Term 081

King Fahd University of Petroleum and Minerals

EE 656: Robotics & Control

HW #8

Harmonic Potential Field Motion Planning

Homework 8

By

Mohammad Shahab

227598

For

Dr. Ahmad Masoud

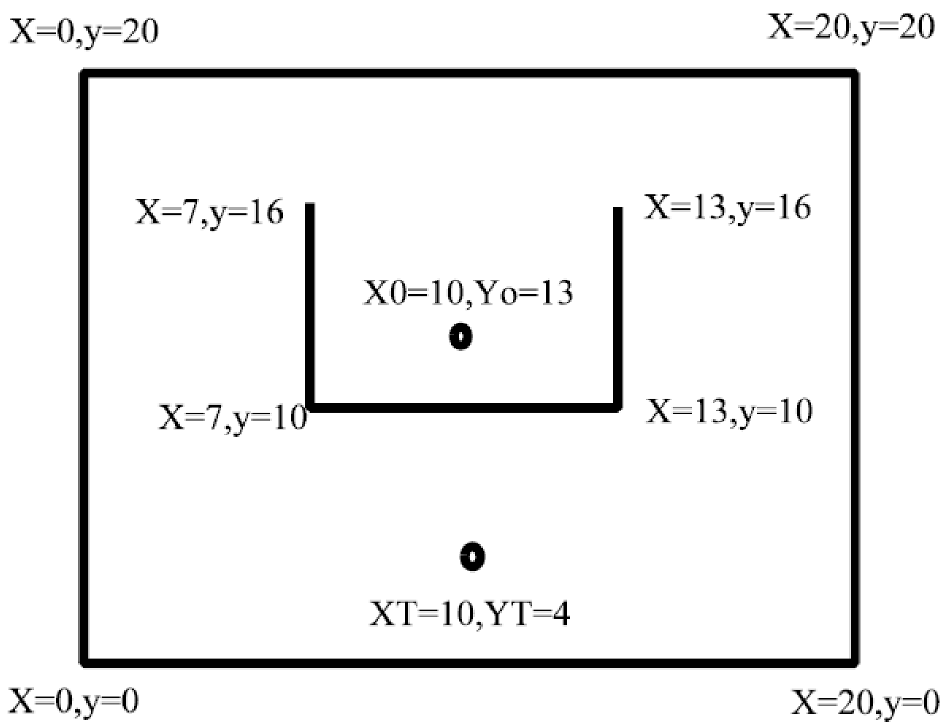
07 February, 2009

EE 656: Robotics & Control

HW# 8

Harmonic Potential Field Motion Planning

We have the this workspace



The black lines correspond to borders (obstacles) in the space. The target position is

$$\boldsymbol{x}_T = \begin{bmatrix} 10 \\ 4 \end{bmatrix}$$

So to build the vector field that will steer motion in $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$, we need to solve the **Partial Differential Equation Boundary Value Problem** (PDE BVP) in Drichilet Setting. The Problem formulated as:

$$\nabla^2 \mathrm{V}(\mathbf{x}) = 0$$

$$with$$

$$V(\boldsymbol{x}_T) = 0$$

$$V(\{\boldsymbol{x} \in \partial O\}) = 1$$

Then, motion will be steered as

$$\dot{\boldsymbol{x}} = -\nabla V(\boldsymbol{x})$$

So,

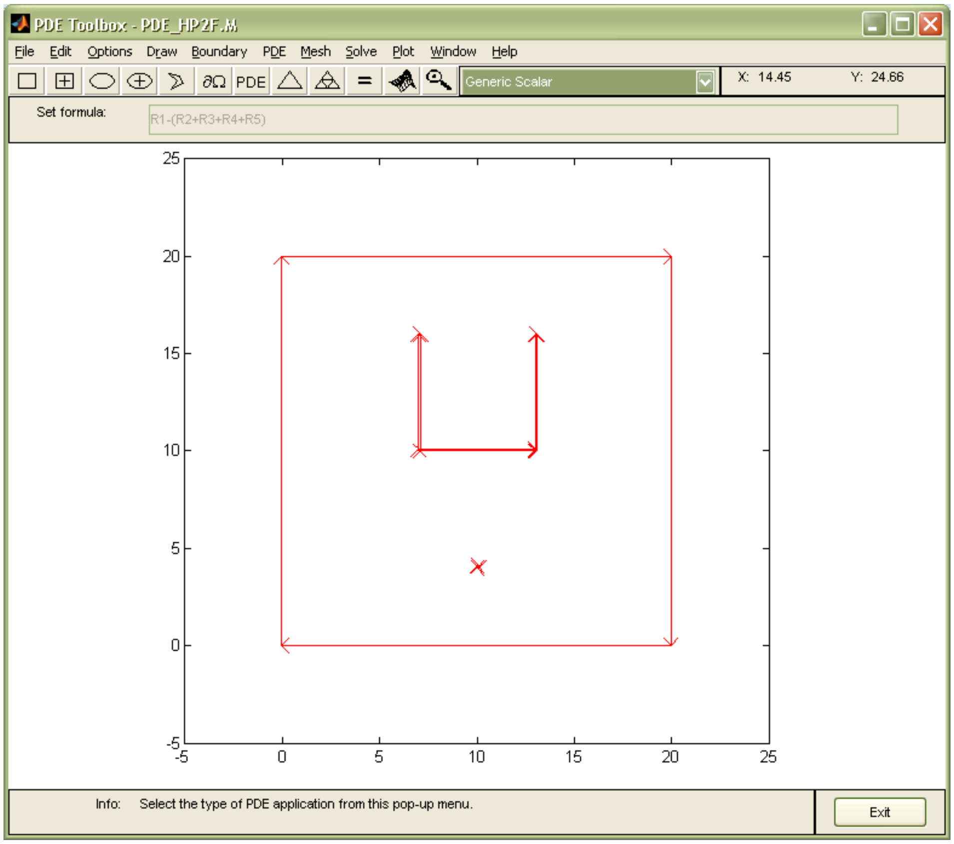
$$\dot{x} = -\frac{\partial V}{\partial x}$$

$$\dot{y} = -\frac{\partial V}{\partial y}$$

First: Solve PDE BVP

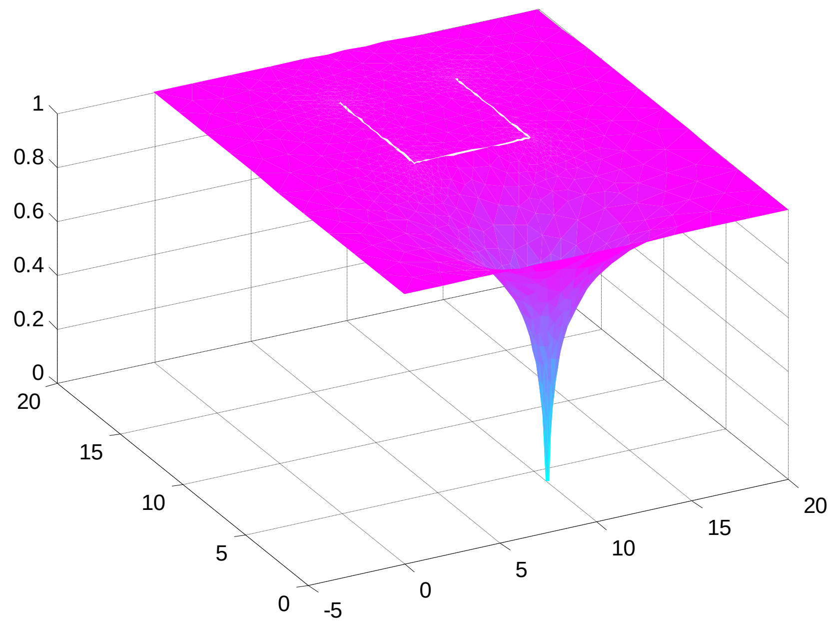
So, to find the 'surface' V(x), MATLAB's PDE toolbox GUI tool is used.

1. The space is structured as shown in figure below.

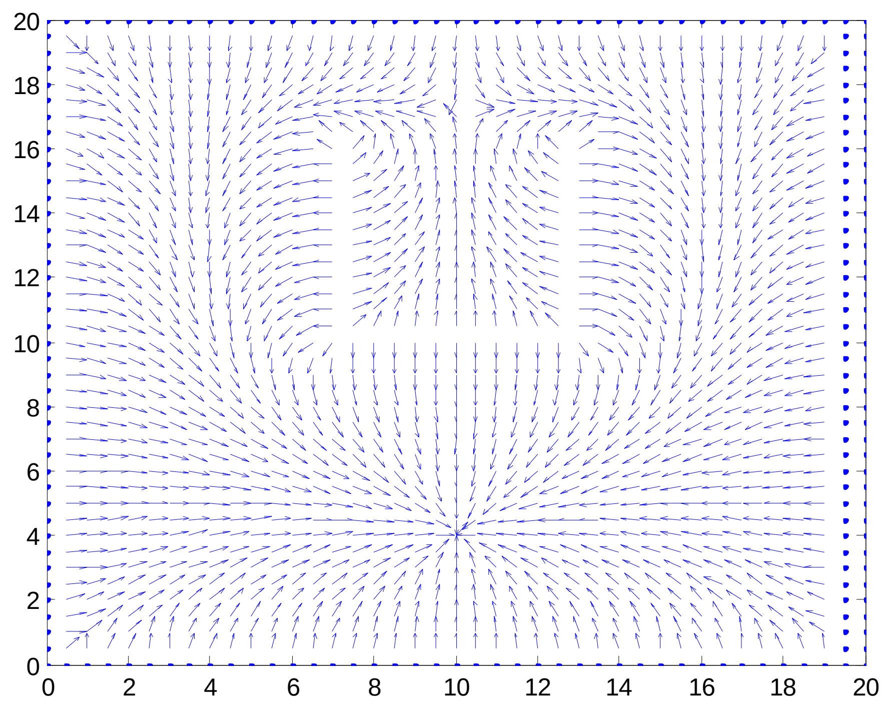


2. Then, boundary values are set as in Drichilet setting. With
   - Target: the alone point (marked X @ (10,4)), $V(\boldsymbol{x}_T) = 0$
   - Borders: other 'red' borders, $V(\{\boldsymbol{x} \epsilon \partial O\}) = 1$

3. The next step is to configure the PDE in:
   - $-div(c \cdot \nabla \mathrm{V}(\mathbf{x})) = 0 \equiv \nabla^2 \mathrm{V}(\mathbf{x}) = 0$
4. Solving the problem will produce the surface below.
5. The Solution of the PDE is exported to MATLAB workspace and saved to a '.mat' file



6. The environment structure is saved as an m-file.
7. The vector field generated is shown below



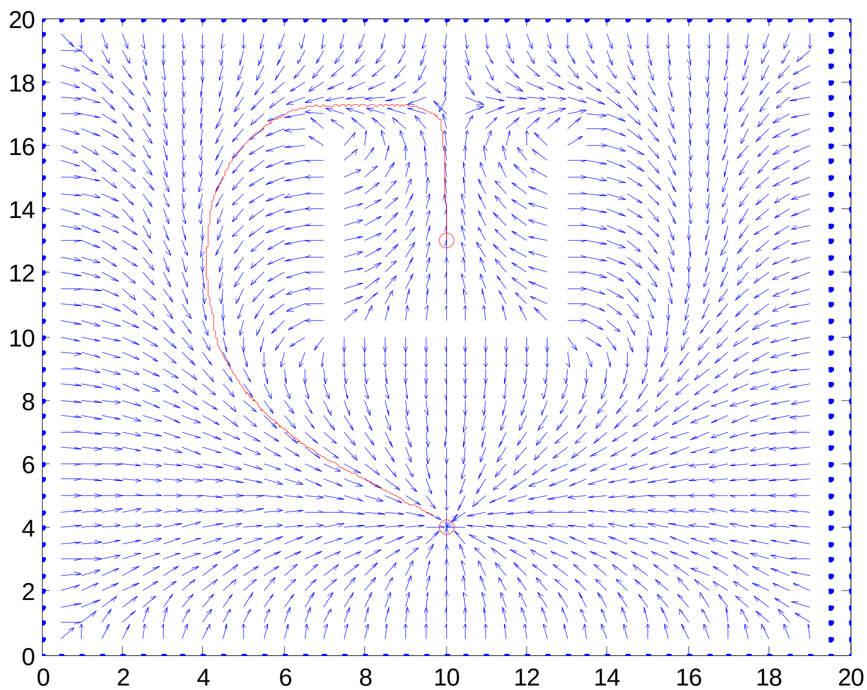
Second: Motion Generation

Motion is generated by

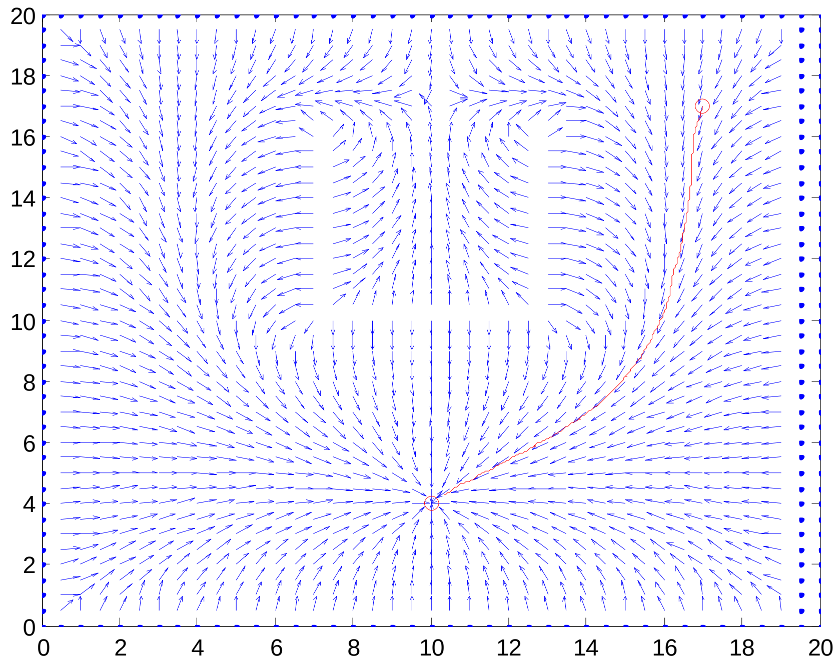
$$\dot{\boldsymbol{x}} = -\nabla V(\boldsymbol{x})$$

The available vector field can generate motion for any initial position of the robot.

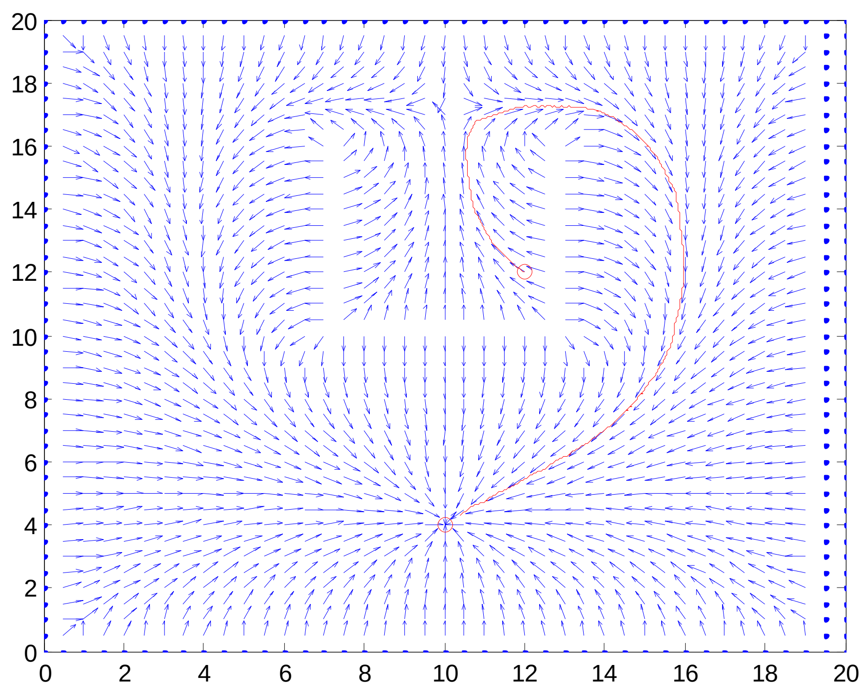
- For us here, the **required start position** is $\boldsymbol{x}_0 = \begin{bmatrix} 10 \\ 13 \end{bmatrix}$. You can see the path generated in figure below



- ***Let us try*** other start position for the same 'field', $\boldsymbol{x}_0 = \begin{bmatrix} 17 \\ 17 \end{bmatrix}$



- Another test for $\boldsymbol{x}_0 = \begin{bmatrix} 12 \\ 12 \end{bmatrix}$



Appendix: MATLAB files

Motion Generation m-file (***NOTE***: PDE solution should be provided in '.mat' file to be loaded in start of program)

```
clear all
close all
clc

load hpf_rob

x0=[12;12];

xT=[10;4];

dh=.05;

x=-5:10*dh:35;
y=-5:10*dh:35;

[X,Y]=meshgrid(x,y);

[FX,FY]=pdegrad(p,t,Vx);

% fx=tri2grid(p,t,FX',x,y);
% fy=tri2grid(p,t,FY',x,y);

Vxy=tri2grid(p,t,Vx,x,y);

pdesurf(p,t,Vx)

[fx,fy]=gradient(Vxy,dh);

nor1=sqrt(fx.^2+fy.^2);

FX=-fx./nor1;
FY=-fy./nor1;

figure
quiver(x,y,FX,FY,.7)
axis([0 20 0 20])

hold on
plot(x0(1),x0(2),'ro',xT(1),xT(2),'ro')

hold off

ss=1;
k=1;
xp=[];
yp=[];
xp(1)=x0(1);
yp(1)=x0(2);
ix=[];
iy=[];
jx=[];
jy=[];

fx1=[];
fy1=[];

while ss
```

```
    Pw=sqrt(((X-xp(k)).^2)+((Y-yp(k)).^2));
    xw(k)=min(min(Pw));

    [iix,iiy]=find(Pw==xw(k));



         ix(k)=iix(1);
         iy(k)=iiy(1);

         fx1(k)=FX(ix(k),iy(k));
         fy1(k)=FY(ix(k),iy(k));



    xp(k+1)=xp(k)+dh*(fx1(k));
    yp(k+1)=yp(k)+dh*(fy1(k));




    if (sqrt((xp(k+1)-xT(1)).^2+(yp(k+1)-xT(2)).^2)<=0.5)
        ss=0;
    end

    k=k+1;
end

hold
plot([xp],[yp],'r')
```