SimpleRobotRJ

Simple Robot Model with Revolute Joint

```
myRob = SimpleRobotRJ(KinePara,DispPara);
```

Table of Contents

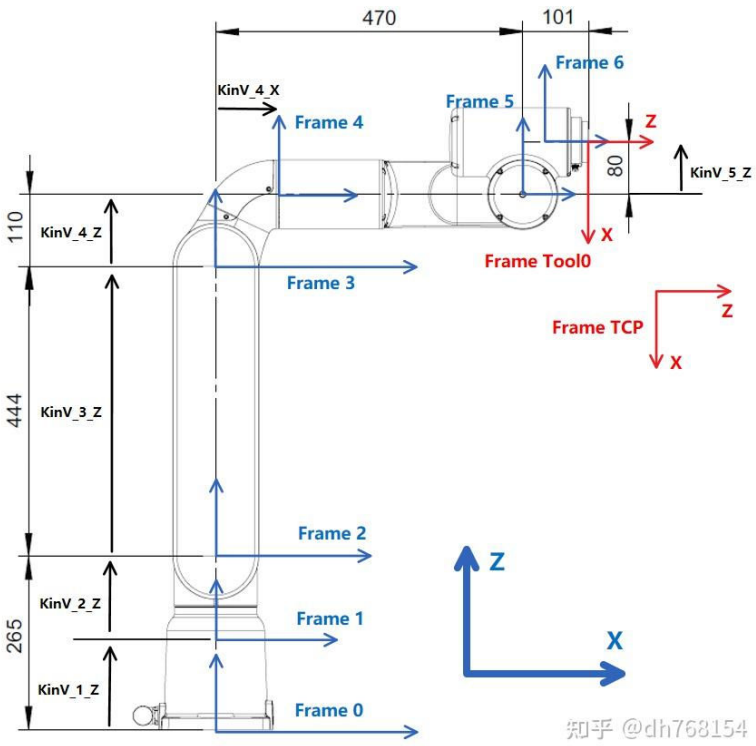
Constructor Input:

KinePara :

Kinematic Model: Set all Joint Frame Aligned with Base Frame.

- KinePara.kinv: vector from previous frame, 1st is from world to base
- KinePara.rotdirc: joint rotation axis, can be + or -
- KinePara.flange: last joint frame to tool flange frame
- KinePara.base: from world frame to base frame
- KinePara.tcp: from flange frame to tcp

Example: CRB 15000



DispPara：

- DispPara.jsize: joint is represent using circle, circle radius
- DispPara.cirpts: points on the circle
- DispPara.offorder: define how to draw the links, 2,3,1 means draw y dirction first, then z and x dirction. if robot is 6 axis robot, there are 7 links, the input is a 7*3 matrix.

Already defined 2 robot:

```
[KinePara,DispPara,PreSetting] = model_UR5(tcp,base);
[KinePara,DispPara,PreSetting] = model_GoFa(tcp,base);
```

Properties

Read and Write:

- showag: update joint angle on the title while moving, default true
- tcptrace: show joint movement tcp trace for single commend, default true
- keeptrace: keep all tcp trace until set keeptrace to false, default false
- jrange: joint limit, default +-pi

- home: home pose, default all 0
- floorH: floor position, default 0
- frame0Size: base frame size, if not defined, will not show this frame
- frametSize: tcp frame size, if not defined, will not show this frame
- RobName

Read Only:

- cjoint: current joint value
- cpose: current tcp pose, 4*4 matrix

Methods

ShowRobot

Plot Robot

```
myRob.ShowRobot;

ag = [0,0,0,0,pi/6,0];
myRob.ShowRobot(ag); % add initial pose
% ag is optional input, if not specify, will be all 0
```

MoveAbsJ

If size of Input angle is 1*n, the robot will move from current pose to the input angle with max angle step of [1,1,1,5,5,5] deg. If size of Input angle > 1, robot will follow the input angle (no interpulation).

Move from Current Pose to Specific Pose:

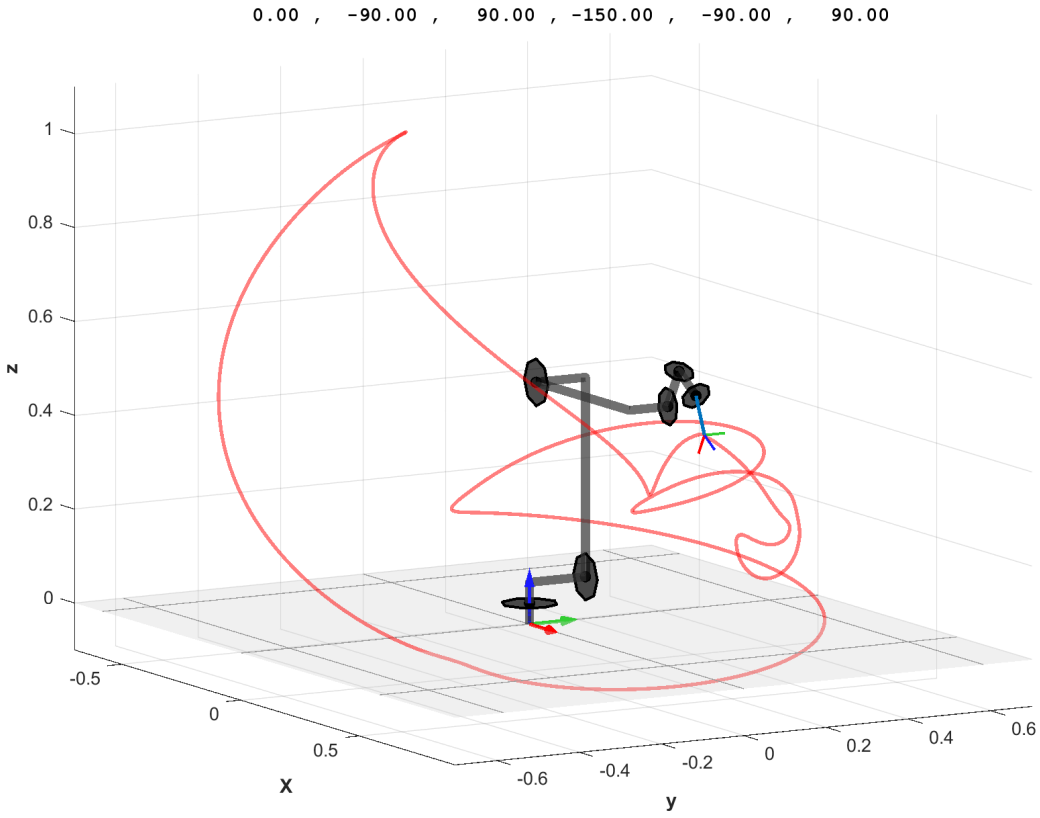
```
myRob.MoveAbsJ([10,20,30,40,50,60]/180*pi);
```

Specify Multiple Poses

```
myStation.MoveAbsJ(Muti_Angles);
```

Other Optional input for debugging:

- MotionType: 'continous' or 'jump', default is 'continous'. If size of Input angle is 1*n, 'jump' means no interpulation between current pose to input pose.
- TimePause: time pause between each movement step
- AngleStep: default is [1,1,1,5,5,5] deg, used for angle interpulation if input angle is 1*n



GoHome / GoZero

Go home pose and zero pose

```
myRob.GoHome;
myRob.GoZero;
```

fkine

Calculate forward kinematics, output is a 4*4*n matrix, each frame ref to world frame.

First layer is from world to base, last layer is flange to tcp.

```
Tw = myRob.fkine; % fkine for current pose
Tw = myRob.fkine([10,20,30,40,50,60]/180*pi); % Specify joint angle
```

get_jacob0 / get_jacobb

jacobian and body jacobian.

```
J0 = myRob.get_jacob0; % for current pose
J0 = myRob.get_jacob0([10,20,30,40,50,60]/180*pi);

Jb = myRob.get_jacobbb;
Jb = myRob.get_jacob0([10,20,30,40,50,60]/180*pi);
```

manipulability

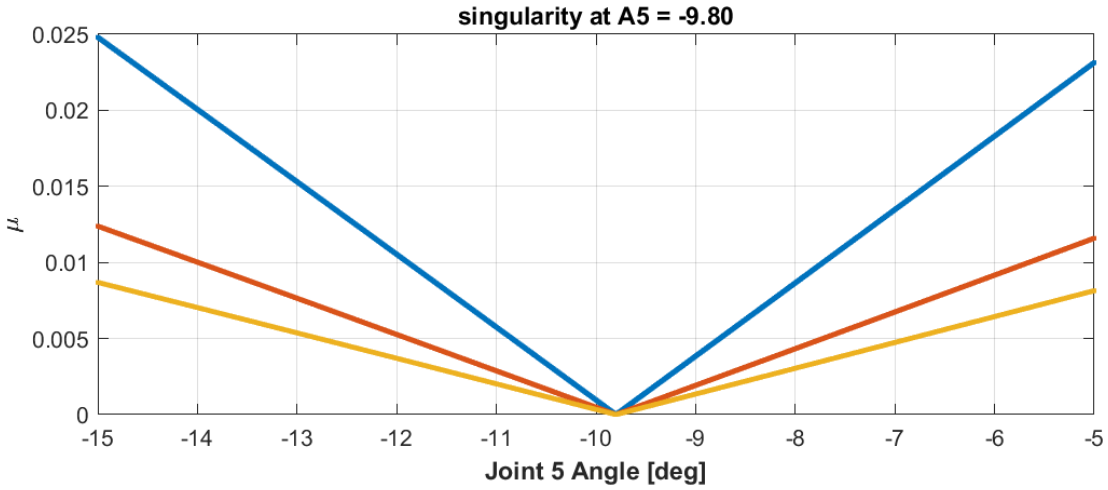
Check singularity, close to 0 means close to singularity.

```
n = 1000;
ag5 = linspace(-15,-5,n)/180*pi;
ags = zeros(n,6);
ags(:,5) = ag5;
mu = NaN(n,3);
for i = 1:n
    mu(i,:) = myRob.manipulability(ags(i,:));
end
```

Output 1*3 vector, 3 methods:

1. smallest singular value
2. min singular value / max singular value
3. abs(det(jacobian))

example: GoFa A5 Singularity Check



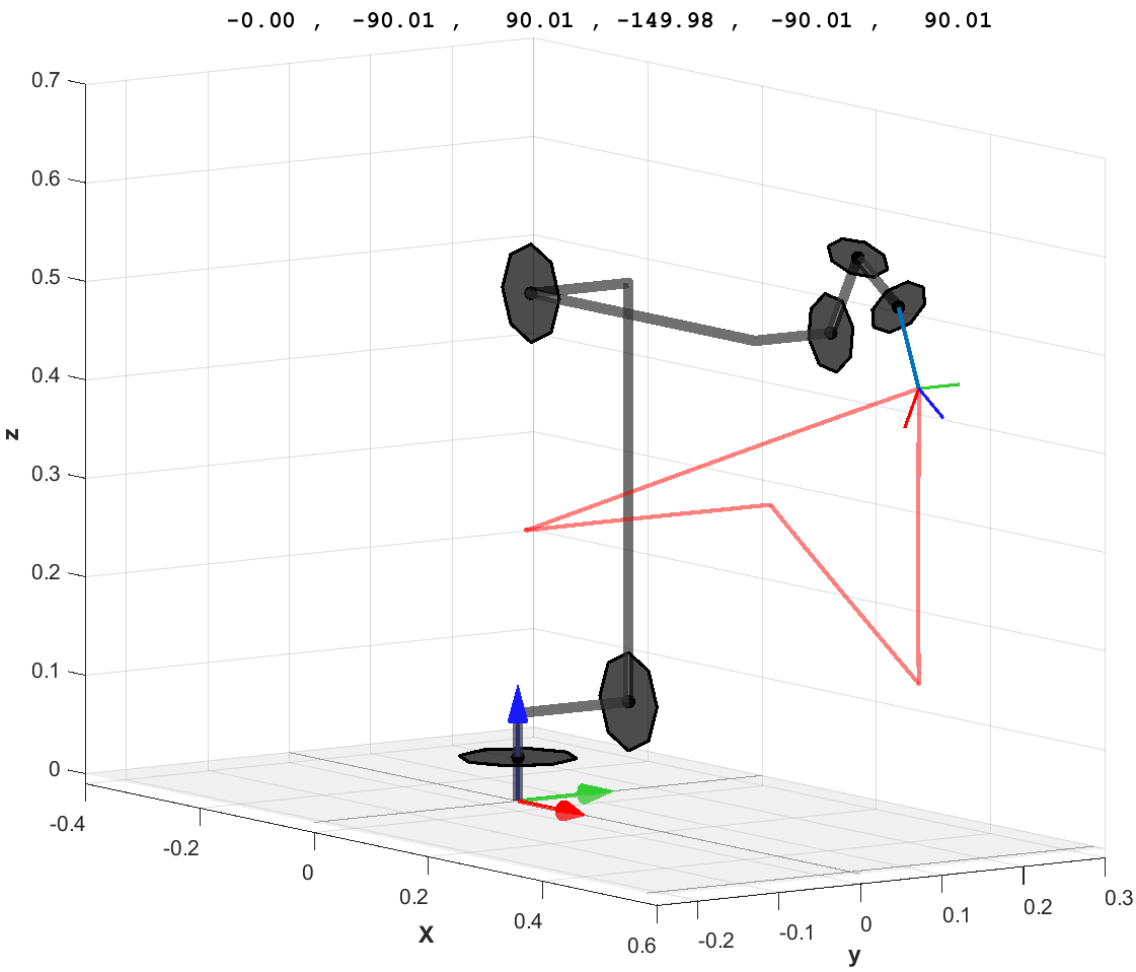
RRMove

linear movement using simple resolved rate control.

```
myRob.RRMove(T_goal);
```

Optional input:

- 'max_step'：max rotate angle for each joint, default 1 deg for 1st 3 joint, others 2 deg, unit is rad
- 'maxloop'：default 1000, incase singularity, never reach goal
- 'k'：learning rate，default 0.1
- 'errlim'：break loop criteria, [distance,angle], default [1/10/1000,0.1/180*pi])
- 'transpose'：use jacobian transpose instead inverse，default false
- 'showframe'：show initial and goal pose, default false
- 'warnmaxloop'：if reach max loop, show warning, default true
- 'flexwrist' : position only, to cross the singularity



Offs

Keep orientation, translation only.

```
myRob.Offs([x,y,z]); % ref to world frame (default)
myRob.Offs([x,y,z],'world'); % ref to world frame
myRob.Offs([x,y,z],'tool');  % ref to tool frame
```

Optional input:

Other optional input is same as RRMove

Rots

Keep position, rotation only.

```
myRob.Rots(theta,'z'); % rotation about z axis, ref to tool frame (default)
myRob.Rots(theta,'y','world'); % ref to world frame
```

Optional input:

Other optional input is same as RRMove

'flexwrist' is not a valid input in this function

get_cpose

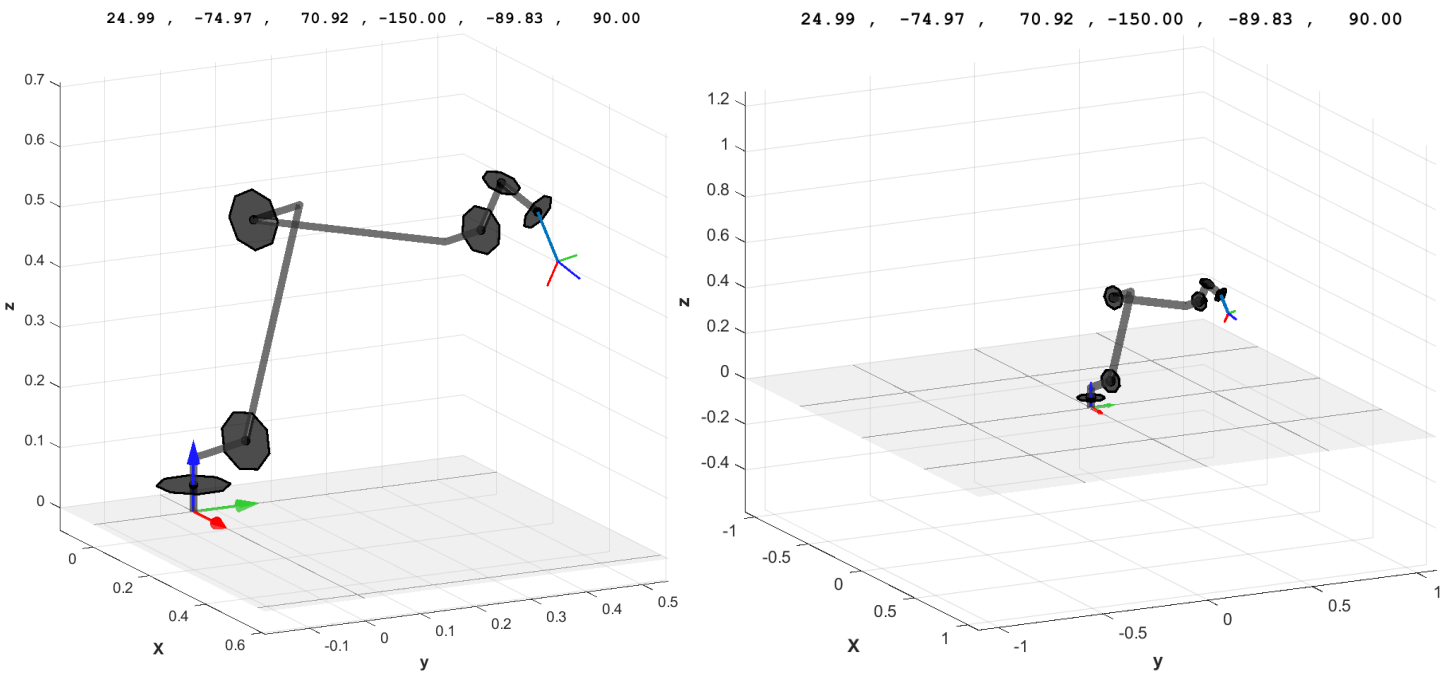
Get tcp pose. if want to get current tcp pose, use obj.cpose is the same.

```
T_tcp = myRob.get_cpose; % tcp frame (default)
T_tcp = myRob.get_cpose('tcp'); % flange frame
T_tcp = myRob.get_cpose('tool0'); % flange frame
```

zoomin / zoomout

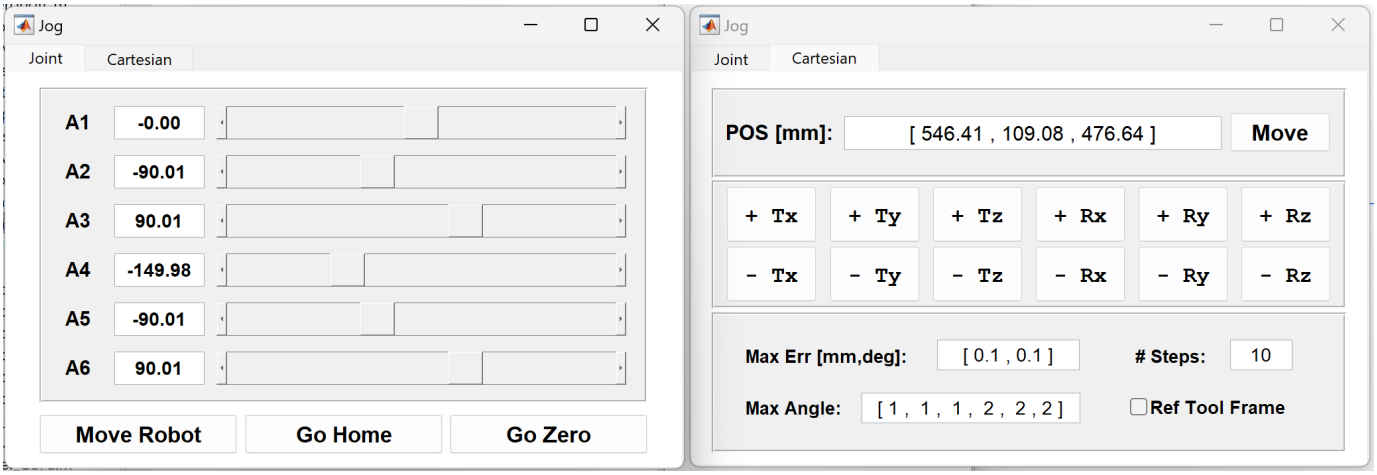
Zoom in/out the figure

```
myRob.zoomin;
myRob.zoomout;
```



Jog

Teach Pendant



manipulability

Manipulability Measure.

```
[mu,v] = myRob.manipulability; % for current pose
[mu,v] = myRob.manipulability(ag); % for specific pose
```

Output:

- mu: [min(sig) , min(sig)/max(sig) , abs(det(J))]
- v: 3*2 matrix, used to check which degree of freedom is limited, [trans,rot]

get_ellipse

get current manipulability ellipsoid plotting data. Use surf function to plot.

```
[xx,yy,zz] = myRob.get_ellipse;
```

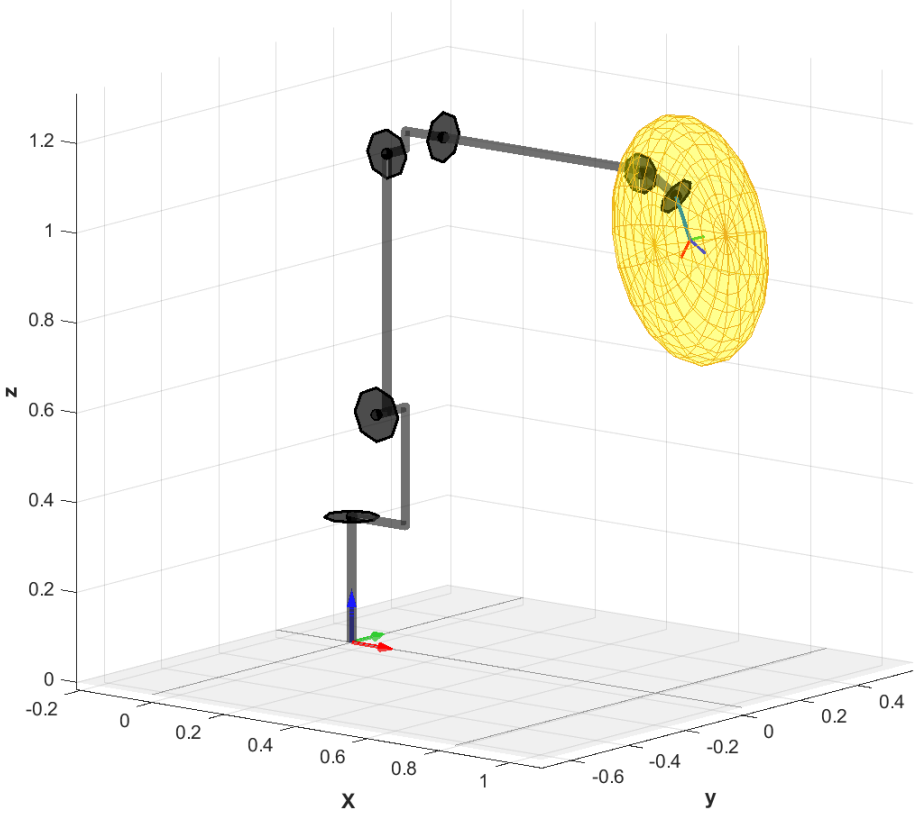
Optional Input:

- **'n'** : ellipsoild resolution

- **'scale'** : scale factor of the ellipsoild
- **'type'** : 'pos', 'orit', 'force' or 'trq'

```
surf(xx,yy,zz,'EdgeColor',[0.9290 0.6940 0.1250],...
    'FaceColor','y','FaceAlpha',0.25,'AlphaData',0.5); hold on
```

```
    -0.00 ,    0.00 ,    0.00 ,    0.00 ,   30.00 ,    0.00
```



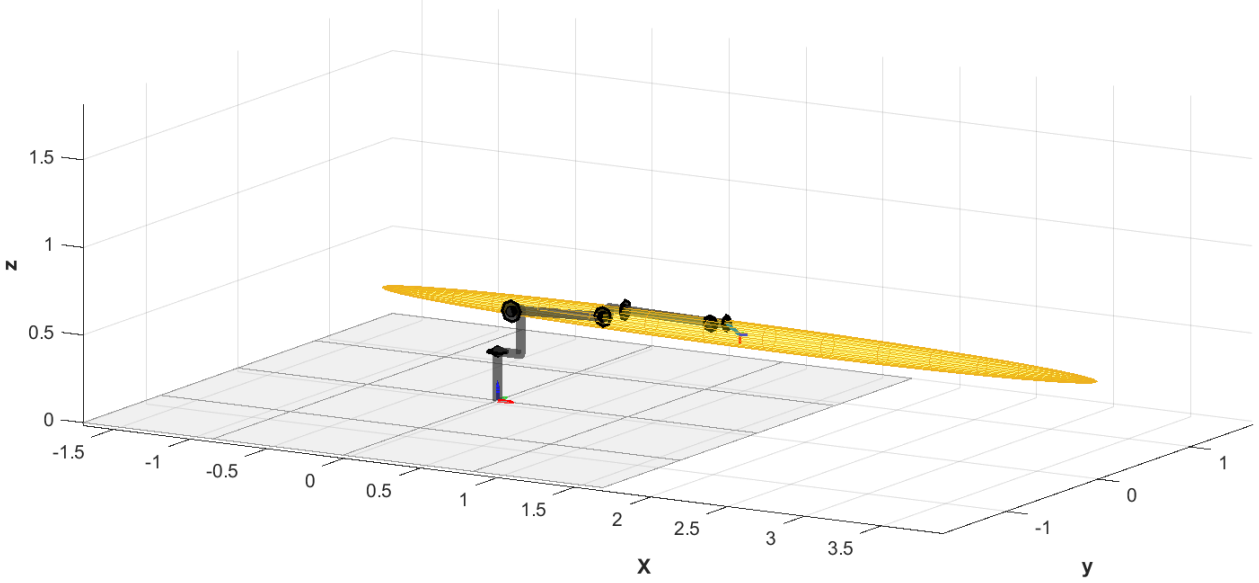
This method is also integrated in to the model, set properties to display:

```
myRob.elipSize = 0.1; % size factor, Optional
myRob.elipN = 20; % ellipsoild resolution
myRob.elipType = 'pos'; % 'pos', 'orit', 'force' or 'trq
myRob.showellip = true; % distplay ellipsoild
```

The ellipsoild will move when jogging:

force ellipsoild when close to singularity:

```
-0.00 ,   87.46 ,  -86.47 ,    0.00 ,   -5.12 ,    0.00
```



lwl

In memorial of Dr. Li Wenliang, who sounded the alarm on COVID-19

```
myRob.lwl;
```