Optimization Techniques Intermediate Lab report

Suruthi Omprakash NATTAMAI	**M1-CORO-CSYS**	**Date: Friday 10 April,2020**
Nitha Elizabeth JOHN	**M1-CORO-EMARO**	
Raghuveer SIDDARABOINA	**M1-CORO-EMARO**	

Manufacturing Problem: The manufacturing time for the production of two parts were given. The maximum time allocated per week was also given. The objective of this optimization problem is to maximize the profit. The *number of part 1* and *number of part 2* are considered as the variables for this problem:

$$\text{i.e. } \mathbf{x} = (x_1, x_2).$$

The objective function is given by $f(\mathbf{x}) = f(x_1, x_2) = 50\, x_1 + 100\, x_2$

Two types of constants will be considered for every optimization problem. The equality constants given by $H(\mathbf{x}) = 0$ where $H = (h1 , h2, \ldots\ldots hp)$ and inequality constants given by $G(\mathbf{x}) \leq 0$ where $G= (g1,g2,\ldots.gn)$.

$g_1(x) = 10x_1 + 5x_2 - 2500$

$g_2(x) = 4x_1 + 10x_2 - 2000$

$g_3(x) = 1x_1 + 1.5x_2 - 450$

$g_4(x) = -x_1$

$g_5(x) = -x_2$

The aim is to find x^* that maximizes f(x). An excel sheet was created to find the optimized solution to the problem. The control point was found to be (187.5, 125).

	A	B	C	D	E	F
1		**Manuactuing Time**		**Max time per week**		
2	Type of machine	P1	P2			
3	M1	10	5	2500	0	-5
4	M2	4	10	2000	0	-2
5	M3	1	1.5	450	-75	-75.5
6	Profit	50.00 €	100.00 €		21,875.00 €	
7	x	187.5	125			
8		187	125			21850
9		188	125		21,900.00 €	
10						
11						

Project: Optimization of a trajectory AB inside a 10x10 room by avoiding collisions against obstacles using MATLAB.

For this project we consider a room of size 10x10 with obstacles in the form of circles and walls. The objective here is to optimize the trajectory between Point A in the left to Point B in the right.

The program contains three functions and a start script. The three functions are:

(1) Objective function: Here we minimize the sum of the norms of the individual segments between each chosen point.

(2) Constraint function: We set the constraints such that:
   - The distance from the centre of the circle to each point must be greater than the radius of the circle. We do this to ensure that the chosen point is always out of the circle.
   - The distance between two points should be greater than the radius of the circle so that the line segment does not intersect the circle.
   - We can also set a constraint for the distance between two points to be at least 1cm such that no two points lie at the same co-ordinates.
   -

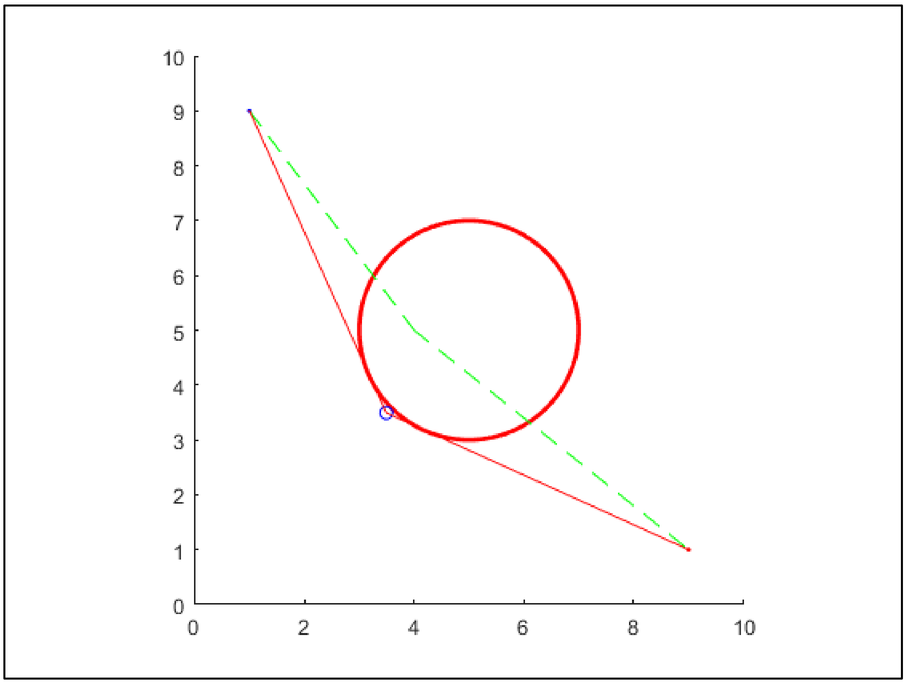
(3) Distance function: Here we discretize each individual segments of the trajectory with n number of points in order to calculate the minimum distance between the centre of circle to the guess point with respect to the initial points.

In the start function we set the initial coordinates of the points, set the upper bound, lower bound values and the centre of the circle and also define the radii of the circles. We optimise the objective function with the help of fmincon() function which calls the objective function and the constraint function.

The project is divided into the following five parts. The dashed line represents the non-optimal path and the red line represents the optimal trajectory which satisfies all the constraints.

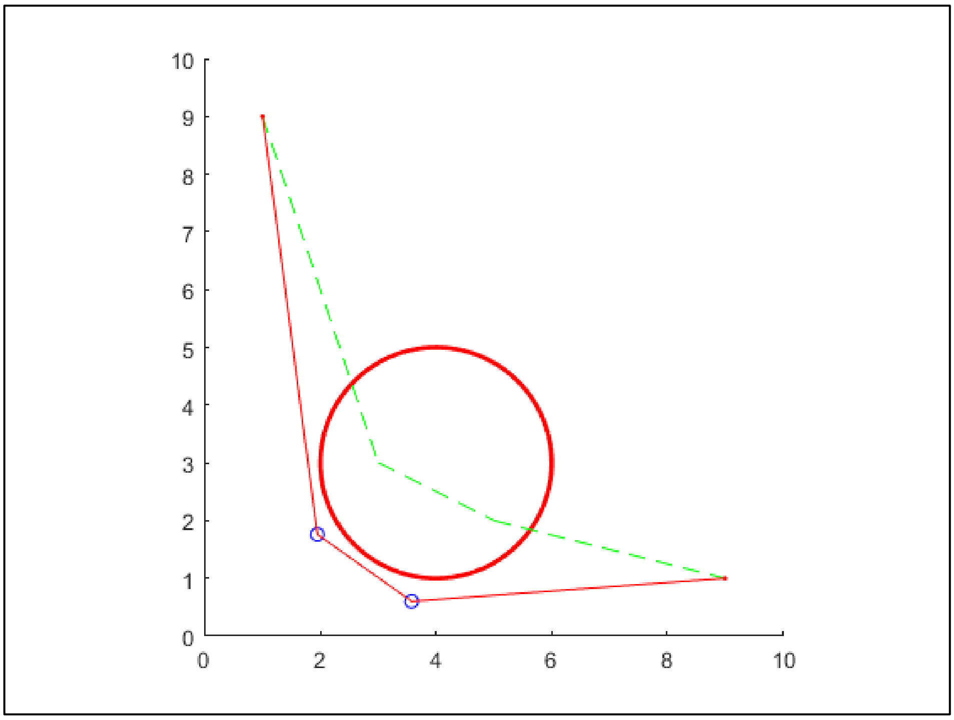
Step 1: 1 Circle and 1 guess point

Here we take the radius of the circle equal to 2 and the centre at [5,5]. We set the initial guess points such that the trajectory would intersect the circle and then optimize this initial point so that it chosen at a coordinate outside the radius of the circle. Hence, the trajectory to reach from A to B does not pass through the circle.



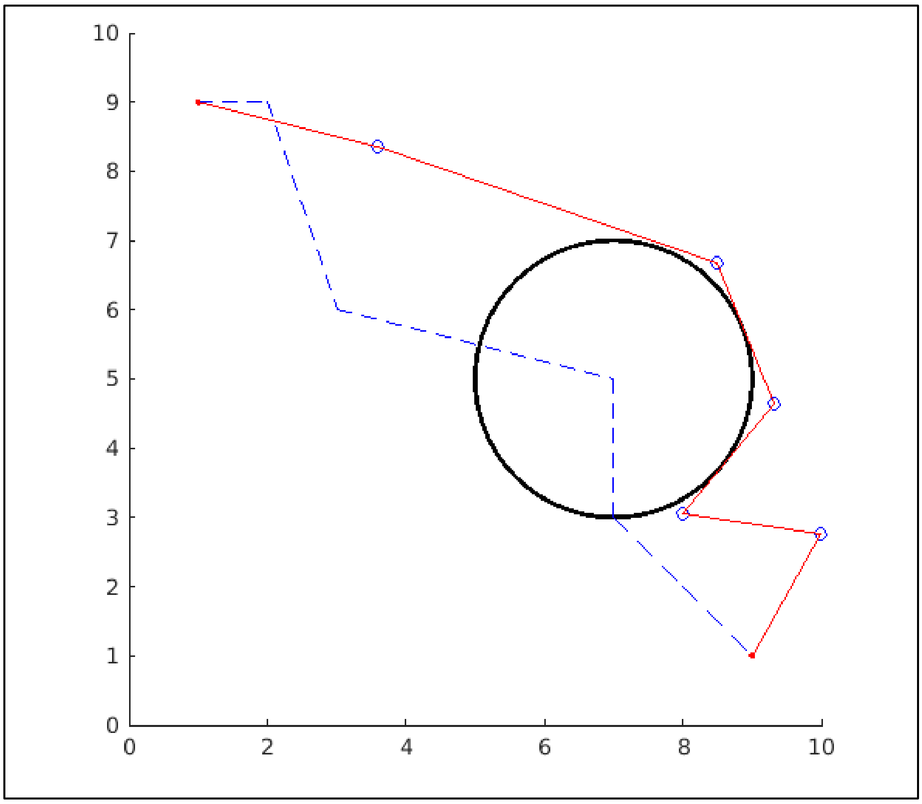
Step 2: 1 Circle and 2 guess points

Here we take a circle of radius equal to 2 and centre at [4,3]. We take two initial guess points which would intersect the circle to reach from A to B and then optimize it so that the trajectory does nor intersect the circle. Here we need to add a constraint to keep some minimum distance between these two points so that they do not coincide.



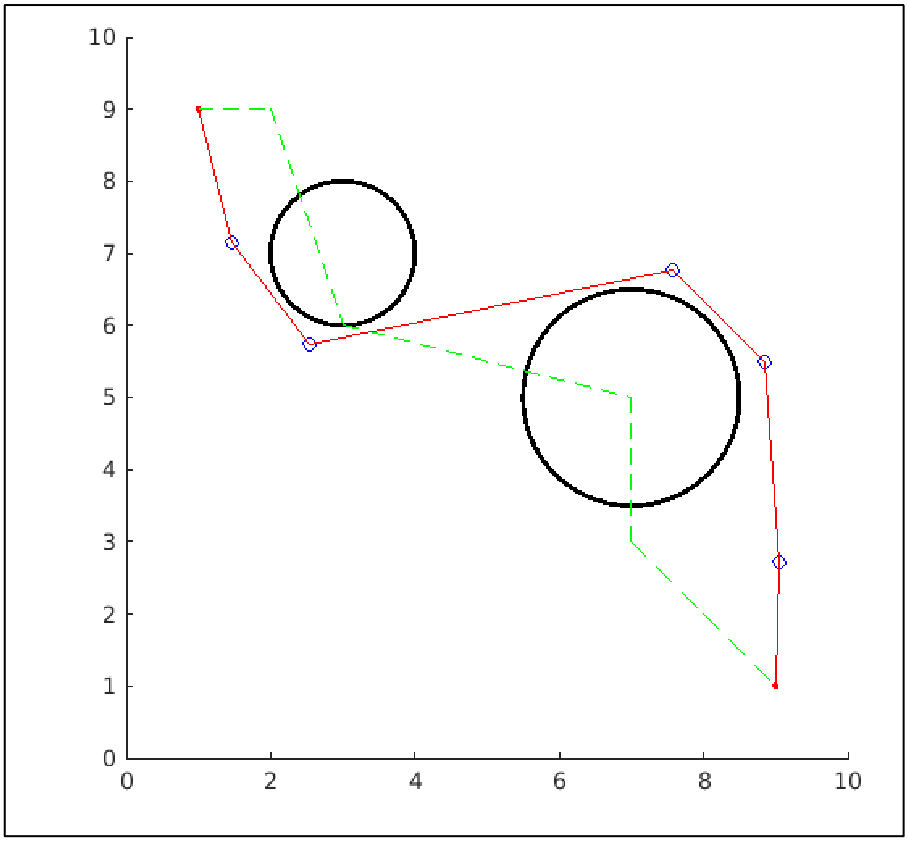
Step 3 : 1 Circle and 5 guess points

The room of the prescribed size was considered with a single obstacle in the form a single cirle.The circle considered here was of radius 3 centered at location (3,3).The discretization was done done between the first_point at location (1 , 9) and last_point (9,1) . Multiple intermediate guess points were chosen in beween the first and last point (in code given as *first_point* , *last_point*).The objective function was writen so as to minimise the sum of norms of the indivudual segments.The *fmincon()* was used to call the new altered objective function and constraint function. The optimal trajectory was found beween initial and final points without colliding with the obstacle. Multiple for loops were used for plotting the new optimised trajectory with 5 guess points.The radius of the circle was chosen to be 2.



Step 4 : 2 Circles and 5 guess points

The same room is considerd in this step also. Now the number of obstacles is increased.Two circles of different radius (*radius_1* = 1.5 , *radius_2* =1) centerd at position (7,5) and (3,7)is considered.The aim of this section is to optimise the trajecory in a such a way that it doesnot collide with any of the circles.Multiple guess points were also implemented.The plot of the optimised trajectory is given below.



10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10