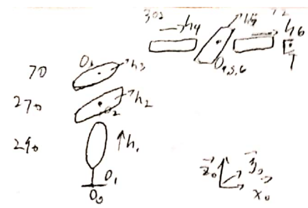


$$P_{01} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_{12} = \begin{bmatrix} 0 \\ 0 \\ 290 \end{bmatrix} \quad P_{23} = \begin{bmatrix} 0 \\ 0 \\ 270 \end{bmatrix} \quad P_{34} = \begin{bmatrix} 302 \\ 0 \\ 70 \end{bmatrix}$$

$$P_{45} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_{56} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_{6T} = \begin{bmatrix} 72 \\ 0 \\ 0 \end{bmatrix}$$

$R_{01} = R_z(q_1)$ $\quad R_{12} = R_y(q_2)$ $\quad R_{23} = R_y(q_3)$ $\quad R_{34} = R_x(q_4)$ $\quad R_{45} = R_y(q_5)$

$R_{56} = R_x(q_6)$ $\quad R_{6T} = I$

$$P_{0T} = \cancel{P_{01}} + R_{01}P_{12} + R_{01}R_{12}P_{23} + R_{01}R_{12}R_{23}P_{34} + \cancel{R_{01}R_{12}R_{23}R_{34}P_{45}}$$
$$+ \cancel{R_{12}R_{23}R_{34}R_{45}P_{56}} + \underbrace{R_{01}R_{12}R_{23}R_{34}R_{45}R_{56}}_{R_{06} = R_{0T}}P_{6T}$$

$$P_{0T} = R_z(q_1)P_{12} + R_z(q_1)R_y(q_2)P_{23} + R_z(q_1)R_y(q_2)R_y(q_3)P_{34} + R_{0T}P_{6T}$$

$$R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) = P_{12} + R_y(q_2)P_{23} + \underbrace{R_y(q_2)R_y(q_3)}_{R_y(q_2+q_3)}P_{34}$$

$e_y^T R_y(\theta) = e_y^T$

$$e_y^T R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) = e_y^T(P_{12} + P_{23} + P_{34})$$

Subproblem 4 $\quad d = h^T R(k,q)P$

solve q_1
$$\begin{cases} d = e_y^T(P_{12}+P_{23}+P_{34}) \\ h = e_y \\ k = -e_z \\ P = P_{0T} - R_{0T}P_{6T} \end{cases}$$

q_1 might have 0, 1, 2 solution

$$R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) - P_{12} = R_y(q_2)(P_{23} + R_y(q_3)P_{34})$$
$$\| R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) - P_{12} \| = \| P_{23} + R_y(q_3)P_{34} \|$$

Sub 3

$d = \| P_1 + R(k,q)P_2 \|$

solve q_3
$$\begin{cases} d = \| R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) - P_{12} \| \\ k = e_y \\ P_2 = P_{34} \\ P_1 = P_{23} \end{cases}$$

0, 1, 2 solution

4 solution for (q_1, q_3) $\quad$ For each solution solve q_2

Subproblem 1 $\quad P_2 = R(k, q_1)P_1$

solve q_2
$$\begin{cases} P_1 = P_{23} + R_y(q_3)P_{34} \\ P_2 = R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) - P_{12} \\ k = e_y \end{cases}$$

1 solution for each (q_1, q_3)

4 solution for (q_1, q_2, q_3)

$$R_y(-(q_2+q_3))R_z(-q_1)R_{0T} = R_x(q_4)R_y(q_5)R_x(q_6)$$
$$e_x^T R_y(-(q_2+q_3))R_z(-q_1)R_{0T}e_x = \underbrace{e_x^T R_x(q_4)}_{e_x^T}R_y(q_5)e_x$$

Subproblem 4 $\quad d = h^T R(k,q)P$
$$\begin{cases} P = e_x \\ k = e_y \\ h = e_x \\ d = e_x^T R_y(-(q_2+q_3))R_z(-q_1)R_{0T}e_x \end{cases}$$

Solve 0, 1, 2, q_5

8 soluton for (q_1, q_2, q_3, q_5) for each

solve. $R_y(-(q_2+q_3))R_z(-q_1)R_{0T}e_x = R_x(q_4)R_y(q_5)e_x$.

subproblem 1

solve for q_4
$$\begin{cases} k = e_x \\ p_1 = R_y(q_5)e_x \\ p_2 = R_y(-(q_2+q_3))R_z(-q_1)R_{0T}e_x \end{cases}$$

take transpose of both side.

$$R_{0T}^T R_z(q_1)R_y(q_2+q_3) = R_x(-q_6)R_y(-q_5)R_x(-q_4)$$

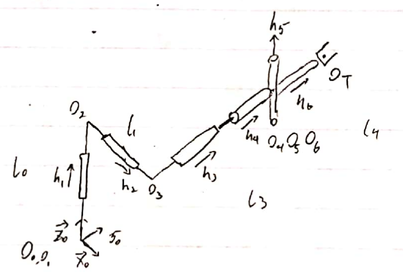
$$R_{0T}^T R_z(q_1)R_y(q_2+q_3)e_x = R_x(-q_6)R_y(-q_5)e_x$$

subproblem 1 solve q_6

1 solution
$$\begin{cases} p_1 = R_y(-q_5)e_x \\ p_2 = R_{0T}^T R_z(q_1)R_y(q_2+q_3)e_x \\ k = -e_x \end{cases}$$

One of the eight $(q_1, q_2, q_3, q_4, q_5, q_6)$ should map to the correct one

Standford arm:



$$P_{01} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_{12} = \begin{bmatrix} 0 \\ 0 \\ l_0 \end{bmatrix} \quad P_{23} = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \quad P_{34} = \begin{bmatrix} 0 \\ l_3+q_3 \\ 0 \end{bmatrix} \quad P_{45} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_{56} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_{6T} = \begin{bmatrix} 0 \\ l_4 \\ 0 \end{bmatrix}$$

$$R_{0T} = R_{01}R_{12}R_{23}R_{34}R_{45}R_{56}R_{6T}$$
$$= R_z(q_1)R_x(q_2)\, I\, R_y(q_4)R_z(q_5)R_y(q_6)\, I$$

$$P_{0T} = \cancel{P_{01}} + R_{01}P_{12} + R_{01}R_{12}P_{23} + \cancel{R_{01}R_{12}R_{23}R_{34}P_{45}} + \cancel{R_{01}R_{12}R_{23}R_{34}R_{45}P_{56}} + R_{01}R_{12}R_{23}R_{34}R_{45}R_{56}P_{6T}$$
$$+ R_{01}R_{12}R_{23}P_{34} \qquad P_{34} = (l_3+q_3)e_y$$

$$\begin{cases} R_{0T} = R_z(q_1)R_x(q_2)R_y(q_4)R_z(q_5)R_y(q_6) \\ P_{0T} = R_{01}P_{12} + R_{02}P_{23} + R_{03}(l_3+q_3)e_y + R_{0T}P_{6T} \end{cases}$$

$R_{03} = R_{02}$
since $R_{23} = I$

then $P_{0T} - R_{0T}P_{6T} = R_{01}P_{12} + R_{02}(P_{23} + (l_3+q_3)e_y)$

$$R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) = P_{12} + R_x(q_2)(P_{23} + (l_3+q_3)e_y)$$

since $e_x^T R_x(\theta) = e_x^T$

$$e_x^T R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) = e_x^T P_{12} + e_x^T(P_{23} + (l_3+q_3)e_y)$$

$$e_x^T R_z(-q_1)e_z = e_x^T(P_{12} + P_{23} + (l_3+q_3)e_y)(P_{0T} - R_{0T}P_{6T})^T e_z$$

solve for q_3

$$\underbrace{e_x^T e_z}_{\text{know.}} = \underbrace{e_x^T(P_{23} + l_3 e_y)(P_{0T} - R_{0T}P_{6T})^T e_z}_{\text{know}} + e_x^T q_3 e_y (P_{0T} - R_{0T}P_{6T})^T e_z$$

one solution at max

Now q_3 is know

$$R_z(-q_1)(P_{0T} - R_{0T}P_{6T}) = P_{12} + R_x(q_2)(P_{23} + (l_3+q_3)e_y)$$

$$\|P_{0T} - R_{0T}P_{6T}\| = \|P_{12} + R_x(q_2)(P_{23} + (l_3+q_3)e_y)\|$$

use sub problem 3
Solve for q_2

$$\begin{cases} d = \|P_{0T} - R_{0T}P_{6T}\| \\ P_1 = P_{12} \\ P_2 = P_{23} + (l_3+q_3)e_y \\ k = e_x \end{cases}$$

0, 1, 2 Solution

(q_2, q_3) 2 solution

$$R_z(q_1)(P_{0T} - R_{0T}P_{6T}) = P_{12} + R_x(q_2)(P_{23} + (l_3+q_3)e_y)$$

use subproblem 1
Solve for q_1

$$\begin{cases} P_1 = P_{0T} - R_{0T}P_{6T} \\ P_2 = P_{12} + R_x(q_2)(P_{23} + (l_3+q_3)e_y) \\ k = -e_z \end{cases}$$

one solution for each (q_2, q_3)

(q_1, q_2, q_3) two solution

$$R_{0T} = R_z(q_1)R_x(q_2)R_y(q_4)R_z(q_5)R_y(q_6)$$

$$R_x(-q_2)R_z(-q_1)R_{0T} = R_y(q_4)R_z(q_5)R_y(q_6)$$

time e_y^T on left
e_y on right

$$e_y^T R_x(-q_2)R_z(-q_1)R_{0T}e_y = e_y^T R_z(q_5)e_y$$

Sub problem 4
solve for q_5

$$d = h^T R(k,q)P$$

$$\begin{cases} P = e_y \\ k = e_z \\ h = e_y \\ d = e_y^T R_x(-q_2)R_z(-q_1)R_{0T}e_y \end{cases}$$

0, 1, 2 solution

thus 4 solution for (q_1, q_2, q_3, q_5)

$$R_x(-q_2)R_z(-q_1)R_{0T}e_y = R_y(q_4)R_z(q_5)e_y$$

Sub problem 1
Solve for q_4
one solution

$$\begin{cases} P_1 = R_z(q_5)e_y \\ P_2 = R_x(-q_2)R_z(-q_1)R_{0T}e_y \\ k = e_y \end{cases}$$

4 solution for $(q_1, q_2, q_3, q_4, q_5)$

take transpose

$$R_{0T}^T R_z(q_1)R_x(q_2) = R_y(-q_6)R_z(-q_5)R_y(-q_4)$$

$$R_{0T}^T R_z(q_1)R_x(q_2)e_y = R_y(-q_6)R_z(-q_5)e_y$$

Subproblem 1
Solve for q_6
one solution

$$\begin{cases} P_1 = R_z(-q_5)e_y \\ k = -e_y \\ P_2 = R_{0T}^T R_z(q_1)R_x(q_2)e_y \end{cases}$$

$(q_1, q_2, q_3, q_4, q_5, q_6)$
4 solution

Problem 2

Based on the algorithm above, the inverse kinematics for ABB 120

```
%
% Inverse Kinematics Example for ABB 120
%

clear all;close all;

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

h1=ez;h2=ey;h3=ey;h4=ex;h5=ey;h6=ex;
p01=[0, 0, 0]';p12=[0, 0, 290]';p23=[0, 0, 270]';
p34=[302,0,70]';p45=[0, 0, 0]';p56=[0, 0, 0]';
p6T=[72, 0, 0]';

abb120.H=[h1 h2 h3 h4 h5 h6];
abb120.P=[p01 p12 p23 p34 p45 p56 p6T];
abb120.joint_type=[0 0 0 0 0 0]; % 6R robot
n=6;

%N=10;
N=1;
for j=1:N
    q=(rand(6,1)-.5)*2*pi/2;
    % q=zeros(6,1);
    T=fwdkinrec(1,eye(4,4),q,abb120)
    R=T(1:3,1:3);p=T(1:3,4);

    tstart(j)=tic;
    % solve for q1

    q1=subprob4(-ez,ey,p-R*p6T,ey'*(p12+p23+p34));

    % solve for q3

    q3a=subprob3(ey,-p34,p23,norm(-p12+rotz(-q1(1))*(p-R*p6T)));
    q3b=subprob3(ey,-p34,p23,norm(-p12+rotz(-q1(2))*(p-R*p6T)));

    % solve for q2

    q2_a1=subprob1(ey,p23+roty(q3a(1))*p34,rotz(-q1(1))*(p-R*p6T)-
        ↪ p12);
    q2_a2=subprob1(ey,p23+roty(q3a(2))*p34,rotz(-q1(1))*(p-R*p6T)-
        ↪ p12);
```

```
    q2_b1=subprob1(ey,p23+roty(q3b(1))*p34,rotz(-q1(2))*(p-R*p6T)-
        ↪ p12);
    q2_b2=subprob1(ey,p23+roty(q3b(2))*p34,rotz(-q1(2))*(p-R*p6T)-
        ↪ p12);

    % put all 8 solutions together
    qsol=zeros(6,9);
    qsol(:,9)=q; % last column is the original
    qsol(1:3,1)=[q1(1);q2_a1;q3a(1)];
    qsol(1:3,2)=[q1(1);q2_a2;q3a(2)];
    qsol(1:3,3)=[q1(2);q2_b1;q3b(1)];
    qsol(1:3,4)=[q1(2);q2_b2;q3b(2)];

    % wrist angles

    for i=1:4
        qsol(1:3,i+4)=qsol(1:3,i);
        R03=rot(h1,qsol(1,i))*...
            rot(h2,qsol(2,i))*...
            rot(h3,qsol(3,i));
        % [q4vec,q5vec]=subproblem2(h4,h5,h6,R03'*R*h6);
        [q4vec,q5vec]=subprob2(-h4,h5,R03'*R*h6,h6);
        q4a=q4vec(1);q4b=q4vec(2);
        q5a=q5vec(1);q5b=q5vec(2);
        qsol(4:5,i)=[q4a;q5a];
        qsol(4:5,i+4)=[q4b;q5b];

        R05a=R03*rot(h4,q4a)*rot(h5,q5a);
        R05b=R03*rot(h4,q4b)*rot(h5,q5b);

        R56=R05a'*R;
        qsol(6,i)=atan2(.5*h6'*vee(R56-R56'),...
                        .5*(trace(R56)-1));
        R56=R05b'*R;
        qsol(6,i+4)=atan2(.5*h6'*vee(R56-R56'),...
                          .5*(trace(R56)-1));
    end

    telapse(j)=toc(tstart(j));

% check forward kinematics still agrees

   for i=1:8
       [T,J]=fwddiffkinrec(1,eye(4,4),zeros(6,n),...
                           qsol(:,i),abb120);
      R2=T(1:3,1:3);p2=T(1:3,4);
```

```
        jac(j,i,1:6,1:6)=J;
        if max(max(isnan(J)))==0;
            jacsv(j,1:6,i)=svd(J);
        else
            jacsv(j,1:6,i)=NaN*ones(1,6);
        end
        RR(j,i,1:3,1:3)=R2;pp(j,1:3,i)=p2;
        errp(j,i)=norm(p2-p);errR(j,i)=norm(R2-R);
    end


    disp('*** q and 8 solutions of inverse kinematics ***');
    disp(q)
    disp(qsol)
    disp('*** p0T and R0T comparison ***');
    disp(errp);
    disp(errR);

    disp('Jacobian singular values for the 8 solutions');
    disp(squeeze(jacsv(j,:,:)));
    disp('Minimum Jacobian singular values for the 8 solutions');
    disp(min(squeeze(jacsv(j,:,:))));
end

disp('computation time');
disp(telapse)
```

b)

Randomly generate the qs, and calculate the forward kinematics to get the rotation matrix and P_{0T}, then use the subproblems to calculate all the possible solution.

```
*** q and 8 solutions of inverse kinematics ***
    1.2901
   -0.9995
   -0.7420
   -1.1136
   -1.1433
    1.1602

    1.2901    1.2901    4.4317    4.4317    1.2901    1.2901    4.4317    4.4317    1.2901
   -0.3558   -0.9995    0.9995    0.3558   -0.3558   -0.9995    0.9995    0.3558   -0.9995
    4.3392   -0.7420    4.3392   -0.7420    4.3392   -0.7420    4.3392   -0.7420   -0.7420
    1.7181    2.0280   -1.3557   -1.7400   -1.4235   -1.1136    1.7859    1.4016   -1.1136
    0.9710    1.1433    0.9896    0.9762   -0.9710   -1.1433   -0.9896   -0.9762   -1.1433
   -1.3679   -1.9814   -1.4895   -0.8148    1.7737    1.1602    1.6521    2.3268    1.1602

*** p0T and R0T comparison ***
   1.0e-12 *

    0.2412    0.2542    0.3640    0.1608    0.2412    0.2542    0.3640    0.1608

   1.0e-15 *

    0.3588    0.4915    0.6849    0.4202    0.6691    0.3627    0.4739    0.5688
```

Figure 1: One of the eight solution obtained match with the original q

As indicated, the output error of P_{OT} and rotation matrix for each solution has is almost zero.

c)

```
clear all;close all;

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

h1=ez;h2=ey;h3=ey;h4=ex;h5=ey;h6=ex;
p01=[0, 0, 0]';p12=[0, 0, 290]';p23=[0, 0, 270]';
p34=[302,0,70]';p45=[0, 0, 0]';p56=[0, 0, 0]';
p6T=[72, 0, 0]';

abb120.H=[h1 h2 h3 h4 h5 h6];
abb120.P=[p01 p12 p23 p34 p45 p56 p6T];
abb120.joint_type=[0 0 0 0 0 0]; % 6R robot
n=6;
NN = 1000;
for j=1:NN
%starting
    q=(rand(6,1)-.5)*2*pi/2;
    % q=zeros(6,1);
    T=fwdkinrec(1,eye(4,4),q,abb120);
```

```
R=T(1:3,1:3);p=T(1:3,4);
tstart(j)=tic;
% solve for q1
q1=subprob4(-ez,ey,p-R*p6T,ey'*(p12+p23+p34));
% solve for q3
q3a=subprob3(ey,-p34,p23,norm(-p12+rotz(-q1(1))*(p-R*p6T)));
q3b=subprob3(ey,-p34,p23,norm(-p12+rotz(-q1(2))*(p-R*p6T)));
% solve for q2
q2_a1=subprob1(ey,p23+roty(q3a(1))*p34,rotz(-q1(1))*(p-R*p6T)-
    ↪ p12);
q2_a2=subprob1(ey,p23+roty(q3a(2))*p34,rotz(-q1(1))*(p-R*p6T)-
    ↪ p12);
q2_b1=subprob1(ey,p23+roty(q3b(1))*p34,rotz(-q1(2))*(p-R*p6T)-
    ↪ p12);
q2_b2=subprob1(ey,p23+roty(q3b(2))*p34,rotz(-q1(2))*(p-R*p6T)-
    ↪ p12);
% put all 8 solutions together
qsol=zeros(6,9);
qsol(:,9)=q; % last column is the original
qsol(1:3,1)=[q1(1);q2_a1;q3a(1)];
qsol(1:3,2)=[q1(1);q2_a2;q3a(2)];
qsol(1:3,3)=[q1(2);q2_b1;q3b(1)];
qsol(1:3,4)=[q1(2);q2_b2;q3b(2)];
% wrist angles
for i=1:4
    qsol(1:3,i+4)=qsol(1:3,i);
    R03=rot(h1,qsol(1,i))*...
        rot(h2,qsol(2,i))*...
        rot(h3,qsol(3,i));
    % [q4vec,q5vec]=subproblem2(h4,h5,h6,R03'*R*h6);
    [q4vec,q5vec]=subprob2(-h4,h5,R03'*R*h6,h6);
    q4a=q4vec(1);q4b=q4vec(2);
    q5a=q5vec(1);q5b=q5vec(2);
    qsol(4:5,i)=[q4a;q5a];
    qsol(4:5,i+4)=[q4b;q5b];

    R05a=R03*rot(h4,q4a)*rot(h5,q5a);
    R05b=R03*rot(h4,q4b)*rot(h5,q5b);

    R56=R05a'*R;
    qsol(6,i)=atan2(.5*h6'*vee(R56-R56'),...
                    .5*(trace(R56)-1));
    R56=R05b'*R;
    qsol(6,i+4)=atan2(.5*h6'*vee(R56-R56'),...
                      .5*(trace(R56)-1));
end
```

```
    telapse(j)=toc(tstart(j));

    [Td,J]=fwddiffkinrec(1,eye(4,4),zeros(6,n),q,abb120);
    Rd=Td(1:3,1:3);pd=Td(1:3,4);Qd=R2q(Rd);
    % initial guess is zero joint angle
    q0=zeros(n,1);
    % parameters
    N=20;alpha=0.5;epsilon=.1;
    % iterative inverse kinematics algorithm
    tstart=tic;
    [qmat,p0T,Q0T]=invkiniter(abb120,q0,Td,N,alpha,epsilon);
    telapse1(j)=toc(tstart);
end
disp('mean for the iterative inverse kinematics');
disp(mean(telapse1(100:1000))); %for the iterative inverse kinematics
disp('mean for the subproblems');
disp(mean(telapse(100:1000))); %for subproblems
disp('time ratio between iterative and subproblem');
disp(mean(telapse1(100:1000))/mean(telapse(100:1000)));
```

As shown in the code, two approaches has been run for 1000 times and the computing time from 100 to 1000 times are recorded. The exact solution using subproblem is about 27 times faster than the iterative if we exclude the first 100 rounds. An example output goes like:

```
>> hw7compare
mean for the iterative inverse kinematics
    0.0091

mean for the subproblems
   3.3661e-04

time ratio between iterative and subproblem
   27.0612
```

Figure 2: Computing time comparison for ABBirb 120

For the errors, the typical error of R and P for exact subproblem solution has the magnitude of 1.0e-12 for P and the magnitude of 1.0e-14 for R. The position error (mm) quaternion error in the iterative method has a magnitude about 1.0e-03. Thus the exact solution using subproblem has better accuracy. However, the subproblem method does not always have a solution while the iterative method will return a solution. In the iterative method, a good initial guess will make the P and R converge fast compared to a initial guess far away from the target. For example, we can define the initial guess as $q0 = q + \pi/10$, we will get the following results:

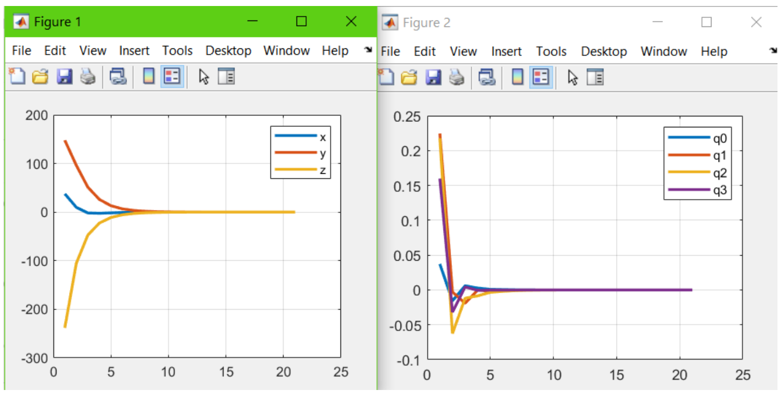


Figure 3: when initial guess is close to the target

Record the q to use it again, and let the initial guess be $q0 = q + \pi/3$, the result will then become:

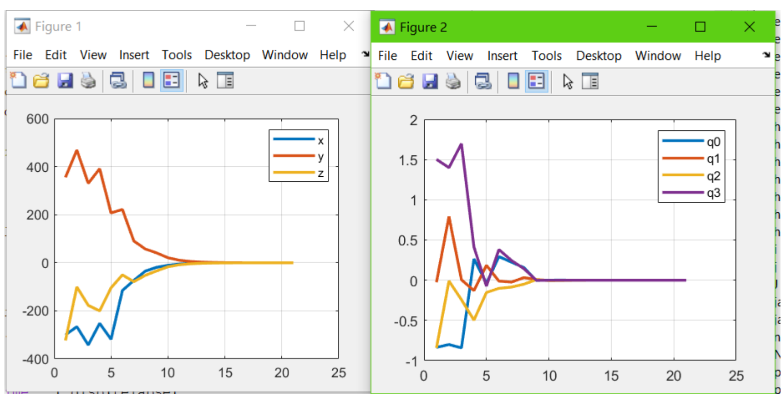


Figure 4: when initial guess is far from the target

When the initial guess is far from the target, the curve of arm becomes less smooth.
In the subproblem method, given the R and P, there are at most 8 possible different poses from the ABBirb 120. For each pose, the Jacobian will be different, and the singularity associated with the Jacobian will be different:

```
Jacobian singular values for the 8 solutions
  652.5010  651.6800  651.6997  652.5158  652.5010  651.6800  651.6997  652.5158
  112.0258  111.4774  114.2645  114.1220  112.0258  111.4774  114.2645  114.1220
   79.6790   75.8158   75.8270   79.6827   79.6790   75.8158   75.8270   79.6827
    1.0171    1.0229    1.0162    1.0211    1.0171    1.0229    1.0162    1.0211
    0.1693    0.1670    0.1753    0.1741    0.1693    0.1670    0.1753    0.1741
    0.0662    0.0696    0.0697    0.0662    0.0662    0.0696    0.0697    0.0662

Minimum Jacobian singular values for the 8 solutions
    0.0662    0.0696    0.0697    0.0662    0.0662    0.0696    0.0697    0.0662
```

Figure 5: Different q with different Jacobian

The minimum Jacobian will indicate how close the arm is to the singularity. For the infeasible pose, the iterative method will make the arm reach to the boundary singularity since it will stretch and try to reach to the target as close as possible. There is no joint stop in this question. In reality, not all robot joint will not be able to rotate 360 degree without any constraints.