

# 硕士学位论文

## 系统级层次化测试性建模与分析

### TESTABILITY MODELING AND ANALYSIS OF HIERARCHICAL SYSTEM

郑 应 荣

哈尔滨工业大学

2014 年 6 月

国内图书分类号：TP206<sup>+</sup>.1  
国际图书分类号：621.3

学校代码：10213  
密级：公开

## 工程硕士学位论文

# 系统级层次化测试性建模与分析

硕 士 研 究 生：郑应荣

导 师：杨春玲 教授

申 请 学 位：工程硕士

学 科：电气工程

所 在 单 位：电气工程及自动化学院

答 辩 日 期：2014 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TP206<sup>+</sup>.1

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

# **TESTABILITY MODELING AND ANALYSIS OF HIERARCHICAL SYSTEM**

<b>Candidate:</b>	Zheng Yingrong
<b>Supervisor:</b>	Prof. Yang Chunling
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Speciality:</b>	Electrical Engineering
<b>Affiliation:</b>	School of Electrical Engineering and Automation
<b>Date of Defence:</b>	June, 2014
<b>Degree-Confering-Institution:</b>	Harbin Institute of Technology

## 摘 要

测试性是反映系统内部故障能否被准确及时地检测与隔离的一种设计特性，其水平高低关乎系统能否持久稳定地发挥其使用效能。高度集成化和微型化使得系统测试难度与日俱增，而航天军事领域对于复杂系统的测试性要求越来越高，因此测试性设计越来越受到人们的重视。

作为测试性设计的核心，测试性建模与分析能够客观评估系统测试性设计水平，对于优化测试性设计方案、减少测试时间和费用、提高测试效率意义重大。实际测试维修工程一般都是针对系统级开展的，然而目前对系统级的测试性建模与分析方法研究还不够深入。本文将系统层次划分、测试性指标分配、测试性建模及其分析方法结合起来构成一个完整的系统级测试性模型分析与评价体系，并设计相应的测试性建模与分析软件。主要工作包括以下几个部分：

首先，分析实际的测试、诊断与维修活动对于系统层次划分的需求，提出了系统层次划分的原则和方法；明确系统测试性指标分配的一般流程及其数学模型，研究三种经典的测试性分配算法，从故障率、故障严酷度、故障检测与隔离难度、平均修复时间和测试维修费用等角度确定各单元测试性指标分配额。

其次，研究故障-信号-测试(Fault-Signal-Test, FST)模型的基本思想、结构描述及其测试性分析方法；针对 FST 模型不能处理故障与测试之间不确定性关系的缺陷，提出融合概率信息的 FST 模型，并给出了该模型的测试性分析方法。

然后，完成系统级层次化测试性建模与分析软件的开发，包括软件总体架构的设计，层次化测试性建模平台的构建，以及测试性分析平台中模型信息提取模块、测试性指标分配模块和测试性分析模块的设计，实现对系统级层次化测试性模型的综合评价。

最后，利用设计的软件和商业软件 TEAMS 分别建立某电子侦察系统的层次化测试性模型，并进行测试性分配和测试性分析，并与实际测试工程统计数据进行比较，验证本文提出的系统级测试性建模与分析方法的正确性与可靠性。

**关键词：**测试性建模与分析；FST 模型；系统层次划分；测试性指标分配

## Abstract

Testability is a design characteristic which reflects whether the system internal fault could be detected and isolated accurately and timely. A high level of testability guarantees system a durable and steady performance. With the integration and miniaturization of the complex and sophisticated electronic systems, a significant increase in the difficulty of testability analysis and fault diagnosis happens. However, the demand for system testing of many critical area such as aerospace and military is higher and higher. It is indispensable of design for testability to improve the level of system testability.

As the key of design for testability based on the model, testability modeling and analysis can evaluate testability level of system both in qualitatively and quantitatively perspective, guide designers to make perfect testing schedule and diagnosis strategy, thus save test time, reduce test costs, and improve test efficiency. Up to now, there is not a general method suitable for the system level, while the actual test and maintenance works are mostly aimed at the system level. Therefore, studying the method of testability modeling and analysis for system is of great significance. This paper puts system hierarchical division, testability index allocation, testability modeling and analysis together to build a testability evaluation system. Main work of the paper includes the following sections.

At first, research in the demand for system hierarchy division of actual test, diagnosis and maintenance are done, and the principles and methods of system hierarchy division are proposed. The general procedures and mathematical of testability index allocation are specified. Three testability index allocation methods are studied. And their similarities and differences are compared in terms of factors concerned, allocation difficulty and result.

Besides, the fundamental, structure description, testability analysis methods and procedures of FST(Fault-Signal-Test) model are studied. The shortcoming of FST model is pointed out: it ignores probabilistic information of test and is unable to deal with the uncertain relationship between tests and faults. For this reason, probability integration FST model is proposed and corresponding testability analysis methods are provided.

And then, the development of the hierarchical system testability modeling and analysis software is completed. The each function module in the overall architecture of this software, including the hierarchical modeling platform, model information extraction module, testability index allocation module and testability analysis module, are designed. The comprehensive evaluation of system testability model is realized.

Finally, the hierarchical testability model of a certain electronic reconnaissance system is modeling respectively by the software designed in this paper and the authoritative software TEAMS. Testability index allocation and Testability analysis of its testability model are accomplished; the results of both pieces of software are compared to actual test engineering statistics to verify the validity and reliability of system testability modeling and analysis methods proposed in this paper.

**Keywords:** testability modeling and analysis, FST model, system hierarchy division, testability index allocation

# 目 录

摘 要 .....	I
Abstract .....	II
第 1 章 绪 论 .....	1
1.1 课题背景及研究的目的和意义 .....	1
1.2 国内外研究现状 .....	2
1.2.1 系统层次化建模与测试性指标分配技术 .....	2
1.2.2 测试性建模与分析技术 .....	4
1.2.3 测试性辅助设计工具 .....	5
1.3 本文的主要研究内容 .....	6
第 2 章 系统层次划分与测试性指标分配方法研究 .....	8
2.1 系统层次划分方法 .....	8
2.1.1 系统层次划分需求分析 .....	8
2.1.2 系统层次划分原则 .....	10
2.2 测试性指标分配方法 .....	11
2.2.1 测试性分配的流程及其数学模型 .....	12
2.2.2 测试性指标分配算法 .....	15
2.2.3 分配算例 .....	18
2.3 本章小结 .....	20
第 3 章 测试性建模与分析方法研究 .....	21
3.1 FST 模型 .....	21
3.1.1 FST 模型的基本思想 .....	21
3.1.2 FST 模型的描述方法 .....	22
3.1.3 FST 模型的建立过程 .....	24
3.2 FST 模型的测试性分析 .....	25
3.2.1 相关矩阵生成 .....	25
3.2.2 静态特性分析 .....	27
3.2.3 测试性指标计算 .....	28
3.3 融合概率信息的 FST 模型 .....	30
3.3.1 融合概率信息的 FST 模型的描述方法 .....	30
3.3.2 融合概率信息的 FST 模型的测试性分析 .....	32
3.4 模型实例分析 .....	33

3.5 本章小结.....	37
第 4 章 系统级层次化测试性建模与分析软件设计.....	38
4.1 软件总体架构 .....	38
4.2 测试性建模平台的构建.....	39
4.3 测试性分析平台程序设计 .....	40
4.3.1 模型信息提取模块设计 .....	40
4.3.2 测试性指标分配模块设计 .....	44
4.3.3 测试性分析模块设计 .....	45
4.4 本章小结.....	49
第 5 章 实验验证 .....	50
5.1 某野外电子侦察系统功能原理及其层次划分 .....	50
5.2 系统级层次化测试性建模与分析方法验证.....	52
5.2.1 测试性指标分配方法验证 .....	53
5.2.2 测试性分析方法验证 .....	54
5.3 本章小结.....	59
结 论 .....	60
参考文献 .....	61
攻读硕士学位期间发表的论文及其它成果 .....	65
哈尔滨工业大学学位论文原创性声明和使用权限.....	66
致 谢 .....	67



# 第 1 章 绪 论

## 1.1 课题背景及研究的目的和意义

随着科技的不断进步,系统在结构和功能方面的复杂性与先进性显著提升,正朝着高度集成化和微型化的趋势快速发展,随之也出现了许多问题,系统结构和功能愈加复杂而且尺寸不断缩小,使得系统的测试分析和故障诊断的难度与日俱增,然而航天军事等重要领域对于系统的测试时间、测试费用和测试效率等方面的要求却越来越苛刻。传统的测试方法依赖于外部辅助检测工具或者测试设备,依靠工程师根据经验开展测试工作以完成对系统的故障诊断和维修维护,这样不仅需要消耗大量的时间和人力、物力、财力等资源,而且效率较低,一般都达不到预期效果,遇到较为复杂的系统往往事倍功半,无法满足相关领域的维护要求。经过大量的实践总结,人们逐渐认识到,盲目地追求系统在性能、尺寸和成本方的设计最优化只能满足生产制造的需求,并不能从根本上为系统的最终用户在测试维修方面带来便利,要解决复杂系统的测试难题,实现对系统工作状态的实时监测和故障的准确定位,就必须在系统所有设计阶段都将系统的测试性作为设计目标之一,从系统概念形成到 subsystem 设计再到系统集成的过程中,将系统测试性与系统的功能、性能和使用寿命周期总成本统一考虑,保证系统兼具优越的使用性和良好的测试性<sup>[1-4]</sup>。目前,复杂系统主要存在以下几个方面的测试性难题:系统高度集成化和微型化大幅度低限制了有效测试节点的数目,一些外部测试设备不能在系统中添加足够的有效测试节点,无法获取故障诊断所需的所有必要测试信息;系统的结构和功能的复杂度较高,完成测试所需的时间变长、费用增高,测试效率降低,而且测试存在不确定因素,故障诊断结果并非完全可靠。为解决上述问题,测试界的许多学者及研究机构进行了广泛深入的研究探索,开创了众多经典的测试性理论、方法和技术,建立了一套相对完整的基于模型的测试性设计体系。

测试性作为系统本身的一种固有属性,反映了系统工作状态和故障信息是否能够被外部测试设备或者内部测试电路准确、及时地获取的能力<sup>[5]</sup>,测试性水平的高低直接决定了系统维护的难易程度。基于模型的测试性设计体系主要包括测试性方案设计、建模与分析、验证与评估三个部分,设计人员在分析系统的结构和原理的基础上,合理设计测试节点和测试,制定初步的测试性设计方案,但是由于设计人员的设计水平参差不齐、考虑问题的角度和全面性也不尽相同,测试性设计方案千差万别,因此需要通过统一的标准、利用一定的方法对测试性设计的优劣进行客观的评价,这正是测试性建模与分析的意义所在

[6-8]。测试性建模与分析能够对系统的测试性设计方案进行定性分析和定量评估,求解系统的静态特性和测试性指标,指出测试性方案设计中存在的不足,辅助设计人员制定一个测试时间短、费用低、效率高的完善的故障诊断策略。

经过深入细致的研究,测试界已经提出了许多实用的测试性建模与分析方法和技术,然而这些研究的对象都仅限于板级电路或者单层系统<sup>[9]</sup>,对于多层次复杂系统的研究还不够深入,也没形成一种通用有效的方法或手段,目前成型的测试性建模与分析方法大都忽略了测试存在的不确定性,将测试和故障之间的关系视为确定的。然而实际工程中,由于系统的结构通常十分复杂,为了减少测试时间、提高维修效率以保障系统安全性,测试诊断维修时并不关注具体的板级电路,而是根据现场具体情况,对系统级可更换单元进行分析,将故障定位到子系统或者子模块,维修时利用备份的子系统或者子模块直接替换可能发生故障的单元,这就要求在测试性设计阶段必须对系统进行层次划分<sup>[10]</sup>。另外受到系统自身及其外界环境中相关因素的干扰,测试结果往往并非一定可靠<sup>[11]</sup>,系统已经发生的故障未必能够被相应的测试检测出来,测试结果也可能存在虚警误报等情况。因此,研究系统级层次化测试性建模与分析方法,在测试性模型中引进测试对于所检测故障的概率信息,建立一个完整的系统级测试性评级体系,并开发一套相应的计算机软件辅助设计工具,对于降低设计难度,缩短设计周期,减少设计支出,指导设计人员提高可测试性设计方案的水平并制定最优化的故障诊断策略等方面大有裨益。

## 1.2 国内外研究现状

### 1.2.1 系统层次化建模与测试性指标分配技术

高集成、多层次是现代高新技术电子装备的显著特征,多层次复杂设备在航天航空领域、军事武器系统及一些高精密工业集成系统等对系统测试性、维修性与可靠性要求极为严格的重要领域中应用广泛,但是针对板级电路的测试性设计与分析方法无法适应多层次复杂系统的测试性维修性保障要求,因此国外学者及研究机构纷纷投身到多层次复杂系统的测试性建模与分析理论、方法及技术的研究浪潮中,目前成果斐然。20 世纪 80 年代开始,美军就开始重视并深入研究复杂系统的测试性建模方法,经过多年的深入探索,发布了 MIL-STD-2165 军用设备测试性与维修性大纲,为多层次设备的测试性设计工程提供指导; Rahaman 和 Ciani 分别对某机械数控系统和航空电子设备进行分层建模,提出了信息流层次化测试性分析方法<sup>[12, 13]</sup>; Belli 提出了基于事件的层次化故障诊断方法<sup>[14]</sup>; 美国 QSI 公司和 DSI 公司经过多年的设计经验积累,开发

了各自的测试性建模与分析软件，不仅支持系统自顶向下层次化建模，同时支持在系统的任意层级设置各自的故障与测试，将复杂系统分解为不同梯级的简单系统，极大地降低分析难度和计算容量，为军事航天领域复杂设备的测试性建模与分析提供了技术支持。国内在多层系复杂系统的建模与分析技术方面与国外有一定差距，许多研究所和高校相继开展了广泛深入的研究工作。我军先后制定了 GJB431-88 装备层次划分标准和 GJB 2961-1997 修理级别分析标准，对军用设备的设计和维修层次划分提供了模板；龙兵阐述了系统模块化的基本原则，提出了模块化故障诊断方法<sup>[15]</sup>；国防科技大学通过对系统进行层次与模块划分，提出了一种层次化相关性模型的测试性分析方法<sup>[9]</sup>；文献[16]和文献[17]利用 FST 模型描述各层次模块的功能、结构及测试性信息，提出了基于 FST 模型层次化建模方法。

测试性分配问题其实是一个优化问题，通过协调各级单元的指标分配额以满足系统设计指标约束条件，并实现测试成本最低、诊断时间最少等最优化目标。测试性指标分配技术是测试性工程中重要的一环，为制定设计方案提供依据。90 年代初，美军在制定测试性工程体系时，明确了测试性分配的地位，并制定了测试性分配的相关方案，按照系统的层次结构和系统整体指标要求，自顶向下为各级单元计算指标分配额，各级单元根据给定的分配额进行各自的测试性设计，从而降低复杂系统的测试性设计难度，简化设计流程，提高方案的可靠性。国外学者纷纷提出了独到的见解并给出了众多分配方法，文献[18]采用经验分配和拒识率分配等多种分配算法对同一系统进行了测试性分配，并对不同分配方法的差异进行了对比；Junichi Onishi 等人对 ISC（代理约束）方法进行了改进，提出了一种能够精确计算各单元指标分配额的解决方案<sup>[19]</sup>；Payman Dehghanian 等人利用模糊层次分析法求解各级单元影响测试性指标相关因素的权重，根据权重进行测试性分配<sup>[20]</sup>；Ranjan Kumar 等人多层系统的测试性分配问题进行了分析，提出了基于层次遗传算法的指标分配技术<sup>[21]</sup>。我国也在 1995 年制定的 GJB2547 装备测试性大纲中，从性质、要求和流程等方面对测试性分配作出详细的说明和规定。目前我国测试领域常用的测试性分配算法主要包括经验法、等值法、故障率法和加权法四种，许多学者对这些方法的优缺点进行了详尽深入的论述分析，并尝试进行改进完善，王权伟等人将测试性分配问题转化为多目标约束决策问题，提出了线性规划分配模型<sup>[22]</sup>；沈亲沐采用抽象的系统混联模型对测试性分配问题进行了细致的分析，并应用改进的 AHP 实现了复杂系统的测试性分配<sup>[23]</sup>；张琦等人以故障危害性最低、故障诊断率最高和测试费用最少为优化目标，提出了基于 NSGA-II（多目标遗传算法）的测试性分配方法<sup>[24]</sup>。

### 1.2.2 测试性建模与分析技术

通过将近 40 年的探索、总结和积累,美国测试领域对于测试性设计的研究经历了由基于经验的设计到基于结构的设计,再逐步过渡到基于模型的设计三个阶段的发展过程,目前已形成了一套理论完备、技术成熟、能够满足国防军事发展需求的测试性体系。自 20 世纪 80 年代初开始,国外许多设计人员和研究机构对系统测试性设计进行了大量深入的研究,在测试性建模技术与分析方法方面取得了丰硕显著的研究成果,提出了逻辑性模型、结构式模型、信息流模型和故障-信号-测试模型等具有里程碑意义的建模方法。逻辑性模型又称为相关性模型,该利用采用有向图建模,不考虑故障模式和测试方法,只关心系统各单元模块与测试节点之间的关联信息<sup>[25, 26]</sup>,因而不能描述故障与测试的关联信息,其推广应用受限于自身缺陷;结构式模型是在刻画系统组成结构的基础上,为各组成部分加入了故障模式和故障信号的传播方向,从而表现系统各单元模块的连接关系以及故障的传播途径<sup>[27]</sup>,但是该模型不能反映系统功能,不能刻画故障模式与测点测试之间的相关性,其分析结果的可信度不够高;信息流模型将系统功能、故障模式与测点测试等相关信息融合到有向图中,利用系统的测试信息流进行测试性指标的计算,该模型与逻辑性模型和结构式模型相比虽然有一定的进步和创新,但是它不能反映系统的实际结构,设计人员根据自身经验建立的模型可能与系统实际结构存在差异<sup>[28-30]</sup>,从而使分析结果产生偏差,一般应用于对系统测试性的定性分析。美国学者 Pattipati 和 Deb 通过对逻辑性模型、结构式模型和信息流模型的优缺点进行了仔细深入的分析总结,提出了故障-信号-测试(Fault-Signal-Test, FST)模型<sup>[31-36]</sup>,该模型以有向图描述系统的实际结构和功能,同时将系统的故障、在系统中设置的测试节点和测试、系统各单元模块的信号及三者的相关关系相关融合在有向图中,能够定性和定量表达系统的测试性水平,方便设计人员制定和优化测试性设计方案和故障诊断策略。在许多方面, FST 模型具有其它模型无法匹及的优越性,因而引起了众多学者和研究机构的兴趣,其相关建模理论和分析技术日臻完善<sup>[37-39]</sup>。

国内测试领域在系统测试性设计方面的研究起步较晚,在测试性建模技术与分析方法方面,目前大多数学者和研究机构主要致力于研究信息流模型和 FST 模型的改进与应用,虽然相对其它模型来说,这两种模型发展较为完善、应用也比较广泛,但是它们也存在一些瓶颈。不管系统的复杂程度有多高、外界影响有多严重、测试难度有多大,现有的模型都仅考虑了系统的功能而无视系统或测试本身或多或少可能存在的不确定信息,认为测试结果是 100%可靠的,以测试和故障的关系不存在不确定因素为前提进行测试性分析,这样做可能导致模型分析结果与实际系统验证结果产生一定的偏差,在许多关键领域,这种偏差

是不能容忍的；现有模型描述的都是系统静止状态下的特性，对于故障产生、传播及诱发其它故障等动态过程的描述无能为力，也不能表现测试施加的顺序和故障诊断的流程。为解决这些问题，国内学者和研究机构集思广益，展开了激烈的探讨的深入的研究，目前已经取得了一系列突破性成果。为改进 FST 模型的不足，龙兵将不确定信息有机融合到有向图中<sup>[15]</sup>；国防科技大学故障诊断课题组详细阐述了确定性相关矩阵存在的不足与隐患，提出了利用不确定参数描述系统和测试的不确定信息的相关性建模方法<sup>[9]</sup>；文献[40]、[41] 和[42]分别运用 EXPRESS-G 语言、XML 语言和 Web 本体语言对故障-测试相关性、系统与测试的可靠性数据、系统工作与测试维修环境等与关乎系统测试性的信息进行了全面统一的描述，提高了测试性设计的可靠性，为测试性设计所有阶段各部门之间实现信息共享提供了新思路。

### 1.2.3 测试性辅助设计工具

复杂系统的测试性建模与分析是一项庞大的工程，若是没有专门软件工具的辅助，需要耗费较多的时间、资源和成本，并且无法保障系统设备及时稳定地发挥其效用，这在军事国防领域是不允许的。20 世纪 90 年代，国外许多设计人员和研究机构尝试利用计算机开发测试性辅助设计软件工具，以摆脱人工设计的困境，减轻人力负担，提高工作效率，优化设计水平。截至目前，大量优秀的测试性设计辅助工具相继问世，如 Robach 等研究员研发的 CATA（测试性辅助分析计算机系统），美国康涅狄格大学开发的 START（系统测试性分析与诊断工具），美军设计的 DARTS（诊断分析与维修设计工具箱），以色列开发的 AITEST（人工智能测试软件），法军设计的 AGENDA（诊断树自动生成工具）。这些测试性辅助工具都是专门为军方、航空公司等特殊用户制定的，只能用于特定的项目场合，无法满足普通用户群体的需求。为此，美国 DSI 公司和 QSI 公司分别开发了 eXpress（系统健康管理预测、测试性设计与故障诊断工具箱）和 TEAMS（测试性工程与维护系统软件），由于这两款软件通用性强，能够为多领域的用户提供解决方案，因此应用范围最为广泛，无论在军事国防工程还是在民用工业项目，二者都发挥了它们强大的辅助设计功能。

对于系统测试性描述的实现，eXpress 工具箱采用混合诊断模型，而 TEAMS 软件采用 FST 模型，二者都能对复杂多层次系统进行测试性建模与分析，将各种测试性相关信息融合到单元模块中供给用户选择修改，对系统工作过程中可能出现的故障状态进行预测，利用人工智能辅助设计人员指导故障诊断策略，通过图表和数据定性、定量地评价系统的测试性设计水平，提供测试性设计方案的改进方法，并自动生成 FMECA（故障影响度及危害性分析）报告。TEAMS

还拥有 eXpress 所不具备的强大功能, TEAMS 能够实时反馈系统的健康状况, 准确及时地对系统的故障状态进行监测与诊断, 并且支持远程监控和维护。eXpress 和 TEAMS 界面友好、操作简单、方便直观, 并且分析结果准确、及时、可靠, 因此广泛应用于国防军事、航天航空航海以及民用工业等众多领域, 在测试性设计工程发挥着重要的作用, 如美军 F22 战机、TRIDENT 核潜艇和法军 MATRA 系列导弹等装备的测试性设计都是利用 eXpress 完成的<sup>[43, 44]</sup>, NASA 国际空间站与无人机故障诊断项目、波音与西科斯基直升机引擎测试诊断项目、霍尼韦尔公司的电信集成故障预报项目等工程都是依靠 TEAMS 来实现系统测试性建模与分析的<sup>[43, 45]</sup>。

我国由于测试性设计水平比较落后, 软件研发能力与欧美发达国家差距加大, 目前国内测试性辅助设计工具自主开发的机构和成果都屈指可数。北京可维创业科技公司为满足我国军事及民用产品测试性、安全可靠性和维修保障性等设计分析的需求, 与北京航空航天大学合作, 开发了 GARMS (可靠性工程集成系统)<sup>[43]</sup>, 该软件提供系统测试性预计、分配及建模与分析, 打破了我国测试性辅助设计分析工具匮乏的窘境, 目前该软件的主要用户仅限于部分航天军事研究所、少数高校和几家民企, 其推广应用还依赖于自身功能及可靠性的完善。国防科技大学借鉴了 TEAMS 的许多经验, 吸收了国内外测试领域的先进研究成果, 并结合我国相关标准规范, 自主设计了 TADES (测试性设计、分析及评估软件)<sup>[43]</sup>, 该软件以功能较为丰富, 但是其自身可靠性与测试性分析结果的可信度还有待鉴定。

### 1.3 本文的主要研究内容

本文采用 FST 建模方法, 对复杂系统进行系统级层次化测试性建模, 完成系统测试性指标分配、静态特性分析和测试性指标定量计算等工作, 并设计相应的测试性建模与分析软件, 对测试性建模与分析方法进行验证。课题主要研究内容如下:

#### (1) 系统层次划分与测试性指标分配方法研究

研究系统层次划分原则, 根据军事国防工程项目对复杂电子系统测试、诊断与维修的要求, 对系统进行合理的层次划分; 研究如何将系统设计指标要求合理分配给各层级单元, 以指导测试性设计工作, 分别采用故障率法、加权法和综合加权法完成测试性指标分配, 并对三种方法的分配结果进行比较分析。

#### (2) 系统测试性建模与分析方法研究

研究基于 FST 模型的系统测试性建模原理、相关矩阵生成方法、静态特性分析和测试性指标计算流程, 针对基本 FST 模型由于忽略了测试的不确定性而

可能导致不利影响,在基本 FST 模型中添加测试与相应故障之间的不确定信息,提出了融合概率信息的 FST 模型,为制定可靠的并且符合工程实际的测试性设计方案提供技术支持。

### (3) 系统级层次化测试性建模与分析软件设计

设计软件总体架构,利用建模仿真软件 OpenModelica 构建系统级层次化测试性建模平台,并完成 FST 模型测试性分析程序的设计,实现系统级层次化建模、模型信息提取、测试性指标分配和系统测试性分析等功能。

### (4) 建模与分析方法验证

以某野外电子侦察系统为研究对象,分别利用本文设计的软件和商业软件 TEAMS 建立其系统级层次化测试性模型,完成测试性指标分配、顶层系统与各层次单元系统的测试性分析,并与实际测试工程统计数据进行比较,验证本文提出的系统级层次化测试性建模与分析方法的准确性及可靠性。

## 第 2 章 系统层次划分与测试性指标分配方法研究

研究系统层次划分与测试性指标分配方法有助于为测试性设计提供科学可靠的依据，保证测试性设计的可靠性。本章首先分析了系统层次划分的需求，提出了层次划分的原则和方法；然后阐述了测试性指标分配的一般流程并建立了其数学模型，研究了几种常用经典测试性分配算法的原理与过程，并对这几种算法的优缺点进行了对比。

### 2.1 系统层次划分方法

国内外学者普遍这样定义系统：系统是一个具有特定功能的集合体，这个集合体由若干相互作用、相互依赖的单元组成，一个系统又可以作为一个组成单元隶属于一个更大的集合体。系统的概念是相对的，一个实际的系统可以根据功能结构划分的差异将其各个组成单元分为若干不同的层级，不同的层级单元具有不同的功能结构属性，不同层级单元之间相互关联、相互作用；同一层级各单元之间相互独立，没有直接关系<sup>[47]</sup>。

#### 2.1.1 系统层次划分需求分析

按照设备的约定装配层次来划分，我国 GJB431-88 产品层次、互换性标准将装备分为 8 个等级的功能层次，从高到低依次为系统、分系统、装置、机组、单元体、组件、部件和零件，支持设备的研制、采购、生产和维修等活动。按照设备的测试性设计来划分，TEAMS 软件为用户提供 8 种可选的层次等级，包括系统、分系统、外场可更换单元(LRU)、内场可更换单元(SRU)、模块、子模块、元件和故障模式，用户可以根据测试性设计与分析需求选择合适的层次。由于划分的角度不同，对于同一系统，TEAMS 和 GJB431-88 划分的结果必然会有差异，而且两种划分方法的适用场合也不同，GJB431-88 划分方法更适合系统的研制与组装，在系统的测试性设计和维修诊断分析活动中，TEAMS 的划分方法更实用。

在不同维修测试场合，对于系统层次划分的细致程度不同，故障单元的层次定位也不同。考虑单故障的情况，理想条件下，系统一旦发生故障，若是能够将故障直接隔离定位到唯一的故障本质（即最底层电路元件的故障模式），更换故障元件即可使系统恢复正常，这显然是最佳结果；然而在系统实际工作过程中，由于工作环境、测试手段和维修费用及周期等各方面因素的制约，考虑到实际环境的可操作性，将故障隔离定位到唯一的故障本质是不必要也不现实



的，为了保障工程的顺利进行，最为可靠的做法是，根据现场具体情况，对系统进行自顶向下的测试分析，将故障定位到可更换的子系统或者子模块，维修时利用备份的子系统或者子模块直接替换可能发生故障的单元。TEAMS 虽然提供了可以选择的层次等级，但是它进行故障诊断分析时都将故障定位到用户划分的最低层级，这显然不符合维修测试的要求。

为了实现测试维修资源的合理配置，提高测试维修工作的效率，保障系统的长久完好性与即时可用性，根据现场维修条件、测试手段和诊断周期与费用，针对较为复杂的武器装备，GJB 2961-1997 修理级别分析国家军用标准对测试维修活动进行了等级划分，并对各级别的修理级活动负责保障的系统层级单元进行了约定。该标准规定，对于海陆空军用装备均采用三级维修体制，按照自顶向下的顺序，从基层级（O 级，使用现场）、中继级（I 级，修理所）和基地级（D 级，修理厂）三个由低到高的维修级别依次展开；将装备的约定层次与维修级别进行结合，将系统分为三个层级，从高到低依次是外场可更换单元（LRU）、内场可更换单元（SRU）和内场可更换子部件（SSRU），如图 2-1 所示。

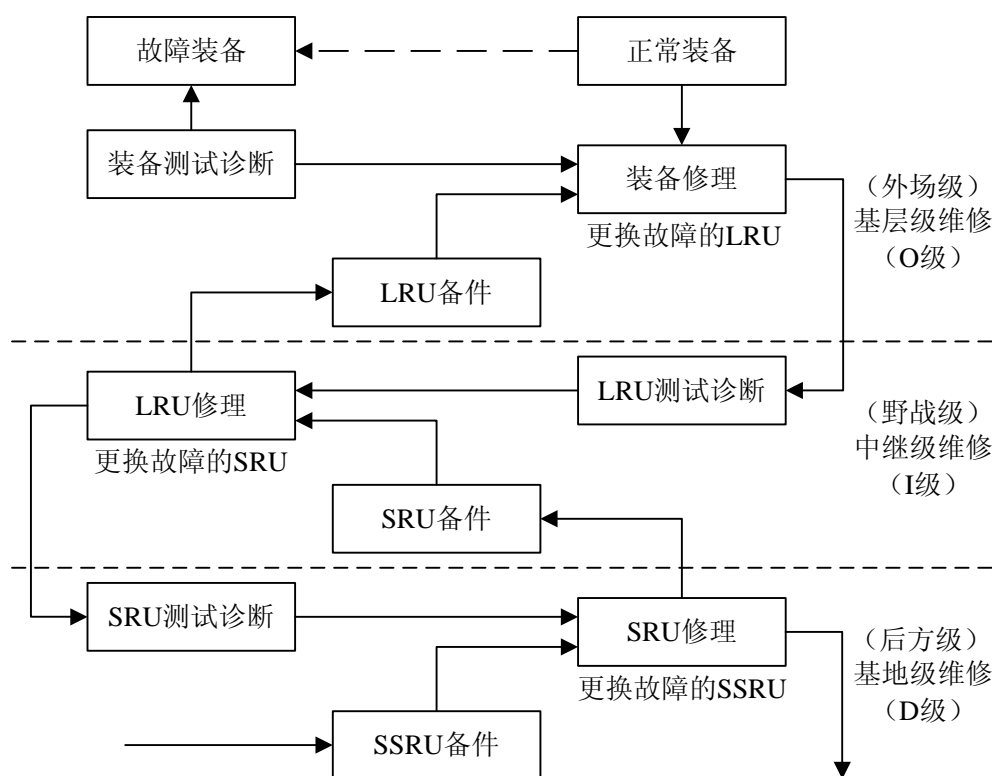


图2-1 三级维修体制示意图

对于大型武器装备的测试维修工作按照从 O 级到 I 级再到 D 级的顺利逐级开展，测试维修条件随着测试维修等级的升高逐步提高，相应的故障诊断隔离精度要求也越来越高。在基层诊断维修时，由于受到测试工具、诊断水平以及

其他条件的限制，基层测试诊断一般都只需要将装备故障定位到 LRU 即可，通过更换可能发生故障的单元，使装备系统快速恢复到正常工作状态。在中继级诊断维修时，测试维修条件相对基层级较好，一般要求将故障定位到 SRU。而在基地级诊断维修时，由于已经具备丰富的测试维修工具和足够的诊断分析水平，一般要求对故障的 SRU 进行详细的测试与分析，将故障定位到 SSRU。

可测试性设计工作服务于测试维修活动，由于测试维修活动对系统层次进行了划分，这就要求在可测试性设计阶段必须对系统进行功能结构层次分解，将系统分解为与测试维修需求相匹配的层次结构，这有这样才能使可测试性设计能够不同场合的测试诊断分析需求。

### 2.1.2 系统层次划分原则

在进行可测试性设计过程中，层次划分的合理性关系到对系统的分析正确与否，同时也关乎武器装备的可测试性、可靠性和安全性。为了使系统具有合理的层次性，确保测试性设计的可靠性满足实际测试维修要求，保证在测试维修过程中各级维修活动能够顺利进行，系统层次划分应遵循一定的标准和原则：

(1) 为符合测试诊断分析过程中对故障模式的推理思维方式，采用自顶向下原则划分。对系统进行层次划分需要自上而下，使各层级对应的故障模式符合诊断逻辑，结构由粗到细，层层递进，清晰明了。当发生某种故障时，系统相关层级就会表现出某种与之对应的征兆，较低层级表现为某些结构单元失效或损坏，较高层级则表现为某些组件功能失常、性能下降，或各功能组件运转不协调等，低层级子故障总是其高层级父故障的特例，高层级父故障则是其低层级子故障的概括。故障诊断推理是按照自顶向下的原则先隔离高层级单元再定位低层级单元，将高层级的故障模式逐级向下分解，这样分解能够表现系统各层级的故障模式的传播方式及其之间的因果关系，可以清晰完备地表达系统的故障模式层次结构，为系统可测试性设计提供指导。

(2) 层次划分不能违背系统各部分功能结构之间的逻辑关系，要根据系统的自身功能结构复杂程度和系统设计分析需求进行。按照由大到小、由浅入深的逻辑，一个复杂系统的总体功能可以划分为若干个具有相对独立的子功能，实现这些子功能的各个单元就构成了一个层级，该层级各功能单元同时为上级（即系统）提供服务以实现系统的整体功能；对某一层级的某个功能来说，该分功能又可以继续向下划分，如此递推逐级分解，得到由最简单的功能结构单元构成的层级，即最低层级。上下层级单元之间具有隶属关系，一个下层级单元隶属于其对应的一个上层级单元，一个上层级单元在其模型中能够体现出其相应的各个下层级单元<sup>[48]</sup>。同一层次各单元之间相互独立，互不干扰。系统

各级单元的功能定义和结构界定要根据系统特性和测试维修的要求进行，这样才能确保测试性设计层次划分的准确性与合理性。

(3) 层次划分深度要适应诊断分析的要求。复杂装备系统一般采用分层诊断、分级维修原则，不同维修场合所要求的故障诊断精度一般都不相同，因此层次分解的细化程度决定于测试维修性的分析要求，系统需要分解到的最低层级对应于系统故障要求隔离到的最小可更换单元所在的层级。按照 GJB 2961-1997 中的相关规定，一个系统划分为 3 个层级，包括 LRU 层级、SRU 层级和 SSRU 层级。在基层级维修时，受限于测试条件和测试手段，故障隔离精度要求仅到 LRU 层级，只需要将待测系统划分到 LRU 级即可，中继级和基地级则分别需要将系统最低层级定位至 SRU 层级和 SSRU 层级。在可测试性设计过程中，也需要如此划分系统层次，根据系统实际特性将其划分为 LRU、SRU 和 SSRU 三个层级，确保在实际维修过程中，既符合各维修级别的维修测试精度，又能缩短测试诊断时间，充分保障装备的完好性。

(4) 各层级的故障模式根据各层级的功能结构特性分别定义。高层级的故障是由于低层级的故障传播而引起的，其本质都是最低层级的故障，不同层级的故障模式表现不一致，各个维修级别定位的可更换单元层级不同，各维修级别关注的故障表现形式也不一样。因此，不妨将同一最低层级故障模式在不同高层级的不同故障征兆形式看作不同的故障模式，即对各个层级分别定义故障模式，不同层级本质相同的故障模式相互独立，各层级单元的故障率都能够获取。对各个层级分别进行可测试性设计，进行不同级别的测试维修活动时，只需要分析与该测试级别相对应的可更换层级单元的故障模式即可。这样就可以降低测试维修规模，既能准确定位故障，又能有效提高测试效率。

按照上述标准和原则在可测试性设计过程中对系统进行层次划分，并按照所划分的层次结构对各个层级分别进行可测试性设计，不仅能够清晰合理地描述系统的功能层次结构和各单元的测试性信息，满足不同测试维修场合不同测试维修等级对故障隔离定位的精度，而且能够减轻测试诊断分析的负荷，提高工作效率，缩短测试维修周期，维持系统的安全、稳定与可靠，最大程度地保障系统的使用效能。

## 2.2 测试性指标分配方法

在制定系统测试性设计方案时，需要首先明确系统各层级组成单元的测试性定量指标要求，然后将其作为依据以指导测试性设计。测试性指标分配就是在测试性设计的初步阶段，遵循指定的原则，利用一定的方法，依据系统的层次结构，按照自顶向下的顺序将系统的测试性定量指标要求逐级分配给系统各

个组成单元的过程。由于系统各层级组成单元的测试性能能够影响系统整体的测试性，测试性分配的好坏决定了按照方案制定的系统的测试性能能否达到设计要求，因此有必要对测试性指标分配技术进行深入研究。

### 2.2.1 测试性分配的流程及其数学模型

测试性工程中，用测试性参数来描述系统故障检测和隔离能力等测试性特征，其量化值称为测试性指标，测试性指标能够衡量系统的可测试性设计水平。由于系统测试性特征的不同，可以对系统定义多个不同的测试性参数，根据系统的故障检测能力特征，可以定义故障检测时间(FDT)、故障检测率(FDR)和虚警率(FAR)等参数；根据系统故障隔离能力特征，可以定义故障隔离时间(FIT)、故障隔离率(FIR)和误拆率(FFR)等参数。

系统可测试性设计的根本目的和基本要求是测试能够准确及时地对系统发生的故障进行检测与隔离，准确性体现在系统具有较高的 FDR 和 FIR，及时性表现为 FDT 和 FIT 足够低，这四个参数在测试维修活动中具有普遍性、可观测性、可控性和可验证性，一般作为衡量系统测试性水准的首选参数集。考虑到测试性设计阶段，最关注的参数是故障检测率和隔离率，故障检测和隔离时间只有经过对系统进行实际测试维修过程才能统计整理得到，因此可以忽略故障检测和隔离时间，仅针对故障检测率和隔离率进行分析计算。

故障检测率(FDR)是指在规定时间内按照指定的测试方法，能够正确检测到的故障总数与被测单元实际发生的故障总数的比值；故障隔离率(FIR)是指在规定时间内按照指定的测试方法，能够正确将检测到的故障区分至所要求故障分辨率水平以内的故障总数与检测到的故障总数的比值。通常 FDR 和 FIR 都用百分数表示。

对于一个复杂的多层系统，其同一层级的不同单元或者不同层级的不同单元之间在结构、功能、性能和故障模式等方面都或多或少存在差异，若将系统的整体测试性指标直接作为各层级单元的指标是不合理的，这样的测试性设计也是不可靠的，无法达到任务所要求的测试性水平。正确的做法是，根据系统各层级单元的功能结构特性和测试维修需求，采用合理的方法确定各层级单元的测试性定量指标，测试性指标分配的具体流程如图 2-2 所示。

在分配系统测试性指标时，需要来明确系统的体系结构和目标单元所在的层级，根据上一节系统层次划分的原则将系统分为 LRU 层级、SRU 层级和 SSRU 层级，由于故障最小只需要定位到 SSRU 层级，对系统进行测试性分析时，需要研究的最小模型并不是 SSRU，而是由数个 SSRU 组成的 SRU，因此测试性指标分配的最低层级为 SRU 层级，即只需要将系统的测试性指标逐级向下分配

到 LRU 层级和 SRU 层级即可，SSRU 层级不需要分配。系统测试性指标分配层级结构图如图 2-3 所示。

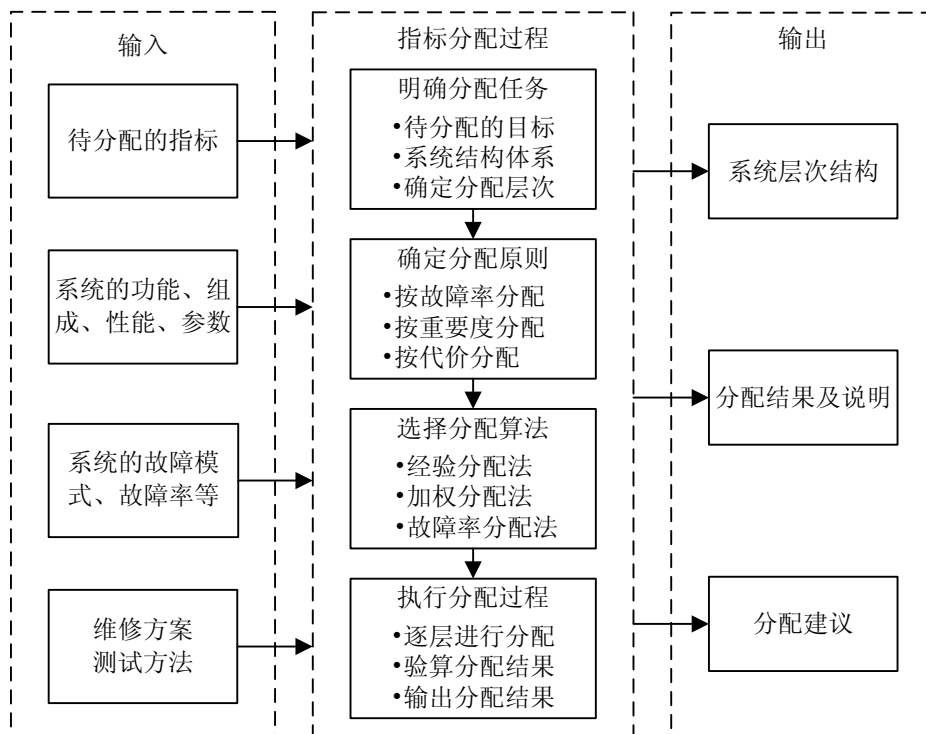


图 2-2 测试性指标分配流程图

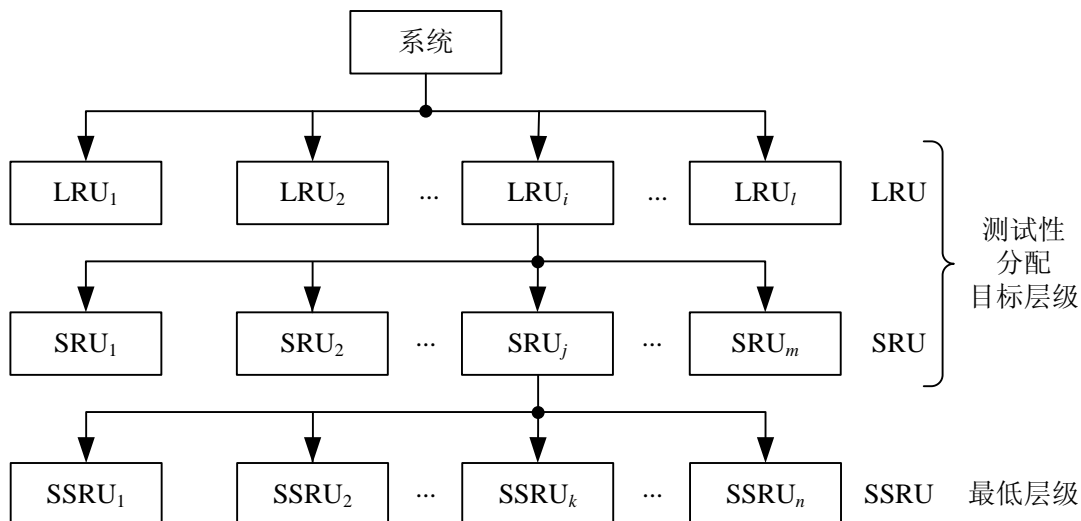


图 2-3 测试性分配层次结构图

系统整体测试性指标到局部单元测试性指标的定量分配，需要通过建立能够体现整体指标和局部单元指标的关系数学模型来实现，单个层级测试性指标定量分配问题可以用式(2-1)至(2-4)所示的数学模型来表示。

$$P_i = f_1(P_{sr}, K_i) \quad (i=1, 2, \dots, N) \quad (2-1)$$

$$P_s = f_2(P_1, P_2, \dots, P_i, \dots, P_N) \quad (2-2)$$

$$0 \leq P_i \leq 1 \quad (i=1, 2, \dots, N) \quad (2-3)$$

$$P_{sr} \leq P_s \leq 1 \quad (2-4)$$

式(2-1)描述的是整体指标的分配过程，其中， $P_{sr}$ 为待分配的变量，是测试性设计所需要达到的系统整体指标（故障诊断率或者故障隔离率量值）； $K_i$ 是第*i*个单元的测试性综合权重； $N$ 代表该层级可更换单元的总数； $f_1$ 是该层级的指标分配函数； $P_i$ 是第*i*个单元的指标分配额，其值与 $K_i$ 成正比。

式(2-2)描述的是分配结果的验证过程， $f_2$ 是指标分配的验证函数， $P_s$ 为利用各单元的指标分配额验算得到的系统整体指标。

式(2-3)和式(2-4)测试性分配的约束条件，各单元的指标分配额必须在区间[0, 1]之内，且系统整体指标的验算值不能比设计要求值小。只有完全满足约束条件的分配结果才是合理有效的。

测试性设计中通常考虑的测试性指标影响因子主要包括故障出现的频率、故障对系统性能的影响程度、故障检测与隔离难度、平均故障修复时间和测试维修费用，这五个影响因子权重的计算方法如表 2-1 所示。

表2-1 分配单元影响因子权重的计算方法

权重	影响因子	计算公式
$K_i^1$	故障率	$K_i^1 = \lambda_i / \sum_{i=1}^N \lambda_i$
$K_i^2$	故障影响程度	$K_i^2 = F_i / \sum_{i=1}^N F_i$
$K_i^3$	故障检测与隔离难度	$K_i^3 = T_i^{-1} / \sum_{i=1}^N T_i^{-1}$
$K_i^4$	平均故障修复时间	$K_i^4 = M_i^{-1} / \sum_{i=1}^N M_i^{-1}$
$K_i^5$	测试维修费用	$K_i^5 = C_i^{-1} / \sum_{i=1}^N C_i^{-1}$

表 2-1 中， $\lambda_i$ 、 $F_i$ 、 $T_i$ 、 $M_i$ 和 $C_i$ 分别为第*i*个单元的故障率、故障数目、测试时间、平均故障修复时间和测试维修成本，在这些参数的量值不可获取的情况下，分配单元影响因子权重也可以通过专家评分的方法来定性量化，评分方法如下：将所有的影响因子都分为 5 个不同的重要等级，用数值 1-5 表示针对一种影响影子各单元的权重，数值越大表示权重越大，数值大小根据各单

元的实际特性来确定，通常情况下， $K_i^1$ 和 $K_i^2$ 分别与故障率和系统受故障影响程度成正比，而 $K_i^3$ 、 $K_i^4$ 、 $K_i^5$ 则分别与测试时间、平均故障修复时间和测试维修费用成反比。

在测试性分配的数学模型中，分配函数 $f_i$ 是求解分配结果的关键，分配函数根据所采用的分配算法来确定，分配算法在下一节中说明。分配单元的综合权重 $K_i$ 是分配函数 $f_i$ 的核心， $K_i$ 的确定需要首先考量该分配单元各影响因子的权重，然后根据各影响因子的相对重要程度再加权求和，可以用式(2-5)求解分配单元的综合权重 $K_i$ 。

$$\begin{cases} K_i = \sum_{j=1}^5 (K_j^0 \times K_i^j) \\ \sum_{j=1}^5 K_j^0 = 1 \quad 0 \leq K_j^0 \leq 1 \end{cases} \quad (2-5)$$

其中 $K_i^j$ 表示第 $i$ 个单元的第 $j$ 个影响因子的权重； $K_j^0$  ( $1 \leq j \leq 5$ )为第 $j$ 个影响因子的重要度，即所有影响因子中，第 $j$ 个影响因子所占的比重。

求得各单元的指标分配额之后，需要根据分配验证函数对分配结果进行核算，验证分配结果是否服从约束条件，如不满足，需要调整修正分配结果，直到满足约束条件为止。故障检测率和隔离率指标的验证函数分别如式(2-6)和(2-7)所示。

$$\gamma_{\text{FDs}} = \frac{\sum_{j=1}^N (\gamma_{\text{FDj}} \cdot \lambda_j)}{\sum_{j=1}^N \lambda_j} \quad (2-6)$$

$$\gamma_{\text{FIs}} = \frac{\sum_{j=1}^N \gamma_{\text{FIj}} \cdot (\gamma_{\text{FDj}} \cdot \lambda_j)}{\sum_{j=1}^N (\gamma_{\text{FDj}} \cdot \lambda_j)} \quad (2-7)$$

式(2-6)和式(2-7)中， $\gamma_{\text{FDj}}$ 和 $\gamma_{\text{FIj}}$ 分别表示第 $j$ 个单元分配所得故障检测率和故障隔离率； $\lambda_j$ 表示第 $j$ 个单元的故障率； $\gamma_{\text{FDs}}$ 和 $\gamma_{\text{FIs}}$ 分别表示由各单元分配所得故障检测率和故障隔离率验算得的系统整体的故障检测率和故障隔离率。

### 2.2.2 测试性指标分配算法

本节研究三种常用的测试性分配算法：加权分配算法、故障率分配算法和综合加权分配算法。

### (1) 加权分配算法

加权分配算法综合了需要考虑的所有影响因子，依据各单元综合权重  $K_i$  的数值大小按线性比例进行分配。权重  $K_i$  越大，单元所得分配额就越大。加权分配算法的故障检测率和隔离率指标的分配函数分别如式(2-8)和式(2-9)所示。

$$\gamma_{FDi} = \frac{\gamma_{FDsr} \cdot K_i \cdot \sum_{j=1}^N \lambda_j}{\sum_{j=1}^N K_j \cdot \lambda_j} \quad 1 \leq i \leq N \quad (2-8)$$

$$\gamma_{Fli} = \frac{\gamma_{Flsr} \cdot K_i \cdot \sum_{j=1}^N (\gamma_{FDj} \cdot \lambda_j)}{\sum_{j=1}^N K_j \cdot (\gamma_{FDj} \cdot \lambda_j)} \quad 1 \leq i \leq N \quad (2-9)$$

其中， $K_i$  为第  $i$  个单元的综合权重； $\gamma_{FDsr}$  和  $\gamma_{Flsr}$  均为待分配变量，分别表示设计所要求的故障检测率和故障隔离率指标； $\gamma_{FDi}$  和  $\gamma_{Fli}$  均为分配额，表示第  $i$  个单元分配所得故障检测率和故障隔离率指标。

加权分配算法的流程如图 2-4 所示。

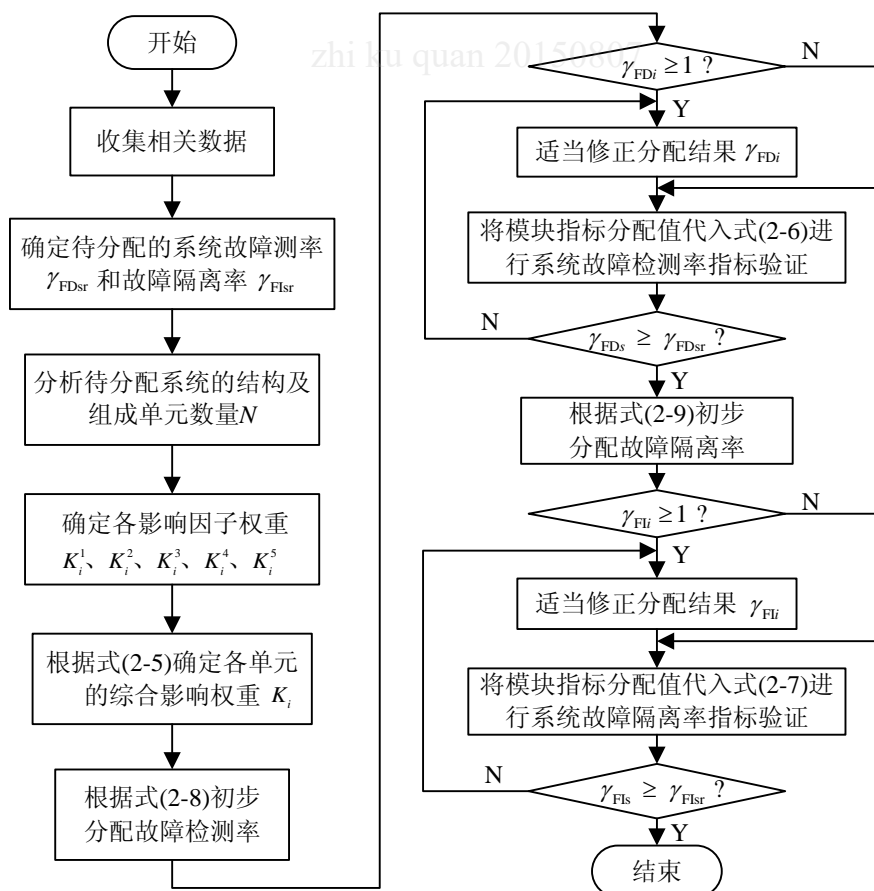


图 2-4 加权分配算法的流程图



由于考虑了多种因素对系统测试性的影响，在应用加权分配算法之前，需要考察系统测试维修需求并据此评价影响系统测试性的各因子的相对重要性，通过收集整理分析相关资料和数据确定各单元各影响因子的权重，从而计算出各单元的综合权重，再根据分配函数得到各单元的分配指标。由于加权分配结果与综合权重  $K_i$  的大小成线性正比，若  $K_i$  大小悬殊， $K_i$  较大的单元分配所得的指标可能会出现大于 1 的情况，即不满足约束条件(2-3)，所以经常需要人为对分配结果调整修正，以完全满足约束条件。

### (2) 故障率分配算法

故障率分配算法仅考虑故障率这一影响系统测试性的因素，将单元故障率的量值作为单元的综合权重进行测试性分配。故障率分配算法的故障检测率和隔离率指标的分配函数只要将式(2-8)和式(2-9)所示的加权分配函数中的  $K_i$  替换为  $\lambda_i$  即可，分别如式(2-10)和式(2-11)所示。

$$\gamma_{FDi} = \frac{\gamma_{FDsr} \cdot \lambda_i \cdot \sum_{j=1}^N \lambda_j}{\sum_{j=1}^N \lambda_j^2} \quad 1 \leq i \leq N \quad (2-10)$$

$$\gamma_{Fli} = \frac{\gamma_{Flsr} \cdot \lambda_i \cdot \sum_{j=1}^N (\gamma_{FDj} \cdot \lambda_j)}{\sum_{j=1}^N \lambda_j \cdot (\gamma_{FDj} \cdot \lambda_j)} \quad 1 \leq i \leq N \quad (2-11)$$

故障率分配算法从根本上可以看作加权分配算法的简化和特例，二者的分配流程类似，只是所考虑的影响因子数目不一样而已，因此故障率分配算法也存在分配结果可能不合理的问题，其处理方法与加权分配算法相同。

### (3) 综合加权分配算法

当系统结构较为复杂，组成单元的数目较多且各单元的功能特性都存在差异时，如果应用加权分配算法得到的分配结果不合理，那么对分配结果进行人为调整修正就显得十分繁琐而且是不现实的。综合加权分配算法约束了分配结果的上限，解决了加权分配算法分配结果可能不合理的问题。

综合加权分配算法利用单元的故障检测率分配额与其综合权重的正比关系，构建如式(2-12)所示的等比例关系式：

$$\frac{\gamma_{FDi} - \gamma_{FDsr}}{K_i - K_{FDsr}} = \frac{\gamma_{FDmax} - \gamma_{FDsr}}{K_{max} - K_{FDsr}} \quad (2-12)$$

其中， $K_{max}$  表示所有单元综合权重中的最大值； $\gamma_{FDmax}$  表示综合权重为  $K_{max}$  的单元分配所得故障检测率； $K_{FDsr}$  为对应系统故障检测率设计要求值的系统权重。

对式(2-12)进行变换, 可得

$$\gamma_{FDi} = \frac{\gamma_{FDmax} - \gamma_{FDsr}}{K_{max} - K_{FDsr}} (K_i - K_{FDsr}) + \gamma_{FDsr} \quad 1 \leq i \leq N \quad (2-13)$$

联立式(2-6)和式(2-13), 得到

$$K_{FDsr} = \sum_{i=1}^N \lambda_i K_i / \sum_{i=1}^N \lambda_i \quad (2-14)$$

式(2-13)和式(2-14)即为综合加权分配算法的故障检测率分配函数。同理, 可求得故障隔离率的分配函数, 见式(2-15)和式(2-16)。

$$\gamma_{Fli} = \frac{\gamma_{Flmax} - \gamma_{Flsr}}{K_{max} - K_{Flsr}} (K_i - K_{Flsr}) + \gamma_{Flsr} \quad 1 \leq i \leq N \quad (2-15)$$

$$K_{Flsr} = \sum_{i=1}^N \gamma_{FDi} \lambda_i K_i / \sum_{i=1}^N \gamma_{FDi} \lambda_i \quad (2-16)$$

其中,  $\gamma_{Flmax}$  表示综合权重为  $K_{max}$  的单元分配所得故障隔离率;  $K_{Flsr}$  为对应系统故障隔离率设计要求值的系统权重。

$\gamma_{FDmax}$  和  $\gamma_{Flmax}$  可以视为单元指标分配值的上限, 即单元可实现的最大故障检测率和隔离率, 这两个参量需要设计人员在分配前根据系统单元测试性指标的可实现性人为给定。由于综合加权分配算法对分配结果的最大值进行了约束, 因此可以避免分配结果出现大于 1 的情况。

综合加权分配算法的流程也与加权分配算法类似, 但是在分配前需要人为指定分配结果上限, 然后计算各单元的综合权重, 将综合权重最大的单元指标分配值设置为分配额的上限值, 然后根据分配函数进行指标分配。

### 2.2.3 分配算例

本节通过一个简单案例对三种分配算法进行分析比较, 该案例主要是将系统指标分配到 LRU 层级的 6 个单元, 如图 2-5 所示, 故障检测率、隔离率的设计要求值和最大可能实现值分别为 95.0%、83.0%和 98.0%、92.0%, 系统各单元影响因子的相关数据和综合权重见表 2-2, 各影响因子的相对重要度相同。

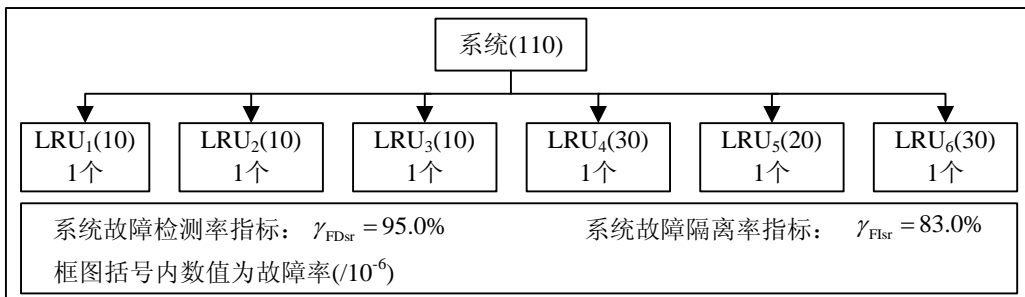


图 2-5 待分配系统实例

表2-2 待分配系统各单元影响因子数据

单元	数量	$\lambda_i/10^{-6}$	$F_i$	$T_i$	$M_i$	$C_i$	$K_i$
LRU1	1	10	3	2	15	20	0.6293
LRU2	1	10	3	1	25	40	0.6281
LRU3	1	10	3	1	5	20	1.3554
LRU4	1	30	5	1	30	50	0.8566
LRU5	1	20	7	3	15	20	0.8341
LRU6	1	30	5	3	30	50	0.8566
系统	6	110	—	—	—	—	—

分别采用三种分配算法对该案例进行测试性分配,故障率分配结果如表 2-3 所示, 加权分配与综合加权分配结果如表 2-4 所示。

表2-3 故障率分配结果

单元	数量	$\lambda_i/10^{-6}$	$\gamma_{FDi}(\%)$		$\gamma_{Fi}(\%)$	
			计算值	修正值	计算值	修正值
LRU1	1	10	41.80	91.00	31.92	75.00
LRU2	1	10	41.80	91.00	31.92	75.00
LRU3	1	10	41.80	91.50	31.92	75.00
LRU4	1	30	125.40	98.00	95.77	90.00
LRU5	1	20	83.60	93.00	63.85	80.00
LRU6	1	30	125.40	98.00	95.77	90.00
系统	6	110	95.00	95.18	83.00	84.31

表2-4 加权分配和综合加权分配结果

单元	数量	$K_i$	加权分配算法				综合加权分配算法	
			$\gamma_{FDi}(\%)$		$\gamma_{Fi}(\%)$		$\gamma_{FDi}(\%)$	$\gamma_{Fi}(\%)$
			计算值	修正值	计算值	修正值		
LRU1	1	0.6293	73.56	93.00	60.84	80.00	93.99	79.91
LRU2	1	0.6281	73.42	93.00	60.72	80.00	93.98	79.89
LRU3	1	1.3554	158.43	98.00	131.03	92.00	98.00	92.00
LRU4	1	0.8566	100.12	96.00	82.81	89.00	95.24	83.69
LRU5	1	0.8341	97.49	95.50	80.64	88.00	95.12	83.32
LRU6	1	0.8566	81.42	94.00	67.34	80.00	94.36	81.03
系统	6	—	95.00	95.07	83.00	85.07	95.00	83.00

对照表 2-3 和表 2-4, 不难看出, 若按故障率分配算法分配, 则各单元的指标分配值与其故障率一致, 故障率越大的单元分配结果也就越大; 若按加权或者综合加权分配算法分配, 则各单元的指标分配值与其综合权重一致; 三种分配算法都能得到合理的分配结果。由于只考虑故障率因子, 故障率分配算法的过程比加权或者综合加权分配算法相对简单, 后两种考虑的因素全面, 分配结

果较为合理，但是分配过程比较繁琐。综合加权分配算法增加了分配上界约束条件，因此分配结果不需要修正，而故障率和加权分配算法都存在分配结果越界的问题，需要人为调整。

## 2.3 本章小结

本章首先分析了测试维修工程对于系统层次划分的需求，提出了系统层次划分方法和原则；然后建立测试性分配的数学模型，研究了加权法、故障率法和综合加权法三种常用测试性分配算法的原理与过程，从故障率、故障严酷度、故障检测与隔离难度、平均修复时间和测试维修费用等角度确定各单元测试性指标分配额；并利用三种算法分别对一个简单案例进行了分配，三种分配算法在分配原则和分配流程方面存在各自的优缺点，但均能得到合理的分配结果。

## 第 3 章 测试性建模与分析方法研究

测试性建模与分析工作就是依据待评估方案对系统测试性指标进行定性分析与定量计算来评价和验证测试性设计方案的可实现性，为测试性设计方案的改善提供指导性意见，从而确保系统的测试性能够满足相关维修测试的要求。本章首先研究了 FST 建模与分析方法；然后针对 FST 模型存在的不足，提出了融合概率信息的 FST 模型的测试性建模方法；最后，分别利用两种 FST 模型对一个简单系统进行建模与分析，从建模过程、分析流程和分析结果等角度对二者的异同进行了比较。

### 3.1 FST 模型

对系统进行测试性分析之前必须明确系统的测试性信息，包括系统的故障信息（系统可能会出现的故障模式）、测试信息（系统中添加的测试）以及故障-测试之间的关联信息（何种故障模式能够被何种测试所检测）等，建立测试性模型就是进行收集、整理系统测试性信息并在模型中直接或间接体现这些信息的过程。FST 模型能够直接表现系统的故障信息、测试信息，并通过反映故障在系统中的传播特性间接地体现故障-测试之间的关联信息，由于其建模与分析过程简单并且结果准确可靠，目前已经广泛应用于测试性研究领域。

#### 3.1.1 FST 模型的基本思想

在 FST 模型中，可以对系统组成单元设置完全故障模式和功能故障模式，在各组成单元中可以添加其影响的信号；可以在各单元之间的连接线上添加测试节点，在测试节点中可以设置相应测试及其所检测的信号；根据有向图所反映的故障（信号）在系统中的传播特性，从模型中提取出故障与信号、信号与测试之间的关联信息，进而生成故障-测试关联矩阵，再利用一定的算法来处理关联矩阵中蕴含的测试性信息，从而求解出系统的静态特性以及故障检测率与故障隔离率等量化指标。

故障模式、信号和测试是分析 FST 模型必备的三要素，下面对这三个要素进行详细说明。

(1) 故障模式：即系统的单元模块的故障形式，此处的系统是层次化系统中某一层级的某一个单元，可以是子系统、LRU 或者 SRU 等，各单元模块的故障模式的相互独立的。FST 模型只考虑单故障的情况，即同一时刻只可能出现一种故障模式，由于实际系统在工作过程中在同一时刻出现单一故障模式的

概率远远大于出现多种故障模式的概率，因此这种假设是合理的。从故障影响作用的范畴，将各单元模块的故障模式分为完全故障与功能故障；完全故障是指系统中某个单元模块发生故障之后，不仅会造成该单元模块的失效或损坏，甚至会影响其他单元模块功能的正常发挥，从而导致整个系统功能失常，这种故障模式用符号  $G$  表示；功能故障是指某各单元模块的故障发生以后，仅表现为该单元的功能失常，不会波及其他单元模块，对整个系统产生的影响很小，可以忽略不计，系统仍然能够正常工作，这种故障模式用符号  $F$  表示。由于对每个单元模块都设置了  $G$  和  $F$  两种故障模式，相当于每个故障单元的故障数目为 2，因此系统的故障总数为系统单元模块数目之和的 2 倍。

(2) 信号：在 FST 模型中这是一个泛称，与单元模块和测试相关联，可以是单元模块的某种功能、某种特性、某种状态或者某个参数，只要能够表征故障的发生，都可以称之为信号，用符号  $S$  表示。由于系统所在层级不同，其中的单元模块的规模也有差异，但是信号的关联不受对象的规模限制，对象大到一个复杂的装备系统，小到某个部件或者某个零件皆可；信号的定义不拘泥于形式，既可以是定量的表示，也可以是定性的描述，到底是何种形式取决于信号所关联对象的性质；信号是 FST 模型中连接故障模式与测试的纽带，既与系统中的故障相关联，又与其中添加的测试相关联，利用这种映射关系能够方便地获取故障-测试相关矩阵。

(3) 测试：与信号相关联，用来检测系统中的信号，进行测试性分析时需要利用信号与测试的关联关系，是测试性模型中不可或缺的元素，用符号  $t$  表示。测试施加于测试节点处，一个测试测试节点可以设置多个测试，一个测试可以检测多个信号，一个信号可以与多个测试相关联。可测试性设计的主要任务就是测试节点的设置和测试的施加，测试节点和测试数目的选择合理与否决定了可测试性设计水平的高低。如何合理设置数量少、效率高的测试节点和测试，以实现测试性指标的最大化，是测试性建模与分析的主要研究内容。

### 3.1.2 FST 模型的描述方法

作为一种测试性分析的系统可视化建模技术，对于经过可测试性设计的系统，FST 模型采用单元模块节点和测试节点来描述其组成单元，虽然都是一种抽象的图形化表现方式，但是二者所描述的内容是有差异的，在建立模型与分析模型的过程中，不能将二者混为一谈。单元模块节点代表系统中的 LRU、SRU 或者 SSRU 等，用方框表示；测试节点即系统中添加的测试节点，用圆圈表示。在 FST 模型中，单元模块节点与单元模块节点之间、单元模块节点与测试节点通过带有单箭头的连接线（即有向边）两两连接起来，以表示故障或者信号在

系统中的传播路径和传播方向，这种传播方向是不可逆的，因此有向边连接的两个节点之间的关系是固定不变的。

**FST** 模型通过建立一个能够精确能够反映各单元模块之间的相互连接作用关系和测试节点与各单元模块之间检测与被检测的关系的可视化有向图，只需要对有向图进行结构分析就能提取出隐含在模型中的故障模式与测试之间的关联信息。

在形式上，**FST** 模型中各组成单元可以描述如下：

(1) 有限的单元模块集： $\mathbf{M} = \{m_1, m_2, m_3, \dots, m_l\}$ ， $|\mathbf{M}| = l$ ；单元模块  $m_l$  代表系统中的 **LRU**、**SRU** 或者 **SSRU** 等，每个  $m_l$  都具有完全故障和功能故障两种模式；

(2) 有限的测试节点集： $\mathbf{TP} = \{tp_1, tp_2, tp_3, \dots, tp_m\}$ ， $|\mathbf{TP}| = m$ ；集合 **TP** 中的各元素（即测试节点）相互独立；

(3) 有限的系统信号集： $\mathbf{S} = \{s_1, s_2, s_3, \dots, s_k\}$ ， $|\mathbf{S}| = k$ ；集合 **S** 中的各元素（即信号）相互独立；

(4) 有限的测试集： $\mathbf{T} = \{t_1, t_2, t_3, \dots, t_n\}$ ， $|\mathbf{T}| = n$ ；集合 **T** 中各元素（即测试）互不干扰，关联各自检测的故障信号，并隶属于各自所在的测试节点；

(5) 有限的有向边集： $\mathbf{E} = \{e_{ij}\}$ ；元素  $e_{ij}$  表示有向图中连接节点  $i$  和节点  $j$ ，且由节点  $i$  指向节点  $j$  的有向边；

(6) 一个测试节点  $tp_m$  包含的有限测试集： $\mathbf{SP}(tp_m)$ ，此处  $\mathbf{SP}(tp_m) \subseteq \mathbf{T}$ ，并且  $\mathbf{SP}(tp_1) \cup \mathbf{SP}(tp_2) \cup \dots \cup \mathbf{SP}(tp_m) = \mathbf{T}$ ；

(7) 一个单元模块  $m_i$  影响的有限信号集： $\mathbf{SM}(m_i)$ ，此处  $\mathbf{SM}(m_i) \subseteq \mathbf{S}$ ，并且  $\mathbf{SM}(m_1) \cup \mathbf{SM}(m_2) \cup \dots \cup \mathbf{SM}(m_i) = \mathbf{S}$ ；

(8) 一个测试  $t_j$  能够检测的有限信号集： $\mathbf{ST}(t_j)$ ，此处  $\mathbf{ST}(t_j) \subseteq \mathbf{S}$ ；

(9) 系统结构有向图： $\mathbf{DG} = \{\mathbf{M}, \mathbf{TP}, \mathbf{E}\}$ ，**DG** 是一个有向无环图，反映系统的组成结构、各部分的连接关系以及在系统中添加的测试节点。

可以用一个简单的有向图充分体现上述结构单元，以展现 **FST** 模型的结构组成，如图 3-1 所示。

从图 3-1 中的 **FST** 模型可以快速得到一些测试性信息，如该系统包含  $m_1$ 、 $m_2$  和  $m_3$  三个单元模块，信号集  $\mathbf{SM}(m_1)$  与单元模块  $m_1$  相关联，测试集合  $\mathbf{SP}(tp_1)$  中的测试都设置在测试节点  $tp_1$  处，信号集  $\mathbf{ST}(t_j)$  中的信号都能被测试  $t_j$  所检测。一个完整的 **FST** 模型与一个电路系统的原理框图类似，但是所传达的信息量不同，**FST** 模型反映了系统的功能结构信息，又描述了系统的测试性信息，完全

故障和功能故障作为固有属性集成在单元模块中，信号作为故障的传播形式与影响它的单元模块和能够检测它的测试关联，测试节点描述测试施加的位置，所有的测试信息有机地融合在反映系统结构和信号方向的有向图中。FST 模型建模过程简单，模型结构清晰，模型信息丰富，即使是复杂的多层系统，也能以简明直观的形式表现出来，便于开展测试性分析工作。

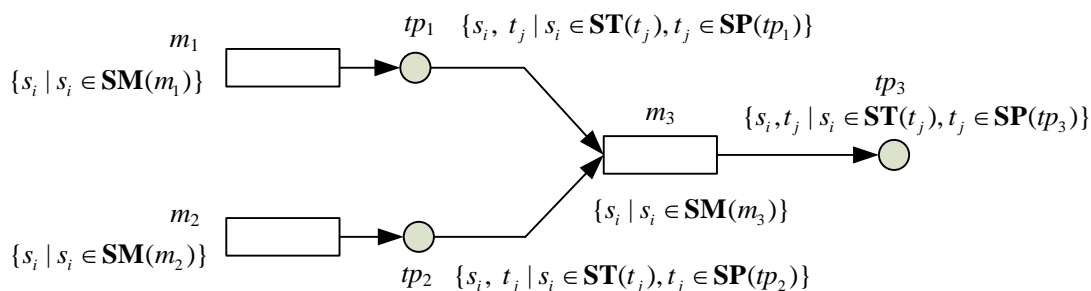


图3-1 FST模型的结构示意图

### 3.1.3 FST 模型的建立过程

多层复杂系统也能用 FST 模型来描述。按照具体的测试维修场合，将复杂的系统分解简化，建立各个层级各个单元系统的 FST 模型，对各个单元系统分别进行测试性分析，如此可以将故障隔离定位到系统的任意层级，满足不同场合的测试维修要求。在建立系统的测试性模型之前，需要明确测试诊断的精度需求，根据系统的功能结构对系统进行合理的层次划分，这样才能得到合理的分析诊断结果。FST 建模与分析具体流程如图 3-2 所示。

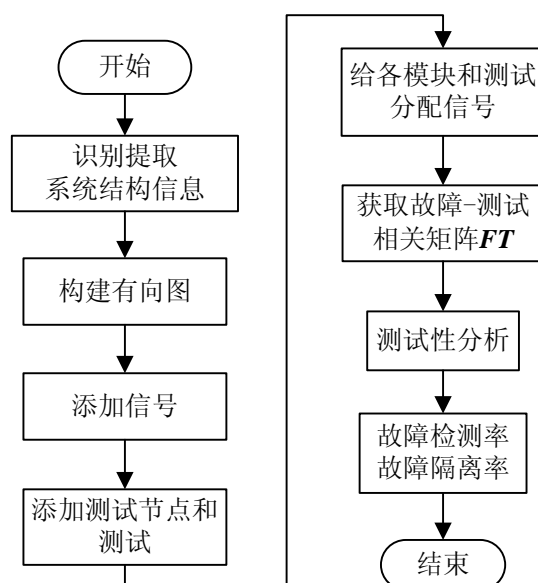


图3-2 FST建模与分析流程图



首先，从已经划分好层次的系统中选择需要分析的层级单元系统，对其功能和结构进行分析，确定其中包含的单元模块，根据实际系统内部结构利用有向边将各个单元连接起来，建立在结构和组成单元方面均与系统一致的有向图。接着，根据各单元模块的特性定义该系统有限的信号集，并将各个被影响的信号添加到相应的单元模块中，信号的设置要求研发人员掌握系统的原理和特性，如果信号设置不合理会影响到测试性模型分析结果的准确性和可靠性。然后，制定系统测试性设计方案，根据系统故障模式设计相应的测试，并确定测试施加的位置，即测试节点之所在，测试节点一般设置在单元模块的输出端口，测试节点和测试的选取合理与否，是系统的测试性指标是否能够满足测试性设计要求的關鍵，需要研发人员具备相关测试性设计经验。至此，就完成了 FST 模型的建立，接下来就可以对系统进行测试性分析，计算故障检测率和隔离率等指标，判断是否满足可测试性设计要求，若是不满足，则需要对测试节点和测试进行改进，直至测试性指标符合相关设计要求为止。

## 3.2 FST 模型的测试性分析

基于 FST 模型的测试性分析的内容主要包括故障与测试相关性矩阵的生成、静态特性分析以及测试性指标的计算。后两者都依赖于故障-测试相关矩阵，因此相关性矩阵的生成是开展测试性分析的关键；静态特性分析根据相关矩阵推导出系统的固有测试性，固有测试性包括未检测故障集、模糊集和冗余测试集等，固有测试性是对测试性设计方案定性评估，能够为修正和改进测试性设计提供指导；得到相关矩阵和系统固有测试性之后可以进行测试性指标计算，求解系统故障检测率与隔离率等参数的量化值，通过对比计算结果与设计要求指标，对测试性设计方案的水平进行定量、客观的评价。

### 3.2.1 相关矩阵生成

在 FST 模型中，在不考虑虚警、漏检等特殊情况的前提下，故障模式和与之相关联的测试之间存在因果作用关系。如果故障  $f_i$  与测试  $t_j$  相互关联，则存在这样的情况：由于  $f_i$  的发生会导致  $t_j$  不通过；若  $t_j$  通过则能够确定  $f_i$  没有发生。这说明  $f_i$  与  $t_j$  之间的状态是可以相互推导的，二者之间的因果关系是确定的，这种关系称为故障与测试的相关性，用相关性矩阵来描述。

假设 FST 模型中有  $m$  个单元模块，并添加了  $n$  个测试，由于每个单元模块都有两种故障模式（完全故障和功能故障），则故障数目为  $2m$ ，该模型对应的相关矩阵为  $FT_{2m \times n}$ ，具体形式如式(3-1)所示。

$$\mathbf{FT}_{2m \times n} = \begin{pmatrix} ft_{11} & ft_{12} & \cdots & ft_{1n} \\ ft_{21} & ft_{22} & \cdots & ft_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ft_{2m1} & ft_{2m2} & \cdots & ft_{2mn} \end{pmatrix} \quad (3-1)$$

$\mathbf{FT}_{2m \times n}$  是一个二维矩阵，行向量  $\mathbf{F}_i = [ft_{i1}, ft_{i2}, \dots, ft_{in}]$  代表模型中所有测试对故障  $f_i$  激励信号的响应，即所有测试对故障  $f_i$  的检测结果，体现了故障  $f_i$  的检测难度；列向量  $\mathbf{T}_j = [ft_{1j}, ft_{2j}, \dots, ft_{mj}]^T$  代表模型中测试  $t_j$  对所有故障激励信号的响应，即测试  $t_j$  对所有故障的检测结果，反映了测试  $t_j$  的检测能力。 $\mathbf{FT}_{2m \times n}$  还是一个二值矩阵，其任意元素  $ft_{ij}$  的取值非0即1，即测试对故障激励信号的响应只有0和1两种取值，分别对应测试通过和测试不通过。 $ft_{ij} = 1$  即测试  $t_j$  不通过，说明故障  $f_i$  可以被测试  $t_j$  检测出，则测试  $t_j$  与故障  $f_i$  相关联； $ft_{ij} = 0$  即测试  $t_j$  不通过，说明故障  $f_i$  无法被测试  $t_j$  检测出，则测试  $t_j$  与故障  $f_i$  不相关。

FST 模型相关性矩阵的生成具有许多难点，包括几下几个方面：

(1) 节点之间的每条有向边都可能传播多个单元模块的故障信号，各个故障信号关联的单元模块可能不同，利用有向边后续测试节点对故障进行定位前，需要对各个单元模块做可达性分析；

(2) 任意单元模块都包含完全故障模式和功能故障模式，二者具有相同的故障传播方式，都能沿有向边向后续节点传播，并被可达测试节点中的测试检测到。但由于传播能力存在差异，因此二者的可达测试集也不一样，完全故障可以沿有向边任意传播，并被其所有可达测试节点中的所有测试所检测；而功能故障只能沿有向边局部传播，是否能够被其可达测试节点中的测试所检测，取决于它影响的信号是否与测试相关联。因此，必须对完全故障和功能故障进行识别判断；

(3) 当系统功能结构较为复杂且组成单元和故障模式较多时，其 FST 模型中的故障、信号和测试数量巨大，并且两两关系十分复杂，由于有向图结构是通过单元模块（故障）与信号、测试与信号这两组关系间接反映测试与故障的关联关系的，因此相关性矩阵不能直接从中读取，所以分析有向图结构以获取故障与测试相关性的过程较为繁琐。

为了获取故障-测试相关性矩阵  $\mathbf{FT}_{2m \times n}$ ，需要对 FST 模型有向图结构中的单元模块和信号分别进行可达性分析，引入两个二值、二维辅助分析矩阵：单元模块与测试可达性矩阵  $\mathbf{MT}_{m \times n}$  和单元模块-信号-测试相关性矩阵  $\mathbf{MST}_{m \times n}$ 。

$\mathbf{MT}_{m \times n} = (mt_{ij})_{m \times n}$  描述了各单元模块到各测试的可达性。 $mt_{ij} = 1$  表示单元模

块  $m_i$  能够沿有向边传播至测试  $t_j$  所在测试节点,  $mt_{ij}=0$  表示在单元模块  $m_i$  与测试  $t_j$  所在测试节点之间没有任何有向通路。以任意模块  $m_i$  为起点, 沿有向边对整个结构进行深度或者广度搜索遍历, 取得  $m_i$  的可达测试节点集合, 将其中各测试节点的包含的测试集合并起来就构成了  $m_i$  的可达测试集, 根据该集合可以确定  $MT_{m \times n}$  矩阵的第  $i$  行元素, 依次类推就能得到完整的  $MT_{m \times n}$  矩阵。

$MST_{m \times n} = (mst_{ij})_{m \times n}$  描述单元模块、信号与测试三者之间的关联信息, 是为了辅助区分完全故障和功能故障而定义的。若单元模块  $m_i$  影响的信号集  $SM(m_i)$  与测试  $t_j$  能够检测的信号集  $ST(t_j)$  有交集, 即  $SM(m_i) \cap ST(t_j) \neq \emptyset$ , 则  $mst_{ij}=1$ ; 若单元模块  $m_i$  影响的信号集  $SM(m_i)$  与测试  $t_j$  能够检测的信号集  $ST(t_j)$  没有公共信号, 即  $SM(m_i) \cap ST(t_j) = \emptyset$ , 则  $mst_{ij}=0$ 。

由于故障模式与单元模块存在一一对应关系, 因此故障模式继承了单元模块的所有性质, 单元模块影响的信号集也就是其对应故障影响的信号集, 单元模块的可达测试集也就是其对应故障的可达测试集,  $MT_{m \times n}$  矩阵也能表达故障的可达性,  $MST_{m \times n}$  亦能表达故障与信号、测试之间的关系。用  $f_{iG}$  和  $f_{iF}$  表示分别单元模块  $m_i$  的完全故障模式和功能故障模式, 则二者与测试  $t_j$  相关的判断条件如下:

(1) 若测试  $t_j$  包含于单元模块  $m_i$  的可达测试集, 则  $f_{iG}$  与  $t_j$  相关,  $f_{iG}$  和  $t_j$  的相关关系可以用式(3-2)表示:

$$ft_{i(G)j} = mt_{ij} \quad (3-2)$$

(2) 若测试  $t_j$  包含于单元模块  $m_i$  的可达测试集, 并且  $t_j$  关联的信号集  $ST(t_j)$  与  $m_i$  影响的信号集  $SM(m_i)$  有交集, 则  $f_{iF}$  与  $t_j$  相关,  $f_{iF}$  和  $t_j$  的相关关系可以用式(3-3)来描述:

$$ft_{i(F)j} = mt_{ij} \times mst_{ij} \quad (3-3)$$

于是, 在得到  $MT_{m \times n}$  矩阵与  $MST_{m \times n}$  矩阵之后, 根据式(3-2)和(3-3)就能获取故障与测试关联矩阵  $FT_{2m \times n}$ 。

### 3.2.2 静态特性分析

静态特性分析的对象是未检测故障集、模糊集和冗余测试集等系统的固有测试性。

未检测故障是指在系统中已发生但是施加的所有测试都检测不到的故障; 在相关性矩阵中元素均为零的行向量对应的故障即为未检测故障; 所有未检测

故障构成的集合就是未检测故障集。为了改进测试方案以提高故障检测率，应增加能够检测未检测故障的测试，直到相关矩阵中任意行向量的元素不全为零。

系统中由于某些故障虽然能够被所加测试检测出，但是他们具有相同的测试特征，相互之间无法区分、隔离，这样的故障组称为模糊组；在相关矩阵中完全相同的行向量对应的故障模式构成一个模糊组，模糊组可能不存在或者不止一个，这完全由测试方案决定；所有模糊组构成的集合称为模糊集。为了提高故障隔离率，改善测试方案，应该添加能够区分模糊集中故障模式的测试，尽可能减少相关性矩阵中相同的行向量数。

冗余测试可以是无法检测出系统中的任何故障的测试，也可以是由于系统任意的故障具有完全相同的检测结果的测试；在相关性矩阵中元素均为零的列向量对应的测试，以及具有完全相同的列向量对应的测试组成的集合就是冗余测试集。对于冗余测试集中毫无故障检测能力测试应该剔除，至于具有相同故障检测结果的测试保留一个即可，这样可以提高测试利用率，节约测试消耗。

利用相关性矩阵进行静态特性分析的方法如图 3-3 所示。

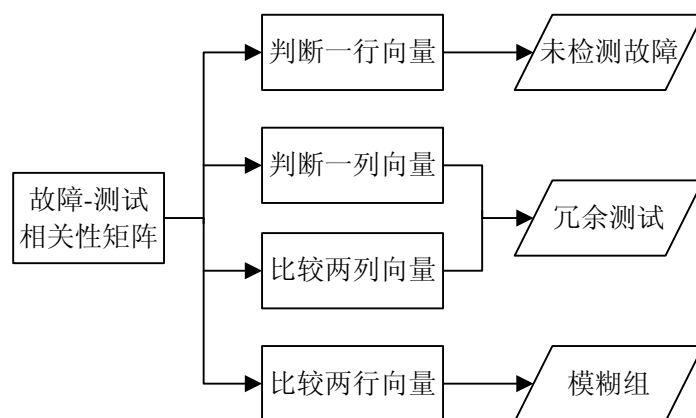


图3-3 静态特性分析示意图

### 3.2.3 测试性指标计算

测试性分析的根本目的是判断系统的测试性是否能达到设计使用要求，故障检测率和故障隔离率是最难体现系统测试性水平的两个参数，测试性指标计算主要是根据故障-测试相关性矩阵和静态特性分析的结果，对故障检测率和故障隔离率进行定量计算，为评价系统测试性提供依据。

#### (1) 故障检测率(FDR)

根据故障检测率的定义，FDR 等于能够正确检测到的故障总数  $N_D$  占被测单元实际发生的故障总数  $N_T$  的百分比。被测单元可以是系统层次结构中的任意层级的单元系统，也可以是整个系统；虚警不包括在故障检测数  $N_D$  和故障总数

$N_T$  中。FDR 可以根据式(3-4)求解。

$$\text{FDR} = \frac{N_D}{N_T} \times 100\% \quad (3-4)$$

式(3-4)为 FDR 的统计模型,适用于能够得到故障总数和故障检测数等相关统计数据的系统。FST 模型中,故障总数取决于单元模块数,用故障总数减去未检测故障集中的故障数即为故障检测数,因此,式(3-4)也适用于 FST 模型故障检测率的计算。

## (2) 故障隔离率(FIR)

根据故障隔离率的定义, FIR 等于能够正确将检测到的故障区分至所要求分辨率  $L$  以内的故障总数  $N_L$  占检测到的故障总数  $N_D$  的百分比。分辨率  $L$  是指测试方法的故障模式区分水平或故障隔离模糊度,简称模糊度,即系统能够容忍的最大模糊组的故障总数,如  $L=2$  表示最多允许 2 个故障不能被给定测试所区分。FIR 可以根据式(3-5)求解。

$$\text{FIR} = \frac{N_L}{N_D} \times 100\% \quad (3-5)$$

式(3-5)为 FIR 的统计模型,适用于能够得到故障隔离数和故障检测数等相关统计数据的系统。FST 模型中,根据单元模块总数、未检测故障集中的故障数和模糊集中的故障数可以计算出  $N_D$  和  $N_L$  的,因此,可以利用式(3-6)求解 FST 模型的故障隔离率。

FDR 和 FIR 除了统计模型,还有预计模型,如式(3-6)和式(3-7)所示。

$$\text{FDR} = \frac{\lambda_D}{\lambda} = \frac{\sum_{i=1}^{N_D} \lambda_i}{\sum_{i=1}^{2m} \lambda_i} \times 100\% \quad (3-6)$$

其中,  $\lambda_i$  为常数,代表第  $i$  个故障模式的故障率;  $\lambda_D$  表示检测到的故障集的故障率总和;  $\lambda$  表示故障集中的故障率总和。

$$\text{FIR} = \frac{\lambda_L}{\lambda_D} = \frac{\sum_{i=1}^{N_L} \lambda_{Li}}{\lambda_D} \times 100\% \quad (3-7)$$

其中,  $k$  代表能够隔离到的在模糊度  $L$  范围内的故障总数;  $\lambda_{Li}$  表示能够隔离到的在模糊度  $L$  范围内的故障集中,第  $i$  个故障模式对应的故障率;  $k$  代表第  $i$  个故障模式的故障率。

由于 FDR 和 FIR 的预计模型不仅要求故障隔离数和故障检测数可获取,还需要对系统进行 FMECA 分析,或者查找系统各个组成部分的故障率参数。在计算 FDR 和 FIR 时根据相关数据获取难度,统计模型和预计模型二者择一即可。

### 3.3 融合概率信息的 FST 模型

在实际测试过程中，由于存在诸多可能影响测试结果的因素，如系统结构的复杂性、电磁干扰、外界环境影响以及测试工具、测试手段固有的误差等等，测试结果不可能完全可靠，故障与测试之间的因果关系因具有不确定性而变得十分复杂，即测试通过并非代表系统一定不存在故障，也可能是漏检所致；测试不通过也并非意味着系统一定不正常，也可能是虚警所致。而在前面阐述的 FST 建模与分析方法，是基于系统中不存在虚警和漏检等不确定性因素即测试结果确定可靠的假设条件的，因此相关性矩阵所刻画的故障模式与测试之间的关联关系是确定的，即测试通过与否完全取决于系统是否正常，这会使计算得到的指标高于现实情况，增加测试性设计的不可靠性。为了使测试性分析更接近实际情况，以指导研究人员设计更为可靠的测试性方案，有必要研究能够描述并分析存在不确定测试性信息的建模方法，本节对 FST 模型进行了改进，提出了融合概率信息的 FST 模型。

#### 3.3.1 融合概率信息的 FST 模型的描述方法

测试结果确定可靠的条件下，测试对故障的检测概率取值只能为 0 或 1，考虑测试结果的不确定性之后，就变为连续区间[0, 1]，因此测试的不确定性可以用测试对故障的检测概率来表示，概率取值越接近 1，表示测试结果的不确定性越小，反之，则表示测试结果的不确定性越大；测试对故障的检测概率取值为 0 时，表示测试与故障不相关。FST 模型中，完全故障和功能故障是作为单元模块的固有属性来考虑的；类似的，不妨将测试对故障信号的检测概率也作为测试节点中测试的固有属性，即将测试性概率信息融合到测试节点的测试中，每个测试的属性中不仅需要关联其能够检测的故障信号，还需要设置其能够检测出故障信号的概率信息，故障与测试的因果关系不再是确定的，如此不仅完好地保留了 FST 模型原有的功能，而且能够描述测试的不确定性，这样改进的 FST 模型称为融合概率信息的 FST 模型，其结构示意图如图 3-4 所示。

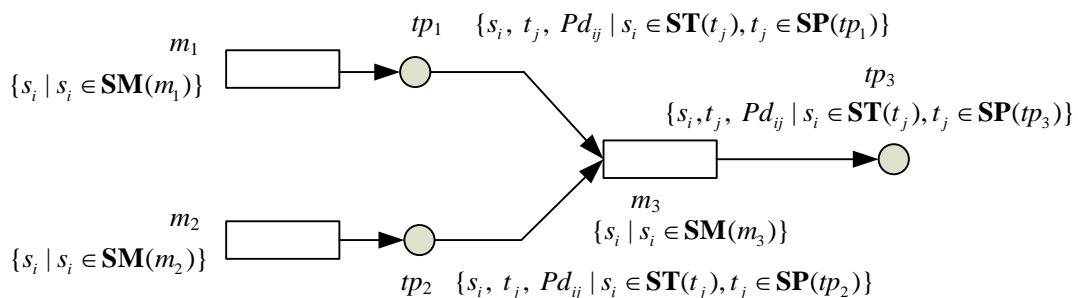


图3-4 融合概率信息的FST模型的结构示意图

融合概率信息的 FST 模型的结构描述与基本的 FST 模型基本一致，唯一的差异就是测试有无概率信息；融合概率信息的 FST 模型的分析也是根据有向图所反映的故障传播特性，对故障与信号、信号与测试之间的关系进行分析，从而推导出故障与测试之间的关联关系，但是融合概率信息的 FST 模型中测试与信号之间存在不确定性，因此测试与故障之间的相关关系也具有不确定性，之前的相关性矩阵是不具备这种能力的，需要对其进行改进。

假设所有故障和测试都是二值的，故障只具有两种状态，发生和不发生；测试也存在两种结果，通过和不通过。用符号  $T$  和  $F$  表示故障和测试的二值性，对于故障， $T$  代表故障发生， $F$  代表故障不发生；对于测试， $T$  表示测试不通过， $F$  表示测试通过。假设故障  $f_i$  与测试  $t_j$  相互关联，以条件概率的形式来二者的不确定性因果关系，主要包括以下四种情形：

(1)  $P(t_j = T | f_i = T)$ ：表示在故障  $f_i$  发生的条件下测试  $t_j$  不通过的概率，即测试  $t_j$  对故障  $f_i$  的检测概率；

(2)  $P(t_j = F | f_i = T)$ ：表示在故障  $f_i$  发生的条件下测试  $t_j$  通过的概率，即测试  $t_j$  对故障  $f_i$  的漏检概率；

(3)  $P(t_j = T | f_i = F)$ ：表示在故障  $f_i$  不发生的条件下测试  $t_j$  不通过的概率，即测试  $t_j$  对故障  $f_i$  的虚警概率；

(4)  $P(t_j = F | f_i = F)$ ：表示在故障  $f_i$  不发生的条件下测试  $t_j$  通过的概率，即测试  $t_j$  对无故障  $f_i$  的检测概率。

对于任意相关的测试与故障，他们都存在上述四种不确定关系，但是在测试性设计阶段，人们最关心的是测试能够正确检测已发生的故障的概率，即测试对故障的检测概率  $P(t_j = T | f_i = T)$ 。于是，对 FST 模型中的测试  $t_j$  引入一个不确定参数  $Pd_{ij}$ ，定义下：

$$Pd_{ij} = P(t_j = T | f_i = T) \quad (3-8)$$

$Pd_{ij}$  表示的是测试  $t_j$  对故障  $f_i$  的检测概率，其值与从 FST 模型中提取的相关性矩阵  $FT_{2m \times n}$  有密切关系，若  $FT_{2m \times n}$  中的元素  $ft_{ij} = 0$ ，即测试  $t_j$  和故障  $f_i$  不相关，则  $Pd_{ij} = 0$ ；若  $ft_{ij} = 1$ ，即测试  $t_j$  和故障  $f_i$  相关，则  $0 < Pd_{ij} \leq 1$ ，此时  $Pd_{ij}$  的取值由故障的复杂性、测试的可靠性以及外界环境等不确定因素共同决定。

将测试对故障的检测概率和故障-测试性关系矩阵结合起来，定义一个概率信息相关性矩阵  $D_{2m \times n}$ ，用其元素  $d_{ij}$  来定量表示测试  $t_j$  和故障  $f_i$  的不确定性，

$D_{2m \times n}$  和  $d_{ij}$  定义如下：

$$\mathbf{D}_{2m \times n} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{2m1} & d_{2m2} & \cdots & d_{2mn} \end{pmatrix} \quad (3-9)$$

$$d_{ij} = Pd_{ij} \cdot ft_{ij} \quad (3-10)$$

概率信息相关矩阵  $\mathbf{D}_{2m \times n}$  与相关矩阵  $\mathbf{FT}_{2m \times n}$  结构相同，意义相近，二者均为二维矩阵，均表示故障于测试的因果相关关系，行向量均表示故障的复杂性和检测难度，列向量都能反映测试的检测能力。二者也存在一些差异， $\mathbf{FT}_{2m \times n}$  是一个二值矩阵，其各元素表示的故障与测试因果关系是确定的，测试必定能够检测出与其相关的故障；而  $\mathbf{D}_{2m \times n}$  是一个多值矩阵，其各元素包含了概率信息，体现了测试的不确定性。

### 3.3.2 融合概率信息的 FST 模型的测试性分析

与基本的 FST 模型相似，对融合概率信息的 FST 模型的分析是从概率信息相关矩阵生成、静态特性分析和测试性指标计算三个方面展开的。概率信息相关矩阵生成也是通过解析模型中蕴含的所有测试性信息推导测试与故障的因果关系，在获取概率信息相关矩阵  $\mathbf{D}_{2m \times n}$  之前，需要确定不确定参数  $Pd_{ij}$  和相关矩阵  $\mathbf{FT}_{2m \times n}$ ，相关矩阵  $\mathbf{FT}_{2m \times n}$  依照上一节的方法生成得到，不确定参数  $Pd_{ij}$  的确定比较繁琐，可以根据设计经验以及虚警率的统计值进行合理设定。静态特性分析与上一节的方法完全一样，都依赖于相关矩阵  $\mathbf{FT}_{2m \times n}$ ，通过分析  $\mathbf{FT}_{2m \times n}$  行向量与列向量的特点，推导出未检测故障集、模糊集和冗余测试集等固有测试性。测试性指标的计算根据静态特性分析结果推导出检测出和隔离出的故障总数，考虑到存在不确定信息，利用预计模型求解故障检测率和隔离率的定量值。下面详细论述融合概率信息的 FST 模型的测试性指标计算方法。

假设当系统发生故障  $f_i$  时，进行了  $n$  次测试，统计各个测试的输出结果，以集合  $\mathbf{H} = \{A_1, A_2, \dots, A_n\}$  表示故障  $f_i$  被单个测试检测到的事件集，其元素  $A_j$  代表故障  $f_i$  发生时测试  $t_j$  结果为不通过的事件，即  $A_j = \mathbf{H}(t_j = T | f_i = T)$ ，由于各个测试之间是相互独立，因此集合  $\mathbf{H}$  中的各个元素也是相互独立的。将针对故障  $f_i$  的检测率用  $\xi_{FDi}$  表示，其意义为当故障  $f_i$  发生时，系统中至少有一个测试能够正确报警的概率， $\xi_{FDi}$  表示如下：

$$\xi_{FDi} = P(A_1 \cup A_2 \cup \cdots \cup A_n) \quad (3-11)$$

根据概率加法公式，上式等价于：



$$\begin{aligned} \xi_{FDi} = & \sum_{j=1}^n P(A_j) - \sum_{1 \leq j < k \leq n} P(A_k A_j) + \sum_{1 \leq j < k < h \leq n} P(A_j A_k A_h) + \\ & \cdots + (-1)^{n-1} P(A_1 A_2 \cdots A_n) \end{aligned} \quad (3-12)$$

又因为事件  $A_1, A_2, \dots, A_n$  相互独立，即对任意整数  $k$  ( $1 < k \leq n$ )，任意  $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ ，满足  $P(A_{i_1} A_{i_2} \cdots A_{i_k}) = P(A_{i_1}) P(A_{i_2}) \cdots P(A_{i_k})$ ，所以

$$\begin{aligned} \xi_{FDi} = & \sum_{j=1}^n P(A_j) - \sum_{1 \leq j < k \leq n} P(A_j) P(A_k) + \sum_{1 \leq j < k < h \leq n} P(A_j) P(A_k) P(A_h) + \\ & \cdots + (-1)^{n-1} \prod_{j=1}^n P(A_j) \end{aligned} \quad (3-13)$$

对于事件  $A_j$ ，其概率表示的就是测试  $t_j$  和故障  $f_i$  的不确定性，即

$$P(A_j) = d_{ij} \quad (3-14)$$

故  $\xi_{FDi}$  可由概率信息相关矩阵求得：

$$\xi_{FDi} = \sum_{j=1}^n d_{ij} - \sum_{1 \leq j < k \leq n} d_{ij} d_{ik} + \cdots + (-1)^{n-1} \prod_{j=1}^n d_{ij} \quad (3-15)$$

由于故障检测 FDR 和故障检隔离 FIR 的统计模型没有考虑测试的不确定性，因此在计算融合概率信息的 FST 模型的测试性指标时，需要采用二者的预计模型。FDR 和 FIR 分别采用式(3-16)和式(3-17)进行计算求解。

$$FDR = \frac{\lambda_D}{\lambda} = \frac{\sum_{i=1}^{N_D} \lambda_i \cdot \xi_{FDi}}{\sum_{i=1}^{2m} \lambda_i} \times 100\% \quad (3-16)$$

$$FIR = \frac{\lambda_L}{\lambda_D} = \frac{\sum_{i=1}^{N_L} \lambda_{Li} \cdot \xi_{FDi}}{\sum_{i=1}^{N_D} \lambda_i \cdot \xi_{FDi}} \times 100\% \quad (3-17)$$

### 3.4 模型实例分析

本节利用某天馈组件对上述两种FST模型的建模与分析过程进行说明，并将两种方法得到的测试性指标计算结果作对比，以验证本章提出的融合概率信息的FST模型的正确性和可行性。

某天馈组件由合路开关  $m_1$  和  $m_5$ 、衰减器  $m_2$ 、限幅器  $m_3$ 、低噪放  $m_4$  和  $m_6$  等6个单元模块构成，其中设置了6个测试节点，首先，根据系统的实际结构，利用有向边将各单元模块和测试节点连接起来，建立图3-5所示的FST模型。

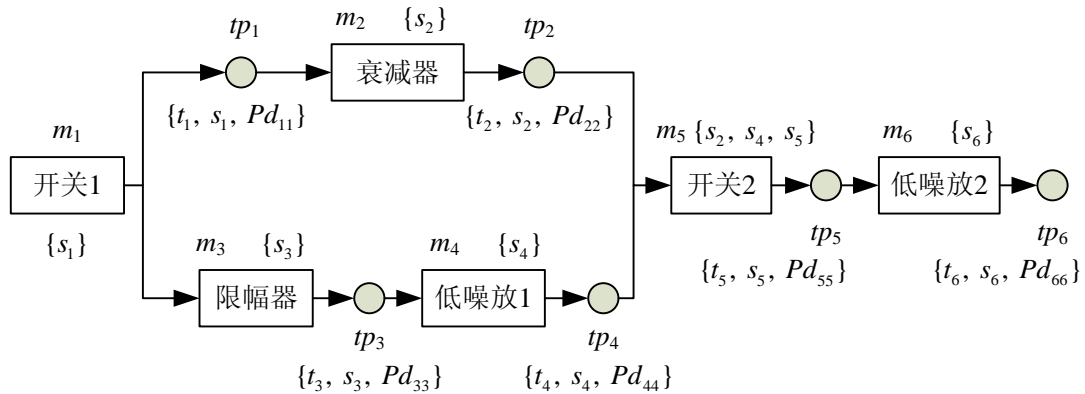


图3-5 某天馈组件的FST模型

接着，根据各单元模块的功能结构和故障征兆等基本属性，定义信号集  $\mathbf{S} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ ，并分配到各个单元模块，信号的定义及其关联的单元模块如表3-1所示。

表3-1 某天馈组件信号的定义

信号	单元模块	定义
$s_1$	$m_1$	合路开关1的输出电流
$s_2$	$m_2$	衰减器的功率衰减量
$s_3$	$m_3$	限幅器输出信号的电压波幅
$s_4$	$m_4$	低噪放1的放大倍数
$s_5$	$m_5$	合路开关2的输出电流
$s_6$	$m_6$	低噪放2的放大倍数

根据系统各单元模块输入输出的结构特征和测试维修需求，定义测试集  $\mathbf{T} = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ ，并为各个测试分配其检测的信号，测试节点、测试与信号的关系见表3-2。

表3-2 某天馈组件测试与信号的关联关系

测试节点	测试	信号
$tp_1$	$t_1$	$s_1$
$tp_2$	$t_2$	$s_2$
$tp_3$	$t_3$	$s_3$
$tp_4$	$t_4$	$s_4$
$tp_5$	$t_5$	$s_5$
$tp_6$	$t_6$	$s_6$

然后，读取并分析有向图包含的信息，逐个分析单元模块完全故障和功能故障与测试的因果关系，从而构建相关性矩阵。以衰减器 $m_2$ 为例，与 $m_2$ 关联的信号为 $s_2$ ，下面考虑 $m_2$ 的两种故障模式 $m_2(G)$ 和 $m_2(F)$ 与各测试之间的关系。根据故障信号的传播方向，衰减器 $m_2$ 的可达测试节点为 $tp_2$ 、 $tp_5$ 和 $tp_6$ ，由完全故障的传播影响特性可知，这三个测试节点中包含的所有测试构成了 $m_2(G)$ 的可达测试集，即测试 $t_2$ 、 $t_5$ 和 $t_6$ 都与 $m_2(G)$ 相关。对于功能故障 $m_2(F)$ ，其可达测试集与完全故障 $m_2(G)$ 相同，但是与 $m_2(F)$ 相关的测试必须是 $m_2(F)$ 可达测试集中与 $m_2(F)$ 具有相同关联信号的测试， $m_2(F)$ 的可达测试集中只有 $t_2$ 与 $m_2(F)$ 有公共关联信号 $s_2$ ，因此只有 $t_2$ 与 $m_2(F)$ 相关，包括 $t_5$ 和 $t_6$ 在内的其他测试与 $m_2(F)$ 都不相关。这样就得出了衰减器 $m_2$ 的两种故障模式与各个测试的相关关系，按照此法对分别其他单元模块一一分析，然后将所有单元模块的分析结果组合起来就能得到该天馈组件的故障-测试相关矩阵，如表3-3所示。

表3-3 故障-测试相关矩阵

编号	故障	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$
$f_1$	$m_1(G)$	0	1	1	1	1	0
$f_2$	$m_1(F)$	0	1	0	0	0	0
$f_3$	$m_2(G)$	1	1	1	1	1	1
$f_4$	$m_2(F)$	0	0	0	0	0	1
$f_5$	$m_3(G)$	0	0	0	1	1	0
$f_6$	$m_3(F)$	0	0	0	1	0	0
$f_7$	$m_4(G)$	1	0	0	1	1	0
$f_8$	$m_4(F)$	1	0	0	0	0	0
$f_9$	$m_5(G)$	0	0	1	1	1	0
$f_{10}$	$m_5(F)$	0	0	1	0	0	0
$f_{11}$	$m_6(G)$	0	0	0	0	1	0
$f_{12}$	$m_6(F)$	0	0	0	0	1	0

融合概率信息的FST模型需要在基本FST模型中添加测试的不确定信息，其中的故障模式、信号与测试均与基本的FST模型相同，其概率信息相关性矩阵的获取需要先读取模型中各个测试与对于各个故障信号的检测概率，再将基本FST模型的相关性矩阵的各元素乘上对应的概率信息即可，最后求解得到的概率信息相关性矩阵如表3-4所示。此处测试对故障的检测概率是根据实际经验得到的估计值，并非精确的统计值，但由于一般测试的漏检率与虚警率都不大于

5%，所以可以利用估计值近似替代统计值。

表3-4 故障-测试概率信息相关矩阵

编号	故障	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$
$f_1$	$m_1(G)$	0	0.99	0.99	0.98	0.99	0
$f_2$	$m_1(F)$	0	0.96	0	0	0	0
$f_3$	$m_2(G)$	0.97	0.99	0.97	0.99	0.98	0.95
$f_4$	$m_2(F)$	0	0	0	0	0	0.91
$f_5$	$m_3(G)$	0	0	0	0.96	0.98	0
$f_6$	$m_3(F)$	0	0	0	0.94	0	0
$f_7$	$m_4(G)$	0.95	0	0	0.96	0.99	0
$f_8$	$m_4(F)$	0.95	0	0	0	0	0
$f_9$	$m_5(G)$	0	0	0.97	0.96	0.92	0
$f_{10}$	$m_5(F)$	0	0	0.92	0	0	0
$f_{11}$	$m_6(G)$	0	0	0	0	0.93	0
$f_{12}$	$m_6(F)$	0	0	0	0	0.95	0

根据该系统的FMECA报告得到系统各个单元模块的故障率，如表3-5所示。

表3-5 各单元模块的故障率( $\lambda_i/10^{-6}$ )

故障率	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$\lambda_6$
数值	10	10	10	30	20	30

在模糊度为1的条件下，根据基本FST模型和融合概率信息的FST模型计算得到的测试性指标见表3-6所示。

表3-6 两种FST模型的测试性指标计算结果

测试性指标	FST 模型	融合概率信息的 FST 模型
FDR(%)	100.00	99.42
FIR(%)	83.33	83.12

由表3-6可以看出，两种FST模型的测试性指标计算结果相近，但是融合概率信息的FST模型的FDR和FIR计算结果都比基本FST模型的小一些，这是由于融合概率信息的FST模型考虑了测试存在的不确定性，测试对故障的检测概率可能小于1，其结果更符合实际测试维修情况，因此本章提出的融合概率信息的FST模型是可信的。

### 3.5 本章小结

本章深入研究了基于 FST 模型的测试性建模与分析方法，分析了基本 FST 模型在处理测试不确定信息方面存在的不足，对其进行了改进，在测试中添加不确定信息，用概率信息相关性矩阵描述测试对于故障检测的不确定性，提出了融合概率信息的 FST 模型。以某天馈组件为例分别建立其 FST 模型和融合概率信息的 FST 模型，并对两种方法计算得到的测试性指标进行了对比，结果表明：融合概率信息的 FST 模型不仅正确可行，而且更符合测试维修工程实际。

## 第 4 章 系统级层次化测试性建模与分析软件设计

作为一种测试性辅助设计的工具，系统级层次化测试性建模与分析软件能够有效降低测试性设计过程中研发时间和人力、物力、财力等资源的消耗，为研发人员进行测试性方案设计、论证和改善提供理论和技术上的指导，从而为测试维修工程的有序开展奠定基础。本章首先对软件的总体框架及其各组成部分之间的关系进行了说明，然后详细阐述各功能模块的设计实现方法。

### 4.1 软件总体架构

系统级层次化测试性建模与分析软件主要包括测试性建模平台、测试分析工具和分析结果显示窗口三个组成部分。软件总体框架及各其功能模块之间的关系如图 4-1 所示。

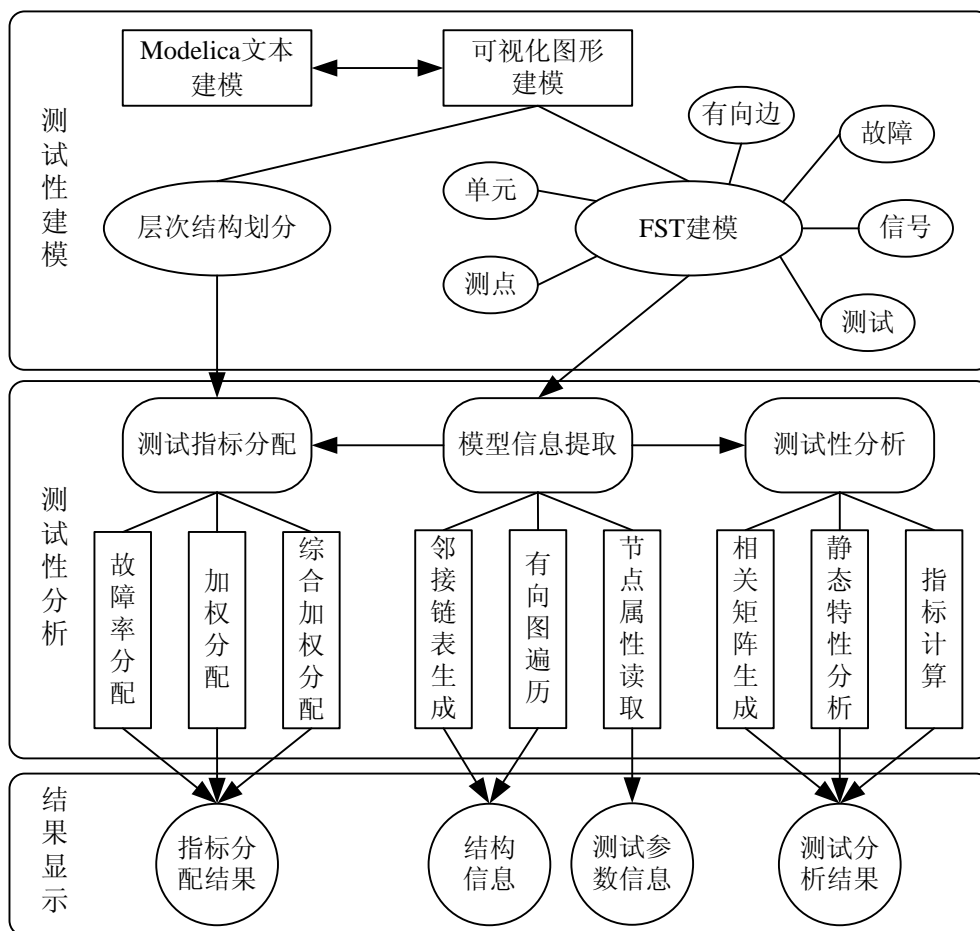


图 4-1 系统层次化测试性建模与分析软件的总体架构

测试性建模平台通过 **Modelica** 语言来描述各单元模块的结构、功能和连接关系,以实现系统建模。该平台同时支持 **Modelica** 文本建模和可视化图形建模,二者可以通过 **Modelica** 编译器相互转化。可视化模型提供 **FST** 模型的基本要素,能够用有向边来表示各单元模块及测试节点之间的信号传递方向,在单元模块和测试节点的属性设置中可以添加必要的故障信息、信号信息和测点测试信息和测试性指标影响因素量化参数等;可视化模型还能够以层次树的形式清晰表示复杂系统的层次结构,从而构成完整的系统层次化测试性模型。

测试性分析平台主要由模型信息提取、测试性指标分配和测试性分析三个模块组成。模型信息提取模块根据 **FST** 模型的结构生成邻接链表,依据邻接链表生成各个节点的可达节点集合,然后读取各节点的名称及测试性参数等属性,得到 **FST** 模型的结构信息和测试性参数信息,并将这些信息提供给测试性分配模块和测试性分析模块;测试性指标分配模块根据模型层次结构和各层级模型信息调用分配算法为各层级单元模块分配测试性指标;测试性分析模块根据模型信息生成相关矩阵,然后进行静态特性分析和测试性指标计算。

结果显示平台负责输出测试性分析平台得到的模型信息、测试性分配结果和定性与定量测试性分析结果等相关数据,并在文本文件中保存下来。

## 4.2 测试性建模平台的构建

系统测试性建模与分析软件是在开源建模仿真软件 **OpenModelica** 的基础上添加测试性建模与分析的功能完成的。**OpenModelica** 支持可视化图形建模和 **Modelica** 文本建模,提供机械控制、液压传动和电气电子等多个领域的元件模型库,还支持自定义元件模型库,各元件模型可以根据需要设置相关属性参数,并且提供了丰富的 **API** 函数,用户可以调用 **API** 函数通过 **OMC** 代理器获取模型中各元件的属性和元件之间的连接关系。因此可以利用 **OpenModelica** 的建模功能,设计测试性建模界面,自定义测试性模型库,并为库中的元件或者单元模块添加故障模式、影响信号和测试性指标影响因素等测试性信息,为测试节点定义测试、信号和测试的不确定信息等,从而实现在 **OpenModelica** 软件中建立 **FST** 模型的功能;建立一个层次树结构,以树的层次表现系统的层次,树中的每个节点关联一个 **FST** 模型,用户可以添加或者删除树节点来实现模型层次的修改,实现层次化建模。

测试性建模平台主要包括 **FST** 建模界面和层次树结构部件。**FST** 建模界面包括三个部分:**Modelica** 文本视图窗口,支持 **Modelica** 文本模型编辑,实现 **Modelica** 文本建模;图标视图窗口,以具有输入/输出端口的框图将模型抽象化,可以添加到元件模型库中作为元件直接使用;结构视图窗口,用户可以从元件

模型库中拖拽元件或单元模块，用有向边连接元件之间的输入输出端口，实现可视化图形建模。Modelica 文本视图中用 Modelica 语言描述的模型可以通过 OpenModelica 编译器转化为对应的可视化图形，模型及其各元件的外观由 Modelica 注释文本经过编译器处理得到。层次树的结构视图及其节点的添加与删除也是通过 OpenModelica 编译器实现的。测试性建模编辑界面(OMEdit)与 OpenModelica 编译器的信息交互是通过 OMC 代理器完成的，OMC 代理器通过与 CORBA 接口通信将 OMEdit 的命令发送给 OpenModelica 编译器，OpenModelica 编译器再将编译处理结果经过 CORBA 接口返回给 OMC 代理器，完成相应的动作。测试性建模平台功能示意图如图 4-2 所示。

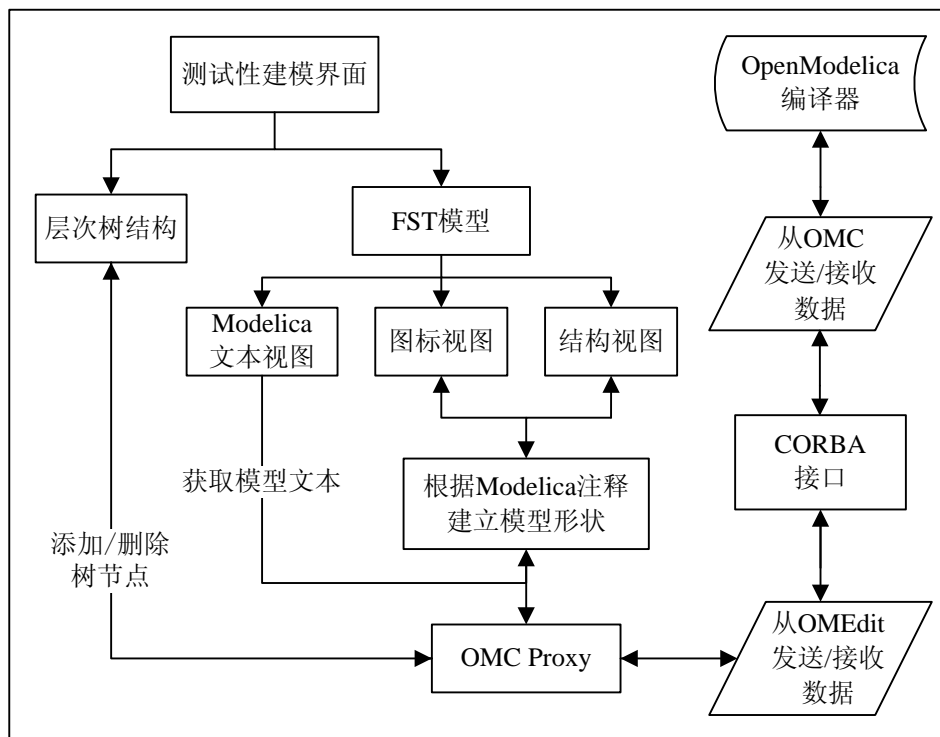


图 4-2 测试性建模平台功能示意

### 4.3 测试性分析平台程序设计

测试性分析平台主要包含模型信息提取模块、测试性指标分配模块和测试性分析模块，后两者的实现都依赖于模型信息提取。

#### 4.3.1 模型信息提取模块设计

##### (1) 邻接链表生成

FST模型是一个有向无环图，程序是不能直接对FST模型进行分析的，需要先将其转化为数据链表的形式存储下来。



有向图的数学模型可以表示为  $G=(V,E)$ ，其中  $V$  是有向图的有限节点（单元模块与测试节点）集， $E$  是有限的有向边集合。在有向图中，若有向边由节点  $V_1$  指向节点  $V_2$ ，则该有向边可以表示为  $\langle V_1, V_2 \rangle$ ， $V_1$  称为该有向边的起点， $V_2$  称为该有向边的终点，并称  $V_1$  和  $V_2$  相邻接， $V_2$  是  $V_1$  的邻接节点（ $V_1$  邻接到  $V_2$ ， $V_2$  邻接自  $V_1$ ）。

常用的有向图存储数据链表有邻接矩阵和邻接链表两种形式，本文采用后一种形式。邻接链表是一种链式结构的数据存储方式，它将有向图中各节点的所有邻接节点分别存储到一个单链表中，一个节点对应一个单链表，一个有向图的单链表总数等于其节点总数（即单元模块与测试节点的总数），一个单链表中的节点总数等于其对应的有向图节点的邻接节点总数。邻接链表通过记录有向图中各个节点的连接关系和各条有向边的传递方向，从而将有向图结构完整地保存下来，为测试性分析平台的其他模块奠定基础。

以图4-3所示的有向图为例，来说明邻接链表的存储方式。图4-3中有向图包括5个节点，7条有向边，用数字0-4对每个节点编号，如节点A（0号）有B（1号）和E（4号）两个邻接节点，所以在A对应的连接链表中填入1和4，同理可得出其它节点的连接链表。

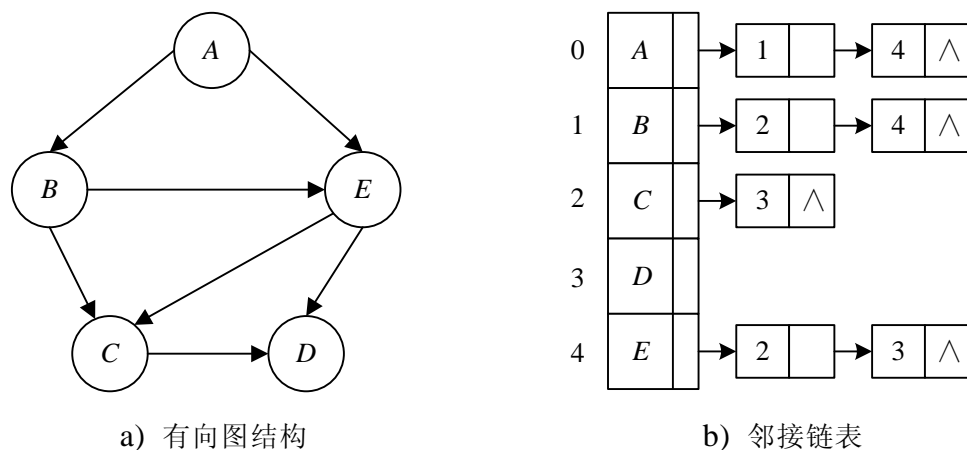


图 4-3 某有向图结构及其邻接链表

## (2) 有向图遍历

各单元模块（故障节点）的可达测试节点集是生成故障与测试相关矩阵的基础，任意节点的可达节点集合通过对有向图遍历的方法来获取，有向图的遍历即从任意节点开始，沿有向边对所有节点完成一次访问的过程。常用的图遍历算法包括两种：广度优先搜索(Breadth First Search, BFS)算法和深度优先搜索(Depth First Search, DFS)算法。由于广度优先搜索算法所需存储量较大，计算时间长，效率较低，故本文采用深度优先搜索算法。

深度优先搜索算法以邻接链表为基础，其程序流程如图 4-4 所示，从任意节点  $V$  开始，访问  $V$  的邻接链表中的第一个节点  $W_1$ ，再从  $W_1$  出发，访问没有被访问过的  $W_1$  的任意邻接节点  $W_2$ ，如此，再从  $W_2$  出发一直访问到邻接链表为空的节点  $W_n$  为止；然后退回到  $W_n$  的上一个访问节点  $W_{n-1}$ ，若  $W_{n-1}$  还有未被访问过的邻接节点，就接着访问下去，否则在退回到  $W_{n-1}$  的上一个访问节点  $W_{n-2}$ ，在节点  $W_{n-1}$  之后访问到的节点组成的集合即为  $W_{n-1}$  的可达测试节点集，如此重复直到所有节点都被访问过一次，这样就能够得到所有节点的可达测试节点集。为了避免重复访问，定义一个一维数组  $Visited$ ，其元素个数等于有向图的节点总数，起初， $Visited$  中所有元素值均为 0，表示节点都未被访问过，若节点被访问过，则将数组  $Visited$  中对应的元素值置 1。

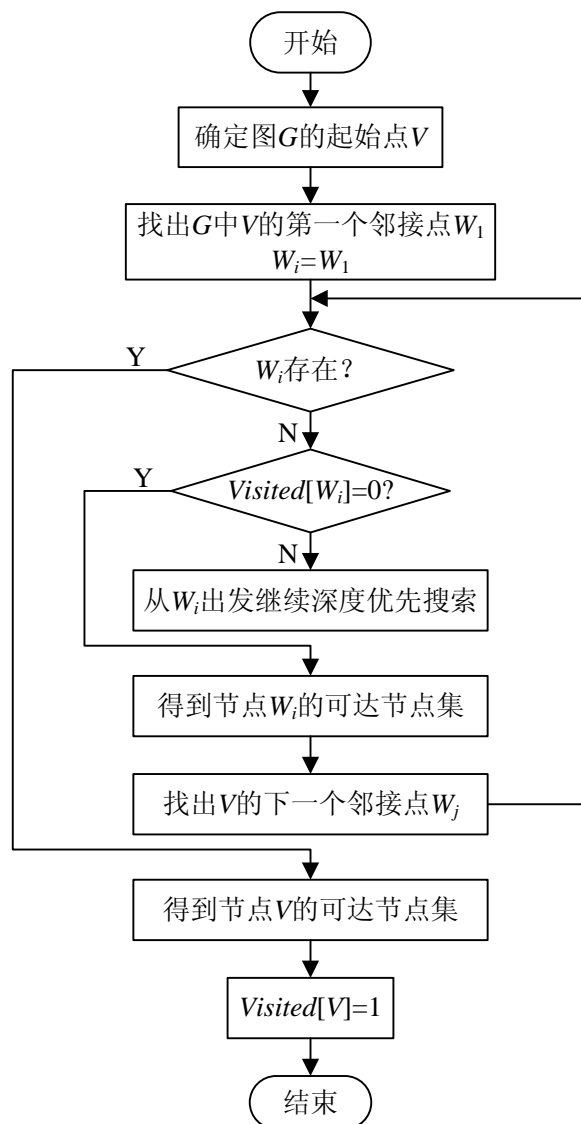


图 4-4 深度优先搜索算法程序流程图

### (3) 节点属性读取

测试性指标分配和测试性分析都是以各个节点的测试性参数为基础的，节点参数主要包括各个单元模块的测试性指标影响因子的量化值，各个单元模块关联的信号，各个测试节点中的测试，各个测试关联的信号及其概率信息等，这些参数都作为固有属性存储在单元模块或测试节点的属性列表中，只需要调用 OpenModelica 提供的 API 函数即可获取。

节点包括单元模块和测试节点，读取时需要判断，分别存储到各自的列表。节点属性读取程序的流程如图 4-5 所示，

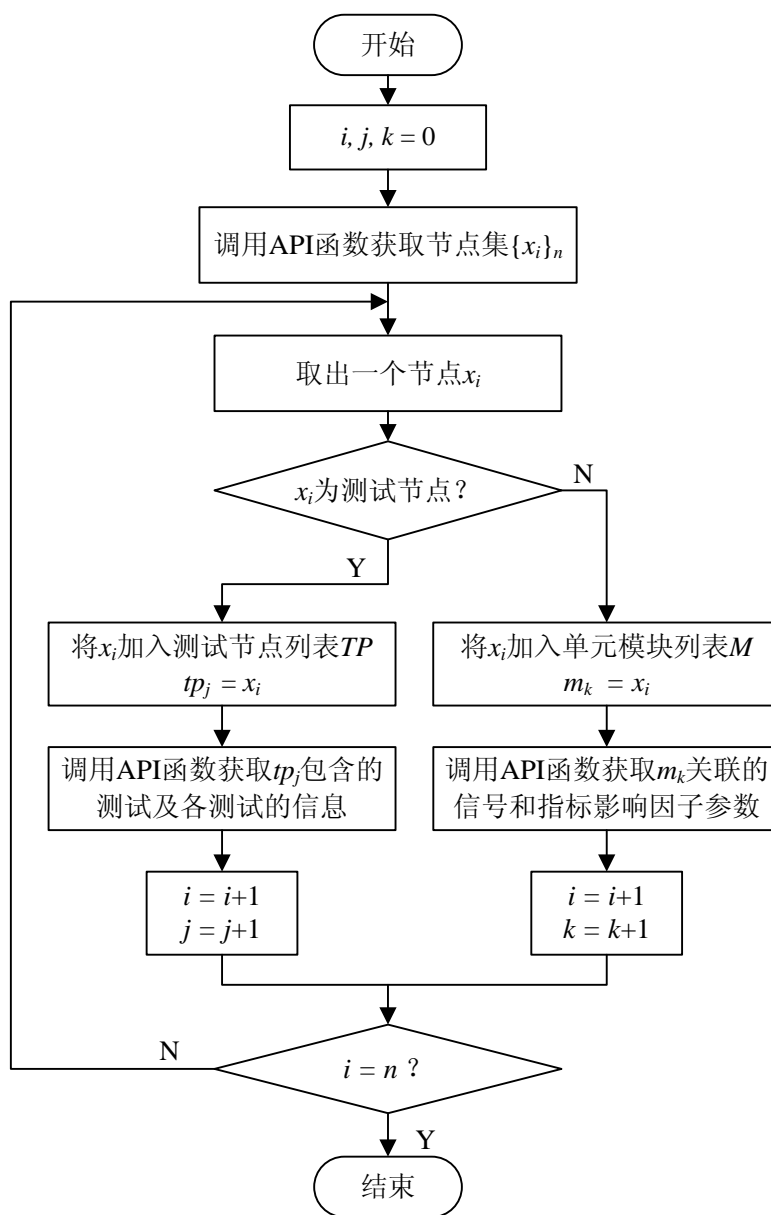


图 4-5 节点属性读取程序流程图

### 4.3.2 测试性指标分配模块设计

测试性指标分配模块的程序流程如如图 4-6 所示。

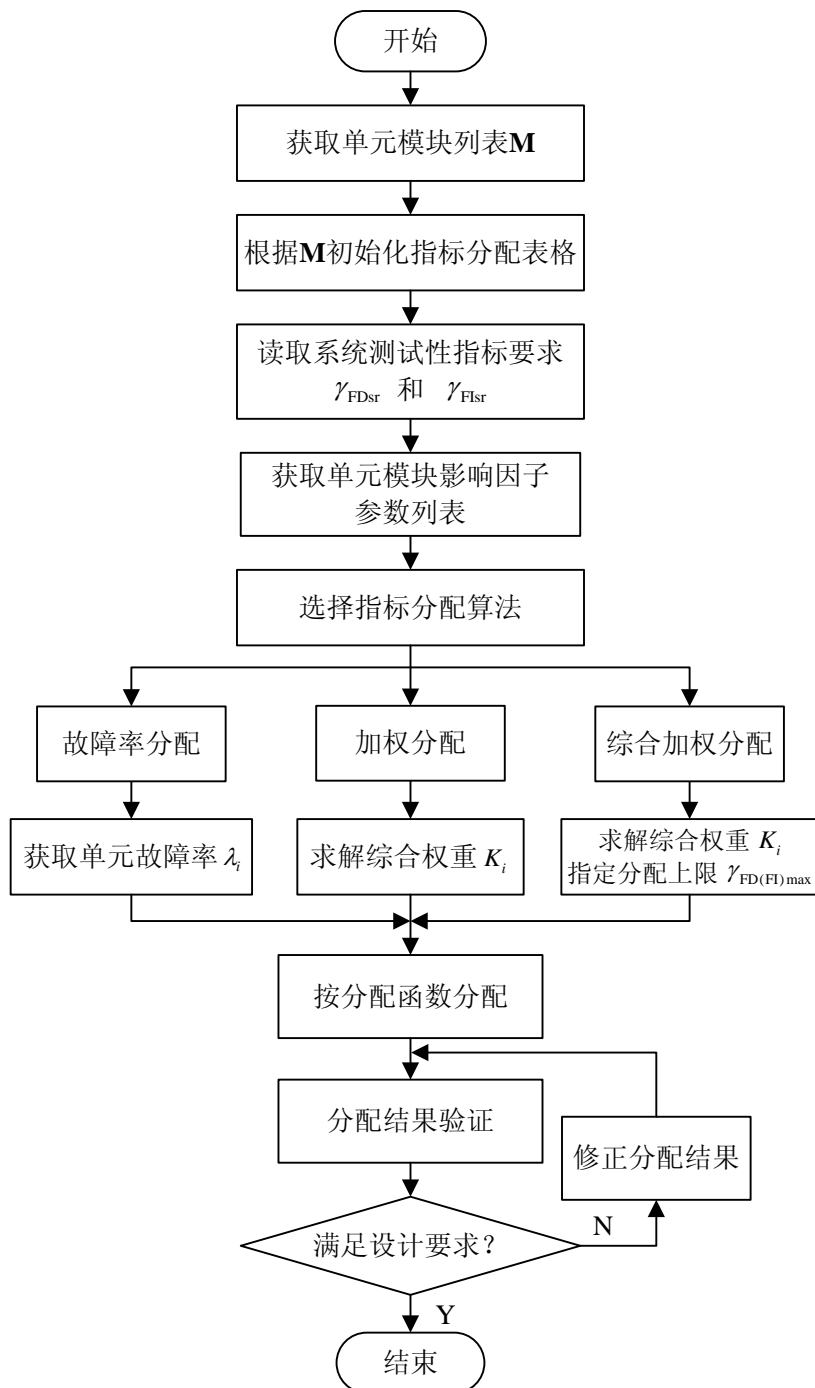


图 4-6 测试性指标分配模块程序流程图

该模块通过调用模型信息提取模块得到的单元模块列表和单元模块影响因子参数列表，根据系统设计要求，利用第 2 章阐述的三种分配算法进行测试性

分配。若存在不合理的分配额，可在单元分配额列表中适当调整，若验证指标不满足要求，再次修正，直至各单元分配额合理且验证指标能够满足要求为止。

### 4.3.3 测试性分析模块设计

#### (1) 相关矩阵生成算法程序设计

相关矩阵生成算法程序流程如图 4-7 所示。

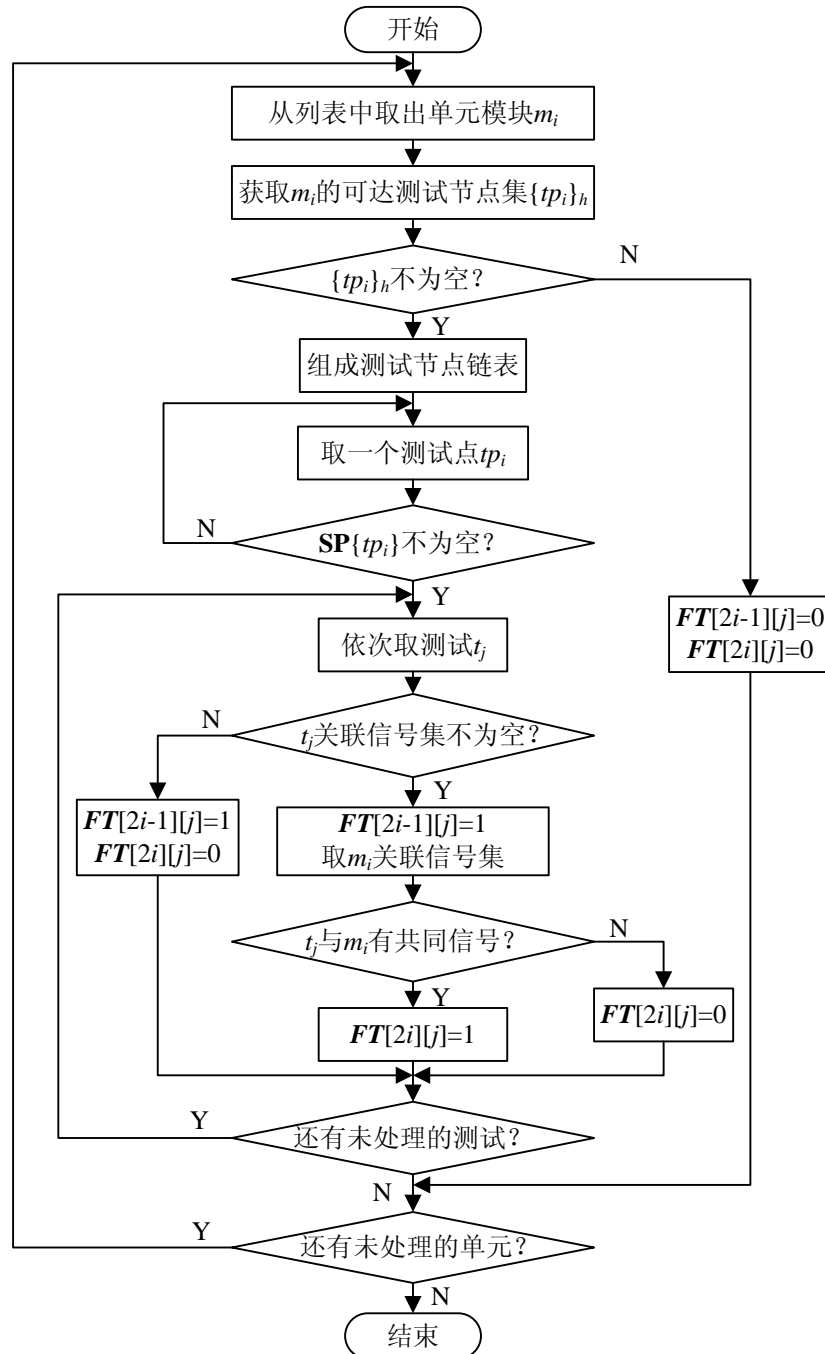


图 4-7 相关性矩阵生成算法程序流程图

相关矩阵算法直接利用了模型信息提取模块得到的 FST 模型中各单元模块（故障）的可达测试节点集。单元模块  $m_i$  的完全故障  $m_i(G)$  和功能故障  $m_i(F)$  分别对应相关矩阵  $FT$  的第  $2i-1$  行和第  $2i$  行，依次从  $m_i$  各可达测试节点的测试集中取出一个测试  $t_j$ ， $m_i(G)$  与  $t_j$  相关，相关矩阵中对应的元素  $FT[2i-1][j]=1$ ；判断  $t_j$  与  $m_i$  是否存在相同的关联信号，若存在，则  $FT[2i][j]=1$ ，否则  $FT[2i][j]=0$ 。如此就能得到完整的相关矩阵  $FT$ 。

## (2) 未检测故障分析算法程序设计

未检测故障集的获取依赖于相关矩阵  $FT$ 。为了标记故障是否被检测，定义一个一维二值数组  $undetected$ ， $undetected$  的元素个数与  $FT$  的行数相同，并且元素  $undetected[i]$  与行向量  $FT[i]$  对应相同的故障。读取相关矩阵，对各行向量进行判断，若  $FT[i]$  的元素全为零，说明对应的故障不能被检测出，则  $undetected[i]=1$ ；若  $FT[i]$  的元素不全为零，说明对应的故障能够被检测出，则  $undetected[i]=0$ 。 $undetected$  的所有元素值都能被确定之后，其中为零的元素对应的故障组成的集合即为未检测故障集。未检测故障程序流程如图 4-8 所示。

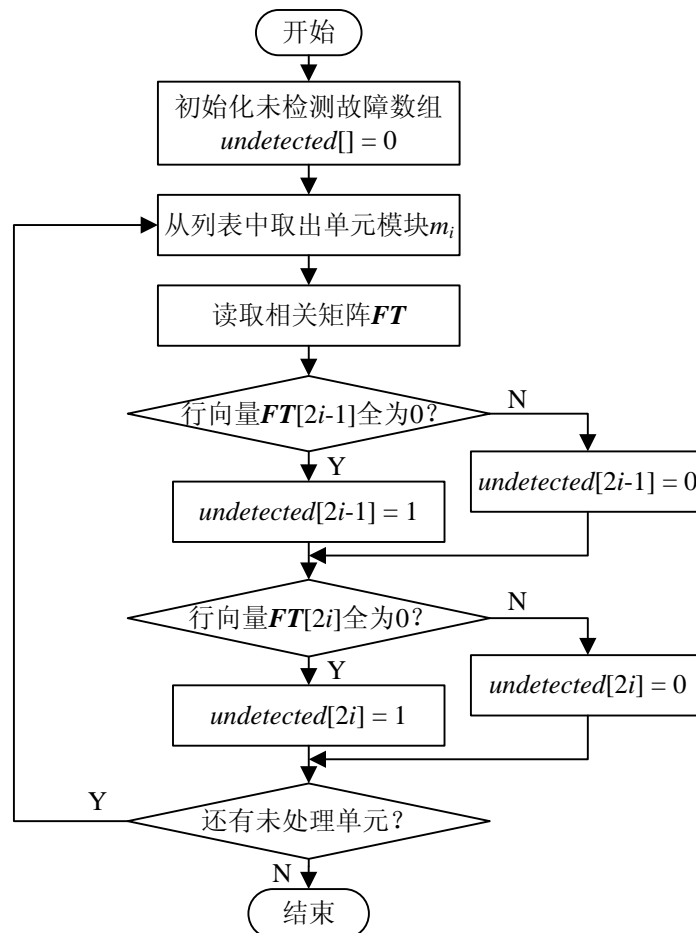


图 4-8 未检测故障算法程序流程图

### (3) 模糊组分析算法程序设计

模糊组分析算法的程序流程如图 4-9 所示。

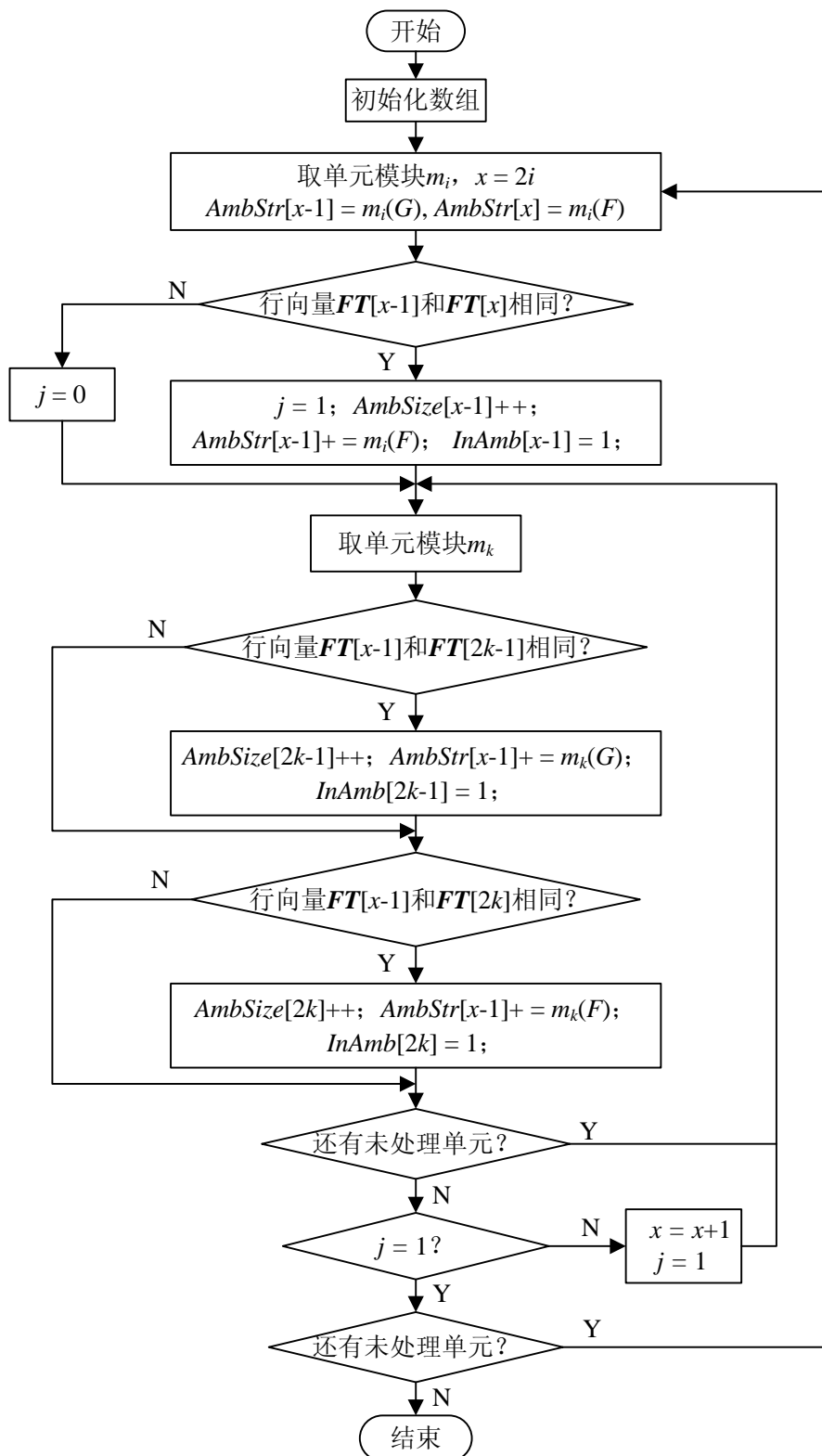


图 4-9 模糊组分析算法程序流程图

模糊组分析算法与未检测故障分析类似，都是研究相关矩阵  $FT$  的行向量。定义  $AmbStr$ 、 $InAmb$  和  $AmbSize$  三个数组，其元素个数均与  $FT$  的行数相同，且这三个数组中第  $i$  个元素与  $FT$  中第  $i$  个行向量对应的故障相同。 $AmbStr$  用来存储模糊集各模糊组包含的所有故障；二值数组  $InAmb$  用来指示故障是否是隶属于模糊集，初始化时其元素值均为 0，若其元素被置 1，则表示对应的故障包含于模糊集； $AmbSize$  用来标记各个故障的模糊度。读取相关矩阵，分别对数组  $AmbStr$ 、 $InAmb$  和  $AmbSize$  进行初始化，任取单元模块  $m_i$ ，比较行向量  $FT[2i-1]$  与  $FT[2i]$ ，若二者完全相同，把  $m_i(F)$  添加到  $m_i(G)$  的模糊组  $AmbStr[2i-1]$  中，并将  $InAmb[2i-1]$  和  $InAmb[2i]$  置 1，之后可以不再分析  $m_i(F)$ ；否则， $FT[2i-1]$  与  $FT[2i]$  分别于其余行向量作比较，将与其完全相同的行向量对于的故障添加到二者各自的模糊组。如此将所有  $InAmb$  中未被置 1 的元素对应的故障都分析一遍，这样就能得到所有模糊组，构成该模型的模糊集。

#### (4) 冗余测试分析算法程序设计

冗余测试分析算法的流程如图 4-10 所示。

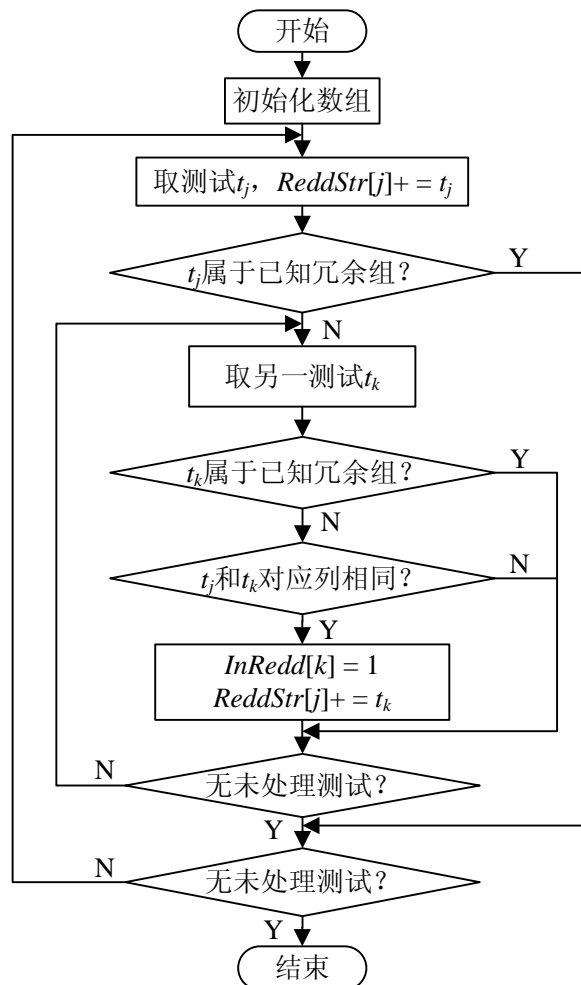


图 4-10 冗余测试分析算法程序流程图



冗余测试分析同样以相关矩阵  $FT$  为基础的,从  $FT$  的列向量提取冗余测试。定义  $ReddStr$ 、 $InRedd$  两个数组,其元素个数均与  $FT$  的列数相同,且这两个数组中第  $j$  个元素与  $FT$  中第  $j$  个列向量对应的测试相同。 $ReddStr$  用来存储各冗余测试组包含的测试; $InRedd$  用来指示测试是否是隶属于冗余测试集,初始化时其元素值均为 0,若其元素被置 1,则表示对应的测试包含于冗余测试集,为冗余测试。读取相关矩阵,取出一个测试  $t_j$ ,  $t_j$  对应列向量  $FT[j]$ ,把  $FT[j]$  与其他列向量分别作比较,判断是否存在与  $FT[j]$  完全相同的列向量,若存在,则将其对应的测试添加到数组  $ReddStr[j]$  中,并将  $InRedd$  中对应的元素置 1;再取出下一个测试  $t_{j+1}$ ,判断  $t_{j+1}$  是否为已知的冗余测试(即  $InRedd$  中  $t_{j+1}$  对应的元素是否为 1),若是,则跳过  $t_{j+1}$ ,若不是,则将  $t_{j+1}$  对应列向量  $FT[j+1]$  与其他列向量分别作比较,重复上面的步骤,这样就能提取出所有的冗余测试组,从而构成该模型的冗余测试集。

#### 4.4 本章小结

本章首先概述了系统级测试性建模与分析软件的总体架构及各功能模块之间的关系,然后对如何利用建模仿真软件 OpenModelica 构建测试性建模平台进行了简要阐述,最后详细说明了测试性分析平台中模型信息提取、测试性指标分配和测试性分析三个模块包含的各个相关算法的程序设计流程,完成了系统级测试性建模与分析软件的设计。

## 第 5 章 实验验证

本章以某野外电子侦察系统为例，根据该系统的结构及功能原理对其进行层次划分，建立各个层级单元系统的 FST 模型，利用本文设计的软件和 TEAMS 软件对其进行测试性分析，并与实际工程统计数据进行比较，从而验证本文提出的系统级层次化测试性建模与分析方法的有效性。

### 5.1 某野外电子侦察系统功能原理及其层次划分

军事情报为分析敌方动向和制定作战计划提供信息参考，是现代信息化战争中占有重要的地位。电子侦察系统作为获取军事情报的主要工具，在情报机构中的应用日益广泛，其价值与地位不可替代。

雷达和通信系统是通过向外界空间环境辐射电磁波信号来工作的，电子侦察利用电子侦察系统从复杂的电磁环境中全方位探测收集这些微弱电磁信号，并对其进行低噪放大、频率变换、筛选分类、模数转换和数字滤波等处理，提取出不同频段、不同类型的数字信号，然后对这些数字信号进行分析，根据数据库中记录的辐射源特征，推断、识别出辐射源的类型、功能和威胁程度等信息，为战备分析和军事行动提供情报支持。

为了能够及时准确获取情报，必须保证电子侦察系统的安全性、稳定性与可靠性，电子侦察系统一旦出现故障，需要能够在短时间内完成测试维修以保障情报侦察工作能够顺利持续进行，这就要求电子侦察系统具有较高的测试维修水平，其可测试性设计的好坏显得尤为重要，因此需要对电子侦察系统进行测试性建模与分析。

某野外电子侦察系统的总体结构如图 5-1 所示。

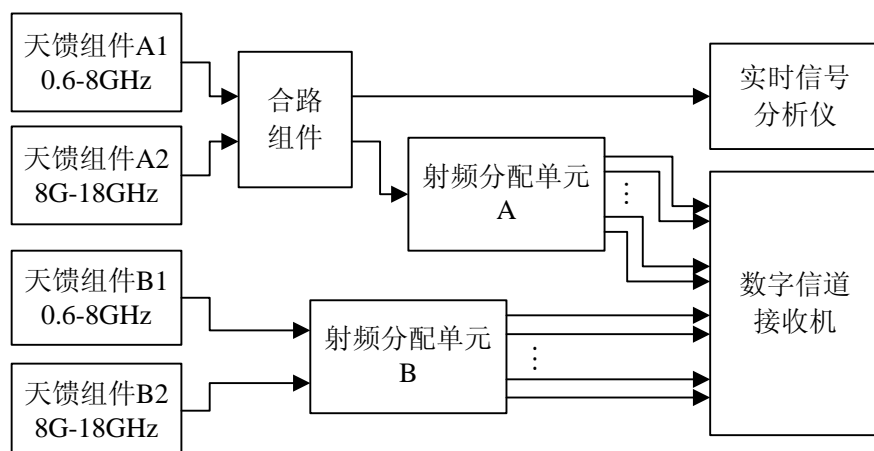


图 5-1 某野外电子侦察系统总体结构

该系统主要由 4 个天馈组件（A1、A2、B1 和 B2）、1 个合路组件、1 个实时信号分析仪、2 个射频分配单元（A 和 B）和 1 个数字信道接收机组成。

天馈组件分 0.6-8GHz 和 8G-18GHz 两个频段探测接收雷达和通信系统向外辐射的宽频模拟信号；合路组件将天馈组件接收到的各频段信号合为一路输出；实时信号分析仪对所探测到信号的功率电平、持续时间和调制特性等特征参数进行实时分析显示；射频分配单元将接收到的模拟信号进行功率放大、频率变换和频带筛选分类；数字信道接收机对宽带和窄带信号分别进行模数转换和数字滤波处理，根据转台方位信息和数据库中存储的辐射源的特征参数等信息进行分析，对辐射源进行方位、类型和功能等信息推断。

该电子侦察系统主要应用于野外环境，其修理级别一般为基层级和中继级，故障分别定位到 LRU 和 SRU，根据本文第 2 章阐述的系统层次划分原则，将该系统划分为 LRU 和 SRU 两个层级，LRU 层级包括天馈组件、合路组件、实时信号分析仪、射频分配单元和数字信道接收机，各 LRU 层级单元又由若干 SRU 组成，各层级单元组成如表 5-1 所示，各 LRU 的结构见图 5-2 至 5-5 所示。

表5-1 电子侦察系统各层级单元

系统	LRU	数量	SRU	数量
电子侦察系统	天馈组件	4	合路开关	2
			衰减器	1
			限幅器	1
			低噪声放大器	2
			功分器	2
	合路组件	1	合路开关	1
			放大器	1
	实时信号分析仪	1	—	—
	射频分配单元	2	放大功分单元	2
			变频单元	5
			合路分模块	1
			中频输出模块	3
			视频开关电路	1
	数字信道接收机	1	宽带信道化组件	2
			综合控制组件	1
			数字接收机	1
			宽带信号处理模块	1
			窄带信号处理模块	1

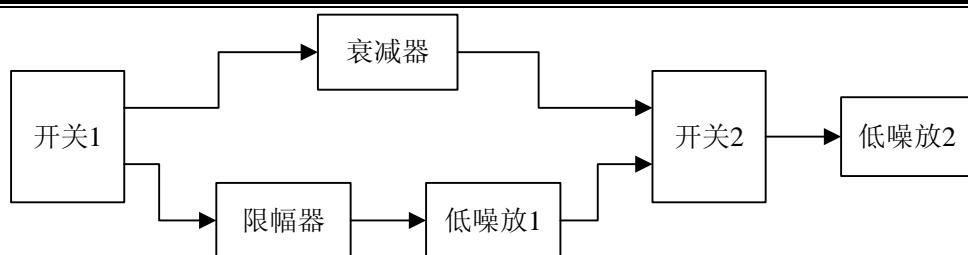


图 5-2 天馈组件的结构框图

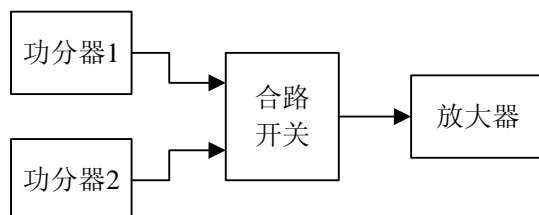


图 5-3 合路组件的结构框图

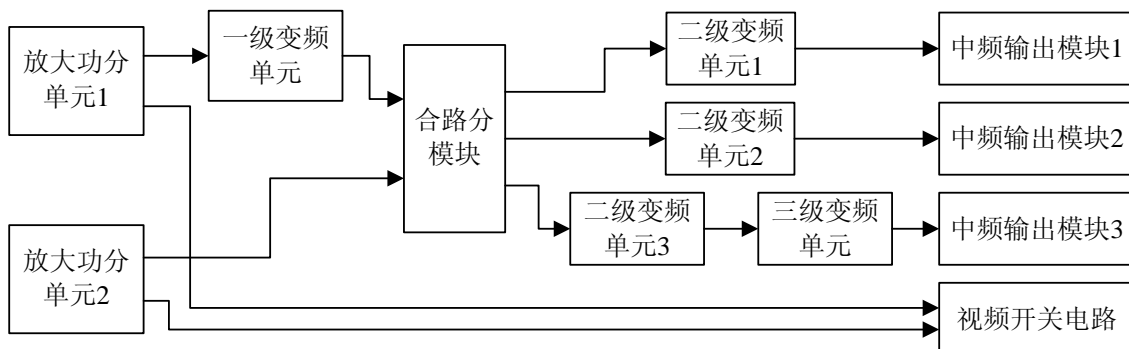


图 5-4 射频分配单元的结构框图

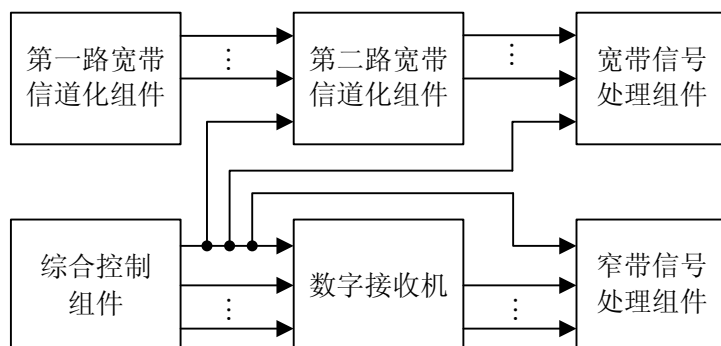


图 5-5 数字化信道接收机的结构框图

## 5.2 系统级层次化测试性建模与分析方法验证

上一节将电子侦察系统划分为 LRU 和 SRU 两个层级，因此，该系统的测试性分配目标层为 LRU 层级，测试性分析对象为各 LRU 组成的顶层系统和由若干 SRU 组成的各 LRU 系统，由于实时信号分析仪没有 SRU，故此处不对其进行测试性分析。

### 5.2.1 测试性指标分配方法验证

首先根据电子侦察系统的层次结构，利用本文设计测试性建模与分析软件建立其层次化模型，利用树形结构表示层次关系，如图 5-6 所示。电子侦察系统划分为两个层次，所以其树形结构包含两个级别的节点，顶层系统 Elec\_Rec\_Sys 位于树形结构的最上层，天馈组件 Ant\_FB、合路组件 Ant\_Combiner 和射频分配单元 RFDU 等 LRU 作为顶层系统的子节点包含于 Elec\_Rec\_Sys 中，该结构简明直观，清晰地表现了系统的层次结构。

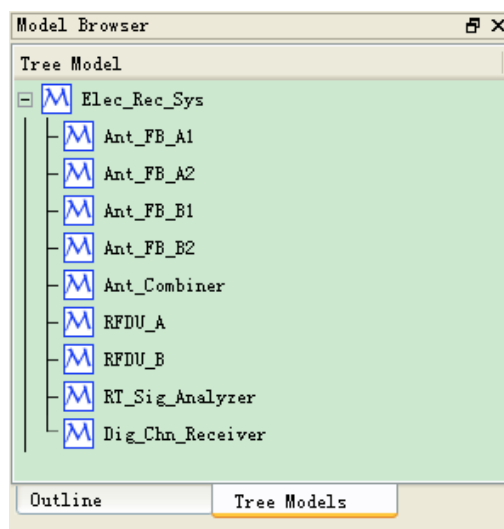


图 5-6 利用树形结构表示电子侦察系统的层次结构

接着，根据树形结构将系统将顶层系统指标逐级分配给各 LRU，顶层系统的故障检测率和隔离率的设计要求量值分别为 96.0%和 88.0%，最大可能实现值分别为 98.0%和 94.5%。系统各单元影响因子的相关数据见表 5-2，各影响因子的相对重要度相同。

表5-2 待分配系统各单元影响因子数据

单元	数量	$\lambda_i/10^{-6}$	$F_i$	$T_i$	$M_i$	$C_i$
LRU1（天馈组件）	4	110	2	3	120	200
LRU2（合路组件）	1	80	2	6	20	30
LRU3（实时信号分析仪）	1	20	2	2	90	150
LRU4（射频分配单元 A）	1	240	7	8	45	75
LRU5（射频分配单元 B）	1	150	5	8	20	50
LRU6（数字信道接收机）	1	180	9	9	80	250
系统	9	1110	—	—	—	—

利用本文设计的软件按照表 5-2 将影响因子数据录入各单元的属性列表，例如射频分配单元 A 的属性列表如图 5-7 所示。之后调用相关功能模块对该系

统进行测试性指标分配，可以根据需要在故障率法、加权法和综合加权法三种分配算法中任选其一完成测试性分配，三种分配算法的分配结果如表 5-3 所示。由于各单元的故障率和综合权重差距较大，因此故障率分配算法和加权分配算法的分配结果会出现不合理的情况，表 5-3 中的结果均已修正，综合加权算法无需修正。经过验证，三种分配算法都能得到满足系统设计要求的分配结果。

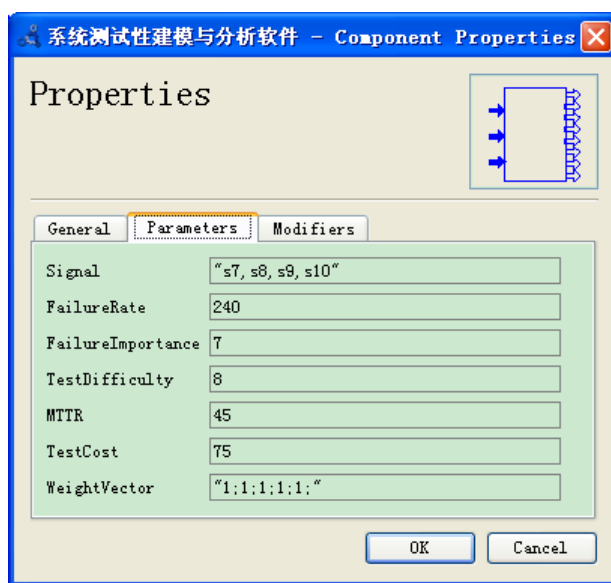


图 5-7 射频分配单元 A 的属性列表

表5-3 电子侦察系统测试性分配结果

单元	数量	$\lambda_i/10^{-6}$	故障率分配			加权分配		综合加权分配	
			$\gamma_{FDi}(\%)$	$\gamma_{Fli}(\%)$	$K_i$	$\gamma_{FDi}(\%)$	$\gamma_{Fli}(\%)$	$\gamma_{FDi}(\%)$	$\gamma_{Fli}(\%)$
LRU1	4	110	95.50	83.00	0.6293	94.00	83.00	94.50	83.02
LRU2	1	80	92.00	80.00	0.6281	98.00	83.00	98.00	83.63
LRU3	1	20	80.00	75.00	1.3554	94.00	94.50	94.69	94.50
LRU4	1	240	98.00	94.50	0.8566	98.00	93.00	97.23	91.98
LRU5	1	150	97.00	91.00	0.8341	98.00	93.00	97.21	91.89
LRU6	1	180	97.50	93.50	0.8566	97.00	88.00	96.28	88.86
系统	9	1110	96.04	88.03	—	96.18	88.21	96.00	88.00

### 5.2.2 测试性分析方法验证

由于各 LRU 的建模与分析方法及流程完全相同，故本小节在验证测试性分析方法时，仅对顶层系统和射频分配单元 A 的建模与分析作详细说明，其余各 LRU 仅给出分析结果。

按照加权分配结果系统进行可测试性设计，合理设置测试节点，然后根据系统各单元的功能结构特性，定义系统的信号集与测试集，并将信号和测试分

别分配到相应的单元模块和测试节点中，顶层系统和射频分配单元 A 中测试节点、测试、信号与单元模块的关联信息分别如表 5-4 和表 5-5 所示。

表 5-4 顶层系统测试节点、测试、信号与单元模块的关联信息

测试节点	测试	测试关联的信号	信号关联的单元模块
tp1	t1	s1	天馈组件 A1
tp2	t2	s2	天馈组件 A2
tp3	t3	s3	天馈组件 B1
tp4	t4	s4	天馈组件 A2
tp5	t5	s5	合路组件
tp6	t6	s6	实时信号分析
tp7	t7	s7	射频分配单元 A
tp8	t8	s8	射频分配单元 A
tp9	t9	s9	射频分配单元 A
tp10	t10	s10	射频分配单元 A
tp11	t11	s11	射频分配单元 B
tp12	t12	s12	射频分配单元 B
tp13	t13	s13	射频分配单元 B
tp13	t14	s14	数字信道接收机

表 5-5 射频分配单元 A 测试节点、测试、信号与单元模块的关联信息

测试节点	测试	测试关联的信号	信号关联的单元模块
tp_1	t_1	s_1	放大功分单元 1
tp_2	t_2	s_2	放大功分单元 2
tp_3	t3_	s_3	一级变频单元
tp_4	t_4	s_4	视频开关电路
tp_5	t_5	s_5	合路分模块
tp_6	t_6	s_6	一级变频单元 1
tp_7	t_7	s_7	中频输出模块 1
tp_8	t_8	s_8	一级变频单元 2
tp_9	t_9	s_9	中频输出模块 2
tp_10	t_10	s_10	二级变频单元 3
tp_11	t_11	s_11	三级变频单元
tp_12	t_12	s_12	中频输出模块 3

利用本文设计的软件建立的系统有向图结构，依据表 5-4 和表 5-5 在各单元模块中添加与之相关联的信号，在各测试节点中添加与之相关联的测试、测试关联的信号以及测试的概率信息，从而构成完整的 FST 模型，顶层系统和射频分配单元 A 的 FST 模型分别如图 5-8 和图 5-9 所示。

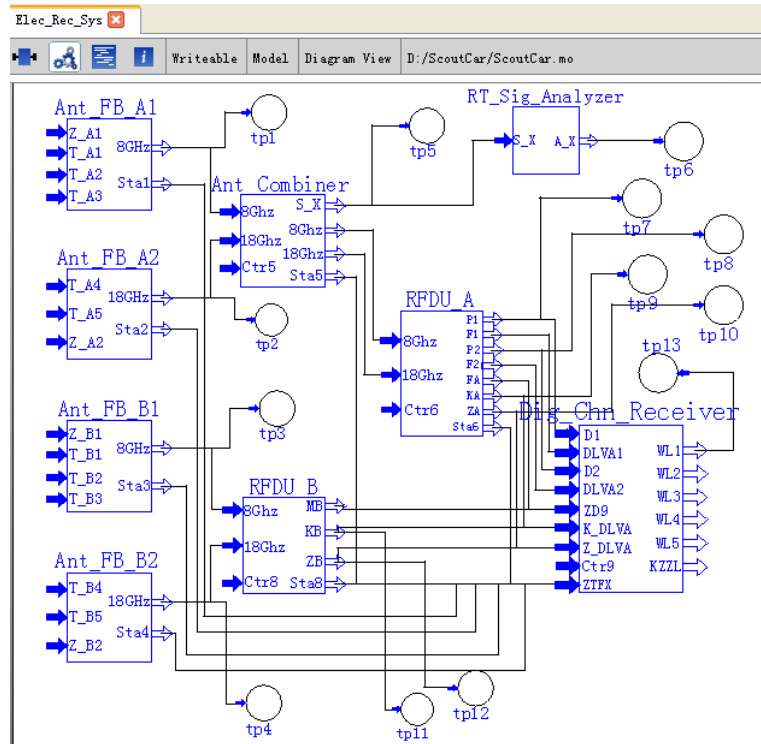


图 5-8 顶层系统的 FST 模型

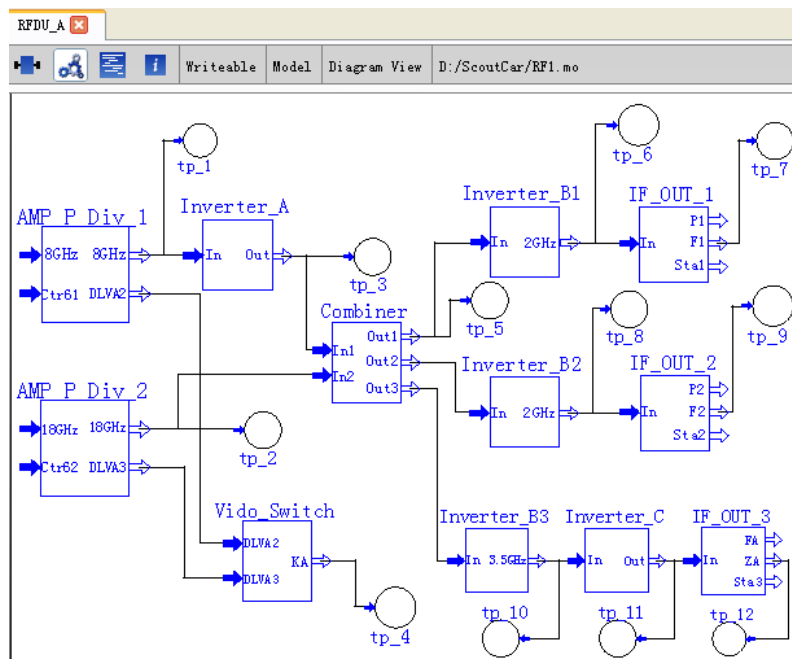
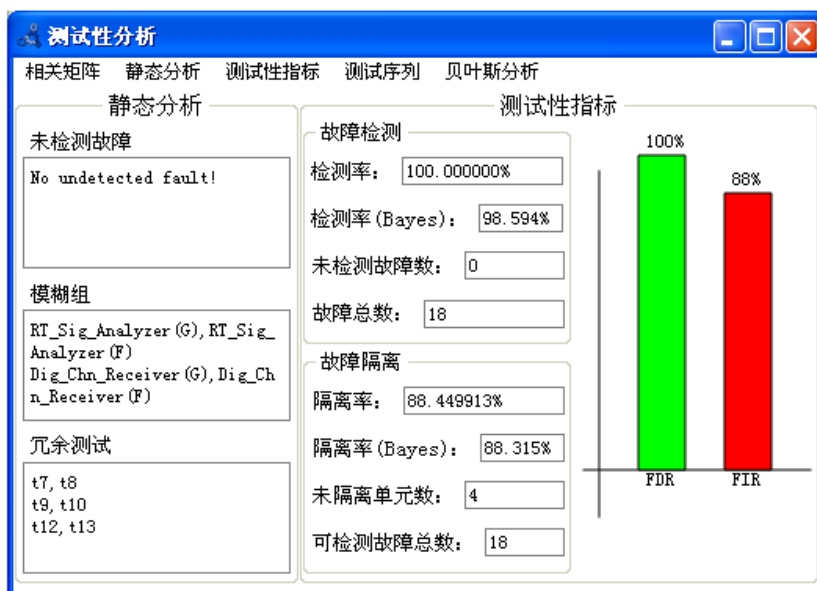


图 5-9 射频分配单元的 FST 模型

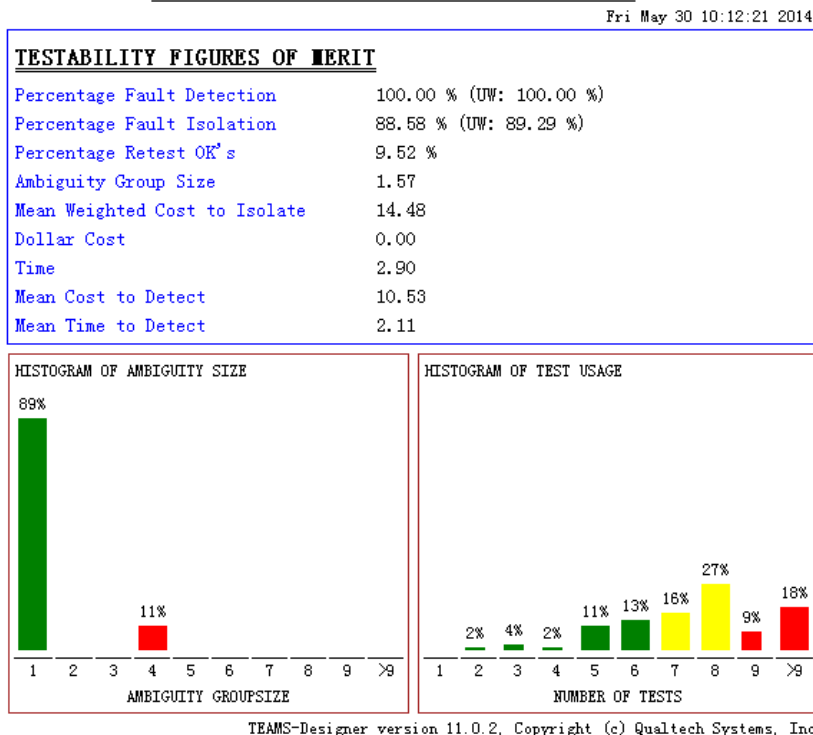


然后分别利用本文所设计软件和 TEAMS 软件的测试性分析模块，对系统的 FST 模型进行测试性评估，顶层系统和射频分配单元 A 的测试性结果如图 5-10 和图 5-11 所示。



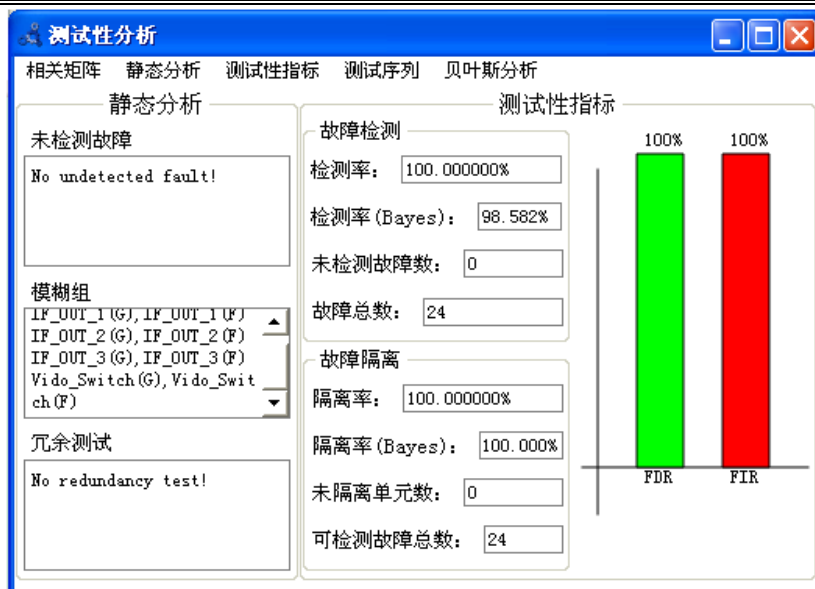
a) 本文设计的软件对顶层系统的 FST 模型分析结果

### Testability Report for 电子侦察系统



b) TEAMS 软件对顶层系统的 FST 模型分析结果

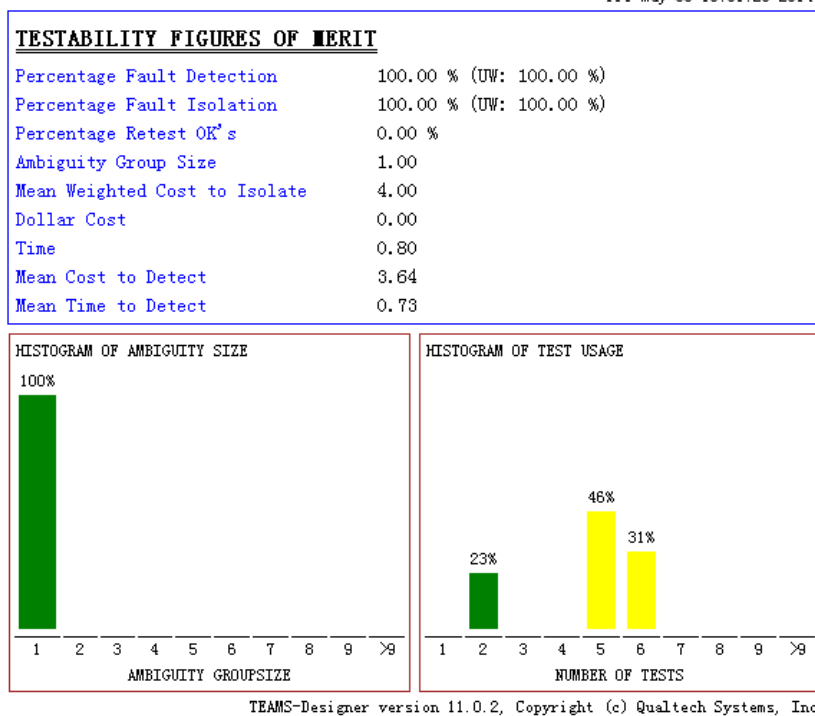
图 5-10 顶层系统测试性模型分析结果



a) 本文设计的软件对射频分配单元 A 的 FST 模型分析结果

### Testability Report for 射频分配单元A

Fri May 30 15:31:26 2014



b) TEAMS 软件对射频分配单元 A 的 FST 模型分析结果

图 5-11 射频分配单元 A 测试性模型分析结果

其余各 LRU 的分析结果如表 5-6 所示，TEAMS 软件只采用基本的 FST 模型，而本文设计的软件兼容基本的和融合概率信息的 FST 模型，表 5-6 给出的本文软件的分析结果均基于融合概率信息的 FST 模型。为进行更为全面的对比分析，在表 5-7 中还给出了某使用方提供的该系统的测试工程统计数据。

表5-6 电子侦察系统测试性模型分析结果

单元	数量	TEAMS 软件		本文设计的软件	
		FDR(%)	FIR(%)	FDR(%)	FIR(%)
顶层系统	1	100.00	88.58	98.59	88.32
天馈组件	4	100.00	83.33	99.42	83.12
合路组件	1	100.00	100.00	99.57	100.00
射频分配单元 A	1	100.00	100.00	98.58	100.00
射频分配单元 B	1	100.00	100.00	98.65	100.00
数字信道接收机	1	100.00	88.83	98.68	88.65

表5-7 电子侦察系统实际测试工程统计数据

单元	数量	FDR(%)	FIR(%)
顶层系统	1	98.00	88.25
天馈组件	4	99.00	83.10
合路组件	1	99.50	99.90
射频分配单元 A	1	98.50	99.00
射频分配单元 B	1	98.50	99.50
数字信道接收机	1	98.00	88.50

从表 5-6 可以看出，两款软件计算得到的故障检测率和隔离率指标都非常相近。对于顶层系统，其设计要求的故障检测率和故障隔离率指标分别为 96.0% 和 88.0%；TEAMS 得到结果的为 100% 和 88.58%，本文设计的软件得到的结果是 98.59% 和 88.32%，二者误差在 0.5% 以内，而且分析结果都比设计要求值大，都表明顶层测试性设计方案能够满足指标要求；对于射频分配单元 A，两款软件的分析结果均比加权分配额要大，二者误差也小于 1.5%，都能够说明该 LRU 测试性指标能够满足设计要求；其余各 LRU 也有类似的结论。

对比表 5-6 与表 5-7 中的数据，两款软件对电子侦察系统的测试性模型分析结果与该系统实际测试工程统计数据一致，误差均不超过 2.0%；由于采用了融合概率信息的 FST 模型，在分析过程中考虑了测试的不确定性，本文设计软件的分析结果比 TEAMS 更接近实际统计值。因此，本文提出的测试性建模与分析方法是有效可靠的。

### 5.3 本章小结

本章以某野外电子侦察系统为研究对象，利用文本所设计的测试性建模与分析软件对其进行了测试性分析，并将结果与商业软件 TEAMS 得出的结果进行了分析比较，二者对于同一系统的测试性分析结果相同，都与实际测试工程统计数据一致，验证了本文提出的测试性建模与分析方法的准确性和可靠性。

## 结 论

测试性设计作为提高系统测试性、可靠性和稳定性的重要手段之一，在军事国防工程和工业生产中日益受到重视。系统级层次化测试性建模与分析关乎测试性设计的好坏，是测试性设计的核心内容。本文主要研究了系统层次划分方法、测试性指标分配、测试性建模及其分析方法的原理和流程，构建一个完整的系统级测试性模型分析与评价体系，并设计相应的测试性建模与分析软件，为提高测试性设计水平提供辅助工具。本文的主要研究成果如下：

(1) 在总结归纳以往经验的基础上提出了复杂系统的层次划分方法，该方法符合系统的功能结构并且满足实际系统的测试、诊断与维修活动需求。制定了故障率法、加权法和综合加权法三种测试性分配算法的实施方案，并对三种算法的分配过程及结果进行了对比，故障率法过程相对简单，但其分配结果可能存在不合理的情况；加权法和综合加权法虽然过程繁琐，但其考虑的因素较为全面，结果也更为合理。

(2) 对基本 FST 模型进行了改进，提出了融合概率信息的 FST 模型，该模型能够描述测试与故障的不确定关系，并在测试性分析过程中利用概率信息相关矩阵处理这种不确定信息，通过实例分析表明：融合概率信息的 FST 模型与基本 FST 模型均能对系统测试性进行描述与评价，但前者更贴近工程实际，其测试性分析结果也更为可信。

(3) 设计系统级层次化建模与分析软件，该软件由测试性建模平台、测试分析工具和分析结果显示窗口三部分组成，能够建立复杂系统的层次化树形模型并根据层次结构实现测试性指标分配，能够对各层模型分别进行测试性分析并求解静态特性和测试性指标，验证测试性设计是否满足要求。以某电子侦察系统为研究对象，将本文设计的软件与商业软件 TEAMS 进行了对比，二者的分析结果均与实际测试工程统计数据一致，验证了本文提出的系统级层次化测试性建模与分析方法的准确性与可靠性。

需要完善的地方如下：

(1) 在利用综合加权分配算法进行测试性分配时，仅约束了分配额的上限，综合权重较小的单元模块其分配额可能较低，有待改进。

(2) 在融合概率信息的 FST 模型中不确定参数都是根据实际经验得到的估计值，可能存在欠妥之处，需要研究相关理论算法以获取精确的不确定参数值。

## 参考文献

- [1] Ungar L Y. Testability design prevents harm[J]. Aerospace and Electronic Systems Magazine, IEEE, 2010, 25(3): 35-43.
- [2] Saeed S M, Sinanoglu O. Design for Testability Support for Launch and Capture Power Reduction in Launch-Off-Shift and Launch-Off-Capture Testing[J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2014, 22(3): 516-521.
- [3] Khan S, Phillips P, Jennions I, et al. No Fault Found events in maintenance engineering Part 1: Current trends, implications and organizational practices[J]. Reliability Engineering & System Safety, 2014, 123: 183-195.
- [4] Khan S, Phillips P, Hockley C, et al. No Fault Found events in maintenance engineering Part 2: Root causes, technical developments and future research[J]. Reliability Engineering & System Safety, 2014, 123: 196-208.
- [5] Roy D. Design, system model and development of customized electronic light barriers for robotic and mechatronic applications[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(4): 153-172.
- [6] Nikfard P, Najafabadi M K, Rouhani B D, et al. An Empirical Analysis of a Testability Model[C]. Informatics and Creative Multimedia (ICICM), 2013 International Conference on. IEEE, 2013: 63-69.
- [7] Nikfard P, Najafabadi M K, Rouhani B D, et al. An Empirical Study into Model Testability[C]. Informatics and Creative Multimedia (ICICM), 2013 International Conference on. IEEE, 2013: 85-92.
- [8] Harmanani H, Papachristou C, Chiu S, et al. SYNTTEST: an environment for system-level design for test[C]. Proceedings of the conference on European design automation. IEEE Computer Society Press, 2012: 402-407.
- [9] 杨鹏. 基于相关性模型的诊断策略优化设计技术[D]. 长沙: 国防科学技术大学, 2008: 15-20.
- [10] 刘哲. 一种基于模糊评价分层模型的构件可测试性评价方法[D]. 西南大学, 2011: 16-28.
- [11] Wright D, Strigini L. Bounds on survival probability given mean probability of failure per demand; and the paradoxical advantages of uncertainty[J]. Reliability Engineering & System Safety, 2014, 128: 183-195.
- [12] Rahaman H, Kole D K, Das D K, et al. Fault diagnosis in reversible circuits under missing-gate fault model[J]. Computers & Electrical Engineering, 2011, 37(4): 475-485.

- [13] Ciani L, Catelani M. A fault tolerant architecture to avoid the effects of Single Event Upset (SEU) in avionics applications[J]. Measurement, 2014, 54: 256-263.
- [14] Belli F, Beyazit M, Mathur A P, et al. Modeling, Analysis, and Testing of System Vulnerabilities[J]. Advances in Computers, 2012, 84: 39-92.
- [15] 龙兵. 多信号建模与故障诊断方法及其在航天器中的应用研究[D]. 哈尔滨: 哈尔滨工业大学, 2005: 34-43.
- [16] 吕晓明, 黄考利, 连光耀. 基于多信号流图的分层系统测试性建模与分析[J]. 北京航空航天大学学报, 2011, 37(9): 1151-1155.
- [17] 吕政良, 王红, 勒洋, 等. 基于多信号流图的高层模块描述[J]. 清华大学学报, 2011, 51(7): 884-888.
- [18] Emmert G. Method for improving design testability through modeling[C]. AUTOTESTCON, 2010 IEEE. IEEE, 2010: 1-4.
- [19] Onishi J, Kimura S, James R J W, et al. Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method[J]. Reliability, IEEE Transactions on, 2007, 56(1): 94-101.
- [20] Dehghanian P, Fotuhi-Firuzabad M, Bagheri-Shouraki S, et al. Critical component identification in reliability centered asset management of power distribution systems via fuzzy AHP[J]. Systems Journal, IEEE, 2012, 6(4): 593-602.
- [21] Kumar R, Izui K, Masataka Y, et al. Multilevel redundancy allocation optimization using hierarchical genetic algorithm[J]. Reliability, IEEE Transactions on, 2008, 57(4): 650-661.
- [22] 王权伟, 蒋里强, 王维兴. 武器系统维修性分配的线性规划方法研究[J]. 装备指挥技术学院学报, 2006, 17(3): 33-36.
- [23] 沈亲沐. 装备系统级测试性分配技术研究及应用[D]. 长沙: 国防科技大学, 2004: 33-44.
- [24] 张琦, 朱春生, 冉红亮, 等. 基于 NSGA-II 的测试性指标分配方法[J]. 南京理工大学学报: 自然科学版, 2012, 36(4): 650-655.
- [25] 李世英, 曲长征, 薛文力. 信息化装备体系的 RMS 参数体系框架研究[J]. 装甲兵工程学院学报, 2008, 22(6): 39-42.
- [26] 刘刚, 黎放, 胡斌. 基于相关性模型的舰船装备测试性分析与建模[J]. 海军工程大学学报, 2012, 24(4): 46-51.
- [27] S. D. Huynh, M. Soma, et al. Testability Analysis and Multi-frequency ATPG for Analog Circuits and Systems[C]. IEEE/ACM International Conference on

Computer Aided Design, 1998. ICCAD, 376-383.

- [28] 刘明, 左洪福. 航空维修策略研究[J]. 飞机设计, 2007, 27(3): 42-45.
- [29] O'Donnell S J, Zarcone A. Managing evolving hardware and software requirements[C]. Proceeding of the IEEE AUTOTESTCON, Anaheim, 2009: 130-135.
- [30] Yang L T, Mauzio J. Testing methodologies for embedded systems and systems-on-chip[C]. First International Conference on Embedded Software and Systems, ICESS, 2005:15-24.
- [31] 王刚. 装备测试性参数优化选择技术研究[D]. 长沙: 国防科学技术大学, 2010: 26-35.
- [32] Huang C C, Kuo C M. The transformation and search of semi-structured knowledge in organizations[J]. Journal of Knowledge Management, 2003, 7(4): 106-123.
- [33] 张强, 胡俊, 胡政, 等. 基于多信号流模型的航天器供电系统测试性建模与分析[J]. 载人航天, 2012, 18(6): 39-46.
- [34] Liu Haiming, Yi Xiaoshan. Analysis and modeling of testability based on multi-signal flow graphs[J]. China Measurement Technology, 2007, 33(1): 49-51.
- [35] Yang Zhiyong, Xu Aiqiang, Niu Shuangcheng. Modeling and analysis of system testability based on multi-signal model[J]. Journal of Engineering Design, 2007, 14(5):364-368.
- [36] CHEN Shijie, LIAN Ke, WANG Houjun. Fault diagnosis method of radar receiver using multi-signal flow graphs model[J]. Journal of University of Electronic Science and Technology of China, 2009, 38(1): 87-91.
- [37] 许辉, 梁力. 基于多信号模型的测试性分析方法研究[J]. 计算机测量与控制, 2012, 20(4): 914-916.
- [38] 彭培, 周沫. 基于多信号模型的舰船电子装备测试性分析和评估[J]. 舰船电子工程, 2008, 28(10): 180-187.
- [39] 张勇, 邱静, 刘冠军. 测试性模型对比及展望[J]. 测试技术学报, 2011, 25(6): 504-514.
- [40] 钱彦岭. 测试性建模技术及其应用研究[D]. 长沙: 国防科技大学, 2002: 17-36.
- [41] 陈希祥. 装备测试性方案优化设计技术研究[D]. 长沙: 国防科技大学, 2010: 28-40.
- [42] 连光耀. 基于信息模型的装备测试性设计与分析方法研究[D]. 石家庄: 机械工程学院, 2007: 25-37.

- [43] 邱静, 刘冠军, 杨鹏, 等. 装备测试性建模与设计技术[M]. 北京: 科学出版社, 2012.
- [44] Gould E. Modeling it both ways: Hybrid diagnostic modeling and its application to hierarchical system designs[J]. Proceedings of the IEEE AUTOTESTCON, San Antonio, 2013: 576-582.
- [45] Pattipati K R, Raghavan V, Shakeri M, et al. TEAMS: Testability engineering and maintenance system[C]. Proceedings of the American Control Conference, Baltimore, 2009, 2: 1989-1995.
- [46] Deb S, Domagala C, Shrestha R, et al. Model-based testability assessment and directed troubleshooting of shuttle wiring systems[C]. Proceedings of the SPIE AeroSense Conference, Orlando, 2011: 163-173.
- [47] 魏振春. 离散事件控制系统的分层建模与规则化描述方法研究[D]. 合肥: 合肥工业大学, 2007: 9-18.
- [48] 周德群, 章玲. 集成 DEMATEL/ISM 的复杂系统层次划分研究[J]. 管理科学学报, 2008, 11(2): 20-26.



## 攻读硕士学位期间发表的论文及其它成果

### （一）发表的学术论文

- [1] Chunling Yang, Yingrong Zheng, Min Zhu, et al. A Testability Allocation Method Based on Analytic Hierarchy Process and Comprehensive Weighted[C]. Industrial Electronics & Applications(ICIEA), 2014 9th IEEE Conference on. IEEE, 2014: 113-116. (EI 检索源)

### （二）竞赛获奖

- [1] 张思文, 童陈敏, 郑应荣. 开关电源可测性设计及故障诊断系统. 中国研究生电子设计竞赛全国三等奖. 2012.08.

## 哈尔滨工业大学学位论文原创性声明和使用权限

### 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《系统级层次化测试性建模与分析》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：郑应荣

日期：2014 年 6 月 26 日

### 学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2) 学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3) 研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：郑应荣

日期：2014 年 6 月 26 日

导师签名：杨春金

日期：2014 年 6 月 26 日

## 致 谢

时间荏苒，两年的硕士时光一晃而过，在即将离开的日子里心中感慨万千。感谢培育了我六年的母校哈尔滨工业大学，希望母校越来越好，逐步实现世界一流大学的目标。

感谢我的导师杨春玲教授，她是一位慈爱而善良的老师，严厉而认真的老师。杨老师对待科研严谨，对待生活乐观，对待同事热情，对待学生耐心，这些优秀的品质对我产生了深深的影响，她不但指导我完成了本课题的研究工作和论文的撰写，还教会了我许多为人处事之道，我会时刻把老师的教诲铭记心中，激励自己在各方面不断进步。

感谢刘思久教授在科研生活中给我的指导和帮助，刘老师渊博的知识和对科研事业的热爱让我深深的佩服和崇敬；感谢实验室朱敏老师和张岩老师对我学习生活中的帮助和关心，每周和刘老师、朱老师的学术讨论让我认识到自己的不足，及时的改正错误，让我少走了很多弯路，在此深表感激。

感谢实验室的各位师兄姐妹，在你们的陪伴下我度过了愉快的两年，让我的生活中充满了色彩，和你们一起学习、讨论、游玩的日子是那么的有意义。在此要特别感谢陈宇、童陈敏、刘国成、王荔和左振鹏，谢谢你们在学习和生活中对我的帮助和支持。

感谢和我一起奋斗的同学以及和我一起生活了两年的室友，谢谢你们对我的关心和包容。

最后要感谢我的家人，谢谢你们对我的养育和照顾，遇到困难时是你们给我最大的安慰和力量，让我充满希望，坚定地前行，谢谢你们这么多年的付出和奉献，今后的我会更加努力，报答你们，报答社会，祝你们永远健康、快乐。