基于 Markov 过程的硬/软件综合系统可靠性分析

于 敏[1],何正友[2],钱清泉[2]
(1.西南交通大学信息科学与技术学院,四川成都 610031; 2.西南交通大学电气工程学院,四川成都 610031)

摘 要: 现代大型监控系统通常是一个复杂的硬/软件综合系统,其可靠性分析对于系统的设计、评估具有重要意义.综合考虑硬件、软件特点以及两者之间的相互作用关系,提出一种基于 Markov 过程的综合系统可靠性分析模型,模型中将系统失效分为硬件失效、软件失效与硬/软件结合失效.实际应用中,由于系统的状态数较大,提出利用循环网络方法对 Markov 状态转移方程进行求解,从而方便地得到系统处于各状态的瞬时概率与稳态概率.通过分析硬/软件综合系统可靠度、可用度与系统可靠性参数之间的关系,指出硬/软件结合失效将影响系统可用度,忽略硬/软件结合失效将导致可靠性估计值偏离实际值.

关键词: 硬/软件综合系统; Markov 过程; 硬件失效; 软件失效; 硬/软件结合失效; 可靠度; 可用度
中图分类号: TB114.3 **文献标识码:** A **文章编号:** 0372-2112 (2010) 02-0473-07

Reliability Analysis of Combined Hardware/Software System Based on Markov Process

YU Min[1], HE Zheng-you[2], QIAN Qing-quan[2]
(1. *School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan* 610031, *China*;
2. *School of Electric Engineercng, Southwest Jiaotong University, Chengdu, Sichuan* 610031, *China*)

Abstract: Present-day large-scale monitor systems are typical complex hardware/software systems, the reliability analysis of which is very important to their design and assessment. Considering characteristic of hardware, software and mutual function relations between them, this article proposes a reliability model of combined hardware/software system based on Markov process. This model divides system failures into three categories: hardware failures, software failures and hardware/software interaction failures. Because of the greater number of system states in practical application, the Markov model may cause analysis of system to be very difficult, so the solution based on the recursive network method is presented to solve state transition equation, and thus obtain instantaneous and stable probability of each states conveniently. Through reliability and availability analysis of combined hardware/software system related to system reliability parameters points out that hardware/software interaction will influence system availability, if ignoring the hardware/software interaction failures, the estimation of reliability will deviate from actual value.

Key words: combined hardware/software; Markov process; hardware failures; software failures; hardware/software interaction failures; reliability; availability

1 引言

诸如工业控制、卫星、通信等复杂计算机系统均是包括硬件子系统和软件子系统的硬/软件综合系统,系统功能由复杂的硬件及软件共同支持下才能完成,硬件或软件故障都有可能引起系统失效.针对硬/软件综合系统可靠性,国内外部分学者已开展了相关研究[1~3],但分析模型一般假定硬件与软件是相互独立的,并没有考虑两者的区别以及两者的相互作用关系,然而,硬件与软件之间是相互影响的,一方面,软件方法通过"故障-容错"可以消除硬件故障对系统造成的影响[4,5],另一方面,未预料到的硬件失效亦会使运行在这些硬件上的软件故障,给系统的可靠性带来变数[6].因此,为了评价硬/软件综合系统的可靠性,必须从硬/软件结合的角度来认识问题并建立综合系统的可靠性模型.

文献[7]在考虑硬/软件结合故障的基础上建立了硬/软件综合系统可靠度分析模型,但未考虑可维修性,然而,除导弹、卫星等系统之外多数硬/软件综合系统都属于可维修系统.硬件或软件故障可以通过维修恢复正常,其中,部分硬件子系统故障使综合系统处于降级工作状态,尽管系统能够正常运行,也必须及时维修才能维持系统容错的冗余结构.本文考虑系统的容错技术、根据系统的多种失效及维修模式、利用 Markov 过程建立了硬/软件综合系统可靠性模型,考虑到当系统的状态数目较大时 Markov 状态转移方程的求解变得很困难,提出了利用循环网络方法对其求解,最后通过数值计算对硬/软件综合系统可靠度、可用度进行分析.

收稿日期:2008-08-02;修回日期:2009-10-23
基金项目:国家自然基金(No.50878188)

2 硬/软件综合系统失效模式

在对硬/软件综合系统进行可靠性分析时需区分“故障”与“失效”,故障是与系统相互作用的一种错误,若系统无故障保护机构,则故障在系统级上就会表现为失效.本节在分析硬件、软件故障原因的基础上,结合硬件、软件各自的容错方式给出硬/软件综合系统的失效模式.

2.1 故障原因

可将硬/软件综合系统分为硬件、软件两个子系统,相应地将故障分为硬件故障与软件故障:

(1)硬件故障根据故障的持续时间又可分为永久性故障和暂时性故障.永久性故障是硬件物理性能随工作时间递增而退化所致,暂时性故障则是由外部或内部不确定因素(如温度、电磁干扰、内部噪音等)引起的[8].永久性故障的持续时间一般被认为超过系统工作时间,需要进行维修或更换才可复原,而暂时性故障持续时间很短,譬如不超过几个毫秒,所以,暂时性故障不一定需要修理或更新.

硬件的永久性故障按范围可进一步分为“局部故障”和“全局故障”,如电源故障会使整个硬件子系统失效,因此它属于全局故障,再如内存或存储器某些部分的永久性故障,可通过停止故障部分使系统降级运行,故它们属于局部故障.

(2)软件故障只与软件内部的缺陷有关,缺陷导致系统运行中出现可感知的不正常、不正确和不按规范执行的错误状态[9],软件缺陷是由于在需求分析、设计或程序编写等阶段中引入的,其一旦生成则将长期潜伏在软件中直到排除为止,并随时都有导致软件失效的可能.

2.2 容错技术

硬件或软件一旦发生故障将导致子系统状态出错,并可能引发子系统或综合系统的失效,而容错技术则可使系统在硬件或软件出现故障的情况下仍能正常或降级运行,从而避免整个系统失效.

容错技术的出发点就是承认故障是不可避免的,然后通过冗余资源提供的信息来克服故障影响,冗余资源概括起来有硬件、信息、时间和软件 4 种冗余[10],但在硬/软件综合系统中最终都将体现为硬件容错和软件容错,其中,硬件容错主要是通过冗余硬件和相应的错误处理软件来实现[11],软件容错则常以设计相异性为理念,通过软件冗余屏蔽软件中的错误.

硬件或软件故障时,系统主要有“失效接替”、“失效弱化”及“失效保护”3 种容错策略,其中,失效接替是指由备用模块接管故障模块并继续工作;失效弱化指当系统某部分出现故障后允许缓慢降级,其余部分继续工作使系统的功能维持到任务结束或修复完成;而失效保护则是系统安全的最后防线,即当故障超过了系统的容错能力时,该方法在一个安全状态下终止系统并使其进入安全状态.

2.3 失效模式

根据硬件、软件的故障原因及相应的容错技术,将硬/软件综合系统的失效模式分为硬件失效、软件失效与硬/软件结合失效.

硬件失效主要由硬件的全局故障、未能及时恢复的暂时性故障与超过了系统容错能力的硬件局部故障引起的.硬件全局故障显然会使硬件失效;为了消除硬件暂时性故障的影响,常常采用指令重试或程序卷回,即再做一遍或若干遍直到瞬时故障消失为止,但是在系统运行过程中,可能因为未能及时从故障中恢复而造成系统失效;当硬件局部故障发生时,系统执行故障检测、隔离、重组等一系列动作,这些行为都可用覆盖率来度量,而不完备的覆盖率仍会引起系统的失效.

软件失效是由软件缺陷引发的故障而引起的,对于容错软件而言,大部分软件故障可通过容错策略进行屏蔽,如采用 N 版本技术的容错软件的失效仅由多数版本软件的重合故障及共模故障引起,这里的重合故障指多数版本软件同时出现差错,并可通过裁决器正确判断,系统可安全停机;共模故障则是由于出现差错的模式是一样的,检测算法无法辨别而使系统进入不安全状态.

硬/软件结合失效是指硬件局部故障引起的软件失效,硬件局部故障可能使软件在一个不同工作配置下的异常情况,将这种由于硬件操作环境的改变而引起的软件失效称为硬/软件结合失效,如内存模块局部故障是一种操作环境的异常配置,可能会使软件在这种异常环境下造成失效.

硬/软件结合失效反映了硬件故障对软件的影响,但综合系统中的软件也会影响到硬件的失效情况,这主要是由于一个实际的容错系统多数要借助于软件来实现,即软件的故障检测、重组等也将影响硬件的可靠性.

3 硬/软件综合系统的 Markov 模型

本节在考虑硬件与软件不同故障类型、容错技术以及失效模式的基础上建立硬/软件综合系统可靠性模型,系统具有多种故障、容错、失效及维修模式,目前最适合建立这种可靠性模型的方法是 Markov 模型[12,13],利用 Markov 模型可以较为真实地反映系统的工作情况.运用 Markov 模型来评估系统可靠性指标时,硬/软件综合系统的各个状态可以看作是一个离散时间、离散状态的 Markov 过程,为了应用 Markov 过程建立系统可靠性模型作如下假设与分析:

(1)假定软件采用 N 版本技术,并设 $N=3$,即系统配置有 3 套功能相同、版本不同的程序;根据 Jelinski-Moranda(JM)模型[14],软件失效率是随时间变化的函数,设单一软件版本故障、多数版本(2 个或 3 个)软件重合故障与共模故障的失效率分别为 $\lambda_{s0}(t)$、$\lambda_{s1}(t)$、$\lambda_{s2}(t)$;

(2)设硬件暂时性故障失效率为 λ_{ht},在允许时间内修复的覆盖率为 α,此时可认为修复时间忽略不计,即修复率为 ∞(记为 μ_∞);未修复的暂时性故障则需额外的恢复时间,设恢复率为 μ_{ht};

(3)设硬件全局故障、局部故障的失效率分别为 λ_{ha0}、λ_{hp},局部故障可使系统降级工作(为了简化模型,假设硬件只有一级降级状态),设局部故障的检测与重组的覆盖率分别为 c、r,重组成功后降级运行系统的硬/软件结合故障与硬件全局故障的失效率分别为 λ_{hs1}、λ_{ha1};未检测到的局部故障将使系统由于不能重组而使硬件全局故障或硬/软件结合故障,设失效率分别为 λ_{ha2}、λ_{hs2};

(4)设维修时间服从指数分布,硬件永久性故障、软件故障与硬/软件结合故障的维修率分别为 μ_{hp}、μ_s 与 μ_{hs},其中,单一软件版本故障的修复时间亦可忽略不计,所以假设其修复率为 μ_∞.

为区别不同情形,定义硬/软件综合系统的状态如下:

状态 0:硬件与软件都正常工作,系统正常工作;其中,0_h、0_s 分别为系统中存在可自动恢复的硬件暂时性故障与可通过容错策略进行屏蔽的单一版本软件故障,由于它们不影响系统正常工作,把它们也都定义作状态 0;

状态 1:硬件发生暂时性故障,且在允许时间范围内恢复失败,系统失效;

状态 2:硬件发生不可测局部故障,系统降级工作;

状态 3:硬件发生可测且可重组的局部故障,系统降级工作;

状态 4:硬件发生可测但重组失败的局部故障,系统安全停机;

状态 5:硬件发生全局故障,系统失效;

状态 6:硬件局部故障引起软件失效,系统失效;

状态 7:多数版本软件发生重合故障,系统安全停机;

状态 8:多数版本软件发生共模故障,系统失效.

由定义可知,状态 0 为系统的正常工作状态,状态 2 与状态 3 为降级工作状态,状态 4 与状态 7 为安全状态,其它状态则为系统失效状态.Δt 时间内系统不同状态之间的状态转移图可见图 1,为了简便,在图 1 中略去了状态之间的转移率系数 Δt,并略去了系统停留在原状态的概率.

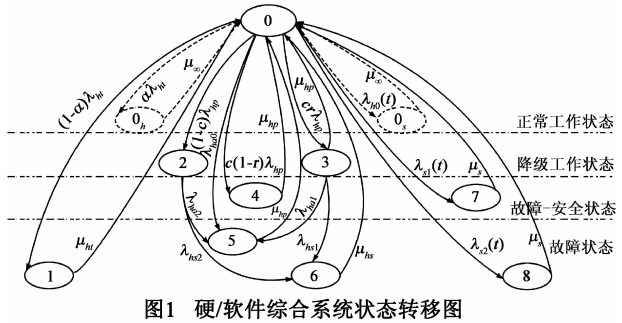


图1　硬/软件综合系统状态转移图

根据硬/软件综合系统的状态转移图可写出微系数矩阵 $\boldsymbol{P}(\Delta t)$,即

$$\boldsymbol{P}(\Delta t)=\begin{bmatrix} \times & (1-\alpha)\cdot\lambda_{ht}\Delta t & (1-c)\cdot\lambda_{hp}\Delta t & cr\lambda_{hp}\Delta t & c(1-r)\cdot\lambda_{hp}\Delta t & \lambda_{ha0}\Delta t & 0 & \lambda_{s1}(t)\Delta t & \lambda_{s2}(t)\Delta t \\ \mu_{ht}\Delta t & \times & & & & & & & \\ 0 & & \times & & & \lambda_{ha2}\Delta t & \lambda_{hs2}\Delta t & & \\ \mu_{hp}\Delta t & & & \times & & \lambda_{ha1}\Delta t & \lambda_{hs1}\Delta t & & \\ \mu_{hp}\Delta t & & & & \times & & & & \\ \mu_{hp}\Delta t & & & & & \times & & & \\ \mu_{hs}\Delta t & & & & & & \times & & \\ \mu_s\Delta t & & & & & & & \times & \\ \mu_s\Delta t & & & & & & & & \times \end{bmatrix} \tag{1}$$

式(1)中未写出的元素都为 0,对角线元素用“×”表示,其值为

$$P_{ii}(\Delta t)=1-\sum_{j\neq i}P_{ij}(\Delta t)\quad(i=0,1,\cdots,8)$$

由微系数矩阵可以得到 Δt 时间内各状态转移概率,再利用全概率公式可得系统的状态转移方程为:

$$\begin{cases} P_0(t+\Delta t)=[1-(1-\alpha)\lambda_{ht}\Delta t-\lambda_{hp}\Delta t-\lambda_{ha0}\Delta t-\lambda_{s1}(t)\Delta t-\lambda_{s2}(t)\Delta t]P_0(t)+\mu_{ht}\Delta tP_1(t) \\ \qquad +\mu_{hp}\Delta tP_3(t)+\mu_{hp}\Delta tP_4(t)+\mu_{hp}\Delta tP_5(t)+\mu_{hs}\Delta tP_6(t)+\mu_s\Delta tP_7(t)+\mu_s\Delta tP_8(t) \\ P_1(t+\Delta t)=(1-\alpha)\lambda_{ht}\Delta tP_0(t)+(1-\mu_{ht}\Delta t)P_1(t) \\ P_2(t+\Delta t)=(1-c)\lambda_{hp}\Delta tP_0(t)+[1-\lambda_{ha2}\Delta t-\lambda_{hs2}\Delta t]P_2(t) \\ P_3(t+\Delta t)=cr\lambda_{hp}\Delta tP_0(t)+(1-\mu_{hp}\Delta t-\lambda_{ha1}\Delta t-\lambda_{hs1}\Delta t)P_4(t) \\ P_4(t+\Delta t)=c(1-r)\lambda_{hp}\Delta tP_0(t)+(1-\mu_{hp}\Delta t)P_4(t) \\ P_5(t+\Delta t)=\lambda_{ha0}\Delta tP_0(t)+\lambda_{ha2}\Delta tP_2(t)+\lambda_{ha1}\Delta tP_3(t)+(1-\mu_{hp}\Delta t)P_5(t) \\ P_6(t+\Delta t)=\lambda_{hs2}\Delta tP_2(t)+\lambda_{hs1}\Delta tP_3(t)+(1-\mu_{hs}\Delta t)P_6(t) \\ P_7(t+\Delta t)=\lambda_{s1}(t)\Delta tP_0(t)+(1-\mu_s\Delta t)P_7(t) \\ P_8(t+\Delta t)=\lambda_{s2}(t)\Delta tP_0(t)+(1-\mu_s\Delta t)P_8(t) \end{cases} \tag{2}$$

式(2)中,$P_i(t)$ $(i=0,\cdots,8)$为系统在 t 时刻处于状态 i 的概率,$P_i(t+\Delta t)$为系统在 t 时刻经过 Δt 时间处于状态 i 的概率,由于初始时刻硬件、软件子系统都正常工作,系统处于状态 0,因此,系统初始状态概率为 $P_0(0)=1, P_i(0)=0(i=1,\cdots,8)$.

4 硬/软件综合系统可靠性分析

可靠度、可用度是系统的重要可靠性指标,下面将在软件可靠性模型与循环网络模型研究的基础上对综合系统的可靠度、可用度进行分析.

4.1 容错软件可靠性模型

本文容错软件可靠性采用 JM 模型,它是最具代表性的符合 Markov 过程的软件可靠性模型.

由于单一版本软件故障不影响系统的可靠性,此处只考虑引起多数版本软件重合故障与共模故障的缺陷,分别记作第 1、2 类缺陷.假设缺陷是相对独立的,每类缺陷中的缺陷导致软件故障的概率相同,一旦查出即在所有涉及到的版本软件中都排除,且每次只排除一个缺陷,并假设在排错过程中不引入新的缺陷.

JM 模型反映了软件中第 i 类($i=1,2$,本节下同)剩余缺陷数 n_i 与软件失效率 $\lambda_{si}(t)$之间的关系.初始时刻软件中第 i 类残余缺陷数用一个未知但固定的常数用 $N_{0,i}$表示,当 k_i 个缺陷被排除后 $n_i=N_{0,i}-k_i$,设 $\lambda_{s,i}$ 为第 i 类故障的失效率比例常数,则 t 时刻软件的第 i 类故障失效率函数为:

$$\lambda_{si}(t)=(N_{0,i}-k_i)\lambda_{s,i}=n_i\lambda_{s,i} \tag{3}$$

软件从 k_i 个缺陷被排除到发生第 k_i+1 次故障的时间估计值为:

$$\hat{MTBF}_{k_i+1,i}=\frac{1}{\lambda_{k_i,i}}=\frac{1}{(N_{0,i}-k_i)\lambda_{s,i}} \tag{4}$$

根据式(3)与式(4)可以计算出 t 时刻软件第 i 类故障失效率,如重合故障失效率曲线如图 2 所示.

图 2 中初始重合错误数 $N_{0,1}=5$,失效率比例常数 $\lambda_{s,1}$取为 0.008,从图 2 可以看出,软件失效率是时间的减函数,但在两次故障期间,失效率 λ_{s1}为一个常数,因此,它可以作为下一个区间的初始条件而应用于 Markov 过程中.本文关心的是综合系统可靠度或可用度曲线的形状,所以,计算中根据 JM 模型计算 t 时刻软件中有 n_i 个第 i 类残余故障的概率为:

$$P_{n,i}(t)=\binom{N_{0,i}}{n_i}e^{-n_i\lambda_i t}(1-e^{-\lambda_i t})^{N_{0,i}-n_i},0\leqslant n_i\leqslant N_{0,i} \tag{5}$$

当观测系统可靠性规律时,式(5)仍将成立,在接下来的计算中,我们将使用这个方案来计算系统随时间变化的软件失效率值.

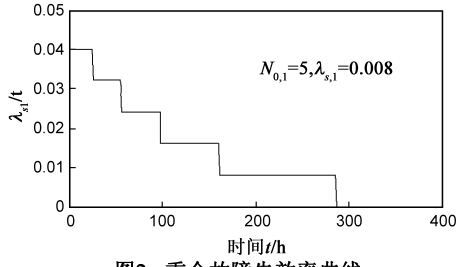


图2 重合故障失效率曲线

4.2 循环网络模型

系统的稳态可靠性指标可通过线性方程组的求解而得到,但瞬时可靠性指标则需通过对状态转移方程的求解而获得,常用的求解方法为先利用微分公式进行化简,然后通过拉普拉斯变换及其反变换对差分方程求解,但当系统状态数较大时,它们一般并不容易反演出来,所以,本文提出借助循环网络模型得到系统 t 时刻在各个状态的概率值.

循环网络模型分为输入层和输出层,对于有 n 个状态的状态转移图,每层将包括 n 个节点,并分别与 Markov 模型的 n 个状态对应,模型如图 3 所示.

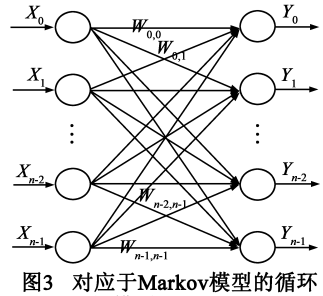


图3 对应于Markov模型的循环网络模型

图 3 中 $W_{i,j}$代表输入层 i 节点与输出层 j 节点的连结权值,用于描述 Δt 时间内系统从一个状态 i 转移到另一个状态 j 的概率,t 时刻模型输入层 $\boldsymbol{X}$(输入列向量)与输出层 $\boldsymbol{Y}$(输出列向量)的表达式见式(6)与式(7):

$$\boldsymbol{X}=\begin{bmatrix}X_0\\X_1\\\vdots\\X_{n-1}\end{bmatrix}=\begin{bmatrix}P_0(t)\\P_1(t)\\\vdots\\P_{n-1}(t)\end{bmatrix} \tag{6}$$

$$\boldsymbol{Y}=\begin{bmatrix}Y_0\\Y_1\\\vdots\\Y_{n-1}\end{bmatrix}=\begin{bmatrix}P_0(t+\Delta t)\\P_1(t+\Delta t)\\\vdots\\P_{n-1}(t+\Delta t)\end{bmatrix} \tag{7}$$

其中,输出层节点的处理特性为:

$$Y_j=\sum_{i=0}^{n-1}W_{i,j}\cdot X_i \tag{8}$$

于是,循环网络模型的计算过程为:

$$\begin{cases}\boldsymbol{Y}(t+\Delta t)=\boldsymbol{W}\boldsymbol{X}(t)\\\boldsymbol{X}(t+\Delta t)=\boldsymbol{Y}(t+\Delta t)\end{cases} \tag{9}$$

如对状态转移方程(2)求解,$\boldsymbol{W}$ 可取为式(1)所对应的微系统矩阵 $\boldsymbol{P}(\Delta t)$的转置,即:

$$\boldsymbol{W}=\boldsymbol{P}^{\mathrm{T}}(\Delta t)$$

根据初始状态概率 $\boldsymbol{P}(0)$得向量 $\boldsymbol{X}(0)=[1\,0\,0,\cdots,0]$,代入式(9),求出 $\boldsymbol{X}(0+\Delta t)=\boldsymbol{X}(\Delta t)$;再将 $\boldsymbol{X}(\Delta t)$代入同一公式,求出 $\boldsymbol{X}(2\Delta t)$;如此反复进行,得向量序列 $\{\boldsymbol{X}(k\Delta t)(k=0,1,2,\cdots)\}$;当合理设定 Δt 取值,经过循环网络方法可得 $t=k\Delta t$ 时刻系统处于各状态概率:

$$P_i(t)=X_i(t)(i=0,\cdots,8) \tag{10}$$

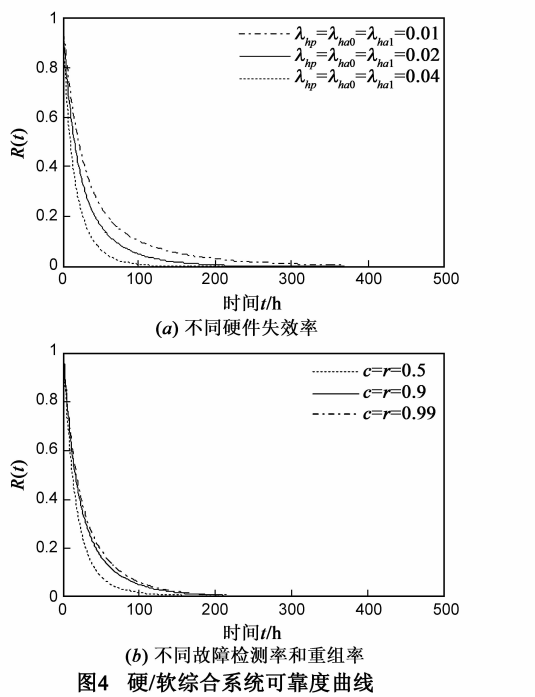
4.3　系统可靠度

系统可靠度 $R(t)$是指从初始时刻起直到时刻 t 系统一直处于工作状态的概率,因此,它可看作是无修复时系统处于正常工作状态或降级工作状态的概率,所以,为求硬/软件综合系统可靠度,可令系统故障状态为吸收状态,即在模型中令 $\mu_h=\mu_{hp}=\mu_s=\mu_{hs}=0$.

由于不对软件修复,因此,t 时刻软件中仍存在第 i 类($i=1,2$)残余缺陷数始终为 $N_{0,i}$,所以,第 i 类软件故障的失效率为常数 $\lambda_{s,i}(t)=\lambda_{s,i}N_{0,i}$,此时,利用循环网络模型求出系统 t 时刻在各个状态的概率,则硬/软件综合系统的可靠度为:

$$R(t)=P_0(t)+P_2(t)+P_3(t) \tag{11}$$

作为数值举例并为了减少可变参数的数量,假定系统的典型参数为:(1)硬件故障参数:$\lambda_{hp}=\lambda_{ha0}=\lambda_{ha1}=0.02$/h, $\lambda_{ha2}=0.9$/h, $c=r=0.9$, $\lambda_{ht}=0.9$/h, $\alpha=0.99$;(2)软件故障参数:$N_{0,1}=10$、$N_{0,2}=5$, $\lambda_{s,1}=\lambda_{s,2}=0.001$/h;(3)硬/软件结合故障参数:$\lambda_{hs1}=0.001$/h, $\lambda_{hs2}=0.9$/h,经过数值计算可以得到系统随时间变化的可靠度曲线如图 4 所示,同时,图 4 反映了失效率、覆盖率等参数与综合系统可靠度的关系.



(*a*) 不同硬件失效率

(*b*) 不同故障检测率和重组率

图4　硬/软综合系统可靠度曲线

从图 4 可以看出,随着时间的增加硬/软件综合系统可靠度降低并最终趋于零.图 4(a)反映了当其它参数固定时,若硬件子系统永久性故障失效率($\lambda_{hp}=\lambda_{ha0}=\lambda_{ha1}$)增加,系统的可靠度降低;图 4($b$)则描述了在其它参数不变的情况下,若故障检测率 c 和重组率 r 增加,则系统的可靠度将有所提高,这也符合实际情况.硬/软件综合系统可靠度还会受其它因素影响,一般来讲,系统可靠度随着覆盖率的增大而增大,随着子系统失效率的增大而减小,因此,工程中若要提高综合系统的可靠度,不但要降低系统的失效率,亦要提高容错设计中故障检测、重组等的覆盖率.

4.4　系统可用度

可用性是可修系统在时刻 t 处于正常工作状态或降级工作状态的能力,可用性的度量一般是瞬时可用度,记为 $A(t)$.根据 t 时刻系统在各个状态的概率计算硬/软件综合系统的瞬时可用度为:

$$A(t)=P_0(t)+P_2(t)+P_3(t) \tag{12}$$

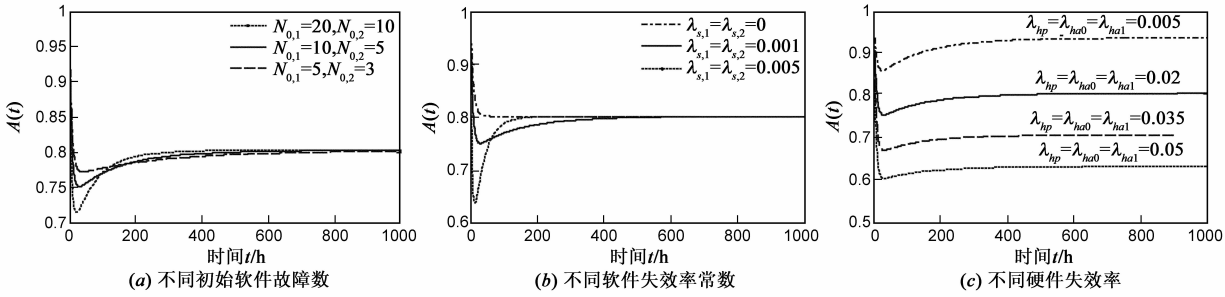
稳态可用度为 t 趋于∞时 $A(t)$的极限,即:

$$A(\infty)=A=\lim_{t\to\infty}A(t) \tag{13}$$

为了说明问题,下面给出综合系统的可用度数值分析,其中,失效率与覆盖率的典型参数与系统可靠度分析时相同,并假定系统维修率典型取值为:$\mu_{hp}=0.1$/h, $\mu_{ht}=0.9$/h, $\mu_s=0.12$/h, $\mu_{hs}=0.08$/h.经过数值计算可得 t 时刻瞬时可用度及稳态可用度,为了进一步

研究系统参数与可用度的关系,图 5 为系统取不同参数时硬/软件综合系统的可用度曲线.

从图 5 可以看出,随着时间的增长系统可用度开始下降,并在某一时刻达到最小点,随后,系统可用度逐渐增大,最终收敛于一个常数,这个常数为系统稳态可用度值.



(*a*) 不同初始软件故障数 (*b*) 不同软件失效率常数 (*c*) 不同硬件失效率

图5 硬/软件综合系统可用度曲线

图 5(*a*)反映了初始软件缺陷数($N_{0,1}$, $N_{0,2}$)分别取(20,10)、(10,5)与(5,3)时对系统可用度的影响,初始软件故障数越大,可用度最小值将越小,这是由于软件在投入使用的初始阶段存在的剩余错误造成的;而达到稳态时的系统可用度值是相同的,这是因为软件中残余缺陷被排除,软件中初始缺陷数对稳态可用度是没有影响的.所以,软件故障只影响系统可用度的初始阶段,而软件测试经过的时间越长,初始软件缺陷数对系统的影响将越小,作为工程应用,综合系统应在达到可用度极小值之前仍需进行大量软件测试.

图 5(*b*)为软件失效率比例常数 $\lambda_{s,1}=\lambda_{s,2}$分别为 0/h、0.001/h 与 0.005/h 时的系统可用度曲线,当 $\lambda_{s,1}=\lambda_{s,2}=0$/h 时,系统将不受软件故障影响,此时,综合系统的可用度曲线不存在最小点;当失效率比例常数不为 0 时,随着其值的增大系统可用度最小值越小,并使系统更快的达到稳态,这是由于当失效率比例常数增大时,将使软件在早期更易失效,使早期的可用度越小并使缺陷更易暴露及排除,所以系统更容易达到稳态.

图 5(*c*)为硬件永久性故障失效率 $\lambda_{hp}=\lambda_{ha0}=\lambda_{ha1}$ 从 0.005/h 变化到 0.05/h 时的系统可用度曲线,正如所预料的那样,图 5(*c*)中系统的可用度随着硬件失效率的增加而降低,且随着失效率的逐步增加,可用度曲线间的间距越来越小.同理,硬/软件综合系统可用度也将会受其它因素的影响,一般而言,系统可用度将随着覆盖率(α、c 与 r)与维修率(如 μ_h、μ_s 等)的增大或失效率的减小(如 λ_{hp}、λ_{ha0}等)而增大.

图 5 分析了系统参数对硬/软件综合系统可用度的影响,下面将进一步研究容错技术与硬/软件结合失效对系统可用度的影响,相应的可用度曲线如图 6 与图 7 所示.

图 6 反映了硬件容错技术对硬/软件综合系统可用度的影响,当不考虑硬件容错技术时,一旦硬件发生局部故障,系统就会立即失效,故可令 $\lambda_{hp}=0$、$\lambda_{ha0}=0.04$/h,图 6 验证了不考虑硬件容错技术的系统可用度明显低于采取容错技术时的情况.事实上,软件容错技术亦可以提高系统的可用度,如本文若未采用软件容错技术,则当软件发生第一类软件故障时,系统亦将立即失效.

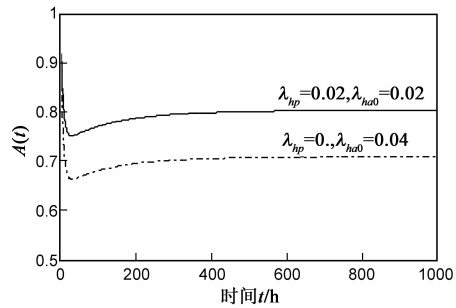


图6 硬件容错技术对系统可用度的影响

图 7 中分别为忽略与考虑硬/软件结合失效时的系统可用度曲线,其中,忽略硬/软件结合失效($\lambda_{hs1}=\lambda_{hs2}=0$)时的系统可用度将高于考虑硬/软件结合时的可用度,因此,若对实际系统的可靠性进行预计,忽略硬/软件结合失效将导致可靠性估计值偏离实际值,甚至可能造成错误.

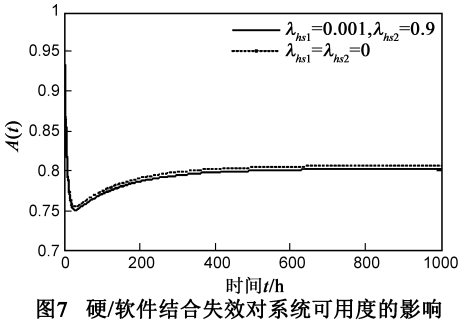


图7 硬/软件结合失效对系统可用度的影响

5 结论

本文在综合考虑了硬/软件综合系统多种失效方式的基础上,利用 Markov 过程建立了系统可靠性分析模型,实际应用中,由于系统状态数较大,提出利用循

环网络方法对状态转移方程求解,该方法避免了求解过程中拉普拉斯反演的困难,并易于编程实现,可以方便地得到系统处于各状态的概率.通过对系统可靠度与可用度进行分析,得到以下结论:

(1)由于软件中残余的故障的影响,系统可用度曲线在早期阶段达到一个最小点,因此,建议系统在达到这个最小点之前仍需要测试,避免系统可用度最小值;

(2)通过对系统可用度敏感性分析,得到系统可用度与系统硬件、软件参数之间的关系,如系统可用度随着硬件失效率增加而降低,且当失效率较大时,降低部件的失效率对增加系统可用度有明显效果等;

(3)针对容易忽略的硬/软件结合失效,本文验证了硬/软件结合失效对系统可用度影响,忽略硬/软件结合失效将导致综合系统可用度估计值偏离实际值,给系统可靠性预计造成误差.

参考文献:

[1] Friedman M A, Tran P. Reliability techniques for combined hardware/software systems [A]. In Proc, IEEE RAMS'92 [C]. Reliability and Maintainability Symposium, 1992. 290 – 293.

[2] Goel A L, Soenjoto J. Models for hardware/software operational performance evaluation[J]. IEEE Transactions on Reliability, 1992, R – 30: 232 – 239.

[3] Sumita U, Masuda Y. Analysis of software availability/reliability under the influence of hardware failures[J]. IEEE Transactions on Software Engineering, 1986, SE – 12: 32 – 41.

[4] Welke S R, Johnson B W, Aylor J H. Reliability modeling of hardware/software systems[J]. IEEE Transactions on Reliability, 1995, 44(3): 413 – 418.

[5] Kumar V K, Bechta D J. Reliability analysis of complex hardware-software systems[A]. In Proc IEEE RAMS'09[C]. Reliability and Maintainability Symposium, 1999. 178 – 182.

[6] Dugan, J B. Reliability analysis of a hardware and software fault tolerant parallel processor [A]. In Proc IEEE RDS'94 [C]. Reliability Distributed Systems, 1994. 74 – 83.

[7] Teng X, Pham H, Jeske D R. Reliability modeling of hardware and software interactions, and its applications[J]. IEEE Transactions on Reliability, 2006, 55(4): 571 – 577.

[8] 汪东升,郑纬民,王春露. TMR 计算机系统升级/降级重构技术[J].电子学报,1997,25(8):41 – 44.
Wang Dong-sheng, Zheng Wei-min, Wang Chun-lu. Upgrade and degradation reconfiguration for a TMR computer system [J]. Acta Electronica Sinca, 1997, 25(8): 41 – 44. (in Chinese)

[9] Goseva-Popstojanova K, Mathur A P, Trivedi K S. Comparison of architecture-based software reliability models [A]. In Proc IEEE ISSRE'01[C]. Software Reliability Engineering, 2001. 22 – 31.

[10] 秦旭东,陈宗基.基于 Petri 网的容错计算机可靠性[J].计算机工程,2005,31(24):33 – 35.
Qin Xu-dong, Chen Zong-ji. Reliability of fault-tolerant computer systems based on petri nets[J]. Computer Engineering, 2005, 31(24): 33 – 35. (in Chinese)

[11] 唐明,张国平,张焕国.基于汉明纠错编码的 AES 硬件容错设计与实现[J].电子学报,2005,33(11):2013 – 2016.
Tang Ming, Zhang Guo-ping, Zhang Huan-guo. Automatic error correcting hardware implementation of AES algorithm[J]. Acta Electronica Sinca, 2005, 33(11): 2013 – 2016. (in Chinese)

[12] 孔德良,王少萍.可修系统的可用度分析方法研究[J].北京航空航大学学报,2002(2):129 – 132.
Kong De-liang, Wang Shao-ping. Study on availability analysis for repairable system[J]. Journal of Beijing University of Aeronautics and Astronautics, 2002, 28(2): 129 – 132. (in Chinese)

[13] Brown R E, Gupta S, Christie R D, et al. Distribution system reliability assessment using hierarchical Markov modeling[J]. IEEE Transactions on Power Delivery, 1996, 11(4): 1929 – 1934.

[14] 何国伟.软件可靠性[M].北京:国防工业出版社,1998. 90 – 92.

作者简介:

于　敏　女,1982 年生于辽宁葫芦岛.西南交通大学信息科学与技术学院博士研究生.研究方向为轨道交通监控系统可靠性分析,复杂监控系统可靠性研究.

E-mail: yugnm@163.com

何正友　男,1970 年生于四川自贡,教授,博士生导师.研究方向为监控系统可靠性分析、电力系统故障诊断、轨道交通电气自动化、电力系统及其自动化.