```
close all
clear
clc

% PROBLEM 2 - 2.2

% create figure
figure
axis([-6, 6, -6, 6])
grid on
hold on

% save as a video file
v = VideoWriter('Problem3_2.mp4', 'MPEG-4');
v.FrameRate = 3;
open(v);

epsilon = 0.85;
%initial joint values
theta = [pi/8; pi/8; pi/8; pi/8; pi/8];
L = 1;

omega = [0;0;1];

q1 = [0;0;0];
q2 = [L;0;0];
q3 = [2*L;0;0];
q4 = [3*L;0;0];
q5 = [4*L;0;0];

S1 = [omega; -cross(omega, q1)];
S2 = [omega; -cross(omega, q2)];
S3 = [omega; -cross(omega, q3)];
S4 = [omega; -cross(omega, q4)];
S5 = [omega; -cross(omega, q5)];

S_eq = [S1, S2, S3, S4, S5];
M = [eye(3), [5*L;0;0]; 0 0 0 1];

% T with initial joint positions
T_0 = fk(M, S_eq, theta)
```

```
T_0 = 4×4
   -0.3827   -0.9239         0    1.6310
    0.9239   -0.3827         0    3.9375
         0         0    1.0000         0
         0         0         0    1.0000
```

```
R_0 = T_0(1:3, 1:3);
JS = double(JacS(S_eq, theta)) %Space Jacobian
```

```
JS = 6×5
         0         0         0         0         0
         0         0         0         0         0
    1.0000    1.0000    1.0000    1.0000    1.0000
         0    0.3827    1.0898    2.0137    3.0137
         0   -0.9239   -1.6310   -2.0137   -2.0137
         0         0         0         0         0
```

```
Jb = double(adjointM(inv(T_0))*JS) %Body Jacobian
```

```
Jb = 6×5
         0         0         0         0         0
         0         0         0         0         0
    1.0000    1.0000    1.0000    1.0000    1.0000
    3.0137    2.0137    1.0898    0.3827         0
    3.0137    3.0137    2.6310    1.9239    1.0000
         0         0         0         0         0
```

```
J_geometric = double([R_0, zeros(3); zeros(3), R_0] * Jb) %Geometric Jacobian
```

```
J_geometric = 6×5
         0         0         0         0         0
         0         0         0         0         0
    1.0000    1.0000    1.0000    1.0000    1.0000
   -3.9375   -3.5549   -2.8478   -1.9239   -0.9239
    1.6310    0.7071   -0.0000   -0.3827   -0.3827
         0         0         0         0         0
```

```
X = [r2axisangle(R_0);T_0(1:3,4)]
```

```
X = 6×1
         0
         0
    1.9635
    1.6310
    3.9375
         0
```

```
% Problem part 2.2
% Given desired Transformation matrices T_d
T_d = [rotz(pi/2), [-2;4;0]; 0 0 0 1]
```

```
T_d = 4×4
    0.9996   -0.0274         0   -2.0000
    0.0274    0.9996         0    4.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

```
R_d = T_d(1:3, 1:3);
Xd = [r2axisangle(R_d);T_d(1:3,4)]
```

```
Xd = 6×1
         0
         0
    0.0274
   -2.0000
    4.0000
         0
```

```
V = Xd - X
```

```
V = 6×1
         0
         0
   -1.9361
   -3.6310
    0.0625
         0
```

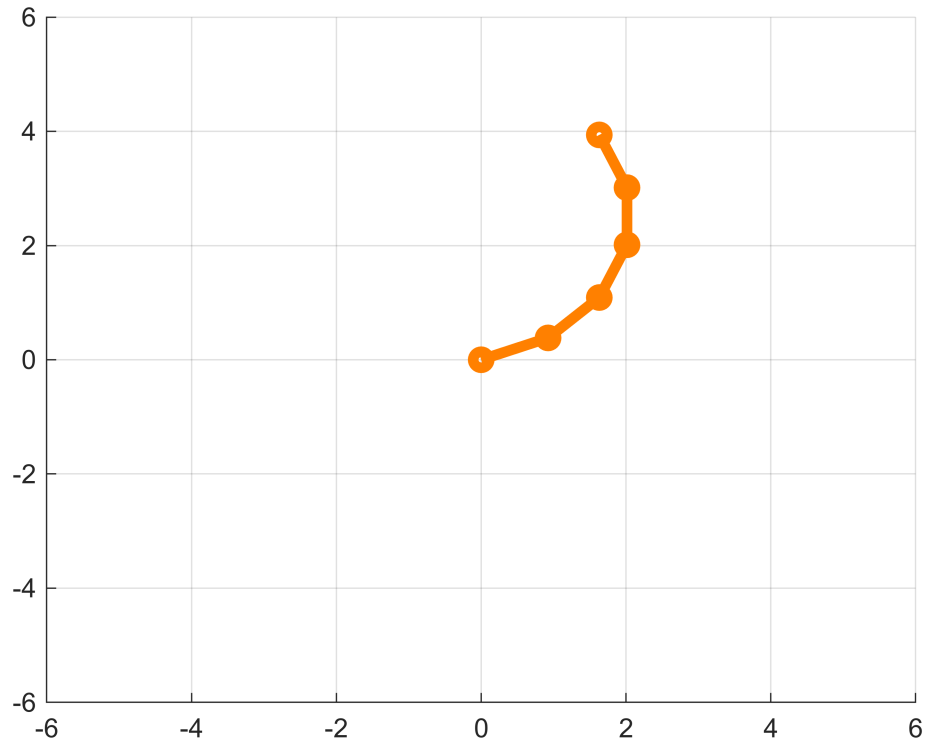
```
while norm(Xd - X) > epsilon
% plot the robot
% 1. get the position of each link
    p0 = [0; 0];
    p1 = [L*cos(theta(1)); L*sin(theta(1))]; % (x,y) position of end of first link
    p2 = [L*cos(theta(1) + theta(2)) + p1(1); L*sin(theta(1) + theta(2)) + p1(2)];
% (x,y) position of end of second link
    p3 = [L*cos(theta(1) + theta(2) + theta(3)) + p2(1); L*sin(theta(1) + theta(2)
+ theta(3)) + p2(2)]; % (x,y) position of end of third link
    p4 = [L*cos(theta(1) + theta(2) + theta(3) + theta(4)) + p3(1); L*sin(theta(1)
+ theta(2) + theta(3) + theta(4)) + p3(2)]; % (x,y) position of end of fourth link
    p_v = [L*cos(theta(1) + theta(2) + theta(3) + theta(4) + theta(5)) + p4(1);
L*sin(theta(1) + theta(2) + theta(3) + theta(4) + theta(5)) + p4(2)];% (x,y)
position of end-effector
    P_v = [p0, p1, p2, p3, p4, p_v];
% 2. draw the robot and save the frame

    cla;
    plot(P_v(1,:), P_v(2,:), 'o-', 'color',[1, 0.5, 0],'linewidth',4)
    drawnow
    frame = getframe(gcf);
    writeVideo(v, frame);

% your code here
    V = Xd - X;
    JS = double(JacS(S_eq, theta)); % Updated Space Jacobian
    Jb = double(adjointM(inv(T_0))*JS); %Updated Body Jacobian
    J_geometric = double([R_0, zeros(3); zeros(3), R_0] * Jb); %Updated Geometric
Jacobian
    delta_theta = double(pinv(J_geometric)*V +(eye(5) -
pinv(J_geometric)*J_geometric)*[0;0;0;0;0]) %null space is zero currently as we set
b = 0

    %Updating theta until the while loop is satisfied to get the desired inverse
kinematics (joint positions), thus simulating the robot
    theta = double(0.1 * delta_theta + theta)
    T_0 = fk(M, S_eq, theta)
    R_0 = T_0(1:3, 1:3);
    X = [r2axisangle(R_0);T_0(1:3,4)];

end
```




```
Warning: The video's width and height has been padded to be a multiple of two as required by the H.264 codec.
delta_theta = 5×1
   -2.0384
    2.1093
    2.9462
    0.3450
   -5.2982
theta = 5×1
    0.1889
    0.6036
    0.6873
    0.4272
   -0.1371
T_0 = 4×4
   -0.1978   -0.9802         0    1.2475
    0.9802   -0.1978         0    3.8200
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
   -0.4784
    1.4861
    1.5218
   -0.6137
   -3.6584
theta = 5×1
    0.1410
    0.7522
    0.8395
    0.3658
   -0.5030
T_0 = 4×4
   -0.0248   -0.9997         0    0.9272
    0.9997   -0.0248         0    3.7702
         0         0    1.0000         0
```

```
         0         0         0    1.0000
delta_theta = 5×1
    0.2709
    1.2079
    0.7480
   -0.9455
   -2.8496
theta = 5×1
    0.1681
    0.8730
    0.9143
    0.2713
   -0.7879
T_0 = 4×4
    0.1316   -0.9913         0    0.6376
    0.9913    0.1316         0    3.7410
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    0.7827
    0.9613
    0.1475
   -1.0744
   -2.2285
theta = 5×1
    0.2464
    0.9692
    0.9290
    0.1638
   -1.0108
T_0 = 4×4
    0.2697   -0.9629         0    0.3720
    0.9629    0.2697         0    3.7243
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.1322
    0.7033
   -0.3194
   -1.0773
   -1.7091
theta = 5×1
    0.3596
    1.0395
    0.8971
    0.0561
   -1.1817
T_0 = 4×4
    0.3896   -0.9210         0    0.1286
    0.9210    0.3896         0    3.7163
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.3558
    0.4326
   -0.6561
   -0.9983
   -1.2772
theta = 5×1
    0.4952
    1.0827
    0.8315
   -0.0437
   -1.3094
```

```
T_0 = 4×4
    0.4921   -0.8706         0   -0.0927
    0.8706    0.4921         0    3.7145
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.4869
    0.1617
   -0.8752
   -0.8720
   -0.9302
theta = 5×1
    0.6439
    1.0989
    0.7440
   -0.1309
   -1.4024
T_0 = 4×4
    0.5789   -0.8154         0   -0.2925
    0.8154    0.5789         0    3.7173
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.5556
   -0.0967
   -1.0011
   -0.7229
   -0.6609
theta = 5×1
    0.7994
    1.0892
    0.6439
   -0.2032
   -1.4685
T_0 = 4×4
    0.6518   -0.7584         0   -0.4717
    0.7584    0.6518         0    3.7232
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.5873
   -0.3369
   -1.0617
   -0.5655
   -0.4566
theta = 5×1
    0.9582
    1.0556
    0.5377
   -0.2598
   -1.5142
T_0 = 4×4
    0.7127   -0.7015         0   -0.6318
    0.7015    0.7127         0    3.7309
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.6029
   -0.5617
   -1.0819
   -0.4055
   -0.3040
theta = 5×1
    1.1185
```

```
    0.9994
    0.4295
   -0.3003
   -1.5446
T_0 = 4×4
    0.7632   -0.6461         0   -0.7743
    0.6461    0.7632         0    3.7394
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.6214
   -0.7832
   -1.0808
   -0.2406
   -0.1920
theta = 5×1
    1.2806
    0.9211
    0.3214
   -0.3244
   -1.5638
T_0 = 4×4
    0.8051   -0.5932         0   -0.9009
    0.5932    0.8051         0    3.7479
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.6630
   -1.0249
   -1.0710
   -0.0594
   -0.1153
theta = 5×1
    1.4469
    0.8186
    0.2143
   -0.3303
   -1.5753
T_0 = 4×4
    0.8396   -0.5432         0   -1.0128
    0.5432    0.8396         0    3.7554
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.7554
   -1.3331
   -1.0573
    0.1668
   -0.0786
theta = 5×1
    1.6225
    0.6853
    0.1086
   -0.3136
   -1.5832
T_0 = 4×4
    0.8680   -0.4965         0   -1.1111
    0.4965    0.8680         0    3.7609
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    1.9510
   -1.8177
   -1.0240
```

```
    0.5158
   -0.1172
theta = 5×1
    1.8176
    0.5035
    0.0062
   -0.2620
   -1.5949
T_0 = 4×4
    0.8914   -0.4532         0   -1.1955
    0.4532    0.8914         0    3.7619
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    2.3703
   -2.8380
   -0.8243
    1.2789
   -0.4298
theta = 5×1
    2.0546
    0.2197
   -0.0762
   -0.1342
   -1.6379
T_0 = 4×4
    0.9106   -0.4133         0   -1.2617
    0.4133    0.9106         0    3.7516
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
    2.4824
   -6.0518
    2.0930
    4.4540
   -3.3763
theta = 5×1
    2.3028
   -0.3855
    0.1331
    0.3112
   -1.9755
T_0 = 4×4
    0.9264   -0.3766         0   -1.2541
    0.3766    0.9264         0    3.6514
         0         0    1.0000         0
         0         0         0    1.0000
delta_theta = 5×1
   -0.5774
    2.6534
   -1.2700
   -2.7358
    1.5711
theta = 5×1
    2.2451
   -0.1201
    0.0061
    0.0377
   -1.8184
T_0 = 4×4
    0.9393   -0.3432         0   -1.3056
    0.3432    0.9393         0    3.6483
         0         0    1.0000         0
         0         0         0    1.0000
```

```
close(v);
close all
```