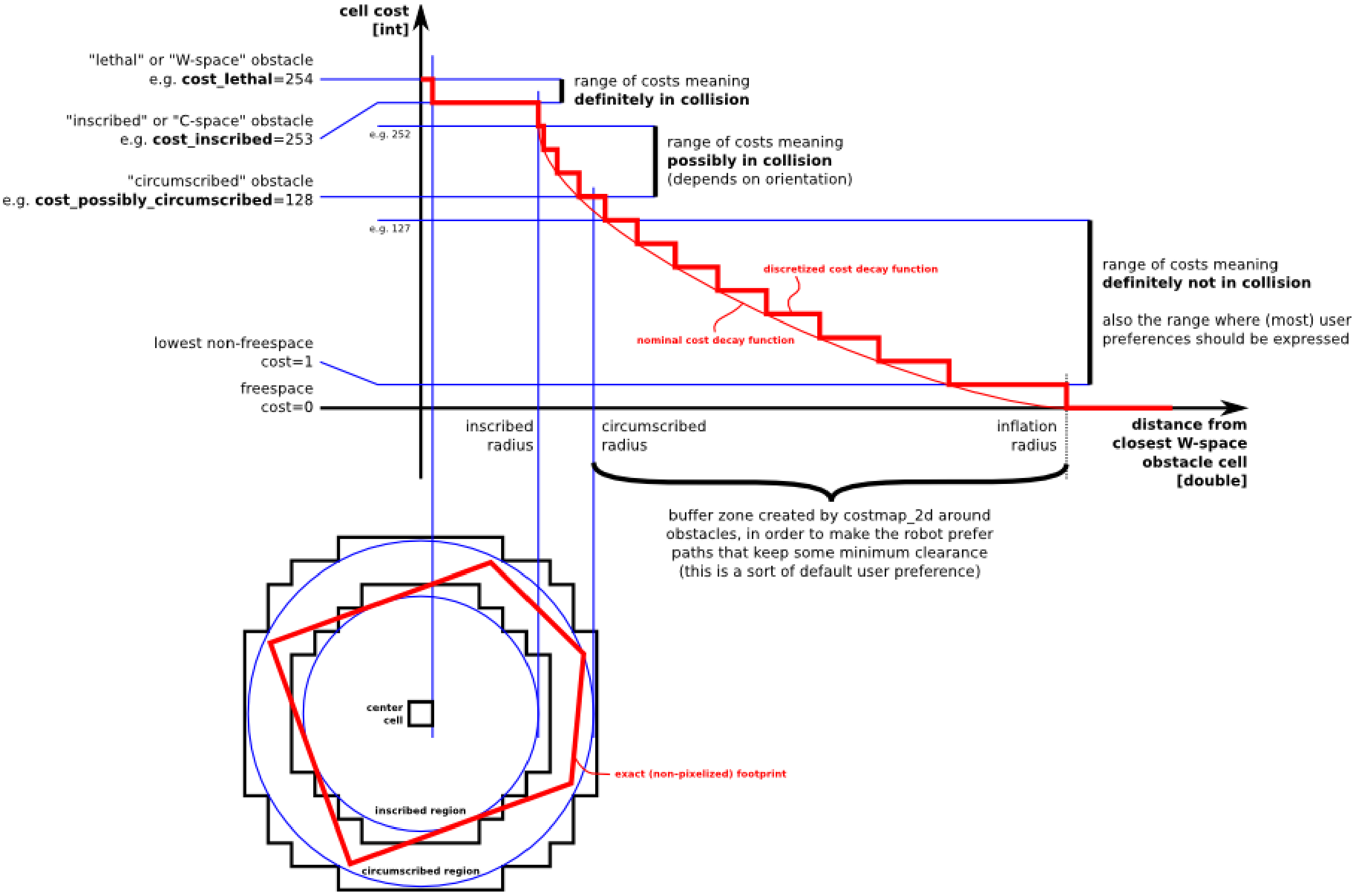
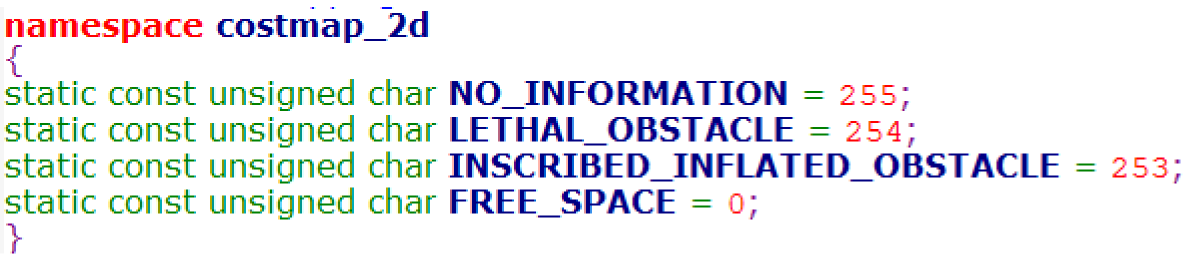
- 1.inflation
    - 参数 15.move_base介绍（3）介绍的`local_costmap`和`global_costmap`的参数 漏了其中的plugin参数

1.inflation

先看下官方的图


cell cost
[int]
"lethal" or "W-space" obstacle
e.g. cost_lethal=254
"inscribed" or "C-space" obstacle
e.g. cost_inscribed=253
"circumscribed" obstacle
e.g. cost_possibly_circumscribed=128
e.g. 252
e.g. 127
lowest non-freespace
cost=1
freespace
cost=0
range of costs meaning
definitely in collision
range of costs meaning
possibly in collision
(depends on orientation)
range of costs meaning
definitely not in collision
also the range where (most) user
preferences should be expressed
discretized cost decay function
nominal cost decay function
inscribed
radius
circumscribed
radius
inflation
radius
distance from
closest W-space
obstacle cell
[double]
buffer zone created by costmap_2d around
obstacles, in order to make the robot prefer
paths that keep some minimum clearance
(this is a sort of default user preference)
center
cell
exact (non-pixelized) footprint
inscribed region
circumscribed region

- 红色为机器人轮廓
- 内部蓝色圆即为内切圆
- 外部蓝色圆即为外切圆



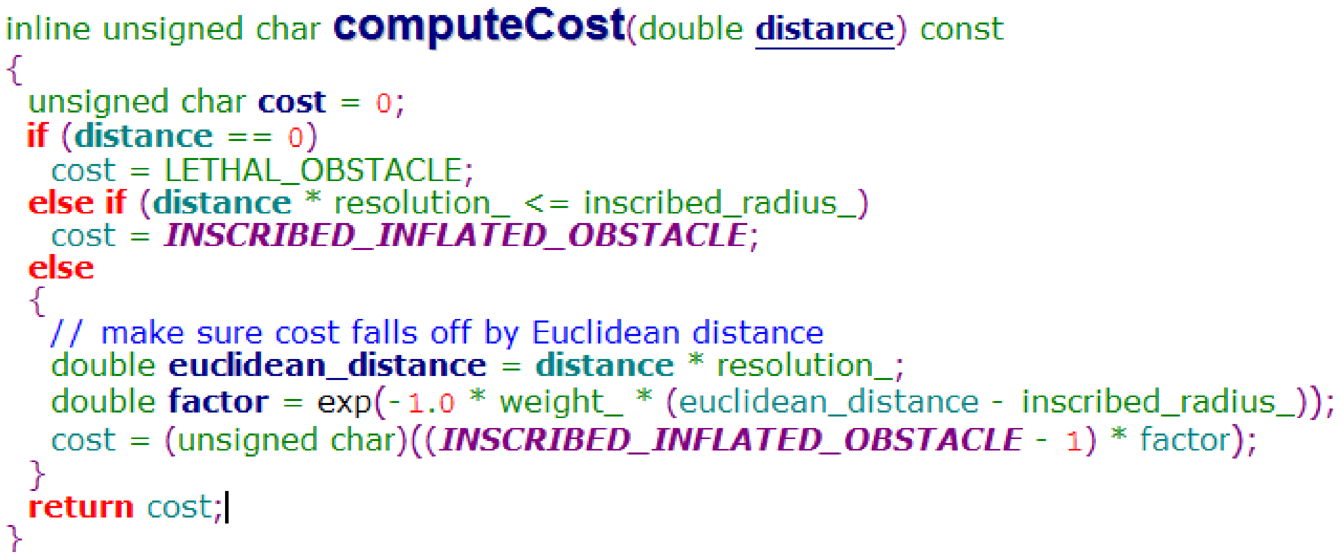
```
namespace costmap_2d
{
static const unsigned char NO_INFORMATION = 255;
static const unsigned char LETHAL_OBSTACLE = 254;
static const unsigned char INSCRIBED_INFLATED_OBSTACLE = 253;
static const unsigned char FREE_SPACE = 0;
}
```

每个网格的值从0~255

- `Lethal(254)`: 网格与机器人中心重合， 肯定导致冲突
- `Inscribed(253)`: 网格外切圆与机器人内切圆重合， 同样肯定导致冲突
- `Possibly circumscribed`: 网格外切圆与机器人外切圆外切， 可能导致冲突(机器人姿态决定)，具体由`inscribed_radius` `inflation_radius`和`cost scaling factor`相关
- `Freespace(0)`: 没有任何障碍，机器人可以占用该网格

- Unknown: 网格信息未知

上图坐标中的红色衰减曲线就标识网格离障碍的距离与cost_value的关系，计算方法具体如下

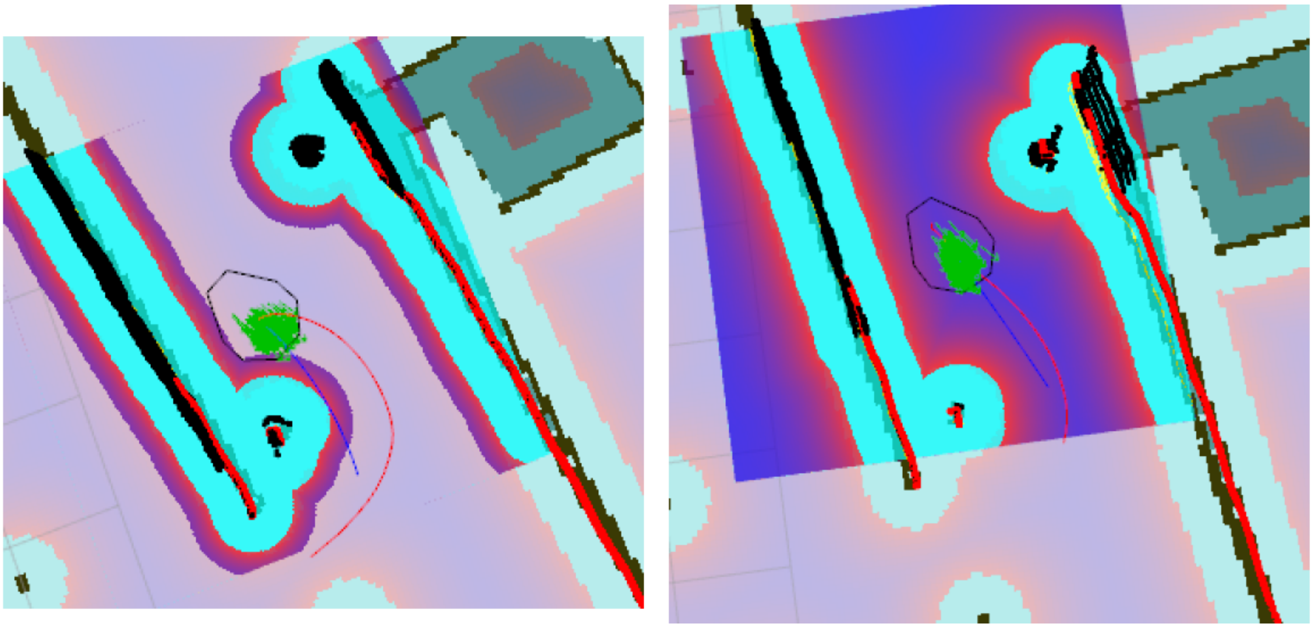


```
inline unsigned char computeCost(double distance) const
{
  unsigned char cost = 0;
  if (distance == 0)
    cost = LETHAL_OBSTACLE;
  else if (distance * resolution_ <= inscribed_radius_)
    cost = INSCRIBED_INFLATED_OBSTACLE;
  else
  {
    // make sure cost falls off by Euclidean distance
    double euclidean_distance = distance * resolution_;
    double factor = exp(-1.0 * weight_ * (euclidean_distance - inscribed_radius_));
    cost = (unsigned char)((INSCRIBED_INFLATED_OBSTACLE - 1) * factor);
  }
  return cost;
}
```

参数

```
inflation_layer:
  cost_scaling_factor:  2.5  # exponential rate at which the obstacle cost drops off (default: 10)
  inflation_radius:     1.2  # max. distance from an obstacle at which costs are incurred for planning paths.
```

- inflation_radius cost_value为0的网格离障碍的距离
- cost_scaling_factor 衰减因子，越大上面的曲线越陡



- 左图 inflation_radius = 0.55 cost_scaling_factor = 5
- 右图 inflation_radius = 1.75 cost_scaling_factor = 2.58