## 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I_pRX,I_pRY,I_pRZ,I_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
                                               putIn(pP.trsf.x)
begin
                                               putIn(pP.trsf.y)
 userPage()
                                               putIn(pP.trsf.z)
 cls()
                                               putIn(pP.trsf.rx)
 link(l_pRX,world)
                                               putIn(pP.trsf.ry)
 link(l_pRY,world)
                                               putIn(pP.trsf.rz)
 link(l_pRZ,world)
                                               putln("")
 link(l_pXYZ,world)
                                               putIn(pResult.trsf.x)
                                               putIn(pResult.trsf.y)
 pP.trsf={11,22,33,44,55,66}
                                               putIn(pResult.trsf.z)
 I_pXYZ.trsf = \{0,0,0,0,0,0,0\}
                                               putIn(pResult.trsf.rx)
 //
                                               putIn(pResult.trsf.ry)
I_pXYZ=appro(I_pXYZ,{pP.trsf.x,pP.trsf.y,pP.trsf.y,0,0,0})
                                               putIn(pResult.trsf.rz)
I_pRX = appro(I_pXYZ, \{0,0,0,pP.trsf.rx,0,0\})
I_pRY=appro(I_pRX,\{0,0,0,0,pP.trsf.ry,0\})
                                              结果发现 pP=pResult
I_pRZ=appro(I_pRY,\{0,0,0,0,0,pP.trsf.rz\})
pResult=I_pRZ
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I_pRX, I_pRY, I_pRZ, I_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()
                                                           putIn(fResult.trsf.x)
  cls()
                                                           putIn(fResult.trsf.y)
  link(l pRX,world)
                                                           putIn(fResult.trsf.z)
  link(l_pRY,world)
                                                           putIn(fResult.trsf.rx)
  link(I pRZ,world)
                                                           putIn(fResult.trsf.ry)
  link(l_pXYZ,world)
                                                           putIn(fResult.trsf.rz)
  fA.trsf={11,33,55,66,77,88}
                                                           putln("")
 I_pXYZ.trsf = \{0,0,0,0,0,0,0\}
                                                           putIn(fA.trsf.x)
                                                           putIn(fA.trsf.y)
 //
I_pXYZ = appro(I_pXYZ, \{fA.trsf.x, fA.trsf.y, fA.trsf.y, 0, 0, 0\})
                                                           putIn(fA.trsf.z)
I_pRX = appro(I_pXYZ, \{0,0,0,0,fA.trsf.rx,0,0\})
                                                           putIn(fA.trsf.rx)
I_pRY=appro(I_pRX,\{0,0,0,0,fA.trsf.ry,0\})
                                                           putIn(fA.trsf.ry)
                                                           putIn(fA.trsf.rz)
I_pRZ=appro(I_pRY,\{0,0,0,0,0,fA.trsf.rz\})
fResult.trsf=I_pRZ.trsf
                                                         end
3.探讨appro函数
猜想:一个已知点右乘 [P][Trsf],得到 pAppro点
                                                       putIn(I_pA2.trsf.y)
GetApproTcp()
                                                       putIn(I_pA2.trsf.z)
begin
                                                       putIn(I_pA2.trsf.rx)
  userPage()
                                                       putIn(I_pA2.trsf.ry)
  cls()
                                                       putIn(I_pA2.trsf.rz)
  link(l pA1,world)
                                                       putln("")
  I_pA1.trsf={10,20,30,15,25,35}
                                                       putIn(l_pA3.trsf.x)
  I_{trA} = \{1, 2, 3, 4, 5, 6\}
                                                       putIn(l_pA3.trsf.y)
                                                       putIn(I_pA2.trsf.z)
   _pA2.trsf=l_pA1.trsf*l_trA
                                                       putIn(l pA3.trsf.rx)
                                                       putIn(I_pA3.trsf.ry)
  l_pA3=appro(l_pA1,l_trA)
                                                       putIn(I_pA3.trsf.rz)
                                                     end
  putIn(I_pA2.trsf.x)
结论:I_pA2=I_pA3 所以 appro 函数等于右成一个[Trsf]
```

## 4.探讨position函数

begin

cls()

//原始

userPage()

```
Frame2World()
                                                                 World2Frame()
               // I_pfA是fA下的一点,在World下的表达
                                                                // I_pwA是world下的一点,在World下的表达
               begin
                                                                begin
                                                                   userPage()
                 userPage()
                                                                   cls()
                 cls()
                                                                   link(l_pwA,world)
                 link(l_pfA,fA)
                                                                   I_pwA.trsf = \{10, 20, 30, 10, 20, 30\}
                 I_pfA.trsf = \{10,20,30,10,20,30\}
                                                                  fA.trsf={15,30,45,45,30,15}
                 fA.trsf={15,30,45,45,30,15}
                                                                 . //
                 //
                                                                   //l_pfA.trsf=position(l_pwA,fA)
                 //l_pwA.trsf=position(l_pfA,world)
                                                                   I_pfA.trsf=!fA.trsf*I_pwA.trs*
                  _pwA.trsf=fA.trsf*l_pfA.trsf
                                                                   //
                 //
                                                                   putIn(l_pfA.trsf.x)
                 putIn(I_pwA.trsf.x)
                                                                   putIn(l_pfA.trsf.y)
                 putIn(I_pwA.trsf.y)
                                                                   putIn(l_pfA.trsf.z)
                 putIn(I_pwA.trsf.z)
                                                                   putIn(I_pfA.trsf.rx)
                 putIn(I_pwA.trsf.rx)
                                                                   putIn(I_pfA.trsf.ry)
                 putIn(I_pwA.trsf.ry)
                                                                   putIn(I_pfA.trsf.rz)
                 putIn(I_pwA.trsf.rz)
                                                                end
                end
5.探讨compose函数
                                                                   //过度
猜想:一个已知点左乘 [Trsf][P],得到 pCompose点
                                                                   l_pfA1.trsf=!fA.trsf*l_pC1.trsf
GetComposeTcp1()
                                                                   l_pfA2.trsf=l_trC*l_pfA1.trsf
//点不在坐标系里偏移
                                                                   I_pC2.trsf=fA.trsf*I_pfA2.trsf
                                                                   //终极
                                                                 //I pC2.trsf=fA.trsf*l trC*!fA.trsf*l pC1.trs
 //l_pC1 是World坐标系下的一点 对应fA下的 l_pfA1
                                                                   l_pC3=compose(l_pC1,fA,l_trC)
 //l_pC2 是World坐标系下的一点 对应fA下的 l_pfA2
  link(l_pC1,world)
                                                                   putIn(I_pC2.trsf.x)
  link(l_pC2,world)
                                                                   putIn(I_pC2.trsf.y)
  link(I_pC3,world)
                                                                   putIn(I_pC2.trsf.z)
  link(l_pfA1,fA)
                                                                   putIn(l_pC2.trsf.rx)
  link(I_pfA2,fA)
                                                                   putIn(I_pC2.trsf.ry)
  I_pC1.trsf = \{10,20,30,40,50,60\}
                                                                   putIn(I_pC2.trsf.rz)
  fA.trsf={15,25,35,45,55,65}
                                                                   putIn("")
  I_trC={1,2,3,4,5,6}
                                                                   putIn(I_pC3.trsf.x)
                                                                   putIn(I_pC3.trsf.y)
 //l_pfA1.trsf=position(l_pC1,fA)
                                                                   putIn(I_pC3.trsf.z)
 //l_pfA2.trsf=l_trC*l_pfA1.trsf
                                                                   putIn(I_pC3.trsf.rx)
 //l_pC2.trsf=position(l_pfA2,world)
                                                                   putIn(l_pC3.trsf.ry)
                                                                   putIn(I_pC3.trsf.rz)
```

end

## 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

```
I_pC3=compose(I_pC1,fA,I_trC)

putIn(I_pC2.trsf.x)

putIn(I_pC2.trsf.y)

putIn(I_pC2.trsf.z)

putIn(I_pC2.trsf.rx)

putIn(I_pC2.trsf.ry)

putIn(I_pC2.trsf.rz)

putIn(I_pC2.trsf.rz)

putIn(I_pC3.trsf.x)

putIn(I_pC3.trsf.y)

putIn(I_pC3.trsf.z)

putIn(I_pC3.trsf.z)

putIn(I_pC3.trsf.rx)

putIn(I_pC3.trsf.rx)

putIn(I_pC3.trsf.rx)
```

end