

基于一致性的FastSLAM算法的优化

耿小毛

上海理工大学理学院, 上海

收稿日期: 2023年12月13日; 录用日期: 2023年12月28日; 发布日期: 2024年1月30日

摘要

本文从可观测性的角度研究FastSLAM算法的一致性问题并对算法进行了改进。相比较于传统的FastSLAM算法, 一种改进的FastSLAM算法被提出从而获得更好的一致性性能。首先, 在重要性采样阶段, 对象标记法用于清晰地标注单个粒子状态。此外, 在地图估计阶段, 将第一估计雅可比矩阵(FEJ)与扩展卡尔曼滤波相结合以此提高了算法的一致性。最后, 通过仿真实例验证了改进的FastSLAM算法的有效性。

关键词

移动机器人, 扩展卡尔曼滤波, FastSLAM, Kullback-Leibler散度(KLD)采样

Optimization of FastSLAM Algorithm Based on Consistency

Xiaomao Geng

College of Science, University of Shanghai for Science and Technology, Shanghai

Received: Dec. 13th, 2023; accepted: Dec. 28th, 2023; published: Jan. 30th, 2024

Abstract

In this paper, the consistency of FastSLAM algorithm is improved from the perspective of observability. Rather than the traditional ones, an improved FastSLAM algorithm is proposed to achieve better consistent performance of FastSLAM. First, in the importance sampling phase, the object labeling method is used to clearly label individual particle states. Moreover, in the stage of map estimation, the First Estimates Jacobian (FEJ) is combined with the extended Kalman filtering to improve the consistency of the algorithm. Finally, the effectiveness of the proposed improved FastSLAM algorithm is validated through a simulation example.

Keywords

First Estimates Jacobian (FEJ), Extended Kalman Filtering, FastSLAM, Kullback-Leibler Divergence (KLD) Sampling

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

同步定位与建图(SLAM)是近年来移动机器人领域的关键问题。随着移动机器人执行任务时面临的环境越来越复杂, SLAM 的研究和发展已经吸引越来越多专家、学者和社会的关注。

SLAM 问题[1], 顾名思义, 可以分解为两个问题: 1) 确定自己的位姿; 2) 周边环境特征建模。SLAM 问题的实现大致可以分为两类: 一类是基于贝叶斯概率模型, 另一类是基于非概率模型。而基于贝叶斯概率模型的方法, 可以从扩展卡尔曼滤波、无迹卡尔曼滤波和粒子滤波的角度分析和解决 SLAM 问题[2]。在 EKF-SLAM 问题中, EKF 可以帮助处理移动机器人同时定位和建图过程中的不确定性, 但也容易造成计算量大和不一致性的问题。为了减小线性化误差, UKF-SLAM [3]被提出, 但外部干扰的影响会导致 UKF-SLAM 算法的精度和鲁棒性降低。为了克服基于卡尔曼滤波的 SLAM 算法的一些问题, Montemerlo 等[4]提出了基于粒子滤波的 FastSLAM 算法。FastSLAM 算法采用粒子集表示机器人的位姿, 并使用 EKF 更新环境地图的特征[5]。

2001 年 Julier 和 Uhlmann [6]首次提出了 SLAM 的一致性问题。由于 SLAM 是一个非线性问题, 计算出的协方差不能保证与实际估计误差匹配。因此, 一致性问题对于 SLAM 系统的准确性和鲁棒性非常重要。在[7]中, 我们可以了解到, 如果估计误差(i)为零均值, 且(ii)协方差矩阵小于或等于由滤波计算出的协方差矩阵, 则状态估计器是一致的。近年来基于卡尔曼滤波的 SLAM 算法一致性研究为解决这一问题提供了新的思路。2004 年, J. A. Castellanos [8]提出了一种新的基于 EKF-robotcentric mapping 的 SLAM 算法, 大大降低了线性化误差, 提高了建图的一致性。2017 年, 黄国权[9]通过将正则雅可比矩阵直接投影到信息可用子空间中, 或通过寻找最优线性化点间接计算适当的滤波雅可比矩阵, 使滤波器仅从每个源的测量中获得实际可用信息, 从而大大提高了一致性。

FastSLAM 算法结合了粒子滤波和扩展卡尔曼滤波的优点, 对 SLAM 系统进行了分析和研究, 因此 FastSLAM 算法的一致性也是一个值得关注的问题。基于以上讨论, 本文将开发一种改进的 FastSLAM 算法, 通过改进扩展卡尔曼滤波算法在地图估计阶段的一致性, 以及在采样阶段使用目标标记方法标记单个粒子, 增强 SLAM 系统的一致性。主要贡献如下: 1) 提出了一种改进了一致性的 FastSLAM 算法。2) 提出了一种新的目标标记方法, 可以清晰地标记单个粒子的状态。3) 为了提高 SLAM 过程中估计量的一致性, 提出了第一估计雅可比矩阵(FEJ)扩展卡尔曼滤波算法。4) 在重采样阶段提出了 Kullback-Leibler 散度(KLD)采样方法, 动态改变粒子总数。本文剩余部分内容的组织框架如下: 第 2 节介绍了传统的 FastSLAM 算法。第 3 节详细介绍了改进的 FastSLAM 算法, 并且这种改进的算法通过改变采样方法, 使用每个状态变量的第一个可用估计来计算滤波器雅可比矩阵, 从而提高了算法的一致性。第 4 节介绍了 FastSLAM 算法的一致性测量方法。此外, 在第 5 节中给出了仿真, 以验证改进 FastSLAM 算法的有效性。最后, 在本篇文章的第 6 节给出总结。

2. 传统的 FastSLAM 算法

粒子滤波是一种用样本来表示概率密度分布的蒙特卡罗方法。该方法可以用在任何状态模型中，并且当样本空间面积无限大时，可以用来近似任何形式的概率密度分布[10]。Murphy 等人[11]引入了一种更有效的 Rao-Blackwelized 粒子滤波(RBPF)来解决 SLAM 问题，更有效地利用了粒子滤波，为 SLAM 的研究提供了理论基础。RBPF 是一种逼近贝叶斯解的方法，贝叶斯方法可以保持多个假设，具有很强的鲁棒性。Montemerlo 等人[4] [12]提出了基于粒子滤波的 FastSLAM 算法。FastSLAM 算法的基本思想是将联合后验概率分布分解为机器人路径部分和以机器人路径为条件的建图部分。传统 FastSLAM 算法的主要过程主要包括以下四个步骤：位姿采样与更新、地图更新、粒子权重更新、粒子重采样[5]，如图 1 所示。

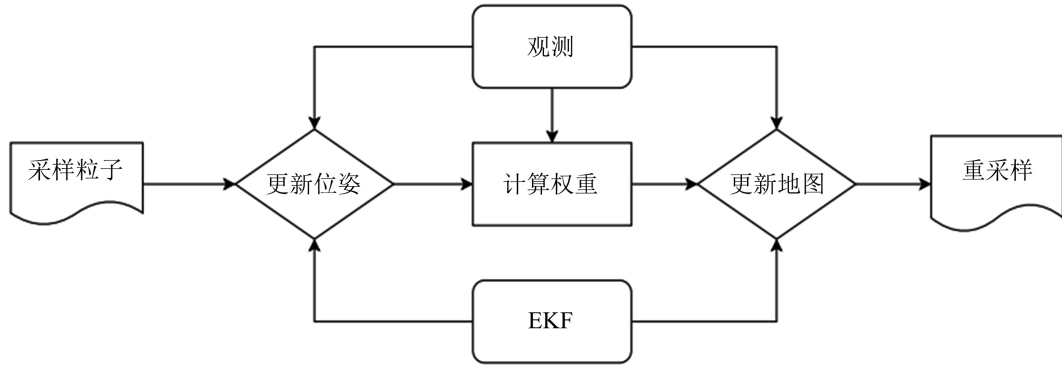


Figure 1. Traditional FastSLAM algorithm flow chart
图 1. 传统的 FastSLAM 算法流程图

2.1. 姿态采样与更新

FastSLAM 算法是一种基于粒子滤波的 SLAM 方法。在 FastSLAM 中，一个粒子代表一个机器人的潜在轨迹和环境地图。具体地，一个粒子包含了机器人的位姿 x_t 和与之对应的 M 个路标估计器 $l_i, i=1, \dots, M$ ，这些估计器用于跟踪机器人与特定路标之间的相对位置。SLAM 后验估计分成两部分：机器人自身定位问题和基于条件独立的 M 个相互独立路标的估计问题后验估计[11]。

$$p(x_{1:t}, l_{1:M} | z_{1:t}, u_{1:t}, K_t) = p(x_{1:t} | z_{1:t}, u_{1:t}, K_t) \cdot \prod_{i=1}^M p(l_i | x_{1:t}, z_{1:t}, u_{1:t}, K_t) \quad (1)$$

其中 $z_{1:t}$ 为时间步骤 1 到 t 的观测值， $u_{1:t}$ 为时间步骤 1 到 t 的控制输入， $K_t = \{k_1, \dots, k_t\}$ 为时刻 t 感知到的地标指数集。

$$X_t = \{x_{1:t}^{[n]}\}_n = \{x_1^{[n]}, x_2^{[n]}, \dots, x_t^{[n]}\} \quad (2)$$

一个粒子 $x_{1:t}^{[n]} \in X_t$ 表示一个机器人的位姿。我们使用上标符号 $[n]$ 来表示集合中的第 n 个粒子。在 FastSLAM 过程中，利用粒子滤波算法计算机器人位姿的后验概率分布，并进行重要性采样。然而，后验概率分布一般是事先不知道的，因此很难从后验概率分布中进行采样。因此，假设一个概率分布函数 $x_t^{[n]} \sim p(X_t | x_{t-1}^{[n]}, u_t)$ ，它尽可能接近后验概率分布。这种概率分布函数称为提议分布或重要性密度函数。

2.2. 更新地图

FastSLAM 使用 EKF 来执行有条件的地标位置估计和更新地图。

$$p(l_i | x_{1:t}, z_{1:t}, u_{1:t}, K_t) \quad (3)$$

FastSLAM 算法中的完整后验路径和地标位置由集合 X_t 表示, 其中 $\lambda_m^{[n]}$ 和 $\Theta_m^{[n]}$ 表示和第 n 个粒子相关的第 m 个地标 l_i 的均值和协方差。

$$X_t = \left\{ x_{1:t}^{[n]}, \lambda_1^{[n]}, \Theta_1^{[n]}, \dots, \lambda_m^{[n]}, \Theta_m^{[n]} \right\}_n \quad (4)$$

可以根据是否 $k_i = m$ 来计算第 m 个地标位置 l_m 的后验, 即在时间 t 时是否观察到 l_m 。

对于 $k_i = m$, 我们得到

$$p(l_m | x_{1:t}, z_{1:t}, u_{1:t}, K_t)$$

$$\stackrel{Bayes}{\propto} p(z_t | l_m, x_{1:t}, z_{1:t-1}, u_{1:t}, K_t) p(l_m | x_{1:t}, z_{1:t-1}, u_{1:t}, K_t) \quad (5)$$

$$\stackrel{Markov}{=} p(z_t | l_m, x_t, k_t) p(l_m | x_{1:t-1}, z_{1:t-1}, u_{1:t-1}, K_{t-1}) \quad (6)$$

对于 $k_i \neq m$, 我们简单地保持高斯函数不变:

$$p(l_m | x_{1:t}, z_{1:t}, u_{1:t}, K_t) = p(l_m | x_{1:t-1}, z_{1:t-1}, u_{1:t-1}, K_{t-1}) \quad (7)$$

在 FastSLAM 算法中, 使用 EKF 来更新方程(5)。机器人的运动模型使用粒子滤波来估计, 而测量模型则使用 EKF 来近似, 从而得到一个完全高斯分布 $p(l_m | x_{1:t}, z_{1:t}, u_{1:t}, K_t)$, 以便更好地估计机器人的位置和地图的状态。这是利用采样来近似机器人姿态分布的结果[4]。

2.3. 重采样粒子

在最后一步, 在更新完所有 N 个粒子后, 计算粒子的权重系数 $\omega_t^{[n]}$, 粒子的权重系数是由目标分布(target distribution)和提议分布(proposal distribution)的比值组成, 然后通常计算有效样本大小 $N_{eff} = \frac{1}{\sum_{n=1}^N (\omega_t^{[n]})^2}$,

并选择一个阈值 N_{th} , 如果 $N_{eff} < N_{th}$, 表示粒子差异较大, 则进行重采样, 去除权重系数较小的粒子。

粒子权重系数的计算公式如下:

$$\omega_t^{[n]} = \frac{p(x_{1:t}^{[n]} | z_{1:t}, u_{1:t}, K_t)}{p(x_{1:t}^{[n]} | z_{1:t-1}, u_{1:t}, K_{t-1})} \quad (8)$$

3. 改进的 FastSLAM 算法

注意到传统的 FastSLAM 算法受到线性化和重采样引起的粒子退化和粒子贫乏问题[10]的困扰, 以及由这些问题引起的不一致等情况。因此, 我们迫切需要开发一种高效、新颖的 FastSLAM 算法。对于传统 FastSLAM 算法不一致问题的改进或解决, 主要是从改进算法重采样[13] [14]的角度着手。结合改进的重采样方法, 提出了一种将改进的粒子采样和地图估计相结合的 FastSLAM 算法。改进后的算法将进一步保证算法的准确性和一致性。改进后的 FastSLAM 算法伪代码如图 2 所示。

3.1. 采样新姿态

在粒子采样过程中, 依赖于足够数量的粒子的算法确实可以提高算法的精度, 但与此同时却也增加了计算复杂度, 效率较低。与传统的 FastSLAM 算法相比, 为了保持足够高的算法精度和较小的计算复杂度, 该算法进行了改进。我们给采样的粒子添加标签, 并根据标签信息修改粒子的权重, 这进一步提

高了新粒子靠近正确轨迹的概率，因此保证了定位的准确性和效率。

假设粒子数为 S 。由于在 FastSLAM 中，一个粒子代表一个机器人的潜在轨迹和环境映射，我们在机器人状态 $x \in X$ 后加上一个标签 $s \in S$ 。一般来说，标签 s 是离散标签空间 $S = \{\alpha_n : n \in N\}$ 中的一个元素，它的元素 α_j 是不同的。此外，空间 N 表示正整数的集合。对于粒子集，每个粒子 $x \in X$ 都有唯一的标签。因此，标签的 RFS [15] 是空间 $X \times S$ 上的有限集值随机变量，其中标签空间的所有粒子都包含唯一的标签。

如[16]中所介绍的，为每个粒子分配一个标签 $s = (t, n)$ ，每个粒子是一对出生时间 t 和唯一的指标 $n \in N$ ，因此， t 时刻新生物体的标记状态空间为 $S_t = t \times N$ ， t 时刻之前生成的所有物体的标记状态空间为 $S_{0:t-1}$ 。由于新对象的标签集与现有对象不相交，故 t 时刻所有对象的标签空间为 $S_{0:t} = S_{0:t-1} \cup S_t$ 。

基于姿态估计，机器人姿态估计对应 M 个路标估计器 $l_i, i = 1, \dots, M$

$$p\left(x_{1:t}^{\alpha_n}, l_{1:M} \mid z_{1:t}, u_{1:t}, K_t\right) = p\left(x_{1:t}^{\alpha_n} \mid z_{1:t}, u_{1:t}, K_t\right) \cdot \prod_{i=1}^M p\left(l_i \mid x_{1:t}^{\alpha_n}, z_{1:t}, u_{1:t}, K_t\right) \quad (9)$$

机器人的位姿表示为

$$X_t^S = \left\{ x_{1:t}^{[\alpha_n]} \right\}_n = \left\{ x_1^{[\alpha_n]}, x_2^{[\alpha_n]}, \dots, x_t^{[\alpha_n]} \right\}_n \quad (10)$$

算法1 改进的FastSLAM 算法

输入：采样的粒子

输出：机器人位姿

- 1: 初始化粒子
 - 2: 给粒子贴标签并且采样新位姿
 - 3: while (新粒子被添加) do
 - 4: 对于每个粒子应用第一估计雅可比矩阵EKF去更新路标估计
 - 5: 计算并更新重要性权重
 - 6: 重采样。当有效样本数 N_{eff} 小于阈值 N_{th} 时，则表明粒子差异较大，剔除权重系数较小的粒子，此时粒子不充足
 - 7: while (粒子不充足) do
 - 8: 通过融合粒子标签数据计算新粒子是否被添加
 - 9: if 新粒子被添加 then
 - 10: 通过粒子标签模型给粒子重新加权
 - 11: else
 - 12: 低方差重采样
 - 13: end if
 - 14: 利用KLD采样确定重采样的粒子总数
 - 15: 重采样。当有效样本数 N_{eff} 小于阈值 N_{th} 时，则表明粒子差异较大，剔除权重系数较小的粒子，此时粒子不充足
 - 16: end while
 - 17: end while
-

Figure 2. Improved FastSLAM algorithm pseudo code

图 2. 改进的 FastSLAM 算法伪代码

3.2. 更新对于每个粒子的地标位置估计

在运动模型中采样新的位姿后，更新每个粒子中地标位置的估计。在这一步中，我们结合 FEJ 的思想来提高地标位置估计的准确性。

在 FastSLAM 中，利用 EKF 计算了标记位置 $P(l_i | x_{1:t}^{\alpha_n}, z_{1:t}, u_{1:t}, K_t)$ 的估计值，很容易得到第 i 个标记位置 l_i 的后验估计值。其计算取决于在时刻 t 是否观测到地标 l_i ，即是否满足条件 $i = k_t$ 。

如果在 t 时刻观测到地标，即 $i = k_t$ ，则

$$P(l_{i=k_t} | x_{1:t}^{\alpha_n}, z_{1:t}, u_{1:t}, K_t) \propto P(z_{1:t} | l_{k_t}, x_t, k_t) \cdot P(l_{k_t} | x_{1:t-1}, z_{1:t-1}, u_{1:t-1}, K_{t-1}) \quad (11)$$

如果在 t 时刻未观测到地标，即 $i \neq k_t$ ，则：

$$P(l_{i \neq k_t} | x_{1:t}^{\alpha_n}, z_{1:t}, u_{1:t}, K_t) \propto P(l_i | x_{1:t-1}^{\alpha_n}, z_{1:t-1}, u_{1:t-1}, K_{t-1}) \quad (12)$$

3.2.1. EKF-SLAM 算法的预测与更新

为了算法的描述简便，在每个粒子中，机器人位姿和地标位置信息表示为

$$X'_t = \begin{bmatrix} P_{x_t}^{\alpha_n} \\ \phi_{x_t}^{\alpha_n} \\ P_L \end{bmatrix} = \begin{bmatrix} x_t^{\alpha_n} \\ P_L \end{bmatrix} \quad (13)$$

其中 $x_t^{\alpha_n}$ 表示机器人的位姿(包括位置和方向)， P_L 表示地标的位置。EKF-SLAM 递归遵循两个步骤：预测和更新，这两个步骤分别基于离散时间过程和观测模型[10]。EKF 的预测方程为：

$$\hat{P}_{x_{t+1|t}}^{\alpha_n} = \hat{P}_{x_{t|t}}^{\alpha_n} + C(\hat{\phi}_{x_{t|t}}^{\alpha_n}) x_t \hat{P}_{x_{t+1}}^{\alpha_n} \quad (14)$$

$$\hat{\phi}_{x_{t+1|t}}^{\alpha_n} = \hat{\phi}_{x_{t|t}}^{\alpha_n} + x_t \hat{\phi}_{x_{t+1}}^{\alpha_n} \quad (15)$$

$$\hat{P}_{L_{t+1|t}} = \hat{P}_{L_{t|t}} \quad (16)$$

其中 $C(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ 为 2×2 的旋转矩阵， $x_t^{\alpha_n} \hat{X}_{t+1}^{\alpha_n} = \begin{bmatrix} x_t^{\alpha_n} \hat{P}_{x_{t+1}}^{\alpha_n} \\ x_t^{\alpha_n} \hat{\phi}_{x_{t+1}}^{\alpha_n} \end{bmatrix}$ 为机器人从 t 时刻到 $t+1$ 时刻基于里程

的运动估计。该估计被高斯白噪声 $w_t = x_t^{\alpha_n} X_{t+1}^{\alpha_n} - x_t^{\alpha_n} \hat{X}_{t+1}^{\alpha_n}$ 所破坏，该噪声的均值为 0，协方差矩阵为 Q_t 。

除了状态预测方程，线性化的误差状态方程在 EKF 中也是非常必要的。真实状态预测方程可以表示为

$$P_{x_{t+1|t}}^{\alpha_n} = P_{x_t}^{\alpha_n} + C(\phi_{x_t}^{\alpha_n}) x_t^{\alpha_n} P_{x_{t+1}}^{\alpha_n} \quad (17)$$

$$\phi_{x_{t+1|t}}^{\alpha_n} = \phi_{x_t}^{\alpha_n} + x_t^{\alpha_n} \phi_{x_{t+1}}^{\alpha_n} \quad (18)$$

$$P_{L_{t+1|t}} = P_{L_t} \quad (19)$$

将式(17)~(19)和式(14)~(16)相减，得到误差状态估计方程

$$\tilde{P}_{x_{t+1|t}}^{\alpha_n} = \tilde{P}_{x_{t|t}}^{\alpha_n} + C(\phi_{x_t}^{\alpha_n}) x_t^{\alpha_n} P_{x_{t+1}}^{\alpha_n} - C(\hat{\phi}_{x_{t|t}}^{\alpha_n}) x_t^{\alpha_n} \hat{P}_{x_{t+1}}^{\alpha_n} \quad (20)$$

$$\tilde{\phi}_{x_{t+1|t}}^{\alpha_n} = \tilde{\phi}_{x_{t|t}}^{\alpha_n} + x_t^{\alpha_n} \tilde{\phi}_{x_{t+1}}^{\alpha_n} \quad (21)$$

$$\tilde{P}_{L_{t+1|t}} = \tilde{P}_{L_{t|t}} \quad (22)$$

显然，式(20)是非线性的，所以我们在估计状态 $\hat{\phi}_{x_t|t}^{\alpha_n}$ 和 $x_t^{\alpha_n} \hat{P}_{x_{t+1}}^{\alpha_n}$ 将其线性化：

$$\tilde{P}_{x_{t+1}|t}^{\alpha_n} = \tilde{P}_{x_t|t}^{\alpha_n} + JC\left(\hat{\phi}_{x_t|t}^{\alpha_n}\right) x_t^{\alpha_n} \tilde{P}_{x_{t+1}}^{\alpha_n} \tilde{\phi}_{x_t|t}^{\alpha_n} + C\left(\hat{\phi}_{x_t|t}^{\alpha_n}\right) x_t^{\alpha_n} \tilde{P}_{x_{t+1}}^{\alpha_n} \quad (23)$$

其中 $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ 。将误差状态预测方程(21)，(22)，(23)转化为分块矩阵，得到如下所熟悉的方程：

$$\begin{aligned} \tilde{X}'_{t+1|t} &= \begin{bmatrix} \phi_{x_t}^{\alpha_n} & 0_{3 \times 2} \\ 0_{2 \times 3} & I_2 \end{bmatrix} \begin{bmatrix} \tilde{X}'_{t|t} \\ \tilde{P}_{t|t} \end{bmatrix} + \begin{bmatrix} G_{x_t}^{\alpha_n} \\ 0_{2 \times 2} \end{bmatrix} w_t \\ &= \phi_t \tilde{X}'_{t|t} + G_t w_t \end{aligned} \quad (24)$$

其中，状态转移矩阵 $\phi_{x_t}^{\alpha_n}$ 和 $G_{x_t}^{\alpha_n}$ 具体表示为

$$\begin{aligned} \phi_{x_t}^{\alpha_n} &= \begin{bmatrix} I_2 & JC\left(\hat{\phi}_{x_t|t}^{\alpha_n}\right) x_t^{\alpha_n} \hat{P}_{x_{t+1}}^{\alpha_n} \\ 0_{1 \times 2} & 1 \end{bmatrix} \\ &\equiv \begin{bmatrix} I_2 & J\left(\hat{P}_{x_{t+1}|t}^{\alpha_n} - \hat{P}_{x_t|t}^{\alpha_n}\right) \\ 0_{1 \times 2} & 1 \end{bmatrix} \end{aligned} \quad (25)$$

$$G_{x_t}^{\alpha_n} = \begin{bmatrix} C\left(\hat{\phi}_{x_t|t}^{\alpha_n}\right) & 0_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{bmatrix} \quad (26)$$

EKF 中使用的观测值是机器人相对地标位置的一个函数：

$$\begin{aligned} z_t &= h\left(X'_t\right) + v_t \\ &= h\left(C^T\left(\phi_{x_t}^{\alpha_n}\right)\left(P_{L_t} - P_{x_t}^{\alpha_n}\right)\right) + v_t \end{aligned} \quad (27)$$

其中， v_t 为高斯测量噪声，其均值为 0，协方差为 R_t 。这里，我们假设函数 h 为测量函数，且它是可逆的，即通过 z_t ，我们可以得到相对于机器人的地标位置的估计值。一般情况下，观测函数是非线性的，因此在 EKF 中将其线性化。线性化后的观测误差方程为：

$$\begin{aligned} z_t &= \begin{bmatrix} H_{x_t}^{\alpha_n} & H_{L_t} \end{bmatrix} \begin{bmatrix} \tilde{X}'_{t|t-1} \\ \tilde{X}'_{L_t|t-1} \end{bmatrix} + v_t \\ &= H_t \tilde{X}'_{t|t-1} + v_t \end{aligned} \quad (28)$$

式中， $H_{x_t}^{\alpha_n}$ 和 H_{L_t} 分别为函数 $\hat{X}'_{t|t-1}$ 关于机器人位姿和地标位置在状态估计 $\hat{X}'_{t|t-1}$ 处估计的雅可比矩阵。根据链式法则，我们得到

$$H_{x_t}^{\alpha_n} = (\nabla h_t) C^T\left(\hat{\phi}_{x_t|t-1}^{\alpha_n}\right) \cdot \left[-I_2 \quad -J\left(\hat{P}_{L_t|t-1}^{\alpha_n} - \hat{P}_{x_t|t-1}^{\alpha_n}\right)\right] \quad (29)$$

$$H_{L_t} = (\nabla h_t) C^T\left(\hat{\phi}_{x_t|t-1}^{\alpha_n}\right) \quad (30)$$

其中 ∇h_t 表示函数 h 关于相对机器人的地标位置的雅可比矩阵(也就是说，关于 $x_t^{\alpha_n} P_L = C^T\left(\phi_{x_t}^{\alpha_n}\right) \cdot \left(P_L - P_{x_t}^{\alpha_n}\right)$ 在估计值 $\hat{X}'_{t|t-1}$ 的雅可比矩阵)。

3.2.2. EKF-SLAM 的可观测性分析

本文分析了影响一般移动机器人运动一致性的具体原因。黄国权等人[17][18][19]使用滤波器的误差状态系统模型分析移动机器人一致性不足的问题。

该研究发现标准的 EKF-SLAM 模型中存在一个不可观测的二维子空间, 在全局参考系的位置和角度上有三个不可观测的自由度。当系统和观测模型的雅可比矩阵在每个时间步期间的最近的状态估计中, 使用线性误差状态系统进行估计时, 可观测子空间的维数高于实际非线性 SLAM 系统子空间的维数, 在状态空间方向上缺乏可用信息导致 EKF 的协方差估计受到削弱, 这是引起不一致性的主要原因。

由于 EKF-SLAM 的线性化误差状态模型是时变的, 因此可以从局部观测矩阵进行可观测性分析。对于 EKF-SLAM 系统(式(24)和式(28)), t 时刻到 $t+m$ 时刻间隔的局部观测矩阵设为

$$\Lambda = \begin{bmatrix} H_t \\ H_{t+1}\phi_{x_t}^{\alpha_n} \\ \vdots \\ H_{t+m}\phi_{x_{t+m-1}}^{\alpha_n} \cdots \phi_{x_t}^{\alpha_n} \end{bmatrix} \quad (31)$$

这个公式可以进一步推广通过代入 ϕ_i 和 H_i (方程(24)和(28)), 可以得到

$$\Lambda = \begin{bmatrix} H_{x_t}^{\alpha_n} & H_{L_t} \\ H_{x_{t+1}}^{\alpha_n} \phi_{x_t}^{\alpha_n} & H_{L_{t+1}} \\ \vdots & \vdots \\ H_{x_{t+m}}^{\alpha_n} \phi_{x_{t+m-1}}^{\alpha_n} \cdots \phi_{x_t}^{\alpha_n} & H_{L_{t+m}} \end{bmatrix} \quad (32)$$

$$= \text{diag}(H_{L_t}, \dots, H_{L_{t+m}}) \cdot \begin{bmatrix} H_{L_t}^{-1} H_{x_t}^{\alpha_n} & I_2 \\ H_{L_{t+1}}^{-1} H_{x_{t+1}}^{\alpha_n} \phi_{x_t}^{\alpha_n} & I_2 \\ \vdots & \vdots \\ H_{L_{t+m}}^{-1} H_{x_{t+m}}^{\alpha_n} \cdots \phi_{x_{t+m-1}}^{\alpha_n} & I_2 \end{bmatrix} \quad (33)$$

其中 $\text{diag}(\cdot)$ 为块对角化矩阵。在时间间隔 t 和 $t+m$ 之间, 系统是局部可观的, 当且仅当局部可观矩阵 Λ 是满秩的。 $\text{diag}(H_{L_t}, \dots, H_{L_{t+m}})$ 是非奇异的。

3.2.3. 标准 EKF-SLAM 的可观性

在本节中, 我们将讨论标准 EKF-SLAM 的可观性, 其中雅可比矩阵是在估计值处计算的。

$$\begin{aligned} \phi_{x_{t+1}}^{\alpha_n} \phi_{x_t}^{\alpha_n} &= \begin{bmatrix} I_2 & J(\hat{P}_{x_{t+2|t+1}}^{\alpha_n} - \hat{P}_{x_{t+1|t+1}}^{\alpha_n}) \\ 0_{1 \times 2} & 1 \end{bmatrix} \cdot \begin{bmatrix} I_2 & J(\hat{P}_{x_{t+1|t}}^{\alpha_n} - \hat{P}_{x_{t|t}}^{\alpha_n}) \\ 0_{1 \times 2} & 1 \end{bmatrix} \\ &= \begin{bmatrix} I_2 & G \\ 0_{1 \times 2} & 1 \end{bmatrix} \end{aligned} \quad (34)$$

其中 $G = J(\hat{P}_{x_{t+2|t+1}}^{\alpha_n} - \hat{P}_{x_{t|t}}^{\alpha_n} + (\hat{P}_{x_{t+1|t}}^{\alpha_n} - \hat{P}_{x_{t+1|t+1}}^{\alpha_n}))$

通过归纳法, 我们可以得到

$$\phi_{x_{t+i-1}}^{\alpha_n} \phi_{x_{t+i-2}}^{\alpha_n} \cdots \phi_{x_t}^{\alpha_n} = \begin{bmatrix} I_2 & D \\ 0_{1 \times 2} & 1 \end{bmatrix} \quad (35)$$

$D = J(\hat{P}_{x_{t+i|t+i-1}}^{\alpha_n} - \hat{P}_{x_{t|t}}^{\alpha_n} - \sum_{j=t+1}^{t+i-1} (\hat{P}_{x_{j|j}}^{\alpha_n} - \hat{P}_{x_{j|j-1}}^{\alpha_n}))$, 且 $i > 0$ 。结合(28), (29), (30), 得到

$$\begin{aligned} H_{L_{t+i}}^{-1} H_{x_{t+i}}^{\alpha_n} \phi_{x_{t+i-1}}^{\alpha_n} \cdots \phi_{x_t}^{\alpha_n} &= -C^{-T}(\phi_{x_{t+i|t+i-1}}^{\alpha_n})(\nabla h_{t+i})^{-1}(\nabla h_{t+i}) \cdot C^T(\phi_{x_{t+i|t+i-1}}^{\alpha_n}) \begin{bmatrix} I_2 & J(P_{L_{t+i|t+i-1}} - P_{x_{t+i|t+i-1}}^{\alpha_n}) \\ 0_{1 \times 2} & 1 \end{bmatrix} \\ &= -[I_2 \quad F] \end{aligned} \quad (36)$$

其中

$$E = J \left(\hat{P}_{x_{t+i|t+i-1}}^{\alpha_n} - \hat{P}_{x_{t|t}}^{\alpha_n} + \sum_{j=t}^{t+i-2} \left(\hat{P}_{x_{j+1|j}}^{\alpha_n} - \hat{P}_{x_{j+1|j+1}}^{\alpha_n} \right) \right),$$

$$F = J \left(P_{L_{t+i|t+i-1}} - \hat{P}_{x_{t|t}}^{\alpha_n} + \sum_{j=t}^{t+i-2} \left(\hat{P}_{x_{j+1|j}}^{\alpha_n} - \hat{P}_{x_{j+1|j+1}}^{\alpha_n} \right) \right)$$

利用这个结果，我们可以把(33)中的矩阵 Q 写成

$$Q = \begin{bmatrix} -I_2 & A_1 & I_2 \\ -I_2 & A_2 & I_2 \\ -I_2 & A_3 & I_2 \\ \vdots & \vdots & \vdots \\ -I_2 & A_k & I_2 \end{bmatrix} \quad (37)$$

其中

$$A_1 = -J \left(\hat{P}_{L_{t|t-1}} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right)$$

$$A_2 = -J \left(\hat{P}_{L_{t+1|t}} - \hat{P}_{x_{t|t}}^{\alpha_n} \right)$$

$$A_k = -J \left(\hat{P}_{L_{t+i|t+i-1}} - \hat{P}_{x_{t|t}}^{\alpha_n} + \sum_{j=t}^{t+i-2} \left(\hat{P}_{x_{j+1|j}}^{\alpha_n} - \hat{P}_{x_{j+1|j+1}}^{\alpha_n} \right) \right), k = 2, \dots, i$$

我们注意到，对任何给定状态变量的估计在不同的时间通常是不同的，从而我们可以得到下列不等式一般成立：

$$\hat{P}_{L_{t|t-1}} \neq \hat{P}_{L_{t+1|t}} \neq \dots \neq \hat{P}_{L_{t+i|t+i-1}} \quad (38)$$

和

$$\hat{P}_{x_{t+1|t}}^{\alpha_n} - \hat{P}_{x_{t+1|t+1}}^{\alpha_n} \neq 0_{2 \times 1} \quad (39)$$

因此，秩(Q) = 3 的矩阵 Q 的第三列通常是一个具有不规则元素的向量。由此可见，标准 EKF-SLAM 中采用的线性化误差状态模型的可观性与基本非线性系统不同。具体而言，EKF 通过对时间间隔 $[t, t+m]$ 采集的观测值进行处理，得到状态空间的三维信息。然而，测量实际上只提供关于状态空间的二维信息。因此，EKF 在基本非线性 SLAM 系统的不可观测方向上获得了错误的信息，从而导致不一致。

3.2.4. 改进的 EKF-SLAM 算法

通过对矩阵 Q 的深入研究，我们了解到即使在状态真值处没有估计雅可比矩阵，也可以得到具有三维不可观测子空间的 EKF 系统模型。我们的分析表明，误差状态模型在计算 EKF 雅可比矩阵时通过使用每个状态变量的第一个可用估计值可以具有与基本非线性模型相同的可观测性。为了达到这一目的，我们对传统算法做了以下两点修改：首先，状态转移雅可比矩阵 $\phi_{x_t}^{\alpha_n}$ 的计算如式(25)所示。这里，我们采用变式形式：

$$\phi_{x_t}^{\alpha_n} = \begin{bmatrix} I_2 & J \left(\hat{P}_{x_{t+1|t}}^{\alpha_n} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right) \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \quad (40)$$

与式(25)不同的是，这里使用的是机器人位置信息的预测值 $\hat{P}_{x_{t+1|t}}^{\alpha_n}$ ，而不是估计值 $\hat{P}_{x_{t|t}}^{\alpha_n}$ 。其次，对于观测到的雅可比矩阵 $H_{x_t}^{\alpha_n}$ 的计算，我们通常使用第一次观测到的路标的值。因此，如果在时间 t 处第一

次观测到一个路标，我们将计算观测到的机器人位姿的雅可比矩阵如下：

$$H_{x_t}^{\alpha_n} = (\nabla h_t) C^T \left(\hat{\phi}_{x_{t|t-1}} \right) \cdot \begin{bmatrix} -I_2 & -J \left(\hat{P}_{L_{t|t}} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right) \end{bmatrix} \quad (41)$$

其中 $\hat{P}_{L_{t|t}}$ 是路标的第一个观测值。

基于上述的修改，在滤波算法的雅可比矩阵的计算中，只会出现所有地标位置的第一次观测值和机器人姿态的先验估计。它将影响可观测矩阵 Q ，这里将得到一个新的矩阵 Q' ，这种方法被称为第一估计雅可比矩阵 EKF(FEJ-EKF)算法。通过替换传统的 FastSLAM 算法，得到了本文提出的改进算法。

$$Q = \begin{bmatrix} -I_2 & -J \left(\hat{P}_{L_{t|t}} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right) & I_2 \\ -I_2 & -J \left(\hat{P}_{L_{t|t}} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right) & I_2 \\ -I_2 & -J \left(\hat{P}_{L_{t|t}} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right) & I_2 \\ \vdots & \vdots & \vdots \\ -I_2 & -J \left(\hat{P}_{L_{t|t}} - \hat{P}_{x_{t|t-1}}^{\alpha_n} \right) & I_2 \end{bmatrix} \quad (42)$$

我们可以了解到这个矩阵的秩是 2。因此，FEJ-EKF 算法是基于一个不可观测子空间维数为 3 的误差状态模型。我们强调 FEJ-EKF 估计器可以在实际应用中实现，因为它确实不利用任何状态真值。

3.2.5. 计算重要性权重

更新地图后，计算粒子的重要权重

$$\omega_t^{[\alpha_n]} = \frac{p \left(x_{1:t}^{[\alpha_n]} \mid z_{1:t}, u_{1:t}, K_t \right)}{p \left(x_{1:t}^{[\alpha_n]} \mid z_{1:t-1}, u_{1:t}, K_{t-1} \right)} \quad (43)$$

3.2.6. 重采样粒子

最后，在重采样阶段，我们引入了一种采样方法 Kullback-Leibler Divergence (KLD) sampling [20]，它可以随时改变粒子数，以确定采样过程中保持近似质量所需的最小粒子数。KLD 采样的核心思想是：假设从 a 个有效子空间的离散分布中采样 n 个粒子，从每个子空间中采样的粒子数表示为向量 $X = \{x_1, x_2, \dots, x_a\}$ ，子空间对应粒子的真实后验概率为向量 $P = \{P_1, P_2, \dots, P_a\}$ ，然后用 $\hat{P} = \{\hat{P}_1, \hat{P}_2, \dots, \hat{P}_a\} = n^{-1}X$ 得到 P 的最大似然估计，最后得到 P 的似然概率统计值 γ_n 为

$$\lg \gamma_n = \sum_{i=1}^a x_i \lg \left(\frac{\hat{P}_i}{P_i} \right) = n \sum_{i=1}^a \hat{P}_i \lg \left(\frac{\hat{P}_i}{P_i} \right) \quad (44)$$

其中 x_i 是第 i 个子空间中采样的粒子数； P_i 是第 i 个子空间的真实后验概率； \hat{P}_i 是第 i 个子空间的最大似然估计； n 为粒子总数； a 是有效子空间的个数。

定义 1. (KLD [20]) 真实后验分布 P 与最大似然估计 \hat{P} 之间的差值称为 Kullback-Leibler 散度(KLD)，并且表示为

$$\kappa(P, \hat{P}) = \sum_{i=1}^a \hat{P}_i \lg \left(\frac{\hat{P}_i}{P_i} \right) \quad (45)$$

$\kappa(P, \hat{P}) \geq 0$ ，并且当且仅当两个分布相同时， $\kappa(P, \hat{P})$ 等于 0。

由(44)和(45)可得

$$\lg \gamma_n = n\kappa(P, \hat{P}) \quad (46)$$

当粒子总数 n 趋于正无穷大时, P 的似然概率统计量收敛于 $a-1$ 个自由度的卡方分布, 即

$$2\lg \gamma_n = 2n\kappa(P, \hat{P}) \rightarrow \chi_{a-1}^2, \quad n \rightarrow \infty \quad (47)$$

KLD 小于或等于 ε 的概率定义为 $P_p(\kappa(P, \hat{P}) \leq \varepsilon)$, 所以

$$\begin{aligned} P_p(\kappa(P, \hat{P}) \leq \varepsilon) &= P_p(2n\kappa(P, \hat{P}) \leq 2n\varepsilon) \\ &= P_p(\chi_{a-1}^2 \leq 2n\varepsilon) \end{aligned} \quad (48)$$

其中 ε 是假设 P 是真实分布下的最小阈值。 χ_{a-1}^2 的分位数可表示为

$$P_p(\chi_{a-1}^2 \leq \chi_{a-1, 1-\delta}^2) = 1 - \delta \quad (49)$$

其中, $\chi_{a-1, 1-\delta}^2$ 为 χ_{a-1}^2 在标准正态分布上限 $1-\delta$ 下的分布。选择适当的 n , 使 $2n\varepsilon = \chi_{a-1, 1-\delta}^2$, 然后可得

$$P_p(\kappa(P, \hat{P}) \leq \varepsilon) = 1 - \delta \quad (50)$$

为了计算 n 的值, 我们可以使用一种很好的估计方法——Wilson-Hilferty 变换[21]来近似 n 的值, 得到

$$\begin{aligned} n &= \frac{1}{2\varepsilon} \chi_{a-1, 1-\delta}^2 \\ &\doteq \frac{a-1}{2\varepsilon} \left\{ 1 - \frac{2}{q(a-1)} + \sqrt{\frac{2}{q(a-1)}} z_{1-\delta} \right\}^3 \end{aligned} \quad (51)$$

其中 $z_{1-\delta}$ 为标准正态分布的上限 $1-\delta$ 的分位数。

这导致了在 KLD 上具有 ε 上限的近似离散分布所需的样本量的推导。由式(19)可知, n 与误差范围 ε 成反比。

在重采样过程中, 我们首先对粒子标签数据进行融合, 以计算是否添加了新的粒子。如果是, 标签检测模型将重新分配粒子的权重; 如果不是, 则执行低方差重采样。最后, 利用 KLD 采样确定粒子重采样过程中所需的粒子总数, 实现了粒子总数的动态修改, 降低了算法的计算复杂度。然后根据粒子的权重进行采样。

4. FastSLAM 一致性的评估

为了更好地评价估计器的性能, 有必要对估计算法进行一致性检查。如[7]中所述, 假设真值为 X_{g_k} , 那么算法保持一致性必须满足以下条件:

$$E[X_{g_k} - \hat{X}_{g_k}] = 0 \quad (52)$$

$$E\left[(X_{g_k} - \hat{X}_{g_k})(X_{g_k} - \hat{X}_{g_k})^T\right] \leq P_{g_k} \quad (53)$$

一般来说, 为了衡量一个算法是否一致, 人们会将其估计值与从理想贝叶斯滤波中获得的概率密度函数(PDF)进行比较。然而, 这对于 FastSLAM 算法是不实际的。当“真实”PDF 不可用时, 真实状态 x_k 是已知的, 我们可以使用归一化估计误差平方(NEES) [7]来表征滤波器的性能,

$$\varepsilon_k = (x_k - \hat{x}_k)^T P_k^{-1} (x_k - \hat{x}_k) \quad (54)$$

其中 $\{\hat{x}_k, P_k\}$ 是估计的均值和协方差。

通过检查 N 次蒙特卡洛运行的滤波的平均 NEES，可以发现滤波算法的一种一致性的度量方法。在滤波算法是一致的且近似线性高斯的假设下， ε_k 的值为 χ^2 (卡方)，分布自由度为 $\dim(x_k)$ 。当 N 趋于无穷大时， ε_k 的平均值趋于状态维数。

$$E[\varepsilon_k] = \dim(x_k) \tag{56}$$

该假设的有效性可以进行一个 χ^2 可接受性检验。

FastSLAM 的一致性是通过执行多次蒙特卡洛运行和计算平均 NEES 来评估的。给定 N 次运行，平均 NEES 计算为

$$\bar{\varepsilon}_k = \frac{1}{N} \sum_{i=1}^N \varepsilon_{ik} \tag{57}$$

给定一个一致的线性 - 高斯滤波的假设， $N\bar{\varepsilon}_k$ 的密度为 χ^2 ，自由度为 $N \dim\{x_k\}$ 。

5. 仿真结果

在本节中，我们将通过实验来证明新颖的 FastSLAM 算法的有效性。FastSLAM 算法的关键特性是由嵌入在粒子滤波中的蒙特卡罗采样高度说明的。我们使用了一个小的自生成里程数据和一个图形函数[22]。粒子的权重通过新的测量方法不断修正和重估。由于新的测量噪声，特征映射有偏差，粒子权重被外部测量错误地操纵[23]。

在图 3 和图 4 中，黑色虚线表示最佳加权粒子的轨迹，绿色实线表示真实粒子的轨迹。在这个图中有 25 个地标和 100 个粒子，其中红色虚线代表传感器测量线。图 3 为标准 FastSLAM 算法中粒子的运动轨迹，图 4 为理想 FastSLAM 算法中粒子的运动轨迹。

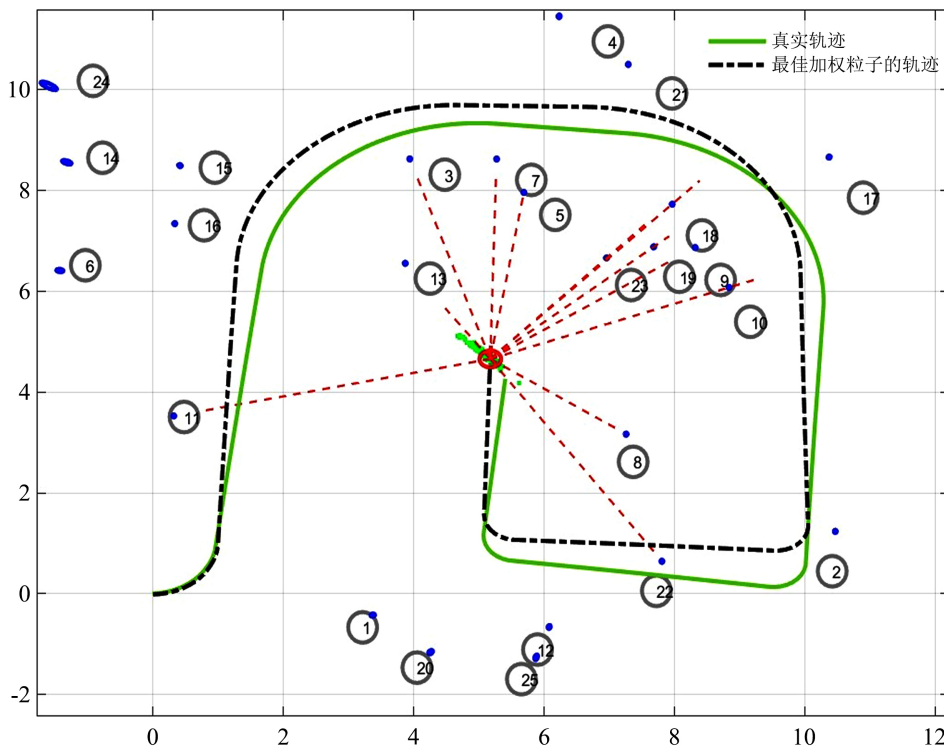


Figure 3. Particle trajectory in standard FastSLAM algorithm

图 3. 标准的 FastSLAM 算法中粒子的运动轨迹

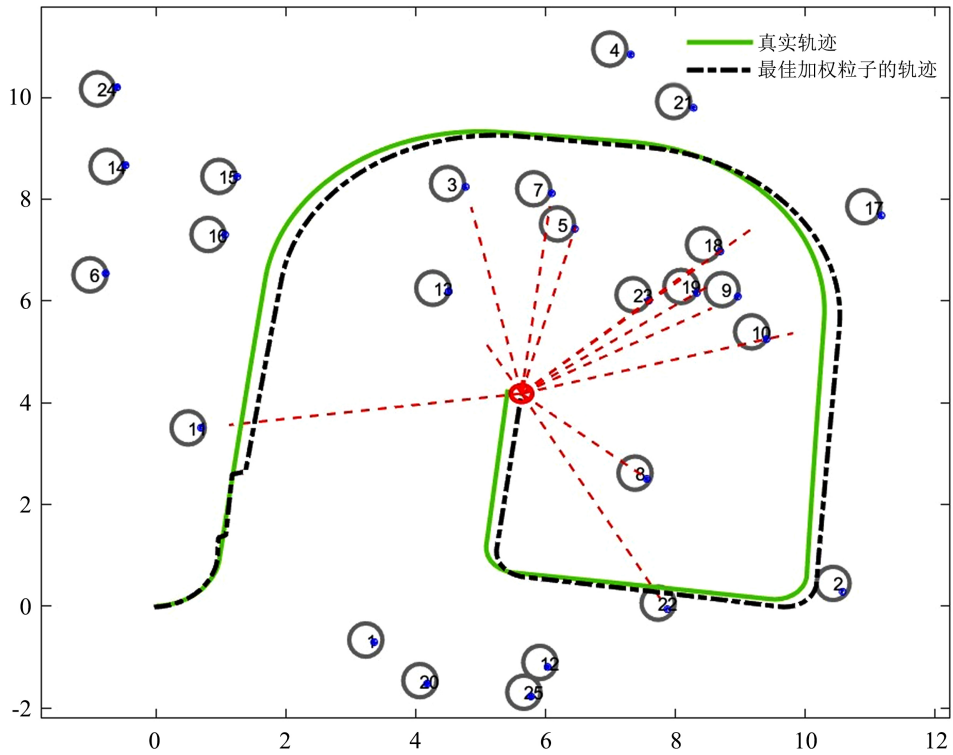


Figure 4. Particle trajectory in the ideal FastSLAM algorithm

图 4. 理想的 FastSLAM 算法中粒子的运动轨迹

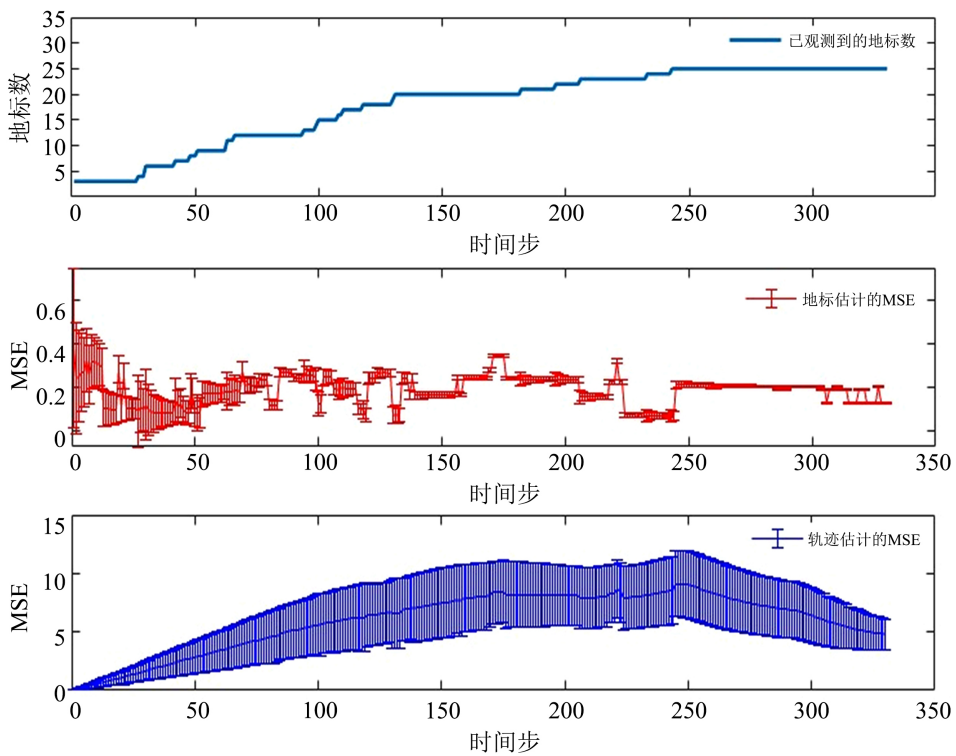


Figure 5. Landmark estimation error of standard FastSLAM algorithm

图 5. 标准的 FastSLAM 算法的地标估计误差

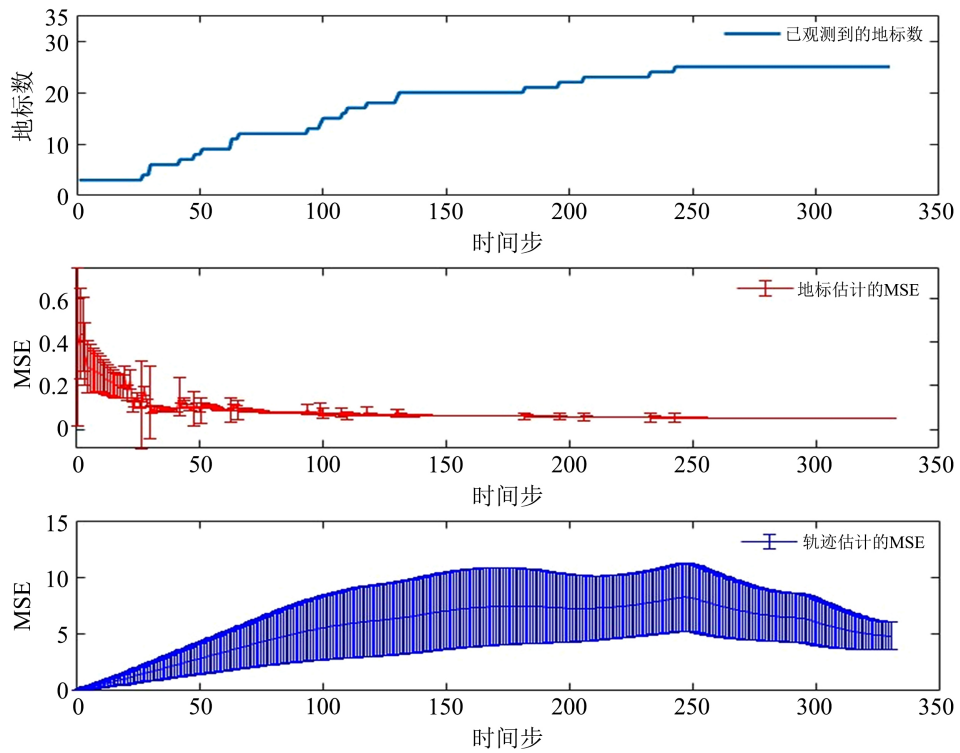


Figure 6. The landmark estimation error of the ideal FastSLAM algorithm

图 6. 理想的 FastSLAM 算法的地标估计误差

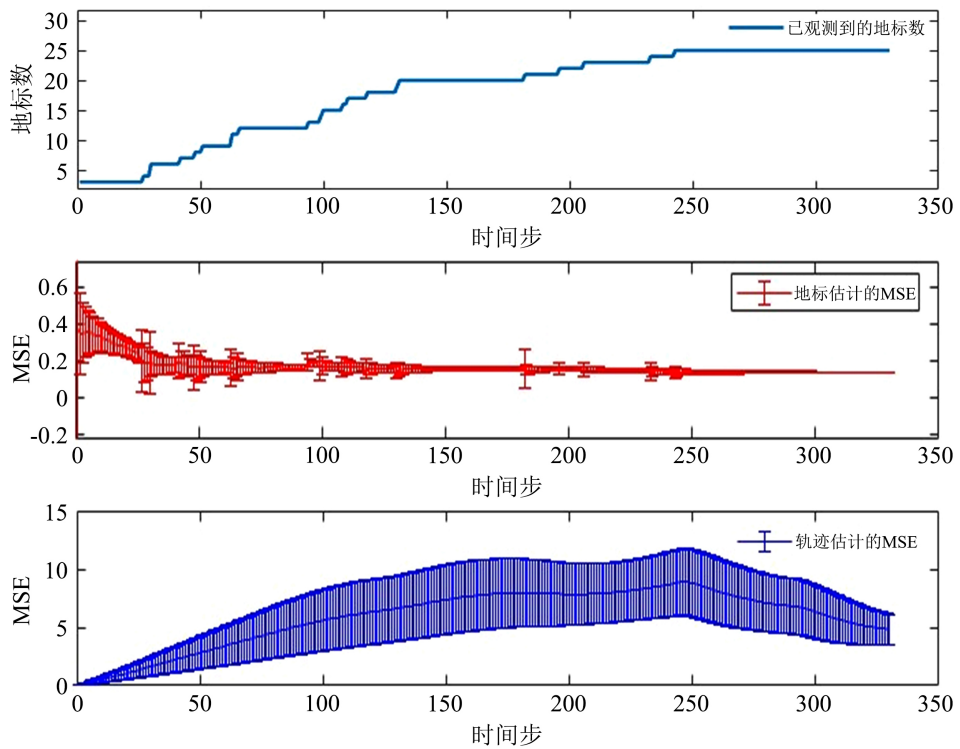


Figure 7. The landmark estimation error of the improved FastSLAM algorithm

图 7. 改进的 FastSLAM 算法的地标估计误差

图 5~7 为地图误差，绘制了观测到的地标及其均方位置误差。一开始，误差幅度是相当大的。红色的误差条表示误差值上下变化。红色误差条越长，不确定性越大。通过 FastSLAM 中红色误差条的表示，我们可以观察到在很长一段时间内存在相当大的不确定性。图 5 显示了标准 FastSLAM 算法的误差，我们可以观察到在很长一段时间内存在不确定性。图 6 为理想的 FastSLAM 算法的误差。本文提出的 FastSLAM 算法的误差如图 7 所示。我们可以观察到，地标估计的均方误差逐渐收敛到 0，这意味着不确定性逐渐变小，最终趋于稳定。这很好地说明了我们提出的算法的有效性。对于机器人的轨迹误差，它是相当稳定的，并很好地限制在小于 15 m。

图 8 中的 NEES 结果提供了更正式的一致性度量，图 8 中的三条曲线表示的机器人位姿的 NEES 随时间变化的曲线，在这些图中，青色代表标准的 FastSLAM 算法关于机器人位姿的 NEES 结果，紫色代表理想的 FastSLAM 算法关于机器人位姿的 NEES 结果，红色代表改进的 FastSLAM 算法关于机器人位姿的 NEES 结果。这些结果是基于机器人的位姿进行计算的，在每种情况下，都可以看到滤波器变得迅速地乐观。总的趋势是，改进的 FastSLAM 算法的关于机器人的位姿的 NEES 表现较好，所得到的机器人的位姿估计和地图的一致性得到提高。

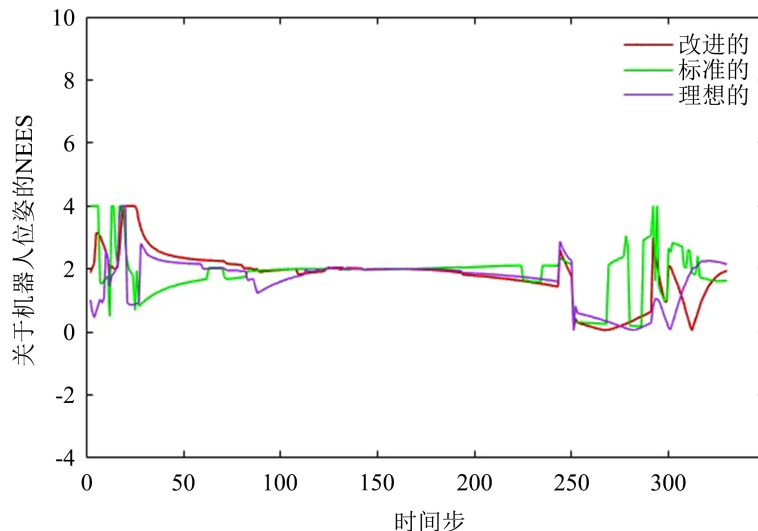


Figure 8. NEES on robot pose for three FastSLAM algorithms

图 8. 三种 FastSLAM 算法的关于机器人位姿的 NEES

6. 结论

在这篇文章中，我们对 FastSLAM 算法进行了改进。与传统算法相比，改进算法的优点有三个方面。首先，采用目标标记法，清晰地标记单个粒子的状态，从而对粒子权重进行修正，提高了算法的实时性和精度。其次，基于可观测性条件，在地标估计阶段融入 FEJ 思想，增强地标估计的一致性，构建满足一致性的地图。第三，在重采样阶段，引入了 KLD 采样方法，该方法可以随时改变采样过程中的粒子数，从而确定采样过程中保持近似质量所需的最小粒子数，动态改变总粒子数，减少算法的计算量。在实际应用中，这将可能更进一步改善机器人同步定位与建图过程中的定位精确度和所构建地图的一致性，同时在此过程中，算法的计算量也会被降低。

参考文献

- [1] Davison, A.J., Reid, I.D., Molton, N.D., *et al.* (2007) Monoslam: Real-Time Single Camera Slam. *IEEE Transactions*

- on *Pattern Analysis and Machine Intelligence*, **29**, 1052-1067. <https://doi.org/10.1109/TPAMI.2007.1049>
- [2] 刘俊恺. 基于 ROS 平台的 FastSLAM 算法改进与应用研究[D]: [硕士学位论文]. 南昌: 华东交通大学, 2019.
- [3] Martinez-Cantin, R. and Castellanos, J.A. (2005) Unscented Slam for Large-Scale Outdoor Environments. *The IEEE/RSJ International Conference on Intelligent Robots & Systems*, Edmonton, AB, 2-6 August 2005, 3427-3432. <https://doi.org/10.1109/IROS.2005.1545002>
- [4] Montemerlo, M., Thrun, S., Koller, D., *et al.* (2002) FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *The 18th National Conference on Artificial Intelligence (AAAI)*, California, 28 July-1 August 2002, 593-598.
- [5] 赵挽东. 复杂场景下机器人 SLAM 算法研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2019.
- [6] Julier, S.J. and Uhlmann, J.K. (2001) A Counter Example to the Theory of Simultaneous Localization and Map Building. *The 2001 ICRA, IEEE International Conference on Robotics and Automation*, Seoul, 21-26 May 2001, 4238-4243.
- [7] Bar-Shalom, Y., Kirubarajan, T. and Li, X.R. (2001) Estimation with Applications to Tracking and Navigation. John Wiley & Sons, New York. <https://doi.org/10.1002/0471221279>
- [8] Castellanos, J.A., Neira, J. and Tardós, J.D. (2004) Limits to the Consistency of EKF-Based Slam. *IFAC Proceedings Volumes*, **37**, 716-721. [https://doi.org/10.1016/S1474-6670\(17\)32063-3](https://doi.org/10.1016/S1474-6670(17)32063-3)
- [9] Huang, G. (2017) Towards Consistent Filtering for Discrete-Time Partially-Observable Nonlinear Systems. *Systems & Control Letters*, **106**, 87-95. <https://doi.org/10.1016/j.sysconle.2017.06.006>
- [10] 姜晓燕. 基于粒子滤波和一致性分析的同时定位与地图构建算法研究[D]: [硕士学位论文]. 青岛: 中国海洋大学, 2014.
- [11] Murphy, K.P. (1999) Bayesian Map Learning in Dynamic Environments. *Advances in Neural Information Processing Systems*, **12**, 1015-1021.
- [12] Montemerlo, M., Thrun, S., Koller, D., *et al.* (2003) FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping That Provably Converges. *The IJCAI*, **3**, 1151-1156.
- [13] He, B., Liang, Y., Feng, X., *et al.* (2012) Auv Slam and Experiments Using a Mechanical Scanning Forward-Looking Sonar. *Sensors*, **12**, 9386-9410. <https://doi.org/10.3390/s120709386>
- [14] Zhang, L., Meng, X.J. and Chen, Y.W. (2009) Convergence and Consistency Analysis for FastSLAM. *The 2009 IEEE Intelligent Vehicles Symposium*, Xi'an, 3-5 June 2009, 447-452.
- [15] Vo, B.T. and Vo, B.N. (2013) Labeled Random Finite Sets and Multi-Object Conjugate Priors. *IEEE Transactions on Signal Processing*, **61**, 3460-3475. <https://doi.org/10.1109/TSP.2013.2259822>
- [16] Vo, B.N., Vo, B.T. and Phung, D. (2014) Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter. *IEEE Transactions on Signal Processing*, **62**, 6554-6567. <https://doi.org/10.1109/TSP.2014.2364014>
- [17] Huang, G.P., Mourikis, A.I. and Roumeliotis, S.I. (2008) Analysis and Improvement of the Consistency of Extended Kalman Filter-Based Slam. *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, 19-23 May 2008, 473-479. <https://doi.org/10.1109/ROBOT.2008.4543252>
- [18] Huang, G., Mourikis, A.I. and Roumeliotis, S.I. (2010) Observability-Based Rules for Designing Consistent EKF Slam Estimators. *International Journal of Robotics Research*, **29**, 502-528. <https://doi.org/10.1177/0278364909353640>
- [19] Huang, G., Mourikis, A.I. and Roumeliotis, S.I. (2009) A First-Estimates Jacobian EKF for Improving Slam Consistency. In: Khatib, O., Kumar, V. and Pappas, G.J., Eds., *Experimental Robotics*, Vol. 54, Springer, Berlin Heidelberg, 373-382. https://doi.org/10.1007/978-3-642-00196-3_43
- [20] Fox, D. (2003) Adapting the Sample Size in Particle Filters through Kld-Sampling. *The International Journal of Robotics Research*, **22**, 985-1003. <https://doi.org/10.1177/0278364903022012001>
- [21] Johnson, N.L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions. 2nd Edition, Vol. 289, John Wiley & Sons, New York.
- [22] SLAM-Algorithms-Octave. <https://github.com/kiran-mohan/SLAM-Algorithms-Octave.git>
- [23] FastSLAM. <https://github.com/XiaoLiSean/fastSLAM.git>