IRB140 Workspace exercise

Javier Cabrera

Link: https://drive.matlab.com/sharing/0a86102f-bff8-487a-becf-cec9e27c6380

Make the same exercise for the irb140 manipulator and plot the working area as shown in the figure

Type of motion	Range of movement
Axis 1: Rotation motion	+180° to - 80°
Axis 2: Arm motion	+110° to -90°
Axis 3: Arm motion	+50° to -230°
Axis 4: Wrist motion	+200° to +200° Default +165 revolutions to -165 revolutions Max**)
Axis 5: Bend motion	+120° to -120°
Axis 6: Turn motion	+400° to -400° Default +163 revolutions to -163 revolutions Max**)

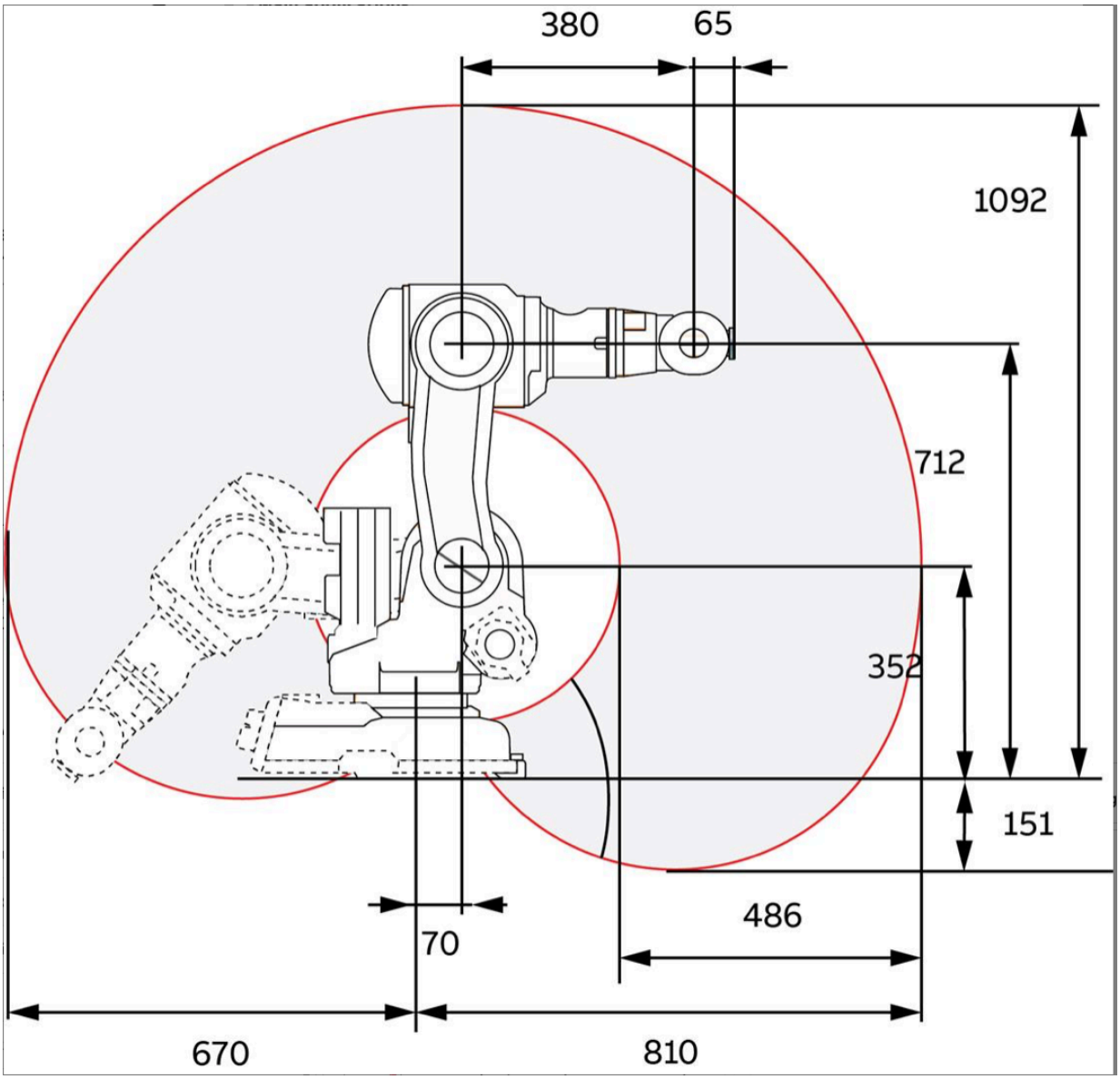


Table of Contents

Invoque IRB140

```
clear;
close all;
mdl_irb140;
```

```
robot =

IRB 140 [ABB]:: 6 axis, RRRRRR, stdDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|      0.352|       0.07|    -1.5708|          0|
|  2|         q2|          0|       0.36|          0|          0|
|  3|         q3|          0|          0|     1.5708|          0|
|  4|         q4|       0.38|          0|    -1.5708|          0|
|  5|         q5|          0|          0|     1.5708|          0|
|  6|         q6|          0|          0|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+
```

Calculate the limits

```
degree_limits = [NaN NaN;-180 20; -50 230] % Based on table ranges and having into acco
```

```
degree_limits = 3×2
   NaN   NaN
  -180    20
   -50   230
```

```
rad_limits = deg2rad(degree_limits)
```

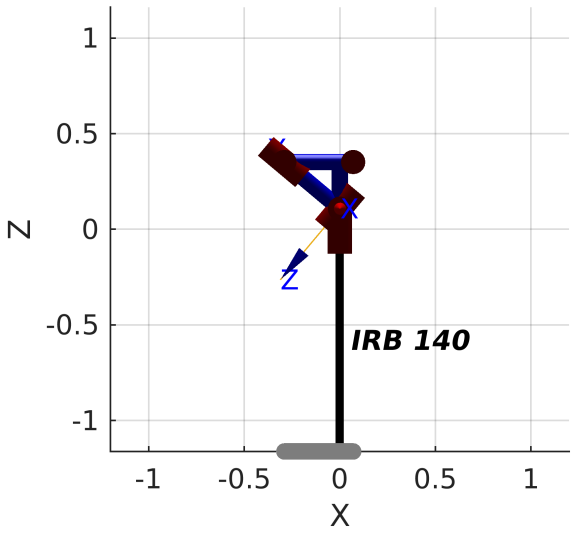
```
rad_limits = 3×2
       NaN       NaN
   -3.1416    0.3491
   -0.8727    4.0143
```

```
for x = 2:3 % We only need the 2nd and 3rd axis limits
    irb140.links(x+1).qlim = rad_limits(x,:);
end
```

Initial approach: Trailing workspace while rotating both axes at same time

Plot the IRB initial position

```
initial_position = [qr(1) rad_limits(2,1) rad_limits(3,1) qr(4:6)];
irb140.plot(initial_position)
view(0,0)
```



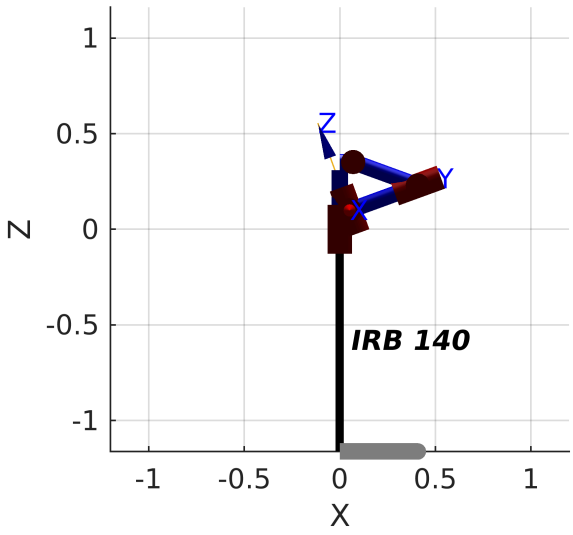
Build the movement matrix

```
iterations = 200;

q1 = zeros(iterations,1) + qr(1);
q2 = linspace(rad_limits(2,1),rad_limits(2,2),iterations)';
q3 = linspace(rad_limits(3,1),rad_limits(3,2),iterations)';
q4_6 = zeros(iterations,3) + qr(4:6);

Q = [q1 q2 q3 q4_6]
```

```
Q = 200×6
         0   -3.1416   -0.8727         0    1.5708   -1.5708
         0   -3.1241   -0.8481         0    1.5708   -1.5708
         0   -3.1065   -0.8235         0    1.5708   -1.5708
         0   -3.0890   -0.7990         0    1.5708   -1.5708
         0   -3.0714   -0.7744         0    1.5708   -1.5708
         0   -3.0539   -0.7499         0    1.5708   -1.5708
         0   -3.0363   -0.7253         0    1.5708   -1.5708
         0   -3.0188   -0.7008         0    1.5708   -1.5708
         0   -3.0013   -0.6762         0    1.5708   -1.5708
         0   -2.9837   -0.6516         0    1.5708   -1.5708
    ⋮
```

Plot the movement

```
irb140.plot(Q,'trail','--','jaxes','zoom',2)
```

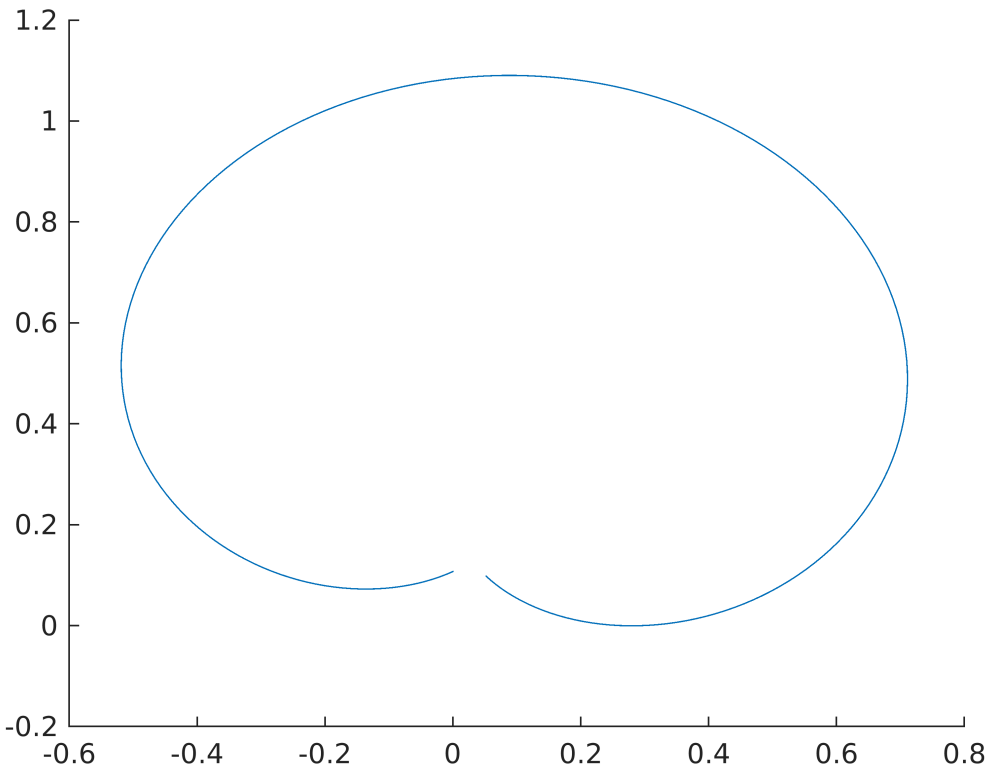


Plot the trail

```
T = irb140.fkine(Q);
ft = [T.t]
```

```
ft = 3×200
    0.0011   -0.0094   -0.0203   -0.0315   -0.0431   -0.0550   -0.0672   -0.0797 ···
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.1077    0.1020    0.0968    0.0920    0.0876    0.0838    0.0806    0.0778
```

```
figure
plot3(ft(1,:),ft(2,:), ft(3,:))
view(0,0);
```



The result is very similar to the objective, but it isn't precise. This is because, excluding the rotation edges of q2, the other points could have q3 in its maximum length by having the arm straight.

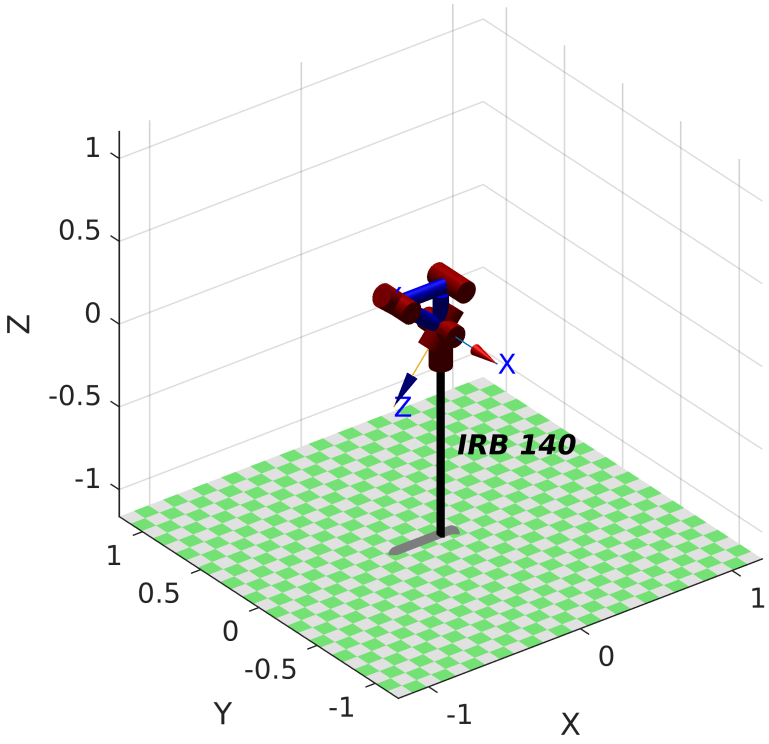
Second approach: Trailing full rotation of q3 at limits of q2

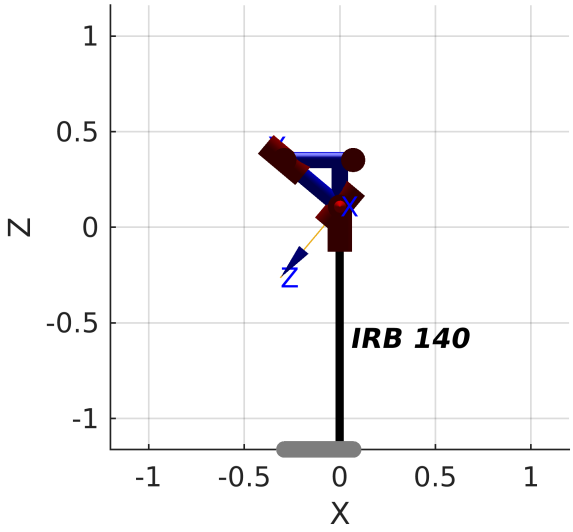
The workspace will be calculated in 3 parts, being the 2 first ones the edges of rotation of q2 having the full rotation of q3 and the third being q3 fixed in a straight position and having q2 rotate.

Part 1 - With q2 fixed at slowest angle

Plot the IRB initial position

```
initial_position = [qr(1) rad_limits(2,1) rad_limits(3,1) qr(4:6)];
figure
irb140.plot(initial_position);
```


1
0.5
0
-0.5
-1
Z
X
Z
IRB 140
1
0.5
0
-0.5
-1
Y
1
0
-1
X


1
0.5
0
-0.5
-1
Z
X
Z
IRB 140
-1
-0.5
0
0.5
1
X

Build the movement matrix

```
iterations = 200;

q1 = zeros(iterations,1) + qr(1);
q2 = zeros(iterations,1) + (rad_limits(2,1));
q3 = linspace(rad_limits(3,1),rad_limits(3,2),iterations)';
q4_6 = zeros(iterations,3) + qr(4:6);

Q = [q1 q2 q3 q4_6]
```

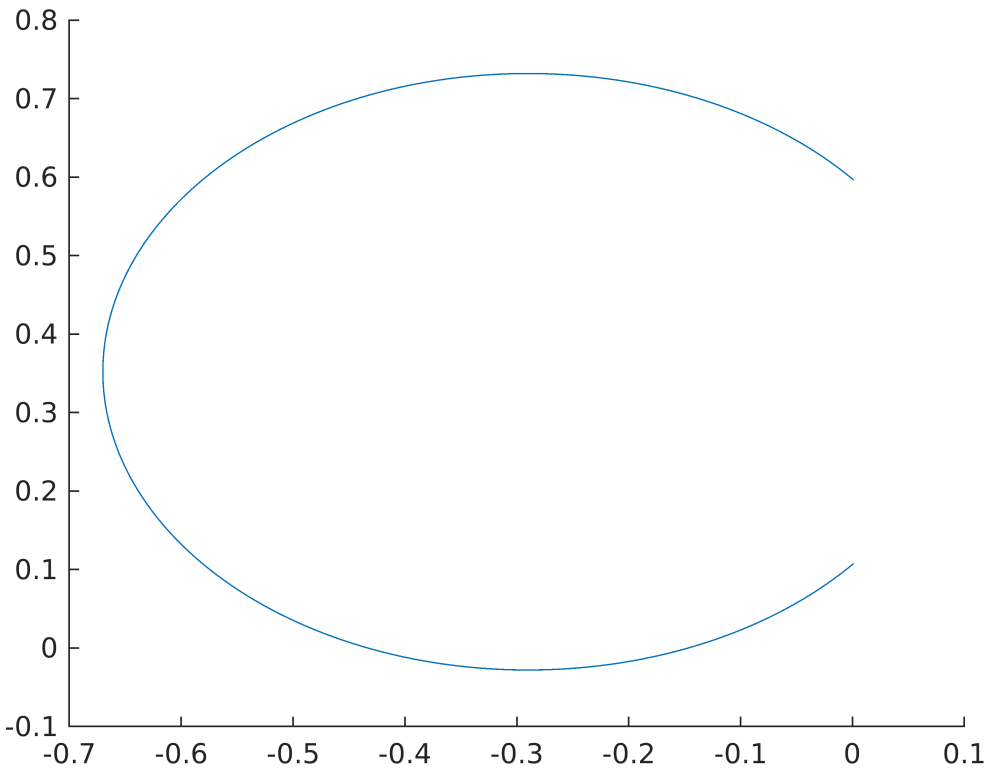
```
Q = 200×6
         0   -3.1416   -0.8727         0    1.5708   -1.5708
         0   -3.1416   -0.8481         0    1.5708   -1.5708
         0   -3.1416   -0.8235         0    1.5708   -1.5708
         0   -3.1416   -0.7990         0    1.5708   -1.5708
         0   -3.1416   -0.7744         0    1.5708   -1.5708
         0   -3.1416   -0.7499         0    1.5708   -1.5708
         0   -3.1416   -0.7253         0    1.5708   -1.5708
         0   -3.1416   -0.7008         0    1.5708   -1.5708
         0   -3.1416   -0.6762         0    1.5708   -1.5708
         0   -3.1416   -0.6516         0    1.5708   -1.5708
    ⋮
```

Plot the trail

```
T = irb140.fkine(Q);
ft1 = [T.t]
```

```
ft1 = 3×200
    0.0011   -0.0050   -0.0112   -0.0177   -0.0243   -0.0310   -0.0379   -0.0450 ···
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.1077    0.1007    0.0937    0.0870    0.0804    0.0739    0.0677    0.0615
```

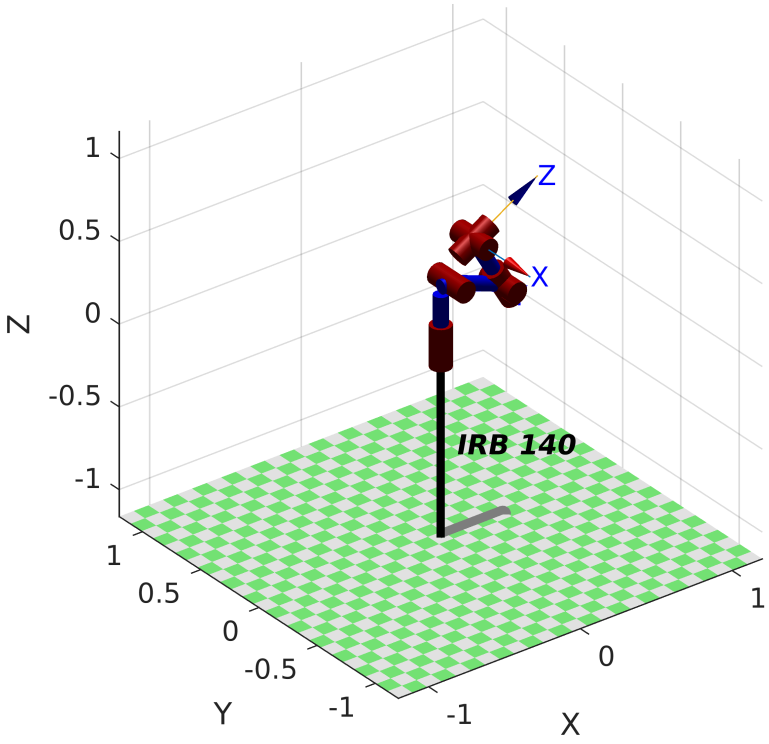
```
figure
plot3(ft1(1,:),ft1(2,:), ft1(3,:))
view(0,0);
```

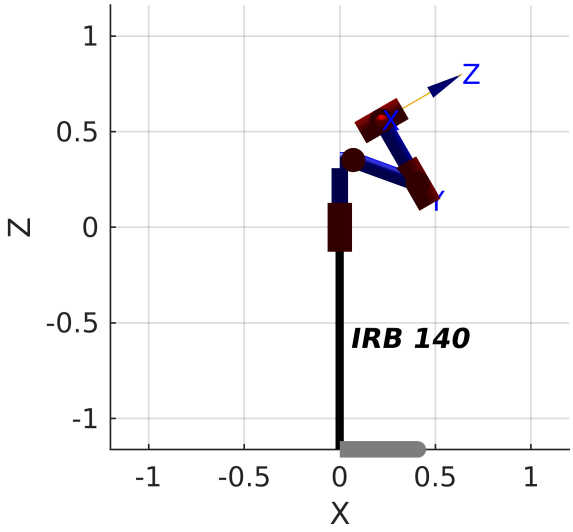


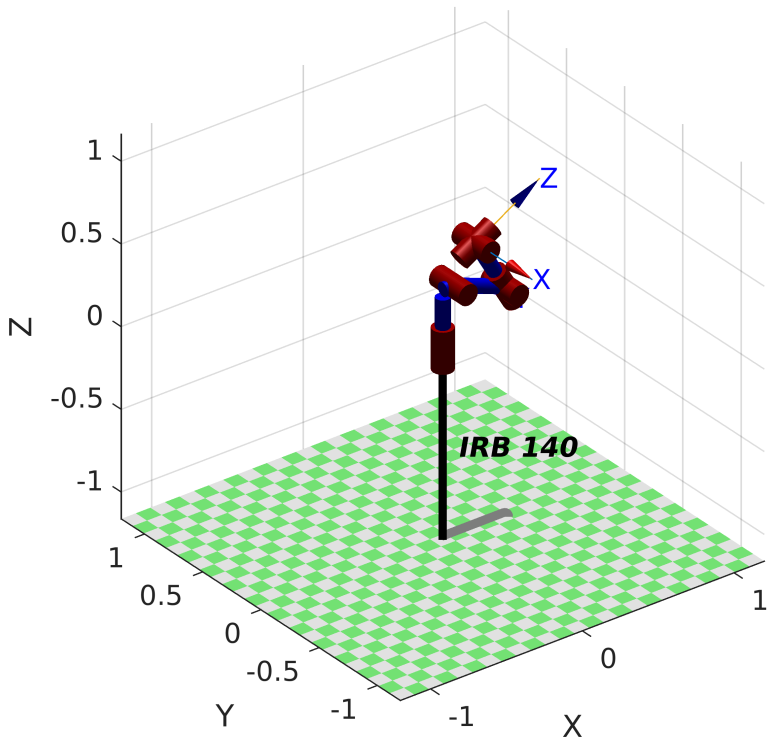
Part 2 - With q2 fixed at slowest angle

Plot the IRB initial position

```
initial_position = [qr(1) rad_limits(2,2) rad_limits(3,1) qr(4:6)];
figure;
irb140.plot(initial_position);
```


Z
X
IRB 140
1
0.5
0
-0.5
-1
Z
Y
X


Z
X
Y
IRB 140
1
0.5
0
-0.5
-1
Z
X



Build the movement matrix

```
iterations = 200;

q1 = zeros(iterations,1) + qr(1);
q2 = zeros(iterations,1) + (rad_limits(2,2));
q3 = linspace(rad_limits(3,1),rad_limits(3,2),iterations)';
q4_6 = zeros(iterations,3) + qr(4:6);

Q = [q1 q2 q3 q4_6]
```

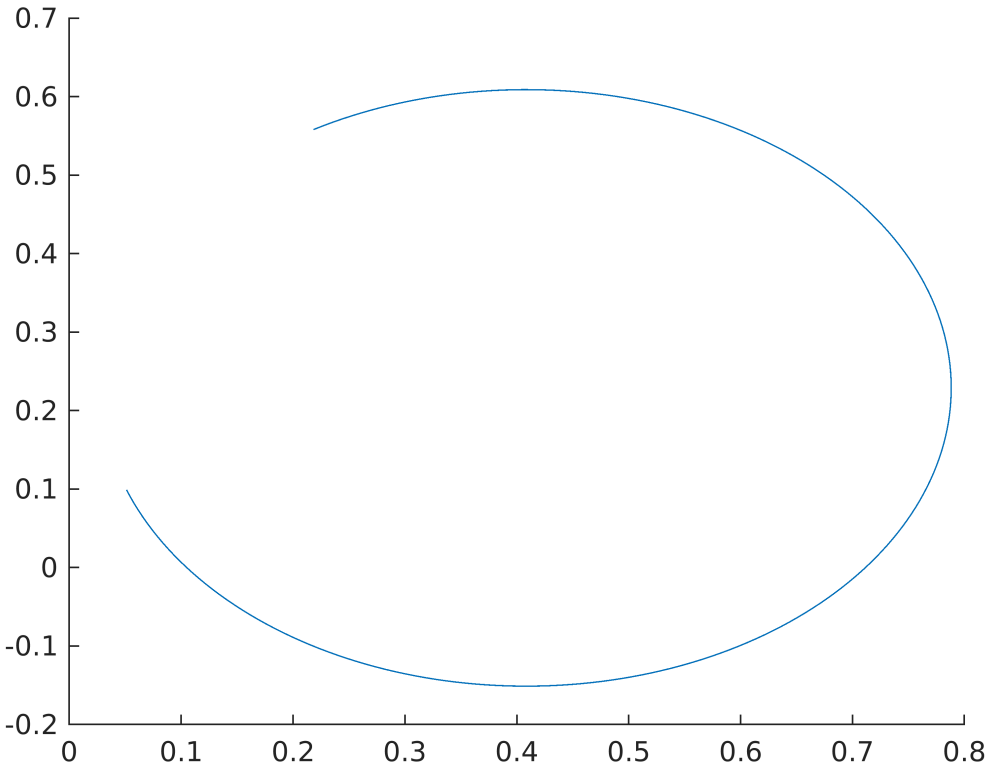
```
Q = 200×6
         0    0.3491   -0.8727         0    1.5708   -1.5708
         0    0.3491   -0.8481         0    1.5708   -1.5708
         0    0.3491   -0.8235         0    1.5708   -1.5708
         0    0.3491   -0.7990         0    1.5708   -1.5708
         0    0.3491   -0.7744         0    1.5708   -1.5708
         0    0.3491   -0.7499         0    1.5708   -1.5708
         0    0.3491   -0.7253         0    1.5708   -1.5708
         0    0.3491   -0.7008         0    1.5708   -1.5708
         0    0.3491   -0.6762         0    1.5708   -1.5708
         0    0.3491   -0.6516         0    1.5708   -1.5708
    ⋮
```

Plot the trail

```
T = irb140.fkine(Q);
ft2 = [T.t]
```

```
ft2 = 3×200
    0.2183    0.2264    0.2347    0.2430    0.2515    0.2600    0.2687    0.2774 ···
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.5580    0.5625    0.5669    0.5711    0.5750    0.5788    0.5823    0.5856
```

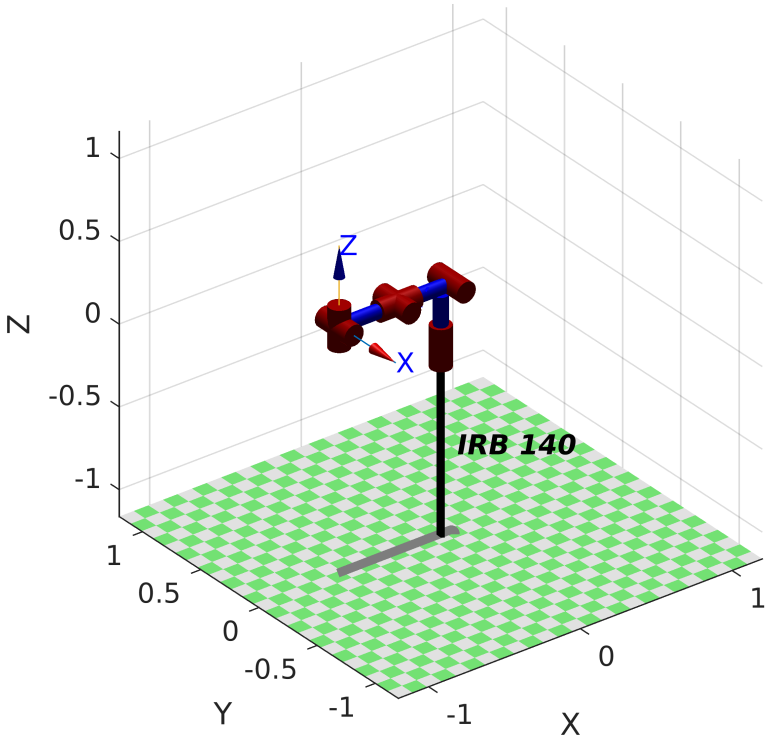
```
figure
plot3(ft2(1,:),ft2(2,:), ft2(3,:))
view(0,0);
```

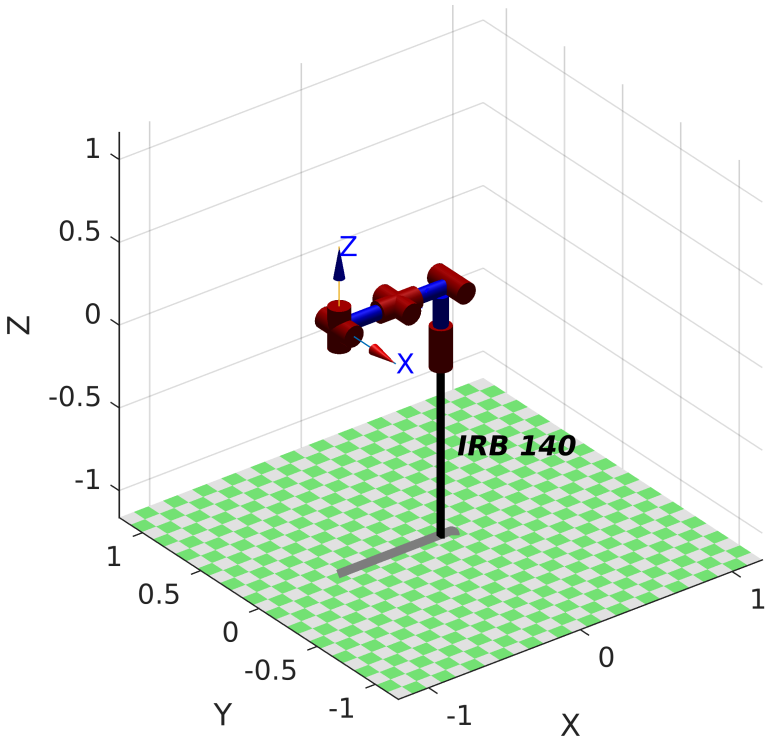


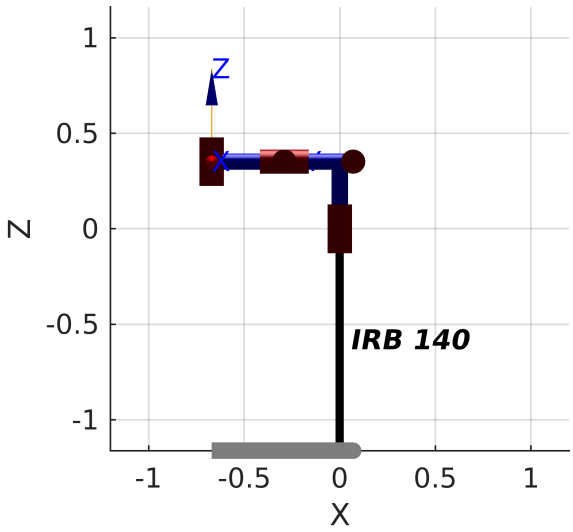
Part 3 - With q3 fixed at straight angle

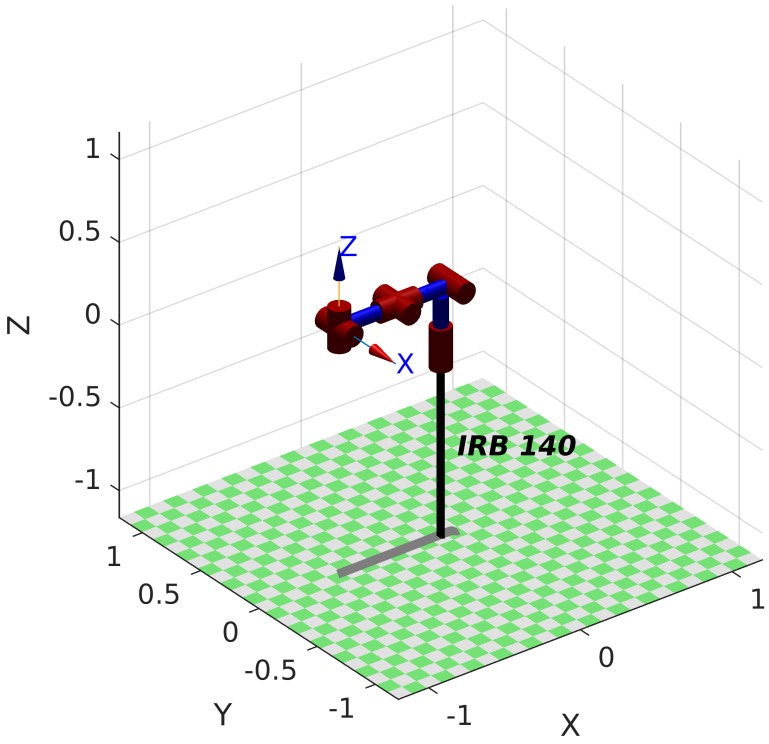
Plot the IRB initial position

```
initial_position = [qr(1) rad_limits(2,1) qr(3:6)];
figure
irb140.plot(initial_position);
```


Z
X
IRB 140
1
0.5
0
-0.5
-1
Z
Y
X


Z
X
IRB 140
1
0.5
0
-0.5
-1
Z
Y
X





Build the movement matrix

```
iterations = 200;

q1 = zeros(iterations,1) + qr(1);
q2 = linspace(rad_limits(2,1),rad_limits(2,2),iterations)';
q3_6 = zeros(iterations,4) + qr(3:6);

Q = [q1 q2 q3_6]
```

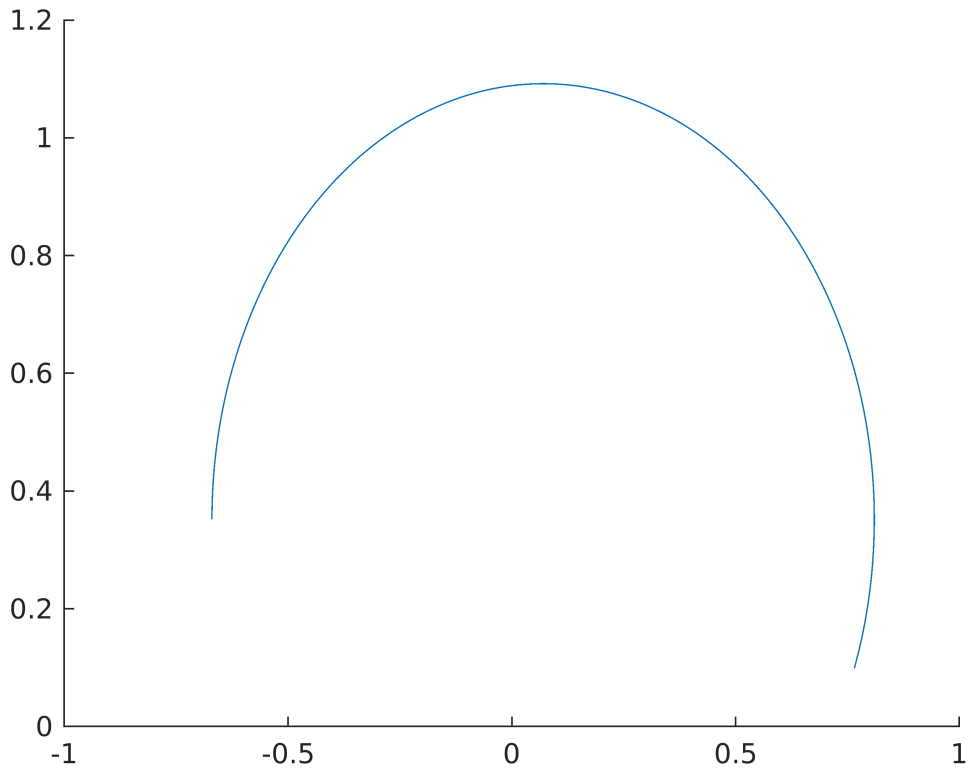
```
Q = 200×6
         0   -3.1416    1.5708         0    1.5708   -1.5708
         0   -3.1241    1.5708         0    1.5708   -1.5708
         0   -3.1065    1.5708         0    1.5708   -1.5708
         0   -3.0890    1.5708         0    1.5708   -1.5708
         0   -3.0714    1.5708         0    1.5708   -1.5708
         0   -3.0539    1.5708         0    1.5708   -1.5708
         0   -3.0363    1.5708         0    1.5708   -1.5708
         0   -3.0188    1.5708         0    1.5708   -1.5708
         0   -3.0013    1.5708         0    1.5708   -1.5708
         0   -2.9837    1.5708         0    1.5708   -1.5708
    ⋮
```

Plot the trail

```
T = irb140.fkine(Q);
ft3 = [T.t]
```

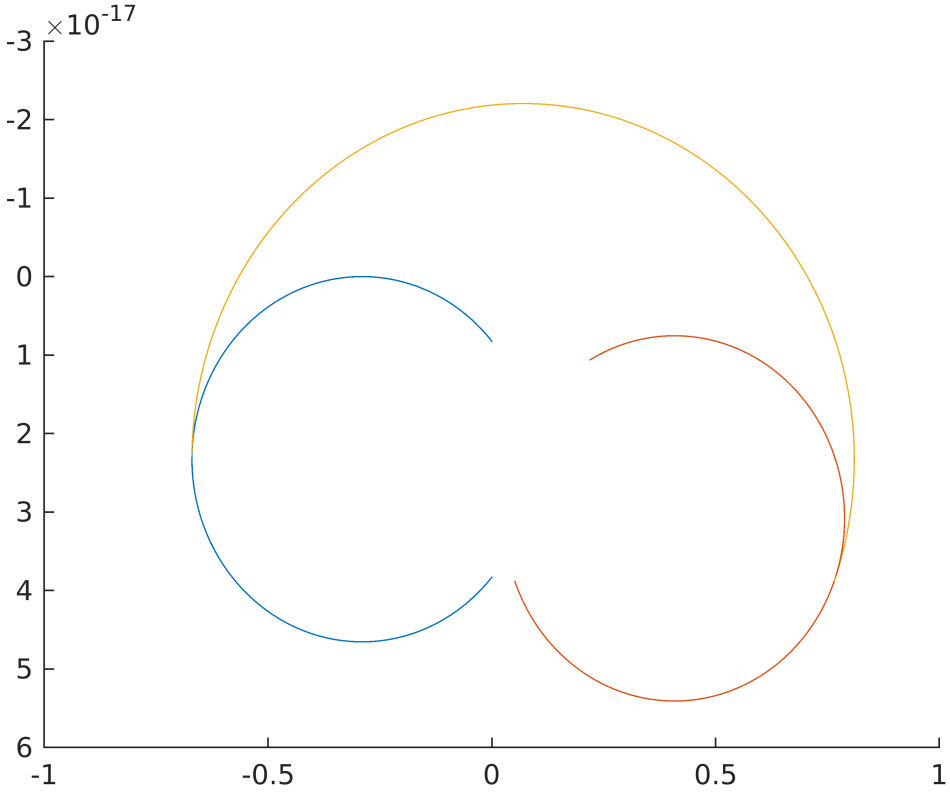
```
ft3 = 3×200
   -0.6700   -0.6699   -0.6695   -0.6690   -0.6682   -0.6672   -0.6659   -0.6644 ···
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.3520    0.3650    0.3780    0.3909    0.4039    0.4168    0.4297    0.4426
```

```
figure
plot3(ft3(1,:),ft3(2,:), ft3(3,:))
view(0,0);
```



Plot the combined trail

```
figure
hold on
plot3(ft1(1,:),ft1(2,:), ft1(3,:))
plot3(ft2(1,:),ft2(2,:), ft2(3,:))
plot3(ft3(1,:),ft3(2,:), ft3(3,:))
set(gca, 'YDir','reverse') % No sé porque se imprime del revés el workspace, cuando en
hold off
```


×10-17
-3
-2
-1
0
1
2
3
4
5
6
-1
-0.5
0
0.5
1