Robotics Lab 09

Syeda Manahil Wasti, Syed Mujtaba Hassan, Aatiqa Khalid

May 14, 2023

1. Determine the conditions at which the rank of Jacobian drops below its maximal rank and the corresponding singular configurations.

   - We'll choose the origin of our end-effector frame at the same location as the origin of frame 3, i.e. $o_4 = o_3$. This will effectively set $a_4 = 0$ in our DH parameters and the Jacobian simplifies significantly.
   - You can use the function subs to substitute a4 = 0, e.g. if the Jacobian is stored in J, then we use subs(J(t),'a_4',0).
   - Looking at the Jacobian, you would realize that you can find the rank by finding the determinant of the top left 3 × 3 block of Jacobian matrix. The MATLAB function for determinant is det.
   - Remember to make judicious use of simplify and expand.

   (a) Show that the Jacobian loses rank when:
   $a_2 a_3 \sin\theta_3[a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3)] = 0$.
   This is the same condition as the one obtained for RRR arm in your book.

Solution:

```
syms('theta_1','theta2','theta3', 'theta4', 'a1', 'a2', 'a3', 'a4', 'd1',
    'd2', 'd3', 'd4')

alpha1 = pi/2;
alpha2 = 0;
alpha3 = 0;
alpha4 = 0;

A1 = [cos(theta_1), -sin(theta_1)*cos(alpha1), sin(theta_1)*sin(alpha1),
    a1*cos(theta_1);
      sin(theta_1), cos(theta_1)*cos(alpha1), -cos(theta_1)*sin(alpha1),
         a1*sin(theta_1);
      0, sin(alpha1), cos(alpha1), d1;
      0, 0, 0, 1];

A2 = [cos(theta2), -sin(theta2)*cos(alpha2), sin(theta2)*sin(alpha2), a2*
    cos(theta2);
      sin(theta2), cos(theta2)*cos(alpha2), -cos(theta2)*sin(alpha2), a2*
         sin(theta2);
      0, sin(alpha2), cos(alpha2), d2;
      0, 0, 0, 1];

A3 = [cos(theta3), -sin(theta3)*cos(alpha3), sin(theta3)*sin(alpha3), a3*
    cos(theta3);
      sin(theta3), cos(theta3)*cos(alpha3), -cos(theta3)*sin(alpha3), a3*
         sin(theta3);
      0, sin(alpha3), cos(alpha3), d3;
      0, 0, 0, 1];
```

```
A4 = [cos(theta4), -sin(theta4)*cos(alpha4), sin(theta4)*sin(alpha4), a4*
    cos(theta4);
      sin(theta4), cos(theta4)*cos(alpha4), -cos(theta4)*sin(alpha4), a4*
         sin(theta4);
      0, sin(alpha4), cos(alpha4), d4;
      0, 0, 0, 1];

T_01 = simplify(A1);
T_02 = simplify(A1*A2);
T_03 = simplify(A1*A2*A3);
T_04 = simplify(A1*A2*A3*A4);

o0 = [0; 0; 0];
o1 = T_01(1:3,end);
o2 = T_02(1:3,end);
o3 = T_03(1:3,end);
o4 = T_04(1:3,end);

z0 = simplify(T_01(1:3,3));
z1 = simplify(T_01(1:3,3));
z2 = simplify(T_02(1:3,3));
z3 = simplify(T_03(1:3,3));

P1 = simplify(T_04(1:3,4));
P2 = vpa(P1,2);
P = simplify(P2);

dx = simplify(P(1,1));
dy = simplify(P(2,1));
dz = simplify(P(3,1));

Jv = [diff(dx, theta_1), diff(dx, theta2), diff(dx, theta3), diff(dx,
    theta4);
        diff(dy, theta_1), diff(dy, theta2), diff(dy, theta3), diff(dy,
           theta4);
        diff(dz, theta_1), diff(dz, theta2), diff(dz, theta3), diff(dz,
           theta4)];
Jv = simplify(Jv);

Jw = [z0 z1 z2 o0];
Jw = simplify(Jw);

J = [simplify(Jv); simplify(Jw)];
J = simplify(J);

J_s =  simplify(subs(J,'a4',0));

J_det = simplify(det(J_s(1:3,1:3)));

Jd = expand(J_det)
det_theta3_0 = subs(Jd,'theta3',0) % determinant when theta3 0 is 0 i.e.,
    Jacobian loses rank and there is a singularity
```

We get the following result:

Jd = $-\cos(\theta_2)\sin(\theta_3)\,a_2^{\,2}\,a_3 - \sin(\theta_2)\,a_2\,a_3^{\,2}\cos(\theta_3)^2 - \cos(\theta_2)\sin(\theta_3)\,a_2\,a_3^{\,2}\cos(\theta_3) + 1.0\sin(\theta_2)\,a_2\,a_3^{\,2} - a_1\sin(\theta_3)\,a_2\,a_3$

det_theta3_0 = 0

This shows us that Jacobian loses it's rank when $\theta_3 = 0$ i.e., when determinant is 0.
To show that the determinant obtained is equal to the equation provided we check with the following MATLAB code:

```
det_theta3_0 = subs(Jd,'theta3',0) % determinant when theta3 0 is 0 i.e.,
    Jacobian loses rank and there is a singularity

given = a2*a3*sin(theta3)*(a2*cos(theta2)+a3*cos(theta2+theta3));

given_theta3_0 = subs(given, 'theta3', 0) % the given equation is also 0
    when theta3 0 is 0 i.e., both expressions are the same
```

We get to see that both expressions are the same and are 0 when $\theta_3 = 0$:

det_theta3_0 = 0

given_theta3_0 = 0

(b) The previous expression corresponds to three possible configurations. At each position, in which direction is the arm unable to move?

(i) Position A: the arm is fully stretched.

Solution: θ_2, θ_3, and θ_4 will be zero and the arm will not be able to move in any of the axes instantaneously.

(ii) Position B: the end-effector is right above the base, without the arm being stretched.

Solution: the arm will not be able to move in x and y axes instantaneously.

(iii) Position C: the end-effector is above the base and the arm is stretched.

Solution: the arm is not able to move along z-axis instantaneously.

2. Randomly spread 10 cubes on one side of your robot's workspace. Out of these 10 cubes, five should be a mix of red and yellow, and five a mix of blue and green cubes. Designate spots in your robot's workspace for a red bin and a yellow bin. Assume that there are no other items in the workspace. Make your robot system place all the red cubes in the red bin and the yellow cubes in the yellow bin. You'll be scored by the RA after they have randomly placed cubes in the workspace. You'll get 10 points for every correctly placed cube, -10 points for each cube placed in the wrong bin, and -5 points each time a cube falls enroute. You're also to submit a video of a successful execution and MATLAB code.

Solution:
In this task we will use our lab 7 code to get the image coordinates of objects i.e., their centroids from camera image and provide it to the code from task 8.5 which will give us the real world coordinates. We will then give these coordinates to code from lab 6 as input. Lab 7 code will also give us the color of each object and the number of objects of each color. This will help in placing the colored objects in their respective bins.

3. Again, randomly spread 10 cubes in the same color distribution - five from the set of red and yellow, and five from blue and green. But now it is possible that the cubes are stacked to any level. Designate spots in your robot's workspace for a red bin and a yellow bin. Assume that there are no other items in the workspace. Make your robot system place all the red cubes in the red bin and the yellow cubes in the yellow bin. You'll be scored by the RA after they have randomly placed cubes in the workspace. You'll get 6 points for every correctly placed cube, -6 points for each cube placed in the wrong bin, and -3 points each time a cube falls enroute. You're also to submit a video of a successful execution and MATLAB code.

Solution:
This task will be the same as previous one but we will also have stacked cubes and we can get that information from depth camera i.e., we will get the value of z coordinate which was fixed before as camera was at a specific height from the objects whereas now since the cubes are stacked, we will have a z value that is variable. The rest of the pick and place task will be executed in the same way as previous task.