

神经网络状态监测和诊断的分布式实现

程巍¹, 陈前²

(1. 南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016;

2. 南京航空航天大学 航空宇航学院, 江苏 南京 210016)

摘要: 工业设备的大型化和工业过程的日趋复杂, 使得对工业过程的状态监测和故障诊断显得尤为重要。文中在多元统计过程控制方法的基础上, 将神经网络用于过程的监测, 并在基于 CORBA 技术的多客户端工业过程中实时监测软件系统实现。考虑工业系统故障预知的滞后, 软件设计中采用了特殊的复合神经网络结构以便于维护和拓展。FCCU 的实验数据表明: 系统能很好监测和判断过程运行状态, 并且能够在客户端图形化显示, 操作简单。

关键词: 神经网络; 状态监测; CORBA; 分布式技术

中图分类号: TP183

文献标识码: A

文章编号: 1673-629X(2006)07-0111-03

Implementation of Distributed Monitoring and Diagnosing with Neural Network

CHENG Wei¹, CHEN Qian²

(1. Coll. of Info. Sci. and Tech., Nanjing Univ. of Aeronautics and Astronautics, Nanjing 210016, China;

2. Coll. of Aerospace Eng., Nanjing Univ. of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: Status monitoring and fault diagnosis are the great importance in chemical engineering process, especially by now the equipment is becoming more and more complex. On the basis of the MSPC, this paper adopts the neural network for process monitoring and adds it to the multi-client distributed monitoring system designed based on CORBA. Considered the lag of the fault detection, the design introduces a specific compound network architecture that makes the software convenient for maintainability and extension. The experiment data shows that the graphical results could well-reflect the status of the FCCU system.

Key words: neural network; status monitoring; CORBA; distributed technique

0 引言

近年来,随着科学技术的进步和经济的发展,为了满足人类日益增长的物质需求,实现更高的生产率,工程系统正朝着大型化、复杂化的方向发展^[1]。这使得新一代的工业生产过程系统日益庞大和复杂,难以维护。在实际生产过程中往往微小子系统的故障将导致整个生产线的停产,给企业或社会带来重大损失。因此,结合各种先进的现代化的监控和故障诊断技术构建可复用的监测和诊断系统显的尤为重要。

如今社会上投入使用的监控系统大多针对某一种工业设备或运作过程,过于局限、可扩展性差、缺乏统一规划、开发费用高、维护费高、更不便于系统集成;而且跨平台性差,系统对运行环境、操作平台的依赖性强,移植困难。

分布式多元统计过程控制(DMSPC)在设计过程中吸取上述的缺点与不足,是基于公用对象请求代理体系(CORBA)技术而开发一种功能强大、图形界面下操作直观且用户界面统一友好的多客户端监控和诊断分布式系统。文中就是在多元统计的基础上^[2]将神经网络集成到 DMSPC 系统,扩展系统的监测方法,增加其诊断功能。

1 系统建模方法

1.1 人工神经网络

人工神经网络是人工智能的一个分支,它用于设备故障诊断,起源于 20 世纪 80 年代末期^[3]。它的自学习能力克服了传统诊断方法知识获取的瓶颈,并且其强大的并行计算能力和自学习功能及联想功能,很适合用于故障监测与识别。

目前,人工神经网络在故障诊断领域的应用主要为两个方面:

一方面为对系统建模,预测系统的输出,从而对整个系统进行监测。人工神经网络具有强大的函数拟合功能,在给出一定的训练数据的情况下,通过自身的自学习能

收稿日期:2005-10-05

作者简介:程巍(1982-),男,江苏南京人,硕士研究生,研究方向为软件设计、故障诊断;陈前,教授、博导,研究方向为工程动力学系统参数识别、振动控制,大系统状态监测与故障诊断等。

力,能有效地拟合出系统模型。其一般诊断过程是:先获得诊断对象在正常状态下的输入输出数据,对人工神经网络进行训练,建立系统在正常状态下的模型;然后即可在给定输入数据的情况下,通过训练好的人工神经网络模型预测出系统的输出状态,从而判断系统的实际工作状态,达到状态监测的目的。

另一方面则是对故障进行诊断分类。故障诊断过程实际上就是一个模式识别的过程,即实现从故障征兆空间到故障空间的映射。人工神经网络的自学习、联想记忆等能力,使其特别适合于故障的识别分类。诊断过程一般为:先获得诊断对象的训练数据,其包括两组数据,分别是输入数据和目标输出数据;然后对人工神经网络进行训练,得到输入输出的关系。诊断时只需将待诊断数据输入网络,即可得到诊断结果。

综上所述,基于神经网络的特点,它比传统的状态监测方法更加适合于现代工业大系统;同时,结合多元统计过程控制方法可以对工业大系统实施更完整的状态监测并随之进行诊断。但神经网络用于故障识别时,由于工业生产过程的复杂的动力学特性和设备的高耦合性,在软件设计过程中不可能考虑到所有故障状态,需要不断增加和完善。因此,在诊断的设计过程中,参考案例库的设计模式,对每一种故障单独进行训练建模,当增加故障模式时,只需根据其输入输出数据特性增加一次训练而无需增加系统代码,从而大大提高了系统的扩展性。具体结构如图 1 所示。

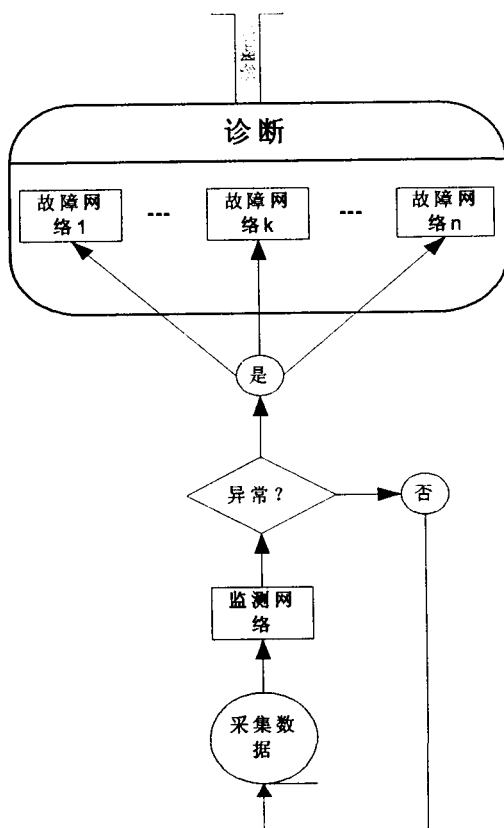


图 1 神经网络监测-诊断结构

1.2 核函数概率密度估计 KDE

一般的概率估计法总是需要已知或假定总体的概率分布函数而只是某些具体的参数值未知,如常常假设数据服从正态分布,然后对数据样本进行分析而估计出参数值。而工业生产过程中或实验模型的输出数据往往无法预知数据的总体分布,在这方面传统的参数估计已经不能满足现代化的生产过程监控。核函数估计法 KDE 是非参数估计法,用来估计各采样点的联合概率密度函数,它可以从一定的数据样本中,不需要任何先验知识(或假定)确定数据的概率密度分布。近年 KDE 已被提出并在工程应用中得到了认可^[1]。

结合文中运用的单模式神经网络训练和 KDE 估计,将神经网络训练样本的输出数据用 KDE 分析统计特性,确定数据的概率密度分布与置信区间,通过对照密度分布(以图表形式给出)或置信区间,判断监测数据异常与否。

2 DMSPC 的通信结构

2.1 CORBA 核心概念

对象管理组织(OMG)致力于解决分布式应用程序的复杂性与高成本问题。CORBA 顺应软件技术发展的潮流,成功地融合了两种技术:一是基于消息传递的分布式客户机/服务器技术;二是面向对象软件技术。CORBA 采用面向对象方法创建在应用程序之间可重用与可共享的软件组件,每一个对象对外隐藏了它内部工作的细节,并提供一个定义良好的外部接口,从而降低应用程序的复杂性。一旦实现并测试一个对象后,它可以在新应用环境中多次被重用,因而减少了开发应用程序的成本。

IDL 和 GIOP 是 CORBA 的两个非常重要的抽象概念。IDL 独立于具体编程语言和具体平台,是一个纯规范,不涉及实现。GIOP 使 CORBA 获得了对于网络协议的无关性,在实现时能映射到具体的传输层协议,如在因特网 ORB 互联协议就是 GIOP 在 TCP/IP 上的映像。这使得 CORBA 可以兼容多种网络环境,对于保证 CORBA 产品之间的互操作意义十分重大^[4]。

2.2 系统通信体系结构

正是 CORBA 的平台无关性很适合用于集成企业内的异类计算机系统,在设计 DMSPC 时,采用了基于 CORBA 的分布式设计。

考虑服务器承载实时数据采集任务、大量的数据计算分析和避免多客户的操作平台冲突,DMSPC 系统在服务计算端和客户端分别采用 C++ 和 Java 语言实现,通过定义的 IDL 接口文件,具体实现采用 Inprise 公司的 VisiBroker 作为 CORBA 工具,客户端与服务端的通信用 VisiBroker 的智能代理(smart agent)来实现。Osagent 是一种动态分布式目录服务,客户程序调用对象实现方法时,由智能代理自动查找指定的实现,从而建立客户程序与实现之间的连接^[5]。DMSPC 的具体通信结果如图 2 所示。

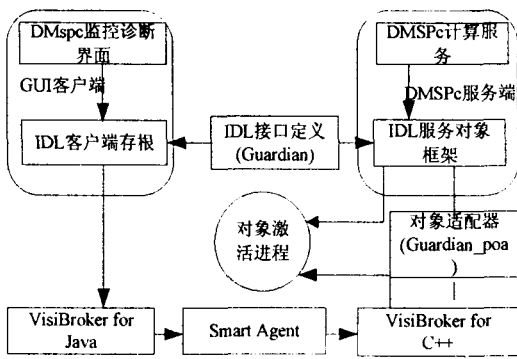


图 2 DMSPC 通信结构

3 DMSPC 流程

DMSPC 的服务器负责工业过程的生产数据的采集和处理,处理结果传到客户端,客户端将处理结果以图形化的形式呈现给操作人员,具体流程如下:

- 1) 数据采集;
- 2) 数据预处理;
- 3) 建模。用户通过客户端界面选择多元计算方法或神经网络方法在服务器端建立分析模型;
- 4) 监测。实现离线/在线监测,用户可以通过系统运行采集的数据在线监测,也可以利用预先采集的数据进行判别过程的运转状态;
- 5) 诊断。若监测发现异常,发出诊断指令,服务器给出处理结果。

4 实验测试及显示

4.1 实验数据来源和参数设定

实验主要就化学工业中流体催化裂化(Fluid Catalytic Cracking)炼油过程的仿真系统——FCCU 的产生数据进行分析。关于 FCCU 的具体参数指标可参考文献[6]。

该系统可模拟产生 36 个输出,引入 7 类故障,根据仿真系统的输出特性,实验时选择其中的 10 个变量作为网络的输入,分别是:提升管的温度、主蒸馏塔压力、湿气压缩机的吸入流量、湿气进入气体恢复器时的流速、再生床的温度、烟气道处温度、集气室中烟气的 O₂ 含量、集气室中烟气的 CO 含量、竖管中催化剂的水平位置和湿气压压缩机入口阀门的位置;6 个变量作为网络的输出,分别为:湿气的流速、新原料的流速、鼓风机空气流速、助燃鼓风机排气压、提升空气的实际速度、再生器气体含量。监测网络为 BP 网络:输入层节点 10、输出层节点 6、一个隐含层设定单元数 9、误差训练限 1E-4、训练样本数 1000。网络训练结束分别选取正常数据、随机抽取的故障 3 数据来测试服务器端建立的监测网络。

4.2 可视化图表显示

为了让监控人员摆脱繁芜的数据,能直观地了解系统的运行情况和准确地判断工业系统的运行状态,客户端采用多种监测指标图形:输入变量变换图、输出变量均方预

测误差(SPE, Squared Prediction Error)图、输出变量联合概率分布图(PDF)和输出变量的联合概率等高线(Contour)图辅助形象显示。

SPE 统计图是将训练数据的网络输出与标准输出做均方误差。若系统正常则对输出变量的预测误差应稳定在某一水平,否则预测误差将超出预警限。如图 3 为正常数据和故障 3 测试数据的输入变量和 SPE 统计图,输入变量图的上下限给出了经过变换后正常数据的分布区间, SPE 的上下限分别是 99% 和 95% 的置信限,由图看出的故障 3 数据的 SPE 统计量大部分都超出了 99% 和 95% 的置信限,输入变量也超出了正常数据的分布区间。

联合概率分布(PDF)图,是 KDE 计算网络输出数据的联合概率值为纵坐标的时间序列图,在该图上同时画出系统的安全控制限 95% 和 99%,图 4 为故障 3 的 PDF 图。故障 3 的 PDF 点几乎都落在了 99% 限外,因此探测出异常状态。

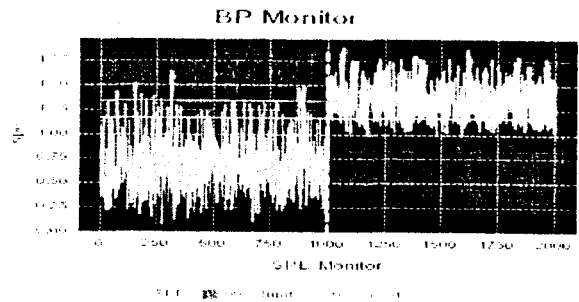


图 3 故障 3 SPE,左 1000 点为正常数据,右为故障数据

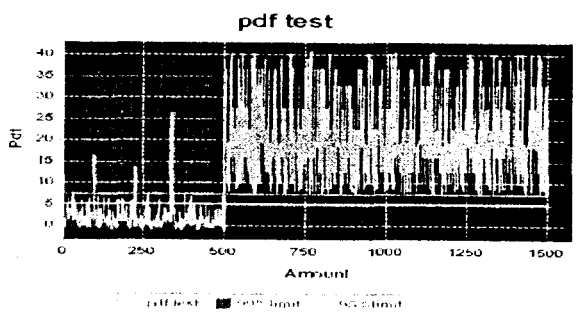


图 4 故障 3 PDF,左 500 点为正常数据

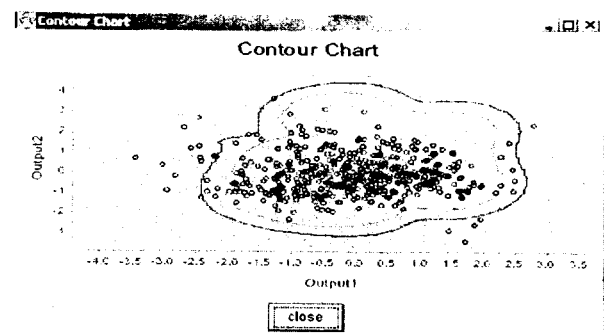


图 5 500 点正常数据的 Contour 图

(注:上述图形均采用 JfreeChart 组件绘制而成。)

联合概率等高图则是将观测值的时间序列和置信限

(下转第 179 页)

test 文件,到这里,在 μ CLinux 中添加设备驱动程序的工作可以说是完成了,你就可以使用自己的新设备 test 了。

4 设备驱动机制的关键问题

4.1 内存操作

在设备驱动程序中动态开辟内存,不是用 malloc 函数,而是用 kmalloc 函数。这个函数运行很快,除非它被阻塞。或者用 get-free-pages 直接申请页。释放内存用的是 kfree 或 free-pages。

4.2 中断处理

设备驱动程序通过调用 request_irq() 函数申请中断,将一个硬件处理函数挂到相应的处理队列中^[4]。 μ CLinux 系统中对中断的处理属于系统的核心部分,因此,如果外部设备与系统之间以中断方式进行数据交换,就必须把该设备的驱动程序作为系统内核的一部分。通过 request_irq 函数调用就可以把相关的中断号和具体的中断处理程序相关联。

```
Int request_irq(unsigned int irq,
Void (* handler)(int, void *, struct pt_regs *),
Unsigned int log flags,
Const char * device);
```

其中:handler 是中断处理子程序指针,中断产生时由操作系统自动调用此函数;参数 irq 为中断号;pt_regs 为中断发生时寄存器的内容;device 为设备名称;flags 确定了中断处理程序的一些特性。Flags = SA_INTERRUPT 表示该中断处理函数是快速中断,在其运行时所有中断都被屏蔽;不设置 SA_INTERRUPT 表示慢速处理程序,这种程序运行时除了正在处理的中断外,其他中断都没有被屏蔽;flags = SA_SHIRQ 表示共享此中断处理程序。函数调用成功返回 0 值,返回 -EINVAL 表示中断号超出范围或者 handler = NULL,返回 -BUSY 表示中断已经被占用且

不能共享。

4.3 用户空间和内核空间

设备驱动程序是在“内核空间”中运行的,而一般应用程序则是在“用户空间”中运行。在 μ CLinux 系统中,操作系统内核程序在最高级执行(也称为“管理员态”),在内核空间可以执行对硬件的任何操作^[5],而一般的应用程序则运行在最低级(“用户空间”)。在用户空间操作,系统禁止对硬件的直接访问和对内存的未授权访问。

μ CLinux 通过系统调用和硬件中断完成从用户空间到内核空间的控制转移。执行系统调用的内核代码在进程的上下文上执行(它代表调用进程操作,而且可以访问进程地址空间的数据)。但与此不同,处理中断的函数代码相对进程而言是异步的,而且与任何一个进程都无关。

5 结束语

文中通过介绍在 μ CLinux 操作系统下设备驱动的相关问题,使用户了解嵌入式系统设备驱动的开发流程。借助 μ CLinux 的开放性,能够灵活快捷开发嵌入式应用系统。

参考文献:

- [1] 邹思轶. 嵌入式 Linux 的设计与应用[M]. 北京:清华大学出版社,2002.
- [2] Rubini A, Corbet J. LINUX 设备驱动程序[M]. 魏永明, 骆刚, 姜君译. 北京:中国电力出版社,2002.
- [3] 贾明, 严世贤. Linux 下的 C 编程[M]. 北京:人民邮电出版社,2001.
- [4] 翟鸿鸣. Linux 系统实时性能增强方法的研究[J]. 微机发展, 2003, 13(S): 1-3.
- [5] 王俊锋, 冯志彪. 如何编写设备驱动程序[J]. 微机发展, 2005, 15(2): 25-27.

(上接第 113 页)

同时投影任意输出主元空间的某一个平面上而形成的二维等高图。Contour 图的优点直观,尤其是网络输出主元维数较低的时候,但随着主元空间维数的升高,全面反映系统分布的图将会越来越多而且精确性会受到影响。如主元空间为三维时需要 3 张图,四维时候需 6 张图。图 5 是正常数据联合概率密度的前两个输出主元的二维投影图,内外线分别为模型样本输出的 95% 和 99% 置信线,测试正常样本几乎都落在了 95% 置信区内。

5 结束语

目前分布式系统在当前各个领域应用广泛,它的介入既让这些领域取得了新的发展与变革,也给自己充实了新的概念与方法。文中将分布式和神经网络结合起来,将其应用到工业企业内部便于对工业设备的监测。考虑到系统的异构性,采用 CORBA 规范来实现,充分保证了系统

的跨平台性和服务端的实时性两大优点。从对 FCCU 数据的实验结果分析,软件具有一定的实用价值。

参考文献:

- [1] 陈前. 关于大系统的状态监测与故障诊断[J]. 振动、测试与诊断, 2002, 22(3): 163-174.
- [2] 刘道敏, 陈前. 多元统计过程状态监测分布式系统的实现[J]. 工业控制计算机, 2004(11): 21-23.
- [3] 虞和济, 陈长征, 张省, 等. 基于神经网络的智能诊断[M]. 北京:冶金工业出版社, 2000.
- [4] 朱三元, 钱乐秋, 宿为民. 软件工程技术概论[M]. 北京:科学出版社, 2002.
- [5] Borland/Inprise 公司. VisiBroker for C++ [M]. 北京:机械工业出版社, 2000.
- [6] McFarlane R C, Reineman R C, Bartee J F, et al. Dynamic Simulator for a Model IV Fluid Catalytic Cracking Unit[J]. Computers Chem Engng, 1993, 17(3): 275-300.