

基于滤波的方法

KF: kalman算法, 适合用于线性的高斯分布的状态环境中

EKF: 拓展的kalman滤波, 解决在非高斯和非线性环境下的机器人定位算法

问题: 在现实应用中计算量、内存消耗上不是很高效。

这就引出了MCL算法

蒙特卡洛定位 (MCL), 也称为**粒子滤波器定位**, 是一种机器人使用**粒子滤波器**进行**定位**的算法。给定环境图, 该算法会估计机器人在移动并感测环境时的**位置和方向**。该算法使用**粒子过滤器**表示可能状态的**分布**, 每个粒子表示可能的状态, 即机器人所在位置的假设。该算法通常从**配置空间**上的均匀均匀的粒子随机分布开始, 这意味着机器人没有关于其位置的信息, 并假定它同样有可能位于空间的任何一点。每当机器人移动时, 它都会移动粒子以预测运动后的新状态。每当机器人感测到某物时, 都会基于**递归贝叶斯估计** (即实际感测数据与预测状态的相关程度) 对粒子重新采样。最终, 粒子应会聚到机器人的实际位置。

粒子滤波很粗浅的说就是一开始在地图空间很均匀的撒一把粒子, 然后通过获取机器人的motion来移动粒子, 比如机器人向前移动了一米, 所有的粒子也就向前移动一米, 不管现在这个粒子的位置对不对。使用每个粒子所处位置模拟一个传感器信息跟观察到的传感器信息 (一般是激光) 作对比, 从而赋给每个粒子一个概率。之后根据生成的概率来重新生成粒子, 概率越高的生成的概率越大。这样的迭代之后, 所有的粒子会慢慢地收敛到一起, 机器人的确切位置也就被推算出来了。

MCL的计算步骤:

1. **随机产生一堆粒子**: 粒子可以有位置、方向和/或任何其他需要估计的状态变量。每一个都有一个权值(概率), 表明它与系统的实际状态匹配的可能性有多大。用相同的权重初始化每个变量。
2. **预测粒子的下一个状态**: 根据真实系统行为的预测来移动粒子。
3. **更新**: 根据测量结果更新粒子的权重。与测量值密切匹配的粒子的权重要高于与测量值不太匹配的粒子。
4. **重新采样**: 抛弃高度不可能的粒子, 代之以更可能的粒子的副本。
5. **计算估计值**: 可选地, 计算粒子集的加权平均值和协方差来得到状态估计。

粒子滤波的基本步骤为上面所述的5步, 其本质是使用一组有限的**加权随机样本 (粒子)** 来近似表征任意**状态的后验概率密度**。粒子滤波的优势在于对复杂问题的求解上, 比如高度的非线性、非高斯动态系统的状态递推估计或概率推理问题。

算法伪码:

时间 t , 输入: 先前的置信值 $X_{t-1} = \{x_{t-1}^{[1]}, x_{t-1}^{[2]}, \dots, x_{t-1}^{[M]}\}$, u_t 一个驱动命令, z_t 是从传感器接收的数据, 算法输出新的置信值 X_t 。

Algorithm MCL(X_{t-1}, u_t, z_t):

$\bar{X}_t = X_t = \emptyset$

for $m = 1$ *to* M :

$x_t^{[m]} = \text{motion_update}(u_t, x_{t-1}^{[m]})$

$w_t^{[m]} = \text{sensor_update}(z_t, x_t^{[m]})$

$\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$

endfor

for $m = 1$ *to* M :

draw $x_t^{[m]}$ *from* \bar{X}_t *with probability* $\propto w_t^{[m]}$

$X_t = X_t + x_t^{[m]}$

endfor

return X_t

地磁定位：

原理：。。。

数据采集：

1. 单点采集
2. 连续采集，序列定位

室内地磁定位的不足

1、地磁数据采集工作

地磁定位技术的使用要始于对现场地磁数据的采集，而且每当室内环境有较大的变动，比如重新装修的情况时，需要重新对现场地磁数据进行采集更新，这相当于给用户带来了较多的工作量。

2、信号干扰

地磁信号本身是容易受到金属物的干扰，室内环境又很难保持一成不变，假如用户周边突然出现驶过一辆汽车，现场磁场受到干扰，对其位置的获取势必会有较多影响，因此地磁定位的稳定性值得商榷。

3、初始位置的获取

地磁定位是相对位置，所以单纯依靠地磁来定位是没法立刻获取初始位置的，很多方案都要求用户在使用定位前在室内行走5~8米，以帮助获取初始位置，也有很多方案采取其他定位技术（比如PDR/iBeacon/Wifi）来辅助获取初始位置，这也是室内地磁定位技术最难解决的问题。