文章编号:1001-506X(2010)12-2584-06

基于可用度模型的故障预测与健康管理方法

王昊天[1],石 健[2]

(1. 北京航空航天大学电子信息工程学院,北京 100191;
2. 北京航空航天大学自动化科学与电气工程学院,北京 100191)

摘 要:为了将故障预测与健康管理(prognostics and health management,PHM)技术应用到工程实践中,提出了基于可用度模型的 PHM 方法。首先通过广义随机 Petri 网(generalized stochastic Petri nets,GSPN)和连续马尔科夫链(continuous time Markov chain,CTMC)建立基本单元的软硬件可用度模型和健康状态转换图,通过求解微分方程得到基本单元软硬件的可用度数值。然后综合软硬件之间的故障相关性建立基本单元的完整可用度模型,并利用事件调度仿真机制得到其可用度的解。最后将基本单元故障模型同通用的可修系统稳态可用度模型对比,得到"可用度-故障率-维修率"形式的 PHM 计算模型,并以此作为工程应用中 PHM 分析的有效手段。

关键词:故障预测与健康管理;广义随机 Petri 网;连续马尔可夫链;可用度

中图分类号:TP 202;V 215.7 **文献标志码**:A **DOI**:10.3969/j.issn.1001-506X.2010.12.19

Method of prognostics and health management based on availability model

WANG Hao-tian[1], SHI Jian[2]

(1. *School of Electronics and Information Engineering, Beihang Univ., Beijing 100191, China*;
2. *School of Automation Science and Electrical Engineering, Beihang Univ., Beijing 100191, China*)

Abstract: To apply prognostics and health management (PHM) technology in realistic engineering, a method of PHM based on availability model is proposed. Firstly, the hardware and software model of the basic unit and its healthy state transition diagram are established by generalized stochastic Petri nets (GSPN) and continuous time Markov chain (CTMC). The availability of the basic unit could be calculated by sloving differential equations. Combining the relationship of the hardware and software faults, the unit availability model is obtained. Based on event scheduling simulation, the result of the basic unit availability is achieved. Finally, a comparison is made between the failure model of the basic uint and the model of repairable system theory, the "availability-failure rate-maintenance rate" calculation model is obtained and is applied for PHM analysis in engineering.

Keywords: prognostics and health management (PHM); generalized stochastic Petri nets (GSPN); continuous time Markov chain (CTMC); availability

0 引 言

故障预测与健康管理(prognostics and health management,PHM)技术,能够使设计人员、维护人员和生产人员等动态地理解一个系统或产品的健康状态,从而帮助他们处理各种信息,对生命周期管理做出决定。在国外,PHM 技术已经在航空航天等国防项目中得到了实际的工程应用,目前正在积极地将 PHM 技术引入到民用工业生产中;而在国内,PHM 技术只是停留在方案论证和框架研究阶段[1-4]。

广义随机 Petri 网(generalized stochastic Petri nets,GSPN)以其并行、异步操作和状态可达等特性,被广泛应用到计算机及通信系统的性能评价模型中[5-10]。但是在 PHM 分析领域,基于 GSPN 模型的方法并没有得到工程应用。实际上,作为图形工具,GSPN 除了具有类似流程图、框图和网图的可视化描述功能外,还可以通过标记(token)的流动来模拟实际系统的动态运行行为,这对于描述一个复杂系统的故障/维修过程十分有利[11]。同时作为数学工具,GSPN 模型可以通过建立状态转移图、代数方程和其他方法(如仿真方法)来解算,将 GSPN 模型应用到 PHM 研究中,可以简化系统 PHM 的模型建立和求解过程。

收稿日期:2010-01-13;**修回日期**:2010-04-12。

基金项目:中国国防重点基金(51419020404HK0150);国家自然科学基金(60879024)资助课题

作者简介:王昊天(1983-),男,博士研究生,主要研究方向为航空电子、数据通信、QOS 性能保证、故障预测与健康管理、实时性与可靠性。E-mail:qi1983xiong@ee.buaa.edu.cn

本文将待测系统的可用度模型应用到 PHM 技术中,考虑系统各个构件的相关性,应用 GSPN 建模,并综合利用连续马尔可夫链(continuous time Markov chain, CTMC)和事件调度仿真方法[12]求解 GSPN 模型的可用度值。通过与基本可修理论中的模型[13]对比,建立"可用度-故障率-维修率"形式的 PHM 计算模型,对目标系统进行 PHM 操作。

1　基本单元的可用度模型

GSPN 在 Petri 网的基本要素(位置、变迁、弧和标记)外,定义了三个特有元素:时间变迁、立即变迁和禁止弧。

(1) 每一个时间变迁服从负指数分布,表示变迁元素从使能到点火的延时。当几个变迁元素同时使能时,具有最小延时的变迁元素最先点火。

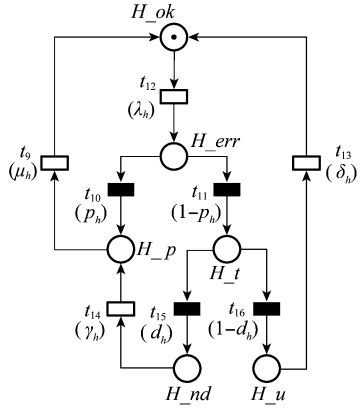
(2) 立即变迁定义为零延迟时间点火,其优先级高于时间变迁。多个立即变迁之间以发生概率值选择流向。

(3) 禁止弧所连接的位置,其原可实施条件变为不可实施条件,原不可实施条件变为可实施条件,且在相连的变迁实施时,没有标记从相连的位置中移出。

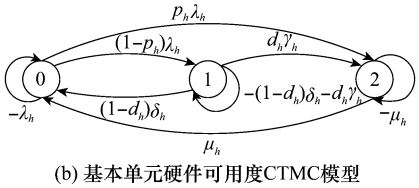
任何系统均可以分解为若干基本单元,而基本单元是一个典型的结构相关的子系统。首先分别建立基本单元的硬件可用度模型和软件可用度模型[14-15],再综合软硬件之间的故障关联性,得到基本单元的整体可用度模型。

1.1　基本单元的硬件可用度模型

图 1(a)给出了基本单元硬件可用度的 GSPN 模型,其中,*H_ok* 表示单元硬件的正常运行状态;*H_err* 表示单元硬件的异常状态;*H_t* 表示单元硬件的临时故障状态;*H_p* 表示单元硬件的永久故障状态;*H_u* 表示单元硬件的可修复故障状态;*H_nd* 表示单元硬件不可修复故障状态。



(a) 基本单元硬件可用度GSPN模型



(b) 基本单元硬件可用度CTMC模型

图 1　硬件可用度模型

单元硬件处于正常运行状态 *H_ok* 下,由于某种原因,以 λ_h 的故障率发生故障 *H_err*。硬件发生的故障 *H_err* 可能以概率 P_h 成为需要人工维修的永久故障 *H_p*,也可能以概率为 $1-P_h$ 成为临时故障 *H_t*。硬件发生永久故障 *H_p* 时需要人工进行维修,维修率为 μ_h。硬件发生的临时故障 *H_t* 可能以概率 $1-d_h$ 转化为自消除的故障 *H_u*,*H_u* 再以 δ_h 的自恢复率修复;临时故障 *H_t* 也可能以概率 d_h 转化为不可自修复临时故障 *H_nd*,*H_nd* 再以 γ_h 转化为永久故障 *H_p*。

鉴于 GSPN 模型同构于 CTMC,将基本单元硬件可用度 GSPN 模型等价于图 1(b)所示的 CTMC 模型,其中,状态 0 为单元硬件处于健康工作状态;状态 1 为临时故障状态;状态 2 为单元硬件处于需要人工维修的永久故障状态。

由图 1(b)可以得到基本单元硬件可用度模型的状态转移速率矩阵为

$$\boldsymbol{V}_H=\begin{bmatrix}-\lambda_h & \lambda_h(1-P_h) & \lambda_h P_h\\ (1-d_h)\delta_h & -(1-d_h)\delta_h-d_h\gamma_h & d_h\gamma_h\\ \mu_h & 0 & -\mu_h\end{bmatrix} \tag{1}$$

由 $\boldsymbol{V}_H$ 可以得到基本单元硬件 CTMC 模型的微分方程

$$\begin{cases}P'_{H0}(t)=-l_hP_{H0}(t)+(1-d_h)d_hP_{H1}(t)+m_hP_{H2}(t)\\ P'_{H1}(t)=l_h(1-P_h)P_{H0}(t)-[(1-d_h)d_h+d_hg_h]P_{H1}(t)\\ P'_{H2}(t)=l_hP_hP_{H0}(t)+d_hg_hP_{H1}(t)-m_hP_{H2}(t)\end{cases}$$

如果时刻 $t=0$ 时单元硬件处于健康工作状态,即初始条件为 $P_{H0}(0)=1$,$P_{H1}(0)=0$,$P_{H2}(0)=0$,则可求得硬件各状态的稳态概率

$$P_{H0}=\frac{\mu_h(1-d_h)\delta_h+\mu_hd_h\gamma_h}{\lambda_hd_h\gamma_h+\lambda_h\delta_h(1-d_h)P_h+\mu_h(1-d_h)\delta_h+\mu_hd_h\gamma_h+\mu_h\lambda_h(1-P_h)}$$

$$P_{H1}=\frac{\mu_h\lambda_h(1-P_h)}{\lambda_hd_h\gamma_h+\lambda_h\delta_h(1-d_h)P_h+\mu_h(1-d_h)\delta_h+\mu_hd_h\gamma_h+\mu_h\lambda_h(1-P_h)}$$

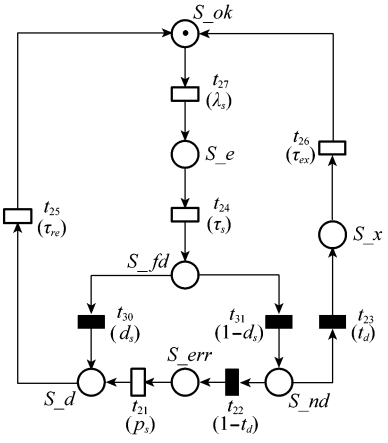
$$P_{H2}=\frac{\lambda_hd_h\gamma_h+\lambda_h\delta_h(1-d_h)P_h+\mu_h(1-d_h)\delta_h}{\lambda_hd_h\gamma_h+\lambda_h\delta_h(1-d_h)P_h+\mu_h(1-d_h)\delta_h+\mu_hd_h\gamma_h+\mu_h\lambda_h(1-P_h)}$$

从而得到基本单元硬件可用度为硬件可以正常工作的可用度和,即健康状态和临时故障状态的稳态概率之和,$A_H=P_{H0}+P_{H1}$。

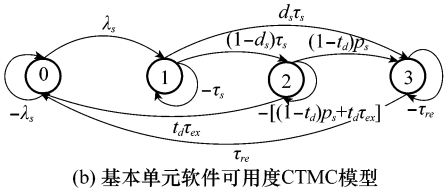
1.2　基本单元的软件可用度模型

图 2(a)给出了基本单元软件可用度的 GSPN 模型,其中,*S_ok* 表示软件的正常工作状态;*S_e* 表示软件处于异常状态;*S_fd* 表示系统对软件错误的检测状态;*S_d* 表示系统检测到软件故障状态;*S_nd* 表示系统未检测到软件故障状态;*S_err* 表示未检测到的软件错误转化为软件故障的中间状态;*S_x* 表示软件的临时故障状态。

软件运行在正常状态 *S_ok* 下,由于某种原因,导致软件处于异常 *S_e*,使得软件的性能下降。对于软件的异常,系统以 τ_d 的速率完成对软件异常的检测 *S_fd*,软件的异常可能以概率 d_s 被检测到,成为软件的故障 *S_d*,需要重启软件,重启速率为 τ_{re};软件的异常也可能以概率 $1-d_s$ 未被系统检测到至 *S_nd*,未被检测到的软件异常可能以概率 $1-t_d$ 最终以速率为 p_s 转化为软件的故障 *S_d*,也可能以概率 t_d 经过一定的时间以恢复率为 τ_{ex} 恢复正常。



(a) 基本单元软件可用度GSPN模型



(b) 基本单元软件可用度CTMC模型

图 2 软件可用度模型

同基本单元硬件可用度模型一样，基本单元软件可用度 GSPN 模型等效于图 2(b)所示的 CTMC 模型，其中，状态 0 表示软件处于健康工作状态；状态 1 表示软件发生异常的状态；状态 2 表示软件处于未检测的异常状态；状态 3 表示软件处于检测到的故障状态，需要重启软件。

由图 2(b)可以得到基本单元软件可用度的状态转移率矩阵为

$$\boldsymbol{V_S}=\begin{bmatrix}-\lambda_s & \lambda_s & 0 & 0\\ 0 & -\tau_d & (1-d_s)\tau_d & d_s\tau_d\\ t_d\tau_{ex} & 0 & -[(1-t_d)p_s+t_d\tau_x] & (1-t_d)\tau_{ex}\\ \tau_{re} & 0 & 0 & -\tau_{re}\end{bmatrix} \tag{2}$$

由 $\boldsymbol{V_S}$可以得到基本单元软件 CTMC 模型的微分方程为

$$\begin{cases}P'_{S0}(t)=-\lambda_sP_{S0}(t)+t_d\tau_{ex}P_{S2}(t)+\tau_{re}P_{S3}(t)\\ P'_{S1}(t)=\lambda_sP_{S0}(t)-\tau_dP_{S1}(t)\\ P'_{S2}(t)=(1-d_s)\tau_dP_{S1}(t)-[(1-t_d)p_s+t_d\tau_{ex}]P_{S2}(t)\\ P'_{S3}(t)=d_s\tau_dP_{S1}(t)+(1-t_d)\tau_{ex}P_{S2}(t)-\tau_{re}P_{S3}(t)\end{cases}$$

如果 $t=0$ 时软件处于健康工作状态，即初始条件为 $P_{S0}(0)=1$，$P_{S1}(0)=0$，$P_{S2}(0)=0$，$P_{S3}(0)=0$，则可求得软件各状态的稳态概率为

$$P_{S0}=\frac{\tau_{re}\tau_d[t_d\tau_{ex}+(1-t_d)p_s]}{(\lambda_s\tau_d+\tau_{re}\tau_d+\tau_{re}\lambda_s)[t_d\tau_{ex}+(1-t_d)p_s]+(1-d_s)\tau_d\lambda_s(\tau_{re}-t_d\tau_{ex})}$$

$$P_{S1}=\frac{\tau_{re}\lambda_s[t_d\tau_{ex}+(1-t_d)p_s]}{(\lambda_s\tau_d+\tau_{re}\tau_d+\tau_{re}\lambda_s)[t_d\tau_{ex}+(1-t_d)p_s]+(1-d_s)\tau_d\lambda_s(\tau_{re}-t_d\tau_{ex})}$$

$$P_{S2}=\frac{(1-d_s)\tau_{re}\lambda_s}{(\lambda_s\tau_d+\tau_{re}\tau_d+\tau_{re}\lambda_s)[t_d\tau_{ex}+(1-t_d)p_s]+(1-d_s)\tau_d\lambda_s(\tau_{re}-t_d\tau_{ex})}$$

$$P_{S3}=\frac{d_s\lambda_s[t_d\tau_{ex}+(1-t_d)p_s]+(1-d_s)\tau_d\tau_{ex}\lambda_s}{(\lambda_s\tau_d+\tau_{re}\tau_d+\tau_{re}\lambda_s)[t_d\tau_{ex}+(1-t_d)p_s]+(1-d_s)\tau_d\lambda_s(\tau_{re}-t_d\tau_{ex})}$$

从而可以得到基本单元的软件可用度为软件可以正常工作的可用度和，即除软件故障状态之外其他状态的稳态概率之和，$A_S=P_{S0}+P_{S1}+P_{S2}$。

1.3 基本单元可用度模型

基本单元可用度模型须考虑软硬件可用度模型的故障相关性，即：单元硬件在运行过程中的临时故障有可能导致软件运行的异常，在极端的情况之下，甚至可能导致软件的故障，因此在单元的可用度模型中应当体现这种故障传递的过程。综合基本单元的软硬件可用度模型，并考虑单元硬件故障对软件运行的影响，可以得到基本单元的可用度模型，如图 3 所示。

图 3 中 *Prop* 表示硬件故障向软件传播的状态；*FL* 表示基本单元的完全故障状态。当单元处于临时故障状态 *H_u* 时，单元硬件一方面以变迁 t_{13} 所标示的速率进行自恢复，在此期间，硬件的临时故障可能要传播到软件，即位置 *Prop* 中产生标记，将导致软件发生下列可能的行为：

(1) 如果软件处于正常运行状态 *S_ok*，传播到软件的故障导致软件运行发生异常，即位置 *S_e* 中产生标记，此后软件以独立的故障转移过程进行处理；

(2) 如果软件处于异常的运行状态，传播到软件的故障将直接导致软件发生可检测到的故障 *S_d* 或未检测到的故障 *S_nd*。

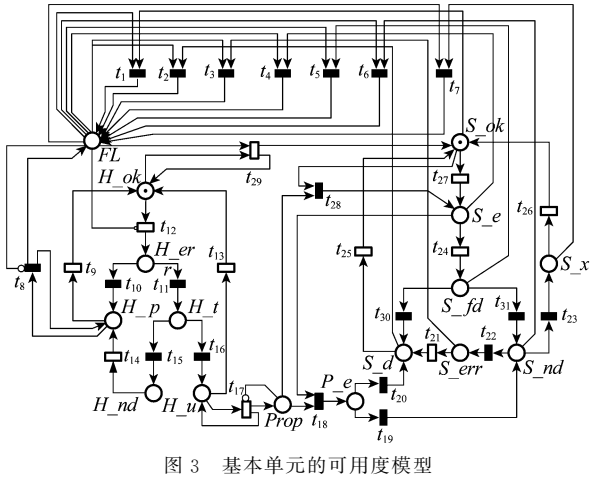


图 3 基本单元的可用度模型

当单元硬件发生需要人工维修的故障 *H_p* 时，单元系统处于完全的故障状态，即位置 *FL* 中产生标记，模型通过

位置 t_1-t_7 清除软件的所有标记。但系统完成维修后,硬件恢复完好状态 H_ok,系统以 t_{29} 所标示的速率重启机器,清除位置 FL 中的标记,单元系统的软硬件进入正常工作状态。

由于基本单元 GSPN 模型的复杂性,很难建立起基本单元的 CTMC,更难于对 CTMC 求解。为了得到基本单元的可用度,需要采用仿真方法。为此,采用了基于事件调度的仿真机制,并取得了很好的效果[12]。通过对基本单元 GSPN 模型进行仿真,可以得到单元的有效标识(出现概率较高的标识,是其他标识概率的 10 倍以上),如表 1 所示。

表 1　基本单元 GSPN 模型有效状态

标识	H_ok	H_e	H_t	H_u	H_p	H_nd	$Prop$	P_e	S_err	S_nd	S_fd	S_x	S_e	S_ok	S_d	FL
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

在表 1 中,标识 1 为单元软、硬件均处于完好的状态;标识 2 中单元的硬件处于完好的状态,而基本单元的软件发生异常,在此状态之后系统将对软件进行自检测;标识 3 表示单元的硬件处于完好的状态,而基本单元软件发生故障,此时需要重新启动软件;标识 4 中,基本单元的软件发生的异常不会影响系统的正常运行,基本单元软件经过一定的时间将恢复正常;标识 5 表示基本单元的硬件发生了临时故障,而软件运行良好,在此状态下,基本单元的硬件的临时故障可能传递到软件中,导致软件运行的异常,另一种可能为硬件的临时故障尚未传递到软件即恢复正常;标识 6 表示单元硬件完成了维修,等待重启的状态;标识 7 表示基本单元的硬件发生故障,需要进行维修的状态。

当基本单元处于标识 3、标识 6 和标识 7 时,基本单元无法正常工作,因此定义这三个标识为基本处理单元的故障状态。基本单元所处的其余标识,虽然软、硬件可能发生异常,但并不影响基本单元功能的实现,仍可正常工作,称这些状态为基本单元的健康状态。

表 2 所示为基本处理单元硬件的故障率对基本单元的可用度的影响。

表 2　硬件故障对基本单元可用度的影响

MTTF/h	$\lambda_h/(\times10^{-7}/s)$	A_W	$P(stat=1)$	$P(stat=3)$	$P(stat=6)$	$P(stat=7)$
200	13.8	0.995 812	0.995 434	0.000 110	0.000 302	0.003 776
400	6.94	0.997 944	0.997 719	0.000 094	0.000 153	0.001 809
600	4.63	0.998 562	0.998 361	0.000 091	0.000 105	0.001 242
800	3.47	0.998 933	0.998 791	0.000 084	0.000 075	0.000 908
1 000	2.77	0.999 084	0.998 961	0.000 079	0.000 063	0.000 774
1 200	2.31	0.999 260	0.999 135	0.000 079	0.000 040	0.000 621
1 400	1.98	0.999 350	0.999 243	0.000 080	0.000 044	0.000 526
1 600	1.74	0.999 408	0.999 312	0.000 080	0.000 039	0.000 473

表 2 中,MTTF 表示平均无故障时间(mean time to failure, MTTF)[13];λ_h 表示维修率[13];A_W 表示基本单元的可用度;$P(stat=1)$、$P(stat=3)$、$P(stat=6)$和 $P(stat=7)$ 分别表示单元处于标识 1、标识 3、标识 6 和标识 7 的概率。由表 2 可知,随着单元硬件的故障率的降低,单元的可用度不断升高。同时软件的故障概率(即单元处于标识 3 的概率)不断降低,最终稳定在一个常值。这表明,在硬件的故障率较高时,软件的故障同时受自身故障和硬件故障的影响。随着硬件故障率的降低,硬件的临时故障传播到软件的概率也降低,软件的故障主要由软件自身故障率造成。

表 3 给出了基本处理单元软件的故障率对基本单元的可用度的影响。

表 3　软件故障对基本单元可用度的影响

MTTF/h	$\lambda s/(\times10^{-7}/s)$	A_W	$P(stat=1)$	$P(stat=3)$	$P(stat=6)$	$P(stat=7)$
200	27.7	0.999 144	0.998 808	0.000 361	0.000 039	0.000 456
400	13.8	0.999 290	0.999 108	0.000 190	0.000 040	0.000 480
600	9.26	0.999 359	0.999 261	0.000 127	0.000 039	0.000 475
800	6.94	0.999 400	0.999 290	0.000 097	0.000 039	0.000 464
1 000	5.56	0.999 401	0.999 305	0.000 077	0.000 040	0.000 482
1 200	4.63	0.999 418	0.999 331	0.000 067	0.000 040	0.000 475
1 400	3.97	0.999 439	0.999 356	0.000 058	0.000 039	0.000 464
1 600	3.47	0.999 443	0.999 364	0.000 053	0.000 039	0.000 466

由表 3 可知，随着软件故障率的降低，基本单元系统的故障发生概率降低，基本单元系统的可用度不断提高。根据上述分析可知，基本单元的可用度并不是硬件可用度和软件可用度的简单相乘，而是加入了软硬件之间的故障相关性后，综合软硬件自身的可用度值而得到的。事实上，基本单元的可用度要低于软、硬件可用度的乘积，如图 4 所示。

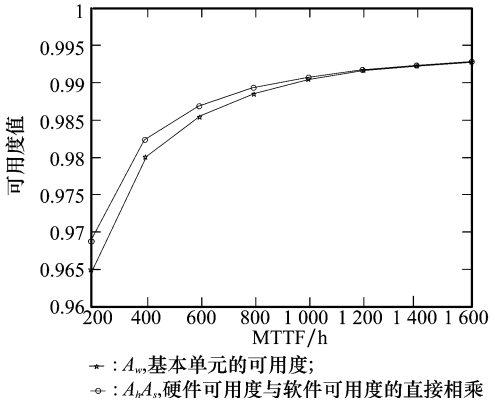


图 4 基本单元可用度

2 PHM 计算模型

根据文献[13]，图 5(a)给出了基本可修系统的 CTMC 模型。若初始时刻 $t=0$ 时，系统处于正常工作状态，则通过求解 CTMC 状态转移方程，可以得到基本可修系统的可用度 P_0 和不可用度 P_1。

$$\begin{cases} P_0(t) = \dfrac{\mu}{\lambda+\mu} + \dfrac{\lambda}{\lambda+\mu}\mathrm{e}^{-(\lambda+\mu)t} \\ P_1(t) = \dfrac{\lambda}{\lambda+\mu} - \dfrac{\lambda}{\lambda+\mu}\mathrm{e}^{-(\lambda+\mu)t} \end{cases} \tag{3}$$

式中，μ 表示维修率；λ 表示故障率。

当 $t\to\infty$ 时，得到可修系统的稳态可用度 A_t

$$A_t = \frac{\mu}{\lambda+\mu} \tag{4}$$

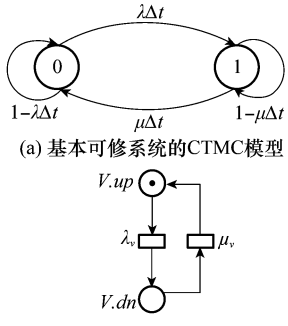


图 5 基本单元 PHM 计算模型

为了简化基本处理单元的可用度分析，将基本单元可用度模型定义为两种状态：故障状态和健康状态。即表 3 中，基本单元无法正常工作的状态(标识 3、标识 6 和标识 7)，定义为故障状态；而基本单元处于其余状态时，虽然可能会有异常发生，但并不影响基本单元的正常运行，这里定义为健康状态。从而得到了与图 5(a)同构的基本单元用于 PHM 计算的 GSPN 模型，如图 5(b)所示。其中，$V.up$ 表示基本单元处于健康状态，$V.dn$ 表示基本单元处于故障状态。

与可修系统稳态可用度 A_t 比较，可得单元稳态可用度 A_W 的计算公式

$$A_W = \frac{\mu_d}{\lambda_d+\mu_d} \tag{5}$$

由对基本单元可用度模型的有效状态的分析可知，基本处理单元的故障状态包括标识 3、标识 6 和标识 7。其中，标识 3 表示系统软平均历时时间为 $1/\tau_{re}$，标识 6 平均历时时间为 $1/\tau_{re}$，标识 7 平均历时时间为 $1/\mu_h$，则单元处于不可用状态的平均历时时间为

$$T = \frac{[P(stat=3)+P(stat=6)]/\tau_{re} + P(stat=7)/\mu_h}{P(stat=3)+P(stat=6)+P(stat=7)}$$

取 $\mu_d=1/T$ 作为简化后基本单元的维修率，根据 A_W 的表达式可以确定简化后基本处理单元的等效故障率为

$$\lambda_d = \frac{(1-A_W)}{A_W}\times\mu_d \tag{6}$$

至此得到了基本单元可用度简化后的“可用度-故障率-维修率”形式的 PHM 计算模型，可将此计算模型作为基本模块应用到复杂系统的 PHM 分析中。

(1) 由 GSPN 模型得到可用度，再从系统整体有效状态获得维修率，从而预测系统各个构件的故障率；

(2) 系统升级时，在给定故障率和维修率指标下，得到子系统的可用度，以可用度的数值选取较优的备选子系统方案；

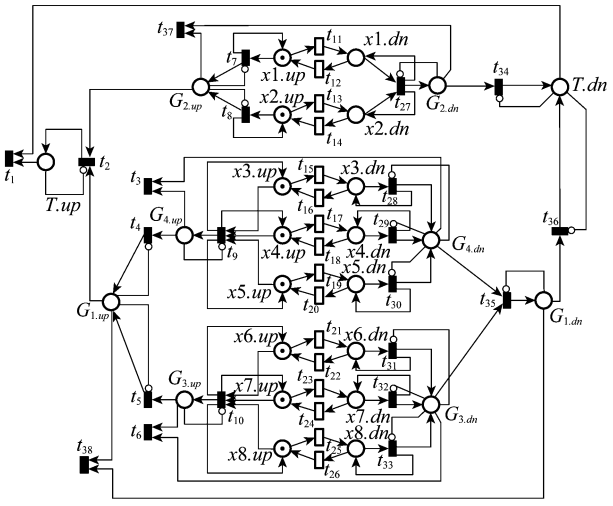
(3) 从 GSPN 模型获得可用度值后，再根据给定的故障率指标，来确定设备的维修周期。

3 工程应用举例

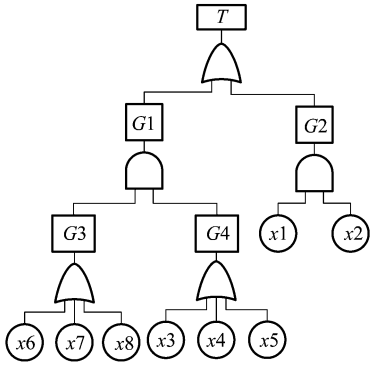
在本节中，以某工程项目中的一个子系统为例，说明本文提出的基于可用度模型的 PHM 方法的应用过程。本例的目的是在给定故障率和维修周期下，得到构件和子系统整体的可用度值，再以可用度的值，来检验给定维修周期的合理性。图 6(a)给出了该子系统的 GSPN 模型，图 6(b)给出该子系统的逻辑连接模型。

在图 6(b)中，小圆圈表示一个构件，对应图 6(a)中的 $x.up$ 和 $x.dn$；方框表示一个功能系统，对应图 6(a)中的 $G.up$ 和 $G.dn$；矩形表示该子系统自身，对应图 6(a)中的 $T.up$ 和 $T.dn$；三角连接符表示“逻辑与”关系，即为图 6(a)中的串联关系；拱形连接符表示“逻辑或”关系，即为图 6(a)中的并联关系。

将该子系统 T 的可用度模型分解为构件 x 的可用度模型的组合，而每个 x 即为第三部分提出的 PHM 计算模型，经过事件调度仿真，得到子系统的有效状态，如表 4 所示。



(a) 子系统的GSPN可用度模型



(b) 子系统的逻辑连接可用度模型

图 6　子系统的可用度模型

表 4　子系统的有效状态

标识	X1	X2	X3	X4	X5	X6	X7	X8
1	1	1	1	1	1	1	1	1
2	1	1	1	0	1	1	0	1
3	1	1	1	1	1	1	0	1
4	1	1	1	0	1	0	1	1

该子系统的额定参数如表 5 所示(互为冗余的构件合并在一起给出)。

表 5　构件的 PHM 数值

名称	$x1,x2$	$x3,x6$	$x4,x7$	$x5,x8$
$\lambda_d/(10^{-4}/\mathrm{s})$	2.77	3.74	3.74	2.57
$\mu_d/(10^{-4}/\mathrm{s})$	1.157	27.8	1.157	1.157
A_W	0.295 153	0.881 420	0.236 267	0.310 437

将表 5 中的构件可用度值,代入子系统 GSPN 模型中,利用事件调度仿真,可得各个构件和整体的可用度值,如表 6所示。

表 6　子系统的可用度数值

名称	状态 1	状态 2	状态 3	状态 4
A_W	0.940 314	0.021 347	0.001 064	0.024 362

由表 6 可知构件 $x4$ 和 $x7$ 的故障影响了子系统整体的可用度,应将其维修率提升。反复使用上述方法,最终确定将 $x4$ 和 $x7$ 的维修率提升一倍,即为 $2.314\times10^{-4}/\mathrm{s}$,可使系统的整体可用度由原来的约 94%提升为约 98%,减小了故障的发生,提升了子系统的健康状态时间。

4　结　论

本文为 PHM 技术提供了一种基于可用度模型的方法,综合考虑系统各个构件的相关性,而不是机械地将各个构件的故障率直接相加,提高了 PHM 模型的真实程度;综合利用 GSPN、CTMC 和事件调度方法获取可用度的值,降低了求解的复杂程度;提供了"可用度-故障率-维修率"形式的 PHM 计算模型,可应用到实际的国防工程中,扩大了 PHM 技术的应用范围。

参考文献:

[1] John W S, Mark A K, Timothy J W. IEEE standards for prognostics and health management [J]. *IEEE A&E Systems Magazine*, 2009,24(9):34 - 41.

[2] Eli D. Introduction to the special section on prognostics and health management [J]. *IEEE Trans. on Reliability*, 2009,58(2):262 - 263.

[3] 曾声奎, Michael G P, 吴际. 故障预测与健康管理(PHM)技术的现状与发展[J]. 航空学报, 2005,26(5):626 - 632.

[4] 孙博, 康锐, 谢劲松. 故障预测与健康管理系统研究和应用现状综述[J]. 系统工程与电子技术, 2007,29(10):1762 - 1767. (Sun Bo, Kang Rui, Xie Jinsong. Research and application of the prognostic and health managment systems [J]. *Systems Engineering and Electronics*,2007,29(10):1762 - 1767.)

[5] Wang H. Modeling and simulation of avionics blueprint based generalized stochastic Petri nets[C]// He F, Xiong H. *IEEE International Conference on Information and Automation*,2008:1560 - 1565.

[6] Schneeweiss W G. Review of Petri net picture book and Petri nets for reliability modeling [J]. *IEEE Trans. on Reliability*, 2006,55(2):391 - 392.

[7] Mark B, Chirs M, Antonio E, et al. Improving digital system diagnostics through prognostic and health management technology [J]. *IEEE Trans. on Instrumentation and Measurement*, 2009,58(2):255 - 262.

[8] 王永翔, 王立德. 基于广义随机 Petri 网的 MVB 网络吞吐性能分析[J]. 北京交通大学学报, 2008,32(5):98 - 101.

[9] 杜宾, 邱菀华. 多级保障系统的广义随机 Petri 网建模与分析[J]. 北京理工大学学报, 2009,29(11):1030 - 1034.

[10] 马增治. 汽车制造供应链系统建模及性能分析[D]. 长春:吉林大学机械科学与工程学院,2008.

[11] 林闯. 随机 Petri 网和系统性能评价[M]. 北京:清华大学出版社, 2000.

[12] 汤道宇, 王少萍. 基于事件调度的随机 Petri 网仿真[J]. 系统仿真学报, 2004,16(3):551 - 559.

[13] 王少萍. 工程可靠性[M]. 北京:北京航空航天大学出版社, 2000.

[14] Karama K. Fault-tolerant system dependability-explicit modeling of hardware and software component-interactions[J]. *IEEE Trans. on Reliability*, 2000,49(4):363 - 376.

[15] Karama K, Marie B. Availability of CAUTRA, a subset of the French air traffic control system [J]. *IEEE Trans. on Computers*, 1999,48(5):528 - 535.