Optimal Kinematic Design of Robots

Lab 1: Optimal placement and kinematic design of a SCARA robot: 4 hours

Objective:

The goal of this lab is to design a SCARA robot that must be able to perform **any cutting trajectories** in a prescribed rectangular area. The robot is constrained to operate in a workshop area of size 4m×4m. For cost reasons, the link lengths should not exceed 2m.

Work plan

- Given a SCARA robot with joint limits ±132° and ±141° for theta1 and theta2, respectively, and link lengths L1= L2=1 and given a set of **disc obstacles** around the robot, write a first Matlab function that determines the *optimal* **placement of the robot base**, such that the robot is able to follow any **cutting trajectories** in a prescribed rectangular area.
- If the prescribed rectangular area cannot be fully covered whatever the placement, which may happen in case of high obstruction, the Matlab function should give the **best placement** (i.e. corresponding to the greatest covered area of the prescribed rectangle).
- Write a second function that **optimizes the link lengths** together with the base placement for a full covering of the rectangular area when no feasible placement is found with L1=L2=1.
- Plots/animations should be provided for 3 different case studies (low, medium, high obstruction) to demonstrate that your codes work successfully.

Tips

- The prescribed rectangular area will be approximated by a series of equally distributed points;
- To verify that none of the two links intersect the disk obstacles, you will need to write a function that calculates the intersection between a line segment and a disk. This function is given next page;
- To calculate the IGM of the simplified SCARA robot, do not use the Matlab 'solve' function (it is too slow!). See IGM equations next page;
- The Matlab functions Fmincon (local search with constraints, reconstruction of the gradient) or simulannealbnd (global search with bounds, simulated anealing) can be used here but you are free to use any optimization means. You can also resort to a systematic search, which will give the 'global' optimal solution (in theory), at the cost of a high calculation time. If this strategy is feasible with 2 optimization variables, it is hardly conceivable with 4 variables.

Bonus question :

If, in addition, a "good" dexterity of the SCARA robot is requested in the prescribed rectangular area, how would you modify your algorithms?

Deliverables

- A pdf file with all necessary explanations (max 6 pages not incl. code);
- Your Matlab function. Provide sufficient comments in your Matlab code;
- Send a zipped file and name it as follows: NA1-NA2-2 where NA1 and NA2 are the first three letters of your last name;

- Due date: 30 October

Intersection test between a line segment and a disc:

Circle equation: $(x - x_c)^2 + (y - y_c)^2 = R^2$
Segment equations : $x = x_a(1 - t) + tx_b$; $y = y_a(1 - t) + ty_b$ and $0 \le t \le 1$

```
function cols = checkObstacles(XA, YA, XB, YB, XC,YC,R)

%check if one given segment intersect one given obstacle
% returns 1 if collides, 0 if not
% segment AB, disc obstacle : center C, radius R

cols = 0;
a = ((XB-XA)^2 + (YB-YA)^2);
b = (2*XA*(XB-XA) - 2*XC*(XB-XA) + 2*YA*(YB-YA) - 2*YC*(YB-
YA));
c = XA^2 + XC^2 - 2*XA*XC + YA^2 + YC^2 - 2*YA*YC - R^2;

delta = b^2 - 4*a*c;

if (delta>=0)
    t1 = (-b - sqrt(delta))/(2*a);
    t2 = (-b + sqrt(delta))/(2*a);

    if ((t1 >= 0 && t1 <= 1) || (t2 >= 0 && t2 <= 1))
        cols = 1;

    end
end

end
```

Inverse geometric model:

The DGM is given by: x=L1*cos(theta1)+L2*cos(theta1+theta2)
y= L1*sin(theta1)+L2*sin(theta1+theta2)

This is a system of "type 8" as studied by Paul in 1981. Its solution is recalled below:

upon squaring and adding the resulting two equations, one comes up with a linear equation in C2=cos(theta2) whose solution is: C2=cos(theta2)=$(x^2+y^2-L1^2-L2^2)/2L1L2$. The two solutions are given by theta2=ATAN2(±sqrt(1–$C2^2$), C2).
For **each** solution theta2, one then has to solve a system of two linear equations in sin(theta1) and cos(theta1), which yields **one** solution for theta1, namely: theta1=ATAN2(S1, C1) where S1=(B1y–B2x)/($B1^2+B2^2$), C1==(B1x+B2y)/($B1^2+B2^2$) and B1=(L1+L2C2), B2= L2S2, S2=sin(theta2).