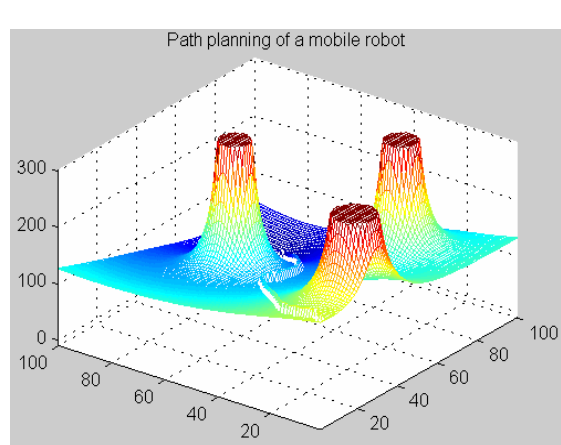
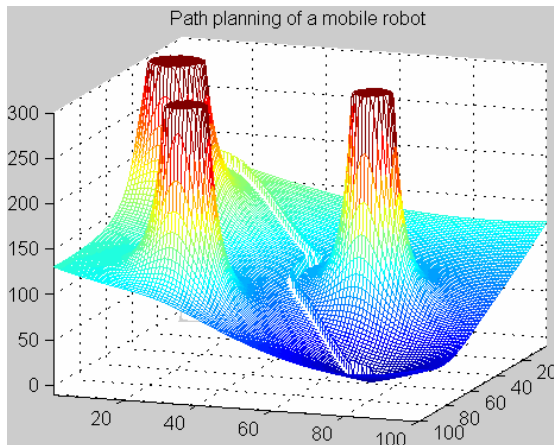
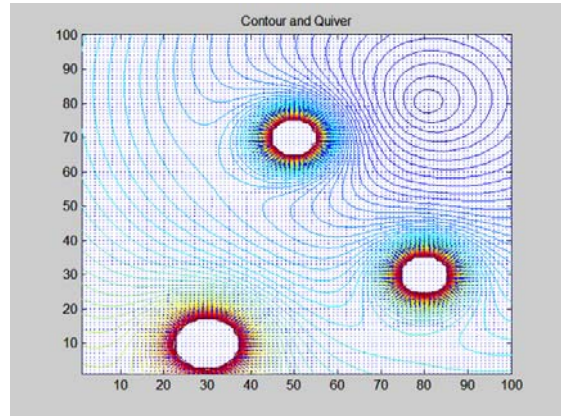
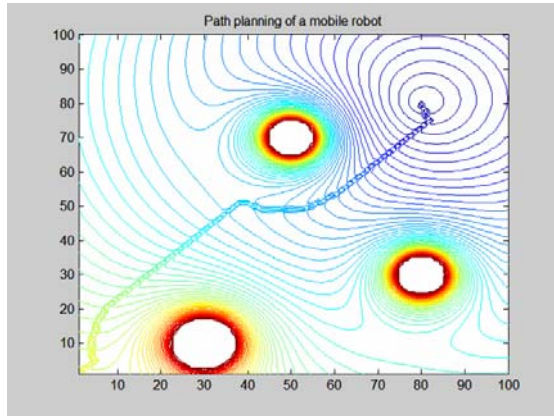


The lower MATLAB functions display four potential field plots showing reaction and attraction forces to obstacles and goal locations respectively.



```
function [A] = Mobile2(K1,XY1,K2,XY2,K3,XY3,Kg,XYGoal)
% Example showing Potential Field attraction/reaction forces.
% The checking function used in this code will return the nearest
% deeper location (a1,b1) around current location (a,b) in
% a 3D plane which altitude/depth is saved in matrix A.

% 10 x 10 Matrix grid converted to a 100 x 100 grid !!

% [A] = Mobile2(90,[3,1],70,[5,7],70,[8,3],15,[8,8]);
% [A] = Mobile2(90,[3,1],70,[3,5],70,[8,8],15,[7,4]);
% [A] = Mobile2(90,[3,1],70,[2,6],70,[5,5],15,[8,8]);
% [A] = Mobile2(90,[3,1],70,[3,5],70,[5,10],15,[10,10]);

y1=XY1(1);
x1=XY1(2);
y2=XY2(1);
x2=XY2(2);
y3=XY3(1);
x3=XY3(2);
GoalY=XYGoal(1);
GoalX=XYGoal(2);

i=1:100;
j=1:100;

for I = 1:100,
    for J = 1:100,
        R1 = (J/10-y1)^2+(I/10-x1)^2; % DIST1^2 is the squared distance from obstacle 1 to current point
        R2 = (J/10-y2)^2+(I/10-x2)^2; % DIST2^2 is the squared distance from obstacle 2 to current point
        R3 = (J/10-y3)^2+(I/10-x3)^2; % DIST3^2 is the squared distance from obstacle 3 to current point
        RG = sqrt((J/10-GoalY)^2+(I/10-GoalX)^2); % DISTG is the distance from current position to Goal
        %Power = SUM of Ki/DISTi^2 + Kg DISTG
        A(I,J) = K1/(R1+.001)+ K2/(R2+.001)+K3/(R3+.001)+ Kg*RG;
        if (A(I,J)>300)
            A(I,J)=300;
        end
    end
end
end
```

```

% Searching the path
a=1;
b=1;
B=zeros(length(A));
value=(abs(a-GoalX*10)+abs(b-GoalY*10));
while value>0,
    [a1,b1] = checking(A,a,b,100,100);
    if ((a1==a)&(b1==b))
        a=a1;
        b=b1;
        B(a,b)=15;
        Break;
    else
        a=a1;
        b=b1;
        B(a,b)=15;
    end
    value=(abs(a-GoalX*10)+abs(b-GoalY*10));
end

%A(i,j)=A(i,j)+B(i,j);

figure(1)
mesh(i,j,A(i,j)+B(i,j))
title('Path planning of a mobile robot')
axis([1,100,1,100,-10,300])
view([-20,-15,20])
figure(2)
mesh(i,j,A(i,j)+B(i,j))
title('Path planning of a mobile robot')
axis([1,100,1,100,-10,300])
view([100,35,30])
figure(3)
contour(i,j,A(i,j)+B(i,j),60)
title('Path planning of a mobile robot')
figure(4)
contour(i,j,A(i,j),60)
[px,py] = gradient(A,3,3);
title('Contour and Quiver')
hold on
quiver(i,j,-px,-py,2);

% End of function "Mobile2"

```

```

function [x,y] = checking(A,a,b,X,Y);
% Usable to mobile robot navigation problem

% b
% ^
% |      A(a,b)
%
% A13 A23 A33
% A12 X A32
% A11 A21 A31  -> a

A11 = 50000;
A12 = 50000;
A13 = 50000;
A21 = 50000;
A23 = 50000;
A31 = 50000;
A32 = 50000;
A33 = 50000;
if (a>=2)&(b>=2)&(a<X)&(b<Y)
    A11 = A(a-1,b-1);
    A21 = A(a,b-1);
    A31 = A(a+1,b-1);
    A12 = A(a-1,b);
    A32 = A(a+1,b);
    A13 = A(a-1,b+1);
    A23 = A(a,b+1);
    A33 = A(a+1,b+1);
elseif (a>=2)&(b>=2)&(a<X) % eliminates b+1
    A11 = A(a-1,b-1);
    A21 = A(a,b-1);
    A31 = A(a+1,b-1);
    A12 = A(a-1,b);
    A32 = A(a+1,b);
elseif (a>=2)&(a<X)&(b<Y) % eliminates b-1
    A12 = A(a-1,b);
    A32 = A(a+1,b);
    A13 = A(a-1,b+1);
    A23 = A(a,b+1);
    A33 = A(a+1,b+1);
elseif (a>=2)&(b>=2)&(b<Y) % eliminates a+1
    A11 = A(a-1,b-1);
    A21 = A(a,b-1);
    A12 = A(a-1,b);
    A13 = A(a-1,b+1);
    A23 = A(a,b+1);
elseif (b>=2)&(a<X)&(b<Y) % eliminates a-1
    A21 = A(a,b-1);
    A31 = A(a+1,b-1);
    A32 = A(a+1,b);
    A23 = A(a,b+1);
    A33 = A(a+1,b+1);
elseif (a==1)&(b==1)
    A32 = A(a+1,b);
    A23 = A(a,b+1);
    A33 = A(a+1,b+1);

```

```

elseif (a==X) & (b==1)
    A12 = A(a-1,b);
    A13 = A(a-1,b+1);
    A23 = A(a,b+1);
elseif (a==1) & (b==Y)
    A21 = A(a,b-1);
    A31 = A(a+1,b-1);
    A32 = A(a+1,b);
elseif (a==X) & (b==Y)
    A11 = A(a-1,b-1);
    A21 = A(a,b-1);
    A12 = A(a-1,b);
elseif (a==1)
    A21 = A(a,b-1);
    A23 = A(a,b+1);
    A31 = A(a+1,b-1);
    A32 = A(a+1,b);
    A33 = A(a+1,b+1);
elseif (a==X)
    A21 = A(a,b-1);
    A23 = A(a,b+1);
    A11 = A(a-1,b-1);
    A12 = A(a-1,b);
    A13 = A(a-1,b+1);
elseif (b==1)
    A12 = A(a-1,b);
    A13 = A(a-1,b+1);
    A23 = A(a,b+1);
    A32 = A(a+1,b);
    A33 = A(a+1,b+1);
elseif (b==Y)
    A11 = A(a-1,b-1);
    A21 = A(a,b-1);
    A31 = A(a+1,b-1);
    A12 = A(a-1,b);
    A32 = A(a+1,b);
end
% A13 A23 A33
% A12 X A32
% A11 A21 A31 =>a
if (A11<A21) & (A11<A31) & (A11<A12) & (A11<A32) & (A11<A13) & (A11<A23) & (A11<A33)
    x=a-1;
    y=b-1;
elseif (A21<A31) & (A21<A12) & (A21<A32) & (A21<A13) & (A21<A23) & (A21<A33)
    x=a;
    y=b-1;
elseif (A31<A12) & (A31<A32) & (A31<A13) & (A31<A23) & (A31<A33)
    x=a+1;
    y=b-1;
elseif (A12<A32) & (A12<A13) & (A12<A23) & (A12<A33)
    x=a-1;
    y=b;
elseif (A32<A13) & (A32<A23) & (A32<A33)
    x=a+1;
    y=b;
elseif (A13<A23) & (A13<A33)
    x=a-1;
    y=b+1;
elseif (A23<A33)
    x=a;
    y=b+1;
else
    x=a+1;
    y=b+1;
end

```