控制与决策
Control and Decision

基于弹性机制的萤火虫优化粒子滤波算法

田梦楚, 柳林燕, 陈志敏, 方昱斌

引用本文:
田梦楚,柳林燕,陈志敏,方昱斌. 基于弹性机制的萤火虫优化粒子滤波算法[J]. 控制与决策, 2024, 39(2): 420–428.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2022.0195

基于弹性机制的萤火虫优化粒子滤波算法

田梦楚[1†], 柳林燕[1], 陈志敏[2], 方昱斌[1]

(1. 南京理工大学 智能制造学院，南京 210094；2. 中国卫星海上测控部，江苏 江阴 214431)

摘 要: 针对标准粒子滤波重采样导致的粒子贫化问题,提出一种基于弹性机制的萤火虫优化粒子滤波算法. 首先,利用萤火虫算法的吸引和移动机制,设计最优粒子引导粒子群体朝高似然区域移动的粒子运动控制策略;然后,评估粒子实时分布情况,根据每次迭代的高似然区域粒子占比值自适应控制粒子的优化强度;最后,检测最优粒子周围的粒子密度,引入弹簧的弹性机制,根据粒子密集度对判断区域内的粒子进行位置调整,使得粒子分布更加合理,提高粒子滤波的精度. 实验结果表明,在粒子数目较少的情况下,改进算法滤波精度较标准粒子滤波提高12 %～25 %;在同等滤波精度需求下,改进算法的运算时间比标准粒子滤波的运算时间减少20 %～30 %,改进算法的综合性能更优.

关键词: 粒子滤波；萤火虫算法；粒子贫化；弹性机制；智能优化

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2022.0195

引用格式: 田梦楚, 柳林燕, 陈志敏, 等. 基于弹性机制的萤火虫优化粒子滤波算法 [J]. 控制与决策, 2024, 39(2): 420-428.

Firefly optimized particle filter algorithm based on spring mechanism

TIAN Meng-chu[1†], *LIU Lin-yan*[1], *CHEN Zhi-min*[2], *FANG Yu-bin*[1]

(1. School of Intelligent Manufacturing，Nanjing University of Science and Technology，Nanjing 210094，China；
2. China Satellite Maritime Tracking and Controlling Department，Jiangyin 214431，China)

Abstract: Aiming at the problem of particle impoverishment caused by the resampling process in standard particle filter, we propose a firefly optimized particle filter algorithm based on the spring mechanism. Firstly, adopting the attraction and movement mechanism of the firefly algorithm, we design the motion controlled strategy where the optimal particle is used to guide the particles moving towards the high likelihood region. Secondly, the real-time distribution of particles is evaluated, and the optimization intensity of particles is adaptive controlled by the proportion of particles in the high likelihood region. Finally, the density of the particles around the optimal particle is detected, and the elastic mechanism of spring is introduced, and the positions of particles are adjusted according to the particle density around the optimal particle, which can make the distribution of particles more reasonable and enhance the precision of the particle filter. The experimental results show that the filtering accuracy of the improved algorithm is 12 % to 25 % higher than that of the standard particle filter when the number of particles is small, and the operation time of the improved algorithm is about 20 % to 30 % less than that of the standard particle filter under the same filtering accuracy requirements, and the improved algorithm has the better comprehensive performance.

Keywords: particle filter；firefly algorithm；particle impoverishment；spring mechanism；intelligent optimization

0 引 言

贝叶斯(Bayesian)滤波是一种协调先验信息与当前信息的统一方法,其利用系统状态的先验概率密度和当前量测,计算后验概率密度[1-2]. 针对不同的系统,贝叶斯滤波有其不同的实现方法. 线性条件下,卡尔曼滤波(KF)能够得到最优估计值. 非线性高斯条件下,扩展卡尔曼滤波(EKF)利用泰勒级数展开将非线性问题线性化,无迹卡尔曼滤波(UKF)通过采样点线性回归近似非线性函数,2种算法均属于次优滤波算法[3-4]. 但是,在非线性非高斯条件下,UKF和EKF

收稿日期: 2022-01-26；**录用日期:** 2022-09-20.
基金项目: 国家自然科学基金项目(62103190).
†通讯作者. E-mail: tianmengchu@163.com.

的性能下降明显,难以满足较高的精度要求. 粒子滤波(particle filter, PF)利用蒙特卡罗方法实现递推贝叶斯滤波[5-7],其不受非线性非高斯限制,能够解决实际系统中普遍存在的非线性非高斯滤波问题[8-9]. 目前,粒子滤波已广泛应用于诸多实际工程领域,如目标跟踪、导航制导、状态监视、故障诊断、参数估计、系统辨识、计算机视觉、金融领域等[10-15].

粒子滤波以样本形式而非函数形式递推状态向量的后验概率密度函数,其存在固有的粒子权值退化问题[16],重采样技术可解决该问题,但是会破坏粒子多样性,造成粒子贫化问题,导致滤波精度下降[17-19]. 针对该问题,许多学者提出了多种改进重采样算法,文献[20-22]均是对标准重采样方法的粒子采样机制进行改进,改善了粒子多样性. 但是,这些方法仍然涉及对小权值粒子的舍弃,而小权值粒子中也含有一定的信息,盲目舍弃会影响信息的完整性.

随着智能优化算法的深入研究,许多学者将智能优化算法与粒子滤波相结合,利用智能优化算法的寻优机制解决粒子滤波的粒子权值退化问题[23]. 文献[24]将遗传算法引入粒子滤波中,设定阈值对粒子进行分组,利用遗传算法的交叉和变异操作优化分组粒子,提高了粒子多样性,但是,该方法以随机方式选取交叉操作中的转化对象,优化效率较低,运算负担较大. 文献[25]提出了一种基于自控蝙蝠算法的粒子滤波,其用粒子表征蝙蝠个体,通过模拟蝙蝠群体的搜索行为,驱使粒子向高似然区域移动,使得粒子分布更加合理,但是,该方法的粒子信号响度更新方式会影响粒子精细搜索的能力,限制粒子滤波精度的提高. 文献[26]提出了一种蜂鸟优化粒子滤波,很好地平衡了蜂鸟算法的局部搜索能力与全局搜索能力,通过自搜索和引导搜索指导粒子移动,解决了粒子贫化问题,但是,该方法优化过程较为复杂,运算负担较大. 文献[27]提出了一种灰狼粒子滤波,通过模拟灰狼群狩猎过程中的4种行为模式更新粒子位置,改善粒子分布,提高粒子多样性,但是,该方法以最大迭代次数为优化终止条件,优化强度的控制依赖于经验值. 文献[28]提出了萤火虫优化粒子滤波方法,其利用最优粒子引导粒子群体的运动,寻优效率较高,但是,优化强度的控制依赖于经验值,可能会由于过度优化而造成粒子不合理地聚集于最优粒子周围,影响下一时刻的粒子搜索范围,导致滤波精度不稳定.

现有大多数智能优化粒子滤波方法主要是利用智能优化算法对粒子状态进行优化,仅依靠经验设置迭代终止阈值或最大迭代次数,可能由于过度优化而造成滤波精度下降问题. 针对该问题,本文提出一种基于弹性机制的萤火虫优化粒子滤波方法(SFA-PF). 首先,所提出方法以最优粒子引导其他粒子运动,结合粒子实时分布情况建立优化强度判据,精准控制优化强度;然后,进一步利用高似然粒子占比值衡量最优粒子周围的粒子密度,当粒子密度过大时,引入弹簧的弹性机制,通过弹力作用,驱散过度聚集于判断区域内的粒子,更精准地调整粒子滤波的粒子分布;最后,在优化完成后,对粒子权值进行补偿,使得优化前和优化后的带权值粒子在理论上服从同一分布.

1　粒子滤波

粒子滤波是一种通过蒙特卡罗思想实现递推贝叶斯滤波的方法,其核心思想是利用N个独立同分布的粒子$\{x^i(k)\}_{i=1}^N$及其对应的权值$\{\omega^i(k)\}_{i=1}^N$近似地描述后验概率密度函数. 假设k时刻的状态和量测分别为$x(k)$和$z(k)$,记k时刻及其之前时刻的量测为$Z(k)=\{z(1),z(2),\dots,z(k)\}$,则利用带权值样本表征状态向量的后验概率密度函数形式如下:

$$p(x(k)|Z(k))=\sum_{i=1}^{N}\omega^i(k)\delta(x(k)-x^i(k)),\quad(1)$$

$$\tilde{\omega}^i(k)\propto\tilde{\omega}^i(k-1)p(z(k)|x^i(k)).\quad(2)$$

其中:$\delta(\cdot)$为狄克拉函数,$p(z(k)|x^i(k))$为似然函数.

2　人工萤火虫算法

人工萤火虫算法(firefly algorithm, FA)是通过模拟萤火虫群体行为构造出的一类随机优化算法[29]. 从数学角度对标准人工萤火虫算法进行如下描述.

1) 萤火虫的相对荧光亮度为

$$I=I_0\times\mathrm{e}^{-\gamma r_{ij}^2}.\quad(3)$$

其中:I_0为萤火虫的最大萤光亮度,γ为光强吸收系数,r_{ij}为萤火虫i与萤火虫j间的空间距离.

2) 萤火虫的吸引度为

$$\beta=\beta_0\times\mathrm{e}^{-\gamma r_{ij}^2},\quad(4)$$

其中β_0为最大吸引度.

3) 萤火虫i被萤火虫j吸引移动的位置更新公式如下式所示:

$$x_i=x_i+\beta\times(x_j-x_i)+\alpha\times(\mathrm{rand}-1/2).\quad(5)$$

其中:x_i、x_j为萤火虫i和j所处的空间位置,$\alpha\in[0,1]$为步长因子,rand为$[0,1]$上服从均匀分布的随机数.

3 基于弹性机制的萤火虫优化粒子滤波

3.1 算法原理

本文设计一种基于弹性机制的萤火虫优化粒子滤波,该方法利用萤火虫算法的吸引和移动机制优化粒子分布,进一步设计粒子相互作用模式,建立优化强度判别标准,防止粒子不合理聚集以及过度优化带来的运算负担和低质量收敛问题.

首先,建立粒子移动机制,将粒子视为萤火虫个体,通过萤火虫算法自身的寻优机制,使得小权值粒子向大权值粒子移动,更加合理地利用了小权值粒子包含的信息,为提高算法的效率,每次迭代均以最优粒子指导其他粒子的运动,提高了粒子的优化效率;然后,控制算法优化强度,以最优粒子为圆心,取判断半径内的区域为高似然区域,通过计算高似然区域内的粒子占比值描述粒子分布情况,设置最大高似然粒子占比值,当实时计算的高似然粒子占比值大于或等于最大高似然粒子占比值时,停止优化;最后,增加粒子密度检测环节,迭代终止后,检测最优粒子周围的粒子密度,当密度过大时引入弹簧的弹性机制,在最优粒子与判断区域内粒子间构建弹簧系统,驱使判断区域内的聚集粒子散开,使得粒子分布更加合理.

3.2 具体实现方法

3.2.1 吸引和移动机制

1) 粒子荧光亮度.

引入最新的量测值衡量粒子位置优劣,定义粒子荧光亮度公式为

$$I_m^i(k) = I_0 \times \mathrm{e}^{-\gamma|z(k)-z_m^i(k)|}. \tag{6}$$

其中: $I_m^i(k)$为k时刻第m次迭代粒子i的荧光亮度, $z(k)$为k时刻滤波器最新的量测值, $z_m^i(k)$为k时刻第m次迭代粒子i的预测值.

荧光亮度值的意义在于判断粒子位置的优劣,粒子的荧光亮度值越大,该粒子的位置越优,每次迭代时,将荧光亮度值最大的粒子定义为该次迭代的最优粒子gbest,有

$$\begin{aligned}&\mathrm{gbest}_m(k) \in \{x_m^i(k),\ i=1,2,\dots,N|I(x)\} = \\ &\max\{I(x_m^i(k)),\ i=1,2,\dots,N\}.\end{aligned} \tag{7}$$

其中: $\mathrm{gbest}_m(k)$为k时刻第m次迭代的最优粒子状态值, $x_m^i(k)$为k时刻第m次迭代粒子i的状态值, $I(\cdot)$为求荧光亮度操作.

2) 粒子吸引度.

SFA-PF利用最优粒子引导其他粒子运动,最优粒子gbest与粒子i间吸引度的大小直接影响粒子i朝最优粒子gbest移动的距离,最优粒子gbest对粒子i的吸引度公式为

$$\beta_m^i(k) = 1 - \beta_0 \times \mathrm{e}^{-\gamma r^2_{i\mathrm{gbest}_m(k)}}. \tag{8}$$

其中: $\beta_m^i(k)$为k时刻第m次迭代最优粒子gbest对粒子i的吸引度, $r_{i\mathrm{gbest}_m(k)}$为粒子i与最优粒子gbest的空间距离.

3) 粒子位置更新.

SFA-PF以最优粒子引导其他粒子运动,粒子的位置更新公式如下式所示:

$$\begin{aligned}x_m^i(k) = x_{m-1}^i(k) + \beta \times (\mathrm{gbest}_m(k) - \\ x_{m-1}^i(k)) + \alpha \times (\mathrm{rand} - 1/2).\end{aligned} \tag{9}$$

其中: $x_m^i(k)$为粒子i在k时刻第m次迭代后的状态值, $x_{m-1}^i(k)$为粒子i在k时刻第$m-1$次迭代后的状态值.

3.2.2 优化强度控制

本文利用高似然粒子占比值描述粒子分布情况,结合最大迭代次数控制优化强度,并设置粒子密度检测机制,控制最优粒子周围的粒子密度,以改善粒子分布,使得大部分粒子均集中于高似然区域,且其他区域也有粒子分布.

1) 粒子密集度.

首先,定义粒子密集度,制定量化指标,判断粒子是否过于密集. 第m次迭代时,完成位置更新后,大多数粒子均朝最优粒子方向移动. 以第m次迭代的最优粒子为圆心,取判断半径为r_{judge},将判断区域内的粒子数目定义为粒子密集度cr_m,有

$$\mathrm{cr}_m = \mathrm{find}\{\|x_m^i(k) - \mathrm{gbest}_m(k)\| \leqslant r_{\mathrm{judge}}\}. \tag{10}$$

其中: $x_m^i(k)$为k时刻第m次迭代时粒子i经过移动后的位置, $\mathrm{gbest}_m(k)$为k时刻第m次迭代时最优粒子的位置, find为获取判断区域内粒子数目的操作.

2) 高似然粒子占比值.

定义高似然粒子占比值,利用高似然粒子占比值与最大迭代次数相结合的方式控制粒子优化强度,高似然粒子占比值定义为

$$\mathrm{pr}_m = \frac{\mathrm{cr}_m}{N}. \tag{11}$$

当第m次迭代的高似然粒子占比值大于所设置的最大高似然粒子占比值$\mathrm{pr}_{\max}$或迭代次数大于最大迭代次数,停止优化.

3.2.3 粒子密度检测

衡量迭代终止后的粒子分布情况,若实时计算的高似然粒子占比值超过设置的阈值$\mathrm{pr}_{\mathrm{th}}$(该阈值应略

高于最大高似然粒子占比值),则表明最优粒子周围的粒子过于密集,需要调整最优粒子周围的粒子分布. 构建判断区域内粒子与最优粒子间的弹簧系统,弹簧的压缩量等于粒子朝最优粒子移动的距离. 当粒子过于密集时,弹簧将粒子朝被压缩的相反方向弹开,压缩量越大,弹簧的弹力越大,粒子被弹开的距离越远.

1) 回弹系数.

第m次迭代时,粒子朝最优粒子移动,移动的距离即弹簧的压缩量,则弹簧压缩量定义为

$$\Delta x_m^i = \|x_m^i(k) - x_{m-1}^i(k)\|. \tag{12}$$

由胡克定律,对应的弹力为

$$F_m^i = k_s \cdot \Delta x_m^i, \tag{13}$$

其中k_s为粒子间弹簧的弹性系数.

定义粒子被弹开的回弹系数为

$$\zeta = \mathrm{e}^{-c/F_m^i}, \tag{14}$$

其中c为常数. 弹开距离应与粒子移动距离成正比,且必须小于粒子的移动距离,以避免粒子无效寻优,因此设置常数c调整回弹系数,而弹性系数的作用是为了完整引入弹簧的弹性理论. 将式(13)代入(14),可得到回弹系数

$$\zeta = \mathrm{e}^{-(c/k_s)/\Delta x_m^i},$$

其中: c/k_s为一个常数; k_s的取值对回弹系数的影响可通过c来消除. 因此为了方便计算,直接取弹性系数为1.

2) 弹性机制作用下的粒子位置更新.

将判断区域内的粒子沿被压缩的相反方向弹开,弹开的距离受回弹系数影响,被弹开粒子的位置更新公式为

$$\widehat{x}_m^i(k) = x_m^i(k) + \zeta \times (x_{m-1}^i(k) - x_m^i(k)). \tag{15}$$

3.2.4　粒子权值补偿

经过萤火虫算法优化后的粒子状态发生了改变,粒子集所表征的分布密度函数一般不再是$p(x(k)|Z(k-1))$,需要采取补偿措施,使得优化前后的带权值粒子集从理论上服从同一分布$p(x(k)|Z(k-1))$. 本文利用重要性采样思想并结合权值的定义方式,对优化后的粒子权值进行补偿,此处的权值补偿过程不同于式(2)的粒子权值求解过程,而是对优化前后带权值粒子的同分布统一过程.

假设$k-1$时刻的粒子集$\{x^i(k-1), \omega^i(k-1)\}_{i=1}^N$表征后验概率密度函数$p(x(k-1)|Z(k-1))$, k时刻经过状态转移得到由新的预测粒子集$\{\tilde{x}^i(k)\}_{i=1}^N$表征一步预测密度函数$p(x(k)|Z(k-1))$,采用改进萤火虫算法对预测粒子集进行迭代寻优,得到优化后新的粒子集$\{x^i(k)\}_{i=1}^N$,假设优化后的粒子集表征概率密度函数$g(x(k))$. 换言之,经过优化后,样本集$\{x^i(k)\}_{i=1}^N$可看作是从概率密度函数$g(x(k))$中采样得到,而非从概率密度函数$p(x(k)|Z(k-1))$中采样得到,因此,粒子位置变化的同时也应对权值进行相应补偿,保证优化前后的带权值粒子分布统一性. 将$g(x(k))$视为$p(x(k)|Z(k-1))$的重要性密度函数,此时由权值的定义,设粒子i的权值补偿率为$R^i(k)$,有

$$R^i(k) = \frac{p(x^i(k)|Z(k-1))}{g(x^i(k))}, \tag{16}$$

结合式(2),得到优化后粒子的权值补偿更新公式为

$$\omega^i(k) \propto \omega^i(k-1) R^i(k) p(z(k)|x^i(k)). \tag{17}$$

3.3　算法具体实现步骤

step 1: $k=0$时刻,初始化样本集合$\{x_0^i, \omega_0^i\}_{i=1}^N$,设置重要参数. 采样$N$个粒子$\{x_0^i, i=1,2,\ldots,N\}$作为算法的初始粒子. 重要性密度函数如下式所示:

$$x^i(k) \sim p(x(k)|x(k-1)). \tag{18}$$

设置萤火虫算法的最大吸引度β_0,光强吸收系数γ,步长因子α,判断半径为r_{judge},最大高似然粒子占比值$\text{pr}_{\max}$,弹簧的弹性系数k_s,最大迭代次数$\text{it}_{\max}$.

step 2: 预测k时刻粒子状态$\{\tilde{x}^i(k)\}_{i=1}^N$. 由$k-1$时刻的粒子集$\{x^i(k-1)\}_{i=1}^N$和系统动态模型,预测$k$时刻的粒子集合为$\{\tilde{x}^i(k)\}_{i=1}^N$.

step 3: 更新粒子权值$\{\tilde{\omega}^i(k)\}_{i=1}^N$. 引入$k$时刻最新量测值$Z(k)$,由式(2)计算粒子权值.

step 4: 由下式计算有效粒子量N_{eff}:

$$N_{\text{eff}} = 1 \Big/ \sum_{i=1}^{N} (\omega^i(k))^2. \tag{19}$$

当有效粒子量小于设定阈值时,优化粒子集执行step 5,否则直接转至step 11.

step 5: 计算各粒子荧光亮度,并更新最优粒子. 由式(6)计算各粒子的荧光亮度值,利用式(7)更新全局最优粒子.

step 6: 计算粒子吸引度,更新粒子位置. 由式(8)计算最优粒子对任意粒子的吸引度,并由式(9)更新粒子位置.

step 7: 计算高似然粒子占比值. 由式(10)和(11)计算最优粒子判断半径内的粒子密集度和高似然粒子占比值.

step 8: 判断优化终止条件. 当高似然粒子占比值大于等于最大高似然粒子占比值,或迭代次数达到最大迭代次数时,停止优化; 否则,转至step 5.

step 9: 检测粒子密度. 当优化后的高似然粒子占

比值大于设定的阈值$\mathrm{pr_{th}}$时,由式(12)～(15),将高似然区域内粒子按原移动方向弹开,更新粒子位置.

step 10: 优化后粒子的权重补偿和更新. 由式(16)和(17)对优化后的粒子进行权值补偿和更新,并对权值进行归一化处理.

step 11: 由下式输出估计状态:

$$\hat{x}(k)=\sum_{i=1}^{N}\omega^i(k)x^i(k). \tag{20}$$

3.4 SFA-PF算法复杂度分析

与采用系统重采样算法的标准粒子滤波相比,SFA-PF以迭代寻优过程取代标准粒子滤波的重采样过程,分别对SFA-PF的寻优过程和PF重采样过程的运算复杂度进行分析. 假设粒子数目为N,PF运行时长内共有T_1个时刻执行系统重采样操作;SFA-PF运行时长内共有T_2个时刻执行优化操作,设置最大迭代次数为M.

1) SFA-PF迭代寻优过程的复杂度分析.

在粒子寻优阶段,每个粒子状态更新的运算复杂度为$3\times O(1)$;在优化强度判断阶段,需要计算高似然粒子占比值,判断每个粒子到最优粒子的距离,该部分的运算复杂度为$1\times O(1)$. 当最优粒子周围粒子过于密集时,引入弹性机制,粒子状态调整部分的运算复杂度为$2\times O(1)$,则每次迭代每个粒子完整的状态更新运算复杂度为$6\times O(1)$,所有粒子完成一次迭代的状态更新的运算复杂度为$6\times N\times O(1)$. 以最大迭代次数为M,运行时长内执行优化次数为T_2计算,SFA-PF迭代寻优过程的总体运算复杂度为$O(6\times N\times M\times T_2)$. 由于本文结合粒子分布情况与最大迭代次数控制优化强度,大多数时刻不会达到最大迭代次数.

此外,弹性机制只在优化过度情况下引入,其引入次数应小于优化执行次数T_2,且只有判断区域内的粒子受弹簧弹力影响被弹开,被调整的粒子数目小于N. 因此,SFA-PF实际迭代寻优过程的总体运算复杂度应小于$O(6\times N\times M\times T_2)$.

2) PF系统重采样过程的复杂度分析.

粒子滤波的系统重采样过程涉及粒子的相互交互,因此其运算复杂度为$O(N\times N)$,以运行时长内执行重采样次数为T_1计算,PF重采样过程的总体运算复杂度为$O(N\times N\times T_1)$.

综上所述,相同粒子数目条件下,SFA-PF以粒子自优化过程取代标准粒子滤波的重采样过程,增加的运算复杂度为$O(6\times N\times M\times T_2)$,减少的运算复杂度为$O(N\times N\times T_1)$. SFA-PF通过优化粒子状态改善粒子分布,这种分布的改善作用具有后效性,因此,SFA-PF执行优化操作的次数小于PF执行系统重采样操作的次数,即$T_1>T_2$,且每次优化的迭代次数较少,在仿真实验部分也证实了这一点. 虽然从运算复杂度分析,SFA-PF与PF相差不大,但是,改进萤火虫优化算法的运算公式比系统重采样算法的运算公式复杂,因此,总体运算时间SFA-PF略高于PF,这与仿真实验部分的数据相符.

4 仿真实验

仿真软件环境为Matlab 2010 b,选取文献[30]的典型非线性系统案例,测试所提出SFA-PF算法的性能.

文献[30]中的非线性单变量不稳定增长模型,其过程模型和量测模型如下:

$$x(t)=0.5x(t-1)+\frac{25x(t-1)}{1+[x(t-1)]^2}+8\cos[1.2(t-1)]+w(t), \tag{21}$$

$$z(t)=\frac{x(t)^2}{20}+v(t), \tag{22}$$

其中$w(t)$和$v(t)$为系统噪声和量测噪声.

4.1 重要参数设定

本文通过实验对影响改进算法性能的重要参数进行研究分析. 在改进算法中,光强吸收系数决定了粒子朝最优粒子移动的效率,直接影响优化过程的有效性;最优粒子判断半径为r_{judge},决定了优化强度和最终的粒子分布情况,直接影响改进萤火虫算法对粒子滤波粒子分布的改善效果.

1) 光强吸收系数设定.

标准萤火虫算法一般取光强吸收系数为1,是因为标准萤火虫算法吸引度与荧光亮度呈正比,且吸引度大小随萤火虫个体距离的增大而减少. 当2个萤火虫间的距离较远时,其吸引度趋近于0,能够遵循萤火虫朝距离较近的更亮萤火虫移动的规则. 但是,利用萤火虫算法优化粒子滤波时,以最优粒子指导其他粒子运动,若沿用标准萤火虫算法的吸引度公式,则最优粒子对较远的粒子吸引度过小,寻优效率较低. 为了测试改进算法中光强吸收系数对粒子吸引度的影响,分别取光强吸收系数γ为0.001、0.005、0.01、0.05、0.1、0.5、1. 图1为不同光强吸收系数取值时吸引度随粒子与最优粒子距离变化的曲线.

由图1可见,随着光强吸收系数的增大,同等距离下最优粒子对其他粒子的吸引度逐渐增大,且光强吸收系数越大,吸引度的增长越快. 当$\gamma=0.05$时,最优粒子对距离大于10的粒子,吸引度趋近1. 经过一次迭代后,几乎全部粒子均集中于以最优粒子为

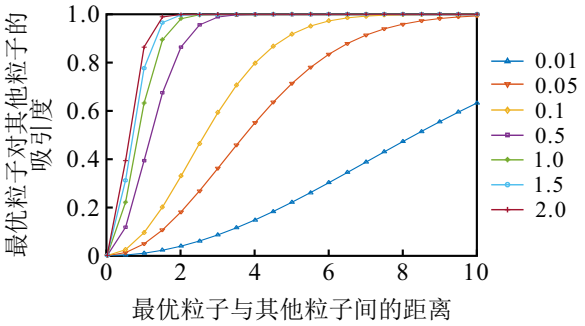

图 1　不同光强吸收系数的粒子吸引度变化曲线

圆心,10为半径的区域内,此时粒子收敛速度过快,不利于迭代寻优. γ取值越大,粒子的收敛速度越快,当$\gamma=1$时,仅经过一次迭代所有粒子均集中于最优粒子2.5半径区域内,导致粒子过早收敛. 当$\gamma=0.001$时,最优粒子对距离小于10的粒子,吸引度小于0.1,此时优化效率过低. 最优粒子与粒子间距离较近的情况下,吸引度过小可能导致扰动项的作用大于最优粒子的吸引作用,大多数粒子处于无序随机搜索状态,影响粒子的精细搜索能力. 当光强吸收系数取值取0.005或0.01时,随着最优粒子与其他粒子距离的增大,吸引度增长相对平缓,有利于粒子逐步寻优,且$\gamma=0.01$时的粒子寻优效率高于$\gamma=0.005$,因此,取光吸收系数为0.01.

2) 最优粒子判断半径设定.

利用群智能优化算法优化粒子滤波的粒子分布时,通常设置最大迭代次数控制优化强度,但是,群智能优化算法属于一种元启发式算法,其搜索过程存在一定的随机性,仅依靠最大迭代次数难以实现高效寻优. 本文通过设置高似然粒子占比值量化粒子分布情况,综合粒子分布情况和最大迭代次数控制粒子优化强度. 在优化过程中,粒子整体朝最优粒子方向运动,通过最优粒子周围的粒子密度值衡量优化进程,高似然粒子占比值取决于最优粒子判定半径,分别取判断半径r_{judge}为0.1、0.5、1、1.5、2、3时,部分时刻迭代10次的高似然粒子占比值如表1所示.

表1　不同时刻不同判断半径条件下的高似然粒子占比值　　%

判断半径		迭代次数									
		1	2	3	4	5	6	7	8	9	10
$r_{\text{judge}}=0.1$	$k=8$	5	2	3	10	7	7	8	17	18	14
	$k=42$	1	1	0	3	9	11	17	8	8	10
	$k=98$	4	5	10	20	14	20	10	17	17	21
$r_{\text{judge}}=0.5$	$k=8$	5	23	25	31	38	30	30	41	40	49
	$k=42$	10	16	17	17	19	19	27	25	30	31
	$k=98$	6	34	46	54	46	50	55	57	58	73
$r_{\text{judge}}=1$	$k=8$	8	35	44	46	48	51	57	62	66	67
	$k=42$	12	23	25	31	38	47	53	54	59	69
	$k=98$	10	46	55	64	65	68	78	80	83	91
$r_{\text{judge}}=1.5$	$k=8$	43	48	58	78	87	95	98	97	98	99
	$k=42$	29	49	55	65	66	73	74	79	88	90
	$k=98$	65	88	98	100	100	100	100	100	100	100

如表1所示,当取判断半径为0.1和0.5时,判断区域内的粒子数量随迭代次数的增加呈上升趋势,但是,增长过程存在波动,这是因为当粒子距最优粒子较近时,扰动项的作用可能大于最优粒子对该粒子的吸引作用,在扰动项的作用下粒子在最优值附近搜寻新的更优值,此时粒子有可能朝最优粒子的相反方向运动,因此,判断区域内的粒子数量可能存在小幅度的减少. 当判断半径大于1时,判断区域内的粒子数量持续增加,直至所有粒子均移动至判断区域内,且判断半径越大,高似然粒子占比值经过前几次迭代便达到100%,不利于通过粒子分布反映粒子优化情况; 当判断半径为1时,随着迭代次数的增加,高似然粒子占比值增长相对平稳,能够很好地反映粒子经过优化后朝最优粒子运动的情况. 因此,本文选择判断半径为1.

4.2　算法性能测试

为了表明SFA-PF的优势,将SFA-PF同时与采用系统重采样方法的标准PF、文献[23]提出的粒子群优化粒子滤波(PSO-PF)、文献[26]提出的蜂鸟优化粒子滤波(HOA-PF)以及文献[28]提出的萤火虫优化

粒子滤波(FA-PF)进行对比测试. PSO-PF、HOA-PF、FA-PF 和 SFA-PF 均是在标准 PF 基础上融入智能优化算法优化粒子集,为了保证5种算法比较的公平性,各算法分配的粒子数目相同,模型的噪声条件一致,几种群智能优化粒子滤波的最大迭代次数相同,其他参数参考对应文献设置. 在 PSO-PF 部分: 设置学习因子 $c_1 = c_2 = 2$,惯性权重线性递减,其最大值为0.9,最小值为0.3,最大迭代次数为10. 在 HOA-PF 部分: 粒子自搜索阶段,粒子朝最优值搜索的步长因子设置为0.01,利用 Mantegna 方法生成服从 Levy 分布的随机步长;粒子引导搜索阶段,设置领地粒子的巡逻因子为0.1,跟随粒子搜寻到领地粒子的概率为0.5,柯西分布的位置参数为0,尺度参数为1,最大迭代次数为10. 在 FA-PF 部分: 设置最大吸引度为1,步长因子为0.5,光强吸收系数为0.01. SFA-PF 在 FA-PF 的基础上,设置判断半径为1,最大高似然粒子占比值为50%,最优粒子周围粒子密度阈值为60%,最大迭代次数为10. 设系统噪声方差 $Q=1$,量测噪声方差 $R=1$,滤波时间步数为60.

均方根误差(RMSE)公式为

$$\mathrm{RMSE} = \left[\frac{1}{T}\sum_{t=1}^{T}(x_k - \hat{x}_k)^2\right]^{1/2}. \tag{23}$$

其中: T 为滤波时间步数, x_k 为 k 时刻的真实状态值.

当粒子数目 $N=100$ 时,仿真结果如图2和图3所示. 由图2和图3可见,与采用系统重采样方法的标准 PF 相比,群智能优化粒子滤波方法具有更好的状态估计精度,这是由于群智能优化粒子滤波以粒子的迭代寻优过程取代重采样过程,提高粒子整体质量的同时,保持了粒子多样性. 相比 PSO-PF、HOA-PF 与 FA-PF, SFA-PF 状态精度更高,这是因为前3种智能优化算法通过设置最大迭代次数控制优化强度,优化效果依赖于经验值,而 SFA-PF 通过实时评估粒子分布情况,自适应控制优化强度,避免了过度优化可能造成的精度和速度损失. 此外, SFA-PF 还增加了粒子密度检测环节,利用弹性机制调整粒子位置,缓解粒子聚集现象,使得粒子分布更加合理. 因此,相比于前4种算法, SFA-PF 具有更高的状态估计精度.

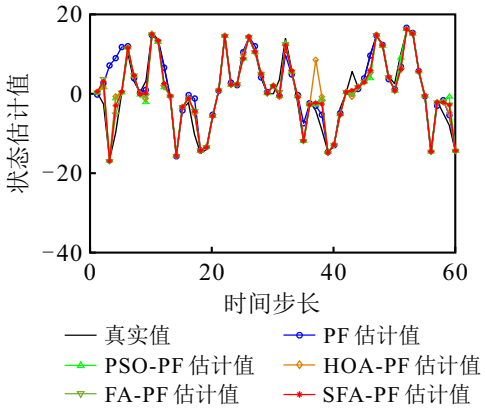

图 2 不同算法的状态估计值对比

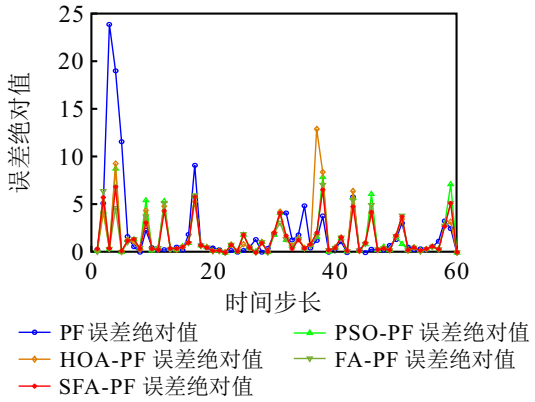

图 3 不同算法的状态估计误差绝对值对比

为了进一步研究改进萤火虫算法与粒子滤波相结合的运算负担,也为了使得状态估计精度的仿真结果更客观,分别取粒子数目为20、50、100和200,仿真50次,取5种算法50次仿真的平均RMSE与平均运算时间进行对比,结果如表2所示. 在滤波过程中,当有效粒子量小于设定阈值时,采取措施解决粒子权值退化问题, PF 采用的是系统重采样方法, SFA-PF 是利用智能优化算法替代传统重采样算法,其实现机理与传统重采样截然不同,但是2种操作的目的均是为了解决粒子权值退化问题. 分别列出 PF 和 SFA-PF 仿真过程中重采样算法和改进萤火虫优化算法的平均执行次数(50次仿真的总重采样次数/50或50次仿真的总优化次数/50)以及 SFA-PF 每次优化的平均迭代次数(50次仿真的总迭代次数/总优化次数),计算结果取最近的整数,对比结果如表3所示.

表 2 不同粒子数目条件下不同算法的滤波结果对比

粒子数目	平均RMSE					运算时间/s				
	PF	PSO-PF	HOA-PF	FA-PF	SFA-PF	PF	PSO-PF	HOA-PF	FA-PF	SFA-PF
$N=20$	6.129 5	4.687 2	4.602 1	4.612 5	4.590 1	0.092 4	0.130 8	0.175 9	0.110 2	0.107 7
$N=50$	4.916 9	4.351 9	4.415 7	4.301 7	4.271 1	0.103 4	0.151 6	0.210 2	0.152 7	0.131 3
$N=100$	4.603 8	4.245 0	4.003 0	4.240 1	4.162 8	0.134 5	0.189 5	0.252 8	0.189 3	0.175 9
$N=200$	4.341 4	4.178 1	3.928 4	4.027 1	4.005 7	0.192 8	0.309 7	0.321 5	0.290 1	0.269 8

表3　系统重采样和SFA执行次数对比

粒子数目	平均执行次数		平均迭代次数	
	PF	SFA-PF	PF	SFA-PF
$N=20$	39	12	–	4
$N=50$	37	10	–	3
$N=100$	36	7	–	4
$N=200$	36	8	–	4

由表2可见,随着粒子数目的增加,5种算法的估计精度均有不同程度的提高,且PF的精度提高幅度更大. 当粒子数目为20时,PF与其他4种算法相比,误差较大;当粒子数目为200时,PF的误差与其他4种算法的误差差距逐渐缩小,表明增大粒子数目可提高滤波精度,相比其他4种智能优化粒子滤波,PF的精度更依赖粒子数目. 与PSO-PF、HOA-PF和FA-PF相比,SFA-PF有较好的状态估计精度和较少的运算时间,其自适应控制优化强度、检测粒子密度,精准地改善粒子分布的同时避免过度优化带来的运算负担. 由表3可见,粒子数目从20增加至200时,PF执行系统重采样的次数以及SFA-PF执行优化操作的次数变化均不大. 同等粒子数目条件下,SFA-PF执行优化操作的次数小于PF执行重采样操作的次数,且优化的平均迭代次数较少. 这是由于粒子滤波是以粒子作为传播途径,表征状态的概率密度分布,上一个时刻的粒子状态将会作为下一个时刻预测过程的输入,优化不只会影响当前时刻的粒子分布,也会影响下一时刻的粒子传播. 因此由数据可见,SFA-PF执行优化操作的次数明显少于PF执行重采样操作的次数. 综上所述,所提出算法具有很好的估计精度且达到同等精度需要的粒子数目和运算时间均更少.

5　结　论

粒子滤波存在固有的粒子权值退化问题,利用重采样操作可解决该问题,但是,传统重采样的实现方式会破坏粒子多样性,造成粒子贫化问题. 针对该问题,本文将改进的萤火虫算法与粒子滤波相结合,以最优粒子指导其他粒子朝高似然区域移动,通过评估粒子分布情况,控制优化强度. 进一步检测最优粒子周围的粒子密度,当粒子过度聚集时,引入弹性机制调整最优粒子周围的粒子密度,使得粒子分布更加合理. 优化完成后,对粒子权值进行补偿,使得优化前后的带权值粒子集从理论上服从同一分布,保证了贝叶斯滤波的理论完整性. 本文通过实验测试对影响改进算法性能的重要参数进行研究分析,并通过实验结果验证了所提出算法可改善粒子分布,解决粒子贫化问题,提高粒子滤波的综合性能.

参考文献(References)

[1] Šimandl M, Královec J, Söderström T. Advanced point-mass method for nonlinear state estimation[J]. Automatica, 2006, 42(7): 1133-1145.

[2] Jana N, Kumar S, Chatterjee K. Bayes estimation for exponential distributions with common location parameter and applications to multi-state reliability models[J]. Journal of Applied Statistics, 2016, 43(15): 2697-2712.

[3] Wang X X, Xu Z S, Gou X J, et al. Tracking a maneuvering target by multiple sensors using extended Kalman filter with nested probabilistic-numerical linguistic information[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(2): 346-360.

[4] Hien N V, He N V, Truong V T, et al. A MBSE application to controllers of autonomous underwater vehicles based on model-driven architecture concepts[J]. Applied Sciences, 2020, 10(22): 8293.

[5] 胡士强, 敬忠良. 粒子滤波算法综述[J]. 控制与决策, 2005, 20(4): 361-365.
(Hu S Q, Jing Z L. Overview of particle filter algorithm[J]. Control and Decision, 2005, 20(4): 361-365.)

[6] 夏楠, 王珏, 李博. 基于粒子滤波和交互多模型的移动定位方法[J]. 电子学报, 2019, 47(1): 197-203.
(Xia N, Wang J, Li B. A mobile localization method based on particle filter and interacting multiple models[J]. Acta Electronica Sinica, 2019, 47(1): 197-203.)

[7] Saha S, Gustafsson F. Particle filtering with dependent noise processes[J]. IEEE Transactions on Signal Processing, 2012, 60(9): 4497-4508.

[8] Bisis S K, Dempster A G. Approximating sample state vectors using the ESPT for computationally efficient particle filtering[J]. IEEE Transactions on Signal Processing, 2019, 67(7): 1918-1928.

[9] Garcia-Fernandez A F. Track-before-detect labeled multi-bernoulli particle filter with label switching[J]. IEEE Transactions on Aerospace and Electronic Systems, 2016, 52(5): 2123-2138.

[10] 陆耿虹, 冯冬芹. 基于粒子滤波的工业控制网络态势感知建模[J]. 自动化学报, 2018, 44(8): 1405-1412.
(Lu G H, Feng D Q. Modeling of industrial control network situation awareness with particle filtering[J]. Acta Automatica Sinica, 2018, 44(8): 1405-1412.)

[11] 李宏坤, 杨蕊, 任远杰, 等. 利用粒子滤波与谱峭度的滚动轴承故障诊断[J]. 机械工程学报, 2017, 53(3): 63-72.
(Li H K, Yang R, Ren Y J, et al. Rolling element bearing diagnosis using particle filter and kurtogram[J]. Journal of Mechanical Engineering, 2017, 53(3): 63-72.)

[12] Ma Y, Chen Y, Zhou X W, et al. Remaining useful life prediction of lithium-ion battery based on Gauss — Hermite particle filter[J]. IEEE Transactions on Control Systems Technology, 2019, 27(4): 1788-1795.

[13] Yang T, Laugesen R S, Mehta P G, et al. Multivariable

feedback particle filter[J]. Automatica, 2016, 71: 10-23.

[14] Zhang Z J, Chen J. Simultaneous data reconciliation and gross error detection for dynamic systems using particle filter and measurement test[J]. Computers & Chemical Engineering, 2014, 69: 66-74.

[15] 李天成, 范红旗, 孙树栋. 粒子滤波理论、方法及其在多目标跟踪中的应用[J]. 自动化学报, 2015, 41(12): 1981-2002.
(Li T C, Fan H Q, Sun S D. Particle filtering: Theory, approach, and application for multitarget tracking[J]. Acta Automatica Sinica, 2015, 41(12): 1981-2002.)

[16] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering[J]. Statistics and Computing, 2000, 10: 197-208.

[17] Yin S, Zhu X P. Intelligent particle filter and its application to fault detection of nonlinear system[J]. IEEE Transactions on Industrial Electronics, 2015, 62(6): 3852-3861.

[18] 胡昌华, 张琪, 乔玉坤. 强跟踪粒子滤波算法及其在故障预报中的应用[J]. 自动化学报, 2008, 34(12): 1522-1528.
(Hu C H, Zhang Q, Qiao Y K. A strong tracking particle filter with application to fault prediction[J]. Acta Automatica Sinica, 2008, 34(12): 1522-1528.)

[19] Zhang C, Taghvaei A, Mehta P G. A mean-field optimal control formulation for global optimization[J]. IEEE Transactions on Automatic Control, 2019, 64(1): 282-289.

[20] 滕飞, 薛磊, 李修和. 基于Student's t分布的自适应重采样粒子滤波算法[J]. 控制与决策, 2018, 33(2): 361-365.
(Teng F, Xue L, Li X H. Self-adaptive resampling particle filter based on student's t distribution[J]. Control and Decision, 2018, 33(2): 361-365.)

[21] 崔昊杨, 张宇, 周坤, 等. 基于仿生算法改进粒子滤波的SLAM算法精度预测[J]. 控制与决策, 2021, 36(1): 166-172.
(Cui H Y, Zhang Y, Zhou K, et al. Accuracy predition of SLAM algorithm based on bionic algorithm to improve particle filter[J]. Control and Decision, 2021, 36(1): 166-172.)

[22] Zhang X Y, Liu D, Lei B Y, et al. An intelligent particle filter with resampling of multi-population cooperation[J]. Digital Signal Processing, 2021, 115: 103084.

[23] 方正, 佟国峰, 徐心和. 粒子群优化粒子滤波方法[J]. 控制与决策, 2007, 22(3): 273-277.
(Fang Z, Tong G F, Xu X H. Particle swarm optimized particle filter[J]. Control and Decision, 2007, 22(3): 273-277.)

[24] Yin S, Zhu X P. Intelligent particle filter and its application to fault detection of nonlinear system[J]. IEEE Transactions on Industrial Electronics, 2015, 62(6): 3852-3861.

[25] 陈志敏, 吴盘龙, 薄煜明, 等. 基于自控蝙蝠算法智能优化粒子滤波的机动目标跟踪方法[J]. 电子学报, 2018, 46(4): 886-894.
(Chen Z M, Wu P L, Bo Y M, et al. Adaptive control bat algorithm intelligent optimization particle filter for maneuvering target tracking[J]. Acta Electronica Sinica, 2018, 46(4): 886-894.)

[26] Zhang Z R, Huang C Q, Ding D L, et al. Hummingbirds optimization algorithm-based particle filter for maneuvering target tracking[J]. Nonlinear Dynamics, 2019, 97(2): 1227-1243.

[27] Xie Y X, Wang S L, Fernandez C, et al. Improved gray wolf particle filtering and high-fidelity second-order autoregressive equivalent modeling for intelligent state of charge prediction of lithium-ion batteries[J]. International Journal of Energy Research, 2021, 45(13): 19203-19214.

[28] 田梦楚, 薄煜明, 陈志敏, 等. 萤火虫算法智能优化粒子滤波[J]. 自动化学报, 2016, 42(1): 89-97.
(Tian M C, Bo Y M, Chen Z M, et al. Firefly algorithm intelligence optimized particle filter[J]. Acta Automatica Sinica, 2016, 42(1): 89-97.)

[29] Yang X S. Firefly algorithm, stochastic test functions and design optimisation[J]. International Journal of Bio-Inspired Computation, 2010, 2(2): 78-84.

[30] 滕飞, 薛磊, 李修和. 基于KLD的蝙蝠算法优化自适应粒子滤波[J]. 控制与决策, 2019, 34(3): 561-566.
(Teng F, Xue L, Li X H. Adaptive particle filter with bat optimization based on KLD sampling[J]. Control and Decision, 2019, 34(3): 561-566.)

作者简介

田梦楚(1987–), 女, 博士后, 从事控制理论、智能优化等研究, E-mail: tianmengchu@163.com;

柳林燕(1985–), 女, 副教授, 博士, 从事数字化设计、数据采集等研究, E-mail: liulinyan@njust.edu.cn;

陈志敏(1986–), 男, 高级工程师, 博士, 从事控制理论、信息处理等研究, E-mail: chenzhimin@188.com;

方昱斌(1990–), 男, 博士后, 从事自适应控制理论、结构振动控制理论等研究, E-mail: fangyubin91@163.com.