HW2B

Table of Contents

q = (50, -45, 30, 20, 10, 0)

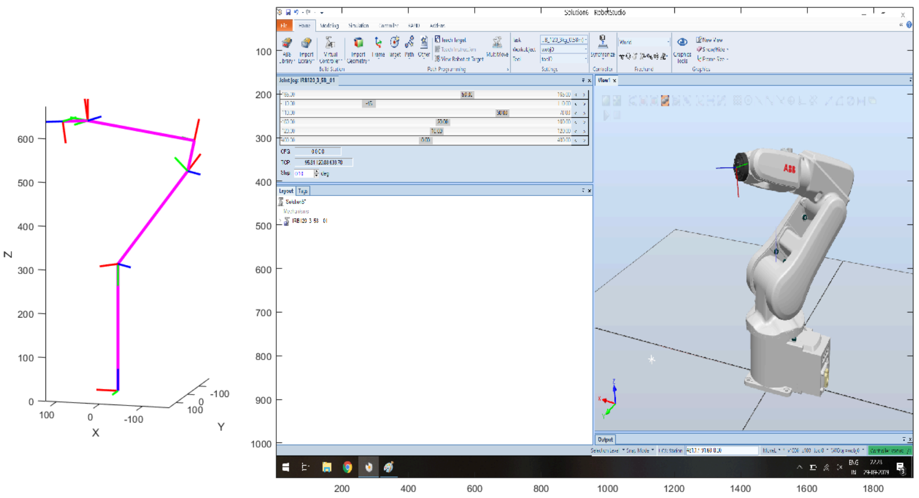
For configuration:

```
q = (50, -45, 30, 20, 10, 0)
```

```
subplot(1,4,1);
[T] = plotArm(50, -45, 30, 20, 10, 0, f);
subplot(1,4,[2 3 4]);
image(imread('Images\01.png'));
set(gcf, 'Position', get(0, 'Screensize'));
```

```
T =

   -0.2119   -0.7767    0.5931   95.8449
    0.2715    0.5362    0.7992  120.8760
   -0.9388    0.3304    0.0973  633.7005
         0         0         0    1.0000
```



q = (10, 50, -45, 20, 10, 0)

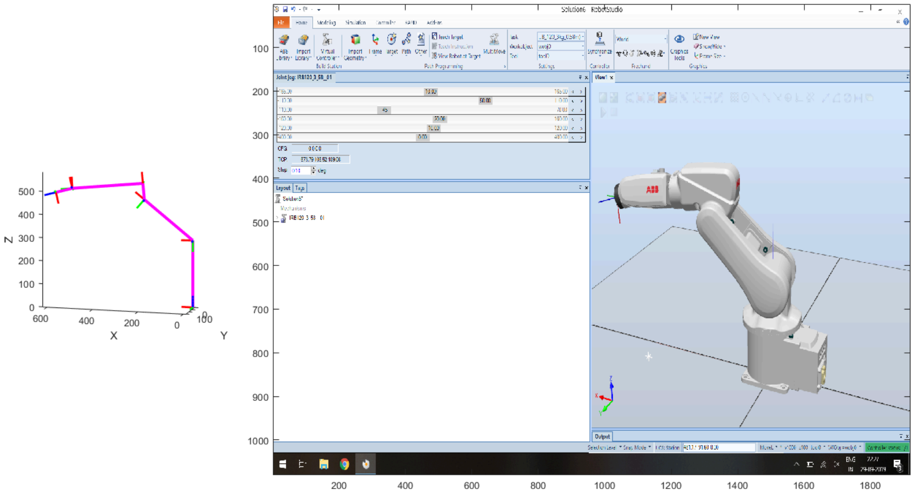
For configuration:

```
q = (10, 50, -45, 20, 10, 0)
```

```
subplot(1,4,1);
[T] = plotArm(10, 50, -45, 20, 10, 0, f);
subplot(1,4,[2 3 4]);
image(imread('Images\02.png'));
set(gcf, 'Position', get(0, 'Screensize'));
```

```
T =

   -0.3083   -0.1338    0.9418  573.7904
    0.2877    0.9306    0.2264  105.5169
   -0.9068    0.3407   -0.2484  489.0814
         0         0         0    1.0000
```



q = (50, -45, 30, 50, -20, 40)

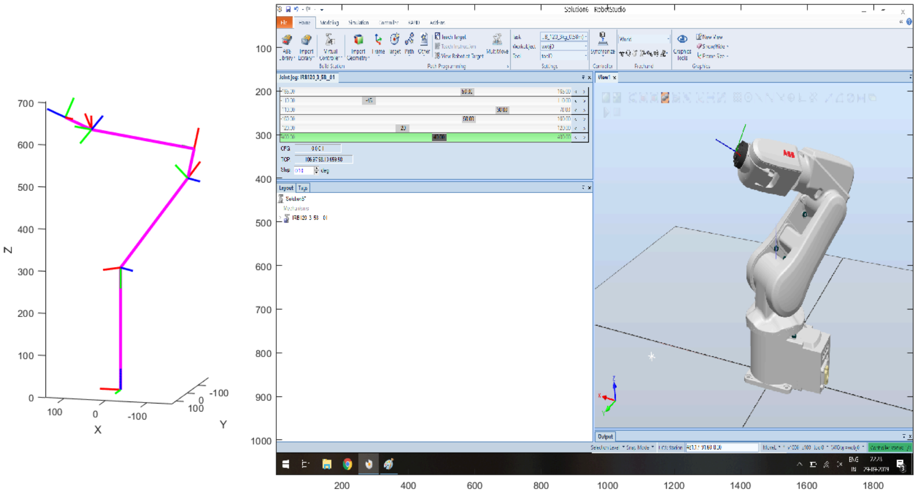
For configuration:

```
q = (50, -45, 30, 50, -20, 40)
```

```
subplot(1,4,1);
[T] = plotArm(50, -45, 30, 50, -20, 40, f);
subplot(1,4,[2 3 4]);
image(imread('Images\03.png'));
set(gcf, 'Position', get(0, 'Screensize'));
```

```
T =

   -0.5812   -0.3215    0.7476  106.9666
    0.8080   -0.3369    0.4833   98.1304
    0.0965    0.8850    0.4556  659.4977
         0         0         0    1.0000
```



q = (-30, -60, 20, 40, -20, 40)

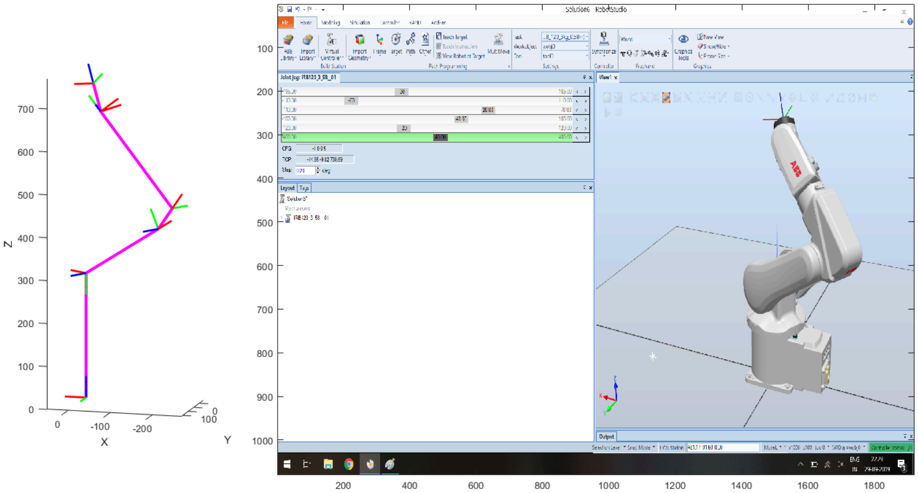
For configuration:

```
q = (-30, -60, 20, 40, -20, 40)
```

```
subplot(1,4,1);
[T] = plotArm(-30, -60, 20, 40, -20, 40, f);
subplot(1,4,[2 3 4]);
image(imread('Images\04.png'));
set(gcf, 'Position', get(0, 'Screensize'));
```

```
T =

    0.7283   -0.5782    0.3676  -14.6463
    0.6824    0.5631   -0.4661   -9.8216
    0.0625    0.5903    0.8047  730.6854
         0         0         0    1.0000
```



q = (0, 32, 45, 30, 56, 40)

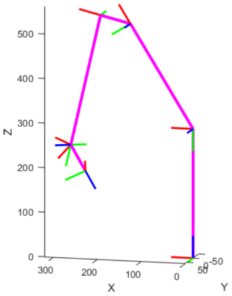
For configuration:

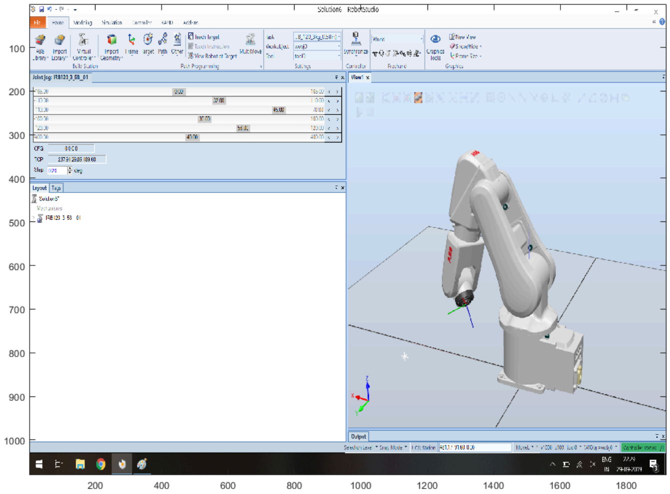
```
q = (0, 32, 45, 30, 56, 40)
```

```
subplot(1,4,1);
[T] = plotArm(0, 32, 45, 30, 56, 40, f);
subplot(1,4,[2 3 4]);
image(imread('Images\05.png'));
set(gcf, 'Position', get(0, 'Screensize'));
```

```
T =

   -0.1912    0.7964   -0.5738  237.9075
    0.7709    0.4837    0.4145   29.8454
    0.6076   -0.3631   -0.7064  189.6013
         0         0         0    1.0000
```





Joint Jog: IRB120_3_58_01

Task: IRB_120_3kg_0.58m
Workobject: wobj0
Tool: tool0

-165.00	0.00	165.00
-110.00	32.00	110.00
-110.00	45.00	70.00
-160.00	30.00	160.00
-120.00	56.00	120.00
-400.00	40.00	400.00

CFG: 0 0 0 0

TCP: 237.91 29.85 189.60

Step: 0.01 deg

Appendix

Plot Arm Function

```
function [T] = plotArm(val1, val2, val3, val4, val5, val6, f)

syms q1 q2 q3 q4 q5 q6
q = [q1 q2 q3 q4 q5 q6].';
vals = [val1 val2 val3 val4 val5 val6].';
[n_dof,m] = size(q);
Individual_Transforms = sym(zeros(4,4,n_dof));
Transforms = sym(zeros(4,4,n_dof));

thetas = q;
thetas(2,1) = q2 - 90;
thetas(6,1) = q6 + 180;
d = [290 0 0 302 0 72].';
a = [0 270 70 0 0 0].';
alpha = [-90 0 -90 90 -90 0].';

for i = 1:n_dof
    Individual_Transforms(:,:,i) = ...
                dhparam2matrix(thetas(i,1), d(i,1), a(i,1),
 alpha(i,1));
    if i == 1
        Transforms(:,:,1) = Individual_Transforms(:,:,1);
    else
        Transforms(:,:,i) = ...
                Transforms(:,:,i-1)*Individual_Transforms(:,:,i);
    end
end
```

```
T01 = Individual_Transforms(:,:,1);
T12 = Individual_Transforms(:,:,2);
T23 = Individual_Transforms(:,:,3);
T34 = Individual_Transforms(:,:,4);
T45 = Individual_Transforms(:,:,5);
T56 = Individual_Transforms(:,:,6);

T06 = Transforms(:,:,6);

solution = double(simplify(fk_solve(Transforms, q, vals, 0)));
T = solution(:,:,6)
indTransforms = double( ...
                    simplify(fk_solve(Individual_Transforms, q, vals,
 0)));

f = drawManip(solution, indTransforms, 'ABB_Robot');

end
```

dhparam2matrix Function

```
function [trans] = dhparam2matrix(theta, d, a, alpha)
    %DHPARAM2MATRIX This function returns a Homogenous Transformation
 matrix
    % given the input parameters a, alpha, theta and d derived from
 the
    % DH Parameters of the robot.
    %   Please enter all values in degrees and not radians.

    cos_alpha = @cosd;
    sin_alpha = @sind;
    cos_theta = @cosd;
    sin_theta = @sind;

    a11 = cos_theta(theta);
    a12 = -sin_theta(theta)*cos_alpha(alpha);
    a13 = sin_theta(theta)*sin_alpha(alpha);
    a14 = a*cos_theta(theta);
    a21 = sin_theta(theta);
    a22 = cos_theta(theta)*cos_alpha(alpha);
    a23 = -cos_theta(theta)*sin_alpha(alpha);
    a24 = a*sin_theta(theta);
    a31 = 0;
    a32 = sin_alpha(alpha);
    a33 = cos_alpha(alpha);
    a34 = d;
    scl = 1;

    trans = [a11 a12 a13 a14;
             a21 a22 a23 a24;
             a31 a32 a33 a34;
             0   0   0   scl];
end
```

fk_solve Function

```
function [T] = fk_solve(Transforms, states, vals, n)
%FK_SOLVE This function substitutes the states in the transformation
 matrix
%with the values specified while maintaining the joint constraints
 imposed
%by the ABB Arm
%   @param transforms is the 4x4xm transformation matrices to solve
%   @param states are the states
%   @param vals are the values required
%   @param n is the T of n wrt 0 with substituted values.
%

q = vals;

if abs(q(1,1)) > 165
    disp('Joint 1 has reached max limit')
    if q(1,1) > 0
        q(1,1) = 165;
    else
        q(1,1) = -165;
    end
end

if abs(q(2,1)) > 110
    disp('Joint 2 has reached max limit')
    if q(2,1) > 0
        q(2,1) = 110;
    else
        q(2,1) = -110;
    end
end

if q(3,1) > 70 || q(3,1) < -110
    disp('Joint 3 has reached max limit')
    if q(3,1) > 0
        q(3,1) = 70;
    else
        q(3,1) = -110;
    end
end

if abs(q(4,1)) > 160
    disp('Joint 4 has reached max limit')
    if q(4,1) > 0
        q(4,1) = 160;
    else
        q(4,1) = -160;
    end
end

if abs(q(5,1)) > 120
    disp('Joint 5 has reached max limit')
    if q(5,1) > 0
        q(5,1) = 120;
    else
        q(5,1) = -120;
    end
end

if abs(q(6,1)) > 400
    disp('Joint 6 has reached max limit')
    if q(6,1) > 0
        q(6,1) = 400;
    else
        q(6,1) = -400;
    end
end

Transforms = subs(Transforms,[states], [q]);
if n == 0
    T = Transforms(:,:,:);
else
    T = Transforms(:,:,n);
end
end
```

drawManip Function

```
function [f] = drawManip(Transformations, indTransforms, Name)
%DRAWMANIP Summary of this function goes here
%   Detailed explanation goes here
[l,m,n] = size(Transformations);
f = gcf;
az = -165;
el = 11;
rotate3d on
grid on
xlabel('X');
ylabel('Y');
zlabel('Z');
R1 = [1 0 0;
      0 1 0;
      0 0 1];
x = [0, Transformations(1,4,1)];
y = [0, Transformations(2,4,1)];
z = [0, Transformations(3,4,1)];
```

```
plot3(x, y, z,'m','LineWidth',3);
hold on
drawAxisLines(f,R1,[0;0;0]); % indTransforms(1:3,1:3,1)
 Transformations(1:3,4,1)
view(az,el);
axis('equal');
for i = 2:n
    x = [Transformations(1,4,i-1), Transformations(1,4,i)];
    y = [Transformations(2,4,i-1), Transformations(2,4,i)];
    z = [Transformations(3,4,i-1), Transformations(3,4,i)];
%
 drawAxisLines(f,indTransforms(1:3,1:3,i),Transformations(1:3,4,i))

 drawAxisLines(f,Transformations(1:3,1:3,i-1),Transformations(1:3,4,i-1));
    plot3(x, y, z,'m','LineWidth',3);
end

drawAxisLines(f,Transformations(1:3,1:3,i),Transformations(1:3,4,i));

hold off
end
```

drawAxisLines Function

```
function [p] = drawAxisLines(fig, rot, p_in)
%DRAWAXISLINES Summary of this function goes here
%   Detailed explanation goes here
x = [1 0 0].';
y = [0 1 0].';
z = [0 0 1].';
rX = rot*x;
rY = rot*y;
rZ = rot*z;
rX = 50*(rX(:,1)) + p_in;
rY = 50*(rY(:,1)) + p_in;
rZ = 50*(rZ(:,1)) + p_in;
ax_x = [p_in(:,1), rX(:,1)];
ax_y = [p_in(:,1), rY(:,1)];
ax_z = [p_in(:,1), rZ(:,1)];
hold on
xlabel('X');
ylabel('Y');
zlabel('Z');
plot3(ax_x(1,:), ax_x(2,:), ax_x(3,:), 'r', 'LineWidth',2);
hold on
plot3(ax_y(1,:), ax_y(2,:), ax_y(3,:), 'g', 'LineWidth',2);
plot3(ax_z(1,:), ax_z(2,:), ax_z(3,:), 'b', 'LineWidth',2);
end
```

Published with MATLAB® R2019a