

**Doc No.:** AN-232

**Version:** 1.1

**Date:** 27 February 2009

**Subject:** SCARA ARM ROBOT TRANSFORMATION – FRAME 1

## APPLICATION NOTE

[www.triomotion.com](http://www.triomotion.com)

### 1. What is “Frame 1” ?

“Frame 1” is built in to Motion Coordinators from version 1.6607 onwards and is available in certain special versions before that. “Frame 1” is the name given to the mathematical transformations required to simplify control of mechanisms such as simple 2 axis “SCARA” robots. See illustration below. Using the frame transformation allows the mechanism to be programmed in X-Y coordinates. The transformation continuously converts the programmed XY positions into the axis 0 and 1 positions.

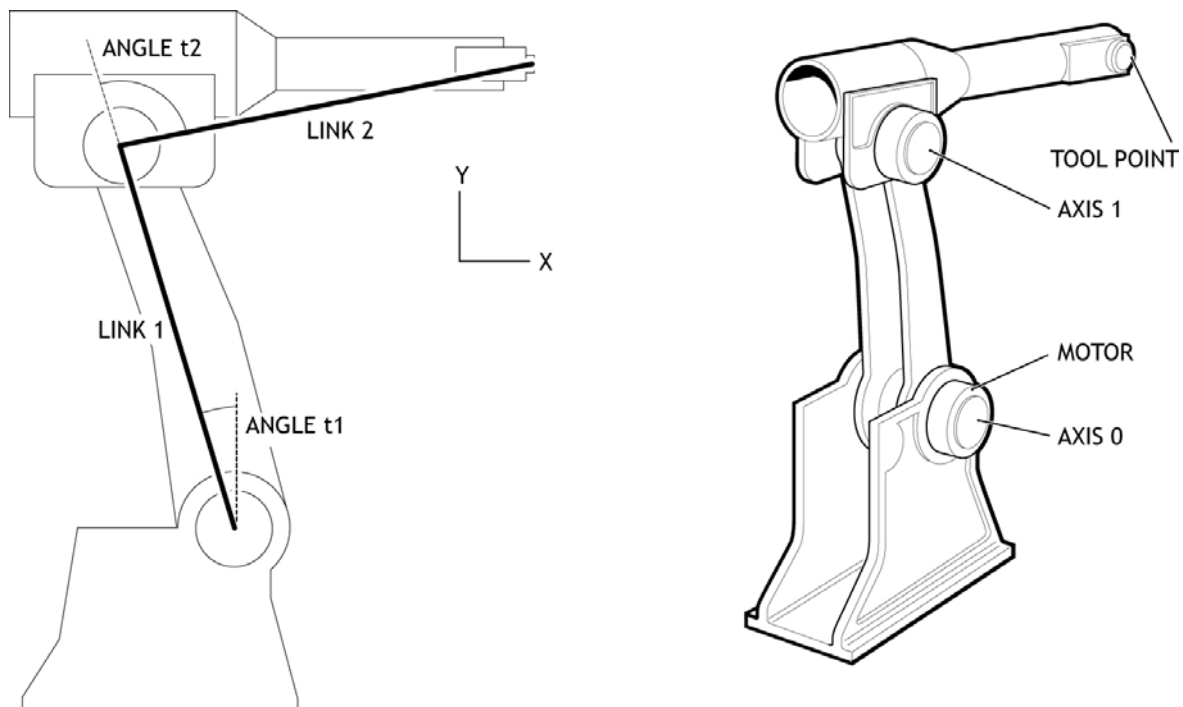


Figure 1: 2-Axis Robot. Second motor is carried on the end of Link 1

## 2. Inverse (Reverse) Transformation:

When the “Frame 1” is first switched on an inverse of the transformation is used to convert axis positions into the X and Y positions. For this process to work the Reverse Transformation has to make some assumptions:

1. The axes are already datumed and their angles set relative to a determined geometric position
2. The link lengths must be known
3. The motor encoder edges/radian must be known

The parameters are illustrated on the diagram to the left of figure 1 above:

Note how the 2 motors’ absolute positions are datumed relative to the “straight up” position. In other words, the zero angle on each axis is with the arms in line and vertical. Of course it is not necessary for the motors to actually go to this position:

## 3. Parameter Setting:

Some things to note:

- The link lengths and offsets are in microns. (mm/1000)
- The encoder ratios are in position edges/radian
- The six required parameters are entered into TABLE positions 0..5
- The FRAME only works on axes 0 and 1. This cannot be changed. Other axes work as usual.

Here is an example:

```

'
' Enter Configuration Parameters:
'
TABLE(0,300000) ' Length of arm 1 in mm * 1000
TABLE(1,445000) ' Length of arm 2 in mm * 1000
TABLE(2,2546.4791) ' edges per radian for joint 1
TABLE(3,2546.4791) ' edges per radian for joint 2
TABLE(4,0) ' Internal value. Set to 0 to force frame re-calculation
TABLE(5,1) ' set to 1 for second joint fixed to arm 1

```

## 4. Homing

As mentioned in section 2, the 2 motors must be datumed BEFORE they are switched into FRAME=1. Wherever the home switches are placed, the positions of axes 0 and 1 need to be defined so that the effective zero angle is with both arms vertical.

```

FRAME=0

DATUM(3) AXIS(0) ' find home switch for lower part of arm
WAIT IDLE
DATUM(3) AXIS(1) ' find upper arm home position
WAIT IDLE
' The mechanical layout may make it impossible to home at (0,0)
' Define the home position values as their true angle (in edges)
DEFPOS(-3555,6667) ' say home position is -80 deg and +150 deg
WAIT UNTIL OFFPOS=0

```

```
' move both arms to start position PI/4 radians (45 degrees)
MOVEABS (-TABLE (2) *0.7854, TABLE (3) *0.7854*2)
WAIT IDLE

WA (100)
FRAME=1
```

After setting FRAME=1 the tip of the second arm will be set with X, Y as (0,42426) in the example. This effectively makes the (0,0) XY position to be the bottom joint of the lower arm.

All the normal move types can then be run within the FRAME=1 setting until it is reset by setting FRAME=0. The FRAME=1 mathematics effectively make the resolution of axes 0 and 1 microns. The usual UNITS parameter can be set to work in mm:

For example:

```
UNITS AXIS (0)=1000
UNITS AXIS (1)=1000
```

## 5. DPOS and TRANS\_DPOS

When running in the FRAME=1 mode the axis positions before the transformation are stored in axis parameter DPOS. The positions after the transformation are stored in TRANS\_DPOS. When FRAME=0 these values are 1:1. The MPOS values effectively track TRANS\_DPOS. The MPOS display in Motion Perfect can often be used as a quick way to examine TRANS\_DPOS.

## 6. Second motor in base

FRAME=1 can also handle the situation where the second motor is located in the base of the robot and the upper arm is moved using a linkage or drive belt from the base. See figure 2 below.

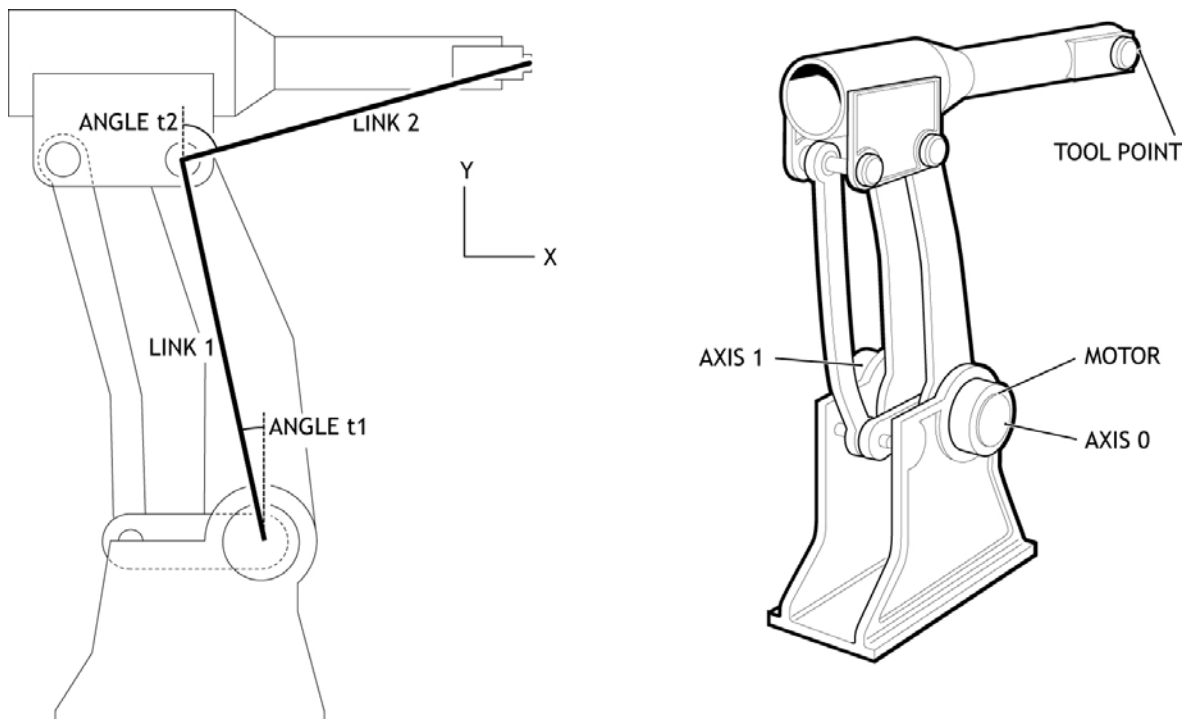


Figure 2: 2-Axis Robot. Second motor in base with link arm to move upper part.

With this mechanical layout, the angle t2 of the motor on axis 1 is related to the base and not to the lower arm as is the situation when the motor is carried on the arm. This arrangement changes the way the FRAME must calculate the joint angles.

- When both motors are fixed in the base set the value of TABLE(5) to 0.
- When axis 1 motor is carried on the end of the arm, set TABLE(5) to 1.

Example for both motors in base:

```

,
' Enter Configuration Parameters:
,
TABLE(0,400000) '          Link 1 in mm * 1000
TABLE(1,250000) '          Link 2 in mm * 1000
TABLE(2, 4096*5/(2*PI)) ' t1 in edges per radian
TABLE(3, 4096*3/(2*PI)) ' t2 in edges per radian
TABLE(4,0) ' Internal value. Set to 0 to force frame re-calculation
TABLE(5,0) ' set to 0 for second joint fixed to base

```

As well as the TABLE initialisation, the calculation for the first move to a starting position, which is done with FRAME 1 off, has a different expression for the second joint angle.

```

' move both arms to start position PI/4 radians (45 degrees)
MOVEABS(-TABLE(2)*0.7854, TABLE(3)*0.7854)

```

After the FRAME=1 command, the arms are controlled in exactly the same way for both types. It makes no difference whether the second motor is in the base or carried on the arm. All moves commands are made using X and Y coordinates in mm \* 1000.

## 7. Motion Commands

After the FRAME is running, all move commands are available to control the axes in X and Y.

- MOVE(x, y)
- MOVEABS(pos\_x, pos\_y)
- MOVE(x) AXIS(0) or MOVE(y) AXIS(1)
- MOVEABS(pos\_x) AXIS(0) or MOVEABS(pos\_y) AXIS(1)
- MOVEMODIFY(pos) ' on either axis 0 or axis 1
- MOVELINK(dist, linkdist, accdist, decdist, master\_axis)
- CAMBOX(tstart, tend, mult, linkdist, master\_axis)

This makes the application of a robot arm for many pick-and-place, welding, handling and other uses very easy and versatile. The linked moves especially allow the robot arm to be used where the tool must follow a conveyor or similar.

If other axes are used to control a “wrist” attached to the tool point, these can be controlled by the usual axis commands without needing a further frame transform. For example a tool can be held at a constant angle to the base datum by using the ADDAX command to maintain the relationship between tool angle and base.

Full kinematics for controlling a 3 axis wrist and rotating base are beyond the scope of this frame transform and would be handled by an external processing device.