工况数据模拟生成系统①

严盈富[1], 户伟利[2], 祝中良[3]

[1](南昌航空大学 无损检测技术教育部重点实验室, 南昌 330063)
[2](南昌航空大学 信息工程学院, 南昌 330063)
[3](南昌航空大学 软件学院, 南昌 330063)

摘 要: 为模拟工业现场各种设备的工况数据, 供监测人员监测数据时使用, 提出构建工况数据模拟生成系统的需求, 并对两种工况数据模拟生成的算法进行了探讨, 将工况数据模拟生成算法从单模型无关系模式扩展到多模型有关系模式. 实际应用表明, 提出的工况数据模拟生成算法能有效的模拟工业现场各设备的工况数据.

关键词: 模拟生成; 算法; 工况数据; 模型; 设备

Simulating Generation System of Condition Data

YAN Ying-Fu[1], HU Wei-Li[2], ZHU Zhong-Liang[3]

[1](The Key Lab of Nondestructive Testing Technology of Ministry of Education, Nanchang Hangkong University, Nanchang 330063, China)
[2](School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China)
[3](School of Software, Nanchang Hangkong University, Nanchang 330063, China)

Abstract: In order to simulate the condition data of the various equipments in the industry locales and provides a data monitoring method for the monitoring personnel, the requirement for constructing a simulating generation system of the condition data is proposed, and two algorithms of the simulating generation of the condition data are discussed. The simulating generation algorithm of the condition data is extended from the single-model irrelevant pattern to the multi-model relevant pattern. The practical application shows that with the adoption of the simulating generation algorithm of the work condition data provided in the essay, the condition data of the various equipments in the industrial locales can be effectively simulated.

Key words: simulating-generation; algorithm; condition-data; model; equipment

1 引言

随着现代工业自动化水平的日益提高, 现代工业系统规模不断扩大, 系统各部分之间协作的复杂性迅速增加. 一旦工业系统中某部分发生故障, 将使整个系统无法正常工作, 造成巨大的停机损失. 因此, 对于大型复杂工业系统, 人们迫切希望能获取现场设备的实时工况数据. 通过这些数据的变化规律, 预测现场设备可能出现的各种故障, 提高系统的可靠性和稳定性[1]. 由此, 需要开发一种监测系统, 对现场设备的工况数据进行监测.

为了对监测系统接收数据的能力进行验证, 需要开发一种工况数据模拟生成系统(以下简称模拟系统). 模拟系统主要用于模拟生成大批量的工况数据, 并将这些数据封装后发送给监测系统[2]. 工况数据模拟生成算法是模拟系统的核心. 为提高模拟生成工况数据的有效性和准确性, 同时也为优化模拟系统性能, 提高系统运行效率, 本文对工况数据模拟生成的算法进行了研究, 这对复杂工业系统故障诊断和状态监测具有重要意义.

2 工况数据模拟生成算法设计

2.1 工况数据模型简述

工业现场各设备生成的各个工况数据都对应于一

① 收稿时间:2013-06-06;收到修改稿时间:2013-07-16

定的时间点，例如设备在启动过程中生成对应设备启动过程的工况数据，设备在停止过程中生成对应设备停止过程的工况数据. 因此，在设计工况数据模拟生成算法时，可以将模拟生成工况数据的模型设置成关于时间的一元表达式. 由于同类设备的相同工况数据值按时间轴生成的曲线可能存在趋势相同，但数据值不同的情况，因此对于同类设备的相同工况的工况数据模型，应该允许其带有不同的工况数据模型参数. 因此，工况的工况数据模型应该由模型表达式和模型参数两部分组成. 如图 1 所示.

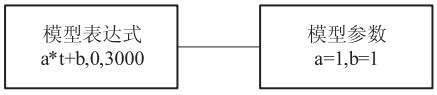

图 1 工况数据模型组成图

单个工况的工况数据模型可能是非分段数据模型，如图 1 所示，也可能是分段数据模型，如图 2 所示. 图 2 中的分段数据模型是一个关于时间 *t* 的分段函数，该模型共分为三段，其中第一段是一个余弦函数，其定义域为 0<*t*<=1000，参数 *a*=1；第二段是一个线性函数，其定义域为 1000<*t*<=2000，参数 *b*=0.3, *c*=1；第三段是一个正弦函数，其定义域为 2000<*t*<=3000，参数 *d*=1, *e*=0.5[3].

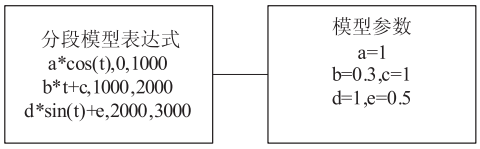

图 2 分段数据模型图

工况的模型表达式和模型参数构成一个字符串映射对，其中映射的键类型对应工况数据模型字符串，映射的值类型对应工况数据模型参数字符串. 以图 2 中的工况数据模型为例，该分段数据模型对应的映射共包含三个键值对，即键“*a*cos*(*t*),0,1000”，值“*a*=1”；键 “*b*t+c*,1000,2000”，值 “*b*=0.3,*c*=1”；键 “*d*sin*(*t*)+e, 2000,3000”，值“*d*=1,*e*=0.5”. 如图 3 所示. 因为分段模型各段的定义域不相同，所以该映射是一个单值映射，键和值之间是一一对应的关系[4].

2.2 数据库设计

为实现模拟生成各类设备的工况数据的功能，必须允许用户对工况的相关信息进行定制. 本文将

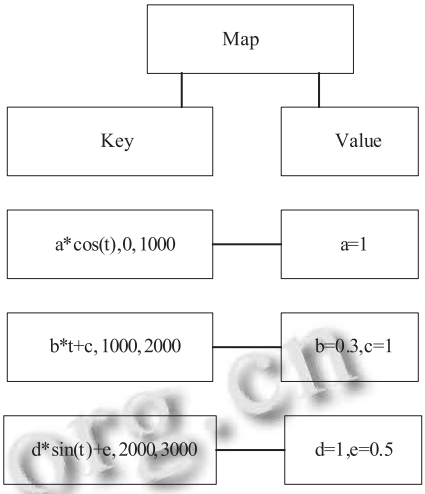

图 3 分段数据模型对应的 Map 键值对

这些信息称为工况数据字典. 为能完整的描述一个工况的相关信息，工况数据字典中应包含如下内容，工况 *ID*、设备号、工况类型、工况数据模型、工况的采样周期、采样时长、开始采样时间、工况值的上限和工况值的下限[5]. 下面分别对各个字段的意义进行解释.

工况 *ID*: 每个工况独有的 *ID* 值.

设备号: 设备的出厂序列号.

工况类型: 标明该工况所属的类型，如开关量、数字量和模拟量等. 其中开关量是指只有 0 和 1 两种状态的数值量；数字量是指在分段区间内数值恒定的量；模拟量是指数值以某一函数规律变化的数值量.

工况数据模型: 模拟生成工况数据时所采用的关于时间 t 的一元函数，包括表达式和参数两部分.

工况的采样周期: 对工况数据进行采样的周期.

采样时长: 对于某设备的某工况共需要采样多长时间.

开始采样时间: 开始对某工况进行采样的时间.

工况值的上限: 某些工况可能存在上限值，如泵车的液压油温度不能超过 800 摄氏度，否则视作异常.

工况值的下限: 某些工况可能存在下限值，如泵车的泵送次数工况不能小于 20 次/秒，否则视作异常.

为简化问题，本文采用归纳法来得出工况数据模拟生成系统的一般性数据库设计方法. 文章先讨论了一种简单的单模型无关系模式的数据库设计方法，然后对这种方法进行扩展，提出了一种多模型有关系模式的数据库设计方法.

2.2.1 单模型无关系模式的数据库设计

单模型无关系模式的工况数据模拟生成算法中只存在工况数据模型，不存在其他的数据模型，各工况之间也不存在互斥和依存等各种关系，因此是一种比较简单的工况数据模拟生成算法. 该算法从工况的开始采样时间开始以指定的采样周期生成数据，到工况的定义域上限或者采样时长结束生成工况数据. 算法共涉及三张数据表，如表 1~表 3 所示[6].

表 1　设备工况表

工况 ID	设备类型	设备号	工况名称	工况类型
bc001	泵车	12BC000001	泵送次数	模拟量
bc002	泵车	12BC000001	总泵送方量	模拟量

表 2　工况信息表

工况 ID	设备号	工况数据模型 ID	采样周期[秒]	开始采样时间	采样时长[小时]	工况值下限	工况值上限
bc001	12BC000001	1	5	2011-5-30 08:00:00	8	20	
bc002	12BC000001	2	5	2011-5-30 08:00:00	10		

表 3　工况数据模型表

工况数据模型 ID	模型表达式	模型参数
1	a*cos(t),0,1000 b*t+c,1000,2000 d*sin(t)+e,2000,3000	a=1 b=0.3,c=1 d=1,e=0.5
2	a*t+b,0,3000	a=1,b=1

单模型无关系模式的工况数据模拟生成算法只包含工况数据模型，没有包含工况的其他模型，而且没有描绘出该工况与其他工况之间的关系. 在现实的工业现场环境下，设备所处的环境是复杂多变的，各台设备之间，机器内部的各个零部件之间，机器产生的各个工况之间都有可能相互影响. 因此，单模型无关系模式的工况数据模拟生成算法具有一定的局限性[7].

2.2.2 多模型有关系模式的数据库设计

多模型有关系模式的工况数据模拟生成算法在单模型无关系模式的算法上进行了扩展，在工况数据字典中不但定义了工况数据模型，还定义了工况噪声模型、衰变模型和关系模型.

因为工业现场设备在运行的过程中可能会受到各种噪声干扰，所以引入噪声模型进行描述. 设备运行一段时间以后，一般是 4 个月至半年，零部件开始发生磨损，性能开始下降，因此引入衰变模型对设备的性能下降进行描述. 现场设备的各个工况之间一般不都是相互独立的，其中可能有些工况存在相互关系，如互斥关系和依存关系等. 下面分别对各种模型进行介绍.

(1) 数据模型：数据模型是关于时间的一维表达式. 它主要模拟现场设备正常运行时各工况的数据值.

(2) 噪声模型：噪声模型可以是关于时间的一维变元，也可以是随机的高斯噪声或椒盐噪声. 它主要模拟工作环境中噪声对设备工况数值造成的干扰[8].

(3) 衰变模型：衰变模型是关于时间的一元表达式. 它主要模拟当设备运行一段时间后，部分零部件发生磨损，这时工况数值与正常值比较时发生改变的数值.

(4) 关系模型：关系模型是该工况关于另一工况的一元表达式. 它主要用于模拟两个工况之间存在关系时，其中一个工况的数值. 假设工况 *A* 和工况 *B* 之间存在关系，则它们可能存在互斥和依存两种关系. 其中互斥关系是指工况 *A* 存在时，工况 *B* 不能同时存在；依存关系是指工况 *A* 数值的计算需要依赖于工况 *B* 的数值. 其中依存关系还存在一种比较特殊的情况，即工况 *A* 数值的计算需要依赖工况 *B* 上次计算的数值. 用数学表达式表示为：$f(A)=a*f(B\text{-}1)+b$，其中 $f(A)$为工况 *A* 该次计算的数值，$f(B\text{-}1)$为工况 *B* 上次计算的数值[9].

多模型有关系模式的数据库设计与单模型无关系模式的数据库设计相似，只是在工况信息表中加入了噪声模型 *ID*、衰变模型 *ID*、关系模型 *ID* 等字段，并增添了噪声模型表、衰变模型表、关系模型表.

下面列出有改变的表的各个字段. 其中衰变模型表的各个字段与数据模型表的各个字段相似，不再单

独列出.

表 4 扩展后的工况信息表

工况 ID	设备号	数据模型 ID	噪声模型 ID	衰变模型 ID	关系模型 ID	采样周期 [秒]	开始采样时间	采样时长 [小时]	工况值下限	工况值上限
bc001	12BC000001	1	1	1	0	5	2011-5-30 08:00:00	8	20	
bc002	12BC000001	2	2	2	1	5	2011-5-30 08:00:00	10		

表 5 噪声模型表

噪声模型 ID	噪声模型	噪声模型参数
1	a*t+b,0,3000	a=1,b=1
2	高斯噪声	a=1,b=1

表 6 关系模型表

关系模型 ID	模型表达式	模型参数	工况 1 ID	工况 2 ID
1	a*z +b,0,3000	a=1,b=1	bc001	bc002

在工况关系模型表中, 工况 1 *ID* 表示当前要计算的工况的 *ID*, 工况 2 *ID* 表示工况 1 所关联的工况 ID(这种关联关系包括互斥和依存等). 表 6 中工况 1 和工况 2 是依存关系, 且工况 *bc*001 的计算数值依赖于工况 *bc*002 上一次计算的数值, 表达式为 $f(bc001)=a*f(bc002\text{-}1)+b$, 其中 $f(bc001)$表示工况 *bc*001 该次计算的数值, $f(bc002\text{-}1)$表示工况 *bc*002 上次计算的数值. 标志变量 *z* 代表所依存工况上次计算的数值.

工况间的依存关系还存在另一种模型, 即工况 1 当前的计算数值依赖于工况 2 本次计算的数值, 例如 $f(bc001)=a*f(bc002)+b$, 其中 $f(bc001)$表示工况 *bc*001 该次计算的数值, $f(bc002)$表示工况 *bc*002 该次计算的数值, *a*、*b* 为模型中的参数. 当遇到以上情况时, 使用 *x* 表示被依赖工况该次计算的数值. 此时的工况关系模型表达式为 *a***x*+*b*.

表 5 工况噪声模型表中 *ID* 为 2 的噪声模型为高斯模型, 高斯模型存在两个参数 *a* 和 *b*, 其中 *a* 是高斯模型的斜率, *b* 是高斯模型的截矩. 在计算工况的高斯噪声时, 先使用 java 语言自带的高斯噪声函数生成一个浮点型噪声数据 *data*, 再根据表达式 *a***data*+*b* 计算出高斯噪声并返回.

多模型有关系模式的工况数据模拟生成算法弥补了单模型无关系模式算法的不足, 在原算法的基础上引入了噪声模型、衰变模型和关系模型, 特别是关系模型的引入, 使得该算法更具有普遍性, 能更准确的模拟工业现场设备的实时工况数据.

2.3 算法设计

2.3.1 单模型无关系模式的模拟算法

单模型无关系模式的工况数据模拟生成算法主要包含以下步骤.

① 根据用户选择的要模拟生成数据的工况 *ID* 和设备号, 查询表 1 设备工况表和表 2 工况信息表. 得出工况的相关信息, 主要包括工况的类型和数据模型 *ID* 以及工况的开始采样时间、采样时长等. 以生成设备 12*BC*000001 的 *bc*001 工况为例, 查询表 1 和表 2 得, 工况 *bc*001 的工况类型为模拟量, 工况数据模型 *ID* 为 1, 工况开始采样时间为 2011-5-30 08:00:00, 采样时长为 8 小时, 采样周期为 5 秒, 工况值下限为 20.

② 根据①中查询到的工况数据模型 *ID*, 继续查询表 3 工况数据模型表, 得到工况数据模型. 以设备 12*BC*000001 的 *bc*001 工况数据生成为例, 在①中已查得该工况的数据模型 *ID* 为 1, 于是查询表 3, 得到 *bc*001 的工况数据模型为 *a***cos*(*t*),0,1000; *b***t*+*c*,1000, 2000; *d***sin*(*t*)+*e*,2000,3000, 数据模型参数为 *a*=1; *b*=0.3,*c*=1; *d*=1,*e*=0.5(各分段模型和模型参数间以分号分隔).

③ 根据②中查询到的模型及①中的工况相关信息, 生成工况数据, 并存储到数据库[10].

工况数据模拟生成的过程如以下 java 伪代码.

```
for(int t>=工况开始时间;t<=工况开始时间+数据模型定义域上限;t+=周期){
        结果=Expression(t);
        验证上下限;
        保存到数据库;
}
```

其中 *Expression* 表示③中查询到的工况数据模型表达式. 这里使用一个 *java for* 循环生成工况数据. 例如对于设备 12*BC*000001 的 *bc*001 工况数据生成过程, 其对应伪代码为:

```
for(int  t>=0;t<=3000;t+=5){
        结果=使用 JEP 解析表达式 a*cos(t),0,1000; b*t+c,1000,2000; d*sin(t)+e,2000,3000. 参数为 a=1; b=0.3,c=1; d=1,e=0.5;
        if(结果>=20){
            保存到数据库;
        }
}
```

其中 *JEP*(*Java Math Expression Praser*)*Java* 表达式解析器是一个开源的数学表达式解析库. 通过使用 *JEP*, 用户可以以字符串的形式输入一个任意的公式, 然后快速的计算出结果. *JEP* 允许用户为函数指定参数.

但现有的 *JEP* 表达式解析库只支持非分段表达式解析, 不支持分段表达式解析, 本文将讨论一种使用 *JEP* 解析分段表达式的方法.

使用 *JEP* 解析分段表达式的步骤如下.

① 从工况数据模型表和工况数据模型参数表中取出工况数据模型和模型参数. 以设备 12*BC*000001 的 *bc*001 工况数据模型解析为例, 其工况数据模型为 *a***cos*(*t*),0,1000; *b***t*+*c*,1000,2000; *d***sin*(*t*)+*e*,2000,3000, 模型参数为 *a*=1; *b*=0.3,*c*=1; *d*=1,*e*=0.5.

② 将①中取出的工况数据模型和模型参数分段存储到 *map* 集合中. 以设备 12*BC*000001 的 *bc*001 工况数据模型解析为例, 需要先构造一个 *map* 集合, 然后对 *map* 集合中的键值对进行设置. *java* 代码格式如下:

```
java.util.Map<String,String>
map=new     HashMap<String,String>();
        map.put("a*cos(t),0,1000", "a=1");
        map.put("b*t+c,1000,2000", "b=0.3,c=1");
        map.put("d*sin(t)+e,2000,3000","d=1,e=0.5");
```

设置以后 *map* 中的键值对如图 3 所示[11].

③ 生成数据时, 在 *for* 循环中会传入一个当前生成时间 *t* 的数值, 将 *t* 值与②中 *map* 集合中分段函数各段的定义域进行比较, 看 *t* 值是否在某段的定义域中, 如果在定义域中, 则按该段的数据模型生成工况数据; 如果对 *map* 遍历完成之后, 仍未找到 *t* 所在的定义域, 则返回一个极大的负数, 例如-LONGMAX(长整型的最大值), 表示未找到 *t* 所在的定义域, 无法生成工况数据. 以设备 12*BC*000001 的 *bc*001 工况数据模拟生成为例, 假设当前生成时间 *t* 为 1500s, 则对②中的 *map* 集合进行遍历, 发现 1000<1500<2000, 在分段函数 *b***t*+*c* 段的定义域内, 然后调用 *JEP* 对该段进行解析, 生成工况数据. 这样就将解析分段表达式的问题转化为解析非分段表达式的问题.

2.3.2 多模型有关系模式的模拟算法

多模型有关系模式的工况数据模拟生成算法与单模型无关系模式的工况数据模拟生成算法类似, 但对工况数据模型进行了扩展, 能更准确的模拟工业现场各设备的工况数据. 该算法在原工况数据模型的基础上增加了工况噪声模型、工况衰变模型和工况关系模型. 各模型的优先级如图 4 所示.

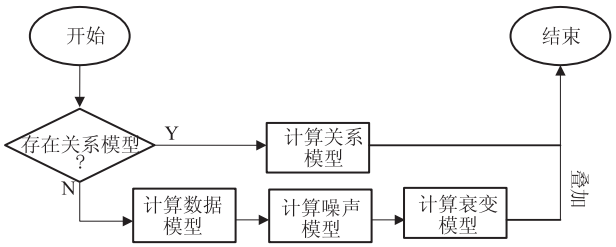

图 4　工况各模型优先级图

在图 4 中, 生成工况数据时, 先检查有无关系模型, 若存在关系模型, 则按关系模型计算后直接返回; 若不存在关系模型, 则分别计算数据模型、噪声模型和衰变模型再叠加后返回. 各模型的定义见 2.2.2 节.

下面以设备 12*BC*000001 的 *bc*002 工况数据生成为例, 介绍多模型有关系的工况数据模拟生成过程.

① 查询表 1 设备工况表和表 4 扩展后的工况信息表, 得出设备 12*BC*000001 的 *bc*002 工况的相关信息. 这些信息包括, 工况 *bc*002 的数据模型 *ID* 为 2、噪声模型 *ID* 为 2、衰变模型 *ID* 为 2、关系模型 *ID* 为 1, 工况的开始采样时间为 2011-5-30 08:00:00, 采样时长为 10 小时, 采样周期为 5 秒.

② 根据①中查询到的各模型及参数的 *ID* 信息, 查询设备 12*BC*000001 的 *bc*002 工况的各模型及参数表, 包括工况数据模型表、数据模型参数表、工况噪声模型表、噪声模型参数表、工况衰变模型表、工况衰变模型参数表、工况关系模型表和工况关系模型参数表. 得到 bc002 工况的各模型及模型参数信息.

③ 根据②中查询到的各模型及参数信息，按各模型的优先级进行计算，返回工况数据值. 例如对于设备 12*BC*000001 的 *bc*002 工况，首先检查发现其存在 *ID* 为 1 的关系模型，在关系模型表中查询得到其关系模型为 *a***z*+*b*，且工况 2 *ID* 为 *bc*001，表示工况 *bc*002 的数值依赖于工况 *bc*001 上次生成的数值. 于是查询工况数据表中工况 *bc*001 上次生成的数值，假设为 *data*，然后使用 *JEP* 对 *a***data*+*b* 进行解析，生成工况数据；若工况数据表中还不存在工况 *bc*001 的工况数据值，则直接返回 *a**0+*b* 作为工况数据值.

2.4 实验结果分析

按表 1、表 2、表 3 定制工况 *bc*001 的工况数据模型，按表 1、表 4、表 5、表 6 定制工况 *bc*002 的工况数据模型，并生成设备 12*BC*000001 的 *bc*001 工况和 *bc*002 工况的工况数据曲线图如图 5 和图 6 所示. 这两种模型定制方式分别代表了单模型无关系模式的工况

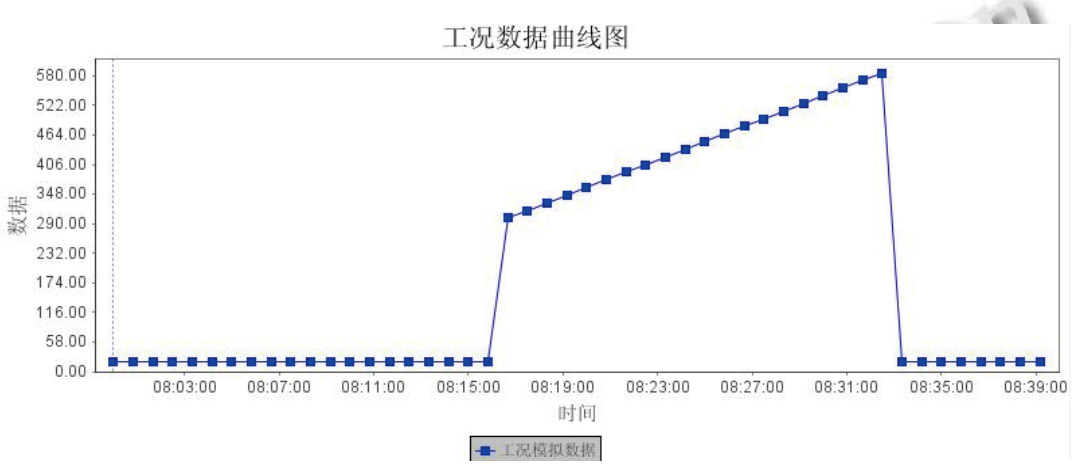

图 5 工况 *bc*001 曲线图

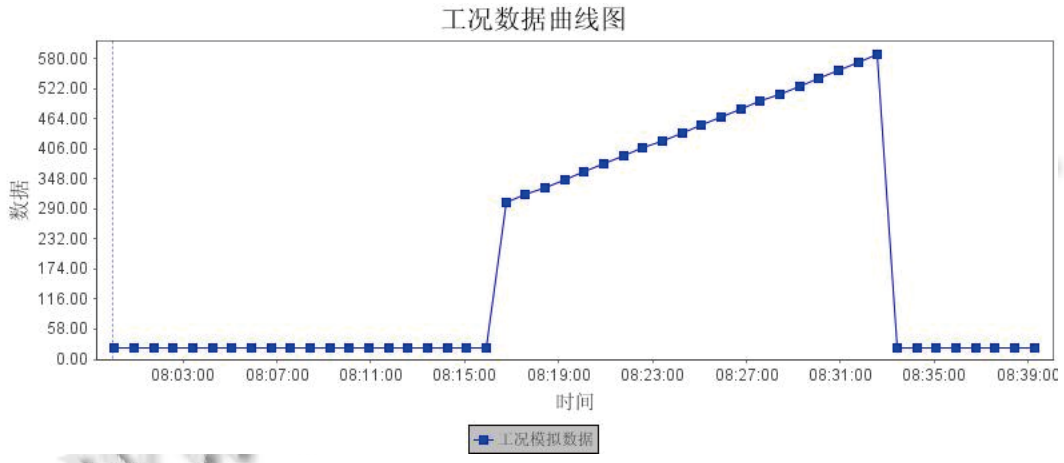

图 6 工况 *bc*002 曲线图

数据模拟生成方式和多模型有关系模式的工况数据模拟生成方式. 图中曲线使用 *JFreeChart* 图形库绘制. *JFreeChart* 是 *java* 平台上的一个开放的图表绘制类库，利用它可生成饼图(*pie charts*)、柱状图(*bar charts*)、散点图(*scatter plots*)、时序图(*time series*)、甘特图(*Gantt charts*)等多种图表[12].

由图中可以看出，工况 *bc*002 的工况数据值与工况 *bc*001 的工况数据值类似，只是在生成时间上比 *bc*001 滞后了一个周期(即 5s). 两者的数值差别很小，相差仅为 1，从图像上看不出来. 但是从工况 bc002 与工况 *bc*001 之间的关系模型可以看出两者数值上的关系，即 *f*(*z*)=*a***z*+*b*,0,3000，且 *a*=1, *b*=1. 其中 *z* 表示工况 bc001 上一次计算的数值，f(z)表示工况 bc002 该次计算的数值，即 bc002 该次计算的数值等于 bc001 上次计算的数值加 1. 这验证了系统生成数据的正确性.

用户还可以根据实际应用需要，定制任意设备的

任意工况数据模型, 这对复杂工业系统故障诊断和状态监测具有重要意义.

3 结语

本文主要论述了工况数据模拟生成系统的设计过程, 采用归纳法分析和比较了两种不同的工况数据模拟生成算法. 其中单模型无关系模式的工况数据模拟生成算法较简单, 且易于实现, 但其不能模拟现场设备各工况之间的关系, 也不能模拟环境中的噪声对工况数据模拟生成的影响, 还不能模拟设备运行一段时间后零部件发生衰变的状态, 具有很大的局限性. 多模型有关系模式的工况数据模拟生成算法弥补了单模型无关系模式的工况数据模拟生成算法的不足, 但由于逻辑较复杂, 造成了一定的效率损失.

参考文献

1 严新平,周强.机械系统工况监测与故障诊断.武汉:武汉理工大学出版社,2009.17–30.
2 宏宇.基于 FPGA 的风电监测系统数据采集单元设计[学位论文].北京:北京化工大学,2011.
3 宋扣生.分段函数解析.理科考试研究(高中版),2004,11(8): 8–10.
4 埃克尔. Java 编程思想.北京:机械工业出版社,2007.45–47.
5 赵铁栓,蔡应强.混凝土搅拌输送车液压驱动系统仿真.建筑机械,2005,(8).
6 刘丽珏,张龙祥.计算机软件.JDBC 与 Java 数据库程序设计.北京:人民邮电出版社,2001.
7 Geng Z, Chen J. Investigation into piston-slap-induced vibration for engine condition simulation and monitoring. Journal of Sound and Vibration, 2005, 282(3): 735–751.
8 山拜,达拉拜,黄玉划.几类非高斯噪声模型的转换研究.电子学报,2004,32(7):1090–1093.
9 贺尚红,杨昀梓.基于神经网络的混凝土泵车发动机万有特性建模与工况优化.中南大学学报(自然科学版),2010, 41(4).
10 黄虎,束鹏程,李志浩.风冷热泵冷热水机组结霜工况下工作过程动态仿真及实验验证.流体机械,2000,28(3):49–52.
11 李钟蔚,宋坤.Java 开发实战宝典.北京:清华大学出版社,2010.348–353.
12 Zhou P, Ye W. Application of JFreeChart in statistics and analysis of financial data. Journal of Chongqing Institute of Technology (Natural Science), 2008, 11: 38.

(上接第 143 页)

(3) 挂接具体驱动程序, 此项不需要修改.

通过以上 3 步即可完成 SiI3512 驱动程序的移植. 系统启动之后, 即可使用分区工具进行分区, 以便使用 SATA 硬盘存储多媒体数据.

4 结语

本文通过分析 Linux 2.4.15 内核的 IDE 驱动程序框架和其对应的源码, 摸索和总结出了一套在 Linux 下使用 IDE 框架来移植和使用 SATA 驱动程序的通用方法, 使得驱动开发者可以绕开浩瀚的驱动代码, 简单套用本文介绍的步骤, 即可完成 SATA 驱动程序的移植, 同时也对其它相关的驱动程序移植有一定的参考作用.

参考文献

1 毛德操,胡希明.LINUX 内核源代码情景分析(上、下).杭州:浙江大学出版社,2001.
2 linux 2.4.15、linux 2.4.23 内核源代码.https://www.kernel.org/pub/linux/kernel/v2.4/
3 Silicon Image. Inc.; SiI3512 Data Sheet, 2007.
4 Corbet J, Rubini A, Greg Kroah-Hartman Linux. 设备驱动程序.北京:中国电力出版社,2006.