

基于混沌的 PSO 粒子滤波算法

李 明, 逢 博, 年福忠

(兰州理工大学计算机与通信学院, 兰州 730050)

摘 要: 粒子群优化(PSO)粒子滤波算法容易陷入局部最优, 从而降低算法精度。针对该问题, 提出一种基于混沌的 PSO 粒子滤波算法。该算法通过混沌搜索算法找到全局最优位置, 驱散聚集在局部最优的粒子群, 使其向全局最优位置靠近, 增加有效估计粒子数, 抑制粒子退化与枯竭问题。仿真结果表明, 与传统的粒子滤波算法和 PSO 粒子滤波算法相比, 改进算法的估计精度有较大提高。

关键词: 粒子滤波; 混沌搜索算法; 粒子群优化算法; 局部最优; 粒子退化; 粒子枯竭

Particle Swarm Optimization Particle Filtering Algorithm Based on Chaotic

LI Ming, PANG Bo, NIAN Fu-zhong

(School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China)

【Abstract】 Particle Swarm Optimization Particle Filtering(PSOPF) algorithm is easy to fall into local optimum, so the particles can not move to the global optimal location, and reduce algorithm precision. According to this problem, the paper proposes a Particle Swarm Optimization Particle Filtering based on Chaotic(CPSOPF) algorithm. Through the chaotic search algorithm, this algorithm makes particles find the global optimal location, dispels particle swarm at local optimum location and makes them move to global optimal location. So the number of effective particles increases, which can effectively restrain particles degradation and exhaustion. Simulation results show that the CPSOPF algorithm can remarkably improve the estimation accuracy compared with of the conventional Particle Filtering(PF) and the traditional PSOPF algorithm.

【Key words】 Particle Filtering(PF); chaotic search algorithm; Particle Swarm Optimization(PSO) algorithm; local optimal; particle degeneracy; particle impoverishment

DOI: 10.3969/j.issn.1000-3428.2012.08.044

1 概述

粒子滤波(Particle Filtering, PF)是一种基于蒙特卡罗思想与贝叶斯估计理论的统计滤波方法^[1]。其依据大数定理, 采用蒙特卡罗方法来解决贝叶斯估计中的积分问题, 即序贯蒙特卡罗方法。理论上适用于任何可用状态空间模型表示的非高斯背景下的非线性随机系统, 其精度可逼近最优估计。近年来在国防军事^[2]、工业生产^[3]、经济预测^[4]、视频监控^[5]等领域得到成功的应用。

由于常规的粒子滤波没有考虑当前的量测, 因此从重要性密度函数中采样得到的粒子与从真实后验概率密度函数采样得到的粒子有很大偏差, 当重要性权重的方差随着时间递增时, 权重只集中在少数的粒子上, 其他粒子的权值非常小, 可以忽略不计, 从而使大量计算都浪费在对估计不起作用的粒子上, 使得粒子集无法表达真实的后验概率分布, 这就是粒子的退化现象。为了克服退化现象, 采用重采样方法, 剔除权值较小的粒子, 复制权值较大的粒子, 又会产生粒子枯竭现象。

针对上述粒子滤波存在的问题, 科研人员提出了许多改进方法。文献[6]提出的无味粒子滤波(Unscented Particle Filtering)采用无味卡尔曼滤波来产生重要性密度函数, 因为引入了最新的量测值, 所以提高了粒子滤波的性能, 但也大大增加了计算量。文献[7]将遗传算法引入粒子滤波中, 提升了重采样粒子的多样性, 但算法存在早熟现象。文献[8]提出的粒子群优化粒子滤波(Particle Swarm Optimization Particle

Filtering, PSOPF)算法采用 PSO 算法将粒子向高似然区驱动, 从而改善了粒子退化与枯竭现象, 但由于 PSO 算法本身易陷入局部最优, 很难将粒子驱动到全局最优位置, 因此在一定程度上并没有很好地改善粒子滤波算法的估计精度。

本文提出了一种基于混沌的粒子群优化粒子滤波(Particle Swarm Optimization Particle Filtering based on Chaotic, CPSOPF)算法。当 PSOPF 算法陷入局部最优时, 通过混沌搜索(Chaotic Search)算法找到全局最优位置, 驱散聚集在局部最优位置的粒子群, 使 PSOPF 算法适时地跳出局部最优, 驱动粒子向全局最优位置移动, 能够增加有效粒子的数目, 提高粒子多样性, 有效地抑制粒子退化与枯竭现象, 提高算法的精度。同时, 用于估计的有效粒子数增加, 对估计不起作用的粒子数减少, 从而减少了对估计不起作用的粒子的计算, 在同等粒子数下, 在保证算法高精度的同时又可提高算法的效率。

2 PSO 粒子滤波算法

PSO 算法^[9]是一种依据粒子的适应值进行操作, 通过粒子间的合作与竞争产生的群体智能指导优化搜索。把粒子看作一个没有重量与体积的微粒, 在 n 维空间中以一定的速度飞行, 飞行速度由个体的飞行经验与群体的飞行经验动态调

基金项目: 甘肃省自然科学基金资助项目(1014RJZA028)

作者简介: 李 明(1959—), 男, 教授, 主研方向: 智能信息处理; 逢 博, 硕士研究生; 年福忠, 副教授

收稿日期: 2011-08-11 **E-mail:** pangbo59021620@163.com

整。PSO 算法的实质是利用本身信息、个体极值信息和全局极值信息这 3 种信息指导粒子的下一步迭代位置。近年来有学者将 PSO 算法引入粒子滤波中, 如文献[8]提出的粒子群优化粒子滤波算法, 通过驱动粒子向高似然区域移动, 来增加粒子多样性, 在缓解粒子退化与枯竭问题上取得了一定的成果。

PSOPF 算法可表述为: 首先将最新的量测值引入采样过程, 并定义适应度函数为:

$$z_k \sim \text{fitness} = \exp\left[-\frac{1}{2R_k}(z_k - \hat{z}_{k|k-1})^2\right]$$

对粒子的初始化: 从先验概率密度 $p(x_0)$ 产生粒子群 $\{x_0^i\}_{i=1}^N$, 所有粒子权值为 $1/N$ 。计算重要性权值: $w_k^i = w_{k-1}^i \exp\left[-\frac{1}{2R_k}(z_k - \hat{z}_{k|k-1})^2\right]$ 。再引入 PSO 算法, 用式(1)和式(2)更新每个粒子的速度和位置, 使粒子不断地靠近真实状态:

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + c_1r_{1j}[p_{ij}(t) - x_{ij}(t)] + c_2r_{2j}[p_{gj}(t) - x_{ij}(t)] \quad (1) \\ x_{ij}(t+1) &= x_{ij}(t) + v_{ij}(t+1) \quad (2) \end{aligned}$$

其中, 下标 i 表示第 i 个粒子; j 表示粒子的第 j 维; t 表示第 t 代; w 称为惯性因子, 用惯性权重来控制前面速度对当前速度的影响; c_1 、 c_2 表示加速度, $c_1 = c_2 = 2$, c_1 调节粒子向自身最优位置飞行, c_2 调节粒子向全局最优位置飞行; r_1 、 r_2 为在 $[0,1]$ 间取值的相互独立的随机函数。最后归一化权值 $w_k^i = w_k^i / \sum_{i=1}^N w_k^i$ 。对后验概率密度 $p(x_k | z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$ 进行重采样, 产生新的粒子集 $\{x_0^i\}_{i=1}^N$, 使得 $p(x_k^i = x_k^i) = w_k^i$, 由于重采样是独立同分布的, 因此权值被重新设置为 $1/N$, 从而达到复制权值较大的粒子、剔除权值较小粒子的目的。输出状态估计 $\hat{x}_i = \sum_{i=1}^N w_k^i x_k^i$ 。

由于重采样是为了克服粒子退化现象而提出的, 因此是否重采样取决于粒子退化的程度。本文采用一种有效抽样尺度 $N_{\text{eff}} = 1 / \sum_{i=1}^N (w_k^i)^2$ 来度量退化程度, 设定有效样本数 $N_{\text{threshold}}$ 作为阈值, 当 $N_{\text{eff}} < N_{\text{threshold}}$ 时, 则进行重采样。这样就无需在每个时刻都进行重采样, 从而在一定程度上降低了算法的复杂度。

虽然 PSOPF 算法通过不断更新粒子的速度与位置, 驱动粒子向高似然区域移动, 在一定程度上能够抑制粒子的退化与枯竭问题。但由于 PSO 自身的缺点, 因此粒子群在追逐最优粒子时, 随着越来越接近最优粒子, 其速度越来越小, 粒子群表现出强烈的趋同性, 容易陷入局部最优, 特别当似然函数呈多峰状态时, 会使大量粒子聚集在次优位置, 无法到达最优位置, 并不能有效地改善粒子退化与枯竭现象, 影响算法的精确度。

本文采用适应度方差作为判断粒子陷入局部最优的准则。设粒子群的数量为 n , f_i 为第 i 个粒子的适应度, f_{av} 为粒子群的平均适应度, $f_s = \begin{cases} \max |f_i - f_{\text{av}}| & \text{if } f_i - f_{\text{av}} \neq 0 \\ 1 & \text{otherwise} \end{cases}$ 为适应度方差的标准值, 则粒子群体的适应度方差为 $\sigma^2 = \sum_{i=1}^n [f_i - f_{\text{av}}/f_s]^2$, σ^2 反映粒子群体的收敛程度, σ^2 越小, 群体越接近收敛。根据实际情况设定一收敛阈值 a , 当 $\sigma^2 < a$ 时, 则认为算法陷入局部最优。经多次实验, 本文 a 取 0.03。

3 基于混沌的 PSO 粒子滤波算法

混沌^[10]是一种貌似无规则的运动, 指在确定性的非线性系统中, 不附加任何因素就可以出现的类似随机的行为。混沌具有遍历性、随机性和规律性等特点, 能够在一定范围内按照自身的规律不重复地遍历所有的状态, 具有易跳出局部最优、搜索速度快、全局渐近收敛等优点。本文采用文献[11]中的分段 Logistic 混沌映射初始化各粒子:

$$x_{ij}(t+1) = \begin{cases} 4\mu x_{ij}(t)(0.5 - x_{ij}(t)) & 0 \leq x_{ij}(t) < 0.5 \\ 1 - 4\mu(1 - x_{ij}(t))(x_{ij}(t) - 0.5) & 0.5 \leq x_{ij}(t) \leq 1 \end{cases} \quad (3)$$

其中, $x_{ij}(t+1) \in (0,1)$, 本文初值 $x_{ij}(t)$ 取 0.345; μ 是控制参数, $\mu = 4$ 时, 系统完全处于混沌状态。与一般的 Logistic 或 Tent 混沌映射相比, 式(3)产生的混沌序列在 $(0,1)$ 间分布的对称性较好, 具有良好的随机性和对初值敏感性, 并且在生成混沌序列时无需进行扰动运算, 使算法具有更好的效率。

利用上式产生的混沌序列进行混沌搜索:

$$p'_{gj} = p_{gj} + R(2x_{ij}(t) - 1) \quad (4)$$

其中, R 为搜索半径, 本文采用 2 种不同搜索半径进行搜索, 一种是以 p_{gj} 为中心进行的混沌搜索: $R_1 = \eta p_{gj}$; 另一种是以原点为中心的混沌搜索: $R_2 = \rho(x_{\text{max}} - x_{\text{min}})$ 。其中, η 为混沌搜索系数, 取 1.5; x_{max} 和 x_{min} 是解空间的上、下界; ρ 是混沌搜索的收缩因子, 以减小混沌搜索的范围, 增加搜索精度, 本文取 $\rho = 1.1$ 。

基于混沌的 PSO 粒子滤波算法基本步骤如下:

Step1 取得量测值, 定义适应度函数:

$$z_k \sim \text{fitness} = \exp\left[-\frac{1}{2R_k}(z_k - \hat{z}_{k|k-1})^2\right] \quad (5)$$

其中, R_k 是量测噪声方差; z_k 为最新量测值; $\hat{z}_{k|k-1}$ 为预测量测值。

Step2 初始化: 在 $k = 0$ 时, 从重要性密度函数采样 N 个粒子, 用 $\{x_{0k}^i, w_{0k}^i\}_{i=1}^N$ 表示, 令每个样本的初始权值为 $\{w_k^i = 1/N, i = 1, 2, \dots, N\}$, 重要性密度函数取转移先验概率:

$$x_k^i \sim q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i)$$

Step3 权值更新: 根据最新量测值更新当前粒子权值:

$$\begin{aligned} w_k^i &= w_{k-1}^i p(z_k | x_{k-1}^i) = w_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)} = \\ &w_{k-1}^i p(z_k | x_k^i) = w_{k-1}^i \exp\left[-\frac{1}{2R_k}(z_k - \hat{z}_{k|k-1})^2\right] \end{aligned} \quad (6)$$

Step4 利用 PSO 算法根据下式来更新每个粒子的速度与位置, 使粒子不断地向真实状态靠近。

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + c_1r_{1j}[p_{ij}(t) - x_{ij}(t)] + c_2r_{2j}[p_{gj}(t) - x_{ij}(t)] \\ x_{ij}(t+1) &= x_{ij}(t) + v_{ij}(t+1) \end{aligned}$$

计算 $\sigma^2 = \sum_{i=1}^n [f_i - f_{\text{av}}/f_s]^2$, 设定阈值 a , 判断方差 σ^2 是否小于 a , 若方差 σ^2 不小于 a , 则转到 Step6, 若 $\sigma^2 < a$, 则向下进行。

Step5 用分段 Logistic 混沌映射初始化各粒子:

$$x_{ij}(t+1) = \begin{cases} 4\mu x_{ij}(t)(0.5 - x_{ij}(t)) & 0 \leq x_{ij}(t) < 0.5 \\ 1 - 4\mu(1 - x_{ij}(t))(x_{ij}(t) - 0.5) & 0.5 \leq x_{ij}(t) \leq 1 \end{cases}$$

利用上式产生的混沌序列进行混沌搜索:

$$p'_{gj} = p_{gj} + R(2x_{ij}(t) - 1)$$

找到全局最优位置 p'_{gj} , 再利用 PSO 算法更新粒子速度

与位置，摆脱次优位置，跳出局部最优，驱动粒子向全局最优位置靠近。

Step6 权值归一化: $w_k^i = w_k^i / \sum_{i=1}^N w_k^i$ 。

Step7 重采样: 当 $N_{eff} = 1 / \sum_{i=1}^N (w_k^i)^2 < N_{threshold}$ 时, 进行重采样。对 $\{x_{0,k}^i, w_k^i\}_{i=1}^N$ 原来的带权粒子重采样, 得等权粒子 $\{x_{0,k}^i, N^{-1}\}_{i=1}^N$ 。

Step8 状态估计: $\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i$; 方差估计: $p_k = \sum_{i=1}^N w_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T$ 。

Step9 判断 k 时刻是否为目标最后时刻, 若是, 则算法结束, 否则, 令 $k = k + 1$, 返回 Step2, 递推估计下一时刻目标状态的后验概率。

本文 CPSOPF 算法的流程如图 1 所示。

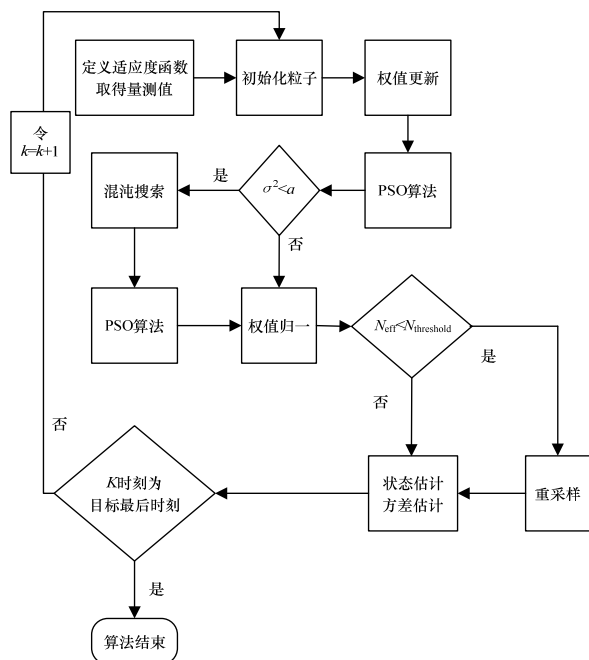


图1 CPSOPF 算法流程

4 仿真实验与结果分析

本文选取非静态增长模型(UNGM 模型), 应用 Matlab 软件进行仿真。系统过程模型与状态模型如下:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1+[x(t-1)]^2} + 8\cos[1.2(t-1)] + w(t) \quad (7)$$

$$z(t) = x(t)^2/20 + v(t) \quad (8)$$

该系统的特征是高度非线性的, 似然函数呈双峰状态, 这种双峰性使得用传统滤波方法很难解决问题。

为验证 CPSOPF 算法的性能, 本文选择在量测噪声方差 $R=1$ 、过程噪声方差分别为 $Q=10$ 和 $Q=20$ 2 种不同的噪声条件下进行仿真实验。对粒子数分别取 $N=100$ 和 $N=500$ 的 PF 算法、粒子数取 $N=100$ 的 PSOPF 算法和 CPSOPF 算法进行仿真比较。单次实验的均方根误差公式为: $RMSE = [\frac{1}{N} \sum_{i=1}^N (x_n - \hat{x}_n)^2]^{1/2}$ 。平均有效样本数为 \bar{N}_{eff} , 在同等情况下, 有效样本越多, 估计精度越高。时间步长为 50, 仿真 50 次, 算法终止条件设为迭代次数 $n=50$ 次。3 种算法单次实验的估计仿真图与残差值见图 2、图 3。

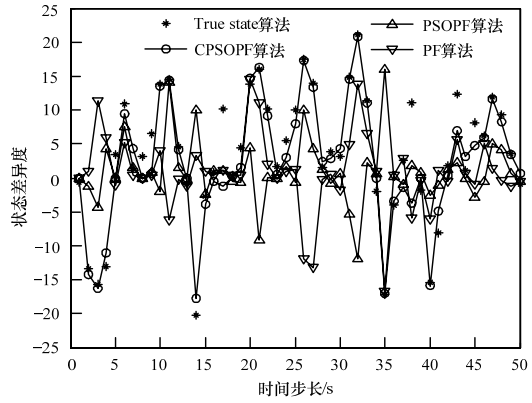


图2 PF、PSOPF、CPSOPF 状态估计仿真图

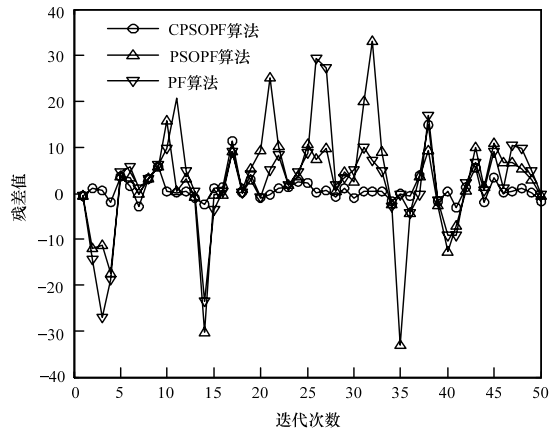


图3 PF、PSOPF、CPSOPF 残差值

统计 50 次实验数据的平均值, 结果见表 1。

表1 滤波性能的数据统计

条件	算法	粒子数 N	平均有效样本数	RMSE	运行时间/s
$R=10$ 、 $Q=1$	PF	100	12.791	2.703 1	0.230 7
	PF	500	18.084	2.274 8	0.509 8
	PSOPF	100	19.295	1.794 6	0.416 2
	CPSOPF	100	32.826	1.074 5	0.423 7
$R=20$ 、 $Q=1$	PF	100	13.289	4.349 6	0.241 8
	PF	500	19.104	4.143 5	0.523 9
	PSOPF	100	19.237	2.683 7	0.453 8
	CPSOPF	100	34.673	1.092 6	0.460 2

从图 2 中可看出, CPSOPF 算法相对于 PSOPF 与 PF 算法, 能够更精确地表达粒子真实状态。从图 3 可看出, 3 种算法中 CPSOPF 的波动最小, 说明其估计值与真实量测值最接近, 也就是说 CPSOPF 算法的估计精度最高。由表 1 的统计数据可知, 在相同噪声环境下, 相比于 PF 算法和 PSOPF 算法, 本文 CPSOPF 算法的有效样本数最多, 在抑制粒子退化和增加粒子多样性方面最有效。同时, CPSOPF 算法的 RMSE 值是最小的, 说明其估计精度是最高的。将粒子数增加到 500 的 PF 算法也不及 CPSOPF 算法的估计精度, 且比 CPSOPF 算法估计所用时间长, 说明在同等精度要求下, 与标准粒子滤波相比, CPSOPF 算法所用时间相对较短, 提高了算法的效率。

在噪声增加的情况下, PF 算法的 RMSE 值增加最多, 而 PSOPF 算法的 RMSE 也有改变, CPSOPF 算法的 RMSE 值改变最小, 说明 CPSOPF 的抗噪性能最好, 在噪声增加的情况下仍能很好地抑制粒子退化, 增加粒子多样性, 保持算法估计的高精度。(下转第 140 页)

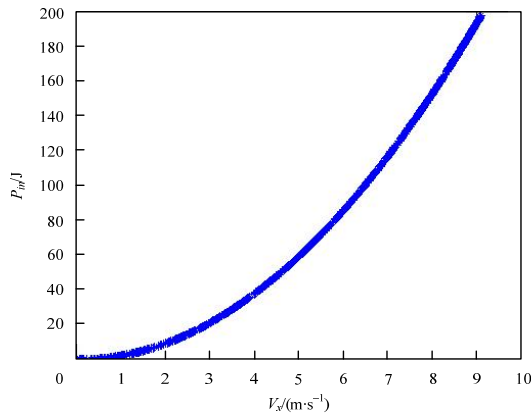


图8 优化后功率和速度关系

5 结束语

步态优化是提高机器人运动效率重要方法, 本文研究的蛇形机器人在平面上的蜿蜒运动步态是其基础步态, 主要针对多关节蛇形机器人欲在给定速度下消耗较小功率的蜿蜒步态运动的优化问题, 这为扩展到对蛇形机器人的多步态运动优化问题提供了研究基础。通过对蛇形机器人运动进行动力学模型分析, 在分析中考虑摩擦力情形, 对基本步态进行跟踪控制, 得到步态优化的目标函数, 采用多目标进化算法 NSGA-II 进行优化, Matlab 仿真给出了相应的优化结果, 仿真结果表明, 在给定运动速度下蛇形机器人消耗最小的功率, 则蜿蜒运动的步态参数依赖于摩擦力模型, 且在功率和速度之间寻找到了最优值, 算法能以较快的速度和效率达到变量优化目的, 表明 NSGA-II 算法对于解决蛇形机器人步态多目标优化问题是有效的。

参考文献

- [1] Saito M, Fukaya M, Iwasaki T. Serpentine Locomotion with Robotic Snakes[J]. IEEE Control Systems Magazine, 2002, 22(1):

(上接第 136 页)

5 结束语

本文针对粒子群优化粒子滤波易陷入局部最优问题, 引入混沌思想, 提出了一种基于混沌的 PSO 粒子滤波算法。当 PSO 滤波算法陷入局部最优时, 通过混沌搜索算法找出全局最优位置, 帮助算法及时跳出局部最优, 驱动粒子向全局最优位置靠近, 使用于估计的有效粒子增加, 进而抑制粒子退化与枯竭, 增加粒子多样性。仿真结果表明, 基于混沌的 PSO 粒子滤波算法提高了算法的精度, 并在一定程度上提高了算法的效率, 滤波性能明显优于传统的粒子滤波算法和 PSO 粒子滤波算法。

参考文献

- [1] Djuric P M, Kotecha J H, Zhang Jianqui, et al. Particle Filterings[J]. IEEE Signal Processing Magazine, 2003, 20(10): 19-38.
- [2] Valyrakis A, Tsakonas E E, Sidiropoulos N D, et al. Stochastic Modeling and Particle Filtering Algorithms for Tracking a Frequency-hopped Signal[J]. IEEE Trans. on Signal Processing, 2009, 57(8): 3108-3118.
- [3] Sankaranarayanan A C, Srivastava A, Chellappa R. Algorithmic and Architectural Optimizations for Computationally Efficient Particle Filter[J]. IEEE Trans. on Image Processing, 2008, 17(5):

64-81.

- [2] 林铨云, 董家礼. 多目标优化的方法与理论[M]. 长春: 吉林教育出版社, 1992.
- [3] Mehta V, Brennan S, Gandhi F. Rigid-link Snake Robot Dynamics and Optimally Efficient Gait Confirmed by Experimentation[J]. IEEE Transactions on Automatic Control, 1985, 30(9): 881-889.
- [4] 马小妹, 李宇龙, 严浪. 传统多目标优化方法和多目标遗传算法的比较综述[J]. 电气传动自动化, 2010, 32(3): 48-50.
- [5] Kalyanmoy D, Agarwal S, Meyarivan T. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-190.
- [6] 王洪涛, 刘玉田. 基于 NSGA-II 的多目标输电网架最优重构[J]. 电力系统自动化, 2009, 33(23): 14-18.
- [7] Hirose S. Biologically Inspired Robots: Snake-like Locomotors and Manipulators[M]. [S. l.]: Oxford University Press, 1993.
- [8] Mehta X, Brennan S, Gandhi F. Experimentally Verified Optimal Serpentine Gait and Hyperredundancy of a Rigid-link Snake Robot[J]. IEEE Transactions on Robotics, 2008, 24(2): 198-209.
- [9] Saito M, Fukaya M, Iwasaki T. Serpentine Locomotion with Robotic Snakes[J]. IEEE Control Systems Magazine, 2002, 22(1): 64-81.
- [10] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II[J]. IEEE Trans. on Evolutionary Computation, 2002, 6(2): 187-192.
- [11] Brown D E, Huntley C L. A Practical Application of Simulated Annealing[J]. Pattern Recognition, 1992, 25(4): 401-402.
- [12] 王艳, 曾建潮. 解决多目标优化问题的拟态物理学优化算法[J]. 计算机工程, 2010, 36(20): 188-190.
- [13] 雷德明, 严新平. 多目标智能优化算法及其应用[M]. 北京: 科学出版社, 2009: 47-48.

编辑 索书志

737-748.

- [4] Ruan Yanhua, Willett P, Marrs A, et al. Practical Fusion of Quantized Measurements via Particle Filter[J]. IEEE Trans. on Aerospace and Electronic Systems, 2008, 44(1): 15-29.
- [5] 刘一鸣, 周尚波. 基于多特征融合的粒子滤波视频跟踪算法[J]. 计算机工程, 2010, 36(22): 228-230.
- [6] Hu Haidong, Huang Xianlin, Li Mingming. Federated Unscented Particle Filtering Algorithm for SINS/CNS/GPS System[J]. Journal of Central South University of Technology, 2010, 17(4): 778-785.
- [7] 李翠芸, 姬红兵. 新遗传粒子滤波的红外弱小目标跟踪与检测[J]. 西安电子科技大学学报, 2009, 36(4): 619-644.
- [8] 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.
- [9] de Oca M A, Stutzle T, Birattari M, et al. Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm[J]. IEEE Trans. on Evolutionary Computation, 2009, 13(5): 1120-1132.
- [10] 谭跃, 谭冠政, 涂立. 一种新的混沌差分进化算法[J]. 计算机工程, 2009, 35(11): 216-218.
- [11] 范九伦, 张雪锋. 分段 Logistic 混沌映射及其性能分析[J]. 电子学报, 2009, 37(4): 720-725.

编辑 张正兴