AMCL算法框架与实现

1. **贝叶斯滤波算法**
2. **参数说明**

**状态**，状态是会对未来产生影响所有的机器人因素和环境因素，对于机器人定位来说，状态变量为机器人位姿（pose），包含机器人相对全局坐标系的位置和方向，时间的状态表示为。

**环境测量数据**，环境测量数据提供了环境的暂态信息，测量数据的例子包括摄像机的图像、测距扫描等，在时间的测量数据表示为。

**控制数据，**控制数据携带环境中关于状态改变的信息，在移动机器人中，控制数据的典型例子就是机器人的速度。控制数据的另外一个例子就是里程计，里程计是测量机器人轮子运动的传感器，虽然里程计是测量传感器，但是仍把里程计视为控制数据，因为它传达了状态变化的信息。控制数据用表示。

**状态转移模型（状态转移概率）**



这里我们假设状态为完整性的，即如果知道了，那过去的测量和控制不会传递状态的信息。

**测量模型（测量概率）**



这里易知，时间的测量只由提供信息，过去的测量和控制都不提供信息，即只与有关。

**置信度**，置信度反映了机器人有关环境状态的内部信息，对于机器人定位，置信度表示为，表明了状态的可信度。

1. **贝叶斯滤波算法框架**

计算置信度的过程分为**预测**和**修正**两步

**预测的过程**是指基于以前状态的后验，综合时间之前的测量，预测时刻的状态，即计算

**修正的过程**称为测量更新的过程，即通过来计算

1. **贝叶斯滤波算法数学推导**

假设时刻的置信度正确初始化。

预测过程：



上式的第二步到第三步是利用贝叶斯公式，第三步到第四步是利用状态为完整性的假设

修正过程：



又因为，上式为



表示对的归一化。

由测量模型可知，最终，修正过程为



其中，表示归一化。

因此贝叶斯滤波算法的递推公式为

**预测过程**：

**更新过程：**

该算法的具体实现需要三个概率分布，初始置信度分布，测量概率，状态转移概率。

1. **粒子滤波算法**

粒子滤波是贝叶斯滤波的一种非参数实现，其主要思想是用一系列从后验得到的随机状态采样表示后验。在粒子滤波中，后验分布的样本叫作粒子，有



每个粒子是状态在时刻的一种具体实例，这里的代表粒子集的粒子数量。粒子滤波的直观感觉就是用一系列粒子来近似置信度，在理想情况下（即粒子数量足够大），粒子集中状态假设的可能性与贝叶斯滤波的后验成正比例：



1. **粒子滤波算法框架**

与贝叶斯滤波一样，粒子滤波算法有上一个时间步长的置信度递归构建置信度，因为置信度有粒子集表示，所以粒子滤波由粒子集递归构建粒子集，粒子滤波算法输入是粒子集和控制及测量。算法通过处理粒子集中的每个粒子构造出一个暂时的粒子集，随后，通过重采样将粒子集转换为粒子集。

详细说明有三步

**1、采样**：

从状态转移分布中采样，经过步迭代后得到的粒子集就是的滤波表示

**2、计算粒子权重**：

将表示为粒子的权值，加权后的粒子集近似表示贝叶斯滤波的后验。

**3、重采样**：

重采样在个粒子中的粒子集中抽取个粒子，重新组成粒子集，抽取每个粒子的概率根据其权值决定，权值大的粒子抽取的概率大，权值小的粒子抽取概率小。

1. **粒子滤波的数学推导**

粒子滤波的思想是用粒子群来近似置信度，求解置信度的期望就能够得到状态的估计。但置信度的分布并不知道，不能够直接去采样，需要引用重要性采样来解决这个问题。

我们无法从目标分布中采样，就从一个已知的建议分布中去采样，目标分布的期望问题就变成



其中

考虑到状态有马尔可夫性，有



上式中，第一行和第三行有贝叶斯公式得到，第二行和第四行有马尔可夫性得到

另一方面



上式中，第一步到第二步是因为从，中推测的可信度和从，中推测的可信度是一样的，第二步到第三步中，我们假设建议分布具有马尔可夫性，

所以



通常，为了方便处理，有



所以



以上就是序贯重要性采样（Sequential importance sampling, SIS）粒子滤波。

SIS粒子滤波有两步：

**采样**：

根据递推**计算各个粒子的权重**，最后将粒子的权重求和，作归一化处理。

然而在实际过程中，会出现粒子权重退化的问题，因此有了重采样。

**重采样的思路**是在保证粒子数目不变的情况下，舍弃权重小的粒子，复制权重大的粒子，而复制粒子的过程是根据粒子权重的比例去分配。

以一个简单的例子说明

假设有3个粒子，在第k时刻的时候，他们的权重分别是0.1, 0.1 ,0.8, 然后计算他们的概率累计和得到： [0.1, 0.2, 1]。接着，我们用服从[0,1]之间的均匀分布随机采样3个值，假设为0.15 , 0.38 和 0.54。也就是说，第二个粒子复制一次，第三个粒子复制两次。

通过重采样的过程我们可以知道，重采样后的粒子权重都为，所以粒子权重的递推公式可以简化为

1. **粒子滤波在MCL的运用**

在MCL中，粒子滤波的步骤如下

1、状态的初始化：在已知机器人位姿的情况下，位姿初始化为高斯分布；在未知机器人位姿的情况下，需要全局定位，位姿初始化为在地图上初始化为均匀分布。

2、采样粒子：

3、计算权重：，计算权重和，对每个粒子归一化处理

4、状态估计：，表示粒子归一化

5、重采样

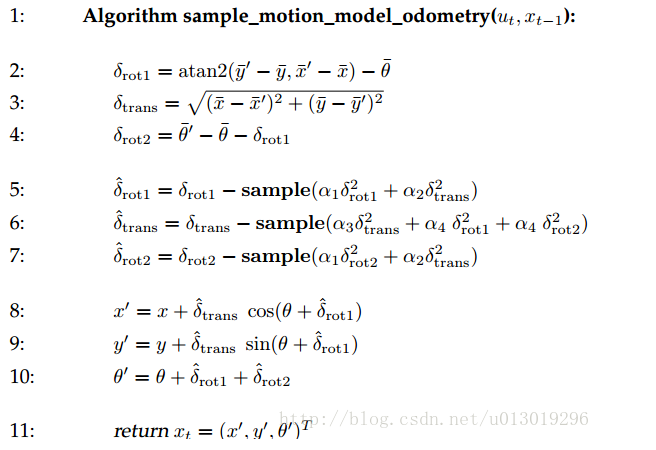
在概率机器人中，表示机器人的运动模型，表示激光雷达的测量模型

1. **运动模型**

运动模型由状态转换概率构成，通常有**速度模型**和**里程计模型**两种运动模型，速度模型假定运动数据为机器人电动机的速度指令；里程计模型假设机器人具有测距信息。

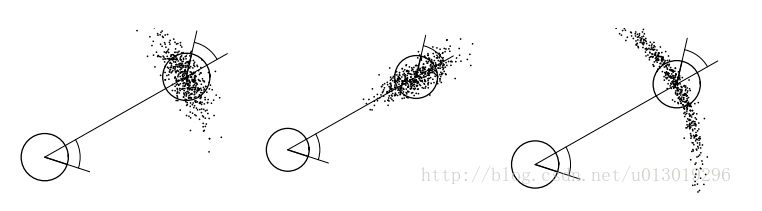
里程计模型往往比速度模型更精确，但是里程计信息仅在执行完运动控制后才获得，因此里程计模型通常用于估计，速度模型用于概率运动规划。

里程计模型的采样算法（《概率机器人》5.4）



这里的sample(x)表示均值为0，方差为x 的高斯分布采样。

根据给定的运动模型采样算法，，对于不同的误差参数也会有不同的概率分布



一个模型的采样参数是中等的，可以说是正常的，第二个和第三个扥别是比较大的平移和旋转的误差所造成的。

1. **测量模型**

对于AMCL，测量模型采用测距仪的波束模型（《概率机器人》6.3），原书的测量模型采用四类测量误差，而AMCL只采用两类测量误差：局部测量和随机测量。局部测量噪声是一个有一个窄的均值、标准偏差为的高斯建模，用表示。

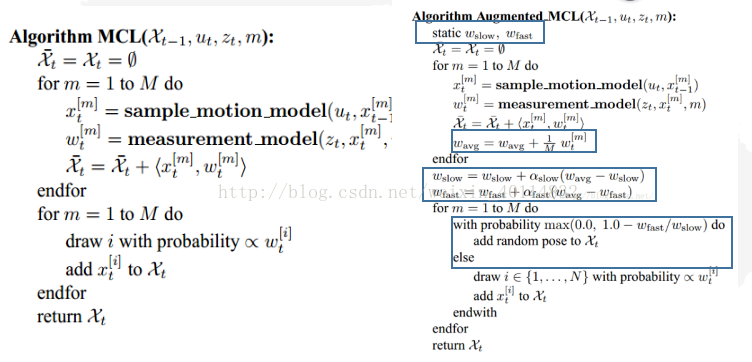


式中，距离表示测量的真实距离，表示具有均值为，标准偏差的正态分布，为归一化因子。

随机测量为的均匀分布



1. **AMCL算法**
2. **AMCL与MCL算法的比较**



如上图，AMCL在右；MCL在左。

MCL算法在前面已经提到，与MCL相比，AMCL能够解决失效恢复的问题，即当机器人全局定位失效时能够从失效中恢复出来。而解决这个问题的方法就是在增加粒子，在增加粒子的过程又会遇到两个问题：第一，应该增加多少粒子；第二，从哪些分布中产生粒子。

AMCL通过对平均测量概率的监控来实现解决失效定位的问题，增加两个参数和。



其中，

表示用来跟踪权重短期均值，表示用来跟踪权重长期均值。当短期均值大于长期均值，重采样以原来方式进行；当短期均值小于长期均值，重采样过程中按照两者之间的比例增加随机采样。

1. **AMCL算法的实现**

AMCL算法的实现来源于<https://github.com/ros-planning/navigation/tree/kinetic-devel/amcl>

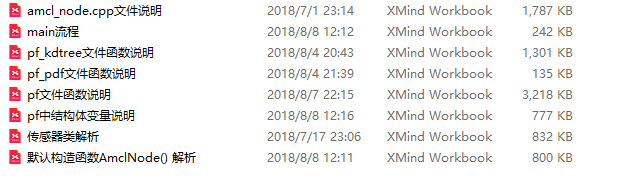
在该源码中，AMCL通过调用AmclNode类中的laserReceived函数实现，AmclNode()主要流程如下



在laserReceived函数中，实现了AMCL算法的过程，具体步骤如下：



AMCL中思维导图笔记说明



amcl\_node.cpp文件说明：记录amcl\_node.cpp的实现，主要是AmclNode类中的成员变量以及默认构造函数 AmclNode()。

main流程：记录AMCL在ROS中的启动流程，函数入口在默认构造函数 AmclNode()中。（重要！！）

pf\_kdtree文件函数说明：记录pf\_kdtree.c的实现，主要是kdtree的初始化、结点插入、kdtree 中的叶子结点的聚类

pf\_pdf文件函数说明：记录pf\_pdf.c 的实现，主要是从分布函数中采样

pf文件函数说明：记录pf.c的实现，主要是粒子滤波器的初始化、运动模型、激光雷达的测量模型、重采样、计算粒子簇的簇统计信息（重要！！）

pf中结构体变量说明：记录各个结构体的联系（重要！！）

传感器类解析：记录里程计和激光雷达两种传感器的模型初始化和更新。

1. **AMCL算法的结果**

|  |  |  |
| --- | --- | --- |
| 真实坐标（cm，cm） | 定位坐标（cm，cm） | 误差（cm，cm） |
| （73，00） | （26.5，-0.96） | （46.5，0.96） |
| （30，25） | （18.6，-35.5） | （11.4，60.5） |
| （87，22） | （93.3，87.5） | （-6.3，-65.5） |

