doi: 10.3969/j.issn.1006-1576.2011.08.025

一种基于故障树分析的软件设计方法

李祥明，吴学军，王玮，茅文浩
（中国卫星海上测控部技术部，江苏 江阴 214431）

摘要：为尽量减少被软件设计过程传播的故障数，提出一种基于故障树分析（fault-tree analysis，FTA）的软件设计方法。以故障树分析为基础，从软件的静态和动态特性出发，将其引入到软件设计开发过程中，并以实例进行建模、分析。结果表明：该方法计算复杂度低、精确度高，建模过程简单，适合于规模小的系统。在海上测控软件的设计开发过程中应用该方法，提高了软件的可靠性和安全性。

关键词：静态故障树分析；动态故障树分析；软件设计
中图分类号：TP311.52　　**文献标志码**：A

A Software Design Method Based on Fault-Tree Analysis

Li Xiangming, Wu Xuejun, Wang Wei, Mao Wenhao
(Technology Department, Satellite Maritime Tracking & Controlling Department of China, Jiangyin 214431, China)

Abstract: For reduces as far as possible the number of failure by the software design process dissemination, put forward a new software design method based on fault-tree analysis. On the basis of fault-tree analysis, it should its import software design development process according to these characteristic of static and dynamic of software, and taking example process modeling and analysis. The result shows that the method has low complexity, high accuracy of calculate, process simple of modeling, suitability for small-scale system. The method can advance reliability and security in the development process of maritime test and control.

Keywords: static fault-tree analysis; dynamic fault-tree analysis; software design

0　引言

目前用于软件开发过程的软件可靠性分析方法主要有 2 种：软件失效模式和影响分析（failure model and efects analysis，FMEA）以及故障树分析法（fault-tree analysis，FTA）。FMEA 主要是在软件开发阶段，通过识别软件失效模式，研究分析各种失效模式产生的原因及其造成的后果，以尽早发现潜在的问题，并采取相应的措施，从而提高软件的可靠性和安全性。FTA 则是“在系统设计过程中，通过对可能造成系统失效的各种因素（硬件、软件、环境、人为因素等）进行分析，画出逻辑图（即故障树），从而确定系统失效原因的各种可能组合方式或其发生概率，采取相应纠正措施，提高系统可靠性的一种设计分析方法”[1]。从形式上看，FMEA 是从细节出发，分析对相关模块的影响，是一种归纳的方法，而 FTA 则是从故障顶事件出发，查找发生故障的所有可能原因，是一种演绎的方法。

在软件开发过程中，由于软件缺陷有传播放大的特性，越到软件开发的后期，前期的软件设计缺陷会被成倍数放大，修改的成本会越高[2]。为尽量减少被软件设计过程传播的故障数，笔者将 FTA 引入到软件设计开发的过程，通过在软件设计各个阶段进行 FTA 分析，确定软件故障原因的各种可能组合方式及其发生概率，便于软件开发人员在设计过程中采取有效措施，防止故障被后续阶段传播并扩大，以期有效提高软件产品的可靠性和安全性。

1　故障树分析法

1.1　静态故障树分析法

传统的 FTA 方法以概率论和布尔代数为基础，将故障树中基本事件的概率表示为确定值。在进行分析时，将系统不希望发生的故障状态定义为“顶事件”，通过分析寻找出导致顶事件发生的所有可能的“中间事件”，接着分析每一个中间事件发生的可能原因，以此类推，直至追踪到最后一级的基本事件，即“底事件”，并根据所有事件间的逻辑关系以图形形式描述成故障树图。采用行列法（fusselly-vesely 算法[3]）求解出全体最小割集，这些最小割集的组合对应了故障的所有可能基本原因的组合。通过对所有最小割集进行定性或定量分析，确定各最小割集发生故障的可能性及概率，在软件设计中只要针对这些最小割集采取相应措施就可以

收稿日期：2011-04-15；修回日期：2011-05-24
作者简介：李祥明（1968—），男，浙江人，硕士，高级工程师，从事航天测控软件总体设计、GJB5000A 应用研究。

防止故障的发生。

静态故障树分析法将故障树图中的各同层事件视为相互独立，不存在时序和优先关系，其分析过程通常为：

1) 从顶事件开始，使用逻辑运算符逐层建立故障树；

2) 用行列法求解所有最小割集；

3) 定性分析故障发生原因，对最小割集制定相应措施；

4) 定量计算分析各最小割集和顶事件的故障概率，根据计算结果对发生故障概率大的事件采取预防措施。

1.2 动态故障树分析法

1.2.1 动态故障树分析法

在现实中，有很多系统具有动态的特性，各事件之间存在诸多的依赖关系，某一事件的发生往往取决于当时系统所处的运行状态。在发生故障时，各事件之间存在时序、优先等关系，这是传统的 FTA 无法描述的。动态故障树分析法(dynamic fault-tree analysis，DFTA)就是在传统 FTA 和马尔科夫(Markov)方法的基础上，根据所分析系统的动态特性，引入能够反映事件序列动态特性的逻辑运算符，建立动态故障树，采用动态分析模型对故障树进行分析的过程。

1.2.2 动态故障树的处理方法

在建立动态故障树后，对故障树中的静态部分可以采用行列法进行求解，对于简单小规模的静态故障树也可采用二元决策图(binary decision diagram，BDD)方法进行计算分析。对 DFT 部分，采用 Markov 链进行求解[4-5]。Markov 过程是一种特殊的随机过程，当前时刻的系统状态只与迁移时刻的状态有关，而与其他任何时刻的状态无关，即“无后”特性。在 Markov 链中，以一组圆圈代表系统所具有的不同状态，用一组带有权值的有向线段表示系统以权值大小的转移概率从一个状态转移到另一个状态。

由于 Markov 因其状态空间的规模随系统规模增大而呈指数增长，会导致 Markov 模型建立和求解繁琐甚至由于运算量太大而无法使用，因此要求被分析的系统规模不能太大。通常情况下，整个 DFT 模型中动态的部分并不太多。因此，在用 DFT 分析方法对整个 DFT 模型进行处理时，必须在整个 DFT 模型中识别出独立子树，将动态故障子树和静态故障子树区分开来，然后决定采用一般 FTA 的处理方法，还是用 Markov 模型对独立子树而不是整个 DFT 进行处理，也就是用几个 Markov 模型和一般 FTA 模型将整个 DFT 进行分解，使每个独立处理模型规模变小，以简化分析过程，并避免 Markov 分析方法不适应大规模系统的缺点，对静态故障子树也就可以采用简单 BDD 方法进行分析计算了。

2 故障树软件设计方法

2.1 故障模型的建立

在一般的软件系统中，软件事件的静态关系可以用与、或来表示，因此用与逻辑门和或逻辑门就可以构建出软件系统的故障树，对于具有动态特性的事件则必须建立新的逻辑运算符，以反映软件系统的动态特性。假设在某一软件系统中，软件模块之间的动态关系包括时序关系、比例关系、触发关系、备份选择关系，可以表示为如下逻辑运算符：

1) 时间顺序门

时间顺序门(time sequence gate，TSG)，当多个底事件以固定的时间顺序依次发生时，顶事件才会发生，如图 1。

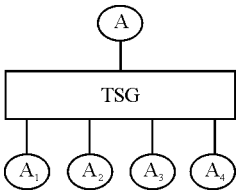

图 1 时间顺序门

2) 比例门

比例门(percent gate，PG)，当全部底事件同时工作，当一定数量(百分比)的底事件发生时，引起顶事件发生，如图 2。

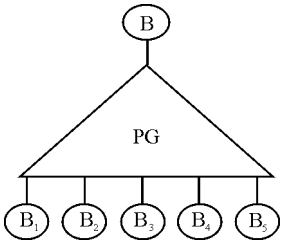

图 2 比例门

3) 触发门

触发门(functional dependency gate，FDG)，当触发事件发生时，就会引起顶事件的发生；或者当基本事件发生时也可引起顶事件的发生，如图 3。

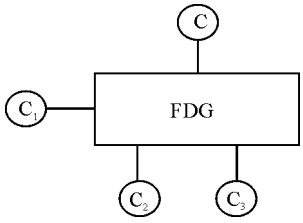

图 3　触发门

4) 备份选择门

备份选择门(spare choice gate，SCG)，所有的备份选择项都处于不确定的状态，一些处于工作状态，另一些处于非工作状态，选择的顺序按照失效率的高低进行排列，失效率低的备份项作为优先的预选项，失效率高的备份项则排列在后面。当主选项有效时，不考虑备份项的状态，当主选项失效时，备份项才发挥作用。当主选项和所有备份项均失效时，顶事件才会发生，如图 4。

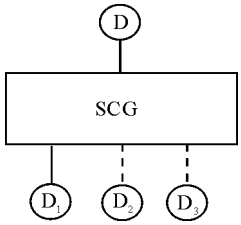

图 4　备份选择门

利用传统 FTA 的逻辑运算符和动态逻辑运算符，可以达到对具有静态和动态特性的软件系统进行故障分析的目的，建立起不同级别的软件故障树，接下来就可以分解故障树，根据静态和动态特性的不同，对每个独立的子树分别采用 BDD 方法和 Markov 方法进行分析计算，如图 5。

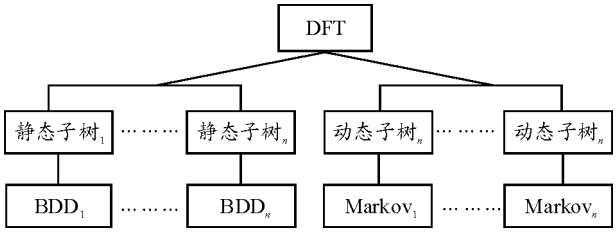

图 5　故障树分解

2.2　实例分析

2.2.1　实例建模

现假设一个软件功能部件 X 有 4 个功能模块 A、B、C、D 组成，当模块 A 或模块 B 发生失效或模块 C 和模块 D 同时失效时，功能部件 X 失效，其中：

1) 模块 A 由 2 个功能子模块 A_1、A_2 组成，2 个功能子模块同时工作，当功能子模块 A_1 发生故障后，功能子模块 A_2 也发生故障时，则模块 A 故障；

2) 模块 B 由 5 个功能子模块 B_1、B_2、B_3、B_4、B_5 组成，当其中的任何 3 个发生故障时，模块 B 发生故障；

3) 模块 C 由 C_1、C_2、C_3 组成，当 C_1 故障时，引起 C 故障；或者 C_2、C_3 都故障时，C 也故障；

4) 模块 D 由 D_1、D_2、D_3 组成，发生故障的概率从低到高分别为 D_1、D_2、D_3，D_1 为主件，D_2、D_3 为备件，当 D_1 故障时，D_2 开始工作；同样，D_2 故障时，D_3 才开始工作，当 D_1、D_2、D_3 均为故障时，D 故障。

通过分析可知：

1) 模块 A、B 与 CD 为或门关系，C、D 为与门关系，当模块 C、D 同时发生故障或 A、B 之一发生故障时，功能部件 X 故障；

2) 模块 A 为时间顺序门，当功能子模块 A_1、A_2 依次发生故障时，模块 A 故障；

3) 模块 B 为比例门，当 B_1、B_2、B_3、B_4、B_5 中任意 3 个发生故障时，模块 B 故障；

4) 模块 C 为触发门，当 C_1 故障时，C 故障；或者 C_2、C_3 同时故障时，C 也故障；

5) 模块 D 为备份选择门，当 D_1、D_2、D_3 均为故障时，D 故障。

基于以上分析，可以得到功能部件 X 的 DFT 模型，如图 6。

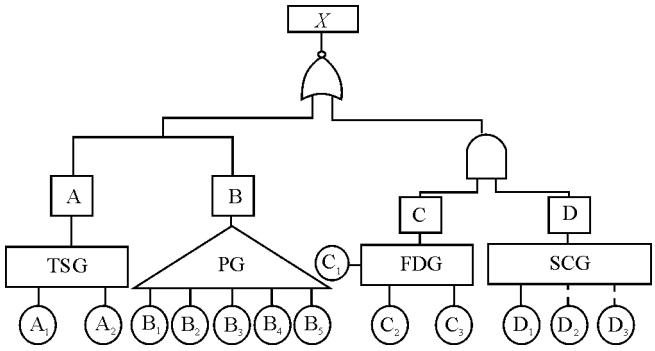

图 6　部件 X 的故障树

2.2.2　静态分析

根据对图 6 的分析，可以得到，由模块 A、B、C、D 组成的关系为静态逻辑关系，可以划分为静态 FTA 部分如图 7，可以用 BDD 方法进行计算。根据 A、B、C、D 的逻辑关系，可以得到静态 BDD 图如图 8，其中 0 代表未故障状态，1 代表故障状态。

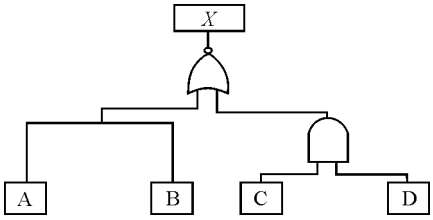

图 7 静态故障子树

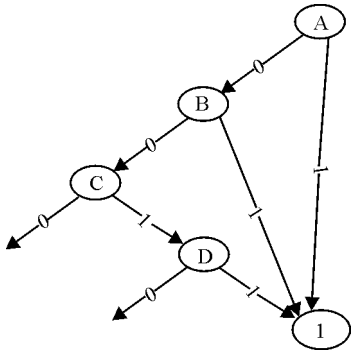

图 8 静态 BDD 图

根据静态 BDD 图，功能部件 X 的结构函数可以表示为：

$$\Phi(X)=\mathrm{A}\cup\overline{\mathrm{A}}\,\mathrm{B}\cup\overline{\mathrm{A}}\,\overline{\mathrm{B}}\,\mathrm{CD}=\mathrm{A}\cup\mathrm{B}\cup\mathrm{CD} \quad (1)$$

相应地，功能部件 X 发生故障的概率为：

$$P(X)=P(\Phi(X)=1)=P(\mathrm{A})+P(\overline{\mathrm{A}}\,\mathrm{B})+P(\overline{\mathrm{A}}\,\overline{\mathrm{B}}\,\mathrm{CD})=$$
$$P(\mathrm{A})+(1-P(\mathrm{A}))P(\mathrm{B})+(1-P(\mathrm{A}))(1-P(\mathrm{B})P(\mathrm{C})P(\mathrm{D})=$$
$$P(\mathrm{A})+P(\mathrm{B})+P(\mathrm{C})P(\mathrm{D}) \quad (2)$$

由以上分析可知，在不考虑模块 A、B、C、D 之间故障发生概率差异因素和模块功能重要性不同因素时，模块 A 和 B 都可直接引起功能部件 X 发生故障，而且功能部件 X 发生故障概率的主要成分也是由 A、B 组成。因此，在软件设计时，要把 A、B 作为功能部件 X 的安全关键模块，在安全性、可靠性设计等方面给予更多的关注，以提高功能部件 X 的可靠性。

2.2.3 DFT 分析

通过对图 6 的分析可以得知，模块 A、B、C、D 均为动态逻辑关系，可以划分为 DFT 部分。其中，模块 A 为时间顺序门，模块 B 为比例门，模块 C 为触发门，模块 D 为备份选择门，对于这些模块可以用 Markov 链进行求解。在用 Markov 链进行求解时，用带标识的圆圈表示系统所示的状态，通常用 N_i 表示中间状态，用 Fa 表示故障状态，用带有权值(故障概率)的有向线表示以权值大小的概率从一个状态转移到另一个状态。

1) 模块 A-时间顺序门

在某时刻 t，模块 A 的状态可能为如下之一：

$$\mathrm{A}(t)=\begin{cases}N_0,\ \mathrm{A}_1、\mathrm{A}_2\ 2\ \text{个子模块正常，模块A正常}\\ \mathrm{N}_1,\ \mathrm{A}_1\ \text{故障，}\mathrm{A}_2\ \text{正常，模块A正常}\\ \mathrm{N}_2,\ \mathrm{A}_1\ \text{正常，}\mathrm{A}_2\ \text{故障，模块A正常}\\ \mathrm{N}_3,\ \mathrm{A}_2\ \text{故障后}\mathrm{A}_1\ \text{发生故障，模块A正常}\\ \mathrm{Fa},\ \mathrm{A}_1\ \text{故障后}\mathrm{A}_2\ \text{发生故障，模块A故障}\end{cases}$$

假设 A_1 发生故障的概率为 λ_1，A_2 发生故障的概率为 λ_2，则模块 A 的状态转移概率为：

$$P(\mathrm{N}_0-\mathrm{N}_0)(\Delta t)=1-(\lambda_1+\lambda_2)\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N}_0-\mathrm{N}_1)(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N}_0-\mathrm{N}_2)(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N}_1-\mathrm{N}_1)(\Delta t)=1-\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N}_2-\mathrm{N}_2)(\Delta t)=1-\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N}_1-\mathrm{Fa})(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N}_2-\mathrm{N}_3)(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$

其中，$\sigma(\Delta t)$为高阶无穷小量，可以忽略。状态自我转移可以认为是一种状态保持，对系统没有影响，也可以不作分析。由以上状态转移过程及转移概率得到相应的 Markov 状态转移链，如图 9。

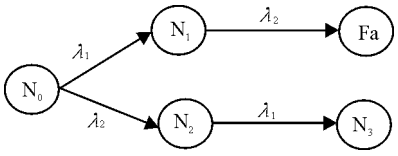

图 9 模块 A 的 Markov 状态转移链

由于功能子模块 A_1、A_2 为 2 个相互独立的模块，没有逻辑关系，因此可认为这 2 个功能子模块发生故障的事件是相互独立的。根据 Markov 状态转移链，模块 A 的故障概率为：

$$P(\mathrm{A})=P(\mathrm{N}_0-\mathrm{N}_1)(\Delta t)P(\mathrm{N}_1-\mathrm{Fa})(\Delta t)=$$
$$(\lambda_1\Delta t+\sigma(\Delta t))(\lambda_2\Delta t+\sigma(\Delta t))=$$
$$\lambda_1\lambda_2=$$
$$P(\mathrm{A}_1)P(\mathrm{A}_2) \quad (3)$$

2) 模块 B-比例门

在某时刻 t，模块 B 的状态可能为如下之一：

$$B(t)=\begin{cases}N_0,\ B_1、B_2、B_3、B_4、B_5正常，B正常\\ N_1,\ B_1故障，B_2正常，B_3正常，B_4正常，B_5正常，B正常\\ N_2,\ B_1故障，B_2故障，B_3正常，B_4正常，B_5正常，B正常\\ Fa_1,\ B_1故障，B_2故障，B_3故障，B_4正常，B_5正常，B故障\\ Fa_2,\ B_1故障，B_2故障，B_3正常，B_4故障，B_5正常，B故障\\ Fa_3,\ B_1故障，B_2故障，B_3正常，B_4正常，B_5故障，B故障\\ N_3,\ B_1正常，B_2故障，B_3正常，B_4正常，B_5正常，B正常\\ N_4,\ B_1正常，B_2故障，B_3故障，B_4正常，B_5正常，B正常\\ Fa_4,\ B_1正常，B_2故障，B_3故障，B_4故障，B_5正常，B故障\\ Fa_5,\ B_1正常，B_2故障，B_3故障，B_4正常，B_5故障，B故障\\ N_5,\ B_1正常，B_2正常，B_3故障，B_4正常，B_5正常，B正常\\ N_6,\ B_1正常，B_2正常，B_3故障，B_4故障，B_5正常，B正常\\ Fa_6,\ B_1正常，B_2正常，B_3故障，B_4故障，B_5故障，B故障\\ N_7,\ B_1正常，B_2正常，B_3正常，B_4故障，B_5正常，B正常\\ N_8,\ B_1正常，B_2正常，B_3正常，B_4故障，B_5故障，B正常\\ N_9,\ B_1正常，B_2正常，B_3正常，B_4正常，B_5故障，B正常\end{cases}$$

假设 B_1 发生故障的概率为 λ_1，B_2 发生故障的概率为 λ_2，B_3 发生故障的概率为 λ_3，B_4 发生故障的概率为 λ_4，B_5 发生故障的概率为 λ_5，则模块 B 的状态转移概率为：

$$P(N_0-N_1)(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$
$$P(N_1-N_2)(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(N_2-Fa_1)(\Delta t)=\lambda_3\Delta t+\sigma(\Delta t)$$
$$P(N_2-Fa_2)(\Delta t)=\lambda_4\Delta t+\sigma(\Delta t)$$
$$P(N_2-Fa_3)(\Delta t)=\lambda_5\Delta t+\sigma(\Delta t)$$
$$P(N_0-N_3)(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(N_3-N_4)(\Delta t)=\lambda_3\Delta t+\sigma(\Delta t)$$
$$P(N_4-Fa_4)(\Delta t)=\lambda_4\Delta t+\sigma(\Delta t)$$
$$P(N_4-Fa_5)(\Delta t)=\lambda_5\Delta t+\sigma(\Delta t)$$
$$P(N_0-N_5)(\Delta t)=\lambda_3\Delta t+\sigma(\Delta t)$$
$$P(N_5-N_6)(\Delta t)=\lambda_4\Delta t+\sigma(\Delta t)$$
$$P(N_6-Fa_6)(\Delta t)=\lambda_5\Delta t+\sigma(\Delta t)$$
$$P(N_0-N_7)(\Delta t)=\lambda_4\Delta t+\sigma(\Delta t)$$
$$P(N_7-N_8)(\Delta t)=\lambda_5\Delta t+\sigma(\Delta t)$$
$$P(N_0-N_9)(\Delta t)=\lambda_5\Delta t+\sigma(\Delta t)$$

由于所有的故障状态都导致模块 B 故障，因此，可以将 Fa_1、Fa_2、Fa_3、Fa_4、Fa_5、Fa_6 统一表示为 Fa。由以上状态转移过程及转移概率得到相应的 Markov 状态转移链，如图 10。

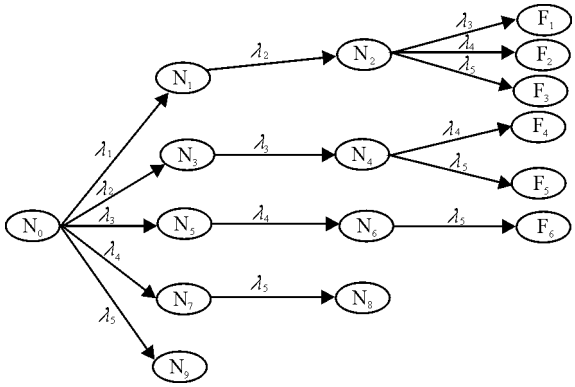

图 10　模块 B 的 Markov 状态转移链

可以得到模块 B 的故障概率为：

$$P(B)=\lambda_1\lambda_2(\lambda_3+\lambda_4+\lambda_5)+\lambda_2\lambda_3(\lambda_4+\lambda_5)+\lambda_3\lambda_4\lambda_5=$$
$$P(B_1)P(B_2)(P(B_3)+P(B_4)+P(B_5))+$$
$$P(B_2)P(B_3)(P(B_4)+P(B_5))+P(B_3)P(B_4)P(B_5)\quad(4)$$

3) 模块 C–触发门

在某时刻 t，模块 C 的状态可能为如下之一：

$$C(t)=\begin{cases}N_0,\ C_1、C_2、C_3正常，C正常\\ Fa_1,\ C_1故障，C_2正常，C_3正常，C故障\\ N_1,\ C_1正常，C_2故障，C_3正常，C正常\\ N_2,\ C_1正常，C_2正常，C_3故障，C正常\\ Fa_2,\ C_1正常，C_2故障，C_3故障，C故障\\ Fa_3,\ C_1故障，C_2故障，C_3正常，C故障\\ Fa_4,\ C_1故障，C_2正常，C_3故障，C故障\end{cases}$$

假设 C_1 发生故障的概率为 λ_1，C_2 发生故障的概率为 λ_2，C_3 发生故障的概率为 λ_3，则模块 C 的状态转移概率为：

$$P(N_0-N_0)(\Delta t)=1-(\lambda_1+\lambda_2+\lambda_3)\Delta t+\sigma(\Delta t)$$
$$P(N_0-N_1)(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(N_0-N_2)(\Delta t)=\lambda_3\Delta t+\sigma(\Delta t)$$
$$P(N_0-Fa_1)(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$
$$P(N_1-N_1)(\Delta t)=1-(\lambda_1+\lambda_3)\Delta t+\sigma(\Delta t)$$
$$P(N_1-Fa_2)(\Delta t)=\lambda_3\Delta t+\sigma(\Delta t)$$
$$P(N_1-Fa_3)(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$
$$P(N_2-N_2)(\Delta t)=1-(\lambda_1+\lambda_2)\Delta t+\sigma(\Delta t)$$
$$P(N_2-Fa_2)(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(N_2-Fa_4)(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$

由于所有的故障状态都导致模块 C 故障，因此，可以将 Fa_1、Fa_2、Fa_3、Fa_4 统一表示为 Fa。由以上状态转移过程及转移概率得到相应的 Markov 状态转移链，如图 11。

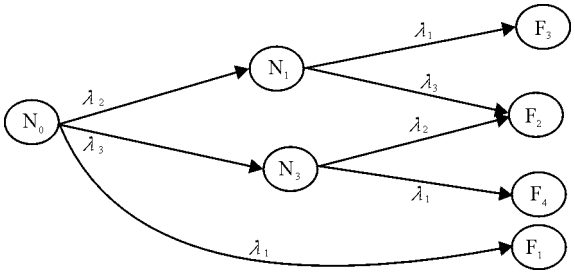

图 11 模块 C 的 Markov 状态转移链

可以得到模块 C 的故障概率为：

$$P(\mathrm{C})=\lambda_1+2\lambda_2\lambda_3+\lambda_1\lambda_2+\lambda_1\lambda_3=$$
$$P(\mathrm{C_1})+2P(\mathrm{C_2})P(\mathrm{C_3})+P(\mathrm{C_1})P(\mathrm{C_2})+P(\mathrm{C_1})P(\mathrm{C_3}) \tag{5}$$

4) 模块 D−备份选择门

在某时刻 t，模块 D 的状态可能为如下之一：

$$D(t)=\begin{cases}\mathrm{N_0}，\mathrm{D_1}、\mathrm{D_2}、\mathrm{D_3}正常，\mathrm{D}正常\\ \mathrm{N_1}，\mathrm{D_1}故障，\mathrm{D_2}正常，\mathrm{D_3}正常，\mathrm{D}正常\\ \mathrm{N_2}，\mathrm{D_1}故障，\mathrm{D_2}故障，\mathrm{D_3}正常，\mathrm{D}正常\\ \mathrm{Fa}，\mathrm{D_1}故障，\mathrm{D_2}故障，\mathrm{D_3}故障，\mathrm{D}故障\end{cases}$$

假设 $\mathrm{D_1}$ 发生故障的概率为 λ_1，$\mathrm{D_2}$ 发生故障的概率为 λ_2，$\mathrm{D_3}$ 发生故障的概率为 λ_3 则模块 D 的状态转移概率为：

$$P(\mathrm{N_0}-\mathrm{N_0})(\Delta t)=1-\lambda_1\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N_0}-\mathrm{N_1})(\Delta t)=\lambda_1\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N_1}-\mathrm{N_1})(\Delta t)=1-\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N_1}-\mathrm{N_2})(\Delta t)=\lambda_2\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N_2}-\mathrm{N_2})(\Delta t)=1-\lambda_3\Delta t+\sigma(\Delta t)$$
$$P(\mathrm{N_2}-\mathrm{Fa})(\Delta t)=\lambda_3\Delta t+\sigma(\Delta t)$$

由以上状态转移过程及转移概率得到相应的 Markov 状态转移链，如图 12。

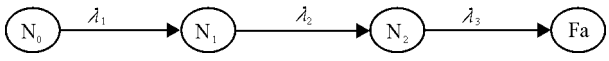

图 12 模块 D 的 Markov 状态转移链

可以得到模块 D 的故障概率为：

$$P(\mathrm{D})=\lambda_1\lambda_2\lambda_3=P(\mathrm{D_1})P(\mathrm{D_2})P(\mathrm{D_3}) \tag{6}$$

在已知各个子模块的故障概率的情况下，就可以求出各个模块的故障概率，计算出功能部件 X 的故障概率，从而得出各模块以及功能部件 X 的可靠性结论。

通过以上分析可知，各种 DFT 模块的故障概率大小以及故障构成因素，在不考虑各子模块概率差异因素时，由于触发门可以通过单个事件造成模块故障，因此可靠性最差，其次为比例门，时间顺序门和备份选择门发生条件最为苛刻，其可靠性也就相对较高。因此，在软件设计时，要慎重考虑触发门的使用，并对触发事件进行严格控制，通过多重条件判断形成结果，提高触发事件的可信度；对于比例门，由于是通过随机组合达到比例条件而触发故障的，因此在比例条件不高、子模块较多的情况下，发生故障的概率较大，需要提高每个子模块的可靠性，以降低整个模块的故障概率；同比例门一样，时间顺序门和备份选择门也需要对每个子模块进行控制，降低故障率，提高模块可靠性。

2.3 软件设计

由于造成软件故障的原因有很多种，产生的危害程度也不一样[6]，将 FTA 引入到软件设计时，需要在软件设计的每一个阶段进行 FTA。由于在软件开发的不同阶段(系统设计、需求分析、概要设计、详细设计)，其设计的对象是不同的，应根据不同的阶段和不同的对象分析，采用不同的分析方法。软件系统级 FTA 在软件设计阶段的早期进行，分析对象是设计阶段早期的高层次的子系统、模块及软硬件接口，目的是对软件体系结构进行故障分析；在需求分析级 FTA 的分析对象是软件的功能部件以及与外部接口，目的是分析各功能部件之间以及软件与外部之间的故障关系；软件概要设计级 FTA 的分析对象是已经经过设计的软件功能模块或部件以及相互接口，目的是找出各种故障产生的原因及其发生概率，并提出相应的预防和改进措施；软件详细级 FTA 分析的任务是深入软件模块内部，通过分析每个软件单元的变量和算法的故障模式，确定产生故障的原因，保证软件单元的可靠性。

在设计过程中，首先要理清各模块之间的逻辑关系，勾画出模块之间的故障关系，形成 FTA 模型；其次对 FTA 模型进行分解，划分成规模较小的故障子树，根据每个子树的静态、动态特性分别采用 BDD 和 DFT 进行分析，分析每个模块及整个系统的故障原因，计算相应的故障概率；最后根据各模块的功能关键等级和故障重要度找出整个系统的重点对象。

对发生故障率高、对系统故障重要度大的软件模块以及软件功能关键部件要采取安全措施，进行安全性设计，加强验证，并在后续设计中进一步细化安全措施，降低故障发生概率，这样就能成倍地提高系统的可靠性和安全性，降低使用风险。对所有其他的中间事件、底事件也都必须认真分析，认

真设计，因为在一个软件系统中，任何存在的缺陷，只要发生故障都会对系统的运行发生影响。

在建立故障树模型时，由于分析计算的复杂度随故障树的规模迅速增加，因此故障树的规模不宜太大，一般不超过 3 阶，对于规模过大的故障树，可以分解为多个阶数小的子树进行分析，以简化分析计算过程。

3　结论

综上所述，在软件设计过程中，采用 FTA 和 DFT 分析方法，可以直观地构造出软件系统的故障模型，能定性地分析出各种软件模块的故障重要度，定量地计算出软件发生故障的概率，为软件可靠性设计提供重要参考；由于 Markov 方法能够表征系统的动态特性，而且建模过程相对简单，计算复杂度低、精度高，是一种比较理想的动态分析方法，但只适合于规模小的系统。

该方法已在神舟七号飞船和嫦娥二号发射任务海上测控软件的设计开发过程中得到了应用，在不同阶段分别找出了多个故障，并提出了相应的安全措施，提高了海上测控软件的可靠性和安全性。结果表明，采用基于故障树分析的软件设计方法在提高软件可靠性上有较高的工程价值。

参考文献：

[1] 国家标准 GB3187-82 可靠性基本名词术语及定义[S].

[2] 石柱. 军用软件能力成熟度模型可重复级实施指南[Z]. 北京: 中国标准出版社, 2006.

[3] 朱云鹏. 基于故障树分析法的软件测试技术研究[J]. 计算机工程与设计, 2008, 29(13): 3388.

[4] 常柏林, 孙中泉, 刘阳. GJB5000A2 级军用软件过程管理[J]. 四川兵工学报, 2010, 31(10): 141.

[5] Dugan J B, Coppit D. Developing aloe-cost high-quality software tool for dynamic fault-tree analysis[J]. IEEE Trans.on Reliability, 2000, 49(1): 49-59.

[6] 张仝伟, 石柱. SFMEA 方法在飞行控制软件中的应用[J]. 航天控制, 2007, 25(2): 58-62.

****

（上接第 77 页）

对比两阵型的方位角误差公式 (1) 、公式 (3) ，在各项参数数值相同的情况下，由公式计算得到的两阵型方位角误差相同。由于式 (2) 较复杂，不便与式 (4) 直接对比，为直观、量化立体五元十字阵与平面五元十字阵俯仰角定向性能的差异，下面通过计算机进行仿真。

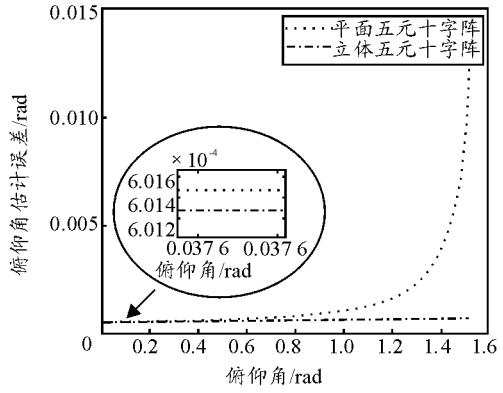

图 2　俯仰角估计均方误差比较图

根据两阵型俯仰角误差公式，设声速 c=340 m/s、时延标准偏差为 δ_τ=5 μs，令结结构尺寸参数 h=2 m，a=2.5 m，俯仰角 θ 的变化范围为 $\left(0,\frac{\pi}{2}\right)$，对平面五元十字阵、立体五元十字阵的俯仰角误差进行仿真，可得到如图 2 曲线。从图 2 中可以看出，在整个区间内立体五元十字阵的方位角误差精度均高于平面五元十字阵，并且立体五元十字阵变化趋势平缓，测区间内任意角度均能保证较高的精度。

5　结论

仿真结果证明：立体五元十字阵的俯仰角精度较高，俯仰角变化时其误差波动较小，能够在所给区间内保持一个相对稳定的精度；同时，由于俯仰角的精度对方位角的测量也有一定影响，所以立体五元十字阵的整体定向性能较好，具有很高的工程应用价值。

参考文献：

[1] Ferguson B G, Lo K W. Passive ranging errors due to multipath distortion of deterministic transient signals with application to localization of small arms fire[J]. Acoust. Sac. Am., 2002, 111(1): 117-128.

[2] Lo K W, Ferguson B G, et al. Aircraft flight parameter estimation using acoustic multipath delays[J]. IEEE Trans. on Aerosp. Electron. Syst., 2003, 39(1): 259-267.

[3] 马社祥, 赵朝霞. 空间交叉阵被动声定位算法及其性能分析[J]. 系统工程与电子技术, 2009(3): 553-556.

[4] 程翔, 张河. 高低四元阵定位算法及其精度分析[J]. 探测与控制学报, 2006(4): 12-14.

[5] 陈华伟, 赵俊渭, 郭业才. 五元十字阵被动声定位算法及其性能研究[J]. 探测与控制学报, 2003(4): 11-16.

[6] 贾云得, 冷树林, 刘万春, 等. 四元被动声敏感阵列定位模型分析和仿真[J]. 兵工学报, 2001(2): 206-209.

[7] 庞学这, 张效民, 杨向锋. 利用圆锥阵的低空声目标定向分析[J]. 兵工自动化, 2005(1): 1-2.