Robotics HW5

Yunfan Gao

October 2020

Problem 1

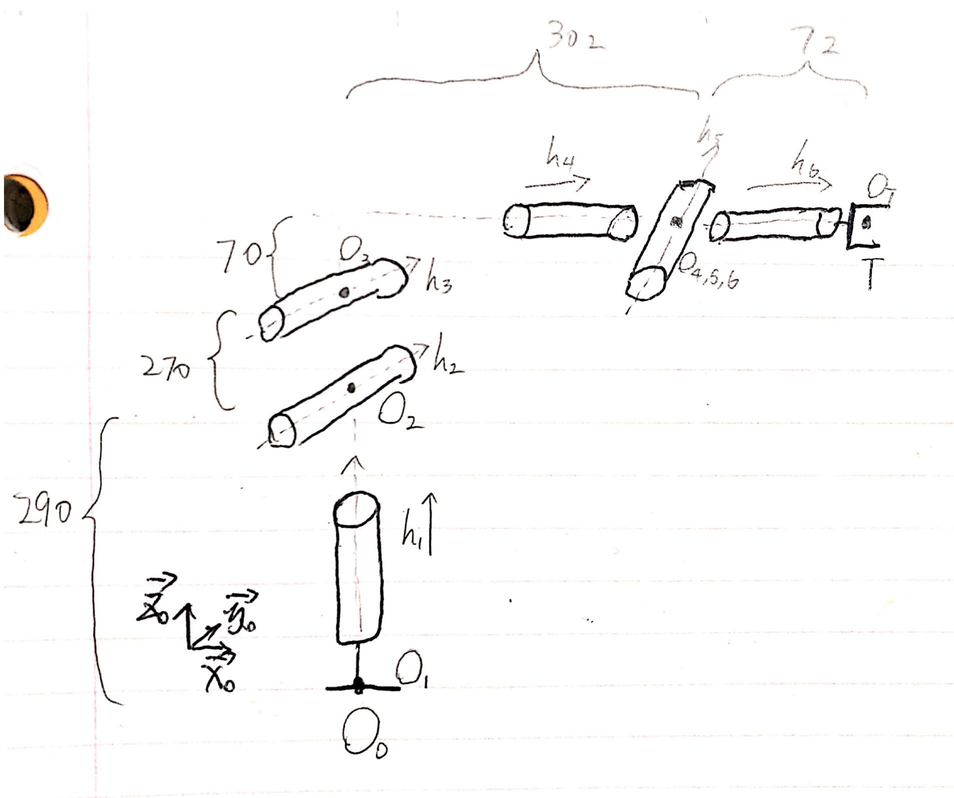

Figure 1: Product-of-Exponential diagram of ABB IRB 120

$e_x = [1; 0; 0], e_y = [0; 1; 0], e_z = [0; 0; 1], z_v = [0; 0; 0]$
As indicated above, we can find that the joint axes:
$H = [h_1 \mid h_2 \mid h_3 \mid h_4 \mid h_5 \mid h_6] = [e_z \quad e_y \quad e_y \quad e_x \quad e_y \quad e_x]$
and the link:

$P = [P_{01} \mid P_{12} \mid P_{23} \mid P_{34} \mid P_{45} \mid P_{56} \mid P_{6T}] = [z_v \quad 290e_z \quad 270e_z \quad 302e_x + 70e_z \quad 0 \quad 0 \quad 72e_x]$

For Standford arm:

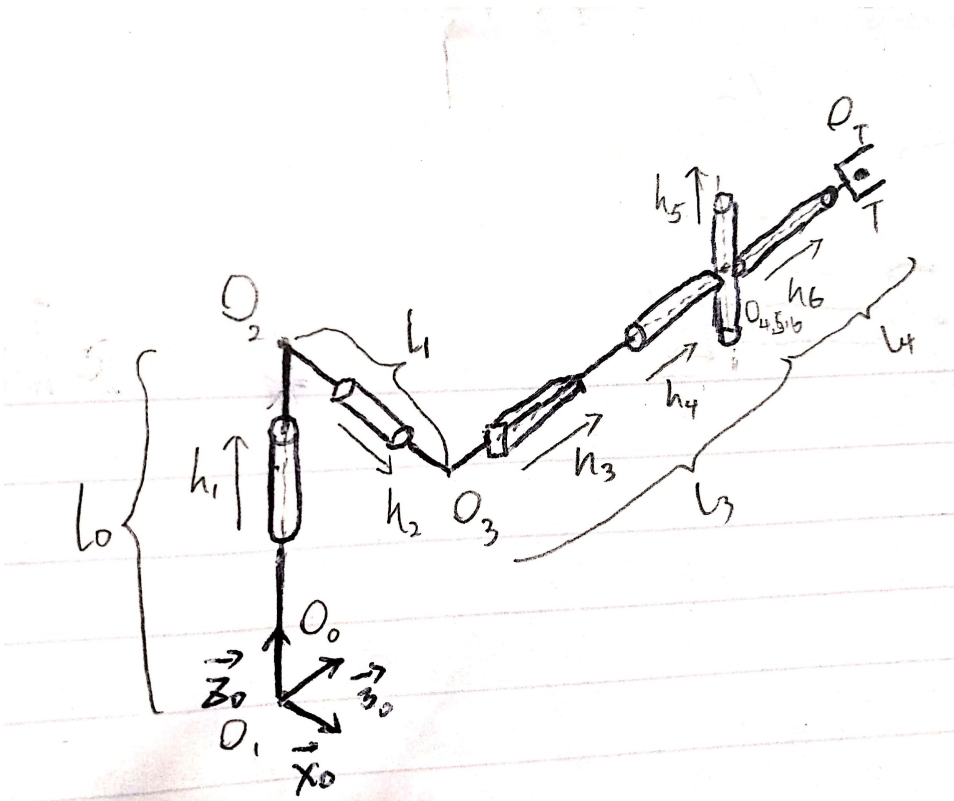

Figure 2: Product-of-Exponential diagram of Standford arm

The joint axes:
$H = [h_1 \mid h_2 \mid h_3 \mid h_4 \mid h_5 \mid h_6] = [e_z \quad e_x \quad e_y \quad e_y \quad e_z \quad e_y]$
and the link:
$P = [P_{01} \mid P_{12} \mid P_{23} \mid P_{34} \mid P_{45} \mid P_{56} \mid P_{6T}] = [z_v \quad l_0e_z \quad l_1e_x \quad l_3e_y \quad 0 \quad 0 \quad l_4e_y]$

b)

From the SDH graph, we can see $R_{6T} = [(X_T)_6 \mid (Y_T)_6 \mid (Z_T)_6] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

$p_{01} = 290z_0 + 0x_1$
$p_{12} = 0z_1 + 270x_2$

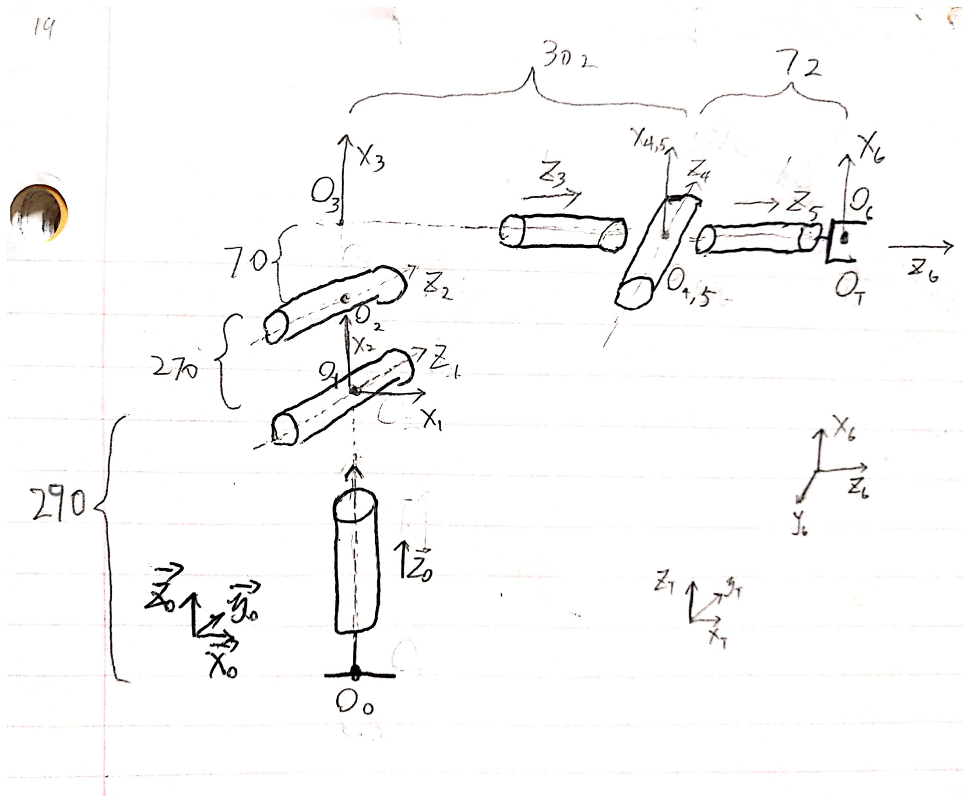

Figure 3: SDH diagram of ABB IRB 120

$p_{23} = 0z_2 + 70x_3$
$p_{34} = 302z_3 + 0x_4$
$p_{45} = 0z_4 + 0x_5$
$p_{56} = 72z_5 + 0x_6$
Read the angle and the θ angle from the graph, we get the SDH parameters:

i	d	a	α	θ
1	290	0	$-\frac{\pi}{2}$	0
2	0	270	0	$-\frac{\pi}{2}$
3	0	70	$-\frac{\pi}{2}$	0
4	302	0	$\frac{\pi}{2}$	0
5	0	0	$-\frac{\pi}{2}$	0
6	72	0	0	0

Now for the standford arm:

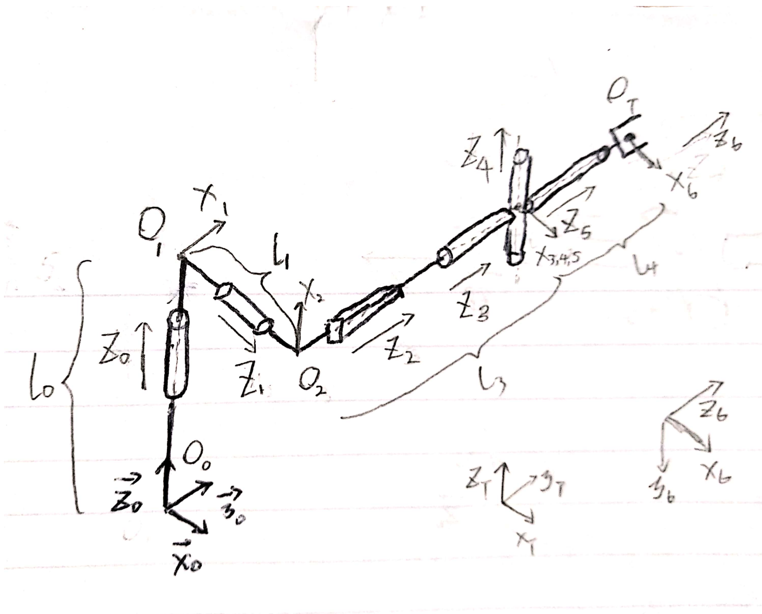

Figure 4: SDH diagram of Standford Arm

$$R_{6T} = [(X_T)_6 \mid (Y_T)_6 \mid (Z_T)_6] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Read off from graph to get the SDH parameters for Standford:

i	0	a	α	θ
1	l_0	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
2	l_1	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
3	l_3	0	0	$\frac{\pi}{2}$
4	0	0	$\frac{\pi}{2}$	0
5	0	0	$-\frac{\pi}{2}$	0
6	l_4	0	0	0

c)

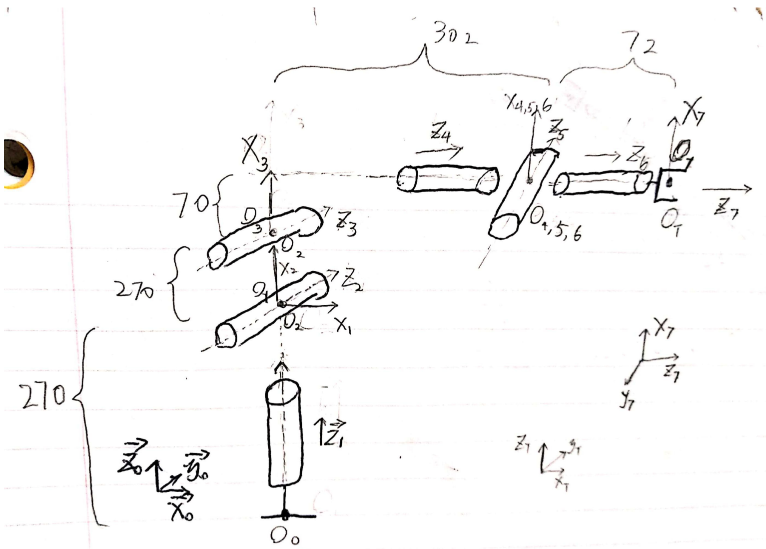

Figure 5: MDH diagram of ABB IRB 120

$$R_{7T} = [(X_T)_7 \mid (Y_T)_7 \mid (Z_T)_7] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

From graph, we can see that the parameters are:

i	d	a_{i-1}	α	θ
1	290	0	0	0
2	0	0	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$
3	0	270	0	0
4	302	70	$-\frac{\pi}{2}$	0
5	0	0	$\frac{\pi}{2}$	0
6	0	0	$-\frac{\pi}{2}$	0
6	72	0	0	0

Now for the Standford Arm:

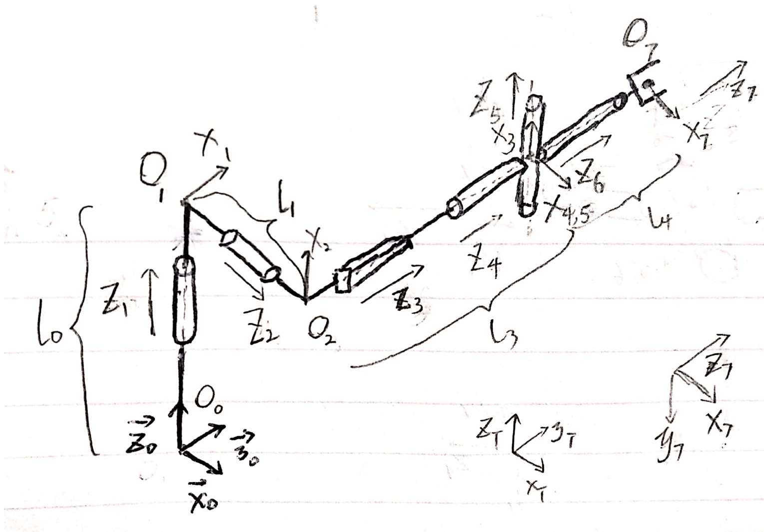

Figure 6: MDH diagram of Standford Arm

$$R_{7T} = [(X_T)_7 \mid (Y_T)_7 \mid (Z_T)_7] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

From graph, we can see that the parameters are:

i	d	a_{i-1}	α	θ
1	l_0	0	0	$\frac{\pi}{2}$
2	l_1	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
3	l_3	0	$\frac{\pi}{2}$	0
4	0	0	0	$\frac{\pi}{2}$
5	0	0	$\frac{\pi}{2}$	0
6	0	0	$-\frac{\pi}{2}$	0
7	l_4	0	0	0

This is Matlab verification of Problem 1
POE, SDH, MDH for ABB IRB 120:

Listing 1: code verification for ABB

```
%
% This is verification for ABB

syms q1 q2 q2 q3 q4 q5 q6 real
syms a b c d e f g positive

ex=[1;0;0];ey=[0;1;0];ez=[0;0;1];zv=[0;0;0];
```

```
q=[q1;q2;q3;q4;q5;q6];
n=length(q);

% POE

P=[zv,290*ez,270*ez,302*ex+70*ez,zv,zv,72*ex];
H=[ez,ey,ey,ex,ey,ex];
jointtype=zeros(n,1);jointtype(4)=0;

[R1,p1]=generalfwdkin(q,jointtype,H,P,n);
T1=[R1 p1;zeros(1,3) 1];

% SDH

D=[290 0 0 302 0 72]';
A=[0 270 70 0 0 0]';
af=[-pi/2 0 -pi/2 pi/2 -pi/2 0]';
th=[0 -pi/2 0 0 0 0]';

T=eye(4,4);
for i=1:n
    if jointtype(i)==0
      T=T*sdh(q(i)+th(i),af(i),D(i),A(i));
    else
      T=T*sdh(th(i),af(i),q(i)+D(i),A(i));
    end
end
R6T=[0 0 1; 0 -1 0; 1 0 0];
T2=T*[R6T zeros(3,1);zeros(1,3) 1];

% verify POE matches with SDH
simplify(T1-T2)

% MDH

D=[290 0 0 302 0 0 72]';
A=[0 0 270 70 0 0 0]';
af=[0 -pi/2 0 -pi/2 pi/2 -pi/2 0]';
th=[0 -pi/2 0 0 0 0 0]';

T=eye(4,4);
for i=1:n
    if jointtype(i)==0
        T=T*mdh(q(i)+th(i),af(i),D(i),A(i));
    else
        T=T*mdh(th(i),af(i),q(i)+D(i),A(i));
```

```
    end
end
T=T*mdh(th(n+1),af(n+1),D(n+1),A(n+1));
R7T=[0 0 1; 0 -1 0; 1 0 0];
T3=T*[R7T zeros(3,1);zeros(1,3) 1];

% verify POE matches with MDH
simplify(T1-T3)
```

With final output:

```
ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]


ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
```

Figure 7: Output for ABB IRB 120

POE, SDH, MDH for Standford:

Listing 2: Code verification for Standford

```
%
% This is verification of POE, SDH, MDH for standford


syms q1 q2 q2 q3 q4 q5 q6 real
syms l0 l1 l3 l4 positive

ex=[1;0;0];ey=[0;1;0];ez=[0;0;1];zv=[0;0;0];

q=[q1;q2;q3;q4;q5;q6];
n=length(q);

% POE

P=[zv,l0*ez,l1*ex,l3*ey,zv,zv,l4*ey];
H=[ez,ex,ey,ey,ez,ey];
jointtype=zeros(n,1);jointtype(3)=1;

[R1,p1]=generalfwdkin(q,jointtype,H,P,n);
T1=[R1 p1;zeros(1,3) 1];
```

```
% SDH

D=[l0 l1 l3 0 0 l4]';
A=[0 0 0 0 0 0]';
af=[pi/2 pi/2 0 pi/2 -pi/2 0]';
th=[pi/2 pi/2 pi/2 0 0 0]';

T=eye(4,4);
for i=1:n
    if jointtype(i)==0
      T=T*sdh(q(i)+th(i),af(i),D(i),A(i));
    else
      T=T*sdh(th(i),af(i),q(i)+D(i),A(i));
    end
end
R6T=[1 0 0; 0 0 -1; 0 1 0];
T2=T*[R6T zeros(3,1);zeros(1,3) 1];

% verify POE matches with SDH
simplify(T1-T2)

% MDH

D=[l0 l1 l3 0 0 0 l4]';
A=[0 0 0 0 0 0 0]';
af=[0 pi/2 pi/2 0 pi/2 -pi/2 0]';
th=[pi/2 pi/2 0 pi/2 0 0 0]';

T=eye(4,4);
for i=1:n
    if jointtype(i)==0
        T=T*mdh(q(i)+th(i),af(i),D(i),A(i));
    else
        T=T*mdh(th(i),af(i),q(i)+D(i),A(i));
    end
end
T=T*mdh(th(n+1),af(n+1),D(n+1),A(n+1));
R7T=[1 0 0; 0 0 -1; 0 1 0];
T3=T*[R7T zeros(3,1);zeros(1,3) 1];

% verify POE matches with MDH
simplify(T1-T3)
```

With final output:

```
ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]


ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
```

Figure 8: Output for Standford

Problem 2

For ABB IRB 120 with a random q, POE, SDH, and MDH all generate the same (R0T, p0T):

Listing 3: Code for Problem 2

```
%
% This is HW5_2 robot verification
%ABB
clear all;close all;clc;
%syms q1 q2 q2 q3 q4 q5 q6 real
q=(rand(6,1)-.5)*2*pi
ex=[1;0;0];ey=[0;1;0];ez=[0;0;1];zv=[0;0;0];
%q=[q1;q2;q3;q4;q5;q6];
n=length(q);

% POE

P=[zv,290*ez,270*ez,302*ex+70*ez,zv,zv,72*ex];
H=[ez,ey,ey,ex,ey,ex];
jointtype=zeros(n,1);jointtype(4)=0;

[R1,p1]=generalfwdkin(q,jointtype,H,P,n);
T1=[R1 p1;zeros(1,3) 1]

% SDH

D=[290 0 0 302 0 72]';
A=[0 270 70 0 0 0]';
af=[-pi/2 0 -pi/2 pi/2 -pi/2 0]';
```

```
th=[0 -pi/2 0 0 0 0]';

T=eye(4,4);
for i=1:n
    if jointtype(i)==0
      T=T*sdh(q(i)+th(i),af(i),D(i),A(i));
    else
      T=T*sdh(th(i),af(i),q(i)+D(i),A(i));
    end
end
R6T=[0 0 1; 0 -1 0; 1 0 0];
T2=T*[R6T zeros(3,1);zeros(1,3) 1]

% MDH

D=[290 0 0 302 0 0 72]';
A=[0 0 270 70 0 0 0]';
af=[0 -pi/2 0 -pi/2 pi/2 -pi/2 0]';
th=[0 -pi/2 0 0 0 0 0]';

T=eye(4,4);
for i=1:n
    if jointtype(i)==0
        T=T*mdh(q(i)+th(i),af(i),D(i),A(i));
    else
        T=T*mdh(th(i),af(i),q(i)+D(i),A(i));
    end
end
T=T*mdh(th(n+1),af(n+1),D(n+1),A(n+1));
R7T=[0 0 1; 0 -1 0; 1 0 0];
T3 = T*[R7T zeros(3,1);zeros(1,3) 1]

%plot the configuration
p1 = T1(1:3,4);
R = T1(1:3,1:3);
quat1 = R2q(R);
robot=loadrobot('abbirb120');
c=setJoint(robot,q);show(robot,c);
hold on;
plotTransforms(p1'/1000,quat1');
```

Output:

```
q =

   -1.4131
    2.0606
    1.8396
   -1.4216
    0.7269
   -2.5035


T1 =

   -0.7235   -0.5861   -0.3646  -56.6617
    0.3653    0.1231   -0.9227   55.0231
    0.5857   -0.8008    0.1250  362.1005
         0         0         0    1.0000


T2 =

   -0.7235   -0.5861   -0.3646  -56.6617
    0.3653    0.1231   -0.9227   55.0231
    0.5857   -0.8008    0.1250  362.1005
         0         0         0    1.0000


T3 =

   -0.7235   -0.5861   -0.3646  -56.6617
    0.3653    0.1231   -0.9227   55.0231
    0.5857   -0.8008    0.1250  362.1005
         0         0         0    1.0000
```

Figure 9: With a randomly generated q, all T with different methods are same

and the visualization of ABB IRB 120:

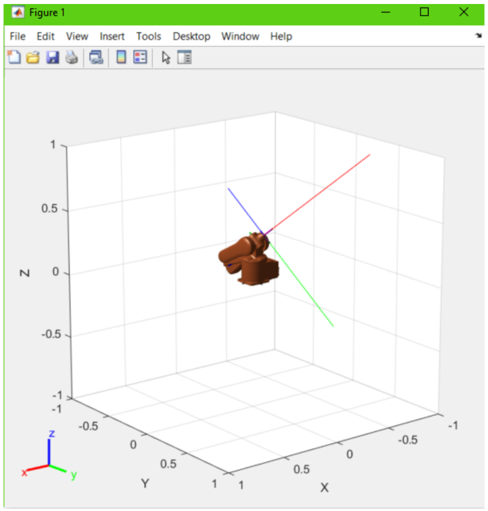

Figure 10: Matlab calculation and our method align

With a different q: (0.0035 0.2549 -0.9730 -1.3105 1.2788 2.7274)

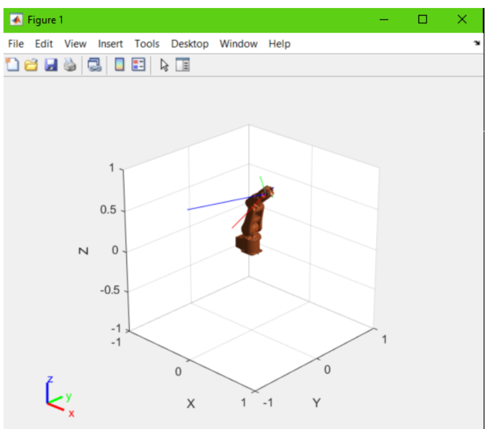

Figure 11: with a different q

Problem 3

Listing 4: Code for URDF

```
<?xml version="1.0"?>
<robot name="origins">

 <material name="blue">
   <color rgba="0 0 0.8 1"/>
  </material>

  <material name="white">
    <color rgba="1 1 1 1"/>
  </material>

  <material name="red">
        <color rgba="1 0 0 1"/>
  </material>

  <link name="base_link">
    <visual>
      <geometry>
        <cylinder length="0.3" radius="0.05"/>
      </geometry>
    </visual>
  </link>

  <link name="right_leg">
    <visual>
      <geometry>
        <cylinder length="1" radius="0.03"/>
      </geometry>
      <origin rpy="0 1.57075 0" xyz="0 0 0"/>
      <material name="blue"/>
    </visual>
  </link>

  <joint name="base_to_right_leg" type="fixed">
    <parent link="base_link"/>
    <child link="right_leg"/>
    <origin xyz="0.5 0 0"/>
  </joint>

  <link name="second_base_link">
    <visual>
      <geometry>
```

```
      <cylinder length="0.2" radius="0.05"/>
    </geometry>
  </visual>
</link>

<joint name="second_joint" type="fixed">
  <parent link="right_leg"/>
  <child link="second_base_link"/>
  <origin xyz="0.5 0 0"/>
</joint>

<link name="second_leg_link">
  <visual>
    <geometry>
      <cylinder length="0.5" radius="0.03"/>
    </geometry>
    <origin rpy="0 1.57075 0" xyz="0 0 0"/>
    <material name="red"/>
  </visual>
</link>

<joint name="second_right_leg" type="fixed">
  <parent link="second_base_link"/>
  <child link="second_leg_link"/>
  <origin xyz="0.25 0 0"/>
</joint>

<link name="sphere">
  <visual>
    <geometry>
      <sphere radius="0.1"/>
    </geometry>
    <material name="white"/>
  </visual>
</link>
<joint name="sphere place" type="fixed">
  <parent link="second_leg_link"/>
  <child link="sphere"/>
  <origin xyz="0.25 0 0"/>
</joint>
</robot>
```

Enter the command:

```
>> show(importrobot('hw5_3.urdf'))

ans =

  Axes (Primary) with properties:

             XLim: [-2 2]
             YLim: [-2 2]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Show all properties
```

Figure 12: Matlab command

With output:

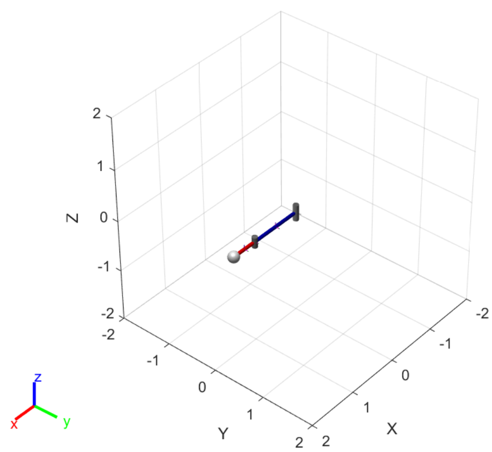

Figure 13: Two linked arm built by URDF script