第 35 卷第 9 期
2018 年 9 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 35 No. 9
Sep. 2018

DOI: 10.7641/CTA.2018.60849

二阶中心差分粒子滤波FastSLAM算法

代嘉惠[1,2†], 许鹏程[2], 李小波[2]

(1. 重庆大学 资源及环境科学学院, 重庆 400030; 2. 中煤科工集团重庆研究院有限公司, 重庆 400037)

摘要: 为改善机器人同时定位与建图(simultaneous localization and mapping, SLAM)算法中非线性系统状态估计精度不高, 计算繁杂的问题, 本文创新性地提出了基于二阶中心差分滤波并融合最新观测数据来产生建议分布函数的新算法. 新算法基于二阶sterling插值公式处理SLAM中的非线性系统问题, 无须计算雅可比矩阵, 容易实现. 此外, 该算法使用Cholesky分解技术, 在SLAM概率估计中直接依据协方差平方根因子进行传播, 保证协方差矩阵正定性的同时减小了局部线性化的截断误差. 仿真试验表明, 在粒子数相同的情况下, 二阶中心差分FastSLAM(second-order FastSLAM, SOFastSLAM)在不同噪声条件下的估计精度均优于FastSLAM2.0和UFastSLAM算法, 且用时最少, 证实了SOFastSLAM算法的优越性.

关键词: FastSLAM; 即时定位与地图构建; 协方差矩阵; 粒子滤波; 二阶中心差分粒子滤波; 移动机器人

引用格式: 代嘉惠, 许鹏程, 李小波. 二阶中心差分粒子滤波FastSLAM算法. 控制理论与应用, 2018, 35(9): 1382 – 1390

中图分类号: TP273　　**文献标识码**: A

Second order central difference particle filter FastSLAM algorithm

DAI Jia-hui[1,2†], XU Peng-cheng[2], LI Xiao-bo[2]

(1. College of Resources and Environmental Sciences, Chongqing University, Chongqing 400030, China;
2. China Coal Technology Engineering Group Chongqing Research Institute, Chongqing 400037, China)

Abstract: For improving the accuracy of nonlinear system state estimation and calculating complex problems in a S-LAM algorithm, this paper proposes an innovative method based on a second-order central difference filter and the novel observational data to generate a proposal distribution function. The new algorithm uses the second order sterling interpolation formula to handle the nonlinear system problem in SLAM without calculating the Jacobian matrix and is easily implemented. Furthermore, by using Cholesky decomposition technology to directly propagate the covariance square root factor in SLAM probability estimation, the proposed algorithm not only guarantees the covariance matrix is positive and definite but also reduces the truncation error of local linearization. Notably, as verified by simulation testing with equivalent number of particles, the second-order center differential FastSLAM (SOFastSLAM) showed better estimation accuracy than that of FastSLAM2.0 and UFastSLAM algorithms in different noise conditions with the shortest computation time, confirming the superiority of the SOFastSLAM.

Key words: FastSLAM; simultaneous localization and mapping; covariance matrix; particle filter; second order central difference particle filter; mobile robot

Citation: DAI Jiahui, XU Pengcheng, LI Xiaobo. Second order central difference particle filter FastSLAM algorithm. *Control Theory & Applications*, 2018, 35(9): 1382 – 1390

1 引言(Introduction)

移动机器人在未知环境中依靠自身携带的传感器获取环境的特征信息, 增量式地创建全局环境地图, 并同时进行定位[1](simultaneous localization and mapping, SLAM), 是移动机器人在未知环境中实现智能自主定位与导航的关键技术之一. 1980年起, SLAM技术已引起国际上机器人领域的研究热潮. 目前已有多种解决此问题的方法, 如基于概率估计的方法, 以及非概率估计方法[2], 目前主流的算法是基于概率论的估计方法.

最早使用的概率估计方法是基于卡尔曼滤波的SLAM 算法[3–4], 以及在其基础上改良的 unscented

收稿日期: 2016−11−11; 录用日期: 2018−05−07.

†通信作者. E-mail: wjj-123-321@163.com; Tel.: +86 18623144466.

本文责任编委: 吴敏.

国家自然科学基金项目(61164015, 61305132), 江西省自然科学基金项目(20151BAB207043)资助.

Supported by the National Natural Science Foundation of China (61164015, 61305132) and the National Natural Science Foundation of Jiangxi Province (20151BAB207043).

Kalman filter SLAM (UKFSLAM)算法[5]和extended Kalman filter SLAM (EKFSLAM)算法[6]. 然而, 这类算法具有数据的关联性差和正比于特征个数的二次方计算量与存储量的问题. 2003年, Montermerlo 提出了基于Rao-Blackwellized particle filter的FastSLAM算法[7]. 其具有适用于复杂的环境、数据的关联性好、计算代价低等特点, 已经成为SLAM算法框架研究方面的流行趋势. FastSLAM算法根据环境特征的独立性, 将高维的移动机器人定位与建图问题转化成利用Rao-Blackwillised公式(RB公式)解耦为定位后建图的低维问题[8], 使SLAM算法的运算效率更高. 其中, FastSLAM1.0[9]基于移动机器人前一时刻的位置和控制变量进行下一时刻的位置预测, 容易发生粒子退化问题. 在FastSLAM2.0[7]版本中, Montermerlo等人使用EKF将最新的路标观测数据融入重要性函数分布之中, 使得采样得到的粒子向似然区域移动, 较为符合移动机器人的实际情况. 实验表明, FastSLAM2.0基本上把粒子退化问题解决了. 基于类似的思想, 在FastSLAM算法框架下, 学者Grisetti等提出了栅格地图中的FastSLAM算法[10]. 该方法将地图离散化, 同时使用激光和里程计信息来设计SLAM的重要性采样函数, 但依旧存在栅格的精度严重影响FastSLAM算法的问题. 随后, Kim等人提出了UFastSLAM算法, 其关键技术是使用UPF[11]代替了FastSLAM2.0中的EKF, 从而使滤波精度得到了很大的提升. 但是, 该算法不足之处就是运算花费的时间比较长, 效率不高. 2009年Arasaratna和Haykin提出了容积卡尔曼滤波算法(cubature Kalman filter, CKF)[12]. CKF算法具备数值精度高、滤波稳定性以及非线性逼近性能优秀, 且CKF滤波精度较高、实现较为简单. 当然, 其算法也有一些固有的缺陷: SLAM滤波的预测和更新迭代过程均需要连续执行协方差矩阵平方根分解和操作协方差矩阵重构, 这些数值敏感且繁琐的操作无疑耗费了大量时间. 学者朱奇光等提出了一种球面单径容积FastSLAM算法(spherical single radius capacity, SSRC-FastSLAM)[13]. 该算法运用3阶球面单径准则来计算SLAM中的非线性高斯权重积分用于提高算法精度. 虽然精度较高, 但是仍然无法克服容积卡尔曼滤波算法的一些固有缺陷. 学者祝继华等提出了基于中心差分粒子滤波的SLAM算法[14], 此算法虽然计算简单, 但是只能达到一阶泰勒精度.

经过调研发现, 现有的SLAM算法存在要么精度高, 但是计算复杂; 要么计算简单速度快, 但精度不高的问题. 针对这些问题, 本文创新性的提出了二阶中心差分FastSLAM算法(second-order FastSLAM, SO-FastSLAM). 该算法基于Sterling插值公式[15], 只利用有限点的函数值, 而不是对函数求导, 从而取得更简单的逼近公式. 采用二阶中心差分滤波方法[16]并融合最新观测数据来产生重要性分布函数[17], 并进行仿真实验, 对其性能进行验证. 该算法的主要特点在于: 1) 使用二阶中心差分粒子滤波算法, 不使用Taylor展开公式、不用计算雅可比矩阵, 精度能达到二阶泰勒级数截断, 计算的复杂性大大降低. 2) 粒子的结构由SLAM状态及其不确定性的协方差平方根因子组成, 在传播过程中保证了协方差矩阵的正定性与对称性, 同时减少了计算过程中的数值误差.

2 SLAM 问题的描述 (Description of SLAM problem)

使用以Bayes为核心的滤波器是很多SLAM算法的相同之处, 它要求预测与观测方程均是线性的, 如Kalman滤波器[18]. EKF滤波器允许使用非线性的预测与观测方程, 但其本质是利用一阶Taylor公式将非线性方程转化为线性方程. 在复杂或者维数较高的SLAM模型中采用EKF滤波往往精度较低, 甚会导致发散、不收敛等问题的出现.

2.1 FastSLAM算法框架(FastSLAM algorithm framework)

FastSLAM采用更加详尽的分布估计解决非线性的SLAM问题. 不同于只估计机器人位置的传统SLAM算法, FastSLAM算法的关键在于使用了机器人的全观测数据$Z^k=\{z_1,z_2,\cdots,z_k\}$、全控制信息$U^k=\{u_1,u_2,\cdots,u_k\}$, 来估计机器人的运动轨迹$S^k=\{s_1,s_2,\cdots,s_k\}$. 因此, 根据SLAM问题的环境特征位置得出独立性结论: 若已经知道移动机器人运动轨迹$S^k=\{s_1,s_2,\cdots,s_k\}$, 环境特征$\theta^m=\{\theta_1,\theta_2,\cdots,\theta m\}$位置相对独立, 则联合后验分布$p(s^k,\theta|z^k, u^{k-1},m^k)$可表示为

$$p(s^k,\theta|z^k,u^{k-1},m^k)=\\ p(s^k|z^k,u^{k-1},m^k)\prod_{m=1}^{M}p(\theta|s^k,z^k,u^{k-1},m^k).\quad(1)$$

此为FastSLAM算法的核心. 其中: m^k为相关性变量.

考虑环境$\theta^m=\{\theta_1,\theta_2,\cdots,\theta m\}$的独立特性, Fast SLAM算法分别采用了粒子滤波算法和卡尔曼滤波算法来估计机器人位姿和环境路标. 采用N个粒子的粒子滤波器估计机器人轨迹后验$p(s^k|z^k,u^{k-1})$, 其中每一个粒子单独维护一幅环境地图θ和机器人路径s^k[19]. 同时, 假设机器人轨迹已知, 鉴于路标彼此之间相互独立, 则对每个环境路标(共M个)后验$p(\theta|s^k,z^k,u^{k-1})$应用单独的卡尔曼滤波器进行估算. 这样, 共由$N\times M+1$个滤波器构成了一个完整的SLAM算法[20].

在FastSLAM中, 每一个粒子的组成如下:

$$X_k^{[i]}=\{<x_k^{[i]},p_k^{[i]}>,\cdots,(\mu_k^{[i][m]}\sum_k^{[i][m]})\cdots\}.\quad(2)$$

在式(2)中: $X_k^{[i]}$表示编号为i的粒子; $x_k^{[i]}$为i粒子对机器人当前位姿的假设; $p_k^{[i]}$为i粒子的协方差; $(\mu_k^{[i][m]}, \sum_k^{[i][m]})$为$i$粒子所维持的特征地图中第$m$个路标的全局坐标和它的不确定性协方差.

2.2 SLAM 算法基本流程 (The process of SLAM algorithm)

一般来说SLAM算法可以分成以下几个步骤.

1) 通过采样机器人最新位姿信息, 扩展其路径的后验概率.

2) 将机器人传感器上观测到的路标信息进行估计更新.

3) 计算粒子重要性权重, 决定是否需要进行重采样处理.

3 二阶中心差分滤波器(Second-order central difference filter)

二阶中心差分滤波器(second-order center differential filter, SOCDF)[21]是利用Stirling插值公式将非线性模型按中心差分形式展开[22], 不需要计算函数的雅可比矩阵, 甚至非线性函数不连续且存在奇异点时也能进行状态估计错误!未找到引用源.

设$x \in \mathbb{R}^n$为n维矢量, 则$y = y(x)$在$\bar{x}$处用Stirling插值公式展开为

$$y \approx f(\bar{x}) + \bar{D}_{\mathrm{x}} f(\bar{x}) + \frac{1}{2!} \bar{D}_{\mathrm{x}}^2 f(\bar{x}), \tag{3}$$

式中$\bar{D}_{\mathrm{x}}$和$\bar{D}_{\mathrm{x}}^2$分别表示一阶差分和二阶差分算子, 具体算法如下:

$$\bar{D}_{\mathrm{x}} f(\bar{x}) = \frac{1}{\lambda} \left[\sum \Delta x_p \mu_p \delta_p\right] f(\bar{x}), \tag{4a}$$

$$\begin{aligned}\bar{D}_{\mathrm{x}}^2 f(\bar{x}) = \frac{1}{\lambda^2} [&\sum_{p=1}^{n} \Delta x_p^2 \delta_p^2 + \\ &\sum_{p=1}^{n} \sum_{\substack{q=1 \\ q \neq p}}^{n} \Delta x_p \Delta x_q (\mu_p \delta_p)(\mu_p \delta_p)] f(\bar{x}).\end{aligned} \tag{4b}$$

在这里: δ_p表示其偏微分算子; μ_p表示均值算子; λ为给定的变量步长, 其最优选择是$\lambda^2 = \mathrm{e}^{3\times(n_{\mathrm{x}}+n_{\mathrm{y}})}$($n_{\mathrm{x}}$, n_{y}表示X的维数).

二阶中心差分滤波器不需要计算函数的偏导数, 且可应用于任意非线性函数. 滤波过程中采用了协方差的矩阵平方根形式, 保证了协方差矩阵在传播过程中的正定性, 具有良好的数值特性且滤波精度较高.

4 二阶中心差分SLAM算法(Second-order central difference SLAM algorithm)

结合FastSLAM算法基本框架, 本文引入二阶中心差分粒子滤波, 通过二阶中心差分粒子滤波提高新算法的精度和效率. 具体算法如下:

4.1 机器人的位姿估计(Robot pose estimation)

对于一般的非线性系统, 在离散的时间内系统状态方程和观测方程为

$$x_{k+1} = f_k(x_k, u_k, w_k), \tag{5}$$

$$y_k = g_k(x_k, v_k), \tag{6}$$

其中: x_{k+1}表示机器人的位置参数; y_k是环境特征观测值; f表示非线性机器人运动学模型; g表示环境观测模型; u_k为在离散时间区间$[k+1, k]$之内对机器人的控制作用; w_k为机器人上所安装的传感器的观测噪声, 其协方差为R; v_k为机器人的控制噪声, 其协方差矩阵为Q.

采用Cholesky分解法可以得到

$$\begin{cases} Q = S_{\mathrm{v}} \times S_{\mathrm{v}}^{\mathrm{T}}, \\ \bar{P} = \bar{S}_{\mathrm{x}} \times \bar{S}_{\mathrm{x}}^{\mathrm{T}}, \end{cases} \tag{7}$$

$$\begin{cases} R = S_{\mathrm{w}} \times S_{\mathrm{w}}^{\mathrm{T}}, \\ \hat{P} = \hat{S}_{\mathrm{x}} \times \hat{S}_{\mathrm{x}}^{\mathrm{T}}. \end{cases} \tag{8}$$

过程噪声协方差矩阵Q和观测噪声协方差矩阵R的因子S_{v}与S_{w}可以预先得到. 预测协方差$\bar{P}$的因子$\bar{S}_{\mathrm{x}}$和估计协方差$\hat{P}$的因子$\hat{S}_{\mathrm{x}}$在滤波过程中不断的被更新和修正.

基于多元函数二阶中心插值近似公式算法, 计算每个粒子的一阶均差向量:

$$\begin{aligned}S_{\mathrm{x}\hat{\mathrm{x}}}^{(1)}(k) = \{&\frac{1}{2\lambda}(f(\hat{x}_{k-1} + \lambda \hat{S}_{\mathrm{x},j}, \bar{v}_{k-1}) - \\ &f(\hat{x}_{k-1} - \lambda \hat{S}_{\mathrm{x},j}, \bar{v}_{k-1}))\},\end{aligned} \tag{9}$$

$$\begin{aligned}S_{\mathrm{xv}}^{(1)}(k) = \{&\frac{1}{2\lambda}(f(\hat{x}_{k-1}, \bar{v}_{k-1} + \lambda S_{\mathrm{v},j}) - \\ &f(\hat{x}_{k-1}, \bar{v}_{k-1} - \lambda S_{\mathrm{v},j}))\},\end{aligned} \tag{10}$$

$$\begin{aligned}S_{\mathrm{y}\hat{\mathrm{x}}}^{(1)}(k) = \{&\frac{1}{2\lambda}(g(\bar{x}_k + \lambda \bar{S}_{\mathrm{x},j}, \bar{w}_k) - \\ &g(\bar{x}_k - \lambda \bar{S}_{\mathrm{x},j}, \bar{w}_k))\},\end{aligned} \tag{11}$$

$$\begin{aligned}S_{\mathrm{yw}}^{(1)}(k) = \{&\frac{1}{2\lambda}(g(\bar{x}_k, \bar{w}_k + \lambda S_{\mathrm{w},j}) - \\ &g(\bar{x}_k, \bar{w}_k - \lambda S_{\mathrm{w},j}))\}.\end{aligned} \tag{12}$$

计算系统中每个粒子的二阶均差向量:

$$\begin{aligned}S_{\mathrm{x}\hat{\mathrm{x}}}^{(2)}(k) = \{&\frac{\sqrt{\lambda^2 - 1}}{2\lambda^2}(f(\hat{x}_{k-1} + \lambda \hat{S}_{\mathrm{x},j}, \bar{v}_{k-1}) + \\ &f(\hat{x}_{k-1} - \lambda \hat{S}_{\mathrm{x},j}, \bar{v}_{k-1}) - 2f(\hat{x}_{k-1}, \bar{v}_{k-1}))\},\end{aligned} \tag{13}$$

$$\begin{aligned}S_{\mathrm{xv}}^{(2)}(k) = \{&\frac{\sqrt{\lambda^2 - 1}}{2\lambda^2}(f(\hat{x}_{k-1}, \bar{v}_{k-1} + \lambda \hat{S}_{\mathrm{v},j}) + \\ &f(\hat{x}_{k-1}, \bar{v}_{k-1} - \lambda \hat{S}_{\mathrm{v},j}) - 2f(\hat{x}_{k-1}, \bar{v}_{k-1}))\},\end{aligned} \tag{14}$$

$$\begin{aligned}S_{\mathrm{y}\hat{\mathrm{x}}}^{(2)}(k) = \{&\frac{\sqrt{\lambda^2 - 1}}{2\lambda^2}(g(\bar{x}_k + \lambda \bar{S}_{\mathrm{x},j}, \bar{w}_k) + \\ &g(\bar{x}_k - \lambda \bar{S}_{\mathrm{x},j}, \bar{w}_k) - 2g(\bar{x}_k, \bar{w}_k))\},\end{aligned} \tag{15}$$

$$S_{\rm yw}^{(2)}(k)=\{\frac{\sqrt{\lambda^2-1}}{2\lambda^2}(g(\bar{x}_k,\bar{w}_k+\lambda\bar{S}_{{\rm w},j})+ g(\bar{x}_k,\bar{w}_k-\lambda\bar{S}_{{\rm w},j})-2g(\bar{x}_k,\bar{w}_k))\}, \quad (16)$$

式中: $\hat{S}_{{\rm x},j}, S_{{\rm v},j}, \bar{S}_{{\rm x},j}, S_{{\rm w},j}$分别表示矩阵$\hat{S}_{\rm x}, S_{\rm v}, \bar{S}_{\rm x}, S_{\rm w}$的第$j$列, $\hat{x}_{k-1}, \bar{x}_k$分别表示机器人状态在$k-1$时刻的估计值和k时刻的预测值.

4.1.1 机器人状态预测(Robot state forecast)

为计算机器人在$k-1$时刻位姿的估计值$\hat{x}_{k-1}$经运动学模型f传播后, 在k时刻位姿的预测值$\bar{x}_k$, 需按下式进行计算:

$$\bar{x}_k=\frac{\lambda^2-n_{\rm x}-n_{\rm y}}{\lambda^2}f(\hat{x}_{k-1},\bar{v}_{k-1})+\frac{1}{2\lambda^2}\sum_{i=1}^{n_{\rm x}}[f(\hat{x}_{k-1}+\lambda\hat{s}_{{\rm x},i},\bar{v}_{k-1})+f(\hat{x}_{k-1}-\lambda s_{{\rm x},i},\bar{v}_{k-1})]+\frac{1}{2\lambda^2}\sum_{i=1}^{n_{\rm v}}[f(\hat{x}_{k-1},\bar{v}_{k-1}+\lambda s_{{\rm v},i})+f(\hat{x}_{k-1},\bar{v}_{k-1}-\lambda s_{v,i})], \quad (17)$$

式中$n_{\rm x}, n_{\rm y}$分别是机器人状态向量和量测噪声向量的维数.

预测状态误差均差矩阵为

$$\bar{S}_{\rm x}(k)=[S_{{\rm x}\hat{\rm x}}^{(1)}\ \ S_{\rm xv}^{(1)}\ \ S_{{\rm x}\hat{\rm x}}^{(2)}\ \ S_{\rm xv}^{(2)}]. \quad (18)$$

这里需要使用Householder变换将长方形矩阵$\bar{S}_{\rm x}(k)$变换成三角化的方形Cholesky因子, 同时保证变换前后的$\bar{S}_{\rm x}(k)\times\bar{S}_{\rm x}^{\rm T}(k)$值不变, 即对$\bar{S}_{\rm x}^{\rm T}(k)$进行QR分解, 得到新的矩阵$\bar{S}_{\rm x}(k)$:

$$\begin{cases}[Q,R]=qr(\bar{S}_{\rm y}^{\rm T}(k)),\\ \bar{S}_{\rm x}(k)=R.\end{cases} \quad (19)$$

在式(19)中, $qr(\cdot)$为QR分解的操作函数, 变换后的$\bar{S}_{\rm x}(k)\times\bar{S}_{\rm x}(k)$为$n_{\rm x}\times n_{\rm x}$的矩阵.

4.1.2 机器人状态更新(Robot state update)

当编号为m的路标再一次被移动机器人观测到时, 运用机器人对此路标的重复观测所得到的信息对预测的机器人状态$\bar{x}_k$及其协方差矩阵进行更新, 过程如下:

预测测量方程:

$$\bar{y}_k=\frac{\lambda^2-n_{\rm x}-n_{\rm w}}{\lambda^2}g(\bar{x}_k,\bar{w}_k)+\frac{1}{2\lambda^2}\sum_{i=1}^{n_{\rm x}}[g(\bar{x}_k-\lambda\bar{s}_{{\rm x},i},\bar{w}_k)+g(\bar{x}_k+\lambda\bar{x}_{{\rm x},i},\bar{w}_k)]+\frac{1}{2\lambda^2}\sum_{i=1}^{n_{\rm w}}[g(\bar{x}_k,\bar{w}_k+\lambda s_{{\rm w},i})+g(\bar{x}_k,\bar{w}_k-\lambda s_{{\rm w},i})], \quad (20)$$

式中$n_{\rm w}$是量测噪声向量的维数.

预测测量均方差矩阵为

$$\bar{S}_{\rm y}(k)=[S_{{\rm y}\bar{\rm x}}^{(1)}\ \ S_{\rm yw}^{(1)}\ \ S_{{\rm y}\bar{\rm x}}^{(2)}\ \ S_{\rm yw}^{(2)}]. \quad (21)$$

这里同样需要使用Householder变换, 具体过程如下:

$$\begin{cases}[Q,R]=qr(\bar{S}_{\rm y}^{\rm T}(k)),\\ \bar{S}_{\rm y}(k)=R.\end{cases} \quad (22)$$

在式(22)中, $qr(\cdot)$为QR分解操作函数, 变换后的$\bar{S}_{\rm y}(k)\times\bar{S}_{\rm y}(k)$为$n_{\rm y}\times n_{\rm y}$的矩阵.

为了使用卡尔曼滤波技术对机器人状态进行更新, 首先计算移动机器人预测状态和环境路标观测的交互协方差:

$$P_{\rm xy}(k)=\bar{S}_{\rm x}(k)\times[S_{{\rm y}\bar{\rm x}}^{(1)}(k)]^{\rm T}. \quad (23)$$

基于卡尔曼滤波算法, 机器人状态更新为

$$K_k=P_{\rm xy}(k)[\bar{S}_{\rm y}(k)\ \ \bar{S}_{\rm y}^{\rm T}(k)]^{-1}. \quad (24)$$

K_k为Kalman滤波增益, 滤波状态方程为

$$\hat{x}_k=\bar{x}_k+K_k(y_k-\bar{y}_k). \quad (25)$$

计算Cholesky因子:

$$\hat{S}_{\rm x}(k)=[\bar{S}_{\rm x}(k)-K_kS_{\rm yx}^{(1)}(k)\ \ K_kS_{\rm yw}^{(1)}\ \ K_kS_{\rm yx}^{(2)}\ \ K_kS_{\rm yw}^{(2)}]. \quad (26)$$

故k时刻滤波后的机器人状态量为$\hat{x}_k$, 可得观测信息协方差矩阵$\hat{P}(k)=S_{\rm x}(k)\times S_{\rm x}^{\rm T}(k)$.

4.1.3 重采样 (Importance weight and resampling strategy)

在k时刻, 若移动机器人没有观测到地图中已存储的路标, 则根据式(17)–(18)形成的重要性分布进行采样得到新一代粒子. 如果机器人在k时刻观测到一个或多个路标, 则依照观测次序使用(式(20)–(26))每一个路标的观测值对机器人状态及其协方差矩阵进行更新, 每次更新均采用上一次的结果作为初始值. 更新完毕后, 将从机器人状态分布中采集新一批的粒子, 即: $\hat{x}_k^{[i]}\sim N(\hat{x}_k^{[i]},\hat{P}(k))$.

同时, 根据所观测到的新信息和新信息的协方差来计算新一代粒子的重要性权重$\omega_k^{[i]}$:

$$\omega_k^{[i]}\propto\prod{\rm e}^{\frac{(y_k-\bar{y}_k)(y_k-\bar{y}_k)^{\rm T}}{\hat{P}\hat{P}^{\rm T}}}. \quad (27)$$

4.2 环境特征估计(Environment feature estimation)

在k时刻, 对算法中的每一个表示移动机器人位姿的粒子$\hat{x}_k$进行更新之后, 就进入了环境特征的更新环节. 在这一阶段, 对目前时刻重复观测到的路标和首次观测到的路标需使用不同的策略进行处理.

4.2.1 对重复观测到的路标的处理(Deal with the observed landmarks)

在k时刻, 对于重复观测到的路标, 本文采用最大似然估计进行路标匹配. 基于路标特征的离散性, 使用二阶中心差分滤波器对路标特征进行更新, 因此在机器人位姿参数$\hat{x}_k^{[i]}$已估计得到的情况下, 索引号为

m的路标$\langle\mu_{k-1}^{[m]},\Sigma_{k-1}^{[m]}\rangle$观测值估计为

$$\begin{aligned}&\hat{y}_k=\\&\frac{\lambda^2-n_{\rm x}-n_{\rm w}}{\lambda^2}g(\hat{x}_k^{[i]},\bar{w}_k)+\\&\frac{1}{2\lambda^2}\sum_{i=1}^{n_{\rm x}}[g(\hat{x}_k^{[i]}+\lambda\hat{s}_{{\rm x},i},\bar{w}_k)+g(\hat{x}_k^{[i]}-\lambda\hat{s}_{{\rm x},i},\bar{w}_k)]+\\&\frac{1}{2\lambda^2}\sum_{i=1}^{n_{\rm w}}[g(\hat{x}_k^{[i]},\bar{w}_k+\lambda s_{{\rm w},i})+g(\hat{x}_k^{[i]},\bar{w}_k-\lambda s_{{\rm w},i})].\end{aligned}\tag{28}$$

路标观测估计均方差矩阵为

$$\bar{S}_{\hat{\rm y}}(k)=[S_{\hat{\rm y}\bar{\rm x}}^{(1)}\ \ S_{\hat{\rm y}{\rm w}}^{(1)}\ \ S_{\hat{\rm y}\bar{\rm x}}^{(2)}\ \ S_{\hat{\rm y}{\rm w}}^{(2)}].\tag{29}$$

观测信息平方根因子由下式可得:

$$\begin{cases}[Q,R]=qr(\bar{S}_{\hat{\rm y}}^{\rm T}(k)),\\ \bar{S}_{\hat{\rm y}}(k)=R.\end{cases}\tag{30}$$

采用卡尔曼滤波计算移动机器人预测状态和环境路标观测的交互协方差

$$P_{{\rm x}\hat{\rm y}}(k)=\hat{S}_{\rm x}(k)\times[S_{\hat{\rm y}\bar{\rm x}}^{(1){\rm T}}(k)].\tag{31}$$

基于卡尔曼滤波算法, 机器人状态更新为

$$K_k=P_{{\rm x}\hat{\rm y}}(k)\times[\bar{S}_{\hat{\rm y}}(k)\times\bar{S}_{\hat{\rm y}}^{\rm T}(k)]^{-1},\tag{32}$$

K_k为卡尔曼增益. 故k时刻滤波后的更新路标为

$$\mu_k^{[m]}=\mu_{k-1}^{[m]}+K_k(y_k-\bar{y}_k),\tag{33}$$

$$\begin{aligned}\varepsilon(k)=[&\hat{S}_{\rm x}(k)-K_kS_{{\rm y}\hat{\rm x}}^{(1)}(k)\\&K_kS_{\rm yw}^{(1)}\quad K_kS_{{\rm y}\hat{\rm x}}^{(2)}\quad K_kS_{\rm yw}^{(2)}].\end{aligned}\tag{34}$$

协方差矩阵$\Sigma(k)=\varepsilon(k)\times\varepsilon^{\rm T}(k)$.

4.2.2 对首次观测到的路标的处理(Deal with the new landmarks)

对于首次观测到的路标(记其索引号为m),基于路标的独立性和算法效率的因素, 本文采用EKF滤波器对每个观测到的新路标进行初始化

$$x_k^{[i]}\sim p(x_k|s_{k-1}^{[i]},u_k),\tag{35}$$

路标位置均值

$$\mu_{n,k}^{[m]}=g^{-1}(z_k,s_k^{[m]}),\tag{36}$$

路标位置方差

$$\Sigma^m(k)=(\hat{P}(k)R_t^{-1}\hat{P}(k)^t)^{-1}.\tag{37}$$

4.3 算法流程及分析(Algorithm process and analysis)

SOFastSLAM的基本流程如下:

1) 抽样: 在k时刻对每个粒子前一时刻$S_{k-1}^{[i]}$计值, 使用移动机器人的位置量$\hat{x}_{k-1}^{[i]}$、方差$p_{k-1}^{[i]}$, 利用式(7)–(19), 获得机器人位姿的建议分布函数, 并从中抽样获得新的粒子集$\bar{x}_k^{[i]}$;

2) 计算权重: 根据式(27)计算出每个粒子的权重值;

$$\mathrm{Neff}=\frac{1}{\sum\limits_{t=1}^{m}\left(\tilde{w}^{[t]}\right)^2}.\tag{38}$$

3) 判断是否需要进行重采样: 通过式(38)计算有效粒子数Neff来确定是否需要重采样. 当Neff小于某个域值时, 进行重采样, 以减弱粒子的退化现象;

4) 地图更新与扩展: 针对每个粒子, 将首次观测到的特征加入地图, 并对重复观测到的粒子进行位置的更新操作;

5) 重复执行上述式(7)–(38)过程, 直到计算完k时刻所有新的观测值为止.

通过算法分析, 将非线性模型进行线性化的目的在于计算高斯分布经过非线性函数作用后的转移密度. FastSLAM是对非线性函数$f(x)$在$\hat{x}_{k-1}$处线性化所得到, 即$f(x)=f(\hat{x})+\nabla f(\hat{x})(x-\hat{x})$. 其中$\nabla f(\hat{x})$为雅可比矩阵. 显而易见, 当$f(\hat{x})$的非线性程度较高时, 转移密度$N(f(\hat{x}),\hat{p})$ 计算将缺乏准确性.

本文使用二阶中心差分算法进行移动机器人的同时定位与建图. 对维数为$n_{\rm x}$的状态变量进行计算时, 计算点数为$n_{\rm x}$, 数量低于UFastSLAM中无迹变换所需要的$2n_{\rm x}+1$, 以及平方根容积粒子滤波算法所需要的$2n_{\rm x}$. 由于用于计算的点数较少, 所以计算效率得到了很大的提高. 根据文献[24], 应用CHOLESKY分解对协方差矩阵$\hat{P}(k-1)$平方根进行计算的复杂度为$\mathrm{O}(n_{\rm x}^3/6)$, 且计算二阶中心差分点所需要的粒子数为$n_{\rm x}$; 重构协方差$\hat{P}(k)$的复杂度为$\mathrm{O}(n_{\rm x}^3)$, 即传播协方差矩阵的复杂度为$\mathrm{O}(n_{\rm x}^3+n_{\rm x}^3/6)$. 由于每个粒子均保存着各自的特征地图, 故重抽样的复杂度为$M\times\mathrm{O}(n_{\rm x}^3+n_{\rm x}^3/6)$($M$为特征数). 利用文献中的二叉树结构存储特征, 复杂度有所下降. 通过以上分析可知, 二阶中心差分滤波算法的复杂度小于FastSLAM2.0.

本文中, 直接使用协方差平方根因子$\hat{P}(k)$进行计算, 以始终严格保证协方差矩阵的正定性与对称性. 数值上更加稳定的同时, 使用QR分解, 以使得到的协方差平方根因子为三角方阵, 进而使用高效的回代算法求解SLAM问题中涉及的逆矩阵问题, 从而进一步提高SLAM的求解效率.

5 仿真研究(Simulation result)

仿真测试是在MATLAB2015平台下, 使用悉尼大学野外机器人中心(ACFR)发布的SLAM算法仿真器下展开的. 为进一步了解和比较不同算法之间的性能, 本文采用了Car-like模型作为移动机器人的运动模型, Bearing-Rang激光雷达模型作为环境观测模型, 同时选取了FastSLAM2.0及UFastSLAM算法, 并结合仿真实验和实际数据与本文提出的SOFastSLAM算法进行比较.

图1–2所示为仿真环境图, 其中: 绿色的“※”表

示为特征点; 曲线为小车行驶的参考路径, 即小车从原点开始按照参考路径运行2圈后停止. 小车的前后轴距离为4 m, 传感器的有效距离为40 m, 且具有180°的前视角.

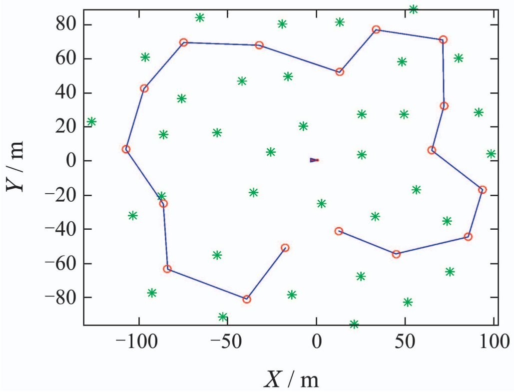

图 1　仿真环境图

Fig. 1　Simulation environment

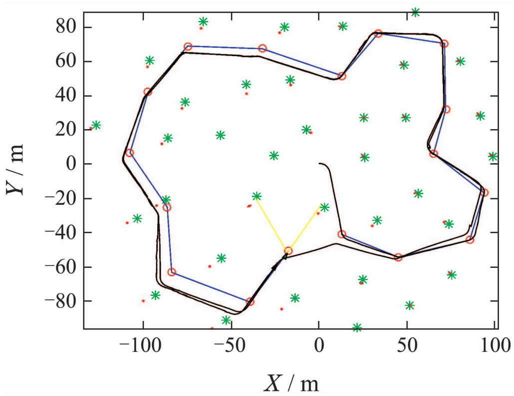

图 2　机器人路径图

Fig. 2　Robot path graph

5.1　不同测量噪声条件下的SLAM算法性能(Performance of SLAM in different measurement noise)

传统的FastSLAM算法中使用的重要性函数并不包含当前的环境观测信息, 因此, 极易造成粒子退化问题. 为验证本文提出的二阶中心差分粒子滤波SLAM算法能有效的避免粒子的退化问题, 在本组合仿真实验中, 作者将粒子的个数均设为20个, 里程计的噪声均设为($\sigma_{\rm v}=0.3$ m/s, $\sigma_{\rm G}=3°$), 使用5组不同的环境噪声: (0. 1 m, 1°), (0. 2 m, 1°), (0. 2 m, 2°), (0.3 m, 3°), (0. 4 m, 5°). 算法的性能评估标准采用均方根误差(root mean square error, RMSE), 定义为

$$\text{RMSE}=\Big(\frac{\sum\limits_{k=0}^{Ns}|x_k-\hat{x}_k|^2}{Ns-1}\Big)^{\frac{1}{2}},\tag{39}$$

式中: x_k为移动机器人在k时刻的真实状态, $\hat{x}_k$为采用SLAM算法获得的估计值.

针对每组既定的噪声水平, 依次使用FastSLAM 2.0算法、UFastSLAM算法和本文提出的SOFastSLAM算法执行50次仿真实验, 并对机器人路径RMSE平均值进行作图(图3)比较.

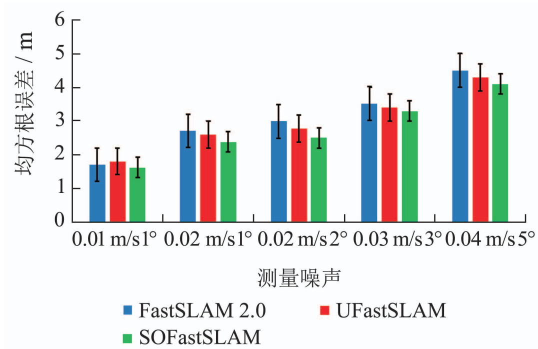

图 3　不同观测噪声水平下SLAM算法的结果比较

Fig. 3　Algorithm comparison in different measurement noise

由图3可知, 随着噪声的增加, 3种算法的RMSE及标准差均逐步增加, 而SOFastSLAM算法的RMSE均值却低于另外2种算法. 由此说明, SOFastSLAM算法的精度高于另外2种算法, 且其标准差的增长速度也明显慢于其它算法. 这意味着SOFastSLAM算法的稳定性较之另外两种算法更高. 此外, 3种算法都避免了粒子在较低的测量噪声水平中的退化问题, 其原因在于3种算法均融合了当前环境的观测信息. 经过预测后的粒子在滤波更新阶段会被吸引到高观测似然区.

5.2　不同粒子数量下的SLAM算法性能(The performance of SLAM with different number of particles)

研究SOFastSLAM算法在粒子数增长的情况下的性能, 在第2组仿真实验中, 里程计的噪声均设为($\sigma_{\rm v}=0.3$ m/s, $\sigma_{\rm G}=3°$), 观测噪声设为($\sigma_{\rm r}=0.2$ m, $\sigma_\theta=6°$). 针对3种不同的算法, 分别用5组粒子数: 1) $M=5$; 2) $M=10$; 3) $M=20$; 4) $M=50$; 5) $M=100$进行实验, 同时在该条件下对FastSLAM 2.0和UFastSLAM算法进行仿真, 并对3种算法中移动机器人轨迹的RMSE误差进行比较, 实验结果如图4 所示.

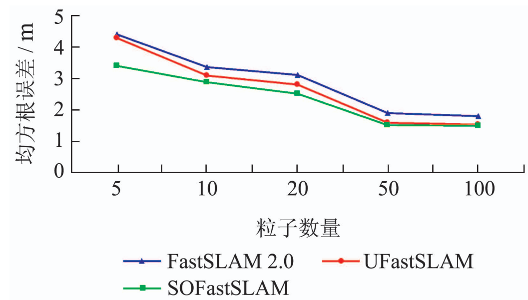

图 4　不同粒子数量下算法RMSE值的对比

Fig. 4　RMSE comparison in different number of particles

由图可知, 随着粒子个数的增加, 3种算法的RMSE

均呈减小趋势, 并逐步趋于稳定. 当粒子数量大于50时, 3种算法的稳定性与精度趋于接近.

6 实验研究(Experiment research)

本次实验是基于澳大利亚悉尼维多利亚公园数据包开展的SLAM算法实验研究[25–26]. 该数据由安装在移动机器人前方的SICK激光在车辆行驶30 min、运行路径超过4 KM的情况下采集得到, 采集地点为具有多树特征的悉尼维多利亚公园. 同时使用GPS传感器捕获机器人的实际运动轨迹, 以评估算法效果(GPS数据并不参与SLAM计算). 图5显示的是移动机器人基于GPS记录轨迹和里程计信息记录的航迹推算轨迹. 考虑到公园建筑物、植被等遮挡物对机器人的干扰, GPS传感器仅能提供间断的定位信息. 与此同时, 由于里程计的测量噪声、车辆的运动学模型不够精确等因素, 导致航迹推算轨迹有所发散, 严重偏离GPS轨迹.

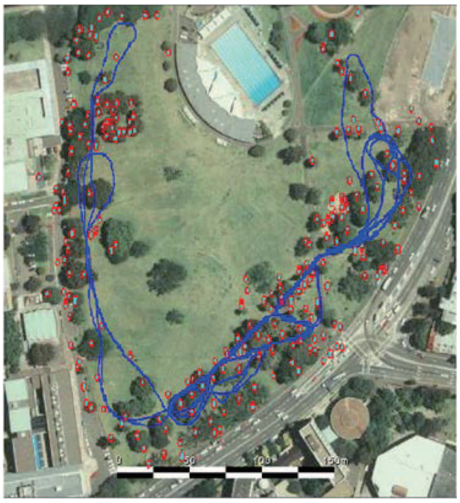

图 5 标准GPS地图

Fig. 5 Standard GPS map

在测试程序中, 本文采用了最大似然分布来进行地图的特征匹配. 基于该数据集, 对SOFastSLAM, FastSLAM2.0, UFastSLAM3种算法展开对比[27–28]. 为了保证实验环境的一致性, 本文在实验中对不同的算法均采用10个粒子来估计小车位置和环境特征位置, 统一设置过程噪声和观测噪声分别为$\sigma_{\rm v}=1$ m/s, $\sigma_{\rm G}=3°$; $\sigma_{\rm r}=0.4$ m, $\sigma_{\theta}=2°$. 为了对SLAM算法的性能进行定性评估, 对上述3种算法均进行10次测试, 并绘制其SLAM轨迹(图6–8), 评价标准参考算法的RMSE平均值和平均计算时间(图9), 同时与校准后的Google地图进行比较(图5).

由图9可知, SOFastSLAM与FastSLAM的计算时间差不多相等, 但是RMSE值却远低于FastSLAM. UFastSLAM的计算时间不仅高于SOFastSLAM, 且计算的精度还低于它.

从以上实验结果可知, 本文所提出的SOFastSLAM算法明显优于FastSLAM2.0和UFastSLAM, 从而进一步验证了本文所提出的SLAM算法的优越性.

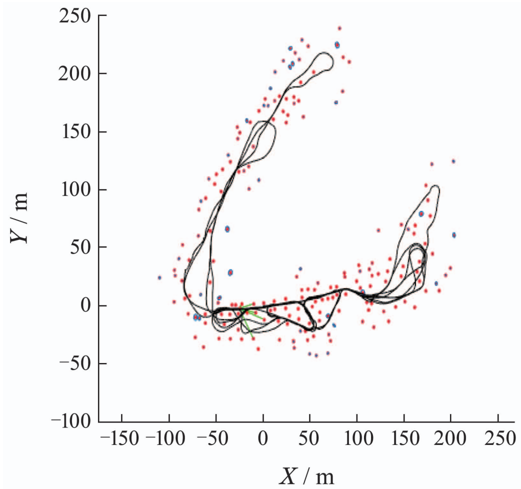

图 6 UFastSLAM测试图

Fig. 6 UFastSLAM test graph

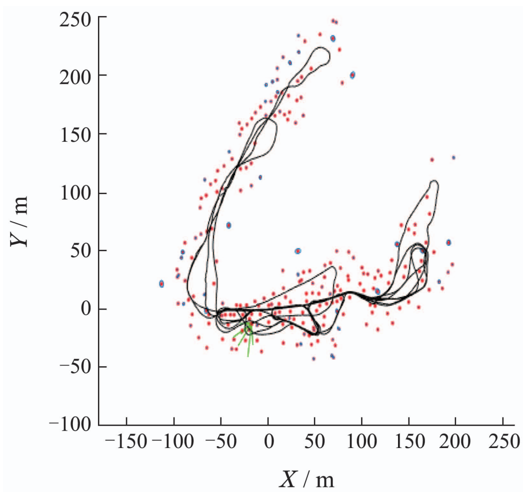

图 7 FastSLAM2.0测试图

Fig. 7 FastSLAM2.0 test graph

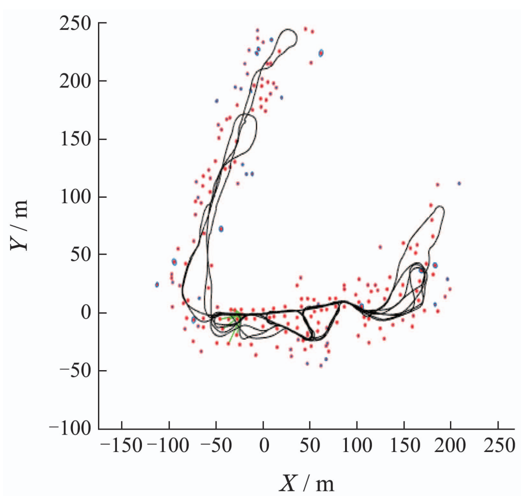

图 8 SOFastSLAM测试图

Fig. 8 SOFastSLAM test graph

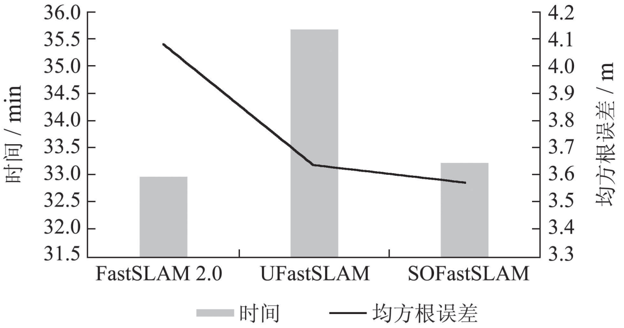

图 9 RMSE与时间对比

Fig. 9 Comparison between RMSE with time

7 结论(Conclusions)

在SLAM问题中, 如何得到与后验分布契合度较高的重要性采样函数是基于粒子滤波SLAM算法的关键. 本文在FastSLAM2.0算法的基础上, 使用性能优越的二阶中心差分粒子滤波法代替原算法中的EKF滤波算法, 显著提高了机器状态估计精度. 与此同时, 通过在机器人位姿概率密度传递过程中使用协方差的平方根因子替代传统协方差进行计算的方式, 避免了对计算数值敏感的协方差矩阵进行平方根操作, 有效的降低了计算量, 简化了计算难度. 通过仿真和实验验证, 证实了SOFastSLAM算法在精度和可靠性方面均优于其他同类算法.

致谢

感谢重庆大学、重庆科华机器人技术研究所所有同学、同事在本论文撰写过程中提供的支持与帮助, 感谢悉尼大学研究者提供的SLAM仿真软件包, 以及维多利亚数据和特征提取算法的程序.

参考文献(References):

[1] THRUN S, LIU Y, KOLLER D, et al. Simultaneous localization and mapping with sparse extended information filters [J]. *International Journal of Robotics Research*, 2010, 23(7/8): 693 – 716.

[2] SMITH R, SELF M, CHEESEMAN P. A stochastic map for uncertain spatial relationships [C] //*The 4th International Symposium on Robotics Research*. Santa Clara: MIT Press, 1988: 467 – 474.

[3] DURRANTWHYTE H, BAILEY T. Simultaneous localization and mapping: part I [J]. *IEEE Robotics & Automation Magazine*, 2006, 13(2): 99 – 110.

[4] BAILEY T, DURRANT W H. Simultaneous localization and mapping (SLAM): part II [J]. *IEEE Robotics & Automation Magazine*, 2006, 13(3): 108 – 117.

[5] 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.

[6] MOUTARLIER P, CHATILA R. An experimental system for incremental environment modelling by an autonomous mobile robot [C] //*International Symposium on Experimental Robotics I*. Montreal: Springer-Verlag, 1989: 327 – 346.

[7] MONTEMERLO M, THRUN S, KOLLER D, et al. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges [C] //*Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence.* Acapulco, Mexico: [s.n.], 2003: 1151 – 1156.

[8] XIN Jing, JIA Weijuan, GOU Jiaolong. Gaussian mixture probability hypothesis density SLAM algorithm [J]. *Journal of Xi'an University of Technology*, 2014, 30(1): 13 – 21.
(辛菁, 贾渭娟, 苟蛟龙. 高斯混合概率假设密度SLAM算法 [J]. 西安理工大学学报, 2014, 30(1): 13 – 21.)

[9] MONTEMERLO M, THRUN S, KOLLER D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem [C] //*Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI Press, 2002: 593 – 598.

[10] GRISETTI G, STACHNISS C, BURGARD W. Improved techniques for grid mapping with rao-blackwellized particle filters [J]. *IEEE Transactions on Robotics*, 2007, 23(1): 34 – 46.

[11] MERWE R. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models* [D]. Portland, OR: Oregon Health and Science University, 2004.

[12] ARASARATNAM I, HAYKIN S. Cubature Kalman filters [J]. *IEEE Transactions on Automatic Control*, 2009, 54(6): 1254 – 1269.

[13] ZHU Qiguang, YUAN Mei, WANG Ziwei, et al. A robot spherical simplex-radial cubature FastSLAM algorithm [J]. *Robot*, 2015, 37(6): 708 – 717.
(朱奇光, 袁梅, 王梓巍, 等. 机器人球面单径容积FastSLAM算法 [J]. 机器人, 2015, 37(6): 708 – 717.)

[14] ZHU Jihua, ZHENG Nanning, YUAN Zejian, et al. A SLAM algorithm based on central difference particle filter [J]. *Acta Automatica Sinica*, 2010, 36(2): 249 – 257.
(祝继华, 郑南宁, 袁泽剑, 等. 基于中心差分粒子滤波的SLAM算法 [J]. 自动化学报, 2010, 36(2): 249 – 257.)

[15] ITO K. Gaussian filter for nonlinear filtering problems [C] //*Proceedings of the 39th IEEE Conference on Decision and Control.* Sydney, NSW: IEEE, 2002: 1218 – 1223.

[16] SCHEI T S. A finite-difference method for linearization in nonlinear estimation algorithms [J]. *Automatic*, 1997, 33(11): 2053 – 2058.

[17] MONTEMERLO M, THRUN S. Simultaneous localization and mapping with unknown data association using FastSLAM [C] //*2003 IEEE International Conference on Robotics and Automation*. Taipei, Taiwan: IEEE, 2003: 1985 – 1991.

[18] BALAKRISHNAN A V. The Kalman filter [J]. *The Mathematical Intelligencer*, 1978, 1(2): 90 – 92.

[19] ZHANG Liang, *Research on the algorithm of mobile robot simultaneous localization and mapping* [D]. Hangzhou: Zhejiang University, 2009.
(张亮. 移动机器人同步定位与地图重建算法研究 [D]. 杭州: 浙江大学, 2009.)

[20] SONG Yu, LI Qingling, KANG Yifei, et al. SLAM with square-root cubature rao-blackwillised particle filter [J]. *Acta Automatica Sinica*, 2014, 40(2): 357 – 367.
(宋宇, 李庆玲, 康轶非, 等. 平方根容积Rao-Blackwillised粒子滤波SLAM算法 [J]. 自动化学报, 2014, 40(2): 357 – 367.)

[21] SHI Yong, HAN Chongzhao. Particle filter using second-order central difference [J]. *Journal of Xi'an Jiaotong University*, 2008, 42(4): 409 – 413.
(石勇, 韩崇昭. 二阶中心差分粒子滤波算法 [J]. 西安交通大学学报, 2008, 42(4): 409 – 413.)

[22] SCHEI T S. A finite-difference method for linearization in nonlinear estimation algorithms [J]. *Automatica,* 2003, 51(10): 2592 – 2601.

[23] GONCHAROV V L. STEFFENSEN J F. Interpolation [J]. *Uspekhi Mat Nauk,* 1936(1): 290 – 292.

[24] WAN E, VAN DER MERWE R, The square-root unscented Kalman filter for state and parameter-estimation [C] //*IEEE International Conference on Acoustics, Speech, and Signal Processing.* Salt Lake City, UT, USA: IEEE, 2001: 3461 – 3464.

[25] ZHAO Lina. *Research and simulation of the FastSLAM2.0 algorithm and PGR navigation algorithm* [D]. Dalian: Dalian Jiaotong University, 2008.
(赵丽娜. FastSLAM2.0控制法则与PGR导航法则的结合研究及其仿真 [D]. 大连: 大连交通大学, 2008.)

[26] SMITH R, SELF M, CHEESEMAN P. Estimating uncertain spatial relationships in robotics [J]. *Machine Intelligence & Pattern Recognition,* 1988, 5(5): 435 – 461.

[27] NETO A D M, ROSA P F F, PELLANDA P C. Environment exploration with visual FastSLAM technique: Experiments and results [C] //*2011 the 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. Budapest: IEEE, 2011: 1 – 7.

[28] KURT-YAVUZ Z, YAVUZ S. A comparison of EKF, UKF, FastSLAM2.0, and UKF–based FastSLAM algorithms [C] //*The 16th International Conference on Intelligent Engineering Systems.* Lisbon: IEEE, 2012: 37 – 43.

作者简介:

代嘉惠 (1985–), 男, 博士研究生, 研究领域为智能机器人系统, E-mail: wjj-123-321@163.com;

许鹏程 (1987–), 男, 硕士研究生, 研究领域为智能机器人系统、嵌入式系统, E-mail: 396340317@qq.com;

李小波 (1974–), 男, 副教授, 研究领域为智能机器人系统、机器人控制, E-mail: 2253755492@qq.com.