

汽轮机虚拟传感器数据前处理 DPS算法及其应用

王晓放^{*1}, 李刚¹, 周宇阳², 陈广兵²

(1.大连理工大学 能源与动力学院, 辽宁 大连 116024

2.上海新华控制工程有限公司, 上海 200245)

摘要: 针对汽轮机虚拟传感器数据前处理问题, 根据 DPS算法的基本原理编制 C程序. 为确保算法的通用性, 由工程应用中各种原始参数对应的典型线形给出 DPS算法的简化结果; 给出影响数据取舍精度的公差准则. 以某汽轮机组虚拟传感器数据前处理的一个实例为例, 应用本算法有效地解决了虚拟传感器的实时仿真问题. 结果表明: DPS算法应用于工程中大部分的数据精简都能够起到行之有效的作用, 而所建议的公差准则对于工程中的大多数原始数据线形都是适用的.

关键词: 汽轮机; 虚拟传感器; 数据处理; DPS算法

中图分类号: TK14; TH89 **文献标识码:** A

0 引言

随着计算机技术的飞速发展, 人们每天都要解决大量的数字信息, 尤其是在工程实际和科学计算中. 然而, 监测系统采集的数字信息往往庞大而繁冗, 以新华控制工程有限公司 XDPS系统为例, 一台引进型 300MW 机组 1h 的历史数据大约为 50 MB. 因此, 提取出关键有效的数字信息是解决工程问题的首要前提.

传感器是信息采集系统的首要部件, 负责对各种信息的感知、采集、转换、传输和处理. 而虚拟传感器并不是对待测信息直接进行上述一整套服务, 它仅对待测信息的相关信息进行如上述处理, 然后通过信息之间的关联, 回归出相关信息与待测信息之间的函数关系, 以达到间接测量的目的. 虚拟传感器是现场物理参数的一个解析或者数值模型, 它与实际传感器处于并发的的工作状态, 在实际传感器信号不发生突变的情况下, 虚拟传感器可以有效地提示出信号的失真或者漂移, 并且可以直接作为暂时性的代替信号, 这种检错与容错的手段对于那些关乎机组安全、经济运行的

重要测点是十分必要的. 上述函数关系可表示为

$$X = f(X_1, X_2, \dots, X_n)$$

式中: X 为待测信息 (被虚拟参数); X_1, X_2, \dots, X_n 为 X 的相关信息 (相关参数).

虚拟传感器的品质取决于训练样本 (相关信息的集合) 是否真实而准确地反映现场对象的特性, 从建模的角度看必须简洁而不失代表性. 本文采用 DPS算法将那些有价值的点保留下来, 以节省数据存储的空间和分析数据的时间, 缩短工程开发的周期, 实现对资源合理有效的配置.

1 DPS算法

DPS (Douglas-Peucker line simplification) 算法被广泛应用于数据的精简和压缩.

DPS算法采用点到边线的距离作为选取关键点的判定准则^[1]. 首先, 确定直线的起点 P_s 和端点 P_e 并将其连接, 假设它是第一次简化后的直线图形 L_0 . 然后, 求取除起点和端点外, 其他所有点 P_i 到假设直线 L_0 的距离. 根据精度要求, 选择公差 Δ 作为数据取舍的依据, 如果上述计算的

收稿日期: 2006-04-16; 修回日期: 2007-07-30.
作者简介: 王晓放* (1960-), 女, 教授.

都小于公差 X 也就是所得点都落在 X 公差带范围内, 则这些点可以忽略, 最初假设的直线 L_0 就成为简化后的直线. 函数 $Dis()$ 是求解中间点到两个端点的距离, 数学表达式为

$$Dis(P_s, P_e, P_i) < X$$

如果有一个或几个距离都大于公差 X 那么就选择到假设直线距离最远的一个点, 这点称为关键点^[2]. 选取关键点的过程如图 1 所示.

第一个关键点选出以后, 再加上起点和端点, 这 3 个点依次连接形成两条直线 L_{11} 和 L_{12} . 假设它们就是第二次简化后的直线图形, 用同样的方法找出这两条直线间的关键点, 以此类推, 直到直线之间的所有点都落在 X 公差带范围内为止. 最后, 将所有关键点依次相连, 即为简化后的线形.

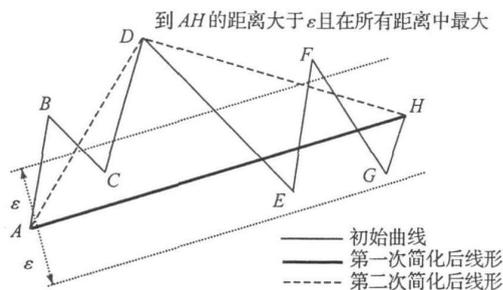


图 1 DPS 算法关键点选择

Fig. 1 Key points selected by DPS

2 DPS 算法的程序实现

DPS 算法的程序以 Windows 操作系统为开发平台, 采用 Visual C++ 6.0 为开发工具编制而成. 在设计这个程序时, 按照前述基本思想, 主体采用 3 个函数, 分别为 $DPSim()$ 、 $Dis()$ 和主调函数 $Main()$. $DPSim()$ 函数按照 DPS 算法筛选关键点; $Dis()$ 函数求取点到另外两点所构成的直线的距离. 因为该程序处理的对象数据量庞大, 所以, 对程序的效率和运行时间必须提出一些要求: 在主调函数 $Main()$ 中, 添加两类标志位, 以确保程序以最高的效率运行. 经过监测, 对于 24 000 个数据的样本, 运行时间平均为 40 ms (以上结果是在奔腾 IV 2.66 GHz, 内存 512 MB 配置下测试的). 程序的流程图如图 2 所示.

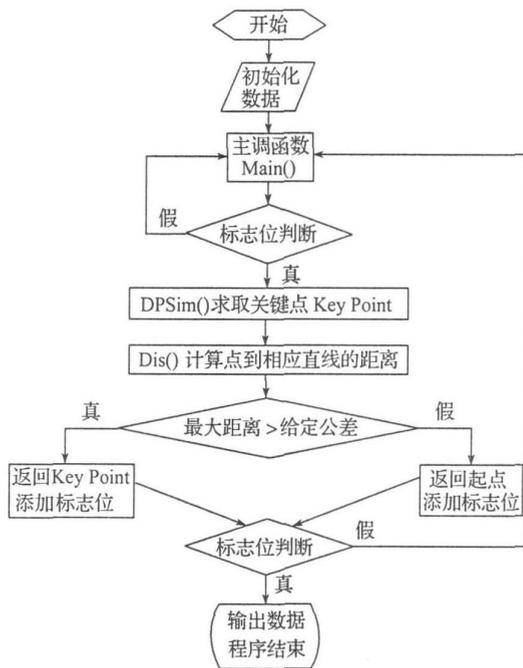


图 2 DPS 算法程序流程图

Fig. 2 Flow chart of DPS algorithm

3 DPS 算法的工程应用实例

3.1 公差选择对简化结果的影响

DPS 算法的公差 X 决定了数值取舍的精度, 不同的公差选择会导致不同的简化结果. 针对某电厂在线采集的原始数据, 采样周期 5 s, 样本量 14 400 个, 采用不同的公差进行简化, 输出结果线图如图 3 所示.

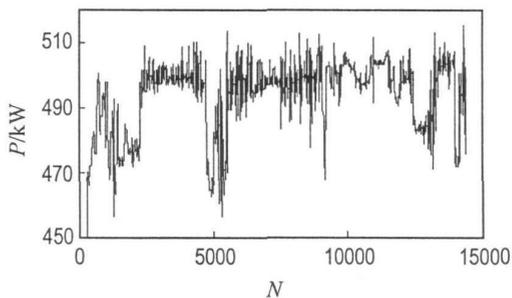


图 3 原始采集数据线图

Fig. 3 Chart of original data

图 4 为与图 3 选用相同的数据样本, 相同的采样时间, 采用不同简化公差获得的简化数据线图. 比较图 3 和图 4(a) 发现, 线形几乎没有变化, 而数据量却减少了 80%; 随着公差的变大, 线形的变化也越来越大, 与图 4(a) 相比, 图 4(b) 的线形基本保持原形, 只是在很小的局部 (图中画圈部

分) 发生了微弱的失真, 简化结果还算令人满意; 图 4(c) 由于选择的公差比较大, 线形已经较原形

有了较大变化, 局部失真比较严重. 公差对关键点数量的影响见表 1.

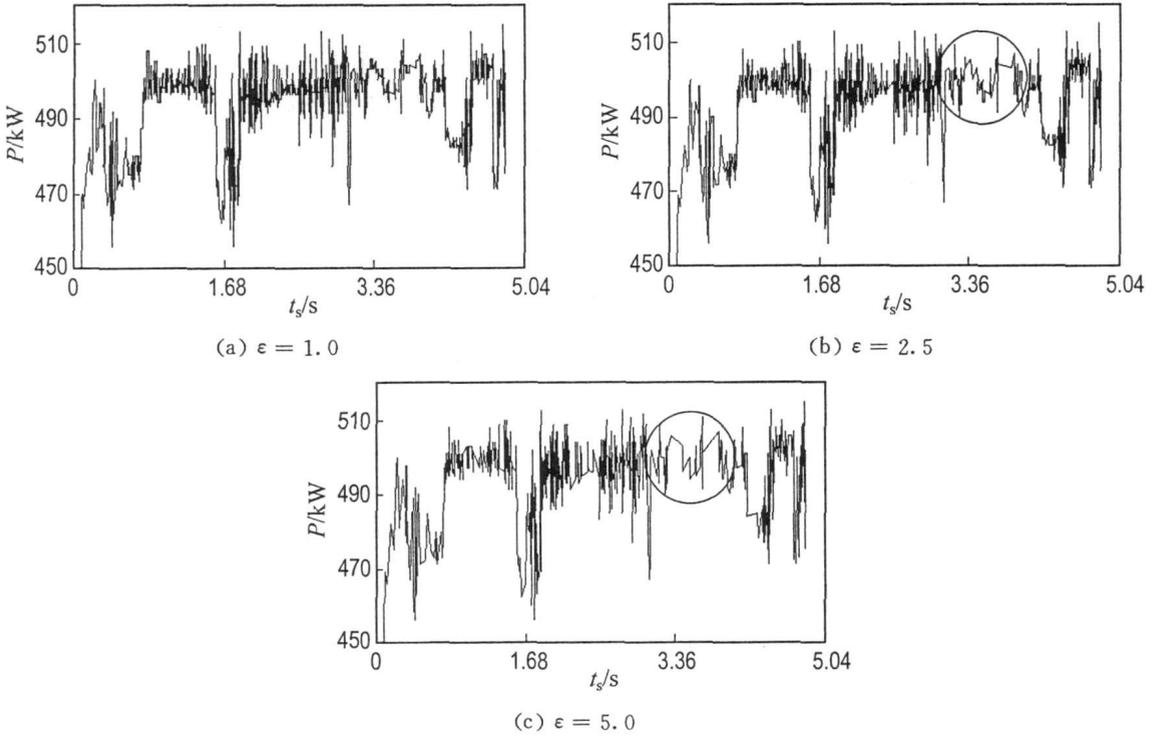


图 4 不同简化公差获得的简化数据线图

Fig. 4 Chart of the different compact data at different X

表 1 公差不同关键点数量的比较

Tab. 1 Comparison of key points at different X

X	关键点	压缩率 %
1.0	2 856	19. 8
2.5	1 490	10. 3
5.0	734	5. 1

显然, 选择一个合适的公差对简化结果很重要. 根据对大量数据进行简化分析和概括总结, 本文提出公差“2% ~ 4%”的准则. 即, 对于一个数据样本 {V₁, V₂, V₃, ..., V_n}, 取样本中最大值 V_{max} 和最小值 V_{min} 代数差的 2% ~ 4% 作为程序中的公差值, 数学表达式为

$$X = (V_{max} - V_{min}) \times (2\% \sim 4\%)$$

3.2 工程中典型线形简化实例

工程实际中遇到的线形多种多样, 有的非常陡峭, 变化剧烈; 有的则光滑平坦, 变化微小; 有的则具有明显的阶跃特征. 分别选取具有上述 3 种特征的参数, 利用 DPS 算法进行简化, 输出结果线图如图 5~ 7 所示, 其中原始数据样本量 23 760

个.

图 5 描绘的物理量是汽轮机转子体积平均温度, 公差为 10℃, 简化后只有 11 个有效点 (图中黑色圆点); 图 6 描绘的物理量是汽轮机转子外表面积温差, 公差为 3℃, 简化后有 94 个有效点; 图 7 描绘的物理量是汽轮机转速, 公差为 300 r/min, 简化后有 15 个有效点.

从以上三图可以看出, 图 5 和图 6 的公差选择, 基本符合前面叙述的“2% ~ 4%”的准则. 对于具有明显阶跃特征的曲线, 如图 7, 它的公差选择可以遵循“2% ~ 4%”的准则, 也可以根据阶跃程度和曲线的平滑程度适当放大.

实践表明, 公差的正确选择十分重要, 过大或过小都不好. 研究曲线的局部变化特征时, 往往希望曲线在局部具有更多的点, 如果将公差 X 取值很大, 虽然曲线形状基本保持不变, 但是描述曲线局部的点就会很少, 这样显然不利于把握局部特性, 也有悖于线形简化的初衷; 相反, 公差 X 也不能取得太小, 那样将起不到精简数据的作用.

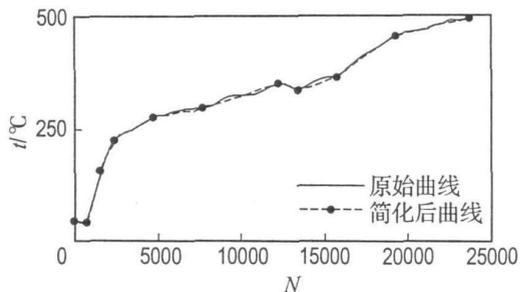


图 5 光滑线形简化前后比较

Fig. 5 Comparison between simplified smooth curve and before

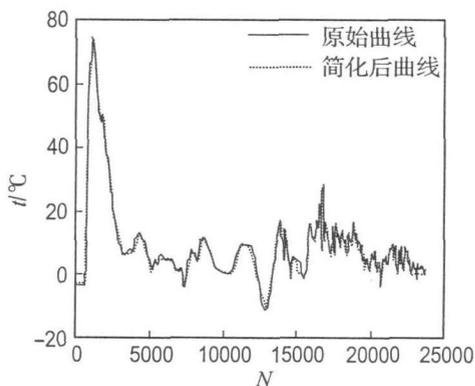


图 6 陡峭线形简化前后比较

Fig. 6 Comparison between simplified steep curve and before

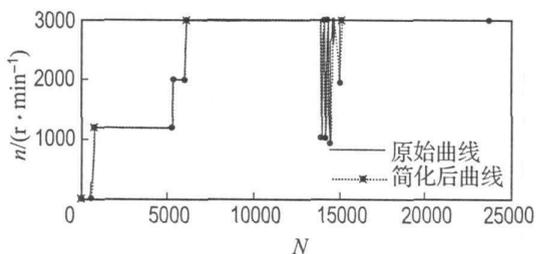


图 7 阶跃线形简化前后比较

Fig. 7 Comparison between simplified step curve and before

3.3 DPS算法在虚拟传感器中的应用

3.3.1 虚拟传感器的数据前处理 虚拟传感器的输入参数来自现场其他监测值,选择的标是与其与被虚拟参数相关性的强弱,那些与被虚拟参数相关性强的参数应该被优先选择. 虚拟传感器的输入参数组合有很多,如果有 n 个相关参数,其组合数为 $2^n - 1$, 14 个相关参数就有 $2^{14} - 1 = 16383$ 个参数组合. 借助遗传算法的全局搜索能力可以有效地找出参数优选的组合^[3]

然而,遗传算法的参数优选只是解决了虚拟传感器参数的横向比较,它可以优选出哪些参数的组合对于被虚拟参数是最优解,但是参数数据量大的问题使得遗传算法存在运行速度慢,不能实现实时仿真跟随的缺憾. 根据前述 DPS 算法的优点,如果将参数优选与数据精简结合起来(见图 8),既满足了参数优选的要求,又可解决计算速度过慢的问题.

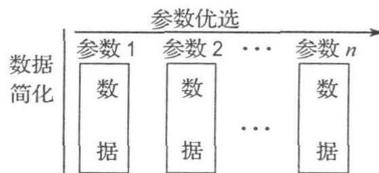


图 8 参数优选与数据简化
Fig. 8 Parameters optimization and data simplification

3.3.2 DPS算法在汽轮机虚拟传感器数据精简中的应用 转子温度是一个非常重要的物理量,它直接关系到转子的寿命和汽轮机的安全运行,现要对电厂中这个重要的物理量进行跟踪监测,采用真实传感器和虚拟传感器两种监测手段来实现. 在虚拟传感器中,选择汽轮机入口的主汽温和主汽压作为转子温度的相关信息,采用误差反传神经网络(back propagation network,简称 BP 网络)来回归主汽温、主汽压和转子温度的非线性关系.

由于电厂实际采集的上述数据非常庞大,利用 DPS 算法原理,将虚拟传感器的训练样本进行简化. 将一个数量级为 20 000 的训练样本,依次简化为 2 000 和 200,比较 3 组数据的回归时间,来体现 DPS 算法在虚拟传感器数据前处理中的重要作用.

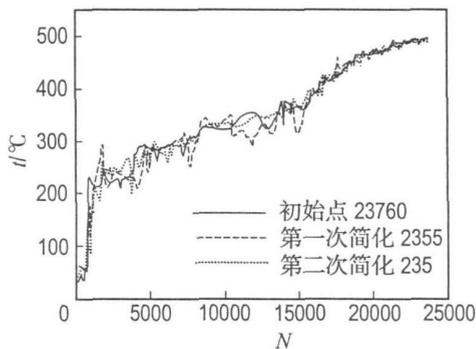


图 9 三组回归曲线的比较

Fig. 9 Comparison of three regressive curves

简化后的 3 组数据,按照 BP 网络算法,选取与标准值误差为 0.01 回归出 3 条曲线,如图 9 所示. 由图可以看出,3 条曲线趋势走向基本相同,在低温区和中温区误差较大,在高温区基本重合,这其中存在 BP 网络算法本身需要改进的地方. 本文更加关心简化前、后回归这些数据所需要的时间. 对于每一组数据分别训练 8 次,然后计算出平均时间. 时间比较见表 2,结果表明,经过简化的数据,回归时间明显减少,基本满足实时要求.

表 2 三组数据回归时间比较

Tab. 2 Comparison of three sets of data regression time

序号	原始数据	一次简化	二次简化
1	90.250	14.187	7.344
2	90.516	12.453	6.641
3	89.750	13.719	6.563
4	90.453	11.687	5.250
5	90.187	11.235	4.859
6	89.781	11.203	5.140
7	90.437	10.859	4.875
8	91.297	10.937	5.297
平均值	90.333	12.035	5.746

4 结 论

(1) 针对 DPS 算法,对公差的选择进行了概

括分析总结,并提出“2% ~ 4%”的准则,这一准则对于工程中的大多数线形都是适用的,简化结果令人满意.

(2) DPS 算法应用于工程中大部分的数据精简都能够起到行之有效的作用. 本文将 DPS 算法应用到汽轮机虚拟传感器的数据前处理中,使得 BP 神经网络的回归时间大大减少,解决了虚拟传感器实时仿真跟随的问题,对于监测机组安全运行和提高机组经济效益有重要工程意义.

(3) 需要注意的是,受系统误差等诸多因素的影响,原始数据中偶尔会出现异常增大或变小的情况,在按照上述原则选取公差时应予先行剔除.

参考文献:

- [1] KONRAD E. A correction to the Douglas-Peucker line generalization algorithm [J]. *Comput & Geosci*, 2002, 28(8): 995-997
- [2] DE HALLEUX J. A C++ implementation of Douglas-Peucker Line Approximation Algorithm [M/OL]. (2003-03-04) [2004-06-08] <http://www.codeproject.com/cpp/dphull.asp>, html
- [3] 赵 瑜,周宇阳. 基于径向基神经网络的热力参数虚拟传感器 [J]. *汽轮机技术*, 2002, 44(6): 356-358

DPS algorithm of steam turbine virtual sensor data pre-process and its application

WANG Xiao fang^{* 1}, LI Gang¹, ZHOU Yu yang², CHEN Guang bing²

(1.School of Energy and Power Eng., Dalian Univ. of Technol., Dalian 116024, China;

2.Shanghai Xinhua Control Eng. Co., Ltd., Shanghai 200245, China)

Abstract Aiming at the problem of steam turbine virtual sensor data pre-process, a C program is obtained based on the principle of Douglas-Peucker line simplification (DPS) algorithm. For ensuring the applicability of algorithm, the simplified result of different typical line types in engineering is achieved and a chosen-data tolerance rule of affecting precision is put forward. An example of turbine virtual sensor data pre-process is shown, and the real-time simulation problem of virtual sensor is resolved available with DPS algorithm. It is concluded that DPS algorithm plays an important role in the data simplification in most of the engineering fields and that the tolerance rule put forward here is applicable for most line types in engineering.

Key words steam turbine; virtual sensor; data process; DPS algorithm