

国内图书分类号: TP206

学校代码: 10213

国际图书分类号: 621.3

密级: 公开

## 工程硕士学位论文

# 基于多信号流图的系统测试性 建模分析及软件设计

硕士研究生: 丁昊

导师: 吕超 副教授

申请学位: 工程硕士

学科、专业: 电力电子与电力传动

所在单位: 电气工程系

答辩日期: 2013年6月

授予学位单位: 哈尔滨工业大学

Classified Index: TP206

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

**MODELING, ANALYSIS AND  
SOFTWARE DESIGN FOR SYSTEM  
TESTABILITY BASED ON  
MULTI-SIGNAL FLOW GRAPH**

<b>Candidate:</b>	Ding Hao
<b>Supervisor:</b>	Assoc.Prof. Lyu Chao
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Specialty:</b>	Power Electronics and Electrical Drive
<b>Affiliation:</b>	Dept. of Electrical Engineering
<b>Date of Defence:</b>	June, 2013
<b>Degree-Conferring-Institution:</b>	Harbin Institute of Technology

## 摘 要

随着科技的发展,大型系统与大型设备的性能不断提高,与此同时,系统的复杂度也随之增长,导致故障诊断的难度日益加大,并产生高昂的测试费用,因此对系统进行测试性设计与分析的需求也越来越迫切。目前国内缺少自主开发的测试性设计分析软件,已经成为制约测试性研究发展的关键问题。因此,深入研究系统测试性设计与分析理论,自主研发用于测试性设计、分析的软件具有重要的意义。

首先,研究了多信号流图模型的建模思想,结合具体实例对建模步骤进行了详细阐述;制定了测试性图形化建模软件的总体设计方案。按照该方案通过 Visio 绘图控件完成对 Visio 的二次开发,建立了图形化建模环境;开发了建模所用的工具箱;研究实现了文本文档对模型属性的读写、故障-测试相关矩阵 Excel 表格形式的输出等功能。

其次,研究了多信号流图模型相关矩阵的生成方法。编程实现了对 Visio 后台系统文件的读取、分析,得到了模型的邻接矩阵;研究了由邻接矩阵求取可达性矩阵的算法;提取可达性矩阵中故障对应的行、测试对应的列生成完全故障矩阵;在完全故障矩阵之上,结合故障和测试功能信号的相关性生成了功能故障相关矩阵;最后完全故障矩阵和功能故障矩阵合成相关矩阵。

再次,结合雷达发射机系统的实例,利用其相关矩阵,从静态测试性分析和动态测试性分析两个方面对测试性分析进行了详细研究。静态测试性分析中,主要通过对矩阵行列关系的分析得到故障检测率、故障隔离率、模糊组、冗余测试和未检测故障等参数;动态测试性分析中,通过引入遗传算法完成了测试集合的求解;利用 AO\*算法以搜索与或图的形式完成了测试序列的求解。

最后,介绍了所开发测试性图形化建模软件与分析软件的各项功能,并以雷达发射机系统为例,通过与国外测试性商业软件 TEAMS 的对照,分别对建模与分析两部分的功能进行了验证,其分析结果的一致,说明了开发方案的可行性、开发软件的可靠性。

课题研究工作实现了图形化输入、模型建立、测试性分析和结论输出等测试性分析软件工具的关键组成部分,界面友好、结论准确、自动化程度高,基本达到了目前商业化测试性分析软件的水平。

**关键词:** 测试性分析; 多信号流图模型; 图形化建模; 相关矩阵

## Abstract

With the development of science and technology, there appear all kinds of large scale systems and equipments with more powerful functions. Accordingly, the structure of system is more complex which leads to the difficulty of fault diagnosis and the high fees of maintenance. For the reasons above, it is necessary to conduct testability design and analysis. Lack of domestic self-developed testability design and analysis software has become a major shackle which has restricted the development of the domestic testability research. Therefore, in-depth study theories of system testability and the independent research and development of software for testability design and analysis are in great significance.

Firstly, this paper studies the modeling idea of multi-signal flow graph model, illustrates the modeling steps with a concrete example; and develops an overall design scheme of the graphical modeling software for testability. By embedding the Visio drawing interface to build a multi-signal flow diagram drawing environment, this paper develops a toolkit of components needed for modeling. This paper completes properties of the module by using the text documents attribute set, save, and outputs dependency matrix by the use of Excel table.

Secondly, this paper discusses the method to generate dependency matrix based on multi-signal flow graph; through analysis of the Visio backstage system files obtains the adjacency matrix, then to get reachability matrix, and to complete failure matrix by extraction the corresponding fault line and corresponding test column of reachability matrix, unites fault and test functions to generate a functional failure correlation matrix, then combines complete failure matrix and functional matrix to dependency matrix.

Thirdly, taking the radar transmitter system as an example, through analysis of the dependency matrix, completes system test analysis. In static testability analysis, through the analysis of the dependency matrix information of obtains indicators such as fault detection rate, fault isolation rate, undetected faults, ambiguity groups, and redundant test will be ranked. In dynamic testability analysis, genetic algorithm is used to find the optimal test set; AO\* algorithm is used to search for and/or figures to complete the solution of the test sequence.

Finally, taking the radar transmitter system for instance, this paper introduces the testability graphical modeling software and analysis software functions. Through comparing with foreign commercial software testability-TEAMS, the same result verifies the reliability of the developed program and software.

This paper realizes the important components of testability analysis software, such as graphic user input interface, model set up, testability analysis, and results output. The software has the features of friendly interface, good accuracy, and a high degree of

automation. It is considered to be comparable with the present commercial testability software.

**Keywords:** Testability analysis, Multi-signal flow graph model, Graphical modeling, Dependency matrix

# 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 国外测试性研究概况 .....	2
1.2.1 测试性建模分析技术 .....	2
1.2.2 测试性辅助工具研发 .....	3
1.2.3 测试性技术应用 .....	4
1.3 国内测试性研究概况 .....	5
1.3.1 测试性建模分析技术 .....	5
1.3.2 测试性辅助工具研发 .....	6
1.3.3 测试性技术应用 .....	6
1.4 课题研究总体方案 .....	7
1.5 本文主要研究内容 .....	7
第 2 章 多信号流程图模型的图形化建模软件设计实现 .....	9
2.1 引言 .....	9
2.2 多信号流程图测试性建模方法研究 .....	9
2.2.1 多信号流程图模型基本理论 .....	9
2.2.2 多信号流程图建模步骤 .....	11
2.2.3 多信号流程图建模实例 .....	11
2.3 图形化建模软件设计方案 .....	14
2.4 Visio 绘图控件开发 .....	15
2.4.1 Visio 控件的应用 .....	15
2.4.2 Visio 对象模型的引用 .....	17
2.5 图形化建模软件的设计实现 .....	20
2.5.1 数据模型的构建 .....	21
2.5.2 模型属性读写的实现 .....	22
2.5.3 相关矩阵输出的实现 .....	23
2.5.4 绘图工具箱的开发 .....	24
2.6 本章小结 .....	25

<b>第 3 章 故障-测试相关矩阵的生成</b>	26
3.1 引言	26
3.2 完全故障相关矩阵的生成	26
3.2.1 邻接矩阵的定义与求取	26
3.2.2 可达性矩阵的定义与求取	28
3.2.3 完全故障相关矩阵的生成	30
3.3 功能故障相关矩阵的生成	33
3.3.1 功能故障相关矩阵的定义	33
3.3.2 功能故障相关矩阵的生成	34
3.4 故障-测试相关矩阵的生成	35
3.5 本章小结	36
<b>第 4 章 基于多信号流图模型的测试性分析</b>	37
4.1 引言	37
4.2 测试性分析	37
4.2.1 静态测试性分析	37
4.2.2 动态测试性分析	38
4.3 测试选择问题研究	39
4.3.1 遗传算法的模型构建	40
4.3.2 算法有效性验证	43
4.4 测试序列求解方法研究	46
4.4.1 测试序列求解问题的数学模型	46
4.4.2 与或图表示形式	46
4.4.3 启发式搜索算法	47
4.5 本章小结	50
<b>第 5 章 软件实现及验证</b>	51
5.1 引言	51
5.2 测试性图形化建模软件	51
5.2.1 图形化建模软件功能介绍	51
5.2.2 图形化建模软件功能验证	52
5.3 测试性分析软件	54
5.3.1 测试性分析软件功能介绍	54
5.3.2 测试性分析软件功能验证	55
5.4 本章小结	57

结 论 .....	58
参考文献 .....	59
攻读学位期间发表的学术论文 .....	63
哈尔滨工业大学学位论文原创性声明和使用权限 .....	64
致 谢 .....	64



# 第1章 绪 论

## 1.1 课题背景及研究的目的和意义

随着当今科学技术的持续进步，各类大型系统的性能不断提高，系统结构和功能的复杂性也大大增加。为了确保系统的可靠持续运行，必须对其运行状态进行实时在线监控，对系统发生的故障进行准确的检测、诊断与隔离维修。在这个过程中，不可避免地遇到测试时间长、故障诊断困难以及日常维护耗时长、费用高等问题。针对这些问题，相关研究人员开展了广泛的研究工作，提出了大量先进的测试观点和技术，研发了功能强大、种类齐全的测试装置系统。但是在研究和工程实践中愈发明显地发现单纯的强调外部测试系统的开发不能从根本上解决大型复杂系统的测试问题。为了能够进行快速准确的测试，必须在系统装备设计研发中考虑今后测试的需求，即设计研发中全面考虑测试与诊断的问题，使其具有良好的机内测试性，并为外部测试提供良好的测试特性和便捷的测试接口。为此，研究人员提出了测试性的概念，即系统或设备(产品)能够及时准确地确定其状态(可工作、不可工作或性能下降)并隔离其内部故障的一种设计特性<sup>[1]</sup>。

系统的测试性是便于对其进行测试与诊断的一种重要特性，自始至终贯穿于系统设计、开发、使用和消亡的整个生命周期中，对系统维护性的完善、可靠性的增强和可用性的提高具有重要影响。一个具有良好测试性的系统，能够大大减少其故障检测和隔离所需要的时间，从而显著缩短维修时间，并能够降低对维修人员的相应技能要求，从而达到提高系统可靠性、降低寿命周期费用的目的<sup>[2]</sup>。在分析设计阶段开展测试性分析设计，能够节省大量的建模时间，提高系统设计效率。对已经设计完备、正在运行使用的系统进行升级改造、日常维护，也需要进行测试性分析，来获得最优的测试方案用来指导维护工作，从而实现检修维护工作的及时、准确和低成本。

从上世纪七十年代起，国外的相关研究机构、高校就已经开始进行测试性方面的研究。时至今日，测试性分析已经被广泛认可并大量应用于航空、航天等相关领域之中<sup>[3,4]</sup>。其中，美国先后颁布了一系列相应的测试性标准规范，将测试性作为与可靠性、维护性并列的独立学科，并开发了测试性分析与设计辅助软件，具有代表性的有DSI公司的eXpress和QSI公司的TEAMS。在军用电子系统和武器装备的研发和使用中，美国军方通过测试性设计、分析的应用，取得了良好效果，不仅大大降低了电子系统和武器装备的维护费用，而且很好地延长了其使用寿命<sup>[1]</sup>。

目前，国内进行测试性分析主要采用国外的测试性商业分析软件eXpress和

TEAMS。它们均采用可视化的图形建模技术，建模过程直观、简单易学。但是，这两款软件都十分昂贵，且其进行指标分析所采用的方法和过程不为外界所熟知，国内测试性分析人员不是十分清楚整个测试性分析流程，不好把握问题本质，阻碍了测试性技术的更好应用。因此，根据国内测试性分析需求，研究现有测试性分析方法，充分考虑系统设计与维护的测试性分析需求，开发测试性设计及分析计算机辅助软件是十分必要的。这必将对国内测试性分析水平的提高、系统故障诊断能力的增强产生深远影响。

## 1.2 国外测试性研究概况

经过了 30 多年的研究探索，国外尤其是美国对测试性体系的认识十分完备，相应的学科建制十分齐全，其研究已经处于较高水平。

### 1.2.1 测试性建模分析技术

美国国防部在 1978 年、1983 年及 1985 年分别颁布了 MIL-STD-471A 通告 2 《设备或系统的 BIT、外部测试、故障隔离和测试性特性要求的验证及评价》、MIL-STD-470A 《系统及设备维修性大纲》和 MIL-STD-216《电子系统及设备的测试性大纲》。其中，MIL-STD-216 的颁布实施是测试性发展史上的重大里程碑。自此，测试性成为一门独立学科，其重要性不亚于维护性与可靠性<sup>[5]</sup>。美国制定并颁布的有关测试性方面的军用标准如表 1-1 所示。

表 1-1 美国制定颁布的部分测试性标准

标准编号	标准全称
MIL-STD-1591	机载故障诊断子系统的分析与综合
MIL-STD-2076	被测装置与 ATE 的兼容性要求
MIL-STD-2077	TPS 一般要求
MIL-STD-415D	测试性设备设计准则
MIL-STD-1309	测量、测试与诊断术语
MIL-STD-2084	电子系统和设备维护性要求

与此同时，非官方的机构和公司也结合各自研究成果，陆续发布了一些测试性手册和指南，例如，美国罗姆航空发展中心(Rome Air Develop Center, RADC)发布的《RADDC 测试性手册》、美国航空无线电公司(Aeronautics Radio Incorporation, ARINC)的《BITE 设计和使用指南》、高级测绘工程(Advanced Test Engineering, ATE)公司的《SMTA 测试性指南》等。大量测试性标准和指南的制定和颁布，很大程度上促进了测试性技术的普及和发展。

从上世纪 80 年代中后期开始，科研院所和高校开始了测试性建模技术的研究，提出了多种测试性模型。这其中比较具有代表性的是 DSI 公司提出的的相关性模型(dependency model)<sup>[6]</sup>、ARINC 公司提出的信息流模型(information flow

model)<sup>[7]</sup>以及康涅狄格大学提出的多信号流图模型(multi-signal flow graph model)<sup>[8]</sup>。

信息流模型是由 ARINC 公司的 William R. Simpson 和美国国防分析研究所(Institute for Defense Analyses)的 John W. Sheppard 在 80 年代后期共同提出的。他们完成了信息流模型基本结构的定义,确定了模型的数学描述过程,分析了基于该模型的测试性指标计算方法<sup>[9,10]</sup>。在此基础上, Inevicius 和 Simpson 研究了信息流模型在故障诊断和隔离方面的应用<sup>[11,12]</sup>。信息流模型充分考虑了功能(故障)和测试之间的相关关系,并以有向图的形式完成了这种相关关系的表述,这是信息流模型的创新性和优越性。但是,过分依赖模型的结构和用户的主观判断,会使模型存在不合理的地方,因此该模型比较适合进行定性分析,其测试性分析与诊断结果的可靠性较差。

美国康涅狄格大学在 20 世纪 90 年代提出了多信号流模型<sup>[13]</sup>。该模型的创始人 Somnath Deb 和 Krishna R Pattipati 在对逻辑模型、相关性模型、信息流模型进行研究总结的基础上,仔细分析了这些模型的优势和不足,以及导致模型不足的原因,创立了多信号流图模型。它结合了结构模型和信息流模型的优点,同时克服了以往模型的不足,将测试、信号和故障有机的结合起来,兼顾了系统的功能和结构,以定量的方式来表达测试性指标,对故障诊断策略的制定起到了至关重要的作用。多信号流图模型一经提出,就得到了测试性技术研究人员的关注,大量学者对其进行深入研究,取得了丰硕的研究成果<sup>[14-16]</sup>。

### 1.2.2 测试性辅助工具研发

国外除进行测试性技术理论研究外,也非常重视理论成果在生产实际中的转化、应用,一些研究院所、公司先后研发了测试性计算机辅助软件<sup>[17-20]</sup>,其中比较具有代表性的如表 1-2 所示。

表 1-2 测试性计算机辅助软件

测试性设计计算机辅助软件	软件研发机构
系统测试性与维护程序(STAMP)	ARINC 公司
系统测试性分析工具(STAT)	DSI 公司
武器系统测试性分析工具(WSTA)	DSI 公司
测试性分析与研究工具(START)	康涅狄格大学
诊断分析与维护工具箱(DARTS)	美军测试计量与诊断装备研究机构

上面所列举的绝大多数测试性辅助设计分析软件都是各国军方独立或联合科研院所、公司共同开发的,基本都是针对武器装备而开发的,面向民用系统装备的少之又少。近年来业内开始重视民用测试性辅助设计分析软件的研发,广泛使用的是 DSI 公司的 eXpress 和 QSI 公司的 TEAMS。

eXpress 是一个功能全面、高度集成的应用软件，用户能够在其提供的开发环境中进行系统故障诊断、健康管理预测(DPHM)和系统全面测试性工程的设计、采集、集成、评估和优化。eXpress 能够在系统的整个生命周期内节约巨大的短期和长期的成本，效果十分显著。

TEAMS (Testability Engineering and Maintenance System，测试性工程与维护系统)是一套商业软件，能够为用户提供系统的测试性、维护性、运行情况监控方案。它具有模块化推理(MBR—Model Based Reasoning)的优点和能力，即简便、直观、实用性强。为了更好、更快地向用户提供关于系统测试、维护及运行情况的监控方案以指导生产实践，TEAMS 能够采纳、输入专家的意见，更加人性化。TEAMS 提供的解决方案中设置的各项参数也能够及时的更新。此外，TEAMS 还引入了反馈机制。TEAMS Designer 是该软件套装的核心组件，用来为大型复杂系统和产品进行测试序列的设计和测试性设计分析。它使用新颖的多信号建模技术和信息理论-AI 算法，通过友好的 GUI 界面提供测试性设计环境；能够完成分析对象的测试性分析，找出对象模型中存在的问题，并提供详细的分析报告，如测试性指标报告、故障诊断树和故障相关性报告等，为系统测试性初步设计和故障诊断提供辅助决策<sup>[21,22]</sup>。

### 1.2.3 测试性技术应用

美国军方在军用电子系统和设备中广泛推广测试性技术，取得了良好成效。据美国海军调查，对 F/A-18、F-14、A-16E 和 S-3A 四种海军主战飞机的 239 个关键部件进行的测试性技术改进，将其使用和维修费用降低了 30%，并且指出，如果在飞机整个研发周期中都充分考虑测试性分析，能够将飞机全寿命周期费用降低到原来的 80%-90%。以 F-18 战斗机为例，数据统计分析表明，与它要替代的 F-4J 飞机相比，每飞行 1 小时所需的平均维护工时由 48 小时降低到 18 小时，节约了 30 小时。为了便于比较，根据 1979 年海军资源模型得出的 F-4J 飞机使用数据，制定出相当于 F-18 飞机服役 20 年的飞行计划(总计 262 万小时)，其飞行保障费用与 F-4J 飞机相比，可节约 20 多亿美元<sup>[23]</sup>。测试性技术在 F-22 战机上也得到了成功应用，成为改善装备维修性的重要手段。据统计，一个飞行中队的 24 架 F-22 在 20 年内的维修和保障费用估计比相同数量的 F-15 节省 5 亿美元。在其第四代战斗机 JSF (Joint Strike Fighter，联合打击战斗机)中，测试性技术得到了更加广泛应用，以此为主要支撑技术之一建立了自主维修保障系统，预计 JSF 飞机可减少维修人力 20%-40%，保障规模缩小 50%，出动架次率提高 25%，使用寿命达 8000 飞行小时，大大提高了维修保障效率和自动化、智能化程度。这充分表明在同等条件下，采用测试性分析技术能够大大提高系统效能<sup>[24]</sup>。

此外，测试性技术也已经在林肯汽车、某型号导弹发射车、黑鹰直升机、RAH-66 科曼奇直升机和 V-22 倾转旋翼机等为代表的测试性分析和设计中得到了卓越体现，获得了巨大的经济效益<sup>[25]</sup>。

### 1.3 国内测试性研究概况

我国测试性设计与分析相关研究起步较晚，在认识和技术方面都与国外存在着很大差距。国内一直到 80 年代中期才开始在军用电子系统的设计过程中要求测试性设计。近年来，测试性受到越来越多的关注，各方面研究不断取得进展。

#### 1.3.1 测试性建模分析技术

1995 年，当时的国防科工委颁布了国军标 GJB2547-95《装备测试性大纲》用来指导、规范国内测试性的设计、分析、验证和管理，大大推动了我国测试性设计理念的普及和发展<sup>[26]</sup>。1998 年颁布的 GJB3385-98《测试与诊断术语》规定了与硬件产品的测试和诊断相关的常用术语及其定义<sup>[27]</sup>。此后，我国又相继发布了一些测试性标准，如表 1-3 所示。这些标准的制定大大推动了我国测试性设计技术的普及和推广。

表 1-3 国内现行的测试性标准

标准编号	标准全称	制定颁布部门
SJ/T10566-1994	测试性总线第一部分：标准测试存取口与边界扫描结构	电子工业部
HB/Z301-1997	航空电子系统和设备测试性设计指南	航空总公司
HB7503-1997	测试性预计程序	航空总公司
GJB3385-1998	测试与诊断术语	国防科工委
QJ3050-1998	航天产品故障模式、影响及危害分析指南	航空总公司
QJ3051-1998	航天产品测试性设计准则	航空总公司
SJ20695-1998	地面雷达测试性设计指南	电子工业部
GJB3970-2000	军用地面雷达测试性要求	国防科工委
GJB2460-2001	侦查雷达测试性通用标准	国防科工委

测试性建模方面，目前深入研究和广泛应用的是信息流模型和多信号流图模型。杨鹏<sup>[28]</sup>提出了不确定模型的相关理论与建模方法，用来解决经典测试性模型只能描述故障信息的弊端。钱彦岭利用 EXPRESS-G 语言开发了测试性信息描述模型<sup>[29]</sup>，连光耀在其基础上开发了基于 XML 语言的测试性信息描述模型<sup>[30]</sup>，陈希祥研究了基于本体语言的测试性信息描述模型<sup>[31]</sup>。

测试性设计方面，研究的热点是基于模型进行测试的选择与诊断策略的构建。连光耀<sup>[30]</sup>利用粒子群算法求解测试选择问题；薛凯旋等基于信息流模型，利用改进的列表寻优法进行测试选择<sup>[32]</sup>；陈希祥等提出了基于布尔逻辑相关矩阵的测试优化选择方法，将其作为测试性方案的一部分进行研究<sup>[33]</sup>。诊断策略构建方面，龙兵提出了基于量子行为粒子群优化的诊断策略搜索算法，并用于卫星的故

障诊断<sup>[34]</sup>；田仲等基于相关性模型研究了测试点最少的诊断策略，设计了考虑可靠性和测试费用等因素的故障诊断策略<sup>[1]</sup>；杨鹏等针对多故障、多级维修、多模式、多回路、测试有先后约束、多值测试和测试不可靠的情况，提出了相应的诊断策略构建方法<sup>[35-38]</sup>。

### 1.3.2 测试性辅助工具研发

我国在测试性辅助工具研发方面起步较晚，开展测试性辅助工具研发的单位较少，投入的人力、物力明显不足，对测试性辅助工具研发的重视程度不够。目前，北京航空航天大学与可维创业科技公司共同研发了一款 CAD 软件—可维 ARMS，该软件能够完成对系统的测试性分析。国防科技大学开发了 TADES 软件，该软件能够实现系统的测试性需求分析、测试性指标分析、设计以及评估等功能。这两款软件开创了国内自主设计研发测试性辅助工具的先河，与国外的 TEAMS 软件相比，TADES 软件更加符合我国测试性相关国标的要求，更加贴近国内测试性技术的实际需求，在功能方面更加强大，但在稳定性、可靠性、可信度等方面还有待进一步改进。

此外，北京长城航空测控技术研究所(航空 634 所)、北京航天测控技术有限公司、电子科技大学以及北京联合信标测试技术有限公司等相关单位也都开展了测试性辅助工具的研发工作。

### 1.3.3 测试性技术应用

目前，越来越多的研究院所和高校注意到系统测试性设计与分析的巨大潜力和广阔前景，在各自擅长的研究领域积极开展了针对特定对象的测试性设计分析方法研究<sup>[39,40]</sup>。如国防科技大学不仅对装备测试性总体技术进行了详细研究，还对基于模型的测试性技术体系及建模、分析、设计与评估等关键技术展开了深入探索。北京航空航天大学、军械工程学院主要依据《装备测试性大纲》所提出的项目要求对相应技术开展了广泛研究。电子科技大学主要针对电子系统，特别是集成电路系统进行了大量的测试性设计工作。

国内相关单位针对具体设备，也开展了初步的研究，其中最为典型的是在航空机载设备和雷达系统上的应用<sup>[41,42]</sup>。其中电子科技集团第十四研究所研制的某型号有源相控阵雷达的机内测试系统，提出了针对大型复杂系统的测试性设计原则，实现了对雷达系统的检测；中国航空工业集团公司洛阳电光设备研究所(航空 613 研究所)对某型号飞机以 CPU 为核心的子系统进行了测试性设计；中国航天科工集团二院二十三所就某系列导弹相关电子设备开展了测试性设计工作；国防科技大学对导弹控制系统、鱼雷、机电跟踪伺服平台、直升机航姿系统等进行了测试性设计研究；电子科技大学在电子装备自动检测与故障诊断系统设计方面进

行了较深入的研究。

## 1.4 课题研究总体方案

在明确课题目的的基础上，通过对测试性建模分析体系中各个组成部分的原理及相互关系的充分研究，确定了本课题的总体方案，如图 1-1 所示。

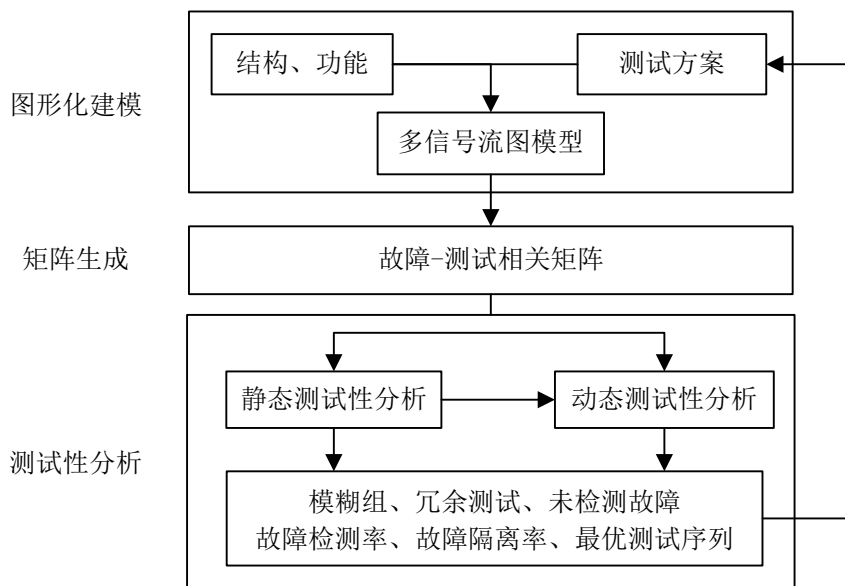


图1-1 本课题总体方案框图

首先，通过对系统结构和功能的分析，提取测试性建模的相关信息，建立系统的多信号流图模型，这是进行测试性分析的基础；然后，通过对多信号流图模型的分析，实现系统故障-测试相关矩阵的自动生成，该矩阵是测试性分析的具体研究对象；最后，通过对相关矩阵的分析完成系统的测试性分析，完成模糊组、冗余测试、未检测故障等测试性指标的计算，求出系统模型的故障检测率、故障隔离率，并生成系统的最优测试序列，用于指导测试方案的制定。

## 1.5 本文主要研究内容

本课题通过对多信号流图模型测试性建模理论与分析方法的研究，利用 Microsoft Visio 绘图控件开发了测试性建模软件，自动生成多信号流图模型故障-测试相关矩阵，通过对相关矩阵的测试性分析得到测试性分析数据，并生成测试序列，由此能够对测试方案进行优化改进，减少测试时间、降低测试成本，对系统实现更有效、更经济的运行维护。

本课题的主要研究内容包括以下几个方面：

(1) 研究了多信号流图模型的建模思想，并结合具体实例介绍了建模步骤；确定了测试性图形化建模软件的设计方案与开发平台，通过 Visio 绘图控件的二

次开发，完成图形化建模环境的构建；研究并实现模型属性的保存、写入和读取，及故障-测试相关矩阵的保存、输出。

(2) 详细讲述并完成了生成多信号流图模型故障-测试相关矩阵的算法。结合具体多信号流图模型，从求取完全故障相关矩阵和功能故障相关矩阵两个方面进行了具体讲述，分别给出了实现算法与具体流程图。其中，求取完全故障相关矩阵时，先分析模型有向图中相邻图元的连接关系，求出有向图的邻接矩阵，进而求出其可达性矩阵，通过对可达性矩阵的分析生成了完全故障相关矩阵。结合完全故障相关矩阵与测试、信号的相关关系，求取功能故障相关矩阵。

(3) 通过开发的测试性分析软件完成相关矩阵的分析，从静态分析和动态分析两个方面进行了测试性分析方法研究。其中，静态测试性分析主要识别系统的故障检测率、故障隔离率、未检测故障、模糊组和冗余测试；动态测试性分析给出最优测试序列，并计算出加权故障检测率和隔离率。

(4) 介绍了测试性图形化建模软件与测试性分析软件的各项功能，以某雷达发射机系统为例，分别对建模与分析两部分的功能进行了验证。



## 第2章 多信号流图模型的图形化建模软件设计实现

### 2.1 引言

多信号流图模型在形式上可以看作是系统结构模型上添加了多个信号流图。它在所要建模系统的结构分析、功能信号提取基础上,进行相应测试的添加,完成系统各组成部分可能发生的故障与测试之间相关关系的描述<sup>[43]</sup>。多信号流图模型通过在故障空间的建模,准确、简洁明了地表征了故障模式的传播过程,与其他建模方式相比,能够显著降低建模难度、减少建模开支,是目前应用最广、建模效率最高的测试性建模方法。

测试性分析软件的一个重要组成部分是图形化建模环境。本课题图形化建模环境的构建是通过对绘图工具 Visio 进行二次开发实现的,利用 Visio 中的简单图形来实现复杂系统模型的建立,降低了建模难度,提高了建模效率。

### 2.2 多信号流图测试性建模方法研究

#### 2.2.1 多信号流图模型基本理论

多信号流图模型是一种简单直观、条理清晰的建模方法,能够准确描述系统中的功能(故障)与模块的相关性,具有良好的建模效率<sup>[44]</sup>。

##### 2.2.1.1 多信号流图建模的基本思想

多信号流图模型的建模主要基于以下认识:

(1) 为了进行系统的故障诊断,测试性分析研究人员所关心的只是故障(或原因)是如何传播到各个测试点的。系统中的一个或多个元件发生故障,能够被不同的测试点的测试所检测到。进行 FMECA(Failure Mode, Effect and Criticality 故障模式、影响和危害),是为了跟踪故障的影响,并评估该故障对系统性能的影响。实施 DFT(Design For Test, 测试性设计),是为了确保系统完全的可观测性,以便测试性设计分析人员能够更好、更快地查明故障原因。在现场维护中,目标是在最短的时间内以最小的代价查找出故障产生的原因。在系统的故障空间进行建模能够满足上述几种情况的需求。因此,能够根据因果相关关系对系统进行建模,也就是研究故障节点是如何影响与它相邻的节点的<sup>[45]</sup>。

(2) 模型的故障空间是多维的,测试不是单纯的通过与否。由于功能空间是多维的,而不是一维的,因此相对应的故障空间也是多维的。例如,矩形波信号发生器可以生成指定振幅、相位和频率的矩阵波。如果矩形波信号发生器输出的矩形波不是指定的振幅、相位或频率,那么就可以认为矩形波信号发生器发生了

故障。

(3) 由于故障空间的多维性决定了故障状态的多维性、任意性，因此完全没有必要根据精确的定量关系来描述故障状态。为了更好的阐明这个认识，下面以一个三级放大器发生故障时的状态为例进行讲解。三级的增益分别为 2、3 和 4，其总增益为 24。在其发生故障时，总增益变为 12，不能因为第一级放大器的增益是 2 就认为一定是它发生了故障，任何一级放大器都有可能发生故障使总增益减半。因此，当多个组件中相同的属性发生变化时，定量分析能够表达的含义很有限。所以，只需要定义最重要的功能信号(或函数空间的尺度)并把它们与相应的组件和测试对应起来。

(4) 需要考虑两种不同类型的故障：完全故障和功能故障。以一个无源滤波器为例，它包含一个电感和一个电容。如果电感或电容发生故障，导致中心频率发生偏移，这是一个影响系统功能发挥的故障，即功能性故障。如果电路发生短路故障，使其输出功率变为 0，那么这是一个完全性故障，即造成信息中断、使其超出正常运行范围的重大故障。因此，一个模块的故障可能影响其专属属性，也可能影响所有与其相关联的属性<sup>[46]</sup>。

#### 2.2.1.2 多信号流图模型的结构

多信号流图模型相当于在结构模型基础上叠加了若干个单信号，主要包括以下几部分<sup>[47-49]</sup>：

- (1) 系统组件(可能发生故障的模块)的有限集合  $C = \{c_1, c_2, \dots, c_l\}$ ；
- (2) 与系统故障模块相关的信号集合  $S = \{s_1, s_2, \dots, s_k\}$ ；
- (3) 系统可用的测试点的有限集合  $TP = \{TP_1, TP_2, \dots, TP_p\}$ ；
- (4) 系统可用的测试的有限集合  $T = \{t_1, t_2, \dots, t_n\}$ ；
- (5) 每个测试点  $TP_p$  对应的测试集  $SP(TP_p)$ ；
- (6) 每个故障模块  $c_i$  相关的信号集合  $SC(c_i)$ ；
- (7) 每个测试  $t_j$  能够检测的信号集合  $ST(t_j)$ ；
- (8) 有向图  $DG = \{C, TP, E\}$ ，其中，有向图的边  $E$  表示的是系统的物理连接。

#### 2.2.1.3 多信号流图模型的矩阵表示

相关矩阵，即故障-测试相关矩阵，是多信号流图模型除有向图外的另一种表示形式。它在完成多信号流图模型的数学抽象基础之上，以布尔矩阵的形式描述了模型中所有可能发生的故障  $C = \{c_1, c_2, \dots, c_m\}$  与所有可用测试  $T = \{t_1, t_2, \dots, t_n\}$  之间的相关关系。其具体形式如下<sup>[50]</sup>：

$$D_{m \times n} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix} \quad (2-1)$$

其中,  $d_{ij}$  表示的是故障  $c_i$  与测试  $t_j$  的相关性。当故障  $c_i$  发生且测试  $t_j$  能够发现时,  $d_{ij}=1$ ; 否则  $d_{ij}=0$ 。

### 2.2.2 多信号流图建模步骤

多信号流图建模主要有以下 4 个步骤<sup>[44]</sup>:

- (1) 进行所要建模系统的结构和功能分析, 根据功能的不同完成系统结构的划分, 得到系统的组成模块及其功能信号, 确定能够添加测试的测试点位置、能够采用的测试及其测试信号等信息;
- (2) 根据对所建模系统的结构和功能划分, 绘制系统的结构框图;
- (3) 按照分析得到的模块、测试的特性, 进行相关信号的设置;
- (4) 为了符合系统的实际运行情况, 需要对所建模型进行适当的微调、校正, 并进行有效性验证。

### 2.2.3 多信号流图建模实例

某雷达发射机系统的结构框图如图 2-1 所示。根据对其结构和功能的分析, 将该系统划分为 10 个功能模块。同时, 图 2-1 也给出了该系统可以设置测试的 13 个测试点的位置。

根据 2.2.2 节所述的基于多信号流图模型的测试性建模方法, 雷达发射机系统测试性模型要素如下:

系统故障源  $C = \{c_1, c_2, \dots, c_{12}\}$ ; 与故障源相关的特征信号  $S = \{s_1, s_2, \dots, s_{26}\}$ ; 13 个测试点  $TP = \{TP_1, TP_2, \dots, TP_{13}\}$ ; 22 个测试  $T = \{t_1, t_2, \dots, t_{22}\}$ 。

每个故障源  $c_i$  的相关信号、每个测试  $t_j$  检测的信号与每个测试点  $TP_p$  所包含的测试的相互对应关系详见表 2-1。

各模块属性设置如下: 仅可能发生完全故障(G)的故障源是:  $c_{10}$ ; 仅可能发生功能故障(F)的故障源是:  $c_4, c_5, c_6, c_7, c_{11}$ ; 既可能发生完全故障(G), 又可能发生功能故障(F)的故障源是:  $c_1, c_2, c_3, c_8, c_9, c_{12}$ 。

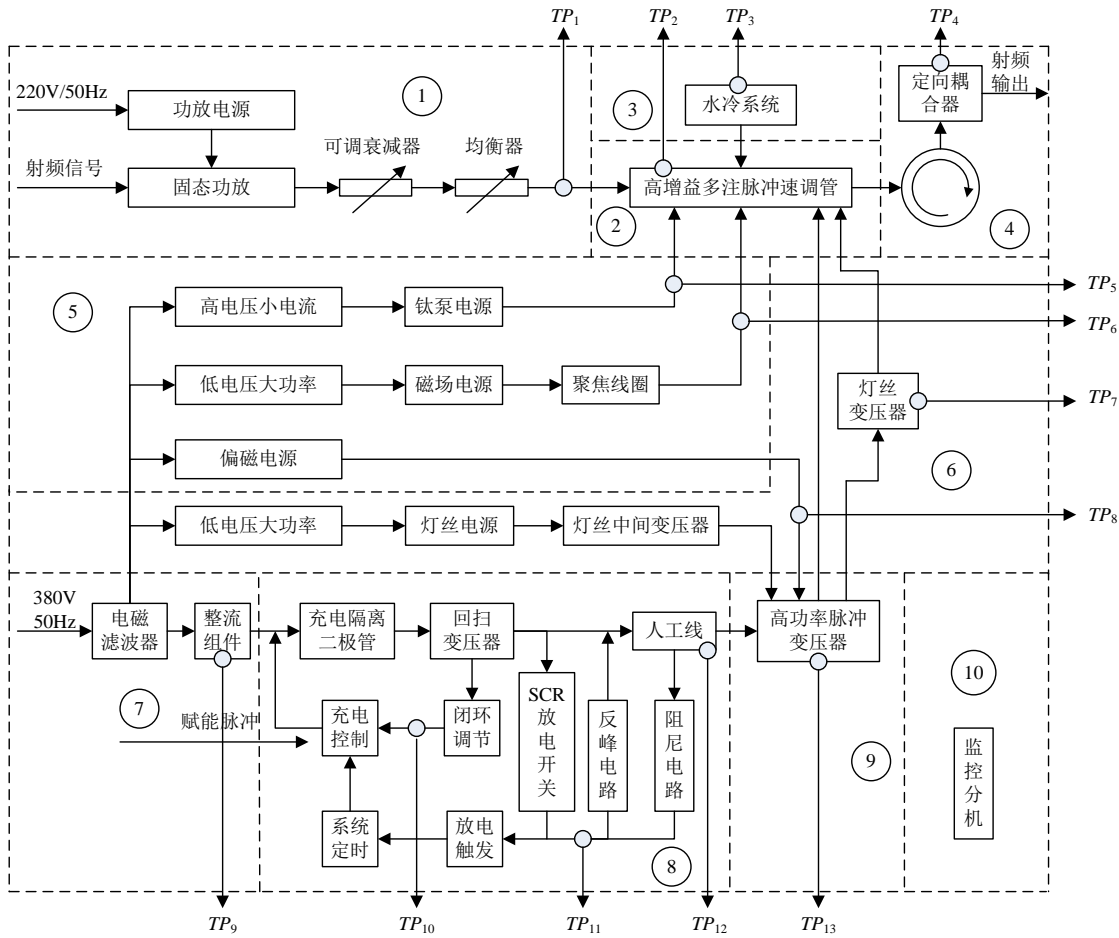


图 2-1 某雷达发射机系统结构框图

表 2-1 雷达发射机系统多信号流图模型组成单元对照表

编号	故障源	信号编号	特征信号	测试	测试点
$c_1$	功放分机	$s_1$	功放输出功率信号	$t_2$	$TP_1$
		$s_2$	放大器功率增益	$t_1$	
$c_2$	多注脉冲速调管	$s_3$	速调管幅频特性	$t_3$	$TP_2$
		$s_4$	速调管输入输出特性	$t_4$	
		$s_5$	速调管收集极电流	$t_5$	
		$s_6$	速调管管体电流	$t_6$	
$c_3$	水冷系统	$s_7$	收集极水流量	$t_7$	$TP_3$
		$s_8$	收集极水温	$t_7$	
		$s_9$	环流器水流量	$t_7$	
		$s_{10}$	管体水流量	$t_7$	

续表 2-1 雷达发射机系统多信号流图模型组成单元对照表

$c_4$	定向耦合器	$s_{11}$	射频包络	$t_8$	$TP_4$
		$s_{25}$	发射机输出功率	$t_{21}$	
$c_5$	钦泵电源	$s_{13}$	钦泵电压	$t_9$	$TP_5$
		$s_{14}$	钦泵电流	$t_{10}$	
$c_6$	磁场电源	$s_{12}$	磁场电流	$t_{11}$	$TP_6$
$c_7$	偏磁电源	$s_{16}$	偏磁电流	$t_{13}$	$TP_8$
$c_8$	灯丝电源	$s_{15}$	灯丝电流	$t_{12}$	$TP_7$
$c_9$	高压电源	$s_{17}$	高压电源电压	$t_{14}$	$TP_9$
		$s_{18}$	高压电源纹波系数	$t_{15}$	
$c_{10}$	充电控制分机	$s_{19}$	赋能电流	$t_{16}$	$TP_{10}$
		$s_{20}$	调制器充电电流	$t_{17}$	
		$s_{21}$	IGBT 驱动信号	$t_{18}$	
$c_{11}$	人工线	$s_{22}$	调制器反峰电流	$t_{19}$	$TP_{11}$
		$s_{23}$	人工线电压	$t_{20}$	$TP_{12}$
		$s_{24}$	脉冲功率	$t_{20}$	
$c_{12}$	高功率脉冲变压器	$s_{26}$	调制波信号	$t_{22}$	$TP_{13}$

基于以上分析建立的雷达发射机系统多信号流图模型示意图如图 2-2 所示。

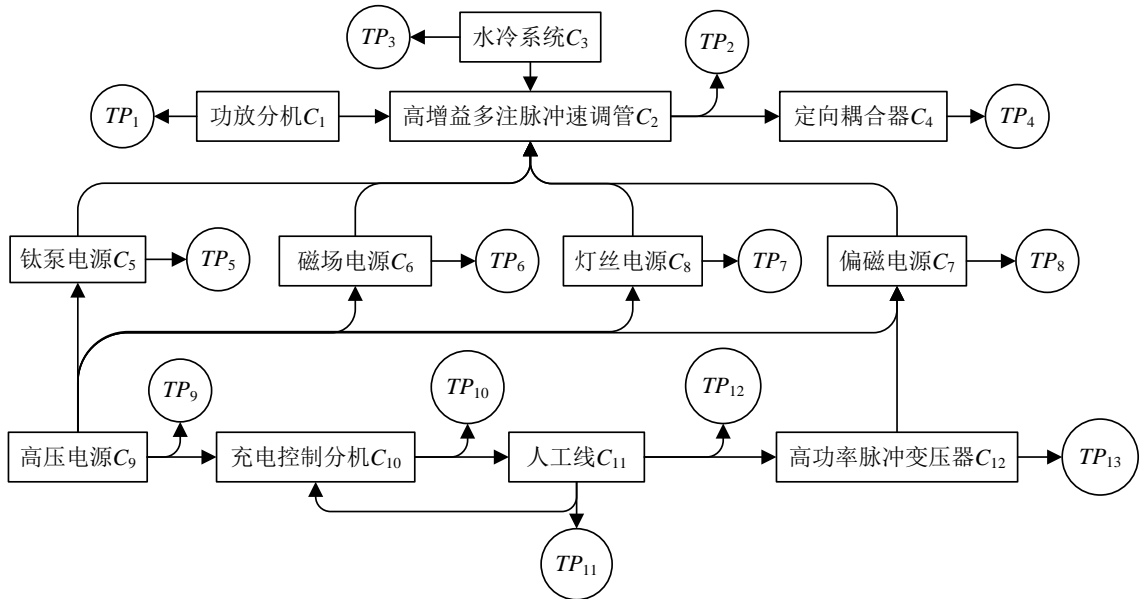


图 2-2 雷达发射机系统多信号流图模型示意图

## 2.3 图形化建模软件设计方案

基于多信号流图模型的测试性图形化建模软件总体设计方案如图 2-3 所示。用户首先完成所要建模系统结构原理的分析，对系统的结构和功能进行合理划分，然后以拖拽的方式从建模工具箱中选取多信号流图模型所需的图元到建模软件的绘图环境，完成多信号流图的绘制。在此基础之上，在属性设置对话框中进行故障模块和测试的属性设置，完成系统多信号流图模型的建立。通过相关矩阵生成模块完成对多信号流图后台系统文件的遍历，得到故障模块、测试间的连接关系，以此为基础求取多信号流图的邻接矩阵，根据邻接矩阵求取可达性矩阵，并结合属性设置模块提供的故障模块、测试的相关功能信号生成系统的故障-测试相关矩阵。

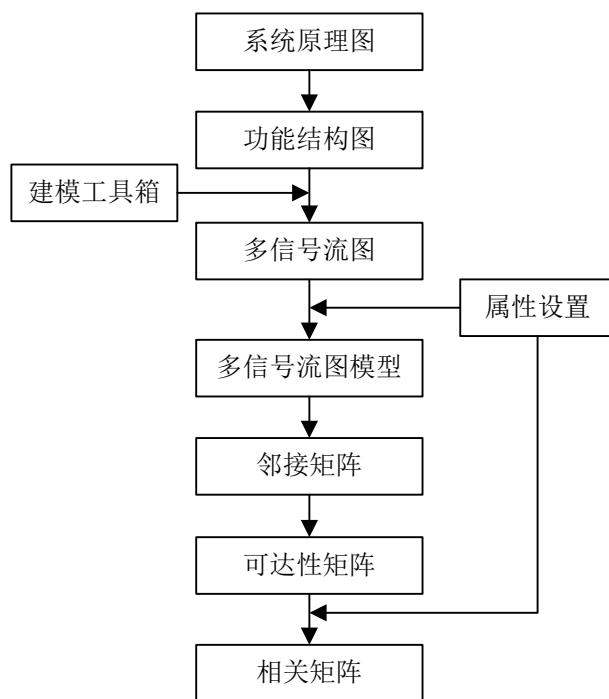


图 2-3 图形化建模软件总体设计方案

图形化建模软件的开发主要包含以下 4 部分：

### (1) 图形化建模环境

图形化建模环境是软件的主编辑界面，用户在该界面能够利用从建模工具箱中选择的图元完成多信号流图的绘制。本课题通过对绘图工具 Visio 进行二次开发实现了图形化建模环境的构建。

### (2) 属性设置模块

在属性设置模块中，用户能够完成故障模块和测试的属性设置，分别添加不同的相关功能信号，并实现属性的保存。

### (3) 相关矩阵生成模块

相关矩阵是多信号流图的等效表现形式，是后续测试性分析的基础。本课题开发了相应的程序来生成多信号流图模型的相关矩阵。

### (4) 建模工具箱

针对多信号流图模型中的模块、测试点、开关和表决节点以及连接线这五种组成要素，分别建立与之对应的图元，并生成一个模具文件作为图形化测试性建模软件的建模工具箱，方便用户使用。

## 2.4 Visio 绘图控件开发

Visio 绘图控件是一种 Microsoft ActiveX 控件。通过该控件用户能够实现对 Visio 绘图界面的访问，进而能够访问界面中的各类图形对象。在实现绘图界面的访问基础之上，用户使用该控件还可以将 Visio 绘图软件嵌入到自主开发的软件中，并提供其全部功能。在这个过程中，用户还能够自定义其外观<sup>[51,52]</sup>。

本课题通过对 Visio 绘图控件二次开发，将其嵌入到编写的程序中，完成测试性图形化建模环境的构建。采用这种基于已有成熟绘图软件二次开发的方式进行软件开发主要有以下几个方面的优势：

(1) 软件开发周期短、难度小。相比从底层开发一个全新的建模环境，这种二次开发的方式无论是在工作量上，还是开发周期以及难度上，都具有较大优势；

(2) 多信号流图模型中使用的图元相对固定，主要是故障模块、测试、开关、表决和连接线五种，都可以在 Visio 中找到相应的图元与之对应。Visio 现有功能完全能够满足测试性图形化建模软件的需求；

(3) Microsoft Visio 绘图软件是一个开放的系统，支持在其基础上的自主开发，而且 Visio 绘图控件提供了对 Visio 二次开发的途径与媒介。通过对 Visio 所绘制模型图后台文件的分析，能够得到模型间的连接关系，为后续的分析研究奠定了基础。

### 2.4.1 Visio 控件的应用

在安装 Visio 及 Visio 软件开发包(SDK, Software Development Kit)后，就可以在 Visual Studio 的工具箱中添加 Visio 控件。新建 C#语言的 Windows 应用程序，然后添加一个窗口资源作为程序开发的基本窗口。具体过程如图 2-4 所示。

如果工具箱中没有 Microsoft Office Visio 12.0 Drawing Control 控件，则右键工具箱选择“选择项”，选择 COM 组件中的 Microsoft Office Visio 12.0 Drawing Control，如图 2-5 所示。然后就可以像使用普通组件一样使用 Visio 绘图控件。

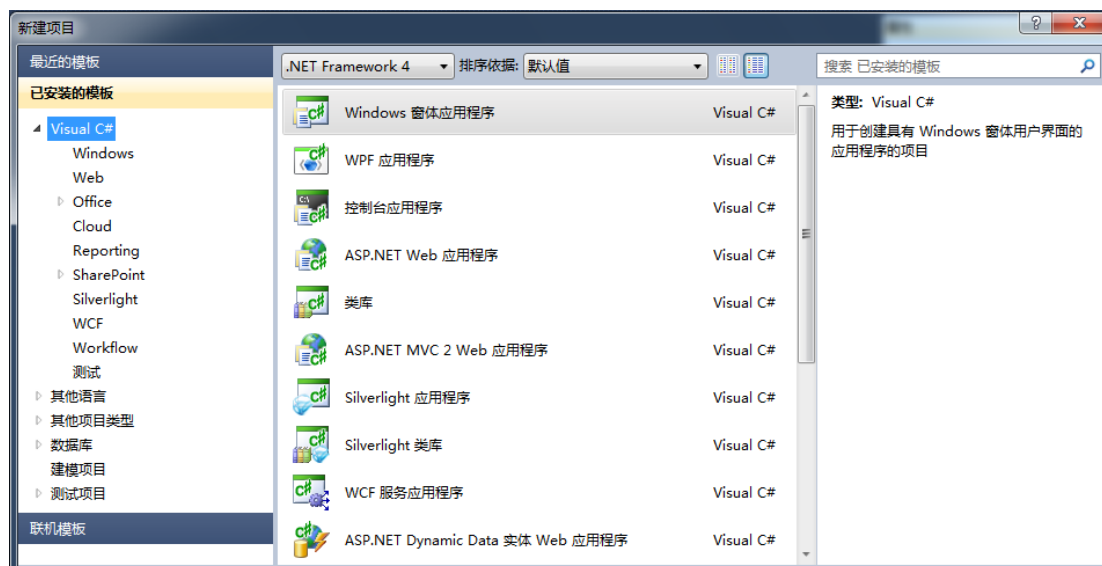


图 2-4 新建 C#语言的 Windows 窗体应用项目

向窗口中添加控件完成后会在程序中生成如下代码：

```
private AxMicrosoft.Office.Interop.VisOcx.AxDrawingControl axDrawingControl1;
```

然后就可以使用 axDrawingControl1 变量进行相关操作。

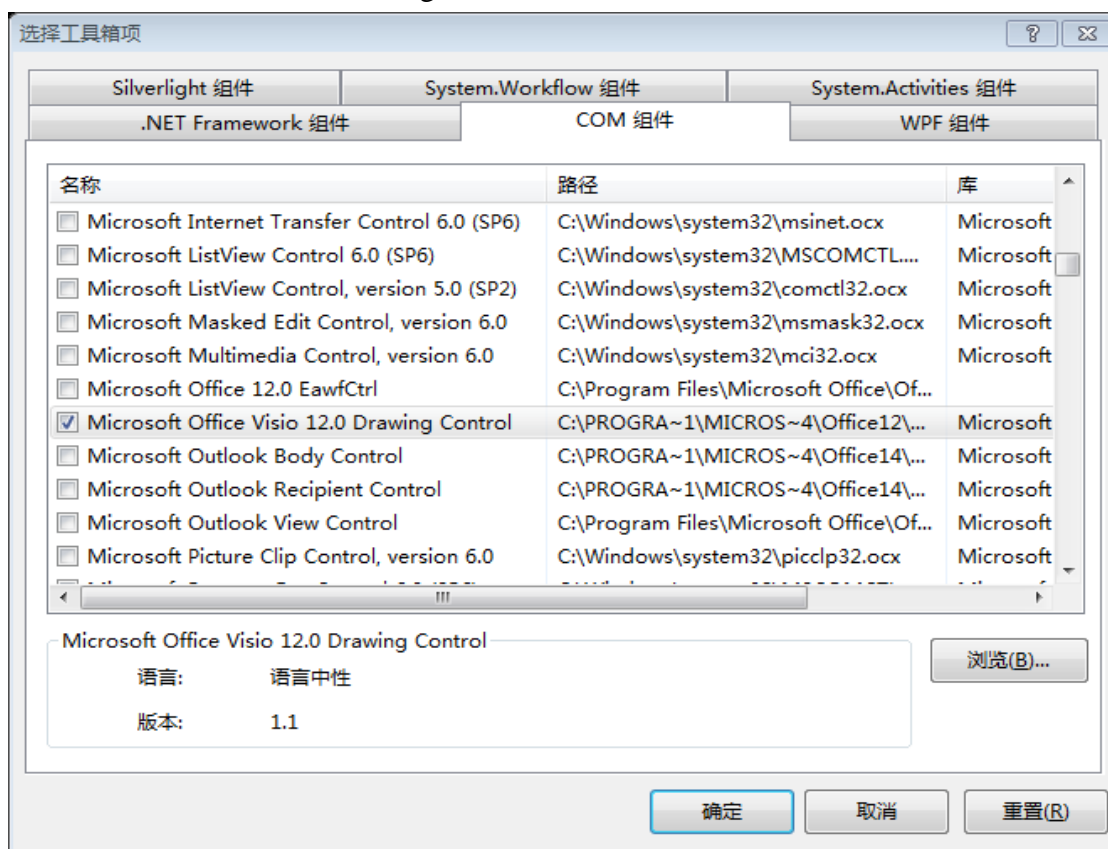


图 2-5 添加 Visio 控件到 Visual Studio 工具箱中



## 2.4.2 Visio 对象模型的引用

在完成 Visio 绘图控件的添加后，还要完成对 Visio 文件中对象模型的引用，才能以编程的方式来驱动这些模型。应用程序中的对象模型不仅仅指对象本身，还包括了对象的属性(数据)、方法(行为)和事件响应等<sup>[53,54]</sup>。用户在 Visio 主编辑界面中能够看到的图元都有着与之相对应的对象模型，对应关系如图 2-6 所示。例如，Shape 对象表示的是用户可以使用指针选中的 Visio 绘图页面中的内容，这包括页面中的图形、组合、辅助线、控件，或者被链接、嵌入或导入到 Visio 绘图图中的其他应用程序中的对象。

对象模型的集合也是一种对象。“集合”包含同种类型的若干个对象。例如，在 Visio 应用程序中 Page 对象表示打开的某一个绘图页面；而 Pages 集合则表示在该应用程序中打开的全部页面，包括全部的绘图页面、前景页、背景页等。

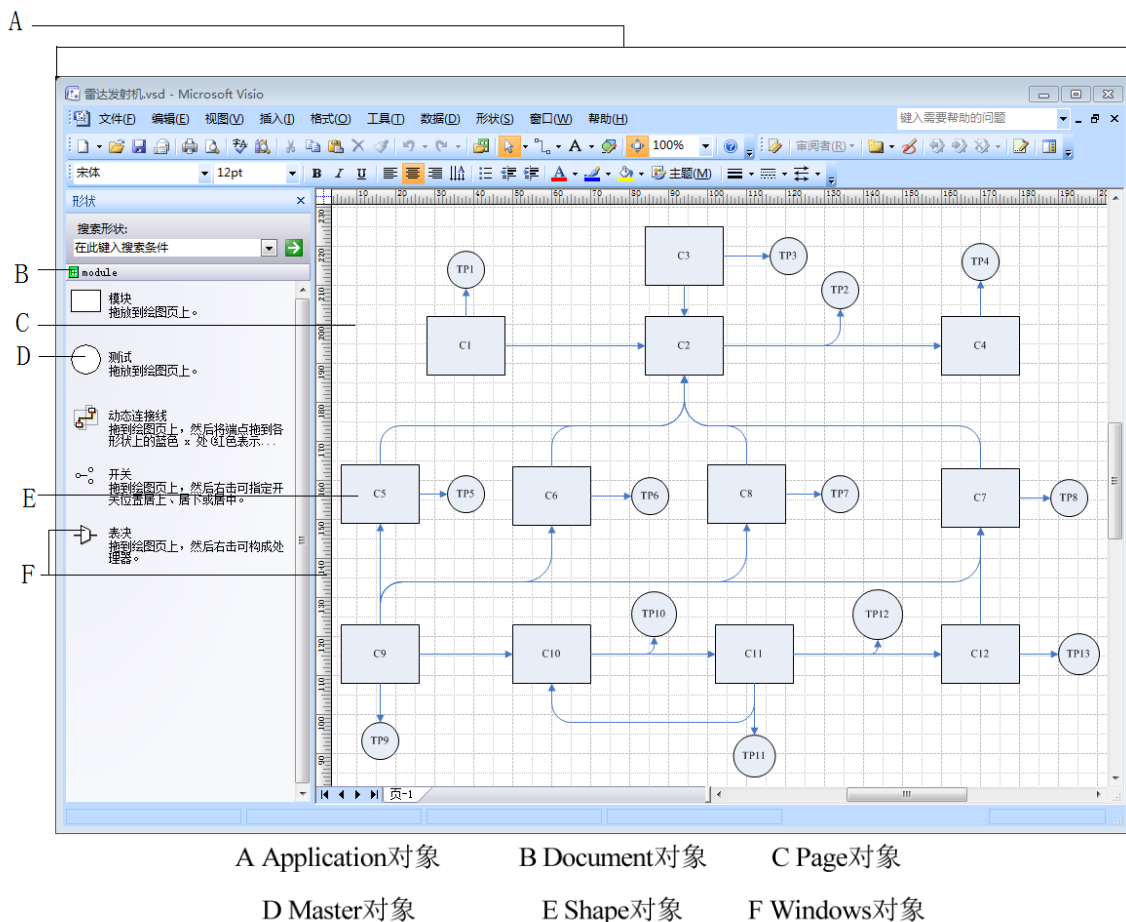


图 2-6 Visio 对象模型中对象与 Visio 绘图页面中图元的对应关系示意图

用户在使用 Visio 对象模型进行操作时，具体操作要分两步进行：第一步，用户需要完成对所要操作的 Visio 对象模型的引用、声明；第二步，用户通过对

Visio 对象模型的属性和方法的操作来完成某些既定操作。一般来讲,在实现对所要操作的对象的引用时,用户都是按照 Visio 对象模型级别的高低从高到低逐级完成引用<sup>[55]</sup>。Visio 对象模型的从属关系如图 2-7 所示。其中,一些高级别的对象模型存在的意义就在于提供对其他低级别对象模型的访问。例如,要完成对 Connect 对象的引用,需要先完成对象模型中较高级别对象的引用,在此处具体是指其存在的载体 Shape 对象和包含 Shape 对象的 Page 对象。

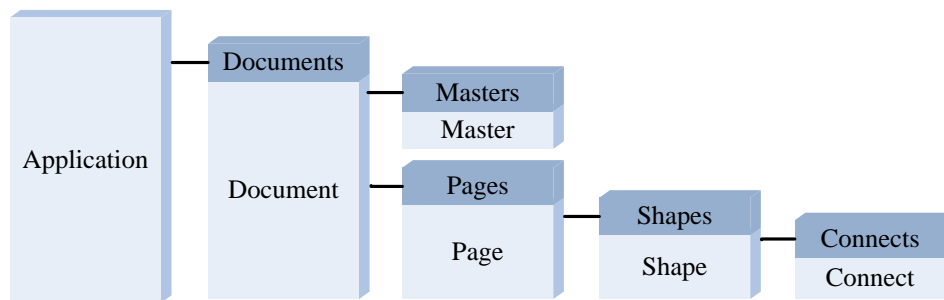


图 2-7 Visio 对象模型从属关系示意图

下面介绍本课题设计图形化建模软件中使用到的对象模型及其相关操作。

本课题通过 Visio 绘图控件的 API(Application Programming Interface, 应用程序编程接口)来访问 Visio 对象模型。

#### (1) Application 对象

Application 对象是调用 Visio 绘图控件对象模型的入口。使用绘图控件的 Document 或 Window 属性能够完成对 Visio Application 对象的引用。本课题利用下面的 C#代码获得对 Visio Application 对象的引用。

```

using Microsoft.Office.Interop.Visio;
private Microsoft.Office.Interop.Visio. Application application = null;
application=(Microsoft.Office.Interop.Visio.Application)
drawingControl.Window.Application;

```

#### (2) Document 对象

Document 对象表示的是在 Microsoft Visio 实例中打开的绘图文件(.vsd 或.vds)、样本文件(.vss 或.vsx)或模板文件(.vst 或.vtx)。通过对 Application 对象的 ActiveDocument 属性的操作能够获得当前活动文档对应的 Document 对象。

```

Document doc=this.axDrawingControl1.Window.Application.ActiveDocument;

```

通过如下代码将建模工具箱添加到应用程序中:

```

this.axDrawingControl1.Window.Application.Documents.OpenEx("D:\module.
vss", (short)VisOpenSaveArgs.visOpenRO+(short)VisOpenSaveArgs.visOpenDocked);

```

#### (3) Page 对象

Page 对象表示绘图页面。要使用绘图页面,同样需要按图 2-7 所示从属关系获得 Page 对象的引用。要获得文档的页面,需要获得 Document 或 ThisDocument

对象的 Pages 集合，并从该集合获得 Page 对象。可以通过它在集合中的索引来获得页面，或者可以按照名称来获得它。要获得应用程序的活动页面，可以通过获得全局对象或 Application 对象的 ActivePage 属性的方式。

本课题先获得当前活动文档，然后再去得到 Page 对象。

```
Pagepage=doc.Pages.get_ItemFromID(0);
```

#### (4) Shape 对象

Shape 对象是软件开发使用的最基本的 Microsoft Visio 对象。Shape 对象表示基本图形、组合图形、辅助点、辅助线，或者嵌入的对象等。此外，主图形、页面或文档的公式等也都属于 Shape 对象。

用户首先需要完成对 Shape 对象的引用，才能够得到 Visio 界面中图形的相关信息。为了完成 Shape 对象的引用，用户必须先完成对图形对象的 Shapes 集合的引用，这可以通过 Page 对象的 Shapes 属性来实现。

本课题直接采用 foreach 语句来遍历页面上的 Shapes 集合。

```
foreach(Shapeshapeinpage.Shapes)
{
    stringid=shape.Text;
    ...
}
```

#### (5) Connect 对象

在 Visio 绘图页面中，不同的 Visio 图形是可以被连接或粘附到另一个 Visio 图形中的。在 Visio 对象模型中，Connect 对象就是用来表示这种图形间的连接关系的，例如，组织结构图中的连线和模块框图等。

类似于前面对象的引用，通过高级别 Shape 对象的属相可以获得 Connects 集合，从而得到 Connect 对象。

在完成对 Connect 对象的引用后，就获得了相应的开发权限，能够查看图形间的相互连接关系，以及它们的连接方式。对于 Page，Master 和 Shape 对象，通过获得 Connect 对象的属性，还可以分析直接的图形(如流程图)或连接的图表(如组织结构图)。

Page 对象的 Connects 属性将返回一个包括页面中每个连接的 Connect 对象的 Connects 集合。Master 对象的 Connects 属性将返回一个包括主图形中每个连接关系的 Connect 对象的 Connects 集合。

本文通过 foreach 语句遍历 Shape 的 Connects 属性集合，得到元素的连接关系。第一次得到的是连接的起始元素，第二次得到的是连接的终点元素。

```
foreach(Connectconinshape.Connects)
{
```

```

if(x== 0)
{
    fid=con.ToSheet.Text;
}
else
{
    tid=con.ToSheet.Text;
}
x++;
}
    
```

## 2.5 图形化建模软件的设计实现

按照图形化建模软件总体设计方案，本课题完成了软件的开发。软件开发的相应程序框图如图 2-8 所示。

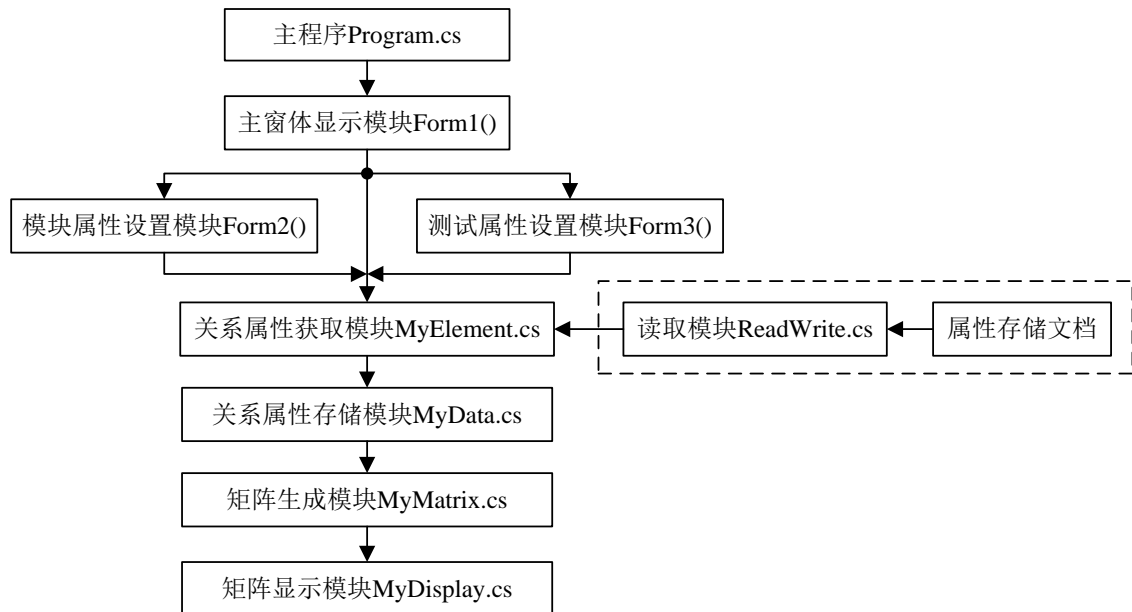


图 2-8 图形化建模软件程序框图

本课题所开发的测试性图形化建模软件如图 2-9 所示。该建模软件采用用户熟悉的 Windows 界面风格，用户能够进行多信号流图模型的绘制、编辑，完成模块的连接和模块故障属性的设置，得到故障-测试相关矩阵。为方便用户使用，图形化建模软件设计了建模所需图元的模具，并以工具箱的形式集成在软件中。软件还设计了功能齐全的菜单栏和简洁方便的工具栏，并完整兼容了 Visio 的右键菜单和 Windows 系统中的默认快捷键，极大地方便了用户的测试性建模，只需要用鼠标进行简单的拖拽、点击就可以完成绝大多数的操作，降低了建模的难度。

同时软件提供了文本文档对模块属性的批量操作，能够减少很多的重复性操作，提高了建模效率。

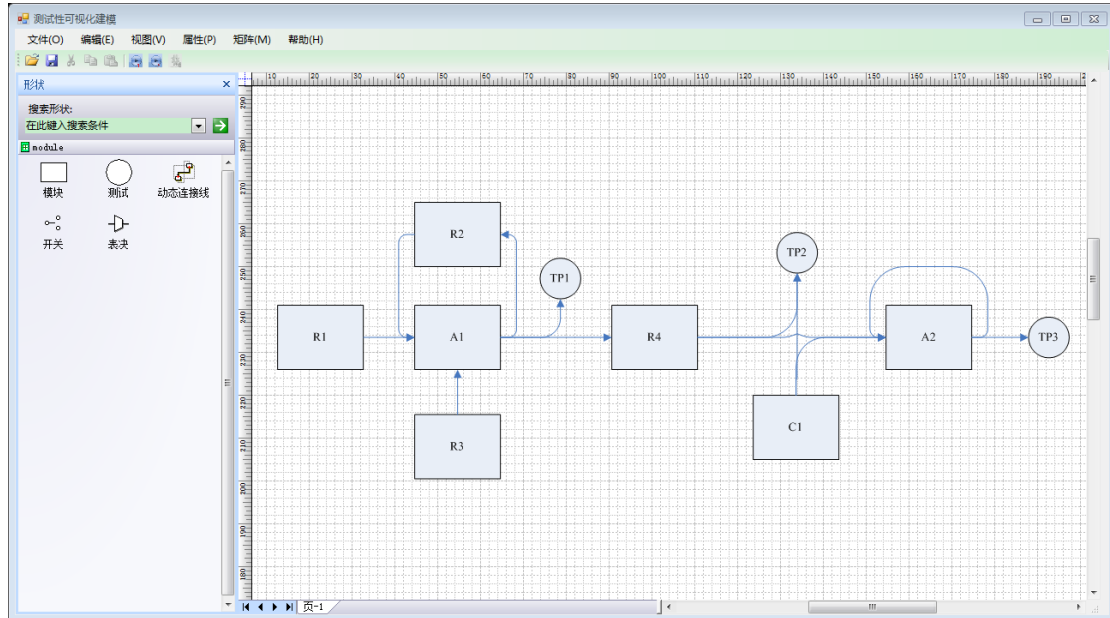


图 2-9 图形化建模软件的主界面

### 2.5.1 数据模型的构建

通过对 SelectionChanged 事件的监听，改变菜单项和工具栏项的状态，使用户可以进行属性设置，并且得到当前所选 Shape 对象，进而构建数据结构模型用于数据处理。添加 SelectionChanged 事件的 C#代码如下：

```
this.axDrawingControl1.SelectionChanged+=new AxMicrosoft.Office.  
Interop.VisOcx.EVisOcx_SelectionChangedEventHandler(this.axDrawing  
Control1_SelectionChanged);
```

本课题为模块和测试点定义了一个数据类，用于保存基本信息和属性，具体定义如下：

```
class MyElement  
    private int mindex;           //节点索引值  
    private string id;           //节点名称  
    private string type;         //节点类型  
    private List<string> signals; //模块属性  
    private Dictionary<string, string> ts; //测试属性  
    private List<MyElement> parents; //节点邻接信息  
    private List<MyElement> children; //节点邻接信息
```

初始化及添加节点部分代码如下：

```
MyElement ele = new MyElement(id, type);
```

`MyData.getInstance().addElement(ele);`

### (1) 模块属性设置

通过图 2-10(a)所示的对话框实现模块属性的设置，通过两个属性变量得到模块的名称和相关信号。用户点击确定按钮的时候，将属性添加或更新到统一数据结构 `MyData` 类中，为下面矩阵的生成准备好数据。

### (2) 测试点属性设置

和模块节点类似，利用下面图 2-10(b)所示的对话框实现测试点属性的设置，两个属性变量分别得到测试点名称、测试与测试信号。当用户点击确定的时候，同样将属性更新或添加到数据类 `MyData` 中，为相关矩阵的生成准备数据。

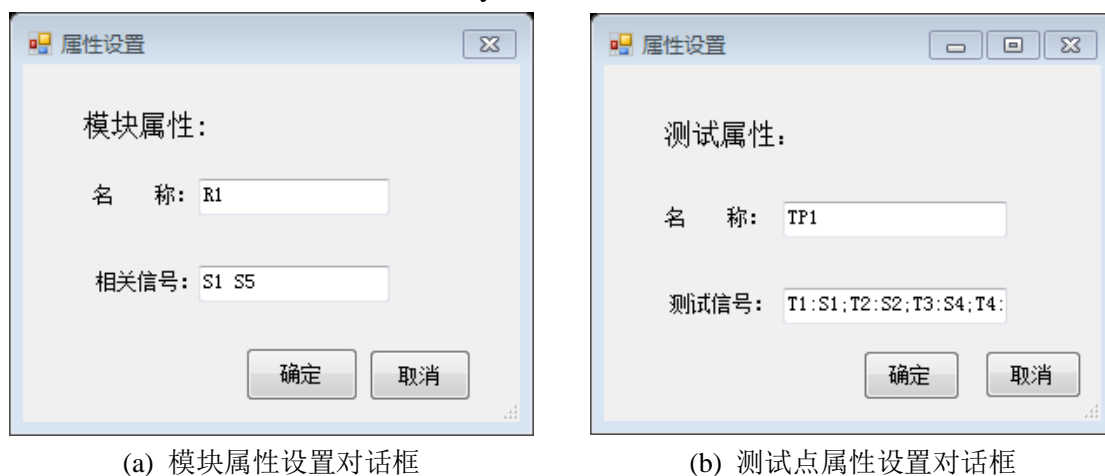


图 2-10 属性设置对话框

## 2.5.2 模型属性读写的实现

完成模块属性和测试点属性设置后，本课程开发的建模软件提供模型属性的导出功能；对于已经存在的模型，提供模型属性的导入功能。

模型属性导出过程：遍历已保存数据结构 `MyData` 中的数据元素，对模型中已完成属性设置的节点，把节点名称和属性信息按照如下格式写入文本文档，方便以后的属性批量导入设置：

```
节点名称 1
模块属性
...
节点名称 m
测试属性
...
```

模型属性导入过程：将之前已保存的属性文件或手工按照上述既定格式编写的属性文件导入到当前模型中。详细过程如下：读取文件中节点名称和相关属

性，对于模型中部分已设置属性的节点进行属性的更新，以文件中属性为最新属性；若读取到的节点未设置属性，即在当前数据结构 MyData 中未找到相应元素，则对模型遍历搜索，若找到对应节点，完成属性设置，并将此节点加入到数据结构 MyData 中。若未找到对应名称的节点，提示文件读取有误，跳过该节点，继续直至完成对文件的读取。

### 2.5.3 相关矩阵输出的实现

为了方便用户查看和后续对相关矩阵的操作，本课题所设计图形化建模环境选择 Microsoft Office 的另一组成单元 Excel 作为相关矩阵的输出载体。要在 C# 语言开发的软件环境中完成 Excel 文件的生成，首先需要添加对 Excel 引用，添加引用过程如图 2-11 所示。

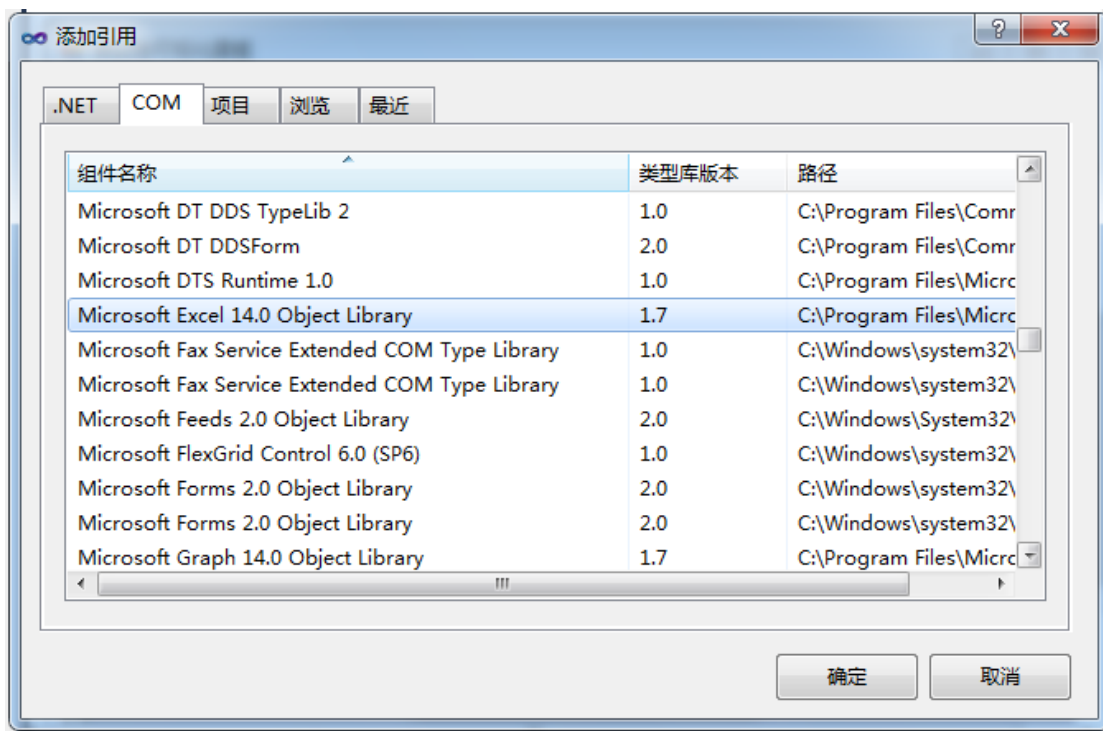


图 2-11 添加 Excel 的引用

然后导入 Excel 命名空间，方便在程序中使用 Excel 等相关对象，其操作步骤类似 Visio 对象的引用，使用的语句如下：

```
using Excel = Microsoft.Office.Interop.Excel;
```

(1) 获取 Excel Application 对象

```
Excel.Application xapp = new Excel.Application();
```

(2) 添加 Excel 工作簿

```
Excel.Workbook xBook = xapp.Workbooks.Add (Missing.Value);
```

(3) 得到 Excel 工作表



```
Excel.Worksheet xSheet = (Excel.Worksheet)xBook.Sheets[1];
```

(4) 为单元格赋值

```
xSheet.Cells[i, j] = value;
```

(5) 表格内容填充完成后，选择保存路径和文件名，保存表格，关闭工作簿和 Excel 应用对象

```
xSheet.SaveAs(filepath_name,Missing.Value,Missing.Value,Missing.Value);
```

```
xBook.Close(Type.Missing, Type.Missing, Type.Missing);
```

```
xapp.Workbooks.Close();
```

```
xapp.Quit(); //确保 Excel 对象从内存中退出
```

## 2.5.4 绘图工具箱的开发

模具是为了绘图方便而集中在一起的一组相关图件的集合。Visio 提供了大量的不同类型的模具，利用模具中的图件可以迅速生成相应的图形。Visio 还提供模具的自主开发，模具的新建过程如图 2-12 所示。打开 Visio 程序，选择文件->形状->新建模具，创建软件中用到的模具。从形状中选择软件中需要的形状，并且重命名。模具创建完成后保存为.vss 文档，供图形化测试性建模软件使用。

结合本课题测试性图形化建模的实际需求，创建并绘制了如图 2-13 所示的模具作为软件的建模工具箱。该模具包括了测试性建模所需的五种图元，它们是模块、测试点、动态连接线、开关和表决，能够很好的满足测试性建模需求。

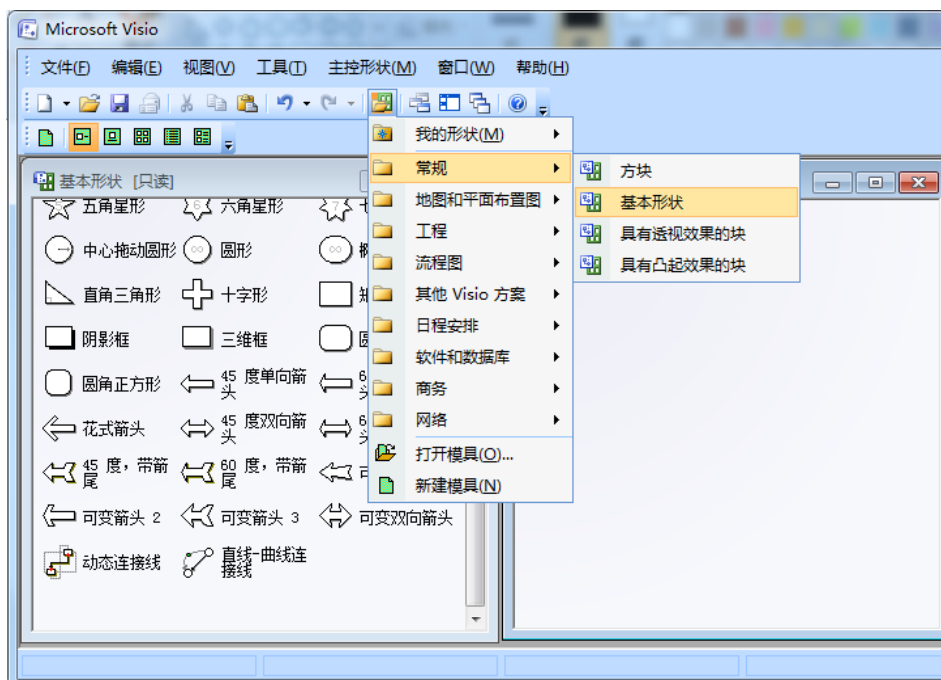


图 2-12 模具的新建过程



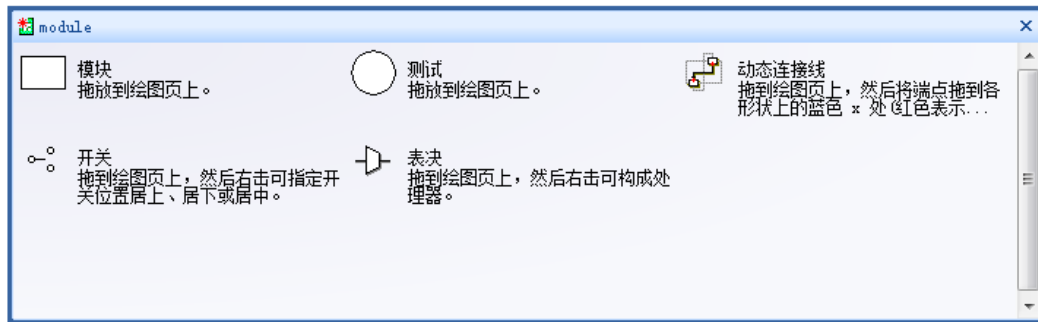


图 2-13 建模工具箱

## 2.6 本章小结

本章首先研究了多信号流图模型建模的思想，并结合具体实例讲解了建模步骤，然后确定了测试性图形化建模软件的总体设计方案，在实现 Visio 绘图控件开发的基础上，完成了图形化建模环境的构建，最后介绍了如何通过文本文档读写模型属性以及如何实现故障-测试相关矩阵的 Excel 表格形式的输出。

## 第3章 故障-测试相关矩阵的生成

### 3.1 引言

在多信号流图模型中，通过定义故障模块和测试之间的关联信号来确定二者之间的相互关系，并以此为基础生成故障-测试相关矩阵，进而通过对相关矩阵的分析完成模型的测试性分析与故障诊断。因此，可以看出故障-测试相关矩阵是连接多信号流图模型与测试性分析的纽带和桥梁。被测对象的故障-测试相关矩阵是其组成单元的故障模式与设置的测试之间相关性的数学表示，是多信号流图模型的等价表现形式<sup>[56]</sup>。

采用本文第二章所开发的测试性图形化建模软件建立被测对象的多信号流图模型，通过对模型后台系统文件的读取识别所有节点之间的连接关系，并结合节点之间的信号关联关系生成相关矩阵。通过本文第二章所讲述的多信号流图模型理论可知，多信号流图模型的故障分为两大类：一类是完全故障，另一类是功能故障。因此，故障-相关矩阵也包含两大类信息<sup>[45]</sup>：完全故障信息和功能故障信息。大型复杂被测对象的节点数目众多，故障与测试之间的相关关系较为复杂，按照文献[2]中所提出的方法，通过遍历模型图的方式直接求取相关矩阵，其算法复杂、处理过程繁琐。本课题先遍历模型图生成完全故障相关矩阵，然后通过分析完全故障相关矩阵和测试的相应关联信号的关系来求取功能故障相关矩阵，最后把完全故障矩阵和功能故障矩阵组合在一起就得到所分析对象的故障-测试相关矩阵。

### 3.2 完全故障相关矩阵的生成

完全故障相关矩阵表示的是测试能否检测到故障，即所在节点之间是否具有可达性。因此，可用简单有向图普通节点的分析方法来分析故障模块和测试点的相关关系。本课题首先将被测对象的多信号流图模型看作简单的有向图，得到其邻接矩阵。其次采用 Warshall 算法从邻接矩阵得到可达性矩阵。最后提取各个模块所在的行和各个测试所在的列，生成完全故障相关矩阵。

#### 3.2.1 邻接矩阵的定义与求取

邻接矩阵是一个用来表示有向图中各元素之间连接状态的布尔矩阵，具体定义如下：

有向图  $G=\langle V, E \rangle$  中，有  $n$  个结点  $V=\{v_1, v_2, \dots, v_n\}$ ， $m$  条边  $E=\{\langle v_i, v_j \rangle | v_i \in V, v_j \in V\}$ 。有序对  $\langle v_i, v_j \rangle$  表顶点  $v_i$  到  $v_j$  有一条有向边，则  $n$  阶方阵  $A(G)=(a_{ij})$  称为  $G$

的邻接矩阵。其中：

$$a_{ij} = \begin{cases} 1, <v_i, v_j> \in E \\ 0, <v_i, v_j> \notin E \end{cases} \quad (3-1)$$

从邻接矩阵的定义可知，有向图与其相互映射，已知二者中任意一个就可以推导出与之对应的另外一个<sup>[57]</sup>。

本课题通过定义函数 `MyMatrix getCompleteMatrix()` 来求取多信号流图相应的邻接矩阵。求取多信号流图邻接矩阵的流程图如图 3-1 所示。

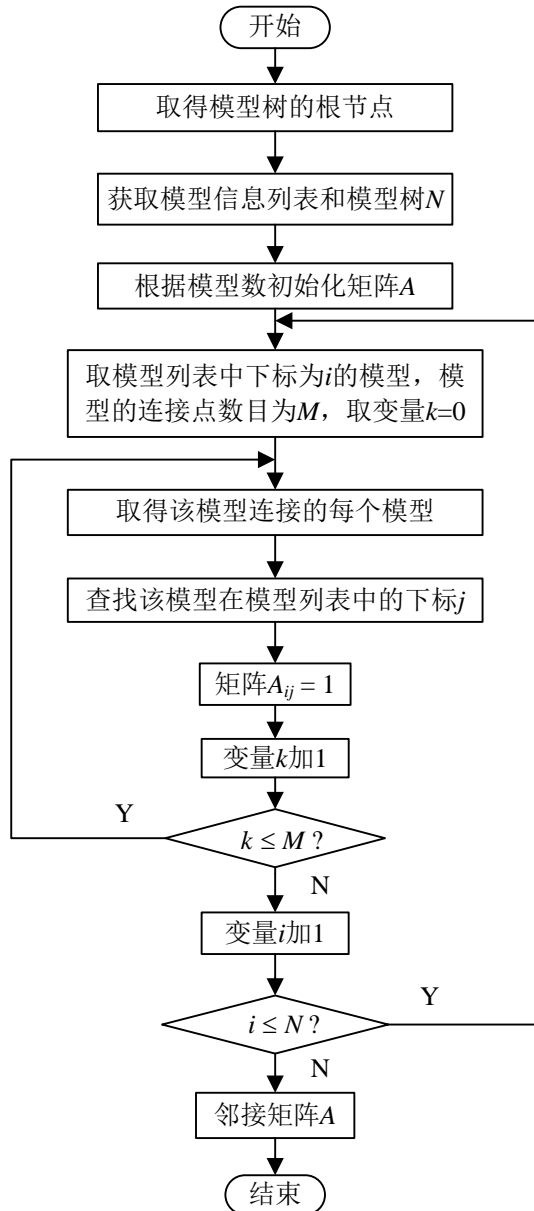


图 3-1 多信号流图模型中求取邻接矩阵的算法流程图

求取多信号流图邻接矩阵的具体步骤如下：

(1) 遍历 Visio 文档中的元素，初始化 MyData 数据结构，将元素间的连接关系保存到元素的 Element 结构中；

(2) 调用 MyMatrix getOriginalMatrix() 函数，检查每个元素 Element 结构中的后续节点是否为空，如果不为空，则邻接矩阵中相应的值赋为 1，并遍历其后续节点。如果 Element 结构中的后续节点为空，邻接矩阵中相应的值赋为 0；

(3) 分析完模型列表中所有模型，就能够生成多信号流图模型的邻接矩阵 A。

### 3.2.2 可达性矩阵的定义与求取

有向图  $G = \langle V, E \rangle$  中，有  $n$  个结点  $V = \{v_1, v_2, \dots, v_n\}$ ， $m$  条边  $E = \{\langle v_i, v_j \rangle | v_i \in V, v_j \in V\}$ ，则称  $n$  阶方阵  $P(G) = (p_{ij})$  为有向图  $G$  的可达性矩阵<sup>[58]</sup>。其中：

$$p_{ij} = \begin{cases} 1, & \text{从 } v_i \text{ 到 } v_j \text{ 存在通路,} \\ 0, & \text{从 } v_i \text{ 到 } v_j \text{ 不存在任何通路.} \end{cases} \quad (3-2)$$

由上述定义可知，可达性矩阵表示的是有向图中各结点能否可达或者是有向图中是否有回路。

在讨论多信号流图的可达性时，不需要考虑任意两个结点之间通路的长度和条数，只研究是否存在通路。

根据邻接矩阵求取可达性矩阵的方法：

(1) 一般算法

定理 1：在一个具有  $n$  个结点的有向图中，若从  $v_i$  到  $v_j$  存在一条通路，则从  $v_i$  到  $v_j$  ( $i \neq j$ ) 存在一条长度不大于  $n-1$  的基本通路。

定理 2：设  $A(G)$  是有向图  $G$  的邻接矩阵，则矩阵  $A(G)^l$  中的第  $i$  行、第  $j$  列的元素  $a_{ij}^l$  表示的是有向图  $G$  中连接结点  $v_i$  和  $v_j$  长度为  $l$  的通路的数目。

由上述两个定理可得到由邻接矩阵求取可达性矩阵的一般方法：

(a) 对有向图  $G$  的邻接矩阵  $A(G)$  进行如下变换：

$$B(G) = A(G) + (A(G))^2 + \dots + (A(G))^n = \sum_{k=1}^n (A(G))^k \quad (3-3)$$

其中， $B(G)$  是表示有向图  $G$  中任意两结点之间通路数目的矩阵， $(A(G))^k$  是  $A(G)$  的  $k$  次幂矩阵。

(b) 将矩阵  $B(G)$  中的全部非零元素置换为 1，为零的元素保持不变，就得到有向图  $G$  的可达性矩阵  $P(G) = (p_{ij})$ ，其中：

$$p_{ij} = \begin{cases} 1, & b_{ij} \neq 0 \\ 0, & b_{ij} = 0 \end{cases}, \quad B(G) = (b_{ij}) \quad (3-4)$$

一般性算法求取可达性矩阵的计算比较麻烦，可以在其基础上加以改进，给

出布尔矩阵算法。

### (2) 布尔矩阵算法

在求取有向图  $G$  的可达性矩阵  $P(G)$  的过程中，只关心任意两个结点之间是否存在通路，并不关心两个结点之间的通路数目。因此，可以将矩阵  $A(G)$ ， $A(G)^2, \dots, A(G)^k$ ，分别改写为布尔矩阵  $A(G)^{(1)}, A(G)^{(2)}, \dots, A(G)^{(k)}$ 。其中， $A^{(k)}$  表示的是在布尔运算意义下的幂矩阵。

当有向图中的结点较多时，用布尔矩阵算法求取可达性矩阵仍较为麻烦，因此本课题采用下面讲述的 Warshall 算法。

### (3) Warshall 算法

在求解可达性矩阵  $P(G)$  的过程中，若是将有向图  $G$  的邻接矩阵  $A(G)$  作为结点集  $V$  上的关系  $R$  的关系矩阵，则求取可达性矩阵  $P(G)$  就相当于求取邻接矩阵  $A(G)$  的传递闭包  $A^+$ 。因此，也可用求传递闭包的 Warshall 算法求取可达性矩阵  $P(G)$  [59]。

利用 Warshall 算法求取邻接矩阵  $A(G)$  的可达性矩阵  $P(G)$  的计算步骤如下：

(a) 取  $i=1, 0 \leq i \leq 1$ ;

(b) 取  $j=1, 0 \leq j \leq 1$ ;

(c) 如果邻接矩阵  $A_{ij}=1$ ，即  $A$  的第  $j$  行、第  $i$  列的元素为 1，则对每一个  $k(k=1, 2, \dots, n)$  进行如下运算： $A_{jk} = A_{jk} \cup A_{ik}$ ，将  $A$  的第  $j$  行、第  $k$  列的元素与第  $i$  行、第  $k$  列的元素进行逻辑加运算，并把结果赋值给  $A$  的第  $j$  行、第  $k$  列的元素。

(d)  $j++$ ，若  $j \leq n$ ，则转到(c)，否则向下执行；

(e)  $i++$ ，若  $i \leq n$ ，则转到(b)，否则停止。

本课题利用 Warshall 算法求取模型的可达性矩阵，其流程图如图 3-2 所示。

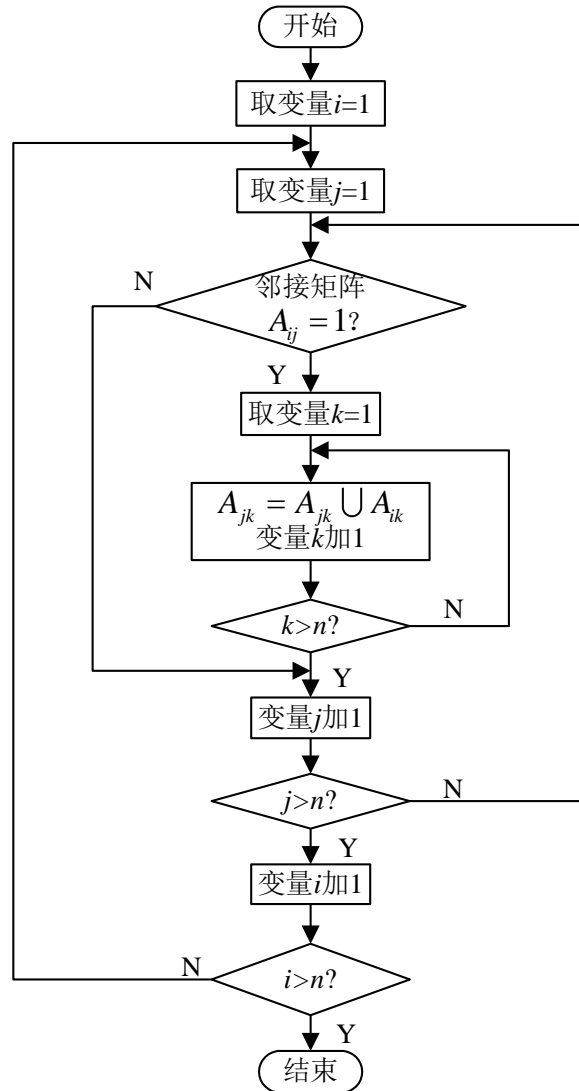


图 3-2 求取可达性矩阵的 Warshall 算法流程图

### 3.2.3 完全故障相关矩阵的生成

求出的可达性矩阵  $P(G)$  是一个  $n$  阶方阵，表示的是有向图  $G$  中各个结点之间的相互连通关系。具体到多信号流图模型，表示的是各个模块及测试点之间的相互连通关系。通过提取各个模块所在的行和各个测试所在的列，就可以得到有向图  $G$  的完全故障相关矩阵。

下面结合图 2-1 所示的雷达发射机具体阐述完全故障相关矩阵的生成过程。

雷达发射机系统的多信号流图模型要素详见表 2-1。按照 3.2.1 讲述的求取多信号流图模型邻接矩阵的方法，求得雷达发射机的多信号流图  $G$  的邻接矩阵  $A(G)$ ，如表 3-1 所示。然后，按照 3.2.2 讲述的方法，由雷达发射机的多信号流图的邻接矩阵  $A(G)$  求取多信号流图的可达性矩阵  $P(G)$ ，如表 3-2 所示。

表 3-1 雷达发射机系统多信号流图的邻接矩阵  $A(G)$

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$	$c_{11}$	$c_{12}$	$TP_1$	$TP_2$	$TP_3$	$TP_4$	$TP_5$	$TP_6$	$TP_7$	$TP_8$	$TP_9$	$TP_{10}$	$TP_{11}$	$TP_{12}$	$TP_{13}$
$c_1$	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
$c_2$	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$c_3$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
$c_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
$c_5$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
$c_6$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
$c_7$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
$c_8$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
$c_9$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
$c_{10}$	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$c_{11}$	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
$c_{12}$	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$TP_1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_6$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_7$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_8$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_9$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{10}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{11}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{12}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{13}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 3-2 雷达发射机系统多信号流图的可达性矩阵  $P(G)$

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$	$c_{11}$	$c_{12}$	$TP_1$	$TP_2$	$TP_3$	$TP_4$	$TP_5$	$TP_6$	$TP_7$	$TP_8$	$TP_9$	$TP_{10}$	$TP_{11}$	$TP_{12}$	$TP_{13}$
$c_1$	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
$c_2$	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
$c_3$	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
$c_4$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
$c_5$	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
$c_6$	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
$c_7$	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0
$c_8$	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0
$c_9$	0	1	0	1	1	1	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0
$c_{10}$	0	1	0	1	0	0	1	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1	1	1	1
$c_{11}$	0	1	0	1	0	0	1	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1	1	1	1
$c_{12}$	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1
$TP_1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_6$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_7$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_8$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_9$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{10}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{11}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{12}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$TP_{13}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



多信号流图模型的每个测试点有若干个测试。如果模块与测试点之间具有可达性，则模块与附属于测试点的测试间也具有可达性。这也意味着测试点和测试同模块间的可达性等价。以可达性矩阵  $P(G)$  中的  $c_1, c_2, \dots, c_l$  为行，以  $TP_1, TP_2, \dots, TP_{13}$  为列，并根据测试点  $TP_1, TP_2, \dots, TP_{13}$  与测试的包含关系(见表 3-3)，选取  $P(G)$  中对应的元素进行列扩展就生成雷达发射机的完全故障相关矩阵  $D(G)$  如表 3-4 所示。

表 3-3 测试点与测试的对应关系

测试	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_{21}$	$t_9$	$t_{10}$
测试点	$TP_1$	$TP_1$	$TP_2$	$TP_2$	$TP_2$	$TP_2$	$TP_3$	$TP_4$	$TP_4$	$TP_5$	$TP_5$
测试	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$	$t_{22}$
测试点	$TP_6$	$TP_7$	$TP_8$	$TP_9$	$TP_9$	$TP_{10}$	$TP_{10}$	$TP_{10}$	$TP_{11}$	$TP_{12}$	$TP_{13}$

 表 3-4 雷达发射机系统完全故障测试相关矩阵  $D(G)$ 

$C \setminus T$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$	$t_{21}$	$t_{22}$
$c_1(G)$	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_2(G)$	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_3(G)$	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_4(G)$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_5(G)$	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0
$c_6(G)$	0	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
$c_7(G)$	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0
$c_8(G)$	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
$c_9(G)$	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$c_{10}(G)$	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1	1
$c_{11}(G)$	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1	1
$c_{12}(G)$	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1

### 3.3 功能故障相关矩阵的生成

#### 3.3.1 功能故障相关矩阵的定义

在测试节点与故障节点具有可达性时，比较故障模块信号与测试集的信号是否存在公共信号，如果存在，则该故障节点的行与该测试节点的列对应元素为 1，否则为 0，得到功能故障相关矩阵  $D(F)$ ：

$$\left. \begin{array}{l} d_{ij}(G) = 1 \\ SC(c_i) \cap ST(t_j) = \emptyset \end{array} \right\} \Rightarrow d_{ij}(F) = 1 \quad (3-5)$$

其中， $c_i$  表示故障模块， $TP_k$  表示测试点， $t_j$  表示测试点测试， $SC(c_i)$  表示与故障模块  $c_i$  相关的信号， $ST(t_j)$  表示测试  $t_j$  中所包含的信号。

### 3.3.2 功能故障相关矩阵的生成

根据上述定义，已知完全故障相关矩阵  $D(G)$  的前提下，求取多信号流图模型功能故障相关矩阵的算法流程如图 3-3 所示：

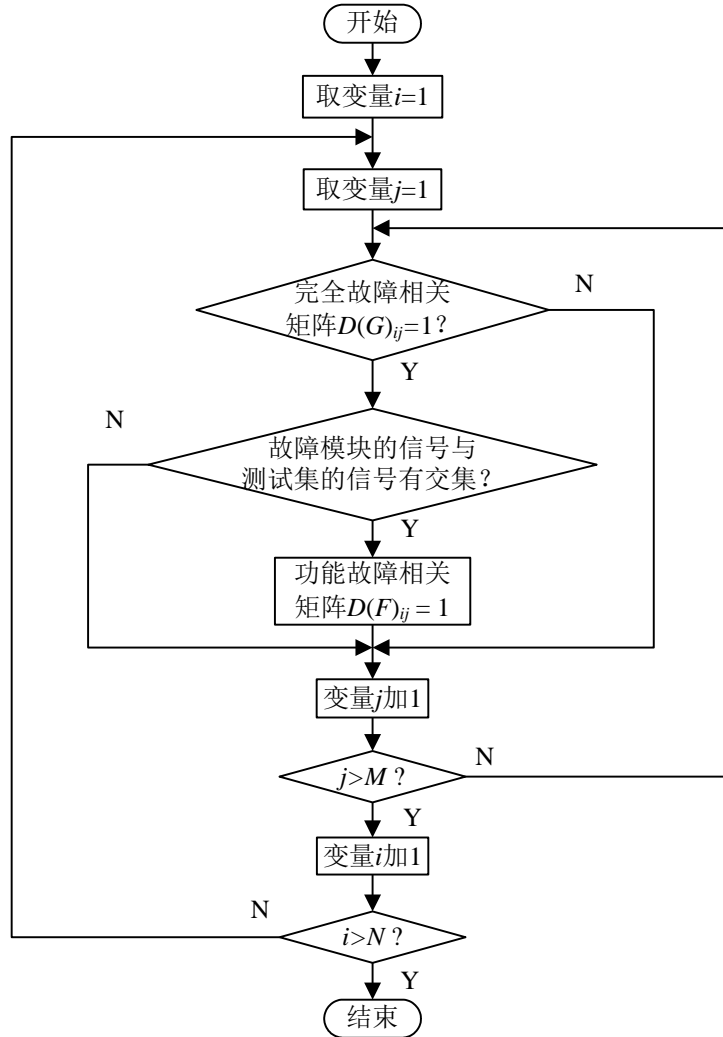


图 3-3 求取多信号流图模型功能故障相关矩阵算法流程图

求取功能故障相关矩阵的具体步骤如下：

- (1) 列出故障模块列表  $SC[N]$  和测试集列表  $ST[M]$ ；
- (2) 设置变量  $i, 0 \leq i \leq N$ ， $N$  为故障模块数目，取  $i=0$ ；
- (3) 设置变量  $j, 0 \leq j \leq M$ ， $M$  为测试集数目，取  $j=0$ ；

(a) 根据完全故障相关矩阵  $D(G)_{ij}$ ，即  $D(G)$  的第  $i$  行、第  $j$  列，如果  $D(G)_{ij} = 1$ ，则对故障节点信号  $SC[i]$  与测试集的信号  $ST[j]$  对比。如果存在公共信号，则  $D(F)_{ij} = 1$ ，否则向下执行；

(b) 变量  $j$  加 1，判断  $j$  是否大于测试集的数目  $M$ ，如果小于  $M$ ，重复步骤

(a), 否则向下执行;

(4) 变量  $i$  加 1, 判断  $i$  是否大于故障模块的数目  $N$ , 如果小于  $N$ , 重复步骤 (2), 否则停止, 生成功能故障相关矩阵。

按照流程图, 运用上述算法, 求得图 2-1 所示的雷达发射机的多信号流图模型的功能故障相关矩阵  $D(F)$ 为:

表 3-5 雷达发射机系统功能故障测试相关矩阵  $D(F)$

$C \setminus T$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$	$t_{21}$	$t_{22}$
$c_1(F)$	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_2(F)$	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_3(F)$	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_4(F)$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_5(F)$	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
$c_6(F)$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$c_7(F)$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
$c_8(F)$	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
$c_9(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
$c_{10}(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
$c_{11}(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
$c_{12}(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

### 3.4 故障-测试相关矩阵的生成

前面两小节分别给出了求取完全故障相关矩阵  $D(G)$ 和功能故障相关矩阵  $D(F)$ 的方法。在进行测试性分析时, 根据模块的实际属性设置将生成的完全故障矩阵和功能故障矩阵组合在一起, 就得到了故障-测试相关矩阵。

图 2-1 所示的雷达发射机的多信号流图模型中, 按照各模块属性的设置, 将同一个故障模块所对应的两种故障属性放在一起就得到图 2-1 所示雷达发射机的多信号流图模型的相关矩阵, 如表 3-6 所示:

表 3-6 雷达发射机系统多信号流图模型的故障-测试相关矩阵

$C \setminus T$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$	$t_{21}$	$t_{22}$
$c_1(G)$	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_1(F)$	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_2(G)$	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_2(F)$	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_3(G)$	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_3(F)$	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$c_4(F)$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$c_5(F)$	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
$c_6(F)$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$c_7(F)$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
$c_8(G)$	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
$c_8(F)$	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
$c_9(G)$	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

续表 3-6 雷达发射机系统多信号流图模型的故障-测试相关矩阵

$c_9(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
$c_{10}(G)$	0	0	1	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1
$c_{11}(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
$c_{12}(G)$	0	0	1	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1
$c_{12}(F)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

### 3.5 本章小结

本章详细阐述了根据多信号流图模型求取故障-测试相关矩阵的方法。首先，通过对 Visio 后台系统文件的提取、分析，求取了模型的邻接矩阵；然后根据邻接矩阵求取了可达性矩阵；接下来将相关矩阵的求取问题分解为完全故障相关矩阵求取和功能故障相关矩阵求取两个方面；最后合成故障测试相关矩阵。结合具体多信号流图模型，对上述过程中几种矩阵的求解算法和流程进行了详细阐述。

## 第4章 基于多信号流程图模型的测试性分析

### 4.1 引言

根据上一章生成的故障-测试相关矩阵，即可开展系统的测试性分析。具体包括静态测试性分析和动态测试性分析。通过静态测试性分析，可以得到系统的故障检测率、故障隔离率、未检测故障、冗余测试和模糊组等测试性指标；通过动态测试性分析，可以得到最优测试序列，用来指导系统的测试与维护。

### 4.2 测试性分析

#### 4.2.1 静态测试性分析

静态测试性分析是指通过对所研究系统故障-测试相关矩阵的分析(统计分析或者布尔“或”运算等)，得到系统的固有测试属性的一种分析方法。静态测试性分析仅由系统的故障测试相关关系决定，不受其他外界因素影响<sup>[28]</sup>。

##### 4.2.1.1 静态测试性分析相关指标

静态测试性分析主要分析指标包括故障检测率、故障隔离率、未检测故障、冗余测试和模糊组等。

各指标具体概念如下<sup>[60]</sup>：

- (1) 故障检测率 (FDR, Fault Detection Rate)：用规定的方法正确检测到的故障数与故障总数之比，用百分数表示。
- (2) 故障隔离率 (FIR, Fault Isolation Rate)：用规定的方法将检测到的故障正确隔离到不大于规定模糊度的故障数与检测到的故障数之比，用百分数表示。
- (3) 未检测故障：不能被已选用的测试所识别的故障源。
- (4) 冗余测试：一组具有相同检测特征的测试，能检测到一组相同的故障源。
- (5) 模糊组：一组具有相同可检测故障特征的故障源，能被同一组测试检测到。

各指标具体计算方法如下：

- (1) 故障检测率的计算公式：

$$FDR = \frac{M}{N} \times 100\% \quad (4-1)$$

其中， $M$ 、 $N$  分别是故障-测试相关矩阵中非零行的行数与总行数。

- (2) 故障隔离率的计算公式：

$$FIR = \frac{m}{n} \times 100\% \quad (4-2)$$

其中,  $m$ 、 $n$  分别是故障-测试相关矩阵中所有故障表征唯一的行数与总行数。

(3) 未检测故障: 如果相关矩阵  $D$  中存在全零行, 则该行所对应的故障即为未检测故障。

(4) 模糊组: 比较相关矩阵  $D$  中各行, 如果有  $F_i = F_j (i \neq j)$ , 则对应的故障是不可区分的, 可作为一个模糊组处理, 并合并为一行。

(5) 冗余测试: 比较相关矩阵  $D$  中各列, 如果有  $T_i = T_j (i \neq j)$ , 则对应的测试互为冗余测试。

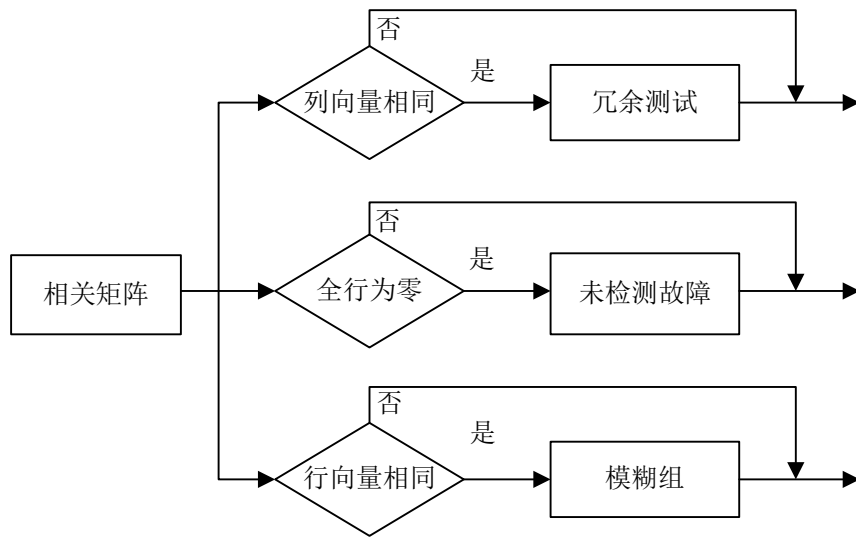


图4-1 静态测试性分析示意图

#### 4.2.1.2 静态测试性分析实例

下面以上一章中的雷达发射机系统为例, 根据其故障-测试相关矩阵(表 3-6)进行静态测试性分析可知:

该相关矩阵有 18 行, 即 18 种可能发生的故障, 没有全零行, 因此没有未检测故障, 同时能够计算故障检测率  $FDR=100\%$ ; 没有相同的行, 因此没有模糊组, 同时能够计算故障隔离率  $FIR=100\%$ ; 该矩阵有 7 组相同的列向量, 即 7 组冗余测试, 分别是  $\{t_1, t_2\}$ 、 $\{t_3, t_4, t_5, t_6\}$ 、 $\{t_8, t_{21}\}$ 、 $\{t_9, t_{10}\}$ 、 $\{t_{14}, t_{15}\}$ 、 $\{t_{16}, t_{17}, t_{18}\}$  和  $\{t_{19}, t_{20}\}$ 。根据静态测试性分析结果就可以调整系统的多信号流图模型, 将冗余的测试删减掉, 节约了测试时间, 减少了测试费用, 降低了故障诊断、排除难度。

#### 4.2.2 动态测试性分析

动态测试性分析除了受所分析系统的故障-测试相关矩阵的影响, 还要考虑故

障率、测试代价、测试需要的时间等。因此，其分析结果具有一定的不确定性、动态性。系统的动态测试性分析主要参数包括加权故障检测率、加权故障隔离率和测试序列等。

除了静态测试性分析中根据相关矩阵中的行向量来计算检测率和隔离率外，还可以根据每种故障源实际的故障率来计算检测率和隔离率。由于通过这种方式计算得到的故障检测率和隔离率与故障率有着密切的联系，也为了与静态分析中的故障检测率和隔离率加以区分，因此称之为加权故障检测率和加权故障隔离率。

(1) 加权故障检测率 (Weighted FDR) 的计算公式为：

$$FDR_w = \frac{\lambda_D}{\lambda} = \frac{\sum_{i=1}^n \lambda_{Di}}{\sum_{i=1}^m \lambda_i} \times 100\% \quad (4-3)$$

其中， $\lambda$  表示系统中可能发生故障(无论其能否被检测到)的所有故障源的故障率之和； $\lambda_D$  表示系统中能够被检测到的故障率之和； $\lambda_{Di}$  表示能够被检测到的每个故障源的故障率(故障测试相关矩阵中，第  $i$  个非零行的故障率)； $\lambda_i$  表示各个故障源的故障率； $n$  表示能够被检测到的故障源的个数(相关矩阵中非零行的行数)； $m$  表示系统可能发生故障的故障源的总数(相关矩阵的总行数)。

(2) 加权的故障隔离率 (Weighted FIR) 的计算公式为：

$$FIR_w = \begin{cases} \frac{\sum_{k=1}^M \lambda_k}{\sum_{i=1}^N \lambda_i} \times 100\%, & \lambda_0 = 0 \\ \lambda_0 + (1 - \lambda_0) \frac{\sum_{k=1}^M \lambda_k}{\sum_{i=1}^N \lambda_i} \times 100\%, & \lambda_0 \neq 0 \end{cases} \quad (4-4)$$

其中， $\lambda_0$  是系统的完好系数(系统无故障状态的概率)， $\lambda_i$  表示各个故障源的故障率； $\lambda_k$  是每个可以隔离的故障源的故障率(相关矩阵中，第  $k$  个无重复行所对应的概率)， $M$  表示能够被隔离出的故障源的个数(相关矩阵中无重复行的行数)， $N$  表示系统可能发生故障的故障源的总数(相关矩阵的总行数)。

### 4.3 测试选择问题研究

在完成系统的测试性建模、得到系统的故障-测试相关矩阵之后，需要进行系统测试方案的构建，寻找最佳测试方案，也就是在系统所有能够采用的测试中，

选取能够达到系统测试性指标要求(如故障检测率、故障隔离率、测试代价等)的测试序列。

测试性分析研究的对象无论是功能还是结构都越来越复杂，测试选择问题求解的难度越来越大，而遗传算法在求解类似的组合优化问题上已经有成功的应用先例，取得了良好效果。因此，本课题对测试性分析的测试选择问题进行了研究，提出了一种利用遗传算法的测试选择方法，并通过实例验证了算法的有效性。

#### 4.3.1 遗传算法的模型构建

遗传算法 (Genetic Algorithms, GA)是一种通过模拟自然界中生物进化的过程来求解最优化问题的自适应智能算法<sup>[61]</sup>。该算法沿续了生物进化过程，首先初始化种群个体，按要求选择合适的适应度函数计算每一个种群个体的适应度值，根据其适应度值选择其中的优势个体，同时进行交叉、变异操作得到新的子代种群，不断重复上述计算适应度值、选择、交叉、变异操作过程，直到子代种群满足系统的精度要求<sup>[62]</sup>，其程序流程如图 4-2 所示。

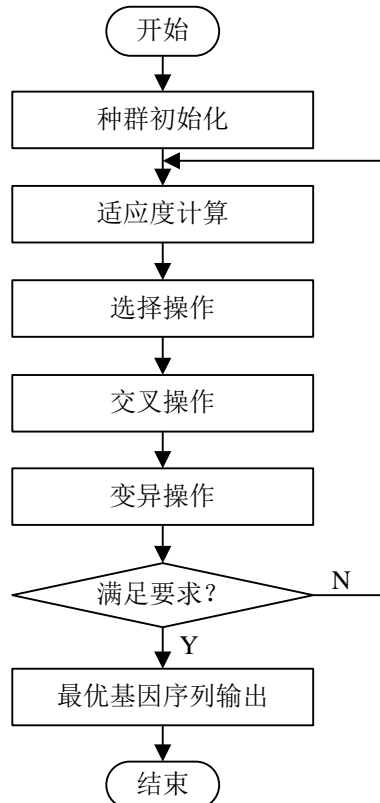


图 4-2 遗传算法程序流程图

##### 4.3.1.1 种群初始化

种群的初始化需要确定种群规模  $N$ 、交叉概率  $P_c$ 、变异概率  $P_m$ 、编码个体



基因和终止进化条件等。种群规模  $N$  应合理选择, 过小的种群规模会降低种群的进化速度, 难于达到最优解; 但是过大的种群规模又会增加程序的计算量, 种群的进化速度也会较慢, 所以种群规模一般控制在 20~100 内, 本课题开发程序中选取  $N=50$ 。种群初始化是为了随机生成种群个体的基因序列, 在种群范围内根据量化精度对解空间进行编码, 利用编码值表示种群个体的基因位, 进而形成一个基因序列。其中编码的形式有二进制编码、格雷码编码和浮点数编码等, 选用不同编码的形式将直接影响 GA 算法的性能。

(1) 二进制编码。它是一种最早被使用的编码形式, 采用小字符编码原则, 容易实现交叉、变异等操作, 对新生成的染色体校验方便, 可以通过模式定理进行理论证明。此外, 它还能够有效地解决高维度、高精度的问题。但是这种编码形式平均编码较长, 占用很大的内存空间, 无法描述一些难于用 0、1 表示的状态。

(2) 格雷码编码。它也是一种二进制编码的演变, 对于两个连续的整数, 格雷码编码能保证编码只相差一位, 从而解决了二进制编码离散化严重的问题。使用格雷码, 算法的局部搜索能力能够显著提高, 交叉、变异操作也容易实现。假如原码为  $A = a_m a_{m-1} \cdots a_2 a_1$ , 其对应的格雷码为  $B = b_m b_{m-1} \cdots b_2 b_1$ , 则它们之间的对应关系如式(4-5)所示:

$$\begin{cases} b_m = a_m \\ b_i = a_{i+1} \oplus a_i \end{cases} \quad i = m-1, m-2, \dots, 1 \quad (4-5)$$

(3) 浮点数编码。它不需要专门的编码和解码过程、算法简单, 在搜索宽范围解空间寻求最优解时有明显的优势, 其计算精度更高, 与实际问题的贴合更加紧密, 容易处理复杂的决策变量约束条件。

综合以上对比分析, 结合本课题开发程序的编码需求, 选择浮点数编码方式, 直接对测试序号进行编码, 完成种群的初始化。

#### 4.3.1.2 适应度函数

为选取到最优基因序列, 找到种群里最符合要求的染色体, 可以根据自然界的进化理论, 模仿“优胜劣汰”的自然进化过程。因此需要合理评价种群个体的优劣, 引入适应度函数来度量种群个体在优化计算中接近最优解的优良程度。所以适应度大的种群个体, 即更接近最优解的个体, 将遗传到子代, 而适应度小的个体将被淘汰掉。设计适应度函数时, 应保证适应度函数是一个非负的单值连续函数, 应尽可能简单, 减少计算量, 降低计算成本。

#### 4.3.1.3 选择操作

选择操作根据种群个体的适应度函数值, 按照一定的规则选出其中的优势个体, 一般适应度高的个体遗传到下一代的概率大, 常用的选择方式有:

(1) 轮盘赌选择。它是一种典型的回放式随机采样方法。每个个体进入下一代的概率就等于它的适应度值与整个种群中个体适应度值和的比例，适应度值越高，被选中的可能性就越大，进入下一代的概率就越大。但是由于统计造成的误差可能会导致最优个体被忽略，因此需要加入最优个体保护方式。

(2) 随机竞争选择。它与轮盘赌选择相似，只是每次按轮盘赌选择两个种群个体，将它们进行竞争比对，选出其中的优势个体，重复比对选择直至选满为止。

(3) 最佳保留选择。它在进行轮盘赌选择的同时，将适应度最高的种群个体完整的复制到子代种群中，保证了最优个体的持续遗传。但是这种选择方式容易陷入局部最优解。

综合以上分析，根据系统测试序号的存储结构，本课题采用随机竞争选择。

#### 4.3.1.4 交叉操作

交叉操作对两个相互配对的染色体按某种方式相互交换部分等位基因，从而产生两个新的个体，它是产生新个体的主要方式。交叉操作的过程如图 4-3 所示，选取两个父代个体，交换它们的某些等位基因得到子代个体。

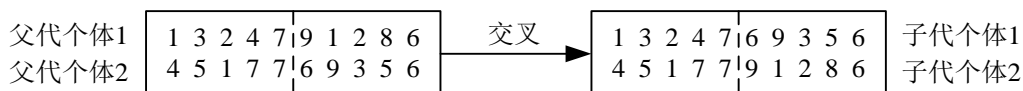


图 4-3 交叉过程示意图

常用的交叉操作包括以下几种：

(1) 单点交叉。其过程是选择两个父代个体，设置一个交叉点，将交叉点后的全部基因进行交换，得到新的个体。

(2) 两点交叉。其过程是选择两个父代个体，设置两个交叉点，将这两个交叉点之间的全部基因进行交换，得到新的子代个体。

(3) 均匀交叉。其过程是将两个父代个体的每一个等位基因以相同的交叉概率进行交换，得到新的子代个体。

(4) 算术交叉。其过程是选择两个父代个体，利用它们的线性组合而产生新的子代个体。

本课题设计遗传算法程序中的交叉操作选用单点交叉。

#### 4.3.1.5 变异操作

变异操作是模仿生物进化过程的变异环节，以较小的概率将父代个体中的某个基因位进行突变，它是产生新个体的辅助方法，可以有效补充由于选择和交叉运算而造成的丢失信息，保证遗传算法的正确性。交叉操作和变异操作共同完成了对解空间的全局搜索和局部搜索，保证遗传算法找到解空间中的最优解。其变异过程如图 4-4 所示，选取一个父代个体，选择某个基因位进行变异。

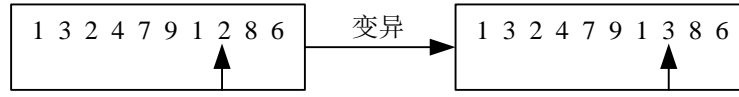


图 4-4 变异过程示意图

常用的变异方法主要有两种：

(1) 基本位变异。对种群个体的某一位或某几位基因值以变异概率进行变异操作。

(2) 均匀变异。它适用于浮点数编码，利用符合某一范围内均匀分布的随机数以变异概率来替换种群个体的原有基因值。均匀变异可以使搜索解可以在解空间内任意变化，增加了种群个体的多样性。

本课题设计遗传算法程序的变异操作采用基本位变异，每次随机选择一个基因进行变异。

### 4.3.2 算法有效性验证

本课题结合雷达发射机系统的相关矩阵建立了遗传算法模型，进行测试的优化选取。该系统要求选出的测试集合能够达到的故障检测率和故障隔离率都不低于 95%。首先，建立一个规模为 50 的种群，设置遗传代数为 100 代，交叉概率为 0.8，变异概率为 0.02。

在染色体编码过程中，采用了实数编码，并且对测试集合中的每一个测试进行编号。每条染色体的长度就是最终测试时需要使用的测试个数，染色体中的每一个基因代表了采用的测试集合中对应的测试。比如，某一条染色体的总长度是 12，其中某位的基因为 32，这表示的是最终选择出的测试集合中总共有 12 个测试，并且该集合中包含第 32 个测试。

选择的适应度函数为：

$$f(T_s) = S(T_s) \left( 1 - \frac{\sum_{t_i \in T_s} c_i / h_i}{\sum_{t_j \in T} c_j / h_j} \right) \quad (4-6)$$

其中， $c_i$  表示测试  $t_i$  的测试代价；

(1)  $h_i$  是测试  $t_i$  的评估函数值，用来表示  $t_i$  的优劣程度，具体形式为：

$$h(t_i) = \frac{d(t_i)}{k(t_i)} \quad (4-7)$$

其中， $d(t_i)$  为测试  $t_i$  能检测到故障集中的故障数； $k(t_i)$  表示  $t_i$  所能检测的故障集中所有故障的最小可测度。某故障的可测度表示该故障能被测试集中测试检测到的个数。

(2)  $S(T_s)$  是选择系数:

$$S(T_s) = \begin{cases} 1, & D \geq D', I \geq I' \\ \left( \frac{DI}{D'I'} \right)^2, & \text{其他} \end{cases} \quad (4-8)$$

其中,  $D$ 、 $I$  分别表示测试集合  $T_s$  对应的系统故障检测率和隔离率,  $D'$ 、 $I'$  分别是要求的故障检测率和隔离率。

在计算得到每一条染色体的适应度函数值之后, 首先, 选择出 3 个适应度函数值最高的个体来同化 3 个适应度最低的个体。然后, 生成一个 0 到 1 之间的随机数并将该随机数与交叉率  $P_c$  进行比较, 如果这个数大于交叉率则对种群进行一次交叉操作, 进而再产生一个 0 到 1 之间的随机数与变异概率  $P_m$  进行比较, 如果大于  $P_m$  则对种群进行一次变异操作。当所有步骤都进行完成以后, 记录下这一代种群当中最优个体及其适应度值。

下面通过求取雷达发射机系统的测试集合, 完成基于遗传算法的测试选择算法有效性的检验。其故障-测试相关矩阵如表 4-1 所示。

按照根据静态测试性分析结果将冗余测试所对应的列删除, 完成对相关矩阵的化简, 以便动态测试性分析使用。

表 4-1 静态测试性分析后的雷达发射机系统的故障-测试相关矩阵

$CT$	$t_1$	$t_3$	$t_7$	$t_8$	$t_9$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{16}$	$t_{19}$	$t_{22}$
$c_1(G)$	1	1	0	1	0	0	0	0	0	0	0	0
$c_1(F)$	1	0	0	0	0	0	0	0	0	0	0	0
$c_2(G)$	0	1	0	1	0	0	0	0	0	0	0	0
$c_2(F)$	0	1	0	0	0	0	0	0	0	0	0	0
$c_3(G)$	0	1	1	1	0	0	0	0	0	0	0	0
$c_3(F)$	0	0	1	0	0	0	0	0	0	0	0	0
$c_4(F)$	0	0	0	1	0	0	0	0	0	0	0	0
$c_5(F)$	0	0	0	0	1	0	0	0	0	0	0	0
$c_6(F)$	0	0	0	0	0	1	0	0	0	0	0	0
$c_7(F)$	0	0	0	0	0	0	0	1	0	0	0	0
$c_8(G)$	0	1	0	1	0	0	1	0	0	0	0	0
$c_8(F)$	0	0	0	0	0	0	1	0	0	0	0	0
$c_9(G)$	0	1	0	1	1	1	1	1	1	1	1	1
$c_9(F)$	0	0	0	0	0	0	0	0	1	0	0	0
$c_{10}(G)$	0	1	0	1	0	0	0	1	0	1	1	1
$c_{11}(F)$	0	0	0	0	0	0	0	0	0	0	1	0
$c_{12}(G)$	0	1	0	1	0	0	0	1	0	0	0	1
$c_{12}(F)$	0	0	0	0	0	0	0	0	0	0	0	1

经过程序多次运行, 给出的结果是{1, 2, 3, 4, 5, 7, 8, 9, 11, 12}, 结合化简后的相关矩阵可知, 选择出的测试集合为{ $t_1, t_3, t_7, t_8, t_9, t_{11}, t_{12}, t_{13}, t_{14}, t_{19}, t_{22}$ }。

表 4-2 遗传算法选择的测试集合

$C \backslash T$	$t_1$	$t_3$	$t_7$	$t_8$	$t_9$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{19}$	$t_{22}$
$c_1(G)$	1	1	0	1	0	0	0	0	0	0
$c_1(F)$	1	0	0	0	0	0	0	0	0	0
$c_2(G)$	0	1	0	1	0	0	0	0	0	0
$c_2(F)$	0	1	0	0	0	0	0	0	0	0
$c_3(G)$	0	1	1	1	0	0	0	0	0	0
$c_3(F)$	0	0	1	0	0	0	0	0	0	0
$c_4(F)$	0	0	0	1	0	0	0	0	0	0
$c_5(F)$	0	0	0	0	1	0	0	0	0	0
$c_6(F)$	0	0	0	0	0	1	0	0	0	0
$c_7(F)$	0	0	0	0	0	0	0	1	0	0
$c_8(G)$	0	1	0	1	0	0	1	0	0	0
$c_8(F)$	0	0	0	0	0	0	1	0	0	0
$c_9(G)$	0	1	0	1	1	1	1	1	1	1
$c_9(F)$	0	0	0	0	0	0	0	0	0	0
$c_{10}(G)$	0	1	0	1	0	0	0	1	1	1
$c_{11}(F)$	0	0	0	0	0	0	0	0	1	0
$c_{12}(G)$	0	1	0	1	0	0	0	1	0	1
$c_{12}(F)$	0	0	0	0	0	0	0	0	0	1

由上表分析可知，没有全零行，所以没有未检测故障，故障检测率为 100%。而且行向量都不相同，所以没有模糊组，故障能够完全隔离，故障隔离率是 100%。故障检测率和隔离率两项分析指标都达到了既定目标，验证了遗传算法求解测试选择性问题的有效性。根据程序运行给出的适应度函数曲线可知，算法在第 24 代左右就能够找到最优解，函数曲线整体上升平稳，具有较好的遗传特性，如图 4-5 所示。

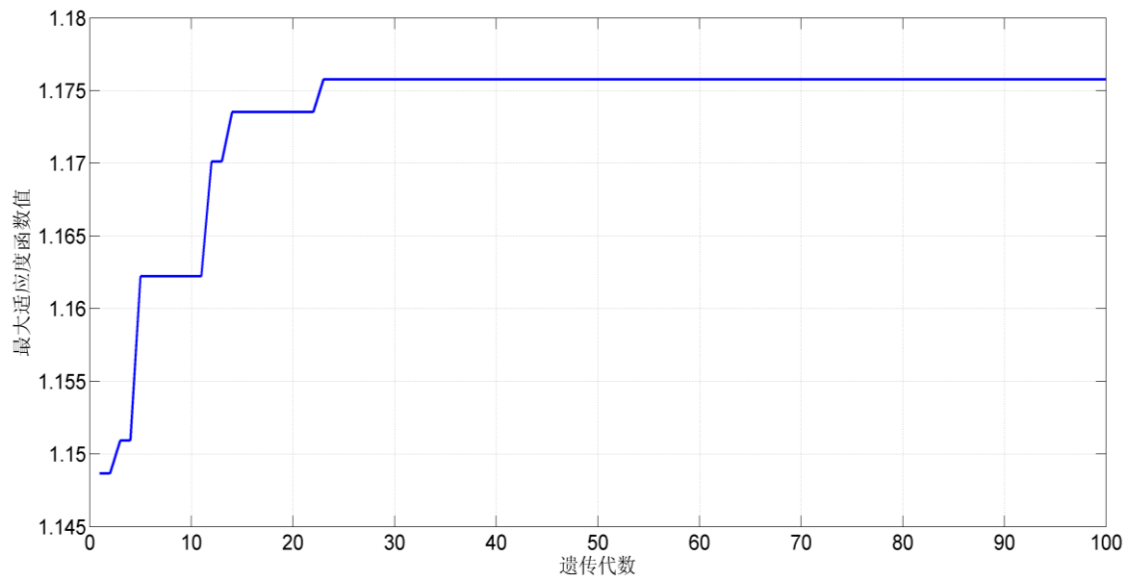


图 4-5 最大适应度函数曲线

## 4.4 测试序列求解方法研究

由遗传算法得出的结果虽然具有较高的故障检测率和隔离率，但仅仅是得到了测试集合，测试之间的先后顺序是未知的、待定的，无法更进一步地去解决问题。本课题采用与或图启发式搜索算法—AO\*算法求解测试序列问题。

### 4.4.1 测试序列求解问题的数学模型

测试序列问题用 $(S, p, T, c, D)$ 之间的相关关系定义。其中 $S = \{s_1, s_2, \dots, s_m\}$ 是测试序列状态的集合， $s_0$ 表示该系统没有发生故障，而 $s_i (1 \leq i \leq m)$ 分别表示该系统可能发生的第 $i$ 个故障； $p = [p(s_0), p(s_1), \dots, p(s_m)]^T$ 是系统各故障状态的故障率集合； $T = \{t_1, t_2, \dots, t_n\}$ 是 $n$ 个可用测试的集合； $c = [c_1, c_2, \dots, c_n]^T$ 是测试代价向量集合，由测试时间、测试成本等因素决定； $D = [d_{ij}]$ 是由故障集和测试集组成的相关矩阵，易知 $d_{0j} (1 \leq j \leq n)$ 值始终为零。

求解最优测试序列即找出一组测试序列，在系统出现任何状态时都能检测到故障源，并且使测试成本最小，其数学模型可以表示为：

$$J = P^T A T = \sum_{i=0}^m \sum_{j=1}^n a_{ij} p(s_i) c_j \quad (4-9)$$

其中， $J$ 为测试代价， $A = [a_{ij}]$ 是 $(m+1) \times n$ 的二元矩阵，如果 $t_i$ 在识别 $s_i$ 的过程中使用到， $a_{ij}$ 的值为1，否则为0。

### 4.4.2 与或图表示形式

测试性分析中使用的测试属于二值测试，即每个测试的结果只有达标和不达标两种可能，因此可以根据测试序列绘制出二叉树形式的故障诊断树，即测试序列与故障诊断树是相互等价的，知道了二者中任意一个就可以推演出另外一个。由于这种等价关系的存在，测试序列的求解问题可以等价于故障诊断树的生成问题。而故障诊断树在形式上又同与或图具有很大的相似性，可以将故障诊断树视为一种广义的与或图。因此本课题采用启发式搜索算法通过对与或图的求解完成对测试序列的求解，问题的转化过程如图4-6所示。

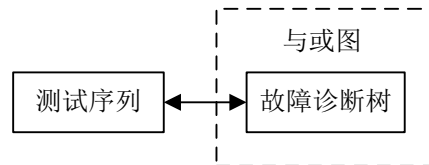


图 4-6 测试序列求解问题转化示意图

与或图包含根节点、非空叶子结点和中间节点。根据与或图同故障诊断树的

对应关系可知，根节点由  $S$  表示，是指需要解决的问题；叶子节点由  $s_i (0 \leq i \leq m)$  表示，是无法再进行分解的问题，即故障；中间节点由  $x$  表示，是一系列子问题，是系统状态的模糊集，包括与节点、或节点和中间叶子节点；中间叶子节点是指无解的节点，它没有子节点。

对模糊集  $x$  进行测试  $t_j$ ，只会出现两种结果即通过或不通过，分别用左、右子节点表示这两种情况。通过测试  $t_j$  是指  $t_j$  不能检测出  $x$  中的子问题，并由该子问题构成左子节点  $x_{jp}$ ；不能通过测试  $t_j$  是指  $t_j$  能够检测到  $x$  中的子问题，并由该子问题构成右子节点  $x_{jf}$ 。

#### 4.4.3 启发式搜索算法

AO\* 算法是一种采用霍夫曼(Huffman)平均编码字长作为启发式评估函数的启发式搜索算法。其中，霍夫曼编码是一种不等字长编码，基本思想是让出现概率大的事件的编码字长尽可能的短，以达到提高编码效率的目的。霍夫曼编码的过程为，对所编码的事件按照概率从小到大进行排列，取概率最小的两个事件进行加和，将得到的新的概率取代原来最小的两个概率，重新排列。重复该过程，直至霍夫曼树构造完成。编码时一般将同级中概率较小的事件作为左分支，默认编码为 0，概率较大的事件最为右分支，默认编码为 1。完成霍夫曼编码后，每个事件都对应着一串二进制编码，将事件的概率与编码长度相乘、累加求和，得到的就是所有编码事件的霍夫曼编码平均字长。

$$w^*(x) = [p(x)]^{-1} \sum_{s_i \in x} w^*(s_i) p(s_i) \quad (4-10)$$

其中， $w^*(x)$  是模糊集  $x$  的平均霍夫曼编码字长， $w^*(s_i)$  是模糊组  $x$  的子集  $s_i$  的霍夫曼编码字长。

根据测试序列确定的故障诊断树和霍夫曼编码生成的霍夫曼树在形式上的相似性，寻找启发式评估函数 HEF。通过霍夫曼编码本身所特有的属性来帮助推导与或图启发式搜索过程中的要用到的启发式评估函数 HEF。

定理：不失一般性地假设测试代价按照升序排序为  $0 \leq c_1 \leq c_2 \leq \dots \leq c_n$ ， $h(x)$  的下限如下：

$$h(x) = HEF = \sum_{j=1}^{w'(x)} c_j + [w^*(x) - w'(x)] \times c_{w'(x)+1} \quad (4-11)$$

其中， $w'(x) = [w^*(x)]$ ，是模糊集  $x$  的霍夫曼平均编码字长的整数部分。

采用公式(4-11)作为评估函数，自上而下进行搜索，当搜索到全部为子节点时，利用公式(4-12)计算故障集实际代价，并用公式(4-13)修正代价，选择最优测

试，自下向上，直到根节点。

$$E_{xj} = c_j + \frac{P(x_{jp})E(x_{jp}) + P(x_{jf})E(x_{jf})}{P(x_{jp}) + P(x_{jf})} \quad (4-12)$$

$$e = \min \{E_{xj}\} \quad (4-13)$$

其中， $E_{xj}$  表示对故障集  $x$  采用测试  $t_j$  分解的实际代价， $x_{jp}$ ， $x_{jf}$  分别为故障集  $x$  通过和未通过测试  $t_j$  的子集， $c_j$  为  $t_j$  测试成本， $P(x_{jp})$ ， $P(x_{jf})$  分别表示  $x_{jp}$ ， $x_{jf}$  的总概率； $E(x_{jp})$ ， $E(x_{jf})$  分别表示故障集  $x_{jp}$ ， $x_{jf}$  实际测试代价。

#### 4.4.3.1 AO\*算法实现步骤

考虑到 MATLAB 强大的矩阵处理能力，在求解最优测试序列问题中有很大优势，本课题采用 MATLAB 来实现 AO\* 算法。其过程主要包括两大步骤：向下评估和向上修正确定最优解，具体来说：

(1) 构建一个元胞数组，存放的元素依次为父节点、左子节点、右子节点、对应测试和评估的测试代价；

(2) 判断故障的集合  $S$  中元素个数是否大于 2，当元素个数大于 2 时继续执行下面的步骤，否则执行步骤(8)；

(3) 对故障的集合  $S$ ，对每个测试按照其测试特性分别进行霍夫曼编码，并评估其测试代价：

$$e = \min \{c_j + HEF(x_{jp}) \times p(x_{jp}) + HEF(x_{jf}) \times p(x_{jf})\} \quad (4-14)$$

其中  $x_{jp}$ ， $x_{jf}$  分别为霍夫曼树的左子节点和右子节点，左子结点表示测试检测不到故障，右子节点表示测试能检测到故障；

(4) 将测试评估代价从小到大进行排列，记录最小的评估值，其对应的测试，相应的父节点、左右子节点；

(5) 将左子结点选作故障的集合  $S$ ，重复步骤(2)到(8)，计算实际的测试代价；

(6) 将右子结点选作故障的集合  $S$ ，重复步骤(2)到(8)，计算实际的测试代价；

(7) 根据左右节点实际测试代价，与元胞数组中测试的评估代价进行比较。如果实际值大于评估值，那么修正评估值，放弃该路径并删除，取次小的测试评估代价对应的测试，重复步骤(4)，否则继续执行；

(8) 继续向上修正评估代价；

(9) 重复(2)到(8)，直到故障的集合  $S$  中的元素个数为 1 或 2 时，选用的测试计算出的实际代价最小，最优诊断树生成，算法结束。

#### 4.4.3.2 AO\*算法实例

为更好的说明 AO\* 算法的具体操作过程，本课题以某系统为例进行详细讲



解。该系统有 5 个故障源  $S = \{s_1, s_2, s_3, s_4, s_5\}$ ， $s_0$  表示系统的无故障状态，5 个故障对应的故障概率  $P = [0.70 \ 0.01 \ 0.02 \ 0.10 \ 0.05 \ 0.12]^T$ ；总共有 6 个能够采用的测试  $T = \{t_1, t_2, t_3, t_4, t_5\}$ ，对应的测试代价  $c = [1 \ 1 \ 1 \ 1 \ 1]$ ；系统的故障-测试相关矩阵为：

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

AO\*算法实现过程如下：

(1) 令  $S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$  为根节点。根据故障-测试相关矩阵，按照 5 个可选测试进行分组，分为 5 组，对每组进行 0/1 霍夫曼编码，计算的测试代价评估值为  $\{2.18, 2.18, 2.22, 2.18, 2.26\}$ 。

以测试  $t_1$  为例， $\{s_0, s_1, s_2\}$  为左子树，编码为 0； $\{s_3, s_4, s_5\}$  为右子树，编码为 1。分别计算左、右子树平均编码字长：

$$HEF_1 = \frac{0.7 \times 1 + 0.02 \times 2 + 0.01 \times 2}{0.7 + 0.02 + 0.01} = 1.04, \quad P_1 = 0.73 \quad (4-15)$$

$$HEF_2 = \frac{0.12 \times 1 + 0.1 \times 2 + 0.05 \times 2}{0.12 + 0.1 + 0.05} = 1.55, \quad P_2 = 0.27 \quad (4-16)$$

评估测试代价：

$$e_1 = c_1 + HEF_1 \times P_1 + HEF_2 \times P_2 = 1 + 1.04 \times 0.73 + 1.55 \times 0.27 = 2.18 \quad (4-17)$$

同理，计算  $t_2, t_3, t_4, t_5$  测试代价的评估值，分别为  $\{2.18, 2.18, 2.22, 2.18, 2.26\}$ 。可以看出  $t_1, t_2, t_4$  的测试代价评估值最小。

(2) 选取测试  $t_1$  继续进行分解。先对左子树选取测试  $t_2, t_3, t_4, t_5$  进行分解，评估代价依次为  $\{1.75, 1.74, 1.74, 1.75\}$ ，则  $t_3, t_4$  测试代价评估值最小。首先选择  $t_3$ ，向下分解时左右子树元素个数分别为 2 和 1，直至分解完成。向上计算实际代价为 1.97，大于评估代价 1.74，放弃该路径，选择  $t_4$  计算实际代价，如果大于 1.74，继续选择  $t_2$ ，否则进行步骤(3)。如果计算结果均大于评估代价，返回(1)，选择一个测试继续分解。

(3) 展开右子树，对右子树选取  $t_2, t_3, t_4, t_5$  分解，测试代价评估值依次为  $\{1.32, 1.32, 1.37, 1.32\}$ ，则  $t_2, t_3, t_5$  评估值最小。向下分解时左右子树元素个数分别为 2 和 1，分解完成。向上计算实际代价为 1.63，该数大于 1.32，放弃该路径，

选择 $t_3$ 计算实际代价，如果大于 1.32，继续选择 $t_5$ ，否则进行步骤(3)。如果计算结果均大于评估代价，返回(1)，选择下一个测试继续分解。

(4) 当计算到故障的集合  $S$  中的元素个数为 1 或 2 时，算法结束，给出最终的测试序列。

上述 AO\*算法分析过程对应的搜索图如图 4-7 所示。

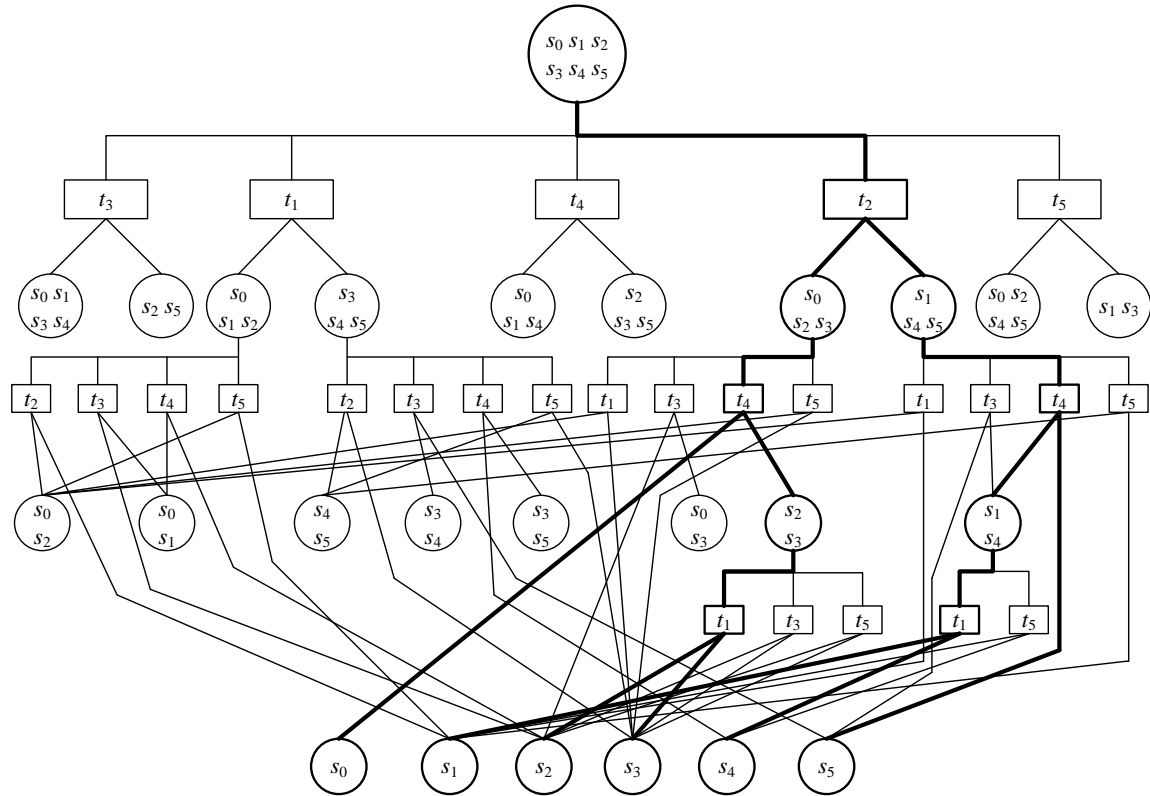


图 4-7 AO\*算法搜索过程示意图

## 4.5 本章小结

本章结合雷达发射机系统的实例，根据其故障-测试相关矩阵，从静态测试性分析和动态测试性分析两个方面对测试性分析进行了详细阐述。静态测试性分析中，介绍了主要参数故障检测率、故障隔离率、模糊组、冗余测试和未检测故障的定义与求解方法。动态测试性分析中，通过引入遗传算法完成了测试集合的求解；通过 AO\*算法完成了测试序列的求解。

## 第5章 软件实现及验证

### 5.1 引言

本课题在前面工作基础上，开发了测试性图形化建模软件，能够完成既定操作，达到了预期目标。基于 MATLAB 强大的数据运算处理能力，用遗传算法进行了测试集合的选取，完成了测试性分析软件的开发。通过与商业化测试性分析软件 TEAMS 的对比，验证了所开发软件的可靠性。

### 5.2 测试性图形化建模软件

#### 5.2.1 图形化建模软件功能介绍

测试性图形化建模软件能够用于系统测试性分析时的多信号流图建模。本课题所开发建模软件的操作界面(详见图2-9)简洁大方，菜单栏功能齐全，工具栏操作快捷方便，并很好的兼容了Visio的右键菜单和微软系统下默认的快捷键操作，用户通过鼠标进行简单的拖拽就能够完成绝大多数的建模操作。建模软件的菜单栏及其下拉菜单如图5-1所示。

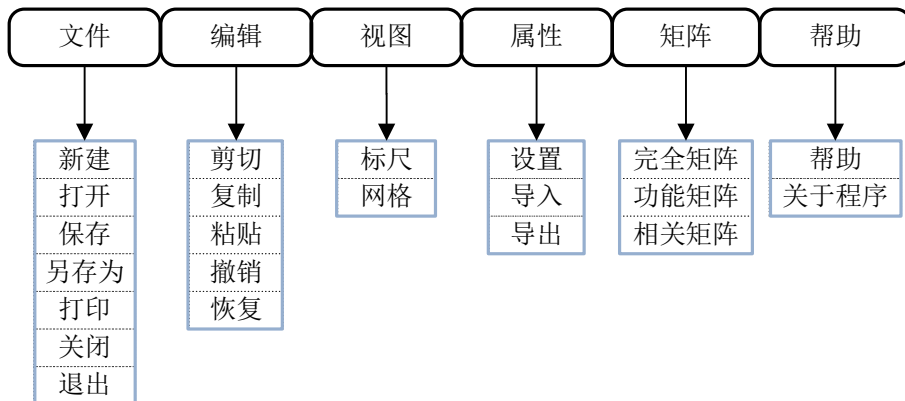


图5-1 测试性图形化建模软件菜单栏

另外，本软件还提供了常用菜单操作的工具栏快捷按钮和右键菜单，极大地方便了日常操作。

总的来说，本课题设计的测试性图形化建模软件能完成以下操作：

#### (1) 图形化建模功能

用户首先要对建模系统的结构和功能进行分析，根据结构和功能的不同将系统划分为不同的组成模块，并确定组成模块的功能信号。然后，从软件提供的建模工具箱中选择相应模块，绘制多信号流图，对故障模块和测试点进行属性设

置，完成其多信号流图建模。

### (2) 故障-测试相关矩阵的自动生成

故障-测试相关矩阵是多信号流图模型的另一种等效表现形式，是对系统进行测试性分析的基础。本课题所开发软件采用一种简洁可靠的算法求取故障-测试矩阵，并能够实现相关矩阵 Excel 表格形式的输出和保存，极大地方便了查看和后续分析。

### (3) 模型图的保存与模块属性的设置、保存

本课题所开发软件能够保存用户绘制的模型图，在必要的时候可以重新进行编辑。此外，本软件还提供模块属性的保存功能，并能够完成文本文档的读取和写入，批量完成属性读写操作，提高了建模效率。

## 5.2.2 图形化建模软件功能验证

下面结合前文使用的雷达发射机系统，具体讲述采用本课题所开发的软件进行测试性建模的步骤，并通过分析结果与理论计算值的对照完成对软件功能的检验。

应用测试性图形化建模软件对 2.1.4 节中讲述的某雷达发射机系统(其结构和测试点分布如图 2-1 所示)进行多信号流图建模的步骤如下：

(1) 打开建模软件，单击菜单栏“新建”，建立某雷达发射机系统的多信号流图模型；

(2) 根据图 2-1 所示的系统结构图，从建模工具箱中拖拽相应的模块(故障模块、测试点模块和连接线等)绘制多信号流图；

(3) 添加故障模块和测试点对应的相关信号，完成其属性设置。所建立的模型图如图 5-2 所示；

(4) 单击“矩阵”，分析模型中各模块的连接关系，生成相关矩阵并保存、展示，如图 5-3 所示。

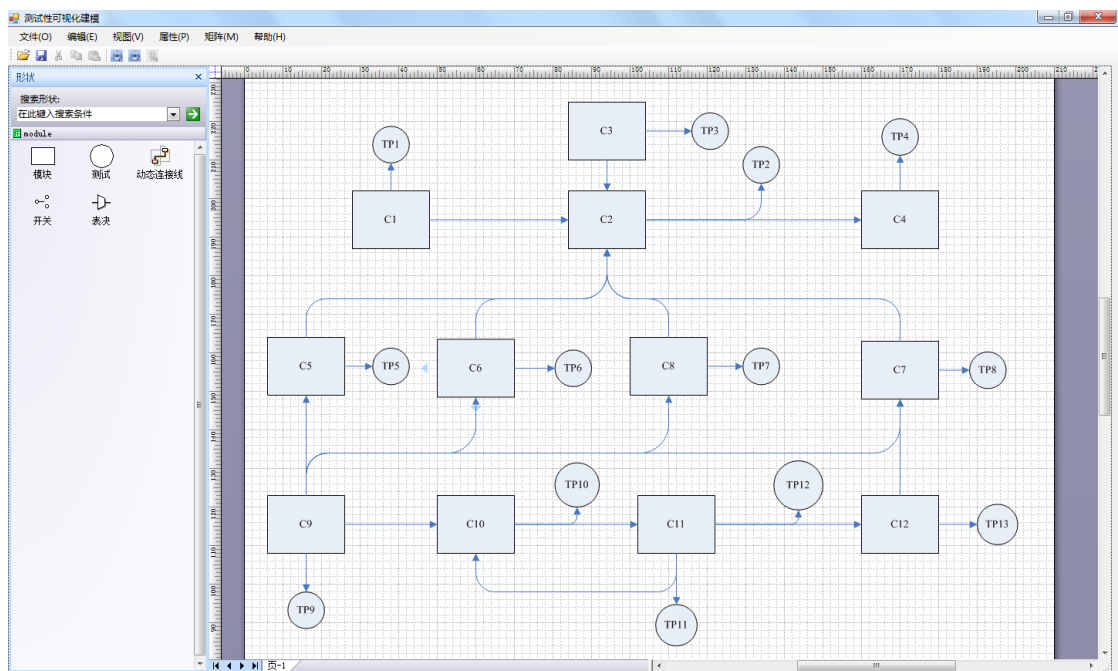


图 5-2 雷达发射机多信号流图模型

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22
2	C1 (G)	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3	C1 (F)	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	C2 (G)	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
5	C2 (F)	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	C3 (G)	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
7	C3 (F)	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	C4 (F)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	C5 (F)	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
10	C6 (F)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
11	C7 (F)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
12	C8 (G)	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
13	C8 (F)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
14	C9 (G)	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	C9 (F)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
16	C10 (G)	0	0	1	1	1	1	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
17	C11 (F)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
18	C12 (G)	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1
19	C12 (F)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

图5-3 雷达发射机多信号流图模型的相关矩阵

对比图 5-3 与表 3-6 中雷达发射机多信号模型故障-测试相关矩阵，能够发现二者完全一致，即软件分析结果与理论计算值完全吻合。从而验证了测试性图形化建模软件相关矩阵生成方法的可行性、功能的可靠性。

## 5.3 测试性分析软件

### 5.3.1 测试性分析软件功能介绍

系统的测试性分析是对其故障-测试相关矩阵的分析。考虑到 MATLAB 在矩阵处理方面的强大功能，本课题采用 MATLAB 作为测试性分析平台。针对测试性分析的静态和动态分析，分别编写了相应的 MATLAB 程序，并采用 GUI 技术，实现了测试性分析软件的可视化。测试性分析软件的输入是图形化建模软件生成的被测系统的故障-测试相关矩阵，能够完成对其 Excel 表格的直接加载；输出结果展示在 GUI 界面中，同时在相关矩阵所在文件夹下保存一份文本格式的分析报告。

将测试性建模软件生成的雷达发射机系统的故障-测试相关矩阵加载到分析软件中，进行测试性分析，得到的结果如图 5-4 所示。

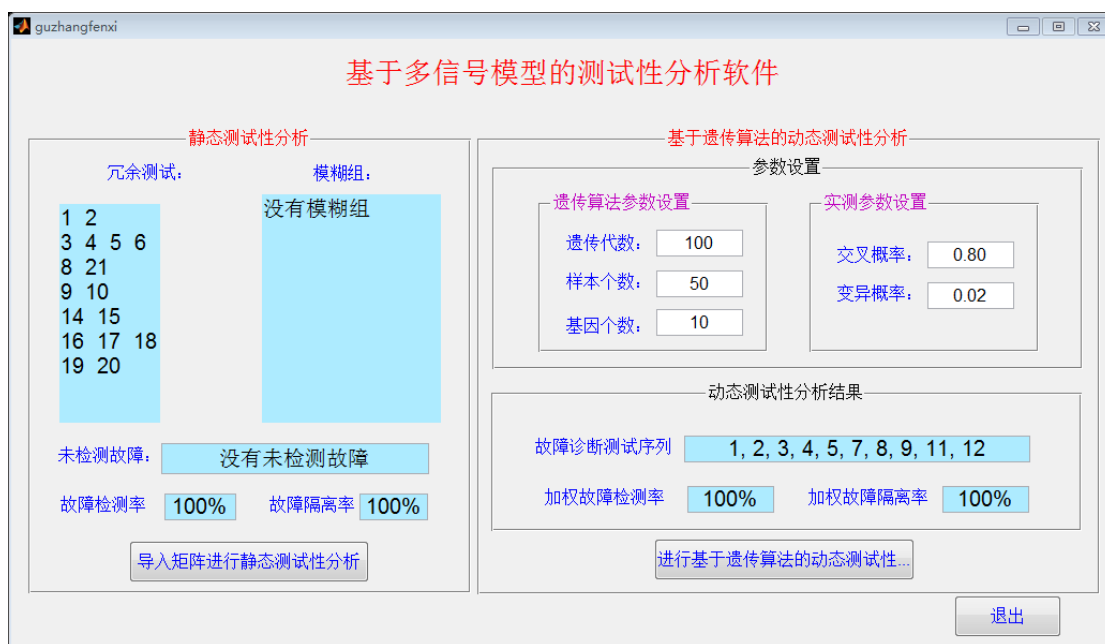


图 5-4 雷达发射机系统的测试性分析结果

测试序列求取模块，即利用 AO\*算法求解最优测试序列同样是在 MATLAB 中完成开发，输入同样是静态测试性分析之后的结果(表 4-1)，对雷达发射机系统进行分析后的结果是 $\{t_4, t_2, t_8, t_{12}, t_7, t_{11}, t_1, t_6, t_9, t_3\}$ ，结合输入矩阵可知实际对应的测试序列为 $\{t_8, t_3, t_{13}, t_{22}, t_{12}, t_{19}, t_1, t_{11}, t_{14}, t_7\}$ 。根据该测试序列，结合相关矩阵就可以绘制出系统的故障诊断树，如图 5-5 所示。根据故障诊断树可知，AO\*算法求解的测试序列，确实做到了故障的 100%检测和隔离，能够用来指导系统的诊断维护。

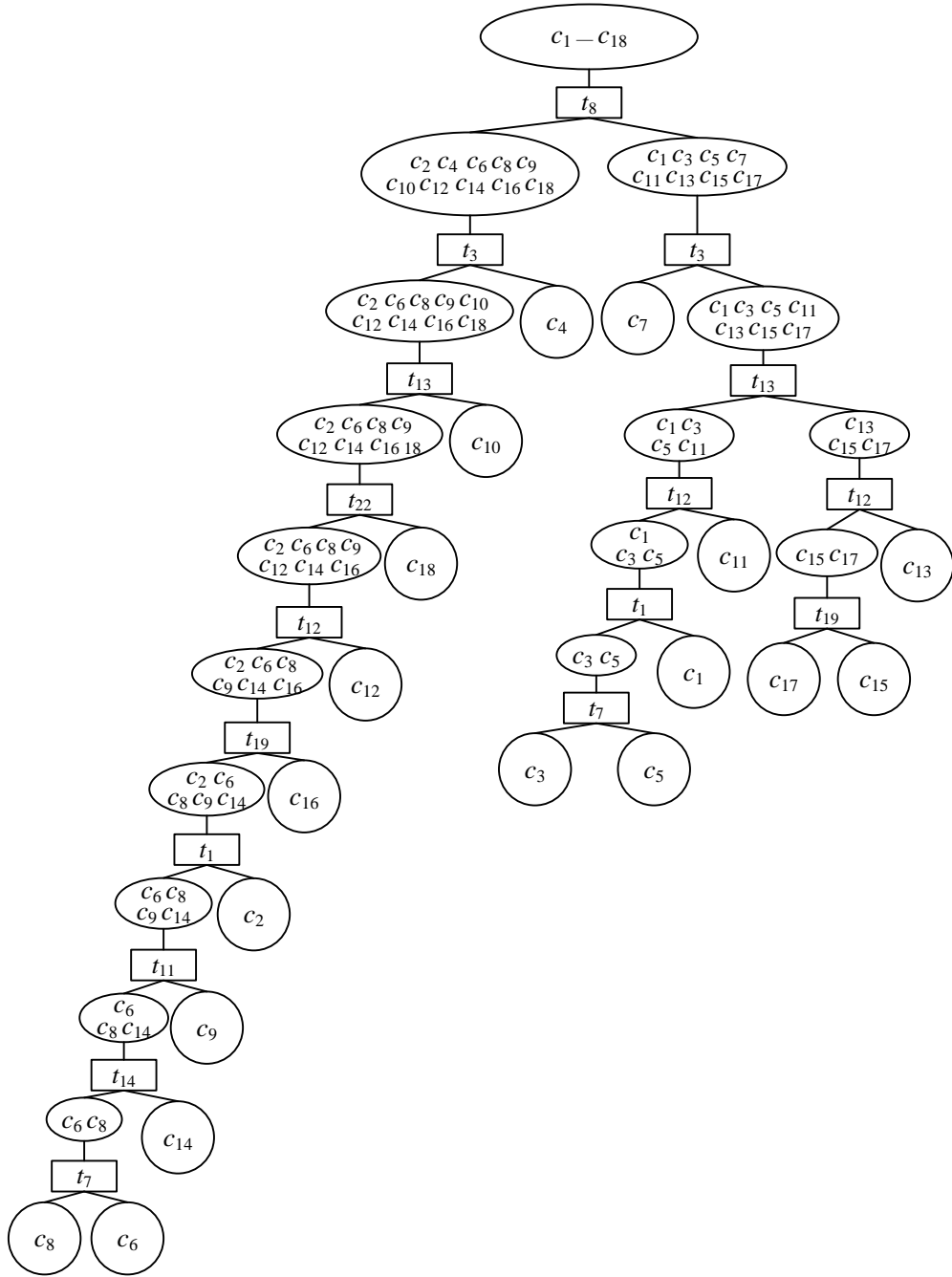


图 5-5 AO\*算法求解的雷达发射机系统故障诊断树

### 5.3.2 测试性分析软件功能验证

为验证本课题所开发测试性分析软件分析结果的可靠性，在 TEAMS 中对雷达发射机系统进行多信号流图建模、分析。建立的模型图如图 5-6 所示，分析结果如图 5-7 所示。

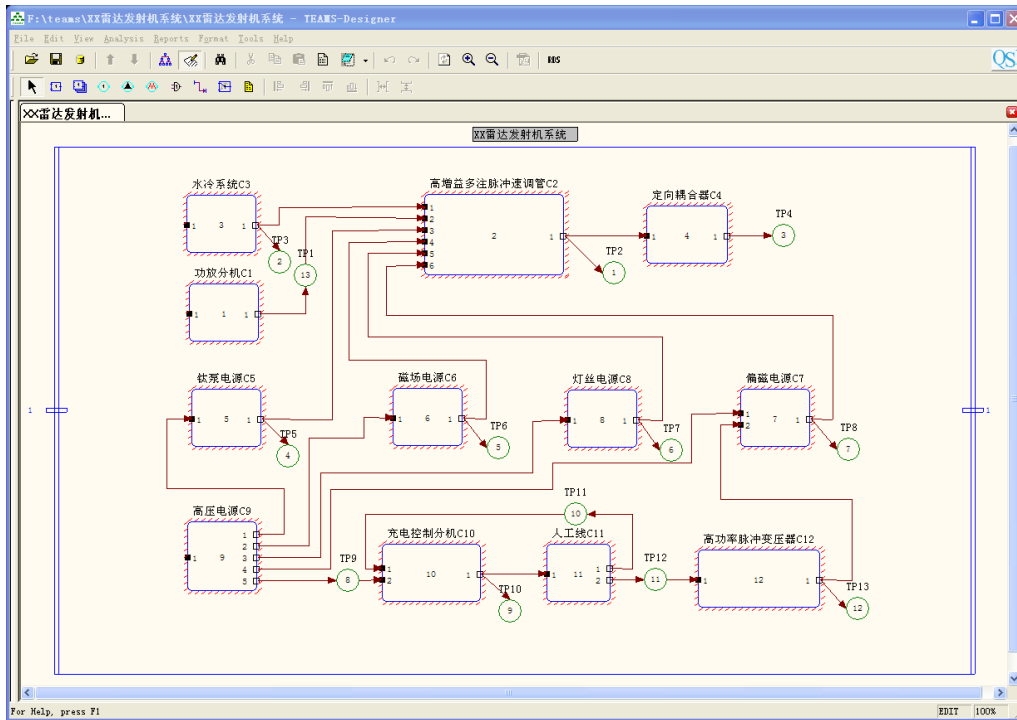


图 5-6 TEAMS 中建立的雷达发射机系统多信号流程图模型

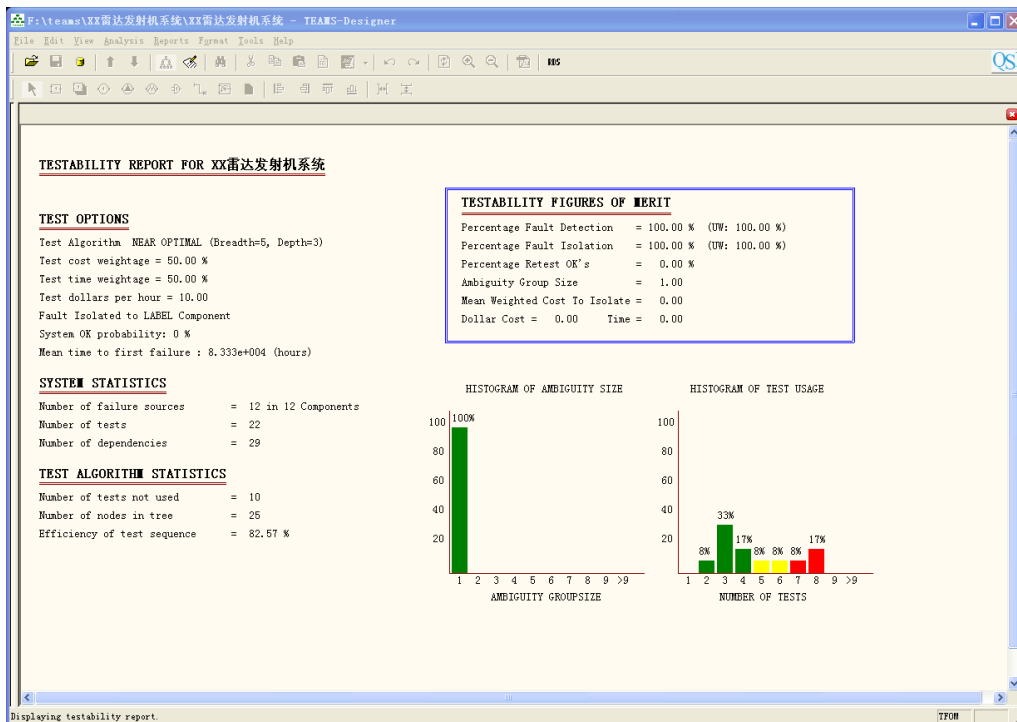


图 5-7 TEAMS 给出的雷达发射机系统测试性分析结果

对照图 5-5 与图 5-7 可以看出，二者关于雷达发射机系统的测试性分析结果一致。从而证明了本课题所开发的测试性分析软件的可靠性。



## 5.4 本章小结

本章介绍了测试性图形化建模软件与分析软件的各项功能，以某雷达发射机系统为例，对建模与分析两部分的功能分别进行了验证，其分析结果与国外 TEAMS 软件基本一致，验证了分析结果的可行性。

## 结 论

本课题在研究多信号流图模型的测试性建模理论基础之上，采用 Visio 绘图控件二次开发的方式建立了多信号流图模型图形化输入方式；基于关联矩阵和可达性矩阵实现了故障-测试相关矩阵的自动获取；基于相关矩阵分别实现了系统的静态、动态测试性分析；开发了测试性分析软件工具，并通过与商业软件的对比验证了所开发软件的性能。

本课题取得的主要研究成果包括：

(1) 多信号流图模型的图形化建模。一方面，能够在系统原理性框图的基础上方便地实现多信号流图的建模过程；另一方面，借助 Visio 控件的图元绘制、有向图构建功能，大大减少了软件开发的工作量。

(2) 基于多信号流图的故障-测试相关矩阵的自动生成。该过程实现了系统在故障空间内的自动建模；相关矩阵包含了功能故障、完全故障等系统全面的故障信息。

(3) 分别实现了静态测试性和动态测试性分析过程。静态测试性分析能够获取测试性设计所必须的测试性参数指标；动态测试性分析方法中，可以根据分析的要求选择遗传算法或者 AO\* 算法作为分析工具。

(4) 对建模与分析两部分软件的功能进行了验证。所得到的分析结果与国外专业测试性分析软件 TEAMS 的分析结果一致。软件界面友好、自动化程度高，所开发的测试性建模分析软件工具能够基本满足一般系统的测试性设计与分析的要求。

本课题在测试性设计、分析方法研究与软件工具开发方面取得了一定成果，但实际应用中测试性设计是一个非常系统的课题，所开发软件的性能可以从如下两个方面加以完善：

(1) 测试性图形化建模软件仅实现了简单系统基于 Visio 绘图控件的单层次建模，而对于较复杂系统，软件应具有分层建模的能力；

(2) 动态测试性分析中，求解最优测试序列算法的效率有待进一步改进、优化，以适应复杂系统分析的需要。

## 参考文献

- [1] 田仲, 石君友. 系统测试性设计分析与验证[M]. 北京航空航天大学出版社, 2003: 8-9.
- [2] 杨智勇, 许爱强, 牛双诚. 基于多信号模型的系统测试性建模与分析[J]. 工程设计学报, 2007, 14(5): 364-368.
- [3] Champlin C. IRIDIUM satellite: A large system application of design for testability[C]. Proceedings of the IEEE AUTOTESTCON, San Diego, 1993: 392-398.
- [4] Lee L, Andrews J. Satellite hierarchical system test using IEEE 1149.1-based COTS test tools[C]. Proceedings of IEEE Systems Readiness Technology Conference, San Francisco, 1999: 193-201.
- [5] 刘海明, 易晓山. 多信号流图的测试性建模与分析[J]. 中国测试技术, 2007, 33(1): 49-50.
- [6] 刘明, 左洪福. 航空维修策略研究[J]. 飞机设计, 2007, 27(3): 42-45.
- [7] O'Donnell S J, Zarcone A. Managing evolving hardware and software requirements. Proceeding of the IEEE AUTOTESTCON, Anaheim, 2009: 1451-1460.
- [8] .Yang L T, Mauzio J. Testing methodologies for embedded systems and systems-on-chip. First International Conference on Embedded Software and Systems, ICESS, 2004:15-24.
- [9] M. Stancic, H. G. Kerkhoff. Testability Analysis Driven Test Generation of Analogue Cores. Microelectronics Journal (MEJO), 2003, 34 (10): 913-917.
- [10] Sheppard J W. Maintaining diagnostic truth with information flow models. Proceedings of the IEEE AUTOTESTCON, 1996: 447-454.
- [11] Nair R, Lin C, Haynes L, et al. Automatic dependency model generation using SPICE event driven simulation[C]. Proceedings of the IEEE AUTOTESTCON, Boston, 1996: 318-328.
- [12] Simpson W R, Sheppard J W. System complexity and integrated diagnostics[J]. Design & Test of Computers, IEEE, 1991, 8(3): 16-30.
- [13] Deb S, Pattipati K R, Raghavan V, et al. Multi-signal flow graphs: a novel approach for system testability analysis and fault diagnosis[J]. Aerospace and Electronic Systems Magazine, IEEE, 1995, 10(5): 14-25.
- [14] 林志文, 刘松文. 基于多信号流模型的系统测试性分析与评估. 计算机测量与控制, 2006, 14(2): 222-224.
- [15] 高旭. 电子系统多信号建模可视化技术的研究与实现[D]. 成都: 电子科

- 技大学, 2011: 1-3; 41.
- [16] 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.
- [17] Franco Jr J R, Scott J. WSTA-The IDSS weapon system testability analyzer[C]. Proceedings of the IEEE AUTOTESTCON, San Francisco, 1987: 435-440.
- [18] 李凯凯, 丁天宝, 吕启元. 装备多信号流图模型的装备故障诊断方法[J]. 火炮发射与控制学报, 2012(1): 68-71.
- [19] Pillari J, Pertowski T, Protin A, et al. Integrating testability analysis tools with automatic test systems (ATS). Proceeding of the IEEE AUTOTESTCON, Atlanta, 1995: 503-507.
- [20] Su L P, De Marc G, Nolan M. DARTS: An Enabling Technology fox Concurrent Engineering[C]. Proceedings of the IEEE AUTOTESTCON, New York, 1994: 345-356.
- [21] Deb S, Pattipati K R, Shrestha R. QSI's integrated diagnostics toolset[C]. Proceedings of the IEEE AUTOTESTCON, Anaheim, 1997: 408-421.
- [22] 王义琴. 基于多信号模型的电子系统可测性算法及软件设计[D]. 成都: 电子科技大学, 2010: 1-2.
- [23] 吕晓明, 黄考利, 连光耀, 等. 复杂装备系统级测试性指标确定方法研究[J]. 计算机测量与控制, 2008, 16(3): 357-359.
- [24] 孔令宽, 胡政, 杨定新, 等. 基于多信号模型的卫星电源测试性建模与分析[J]. 华中科技大学学报 (自然科学版), 2009, 37(8): 190-193.
- [25] 常亮明. 测试性参数及其指标分配[J]. 质量与可靠性, 1996, 4: 35-37.
- [26] 苏永定, 刘冠军, 邱静, 等. 系统测试性指标确定方法[J]. 测试技术学报, 2008, 22(5): 401-405.
- [27] 常春贺, 杨江平, 刘飞. 复杂装备系统测试性指标确定方法[J]. 装备指挥技术学院学报, 2011, 22(3): 110-113.
- [28] 杨鹏. 基于相关性模型的诊断策略优化设计技术[D]. 长沙: 国防科技大学, 2008: 5-8.
- [29] 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: 9-21.
- [30] 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: 军械工程学院, 2007: 17-23.
- [31] 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, 2011: 55-62.
- [32] 薛凯旋, 黄考利, 连光耀, 等. 基于信息流模型和列表寻优法的装备 BIT

- 系统测试选择技术研究[J]. 弹箭与制导学报, 2008, 28(2): 300-302.
- [33] 陈希祥, 邱静, 刘冠军. 测试不确定条件下基于贝叶斯网络的装备测试优化选择技术[J]. 中国机械工程, 2011, 22(4): 379-384.
- [34] 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: 2-3.
- [35] 杨鹏, 邱静, 刘冠军. 基于扩展的关联模型的测试性分析技术研究[J]. 系统工程与电子技术, 2008, 30(2): 371-374.
- [36] 杨鹏, 邱静, 刘冠军. 测试不可靠条件下的诊断策略优化方法[J]. 仪器仪表学报, 2008, 29(4): 850-854.
- [37] 杨鹏, 邱静, 刘冠军. 多模式系统的测试顺序优化[J]. 计算机工程与应用, 2008, 44(6): 17-19.
- [38] 杨鹏, 邱静, 刘冠军. 计算机辅助测试序列优化[J]. 计算机集成制造系统, 2008, 14(5): 1036-1040.
- [39] 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.
- [40] 王厚军. 可测性设计技术的回顾与发展综述[J]. 中国科技论文在线, 2008, 3(1): 52-58.
- [41] 林志文, 贺喆, 郭丽华. D-矩阵在舰船超短波设备综合诊断中的应用[J]. 计算机测量与控制, 2009, 17(11): 2105-2108.
- [42] 张强, 胡骏, 胡政, 等. 基于多信号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.
- [43] 廖国钢, 李军, 陈春艳. 一种基于模型的测试性分析评估方法[J]. 电子设计工程, 2012, 20(24): 17-20.
- [44] 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.
- [45] Deb S, Ghoshal S, Mathur A, et al. Multisignal modeling for diagnosis, FMECA, and reliability[C]. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Orlando, 1998: 3026-3031.
- [46] Liu D, Zeng Z, Huang C, et al. The testability modeling and model conversion technology based on multi-signal flow graph[C]. Proceedings of the IEEE Conference on Prognostics and System Health Management (PHM), Beijing, 2012: 1-8.
- [47] 冯广斌, 连光耀, 黄考利, 等. 一种基于多信号模型的测试性分析系统[J]. 计算机测量与控制, 2011, 19(9): 2102-2104.
- [48] 王成刚, 周晓东, 杨智勇. 多信号模型故障模式与信号概率关联算法[J]. 测试技术学报, 2009, 23(4): 362-365.

- [49] Pattipatti K R, Luo J, Qiao L, et al. Intelligent model-based diagnostics for system monitoring, diagnosis and maintenance: U.S. Patent 7, 260, 501[P]. 2007-8-21.
- [50] Luo J, Tu H, Pattipati K, et al. Graphical models for diagnosis knowledge representation and inference[C]. Proceedings of the IEEE AUTOTESTCON, Valley Forge, 2005: 483-489.
- [51] 宋智军, 邱仲潘. Visual C# 2010 从入门到精通[M]. 北京: 电子工业出版社, 2011: 1-3; 90-95.
- [52] 曹岩. Microsoft Office Visio 2003 简体中文版精通于提高[M]. 西安: 西安交通大学出版社, 2008: 115-131.
- [53] Microsoft 公司 著, 莱恩工作室 译. 开发 Microsoft Visio 解决方案[M]. 北京: 北京大学出版社, 2002: 251-257; 289-306.
- [54] 郭伟伟, 孙强宇. Visio 图形化电气计算软件开发技术的研究[J]. 电力系统保护与控制, 2008, 36(17): 71-74.
- [55] 王刚, 王忠言, 王方楠. 基于 Visio 图形化风电场潮流计算软件的研究[J]. 黑龙江电力, 2011, 33(1): 27-30.
- [56] 王宝龙, 黄考利, 马立元, 等. 基于依赖矩阵的测试性分析[J]. 计算机测量与控制, 2011, 19(6): 1260-1262.
- [57] 刘晓利, 秦奋涛. 有向图的强连通性分析及判别算法[J]. 计算机应用与软件, 2005, 22(4): 138-139.
- [58] 庞倩超. 基于布尔矩阵运算的有向图可达矩阵[J]. 大庆石油学院学报, 2006, 30(6): 99-100.
- [59] 何小亚, 王洪山. 利用关系矩阵求传递闭包的一种方法[J]. 数学的实践与认识, 2005, 3(35): 172-175.
- [60] 张巧炫, 龙兵, 杨兴霖. 基于多信号模型的可测性指标体系研究[J]. 电子测量技术, 2011, 34(10): 19-22.
- [61] 马永杰, 云文霞. 遗传算法研究进展[J]. 计算机应用研究, 2012, 29(4): 1201-1206.
- [62] 边霞, 米良. 遗传算法理论及其应用研究进展[J]. 计算机应用研究, 2010, 27(7): 2425-2429.

## 攻读学位期间发表的学术论文

1. Chao Lyu, Hao Ding, Shanshan Liu, Lixin Wang, Shujuan Wang. An Initial Implementation of Testability Analysis Based on Multi-Signal Flow Graph Model. The 8th IEEE Conference on ICIEA, 2013.06. 19- 2013. 06. 21(已录用).
2. Wang Xiaoling, Liu Chao, Ding Hao, Wang Lixin. An Improved Model for the Transfer Impedance Calculations of Braided Coaxial Cables. The 7th IEEE Conference on IPEMC. 2012.06.02-2012.06.05: 1078-1081.

## 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《基于多信号流图的系统测试性建模分析及软件设计》，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： 丁昊

日期： 2013 年 6 月 30 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： 丁昊

日期： 2013 年 6 月 30 日

导师签名： 吕超

日期： 2013 年 6 月 30 日



## 致 谢

本课题是在导师吕超副教授的悉心关怀和耐心指导下完成的。吕老师渊博的学识，严谨的治学态度以及细致耐心的指导，帮助我顺利地完成了毕业设计，也对我今后的学习和工作产生了极大的影响，让我深受感染。同时在毕业设计期间，课题组的王立欣老师在我毕设课题的选定方面提出了很多建设性的意见。在此，向吕老师和王老师表示深深的感谢！

在毕业设计期间，我也得到了实验室其他同学的无私帮助和指导，没有他们我不可能顺利地完成整个毕业设计，在此一并向他们表示感谢。特别是张刚和张立强师兄，在他们的指导和帮助下我的课题进行地非常顺利。他们严谨求实、刻苦钻研以及乐观向上的态度值得我学习，在此向两位师兄表示衷心的感谢。感谢邓晨华和孙超同学在硕士期间给予我的陪伴与支持，祝他们今后工作顺利。感谢刘珊珊师妹在我毕业设计过程中给予的支持，祝她在求职过程中如愿以偿。感谢实验室其他师弟师妹们给实验室带来的欢声笑语，是你们给实验室带来了蓬勃朝气，也衷心的希望实验室的明天在你们的努力下越来越美好！

感谢父母的养育和培育之恩，他们的鼓励和期望是我努力向前的动力，在此向他们说一声谢谢。

最后也衷心的感谢即将审阅本论文的各位老师，感谢你们提出的宝贵意见和建议，以及为评审工作付出的辛勤劳动。