Contents

MATLAB_Exercise_2

Michael White 3/26/2021

```
clear; clc; close all;
```

Part A "by hand"

```
syms L1 L2 L3 cphi sphi x y;

% Generate known transforms in symbolic form (flip signs of sines for
% inputs to be the same)
tforms.T_3H = [1 0 0 L3; 0 1 0 0; 0 0 1 0; 0 0 0 1];
tforms.T_0H = [cphi sphi 0 x; -sphi cphi 0 y; 0 0 1 0; 0 0 0 1];

% Calculate theta2 using equations 4.14-4.16 from textbook
tforms.T_03 = tforms.T_0H/tforms.T_3H;
c2 = (tforms.T_03(1,4)^2+tforms.T_03(2,4)^2-L1^2-L2^2)/(2*L1*L2);
s2 = sqrt(1-c2^2);
theta2 = atan2(s2,c2);

% Calculate theta1 from equations 4.19 and 4.27 in the textbook
k1 = L1+c2*L2;
k2 = L2*s2;
theta1 = atan2(tforms.T_03(2,4),tforms.T_03(1,4))-atan2(k2,k1);

% Calculate theta3 from equation 4.28 in the textbook
phi = -atan2(sphi,cphi);
theta3 = phi - theta1 - theta2;

% Display symbolic results
disp("Theta 1 = "); disp(theta1);
disp("Theta 2 = "); disp(theta2);
disp("Theta 3 = "); disp(theta3);
```

```
Theta 1 =
atan2(y + L3*sphi, x - L3*cphi) - atan2(L2*(1 - ((x - L3*cphi)^2 + (y + L3*sphi)^2 - L1^2 - L2^2)^2/(4*L1^2*L2^2))^(1/2), L1 + ((x - L3*cphi)^2 + (y + L3*sphi)^2 -

Theta 2 =
atan2(2*(1 - ((x - L3*cphi)^2 + (y + L3*sphi)^2 - L1^2 - L2^2)^2/(4*L1^2*L2^2))^(1/2), ((x - L3*cphi)^2 + (y + L3*sphi)^2 - L1^2 - L2^2)/(L1*L2))

Theta 3 =
- atan2(y + L3*sphi, x - L3*cphi) + atan2(L2*(1 - ((x - L3*cphi)^2 + (y + L3*sphi)^2 - L1^2 - L2^2)^2/(4*L1^2*L2^2))^(1/2), L1 + ((x - L3*cphi)^2 + (y + L3*sphi)^2
```

Part B: Testing Examples

Generate transform matrices for each example

```
examples.i.T_0H = [1 0 0 9; 0 1 0 0; 0 0 1 0; 0 0 0 1];
examples.ii.T_0H = [0.5 -0.866 0 7.5373; 0.866 0.5 0 3.9266; 0 0 1 0; 0 0 0 1];
examples.iii.T_0H = [0 1 0 -3; -1 0 0 2; 0 0 1 0; 0 0 0 1];
examples.iv.T_0H = [0.866 0.5 0 -3.1245; -0.5 0.866 0 9.1674; 0 0 1 0; 0 0 0 1];
nameList = {'i' 'ii' 'iii' 'iv'};

for i = 1:length(nameList)
    %Check to see if points are even in reach of the manipulator
    totalDistance = sqrt(examples.(char(nameList(i))).T_0H(1,4)^2 +...
        examples.(char(nameList(i))).T_0H(2,4)^2);
    if totalDistance > 9
        disp(strcat("Example ",char(nameList(i))," is out of reach"));
    else
        % Calculate values for each theta from sym eqns with subs
        examples.(char(nameList(i))).theta_rad(1) = double(subs(theta1,...
            [sphi cphi x y L1 L2 L3],...
            [examples.(char(nameList(i))).T_0H(1,2) examples.(char(nameList(i))).T_0H(1,1)...
            examples.(char(nameList(i))).T_0H(1,4) examples.(char(nameList(i))).T_0H(2,4) 4 3 2]));
        examples.(char(nameList(i))).theta_rad(2) = double(subs(theta2,...
            [sphi cphi x y L1 L2 L3],...
            [examples.(char(nameList(i))).T_0H(1,2) examples.(char(nameList(i))).T_0H(1,1)...
            examples.(char(nameList(i))).T_0H(1,4) examples.(char(nameList(i))).T_0H(2,4) 4 3 2]));
        examples.(char(nameList(i))).theta_rad(3) = double(subs(theta3,...
            [sphi cphi x y L1 L2 L3],...
            [examples.(char(nameList(i))).T_0H(1,2) examples.(char(nameList(i))).T_0H(1,1)...
            examples.(char(nameList(i))).T_0H(1,4) examples.(char(nameList(i))).T_0H(2,4) 4 3 2]));
```

```
            examples.(char(nameList(i))).theta_deg = 180/pi*examples.(char(nameList(i))).theta_rad;

            % The third angle in the third example was computing as
            % negative, so I added this check
            for j = 1:3
                if examples.(char(nameList(i))).theta_deg(j) < 0
                    examples.(char(nameList(i))).theta_deg(j) = ...
                        examples.(char(nameList(i))).theta_deg(j)+360;
                end
            end

            % Display results to command window
            disp(strcat("Joint angles for example ",char(nameList(i))));
            disp(examples.(char(nameList(i))).theta_deg);
        end

        % Setup example transforms for validation part
        examples.(char(nameList(i))).T_3H = double(subs(tforms.T_3H,...
            [sphi cphi x y L1 L2 L3],...
            [examples.(char(nameList(i))).T_0H(1,2) examples.(char(nameList(i))).T_0H(1,1)...
            examples.(char(nameList(i))).T_0H(1,4) examples.(char(nameList(i))).T_0H(2,4) 4 3 2]));
        examples.(char(nameList(i))).T_03 = examples.(char(nameList(i))).T_0H...
            /examples.(char(nameList(i))).T_3H;
    end
```

```
Joint angles for example i
     0     0     0

Joint angles for example ii
    9.9999   20.0004   29.9989

Joint angles for example iii
    90    90    90

Example iv is out of reach
```

Part C: Validating with Corke

Generate robot from link parameters

```
    L(1) = Link([0 0 4 0],'mod');
    L(2) = Link([0 0 3 0],'mod');
    L(3) = Link([0 0 0 0],'mod');
    robot = SerialLink(L);

    % Calculate thetas from inverse kinematics of example transforms
    corke.i.theta_rad = robot.ikine(examples.i.T_03,[0 0 0],'mask',[1 1 0 0 0 1]);
    corke.ii.theta_rad = robot.ikine(examples.ii.T_03,[0 0 0],'mask',[1 1 0 0 0 1]);
    corke.iii.theta_rad = robot.ikine(examples.iii.T_03,[90 90 -270]*pi/180,'mask',[1 -1 0 0 0 1]);
    corke.iv.theta_rad = robot.ikine(examples.iv.T_03,[0 0 0],'mask',[1 1 0 0 0 1]);
    % The example iv does NOT converge, which matches our calculated result

    disp('Results from Corke toolbox:');

    for i = 1:length(nameList)
        if ~isempty(corke.(char(nameList(i))).theta_rad)
            corke.(char(nameList(i))).theta_deg = corke.(char(nameList(i))).theta_rad*180/pi;
            disp(strcat("The Corke joint angles for example ",char(nameList(i)),':'));
            disp(corke.(char(nameList(i))).theta_deg);
        else
            disp(strcat("No valid joint angles for example ",char(nameList(i))));
        end
    end
```

```
Warning: ikine: iteration limit 500 exceeded (pose 1), final err 4.26772
Warning: failed to converge: try a different initial value of joint coordinates
Results from Corke toolbox:
The Corke joint angles for example i:
     0     0     0

The Corke joint angles for example ii:
    9.9999   20.0004   29.9989

The Corke joint angles for example iii:
   90.0000   90.0000   90.0000

No valid joint angles for example iv
```

Published with MATLAB® R2020b