

# 蒙特卡罗法仿真实例及其结果分析

魏选平, 卞树檀

(西安第二炮兵工程学院, 陕西 西安 710025)

**摘要:**为研究大型复杂系统的可靠性问题, 利用蒙特卡罗可靠性数字仿真法, 在 Windows 操作平台上采用 Fortran PowerStation V4.0 编程工具开发了相应的仿真实例。实验表明, 具体仿真计算结果与理论分析结论吻合, 从而显示出蒙特卡罗仿真法处理可靠性问题的优越性。对多个可靠性数字仿真实验的分析表明, 不同系统的故障树可靠性仿真具有相似的模式, 仿真程序具有普遍的适用性。

**关键词:**蒙特卡罗法; 可靠性; 仿真

**中图分类号:** O242.2 **文献标识码:** A

## The Example of Monte-Carlo Simulation and Result Analysis

WEI Xuan-ping, BIAN Shu-tan

(Xi'an Second Artillery Engineering College, Xi'an Shaanxi 710025, China)

**Abstract:** To study the reliability of the large complicated systems, this paper uses the Fortran PowerStation V4.0 integrated developing environment to develop an example on the Windows platform with the method of Monte-Carlo simulation. The experiment shows that the result of simulation computing and are consistent with that of theoretic analysis. And this indicates the advantages of the Monte-Carlo simulation to deal with the reliability problem. It is clear that fault-tree reliability simulation of different systems has similar model to analyze the multi-reliability numerical simulation experimentation, and the simulation program has common usability.

**Keywords:** Monte-Carlo method; Reliability; Simulation

## 0 引言

随着现代科学技术的不断发展, 大型复杂系统的研制日益增多, 对其可靠性及有效性的分析与评定已成为系统研制过程中不可缺少的重要组成部分。对于某些系统而言, 常规的解析分析法已不能完全解决系统可靠性的有关问题。系统的可靠性问题可以用故障树分析法(FTA)加以解决。当其中包括寿命为非指数分布或当系统的故障树规模较大时, 用可靠性仿真求解系统的可靠性将十分有效<sup>[1]</sup>。当前, 蒙特卡罗方法相关理论的介绍很多, 但应用实例却很少见, 或者是应用实例过程太过简单。

本文详细地叙述了基于 Windows 环境下用 Fortran PowerStation V4.0<sup>[2]</sup> 程序语言开发的蒙特卡罗方法可靠性仿真实例, 并突出该法应用过程和关键环节的介绍, 文末进行了仿真结果的分析, 总结出了值得借鉴的经验。

## 1 仿真实例

设系统 S 的故障树由 6 个门事件和 7 个基本部件组成<sup>[3]</sup>, 如图 1 所示。

已知各基本部件的失效分布函数为  $F_i(t)$ , 其分布类型和特征参数如表 1 所示(表中  $\lambda_i$  为失效率;  $\mu_i$  为正态分布的数学期望;  $\sigma_i$  为正态分布的标准差)。

收稿日期: 2001-06-28; 修回日期: 2001-07-24

作者简介: 魏选平(1969-), 男, 硕士生, 从事系统可靠性评估研究。

表 1 各基本部件失效分布类型及参数

Tab.1 Types of failure distribution of basic parts and parameters

基本部件	失效密度函数 $f_i(t)$ 类别	$f_i(t)$ 的特征参数
$Z_1(101)$	指数分布 $f_1(t)$	$1/\lambda = 2\,500\text{ h}$
$Z_2(201)$	正态分布 $f_2(t)$	$\mu = 1\,000\text{ h}, \sigma = 130\text{ h}$
$Z_3(102)$	指数分布 $f_3(t)$	$1/\lambda = 2\,500\text{ h}$
$Z_4(103)$	指数分布 $f_4(t)$	$1/\lambda = 2\,500\text{ h}$
$Z_5(202)$	正态分布	$\mu = 1\,000\text{ h}, \sigma = 130\text{ h}$
$Z_6(104)$	指数分布	$1/\lambda = 2\,500\text{ h}$
$Z_7(105)$	指数分布	$1/\lambda = 2\,500\text{ h}$

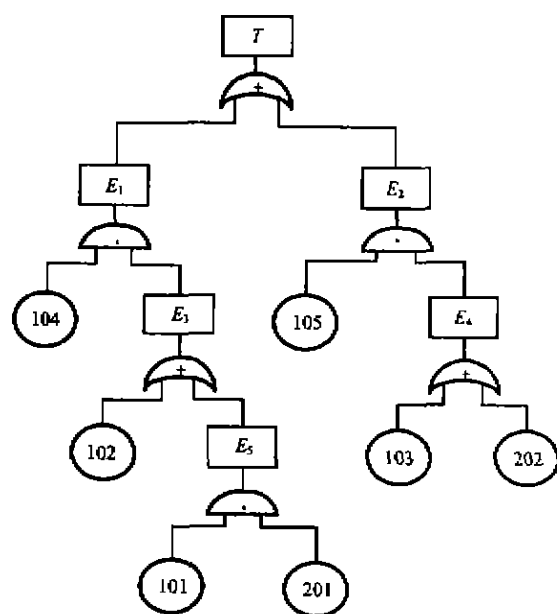


图 1 系统 S 的故障树

Fig 1 System S fault tree

## 2 仿真运行

根据该系统故障树表示的系统可靠性框图,利用故障树仿真法原理,可得该例故障树法可靠性仿真的算法框图,如图 2 所示。

现简要说明如下:

(1) 系统规定的最大工作时间为  $T_{\max}$ , 将它等分为  $M$  个时间间隔, 每个间隔为  $\Delta T$ , 则有

$$\Delta T = T_{\max} / M$$

如图所示, 其中  $t_r$  为第  $r$  个区间终点的时间, 即有

$$t_r = \sum_{i=1}^r \Delta t_i \quad (1)$$

(2) 规定仿真运行总次数为  $N_s$ , 仿真运行的序号为  $N$ , 故  $N = 1, 2, \dots, N_s$ 。

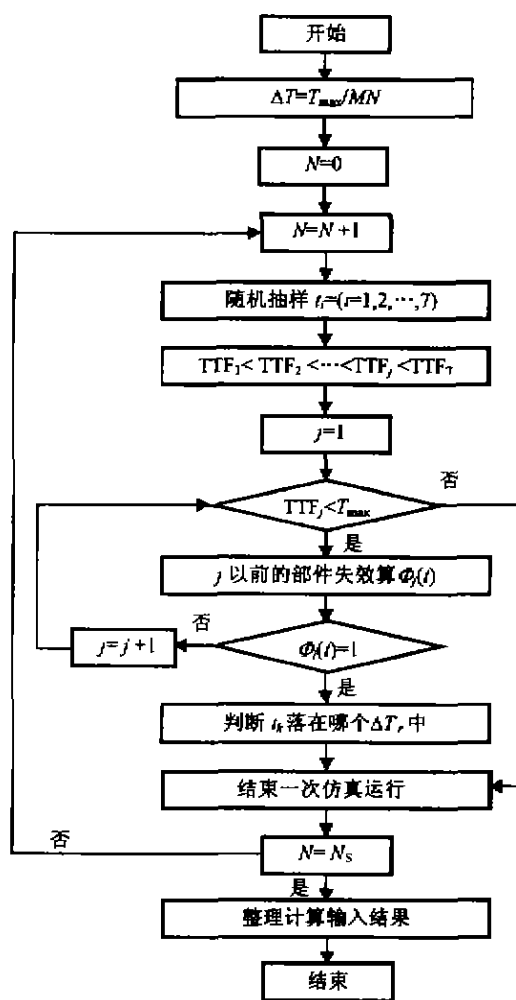


图 2 仿真算法流程图

Fig 2 Flow chart of simulation algorithm

(3) 该系统的故障树中有 7 个底事件, 它们是系统 S 中 7 个基本部件的失效事件, 对于第  $i$  个基本部件, 其失效分布函数  $F_i(t)$  为已知分布, 如表 1 所示。

(4) 由故障树可得结构函数  $\Phi[X(t)]$

(5) 在第  $j$  次仿真运行中,第  $i$  个基本部件  $Z_i$  的失效时间  $t_{ij}$ 、用蒙特卡罗随机抽样产生

$$t_{ij} = F_i^{-1}(\xi_j) \quad (2)$$

因而在第  $j$  次仿真运行中,7 个基本部件的失效时间抽样值为

$$t_{1j}, t_{2j}, \dots, t_{ij}, \dots, t_{7j}$$

(6) 在每次仿真运行中,将  $t_{ij} (i=1, 2, \dots, 7; j=1, 2, \dots, N_s)$  按抽样时间由小到大排序,用 TTF <sub>$i$</sub>  表示,即有

$$\text{TTF}_1 < \text{TTF}_2 < \dots < \text{TTF}_i < \dots < \text{TTF}_7 \quad (3)$$

(7) “通扫故障树”,求出  $t = t_k = \text{TTF}_k$

(8) 当一次仿真运行结束,并求出系统失效时间  $t_b$  后,即判断失效时间  $t_b$  落在各时间区间内。

(9) 重复上述(5)~(8)项内容,直到仿真运行  $N_s$  次

(10) 统计落入各个  $\Delta T_r$  中的失效数  $\Delta m_r$ , 如  $\Delta T_1$  中有  $\Delta m_1$  个失效,  $\Delta T_2$  中有  $\Delta m_2$  个失效等。

(11) 求系统的可靠性指标:系统的平均寿命 MTBF、在  $T_R(I)$  内的故障树的可靠度  $R(I)$ 、失效概率  $P_R(I)$ 。

### 3 仿真结果及其分析

仿真程序在 Windows 环境下用 Fortran Power-Station V4.0 编程调试通过,其仿真运行结果如下。

#### 3.1 打印结果

当仿真次数  $N_s=1\,000$  时的打印结果为:

```
* * * * * THE OUTPUT * * * * *
- - - MTBF = 1 163.400 - - -
```

当仿真次数  $N_s=10\,000$  时的打印结果为:

```
* * * * * THE OUTPUT * * * * *
- - - MTBF = 1 162.705 - - -
```

$R$ 、 $P_R$  的打印数据略。

按照  $R$ 、 $P_R$  打印的数据,利用 Matlab 5.0<sup>[4]</sup> 画出曲线图 3、图 4。

#### 3.2 结果分析

图 3、图 4 给出了  $R(I)$ 、 $P_R(I)$  的仿真结果。从图中可见,由 1 000 次再增加到 10 000 次,其仿真运行结果很接近,说明已达到了稳定状态。但在进

行系统仿真运行时,有些参数的选择可能影响到仿真的精度,因此有必要进行参数选取的探讨。现在就通过仿真试验结果来分析。

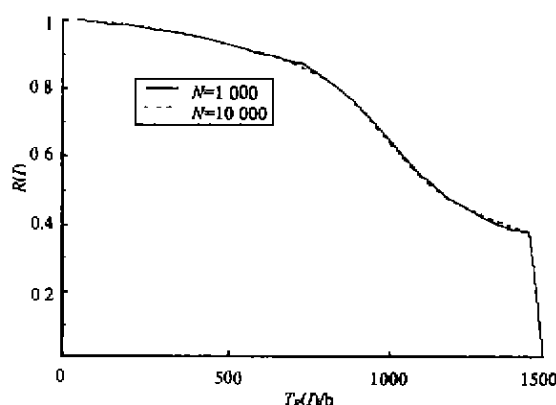


图 3 可靠度  $R(I)$

Fig.3 Reliability  $R(I)$

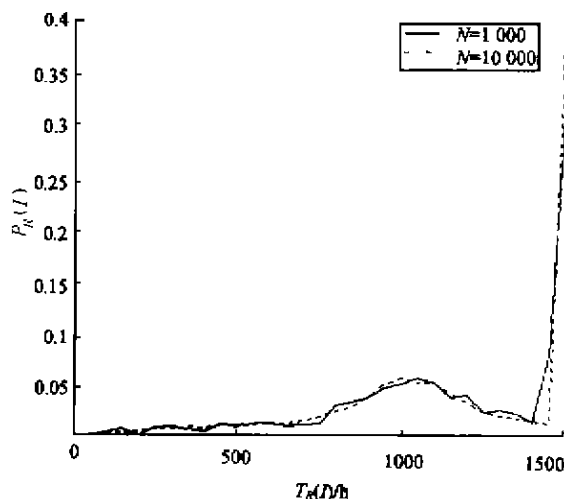


图 4 失效概率  $P_R(I)$

Fig.4 Failure probability  $P_R(I)$

#### a. 仿真次数的选取

蒙特卡罗方法要求仿真次数要足够多,以满足一定的仿真精度。通常可采用试算的方法,即逐步增加仿真运行次数,并观察其输出结果的变化,要求其数值波动的总趋势是稳定收敛。当给出精度要求时,可以通过估计值来比较。本例中没有进行定量分析,但是从 1 000 次与 10 000 次仿真结果可以作出定量分析。从图中可以看出,对可靠度  $R(I)$ 、失效概率  $P_R(I)$  和系统平均寿命 MTBF 的计算而言,只要进行 1 000 次仿真运行即可。

b. 仿真运行中  $T_{max}$  值的影响

通常在仿真时要根据系统可能发生失效时间的估计,人为先定一个最大仿真时间进行运行。若仿真中统计出落在此值以后的失效次数较多,则应加大此数值,直到仿真过程中系统失效时间绝大部分在此值以下为止。由于  $T_{max}$  值取得太大会造成过多的仿真运行,而太小又影响统计的精度,因此要取得恰当。本实验中选取  $T_{max} = 1\ 500\ h$ 。

## c. 结果精度的影响

一般来说,随机数都是经过检验的,因此其均匀性、独立性都符合随机性要求。实际系统仿真表明,由于随机数发生器的初值选取不同,其结果还会有差异。

现结合本实例进行实际试验,当取初值  $J_0 = 8\ 388\ 605$  进行仿真运行时,其结果相差不大。

## d. 语言的优点

仿真程序在 Windows 环境下用 Fortran Power-Station V4.0 程序语言设计,同传统的 DOS 下的基于文本方式的环境相比,Windows 具有强大的编程功能,其中包括标准化图形界面以及模块化和可扩展

展性等。

## 4 结论

经分析,该仿真结果与实际吻合,尤其是作者以此法对多个系统进行仿真实验都普遍适用,且方便易行,充分显示出蒙特卡罗仿真法处理可靠性问题的巨大优越性。通过多个可靠性数字仿真实验,还感到不同系统的故障树可靠性仿真具有相似的模式,其仿真程序有普遍的适用性,对不同的系统,只要分析出其故障树后,再利用仿真程序,可靠性问题就会迎刃而解。

## 参考文献

- [1] 冯耀霖. Fortran PowerStation V4.0 [M]. 西安:电子科技大学出版社,1999.
- [2] 杨为民,盛一兴. 系统可靠性数字仿真[M]. 北京:航空航天大学出版社,1990.
- [3] 宋保维. 系统可靠性设计与分析[M]. 西安:西北工业大学出版社,2000.
- [4] 李一涛,贺勇军. Matlab——应用数学篇[M]. 北京:电子工业出版社,2000.

(上接第 32 页)

$K_p = 5.7, K_l = 0.01, K_D = 54$ 。其中的摩擦力矩  $T_f$  由下式给出:

$$T_f = T(\omega_y) = K_1 \text{sign}(\omega_y) + K_2 \omega_y + K_3 \omega_y \sqrt{|\omega_y|} + K_4 \omega_y^3 \sin(\omega_y t) \quad (44)$$

其中,等式右端第 1 项表示库仑摩擦力矩,第 2 项表示气动力矩,第 3 项表示阻尼摩擦,第 4 项表示动不平衡力矩;各项的系数为:  $K_1 = 5 \times 10^{-4}$ ,  $K_2 = 1 \times 10^{-5}$ ,  $K_3 = 1 \times 10^{-6}$ ,  $K_4 = 1 \times 10^{-8}$ 。

假设其他各种扰动的总和为  $T_{dy} = 5 \times 10^{-4} [\cos(2\pi t + 0.5) + 1] \text{ N} \cdot \text{m}$ ,在有,无补偿两种情况下对系统进行数学仿真,结果如图 3 所示。上下两图的纵坐标分别是卫星角度  $\theta_y$  和角速度  $\omega_y$ ,横坐标为时间  $t$ 。其中,右图为仿真曲线的末端放大。可见,神经网络扰动补偿后系统的响应速度得到了明显提高,系统的稳态误差显著降低。

## 4 结论

本文提出了 RBF 网络的改进 ROLS 学习算法。

基于该算法,利用 RBF 网络建立了扰动力矩的神经网络模型,进而设计了卫星姿态控制系统扰动力矩神经网络补偿控制方案。

数学仿真结果表明,改进的学习算法可以有效建立扰动力矩的 RBF 网络模型,由此提出的控制方案可以提高姿态控制精度。

## 参考文献

- [1] CHEN S, COWAN C F N, GRANT M. Orthogonal least squares learning algorithm for radial basis function networks[J]. IEEE Trans Neural Networks, 1991, 2(2): 302-309.
- [2] Chen S, Chng E S, Alkadhimi K. Regularized orthogonal least squares algorithm for constructing radial basis function networks[J]. INT J CONTROL, 1996, 64(5): 829-837.
- [3] 徐立新,王常虹,庄显义. 基于神经网络模型的伺服系统控制与补偿方法研究[J]. 宇航学报, 1998, 19(3): 83-92.