

辅助粒子滤波算法改进的 UFastSLAM 算法

张国良, 汤文俊, 敬 斌, 程展欣

(第二炮兵工程大学 控制工程系, 710025 西安)

摘 要: UFastSLAM 算法的重采样过程会带来“粒子耗尽”问题, 为提高算法精度和性能, 将辅助粒子滤波算法思想引入 UFastSLAM 算法, 提出了解决 SLAM 问题的方法——辅助 UFastSLAM 算法. 针对 UFastSLAM 算法的特点, 首先, 在其建议分布函数的求取过程中引入最近的观测信息, 增强采样粒子权值的稳定性; 其次, 在其重采样过程中引入 1 次加权和 2 次加权, 增大观测似然度大的粒子比例以缓解粒子退化问题. 实验结果证明, 辅助 UFastSLAM 算法在估计精度、一致性等方面都具有很好的性能.

关键词: 辅助粒子滤波; UFastSLAM; 观测似然度; 1 次加权; 2 次加权; 粒子退化

中图分类号: TP242

文献标志码: A

文章编号: 0367-6234(2012)11-0123-06

Improved UFastSLAM algorithm based on auxiliary particle filter

ZHANG Guo-liang, TANG Wen-jun, JING Bin, CHENG Zhan-xin

(Dept. of Control Engineering, The Second Artillery Engineering University, 710025 Xi'an, China)

Abstract: The UFastSLAM algorithm's resampling process will bring "sample reproverishment" problem. To improve the algorithm performance and increase the estimate accuracy, and based on the thought of auxiliary particle filtering algorithm, a solution named "Auxiliary UFastSLAM" is presented for SLAM problem. In view of the specialty of UFastSLAM algorithm, the recent observation information in its suggestion distribution function's seeking process to enhance stability of particle sampling, and one time weighting and second time weighting in its resampling process to alleviate particle degeneration are introduced. The experimental results indicate that the Auxiliary UFastSLAM algorithm does well on both estimate accuracy and consistency.

Key words: auxiliary particle filtering; UFastSLAM; observation likelihood; one time weighting; second time weighting; particle degeneration

移动机器人的同时定位与地图创建 (SLAM), 是指移动机器人在未知环境中从一个未知位置开始移动, 在移动的过程中利用位姿估计和传感器观测进行自身定位, 同时创建环境地图^[1-2]. 由于 SLAM 问题是实现机器人完全自主的关键, 因此很快成为机器人研究领域的热点问题, 并被誉为机器人研究领域的“圣杯”^[3-4].

FastSLAM 算法是基于粒子滤波 (PF) 算法的 SLAM 算法, 相对于 EKF-SLAM 算法具有很大的优势. 首先, 它能直接应用于非线性非高斯系统;

其次, 它引入了 Rao-Blackwellized 分解^[5], 将 SLAM 问题分解成对机器人运动路径的递归估计和对环境特征位置的 EKF 估计这 2 个独立的问题^[6], 有效降低了计算复杂度. 然而, FastSLAM 算法仍存在很多问题. 首先, 其建议分布函数的求取和环境特征位置的估计依然是基于扩展卡尔曼滤波方法的; 其次, 粒子滤波会造成粒子退化问题. Chanki Kim 等^[6]对 FastSLAM 算法进行改进, 提出了 UFastSLAM 算法^[7]. 该算法运用 UPF 算法^[8]进行机器人位姿的估计, 用 UKF 算法代替 EKF 算法进行环境特征位置的估计, 这样不仅消除了 EKF 算法的线性化误差, 提高了 FastSLAM 算法的精度和一致性, 也减少了必要的采样粒子数. 然而, UFastSLAM 算法依然存在缺点. 首先, 其并没

收稿日期: 2012-02-13.

基金项目: 陕西省自然科学基金资助项目 (2012K06-45); 第二炮兵工程学院创新基金资助项目.

作者简介: 张国良 (1970—), 男, 教授, 博士生导师.

通信作者: 汤文俊, 13468972665@163.com.

有对粒子退化问题进行改善;其次,其粒子采样中不包含最近的观测信息,可能会导致采样粒子权值的不稳定.

针对上述问题,本文运用辅助粒子滤波算法思想对 UFastSLAM 算法进行改进.首先,通过利用系统状态均值等特征来近似计算观测似然度函数,从而将最近的观测信息引入建议分布函数的计算中,增强采样粒子权值的稳定性;其次,为了缓解粒子退化,在重采样过程中引入 1 次加权值和 2 次加权值的计算,增大观测似然度大的粒子的比例,相当于在重采样过程中引入了最近的观测信息,从而使得到的粒子接近于真实系统状态的可能性更高. UFastSLAM 算法和辅助 UFastSLAM 算法的对比仿真实验结果表明了该算法的有效性.

1 SLAM 问题概率模型

如图 1 所示,移动机器人在一个未知环境中向目标移动,同时其自身携带的传感器对环境进行持续观测.其中,在 k 时刻, $\mathbf{x}_k$ 为移动机器人的位姿状态向量; $\mathbf{u}_k$ 为控制向量; $\mathbf{m}_i$ 为环境中第 i 个环境特征的位置状态向量; z_{ik} 为移动机器人对第 i 个环境特征进行的一次观测; $X_{0:k}$ 为移动机器人位姿状态的历史信息; $U_{0:k}$ 为控制输入的历史信息; $\mathbf{m}$ 为所有环境特征位置状态的集合; $Z_{0:k}$ 为所有观测的集合. SLAM 问题的环境特征观测模型和机器人运动模型分别可以用 $P(z_k | \mathbf{x}_k, \mathbf{m})$ 和 $P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ 表示,则在 k 时刻, SLAM 问题可以描述为条件于观测历史信息 $Z_{0:k}$ 、控制输入历史信息 $U_{0:k}$ 和移动机器人初始位姿状态 $\mathbf{x}_0$ 的后验概率分布函数,即 $P(\mathbf{x}_k, \mathbf{m} | Z_{0:k}, U_{0:k}, \mathbf{x}_0)$.

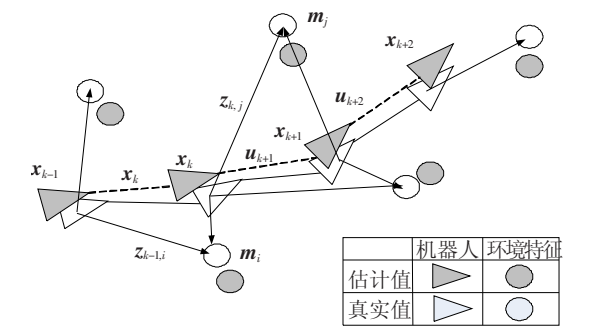


图 1 SLAM 问题概率模型描述

进而,运用概率方法解决 SLAM 问题可以归结为由时间更新和观测更新两个步骤构成的递归过程为:

1) 时间更新为

$$P(\mathbf{x}_k, \mathbf{m} | Z_{0:k-1}, U_{0:k}, \mathbf{x}_0) = \int P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \times P(\mathbf{x}_{k-1}, \mathbf{m} | Z_{0:k-1}, U_{0:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1}.$$

2) 观测更新为

$$P(\mathbf{x}_k, \mathbf{m} | Z_{0:k}, U_{0:k}, \mathbf{x}_0) = \frac{P(z_k | \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} | Z_{0:k-1}, U_{0:k}, \mathbf{x}_0)}{P(z_k | Z_{0:k-1}, U_{0:k})}.$$

2 UFastSLAM 算法基本思想及缺点

2.1 基本思想

UFastSLAM 算法的基本思想是应用 UPF 算法进行机器人位姿状态的估计,应用 UKF 算法进行环境特征位置状态的估计.其主要步骤可以分为:建议分布函数的求取、环境特征位置状态的估计和重要性权值的求取.

在建议分布函数的求取中,首先, UFastSLAM 算法运用 SUT 变换思想,根据对称采样策略对机器人位姿状态采样得到 $2n + 1$ 个 Sigma 点,并根据机器人运动模型对 Sigma 点进行非线性变换;其次,利用变换后的 Sigma 点对机器人位姿状态进行预测估计,得到预测均值和协方差,同时求出 Sigma 点序列的均值及协方差的权值;最后,进行观测更新,得到机器人位姿状态估计的均值和协方差,同时构建 UFastSLAM 算法的建议分布函数,并从中进行粒子采样.

在环境特征位置状态的估计中,首先,与求取建议分布函数一样,运用 SUT 变换思想,根据对称采样策略对环境特征位置状态采样得到 $2n + 1$ 个 Sigma 点,并跟据观测模型对 Sigma 点进行非线性变换;其次,利用变换后的 Sigma 点进行预测观测,进而得到观测新息,新息协方差矩阵和卡尔曼增益;最后,利用卡尔曼增益对环境特征位置进行更新. UFastSLAM 算法的重要性权值的求取与 FastSLAM 算法一样,均是高斯分布的形式.

2.2 UFastSLAM 算法的缺点

在整个过程中, UFastSLAM 算法通过 UKF 算法对非线性模型进行直接处理,一方面避免了繁琐的雅克比矩阵的求解,降低了计算复杂度;另一方面也消除了 EKF 算法线性化处理带来的近似误差.

然而, UFastSLAM 算法存在 2 个与 FastSLAM 算法相同的缺陷:1) 由于采样的粒子中不包含最近的观测信息,会与系统真实状态之间存在较大的差异,粒子权值稳定性不高;2) UFastSLAM 算法依然是基于序贯重要性采样的 SLAM 方法,粒子退化问题无法避免^[9],粒子退化的程度可以用有效样本尺度(Neff)^[10]有效地进行衡量,其一般近似计算为

$$\hat{N}_{\text{eff}} = \frac{N}{(1 + \text{var}(\tilde{\omega}_k^{(i)}))}.$$

式中: N_{eff} 越小说明退化问题越严重. UFastSLAM 算法用重采样方法对粒子退化进行缓解, 每当重采样时, 由于不是每个粒子都会被选, 会使整个机器人位姿历史信息 and 地图估计信息都随着没有被选的粒子而被永远遗忘, 从而会造成 UFastSLAM 算法不能在足够长的时间内保持一致性. 文献 [11] 提出用归一化估计方差 (NEES) 来检验 SLAM 算法的一致性, 并定义

$$\text{NEES} = \varepsilon_k = \tilde{\mathbf{x}}_{klk}^T \mathbf{P}_{klk}^{-1} \tilde{\mathbf{x}}_{klk}.$$

由于 UFastSLAM 算法的重采样过程会使粒子耗尽, 使得协方差 $\mathbf{P}_k$ 逐渐减小, 同时 NEES 会逐渐增大, 直至超出临界点, 则判定 UFastSLAM 算法出现不一致.

NEES 是一种加权距离, 当采用 n 次 Monte-carlo 实验时, 需要用到平均的 NEES (Mean NEES, MNEES) 来进行判断, 即

$$\text{MNEES} = 1/n \sum_{i=1}^n \text{NEES}_i.$$

3 基于辅助粒子滤波算法的粒子采样和重采样

辅助粒子滤波 (APF) 算法^[12] 的主要思想是在序列重要性采样 (SIS) 的基础上, 通过引入一个辅助变量 $j^{(i)}$ ($i, j = 1, 2, \dots, N$), 将下一时刻的观测量用于当前时刻的系统状态估计, 利用最近的观测量求取建议分布函数, 增加观测似然度高的粒子的影响力. 同时进行两次加权值的计算, 使得重采样之后能够使观测似然度大的粒子在粒子集合中保持足够大的比重. 其具体方法和实现步骤为:

1) 在 $k-1$ 时刻系统状态 $\mathbf{x}_{k-1}$ 已知的情况下, 首先, 通过计算获得 $\mathbf{x}_k$ 的均值等特征为

$$\mu_k^{(j^{(i)})} = E[\mathbf{x}_k | \mathbf{x}_{k-1}^{(j^{(i)})}].$$

利用 $\mu_k^{(j^{(i)})}$ 近似获取 k 时刻的观测似然度函数 $p(\mathbf{z}_k | \mu_k^{(j^{(i)})})$, 并将其应用于计算建议分布函数得到 $q(\mathbf{x}_k, j^{(i)} | \mathbf{z}_{1:k})$, 使得建议分布函数 $q(\mathbf{x}_k, j^{(i)} | \mathbf{z}_{1:k})$ 包含了最近的观测信息, 从而变得更加稳定.

2) 进行粒子采样, 同时进行一次加权值的计算来获取新的粒子索引集合, 即观测似然度大的粒子索引集合 $\{j^{(i)}\}_{i=1}^N$.

3) 计算 2 次加权值, 并基于此在新的粒子集合中进行重采样, 从而使观测似然度大的粒子保持足够大的比重, 使粒子云朝期望方向移动的概率增加.

4 辅助 UFastSLAM 算法的具体实现

4.1 分布函数的求取

首先, 给出 $k-1$ 时刻对应于第 i 个粒子的联合状态向量 $\mathbf{x}_{k-1}^{[i]}$ 和状态估计误差协方差矩阵 $\mathbf{P}_{k-1}^{a[i]}$ 的增广矩阵为

$$\begin{aligned} \mathbf{x}_{k-1}^{a[m]} &= [\mathbf{x}_{k-1}^{[i]} \quad 0 \quad 0]^T = \\ &[\mathbf{x}_{x,k-1}^{[i]} \quad \mathbf{x}_{y,k-1}^{[i]} \quad \mathbf{x}_{\theta,k-1}^{[i]} \quad 0 \quad 0]_{1 \times 7}^T, \\ \mathbf{P}_{k-1}^{a[i]} &= \text{diag}[\mathbf{P}_{k-1}^{[i]} \quad \mathbf{Q}_k \quad \mathbf{R}_k]_{7 \times 7}. \end{aligned}$$

式中: $\mathbf{Q}_k$ 、 $\mathbf{R}_k$ 分别为控制噪声协方差阵和观测噪声协方差阵.

对机器人位姿状态进行 Sigma 点采样, 并进行 SUT 变换为

$$\begin{cases} \chi_{k-1}^{a[i][0]} = \mu_k^{a[j^{(i)}]}; \\ \chi_{k-1}^{a[i][m]} = \mu_k^{a[j^{(i)}]} + (\sqrt{(L + \lambda)\mathbf{P}_{k-1}^{a[i]}})_m, \\ m = 1, 2, \dots, L; \\ \chi_{k-1}^{a[i][m]} = \mu_k^{a[j^{(i)}]} - (\sqrt{(L + \lambda)\mathbf{P}_{k-1}^{a[i]}})_{m-L}, \\ m = L + 1, L + 2, \dots, 2L. \end{cases}$$

式中: L 为机器人位姿状态维数; $\lambda = \alpha^2(L + \kappa) - L$ 为选择尺度; α 为尺度参数, 决定 Sigma 点的分布范围, 取值范围为 $0 < \alpha < 1$, 一般是一个很小的值; κ ($\kappa \geq 0$) 是一个比例参数, 决定选取的 Sigma 点距离均值为多远合适, 由于没有决定性的作用, 通常取值为 0. 每个 Sigma 点 $\chi_{k-1}^{a[i][m]}$ 都包含机器人位姿状态分量 $\chi_{k-1}^{[i][m]}$ 、控制噪声分量 $\chi_{k-1}^{u[i][m]}$ 和观测分量 $\chi_{k-1}^{z[i][m]}$ 3 个部分, 表示为

$$\chi_{k-1}^{a[i][m]} = [\chi_{k-1}^{[i][m]} \quad \chi_{k-1}^{u[i][m]} \quad \chi_{k-1}^{z[i][m]}]^T.$$

基于机器人运动模型将 Sigma 点进行非线性变换, 得到 k 时刻机器人位姿状态的估计均值的 Sigma 点为

$$\begin{aligned} \bar{\chi}_k^{[i][m]} &= f(u_k^{[i]} + \chi_k^{u[i][m]}, \chi_{k-1}^{[i][m]}) = \\ &[\bar{\chi}_{x,k}^{[i][m]} \quad \bar{\chi}_{y,k}^{[i][m]} \quad \bar{\chi}_{\theta,k}^{[i][m]}]^T. \end{aligned} \quad (1)$$

式中: $u_k^{[i]}$ 为 k 时刻控制输入; $\chi_k^{u[i][m]}$ 为 k 时刻控制噪声 Sigma 点.

利用 $\bar{\chi}_k^{[i][m]}$ 进行机器人位姿状态的预测估计. 预测估计均值为

$$\mathbf{x}_{klk-1}^{[i]} = \sum_{m=0}^{2L} \omega_g^{(m)} \bar{\chi}_k^{[i][m]}.$$

预测估计误差协方差为

$$\begin{aligned} \mathbf{P}_{klk-1}^{[i]} &= \sum_{m=0}^{2L} \omega_c^{(m)} (\bar{\chi}_k^{[i][m]} - \mathbf{x}_{klk-1}^{[i]}) \cdot \\ &(\bar{\chi}_k^{[i][m]} - \mathbf{x}_{klk-1}^{[i]})^T. \end{aligned}$$

式中: $\omega_g^{[m]}$ 、 $\omega_c^{[m]}$ 分别为 Sigma 点序列的均值及协方差的权值.

$$\omega_g^{[0]} = \frac{\lambda}{L + \lambda}, \omega_c^{[0]} = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta),$$

$$\omega_g^{[m]} = \omega_c^{[m]} = \frac{1}{2(L + \lambda)}, m = 1, 2, \dots, 2L.$$

式中: β 用于吸收后面时刻的后验分布知识, 如果系统状态服从高斯分布, 则 $\beta = 2$ 为最优。

然后, 进行观测更新, 选取观测 Sigma 点为

$$\bar{\gamma}_k^{[i][m]} = h(\chi_k^{[i][m]}, u_{k-1}^{[m]}) + \chi_k^{z[i][m]}.$$

预测观测为

$$\hat{z}_k^{[i]} = \sum_{m=0}^{2L} \omega_g^{(m)} \bar{\gamma}_k^{[i][m]}.$$

新息协方差为

$$S_{kl\ k-1}^{[i]} = \sum_{m=0}^{2L} \omega_c^{[m]} (\bar{\gamma}_k^{[i][m]} - \hat{z}_k^{[i][m]}) \cdot (\bar{\gamma}_k^{[i][m]} - \hat{z}_k^{[i][m]})^T.$$

交叉协方差为

$$P_{kl\ k-1}^{xz[i]} = \sum_{m=0}^{2L} \omega_c^{[m]} (\bar{\chi}_k^{[i][m]} - x_{kl\ k-1}^{[i]}) \cdot (\bar{\gamma}_k^{[i][m]} - \hat{z}_k^{[i][m]})^T.$$

卡尔曼增益为

$$K_k^{[i]} = P_{kl\ k-1}^{xz[i]} (S_{kl\ k-1}^{[i]})^{-1}.$$

更新得到 k 时刻的机器人位姿状态估计的均值和协方差为

$$x_k^{[i]} = x_{kl\ k-1}^{[i]} + K_k^{[i]} (z_k - \hat{z}_k^{[i]}),$$

$$P_k^{[i]} = P_{kl\ k-1}^{[i]} - K_k^{[i]} S_{kl\ k-1}^{[i]} (K_k^{[i]})^T.$$

最后, 利用机器人位姿状态估计均值 $x_k^{[i]}$ 和协方差 $P_k^{[i]}$ 构建辅助 UFastSLAM 算法的建议分布函数, 并从中进行粒子采样为

$$x_k^{[i]} \sim N(x_k^{[i]}, P_k^{[i]}). \quad (2)$$

由式(1)可知, $\bar{\chi}_k^{[i][m]}$ 中已经包含了 $\mu_{k+1}^{a[j(i)]}$ 所携带的信息, 并参与了机器人位姿状态估计均值 $x_k^{[i]}$ 和协方差 $P_k^{[i]}$ 的预测和更新之中, 使得辅助 UFastSLAM 算法的建议分布函数中包含了最近的观测信息。

4.2 1次加权

根据 $p(z_k | \mu_k^{a[j(i)]})$ 计算 k 时刻的观测似然度函数, 继而计算 1 次加权值:

$$p(j^{[i]} | z_{1:k}) \propto p(z_k | \mu_k^{a[j(i)]}) \tilde{\omega}_{k-1}^{[i]}. \quad (3)$$

由此, 在根据式(2)进行采样的同时, 可以由式(3)得到 k 时刻的新的粒子索引集合 $\{j^{(i)}\}_{i=1}^N$. 其中, $\tilde{\omega}_{k-1}^{[i]}$ 为 $k-1$ 时刻的归一化权值。

4.3 2次加权并重采样

根据基于 k 时刻的粒子索引号的粒子集合, 对 $k-1$ 时刻的粒子集合 $\{x_{k-1}^{(i)}\}_{i=1}^N$ 进行重采样, 得到新的粒子索引集合 $\{x_{k-1}^{[j(i)]}\}_{i=1}^N$, 继而计算

$$\hat{x}_{kl\ k-1}^{[i]} = p(x_k | x_{k-1}^{[j(i)]}).$$

基于粒子状态预测值 $\hat{x}_{kl\ k-1}^{[i]}$ 求取 k 时刻的观测似然度函数 $p(z_k | \hat{x}_{kl\ k-1}^{[i]})$, 则计算 2 次加权值得

$$w_k^{[i]} = p(z_k | \hat{x}_{kl\ k-1}^{[i]}) / p(z_k | \mu_k^{[j(i)]}).$$

并进行归一化

$$\tilde{w}_k^{[i]} = w_k^{[i]} / \sum_{i=1}^N w_k^{[i]}. \quad (4)$$

根据式(4)的值计算有效样本尺度 N_{eff} 并决定是否进行 k 时刻粒子集合的重采样。

4.4 环境特征位置状态的估计

应用 UKF 算法进行环境特征位置的估计。 k 时刻, Sigma 点采样为

$$\begin{cases} \chi^{[i][0]} = \mu_{k-1}^{[i]}; \\ \chi^{[i][m]} = \mu_{k-1}^{[i]} + \left(\sqrt{(n + \lambda) \Sigma_{k-1}^{[i]}} \right)_m, \\ m = 1, 2, \dots, n; \\ \chi^{[i][m]} = \mu_{k-1}^{[i]} - \left(\sqrt{(n + \lambda) \Sigma_{k-1}^{[i]}} \right)_{m-n}, \\ m = n + 1, n + 2, \dots, 2n. \end{cases}$$

式中: $\mu_{k-1}^{[m]}$ 、 $\Sigma_{k-1}^{[m]}$ 分别为 $k-1$ 时刻的环境特征位置估计的均值和协方差; n 为该时刻环境特征状态向量的维数。

Sigma 点非线性变换为

$$\bar{Z}_k^{[i][m]} = h(\chi^{[i][m]}, x_k),$$

$$\hat{z}_k^{[i]} = \sum_{m=0}^{2n} \omega_g^{[m]} \bar{Z}_k^{[i][m]},$$

$$\bar{S}_k^{[i]} = \sum_{m=0}^{2n} \omega_c^{[m]} (\bar{Z}_k^{[i][m]} - \hat{z}_k^{[i]}) (\bar{Z}_k^{[i][m]} - \hat{z}_k^{[i]})^T,$$

$$\bar{\Sigma}_k^{[i]} = \sum_{m=0}^{2n} \omega_c^{[m]} (\chi^{[i][m]} - u_{k-1}^{[i]}) (\bar{Z}_k^{[i][m]} - \hat{z}_k^{[i]})^T,$$

$$\bar{K}_k^{[i]} = \bar{\Sigma}_k^{[i]} (\bar{S}_k^{[i]})^{-1}.$$

最后进行环境特征位置估计的均值和协方差的更新为

$$u_k^{[i]} = u_{k-1}^{[i]} + \bar{K}_k^{[i]} (z_k - \hat{z}_k^{[i]}),$$

$$\Sigma_k^{[i]} = \Sigma_{k-1}^{[i]} - \bar{K}_k^{[i]} \bar{S}_k^{[i]} (\bar{K}_k^{[i]})^T.$$

4.5 重要性权值的计算

UFastSLAM 算法的重要性权值为

$$\omega_k^{[i]} = |2\pi\Gamma_k^{[i]}|^{-\frac{1}{2}} e^{-\frac{1}{2}(z_k - \hat{z}_k^{[i]})(\Gamma_k^{[i]})^{-1}(z_k - \hat{z}_k^{[i]})^T}.$$

其中

$$\Gamma_k^{[i]} = (\Sigma_k^{[i]})^T (P_k^{[i]})^{-1} \Sigma_k^{[i]} + \bar{S}_k^{[i]}.$$

5 分 析

基于统一的机器人运动模型、环境特征观测模型、实验环境、噪声模型等, 粒子数为 50, 进行 FastSLAM 算法、UFastSLAM 算法和辅助 UFastSLAM 算法的对比仿真实验。

机器人运动模型为

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = f \left[x_{k-1}, u_k, v_k, k \right] = \begin{bmatrix} x_{k-1} + \frac{\Delta D_k}{\Delta \theta_k} [\cos(\theta_{k-1} + \Delta \theta_k) - \cos \theta_{k-1}] \\ y_{k-1} + \frac{\Delta D_k}{\Delta \theta_k} [\sin(\theta_{k-1} + \Delta \theta_k) - \sin \theta_{k-1}] \\ \theta_{k-1} + \Delta \theta_k \end{bmatrix} + v_k.$$

式中: ΔD_k 为机器人在时间间隔 ΔT 时间内的运动圆弧长度; v_k 为系统噪声.

观测模型为

$$z_k = \begin{bmatrix} \rho_k \\ \varphi_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ \arctan \frac{y_k - y_i}{x_k - x_i} - \theta_k \end{bmatrix} + w_k.$$

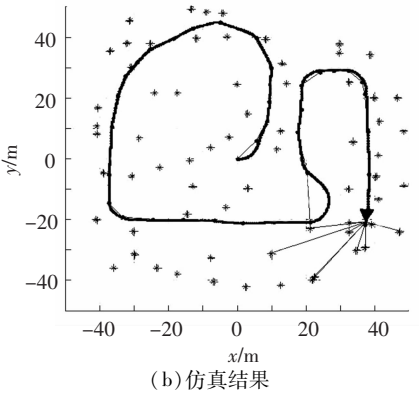
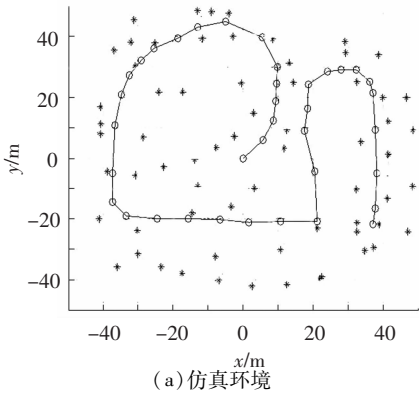


图 2 移动机器人 SLAM 仿真环境和结果

图 3 (a) ~ (c) 分别是 FastSLAM 算法、UFastSLAM 算法和辅助 UFastSLAM 算法的 x 方向、 y 方向的机器人定位误差和定向误差,不难看出,辅助 UFastSLAM 算法估计精度明显比 UFastSLAM 算法和 FastSLAM 算法高.

并且,如表 1、2 所示,当辅助 UFastSLAM 算法,UFastSLAM 算法和 FastSLAM 算法的粒子数均为 50 时,其定位的均方根误差 (RMSE) 依次递增,但计算复杂度前两者稍高于第三者;当使 3 种算法的定位均方根误差大致相当时,其需要的粒

子数则亦依次递增,且计算复杂度大致相当.

图 4 (a)、(b) 分别是 FastSLAM 算法、UFastSLAM 算法和辅助 UFastSLAM 算法对随机抽取的 2 号环境特征位置估计的 x 方向、 y 方向误差.不难看出,辅助 UFastSLAM 算法的估计精度明显比 UFastSLAM 算法和 FastSLAM 算法高.

FastSLAM 算法、UFastSLAM 算法和辅助 UFastSLAM 算法的 MNEES 曲线如图 5 所示.采用 100 次 Monte-carlo 实验,并取显著性水平为 $\alpha = 0.05$,则

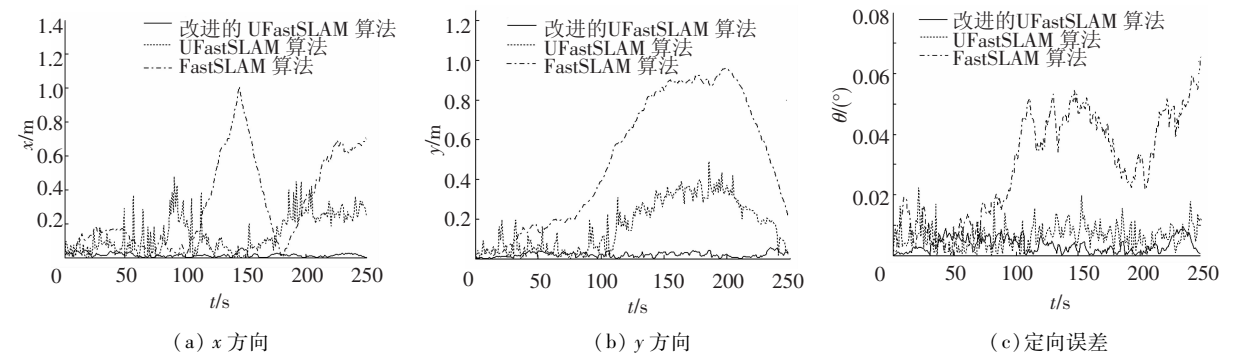


图 3 机器人定位误差

NEES = $\chi^2_{100 \cdot 3, 0.95}$ = 341.4.

那么

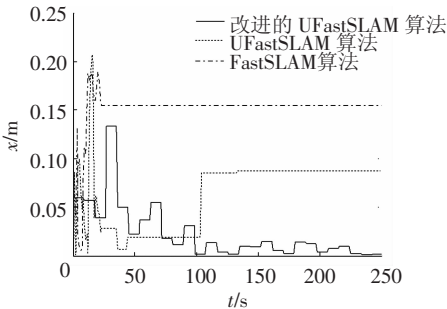
MNEES = $0.01\chi^2_{100 \cdot 3, 0.95}$ ≈ 3.414.

表 1 定位实验结果对比(相同粒子数)

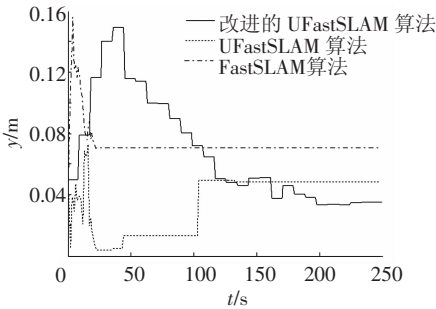
SLAM 算法(粒子数)	定位的均方根误差		
	x 方向/m	y 方向/m	方向角 $\theta/ (^{\circ})$
改进的 UFastSLAM(50)	0.019 3	0.023 2	0.004 1
UFastSLAM(50)	0.181 8	0.213 3	0.007 6
FastSALM(50)	0.411 4	0.598 0	0.034 4

表 2 定位实验结果对比(相近误差)

SLAM 算法(粒子数)	定位的均方根误差		
	x 方向/m	y 方向/m	方向角 $\theta/ (^{\circ})$
改进的 UFastSLAM(8)	0.400 8	0.594 1	0.036 2
UFastSLAM(18)	0.412 4	0.587 4	0.033 5
FastSALM(50)	0.411 4	0.598 0	0.034 4



(a) 2 号特征 x 方向误差



(b) 2 号特征 y 方向误差

图 4 2 号环境特征位置估计误差

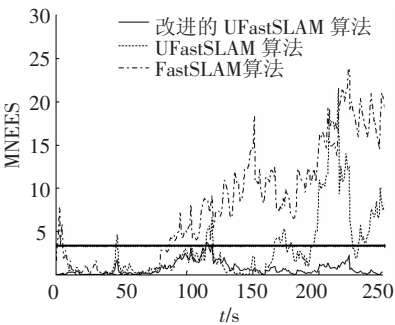


图 5 MNEES 曲线

由图 5 可以看出, FastSLAM 算法大约在 80 s 后趋于不一致, UFastSLAM 算法大约在 200 s 后趋于不一致, 而辅助 UFastSLAM 算法的一致性一直保持很好.

6 结 论

1) 将最近的观测信息引入建议分布函数的计算, 增强采样粒子权值的稳定性.

2) 在重采样过程中, 通过两次加权, 增大观测似然度大的粒子比例.

3) 理论分析和实验结果表明, 辅助 UFastSLAM 算法在估计精度和一致性方面都很好.

参考文献:

[1] 骆云祥. 非线性滤波在移动机器人 SLAM 中的运用 [D]. 南京: 南京理工大学, 2009.

[2] 赵季. 基于激光扫描仪的同步定位与建图研究 [D]. 武汉: 武汉科技大学, 2008.

[3] 李久胜, 李永强, 周荻. 基于 EKF 的 SLAM 算法的一致性分析 [J]. 计算机仿真, 2008, 25(6): 155 – 160.

[4] DURRANT-WHYTE H, BAILEY T. Simultaneous localization and mapping (SLAM): Part I [J]. IEEE Robotics and Automation Magazine, 2006, 13(2): 99 – 110.

[5] JULIER S, UHLMANN J, DURRANT-WHYTE H F. A new method for the nonlinear transformation of means and covariances in filters and estimators [J]. IEEE Transactions on Automatic Control, 2000, 45(3): 477 – 482.

[6] MONTEMERLO M, THRUN S, KOLLER D, et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem [C]//Proceeding Eighteenth National Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 2002: 593 – 598.

[7] KIM Chanki, SAKTHIVEL R, CHUNG Wan-Kyun. Unscented FastSLAM: a robust and efficient solution to the SLAM problem [J]. IEEE Transactions on Robotics, 2008, 24(4): 808 – 820.

[8] Van Der MERWE R, DOUCET A, De FREITAS N, et al. The Unscented Particle Filter [R]. Portland, OR, USA: Oregon Graduate Institute, 2000.

[9] DOUEET A, GODSILL S, ANDRIEU C. On sequential Monte Carlo sampling methods for Bayesian filtering [J]. Statistics and Computing, 2000, 10(3): 197 – 208.

[10] BERGMAN N. Recursive Bayesian estimation: navigation and tracking application [D]. Linkoping: Linkoping University, 1999.

[11] BAILEY T, NIETO J, NEBOT E. Consistency of the FastSLAM algorithm [C]//Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Washington, DA: IEEE Xplore, 2006: 424 – 429.

[12] PITT M, SHEPHARD N. Filtering via simulation; auxiliary particle filters [J]. Journal of the American Statistical Association (S0162 – 1459), 1999, 94(446): 590 – 599.

(编辑 张 红)