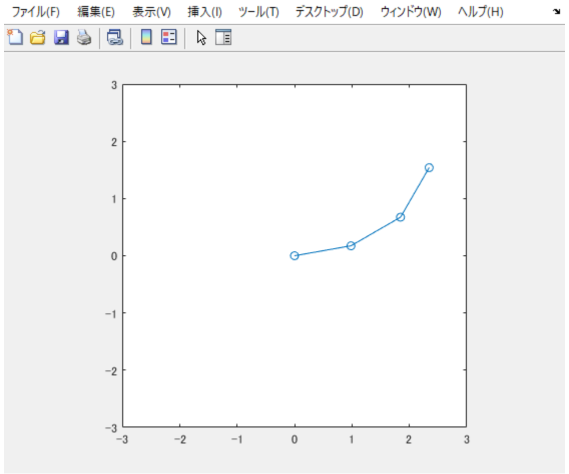
# SAMPLEFKASHOTPLANNER3LINK.M

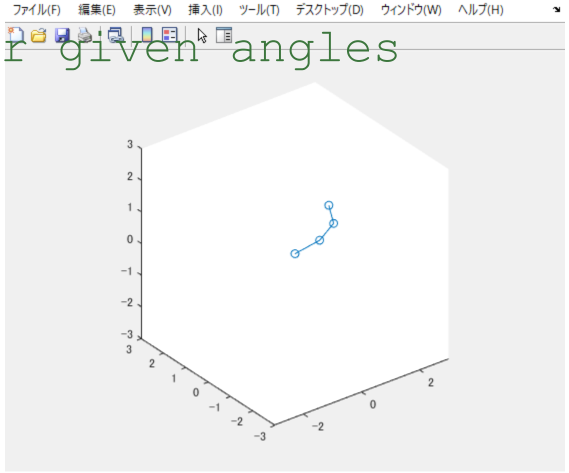
# Setup of Arm Structure

```
% Link lengths
L1 = 1; L2 = 1; L3 = 1;

% Joint angles
q1 = deg2rad(10.0);
q2 = deg2rad(20.0);
q3 = deg2rad(30.0);

% Homegeneous transformation matrices for given angles
% a frame to previous frame
T01 = DH(0,  0, 0, q1 );
T12 = DH(0, L1, 0, q2 );
T23 = DH(0, L2, 0, q3 );
T34 = DH(0, L3, 0, 0  );
```

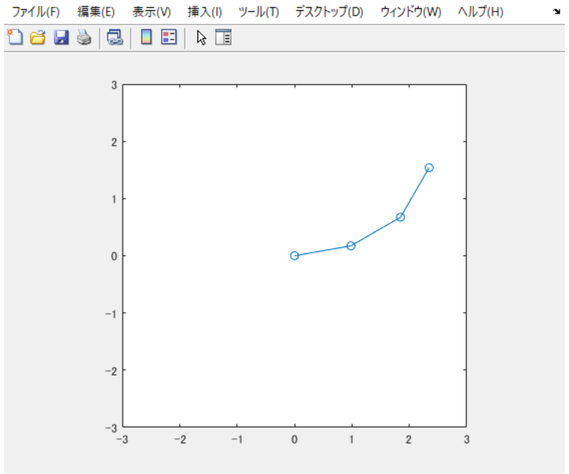


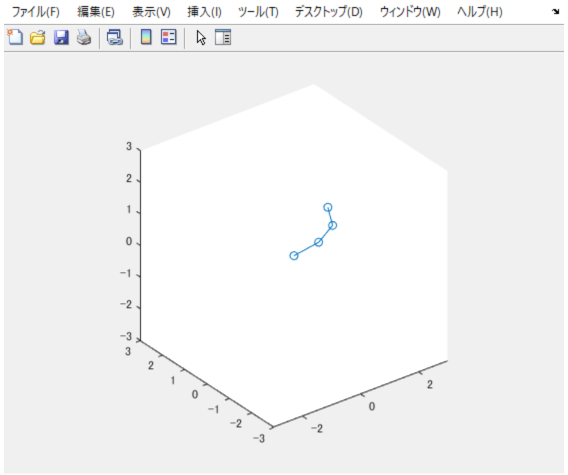


# Position Calculation

```
% a frame to base
T02 = T01 * T12;
T03 = T01 * T12 * T23;
T04 = T01 * T12 * T23 * T34;

% Points in base frame
p00 = [0; 0; 0; 1];
p01 = T01 * [0; 0; 0; 1];
p02 = T02 * [0; 0; 0; 1];
p03 = T03 * [0; 0; 0; 1];
p04 = T04 * [0; 0; 0; 1];
```



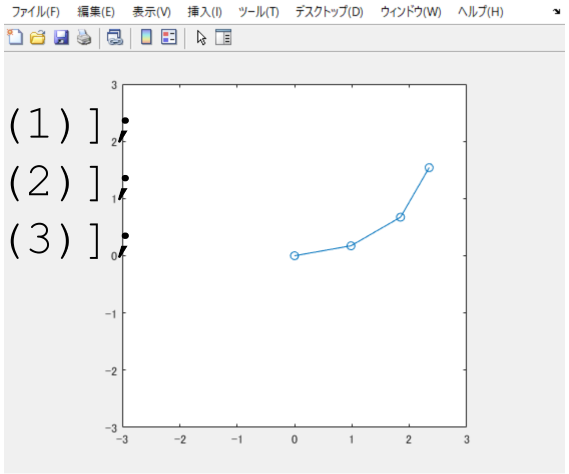


# Display an Arm Pose

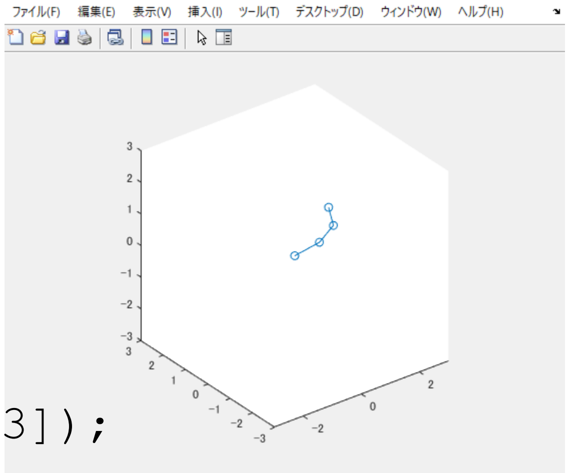
```
% Joint positions of robot pose
x = [p00(1), p01(1), p02(1), p03(1), p04(1)];
y = [p00(2), p01(2), p02(2), p03(2), p04(2)];
z = [p00(3), p01(3), p02(3), p03(3), p04(3)];

% 2D Plot
figure(1);
plot(x, y, '-o');
pbaspect([1, 1, 1]);
xlim([-3, 3]); ylim([-3,3]);

% 3D plot
figure(2);
plot3(x, y, z, '-o');
pbaspect([1, 1, 1]);
xlim([-3, 3]); ylim([-3, 3]); zlim([-3, 3]);
```



ファイル(F) 編集(E) 表示(V) 挿入(I) ツール(T) デスクトップ(D) ウィンドウ(W) ヘルプ(H)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3



ファイル(F) 編集(E) 表示(V) 挿入(I) ツール(T) デスクトップ(D) ウィンドウ(W) ヘルプ(H)
3
2
1
0
-1
-2
-3
3
2
1
0
-1
-2
-3
-2
0
2

# SAMPLEFKSEQUENCEPLANNER3LINK.M

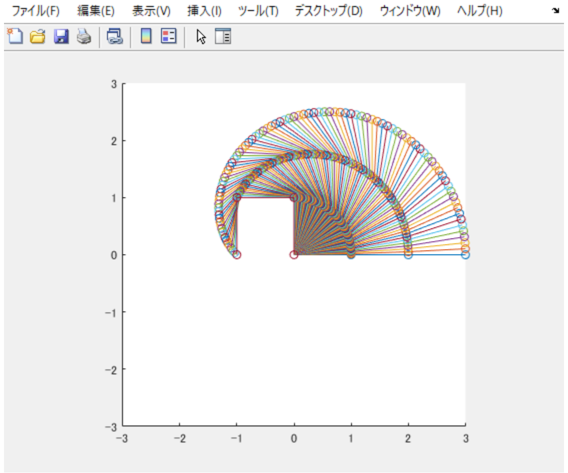
# Setup and Outside of Main loop

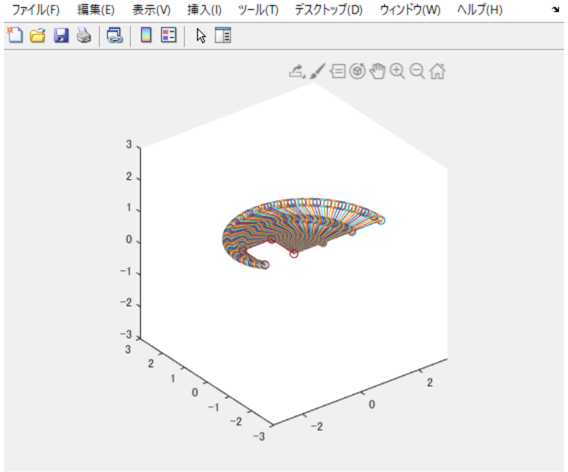
```
% Link lengths
L1 = 1; L2 = 1; L3 = 1;

% Joint angles
q1 = [0:1:90]; q2 = [0:1:90]; q3 = [0:1:90];

% Draw in 2D, Loop for joint angles
figure(1);
hold on
for i = 1:91
…
end
hold off

% Draw in 3D, Loop for joint angles
figure(2);
view(3);
hold on
for i = 1:91
…
end
hold off
```





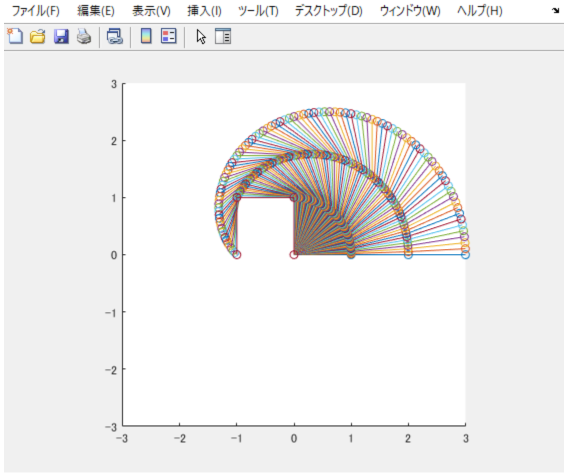
# Inside of Main Loop for 2D Plot

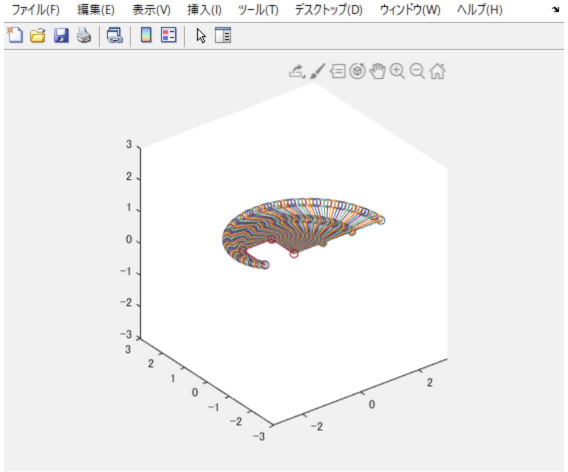
```
% a frame to previous frame
    T01 = DH(0,  0, 0, deg2rad( q1(i)) );
    T12 = DH(0, L1, 0, deg2rad( q2(i)) );
    T23 = DH(0, L2, 0, deg2rad( q3(i)) );
    T34 = DH(0, L3, 0, deg2rad(  0   ) );

    % a frame to base
    T02 = T01 * T12;
    T03 = T02 * T23;
    T04 = T03 * T34;

    % Points in base frame
    p00 = [0; 0; 0; 1];
    p01 = T01 * [0; 0; 0; 1];
    p02 = T02 * [0; 0; 0; 1];
    p03 = T03 * [0; 0; 0; 1];
    p04 = T04 * [0; 0; 0; 1];

    % Plot
    x = [p00(1), p01(1), p02(1), p03(1), p04(1)];
    y = [p00(2), p01(2), p02(2), p03(2), p04(2)];
    z = [p00(3), p01(3), p02(3), p03(3), p04(3)];
    plot(x, y, '-o');
    xlim([-3, 3]); ylim([-3, 3]);
    pbaspect([1 1 1]);
```





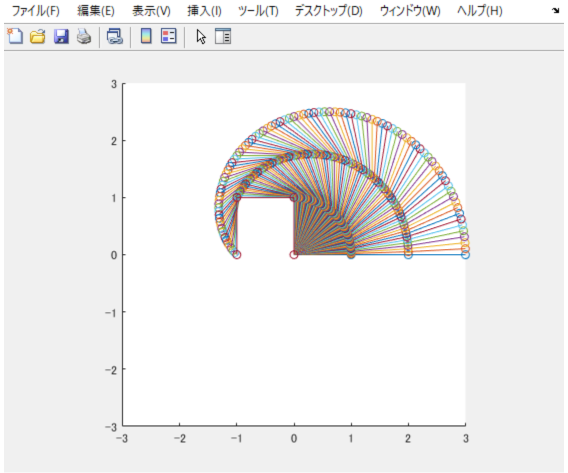
# Inside of Main Loop for 3D Plot

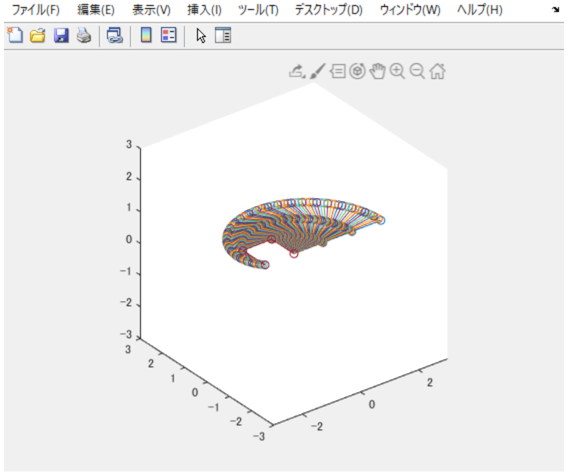
```
% a frame to previous frame
    T01 = DH(0,  0, 0, deg2rad( q1(i)) );
    T12 = DH(0, L1, 0, deg2rad( q2(i)) );
    T23 = DH(0, L2, 0, deg2rad( q3(i)) );
    T34 = DH(0, L3, 0, deg2rad(  0    ) );

    % a frame to base
    T02 = T01 * T12;
    T03 = T02 * T23;
    T04 = T03 * T34;

    % Points in base frame
    p00 = [0; 0; 0; 1];
    p01 = T01 * [0; 0; 0; 1];
    p02 = T02 * [0; 0; 0; 1];
    p03 = T03 * [0; 0; 0; 1];
    p04 = T04 * [0; 0; 0; 1];

    % Plot
    x = [p00(1), p01(1), p02(1), p03(1), p04(1)];
    y = [p00(2), p01(2), p02(2), p03(2), p04(2)];
    z = [p00(3), p01(3), p02(3), p03(3), p04(3)];
    plot3(x, y, z, '-o');
    xlim([-3, 3]); ylim([-3, 3]); zlim([-3, 3]);
    pbaspect([1 1 1]);
```





# Another Instance

Develop a matlab code for the robot arm with the following DH table.

| $i$ | $\alpha_{i-1}$ | $a_{i-1}$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | $\theta_1$ |
| 2 | 90 deg | $L_1$=1 | 0 | $\theta_2$ |
| 3 | 0 | $L_2$=1 | 0 | $\theta_3$ |
| 4 | 0 | $L_3$=1 | 0 | 0 |

# SAMPLEIKPLANER3LINK.M

# FK.m

```
function [fk1, fk2, fk3, fk4] = FK(q1, q2, q3)
%FK returns joint positions of 3-link planer robot

% Link lengths
L1 = 1; L2 = 1; L3 = 1;

% a frame to previous frame
T01 = DH(0,  0, 0, q1 );
T12 = DH(0, L1, 0, q2 );
T23 = DH(0, L2, 0, q3 );
T34 = DH(0, L3, 0, 0  );

% a frame to base
T02 = T01 * T12;
T03 = T01 * T12 * T23;
T04 = T01 * T12 * T23 * T34;

% Points in base frame
% Forward kinematics
p1 = T01 * [0; 0; 0; 1]; fk1 = p1(1:3);
p2 = T02 * [0; 0; 0; 1]; fk2 = p2(1:3);
p3 = T03 * [0; 0; 0; 1]; fk3 = p3(1:3);
p4 = T04 * [0; 0; 0; 1]; fk4 = p4(1:3);
end
```

# Find Jacobian

```
% Symbolic Jacobian
syms th1 th2 th3
[fk1, fk2, fk3, fk4] = FK(th1, th2, th3);
jacob4 = jacobian(fk4, [th1, th2, th3]);
```

# Iteration Setting and Variables

```
% Iteration times for inverse kinematics
tMax = 100; t = 1:tMax;

% Joint angles
q = zeros(3, tMax);
% Initial values
t = 1;
q(1, t) = deg2rad(10.0);
q(2, t) = deg2rad(20.0);
q(3, t) = deg2rad(30.0);

% Joint positions
p1 = zeros(3, tMax);
p2 = zeros(3, tMax);
p3 = zeros(3, tMax);
p4 = zeros(3, tMax);
```

# Inverse Kinematics by Iteration

```
% Target hand position
pd = [0.0; 2.0; 0.0];
% Weight
a = 0.1;
for t = 1:tMax-1
    % Forward kinamatics
    [p1(:,t), p2(:,t), p3(:,t), p4(:,t)] = FK(q(1, t), q(2,
t), q(3, t));
    % Give Jacobian a numerical value
    J = eval(subs(jacob4, [th1, th2, th3], [q(1, t), q(2,
t), q(3, t)]));
    % Find an error between
    q(:,t+1) = q(:,t) + a * pinv(J) * (pd - p4(:,t));
end
t = tMax;
[p1(:,t), p2(:,t), p3(:,t), p4(:,t)] = FK(q(1, t), q(2, t),
q(3, t));
```

# 2D Plot

```
figure(1);
hold on
for t = 1 : tMax
    % Joint positions of robot pose
    x = [p1(1, t), p2(1, t), p3(1, t), p4(1, t)];
    y = [p1(2, t), p2(2, t), p3(2, t), p4(2, t)];
    plot(x, y, '-o');
    pbaspect([1, 1, 1]); xlim([-3, 3]); ylim([-3,3]);
end
hold off;
```

# 3D Plot

```
% 3D Plot
figure(2);
view(3);
hold on
for t = 1 : tMax
    % Joint positions of robot pose
    x = [p1(1, t), p2(1, t), p3(1, t), p4(1, t)];
    y = [p1(2, t), p2(2, t), p3(2, t), p4(2, t)];
    z = [p1(3, t), p2(3, t), p3(3, t), p4(2, t)];
    plot(x, y, '-o');
    pbaspect([1, 1, 1]); xlim([-3, 3]); ylim([-3,3]);
end
hold off;
```