IR_HW_1_REPORT

Question 1

- Attractive field: V=-k*P, if r<=p<=R
- repulsive field: V=k*P, if r<=p<=R
- uniform field: V=V_0
- perpendicular field: V=B_0, B_0*P=c
- tangential field: V=Omega_0 X Omega_0

Question 2 & Question 3

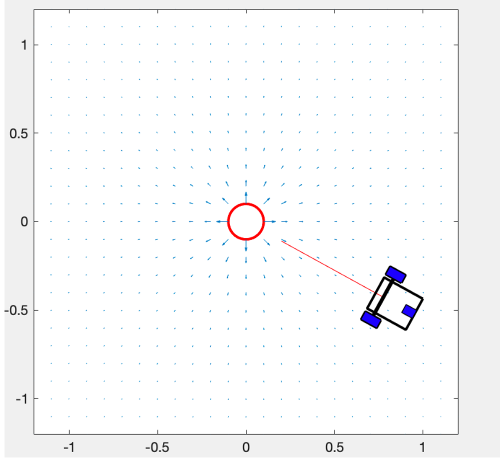
Using common framework

Show force field:

```
%% ===show force field ====
function show_field()
[vec_x,vec_y] = meshgrid(-1.2:0.1:1.2, -1.2:0.1:1.2);
[u,v] = att(vec_x,vec_y);
viscircles([0,0],0.1);
quiver(vec_x,vec_y,u,v,0.5);
hold on
end
% =
```

Name   Figure   Field code: att() function   Simulate code

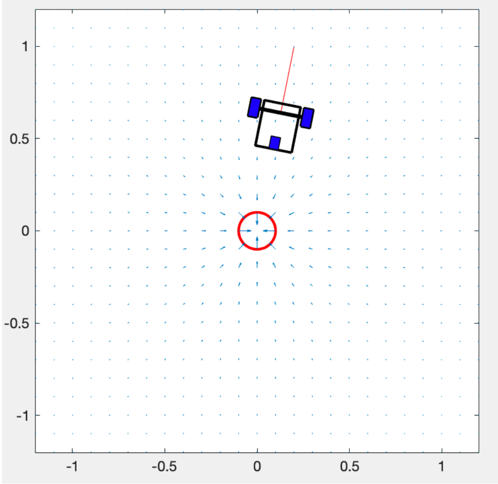
repulsive



```
function [ox,oy] = att(ix,iy)
    dis = 1./sqrt(ix.^2+iy.^2);
    thea = atan(iy./ix);
    ox = dis.*sign(ix).*cos(abs(thea));
    oy = dis.*sign(iy).*sin(abs(thea));
end
```

```
V=0.01; %V ÊÇËÙ¶È£¬WÊÇ½ÇËÙ¶È
while(t<=tfinal)
t=t+T; % increase the time
[t_x,t_y]=att(x(k-1),y(k-1));
x(k)=t_x*V+x(k-1); % calculating x
y(k)=t_y*V+y(k-1); % calculating y
theta(k)= atan(y(k)./x(k)); % calculating
theta
a = t_y*V
% y(k)
draw_robot(); % Draw the robot and it's path
k=k+1; % increase the sampling counter
if abs(y(k-1))<0.1
    break
end
end
```

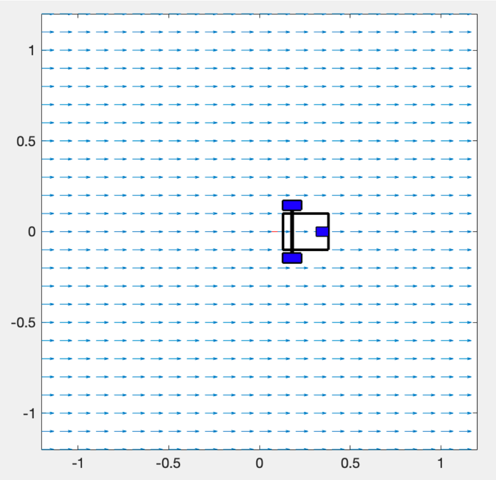
Attractive



```
function [ox,oy] = att(ix,iy)
    dis = 1./sqrt(ix.^2+iy.^2);
    thea = atan(iy./ix);
    ox = -
dis.*sign(ix).*cos(abs(thea));
    oy = -
dis.*sign(iy).*sin(abs(thea));
 end
```

```
V=0.01; %V ÊÇËÙ¶È£¬WÊÇ½ÇËÙ¶È
while(t<=tfinal)
t=t+T; % increase the time
[t_x,t_y]=att(x(k-1),y(k-1));
x(k)=t_x*V+x(k-1); % calculating x
y(k)=t_y*V+y(k-1); % calculating y
theta(k)= atan(y(k)./x(k))+pi; % calculating
theta
a = t_y*V
% y(k)
draw_robot(); % Draw the robot and it's path
k=k+1; % increase the sampling counter
if abs(y(k-1))<0.3
    break
end
end
```

uniform



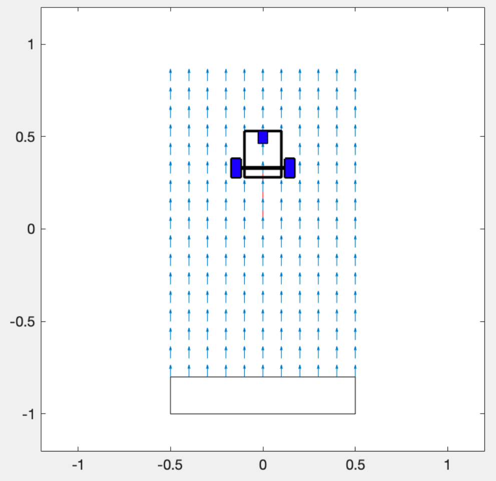
```
%% === Show uniform force field ====
function show_uniform()
[vec_x,vec_y] = meshgrid(-1.2:0.1:1.2,
-1.2:0.1:1.2);
u = V*ones(size(vec_x));
v = 0*ones(size(vec_x));
quiver(vec_x,vec_y,u,v,0.5);
end
% =
```

```
%% =========== The main loop ==========
while(t<=tfinal)
t=t+T; % increase the time
V=0.5;; %V ÊÇËÙ¶È£¬WÊÇ½ÇËÙ¶È
theta(k)=theta(k-1); % calculating theta
x(k)=V*1*T+x(k-1); % calculating x
y(k)=y(k-1); % calculating y
draw_robot(); % Draw the robot and it's path
k=k+1; % increase the sampling counter
end
%=====================================
```

```
%% === Show perpendicular force field
====
function show_perpendicular()
[vec_x,vec_y] = meshgrid(-0.5:0.1:0.5,
-0.8:0.1:0.8);
```

```
%% =========== The main loop ==========
V=0.5; %V is velocity
```

perpendicular

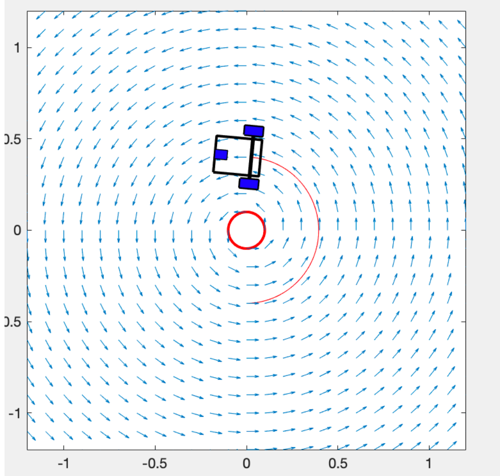


```
u = 0*ones(size(vec_x));
v = V*ones(size(vec_x));
rectangle('Position',[-0.5,-1,1,0.2])
quiver(vec_x,vec_y,u,v,0.5);
hold on

if y(k)>1
    t=tfinal+1
end
end
% =
```

```
while(t<=tfinal)
t=t+T; % increase the time
theta(k)=theta(k-1); % calculating theta
x(k)=x(k-1); % calculating x
y(k)=V*1*T+y(k-1); % calculating y
draw_robot(); % Draw the robot and it's path
k=k+1; % increase the sampling counter
end
```

tangential



```
function [ox,oy] = att(ix,iy)
    thea = atan(iy./ix)-0.001;
    oy = sign(ix).*cos(abs(thea));
    ox = -sign(iy).*sin(abs(thea));
end
```

```
%% =========== The main loop ==========
V=10;W=25; %V ÊÇËÙ¶È£¬WÊÇ½ÇËÙ¶È
while(t<=tfinal)
t=t+T; % increase the time
theta(k)=W*T+theta(k-1); % calculating theta
x(k)=V*cos(theta(k))*T+x(k-1); % calculating x
y(k)=V*sin(theta(k))*T+y(k-1); % calculating y
draw_robot(); % Draw the robot and it's path
k=k+1; % increase the sampling counter
% if abs(y(k-1))<0.1
%     break
% end
end
%====================================
```