Approach to Decomposing Joint Transformation Matrix

Extracting Joint to Joint transformation matrices from the robot joint vectors which include a position vector and a compound transformation matrix.

AP=(x,y,z)

θ 2

θ3

1

θ4

θ1

APDORG

YA

XA

{A}

{D}

{C}

{B}

Step 1: Set a point vector #P to the home position via #P=#PPOINT(0,0, …,0).

Step 2: Execute the ROTJOINT V+ program which outputs a compound transformation matrix for the point #P. Set the rotation angle to 30° just for the joint in question, with all others to zero.

Step 3: Set up transformation equations and successively evaluate the LHS with the RHS from the program output and the prior calculations.

**

**

Step 4: Replace the non-zero or one numeric values in the rotation matrix portion of the transformation with either sinθ = 0.50 or cosθ=0.866.

Alternative: Manually generate all joint frame work in trigonometric equation form and compare the results by entering θ = 0°, 30°, 60°, 90°, etc. for selected joints.

*Adept 550 SCARA robot joint configuration*

*The Adept 550 follows the convention in frame structuring, namely, the joint rotation or translation is about or along Z axis and joint-to-joint frame rotation is done about X axis. The following is a complete set of transformation matrices which has been numerically verified with ROTJOINT.*

*Joint 1 Joint 2 Joint 3 Joint 4*

z

z

y

x

x

x

x

z

y

z

**



when θ4=0. when θ4=90°.

The Adept transformation vectors carry the compound values of . The 4th column vector is



where L1=300 mm, L2=250 mm, and D is the maximum reach in Z direction. The model number “550” comes from L1+L2.

*Adept Six 300 robot joint configuration*

*Assuming that the Adept six axis robot follows the convention in frame structuring, namely, the joint rotation or translation is about or along Z axis and joint-to-joint frame rotation is done about X axis. The following is a set of the transformation matrices which has been partially verified with ROTJOINT.*

*Joint 1 Joint 2 Joint 3 Joint 4 Joint 5 Joint 6*

z

y

y

z

z

z

y

z

z

y

x

y

x

y

x

x

x

x

**

 

