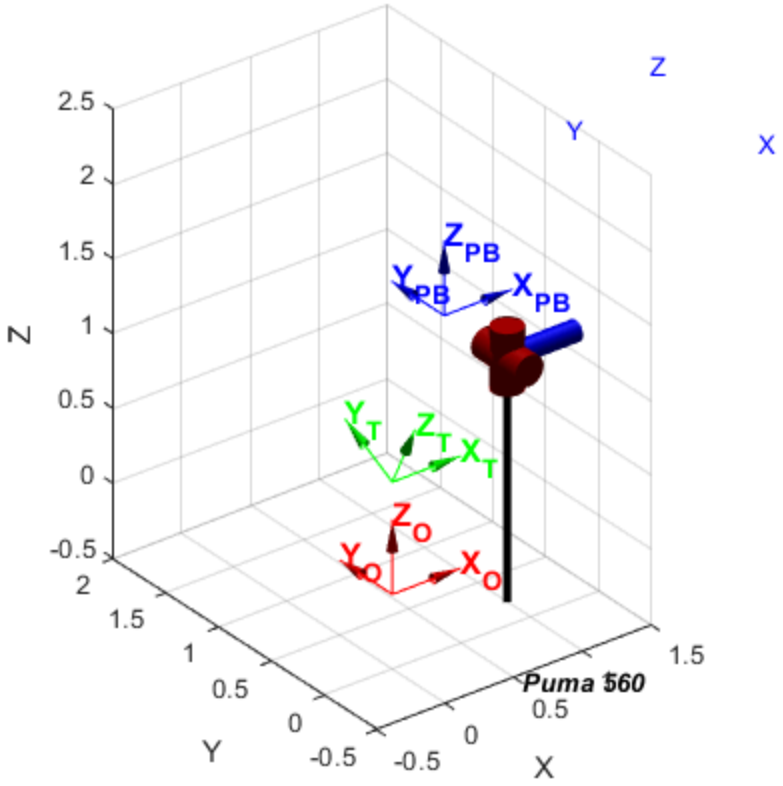
Table of Contents

Sketching the enviroment of the robotics work cell.

```
%Clean variables and closing windows
clc;
clear;
close all;

%Some Important Information
load('Data_groove_weld_fv_torus.mat')  % This is some data you can use
tableDim = [1 1.8]; %Table dimensions
tableHeight = 0.75;     %Table height
tableAlfa = pi/7;  %Table rotation (20º)

radiusExtTorus = 0.90;
diamExtTorus = radiusExtTorus*2;
diamTube = 0.2;
radiusTube = diamTube/2;
amplitudToro = 0.8;
weigthExtTorus = amplitudToro+diamTube;

numHoles = 8;
drillHoleRadius = 0.010;

numTurns = 8;
WeldPointPerTurn = 8;

toolLenght = 0.2;

%Main reference frames.
sketch = figure;
FO = eye(4);
HO=trplot(FO, ... % Plot frame T at the origin
    'frame', 'O', ...
    'color', 'R',...
    'text_opts', {'FontSize', 12, 'FontWeight', 'bold'},...
    'length',0.5,...
    'arrow',...
    'width', 1.2);
hold on;

FT = transl(0,0,tableHeight)*trotx(tableAlfa);
```

```
HT=trplot(FT, ... % Plot frame T at the origin
    'frame', 'T', ...
    'color', 'G',...
    'text_opts', {'FontSize', 12, 'FontWeight', 'bold'},...
    'length',0.5,...
    'arrow',...
    'width', 1.2);
hold on;

FPB = transl(tableDim(1),cos(tableAlfa)*tableDim(2)/2,
(sin(tableAlfa)*tableDim(2)/2)+tableHeight);
HPB=trplot(FPB, ... % Plot frame PB at the origin
    'frame', 'PB', ...
    'color', 'B',...
    'text_opts', {'FontSize', 12, 'FontWeight', 'bold'},...
    'length',0.5,...
    'arrow',...
    'width', 1.2);
hold on;

%Plot the robot Puma
mdl_puma560;
p560.links(2).a = 0.9;
p560.links(4).d = 0.9;
p560.base = FPB*transl(0,-0.6,0);
p560.plot(qz);
hold on;
axis([-0.5 1.5 -0.5 2 -0.5 2.5]);
```

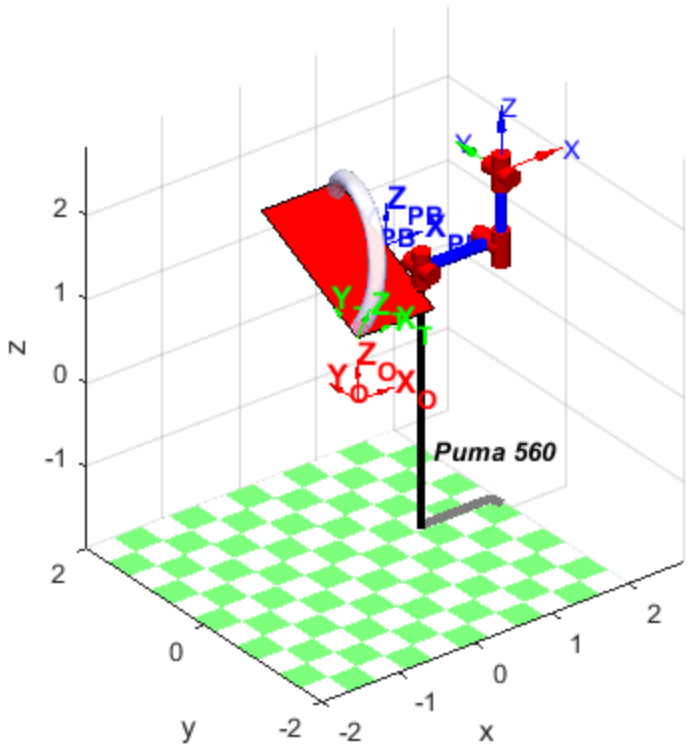
```
Warning: floor tiles too small, making them 2.000000 x bigger - change
 the size
or disable them
```

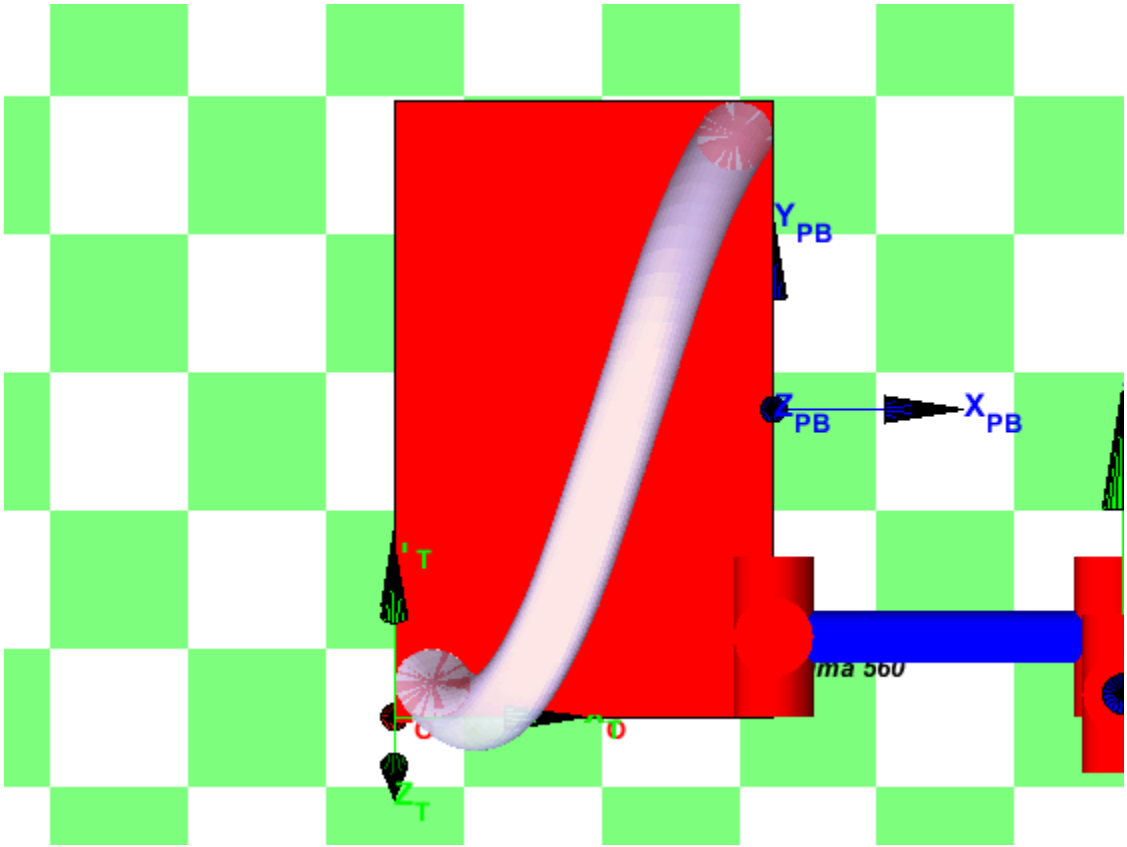


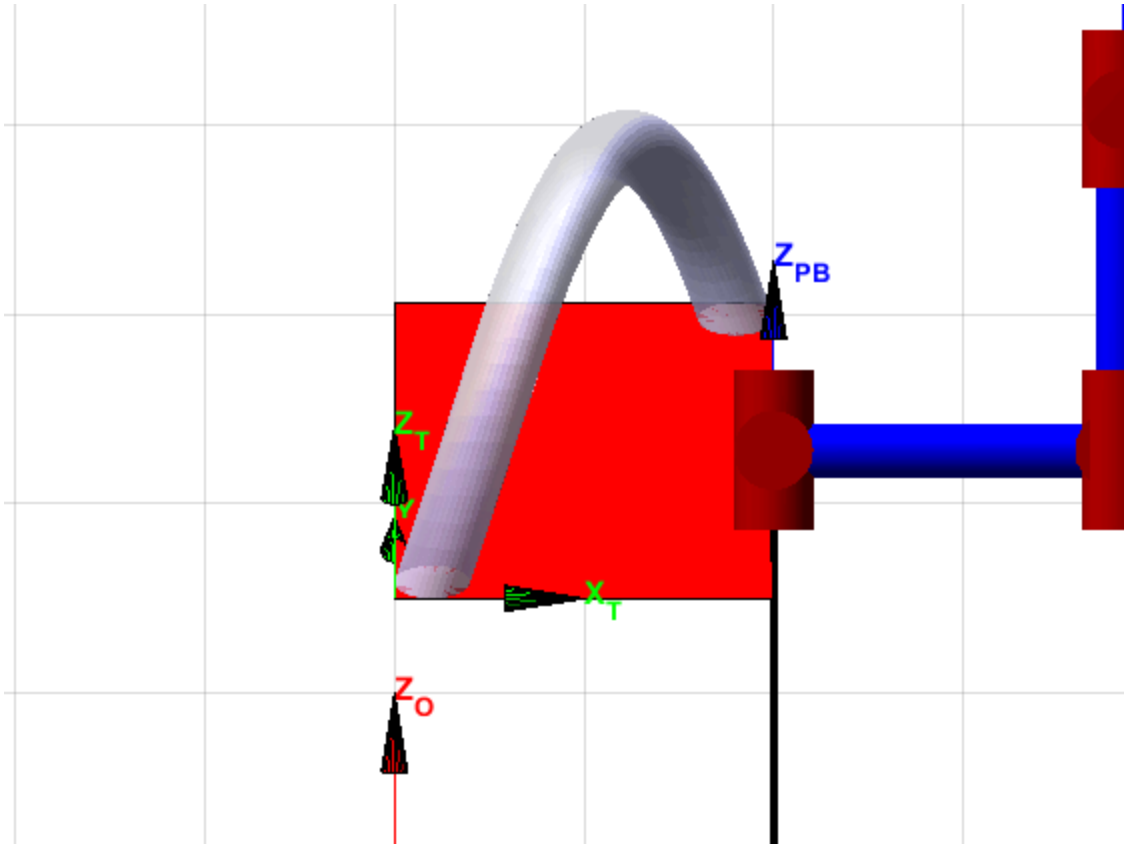
Draw the working table and the torus in working position.

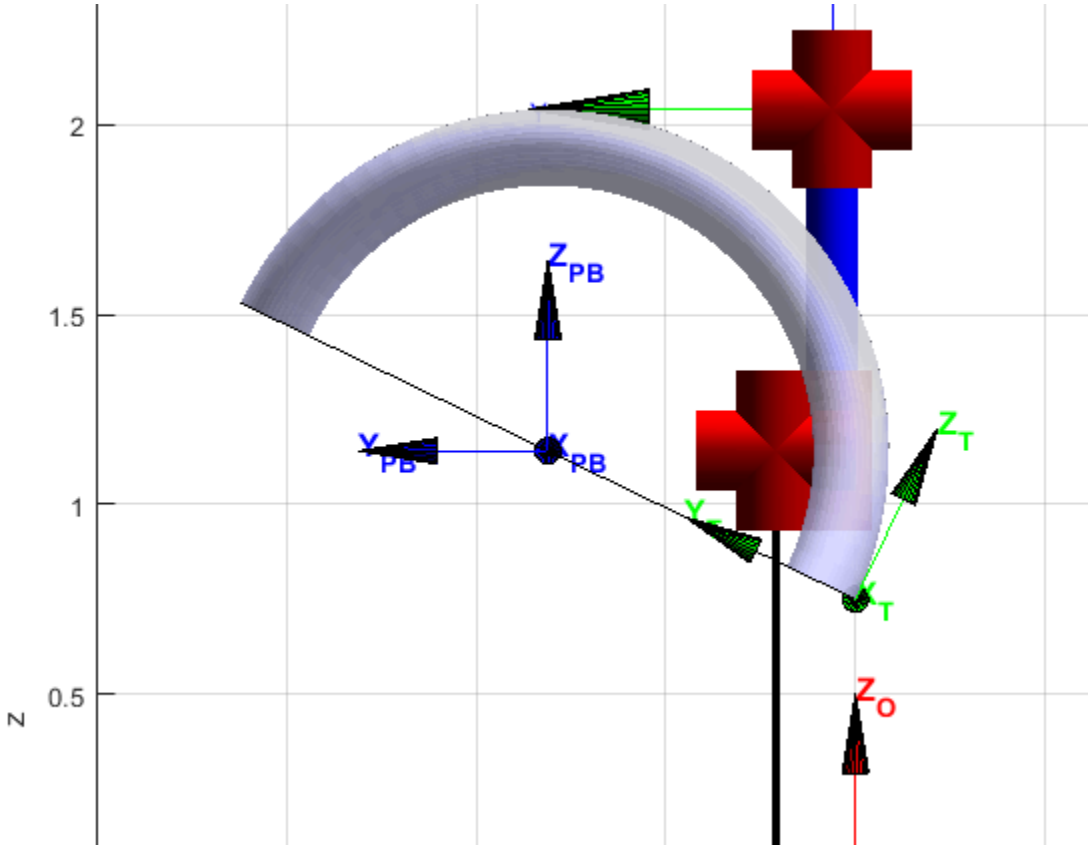
```
%Drawing the Table
coordX = tableDim(1);                    %Max point on X coordinates of
 the table
coordY = cos(tableAlfa)*tableDim(2);     %Max point on Y coordinates of
 the table
coordZ = sin(tableAlfa)*tableDim(2);     %Max point on Z coordinates of
 the table

xlabel('x');
ylabel('y');
zlabel('z');
fill3([0 coordX coordX 0],[0 0 coordY coordY],[0 0 coordZ
 coordZ]+tableHeight,'r');


%Drawing the Torus
fv=stlread('Toro_Robotica.stl');% fv is a struct with faces and
 vertices

%Rotate to get it facing up
fv.vertices=fv.vertices*rotx(pi/2);
```

```
%Translating the torus to [0,0,0]
mi=min(fv.vertices);
fv.vertices=fv.vertices-mi;

%Taking the bounding box of the torus
ma=max(fv.vertices);
mi=min(fv.vertices);
dmami=ma-mi;

%Positioning and posing the torus
fv.vertices=(fv.vertices)*rotz(-pi/2)*rotx(-tableAlfa);
fv.vertices=fv.vertices+[diamTube+amplitudToro+(tableDim(1)-diamTube-
amplitudToro)/2 0 tableHeight];

SS=patch(fv,...
  'FaceColor',  [0.8 0.8 1.0], ...
        'EdgeColor',       'none',        ...
        'FaceLighting',    'gouraud',     ...
        'AmbientStrength', 0.15);

% Add a camera light, and tone down the specular highlighting
camlight('headlight');
material('dull');
alpha (SS,0.7);

% Fix the axes scaling, and set a nice view angle
% axis 'equal'
axis([-0.5 1.5 -0.5 2 -0.5 2.5]);
axis 'equal';

% Save figure to use it later
savefig(sketch,'SketchFig.fig');

%%Give diferent points of view of the scenary: Top, Front, Lateral and
 isometrics view.
%Top View
openfig('SketchFig.fig');
camtarget([tableDim(1)/2, coordY/2, 0]);
campos([tableDim(1)/2, coordY/2, 10]);

%Frontal View
openfig('SketchFig.fig');
camtarget([tableDim(1)/2, 0, coordX/2+tableHeight]);
campos([tableDim(1)/2, -10, coordX/2+tableHeight]);

%Lateral View
openfig('SketchFig.fig');
camtarget([0, coordY/2, coordX/2+tableHeight]);
campos([-10, coordY/2, coordX/2+tableHeight]);
```


Z
X
Y
Z
PB
X
PB
Y
Z
X
T
Z
O
Y
O
X
O
Puma 560
2
1
0
-1
z
2
0
-2
y
-2
-1
0
1
2
x


Y
PB
Z
PB
X
PB
T
ma 560
O
Z
T


Z_PB
Z_T
Y_T
X_T
Z_O

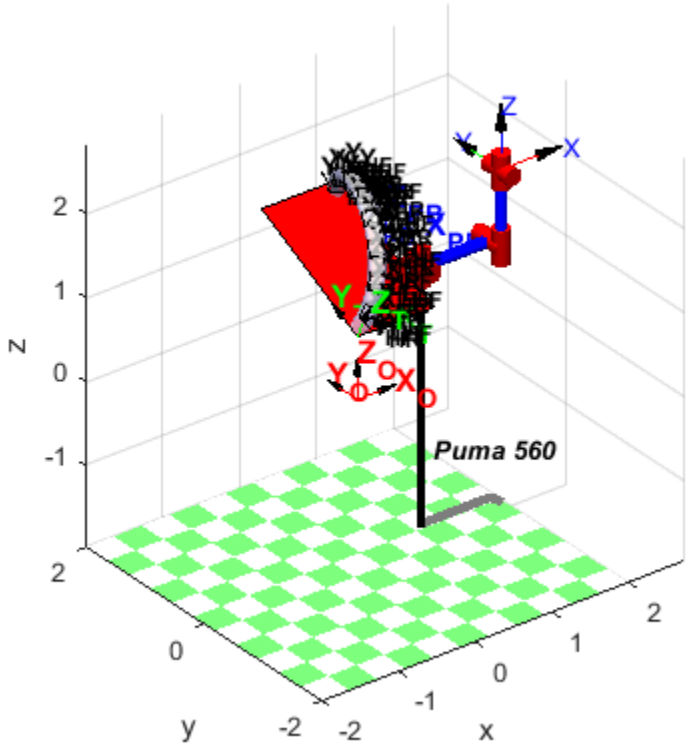

2
1.5
1
0.5
z
Z_PB
Y_PB
X_PB
Z_T
Y_T
X_T
Z_O

Working points.

Give here your code to get the variables to locate: a) The reference frame for all drills holes, such that z-axis is orthogonal to the surface of the torus and the x-axis is in the direction of minimun curvature. Draw in scale the frames

```
drillsFig = openfig('SketchFig.fig');
for i=0:numHoles-1
    PoseTool = trotx(pi/2)*troty(pi/2)*trotx(pi+pi/10)*trotz(pi/2);
    %position before start drilling
    Drilling1_Poses(:,:,i*3+1) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*i/
numHoles)*transl(amplitudToro*i/numHoles,0,0)*transl(-
tableDim(1)+diamTube+toolLenght+0.1,-radiusExtTorus
+radiusTube,0)*PoseTool;
    %position drilling
    Drilling1_Poses(:,:,i*3+2) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*i/
numHoles)*transl(amplitudToro*i/numHoles,0,0)*transl(-
tableDim(1)+diamTube+toolLenght,-radiusExtTorus
+radiusTube,0)*PoseTool;
    %position after start drilling
    Drilling1_Poses(:,:,i*3+3) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*i/
numHoles)*transl(amplitudToro*i/numHoles,0,0)*transl(-
tableDim(1)+diamTube+toolLenght+0.1,-radiusExtTorus
+radiusTube,0)*PoseTool;
    %position of the hole
    Drill1_Plot(:,:,i+1) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*i/
numHoles)*transl(amplitudToro*i/numHoles,0,0)*transl(-
tableDim(1)+diamTube,-radiusExtTorus+radiusTube,0)*PoseTool;

    HH1=trplot(Drill1_Plot(:,:,i+1), ... % Plot frame PB at the origin
        'frame', 'HF', ...
        'color', 'K',...
        'text_opts', {'FontSize', 10, 'FontWeight', 'bold'},...
        'length', 0.2,...
        'arrow',...
        'width', 0.35);
    hold on;

end

hold on;
axis([-0.5 3 -0.5 3 -0.5 3]);

for i=0:numHoles-1
    PoseTool = trotx(pi/2)*troty(pi/2)*trotx(pi/10)*trotz(pi/2);
```

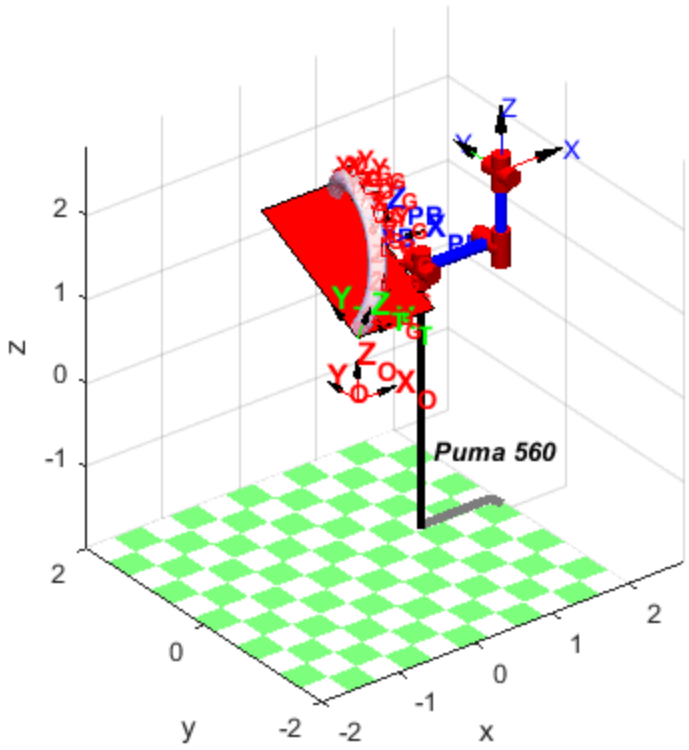
```
    %position of the hole
    Drill2_Plot(:,:,i+1) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*i/
numHoles)*transl(amplitudToro*i/numHoles,0,0)*transl(-tableDim(1),-
radiusExtTorus+radiusTube,0)*PoseTool;

    HH2 = trplot(Drill2_Plot(:,:,i+1), ... % Plot frame PB at the
 origin
         'frame', 'HR', ...
         'color', 'K',...
         'text_opts', {'FontSize', 10, 'FontWeight', 'bold'},...
         'length', 0.2,...
         'arrow',...
         'width', 0.35);
    hold on;
    axis([-0.5 1.5 -0.5 2 -0.5 2.5]);

end

    axis([-0.5 1.5 -0.5 2 -0.5 2.5]);
    axis 'equal';
```



b) Repeat the obove operation for the center of the milling groove. Draw this frames.

```
grooveFig = openfig('SketchFig.fig');
for i=0:numHoles-1
```

```
    PoseTool =
 trotx(pi/2)*troty(pi/2)*trotx(pi/10)*trotz(pi/2)*trotx(pi/2);
    %position before start grooving
    Grooving_Poses(:,:,i*4+1) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*(i-0.1)/
numHoles)*transl(amplitudToro*(i-0.1)/numHoles,0,0)*transl(-
tableDim(1)+radiusTube,-radiusExtTorus-toolLenght-0.1,0)*PoseTool;
    %position start grooving
    Grooving_Poses(:,:,i*4+2) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*(i-0.1)/
numHoles)*transl(amplitudToro*(i-0.1)/numHoles,0,0)*transl(-
tableDim(1)+radiusTube,-radiusExtTorus-toolLenght,0)*PoseTool;
    %position end grooving
    Grooving_Poses(:,:,i*4+3) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*(i+0.1)/
numHoles)*transl(amplitudToro*(i+0.1)/numHoles,0,0)*transl(-
tableDim(1)+radiusTube,-radiusExtTorus-toolLenght,0)*PoseTool;
    %position after start grooving
    Grooving_Poses(:,:,i*4+4) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*(i+0.1)/
numHoles)*transl(amplitudToro*(i+0.1)/numHoles,0,0)*transl(-
tableDim(1)+radiusTube,-radiusExtTorus-toolLenght-0.1,0)*PoseTool;
    %position of the groove
    Groove_Plot(:,:,i+1) = FPB*trotx((-
pi/numHoles)*0.5)*transl((amplitudToro/
numHoles)*0.5,0,0)*trotx(tableAlfa)*trotx(-pi*i/
numHoles)*transl(amplitudToro*i/numHoles,0,0)*transl(-
tableDim(1)+radiusTube,-radiusExtTorus,0)*PoseTool;


    HG=trplot(Groove_Plot(:,:,i+1), ... % Plot frame PB at the origin
        'frame', 'G', ...
        'color', 'R',...
        'text_opts', {'FontSize', 10, 'FontWeight', 'bold'},...
        'length', 0.2,...
        'arrow',...
        'width', 0.35);
    hold on;
    axis([-0.5 1.5 -0.5 2 -0.5 2.5]);



    axis 'equal';
    hold on
end
axis([-0.5 1.5 -0.5 2 -0.5 2.5]);
axis 'equal';
```



c) The reference frames for all welding points, such that z-axis of the tool is orthogonal to the surface of the torus and the x-axis is in the direction of spiral trajectory. Draw in scale the frames

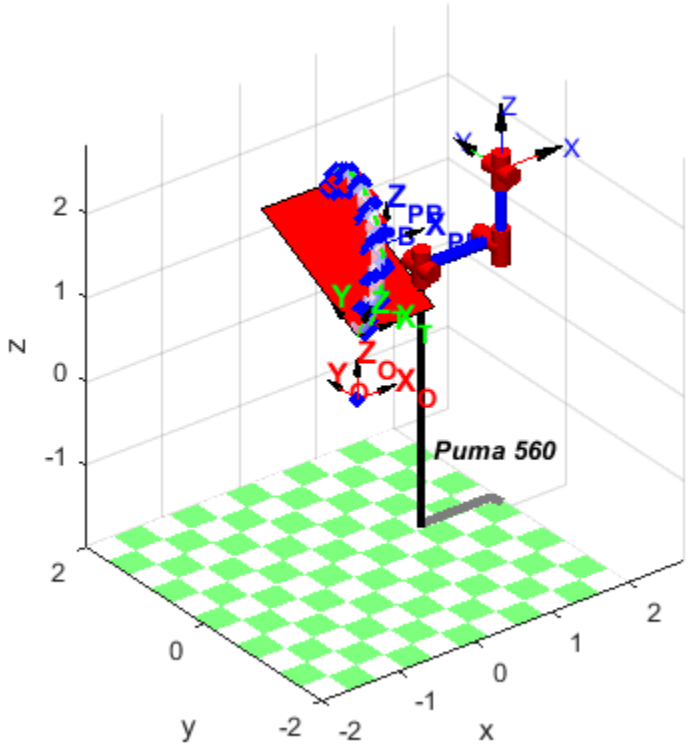
```
weldingFig = openfig('SketchFig.fig');

k=0;
for i=0:numTurns-1
    PoseTool =
 trotx(pi/2)*troty(pi/2)*trotx(pi/10)*trotz(pi/2)*trotx(pi/2);
    for j=0:WeldPointPerTurn-1
        INI = FPB*trotx(-pi*i/numHoles - pi*j/
(numHoles*WeldPointPerTurn) + tableAlfa)*transl((amplitudToro*i/
numHoles) + (amplitudToro*j/(numHoles*WeldPointPerTurn)),0,0)*transl(-
tableDim(1)+radiusTube,-radiusExtTorus+radiusTube,0)*PoseTool;
        Welding_points(:,:,(i+1)*WeldPointPerTurn+(j+1))
 = INI*trotx(2*pi*j/WeldPointPerTurn+pi/2)*transl(0,-
radiusTube,0)*trotx(-pi/2);
        if (j<=4)
                Welding_pointsRobo(:,:,(k*3+1)) = INI*trotx(2*pi*j/
WeldPointPerTurn+pi/2)*transl(0,-radiusTube-toolLenght-0.1,0)*trotx(-
pi/2);
                Welding_pointsRobo(:,:,(k*3+2)) = INI*trotx(2*pi*j/
WeldPointPerTurn+pi/2)*transl(0,-radiusTube-toolLenght,0)*trotx(-
pi/2);
                Welding_pointsRobo(:,:,(k*3+3)) =
 Welding_pointsRobo(:,:,(k*3+1));
```

```
            k= k+1;
        end

    HW=trplot(Welding_points(:,:,(i+1)*WeldPointPerTurn+(j+1)), ... %
 Plot frame PB at the origin
        'frame', 'W', ...
        'color', 'G',...
        'text_opts', {'FontSize', 10, 'FontWeight', 'bold'},...
        'length', 0.1,...
        'arrow',...
        'width', 0.1);
    hold on;
    axis([-0.5 1.5 -0.5 2.25 -0.5 2.5]);

    end
end
axis([-0.5 1.5 -0.5 2 -0.5 2.5]);
axis 'equal';

close all;
```

Robo working environment

```
roboFig = openfig('SketchFig.fig');
for i=0:numHoles-1
    Groove_Scaled = Groove/1000;
    Groove_Scaled(4,:,:) = 1;
    Groove_Scaled(2,:,:) = Groove_Scaled(2,:,:)-
min(Groove_Scaled(2,:,:));
    ma = max(Groove_Scaled(2,:));
    mi = min(Groove_Scaled(2,:));
    Groove_Scaled = Groove_Plot(:,:,i+1)*transl(+(ma-
mi)/2,0,0)*trotz(pi/2)*Groove_Scaled;

 plot3(Groove_Scaled(1,:),Groove_Scaled(2,:),Groove_Scaled(3,:),'G'); %
 plotting the Groove
end
coor_circle=transl(Drill1_Plot)';
scatter3(coor_circle(1,:),coor_circle(2,:),coor_circle(3,:),2,'r','LineWidth',5)
coor_circle=transl(Drill2_Plot)';
scatter3(coor_circle(1,:),coor_circle(2,:),coor_circle(3,:),2,'r','LineWidth',5)
coor_circle=transl(Welding_points)';
scatter3(coor_circle(1,:),coor_circle(2,:),coor_circle(3,:),2,'b','LineWidth',5);

% Q= p560.ikine6s(Drilling1_Poses, 'run');
% axis([-0.5 1.5 -0.5 2.25 -0.5 2.5]);
% p560.plot(Q)
%
% Q= p560.ikine6s(Grooving_Poses, 'run');
% axis([-0.5 1.5 -0.5 2.25 -0.5 2.5]);
% p560.plot(Q)
%
```

```
% Q= p560.ikine6s(Welding_pointsRobo, 'run');
% axis([-0.5 1.5 -0.5 2.25 -0.5 2.5]);
% p560.plot(Q)
```



Published with MATLAB® R2018a