

文章编号: 1001-0920(2013)02-0193-08

基于自适应粒子群优化的新型粒子滤波在目标跟踪中的应用

陈志敏, 薄煜明, 吴盘龙, 段文勇, 刘正凡

(南京理工大学 自动化学院, 南京 210094)

摘要: 针对基于粒子群优化的粒子滤波(PSO-PF)算法精度不高, 实时性差, 难以满足雷达机动目标跟踪的需求, 提出一种基于动态邻域自适应粒子群优化的粒子滤波(DPSO-PF)算法. 该算法可以动态调整粒子邻域环境, 其中每个粒子按照邻域的环境和自身的位置信息自适应地调整相互间的邻域粒子数量, 使邻域粒子数量更为合理, 达到寻优能力与收敛速度的最佳平衡. 最后利用不同模型对该算法进行了仿真实验, 实验结果表明所提出的算法能够提高雷达机动目标跟踪的实时性和精确性.

关键词: 粒子滤波; 粒子群优化; 自适应; 目标跟踪; 局部最优

中图分类号: TP391

文献标志码: A

Novel particle filter algorithm based on adaptive particle swarm optimization and its application to radar target tracking

CHEN Zhi-min, BO Yu-ming, WU Pan-long, DUAN Wen-yong, LIU Zheng-fan

(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China. Correspondent: CHEN Zhi-min, E-mail: chenzhimin@188.com)

Abstract: A particle filter algorithm based on dynamic neighborhood adaptive particle swarm optimization(DPSO-PF) is presented in order to solve the problem of the low precision and complicated calculation of particle filter based on particle swarm optimization(PSO-PF) algorithm. This algorithm can dynamically adjust the particle neighborhood environment, where each particle can adjust the number of particles in the neighborhood based on self-adaptation basis according to the neighborhood environment and their own position information, accordingly a best balance is achieved between optimal seeking and convergence rate. Finally, different models are used for simulation experiment and the results show that the proposed algorithm improves the real-time performance and the precision of maneuvering target tracking by using radar.

Key words: particle filter; particle swarm optimization; adaptive; target tracking; local optimization

0 引言

利用雷达进行机动目标跟踪时, 其量测易受到闪烁噪声的干扰. 闪烁噪声具有明显的非高斯特性, 它是由相位的随机变化与目标不同部位的散射强度的不同所引起的^[1]. 常规的基于 Kalman 理论的滤波方法只适用于高斯噪声下的线性系统, 它在处理雷达目标跟踪等非高斯噪声下的非线性系统时效果较差. 而粒子滤波(PF)^[2]由于没有对其状态函数和观测函数做非线性假设, 也未要求其在非高斯噪声的环境下, 更适用于雷达目标跟踪^[3]. 但是, 粒子滤波存在权值退化的问题^[4], 若采用重采样的方法容易导致粒子匮乏, 从而影响滤波的精度.

粒子群优化(PSO)是一种模拟鸟类集群运动的智能优化方法, 其概念简单、易于实现, 而且能够有效地解决复杂优化问题, 因而得到了广泛的应用. 然而, PSO 易出现局部最优、粒子早熟等问题, 许多学者对其进行了改进. 文献[5]提出一种基于小世界网络的动态调整邻域的 PSO 算法, 该算法利用聚类系数和平均最短距离来动态调整粒子的邻居, 缓解了粒子早熟现象, 但对于单峰值等类型的问题, 其性能较差, 并且收敛速度较慢. 文献[6]提出了具有邻域探测机制的改进型 PSO 算法, 该算法在进化过程中选择粒子最佳位置, 按照半径总体递减的方式进行邻域探测, 并引入了速度变异算子, 提高了种群的多样性, 增强了全

收稿日期: 2011-10-03; 修回日期: 2011-12-15.

基金项目: 国防重点预研项目(40405020201); 高等学校博士学科点专项科研基金课题(200802881017); 南京理工大学自主科研专项计划自主项目(2010ZYTS051); 南京理工大学紫金之星基金项目(AB41381).

作者简介: 陈志敏(1986—), 男, 博士生, 从事粒子滤波、目标跟踪、系统控制等研究; 薄煜明(1965—), 男, 研究员, 博士生导师, 从事目标跟踪、导航制导、系统控制等研究.

局搜索能力,但计算复杂度较大.文献[7]提出了基于动态邻居调整策略的PSO方法,用以解决多目标优化问题,该方法通过计算粒子间的距离和粒子的适应度函数值生成每个粒子各自不同的邻域,提高了优化效率,但该方法是基于非约束条件下的双目标优化函数进行实验的,对于约束条件下的多目标(大于2)优化函数具有不确定性.

基于粒子群优化方法的粒子滤波^[8-10](PSO-PF)是智能优化粒子滤波的代表之一.PSO-PF将最优量测值引入采样分布,并对采样过程进行优化,不断实时更新粒子的速度,使采样分布向后验概率密度较高的区域运动.PSO-PF改善了粒子滤波的权值退化现象,在一定程度上提高了精度,但是由于该算法中粒子的邻域具有固定性,不能充分利用附近粒子的信息,并且在寻优过程中过多的迭代次数导致运算复杂度过大,难以满足雷达目标跟踪的精确性和实时性的要求.文献[11]提出了适用于乘性噪声下的粒子群优化粒子滤波方法,该方法利用常规的多目标优化方法使粒子靠近似然区域和先验分布,缓解了粒子贫乏的现象.文献[12]提出了对大权重粒子多次重复采样的粒子群优化粒子滤波方法,使得粒子更加向真实区域靠近,从而提高了粒子滤波的精度,但同时也增加了运算量,难以在对实时性要求较高的目标跟踪领域应用.

本文提出一种动态调整邻域粒子的新型粒子群优化粒子滤波算法(DPSO-PF).该算法采用邻域扩展因子和邻域限制因子,通过粒子群多样性的变化来自适应调整粒子邻域粒子的数量,而每个粒子则按照自身位置和邻域的环境参数调整相互间的邻域关系,使得所提出的算法既具有较好的局部搜索能力,又能够保持样本的多样性,从而提高目标跟踪的速度和精度.

1 粒子滤波基本原理

粒子滤波是贝叶斯估计基于抽样理论的一种近似算法^[13-14],基本思想是寻找一组在状态空间中的随机样本对条件后验概率密度函数进行近似,用样本均值代替积分运算,从而获得对状态的最小方差估计.假设状态方程和观测方程如下:

$$x_k = f(x_{k-1}, v_{k-1}), \quad (1)$$

$$y_k = h(x_k, n_k). \quad (2)$$

设状态初始概率密度为 $p(x_0|y_0) = p(x_0)$, 预测方程为

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}, \quad (3)$$

状态的更新方程为

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k})}, \quad (4)$$

其中

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k. \quad (5)$$

将重要性函数 $q(x_{0:k}|y_{1:k})$ 改写成

$$q(x_{0:k}|y_{1:k}) = q(x_0) \prod_{j=1}^k q(x_j|x_{0:j-1}, y_{1:j}), \quad (6)$$

则权值公式为

$$w_k = \frac{p(y_{1:k}|x_{0:k})p(x_{0:k})}{q(x_k|x_{0:k-1}, y_{1:k})q(x_{0:k-1}, y_{1:k})} = w_{k-1} \frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}, y_{1:k})}. \quad (7)$$

从 $p(x_{k-1}|y_{1:k-1})$ 中采样 N 个样本点 $\{x_{k-1}^i\}_{i=1}^N$, 得到概率密度为

$$p(x_{k-1}|y_{1:k-1}) = \sum_{i=1}^N w_{k-1}^i \delta(x_{k-1} - x_{k-1}^i), \quad (8)$$

其中 δ 为狄克拉函数.更新重要性权值

$$w_k^i = w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)}. \quad (9)$$

最后进行权重归一化,并输出估计状态

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

$$\tilde{x} = \sum_{i=1}^N w_k^i x_k^i. \quad (11)$$

2 PSO-PF 算法

2.1 PSO 算法基本原理

PSO算法可以表述为:随机初始化一个粒子群,该粒子群的粒子数量为 m , 维数为 n , 其中第 i 个粒子的位置为 $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, 速度为 $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ ^[15]. 在迭代过程中,粒子通过个体极值 $P_i = (P_{i1}, P_{i2}, \dots, P_{in})$ 和全局极值 $G = (g_1, g_2, \dots, g_n)$ 不断更新自身的速度和位置,从而实现粒子的寻优^[16].更新公式如下:

$$V_i = w \times V_i + c_1 \times \text{Rand}() \times (P_i - X_i) + c_2 \times \text{Rand}() \times (G - X_i), \quad (12)$$

$$X_{i+1} = X_i + V_i. \quad (13)$$

其中: Rand 为介于 $(0, 1)$ 区间内的随机数; 惯性系数为 w , w 越大全局搜索能力越强, w 越小局部搜索能力越强; c_1 和 c_2 为学习因子.

2.2 标准 PSO-PF 的主要思想

常规的粒子滤波的重要性采样不是最优的, 但将 PSO 方法融入 PF, 则可以优化采样的过程, 使得粒子群在权重值更新前更加趋向于高似然的区域, 从而缓解粒子匮乏的现象, 可以减少滤波所需的粒子数量. PSO 与 PF 相融合, 主要是利用粒子所经历过的最优状态值 P_{pbest} 和粒子群中的最优粒子的状态值 P_{gbest} , 其中 P_{pbest} 为当前粒子所经历的状态值中适应度函数值最大的粒子的状态值, P_{gbest} 为粒子群中适应度函

数值最大的粒子的状态值. 通过如下公式:

$$V_k^i = |\text{Rand } n| \times (P_{p_{\text{best}}} - X_{k-1}^i) + |\text{rand } n| \times (P_{g_{\text{best}}} - X_{k-1}^i), \quad (14)$$

$$X_k^i = X_{k-1}^i + V_{k-1}^i, \quad (15)$$

对每个粒子的位置和速度进行实时更新, 使粒子不断地向真实状态逼近, 其中 $|\text{Rand } n|$ 和 $|\text{rand } n|$ 为正的高斯分布随机数.

3 DPSO-PF 算法

标准 PSO-PF 在优化过程中, 粒子的邻域粒子是固定的, 不能按照每个粒子的自身状态和邻域环境参数调整邻域的粒子状态, 无法控制每个粒子对邻域粒子的影响力, 容易导致局部最优, 从而影响滤波的速度和精度. 本文引入多样性因子的概念, 并提出了邻域扩展策略和邻域限制策略, 使得 DPSO-PF 可以根据粒子自身和邻域的状态参数, 自适应动态控制邻域粒子的数量, 使得粒子滤波可以达到收敛速度与寻优能力的最佳平衡, 提高滤波器的性能.

3.1 多样性因子概念

在粒子滤波中, 粒子的多样性是滤波器性能的一个重要指标, 多样性程度越高, 说明粒子群的质量越高, 滤波器的性能越好. 然而, 在 PSO-PF 中, 粒子的多样性程度还影响寻优速度和局部搜索能力, 虽然多样性程度的提高会增强算法的全局寻优能力, 但局部搜索能力和寻优速度会有一定程度的下降, 如果盲目提高粒子多样性的程度, 则会降低 PSO-PF 的性能. 对此, 本文引入了多样性因子, 通过考察多样性因子的大小来控制邻域粒子数量的减少或增加, 使每个粒子都能达到最佳的邻域粒子数量.

设粒子的数量为 M , 粒子经历过的最佳状态的适应度函数值的集合为 $Y_k^i = \{Y_k^1, Y_k^2, \dots, Y_k^M\}$, 其中适应度函数为 $Y = \exp\left[-\frac{1}{2R_k}(z_{\text{New}} - z_{\text{Pred}})\right]$. 适应度函数是通过实际值与预测值之间的差值的大小衡量粒子的可信度的一个标准, 若适应度高, 则粒子的可信度高; 若适应度低, 则粒子的可信度低. 取 $Y_{\min} = \min\{Y_k^1, Y_k^2, \dots, Y_k^M\}$, $Y_{\max} = \max\{Y_k^1, Y_k^2, \dots, Y_k^M\}$, 令 $[Y_{\min}, Y_{\max}]$ 为观察区间, 将观察区间 $[Y_{\min}, Y_{\max}]$ 均分为 M 个宽度相同的小区域, 统计 Y_k^i 落入每个小

$$p_i = \frac{e_i}{M}, \quad (16)$$

$$D_t = -\sum_{i=1}^M p_i \log(p_i), \quad (17)$$

则将 D_t 称为多样性因子^[16].

D_t 是一个可以反映 DPSO-PF 的粒子多样性程

度高低的一个参数, 如果在每个小区域内都有粒子落入, 即 $e_i = 1$, 则此时 D_t 最大, 粒子群的多样性程度最高; 反之, 当绝大多数粒子都落入某一个区间内时, D_t 很小, 多样性程度也很低.

在本文算法中, 多样性因子 D_t 用来控制每个粒子邻域的粒子数量, 控制方法如下: 若本轮多样性因子的值小于上轮循环的多样性因子的值, 则说明多样性有下降的趋势, 应减少邻域的粒子数量, 以避免算法陷入局部最优; 若本轮多样性因子的值大于上轮循环的多样性因子的值, 则说明此时多样性有上升趋势, 此时应增加邻域的粒子数量来提高收敛速度.

3.2 邻域扩展

3.2.1 扩展因子概念

DPSO-PF 进行邻域粒子扩展时, 既要让适应度较高的粒子具有更高的扩展概率, 又要避免过快扩展而导致粒子的早熟现象, 因此要求对邻域粒子的扩展应适当. 对此, 本文给出如下定义.

定义 1 将粒子群中所有粒子的适应度值按从小到大排列, 得到数组 h . 令第 i 个粒子的适应度值在数组 h 中的序号为 mark_i , 第 i 个粒子的邻域粒子数量为 C_i , 第 i 个粒子的邻域粒子集合为 $\text{neighbor}(i)$, 可得到

$$\text{Extend}(i) = \frac{\text{mar } k_i C_i}{\frac{1}{C_i} \sum_{j \in \text{neighbor}(i)} (\text{mar } k_j C_j)}, \quad (18)$$

这里将 $\text{Extend}(i)$ 称为粒子 i 的扩展因子.

式(18)中 $\frac{1}{C_i} \sum_{j \in \text{neighbor}(i)} (\text{mar } k_j C_j)$ 表示第 i 个粒子的邻域粒子的扩展状态. 若 $\text{mar } k_j C_j$ 的值较大, 则表明粒子邻域粒子的数量已经过多或者适应度较小. 若 $\text{Extend}(i) < 1$, 则算法给予第 i 个粒子较高的扩展概率; 若 $\text{Extend}(i) \geq 1$, 则不对第 i 个粒子的邻域进行扩展. $\text{Extend}(i)$ 的值由粒子 i 的适应度值和邻域粒子的状态共同决定.

3.2.2 邻域粒子扩展策略

1) 利用式(18)计算第 i 个粒子的扩展因子 $\text{Extend}(i)$.

2) 取介于 $[0, 1]$ 之间的随机数 r , 如果

$$r > \text{Extend}(i),$$

则从粒子 i 的非邻域粒子群中随机选择一个粒子添加为新的邻域粒子, 并对 $P_{g_{\text{best}}}$ 进行更新; 若

$$r \leq \text{Extend}(i),$$

则不增加该粒子的邻域粒子数量.

显然, 对于 $\text{Extend}(i) < 1$ 的粒子, $\text{Extend}(i)$ 的值越小, 粒子 i 的邻域被扩展的概率越大, 而剩余的粒

子的邻域不予扩展. 该策略使得邻域粒子的扩展更为合理.

3.3 邻域限制

3.3.1 限制因子概念

在 PSO 和 PSO-PF 中, 局部最优现象是一个典型问题. 造成局部极值现象的原因主要是适应度值较高的粒子通过邻域对其他的粒子造成较大的影响, 从而引起了局部最优现象, 局部最优现象会降低粒子的多样性程度. 若要降低适应度值较高的粒子对邻域粒子的影响, 可以将粒子的邻域中适应度很高的粒子删除, 但删除粒子的数量要合理. 对此, 本文给出如下定义.

定义 2 令 S_i 表示第 i 个粒子的邻域粒子的数量, 将所有粒子的适应度值从小到大排列, 得到数组 g . 令粒子 i 的适应度值在数组 g 中的序号为 $l\text{mark}_i$, 如果 $l\text{mark}_i = S_i + 1$, 则说明粒子 i 的适应度值比其邻域所有的粒子都好; 如果 $l\text{mark}_i = 1$, 则说明粒子 i 的适应度值在粒子群中最低. 令

$$\text{removal}(i) = \frac{l\text{mark}_i - 1}{S_i}, \quad (19)$$

这里将 $\text{removal}(i)$ 称为粒子 i 的限制因子.

限制因子表明在粒子 i 的邻域粒子中适应度值低于粒子 i 的粒子所占的比例. 随着算法的不断更新迭代, $\text{removal}(i)$ 将随着邻域环境状态的改变而改变.

3.3.2 邻域粒子限制策略

当 $\text{Extend}(i) \geq 1$ 时, 表明粒子 i 的邻域不宜被扩展. 邻域粒子限制策略将对这些粒子的邻域中 $\text{removal}(i)$ 最大的粒子进行限制, 以降低适应度值较高的粒子的影响力. 为了确保 DPSO-PF 算法的顺利进行, 至少应保证粒子在邻域自适应动态调整过程中存在 2 个邻域粒子.

Step 1: 计算粒子 i 的 $\text{Extend}(i)$, 若 $\text{Extend}(i) \geq 1$, 则执行 **Step 2**, 否则不对粒子的邻域进行限制.

Step 2: 计算粒子 i 的邻域粒子的限制因子 $\text{removal}(q)$, 选择限制因子值最大的粒子 q .

Step 3: 如果 $S_i > 2$ 且 $S_q > 2$, 则将粒子 q 从粒子 i 的邻域中删除, 同时更新限制因子的值, 否则不对邻域作出改变.

3.4 DPSO-PF 算法计算步骤

本文 DPSO-PF 算法的计算过程如下.

Step 1: 在 $k = 0$ 时刻进行重要性采样, 将得到的 N 个粒子 $\{x_0^i, i = 1, 2, \dots, N\}$ 作为初始样本. 重要性密度函数为

$$x_k^i \sim q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i). \quad (20)$$

设置适应度函数表达式为

$$Y = \exp \left[-\frac{1}{2R_k} (z_{\text{New}} - z_{\text{Pred}}) \right]. \quad (21)$$

其中: z_{New} 为最新的观测值, z_{Pred} 为预测值.

Step 2: 计算每个粒子的重要性权值

$$w_k^i = w_{k-1}^i p(z_k | x_k^i), \quad (22)$$

其中 $p(z_k | x_k^i)$ 通过将 x_k^i 代入观测方程得到.

Step 3: 利用下式计算粒子的 pb 和 pg .

$$pb_k^i = \begin{cases} pb_k^i, & Y(x_g) < Y(pb_k^i), \\ x_g, & Y(x_g) > Y(pb_k^i); \end{cases} \quad (23)$$

$$pg_k \in \{x_k^1, x_k^2, \dots, x_k^N | Y(x)\} = \max\{Y(x_k^1), Y(x_k^2), \dots, Y(x_k^N)\}. \quad (24)$$

其中初始值 $pb_0^i = x_0^i$.

Step 4: 通过下式计算得到经历 m 次迭代更新后粒子 x_k^{im} 的速度 v_k^{im} :

$$V_k^{im+1} = |\text{Rand } n| \times (pb_{k-1}^i - x_{k-1}^{im}) + |\text{rand } n| \times (pg_k - x_{k-1}^{im}); \quad (25)$$

利用下式使粒子 x_k^{im} 在速度 v_k^{im+1} 的作用下运动至下一次迭代更新的位置 x_k^{im+1} :

$$x_{k+1}^{im+1} = x_k^{im} + v_k^{im+1}. \quad (26)$$

Step 5: 利用式 (21) 和具体模型的观测方程计算粒子的适应度, 并通过式 (23) 和 (24) 继续更新粒子的 pb 和 pg , 并将此时的 pb 的适应度 $Y(pb)$ 记为 G_t .

Step 6: 利用式 (17) 计算粒子的多样式因子 D_t .

Step 7: 对粒子群中粒子的适应度按从小到大排列, 得到数组 $\text{mar } k$.

Step 8: 由最优粒子值 G_t 和多样性因子 D_t 共同决定对粒子的邻域是进行扩展还是限制. 如果 $D_t < D_{t-1}$ 且 $G_t = G_{t-1}$, 则执行 **Step 10**, 否则执行 **Step 9**.

Step 9: 按照邻域扩展策略, 增加粒子的邻域粒子数量, 即在粒子 i 的非邻域粒子中随机选择一个粒子作为新的邻域粒子并对 pb 进行更新, 然后转 **Step 12**.

Step 10: 对粒子 i 和其邻域的所有粒子的适应度按从小到大排列, 得到数组 $l\text{mar } k_i$.

Step 11: 对于每个粒子, 按邻域限制策略更新邻域的结构, 即计算它们的邻域粒子的 $\text{removal}(q)$, 挑选出 $\text{removal}(q)$ 最大的粒子 q , 将粒子 q 从粒子 i 的邻域中删除.

Step 12: 当粒子的最优值符合所设定的阈值 ε 或者到达最大迭代次数时, 停止优化迭代, 否则令 $E_t = E_{t-1}$, $G_t = G_{t-1}$, 并转入 **Step 4**.

Step 13: 计算优化后粒子的权值, 并对其归一化, 即

$$w_k^i = w_k^i / \sum_{i=1}^N w_k^i. \quad (27)$$

Step 14: 状态输出

$$\tilde{x} = \sum_{i=1}^N w_k^i x_k^i. \quad (28)$$

4 仿真实验及性能分析

4.1 算法基本性能仿真测试

为了验证本文算法的精度和运算时间等基本性能, 选取单变量非静态增长模型(UNGM模型), 实验环境为DELL-vostro 1400, 内存为DDR2-2 GB, CPU为Intel-T7250, 过程模型以及量测模型如下:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1 + [x(t-1)]^2} + 8 \cos[1.2(t-1)] + w(t), \quad (29)$$

$$z(t) = \frac{x(t)^2}{20} + v(t), \quad (30)$$

其中 $w(t)$ 和 $v(t)$ 为零均值的高斯噪声. 该模型为似然函数呈双峰状的非线性系统^[18].

基于粒子群优化方法的粒子滤波的粒子更新过程是一个迭代寻优的过程. 一般迭代终止条件有两种情况: 1) 当粒子的最优值满足所设定的需求时; 2) 迭代次数达到所设定的最大迭代次数时. 高性能的粒子群优化粒子滤波以第1)种情况居多, 因而迭代次数少, 实时性好; 而性能较差的粒子群优化粒子滤波算法出现第2)种情况的较多, 实时性较差. 因此, 为了验证本文改进算法的性能, 本文提出优化成功率 η 的概念, 优化成功率的定义如下:

设定计数标识变量temp, 令采样时间为 r_{STEP} , 如果 k 时刻结束时的迭代次数达到算法所设定的最大迭代次数, 则 $temp = temp + 1$; 未达到最大迭代次数, 则视为该时刻优化成功, 其优化成功率为

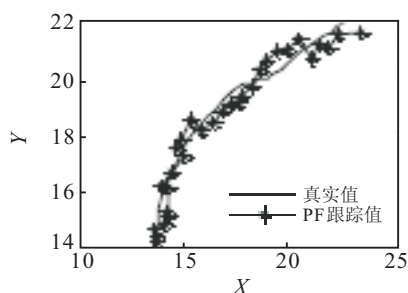
$$\eta = \frac{r_{STEP} - temp}{r_{STEP}}.$$

本文的有效样本定义为归一化后权重大于 $1/N$ (N 为粒子数)的粒子. 用PF、PSO-PF和DPSO-PF对该非线性系统进行状态估计和跟踪, 均方根误差公式为

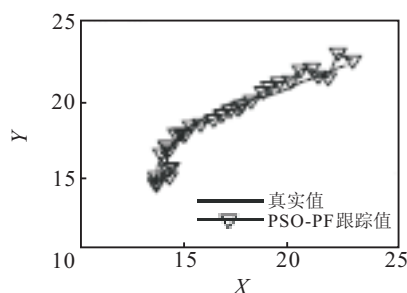
$$RMSE = \left[\frac{1}{T} \sum_{t=1}^T (x_t - \hat{x}_t)^2 \right]^{1/2}.$$

当取粒子数 $N = 100$, 系统噪声方差 $Q = 10$, 量测噪声方差 $R = 1$ 时, 仿真结果如图1所示.

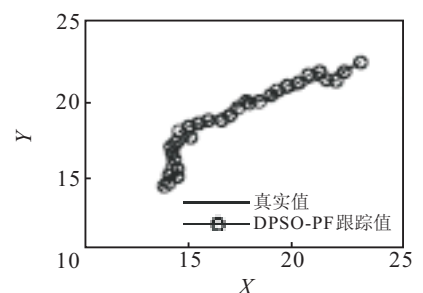
增加系统噪声强度后, 进行500次蒙特卡洛仿真并取平均值, 结果如表1所示.



(a) PF跟踪效果曲线



(b) PSO-PF跟踪效果曲线



(c) DPSO-PF跟踪效果曲线

图2 3种算法跟踪效果图

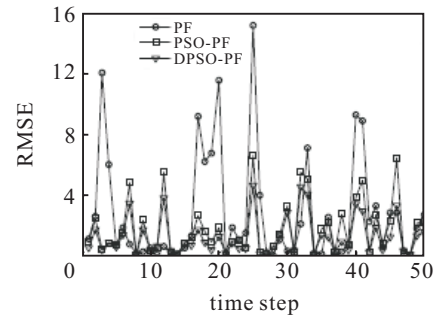


图1 不同算法的均方根误差曲线

表1 UNGM模型仿真结果数据比较

参数	算法	有效样本	$\eta/\%$	RMSE	运算时间/s
$Q = 10$	PF	15.426	—	2.701 2	0.346 2
	PSO-PF	19.104	52.74	1.668 2	0.526 3
	DPSO-PF	22.331	96.83	1.223 6	0.466 3
$Q = 20$	PF	15.237	—	5.592 7	0.341 5
	PSO-PF	19.062	48.22	2.828 7	0.535 7
	DPSO-PF	22.563	92.41	2.146 3	0.477 6

从实验结果可以看出, 融入了粒子群优化的粒子滤波算法的误差明显小于标准粒子滤波算法, 这是由于PSO是一个粒子寻优的过程, 提高了粒子的质量. 所提出的算法虽然在运算步骤上比PSO-PF多了Step 6~Step 11, 但是由于DPSO-PF的优化成功率高于PSO-PF, 使得算法在未达到迭代次数上限的情况下便达到优化需求, 从而减少了寻优迭代的次数, 提高了实时性. 同时, 由于本文算法利用邻域扩展策略和限制策略对粒子的邻域粒子的数量进行了合理化, 提高了粒子群的质量, 使本文算法的精确性和有效样本数较PSO-PF有了较大程度的提高.

4.2 目标跟踪有效性仿真测试

纯角度跟踪模型(BOT)^[19]是一类重要的基本目标跟踪模型. 该模型假定目标在二维空间中运动, 探测传感器智能地观测目标每个时刻的方位角. 该模型为非线性系统, 目标的 V_x 和 V_y 只与前一时刻的自身状态相关, 而 X 和 Y 则不仅与自身状态相关, 还与 V_x 和 V_y 相关, 跟踪预测的复杂度高于UNGM模型, 更有利于验证本文算法对目标跟踪的有效性. 运动方程和观测方程如下:

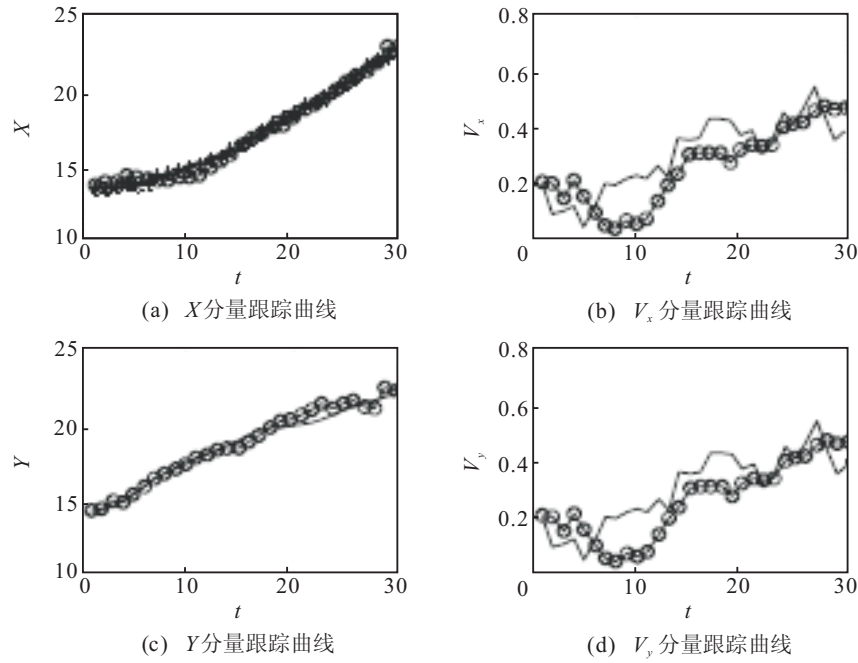


图 3 DPSO-PF 算法各状态变量跟踪曲线

$$X(t) = \phi X(t-1) + w(t), \quad (31)$$

$$z(t) = \tan^{-1}(Y/X) + v(t). \quad (32)$$

其中: $X_t = [X, V_x, Y, V_y]_t$, ϕ 为系统状态转移矩阵. 初始状态为 $X_0 = [13.5, 0.2, 14, 0.15]$, $w(t)$ 和 $v(t)$ 分别服从 $N(0, 0.001)$ 和 $N(0, 0.005)$, 粒子的数量为 100, 采样步数设为 30, 仿真结果如图 2 和图 3 所示.

4.3 协同连续转弯模型

4.3.1 机动目标连续转弯模型建立

在该模型中, 机动目标进行的是连续转弯的非线性运动, 其状态方程和测量方程如下:

$$X(t) = AX(t-1) + BW(t-1), \quad (33)$$

$$Z(n) = HX(t) + V(t). \quad (34)$$

其中

$$A = \begin{bmatrix} 1 & \frac{\sin(wT)}{w} & 0 & \frac{\cos(wT) - 1}{w} & 0 \\ 0 & \cos(wT) & 0 & -\sin(wT) & 0 \\ 0 & \frac{1 - \cos(wT)}{w} & 1 & \frac{\sin(wT)}{w} & 0 \\ 0 & \sin(wT) & 0 & \cos(wT) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (35)$$

$$B = \begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (36)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (37)$$

A 为系统状态矩阵; B 为系统状态噪声矩阵; T 为采样时间; W 为模型误差和外界扰动引起的系统噪声;

$X(t) = [x(t), v_x(t), y(t), v_y(t), w]$ 为目标的状态向量, x 和 y 为目标的位移分量, v_x 和 v_y 分别为 x 和 y 方向的速度分量; w 为角速度, 单位为 rad/s. 当 $w > 0$ 时, 机动目标向左转弯; 当 $w < 0$ 时, 机动目标向右转弯; 当 $w = 0$ 时, 目标进行匀速直线运动.

4.3.2 观测噪声模型建立

闪烁噪声不仅具有非高斯特征, 而且具有明显的“厚尾”分布特性. “厚尾”特性与目标位置闪烁的跳跃性有关^[20]. 雷达跟踪的闪烁噪声可以表示为高斯噪声和具有“厚尾”特征分布的噪声的加权之和. 常用的具有“厚尾”特性的分布有 t 分布、拉普拉斯分布和高斯分布等.

本文采用高斯分布和拉普拉斯分布合成雷达闪烁噪声, 闪烁噪声的概率密度函数如下:

$$p(w) = \lambda[(1 - \varepsilon)p_g(x) + \varepsilon p_t(x)]. \quad (38)$$

其中: λ 为调整闪烁噪声幅度的参数, $\varepsilon \in [0, 1]$ 为闪烁噪声发生的概率, $p_g(x)$ 为如下高斯密度函数:

$$p_g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right). \quad (39)$$

本文通过上述方法合成了一个在实际雷达目标跟踪中典型的闪烁噪声, 具体参数设置如下: 高斯噪声均值为 0, 分配系数 $\varepsilon = 0.05$, $\lambda = 5$, 高斯噪声方差 $\sigma = 0.001$, 拉普拉斯分布参数 $\eta = 0.05$.

4.3.3 机动目标连续转弯模型的实验仿真

在机动目标协同连续转弯模型中, 跟踪误差随着转弯次数的增多而增加, 因此为了更好地对 DPSO-PF 的跟踪性能进行测试, 假设某机动目标进行 4 次转弯, 该机动目标初始位置为 $(1.45 \times 10^5, 1.2 \times 10^5)$ m, 初始

速度为 $(-100, -100)$ m/s. 设机动目标在 $0 \sim 25$ s 进行匀速直线运动, 在 $26 \sim 45$ s 以 $w = 4.77^\circ/\text{s}$ 的角速度进行匀角速度转弯, 在 $46 \sim 65$ s 进行匀速直线运动, 在 $66 \sim 85$ s 以 $w = 6.56^\circ/\text{s}$ 的角速度进行匀角速度转弯, 在 $86 \sim 100$ s 作匀速直线运动, 在 $101 \sim 125$ s 作角速度 $w = -5.96^\circ/\text{s}$ 的匀角速度转弯运动, 在 $126 \sim 150$ s 作匀速直线运动, 在 $151 \sim 175$ s 作角速度为 $w = 5.38^\circ/\text{s}$ 的匀角速度转弯运动. 令采样时间间隔为 1 s, 热噪声观测的距离标准差为 10 m, 方位角观测的标准差为 0.1° , 闪烁效应的观测距离的标准差为 100 m, 方位角标准差为 0.2° . 分别用 PF、PSO-PF 和 DPSO-PF 进行目标跟踪, 调整参数 ϵ 、 λ 和 N , 令 $\epsilon = 0.05$, $\lambda = 5$ 为正常闪烁噪声环境, $\epsilon = 0.1$, $\lambda = 10$ 为强闪烁噪声环境. 粒子数 $N = 100$ 时仿真的目标跟踪曲线如图 4 和图 5 所示, 连续转弯时模型仿真结果数据比较如表 2 所示.

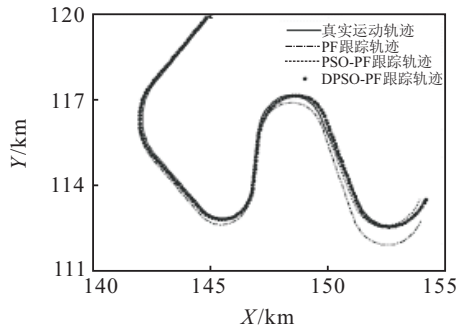


图 4 正常闪烁噪声下的目标跟踪轨迹

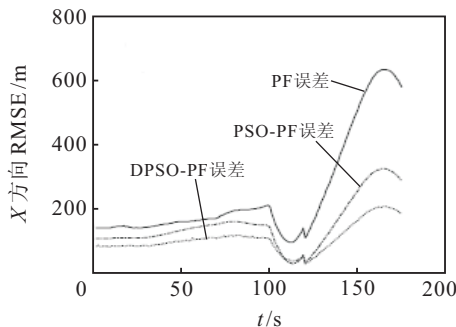


图 5 正常闪烁噪声下的目标跟踪 X 方向误差

表 2 连续转弯模型仿真结果数据比较

参 数	算 法	RMSE	运算时间/s
$(N = 100, \epsilon = 0.05, \lambda = 5)$	PF	238.7362	0.8734
	PSO- PF	137.7302	1.3627
	DPSO- PF	106.7362	1.2441
$(N = 100, \epsilon = 0.1, \lambda = 10)$	PF	263.6272	0.8624
	PSO- PF	167.7302	1.3737
	DPSO- PF	129.7361	1.2590
$(N = 500, \epsilon = 0.1, \lambda = 10)$	PF	217.0476	4.3421
	PSO- PF	156.8372	6.8013
	DPSO- PF	120.6736	5.9752

4.3.4 仿真结果分析

从图 4 和图 5 可以看出: 目标跟踪的误差随着转弯次数的增多而增加, 其中 PF 的跟踪轨迹在第 3 个转

弯处开始出现较明显的偏离, 在第 4 个转弯处几乎已经完全偏离目标的真实轨迹; PSO-PF 在第 4 个转弯处的偏离比较明显, 而 DPSO-PF 在经历了 4 个转弯后仍然可以精确地跟踪到目标, 跟踪轨迹与真实轨迹的偏差不明显. 从表 2 可以看出: 在不同的参数环境下, DPSO-PF 精度最高, 并且运算时间较 PSO-PF 有一定程度的降低, 具有较强的抗噪声干扰能力, 可以满足雷达目标跟踪对精确性、实时性和抗干扰性的要求. 此外, 随着粒子数的增加, PSO-PF 和 DPSO-PF 的精度并不像 PF 那样有明显的提高, 表明基于 PSO 的粒子滤波对粒子数量的依赖不强, 这也说明了基于 PSO 的粒子滤波可以利用较少的粒子数达到所需的精度.

5 结 论

本文通过分析标准 PSO-PF 算法的粒子邻域的局限性, 提出了一种适用于雷达目标跟踪的新型粒子群优化粒子滤波算法 DPSO-PF. 该算法能够自适应地调整粒子的邻域环境, 每个粒子可以按照自身的位置状态和邻域环境参数, 在多样性因子、邻域扩展策略和邻域限制策略的共同作用下对邻域粒子的数量进行合理的优化调整, 降低了局部最优现象发生的概率, 提高了优化成功率, 使算法同时具有较好的寻优能力和较快的收敛速度. 实验结果表明, DPSO-PF 的精确性和实时性较 PSO-PF 都有一定程度的提高, 在雷达目标跟踪领域具有较高的应用价值.

参考文献(References)

- [1] 胡振涛, 潘泉, 梁彦. 基于粒子滤波的模型自适应机动目标跟踪算法[J]. 控制与决策, 2008, 23(12): 1333-1337. (Hu Z T, Pan Q, Liang Y. Model adaptive maneuvering target tracking algorithm based on particle filtering[J]. Control and Decision, 2008, 23(12): 1333-1337.)
- [2] Gordon N, Salmond D J, Smith A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation[J]. IEEE Proc F: Radar and Signal Processing, 1993, 140(2): 107-113.
- [3] Doucet A, Godsill S. On sequential Monte Carlo sampling methods for Bayesian filtering[J]. Statistics and Computing, 2000, 10(1): 197-208.
- [4] Kong A, Liu J. Sequential imputations and Bayesian missing data problems[J]. J American Statistical Association, 1994, 89(2): 278-288.
- [5] Liu Y M, Zhao Q Z, Shao Z Z, et al. Particle swarm optimizer based on dynamic neighborhood topology[C]. Proc of the Intelligent Computing 5th Int Conf on Emerging Intelligent Computing Technology and Applications. Ulsan, 2009: 794-803.
- [6] 武忠勇, 缙锦, 赵志强. 具有自适应邻域探测机制的改进型 PSO 算法[J]. 小型微型计算机系统, 2010, 31(9):

- 1838-1845.
(Wu Z Y, Gou J, Zhao Z Q. Improved particle swarm optimization based on self-adaptive neighborhood explored[J]. J of Chinese Computer Systems, 2010, 31(9): 1838-1845.)
- [7] Hu X H, Eberhart R. Multiobjective optimization using dynamic neighborhood particle swarm optimization[C]. Proc of the 2002 Congress on Evolutionary Computation. Honolulu, 2002: 1677-1681.
- [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.
(Fang Z, Tong G F, Xu X H. Particle swarm optimized particle filter[J]. Control and Decision, 2007, 22(3): 273-277.)
- [9] Zhang G, Cheng Y, Yang F, et al. Particle filter based on PSO[C]. Intelligent Computation Technology and Automation. Changsha, 2008: 121-124.
- [10] Jing W, Zhao H, Song C H, et al. A optimized particle filter based on PSO algorithm[C]. Int Conf on Future BioMedical Information Engineering. Sanya, 2009: 122-125.
- [11] Klamargias A D, Parsopoulos K E. Particle filtering with particle swarm optimization in systems with multiplicative noise[C]. Proc of the 10th Annual Conf on Genetic and Evolutionary Computation. New York, 2008: 57-62.
- [12] Song C H, Zhao H, Jing W, et al. A new particle swarm optimization based unscented particle filtering[C]. The 3rd Int Conf on Bioinformatics and Biomedical Engineering. Beijing, 2009: 1-4.
- [13] Yihua Yu, Xuanyuan Zheng. Particle filter with ant colony optimization for frequency offset estimation in OFDM systems with unknown noise distribution[J]. Signal Processing, 2011, 91(5): 1339-1342.
- [14] Sanjeev M, Maskell S. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking[J]. IEEE Trans on Signal Processing, 2002, 50(2): 174-188.
- [15] Zhang W, Liu Y T. Adaptive particle swarm optimization for reactive power and voltage control in power systems[J]. Lecture Note in Computer Science, 2005, 3612: 449-452.
- [16] Ying Li, Bendu Bai, Yanning Zhang. Improved particle swarm optimization algorithm for fuzzy multi-class SVM[J]. J of Systems Engineering and Electronics, 2010, 21(3): 509-513.
- [17] 张晓绩, 戴冠中, 徐乃平. 遗传算法种群多样性的分析研究[J]. 控制理论与应用, 1998, 15(1): 17-23.
(Zhang X J, Dai G Z, Xu N P. Study of diversity of population in genetic algorithm[J]. Control Theory & Application, 1998, 15(1): 17-23.)
- [18] Xu X, Li B. Adaptive Rao-Blackwellized particle filter and its evaluation for tracking in surveillance[J]. IEEE Trans on Image Processing, 2007, 16(3): 838-849.
- [19] 叶龙, 王京玲, 张勤. 遗传重采样粒子滤波器[J]. 自动化学报, 2007, 33(8): 885-887.
(Ye L, Wang J L, Zhang Q. Genetic resampling particle filter[J]. Acta Automatica Sinica, 2007, 33(8): 885-887.)
- [20] Durovic Z M, Kovacevic B D. QQ-plot approach to robust Kalman filtering[J]. Int J of Control, 1994, 61(4): 837-857.

~~~~~

(上接第192页)

- [18] 王国胤. Rough集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001: 13-36.  
(Wang G Y. Rough set theory and knowledge acquisition[M]. Xi'an: Xi'an Jiaotong University Press, 2001: 13-36.)
- [19] Jiang F, Sui Y F, Cao C G. Some issues about outlier detection in rough set theory[J]. Expert Systems with Applications, 2009, 36(3): 4680-4687.
- [20] Jiang F, Sui Y F, Cao C G. A rough set approach to outlier detection[J]. Int J of General Systems, Taylor and Francis, 2008, 37(5): 519-536.
- [21] Nguyen S H, Nguyen H S. Some efficient algorithms for rough set methods[C]. Proc of the 6th Int Conf on Information Processing and Management of Uncertainty. Granada: Springer-Verlag, 1996: 1451-1456.
- [22] Stanfill C, Waltz D. Toward memory-based reasoning[J]. Communications of the ACM, 1986, 29(12): 1213-1228.
- [23] Bay S D. The UCI KDD repository[DB/OL]. [2011-10-15]. <http://kdd.ics.uci.edu>.
- [24] Harkins S, He H X, Williams G J, et al. Outlier detection using replicator neural networks[C]. Proc of the 4th Int Conf on Data Warehousing and Knowledge Discovery. Aix-en-Provence: Springer-Verlag, 2002: 170-180.
- [25] Hu Q H, Pedrycz W, Yu D R, et al. Selecting discrete and continuous features based on neighborhood decision error minimization[J]. IEEE Trans on Systems, Man, and Cybernetics: Part B, 2010, 40(1): 137-150.