

文章编号: 1674-8085(2016)06-0048-07

基于差分进化的 Unscented FastSLAM2.0 算法

吴迎国^{1,2}, *于春梅^{1,2}

(1.西南科技大学信息工程学院, 四川, 绵阳 621010; 2.西南科技大学特殊环境机器人技术四川省重点实验室, 四川, 绵阳 621010)

摘 要: 针对 FastSLAM2.0 算法中重采样过程带来的“粒子耗尽”问题, 将差分进化引入进来, 提出一种基于差分进化的无迹 FastSLAM2.0 算法。首先采用 unscented 粒子滤波器估计机器人的路径后验概率, 然后采用扩展卡尔曼滤波器对环境路标进行估计和更新, 最后引入改进的差分进化算法代替重采样过程来优化粒子。仿真实验表明, 与 FastSLAM2.0 算法相比, 该方法提高了机器人在路径估计和路标估计上的精度, 验证了算法的有效性。

关键词: 同时定位与地图构建; 移动机器人; 粒子滤波器; unscented 卡尔曼滤波器; 差分进化算法

中图分类号: TP242.6

文献标识码: A

DOI:10.3969/j.issn.1674-8085.2016.06.011

FASTSLAM2.0 ALGORITHM BASED ON UNSCENTED DIFFERENTIAL EVOLUTION ALGORITHM

WU Ying-guo^{1,2}, *YU Chun-mei^{1,2}

(1.School of Information Engineering, Southwest University of Science and Technology, Mianyang, sichuan 621010, China;

2.Key Lab. of Robot Technology Used for Special Environment of Sichuan Province, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China)

Abstract: Resampling process often leads to the sample impoverishment problem in FastSLAM2.0. In order to solve the problem, a SLAM method based on differential evolution (DE) is presented by introducing DE's idea into the unscented FastSLAM2.0. Firstly, it estimates the robot path with unscented particle filter, and uses extended Kalman filter to estimate and update the map. Then using the improved differential evolution algorithm replaces resampling process. Simulation results show that: compared with FastSLAM2.0, the presented method improves the accuracy of robot path and the landmark positions estimation. Results show the validity of the proposed algorithm.

Key words: simultaneous localization and map building (SLAM); mobile robot; particle filter; unscented Kalman filter; differential evolution algorithm

0 引言

同时定位与地图构建 (simultaneous localization and mapping, SLAM) 技术是指移动机器人在未知环境中, 使用自身的传感器来感知周围环境构建地图, 同时通过构建好的地图来实现

机器人定位的过程^[1]。由于 SLAM 在理论和应用方面的重要价值, 使之成为了自主导航领域的研究热点, 受到了越来越多的研究者的关注。SLAM 问题是 1988 年由 Smith 和 Cheeseman 首先提出的^[2], 使用扩展卡尔曼滤波器对机器人的位姿和路标的位置进行估计。近些年来, SLAM 问题得

收稿日期: 2016-07-10; 修改日期: 2016-09-23

基金项目: 特殊环境机器人技术四川省重点实验室开放基金项目(13zxtk06)

作者简介: 吴迎国(1991-), 男, 江苏如东人, 硕士生, 主要从事移动机器人定位与构图方面研究(E-mail:15982963080@163.com);

*于春梅(1970-), 女, 江苏如皋人, 教授, 博士, 主要从事故障诊断、模式识别、机器人控制方向的研究(E-mail:yyycm70@qq.com).

到了迅速的发展,出现了各种解法,它的应用领域也得到了进一步的扩大,已经在室内、室外、海下、空中等多种环境中得到了应用。

通常使用概率方法来描述 SLAM 问题,主要针对的是未知且不确定的环境^[3]。目前有两种较常用的方法,一种是基于 EKF 的 SLAM 算法(EKFSLAM),一种是基于粒子滤波的 SLAM 算法(FastSLAM)。基于 EKF 的 SLAM 算法在观测过程中,用一阶泰勒展开式来近似观测模型,线性化过程容易带来误差,使得在非高斯非线性系统中,滤波器的性能下降甚至导致系统的发散,另外,由于 EKF 算法在每个时间点都要进行一次 jacob 矩阵的计算,所以计算过程较为繁琐,这也严重制约了该算法的发展以及进一步的应用^[4-5]。粒子滤波方法是一种基于贝叶斯估计的算法,它使用粒子的概率密度函数逼近状态的概率密度函数,从而达到最优贝叶斯估计,因此可用粒子滤波方法来解决 SLAM 问题。FastSLAM 算法将 SLAM 过程分为移动机器人路径估计与地图构建两部分。机器人的实际位姿使用粒子滤波器来获得,它的位置是用任意概率密度函数来表示的。FastSLAM 算法主要是通过递归的 Rao-Blackwellized 粒子滤波器来估计机器人的位姿,而预测不相关且相互独立的路标是通过扩展卡尔曼滤波器来实现的。对于地图估计中使用 EKF 带来的线性化误差, Kim 等提出了 UFastSLAM 算法^[6],其重点在于利用无迹变换来估计 SLAM 中的转移概率密度。机器人的位姿估计采用 UPF 方法,UPF 将粒子滤波与 UKF 算法结合起来,使用粒子滤波来估计粒子的后验位姿建议分布,而采用 UKF 进行地图的递归估计。

由于使用重采样方法解决粒子滤波器存在的“粒子退化”问题,而导致粒子多样性缺失,进而使得粒子滤波器出现粒子贫乏问题。粒子滤波本质上就是进化的问题,两者有很多相似之处。它们都是通过评价、选择和更新的迭代过程来优化粒子,因此可以用进化算法来改善粒子滤波的性能。由此,提出使用改进的差分进化算法替代重采样过程来优化粒子,使得更多的粒子能够准确地估计机器人的位姿,改善了粒子的多样性,达到减缓粒子退化和多样性缺失等问题的目的。

1 常规的 FastSLAM 算法

1.1 SLAM 问题描述

移动机器人 SLAM 问题的定义:在一个未知环境中,移动机器人从一个未知的起始点出发,当接到向前运动的指令后,里程计开始记录运动数据,外部传感器(如激光传感器和视觉传感器等)开始采集外部信息,获取周围地图。然后通过里程计采集到的数据和外部传感器获取到的环境地图信息增量式的构建环境地图,最后通过构建好的环境地图来实现移动机器人的自定位^[7],这是一个 Markov 链的过程。

从概率角度来说,SLAM 问题是一个求解机器人位姿和环境地图的联合后验概率问题,机器人当前时刻的位姿和环境地图是通过上一时刻机器人位姿和观测信息来估计获得,联合后验概率表示为:

$$p(x_t, m | z_{1:t}, u_{1:t}) \quad (1)$$

式中, x_t 表示 t 时刻移动机器人的位姿; $u_{1:t}$ 和 $z_{1:t}$ 分别表示系统的控制量和测量值; m 表示环境地图。后验概率密度可以表示为:

$$\underbrace{p(x_t, m | z_{1:t}, u_{1:t})}_{t\text{时刻的后验概率分布}} = \underbrace{p(z_t | x_t, m)}_{\text{观测模型}} \underbrace{\int \underbrace{p(x_t | x_{t-1}, u_t)}_{\text{运动模型}} \underbrace{p(x_{t-1}, m | z_{1:t-1}, u_{1:t-1})}_{t-1\text{时刻的后验概率分布}} dx_{t-1}}_{\text{运动模型}} \quad (2)$$

1.2 FastSLAM 算法的基本思想及步骤

FastSLAM 算法的原理是把 SLAM 过程分解为对机器人路径的估计和基于路径估计的环境地图估计^[8-9],具体的过程公式如下所示:

$$p(x_{1:t}, m_t | z_{1:t}, u_{1:t}) = \underbrace{p(x_{1:t} | z_{1:t}, u_{1:t})}_{\text{运动轨迹的估计}} \prod_{j=1}^N \underbrace{p(m_j | x_{1:t}, z_{1:t}, u_{1:t})}_{\text{路径位置的估计}} \quad (3)$$

式中, N 表示地图特征的个数。FastSLAM 算法的每个粒子样本中都包含一份地图信息,用 N 个相互独立的环境特征分解地图估计,每个环境特征都使用一个二维的 EKF 来估计,所以若在 FastSLAM 算法中,有 M 个粒子,则扩展卡尔曼

滤波器的个数为 $M \times N$ 个。 t 时刻第 i 个粒子的表达形式为:

$$S_t^{[i]} = \langle x_{0,t}^{[i]}, \langle \mu_{1,t}^{[i]}, \Sigma_{1,t}^{[i]} \rangle, \dots, \langle \mu_{N,t}^{[i]}, \Sigma_{N,t}^{[i]} \rangle \rangle \quad (4)$$

式中, $x_{0,t}^{[i]}$ 为第 i 个粒子的移动机器人的位姿估计; $\mu_{N,t}^{[i]}$ 和 $\Sigma_{N,t}^{[i]}$ 分别代表在第 i 个粒子的地图中, 第 N 个环境特征的全局坐标和协方差。

常规 FastSLAM 算法总结为以下四个步骤:

1) 采样。根据建议分布 q 采样, 由粒子集 $\{s_{t-1}^{[i]}\}_{i=1,2,\dots,N}$ 产生下一代粒子集 $\{s_t^{[i]}\}_{i=1,2,\dots,N}$ 。其中 N 表示粒子数目。 t 表示时刻, 建议分布则采用机器人的运动模型, 即,

$$q = p(s_t | u_t, s_{t-1}) \quad (5)$$

2) 粒子权重的计算。每个粒子的权重为:

$$\begin{aligned} \omega_t &= \omega_{t-1} \frac{p(z_t | s_t, z^{t-1}) p(s_t | s_{t-1})}{p(z_t | z^{t-1}) p(s_t | s_{t-1})} \\ &\propto \omega_{t-1} p(z_t | s_t, z^{t-1}) \end{aligned} \quad (6)$$

3) 重采样。根据粒子的权重进行采样, 即粒子的权重越大, 则被选取的概率就越高。

4) 更新地图。对于每个粒子, 都用扩展卡尔曼滤波器更新其维护的地图。

2 基于差分进化算法的粒子优化

针对粒子集重采样过程带来的粒子耗尽问题, 引入差分进化算法替代重采样过程来优化粒子集, 使得粒子的多样性和估计精度得到了改善。差分进化算法模拟生物的进化过程, 它是基于种群的全局搜索策略。变异操作是选取种群中两个个体的差向量作为对象, 充分利用了种群中个体之间的互补信息, 使得搜索结果能更好地向全局最优解逼近。因此, 差分进化算法的全局寻优能力和优化结果分布性都比较好。粒子的退化是影响粒子滤波性能的主要原因, 因此可以用差分进化的思想来优化粒子滤波, 从而改善粒子滤波的性能, 提高精度。在粒子滤波完成一步预测后, 差分进化算法根据候选粒子, 利用实数编码生成初始种群。然后通过变异、交叉和选择三个操作,

生成新的提议分布, 最后再更新粒子的权重并重采样。

基于差分进化算法的粒子优化过程如下:

1) 生成初始种群

DE 算法基于实数编码, 根据问题的要求随机生成初始化种群 $m_t^0 = (m_{t,1}^0, m_{t,2}^0, \dots, m_{t,D}^0)$, $i=1,2,\dots,N$, 其适应值为 $f(m_i^0)$ 。其中, D 为维数, N 为种群规模。

2) 变异操作

变异操作是把种群中随机生成的两个个体的差向量, 按一定准则加权并与第三个个体相加, 生成新的变异个体, 变异操作决定了差分进化算法的搜索能力。

差分进化算法的变异方法有好几种, 主要可分为 2 类: 一类是随机选取基准粒子, 此时算法的收敛速度较慢, 但全局收敛较优; 另一类是选择当前种群中适应度值最大的粒子为基准粒子, 此时虽然收敛速度快, 但是易陷入局部最优。为此, Zhu 等^[10-11]提出了一种新的变异策略, 该策略受粒子群优化算法的启发, 如下所示:

$$U_i = X_i + F_1(X_i - X_{r1}) + F_2(X_{best} - X_i) \quad (7)$$

式中, X_{best} 为当前种群中的适应度值最大的粒子; F_1 为 $[-1,1]$ 之间的随机数; F_2 为 $[0,1.5]$ 之间的随机数, 由于受最有粒子的引导, 该变异策略能保证寻优能力的前提下以更快的速度使算法收敛。

3) 交叉操作

差分进化算法采用的是离散交叉算子。利用式(8)将通过变异操作得到的变异向量和目标向量交叉, 得到测试向量。引入下述交叉操作:

$$V_i = \begin{cases} U_i, \text{rand}(0,1) \leq CR \\ X_i, \text{否则} \end{cases} \quad (8)$$

式中, $CR \in [0,1]$ 为交叉概率。 $\text{rand}(0,1)$ 是 $[0,1]$ 区间内均匀分布的随机数。

4) 选择操作

将变异交叉后的粒子与父代粒子进行对比, 新产生的粒子只有优于父代粒子才能被保留下来。本文选取粒子的权重 ω_i 作为评价函数, 即按

下式进行选择操作产生子代粒子:

$$X_i = \begin{cases} V_i, f(V_i) > f(X_i) \\ X_i, \text{否则} \end{cases} \quad (9)$$

如果接受的新粒子数目小于 $N/2$, 则继续按照式(7), 式(8)和式(9)进行变异, 交叉和选择操作产生新粒子, 直到有效的粒子总数达到 $N/2$, 再把它们的权值都重新设为 $1/N$, 并将新粒子加入优化粒子集中。

3 DE-UFastSLAM 算法的具体实现

过程

FastSLAM 算法将 SLAM 过程分解为机器人的路径估计和基于机器人路径估计的地图估计。

3.1 路径估计

重要性采样是从建议分布中采用, 而不能从目标后验概率分布中直接采样。FastSLAM2.0 为了提高算法的估计精度将最新观测值加入提议分布中, 但是雅克比矩阵和线性化过程带来的误差依然存在。而 UPF 近似状态变量的后验概率密度函数的效果比较好, 因此本文使用 UPF 方法递归 f 进行路径估计。UPF 方法是粒子滤波和 UKF 方法的结合, 对普通粒子滤波算法的重要密度函数选取进行了改进。它的基本思想是将最新的观测值放入 Unscented 变换并生成非线性粒子滤波的建议分布, 其具体的过程如下所示:

扩充并初始化第 i 个粒子中机器人的状态向量的均值和方差。

$$X_{t-1}^{[i]} = \begin{bmatrix} X_{t-1}^{[i]} \\ 0 \\ 0 \end{bmatrix}, \quad P_{t-1}^{[i]} = \begin{bmatrix} P_{t-1}^{[i]} & 0 & 0 \\ 0 & Q_t & 0 \\ 0 & 0 & R_t \end{bmatrix} \quad (10)$$

式中 $X_{t-1}^{[i]}$ 为在 $t-1$ 时刻第 i 个粒子中扩充后机器人的初始状态向量, 相应的协方差矩阵为 $P_{t-1}^{[i]}$ 。 Q_t 和 R_t 分别为过程噪声协方差矩阵和观测噪声协方差矩阵。

DE-UFastSLAM 使用对称采样策略采样 Sigma 点, 利用非线性函数对 Sigma 点进行处理, 把状态向量维数扩充为 $L=7$, 因此采样后为 $2L+1$ 个 Sigma 点。Sigma 点集为:

$$\chi_{t-1}^{a[0][i]} = X_{t-1}^{a[i]}$$

$$\chi_{t-1}^{a[j][i]} = X_{t-1}^{a[i]} + (\sqrt{(L+\lambda)P_{t-1}^{a[i]}})_j, \quad j=1, \dots, L \quad (11)$$

$$\chi_{t-1}^{a[j][i]} = X_{t-1}^{a[i]} - (\sqrt{(L+\lambda)P_{t-1}^{a[i]}})_{j-L}, \quad j=L+1, \dots, 2L$$

其中 $(\sqrt{(L+\lambda)P_{t-1}^{a[i]}})_j$ 表示取均方根矩阵的第 j 列, $\lambda = \alpha^2(L+\kappa) - L$, 通常 $\alpha(0 < \alpha < 1)$ 能够避免在高度非线性状况下采样时的非局部影响。加入 $\kappa(\kappa \geq 0)$ 来使得协方差矩阵的半正定性得到保证。每个 Sigma 点都由状态、控制和观测三部分组成, 如下所示:

$$\chi_{t-1}^{a[j][i]} = [\chi_{t-1}^{[j][i]} \quad \chi_t^{U[j][i]} \quad \chi_t^{z[j][i]}]^T \quad (12)$$

状态预测值:

$$\bar{\chi}^{[j][i]} = f(U_t^{[i]} + \chi_t^{U[j][i]}, \chi_{t-1}^{[j][i]}) = \begin{bmatrix} \bar{\chi}_{x,t}^{[j][i]} \\ \bar{\chi}_{y,t}^{[j][i]} \\ \bar{\chi}_{\theta,t}^{[j][i]} \end{bmatrix} \quad (13)$$

状态预测均值:

$$X_{t|t-1}^{[i]} = \sum_{j=0}^{2L} w_m^{[j]} \bar{\chi}^{[j][i]} \quad (14)$$

状态预测误差矩阵:

$$P_{t|t-1}^{[i]} = \sum_{j=0}^{2L} w_c^{[j]} (\bar{\chi}_t^{[j][i]} - X_{t|t-1}^{[i]})(\bar{\chi}_t^{[j][i]} - X_{t|t-1}^{[i]})^T \quad (15)$$

式中 $w_m^{[j]}$ 和 $w_c^{[j]}$ 分别为均值加权所对应的权值和协方差加权对应的权值:

$$w_m^{[j]} = \begin{cases} \lambda / (n + \lambda), & j=0 \\ 1 / 2(n + \lambda), & j \neq 0 \end{cases} \quad (16)$$

$$w_c^{[j]} = \begin{cases} \lambda / (L + \lambda) + (1 - \alpha^2 + \beta), & j=0 \\ 1 / 2(L + \lambda), & j \neq 0 \end{cases} \quad (17)$$

式中 β 是用来引入后验概率分布高阶信息项的参数。对于高斯分布, $\beta=2$ 。

观测量预测值:

$$\bar{\gamma}_t^{[j][i]} = h(\bar{\chi}_t^{[j][i]}, \mu_{k,t-1}^{[i]}) + \chi_t^{z[j][i]} \quad (18)$$

观测量预测均值:

$$\hat{z}_t^{[i]} = \sum_{j=0}^{2L} w_m^{[j]} \bar{\gamma}_t^{[j][i]} \quad (19)$$

信息协方差:

$$S_t^{[i]} = \sum_{j=0}^{2L} w_c^{[j]} (\bar{\gamma}_t^{[j][i]} - \hat{z}_t^{[i]})(\bar{\gamma}_t^{[j][i]} - \hat{z}_t^{[i]})^T \quad (20)$$

状态与观测互协方差:

$$\Sigma_t^{X,z[i]} = \sum_{j=0}^{2L} w_c^{[j]} (\bar{\chi}_t^{[j][i]} - X_{t|t-1}^{[i]})(\bar{\gamma}_t^{[j][i]} - \hat{z}_t^{[i]})^T \quad (21)$$

卡尔曼增益:

$$K_t^{[i]} = \Sigma_t^{X,z[i]} \bullet (S_t^{[i]})^{-1} \quad (22)$$

滤波值:

$$X_t^{[i]} = X_{t|t-1}^{[i]} + K_t^{[i]}(z_t - \hat{z}_t^{[i]}) \quad (23)$$

滤波误差矩阵:

$$P_t^{[i]} = P_{t|t-1}^{[i]} - K_t^{[i]} S_t^{[i]} (K_t^{[i]})^T \quad (24)$$

3.2 地图估计

每个粒子中都包含有地图信息, 将地图估计分解为 N 个特征估计问题, 假定地图中的每个环境特征均服从独立的二维高斯分布, 则雅克比矩阵为 2×2 的矩阵, 它的计算量并不大, 因此, EKF 较 UKF 来说计算优势更明显。所以, 这里还是用 EKF 来对环境路标进行更新。

3.3 算法实现过程

与常规的 FastSLAM2.0 算法相比, 本文算法有两处改进: 一是在计算各个粒子后验概率的过程中用 UKF 替代 EKF, 二是用改进的差分进化算法来取代重采样过程来优化粒子。详细的实现过程如下:

1) 采样机器人的位姿。利用 UKF 计算各个粒子的后验位姿建议分布, 从建议分布中采样。

2) 用式(6)计算各个粒子的非正则权重, 并归一化。

3) 用 EKF 更新各个粒子的地图。

4) 粒子优化。计算粒子集的有效粒子数

$$N_{eff} = 1 / \sum_{i=1}^N (w_k^i)^2, \text{ 若 } N_{eff} < 2/N, \text{ 则采用文中改}$$

进的差分进化算法来对粒子集 $\{x_{1:k}^i, w_k^i\}_{i=1}^N$ 进行优化, 得到新的粒子集 $\{x_{1:k}^i, 1/N\}_{i=1}^N$ 。

4 实验分析

为了验证方法的正确性和有效性, 我们利用 Tim Bailey 公开的仿真环境, 对文中提出的基于差分进化算法的移动机器人 SLAM 方法进行仿真实验, 并与传统的 FastSLAM2.0 算法进行比较。

仿真环境如图 1 所示, 其中绿色星号表示设定路标, 蓝色曲线表示移动机器人预先设定好的路径轨迹。图中共有 35 个设定好的路标, 它们对于移动机器人来说都是未知的。在仿真中用到的运动模型和观测模型如式(25)和式(26)。机器人的移动速度为 3 m/s, 最大转向角为 30° , 轴距为 4 m, 激光雷达最大观测范围为 30 m, 控制时间间隔为 0.025 s。

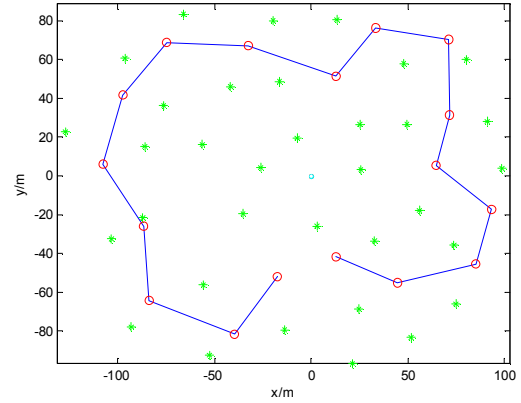


图 1 仿真环境

Fig.1 The simulation environment

运动模型如下:

$$\begin{bmatrix} x_t \\ y_t \\ \varphi_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + V_t \cdot \Delta t \cdot \cos(\varphi_{t-1} + \gamma) \\ y_{t-1} + V_t \cdot \Delta t \cdot \sin(\varphi_{t-1} + \gamma) \\ \varphi_{t-1} + \frac{V_t \cdot \Delta t}{B} \cdot \sin(\gamma) \end{bmatrix} + \delta_t \quad (25)$$

其中 B 为机器人的轴距, Δt 为时间间隔, $u = [V \ \gamma]^T$ 表示运动模型的控制输入, 输入量为速度与转向角, δ_t 为运动高斯噪声。

观测模型如下:

$$\begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2} \\ \tan^{-1} \frac{y_i - y_t}{x_i - x_t} - \varphi_t \end{bmatrix} + \varepsilon_i \quad (26)$$

其中 ε_i 表示观测噪声, 其服从高斯分布。

在仿真环境中分别采用 FastSLAM2.0 算法和 DE-UFastSLAM 算法获得的仿真结果分别如图 2, 图 3 所示。这里粒子数设定为 10 个, 其中红色实心点表示该方法构建的路标, 红色实线表示机器人的估计路径轨迹, 黑色实线表示机器人的实际路径。

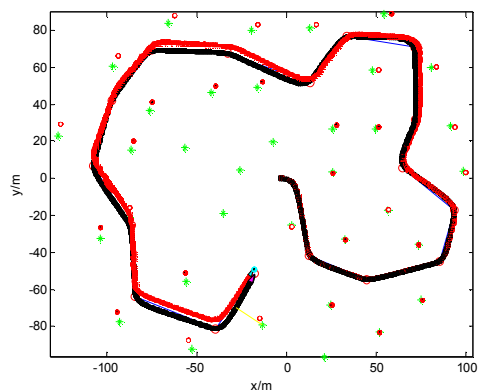


图2 FastSLAM2.0 算法的仿真结果

Fig.2 Simulation result of FastSLAM2.0

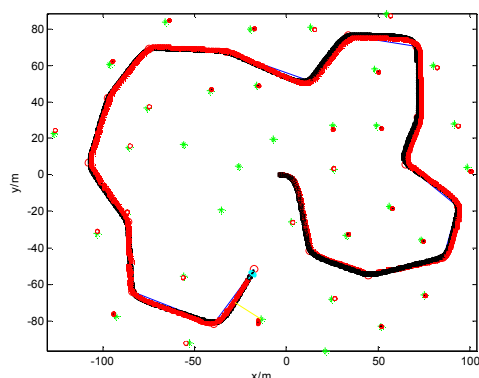


图3 DE-UFastSLAM 算法的仿真结果

Fig.3 Simulation result of DE-UFastSLAM

由图3可以看出,与FastSLAM2.0算法相比,DE-UFastSLAM算法的估计路标与设定路标契合度更高。然后分别对这两种方法的机器人位姿和路标位置估计进行比较,图4、图5为两种算法横、纵坐标估计误差曲线,图6为路标位置的估计误差。

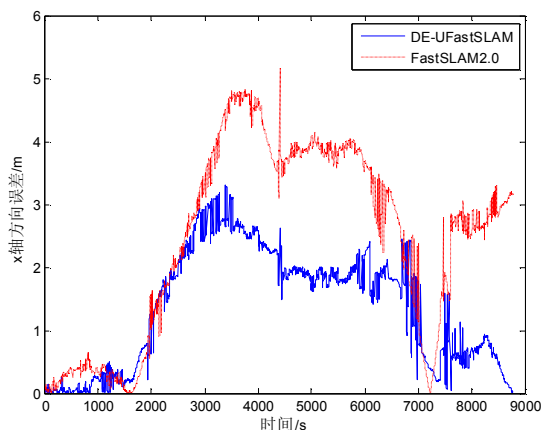
图4 x 轴方向上的位姿估计误差

Fig.4 Position estimation error of the x axis

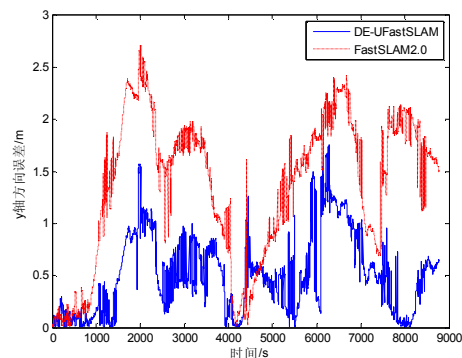
图5 y 轴方向上的位姿估计误差

Fig.5 Position estimation error of the y axis

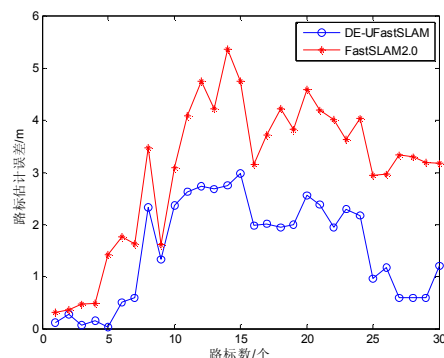


图6 路标位置估计误差

Fig.6 Position estimation error of landmark

从图3,图4很容易看出,DE-UFastSLAM算法对机器人位姿的估计误差比FastSLAM2.0算法小。在路标估计方面,DE-UFastSLAM算法也明显有更好的估计精度。

5 结语

针对FastSLAM2.0算法在重采样过程中产生的“粒子耗尽”问题,本文提出采用改进的差分进化算法替代重采样过程,使得大权值粒子参与到进化过程中,而不是通过简单的复制,保证了小权值粒子也能参与到SLAM过程中,使得权值较小粒子中的路径和地图信息保留下来,同时保证了粒子集的优良性能和粒子的多样性。在路径估计过程中,使用UPF方法计算建议分布,达到无需计算雅克比矩阵、减小SLAM线性化过程中的积累误差、提高算法的计算精度的目的。通过理论分析和仿真实验表明,与FastSLAM2.0算法相比,本文提出的算法在粒子数相同的情况下,定位和构图都更精确。

参考文献

- [1] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping - Part I: the essential algorithms [J]. Robotics&Auto-mation Magazine,2006,13(2): 99-108.
- [2] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[J]. Uncertainty in Artificial Intelligence, 1988, 2:435-461.
- [3] Thrun S, Burgard W, Fox D. A probabilistic approach to concurrent mapping and localization for mobile robots[J]. Machine Learning, 1998, 31(1-3): 29-53.
- [4] Holmes S, Klein U, Murray D W. An $O(N^2)$ square root unscented Kalman filter for visual simultaneous localization and mapping[J]. IEEE Trans.on Pattern Analysis and Machine Intelligence,2009,31(7): 1251-1263.
- [5] Hwang S Y, Song J B. Monocular visior-based SALM in indoor environment using corner,lamp,and door features from upward-looking camera[J].IEEE Trans on Industrial Electornics,2011,58(10):4804-4812.
- [6] Kim C, Sakthivel R, Chung W K. Unscnted FastSLAM: a robust and efficient solution to the SLAM problem[J]. IEEE Transactions on Robotics, 2008, 24(4): 808-820.
- [7] 陈卫东,张飞. 移动机器人的同步自定位与地图创建研究进展[J].控制理论与应用.2005, 22(3): 455-460.
- [8] Thrun S, Montemerlo M, Koller D, et al. FastSLAM: an efficient solution to the simultaneous localization and mapping problem with unknown data association[J]. Machine Learning, 2004,4(3):380-407.
- [9] 汤文俊,张国良,敬斌. 移动机器人FastSLAM算法的对比研究[J].计算机工程与设计,2003, 33(3):1165-1169, 1180.
- [10] Zhu G P,Kwong S. Gbest guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation,2010,217(7): 3166-3173.
- [11] 曹洁,李玉琴,吴迪. 基于改进差分进化的高精度粒子滤波算法[J].上海交通大学学报: 自然科学版,2014,12: 1714-1720.