Scalar fields and their gradients, which are vector fields, can be used in robotics for motion planning. Consider a robot which needs to move in a room to a desired point avoiding some obstacles. The so-called navigation function is constructed for this purpose which is a continuously differentiable scalar field defined on the obstacle-free inside of the room, has a unique minimum at the goal point and attain its maximum value at the boundary of the room and the obstacles. A robot moving in the reverse direction to that of the gradient of the navigation function can avoid the obstacles and reach to the goal without hitting the walls of the room.

Let us consider a spherical room centered at the target point with radius , four spherical obstacles with radii and centers . Let denote the Euclidean distance between and , namely, . The navigation function can then be constructed as

where is a large enough positive number and

for . The number of ’s should be the same as the number of obstacles.

Consider a motion planning problem with two obstacles with and the parameters and given depending on your group number as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Group Number |  | and |  |  |  |  |
| 1 | 6 | 1 | -2 | 2 | 2 | 2 |
| 2 | 6 | 1 | -2 | -2 | 2 | -2 |
| 3 | 6 | 1 | -2 | 2 | 2 | 2 |
| 4 | 6 | 1 | -2 | -2 | 2 | -2 |
| 5 | 6 | 1.5 | -2 | 2 | 2 | 2 |
| 6 | 6 | 1.5 | -2 | -2 | 2 | -2 |
| 7 | 6 | 1.5 | -2 | 2 | 2 | 2 |
| 8 | 6 | 1.5 | -2 | -2 | 2 | -2 |
| 9 | 7 | 1 | -2 | 2 | 2 | 2 |
| 10 | 7 | 1 | -2 | -2 | 2 | -2 |
| 11 | 7 | 1 | -2 | 2 | 2 | 2 |
| 12 | 7 | 1 | -2 | -2 | 2 | -2 |
| 13 | 7 | 1.5 | -2 | 2 | 2 | 2 |
| 14 | 7 | 1.5 | -2 | -2 | 2 | -2 |
| 15 | 7 | 1.5 | -2 | 2 | 2 | 2 |
| 16 | 7 | 1.5 | -2 | -2 | 2 | -2 |
| 17 | 6 | 1 | -2 | 2 | 3 | 2 |
| 18 | 6 | 1 | -2 | -2 | 3 | -2 |
| 19 | 6 | 1 | -2 | 2 | 3 | 2 |
| 20 | 6 | 1 | -2 | -2 | 3 | -2 |
| 21 | 6 | 1.5 | -2 | 2 | 3 | 2 |
| 22 | 6 | 1.5 | -2 | -2 | 3 | -2 |
| 23 | 6 | 1.5 | -2 | 2 | 3 | 2 |
| 24 | 6 | 1.5 | -2 | -2 | 3 | -2 |
| 25 | 7 | 1 | -2 | 2 | 3 | 2 |
| 26 | 7 | 1 | -2 | -2 | 3 | -2 |
| 27 | 7 | 1 | -2 | 2 | 3 | 2 |
| 28 | 7 | 1 | -2 | -2 | 3 | -2 |
| 29 | 7 | 1.5 | -2 | 2 | 3 | 2 |
| 30 | 7 | 1.5 | -2 | -2 | 3 | -2 |
| 31 | 7 | 1.5 | -2 | 2 | 3 | 2 |
| 32 | 7 | 1.5 | -2 | -2 | 3 | -2 |
| 33 | 6 | 1 | -3 | 2 | 2 | 2 |
| 34 | 6 | 1 | -3 | -2 | 2 | -2 |
| 35 | 6 | 1 | -3 | 2 | 2 | 2 |
| 36 | 6 | 1 | -3 | -2 | 2 | -2 |
| 37 | 6 | 1.5 | -3 | 2 | 2 | 2 |
| 38 | 6 | 1.5 | -3 | -2 | 2 | -2 |
| 39 | 6 | 1.5 | -3 | 2 | 2 | 2 |
| 40 | 6 | 1.5 | -3 | -2 | 2 | -2 |
| 41 | 7 | 1 | -3 | 2 | 2 | 2 |
| 42 | 7 | 1 | -3 | -2 | 2 | -2 |
| 43 | 7 | 1 | -3 | 2 | 2 | 2 |
| 44 | 7 | 1 | -3 | -2 | 2 | -2 |
| 45 | 7 | 1.5 | -3 | 2 | 2 | 2 |
| 46 | 7 | 1.5 | -3 | -2 | 2 | -2 |
| 47 | 7 | 1.5 | -3 | 2 | 2 | 2 |
| 48 | 7 | 1.5 | -3 | -2 | 2 | -2 |
| 49 | 7 | 1.5 | -2.5 | 2 | 2 | 2 |
| 50 | 7 | 1.5 | -2.5 | -2 | 2 | -2 |

Use the following MATLAB code

%2024 Fens202 Robot Motion Planning Project

clear all

kappa=2;

StepCurves=.05;%The precision used to plot the level curves

StepVectors=.5;%The precision used to plot the gradient vectors

%The target point

q0=[0;0];

%Centers of the obstacles

q1=[FILL;FILL];

%The followings can be added if there are more obstacles

q2= FILL;

%Radius of the room with center at the target

r0= FILL;

%Radii of the obstacles

r1= FILL;

r2= FILL;

%Add more if there are more obstacles

syms x y % These are variables for symbolic math calculations, you will need Symbolic Math Toolbox of MATLAB. If you use MATLAB ONLINE on your internet browser, the toolbox should be automatically available.

D0=(x-q0(1))^2+(y-q0(2))^2;

D1=(x-q1(1))^2+(y-q1(2))^2;

%The followings can be added if there are more constraints

D2= FILL;

%D3=(x-q3(1))^2+(y-q3(2))^2;

%D4=(x-q4(1))^2+(y-q4(2))^2;

Beta0=r0^2-D0;

Beta1=-r1^2+D1;

%The followings can be added if there are more constraints

Beta2= FILL;

%Beta3=-r3^2+D3;

%Beta4=-r4^2+D4;

Phi= FILL /( FILL 1/kappa);%New betas should be added if there are more constraints

g=gradient(FILL,[x,y]);

SolutionsForMinima=solve(g,[x,y]); % This calculates WHAT?

SolutionsForMinima.x % Try to use this and the following to find out the minima

SolutionsForMinima.y % You may need to calculate root of a polynomial using another MATLAB code or plotting the figure via ezplot(polynomial)

% Use the following codes to plot level curves and normalized gradient

% vector fields

[X,Y] = meshgrid(q0(1)-r0:StepCurves:q0(1)+r0,q0(2)-r0:StepCurves:q0(2)+r0); %This is generating a grid for numeric calculations

F = real(subs(Phi,[x,y],{X,Y})); %Calculating the navigation function on each point of the grid

%For plotting the level curves

%contour(X,Y,F,50) %try this too and explain what is happening on the following

%line

contour(X,Y,F,F(ceil(length(F)/2),1:5:length(F)))

hold on % Type “help hold” on MATLAB command window

viscircles(q0',r0,'Color','black') % Image processing toolbox required for this. You can use MATLAB online instead

viscircles(FILL,r1,'Color','red')

viscircles(q2',FILL,'Color','red')

%add more circles if needed

%calculating the gradient

[XX,YY] = meshgrid(q0(1)-r0:StepVectors:q0(1)+r0,FILL);

G1 = subs(g(1),[x,y],{XX,YY});

G2 = subs(FILL,[x,y],{XX,YY});

%normalizing gradient vectors so that we can see very small vectors as well

G=sqrt(G1.^2+G2.^2);

GG1=-G1./G;

GG2=FILL;

quiver(XX,YY,FILL,FILL,0.5)%plots the normalized gradient vector field

Task 1: For different values of 2,4,6 and 8, plot the level sets of the navigation function (scalar field) as well as the gradient vectors (vector field) and discuss when the navigation function has a unique minimum. Why is it important that the scalar field has a unique minimum?

Task 2: Discuss how the number of minima of the scalar field can be calculated. Use the solve command and ezplot to plot a polynomial.

Task 3: Choose a point in between the wall and obstacles. Discuss how you can use the gradient vector field to reach the target point avoding obstacles. Choose a small time step and write a code for the evaluation of the consecutive velocity vectors for each consecutive time instants.