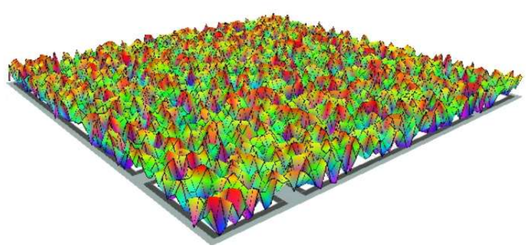
Artificial Potential Fields

Sunday, January 29, 2023 12:02 AM

A typical model field requires model two different types of forces involved:
1) An attractive field generated by the goal
2) A repulsive field generated by obstacles

The potential field is generated when these two force fields are combined.
$U_{total} = U_{att}(X) + U_{rep}(X)$
Where:
U_{total}: Potential Field
U_{att}: Attractive Field
U_{rep}:Repulsive Field

In this example, we will create a attractive field with random value, it looks like a mess so it doesn't have any meaning.

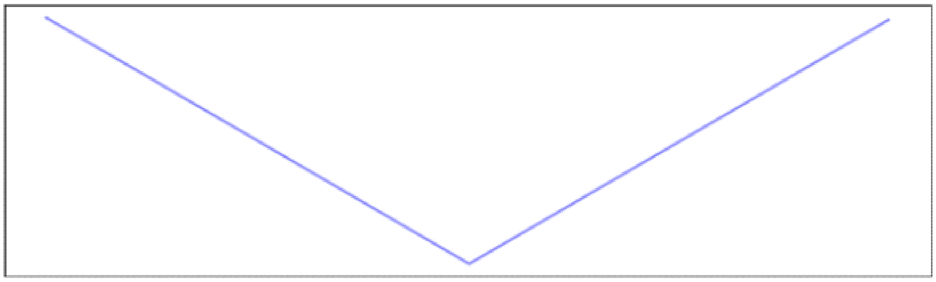


Attractive Potential Fields
The basic idea is to apply an attraction force that can lead either the robot towards the goal. There are several choses on how to implement an attractive field, in all these methods the potential values in the attractive field decreases as it gets closer to the goal. The values in the field should be decrease as it gets closer to the goal.

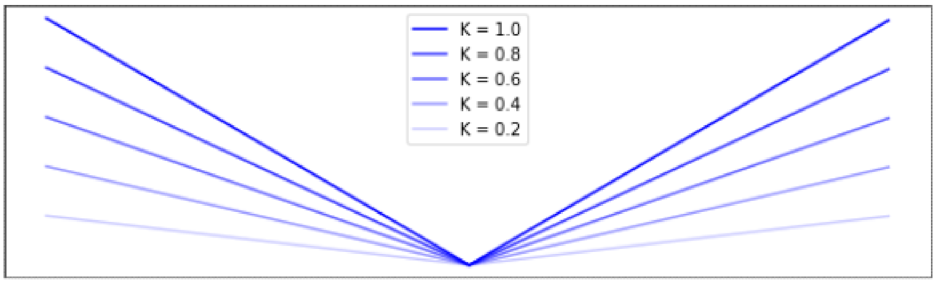
Two types most widely used are canonical and quadratic potential function.

Canonical Potential Field:
In textbooks this field can be expressed as:
$U_{att}(X) = k\|X_{current} - X_{goal}\|$
Where:
k: Attractive Potential Field
$\|X_{current} - X_{goal}\|$: Distance each grid cell and the goal

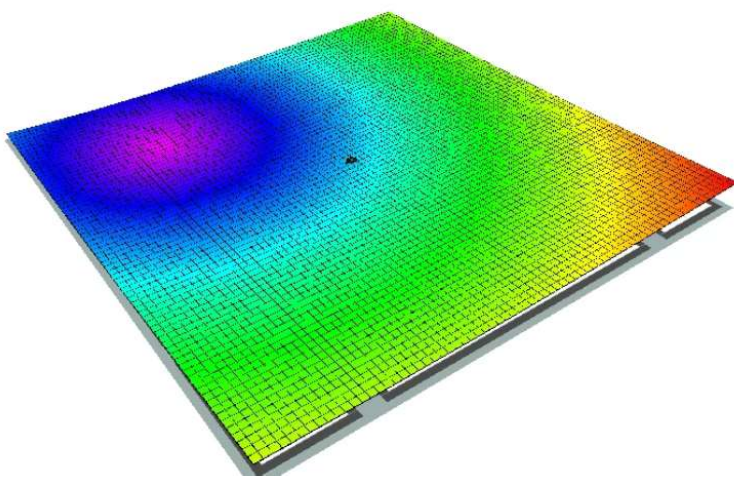
Perhaps the easy way to visualize the canonic potential function is this, a very simple side view of a robot in a one-dimensional map, the slope or derivate represents the intensity of the attractive field:



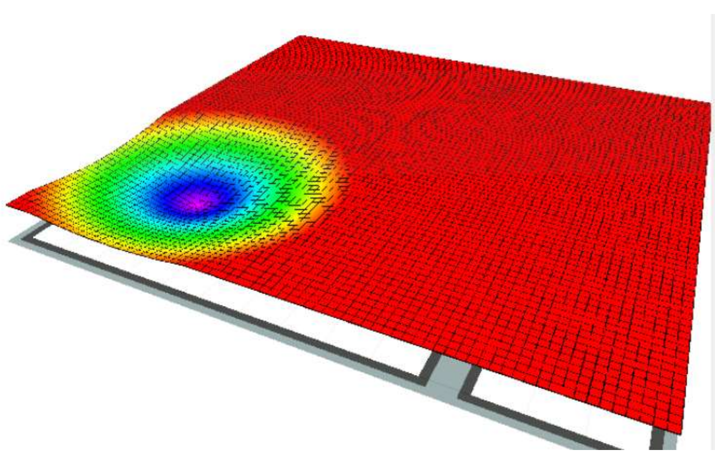
To generate a stronger or more attractive potential field we can change the value of k.



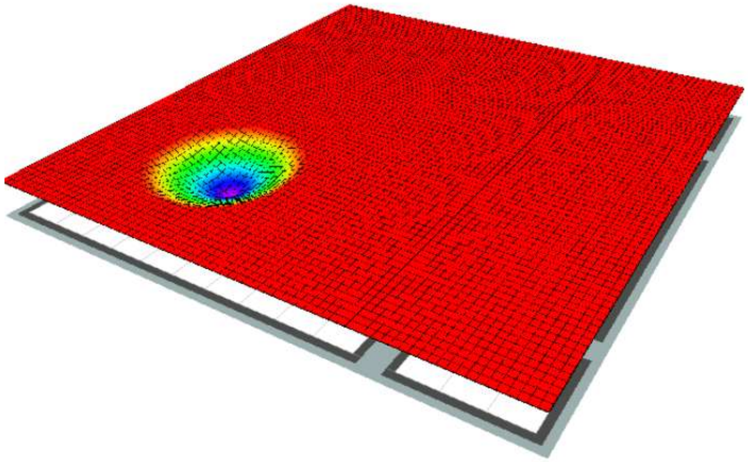
K=1



K=5



K=10

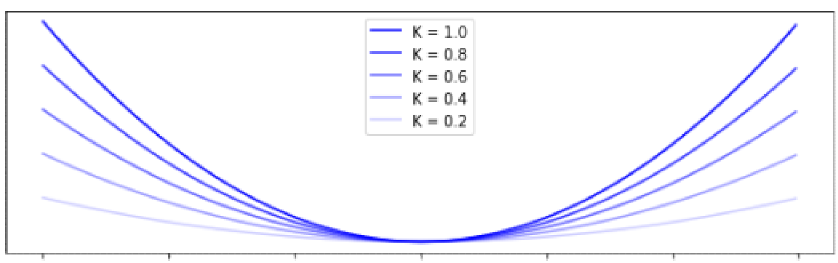


Quadratic Potential Field:

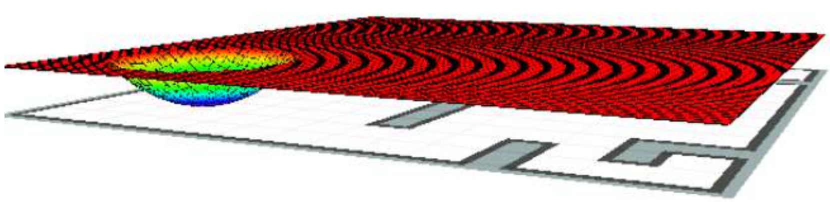
It's another common function to construct an attractive potential field, its formula is very similar to the previous one. Its unique difference that the distance is squared.

$U_{att}(X) = k\left\|X_{current} - X_{goal}\right\|^2$

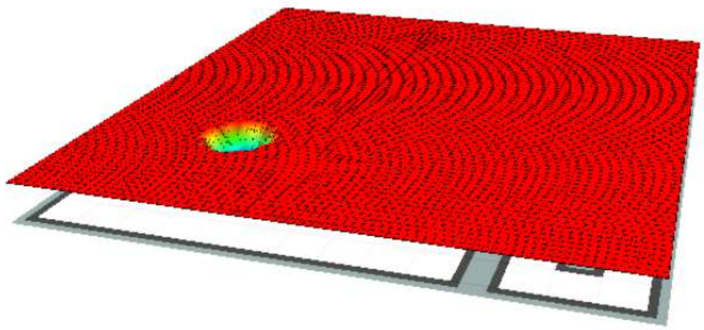
Here we can see how modifying K can affect the potential field



K=1



K=5

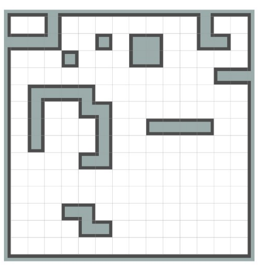


Repulsive Field

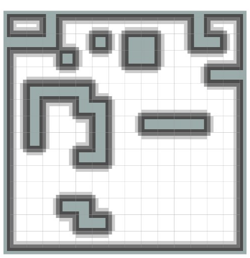
This field pushes away the robot from the obstacles, this field should have the following properties.

1) Never allow a robot collides with obstacles
2) Reinforce keeping an additional safety distance whenever possible
3) Have a limited range of influence

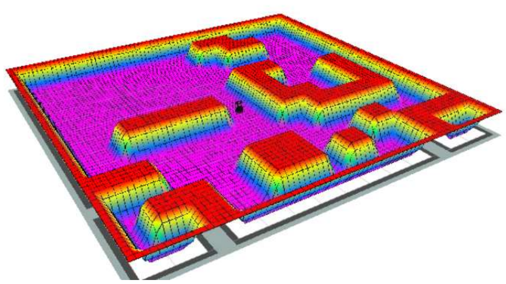
This description is very similar what a cost map does.
Regular Map:



Cost map:



Cost maps are mathematical functions of the environment, which can be modeled using the same procedure used to create repulsive field. A grid cell of a cost map represents a movement cost, whereas in a repulsive field it represents a repulsive force value.

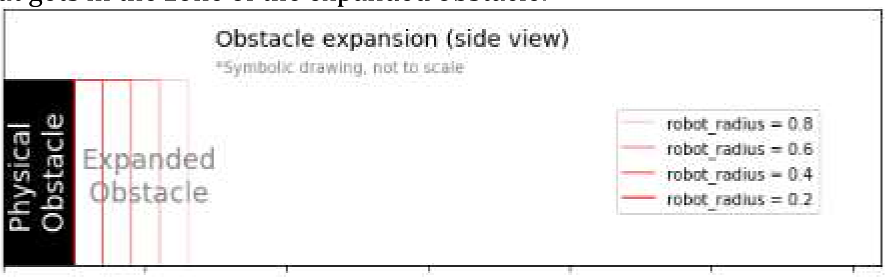


Repulsive Field Parameters:

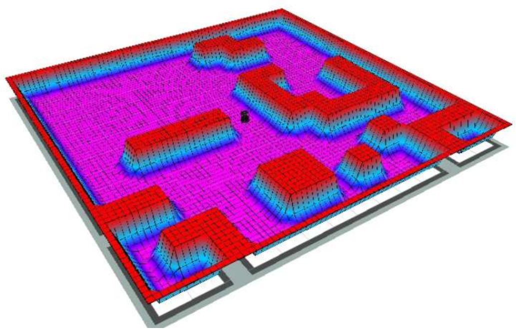
1) A virtual expansion of the size map obstacles, making them thicker than the physical obstacle
2) An additional buffer zone that decreases with growing distance from expanded obstacle (it is also called obstacle inflation)

Expanding size of map obstacles

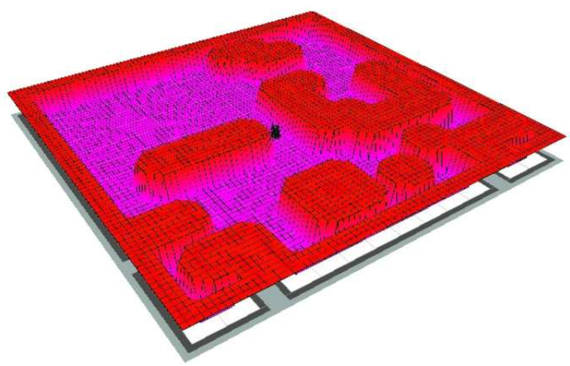
Typically obstacles are expanded by at least half of the robot's width, because a robot's path indicate position, where the center of the robot's base will be moving. This way you avoid the path that gets in the zone of the expanded obstacle.



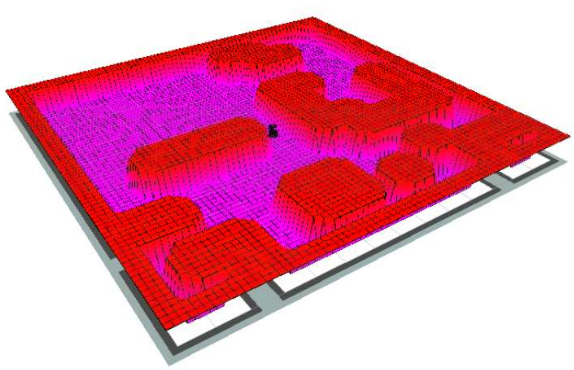
Robot Radius=0



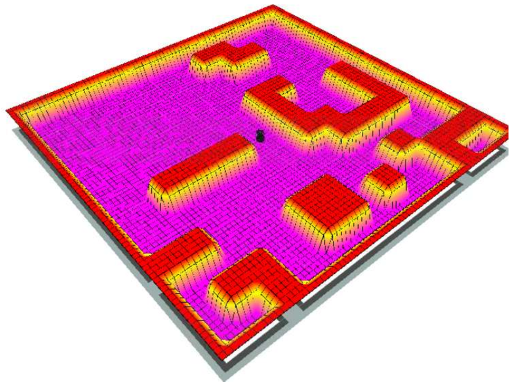
Robot Radius = 0.5



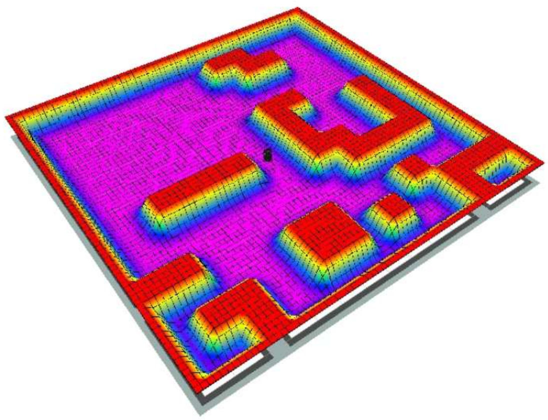
Robot Radius = 1



Inflation Radius = 0.2



Inflation Radius = 1



Additional Buffer Zone

The buffer zone signals that it's preferable to keep an even larger distance from obstacles when possible. If that's not possible, it's okay that the robot's distance to obstacles is lessor. This is achieved by using a decaying function, or a function whose value declines as its distance from an obstacle increases.

There are different forms of decaying functions. For example, the **negative exponential function** is the one implemented in ROS's Costmap_2d package:

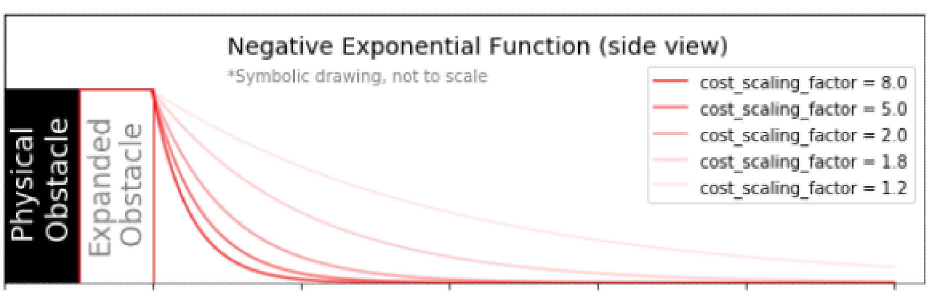
$e^{k(r-d)}$

Where:

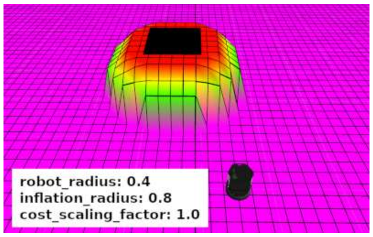
k: Curvature of slope (called Cost Scaling Factor)

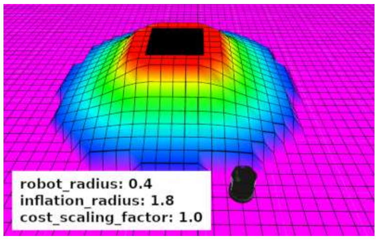
r: Radius of the expanded obstacle (Robot Radius)

d: Distance to the closes obstacle (Inflation Radius is used to maximum distance of influence of the buffer zone)

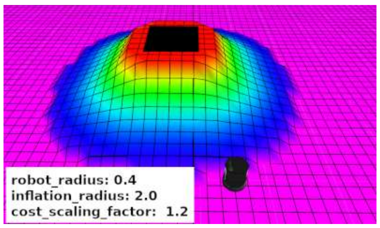


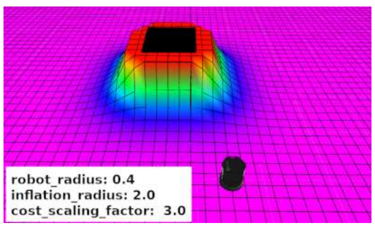
From the image below, we can observe that setting the Cost Scaling Factor higher will make the buffer zone decay
faster.





The Cost Scaling Factor determines the curvature of the slope in the buffer zone. And since it's multiplied by a negative value in the formula $(r - d)$, increasing it's value will decrease the resulting cost values.

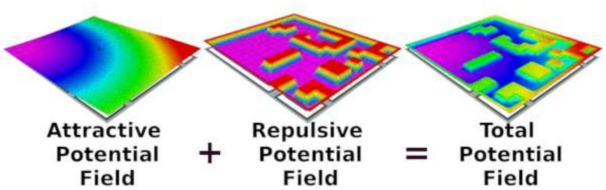




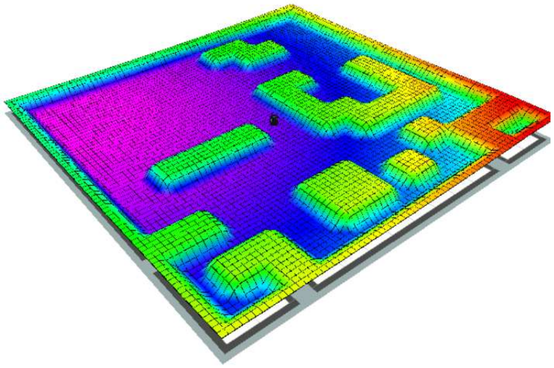
Total Potential Field

The total force acting on the robot is the sum of the goal and repelling forces from the obstacles. The image below shows how a total potential field map is obtained by adding the attractive and repulsive potential field.

$U_{total} = U_{att}(X) + U_{rep}(X)$



Example:

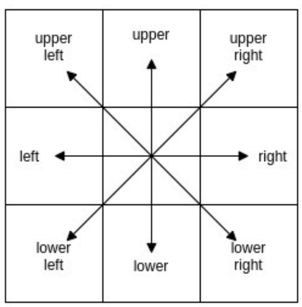


Gradient Descent Algorithm

The gradient or derivate of a function describes the slope of any given point. The Gradient Descent can be defined as:

$$\Delta F(x,y) = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}$$

However when you work with grip maps, you only have discrete values and not a continuous function. So the solution is to calculate the discrete value differences between adjacent grid cells, then we select the minimal value of the neighbors' gradient



Gradient Descent Application: