

Trsf变量的理解

Written By 张华君

pPoint.trsf表示：工具坐标系相对与基坐标系的位姿

(注：记录一个pPoint点，可以理解为记录pPoint点处的工具坐标系)

fRame.trsf表示：用户自定义坐标系相对于基坐标系的位姿

tTool.trsf表示：工具坐标系相对于法兰坐标系的位姿

因为STAUBLI机器人坐标系转换参照第一类欧拉角，三者乘序不可以改变！！

只有当坐标系姿态一致时，才可以实现平移！！平移的顺序可以改变！！

Staubli机器人[trsf]表达：

[X][Y][Z][RX][RY][RZ] ;
[X][Z][Y][RX][RY][RZ] ;
[Y][X][Z][RX][RY][RZ] ;
[Y][Z][X][RX][RY][RZ] ;
[Z][X][Y][RX][RY][RZ] ;
[Z][Y][X][RX][RY][RZ] ;

机器人学导论表达两个关节的[trsf]，[RX][X][RZ][Z]。

Staubli机器人表达两个关节的[trsf]，是先移动[X][Z]再旋转[RX][RZ]。

严重注意这里[RX][X][RZ][Z]=[X][Z][RX][RZ]，但 X、Z、RX、RZ不相等！！

因为从一个坐标系到另外一个坐标系的有很多种表达方式，但最终[trsf]一样！！

1.探讨：point pP 的RX RY RZ代表什么意思？？

// Link()函数把l_pRX, l_pRY, l_pRZ, l_pRXYZ 转换到World下(点的trsf不会变)

// 需要赋值一个pP

// 结论：点pP的6个参数，就是当前工具坐标系

GetPointRxRyRz

begin

userPage()

cls()

link(l_pRX,world)

link(l_pRY,world)

link(l_pRZ,world)

link(l_pXYZ,world)

//

pP.trsf={11,22,33,44,55,66}

l_pXYZ.trsf={0,0,0,0,0}

//

l_pXYZ=appro(l_pXYZ,{pP.trsf.x,pP.trsf.y,pP.trsf.z,0,0,0})

l_pRX=appro(l_pXYZ,{0,0,0,pP.trsf.rx,0,0})

l_pRY=appro(l_pRX,{0,0,0,pP.trsf.ry,0})

l_pRZ=appro(l_pRY,{0,0,0,0,pP.trsf.rz})

pResult=l_pRZ

//

putln(pP.trsf.x)

putln(pP.trsf.y)

putln(pP.trsf.z)

putln(pP.trsf.rx)

putln(pP.trsf.ry)

putln(pP.trsf.rz)

putln("")

putln(pResult.trsf.x)

putln(pResult.trsf.y)

putln(pResult.trsf.z)

putln(pResult.trsf.rx)

putln(pResult.trsf.ry)

putln(pResult.trsf.rz)

End

结果发现 pP=pResult

结论：工具坐标系先移动到{11,22,33,0,0,0}再绕 X 轴旋转{0,0,0,44,0,0}Y 轴旋转{0,0,0,0,55,0}Z 轴旋转{0,0,0,0,0,66}
最终得到的 pResult 点与 pP 点一致。

2.探讨 : frame fA 的RX RY RZ代表什么意思??

GetFrameRxRyRz

// Link()函数把l_pRX, l_pRY, l_pRZ, l_pRXYZ 转换到World下 (点的trsf不会变)
// 需要赋值一个fA
// 结论 : 坐标系fA的6个参数 , 可以理解一个已只且固定的工具坐标系

```
begin
    userPage()
    cls()
    link(l_pRX,world)
    link(l_pRY,world)
    link(l_pRZ,world)
    link(l_pXYZ,world)
    fA.trsf={11,33,55,66,77,88}
    l_pXYZ.trsf={0,0,0,0,0,0}
    //
    l_pXYZ=appro(l_pXYZ,{fA.trsf.x,fA.trsf.y,fA.trsf.z,0,0,0})
    l_pRX=appro(l_pXYZ,{0,0,0,fA.trsf.rx,0,0})
    l_pRY=appro(l_pRX,{0,0,0,fA.trsf.ry,0})
    l_pRZ=appro(l_pRY,{0,0,0,0,fA.trsf.rz})
    fResult.trsf=l_pRZ.trsf
end

//
println(fResult.trsf.x)
println(fResult.trsf.y)
println(fResult.trsf.z)
println(fResult.trsf.rx)
println(fResult.trsf.ry)
println(fResult.trsf.rz)
println("")
println(fA.trsf.x)
println(fA.trsf.y)
println(fA.trsf.z)
println(fA.trsf.rx)
println(fA.trsf.ry)
println(fA.trsf.rz)
```

3.探讨appro函数

猜想 : 一个已知点右乘 [P][Trsf] , 得到 pAppro 点

GetApproTcp()

```
begin
    userPage()
    cls()
    link(l_pA1,world)
    l_pA1.trsf={10,20,30,15,25,35}
    l_trA={1,2,3,4,5,6}

    l_pA2.trsf=l_pA1.trsf*l_trA

    l_pA3=appro(l_pA1,l_trA)

    println(l_pA2.trsf.x)
end

println(l_pA2.trsf.y)
println(l_pA2.trsf.z)
println(l_pA2.trsf.rx)
println(l_pA2.trsf.ry)
println(l_pA2.trsf.rz)
println("")
println(l_pA3.trsf.x)
println(l_pA3.trsf.y)
println(l_pA2.trsf.z)
println(l_pA3.trsf.rx)
println(l_pA3.trsf.ry)
println(l_pA3.trsf.rz)
```

结论 : l_pA2=l_pA3 所以 appro 函数等于右成一个[Trsf]

GetComposeTcp1()

//点在坐标系里偏移

begin

 userPage()

 cls()

 //l_pC1 是fA坐标系下的一点 对应fA下的 l_pfA1

 //l_pC2 是fB坐标系下的一点 对应fA下的 l_pfA2

 link(l_pC1,fA)

 link(l_pC2,fA)

 link(l_pC3,fA)

 l_pC1.trsf={10,20,30,40,50,60}

 fA.trsf={15,25,35,45,55,65}

 l_trC={1,2,3,4,5,6}

$l_pC2.trsf=l_trC*l_pC1.trsf$

 l_pC3=compose(l_pC1,fA,l_trC)

 putln(l_pC2.trsf.x)

 putln(l_pC2.trsf.y)

 putln(l_pC2.trsf.z)

 putln(l_pC2.trsf.rx)

 putln(l_pC2.trsf.ry)

 putln(l_pC2.trsf.rz)

 putln("")

 putln(l_pC3.trsf.x)

 putln(l_pC3.trsf.y)

 putln(l_pC3.trsf.z)

 putln(l_pC3.trsf.rx)

 putln(l_pC3.trsf.ry)

 putln(l_pC3.trsf.rz)

end