Dov Cattan Homework 1 COT 4930

1.1

a. Rotate -pi/4 radians twice

b. Move 6.5 in the y direction

c. Rotate pi/4 radians twice

d. Move 23.5 in the y direction

e. Rotate -pi/4 radians twice

f. Move 20 in the y direction

g. Rotate pi/4 radians twice

e. Move 5 in the y direction

1.2

All post multiplications because every translation in the 3x3 matrix is made based on the perspective of the robot itself

1.3

-1.5,-8.5 to 26.5,28.5

1.4

plotvol([0 30 0 30])

1.5

T0=transl2(-1.5,-8.5)

R0=T0*trot2(-pi/4)

R1=R0*trot2(-pi/4)

T1=R1*transl2(0,6.5)

R2=T1*trot2(pi/4)

R3=R2*trot2(pi/4)

T2=R3*transl2(0,23.5)

R4=T2*trot2(-pi/4)

```
R5=R4*trot2(-pi/4)
T3=R5*transl2(0,20)
R6=T3*trot2(pi/4)
R7=R6*trot2(pi/4)
T4=R7*transl2(0,5)
tranimate2(T0,R0,'nsteps',50)
tranimate2(R1,T1,'nsteps',50)
tranimate2(T1,R2,'nsteps',50)
tranimate2(R3,T2,'nsteps',50)
tranimate2(R4,R5,'nsteps',50)
tranimate2(T3,R6,'nsteps',50)
tranimate2(R7,T4,'nsteps',50)
```

2.1

```
plotvol([-3.5 35 -5 45 0 5])
T1 = transl(-0.25,-3.5,0)
R1 = T1*trotz(deg2rad(90))
tranimate(T1,R1,'nsteps',50)
T3 = transl(0,23.5,0)
T4 = T3*troty(deg2rad(90))
tranimate(T3,T4,'nsteps',50)
T5 = transl(6,0,0)*T4
T6 = T5*trotz(deg2rad(90))
tranimate(T5,T6,'nsteps',50)
T7 = transl(0,20,0)*T6
T8 = T7*trotz(deg2rad(90))
tranimate(T7,T8,'nsteps',50)
T9 = transl(-6,0,0)*T8
T10=T9*trotx(deg2rad(90))
```

```
tranimate(T9,T10,'nsteps',50)
```

2.2

```
%Initialize Rotations

negrotx = rpy2tr(-90,0,0,'deg');

posrotx = rpy2tr(90,0,0,'deg');

negroty = rpy2tr(0,-90,0,'deg');

posroty = rpy2tr(0,90,0,'deg');

negrotz = rpy2tr(0,0,-90,'deg');

posrotz = rpy2tr(0,0,90,'deg');

%Initalize the positions of the back and front of the couch

backinitialposition = [1 0 0 0; 0 1 0 -7; 0 0 1 0; 0 0 0 1];

frontinitialposition = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];

plotvol([-17 17 0 45 0 15])

trplot(backinitialposition,'color','r');

trplot(frontinitialposition,'color','b');

%translations/rotations

move1=backinitialposition*transl(0,7,7)*negrotx

move2=move1*transl(0,7,7)*negrotx

move3 = frontinitialposition*negrotx

move4 = move3*negrotx

move5=move4*transl(0,-7,-7)*negrotx

move6 = move2*negrotx

move7 = move6*negrotx

move8 =move5*transl(0,-7,-7)*negrotx

move9=move8*negrotx

move10=move7*transl(0,7,7)*negrotx

move11=move9*posrotz

move12=move10*transl(7,7,0)*posrotz
```

move13=move12*posrotz

move14=move11*transl(-7,-7,0)*posrotz

move15=move13*posrotx

move16=move14*transl(0,-7,7)*posrotx

move17=move16*posrotx

move18 =move15*transl(0,7,-7)*posrotx

move19=move17*posrotx

move20=move18*transl(0,7,-7)*posrotx

move21=move20*posrotx

move22=move21*transl(4,0,0)

%animations

tranimate(move1,move2,'nsteps',50)

tranimate(move3,move4,'nsteps',50)

tranimate(move5,move6,'nsteps',50)

tranimate(move7,move8,'nsteps',50)

tranimate(move9,move10,'nsteps',50)

tranimate(move11,move12,'nsteps',50)

tranimate(move13,move14,'nsteps',50)

tranimate(move15,move16,'nsteps',50)

tranimate(move17,move18,'nsteps',50)

tranimate(move19,move20,'nsteps',50)

tranimate(move21,move22,'nsteps',50)

3.1

plotvol([-20 20 -20 20 0 15])

%Initial Position

I = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]

% Translations and rotations

```
T1 = transl(6.5,0,6)*I

R1 = troty(-pi/2)*T1

T2 = transl(-1,0,0)*R1

T3 = transl(0,0,10)*T2

R2 = troty(-pi/2)*T3

T4 = transl(5.5,6,0)*R2

T5 = transl(0,0,3.5)*T4

T6 = transl(10,-16,1)*T5

T7 = transl(0,0,-7)*T6

%animations

tranimate(I,T1,'nsteps',50)

tranimate(R1,T2,'nsteps',50)

tranimate(T3,R2,'nsteps',50)

tranimate(T4,T5,'nsteps',50)

tranimate(T6,T7,'nsteps',50)
```

Step 1:

The end effector start position is (-5.0,6.5)

We orientate the robot towards the opposite side of the glued side by rotating by -90 degrees around the y-axis

Step 2:

We now translate the end effector by -1 units to approach the box

Step 3:

Lift the box up by 10 units

Step 4:

Rotate the end effector towards the global negative Z axis (rotation of -90 degrees around the y axis). This will face the glued side in the direction of the sticky side of the other box.

Step 5:

The robot now translates to the X,Y position of the lower box by translating 5.5 in X and 6 in Y.

4.1

```
R1 = eul2r(30,-15,54,'deg')
```

4.2

```
R1e = expm(R1)
```

4.3

```
% axis of rotation unit vector and the angle of rotation for R1
[th,w] = tr2angvec(R1)
```

4.4

```
q = UnitQuaternion(R1)
```

4.5

```
%Rotation Matrix
R2 = R1 * R1
%4.2-4.4 for R2 instead of R1
R2e = expm(R2)
% axis of rotation unit vector and the angle of rotation for R2
[th, w] = tr2angvec(R2)
q = UnitQuaternion(R2)
```