```
% Compute the Spatial Jacobian of a given manipulator.
function Js = SpatialmanipJac(axis_joints, q_joints, type_joints,configs)
% get degree of freedom
[dof,column] = size(type_joints);
% initialize g for later use in iteration.
gTotal = eye(4);

for i = 1:dof
    % distinguish the types of joints
    if type_joints(i) == "R"  % revolute joint case
        xi = [cross(-1*axis_joints(:,i),q_joints(:,i)); axis_joints(:,i)];
        Rot_i = AxisAngle_to_Rot(axis_joints(:,i),configs(i));
        P_i = (eye(3)-Rot_i)*q_joints(:,i);
        T = [Rot_i,P_i;[0 0 0], 1];
    else % pristmatic joint case
        xi = [axis_joints(:,i);[0;0;0]];
        T = [eye(3),axis_joints(:,i)*configs(i);[0 0 0],1];
    end

    % modify xi when i > 1
    if i == 1
        % assign the xi to Jacobian matrix only when i = 1.
        Js(:,i) = xi;

    else
        % calculate g(1,i-1).
        gTotal = gTotal*Tprev;
        % Adjoint matrix conversion
        Adg = AdjointOfg(gTotal);
        % calculate the elements of Jacobian when i > 1.
        Js(:,i) = Adg*xi;
    end

    % store current Transformation matrix for the next iteration.
    Tprev = T;



end
```

```
Not enough input arguments.

Error in SpatialmanipJac (line 4)
[dof,column] = size(type_joints);
```

Published with MATLAB® R2018b