Dated: 08/15/2016

**Homework 5**

**EE 5322**

**Intelligent Control Systems**

**By**

**Diptiman Hazra**

**(1001155533)**

**PROBLEM STATEMENT**

**Mobile Robot Control & Potential Fields**

1. **Potential Field.** Use MATLAB to make a 3-D plot of the potential fields described below.

You will need to use plot commands and maybe the mesh function. The work area is a

Square from (0,0) to (12,12) in the (x,y) plane. The goal is at (10, 10). There are 5 random obstacles being generated. Use a repulsive potential of Ki /(2+ ri ) for each obstacle, with ri the vector to the i-th obstacle. For the target use an attractive potential of Kt rt, with rt the vector to the target. Adjust the gains to get a decent plot. Plot the sum of the potential fields in 3-D

On the x,y-plane square from (0,0) to (12,12).

2. **Potential Field Navigation.** For the same scenario as in Problem 1, a mobile robot starts at

(0,0). The front wheel steered mobile robot has dynamics



with (x,y) the position, Ɵ the heading angle, vt the wheel speed, L the wheel base, and ϕ th steering angle. Set L= 2.

a. Compute forces due to each obstacle and goal as found in previous question. Compute total force on the vehicle at point (x,y).

b. Design a feedback control system for force-field control. Draw your control system.

c. Use MATLAB to simulate the nonlinear dynamics assuming a constant velocity vt and a

steerable front wheel. The wheel should be steered so that the vehicle always goes

downhill in the force field plot. Plot the resulting trajectory in the (x,y) plane. Use a

square from (0,0) to (12,12).

**3. Platoon of Mobile Robots.**

There are 3 robots in a platoon. Robot 1 is the leader. For each robot i take the simplified

Newton’s law dynamics (with mass=1)



with (x,y) the position of the vehicle and ,Fx ,Fy the forces in the x and y direction respectively.

The obstacles are the same as in problems 1 and 2 (ie randomly generated). This time define a circular work area with radius of circle being 20. The goal is to get to the centre of the circle. Make sure the robots are pushed away from the circumference of the circle. The starting points of the robot is random within the circle.

a. Program the forces for the leader node 1 to avoid the obstacles and go to the target. Same

scenario as above (but with these simplified dynamics).

b. Program the force on each follower to stay 2 units from the leader and not to run into each

other. Use repulsive POTENTIAL between followers as Ki /rij2 with rij= distance

between followers i and j. For the potential to the leader, use something like



with riL the distance from follower i to the leader, and rD the desired separation. Play

with the potentials to make it work properly. Compute forces properly using calculus.

c. Simulate. Plots trajectories in (x,y) plane. Start all robots at (0,0).

**DESCRIPTION**

In real time, the approach toward the avoidance of obstacle for a mobile robot has been developed using potential field method. Here everything is proposed based on elementary response and the modelling of the robot environment is mainly motivated first to obtain a map of the free space. In project we have a plane/area and obstacle at certain positions. They give either an attractive force or a repulsive force based on what it is i.e. whether the it’s a goal (attractive force) or an obstacle (repulsive force).

The goal has to have the highest force (attractive force in the entire plane).

**PROCEDURE**

* We define our work area which is a square from (0,0) to (12,12).
* Keep the goal area within the square i.e (10,10) using the distance formula.
* We create 5 random obstacles using rand function.
* Used the repulsive force given by Ki/(2+ri).
* Plotted the graph of the sum of the potential field calculated over a meshgrid.

**MATLAB CODE**

**PROBLEM 1**

clc;

clear all;

close all;

F=0;

m=0;

n=0;

pos = floor(1+(11.\*rand(5,2)));

K = [45,56,67,23,32,-150];

for x = 0:0.02:12

m=m+1;

for y =0:0.02:12

n=n+1;

r1 = abs(sqrt(power((x-pos(1,1)),2)+power((y-pos(1,2)),2)));

r2 = abs(sqrt(power((x-pos(2,1)),2)+power((y-pos(2,2)),2)));

r3 = abs(sqrt(power((x-pos(3,1)),2)+power((y-pos(3,2)),2)));

r4 = abs(sqrt(power((x-pos(4,1)),2)+power((y-pos(4,2)),2)));

r5 = abs(sqrt(power((x-pos(5,1)),2)+power((y-pos(5,2)),2)));

r6 = abs(sqrt(power((x-10),2)+power((y-10),2)));

f1(m,n) = K(1)./(2+r1);

f2(m,n) = K(2)./(2+r2);

f3(m,n) = K(3)./(2+r3);

f4(m,n) = K(4)./(2+r4);

f5(m,n) = K(5)./(2+r5);

f6(m,n) = K(6)\*r6;

PF(m,n) = f1(m,n)+f2(m,n)+f3(m,n)+f4(m,n)+f5(m,n)+f6(m,n);

end

n=0;

end

figure;

[x,y] = meshgrid(0:0.02:12);

mesh(x,y,PF);

**PROBLEM 2**

clc;

clear all ;

close all;

% global pos

input = [0;0;0];

[t,z] = ode23(@myrobot,[0 12],input);

[x,y] = meshgrid(0:0.02:12);

figure ;

plot (z(:,1),z(:,2));

hold on;

pos = [2,7;8,3;6,9;2,3;4,11];

plot(pos(1,1),pos(1,2),'\*');

hold on;

plot(pos(2,1),pos(2,2),'\*');

hold on;

plot(pos(3,1),pos(3,2),'\*');

hold on;

plot(pos(4,1),pos(4,2),'\*');

hold on;

plot(pos(5,1),pos(5,2),'\*');

hold on;

plot(10,10,'\*');

K = [45,56,67,23,32,-150];

m=0;

n=0;

[x,y] = meshgrid(0:0.02:12,0:0.02:12);

r1 = abs(sqrt(power((x-pos(1,1)),2)+power((y-pos(1,2)),2))) + 0.01;

r2 = abs(sqrt(power((x-pos(2,1)),2)+power((y-pos(2,2)),2)))+ 0.01;

r3 = abs(sqrt(power((x-pos(3,1)),2)+power((y-pos(3,2)),2)))+ 0.01;

r4 = abs(sqrt(power((x-pos(4,1)),2)+power((y-pos(4,2)),2)))+ 0.01;

r5 = abs(sqrt(power((x-pos(5,1)),2)+power((y-pos(5,2)),2)))+ 0.01;

r6 = abs(sqrt(power((x-10),2)+power((y-10),2)));

f1(m,n) = K(1)./(r1);

f2(m,n) = K(2)./(r2);

f3(m,n) = K(3)./(r3);

f4(m,n) = K(4)./(r4);

f5(m,n) = K(5)./(r5);

f6(m,n) = K(6)\*r6;

PF(m,n) = f1(m,n)+f2(m,n)+f3(m,n)+f4(m,n)+f5(m,n)+f6(m,n);

figure;

surf(x,y,PF);

hold on;

plot(z(:,1),z(:,2),'r');

**function out = myrobot(t,input)**

vT = 4;

L=2;

pos = [2,7;8,3;6,9;2,3;4,11];

k = 0.9;

K = [45,56,67,23,32,-150];

x = input(1);

y = input(2);

theta = input(3);

xG = 10; yG = 10;

XO\_1 = pos(1,1);YO\_1 = pos(1,2);

XO\_2 = pos(2,1);YO\_2 = pos(2,2);

XO\_3 = pos(3,1);YO\_3 = pos(3,2);

XO\_4 = pos(4,1);YO\_4 = pos(4,2);

XO\_5 = pos(5,1);YO\_5 = pos(5,2);

RG = sqrt(((xG-x)^2)+((yG-y)^2));

FG\_x = K(6)\*(xG-x)/RG;

FG\_y = K(6)\*(yG-y)/RG;

RI\_1 = ((XO\_1-x)^2)+((YO\_1-y)^2);

RI\_2 = ((XO\_2-x)^2)+((YO\_2-y)^2);

RI\_3 = ((XO\_3-x)^2)+((YO\_3-y)^2);

RI\_4 = ((XO\_4-x)^2)+((YO\_4-y)^2);

RI\_5 = ((XO\_5-x)^2)+((YO\_5-y)^2);

FI\_x1 = K(1)\*(XO\_1-x)/RI\_1;

FI\_x2 = K(2)\*(XO\_2-x)/RI\_2;

FI\_x3 = K(3)\*(XO\_3-x)/RI\_3;

FI\_x4 = K(4)\*(XO\_4-x)/RI\_4;

FI\_x5 = K(5)\*(XO\_5-x)/RI\_5;

FI\_y1 = K(1)\*(YO\_1-x)/RI\_1;

FI\_y2 = K(2)\*(YO\_2-x)/RI\_2;

FI\_y3 = K(3)\*(YO\_3-x)/RI\_3;

FI\_y4 = K(4)\*(YO\_4-x)/RI\_4;

FI\_y5 = K(5)\*(YO\_5-x)/RI\_5;

Fx = FG\_x + FI\_x1 + FI\_x2 + FI\_x3 + FI\_x4 + FI\_x5 ;

Fy = FG\_y + FI\_y1 + FI\_y2 + FI\_y3 + FI\_y4 + FI\_y5 ;

theta\_d = atan2(Fx, Fy);

phi = k\*(theta\_d - theta);

out(1) = vT\*cos(phi)\*cos(theta);

out(2) = vT\*cos(phi)\*sin(theta);

out(3) = vT\*sin(phi)/L;

out = out';

**PROBLEM 3**

clc;

clear all ;

close all;

% global pos

input = [0;0;0;0];

[t,z] = ode23(@con\_myrobot,[0 50],input);

[x,y] = meshgrid(0:0.02:12);

figure ;

plot (z(:,2),z(:,4));

hold on;

pos = [2,7;8,3;6,9;2,3;4,11];

plot(pos(1,1),pos(1,2),'\*');

hold on;

plot(pos(2,1),pos(2,2),'\*');

hold on;

plot(pos(3,1),pos(3,2),'\*');

hold on;

plot(pos(4,1),pos(4,2),'\*');

hold on;

plot(pos(5,1),pos(5,2),'\*');

hold on;

plot(10,10,'\*');

K = [45,56,67,23,32,-150];

m=0;

n=0;

[x,y] = meshgrid(0:0.02:12,0:0.02:12);

r1 = abs(sqrt(power((x-pos(1,1)),2)+power((y-pos(1,2)),2))) + 0.01;

r2 = abs(sqrt(power((x-pos(2,1)),2)+power((y-pos(2,2)),2)))+ 0.01;

r3 = abs(sqrt(power((x-pos(3,1)),2)+power((y-pos(3,2)),2)))+ 0.01;

r4 = abs(sqrt(power((x-pos(4,1)),2)+power((y-pos(4,2)),2)))+ 0.01;

r5 = abs(sqrt(power((x-pos(5,1)),2)+power((y-pos(5,2)),2)))+ 0.01;

r6 = abs(sqrt(power((x-10),2)+power((y-10),2)));

f1(m,n) = K(1)./(r1);

f2(m,n) = K(2)./(r2);

f3(m,n) = K(3)./(r3);

f4(m,n) = K(4)./(r4);

f5(m,n) = K(5)./(r5);

f6(m,n) = K(6)\*r6;

PF(m,n) = f1(m,n)+f2(m,n)+f3(m,n)+f4(m,n)+f5(m,n)+f6(m,n);

figure;

surf(x,y,PF);

hold on;

plot(z(:,1),z(:,2),'r');

**function out = con\_myrobot(t,input)**

pos = [2,7;8,3;6,9;2,3;4,11];

k = 0.9;

K = [45,56,67,23,32,-150];

x = input(1);

dxdt = input(2);

y = input(3);

dydt = input(4);

xG = 10; yG = 10;

XO\_1 = pos(1,1);YO\_1 = pos(1,2);

XO\_2 = pos(2,1);YO\_2 = pos(2,2);

XO\_3 = pos(3,1);YO\_3 = pos(3,2);

XO\_4 = pos(4,1);YO\_4 = pos(4,2);

XO\_5 = pos(5,1);YO\_5 = pos(5,2);

RG = sqrt(((xG-x)^2)+((yG-y)^2));

FG\_x = K(6)\*(xG-x)/RG;

FG\_y = K(6)\*(yG-y)/RG;

RI\_1 = ((XO\_1-x)^2)+((YO\_1-y)^2);

RI\_2 = ((XO\_2-x)^2)+((YO\_2-y)^2);

RI\_3 = ((XO\_3-x)^2)+((YO\_3-y)^2);

RI\_4 = ((XO\_4-x)^2)+((YO\_4-y)^2);

RI\_5 = ((XO\_5-x)^2)+((YO\_5-y)^2);

FI\_x1 = K(1)\*(XO\_1-x)/RI\_1;

FI\_x2 = K(2)\*(XO\_2-x)/RI\_2;

FI\_x3 = K(3)\*(XO\_3-x)/RI\_3;

FI\_x4 = K(4)\*(XO\_4-x)/RI\_4;

FI\_x5 = K(5)\*(XO\_5-x)/RI\_5;

FI\_y1 = K(1)\*(YO\_1-x)/RI\_1;

FI\_y2 = K(2)\*(YO\_2-x)/RI\_2;

FI\_y3 = K(3)\*(YO\_3-x)/RI\_3;

FI\_y4 = K(4)\*(YO\_4-x)/RI\_4;

FI\_y5 = K(5)\*(YO\_5-x)/RI\_5;

Fx = FG\_x + FI\_x1 + FI\_x2 + FI\_x3 + FI\_x4 + FI\_x5 ;

Fy = FG\_y + FI\_y1 + FI\_y2 + FI\_y3 + FI\_y4 + FI\_y5 ;

out(1) = dxdt;

out(2) = Fx;

out(3) = dydt;

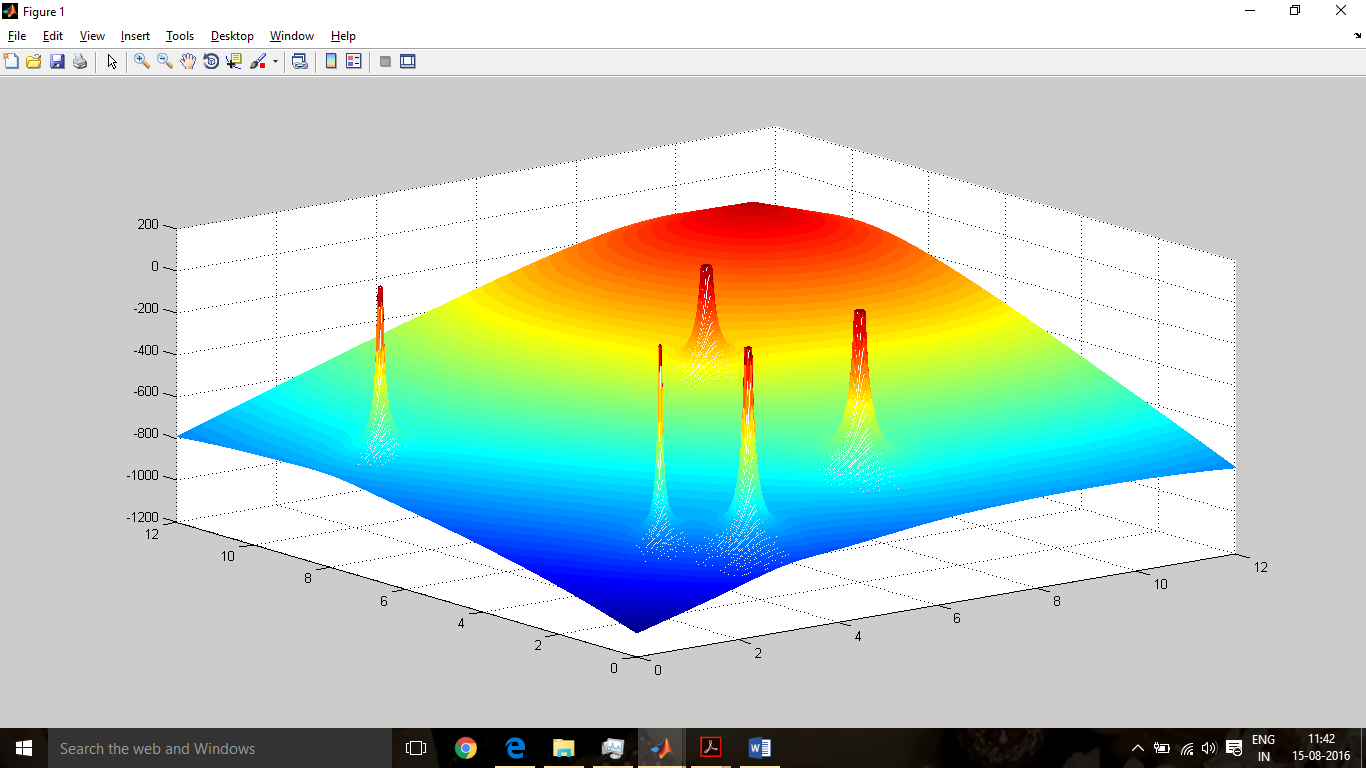
out(4) = Fy;

out = out';

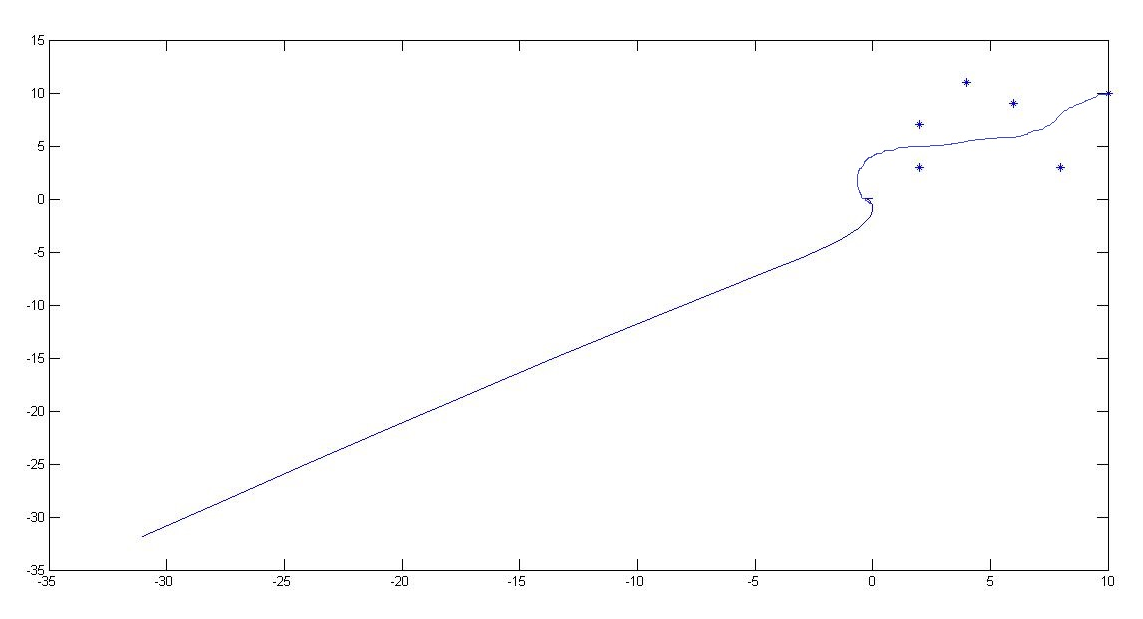
end

**OUTPUT**

**Problem 1**



**For problem 2**



**CONCLUSION**

This is a behaviour based control. But we also have rules based on which we decide when to use these behaviours. We can see that while using the obstacle as repulsive force and goal/target as attractive force we are able to guide the robot to its target with partial impunity. The potential field’s topology us used to search for the target. Although this may not work with dense obstacles; we have implemented a basic way to move the robot control system.