- RAPPORT TD 7&8 -

- Optimization approaches -

Groupe 7 :

- MARÇAL Thomas
- KOSKAS Axel

---

5. Dimension synthesis of a 2R mechanism

1.

Direct kinematics of the 2R mechanism:

$$[x\,;y] = [l_1 * \cos(\theta_1) + l_2 * \cos(\theta_1+\theta_2)\ \ ;\ \ l_1 * \sin(\theta_1) + l_2 * \sin(\theta_1+\theta_2)]$$

Inverse kinematics of the 2R mechanism:

$$\theta_2 = \arccos\left(\frac{-{l_1}^2 - {l_2}^2 + (X^2+Y^2)}{2 * l_1 * l_2}\right)$$

$$\theta_1 = \arctan\left(\frac{Y}{X}\right) - \arctan\,(l_2 * \frac{\sin(\theta_2)}{l_1 + l_2 * \cos(\theta_2)})$$

2. The Jacobian Matrix for the 2R robot is :

$$Jacobian = \begin{pmatrix} -l2 * sin(\theta_1+\theta_2) - l1 * sin\,(\theta_1) & -l2 * sin\,(\theta_1+\theta_2) \\ l2 * cos(\theta_1+\theta_2) + l1 * cos\,(\theta_1) & l2 * cos\,(\theta_1+\theta_2) \end{pmatrix}$$

3. The Stiffness Matrix is :

$$K_\theta = \begin{pmatrix} \frac{3 * E * \mathrm{I}}{l_1^3} & 0 \\ 0 & \frac{3 * E * \mathrm{I}}{l_2^3} \end{pmatrix} with\ I = \frac{\pi\ *\ d^4}{64}\quad and\ \ E = 210\ GPa$$

4.

Decision variables:

Arm Length 1 : $L_1\ noted\ x_1$
Arm Length 2 : $L_2\ noted\ x_2$
Rayon : $R\ noted\ x_3$

Objective function:

$$Min(f(\theta_1,\theta_2)) = \rho_{steel} * \pi * R^2 * L_1 + \rho_{steel} * \pi * R^2 * L_2$$

Constraints:

In this problem, we have a total of 12 constraints, divided into 4 sets of 3 constraints for each vertex of the square that the robot arm must reach.

$$k = \sqrt{\frac{Max(Eigen(J^TJ))}{Min(Eigen(J^TJ))}} > 0.2$$

$$K_x^{-1} * f < 0.1 * 10^{-3}$$

$$x_1 > 0, \qquad x_2 > 0, \qquad x_3 > 0$$

So, in MATLAB, we have :

$$g_1 = 0{,}2 - \mathrm{k}$$

$$g_2 = [K_x^{-1} * f](1) - 0{,}1 * 10^{-3}$$

$$g_3 = [K_x^{-1} * f](2) - 0{,}1 * 10^{-3}$$

Results:

The MATLAB function returns the values :

$$f = [0{,}6734\,;2{,}0246\,;0.0919]$$

$$val = 559.1471$$

So :

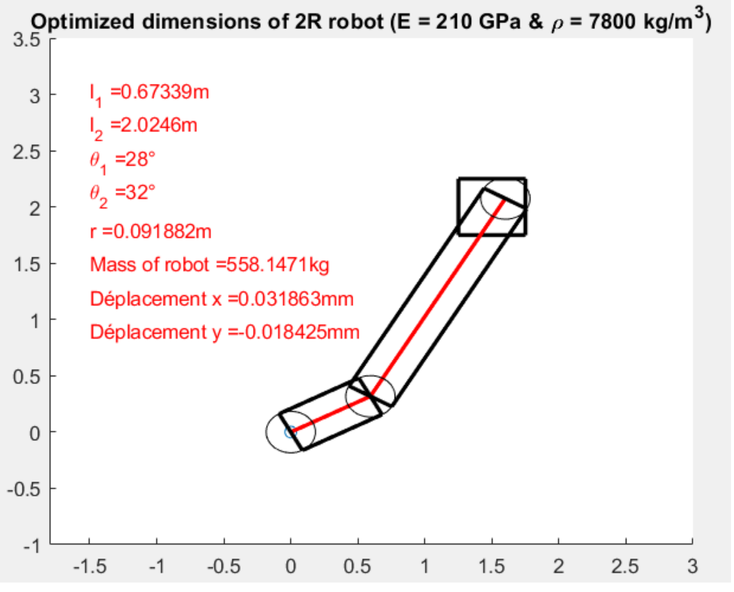
$$\boldsymbol{x_1 = 0.6734} \quad \& \quad \boldsymbol{x_2 = 2{,}0246} \quad \& \quad \boldsymbol{x_3 = 0.0919}$$

In conclusion :

$$L_1 = 0{,}6734\,m$$

$$L_2 = 2{,}0246\,m$$

$$R = 0{,}0919\,m$$



- RAPPORT TD 9&10 -

- Project Work -

We take as position for A and B : A=(1,9) & B=(9,1)

File 1: Implement a constraint function inside the 10x10 room having just one circle and one guess point between A & B.

Decision variables:

Position selon x : x_1

Position selon y : x_2

Fixed variables:

A & B

Objective:

$$\min(f(x)) = || x - A || + || x - B ||$$

Constraints:

$$\min(distance) > radian_{circle}$$

So, in MATLAB, we have :

$$g = radian_{circle} - \min(distance)$$

Results:

We obtain :

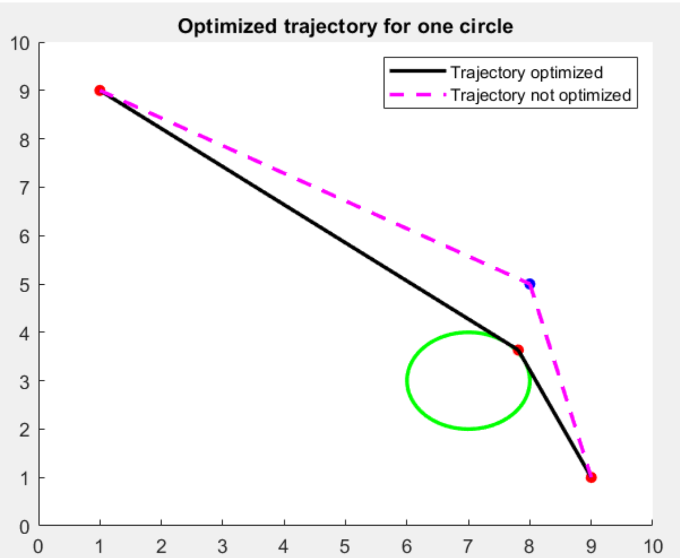
```
f =

    7.8110    3.6338


val =

   11.5607
```

So :



The optimization method consists in minimizing the distance between all the points, while avoiding crossing and intersecting the circles.

File 2: Now introduce 2 circles with 5 intermediate points between A & B and test the results.

Decision variables:

$$x = [x_1; x_2; x_3; x_4; x_5]$$

Fixed variables:

A & B

Objective:

$$\min(f(x)) = \sum_{k=1}^{4} || P_i - P_{i+1} || \qquad \text{with } P = [A ; x ; B]$$

Constraints:

$$\min(distance) > radian_{circle}$$

So, in MATLAB, we have :

$$g = radian_{circle} - \min(distance)$$

Results:

We obtain :

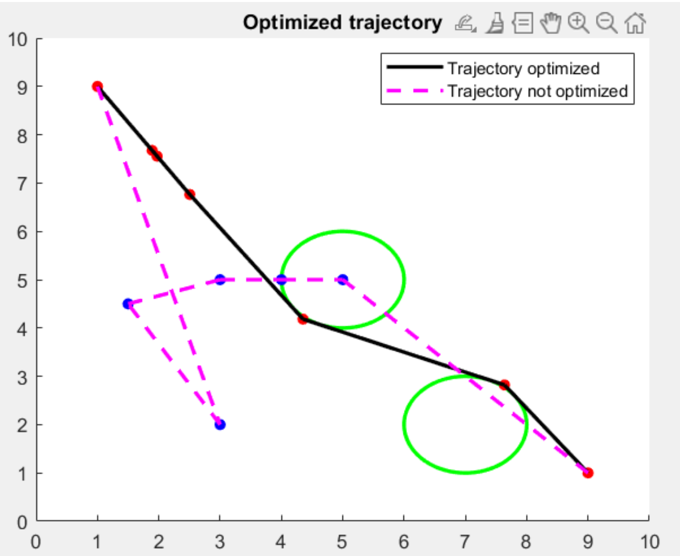
```
f =

    1.8915    7.6758
    1.9724    7.5557
    2.5057    6.7634
    4.3518    4.1837
    7.6381    2.8205


val =

   11.6998
```

So :



The optimization method consists in minimizing the distance between all the points, while avoiding crossing and intersecting the circles.

File 3: Introduce 4 circles with 7 intermediate points between A & B and test the results.

Decision variables:

$$x = [x_1; x_2; x_3; x_4; x_5]$$

Fixed variables:

A & B

Objective:

$$\min\big(f(x)\big) = \sum_{k=1}^{4} || P_i - P_{i+1} || \qquad \text{with } P = [A\ ;x\ ;B]$$

Constraints:

$$\min(distance) > radian_{circle}$$

So, in MATLAB, we have :

$$g = radian_{circle} - \min\,(distance)$$

Results:

We obtain :

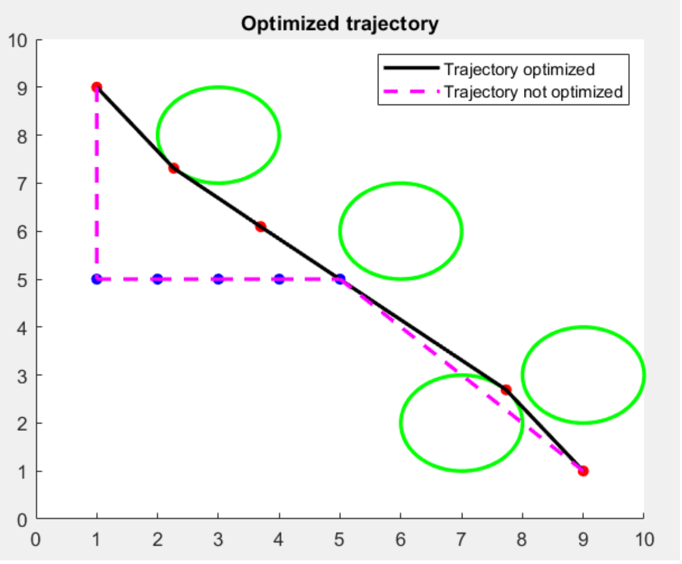
```
f =

    2.2663    7.3111
    3.6922    6.0964
    3.6923    6.0963
    3.6992    6.0905
    7.7318    2.6914


val =

   11.3813
```

So :



The optimization method consists in minimizing the distance between all the points, while avoiding crossing and intersecting the circles.

Final File :

The rectangular walls can now be introduced with two set of circles on the left side of the room and two others on the right side. Additional constraints must be defined to make the trajectory pass in the area between the walls.

Two rectangular walls, each of size 4x2 with their origins at (4, 0) and (4, 6) separates the test region into two rooms.

Decision variables:

$$x = [x_1; x_2; x_3; x_4; x_5; x_6; x_7]$$

Fixed variables:

A & B

Objective:

$$\min\big(f(x)\big) = \sum_{k=1}^{6} || P_i - P_{i+1} || \qquad \text{with } P = [A ; x ; B]$$

Constraints:

$$\min(distance) > radian_{circle}$$

So, in MATLAB, we have :

$$g = radian_{circle} - \min\,(distance)$$

Moreover, we have :

$$if\ (4 < x < 6\ and\ 4 < y < 6)\,\{\, constraint\ for\ the\ wall\}$$
$$else\ \{no\ constraint\}$$

Results:

We obtain :

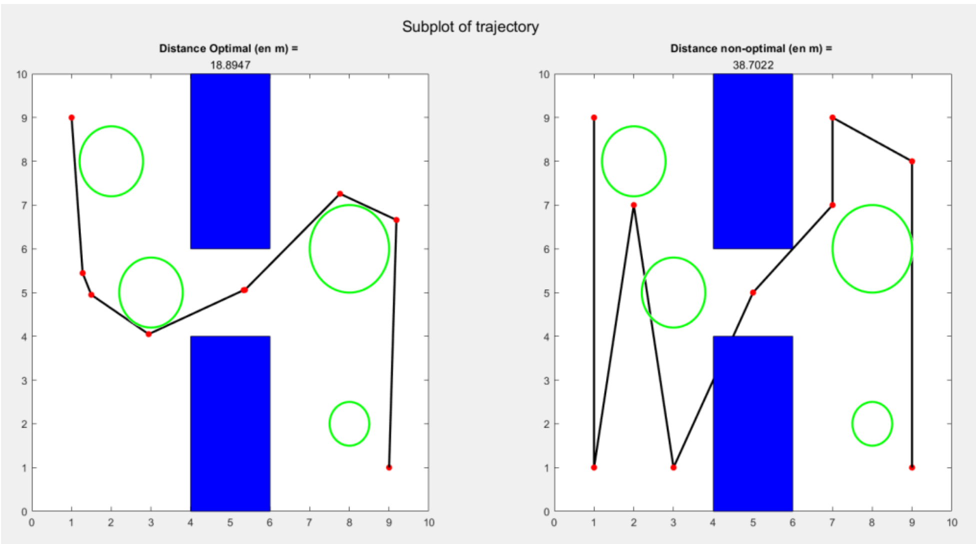
```
f =

    1.2786    5.4445
    1.4965    4.9494
    2.9414    4.0493
    5.3368    5.0554
    5.3664    5.0596
    7.7625    7.2576
    9.1850    6.6593


val =

   18.8947
```

So :



The optimization method consists in minimizing the distance between all the points, while avoiding crossing and intersecting the circles. Moreover, we must avoid touching the walls, so we must make sure that our trajectory passes through both walls.

The methodology of fmincom :

FMINCOM allows to find the minimum for a problem with several variables.

In this case, FMINCOM minimizes the distance between all points, which corresponds to the objective function in our problem.

Moreover, depending on whether we want to avoid walls and circles or not, we can put several constraints to avoid them.

The values LB and UB are used as bounds to create an interval in which our x-coordinates exist.