

系统可靠性数字仿真系统 SRDS

杜小平 (西南石油学院机械工程系)

摘要: 本文建立了系统可靠性这一随机、离散系统的仿真模型,开发了系统可靠性数字仿真系统 SRDS,介绍了该系统的组成原理,着重讨论了系统开发中采用的若干技术。

一、系统建立背景

可靠性技术是一门对国民经济发展起着重要作用的学科。西方工业化国家通过几十年在可靠性基础研究和推广运用上不懈的努力,取得了巨大的成就。主要表现在产品质量的提高,开发成本的下降,产品竞争力的提高等方面。我国可靠性技术的研究与推广工作已在许多行业逐步开展,可靠性技术成为了目前最为活跃的学科之一。

系统可靠性分析的一个基本任务就是根据系统组成元件的可靠性水平,推断系统的可靠性水平。例如,对象航天飞机、人造卫星这样由成千上万个元件组成的大系统进行整体试验而得到系统可靠性指标是困难或不可能的,只能在弄清元件可靠性水平的基础上,依据一定的理论方法或工具手段得到系统的可靠性数量水平。对于一般工程系统而言,大多属于可修复系统(不可修系统仅是一个特例),其特点是系统逻辑关系复杂,系统由众多元件组成,元件以复杂的方式组成系统;属于随机性问题,元件的寿命与维修时间完全是随机的;系统的逻辑关系多于数学关系,用现有的随机过程等理论难以建立便于求解的数模型。这些特点对解析法来说是难以解决的,而恰恰又是计算机系统仿真所擅长的。因此,用计算机仿真法进行系统可靠性分析是必要和可行的。

本文在解决以往普通微机上难以问津的复杂可修复系统可靠性数字仿真上进行了尝试,开发了系统可靠性数字仿真系统 SRDS(System Reliability Digital Simulation)。该系统适合于计算机系统、机械系统、电子系统等可靠性分析。给定系统的结构与各元件的寿命、维修时间分布即可经仿真运行得到系统的各个可靠

性特征量。

二、系统组成

1.功能

SRDS 包含 6 个功能模块,如图 1 所示。

(1)用户登录:建立或进入用户子目录,用户只能在各自的目录下工作。

(2)文件编辑:编辑元件说明文件,该文件包含元件的分布函数与维修时间分布函数;编辑系统结构文件,说明元件是以何种方式(串联、并联、串并联等)组成系统。上述两文件是仿真运行所需的数据文件。

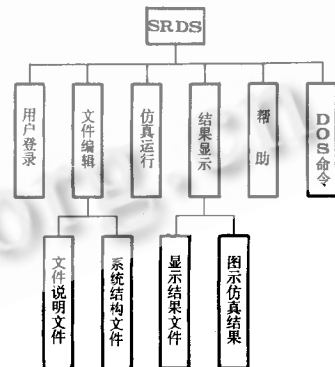


图 1

(3)仿真运行:系统进行仿真运行,产生结果文件。

(4)结果显示:调用文本文件阅读器显示三个结果文件:仿真结果;元件修复——故障——修复时间序列;系统修复——故障——修复时间序列。是否产生后两个文件可由用户确定。所有仿真结果均可图形显示,并可存盘供以后调用。

(5)帮助信息:显示系统使用说明。

(6)DOS 命令:在不退出系统的情况下执行 DOS 命令。

2.运行环境

硬件环境: IBM PC / XT, 286, 386 及其兼容机, VGA 显示器。

软件环境: DOS3.0 以上版本。

3.系统特点

(1)全汉字拉式菜单驱动,层层引导,步步提示。

(2)形象直观,仿真运行时同时显示仿真运行次数及蓝色长条背景上不断伸长的红色长条以提示用户系统运行的进度。

(3)设置登录与口令系统,用户限于本目录下工作。

(4)联机帮助。

(5)输入数据的容错功能,设轩输入数据默认值,自动检测数据的合法性并提示。

三、系统设计

1.仿真模型的建立

系统可靠性数字仿真是用适当的数学模型代替实际的系统,在数字计算机上对系统的可靠性特征进行实验和研究,分析系统的可靠性状态,得到系统可靠性的数量指标。系统可靠性仿真属于动态系统、随机系统、离散系统仿真。

实施仿真的第一步是建立适当的仿真模型,模型建好后还须不断地修改、完善与确认。

引入元件 i 的状态变量 X_i , 取

$$X_i(t) = \begin{cases} 1 & \text{在时刻 } t \text{ 第 } i \text{ 个元件处于故障状态} \\ 0 & \text{在时刻 } t \text{ 第 } i \text{ 个元件处于修复状态} \end{cases}$$

令 x_i 构成的向量为 $X(t)$, 即

$$X(t) = [x_1(t), x_2(t), \dots, x_n(t)]$$

用 $\Phi(t)$ 是 $X(t)$ 的函数, 即

$$\Phi(t) = \begin{cases} 1 & \text{在时刻 } t \text{ 系统处于故障状态} \\ 0 & \text{在时刻 } t \text{ 系统处于修复状态} \end{cases}$$

显然 $\Phi(t)$ 是 $X(t)$ 的函数, 即

$$\Phi(t) = \Phi[X(t)]$$

上式即为系统的结构函数, 即仿真模型。

2.仿真时钟机制的选取

在元件级仿真时, 仿真时钟应以事件预计发生的时刻, 以不等距时间间隔向前推进, 即仿真时钟每次跳跃推进到下一事件的发生时刻, 故这时宜用面向事件的仿真

时钟机制。根据元件寿命分布函数 $F(x)$ 抽样可得寿命 X , 根据元件维修时间分布函数 $G(y)$ 抽样可得到维修时间 Y , $F(x)$ 与 $G(y)$ 交替抽样得到元件修复——故障——修复时间序列。

在得到系统每一元件的修复——故障——修复时间序列后, 即可将所有元件的时间序列混合起来, 从小到大排序, 形成一个新的时间序列, 判断时间序列每一点上各元件所处的状态, 然后代入系统的结构函数得到系统的状态, 最终得到系统的修复——故障——修复时间序列。上述步骤应重复 N 次修复——故障——修复时间序列。

接下来即可进行统计计算。这时的仿真时钟应转向面向时间间隔。确定一个适当的时间间隔, 仿真时钟按这一时间间隔等距推进, 每次推进都扫描所有活动, 并作相应记录, 以作为统计计算的依据。以系统有效度为例 (有效度是系统在某一时刻处于完好状态的概率), 在仿真时钟推进时, 记录区间端点上系统在 N 次仿真中处于完好状态的次数 n , 则有效度就等于 n/N 。

四、研制技术

SRDS 系统采用 C 语言编写。为保证在微机有限的内存空间中顺利实现系统可靠性数字仿真, 提供良好的人机界面, 提高软件的可靠性等, 采取了以下的研制技术。

1.限制仿真运行占用的内存量

对于可修复系统来说, 仅仅一次仿真就要占用大量内存。例如, 若系统由 500 个元件组成, 每个元件的修复——故障——修复时间序列平均有 500 个点, 则 1 次仿真就需记录 $500 \times 500 = 250000$ 个点。按一个浮点数占 2 个字节计算, 光记录这些时间点就需约 488k 的内存空间, 更不用说其它内存开销(如各时间点的状态数组, 人机界面等)以及操作系统所占内存, 故解题规模较大时, 在微机有限的空间上仿真就进行不下去。为此, 在算法设计与程序设想上均需采取一定的技巧。

在算法上, 将两种仿真时钟机制混合使用, 即不是在得到 N 次系统修复——故障——修复时间序列后才进行统计处理, 而是利用面向事件的时钟推移得到一次系统修复——故障——修复时间序列后才进行统计处理, 而是利用面向事件的时钟推移得到一次系统修复——故

障——时间序列后,立即将时钟机制转换为面向时间间隔,进行统计记数,累积有关参数,待N次仿真结束时,再根据累积参数计算系统可靠性参数。当给定最大仿真时间很长时,元件的修复——故障——修复时间序列也将很长,因而需记录较多的时间点,这时可将最大仿真时间分成若干段,然后在每一段中进行仿真运行。另外,还采取了一些其它算法。采用这些算法尽管计算复杂程度大为增加,时间开销加大,但保证了仿真运行所占的空间不会突破微机的内存容量限制,使得在普通微机上得以实现大型复杂系统的可靠性数字仿真。

在程序设计上,主要采用了动态存储技术。在仿真运行过程中要产生许多中间结果,这些结果若以数组描述,则数组的长度是随机变化的。如果将其定义为静态数组,则由于在编译时就预定了其大小且又存储于内存的栈空间中,有可能有时会造成内存资源的浪费,有时又会限制解题的规模。故在程序运行时,根据运行情况,为数组申请动态空间,将数组存放于内存堆空间中去,充分利用机器的整个剩余空间。待这些数组用毕,再释放分配的空间,归还系统。

下面程序片段表明了定义一个动态二维数组 $a[i][j]$ 的方法。

```
.....
int i,n,m;
huge float * * a;
a = (float *)faralloc(n,sizeof(float));
for(i=0;i <= n-1,i++)a[i] = (float *)faralloc(m,sizeof(float));
.....
```

2. 人机界面的设计

SRDS 系统的主要目的是给工程技术人员提供一个系统可靠性分析的有力工具,因此不仅要求其具有先进、高效的算法,同时也应具备一个十分友好的人机界面。因此,我们在人机界面设计时,拟定了以下目标:第一,汉字显示必须独立于汉字操作系统之外,即汉字不是在汉字系统支持下显示的,而是在西文 DOS 支持下工作。实现这一点就能保证程序不受制于特定的汉字操作系统,因而具有通用性,而且更为重要的是将汉字系统占用的空间让位给仿真系统;第二,中、西文字符必然能用屏幕的点阵位置定位;第三,所有的中、西文字符的字号、字型、字符颜色是可选的;第四,图像和文字必须共存;第五,汉字菜单驱动,系统所有功能均由下拉式汉字菜单驱

动。

显然,要实现上述要求,系统只能在西文 DOS 图形方式下工作。在西文图形方式下可以做到图文共存,需要解决的问题主要是汉字的显示与图形方式下输入函数的处理。

要做到精确到点阵的汉字显示,首先必须提取汉字的点阵信息,可以直接到汉字库中提取点阵信息。有了点阵信息就可直接将其写入显示缓冲区,从而以点阵定位显示汉字,而且还能通过一定的算法做到汉字放大,颜色变换等。通过提取不同字体的汉字库点阵信息,可做到变换字体。我们在这里用了 2.13H 中的各种字体的 24 点阵汉字库。

在程序运行时弹出若干对话框供用户输入数据,输入的数据要能显示在适当的位置,且要有插入、删除等功能,而 C 语言在图形方式下没有一个具有上述功能的函数,为此我们开发了一个图形方式下的数据输入函数,能在指定的坐标位置输入并回显数据,且具有模拟光标,插入、删除等功能。

另外,我们采用了立体投影和凹凸式汉字菜单,彩色亮条选择,具有逼真的立体效果,美观新颖。

3. 结构化程序设计

在系统设计时,严格按结构化程序设计的要求进行。在开发中宁可牺牲一点时空,也要提高系统的可靠性、可测试性与可维护性。系统中各个模块做到了功能独立,短小紧凑。其中的一些模块是小型的工具模块,如函数曲线生成器、文本文件阅读器、简易全屏编辑器、屏幕图形随机存储器等,它们中的一部分本身已编译成了可执行文件。

4. 进程管理

为使系统结构紧凑,便于调试与维护,广泛采用了调用子进程的方法。如用户欲查看结果文件 resultfile,则可调文本文件阅读器 readtext.com,对应语句如下:

```
spawnml(P-WAIT,"readtext.com","resultfile",NULL);
又如,在系统中实现 DOS SHELL 功能,可用语句段:
..... / * 储存屏幕当前内容 * /
outcharacter(键入 EXIT 返回 SRDS 系统",x,y,color,n,space,'s');
/ * 在(x,y)处以颜色 color,放大倍数 n,间距 space 显示字体汉字 * /
spawnml(P-WAIT,"c:\command.com",NULL);
..... / * 恢复屏幕内容 * /
```

(下转第 24 页)

(上接第 27 页)

此外还采用了覆盖、口令设置、数据容错、音响提示、磁盘 CACHE 等多项技术,在此不一一叙述。采用上述技术的目的是为了解决以往普通微机无法问津的大型可修复系统的可靠性数字仿真问题,并给用户提供一个良好的人机界面,以及提高系统的可靠性与可维护性。

五、系统验证

对于 SRDS 这样一个大型的随机系统仿真软件,在建立了仿真模型与程序后,还必须对其进行验证。对于随机性问题,我们可以这样验证,假定所有变量均为确定

值,将其代入系统运行。对于确定值,总可用解析方法得到理论解。若理论解与系统运行结果一致,则可以推断程序在逻辑推理上是正确的(对 SRDS 系统而言)。另外,SRDS 系统目前能处理的元件相关分布有正态分布、指数分布、威布尔分布等十余种,其中仅指数分布情况有理论解。可将指数分布情况的仿真解与其它成熟的可靠性分析程序的结果进行对比验证。以上两种验证均已通过。当然,最为重要的是接受实践的检验。目前,SRDS 系统已投入实际工程系统的可靠性数字仿真使用中。

参考文献:(略)