right-hand rule.

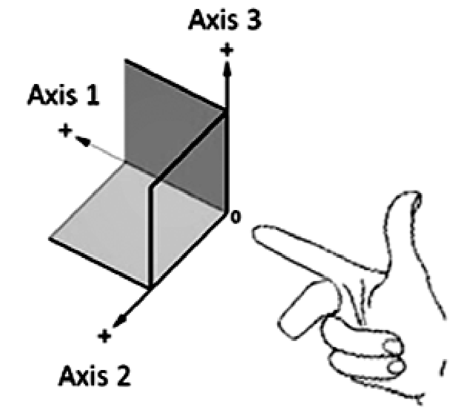

Figure 3A. Right-hand rule coordinate system.

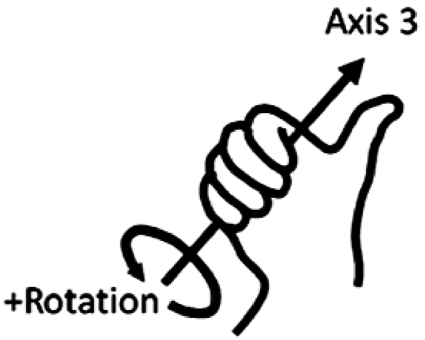

Figure 3B. Right-hand rule direction of positive rotation.

The orientation shown in Figure 4 is one of four that are possible. The other three are attained by rotating the x/y plane at intervals of 90 degrees around the z axis. The names of the axes in either system can be changed to meet application requirements. ***VERY IMPORTANT:*** *The definition of positive motion of the axes is fixed based on the directions that the arrows are pointing in the illustration. Once the coordinate frames have been set-up the positive move directions of the axes or part program must be changed to be consistent with the directions indicated by the coordinate frames.*

Per the mapping we have defined in Figure 4, with no rotations active, motion in the +x direction would result in motion in the +X, motion in +y will result in +Y and +z will result in +Z.

The configuration of the ROTATION command starts from the part coordinate frame and works back towards the servo axis coordinate frame. A new ROTATION command is required for each rotary axis in the system. The rotary axis closest to the part coordinate frame is used as the input for the first ROTATION command. The next rotary axis is associated with the next ROTATION command and so on until all rotary axes in the system have been defined.

For the system shown in Figure 4, the first rotary axis is the A axis which rotates the x/y plane about the z axis. Positive rotation motion will be counter-clockwise per the right hand rule. The rotation command and rotary axes are specified as follows:

```
ROTATION 1 InputAxes x y z OutputAxes X Y Z
ROTATION 1 Angle A 1, 0
```

The ROTATION ANGLE command has arguments for inverting the move direction and for offsetting the angle. To invert the positive move direction the value after the axis name would change from 1 to -1. The final argument is used to specify an angular offset. This offset can be used to compensate for variations in angular position that may occur during the attachment of the part.

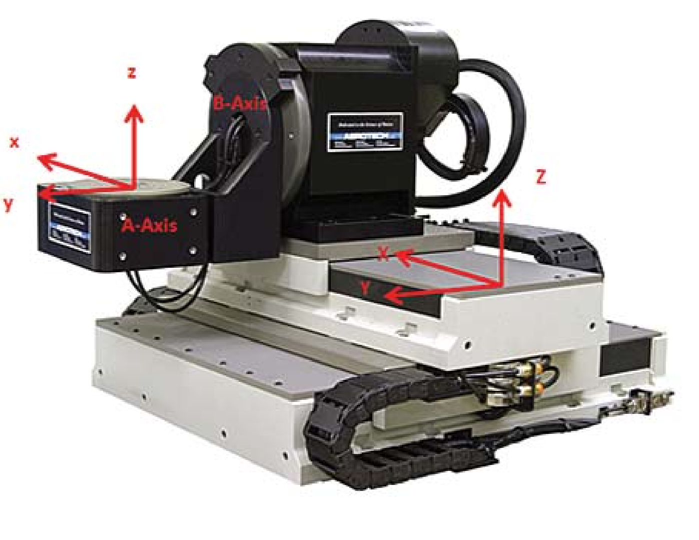

Figure 4. An example of a five-axis system showing axis orientations.

The next rotary axis is the B axis and it rotates in the z/x plane about the y axis of our input plane (per the right hand rule). The input to the second rotation command is as follows:

```
ROTATION 2 InputAxes z x y OutputAxes Z X Y
ROTATION 2 Angle B 1, 0
```

We rotate the z/x plane about the y axis on the input. Starting at 0 angle, input motion in the z axis causes motion in the Z, motion in the x axis causes motion in the X and motion in the y axis causes motion in the Y.

Setting-Up Offsets

In most applications the center of rotation of the rotary axes are offset from each other. Additionally, the origin of the part coordinate system might not be located directly on the center of axis of rotation as shown in Figure 5. The ROTATION Displacement command is used to define the offsets between the axis of rotations and the part coordinate systems. The displacement command has offset values for all three axes. The offset is calculated based on the distance from the reference point to the rotation point. The sign of the displacement is based on the direction of movement from the reference point to the rotation point.

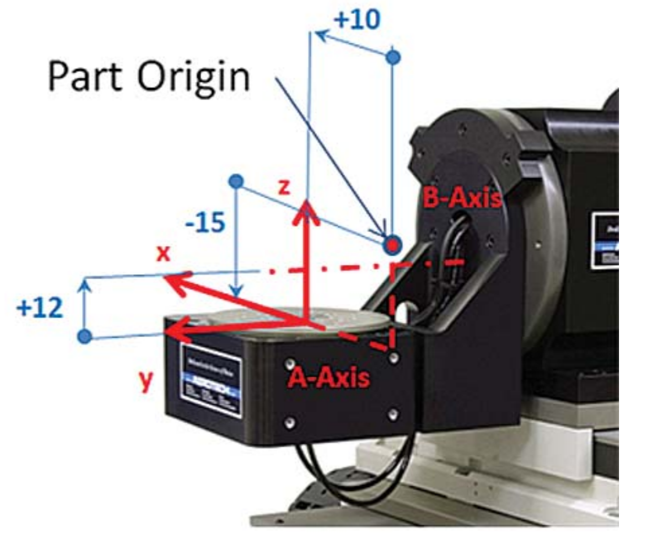

Figure 5. Setting-up offsets.

Using our example system, assume that the A and B axes of rotation intersect, the part origin is offset 15 mm above the tabletop of the A axis and offset by 10 mm in the X direction, and the B axis center of rotation is parallel to and in-line with the y axis and located 12 mm above the y axis. Based on this configuration the offsets for the rotation commands would be as shown in Figure 5.

From the illustration, the distance from the Part Origin to the center of rotation of the A axis along the x axis direction is +10. Along the z axis direction the distance from the Part Origin to the center of rotation is -15. The Part Origin is in line with the y axis so the offset in this direction is 0.

The offsets for the second rotation are calculated in a similar fashion. The starting point is now the origin of our x/y/z frame on the A axis and we calculate the offset to the origin of the point of rotation of the B axis. The coordinate frame for the B axis must be placed on its axis of rotation. The origin of this coordinate frame can be anywhere along the axis of rotation as the angular position along the B axis does not change as a function of where the origin of rotation is located. For example, if a round pipe is rotated about its center the angular position of any point on the pipe does not change based on the point along the pipe where the rotation is applied (Figure 6).

We can therefore place the origin of the point of rotation of the B axis directly above the point of rotation of the A axis which gives an offset of "0" in the x and y directions. The z offset is +12 as we move in the +z direction from the location of our starting point on the surface of the A axis up to the B axis of rotation. The offsets are placed in the order that the axes are defined in the ROTATION INPUT command. For the first rotation the order is x-y-z and for the second rotation the order is z-x-y:

```
ROTATION 1 Displacement 10, 0, -15
ROTATION 2 Displacement 12, 0, 0
```

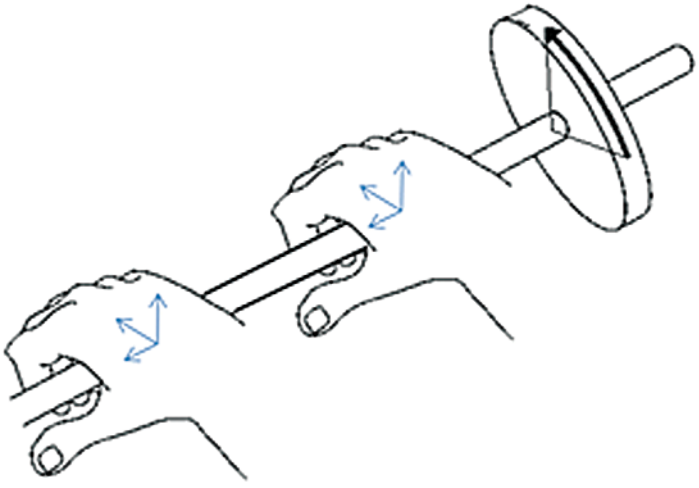

Figure 6.

The rotation commands are enabled/disabled with the following commands:

```
ROTATION 1 ON/OFF
ROTATION 2 ON/OFF
```

Issues with Constant Velocity/Surface Speed

When contouring is active on the A3200 (Velocity ON), the control will ramp-up the vector velocity based on the task acceleration parameters until it reaches the programmed vector velocity given by the "F" word. This velocity will be maintained throughout the program until a condition that forces a deceleration is encountered. Maintaining constant velocity on the part has significant implications for the motion of the servo axes as they may be required to change speed or direction instantaneously.

The "standard" look-ahead which monitors acceleration in an arc or detects a non-tangent move between two program lines will not directly address this problem as the accelerations are produced indirectly by the ROTATION command. There are two additional approaches that can be used to limit the acceleration of the physical axes.

The first method is to use a filter on the velocity command stream of the servo axes. The filter will have an effect similar to ramping, but it will also add a delay to the profile which will cause the path to deviate from the programmed path. The TrajectoryFIRFilter sets the length of an FIR (Finite Impulse Response) filter applied to the velocity command stream. Each "tap" in the filter equates to a 0.125 millisecond time interval. To add 50 milliseconds of filtering, the parameter would be set to 50/0.125 = 400.

The second method for reducing acceleration is to set the dependent axis acceleration limit. The A and B axes are considered dependent axes as their velocity command is derived from the virtual x/y/z axes. Short move times for the x/y/z axes coupled with longer moves on the A/B axes can result in very high speed for the A/B axes. The DependentCoordinatedAccelLimit parameter can be used to limit the acceleration and resulting speed of the dependent axes.

Example Program

Following is a simple example that shows all of the commands in sequence and includes setup for acceleration limiting on the dependent axes.

```
enable X Y Z A B x y z
Rotation 1 OFF
Rotation 2 OFF
Velocity ON
ramp rate 500
f100
home X Y Z A B x y z
seconds
absolute

ROTATION 1 InputAxes x y z OutputAxes X Y Z
ROTATION 1 ANGLE A 1, 0
ROTATION 1 Displacement 10, 0, -15
ROTATION 2 InputAxes z x y OutputAxes Z X Y
ROTATION 2 ANGLE B 1, 0
ROTATION 2 Displacement 12, 0, 0

DependentCoordinatedAccelLimit = 4000

;move physical X/Y/Z to the starting location
;this location is machine and part dependent
;and has no effect on the ROTATION operation
G0 X 5 Y60 Z23
;move rotary axes to starting location
G0 A0 B0
;move virtual axes to starting location
G0 x9 y0
scopetrig
ROTATION 1 ON
ROTATION 2 ON

G1 x10 y2 z1 A2 B3
G1 x12 y3 z1.4 A2.5 B3.7
G1 x9 y2 z1.8 A1.5 B3.8
G1 x8 y2 z2.2 A0 B0

ROTATION 1 OFF
ROTATION 2 OFF
```

Reference Material

Search the A3200 Help File index for more information on the topics listed below. Most of these entries will have multiple items that provide additional information on all of the commands used in this document.

- *ROTATION*
- *DependentCoordinatedAccelLimit*
- *TrajectoryFIRFilter*
- *Axis Type Overview*
- *Lookahead*