

文章编号:1009-3087(2013)01-0146-06

粒子群优化人工免疫粒子滤波器

杜正聪^{1,2},冯大海^{2,3},牛高远^{2,3}

(1. 电子科技大学 电子工程学院,四川 成都 610054;2. 攀枝花学院,四川 攀枝花 617000;3. 西华大学 电气信息学院,四川 成都 610039)

摘要:为解决粒子滤波算法中存在的粒子退化和样本枯竭问题,提出一种新的粒子滤波算法。利用粒子群优化思想促使采样粒子向高似然区域移动,减缓粒子权值的退化;再通过人工免疫算法中的变异操作扩大算法寻找最优值的范围并增加粒子的多样性,避免算法陷入局部最优,增强算法的全局搜索能力,进而缓解样本枯竭。实验表明,该算法比标准粒子滤波的状态估计精度提高近40倍,比扩展卡尔曼粒子滤波提高近28倍,比无迹卡尔曼粒子滤波提高近6倍,滤波效率为37.523%,是标准粒子滤波的37倍,该算法具有更好的实时性和更高的状态估计精度,能有效缓解粒子的退化和样本的枯竭。

关键词:粒子群优化;人工免疫;粒子滤波

中图分类号:TN957.51

文献标志码:A

Particle Swarm Optimization Artificial Immune Particle Filter

DU Zheng-cong^{1,2}, FENG Da-hai^{2,3}, NIU Gao-yuan^{2,3}

(1. College of Electronic Eng., Univ. of Electronic Sci. and Technol. of China, Chengdu 610054, China; 2. Panzhihua Univ., Panzhihua 617000, China; 3. College of Info. and Electronic, Xihua Univ., Chengdu 610039, China)

Abstract: To solve the particle weights degradation and sample impoverishment problems in particle filter which was used in non-Gaussian nonlinear systems, a new kind of particle filter was proposed. The sampling particles could move to high likelihood area by using particle swarm optimization idea, and the weights of degradation of the particles were reduced. Then through the variation operation of the artificial immune, the range of the optimal value was expanded when searching for the optimization, and the diversity of the particles was increased. Simulation results showed that the state estimation accuracy of new algorithm is improved nearly 40 times than standard particle filter. The filtering efficiency is 37.523%, 37 times of the standard particle filter. This algorithm has better real-time and state estimation precision, and could effectively relieve the particle of the weights of the exhaustion of sample degradation.

Key words: particle swarm optimization; artificial immune; particle filter

非线性、非高斯系统的状态估计在信号处理、自动控制、机器学习、目标跟踪、模式识别、无线通信等领域有着广泛的应用,对该系统传统的状态估计方法主要是扩展卡尔曼滤波和修正增益的扩展卡尔曼滤波算法^[1],但这种通过对非线性模型进行1阶泰勒级数展开的近似方法在估计系统状态和方差时存在很大的线性误差。近年来,粒子滤波由于在状态估计中适用范围广、处理能力强、估计精度高等优

点已被广泛应用于各个领域。

粒子滤波(particle filter, PF)最早被提出于20世纪50年代末,但由于计算量大和粒子权值退化严重,一直未得到学术界的重视,直到1993年Gordon^[2]等提出在递推过程中进行重采样,才出现了第一个可以操作的粒子滤波器Bayesian bootstrap filter,这才奠定了粒子滤波算法的基础。粒子滤波权值退化的根源在于取样函数不是真实的后验概率分布,重采样的引入虽能缓解权值退化问题,但同时也导致了样本的贫化。为解决这些问题,许多的学者提出了很多的改进方法。石勇^[3]等利用Stirling插值公式提出了2阶中心差分粒子滤波,但计算过于复杂;杜正聪^[4]等将状态向量分解为转移先验和后

收稿日期:2012-08-26

基金项目:四川省青年科技基金资助项目(2010JQ0041);四川省应用基础研究项目(2011JY0115)

作者简介:杜正聪(1975—),男,教授,博士。研究方向:非线性、非高斯信号处理。

验概率密度两部分,以此作为建议分布函数,提出了混合退火粒子滤波器,但适用范围必须是状态多维且可分解;宁小磊^[5]等利用区间滤波生成重要性密度函数提出了区间衍生粒子滤波器,虽然精度有所提高,但计算量明显增加,算法实时性不好;张琪^[6]等提出了基于权值选择的粒子滤波方法,虽然在一定程度上减缓了权值退化,但采样粒子过多,算法中要分别计算每个粒子的权值并选出权值大的 N 个粒子,计算量过大;邹国辉^[7]等提出了基于优化重采样的粒子滤波算法,通过选取粒子和抛弃粒子的某种线性组合来产生新的粒子,粒子多样性缺失得到缓解,但线性组合中的步长系数 K 选取困难;程水英^[8]等通过在重采样前利用权值高的粒子进行裂变繁殖并以此来取代粒子集中的低权值粒子,提出了裂变自举粒子算法,虽缓解了样本的贫化,但算法中的门限值 N_{th} 对不同的模型设置困难,过大则会导致运算量的增加,过小则不能很好地解决算法中的样本枯竭问题;方正^[9]等利用粒子群优化思想具有全局寻优和快速收敛的特点提出了粒子群优化粒子滤波方法,状态估计的精度有所提高,但容易使粒子集合陷入局部最优,算法的鲁棒性较差;张琪^[10]等将人工免疫算法引入粒子滤波,提出了人工免疫粒子滤波算法的研究,经过多次运算和变异操作选出所需的权值较大的 N 个粒子,虽然增加粒子的多样性,但算法实时性较差。

作者在上述研究基础之上,为使重要性分布函数尽可能和真实后验概率分布相似,提出一种粒子群优化人工免疫粒子滤波算法。该算法从优化的角度出发,利用粒子群优化 (particle swarm optimization, PSO) 思想具有全局寻优和快速收敛的优点使 PF 的采样粒子向高似然区域移动,缓解粒子权值的退化,不同于文献[10]中的对权值大的粒子进行简单的克隆和复制操作来缓解粒子权值的退化,然后通过人工免疫算法中的变异操作来增加粒子的多样性,扩大算法的寻优范围,避免算法出现早熟现象。

1 问题描述和粒子滤波

1.1 问题描述

一般非线性、非高斯系统常用动态状态空间模型 (dynamic state space models, DSSM) 来描述, DSSM 由状态方程和观察方程构成。

状态方程:

$$\mathbf{x}_n = f(\mathbf{x}_{n-1}, \mathbf{d}_n)。$$

观测方程:

$$\mathbf{y}_n = h(\mathbf{x}_n, \mathbf{v}_n)。$$

其中, $\mathbf{x}_n \in \mathbf{R}^{n_x}$ 为系统在 n 时刻的状态向量, $\mathbf{y}_n \in \mathbf{R}^{n_y}$ 为系统状态为 \mathbf{x}_n 时的观测向量, \mathbf{d}_n 、 \mathbf{v}_n 分别为独立同分布 (independently identical distribution, i. i. d) 的过程噪声和观测噪声。 $f: \mathbf{R}^{n_x} \cdot \mathbf{R}^{n_d} \rightarrow \mathbf{R}^{n_x}$ 和 $h: \mathbf{R}^{n_x} \cdot \mathbf{R}^{n_v} \rightarrow \mathbf{R}^{n_y}$ 分别为系统状态转移函数和观测函数。 $p(\mathbf{x}_0)$ 已知, 假定 \mathbf{x}_n 服从 1 阶 Markov 过程, 观测序列 \mathbf{y}_n 相互独立, 令 $\mathbf{x}_{0:n} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}$, $\mathbf{y}_{1:n} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$, 最优贝叶斯估计是为了递归地求出系统状态的概率分布 $p(\mathbf{x}_n | \mathbf{y}_{1:n})$, 当求出了边缘概率密度分布后就可以对系统状态进行最优估计。

1.2 粒子滤波

粒子滤波又叫序贯蒙特卡洛滤波,它是通过寻找 1 组在状态空间中传播的随机样本来近似后验概率密度函数,利用样本的均值来代替积分运算,从而获得状态的最小方差估计。

假设能够从后验概率密度函数 $p(\mathbf{x}_n | \mathbf{y}_{1:n})$ 中抽取 N 个 i. i. d 的样本 $\{\mathbf{x}_n^i\}_{i=1}^N$, 则状态的 pdf 可以表示为 $p(\mathbf{x}_n | \mathbf{y}_{1:n}) \approx \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_n - \mathbf{x}_n^i)$, 函数 $g(\mathbf{x}_n)$ 的期望则可以近似为 $g(\mathbf{x}_n) \approx \frac{1}{N} \sum_{i=1}^N g(\mathbf{x}_n^i)$ 。由于粒子集合 $\{\mathbf{x}_n^i\}_{i=1}^N$ 相互独立, 当 $N \rightarrow \infty$ 时, 根据大数定理可知, 函数 $g(\mathbf{x}_n)$ 的近似值 $g(\mathbf{x}_n)$ 将以概率 1 收敛于真实期望值 $\bar{g}(\mathbf{x}_n)$ 。

2 粒子群优化与粒子群优化粒子滤波

2.1 粒子群优化

粒子群优化算法^[11]最初是由 Kennedy 等在 1995 年为模仿鸟群和鱼群等动物的群集行为提出的一种群体智能型算法, 由于其算法设计简单, 参数设置容易, 且是一种有效的全局寻优算法, 目前已经得到了很多学者和专家的重视。 PSO 是一种基于种群间的合作和竞争的全局寻优智能算法, 它将种群中的每个个体 $x_{i,j}$ 看作是 n 维空间中具有重量和体积的微粒, 每个微粒都有自己的初始位置 $x_{i,j}(0)$ 和速度 $v_{i,j}(0)$, 在每次的迭代过程中, 微粒的飞行速度通过自身的飞行经验和群体的飞行经验进行动态调整, 微粒通过 2 个极值来更新自己的速度和位置^[10], 一个是个体极值 $\mathbf{P}_i = (P_{i,1}, P_{i,2}, \dots, P_{i,n})$, 一个是全局极值 $\mathbf{P}_g = (P_{g,1}, P_{g,2}, \dots, P_{g,n})$, 更新公式如下:

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot r_{1,j} \cdot (P_{i,j} - x_{i,j}(t)) + c_2 \cdot r_{2,j} \cdot (P_{g,j} - x_{i,j}(t)) \quad (1)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (2)$$

其中, j 表示维数, i 表示第 i 个粒子, $r_{1,j} \sim u(0,1)$, $r_{2,j} \sim u(0,1)$, $P_{i,j}$ 表示第 i 个粒子从初始到当前迭代过程中搜索产生的个体极值, $P_{g,j}$ 为种群中每一维的最优值, 称为全局极值, t 为迭代次数, c_1 、 c_2 为加速因子, 一般取 $0 \sim 2$, $r_1 \sim U(0,1)$, $r_2 \sim U(0,2)$ 。经过上面的多次迭代, 最终促使微粒趋于能够使函数适应值处于高似然概率区域附近。

2.2 粒子群优化粒子滤波

由于 PSO 算法和粒子滤波有许多的共同点^[9], 一些学者们为改善粒子滤波的性能, 把 PSO 思想引入 PF 提出了许多的基于 PSO 的粒子滤波算法, 但在经典的粒子群优化粒子滤波 (particle swarm optimization particle filter, PSO-PF) 算法中, 由于 PSO 算法的最大速度等参数很难确定, 文献[12]中提出一种改进的粒子群优化算法, 用一个高斯分布来更新粒子的速度, 其精度和收敛效果明显好于最初的粒子群优化粒子滤波算法。改进的 PSO-PF 算法具体过程如下:

步骤 1: 当得到某一时刻的观测值后, 定义该时刻的适应度函数为:

$$fitness = \exp \left[-\frac{1}{2R_k} (y_{new} - y_{pred})^2 \right] \quad (3)$$

步骤 2: 对该时刻的采样粒子进行速度和位置的更新, 使粒子集合向最优粒子靠拢, 经过多次迭代, 当粒子群分布在真实状态附近, PSO 停止优化, 更新方式为:

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + |randn| \cdot (P_{i,j} - x_{i,j}(t)) + |randn| \cdot (P_{g,j} - x_{i,j}(t)) \quad (4)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (5)$$

其中, $|randn|$ 为正的高斯分布随机数。通过上面的操作, 可以使粒子群体向最优粒子靠近, 粒子集合的权值在观测更新前更加趋于高似然区域, 在一定程度上缓解了粒子权值的退化现象。

步骤 3: 利用最新的观测值计算该时刻每个粒子相应的权值并归一化处理。

$$w_k^i = w_{k-1}^i P(y_k | x_k^i) \quad (6)$$

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^N w_k^i} \quad (7)$$

步骤 4: 重采样, 复制权值大的粒子, 去除权值小的粒子。

步骤 5: 状态估计输出。

尽管取样函数并不是真实的后验概率分布函数, 但通过上面的 PSO 优化过程, 能使采样粒子在权值更新前更加趋于高似然区域, 提高了每个粒子的作用效果, 一定程度上缓解了粒子权值的退化, 但 PSO 在优化过程中容易陷入局部最优, 算法精度不高。

3 人工免疫及粒子群优化人工免疫粒子滤波

3.1 人工免疫算法

人工免疫 (artificial immune, AI) 算法是一种模拟生物免疫系统的优化方法^[13], 抗原和抗体是构成人工免疫系统的重要组成部分。当用 AI 求解优化问题时, 常把抗原对应于待求解的问题, 抗体对应于问题的一个解, 待求解问题的目标函数刻画了抗原于抗体的匹配程度, 通常用亲和力来描述。AI 算法先对亲和力大的抗体进行克隆生成一个临时的克隆种群, 克隆数目与抗体-抗原的亲和力大小成正比; 然后对克隆种群中的每个抗体发生变异, 变异的概率与亲和力大小成反比; 再次计算克隆种群中抗体的亲和力大小, 选出所需的 N 个抗体作为下一代的抗体种群。

人工免疫算法不但能保证亲和力大的抗体存在, 还能在亲和力大的抗体基础之上进行变异, 这样不但能增加抗体的多样性, 还能扩大亲和力更大的抗体存在的可能性, 使算法找到更优抗体。

3.2 粒子群优化人工免疫粒子滤波

尽管 PSO-PF 算法是一种有效的全局寻优算法, 但粒子集合在追寻种群最优粒子的过程中, 所有粒子都向同一个方向移动, 粒子群体表现出强烈的趋同性, 极易使微粒集合陷入局部次优位置附近, 出现早熟现象进而导致算法精度不高。为此, 利用 PSO 具有全局寻优和快速收敛的优点, AI 算法能扩大全局寻优范围和增加粒子多样性的优点, 提出一种新的优化粒子滤波算法——粒子群优化人工免疫粒子滤波 (PSOAI-PF)。利用 PSO 方法使粒子滤波算法中的采样粒子向高似然概率靠近, 减缓粒子退化, 通过人工免疫中的变异操作来扩大 PSO 的寻优范围, 防止 PSO 在寻优过程中陷入局部次优位置, 提高估计精度。

为减小计算量, 算法中引入群体适应度标准差 a ^[14],

$$a = \sqrt{\frac{1}{N} \sum_{i=1}^N (f_i - f_{average})^2} \quad (8)$$

其中, $f_i = w_k^i \cdot f_{\text{average}} = \frac{1}{N} \sum_{i=1}^N w_k^i$, 实验表明, a 值过小, 则粒子集合过于集中, 很难保证粒子的多样性, 还可能使粒子集合陷入局部次优位置附近, 状态估计的精度反而降低。所以当 $a < 1/N$ 时才对粒子滤波算法进行粒子群优化人工免疫优化过程。 a 的引入, 不但能减少计算量, 还能使粒子集合分布在一个恰当的位置。

PSOAI-PF 算法具体实现步骤如下:

步骤1: 初始化, 在 $k = 0$ 时, 从 $p(x_0)$ 分布中采样 N 个粒子得到初始粒子集合 $\{x_0^i, i = 1, \dots, N\}$, 令 $k = 1$;

步骤2: 从建议分布函数 $q(x_k | x_{0:k-1}, z_{0:k})$ 中采样 N 个粒子 $\{x_k^i, i = 1, \dots, N\}$, 计算相应的权值并归一化;

$$w_k^i = \frac{w_{k-1}^i p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{0:k-1}^i, z_{1:k})} \quad (9)$$

$$w_k^i = w_k^i / \sum_{i=1}^N w_k^i \quad (10)$$

步骤3: 计算粒子群体适应度标准差 a , 如果 a 大于 $1/N$, 则进入步骤6, 否则进行步骤4;

$$a = \sqrt{\frac{1}{N} \sum_{i=1}^N (f_i - f_{\text{average}})^2} \quad (11)$$

步骤4: 进入 PSO 算法过程;

1) 初始化该时刻粒子集合的个体极值 $P_p = \{w_k^i\}_{i=1}^N$ 和全局极值 $P_g = \max(w_k^i)$ 。

2) 根据个体极值和全局极值来更新每个粒子的速度 v_{k-1}^i 和位置 x_k^i , 使粒子群体向高似然区域移动, 更新方式如下:

$$v_{k-1}^i = |randn| (P_p^i - x_{k-1}^i) + |randn| (P_g^i - x_{k-1}^i) \quad (12)$$

$$x_k^i = x_k^i + v_{k-1}^i \quad (13)$$

其中, $|randn|$ 为正的高斯分布随机数, P_p^i 为每个粒子的个体极值, P_g^i 为粒子集合的全局极值。

3) 重新计算每个粒子的权值 w_k^i 并归一化为 w_k^i 。

步骤5: 对粒子群进行人工免疫运算, 通过人工免疫的变异操作来增强粒子群体的全局寻优能力;

1) 计算每个粒子的亲和力 $fit_k(i), i = 1, 2, \dots, N$, 权值大的粒亲和力小, 而权值小的粒子亲和力大。

$$fit_k(i) = 1 - w_k^i \quad (14)$$

2) 对每个粒子进行变异。变异操作会使那些权值小的粒子位置变化大, 权值大的粒子位置变化小, 用权值小的粒子通过变异来寻找可能出现的较

好的状态值。变异规则如下:

$$x_k^i = x_k^i + fit_k(i) \cdot randn() \quad (15)$$

3) 计算变异后粒子的权值并进行归一化处理。

4) 计算粒子群体适应度标准差 a , 如果 $a > 1/N$, 转至步骤6, 否则转至步骤4。

步骤6: 对粒子集合进行重采样操作。从粒子集合 $\{x_k^i, i = 1, 2, \dots, N\}$ 中根据权值 w_k^i 重新采样得到 N 个粒子集合, 使每个粒子的权值重新定义为 $\frac{1}{N}$;

步骤7: 令 $k = k + 1$, 估计出 k 时刻的状态值 $\hat{x} = \sum_{i=1}^N x_k^i w_k^i$;

步骤8: 判断是否结束, 若是则退出算法, 否则转至步骤2。

4 算法仿真分析

为验证 PSOAI-PF 算法的性能, 试验中重要性分布函数取先验概率分布 $p(x_k | x_{k-1})$, 取非线性非高斯动态模型为:

$$x_t = 1 + \sin(0.4\pi(t-1)) + 0.5x_{t-1} + w_{t-1} \quad (16)$$

$$y_t = \begin{cases} x_t^2 + v_t, & t \leq 30; \\ -2 + 0.5x_t + v_t, & t > 30 \end{cases} \quad (17)$$

其中, $t = 1, 2, \dots, T$ 。

过程噪声 $w_{t-1} \sim N(3/2, 3/4)$, 观察噪声 $v_t \sim N(0, 1 \times 10^{-5})$, 初始状态值 $x_0 = 1$, 时间 $T = 60$ s, 试验平台为 Matlab7.0, 分别对标准粒子滤波 (PF)、粒子群优化粒子滤波 (PSO-PF)、粒子群优化人工免疫滤波 (PSOAI-PF)、扩展卡尔曼粒子滤波 (EKPF) 和无迹卡尔曼粒子滤波 (UPF) 这 5 种滤波方法进行仿真, 采样粒子数 N 为 200。

几种算法的某次独立实验仿真结果如图 1 和 2 所示。

从图 1 可以看出, PSOAI-PF 算法状态估计结果明显好于 PSO-PF 和 PF, PF 与 PSO-PF 明显存在一些发散点, 这是因为 PSOAI-PF 在促使粒子集合向高似然区域移动的过程中由于粒子变异的存在避免了算法陷入局部最优。

图 2 中 PSOAI-PF 的估计效果与 UKF 的估计效果最好, 2 种算法的估计值几乎都分布在真实状态值附近, EKPF 的估计效果较差, 这是由于它对非线性非高斯模型进行了 1 阶近似而引入过多的线性误差所致, UPF 估计效果好是因为取样函数中加入了最新的观察数据。

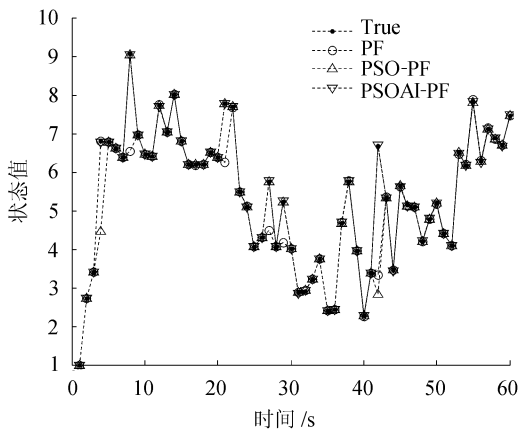


图 1 PSOAI-PF 与 PSO-PF、PF 的估计结果

Fig. 1 Estimate results of PSOAI-PF, PSO-PF and PF

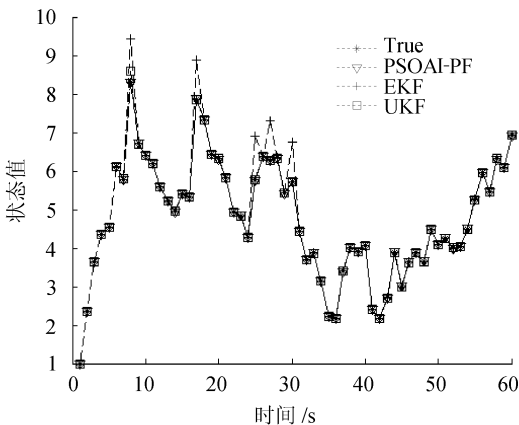


图 2 PSOAI-PF, UKF 和 EKPF 的估计结果

Fig. 2 Estimate results of PSOAI-PF, UKF and EKPF

为比较几种算法对状态的估计精度和运输量, 实验中通过对这五种算法进行了 $M = 60$ 次独立的蒙特卡洛模拟实验, 每次实验中均方根误差定义为:

$$RMSE = \left[\frac{1}{T} \sum_{i=1}^T (x_i - \bar{x}_i)^2 \right]^{1/2}$$

其中, x_i 为真实状态值, \bar{x}_i 为状态估计值。图 3 和 4 给出了几种算法在每次试验中的均方根误差 ($RMSE$) 曲线图。

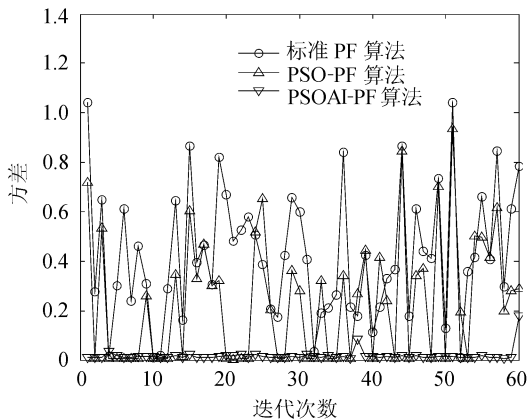


图 3 PSOAI-PF 与 PF、PSO-PF 的 $RMSE$ 曲线

Fig. 3 $RMSE$ curve of PSOAI-PF, PF and PSO-PF

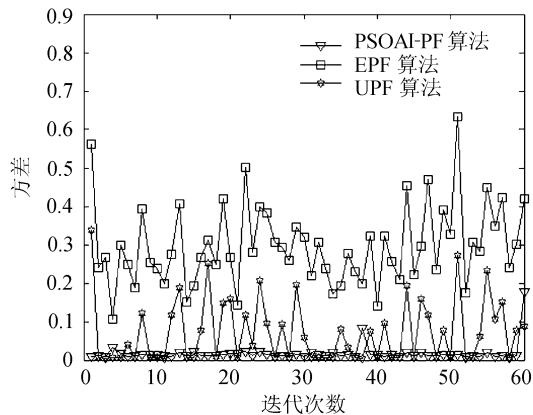


图 4 PSOAI-PF 与 EKPF、UPF 的 $RMSE$ 曲线

Fig. 4 $RMSE$ curve of PSOAI-PF, EKPF and UPF

从图 3 和 4 中可以看出, PSOAI-PF 在每次的试验中 $RMSE$ 都很小, 这表明状态估计值和真实状态十分接近, 状态估计精度较高, EKPF 的 $RMSE$ 在每次试验中都较大, 说明估计值和真实状态相差较远, 估计精度较差。

表 1 给出的是几种算法在 60 次独立蒙特卡洛实验中均方误差根的均值、方差、算法效率以及算法运行的平均时间, 表 1 中的算法效率为 $\eta =$

$$\frac{1}{E_{RMSE} N} \quad [15]$$

表 1 50 次蒙特卡洛实验 $RMSE$ 的均值、方差和平均耗时

Tab. 1 Mean value, variance and mean time consuming of $RMSE$ during 50 times monte carlo experiment

算法	$RMSE$ 均值	$RMSE$ 方差	$\eta / \%$	平均耗时/s
PF	0.420 710	0.061 185 0	1.188	1.397 4
PSO-PF	0.244 900	0.231 860 0	2.042	4.016 6
PSOAI-PF	0.013 325	$1.919 90 \times 10^{-5}$	37.523	1.492 2
EKPF	0.281 400	0.015 022 0	1.777	4.157 2
UPF	0.069 366	0.007 014 1	7.208	8.070 4

从表 1 可以看出: 1) PF 在 60 次的独立实验中平均估算耗时最少, 为 1.397 4 s, 但 $RMSE$ 均值最大, 为 0.420 710; UPF 平均耗时最多, 为 8.070 4 s, $RMSE$ 均值很小, 为 0.069 366, 这是由于 UPF 的取样函数加入了最新的观测值, 相比于 PF, 虽计算量是 PF 的 8 倍, 但精度提高了近 70 倍, UPF 是 1 种用时间代价来换取较高状态估计精度的算法。2) PSO-PF 的耗时大致和 EKPF 的时间相等, 但却是 PF 的 4 倍多, PSO-PF 耗时较多是因为在每次的优化过程中, 都是经过了多次的迭代过程(实验过程中设置为 5 次), 每次迭代过程都必须重新定位粒子的位置和速度; EKPF 耗时较多是因为每次运算都必须进行 1 阶 Taylor 展开和计算雅克比矩阵; 和标准 PF 相比, PSO-PF 的 $RMSE$ 均值提高了 1 倍, 算法效

率($\eta = 2.042$)也提高了近1倍,说明这种粒子群优化思想在粒子滤波中是有效的,但由于存在可能陷入局部最优的情况, *RMSE* 均值和算法效率不是很高。3) PSOAI-PF 在 60 次独立实验中平均耗比 PF 多了 0.015 s, 比 PSO-PF 降低了 2.54 s, 比 EKPF 降低 2.67 s, 比 UPF 降低了 6.58 s; PSOAI-PF 的 *RMSE* 均值是最小的, 为 0.013 325, 比 UPF 提高 6 倍, 比 PSO-PF 提高了近 21 倍, 比 EKPF 提高了近 28 倍, 比 PF 提高近 40 倍, 算法效率是最高的, $\eta = 37.523$ 。4) 从上述的分析可知, PSOAI-PF 无论是从算法的精度、运行的时间还是 *RMSE* 的方差上比较, 都优于其它的 4 种算法, 说明在 PF 中引入 PSOAI 能有效缓解采样粒子权值的退化和样本的枯竭, 能显著提高对状态的估计精度, 缩短算法运行的时间。

5 结 论

提出的 PSOAI-PF 算法中的重要性分布函数虽不是真实后验概率分布, 但通过 PSO 算法的全局寻优和 AI 中的变异能力, 能使取样粒子向真实后验概率密度函数高概率附近靠拢, 从本质上减小取样粒子和从真实后验概率分布中取样得到粒子的差异程度, 从而达到缓解粒子权值退化和样本枯竭的目的。仿真结果表明, PSOAI-PF 算法具有实时性高、稳定性好、状态估计方差小和算法效率高等优点, 因此把这种算法用于对非线性、非高斯系统的状态估计是可行且有效的。

参考文献:

- [1] Bar-Shalom Y, Li X R, Kirubarajan T. Estimation with application to tracking and navigation; Theory, algorithm and software[M]. New York: Wiley, 2001.
- [2] Gordon N J, Salmond D J, Smith A F M. Novel approach to nonlinear/mon-Gaussian Bayesian stste estimation [J]. IEEE-Proceedings—Radar, Sonar and Navigation, 1993, 140(2): 107–113.
- [3] Shi Yong, Han Chongzhao. Particle filter using second-order central difference[J]. Journal of Xi' an Jiaotong University: Natural Science, 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: 自然科学版, 2008, 42(4): 409–413.]
- [4] Du Zhengcong, Tang Bin, Li ke. The hybrid annealed particle filter[J]. Acta Physica Sinica, 2006, 55(3): 999–1005. [杜正聪, 唐斌, 李可. 混合退火粒子滤波器[J]. 物理学报, 2006, 55(3): 999–1005.]
- [5] Ning Xiaolei, Wang Hongli, Zhang Qi, et al. Interval diffracted particle filter[J]. Acta Physica Sinica, 2010, 59(7): 4426–4433. [宁小磊, 王宏力, 张琪, 等. 区间衍生粒子滤波器[J]. 物理学报, 2010, 59(7): 4426–4433.]
- [6] Zhang Qi, Hu Changhua, Qiao Yukun. Particle filter algorithm based on weight selected[J]. Control and Decision, 2008, 23(1): 117–120. [张琪, 胡昌华, 乔玉坤. 基于权值选择的粒子滤波算法研究[J]. 控制与决策, 2008, 23(1): 117–120.]
- [7] Zhou Guohui, Jing Zhongliang, Hu Hongtao. A particle filter algorithm based on optimizing combination resampling[J]. Journal of Shanghai Jiaotong University, 2006, 40(7): 1135–1139. [邹国辉, 敬忠良, 胡洪涛. 基于优化组合重采样的粒子滤波算法[J]. 上海交通大学学报, 2006, 40(7): 1135–1139.]
- [8] Cheng Shuiying, Zhang Jianyun. Fission Bootstrap particle filtering[J]. Acta Electronic, 36(3): 500–504. [程水英, 张剑云. 裂变自举粒子滤波[J]. 电子学报, 2008, 36(3): 500–504.]
- [9] Fang Zheng, Tong Guofeng, Xu Xin He. Particle swarm optimized particle filter [J]. Control and Decision, 2007, 22(3): 0273–0278. [方正, 佟国峰. 粒子群优化粒子滤波器[J]. 控制与决策, 2007, 22(3): 0273–0278.]
- [10] Zhang Qi, Wang Xin, Hu Changhua. Research on artificial immune particle filter [J]. Control and Decision, 2008, 23(3): 293–296. [张琪, 王鑫, 胡昌华. 人工免疫粒子滤波算法的研究[J]. 控制与决策, 2008, 23(3): 293–296.]
- [11] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proc of the IEEE Int Conf on Neural Networks. Piscataway: IEEE Service Center, 1995: 1941–1948.
- [12] Krohling R A. Gaussian swarm: A novel particle swarm optimization algorithm [C]// Proc of the IEEE Conf on Cybernetics and Intelligent Systems. Singapore, 2004: 372–376.
- [13] Duan Fu, Su Tongfen. Modified immune particle swarm optimization algorithm and its application [J]. Journal of Computer Applications, 2010, 30(7): 1883–1884. [段富, 苏同芬. 免疫粒子群算法的改进及应用[J]. 计算机应用, 2010, 30(7): 1883–1884.]
- [14] Liu Xiao, Liu Jinfeng, Zhao Wenbin, et al. The particle swarm optimization algorithm restraining local optimum [J]. Journal of Daqing Petroleum Institute, 2007, 31(6): 101–104. [刘霞, 刘金凤, 赵文斌, 等. 抑制局部最优的粒子群算法[J]. 大庆石油学院学报, 2007, 31(6): 101–104.]
- [15] Zhu Zhiyu, Yang Guanxiao. Stiefel manifold particle filtering [J]. Acta Physica Sinica, 2010, 59(12): 8316–8321. [朱志宇, 杨官校. 基于 Stiefel 流形的粒子滤波器研究[J]. 物理学报, 2010, 59(12): 8316–8321.]