```
robot=FunctionHolder;
a1=robot.a1;a2=robot.a2;a3=robot.a3; a4=robot.a4;
realpathPoints=robot.realpathPoints;
OrientZ=robot.OrientZ;
%% shift of coordinates from end effector to base of Link4
pathPoints=realpathPoints;
for j=1:size(realpathPoints,2)
    pathPoints(1,j)=realpathPoints(1,j)+a4*sin(OrientZ(j));
    pathPoints(2,j)=realpathPoints(2,j)-a4*cos(OrientZ(j));
end
%% Konwn Transformation matrix from end effector to base of robot
Te0=cell(1,size(realpathPoints,2));
for i=1:size(pathPoints,2)
    Te0{i}=[cos(OrientZ(1,i)) -sin(OrientZ(1,i)) 0 pathPoints(1,i);sin(OrientZ(1,i)) ↙
cos(OrientZ(1,i)) 0 pathPoints(2,i);
        0 0 0 pathPoints(3,i);0 0 0 1];
end
velPathPoints=[0 0;0 0;0 0];
%% Trajectory Planning
%From Cartesian coordinates to Joint variable using inverse kinematics
JointVariables=zeros(5,size(pathPoints,2));
JointVariablesdot=zeros(5,size(pathPoints,2));
for i=1:size(pathPoints,2)
    I=robot.InverseKinematics(Te0{i},a1,a2,a3);
    JointVariables(1,i)=I(1,1);
    JointVariables(2,i)=I(1,2);
    JointVariables(3,i)=I(1,3);
    JointVariables(4,i)=I(1,4);
    JointVariables(5,i)=OrientZ(i);   %known orientation of end effector wrt base
end

%% Cubic Polynomial trajectory generation in joint space
T=[0 3] ;                                  % TIme elapsed
t=0:0.05:T(size(T,2));
NoOfJoints=5;                              %joint 5 is not the joint but the ↙
orientation of end effector wrt base.
NoOfCubicPoly=size(pathPoints,2)-1;
coefficentOfPoly=cell(1,NoOfJoints);
theta=cell(1,NoOfJoints);
coefficientMatrix=cell(NoOfJoints,NoOfCubicPoly);
for j=1:NoOfJoints
    for k=1:NoOfCubicPoly
        C1=(robot.CoefficientsOfCubic(T(k),T(k+1),JointVariables(j,k),JointVariables ↙
(j,k+1),JointVariablesdot(j,k),JointVariablesdot(j,k+1)));
        coefficentOfPoly{j}(1,k)=C1(1,1);
        coefficentOfPoly{j}(2,k)=C1(2,1);
        coefficentOfPoly{j}(3,k)=C1(3,1);
        coefficentOfPoly{j}(4,k)=C1(4,1);
        coefficientMatrix{j,k}=C1;
```

```
    end
end

for k=1:NoOfCubicPoly
    for j=1:NoOfJoints
        for i=1:length(t)
            if t(i)<=T(k+1) && t(i)>=T(k)
                theta{j}(i)=double(coefficientMatrix{j,k}(1,1)+coefficientMatrix{j,k} ↙
(2,1)*t(i)+coefficientMatrix{j,k}(3,1)*t(i)^2+coefficientMatrix{j,k}(4,1)*t(i)^3);
            end
        end
    end
end


%% Main Code
% Forward Kinematics
T10=cell(1,size(t,2));T21=cell(1,size(t,2));T32=cell(1,size(t,2));T43=cell(1,size(t, ↙
2));T20=cell(1,size(t,2));
T30=cell(1,size(t,2));T40=cell(1,size(t,2));xm=zeros(1,size(t,2));ym=zeros(1,size(t, ↙
2));zm=zeros(1,size(t,2));
for j=1:length(t)
    T10{j}=[cos(theta{1}(j)) -sin(theta{1}(j)) 0 0;sin(theta{1}(j)) cos(theta{1}(j)) ↙
0 0;0 0 1 a1;0 0 0 1];
    T21{j}=[1 0 0 a2;0 1 0 0;0 0 1 theta{2}(j);0 0 0 1];
    T32{j}=[cos(theta{3}(j)) -sin(theta{3}(j)) 0 a3*cos(theta{3}(j));sin(theta{3}(j)) ↙
cos(theta{3}(j)) 0 a3*sin(theta{3}(j));0 0 1 0;0 0 0 1];
    T43{j}=[cos(theta{4}(j)) -sin(theta{4}(j)) 0 0;sin(theta{4}(j)) cos(theta{4}(j)) ↙
0 0;0 0 1 0;0 0 0 1];
    T20{j}=T10{j}*T21{j};
    T30{j}=T20{j}*T32{j};
    %modified end effector position
    xm(j)=T30{j}(1,4)-a4*sin(theta{5}(j));
    ym(j)=T30{j}(2,4)+a4*cos(theta{5}(j));
    zm(j)=T30{j}(3,4);
end
%% Animation
plot3([0 0],[0 0],[0 0],'Color',[0.5 0.5 0],'LineWidth',2); %Link1
axis([-1.5 1.5 -1.5 1.5 0 3]);
xlabel('x-axis','FontSize',10,'Color',[1,0,0]);
ylabel('y-axis','FontSize',10,'Color',[1,0,0]);
zlabel('z-axis','FontSize',10,'Color',[1,0,0]);
grid on
hold on
h1=plot3([0 0],[0 0],[0 0],'Color',[1 0 0],'LineWidth',2,'DisplayName','Link1'); %↙
link1
h2=plot3([0 0],[0 0],[0 0],'Color',[0 1 0],'LineWidth',2,'DisplayName','Link2'); %↙
link2
h3=plot3([0 0],[0 0],[0 0],'Color',[0 0 1],'LineWidth',2,'DisplayName','Link3'); %↙
```

```
link3
h4=plot3([0 0],[0 0],[0 0],'Color',[0 0 0.5],'LineWidth',2,'DisplayName','Link4'); %↙
link4
h5=plot3([realpathPoints(1,1) realpathPoints(1,2)],[realpathPoints(2,1) ↙
realpathPoints(2,2)],[realpathPoints(3,1) realpathPoints(3,2)], ...
    '--','Color',[0.5 0 0.5],'LineWidth',2,'DisplayName','Line btw start and end'); %↙
straight trajectory
h6=plot3(xm(1),ym(1),zm(1),'--','Color',[1 0 0.5],'LineWidth',↙
2,'DisplayName','EndeffectorPosition'); %end effector position
legend;
i=0;
while i<=3
    for j=(2:length(t))
        set(h1,'XData',[0 T10{j}(1,4)],'YData',[0 T10{j}(2,4)],'ZData',[0 T20{j}↙
(3,4)]);
        set(h2,'XData',[T10{j}(1,4) T20{j}(1,4)],'YData',[T10{j}(2,4) T20{j}↙
(2,4)],'ZData',[T20{j}(3,4) T20{j}(3,4)]);
        set(h3,'XData',[T20{j}(1,4) T30{j}(1,4)],'YData',[T20{j}(2,4) T30{j}↙
(2,4)],'ZData',[T20{j}(3,4) T30{j}(3,4)]);
        set(h4,'XData',[T30{j}(1,4) xm(j)],'YData',[T30{j}(2,4) ym(j)],'ZData',[T30↙
{j}(3,4) zm(j)]);
        set(h6,'XData',xm(1:j),'YData',ym(1:j),'ZData',zm(1:j));
        pause(0.1);
    end
end
```