

```

% Direct Kinematics of an n degrees open chain manipulator
function [g_base_tool] = manipdkin(gst0, axis_joints, q_joints, type_joints, theta)
I = eye(3); % assign a Identity matrix.
[row,column] = size(type_joints);
dof = row; % D.O.F is equal to the number of rows
% initialize gst(theta).
gst_theta = gst0;
for n=dof:-1:1
    % assign points along axis of rotation
    q_i = q_joints(:,n);
    % assign the axis of rotation and theta (angle or displacement) values for convenience.
    axis1_i = axis_joints(1,n);
    axis2_i = axis_joints(2,n);
    axis3_i = axis_joints(3,n);
    theta_i = theta(n,1);

    if type_joints(n)=="R" %check if it is a revolute joint
        % assign the axis of rotation
        w1_i = axis1_i;
        w2_i = axis2_i;
        w3_i = axis3_i;
        w = [w1_i;w2_i;w3_i];
        % convert axis-angle representation to rotation matrix.
        Rot_i = AxisAngle_to_Rot(w,theta_i);
        % element(1,2) in g
        P_i = (I-Rot_i)*q_i;
        % compute matrix T for revolute joints
        T = [Rot_i,P_i;[0 0 0],1];
    else %if it is not revolute joint then it must be a prismatic joint
        % assign axis of displacement
        v = [axis1_i;axis2_i;axis3_i];
        T = [I,v*theta_i;[0 0 0],1];
    end
    % compute gst(theta) iteratively
    gst_theta = T*gst_theta;
end

% return computed gst(theta)
g_base_tool = gst_theta;

end

```

Not enough input arguments.

Error in manipdkin (line 4)
 [row,column] = size(type_joints);

