Robot Homework 1

1.How to generate uniform, perpendicular, attractive, repulse, tangential forces for a robot and obstacles with known positions? (Provide related mathematical formulas)

For uniform: [u v] = k[0, 1]
For perpendicular: [u v] = k[0, 1]
For attractive: [u v] = -k[x, y]
For repulse: [u v] = k[x, y]
For tangential forces: [u v] = k[-y, x]
where u v is the vector component in x and y direction. k is a constant, in next example, I let k = 1.

2.Please simulate the above force fields, and plot the vector force fields. (provide codes and plots of force fields)

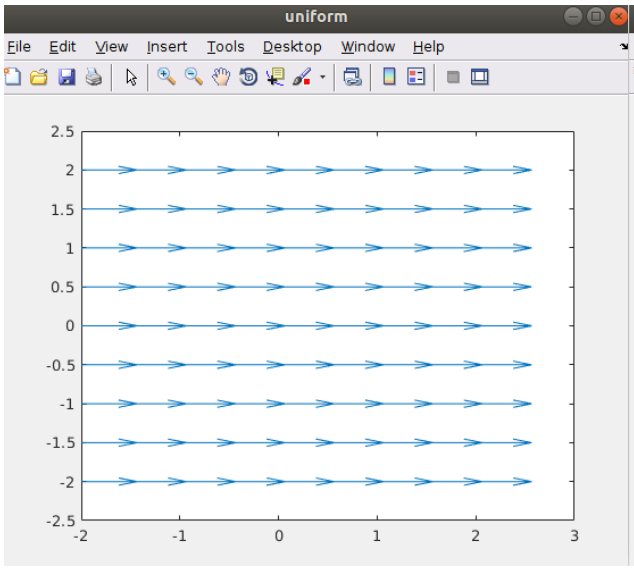
```
function field(x, y, u, v, title)
%FIELD Summary of this function goes here
%   Detailed explanation goes here
% uniform

figure('NumberTitle', 'off', 'Name', title);
quiver(x,y,u,v)
end
```

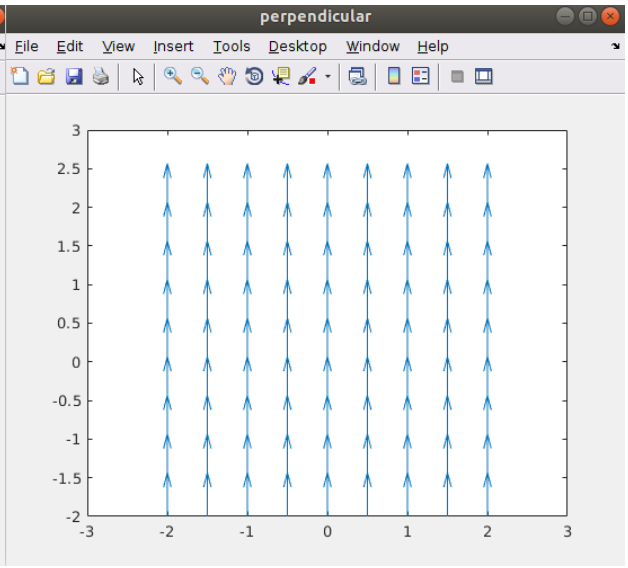
this function is writen in another .m file.

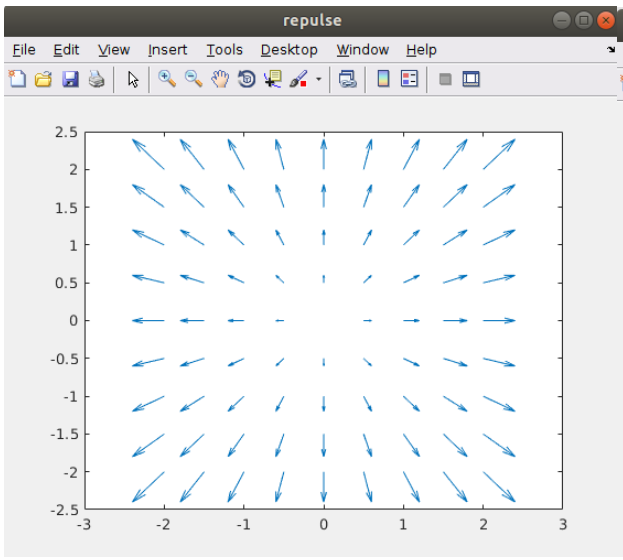
```
% plot uniform, perpendicular, attractive, repulse,
% tangential force fields
[x,y] = meshgrid(-2:0.5:2,-2:0.5:2);

% uniform
u = 0.*x + 1;
v = 0.*y;
field(x, y, u, v, 'uniform')

% perpendicular
u = 0.*x;
v = 0.*y + 1;
field(x, y, u, v, 'perpendicular')

% attractive
u = -x;
v = -y;
field(x, y, u, v, 'attractive')

% repulse
u = x;
```

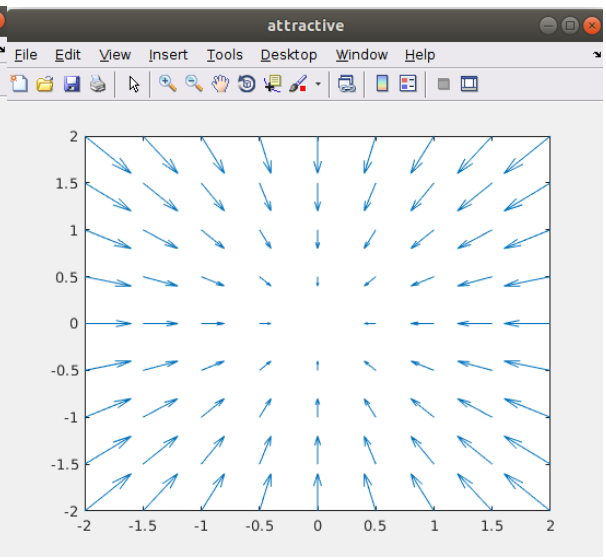
```
v = y;
field(x, y, u, v, 'repulse')

% tangential
u = y;
v = -x;
field(x, y, u, v, 'tangential')
```

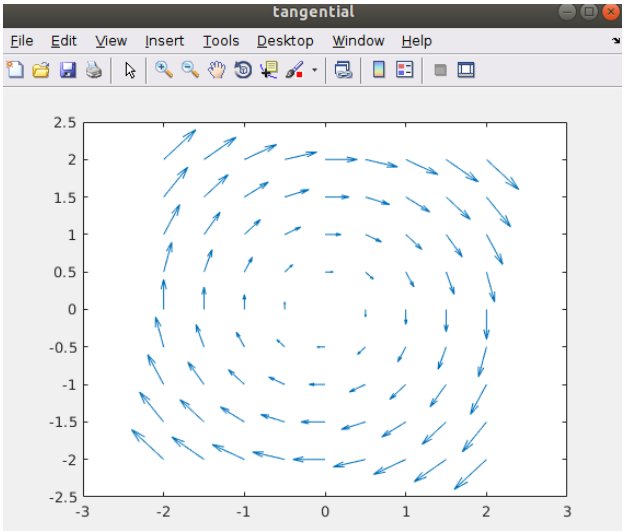
the figures are shown below:











3.Please simulate the motions of a robot for given those force fields. (Provide codes and Plots of simulation results)

vector.m | field.m | simu.m

```
function simu()
clc
close all
clear all
%% ============ Set the paramters ========
T=0.01; % Sampling Time
k=2; % Sampling counter
x(k-1)=-0.9; % initilize the state x
y(k-1)=-1; % initilize the state y
u(k-1)=x(k-1);
v(k-1)=y(k-1);
theta(k-1)=0; % initilize the state theta
tfinal=5; % final simulation time
t=0; % intilize the time

xmin=-2; % setting the figure limits
xmax=2;
ymin=-2;
ymax=2;
[xp,yp] = meshgrid(xmin:0.5:xmax,ymin:0.5:ymax);

% uniform
up = 0*xp;
vp = 0*xp+1;
%====================================
%% ============ The main loop ==========
```

```
while(t<=tfinal)
t=t+T; % increase the time

% define by different force field
x(k)=u(k-1)*T+x(k-1); % calculating x
y(k)=v(k-1)*T+y(k-1); % calculating y
u(k)=0;
v(k)=1;
theta(k)=atan(v(k)/u(k)); % calculating theta
draw_robot(x, y, up, vp); % Draw the robot and it's path
k=k+1; % increase the sampling counter
end
%====================================
%% === Draw the mobile robot & Path ====
function draw_robot(x, y, u, v)

mob_L=0.2; % The Mobile Robot length
mob_W=0.1; % The Mobile Robot width
Tire_W=0.05; % The Tire width
Tire_L=mob_L/2;  % The Tire length
plot(x,y,'-r') % Dawing the Path
axis([xmin xmax ymin ymax]) % setting the figure limits
axis square
hold on

quiver(xp,yp,u,v) %绘制二维矢量场图
```

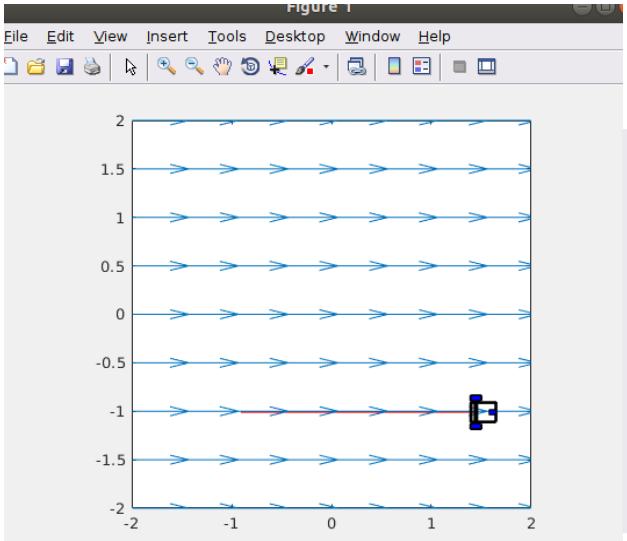
```
% Body
v1=[mob_L;-mob_W];
v2=[-mob_L/4;-mob_W];
v3=[-mob_L/4;mob_W];
v4=[mob_L;mob_W];
%Right Tire
v5=[Tire_L/2;-mob_W-0.02];
v6=[Tire_L/2;-mob_W-Tire_W-0.02];
v7=[-Tire_L/2;-mob_W-Tire_W-0.02];
v8=[-Tire_L/2;-mob_W-0.02];
%Left Tire
v9=[Tire_L/2;mob_W+0.02];
v10=[Tire_L/2;mob_W+Tire_W+0.02];
v11=[-Tire_L/2;mob_W+Tire_W+0.02];
v12=[-Tire_L/2;mob_W+0.02];
%Line
v13=[0;-mob_W-0.02];
v14=[0;mob_W+0.02];
%Front Tire
v15=[mob_L;Tire_W/2];
v16=[mob_L;-Tire_W/2];
v17=[mob_L-Tire_L/1.5;-Tire_W/2];
v18=[mob_L-Tire_L/1.5;Tire_W/2];

R=[cos(theta(k)) -sin(theta(k));sin(theta(k)) cos(theta(k))];
P=[x(k);y(k)]; % Position Matrix
```

```

v1=R*v1+P;
v2=R*v2+P;
v3=R*v3+P;
v4=R*v4+P;

v5=R*v5+P;
v6=R*v6+P;
v7=R*v7+P;
v8=R*v8+P;

v9=R*v9+P;
v10=R*v10+P;
v11=R*v11+P;
v12=R*v12+P;

v13=R*v13+P;
v14=R*v14+P;

v15=R*v15+P;
v16=R*v16+P;
v17=R*v17+P;
v18=R*v18+P;


%Body
```

```
105 -   mob_x=[v1(1) v2(1) v3(1) v4(1) v1(1)];
106 -   mob_y=[v1(2) v2(2) v3(2) v4(2) v1(2)];
107 -   plot(mob_x,mob_y,'-k','linewidth',2)
108
109     %Right Tire
110 -   mob_x=[v5(1) v6(1) v7(1) v8(1) v5(1)];
111 -   mob_y=[v5(2) v6(2) v7(2) v8(2) v5(2)];
112 -   plot(mob_x,mob_y,'-k','linewidth',2)
113 -   fill(mob_x,mob_y,'b')
114
115     %Left Tire
116 -   mob_x=[v9(1) v10(1) v11(1) v12(1) v9(1)];
117 -   mob_y=[v9(2) v10(2) v11(2) v12(2) v9(2)];
118 -   plot(mob_x,mob_y,'-k','linewidth',2)
119 -   fill(mob_x,mob_y,'b')
120
121     %Line Between tires
122 -   mob_x=[v13(1) v14(1)];
123 -   mob_y=[v13(2) v14(2)];
124 -   plot(mob_x,mob_y,'-k','linewidth',3)
125
126     %Front tire
127 -   mob_x=[v15(1) v16(1) v17(1) v18(1) v15(1)];
128 -   mob_y=[v15(2) v16(2) v17(2) v18(2) v15(2)];
129 -   plot(mob_x,mob_y,'-k','linewidth',1)
130 -   fill(mob_x,mob_y,'b')
```

```
131
132 -   drawnow
133 -   hold off
134 -   end
135     %=====================================
136
137
138
139 -   end
```

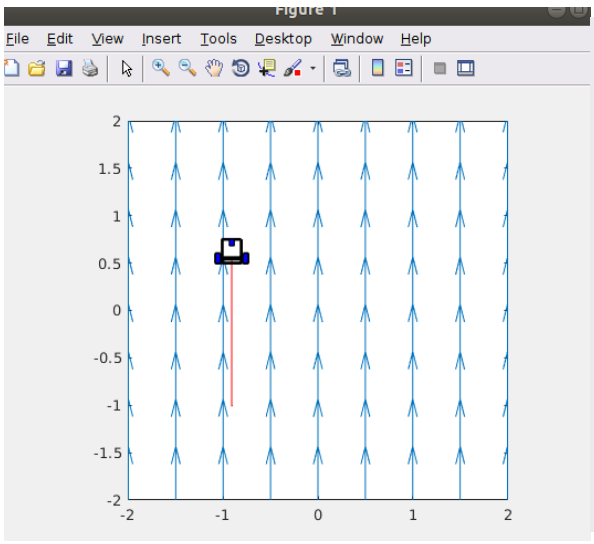
Those above is my code to simulate the motions of the robot. Most of code is provided by prof. Haoqi. The key code is in line 22~38, especially for line 23-24 and 31-35.
Figure for uniform:



key code:

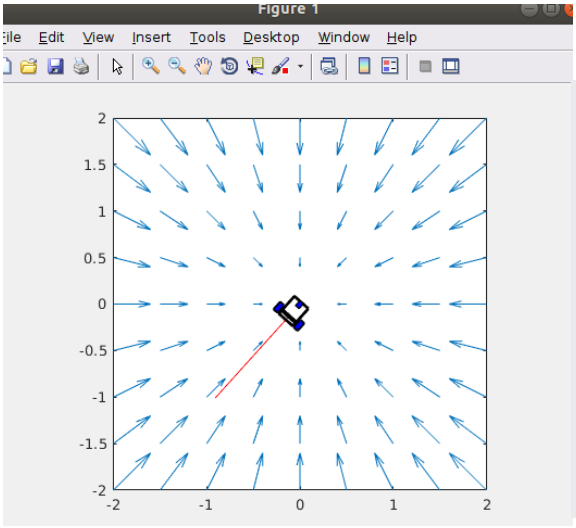
```
22      % uniform
23 -    up = 0*xp+1;
24 -    vp = 0*xp;
25      %=====================================
26      %% ========== The main loop ==========
27 -    while(t<=tfinal)
28 -    t=t+T; % increase the time
29
30      % define by different force field
31 -    x(k)=u(k-1)*T+x(k-1); % calculating x
32 -    y(k)=v(k-1)*T+y(k-1); % calculating y
33 -    u(k)=1;
34 -    v(k)=0;
35 -    theta(k)=atan(v(k)/u(k)); % calculating theta
36 -    draw_robot(x, y, up, vp); % Draw the robot and it's path
37 -    k=k+1; % increase the sampling counter
38 -    end
```

Figure for perpendicular



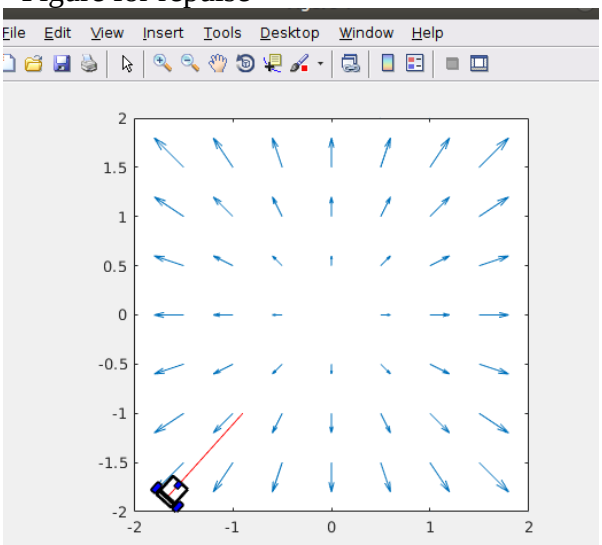

```
22      % uniform
23 -    up = 0*xp;
24 -    vp = 0*xp+1;
25      %=====================================
26      %% ========== The main loop ==========
27 -    while(t<=tfinal)
28 -    t=t+T; % increase the time
29
30      % define by different force field
31 -    x(k)=u(k-1)*T+x(k-1); % calculating x
32 -    y(k)=v(k-1)*T+y(k-1); % calculating y
33 -    u(k)=0;
34 -    v(k)=1;
35 -    theta(k)=atan(v(k)/u(k)); % calculating th
36 -    draw_robot(x, y, up, vp); % Draw the robot
37 -    k=k+1; % increase the sampling counter
38 -    end
```

Figure for attractive



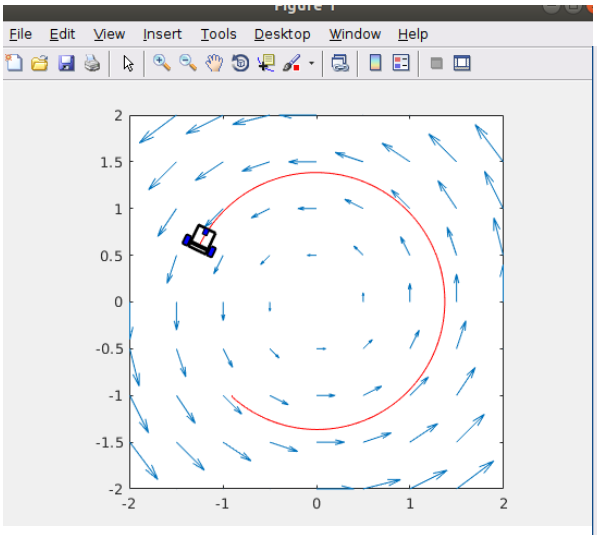

```
22    % uniform
23 -  up = -xp;
24 -  vp = -yp;
25    %=====================================
26    %% ========== The main loop ==========
27 -  while(t<=tfinal)
28 -  t=t+T; % increase the time
29
30    % define by different force field
31 -  x(k)=u(k-1)*T+x(k-1); % calculating x
32 -  y(k)=v(k-1)*T+y(k-1); % calculating y
33 -  u(k)=-x(k);
34 -  v(k)=-y(k);
35 -  theta(k)=atan(v(k)/u(k)); % calculating theta
36 -  draw_robot(x, y, up, vp); % Draw the robot and it
37 -  k=k+1; % increase the sampling counter
38 -  end
```

Figure for repulse




```
22    % uniform
23 -  up = xp;
24 -  vp = yp;
25    %=====================================
26    %% ========== The main loop ==========
27 -  while(t<=tfinal)
28 -  t=t+T; % increase the time
29
30    % define by different force field
31 -  x(k)=u(k-1)*T+x(k-1); % calculating x
32 -  y(k)=v(k-1)*T+y(k-1); % calculating y
33 -  u(k)=x(k);
34 -  v(k)=y(k);
35 -  theta(k)=atan(v(k)/u(k)); % calculating t
36 -  draw_robot(x, y, up, vp); % Draw the robo
37 -  k=k+1; % increase the sampling counter
38 -  end
```

Figure for tangential forces




```
22    % uniform
23 -  up = -yp;
24 -  vp = xp;
25    %=====================================
26    %% ========== The main loop ==========
27 -  while(t<=tfinal)
28 -  t=t+T; % increase the time
29
30    % define by different force field
31 -  x(k)=u(k-1)*T+x(k-1); % calculating x
32 -  y(k)=v(k-1)*T+y(k-1); % calculating y
33 -  u(k)=-y(k);
34 -  v(k)=x(k);
35 -  theta(k)=atan(v(k)/u(k)); % calculating
36 -  draw_robot(x, y, up, vp); % Draw the robo
37 -  k=k+1; % increase the sampling counter
38 -  end
```