

文章编号:1674-2869(2019)04-0392-07

# 一种带变异算子的粒子群优化粒子滤波降噪算法

方 璐,范东亮,张光宇,陈汉新\*

武汉工程大学机电工程学院,湖北 武汉 430205

**摘 要:**提出一种面向机械故障诊断非线性振动信号特征提取及实时滤波降噪的新型粒子群优化粒子滤波(NPSO-PF)算法,是基于带变异算子的粒子群优化粒子滤波算法。应用变异控制函数和操作算子,通过改善粒子滤波(PF)算法粒子贫乏、利用率不高等问题,加速粒子集收敛,减少整个算法运行的时间。仿真结果通过与PF算法和PSO-PF算法相比,论证了提出的NPSO-PF算法具有更低的均方根误差、更短的运行时间、更高的信噪比和更稳定的滤波性能。

**关键词:**变异算子;粒子群优化;粒子滤波;降噪

中图分类号:TP274.2 文献标识码:A doi:10.3969/j.issn.1674-2869.2019.04.017

## Particle Filter Algorithm Based on Particle Swarm Optimization with Mutation Operator for Noise Reduction

FANG Lu, FAN Dongliang, ZHANG Guangyu, CHEN Hanxin\*

School of Mechanical and Electrical Engineering, Wuhan Institute of Technology, Wuhan 430205, China

**Abstract:** A novel particle swarm optimization particle filter (NPSO-PF) algorithm was proposed for the real-time filtering noise reduction of nonlinear vibration signals and feature extraction in fault diagnosis of mechanical system. The particle filter (PF) algorithm was optimized by the particle swarm with the mutation operator, by use of the mutation control function and operator to improve the problems of particle poverty and low utilization rate. The convergence of the particle sets is accelerated, and the running time of the proposed algorithm was reduced. By the comparisons of PF, PSO-PF and NPSO-PF algorithms, the simulation results show that the proposed NPSO-PF algorithm has the advantages of being less root mean square errors, shorter running time, higher signal /noise ratio and with more stable filtering performance.

**Keywords:** mutation operator; particle swarm optimization; particle filter; noise reduction

去噪是信号处理的重要内容,去噪的好坏对后续诊断结果的分析有着重要影响,研究非线性信号的去噪方法是近年来关注的重点<sup>[1]</sup>。黄名钊等<sup>[2]</sup>使用基于粒子群优化(particle swarm optimization, PSO)算法的参数小波阈值处理方法实现了对非线性信号的降噪处理。Ananth等<sup>[3]</sup>使用基于粒子群优化的模糊滤波器,以除去信号中的高密度图像脉冲噪声。这些算法都能实现对非线性信号

的去噪,然而使用PSO算法结构,有易陷入局部最优解的缺陷,特别是处理多局部极值的信号,收敛速度和精度不高。为改善粒子群算法的缺陷,张荣光等<sup>[4]</sup>将粗糙集理论引入粒子群算法中,构建了离散粒子群优化方法,加强了搜索能力。Kar等<sup>[5]</sup>提出了一种基于引力搜索算法(gravitational search algorithm, GSA)的粒子群优化算法(GSA-PSO),用于2种常用模拟电路的优化设计,

收稿日期:2019-04-06

基金项目:湖北省科技厅重大专项(2016AAA056);湖北省教育厅重大项目(Z20101501);国家自然科学基金(51775390)

作者简介:方 璐,硕士研究生。E-mail:fanglu.e@foxmail.com

\*通讯作者:陈汉新,博士,教授。E-mail: pg01074075@163.com

引文格式:方璐,范东亮,张光宇,等.一种带变异算子的粒子群优化粒子滤波降噪算法[J].武汉工程大学学报,2019,41(4):392-398.

取得了不错的效果。

粒子滤波(particle filter, PF)算法在处理非线性问题上具有的独特优势,使得粒子滤波方法成为近年来国内外关注的重点<sup>[6]</sup>。已被广泛应用于故障诊断、导航定位、无线通讯、机器人定位、视觉跟踪等诸多领域<sup>[7]</sup>。然而常规粒子滤波采样过程中采用的采样策略和选择的次优重要性函数,使得其存在粒子贫乏和计算效率低等问题。用智能优化算法来优化PF算法的采样结构,是现阶段PF研究的热点,其中基于PSO算法的PF算法是智能优化粒子滤波算法的代表之一。王尔申等<sup>[8]</sup>提出了一种基于边缘粒子滤波和粒子群优化的联合参数和状态估计的双估计方法,提高了PF算法的性能。然而没有解决PSO算法易陷入局部最优的问题,使得运算结果不稳定,去噪效果较差。为了改善PSO-PF算法的性能,王尔申等<sup>[9]</sup>将混沌序列引入PSO-PF算法中,还有学者将遗传重采样方法和PSO-PF算法相结合<sup>[10-11]</sup>,这些措施都提高了PSO-PF算法的滤波性能,然而随着算法的结构变得复杂,计算量也会加大,并不适合实时的去噪处理。

本文提出一种适用于滤波降噪的新型粒子群优化粒子滤波(novel particle swarm optimization particle filter, NPSO-PF)算法。该算法在采样过程中通过遗传变异控制函数,使粒子优化过程分为前后期,加快了粒子集的收敛,提高了运算速度。同时通过变异操作,控制粒子的多样性,避免了粒子贫乏,利用率不高的问题。将算法用于非线性信号的降噪具有稳定,快速,有效的优点。本文首先介绍PF、PSO和PSO-PF算法原理并分析其缺陷,然后构造出NPSO-PF算法。为验证变异算子的性能,先进行迭代仿真实验,为验证NPSO-PF算法的滤波性能,进行了状态估计实验,为验证NPSO-PF算法的滤波降噪,进行了去噪仿真实验。

## 1 常规粒子滤波算法及其不足

### 1.1 粒子滤波原理

PF算法的本质是基于蒙特卡洛法的最优贝叶斯估计。主要根据系统状态向量的经验条件分布产生的粒子样本来模拟给定系统的当前状态,然后根据各个时刻的测量值不断修正和调整粒子的权值及位置,最后通过调整后的粒子信息修正最初的经验条件分布以获得最精确的估计值<sup>[12]</sup>。

选取非线性动态系统状态方程和观测方程模型如下:

$$\begin{cases} x_k=f(x_{k-1})+v_{k-1} \\ y_k=h(x_k)+n_k \end{cases} \quad (1)$$

其中  $x_k$  为  $k$  时刻的系统状态,  $y_k$  为观测输出,  $v_{k-1}$  为过程噪声,  $n_k$  为观测噪声,  $f(\bullet)$  和  $h(\bullet)$  分别为非线性函数。

PF的具体实现步骤如下:

步骤1:初始化。在  $k=0$  时刻,根据  $P(x_0)$  分布采样得到  $\{x_0^i, i=1, 2, \dots, N\}$  即  $x_0^i \sim P(x_0)$ 。

步骤2:重要性权值计算。采样  $\{x_k^i, i=1, 2, \dots, N\} \sim q(x_k|x_{k-1}, y_k)$ , 计算重要性权值

$$\omega_k^i = \omega_{k-1}^i \frac{P(y_k|x_k^i)P(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)}, i=1, 2, \dots, N \quad (2)$$

归一化重要性权值

$$\tilde{\omega}_k^i = \omega_k^i / \sum_{i=1}^N \omega_k^i \quad (3)$$

步骤3:重采样。根据重要性权值  $\tilde{\omega}_k^i$  从  $\{x_k^i, i=1, 2, \dots, N\}$  集合中重新采样得到  $k$  时刻的新粒子集合  $\{x_0^i, i=1, 2, \dots, N\}$ , 根据  $\omega_k^i = \tilde{\omega}_k^i = 1/N$  得到新的粒子权值。

步骤4:输出。状态估计为:

$$\tilde{x}_k = \sum_{i=1}^N \omega_k^i \tilde{x}_k^i \quad (4)$$

### 1.2 常规粒子滤波的不足

常规PF算法的核心算法的主要缺陷是会出 现样本贫化的问题。位于先验概率分布尾部的观测概率或似然函数,使权值更新后的粒子权值会很小。差异很小的粒子表示的后验概率,重采样后,粒子样本集的多样性会很小甚至只剩单一样本<sup>[13]</sup>,从而产生粒子贫乏的问题。

当系统初始状态未知时,需要采集大量粒子样本才能实现系统的状态估计是常规粒子滤波的另一个缺陷。采集的粒子数目很小,粒子很难收敛到真实状态附近,实现准确的估计;增大粒子集的数目,计算量增大,计算效率低下,不适合实时监测的问题。

## 2 粒子群优化粒子滤波算法及其不足

### 2.1 粒子群优化算法原理

PSO算法是Kennedy和Eberhart等于1995年最先提出的,它通过群体中粒子个体间的协作和竞争来有效的实现复杂空间全局最优解的搜索,是一类模拟群体智能行为的优化算法<sup>[14]</sup>。具体可

表述为:随机初始化一个  $N$  维空间数量为  $m$  的粒子群。其中第  $i$  个粒子位置为  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ , 其速度  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$  它的个体极值为  $P_i = (p_{i1}, p_{i2}, \dots, p_{in})$ , 种群的全局极值为  $G_i = (g_{i1}, g_{i2}, \dots, g_{in})$ 。确定上述两个极值后,就可以根据下述公式来不断更新粒子的速度和位置:

$$\begin{cases} V_i^{k+1} = \omega \cdot V_i^k + c_1 \cdot r \cdot (P_i^k - X_i^k) + c_2 \cdot r \cdot (G_i^k - X_i^k), \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \quad (5)$$

式(5)中:  $r$  是介于  $(0, 1)$  区间的随机数,  $c_1$  和  $c_2$  统称为学习因子或加速度因子, 一般  $c_1 = c_2 = 1.4692$ 。  $\omega$  为惯性系数。

## 2.2 PSO-PF 算法

将 PSO 算法融合到 PF 算法中, 将最新的观测值引入到粒子样本的采样过程中, 并定义粒子群优化算法中的适应度函数为:

$$f_{it} = \exp\left\{-\frac{1}{2R_t}(Y_{\text{new}} - Y_{\text{pred}})\right\} \quad (6)$$

其中,  $R_t$  是观测噪声方差,  $Y_{\text{new}}$  是最新的观测值,  $Y_{\text{pred}}$  是预测观测值。

在采样过程中, 如果粒子集分布在真实状态附近, 根据式(6)可知, 粒子的适应度值会很大。而不在真实状态附近的粒子其适应度值就会很小。此时利用 PSO 寻优算法, 根据当前最优值和公式(5)不断地更新每个粒子的速度与位置, 使得粒子不断地向真实状态附近移动。从而避免遗漏重要粒子, 既保证了粒子的多样性, 又提高了有效粒子的使用率。

## 2.3 PSO-PF 算法的不足

PSO-PF 算法的实质是在 PF 算法的采样过程中先利用 PSO 算法将粒子集向真实状态附近集中, 在重采样过程中能有效的采集对估计真正起作用的粒子, 从而提高 PF 的效率和精度。然而传统的 PSO 算法在迭代后期, 粒子的速度越来越小使粒子在局部空间范围内反复搜索, 容易出现局部最优现象。而随着所有粒子都随公式(5)不断向局部最优值移动, 使粒子丧失了多样性, 从而降低了 PF 的精度和效率。

# 3 带变异算子的 PSO 优化 PF 算法

## 3.1 带变异算子的 PSO 算法

为了改善常规 PSO 优化算法易陷入局部最优解的问题, 鉴于遗传变异算法的变异思想, 将遗传算法的操作算子引入到 PSO 算法中<sup>[15]</sup>。其具体做

法是引入一个变异控制函数, 用来控制变异的粒子数目, 以保持种群内部的多样性, 进而避免过早的在局部最优处收敛。

引入的变异控制函数为:

$$y(h) = \left(1 - \left(h/h_{\max}\right)^\alpha\right)^\beta \quad (7)$$

其中  $h$  表示当前迭代次数,  $h_{\max}$  表示最大迭代次数,  $\alpha$ 、 $\beta$  表示控制系数。

根据式(7), 计算变异算子的控制率:

$$u = m \cdot y(h) \quad (8)$$

其中  $u$  是变异率,  $m$  是预设变异率, 设定后不变。

根据式(8), 计算变异操作的粒子数目:

$$M = [N \cdot u] \quad (9)$$

根据式(9), 随机从粒子群中选取要进行变异操作的粒子, 用实数编码。假设第  $k$  个粒子被选中进行变异操作, 如  $X_k = (x_{k1}, x_{k2}, \dots, x_{kD})$  其中第  $j$  个元素进行变异, 则操作策略为:

$$x_{kj} = x_{kj} + r \cdot y(h) \quad (10)$$

其中  $r \in (-a, a)$  是随机数。

动态惯性系数的计算为:

$$w = w_{\min} + (w_{\max} - w_{\min}) \cdot y(h) \quad (11)$$

其中,  $w_{\max}$  表示最大惯性系数,  $w_{\min}$  表示最小惯性系数。

变异算子控制 PSO 算法的运行是通过粒子的变异率, 动态惯性系数和变异操作来实现的。而变异率是由式(7)确定, 动态惯性系数是由式(11)确定, 变异操作是由式(8)~式(10)来完成的, 其都与算法的迭代次数相关的。在算法迭代的前期, 控制函数和动态惯性系数取较大的值, 进行变异操作的粒子数目和变异尺度都比较大, 既保证了种群内部粒子的多样性, 又使得 PSO 算法大范围的在解空间内进行搜索。在算法的后期, 控制函数和动态惯性系数取较小的值, 进行变异操作的粒子数目和变异尺度都很小, 甚至不进行变异操作, 既使种群内部的粒子能很快进行收敛, 又使 PSO 算法小范围的在解空间内进行搜索。因此, 算法的收敛速度和收敛精度都得到了提高。NPSO 算法具体步骤为:

步骤 1: 初始化群体数目  $N$ , 学习因子  $C_1$  和  $C_2$ , 惯性系数  $w$ , 初始粒子的位置  $X_i$  和速度  $V_i$ , 最大迭代次数  $h_{\max}$ , 预设变异率  $m$ 。

步骤 2: 按式(7)计算控制函数  $y(h)$ , 按式(11)计算惯性系数  $w$ 。

步骤 3: 根据式(8)计算变异率  $u$ , 根据式(9)



计算变异粒子数  $M$  ,按式(10)进行变异操作。

步骤4:对于每个粒子,按式(6)适应度值  $f_{it}[i]$  ,并与其个体最优值  $P_i$  进行比较如果  $f_{it}[i]>P_i$  那么令  $P_i=f_{it}[i]$  。

步骤5:求解全局最优值  $G_i$  ,即如果  $P_i>G_i$  那么令  $G_i=P_i$  。

步骤6:根据式(5)更新粒子的速度  $V_i$  和位置  $X_i$  ;

步骤7:当达到最大迭代次数或者粒子群的最优值符合某阈值  $\varepsilon$  时,则输出结果,否则返回第二步。

3.2 NPSO-PF 算法

NPSO-PF 算法的原理是利用带变异算子的 PSO 算法快速准确的收敛性能来优化 PF 的采样过程。通过适应度值使所有粒子向最优粒子移动,以改善粒子集的分布情况,使粒子集都集中在真实状态附近,从而提高了滤波的精度和效率,具体的步骤为:

步骤1:初始化。在  $k=0$  时刻,根据  $P(x_0)$  分布采样得到  $\{x_0^i, i=1, 2, \dots, N\}$  即  $x_0^i \sim P(x_0)$  。步骤2:NPSO-PF 优化粒子集分布。结合利用3.1节讲述的 NPSO-PF 实施步骤对粒子群进行寻优,当达到迭代次数或者粒子群的最优值符合某阈值  $\varepsilon$  时,输出粒子,即可得到新的粒子样本  $\hat{x}_k^i$  。

步骤3:重要性权值计算。采样  $\{\hat{x}_k^i, i=1, 2, \dots, N\} \sim q(x_k|x_{k-1}, y_k)$  ,计算重要性权值根据式(2)计算新的重要性权值,根据式(3)完成权值归一化。

步骤4:重采样。根据新的重要性权值从  $\{\hat{x}_k^i, i=1, 2, \dots, N\}$  集合中重新采样,该过程同传统粒子滤波算法一致。

步骤5:输出。根据式(4)得到更新后的状态估计。

4 试验仿真与结果分析

4.1 标准 PSO 和引入变异算子的 PSO 仿真对比

为了验证加入变异算子的粒子群优化算法具有不陷入局部最优值的性能以及其准确性和收敛精度,给定非线性函数为:

$$f(x)=-c_1 \exp \left(-0.2 \int_n^1 \frac{1}{n} \sum_{j=1}^n x_j^2\right)-\exp \left(\frac{1}{n} \sum_{j=1}^n \cos \left(2 \pi x_j\right)\right)+c_1+\delta$$

(12)

取  $c_1=20, n=2, \delta=2.71282$  时,该函数为有很

多局部极小值的 Ackley 函数。用此时的  $f(x)$  为目标函数,可以有效的测试算法的收敛性能。分别用引入变异算子的 PSO 算法和标准 PSO 算法对函数进行寻优。根据相关参考文献和经验以及多次试验,其中仿真参数分别为:粒子个数  $N=30$  ,适应度阈值  $\varepsilon=0.001$  ,最大迭代数  $h_{\max}=100$  ,变异控制系数  $\alpha, \beta$  分别为 1.7 和 10,惯性权重  $\omega$  最大和最小值分别为 0.729 8 和 0.1,加速度常数  $c_1$  和  $c_2$  分别为 1.469 2。计算机的处理器是 Inter(R) Core(TM) i7-4970 CPU @3.60 GHz 3.60 GHz。随机取某次仿真结果如图 1 所示。

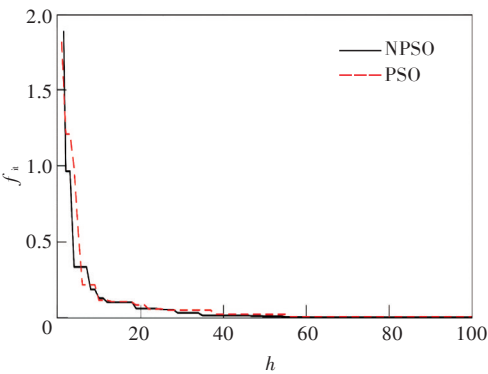


图1 两种算法的迭代曲线  
Fig. 1 Iteration curves of two algorithms

从图1可以看出,带变异算子的 PSO 算法在早期和后期的收敛速度快于标准 PSO 算法,中期略低于标准 PSO 算法。为了比较2种算法的整体收敛性能,在上述实验函数和参数不变的情况下,分别进行了20次实验,得到了2种算法的迭代次数和收敛函数值的平均值。MATLAB 运算结果表明,标准 PSO 算法的平均收敛迭代次数为60次,引入变异算子的 PSO 算法为40次。收敛时的适应度函数平均值分别为0.010和0.001。结果表明,带变异算子的 PSO 算法在函数优化中的应用优于标准的 PSO 算法。

4.2 粒子滤波仿真试验及对比

为验证3种算法的滤波和估计性能,试验选取的动态非线性系统的状态方程和观测方程。

状态方程:

$$x_k=0.5 x_{k-1}+\frac{25 x_{k-1}}{1+x_{k-1}^2}+8 \cos (1.2(k-1))+v_{k-1}$$

(13)

观测方程:

$$y_k=\frac{x_k^2}{20}+n_k$$

(14)

其中  $v_{k-1}$  和  $n_k$  为零均值高斯噪声。 $x_k$  为状态变量,  $y_k$  为观测变量。 $k$  为模拟步长。利用 PF

算法、PSO-PF算法、NPSO-PF算法分别对该非线性系统进行状态估计和跟踪,为比较3种算法的性能,粒子数分别取 $n=50,500,1\,000$ ,取初始状态概率密度函数为 $N(0,2)$ ,测量噪声方差和过程噪声方差分别为 $R_l=1\times10^{-3}$ 和 $Q=1\times10^{-2}$ ,模拟步长为 $k=50$ ,粒子群优化算法相关参数同上,估计值和真实值之间差值的绝对值用 $\sigma$ 表示。随机选取不同粒子数目下某次仿真结果如图2所示。其中均方根误差 $R$ 的计算公式为:

$$R=\left[\frac{1}{T}\sum_{k=1}^T(x_k-\hat{x}_k)^2\right]^{1/2}$$

为了比较3种算法的整体性能,将仿真实验进行30次,测得在3种不同粒子数目下3种算法的

30次运算的均方根误差和程序运行时间的平均值,如表1和表2所示。

从图2可知,NPSO-PF算法估计出的粒子状态更接近粒子的真实状态,估计误差也更小,随着估计的粒子数目的增加效果越来越明显。而PSO-PF算法的估计性能略好于PSO算法,并随着粒子数目的增加效果差异越来越小。从表1和表2的结果来看,NSPO-PF算法的平均估计误差和运行时间要低于另外两者,随着粒子数目的增加,效果更明显。而PSO-PF算法只是略好于PF算法,随着粒子数目的增加两者性能基本相同。综上可知,NPSO-PF算法比PSO-PF和PF算法具有更好的、更稳定的估计性能。

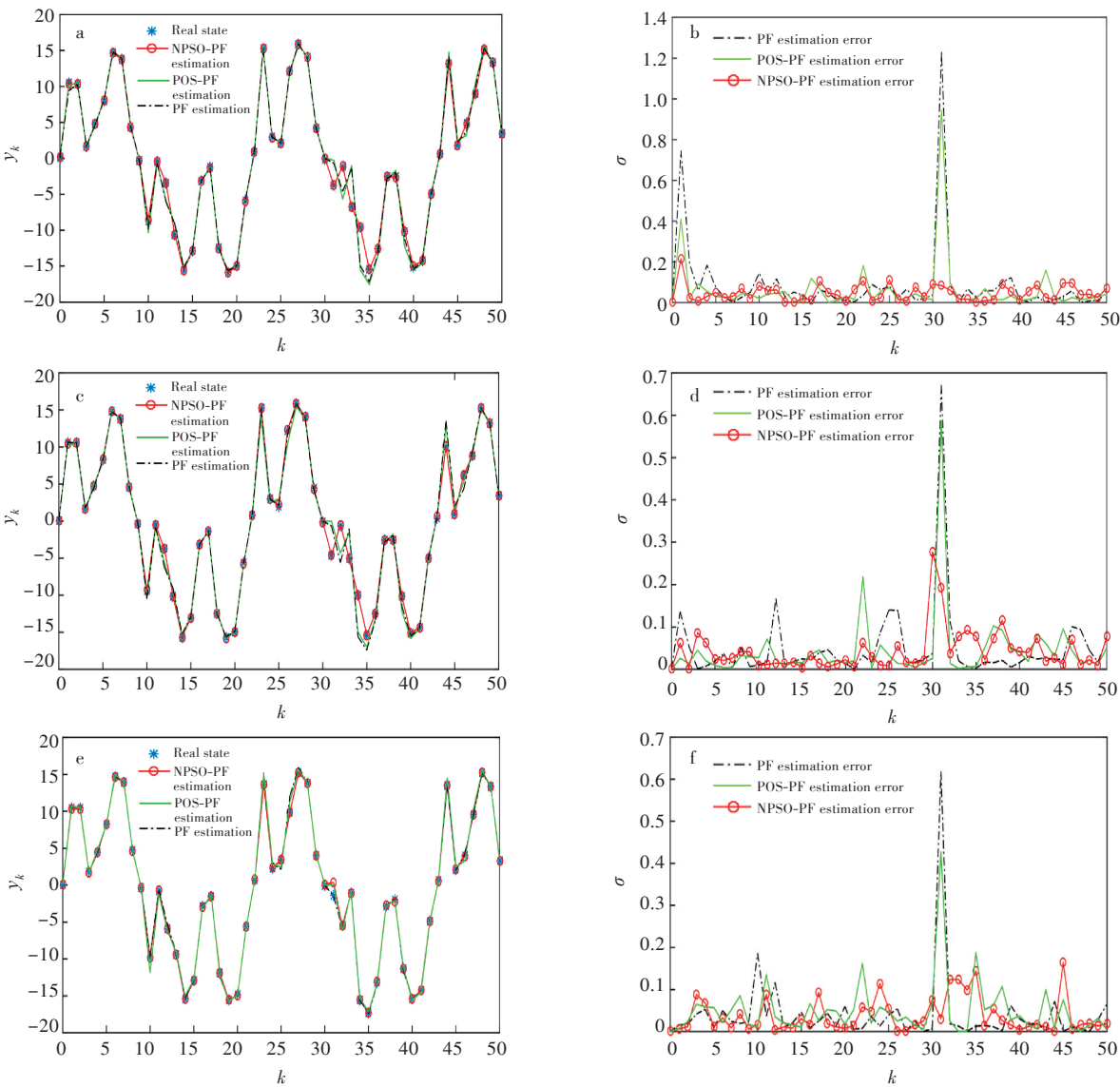


图2 (a) $n=100$ , (c) $n=500$ , (e) $n=1\,000$ 状态估计曲线; (b) $n=100$ , (d) $n=500$ , (f) $n=1\,000$ 误差曲线

Fig. 2 (a) $n=100$ , (c) $n=500$ , (e) $n=1\,000$  state estimation curves;  
(b) $n=100$ , (d) $n=500$ , (f) $n=1\,000$  error curves

表1 三种算法的R

Tab. 1 R of three algorithms

算法	均方根误差		
	n=100	n=500	n=1 000
PF	0.262 7	0.110 1	0.097 4
PSO-PF	0.154 7	0.096 6	0.080 0
NPSO-PF	0.079 7	0.057 2	0.056 1

表2 三种算法的运行时间

Tab. 2 Running times of three algorithms

算法	程序运行时间/s		
	n=100	n=500	n=1 000
PF	0.860 8	0.893 0	1.074 8
PSO-PF	0.759 7	1.051 9	1.111 1
NPSO-PF	0.774 4	0.790 4	0.924 7

4.3 降噪实验仿真

利用上述3种方法对下列信号进行降噪仿真实验,选取的原始信号方程如下:

$$Y(t)=0.01\cos(800\pi t-5)\mathrm{e}^{-0.5\left(\frac{t-0.01}{0.004}\right)^2}+0.003\cos(1\,000\pi t-5)\mathrm{e}^{-0.5\left(\frac{t-0.03}{0.005}\right)^2}+0.01\cos(1\,200\pi t-5)\mathrm{e}^{-0.5\left(\frac{t-0.05}{0.006}\right)^2}+0.005\cos(1\,400\pi t-5)\mathrm{e}^{-0.5\left(\frac{t-0.07}{0.005}\right)^2}+0.002\cos(1\,600\pi t-5)\mathrm{e}^{-0.5\left(\frac{t-0.09}{0.004}\right)^2}\quad(16)$$

加入大小和原始信号一样的随机噪音信号后得:

$$Z(t)=Y(t)+0.005R(t)\quad(17)$$

用原始信号方程(16),加噪信号方程(17)分别代替4.2节中状态方程(13)和观测方程(14),从而构成3种算法中的空间模型,其中 $Y(t)$ 为原始信号峰值变量, $Z(t)$ 为加噪信号峰值变量, $t$ 为模拟时间步长。其中取初始概率密度函数为 $N(0,2)$ ,状态初始值为1,测量噪声方差 $R_t=1\times10^{-4}$ ,过程噪声方差 $Q=1\times10^{-5}$ ,模拟时间步长为 $t=204\,8$ , PSO算法相关参数同上,分别对4.2节中3种粒子数目进行实验仿真,现将粒子数 $n=100$ 时分别用PF算法, PSO-PF算法和NPSO-PF降噪后的结果展示如图3所示。

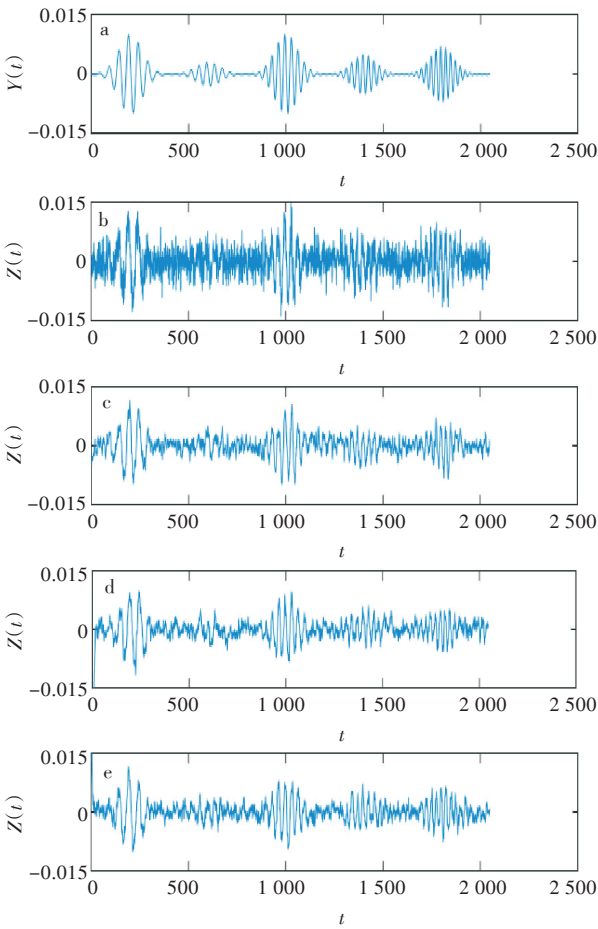


图3 降噪仿真实验结果:(a)原始信号, (b)原始信号加噪,(c)PF算法降噪, (d)PSO-PF算法降噪,(e)NPSO-PF算法降噪

Fig. 3 Simulation experiment results of noise reduction: (a) original signal, (b) original signal plus noise, (c) noise reduction by PF algorithm, (d) noise reduction by PSO-PF algorithm, (e) noise reduction by NPSO-PF algorithm

从图3可以看出,3种算法都具有相当不错的降噪效果,用NPSO-PF算法降噪后的信号峰值更清晰。信噪比是指原始信号与纯噪音信号的比值,信噪比数值越大表示噪音越小,因此信噪比是衡量降噪效果好坏的重要指标。为了更好比较3种算法的降噪性能,对上述实验结果进行信噪比S分析。取50次降噪实验3种算法在3种粒子数目下的信噪比均值进行分析,其结果如表3所示,信

表3 三种算法的仿真降噪结果对比

Tab. 3 Comparisons of simulated noise reduction results from three algorithms

粒子数	滤波前 S	滤波后 S			S 提高值		
		PF	PSO-PF	NPSO-PF	PF	PSO-PF	NPSO-PF
n=100	-0.320 0	1.031 4	1.862 6	3.033 2	1.351 4	2.182 6	3.353 2
n=500	-0.225 6	4.252 6	4.399 5	6.040 5	4.478 2	4.625 1	6.266 1
n=1 000	-0.242 7	4.353 5	4.446 9	6.767 7	4.596 2	4.689 6	7.010 4

噪比计算公式为：

$$S=10\lg\left(\frac{\sum x^2}{\sum (z-x)^2}\right) \tag{18}$$

从表3可以看出经NPSO-PF算法滤波后的信噪比在3种粒子数目的试验下都要比其他2种算法高,而经PSO-PF和PF算法滤波后的信噪比基本一样。说明NPSO-PF算法的降噪性能要优于其他2种算法,而PSO-PF和PF算法的降噪性能基本一样。NPSO-PF算法的信噪比提高值普遍大于其他两种算法,并且随着粒子数目的增加有持续上升的趋势,而PSO-PF和PSO算法基本达到饱和。综上所述,NPSO-PF算法具有更好更稳定的滤波降噪性能。

5 结 语

本文将变异算子带入PSO算法中,构成新型NPSO算法,改善了PSO算法的结构,优化了传统PSO算法易于陷入局部最优解的问题,提高了PSO算法的性能。将其与粒子滤波算法相结合构成NPSO-PF算法,该算法不仅具有更好的估计性能,而且在滤波降噪方面也有更好的效果。仿真实验结果证明NPSO-PF算法适用于信号的滤波降噪处理。

参考文献

[1] 刘天宇,吴金斌,周世健. 附加斜率绝对值法的模态相关EMD-ICA滤波降噪算法及GPS应用研究[J]. 江西科学, 2017(4):509-515.

[2] 黄名钊,高伟. 粒子群优化改进小波阈值函数的去噪研究[J]. 软件, 2017(9):182-186.

[3] ANANTH C, VIVEK T, SELVAKUMAR S, et al. Impulse noise removal using improved particle swarm optimization[J]. Social Science Electronic Publishing, 2017(3):366-370.

[4] 张荣光,胡晓辉,宗永胜,等. 禁忌离散粒子群优化的粗糙集属性约简算法[J]. 小型微型计算机系统, 2017(8):1840-1844.

[5] MALLICK S, KAK R, MANDAL D, et al. Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization[J]. International Journal of Machine Learning and Cybernetics, 2017(8):309-331.

[6] 张威武,郭明香,贺元恺,等. 一种改进的蝴蝶算法优化粒子滤波算法[J]. 西安科技大学学报, 2019(1):119-123.

[7] 陆湛. 基于变分贝叶斯学习的粒子滤波研究[D]. 广州:华南理工大学, 2017.

[8] HAVSNGI R. Joint parameter and state estimation based on marginal particle filter and particle swarm optimization [J]. Circuits, Systems, and Signal Processing, 2018(37):3558-3575.

[9] 王尔申,庞涛,曲萍萍,等. 基于混沌的改进粒子群优化粒子滤波算法[J]. 北京航空航天大学学报, 2016, 42(5):885-890.

[10] TANG G, WEI B, WU D , et al. The optimal wavelengths for light absorption spectroscopy measurements based on genetic algorithm-particle swarm optimization [J]. Journal of Applied Spectroscopy, 2018, 85 (1) : 109-118.

[11] 韩雪,程奇峰,赵婷婷,等. 基于粒子滤波重采样与变异操作的改进粒子群算法[J]. 计算机应用, 2016, 36(4):1008-1014.

[12] 杨柳. 基于RBF-PF和粒子群优化小波神经网络在齿轮箱故障诊断中的研究[D]. 武汉:武汉工程大学, 2016.

[13] 袁开宇,储珺,冷四军,等. 基于特征融合的粒子群优化粒子滤波跟踪方法[J]. 南昌航空大学学报(自然科学版), 2015(1):33-41.

[14] LIU Z , LI H, ZHU P . Diversity enhanced particle swarm optimization algorithm and its application in vehicle lightweight design [J]. Journal of Mechanical Science and Technology, 2019, 33(2):695-709.

[15] 余仁波,赵修平,孟凡磊. 一种带变异算子的PSO算法[J]. 舰船电子工程, 2016, 36(10):26-29.

本文编辑:陈小平