

基于自适应渐消 UKF 的 FastSLAM 算法



王秉洲 王慧斌 沈洁 张丽丽

河海大学计算机与信息学院 南京 210098

(465716083@qq.com)

摘要 同时定位与地图构建(Simultaneous Localization And Mapping, SLAM)是未知环境下实现机器人自主导航的主要方法, FastSLAM 是一个著名的 SLAM 问题解决方法。由于 FastSLAM 使用序贯重要性采样的方法, 随着算法迭代计算, 大部分粒子的权重值变得很小, 只有很少粒子具有较大的权重, 算法发生退化。为了使采样的粒子分布更加精确, 避免粒子出现退化情况, 从而进一步提高 FastSLAM 算法的估计精度, 提出了一种基于自适应渐消无迹卡尔曼滤波(AFUKF)的快速同步定位和地图创建(FastSLAM)算法。针对 FastSLAM 的粒子退化问题, 从研究粒子的建议分布函数出发, 采用渐消无迹卡尔曼滤波(Adaptive Fading Unscented Kalman Filter, AFUKF)代替扩展卡尔曼滤波器(Extended Kalman Filter, EKF)来估计机器人位姿的建议分布函数, 避免了 EKF 的线性化误差。同时, 利用自适应渐消滤波思想产生一种参数可自适应调节的建议分布函数, 使其更接近移动机器人的后验位姿概率分布, 减缓粒子集的退化。在 MATLAB 平台上的仿真实验结果表明, 所提方法的位置估计均方误差比标准 FastSLAM 降低了 28.7%, 即估计精度提升了 28.7%。在与近几年相关算法的对比实验中, 所提方法也取得了较高的估计精度。改变粒子数量条件进行实验时, 随着粒子数量的增加, 各算法的估计精度都在提升, 所提算法依然取得了最好的估计精度。实验结果充分说明, 提出的算法计算建议分布函数更加精确, 有效缓解了 FastSLAM 算法中的粒子退化问题, 从而显著提高了算法的估计精度。

关键词: 同步定位与地图构建; 机器人; 自适应渐消无迹卡尔曼滤波; 粒子退化; 建议分布函数

中图分类号 TP242.6

FastSLAM Algorithm Based on Adaptive Fading Unscented Kalman Filter

WANG Bing-zhou, WANG Hui-bin, SHEN Jie and ZHANG Li-li

School of Computer and Information, Hohai University, Nanjing 210098, China

Abstract Simultaneous localization and mapping(SLAM) is the main method to realize autonomous navigation of robots in unknown environments and FastSLAM algorithm is a popular solution to SLAM problem. Due to the sequential importance sampling method used in FastSLAM, a few of particles have a larger weight while the weight of most particles becomes very small throughout the iterative process, which leads to particle degradation. In order to make the particle distribution more accurate and reduce the particle degradation, a FastSLAM algorithm based on adaptive fading unscented Kalman filter (AFUKF) is proposed to improve the estimation accuracy of FastSLAM algorithm. To overcome the problem of particle degradation in FastSLAM, starting from the study of particle's proposal distribution function, this paper uses adaptive fading unscented Kalman filter (AFUKF) instead of EKF to estimate the proposed distribution function of robot's position to avoid the linearization error of EKF. With using the idea of adaptive fading filter, the proposal distribution is closer to the posterior position of the mobile robot and the particle set degradation is relieved. The simulation results on MATLAB platform show that the mean square error of position estimation of the proposed method is 28.7% lower than that of standard FastSLAM, i. e. the estimation accuracy is improved by 28.7%. And the proposed method achieves high estimation accuracy compared with the related algorithms in recent years. When increasing the increase of the number of particles, the estimation accuracy of each algorithm is improved, and the proposed algorithm still achieves the highest estimation accuracy. The experimental results fully show that the proposed algorithm can calculate the proposed distribution function more accurately and effectively alleviate the particle degradation problem in FastSLAM algorithm, which significantly improve the estimation accuracy of FastSLAM algorithm.

Keywords Simultaneous localization and mapping, Robot, Adaptive fading unscented kalman filter, Particle degradation, Proposal distribution function

收稿日期:2019-07-25 返修日期:2019-10-28 本文已加入开放科学计划(OSID), 请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(51709083)

This work was supported by the National Natural Science Foundation of China(51709083).

通信作者:王慧斌(hbwang@hhu.edu.cn)

1 引言

同时定位与地图构建(SLAM)指在未知环境下,机器人根据传感器提供的周围环境信息和位置估计,逐步确定自身位姿并且建立增量式地图的过程^[1-2]。SLAM不需要预先探明地图,被认为是实现移动机器人在未知环境中自主导航的主要技术途径,受到了越来越多的关注^[3-6]。目前,同时定位与地图构建(SLAM)方法已经在室内机器人定位^[7]、无人驾驶中的汽车定位、空中无人机自主飞行^[8]以及水下无人飞行器自主导航^[9]等多方面得到应用。

基本SLAM方法是基于扩展卡尔曼滤波(EKF-SLAM)实现的,其通过扩展卡尔曼滤波迭代估计移动机器人位姿状态和地图路标信息的联合后验分布。但是,EKF-SLAM算法的计算量会随着地图的增量建立而急剧增大;在非线性的函数泰勒展开式的高阶项无法忽略时,其计算结果会出现较大误差^[10-11]。针对这些问题,Montemarlo^[12]提出了FastSLAM1.0算法,并在之后的工作中继续改进,提出了FastSLAM2.0算法。Fast-SLAM基于Rao-Blackwelled粒子滤波的思想,将联合SLAM空间状态估计分解为采样和解析两部分,通过粒子滤波估计机器人的位姿,并使用EKF递归地维护地图,算法精度相比EKF-SLAM有较大提高^[13-14]。由于FastSLAM1.0直接使用过程函数作为建议分布函数,而没有融合最新的观测信息来得到后验概率分布函数,使得算法中的粒子退化问题十分严重。FastSLAM2.0使用EKF算法递归估计机器人的位姿状态,得到的建议分布函数包含了最新的机器人观测信息,能够缓解算法中的粒子退化问题。另外, Kim等^[15]提出利用无迹卡尔曼滤波(Unscented Kalman Filter, UKF)来替代扩展卡尔曼滤波获得粒子的后验位姿建议分布函数,以减小EKF模型线性化产生的误差,提高状态估计精度。

近年来,有学者^[16]提出了使用自适应渐消扩展卡尔曼滤波(Adaptive Fading Extended Kalman Filter, AFEKF)来计算建议分布函数,从而得到了更精确的建议分布函数;但是,又引入了EKF的模型线性化误差。因此,分析如何获得更高精度的建议分布函数,来降低粒子退化程度,提高FastSLAM算法的估计精度,是本文工作的研究重点。

为了降低粒子退化程度,本文提出在FastSLAM算法中采用自适应渐消无迹卡尔曼滤波来计算粒子的建议分布函数,使粒子更好地近似机器人位姿估计的后验概率密度,缓解了粒子集的退化程度,从而使算法的估计结果更加精确。因此,本文提出了基于自适应渐消UKF的FastSLAM算法,即在FastSLAM算法递归估计过程中,使用AFUKF来产生粒子重要性采样函数,用粒子滤波估计移动机器人的位姿,用AFUKF来完成地图的创建和更新。

2 算法思想

2.1 FastSLAM算法

FastSLAM算法的基本思想是^[12]:将机器人系统状态估计分解为机器人位姿估计和地图估计两部分。

$$p(x_{1:t}, M | z_{1:t}, u_{1:t}) = p(x_{1:t} | z_{1:t}, u_{1:t}) \prod_{n=1}^{N_m} p(m_n | x_{1:t}, z_{1:t}, u_{1:t}) \quad (1)$$

其中, $x_{1:t}$, $z_{1:t}$ 和 $u_{1:t}$ 分别是开始到当前时刻 t 的机器人位姿、观测值和控制值; $M = \{m_1, \dots, m_{N_m}\}$ 是全局地图中各地图点的信息集合。

FastSLAM算法采用粒子滤波来估计机器人的运动轨迹,在每个粒子中使用一个扩展卡尔曼滤波来维护机器人位置的后验信息。每个粒子中包含一份地图信息,同时使用 s 个独立的扩展卡尔曼滤波来估计地图中 s 个地图点的位置信息。因此, t 时刻第 k 个粒子的状态为:

$$\mathbf{X}_t^k = [xv_t^k, \mu_1^k, \Sigma_1^k, \dots, \mu_s^k, \Sigma_s^k]^T = \begin{bmatrix} xv_t^k \\ M \end{bmatrix} \quad (2)$$

其中, xv 为机器人的位姿, s 为该粒子维护的地图中的地图点数量, μ 和 Σ 是观测到的地图点的位置估计均值和协方差矩阵。

2.2 本文算法

本文算法基于FastSLAM算法框架,根据机器人 $t-1$ 时刻的位姿和 t 时刻的机器人控制值来获得 t 时刻机器人的位姿预测。若 t 时刻机器人获取了周围环境的观测信息,则利用观测信息和地图信息完成机器人位姿更新、粒子权重计算以及地图更新。为了缓解粒子退化问题,算法最后对粒子集进行重采样。本文算法的框架如图1所示。

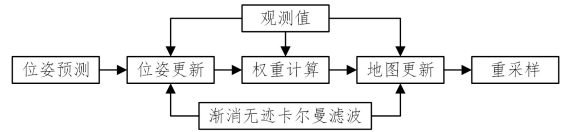


图1 本文算法的框架

Fig. 1 Framework of proposed algorithm

FastSLAM使用扩展卡尔曼滤波计算建议分布函数时,非线性函数的线性逼近会降低算法的估计精度。为了获得更精确的机器人建议分布,本文将渐消滤波的思想和无迹卡尔曼滤波引入FastSLAM算法框架,使用无迹卡尔曼滤波实现机器人位姿预测,使用自适应渐消无迹卡尔曼滤波实现机器人位姿更新和地图更新。

由于无迹卡尔曼滤波没有非线性函数的线性逼近误差,其机器人位姿估计和地图估计结果更加精确;同时,在渐消滤波的渐消因子 λ 自适应调节下,充分利用了当前观测值的有效信息,增强了滤波器的估计能力,使系统状态估计迅速收敛到真实值附近,使滤波性能达到最佳。

3 基于自适应渐消UKF的FastSLAM

3.1 适应渐消无迹卡尔曼滤波

以非线性随机系统为例:

$$x_t = f(x_{t-1}, u_t) + w_{t-1} \quad (3)$$

$$\hat{z}_t = h(x_t) + u_t$$

其中, x_t , \hat{z}_t 分别表示 t 时刻的系统状态值和观测值, $f(\cdot)$ 为系统非线性运动函数, $h(\cdot)$ 为系统非线性观测函数, $w_{t-1} \sim N(0, Q)$ 为过程噪声, $u_t \sim N(0, R)$ 为观测噪声。

根据向量 x_{t-1} 及其协方差矩阵 P_{t-1}^+ 构造 $2n+1$ 个采样点, n 为 x_{t-1} 向量的维数,构建的 sigma 点集为:

$$\begin{aligned} X_{t-1}^0 &= \hat{x}_{t-1}^+ \\ X_{t-1}^i &= \hat{x}_{t-1}^+ + (\sqrt{(n+\lambda)P_{t-1}^+})_i, i=1, \dots, n \\ X_{t-1}^{i+n} &= \hat{x}_{t-1}^+ - (\sqrt{(n+\lambda)P_{t-1}^+})_{i-n}, i=n+1, \dots, 2n \end{aligned} \quad (4)$$

对应权值 w_m^i 和 w_c^i 的计算公式为:

$$w_m^0 = \lambda / (n + \lambda), w_c^0 = \lambda / (n + \lambda) + (1 - \alpha^2 + \beta) \quad (5)$$

$$w_m^i = w_c^i = 1/2(n + \lambda), i = 1, \dots, 2n$$

(1) 状态预测

根据上述 $2n + 1$ 个 sigma 点, 预测值 X_t^i 的计算式为:

$$X_t^i = f(X_{t-1}^i), i = 0, \dots, 2n \quad (6)$$

t 时刻先验状态估计 \hat{x}_t^- 和协方差矩阵 \mathbf{P}_t^- 的计算式为:

$$\hat{x}_t^- = \sum_{i=0}^{2n} w_m^i X_t^i \quad (7)$$

$$\mathbf{P}_t^- = \sum_{i=0}^{2n} w_c^i (X_t^i - \hat{x}_t^-)(X_t^i - \hat{x}_t^-)^T$$

(2) 观测更新

将通过系统方程的 sigma 点集代入观测方程, 得到观测方程相关参数, 具体包括观测值 Z_t^i 及其均值 \hat{z}_t^- 、协方差矩阵 \mathbf{P}_{yy} , 预测值与观测值互相关矩阵 \mathbf{P}_{xy} 。

$$Z_t^i = h(X_t^i), i = 0, \dots, 2n$$

$$\hat{z}_t^- = \sum_{i=0}^{2n} w_m^i Z_t^i \quad (8)$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} w_c^i (X_t^i - \hat{x}_t^-)(Z_t^i - \hat{z}_t^-)^T$$

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} w_c^i (Z_t^i - \hat{z}_t^-)(Z_t^i - \hat{z}_t^-)^T$$

(3) 卡尔曼增益计算

在卡尔曼滤波中, 观测值 z_t 在 t 时刻的新息序列和协方差矩阵为^[17]:

$$\epsilon_t = z_t - h(x_t), \mathbf{V}_t = \mathbf{E}[\epsilon_t \epsilon_t^T] \quad (9)$$

根据最优卡尔曼滤波的思想, 若所求增益阵 \mathbf{K}_t 是最优增益阵, 则在不同时刻的输出新息序列要互不相关, 即:

$$\mathbf{E}[\epsilon_{t+j} \epsilon_t^T] \approx \mathbf{P}_{xy} - \mathbf{K}_t \mathbf{V}_t = 0 \quad (10)$$

传统的 UKF 成为 AFUKF 的条件是在增益阵 \mathbf{K}_t 的求解过程中引入渐消因子 λ , 通过渐消因子的自适应调节, 强制输出的新息序列互不相关^[18]。

渐消因子 λ 的计算公式为:

$$N_t = \mathbf{V}_t - \mathbf{R} - [\mathbf{P}_{xy}]^T [\mathbf{P}_{yy}]^{-1} \mathbf{Q} [\mathbf{P}_{xy}]^{-1} \mathbf{P}_{xy} \quad (11)$$

$$M_t = \mathbf{P}_{yy} - \mathbf{V}_t + N_t$$

$$\lambda = \max\left(1, \frac{\text{tr}(N_t)}{\text{tr}(M_t)}\right)$$

其中, $\text{tr}(\cdot)$ 为求矩阵迹的算子, \mathbf{V}_t 表示 t 时刻实际输出的协方差矩阵, 可由式(12)估算:

$$\mathbf{V}_t = \begin{cases} \epsilon_t \epsilon_t^T, & t = 1 \\ \frac{\rho \mathbf{V}_{t-1} + \epsilon_t \epsilon_t^T}{1 + \rho}, & t \geq 2 \end{cases} \quad (12)$$

$$\epsilon_t = z_t - \hat{z}_t^-$$

其中, ϵ_t 表示 t 时刻的新息, z_t 表示 t 时刻对环境点的真实观测值。

计算引入渐消因子 λ 后的 sigma 点集, 使用 λ 对 sigma 点集进行更新:

$$\begin{aligned} \zeta_{t-1}^0 &= X_{t-1}^0 \\ \zeta_{t-1}^i &= (1 - \sqrt{\lambda_t}) X_{t-1}^i + \sqrt{\lambda_t} X_{t-1}^0 \\ i &= 1, 2, \dots, 2n \end{aligned} \quad (13)$$

通过更新的 sigma 点集计算更新后的观测值, 并引入渐消因子, 即:

$$Z_t^i = h(\zeta_t^i), i = 0, \dots, 2n$$

$$\hat{z}_t^- = \sum_{i=0}^{2n} w_m^i Z_t^i$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} w_c^i (\zeta_t^i - \hat{x}_t^-)(Z_t^i - \hat{z}_t^-)^T \quad (14)$$

$$\mathbf{P}_{yy} = \lambda_t \sum_{i=0}^{2n} w_c^i (Z_t^i - \hat{z}_t^-)(Z_t^i - \hat{z}_t^-)^T$$

$$\mathbf{K}_t = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}$$

(4) 状态更新

$$\hat{x}_t^+ = \hat{x}_t^- + \mathbf{K}_t (z_t - \hat{z}_t^-) \quad (15)$$

$$\mathbf{P}_t^+ = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{P}_{yy} \mathbf{K}_t^T$$

引入渐消滤波的 UKF 与通常的 UKF 相比, 可更大幅度地提取新息中的有效信息。从过程上理解, 当计算结果出现较大误差时, 信息也会随之增大, 通过渐消因子的自适应调节, 增强滤波器的估计能力, 使系统状态估计迅速收敛到真实值附近, 从而使滤波性能达到最佳。

3.2 机器人位姿估计

本文算法使用 AFUKF 产生粒子的建议分布函数, 并使用加权粒子估计机器人的位姿。位姿估计包括位姿预测和位姿更新两个过程。

(1) 位姿预测

首先, 初始化第 k 个粒子中机器人的状态向量, 将机器人位姿、控制值和观测值整合为机器人状态增广向量, 增广后的机器人状态向量及其协方差矩阵为:

$$\mathbf{X}_t^{[k]} = \begin{bmatrix} \mathbf{X}_t^{[k]} \\ 0 \\ 0 \end{bmatrix}, \mathbf{P}_t^{[k]} = \begin{bmatrix} \mathbf{P}_t^{[k]} & 0 & 0 \\ 0 & \mathbf{Q}_t & 0 \\ 0 & 0 & \mathbf{R}_t \end{bmatrix} \quad (16)$$

其中, $\mathbf{X}_t^{[k]}$ 为 $t-1$ 时刻第 k 个粒子状态增广后的机器人状态向量, 增广后的状态向量维数为 L ; $\mathbf{P}_t^{[k]}$ 是其对应的协方差矩阵, \mathbf{Q}_t 和 \mathbf{R}_t 分别是系统中过程噪声和观测噪声的协方差矩阵。

根据式(4)、式(5)生成 sigma 点集 $X_t^{[j][k]}$, 并计算对应权值 $w_m^{[j]}$ 和 $w_c^{[j]}$ 。

计算通过机器人运动方程后的机器人位姿均值和协方差为:

$$\begin{aligned} \bar{X}_t^{[j][k]} &= f(u_{t-1}^j, X_{t-1}^{[j][k]}), i = 0, \dots, 2L \\ \bar{x}_{t|t-1}^{[k]} &= \sum_{i=0}^{2L} w_m^{[i]} \bar{X}_t^{[i][k]} \end{aligned} \quad (17)$$

$$\mathbf{P}_{t|t-1}^{[k]} = \sum_{i=0}^{2L} w_c^{[i]} (\bar{X}_t^{[i][k]} - \bar{x}_{t|t-1}^{[k]})(\bar{X}_t^{[i][k]} - \bar{x}_{t|t-1}^{[k]})^T$$

(2) 位姿更新

当 t 时刻的环境观测信息可以与已建立的地图中的地图点匹配时, 即索引标号为 j 的环境点被机器人重新观测到时, 就可以利用已建立的地图信息和观测信息获取更精确的机器人位姿。

根据式(8)计算观测的预测值 $\hat{n}_{t,j}$ 、观测值协方差 $S_{t,j}^{[k]}$ 、观测值与位姿预测值的互相关协方差 $\Sigma_{t,j}^{x,n[k]}$ 。

此时, 根据渐消自适应 UKF 思想, 计算渐消因子 λ_t 。

$$\begin{aligned} N_t &= \mathbf{V}_t - \mathbf{R}_t - [\Sigma_{t,j}^{x,n[k]}]^T [\mathbf{S}_{t,j}^{[k]}]^{-1} \mathbf{Q}_t [\mathbf{S}_{t,j}^{[k]}]^{-1} \Sigma_{t,j}^{x,n[k]} \\ M_t &= S_{t,j}^{[k]} - \mathbf{V}_t + N_t \end{aligned} \quad (18)$$

$$\lambda_t = \max\left(1, \frac{\text{tr}(N_t)}{\text{tr}(M_t)}\right)$$

其中, \mathbf{R}_t 表示 t 时刻的观测噪声协方差矩阵; \mathbf{Q}_t 表示 t 时刻的系统噪声协方差矩阵; \mathbf{V}_t 表示 t 时刻实际输出的协方差矩阵, 可由式(12)估算。

计算引入渐消因子 λ_t 后的 sigma 点集, 利用 λ_t 对 sigma 点集进行更新:

$$\begin{aligned}\bar{\zeta}_t^{[0][k]} &= \bar{X}_t^{[0][k]} \\ \bar{\zeta}_t^{[i][k]} &= (1 - \sqrt{\lambda_t}) \bar{X}_{t-1}^{[i][k]} + \sqrt{\lambda_t} \bar{X}_t^{[0][k]} \\ i &= 1, 2, \dots, 2L\end{aligned}\quad (19)$$

通过更新的 sigma 点集计算更新后的观测值, 并在 $S_{t,j}^{[k]}$ 中引入渐消因子, 即:

$$\begin{aligned}\bar{N}_{t,j}^{[i][k]} &= h(\bar{\zeta}_t^{[i][k]}, \mu_{t,i-1}^{[k]}) \\ \hat{n}_{t,j}^{[k]} &= \sum_{i=0}^{2L} w_m^{[i]} \bar{N}_{t,j}^{[i][k]} \\ S_{t,j}^{[k]} &= \lambda_t \sum_{i=0}^{2L} w_c^{[i]} (\bar{N}_{t,j}^{[i][k]} - \hat{n}_{t,j}^{[k]})(\bar{N}_{t,j}^{[i][k]} - \hat{n}_{t,j}^{[k]})^T \\ \Sigma_{t,j}^{s,n[k]} &= \sum_{i=0}^{2L} w_c^{[i]} (\bar{\zeta}_t^{[i][k]} - \bar{x}_{t,i-1}^{[k]})(\bar{N}_{t,j}^{[i][k]} - \hat{n}_{t,j}^{[k]})^T\end{aligned}\quad (20)$$

最后计算卡尔曼增益 $K_t^{[k]}$, 并进一步得到 t 时刻机器人的位姿和协方差矩阵的估计值:

$$\begin{aligned}K_t^{[k]} &= \Sigma_{t,j}^{s,n[k]} (S_{t,j}^{[k]})^{-1} \\ \bar{x}_t^{[k]} &= \bar{x}_{t,i-1}^{[k]} + K_t^{[k]} (z_t - \hat{n}_t^{[k]}) \\ P_t^{[k]} &= P_{t,i-1}^{[k]} - K_t^{[k]} S_{t,j}^{[k]} (K_t^{[k]})^T\end{aligned}\quad (21)$$

在 t 时刻, 如果机器人同时观测到了已建立地图中的多个环境点, 则需要依次根据每一个环境点的观测值和地图信息对移动机器人的位姿均值及其协方差矩阵进行更新, 每次更新都将上一次的更新结果作为初始值。

3.3 环境地图估计

算法通过 AFUKF 对环境点进行更新。在此阶段, 需要区分首次观测到的环境点和重复观测到的环境点, 并分别采用不同的方法进行处理。

(1) 环境地图更新

对 t 时刻被重复观测的环境点 j 实行环境地图更新, 根据 $t-1$ 时刻的环境点 j 的估计均值 $\mu_{t,j-1}^{[k]}$ 及其协方差矩阵 $\Sigma_{t,j-1}^{[k]}$ 构建 sigma 点 $\chi^{[i][k]}$, $\mu_{t,j-1}^{[k]}$ 为 N 维向量。

观测值的预测值和卡尔曼增益的计算式为:

$$\begin{aligned}\bar{Z}_t^{[i][k]} &= h(\chi^{[i][k]}, x_t^k), i = 0, 1, \dots, 2N \\ \hat{z}_t^{[k]} &= \sum_{i=0}^{2N} w_m^{[i]} \bar{Z}_t^{[i][k]} \\ \bar{S}_t^{[k]} &= \sum_{i=0}^{2N} w_c^{[i]} (\bar{Z}_t^{[i][k]} - \hat{z}_t^{[k]})(\bar{Z}_t^{[i][k]} - \hat{z}_t^{[k]})^T + R_t \\ \bar{\Sigma}_{t,j}^{[k]} &= \sum_{i=0}^{2N} w_c^{[i]} (\chi^{[i][k]} - \mu_{t,j-1}^{[k]})(\bar{Z}_t^{[i][k]} - \hat{z}_t^{[k]})^T\end{aligned}\quad (22)$$

其中, $h(\cdot)$ 表示机器人观测模型, x_t^k 表示第 k 个粒子状态向量中的机器人位姿信息, $\bar{Z}_t^{[i][k]}$ 是环境点 j 经过非线性观测后的 sigma 点, $\hat{z}_t^{[k]}$ 是观测的预测值, $\bar{S}_t^{[k]}$ 为观测预测值的协方差, $\bar{\Sigma}_{t,j}^{[k]}$ 为互相关协方差, 权值 $w_c^{[i]}$ 和 $w_m^{[i]}$ 按照式(5)计算。

根据式(11)~式(14)计算得到经过渐消因子调整后的观测预测值 $\hat{z}_t^{[k]}$ 、观测值协方差 $\bar{S}_t^{[k]}$ 和互相关协方差 $\bar{\Sigma}_{t,j}^{[k]}$ 。

最后计算卡尔曼增益 $\bar{K}_t^{[k]}$, 并进一步得到 t 时刻更新后的环境点位置估计和协方差矩阵:

$$\begin{aligned}\bar{K}_t^{[k]} &= \bar{\Sigma}_{t,j}^{[k]} (\bar{S}_t^{[k]})^{-1} \\ \mu_{t,i}^{[k]} &= \mu_{t,i-1}^{[k]} + \bar{K}_t^{[k]} (z_t - \hat{z}_t^{[k]}) \\ \Sigma_{t,j}^{[k]} &= \Sigma_{t,j-1}^{[k]} - \bar{K}_t^{[k]} \bar{S}_t^{[k]} (\bar{K}_t^{[k]})^T\end{aligned}\quad (23)$$

(2) 环境地图创建

当环境点首次被观测到时, 需要创建其初始位置估计及其协方差矩阵。在 FastSLAM 算法框架中, 机器人位姿估计和环境地图维护是分离进行的。这意味着可以使用测量噪声协方差矩阵作为环境点的初始协方差。因此, 使用当前观测值 z_t 和观测噪声协方差矩阵 \mathbf{R}_t 生成 sigma 点集 $\psi^{[i][k]}$, 观测值 z_t 的维数为 l , 初始环境点估计位置均值 $\mu_t^{[k]}$ 和协方差矩阵 $\Sigma_t^{[k]}$ 的计算式为:

$$\begin{aligned}\bar{M}_t^{[i][k]} &= h^{-1}(\psi^{[i][k]}, x_t^{[k]}), i = 0, \dots, 2l \\ \mu_t^{[k]} &= \sum_{i=0}^{2l} w_m^{[i]} \bar{M}_t^{[i][k]}\end{aligned}\quad (24)$$

$$\Sigma_t^{[k]} = \sum_{i=0}^{2l} w_c^{[i]} (\bar{M}_t^{[i][k]} - \mu_t^{[k]})(\bar{M}_t^{[i][k]} - \mu_t^{[k]})^T$$

3.4 算法流程

本文算法的具体过程如下。

步骤 1 采样机器人位姿。利用前一时刻的机器人位姿估计结果和当前观测值, 使用式(21)估计机器人位姿的建议分布函数, 并从中采样粒子。

步骤 2 权重计算。使用式(25)计算每个粒子的权重, 生成加权粒子集。

$$w_t^{[k]} = w_{t-1}^{[k]} \frac{p(z_t | x_t^k) p(x_t^k | x_{t-1}^k, u_t)}{q(x_t^k | x_{t-1}^k, z_t, u_t)}\quad (25)$$

步骤 3 地图更新。对于已经存储在粒子中的重复性观测路标, 根据式(23)更新其位置估计均值和协方差矩阵。对于第一次被观测到的路标, 根据式(24)初始化环境路标的位置估计均值和协方差矩阵, 并加入到粒子已建立的环境地图中。

步骤 4 重采样。如果有效样本容量 N_{eff} 小于指定阈值, 则对粒子集进行重采样, 得到新的粒子集。重采样后, 粒子的权重重新分配为 $1/N$ 。

$$N_{\text{eff}} = \text{round}(1 / \sum_{i=1}^N w_i^2)\quad (26)$$

其中, $\text{round}(\cdot)$ 表示向最邻近整数取整计算。

4 仿真结果和实验分析

4.1 仿真模型

本文使用的移动机器人模型的具体描述如下。机器人 t 时刻的位姿表示为 $(x_t, y_t, \varphi_t)^T$, 其中 (x_t, y_t) 是机器人的二维坐标, φ_t 是其航向角。则机器人的位姿为:

$$\begin{bmatrix} x_t \\ y_t \\ \varphi_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v\Delta t \cos(\varphi_{t-1} + \gamma) \\ y_{t-1} + v\Delta t \sin(\varphi_{t-1} + \gamma) \\ \varphi_{t-1} + \gamma \end{bmatrix} + w_t\quad (27)$$

其中, Δt 为运行时间, γ 为 $t-1$ 时刻到 t 时刻内的移动机器人航向角变换值, w_t 表示系统噪声。

本文使用的观测模型为:

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_j - x_t)^2 + (y_j - y_t)^2} \arctan(\frac{y_j - y_t}{x_j - x_t}) - \varphi_t \\ u_t \end{bmatrix}\quad (28)$$

其中, r 和 θ 分别表示观测到的地图点与移动机器人的距离和运动方向夹角, (x_j, y_j) 指观测到的第 j 个地图点的位置坐标, u_t 表示观测噪声。

4.2 仿真结果分析

为了验证所提算法的实际有效性, 本文在 MATLAB 仿真平台上通过独立实验对本文算法和若干对比算法进行 SLAM 仿真对比分析。

首先创建一个基于路标和航标点特征的仿真环境, 仿真环境设置为 23 个地图点以及 16 个航标。机器人的相关参数如下: 运动速度 $v=3\text{ m/s}$; 转向角速度 a_k 为 $\pi/6\text{ rad/s}$; 机器人携带传感器采样间隔时间 $\Delta t=0.025\text{ s}$, 运动过程噪声 $\sigma_v=0.3\text{ m/s}$, $\sigma_G=3$; 观测噪声 $\sigma_r=0.4\text{ m}$, $\sigma_\theta=4^\circ$ 。图 2 给出了创建的仿真环境, 其中星号代表地图点, 圆圈代表航标, 曲线代表确定的运动路径, 机器人由原点 $(0,0)$ 处沿航标逆时针运动。

图 3 是本文算法与 FastSLAM 算法、UFastSLAM 算法^[15]、PSO-FastSLAM 算法^[19]、AFFastSLAM 算法^[16] 以及

GFA-UFastSLAM 算法^[20] 的对比实验图。其中灰色轨迹是各算法估计的机器人运动轨迹, 黑色轨迹是真实轨迹。图 3 上部从左向右依次为 FastSLAM, UFastSLAM 和 PSO-FastSLAM 算法的结果; 下部从左向右依次为 AFFastSLAM, GFA-UFastSLAM 算法及本文算法的结果。从图 3 可以看出, 本文算法估计的机器人运动轨迹和真实轨迹的重合程度最高, 这说明它具有较高的估计精度。

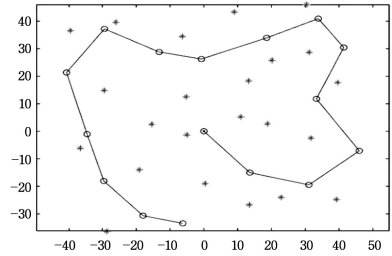


图 2 仿真环境

Fig. 2 Simulation environment

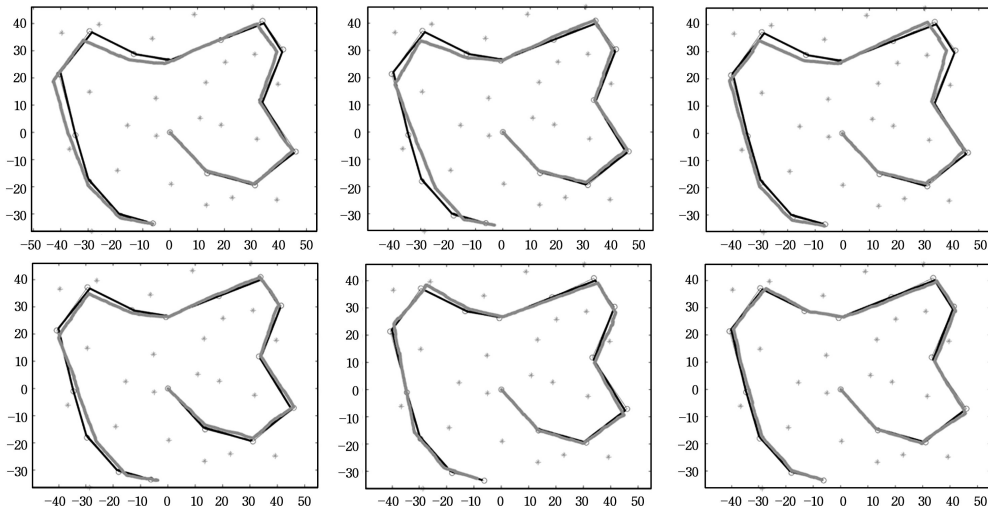


图 3 各算法的仿真结果

Fig. 3 Estimate results of various algorithms

为了避免仿真结果的偶然性, 在同样条件下对几种算法分别测试 20 次, 取均方误差作为估计精度的判定标准。表 1 列出了本文算法和对比算法仿真 20 次的位置、航行角和环境点估计均方误差。可以看出, 本文算法对机器人位姿估计和环境点位置估计的均方误差均为几种算法中最低的。

表 1 $\sigma_r=0.4\text{ m}$, $\sigma_\theta=4^\circ$ 下各算法的估计均方误差

Table 1 Mean value error estimation of various algorithms under

$$\sigma_r=0.4\text{ m}, \sigma_\theta=4^\circ$$

算法	位置均方 误差/m	航行角 均方误差/ $^\circ$	环境点 均方误差/m
FastSLAM	1.1382	0.0287	1.1043
UFastSLAM	1.0619	0.0281	0.9954
PSO-FastSLAM	1.0421	0.0279	0.9872
AFFastSLAM	1.0217	0.0274	0.9793
GFA-UFastSLAM	0.8911	0.0247	0.8769
本文算法	0.8107	0.0235	0.7943

为了避免仿真环境噪声强度带来的偶然性, 保持运动过程噪声 $\sigma_v=0.3\text{ m/s}$, $\sigma_G=3^\circ$ 不变, 改变观测噪声为 $\sigma_r=0.2\text{ m}$, $\sigma_\theta=2^\circ$, 在此条件下再次对几种算法进行 20 次仿真实验。如表 2

所列, 观测噪声强度降低后, 各算法的误差都有所下降, 且本文算法的估计误差仍为几种算法中最低的。

表 2 $\sigma_r=0.2\text{ m}$, $\sigma_\theta=2^\circ$ 下各算法的估计均方误差

Table 2 Mean value error estimation of various algorithms under

$$\sigma_r=0.2\text{ m}, \sigma_\theta=2^\circ$$

算法	位置均方 误差/m	航行角 均方误差/ $^\circ$	环境点 均方误差/m
FastSLAM	0.9194	0.0248	0.8815
UFastSLAM	0.8734	0.0229	0.8234
PSO-FastSLAM	0.8487	0.0211	0.8031
AFFastSLAM	0.8218	0.0204	0.7821
GFA-UFastSLAM	0.7431	0.0193	0.7219
本文算法	0.6982	0.0184	0.6431

为了研究本文算法与其他算法在不同的粒子数条件下的性能, 在运动过程中噪声 $\sigma_v=0.3\text{ m/s}$, $\sigma_G=3^\circ$; 在观测噪声 $\sigma_r=0.4\text{ m}$, $\sigma_\theta=4^\circ$ 条件下, 分别在粒子数为 10, 20, 30, 40 时对几种算法进行 20 次仿真实验, 取几种算法的位置估计均方误差作为评价标准。如图 4 所示, 随着粒子数的增多, 几种算法的位置估计的均方误差都逐渐降低, 即位置估计精度越来越

高。但在同样的估计精度上,本文所用的粒子数量最少,因此本文算法可以用更少的粒子达到更高的估计精度。

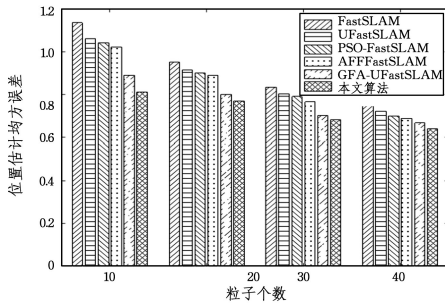


图4 不同粒子数下各算法的均方误差

Fig. 4 Mean value error estimation of various algorithms using different number of particles

结束语 本文将渐消滤波引入 FastSLAM 算法中,提出了一种基于渐消滤波的 FastSLAM 算法。该算法使用渐消无迹卡尔曼滤波计算机器人位姿建议分布函数,利用渐消因子充分提取当前观测值中的有效信息,使得到的建议分布函数更为精确,减缓了粒子的退化程度。在 MATLAB 环境中应用该算法对移动机器人模型进行仿真实验,实验结果表明,本文算法能够有效改善 FastSLAM 算法中的粒子退化情况,且有着更高的估计精度。本文的研究重点在于得到更精确的粒子建议分布函数,但是算法重采样步骤会产生粒子多样性缺乏的问题,未来可以对算法中的重采样步骤进行研究。

参考文献

- [1] DURRANT-WHYTE H, BAILEY T. Simultaneous localization and mapping: part I[J]. IEEE Robotics & Automation Magazine, 2006, 13(2): 99-108.
- [2] BAILEY T, DURRANT-WHYTE H. Simultaneous localization and mapping (SLAM): part II[J]. IEEE Robotics & Automation Magazine, 2006, 13(2): 108-117.
- [3] KARRER M, SCHMUCK P, CHLI M. CVI-SLAM—Collaborative Visual-Inertial SLAM[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 2762-2769.
- [4] GOMEZ-OJEDA R, ZUIGA-NOLA D, MORENO F A, et al. PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments[J]. IEEE Transactions on Robotics, 2019, 35(3): 734-746.
- [5] NIU X N, LIU H Z, YUAN J Z, et al. RGB-D Indoor Location and Map Building Based on Inliers Tracking Statistics[J]. Computer Engineering, 2018, 44(9): 15-21, 27.
- [6] WANG T C, CAI Y F, TANG Z M. SLAM Method Based on Region Particle Swarm Optimization and Partial Gaussian Resampling[J]. Computer Engineering, 2017, 43(11): 310-316.
- [7] HWANG S Y, SONG J B. Monocular vision-based SLAM in indoor environment using corner, lamp, and door features from upward-looking camera[J]. IEEE Transactions on Industrial Electronics, 2011, 58(10): 4804-4812.
- [8] MUNGUIA R, SARQUIS U, YOLANDA B, et al. Vision-Based SLAM System for Unmanned Aerial Vehicles [J]. Sensors, 2016, 16(3): 372.
- [9] WANG H J, FU G X, LI J, et al. Strong tracking CKF based SLAM method for unmanned underwater vehicle[J]. Chinese Journal of Scientific Instrument, 2013, 34(11): 2542-2550.
- [10] CASTELLANOS J, NEIRA J, TARDOS J. Limits to the consistency of EKF-based SLAM[J]. IFAC Proceedings Volumes, 2004, 37(8): 716-721.
- [11] HUANG S, DISSANAYAKE G. Convergence and consistency analysis for extended Kalman filter based SLAM [J]. IEEE Transactions on Robotics, 2007, 23(5): 1036-1049.
- [12] MONTEMARLO M. FastSLAM: A factored solution to the simultaneous localization and mapping problem [C] // Proceedings of the AAAI National Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2002: 593-598.
- [13] BAILEY T, BROOKS A. Hybrid SLAM: Combining FastSLAM and EKF-SLAM for reliable mapping [C] // Proceedings of International Workshop on Algorithmic Foundation of Robotics VI-II. Berlin: Springer, 2009: 647-661.
- [14] ZHANG L, MENG X J, CHEN Y W. Convergence and consistency analysis for FastSLAM [C] // Proceedings of the Intelligent Vehicles Symposium. Piscataway: IEEE Press, 2009: 447-452.
- [15] KIM C, SAKTHIVEL R, CHUNG W K. Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem [J]. IEEE Transactions on Robotics, 2008, 24(4): 808-820.
- [16] LIU D, DUAN J M, YU H X. FastSLAM algorithm based on adaptive fading extended Kalman filter [J]. Systems Engineering and Electronics, 2016, 38(3): 644-651.
- [17] GE Q, SHAO T, CHEN S, et al. Carrier Tracking Estimation Analysis by Using the Extended Strong Tracking Filtering [J]. IEEE Transactions on Industrial Electronics, 2017, 64(2): 1415-1424.
- [18] WANG X X, ZHAO L, XIA Q X, et al. Strong tracking filter based on unscented transformation [J]. Control and Decision, 2010, 25(7): 1063-1068.
- [19] PAN S, LIU H G, LI N, et al. An improved FastSLAM algorithm based on quantum particle swarm optimization immune particle filter [J]. Computer Simulation, 2018(8): 202-205.
- [20] CHEN S M, LIU J K, XIAO J. Unscented FastSLAM2.0 algorithm based on gravitational field optimization [J]. Control Theory and Applications, 2018, 35(8): 139-146.



WANG Bing-zhou, born in 1995, post-graduate. His main research interests include signal processing and so on.



WANG Hui-bin, born in 1967, Ph. D. professor, is a member of China Computer Federation. His main research interests include image processing and multi-source information fusion.