EE 5322 Homework 4.
Mobile Robot Control and Potential Fields.

Victor G. Lopez

1. Potential Field.

A work area for a ground mobile robot is simulated. The attractive potential field for the target in $(10, 10)$ is defined as

$$V_G = K_G r_G$$

where K_G is a design gain and r_G is the distance between the robot and the target:

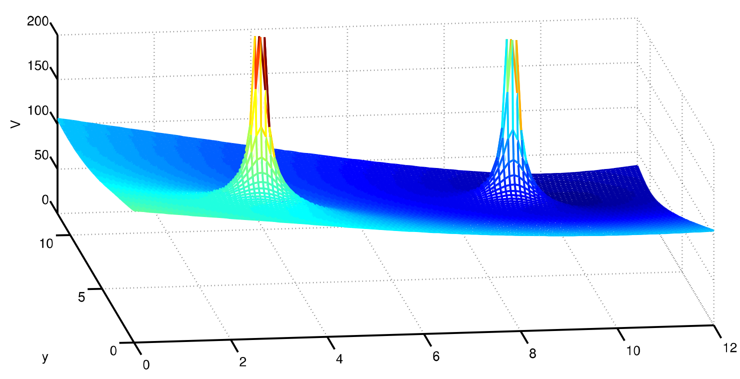
$$r_G = sqrt((10 - x)^2 + (10 - y)^2)$$

Two obstacles in $(3, 3)$ and $(9, 9)$ have a repulsive potential field following the equation

$$V_o = \frac{K_o}{r_o}$$

where K_o and r_o are the equivalents for the gain and the distance from the robot, respectively.

For this exercise, a gain of $K_G = 10$ was selected for the goal potential field and a gain of $K_o = 20$ was used for both obstacles. We can determine the total potential field of the scene by adding this three individual fields: ($V = V_G + V_{o1} + V_{o2}$). The result is plotted using Matlab's *mesh* function and the result is shown in the following image.



The employed Matlab code is now presented:

```
close all
clear all
```

```
clc

xf=0:0.1:12;
yf=0:0.1:12;
[X Y]=meshgrid(xf,yf);

Goal=[10;10];
Obs1=[3;3];
Obs2=[9;9];
KG=10;
Ko=20;

rG=sqrt((Goal(1)-X).^2+(Goal(2)-Y).^2);
VG=KG*rG;  %Potential field for the goal

ro1=sqrt((Obs1(1)-X).^2+(Obs1(2)-Y).^2);
Vo1=Ko./ro1;  %Potential field for the first obstacle

ro2=sqrt((Obs2(1)-X).^2+(Obs2(2)-Y).^2);
Vo2=Ko./ro2;  %Potential field for the second obstacle

mesh(xf,yf,VG+Vo1+Vo2)
axis([0 12 0 12 0 200])
xlabel('x')
ylabel('y')
zlabel('V')
```

2. Potential Field Navigation.

A mobile robot, with the following dynamics, is now included:

$$\dot{x} = v_T cos(\phi) cos(\theta)$$
$$\dot{y} = v_T cos(\phi) sin(\theta)$$
$$\dot{\theta} = \frac{v_T}{L} sin(\phi)$$

where the velocity v_T is constant and the steering angle ϕ is the control input.

The same potential fields of the last exercise are considered. Then, the atraction force components from the goal are

$$F_{Gx} = -\frac{\partial V_G}{\partial x} = K_G \frac{(10 - x)}{r_G}$$
$$F_{Gy} = -\frac{\partial V_G}{\partial y} = K_G \frac{(10 - y)}{r_G}$$

and the repulsion forces of the obstacles are computed as

$$F_{oix} = -\frac{\partial V_{oi}}{\partial x} = -K_o \frac{(x_{oi} - x)}{r_{oi}^3}$$
$$F_{oiy} = -\frac{\partial V_{oi}}{\partial y} = -K_o \frac{(xoi - y)}{r_{oi}^3}$$

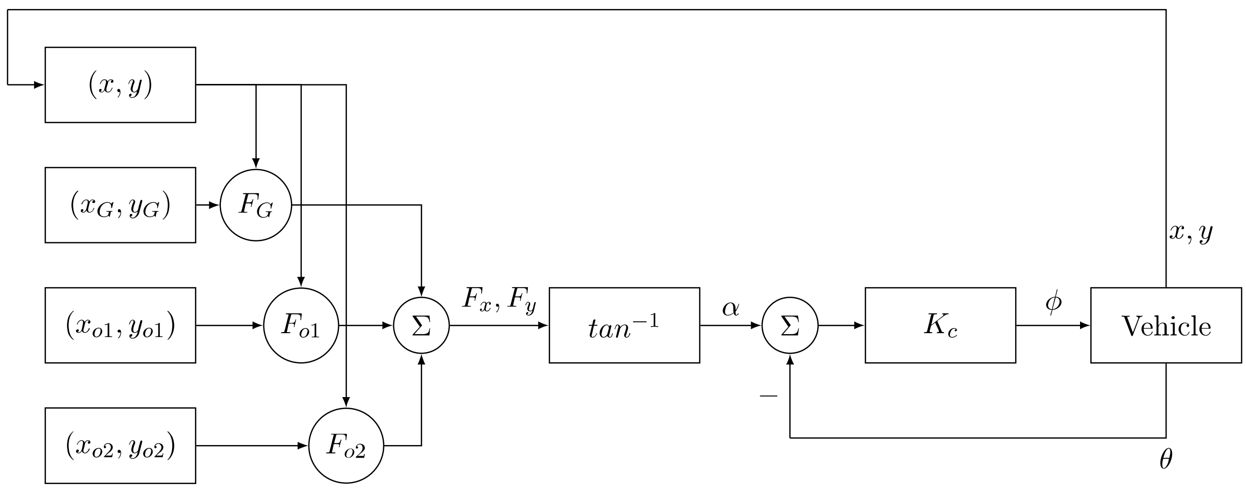
The total force components afecting the vehicle are $F_x = F_{Gx} + F_{o1x} + F_{o2x}$ and $F_y = F_{Gy} + F_{o1y} + F_{o2y}$. The desired heading angle can be computed as

$$\alpha = tan^{-1}(\frac{F_y}{F_x})$$

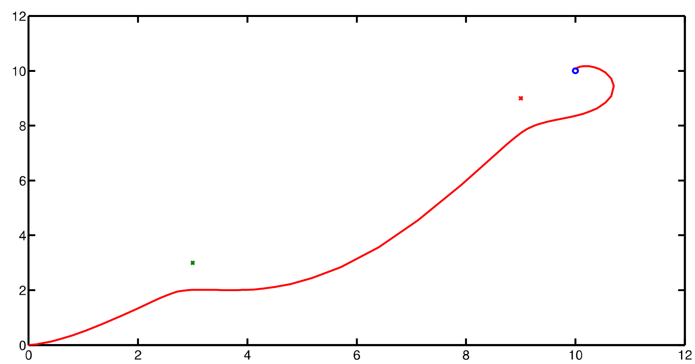
A proportional controller is designed to make the robot arrive to the target. Then, the control input is

$$\phi = K_c(\alpha - \theta)$$

where K_c is the control gain for the system. A block diagram of the control scheme is



Simulation is performed taking $K_G = 30$, $K_o = 30$, $K_c = 2$, $v_T = 1$ and $L = 2$. As the velocity is not involved in the control law computation, another line of code is added to make the vehicle stop when it is close enough to the goal. Initial conditions for the position and initial heading angle of the robot are all set to zero. The system response is shown in the next figure.



It can be seen that the robot arrives to the goal succesfully. The complete Matlab code is shown. Note that the code was structured as two nested functions in one file to avoid the necesity of two Matlab files to use the *ode23* function; the code runs correctly this way, but it can also be easily separated.

```
function [t,x]=H4_PotentialNavigation()

clc

time=[0 300];
```

```
x0=[0;0;0];
[t x]=ode23(@vehicle,time,x0);
plot(x(:,1),x(:,2),'r',Goal(1),Goal(2),'o',Obs1(1),Obs1(2),'x',Obs2(1),Obs2(2),'x')
axis([0 12 0 12])

    function dx=vehicle(t,x)

        Goal=[10;10];
        Obs1=[3;3];
        Obs2=[9;9];
        KG=30;
        Ko=30;

        rG=sqrt((Goal(1)-x(1))^2+(Goal(2)-x(2))^2);
        FGx=KG*(Goal(1)-x(1))/rG;
        FGy=KG*(Goal(2)-x(2))/rG;

        ro1=sqrt((Obs1(1)-x(1))^2+(Obs1(2)-x(2))^2);
        Fo1x=-Ko*(Obs1(1)-x(1))/ro1^3;
        Fo1y=-Ko*(Obs1(2)-x(2))/ro1^3;

        ro2=sqrt((Obs2(1)-x(1))^2+(Obs2(2)-x(2))^2);
        Fo2x=-Ko*(Obs2(1)-x(1))/ro2^3;
        Fo2y=-Ko*(Obs2(2)-x(2))/ro2^3;

        Fx=(FGx+Fo1x+Fo2x);
        Fy=(FGy+Fo1y+Fo2y);

        alpha=atan(Fy/Fx);

        v=1; L=2;
        if rG<0.05 %Stop the vehicle when arrives to the goal
            v=0;
        end
        K=2;
        ph=K*(alpha-x(3));
        dx=[v*cos(ph)*cos(x(3));v*cos(ph)*sin(x(3));v*sin(ph)/L];

    end
end
```

3. Platoon of Mobile Robots.

Two more vehicles are added to form a platoon, with the same objective of arrive to the target. One of the robots will be the leader, which know where the goal is and tries to reach it, avoiding the obstacles on the way as usual. The other two robots don't know where the target is, but they have to follow the leader, avoid the obstacles and stay away from each other. Although they have to follow the leader, they should keep their distance from it.

In this case, the dynamics of each robots are

$$\ddot{x} = F_x$$

$$\ddot{y} = F_y$$

Then, to simulate the system we need to express it in state space equations. Taking $x_1 = x$,

$x_2 = \dot{x}$, $y_1 = y$ and $y_2 = \dot{y}$, the decoupled dynamics can be written as

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_x$$

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_y$$

where F_x and F_y are the forces computed from the potential fields. This equations are employed for each one of the mobile robots.

Aditional to the forces of the previous exercise, some other forces will be considered. The followers will approach the leader using the potential field

$$V_{iL} = \frac{1}{2} K_{iL}(r_{iL} - r_D)^2$$

where r_{iL} is the distance between vehicle i and the leader, and r_D is the desired separation selected as $r_D = 0.5$. Then, the force components of this potential field are computed as

$$F_{iLx} = -\frac{\partial V_{iL}}{\partial x} = K_{iL}(r_{iL} - r_D)\frac{(x_L - x)}{r_{iL}}$$

$$F_{iLy} = -\frac{\partial V_{iL}}{\partial y} = K_{iL}(r_{iL} - r_D)\frac{(y_L - y)}{r_{iL}}$$

where we are considering that the leader position is (x_L, y_L). The followers must also have a potential field that keeps them separate from each other, and this field is selected as

$$V_{ij} = \frac{K_{ij}}{r_{ij}^2}$$

and the corresponding forces are

$$F_{ijx} = -\frac{\partial V_{ij}}{\partial x} = -2K_{ij}\frac{(x_j - x_i)}{r_{ij}^4}$$

$$F_{ijy} = -\frac{\partial V_{ij}}{\partial y} = -2K_{ij}\frac{(y_j - y_i)}{r_{ij}^4}$$

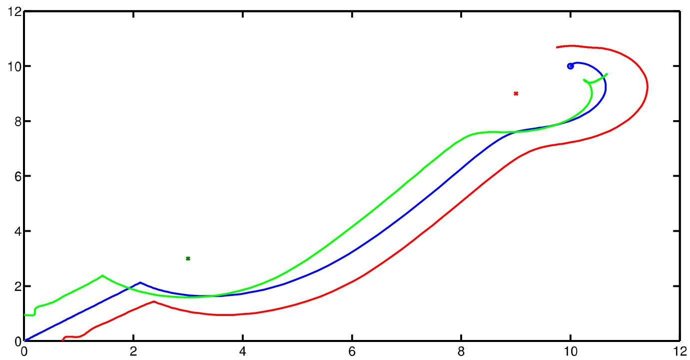
It is important to note that this equations are for the follower i, so that the negative sign produces a repulsive force from follower j. Finally, friction forces are considered to prevent the vehicles to oscillate around their goal. This forces are taken proportional to the velocity of the robot:

$$F_{rx} = -K_F\dot{x}$$

$$F_{ry} = -K_F\dot{y}$$

For the system simulation, the gain values are selected as $K_G = 20$, $K_o = 30$, $K_{Li} = 100$, $K_{ij} = 20$, $K_F = 10$. To avoid having the vehicles trapped in a local minima, small random movements are included when the system velocity is equal to zero. All initial conditions are set to zero. Random movements are also included to separate the followers from the initial condition. Divisions by zero are prevented by setting 0.01 as the minimal value for each fraction denominator; for example:

$$F_{Gx} = K_G\frac{10 - x}{r_G} \quad \rightarrow \quad F_{Gx} = K_G\frac{10 - x}{max(r_G, 0.01)}$$



The response of the system is shown in the image above. It can be seen that the leader (blue line) reaches the goal and the other vehicles follow it and keep their apropriate distance. The Matlab code for this exercise is also presented.

```
function [t,x]=H4_PlatoonRobots()

clc

time=[0 300];
x0=[0;0;0;0;0;0;0;0;0;0;0;0];

[t x]=ode23(@vehicles,time,x0);
plot(x(:,1),x(:,3),'b',x(:,5),x(:,7),'r',x(:,9),x(:,11),'g',...
    Goal(1),Goal(2),'o',Obs1(1),Obs1(2),'x',Obs2(1),Obs2(2),'x')
axis([0 12 0 12])

    function dx=vehicles(t,x)
        Goal=[10;10];
        Obs1=[3;3];
        Obs2=[9;9];
        KG=20;
        Ko=30;
        Kij=20;
        KLi=100;
        KF=10;
        rD=0.5;  %Separation from the leader

        rG=sqrt((Goal(1)-x(1))^2+(Goal(2)-x(3))^2);  %Leader and goal
        FGx=KG*(Goal(1)-x(1))/max([rG 0.01]);  %Avoid division by zero
        FGy=KG*(Goal(2)-x(3))/max([rG 0.01]);

        ro1=sqrt((Obs1(1)-x(1))^2+(Obs1(2)-x(3))^2);  %Leader and Obstacle1
        Fo1x=-Ko*(Obs1(1)-x(1))/ro1^3;
        Fo1y=-Ko*(Obs1(2)-x(3))/ro1^3;

        ro2=sqrt((Obs2(1)-x(1))^2+(Obs2(2)-x(3))^2);  %Leader and Obstacle2
        Fo2x=-Ko*(Obs2(1)-x(1))/ro2^3;
        Fo2y=-Ko*(Obs2(2)-x(3))/ro2^3;

        FrLx=-KF*x(2);
```

```
FrLy=-KF*x(4);

Fx=(FGx+Fo1x+Fo2x+FrLx);
Fy=(FGy+Fo1y+Fo2y+FrLy);

xL=[0 1;0 0]*[x(1);x(2)]+[0;1]*Fx;  %Leader dynamics
yL=[0 1;0 0]*[x(3);x(4)]+[0;1]*Fy;

r1o1=sqrt((Obs1(1)-x(5))^2+(Obs1(2)-x(7))^2);  %Follower1 and Obs1
F1o1x=-Ko*(Obs1(1)-x(5))/r1o1^3;
F1o1y=-Ko*(Obs1(2)-x(7))/r1o1^3;

r1o2=sqrt((Obs2(1)-x(5))^2+(Obs2(2)-x(7))^2);  %Follower1 and Obs2
F1o2x=-Ko*(Obs2(1)-x(5))/r1o2^3;
F1o2y=-Ko*(Obs2(2)-x(7))/r1o2^3;

rL1=sqrt((x(1)-x(5))^2+(x(3)-x(7))^2);  %Follower1 and Leader
FL1x=KLi*(rL1-rD)*(x(1)-x(5))/max([rL1 0.01]);
FL1y=KLi*(rL1-rD)*(x(3)-x(7))/max([rL1 0.01]);

r21=sqrt((x(9)-x(5))^2+(x(11)-x(7))^2);  %Follower 1 and Follower2
F21x=-2*Kij*(x(9)-x(5))/max([r21^4 0.01]);
F21y=-2*Kij*(x(11)-x(7))/max([r21^4 0.01]);

Fr1x=-KF*x(6);
Fr1y=-KF*x(8);

F1x=(F1o1x+F1o2x+FL1x+F21x+Fr1x);
F1y=(F1o1y+F1o2y+FL1y+F21y+Fr1y);

x1=[0 1;0 0]*[x(5);x(6)]+[0;1]*F1x;  %Follower1 dynamics
y1=[0 1;0 0]*[x(7);x(8)]+[0;1]*F1y;

r2o1=sqrt((Obs1(1)-x(9))^2+(Obs1(2)-x(11))^2);  %Follower2 and Obs1
F2o1x=-Ko*(Obs1(1)-x(9))/r2o1^3;
F2o1y=-Ko*(Obs1(2)-x(11))/r2o1^3;

r2o2=sqrt((Obs2(1)-x(9))^2+(Obs2(2)-x(11))^2);  %Follower2 and Obs2
F2o2x=-Ko*(Obs2(1)-x(9))/r2o2^3;
F2o2y=-Ko*(Obs2(2)-x(11))/r2o2^3;

rL2=sqrt((x(1)-x(9))^2+(x(3)-x(11))^2);  %Follower2 and Leader
FL2x=KLi*(rL2-rD)*(x(1)-x(9))/max([rL2 0.01]);
FL2y=KLi*(rL2-rD)*(x(3)-x(11))/max([rL2 0.01]);

r12=sqrt((x(5)-x(9))^2+(x(7)-x(11))^2);  %Follower2 and Follower1
F12x=-2*Kij*(x(5)-x(9))/max([r12^4 0.01]);
F12y=-2*Kij*(x(7)-x(11))/max([r12^4 0.01]);

Fr2x=-KF*x(10);
Fr2y=-KF*x(12);

F2x=(F2o1x+F2o2x+FL2x+F12x+Fr2x);
F2y=(F2o1y+F2o2y+FL2y+F12y+Fr2y);

x2=[0 1;0 0]*[x(9);x(10)]+[0;1]*F2x;  %Follower2 dynamics
y2=[0 1;0 0]*[x(11);x(12)]+[0;1]*F2y;
```

```
        dx=[xL(1);xL(2);yL(1);yL(2);x1(1);x1(2);y1(1);y1(2);x2(1);x2(2);...
            y2(1);y2(2)];

        if x(2)==0 && x(4)==0 %Avoid local minima
            w=0.001*randn(2,1);
            dx=dx+[0;w(1);0;w(2);0;0;0;0;0;0;0;0];
        end

        if r12==0 %Separate followers from initial condition
            w=0.001*randn(4,1);
            dx=dx+[0;0;0;0;0;w(1);0;w(2);0;w(3);0;w(4)];
        end

    end
end
```