

基于 GPU 的 BBPSO-PF 算法及其在故障检测中的应用*

曹 洁^{1,2,3}, 胡文东¹, 王进花², 余 萍², 赵伟吉²

(1. 兰州理工大学 计算机与通信学院, 甘肃 兰州 730050; 2. 兰州理工大学 电气工程与信息工程学院, 甘肃 兰州 730050;
3. 甘肃省制造业信息化工程研究中心, 甘肃 兰州 730050)

摘要: 针对粒子滤波算法在重采样环节出现粒子贫乏导致算法精度不高的问题, 通常采用在状态估计过程中增加粒子数量, 但这种方法会降低算法实时性, 提出了基于图形处理单元(GPU)的骨干粒子群算法优化粒子滤波算法。首先利用骨干粒子群算法优化粒子滤波重采样, 解决了粒子贫乏的缺点。利用骨干粒子群算法中粒子群体之间相互独立运行的特点, 在 GPU 上并行实现骨干粒子群优化的粒子滤波算法, 解决粒子滤波算法在重采样过程中因数据关联而无法充分并行计算的问题。最后, 将其应用到变桨距系统的故障检测中, 提高故障检测的准确度和实时性。实验结果表明: 该方法相较于随机重采样的粒子滤波算法误差降低了 31.2%, 实时性提高了 82.7%。

关键词: 重采样; 并行计算; 粒子滤波; 骨干粒子群优化算法; 实时性

中图分类号: TP391 文献标识码: A 文章编号: 1000-9787(2021)05-0157-04

BBPSO-PF algorithm based on GPU and its application in fault detection*

CAO Jie^{1,2,3}, HU Wendong¹, WANG Jinhua², YU Ping², ZHAO Weiji²

(1. College of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China;
2. College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China;
3. Gansu Manufacturing Information Engineering Research Center, Lanzhou 730050, China)

Abstract: To solve the problem of particle filtering (PF) algorithm occur poor particle quantity in resampling, usually uses increasing number of particles in the process of state estimation, but this method will reduce the real-time performance of the algorithm, a backbone particle swarm optimization (BBPSO) PF algorithm based on GPU is proposed. Firstly, the backbone particle swarm optimization algorithm is used to optimize particle filter resampling. Secondly, using the characteristics of independent operation between particle groups in backbone particle swarm optimization, the particle filtering algorithm of backbone particle swarm optimization is implemented in parallel on GPU, so as to solve the problem that the particle filtering algorithm cannot fully parallel compute due to data association in the process of resampling. Finally, it is applied to the fault detection of variable pitch system to improve the accuracy and real-time of fault detection. Experimental results show that the error of the proposed method is reduced by 31.2% and the real-time performance is improved by 82.7%.

Keywords: resampling; parallel computing; particle filtering (PF); bare bones particle swarm optimization algorithm (BBPSO); real-time

0 引言

变速恒频的双馈发电机主要包含了空气动力系统、变桨距系统、传动系统、控制器和功率转换系统等^[1], 变桨距系统主要用于桨距角的控制, 以适应多变的风速, 由于该系统频繁地在停止与启动状态间跳变, 导致发生故障的概率较高, 因而提高该系统故障诊断的准确度和实时性具有重要的意义。

粒子滤波 (particle filtering, PF) 算法是一种基于蒙特卡洛与递推贝叶斯估计的概率统计算法^[2], 可以有效地处理非线性、非高斯噪声环境下的状态估计问题, 被广泛用于目标跟踪^[3,4]、故障诊断^[5,6]、机器人定位^[7]、人脸跟踪^[8]等领域。而变桨距系统是一个典型的非线性、强耦合系统, 故 PF 算法适合于该系统的故障诊断。PF 算法具有多种重采样方法, 经典方法包括有多项式重采样 (multinomial resampling, MR)、系

收稿日期: 2019-10-17

* 基金项目: 国家自然科学基金资助项目 (61763028); 甘肃省自然科学基金资助项目 (1506RJZA105)

统重采样 (systematic resampling, SR)、残差重采样 (residual resampling, RR) 和随机重采样 (random resampling, RAR)。在状态估计中, 当粒子数足够多时, 一定程度上提高了状态估计精度, 但在粒子数较少的情况下, 存在粒子退化和样本贫化等现象。基于此, 国内外许多学者对 PF 算法进行了改进, 主要分两种: 一是引入遗传算法 (GA) 思想改进的 PF 算法 (GA-PF)^[9]; 另一种是引入粒子群优化 (PSO) 思想改进的 PF 算法 (PSO-PF)^[10, 11]。

在对 PF 算法进行改进的现有研究中, 虽然提高了算法的精度, 但同时增加了算法复杂度, 难以将其应用到实时性要求较高的领域中。为了解决这一问题, 研究学者利用图形处理单元 (graphic processing unit, GPU) 的多线程优势, 提出以并行计算的方式提高算法实时性。文献 [12] 提出了一种并行规约的差分进化粒子滤波方法, 解决粒子滤波计算速度问题。文献 [13] 提出了利用 GPU 平台并行实现粒子滤波算法, 提高计算速度。通过 GPU 并行实现粒子滤波算法, 虽然能够提高算法实时性, 但粒子滤波算法在重采样环节中粒子之间交互频繁, 不能充分发挥出 GPU 的并行优势。

本文提出一种基于 GPU 的骨干粒子群优化 PF 故障检测方法, 该方法利用骨干粒子群算法优化 PF 算法重采样环节, 提高算法精确度, 同时利用骨干粒子群算法所具有的并行结构特点, 将结合后的算法并行化, 提高算法实时性。最后将其应用到故障检测中, 提高故障检测的精确度和实时性。

1 算法简介

1.1 PF 算法

PF 算法可以有效地处理非线性、非高斯噪声环境下的状态估计问题, 基本思想是利用随机样本集合对条件后验概率密度函数进行近似估计, 从而计算样本均值得到状态估计结果。

从先验概率密度 $p(x_{k-1} | y_{1:k-1})$ 中采样 N 个样本点, 得到后验概率密度为

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

式中 δ 为狄拉克函数, w_{k-1}^i 为 $k-1$ 时刻第 j 个粒子的权重, x_{k-1}^i 为 $k-1$ 时刻的状态值, x_{k-1}^j 表示 $k-1$ 时刻第 j 个粒子。

通过多次迭代, 根据粒子权重对粒子进行更新, 最后对粒子权重进行归一化, 对例子加权求和, 得到状态估计结果。

1.2 骨干粒子群优化算法

骨干粒子群优化 (bare bones particle swarm optimization, BBPSO) 算法利用全局最优值和局部最优值进行高斯采样来完成粒子位置的更新, 无需粒子交互。与经典的粒子群算法相比, BBPSO 算法消除了惯性系数、速度阈值、加速系数等控制参数, 使得算法结构更为简洁且易于操作。粒子的位置更新方程为

$$X_{ij}(t+1) = N(u_{ij}, \sigma_{ij}) \quad (2)$$

$$u_{ij}(t) = 0.5(p_{ij}(t) + G_{ij}(t)) \quad (3)$$

$$\sigma_{ij}(t) = |p_{ij}(t) - G_{ij}(t)| \quad (4)$$

在式 (2) ~ 式 (4) 中, $X_{ij}(t+1)$ 为 $t+1$ 时刻粒子 i 位置的, 第 j 维分量 $i=1, 2, \dots, J; j=1, 2, \dots, D$; 其中 J 为种群规模; D 为带求解问题的维数; p_i 为粒子 i 自身的最优位置 (即 $pbest$); G 为群体最优位置 (即 $gbest$); 符号 $N(u_{ij}, \sigma_{ij})$ 为高斯分布。

由以上公式可知, 粒子的位置更新是通过以 $pbest$ 与 $gbest$ 的算术平均值 u_{ij} , $pbest$ 与 $gbest$ 的距离为标准差 σ_{ij} 的高斯采样得到的。该算法也需要局部最优值 (即 $pbest$) 和群体最优值 ($gbest$) 的更新, 更新方程如下

$$P_i(t+1) = \begin{cases} X_i(t+1) & f(X_i(t+1)) < f(p_i(t)) \\ p_i(t) & \text{else} \end{cases} \quad (5)$$

$$G(t+1) = \begin{cases} P_k(t+1) & f(P_k(t+1)) < f(G(t)) \\ G(t) & \text{else} \end{cases} \quad (6)$$

式 (5) 式 (6) 中 $p_i(t)$ 为第 i 个粒子自身最优位置 (即 $pbest$), $G(t)$ 为群体最优位置 (即 $gbest$), $f(t)$ 为算法中的适应度函数。

1.3 并行的 BBPSO-PF 算法

在状态估计中, PF 算法经过数次迭代之后会出现粒子贫化现象。利用 BBPSO 算法优化 PF 算法 (BBPSO-PF), 通过 BBPSO 算法的局部最优解和全局最优解两个参数构造高斯函数进行高斯采样。而 BBPSO 算法结构简单, 每个粒子种群分别利用自己的局部最优解与全局最优解联合进行粒子更新。粒子种群相互独立运行, 适合并行实现。可以利用该算法取代 PF 算法重采样环节, 克服 PF 算法重采样环节不能充分并行的缺点。

首先设置初始状态, 通过 PF 算法的状态方程进行状态更新, 利用测量方程初始化该时刻的状态估计粒子, 并将初始化的粒子传递给 BBPSO 算法进行处理。根据提前设置好的迭代结束条件结束重采样, BBPSO 算法将得到的状态估计结果传送到 PF 算法中进行状态的更新。算法流程如下:

步骤 1 状态初始化 (第一个状态 $k=0$), 从先验分布品 $p = (x_0)$ 中抽取初始化状态 X_0^i , 其中 $i \in (1, 2, 3, \dots, N)$, 设置骨干粒子群算法参数 (迭代结束条件, 取值范围, 种群规模)。

步骤 2 重要性采样阶段 $k=k+1$, 采样 $\hat{X}_k^i \sim q(X_k | X_{0:k-1}^i, Z_{1:k})$ 。计算适应度值。利用适应度值寻找个体最优值 ($pbest$) 和全局最优值 ($gbest$)。

步骤 3 选择阶段 (重采样) 根据式 (2) ~ 式 (4) 更新粒子。根据式 (5) 和式 (6) 更新 $pbest$ 和 $gbest$ 。判断是否达到迭代终止条件, 若达到终止条件, 则执行步骤 2, 若否则执行步骤 3。

并行的 BBPSO-PF 算法结构流程图如图 1 所示, 可以看出, BBPSO 算法在求粒子适应度、个体最优值、粒子更新时都是以粒子群为单位, 相互独立运行, 这种结构适合利用

多线程并行实现, 将其应用到 PF 算法的重采样环节能够解决 PF 算法重采样环节不能充分并行的问题。

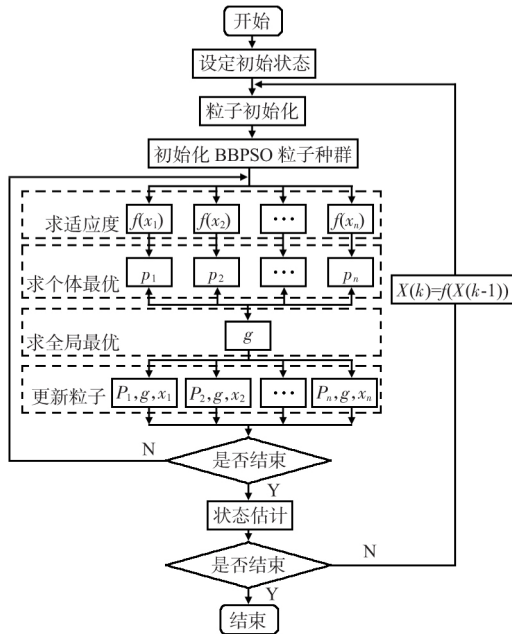


图 1 并行的 BBPSO-PF 流程框图

2 实验仿真与分析

2.1 实验模型

在相同条件下对 BBPSO-PF 算法和 4 种不同重采样方法的 PF 算法从精确度和实时性两个方面进行分析和比较。观察 5 种算法的性能。本文采用一个广泛应用的标量模型, 其状态模型和观测模型为

$$X_k = 0.5X_{k-1} + 2.5X_{k-1} / (1 + X_{k-1}^2) + 8\cos(1.2k) + W_{k-1} \quad (7)$$

$$Z_k = X_k^2 / 20 + V_k \quad (8)$$

实验参数的设置: 初始状态设置为 $x_0 = 0.1$; 状态噪声 $w_k \sim N(0, 0.1)$; 观测噪声 $v_k \sim N(0, 0.1)$; 仿真周期 $T = 50$ 。

2.2 算法精度对比

图 2 给出了粒子数 $N = 6000$ 时, 不同重采样方法的 PF 算法状态估计结果。由图可见, BBPSO-PF 粒子滤波算法状态估计结果更接近真实状态值。

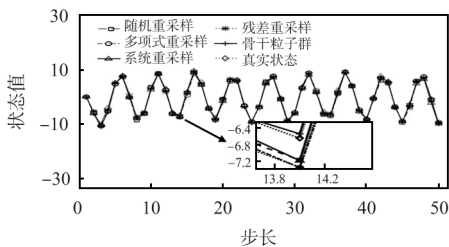


图 2 状态估计结果

2.3 算法实时性对比

在 Window7 系统下采用 VisualStudio2013(CUDA7.5) 开发环境进行实验, CPU 为 Intel® Core™ i3-4150, GPU 为 NVIDIA GeForce GTX 950。在 CUDA 架构上并行实现 BBPSO-PF 算法和 4 种不同重采样方法的 PF 算法, 对比运行时间。为了排除实验的偶然性, 本次实验采用多种粒子数进行分析对比, 每组粒子数均重复运行 10 次, 求取平均

运行时间。实验结果如图 3 所示。可以看出 BBPSO-PF 算法的实时性优于其它重采样方法的 PF 算法, 表明了基于 GPU 的 BBPSO-PF 算法可以有效地提高算法实时性。

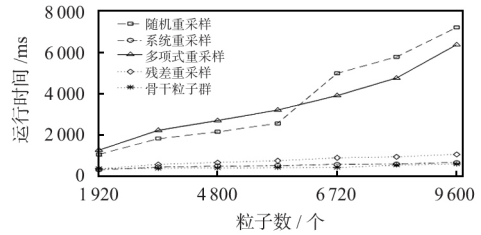


图 3 算法并行执行时间

3 BBPSO-PF 在故障检测中的应用

3.1 故障类型设计

选择文献[14]中提供的风力机桨距系统的数学模型, 根据该数学模型建立的二维状态空间模型如式(9)所示

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w_k \\ y_k = Cx_k + f + v_k \end{cases} \quad (9)$$

状态变量为 $x(t) = [\beta(t) \quad \dot{\beta}(t)]^T$, 输入向量为

$$u(t) = \beta_{ref}(t), \text{ 系数为 } A(t) = \begin{pmatrix} 0 & 1 \\ -\omega_n^2(t) & -2\zeta(t)\omega(t) \end{pmatrix},$$

$$B(t) = [0 \quad \omega_n^2(t)]^T, C = [1 \quad 0], f \text{ 为传感器故障, } y \text{ 为桨距角}$$

w_k 和 v_k 分别为系统噪声和观测噪声。实验中噪声设置为 $w \sim \Gamma(0.1, 0.1)$, $v \sim \Gamma(0.1, 0.1)$ 。风力机系统中的其他参数如文献[14]所示。算法中粒子数 $N = 4800$, 周期 $T = 200$, 离散步长 $\Delta T = 0.01$ 。故障模型设置为三种类型, 分别为:

$$\text{故障 1: 突变故障 } f = \begin{cases} 0 & 0 \leq t \leq 100 \\ 1 & 100 \leq t \leq 200 \end{cases}$$

$$\text{故障 2: 缓变故障 } f = \begin{cases} 0 & 0 \leq t \leq 100 \\ 2\sin\left(\frac{1}{400}\pi t\right) & 100 \leq t \leq 200 \end{cases}$$

$$\text{故障 3: 混合故障 } f = \begin{cases} 0 & 0 \leq t < 80 \\ 1 & 80 \leq t < 130 \\ 2\sin\left(\frac{1}{400}\pi t\right) & 130 \leq t \leq 200 \end{cases}$$

将固定阈值设置为 0.1, 若系统状态估计测量值与真实测量值的残差超过固定阈值, 则认为系统出现故障。

3.2 故障检测结果及分析

利用 BBPSO-PF 算法进行状态估计, 图 4 分别为突变故障、缓变故障和混合故障的故障检测结果。可以看出, PF 算法和 BBPSO-PF 算法在对应的时间点都能检测出相应的故障。但在正常状态的时间段里, PF 算法多次将正常状态检测成故障状态, 而 BBPSO-PF 算法出现该情况较少。从而验证了 BBPSO-PF 算法能很好地提高算法精度。

表 1 记录了 5 次随机重采样方法和 BBPSO-PF 算法故障检测时间及均值。由表 1 可见, 在 GPU 上利用 BBPSO-PF 算法进行故障检测相较于串行 PF 实时性提高了 94.7%, 相较于并行的 PF 算法实时性提高了 89.8%。

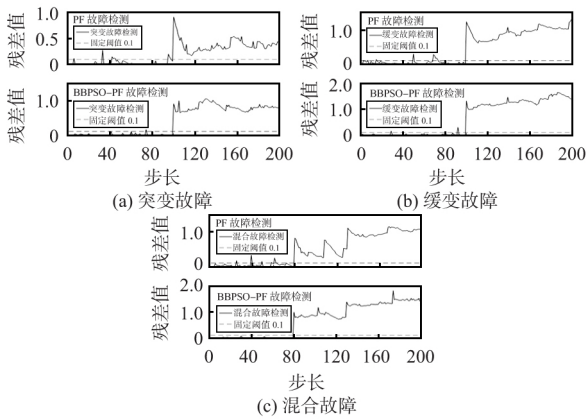


图 4 三种故障检测结果

表 1 故障检测时间

	1	2	3	4	5	AVG
PF 串行	9954	9876	9893	9887	9903	9902.6
BBPSO-PF 串行	1547	1531	1536	1543	1541	1539.2
PF 并行	5237	5186	5194	5207	5191	5203.0
BBPSO-PF 并行	536	527	524	531	527	529.0

4 结 论

针对 PF 算法在精确度和实时性两个方面存在的问题。本文提出面向 GPU 的 BBPSO-PF 的故障检测方法,该方法提高故障检测的精确度和实时性。本文的主要内容和结论:1) 利用 BBPSO 算法以全局最优值和局部最优值联合进行高斯采样的特点,将其与 PF 算法相结合,提高 PF 算法的精度。2) 利用 BBPSO 算法具有并行结构的特点,取代 PF 算法重采样环节,解决其不能充分并行的问题。3) 将面向 GPU 的 BBPSO-PF 的故障检测方法应用到变桨距系统的故障诊断中。实验结果表明,该方法对故障检测的精度和实时性皆有很大的提高。

参考文献:

[1] LIU X ,GAO Z. Takagi-Sugeno fuzzy modelling and robust fault reconstruction for wind turbine systems [C]// International Conference on Industrial Infor-matics ,IEEE 2017: 492 - 495.

[2] 吕德潮,范江涛,韩刚瓷,等. 粒子滤波综述[J]. 天文研究与技术 2013 ,10(4): 397 - 409.

[12] GALOOGAHI H K ,SIM T ,LUCY S. Learning background-aware correlation filters for visual tracking [C]// Proceedings of the IEEE International Conference on Computer Vision ,2017: 1144 - 1152.

[13] MUELLER M ,SMITH N ,GHANEM B. Context-aware correlation filter tracking [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ,2017: 1387 - 1395.

[14] BOYD S ,PARIKH N ,CHU E ,et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations & Trends in Machine Learning ,2010 , 3(1): 1 - 122.

[15] WANG M ,LIU Y ,HUANG Z. Large Margin object tracking with

[3] 丁婷婷,高美凤. 改进粒子滤波的无线传感器网络目标跟踪算法[J]. 传感器与微系统 2016 ,35(7): 140 - 142 ,146.

[4] 谷静,史健芳. 分布式粒子滤波算法在目标跟踪中的应用[J]. 传感器与微系统 2014 ,33(8): 158 - 160.

[5] 孟志强,朱志亮,朱建波,等. 基于粒子滤波的分布式智能故障诊断系统研究[J]. 湖南大学学报(自然科学版), 2018(2): 87 - 94.

[6] SUN Y ,RAN X ,LI Y ,et al. Thruster fault diagnosis method based on Gaussian particle filter for autonomous underwater vehicles[J]. International Journal of Naval Architecture and Ocean Engineering 2016 ,8(3): 243 - 251.

[7] 毛玲,李振波,陈佳品. 基于相对定位的多移动微机器人协作定位方法[J]. 传感器与微系统 2018 ,37(8): 46 - 48.

[8] WANG T ,WANG W ,LIU H ,et al. Research on a face real-time tracking algorithm based on particle filter multi-feature fusion [J]. Sensors 2019 ,19(5): 1245.

[9] 蔡登禹,刘以安. 一种基于遗传算法的改进粒子滤波器[J]. 计算机仿真 2018 ,35(7): 221 - 225.

[10] 田连杰,孙希延,纪元法. 基于遗传粒子滤波的多径信号估计算法[J]. 计算机仿真 2017 ,34(12): 36 - 40.

[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] 曹洁,黄开杰,王进花. GPU 加速的差分进化粒子滤波算法[J]. 计算机应用研究 2018 ,35(7): 51 - 55.

[13] 孙伟平,向杰,陈加忠,等. 基于 GPU 的粒子滤波并行算法[J]. 华中科技大学学报(自然科学版) 2011(5): 63 - 66.

[14] WU D ,LIU W ,SONG J ,et al. Fault estimation and fault-tolerant control of wind turbines using the SDW-LSI algorithm[J]. IEEE Access 2016 ,4: 7223 - 7231.

作者简介:

曹 洁(1966 -) 女,教授,博士研究生导师,研究领域为智能信息处理,信息融合,智能交通。

胡文东(1992 -) 男,硕士研究生,研究方向为并行与分布式处理,E-mail: 2457741164@qq.com。

王进花(1978 -) 女,副教授,硕士研究生导师,研究领域为多源信息融合,故障诊断。

circulant feature maps [C]// Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition , 2017: 4800 - 4808.

[16] WU Y ,LIM J ,YANG M H. Object tracking benchmark [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence ,2015 , 37(9): 1834 - 1848.

作者简介:

王科平(1976 -) 女,博士,副教授,主要研究领域为图像识别,目标跟踪,深度学习。

朱鹏飞(1992 -) 男,硕士研究生,研究方向为图像识别,目标跟踪。

杨 艺(1980 -) 男,博士,讲师,主要研究领域为图像识别,深度学习,强化学习。