第22卷 第1期
1996年1月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 22, No.1
Jan., 1996

硬/软件可靠性增长模型

饶 岚

(清华大学自动化系 北京 100084)

李沛琼 姚一平 王占林

(北京航空航天大学三〇三教研室 北京 100083)

摘 要

在现有的描述计算机系统可靠性增长的模型当中，提出了一个描述硬/软件综合系统可靠性增长的二维跳跃马氏链模型. 该模型基于下面两个主要假设: 1) 在故障间隔期间内，硬、软件的故障率均是常数，硬(软)件的故障率仅在每个硬(软)件的故障恢复点处发生跳跃变化. 2) 同一时间发生多于一次故障的机会近乎为0. 本文推导了系统各种可用度指标的显式表达式，并给出计算复杂度的度量. 为了降低应用时求解的复杂程度，在当前硬、软件可靠性增长模型研究的基础上，提出了一种在工程应用中的简化计算方法.

关键词: 硬/软件系统，可靠性增长模型，马尔可夫过程.

1 引言

在计算机应用系统的研制初期，其可靠性与性能参数都不可能达到规定的要求，必须反复地进行TAAF，使其可靠性与性能逐步提高，直到满足要求为止，这就是可靠性增长过程. 现有的描述计算机系统可靠性增长的模型中绝大部分是单纯描述硬件或软件可靠性增长的，描述硬/软件综合系统可靠性增长的参数模型极为罕见[1]. 其实，在系统研制阶段，硬件、软件的可靠性是共同增长的. 明确这一点，并且积极跟踪系统在硬、软件综合之后的可靠性增长，具有重要的工程意义. 针对上述情况，本文在利用前人在硬、软件分别的可靠性增长研究成果的基础上，提出了一个两维跳跃马氏链系统行为模型. 文中第二部分给出假设及论证，第三部分给出一般模型及性能指标的推导，第四部分给出一个简化算法并计算一个实例，最后做出结论.

2 假设及论证

可靠性增长就是通过在研制和生产(设计)过程中为“强迫暴露”设计和制造(需求)的缺陷而使用一系列试验和新技术手段来系统地和永久地消除与之有关的故障机理，从而达到可靠性增长. 我们有理由相信，在硬/软件系统的综合测试阶段的某一设计水平

本文于1993年10月18日收到

上，即系统发生了某次故障，经采取纠正措施后去除了某种故障模式的阶段，由于系统所具有的潜在故障模式是一定的，到下一次维修前，系统的故障强度应该是不变的．延用一般认为硬、软件的故障过程及维修活动互相统计独立的假设[1]，我们提出下面的假设:

1）软件故障过程互相统计独立．从第 i 个软件故障模式被纠正时刻到下一次软件故障发生的时间依从一个参数为 λ_i，$i=0$，1，… 的负指数分布，不同阶段 λ_i 的跃度依从一增量过程.

2）硬件故障过程互相统计独立．从第 j 个硬件故障模式被纠正时刻到下一次硬件故障发生的时间依从一个参数为 β_j，$j=0$，1，… 的负指数分布，不同阶段 β_j 的跃度依从一增量过程.

3）纠正第 i 个软件故障模式的时间依从参数为 μ_i 的负指数分布，$i=1$，2，….

4）纠正第 j 个硬件故障模式的时间依从参数为 ξ_j 的负指数分布，$j=1$，2，….

5）同一时间发生多于一个故障的机会近乎于 0.

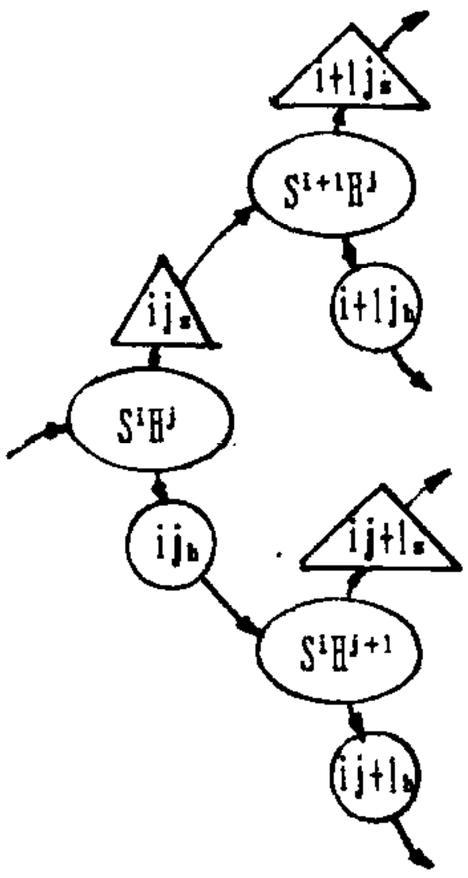

图1 系统部分状态转移图

其中，令系统从 $t=0$ 时刻开始增长试验，在随机的时刻 t_1，t_2，$\cdots t_N$ 相继发生了硬件故障，t_1'，t_2'，$\cdots t_m'$ 发生了软件故障．α_0，β_0 是初始阶段软、硬件子系统的故障率．令 Δt_j、$\Delta t_i'$ 分别为硬、软件子系统的故障纠正时间，则当 $t_i'+\Delta t_i'<t<t_{i+1}'$，且 $t_j+\Delta t_j<t<t_{j+1}$ 时，系统中软件子系统的故障率是 λ_i，硬件子系统的故障率是 β_j，可以画出系统的部分状态转移图如图 1 所示.

图 1 中，S^iH^j 为系统在发生了 i 次软件故障，j 次硬件故障后经过采取纠正措施后的工作状态，其中 $i=0$，1…，$j=0$，1，…;

ij_s 为系统在 S^iH^j 状态又发生软件故障的状态;

ij_h 为系统在 S^iH^j 状态时又发生硬件故障的状态.

3 模型推导

根据现场数据，对硬、软件故障模式发生时间的独立增量过程进行参数估计，可以得到对应的 λ_i，$i=0$，1，…，和 β_j，$j=0$，1，…，然后，根据假设，从第 i 个软件故障纠正时刻到下一次软件故障发生时间 U_i 的 CDF 是

$$P_r\{U_i\leqslant t\}=1-\exp(-\lambda_i t),\ i=0,\ 1,\ \cdots \tag{1}$$

从第 j 个硬件故障模式纠正时刻到下一次硬件故障发生时间 T_j 的 CDF 是

$$P_r\{T_j\leqslant t\}=1-\exp(-\beta_j t),\ j=0,\ 1,\ \cdots \tag{2}$$

对应的硬件故障模式纠正时间 W_j 和软件故障模式纠正时间 V_i 的 CDF 分别是

$$P_r\{W_j\leqslant t\}=1-\exp(-\xi_j t),\ j=0,\ 1,\ \cdots \tag{3}$$

$$P_r\{V_i\leqslant t)=1-\exp(-\mu_i t),\ i=0,\ 1,\ \cdots \tag{4}$$

则系统停留在 S^iH^j 状态的时间 $Y_{ij}=\min\{T_j,\ U_i)$的 CDF 为

$$P_r\{Y_{ij}\leqslant y\}=1-\exp[-(\lambda_i+\beta_j)y].\ i=0,\ 1,\ \cdots,\ j=0,\ 1,\ \cdots \tag{5}$$

这时，软件将在硬件之前发生故障的概率为[2]

$$p_{ij}=P_r[U_i<T_j\}=\frac{\lambda_i}{\lambda_i+\beta_j},\ i,\ j=0,\ 1,\ 2,\ \cdots \tag{6}$$

硬件在软件之前发生故障的概率为

$$q_{ij}=P_r\{T_j<U_i\}=\frac{\beta_i}{\lambda_i+\beta_j}.\ i,\ j=0,\ 1,\ 2,\ \cdots \tag{7}$$

半马氏核 $Q_{k,l}(t),\ k,\ l=S^iH^j,\ ij_s,\ ij_h,\ S^{i+1}H^j,\ S^iH^{j+1},\ i,\ j=0,\ 1,\ \cdots$ 为

$$\begin{aligned}
&Q_{S^iH^j,\ ij_s}(t)=p_{ij}\{1-\exp[-(\lambda_i+\beta_j)t]\},\\
&Q_{S^iH^j,\ ij_h}(t)=q_{ij}\{1-\exp[-(\lambda_i+\beta_j)t]\},\\
&Q_{ij_s,\ S^{i+1}H^i}(t)=1-\exp(-\mu_i t),\\
&Q_{ij_h,\ S^iH^{j+1}}(t)=1-\exp(-\xi_j t).
\end{aligned} \tag{8}$$

令 $X(t)$代表系统在 t 时刻所处的状态，$X(t)=S^iH^j$，ij_s，ij_h 等，则上述的等式描述了 $X(t)$过程的随机行为. 下面将推导系统的各种行为特征的表达式.

3.1　到某一增长阶段的第一转移时间

令 $T_{ij,\ mn}$ 是系统从 S^iH^j 状态到 S^mH^n 状态的第一转移时间，$G_{ij,mn}(t)$是其CDF，则 $G_{ij,mn}(t)$的更新方程为

$$\begin{aligned}
G_{ij,\ mn}(t)&=\sum_{k\in E}\int_0^t Q_{k,\ mn}(t-x)\mathrm{d}G_{ij,\ k}(x)\\
&=G_{ij,\ m-1n}(t)*Q_{S^{m-1}H^n,\ m-1n_s}(t)*Q_{m-1n_s,\ S^mH^n}(t)\\
&\quad+G_{ij,\ mn-1}(t)*Q_{S^mH^{n-1},\ mn-1_h}(t)*Q_{mn-1_h,\ S^mH^n}(t).
\end{aligned} \tag{9}$$

E 是状态空间，$G_{ij,\ ij}(t)=1$，* 为卷积，则(9)式的 $L-S$ 变换为

$$\begin{aligned}
\hat{G}_{ij,\ mn}(s)&=\hat{G}_{ij,\ mn-1}(s)\cdot\ \hat{Q}_{S^{m-1}H^n,\ m-1n_s}(s)\cdot\ \hat{Q}_{m-1n_s,\ S^mH^n}(s)\\
&\quad+\hat{G}_{ij,\ mn-1}(s)\cdot\ \hat{Q}_{S^mH^{n-1},\ mn-1_h}(s)\cdot\ \hat{Q}_{mn-1_h,\ S^mH^n}(s).
\end{aligned} \tag{10}$$

其中

$$\begin{aligned}
\hat{Q}_{S^{mn-1}H^n,\ m-1n_s}(s)&=\frac{\lambda_{m-1}}{s+\beta_n+\lambda_{m-1}},\\
\hat{Q}_{S^mH^{n-1},\ mn-1_h}(s)&=\frac{\beta_{n-1}}{s+\beta_{n-1}+\lambda_m},\\
\hat{Q}_{m-1n_s,\ S^mH^n}(s)&=\frac{\mu_{m-1}}{s+\mu_{m-1}},\\
\hat{Q}_{mn-1_n,S^mH^n}(s)&=\frac{\xi_{n-1}}{s+\xi_{n-1}}.
\end{aligned} \tag{11}$$

将(11)式代入(10)式，得

$$\hat{G}_{ij,\ mn}(s)=\hat{G}_{ij,\ m-1n}(s)a_{m-1n}+G_{ij,\ mn-1}(s)\cdot\ b_{mn-1}, \tag{12}$$

$$a_{m-1n}=\frac{\lambda_{m-1}\lambda_{m-1+}}{(s+\beta_n+\lambda_{m-1})(s+\mu_{m-1})}, \tag{13}$$

$$b_{mn-l}=\frac{\beta_{n-1}\xi_{n-1}}{(s+\beta_{n-1}+\lambda_m)(s+\xi_{n-1})}\ . \tag{14}$$

从上述三式可以推出

$$\hat{G}_{ij,\,i+1j}(s)=a_{ij}\ , \tag{15}$$

$$\hat{G}_{ij,\,ij+1}(s)=b_{ij}\ , \tag{16}$$

$$\hat{G}_{ij,\,i+1j+1}(s)=b_{ij}\ \cdot\ a_{ij+1}+a_{ij}\,b_{i+1j}\ . \tag{17}$$

另外 $\hat{G}_{ij,\,ij}(s)=1$. 原则上，可以用递推求出

$$\hat{G}_{00,\,mn}(s)=\hat{G}_{00,\,m-1n}(s)\cdot\ a_{m-1n}+\hat{G}_{00,\,mn-1}(s)\cdot b_{mn-1}=\sum_{k\in K}\cdots a_{ij}\,b_{i+1j}\ . \tag{18}$$

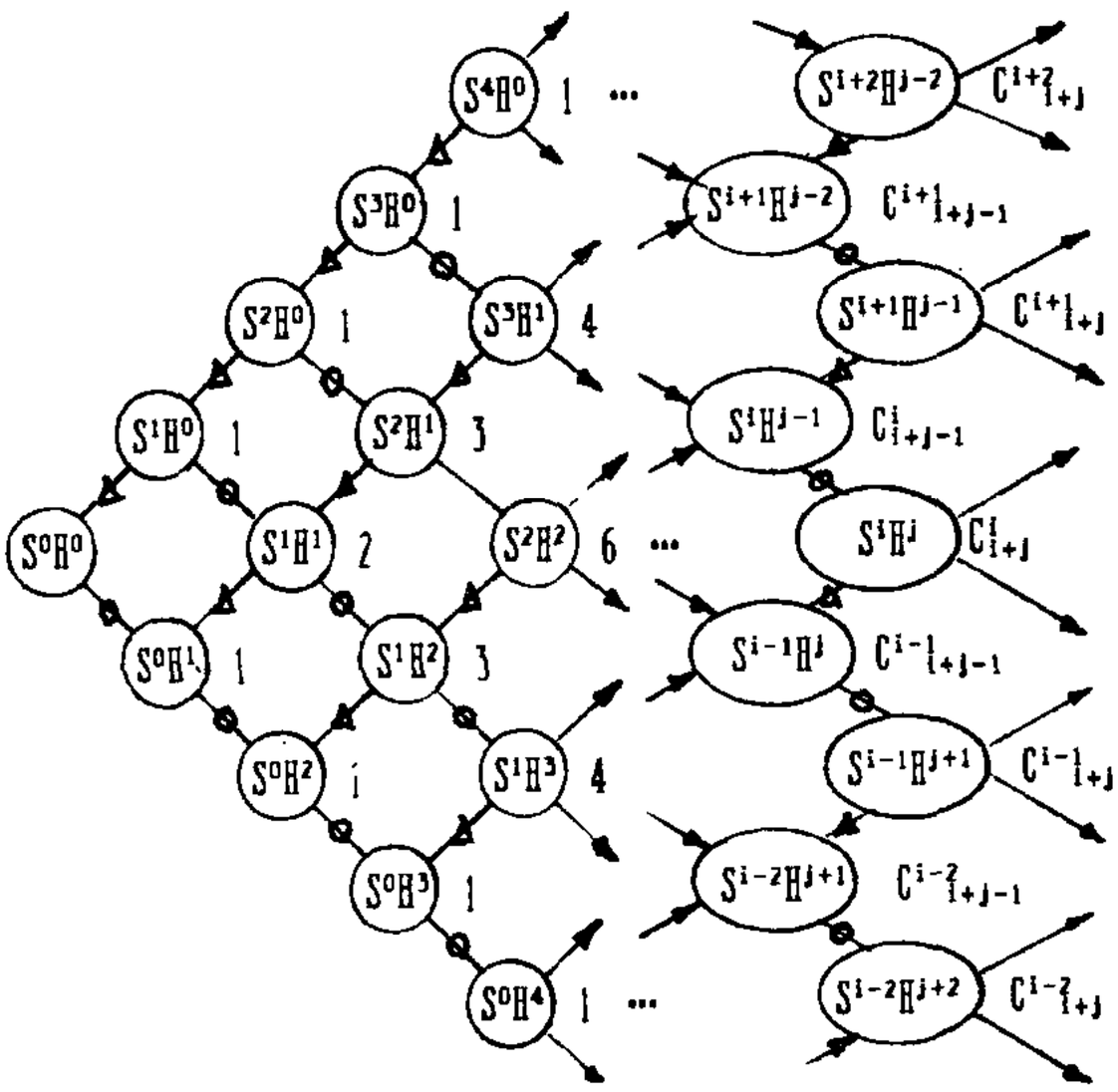

图2 从S^0H^0到S^iH^i的增长路径示意图

其中，k 为对应在 t_1，t_2，$\cdots t_n$，t_1'，t_2'，$\cdots t_m'$ 时系统在故障模式被纠正后的工作状态按时间顺序排列的一条增长路径，K 为所有可能的增长路径的集合. 图2即从 S^0H^0 到 S^iH^j 的增长路径示意图.

系统从开始增长试验到某一可靠性水平 S^mH^n 的累积第一转移时间 $T_{00,\,mn}$ 的CDF的L－S变换为

$$\hat{G}_{00,\,mn}(s)=\sum_{K}\prod_{l=1}^{m+n}\hat{G}_{l-1,\,l}(s). \tag{19}$$

如果状态 l 对应状态是 S^iH^j 状态，则

$$\hat{G}_{l,\,l+1}=\begin{cases}a_{ij}\ \text{当下次故障是软件故障},\\ b_{ij}\ \text{当下次故障是硬件故障}.\end{cases} \tag{20}$$

(19)式又可写为[2]

$$\hat{G}_{00,\,mn}(s)=\begin{cases}\displaystyle\sum_{K_1}\prod_{i=0}^{m-1}\lambda_i\mu_i\prod_{j=0}^{n}\beta_j\xi_j\prod_{l=1}^{2(m+n)}\frac{1}{S+r_1}=\sum_{K}A_{mn}\sum_{l=1}^{2(m+n)}C_l\,\frac{r_l}{s+r_l} & \text{当本次故障是软件故障},\\ \displaystyle\sum_{K_2}\prod_{i=0}^{m}\lambda_i\mu_i\prod_{i=0}^{n-1}\beta_j\xi_j\prod_{l=1}^{2(m+n)}\frac{1}{s+r_1}=\sum_{K}B_{mn}\sum_{l=1}^{2(m+n)}C_l\,\frac{r_l}{s+r_l} & \text{当本次故障是硬件故障}.\end{cases} \tag{21}$$

其中，

$$A_{mn}=\prod_{i=0}^{m-1}\lambda_i\mu_i\prod_{j=0}^{n}\beta_j\xi_j\ ,\ \ B_{mn}=\prod_{i=0}^{m}\lambda_i\mu_i\prod_{j=0}^{n-1}\beta_j\xi_j\ ,\ \ C_l=\frac{1}{r_l}\prod_{\substack{k=l\\k\neq l}}^{2(m+n)}\frac{1}{r_k-r_l}\ ,$$

$r_l=\lambda_i+\beta_j$，或 μ_i，ξ_j. 对上式做L－S反演，得

$$G_{00,\,mn}(t)=\begin{cases}\displaystyle\sum_{K_1}A_{mn}\sum_{l=1}^{2(m+n)}C_l(1-e^{-r_lt}),\\ \displaystyle\sum_{K_2}B_{mn}\sum_{l=1}^{2(m+n)}C_l(1-e^{-r_lt}).\end{cases} \tag{22}$$

3.2 系统可靠度

当系统在 t 时刻刚纠正故障模式，处于 $S^i H^j$ 状态，则系统的可靠度为

$$P_r\{\text{uptime} > t+x \mid X(t)=S^i H^j\}=\exp[-(\lambda_i+\beta_j)x] \tag{23}$$

3.3 系统可用度

令 $P_{00,ij}(t)$ 为系统从 $X(0)=S^0 H^0$ 状态直到 t 时刻仍处于某工作状态 $S^i H^j$ 的概率，即

$$P_{00,ij}(t)=P_r\{X(t)=S^i H^j \mid X(0)=S^0 H^0\},\ i=0,\ 1,\ \cdots,\ j=0,\ 1,\ \cdots \tag{24}$$

可以建立如下的更新方程

$$P_{00,ij}(t)=P_{ij,ij}(t)*G_{00,ij}(t).\ i=0,\ 1,\ \cdots,\ j=0,\ 1,\ \cdots \tag{25}$$

其中 $P_{ij,ij}(t)=\exp[-(\lambda_i+\beta_j)t]+Q_{ij,ij}*P_{ij,ij}(t)$，$i=0,1,\ \cdots,\ j=0,\ 1,\ \cdots$，则系统到$t$ 时刻的系统可用度为

$$A(t)=\sum_{ij=00}^{MN} p_{00,ij}(t)=\sum_{ij}^{MN}\exp[-(\lambda_i+\beta_j)t]*G_{00,ij}(t). \tag{26}$$

令 K_{ij} 为到 $S^i H^j$ 的路径集，其它定义同前，类似地可以推导 $A(t)$的显式表达式为

$$A(t)=\sum_{ij=00}^{MN}\sum_{Kij} A_{ij}(\text{或 } B_{ij})\left[\sum_{l=1}^{2(i+j)} C_l(1-e^{-\eta l t})-\sum_{l=1}^{2(i+j)+1} C_l(1-e^{\eta l t})\right]. \tag{27}$$

这里，$S^M H^N$ 指当硬、软件均不再存在无故障模式时系统的工作状态.

3.4 路径复杂性

从图 2 可以看出，到某一状态 $S^i H^j$ 的最多的可能路径数是在杨辉三角形中对应点的值 $C_{i+j}^i=\dfrac{(i+j)!}{i!j!}$. 因此(22)或(27)式的精确解似乎只有在 $i+j$ 比较小时才有意义，太大会引起组合爆炸，因此必须寻求简化方法. 但对于硬、软件的可靠性均按一定规律随时间的推移而增长的情况，由于时间相依，使得图 2 中许多可能的路径出现的概率极小，可以忽略不计. 从而使得需计算的可能路径数远远小于预期的增长路径数 C_{i+j}^i .

4　模型的工程应用

从上面的推导，我们可以看出，模型的计算还是比较复杂的. 下面利用一个工程实例给出一个实用的工程简化用法.

表 1 是某型飞机在对某项功能进行试飞阶段的硬、软件故障数据.

表　1

故障间隔 (日)	1	1	3	1	2	7	15	18	3
硬件故障 (个)	1			1		2	1	1	
软件故障 (个)		1	1		1				1

在这个例子中，硬、软件子系统纠正故障模式所用的时间远远小于故障间隔时间，因而硬、软件子系统分别的增长过程近乎于互相独立. 利用硬件增长模型拟合寻优程序和软件可靠性模型拟合寻优程序，分别求出拟合的硬、软件增长模型如下:

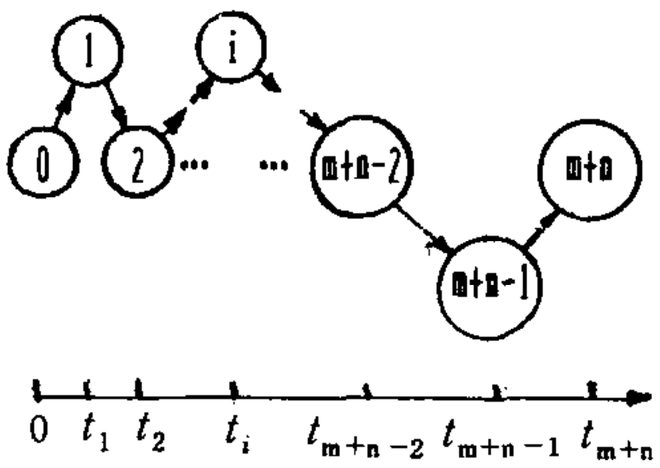

图 3 系统期望状态转移图

$$\begin{cases} E(t_j)=\left(\dfrac{j}{\lambda}\right)^{\frac{1}{\beta}}=\left(\dfrac{j}{1.002}\right)^{(1/0.457)}, \\ \overline{\beta}_j=\dfrac{1}{E(t_{j+1})}. \end{cases} \quad \text{(AMSAA)}$$

$$\begin{cases} \lambda_i=DK^{i-1}=0.693\times 0.373^{i-1}, i=0, 1,\cdots \\ E(t_i')=E(t_{i-1}')+\dfrac{1}{\lambda_i}. \end{cases} \quad \text{(Moranda)}$$

则 $E(t_i')$和 $E(t_j)$可以形成一个期望状态转移链如图 3 所示.

这时 $p_{ij}=1$ $q_{ij}=0$，或 $q_{ij}=1$ $p_{ij}=0$. 可以推得 $G_{00,mn}(t)$的表达式变为[3]

$$G_{00,mn}(t)=\sum_{l=1}^{2(m+n)} C_l(1-e^{-\Delta_l t}). \tag{28}$$

其中 $C_l=\prod_{\substack{k=1\\k\neq l}}^{2(m+n)}\dfrac{\Delta_k}{\Delta_k-\Delta_l}$，$\Delta_l$ 为相临状态间隔时间，它由期望故障间隔时间和故障纠正时间所共同决定. 可以得到$A(t)$的表达式

$$A(t)=\cdot\sum_{l=1}^{2(M+N)} C_l(1-e^{-\Delta_l t})-\sum_{l=1}^{2(M+N)+1} C_l(1-e^{-\Delta_l t}). \tag{29}$$

其中 $\Delta_{2M+2N+1}=\lambda_M+\beta_N$. 则可以得到系统按此增长速度增长 200 天时的系统的可用度曲线如图 4 所示. 图 5 是系统 MTBF 的变化曲线，图中第 2 条曲线是本模型计算的系统 MTBF 随时间变化的曲线，第 3 条曲线是[2] 模型计算的系统 MTBF 变化曲线. 第 1 条曲线是系统实际的 MTBF 变化曲线. 显然本模型的 MTBF 曲线拟合度要好些.

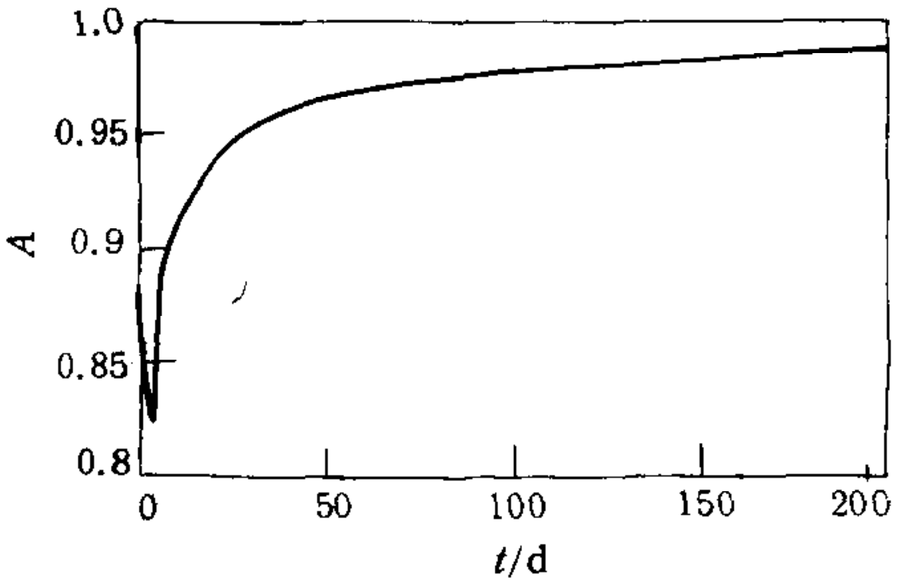

图 4 系统可用度曲线

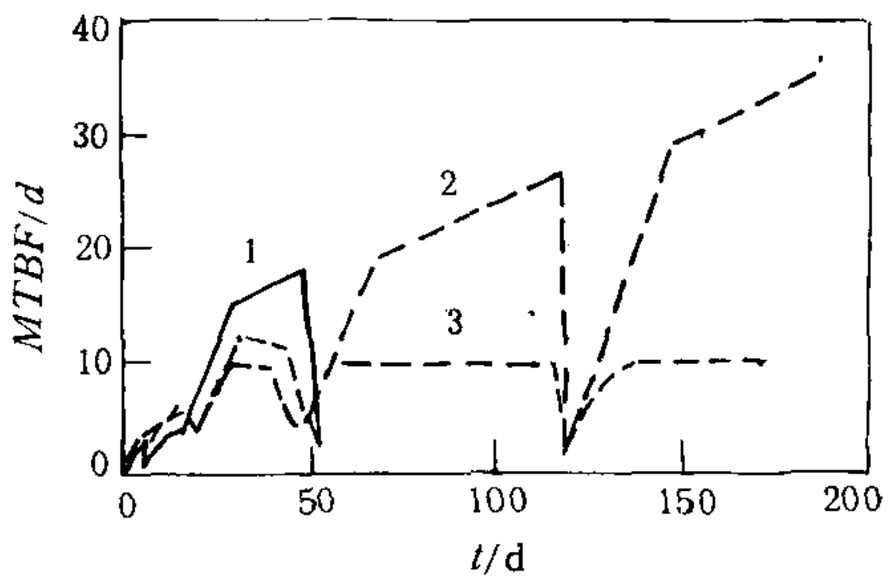

图5 系统MTBF 随时间变化曲线

5 结束语

本文推导了一个基于硬、软件可靠性增长过程中故障、维修跳跃马氏行为的硬／软件综合系统可靠性增长模型，并给出了几个重要的系统行为的度量指标的显式表达式. 这个模型的提出为考察在系统综合阶段的可靠性增长及硬、软件资源的权衡提供了一个有用的工具. 文中还给出一个工程简化算法，一个现场数据的应用结果表明该模型是可信的.

致谢：本文作者在此特别向曹晋华老师表示感谢!

参　考　文　献

[1] 饶岚，姚一平，李沛琼，王占林．硬、软件综合系统可靠性研究进展．航天控制，1994，**12**(1)：60 — 65.

[2] A. L. Goal. Models for Hardware-Software System Operational-Performance Evaluation. *IEEE transactions on Reliability*, 1981, **R－30**(3).

[3] 曹晋华，程侃．可靠性数学引论．北京科学出版社，1986.

MODELING THR RELIABILITY GROWTH OF HARDWARE/SOFTWARE

RAO LAN

(*Dept. of Automation, Tsinghua University Beijing* 100084)

LI PEIQIONG　YAO YIPING　WANG ZANLIN

(*Faculty* 303 *Beijing Univ. of Aero. & Astro., Beijing* 100083)

ABSTRACT

Very few studies have addressed the problem of modeling and evaluating the reliability growth of combined HW/SW system, A new modeling method is presented in this paper to treat those problems, which is based on two main assumptions:

1) The failure rates are constant during periods between failures, while changes on hardware (software) failure rates only occur in each hardware (software) failure recovered points.

2) More than 1 failures occur at one time is impossible.

Analytic expressions of system availability indexes are also presented, as well as computing complexity. An approximate method is given via field data example to reduce the complexity.

Key words: Hardware/Software, reliability growth, markov reward chain.

饶 岚 1965 年生. 1987 年毕业于北京航空学院自动控制系，1990 年在北京航空航天大学获硕士学位，1994 年在北京航空航天大学获博士学位. 现在清华大学自动化系做博士后. 近期研究领域及兴趣: 可靠性理论与技术、机器人和 *CIMS*.

李沛琼 1930 年生. 1955 年毕业于北京航空学院. 现任北京航空航天大学 303 教研室教授. 主要从事系统可靠性与余度技术方面的教学与科研工作. 近期研究领域及兴趣: 可靠性理论与技术、流体控制与飞机操纵系统.