AMCL 算法框架与实现

1. 贝叶斯滤波算法

1.1. 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 t 的状态表示为 x_t 。

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 t 的测量数据表示为 z_t 。

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 u_t 表示。

状态转移模型（状态转移概率）

$$p\left(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}\right) = p\left(x_t \mid x_{t-1}, u_t\right)$$

这里我们假设状态 x_t 为完整性的，即如果知道了 x_{t-1}，那过去的测量 $z_{1:t-1}$ 和控制 $u_{1:t-1}$ 不会传递状态 x_t 的信息。

测量模型（测量概率）

$$p\left(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}\right) = p\left(z_t \mid x_t\right)$$

这里易知，时间 t 的测量 z_t 只由 x_t 提供信息，过去的测量 $z_{1:t-1}$ 和控制 $u_{1:t}$ 都不提供信息，即 z_t 只与 x_t 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 $\mathrm{bel}\left(x_t\right)$ 表示为 $\mathrm{bel}\left(x_t\right) = p\left(x_t \mid z_{1:t}, u_{1:t}\right)$，表明了状态 x_t 的可信度。

1.2. 贝叶斯滤波算法框架

计算置信度 $\mathrm{bel}\left(x_t\right)$ 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 t 的后验，综合时间 t 之前的测量，预测时刻 t 的状态 x_t，即计算 $\overline{\mathrm{bel}}\left(x_t\right) = p\left(x_t \mid z_{1:t-1}, u_{1:t}\right)$

修正的过程称为测量更新的过程，即通过 $\overline{\mathrm{bel}}\left(x_t\right) = p\left(x_t \mid z_{1:t-1}, u_{1:t}\right)$ 来计算 $\mathrm{bel}\left(x_t\right) = p\left(x_t \mid z_{1:t}, u_{1:t}\right)$

1.3. 贝叶斯滤波算法数学推导

假设时刻 t_0 的置信度 $\mathrm{bel}\left(x_0\right)$ 正确初始化。

预测过程：

$$\begin{aligned}\overline{\mathrm{bel}}(x_t) &= p(x_t|z_{1:t-1},u_{1:t})\\ &=\int p(x_t,x_{t-1}|z_{1:t-1},u_{1:t})d(x_{t-1})\\ &=\int p(x_t|x_{t-1},z_{1:t-1},u_{1:t})p(x_{t-1}|z_{1:t-1},u_{1:t})d(x_{t-1})\\ &=\int p(x_t|x_{t-1},u_t)p(x_{t-1}|z_{1:t-1},u_{1:t-1})d(x_{t-1})\\ &=\int p(x_t|x_{t-1},u_t)\mathrm{bel}(x_{t-1})d(x_{t-1})\end{aligned}$$

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 x_t 为完整性的假设

修正过程：

$$\begin{aligned}\mathrm{bel}(x_t) &= p(x_t|z_{1:t},u_{1:t})\\ &=\frac{p(x_t,z_t,z_{1:t-1},u_{1:t})}{p(z_t,z_{1:t-1},u_{1:t})}=\frac{p(z_t|x_t,z_{1:t-1},u_{1:t})\,p(x_t,z_{1:t-1},u_{1:t})}{p(z_t|z_{1:t-1},u_{1:t})\,p(z_{1:t-1},u_{1:t})}\\ &=\frac{p(z_t|x_t,z_{1:t-1},u_{1:t})\,p(x_t|z_{1:t-1},u_{1:t})}{p(z_t|z_{1:t-1},u_{1:t})}\end{aligned}$$

又因为 $p(z_t|z_{1:t-1},u_{1:t})=\int p(z_t|x_t,z_{1:t-1},u_{1:t})\,p(x_t|z_{1:t-1},u_{1:t})d(x_t)$，上式为

$$\mathrm{bel}(x_t)=\eta\, p(z_t|x_t,z_{1:t-1},u_{1:t})\,p(x_t|z_{1:t-1},u_{1:t})$$

η 表示对 $p(z_t|x_t,z_{1:t-1},u_{1:t})\,p(x_t|z_{1:t-1},u_{1:t})$ 的归一化。

由测量模型可知 $p(z_t|x_t,z_{1:t-1},u_{1:t})=p(z_t|x_t)$，最终，修正过程为

$$\mathrm{bel}(x_t)=\eta\, p(z_t|x_t)\overline{\mathrm{bel}}(x_t)$$

其中 $\eta=\int p(z_t|x_t,z_{1:t-1},u_{1:t})\,p(x_t|z_{1:t-1},u_{1:t})d(x_t)$，表示归一化。

因此贝叶斯滤波算法的递推公式为

预测过程： $\overline{\mathrm{bel}}(x_t)=p(x_t|z_{1:t-1},u_{1:t})=\int p(x_t|x_{t-1},u_t)\mathrm{bel}(x_{t-1})d(x_{t-1})$

更新过程： $\mathrm{bel}(x_t)=p(x_t|z_{1:t},u_{1:t})=\eta\, p(z_t|x_t)\overline{\mathrm{bel}}(x_t)$

该算法的具体实现需要三个概率分布，初始置信度分布 $\mathrm{bel}(x_0)=p(x_0)$，测量概率 $p(z_t|x_t)$，状态转移概率 $p(x_t|x_{t-1},u_t)$。

2. 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 $\mathrm{bel}(x_t)$。在粒子滤波中，后验分布的样本叫作粒子，有

$$\chi_t := x_t^{[1]},x_t^{[2]},\cdots,x_t^{[M]}$$

每个粒子 $x_t^{[m]}(1\le m\le M)$ 是状态在时刻 t 的一种具体实例，这里的 M 代表粒子集 χ_t 的粒子数量。粒子滤波的直观感觉就是用一系列粒子 χ_t 来近似置信度 $\mathrm{bel}(x_t)$，在理想情况下（即粒子数量足够大），粒子集 χ_t 中状态假设 x_t 的可能性与贝叶斯滤波的后验 $\mathrm{bel}(x_t)$ 成正比例：

$$x_t^{[m]} \sim \mathrm{bel}\left(x_t\right)=p\left(x_t \mid z_{1:t}, u_{1:t}\right)$$

2.1. 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 $\mathrm{bel}\left(x_{t-1}\right)$ 递归构建置信度 $\mathrm{bel}\left(x_t\right)$，因为置信度有粒子集表示，所以粒子滤波由粒子集 χ_{t-1} 递归构建粒子集 χ_t，粒子滤波算法输入是粒子集 χ_{t-1} 和控制 u_t 及测量 z_t。算法通过处理粒子集 χ_{t-1} 中的每个粒子 $x_t^{[m]}$ 构造出一个暂时的粒子集 $\bar{\chi}_t$，随后，通过重采样将粒子集 $\bar{\chi}_t$ 转换为粒子集 χ_t。

详细说明有三步

1、采样： $x_t^{[m]} \sim p\left(x_t \middle| x_{t-1}^{[m]}, u_t\right)$

从状态转移分布 $p\left(x_t \mid x_{t-1}, u_t\right)$ 中采样，经过步迭代后得到的粒子集就是 $\overline{\mathrm{bel}}\left(x_t\right)$ 的滤波表示

2、计算粒子权重： $w_t^{[m]} \sim p\left(z_t \middle| x_t^{[m]}\right)$

将 $w_t^{[m]}$ 表示为粒子 $x_t^{[m]}$ 的权值，加权后的粒子集 $\bar{\chi}_t$ 近似表示贝叶斯滤波的后验 $\mathrm{bel}\left(x_t\right)$。

3、重采样： $\chi_t \sim \bar{\chi}_t$

重采样在 M 个粒子中的粒子集 $\bar{\chi}_t$ 中抽取 M 个粒子，重新组成粒子集 χ_t，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。

2.2. 粒子滤波的数学推导

粒子滤波的思想是用粒子群 χ_t 来近似置信度 $\mathrm{bel}\left(x_t\right)=p\left(x_t \mid z_{1:t}, u_{1:t}\right)$，求解置信度的期望就能够得到状态 x_t 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 $p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)$ 中采样，就从一个已知的建议分布 $q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)$ 中去采样，目标分布的期望问题就变成

$$\begin{aligned}
E\left[f\left(x_t\right)\right] &= \int f\left(x_t\right) p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right) d\left(x_{0:t}\right) \\
&= \int f\left(x_t\right) \frac{p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)}{q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)} q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right) d\left(x_{0:t}\right) \\
&= \int f\left(x_t\right) \frac{p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)}{q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)} q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right) d\left(x_{0:t}\right) \\
&= \int f\left(x_t\right) w_t q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right) d\left(x_{0:t}\right)
\end{aligned}$$

其中 $w_t = \dfrac{p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)}{q\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)}$

考虑到状态 x_t 有马尔可夫性，有

$$
\begin{aligned}
p\left(x_{0:t} \middle| z_{1:t}, u_{1:t}\right) &= \eta p\left(z_t \middle| x_{0:t}, z_{1:t-1}, u_{1:t}\right) p\left(x_{0:t} \middle| z_{1:t-1}, u_{1:t}\right) \\
&= \eta p\left(z_t \middle| x_t\right) p\left(x_{0:t} \middle| z_{1:t-1}, u_{1:t}\right) \\
&= \eta p\left(z_t \middle| x_t\right) p\left(x_t \middle| x_{0:t-1}, z_{1:t-1}, u_{1:t}\right) p\left(x_{0:t-1} \middle| z_{1:t-1}, u_{1:t}\right) \\
&= \eta p\left(z_t \middle| x_t\right) p\left(x_t \middle| x_{t-1}, u_{1:t}\right) p\left(x_{0:t-1} \middle| z_{1:t-1}, u_{1:t}\right)
\end{aligned}
$$

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

$$
\begin{aligned}
q\left(x_{0:t} \middle| z_{1:t}, u_{1:t}\right) &= q\left(x_t \middle| x_{0:t-1}, z_{1:t}, u_{1:t}\right) q\left(x_{0:t-1} \middle| z_{1:t}, u_{1:t}\right) \\
&= q\left(x_t \middle| x_{0:t-1}, z_{1:t}, u_{1:t}\right) q\left(x_{0:t-1} \middle| z_{1:t-1}, u_{1:t-1}\right) \\
&= q\left(x_t \middle| x_{t-1}, z_{1:t}, u_{1:t}\right) q\left(x_{0:t-1} \middle| z_{1:t-1}, u_{1:t-1}\right)
\end{aligned}
$$

上式中，第一步到第二步是因为从 $z_{1:t}$， $u_{1:t}$ 中推测 $x_{0:t-1}$ 的可信度和从 $z_{1:t\text{-}1}$， $u_{1:t\text{-}1}$ 中推测 $x_{0:t-1}$ 的可信度是一样的，第二步到第三步中，我们假设建议分布 $q\left(x_{0:t} \middle| z_{1:t}, u_{1:t}\right)$ 具有马尔可夫性，

所以

$$
\begin{aligned}
w_t &= \frac{p\left(x_{0:t} \middle| z_{1:t}, u_{1:t}\right)}{q\left(x_{0:t} \middle| z_{1:t}, u_{1:t}\right)} \\
&\propto \frac{p\left(z_t \middle| x_t\right) p\left(x_t \middle| x_{t-1}, u_{1:t}\right) p\left(x_{0:t-1} \middle| z_{1:t-1}, u_{1:t}\right)}{q\left(x_t \middle| x_{t-1}, z_{1:t}, u_{1:t}\right) q\left(x_{0:t-1} \middle| z_{1:t-1}, u_{1:t-1}\right)} \\
&= w_{t\text{-}1} \frac{p\left(z_t \middle| x_t\right) p\left(x_t \middle| x_{t-1}, u_{1:t}\right)}{q\left(x_t \middle| x_{t-1}, z_{1:t}, u_{1:t}\right)}
\end{aligned}
$$

通常，为了方便处理，有

$$
q\left(x_t \middle| x_{t-1}, z_{1:t}, u_{1:t}\right) = p\left(x_t \middle| x_{t-1}, z_{1:t}, u_{1:t}\right) = p\left(x_t \middle| x_{t-1}, u_t\right)
$$

所以

$$
w_t \propto w_{t\text{-}1} p\left(z_t \middle| x_t\right)
$$

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： $x_t^{[m]} \sim p\left(x_t \middle| x_{t\text{-}1}^{[m]}, u_t\right)$

根据 $w_t^{[m]} \propto w_{t\text{-}1}^{[m]} p\left(z_t \middle| x_t^{[m]}\right)$ 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$1/M$，所以粒子权重的递推公式可以简化为$w_t^{[m]} \propto p\left(z_t \middle| x_t^{[m]}\right)$

2.3. 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：$x_t^{[m]} \sim p\left(x_t \middle| x_{t\text{-}1}^{[m]}, u_t\right)$

3、计算权重：$w_t^{[m]}=p\left(z_t \middle| x_t^{[m]}\right)$，计算权重和，对每个粒子归一化处理

4、状态估计：$\bar{x}_t=\sum_{m=1}^{M}\tilde{w}_t^{[m]}x_t^{[m]}$ ，$\tilde{w}_t^{[m]}$表示粒子归一化

5、重采样

在概率机器人中，$p\left(x_t \middle| x_{t-1}, u_t\right)$表示机器人的运动模型，$p\left(z_t \middle| x_t\right)$表示激光雷达的测量模型

3. 运动模型

运动模型由状态转换概率$p\left(x_t \middle| x_{t-1}, u_t\right)$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）

Algorithm sample_motion_model_odometry(u_t, x_{t-1}):

$\delta_{\text{rot1}} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$

$\delta_{\text{trans}} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$

$\delta_{\text{rot2}} = \bar{\theta}' - \bar{\theta} - \delta_{\text{rot1}}$

$\hat{\delta}_{\text{rot1}} = \delta_{\text{rot1}} - \textbf{sample}(\alpha_1 \delta_{\text{rot1}}^2 + \alpha_2 \delta_{\text{trans}}^2)$

$\hat{\delta}_{\text{trans}} = \delta_{\text{trans}} - \textbf{sample}(\alpha_3 \delta_{\text{trans}}^2 + \alpha_4\ \delta_{\text{rot1}}^2 + \alpha_4\ \delta_{\text{rot2}}^2)$

$\hat{\delta}_{\text{rot2}} = \delta_{\text{rot2}} - \textbf{sample}(\alpha_1 \delta_{\text{rot2}}^2 + \alpha_2 \delta_{\text{trans}}^2)$

$x' = x + \hat{\delta}_{\text{trans}}\ \cos(\theta + \hat{\delta}_{\text{rot1}})$

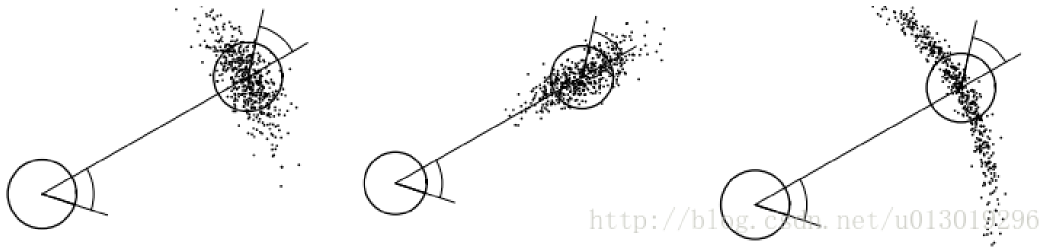
$y' = y + \hat{\delta}_{\text{trans}}\ \sin(\theta + \hat{\delta}_{\text{rot1}})$

$\theta' = \theta + \hat{\delta}_{\text{rot1}} + \hat{\delta}_{\text{rot2}}$

return $x_t = (x', y', \theta')^T$

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，第二个和第三个托别是比较大的平移和旋转的误差所造成的。

4. 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 $\boldsymbol{z}_t^{k^*}$ 、标准偏差 σ_{hit} 为的高斯建模，用 $\boldsymbol{p}_{\text{hit}}$ 表示。

$$\boldsymbol{p}_{\text{hit}}\left(\boldsymbol{z}_t^k \middle| \boldsymbol{x}_t, \boldsymbol{m}\right) = \begin{cases} \eta \mathbb{N}\left(\boldsymbol{z}_t^k; \boldsymbol{z}_t^{k^*}, \sigma_{\text{hit}}^2\right) & 0 \le \boldsymbol{z}_t^k \le \boldsymbol{z}_{\max} \\ 0 & \boldsymbol{other} \end{cases}$$

式中，距离 $\boldsymbol{z}_t^{k^*}$ 表示测量 $\boldsymbol{z}_t^k$ 的真实距离，$\mathbb{N}\left(\boldsymbol{z}_t^k; \boldsymbol{z}_t^{k^*}, \sigma_{\text{hit}}^2\right)$ 表示具有均值为 $\boldsymbol{z}_t^{k^*}$ ，标准偏差 σ_{hit} 的正态分布，$\eta = \left(\int_0^{\boldsymbol{z}_{\max}} \mathbb{N}\left(\boldsymbol{z}_t^k; \boldsymbol{z}_t^{k^*}, \sigma_{\text{hit}}^2\right) \boldsymbol{d}\boldsymbol{z}_t^k\right)^{-1}$ 为归一化因子。

随机测量为 $\left[0;\ \boldsymbol{z}_{\max}\right]$ 的均匀分布

$$\boldsymbol{p}_{\text{rand}}\left(\boldsymbol{z}_t^k \middle| \boldsymbol{x}_t, \boldsymbol{m}\right) = \begin{cases} \dfrac{1}{\boldsymbol{z}_{\max}} & 0 \le \boldsymbol{z}_t^k \le \boldsymbol{z}_{\max} \\ 0 & \boldsymbol{other} \end{cases}$$

5. AMCL 算法

5.1. AMCL 与 MCL 算法的比较

Algorithm MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ to M do
$x_t^{[m]} =$ **sample_motion_model**$(u_t, x_{t-1}^{[m]}$
$w_t^{[m]} =$ **measurement_model**$(z_t, x_t^{[m]},$
$\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
endfor
for $m = 1$ to M do
draw i with probability $\propto w_t^{[i]}$
add $x_t^{[i]}$ to $\mathcal{X}_t$
endfor
return $\mathcal{X}_t$

Algorithm Augmented MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):
static $w_{\text{slow}}, w_{\text{fast}}$
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ to M do
$x_t^{[m]} =$ **sample_motion_model**$(u_t, x_{t-1}^{[m]})$
$w_t^{[m]} =$ **measurement_model**$(z_t, x_t^{[m]}, m)$
$\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
$w_{\text{avg}} = w_{\text{avg}} + \frac{1}{M} w_t^{[m]}$
endfor
$w_{\text{slow}} = w_{\text{slow}} + \alpha_{\text{slow}}(w_{\text{avg}} - w_{\text{slow}})$
$w_{\text{fast}} = w_{\text{fast}} + \alpha_{\text{fast}}(w_{\text{avg}} - w_{\text{fast}})$
for $m = 1$ to M do
with probability $\max(0.0,\ 1.0 - w_{\text{fast}}/w_{\text{slow}})$ do
add random pose to $\mathcal{X}_t$
else
draw $i \in \{1, \ldots, N\}$ with probability $\propto w_t^{[i]}$
add $x_t^{[i]}$ to $\mathcal{X}_t$
endwith
endfor
return $\mathcal{X}_t$

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 W_{fast} 和 W_{slow}。

$$w_{fast} = w_{fast} + \alpha_{fast}\left(w_{avg} - w_{fast}\right)$$

$$w_{slow} = w_{slow} + \alpha_{slow}\left(w_{avg} - w_{slow}\right)$$

其中 $0 \le \alpha_{fast} \ll \alpha_{slow}$， $w_{avg} = \frac{1}{M}\sum_{m=1}^{M} w_t^{[m]}$

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

5.2. 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	2018/7/1 23:14	XMind Workbook	1,787 KB
main流程	2018/8/8 12:12	XMind Workbook	242 KB
pf_kdtree文件函数说明	2018/8/4 20:43	XMind Workbook	1,301 KB
pf_pdf文件函数说明	2018/8/4 21:39	XMind Workbook	135 KB
pf文件函数说明	2018/8/7 22:15	XMind Workbook	3,218 KB
pf中结构体变量说明	2018/8/8 12:16	XMind Workbook	777 KB
传感器类解析	2018/7/17 23:06	XMind Workbook	832 KB
默认构造函数AmclNode() 解析	2018/8/8 12:11	XMind Workbook	800 KB

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。

5.3. 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）