OPTEC PROJECT

Optimal Path Planning of Robot

Prof. Fouad BENNIS

Presented By

Arijit Mallick

Sivanand Yadav Katamani

May 2016

Path Planning

Optimal Path of a robot for a given environment

A. Problem Statement

The goal of this project is to find the optimal path for a robot between the starting point and a given goal point in a 10X10 matrix map, consisting of circular and rectangular obstacles.

Moreover, certain preconditions are to be considered while designing the environment and the algorithm.

We will use random starting and goal positions in the configuration space provided.

Objective function minimises the distance between the starting and finishing point.

Constraint function prevents any intermediate point or line segment connecting those points intersect any obstacle.

B. Steps for designing the setup

We will divide our project into two parts: Initially, we will use a single circular obstacle and random starting and finishing points. Secondly, further extending the constraints and developing the algorithm, we will propose a solution to the given map with multiple obstacles.

C. Mathematical formulation

• The objective function:

The objective function can be defined as follows:

$$distance=\sum_{i=1}^{n}\sqrt{\left(x_{i+i}^{2}-x_{i}^{2}\right)}$$

where n is the number of intermediate points including the starting and finishing points. This variable can be set by the user themselves. Here x denotes the (x,y) co-ordinates of the intermediate points in sequence.

• The constraint function:

The constraint functions for a given circular obstacle can be given as:

$$r_j - dist < 0$$

simply putting, j denotes the obstacle number or radius of the j th circle; *dist* denotes the distance between the centre of the obstacle and the new intermediate point. This ensures that the line segment does not intersect the obstacle. This parameter is well defined inside the constraint function of the attached matlab files.

D. MATLAB Environment and fmincon

• fmincon

fmincon is a MATLAB function which finds the minimum of the constrained nonlinear multivariable function. It has many arguments, and we will use a selected of them. As provided in the MATLAB guide:

$$\min f(x) \text{ such that } \begin{cases} c(x) \leq 0 & [] \\ c_{eq}(x) = 0 & [] \\ A.x \leq b & [] \\ A_{eq}.x \leq b_{eq} & [] \\ lb \leq x \leq ub, & [] \end{cases}$$

b and beq are vectors, A and Aeq are matrices, c(x) and ceq(x) are functions that return vectors, and f(x) is a function that returns a scalar. f(x), c(x) and ceq(x) can be nonlinear functions.

```
options=optimset('Display','Iter','Tolx',1.00e-6,'TolFun',1.00e-8,'MaxIter',100,'MaxFunEvals',500);

optima=fmincon(@(x)objectivefun(x,start,finish),x0,[],[],[],[],start,finish,@(x)constraintfun(x,start,centre,finish,radius),options);
```

In our case, we have applied the *optimset* function in MATLAB in order to set parameters for the *fmincon* function as shown in the above figure. x0 used in the *fmincon*. x0 indicates initial starting point, which can be put as random (x,y)

coordinate in configuration space, which must be within upper bound and lower bound. (*ub, lb* parameters in *fmincon*).

• MATLAB environment

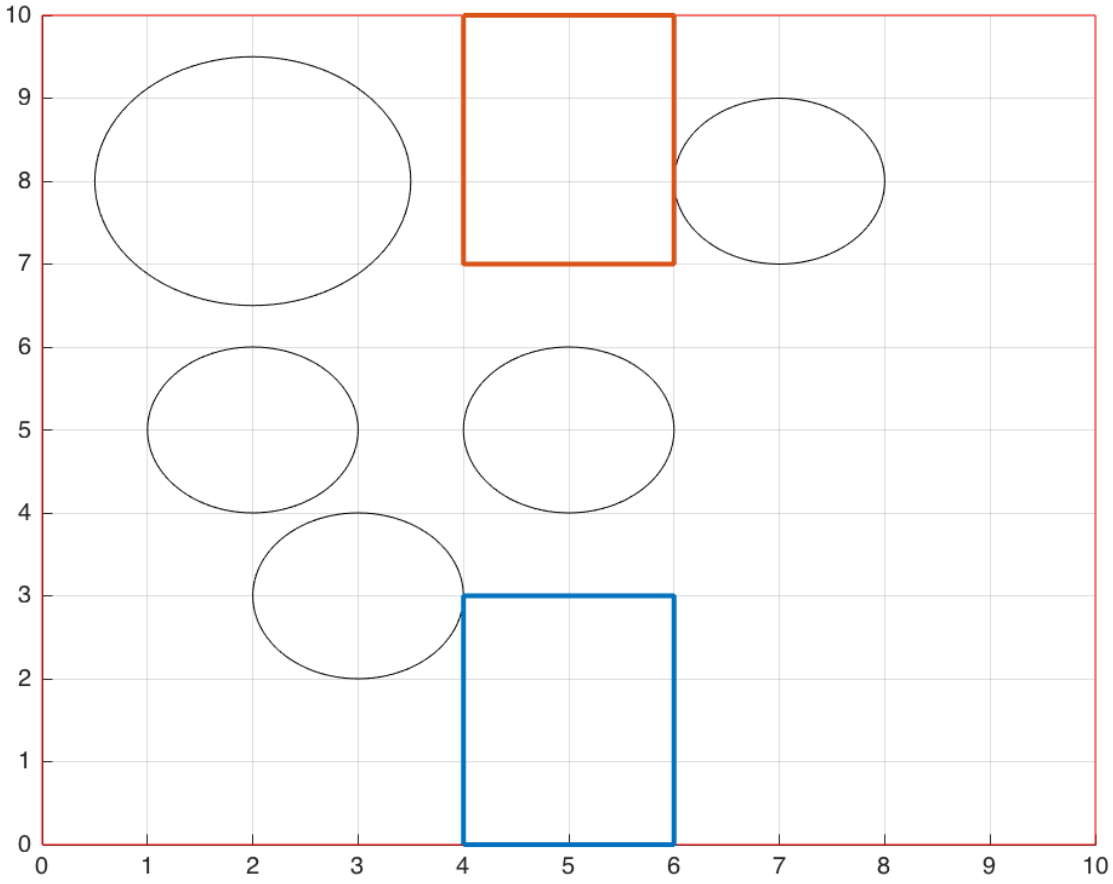


figure 1: Above map shows the presence of obstacles. At the final stage, we will find the optimal path between the initial and final point. Preferably the initial point is on left and final point is on right side of the map. The rectangular boundaries are also the part of the obstacle.

Next, we will take a sample map with single obstacle and single intermediate point. After that we will develop the path planning for more complex map like shown above.

E. RESULTS

• Single Obstacle Map

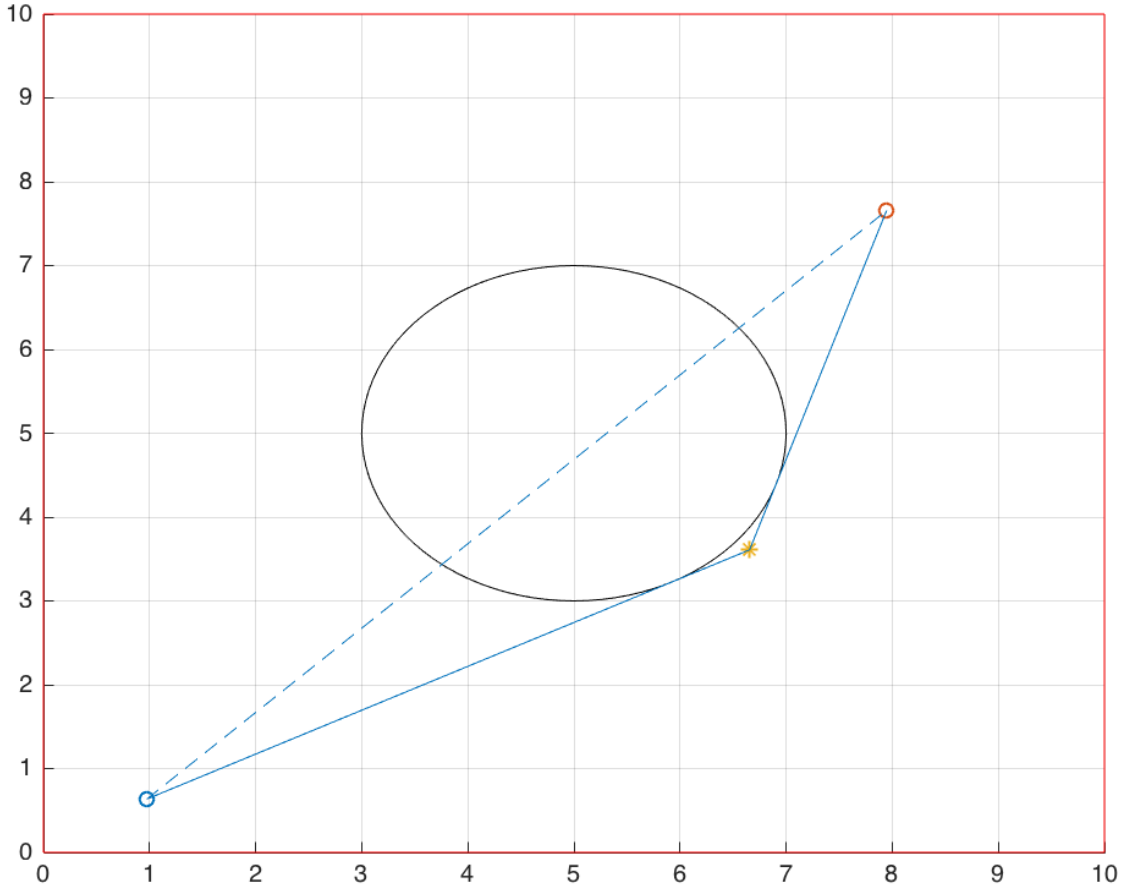
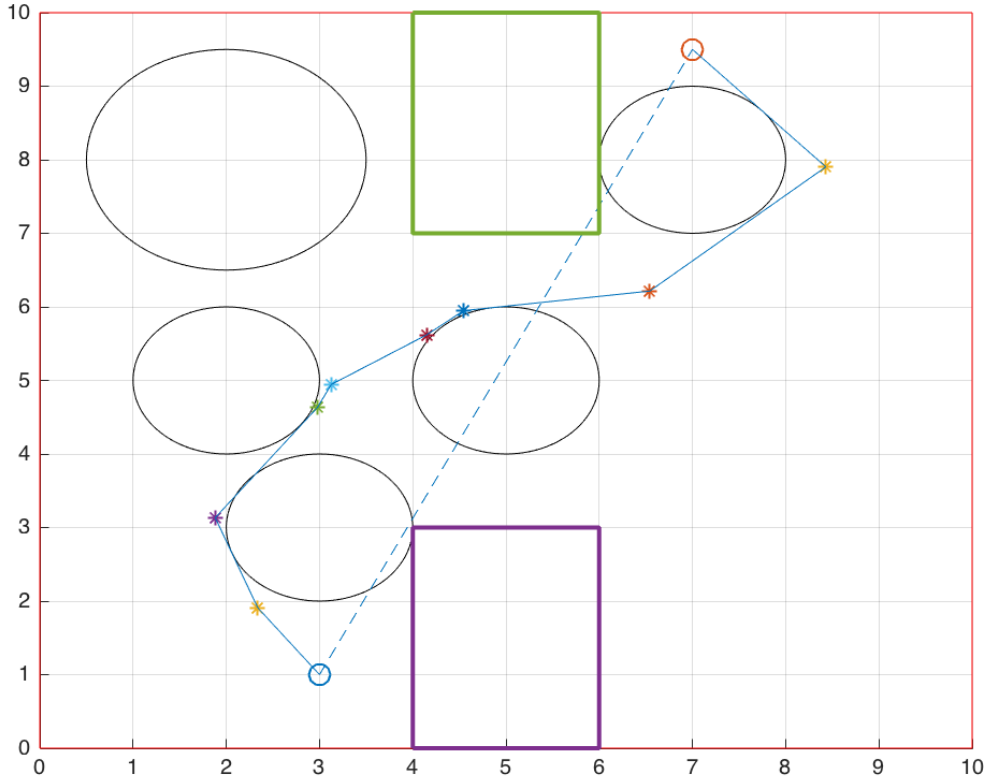
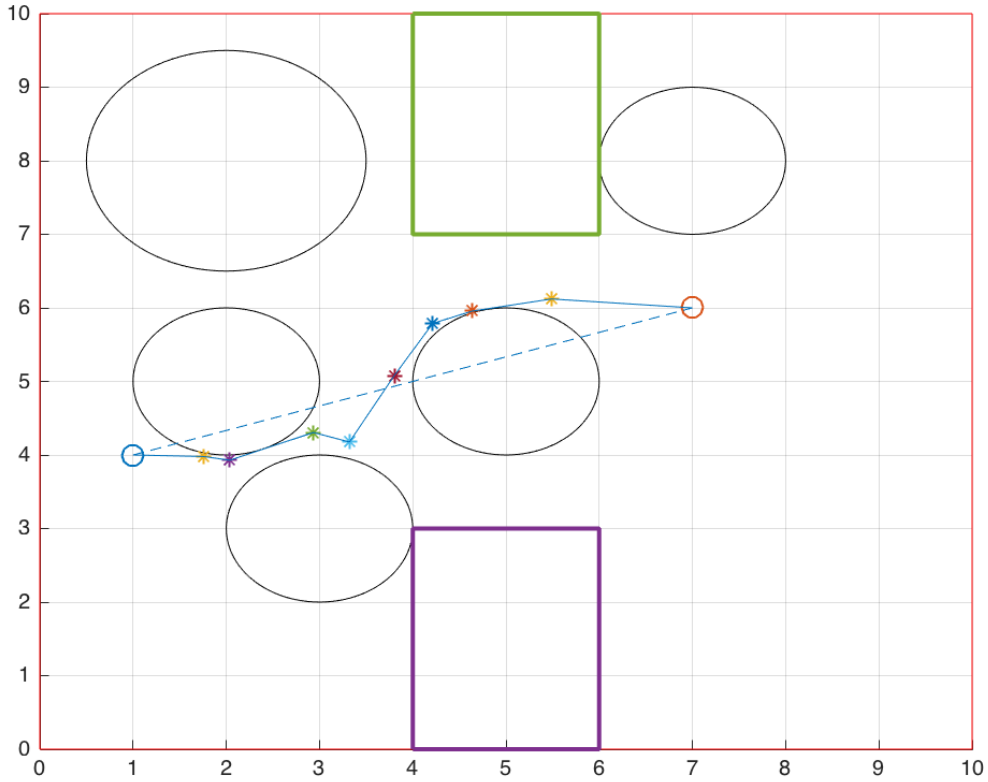


figure 2: In this figure, a preliminary algorithm testing has been done with a single obstacle and a single intermediate point, with (n=1). The blue dot represents starting point and the red dot shows the finishing point. These are generated by random points. The dotted line indicates the path without the obstacle.

In the attached file,
path_planning_1.m : Main file generating the map and fmincon function.
constraintfun.m : Contains the constraint function definition
objectivefun.m : Contains the objective functions as shown in prev. section

• Multiple Obstacle Map (Final Map)





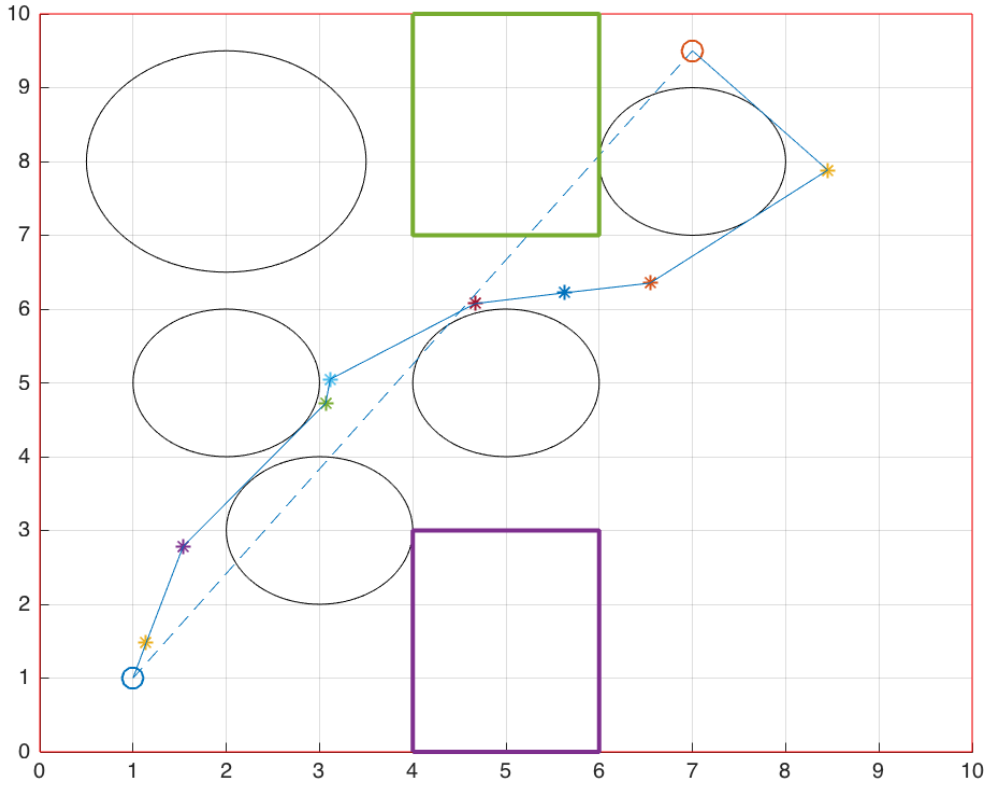


figure 3 (a),(b) and (c): figures show for 8 intermediate points, and at given starting and final points (chosen manually as well as randomly generated); Red dot shows the finishing point where the blue point shows the initial point. Rectangular obstacles are added which are avoided by correctly setting the bounds. figure (a) , figure (b) and figure (c) has start point: [3,1],[1,4] and [1,1] respectively; whereas finishing points are [7,9.5],[7,6] and [7,9.5] respectively. The parameter 'MaxFunEvals' is set at 500 for all cases.

The solution of the aforesaid path is given in the following files:
path_planning_3.m : Main file generating the map and fmincon function.
constraintfun3.m : Contains the constraint function definition
objectivefun3.m : Contains the objective functions as shown
arrayline.m : Works like linspace, divides a line into points, and later used in order to check whether those points lie inside the circle or not.

The optimset parameters are set as follows:
MaxFunEvals: Maximum function evaluation allowed, here 500.
MaxIter: Maximum allowed iterations, here 200.
TolFun: Maximum allowed tolerance function value, here 1e-08.

Tolx: Termination allowed tolerance on x, here set as 1e-06.

Hence, for the given cases, the path planning algorithm successfully approximates a optimal path to the goal point.

F. REFERENCE

1. Matlab function guide: fmincon (Mathworks).
2. Matlab function guide and documentation on plot and polygon functions. (Mathwork).