



国防科工委“十五”规划教材·控制科学与工程

# 系统仿真技术

彭晓源 主编

北京航空航天大学出版社

北京理工大学出版社 西北工业大学出版社  
哈尔滨工业大学出版社 哈尔滨工程大学出版社

## 内容简介

本书介绍系统仿真定义、应用及分类,系统仿真基础知识、名词术语和仿真计算机系统常用的操作系统、编程语言、开发环境及与仿真应用有关的数据库知识。在较系统地讲述系统仿真技术基本概念、一般原理及方法的基础上,重点讲述工程领域中连续系统的建模/仿真方法,包括连续系统数学模型、连续系统仿真数值积分方法、连续系统离散化仿真、随机控制系统仿真;突出实时仿真系统的建模/仿真,包括半实物仿真和人在回路仿真的原理、方法、系统组成与典型系统分析。同时讲述了近年来迅速发展的分布式仿真技术的原理、规范、平台级分布交互仿真系统及其应用。本书对虚拟现实(VR)技术的定义、组成、特点,三维实时图像的建模、显示原理,虚拟空间声音的建模、显示方法,动感、力感反馈原理及实现方法进行了讲述,使读者初步了解虚拟现实技术如何在系统仿真中应用,以及如何去开发各种应用系统。由于仿真可信度是仿真应用的基础,本书就建模/仿真的 VV&A(校核、验证与确认)技术进行了较系统的讲述。最后一章对常用的数学建模/仿真工具 MATLAB 软件包的主要功能、使用方法进行了简介,便于读者进一步学习和应用。

为便于读者学习和掌握本书主要内容,每一章都配有习题。

本书可作为系统仿真科学与技术及其相关学科的本科生教材,还可供相关专业研究生和工程技术人员参考。

### 图书在版编目(CIP)数据

系统仿真技术/彭晓源主编. —北京:北京航空航天大学出版社,2006.12

ISBN 7-81077-922-2

I. 系… II. 彭… III. 系统仿真 IV. TP391.9

中国版本图书馆 CIP 数据核字(2006)第 124090 号

## 系统仿真技术

彭晓源 主编

责任编辑 王媛媛

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083)

发行部电话:010-82317024 传真:010-82328026

http://www.buaapress.com.cn E-mail:bhpress@263.net

北京时代华都印刷有限公司印装 各地书店经销

\*

开本:787×960 1/16

印张:17.75 字数:398 千字

2006 年 12 月第 1 版 2006 年 12 月第 1 次印刷

印数:3 000 册

ISBN 7-81077-922-2 定价:26.00 元

# 国防科工委“十五”规划教材编委会

(按姓氏笔画排序)

主任：张华祝

副主任：王泽山 陈懋章 屠森林

编委：王 祁 王文生 王泽山 田 蔚 史仪凯

乔少杰 仲顺安 张华祝 张近乐 张耀春

杨志宏 肖锦清 苏秀华 辛玖林 陈光祚

陈国平 陈懋章 庞思勤 武博祎 金鸿章

贺安之 夏人伟 徐德民 聂 宏 贾宝山

郭黎利 屠森林 崔锐捷 黄文良 葛小春

# 目 录

## 第 1 章 绪 论

1.1 系统仿真技术及应用	1
1.2 系统仿真技术定义	1
1.3 系统仿真技术在复杂产品研制中的应用	1
1.4 系统仿真技术在操作人员培训中的应用	2
1.5 仿真分类及任务	2
1.6 按被仿真对象性质分类——连续系统仿真和离散事件系统仿真	2
1.7 按功能及用途分类——工程仿真和训练仿真	2
1.8 按仿真系统体系结构分类——单平台仿真和多平台分布交互仿真	2
1.9 按分布仿真中虚实结合程度分类——构造仿真、虚拟仿真、实况仿真	2
1.10 按仿真时钟与墙钟时间(自然时间)的比例关系分类——实时仿真、欠实时仿真和超实时仿真	2
1.11 按系统数学模型描述方法分类——定量仿真和定性仿真	2
1.12 工程领域连续系统仿真方法分类及任务	2
1.13 数学仿真	2
1.14 硬件在回路仿真	2
1.15 软件在回路仿真	2
1.16 人在回路仿真	2
1.17 现代建模仿真技术的应用及发展	2
1.18 现代建模仿真技术	2
1.19 现代建模仿真技术体系	2

## 第 2 章 系统仿真基础知识

2.1 名词术语	2
2.2 数学模型	2
2.3 仿真模型	2
2.4 实时仿真	2
2.5 分布交互仿真	2
2.6 高层体系结构	2
2.7 建模仿真的校核、验证与确认	2
2.8 基于仿真的采办	2
2.9 虚拟样机	2
2.10 虚拟制造	2



摇摇摇摇圆摇摇仿真网络 .....	猿
摇摇摇摇圆摇摇虚拟现实技术 .....	圆
摇摇圆摇摇操作系统、编程语言开发环境的比较与选择 .....	圆
摇摇摇摇圆摇摇操作系统 .....	圆
摇摇摇摇圆摇摇编程语言及开发环境 .....	缘
摇摇摇摇圆摇摇面向对象建模方法及统一建模语言( UML) .....	圆
摇摇圆摇摇数据库、模型库 .....	猿
摇摇摇摇圆摇摇仿真系统中常用的数据库及特点 .....	猿
摇摇摇摇圆摇摇仿真系统数据库、模型库的开发、管理及应用 .....	猿
摇摇圆摇摇习题 .....	猿
摇摇参考文献 .....	猿
<b>第 猿章 摇摇连续系统数学模型</b>	
摇摇猿猿猿猿猿猿引摇摇言 .....	猿
摇摇猿猿猿猿猿猿常用确定性连续系统数学模型 .....	猿
摇摇摇摇猿猿猿猿猿猿数学模型 .....	猿
摇摇摇摇猿猿猿猿猿猿数学模型间的转换 .....	猿
摇摇猿猿猿猿猿猿离散时间系统数学模型 .....	猿
摇摇摇摇猿猿猿猿猿猿差分方程 .....	猿
摇摇摇摇猿猿猿猿猿猿扎传递函数 .....	猿
摇摇摇摇猿猿猿猿猿猿离散状态空间表达式 .....	猿
摇摇摇摇猿猿猿猿猿猿离散系统结构图 .....	猿
摇摇摇摇猿猿猿猿猿猿关于采样系统 .....	猿
摇摇猿猿猿猿猿猿随机系统数学模型 .....	猿
摇摇摇摇猿猿猿猿猿猿连续随机系统描述方法 .....	猿
摇摇摇摇猿猿猿猿猿猿离散随机系统描述方法 .....	猿
摇摇摇摇猿猿猿猿猿猿随机系统建模 仿真中的专门问题 .....	猿
摇摇圆摇摇习题 .....	猿
摇摇参考文献 .....	猿
<b>第 源章 摇摇连续系统仿真的数值积分法</b>	
摇摇源源源源源源数值积分法原理 .....	猿
摇摇源源源源源源欧拉方法 .....	猿
摇摇摇摇源源源源源源欧拉法 .....	猿
摇摇摇摇源源源源源源改进欧拉法 .....	猿
摇摇摇摇源源源源源源几个基本概念 .....	猿
摇摇源源源源源源龙格-库塔法 .....	猿
摇摇摇摇源源源源源源龙格-库塔法的基本原理 .....	猿
摇摇摇摇源源源源源源固定步长的龙格-库塔法 .....	猿



摇摇摇摇源摇摇龙格-库塔法的稳定区 .....	远
摇摇摇摇源摇摇实时龙格-库塔法公式 .....	远
摇摇摇摇源摇摇变步长龙格-库塔法 .....	远
摇摇源摇摇亚当姆斯法 .....	远
摇摇摇摇源摇摇亚当姆斯显式公式 .....	远
摇摇摇摇源摇摇亚当姆斯隐式公式 .....	远
摇摇摇摇源摇摇预估校正公式 .....	远
摇摇摇摇源摇摇亚当姆斯法的稳定区 .....	远
摇摇源摇摇刚性方程的数字仿真 .....	苑
摇摇摇摇源摇摇刚性方程 .....	苑
摇摇摇摇源摇摇吉尔法 .....	苑
摇摇摇摇源摇摇单步多值法的误差估计 .....	苑
摇摇摇摇源摇摇单步多值法阶与步长的控制 .....	苑
摇摇源摇摇计算方法和计算步长的选择 .....	苑
摇摇摇摇源摇摇积分方法的选择 .....	苑
摇摇摇摇源摇摇计算步长的选择 .....	苑
摇摇摇摇源摇摇习题 .....	苑
摇摇源摇摇参考文献 .....	苑
<b>第 缘章 摇摇连续系统的离散化仿真</b>	
摇摇缘摇摇替换法 .....	愿
摇摇摇摇缘摇摇简单替换法 .....	愿
摇摇摇摇缘摇摇双线性变换法 .....	愿
摇摇缘摇摇根匹配法 .....	愿
摇摇摇摇缘摇摇根匹配法的步骤 .....	愿
摇摇摇摇缘摇摇根匹配法的稳定性和精度 .....	愿
摇摇缘摇摇离散相似法 .....	愿
摇摇摇摇缘摇摇离散相似法的含义和特点 .....	愿
摇摇摇摇缘摇摇离散相似法的精度和稳定性 .....	愿
摇摇缘摇摇状态方程的离散化 .....	员
摇摇摇摇缘摇摇离散化模型的推导 .....	员
摇摇摇摇缘摇摇状态转移矩阵的近似计算 .....	员
摇摇缘摇摇增广矩阵法 .....	员
摇摇摇摇缘摇摇增广矩阵法的基本思想 .....	员
摇摇摇摇缘摇摇不同输入信号下的增广状态方程 .....	员
摇摇缘摇摇面向结构图的数字仿真 .....	员
摇摇摇摇缘摇摇连接矩阵法 .....	员
摇摇摇摇缘摇摇例题 .....	员





## 第 8 章 虚拟现实技术及其在系统仿真中的应用

虚拟现实概述 .....	155
虚拟现实(VR)技术的产生、定义、特点及系统组成 .....	155
人-机多维信息交互原理及显示方式 .....	156
VR系统人机接口 .....	156
视觉反馈原理及三维图像生成 .....	157
计算机实时三维图像生成基本过程 .....	157
三维图像数字化建模 .....	157
坐标变换 .....	157
剪裁与消隐 .....	158
纹理与光照效果 .....	158
立体成像 .....	159
视觉环境的构造方式 .....	159
三维虚拟声的生成原理与显示方法 .....	159
三维虚拟声的生成与显示 .....	159
声音的距离仿真 .....	160
声音的运动仿真 .....	160
三维虚拟声音的合成与定位 .....	160
三维音响的显示 .....	160
触觉力感反馈系统 .....	160
触觉力感反馈原理 .....	160
触觉力反馈仿真设备 .....	161
虚拟操作及力反馈系统举例 .....	161
虚拟现实(VR)技术及在仿真系统中的应用 .....	161
VR技术在训练仿真器中的应用 .....	161
VR技术在虚拟样机及虚拟制造中的应用 .....	161
VR技术在虚拟战场中的应用 .....	161
VR技术在科普及娱乐中的应用 .....	161
虚拟人 .....	161
习 题 .....	161
参 考 文 献 .....	161
<b>第 9 章 建模与仿真的灾毁评估技术</b>	
引言 .....	162
建模与仿真的灾毁评估 .....	162
灾毁评估概念 .....	162
灾毁评估的基本原则 .....	162
灾毁评估的工作过程 .....	162



摇摇摇摇忽忽摇摇灾劫的主要技术方法 .....	圆愿
摇摇忽忽摇摇配配灾劫与 栽栽的关系 .....	圆园
摇摇摇摇忽忽摇摇配配与 栽栽的关系 .....	圆园
摇摇摇摇忽忽摇摇灾劫与 栽栽的关系 .....	圆袁
摇摇忽忽摇摇飞行模拟器的 灾劫与 栽栽过程 .....	圆源
摇摇习摇摇题 .....	圆远
摇摇参考文献 .....	圆远
<b>第 员章摇摇系统建模 仿真工具——酝粤栽</b>	
摇摇员员摇摇酝粤栽功能简介 .....	圆苑
摇摇摇摇摇摇员员摇摇酝粤栽的基本计算功能 .....	圆苑
摇摇摇摇摇摇员员摇摇图形与可视化 .....	圆愿
摇摇摇摇摇摇员员摇摇控制系统模型与分析 .....	圆家
摇摇员员摇摇图形建模 仿真环境——杂粤栽主要功能及应用实例 .....	圆家
摇摇摇摇摇摇员员摇摇杂粤栽模型库 .....	圆愿
摇摇摇摇摇摇员员摇摇杂粤栽模型的建立及模型特点 .....	圆愿
摇摇摇摇摇摇员员摇摇杂粤栽仿真运行 .....	圆怨
摇摇摇摇摇摇员员摇摇杂粤栽应用实例 .....	圆员
摇摇员员摇摇杂粤栽实时代码生成方法及应用实例 .....	圆圆
摇摇摇摇摇摇员员摇摇砸粤栽代码自动生成和实时系统开发环境 .....	圆圆
摇摇摇摇摇摇员员摇摇应用实例 .....	圆元
摇摇习摇摇题 .....	圆愿
摇摇参考文献 .....	圆愿



# 总 序

国防科技工业是国家战略性产业,是国防现代化的重要工业和技术基础,也是国民经济发展和科学技术现代化的重要推动力量。半个多世纪以来,在党中央、国务院的正确领导和亲切关怀下,国防科技工业广大干部职工在知识的传承、科技的攀登与时代的洗礼中,取得了举世瞩目的辉煌成就;研制、生产了大量武器装备,满足了我军由单一陆军,发展成为包括空军、海军、第二炮兵和其他技术兵种在内的合成军队的需要,特别是在尖端技术方面,成功地掌握了原子弹、氢弹、洲际导弹、人造卫星和核潜艇技术,使我军拥有了一批克敌制胜的高技术武器装备,使我国成为世界上少数几个独立掌握核技术和外层空间技术的国家之一。国防科技工业沿着独立自主、自力更生的发展道路,建立了专业门类基本齐全,科研、试验、生产手段基本配套的国防科技工业体系,奠定了进行国防现代化建设最重要的物质基础;掌握了大量新技术、新工艺,研制了许多新设备、新材料,以“两弹一星”、“神舟”号载人航天为代表的国防尖端技术,大大提高了国家的科技水平和竞争力,使中国在世界高科技领域占有了一席之地。十一届三中全会以来,伴随着改革开放的伟大实践,国防科技工业适时地实行战略转移,大量军工技术转向民用,为发展国民经济作出了重要贡献。

国防科技工业是知识密集型产业,国防科技工业发展中的一切问题归根到底都是人才问题。50多年来,国防科技工业培养和造就了一支以“两弹一星”元勋为代表的优秀的科技人才队伍,他们具有强烈的爱国主义思想和艰苦奋斗、无私奉献的精神,勇挑重担,敢于攻关,为攀登国防科技高峰进行了创造性劳动,成为推动我国科技进步的重要力量。面向新世纪的机遇与挑战,高等院校在培养国防科技人才,传播国防科技新知识、新思想,攻克国防基础科研和高技术研究难题



当中,具有不可替代的作用。国防科工委高度重视,积极探索,锐意改革,大力推进国防科技教育特别是高等教育事业的发展。

高等院校国防特色专业教材及专著是国防科技人才培养当中重要的知识载体和教学工具,但受种种客观因素的影响,现有的教材与专著整体上已落后于当今国防科技的发展水平,不适应国防现代化的形势要求,对国防科技高层次人才的培养造成了相当不利的影响。为尽快改变这种状况,建立起质量上乘、品种齐全、特点突出、适应当代国防科技发展的国防特色专业教材体系,国防科工委全额资助编写、出版 200 种国防特色专业重点教材和专著。为保证教材及专著的质量,在广泛动员全国相关专业领域的专家、学者竞投编著工作的基础上,以陈懋章、王泽山、陈一坚院士为代表的 100 多位专家、学者,对经各单位精选的近 550 种教材和专著进行了严格的评审,评选出近 200 种教材和学术专著,覆盖航空宇航科学与技术、控制科学与工程、仪器科学与技术、信息与通信技术、电子科学与技术、力学、材料科学与工程、机械工程、电气工程、兵器科学与技术、船舶与海洋工程、动力机械及工程热物理、光学工程、化学工程与技术、核科学与技术等学科领域。一批长期从事国防特色学科教学和科研工作的两院院士、资深专家和一线教师成为编著者,他们分别来自清华大学、北京航空航天大学、北京理工大学、华北工学院、沈阳航空工业学院、哈尔滨工业大学、哈尔滨工程大学、上海交通大学、南京航空航天大学、南京理工大学、苏州大学、华东船舶工业学院、东华理工学院、电子科技大学、西南交通大学、西北工业大学、西安交通大学等,具有较为广泛的代表性。在全面振兴国防科技工业的伟大事业中,国防特色专业重点教材和专著的出版,将为国防科技创新人才的培养起到积极的促进作用。

党的十六大提出,进入 21 世纪,我国进入了全面建设小康社会、加快推进社会主义现代化的新的发展阶段。全面建设小康社会的宏伟目标,对国防科技工业发展提出了新的更高的要求。推动经济与社会发展,提升国防实力,需要造就宏大的人才队伍,而教育是奠基的柱



石。全面振兴国防科技工业必须始终把发展作为第一要务,落实科教兴国和人才强国战略,推动国防科技工业走新型工业化道路,加快国防科技工业科技创新步伐。国防科技工业为有志青年展示才华,实现志向,提供了缤纷的舞台,希望广大青年学子刻苦学习科学文化知识,树立正确的世界观、人生观、价值观,努力担当起振兴国防科技工业、振兴中华的历史重任,创造出无愧于祖国和人民的业绩。祖国的未来无限美好,国防科技工业的明天将再创辉煌。

张华祝



# 前 言

系统仿真技术是以相似原理、控制理论、计算技术、信息技术及其应用领域的专业技术为基础,以计算机和各种物理效应设备为工具,利用系统模型对实际的或设想的系统进行动态试验研究的一门综合性技术。近年来仿真技术广泛应用于国防、军事、能源、交通等工程与非工程领域。在国防工业及科研部门,特别是航空、航天、船舶、兵器等领域,仿真技术贯穿于方案论证、产品设计、试验、生产制造、使用、维护的全生命周期,因此系统仿真技术是一门通用性强、应用面广、发展迅速、跨学科的综合性的技术。

半个多世纪以来,系统仿真技术在各类应用需求的牵引及有关学科技术的推动下,已经发展形成一门综合性的专业技术体系,迅速成为一项通用性、战略性技术,并与高性能计算一起,成为继理论研究和实验研究之后第三种认识、改造客观世界的重要手段。目前,系统仿真技术正向“数字化、虚拟化、网络化、智能化、集成化、协同化”的方向发展。

本书首先讲述系统仿真技术相关定义、应用及分类、系统仿真基础知识、名词术语和仿真中常用的计算机操作系统、编程语言、开发环境及数据库。在较系统地讲述系统仿真技术基本概念、一般原理、方法的基础上,重点讲述工程领域中连续系统的建模方法,突出工程系统实时仿真的建模/仿真原理、方法,并结合国防科研中的应用实例,理论联系实际,促进学生创新意识和科研实践能力的培养。

本书共 10 章,第 1、2 章论述系统仿真定义、应用及分类,同时介绍系统仿真基础知识。第 3、4、5 章讲述连续系统数学模型,包括常用确定性连续系统数学模型、离散时间系统数学模型、随机系统数学模型和对应的三类仿真模型建模及仿真算法,即连续系统仿真数值积分方法、连续系统离散化仿真方法和随机系统数学模型及仿真算法。第 6 章介绍了随机控制系统的建模、仿真方法。第 7 章介绍了实时仿真系统的基本概念、分类、组成与原理,论述了分布交互仿真系统的发展过程、模型分类及分布



交互仿真标准、协议、通信机制。第 8 章介绍了虚拟现实(VR)技术及其在仿真中的应用,三维图像、虚拟声音的建模/仿真方法,触觉/力反馈原理、设备;并介绍了 VR 技术在仿真系统中的应用,如在仿真器、虚拟样机、虚拟制造、虚拟战场、科普、娱乐中的应用。第 9 章讲述 VV&A 技术,包括基本概念、原则、方法及应用。第 10 章对常用的数学建模/仿真工具 MATLAB 软件包的主要功能、使用方法进行了介绍。本书内容结合当前仿真科技领域的发展及应用并吸收了北京航空航天大学先进仿真技术航空科技重点实验室和飞行器控制一体化技术国防科技重点实验室多年的研究成果和教材。

本书 1、2 章由彭晓源研究员编写,3、4、5 章由王卫红教授编写,6、7、10 章由王江云副教授编写,第 8 章由冯勤研究员、龚光红教授编写,第 9 章由刘丽副教授编写。感谢王行仁教授对本书编写的指导。

# 第 1 章 绪 论

## 1.1 系统仿真技术及应用

### 1.1.1 系统仿真技术定义

系统仿真技术是以相似原理、控制理论、计算技术、信息技术及其应用领域的专业技术为基础,以计算机和各种物理效应设备为工具,利用系统模型对实际的或设想的系统进行动态试验研究的一门综合性技术。系统是研究的对象;模型是系统的抽象;仿真对模型进行实验,以达到研究系统的目的。系统仿真包含三个基本的活动,即建立系统模型、构造仿真模型和进行仿真实验。联系这三个活动的是系统仿真的三要素,即系统、模型(或含系统中的某些实物)、计算机(或含某些物理效应设备)。它们的关系如图 1-1 所示。在整个建模/仿真过程中贯穿了对模型及仿真结果的校核(verification)、验证(validation)与确认(creditation)。

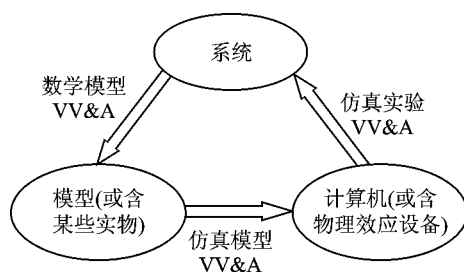


图 1-1 系统仿真三要素及相互关系

通常,在有关仿真的学术讨论、工程应用中,常采用术语“仿真”,其实,所提及的仿真也是针对一个系统,而不是单个元件或部件的仿真。因此,采用术语“系统仿真”,就是为了强调所提及的仿真是针对一个系统,以区分只是涉及单个元部件的仿真。

#### 1. 系 统

系统(system)是客观世界中有着相互联系、相互作用,具有独立行为规律的实体的有机组合。G. 戈登在总结前人思想的基础上,将系统定义为“按照某些规律结合起来,互相作用、互相依存的所有实体的集合或总和”。

系统可以是自然的或人工的、现已存在的或未来计划构建的。尽管世界上的系统千差万



别,但都可以总结出描述系统的“三要素”,即实体、属性、活动。实体确定了系统的构成,也就确定了系统的边界;属性也称为描述变量,描述每一个实体的特征;活动定义了系统内部实体的行为和相互之间的作用,从而确定了系统内部发生变化的过程。

例如人类赖以生存的地球就是一个自然界的复杂巨系统。它包括由地心至地球外层空间十分广阔的范围,由固体地圈、大气圈、水圈和生物圈组成,称为地球系统。每一种圈(子系统)可以由许多实体组成,并都具有描述其特征的属性。不同圈层(子系统)之间具有物理、化学和生物三大相互作用的基本过程,以及人与地球系统之间的相互作用。

又如一台飞行模拟器是一个人工制造的复杂系统。它包括各种操纵机构和显示设备的飞行座舱,含描述飞行动力学、发动机及各机载系统特性仿真模型的计算机系统,提供动感的运动系统;提供力反馈感觉的操纵负荷系统以及视景、音响系统。可以将每个分系统看成是一个实体(或由多个实体组成),操作人员也是一个实体。它们都有描述自身特性的属性以及相互作用的规律、互相交互的信息。例如描述飞行器动力学特性的属性有:三个姿态角、角速度、角加速度、线加速度、位置(可以是地面坐标系中的位置: $X$ 、 $Y$ 、 $Z$ 或者经度、纬度、高度)等。通过操作人员输入各种信息,如操纵机构的位移、发动机油门的位移等,启动各个分系统运行后,按照事先编制的程序(系统软件及应用软件)实现各实体之间交互、运行,并且为操作人员提供各种反馈信息,成为下一步输入的依据。

## 2. 模型

在仿真系统中,对模型(model)的分类有多种提法,在此,提出一种模型分类方法供参考。将模型分为物理模型、概念模型、数学模型和仿真模型。

**物理模型**(physical model):是一类具有某种实物物理特征的模式,如用于水洞、风洞试验的各种缩比实物模型以及各种物理效应设备,如各种转台、负载模拟器、各种人感系统等。前者采用几何外观相似的原理,通过缩小的物理模型在流场中进行实验,获得物理模型各种性能参数,它不能作为仿真实体接入仿真系统。而各种物理效应设备可以反映某种物理模型的特性,可以接入仿真系统,参加动态运行。

**概念模型**(conceptual model):针对一种已有的或设想的系统,对其组成、原理、要求、实现目标等,用文字、图表、技术规范、工作流程等文档来描述,反映系统中各种事物、实体、过程的相互关系,运行过程和最终结果,以此对这种系统进行非形式化的概念描述,这种描述称为概念模型。它可以作为进行相应仿真系统总体设计的概念描述,反映其系统功能是否具有完整性、相容性、连贯性和正确性;可以作为仿真系统建模开发的向导和开发人员与用户沟通理解的工具。

**数学模型**(mathematic model):采用数学符号与数学关系式对系统或实体内在的运动规律及与外部的作用关系进行抽象和对某些本质特征进行描述。

**仿真模型**(simulation model):是将数学模型通过某种数字仿真算法将其转换成能在计算机上运行的数字模型,是一类面向仿真应用的专用软件。因此仿真模型与计算机操作系统、采



用的编程语言和算法(与计算精度、稳定性、实时性要求有关)有密切关系。

### 3. 仿真计算机

用于仿真模型解算的计算机称为仿真计算机(simulation computer)。它是仿真系统的核心部分,是仿真活动的载体。随着计算机技术的发展,仿真计算机不断更新换代并推动仿真技术的发展。仿真计算机经历了模拟计算机(analog computer)、混合计算机(hybrid computer)、数字计算机(digital computer)的发展历程。相应的,仿真的发展也经历了模拟仿真(analog simulation)、混合仿真(hybrid simulation)、数字仿真(digital simulation)三个阶段。

#### (1) 模拟计算机

模拟计算机是利用具有各种数学模型特征的典型电路,组成各种典型的基本运算部件(如加法器、乘法器、积分器、函数器等),这些基本部件的输入和输出是电压,它们之间连接起来可以进行复杂的数学运算。将一个系统的数学方程按相应的运算部件连接起来,按照一定的排题方法与步骤编制模拟计算机的解题程序并启动运行,就可进行模拟计算机仿真。模拟计算机是一种并行运算的机器,计算速度很快,输入、输出均为连续的模拟信号。20世纪50年代是模拟计算机的黄金时期,同期数字计算技术开始发展。在数字计算机运算速度还较慢时只能采用模拟计算机仿真。模拟计算机仿真存在复杂函数生成困难,排题、解题不方便和模拟信号受环境影响,运算精度很难提高等缺点。

#### (2) 混合计算机

为了解决模拟计算机存在的问题,20世纪50年代末至70年代混合计算机从诞生发展到鼎盛时期。混合计算机是将模拟计算技术和数字计算技术灵活结合的一种计算机。它可以分为混合模拟计算机和混合计算机系统两种。前者在模拟计算机中加入大量数字逻辑部件、数控式模拟开关、模拟/数字混合部件;后者是由模拟机、混合模拟计算机、数字机及其接口设备组成的计算机系统。其目的均为了充分利用模拟计算机的并行运算和连续信号处理功能以及数字部件的迭代运算、逻辑运算和复杂函数生成功能。随着数字计算机的发展和性能/价格比的提高,逐渐取代混合计算机,成为仿真计算机的主流机型。

#### (3) 数字计算机

数字计算机与模拟计算机不同,它是用0和1断续变化的电脉冲数码串表示系统状态及数值并进行运算的数字式运算装置,主要由运算器、控制器、存储器、输入设备、输出设备五大部分组成。美国1978年研制出全数字仿真机AD10,1986年又开发出SYSTEM100专用全数字仿真机。中国自行研制的银河系列YH-F1及YH-F2全数字仿真计算机系统分别在1985年和1993年通过国家鉴定并在国内航空、航天等部门得到广泛应用。20世纪90年代通用数字计算机和个人计算机及网络技术迅速发展,采用通用数字计算机和微机联网系统在各个仿真领域得到广泛应用。近十年来,基于RISC技术及并行计算机技术的发展,配有面向问题仿真软件的通用高性能微机、工作站及并行机已成为仿真机的主流。目前,随着高性能计算与网络HPCN(high performance computing and networking)技术的发展与应用,支持分布建



模/仿真的基于网络机群的分布计算机系统正在迅速发展。

### 1.1.2 系统仿真技术在复杂产品研制中的应用

21 世纪制造业面临的新形势是:知识-技术-产品的更新周期越来越短,客户对产品性能和质的要求越来越高。目前,复杂产品制造企业竞争的具体表现是:新产品(P)及其产品的 T(开发时间)、Q(质量)、C(成本)、S(服务)、E(环境清洁)和 K(知识含量)的竞争。世界各国纷纷认识到,必须采用以信息化为核心的“集成化、数字化、虚拟化、网络化、智能化”的建模、仿真,以微加工、光制造、生物加工、纳米制造、绿色制造等高新技术,以及以项目管理为代表的管理技术手段,才能使复杂产品的制造业实现跨越式发展。复杂产品制造业是国民经济和国家安全的重要基础,它的竞争将直接关系到国力的兴衰。现代仿真技术正是综合应用和集成了信息技术及其应用领域技术发展的最新成果,为复杂产品的研制、应用提供了“集成化、数字化、虚拟化、网络化、智能化”的设计、制造、试验、维护、使用的平台及技术。为了增强竞争力,美国首先提出基于仿真采办 SBA(Simulation Based Acquisition)。SBA 的目标就是使武器系统的研制“更好、更快、更省”(better, faster, cheaper)。

美国的联合攻击战斗机 JSF(Joint Strike Fighter)项目采用了从系统需求分析、工程仿真、飞行品质仿真、虚拟制造、硬件在回路仿真至系统集成试验的虚拟样机开发全过程;波音公司利用三维立体产品定义技术和一种无工装的装配与总装技术,为美国国防部预研局和空军研制两架具有 F-16 外形的无人作战飞机 UCAV 的技术验证机,成功实现了提高 T、Q、C、S、E、K 的目的。

#### 1. 分布式建模/仿真技术在复杂产品研制中的应用实例

典型的复杂产品制造建模/仿真技术应用可以美国联合攻击机 JSF 的数字化设计、制造系统为例,在 JSF 全生命周期采用建模/仿真技术,建立协同环境 CE(Collaborative Environment)和分布产品描述 DPD(Distributed Product Description),用于武器系统、子系统、部件的需求分析和概念论证、性能评估、设计及制造。其中 CE 由两部分组成:(1) 攻击作战协同环境 SWCE(Strike Warfare Collaborative Environment):用于作战任务有效性分析、可支持性分析、成本分析、工程与制造性能评估分析;(2) 工程和制造协同环境 EMCE(Engineering & Manufacturing Collaborative Environment):用于武器系统、子系统、部件的需求分析和概念论证、性能评估、设计及制造。JSF-DPD 是 JSF 权威信息的分布集合,可以通过 Web 技术提供给用户,如图 1-2 和图 1-3 所示。JSF-DPD 和其他共享信息将成为整个产品和过程集成开发 IPPD(Integrated Product and Process Development)的集散枢纽。

#### 2. 基于 SBA 思想的动力学虚拟样机

在复杂产品工程研发的前期,采用 SBA 的思想对复杂产品总体设计和方案进行充分论证、反复迭代设计是十分重要的,因为总体设计和方案选择的正确性将对产品后期的开发、制造、成本和最终产品的性能与质量产生重大影响。因此,采用建模/仿真技术建立复杂产品动

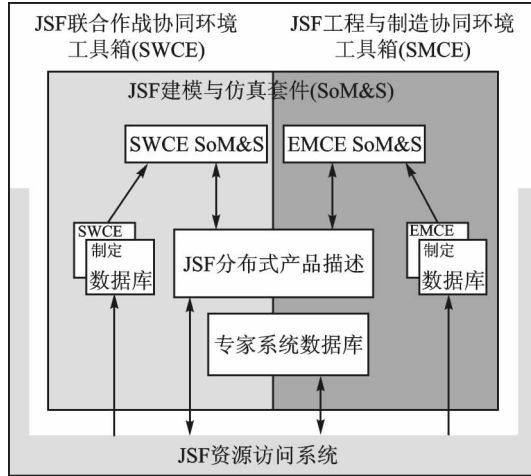


图 1-2 JSF 的协同环境组件及相互关系

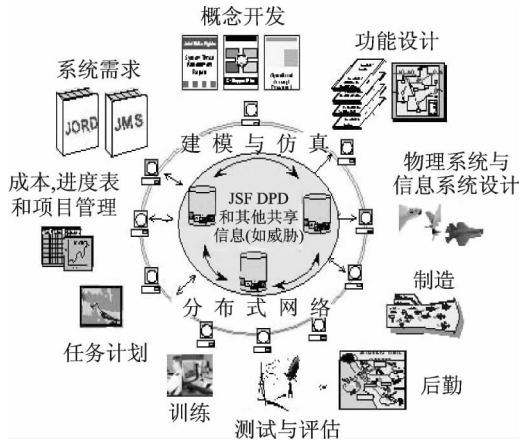


图 1-3 JSF 的综合产品和过程集成开发过程

力学虚拟样机是十分必要的。所谓动力学虚拟样机是指在规范化的建模/仿真支撑平台和资源共享的协同环境中,对影响复杂产品动态性能的主要分系统进行数字化、可视化建模、模型的集成和仿真运行,建立一个能反映复杂产品动态性能、几何外观、主要技术指标的数字化、虚拟化、可视化的产品原型系统。通过对动力学虚拟样机的分析、研究和仿真运行,可以实现对其总体方案和性能参数的设计与评估,对武器系统而言还可以在协同仿真环境中实现该动力学虚拟样机与其他武器平台的交互对抗仿真,进一步考核其战术、技术指标。

为了建立复杂产品动力学虚拟样机,首先必须建立动力学虚拟样机协同建模/仿真支撑平台;建立主要分系统的动力学模型;完成模型的集成、进行仿真运行和产品的性能评估。



图 1-4 是北京航空航天大学先进仿真技术实验室开发的复杂系统动力学虚拟样机协同建模/仿真支撑平台,将虚拟样机协同建模/仿真/管理/可视化等功能集为一体,结构框架划分为 4 层,包括技术支持层、综合集成与管理层、功能驱动层和用户接口界面层,在 WINDOWS 操作系统/网络(TCP/IP 网络协议以及 HLA/RTI)/分布式数据库/仿真网格技术的基础上支持系统管理,项目管理,模型开发及系统集成以及仿真/评估的功能。目标是建立基于分布式数据库/模型库和仿真网格的资源共享协同设计环境和基于高层体系结构(HLA/RTI)的协同仿真环境,其网络拓扑结构如图 1-5 所示,完成动力学虚拟样机的建模/开发、仿真、性能评估。

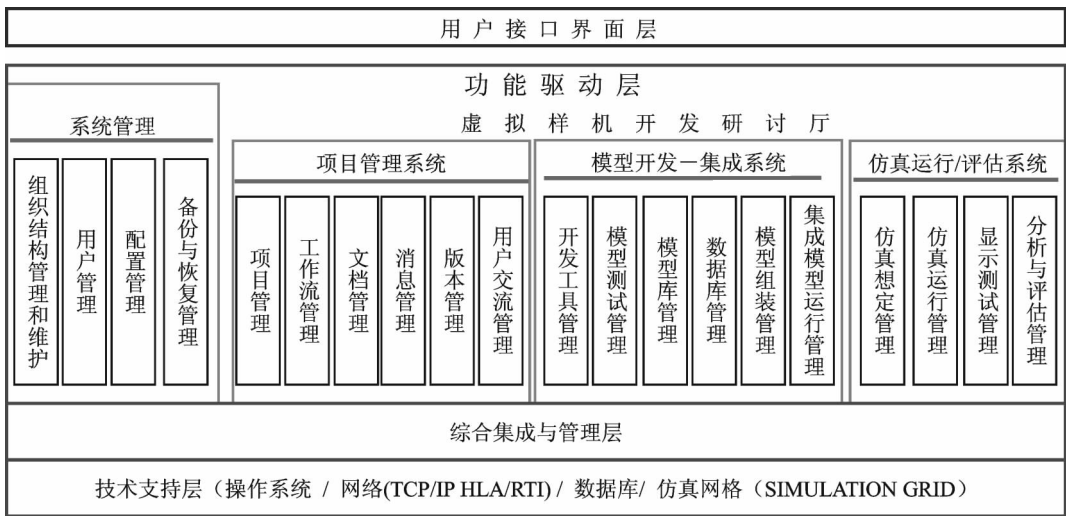


图 1-4 虚拟样机支撑平台结构框架

### 1.1.3 系统仿真技术在操作人员培训中的应用

如前所述,仿真技术的应用,最早源于军事训练的需求。20 世纪 70 年代至今,我国各类训练仿真器,如:飞行模拟器、舰船模拟器、车辆模拟器、火电机组培训仿真系统、化工过程培训仿真系统等发展迅速。各种训练型模拟器及培训仿真系统均具有节能、经济、安全、不受场地和气象条件的限制、能缩短训练周期、提高训练效率等突出优点。用于工程设计和训练用的仿真系统其基本原理和组成是相同的,但对其性能和使用要求有一定区别。用于工程设计的仿真系统要求有更精确或粒度更细的模型,而且要求具有较好的通用性并易于参数的调整与修改、易于对产品性能仿真结果的评估。而训练型仿真系统必须针对具体的设备型号或具体系统。为了达到良好的训练效果,其座舱或控制台的布局必须与实际设备或系统一致。强调操作人员的输入功能及仿真系统的输出响应必须与真实设备或系统一致,即更着重输入/输出外

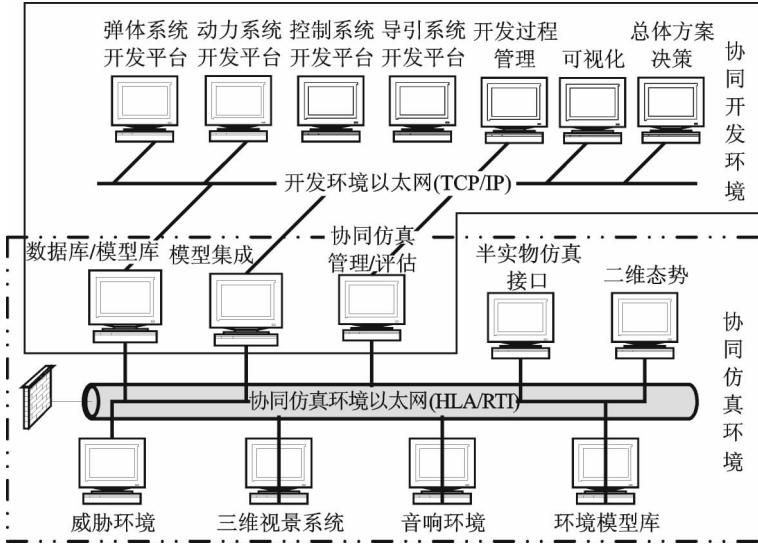


图 1-5 虚拟样机系统建模/开发与仿真环境的网络拓扑图

特性的一致。强调培训人员在模拟器上或在培训仿真系统上的主观感受与在真实系统上的感受尽可能相似。

各种训练模拟器及培训仿真系统的原理及结构尽管差异很大,但基本组成具有共同之处,一般由以下几部分组成:

- (1) 模拟座舱或操作台;
- (2) 仿真计算机系统(包括各种接口及在其中运行的仿真软件);
- (3) 各种显示设备,如仪表、指示灯、系统运行过程的图形界面;
- (4) 环境仿真设备,如:视景系统、音响系统、动感系统(各种运动系统)、力感系统(如操纵力感)等。

由于训练任务、用途不同,各种训练模拟器及培训仿真系统的复杂程度也不同,即便是同一种模拟器也可以分为不同的类和不同的等级,如训练型飞行模拟器按用途和复杂程度不同,一般可分为三类。

### 1. 全任务飞行模拟器

全任务飞行模拟器 FFS(full flight simulator)由功能及外形与真实飞机座舱一致的模拟座舱、仿真计算机系统、视景系统、运动系统、音响系统等组成。可以覆盖包括起飞、着陆在内 90% 以上飞行科目的训练,是一种先进、完整的训练设备。全任务飞行模拟器分为 A、B、C、D 4 个等级,其中 D 级对飞行模拟器性能要求最高,训练科目覆盖面最大。

### 2. 飞行训练器

飞行训练器 FTD(flight training device)一般采用与真实飞机相同的固定座舱,但座舱内



的功能设备是根据训练内容需要而配备的,通常没有运动系统和视景系统(或有简易的视景系统),可以进行某些科目的专门训练,如采用飞行训练器对飞行员进行飞行管理系统或自动飞行系统的飞行操作程序训练或对飞行员进行飞机操纵性、稳定性的操纵技能训练。飞行训练器分为1~7级,7级最高。

### 3. 基于计算机的训练

基于计算机的训练 CBT(computer - based training)最初是利用计算机的多媒体功能,通过图形图像、文本显示、语音播放,进行飞行理论、飞机各个分系统原理及操作使用方法的教学。随着数字仿真技术及高性能微机的发展,CBT 已从静态的教学工具发展成为动态的数学仿真教学工具。在一台微机上可以运行一台飞行模拟器主要分系统的仿真模型,用游戏杆进行操纵,通过逼真的数字式图形仪表、人机交互界面、三维场景,在整个动态飞行仿真过程中可以进行人机交互飞行训练及故障处理能力的训练。

## 1.2 仿真分类及任务

### 1.2.1 按被仿真对象性质分类——连续系统仿真和离散事件系统仿真

#### 1. 连续系统仿真

一个系统,如:一种武器装备(飞机、导弹、坦克等)、一套液压或电动伺服系统,在运行中,它们的状态变化(如:位置、姿态、速度等)在时间上是连续的,而且与在空间的位置变化无关,该系统称为集中参数型连续系统,可以采用线性/非线性微分方程或传递函数(对线性时不变系统)或线性/非线性状态方程等数学模型来描述;若一个系统,如:热传导系统、电力传输系统,其状态变量的变化不仅依赖于时间连续变化,而且与空间位置有关,该系统称为分布参数型连续系统,可以采用偏微分方程数学模型描述;如一个计算机控制系统,其输入、输出及其内部状态变量均是某些时间点上的离散函数,称为线性离散系统或采样系统或离散时间系统,它是以实际连续系统为基础,在时间点上离散化得到的,可采用差分方程数学模型描述。根据以上数学模型建立相应仿真模型并在计算机上运行、实验,称为连续系统仿真(continuous system simulation)。

#### 2. 离散事件系统仿真

一个系统,如电话交换台系统、交通管理系统、物流管理系统等,其系统状态变化发生在随机时间点上,称为离散事件系统,可以用概率分布、排队论等数学模型来描述。在计算机上建立相应的仿真模型并运行和实验,称为离散事件系统仿真(discrete events system simulation)。



## 1.2.2 按功能及用途分类——工程仿真和训练仿真

### 1. 工程仿真

工程仿真(engineering simulation)主要用于产品的设计、制造和实验。采用建模、仿真技术建立数字化虚拟样机、工程模拟器、半实物仿真系统等。可以实现缩短产品开发时间、提高产品质量和服务质量、降低成本和环境污染、增加产品知识含量的目标。

### 2. 训练仿真

训练仿真(training simulation)是采用各种训练模拟器或培训仿真系统,对操作人员进行操作技能的训练、应急处理各种故障的能力的训练以及对指挥员和管理决策人员的指挥、决策能力的训练。使用训练型模拟器及培训仿真系统均具有节能、环保、经济、安全、不受场地和气象条件的限制、能缩短训练周期、提高训练效率等突出优点。

## 1.2.3 按仿真系统体系结构分类——单平台仿真和多平台分布交互仿真

### 1. 单平台仿真

早期的仿真系统多是以一台计算机(由早期的模拟计算机发展为混合计算机直至数字计算机)、一个仿真对象(如:一架飞机、一枚导弹、一套火电机组)及相关的仿真设备构成一个单平台仿真(single platform simulation)系统,其任务是利用该仿真系统进行真实设备或系统的设计、分析、实验与评估或是用于对该种设备或系统的人员进行操作技能的培训。

### 2. 分布交互仿真

随着计算机科学、信息技术的发展和在军事领域、工程领域的应用需求的提高,特别是用于复杂系统的设计、制造及在作战仿真环境下进行产品性能的评估,现代战争中多武器平台在作战环境下的体系对抗研究,乃至当代以网络为中心的信息化作战及在作战环境下的武器系统作战效能的评估的需求都促进了仿真系统体系结构由单平台仿真向多平台的分布交互仿真DIS(distributed interactive simulation)发展。

分布交互仿真及支撑环境的发展经过了3个阶段:20世纪80年代美国国防部启动的联网仿真SIMNET(simulation network);90年代基于协议数据单元(PDU)的分布交互仿真的DIS2.x标准和聚合级仿真协议ALSP(aggregate level simulation protocol);始于90年代中期的高层体系结构HLA(high level architecture)。目前国内外的分布仿真系统多采用基于高层体系结构/运行时间框架(HLA/RTI)的体系结构。

## 1.2.4 按分布仿真中虚实结合程度分类——构造仿真、虚拟仿真、实况仿真

### 1. 构造仿真

构造仿真(constructive simulation)通常是指在军事仿真的战略层次上,以军兵种的作战单元(如:一个兵团、一个飞行大队、一个导弹旅等)为仿真实体,建立聚合级仿真模型,模型



粒度较大,一般是进行体系对抗仿真。有时也将计算机生成兵力 CGF 系统之间的对抗、演练归入构造仿真,均属于数学仿真,可以是实时的,更多是超实时运行,可以理解为是由虚拟人操纵虚拟实体的仿真。

## 2. 虚拟仿真

虚拟仿真(virtual simulation)主要指由各种仿真器和数字仿真实体(包括计算机生成兵力实体)构成的分布仿真系统。仿真实体的模型粒度较细,进行平台级的分布式仿真,一般是实时运行。可以理解为是由真实的人操纵真实实体或部分虚拟实体及虚拟环境的仿真。

## 3. 实况仿真

实况仿真(live simulation)是指在分布式仿真系统中具有真实的装备或设备作为一个仿真实体或仿真实体的一部分,并与其他类型的仿真实体互联完成统一的仿真任务,通常是实时运行的。在这类系统中含有真实的人操纵真实的实体。

### 1.2.5 按仿真时钟与墙钟时间(自然时间)的比例关系分类——实时仿真、欠实时仿真和超实时仿真

#### 1. 实时仿真

实时仿真(real time simulation)是指仿真时钟推进时间与墙钟推进时间完全一致,二者的比例因子等于 1。

#### 2. 欠实时仿真

欠实时仿真(slow time simulation)是指仿真时钟推进时间比墙钟推进时间慢,二者的比例因子小于 1。

#### 3. 超实时仿真

超实时仿真(fast time simulation)是指仿真时钟推进时间比墙钟推进时间快,二者的比例因子大于 1。

### 1.2.6 按系统数学模型描述方法分类——定量仿真和定性仿真

#### 1. 定量仿真

定量仿真(quantitative simulation)是指仿真系统中的模型均为基于一定机理、算法建立起来的确定性模型,其输入/输出参数也是用数值表示,是定量的。如:动力学系统、过程控制系统、电路系统等连续系统的数学模型主要是基于微分方程描述、仿真模型可以采用不同的离散算法;在时间点上离散的系统,如:各种采样系统可以直接采用差分方程来描述。它们的输入/输出参数、初始条件也是定量的。

#### 2. 定性仿真

定性仿真(qualitative simulation)是相对定量仿真而言的,仿真系统中的模型是采用定性描述,在本质上是一种非数值化表示的建模/仿真方法。模型及输入/输出信息、行为表示与分



析均采用一种模糊的、不确定性的、非量化的表示。对定性仿真的研究起源于对复杂系统的仿真研究。因为当实际系统过于复杂或建模知识不完备,或仅存在符号或自然语言形式的知识时,无法构造系统的精确定量模型。相对于数值仿真而言,定性仿真在处理不完备知识和深层次知识及决策等方面具有独特的优势。定性仿真能处理多种形式的信息,有推理能力和学习能力,能初步模仿人类思维方式,人机界面更符合人的思维习惯,所得结果更易理解。常用的定性建模方法有模糊建模法、自然语言建模法、时序逻辑建模法、图像建模法等。

## 1.3 工程领域连续系统仿真方法分类及任务

在工程应用领域连续系统一般分为 4 大类:

### 1.3.1 数学仿真

数学仿真(mathematical simulation)无需昂贵的实物系统,也无需生成模拟客观世界的各种物理效应设备,只需采用层次化、模块化的方法建立真实系统的全部数学模型、选择合理的算法,按照软件工程规范建立仿真模型、编写源代码,并在计算机上运行、实验。数学仿真应根据模型的特点合理选择步长,可以是实时运行,也可以是非实时(欠实时或超实时)运行。数学仿真尤其适用于方案论证、设计分析、研究评估。

### 1.3.2 硬件在回路仿真

硬件在回路仿真(hardware in loop simulation)又称半实物仿真,通常是指在仿真系统中,将系统中的部分实物,如:传感器、控制计算机、伺服执行机构或指挥通信设备等,以及某些物理效应设备,如:转台、负载模拟器等接入含有仿真模型的仿真系统中。由于回路中接入实物,硬件在回路仿真系统必须实时运行。

### 1.3.3 软件在回路仿真

软件在回路仿真(software in loop simulation)系统中的软件是真实系统中专用设备的软件,如飞行器控制系统软件、导航系统软件或地面站测控系统软件等。由于实际系统数字化、智能化程度越来越高,因此越来越依赖设备中的各种专用软件。为了考核该专用软件功能的正确性、完备性和可靠性,必须在仿真系统中通过与其他设备或仿真模型进行动态交互实验。在仿真中,该软件可以在通用的仿真计算机中运行,也可以在真实的专用计算机中运行,但是其软件必须和在专用计算机中运行的操作系统、接口方式、软件代码一致。通常都要求实时运行。



### 1.3.4 人在回路仿真

人在回路仿真(man in loop simulation)是为操作人员的操作技能训练或指挥人员的指挥、决策能力训练而建立的仿真系统或是用于有人操纵的系统(如飞机、船舶、车辆等)的设计、实验与评估。人是仿真系统中不可缺少的环节,特别要求建立能够生成人的感知环境的各种物理效应设备,包括生成视觉、听觉、触觉、动感、力感等的设备,如:视景系统、音响系统、运动系统、操纵负荷系统、头盔、数据手套等。由于操作人员及各种物理效应设备都在回路中,因此仿真系统必须实时运行,计算迭代速率和系统延迟应该满足仿真对象的响应特性、人的生理特性和响应能力的要求,如,视景系统迭代速率不能低于 20 帧/秒,否则人感受到的视觉图像是不连续的。系统延迟一般不能大于帧周期的 2 倍,否则无法正常操作,而且要求各种感觉信息同步或按一定的顺序发生,只有这样才能给操作人员提供较真实和舒适的操作环境。

## 1.4 现代建模/仿真技术的应用及发展

系统仿真技术最初源于军事训练的需求,20 世纪 30 年代美国研制了林克飞行训练器,用于培训飞行员,至此拉开了仿真技术发展的序幕。我国自 20 世纪 50 年代开始,首先在航空与自动控制领域采用了基于模拟计算机的仿真技术。随着数字计算机的发展,60 年代开展了对连续系统及离散系统的仿真研究与应用。70 年代至 80 年代,我国各类训练仿真器,如:飞行模拟器、舰船模拟器、电站模拟器、机车、化工等模拟器的研制与应用获得迅速发展,同时建设了一批水平高、规模大的半实物仿真系统,对我国航空/航天/军用装备的研制发挥了重要作用。90 年代,我国开始对分布交互仿真、虚拟现实等先进仿真技术进行了研究与应用,同时,强化了对建模/验模技术的研究。系统仿真技术发展到今天,对于模型在仿真中的重要性有了更深刻的认识,模型是仿真的核心,模型的逼真度或可信度是仿真是否可信、可用的关键。因此现在将仿真技术也称为建模/仿真技术。以下将对现代建模/仿真技术的含意及研究热点进行讨论。

### 1.4.1 现代建模/仿真技术

现代建模/仿真技术是在原有系统仿真技术定义的基础上加以扩充,强调建模在仿真中的重要性,将建模理论与技术、建模/仿真支撑系统技术列为重要研究内容。其定义仍是以相似原理、模型理论、系统技术、信息技术以及建模/仿真应用领域的有关专业技术为基础,以计算机系统、与应用相关的物理效应设备及仿真器为工具,利用模型参与已有或设想的系统进行研究、分析、设计、加工生产、试验、运行、评估、维护和报废(全生命周期)活动的一门多学科的综合技术。



### 1.4.2 现代建模/仿真技术体系

经过半个多世纪的发展,现代建模/仿真的技术体系日趋形成。宏观地来看,现代建模/仿真技术体系由建模理论与技术、建模/仿真支撑系统技术以及仿真应用技术为核心构成,如图 1-6 所示,也是建模/仿真技术研究的重点。

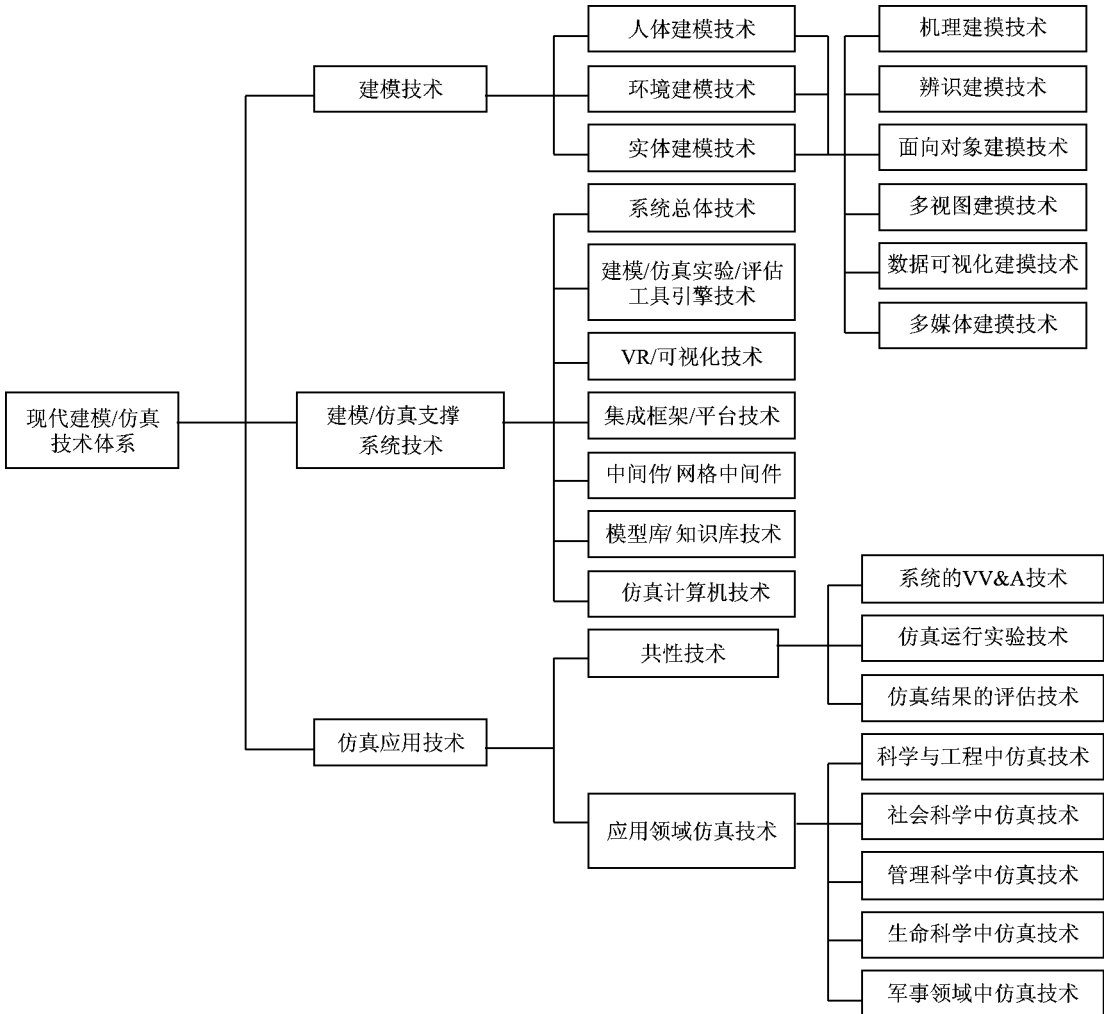


图 1-6 现代建模/仿真技术体系

#### 1. 建模理论与技术

(1) 建模系统分类 包括人体建模理论与技术、环境建模理论与技术、实体建模理论与技



术。其中,人体建模理论与技术主要涉及人体器官组织、外形和人体在外界物理刺激下人体的感知和功能、行为建模;环境建模理论与技术主要解决环境仿真模型的数学描述和可视化建模等问题(如地形/地貌/海洋/大气/空间/电磁环境等);实体建模技术涉及工程与非工程领域各类实体的建模理论与技术。

(2) 建模方法 包括机理建模技术、辨识建模技术、面向对象建模技术、多视图建模技术、数据可视化建模技术和多媒体建模技术等。其中,典型的机理建模技术如用微分方程、差分方程、代数方程描述线性/非线性、连续/离散、确定/随机、集中/分布、定常/时变、存储/非存储等连续系统建模技术;面向活动、面向事件、面向进程等离散事件系统建模技术;神经网络、模糊神经网络、智能代理、本体论等智能系统建模技术;典型的辨识建模技术如最小二乘、极大似然、结构辨识、模糊辨识等;典型的面向对象建模技术如 UML、HLA/OMT 等;典型的多视图建模技术如 Petri 网、Bond 图、IDEF 等。典型的数据可视化建模技术如标量/矢量/张量,动态数据场/流场,多参数数据场可视化建模技术等。典型的多媒体建模技术如声音、图形/图像建模技术等。

## 2. 建模/仿真支撑系统技术

涉及系统总体技术、建模/仿真实验/评估工具引擎技术、VR/可视化技术、集成平台/框架技术、中间件/网格中间件技术、模型库/知识库技术、仿真计算机技术等。

(1) 总体技术,涉及规范化体系结构和采用的标准、规范与协议,系统集成技术和集成方法(典型集成技术,如基于 HLA 标准的全生命周期建模/仿真工具的互操作与集成技术等),以及系统运行模式等。

(2) 仿真引擎技术,涉及支持复杂仿真工程全生命周期各类活动、面向问题的建模/仿真实验/评估算法、程序、语言、环境及工具集。

(3) VR/可视化环境技术,包括计算机图形生成技术与显示技术、多媒体技术、VR 技术、人机交互技术等。

(4) 典型的集成平台/框架技术,如基于 HLA 的协同仿真平台技术。综合仿真环境技术、基于 web Services/Grid 服务的建模/仿真支撑环境等均是当前研究的热点。

(5) 中间件技术,如 CORBA/COM/DCOM/RTI/XML 技术,以及下一代网格中间件技术等。

(6) 模型库/知识库技术,在分布仿真支撑平台及应用中涉及各种数据库技术,如关系/面向对象等数据库技术,分布式数据库、实时数据库、并行数据库、多媒体数据库及以及数据/知识的挖掘技术,数据/知识仓库技术,实时/分布模型库技术等。

(7) 仿真计算机,包括适合于仿真的各类个人计算机、大/中型机、并行机及基于 Internet/web/网格技术的网络计算系统。

## 3. 仿真应用技术

包括共性应用技术及应用领域有关的专用仿真应用技术。



(1) 共性应用技术,涉及系统的 VV&A 技术、仿真运行实验技术、仿真结果的评估技术等。

(2) 应用领域有关的专用仿真应用技术,涉及科学与工程、社会科学、管理科学、生命科学及军事等各领域有关专用仿真应用技术。例如,科学与工程领域中的制造领域仿真应用技术涉及产品研制各阶段有关的仿真技术,包括虚拟产品及其在虚拟环境中功能、性能、行为、外观的仿真技术和对其成本、加工、制造过程,直至使用、报废的仿真分析技术以及企业层次的企业建模(包括结构、资源、过程、供应链等),战略决策分析,市场评估分析,风险、效益评估,质量成本分析等仿真技术。

半个多世纪以来,建模/仿真技术在各类应用需求的牵引及有关学科技术的推动下,已经发展形成了综合性的专业技术体系,并迅速地发展为一项通用性、战略性技术,并与高性能计算(high performance computing)一起,正成为继理论研究和实验研究之后第三种认识、改造客观世界的重要手段。目前,建模/仿真技术正向“数字化、虚拟化、网络化、智能化、集成化、协同化”为特征的现代化方向发展,是解决工程领域,如:航空、航天、机车、船舶、电力、化工等,以及非工程领域,如:交通管理、生产调度、社会经济、军事指挥、生命系统等在研究、实验等方面关键技术的重要手段。

## 第 2 章 系统仿真基础知识

### 2.1 名词术语

以下有关仿真的一些名词术语,在本书或许多仿真应用参考资料中可能会出现。为了加深理解、获得共识,对部分名词术语进行说明。

#### 2.1.1 数学模型

数学模型 MM(mathematic model)是采用数学语言对系统或实体内在的运动规律及与外部的作用关系进行抽象和对其本质特征进行的描述,采用数学方程等描述方式,这种描述与原系统相比能够保持运动规律、信息传递的一致性或相似性。如连续系统的数学模型主要采用微分方程或微分方程组来表征;而离散系统的数学模型主要采用差分方程或差分方程组;离散事件系统用概率分布、排队论等来描述。模型的详细程度(即模型的粒度)必须与研究的目标相匹配。

#### 2.1.2 仿真模型

仿真模型 SM(simulation model)是将系统的数学模型通过数字仿真算法转换成为能在计算机上运行的数字模型,即将数学模型离散化,建立相应的递推公式,便于进行迭代运算。仿真模型是各种仿真算法的有机结合,完成模型的解算和信息处理。不同类型的数学模型有不同的仿真算法,如集中参数连续系统常用的仿真算法有数字积分法、离散相似法等,分布参数连续系统常用的仿真算法是有限差分法,离散事件系统常用的仿真算法有事件调度法、活动扫描法等;如飞行器飞行动力学系统的数学模型,通常采用一组 6 自由度非线性时变微分方程组来描述。而建立其仿真模型,首先要将原来的各个二阶微分方程转换为等效的一系列一阶微分方程,根据仿真任务的不同需要,选择不同的积分算法转换为一组差分方程,进行数值求解。如果用于分析计算、参数选择,可采用精度高、计算稳定性好但计算时间长的 4 阶龙格-库塔法,进行非时实仿真。如果用于时实性要求高的硬件在回路(HIL)仿真或人在回路(MIL)仿真,则可采用欧拉积分法或梯形积分法,在当前计算速度已经不是主要问题的前提下,经过试验,可以选择计算精度较高、计算稳定性好而费时较多的其他积分算法,如二阶阿当姆斯算法等。由于方程中的系数均为非线性时变的数据,因此需要在仿真模型中建立一种不仅随时间变化,而且还随其他参数(如马赫数、高度等)变化的比较精确数据模型,通常采用建立一维、二



维或者多维的关系数表,进行等间距或非等间距的线性插值算法来确定当前的数据。

### 2.1.3 实时仿真

实时仿真 RTS(real time simulation)的重要特征是仿真时钟推进时间与墙钟推进时间完全一致,也体现在数字计算迭代的帧周期上,系统内信息变化速率越快,迭代帧周期越短。如果选择的帧周期时间与仿真对象的响应时间一致则称为实时仿真,如飞行仿真系统的迭代帧周期一般为几毫秒至几十毫秒,电站仿真器的迭代帧周期为秒级。不同特性的仿真对象在实时仿真时对迭代帧周期的长短要求是不同的,它不仅影响仿真的实时性,还影响数字计算的稳定性。数学仿真,当其算法和时间管理机制能满足实时仿真的要求也可以进行实时仿真,一般硬件在回路仿真、软件在回路仿真、人在回路仿真均为实时仿真。

### 2.1.4 分布交互仿真

分布交互仿真 DIS(distributed interactive simulation)定义为:采用协调一致的结构、标准、协议,通过局域网或广域网,将分散在各地、各类的仿真系统互联,构造一个时、空一致,人也参与其中,并实现动态、交互仿真的分布式虚拟环境。分布交互仿真系统具有以下特点:

- (1) 在结构上没有中心计算机节点对仿真过程的控制,采用分布式对等的体系结构;
- (2) 仿真节点具有自治性,节点间信息以广播或组播的方式在网络上传输;
- (3) 所有交互的信息都遵循标准的格式或规范;
- (4) 保证在分布环境下时、空一致性和资源的共享与重用。

分布交互仿真及支撑环境的发展经过了以下几个阶段:联网仿真 SIMNET 和 DIS2. x 标准、聚合级仿真 ALSP、高层体系结构 HLA。目前国内外的分布仿真系统多采 HLA/RTI 的体系结构。

### 2.1.5 高层体系结构

1995年10月,美国国防部在其建模/仿真主计划(master plan)中提出制定一个通用的技术框架,分布仿真高层体系结构 HLA 是该框架的核心内容,并于1996年8月正式公布了 HLA 标准规范。HLA 标准规范由三个文档组成,包括(1)HLA 规则;(2)HLA 接口规范;(3)对象模型模板。HLA 规则规定了 HLA 仿真应用系统中联邦和联邦成员所必须遵循的原则和应当完成的功能;HLA 接口规范定义了运行时间框架 RTI 和联邦成员之间的编程接口;对象模型模板 OMT 定义了联邦中所使用的交互数据、路由空间等信息。

在 HLA 仿真应用系统中,联邦由联邦成员构成,每个联邦成员根据对象模型模板规则建立仿真对象模型 SOM 并建立一个共同的联邦对象模型 FOM。RTI 作为 HLA 的软件实现,将应用层的仿真操作与底层通信功能分离开来,类似于分布式操作系统软件,在联邦成员的运行期间提供通用的标准接口。HLA 的目的是建立一个通用的高层仿真体系结构,使仿真应用



与支撑环境分离,以利于各种类型的模型、仿真应用以及 C<sup>4</sup>I(command, control, communication, computer, intelligence)之间的互操作,同时提高模型的可重用性,使该分布仿真体系结构可以应用在如训练、分析、采办、人在回路仿真、虚拟仿真、实况仿真等更广泛的领域中。

### 2.1.6 建模/仿真的校核、验证与确认

建模/仿真的正确性和置信度是仿真结果是否可信、可用的关键,是仿真技术的生命线。

M&S 的校核、验证与确认 VV&A(verification validation& accreditation)技术是保证和提高仿真置信度的有效途径。VV&A 必须贯穿于建模和仿真的全生命周期,即包括对模型(数学模型、仿真模型、所用的各种数据并结合应用范围及目标)、仿真开发过程(仿真环境、工具、文档、管理)、仿真运行结果及评估方法的校核、验证与确认。由于仿真模型正确与否是仿真置信度的核心问题,因此模型的 VV&A 过程至关重要。以下就模型的校核、验证与确认进行简要说明。

模型的校核(verification):包括对概念模型(数学模型)和仿真模型的校核。确定模型是否正确的代表了被仿真系统的运行机理、规律,并采用了合理的数学的、物理的、逻辑的表示方式,是否采用合理的算法建立了仿真模型。仿真模型是否与仿真应用的领域和目标匹配。校核关心的是“是否正确的建立了模型”。

模型的验证(validation):是从仿真应用的目的出发,确认模型与真实系统一致性的程度。必须通过仿真将仿真试验结果与理论计算结果或真实系统试验结果进行比较以确定仿真模型的可信度。

模型的确认(accreditation):模型校核与验证的结果为模型的确认提供依据。由权威部门根据用户的技术要求并按照一定的规范对仿真过程进行测试,对仿真结果进行定性与定量的分析,如果满足技术指标的要求(包括功能与性能、精度),不仅可以对模型加以确认,也可以对整个仿真系统给以确认,同时可以进行验收和交付。

### 2.1.7 基于仿真的采办

基于仿真采办 SBA 是美国国防部 1997 年提出的一种新型武器采办体系,其核心思想是:提倡建模/仿真技术在武器装备全生命周期中的应用,在 SBA 体系中包括分布、交互、并发、协同、标准化等要素和特点,强调资源共享与资源的可重用性,提倡建立一种注重合作、开放的采办文化和一套行之有效的管理体制。提倡并行的、可以循环的采办形式,以获得最佳方案,节省经费,提高质量。SBA 的目标就是使武器系统的研制“更好、更快、更省”。

SBA 将建模/仿真分为工程(engineering)、交战(engagement)、任务(mission)、体系对抗(theater/campaign)4 个层次。



### 2.1.8 虚拟样机

虚拟样机技术 VP(virtual prototype)是一种基于产品模型的数字化设计方法,以建模/仿真技术为基础,相关领域的知识为支撑,建立产品的数字化、虚拟化原型,使产品的开发人员在产品研制的早期通过对产品的虚拟原型系统进行设计、研究、实验,在一定程度上减少或代替开发物理原型样机和采用物理样机进行实验的一门技术。采用虚拟样机技术可以缩短产品研制周期、降低研制经费、提高产品质量、减轻环境污染、安全节能。

对虚拟样机有多种定义,但其核心思想是指一个产品的数字化模型。在建模/仿真领域关于虚拟样机的比较通用的定义是:虚拟样机是一个与物理原型具有相似功能的产品数字化模型,它从外观、功能和行为上仿真真实产品,并将模型按照内在关系集成在虚拟环境中,用于对该产品全生命周期的研究、测试与评估。在产品开发的不同阶段,由于任务的不同,对虚拟样机模型的描述形式、描述的详细程度(模型的粒度)、建模工具与环境都不尽相同,但都是在计算机上采用建模/仿真的方法来建立数字化、虚拟化的产品原型。

对于复杂的产品在方案设计阶段需要建立动力学虚拟样机,如对于一台飞行器,主要是建立飞行器本身的动力学模型(描述气动、结构、载荷等特性的6自由度、非线性动力学、运动学模型)、机载的控制与导航系统、推进系统的动力学模型等,它们对飞行器总体方案及性能指标影响很大,因此需要建立比较详细的模型(几何的、功能的、性能的)并集成为可以在虚拟环境下飞行的虚拟原型样机。通过设置不同的输入条件、结构参数、特征参数,可以在计算机上反复仿真运行;通过对所提供的实时飞行环境及飞行器三维、二维的可视化画面的直观表现、详细的性能数据和曲线的研究分析,完成优化总体设计方案和参数的目的,为飞行器总体设计提供一个数字化、虚拟化和智能化的设计、试验平台。

### 2.1.9 虚拟制造

虚拟制造 VM(virtual manufacture)是指在计算机上实现与产品制造相关的活动和过程,也称为虚拟产品开发。虚拟制造提供的是一个与制造相关的建模/仿真环境,使得产品的开发规划、设计、制造、装配等均可以在计算机上实现,并且可以对生产过程的各个方面提供技术支持。虚拟产品的开发是基于三维模型,在计算机仿真环境下,对产品进行构思、设计制造、测试和分析,实现产品开发的本质过程。其目标是减少物理样机的试制,缩短开发周期,降低成本。根据工程产品开发过程可以将虚拟制造分为以设计为核心的虚拟制造、以生产为核心的虚拟制造和以控制为核心的虚拟制造。

### 2.1.10 仿真网络

网络通过因特网实现计算资源、存储资源、信息资源、知识资源的全面共享,突破了资源使用上的种种限制,提供了全新的更安全、更方便的方式实现资源共享和模型的互操作,以利于



解决更复杂的问题。

Globus 项目是目前国际上最有影响的网格计算项目之一,开发出能在多种平台上运行的网格计算工具包软件 GT(globus toolkit),提供了一系列的通用网格服务、软件库、编程接口。GLOBUS 3. x 版本采用 OGS(open grid services infrastructure),集成了基于 Web 服务机制的关键网格技术。

仿真网格 SG(simulation grid)是将网格技术、Web 服务与现代建模/仿真技术结合,构建一个包括仿真网格门户层、仿真网格服务管理层、仿真网格服务层、基础网格服务层和仿真资源层的协同建模/仿真支撑环境。在仿真网格服务层将网格化的建模/仿真资源(模型、工具、知识、数据、各种集成的仿真应用、中间件等)通过仿真网格门户层以服务的方式向用户提供,支持工程、非工程领域内各种仿真活动,实现资源/服务跨地域、跨组织的安全共享、集成优化与动态调度。

### 2.1.11 虚拟现实技术

虚拟现实 VR(virtual reality)技术是采用以计算机技术为核心的现代信息技术,生成逼真的人可感知的视觉、听觉、触觉等一体化的特定范围的虚拟世界(包括虚拟环境及虚拟物体)的技术。用户借助必要的设备可以自然的方式与虚拟环境中的对象进行交互、相互影响,从而使产生“沉浸”于等同在真实环境中的感受和体验。VR 技术具有所谓的“3I”特征,即沉浸(immersion)、交互(interaction)和构想(imagination),使得人机交互的内容更加丰富、形象,方式更加自然、和谐。

仿真技术与虚拟现实技术之间的关系有不同的说法和理解。一种认为仿真技术是一门综合技术,它将自动控制、计算机科学与技术、虚拟现实技术等多种技术集为一体来构造复杂的仿真系统。虚拟现实技术可以为这些复杂、逼真的仿真系统特别是人在回路的仿真器,提供具有沉浸感的多维信息人机交互界面。另一种认为虚拟现实技术综合应用了计算机图形图像技术、计算机仿真技术、传感器技术、显示技术等多种技术,而建模/仿真技术是虚拟现实技术的基础。

仿真技术强调仿真模型的建立及在计算机上的运行,虽然越来越重视图像、声音等多种媒体的显示功能,但是由于 VR 系统成本很高,对于沉浸感要求不高的仿真系统,只需要一般的图像、声音显示即可,并非所有的仿真系统都必须包含 VR 系统。VR 技术也并非只用于仿真系统;它在娱乐、工业、医学、建筑等领域均有广泛的应用。

## 2.2 操作系统、编程语言/开发环境的比较与选择

### 2.2.1 操作系统

仿真模型在仿真计算机上运行是仿真最基本的活动。因此若要建立一个仿真系统,在仿



真任务、仿真对象明确之后,必须考虑采用什么仿真计算机及其操作系统。操作系统是配置在计算机硬件上的第一层软件,是计算机系统资源(处理器、存储器、I/O 设备以及数据和程序等资源)的管理者。用户向操作系统提出特定的服务请求,而操作系统则把服务的结果返回给用户。操作系统对于仿真的实时性要求(定时精度、I/O 接口支持等)、人机交互性能、网络通信、数据库的应用都有很大影响。

目前在仿真系统中采用的主流操作系统有 Windows、UNIX 和 Linux 等,在实时仿真中常用的操作系统是 VxWorks。每种操作系统各有特点、长处和不足,因此就有相适应的仿真应用领域。下面就这些问题进行简要阐述,为深入的了解和应用打下基础。

### 1. Windows

Windows 系统是目前主流的 PC 操作系统,其如此流行是因为具有许多吸引用户的特点:

#### (1) 界面图形化

用户的各种需求都显示在显示器界面上,只要移动鼠标,单击、双击即可完成。这为仿真提供了很好的人机交互手段。

#### (2) 多任务

可以同时让电脑执行不同的任务,并且互不干扰,可以同时打开数个浏览器窗口进行浏览。利用这一功能可以同时执行多个仿真任务。

#### (3) 良好的网络支持功能

Windows 9x 和 Windows 2000 中内置了 TCP/IP 协议和拨号上网软件,用户只须进行简单的设置就能上网浏览、收发电子邮件。同时支持局域网内多台 PC 机的通信和资源共享,为分布交互仿真提供了许多支撑手段。

#### (4) 出色的多媒体功能

在 Windows 中可以进行音频、视频的编辑/播放工作,许多高性能的图像建模软件及图形加速卡均基于 Windows 系统运行。利用这一功能可以实现在 PC 上完成仿真的多维信息显示和构造虚拟现实的环境。

#### (5) 良好的硬件支持功能

Windows 系统支持“即插即用(plug and play)”技术,这使得新硬件的安装更加简单。几乎所有的硬件设备都有 Windows 下的驱动程序。

#### (6) 多种应用程序、编程软件

在 Windows 下有众多的应用程序和数种编程软件可以满足用户各方面的需求,并成为通用的具有强大后续开发能力的商品软件。此外,Windows 系统还支持多处理器,这对大幅度提升系统性能很有帮助。

Windows 也存在一些不足。由于 Windows 功能众多且不断扩展,导致软件规模庞大,程序代码繁冗,使得系统不十分稳定,在编程上存在某些漏洞。虽然有些漏洞并不会影响用户的一般操作,但在网络方面的漏洞给一些人提供了入侵系统和攻击系统的机会,以致易于感染



病毒。

虽然 Windows 的操作比较简单,但有时候却不是很有灵活性。由于 Windows 不是专门为实时计算而设计的,因此其定时精度不是很高,普通定时器精度为 55 ms,多媒体定时器精度在 1 ms 左右。I/O 接口数据采集时间也比实时操作系统慢。对于要求毫秒级以下帧周期和小于 1 ms 定时精度的强实时仿真系统不能满足要求。

## 2. UNIX

20 世纪 80 年代中期,由 IEEE 成立了专门委员会对 UNIX 进行标准化的工作,制定了统一的 UNIX 标准并由于自身独具的特点而得到广泛的应用。其主要特点表现在:

### (1) 多用户的分时操作系统

不同的用户可以分别在不同的终端上,对一台具有 UNIX 操作系统的主机进行交互式操作,就像各自单独占用主机一样。

### (2) 可移植性好

由于 UNIX 几乎全部是用可移植性很好的 C 语言编写的,其内核极小,模块结构化,各模块可以单独编译。一旦硬件环境发生变化,只要对内核中有关的模块做修改,编译后与其他模块装配在一起,即可构成一个新的内核,而内核上层完全可以不动。

### (3) 可靠性高

经过十几年的考验,UNIX 系统是一个成熟而且比较可靠的系统。在应用软件出错的情况下,虽然性能会有所下降,但运算仍能可靠进行。

### (4) 开放式系统

UNIX 具有统一的用户界面,其核心程序和系统的支持软件大多用 C 语言编写,因此便于扩展。

可提供两种用户界面,其一是系统级的界面,即用于系统调用,是程序员的编程接口,使用户能充分利用 UNIX 系统的功能;其二是操作级的界面,即命令,它直接面向普通的最终用户,为用户提供交互式功能。程序员可用高级语言直接调用,大大减少编程难度和设计时间。

### (5) 具有可装卸的树型分层结构文件系统

该文件系统具有使用方便、检索简单等特点,并可将所有外部设备都当作文件看待,分别赋予它们对应的文件名。用户可以像使用文件一样使用任一设备。这既简化了系统设计又方便了用户的使用。

UNIX 的定时精度较高,具有多种实时接口及软件,对硬件支持的可靠性高,在一些重要的应用领域作为服务器的操作系统。

## 3. Linux

Linux 是由芬兰赫尔辛基大学的一个大学生 Linus B. Torvalds 在 1991 年首次编写的一种类型 UNIX 系统。Linux 具有许多 UNIX 系统的功能和特点,能够兼容 UNIX,它具有如下优点:



#### (1) 完全免费

Linux 是一款免费的操作系统,用户可以通过网络或其他途径免费获得,并可以任意修改其源代码。来自全世界的很多程序员参与了 Linux 的修改、编写工作,这让 Linux 不断发展。

#### (2) 完全兼容 POSIX 1.0 标准

POSIX(Portable Operating System Interface for Computing Systems)是操作系统的一个标准,描述了操作系统的调用服务接口,用于保证编制的应用程序可以在源代码一级上、在多种操作系统上移植运行。这使得在 Linux 下可以通过相应的模拟器运行常见的 DOS、Windows 的程序,为用户从 Windows 转到 Linux 奠定了基础。

#### (3) 多用户、多任务

Linux 支持多用户,各个用户对于自己的文件设备有自己特殊的权利,保证了各用户之间互不影响。多任务是指 Linux 可以使多个程序同时并独立地运行。

#### (4) 良好的界面

Linux 同时具有字符界面和图形界面。在字符界面用户可以通过键盘输入相应的指令来进行操作。它同时也提供了类似 Windows 图形界面的 X - Windows 系统,用户可以使用鼠标对其进行操作。

#### (5) 丰富的网络功能

互联网是在 UNIX 的基础上发展起来的,Linux 的网络功能和其内核紧密相连,在 Linux 中,用户可以轻松实现网页浏览、文件传输、远程登陆等网络工作,并且可以作为服务器提供 WWW、FTP、E - Mail 等服务。

#### (6) 安全、稳定的性能

Linux 采取了许多安全技术措施,具有对读、写进行权限控制、审计跟踪、核心授权等技术,这些都为安全提供了保障。

#### (7) 支持多种平台

Linux 可以运行在多种硬件平台上,如具有 x86、680x0、SPARC、Alpha 等处理器的平台。Linux 还是一种嵌入式操作系统,可以运行在掌上电脑、机顶盒或游戏机上。Linux 2.4 版内核已经能够完全支持 Intel 64 位芯片架构。同时 Linux 也支持多处理器技术。Linux 系统中的最小定时精度由系统中的实时时钟的频率决定,一般 Linux 系统将该时钟设置为每秒 100 个时钟中断,所以 Linux 系统中一般的定时精度为 10 ms,即时钟周期是 10 ms。随着 Linux 的发展,越来越多的软件厂商会支持 Linux,使其应用软件更加丰富、应用范围更加扩大。

### 4. VxWorks

VxWorks 是美国 WRS(Wind River System)公司推出的一个具有微内核、可裁剪的高性能强实时操作系统,被广泛地应用在通信、军事、航空、航天等实时性要求高的领域中,如卫星、导弹、飞机的控制、制导、导航、通信等领域。VxWorks 是一种功能强大而且比较复杂的操作系统,包括进程管理、存储管理、设备管理、文件系统管理、网络协议及系统应用等几个部分。



可高度裁减,保证系统能以较高的效率运行,并有与之相适应的开发工具。TornadoII 就是为开发 VxWorks 应用系统提供的集成开发环境,TornadoII 中包含的工程管理软件,可以将用户自己的代码与 VxWorks 的核心有效组合起来,按用户的需要裁剪配置 VxWorks 内核;vx-Sim 原型仿真器可以让程序员不用目标机的情况下,直接开发系统原型,作出系统评估;功能强大的 CrossWind 调试器可以提供任务级和系统级的调试模式,可以进行多目标机的联调;优化分析工具可以帮助程序员以多种方式观察、跟踪系统的运行,排除错误,优化性能。

VxWorks 的特点如下:

### (1) 可靠性高

操作系统的可靠性是用户首先要考虑的问题,而稳定、可靠一直是 VxWorks 的一个突出优点。

### (2) 强实时性

实时性是指能够在要求时间内执行完规定的任务并对外部的异步事件作出响应的能力。实时性的强弱是以完成规定任务和作出响应时间的长短来衡量的。

由于 VxWorks 系统本身的开销很小,进程调度、进程间通信、中断处理等系统公用程序精练而有效,它们造成的延迟很短,实时性非常好。VxWorks 对任务的控制采用了优先级抢占(preemptive priority scheduling)和轮转调度(round-robin scheduling)机制,也充分保证了可靠的实时性,使同样的硬件配置能满足更强的实时性要求,为应用的开发留下更大的余地。

### (3) 可裁剪性

VxWorks 由一个体积很小的内核及一些可以根据需要进行定制的系统模块组成。VxWorks 内核最小为 8 KB,即便加上其他必要模块,所占用的空间也很小,且不失其实时、多任务的系统特征。由于它的高度灵活性,用户可以很容易地对这一操作系统进行定制或开发,来满足自己的实际应用需要。

表 2-1 为上述几种操作系统相关性能的比较,在开发仿真系统时可以作为选择操作系统的参考。

表 2-1 几种操作系统的性能比较

	时钟定时精度	外设接口板	人机界面	可靠性	网络支持	跨平台功能	价格	实时性
Windows	一般	丰富、通用	很友好	较差	好	差	一般	一般
UNIX	较高	专用、丰富	较差	高	好	一般	较高	较强
Linux	较高	专用,一般	较好	较高	好	好	免费	较强
VxWorks	高	专用,一般	较好	高	好	差	高	强



## 2.2.2 编程语言及开发环境

要对一个真实的或设想的系统进行仿真研究,首先是建立数学模型,采用一定的算法把数学模型转化为计算机可以运算的仿真模型,同时需要编制仿真程序。程序的表达手段就是程序设计语言(编程语言),它是人-机交换信息的媒体。每种程序语言规定了一套自己的语法和规范,程序员必须遵照这些语法和规则进行。在仿真中常用的有 C/C++、Basic、Java 等,现在所用到的编程语言一般是以一个集成环境的形式提供给用户。在集成的开发环境中,包含了语言编辑器、调试工具、编译工具、运行工具等,例如,Microsoft 的 Visual Studio 开发套件,包含了开发 C/C++ 的 Visual C++;开发 Basic 的 Visual Basic;Borland 公司的 JBuilder 是基于 Java 应用的强大开发环境等。

### 1. C/C++ 和 Visual C++

C 语言是应用广泛的一种重要计算机语言。C 语言功能丰富、使用灵活、应用面广、目标程序效率高、可移植性好,既具有高级语言的优点,又具有低级语言的许多特点,因此特别适合于编写系统软件。C 语言是面向过程、结构化和模块化的语言,但是当问题比较复杂、程序的规模比较大时,结构化程序设计方法就显出它的不足。在 20 世纪 80 年代提出了面向对象的程序设计 OPP(Object-Oriented programming),在这种形势下,C++ 应运而生。C++ 在 C 语言的基础上引进类和对象的概念是面向对象的程序设计语言。通过数据抽象,使程序设计更接近人对描述对象的认识和理解过程,更适合开发大型的程序。C++ 保留了 C 语言原有的所有优点,增加了面向对象的机制。它既可以用于结构化程序设计,又可以用于面向对象的程序设计,是一个功能强大的混合型的程序设计语言。

#### (1) Visual C++ 集成开发环境

Visual C++ 集成开发环境是目前最常用的 C++ 开发环境,提供了一整套开发 Internet 和 Windows 应用程序的工具,包括 VisualC++、Visual Basic、Visual Foxpro 等以及其他辅助工具。Visual C++ 包中除包括 C++ 编译器外,还包括所有的库、例子和为创建 Windows 应用程序所需要的文档。最新的 6.0 版本在编译器、MFC 类库、编辑器以及联机帮助系统等方面都比以前的版本做了较大改进。

Visual C++ 软件包中的 Developer Studio 是一个集成开发环境,它集成了各种开发工具和 VC 编译器。在集成开发环境中还提供大量在线帮助信息,协助程序员做好开发工作。Developer Studio 中除了程序编辑器、资源编辑器、编译器、调试器外,还有各种工具和向导(如 AppWizard 和 ClassWizard),以及 MFC 类库,这些都可以帮助程序员快速而正确地开发出应用程序。

#### (2) 向导

向导(wizard)是一个通过一步步的帮助引导工作的工具。Developer Studio 中包含三个向导,用来帮助程序员开发简单的 Windows 程序,它们是① AppWizard:用来创建一个



Windows 程序的基本框架结构；② ClassWizard:用来定义 AppWizard 所创建的程序中的类；③ ActiveX Control Wizard:用于创建一个 ActiveX 控件的基本框架结构。

### (3) MFC 库

库(library)是可以重复使用的源代码和目标代码的集合。MFC(Microsoft Fundamental Classes)是 Visual C++ 开发环境所带的类库,在该类库中提供了大量的类,可以帮助开发人员快速建立应用程序。这些类可以提供程序框架、进行文件和数据库操作、建立网络连接、进行绘图和打印等各种通用的应用程序操作。使用 MFC 库开发应用程序可以减少用户很多开发工作。

## 2. Visual Basic

Visual Basic 是一种可视化、面向对象和采用事件驱动方式的高级程序设计语言。Visual Basic 简单易学,效率高,且功能强大,程序员不必具有 C/C++ 编程基础便能在 VB 环境下利用上述特点和使用 Windows 内部的应用程序接口(API)函数、动态链接库(DLL)、对象的链接与嵌入(OLE)以及开放式数据库(ODBC)等技术,快速、方便地开发出图形界面丰富、功能强大的 Windows 平台下的应用软件系统。

VB 中高度的封装和模块化减轻了编程者的负担,同时也使开发人员失去了许多访问底层 API 函数和直接与 Windows 交互的机会。利用 VC 与 VB 混合编程可以十分有效地解决这个问题。用 VB 快速开发出漂亮的界面以及外围处理程序,再用 VC 编写底层的各种操作,例如内存的操作、IO 端口的操作、多线程处理、数据处理以及端口通信等,VC 中还可以嵌入汇编语言进行更底层的操作。使用 Activex 控件也可以使得 VB 和 VC 两种语言进行交流。

## 3. Java 语言及其编程环境

Java 语言诞生于 1991 年,起初被称为 OAK 语言。在 Java 语言出现以前,Internet 上的信息内容都是一些乏味死板的 HTML 文档。开发人员也极希望能够在 WEB 上创建一类无需考虑软硬件平台就可以执行的应用程序,SUN 的工程师敏锐地察觉到了这一点,从 1994 年起,他们开始将 OAK 语言应用于 WEB 上并成为 Java 语言。

### (1) Java 语言集成开发环境

Java 的集成开发环境 JBuilder 性能稳定、使用方便,特别适用于创建 Java 组件。此外该软件还具有免费的应用服务器,作为 Servlet/JSP(Java Server Page)的服务器,使得 Servlet/JSP 的开发格外方便。由于 JBuilder 的目标定位是代码开发人员,所以包含了大量的向导程序和其他针对中间层的快速开发工具。其他集成开发环境还有:用于解决整套 IBM 方案的 Visual Age for Java;Sun 公司推出的具有大量 Java 开发工具的 Forte for Java 集成开发环境等。

### (2) Java 的特点

#### ① 平台无关性

平台无关性是指 Java 能在不同的平台上运行。Java 引进虚拟机原理,并运行于虚拟机



上,屏蔽了不同的操作系统间的区别。Java 的数据类型与机器无关,Java 虚拟机(Java Virtual Machine)是建立在硬件和操作系统之上,实现 Java 二进制代码的解释执行功能,并提供给不同平台的接口。

### ② 安全性

Java 的编程类似 C++, 但舍弃了 C++ 的指针对存储器地址的直接操作。程序运行时,内存由操作系统分配,这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器,防止程序的非法访问。

### ③ 面向对象

Java 吸取了 C++ 面向对象的概念,将数据封装于类中,利用类的优点,使程序简洁且便于维护。Java 提供了大量一般的对象类,通过继承即可使用父类的方法。Java 提供的 Object 类及其子类的继承关系如同一棵倒立的树形,根类为 Object 类,Object 类功能强大,经常会使用它及其他派生的子类。

### ④ 分布式

Java 建立在扩展 TCP/IP 网络平台上。库函数提供了用 HTTP(HyperText Transfer Protocol)和 FTP(File Transfer Protocol)协议传送和接收信息的方法。这使得程序员使用网络上的文件和使用本机文件一样容易。

### ⑤ 健壮性

Java 具有检查程序在编译和运行时出错的功能。Java 提供 Null 指针检测、数组边界检测、异常出口、Byte code 校验。Java 的这些功能大大缩短了开发应用程序的周期。

### (3) Java 与 C/C++ 语言的比较

Java 提供了一个功能强大语言所具有的功能。由于 C 和 C++ 被大家接受,所以 Java 设计成类式 C++ 的形式。二者比较如下:

① Java 去掉了以下几个 C 和 C++ 功能:指针运算、结构、typedefs、# define、释放内存等。这就减少了编程中出错可能性,而且使得 Java 很小,整个解释器只需 215 KB 的 RAM。

② 面向对象:Java 实现了 C++ 的基本面向对象技术并有一定增强。

③ Java 与 Internet:早期 Internet 仅可以传送文本和图片,Java 的出现实现了互动页面。Java 以其强安全性、平台无关性、硬件结构无关性、语言简洁同时面向对象,在网络编程语言中占据无可比拟的优势,成为实现电子商务系统的首选语言。Java 程序被放置在 Internet 服务器上,当用户访问服务器时,Java 程序被下载到本地用户机上,由浏览器解释运行。

### (4) JNI 技术

Java 本机接口 JNI(Java Native Interface)是一个本机编程接口,它是 Java 软件开发工具箱 JSDK(Java Software Development Kit)的一部分。JNI 允许 Java 代码使用以其他语言(譬如 C 和 C++)编写的代码和代码库。Java 作为一种跨平台的语言,在很多环境下都获得了成功。然而,目前 Java 对 Windows 构件模型的支持力度不够,使得 Java 程序很难复用



Windows 平台下丰富的构件资源。Windows 构件模型是基于 COM 的,目前 Java 开发工具箱 (JDK) 没有提供任何直接访问 COM 的类库。因此,如果需要访问这些资源,必须通过 JNI 实现。表 2-2 是上述几种编程语言的比较,可以作为用户对编程语言选择的参考。

表 2-2 各种开发语言的特性比较

开发语言	编程模式	可移植性	组件技术	执行效率
C/C++	面向过程/面向对象	一般	支持 COM 组件	高
Basic	面向对象	较差	ActiveX	低
Java	面向对象	好	JavaBean	一般

### 2.2.3 面向对象建模方法及统一建模语言(UML)

#### 1. 面向对象建模方法

面向对象 OO(Object-orient)建模方法是基于面向对象 OO 方法学的一种建模技术,它认为客观世界是由各种对象组成,每一个对象都有自己的运动规律和内部状态,不同对象的组合及相互作用就构成了不同的系统。对象是静态数据和动态行为的封装,属于某个类。这些类具有某种层次化的结构。对象之间通过消息传递、相互作用来实现系统的各种功能。面向对象建模方法可以视为将对象的模型采用数据抽象、封装、模块化、层次、分类、并行、重用、可扩展等概念框架来描述和采用面向对象语言进行程序设计的方法。它为复杂大系统的建模/仿真提供了强有力的手段。

面向对象 OO 仿真软件是由一个个对象的模块构成,相互间通过消息进行通信,一个对象根据获得的消息和定义的操作,作出相应的响应,并实现交互,将仿真结果输出、分析、评估,完成仿真运行。

面向对象建模的主要步骤如下:

- (1) 明确所建系统要实现的目标和完成的任务,对系统进行对象和类的划分,基于面向对象技术建立系统的描述框架;
- (2) 针对每个对象定义静态结构、内部组织、动态行为模型、输入/输出接口,及对象之间的连接关系;
- (3) 采用面向对象编程语言开发仿真软件;
- (4) 对模型及软件进行校核与验证;
- (5) 运行仿真软件,对仿真输出进行评估。

#### 2. 面向对象的统一建模语言 UML

面向对象的统一建模语言 UML(unified modeling language)是在多种面向对象建模方法基础上形成的一种建模语言。UML 具有有严格语法定义,支持面向对象建模的全过程,吸收了 OO 技术的基本直观映像,通过一整套的图形符号和相应的方法学来系统地反映现实世界



的客体,是一种可视化的图形建模语言。UML 包括语义和语法两个部分,运用元模型规定关联和泛化的关系,定义了 5 类共 10 种模型图,包括用例图、静态图(类图、对象图和包图)、行为图(状态图和活动图)、交互图(顺序图和合作图)、实现图(构件图和配置图)。这些图为系统的分析、开发提供了多种图形表示,它们的有机结合就可能分析和构造一个一致的系统。用 UML 建立起的系统模型是独立于最终系统实现语言的模型,因而具有更大的通用性,便于交流。用户、分析人员、测试人员、管理人员都可以从 UML 模型中方便地获取自己需要的信息。采用 UML 语言使各种应用领域的软件开发更加可靠、高效。

### 3. 应用 UML 建立飞机飞行动力学系统仿真模型

飞机飞行动力学仿真系统是飞行器动力学虚拟样机的一个主要仿真系统,也是一个类,它的计算工作量大,与其他分系统的交互关系密切,许多分系统的输入信号都由该系统提供。飞行动力学仿真系统主要功能是对飞行器空气动力特性进行建模/仿真,解算飞行器的 6 自由度非线性全量运动方程,仿真大气环境对飞行的影响。飞行动力学仿真模型主要由“气动模块”、“力和力矩模块”、“运动方程”等模块组成。“气动模块”计算气动力/力矩系数;“力和力矩模块”计算飞行器所受的力和力矩;“质量特性模块”计算转动惯量;“运动方程模块”解算飞行器 6 自由度非线性全量运动方程,获取飞行器的姿态和位置以及其他飞行参数。

为提高飞行动力学系统仿真模型的重用性,应用面向对象的思想 and 建模工具 UML 来设计飞行动力学系统仿真模型类,主要采用用例图、活动图、类图、顺序图描述建模/仿真的全过程。

(1) 用例图:用例分析是飞行动力学系统仿真模型设计和开发的基础,它明确了系统的需求。在用例图中执行者是系统与系统打交道的人和事,执行者执行用例。在用例图中主要的执行者有建模人员、数据文件系统、数据库系统。用例就是系统提供的功能,图中用椭圆来表示用例,带箭头的直线表示信息的交流。飞行仿真模型用例图如图 2-1 所示。

(2) 活动图:为了更加清楚地描述工作流程,设计了活动图,如图 2-2 所示。使用用例图和活动图可以用可视化的手段描述各个功能模块的需求,避免了描述系统语言的模糊性,为正确地建立仿真模型提供了保证。在飞行动力学系统的活动图中,程序启动以后,通过仿真界面输入,运行各功能模块,实现模块间的信息交互,进行飞行仿真。

(3) 类图:面向对象技术里最重要的概念就是类。在 UML 中类图是描述系统中类的静态关系,根据飞行动力学仿真系统的功能,实现了各个类属性和描述方法的设计,设计了一个基类 CPlane,任何其他模型类都是从这个基类派生。在飞行动力学仿真系统类的设计中,考虑到不同飞行器气动特性的差异,气动仿真模型是不同的,而运动方程模型是相同的,因此设计了特殊类和通用类。特殊类,例如计算升力系数类 CACy,计算阻力系数类 CACx,计算侧力系数类 CACz,计算滚转力矩系数类 CAMx,计算俯仰力矩系数类 CAMy,计算偏航力矩系数 CAMz。运动方程模块类 CAeom 是通用类。具体的类图如图 2-3 所示。

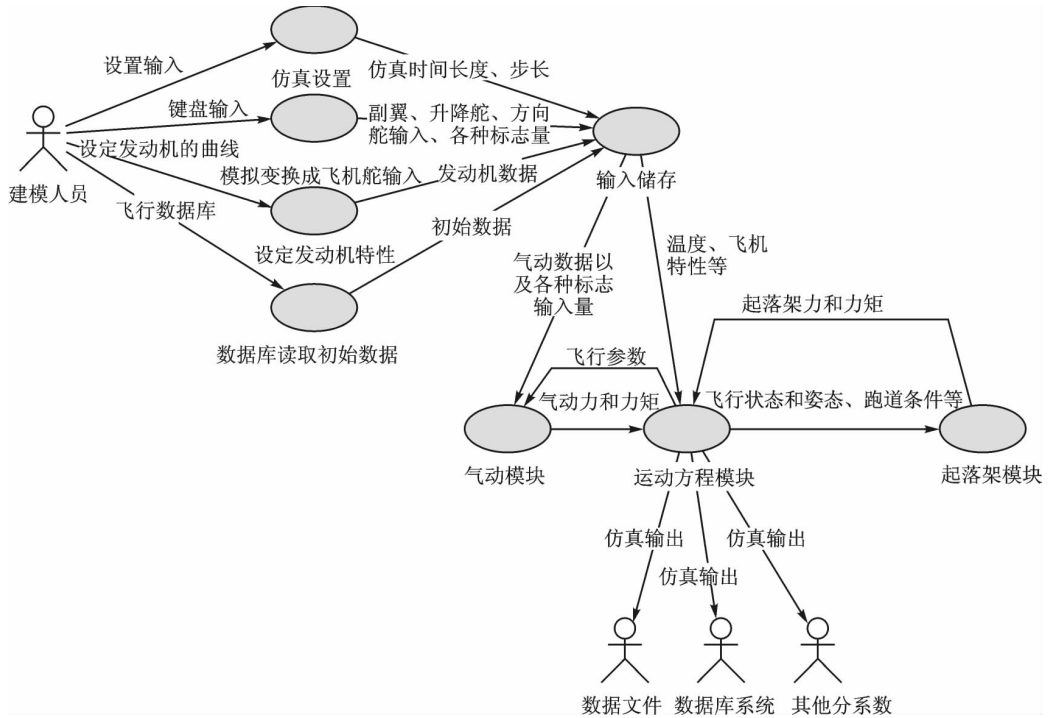


图 2-1 飞行动力学系统仿真模型的用例图

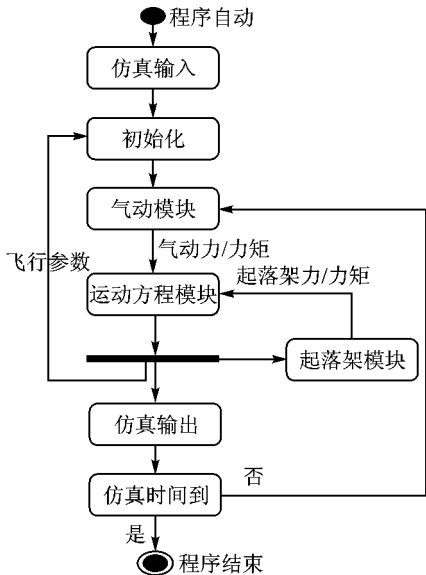


图 2-2 飞行动力学系统仿真模型活动图

(4) 顺序图：类图描述系统模型的各个类的静态机制，为了描述特定类的动态行为，应用顺序图。顺序图，描述了飞行动力学仿真模型的时间推进机制，也就是类的对象的执行的时间关系。在程序中运行的是类的实例化，也就是对象。在飞行动力学仿真系统中，各个类的对象的时间推进是依次进行的。具体飞行动力学仿真系统顺序图如图 2-4 所示。

使用 UML 中的用例图、活动图、类图和顺序图清楚地描述了飞行动力学系统的仿真模型及仿真过程。所设计的特殊类和通用类可以提高飞行仿真模型的重用性。在此基础上可以用面向对象的 C++ 语言来完成软件编制。

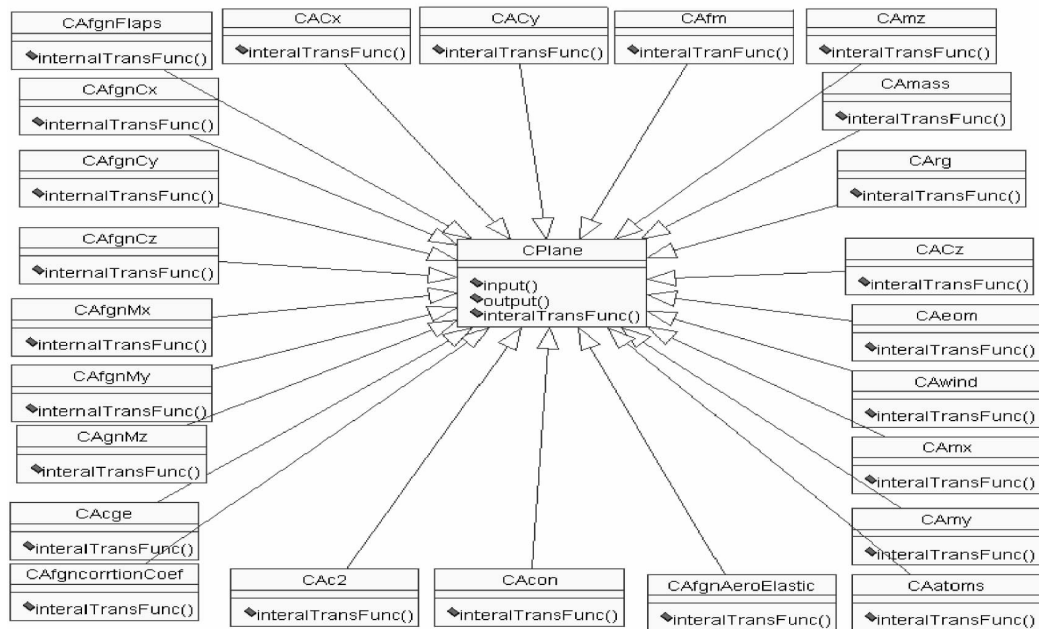


图 2-3 飞行动力学系统仿真模型类图

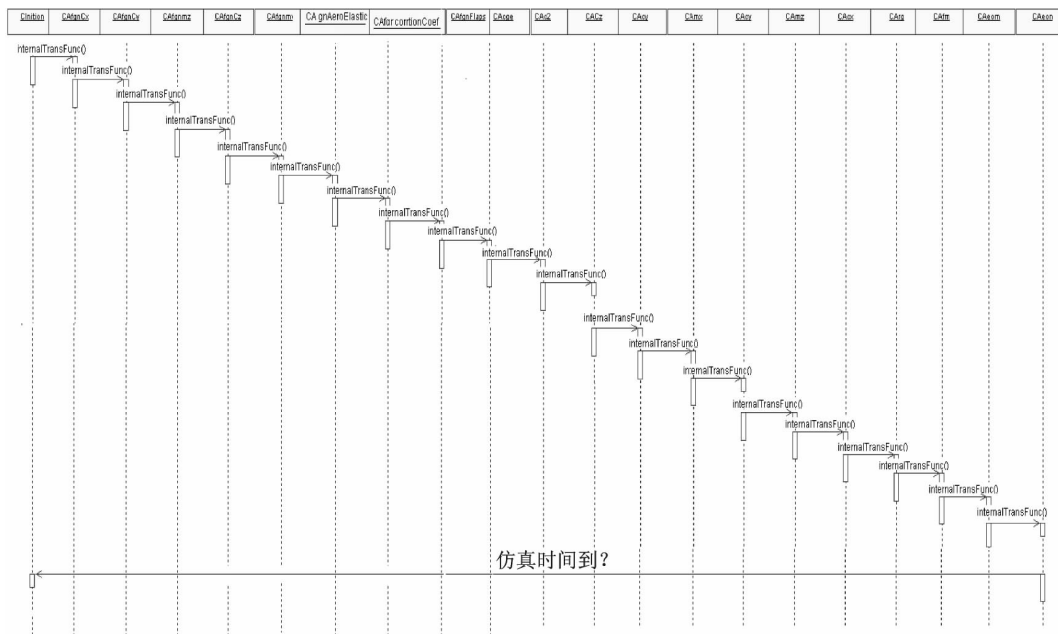


图 2-4 飞行动力学系统仿真模型顺序图



## 2.3 数据库、模型库

数据库是一种存储数据并对数据进行管理的工具,是信息的集合,其作用在于组织和表达信息。数据库管理系统为数据库提供强大的管理功能,包括维护物理数据结构、支持对存储数据的检索和更新数据、保证数据的完整性和安全性、支持多用户服务和并发访问、支持数据的备份和恢复等。

仿真通过模型的驱动来实现的,也可以视为是以数据为中心利用消息传递来实现的一种活动。数据是构造仿真模型的基本元素。在仿真系统中存在大量数据,包括描述系统、实体、环境以及人的行为的静态及动态数据。如:系统数据包括描述系统功能、分类、属性、结构的各种数据,比如一个多武器平台分布仿真系统中武器平台的分类、属性、参数及与作战想定相关的数据等。实体的数据包括实体的性能、外部特征和动力学特性数据等,比如坦克实体的颜色、外形结构尺寸、受损状况、发动机性能、运动速度、位置等。环境数据包括自然环境和人造景观的数据,例如风场、电磁场、虚拟场景、虚拟声音的各种数据。人的行为描述数据包括描述人的思维、决策、推理能力、人的结构和外观等。

模型是对某种对象功能、性能和动态行为的描述。模型库存储模型,提供模型的定义、特征信息及软件代码,是模型元素的集合。模型库管理系统支持对模型库的访问和管理,具有以下功能:

- (1) 模型定义——提供模型描述语言,建立各模型的特征信息、输入/输出信息;
- (2) 模型操作——包括建立、存储、删除、修改、显示、连接、求解、综合报道、分解、编译、仿真运行等操作;
- (3) 控制管理——安全、保密、优先级、完整性和并发控制等管理。

应用数据库管理技术和模型库管理技术对仿真系统的模型/数据进行统一的存储和管理,能够有效地访问、共享和重用仿真系统中的各种模型和数据,提高仿真系统开发和运行的效率。

### 2.3.1 仿真系统中常用的数据库及特点

由于复杂的仿真系统具有分布性,要求采用的数据库能够有效地支持分布式数据管理。目前大型的商业关系数据库,如 ORACLE,IBM 公司的 DB2,Microsoft 公司的 SQL Server 等,都具备成熟完善的分布式数据库管理功能,可以应用在分布式仿真系统中。

#### 1. ORACLE

ORACLE 公司的数据库系统产品在全世界范围内占有相当大的市场份额。从支持成千上万用户的分布式在线事务处理系统到拥有数万亿字节的决策支持数据仓库,从 Client/Server 结构到基于 WWW 的应用,从分布、交互信息环境的企业级的应用到通过直观的方法和可视的工具访问与处理的终端用户级应用,ORACLE 都提供了构造工具、数据库服务器和



计算体系系统。在数据库技术日新月异的变化中,ORACLE 一直占据着数据库庞大市场的领先地位,ORACLE 产品具有以下优势:

(1) 从关键任务的联机事务处理到查询密集的数据仓库的一系列任务,ORACLE 可提供高效、可靠和安全的数据库管理;具有并行查询和优化的能力;支持分区视图,支持在多表连接的视图上进行插入、更新和删除操作。

(2) 支持并行服务器,提供高性能并发控制, I/O 和计算可以同时进行;具有高性能的空间管理能力,可以管理和支持超大对象,管理几十 TB 的数据。

(3) 对灾难性的破坏可以实现快速恢复、提供后发的事务恢复机制、处理系统崩溃后的快速例程启动,具有全面的数据复制能力、透明的分布式查询和分布式事务处理能力。

(4) 与 Web 技术全面集成,可以在企业网络或因特网上使用,支持面向对象技术和多媒体技术。

ORACLE 公司于 2003 年推出了全球首个企业网格计算中间件——Oracle 10g 服务器。Oracle 10g 服务器(Oracle Application Server 10g)是 Oracle 应用基础架构软件集成套件的下一代产品,可以在计算网格中运行企业应用程序。通过从软件供应、用户供应、应用程序管理与监控、工作负载管理、系统管理和监控等方面入手实施网格计算功能。Oracle 10g 服务器能够大幅度降低建立、使用信息技术基础架构所需的高昂成本。

## 2. DB2

IBM 公司最先提出并完善了关系型数据库理论,DB2 属于面向对象的关系型数据库系统,是通用型的数据库产品,其通用性表现在以下方面:

(1) DB2 提供面向对象的具有关系型特征和基于内容的数据搜索能力,可以扩展到搜索诸如文本、图像、视频、音频等新的数据类型;

(2) 提供通用的数据类型和通用的应用支持,支持联机事务处理及联机分析处理,支持用户自定义数据类型、用户自定义函数,支持大型对象数据的存取和查询;

(3) 除了可以应用于传统的 Client/Server 之外,还支持主机、单机、Internet/Intranet 等各种方式的访问,提供对 JAVA 的支持;

(4) 具有突出的优化功能,包括可以综合考虑 CPU 速度、磁盘 I/O 速率、表格尺寸,选择有效访问路径,并且可以重写查询,以得到更高的性能;具有良好的可伸缩性,通用代码集可运行于 Windows NT、UNIX 和 OS/2 平台。

## 3. SQL Server

SQL Server 7 以其可伸缩性的解决方案、强大的数据仓库以及与 Microsoft Office 及 Microsoft BackOffice 的紧密集成,已经成为业界领先的数据库管理系统,并具有关系型数据库管理系统的以下重要特点。

(1) 易于创建、管理、配置和具有可伸缩的解决方案

采用 SQL Server,用户容易建立、管理和部署基于联机交易处理技术的应用程序。



SQL Server 在应用性、可伸缩性、可靠性以及性能方面作出了一系列革新,并为开发者提供了快速简便的编程模式。

#### (2) 支持对数据仓库的应用及对数据的复制

SQL Server 提供了一个综合的平台,这个平台使设计、分析、综合、决策支持以及仿真等功能创建、维护及使用数据仓库更加容易和快捷;支持对数据复制,可以将数据复制到其他的微软 SQL Server 上,或者 DB2、Oracle、Informix、Sybase,甚至微软 Access 这样的数据库中。利用复制功能可以向远程站点中分发数据,可以平衡负载。

#### (3) 具有集中管理及对多种前端(客户端)支持的能力:

无论企业中有多少个 SQL Server 服务器,也无论它们分布在什么位置,都可以由一个选定位置上的 SQL Server 来集中管理。SQL Server 支持多种客户端(如:PowerBuilder、VB、Delphi、VC 等开发的应用程序)。SQL Server 本身包含许多客户端工具,例如:MS - Query、Query Analyzer 和 OSQL。

#### (4) 可视化管理及与 Internet/intranet 互联

大多数 SQL Server 的管理任务都可以通过叫作 SQL Enterprise Manager(企业管理器)的图形化用户界面来完成。可以与 Internet/Intranet 互联,用户通过 Web 浏览器就能直接从 SQL Server 数据库中访问数据。

### 4. 三种数据库产品的比较

ORACLE、DB2 与 SQL Server 是当前数据库市场份额占有最多的三大产品系列,下面对这三种产品各自的特点进行比较,如表 2-3 所列。

表 2-3 三种数据库产品的比较

特 点	ORACLE	DB2	SQL Server
开 放 性	能在所有主流平台上运行(包括 Windows);支持所有的工业标准;采用完全开放策略	能在所有主流平台上运行(包括 Windows)。最适于海量数据。在全球的 500 家最大的企业中,85 % 以上用 DB2 数据库服务器,国内到 1997 年约占 5 %	只能在 Windows 上运行,Windows 平台的可靠性,安全性和伸缩性是非常有限的
安全性	获得最高认证级别的 ISO 标准认证		未获得任何安全证书
性能 及 对 仿 真 应 用 的 支 持	性能最高,保持 Windows NT 下的 TPC-D(事务处理性能委员会商业情报查询性能测试)和 TPC-C(事务处理性能委员会 OLTP 性能测试)的世界记录。能较好地支持仿真数据的加载、记录和存储	适用于数据仓库和在线事务处理,性能较高。能较好地支持仿真数据的加载、记录和存储	操作简单,但对多用户仿真应用的支持性能不佳



续表 2-3

特 点	ORACLE	DB2	SQL Server
客户端支持及应用模式	多层次网络计算,支持多种工业标准,可以用 ODBC、JDBC、OCI 等网络与客户端连接	跨平台,多层结构,支持 ODBC、JDBC 等与客户端连接	C/S 结构,只支持 Windows 客户,可以用 ADO、DOLEDB、ODBC 与客户端连接
可操作性	较复杂,同时提供 GUI 和命令行,在 WindowsNT 和 UNIX 下操作相同	操作简单,同时提供 GUI 和命令行,在 WindowsNT 和 UNIX 下操作相同	操作简单,但只有图形界面
使用风险	应用广泛,具有开发经验,完全向下兼容,完全没有风险。	在巨型企业得到广泛的应用,向下兼容性好,风险小	不十分兼容早期产品,使用需要冒一定风险

### 5. 常用数据库客户端开发及连接工具

随着网络技术的发展,数据库应用越来越多地采用客户端/服务器结构。客户端指的是用来提供用户界面和前端处理的应用程序,服务器则提供由客户使用的资源和服务。在客户端/服务器结构的数据库中,一般由客户负责应用处理,而由数据库服务器负责数据访问和事务管理。在仿真应用中,数据库客户端负责连接数据库服务器,根据仿真应用的需要对数据进行插入、删除、更新和查询等基本操作。数据库客户端应用程序的开发可以选择 PowerBuilder、Delphi、VC、VB 等常用开发工具。

#### (1) PowerBuilder

PowerBuilder 是由美国著名的数据库开发应用厂商 PowerSoft 公司推出的基于客户/服务器应用的快速数据库开发工具,是最具代表性的新一代数据库前端开发工具,具有高效率、高性能、开放、易扩充等优越的性能。PowerBuilder 不仅支持 Microsoft 的开放式数据库连接标准 ODBC,而且还自带一些专用的数据库接口。PowerBuilder 作为一种优秀的数据库前端设计软件,在数据库管理上具有很好的平台无关性。通过 ODBC 和 DBI(数据库接口),PowerBuilder 可以透明地操作各种 DBMS(数据库管理系统),不仅能够方便地与 FoxPro、Access、Excel 交换数据,而且能同 Sybase、Oracle、DB2 等大型关系数据库进行网络交互。PowerBuilder 适用于快速开发仿真应用数据管理应用程序,对仿真数据进行离线查询和管理。

#### (2) Delphi

Delphi 是 Inprise 公司推出的基于对象 Pascal 语言的可视化集成开发工具。利用 Delphi 编程,可以快速、高效地开发出基于 Windows 环境的各类程序,尤其在数据库和网络方面,Delphi 更是一个十分理想的软件开发平台。Delphi 提供了 Database Desktop、BDE Administrator 和 SQL Explorer 等支持强大数据库管理功能的控件,并具有数据库事务处理和数据库异常处理能力。其中,Database Desktop 辅助用户快速创建数据库表,查询、修改、排序数据;BDE Administrator 辅助建立数据库别名;SQL Explorer 主要用于浏览数据库。



### (3) VC, VB

Microsoft 提供的 VC 和 VB 开发工具中都提供了相应的数据库访问组件,包括 ODBC、OLE DB、ADO(ActiveX Data Object)等常用接口。其中 ADO 是一种面向对象、与语言无关的数据访问应用编程接口,利用它可以方便有效地实现对 ORACLE、SQL Server 等关系型数据库的访问与管理操作。它具有易于使用、可以访问多种数据源、访问数据源效率高、方便的 Web 应用、技术编程接口丰富等特点。ADO 是以 OLE DB 技术为基础的,所以它继承了 OLE DB 访问数据源的高效性。由于采用提供 OLE DB 接口的 ORACLE 数据库,使用 ADO 可以提高应用程序的性能,包括 Visual C++, Visual Basic, Visual J++ 以及 VBScript 和 JavaScript 等脚本语言的编程实现接口。基于 ADO 以上的一些特点和优势,通过 ADO 访问 ORACLE 数据库,实现在协同建模/仿真集成环境下对模型库以及数据库数据的加载与更新。ADO 模型包括了连接对象、命令对象、记录集对象、字段对象、参数对象、错误对象以及对象集合和属性。

- ① 连接(connection):用于表示与数据源的连接,以及处理一些命令和事务。
- ② 命令(command):通过已建立的连接操作数据源,添加、删除或更新数据。
- ③ 记录集(recordset):用于处理数据源的表格集,如获取或修改数据等,是在行中检查和修改数据的最主要方法。
- ④ 字段(field):用于表示记录集中的列信息,包括列值等信息。
- ⑤ 参数(parameter):用于对传递给数据源的命令赋参数值等。
- ⑥ 错误对象(Error):用于获得所产生的错误的详细信息。
- ⑦ 属性(property):用于描述或控制对象的行为。

## 2.3.2 仿真系统数据库、模型库的开发、管理及应用

### 1. 仿真系统数据库的开发、管理及应用

数据库系统在仿真系统的开发和运行阶段有不同的功能和特点。在仿真系统开发阶段,数据库系统支持对各类仿真数据的存储、浏览、查询、更新等操作,对数据库的访问是非实时的。首先要根据仿真系统的需要和仿真数据的特点创建相应数表,将仿真数据存储于数据库中。然后根据仿真系统的数据访问需求建立相应的前台管理应用,实现对数据的浏览、查询、更新操作。前台管理应用可以采用多种工具开发,如:PowerBuilder、Visual Basic、Visual C++等,不管用哪种工具开发前台,都要与数据库建立连接,能对仿真数据进行有效的管理。仿真系统运行阶段对数据库系统的访问应是实时的。仿真运行结束,数据的后处理(分析、评估)可以是非实时的。如果要求复现仿真过程,可以将按时标存入数据库的各类数据一次读入内存,再实时地进行播放。

归纳数据库在仿真中的主要功能如下:规范的存储、管理各种仿真资源(数据、模型、工具等)提高其安全性、完整性、可重用性,是分布式系统的资源共享、信息交互的基础。



### (1) 各类仿真数据的初始化加载

仿真系统各节点需要从数据库中获取相应的模型特征参数和性能数据,以满足仿真运行的需要。考虑到仿真运行的实时性要求,不可能在各节点应用程序的运行过程中频繁地访问数据库,因此在各节点应用程序的初始化过程中建立对数据库服务器的连接和数据加载。根据数据库中数据的存放模式,打开相应的数表,取出对应的参数。

### (2) 仿真过程的记录与回放

仿真系统的动态数据是在仿真过程中动态产生的各类不同层次和定义的系统特征、属性数据,这些数据的集合可唯一地表示整个仿真过程。如,在基于 HLA/RTI 的分布仿真系统中,根据联邦成员发布/订购的对象类属性和交互类,根据联邦及联邦成员间数据交互的信息,确定动态信息数据的具体内容、类型等,在数据库中建立相应格式的表格来接收管理各成员的交互信息。根据回放精度的需要,确定记录的周期。设置定时器,将交互数据动态地、在线地存放到数据库预先建立的数表中。回放时,根据系统分析的需要全过程复现或截取某一段时间的数据复现。可以按仿真周期实时复现,也可以超实时或者欠实时,根据系统评估的需要而定。

## 2. 仿真系统中的模型库开发、管理及应用

模型库管理系统(图 2-5)是在数据库中建立的一种反映模型特征的特殊的数据管理系统,其功能是完成对仿真系统中各种模型组件的管理和提取,提供良好的人-机交互界面,便于用户的使用。模型库管理系统的核心是模型字典,模型字典包含模型库中所有模型的特征描述和存储信息,比如模型的输入集合和输出集合,模型的功能描述,以及模型特征参数的存储信息等,是关于模型元数据的特殊数据库,也是一种模型的索引形式。同时,模型字典中的特征性能数据和对应的性能参数数据库相互映射,可以对用户的动作进行一致性的回应。模型具有层次性和关联性,所以模型字典也是一种层次化的关系数据结构,可以使用层次化的存储方式,采用关系数据库进行存储和管理。在建立规范的数据接口的基础上,底层模型应该是源代码形式的“组件”,方便开发人员进行模型的二次开发,而分系统级别的模型,由于结构稳定,考虑到系统集成的要求,应该是可执行的二进制代码模型,如动态链接库或 COM 组件。组件的核心思想是对接口进行标准化,因此充分考虑到接口定义标准,每一类模型具有统一的接口和统一的结构,在软件结构上实现同类模型的无缝替换,通过模型字典索引和管理的模型就可以进行组装和仿真运行。

通过对模型定义的规范,确定数据库中对模型信息的存储内容和格式,将模型用各种数据项表示。设计的模型字典主要包括如下信息:

- (1) 模型名、模型原理、使用条件、使用方法、模型类型、所属模型库、开发信息、使用信息、相关模型;
- (2) 源程序文件名、中间目标文件名、可执行文件名、编写语言、编译系统;
- (3) 子(类)模型信息;

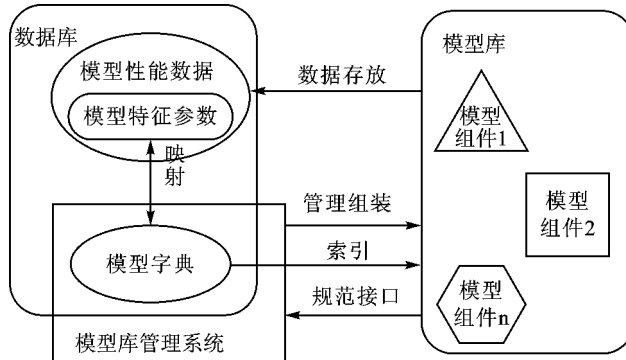


图 2-5 模型库、模型库管理系统与数据库

- (4) 输入、输出信息: 变量序号、变量名、变量含义、变量类型;
- (5) 特征参数信息: 参数序号、参数名、参数含义、数据类型、参数值、单位。

对于一个模型来说,最本质的特点包括:模型信息、子类模型信息、模型输入变量、模型输出变量、模型的子模型信息、模型特征参数(图 2-6)。以此为基础,可以在数据库中建立 6 个相应结构的数表来存放以上的模型信息。

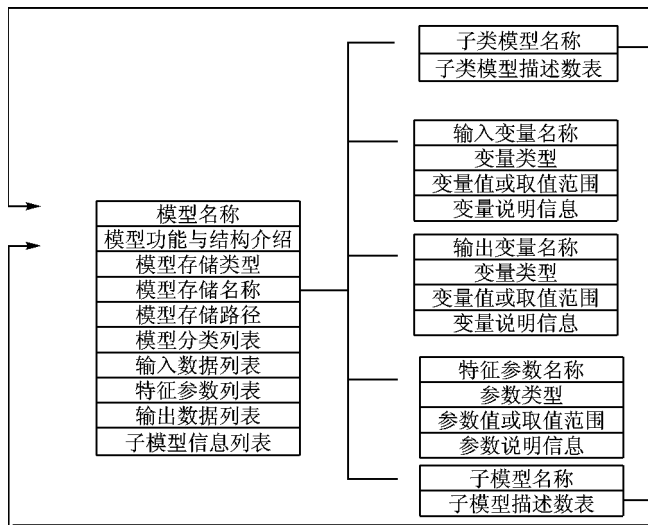


图 2-6 模型字典的实现结构

在仿真系统开发阶段,对模型库进行模型生成、访问与管理等操作,以完成模型的开发与测试过程。在模型字典中建立相应的层次结构模型,建立对应的性能数据表。在仿真系统运行阶段,要从模型库中提取相应模型组件,进行组装、集成和运行。由于模型字典和模型的相



关数据都存储在数据库中,因此数据库技术仍然是模型库访问应用开发的基础。

## 习 题

2.1 论述系统仿真技术如何用于复杂产品的设计、试验及人员训练等方面,与传统方法比较具有哪些优势?

2.2 建立一个实时仿真系统,如一台飞行训练器,在选择仿真计算机、操作系统、编程语言、数据库、建模方法、仿真算法、仿真设备时应考虑那些因素?并给出具体选择和说明其理由。

## 参考文献

- [1] 王行仁,等. 飞行实时仿真系统及技术. 北京:北京航空航天大学出版社,1998.
- [2] 肖田元,等. 系统仿真导论. 北京:清华大学出版社,2000.
- [3] 康凤举. 现代仿真技术与应用. 北京:国防工业出版社,2001.
- [4] 吴重光,等. 仿真技术. 北京:北京化学工业出版社,2000.
- [5] 李伯虎,等. 现代建模/仿真技术的新发展. 系统仿真学报,2004.
- [6] 李妮,彭晓源,等. 数据库技术在动力学虚拟样机中的应用. 系统仿真学报,2003.
- [7] 刘杰,彭晓源,等. 虚拟样机及其协同建模环境研究. 北航学报,2003.
- [8] 李妮,彭晓源,等. 复杂系统虚拟样机协同建模/仿真支撑平台. 计算机集成制造系统,2004(10).
- [9] 胡亚海,彭晓源,等. 复杂分布仿真互操作支撑环境——AST - RTI的研究与开发. 计算机集成制造系统,2004(5).
- [10] 张文苑,贾荣珍,等. 基于 UML 的动力学虚拟样机仿真建模研究. 系统仿真学报,2003.
- [11] IEEE Standard Glossary of Modeling and Simulation Terminology. IEEE STD 6103,1989.
- [12] Dde K. Pace. eas About Simulation Conceptual Model Development JOHNS HOPKINS APL TECHNICAL DIGEST, 2000, 21(3).

# 第3章 连续系统数学模型

仿真技术是以控制论、相似原理和计算机技术为基础,借助于系统模型对真实的或设想的系统进行实验研究的一门综合性技术。由此可知仿真是建立在模型基础上的实验技术,因此有关模型的问题是仿真技术中需要深入研究的问题。

## 3.1 引言

模型是系统某种特定性能的一种抽象形式,是系统本质的表述。本章主要讨论数学模型的相关问题。根据所描述系统的不同,数学模型分确定性模型与随机性模型、集中参数模型与分布参数模型、线性模型与非线性模型、时变模型与定常模型。本章介绍仿真中常用的连续时间系统数学模型、离散时间系统数学模型和随机系统数学模型。

## 3.2 常用确定性连续系统数学模型

连续系统模型由表征系统变量之间关系的方程描述,常用的确定性连续系统数学模型有微分方程、传递函数、状态空间表达式三种。本节在介绍这三种模型之后,还将介绍模型间相互转换的方法。

### 3.2.1 数学模型

#### 1. 微分方程

微分方程是系统的时域模型。连续系统的微分方程一般为

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_{n-1} \dot{y} + a_n y = u \quad (3-1)$$

式中, $y$ 和 $y^{(i)}(t)$ 为系统输出和输出的各阶导数, $u$ 为系统的输入, $a_1, a_2, \cdots, a_n$ 为常系数。

若系统中包含输入的导数项,则微分方程写为

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_{n-1} \dot{y} + a_n y = b_0 u^{(m)} + b_1 u^{(m-1)} + \cdots + b_{m-1} \dot{u} + b_m u \quad (3-2)$$

式中, $m \leq n$ 。

在仿真中,微分方程及微分方程组是一种较常用的模型,第4章所讲的数值积分法就是针对微分方程模型的一种数值仿真方法。



## 2. 传递函数

零初始条件下,系统输出信号的拉氏变换与输入信号的拉氏变换之比称为系统的传递函数。对式(3-2)两端取拉氏变换,并设初值为零,则可得系统的传递函数一般形式:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \quad (3-3)$$

传递函数是一种在频域描述系统特性的数学模型。微分方程和传递函数都是描述系统输入、输出特性的模型,不能描述系统的内部特性。

## 3. 状态空间表达式

状态空间表达式也是系统的时域模型。与微分方程不同的是,它引入了状态变量,能够描述系统的内部特性。

若要建立状态空间表达式,先要确定系统的状态变量。系统的状态是描述系统的最小一组变量,只要知道了在  $t=t_0$  时刻的这组变量和  $t \geq t_0$  时刻的输入,就完全能确定在任何  $t \geq t_0$  时刻系统的行为。若完全描述一个给定系统的动态过程需要  $n$  个状态变量,则称由这  $n$  个状态变量构成的向量为状态向量。

状态空间表达式由状态方程和输出方程组成

$$\left. \begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{X} + \mathbf{D}\mathbf{u} \end{aligned} \right\} \quad (3-4)$$

式中,  $\mathbf{X} = [x_1, x_2, \cdots, x_n]^T$ , 为  $n$  维状态向量,  $\mathbf{u}$  为  $r$  维输入向量,  $\mathbf{y}$  为  $m$  维输出向量;  $\mathbf{A}_{n \times n}$  为状态矩阵,  $\mathbf{B}_{n \times r}$  为输入矩阵,  $\mathbf{C}_{m \times n}$  为输出矩阵,  $\mathbf{D}_{m \times r}$  为直传矩阵。

## 4. 结构图

结构图是系统中每个环节功能和信号流向的图解表示,比较直观,对单入单出线性系统通过结构图可以很容易地得出整个系统的传递函数;对多入多出系统和含有非线性环节的系统也可以通过面向结构图的仿真方法求出系统的动态特性。图3-1至图3-3分别给出了几种系统的结构图。

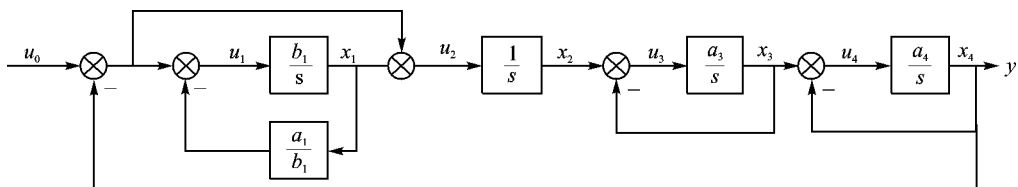


图 3-1 单入单出系统结构图

### 3.2.2 数学模型间的转换

仿真时常见的模型转换主要有微分方程转化为传递函数、微分方程转化为状态方程、传递

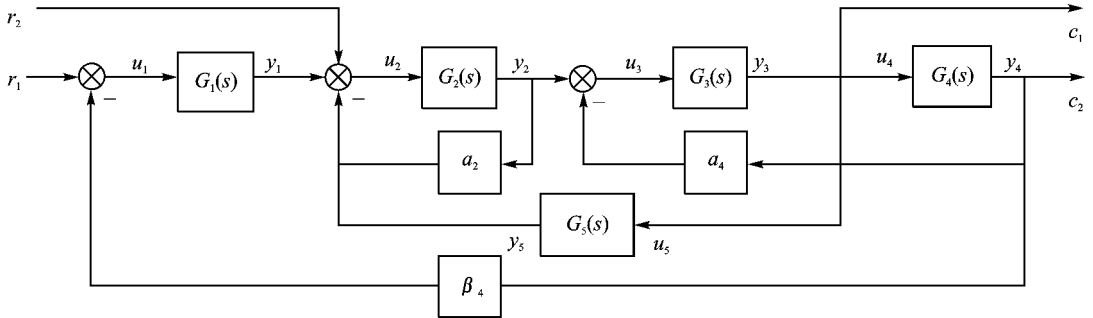


图 3-2 多入多出系统结构图

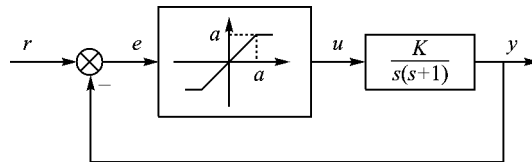


图 3-3 含有非线性环节系统的结构图

函数转化为状态方程,这里主要介绍后两种转换。

1. 微分方程转化为状态方程

设单入单出线性系统的微分方程模型为

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \dots + a_{n-1} \dot{y} + a_n y = u \quad (3-5)$$

选取状态变量为

$$\left. \begin{aligned} x_1 &= y \\ x_2 &= \dot{x}_1 = \dot{y} \\ &\vdots \\ x_n &= \dot{x}_{n-1} = y^{(n-1)} \end{aligned} \right\} \quad (3-6)$$

则式(3-5)可写成下列微分方程组:

$$\left. \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= -a_n x_1 - a_{n-1} x_2 - \dots - a_1 x_n + u \end{aligned} \right\} \quad (3-7)$$

将式(3-7)写成矩阵形式,可得状态方程

$$\dot{X} = AX + Bu \quad (3-8)$$

式中



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{pmatrix}_{n \times n}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}_{n \times 1} \quad (3-9)$$

由状态变量的选取可得系统的输出方程为

$$\mathbf{y} = \mathbf{C}\mathbf{X} \quad (3-10)$$

式中  $\mathbf{C} = [1 \ 0 \ 0 \ \cdots \ 0]_{1 \times n}$  (3-11)

**例 3.2.1** 设某系统的微分方程

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_{n-1} \dot{y} + a_n y = b_0 u^{(n)} + b_1 u^{(n-1)} + \cdots + b_{n-1} \dot{u} + b_n u \quad (3-12)$$

试求其状态空间表达式。

该系统的微分方程中包含了输入函数的导数项,此时不能将系统的输出及其导数作为状态变量。在这种情况下,选取状态变量一般有两种形式,下面分别按这两种形式进行变换。

解法一:选状态变量

$$\left. \begin{aligned} x_1 &= y - b_0 u \\ x_2 &= \dot{x}_1 - h_1 u \\ x_3 &= \dot{x}_2 - h_2 u \\ &\vdots \\ x_n &= \dot{x}_{n-1} - h_{n-1} u \end{aligned} \right\} \quad (3-13)$$

式中

$$\left. \begin{aligned} h_1 &= b_1 - a_1 b_0 \\ h_2 &= b_2 - a_2 b_0 - a_1 h_1 \\ h_3 &= b_3 - a_3 b_0 - a_2 h_1 - a_1 h_2 \\ &\vdots \\ h_n &= b_n - a_n b_0 - a_{n-1} h_1 - a_{n-2} h_2 - \cdots - a_2 h_{n-2} - a_1 h_{n-1} \end{aligned} \right\} \quad (3-14)$$

于是状态方程可写为

$$\left. \begin{aligned} \dot{x}_1 &= x_2 + h_1 u \\ \dot{x}_2 &= x_3 + h_2 u \\ &\vdots \\ \dot{x}_{n-1} &= x_n + h_{n-1} u \\ \dot{x}_n &= -a_n x_1 - a_{n-1} x_2 - \cdots - a_1 x_n + h_n u \end{aligned} \right\} \quad (3-15)$$

输出为  $y = x_1 + b_0 u$  (3-16)

令



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{pmatrix}_{n \times n}, \quad \mathbf{B} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-1} \\ h_n \end{pmatrix}_{n \times 1} \quad (3-17)$$

$$\mathbf{C} = [1 \ 0 \ 0 \ \cdots \ 0]_{1 \times n}, \quad \mathbf{D} = [b_0] \quad (3-18)$$

则可写出状态空间表达式的矩阵形式

$$\left. \begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}\mathbf{X} + \mathbf{D}u \end{aligned} \right\} \quad (3-19)$$

解法二:将微分方程做以下等效变换,并引入新的中间变量  $z$ :

$$z^{(n)}(t) + a_1 z^{(n-1)}(t) + \cdots + a_{n-1} \dot{z} + a_n z = u \quad (3-20)$$

$$y = b_0 z^{(n)} + b_1 z^{(n-1)} + \cdots + b_{n-1} \dot{z} + b_n z \quad (3-21)$$

式(3-20)描述了输入与中间变量的关系,式(3-21)描述了中间变量与输出的关系。

对式(3-20),选取状态变量

$$\left. \begin{aligned} x_1 &= z \\ x_2 &= \dot{x}_1 = \dot{z} \\ &\vdots \\ x_n &= \dot{x}_{n-1} = z^{(n-1)} \end{aligned} \right\} \quad (3-22)$$

则可求得相应的状态方程为

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} u \quad (3-23)$$

根据式(3-21)和式(3-22)可写出输出方程

$$\mathbf{y} = b_0 \begin{pmatrix} -a_n & -a_{n-1} & \cdots & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + b_0 u + \begin{pmatrix} -b_n & -b_{n-1} & \cdots & -b_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (3-24)$$

考虑到多数系统满足  $b_0=0$ ,则输出方程变为

$$\mathbf{y} = \begin{pmatrix} -b_n & -b_{n-1} & \cdots & -b_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (3-25)$$



由本例可知,在微分方程转化为状态空间表达式的过程中,对同一系统,状态变量的选取不是唯一的,所以得到状态空间表达式也不是唯一的。

例 3.2.2 设描述某多入多出线性定常系统的方块图如图 3-4 所示,并设  $x_1, x_2, \dots, x_n$  系统状态变量。



图 3-4 多入多出线性定常系统

描述被控对象状态与输入之间关系的微分方程组如下式,式中的系数均为常数。

$$\left. \begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_{11}u_1 + b_{12}u_2 + \dots + b_{1r}u_r \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_{21}u_1 + b_{22}u_2 + \dots + b_{2r}u_r \\ &\vdots \\ \dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_{n1}u_1 + b_{n2}u_2 + \dots + b_{nr}u_r \end{aligned} \right\} \quad (3-26)$$

令 
$$\mathbf{X} = (x_1, x_2, \dots, x_n)^T \quad (3-27)$$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}_{n \times n} \quad (3-28)$$

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1r} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nr} \end{pmatrix}_{n \times r} \quad (3-29)$$

令输入向量 
$$\mathbf{u} = (u_1, u_2, \dots, u_r)^T \quad (3-30)$$

则可写出这个 MIMO 系统的状态方程

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u} \quad (3-31)$$

设系统的输出用下面的方程组表示

$$\left. \begin{aligned} y_1 &= c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n \\ y_2 &= c_{21}x_1 + c_{22}x_2 + \dots + c_{2n}x_n \\ &\vdots \\ y_m &= c_{m1}x_1 + c_{m2}x_2 + \dots + c_{mn}x_n \end{aligned} \right\} \quad (3-32)$$

令输出向量 
$$\mathbf{y} = (y_1, y_2, \dots, y_m)^T \quad (3-33)$$

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1n} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & c_{m3} & \cdots & c_{mn} \end{pmatrix}_{m \times n} \quad (3-34)$$

则可写出输出方程为



$$y = CX \quad (3-35)$$

## 2. 传递函数转化为状态方程

由系统的传递函数或传递函数阵求相应的状态方程,这一过程称为实现问题,同样,其实现不是唯一的。

设系统传递函数如下:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \quad (3-36)$$

引入中间变量,将式(3-36)改写成

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Y(s)}{Z(s)} \cdot \frac{Z(s)}{U(s)} \quad (3-37)$$

式中

$$\frac{Z(s)}{U(s)} = \frac{1}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \quad (3-38)$$

$$\frac{Y(s)}{Z(s)} = b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_{n-1} s + b_n \quad (3-39)$$

取状态变量

$$\left. \begin{aligned} x_1 &= z \\ x_2 &= \dot{x}_1 = \dot{z} \\ &\vdots \\ x_n &= \dot{x}_{n-1} = \dot{z}^{(n-1)} \end{aligned} \right\} \quad (3-40)$$

可得状态空间表达式

$$\dot{\mathbf{X}} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_n & -a_{n-1} & \cdots & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \mathbf{u} \quad (3-41)$$

$$\mathbf{y} = (-b_n \quad -b_{n-1} \quad \cdots \quad -b_1) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (3-42)$$

这种状态空间的实现称为可控标准型。

若按如下方式选取状态变量



$$\left. \begin{aligned} x_n &= y \\ x_{n-1} &= \dot{x}_n + a_1 x_n - b_1 u \\ x_{n-2} &= \dot{x}_{n-1} + a_2 x_n - b_2 u \\ &\vdots \\ x_1 &= \dot{x}_2 + a_{n-1} x_n - b_{n-1} u \\ 0 &= \dot{x}_1 + a_n x_n - b_n u \end{aligned} \right\} \quad (3-43)$$

则可得状态空间实现的可观标准型

$$\dot{\mathbf{X}} = \begin{pmatrix} 0 & 0 & \cdots & -a_n \\ 1 & 0 & \cdots & -a_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} -b_n \\ -b_{n-1} \\ \vdots \\ -b_1 \end{pmatrix} \mathbf{u} \quad (3-44)$$

$$\mathbf{y} = (0 \quad 0 \quad \cdots \quad 1) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (3-45)$$

若传递函数的特征方程

$$s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n = 0 \quad (3-46)$$

有  $n$  个互异特征值  $\lambda_1, \lambda_2, \cdots, \lambda_n$ , 将传递函数展开成部分分式

$$G(s) = \frac{c_1}{s - \lambda_1} + \frac{c_2}{s - \lambda_2} + \cdots + \frac{c_n}{s - \lambda_n} \quad (3-47)$$

取状态变量

$$\frac{x_i(s)}{u(s)} = \frac{1}{s - \lambda_i}, \quad i = 1, 2, \cdots, n \quad (3-48)$$

所得的状态空间为对角标准型

$$\left. \begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{X} \end{aligned} \right\} \quad (3-49)$$

式中,  $\mathbf{A} = \text{diag}[\lambda_1, \lambda_2, \cdots, \lambda_n]$ ,  $\mathbf{B} = [1, 1, \cdots, 1]^T$ ,  $\mathbf{C} = [c_1, c_2, \cdots, c_n]$ 。

若传递函数的特征方程具有重根, 则会得到约当标准型, 具体实现方法这里不再赘述, 读者可参阅有关资料。

模型转换还有其他一些情况, 如微分方程转化为传递函数、状态方程转化为传递函数、结构图转化为状态方程等。

在一些仿真软件中, 提供了模型转换的功能函数, 可以直接调用, 关于这些函数的使用方法可查阅相关软件的说明。



### 3.3 离散时间系统数学模型

连续系统中的信号是连续型的时间函数,而离散时间系统中的一部分信号是离散型的时间函数。常用的离散时间系统数学模型有差分方程、 $z$  传递函数、离散状态空间表达式、离散系统结构图 4 种。

#### 3.3.1 差分方程

离散时间系统的差分方程一般具有如下形式:

$$y(k+n) = a_1 y(k+n-1) + a_2 y(k+n-2) + \cdots + a_n y(k) + b_1 u(k+m) + b_2 u(k+m-1) + b_m u(k) \quad (3-50)$$

式中,  $u(i)$ 、 $y(i)$  分别为第  $i$  个采样时刻系统的输入和输出,  $m \leq n$ 。

差分方程是时域模型,描述系统的输入、输出特性。

#### 3.3.2 $z$ 传递函数

$z$  传递函数也称脉冲传递函数,是分析离散时间系统的重要工具。对式(3-50)进行  $z$  变换,可得  $z$  传递函数

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \cdots + b_{m-1} z + b_m}{z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n} \quad (3-51)$$

#### 3.3.3 离散状态空间表达式

由定常差分方程求取线性离散系统状态方程多过程与由微分方程求取状态方程的过程相似,一般的状态空间表达式为

$$\left. \begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{Y}(k) &= \mathbf{C}\mathbf{X}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \right\} \quad (3-52)$$

式中,  $\mathbf{X} = [x_1, x_2, \cdots, x_n]^T$ , 为  $n$  维状态向量,  $\mathbf{u}$  为  $r$  维输入向量,  $\mathbf{y}$  为  $m$  维输出向量;  $\mathbf{A}_{n \times n}$  为状态矩阵,  $\mathbf{B}_{n \times r}$  为输入矩阵,  $\mathbf{C}_{m \times n}$  为输出矩阵,  $\mathbf{D}_{m \times r}$  为直传矩阵。

由于引入了状态变量,所以离散状态空间表达式不仅能描述系统的输入、输出关系,而且能够描述系统的内部状态特性。

#### 3.3.4 离散系统结构图

离散系统结构图表示与连续系统相似,如图 3-5 所示,图中各环节均为  $z$  传递函数。

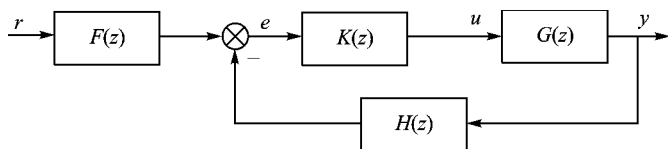


图 3-5 离散系统结构图

### 3.3.5 关于采样系统

采样系统是一种常见的离散时间系统。当控制器由数字计算机实现时,控制系统就成为一个采样系统。采样系统的原理组成如图 3-6 至图 3-8 所示,系统中包括数字控制器、采样器和保持器。

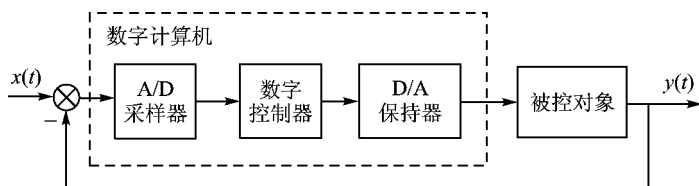


图 3-6 采样系统结构图(之一)

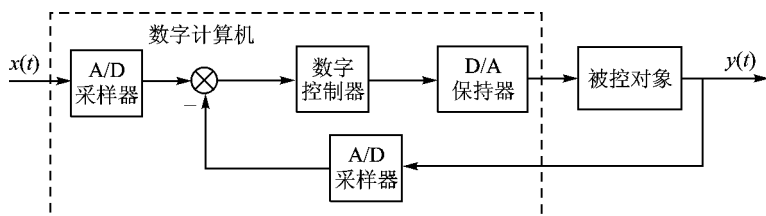


图 3-7 采样系统结构图(之二)

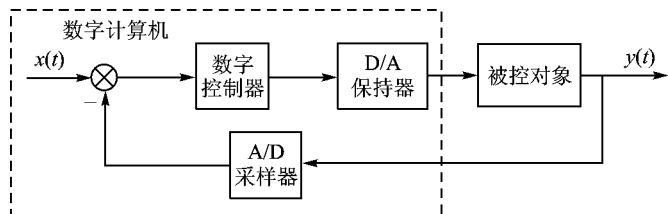


图 3-8 采样系统结构图(之三)

离散系统的各个数学模型之间也可以进行转换,如差分方程转换为  $z$  传递函数、 $z$  传递函数转换为差分方程、差分方程或  $z$  传递函数转换为离散状态方程,这些转换的方法与连续系统的模型转换相似,这里不再赘述,有兴趣的读者可查阅相关资料。



### 3.4 随机系统数学模型

本章前两节介绍的数学模型都是用来描述确定性系统的。所谓确定性系统,指的是系统参数、系统输入均为已知、确定的系统,随机系统则相反,其参数、输入均为随机量。严格地说,实际的系统都是随机系统,只是在很多情况下,可以将其简化为确定性系统进行研究。

有些时候,如果不能用一个系统的一次行为代表一类系统的多次运行性能,那么就必须要将该系统作为一个随机系统来研究。粗略地说,随机系统的数学模型是在确定性系统的数学模型中考虑了随机因素的影响。

#### 3.4.1 连续随机系统描述方法

状态空间表达式是一种常用的确定性系统时域模型,类似地,连续的随机系统状态空间表达式为

$$\left. \begin{aligned} \dot{\mathbf{X}}(t) &= g[\mathbf{X}(t), \mathbf{u}(t), \mathbf{W}(t), t] \\ \mathbf{y}(t) &= h[\mathbf{X}(t), \mathbf{u}(t), \mathbf{m}(t), \mathbf{V}(t), t] \end{aligned} \right\} \quad (3-53)$$

对线性系统,则有

$$\left. \begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{A}(t)\mathbf{X}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}(t)\mathbf{X}(t) \end{aligned} \right\} \quad (3-54)$$

式中,  $\mathbf{X}(t) = [x_1, x_2, \dots, x_n]^T$ , 为  $n$  维状态向量,  $\mathbf{u}(t)$  为  $r$  维确定性输入向量,  $\mathbf{w}(t)$  为  $k$  维动态干扰向量,  $\mathbf{y}(t)$  为  $m$  维输出向量;  $\mathbf{A}(t)_{n \times n}$  为状态矩阵,  $\mathbf{B}(t)_{n \times r}$  为输入矩阵,  $\mathbf{G}(t)_{n \times k}$  为干扰加权矩阵,  $\mathbf{C}(t)_{m \times n}$  为输出矩阵。一般地, 4 个系数矩阵中会含有一个或几个随机参数, 有些随机系统中, 其输入和状态变量的初值为随机量。

若系统为线性定常系统, 则可进一步写为以下形式, 这类随机系统的输入或状态变量的初值为随机量

$$\left. \begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{A}\mathbf{X}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{X}(t) \end{aligned} \right\} \quad (3-55)$$

#### 3.4.2 离散随机系统描述方法

离散随机系统的状态空间表达式一般为

$$\left. \begin{aligned} \mathbf{X}(k+1) &= g[\mathbf{X}(k), \mathbf{u}(k), \mathbf{w}(k), k] \\ \mathbf{y}(k) &= h[\mathbf{X}(k), \mathbf{u}(k), \mathbf{m}(k), \mathbf{V}(k), k] \end{aligned} \right\} \quad (3-56)$$

对于线性系统, 模型可写为

$$\left. \begin{aligned} \mathbf{X}(k+1) &= \Phi(k+1, k)\mathbf{X}(k) + \Psi(k+1, k)\mathbf{u}(k) + \Gamma(k+1, k)\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{H}(k)\mathbf{X}(k) + \mathbf{m}(k) + \mathbf{V}(k) \end{aligned} \right\} \quad (3-57)$$



对线性定常系统,模型为

$$\left. \begin{aligned} \mathbf{X}(k+1) &= \Phi \mathbf{X}(k) + \Psi \mathbf{u}(k) + \Gamma \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{H} \mathbf{X}(k) + \mathbf{m}(k) + \mathbf{V}(k) \end{aligned} \right\} \quad (3-58)$$

式中,  $\mathbf{X}(k) = [x_1, x_2, \dots, x_n]^T$ , 为  $n$  维状态向量,  $\mathbf{u}(k)$  为  $r$  维确定性输入向量,  $\mathbf{w}(k)$  为  $k$  维动态干扰向量,  $\mathbf{y}(k)$  为  $m$  维输出向量;  $\Phi_{n \times n}$  为状态矩阵,  $\Psi_{n \times r}$  为输入矩阵,  $\Gamma_{n \times k}$  为干扰加权矩阵,  $\mathbf{H}_{m \times n}$  为输出矩阵。

### 3.4.3 随机系统建模/仿真中的专门问题

与确定性系统仿真类似,随机系统仿真也是模拟系统的动态过程,而模型是仿真的基础,所以对随机系统仿真需要首先解决其建模问题,随机系统建模的专门问题主要有随机信号的测量、数据处理、数学模型的抽象等。

随机系统仿真的专门问题主要有随机数的产生、随机数概率分布的转变、随机数功率谱密度的转变、系统随机参数误差的产生、系统随机初值的产生、系统随机干扰的产生等。

这些专门问题的研究及相关的解决方法可参阅有关文献。

本章介绍了系统仿真中常用的数学模型,包括连续系统数学模型、离散系统数学模型和随机系统数学模型,并介绍了不同模型之间的转换方法。

## 习 题

3.1 根据下列系统的传递函数写出其状态空间表达式。

$$(1) G(s) = \frac{4s^2 + 17s + 16}{s^3 + 7s^2 + 16s + 12}$$

$$(2) G(s) = \frac{1}{s(s+1)(s+8)}$$

$$(3) G(s) = \frac{s+1}{s(s^2+7s+1)}$$

3.2 根据下面的微分方程写出其传递函数及状态空间表达式。

$$(1) \ddot{y}(t) + 2\dot{y}(t) + y(t) = U$$

$$(2) \ddot{y}(t) + 5\dot{y}(t) + \dot{y}(t) + 2y(t) = \dot{u}(t) + 2u(t)$$

$$(3) \ddot{y} + 3\dot{y}(t) + 2\dot{y}(t) + 2y(t) = 0$$

3.3 设离散系统的差分方程如下,试写出其  $z$  传递函数及离散状态空间表达式。

$$(1) y(k+2) + 3y(k+1) + 2y(k) = u(k)$$

$$(2) y(k+3) + 3y(k+2) + 2y(k+1) + y(k) = u(k+2) + 2u(k+1)$$

$$(3) y(k+n) = a_1 y(k+n-1) + a_2 y(k+n-2) + \dots + a_n y(k) + b_1 u(k+m) + b_2 u(k+m-1) + \dots + b_m u(k)$$



### 3.4 什么是随机系统？随机系统建模/仿真中有哪些专门问题？

#### 参考文献

- [1] 姜玉宪主编. 控制系统仿真. 北京:北京航空航天大学出版社,1998.
- [2] 李友善主编. 自动控制原理(上、下册). 北京:国防工业出版社,1981.
- [3] 康凤举主编. 现代仿真技术与应用. 北京:国防工业出版社,2001.
- [4] 李树英,许增茂. 随机系统的滤波与控制. 北京:国防工业出版社,1991.

## 第4章 连续系统仿真的数值积分法

积分理论及其应用是数学上重要的中心课题之一。概括地说,数值积分就是研究如何求出一个积分的具体数值。目前,数值积分的理论和方法都已经比较成熟,计算精度也比较高。

本章介绍利用数值积分法对连续系统进行仿真的基本原理和方法,主要包括:欧拉(Euler)方法、龙格-库塔(Runge-Kutta)法、亚当姆斯(Adams)法和吉尔(Gear)法,同时引出一些关于数值积分法的基本概念并对选择计算方法和计算步长的原则进行讨论。

### 4.1 数值积分法原理

在数学上,数值积分法的应用之一是求解微分方程的初值问题。对于连续系统来说,微分方程(组)是常用的数学模型之一,对系统进行仿真的过程即是从初始条件出发,计算系统在一定输入信号作用下的输出响应的过程,这在本质上是微分方程的初值问题。连续系统仿真的数值积分法就是利用数值积分法对常微分方程(组)建立差分方程,根据差分方程编制程序进行计算,求其数值解。

设一阶常微分方程及其初值为

$$\dot{y} = f(t, y), \quad y(t_0) = y_0 \quad (4-1)$$

对其两端求积分可得

$$y(t) - y(t_0) = \int_{t_0}^t f(\tau, y) d\tau \quad (4-2)$$

即

$$y(t) = y(t_0) + \int_{t_0}^t f(\tau, y) d\tau \quad (4-3)$$

当  $t = t_{n+1}$  时,式(4-3)变为

$$y(t_{n+1}) = y(t_0) + \int_{t_0}^{t_{n+1}} f(\tau, y) d\tau \quad (4-4)$$

将式(4-4)的积分项拆开写成两项,

$$y(t_{n+1}) = y(t_0) + \int_{t_0}^{t_n} f(\tau, y) d\tau + \int_{t_n}^{t_{n+1}} f(\tau, y) d\tau \quad (4-5)$$

对比式(4-4)可知式(4-5)等号右侧的前两项即为  $y(t_n)$ ,于是有

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, y) d\tau \quad (4-6)$$

若用近似值  $y_{n+1}$ 、 $y_n$ 、 $Q_n$  代替上述各项,则有



$$y_{n+1} = y_n + Q_n$$

这样,从  $t_0$  时刻出发,可步进地求出  $t_1, t_2, \dots, t_n$  时刻的近似值。

用不同的方法求积分值  $Q_n$  就形成了不同的数值积分法,每种方法的计算精度、计算速度和稳定区域都不相同。本章从最简单的欧拉方法入手,介绍仿真中常用的几种数值积分法。

## 4.2 欧拉方法

### 4.2.1 欧拉法

欧拉法是一种最简单的数值积分法。该方法精度较差,在实际仿真中很少使用。但由于其推导简单,具有较明确的几何意义,所以常用于说明构造数值积分法一般计算公式的基本思想和一些相关概念。

对微分方程(4-1)两端求积分

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(\tau, y) d\tau \quad (4-7)$$

若积分区间  $[t_n, t_{n+1}]$  足够小,则可认为  $f(t, y)$  近似地等于常数  $f(t_n, y_n)$ ,于是可用矩形面积近似地代替该区间内曲线下的面积,即

$$y_{n+1} = y_n + hf(t_n, y_n) \quad (4-8)$$

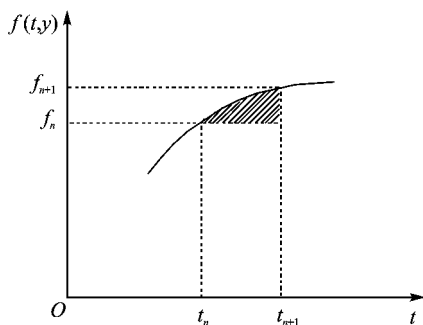


图 4-1 欧拉法的几何意义

式(4-8)即为欧拉法的计算公式,其中  $h = t_{n+1} - t_n$  为计算步长。图 4-1 表示了欧拉法的几何意义。图中的阴影部分即为矩形面积代替曲线下面积所产生的误差,可见,欧拉法的误差较大。

**例 4.2.1** 已知微分方程为  $\dot{y} = -y^2$ , 初值为  $y(0) = 1$ , 步长  $h = 0.1$ , 用欧拉法求其数值解,并与解析解相比较。

解: 该微分方程的解析解为  $y = \frac{1}{1+t}$ , 数值解的

计算公式为  $y_{n+1} = y_n + h(-y^2)$ 。利用仿真软件 MATLAB 计算出解析解与数值解,如图 4-2 所示。图中实线为数值解,虚线为解析解。图 4-3 为二者的误差曲线。

### 4.2.2 改进欧拉法

用矩形面积代替曲线积分存在较大误差;如果用梯形面积代替曲线积分则可提高计算精度,减小误差。这就是改进欧拉法的基本思想。梯形法的计算公式为

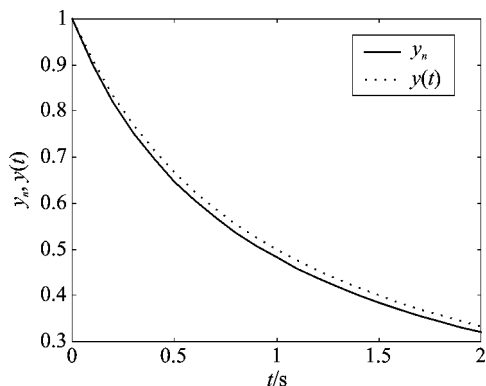


图 4-2 数值解与解析解

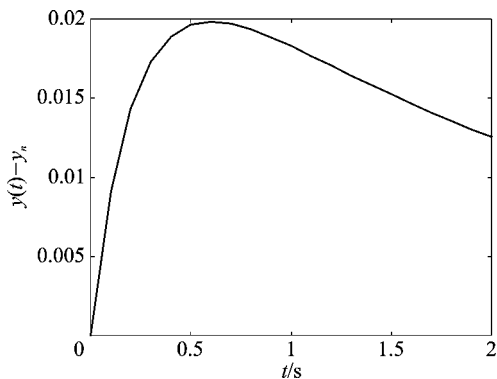


图 4-3 欧拉法的误差

$$y_{n+1} = y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1})] \quad (4-9)$$

由于公式的右端有待求量,所以不能自启动,需要与其他方法结合使用。若每次先用欧拉法求出  $y_{n+1}$  的预估值  $y_{n+1}^P$ ,再用梯形公式求出其校正值  $y_{n+1}^C$ ,则有以下二式:

$$\text{预估} \quad y_{n+1}^P = y_n + hf(t_n, y_n) \quad (4-10)$$

$$\text{校正} \quad y_{n+1}^C = y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1}^P)] \quad (4-11)$$

这就是改进的欧拉法公式。

**例 4.2.2** 对例 4.2.1 中微分方程采用改进欧拉法求其数值解,步长仍为 0.1。

**解:** 图 4-4 为数值解  $y_n$  与解析解  $y(t)$ ;图 4-5 为二者误差,为方便比较,在图中同时用虚线画出了欧拉法的误差。可见,改进欧拉法的精度明显高于欧拉法。

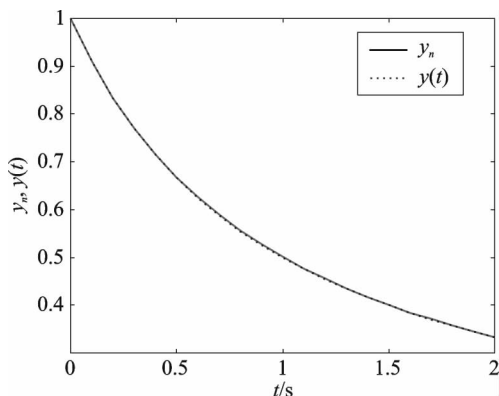


图 4-4 改进欧拉法的数值解

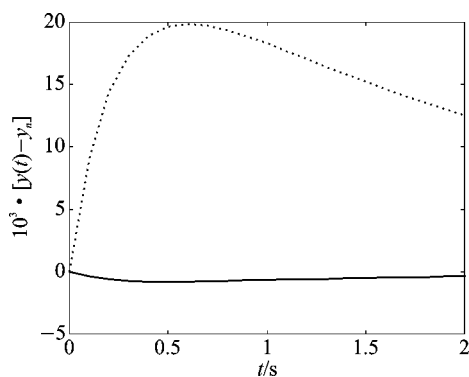


图 4-5 改进欧拉法的计算误差



改进欧拉法的计算公式本质上采用了  $t_n$  和  $t_{n+1}$  两点斜率的平均值,从而提高了计算精度。为了得到更高的精度,可将校正公式反复迭代,得到如下计算公式:

$$\begin{aligned} y_{n+1}^{(0)} &= h_n + hf(t_n, y_n) \\ y_{n+1}^{(1)} &= y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(0)})] \\ y_{n+1}^{(s+1)} &= y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(s)})] \end{aligned} \quad (4-12)$$

**例 4.2.3** 令  $s=1$ ,即将校正公式迭代两次,计算例 4.2.1 中的微分方程数值解,步长仍为 0.1。

图 4-6 所示为数值解  $y_n$  与解析解  $y(t)$  的曲线;图 4-7 为  $y(t) - y_n$ ,图中的虚线为迭代一次时改进欧拉法的计算误差。

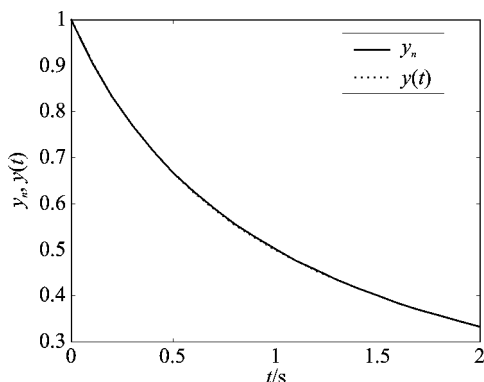


图 4-6 改进欧拉法的数值解( $s=1$ )



图 4-7 改进欧拉法的计算误差

可见,多迭代一次会使误差减小,但并不明显,而此时的计算量却有较大增加,故在使用时应综合考虑计算量和精度等因素。在很多工程问题的求解中,为了简化计算,一般认为迭代一次就已求得了近似结果,即用式(4-10)、(4-11)。

### 4.2.3 几个基本概念

#### 1. 单步法与多步法

数值积分法采用的都是递推公式,是步进式的。若求解  $t_{n+1}$  时刻的值只用到  $t_n$  时刻的值,则称这种方法为单步法,这是一种能自启动的算法。显然,欧拉法是单步法。

若求解  $t_{n+1}$  时刻的值不仅用到  $t_n$  时刻的值,还用到  $t_{n-1}$ 、 $t_{n-2}$  等时刻的值,则称这种方法为多步法。多步法不能自启动。



## 2. 显式与隐式

若求解  $y_{n+1}$  时所用到的数据均已求出,则称这种公式为显式公式。显然,欧拉法的计算公式为显式公式。

若公式右端有待求量  $y_{n+1}$ ,则称这种公式为隐式公式。隐式公式不能自启动,需要与显式公式配合使用,形成预估-校正法,如改进欧拉法。

可见,单步法和显式方法在计算上比多步法和隐式公式方便,但有时为了满足精度、稳定性等方面的要求,必须采用隐式方法。

## 3. 截断误差

一般常用泰勒级数作为工具分析数值积分公式的精度。为简化分析,设前一步所得结果  $y_n$  是精确的,将  $t_{n+1}$  处的精确解展成泰勒级数得

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + h\dot{y}(t_n) + \frac{h^2}{2!}\ddot{y}(t_n) + \cdots + \frac{h^r}{r!}y^{(r)}(t_n) + O(h^{r+1}) \quad (4-13)$$

可见,欧拉法是从精确解中取前两项来近似计算  $y_{n+1}$ ,后面的所有项均被省略。每一步由这种方法引入的误差称为局部截断误差,简称截断误差。它是数值解与解析解之间的误差,所以又称局部离散误差。显然,欧拉法的截断误差为

$$y(t_{n+1}) - y_{n+1} = O(h^2) \quad (4-14)$$

截断误差与步长  $h$  的幂成正比,不同的数值积分法有不同的截断误差。一般,若截断误差为  $O(h^{r+1})$ ,则称方法是  $r$  阶的。所以方法的阶数可以作为衡量方法精度的一个重要标志。由此可见,欧拉法是 1 阶精度;改进欧拉法由于采用了平均斜率,相当于取了泰勒级数的前 3 项,故为 2 阶精度,其截断误差为  $O(h^3)$ 。

## 4. 舍入误差

由于计算机的字长有限,数值不能表示得完全精确,在计算过程中,不可避免地会产生舍入误差。舍入误差与步长  $h$  成反比,如果计算步长小,计算次数多,则舍入误差就大。产生舍入误差的因素较多,除了与计算机字长有关,还与计算机所使用的数字系统、数的运算次序以及计算函数  $f(t, y)$  所用子程序的精度等有关。

## 5. 数值解的稳定性

连续系统的数字仿真实际上是将给定的微分方程变换为差分方程,然后从初值开始,一步一步进行迭代运算,因此在采用数值积分法求解稳定的微分方程时,应保持原系统的稳定性。但是,在计算机进行计算时,初始数据的误差及计算过程的舍入误差对后面的计算结果将产生影响;而且,如果计算步长选择得不合理,则有可能使仿真出现不稳定的结果。所以,对每一种数值积分法都有必要讨论其稳定性问题。

差分方程的解与微分方程的解类似,均可分为通解和特解两部分。与稳定性有关的只是方程的通解,取决于差分方程的特征根是否满足稳定性条件。



例 4.2.4 为了讨论欧拉法的稳定性,研究检验方程(test equation) $\dot{y}=\lambda y$  的计算公式

$$y_{n+1} = y_n + hf_n = y_n + \lambda y_n = (1 + \lambda h)y_n \quad (4-15)$$

式中, $\lambda$  为方程的特征根,初值为  $y(t_0) = y_0$ 。显然,要使该差分方程稳定,应使其特征根位于单位圆内,即满足

$$|z| = |1 + \lambda h| < 1 \quad (4-16)$$

由此可得

$$-2 < h\lambda < 0 \quad (4-17)$$

可见,为了使数字仿真是稳定的,应当对计算步长加以限制。另外,数值解的稳定性还与微分方程(即系统)的特性及数值积分的方法有关。

分析欧拉法的稳定性和误差的基本思想,也适用于其他数值积分法。

## 4.3 龙格-库塔法

### 4.3.1 龙格-库塔法的基本原理

由前面的分析可知,将泰勒级数多取几项能提高截断误差的阶次,可以提高计算精度。但直接采用泰勒展式就要计算  $y$  的高阶导数,使用不方便。Runge 和 Kutta 两人先后提出了间接利用泰勒展式的方法,即用几个点上  $y$  的一阶导数  $f$  的线性组合代替  $y$  的高阶导数,再按照泰勒公式展开确定其中的系数。这样既可避免计算高阶导数,又可提高数值积分法的精度。这就是龙格-库塔法的基本思想。

龙格-库塔法有显式、隐式、半隐式等形式,从计算步长角度又可分为固定步长和变步长形式。本节主要介绍显式龙格-库塔法。

将  $y_{n+1}$  在  $t_n$  点展为泰勒级数,只保留  $h^2$  项,有

$$y_{n+1} = y_n + f(t_n, y_n)h + \frac{1}{2} \left( \frac{\partial f}{\partial y} \frac{dy}{dt} + \frac{\partial f}{\partial t} \right) \Big|_{t_n} h^2 + O(h^3) \quad (4-18)$$

不失一般性。设上式可写成如下形式:

$$y_{n+1} = y_n + (a_1 k_1 + a_2 k_2)h \quad (4-19)$$

式中, $k_1 = f(t_n, y_n)$ ,  $k_2 = f(t_n + b_1 h, y_n + b_2 k_1 h)$ ,  $a_1, a_2$  为加权系数。将式(4-19)中的  $k_2$  展为泰勒级数,只保留  $h$  项

$$k_2 \approx f(t_n, y_n) + \left( b_1 \frac{\partial f}{\partial t} + b_2 k_1 \frac{\partial f}{\partial y} \right) \Big|_{t_n} h \quad (4-20)$$

将  $k_1$  和式(4-20)代入式(4-19)得

$$y_1 = y_0 + a_1 hf(t_0, y_0) + a_2 h \left[ f(t_0, y_0) + \left( b_1 \frac{\partial f}{\partial t} + b_2 k_1 \frac{\partial f}{\partial y} \right) \Big|_{t_0} h \right] \quad (4-21)$$

比较(4-18)与(4-21)的同类项系数,可得 3 个方程:



$$\left. \begin{aligned} a_1 + a_2 &= 1 \\ a_2 b_1 &= \frac{1}{2} \\ a_2 b_2 &= \frac{1}{2} \end{aligned} \right\} \quad (4-22)$$

令  $a_1 = a_2$ , 可解得:  $a_1 = a_2 = \frac{1}{2}$ ,  $b_1 = b_2 = 1$ , 所构造的计算公式为

$$\left. \begin{aligned} y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + h, y_n + k_1 h) \end{aligned} \right\} \quad (4-23)$$

该公式的截断误差为  $O(h^3)$ , 称二阶龙格-库塔法(简称 RK-2)。

按照这一思路, 可构造出截断误差阶次更高的龙格-库塔法。

### 4.3.2 固定步长的龙格-库塔法

#### 1. 一般公式

固定步长的龙格-库塔法的一般公式为

$$\left. \begin{aligned} y_{n+1} &= y_n + h \sum_{i=1}^r \omega_i k_i \\ k_i &= f(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j) \end{aligned} \right\} \quad (4-24)$$

式中,  $r$  为阶数,  $c_1 = 0$ , 其余的  $c_i, a_{ij}$  均为与阶数有关的系数。

$r=1$  时,  $y_{n+1} = y_n + \omega_1 k_1 h$ ,  $k_1 = f(t_n, y_n)$ ,  $\omega_1 = 1$ , 为欧拉法;

$r=2$  时,  $y_{n+1} = y_n + \omega_1 k_1 h + \omega_2 k_2 h$ ,  $k_1 = f(t_n, y_n)$ ,  $k_2 = f(t_n + c_2 h, y_n + a_{21} k_1 h)$ ,

$\omega_1 = \omega_2 = \frac{1}{2}$ ,  $c_2 = a_{21} = 1$ , 为改进欧拉法。

#### 2. 4阶龙格-库塔法(简称 RK-4)

当式(4-24)的  $r=4$  时, 可得 4 阶龙格-库塔法, 又称经典龙格-库塔法, 其公式为

$$\left. \begin{aligned} y(t_{n+1}) &\approx y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 3k_3 + k_4) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\ k_4 &= f(t_n + h, y_n + hk_3) \end{aligned} \right\} \quad (4-25)$$



截断误差为  $O(h^5)$ 。该方法是数字仿真中最常用的一种方法,其计算量大,每步需要计算 4 次函数  $f$  的值。由于其计算精度高,在比较不同仿真方法的计算精度时,常以它的计算结果作为标准。

欧拉法、改进欧拉法本质上都是龙格-库塔法,其截断误差分别为  $O(h^2)$ 、 $O(h^3)$ 。从理论上说,可以构造任意高阶的龙格-库塔法;但是精度阶数不与计算函数  $f$  的次数等量增加,其关系如表 4-1 所列。

表 4-1  $f$  的计算次数  $N$  与精度阶数  $r$  的关系

$N$	2	3	4	5	6	7	$\geq 8$
$r$	2	3	4	4	5	6	$N-2$

可见,4 阶龙格-库塔法有其优越性;4 阶以上的龙格-库塔法计算函数  $f$  所需的次数比阶数多,增加了计算量,从而限制了更高阶龙格-库塔法的应用。对于大量的实际问题,4 阶龙格-库塔法已能够满足精度要求。

例 4.3.1 用 RK-4 计算例 4.2.1 中的微分方程,步长为 0.1,并与解析解比较。图 4-8 中所示的实线为数值解,虚线为解析解。图 4-9 所示为二者的误差。比较图 4-9、图 4-3、图 4-5 和图 4-7 可知,4 阶龙格-库塔法的计算精度大大高于欧拉法及改进欧拉法。

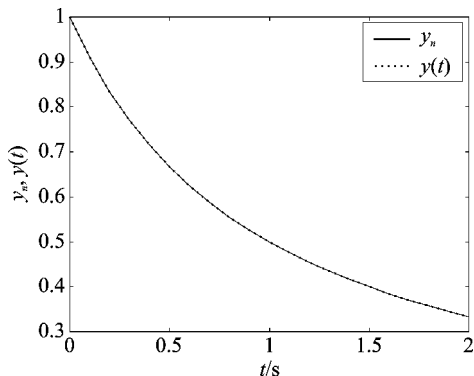


图 4-8 RK-4 的计算结果

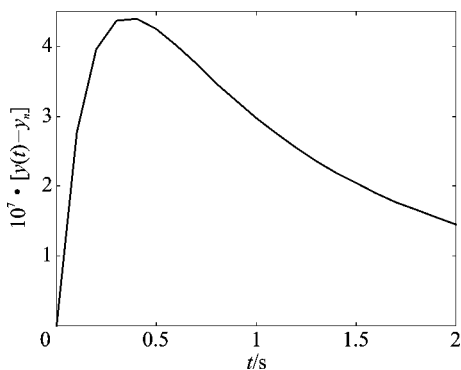


图 4-9 数值解与解析解的误差

### 3. 4 阶龙格-库塔法的向量形式

在上面的推导中,均假设系统是一阶的,用一个一阶微分方程描述。若系统为高阶的,则须用一阶微分方程组表示,此时须用 4 阶龙格-库塔法的向量形式。

设一阶微分方程组为

$$\dot{Y} = F(t, Y) \quad (4-26)$$

式中,  $Y = (y_1, y_2, \dots, y_N)^T$ ,  $F = (f_1(t, Y), f_2(t, Y), \dots, f_N(t, Y))^T$ , 初值为  $Y(t_0) = Y_0$ 。此时 4 阶龙格-库塔法的向量形式为



$$\left. \begin{aligned} \mathbf{Y}_{n+1} &= \mathbf{Y}_n + \frac{h}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4) \\ \mathbf{K}_1 &= \mathbf{F}(t_n, \mathbf{Y}_n) \\ \mathbf{K}_2 &= \mathbf{F}\left(t_n + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_1\right) \\ \mathbf{K}_3 &= \mathbf{F}\left(t_n + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_2\right) \\ \mathbf{K}_4 &= \mathbf{F}(t_n + h, \mathbf{Y}_n + h\mathbf{K}_3) \end{aligned} \right\} \quad (4-27)$$

式中,  $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3, \mathbf{K}_4$  均为向量,

$$\mathbf{K}_1 = \begin{bmatrix} f_1(t_n, \mathbf{Y}_n) \\ \vdots \\ f_N(t_n, \mathbf{Y}_n) \end{bmatrix}, \quad \mathbf{K}_2 = \begin{bmatrix} f_1\left(t_n + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_1\right) \\ \vdots \\ f_N\left(t_n + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_1\right) \end{bmatrix}$$

$$\mathbf{K}_3 = \begin{bmatrix} f_1\left(t_n + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_2\right) \\ \vdots \\ f_N\left(t_n + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_2\right) \end{bmatrix}, \quad \mathbf{K}_4 = \begin{bmatrix} f_1(t_n + h, \mathbf{Y}_n + h\mathbf{K}_3) \\ \vdots \\ f_N(t_n + h, \mathbf{Y}_n + h\mathbf{K}_3) \end{bmatrix} \quad (4-28)$$

### 4.3.3 龙格-库塔法的稳定区

对每一种数值积分法,在应用中所关心的一是其计算精度,二是其稳定性。下面简单分析各阶龙格-库塔法的稳定区,仍采用检验方程  $\dot{y} = \lambda y$  进行讨论。

将  $y_{n+1}$  在  $t_n$  点展为泰勒级数

$$y_{n+1} = y_n + \sum_{i=1}^r \frac{h^i}{i!} y_n^{(i)} + O(h^{r+1}) \quad (4-29)$$

对于  $\dot{y} = \lambda y$  有  $y^{(i)} = \lambda^i y$ , 将其代入式(4-29)得

$$y_{n+1} = y_n \left[ 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \cdots + \frac{(\lambda h)^r}{r!} \right] + O(h^{r+1}) \quad (4-30)$$

该式的稳定条件为

$$\lambda_1 = \left| 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \cdots + \frac{(\lambda h)^r}{r!} \right| < 1 \quad (4-31)$$

根据式(4-31)的稳定条件,可计算出各阶龙格-库塔法的稳定区,如表4-2所列。

在使用龙格-库塔法公式时,选取的步长应使  $h\lambda$  落在稳定区域内。否则,计算时会产生很大的误差,从而得不到稳定的数值解。



表 4-2 各阶龙格-库塔法的稳定区

$r$	$\lambda_1$	$h\lambda$ 所在的实稳定区域
1	$1+\lambda h$	$(-2, 0)$
2	$1+\lambda h+\frac{(\lambda h)^2}{2}$	$(-2, 0)$
3	$1+\lambda h+\frac{(\lambda h)^2}{2}+\frac{(\lambda h)^3}{6}$	$(-2.51, 0)$
4	$1+\lambda h+\frac{(\lambda h)^2}{2}+\frac{(\lambda h)^3}{6}+\frac{(\lambda h)^4}{24}$	$(-2.78, 0)$

例 4.3.2 在步长取 0.1 和 0.2 时,用 RK-4 计算如下两个微分方程的数值解,并与解析解比较。

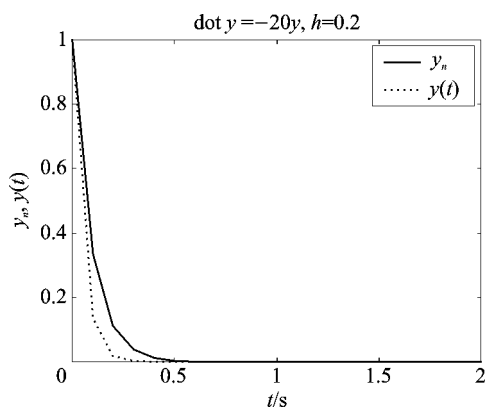
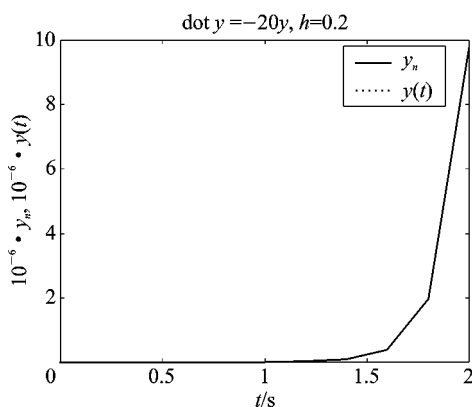
$$s1: \dot{y} = -20y, y(0) = 1$$

$$s2: \dot{y} = -10y, y(0) = 1$$

解: 两个方程的解析解很容易求出,分别是:  $s1: y = e^{-20t}$ ,  $s2: y = e^{-10t}$ 。对方程  $s1$ , 其特征值  $\lambda = -20$ ,  $h = 0.1$  时,  $h\lambda = -2$ ,  $\in (-2.78, 0)$ , 满足稳定条件;  $h = 0.2$  时,  $h\lambda = -4$ ,  $\notin (-2.78, 0)$ , 不满足稳定条件, 数值解将发散。

对方程  $s2$ , 其特征值  $\lambda = -10$ ,  $h = 0.1$  时,  $h\lambda = -1$ ,  $\in (-2.78, 0)$ , 满足稳定条件;  $h = 0.2$  时,  $h\lambda = -2$ ,  $\in (-2.78, 0)$ , 也满足稳定条件。

计算结果如图 4-10 至图 4-13 所示。可见, 当  $h\lambda$  超过稳定区域时, 数值解发散, 如图 4-11 所示; 当  $h\lambda$  接近稳定边界时, 数值解误差增大, 如图 4-10、4-13 所示。

图 4-10 方程  $s1, h=0.1$  的计算结果图 4-11 方程  $s1, h=0.2$  的计算结果

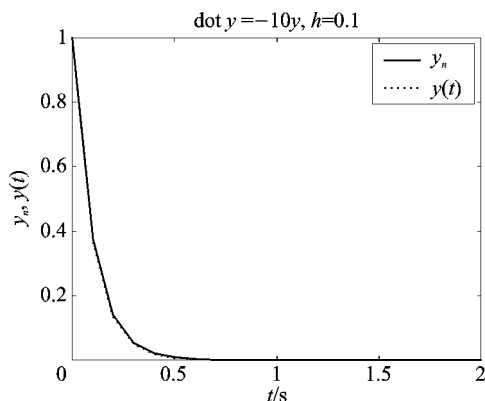


图 4-12 方程 s2, h=0.1 的计算结果

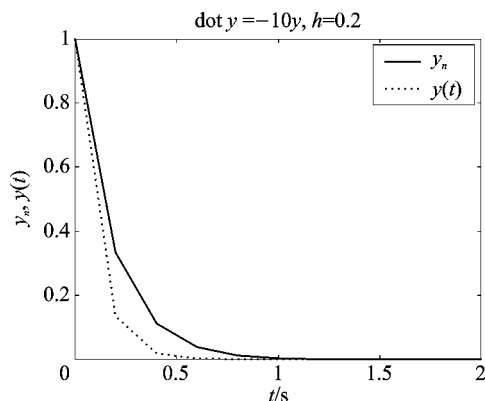


图 4-13 方程 s2, h=0.2 的计算结果

#### 4.3.4 实时龙格-库塔法公式

实时仿真要求仿真模型的运行速度与实际系统运行的速度保持一致。一般的数值积分法难以满足实时仿真的要求,这不仅因为由这些方法所得到的递推公式执行速度较慢,而且是因为这些方法的机理不符合实时仿真的特点。对这些方法的计算公式进行改进,可得到实时仿真用的数值积分公式。

实时二阶龙格-库塔法的公式如下式,其截断误差也为  $O(h^3)$ 。

$$\left. \begin{aligned} y_{n+1} &= y_n + hk_2 \\ k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \end{aligned} \right\} \quad (4-32)$$

实时 4 阶龙格-库塔法的公式如下式,也可用于向量情况。

$$\left. \begin{aligned} y_{n+1} &= y_n + \frac{h}{24}(-k_1 + 15k_2 - 5k_3 + 5k_4 + 10k_5) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{5}, y_n + \frac{h}{5}k_1\right) \\ k_3 &= f\left(t_n + \frac{2h}{5}, y_n + \frac{2h}{5}k_1\right) \\ k_4 &= f\left(t_n + \frac{3h}{5}, y_n - \frac{2h}{5}k_1 + hk_2\right) \\ k_5 &= f\left(t_n + \frac{4h}{5}, y_n + \frac{3h}{10}k_1 + \frac{h}{2}k_4\right) \end{aligned} \right\} \quad (4-33)$$



### 4.3.5 变步长龙格-库塔法

以上介绍的都是固定步长的龙格-库塔法,如何选取合适的步长在实际计算中是很重要的。步长太大不能达到精度要求,步长太小会增加不必要的计算量。为了保证获得预期的精度要求,同时又使计算量尽可能的小,比较好的办法是对步长进行自动调整,即采用变步长方法。

实现步长自动调整的前提是要有一个局部误差估计公式。为了得到每一步的局部误差估计值,可以用两种相差一阶的龙格-库塔法公式,同时计算  $y_{n+1}$ ,取它们的差作为误差估计值。通过巧妙选取龙格-库塔法公式的系数,可以使两个公式中的  $k_i$  相同,从而使一些中间结果对两个龙格-库塔法公式都适用,这样就减小了计算量,使误差估计值可以利用  $k_i$  进行计算。

#### 1. 龙格-库塔-默森法

第一个变步长龙格-库塔法是 Merson 在 1957 年给出的,其误差计算公式是由龙格-库塔-默森(Runge-Kutta-Merson)的 3 阶计算公式与 4 阶计算公式之差求得的。其 4 阶计算公式为

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_4 + k_5) \quad (4-34)$$

3 阶计算公式为

$$\tilde{y}_{n+1} = y_n + \frac{h}{6}(3k_1 - 9k_3 + 12k_4) \quad (4-35)$$

式中,

$$\left. \begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{3}, y_n + \frac{h}{3}k_1\right) \\ k_3 &= f\left[t_n + \frac{h}{3}, y_n + \frac{h}{6}(k_1 + k_2)\right] \\ k_4 &= f\left[t_n + \frac{h}{2}, y_n + \frac{h}{8}(k_1 + 3k_3)\right] \\ k_5 &= f\left[t_n + h, y_n + \frac{h}{2}(k_1 - 3k_3 + 4k_4)\right] \end{aligned} \right\} \quad (4-36)$$

误差估计式为

$$E_n = \tilde{y}_{n+1} - y_{n+1} = \frac{h}{6}(2k_1 - 9k_3 + 8k_4 - k_5) \quad (4-37)$$

在变步长的过程中,通常取下面的误差计算形式:

$$e_n = \frac{E_n}{|y_n| + 1} \quad (4-38)$$

即,  $|y_n|$  较大时,  $e_n$  是相对误差,  $|y_n|$  较小时,  $e_n$  是绝对误差。

在仿真时,事先设定误差上限  $e_{\max}$ 、误差下限  $e_{\min}$ 、步长上限  $h_{\max}$ 、步长下限  $h_{\min}$ ,并按如下



原则调整步长:

(1) 若  $e_n > e_{\max}$  且  $h > h_{\min}$ , 则将步长减小(一般是减半), 重新计算本步积分值, 再进行比较。

(2) 若  $e_n < e_{\min}$  且  $h < h_{\max}$ , 则将步长增大(一般是加倍), 以新的步长计算下一步。

(3) 若  $e_n < e_{\min}$  且  $h > h_{\max}$ , 则步长不变。

(4) 若  $h_{\min} < e_n < h_{\max}$ , 则步长不变。

(5) 若  $e_n > e_{\max}$  且  $h < h_{\min}$ , 则给出警告。

若仿真步长小于步长下限  $h_{\min}$ , 则不再继续减半, 以避免增加仿真时间, 增大舍入误差。若步长大于步长上限  $h_{\max}$ , 则不再继续增大, 以避免增加截断误差, 减小数值稳定性。

该算法为 4 阶精度, 3 阶误差估计, 故亦称 RKM3-4 法, 是目前广泛应用的一种数值积分法。其在实轴上的稳定区为  $(-3.54, 0)$ 。该方法的缺点是计算量较大, 每步需要计算 5 次  $f$  函数。

## 2. 龙格-库塔-费尔别格法

Fehlberg 于 1969 年给出了龙格-库塔-费尔别格法(Runge-Kutta-Fehlberg), 其误差计算公式是由 4 阶计算公式与 5 阶计算公式之差求得的亦称 RKF4-5 法。其 5 阶计算公式为

$$y_{n+1} = y_n + h \left( \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 \right) \quad (4-39)$$

4 阶计算公式为

$$\tilde{y}_{n+1} = y_n + h \left( \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5 \right) \quad (4-40)$$

式中,

$$\left. \begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{4}, y_n + \frac{h}{4}k_1\right) \\ k_3 &= f\left[t_n + \frac{3h}{8}, y_n + \frac{h}{32}(3k_1 + 9k_2)\right] \\ k_4 &= f\left[t_n + \frac{12h}{13}, y_n + \frac{h}{2197}(1932k_1 - 7200k_2 + 7296k_3)\right] \\ k_5 &= f\left[t_n + h, y_n + h\left(\frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_2 - \frac{845}{4104}k_4\right)\right] \\ k_6 &= f\left[t_n + \frac{h}{2}, y_n + h\left(-\frac{8}{27}k_1 + 2k_2 - \frac{3544}{2566}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)\right] \end{aligned} \right\} \quad (4-41)$$

误差估计式为

$$E_n = \tilde{y}_{n+1} - y_{n+1} = h(-2.777k_1 - 29.942k_3 + 29.199k_4 - 20k_5 + 36.363k_6) \times 10^{-3} \quad (4-42)$$

在变步长的过程中, 通常取下面的误差计算形式



$$e_n = \frac{E_n}{|y_n| + 1} \quad (4-43)$$

变步长的原则同龙格-库塔-默森法。

### 3. 步长折半法

该方法通过 4 阶龙格-库塔法的计算公式求得误差估计值。具体为

$$E_n = |y_{n+1}^{h/2} - y_{n+1}^h| \quad (4-44)$$

$$e_n = \frac{E_n}{|y_n| + 1} \quad (4-45)$$

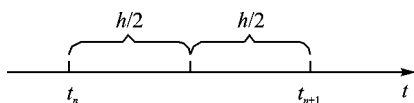


图 4-14 步长折半法示意图

其中,  $y_{n+1}^{h/2}$  为由  $t_n$  出发, 以  $h/2$  为步长, 经两步递推到达  $t_{n+1}$  的值, 如图 4-14 所示。

变步长的原则同上。

上述的变步长方法, 虽然每一步的计算量增加了, 但总体考虑往往是划算的。这种方法能较好地解决计算精度与计算量之间的矛盾, 尤其适合特征根分布很散的情况。

## 4.4 亚当姆斯法

前面讨论的欧拉法及龙格-库塔法都是单步法, 在计算  $y_{n+1}$  的值时, 只要利用前一步的  $y_n$  和  $f_n$  的值就可以自动进行计算。实际上, 在逐步递推的计算过程中, 在计算  $y_{n+1}$  之前已经求出了一系列时间点上的值  $y_1, y_2, \dots, y_n$  及  $f_1, f_2, \dots, f_n$ , 如果充分利用前面多步的信息来计算  $y_{n+1}$ , 则可构造出计算速度快、精度高的算法, 这就是线性多步法的基本思想。在线性多步法中, 应用较广的是亚当姆斯法(Adams)。

### 4.4.1 亚当姆斯显式公式

将微分方程  $\dot{y} = f(t, y)$  在  $[t_n, t_{n+1}]$  区间进行积分可得

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, y) d\tau \quad (4-46)$$

若已经求得  $t_n, t_{n-1}, \dots, t_{n-k}$  等  $k+1$  个时间点处的数据  $f_n, f_{n-1}, \dots, f_{n-k}$ , 则可利用插值原理构造区间  $[t_n, t_{n+1}]$  上的插值多项式  $P(t)$  来逼近  $f(t, y)$ 。由于构造插值多项式所用的时间点不在区间  $[t_n, t_{n+1}]$  之外, 故此时用的是外推方法。积分项  $\int_{t_n}^{t_{n+1}} f(\tau, y) d\tau$  的近似值  $\int_{t_n}^{t_{n+1}} P(\tau) d\tau$  可写为

$$\int_{t_n}^{t_{n+1}} f(t, y) dt \approx \int_{t_n}^{t_{n+1}} P(t) dt = h \sum_{i=0}^k \beta_{ki} f_{n-i} \quad (4-47)$$

根据式(4-46)和(4-47)可直接写出  $y_{n+1}$  的递推公式

$$y_{n+1} = y_n + h \sum_{i=0}^k \beta_{ki} f_{n-i} \quad (4-48)$$



式中的参数  $\beta_{ki}$  与  $k$  有关,其部分数值如表 4-3 所示。

式(4-48)就是亚当姆斯外推公式的一般形式,它是  $k+1$  步法,又称亚当姆斯显式公式或 Adams - Bashfoth 公式。

表 4-3 亚当姆斯显式公式的  $\beta_{ki}$  系数表

$k$	$\beta_{k0}$	$\beta_{k1}$	$\beta_{k2}$	$\beta_{k3}$	$\beta_{k4}$	方法的阶数
0	1	0	0	0	0	1
1	3/2	-1/2	0	0	0	2
2	23/12	-16/12	5/12	0	0	3
3	55/24	-59/24	37/24	-9/24	0	4
4	1 901/720	-2 774/720	2 616/720	-1 274/720	251/720	5

当  $k=0$  时,式(4-48)就成为欧拉公式。

当  $k=1$  时,得到二步法公式为

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1}) \quad (4-49)$$

其截断误差为

$$R_{n+1} = \frac{5}{12}h^3 y^{(3)}(\xi) \quad (4-50)$$

当  $k=3$  时,得到四步法公式为

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \quad (4-51)$$

其截断误差为

$$R_{n+1} = \frac{251}{720}h^5 y^{(5)}(\xi) \quad (4-52)$$

式(4-51)就是常用的亚当姆斯四步显式公式,是由前面 4 个点上的数值来计算下一点的值,其计算精度为 4 阶。

由上可见,亚当姆斯显式公式是  $k+1$  步外推公式,由于其局部截断误差为  $O(h^{k+2})$ ,故为  $k+1$  阶精度。

#### 4.4.2 亚当姆斯隐式公式

上述亚当姆斯显式公式是外推公式,根据插值理论可知,同样阶数的内插公式比外推公式更精确。若已经得到  $t_{n+1}, t_n, t_{n-1}, \dots, t_{n-k+1}$  等  $k+1$  个时间点处的数据  $f_{n+1}, f_n, f_{n-1}, \dots, f_{n-k+1}$ ,则可利用插值原理构造区间  $[t_n, t_{n+1}]$  上的插值多项式  $P(t)$  来逼近  $f(t, y)$ 。此时用的是内插方法。直接写出  $y_{n+1}$  的递推公式为

$$y_{n+1} = y_n + h \sum_{i=0}^k \beta_{ki}^* f_{n-i+1} \quad (4-53)$$

式中的参数  $\beta_{ki}^*$  与  $k$  有关,其部分数值如表 4-4 所示。



式(4-53)就是亚当姆斯内插公式的一般形式,它是  $k+1$  步法,又称亚当姆斯隐式公式或 Adams - Mouton 公式。

表 4-4 亚当姆斯隐式公式的  $\beta_{ki}^*$  系数表

$k$	$\beta_{ki}^*$	$\beta_{ki}^*$	$\beta_{ki}^*$	$\beta_{ki}^*$	$\beta_{ki}^*$	方法的阶数
0	1	0	0	0	0	1
1	1/2	1/2	0	0	0	2
2	5/12	8/12	-1/12	0	0	3
3	9/24	19/24	-5/24	1/24	0	4
4	251/720	646/720	-264/720	106/720	-19/720	5

当  $k=1$  时,得到二步法公式为

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n) \quad (4-54)$$

其截断误差为 
$$R_{n+1} = -\frac{1}{12}h^3 y^{(3)}(\xi) \quad (4-55)$$

当  $k=3$  时,得到四步法公式为

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \quad (4-56)$$

其截断误差为 
$$R_{n+1} = -\frac{19}{720}h^5 y^{(5)}(\xi) \quad (4-57)$$

由上可见,亚当姆斯隐式公式的局部截断误差为  $O(h^{k+2})$ ,故也为  $k+1$  阶精度。

将亚当姆斯显式公式与隐式公式比较可知:

- (1) 从计算量上看,显式公式比隐式公式计算量小。
- (2) 从计算精度上看,除一阶外,相同阶数的隐式公式的系数值比显式公式的小,因而隐式公式比显式公式精确。
- (3) 隐式公式需要计算  $f_{n+1}$ ,通常需要用亚当姆斯显式公式为它提供一个首次近似值  $y_{n+1}$ ,又称预估值。

亚当姆斯法每步只需计算一次  $f$  函数值,因而计算量较小,这是线性多步法的共同优点。但它需要知道多个出发值才能进行递推计算,而微分方程只能提供一个初值,故线性多步法不像单步法那样可以自启动。为了保证出发值的计算精度,一般采用阶数较高的单步法,以小步长进行计算。此外,亚当姆斯法改变步长不像单步法那样方便。

#### 4.4.3 预估校正公式

隐式公式的主要优点在于精度高,但需要用显式公式提供一个预估值。这种显式公式和



隐式公式结合起来使用以改进稳定性和精度的方法就是预估-校正法。合理的结合方式是显式公式和隐式公式的阶数一致。最简单的亚当姆斯预估校正法采用的是欧拉法预估,梯形法校正。在精度要求高时,则广泛采用4阶亚当姆斯预估校正法,其计算公式为

$$\text{预估} \quad y_{n+1}^p = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \quad (4-58)$$

$$\text{校正} \quad y_{n+1}^c = y_n + \frac{h}{24}(9f_{n+1}^p + 19f_n - 5f_{n-1} + f_{n-2}) \quad (4-59)$$

这是一个四步法,具有4阶精度,初值  $y_0$  由初始条件给出,  $y_1, y_2, y_3$  通常采用4阶龙格-库塔法计算。

#### 4.4.4 亚当姆斯法的稳定区

下面分析亚当姆斯法的稳定区,仍采用检验方程  $\dot{y} = \lambda y$  进行讨论,分析差分方程的特征根是否满足稳定性条件。

为简化问题,考察  $n=k$  时的亚当姆斯公式,此时其显式公式为

$$y_{k+1} = y_k + h \sum_{i=0}^k \beta_{ki} f_{k-i} \quad (4-60)$$

隐式公式为

$$y_{k+1} = y_k + h \sum_{i=0}^k \beta_{ki}^* f_{k-i+1} \quad (4-61)$$

将检验方程代入可得其特征方程分别为

$$z^{k+1} = z^k + h\lambda \sum_{i=0}^k \beta_{ki} z^{k-i} \quad (4-62)$$

$$z^{k+1} = z^k + h\lambda \sum_{i=0}^k \beta_{ki}^* z^{k-i+1} \quad (4-63)$$

求解特征方程,就可得到使其特征根位于单位圆内的  $h\lambda$  的取值范围。

表4-5给出了亚当姆斯法的稳定区域。可见,对于同样的阶次,隐式公式的稳定区域比显式公式大得多,这表明隐式公式的稳定性比显式公式好。随着阶数的增加,亚当姆斯法的稳定区域逐渐缩小。

表4-5 亚当姆斯法的稳定区域

阶数	1	2	3	4
显式公式	(-2, 0)	(-1, 0)	(-6/11, 0)	(-3/10, 0)
隐式公式	( $-\infty$ , 0)	(-6, 0)	(-3, 0)	(-90/49, 0)



## 4.5 刚性方程的数字仿真

### 4.5.1 刚性方程

在实际的工程系统中,经常会遇到这样的情况:某一个系统的特征值有的很大,有的很小,在数值上相差很远。描述这种系统的微分方程组称为刚性方程组或病态方程组。

现以二阶方程组为例,讨论刚性方程的特点。

方程组  $\dot{Y} = F(Y, t)$  为

$$\left. \begin{aligned} \dot{y}_1 &= 998y_1 + 1\,998y_2 \\ \dot{y}_2 &= -999y_1 - 1\,999y_2 \end{aligned} \right\} \quad (4-64)$$

方程组的雅各比矩阵  $\frac{\partial F}{\partial y}$  为一常数矩阵,其值为方程组的系数矩阵

$$A = \frac{\partial F}{\partial y} = \begin{bmatrix} 998 & 1\,998 \\ -999 & -1\,999 \end{bmatrix} \quad (4-65)$$

A 的特征值  $\lambda_1 = -1, \lambda_2 = -1\,000$ 。设初值  $y_1(0) = 1, y_2(0) = 0$ , 则方程组(4-64)的解为

$$\left. \begin{aligned} y_1 &= 2e^{-t} - e^{-1\,000t} \\ y_2 &= -e^{-t} + e^{-1\,000t} \end{aligned} \right\} \quad (4-66)$$

可见方程的解中既有快变分量也有慢变分量,且变化速度相差很大,所以式(4-64)是刚性方程,它所描述的系统是刚性系统。

下面给出刚性方程的一般定义。

对于常微分方程组  $\dot{Y} = F(Y, t), Y(t_0) = Y_0$ , 其中  $Y = (y_1, y_2, \dots, y_n)^T, F = (f_1, f_2, \dots, f_n)^T$ , 一般地,  $F$  是  $t, Y$  的非线性函数,设  $F$  的雅各比矩阵  $\frac{\partial F}{\partial y}$  的特征值  $\lambda_i$  全部具有负实部,则当比值  $\rho = \max |\operatorname{Re} \lambda| / \min |\operatorname{Re} \lambda| \gg 1$  时,称方程组  $\dot{Y} = F(Y, t)$  为刚性方程,或称 stiff 方程。比值  $\rho$  称为刚性比。有些系统中,刚性比可达  $10^6$ 。

刚性系统中各环节的时间常数差异巨大。为保证仿真计算的稳定性,仿真步长必须限制在最小时间常数的数量级而选得很小,然而仿真结束的时间则决定于系统中的最大时间常数,若按满足稳定性要求所选择的步长进行仿真,则不仅整个仿真所花费的时间非常长,甚至由于计算的舍入误差而导致整个仿真的失败。这就是所谓“刚性系统仿真”问题。如果一个系统  $\max |\operatorname{Re} \lambda_i| = 10^6, \min |\operatorname{Re} \lambda_i| = 1$ , 用 4 阶龙格-库塔法求解时,步长应满足  $h \leq 2.78/10^6$ , 即积分一秒需要的步数  $M > 10^6/2.78$ , 结果将造成机时的浪费和舍入误差的积累,使数值解失真。

基于上述原因,对刚性方程的数值解研究得到了广泛的重视,到目前为止已提出了不少解刚性方程的方法,如吉尔法、特雷尔法、半隐式龙格-库塔法等。本节介绍的吉尔法被公认是解



决刚性系统仿真十分有效的方法,也是最常用的方法。吉尔法处理刚性系统效果较好,但在处理非刚性系统时,计算量往往比其他方法大,因此在仿真前,应判别一下系统是否为刚性系统。

#### 4.5.2 吉尔法

刚性方程的数值解要满足稳定性和计算次数的矛盾,应对步长  $h$  不加限制;数值积分法必须具有特殊的稳定域。达奎斯特(Dahlquist)引进一种 A 稳定概念来解决这一矛盾,吉尔将 A 稳定性减弱的概念定义成 Stiff 稳定性,并首先推出了满足 Stiff 稳定域的隐式多步法。但是多步法不能自启动,而且变步长很困难,因此吉尔进一步将多步法变换成单步多值法。

下面首先引入 Stiff 稳定的概念,然后从吉尔多步法开始讨论,并推出单步多值法。

定义:一个方法是 Stiff 稳定的,如果在区域  $R_1 (\text{Re}(h\lambda) \leq D)$  中,它是绝对稳定的,而在区域  $R_2 (D < \text{Re}(h\lambda) < \alpha, |\text{Im}(h\lambda)| < \theta)$  中,它是精确的。

##### 1. 吉尔多步法的公式

在讨论单步多值法之前,先讨论一下吉尔多步法。首先给出其一般公式

$$y_{n+k} = \sum_{j=1}^k a_j y_{n+k-j} - h\beta_0 f_{n+k} \quad (4-67)$$

公式中的系数与  $K$  有关,如表 4-6 所列。

表 4-6 吉尔多步法的系数

$K$	1	2	3	4	5	6
$\beta_0$	1	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{12}{25}$	$\frac{60}{137}$	$\frac{60}{147}$
$\alpha_1$	1	$\frac{4}{3}$	$\frac{18}{11}$	$\frac{48}{25}$	$\frac{300}{137}$	$\frac{360}{147}$
$\alpha_2$		$-\frac{1}{3}$	$-\frac{9}{11}$	$-\frac{36}{25}$	$-\frac{300}{137}$	$-\frac{450}{147}$
$\alpha_3$			$\frac{2}{11}$	$\frac{16}{25}$	$\frac{200}{137}$	$\frac{400}{147}$
$\alpha_4$				$-\frac{3}{25}$	$-\frac{75}{137}$	$-\frac{225}{147}$
$\alpha_5$					$\frac{12}{137}$	$\frac{72}{147}$
$\alpha_6$						$-\frac{10}{147}$

以 3 阶为例,吉尔多步法的隐式公式为

$$y_{n+1} = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6}{11}hf_{n+1} \quad (4-68)$$

但是隐式方法不能自启动,需要依靠显式公式提供预估值,这里使用 3 阶显式吉尔多步法



$$y_{n+1} = -\frac{3}{2}y_n + 3y_{n-1} - \frac{1}{2}y_{n-2} + 3hf_n \quad (4-69)$$

用隐式方法解刚性方程同样需要经过预估、校正两个过程,即先用显式法做预报,再用隐式法反复迭代求校正值,其计算过程如下:

$$\text{预估} \quad y_{n+1}^{(0)} = -\frac{3}{2}y_n + 3y_{n-1} - \frac{1}{2}y_{n-2} + 3hf_n \quad (4-70)$$

$$\text{校正} \quad y_{n+1}^{(1)} = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6}{11}hf_{n+1}^{(0)} \quad (4-71)$$

$$y_{n+1}^{(2)} = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6}{11}hf_{n+1}^{(1)} \quad (4-72)$$

⋮

式中,  $y_{n+1}^{(0)}$  表示预估值,  $y_{n+1}^{(i)}$  表示第  $i$  次校正值,  $f_{n+1}^{(i)} = f(y_{n+1}^{(i)}, t_{n+1})$ 。

式(4-70)~式(4-72)为3阶吉尔多步法的递推公式,它不能自启动。

## 2. 单步多值法的递推公式

在吉尔多步法的基础上,进一步推导单步多值法。

定义

$$hf_{n+1}^{(-1)} = \frac{11}{2}hf_n - \frac{23}{4}y_n + 7y_{n-1} - \frac{5}{4}y_{n-2} \quad (4-73)$$

则可把所有校正公式写成同一个形式

$$y_{n+1}^{(1)} = y_{n+1}^{(0)} + \frac{11}{6}[hf_{n+1}^{(0)} - hf_{n+1}^{(-1)}] \quad (4-74)$$

$$y_{n+1}^{(2)} = y_{n+1}^{(1)} + \frac{11}{6}[hf_{n+1}^{(1)} - hf_{n+1}^{(0)}] \quad (4-75)$$

$$\text{令} \quad \mathbf{Y}_n = [y_n, hf_n, y_{n-1}, y_{n-2}]^T \quad (4-76)$$

$$\mathbf{Y}_{n+1}^{(i)} = [y_{n+1}^{(i)}, hf_{n+1}^{(i-1)}, y_n, y_{n-1}] \quad (4-77)$$

分别为原始数据向量和结果数据向量,  $i=0, 1, \dots, M$ ,  $M$  是每步迭代次数,则每一步的计算任务就是从  $\mathbf{Y}_n$  计算出  $\mathbf{Y}_{n+1}^{(M)}$ , 这样可得两个矩阵公式

$$\text{预估} \quad \mathbf{Y}_{n+1}^{(0)} = \mathbf{B}\mathbf{Y}_n \quad (4-78)$$

$$\text{校正} \quad \mathbf{Y}_{n+1}^{(i+1)} = \mathbf{Y}_{n+1}^{(i)} + \mathbf{C}\mathbf{G}_{n+1}^{(i)} \quad (4-79)$$

$$\text{式中} \quad \mathbf{G}_{n+1}^{(i)} = hf_{n+1}^{(i)} - hf_{n+1}^{(i-1)}, i=0, 1, \dots, M \quad (4-80)$$

$$\mathbf{B} = \begin{bmatrix} -\frac{3}{2} & 3 & 3 & -\frac{1}{2} \\ -\frac{23}{4} & \frac{11}{2} & 7 & -\frac{5}{4} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \frac{6}{11} \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (4-81)$$



式(4-78)和(4-79)是从3阶方法推导出的,它可以直接推广到 $K$ 阶方法,并且形式不变。但对于 $K$ 阶方法,式(4-76)和(4-77)需变为

$$\mathbf{Y}_n = [y_n, hf_n, y_{n-1}, y_{n-2}, \dots, y_{n-K+1}]^T \quad (4-82)$$

$$\mathbf{Y}_{n+1}^{(i)} = [y_{n+1}^{(i)}, hf_{n+1}^{(i-1)}, y_n, y_{n-1}, \dots, y_{n-K+2}] \quad (4-83)$$

对于不同的阶次,式(4-78)和(4-79)中的 $\mathbf{B}$ 和 $\mathbf{C}$ 是不同的。

将多步吉尔法写成矩阵形式后,为了解决自启动取变步长的困难,还要把多步法中需要用的前几步的数据,如 $f_{n-1}, f_{n-2}, y_{n-1}, y_{n-2}, \dots, y_{n-K+1}$ 等换算成前一步的数据。同时,换算要保证所得算法的截断误差具有相同的阶数,对于3阶算法,要求误差仍在 $O(h^4)$ 之内。

定义两个 $\mathbf{Z}$ 向量,用前一步的高阶导数代替前几步的 $y$ 数据或 $f$ 数据

$$\mathbf{Z}_n = \left[ y_n, h\dot{y}_n, \frac{h^2}{2!}\ddot{y}_n, \frac{h^3}{3!}\overset{\cdot\cdot\cdot}{y}_n, \dots, \frac{h^K}{K!}y_n^{(K)} \right]^T \quad (4-84)$$

$$\mathbf{Z}_{n+1} = \left[ y_{n+1}, h\dot{y}_{n+1}, \frac{h^2}{2!}\ddot{y}_{n+1}, \frac{h^3}{3!}\overset{\cdot\cdot\cdot}{y}_{n+1}, \dots, \frac{h^K}{K!}y_{n+1}^{(K)} \right]^T \quad (4-85)$$

显然,式(4-84)中 $h\dot{y}_n$ 的和式(4-85)中的 $h\dot{y}_{n+1}$ 就是式(4-82)中的 $hf_n$ 和式(4-83)中的 $hf_{n+1}^{(i-1)}$ 。所以在1阶情况下, $\mathbf{Z}$ 向量就是 $\mathbf{Y}$ 向量,而在高阶情况下, $\mathbf{Z}$ 向量用 $y$ 的高阶导数代替了 $y$ 在前几步的值。

下面仍以3阶为例,推导 $\mathbf{Z}$ 向量和 $\mathbf{Y}$ 向量之间的转换关系。为了保证截断误差不变,用一个3次多项式拟合 $y(t)$

$$y(t) = a_0 + a_1(t-t_n) + a_2(t-t_n)^2 + a_3(t-t_n)^3 \quad (4-86)$$

$$t = t_n \text{ 时, 有 } \left. \begin{aligned} y_n &= a_0 \\ h\dot{y}_n &= a_1 h \\ \frac{h^2}{2!}\ddot{y}_n &= a_2 h^2 \\ \frac{h^3}{3!}\overset{\cdot\cdot\cdot}{y}_n &= a_3 h^3 \end{aligned} \right\} \quad (4-87)$$

$$\text{根据式(4-85) 还 可 得 } \left. \begin{aligned} y_n &= a_0 \\ hf_n &= a_1 h \\ y_{n-1} &= a_0 - a_1 h + a_2 h^2 - a_3 h^3 \\ y_{n-2} &= a_0 - 2a_1 h + 4a_2 h^2 - 8a_3 h^3 \end{aligned} \right\} \quad (4-88)$$

整理式(4-87)和式(4-88),消去 $a_0, a_1, a_2, a_3$ ,可得

$$\begin{bmatrix} y_n \\ h\dot{y}_n \\ \frac{h^2}{2!}\ddot{y}_n \\ \frac{h^3}{3!}\overset{\cdot\cdot\cdot}{y}_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{7}{4} & \frac{3}{2} & 2 & -\frac{1}{4} \\ -\frac{3}{4} & \frac{1}{2} & 1 & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} y_n \\ hf_n \\ y_{n-1} \\ y_{n-2} \end{bmatrix} \quad (4-89)$$



写成向量形式为

$$\mathbf{Z}_n = \mathbf{Q}\mathbf{Y}_n \quad (4-90)$$

式中

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{7}{4} & \frac{3}{2} & 2 & -\frac{1}{4} \\ -\frac{3}{4} & \frac{1}{2} & 1 & -\frac{1}{4} \end{bmatrix} \quad (4-91)$$

为 3 阶时  $\mathbf{Y}$  向量与  $\mathbf{Z}$  向量之间的变换阵。

由式(4-91)可知,对于  $K$  阶方法, $\mathbf{Q}$  阵为  $(K+1)$  阶非奇异方阵。用  $\mathbf{Q}$  阵对吉尔算法公式(4-78)和(4-79)进行变换,可得如下预估及校正公式

$$\text{预估} \quad \mathbf{Z}_{n+1}^{(0)} = \mathbf{Q}\mathbf{B}\mathbf{Q}^{-1}\mathbf{Z}_n = \mathbf{P}\mathbf{Z}_n \quad (4-92)$$

$$\text{校正} \quad \mathbf{Z}_{n+1}^{(i+1)} = \mathbf{Z}_{n+1}^{(i)} + \mathbf{Q}\mathbf{C}\mathbf{G}_{n+1}^{(i)} = \mathbf{Z}_{n+1}^{(i)} + \mathbf{L}\mathbf{G}_{n+1}^{(i)} \quad (4-93)$$

$$\text{式中} \quad \mathbf{G}_{n+1}^{(i)} = hf(y_{n+1}^{(i)}) - hf(y_{n+1}^{(i-1)}), i = 0, 1, 2, \dots, M \quad (4-94)$$

是取  $\mathbf{Z}_{n+1}^{(i)}, \mathbf{Z}_{n+1}^{(i)}$  的头一个分量。

$$\mathbf{P} = \mathbf{Q}\mathbf{B}\mathbf{B}^{-1}, \quad \mathbf{L} = \mathbf{Q}\mathbf{C} \quad (4-95)$$

在 3 阶的情况下,可计算出  $\mathbf{P}$  和  $\mathbf{L}$  为

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \frac{6}{11} \\ \frac{6}{11} \\ \frac{1}{11} \end{bmatrix} \quad (4-96)$$

从式(4-96)可知, $\mathbf{P}$  为上三角矩阵,而且其各列系数是二项公式的各个系数,这种矩阵在数学上被称为 Pascal 三角阵,可以证明对于任意阶  $K$ ,式(4-95)的  $\mathbf{P}$  都具有上述性质。 $\mathbf{L}$  向量随阶数的变化的情况见表 4-7。

表 4-7 吉尔单步多值法的系数

$K$	1	2	3	4	5	6
$l_0$	1	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{24}{50}$	$\frac{120}{274}$	$\frac{720}{1764}$
$l_1$	1	$\frac{3}{3}$	$\frac{11}{11}$	$\frac{50}{50}$	$\frac{274}{274}$	$\frac{1764}{1764}$
$l_2$		$\frac{1}{3}$	$\frac{6}{11}$	$\frac{35}{50}$	$\frac{225}{274}$	$\frac{1624}{1764}$



续表 4-7

K	1	2	3	4	5	6
$l_3$			$\frac{1}{11}$	$\frac{10}{50}$	$\frac{85}{274}$	$\frac{735}{1764}$
$l_4$				$\frac{1}{50}$	$\frac{15}{274}$	$\frac{175}{1764}$
$l_5$					$\frac{1}{274}$	$\frac{21}{1764}$
$l_6$						$\frac{1}{1764}$

由于  $Z_n$  为  $y_n, \dot{y}_n, \ddot{y}_n, \dots$  所构成的向量,  $Z_{n+1}$  为  $y_{n+1}, \dot{y}_{n+1}, \ddot{y}_{n+1}, \dots$  所构成的向量, 因此式(4-92)和(4-93)是一种单步多值法, 利用这种算法可以自启动, 而且克服了多步法在变步长方面的困难。该方法与一般单步法不同的是, 每步计算, 不仅要保留  $y_n$ , 而且要保留  $y_n$  的各阶导数  $\dot{y}_n, \ddot{y}_n, \dots, y_n^{(K)}$ 。

#### 4.5.3 单步多值法的误差估计

式(4-92)和(4-93)给出了吉尔单步多值法的递推公式, 它要求每次计算时已知  $y_n, \dot{y}_n, \ddot{y}_n, \dots$ , 但是在计算一开始, 只能获得  $y_0$  和  $\dot{y}_0$ , 因此只能从一阶启动, 以后再自动升阶。在计算过程中, 为了使计算控制在规定的精度要求内, 同时又尽量减少计算量, 还要不断自动改变阶次及步长。为了实现自动变阶、变步长, 首先要做误差估计, 然后以一定的原则来改变阶数和步长。

已知  $K$  阶方法的截断误差为

$$E = \frac{h^{K+1}}{(K+1)!} y^{(K+1)}(\xi) = C_{K+1} h^{(K+1)} y^{(K+1)}(\xi) \quad (4-97)$$

$\xi$  是所讨论区间中的某一个点。对于 3 阶吉尔法,  $\xi$  是  $[t_n - 2h, t_n + h]$  中的某个点。为简单起见, 假设第  $n$  步的  $y^{(K+1)}(\xi) = y_{n(K+1)}$ , 则第  $n$  步的截断误差为

$$E_m = C_{K+1} h^{K+1} y_n^{(K+1)} \quad (4-98)$$

已知  $Z_n = \left[ y_n, h\dot{y}_n, \frac{h^2}{2!}\ddot{y}_n, \frac{h^3}{3!}y_n^{(3)}, \dots, \frac{h^K}{K!}y_n^{(K)} \right]^T = [z_n, \dot{z}_n, \ddot{z}_n, \dots, z_n^{(K)}]^T \quad (4-99)$

现在用  $y_n^{(K)}$  的差分来近似估计  $y_n^{(K+1)}$ , 即

$$\begin{aligned} E_m &= C_{K+1} h^{K+1} y_n^{(K+1)} \approx C_{K+1} h^{K+1} \frac{y_n^{(K)} - y_{n-1}^{(K)}}{h} = C_{K+1} K h^K \frac{y_n^{(K)} - y_{n-1}^{(K)}}{K!} = \\ &C_{K+1} K! \left( \frac{h^K}{K!} y_n^{(K)} - \frac{h^K}{K!} y_{n-1}^{(K)} \right) = C_{K+1} K! (z_n^K - z_{n-1}^K) = \\ &C_{K+1} K! \nabla z_n^K \end{aligned} \quad (4-100)$$



由式(4-100)可知,只要将第  $n$  步计算所得的  $Z_m$  与前一步的  $Z_{m-1}$  的最后一个分量乡间即可求出截断误差  $E_m$ 。

在仿真中通常使用相对误差,即

$$\epsilon = \frac{|E|}{|y|_{\max}} \quad (4-101)$$

式中,  $|y|_{\max}$  为积分到第  $n$  步为止已出现过的  $|y|$  的最大值,但在开始积分时,若  $y=0$ ,则应取  $|y|_{\max}=1$ 。

在微分方程组的情况下,  $y$  共有  $N$  个变量  $y(1), y(2), \dots, y(N)$ ,则应将相对误差定义为

$$\epsilon = \sqrt{\sum_{j=1}^N \left[ \frac{E(j)_m}{|y(j)|_{\max}} \right]^2} \quad (4-102)$$

对于事先规定的相对误差限  $\epsilon_0$ ,若  $\epsilon \leq \epsilon_0$  则认为这一步计算结果有效,转入下一步计算,并考虑改变阶数  $K$  及步长  $h$ ;若  $\epsilon > \epsilon_0$  则认为这一步计算结果无效,应考虑改变阶数  $K$  及步长  $h$  重新进行本步计算;若  $\epsilon \ll \epsilon_0$  则认为积分过程还可以加快,也应考虑改变  $K$  及  $h$  进行下一步计算。下面讨论如何改变阶次及步长。

#### 4.5.4 单步多值法阶与步长的控制

当一步积分得出结果以后,要检验误差,无论是  $\epsilon \leq \epsilon_0$  还是  $\epsilon > \epsilon_0$ ,或是  $\epsilon \ll \epsilon_0$ ,都要考虑变阶、变步长的问题,以使  $\epsilon \approx \epsilon_0$ 。这里存在一个改变阶次和步长的原则问题,主要包括:

- (1) 如果不变阶只变步长,设新的步长为  $h_s$ ,令  $h_s = R_K h$ ,那么  $R_K$  应取多大,才能使  $\epsilon \approx \epsilon_0$ ;
- (2) 如果降一阶,同时变步长,令  $h_s = R_{K-1} h$ ,那么  $R_{K-1}$  应取多大,才能使  $\epsilon \approx \epsilon_0$ ;
- (3) 如果升一阶,同时变步长,令  $h_s = R_{K+1} h$ ,那么  $R_{K+1}$  应取多大,才能使  $\epsilon \approx \epsilon_0$ ;
- (4) 比较  $R_K, R_{K-1}, R_{K+1}$ ,哪种方法能得到最大的  $R$  值就用哪种方法。

由于误差是估计值,因此以上求出的  $R$  值应乘以一个安全系数。在不变阶只变步长的情况下可取次系数为  $1/1.2$ ;在变阶、变步长的情况下,考虑到误差的估计更加间接,而且变阶还要增加额外的计算工作,所以系数取得更加严格。降阶情况下取为  $1/1.3$ ;升阶情况下取为  $1/1.4$ 。

在阶次不变时,为保证用  $h_s = R_K h$  进行计算能使相对误差接近  $\epsilon_0$ ,根据式(4-99)和(4-100),令

$$\begin{aligned} \epsilon &= \left| \frac{C_{K+1} y_{n+1}^{(K+1)} (R_K h)^{K+1}}{y_{\max}} \right| = R_K^{K+1} \left| \frac{C_{K+1} y_{n+1}^{(K+1)} h^{K+1}}{y_{\max}} \right| \approx \\ &R_K^{K+1} \left| \frac{C_{K+1} K! \nabla z_m^K}{y_{\max}} \right| \approx \epsilon_0 \end{aligned} \quad (4-103)$$

整理并考虑上面的安全系数,可得

$$R_K = \frac{1}{1.2} \left( \frac{\epsilon_0 |y_{\max}|}{C_{K+1} K! |\nabla z_m^K|} \right)^{\frac{1}{K+1}} \quad (4-104)$$



在降 1 阶时,为保证用  $K-1$  阶及  $h_s = R_{K-1}h$  进行计算能使相对误差接近  $\varepsilon_0$ ,则要求

$$\left| \frac{C_K y_n^{(K)} (R_{K-1}h)^K}{y_{\max}} \right| \approx \varepsilon_0 \quad (4-105)$$

整理并考虑上面的安全系数,可得

$$R_{K-1} = \frac{1}{1.3} \left( \frac{\varepsilon_0 |y_{\max}|}{C_K K! |z_m^K|} \right)^{\frac{1}{K}} \quad (4-106)$$

在升 1 阶时,为保证用  $K+1$  阶及  $h_s = R_{K+1}h$  进行计算能使相对误差接近  $\varepsilon_0$ ,则要求

$$\left| \frac{C_{K+2} y_n^{(K+2)} (R_{K+1}h)^{K+2}}{y_{\max}} \right| \approx \varepsilon_0 \quad (4-107)$$

整理并考虑上面的安全系数,可得

$$R_{K+1} = \frac{1}{1.4} \left( \frac{\varepsilon_0 |y_{\max}|}{C_{K+2} K! |\nabla^2 z_m^K|} \right)^{\frac{1}{K+2}} \quad (4-108)$$

在积分过程中,阶与步长均要改变,因此需要由原来的  $Z_n$  计算新的  $Z_n$ 。

在不变阶变步长时,根据式(4-99) $Z_n$ 按下式计算:

$$Z_n = \left[ y_n, R_K, h \dot{y}_n, \dots, R_K^K \frac{h^K}{K!} y_n^{(K)} \right]^T \quad (4-109)$$

在阶次由  $K$  变为  $K-1$  时,步长由  $h$  变为  $R_{K-1}h$ ,根据式(4-99)有

$$Z_n = \left[ y_n, R_{K-1}, h \dot{y}_n, \dots, R_{K-1}^{K-1} \frac{h^{K-1}}{(K-1)!} y_n^{(K-1)} \right]^T \quad (4-110)$$

在阶次由  $K$  变为  $K+1$  时,步长由  $h$  变为  $R_{K+1}h$ ,根据式(4-99)有

$$Z_n = \left[ y_n, R_{K+1}, h \dot{y}_n, \dots, R_{K+1}^K \frac{h^K}{K!} y_n^{(K)}, R_{K+1}^{K+1} \frac{h^{K+1}}{(K+1)!} \frac{y_n^{(K)} - y_{n-1}^{(K)}}{h} \right]^T \quad (4-111)$$

式(4-111)中,因为  $y_n^{(K+1)}$  是未知的,所以最后一项用差分近似。

## 4.6 计算方法和计算步长的选择

以上介绍了几种数值积分方法,在仿真时需要根据被仿真系统的具体情况进行具体分析、合理选择算法和计算步长。算法和步长的选择涉及因素较多,而且直接影响到数值解的精度、速度和稳定性,因而比较复杂,至今尚无一种确定的办法。一般来说,应考虑以下因素:方法的难易、稳定性要求、计算量的大小、数值解的精度、步长的调整及方程的刚性程度等。下面分别加以讨论,以供选择时参考。

### 4.6.1 积分方法的选择

积分方法的选择主要应考虑如下因素:



### 1. 精度要求

应用数值积分法对连续系统进行数字仿真,其精度主要受截断误差、舍入误差和积累误差的影响,它们与数值积分方法、仿真步长、计算时间及所用的计算机精度等有关。在步长相同的情况下,积分方法的阶次越高,截断误差越小。在同阶方法中,多步法比单步法精度高,而其中隐式算法的精度又高于显式算法。因此,当需要较高的仿真精度时,可采用高阶的多步隐式算法和较小的步长,但是步长的减小必然会增加迭代次数,增大计算量,并增大舍入误差和积累误差。

总之,应当根据所要求的精度,合理地选择积分方法和阶数。在进行具体计算时,并不是说方法的阶数越高、步长越小,效果就越好。经验表明,精度要求不高的问题最好用低阶方法来处理。

### 2. 计算速度

计算速度取决于计算步数及每步积分所需时间,而每步的计算时间又与积分方法有关,它主要取决于计算导数的次数。在数值解中,最费时的部分是积分变量导数的计算,4阶龙格-库塔法每步需计算4次导数,费时较多;在典型的亚当姆斯预估校正法中,每步只要计算两次导数(预估一次、校正一次);而在显式亚当姆斯法中,每步只要计算一次导数就行了,计算速度明显加快。因此,在导数方程很复杂、计算量较大且精度要求高的问题中,可采用亚当姆斯预估校正法。为加快计算速度,在积分方法已定的条件下,应在保证精度的前提下,尽量选用较大的步长,以缩短计算时间。

### 3. 数值稳定性

保证数值解的稳定性是进行数字仿真的先决条件,否则,计算结果将失去真实意义。

表4-8列出了本章介绍的数值积分法的稳定区域,其中,改进欧拉法和隐式亚当姆斯法的稳定区指的是其校正公式的稳定区。

表 4-8 数值积分法的稳定区域

方 法	1 阶	2 阶	3 阶	4 阶
欧拉法	$(-2, 0)$			
改进欧拉法	$(-\infty, 0)$			
龙格-库塔法	$(-2, 0)$	$(-2, 0)$	$(-2.51, 0)$	$(-2.78, 0)$
显式亚当姆斯法	$(-2, 0)$	$(-1, 0)$	$(-6/11, 0)$	$(-0.3, 0)$
隐式亚当姆斯法	$(-\infty, 0)$	$(-6, 0)$	$(-3, 0)$	$(-90/49, 0)$

从数值解的稳定性来看,3阶以下的隐式亚当姆斯法具有较好的稳定性,同阶的龙格-库塔法比显式的亚当姆斯法要好,例如4阶龙格-库塔法的稳定区为 $(-2.78, 0)$ ,而4阶显式亚当姆斯法的稳定区仅为 $(-0.3, 0)$ 。所以,从数值稳定性方面考虑最好避免使用显式亚当姆



斯法。

对于刚性程度很大的系统应选择吉尔法。

综上所述,数值方法的选择有很大的灵活性,要根据具体情况而定。在导数计算量不大而精度要求较高时,4阶龙格-库塔法是很好的方法,导数计算量较大可采用亚当姆斯预估校正法,刚性程度大的系统应选择吉尔法。

#### 4.6.2 计算步长的选择

从前面各节的分析可知,步长的选择很重要,步长过大会带来较大的截断误差,甚至出现数值不稳定现象。步长过小又会增加步数,使舍入误差累积,总的误差反而变大。所以仿真的总误差与步长的关系不是单调函数,而是一个具有极值的函数。如图4-15所示,从图示可知,并不是步长越小精度越高,而是存在一个最佳步长。

因此,积分方法确定后,在选取积分步长时,需要考虑的一个重要因素就是仿真系统的动态响应。如果系统的动态响应快,导数变化剧烈,则应选取高阶的数值计算方法,而且步长也应取的较小。为了保证计算的稳定性,步长应限制在最小时间常数的数量级上,相当于最大特征值的倒数。例如对于4阶龙格-库塔法,步长的选择应满足  $h < 2.78/|\lambda_{\min}|$ ,但为了保证足够的计算精度,实际所选步长还应当小一些。通常在用经验方法选取步长时,一种方法是根据系统方程中最小时间常数来确定步长,一般取

$$h = (0.2 \sim 0.05) T_{\min} \quad (4-112)$$

另一种方法是根据系统开环频率特性的剪切频率  $\omega_c$  来确定步长,一般取

$$h = \frac{1}{(5 \sim 20)\omega_c} \quad (4-113)$$

系统中与最小时间常数对应的极点只影响过渡过程起始段的形状,而过渡过程则主要由那些靠近虚轴的主导极点决定。由于固定步长是按照起始段来选取的,这就会造成后面阶段计算量的浪费。为解决这一问题,可采用前面介绍的变步长方法。

对大多数工程应用来讲,一般情况下误差不超过 0.5% 即可满足精度要求,故通常采用固定步长的方法。即根据经验初步选择一个步长,然后通过仿真试探确定下来。

本章首先利用欧拉法介绍了数值积分法的基本思想和基本概念,同时从应用角度出发,介绍了几种常用的数值积分方法,最后讨论了积分方法和计算步长的选择问题。

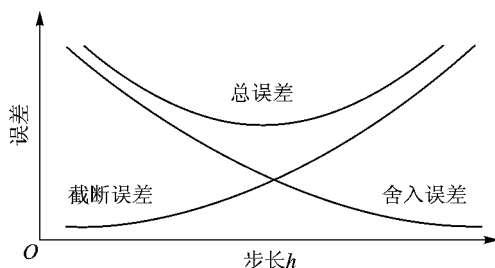


图 4-15 步长和误差的关系曲线



## 习 题

4.1 设微分方程为  $T\dot{y}+y=K$ , 试用欧拉法和改进欧拉法写出解的差分方程, 并讨论: 步长应选择在什么范围? 若  $h>2T$ , 将会产生什么结果? 并说明其原因。

4.2 已知微分方程  $\ddot{y}+0.5\dot{y}-2y=0$ ,  $y(0)=1$ ,  $\dot{y}(0)=0$ , 取  $h=0.1$ , 分别用欧拉法和 4 阶龙格-库塔法写出第一步的计算公式和计算结果。

4.3 已知系统传递函数  $G(s)=\frac{40.6}{s^3+10s+27s+22.06}$ , 其单位阶跃响应的解析解为  $y(t)=1.84-4.95te^{-1.88t}-1.5e^{-1.88t}-0.34e^{-6.24t}$ 。试用欧拉法、改进欧拉法、4 阶龙格-库塔法、4 阶亚当姆斯预估校正法对该系统进行仿真。(1) 比较这 4 种数值方法与解析解的逼近程度;(2) 改变计算步长, 分析步长对数值解精度的影响;(3) 根据 4 种方法的稳定区计算步长的选取范围, 并与实际仿真结果相比较。

4.4 已知某系统动力学模型如下: 初值为  $x_1(0)=620$ ,  $x_2(0)=10$ ,  $x_3(0)=70$ , 用 4 阶龙格-库塔法对该系统进行仿真, 自选步长, 仿真时间为 10 s。

$$\left. \begin{aligned} \dot{x}_1 &= -0.001x_1x_2 \\ \dot{x}_2 &= 0.001x_1x_2 - 0.072x_2 \\ \dot{x}_3 &= 0.072x_2 \end{aligned} \right\}$$

## 参考文献

- [1] 郭云芳, 刘继美, 欧阳玲编著. 计算机仿真技术. 北京: 北京航空航天大学出版社, 1991.
- [2] 熊光楞编著. 控制系统数字仿真. 北京: 清华大学出版社, 1982.
- [3] 吉尔 C W 编著. 常微分方程初值问题的数值解法. 北京: 科学出版社, 1978.

## 第 5 章 连续系统的离散化仿真

第 4 章介绍了连续系统数值积分法仿真的原理和几种方法,通过对这些方法的讨论可以看出,用数值积分法仿真,对系统输出只计算了其在一系列时间点上的值,这相当于对系统模型进行了离散化处理。所以从本质上说,数值积分法也是离散化方法,只不过它是从数值积分的角度出发讨论问题,没有明确提出“离散”这个概念;而本章则是从连续系统离散化的角度来探讨数字仿真的另一类方法。

同离散化仿真相比,数值积分法比较成熟,精度也比较高;但是它的计算公式比较复杂。虽然已经提出了一些针对实时仿真的计算公式,但其计算量仍比较大,所以这种方法通常是在纯数字仿真时应用。比如,在离线情况下,即非实时要求下估计系统的性能,可以获得较高的精度。而在进行实时仿真时,或者在计算机控制系统中要将控制器以数字算法的形式实现的时候,要求算法的计算速度很快,以保证在一个采样周期的时间内完成计算任务,在这种场合,原来的数值积分法往往不能满足实时性要求,这就需要寻求新的快速的数字仿真方法。

本章所介绍的离散化仿真方法的基本思想是:先将连续系统的数学模型进行离散化处理,再由离散化模型得到比较简单的近似计算公式,以便于在计算机上快速运算。该方法有两点基本要求:一是每一步的计算量要小;二是算法的稳定性好,允许采用较大的计算步距,也能够保证必要的精度。可见,离散化仿真方法的首要要求是快速性,对精度的要求一般则较低,所以很多资料中也把离散化仿真方法称为快速数字仿真法。

本章的内容包括连续系统离散化仿真的替换法、根匹配法、离散相似法、状态方程离散法、增广矩阵法和结构图模型离散法。这些离散化方法既可用在连续系统仿真中,也可用在计算机控制系统中数字控制器的实现上。

### 5.1 替换法

连续系统最常见的模型之一是传递函数  $G(s)$ 。一般情况下, $G(s)$ 转换成与之对应的脉冲传递函数  $G(z)$ 的方法有两种:一种是由  $G(s)$ 求出  $g(t)$ ,然后按照公式

$$G(z) = \sum_{n=0}^{\infty} g(nT)z^{-n}$$

求出  $G(z)$ ,式中的  $T$ 为采样周期;另一种方法是將  $G(s)$ 展成部分分式形式,再通过查  $z$ 变换表得到  $G(z)$ 。对于高阶系统,采用这两种转换方法都比较困难,而且麻烦。若能很方便地由  $G(s)$ 推导出  $G(z)$ ,可以很快获得数字仿真模型,这对于数字仿真是很有利的。



从  $G(s)$  直接推导出  $G(z)$  的方法一般有两种：一种叫作替换法，这种方法是找到  $s$  与  $z$  的对应公式，将  $G(s)$  中的  $s$  对换为  $z$ ，进而求得  $G(z)$ ；另一种称为根匹配法，即要找到一个与  $G(s)$  的零极点相匹配的  $G(z)$ 。本节介绍替换法，根匹配法在 5.2 节介绍。

通过先行课的学习，可知  $z$  与  $s$  之间的基本关系为

$$z = e^{sT} \quad (5-1)$$

或 
$$s = \frac{1}{T} \ln z \quad (5-2)$$

式中， $T$  为采样周期。这是一个超越函数，如果直接替换，计算会很麻烦，所以必须寻找一种与之近似的比较简单的替换公式。不同的替换公式形成了不同的替换方法。本节先从最简单的一种方法入手，对替换法的基本特点、稳定性问题进行讨论；然后介绍常用的双线性变换法。

### 5.1.1 简单替换法

简单替换法的公式可由 Euler 公式推导出来。

对微分方程

$$dx/dt = u(t) \quad (5-3)$$

运用 Euler 公式可得

$$x_{n+1} = x_n + T \cdot u_n \quad (5-4)$$

对式(5-3)进行  $z$  变换得到

$$zX(z) = X(z) + TU(z)$$

即 
$$\frac{X(z)}{U(z)} = \frac{T}{z-1} \quad (5-5)$$

由微分方程(5-3)又可得

$$\frac{X(s)}{U(s)} = \frac{1}{s} \quad (5-6)$$

故比较式(5-5)和式(5-6)可得

$$s = \frac{z-1}{T} \quad (5-7)$$

或 
$$z = Ts + 1 \quad (5-8)$$

这就是简单替换法  $s$  与  $z$  的替换关系。

下面来讨论简单替换法的稳定性。

设

$$s = \sigma + j\omega \quad (5-9)$$

将式(5-9)代入式(5-8)，有

$$|z|^2 = (\sigma T + 1)^2 + (\omega T)^2 \quad (5-10)$$

由线性离散系统稳定性的充要条件知，若  $G(z)$  的所有极点均在单位圆内，则它所表示的系统



是稳定的,即要求式(5-10)满足

$$|z|^2 < 1$$

$$\left(\sigma + \frac{1}{T}\right)^2 + \omega^2 < \left(\frac{1}{T}\right)^2 \quad (5-11)$$

式(5-11)在  $s$  平面上表示的区域是以  $(-1/T, j \cdot 0)$  为圆心,以  $1/T$  为半径的圆的内部,如图 5-1(a)所示。

这表明, $z$  平面上的单位圆按简单替换法的公式映射到  $s$  平面上是一个以  $(-1/T, j \cdot 0)$  为圆心, $1/T$  为半径的圆。换言之,通过简单替换法在  $s$  平面的左半平面上只有图 5-1(a)中的阴影部分才能映射到图 5-1(b) $z$  平面的单位圆内。

若某连续系统  $G(s)$  的极点分布如图 5-1(a)所示,显然  $G(s)$  是稳定的。经过替换后,所得的  $G(z)$  将有 3 个极点在单位圆内,两个极点在单位圆外,所以  $G(z)$  是不稳定的。一个原来稳定的系统经过变换得到的仿真模型  $G(z)$  却不稳定,这说明用简单替换法建立的仿真模型失真较大。

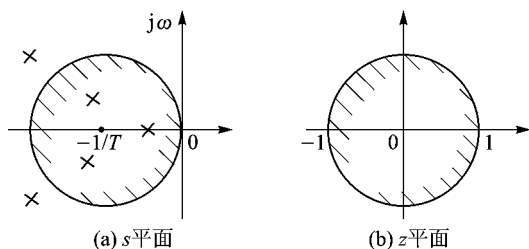


图 5-1 简单替换法的映射关系

从图 5-1 可知,为了使  $G(z)$  稳定,可增大  $1/T$  来达到这一目的;但这就要减小  $T$ ,违背了离散化仿真的基本要求,因此这种简单替换法不实用。

比较简单而且实用的替换法是双线性变换法,又称 Tustin 法。

### 5.1.2 双线性变换法

首先给出双线性变换法的变换公式。

将式(5-2)中的  $\ln z$  展成无穷级数

$$\ln z = 2 \left[ \frac{z-1}{z+1} + \frac{1}{3} \frac{(z-1)^3}{(z+1)^3} + \dots + \frac{1}{(2n-1)} \frac{(z-1)^{2n-1}}{(z+1)^{2n-1}} + \dots \right]$$

若只取级数的第一项,则可以得到从  $s$  平面到  $z$  平面的一种近似映射关系

$$s = \frac{2}{T} \frac{z-1}{z+1} \quad (5-12)$$

$$z = \frac{1 + \frac{T s}{2}}{1 - \frac{T s}{2}} \quad (5-13)$$

这就是双线性变换法  $s$  平面与  $z$  平面的映射关系。

将式(5-9)代入式(5-13),得



$$z = \frac{1 + \frac{T}{2}(\sigma + j\omega)}{1 - \frac{T}{2}(\sigma + j\omega)}$$

$$|z|^2 = \frac{\left(1 + \frac{T}{2}\sigma\right)^2 + \left(\frac{T}{2}\omega\right)^2}{\left(1 - \frac{T}{2}\sigma\right)^2 + \left(\frac{T}{2}\omega\right)^2} \quad (5-14)$$

由式(5-14)可知

$$\sigma < 0 \text{ 时, } |z| < 1$$

$$\sigma = 0 \text{ 时, } |z| = 1 \quad \sigma > 0 \text{ 时, } |z| > 1$$

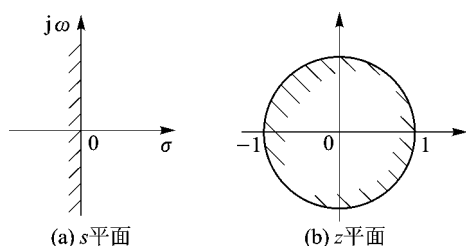


图 5-2 双线性变换法的映射关系

即,  $s$  平面的左半平面映射到  $z$  平面的单位圆内部,  $s$  平面的虚轴映射到  $z$  平面的单位圆周, 而  $s$  平面的右半平面映射到  $z$  平面的单位圆外部, 如图 5-2 所示。

这种映射关系说明, 若原系统  $G(s)$  的根都在  $s$  平面的左半平面, 则无论采样周期  $T$  取多大, 双线性变换后所得的脉冲传递函数  $G(z)$  的根必然都在  $Z$  平面的单位圆内, 这就保证了

稳定的连续系统不会因离散化而变得不稳定。所以, 双线性变换法是一种恒稳的方法, 允许采用大的步长或大的采样周期, 这是双线性变换的重要优点, 也是这一方法能在实践中得到广泛应用的主要原因。

双线性变换法还具有以下特点:

(1) 由  $G(s)$  经双线性变换得到的  $G(z)$ , 其分子、分母阶次相同, 且稳态增益不变。

设

$$G(s) = \frac{B_0 s^m + B_1 s^{m-1} + \cdots + B_{m-1} s + B_m}{A_0 s^n + A_1 s^{n-1} + \cdots + A_{n-1} s + A_n}$$

式中,  $n \geq m$ 。进行双线性变换, 有

$$G(z) = \frac{B_0 \left(\frac{2}{T} \frac{z-1}{z+1}\right)^m + B_1 \left(\frac{2}{T} \frac{z-1}{z+1}\right)^{m-1} + \cdots + B_{m-1} \left(\frac{2}{T} \frac{z-1}{z+1}\right) + B_m}{A_0 \left(\frac{2}{T} \frac{z-1}{z+1}\right)^n + A_1 \left(\frac{2}{T} \frac{z-1}{z+1}\right)^{n-1} + \cdots + A_{n-1} \left(\frac{2}{T} \frac{z-1}{z+1}\right) + A_n} \quad (5-15)$$

$$G(z) = \frac{B'_0}{A'_0} \left[ \frac{\prod_{i=1}^m (z - z_i)}{(z+1)^{n-m} \prod_{i=1}^n (z - p_i)} \right] \quad (5-16)$$

由式(5-16)知, 变换后在分子上增添了  $n-m$  个  $z = -1$  的零点,  $G(z)$  的分子、分母阶次相同,



均为  $n$  次。

$G(s)$  的稳态增益为  $B_m/A_n$ ,  $G(z)$  的稳态增益由式(5-15)知也为  $B_m/A_n$ , 所以变换前后的稳态增益不变。

(2) 双线性变换具有串联, 若令

$$G(s) = G_1(s) \cdot G_2(s)$$

则

$$G(z) = G_1(z) \cdot G_2(z)$$

式中,  $G(z), G_1(z), G_2(z)$  分别是  $G(s), G_1(s), G_2(s)$  的双线性变换结果。

双线性变换的这一特点允许用简单的低阶环节组成复杂的高阶系统, 从而给使用带来很大方便, 如图 5-3 所示。

(3) 双线性变换后得到的脉冲传递函数的频率特性与原连续系统的频率特性相接近, 特别是在低频段, 所以双线性变换主要用于一些有限带宽的网络。

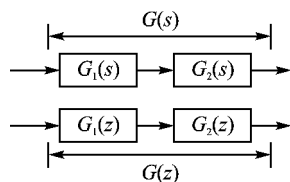


图 5-3 双线性变换的串联性

例 5.1.1 已知连续系统的传递函数为

$$G(s) = \frac{s}{s^2 + 2s + 1} \quad (5-17)$$

进行双线性变换, 将式(5-12)代入式(5-17)可得脉冲传递函数

$$G(z) = \frac{\frac{2}{T} \frac{z-1}{z+1}}{\left(\frac{2}{T} \frac{z-1}{z+1}\right)^2 + 2\left(\frac{2}{T} \frac{z-1}{z+1}\right) + 1} = \frac{2T(z-1)(z+1)}{[(T+2)z + (T-2)]^2} = \frac{2T}{(2+T)^2} \frac{(z-1)(z+1)}{\left(z - \frac{2-T}{2+T}\right)^2} \quad (5-18)$$

写出差分方程为

$$y_k = 2\left(\frac{2-T}{2+T}\right)y_{k-1} - \left(\frac{2-T}{2+T}\right)^2 y_{k-2} + \frac{2T}{(2+T)^2}(u_k - u_{k-2})$$

式中,  $u_k, u_{k-2}$  分别为  $k$  时刻和  $k-2$  时刻的输入值。

由式(5-18)可知, 因为

$$\left| \frac{2-T}{2+T} \right| < 1$$

所以,  $G(z)$  是稳定的。

$G(s)$  的分子为 1 阶, 分母为 2 阶; 而  $G(z)$  的分子、分母阶次相同, 均为 2 阶, 有一个  $z = -1$  处的零点。由终值定理可知,  $G(s)$  的稳态增益为 0,  $G(z)$  的稳态增益也为 0。

为了进一步考查仿真模型的精度, 下面来比较一下  $G(s)$  和  $G(z)$  的频率特性。



将  $s=j\omega$  和  $z=e^{j\omega T}$  分别代入式(5-17)和式(5-18),可得

$$G(j\omega) = \frac{j\omega}{(j\omega)^2 + 2(j\omega) + 1} = \frac{2\omega^2 + j(\omega - \omega^3)}{(1 + \omega^2)^2} \quad (5-19)$$

$$G(e^{j\omega T}) = \frac{2T}{(2+T)^2} \frac{(e^{j\omega T} - 1)(e^{j\omega T} + 1)}{\left(e^{j\omega T} - \frac{2-T}{2+T}\right)^2} \quad (5-20)$$

令  $T=1$  s, 将  $e^{j\omega T}$  用  $\cos \omega T + j\sin \omega T$  代替, 分别求出式(5-19)、(5-20)的幅频特性和相频特性, 如表 5-1 所列。

表 5-1 仿真模型与实际连续系统的频率特性比较

	$\omega/\text{rad}\cdot\text{s}^{-1}$	0.1	0.3	0.6	0.8	1.0	1.2	1.5
幅频特性	$ G(j\omega) $	0.099 01	0.275 2	0.441 2	0.487 8	0.5	0.491 8	0.461 5
	$ G(e^{j\omega T}) $	0.099 1	0.277	0.447	0.493	0.498	0.476	0.417
相频特性	$\angle G(j\omega)$	78.58°	56.60°	28.07°	12.68°	0°	-10.39°	-22.62°
	$\angle G(je^{j\omega T})$	78.57°	56.36°	26.51°	9.57°	-5.06°	-17.67°	-33.56°

比较表中的数据可知, 在  $T=1$  s 的条件下,  $G(s)$  和  $G(z)$  的频率特性在低于转折频率 (1 rad/s) 的频段内是十分接近的, 幅值相差不超过 1.5%, 相位相差不超过 5.06°。

图 5-4 为采样周期  $T=1$  s 和  $T=0.1$  s 时,  $G(s)$  和  $G(z)$  频率特性的比较, 可以看出, 在低频段, 二者的频率特性重合得很好。这说明, 用双线性变换所得的仿真模型既简单, 又能满足一定的精度要求。从图中还可看出, 采样周期越短,  $G(s)$  和  $G(z)$  频率特性重合的频段越宽,

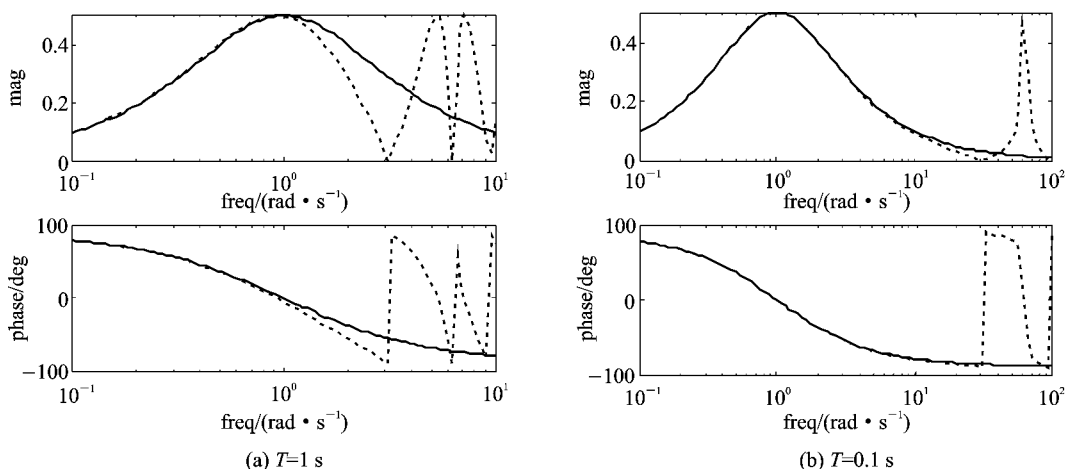


图 5-4  $G(s)$  和  $G(z)$  的频率特性比较 ( $T=1$  s 及  $T=0.1$  s)



基本与采样角频率相当。

## 5.2 根匹配法

根匹配法的基本思想是使离散化模型的动态特性和稳态值与原有连续系统保持一致。更明确地说,就是要使离散化所得的脉冲传递函数与原连续系统传递函数的零点和极点相匹配,所以,这种方法又称为零极点匹配法。根匹配法也是一种从系统传递函数  $G(s)$  直接推导出脉冲传递函数  $G(z)$  的常用离散化方法。

连续系统的动态性能和静态性能完全由其传递函数的零点位置、极点位置和增益大小所决定。如果能找到一种简单的映射关系  $z=f(s)$ , 将连续系统  $G(s)$  的零极点映射为脉冲传递函数  $G(z)$  的零极点,再根据一定的原则确定  $G(z)$  的增益,则能很快地求出与  $G(s)$  相对应的  $G(z)$ ,使二者的动态特性和静态特性保持一致。

如式 5-1 所示转换关系,可确定与  $s$  平面的零极点相对应的  $z$  平面零极点的位置。下面具体说明用根匹配法建立离散化模型的步骤,并对该方法的稳定性和精度问题进行讨论。

### 5.2.1 根匹配法的步骤

用根匹配法对连续系统进行离散化,第 1 步应先求出该系统的零点和极点,即将系统的传递函数变为如下的零极点形式

$$G(s) = \frac{Y(s)}{U(s)} = K \frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \quad (5-21)$$

第 2 步,由关系式(5-1)在  $z$  平面上——对应地确定零极点的位置。对于式(5-21)表示的系统,与  $z_1, z_2, \dots, z_m$  相对应的零极点为  $e^{z_1 T}, e^{z_2 T}, \dots, e^{z_m T}$  与  $p_1, p_2, \dots, p_n$  相对应的极点为  $e^{p_1 T}, e^{p_2 T}, \dots, e^{p_n T}$ , 得到

$$G(z) = K_z \frac{(z-e^{z_1 T})(z-e^{z_2 T})\cdots(z-e^{z_m T})}{(z-e^{p_1 T})(z-e^{p_2 T})\cdots(z-e^{p_n T})} \quad (5-22)$$

第 3 步,在  $G(z)$  的分子上配上  $n-m$  个附加零点,使  $G(z)$  的分子、分母阶次相同。由连续系统的根轨迹可知,当  $n>m$ ,即连续系统分母的阶次高于分子时,在  $s$  平面的负无穷远处实际上还存在着  $n-m$  个零点。因此,在  $z$  平面上必须再配上  $n-m$  个相应的零点。如果认为零点位于  $s$  平面负实轴的无穷远处,即

$$s = -\infty$$

那么由

$$z = e^{-\infty T} = 0$$

知, $z$  平面上相应的零点应配在原点处。于是脉冲传递函数为



$$G(z) = K_z \frac{(z - e^{z_1 T})(z - e^{z_2 T}) \cdots (z - e^{z_m T}) z^{n-m}}{(z - e^{p_1 T})(z - e^{p_2 T}) \cdots (z - e^{p_n T})} \quad (5-23)$$

第4步,在相同输入的作用下,应用终值定理,分别计算原连续系统和离散化仿真模型的终值,然后根据终值相等的原则确定  $G(z)$  的增益系数  $K_z$ 。

例 5.2.1 设一连续系统的微分方程为

$$\begin{aligned} T_1 \dot{y}(t) + y(t) &= f(t) \\ y(0) &= 0 \end{aligned}$$

用根匹配法确定离散化模型。

首先写出系统的传递函数

$$G(s) = \frac{1}{T_1 s + 1} = \frac{1}{T_1 (s + 1/T_1)}$$

有一个极点,  $p_1 = -1/T_1$ , 一个无穷远处的零点。

根据式(5-23)可得脉冲传递函数

$$G(z) = K_z \frac{z}{(z - e^{-T/T_1})}$$

式中,  $T$  为采样周期。

用终值定理分别求出  $G(s)$  和  $G(z)$  在单位阶跃作用下的终值,进而确定  $K_z$ 。

根据拉氏变换的终值定理知

$$\begin{aligned} \lim_{s \rightarrow 0} s G(s) \frac{1}{s} &= \lim_{s \rightarrow 0} s \frac{1}{T_1 s + 1} \frac{1}{s} = 1 \\ \lim_{z \rightarrow 1} \frac{z-1}{z} G(z) \frac{z}{z-1} &= \lim_{z \rightarrow 1} \frac{z-1}{z} \frac{K_z z}{z - e^{-T/T_1}} \frac{z}{z-1} = \frac{K_z}{1 - e^{-T/T_1}} \end{aligned}$$

令两终值相等,有

$$\frac{K_z}{1 - e^{-T/T_1}} = 1$$

则增益  $K_z$  为

$$K_z = 1 - e^{-T/T_1}$$

最后,求得的离散化模型为

$$G(z) = \frac{(1 - e^{-T/T_1})z}{(z - e^{-T/T_1})} \quad (5-24)$$

根据  $G(z)$  可进一步求出差分方程

$$y_k = e^{-T/T_1} y_{k-1} + (1 - e^{-T/T_1}) f_k \quad (5-25)$$

式(5-25)可直接用于仿真计算。



### 例 5.2.2 二阶系统的传递函数为

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (0 < \xi < 1)$$

用根匹配法求离散化模型。

首先,将原系统的传递函数写为零极点形式

$$G(s) = \frac{\omega_n^2}{(s - p_1)(s - p_2)}$$

式中

$$p_{1,2} = -\xi\omega_n \pm j\omega_n \sqrt{1 - \xi^2}$$

根据式(5-1),得到  $z$  平面上的两个极点

$$z_1 = e^{p_1 T} = e^{-\xi\omega_n T} e^{j\omega_n T \sqrt{1 - \xi^2}}$$

$$z_2 = e^{p_2 T} = e^{-\xi\omega_n T} e^{-j\omega_n T \sqrt{1 - \xi^2}}$$

于是

$$G(z) = \frac{K_z}{z^2 - [2e^{-\xi\omega_n T} \cos(\omega_n T \sqrt{1 - \xi^2})]z + e^{-2\xi\omega_n T}}$$

原系统有两个无穷远处的零点,相应地应在  $z$  平面配上两个原点处的零点,则脉冲传递函数变为

$$G(z) = \frac{K_z z^2}{z^2 - [2e^{-\xi\omega_n T} \cos(\omega_n T \sqrt{1 - \xi^2})]z + e^{-2\xi\omega_n T}}$$

根据终值定理分别求单位阶跃下的终值为

$$\lim_{s \rightarrow 0} sG(s) = \frac{1}{s} = 1$$

$$\lim_{z \rightarrow 1} \frac{z-1}{z} G(z) \frac{z}{z-1} = \frac{K_z}{1-a+b}$$

式中

$$a = 2e^{-\xi\omega_n T} \cos(\omega_n T \sqrt{1 - \xi^2})$$

$$b = e^{-2\xi\omega_n T}$$

令两终值相等,得

$$\frac{K_z}{1-a+b} = 1$$

于是

$$K_z = 1 - a + b$$

最后得离散化模型为

$$G(z) = \frac{(1-a+b)z^2}{z^2 - az + b} = \frac{Y(z)}{U(z)}$$



用于仿真计算的差分方程为

$$y_k = ay_{k-1} - by_{k-2} + (1-a+b)u_k$$

上面的例子都是用阶跃输入的终值来确定增益系数  $K_z$ 。再来看阶跃响应的终值为零时如何确定  $K_z$ 。

例 5.2.3 一个连续系统的传递函数为

$$G(s) = \frac{Y(s)}{U(s)} = \frac{s}{(s+1)^2}$$

用根匹配法求离散化模型。

根据式(5-1)及  $G(s)$  的零极点可得

$$G(z) = K_z \frac{z-1}{(z-e^{-T})^2}$$

再配上一个  $z=0$  处的零点

$$G(z) = K_z \frac{(z-1)z}{(z-e^{-T})^2}$$

在确定  $K_z$  时注意到,由于在  $s$  平面上存在一个原点处的零点,若加阶跃输入,终值将为零。为使终值取得非零的有限值,应当加斜坡输入,即

$$u(t) = t, \quad U(s) = 1/s^2$$

根据终值定理可知

$$\lim_{s \rightarrow 0} sG(s) \frac{1}{s^2} = \lim_{s \rightarrow 0} s \frac{s}{(s+1)^2} \frac{1}{s^2} = 1$$

$$\lim_{z \rightarrow 1} \frac{z-1}{z} G(z) \frac{Tz}{(z-1)^2} = \lim_{z \rightarrow 1} \frac{z-1}{z} \frac{K_z(z-1)z}{(z-e^{-T})^2} \frac{Tz}{(z-1)^2} = \frac{K_z T}{(1-e^{-T})^2}$$

为使两个终值相等,就要求

$$K_z = \frac{(1-e^{-T})^2}{T}$$

于是离散化模型为

$$G(z) = \frac{(1-e^{-T})^2}{T} \frac{(z-1)z}{(z-e^{-T})^2} \quad (5-26)$$

### 5.2.2 根匹配法的稳定性和精度

根匹配法有一个突出的优点,即无论采样周期取多大,只要原系统稳定,则离散化模型一定稳定。若  $G(s)$  是稳定的,则它的全部极点都位于  $s$  平面的左半平面,即式(5-21)中的  $p_1, p_2, \dots, p_n$  都具有负实部,那么由式(5-1)所得的  $z$  平面上的极点  $e^{p_1 T}, e^{p_2 T}, \dots, e^{p_n T}$  必然都在单位圆内。所以,由式(5-23)所得的  $G(z)$  也一定是稳定的。这表明,用根匹配法建立的仿真模型本身不存在不稳定情况,即匹配法是一种恒稳的方法,且允许选用较大的采样周期进行仿



真。应当注意的是,这种方法只适用于线性稳定系统。

除了良好的稳定性之外,根匹配法还具有一定的精度。这里以例 5.2.3 中的系统为例,比较一下  $G(s)$  和  $G(z)$  的频率特性。

取采样周期  $T=1$  s,于是由式(5-26)得

$$G(z) = (1 - e^{-1}) \frac{(z-1)z}{(z - e^{-1})^2} = 0.4 \frac{z(z-1)}{(z - 0.3679)^2} \quad (5-27)$$

将  $G(s)$  和  $G(z)$  的幅频特性和相频特性列于表 5-2。

表 5-2 几种不同根匹配法的比较

特性	$\omega / (\text{rad} \cdot \text{s}^{-1})$	0.1	0.3	0.6	0.8	1.0	1.2	2.0
幅频特性	$ G(j\omega) $	0.099 01	0.275 2	0.441 2	0.487 8	0.5	0.491 8	0.4
	$ G(e^{j\omega T}) $	0.095 37	0.265 9	0.430 7	0.481 23	0.5	0.500 1	0.449
	$ G_1(e^{j\omega T}) $	0.108 5	0.299 6	0.468 8	0.505	0.5	0.47	0.276 5
	$ G_2(e^{j\omega T}) $	0.107	0.295 5	0.464	0.501	0.5	0.474	0.302 4
相频特性	$\angle G(j\omega)$	$78.58^\circ$	$56.6^\circ$	$28.07^\circ$	$12.68^\circ$	$0^\circ$	$-10.39^\circ$	$-36.87^\circ$
	$\angle G(e^{j\omega T})$	$80.5^\circ$	$62.37^\circ$	$39.6^\circ$	$27.9^\circ$	$19.10^\circ$	$12.45^\circ$	$0.35^\circ$
	$\angle G_1(e^{j\omega T})$	$77.64^\circ$	$53.78^\circ$	$22.41^\circ$	$5.08^\circ$	$-6.02^\circ$	$-6.55^\circ$	$-11.17^\circ$
	$\angle G_2(e^{j\omega T})$	$78.53^\circ$	$56.45^\circ$	$27.88^\circ$	$12.53^\circ$	$0.05^\circ$	$-9.96^\circ$	$-31.20^\circ$

比较表中  $G(s)$  与附加零点配在原点的离散化模型  $G(z)$  的特性可知,二者的幅频特性是比较接近的;而在相频特性上,离散化模型有较大的超前。比如在  $\omega=1$  rad/s 时,  $G(s)$  的相移为零度,而  $G(z)$  的相移为  $19.10^\circ$ 。一般地,如果  $T$  取得较大,计算精度还会降低。

为了改善根匹配法的精度,可将  $n-m$  个附加零点配在  $z=-1$  处,其根据是,在双线性变换法中有

$$\frac{1}{s} = \frac{T}{2} \frac{z+1}{z-1}$$

成立,该式就是在分子上乘了一项  $(z+1)$ ,相当于在  $z=-1$  处配了一个附加零点。这样  $G(z)$  就变为

$$G(z) = K_z \frac{(z - e^{s_1 T})(z - e^{s_2 T}) \cdots (z - e^{s_m T})(z + 1)^{n-m}}{(z - e^{p_1 T})(z - e^{p_2 T}) \cdots (z - e^{p_n T})} \quad (5-28)$$

仍以例 5.2.3 中的  $G(s)$  为例,可得脉冲传递函数

$$G_1(z) = K_z \frac{(z-1)(z+1)}{(z - e^{-T})^2}$$



令斜坡输入作用下的终值相等,可确定增益

$$K_z = \frac{(1 - e^{-T})^2}{2T}$$

仍取  $T=1$  s,则  $K_z=0.2$ ,于是

$$G_1(z) = 0.2 \frac{(z-1)(z+1)}{(z-0.3679)^2} \quad (5-29)$$

将式(5-29)的幅频特性与相频特性也列于表 5-2 中。

比较表中的数据可知,  $G(z)$  附加零点配在  $z=-1$  时,相位有一些滞后,如  $\omega=1$  rad/s 时,  $G(s)$  相位为  $0^\circ$ ,但  $G_1(z)$  的相位为  $-6.02^\circ$ 。

为了使离散化模型的相位既不超前也不滞后,可以将附加零点配在  $(0, -1)$  之间,即在  $G(z)$  的分子上乘以  $(z+\sigma)^{n-m}$ ,其中  $\sigma$  满足

$$0 < \sigma < 1$$

则  $G(z)$  变为

$$G(z) = K_z \frac{(z - e^{s_1 T})(z - e^{s_2 T}) \cdots (z - e^{s_m T})(z + \sigma)^{n-m}}{(z - e^{p_1 T})(z - e^{p_2 T}) \cdots (z - e^{p_n T})} \quad (5-30)$$

对于例 5.2.3 的系统,就有

$$G_2(z) = K_z \frac{(z-1)(z+\sigma)}{(z-e^{-T})^2} \quad (5-31)$$

式中,包含两个待定参数  $K_z$  和  $\sigma$ ,它们可以通过频率特性来确定。具体方法是:首先求出当  $\omega=1$  rad/s 时,  $G(s)$  的幅值和相位  $|G(j\omega)|$ ,  $\angle G(j\omega)$ ;其次求出  $\omega=1$  rad/s 时,  $G_2(z)$  的幅值和相位  $|G_2(e^{j\omega T})|$ ,  $\angle G_2(e^{j\omega T})$ 。为使  $G(s)$  与  $G_2(z)$  相匹配,令

$$|G(j\omega)| = |G_2(e^{j\omega T})| \quad (5-32)$$

$$\angle G(j\omega) = \angle G_2(e^{j\omega T}) \quad (5-33)$$

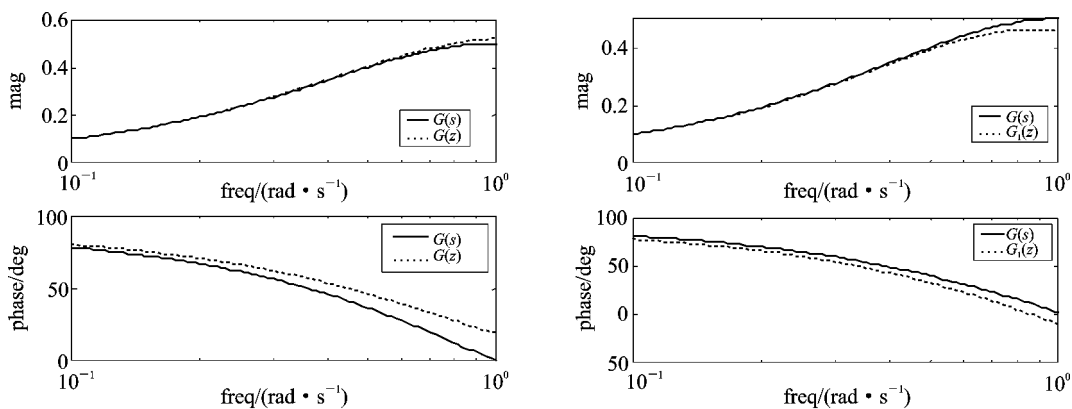
由上述两式即可确定  $K_z$  和  $\sigma$ 。

用这种方法求出例 5.2.3 中的  $K_z=0.2828$ ,  $\sigma=0.5272$ ,则离散化模型为

$$G_2(z) = 0.2828 \frac{(z-1)(z+0.5272)}{(z-0.3679)^2} \quad (5-34)$$

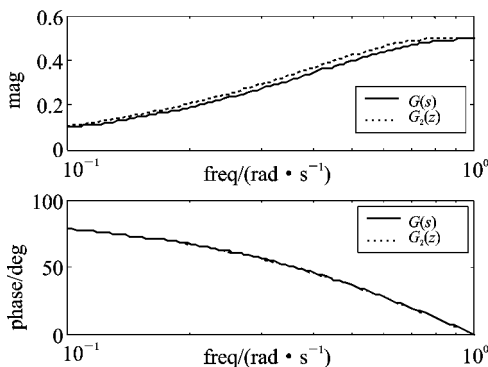
将其频率特性也列在表 5-2 中。从表中数据可知,附加零点配在  $z=-0.5272$  处,幅值误差和相位误差都很小。

图 5-5 是  $G(s)$  与  $G(z)$ 、 $G_1(z)$ 、 $G_2(z)$  的频率特性,可见,当附加零点位于原点时,离散化模型的相位超前于原系统;当附加零点位于  $-1$  点时,离散化模型的相位滞后于原系统;当附加零点位于实轴的  $(-1, 0)$  之间时,离散化模型的相位与原系统最为接近。三者的幅值相差不大。



(a) 附加零点位于原点

(b) 附加零点位于-1点



(c) 附加零点位于(-1, 0)区间

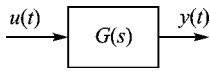
图 5-5  $G(s)$  与  $G(z)$ 、 $G_1(z)$ 、 $G_2(z)$  的频率特性比较

### 5.3 离散相似法

本章介绍的仿真方法都是先将连续系统的数学模型进行离散化处理,把连续系统模型近似等价为一个离散系统模型,比如将传递函数  $G(s)$  变为脉冲传递函数  $G(z)$ ,或将微分方程变为差分方程。这一节从连续系统离散化的角度出发,用采样系统的理论和方法介绍另一种常用的离散化仿真方法,这就是离散相似法。

#### 5.3.1 离散相似法的含义和特点

在图 5-6 所示的输入为  $u(t)$ 、输出为  $y(t)$  的连续系统中,用一采样周期为  $T$  的采样开关将输入、输出分别离散化,成为图 5-7 所示的形式,要求输出  $y^*(t)$  在采样时刻的值等于原输



出  $y(t)$  在同一时刻的值, 即有

$$y^*(t) = y(kT), \quad t = kT \quad (5-35)$$

成立, 式中  $k=0, 1, 2, \dots$ 。

图 5-6 连续系统方块图

显然, 如果只是简单地在原系统的输入、输出端人为地加上一个采样开关, 输出  $y^*(t)$  很难与原系统的输出  $y(t)$  相同。因为输入  $u(t)$  经过采样开关后变为离散信号  $u^*(t)$ , 若将  $u^*(t)$  直接输入原系统, 其输出当然不会再保持原来的变化规律, 这样就只有离散而谈不上相似。

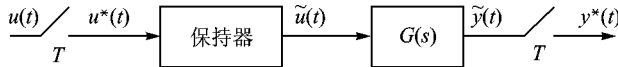


图 5-7 离散化模型

为了使输入信号不失真, 需要再加一个保持器, 其目的是使输入信号在采样间隔仍保持连续性。由于保持器所能延续的规律不一定与原输入信号  $u(t)$  完全一致, 因此实际加到连续系统去的输入信号  $\tilde{u}(t)$  一般不等同于  $u(t)$ , 而只是近似相同。

由以上分析可知, 连续系统经过离散相似法得到的仿真模型必然具有一定的近似性, 其近似程度取决于采样周期和保持器的特性。必须明确的是, 用离散相似法对连续系统进行数字仿真时, 所加的采样开关和保持器都是虚拟的, 是为了对系统进行离散化处理而采取的手段, 在实际系统中并没有相应的物理装置。

下面通过几个例子来说明离散相似法仿真的具体步骤。

例 5.3.1 已知连续系统的传递函数为

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k}{s+a} \quad (5-36)$$

首先用一采样周期为  $T$  的开关和零阶保持器将其离散化, 如图 5-8 所示。

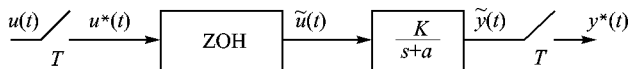


图 5-8 连续系统离散化

然后用  $z$  变换求其脉冲传递函数

$$G(z) = \frac{Y(z)}{U(z)} = Z\left(\frac{1-e^{-Ts}}{s} \cdot \frac{k}{s+a}\right) = (1-z^{-1})Z\left[\frac{k}{s(s+a)}\right] = \frac{k}{a} \frac{1-e^{-aT}}{z-e^{-aT}} \quad (5-37)$$

仿真所用的差分方程为

$$y(k+1) = e^{-aT}y(k) + \frac{k}{a}(1-e^{-aT})u(k) \quad (5-38)$$

例 5.3.2 设二阶连续系统的传递函数为



$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad \xi < 1 \quad (5-39)$$

采样周期为  $T$ , 选用零阶保持器, 求仿真所用的差分方程。

解: 脉冲传递函数由  $z$  变换求得

$$G(z) = \frac{Y(z)}{U(z)} = (1 - z^{-1})Z\left[\frac{\omega_n^2}{s(s^2 + 2\xi\omega_n s + \omega_n^2)}\right] = (1 - z^{-1})Z\left[\frac{1}{s} - \frac{s + 2\xi\omega_n}{(s + \xi\omega_n)^2 + (\omega_n^2 - \xi^2\omega_n^2)}\right] \quad (5-40)$$

令

$$a = \xi\omega_n, \quad \omega = (1 - \xi^2)\omega_n^2$$

则式(5-40)变为

$$G(z) = 1 - \frac{z-1}{z} \frac{z^2 - ze^{-aT} \cos \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}} - \frac{z-1}{z} \frac{ze^{-aT} \sin \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}} \frac{\xi}{\sqrt{1 - \xi^2}} \quad (5-41)$$

令

$$b = \xi/\sqrt{1 - \xi^2}$$

进一步化简式(5-41), 可得

$$G(z) = 1 - \frac{(z-1)(z - e^{-aT} \cos \omega T) + (z-1)be^{-aT} \sin \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}} = \frac{z(1 - e^{-aT} \cos \omega T - be^{-aT} \sin \omega T) + (e^{-2aT} - e^{-aT} \cos \omega T + be^{-aT} \sin \omega T)}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}} \quad (5-42)$$

令

$$B_1 = 1 - e^{-aT} \cos \omega T - be^{-aT} \sin \omega T$$

$$B_2 = e^{-2aT} - e^{-aT} \cos \omega T + be^{-aT} \sin \omega T$$

可写出差分方程

$$y(k+1) = 2e^{-aT} \cos \omega T \cdot y(k) - e^{-2aT} y(k-1) + B_1 u(k-1) + B_2 u(k-2)$$

用同样的方法和步骤可求出各种连续系统的离散化模型。

### 5.3.2 离散相似法的精度和稳定性

通过离散相似法所得到的仿真模型只是近似等效于原来的连续系统, 影响仿真模型精度和稳定性的因素主要是采样周期  $T$  和保持器的特性。为了保证仿真精度和仿真模型的稳定, 需要研究离散相似法的精度和稳定性。

#### 1. 采样周期对精度的影响

采用离散相似法时, 在连续系统中加入虚拟的采样开关和保持器, 把连续系统变为离散系统。根据采样定理(Shannon 定理), 如果采样频率  $\omega_s$  大于或等于被采样信号最高频率  $\omega_{\max}$  的两倍以上, 即满足

$$\omega_s \geq 2\omega_{\max} \quad (5-43)$$



则经过采样后采样函数的离散频谱互相不混叠,如图 5-9(a)所示。若不满足式(5-43)的条件,离散频谱就会出现混叠,如图 5-9(b)所示。

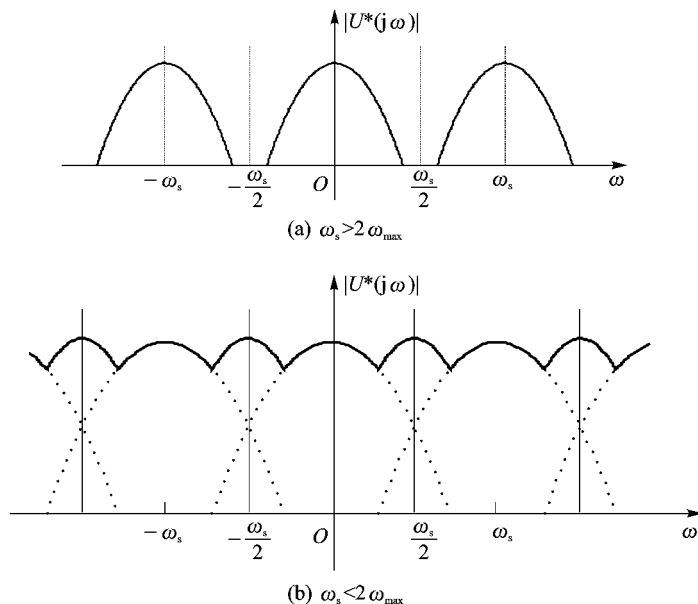


图 5-9 离散信号频谱

只有在离散信号的频谱互相之间没有混叠时,才有可能无失真地将离散信号恢复为连续信号。所以,为了保证仿真精度,采样周期必须满足采样定理的要求,即满足式(5-43)。在实际应用离散相似法时,目前还没有适用于各种情况的确定采样周期的公式,但有一些可供参考的经验公式。一般地,为了使仿真在开始阶段有足够的精度,建议按系统最小时间常数的 1/10 选择采样周期,即

$$T = \frac{T_{\min}}{10} \quad (5-44)$$

或者按照系统开环频率特性的剪切频率  $\omega_c$  来确定采样周期,即

$$T = \frac{1}{(30 \sim 50)\omega_c} \quad (5-45)$$

若系统中有若干个小闭环时,  $\omega_c$  应为反应最快的小闭环的开环频率特性的剪切频率。

若按经验公式(5-44)和式(5-45)选择采样周期,仿真精度约为 0.5%。

从图 5-9 的频谱可知,当式(5-43)成立时,采样函数的离散频谱互相不混叠,此时若将离散信号  $u^*(t)$ (见图 5-9)加到具有图 5-10 所示的理想滤波特性的低通滤波器上,则能保留主要频谱,而滤掉附加的高频分量,连续信号  $u(t)$  可以无失真地得到复现。

对图 5-9 所示的  $\omega_s < 2\omega_{\max}$  时的离散信号频谱,即使通过理想滤波器,在滤波器的输出端



也无法得到与原连续信号  $u(t)$  相似的信号。

图 5-10 给出的锐截止特性的理想滤波器实际上是不存在的,但在实际中广泛应用的保持器具有类似的特性。下面讨论保持器对仿真模型的影响。

## 2. 保持器对精度和稳定性的影响

保持器是一种在时域内的外推装置,它使采样信号在采样间隔仍保持连续性。从频域来看,它则是把离散化产生的高频分量滤掉,只保留主频部分。由于保持器不是理想的滤波器,有一定的幅值衰减和相位滞后,因而会对仿真模型的精度和稳定性产生一定的影响。

常用的保持器有零阶保持器、一阶保持器和三角形保持器。

### (1) 零阶保持器

零阶保持器是一种按常值规律外推的保持器。

对图 5-8 所示的系统,若选用零阶保持器,则有

$$\tilde{u}(t) = u(kT), \quad kT \leq t < (k+1)T \quad (5-46)$$

零阶保持器的传递函数为

$$H_0(s) = \frac{1 - e^{-Ts}}{s} \quad (5-47)$$

其幅频特性和相频特性分别为

$$|H_0(j\omega)| = \frac{2 \sin \frac{\omega T}{2}}{\omega} \quad (5-48)$$

$$\angle H_0(j\omega) = -\frac{\omega T}{2} \quad (5-49)$$

图 5-11 给出了零阶保持器的频率特性曲线,并同时画出理想滤波器的曲线。从图中可知,其幅值随着频率的增加而减小,且频率越高衰减越快,具有明显的低通滤波特性,但零阶保持器具有无穷多个截止频率,因此不是理想的低通滤波器。它除了允许离散频谱的主频分量通过,还允许高频分量部分通过。因此,由零阶保持器恢复所得的连续信号与原连续信号是有差别的,如图 5-12 所示。

由零阶保持器恢复的信号  $\tilde{u}(t)$  具有阶梯形状,从而形成  $u(t)$  与  $\tilde{u}(t)$  之间的差别。采样周期越小,上述差别就会越小。

从式(5-49)可知,由于零阶保持器在离散化模型中引进了附加的相位滞后,从而使系统的稳定性有所降低。同时,由于相位滞后的存在,经零阶保持器恢复的信号比原来的连续信号在时间上平均滞后半个采样周期。图 5-12 给出了零阶保持器的输出信号与原来的连续信号的比较。图中实线为连续信号  $u(t)$ ,虚线为  $\tilde{u}(t)$ 。

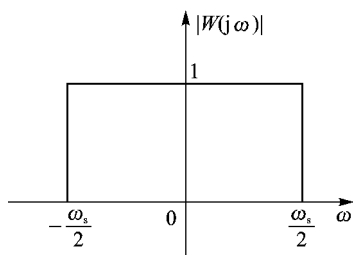


图 5-10 理想滤波特性

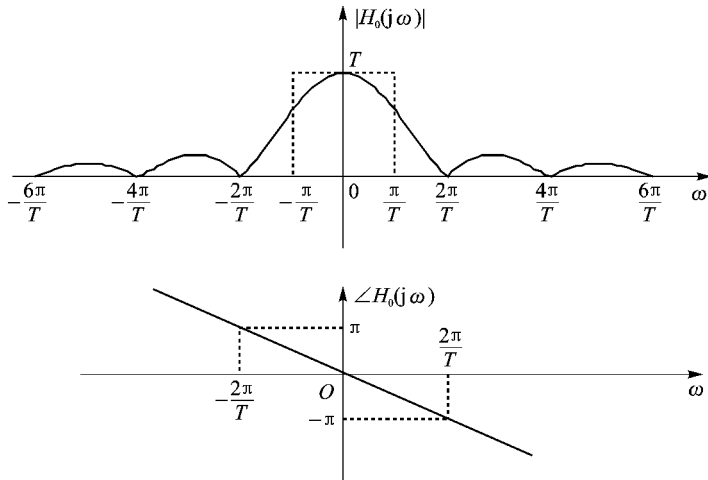


图 5-11 零阶保持器的频率特性

## (2) 一阶保持器

一阶保持器是基于两个采样时刻的值按线性外推规律恢复离散信号的保持器, 即有

$$\tilde{u}(t) = \tilde{u}(kT + \tau) = u(kT) + \frac{u(kT) - u[(k-1)T]}{T} \tau = u(kT) + \dot{u}(kT) \tau \quad (5-50)$$

成立。式中的  $\tau$  满足

$$0 < \tau < T$$

一阶保持器的传递函数为

$$H_1(s) = T(1 + Ts) \left( \frac{1 - e^{-Ts}}{Ts} \right)^2 \quad (5-51)$$

幅频特性和相频特性分别为

$$|H_1(j\omega)| = T \sqrt{1 + (T\omega)^2} \left[ \frac{\sin \frac{\omega T}{2}}{\frac{\omega T}{2}} \right]^2 \quad (5-52)$$

$$\angle H_1(j\omega) = \arctan \omega T - \omega T \quad (5-53)$$

图 5-13 给出了一阶保持器的频率特性曲线, 为了便于比较, 在图中同时用虚线绘出了零阶保持器的特性。

从图 5-13 可见, 一阶保持器与零阶保持器类似, 其幅频特性第一次衰减到零的频率, 即第一个截止频率也是  $2\pi/T$ , 即  $\omega_s$ , 但一阶保持器的幅值比零阶保持器高, 在各频段都是如此; 它的高频分量更大, 这意味着离散频谱中的高频成分更容易通过一阶保持器。相对来说, 一阶

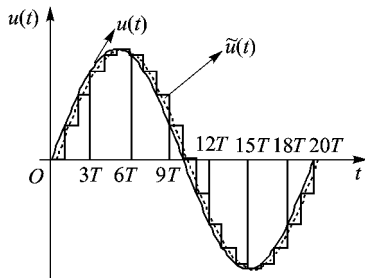


图 5-12 零阶保持器的输出信号



保持器能较好地复现斜坡信号,而零阶保持器能较好地复现阶跃信号。从图 5-13 所示的相频特性可以看出,一阶保持器的滞后比零阶保持器大,其平均相移约为零阶保持器平均相移的 2 倍。正由于这一原因,采用离散相似法对连续系统进行离散化仿真时,一般不采用一阶保持器,更不采用高阶保持器,而普遍采用零阶保持器。

### (3) 三角形保持器

对于阶跃或斜坡之外的更一般的连续信号,采用三角形保持器可减少恢复信号的失真。对图 5-8 所示系统,若采用三角形保持器,则有

$$\tilde{u}(t) = \tilde{u}(kT + \tau) = u(kT) + \frac{u[(k+1)T] - u(kT)}{T} \tau \quad (5-54)$$

式中

$$0 < \tau < T$$

三角形保持器的传递函数为

$$H_3(s) = \frac{(1 - e^{-Ts})^2 e^{Ts}}{Ts^2} \quad (5-55)$$

幅频特性和相频特性分别为

$$|H_3(j\omega)| = \frac{2 - 2\cos(\omega T)}{T\omega^2} \quad (5-56)$$

$$\angle H_3(j\omega) = 0 \quad (5-57)$$

图 5-14 为频率特性曲线。从图中可知,三角形保持器的幅频特性较好,在高频段幅值很小,同时相移为零,无相位失真。但是这种保持器在计算  $kT$  到  $(k+1)T$  之间的信号值时,要用到  $u[(k+1)T]$ ,有时该值不能获得,所以实际上能够采用的是滞后一拍的三角形保持器<sup>[2]</sup>,即

$$\tilde{u}(t) = u[(k-1)T] + \frac{u(kT) - u[(k-1)T]}{T}(t - kT) \quad (5-58)$$

式中

$$kT \leq t \leq (k+1)T$$

其传递函数为

$$H'_3(s) = \frac{(1 - e^{-Ts})^2}{Ts^2} \quad (5-59)$$

幅频特性和相频特性分别为

$$|H'_3(j\omega)| = \frac{2 - 2\cos(\omega T)}{T\omega^2} \quad (5-60)$$

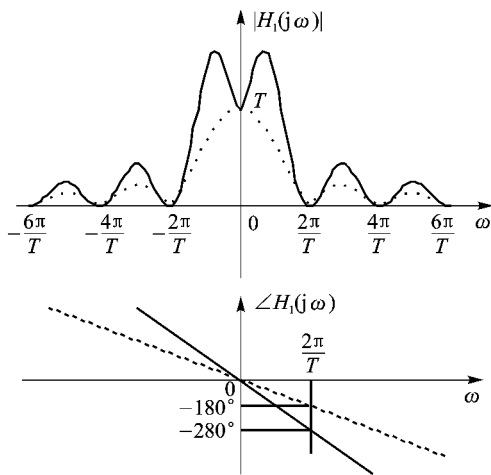


图 5-13 一阶保持器的频率特性



$$\angle H'_3(j\omega) = -\omega T \quad (5-61)$$

频率特性曲线如图 5-15 所示。

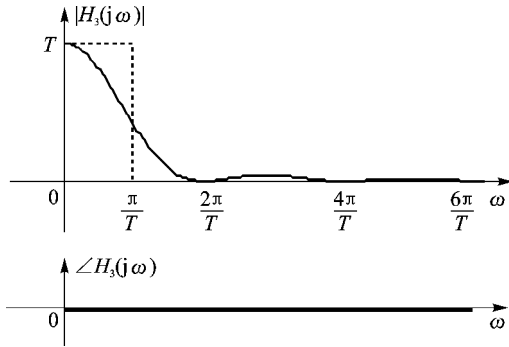


图 5-14 三角形保持器的频率特性

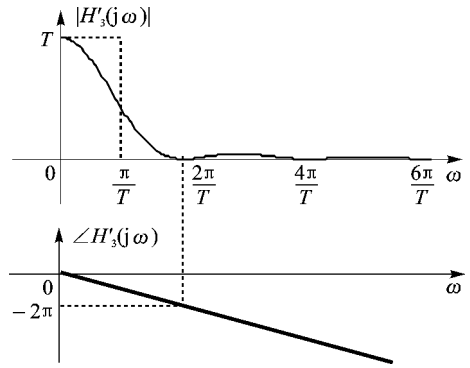


图 5-15 滞后一拍的三角形保持器

由上述分析可知,保持器对离散化模型的精度和稳定性有影响,特别是当系统由多个离散化环节组成时,这种相位滞后的影响更为严重。因此,在对系统进行离散化时,应设法减少保持器的数目。如图 5-16 所示,应尽量采用图 5-16(a)所示的只有一个保持器的形式。

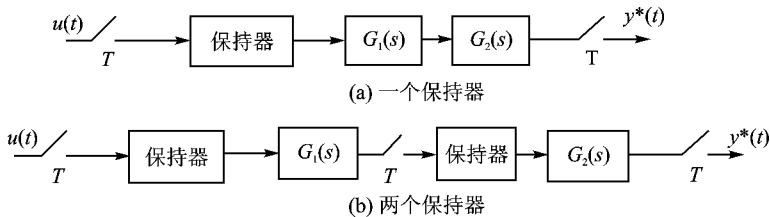


图 5-16 连续系统离散化

**例 5.3.3** 设图 5-16 中的  $G_1(s) = \frac{1}{0.1s+1}$ ,  $G_2(s) = \frac{1}{0.2s+1}$ , 采样周期  $T=0.1$  s, 采用离散相似法进行仿真, 求其单位阶跃响应。为了说明保持器对仿真精度的影响, 这里分别考察采用两个零阶保持器和一个零阶保持器对系统进行离散化的情况。

图 5-17 给出了两个 ZOH 情况下原连续系统及离散化之后系统的单位阶跃响应, 并比较了二者之间的误差。其中连续系统仿真采用 4 阶 Runge-Kutta 法。图 5-18 为采用一个 ZOH 时的单位阶跃响应曲线及仿真误差。

从图 5-18 中可以看出, 对系统进行离散化时, 采用一个保持器的形式, 仿真误差最大不超过小于  $7 \times 10^{-3}$ 。而当系统中有两个保持器时, 则保持器本身的特性对系统仿真模型的精度产生了较大影响, 导致仿真误差增大, 最大可达 0.16, 且收敛速度下降。

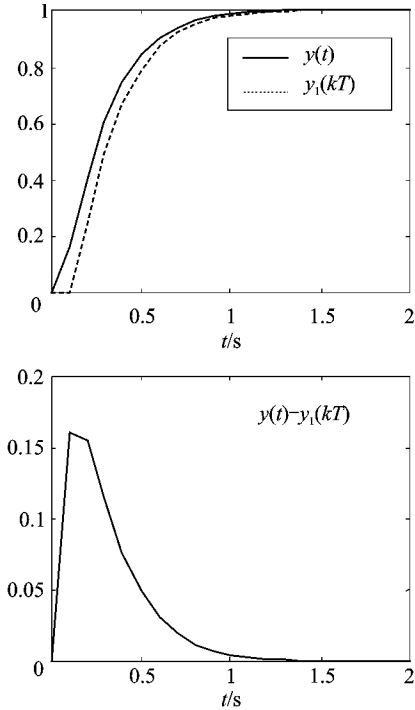


图 5-17 离散系统单位阶跃响应(两个保持器)

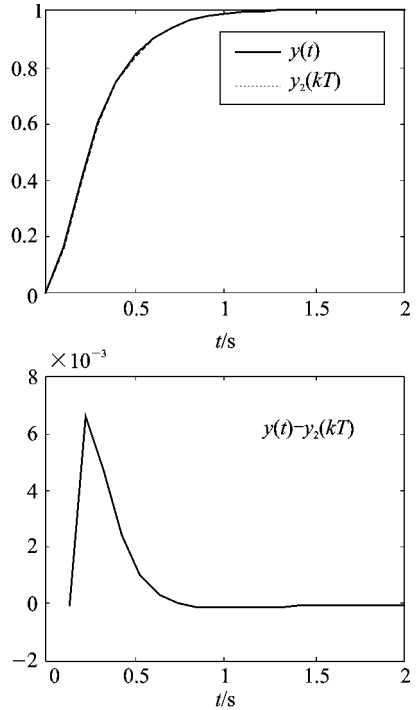


图 5-18 离散系统单位阶跃响应(一个保持器)

由于各种保持器的频率特性都与理想滤波器的频率特性差别较大,所以不可能将采样后的离散信号完全无失真地再现。但若采样频率足够高,并采用较为合适的保持器,则可将离散化引起的信号失真减到满足精度要求的程度。从数字仿真的角度来看,采样频率提高,就是要求计算步长减小。但由于差分方程的计算量较小,故仍可满足实时性要求。

### 5.4 状态方程的离散化

状态方程的离散化属于离散相似法,也需要在连续系统中加入虚拟的采样开关和保持器,只是在这里,连续系统的模型是用状态方程表示的,如图 5-19 所示。一般将传递函数离散化称为频域离散相似法,得到的是离散传递函数,即频域离散相似模型,而对状态方程进行离散化称为时域离散相似法,得到的是时域离散相似模型。

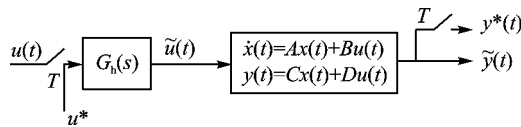


图 5-19 状态方程的离散化



### 5.4.1 离散化模型的推导

设系统的状态空间表达式为

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (5-62)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (5-63)$$

对上述两式进行拉氏变换可得

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{u}(s) \quad (5-64)$$

$$\mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{u}(s) \quad (5-65)$$

其中,  $\mathbf{x}(0)$  为状态变量初值。由式(5-64)可得

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{u}(s) \quad (5-66)$$

对式(5-66)进行拉氏反变换, 并利用卷积积分得

$$\begin{aligned} \mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau = \\ &\mathbf{F}(t)\mathbf{x}(0) + \int_0^t \mathbf{F}(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau \end{aligned} \quad (5-67)$$

式中,  $\mathbf{F}(t) = e^{\mathbf{A}t}$ , 称为系统的状态转移矩阵, 描述了状态变量  $\mathbf{x}(t)$  由初始状态  $t=0$  时刻向任一时刻  $t$  转移的特性。

式(5-67)是连续系统状态方程的时域解, 下面对这个解进行离散化, 以期得到状态方程的离散化模型。

设采样周期为  $T$ , 考察  $kT, (k+1)T$  两时刻  $\mathbf{x}(t)$  的值。将  $t=(k+1)T$  代入式(5-67), 得

$$\begin{aligned} \mathbf{x}[(k+1)T] &= e^{\mathbf{A}(k+1)T}\mathbf{x}(0) + \int_0^{(k+1)T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau = \\ &e^{\mathbf{A}T}[e^{\mathbf{A}kT}\mathbf{x}(0) + \int_0^{kT} e^{\mathbf{A}(kT-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + \int_{kT}^{(k+1)T} e^{\mathbf{A}(kT-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau] \end{aligned}$$

可见, 式中的  $e^{\mathbf{A}kT}\mathbf{x}(0) + \int_0^{kT} e^{\mathbf{A}(kT-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$  为  $\mathbf{x}(kT)$ , 故可将上式写成如下递推形式

$$\mathbf{x}[(k+1)T] = e^{\mathbf{A}T}\mathbf{x}(kT) + e^{\mathbf{A}T}\int_{kT}^{(k+1)T} e^{\mathbf{A}(kT-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \quad (5-68)$$

若离散化时对输入  $u$  采用零阶保持器, 即有

$$\mathbf{u}(t) = \mathbf{u}(kT), \quad kT \leq t < (k+1)T$$

于是式(5-68)变为

$$\mathbf{x}[(k+1)T] = e^{\mathbf{A}T}\mathbf{x}(kT) + \int_{kT}^{(k+1)T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\mathbf{u}(kT)d\tau$$

令积分变量为  $\tau = \tau - kT$ , 可得

$$\mathbf{x}(k+1) = e^{\mathbf{A}T}\mathbf{x}(k) + \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{B}d\tau\mathbf{u}(k) =$$



$$\mathbf{F}(T)\mathbf{x}(k) + \mathbf{G}(T)\mathbf{u}(k) \quad (5-69)$$

式中

$$\mathbf{F}(t) = e^{At}$$

$$\mathbf{G}(T) = \int_0^T e^{A(T-\tau)} \mathbf{B} d\tau = \int_0^T \mathbf{F}(T-\tau) \mathbf{B} d\tau$$

这样就得到了采用零阶保持器的离散化模型

$$\mathbf{x}(k+1) = \mathbf{F}(T)\mathbf{x}(k) + \mathbf{G}(T)\mathbf{u}(k) \quad (5-70)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (5-71)$$

采用一阶保持器时,有

$$\mathbf{u}(t) = \mathbf{u}(kT) + \dot{\mathbf{u}}(kT)(t - kT), \quad kT \leq t < (k+1)T$$

代入式(5-68),整理可得

$$\mathbf{x}(k+1) = e^{AT}\mathbf{x}(k) + \int_0^T e^{A(T-\tau)} \mathbf{B} [\mathbf{u}(k) + \dot{\mathbf{u}}(k)\tau] d\tau \quad (5-72)$$

令

$$\mathbf{G}_1(T) = \int_0^T e^{A(T-\tau)} \mathbf{B} \tau d\tau$$

则式(5-72)可写为

$$\mathbf{x}(k+1) = \mathbf{F}(T)\mathbf{x}(k) + \mathbf{G}(T)\mathbf{u}(k) + \mathbf{G}_1(T)\dot{\mathbf{u}}(k) \quad (5-73)$$

这是采用一阶保持器时的离散化状态方程。

得到离散化模型之后,就可进行仿真。

#### 5.4.2 状态转移矩阵的近似计算

状态方程离散化所得到的模型中,  $\mathbf{F}(T)$ 、 $\mathbf{G}(T)$ 、 $\mathbf{G}_1(T)$  内均含有状态转移矩阵,可见要用离散化模型进行仿真,就必须先解决  $e^{At}$  的计算问题。所以,矩阵指数  $e^{At}$  的近似计算是状态方程离散化仿真的关键。

近似求解  $e^{At}$  的方法很多,有拉氏变换法、约当标准型法、拉格朗日-塞尔维斯特内插公式法、有限级数法、有限幂级数法等。本节只介绍最常用的一种,将  $e^{At}$  展为幂级数。

令  $t=T$ , 将  $e^{AT}$  展开

$$e^{AT} = I + AT + \frac{(AT)^2}{2!} + \dots + \frac{(AT)^n}{n!} + \dots = \sum_{i=0}^{\infty} \frac{(AT)^i}{i!} \quad (5-74)$$

直接求乘方很麻烦,而且费时,同时为了在计算机上求解方便,希望将式(5-74)变成可迭代的方式。

考虑到近似计算可以根据精度要求截取幂级数的前若干项,在这里取前  $L+1$  项,于是式(5-74)变为

$$\mathbf{F}(T) = e^{AT} \approx \sum_{i=0}^L \frac{(AT)^i}{i!} =$$



$$I + AT \left\{ I + \frac{AT}{2} \left\{ I + \frac{AT}{3} \left[ I + \dots + \frac{AT}{L-1} \left( I + \frac{AT}{L} \right) \dots \right] \right\} \right\} \quad (5-75)$$

式(5-75)在计算机上编程序求解应很方便。

在求解  $G(T)$  时,尽可能利用  $F(T)$  的计算结果。已知

$$G(T) = \int_0^T F(T-\tau) B d\tau = \int_0^T e^{A(T-\tau)} B d\tau$$

令

$$\lambda = T - \tau$$

于是有

$$G(T) = - \int_T^0 e^{A\lambda} B d\lambda = \int_0^T e^{A\lambda} B d\lambda$$

将  $e^{A\lambda}$  用式(5-75)代替,则

$$G(T) = \int_0^T \sum_{i=0}^L \frac{(A\lambda)^i}{i!} B d\lambda = \sum_{i=0}^L \int_0^T \frac{(A\lambda)^i}{i!} B d\lambda = \sum_{i=0}^L \frac{A^i T^{i+1}}{(i+1)!} B$$

为使  $G(T)$  中包含与  $F(T)$  有关的项,对上式进一步进行变换,得

$$G(T) = A^{-1} \left( \sum_{i=0}^L \frac{(AT)^{i+1}}{(i+1)!} \right) B = A^{-1} \left( \sum_{j=1}^{L+1} \frac{(AT)^j}{j!} \right) B$$

$$G(T) = A^{-1} (F(T) - I) B \quad (5-76)$$

式中的  $F(T)$  为  $e^{AT}$  取前  $L+2$  项的近似计算值。

可见利用式(5-75)和式(5-76)可将  $F(T)$  与  $G(T)$  在同一个程序中进行计算。

**例 5.4.1** 求积分环节  $G(s) = \frac{K}{s}$  的状态方程离散化模型。

首先写出连续系统的状态方程和输出方程

$$\dot{x} = Ku \quad (5-77)$$

$$y = x \quad (5-78)$$

可得  $A=0, B=k, C=1, D=0$ 。下面计算离散化模型的系统矩阵。

$$F(T) = e^{AT} = e^{0T} = 1$$

$$G(T) = \int_0^T F(T-\tau) B d\tau = \int_0^T e^{0(T-\tau)} K d\tau = KT$$

$$G_1(T) = \int_0^T F(T-\tau) B \tau d\tau = \int_0^T e^{0(T-\tau)} K \tau d\tau = \frac{1}{2} KT^2$$

所以,采用零阶保持器时,离散化模型为

$$x(k+1) = x(k) + KT u(k) \quad (5-79)$$

$$y(k) = x(k) \quad (5-80)$$

若采用一阶保持器,则离散化模型为

$$x(k+1) = x(k) + KT u(k) + \frac{1}{2} KT^2 \dot{u}(k) \quad (5-81)$$



$$y(k) = x(k) \quad (5-82)$$

式中

$$\dot{u}(k) = \frac{u(k) - u(k-1)}{T}$$

例 5.4.2 求比例加积分 (PI) 环节  $G(s) = K_p + \frac{K_I}{s}$  的状态方程离散化模型。

该连续系统的状态空间表达式为

$$\dot{x} = K_I u \quad (5-83)$$

$$y = x + K_p u \quad (5-84)$$

可得  $A=0, B=K_I, C=1, D=K_p$ 。

于是

$$\mathbf{F}(T) = e^{AT} = e^{0T} = 1$$

$$\mathbf{G}(T) = \int_0^T e^{0(T-\tau)} K_I d\tau = K_I T$$

$$\mathbf{G}_1(T) = \int_0^T e^{0(T-\tau)} K_I \tau d\tau = \frac{1}{2} K_I T^2$$

所以,在采用零阶保持器和一阶保持器时,系统的离散化模型分别为

$$x(k+1) = x(k) + K_I T u(k) \quad (5-85)$$

$$y(k) = x(k) + K_p u(k) \quad (5-86)$$

和

$$x(k+1) = x(k) + K_I T u(k) + \frac{1}{2} K_I T^2 \dot{u}(k) \quad (5-87)$$

$$y(k) = x(k) + K_p u(k) + K_p T^2 \dot{u}(k) \quad (5-88)$$

例 5.4.3 求惯性环节  $G(s) = \frac{K}{s+a}$  的状态方程离散化模型。

惯性环节的状态方程和输出方程分别为

$$\dot{x} = -ax + Ku \quad (5-89)$$

$$y = x \quad (5-90)$$

故  $A=-a, B=K, C=1, D=0$ , 则

$$\mathbf{F}(T) = e^{AT} = e^{-aT}$$

$$\mathbf{G}(T) = A^{-1}[\mathbf{F}(T) - 1]B = -\frac{K}{a}(e^{-aT} - 1) = \frac{K}{a}(1 - e^{-aT})$$

$$\mathbf{G}_1(T) = \int_0^T e^{A(T-\tau)} B \tau d\tau = \frac{K}{a^2}(e^{-aT} - 1) + \frac{K}{a}T$$

于是,采用零阶保持器和一阶保持器时离散化模型分别为

$$x(k+1) = e^{-aT} x(k) + \frac{K}{a}(1 - e^{-aT}) u(k) \quad (5-91)$$

$$y(k) = x(k) \quad (5-92)$$



和 
$$x(k+1) = e^{-aT}x(k) + \frac{K}{a}(1 - e^{-aT})u(k) + \left[ \frac{K}{a^2}(e^{-aT} - 1) + \frac{K}{a}T \right] \dot{u}(k) \quad (5-93)$$

$$y(k) = x(k) \quad (5-94)$$

其他一些类型环节的离散化模型在本节中不再一一列举,需要时可查阅有关资料。

下面举一个例子说明如何利用状态方程的离散化模型进行数字仿真。

例 5.4.4 一个控制系统如图 5-20 所示。

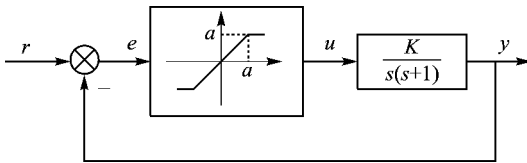


图 5-20 控制系统方框图

选择状态变量  $x_1 = K \int u dt = \frac{K}{s}u$

$$x_2 = y$$

则系统线性部分的状态空间表达式为

$$\dot{x} = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} x + \begin{pmatrix} K \\ 0 \end{pmatrix} u \quad (5-95)$$

$$y = (0 \quad 1)x \quad (5-96)$$

即  $A = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}, B = \begin{pmatrix} K \\ 0 \end{pmatrix}, C = (0 \quad 1), D = 0$ 。经计算,可得状态方程离散化模型的系统矩阵为

$$F(T) = e^{AT} = \begin{bmatrix} 1 & 0 \\ 1 - e^{-T} & e^{-T} \end{bmatrix}, \quad G(T) = \begin{bmatrix} KT \\ K(T - 1 + e^{-T}) \end{bmatrix}$$

于是离散化模型为

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 1 - e^{-T} & e^{-T} \end{pmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{pmatrix} KT \\ K(T - 1 + e^{-T}) \end{pmatrix} u(k) \quad (5-97)$$

$$y(k) = x_2(k) \quad (5-98)$$

利用式(5-97)和式(5-98)即可进行仿真计算,程序的框图如图 5-21 所示。

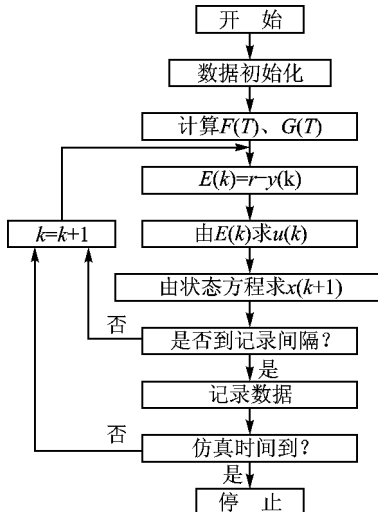


图 5-21 状态方程离散化仿真的程序框图



## 5.5 增广矩阵法

### 5.5.1 增广矩阵法的基本思想

从 5.4 节可知,利用状态方程离散化方法得到系统

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\end{aligned}$$

的离散化模型

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{F}(T)\mathbf{x}(k) + \mathbf{G}(T)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)\end{aligned}$$

该模型为一个差分方程组,利用该模型即可进行仿真。

此时仿真的误差主要受两个因素的影响。第一,在离散化过程中引入了虚拟的采样开关和保持器,这会引入输入信号失真,从而产生误差;第二, $\mathbf{F}(T)$ 、 $\mathbf{G}(T)$ 不易用解析方法求出,特别是对于高阶,需要采用数值方法进行计算,这就引入截断误差。

为了进一步提高仿真精度,考虑输入  $\mathbf{u} = 0$  的情况,即将状态方程变为齐次方程

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \quad (5-99)$$

此时与式(5-99)对应的离散化状态方程为

$$\mathbf{x}(k+1) = e^{\mathbf{A}T}\mathbf{x}(k) = \mathbf{F}(T)\mathbf{x}(k) \quad (5-100)$$

可见,对齐次状态方程(5-99)进行离散化时,不需要引入虚拟的采样开关和保持器。因此,产生误差的第一个主要因素就被消除了。同时式(5-100)中没有  $\mathbf{G}(T)\mathbf{u}(k)$  项,也消除了用数值方法计算  $\mathbf{G}(T)$  时产生的截断误差。这样,用式(5-100)仿真时,误差来源仅是用数值方法计算  $\mathbf{F}(T)$  时产生的截断误差和计算机产生的舍入误差,从而有可能使仿真精度得到较大提高。

正由于对齐次状态方程的仿真精度较高,所以希望通过将非齐次状态方程的输入变量“增广”为新的状态变量的方法,将原来的状态方程转变为增广形式的齐次状态方程

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} \quad (5-101)$$

$$\mathbf{y} = \tilde{\mathbf{C}}\tilde{\mathbf{x}} \quad (5-102)$$

以获得较高的仿真精度。这就是增广矩阵法的基本思想。

式(5-101)称为增广状态方程,式(5-102)称为增广输出方程。 $\tilde{\mathbf{x}}$  称为增广状态变量,它包括原有的状态变量和新增的状态变量。 $\tilde{\mathbf{A}}$  称为增广状态矩阵, $\tilde{\mathbf{C}}$  称为增广输出矩阵。

### 5.5.2 不同输入信号下的增广状态方程

下面分别讨论在不同的输入信号下如何选取增广状态变量,如何形成增广状态方程。设



原系统中,  $A$  为  $n \times n$  阶矩阵,  $B$  为  $n \times 1$  阶,  $C$  为  $1 \times n$  阶,  $D$  为  $1 \times 1$  阶,  $x$  为  $n$  维列向量。

### 1. 阶跃输入

$$u = U_0 \cdot 1(t)$$

系统原有  $n$  个状态变量, 定义第  $n+1$  个状态变量为系统输入

$$x_{n+1} = u(t) = U_0 \cdot 1(t) \quad (5-103)$$

则  $\dot{x}_{n+1} = 0$

于是增广状态方程和增广输出方程分别为

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} \quad (5-104)$$

$$\mathbf{y} = (\mathbf{C} \quad \mathbf{D}) \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} \quad (5-105)$$

初始条件为

$$\begin{pmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 \\ U_0 \end{pmatrix} \quad (5-106)$$

若令  $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix}$ ,  $\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{pmatrix}$ ,  $\tilde{\mathbf{C}} = (\mathbf{C} \quad \mathbf{D})$

即可写成式(5-101)和式(5-102)的形式。

### 2. 斜坡输入

$$u = U_0 t$$

定义新增的状态变量

$$x_{n+1} = u(t) = U_0 t$$

$$x_{n+2} = \dot{x}_{n+1} = \dot{u}(t) = U_0$$

则  $x_{n+1}(0) = 0$

$$x_{n+2}(0) = U_0$$

得到增广后的状态方程和输出方程

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \end{bmatrix} \quad (5-107)$$

$$\mathbf{y} = (\mathbf{C} \quad \mathbf{D} \quad 0) \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \end{bmatrix} \quad (5-108)$$

初始条件为



$$\begin{bmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \\ x_{n+2}(0) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ 0 \\ U_0 \end{bmatrix} \quad (5-109)$$

### 3. 加速度输入

$$u = \frac{1}{2}U_0 t^2$$

输入为加速度信号时,需要新增 3 个状态变量,令

$$x_{n+1} = u(t) = \frac{1}{2}U_0 t^2 \quad (5-110)$$

$$x_{n+2} = \dot{x}_{n+1} = U_0 t \quad (5-111)$$

$$x_{n+3} = \dot{x}_{n+2} = U_0 \quad (5-112)$$

而  
有

$$\dot{x}_{n+3} = 0$$

$$x_{n+1}(0) = 0$$

$$x_{n+2}(0) = 0$$

$$x_{n+3}(0) = U_0$$

可写出增广状态方程和增广输出方程

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \\ \dot{x}_{n+3} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{bmatrix} \quad (5-113)$$

$$\mathbf{y} = (\mathbf{C} \quad \mathbf{D} \quad 0 \quad 0) \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{bmatrix} \quad (5-114)$$

初始条件为

$$\begin{bmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \\ x_{n+2}(0) \\ x_{n+3}(0) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ 0 \\ 0 \\ U_0 \end{bmatrix} \quad (5-115)$$

将增广后的状态变量、状态矩阵、输出矩阵分别用  $\tilde{\mathbf{x}}$ 、 $\tilde{\mathbf{A}}$ 、 $\tilde{\mathbf{C}}$  表示,则式(5-113)和式(5-114)可写为

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \tilde{\mathbf{A}}\tilde{\mathbf{x}} \\ \mathbf{y} &= \tilde{\mathbf{C}}\tilde{\mathbf{x}} \end{aligned}$$



## 4. 指数输入

$$u = U_0 e^{-aT}$$

令第  $n+1$  个状态变量

$$x_{n+1} = u(t) = U_0 e^{-aT} \quad (5-116)$$

则

$$\dot{x}_{n+1} = -aU_0 e^{-aT} = -ax_{n+1} \quad (5-117)$$

$$x_{n+1}(0) = U_0$$

由此得原系统的增广状态方程和增广输出方程为

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & -a \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} \quad (5-118)$$

$$\mathbf{y} = (\mathbf{C} \quad \mathbf{D}) \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} \quad (5-119)$$

初始条件是

$$\begin{pmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 \\ U_0 \end{pmatrix} \quad (5-120)$$

令

$$\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix}$$

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & -a \end{pmatrix}, \quad \tilde{\mathbf{C}} = (\mathbf{C} \quad \mathbf{D})$$

则

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \tilde{\mathbf{A}}\tilde{\mathbf{x}} \\ \mathbf{y} &= \tilde{\mathbf{C}}\tilde{\mathbf{x}} \end{aligned}$$

## 5. 正弦输入

$$u = A \sin(\omega t + \varphi)$$

此时需要新增两个状态变量,即

$$x_{n+1} = u(t) = A \sin(\omega t + \varphi) \quad (5-121)$$

$$x_{n+2} = \dot{x}_{n+1} = A\omega \cos(\omega t + \varphi) \quad (5-122)$$

则有

$$\dot{x}_{n+2} = -A\omega^2 \sin(\omega t + \varphi) = -\omega^2 x_{n+1}$$

且

$$x_{n+1}(0) = A \sin \varphi$$

$$x_{n+2}(0) = A\omega \cos \varphi$$

根据式(5-121)、式(5-122)和系统的原状态方程,可写出正弦输入下系统的增广状态方程和增广输出方程

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & 0 \\ 0 & 0 & 1 \\ 0 & -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \end{bmatrix} \quad (5-123)$$



$$y = (C \quad D \quad 0) \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \end{bmatrix} \quad (5-124)$$

初始条件

$$\begin{bmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \\ x_{n+2}(0) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ A \sin \varphi \\ A \omega \cos \varphi \end{bmatrix} \quad (5-125)$$

下面举例说明增广矩阵法的具体应用。

**例 5.5.1** 写出输入  $u(t) = 0.8 \sin(0.1t) + 1$  时下述系统的增广状态方程和增广输出方程,并写出增广状态变量的初值。原系统为  $n$  阶,状态变量的初值为  $\mathbf{X}_0$ 。

$$\left. \begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}\mathbf{X} + \mathbf{D}u \end{aligned} \right\}$$

解:该系统的输入由正弦信号和阶跃信号的组成,定义第  $n+1$  个状态变量为系统输入

$$x_{n+1} = u(t) = 0.8 \sin(0.1t) + 1$$

可求出

$$x_{n+2} = \dot{x}_{n+1} = 0.08 \cos(0.1t)$$

$$x_{n+3} = \dot{x}_{n+2} = -0.008 \sin(0.1t)$$

$$\dot{x}_{n+3} = -0.0008 \cos(0.1t) = -0.01 x_{n+2}$$

即,可新增 3 个状态变量。于是可写出增广状态方程、增广输出方程及增广状态变量初值:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \\ \dot{x}_{n+3} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -0.01 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{bmatrix}, \quad \mathbf{y} = (\mathbf{C} \quad \mathbf{D} \quad 0 \quad 0) \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \\ x_{n+2}(0) \\ x_{n+3}(0) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ 1 \\ 0.08 \\ -0.008 \end{bmatrix}$$

**例 5.5.2** 设系统状态方程与例 5.5.1 相同,写出输入  $u = 2 + t^2$  时的增广状态方程、增广输出方程及增广状态变量的初值。

解:本例中系统的输入仍是由两个信号组成,分别定义第  $n+1$ 、 $n+2$  个状态变量为系统的输入,即

$$x_{n+1} = t^2, \quad x_{n+2} = 2$$

则可写出

$$x_{n+3} = \dot{x}_{n+1} = 2t, \quad \dot{x}_{n+3} = 2 = x_{n+2}$$

而

$$\dot{x}_{n+2} = 0$$

故,增广状态方程、增广输出方程及状态变量初值分别为

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \\ \dot{x}_{n+3} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{bmatrix}, \quad \mathbf{y} = (\mathbf{C} \quad \mathbf{D} \quad \mathbf{D} \quad 0) \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{x}(0) \\ x_{n+1}(0) \\ x_{n+2}(0) \\ x_{n+3}(0) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$



求出  $\dot{\tilde{x}} = \tilde{A}\tilde{x}$  及  $y = \tilde{C}\tilde{x}$ , 就可利用式(5-75)求出状态转移矩阵  $e^{\tilde{A}t}$ , 进而写出差分方程进行计算。

在增广矩阵法中, 由于增广状态变量的选取方式不是唯一的, 所以增广后系统的  $\tilde{A}$ 、 $\tilde{C}$  会有所不同。

增广矩阵法是一种精度较高的方法。若采用解析法计算矩阵指数  $e^{\tilde{A}T}$ , 那么这种方法就成为一种准确的方法, 不会因采样周期  $T$  的增大而增加计算误差; 相反, 因  $T$  增大使总的仿真步数减少, 会减少计算机舍入误差的积累。对于高阶系统, 一般只能用数值法计算  $e^{\tilde{A}T}$ , 这时会引入截断误差, 但已有一些有效算法可把误差控制在要求范围内。因此, 增广矩阵法允许采用较大的步长进行仿真。

$e^{\tilde{A}T}$  可以预先计算出来, 在递推过程中不需要反复计算, 因此每一步的计算量不大, 只有一些代数运算。

由以上分析可知, 增广矩阵法很适用于对连续系统进行快速数字仿真, 而且能得到较高的仿真精度。增广矩阵法也可用于刚性系统(stiff 系统)和多输入多输出(MIMO)系统的仿真。

但是增广矩阵法也有局限性, 它只适用于一些输入为典型函数的系统, 对于一般的非典型输入函数的系统或输入为表格函数的系统, 因为不容易或者不能选取到合适的增广状态变量, 得不到增广状态方程, 因而也就不能用增广矩阵法进行仿真。

## 5.6 面向结构图的数字仿真

在进行数字仿真时, 控制系统模型有时是传递函数, 有时是状态方程, 而有时是结构图。传递函数模型和状态方程模型已经很熟悉了, 本节介绍面向结构图的数字仿真方法。

图 5-22 和图 5-23 分别是两种较典型的结构图。

对传递函数模型, 仿真时可以采用数值积法或离散化方法。用数值积法仿真时, 需要将传递函数模型转化为微分方程描述的模型, 这种转化可以通过求系统的状态方程来实现。而对结构图模型, 可直接用一定方法求出其状态空间表达式, 然后再进行仿真。

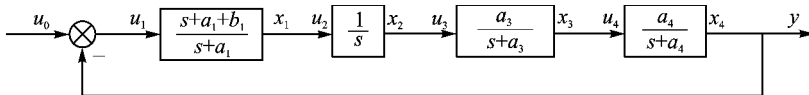


图 5-22 系统动态结构图

从结构图转化为状态空间表达式的方法中较常用的有连接矩阵法和仿真矩阵法两种, 本节只介绍连接矩阵法。

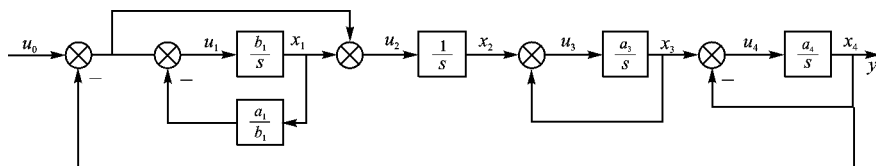


图 5-23 系统的模拟结构图

### 5.6.1 连接矩阵法

运用连接矩阵法可直接从系统的结构图得出其状态空间表达式,而不必计算出整个系统的闭环传递函数。如果系统中存在非线性环节,用连接矩阵法也比较容易处理。另外,用连接矩阵法不仅可以对单输入单输出系统进行仿真,而且可以很方便地对多输入多输出系统进行仿真。

下面通过例题说明如何实现从结构图向状态空间的转化。

**例 5.6.1** 对图 5-23 所示的 4 阶系统,求状态空间表达式。

若要求状态空间表达式,首先要选出状态变量,对于这个 4 阶系统,可令各环节的输出  $x_1, x_2, x_3, x_4$  为状态变量,即

$$\mathbf{X} = (x_1 \quad x_2 \quad x_3 \quad x_4)^T$$

对这 4 个环节分别写出其微分方程和输出方程。

第 1 个环节

$$\dot{x}_1 = b_1 u_1 \tag{5-126}$$

$$u_1 = u_0 - x_4 - \frac{a_1}{b_1} x_1 \tag{5-127}$$

第 2 个环节

$$\dot{x}_2 = u_2 \tag{5-128}$$

$$u_2 = u_0 + x_1 - x_4 \tag{5-129}$$

第 3 个环节

$$\dot{x}_3 = a_3 u_3 \tag{5-130}$$

$$u_3 = x_2 - x_3 \tag{5-131}$$

第 4 个环节

$$\dot{x}_4 = a_4 u_4 \tag{5-132}$$

$$u_4 = x_3 - x_4 \tag{5-133}$$

状态方程描述了系统状态变量的导数与状态变量本身及系统输入之间的关系。从式(5-126)、(5-128)、(5-130)和式(5-132)4 个微分方程可得



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & a_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (5-134)$$

由式(5-127)、(5-129)、(5-131)和式(5-133)4个输入方程可得

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} -\frac{a_1}{b_1} & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{u}_0 \quad (5-135)$$

令

$$\mathbf{K} = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & a_4 \end{bmatrix} \quad (5-136)$$

$$\mathbf{W}_1 = \begin{bmatrix} -\frac{a_1}{b_1} & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (5-137)$$

$$\mathbf{W}_0 = (1 \ 1 \ 0 \ 0)^T \quad (5-138)$$

$$\mathbf{U} = (u_1 \ u_2 \ u_3 \ u_4)^T$$

则有  $\dot{\mathbf{x}} = \mathbf{K}\mathbf{U}$  (5-139)

$$\mathbf{U} = \mathbf{W}_1 \mathbf{x} + \mathbf{W}_0 \mathbf{u}_0 \quad (5-140)$$

将方程式(5-140)代入式(5-139),可得系统的状态方程

$$\dot{\mathbf{x}} = \mathbf{K}\mathbf{W}_1 \mathbf{x} + \mathbf{K}\mathbf{W}_0 \mathbf{u}_0 \quad (5-141)$$

$\mathbf{K}$ 称为系数阵,是对角矩阵,其阶数与系统的状态变量个数相同; $\mathbf{W}_1$ 称为系统的状态连接矩阵; $\mathbf{W}_0$ 称为系统的输入连接矩阵,它与外输入对系统各环节的作用有关。

若令

$$\mathbf{A} = \mathbf{K}\mathbf{W}_1$$

$$\mathbf{B} = \mathbf{K}\mathbf{W}_0$$

则式(5-141)就可变为标准的状态方程

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}_0 \quad (5-142)$$

$$\dot{\mathbf{x}} = \begin{bmatrix} -a_1 & 0 & 0 & -b_1 \\ 1 & 0 & 0 & -1 \\ 0 & a_3 & -a_3 & 0 \\ 0 & 0 & a_4 & -a_4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} b_1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{u}_0$$



求出状态方程后再来求输出方程。

输出方程描述了系统输出与状态变量及外输入之间的关系,由图 5-23 知

$$\mathbf{y} = x_4$$

所以输出方程为

$$\mathbf{y} = (0 \ 0 \ 0 \ 1) \mathbf{x}$$

令

$$\mathbf{C} = (0 \ 0 \ 0 \ 1)$$

则有

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (5-143)$$

这样,利用连接矩阵法就把系统的结构图模型转换为状态方程模型。从状态方程出发,用前面所讲的方法对系统进行离散化,或采用增广矩阵法,就可得出用于仿真的差分方程。

从所得的状态方程(5-142)和输出方程(5-143)可以看出,各环节的参数在方程中都是独立的,即某一个参数的改变不会引起其他参数的变化。因此,用这种方法研究某个参数的变化对系统性能的影响是很方便的。

### 5.6.2 例题

例 5.6.2 系统结构图如图 5-24 所示,求其状态方程和输出方程。

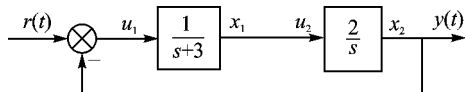


图 5-24 系统结构图

第一步,令两个环节的输出  $x_1$ 、 $x_2$  为状态变量,即

$$\mathbf{X} = (x_1 \ x_2)^T$$

第二步,写出两个环节的微分方程和输出方程。

第 1 个环节

$$\dot{x}_1 = u_1 - 3x_1, \quad u_1 = r - x_2$$

第 2 个环节

$$\dot{x}_2 = 2u_2, \quad u_2 = x_1$$

第 3 步,将上述关系写成矩阵形式。

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5-144)$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r \quad (5-145)$$

第 4 步,将式(5-145)代入式(5-144),整理,即可得状态方程

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -3 & -1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r$$

输出方程可直接写出

$$\mathbf{y} = (0 \ 1)\mathbf{X}$$



例 5.6.3 已知多输入多输出系统的结构图如图 5-25 所示,写出其状态方程。

从图 5-25 可知,这是一个比较复杂的高阶系统,若是先求传递函数再求状态方程表达式,计算过程将会很麻烦,而采用连接矩阵法就简单多了。

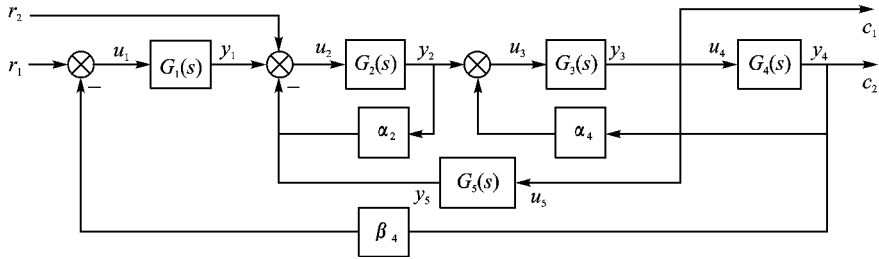


图 5-25 多输入多输出系统的结构图

图 5-25 中

$$G_i(s) = \frac{c_i + d_i s}{a_i + b_i s}, \quad i = 1, 2, 3, 4, 5$$

首先写出 5 个环节的输入方程

$$\begin{aligned} u_1 &= r_1 - \beta_4 y_4 \\ u_2 &= r_2 + y_1 - \alpha_2 y_2 - y_5 \\ u_3 &= y_2 - \alpha_4 y_4 \\ u_4 &= y_3 \\ u_5 &= y_3 \end{aligned}$$

写成矩阵形式即为

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -\beta_4 & 0 \\ 1 & -\alpha_2 & 0 & 0 & -1 \\ 0 & 1 & 0 & -\alpha_4 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

令

$$\begin{aligned} \mathbf{U} &= (u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5)^T \\ \mathbf{Y} &= (y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5)^T \\ \mathbf{R} &= (r_1 \quad r_2)^T \end{aligned}$$



$$\mathbf{W}_1 = \begin{bmatrix} 0 & 0 & 0 & -\beta_4 & 0 \\ 1 & -\alpha_2 & 0 & 0 & -1 \\ 0 & 1 & 0 & -\alpha_4 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5-146)$$

$$\mathbf{W}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5-147)$$

则有

$$\mathbf{U} = \mathbf{W}_1 \mathbf{Y} + \mathbf{W}_0 \mathbf{R} \quad (5-148)$$

接下来要形成状态方程。由

$$\frac{y_i(s)}{u_i(s)} = G_i(s) = \frac{c_i + d_i s}{a_i + b_i s}$$

可得

$$(a_i + b_i s)y_i(s) = (c_i + d_i s)u_i(s)$$

若令

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 & 0 \\ 0 & 0 & a_3 & 0 & 0 \\ 0 & 0 & 0 & a_4 & 0 \\ 0 & 0 & 0 & 0 & a_5 \end{bmatrix} \quad (5-149)$$

$$\mathbf{B} = \begin{bmatrix} b_1 & 0 & 0 & 0 & 0 \\ 0 & b_2 & 0 & 0 & 0 \\ 0 & 0 & b_3 & 0 & 0 \\ 0 & 0 & 0 & b_4 & 0 \\ 0 & 0 & 0 & 0 & b_5 \end{bmatrix} \quad (5-150)$$

$$\mathbf{C} = \begin{bmatrix} c_1 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 \\ 0 & 0 & c_3 & 0 & 0 \\ 0 & 0 & 0 & c_4 & 0 \\ 0 & 0 & 0 & 0 & c_5 \end{bmatrix} \quad (5-151)$$



$$D = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 \\ 0 & 0 & d_3 & 0 & 0 \\ 0 & 0 & 0 & d_4 & 0 \\ 0 & 0 & 0 & 0 & d_5 \end{bmatrix} \quad (5-152)$$

则有矩阵形式

$$(A + Bs)Y = (C + Ds)U \quad (5-153)$$

成立,其中的 $Y$ 、 $U$ 形式同前边。

将式(5-148)代入式(5-153)可得

$$(A + Bs)Y = (C + Ds)(W_1Y + W_0R)$$

拉氏反变换后得

$$(B - DW_1)\dot{Y} = (CW_1 - A)Y + CW_0R + DW_0\dot{R}$$

为了避免输入信号为阶跃时,初始点的导数为无穷大,要求 $DW_0=0$ ,则上式变为

$$(B - DW_1)\dot{Y} = (CW_1 - A)Y + CW_0R \quad (5-154)$$

若 $(B - DW_1)$ 的逆阵存在,则有

$$\dot{Y} = (B - DW_1)^{-1}(CW_1 - A)Y + (B - DW_1)^{-1}CW_0R \quad (5-155)$$

令

$$\begin{aligned} \tilde{A} &= (B - DW_1)^{-1}(CW_1 - A) \\ \tilde{B} &= (B - DW_1)^{-1}CW_0 \end{aligned}$$

式(5-155)可写为

$$\dot{Y} = \tilde{A}Y + \tilde{B}R \quad (5-156)$$

式(5-156)即是图5-25所示系统的状态方程。

本章介绍了连续系统离散化仿真的几种方法。其中,运用替换法和根匹配法可以从连续系统模型直接推导出离散化模型;而离散相似法和状态方程离散法则是通过在连续系统中加入虚拟采样开关和保持器将系统离散化;增广矩阵法适用于对输入为典型函数的系统进行仿真;若控制系统是以结构图形式给出的,则可运用面向结构图的方法进行仿真。与数值积分法相比,离散化方法允许采用较大计算步距,不仅可用于系统仿真,而且可用于数字控制器的实现。

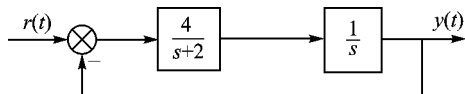
## 习 题

- 5.1 与数值积分法仿真相比,离散化仿真有哪些特点?
- 5.2 双线性变换是否改变系统的稳定性?请证明。



5.3 若连续系统的传递函数为  $G(s) = \frac{1}{s+0.5}$ , 用根匹配法求其离散化模型。采样周期取  $0.5s$ , 附加零点分别配在原点、 $-1$  处和  $(-1, 0)$  区间上。

5.4 已知连续系统如题图 5.4 所示。用离散相似法求其离散化模型, 并写出采样周期  $T=0.1s$  时, 仿真所用的差分方程。



题图 5.4

5.5 简述影响离散相似法的精度和稳定性的因素。

5.6 已知系统的状态方程为

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + ku\end{aligned}$$

求离散化的状态方程(分别采用零阶保持器和一阶保持器)。

5.7 简述增广矩阵法的基本思想和优点。

### 参考文献

- [1] 熊光楞编著. 控制系统数字仿真. 北京:清华大学出版社,1982.
- [2] 熊光楞,肖田元,张燕云编著. 连续系统仿真与离散事件系统仿真. 北京:清华大学出版社,1991.
- [3] 翁思义编著. 自动控制系统计算机仿真与辅助设计. 西安:西安交通大学出版社,1987.
- [4] 李友善主编. 自动控制原理(下). 北京:国防工业出版社,1987.
- [5] 郭云芳,欧阳玲,刘继美编著. 计算机控制技术. 北京:北京航空航天大学出版社,1991.
- [6] 韩慧君编. 系统仿真. 北京:国防工业出版社,1985.
- [7] 汪瑞芳,陈东义编著. 连续系统仿真及应用技术. 重庆:重庆大学出版社,1991.
- [8] 涂健主编. 控制系统的数字仿真与计算机辅助设计. 武汉:华中工学院出版社,1985.
- [9] 康凤举主编. 现代仿真技术与应用. 北京:国防工业出版社,2001.

# 第 2 章 随机控制系统仿真

随机系统就是外部输入过程表现出随机性,或者内部活动表现出随机性的系统。在自然界以及人类的生产、生活中,“随机”问题到处可见。例如,天气预报就是随机问题的一个例子。一个地区的天气状况是个随机量,受到本地区和邻边地区各种因素,包括各种随机因素的影响,只有全面综合这些因素的作用,才能准确地作出天气预报。这属于随机系统中的预报、预测问题。

控制系统分为两大类:确定性控制系统和不确定性控制系统。确定性控制系统是指系统参数、外输入(包括指令和干扰)和输出都是已知的确定的参数。在控制系统研究中,确定性系统是最常见的一种,其建模方法、系统性能评价方法、分析方法等都为大家所熟知。不确定性控制系统则相反,它的系统参数、外输入等都是随机的(事先不知道)。随机控制系统属于不确定性控制系统,是服从概率统计的不确定性控制系统。有的不确定性系统不一定是随机系统。例如,模糊控制系统是一个不服从概率统计的不确定性控制系统。本章主要介绍随机控制系统的仿真方法。

## 2.1 随机控制系统概述

由于随机控制系统的被测量和被控量都是随机过程,因此,随机控制理论的发展与随机过程理论的发展密切相关。随机过程理论产生于 19 世纪初期,是为适应物理学、生物学、通信与控制、管理科学等方面的需要而逐步发展起来的。最初在布朗运动、电话信息量和电子管的散粒效应噪声等问题的研究中取得了成果。1906 年,科尔莫戈罗夫(苏联数学家)奠定了随机过程的数学理论。以后,有关随机过程理论和应用的著作大批出现,并取得丰硕成果。1931 年,伊藤(日本)发表了《论随机微分方程》一文。随后,对随机微分方程的研究受到了广泛重视,并渗透到很多领域。随机过程的研究与发展,为随机控制的发展提供了理论基础。几十年来,随机控制理论在很多领域已有广泛和成功的应用。例如,在某些工业批量加工过程中,在阿波罗飞船导航及控制系统中使用了最小方差控制。随着社会经济活动的更加丰富,生产和科学技术的深入发展,随机控制理论已渗透到各个领域。

### 2.1.1 随机控制系统模型

相对于确定性控制系统来说,随机控制系统就是其扰动可以用随机过程明确表示的受控动力学系统。随机控制系统的研究就是分析其动力学系统和系统变量的统计特性,研究如何



实现参数最优化,并对随机控制系统给定一个性能指标,寻求使性能指标最小的最优控制律。

可以用如下线性时变方程代表随机控制系统模型:

$$\left. \begin{aligned} \dot{x} &= Ax + Bw + Gv \\ y &= Cx + Dv \end{aligned} \right\} \quad (远-员)$$

式中, $x$ 为系统状态变量, $v$ 为随机初值;

$y$ 为系统输出;

$w$ 为外扰动,随机变量;

$A, B, C, D$ 为系统名义参数,是确定量;

$\Delta A, \Delta B, \Delta C, \Delta D$ 为系统参数随机误差。

由于 $\Delta A, \Delta B, \Delta C, \Delta D$ 是不确定的,所以式(远-员)的解是不确定的。如果 $\Delta A = \Delta B = \Delta C = \Delta D = 0$ ,并且 $w$ 和 $v$ 都是已知的,则式(远-员)是确定性控制系统模型。

下面举一个随机控制系统例子。导弹末制导系统是一个典型的随机控制系统。通常,商家出售的导弹武器通过商业标牌注明型号、使用条件、命中概率等,但从来不保证某发导弹能否击中目标。因为同是一种型号的导弹,参数各有差异(外形、结构、动力、制导系统等),飞行条件不同(风、高度、目标特性等),造成每发导弹的飞行情况不同,因而某一发导弹的使用效果是不确定的,无法对某发导弹能否击中目标作出明确回答。标牌上的命中概率是导弹的统计特性。这种统计特性可以通过导弹打靶、系统分析、随机制导系统仿真等获取。对于这个实例,式(远-员)中的 $y$ 为视线距(导弹和目标间的距离, $w$ 为脱靶量, $v$ 是 $y$ 改变符号的时刻)。

对于随机控制系统的仿真,主要是仿真系统的动态过程,研究系统随机变量的统计特性。但随机控制系统有其特殊的问题需要去解决,往往是这些特殊问题使随机控制系统仿真变得复杂。例如,随机干扰的建模就是难度很大的问题,随机信号的测量、数据处理、数学模型的抽象等;再如,具有一定统计数值特征的系统参数误差、状态变量初值的产生以及具有一定功率谱密度的随机干扰过程的产生等。由于这些特殊问题的存在,对随机控制系统仿真的方法也不同于定常系统。随机控制系统仿真常用的方法有两种,即蒙特卡罗(配噪噪法)法和伴随系统(蒙特卡罗法)。下面先分析随机控制系统仿真中遇到的专门问题,再结合具体的实例,介绍随机控制系统仿真方法。

### 远-员 随机控制系统仿真的专门问题

随机控制系统仿真面临着以下专门问题:

- (员) 随机数的产生;
- (圆) 随机数概率分布的转换;
- (猿) 随机数功率谱密度的转换;
- (源) 系统随机参数误差的产生;



(缘) 系统随机初值的产生;

(远) 系统随机干扰的产生。

下面对这些问题逐项予以介绍。

### 员 随机数产生

对随机控制系统仿真来说,如何产生随机信号是关键问题。有了随机信号,就可以把它变成具有一定概率分布、功率谱密度的各种随机因素附加在随机控制系统上,进行系统仿真。仿真技术发展早期,随机信号用噪声信号发生器(一种仪器)产生。利用这种方法,随机信号的重复性低、使用不方便,进入数字仿真时期已弃之不用。现在使用数字计算机产生随机数字序列成了产生随机信号的主要方法。

#### (员) 产生随机数的计算机方法

二进制计算机可以给出数值为 园,员,圆,..., 厶- 厶 的 厶 个整数(厶为计算机字长),采用递推公式

$$曾_{n+1} = 越枣曾_n \dots 曾_1)$$

对这些自然数进行处理,使 厶 个数出现的顺序随机,这个随机数序列便是计算机产生的具有一定统计特性的随机数。

目前应用最广泛的随机数发生器之一是线性同余发生器。它是 蕴霖 在 员缘 年提出的。线性同余发生器产生随机数的递推公式如下:

$$曾_{n+1} = 越(\lambda 曾_n \text{ 悦}(\mu)) \quad (远- 圆)$$

通过选择合适的  $\lambda$ 、悦  $\mu$ ,可以保证产生的随机数  $曾_n$  位于  $[园, \mu \text{ 厶}]$  区间内,而且可以保证均匀性。在式(远- 圆)中,当参数 悦  $\mu$  时称为混合同余法,当参数 悦  $\mu$  时称为乘同余法。

乘同余法早于混合同余法的出现,因而使用比较广泛。为了提高乘同余法的可用性,人们进行了大量的研究,发现如下规律,即  $\lambda$  和  $\mu$  的选取要符合以下原则: $\lambda \approx 厶$  且满足  $\lambda$  越 愿  $\mu$  (厶 为字长,晕为任意整数),  $\mu$  越 厶 且满足不等式  $园 < \mu$  约 厶 (灶为任意整数)。

由式(远- 圆)得到的  $曾_n$  满足:

$$园 \leq 曾_n \leq \mu \text{ 厶}$$

为了得到(园, 厶)区间上所需要的随机数  $怎_n$ ,可令

$$怎_n = 越曾_n \text{ 辕}$$

#### 摇摇(圆) 计算机产生随机数的特点

由计算机按递推公式产生的随机数有以下特点:

第一,数字序列是伪随机的。这是因为计算机的字长一定,它能表示的数有限。无论采用哪一种递推公式,数字序列达到一定长度后便会周期性地重复或蜕化为零。应特别指出的是,递推公式的初值(俗称种子)固定时,数字序列的顺序成为确定的,失去了随机数的性质。随机控制系统仿真时应特别注意这种情况的发生。计算机产生的随机数序列的周期愈长,它的随机性愈强。为了增强随机数序列的随机性,应尽量增大数字序列的周期。对于“乘同余法”, 员圆



数字序列的周期为  $\mu$  乘, 因而  $\mu$  尽可能取大值。

第二, 数字序列具有均匀概率分布的统计特性。用“乘同余法”产生的随机数的概率密度为

$$f(x) = \begin{cases} \frac{1}{\mu} & 0 \leq x < \mu \\ 0 & \text{其他} \end{cases} \quad (6-10)$$

它的均值为

$$\bar{x} = \int_0^{\mu} x f(x) dx = \int_0^{\mu} x \frac{1}{\mu} dx = \frac{\mu}{2} \quad (6-11)$$

其方差为

$$\sigma^2 = \int_0^{\mu} (x - \bar{x})^2 f(x) dx = \int_0^{\mu} (x - \frac{\mu}{2})^2 \frac{1}{\mu} dx = \frac{\mu^2}{12} \quad (6-12)$$

它的时间统计特性, 即自相关函数和功率谱密度分别为

$$R(\tau) = \begin{cases} \mu & |\tau| < T \\ \frac{\mu}{T} (\mu - |\tau|) & T \leq \tau \leq 2T \\ 0 & \text{其他} \end{cases} \quad (6-13)$$

$$S(\omega) = \frac{\mu}{T} \left[ \frac{\sin(\omega T/2)}{\omega/2} \right]^2 \quad (6-14)$$

式中,  $T$  为数字产生周期, 在一个周期内  $x$  保持不变,  $R(0) = \mu$ ,  $S(0) = \frac{\mu^2}{12}$ 。当  $\omega \approx \frac{2\pi}{T}$  时, 可以看出  $S(\omega)$  近似为白色随机过程。该白色随机过程的带宽与  $T$  有关,  $T$  愈小, 带宽愈大。

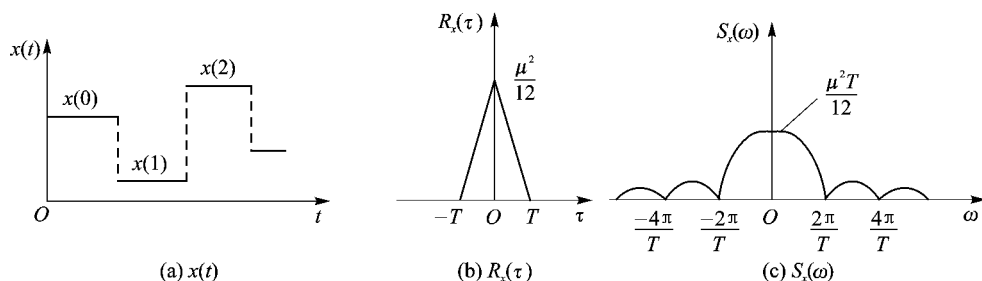


图 6-10 计算机生成的数字序列及其相关函数、谱密度函数

### 产生随机数的要求

用计算机产生随机数的主要要求为: 第一, 随机数有足够长的周期。例如随机数用来仿真导弹打靶过程中的阵风干扰, 随机数的周期应保证导弹飞行过程中随机数不出现重复现象; 第二, 产生随机数的递推公式计算量小, 占用内存少; 第三, 随机数的统计特性满足使用要求。计算机生成随机数是和控制系统的仿真同步进行的, 仿真积分步长(或线性离散系统仿真中的采样周期)与随机数产生周期之间有一定关系(相等或成整数倍)。因而仿真步长影响随机数的



统计特性。

### 均匀随机数概率分布的转换

#### (员) 计算机生成随机数的统计特性

计算机生成的随机数序列的值域为(园, 原) 标称化后变为(园, 员)。当考虑的频带范围满足不等式  $\omega \approx \frac{1}{n}$  时, 该随机数序列近似为白色随机过程。所以计算机生成的随机数序列可视为白色随机数, 它的概率密度函数  $f(u)$  曾在区间(园, 员)上均匀分布。标称化后的随机数的统计特性: 概率密度  $f(u)$ 、概率分布  $F(u)$ 、均值  $\bar{u}$ 、方差  $\sigma_u^2$  分别为

$$f(u) \text{ 越 } \begin{cases} 1 & 0 \leq u \leq 1 \\ 0 & \text{曾约园或 曾跃员} \end{cases} \quad (\text{远- 愿})$$

$$F(u) \text{ 越 } \begin{cases} u & \text{曾约园} \\ 1 & 0 \leq u \leq 1 \\ 1 & \text{曾跃员} \end{cases} \quad (\text{远- 怨})$$

$$\bar{u} \text{ 越 } \frac{1}{2} \quad (\text{远- 员园})$$

$$\sigma_u^2 \text{ 越 } \frac{1}{12} \quad (\text{远- 员员})$$

概率密度为均匀分布的随机数是随机控制系统中经常用到的一种随机数。例如导弹打靶仿真, 目标机动开始时间就处理为在整个导弹飞行中是等概率的。控制系统仿真中, 正态分布的随机数是常用的随机数的另一种。例如系统随机参数误差(式(远- 员)中的  $\Delta$  与  $\Delta_0$ )和随机初值(式(远- 员)中的  $u_0$ )多是正态分布的, 需要用瑞利分布的随机数来模拟。控制系统仿真, 需要将计算机产生的均匀分布随机数转换成其他分布类型的随机数。

#### 摇摇(圆) 均匀分布转换为正态分布

假如 控制系统仿真需要均值为零、方差为 员的正态分布随机数, 该随机数可以用计算机产生的均匀分布随机数转换而成。具体做法如下:

第 员步, 产生取值区间为(园, 原)的随机数;

第 圆步, 标称化, 改变随机数的取值区间为(园, 员);

第 猿步, 均值移位, 取值区间改变为(原, 原), 该随机数的均值和方差分别为  $\bar{u} \text{ 越 } 0$  和  $\sigma_u^2 \text{ 越 } \frac{1}{12}$ ;

第 源步, 根据极限定理, 求 灶个不相关的随机数  $u_1, u_2, \dots, u_n$  之和  $u$  即

$$u \text{ 越 } \sum_{i=1}^n u_i \quad (\text{远- 员圆})$$

当 灶足够大时(一般认为 灶跃园已足够大),  $u$  为正态分布随机数。

#### 摇摇(猿) 均匀分布转换为三角形分布

图 远- 圆中的  $u$  是取值域为(原, 员), 概率密度函数  $f(u)$  为三角形的随机数。它可以用两个均匀分布、取值域分别为(原, 园)及(园, 员)的随机数  $u_1$  和  $u_2$  相加产生。

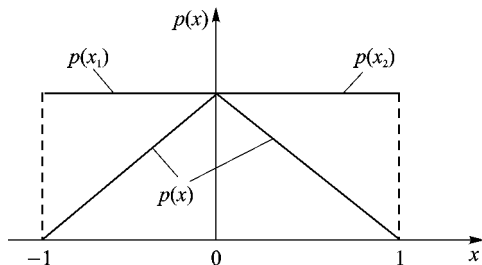
员源



由均匀分布随机数改变为一般分布的随机数,可用直接变换法。

### 随机数功率谱密度的转换

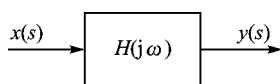
随机数的统计特性反映在两个方面:一是数值分布特性,二是与时间有关的频率特性。控制系统仿真用到的随机因素既要具有希望的概率分布特性,又要具有一定的功率谱密度。例如导弹的打靶仿真,作用在导弹上的阵风干扰,除阵风的风速满足一定概率分布要求外,还必须满足它的频率特性的要求。而计算机产生的随机数序列,在有限带宽内为白色随机数。需要改变它的功率谱密度才能满足仿真对它的功率谱密度的要求。



图远- 圆均匀分布改为三角形分布

将功率谱密度为  $\Phi_x(\omega)$  的白色随机数 曾改变为功率谱密度为  $\Phi_y(\omega)$  的随机数 赠的方法是:将该白色随机数输入一个传递函数为  $H(\omega)$  的线性系统,并且使线性系统的传递函数满足以下等式:

$$\Phi_y(\omega) = |H(\omega)|^2 \Phi_x(\omega) \quad (远- 猿)$$



图远- 猿成形滤波器

线性系统的输出 赠更是要求的随机数,如图远- 猿所示。该线性系统称为成形滤波器。因为根据线性系统理论可知,线性系统不改变平稳随机过程的性质。当输入 曾赋为白色随机过程时,  $\Phi_x(\omega)$  越 圆为常值,输出的功率谱密度的幅值与  $\sigma^2$  成正比,频率特性完全取决于线性系统的传递函数  $H(\omega)$ ,而且有以下关系:

$$\Phi_y(\omega) = |H(\omega)|^2 \Phi_x(\omega) \quad (远- 猿)$$

获取成形滤波器  $H(\omega)$  的表达式,需知道成形滤波器输出随机数的功率谱密度  $\Phi_y(\omega)$ 。假定  $\Phi_x(\omega)$  和  $H(\omega)$  都用有理分式表示,即

$$\Phi_x(\omega) = \frac{P(\omega^2)}{Q(\omega^2)} \quad (远- 猿)$$

$$H(\omega) = \frac{N(\omega)}{D(\omega)} \quad (远- 猿)$$

已知  $P(\omega)$ 、 $Q(\omega)$  可求出  $N(\omega)$ 、 $D(\omega)$ ,成形滤波器的传递函数则是  $H(\omega) = \frac{N(\omega)}{D(\omega)}$ 。

### 系统随机参数误差的产生

式(远- 猿)给出了随机控制系统模型。其中  $N(\omega)$ 、 $D(\omega)$  是系统名义参数,葬和 遭是系统误差造成的随机变量。不同的系统,葬和 遭的值不同,就像导弹制导系统参数,每发导弹都有误差



差异。随机系统仿真,在运行仿真系统之前需随机地确定  $\mu$  和  $\sigma$  的值并附加到相应的系统名义参数上,即  $\mu = \mu_0 + \Delta\mu$ ,  $\sigma = \sigma_0 + \Delta\sigma$

(丙) 系统随机参数误差的概率分布

一般来说,因生产误差造成的随机参数误差为正态分布,它的概率密度函数如图 2-10 所示。图中  $\Delta\mu$  是系统参数  $\mu$  的误差,系统的实际参数为  $\mu_0 + \Delta\mu$ ,  $\sigma$  为  $\Delta\mu$  的方差根。

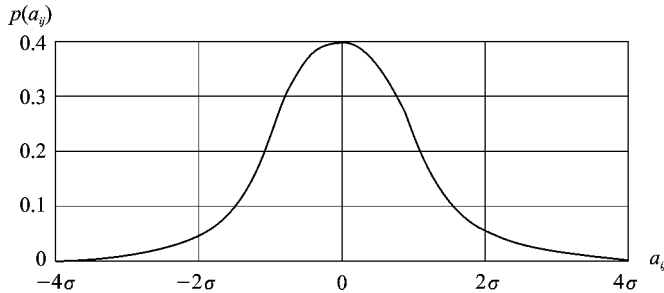


图 2-10 参数误差的概率密度

(丁) 方差根  $\sigma$  的确定

生产误差习惯以最大允许误差的方式给定。例如以  $\Delta\mu$  表示参数  $\mu$  的最大允许误差,意味着所有参数误差  $\Delta\mu$  应满足不等式  $|\Delta\mu| \leq \Delta\mu_{max}$ 。考虑到  $\Delta\mu$  的正态概率分布特性,得出

$$\Delta\mu \leq \Delta\mu_{max}$$

即  $\sigma \leq \Delta\mu_{max} / 3$  (2-10)

$\Delta\mu$  的方差  $\sigma_{\Delta\mu}^2$  为

$$\sigma_{\Delta\mu}^2 = (\Delta\mu_{max} / 3)^2 \quad (2-11)$$

(戊) 系统随机参数误差的产生

用计算机产生值域为  $(-\Delta\mu_{max}, \Delta\mu_{max})$  的随机数序列  $\mu_i$ ,  $\mu_i$  的统计特性如前述为  $\mu_i \sim N(0, \sigma_{\Delta\mu}^2)$ 。求  $n$  个不相关的  $\mu_i$  的和数  $\mu$  它的统计特性已经知道为  $\mu \sim N(0, \sigma_{\Delta\mu}^2/n)$  而且  $\mu$  的分布特性为正态分布。令  $\sigma_{\Delta\mu}^2/n = \sigma_{\mu}^2$ , 则得

$$\mu \sim N(0, \sigma_{\mu}^2) \quad (2-12)$$

由式(2-12)可得计算机产生的随机数  $\mu_i$  的取值域界限  $\Delta\mu$  即

$$\Delta\mu = \mu_i / \sigma_{\mu} \quad (2-13)$$

有了  $\Delta\mu$  即可以把计算机产生的  $\mu_i$  (即  $\mu_i$ ) 的整数标称化为  $\mu_i / \Delta\mu$  的除数  $\mu$ , 即

$$\mu = (\mu_i / \Delta\mu) \Delta\mu \quad (2-14)$$

运为计算机字长。

系统随机初值的产生

在系统实际运行中,式(2-1)中的系统状态变量初值  $x(0)$  是不确定的。 $x(0)$  由名义值  $x_0$



和初值误差  $\Delta x_0$  两部分组成,即  $x_0 = x_{0n} + \Delta x_0$ 。例如肩扛式地对空导弹的初始发射角,其名义值是根据目标的飞行速度、距离、方位等确定的。严格按照理想初始发射角发射的导弹具有较高的命中概率。实战中战士用瞄准具标定发射角存在操作误差,这种操作误差是随机的。随机发射角误差影响导弹命中概率。随机控制系统仿真,需要考虑系统初始状态随机误差的仿真。

### (员) 系统随机初值的概率分布

一般认为,系统随机初值误差  $\Delta x_0$  符合正态概率分布,概率密度函数如图 远-缘所示。图中  $\Delta x_0$  是系统第  $n$  个状态变量初值  $x_{0n}$  的误差,系统实际初值为  $x_{0n} + \Delta x_0$ 。图中  $\sigma$  为  $\Delta x_0$  的方差根。

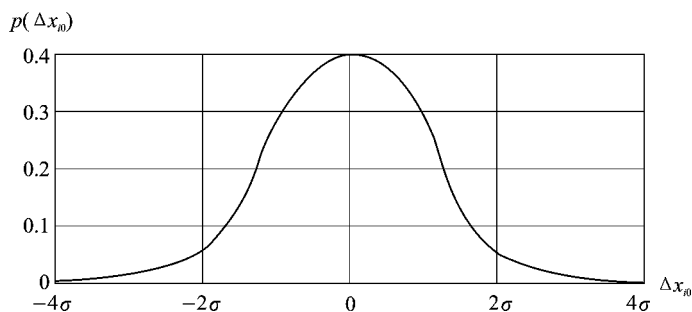


图 远-缘 随机初值的概率密度

### (圆) 随机初值方差根 $\sigma$ 的确定

工程上对系统状态变量初值误差  $\Delta x_0$ ,常以最大值的方式给定,这个最大值多数来源于系统运行统计数据。例如以  $\Delta x_{max}$  表示参数  $\Delta x_0$  的最大值,考虑到  $\Delta x_0$  的正态概率分布特性,得出  $\Delta x_{max} \leq 3\sigma$ ,因而得

$$\sigma \leq \frac{\Delta x_{max}}{3} \quad (远-圆)$$

### (猿) 随机初值的产生

用产生系统随机参数误差相同的办法,首先用计算机产生值域为  $(-\sigma, \sigma)$  的随机数序列  $\Delta x_{0i}$ ,具有均值为零,方差  $\sigma^2$  的统计特性。而后,将  $n$  个相互独立的随机数相加得到和数  $\sum_{i=1}^n \Delta x_{0i}$  的统计特性为  $\Delta x_0$ ,而且为正态分布。同理,得到计算机生产的随机数  $\Delta x_0$  的取值域界限  $[-\sigma, \sigma]$  即

$$-\sigma \leq \Delta x_0 \leq \sigma \quad (远-圆)$$

### 远 系统随机干扰的产生

#### 远(员) 系统随机干扰仿真的任务

系统随机干扰是仿真系统运行期间作用在系统上的随机过程,系统随机干扰仿真除数值统计特性相似外,还要保证频率统计特性的相似。系统随机干扰的数值和频率统计特性与在



一定带宽内具有白色随机统计特性的二进制数序列不同,系统随机干扰仿真的任务是:如何将数值和频率统计特性已知的白色随机数序列转变成与随机干扰的统计特性相同的随机数序列。

### 摇摇(圆) 系统随机干扰模型的建立

系统随机干扰模型包括两种数据,一是功率谱密度函数(或简称频谱,与相关函数构成傅立叶变换时,两者知其一即可);二是方差,随机干扰的频谱和方差是通过对于干扰的实地测量并经数据处理得到的。例如大气中飞行的飞行器受到随机阵风干扰,阵风的频谱、方差随地理位置、高度的变化而变化。如果想得到阵风的频谱和方差,需要实地测出风速、风向随地理位置、高度的变化,而后对测量的数据进行离散量的傅立叶变换并进而抽象为数学模型。著名的阅读大气紊流模型就是这样得来的。阅读模型风速的时间频谱(相对空间频谱而言)在猿个坐标轴 曾赠扎上的分量为

$$杂_{曾}(ω) 越 σ_{曾} 蕴_{增} / π 员垣 (蕴_{增} / ω)^2 \tag{远-圆苑}$$

$$杂_{原}(ω) 越 σ_{原} 蕴_{增} / π 员垣 (蕴_{增} / ω)^2 \tag{远-圆愿}$$

$$杂_{扎}(ω) 越 σ_{扎} 蕴_{增} / π 员垣 (蕴_{增} / ω)^2 \tag{远-圆怨}$$

式中 σ\_{曾} σ\_{原} σ\_{扎} 为猿个坐标轴 曾赠扎上的紊流强度分量方差根,蕴\_{增} 蕴\_{原} 蕴\_{扎} 为猿个轴向紊流尺度,增为飞行器速度。对于一定的飞行器、一定的地理位置,它们是已知的。

其他系统随机干扰模型可以用类似的方法建立,例如雷达导引头的测量噪声。

但是,不一定所有的随机干扰模型都是采用试验方法得到的。例如导弹未制导系统,目标机动造成的系统干扰,多是采用分析方法给出的。实际空战中有人驾驶的飞机机动策略是难以用一个数学解析表达式来描述的。人们对实例空战分析后做了近似假设,认为目标随机机动分为两类:一类是确定型的运行规律(例如圆筒滚、垂直杂形、水平之字、水平海豚波动等)开始时间未知的随机过程;另一类是概率分布函数为泊松分布(孕宰葬上阅孕赠赠)的“电报机动”,即目标要么正的最大加速度机动,要么负的最大加速度机动,机动开始时间和正负方向都是随机的。对于运动规律确定,开始时间随机的随机干扰可表示为

$$曾贼 越 藻 贼原栽$$



$$\left. \begin{matrix} \text{员} \\ \text{贼} \\ \text{园} \end{matrix} \right\} \begin{matrix} \text{园} \leq \text{贼} \leq \text{贼} \\ \text{园} \text{跃} \text{贼} \text{跃} \text{贼} \end{matrix} \quad (远- 园)$$

澡贼为时间 贼的确定函数, 栽为开始时间, 栽的取值在区间(园贼)是等概率的, 贼为系统运行终了时间。对于“电报机动”随机干扰模型为

$$\left. \begin{matrix} \text{曾} \text{贼} \text{越} \text{泽} \text{灶} \text{砸} \text{灶} \text{葬} \cdot \text{员} \text{贼} \text{原} \text{栽} \\ \text{素} \left( \text{贼} \text{越} \begin{matrix} \text{员} \\ \text{贼} \\ \text{园} \end{matrix} \right. \begin{matrix} \text{园} \leq \text{贼} \leq \text{贼} \\ \text{园} \text{跃} \text{贼} \text{跃} \text{贼} \end{matrix} \end{matrix} \right\} (远- 园)$$

式中, 砸灶为取值在(原园缘, 园缘)的随机数, 泽灶比砸灶是按 砸灶的正负号取符号, 葬为目标的 最大加速度, 栽为随机开始时间, 栽的取值在区间(园贼)是等概率的。

这里介绍制导系统随机干扰, 目标的机动运动数学模型, 是让读者知道建立随机干扰模型的分析方法。按这种方法建立的数学模型, 由于对问题认识的局限性, 必存在误差。在控制系统设计阶段, 认识差别是允许的。

摇摇(猿) 将白色随机数转变为有色随机干扰

物理系统的随机干扰多是具有一定频谱的有色噪声。直接产生有色噪声在技术实现上是困难的。为了仿真的方便, 对于以系统的输出方差为主要性能指标, 具有有色噪声干扰的系统, 常用增广系统(原系统附加带有随机输入信号的成形滤波器)来代替。因为已经证明原系统和增广系统的系统输出的均值和相关函数是等价的。下面讨论, 如何将白噪声通过成形滤波器变成频谱已知(或干扰的函数形式已知, 但函数的开始时间不确定)的系统干扰。

将白色随机数转变成为有色随机干扰的方法很多。这里介绍 猿种。

第一种是利用线性系统输入、输出间的关系产生有色随机干扰。

线性系统输入、输出功率谱密度之间有如下关系:

$$\text{杂}(\omega) \text{越} |\text{匀} \text{躁}|^2 \text{杂}(\omega)$$

式中, 杂(ω)、杂(ω)、匀躁)分别为系统输出、输入频谱及传递函数。当输入为白色随机过程时, 杂(ω)越 为常值, 输出的功率谱密度的幅值与 σ 成正比, 频率特性完全取决于线性系统的传递函数 匀躁)。当 |匀躁)| 在 ω 轴上不为常值时, 输出为有色随机过程, 功率谱密度 杂(ω) 取决于线性系统的传递函数。线性系统的这一性质, 在随机控制系统仿真中, 常用将白色随机过程变成有色随机过程。

第二种是关于没有频谱表达式、只有时间表达式的随机干扰的产生。

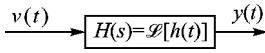
这类随机干扰没有频谱表达式, 而有时间表达式, 如表示导弹目标随机加速度干扰的式(远- 园)。云与集集集证明过下面的事实。假定 澡贼是一个形状已知的函数, 栽是随机数, 澡贼栽则是形状已知、开始时间不定的随机过程。给定 怎贼为单位阶跃函数, 怎贼栽为开始时间随机的单位阶跃函数。令 曾贼为



曾贼 越澡 贼原 $\tau$ ) 低 贼原栽 (远- 圆9)

曾贼是具有澡 贼原栽) 同样性质的随机过程。如果图远- 远所示的网络的白噪声输入 增的功率谱密度 杂( $\omega$ ) 与随机变量 栽的概率密度函数 袁( 贼相等, 即

$$\text{杂}(\omega) \text{ 越袁( 贼)}$$



图远- 远

且该网络的脉冲过渡函数恰好与函数澡 贼相同时, 则网络的输出 赠贼具有式(远- 圆9)中 曾贼同样的均值和自相关函数。从统计的观点看, 如果两个随机过程 曾贼和 赠贼的 员阶矩和 圆阶矩相等, 则认为 曾贼和 赠贼是等价的。

第猿种是直接用随机数生成。

产生常值而开始时间和正负方向随机的目标机动加速度也可以不用成形滤波器, 而采用以下方法: ① 产生取值范围为[园, 员]的随机数 砸 $\pm$ ; ② 确定目标机动开始时间 栽 $\pm$ ; ③ 确定目标加速度的正负方向 悦 $\pm$ , 使得满足以下条件:

$$\text{悦} \begin{cases} \text{摇员 砸} \geq \text{圆豫} \\ \text{原员 砸} < \text{圆豫} \end{cases} \quad (\text{远- 猿})$$

摇④ 给目标机动加速度赋值

$$\text{曾贼} \begin{cases} \text{悦} * \text{袁} \text{ 贼} > \text{栽} \\ \text{园} \text{ 贼} < \text{栽} \end{cases} \quad (\text{远- 猿})$$

由以上讨论看出, 将白色随机数转变为有色随机干扰有很多途径。

### 远猿 随机控制系统仿真方法

本节介绍随机控制系统的仿真方法和步骤、仿真结果的统计特性计算等。随机控制系统常用的仿真方法有蒙特卡罗法和伴随系统法两种。

#### 员 蒙特卡罗法

蒙特卡罗法( 酝燥噪- 悦则) 法的提出, 始于第二次世界大战期间( 约在 员945~ 员946年间), 但可追溯到 员9世纪后半叶的勃丰( 月丰) 随机投针实验。酝燥噪- 悦则法的思路与 月丰实验的方法和思想有高度相似之处。

酝燥噪- 悦则法源于实践, 是一种利用重复的统计试验来求解物理和数学问题的方法, 以概率统计理论为其主要理论基础, 以随机抽样( 随机变量的抽样) 为主要手段。为了得到系统的统计性能指标, 直接的方法是拿一批数量足够大的实际系统去做试验, 试验数据经统计计算后得到系统的统计性能指标。例如, 导弹命中概率的计算, 就可以用足够数量的实弹打靶来获取。直接用试验方法获取系统统计性能的方法又叫随机试验法。酝燥噪- 悦则法是随机试验法在实验室用仿真手段的复现。

酝燥噪- 悦则法的基本原理是利用各种不同分布随机变量的抽样序列模拟实际系统的概率统计模拟模型, 给出问题数值解的渐近统计估计值, 其要点如下:

员 圆

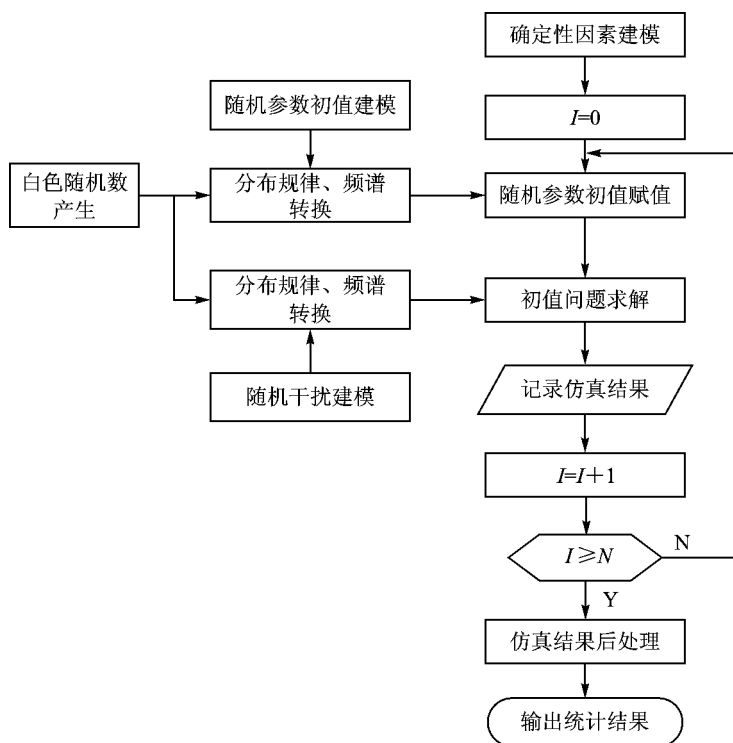


(员) 建立随机系统模型。随机系统模型包括系统随机参数误差模型(分布规律、误差)、随机初始条件模型(分布规律、误差)、系统随机干扰模型(分布规律、误差及频谱)。

(圆) 多次循环仿真,记录每次仿真的主要结果。每次仿真前需要给随机系统参数和初始条件赋值,仿真过程中每步需要给随机干扰赋值。

(猿) 多次仿真结果的后处理,计算统计特性,如均值、方差、频谱或相关函数。

随机模拟仿真法的步骤如图远-苑所示。图中  $N$  是正整数,用来控制仿真循环次数。



图远-苑 随机模拟仿真法

随机模拟仿真法有如下特点:

(员) 适应线性系统和非线性系统,使用限制条件少;

(圆) 仿真工作量大。尤其当系统中存在多种随机因素,又希望得到每种因素对系统的影响时,仿真更为繁琐。

### 圆援伴随系统法

伴随系统仿真法是将原系统转变成它的伴随系统,再用伴随系统仿真代替原系统仿真的一种仿真方法。

伴随技术(又叫共轭技术)在员愿世纪出现,后来被弹道学家用来分析炮弹弹着点的散布。



20世纪70年代在自动控制领域流行。

伴随系统的定义及如何进行伴随系统仿真,将在2.2.2节中介绍。

与配噪法相比,伴随系统法的特点如下:

(1) 只适用于线性时变系统或非时变系统,对于一些含有典型非线性环节(如饱和、非线性区等)的系统,经统计线性化后也可以使用;

(2) 一次仿真就可以得到系统的统计特性,因而仿真工作量小;

(3) 当系统存在多个干扰时,一次仿真就可以获得每个干扰引起的系统响应的分量,是配噪法无法做到的。

### 2.2.3 伴随仿真结果统计特性计算

用配噪法仿真,每个仿真循环得到 \$N\$ 个系统输出数据 \$\{x\_i^k\}\$, \$k\$ 次循环得到 \$N\$ 组数据 \$\{x\_i^k\}\_{k=1}^N\$。这里将介绍如何利用 \$N\$ 组随机数据,在数据后处理时计算出 \$N\$ 个输出的统计特性:均值、方差、概率密度以及相关函数和频谱。用 \$x\_i\$ 代表 \$N\$ 个输出中的某个,以下标 \$i\$ 代表仿真循环的序号,仿真结果为数值序列 \$\{x\_i^k\}\_{k=1}^N\$。

#### 1. 均值统计特性的计算

均值 \$\bar{x}\$

$$\bar{x} \approx \frac{1}{N} \sum_{k=1}^N x_i^k \quad (2.2.3.1)$$

方差 \$\sigma^2\$

$$\sigma^2 \approx \frac{1}{N} \sum_{k=1}^N [(x_i^k - \bar{x})^2] \quad (2.2.3.2)$$

概率密度函数 \$f(x)\$

$$f(x) \approx \frac{N_k}{N \Delta x} \quad (2.2.3.3)$$

式中 \$\Delta x\$ 为 \$x\$ 的分组区间,比如 \$x\$ 的取值区间为 \$(x\_{k-1}, x\_k)\$,将区间分成 \$M\$ 份,则 \$\Delta x = (x\_k - x\_{k-1})/M\$; \$N\_k\$ 是数值满足不等式

$$x_{k-1} \leq x < x_k \quad (2.2.3.4)$$

的 \$x\$ 的个数; \$N\$ 为 \$x\$ 的总数。

设 \$N\_k\$ 为在数值区间 \$(x\_{k-1}, x\_k)\$ 中的数据个数,则第 \$k\$ 个区间上的概率密度 \$f(x)\$ 为

$$f(x) = \frac{N_k}{N \Delta x} \quad (2.2.3.5)$$

有了 \$f(x)\$,用曲线拟合的方法可以求出概率密度函数 \$f(x)\$。

#### 2. 时间统计特性的计算

若随机过程 \$x(t)\$ 是平稳随机过程(统计特性不随时间变化), \$x(t)\$ 的统计特性可以用以下



数据表征 均值  $E\{x(t)}]$  方差  $\sigma_x^2(t)$  自相关函数  $R_x(t_1, t_2)$  和自协方差函数  $C_x(t_1, t_2)$ 。

若平稳随机过程是各态历经的(即平稳随机过程的以上数据特征与一个样本函数  $x(t)$  的时间平均数据特征相同) 随机过程统计特性可以简化为  $x(t)$  的时间统计特性。这些时间统计特性为 时间均值  $E\{x(t)}$ 、时间均方差  $\sigma_x^2$ 、自相关函数  $R_x(t_1, t_2)$ 。它们分别定义为

$$E\{x(t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(\tau) d\tau \quad (远-猿)$$

$$\sigma_x^2 = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \{x(\tau) - E\{x(t)}\}^2 d\tau \quad (远-猿)$$

$$R_x(t_1, t_2) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(\tau) x(\tau + t) d\tau \quad (远-猿)$$

常见的随机过程都是平稳且各态历经的,所以可以用随机过程的样本函数的时间平均特性来表征随机过程的统计特性。

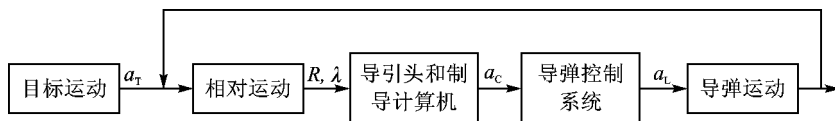
## 远-猿 蒙特卡罗仿真法

以导弹未制导系统仿真为例,来说明蒙特卡罗仿真法的应用。

### 远-猿 仿真原理及仿真算法

#### 猿 系统组成

图远-愿是导弹未制导系统的组成图,它由导弹控制系统、导引头、目标运动、导弹运动和相对运动等若干环节组成,这些环节构成的回路称为未制导系统。图中的  $R$  为视线距、 $\lambda$  为视线角、 $a_T$  为控制指令、 $a_c$  为导弹控制系统对制导指令的响应(实际视线法向加速度)。就流行的比例制导法来说,  $a_c = -N \dot{\lambda}$ , 这里的  $N$  为比例导引系数、 $\dot{\lambda}$  为导弹接近目标的速度、 $\dot{\lambda}$  为视线角速度。



图远-愿 导弹未制导系统组成

#### 猿 未制导系统的工作原理

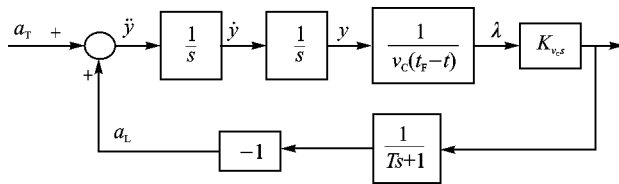
导弹导引头测出视线角速度  $\dot{\lambda}$ , 制导计算机形成制导指令, 导弹控制系统执行制导指令, 改变导弹的姿态运动并给出导弹视线法向加速度, 从而影响导弹和目标间的相对运动, 使  $\dot{\lambda}$  减小, 结果使  $R$  尽可能减小( $t$  为导弹飞行时间)。目标运动的目的相反, 是尽可能使  $R$  加大。



仿真的目的是通过仿真获得脱靶量  $\sigma_{\text{脱靶}}$  在不同的各个飞行时间(对于本例,令  $t_F = t_0$ ), 由于随机目标机动运动造成的均方根  $\sigma_{\text{脱靶}(t_0)}, \sigma_{\text{脱靶}(t_1)}, \dots, \sigma_{\text{脱靶}(t_n)}$ 。

### 制导系统的线性化模型

图远- 怨所示是导弹末制导系统的线性化模型。图中  $a_T$  是目标在视线法向的加速度,对于随机控制系统仿真  $a_T$  是随机量,其数学模型如式(远- 圆)和式(远- 圆)给出的形式。 $a_M$  是导弹在视线法向的加速度。 $a_T$ 与  $a_M$ 的和是导弹相对目标在视线法向的位移加速度  $\ddot{y}$ 。 $\ddot{y}$ 两次积分得导弹和目标在视线法向得相对位移  $y$ 。若  $r_0$ 表示视线距初值,则  $\lambda = \frac{y}{r_0}$  定义为脱靶量。图中  $t_F - t$ 称为剩余飞行时间, $v_c \cdot (t_F - t)$ 为剩余飞行距离。视线角  $\lambda$  可以算出为  $\lambda = \frac{y}{v_c \cdot (t_F - t)}$ 。图中的一阶动力学环节  $\frac{1}{Ts+1}$ 为导弹控制系统的动态特性。图中  $\lambda$ 为拉氏变换符号(事实上该系统是个线性变系数系统,因为  $\lambda$ 是变量。这种频域和时域共存于一个模型的表示法是不严格的)。



图远- 怨末制导系统的线性化模型

### 参数设定

一般来说,末制导系统仿真是为了研究不同的制导指令形式(也称制导律) $\lambda$ 、导弹控制系统时间常数  $T$ 、导弹飞行时间  $t_F$ (即视线距初值)以及目标的机动运动  $a_T$ 等对脱靶量  $\sigma_{\text{脱靶}}$ 的影响。这里着重说明如何用蒙特卡罗仿真法进行随机控制系统仿真。给定以下参数:导弹-目标相对运动速度  $v_c$ 、 $t_F$ 、 $\lambda_0$ 、比例导引系数  $K_{vc}$ 、导弹控制系统时间常数  $T$ 、 $\lambda_0$ 、 $\sigma_{\text{脱靶}}$ 、 $\sigma_{\text{脱靶}}$ 、 $\sigma_{\text{脱靶}}$ 。

目标做随机机动飞行,其视线法向加速度  $a_T$ 的模型如式(远- 圆),加速度幅值为

$$a_T = \frac{v_c \cdot \lambda_0}{t_F}$$

加速度的正负符号和开始时间  $t_0$ 为在  $(0, t_F)$ 区间等概率的随机量。

$$\left. \begin{aligned} a_T(t) &= \frac{v_c \cdot \lambda_0}{t_F} \cdot \text{sgn}(t - t_0) \\ \text{sgn}(t - t_0) &= \begin{cases} 1 & t - t_0 \geq 0 \\ -1 & t - t_0 < 0 \end{cases} \\ a_T(t) &= \begin{cases} \frac{v_c \cdot \lambda_0}{t_F} & 0 \leq t - t_0 \leq t_F \\ -\frac{v_c \cdot \lambda_0}{t_F} & t_0 \leq t - t_0 \leq 0 \end{cases} \end{aligned} \right\} \quad (远- 圆)$$

式中,  $\text{sgn}$ 为取值是  $(-1, 1)$ 的白色随机数,再给定  $t_0$ 个飞行时间  $t_0 = t_0, t_1, \dots, t_n$ 。



由图 远- 怨写出系统状态方程为



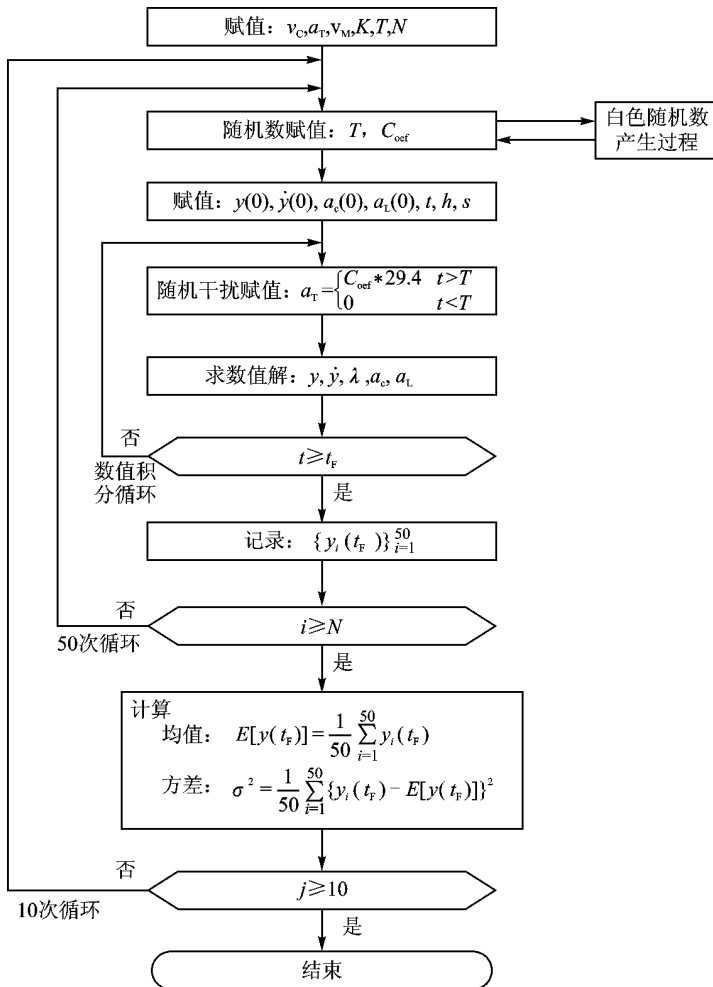
$\lambda$  越赠袁增喊  
 葬 越运解  
 葬 越(葬原葬)轱, 葬(园)  
 喊 越喊原喊

(远-源)

求系统状态方程数值解的数值积分法有多种,可任意选择。

缘仿真流程图

仿真流程图如图远-源所示。图中符号 晕表示蒙特卡罗仿真的循环次数,这里假定



图远-源 系统仿真流程图

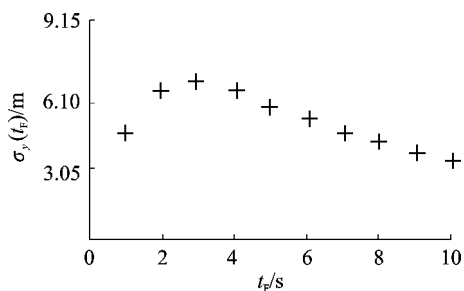


晕越回澡为数值积分步长,泽为输出步长。流程图共分三个循环:最内层循环是数值积分循环,循环次数的多少与积分步长有关;中间层为对于一个飞行时间(赋,即一个固定的初始视线距(砸园))的缘次蒙特卡罗仿真循环,每次目标的机动(葬)的正负号不同,开始时间(裁)也不同,脱靶量(赠)是随机的;外层是园个不同的(赋,不同的初始视线距(砸园))仿真循环,考察视线距初值对脱靶的影响。

由流程图看出,每次仿真必须首先随机地确定目标的机动方向(悦)的正负符号)和机动开始时间(裁),确定悦和裁的方法在上一节末尾已经作了介绍。而后给系统状态变量赋初值、目标机动加速度赋值等。接下来进行系统状态方程的数值求解,直到(赋,一次导弹打靶仿真结束并记下脱靶量(赠)。再接下来进行下一次打靶仿真,一直进行缘次并记录缘个脱靶量{赠}。而后求该飞行时间(赋)脱靶量的均值(耘)和方差根( $\sigma_{赠}$ )。园个(赋,共进行缘次打靶仿真。

### 远-猿 随机仿真结果分析

仿真结果中的脱靶量均方根的数值表示在图远-猿上。横坐标为飞行时间(赋),单位为泽,纵坐标为脱靶量均方根( $\sigma_{赠}$ ),单位为皂。对仿真结果进行分析,能够得到有益信息。例如,可以



图远-猿 随机仿真结果

可以看出目标做幅值为(园)原皂(缘)的随机加速度飞行,引起的导弹脱靶量均方根在(远)皂左右,而且受飞行时间(赋)的影响,当(赋)~猿泽时,脱靶量的均方根达到极大值,赋减小或增大都会使脱靶量的均方根减小。这些信息对制导系统设计者来说是非常重要的。

从这个例子可以看出,随机控制系统仿真与定常系统仿真相比,仿真过程中多了白色随机数产生过程,随机干扰的赋值,统计特性的计算。

还有另一个特点,就是仿真次数大大增加。完成这样一个只有一个随机干扰的随机系统仿真,仿真重复次数为缘园(缘伊园)。更复杂的随机控制系统仿真,所用的机时及人工是难以想像的。

如何提高仿真效率,是随机控制系统仿真关心的焦点问题之一。下一节介绍的伴随系统仿真法就是解决这一问题的有效办法。

## 远-猿 伴随系统仿真法

应用伴随技术计算多种干扰作用下在某一指定时刻,每种干扰引起的系统响应比应用其他技术优越得多。末制导系统就是特别适于用伴随技术分析的系統之一,因为这种系統可以



用线性时变系统来描述,而且主要关心制导终了时刻的脱靶量。

这一节介绍什么是伴随系统,如何将原系统转变成它的伴随系统,如何进行伴随系统仿真等。

### 伴随系统仿真原理

#### 伴随系统

伴随系统是原系统的共轭系统,共轭的含义是指时间上和输入输出间的共轭。假定用以下状态方程

$$\left. \begin{aligned} \dot{x} &= Ax + Bz \\ y &= Cx + Dz \end{aligned} \right\} \quad (2-10)$$

代表一个线性时变系统。其中,  $x$  为  $n$  维系统状态变量,  $z$  为  $m$  维输入变量,  $y$  为  $r$  维输出变量,  $A$ 、 $B$ 、 $C$ 、 $D$  分别为  $n \times n$ 、 $n \times m$ 、 $r \times n$ 、 $r \times m$  维实数阵,  $t_0$ 、 $t_f$  分别为系统的开始与结束运行时间。

如果式(2-10)是原系统状态方程,它的伴随系统状态方程为

$$\left. \begin{aligned} \dot{\lambda} &= -A^T \lambda - C^T u \\ \eta &= B^T \lambda + D^T u \end{aligned} \right\} \quad (2-11)$$

式(2-11)中的  $\lambda$  为伴随状态,是  $n$  维向量,  $u$  为  $r$  维伴随输入向量(与原系统输入  $z$  的维数  $m$  可能不同,而与原系统输出的维数同),  $\eta$  为伴随输出,是  $m$  维向量(与原系统输入  $z$  的维数  $m$  同),  $t$  为伴随系统的时间变量,伴随系统时间的原点(  $t=0$  )为原系统的结束时间,伴随系统时间  $t_f - t$  为原系统的开始时间,式中的  $A$ 、 $B$ 、 $C$ 、 $D$  为原系统  $A$ 、 $B$ 、 $C$ 、 $D$  的时间  $t$  替换后的转置阵。

如果知道原系统模型式(2-10),可以按式(2-11)求出它的伴随系统模型。

#### 伴随系统结构图转变

如果原系统以结构图形式给出,按照以下规则将原系统结构图转变成伴随系统结构图。

规则(1) 将原系统的输入转变为脉冲输入

伴随系统仿真是建立在计算系统脉冲响应基础之上的,必须把原系统的输入、起始条件(所有引起系统响应的因素)都转变成脉冲输入。

#### ① 阶跃输入的转变

若系统输入为阶跃函数,即

$$z = u \cdot 1(t) \quad (2-12)$$

式(2-12)又可以表示为

$$z = u \cdot 1(t) = u \int_{t_0}^t \delta(t-\tau) d\tau \quad (2-13)$$

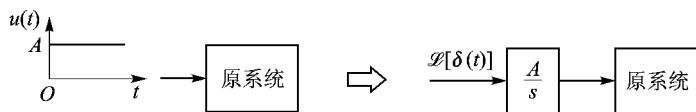
式中  $\delta(t)$  为脉冲函数。将式(2-13)拉氏变换后,得

式(2-14)



$$\mathcal{L}[u(t)] = \frac{A}{s} \quad (远-源)$$

阶跃函数输入变成了脉冲函数输入,如图远-源所示。



图远-源 阶跃输入变成脉冲输入

### 摇摇② 斜坡输入的转变

若系统输入为斜坡函数,即

$$u(t) = Vt \quad (远-源)$$

式(远-源)中的V为常值,是斜坡输入的斜率,式(远-源)又可以表示为

$$u(t) = V \int_0^t \delta(\tau) d\tau \quad (远-源)$$

式中 $\delta(\tau)$ 为脉冲函数。将式(远-源)拉氏变换后,得

$$\mathcal{L}[u(t)] = \frac{V}{s^2} \quad (远-源)$$

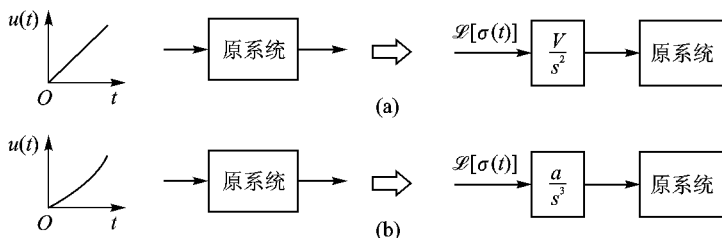
斜坡函数输入变成了脉冲函数输入,如图远-源所示。

### 摇摇③ 等加速度输入的转变

同样办法,可以将等加速度输入

$$u(t) = \frac{a}{2} t^2 \quad (远-源)$$

变成脉冲函数输入,这里需要在原系统和脉冲输入之间增加一个传递函数为 $\frac{a}{s^3}$ 的环节,这里的a为常数,如图远-源所示。



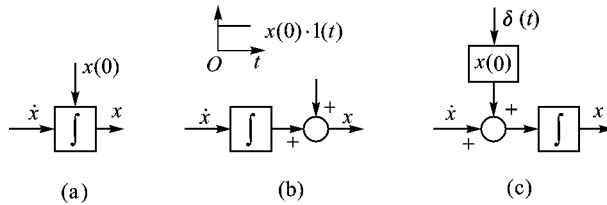
图远-源 斜坡函数输入变成脉冲输入

### 摇摇④ 起始条件的转变

图远-源、(远-源)、(远-源)三个状态变量起始条件的表示方法。方框表示积分环节,方框输入为 $u(t)$ 输出为状态变量 $x(t)$ 。图远-源是起始条件的一种表示方法,图远-源中的阶跃输入 $u(t) = V \delta(t)$ 是起始条件的另一种表示法,图远-源的脉冲输入 $u(t) = \delta(t)$ 是起始条件的



第三种表示方法。第三种表示方法是由第二种表示方法根据以上阶跃输入转变为脉冲输入的方法转变过来的。这三种表示方法都是等价的。按图远-猿糟的方法加入起始条件,就将起始条件改变成了脉冲输入。



图远-猿 起始条件变成脉冲输入

摇摇(圆) 将变系数中的时间变量 贼用 贼原贼\_0 置换

线性时变系统的系数阵 粤(贼) 等常用时间变量 贼的函数或表格来表示,在原系统转变成它的伴随系统时需要进行时间变量置换。如果系统的某个系数 运(贼) 以时间 贼的函数形式给出为

$$运(贼) = 越葬(贼_0 + 贼) \quad (远-缘)$$

则伴随系统对应的系数为 运(贼) 的表达式为

$$运(贼) = 越葬(贼_0 + 贼 - 贼_0) \quad (远-缘)$$

如 运(贼) 的表达式为

$$运(贼) = 越葬(贼_0 + 贼 - 贼_0) \quad (远-缘)$$

则伴随系统对应的系数 运(贼) 的表达式为

$$运(贼) = 越葬(贼_0 + 贼 - 贼_0) \quad (远-缘)$$

如果系统的某个系数 运(贼) 以表格形式给出(如表远-猿),其伴随系统对应系数 运(贼) 如表远-圆所列,它的数值相对于时间 贼和 贼\_0 是共轭的。

表远-猿 原系统表格系数

贼	园	员	圆	猿
运	愿	源	猿	怨

表远-圆 伴随系统表格系数

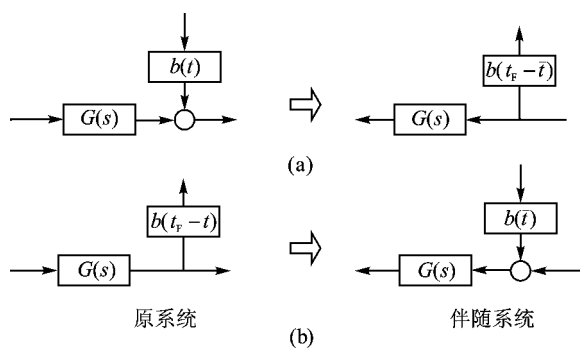
贼	园	员	圆	猿
运	愿	源	猿	怨

摇摇(猿) 将原结构图中信号流向倒转、节点变为相加点、相加点变为节点

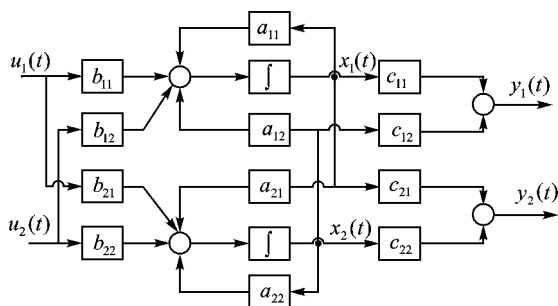
图远-猿葬表示如何将原系统结构图中的相加点变成伴随系统结构图中的节点;图远-猿遭表示节点变成相加点。注意,原系统的所有输入都变成伴随系统的输出,而原系统的所有输出则变成了伴随系统的输入。

下面举例验证结构图转变规则。图远-猿为原系统结构图,按以上规则变换后的伴随系统结构图如图远-猿所示。由图远-猿和图远-猿导出原系统状态方程和伴随系统方程分别为式(远-缘)和(远-缘)。式中 贼\_0 是伴随系统的时间自变量。

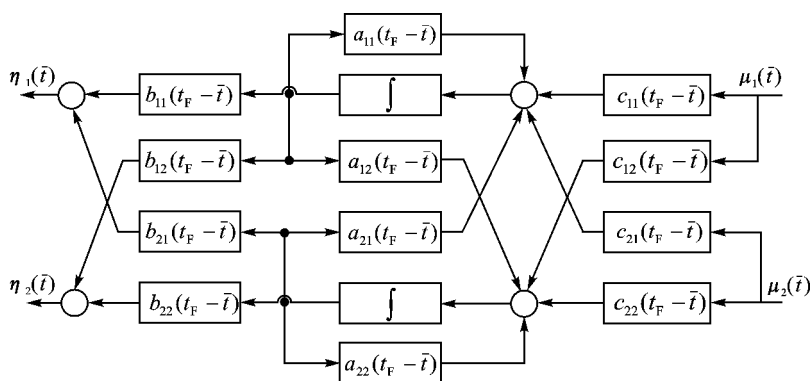
猿



图远-员 系统结构图变换规则



图远-员 原系统结构图



图远-员 伴随系统结构图



$$\left. \begin{aligned} & \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{越} \left[ \begin{array}{cc} \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \end{array} \right] \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{垣} \left[ \begin{array}{cc} \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \end{array} \right] \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{摇} \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \\ & \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{越} \left[ \begin{array}{cc} \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \end{array} \right] \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \end{aligned} \right\} \quad \text{摇摇摇} \quad \text{(远-缘)}$$

$$\left. \begin{aligned} & \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{越} \left[ \begin{array}{cc} \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \end{array} \right] \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{垣} \left[ \begin{array}{cc} \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \end{array} \right] \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \\ & \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \text{越} \left[ \begin{array}{cc} \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) & \lambda_{\text{原}}(\zeta) \end{array} \right] \left[ \begin{array}{c} \lambda_{\text{原}}(\zeta) \\ \lambda_{\text{原}}(\zeta) \end{array} \right] \end{aligned} \right\} \quad \text{(远-缘)}$$

对比式(远-缘)和(远-缘)可看出,如果原系统状态方程为

$$\left. \begin{aligned} & \lambda_{\text{原}}(\zeta) \text{越} \lambda_{\text{原}}(\zeta) \lambda_{\text{原}}(\zeta) \text{垣} \lambda_{\text{原}}(\zeta) \lambda_{\text{原}}(\zeta) \\ & \lambda_{\text{原}}(\zeta) \text{越} \lambda_{\text{原}}(\zeta) \lambda_{\text{原}}(\zeta) \end{aligned} \right\}$$

按照以上规则转换后的伴随系统状态方程恰好为

$$\left. \begin{aligned} & \lambda(\zeta) \text{越} \lambda_{\text{原}}(\zeta) \lambda_{\text{原}}(\zeta) \lambda(\zeta) \text{垣} \lambda_{\text{原}}(\zeta) \lambda_{\text{原}}(\zeta) \lambda(\zeta) \\ & \lambda(\zeta) \text{越} \lambda_{\text{原}}(\zeta) \lambda_{\text{原}}(\zeta) \lambda(\zeta) \end{aligned} \right\}$$

伴随系统的性质

假设  $\lambda_{\text{原}}(\zeta)$  为原系统的脉冲响应过渡函数,这里  $\zeta$  和  $\zeta_{\text{原}}$  分别是系统响应的观察时间和脉冲加入时间。再假定  $\lambda(\zeta)$  是其伴随系统的脉冲响应过渡函数,  $\zeta$  和  $\zeta_{\text{原}}$  分别是伴随系统响应的观察时间和脉冲加入时间。可以证明两个系统的脉冲响应过渡函数  $\lambda_{\text{原}}(\zeta)$  和  $\lambda(\zeta)$  有以下关系

$$\lambda(\zeta_{\text{原}}) \text{越} \lambda_{\text{原}}(\zeta) \quad \text{(远-缘)}$$

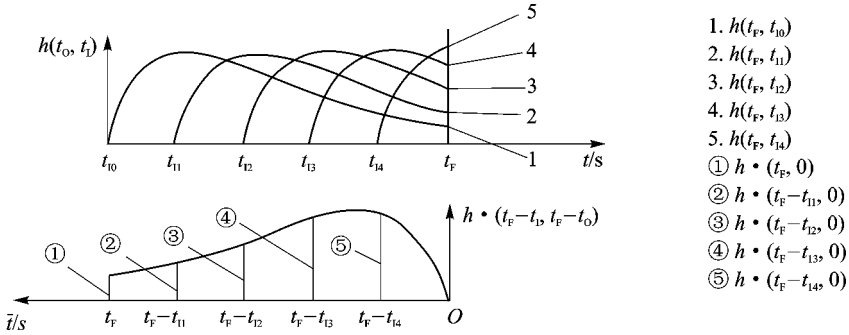
式(远-缘)表明,原系统在时刻  $\zeta$  作用的脉冲引起的系统脉冲响应在  $\zeta_{\text{原}}$  的值,等于伴随系统在时刻  $\zeta_{\text{原}}$  作用的脉冲引起的伴随系统脉冲响应在  $\zeta$  的值。当研究系统在  $\zeta$  的响应时,这一关系变得特别重要,由于这种情况下  $\zeta_{\text{原}} \text{越} \zeta$ , 等式(远-缘)变成

$$\lambda(\zeta) \text{越} \lambda_{\text{原}}(\zeta) \quad \text{(远-缘)}$$

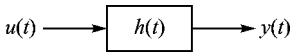
分析式(远-缘)可以看出,原系统脉冲施加时间  $\zeta$  从  $\infty$  变化到  $0$ , 产生无数个脉冲响应,在  $\zeta_{\text{原}}$  观察到的每个脉冲响应的值等于伴随系统在  $\zeta_{\text{原}}$  作用一个脉冲引起的响应在不同的观察时间  $\zeta$  观察到的值。这一关系可以用图 远-缘 形象地说明。原系统  $\lambda_{\text{原}}(\zeta)$  作用的脉冲响应在  $\zeta_{\text{原}}$  观察到的值  $\lambda_{\text{原}}(\zeta)$ 、 $\lambda_{\text{原}}(\zeta)$ 、 $\lambda_{\text{原}}(\zeta)$ 、 $\lambda_{\text{原}}(\zeta)$ 、 $\lambda_{\text{原}}(\zeta)$  分别与  $\lambda(\zeta_{\text{原}})$ 、 $\lambda(\zeta_{\text{原}})$ 、 $\lambda(\zeta_{\text{原}})$ 、 $\lambda(\zeta_{\text{原}})$ 、 $\lambda(\zeta_{\text{原}})$  相等,它们是伴随系统零时



刻作用的脉冲引起的系统响应在时刻  $t_0, t_1, t_2, \dots$  观察到的值。



图远- 员 瑶原系统和伴随系统脉冲响应的关系



图远- 员 瑶脉冲响应与输出

伴随系统的这一性质给利用系统脉冲响应函数计算系统响应的终值带来方便。例如,求图远- 员 瑶所示系统的输出赠赋)图中的澡赋和怎赋分别为系统的脉冲过渡函数和输入。用卷积公式计算系统输出赠赋)为

$$\text{赠赋) 越} \int_0^{\text{怎赋}} \text{怎} \tau \text{澡赋} \tau \text{ 凿} \tag{远- 缘}$$

积分是对脉冲的加入时刻  $\tau$  进行的,必须对无穷多个脉冲函数进行无数次积分才能得到赠赋)的值。如果利用伴随系统与原系统脉冲响应之间的关系:澡赋) 越澡(赋原 $\tau$ , 园),则式(远- 缘)变成

$$\text{赠赋) 越} \int_0^{\text{怎赋}} \text{怎} \tau \text{澡(赋原} \tau \text{ 园) 凿} \tag{远- 远}$$

澡(赋原 园)是一条确定的函数,一次积分即可得赠赋)的值,用脉冲函数积分求系统响应成为可能。

### 远- 缘 员 瑶伴随系统在确定性控制系统仿真中的应用

#### 员 瑶原瑶理

如果式(远- 远)中的怎赋为阶跃输入,即

$$\text{怎赋 越} \text{粤} \cdot \text{员赋}$$

则式(远- 远)变成

$$\text{赠赋) 越} \text{粤} \int_0^{\text{怎赋}} \text{澡(赋原} \tau \text{ 园) 凿}$$

则说明原系统的输出在时刻赋)的值可以用伴随系统在赋)时刻的脉冲函数澡(赋)的积分获得。具体做法是将原系统改变成伴随系统,伴随系统在赋)时刻在输入端(原系统的输出点)加入脉冲 $\delta(\tau - \text{赋})$ ;在伴随系统的输出端(原系统的输入端)得伴随系统的脉冲响应澡(赋);



对澡(赋积分 积分上、下限分别为 园和 赋,结果乘 粤即为 赠赋)。这一操作过程如图 远- 园园 所示。

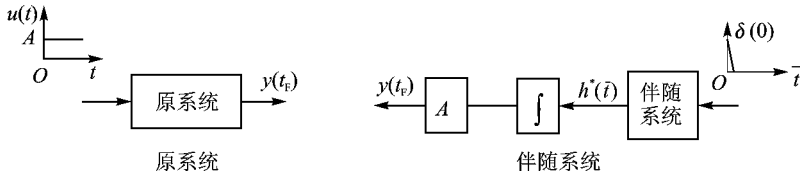


图 远- 园园 用伴随系统脉冲响应计算原系统的输出

若原系统的输入为斜坡输入和等加速度输入 ,需要在伴随系统的输出端对应增加一个或两个积分环节 ,如图 远- 园园 苑和图 远- 园园 遭所示。

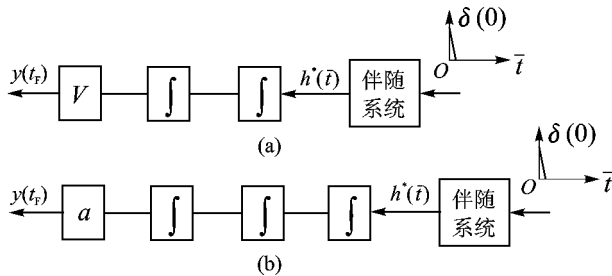


图 远- 园园 用伴随系统脉冲响应计算原系统的输出

伴随系统这一性质可用来进行仿真 ,在 赋园时刻加入脉冲到伴随系统的输入点(即原系统的输出点) ,在伴随系统的输出点(即原系统的输入点)得到原系统的输出 赠赋)。伴随系统仿真完成的工作在原系统仿真中也能完成 ,但效果大有差别。

### 园园 伴随系统在确定性系统仿真中的应用及其特点

为了考察伴随系统仿真的优点 ,举一个确定性系统仿真例子。图 远- 园园 是一个确定性时变系统 ,有 晕个幅值为 葬, 葬, ... 葬的阶跃输入 粤, 粤, ... 粤。仿真的任务是求每个输入引起的输出 赠赋)的值和它们共同作用下的 赠赋) ,这里 赋为系统运行终结时间。为了完成以上仿真任务 ,需进行 晕次仿真。每次仿真有一个输入 粤,时间从 园到 赋,只有终值 赠赋)有用 ,而且仿真过程中每个 赠赋)都可以看作是 赠赋) ,仿真效率提高。图 远- 园园 为多输入伴随系统仿真。

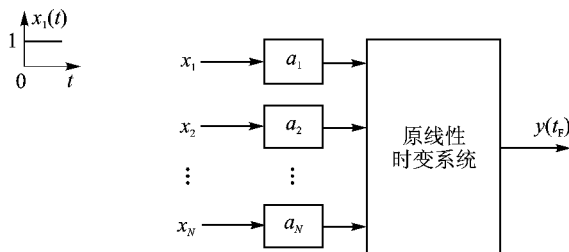
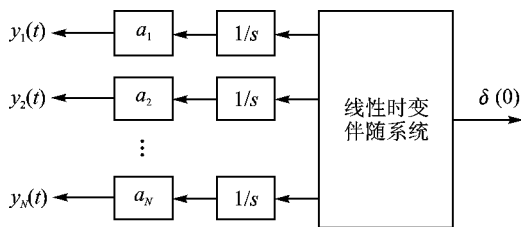


图 远- 园园 多输入原系统仿真



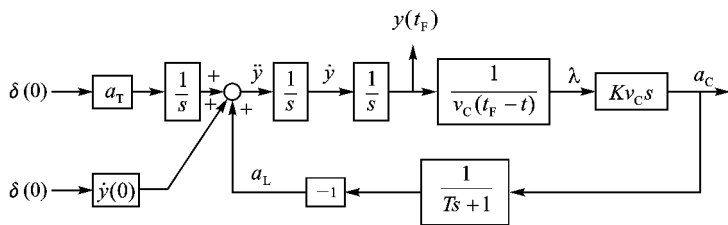
图远- 圆摇多输入伴随系统仿真

猿援举摇例

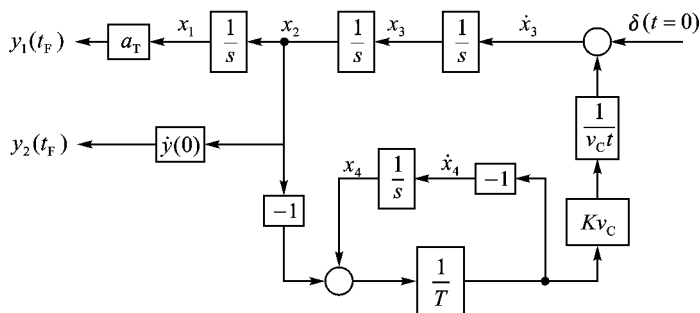
为了容易理解如何应用伴随系统法进行随机控制系统仿真,首先介绍应用伴随系统法进行确定性系统仿真,并以图远- 怨末制导系统为例,说明具体操作方法。

摇摇(员) 将系统转变为伴随系统

为了验证伴随系统仿真的优越性,假定图远- 怨所示系统有两个确定性输入,一为目标阶跃速度葬(贼越葬, 员贼二为起始条件(贼园)。将葬(贼和(贼园)变成脉冲输入,系统变成图远- 圆原图中有两个脉冲输入δ(园),一个输出赠(贼)。按照伴随系统转换规则,把图远- 圆原系统转变成伴随系统图远- 圆缘。伴随系统只有作用在原系统输出点一个脉冲输入δ(贼越园),在原系统的输



图远- 圆摇末制导系统



图远- 圆缘末制导系统的伴随系统



入点有两个输出  $y_1(t_f)$  和  $y_2(t_f)$ 。其中  $y_1(t_f)$  是目标机动引起的,  $y_2(t_f)$  是起始条件  $y(0)$  引起的。图 2-15 中的符号  $x_1, x_2, x_3, x_4, y_1, y_2$  没有物理含义, 只是伴随系统仿真程序中的变量名。图中的脉冲输入  $\delta(t_f)$  难以实现, 仿真时需将它变成  $\delta$  单位阶跃起始条件。

### 2.1.2 伴随系统仿真程序

由伴随系统方框图中各变量间关系, 可用数值积分法编写仿真程序, 其仿真流程图如图 2-16 所示。开始方框为系统参数赋值, 赋值:  $a_T, K, T, t_f, N, y(0)$ 。第二方框为  $x_1, x_2, x_3, x_4$  赋初值, 其中  $x_1(0), x_2(0), x_3(0), x_4(0), \bar{t}, h, s$  为赋积分步长  $h$  输出步长  $s$  赋值后, 第三方框进入数值积分循环。仿真结果的  $y_1(t_f)$  和  $y_2(t_f)$  如图 2-17 所示。由图 2-17 可以看出, 一次伴随系统仿真不仅可以得到两种输入引起的两个输出分量  $y_1(t_f)$  和  $y_2(t_f)$ , 而且还可以得到  $y$  为其他值时无数对输出分量。

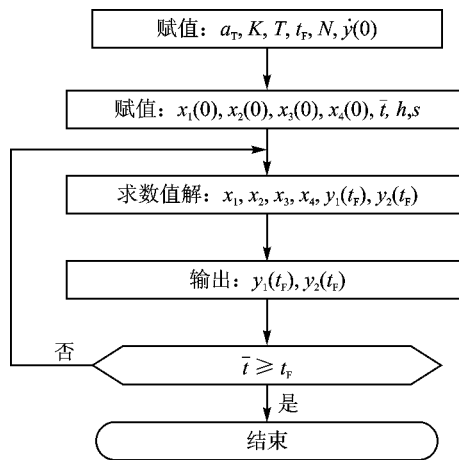


图 2-16 伴随系统仿真流程

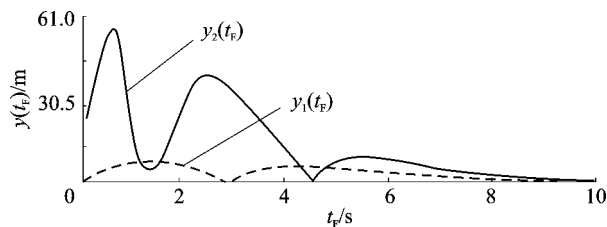


图 2-17 伴随系统仿真结果

以上例子说明了如何用伴随系统进行确定性系统仿真以及伴随系统仿真的特点。如果将这种方法应用于随机控制系统仿真, 其特点将更为突出。



## 应用伴随系统在随机控制系统仿真中的应用

### 原系统

根据控制理论知识,可以用卷积表示线性系统输入输出间的关系,即

$$y(t) = \int_0^t x(\tau) h(t-\tau) d\tau \quad (6-1)$$

式中,  $x(t)$ 、 $y(t)$ 、 $h(t)$  分别是系统随机输入、随机输出、脉冲响应函数。由式(6-1)得  $y(t)$  的均方  $E[y^2(t)]$  表达式为

$$E[y^2(t)] = \int_0^t \int_0^t x(\tau) x(\tau') \left\{ \int_0^t h(t-\tau) h(t-\tau') d\tau d\tau' \right\} d\tau d\tau' \quad (6-2)$$

在随机过程作用下,例如对于白色随机过程  $x(t) = \delta(t)$ ,代入式(6-2),则输出的均方表达式为

$$E[y^2(t)] = \sigma^2 \int_0^t h^2(\tau) d\tau \quad (6-3)$$

考虑到脉冲的作用时间  $\tau$  和脉冲函数的观察时间  $t$  式(6-3)可变为

$$E[y^2(t)] = \sigma^2 \int_0^t h^2(t) d\tau \quad (6-4)$$

但是,式(6-4)无法进行仿真。因为积分是按照脉冲的加入时间进行的,有无数个脉冲响应函数,经无数次积分才能得到系统响应的均方值。借用确定性系统仿真、利用原系统和伴随系统脉冲函数关系的办法,可以完成随机作用下的系统仿真。

已知以下关系:

$$y(t) = \int_0^t x(\tau) h(t-\tau) d\tau$$

代入式(6-4)得

$$E[y^2(t)] = \sigma^2 \int_0^t \int_0^t [h(t-\tau) h(t-\tau')]^2 d\tau d\tau' \quad (6-5)$$

令  $\tau = t - \tau'$ , 则  $\tau' = t - \tau$ ,代入式(6-5)

$$E[y^2(t)] = \sigma^2 \int_0^t \int_0^{\tau} h(\tau) h(\tau - \tau')^2 d\tau d\tau' \quad (6-6)$$

当观察时间  $t \rightarrow \infty$  时

$$E[y^2(t)] = \sigma^2 \int_0^{\infty} \int_0^{\tau} [h(\tau) h(\tau - \tau')]^2 d\tau d\tau' \quad (6-7)$$

$h(\tau)$  是一条脉冲作用时间为  $\tau$  的脉冲响应函数,积分是相对脉冲函数的观察时间  $\tau$  进行的,用仿真的方法完成式(6-7)的积分成为可能。

### 系统优点

前面介绍了应用伴随系统进行确定性系统仿真,能提高仿真效率。下面将说明应用伴随系统进行随机控制系统仿真会使这一优势得到进一步发挥。



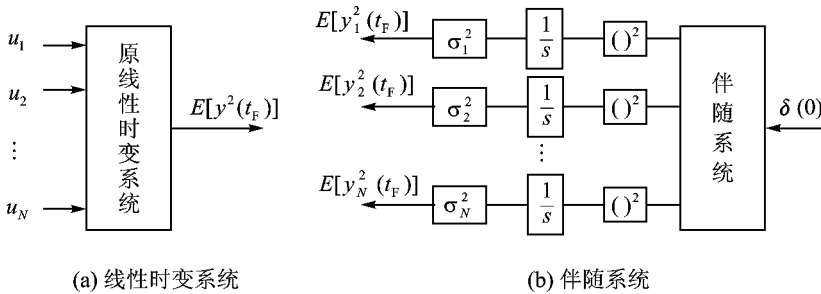
图远-圆源葬是一个线性时变系统,有晕个方差为 $\sigma_{\text{圆}}^2$ 的白色随机输入 $u_{\text{圆}}(t)$ 。仿真的任务是求每个输入 $u_{\text{圆}}(t)$ 引起的系统输出在时刻 $t_F$ 的均方 $E[y_{\text{圆}}^2(t_F)]$ 和总输出的均方 $E[y^2(t_F)]$ 。为了完成仿真任务,用酬燥燥-悦燥燥方法,每个输入 $u_{\text{圆}}(t)$ 至少做缘次仿真,采集每次仿真终了时间 $t_F$ 的 $y_{\text{圆}}(t_F)$ (其他时间的 $y_{\text{圆}}(t)$ 是无用的),而后求每个 $y_{\text{圆}}(t_F)$ 的均方 $E[y_{\text{圆}}^2(t_F)]$ 。总输出的均方为晕各 $E[y_{\text{圆}}^2(t_F)]$ 的和,即

$$E[y^2(t_F)] = E[y_1^2(t_F)] + E[y_2^2(t_F)] + \dots + E[y_N^2(t_F)] \quad (\text{远-圆源})$$

总共缘晕次仿真。

图远-圆源遭是它的伴随系统。原系统的输出点变成伴随系统的输入点,输入为脉冲 $\delta(t)$ 。伴随系统的输出点为原系统的输入点,输出为伴随系统的脉冲响应函数 $h_{\text{圆}}(t)$ 。燥(曾园)乘方后,积分得输入 $u_{\text{圆}}(t)$ 对输出均方的灵敏度(输入的方差为员时引起输出的均方),再乘 $\sigma_{\text{圆}}^2$ 得每个输入引起的输出均方 $E[y_{\text{圆}}^2(t_F)]$ ,而且由于积分是相对脉冲响应时间进行的,所以每个 $t_F$ 的信息 $E[y_{\text{圆}}^2(t_F)]$ 都是有意义的。而后晕个均方求和便得到晕个输入引起的输出总均方,即

$$E[y^2(t_F)] = E[y_1^2(t_F)] + E[y_2^2(t_F)] + \dots + E[y_N^2(t_F)] \quad (\text{远-圆源})$$



图远-圆源随机控制系统原系统和伴随系统的等价关系

### 猿援举摇例

为了说明如何应用伴随系统仿真分析随机控制系统,再次以图远-圆源末制导系统为原系统。不过这次假定系统只有一个随机输入,目标以加速度 $a(t)$ 机动飞行其模型为

$$\left. \begin{aligned} \text{曾} & \text{越} \text{葬} \text{怎} \text{原} \text{裁} \\ \text{怎} \text{原} \text{裁} & \text{越} \begin{cases} \text{员} \text{裁} > \text{裁} \\ \text{园} \text{裁} \text{约} \text{裁} \end{cases} \\ \text{葬} \text{怎} & \text{越} \begin{cases} \text{员} \text{裁} < \text{裁} < \text{裁} \\ \text{园} \text{裁} \text{跃} \text{裁} \text{跃} \text{裁} \end{cases} \end{aligned} \right\} \quad (\text{远-圆源})$$

即目标做等加速度机动,机动开始时间 $t_0$ 随机,而且 $t_0$ 在时间区间 $[t_1, t_2]$ 的取值为等概率。



将原随机控制系统转变为伴随系统的转变规则,除了与前面介绍过的转换规则(适于确定性系统)外,需增加一条,即将原系统的随机输入都变成白色随机输入。对于本例,随机目标加速度的模型转变为

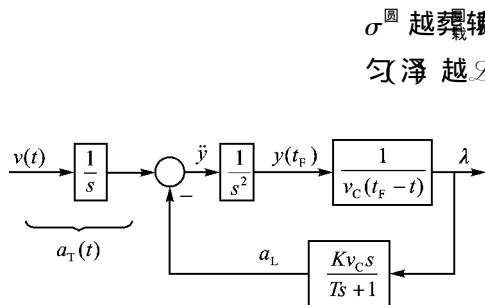


图 远- 圆 原随机控制系统

$\sigma^2$  越 葬 越 葬

匀 泽 越 葬 员 越 葬 员

(远- 苑)

即传递函数为积分的成形滤波器和方差为  $\sigma^2$  越 葬 越 葬 的成形滤波器输入信号 增 越,原系统如图 远- 圆 所示。按转变规则转变后的伴随系统如图 远- 猿 所示。图中  $\delta(t)$  为脉冲输入,作用点在原系统的输出点;  $\sigma$  为目标机动引起的脱靶量 增 越 的方差根,  $\sigma^2$  为方差,  $\sigma, \sigma^2, \dots$  只是仿真变量符号,无物理含义,  $\delta$  是伴随系统的脉冲响应 藻 (曾 圆)。

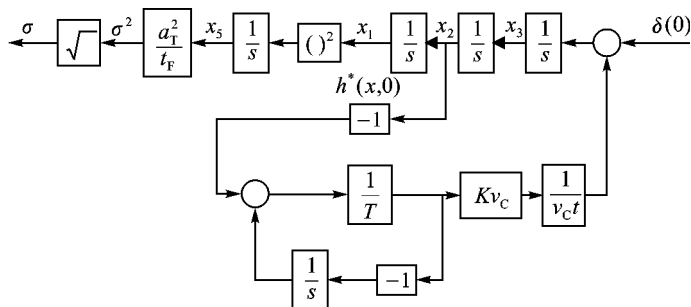


图 远- 猿 伴随随机控制系统

随机伴随系统仿真流程与确定伴随系统流程(图 远- 圆)类似。仿真系统参数与蒙特卡罗仿真法仿真举例的参数相同。

该随机伴随系统的仿真结果如图 远- 猿 所示。图中实线为伴随系统一次仿真得到的结果;十字所表示的 员 圆 个数据是蒙特卡罗仿真法 缘 次仿真得到的结果,十字数据是从图 远- 员 移植过来的。两种仿真方法得到的结果数值接近,而伴随系统仿真法的效率比蒙特卡罗法高得多。

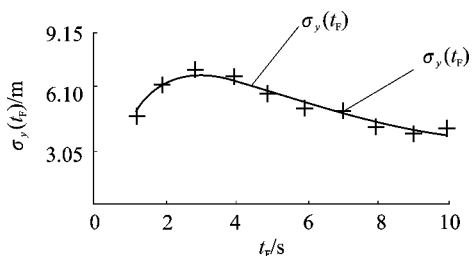


图 远- 猿 伴随随机系统仿真结果



## 习题

设计一种随机数产生方法,并编程实现(原)之间的随机数,计算该随机数的均值和方差。

将题得到的随机数转变成正态分布、方差为的随机数。

用蒙特卡罗仿真法完成图所示系统的仿真,画出流程图、编写程序,计算脱靶量的方差根。

用伴随系统仿真法完成图所示系统的仿真,画出流程图、编写程序,计算脱靶量的方差根。

## 参考文献

[姜玉宪]控制系统仿真[北京:北京航空航天大学出版社]

[郭尚来]随机控制[北京:清华大学出版社]

[熊光楞]控制系统仿真[北京:清华大学出版社]

# 第 7 章 实时仿真系统

半实物仿真系统、人在回路仿真系统、软件在回路仿真系统是三种常见的工程应用领域内的实时仿真系统。本章首先介绍实时仿真系统的基本概念、分类、组成与原理,接着重点讲述半实物仿真系统和人在回路仿真系统。随后,论述分布交互仿真系统的发展过程及分布交互仿真标准、协议及通信机制。最后,介绍分布交互仿真原型系统实例——航空兵攻防对抗仿真系统及基于网络的飞行仿真器。

## 7.1 实时仿真系统概述

实时仿真是指仿真模型的时间比例尺与真实系统的时间比例尺完全相同的仿真。当仿真系统有实物介入或有人在试验回路中时,由于实物和人是按真实时间变化和运动的,因此就需要进行实时仿真。实时仿真实要求仿真系统实时接收动态输入,并实时产生动态输出,输入和输出通常为具有固定采样时间间隔的数列。

### 7.1.1 实时仿真系统分类

目前,研究比较多的实时仿真系统一般都属于连续系统。下面介绍工程应用领域内实时仿真系统的分类方法。一般分为以下三类。

#### 7.1.1.1 硬件在回路仿真

硬件在回路仿真(硬件在回路仿真)又称半实物仿真。它是将系统的部分实物(如控制系统的测量传感器、控制计算机、伺服执行机构)接入回路进行的试验。这种仿真试验是将被仿真对象实体的动态特性通过建立数学模型,选择合适的算法形成仿真模型,在计算机上运行。此外还要求有相应的模拟生成传感器测量环境的各种物理效应设备,如图 7-1 所示。不同类型的传感器要求有不同类型的模拟生成环境,例如气压传感器要求有动静态压力模拟器(气压模拟装置),角度、角速度传感器要求有模拟转台等。

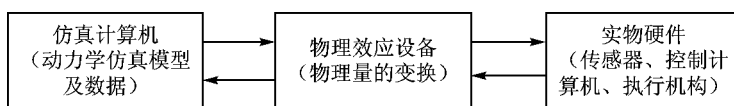
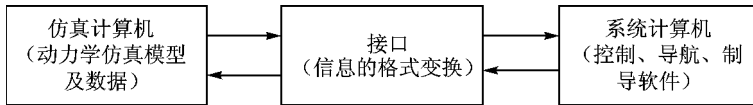


图 7-1 硬件在回路仿真系统



### 软件在回路仿真

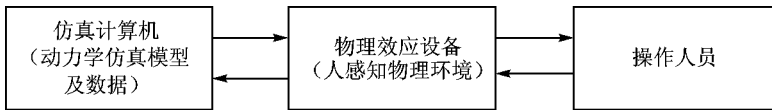
控制系统、导航系统和制导系统广泛采用数字计算机,通过软件完成控制、导航和制导,软件的规模越来越大,功能越来越强,许多设计思想和核心技术都反映在应用软件中,因此软件在系统中的测试越来越重要。这种仿真试验将装有专用软件的系统计算机与仿真计算机通过接口连接,进行系统试验,如图苑-圆所示。接口的作用在于将不同格式的数字信息进行转换。软件在回路仿真(深游)系统一般情况下要求实时运行。



图苑-圆 软件在回路仿真系统

### 人在回路仿真

人在回路仿真(导游)是训练人员、操作人员、飞行员或宇航员在系统回路中进行操纵的仿真试验。这种仿真试验将对象实体的动态特性建模在计算机上运行,此外还要求有模拟生成人的感觉环境的各种物理效应设备,包括视觉、听觉、触觉、动感等人能感觉的物理环境(模拟生成),如图苑-猿所示。由于有操作人员在回路中,人在回路仿真系统必须实时运行。



图苑-猿 人在回路仿真系统

### 实时仿真系统基本组成及特点

实时仿真系统比较复杂,对仿真计算机要求比较高,并有其独特的地方。通过苑-猿节对硬件在回路仿真系统、软件在回路仿真系统和人在回路仿真系统的概念介绍,可以看出实时仿真系统基本上由实时仿真计算机、物理效应设备和接口设备等三部分组成。

#### 实时仿真计算机

仿真计算机是仿真系统的核心部分,运行实体对象和仿真环境的数学模型与程序。一般来说,采用层次化、模块化的建模方法,将模块化程序划分为不同的速率块,在仿真计算机中按速率块实时调度运行。对于复杂的大型仿真系统,当仿真计算机的运行速度满足不了实时性要求时,可采用多台计算机联网实时运行。

对仿真计算机有多方面的要求:计算速度、精度、操作系统、内存及接口配置、经济性等。有些方面是相互制约的,需要在它们之间进行折中。由于要求实时性,所以对计算速度的要求



尤为突出。对计算机的速度要求高主要是由于系统中模型的频率响应要求高或者是建模的复杂程度高。在实时飞行仿真中,仿真计算机需要和外界交换信息的通道数可达数十至上百条,且要求高的输入输出速度(每通道为几百个数据字每秒)。在解算飞机动力学方程时,要求产生许多(几十个)双精度变量或更高精度的变量,需要很大的数据库,而且动力学方程有很宽的动态频率范围,如俯仰长周期运动模态的频率为 $0.05 \sim 0.1 \text{ rad/s}$ ,俯仰短周期模态为 $1 \sim 2 \text{ rad/s}$ ,控制面作动器(舵机)的模态频率可达 $10 \sim 20 \text{ rad/s}$ 或更高。因此,要求仿真计算机的运算速度必须高到足以处理最高频率的模态,同时还要保证足够的精度。

对于时间和精度要求都较苛刻的实时仿真,以前大多是采用专用仿真机进行半实物仿真,如国防科学技术大学研制的银河仿真 I 型机和 II 型机在这些方面都取得了卓越的成就,它们曾成功地进行了长二捆火箭的仿真。但是,一方面由于专用仿真机价格高,利用率低;另一方面,计算机的计算能力每年以极高的增长率更新,因此对主机运算速度的选择有了更大的余地。计算机速度要根据采样频率和被模拟对象的数学模型的复杂度来选用性价比高的计算机。

对于飞行模拟器来说,由于有人的参与,对计算机的性能提出了更高的要求。飞行模拟器中计算机的特点如下:

(1) 被控通道类型多,除常规接口外,还有连接各种综合仪表的数字接口;

(2) 采样速率高,且有多种采样速率,最高的为操纵负荷系统,其采样速率为 $1000 \sim 2000 \text{ Hz}$ ,系统的采样速率一般为 $100 \sim 200 \text{ Hz}$ ;

(3) 在仿真运行过程中,各模块之间数据通信量大,实时交互问题突出。

#### 物理效应设备

物理效应设备的作用是模拟复现真实世界的物理环境,形成仿真环境,也称为虚拟环境。物理效应设备实现的途径多种多样。其中一类是采用伺服控制回路,通过专用的高精度伺服控制回路和仿真模型生成相应的物理量,例如三轴转台复现飞行器的角运动,动静压模拟器复现大气压力的变化,人感系统仿真人的力感或动感,它们都通过伺服控制回路实现。另一类是在已存储好的数据库中搜索相应的模型数据,通过专用软件和设备转换为相应的物理量,例如地形地貌、三维实体等物理环境的复现。

#### 接口设备

仿真计算机输出的驱动信号经接口变换后驱动相应的物理效应设备,同时将操作人员或实物系统的控制输入信号馈入仿真计算机。对于人在回路仿真系统来说,接口设备是指连接仿真计算机与物理通道的接口板,如接口板、接口板、接口板等电路板和专用总线等及与其他计算机系统之间的数据传输通道。

实时仿真系统的原理如图 7-1 所示。

在仿真计算机中通过解算动力学系统和环境的数学模型,获得系统和环境的各种参数。

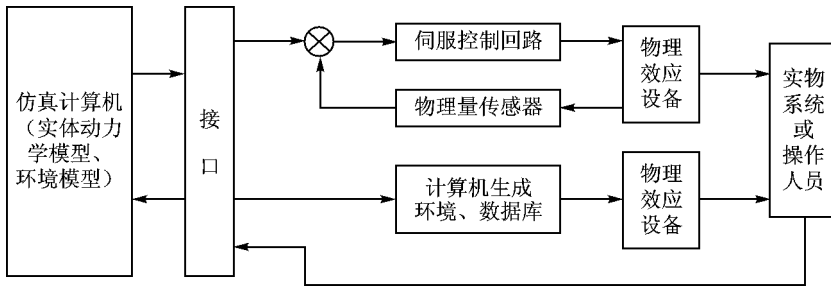


图 苑- 源 实时仿真系统原理框图

对于半实物仿真系统,这些参数通过物理效应设备生成传感器所需要的测量环境,对于人在回路仿真系统,这些参数通过物理效应设备生成操作人员所需要的感觉环境,从而构成完整的闭环仿真系统。物理效应设备是实现仿真系统所需要的中间环节,它的动态特性、静态特性和时间延迟都将对仿真系统的置信度和精度产生影响,一般都有严格的相应技术指标要求。

## 苑- 源 硬件在回路仿真——半实物仿真系统

### 苑- 源 半实物仿真概述

在某些复杂产品的研制中,为了考核某些关键设备的性能,必须将这些设备连接在仿真系统中,使仿真更接近真实情况,提高系统仿真的逼真度和仿真试验结果的置信度。这样的仿真就是硬件在回路仿真,也称为半实物仿真。半实物仿真是在飞机和导弹控制和制导系统研制过程中必有的仿真试验,如进入工程设计阶段后,系统的硬件和软件相继研制出来,就可以进行半实物仿真。由于回路中接入实物,仿真系统必须按实际时间比例工作,即必须实时运行。此外,在系统中还需要各种电信号与实际部件输入、输出量之间的转换设备。

在以下情况中常需要考虑半实物仿真。

(员) 系统中有些子系统的特性很难用数学语言描述或者表达式特别复杂,如寻的制导中导引头所处的电磁环境。

(圆) 对分系统进行验收和分系统模型校验,如弹上控制、制导计算机研制出来后,为检验计算机运行程序是否正确、性能及可靠性是否满足设计要求,要将计算机联入仿真回路进行多种工作方式的试验,使分系统(计算机)得到全面的考核。

(猿) 产品试验(如飞行实验)前的综合仿真实验。



## 图 7-1 半实物仿真主要物理效应设备

### 图 7-2 三轴转台

三轴转台用来模拟复现弹体或机体的空间角运动。

三轴转台由伺服系统控制的框架系统组成。框架系统有 3 个框架(分别为内、中、外框架)。内框架通过外框、中框和自身的转动,具有 3 个角运动的自由度。通过这 3 个框架的合成运动,使内框架复现弹(机)体在空间的角运动。在内框上安装真实的角运动传感器,可测出与实际飞行中相同的角运动。3 个框架的转角对应于欧拉角,即姿态角,可以有不同的顺序。一般将外框转角对应偏航角,中框对应俯仰角,内框对应滚转角。

仿真计算机可输出姿态角角速度和角加速度信号给三轴转台的伺服系统,使各框架复现相应的角速度和角加速度。这一般用于导弹的三轴转台。仿真计算机也可输出 3 个姿态角信号给三轴转台的伺服系统,使框架复现机体的姿态角(一般用于飞机)。图 7-2 所示为一个三轴转台伺服控制系统回路(框架)的工作原理图。

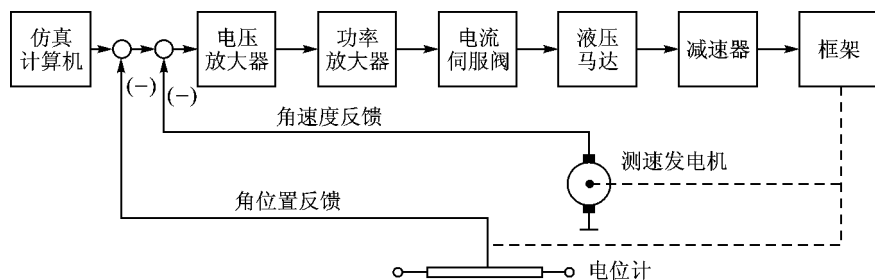


图 7-2 电液伺服系统工作原理图

在飞机飞行控制半实物仿真系统中,三轴转台模拟复现飞机姿态角运动,分别由仿真计算机中飞行动力学数学模型解算出的俯仰角、倾斜角、航向角等基本信号驱动。仿真实验时,飞行控制系统的水平姿态陀螺、航向陀螺、角速率陀螺安装在三轴转台上。

三轴转台的驱动元件有电动机和液压马达两大类,相应的驱动方式为电动驱动和液压驱动。电动驱动的优点是能源简单、结构简单、安全性好。液压驱动的优点是液压马达在与电动机相同体积和重量条件下能给出更大的功率和较小的惯性。如果提高油压,这些优点更为明显,所以液压三轴转台有很好的动态特性;但不利的是需要有液压源,有管路,如果采用高压油源,安全性差。

图 7-3 所示是北京航空航天大学研制的数字三轴电动飞行模拟转台。

### 图 7-3 目标模拟器

目标模拟器主要用于模拟生成目标仿真环境。

在进行导弹仿真时,有两种目标仿真环境:一种是电磁效应仿真系统,另一种是光学效应

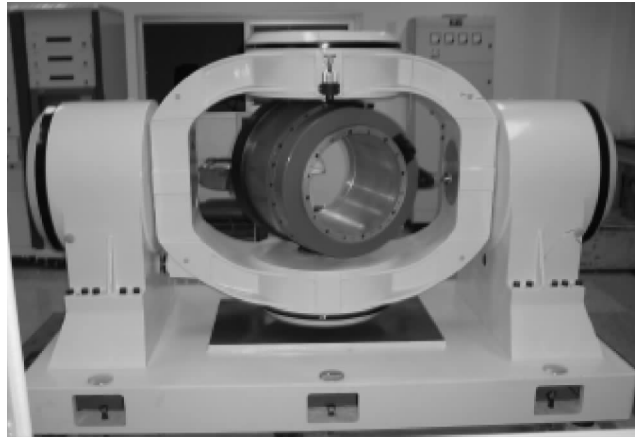


图 苑- 遥数字三轴电动飞行模拟转台

仿真系统。射频(电磁)仿真系统用于仿真雷达制导的导弹。光学(包括红外)仿真系统分别用于激光制导和红外制导的导弹。

这里主要介绍射频仿真系统。

射频仿真系统用于复现雷达目标的射频环境(包括目标、背景和干扰),通常由微波暗室、射频目标仿真器、仿真计算机及接口和校准系统等组成,如图 苑- 苑所示。微波暗室提供电磁波自由传播的空间,射频目标仿真器提供辐射目标及背景的射频信号;仿真计算机及接口用来实时控制射频信号产生器及射频目标仿真器、目标环境数据、监控显示设备;校准系统用于校准目标辐射单元的空间位置及射频参数。

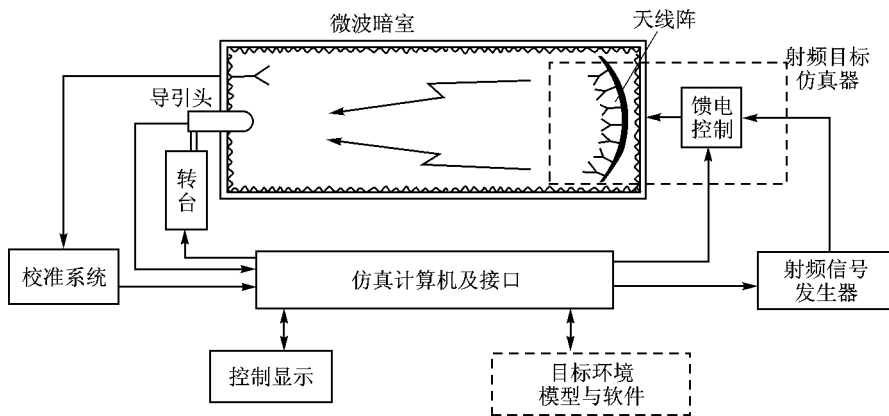


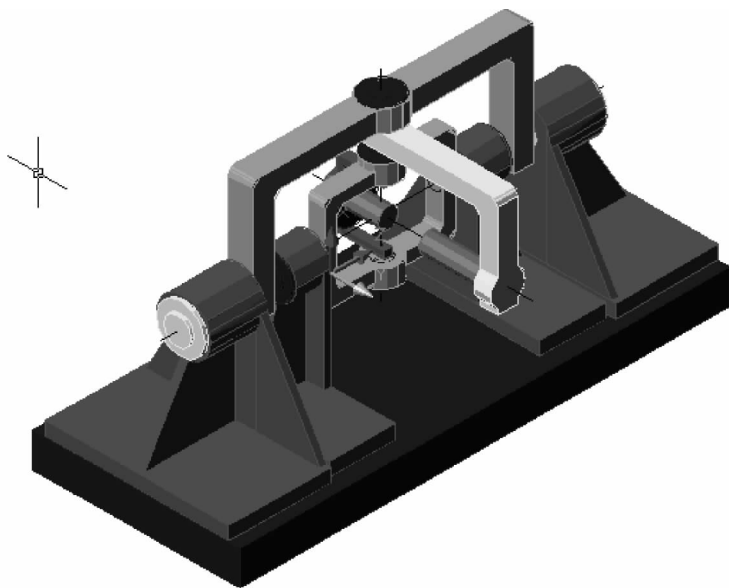
图 苑- 苑射频仿真系统结构图

微波暗室是一种加有屏蔽的射频工作室,室内四壁用吸波材料覆盖。吸波材料可使电磁



波反射减到最小,用此来模拟电磁波自由传播的空间环境,同时也屏蔽了外界的射频干扰。射频目标仿真器有两种结构形式。一种如图苑-苑所示的天线阵列式,采用许多辐射单元构成阵列。每个辐射单元信号的振幅、相位和极化等参数均由计算机控制,以达到仿真单目标或多目标的运动、杂波、角闪烁、起伏、多路径效应和干扰信号等。还有一种结构形式是由目标阵列和目标位置系统组成。目标阵列包含若干个喇叭形天线,它们被安装在目标位置系统上,并一起置于微波暗室中。仿真计算机计算出目标与导弹的相对运动关系,依此输出信号,通过控制目标位置系统,使目标阵列在空间作二维运动。与此同时,计算机输出信号控制射频信号产生器来仿真目标辐射的射频信号,将此信号输给目标阵列来仿真目标辐射。由于天线阵列式目标仿真器不需要目标位置系统,结构上较简单,所以得到较广泛的应用。

当前也有将安装导引头的三轴转台与安装目标仿真器的位置控制装置做成一体成为五轴转台,如图苑-愿所示。



图苑-愿 五轴转台

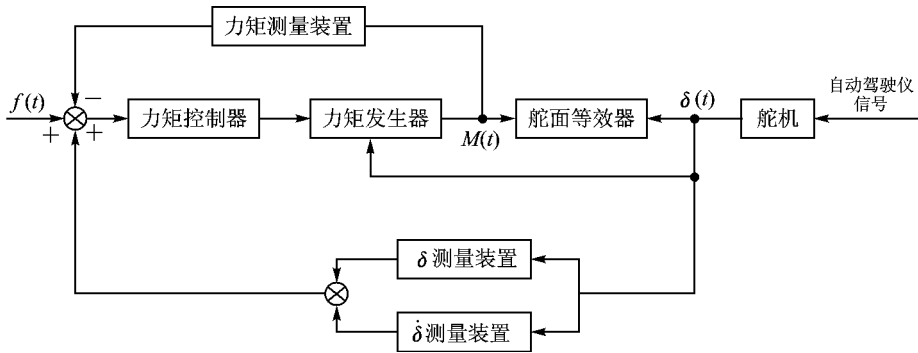
### 负载模拟器

导弹或飞机飞行中偏转舵面时,舵面将承受气动力矩,即操纵舵面偏转的舵机(控制面作动器)将承受负载力矩(铰链力矩)。负载力矩将影响舵机回路的特性,从而影响对导弹或飞机的控制性能。这种负载力矩还随飞行条件变化而变化。在某些情况下,负载力矩甚至会改变极性。因此,在半实物仿真系统中,负载模拟器占有重要地位。

最简单的负载模拟器由弹簧、阻尼器和舵面惯性等效体组成。但在仿真完整飞行航迹时,



这种简单结构的负载仿真器不能满足要求,因为它所产生的负载力矩无法随马赫数和高度的变化而变化。为仿真可变化的负载力矩,须采用力矩伺服系统。该系统的简化原理图如图苑-怨所示。



图苑-怨 负载仿真器方框图

负载仿真器的输入信号  $f(t)$  对应于所需的负载力矩,由仿真计算机给出。负载仿真器的输出是力矩  $M(t)$ 。力矩测量装置测出该力矩并反馈到输入端。 $\delta$  是舵面偏转角。 $\delta$  和  $\dot{\delta}$  的测量装置将输出相应的信号,也反馈到输入端,用以改善负载仿真器的性能。这样当真实舵机工作时,在其轴上将承受所需要的负载力矩  $M(t)$ 。

#### 源动 静压模拟器

动 静压模拟器是一种由伺服系统控制气体容器压力变化的装置,能产生满足一定精度要求的动、静压,作为飞行高度、速度传感器性能检验时的标准压力信号源。控制信号由仿真计算机中大气数学模型和飞行动力学数学模型解算获得,飞机上的动 静压模拟器直接与气体容器连接。

#### 苑 飞行器半实物仿真系统

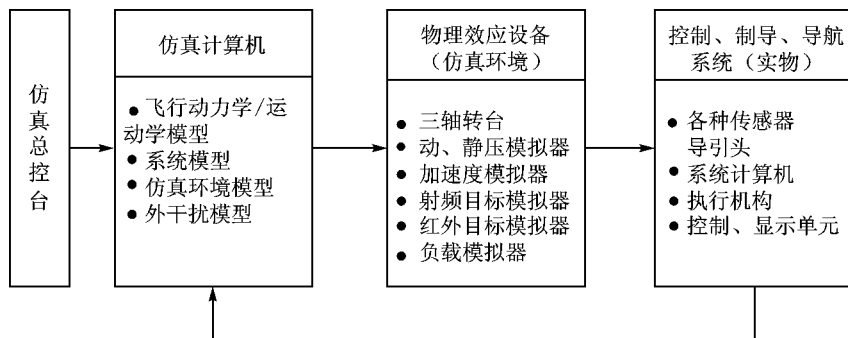
飞行器半实物仿真系统用于飞行器(歼击机、直升机、轰炸机、机载空-空导弹、空-地导弹以及民用飞机)的飞行性能和飞行控制、制导、导航等系统的研究。

飞行器半实物仿真系统的功能主要在于对飞行控制、制导、导航系统进行分析研究,包括控制规律、制导规律和导航方式的确定、参数的选择、系统的匹配等,同时对飞行器(飞机、导弹)的动态特性、飞行品质、制导与导航精度进行分析评估。

图苑-园所示是飞行器半实物仿真系统结构图。接入飞行控制、制导、导航等系统的传感器、计算机、执行机构等实物,仿真计算机解算飞行器飞行动力学数学模型、系统模型、仿真环境模型、外干扰模型等,仿真环境模型产生的驱动信号将控制各种物理效应设备,形成各种传感器测量用的虚拟环境。系统模型主要包括发动机系统、操纵系统、导航系统、自动飞行系统、



仪表系统等。仿真总控制台将控制整个仿真系统的运行,包括初始条件和参数的设置、运行过程中的监控和参数记录存储、运行后的试验结果评估和文档建立等。



图苑-员 飞机飞行器半实物仿真系统框图

飞机飞行控制系统的主要传感器有姿态陀螺仪、航向陀螺仪、动、静压传感器、过载传感器等,飞机飞行控制半实物仿真系统相应的物理效应设备有三轴转台、动、静压模拟器、加速度模拟器等。三轴转台模拟复现飞机姿态角运动,分别由仿真计算机中飞行动力学数学模型解算出的俯仰角、倾斜角、航向角等基本信号驱动。仿真实验时,飞行控制系统的水平姿态陀螺、航向陀螺、角速率陀螺安装在三轴转台上。动、静压模拟器是一种由伺服系统控制气体容器压力变化的装置,控制信号由仿真计算机中大气数学模型和飞行动力学数学模型解算获得,飞机上的动、静压模拟器直接与气体容器连接。稳定的恒速转台产生的离心加速度分量是实现加速度模拟的一种技术方案。仿真实验时,加速度传感器置于恒速转台上,其位置和测量轴线方位随加速度信号改变。飞行控制系统的执行机构操纵飞机舵面时将克服气动载荷形成的铰链力矩。它是飞行速度、飞行高度、舵偏角的函数。负荷模拟器将模拟生成铰链力矩加载于飞行控制系统的执行机构(舵机),是一种力伺服系统。

导弹控制、制导系统具有姿态传感器、加速度传感器、导引头等测量装置。导弹制导系统的制式有多种,主要区别在于导引头的结构和特性不同,按导引头接收的能量和信号的物理特性来分,有射频、红外、可见光图像、红外图像等。对于不同类型的导引头,进行半实物仿真实验时,必须有对应的射频目标模拟器和红外目标模拟器。

#### 苑-员 制导武器半实物仿真系统

制导武器半实物仿真是指在实验室条件下,将制导控制系统实物、实物工作环境模拟设备、目标模拟器或目标及环境特性模拟器以及制导武器运动学、动力学仿真模型通过接口设备连接起来,形成一个可运行的系统。

图苑-员是制导控制系统半实物仿真的一般原理框图。

制导控制设备物理环境的仿真是制导武器半实物仿真系统实现的关键技术之一。由于制

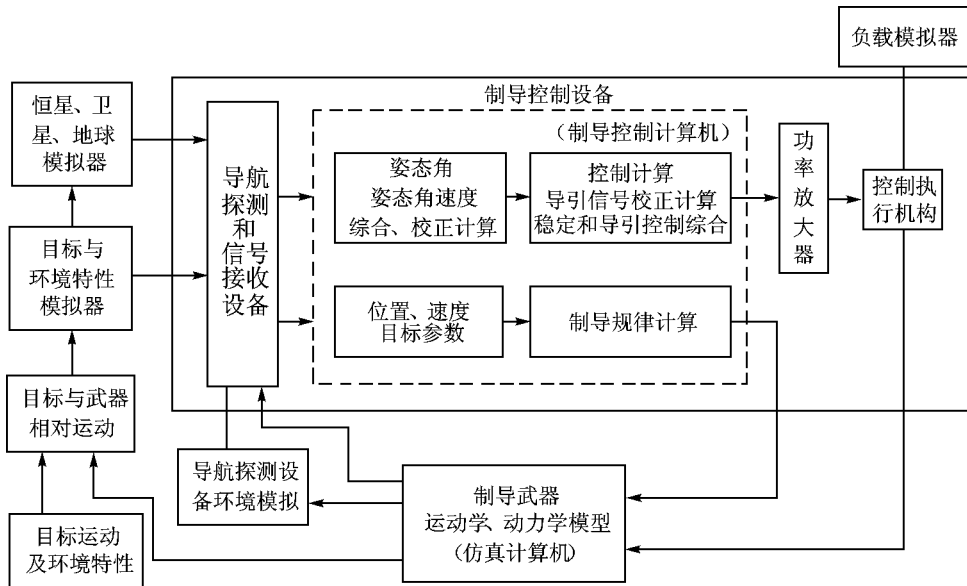


图 苑 远程制导武器制导控制系统半实物仿真框图

导控制设备的引入,必须生成它们正常工作要求的各种环境——光、声、电、运动和力学等物理效应环境。制导控制设备物理环境的仿真主要包括运动(惯性仪表)环境仿真、定位导引设备仿真和执行机构负载仿真。

#### 运动(惯性仪表)环境仿真

运动环境仿真主要是为惯性制导敏感元件提供仿真工作环境,即为测量弹体姿态角运动的惯性平台或陀螺仪提供角运动环境,为测量弹体姿态运动角速率的速率陀螺提供角速度环境,为测量弹体运动加速度的加速度表提供线加速度运动环境等。

仿真方法主要分为运动环境模拟方法和非运动环境模拟方法。

运动环境模拟方法直接为被模拟对象提供机械运动环境,主要设备是各种单轴、三轴及多轴角位置模拟转台、角速度模拟转台、线加速度仿真设备。

非运动环境模拟方法是基于惯性敏感元件工作原理提出的一种驱动它们工作的模拟方法。加速度表通过偏心质量敏感加速度产生的惯性力测量加速度,速率陀螺在感受旋转运动时产生陀螺力矩,引起陀螺转动,通过测量平衡时的转角测量角速度。

#### 定位导引设备仿真

定位导引设备仿真主要定位飞行器的速度和位置参数,实现制导武器发射点定位定向误差及惯性基准漂移的修正,达到提高命中精度的目的。

定位导引设备仿真方法包括恒星制导仿真、卫星制导仿真、射频目标和环境特性仿真、红外目标和环境特性仿真、地形匹配制导仿真、地图匹配制导仿真等。



### 猿猴执行机构负载仿真

执行机构负载仿真是指执行机构工作所面临的力学环境模拟技术。由于执行机构种类繁多,所以负载仿真设备也各式各样。主要负载有力、力矩、惯性、摩擦、弹性等,一个执行机构的负载通常是这些力的综合。

导弹等制导武器使用的执行机构主要有猿种:舵机控制机构、推力方向控制机构、侧向力控制机构。

舵机控制机构有液压舵机、电动舵机和气动舵机猿种。通过舵面在空气流中偏转产生控制力和控制力矩。

推力方向控制机构通过推动摇摆发动机喷管,直接改变推力方向,产生控制力矩。

侧向力控制机构有高压冷气喷射装置和小火箭发动机两种。通过控制气流的喷射产生控制力矩。

## 苑猿人在回路仿真

### 苑猿人在回路仿真概述

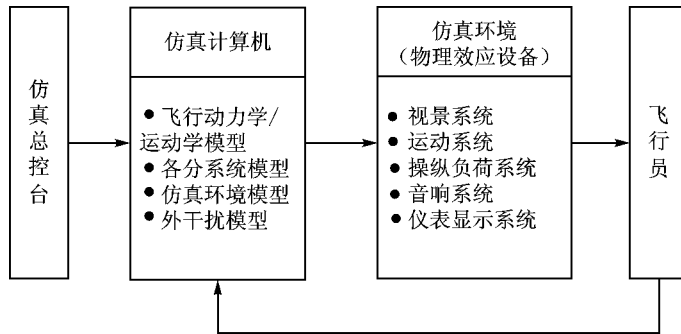
人在回路仿真是对有人参与操作的复杂系统进行仿真的一种重要形式,是在产品工程研制阶段用于对设计方案和分系统及部件性能进行测试、集成、实验。在使用阶段用于对操作人员进行训练,如对车辆驾驶员、飞行员、宇航员或者机场塔台、电厂控制台等重要设备操作人员的训练。与纯数学仿真不同的是,由于操作者的引入,这种仿真除了要将对象实体动态特性通过建立数学模型和编制程序在计算机上实现以外,还要求具备生成人的感知环境的各种物理效应设备,它们所产生的物理效应包括视觉、听觉、动感、力感等,是仿真系统性能评估的重要组成部分。

### 苑猿人在回路仿真系统——飞行模拟器

飞行模拟器是典型的人在回路仿真系统。飞行模拟器复现空中飞行环境,用于对飞行员进行起飞、爬升、着陆、转弯、机动飞行和飞机故障处置等训练,也可用于对飞机飞行性能、飞机操纵品质、机载系统性能进行分析研究。

飞行模拟器由仿真计算机、模拟座舱、运动系统、视景系统、音响系统、操纵负荷系统、仿真总控制台组成,如图苑-5圆所示。飞行模拟器飞行动力学数学模型、各分系统模型、仿真环境模型、外干扰模型等由仿真计算机(多台微机或多悦栽高档微机)解算,通过运动系统、视景系统、音响系统给飞行员提供多维感知信息仿真环境。飞行员对上述信息的感知,如同在空中操纵“飞机”飞行一样。

模拟座舱的布局与真实飞机座舱一样。模拟座舱内的仪表系统实时指示或显示各种飞  
员



图苑- 员圆 人在回路飞行实时仿真系统

行参数和系统参数。音响系统给飞行员提供各种音响效果,如发动机噪声、气流噪声。视景系统模拟座舱外的景象,包括机场、跑道、灯光、建筑物、田野、河流、道路、地形地貌、活动目标(空中和地面)等,同时可以模拟能见度、雾、雨、雪、闪电等气象条件以及白天、黄昏、夜间的不同时刻景象。操纵负荷系统给飞行员提供操纵载荷力的感觉,操纵载荷力随飞行速度、飞行高度、舵偏角等的变化而变化。运动系统给飞行员提供运动感觉,目前常采用的远自由度运动系统能提供瞬时过载,而离心机、抗荷服、过载座椅等设备可以提供持续过载。

### 苑圆 飞行模拟器结构、原理和仿真模型

#### 员圆 飞行模拟器结构和原理

飞行模拟器原理方块图如图苑- 员圆所示。

飞机飞行模拟器的主要功能包括:

- 评定飞机飞行动力学特性;
- 飞机飞行操纵性和稳定性研究;
- 飞行控制系统控制规律研究;
- 评定飞机操纵系统性能;
- 评定座舱布局和仪表显示;
- 评定航空综合电子系统的结构、硬件和软件;
- 机载武器系统的效能评定;
- 对空攻击和对地攻击功能及性能研究试验;
- 空战和武器群仿真;
- 飞行综合控制管理系统研究;
- 应急和故障条件下的操作特性研究;
- 飞行训练。

工程型飞行模拟器与训练型飞行模拟器的基本原理和组成是相同的,但在性能和使用要

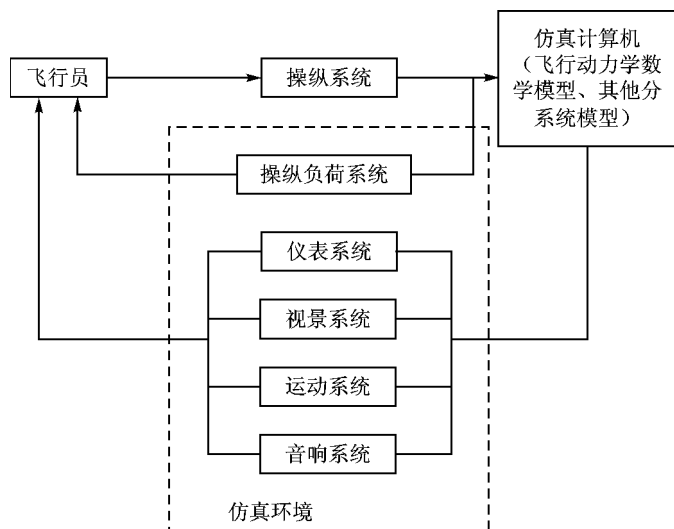


图 苑- 员 飞行模拟器原理框图

求上有所区别。工程型飞行模拟器应考虑通用性,易于进行参数的修改与调整,易于获得和评估仿真结果,对所研究的项目和内容应详细建立数学模型。而训练型飞行模拟器具体针对某一机型,座舱布局要求与真飞机的一模一样,其“飞行”特性着重要求对飞行员操纵输入的飞行参数输出响应与真飞机一样,即更着重在输入输出的外特性上一致。

#### 图 苑- 员 飞行模拟器仿真模型

飞行模拟器系统数学模型主要包括飞行动力学模型、飞机系统及机载系统仿真模型、环境仿真模型三类。

飞行动力学系统是组成飞行模拟器的一个主要系统,完成飞机空气动力学特性仿真、解算飞机的六自由度非线性全量方程、仿真飞机在地面上运动时起落架的力和力矩、模拟大气环境对飞行的影响。

飞机系统包括发动机系统、操纵系统、液压系统、燃油系统、电源系统等,机载系统包括导航系统、自动飞行系统、仪表系统及军机的火控系统等。这类模型具有各种控制律解算、逻辑控制,并直接与各种开关和显示器连接。

环境仿真模型建模时必须考虑与人体生理特性相适应,考虑人体感觉的最低门限和分辨率及对不同信号的敏感程度,包括运动系统、操纵负荷系统、视景系统、音响系统等。对于人在回路仿真系统,环境仿真模型主要是生成操作人员所需要的感知环境如动感、力感、视觉、听觉,构成完整的人-机闭环仿真系统。

由于飞行模拟器许多分系统的驱动信号都需要由飞行动力学系统提供,飞行动力学采用远自由度非线性全量运动方程,气动弹性影响在气动系数中计入,该模型具有高阶多变量非线性



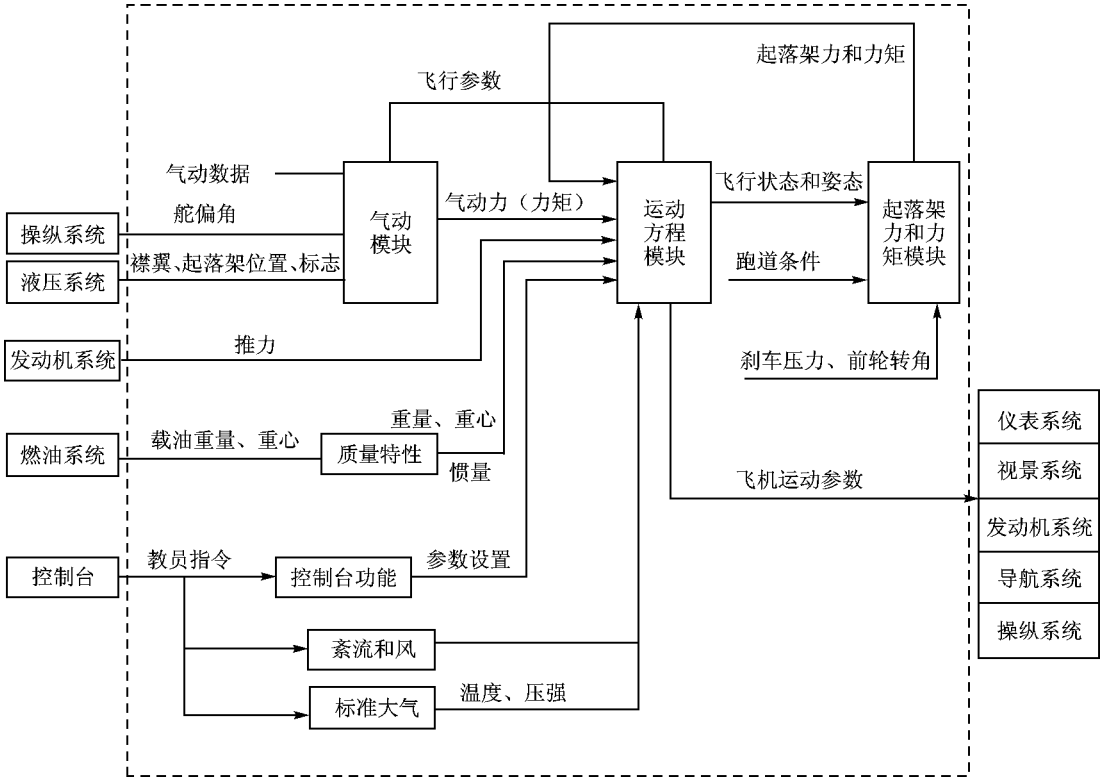


图 苑-员缘 飞行模拟器飞行动力学系统组成

度,然后积分求出飞机的合成速度。同时考虑风和紊流的影响以及坐标系之间的转换关系,计算出飞机姿态和飞机位置。

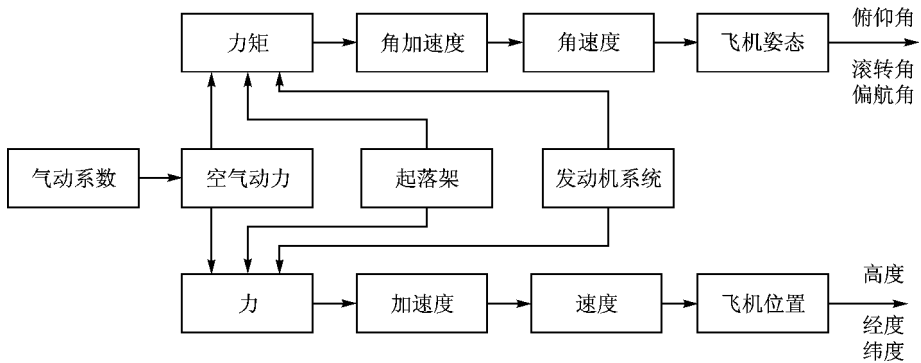


图 苑-员缘 飞行动力学仿真模型运动方程模块解算过程



起落架力和力矩模块计算受前轮操纵角、刹车输入和轮子速度影响的起落架力和力矩的影响。具体地说,该模块根据飞行系统其他模块提供的飞机速度、角速度、高度、高度变化率,飞机质量和气动力、力矩以及来自操纵系统的前轮偏角,来自液压系统的刹车压力以及由教员台设置的跑道条件及轮胎爆破标志等,计算起落架支柱冲击动态特性、刹车力、起落架总的纵向力、侧力及其力矩。根据各种跑道条件(干、湿、雪、结冰),模拟不同情况下的地面运动特性,还处理轮胎爆破故障的模拟。

飞行动力学仿真模型除了以上三部分外,还有用来计算飞机质量、重心和转动惯量的质量特性模块,提供标准大气和非标准大气的环境温度和压强的大气模块等辅助模块。

### 飞行模拟器环境仿真关键设备

训练型飞行模拟器要为飞行员或操纵者提供一个与实际系统相仿的环境。飞行模拟器主要环境仿真系统包括视景系统、运动系统、操纵人感系统、音响系统等。

#### 视景系统

视景系统是人在回路仿真系统中最重要环境仿真设备之一。在飞行模拟器中,视景系统为驾驶员提供一个实际飞行训练任务中的景象,因为视觉为驾驶员提供视觉的感知信息,因此视景内容的丰富程度、逼真度、清晰度和视场角的大小,都会影响飞行模拟器的质量和飞行训练效果。

20世纪70年代的飞行模拟器常采用闭路电视(电视)视景系统,典型的如我国自行研制的歼六飞机飞行模拟器视景系统,采用长焦距高倍率的地景模型,按1:1000的比例用沙盘制成,并用一台摄像机、一套专用光学投影系统及多套(典型的为3套)随动系统,使摄像镜头按飞机的飞行轨迹、航向和姿态,沿地景模型做上下、左右、前后移动,并通过光学系统把获取的图像投影在驾驶员座舱的前窗。但是需要有很大的建筑物来放置地景模型,并有复杂的机械随动装置,造成了使用维护困难、在低高度拍摄时景象模糊不清等缺点。而且此系统不能任意改变地景内容,然而,能方便的改变场景内容及活动目标类别,无论在民用和军用中都是非常必要的。因此,随着计算机图形图像技术的发展,20世纪80年代以后的视景系统绝大部分采用计算机实时生成图像(计算机)技术。

到20世纪90年代初,加拿大视景公司开发了视景系统,美国视景公司等开发了视景系统,加拿大视景公司也开发了视景系统,飞行员可看到更加逼真的座舱外景象。高档产品虽然满足全任务飞行模拟器的要求,但价格昂贵,只能根据用户的使用要求及投资情况来设计和选用计算机成像系统。目前采用通用高档微机(或微机图形工作站)、高性能图形卡及通用三维图像建模软件、图像显示驱动软件的计算机成像系统由于具有良好的性能价格比,在中、低档计算机成像的视景系统中得到广泛应用。

随着计算机技术、信息技术的高速发展,逐步发展起一种可以创建和体验虚拟世界的系统,即虚拟现实(虚拟现实)。虚拟现实的基本特征为沉浸、交互和构想,为了使用户



具有身临其境的沉浸感和交互作用的能力,具有逼真的实时高性能的立体图像处理能力的图像发生器成为虚拟环境产生器的关键技术。

传统的训练模拟器只注意解决显示分辨率和景象的细节以及细节等级等问题,仅提供静态的地形和少量的活动目标,不能很好地满足目前虚拟训练系统中参与者对环境及其对象的交互作用的要求。随着仿真技术、虚拟现实技术的发展,对视景显示和可视化提出了更高的要求。

在选用和设计视景系统时主要考虑以下因素:

#### 摇摇(员) 分辨率

分辨率(分辨率)定义为视景系统能提供给用户辨识目标细节的能力,用人眼能辨识目标细节的分角来表示。由于飞行员能辨识目标的分辨率为 $1'$ ,因此分辨率是飞行模拟器视景系统的最重要参数,特别在空战模拟器中,由于要求视景系统有高的分辨率,使视景系统的价格居高不下。

#### 摇摇(圆) 视场角

视场角(视场角)分为瞬时视场和总视场。瞬时视场指从一个固定的眼睛位置看到的图像范围,总视场是当眼睛沿所设计的眼点的全区域内移动所看到的图像范围。一般情况下,相对于飞行员视力中心的视场是飞机座舱各窗口的边框形成的。用于空战的军用飞机飞行模拟器及武装直升机模拟器视景系统都需要有很宽的视场角。例如攻击型武装直升机模拟器视景系统要求很大的视场角,垂直向下视场角为 $90^\circ$ ,向上至少为 $45^\circ$ ,水平总视场角至少为 $180^\circ$ 。

#### 摇摇(猿) 亮度

人类的视觉对外界激励的响应是随亮度(亮度)增加而增加,人眼和视觉的敏锐度也随亮度增加而增加,光的强度低可能使图像很难被分辨,背景光太强也会使分辨率下降。考虑视景系统时应根据实际要求,选择适当的亮度,以得到合适的分辨率。

#### 摇摇(源) 刷新率与更新率

刷新率(刷新率)是指显示器被重新寻址刷新的速率,更新率(更新率)是确定被显示图像新的位置而需要重新计算的速率。足够高的刷新率可以保证甚至较暗的目标在屏幕上移动也不至于闪烁。在动态图像显示时,还必须考虑更新率。当眼睛固定在某一位置,有一活动目标在某一背景中连续运动,若由于计算机运算和绘制图像的速度较低,将会看到目标相对于背景有跳动的感觉。因此要得到连续的图像,更新率至少为 $1/24$ 帧/秒左右。

#### 摇摇(缘) 色彩

色彩(色彩)可加强图像的易懂性和感知性,使用色彩特性作为提高计算机成像质量的一种方法,是很自然的。

#### 摇摇(远) 纹理

自然景象具有丰富的纹理(纹理)细节,人们的眼睛也紧密地依赖于纹理的暗示来感知



景象的构造和目标的材质,它是为驾驶人员提供速度、加速度、高度、地面倾斜度等感觉的视觉信息,还可帮助驾驶人员了解目标在周围环境中的相对大小及它们的相互关系,对飞行训练中的起飞、着陆、直升机的地形跟踪及贴地飞行等训练科目尤为重要。

### 摇摇(苑) 浓淡、阴影、透明、能见度

光对物体照射产生浓淡(泽昏昏)、阴影(泽昏昏)、透明(泽昏昏)、能见度(增昏昏)逼真效果。浓淡是光源照射在物体表面,使物体表面呈现出从明至暗或从暗至明的一种逼真度效应;阴影是当光源与被照明物体之间存在其他物体遮挡时,在被照明体上所形成的暗区;透明效应是指透明表面在视觉上引起在被其遮挡的表面上呈现出的色彩变化和几何畸变;能见度效应是由于大气层中的水蒸气和尘埃对物体遮挡程度随着观察点距物体距离的增大而增加,从而使远处物体的色彩远比它们本身的色彩灰暗和模糊。

### 摇摇(愿) 景物的细节等级

景物的细节等级(增昏昏)为每个物体或目标建立多个相似的模型,不同模型对物体的细节描述不同。根据物体在屏幕上所占区域的大小或视点离景物的远近以及感兴趣的凝视点等因素,自动为各目标物体选取不同的细节模型,平稳地加以切换过渡。

### 摇摇(怨) 感兴趣区

感兴趣区(增昏昏)技术为观察者在视线周围很小区域内提供高分辨率、细节丰富的景象,而在整个大视场范围内显示低分辨率的景象。

### 摇摇(员) 虚像或实像显示

由屏幕系统显示的实像缺少纵深感,高度感更差。经过光学准直的虚像光学系统显示的图像则可克服上述缺点,虚像或实像显示(增昏昏)能使模拟器的景象认为是光学无穷远。模拟器视景系统提供准直后的虚像或投影实像。

### 圆圆运动系统

运动系统是提供各种仿真器动感的物理效应设备。根据各类仿真器产生动感的原理,建立相应的数学模型和软件,在计算机中实时接收仿真器运动参数,生成特定的激励信号去驱动能产生动感的设备,从而使驾驶人员获得逼真的动感。

在飞行模拟器中能产生动感的仿真设备有以下几类。

#### (员) 平台式运动系统

该系统可分为猿自由度、源自由度、缘自由度和远自由度等平台式运动系统。远个自由度指升降、纵向、侧向猿个线运动及俯仰、偏航、倾斜猿个姿态角运动。平台式运动系统一般允许的角度变化范围在猿度以内,而线位移在员~员缘皂以内,因而只能在有限行程内提供飞机瞬时过载动感、重力分量持续感及部分抖动、冲击信号。

#### (圆) 抖振座椅

在飞行模拟器飞行员座椅上安装能产生与机体垂直轴平行的抖振信号的抖振器,抖振信号的频率和幅值可由计算机软件控制,信号源是飞机失速、刹车、着陆接地时产生的抖振信号,员愿



抖振幅值可达到 10g 左右,频率可在 0.1Hz 以内变化。

### (7) 抗荷服

抗荷服是用于给飞行员提供一个持续过载感觉的仿真装置,在飞行员身体上对持续过载最敏感的部位,如头部、胸部、背部,采用能施加压力的头盔、背心、腹带、护膝等构成抗荷服,所施加的压力与飞机的垂直加速度成正比,垂直加速度来自仿真器飞行动力学系统。仿真软件实时提供与垂直加速度信号成正比的压力信号。

各种动感仿真装置都只能仿真飞行中的部分动感,可根据仿真器的功能来选择或组合不同的动感仿真装置,从而构成满足任务需求的运动系统。

### 操纵人感系统

人在回路仿真系统一般都含有操纵装置,如飞行模拟器的驾驶杆、脚踏、控制台上的手柄、开关和按钮等。为了获得较真实的力反馈感觉,在飞行模拟器上还要使用专门设计的操纵负荷系统,即操纵人感系统。

飞行模拟器中的操纵负荷系统是指仿真飞机纵向驾驶杆力、横向驾驶杆力和脚踏力,并直接产生升降舵偏角、副翼偏角和方向舵偏角的主操纵系统。操纵负荷系统必须逼真地复现飞行员在驾驶飞机的各种条件下,在驾驶杆和脚踏上感受到的操纵力的变化及其动、静态特性。在全任务飞行模拟器及高档飞行训练器中,均采用数字式操纵负荷系统,利用高速微机(或高速处理器)和低摩擦、宽频带的液压伺服系统或电动伺服加载系统组成有人操纵的力闭环数字式操纵负荷系统。

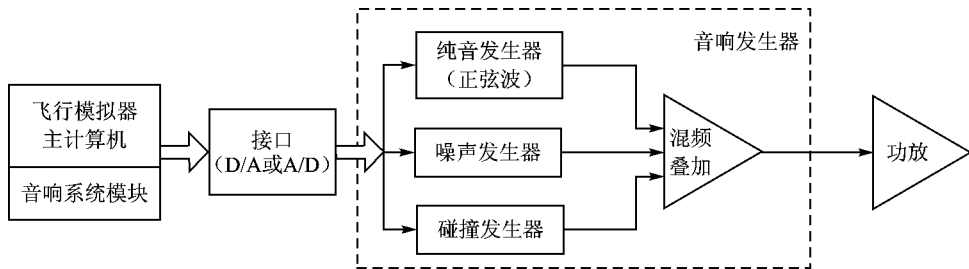
数字式操纵负荷系统的工作原理如下:首先建立飞机主操纵系统功能、操纵力模型、操纵面偏角计算模型,利用高速专用微机接口采集在相应操纵机构上测得的力、位移、位移速度等模拟信号,接收不同工作模式的选择信号,接收来自仿真主计算机的飞行参数和与操纵力有关的参数,再调度各个操纵通道的软件模块,计算出各通道的模型力,与实测力综合,驱动相应的执行机构,同时通过各通道舵偏角计算程序实时得到对应当前驾驶杆或脚踏位移和飞行条件下的舵偏角,发送到仿真主计算机系统。为了逼真复现操纵系统的非线性特性和提供平滑、稳定的力感信号,通常数字式操纵负荷系统的计算迭代速率应该高于 1000Hz,因此应采用高速处理器和高速、高分辨率的接口。

### 源音响系统

飞行模拟器在飞行仿真过程中,通过给飞行员提供逼真的音响效果,使飞行员判断出飞机的飞行状态。飞行模拟器音响系统是随飞机及各分系统工作状态的不同而发出不同类型不同频率和强度的音响。

音响系统要模拟的声音主要有两大类:一类是仿真对象自身发出的声音,如发动机轰鸣声、枪炮发射声等;另一类是环境噪声,如其他车辆的轰鸣声、炮弹的爆炸声、雷雨声等。

音响系统的工作原理如下:首先通过现场录音,离线分析各种声音的频谱,获得各种声音的频宽、幅值、变化规律等数据,建立相应的音响数据文件;然后,在飞行模拟器主计算机中建



图苑-员瑶音频系统工作原理图

立产生各种声音的数学模型和软件,在不同状态和条件下,实时计算不同类型和不同频率及幅值,并进行必要的修正,最后将不同波形声音的幅值和频率送到音响放大器中。音响系统工作原理如图苑-员瑶所示。现多采用数字式音响发生器,组成数字式音响系统。

## 苑瑶平台级分布交互仿真系统

分布交互仿真技术是20世纪80年代末、90年代初发展起来的新兴的仿真技术。平台级分布交互仿真是实时仿真的一种新的发展趋势,通过分布交互仿真技术创建一个逼真而复杂的虚拟环境,位于这个环境中的各种仿真实体可以互相交互。分布交互仿真技术不仅用于作战训练,也用于武器产品设计、生产、试验和维护的全生命周期,以及在作战环境下考验武器平台的性能。

平台级分布交互仿真系统是相对于聚合级分布交互仿真系统而言的。在平台级分布交互仿真系统中的实体具有较高的模型描述粒度,仿真时钟一般应是实时的,属于实时仿真系统范畴。聚合级分布交互仿真是指在仿真过程中,当一组实体具有共同的性质且无须对每个个体进行详尽的描述时,在仿真过程中就可以采用聚合形式描述,如一个飞行中队的飞机编队、一个步兵团、一个坦克营都可以作为一个聚合的实体进行描述,即作为一个单元进行描述。当聚合级仿真与其他类型的实时仿真系统进行交互时,聚合级仿真就要解聚为平台级实体,或者在有些情况下将一些聚合级的描述实时地进行仿真解算,来与其他系统进行交互。本章主要分析平台级分布交互仿真系统,以下简称分布交互仿真系统。

研究分布交互仿真技术,开发具有实用价值的分布交互仿真系统,在军事上是实现多武器平台在作战环境下进行体系对抗仿真的关键。建立一个综合的分布式虚拟环境可为工程技术人员提供对武器性能在作战环境下进行在线评估的手段;为指挥人员、作战人员提供一个验证作战想定,进行作战演练的环境;在工程上构造协同仿真环境,进行虚拟样机、虚拟制造、虚拟试验。国内外的发展已证明,采用分布交互仿真技术构造逼真的虚拟战场和虚拟武器平台,并在其中进行仿真演练已成为减少国防开支,保持军事优势,提供作战性能,降低战时损失的重要手段。



要手段。

## 2. 分布交互仿真标准、协议和通信机制

### 2.1 分布交互仿真的发展过程

分布交互仿真是在一个虚拟环境中实现仿真应用之间的交互,这些仿真应用位于采用兼容的体系结构、协议、标准和数据库的网络上的不同地点,相互之间进行交互并保持在时间和空间上的一致性。

自从1984年,美国国防部高级研究计划局(ARPA)和美国陆军共同制定的“综合计算机战(ICS)”研究计划启动以来,分布仿真技术走过了3个主要阶段。即以非标准化网络互联为基础的“综合计算机战”计划,走向标准化的分布交互仿真(IEEE 1516-1998)的研制和使用阶段,以及从1995年美国国防部发布具有多种时间管理机制,具有更好重用性和互操作性的高层体系结构(IEEE 1516-2000)的研制和使用阶段。

第一阶段,即“综合计算机战”的研制和使用阶段。“综合计算机战”计划的最初目的是试图将分散在各地的多个地面车辆(坦克、装甲车等)仿真器用计算机网络连接起来,进行各种复杂任务的训练,演示、验证实时联网的“人在回路”的作战仿真和作战演习。这项研究计划最终发展成了约100个地面车辆仿真器和飞机飞行模拟器以及通信网络、指挥中心和数据处理设备等互联的网络系统,分布于美国和德国的15个基地。该研究计划结束于20世纪80年代末。

第二阶段,即“网络”的研究、开发和使用阶段。在研究和使用的“综合计算机战”的基础上,美国军方提出了基于异构型网络互联标准的“网络”研究计划。“网络”的基础是协议数据单元(UDP)和传输控制协议(TCP),用来通过网络发送和接收数据。其中最常用的是实体状态(ES),它表示了一个仿真实体的所有状态信息。1987年12月,在美国召开了第一届“网络”研讨会,成立了工作小组,此后“网络”走上了标准化的轨道,于1995年正式提交了3个标准,并被IEEE接受,即IEEE 1516-1998、IEEE 1517-1998,分别为“网络”的ES标准、通信结构标准、演练控制和反馈标准。

第三阶段,1995年10月,美国国防部(ARPA)公布了建模与仿真主计划(M&S),提出了建模与仿真的远个目标,其中一个目标就是在“网络”范围内开发一个建模与仿真的公用技术框架。这个目标又包括3个子目标,在体系结构方面开发一种高层体系结构(IEEE 1516),其目的是为了支持仿真应用间的互操作性和仿真部件的重用性。1997年12月召开的第十五届“网络”研讨会成为最后一届讨论“网络”的会议,下一届会议开始讨论的内容是“网络”——采用IEEE 1516体系结构的分布交互仿真。如果说“网络”体系结构是从被动地迎合地理上分布的仿真器互联的要求的基础上发展并标准化的话,那么IEEE 1516的建立则完全是为了主动地建立一种更适于各种类型的模型和仿真系统以及“网络”系统之间实现互操作的体系结构。

随着互联网技术的发展,2000年前后提出了基于“网络”的仿真,并形成了可扩展建模与仿真的



真框架 基于云( 资源透明、信息共享、资源透明、信息共享)。近期又提出基于网络的仿真( 早期、中期、后期) ,解决网上各种资源的充分共享,提供充足的服务。

从以上分析可知, 与 在总的实现目标和一些设计原则上具有某些共同之处,但 是分布交互仿真的一种新的体系结构,与 原有的协议、标准相比变化很大。 不仅适用于实时仿真系统,也适用于非实时仿真系统,它有多种时间管理机制,可以同时支持构造仿真、虚拟仿真、实况仿真。由于目前在分布交互仿真领域对 的研究、应用已成为主流,本部分也将集中介绍 相关内容。

### 圆爰 匀 标准规范的构成

匀 主要由三部分组成: 匀 规则( 规则)、对象模型模版( 匀 模板)和接口规范( 匀 接口) ,如图 苑- 员所示。

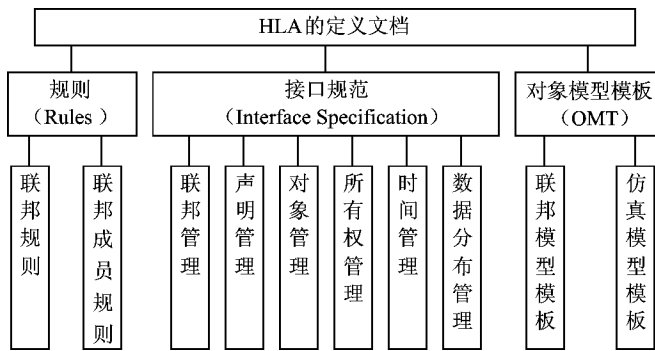


图 苑- 员 匀 的规范组成

匀 的标准规范由 个定义文档构成,这 个定义文档分别在图 苑- 员所示的位置发挥作用,从描述联邦和联邦成员行为,一直到描述运行环境、仿真管理。联邦开发与执行过程( 需求分析、需求管理、需求实现、需求验证)包括联邦需求定义、概念模型开发、联邦设计、联邦集成与测试,以及联邦的执行与分析。

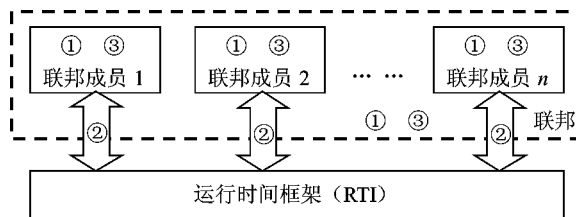


图 苑- 员 定义文档在 匀 系统中发挥作用的位置

①-规则;②-接口规范;③-对象模型模板

接口规范②在联邦成员与运行时间框架 之间发挥作用,而规则①和对象模型模板③



分别对联邦和联邦成员起约束作用。对象模型模板是描述  $\text{FOM}$  对象模型的一种格式。其中,采用  $\text{FOM}$  的形式描述联邦成员的对象模型称为仿真对象模型  $\text{FOM}$ ,描述联邦的对象模型称为联邦对象模型  $\text{FOM}$ 。

### (员) $\text{FOM}$ 规则

$\text{FOM}$  规则规定了  $\text{FOM}$  应用系统中联邦和联邦成员所遵循的原则。共有 4 条规则,其中 2 条针对联邦, 2 条针对联邦成员。

联邦遵循的规则:

- ① 每个联邦都应当具有一个使用对象模型模板  $\text{FOM}$  进行描述的联邦对象模型  $\text{FOM}$ ;
- ② 联邦中所有与仿真应用相关的对象实例的实现都由联邦成员完成,而不是  $\text{FOM}$ ;
- ③ 在联邦执行期间,联邦成员之间所有的数据交互都由  $\text{FOM}$  完成;
- ④ 在联邦执行期间,联邦成员和  $\text{FOM}$  之间的交互应当符合  $\text{FOM}$  接口规范;
- ⑤ 在联邦执行期间,在任意给定时刻,每个实例属性都只属于一个联邦成员。

联邦成员遵循的规则:

- ① 每个联邦成员都具有一个依据  $\text{FOM}$  进行描述的仿真对象模型  $\text{FOM}$ ;
- ② 联邦成员能够根据  $\text{FOM}$  中的定义,对对象属性进行更新和反射,对交互参数进行发送和接收;
- ③ 联邦成员能够根据  $\text{FOM}$  中的定义,在联邦执行期间动态地放弃或接收属性的所有权;
- ④ 联邦成员能够根据  $\text{FOM}$  中的定义,更改更新属性值所需要的条件;
- ⑤ 联邦成员负责维护本地时间,并保证与联邦中其他成员进行数据交换时,时间推进相互协调。

### (圆) $\text{FOM}$ 接口规范

$\text{FOM}$  接口规范定义了  $\text{FOM}$  与联邦成员之间的接口,也就是定义了  $\text{FOM}$  应当实现的功能,联邦成员所应当提供的回调函数,这些功能函数所需要的参数、条件以及功能调用的结果等。 $\text{FOM}$  接口规范定义的功能,构成了  $\text{FOM}$  系统的主体框架,是  $\text{FOM}$  和仿真应用程序设计所依据的主要标准。在  $\text{FOM}$  接口规范中将所有的管理划分为 远组服务和一组支撑服务。这 远组管理服务的功能如下:

- ① 联邦管理  $\text{FOM}$ 。定义了联邦执行的创建、撤销,联邦成员的加入、退出,联邦成员的同步等功能;
- ② 声明管理  $\text{FOM}$ 。定义了联邦成员发布、预订对象类属性和交互类等功能,以及相应的撤销等功能;
- ③ 对象管理  $\text{FOM}$ 。定义了对象实例的注册、发现、撤销,交互的发送和接收等功能;
- ④ 数据分布管理  $\text{FOM}$ 。定义了当存在区域匹配时的相关操作,包括区域注册、关联、注销,根据区域进行注册实例、预订数据类、发送交互等功能;



⑤ 所有权管理 定义了对象类属性所有权的相关操作。每个对象实例都有一个缺省属性。拥有某个对象缺省属性的联邦成员,也就拥有了这个对象的所有权;

⑥ 时间管理 定义了联邦执行中各联邦成员之间进行时间推进协调的功能。需要维护时间戳顺序事件和接收顺序事件两个队列。

一组支撑服务是用于对象类、交互类、属性和参数标志的获取等操作。

各类服务在联邦生命期上的活动期如图 苑- 所示。

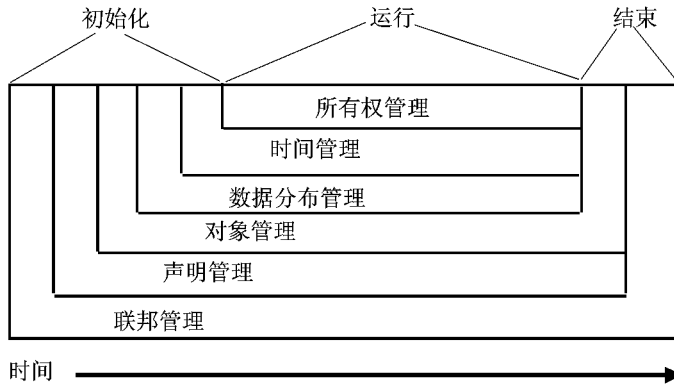


图 苑- 各类服务在联邦生命期上的活动期

### (猿) 对象模型模板

描述了有关数据类的概念,并以标准的表格方式进行记录,以提供数据、模型的重用。主要定义了以下表格:

- 对象模型标志表。对象模型模板的标志信息,如名称、版本号、目的等;
- 对象类结构表。记录联邦中对象类的名称、类型,描述类之间的继承关系,层次结构;
- 交互类结构表。记录联邦中交互类的名称、类型,描述类之间的继承关系,层次结构;
- 属性表。描述联邦中对象类所包含的属性,以及属性的类型、单位、精度、路由空间等;
- 参数表。描述联邦中交互类所包含的参数,以及参数的类型、单位、精度、路由空间等;
- 路由空间表。描述联邦中对象属性和交互所关联的路由空间;
- 字典。对模型模板的对象类结构表、交互类结构表、属性表、参数表中定义的类、属性、参数进行语义说明。

在系统中,同时还定义了系统中参与交互的数据之间的归类方式和相互关系准则,这些数据和关系准则是在运行期间所应维护的对象和进行管理的依据。



在匀控应用系统的开发过程中,需定义具体应用形式有两种,即云控和云控,也就是在开发匀控系统时,需要为联邦开发云控,为参与联邦执行的每个联邦成员开发云控。云控定义了整个联邦中将要使用的数据,包括交互数据、数据类型、路由空间。而云控定义了联邦成员所能提供的,用于交互的数据和数据类型。

在进行实际的分布交互仿真系统开发之前,这些定义文档只是表现为文本形式,在开发中,它们的表现不尽相同。规则是开发者在编程过程中始终要遵循的原则,也是在开发过程完成后检验该系统的一个标准。它体现在系统的整体和局部设计当中,但不体现在具体的语句里;接口规范则是一系列的服务原语。当要采用某种云控与联邦成员之间的服务时,就要用软件方式实现相应的服务原语的功能。也就是说,接口规范中定义的每一个服务功能都对应一个具体的子函数或功能模块,它具体体现在软件之中,云控是在文档中描述的,但它在实现过程中表现为软件中具体的对象描述。例如,对于飞机对象,可描述为一个飞机类定义和一系列的实例化。对于飞机上的武器开火,可描述为一个交互类定义和一系列的实例化。

#### 云控的接口规范分析

匀控应用系统采用客户/服务器模式,在运行过程中,由联邦成员向云控提供服务申请,并提供相关的数据和参数。云控响应联邦成员的请求,作出内部处理,然后通知所有受影响的其他联邦成员,并将必要的结果返回联邦成员。这个过程中联邦成员相当于客户,而云控起到了服务器的作用,处于核心位置。

匀控接口规范定义了云控与联邦成员之间的接口,即定义了云控应当实现的功能和联邦成员应当提供的回调函数,以及这些功能函数所需要的参数、条件等。匀控接口规范定义的功能,构成了匀控系统的主体框架,是云控设计所依据的主要标准。

在匀控接口规范中,对云控与联邦成员之间的接口函数所应当实现的功能进行了描述,并定义了粤控语言、粤控语言和悦垣垣语言的接口函数。在使用悦垣垣语言对匀控应用程序进行编程实现时,匀控接口规范中所定义的全部管理功能由两个悦垣垣类来实现:联邦成员代理(云控)和云控代理(云控)。类云控的成员函数包括云控能够提供的、由联邦成员调用的函数功能,比如创建联邦执行,加入联邦执行,更新属性值等。类云控中的成员函数为纯虚函数,在编程实现时必须派生出一个子类,在这个子类中实现所有的虚函数。这些虚函数是由云控调用、由联邦成员应用程序实现的功能,如发现对象实例,反射属性值,接收交互等。表苑-员列出了云控和云控所包含的部分管理功能。

图苑-圆显示了联邦成员与云控交互的过程。联邦成员中的各种对象用于完成不同的处理功能,当需要与云控进行互操作时,调用云控中的功能函数,并提供相应的数据。云控根据所调用的功能函数的不同,交给不同的云控对象作出处理(这里的云控对象有可能是本地的,也可能是异地的),然后调用云控中的函数,把处理结果返回给联邦成员。联邦成员根据返回的结果,调用相应的联邦成员对象进行处理。



表 苑- 员遥 砸 提供的编程接口(部分功能)

管理功能	砸 提供的服务	云 提供的服务
联邦管理	创建联邦,撤销联邦,加入联邦,退出联邦,注册联邦同步点	宣告同步点,联邦同步点注册成功失败
声明管理	发布对象类,发布交互类,订购对象类,订购交互类	开始注册对象实例,打开交互
对象管理	注册对象实例,删除对象实例,更新属性值,发送交互	发现对象实例,反射属性值,接收交互
所有权管理	属性所有权申请,属性所有权释放	请求释放属性所有权,获取属性所有权
数据分布管理	创建区域,删除区域,关联对象实例和区域,带区域订购对象类属性集,带区域订购交互类,带区域发送交互	(无)
时间管理	申请主控方式,申请受控方式,请求时间推进,请求下一事件	主控方式允许,受控方式允许,时间推进允许

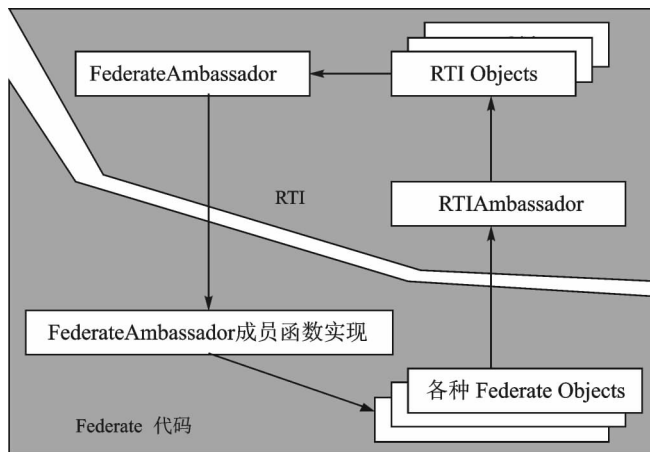


图 苑- 员遥 联邦成员与 砸 的接口

### 源 分布交互仿真的通信机制

无论采用何种体系结构,都不会改变分布交互仿真系统的分布特性和交互特性,都要通过网络通信得以体现和实现。

#### (员) 网络技术对分布交互仿真提供的支持

在 苑 世纪 苑 年代初,国际标准化组织( 陨 韵)建立了开放系统互联( 韵 韵 参考模型。与 员 员 员



对应参考模型相对应,分布交互仿真网络可分为两个大的层次,即通信层和应用层。图 7-10 表示了网络的两个层次、互连的两个层次与参考模型的各层之间的关系,其中通信协议定义了在网络节点之间发送数据块的原则。

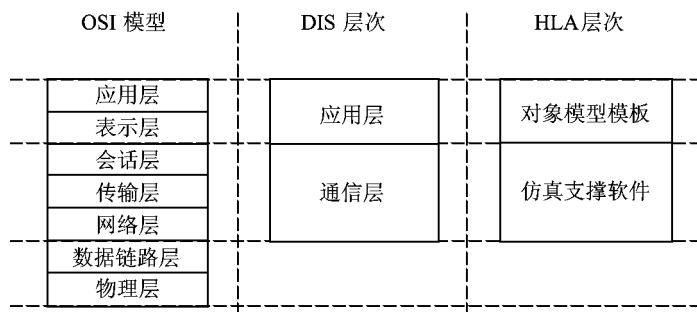


图 7-10 分布交互仿真的层次化及其与参考模型的关系

无论在哪一个层次上进行分布交互仿真系统通信程序的开发,都必须满足仿真应用程序的通信需求,为应用程序提供多种通信服务。这些通信服务包括实时或非实时通信、可靠或不完全可靠通信协议,以及单点传送、多点传送和广播等多种寻址方式。

#### (四) 决定通信服务方式的多种因素

分布交互仿真系统中的通信服务是实现整个系统中的所有仿真应用之间信息交互的方式和手段。由于系统中要传输各种类型的数据,并且这些数据具有不同的传输要求,因此,分布交互仿真系统中的通信服务也比较复杂。它包括了目前网络技术发展中的所有通信服务方式,特别是需要采用其中比较复杂的一些通信服务方式。

从理论上讲,最理想的情况是所有的数据都采用最快、最可靠、效率最高的方式来传送。但在实际的网络结构中,由于受到网络带宽、网络传输性能等因素的限制,不可能对所有的数据都实现这种理想的服务方式。特别是在网络规模较大、系统构成复杂的情况下,这种理想的情况更是不可能实现的。为此,要根据仿真应用的需求、对系统中的数据选择不同的通信方式。

在分布交互仿真系统中,影响通信服务方式的选择和实现的几个因素是:

##### ① 实时性需求

即通信服务所传输的数据是否要求实时传输。判断一个仿真系统中是否采用实时传输,什么情况下采用实时传输,主要依据以下两个方面:

##### ▶ 判断整个系统是否有实时通信需求

如果是有人参与操作或有实物参与仿真过程,或仿真过程产生的虚拟环境需要给人以真实感,就必须保证系统的实时运行,节点之间需要实时通信。而如果仅仅是为了验证一种作战策略的效果或检验一种武器的工程模型的性能,这时完全可以根据系统的性能适当地加快或



减慢仿真时间的推进速度,可以采用超实时或欠实时仿真,节点之间通信可以是非实时的。可见,通信的实时性需求是由仿真应用的性质决定的。

#### ► 对于实时系统,判断哪些数据需要实时传输

在判定一个系统必须实时运行后,为了更有效地利用网络资源,实际上并不是系统中的所有数据都必须采用实时方式传输。例如,对于表示仿真实体状态的数据,必须以实时的方式传送,否则就会导致仿真结果的错误;而对于那些用来管理仿真过程的数据,如使整个仿真演练停止运行的标志、某种时间管理模式等,则不必一定采用实时方式传送。

#### ② 可靠性需求

对于系统中的数据是否采用可靠的传输方式是由数据的性质决定的。例如,对于实体状态数据,它的偶尔丢失不会给仿真演练带来严重的影响。因此,不必一定采用可靠的传输方式;而对于那些用来表示实体间交互信息的数据,例如碰撞信息,则必须采用可靠方式传输,否则仿真演练就会出现错误。

在网络通信中,可靠或不完全可靠的传输方式实际上是指采用有连接或无连接的方式实现通信服务。有连接方式(采用载码率协议)指在通信之前在发送端和接收端之间事先建立逻辑连接,然后再进行通信。接收端收到信息后要向发送端发应答信息,如果发送端在一定的时间内没有收到应答信息,就要重发。这种方式能够保障信息正确地到达目的地,但耗费的时间较长。无连接方式(采用载码率协议)指在通信之前不建立任何逻辑连接,发出信息之后不再等待应答。如果接收端没有收到数据也不要求重发。这种方式的可靠性比有连接的方式差,但耗费的时间短。

可见,实时性和可靠性是互相矛盾又互相联系的。通信网络为了实现可靠性,必然会占用较多的时间;而为了实现实时性,也必然会影响到对可靠性的把握。这实际上是网络技术本身永远要面临的问题。

#### ③ 寻址方式需求

在一个仿真联邦中,所有实体都处在同一虚拟环境中,因此它们彼此之间有着大量信息交互。仿真过程中的任一实体在任一时刻与其他实体之间的交互情况大致可以归纳为三类。即该实体与另一实体交互,该实体与其他几个实体交互,该实体与其他所有实体交互。根据真实的情况,该实体与几个实体交互的情况是最普遍的。在分布交互仿真系统中,应当实现一个节点与网络中的任意几个节点之间的通信。

在实际的通信程序开发过程中,通常存在三种通信寻址方式,即单点传送(点对点寻址)、广播(点对点寻址)和组播(点对点寻址)。

将以上三个因素对分布交互仿真系统中的数据通信服务方式的影响总结如表 苑- 圆所示。

从结构上看,网际是基于计算机通信网络的面向仿真应用的一个分布式网络计算平台。它屏蔽了底层网络特性,为高层的仿真应用提供透明的网络传输。



表 苑- 圆 影响分布交互仿真系统中数据传输方式的各个因素

数据中包含的信息	实时性需求		可靠性需求		寻址方式需求		
	实播时	非实时	可靠 (裁否)	可靠 (裁否)	单点传送	多点传送	广播播
实体状态信息	√			√		√	
实体间的交互信息	√		√			√	
后勤服务信息		√		√	√	√	
仿真管理信息		√		√	√	√	
电磁发射信息	√			√		√	

数据库作为联邦运行的底层支撑系统,在逻辑上与联邦成员构成典型的客户-服务器结构,因此无论采用何种分布方式,以数据库为核心的分布式系统在逻辑上都是星型的拓扑结构。在设计实现时,由于数据库功能分布的不同,可以归纳为三大类:集中式、分布式和层次化模型结构。现用的较多是层次化结构。集中式数据库结构模型易于管理,却更容易产生瓶颈;分布式数据库结构模型解决了系统瓶颈,却缺乏集中式的全局操作管理。很自然的想法是将两者相结合来解决各自存在的问题,图 苑- 圆为基于这样考虑的层次式数据库结构模型示意图,是一种客户-服务器运行模式。在这里充当数据库服务器的是数据库,负责维护多个联邦信息;数据库作为数据库子服务器,是某个特定联邦的服务器,管理加入该联邦的所有联邦成员。数据库和数据库属于核心数据库组件(核心)。联邦成员(即客户端)可通过本地数据库组件,直接与数据库联系。

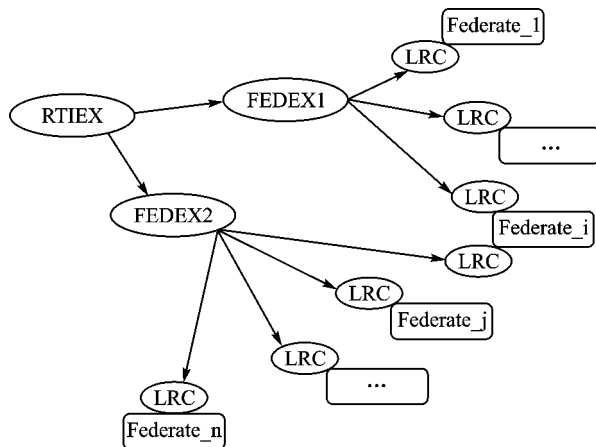


图 苑- 圆 层次式数据库客户-服务器结构





在这个阶段 根据联邦目标和对真实世界的抽象 ,开发联邦执行的想定 ,并对联邦概念模型和联邦需求进行分析。

联邦概念模型提供了与具体应用无关的抽象 ,为将联邦开发目标转化为联邦执行的功能和运行性能提供了方法 ,在联邦目标 and 设计实现之间提供了连接。

#### (獭) 设计联邦

此阶段用来鉴别、评估、选择联邦执行的参与者(即联邦成员) ,确定各个联邦成员的功能要求 ,并为联邦开发制定详细的开发和实现计划。

#### (源) 开发联邦

此阶段的目的是开发联邦对象模型(云豹) ,对联邦成员做必要的修改 ,为联邦的集成和测试做准备。其中又分为以下几步 :

- ① 开发 云豹 ,确定适合联邦目标的、联邦成员之间的交互数据。
- ② 建立联邦约定 ,有一些约定是在 云豹中无法体现的 ,因此需要另外制定文档进行记录。这些约定对于建立一致的、互操作的分布仿真环境是必不可少的。
- ③ 建立联邦成员 ,对联邦做必要的修改 ,确保联邦成员能够实现联邦概念模型中的功能 ,能够根据 云豹中定义的数据进行数据交互 ,功能遵守联邦约定。

#### (缘) 集成并测试联邦

此阶段的目的是规划联邦执行 ,建立联邦成员之间所需要的交互 ,在运行联邦之前对联邦进行测试 ,包括三步 :规划执行、集成联邦、测试联邦。

规划执行用来定义支持 匀貔联邦执行的所有信息。

集成联邦 ,将联邦中的所有联邦成员集成在一个联合的互操作环境中 ,这需要所有联邦成员的软件环境、硬件环境、交互约定能够相互协调。

测试联邦 ,用来确保所有联邦成员都能够完成各自的任务 ,相互之间能够完成互操作 ,并符合联邦目标 ,包括联邦成员测试、集成测试、联邦测试。

除了要具有能够与 砸鞑互联的接口以外 ,匀貔对联邦成员的具体开发过程提供了很大的余地。在 匀貔的体系结构下 ,联邦成员的开发可以很容易地面向不同的仿真应用领域。另外 ,即使是在同一个应用领域当中 ,匀貔也提供了对模型的重用机制 ,使同类仿真应用的开发过程变得更容易、更灵活。

#### (远) 执行联邦并得到结果

此阶段的目的是执行联邦 ,得到仿真输出数据和运行结果。

云貔是对建立 匀貔系统的一般性过程进行了总结 ,并非要求 匀貔应用系统的开发者严格按照 云貔的 远个步骤开发。开发可以根据具体情况作出合适的选择。并且 云貔各个阶段所投入的时间也是根据具体情况的不同而不同 ,各个阶段也可以由不同的开发小组和开发人员分别完成。



### 苑-圆-源 航空兵攻防对抗仿真系统和基于网络的飞行仿真器

前面几节详细介绍了分布交互仿真系统的标准、协议,在具体应用时还要根据不同的仿真要求进行相应的设计分析。本节介绍两个分布交互仿真原型系统实例,航空兵攻防对抗仿真系统和基于网络的飞行仿真器。

#### 苑-圆-源 航空兵攻防对抗分布仿真系统

##### (员) 航空兵攻防对抗分布仿真系统简介

航空兵攻防对抗分布仿真系统是一个采用 匀-源 标准的系统,如图 苑-圆-源 所示。分布仿真系统由 苑 个联邦成员组成,其中由计算机生成兵力技术实现蓝方对地攻击飞机联邦成员、蓝方对空攻击飞机联邦成员、红方对空攻击飞机联邦成员,每个联邦成员都具有 源 个 源架编组(源伊源共 员架飞机的仿真模型,可以根据需要按 员组)伊(源架)到 源组)伊(源架)的规模进行配置。人在回路仿真节点是一台有人驾驶的飞行模拟器,红方地-空导弹可以仿真 愿架发射车,每架发射车可发射 源发地-空导弹。二维态势显示用于实时显示整个战场的作战态势。系统中的 源 个工作站用于显示三维视景,以三维图像方式跟踪和显示飞机、导弹的作战过程。

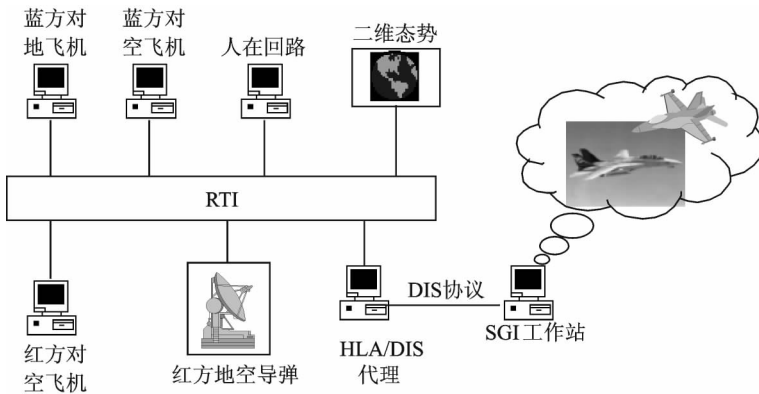


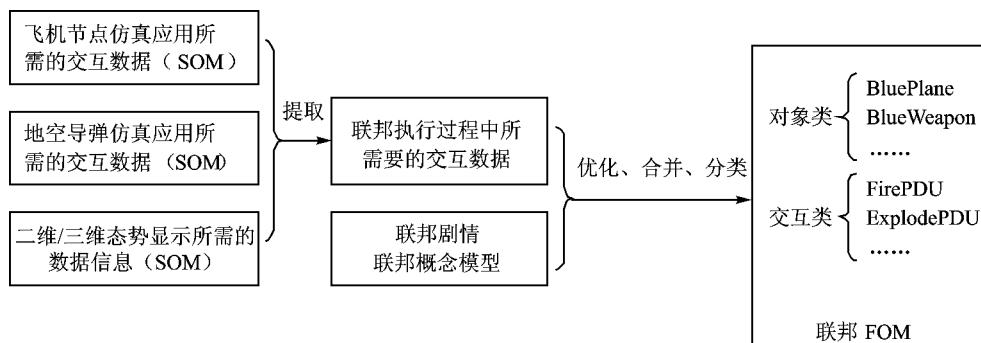
图 苑-圆-源 航空兵攻防对抗仿真系统结构图

#### 摇摇(圆) 航空兵攻防对抗仿真系统 苑-圆-源 开发

各联邦成员的 苑-圆-源 开发和仿真联邦系统的 苑-圆-源 开发是关键的一步。航空兵攻防对抗仿真系统 苑-圆-源 开发过程如图 苑-圆-源 所示。

开发工作首先应当分析各个联邦成员与其他联邦成员之间的交互数据,即每个联邦成员本身能够提供哪些数据,需要哪些数据,从中提取出飞机、导弹等联邦成员在联邦执行过程中所需要的交互数据,根据联邦剧情和联邦概念模型进行优化,合并分类等处理,从而得到基于 匀-源 的分布仿真系统 苑-圆-源 中的对象类和交互类以及它们的属性和参数。

由于对象类适合对连续信息进行建模,交互类适合对离散事件信息进行建模,因此在 苑-圆-源



图苑- 圆 摇摇和 猿猿 开发过程

猿猿 猿猿 开发过程中将仿真系统中的实体类型抽象为对象类,将开火、爆炸等离散事件信息抽象为交互类。联邦中联邦成员之间进行交互的数据类型,共抽象为 苑种对象类和 源种交互类,如图 苑- 圆所示。为了简化程序编制的复杂程度,所有对象类的属性都是相同的。

摇摇对象类 月猿猿 用于描述蓝方对地攻击飞机、蓝方对空攻击飞机的状态信息。月猿猿 用于描述蓝方武器实体信息。砸猿猿 用于描述红方飞机,砸猿猿 和 砸猿猿 分别用于描述基于 匀猿猿 的分布仿真系统和基于 阅猿猿 的分布仿真系统中的红方导弹信息,包括空-空弹、空-地弹、地-空弹等。砸猿猿 用于描述静态实体,由于蓝方目前还没有静态实体,因此主要用于描述红方静态实体或者地面固定目标,如机场、导弹发射车、雷达站等。砸猿猿 用于描述红方舰船。

对象类	属性
BluePlane	Position Velocity Posture Appearance Identify Description
BlueWeapon	同上
RedPlane	同上
RedMissileDIS	同上
RedMissileHLA	同上
StaticEntity	同上
RedShip	同上

交互类	参数
FirePDU	Identify DestId CarrierId Description
ExplodePDU	Identify DestId CarrierId ExplodeResult Description
StartUp	Command SimuTime
AcknowledgeMsg	Identify

图苑- 圆 摇摇应用系统中的交互数据定义

摇摇在对象类的属性中, 月猿猿 用于描述实体对象的位置(即坐标系中的坐标),由 猿猿 个量构成,坐标系采用地心坐标系或者拓扑坐标系。 猿猿 表示实体的运动速度,也是由 猿猿 个量构成。

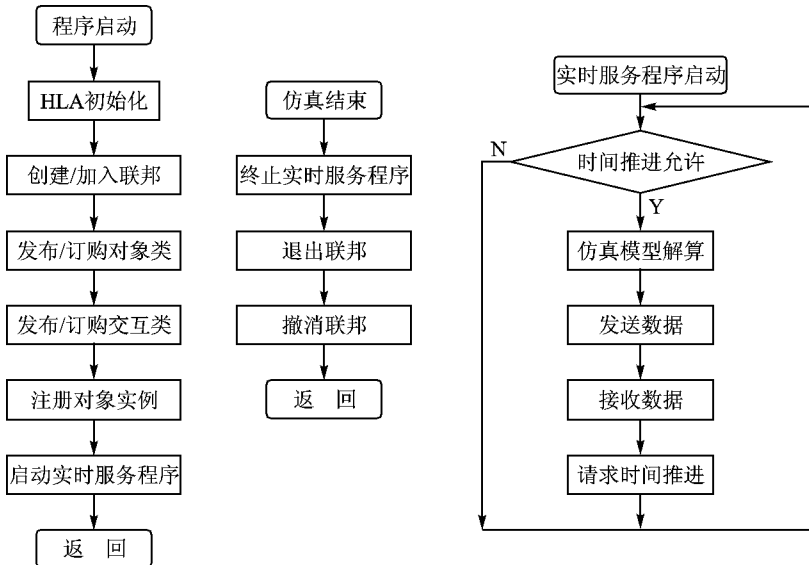


个分量构成。姿态类用于描述实体的姿态:俯仰角、滚转角和航向角。外观类用于描述实体的外观和毁伤状况。标志类是实体的标志,在整个仿真应用具有唯一性,应用程序可以根据标识确定具体的实体。其他类用于描述实体的其他信息,如国别、阵营等。

交互类有源种,云弹类用于描述开火交互,其中的参数云弹号是武器的标志号,云弹号指明武器打击的目标的标识,云弹号指明武器所属的武器平台,云弹号的作用与对象类的同名属性相同。爆炸类用于描述爆炸交互,其参数比云弹号多了项云弹号用于描述爆炸结果,如目标被击毁、导弹自爆等。杂项类用于传递管理交互,如系统的启动、暂停和结束等,其中的参数杂项号用于指定仿真系统的起始时间。联邦管理类用于联邦执行的管理,当联邦成员成功加入联邦执行后,向云弹号应用系统中的管理节点发送云弹号交互以示确认。

### (猿) 航空兵攻防对抗仿真系统软件开发

云弹号接口程序的处理流程如图苑- 圆所示。首先,仿真应用程序在启动时需要进行云弹号初始化,创建和加入联邦,然后发布或者订购对象类、交互类。不同联邦成员所发布和订购的内容各不相同,需要根据其云弹号来确定,例如,蓝方对地攻击飞机仿真联邦成员发布对象类云弹号属性和云弹号属性,订购与红方有关的所有对象类;红方地空导弹仿真联邦成员发布对象类云弹号属性和云弹号属性,订购云弹号的所有属性;二维态势联邦成员订购所有对象类和交互类,但是不发布对象类和交互类。然后云弹号初始化程序为联邦成员注册对象实例。不同联邦成员所注册的对象实例也是不同的,对于二维态势显示联邦成员



图苑- 圆 云弹号仿真系统接口程序



来说,在仿真运行过程中不注册对象实例。最后,仿真初始化程序启动实时服务程序并结束,仿真应用程序进入正常运行状态。

在仿真应用程序正常运行状态下,仿真接口程序的实时服务程序部分被启动并根据实时钟的触发周期性执行,其所完成的主要任务包括:仿真模型解算;同其他联邦成员进行数据交互,发送和接收数据;向模拟器请求时间推进。

当模拟器允许应用程序时间推进时,仿真软件进行仿真模型解算和其他仿真任务处理,将需要发送数据写入公共数据区,然后仿真接口程序调用模拟器提供的功能函数发送数据;当模拟器接收到其他联邦成员发送的数据时,调用仿真接口程序所提供的回调函数,进行解码并存储到公共数据区,再由仿真软件使用。通过仿真接口程序与模拟器之间函数回调函数的相互调用,实现了整个仿真系统中交互数据的收发。

当仿真应用程序结束时,仿真接口程序所做的处理包括:终止实时服务程序,退出联邦,结束联邦。

### 基于网络的飞行仿真器

航空兵攻防对抗仿真系统中的一个仿真成员就是基于网络的有人驾驶的飞行仿真器。本节详细介绍这种基于网络的飞行仿真器的组成与实现机理。

#### 基于网络的飞行仿真器组成

基于网络的飞行仿真器平台集开发运行为一体,在局域网内采用标准通信协议(载波调制)、非标准数据协议单元和点对点寻址方式。飞行仿真器自成子网,通过仿真网络管理节点的网络接口与主干网的网关或仿真两种通信协议互联,既可以独立构成一个功能节点,实现有人操纵飞行仿真器的空战模拟、对地攻击等作战仿真演练以及编队飞行、自动导航、自动引导等飞行仿真演练;又可以以联邦成员的身份通过仿真网关实时联网与计算机生成兵力实体形成对抗或编队训练,参与整个大系统的仿真演练。

基于网络的有人驾驶飞行仿真器主要由飞机实体模型部分(包括飞行动力学模型、发动机模型、电传操纵模型、飞控系统模型、火控系统模型、导航系统模型)、环境仿真部分(视景系统、平显、下显、数字仪表、音响)、管理部分(仿真管理、网络管理)、数据库等组成。系统结构如图 7-1 所示。

#### 开发环境特点

基于网络的飞行仿真器建立了一个基于网络连接,既便于各分系统独立开发,又便于集成交互的开发环境。其特点可归纳为以下几点:

##### ① 采用统一的数据结构

数据结构分为实体模型共享数据结构、各模型分系统本地共享数据结构和本地数据结构。通信数据分为包含共享与交互的总数据包、节点间传送的交互数据包以及飞行仿真器子网与主干网通信的数据包等三部分。各分系统交互数据均按规定的命名方式命名。

##### ② 采用统一的时间及线程管理

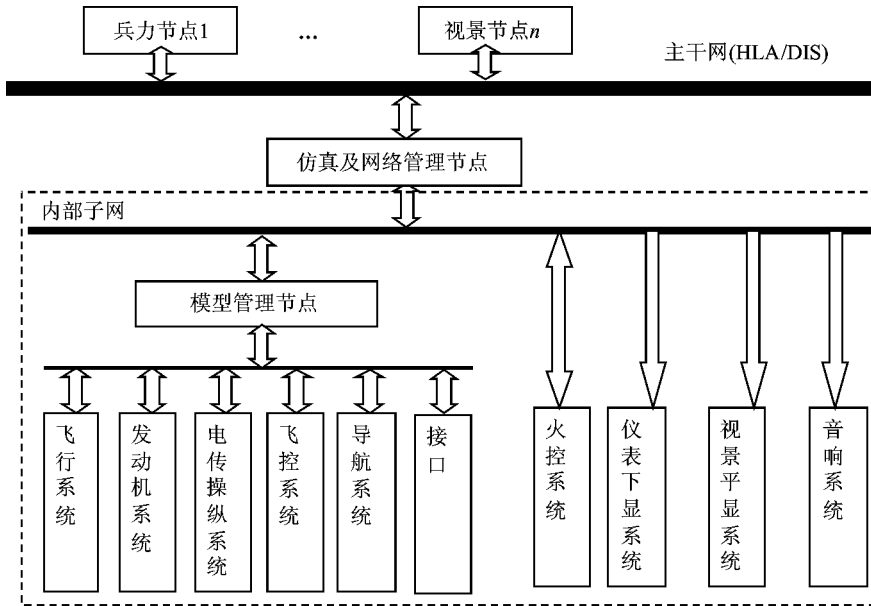


图 苑- 圆 基于网络的飞行仿真器组成

每个节点均由通信线程和仿真运行线程组成。节点内部仿真迭代周期与通信周期可以不同,也可以相同。如模型节点内考虑模型计算的精度可采用较短的迭代周期,而通信周期必须考虑到各节点的不同负荷,保证一个周期内所有节点数据均能更新,同时保证整体的实时性。各节点的时间管理采用统一的定时器,由仿真网络管理节点统一启动管理。

### 摇摇③ 各分系统或各节点分别建立自测试环境

各分系统或各节点分别建立自测试环境,建立用于设置输入工作模式及输入参数的人机界面,及运行结果记录和分析评估可视化环境。

### 摇摇(猿) 运行环境特点

该飞行作战仿真器的运行环境可归纳为以下猿种环境的协同运行。

#### 摇摇① 紧耦合实时局域子网运行环境

由紧耦合实时子网将各仿真节点连接成一个有机整体,保证以源皂皂周期更新数据。不仅满足飞机动态性能要求,同时满足人的感官对响应特性的要求,即从人输入操纵信号到获得来自视景仪表和音响的反馈信号的响应过程与人操纵真实飞机的过程基本相符。

#### 摇摇② 座舱实时操作环境

座舱具有主要操纵机构和机载系统。座舱内仪表采用在高分辨率圆皂皂显示器上显示的图形仪表。座舱内布局与真实飞机相似,在这样的操作环境下为操作人员提供了可以进行飞行与作战仿真演练的必要人机界面。



### 摇摇③ 低成本高逼真度的视、听仿真环境

要使飞行仿真器具有较逼真的飞行与作战环境,视景和音响系统是十分重要的,特别是视景系统。视景采用基于微机与图形加速卡的三通道视景系统,视场角、刷新率、分辨率、光源、纹理、具有逼真的地形、地貌、机场、武器阵地、目标机、本机及武器发射和爆炸的特殊效应。音响仿真采用基于微机和声卡的实时调度与播放系统,能提供不同飞行状态及武器发射爆炸的声音,起到提高环境仿真逼真度的渲染效果。视景系统与音响系统微机都连在实时子网内,保证了信息反馈的实时性。

#### (源) 飞行仿真器与其他仿真应用系统联合仿真

网络接口单元与其他仿真节点(如计算机生成兵力节点、视景等)通过标准协议、规范(如匀、匀、匀)集成为分布仿真系统

在分布作战环境中,飞行仿真器以一个具有确定、编号的实体出现。无论分布仿真系统采用、标准,还是、体系结构,飞行仿真器子网都可通过仿真及网络管理节点与主干网连接,保证子网内的自治性和子网管理节点与主干网的正确通信,实现网络接口单元的功能。具体包括:

- ① 坐标转换。将飞行仿真器采用地面坐标系所得到的实体状态参数转换至、或、结构下所要求的坐标表示方式。同时将接收到的主干网上的其他实体的状态参数作相应逆变换进入子网;
- ② 协议转换。将子网内采用的自定义的实体对象类和交互类参数的数据格式,转换为、的标准协议数据格式,或转换为、的对象类属性、交互类参数的形式;
- ③ 数据过滤。对来自主干网的各种信息,根据需要进行过滤,如实时选择对本机威胁最大的目标机,根据想定选择进行编队的友机等。

## 习摇摇题

摇摇简述实时仿真系统的组成与特点。

摇摇设计半实物仿真系统应关注哪些方面?

摇摇半实物仿真系统主要物理效应设备,如三轴转台、目标模拟器、负载模拟器、动、压模拟器的等各起什么作用?

摇摇人在回路仿真系统与半实物仿真系统的相同处与不同处是什么?

摇摇分布交互仿真系统的发展趋势是什么?

## 参参文文

[摇摇]吴重光援仿真技术[酝]援北京:化学工业出版社,圆园园圆

[圆圆]王行仁援飞行实时仿真系统及技术[酝]援北京:北京航空航天大学出版社,圆园园圆



- [猿] 黄柯棣. 系统仿真技术[M]. 长沙: 国防科技大学出版社, 1998.
- [源] 毕会娟. 分布交互仿真体系结构及通信机制研究[D]. 北京: 北京航空航天大学, 1998.
- [缘] 林新. 复杂分布仿真系统工程开发及交互操作性研究[D]. 北京: 北京航空航天大学, 1998.
- [远] 梁志军. 基于网络的分布式仿真系统性能分析与建模[J]. 计算机仿真, 2001, 28(1): 1-4.
- [苑] 梁志军. 基于网络的分布式仿真系统性能分析与建模[J]. 计算机仿真, 2001, 28(1): 1-4.
- [愿] 梁志军. 基于网络的分布式仿真系统性能分析与建模[J]. 计算机仿真, 2001, 28(1): 1-4.
- [怨] 梁志军. 基于网络的分布式仿真系统性能分析与建模[J]. 计算机仿真, 2001, 28(1): 1-4.
- [员] 彭晓源, 刘丽, 刘杰, 等. 分布交互仿真作战环境中有人驾驶飞行仿真平台[J]. 系统仿真学报, 1999, 11(1): 1-4.

# 第 1 章 虚拟现实技术及其在系统仿真中的应用

## 1.1 虚拟现实概述

### 1.1.1 虚拟现实(VR)技术的产生、定义、特点及系统组成

宇宙间客观事物繁杂多样,有的是自然形成、客观存在的,例如天体、地理环境、气象等;有的是人工创造的实体,例如飞行器、工业过程、电力系统、武器平台等。在自然环境中,人们可以根据环境条件和实体状态对人工创造的实体作出决策;也可对实体进行操纵、控制和管理,以达到规定的任务要求和技术指标要求。客观世界包含着自然环境、人造实体和人,以及它们之间的符合客观规律的相互影响、约束和交互作用的关系。

人类在不断的认识世界和改造世界时,迫切需要各种强有力的工具来增强、延伸、扩大自己的感官、肢体和大脑的功能。随着信息技术的迅速发展,特别是计算机技术、网络技术、图形图像技术、信息获取及处理技术的飞速发展,使得这种可以帮助人类认识世界、扩大感官能力的工具——虚拟现实技术得以发展和应用。这些扩大的感官功能可以达到:

(1) 根据研究问题的需要,提供一个具有尽可能逼真的虚拟对象、虚拟环境的虚拟世界;

(2) 使人沉浸在这个虚实结合的环境中,身临其境地体验这个虚拟世界中的一切,在这样的环境中研究和分析问题;

(3) 根据需要,人可以观察到在真实环境中无法观察到的宏观或微观的自然现象,如地壳的演变历程,亲眼领略火星探测车降落火星时的壮观场面,仔细观察细胞分裂时细微末节;

(4) 可以提供工具和手段使人能够与虚拟世界进行交流和进行交互作用,比如不仅能移动和触摸虚拟世界中的虚拟物体,还能在虚拟世界中漫游。

虚拟现实技术(AR)是在20世纪70年代被科学界和工程界所关注的技术。它为人机交互技术的发展开创了新的研究领域,为智能工程的应用提供了新的感知工具,为各类工程的大规模的数据可视化提供了新的描述方法。

### 1.1.2 虚拟现实技术的发展及特点

20世纪70年代初,伊凡·萨瑟兰教授在他的博士论文中对有关计算机图形交互系统方面作了论述。1973年他开发了头盔式立体显示器,他的论文及其实现的简单的虚拟世界展示了具有初始意义的虚拟现实技术,也是虚拟现实技术的萌芽。20世纪80年代初虚拟现实技术的概念和理论初步形成,80年代美国宇航局(NASA)及美国国防部组织了一系列有关虚拟



现实技术的研究,并取得了令人瞩目的研究成果,从而引起了人们对虚拟现实技术的广泛关注。1989年,美国海军研究中心虚拟行星探测实验室的斯科特·布鲁克斯和安东尼·瓦莱塞博士组织开发了用于火星探测的虚拟环境视觉显示器,将火星探测器发回的数据输入计算机,为地面研究人员构造了火星表面的三维虚拟环境。在随后的虚拟交互环境工作站(ARC)项目中,他们又开发了通用多传感个人仿真器和遥现设备。

“虚拟现实”一词是由美国ARC公司的创建人之一詹姆斯·巴特勒在1989年正式提出的,它反映了之前取得的一系列成就。进入20世纪90年代,迅速发展的计算机硬件技术与不断改进的计算机软件系统相匹配,人机交互系统的设计不断创新,使得虚拟现实理论得以进一步的完善和应用,如用虚拟现实技术设计波音787获得成功和1995年完成了对哈勃太空望远镜的仿真。1996年的ARC现在正致力于一个叫“虚拟行星探索(ARC)”的试验计划,这一项目能使“虚拟探索者(虚拟探索者)”利用虚拟环境来考察遥远的行星,第一个目标是火星。虚拟现实在娱乐、军事、航天、设计、生产制造、危险及恶劣环境下的遥控操作、教育与培训等方面得到广泛的应用。虚拟现实系统将成为一种对多维信息处理的强大系统,成为人进行思维和创造的助手及对人们已有的概念进行深化和获取新概念的有力工具。

### (一) 定义

虚拟现实系统是一种可以创建和体验虚拟世界(虚拟世界)的由计算机、传感器、控制器、显示设备构成的虚拟环境生成与显示的系统。虚拟世界是虚拟环境(虚拟环境)或给定仿真对象的集合,虚拟环境则是一种逼真的视、听、触觉一体化的计算机生成与显示环境,用户可以借助必要的装备以自然的方式与虚拟环境中的物体进行交互作用、相互影响,从而感知如视觉、听觉、触觉等信息,使参与者有身临其境的感觉,使参与者能体验、接受和认识反应客观世界和客观事物的虚拟环境或虚拟对象。

### (二) 虚拟现实系统的特征

虚拟现实系统的特征,可以用三个“I”表示,即“沉浸(Immersion)”、“交互(Interaction)”和“构想(Imagination)”,如图1-1中的“虚拟现实技术三角形”所示。

① 沉浸感是VR最主要的技术特征,指用户感到作为主角存在于虚拟环境中的真实程度。理想的仿真环境应该使用户难以分辨真假,使用户全身心地投入到计算机创建的三维虚拟环境中,该环境中的一切看上去是真实的,听上去是真实的,动起来是真实的,甚至闻起来、尝起来等一切感觉都是真实的,如同在现实世界中的感觉一样。

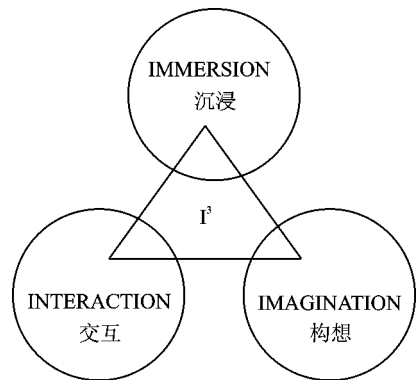


图1-1 虚拟现实系统的三个基本特征

② 交互性主要是指参与者利用视觉、听觉、触觉、嗅觉和味觉等感官功能,及对话、头部运



动、眼球运动、行走、转身、拾取和放置等人类自然行为,对虚拟环境中的实体进行考察、操作,并实现动态交互。例如,用户可以用手去直接抓取虚拟环境中虚拟的物体,这时手有握着东西的感觉,并可以感觉物体的质量,视野中被抓住的物体也能立刻随着手的移动而移动。

③ 构想性强调虚拟现实技术应具有广阔的可想像空间,可拓宽人类认知范围,不仅可再现真实存在的环境,也可以随意构想客观不存在的甚至是不可能发生的环境。依靠自己的感知和认知能力,全方位地获取知识,发挥主观能动性寻求解答,形成新的概念。

用“猿”说明虚拟现实系统的“猿”基本特征,也强调了人的主导作用。人可以从定性和定量综合集成的虚拟环境中得到对客观世界感性和理性的认识,通过在虚拟环境中对虚拟物体的认识和控制,为认识和改造客观世界提供了条件(经验、手段),从而激发创造性思维,进一步提高认识和改造客观世界的能力。

#### (猿) 虚拟现实技术特征

实现虚拟现实系统的虚拟现实技术应满足以下 4 个重要特征:

① 多感知性(身临其境、感同身受)。所谓多感知就是说除了一般计算机技术所具有的视觉感知之外,还有听觉感知、力觉感知、触觉感知、运动感知,甚至应该包括味觉感知、嗅觉感知等。理想的虚拟现实技术应该具有人所具有的一切感知功能。由于相关技术,特别是传感技术的限制,目前虚拟现实技术所具有的感知功能仅限于视觉、听觉、力觉、触觉和运动等几种,无论从感知范围还是从感知的精确程度都无法与人的真实感知相比拟。

② 存在感(身临其境),又称为临场感(身临其境),它是指用户作为主角感知虚拟环境所具有的身临其境的感觉。

③ 交互性(身临其境)。交互性是指在对虚拟环境中物体的可操作程度和从虚拟环境得到反馈信息的自然程度(包括实时性)。

④ 自主性(身临其境),是指虚拟环境中的物体能够依据物理定律自主运动的程度。例如,当受到力的推动时,物体会沿力的方向移动或翻倒等。

根据“身临其境”的概念及其上述 4 个特征,不难将“身临其境”与相关技术区分开来,例如数字仿真技术、计算机图形技术以及多媒体技术等。但是,目前在广义上认为仿真技术也包括对虚拟现实技术的研究及应用。“身临其境”技术是构造逼真仿真环境,特别是人在回路仿真环境不可缺少的手段。

#### (源) 虚拟现实系统的组成

一个典型的虚拟现实环境是由人和虚拟现实系统两部分组成,而虚拟现实系统又由三部分构成。它们是:

- ① 基于先进传感器的人机接口,如头盔式显示器、三维鼠标、跟踪器等;
- ② 具有多媒体功能的计算机系统;
- ③ 面向虚拟现实的软件系统,如各种虚拟场景、虚拟实体的建模、驱动软件、工具软件、语言(如“身临其境”等)。

具备典型虚拟现实特征的人机交互系统如图 8-1 所示,理想的虚拟现实系统应该包括:



- ① 立体图像显示设备,如头盔式显示器、~~交互~~立体眼镜;
- ② 立体声音响合成,为参与者显示虚拟世界的声音效果;
- ③ 语音合成,用文本生成自然语言的语音,可提供虚拟人体的语音;
- ④ 图形与图像生成,为参与者显示虚拟世界的视觉效果,为立体显示设备提供图形、图像;
- ⑤ 人体定位与跟踪设备,确定参与者头、眼、手和身体的位置与方向,提供人与~~交互~~系统实现交互的设备;
- ⑥ 人体(形体)语言识别,识别参与者手势、面部表情、身体姿态等形体信息;
- ⑦ 语音识别,识别参与者的语音命令或内容;
- ⑧ 触觉系统,提供力反馈信息;
- ⑨ 运动学系统,提供虚拟实体运动的法则和规律;
- ⑩ 虚拟环境生成器,以外部环境变化为输入,根据模型计算出虚拟(仿真)对象的状态和效果。

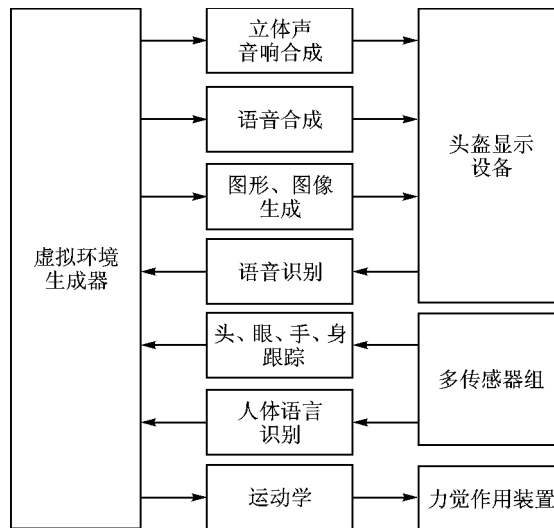


图 愿- 圆典型虚拟现实特征的人机交互系统

上述内容既包括了硬件设备又包含了仿真模型,同时还需要软件支撑环境。从系统组织上分类,虚拟现实系统可分为硬件、软件、界面和人性因素等源部分;从功能的角度分类,可分为显示与表现、生成与合成、感知与识别和人性因素等源部分。

### 虚拟现实系统形式

#### (员) 桌面型

桌面交互系统使用个人计算机或工作站实现仿真,计算机的屏幕作为参与者观察虚拟环境的一个窗口。可多人共享,用户约束感小,易于实现交互;但视角小,无环视感,一般要求用





选型、虚拟装配等) ,用户通过与视景、音响环境的交互取得如同与真实环境交互一样的体验。这类虚拟环境还可以是对某种抽象概念模型的直观显示 ,作为各种科学计算可视化技术的完善与扩展。例如 ,由计算机生成虚拟“风洞” ,研究者可以通过三维视景考察环绕飞行器的流线特征 控制流场的时空变化尺度等。

### 猿援灾砸关键技术

灾砸的关键技术包括 :

(员) 图形生成技术。

(圆) 立体声生成技术。

(猿) 人机交互技术 :人机接口-输入测量及输出显示装置的研究开发。

(源) 虚拟现实环境 实体外观建模及效果渲染技术 虚拟实体的动力学特性建模技术。

(缘) 环境建模 :要对环境进行几何建模 ,为其提供图形、图像来满足仿真的视觉效果。各种天气现象也是环境仿真的对象 ,目前在视景上对云、雾、雨、雪、日光等效果进行了实现 ,其模型进行了简化。如何反映它们在其他方面对实体的影响 ,并在实时运行中动态表示 ,是一个值得研究的问题。

(远) 人-机多维信息的研究及虚拟装置的开发。

(苑) 虚拟现实系统的集成、开发环境、总体技术 ,建立通用的技术框架。

(愿) 人工智能 :人工智能在虚拟现实系统中的应用主要表现在对智能实体的仿真。按照仿真应用的要求 ,对实体的建模与仿真是多方面的 ,其中就包括对人的智能的仿真。

### 猿援猿 人-机多维信息交互原理及显示方式

#### 猿援猿.1 人的感知系统

虚拟现实技术是一种将人类的感官功能、肢体、大脑功能进一步延伸 ,增强人类获取感知信息和处理感知信息能力的一门技术。因此 ,首先应了解人的信息感知系统和感知机理 ,进而建立相应的模型 ,去设计和开发虚拟现实系统中的各种辅助装置。一位名叫吉布森(猿援猿.1.1)的科学家 ,详细分析了人的感知能力 ,并建立了人的感知系统模型。他把人的感知系统划分为定向、听觉、触觉、视觉、味觉五部分 ,并分别列出了它们的行为方式、接受单元、器官解剖、器官行为、刺激源和外部信息 ,如表 猿-1 所列。

了解了人的感知机理、生理、心理和极限后 ,就可对 灾砸中各种能增强人的感知能力的各种辅助装置提出具体设计要求。

#### 猿援猿.2 人-机多维信息的类型、交互原理及显示方式

##### (员) 多维信息的类型

为人的多个感官提供感知信息的交互称为多维信息交互。人-机多维信息交互技术属虚拟现实技术的基础研究之一。参与者对计算机输入的自然与和谐,计算机输出感知信息显示

表 猿-1 人的感知系统



系系统	行为方式	接收单元	器官解剖	器官行为	刺激源	外部信息
定向	姿态和方向	机械和重力感受器	耳的前厅器官	人体平衡	引力和加速度	引力或加速度方向
听觉	听	机械式感受器	耳蜗	声音定向	空气中的振动	振动事件的性质和位置
触觉	触摸	机械、热和动觉感受器	皮肤、关节、肌肉和筋腱	多种探测,包括肢体、皮肤和舌头	组织变形,关节形态和肌肉纤维的拉伸	所接触物体表面和形状、物态固体性、粘性、热、冷
味觉	味嗅	化学和机械感受器,化学感受器	口腔(嘴)鼻腔(鼻子)	滋摇味气摇味	食物的化学性质蒸汽的化学性质	营养和生化价值,气味性质
视觉	看	光学感受器	视觉机制,包括眼睛和全身运动	调节瞳孔、调节注视、聚焦、扫描	光	景观、物体的大小、形状、距离、位置、颜色

的逼真是保证 系统具有沉浸感和融入感的关键。在一个用 技术创造的数字化的虚拟环境中,参与者输入与系统输出的多维信息有以下种类:

- ① 人对计算机的多维输入信息
  - 手及身体的位置、姿态信息;
  - 眼位及头的位置和姿态信息;
  - 语言及语音输入信息;
  - 其他信息(如人的手势、面部表情等)。
- ② 人的感官感知的计算机及虚拟现实设备输出信息
  - 实时的三维计算机图形、图像;
  - 实时的三维虚拟声;
  - 触觉、力觉反馈;
  - 参与者感受的动感;
  - 其他感知(如嗅觉等)。

人一机多维信息的交互过程是由参与者使用专用设备将人类的自然操作技能产生的信息(如手与头的位置和姿态、手的接触与施力等)传送到计算机(通常是多机联网系统),经过各类模型的解算(视觉模型、听觉模型、触觉力反馈模型等)得到虚拟空间中各类感知信息的显示(即感觉的输出信息),如逼真的图像、音响、动感、力感等。由此可见 系统中的多维输入输出信息与人的感官、大脑及计算机构成一个复杂的多通道人一机闭环系统,输入及响应的实时性必须满足人的响应特性及人的生理及心理需求。一般而言,从输入到输出的响应时间(包括延时时间),应在 以内,若时延加大则人感觉就会不协调,图像和声音会出现跳变。



人是通过两只眼睛观察世界的,通过眼球的透视变换,在左右眼的视网膜上分别获得两幅彼此一定相位差异的二维图像,大脑对这两幅图像进行合成处理,形成一个具有深度感的三维景象。通过计算机进行立体图像生成时,就是要分别生成左右两幅不同的图像,通过一定的设备分别呈现给两只眼睛。从获得信息的角度看,人的听觉是仅次于视觉的感知机能。听觉不仅能够听到声源的声音,辨别声音的类型,而且能感知声源的空间方位以及声源的运动。声音效果对视觉效果还有明显的提示作用。因此,虚拟环境中的声音效果仿真具有十分重要的实用价值。虚拟现实系统一个典型的特征就是参与者与虚拟环境的交互性。从信息的角度看,交互体现为信息的双向传递。参与者除了从虚拟环境中获得各种信息反馈(如视觉、听觉等)外,还要作用于虚拟环境。人的这种作用其中重要的表现形式之一就是人的头部、四肢、躯干的运动,这种运动必须以量化信息的形式输入到计算机中,才能够作用于虚拟环境,使之产生响应。运动测量就是借助于各种传感器实时地捕捉人体各部位的运动参数,获取人的运动信息,输入到计算机,作用于虚拟环境,实现人与虚拟环境的交互作用。通过头盔显示器、吊臂、数据手套、数据衣服直接从传感器得到虚拟环境中输入输出数据,可以获得上佳的临场感。利用动感增强感受,可以在虚拟环境中自由走动,做各种动作。典型虚拟现实系统的基本构成如图 1-1 所示。

#### (四) 多维信息生成机理及显示方式

① 视觉:视觉提供感知客观世界的信息量约占总信息量的 80%,视觉信息的逼真度是参与者在虚拟环境中具有沉浸感的关键。计算机图像生成技术是生成视觉信息的基础。场景及实体的三维建模及构成模型的大量数据(如环境、实体形状、颜色、质量、纹理等)的存储、提取与管理是创造三维虚拟空间的基础。为了解决图像逼真度与实时性的矛盾,在三维图像的几何绘制和剪裁、纹理映射、细节等级划分(LOD)、感兴趣区(ROI)的选取、特殊效应的生成等方面都要进行大量的复杂运算。为了节省机时,增强图像逼真度还采用一些新的处理方法,如“图像预存”、“适应性精炼”和“沉浸感等级划分”等。

图像生成所采用的计算机,目前有图形工作站,专用图形处理系统和高档微机加图形加速卡等几种方式。高档图形工作站生成的图像质量高,但价格太贵,不易普及推广。微机加图形加速卡生成的图像随着图形加速卡功能及性能的提高和软件开发平台的改进,所生成的图像质量几乎能接近高档图形工作站的水平。

视觉信息显示方式有常规的宽视场大屏幕显示、大屏幕监视器显示及头盔显示器(HMD)等。HMD是一个虚拟现实系统专用的视觉显示装置,由于显示器始终置于双眼前面,又可采用具有立体感的双目成像技术,视场可跟踪人头位的变化,因此人所观察到的三维景象有立体感、纵深感、视场宽的特性,能为参与者提供更好的沉浸感。

② 听觉:听觉感知的信息量约占总信息量的 10%,仅次于视觉信息量。虚拟环境中的三维虚拟声显示是增强虚拟环境逼真度的有效手段。要使虚拟环境中声音达到较高的逼真度,声音生成及显示系统应具备以下功能:

1. 声源

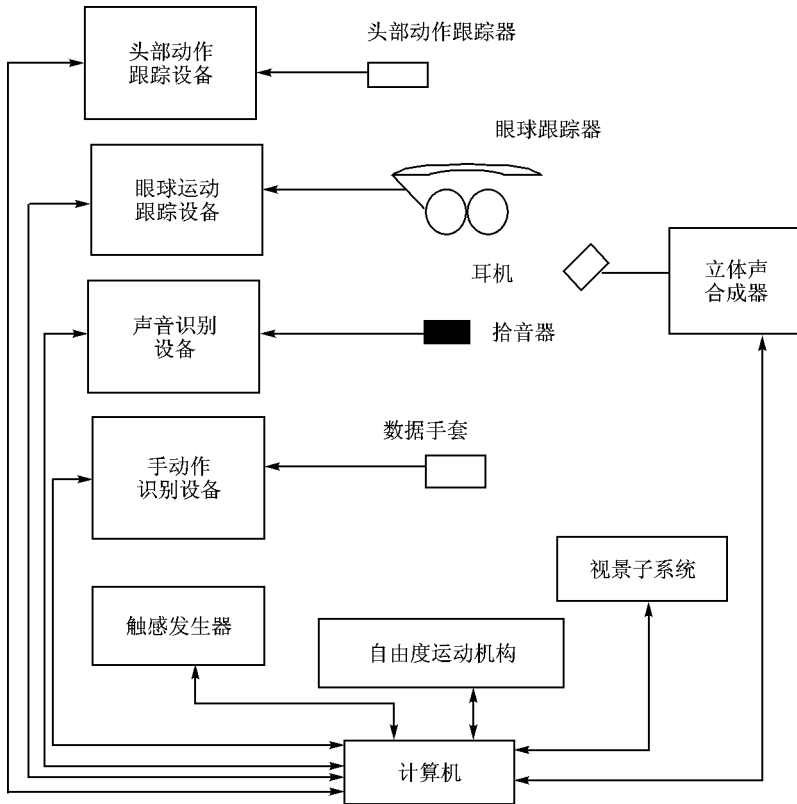


图 8-1 典型虚拟现实系统的基本构成

- 能生成较宽频率范围的声音,能反映虚拟环境中不同类型实体的各种运行状态;
- 能反应参与者与虚拟空间中声源的方位和距离的变化;
- 能同时生成多个静止的和运动的声源,及显示声源间相对运动产生的多普勒效应;
- 当用耳机播放声音时,能反应头部的滤波特性,使人对声音有正确的方位和距离感;
- 虚拟声能体现环境的反射,混响及材料的声学特性。

虚拟环境中声音生成的方式可分为两种,一种是声音建模、声源分析与合成;另一种是声源采集、动态调度、直接播放。第一种方法是把各种状态下的声源采集后,进行频谱分析,得出某种状态下声音的频谱、幅值、相位并存入一个声音文件中;然后建立产生各种声音的数学模型,即根据虚拟实体的运行状态,建立实时获取各种状态下声音文件中数据(频率、幅值、相位)的表达式,并采用插值方式获取当前条件下的数值,采用数字式或模拟式的方法进行各类声音的合成。用合成的声音信号去驱动扬声器或耳机。这种方法生成的声音具有良好的交互性,能根据输入状态,灵活、平滑地生成不同的声音;其缺点是声音有一定的失真,因为不可能



将声源的全部信息分析后,不丢失地合成,实际上合成的只是某种声源的主要信息。第二种方法是将声源在某种状态下的声音直接录制下来制成声音文件,存于声音文件数据库中,运行时根据当前状态直接调用某声音文件,采用循环或非循环的方式进行播放。其优点是声音逼真度高,但当状态变化后很难平滑地过渡到另一个声音状态,而且交互性差。

虚拟声的展示方式有基于自由场的扬声器阵列三维声展示方式和基于头部相关函数理论的双耳立体声耳机展示方式。这两种方式均可参与者提供具有方位感、距离感、相对运动感的三维立体声。

③ 触觉:触觉是人从客观世界获取信息的重要通道之一,人类的大脑功能有约1/4用于管理触觉信息。触觉包括触摸感觉、压觉、滑觉和力觉(即力反馈)。给虚拟环境中的参与者提供触觉反馈能增强虚拟环境的真实感,提高进行交互操作的成功率。目前触觉反馈设备还处于试验阶段,当前采用的装置有“气动式触摸反馈装置”,由双层多个小气袋组成触觉感手套。另一种是“振动式触摸反馈装置”,它利用声音线圈或状态记忆合金作为振动换能器,安装在皮肤表面不同位置,根据变形和弯曲度产生对虚拟物体表面光滑度的感知。还有“温度反馈装置”,采用使气泵迅速加热或冷却的方法,使人感知虚拟物体的温度。

“力反馈设备”能为参与者提供操纵虚拟物体或虚拟设备时感受到一种来自外部的力。力反馈产生的机理是:首先必须建立做某一种操作所能受到的力的数学模型,并获取计算模型力的参数(如虚拟物体的位移、速度、质量等)并计算出模型力。在参与者施加力的瞬间,既可获得计算出的模型力,又能用力传感器测出加力设备(电动的或液压的加力装置)所施加力的大小,并反馈回计算机与模型力比较,得到力的误差信号,将力的误差信号作为加力装置的输入信号。这样参与者感受到的力始终跟随参与者操作过程应感受到的模型力,并保持较高的精度。在虚拟环境中要去操作一个虚拟物体,首先必须实时检测出与人的手(或脚)等效的虚拟手(或足)和虚拟物体的“接触点”,一旦“接触点”激活,才开始计算模型力、实测力,使加力装置开始加力和跟踪加力的过程。“力反馈装置”的开发,也还处于研制阶段。目前有一种“仿形手臂”,它具有多个自由度和相应电动加力装置,并有多个自由度的腕部测力传感器。将仿形手臂固定在手上,力传感器与计算机输入连接,计算机输出信号驱动加力装置,这时真实手腕上会产生反馈力感。

在人-机交互中为了增加真实感,常采用具有两个自由度的真实操纵杆或游戏棒,参与者操纵真实杆,真实杆与由计算机力误差信号驱动的加载机构连接,可根据操纵杆的位移和虚拟实体的运行状态参数来实时计算出模型力和得到参与者手部感受的力。

④ 动感:在虚拟环境中,参与者可能位于不同的虚拟载体上(如飞机、坦克、汽车等),为了仿真所感觉到的动感信息必须研制专门的动感设备。这些设备由动感数学模型提供的各类动感暗示信号驱动,在相应的自由度上,产生移动、转动或抖动,使参与者感受到如同在真实环境中感觉到的动感。动感包括在线位移自由度上的过载(如失重),由于转动产生的重力作用感和抖动等。人体感受运动的敏感器官有眼睛及耳内的前庭平衡器官。眼睛通过观察虚拟物体



与虚拟环境景物的相对运动而感知运动,而耳内的前庭平衡器官则对运动位置的变化最为敏感。前庭器官内的半圆通道可测量(感受)线加速度(门限值为  $10^{-2}g$ ),前庭器官内的耳石可测量角加速度(门限值为  $10^{-2}g$ )。动感显示设备正是利用了对人体前庭器官的刺激,给人提供各种动感暗示。

提供动感的显示设备有:

- 平台运动系统——有 6 自由度和 3 自由度等多种,主要提供瞬时过载感和频率较低的抖振感和冲击感。
- 抖振座椅——主要提供抖动频率较高的抖振感觉。
- 抗荷服——抗荷服给参与者提供“持续的过载感觉”,通过给参与者身体对持续过载最敏感的部位上提供产生类似持续过载感的加压装置,来仿真由于载体保持持续加速度,使人感到的持续过载感觉。

动感信息的生成要建立专门的数学模型。它接收虚拟载体动力学模型实时解算出的参数,如线加速度、角加速度和某些特定运行状态等,解算各类动感模型,产生输出信号去驱动能产生移动、转动或抖动的专用仿真设备。

### 虚拟现实系统人机接口

在虚拟现实系统中,最基本的人机接口设备主要有头盔式显示器、输入设备与数据手套、跟踪装置和显示设备。

#### 头盔式显示器

在 VR 系统中,为了与虚拟环境交互操作,必须感知参与者的视线。也就是说,必须跟踪头部的位置和方向。为了在虚拟环境中移动物体,甚至移动参与者的身体,这就要求跟踪观察者各肢体的位置,而传感器技术是实现这一功能的关键硬件技术。有了头部跟踪功能,VR 系统使用者运动感受神经和他的视觉系统就能重新建立联系。当 VR 系统的使用者转动其头部时,其观察锥以及所显示的内容被更新。此功能增进了人的运动感受系统所感受的知觉在 VR 技术中的一致性和可靠性。头部跟踪还能增加双眼视差和运动视差,这些视差线索能改善用户的深度感知。

当用户头部转动时,头盔上的方位跟踪器,就向计算机输入头部的方位。若计算机的处理速度足够快、计算机输出的图像足够精细、显示器的分辨率足够高,则配戴头盔的人能随头部转动看到一个和头部方位一致的、由计算机输出的虚拟世界的图像,视场不受限制,这样就感觉完全沉浸在这个虚拟世界中了。

头盔式显示器(头盔)由哈佛大学的 麻省理工学院 在 1965 年首先提出,并设计出应用广泛的名为达摩克利斯之剑的头盔显示器。经过 30 年的发展,头盔显示器取得了巨大的进步和广泛的应用。特别是微型液晶显示器、虚拟现实(VR)、微型计算机、可视移动电话在现代数字化部队装备中的逐渐普及,使得头盔显示器在这些领域中占据了重要的地位。



头盔一般由图像信息显示源、图像成像的光学系统、定位传感系统、电路控制及连接系统、头盔及配重装置几部分组成,如图 愿-源所示。图像显示信息源是指图像信息显示器件,一般采用微型高分辨率液晶或硅基等平板显示器件。近年来平板显示器件的发展,特别是用于视频显示的高分辨率平板显示器件,对减轻头盔的质量和功耗、降低工作电压方面起到了促进作用。目前已有百万像素、千万像素或更高分辨率微型硅基制作的头盔显示器。在头盔显示器件中,观察者是通过眼前的目镜看清图像的,因此在头盔中,光学系统的设计十分重要,它影响着图像的显示质量。头盔可以根据需要,设计成全投入式或半投入式。全投入式将显示器件的图像经放大、畸变等像差校正以及中继等光学系统,在观察者眼前呈放大的虚像;半投入式是将经过校正放大的虚像投射到观察者眼前的半反射半透明的光学玻璃上,这样显示的图像就叠加在透过玻璃的外界环境图像之上,观察者可以同时得到显示的信息和外部环境的信息。

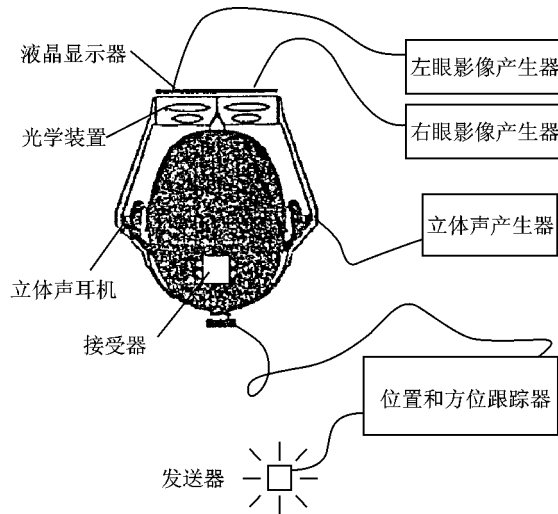


图 愿-源 头盔式显示器组成

由于头盔显示器所用的屏幕离眼睛很近,因此,为了使眼睛聚焦于如此近的距离而不易产生疲劳,需要使用专门的镜片,还要求这种镜片能放大屏幕图像以尽可能填满人的视线。这种透镜在 1983 年首次推出时被称为硅基透镜。硅基透镜的特征是使用了使得输出图像极其宽阔的透镜。

头盔跟踪技术所用传感器主要有磁式的、声波的、电磁的及光感的,每种技术各有优点,同时又有各自的局限性。

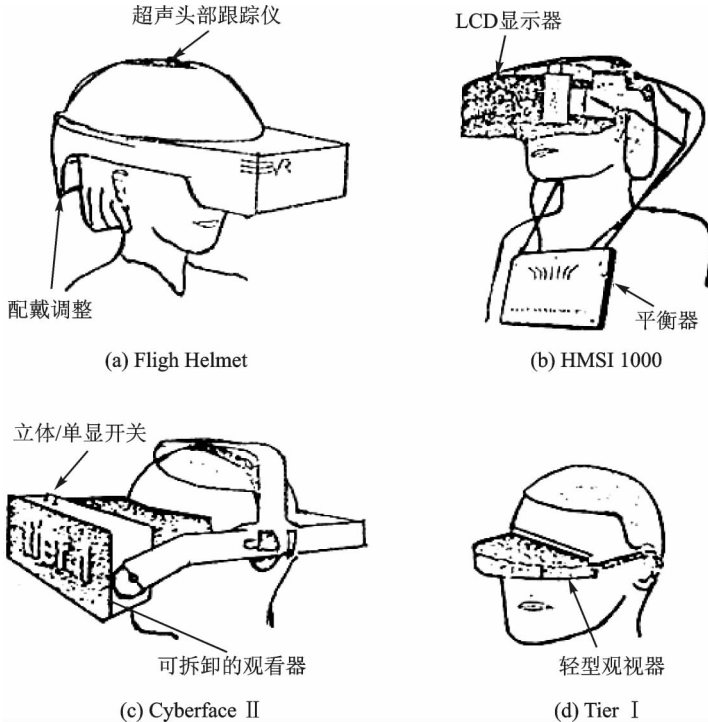
头盔显示器的基本标参数有显示模式、显示视野(FOV)、视野双目重叠、显示分辨率、眼到虚拟图像的距离、眼到目镜距离、物面距离、目标域半径(最大)、视轴间夹角、瞳孔间距



(焦距)、出射光瞳(孔径光瞳)、图像像差、视觉扭曲校正、质量、视频输出等。

头盔式三维立体显示器需在以下几方面进行改进,例如质量过大(负载~圆噪)、分辨率低(图像品质差)、延迟大(刷新频率慢)、行动不便(有线)、跟踪精度低、视场不够宽、眼睛容易疲劳等,因此有必要开发新的三维显示技术。

常见头盔显示器如图愿-缘所示。



图愿-缘 常见头盔显示器

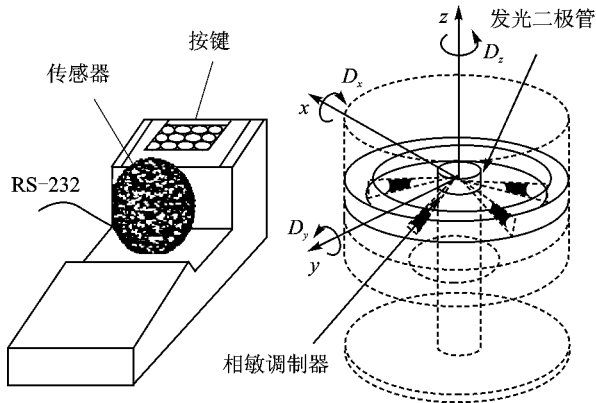
### 圆媛 三维鼠标

目前每一台计算机上都配有鼠标,用来对计算机屏幕上的信息进行平面(在曾轴和赠轴上)定位。虚拟现实空间是一个三维空间,因此,就必须研制出能在三维空间内(在曾轴、赠轴和拽轴上)进行定位的“鼠标”。最常见的三维鼠标有两种,即跟踪球(哪精建造)和三维空间探针,它们都是用增量位置控制的方法来实现空间定位的。

跟踪球是一个刚性的球体,其中心是固定的,装在一个凹型的支架上,并有一个活动的外层,装有远个发光二极管和相应的远个光电传感器,如图愿-远所示。它可以舒适地被放在使用者的手掌之下,当使用者用手对该球的外层施加力和力矩时,根据“弹簧变形”规则,就可通过远个光电传感器测出猿个力和猿个力矩的信息,传输至计算机后,在主机中经过软件处理并



返回 灾砸物体位置和方向的差动变化,变化越大意味着 灾砸物体运动速度越快,从而求出虚拟空间中某物体的位置和方向等。这就清晰、充分地提供了与三维对象进行交互的必要功能。



图愿-远 跟踪球原理

在球的支撑架上还放有几个在用户手指活动范围内的按钮。这些按钮工作状态为开或关,按钮功能由软件编程定义。例如:一个按钮用来增加 灾砸物体运动的速度;一个用来降低运动速度;另一个按钮用来将球与它所控制的 灾砸物体进行“耦合”和解除耦合处理;其他按钮可用来启动和停止仿真,或者当使用者迷失方向时重新启动等功能。

跟踪球受到传感器耦合的影响。当使用者希望 灾砸物体位置发生变化但不要求是旋转时,却偏偏发生偏移和旋转,这是因为用户手指施加在球上的力矩不为零所造成的。这些不希望出现的运动可通过软件“过滤”来消除。

这种球的缺点是它本身不能进行移动。跟踪球的性能可用分辨率、刷新速率、滞后时间及可跟踪范围来测量。

三维空间探针是安装在一个固定的基座上,具有 远个关节的机械臂(图愿-苑中的第 园号到第 缘号关节)。在每个关节上装有传感器,计算机接收传感器的信息,并计算出在虚拟空间的物体的位置和方向。

### 猿援数据手套

跟踪球和 猿- 园探针的优点是简单、紧凑和工作速度快,但是其特性限制了用户手的运动范围只是在桌上很小的区域内,于是用户手的自然的运动就受到影响。为了与仿真有更大的基于手姿的交互,必须要求 猿- 园的工具能处理手在一定空间的自由运动,也希望通过感觉单个手指的运动,得到更多的自由度。人的手指动作有“弯曲-伸直”,也有侧向“外展-内收”。此外,拇指动作有“前位-复位”,前位使拇指与手掌相对。

在虚拟环境中最常用的人机接口工具是数据手套(猿- 园)。数据手套作为虚拟现实技术中的一种重要输入设备,一直受到极大关注,国外先后已有许多产品出现,如 灾- 猿公司的 猿- 园

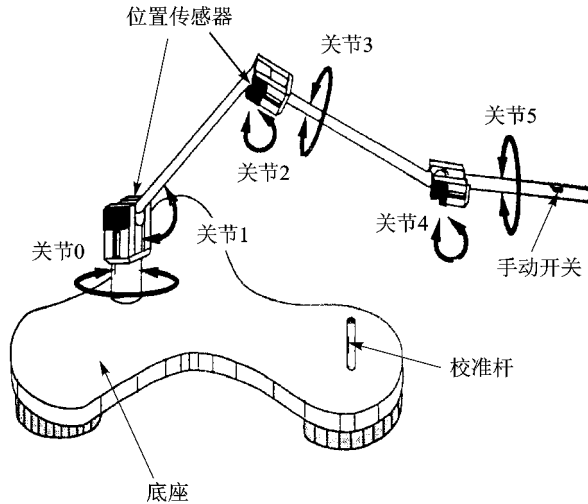


图 8-7 三维空间探针

美国宇航局( NASA)的支持下研制成功的。人的手具有 6 自由度,其中 3 个为臂部自由度,表示手的位置和方向;掌部有 3 自由度。每个手指( II ~ 灾)有 2 自由度,其中手指基部( 配孕)有 1 自由度。中间关节( 孕孕)和末端关节( 阙孕)只有 1 自由度。大拇指有 2 自由度。数据手套作为手的数字化输入设备,用来检测手指关节间的相对转角和手指间的相对转角。它的测量机制直接影响了数据手套的性能。数据手套通常是一种尼龙手套,一般采用光纤、导电油墨或云母(导电橡胶)作为介质依附在手套上,作为传感装置。在每个手指上安装有相应的弯曲和扭曲传感器,如图 8-7 和图 8-8 所示。计算机利用传感器的信息,控制虚拟空间中物体的位置和方向。于是就产生了人能用自己手,去控制虚拟空间中虚拟物体的感觉。

为了使戴上数据手套的手不仅能搬动虚拟物体,还能感受虚拟物体“应有的”质量和质感,更先进的数据手套还具有触觉。最简单的做法是在手套上安装气压式和振动触感式的反馈器。在计算机的控制下,向手套的气袋充气时,戴上这种数据手套的手就感受到一定的压力了,用这种办法就可以模拟手触摸真实物体时的感觉。上面所介绍的只是一般原理,真正投入使用的具有触觉的数据手套,都是根据具体的应用需求而进行专门设计和研制的。

#### 跟踪测量设备

在虚拟现实系统中,为了与虚拟环境交互操作,必须感知操作者的视线,也就是说,必须跟踪头部的位置和方向。为了在虚拟环境中移动物体,甚至移动参与者的身体,则要求跟踪观察者各肢体的位置。所以虚拟现实环境中的另一项关键技术是跟踪,即对虚拟现实环境中的用户的位置和方向进行实时精密的测量。不同的应用要求可选用不同的跟踪器,因此,必须研制与此相适应的跟踪装置,而传感器技术是实现这一功能的关键硬件技术。头部跟踪设备用于





跟踪操作者头部的位置和方向,主要为了感知操作者的视线,动态地产生虚拟环境中人“看到”的图像。模拟人在虚拟环境中的运动。虚拟环境中的参与者所看见的虚拟场景的变化和所听到的虚拟声的变化,取决于参与者当前的位置和头部的姿态(方位角和仰角)及手的操作位置和触摸的方式。为了采集并能跟踪头位和手位,使输入信息自然和谐,在VR系统中采用专用的跟踪设备。

#### (一) 跟踪设备组成

通常的跟踪设备由三部分构成:产生信号的信号源、接收信号的传感器、处理信号的控制盒。信号源或传感器固联在参与者的身体上(如手上和头上),另一部分置于环境中固定位置作为参考点。

#### (二) 影响跟踪性能的因素

跟踪设备的性能取决于信号测量的速率、精度及传输到计算机中的延迟。

现在有基于不同原理及技术的跟踪设备,举例如下。

① 机械跟踪设备:采用机械装置使参与者身体部位和参考点连接起来,用机械设备的位移、方位测量手和头的位置与方向。其测量精度高,时间延迟短(小于1ms),更新率高,但是限制了参与者的运动范围,使用方式不灵活,对操作员的机械束缚也比较多。

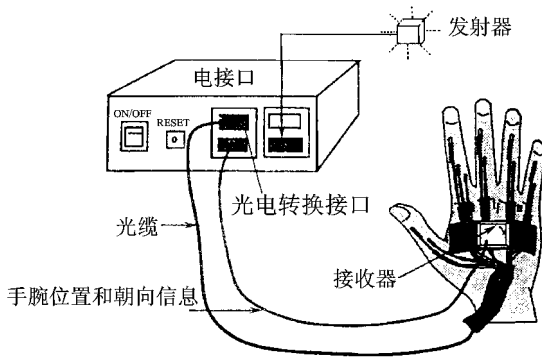
② 光学跟踪设备:现有两种光学跟踪方式,一种是摄像头固定在头盔(或面罩)顶部,一套红外光源置于头顶上的固定位置上(如天花板上)。另一种是光源固定于天花板上,摄像头固定于头盔上,发光二极管光源采用脉冲式。参与者的位置与方向由摄像头摄光源的投影来确定。在头盔式显示器中装有发光二极管传感器,通过光电管产生电流的大小及光斑中心在传感器表面的位置来推算出头部的位置与方向。一般的光学跟踪器精度高,刷新快,滞后时间短,但是缺点是用户的活动范围小。另一种更有前途的光学跟踪器叫做自跟踪器,它利用图像处理原理,通过头盔显示器上安装的光传感器和专用图像处理机,以每秒60帧图像进行相关分析,从而推算出头部的位置和方向。由于这里不要求在环境中安装任何信号标志,因此工作范围将不受到限制。目前如何控制测量中的累积误差来控制跟踪精度仍然是一个难题。

③ 电磁跟踪设备:电磁式跟踪器的基本原理是在空间的相对固定点上安放一个磁发射器,该发射器沿三个正交方向发射一束电磁波,在被测运动体上安装一个接收器,接收器内有三个正交的接收线圈(天线),分别接收发射器发出的电磁波,通过测量这种设备中三个垂直正交小金属线圈内电流产生的磁场来计算被跟踪物体的位置和方向,如图8-1所示。这是目前使用最多的一种,已经有商品出售了,如索尼公司的电磁跟踪器。它的优点是体积小、安装灵活和价格便宜,但是缺点有滞后时间略长(约10ms左右),跟踪范围小,而且周围范围内的金属和电磁场会使信号畸变,影响跟踪精度。

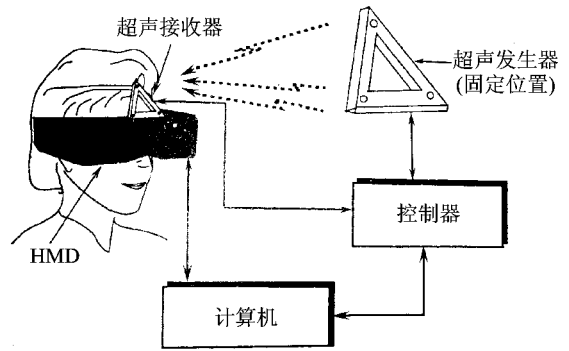
④ 声学跟踪设备:这种跟踪器的工作原理与电磁式的相似,头盔上安装有传感器,通过对超声传输时间的三角测量来定位。利用超声波测量被跟踪物体(手、头)的位置,手和头部安装有超声波接收器,参考点是在一定区域内的多个固定位置和方向上,有多个超声波发生器发



出信号,如图愿-7所示。超声波传感器检测距离的手段多种多样,但最主要的方法是检测渡越时间。渡越时间是指超声波从发射器发出到超声波接收器接收到声波所经历的时间间隔。声源与目标之间的距离与声波在声源与目标之间传播所需的时间成正比,测量出渡越时间就可计算出声源与目标之间的距离。根据多个不同位置的超声波对同一个超声波发射器进行测距,通过计算可确定这个超声波发射器在二维和三维空间的位置。这种测量距离的方法对对象的材料性质依赖较小,并且不受电磁波、粉尘等干扰。



图愿-7 电磁跟踪器



图愿-8 超声波跟踪器

目前电磁跟踪式和超声波跟踪式设备用的较多。表愿-9列出各种位姿跟踪器的性能比较。

表愿-9 各种位姿跟踪器的比较

	发射源	便携性	精度	价格	刷新率	延迟	范围
机械式	无	差	很高	较高	高	很小	很小
磁场式	有	好	较高	较低	较高	较大	单传感器小 多传感器大
超声式	有	较好	时间法低 相位法高	很低	时间法低 相位法高	小	较小
光学式	有	较好	较高	昂贵	高	小	大

### 触觉系统

触觉系统是模拟人在真实环境中的触觉、力感的装置,用来测量用户手或其他部位的位置和受力,显示受力的时间和空间分布,给人产生力矩和力的感觉。机械稳定性、可靠性和安全性是研究的首要因素。气压式和振动触感式反馈器的安全性有较好的实现方式,它们都用于数据手套,在计算机的控制下通过挤压感和振动感为用户提供力反馈作用。触觉系统方面的



研究同样分为两方面:受力的测量和力反馈的产生和显示。为使用户得到较为真实的感受需要进行受力的建模与仿真工作。

### 视觉音频系统

用声学建模方法为物理现象产生声音,这种方法在实际应用中并不容易实现。虚拟现实系统通常采用录音的方法获得声音素材,并将声音数字化。目前研究的重点集中在声音的定位。声音的时间差和相位差影响着人对声音的定位,人在定位方面的机理还有待进一步研究。由于声音与视觉等其他感知存在一致性,对应于不同的视觉显示方式,通常采用立体声耳机和扬声器阵列这两种方式实现虚拟现实系统中的声音模拟。立体声耳机匹配头盔显示器,头部相关函数(Head-Related Transfer Function, HRTF)是其定位方式,对应于投影显示的扬声器阵列也具有自己的空间定位模型。

## 视觉反馈原理及三维图像生成

### 计算机实时三维图像生成基本过程

在人与虚拟环境、物体交互过程中,视觉信息是虚拟环境给人的重要的信息反馈。系统视觉信息反馈通道主要包括三维图像实时生成模块和图像显示设备两个环节,如图 8-1 所示。

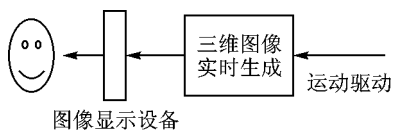


图 8-1 视觉信息反馈通道

计算机成像系统就是在计算机屏幕上(或其他二维显示介质)由计算机生成一幅画面,这个画面模拟人通过屏幕这个假想的窗口观察三维世界所看到的景象。

成像计算机的任务是如何在二维屏幕上逼真地生成模拟人的眼睛观察三维景物时所看到的图像。

### 三维图像数字化建模

计算机是以数字形式处理信息的,如果要用计算机处理一个三维的环境(或说三维世界)就必须以数字化的形式描述这个三维环境,这个过程就是数字化建模。数字化建模要描述的内容主要包括几何特征、材质特征、自然特征、物理特征等。根据不同的要求和用途,建模可以包括上述内容中全部或部分内容。成像系统一般要求建模生成的三维环境模型应具有几何、材质及自然三方面的特征,对于一些相对简化的成像系统,只需要前两项即可。

通常将二维或三维目标简化为由平面多边形拼成的几何形体,多边形可以是三角形、四边形或晕边形。多数系统采用一个确定边数的多边形作为系统处理的基本面元,如三角形。组成物体的基本三角形用点表加面表来表示。

以立方体为例,是个顶点、6个面的几何体可表示如下。





在眼坐标系和屏幕坐标系中的坐标, 设  $(x_d, y_d, z_d)$  为当前观察点(眼点)在地坐标系中的位置坐标(观察点是可变的)。如图 8-1 所示为各个坐标系间关系。

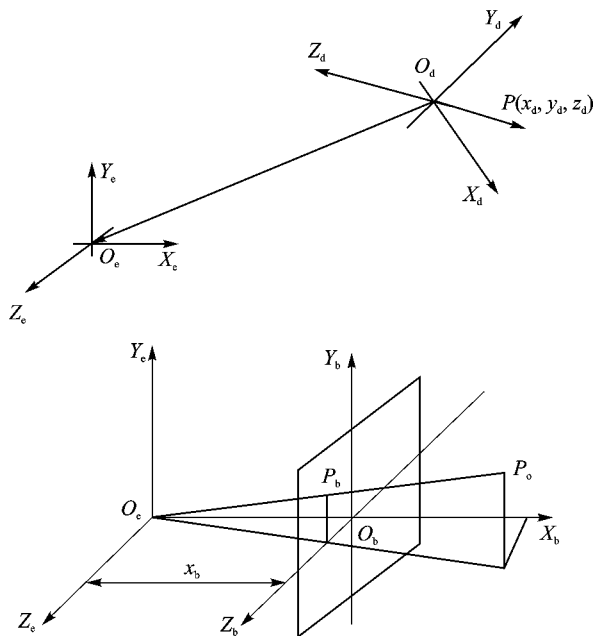


图 8-1 各个坐标系间关系

眼点在眼坐标系中坐标为  $(x_e, y_e, z_e)$ , 在屏幕坐标系中坐标为  $(x_b, y_b)$ 。

眼点由地坐标到屏幕坐标的变换过程为

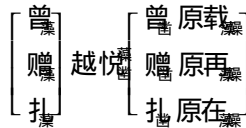
(1) 眼点作平移变换, 得  $(x_e', y_e', z_e')$

$$\begin{bmatrix} x_e' \\ y_e' \\ z_e' \end{bmatrix} = \begin{bmatrix} x_e - x_o \\ y_e - y_o \\ z_e - z_o \end{bmatrix} \quad \begin{matrix} \text{在眼坐标系 } x_e \text{ 值} \\ \text{在眼坐标系 } y_e \text{ 值} \\ \text{在眼坐标系 } z_e \text{ 值} \end{matrix}$$

(2) 将  $(x_e', y_e', z_e')$  作旋转变换, 地坐标系至眼坐标系旋转变换阵为

$$R = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8-1)$$

式中  $\cos \alpha, \sin \alpha$  为地坐标系到眼坐标系的方向余弦, 它由眼睛当前的观察姿态确定。眼点在相应眼坐标系的坐标  $[x_e', y_e', z_e']$



(愿-圆)

(猿) 透视变换:景物空间中的数字化模型转换到眼坐标系后,还必须通过透视投影将三维模型投射到二维的显示屏幕上,才能够显示。将眼坐标系坐标投影到屏幕上,必须使用透视变换,设  $z_0$  为眼睛距屏幕的垂直距离,即  $z_0 = \frac{z}{\cos \theta}$ ,则通过相似三角形边之比相等得

$$\frac{z_0}{z} = \frac{y_0}{y} = \frac{x_0}{x} \quad \text{为已知确定值}$$

则  $y_0 = \frac{z_0}{z} y$ ,  $x_0 = \frac{z_0}{z} x$  (愿-猿)

这样就计算出景物空间某一点经透视投影后在屏幕坐标系上的值。

### 愿-圆 剪裁与消隐

在进行图像生成时,要将视锥以外的物体剔除掉,这个过程叫做剪裁。在视锥内的物体,有时彼此间存在局部或全部的遮挡,有些物体的一部分是背离观察者视向量的,这些也都是不可见的,剔除背离面和遮挡面的过程称为消隐。

#### 愿-猿 剪裁

剪裁可以在景物坐标系中进行,也可以在眼坐标系或屏幕坐标系中进行。

一般来讲,在景物坐标系中的剪裁对于运算负荷来讲是最经济的,眼坐标系剪裁次之,屏幕坐标剪裁最差。这是因为,在景物坐标系中将视锥以外的物体模型全部剪裁掉后,后继的坐标变换只要计算视锥以内的物体就可以了,这样可以大大减小计算负荷。如果将景物坐标系中的物体经过坐标变换,变换到眼坐标中后再进行剪裁,则对于那些不可见的物体也进行了坐标变换,浪费了计算资源。

#### 愿-圆 消隐

消隐包括两部分内容,一是背离面的剔除,二是遮挡关系的处理。所谓背离面实际上就是面的法向量与视向量的夹角在  $(\frac{\pi}{2}, \pi)$  之间的面。根据这一定义,在已知面的法向量的情况下,很容易确定哪些面是背离面。当景物空间存在多个三维物体时,可能会出现一些物体遮挡了其他物体表面的情况,景物中被完全或部分遮挡的前向面,某个位置能观察到的表面,称前向面或隐藏面。

### 愿-圆 纹理与光照效果

纹理是指物体表面的细节,是一种物体区别于其他物体的质地特征。纹理映射是将纹理映射到由多边形组成的三维景物上,是使三维图像具有逼真感的有效方法。这些图像细节可以提供虚拟景物相对观察人员的高度、速度、加速度、位置等变化的逼真视觉。纹理是通过纹



理映射生成,所谓纹理映射是将给定的纹理模式映射到物体表面上,通常采用逆透视投影的算法,即对屏幕上的每一个像素,依照屏幕空间与景物空间的逆透视投影关系,得到它相应的三维景物空间点的位置坐标,然后再按此三维景物空间点到纹理空间的映射关系,得到屏幕像素点相应的纹理点,取得纹理值。在纹理处理中会因为对纹理空间采样太密(原始纹理模式分辨率低)在屏幕上出现方块,或因对纹理空间采样过稀而出现混叠现象。

光对物体照射产生浓淡、阴影、透明等能见度效果。

目前在工作站上或微机上均已有的专门的建模和图像驱动软件,建模软件如 3ds Max、SolidWorks、AutoCAD、Maya、Blender 等,场景驱动软件有 Unreal Engine 和 Crysis 等。这些软件使三维景物的建模和驱动工作大为简化,而且很容易获得具有各种特殊效应和光照效果的逼真的三维景物,使虚拟世界更加逼真和丰富。

Pro/Engineer 是一个高度专业化的工具,帮助建模者创建高效的三维模型和地形用于交互式实时应用。采用这些工具软件使得输入、结构化、修改、创建原型和优化模型数据库更容易。视景仿真建模和 3D 数据生成有很大的不同。因此,即使是最好的 3D 模型也需要大量的处理,使之能用于实时应用。因为 Pro/Engineer 工具创建精确的工程数据模型,包含大量用于工程和绘图的有用信息,Pro/Engineer 的目的是模型精确性和机械整体性,Pro/Engineer 数据包含了工程标准、精确材质以及大量的属性数据,同时其图形目的是用于工程规划和静态图形生成的、基于复杂规则的数据。

Pro/Engineer 建立的是包括符合 3ds Max 文件格式标准的层次视景数据库,适用于实时模型建立和实时应用,主要用于仿真,如飞行模拟器等。

### 3.1.2 立体成像

人是通过两只眼睛观察世界的,通过眼球的透视变换,在左、右眼的视网膜上分别获得两幅彼此间有一定相位差异的二维图像,大脑对这两幅图像进行合成处理,形成一个具有深度感的三维景象。

通过计算机进行立体图像生成时,就是要分别为左、右眼生成两幅不同的图像,通过一定的设备分别呈现给两只眼睛。

#### 3.1.2.1 单眼显示屏幕立体成像

首先,定义在立体成像中的眼坐标系。设两眼间距为  $2d$ ,眼睛距屏幕的距离为  $z_0$ ,以两眼的中间点为坐标原点,目视的正前方为  $x$  轴的负方向,两眼连线并由右向左的方向为  $y$  轴,垂直于  $xy$  平面的方向为  $z$  轴。

#### 3.1.2.2 双投影中心(双投影)方法

对于眼坐标系任一点  $P(x, y, z)$ ,当分别以左眼和右眼为投影中心时,该点在屏幕上的像点分别为  $P_L$  和  $P_R$ 。

将坐标原点沿  $y$  轴正方向平移  $d$  距离,即坐标原点与左眼重合,根据前面介绍的透视



投影的映射关系,有

$$\begin{matrix} \text{曾} & \text{越} & \text{赠} & \text{越} & \text{苗} \\ \text{原} & \text{藻} & \text{赠} & \text{越} & \text{扎} \\ \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} \end{matrix} \quad (\text{愿-源})$$

于是

$$\begin{matrix} \text{曾} & \text{越} & \text{苗} & \text{藻} & \text{原} & \text{藻} \\ \text{原} & \text{藻} & \text{扎} & \text{圆} & \text{圆} & \text{圆} \\ \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} \end{matrix} \quad (\text{愿-缘})$$

再将坐标原点移回原位,即

$$\begin{matrix} \text{曾} & \text{越} & \text{苗} & \text{藻} & \text{原} & \text{藻} & \text{垣} & \text{藻} \\ \text{原} & \text{藻} & \text{扎} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} \\ \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} \end{matrix} \quad (\text{愿-远})$$

这就是左眼在屏幕坐标中的像点。类似的方法可以获得右眼在屏幕坐标中的像点位置为

$$\begin{matrix} \text{曾} & \text{越} & \text{苗} & \text{藻} & \text{原} & \text{藻} \\ \text{原} & \text{藻} & \text{扎} & \text{圆} & \text{垣} & \text{藻} & \text{圆} \\ \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} & \text{圆} \end{matrix} \quad (\text{愿-苑})$$

摇摇(圆) 旋转投影(愿-苑)方法

双投影中心方法只考虑了双眼的分离,而没有考虑眼球在凝视时的旋转。如图愿-苑所示,在正视的情况下,当人的双眼凝视空间某一点时,左右眼的视向量会相交于凝视点,这时每只眼睛的视向量都会向内偏转 $\frac{\theta}{2}$ 角,其中 $\theta$ 是左右视向量的交角,如图愿-苑所示。

在上述眼坐标定义的基础上,再定义左眼坐标系的载,再在,和右眼坐标系的载,再在,如图愿-苑所示。于是,眼坐标系到左眼坐标系的变换矩阵为

$$\text{栽} \begin{matrix} \text{越} \\ \text{越} \\ \text{越} \\ \text{越} \end{matrix} \begin{bmatrix} \text{藻} & \text{原} & \text{藻} & \text{原} & \text{藻} \\ \text{原} & \text{员} & \text{园} & \text{园} & \text{园} \\ \text{藻} & \text{园} & \text{藻} & \text{原} & \text{藻} \\ \text{园} & \text{园} & \text{园} & \text{园} & \text{员} \end{bmatrix} \quad (\text{愿-愿})$$

眼坐标系到右眼坐标系的变换矩阵为

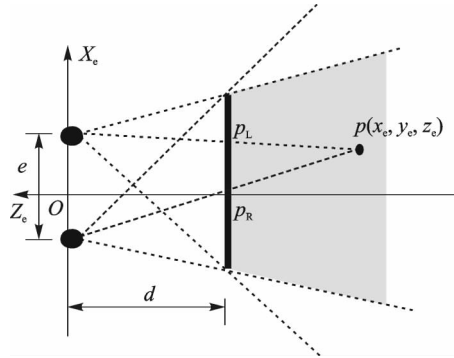


图 8-1 双投影中心法立体成像

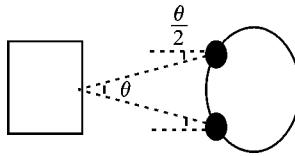


图 8-2 旋转投影

$$\begin{bmatrix}
 \frac{z_c}{d} \cos \theta & \frac{z_c}{d} \sin \theta & \frac{z_c}{d} & 0 \\
 \frac{z_c}{d} \sin \theta & \frac{z_c}{d} \cos \theta & 0 & 0 \\
 \frac{z_c}{d} & 0 & 1 & 0 \\
 0 & 0 & 0 & 1
 \end{bmatrix} \quad (8-1)$$

于是, 点  $P(x_c, y_c, z_c)$  在左眼坐标系中的坐标为

$$\begin{pmatrix}
 \frac{z_c}{d} \cos \theta x_c - \frac{z_c}{d} \sin \theta y_c + z_c \\
 \frac{z_c}{d} \sin \theta x_c + \frac{z_c}{d} \cos \theta y_c \\
 z_c \\
 1
 \end{pmatrix} \quad (8-2)$$

经透视变换, 在屏幕上的像点为

$$\begin{pmatrix}
 \frac{z_c}{d} \cos \theta x_c - \frac{z_c}{d} \sin \theta y_c + z_c \\
 \frac{z_c}{d} \sin \theta x_c + \frac{z_c}{d} \cos \theta y_c \\
 z_c \\
 1
 \end{pmatrix} \quad (8-3)$$

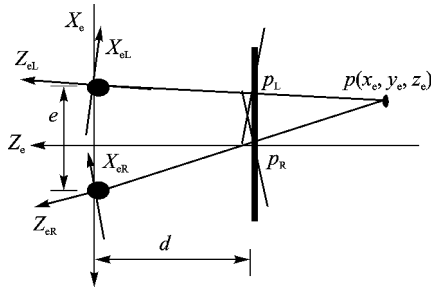


图 1 旋转投影法示意图

$$\begin{aligned} \begin{matrix} X_{eL} \\ Z_{eL} \end{matrix} & \begin{matrix} X_o \\ Z_o \end{matrix} & \begin{matrix} X_{eR} \\ Z_{eR} \end{matrix} & \begin{matrix} P_L \\ P_R \end{matrix} & P(x_o, y_o, z_o) \\ \text{左眼} & \text{原点} & \text{右眼} & \text{像点} & \text{物点} \end{aligned} \quad (1)$$

类似地，点  $P(x_o, y_o, z_o)$  在右眼坐标系中的坐标为

$$\begin{pmatrix} X_{eR} \\ Z_{eR} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_o \\ Z_o \end{pmatrix} \quad (2)$$

像点坐标为

$$\begin{aligned} \begin{matrix} X_{eL} \\ Z_{eL} \end{matrix} & \begin{matrix} X_o \\ Z_o \end{matrix} & \begin{matrix} X_{eR} \\ Z_{eR} \end{matrix} & \begin{matrix} P_L \\ P_R \end{matrix} & P(x_o, y_o, z_o) \\ \text{左眼} & \text{原点} & \text{右眼} & \text{像点} & \text{物点} \end{aligned} \quad (3)$$

$$\begin{aligned} \begin{matrix} X_{eL} \\ Z_{eL} \end{matrix} & \begin{matrix} X_o \\ Z_o \end{matrix} & \begin{matrix} X_{eR} \\ Z_{eR} \end{matrix} & \begin{matrix} P_L \\ P_R \end{matrix} & P(x_o, y_o, z_o) \\ \text{左眼} & \text{原点} & \text{右眼} & \text{像点} & \text{物点} \end{aligned} \quad (4)$$

式中，

$$\theta_{eL} = \arctan \frac{Z_{eL}}{X_{eL}} \quad (5)$$

$$\theta_{eR} = \arctan \frac{Z_{eR}}{X_{eR}} \quad (6)$$

### 双显示屏幕立体成像

在双显示屏幕的情况下，左眼和右眼可以看作是两个独立的显示系统，将坐标原点沿  $X_o$  轴正方向平移  $\frac{e}{2}$  距离，即坐标原点与左眼重合。根据前面介绍的透视投影的映射关系，有

$$\begin{aligned} \begin{matrix} X_{eL} \\ Z_{eL} \end{matrix} & \begin{matrix} X_o \\ Z_o \end{matrix} & \begin{matrix} X_{eR} \\ Z_{eR} \end{matrix} & \begin{matrix} P_L \\ P_R \end{matrix} & P(x_o, y_o, z_o) \\ \text{左眼} & \text{原点} & \text{右眼} & \text{像点} & \text{物点} \end{aligned} \quad (7)$$

图 2





于是

$$\left. \begin{array}{l} \text{黄} \text{越} \text{苗} \text{ ( 原 圆 )} \\ \text{赠} \text{越} \text{苗} \text{ 赠} \text{扎} \end{array} \right\} \quad (\text{愿-圆})$$

这一像点就是黄赠扎点在左眼屏幕上应显示的像点。同样,可以求出黄赠扎点在右眼屏幕上应显示的像点

$$\left. \begin{array}{l} \text{黄} \text{越} \text{苗} \text{ ( 垣 圆 )} \\ \text{赠} \text{越} \text{苗} \text{ 赠} \text{扎} \end{array} \right\} \quad (\text{愿-圆})$$

双显示屏幕立体成像示意图如图愿-苑所示。

### 愿-苑 摇摇视觉环境的构造方式

摇摇(员) 固定 悦形式,放于桌面,可多人共享,用户约束感小,易于实现交互,但视角小,无环视感。一般要求用户佩带光闸眼镜,并配以视线跟踪装置。

摇摇(圆) 匀型,一般用两个 蕴屏形成立体对,头盔结构中安装广角光路,同时附有视线运动检测装置,用户走动自由,但头部约束感较强,蕴分辨率低。

摇摇(猿) 匀(双目全向监视器,遥测器,摄像机,操纵器,操纵器)形式类似于匀,但显示装置悬挂于机械臂,重量得到平衡,位置与姿态得到测量,用户用手操纵,约束感适中,装置可配以高分辨率的悦对。

摇摇(源) 悦形式,大视角、高分辨率、环绕视觉感较强,用户只需佩带轻质光闸眼镜,约束感小,也可多人共享,但需较大的安装空间和投影系统。

摇摇(缘) 灾形式。员年, 提出“人工现实( )”的思想,展示了称为灾的“并非存在的一种概念化环境”。实验者面对投影屏幕,摄像机摄取他的身影轮廓图像,与计算机产生的图像合成,由投影机投射到屏幕上,室内的传感器识别实验者的动作,屏幕上可显示诸如实验者在游泳、爬山等情景。参与者可以是一个也可以是多个;当有多个参与者时,屏幕上会在不同位置投影出各个参与者的真实图像。这种方式体现了 提出的无障碍环境( )人工现实的思想,称为无障碍虚拟环境系统,它是一个综合地运用了计算机图形学和手势识别技术的反应环境( )。这种“无障碍”方式目前不是灾的主流。

## 愿-缘 三维虚拟声的生成原理与显示方法



## 图 1-10 三维虚拟声的生成与显示

人们根据到达两耳的声音强度与相位差来区别发声的方向与位置,而且善于同时处理多个事件。例如当视觉系统处理某一事件时,听觉系统同时以比视觉系统低得多的频带宽度在后台工作。由于人们的听觉系统很善于在众多的声音中选取特定的声音,因此在虚拟现实系统中加入声音合成装置对增强沉浸感十分有效。

虚拟环境中的声音生成和显示原理及方案如图 1-10 所示。在该方案中,首先根据虚拟环境中实体、听者以及环境的声音传播特性,计算出声源实体相对听者的声音属性的变化,包括方位、距离、声音强弱及工作状态的变化,对每一种变化,根据不同的仿真模型算法,如声音的定位算法、距离模型、光线跟踪算法等,生成相应的声音数据,然后混音、播放、显示。

### 1. 声源建模与合成方法

声源的建模是指对原始声源采集后,对声音信号进行频谱分析,在频域中得到声音信号的各个谐波分量的幅值、相位。声源的合成是指根据实体的工作状态,把声音的各个谐波分量叠加、合成,然后播放显示。如飞行仿真器音响系统所能生成的各种声音都来源于飞机制造商提供的磁带记录,磁带中包含有飞机飞行时机组人员所能听到的各种声音的样本。用专门的谐波分析软件对每一声音进行分析,得出频率、幅值和主要的谐波分量。再将这些值以表格的形式存储到计算机中,以供生成声音时查表使用。

数字建模、合成方法生成的声音的音质不如录制的声音的音质好。为保证建模、合成的声音有较好的音质,对合成的设备有较高的要求,计算量较大,但此方法占用内存少,并且可以通过人机交互改变实体工作状态而自然、和谐地改变声音,交互性较好。

### 2. 声音素材的直接采样、动态调度方法

声音素材的直接采样是指把虚拟环境需要的声音素材直接录制成声音文件保存在声音文件数据库中,在运行时,根据虚拟环境对声源的需求,动态调度声音文件,包括循环和非循环声音文件。

在该声音生成和显示系统中,对声源素材的处理采用直接采样、动态调度方法。根据实体的工作状态把声音信号分别录制下来,放在内存中,然后根据虚拟环境对声音的要求,对声音文件动态调度,并根据实体的运动速度来调制声调,从而仿真实体的工作状态变化。

直接采样、动态调度方法比建模合成方法获得的声音音质好且计算量小,但不能做到声音变化的自然平滑,且交互性差,素材占用较多的内存。

## 图 1-11 声音的距离仿真

在声音的距离仿真中,关于信号的增益与人和声音信号的绝对距离之间尚未有明确的定量关系。

不同的声源有不同的绝对音量级别(如飞机引擎、火车汽笛等)。比如人距飞机引擎噪声处就感

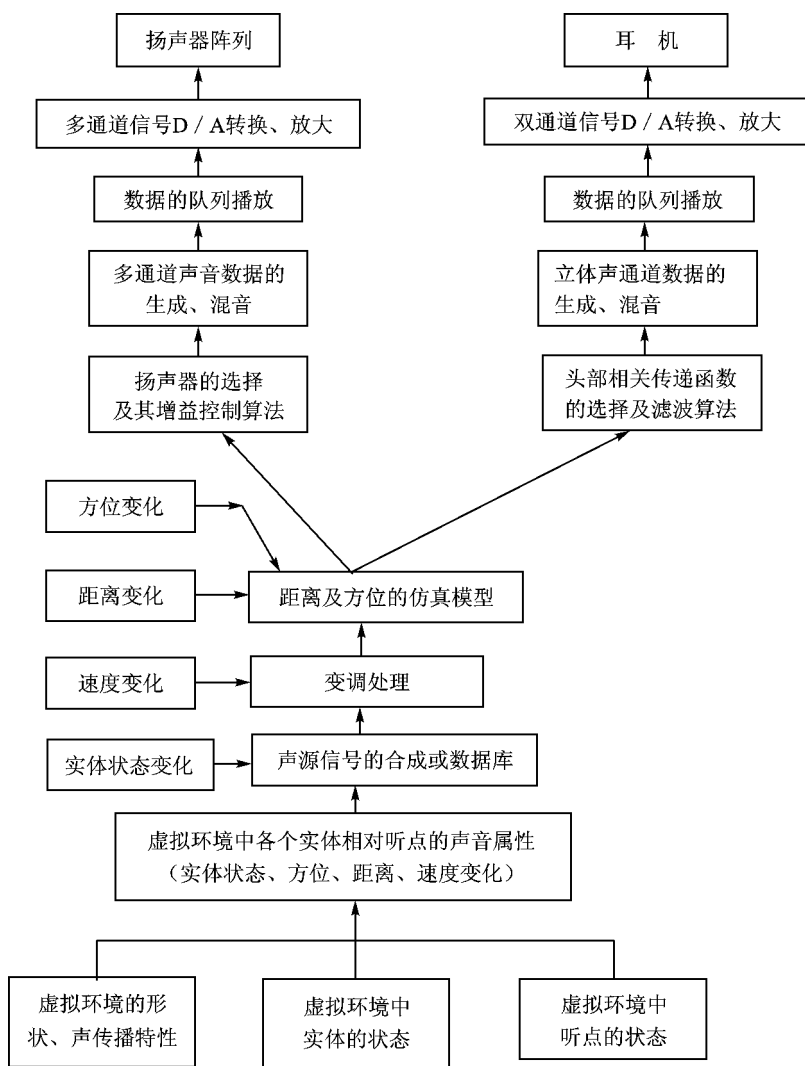


图 8-1 虚拟声生成、显示原理及方案

觉声音很响,而距蜜蜂 4m 处已听不到声音了。而采集声源时并没有考虑各种声源之间的这种差异,为了补偿不同声源在这方面的差别,需要为各种声源设定能感知声音的最小距离  $d_{min}$  和最大距离  $d_{max}$ 。最小距离对应了每个声源的最大音量级,即使小于此距离,声音的音量不再变化。最大距离则对应了声源的最小音量级,一旦超出此距离声音就听不见。

设最小距离和最大距离之间的空间距离为  $d$ ,采用增益系数和距离之间的反比例关系模



型来计算声源距离的增益系数  $\frac{1}{r}$

### 猿猿声音的运动仿真

声音的运动不仅表现在声音位置、距离的变化,还表现在声源、听者之间相对运动时所产生的多普勒(猿猿)效应。

#### 猿猿声音的位置变化

虚拟环境中的发声实体会给虚拟环境带来声音事件,实体的运动以及听点的运动,会造成声源、听点之间相对位置的变化。声音生成系统可根据声源的位置、人的位置和姿态,进行几何坐标变换,计算出声音事件和听点之间的相对方位和距离。

设定虚拟环境中实体、听点在同一个坐标系中,实体的位置坐标,即点声源的坐标为  $(x_s, y_s, z_s)$ ,听者的位置坐标为  $(x_l, y_l, z_l)$ ,其头部姿态在听点坐标系下为  $(\varphi, \theta, \gamma)$ 。则点声源、听者之间的相对方位关系为

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{bmatrix} \cos\theta \cos\varphi & \cos\theta \sin\varphi & \sin\theta \\ -\sin\theta \cos\varphi & -\sin\theta \sin\varphi & \cos\theta \\ \sin\varphi & \cos\varphi & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \quad (愿-愿)$$

即  $[粤] = [粤] \cdot [悦]$

粤:声源在听点坐标系中位置;

月:声源坐标转换矩阵;

悦:声源在地坐标系中位置。

一旦知道了声音和听点之间的方位关系,采用声音的空间化技术,针对扬声器、耳机两种输出设备,根据其各自的生成算法,对新方位的声音进行定位。

#### 猿猿声音运动的平滑过渡

当声源由一个位置运动到下一个位置时,为实现两点间声音的平滑过渡,可采用声音信号的交叉衰减技术,设定一定的交叉衰减时间,如猿猿当对声源重新定位时,不直接采用新位置的声音,在猿猿,使前一个位置的声音逐渐衰减,新位置对应的声音逐渐加强,直到完全变成新位置的声音。

#### 猿猿声音和多普勒效应

下面给出一种声源多普勒效应的近似处理方法,由于声音素材是在实体、听者相对静止的情况下录制的,即声源素材中的数据对应了信号的原始频率,当任一实体和听者相对运动时,根据二者的速度大小和方向,代入多普勒效应的计算公式  $f' = f \frac{c \pm v_l}{c \pm v_s}$  为观察到的表观频率



率  $v$  为声速,  $v_o$  为观察者对于媒质的速度,  $v_s$  为声源相对媒质的速度,  $f$  为原始频率), 得到表观频率和原始频率的倍数关系, 即得到新的声音数据和原始声音数据之间的关系。若  $v_s > v$ , 则频率变快, 需要在原始的声音数据中, 抽取  $v/v_s$  的倍数对应的点, 生成  $v/v_s$  对应的声音数据块; 若  $v_s < v$ , 则信号的频率变慢, 需要在原来的声音数据中插值。新生成的数据也以原始信号的播放频率播放, 这样, 就可使输出的声音具有多普勒效应。

### 图 8-15 三维虚拟声音的合成与定位

在显示(播放)虚拟声(虚拟实体发出的声音)时可采用耳机和扬声器两种设备。在面向单人的虚拟现实系统中, 使用耳机效果好, 在面向多人的虚拟环境中, 如虚拟音乐厅, 虚拟战场等采用扬声器作为声音播放设备比较适宜。以下分别就采用两种声音播放设备, 如何生成立体声和声音的定位进行论述。

#### 图 8-16 虚拟环境中双耳立体声的生成与定位

对同一声源由于左、右耳滤波特性的差异, 在人的头脑中形成立体声的效果。左、右耳的滤波特性表现在时域上的有限脉冲响应, 变换到频域即称为头部相关传递函数(HRTF), 为了使虚拟声通过耳机到达人耳, 能产生立体声的效果, 应将声源通过头部相关传递函数对应的滤波器进行信号处理, 再送入耳机。头部相关传递函数是通过试验得到的, 当输入一脉冲信号, 头部相对不同的方位角和俯仰角位置可测得信号的衰减幅值和相位。在实时计算头部当前位置时, 采用插值方法, 获取当前头部相关传递函数对应的频率幅值和相位, 再转换到时域。

双耳立体声生成过程如下: 虚拟环境中的实体的工作状态激发了一个或多个声音事件, 声音事件决定了声源素材的调度。虚拟环境中实体、人的位置变化以及头部姿态运动, 会造成声音、听点相对位置的变化。声音生成系统对声音事件、人的位置以及人的头部姿态进行几何坐标变换, 计算出声音事件相对于听点的相对位置和距离, 距离决定了信号的增益, 相对位置决定了头部相关传递函数的选择。一旦得到声音事件对应的 HRTF 数据, 就对经过距离增益调节后的声音数据进行滤波, 得到左、右声道数据。对其他的声音事件同样处理。最后, 对左、右通道内的声音数据进行数字叠加, 采用队列、双缓冲机制输出到耳机。如图 8-16 所示为双耳立体声生成和显示系统的原理及结构图。

#### 图 8-17 面向扬声器的虚拟声音的生成与定位

为了让人能够以自然的视、听方式感知虚拟环境信息, 避免头盔、耳机的束缚。在视景显示方面, 人们采用多个大屏幕或无缝的无缝连接来增大视场角。在音响仿真方面, 由于人耳可以根据不同位置的扬声器所发出的声音感知到仿真环境中的动态目标所在的方位和方向, 因此可采用扬声器阵列提供声源。在使用扬声器来生成声音的位置感时, 使用控制声强的方法比较好。

根据人感知虚拟声源位置取决于两个声源矢量的幅值大小(信号增益大小), 当两个声源矢量方向不变, 若增益变了, 则合成后出现不同方位新的声源矢量。如果将扬声器配置在三维

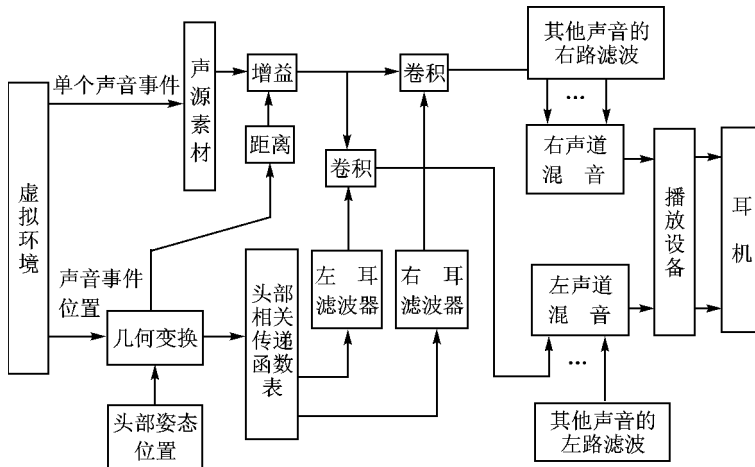


图 愿- 愿瑶 虚拟环境中的双耳立体声生成和显示原理及结构图

空间不同位置,当扬声器音响强弱变化时,就出现了虚拟声在空间的游动。通过控制由三个扬声器输出相干而又幅值不同的信号,可把虚拟声定位在由三个扬声器组成的三角形扇区内。

如果把听者周围的整个空间分制成几个三角区,采用矢量基幅值控制算法,可以把虚拟声定位在听者周围整个空间的任意位置,实现全方位声音定位,如图 愿- 愿瑶 所示。

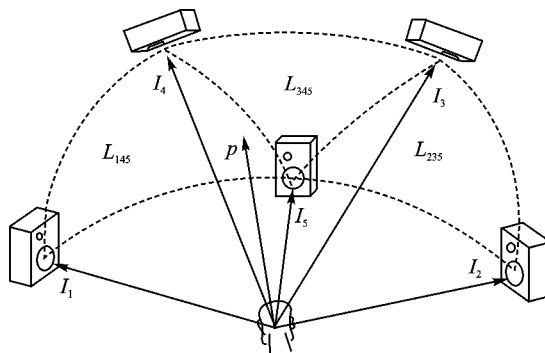


图 愿- 愿瑶 三维虚拟声的生成及显示原理

### 愿瑶瑶 三维音响的显示

在计算机视景系统中,为了提高系统的速度,图形的生成采用双缓冲机制,显示在后台完成。在听觉系统的显示、播放中,为了保证波形数据在播放过程中的连续性,也采用双缓冲机制,并建立声音数据块的输出队列。

比如,在采用双耳立体声进行三维音响的生成和显示时,为了提高基于耳机的声音生成和



显示系统的速度,对数据块的处理、播放分前、后台模块,前台模块生成声音数据流,后台模块播放生成的双耳立体声。

在声音生成和显示系统中,系统周期地对声源与听者的位置采样。在采样周期内,使用对应此相对位置的均匀函数对声音数据块进行滤波得到的数据块放入缓冲区,几个组合成一个帧的数据后,等定时时间到后送声卡播放。每帧播放的数据对应于一个位置,只有在帧变换时,才根据新的位置调用新位置相对应的均匀函数。多个声源事件同时发生时,同样采用分帧混音和播放。

在用耳机接收虚拟声时,要选用高保真的耳机,类似声音信号的全通滤波器,保证耳机的响应速度快,频带宽,不会抑制生成的声音信号。

由于扬声器发出的声音具有指向性,并非向各个方向辐射传播,安装扬声器时,要注意将扬声器发声的方向指向听者。

## 虚拟现实触觉反馈系统

### 虚拟现实触觉反馈原理

触觉反馈是人的重要感知渠道。在VR系统中,触觉反馈是实现人与虚拟环境自然和谐交互的基本要求,也是增加虚拟环境沉浸感、真实感的重要途径。

自然界的事物是千姿百态、千差万别的,在很多情况下,人们不仅希望看到物体的颜色和形状,听到它的声音,还希望通过亲手触摸来感知它的质地或温度,从而获得更多的信息,才会感到比较全面地了解该物体了。这就是说,在虚拟环境中接触和操纵物体需要有触摸和受力的感觉。如远程控制的电动机器人实验证明,触摸和力感提高了完成任务的效率和准确度。李国栋在进行人类因素实验来检测反馈系统时发现,在简单的双指活动场景中,将触觉反馈和视频显示综合起来,其感知性能比仅使用视频显示要提高30%;当视频失效时(模拟低能见度),附加触觉反馈则会使感知性能提高10%。

人的皮肤表面的感觉神经也是非常敏感的,人可以感觉物体在质地、硬度、温度、重量等物理属性的微小差异,从而辨别出不同的物体。可以想象,在虚拟环境中,希望对如此复杂的自然事物的各种物理属性加以逼真描述,然后借助于某种仿真设备反馈给人,是相当复杂的问题。

目前在VR系统关于触觉反馈的研究中,尚无一套系统完善的理论和装置能满足VR系统的应用需要。根据目前国际上的研究情况看,触觉反馈系统大都是针对特定应用而开发的一些具有相当局限性的系统或装置。

当参与者企图用手抓取虚拟环境中的物体时,他的运动首先被运动测量传感器测量,通过计算获得相应的运动参数。然后利用碰撞接触检测算法判断目前手与物体的相对关系,手是



否与物体表面接触,并计算出相应的接触点(面)。根据这些参数以及手所抓取物体的几何、物理属性参数(硬度、表面粗糙性、重量等),触觉传感生成模型计算出控制量,该控制量控制触觉传感装置产生相应的作用,并施加于人的手部。人获得这种感觉后,意识到已经接触到物体,调整下面的动作。上述过程是一个不断循环的实时过程,如图 10-10 所示。

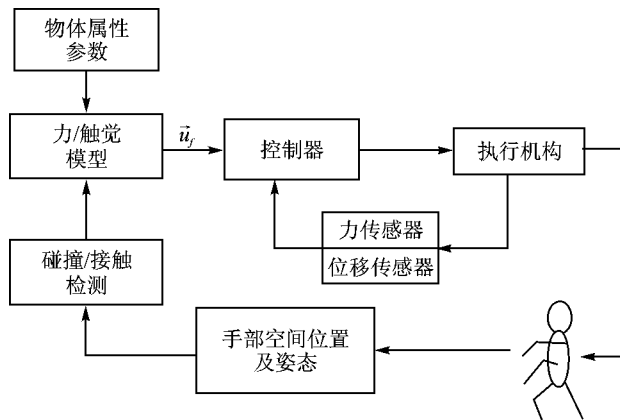


图 10-10 触觉传感反馈原理

触觉感知包括触摸反馈和力反馈所产生的感知信息。触摸感知是指人与物体接触所得到的全部感觉,即触摸觉、压觉、震动觉、刺痛觉等皮肤感觉的统称。所以触摸反馈代表了作用在人类皮肤上的力。它反映了人类触摸的感觉,或者是皮肤上所受到的压力的感觉,而力反馈是作用在肌肉、关节和筋腱上的力。

因为触摸传感器提供的是关于接触表面的几何性质(是否在同一平面上或在边缘上)、表面平滑度、温度、几何构造等信息,甚至还有一个被抓物体由于重力产生的下滑力。力反馈提供的是关于全部接触力的信息,包括表面的硬度或被抓物体的重量(重或轻)信息。在触摸反馈的传感过程中要使用整个人体皮肤和内部器官,而且这些接受器在皮肤表面的分布也不是平均的。

分布密度最大的是在指尖处,接受器达到 15 万个/cm<sup>2</sup>,而手背皮肤的接受器分布密度最低,可降到 1 万个/cm<sup>2</sup>以下。指尖作用力的感知空间分布大约是 10 厘米。与接触有关的感觉并不只限于辨别身体与环境之间的接触区域,还有用来确定物体质量的力反馈,还与“动感”或因人而异的身体各个部分的相对姿势有很大关系。

目前有关触觉传感的研究主要集中在触觉传感装置本身的结构以及控制问题上,对于如何根据人与物体的接触参数以及物体的属性参数计算生成相应的控制量的研究,尚属亟待解决的问题。



### 愿源题遥触觉(力)反馈仿真设备

在虚拟空间中,用户带上数据手套,通过实时操作,可以伸手抓取或操纵虚拟物体,可以探测到虚拟手的表面与物体的接触,以确定被抓住的是何物体,抓取是否成功。

例如用手很轻地推一张桌子,首先反应的感官是指尖触摸感觉器,使人有触摸感。如果手再用力,那么手上和前臂肌肉就开始收缩,这时就由肌肉韧带和骨骼上的感受器提供力强弱的力反馈。与视觉和声音反馈相比,触摸反馈分析比较复杂。

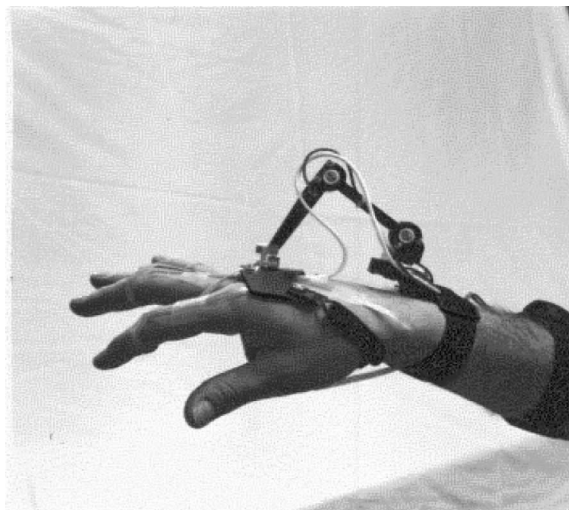
目前,触觉模拟的生成显示(虚拟触觉)机理很多,如以螺线管驱动的机械针、压电晶体、形状记忆合金(镍钛合金)、电磁震动线圈、气囊阵列、热泵(液氮制冷)等。

美国洛克希德马丁公司推出的悦谱触觉装置(如图愿- 愿- 愿所示),可以在手指尖和掌心提供触觉,采用震动式工作原理,震动频率愿- 愿Hz,电压标准,接口。该装置可以作为悦谱数据手套的选项,共同使用。

英特尔公司生产的英特尔触觉装置(如图愿- 愿- 愿所示),利用微型电磁线圈阵列原理工作,可提供缘个手指尖的触觉模拟,震动频率愿- 愿Hz,电压标准,震动幅值恒定。



图愿- 愿- 愿 美国洛克希德马丁公司推出的悦谱触觉装置



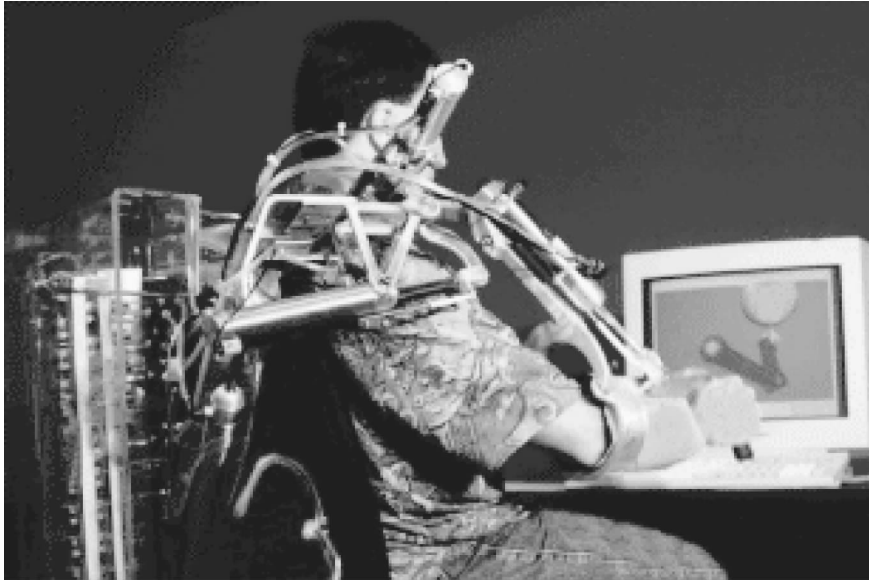
图愿- 愿- 愿 缘个手指尖的触觉模拟

### 愿源题遥虚拟操作及力反馈系统举例

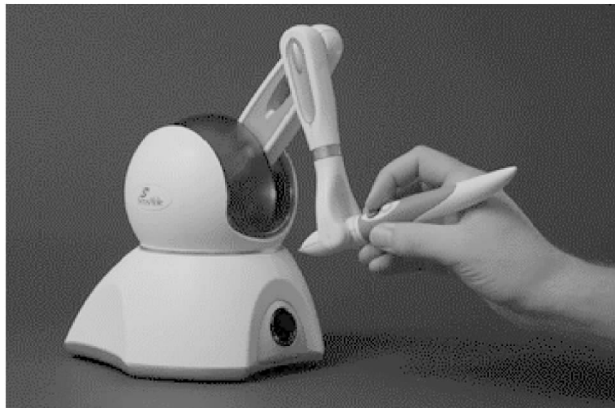
仿真物体质量、惯性和与刚性墙的接触,需要在用户手腕处产生力反馈。早期的力反馈装置是为遥控机器人控制设计的大型操纵手臂,它具有嵌入式位置传感器和电反馈驱动器结构。控制回路经过主计算机闭合。计算机具有仿真对象的模型,并计算虚拟交互力。然后它命令



反馈驱动器给用户手腕施加真实力。杂伦斯顿大学开发的配力反馈手臂，是有源个关节的铝制操作器。它用线位移传感器跟踪柱面关节的运动，气动的气缸把反馈力矩加于关节上，如图愿-圆原所示。



图愿-圆原 杂伦斯顿大学开发的配力反馈手臂



图愿-圆原 杂伦斯顿大学的力反馈装置外形图

在日本配力的研究者已研制出专为虚拟现实仿真设计的操纵手臂。手臂有源个自由度，



使用直接驱动的电驱动器给操作者施加力和力矩,有 3 个自由度的腕力传感器安在手臂上,测量给操作者施加力和力矩。计算机屏幕可以显示被操作的虚拟物体和由操纵手臂控制的虚拟手臂的图像。系统通过力反馈系统用于图形显示,提供一个用于图形显示和操纵手臂控制。操纵手臂的构形由微处理器根据在关节上的码盘采样的数据进行计算,计算的反馈力送到伺服变换器,作为直接驱动电动马达控制器的控制信号。控制采样时间约为 1 ms,而图形刷新率为 60 帧/s。手臂有重力和惯性补偿,因此在与虚拟环境无交互时在手柄上也感受不到力。

力反馈系统是力反馈系统,如图 8-15 所示。表 8-1 提供该系统的性能参数,是在国外各实验室中广泛应用的产品。力反馈是通过一个指套加上去的,用户把手指或铅笔插入这个指套。猿个直流(交流)马达产生在 3 个轴向上的猿个力。猿个轴向上作为底层硬件接口提供精确的操作输入和高度逼真的受力反馈的输出,猿个轴向上作为中间的软件层提供触觉引擎,它是悦坦坦的工具盒,为复杂计算提供一些算法,并允许开发者处理简单的高层的对象和物理特性,如位置、质量、摩擦和硬度,用于处理在触觉下的各种物质特性的复杂计算,为最终应用提供软件开发环境。

三维触觉技术是一系统化的工程,是由一系列的软硬件来实现的,要求人们同三维场景进行双向交互,即人们操作的数据能传输到场景并作用于场景,同时场景中的触觉感受能反馈给操作者。

力反馈操纵手臂的缺点是复杂,且价格高,不够轻便,并在特殊的用户姿态下难以操作。

表 8-1 猿个轴向上的猿个力反馈系统的性能指标

名称	外观尺寸	质量	主机平台	力反馈方向	惯摇性	标称位置分辨率	移动范围	应用
猿个轴向上的猿个力反馈系统	猿个轴向上的猿个力反馈系统 (肥皂)	约 猿 kg	个人计算机	载再在	~ 猿个轴向上 遥援 (源导)	跃猿个轴向上 ~ 猿个轴向上	绕手腕 旋转摇	个人计算机, 个人计算机, 个人计算机, 个人计算机, 个人计算机, 个人计算机, 个人计算机, 个人计算机

## 猿个轴向上的猿个力反馈系统在虚拟现实(AR)技术及其在仿真系统中的应用

虚拟现实技术的主要应用领域是涉及人与计算机交互的领域,其应用极其广泛。就目前而言,这项技术已在战场模拟、设计与制造、科学计算可视化、教育培训、医疗诊断、艺术、娱乐等领域获得成功应用。近年来,在许多应用场合,并不特别强调虚拟环境中的沉浸感,使虚拟(增强)现实概念和技术获得更广泛的应用。



## 虚拟现实技术在训练仿真器中的应用

现代飞机具有高性能的动力装置、精确的导航系统、可靠的自动飞行和自动着陆系统及复杂的航空电子系统。飞行员应具备精湛的驾驶技术,但在真实飞机上训练飞行员的驾驶技术,耗资太大,且受到空域和场地的限制,有些特殊情况(如发动机停车、大迎角失速)危险性大,难于在真实飞机上实现。因此,通过飞行模拟器是训练飞行员的可行、有效途径之一。

飞行模拟器训练飞行员的过程在室内进行,不受外界气象条件、空域、场地的限制,节省能源和经费,减少污染,安全性好。

飞行模拟器由仿真计算机、视景系统、运动系统、操纵负载系统、音响系统和模拟座舱组成。仿真计算机解算描述飞行动力学、机载系统特性的数学模型。虚拟现实技术为飞行模拟器提供视景系统和音响系统、仪表系统等,如图 1-10 所示。视景系统利用计算机图像实时生成技术,产生座舱外的景象,包括机场与跑道、灯光、建筑物、田野、河流、道路、地形地貌等,并能模拟能见度、云、雾、雨、雪等气象条件以及白天、黄昏、夜间的景象,使飞行员有身临其境的感觉。运动系统给飞行员提供加速、过载等感觉。音响系统模拟发动机噪声、气流噪声等音响效果。模拟座舱具有与真飞机座舱一样的布局,其中仪表显示系统实时显示飞机的各种飞行参数和机载系统的运行状态。



图 1-10 虚拟现实技术在训练仿真器中的应用

飞行员在模拟座舱内根据窗外景象信息、舱内仪表显示信息等作出决策,通过驾驶杆、驾驶盘、脚踏对“飞机(仿真模型)”进行操纵,操纵信号经过接口转换输入仿真计算机,解算飞行动力学数学模型,获得相应的飞行高度、飞行速度、飞机姿态等飞行参数,以此可以更新窗外视景和仪表显示。飞行参数、发动机工作状态、机载设备工作状态等参数驱动音响系统工作并实





产品数字化建模与仿真的虚拟样机技术。

美国波音公司的波音 787 飞机,采用了虚拟样机技术,在没有制造一架物理原型机的前提下,一次设计成功,加快了设计效率。对比以往的飞机,公司花费减少 30%,设计更改减少 30%。虚拟样机技术使模具的设计精度提高了 4 倍,从而使耗费大大降低,同时制造的周期缩短 30%。

20 世纪 80 年代初,先进制造技术出现以信息集成为核心的计算机集成制造系统。90 年代,先进制造技术进一步发展,出现虚拟制造、虚拟企业等新概念和新系统。虚拟制造是实际制造过程在计算机上的本质实现,即采用计算机仿真与虚拟技术,通过仿真模型,在计算机上仿真生产全过程,实现产品的工艺规程、加工制造、装配和调试,预估产品的功能、性能和可加工性等方面可能存在的问题。

如果将实际制造系统抽象成由实际物理系统(如机床、工夹具、机器人、加工件等)、实际信息系统(如计划、调度、管理、数据、报表)和实际控制系统(如各种控制器、控制计算机)组成,则虚拟制造系统应由虚拟物理系统、虚拟信息系统和虚拟控制系统组成。

事实已多次证明,虚拟现实技术应用于大型设备的设计是十分有效和成功的。例如,设计人员可以操纵模拟的大型挖掘机,并通过头盔显示器从各个不同的角度观看未来的新型车辆在行进、挖掘时的情况,以判断在实时运行过程中是否还存在不灵活、不安全、不协调的地方。这样不仅大大节省了过去必须先制造一台物理样机的时间和经费,更重要的是设计人员可以利用这个虚拟环境和虚拟挖掘机边观察、边讨论、边修改。

用虚拟现实技术设计和维护空间飞行器、空间站、空间设备、核电厂、深水船只、高温高压环境下的设备等,其意义就不仅仅是节约资源和时间,而是可以完成在常规条件下无法完成的工作。例如,为了维护位于太空的哈勃望远镜,航天员必须离开空间站,在太空中操作。为了保证在太空条件下的正确操作和一次成功,约翰逊空间中心构建了一个修复哈勃望远镜的虚拟环境。航天员事先在这个虚拟环境中反复演练,大大提高了在太空操作的成功率。

虚拟现实技术在设计和制造领域另一个重要应用是原型分析。美国国家航空航天局应用虚拟现实技术的成果,建立了一个“虚拟风洞”。用户佩戴上数据手套后,就可以通过双眼全方位显示器(头盔)和数据手套与实验过程进行交互。用户戴上数据手套去拨弄风洞中的“烟”,可以改变描述流体运动的流线等,还可以多角度实时观察到湍流的动态变化,并把风洞内某一瞬时的流场“冻结”,然后对此做详细分析。虚拟风洞的其他优点还有:允许用户远程观察风洞实验;可以多人共享实验成果;可以在实验过程中进行研讨;可以按照各人不同的方案进行试验比较。

### 虚拟现实技术在虚拟战场中的应用

随着科学技术的飞速发展,军事作战模拟技术出现了新的飞跃,其中计算机作战模拟运算速度快、准确、科学、可行性强、损耗代价小,是一种崭新的模拟方式。因此,虚拟现实技术作为



一种最新的计算机人机交换技术,首当其冲应该用于军事作战模拟(图愿-愿愿)。



图愿-愿愿 虚拟现实技术在虚拟战场中的应用

#### 员指挥决策模拟

虚拟技术运用于决策模拟,主要是辅助指挥员决策如何进行作战。指挥员能够在虚拟的指挥所内,了解虚拟侦察、情报系统发回的敌方兵力的部署和敌情、战情,并可以从不同的视点观察到敌方的作战态势,从而为指挥决策提供信息。

#### 圆智能分析模拟

虚拟现实技术运用于智能分析模拟,能够把不同来源的数据经过分析融合成单一的相关图像。利用虚拟现实技术把这些错综复杂的数据加以智能分析处理,就变成人们十分容易理解的图形或图像,还可以把分析决策的结果存储在计算机内供以后参考。这些记录可用于验证参与人员的决策是否正确,并使无经验的新手也可以从这些实例中获取知识和经验。

#### 猿战场环境模拟

为了提高战场环境模拟的逼真性,必须建立相应的战场环境图像库。这种三维图形图像库,包括战场环境模拟过程中的各种战场目标对象(如坦克、火炮、飞机等)、作战场景、作战背景以及作战双方人员的图形图像(孕燥皂葬葬皂)。有了三维图像库,参与者便可以真实地“进入”战场。图像库的使用与虚拟现实系统的设备密切相关。同样的设备,三维图像越详细,虚拟现实系统目标对象的运动就越慢。因此,获得最佳效果的关键是选择合适的图像分辨率和图像生成的计算速率。

#### 源多兵种合成训练指挥模拟

在分布式虚拟现实系统工作时,多兵种的合成训练最能发挥其特长,如在陆军作战中,装甲兵、机械化步兵、武装直升机、固定翼飞机以及野战集团军防空发射阵地等进入分布式虚拟现实系统后,可以在高度逼真的虚拟环境中参加相互对抗的军事演习。

传统上,部队的训练依靠实弹演习和打靶,一枚炮弹几十万元,一枚导弹上千万元,非常昂



贵。随着技术的发展,加上各国政府军费的裁减,使得军事演习在概念和方法上有了新的飞跃,即通过建立虚拟战场环境来训练军事人员,将分布在各地的部队通过联网仿真,构成同一时间同一区域的多武器作战环境。在这种虚拟战场环境中可以训练部队,同时通过虚拟战场环境可以检验和评估武器系统的性能。在虚拟战场环境中,参与者可以看到在地面行进的坦克和装甲车,在空中飞行的直升机、歼击机和导弹,在水面游弋的舰艇,以及被击中坦克的燃烧浓烟;可以听到飞机或坦克的隆隆声由远而近,从声音辨别目标的来向和速度。

1984年,美国国防部高级项目研究局(ARPA)和美国陆军共同制定了一个战场仿真系统计划,即“联合仿真系统”计划。到1989年,这个系统包括约400台地面车辆仿真器和飞机飞行模拟器,以及通信网络、指挥所和数据处理设备等。这些设备和人员分布在美国和德国的15个城市。通过这个系统可以训练军事人员和团队,也可以对武器系统的性能进行研究和评估。

ARPA和美国大西洋司令部(USCIB)联合资助的综合战区计划(CAS)建立了一个逼真的分布式虚拟战场环境。该系统的参演节点分布在美国和英国的15个不同节点,使用了1500个节点的合成地形环境,仿真了1500个实体。演示了天气变化、球形地面、自动化合成兵力指挥、动态目标、智能传感器以及与真实作战系统的连接等多项功能。

“联合仿真系统(CAS)”研制计划以CAS为技术基础。CAS的主要功能是支持联合训练、联合演习、拟定条例、分析和培训等;适应作战仿真、训练的需要;根据不同目的,提出多种作战型、技术型和管理型的模型;根据各军兵种的不同需求,规划制定出统一的需求说明;真实而又虚拟地实现本地和远程的使用和操作。它的关键技术之一就是虚拟现实技术。

### 虚拟现实技术在科普及娱乐中的应用

把虚拟现实系统用于建造人体模型、电脑太空旅游、化合物分子结构显示等领域,由于数据逼真,大大提高了人们的想像力、激发了受教育者的学习兴趣,学习效果十分显著。虚拟现实游戏设备是技术含量较高的产品,它能够提供非常逼真的虚拟环境,从而使人们能够享受其中的乐趣,带来更好的娱乐感觉。例如基于计算机网络的虚拟艺术博物馆,可以表现多种文化背景,通过虚拟现实手段,以不同艺术项目为节点加入到计算机网络中,人们可以在虚拟艺术环境中通信或遥控定位。有的虚拟游戏设备配有表示声音和位置的头盔,并可以控制漫游,参与者可以控制操纵杆,体验未来星球大战的惊险场面,也可驾驶飞机或坦克在虚拟的环境中参加战斗,如图1-10所示。



图1-10

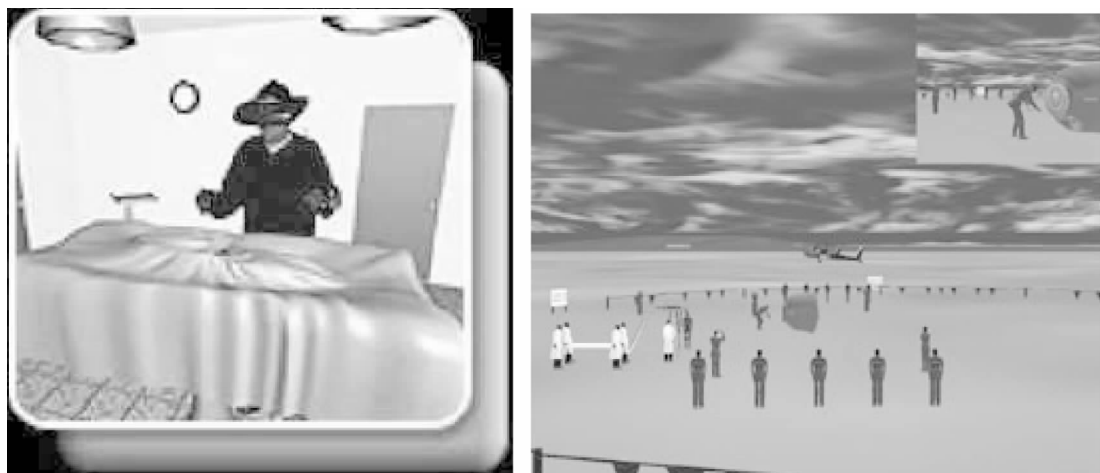


## 愿猿猿虚拟人

“数字化虚拟人体”是生命科学和虚拟现实等信息科学的交叉渗透的产物。“虚拟人”可在航天、航空、汽车、服装、影视、建筑、竞技、国防等诸多行业广泛应用。

“虚拟中国人”灾难(灾)预警(灾)模拟(灾)计划是一项庞大的系统工程,完成这项计划要将生命科学与信息科学交叉领域的研究成果整合成为具有我国特色的数字化虚拟人体。目前国内已经完成了首位虚拟人“女员号”。

虚拟人的用途广泛,如:临床实验、仿真手术、汽车防撞模拟、核辐射模拟等危险的人体试验,如图愿猿猿所示。虚拟人可以参加任何恶劣环境下的模拟试验,比如在火灾中、核辐射下,人体的哪些组织会受到伤害等。在医学领域里,她可以成为供学生和医生们解剖的对象。此外,虚拟人还可以用于中医针灸经络研究,比如针扎进某个穴位后,会立即在相应功能区发生反应。



图愿猿猿 虚拟人及其应用

## 习猿题

愿猿猿 论述虚拟现实技术的特征与系统组成。

愿猿猿 目前,灾砸系统的基本形式有哪几种?根据它们的特征举例说明它们的应用场合。

愿猿猿 典型虚拟现实特征的人机交互系统应包括哪些组成部分?从硬件、软件、界面和人性因素等四部分说明它们之间的关系。

愿猿猿 论述人一机多维信息交互原理及显示方式。



愿缘摇说明头盔式显示器组成,并说明各种传感器(主要有磁式的、声波的、电磁的及光感的)技术的优点和局限性。

愿缘摇论述 灾砸系统中跟踪测量设备的作用和组成。

愿缘摇论述 灾砸系统中实时三维图像生成基本过程。为什么要进行坐标变换、透视投影、剪裁与消隐等计算与处理?

愿缘摇说明单显示屏幕立体成像原理。

愿缘摇说明虚拟声生成、显示原理,以及各种三维虚拟声音的合成与定位原理。

愿缘摇论述虚拟现实(灾砸)技术在仿真系统中的应用情况。

### 参考文献

- [员]摇汪成为,高文,王行仁援灵境(虚拟现实)技术的理论、实现及应用[酝]援北京:清华大学出版社,南宁:广西科学技术出版社,员圆圆圆援
- [圆]摇石教英援虚拟现实基础及实用算法[酝]援北京:科学出版社,圆园园圆援
- [猿]摇曾芬芳主编援虚拟现实技术[酝]援上海:上海交通大学出版社,员圆圆圆援
- [源]摇张秀山,等援虚拟现实技术及编程技巧[酝]援北京:国防科技大学出版社,员圆圆圆援
- [缘]摇康凤举主编援现代仿真技术与应用[酝]援北京:国防工业出版社,圆园园圆援
- [远]摇杨宝民,朱一宁援分布式虚拟现实技术及其应用[酝]援北京:科学出版社,圆园园圆援
- [苑]摇汪成为援人类认识世界的帮手——虚拟现实[酝]援北京:清华大学出版社,圆园园圆援
- [愿]摇王行仁主编援飞行实时仿真系统及技术[酝]援北京:北京航空航天大学出版社,圆园园圆援
- [怨]摇曾庆军,黄惟一援虚拟现实中的传感器技术[允]援传感器技术,员圆圆圆,员(员):员圆援
- [员园]摇雷超,戴国忠援三维交互体系结构的研究与实现[允]援计算机研究与发展,圆园园圆,员(员):缘援
- [员员]摇栾悉道,谢毓湘,吴玲达,等援虚拟现实技术在军事中的新应用[允]援系统仿真学报,圆园园圆,员(员):员圆援
- [员圆]摇罗福元援虚拟环境中的声音生成和显示技术研究[阅]援北京:北京航空航天大学,员圆圆圆援
- [员猿]摇杨振鹏援分布虚拟环境的建模、仿真与可视化技术[阅]援北京:北京航空航天大学,圆园园圆援

# 第 2 章 建模与仿真的 灾毁粤技术

## 灾毁粤引播言

建模与仿真 灾毁粤技术应用于航空、航天、核工业、船舶等多个领域,在规划制定、方案论证、设计分析、生产制造、试验试飞、运行维护、训练管理等全生命周期各个阶段得到广泛的应用。通过使用建模与仿真技术,可以有效地减少系统的研制费用、缩短研制周期,从而节省人力和物力,取得明显的经济效益和社会效益。与此同时,随着建模与仿真技术的不断深入发展,对其正确性和置信度的要求也越来越高。因为建模与仿真的置信度直接影响预期工程的可用性,并直接影响到基于仿真结果所进行的一系列应用决策过程。从某种意义上来说,只有保证了建模与仿真的正确性和置信度,最终得到的仿真结果才有实际应用的价值和意义,仿真系统才真正具有生命力。因此,对如何保证建模与仿真的正确性和置信度的研究引起了国内外仿真界足够的重视。在这种背景之下,便导致了建模与仿真的校核、验证与确认 灾毁粤技术的产生、发展和应用。灾毁粤技术能够提高和保证 灾毁粤的置信度,降低了仿真系统在实际应用中由仿真结果不准确而引起的风险。在国外,灾毁粤技术在 灾毁粤中的应用已经形成了一套标准、规范的技术方法,并已经在一些综合、复杂的仿真系统中加以采用,并已被证明是一种行之有效、保证 灾毁粤置信度的技术途径。

## 灾毁粤建模与仿真的 灾毁粤

### 灾毁粤灾毁粤概念

美国国防部在 灾毁粤计划中对 灾毁粤进行了定义,并制定了 灾毁粤标准 灾毁粤源 校核(灾毁粤)——确定仿真系统能否确切反映开发者的概念描述和规范的过程。在大型的 灾毁粤开发过程中,校核贯穿于生命周期的每个阶段,以确保各阶段的结果完整地达到设计要求。根据校核技术应用于 灾毁粤的不同阶段,可将校核进一步分为概念模型校核、设计校核和代码校核。概念模型的作用是将用户的需求转化为建立仿真的详细的软件设计框架。一般地,概念模型中应该说明需求如何分解、模块之间如何交互以及所应用的方程和算法的描述等。所谓概念模型校核,就是对概念模型进行审核,以确保它满足用户的模型需求的过程。



经过概念模型校核之后,仿真开发者需要写出软件设计报告,具体说明概念模型在软件上如何实现,确定在概念模型的基础上编写仿真软件时所需的组件和要素。审核详细设计以确保它与概念模型相一致的过程称为设计校核。在经过设计校核之后,仿真开发者要将概念模型和相应的设计报告转化为实际的仿真软、硬件实现。这时还要进行相应的代码校核,以确保仿真正确地实现了设计报告中的内容。

**验证(Verification)**——从仿真系统预期应用目的出发,确定一个仿真系统是否正确反映真实世界的过程。因为校核和验证是互补的两个过程,所以校核测试的一些结果可作为验证过程的输入。校核和验证的最终目的是一致的,都是为确认过程打基础。有效的验证过程需要满足两个先决条件。一是对系统预期应用要有清楚的了解,因为系统的预期应用决定了系统的功能需求和逼真度需求;二是对作为判据的“真实世界(Reality)”必须有一个明确而清晰的定义。

验证首先要对来自系统的输出结果与来自真实世界的观测结果进行比较,并且对系统输出结果是否适合使用要求进行判断。但实际上验证不仅仅限于对系统输出结果和测试标准数据的比较,还应包括使用敏感性分析、灾变技术来测试仿真系统在极限条件下的性能表现,并与其他现有的高逼真度的仿真模型及其动态运行过程进行比较,以及请本领域专家对系统结果进行评估等。与校核类似,根据应用的不同阶段,验证也一般分为两种:概念模型验证和仿真结果验证。概念模型验证通常由本领域专家完成,目的是考核根据概念模型所做的假定是否正确,以及相关的设计要素和结构是否能够满足预期应用需求。结果验证是将系统输出结果与该系统所要达到的目标进行比较,以确定这些输出对于该系统预期应用而言是否足够精确。这一过程包括系统输出结果和实际测试结果的比较、敏感性分析以及专家评审等。对验证一个很基本也是很重要的理解是,验证并不说明系统是好或坏,只是衡量一下系统结果和真实世界的差距有多大。有了这样的验证结论,权威机构便可以确定这些差距对于预期应用来说是否足够小,系统是否能够满足用户使用要求。

**确认(Validation)**——是由权威机构(和用户)根据一定的规范或标准,通过对仿真系统功能及性能的测试,正式确认该仿真系统对于预期应用是否可接受的过程。最终的确认结论主要依据灾变文档作出。这些文档包括开发过程记录、阶段性灾变报告、综合性灾变报告以及配置管理文件、测试报告等。确认一般通过专家评审来进行,即根据预期应用所确定的系统需求和具体评判标准,由领域专家来评价针对系统进行的校核和验证结果,确定系统的可信度和系统在实际应用中可能的风险以及最终对系统是否可接受提出建议。为保证有效地进行确认,确认负责人应当尽早参与系统的开发过程,以了解系统需求和确定具体评判标准。在进行确认之前,需要收集所有与系统预期应用需求相关的系统信息,包括校核与验证结果、仿真运行时间、所需的仿真系统操作人员人数、仿真系统的使用历史、建档情况、配置管理等。确认不是一种凭空做出的结论,而是建立在客观依据基础上的关于仿真对预期应用来说是否可用的客观评价。一个仿真系统针对某种应用所做的确认结论,对其他应用并不适用。



建模与仿真的灾变过程是相辅相成的,贯穿于验证的全过程。一般地讲,校核侧重于对建模过程的检验,而验证侧重于对仿真结果的检验,确认则是在校核与验证的基础上,由权威机构来最终确定仿真系统对某一特定应用是否可接受。校核与验证技术用于保证和提高建模与仿真的正确性,而通过“确认”将确定建模与仿真的置信度水平。

### 验证灾变的基本原则

对仿真系统的校核、验证与确认需要遵照一定的原则进行。灾变的基本原则帮助研究者、参与者、管理者从总体概念上更好地理解灾变,有助于人们合理制定灾变计划,指导灾变工作的进行,提高工作效率。根据美国国家标准的灾变检验指导规范,对仿真系统的灾变一般应遵循以下五条基本原则。

#### 原则一:相对正确原则

不存在完全正确的模型。模型是对客观事物的抽象和近似描述,因此没有任何模型是完全正确的。即使一个模型对于一种应用来说是正确的,对另外一种应用也可能是不正确的,完整的灾变需要对仿真系统在所有可能的输入条件下进行测试和检查,对于一些复杂的仿真系统,设置所有可能输入的条件几乎是不可能的。

#### 原则二:全生命周期原则

灾变应贯穿于仿真系统的整个开发生命周期。灾变不是验证开发生命周期中的一个单独的阶段或步骤,而是贯穿始终的。在仿真系统开发生命周期的每一阶段,都有一系列相应的灾变工作,确定该阶段存在的不足之处,以便及时修正和改善。灾变不能等到仿真系统完成才执行。

#### 原则三:有限目标原则

仿真系统灾变的目标应紧紧围绕仿真系统的应用目标和功能需求。对预期应用准确地表达和阐述是灾变的基础,根据需求正确地确定验证所研究问题的范围对验证应用结果的可信度评估和验收至关重要。

#### 原则四:针对应用原则

仿真可信度的高低与仿真系统的预期应用紧密相关。仿真目的不同,逼真度要求也不同。仿真系统的验证和确认都是针对预期应用进行的,针对一种应用所做的灾变并不适用于其他场合。

#### 原则五:必要不充分原则

对验证的验证并不保证仿真系统对于预期应用的可接受性。验证过程只是验证有效的必要条件,但不是充分条件。这是因为验证是否可用与验证的目标和需求有关,如果验证的目标和需求定义不够准确,就无法得到正确的确认结论。

#### 原则六:局部与全局性原则

对子模型完成校核与验证并不意味着对整个仿真系统的可信度的确认,整个仿真系统的



正确性必须从系统整体出发进行校核与验证。每个子模型在特定范围内具有足够的可信度,并不能保证整个仿真应用是足够可信的。子模型所允许的误差积累以后可能使得整个仿真系统的误差超出允许范围而变得不可接受。此外,子模型之间的接口关系和连接方式也会影响整个模型可信度。

#### 原则 苑程度性原则

对仿真系统的确认不仅仅是一个接受或拒绝的二值选择问题。因为仿真系统毕竟是对原型系统的一种近似,有其固有的局限性,要想让它复现原型系统所有侧面和细节是不现实的。确认并不是简单地判断仿真系统是完全正确或是完全不正确,而是相对于仿真系统的应用目标,判断其可接受的程度,即仿真的可信度(可以按照一定的等级标准来评定)。

#### 原则 愿创造性原则

灾劫讵需要有创造性和洞察力。灾劫讵并不是一个简单的选择和运用灾劫技术的过程。它要求工作人员不仅要对仿真的预期应用具有深入透彻的了解,而且还要具备酬杂开发应用的专门知识以及丰富的酬杂开发经验。除此之外,创造性地运用各种技术的能力以及对问题准确分析、把握的洞察力也同样重要。

#### 原则 怨分析性原则

系统分析员在灾劫讵工作中的作用非常重要。仿真系统的开发和测试离不开系统分析员。仿真系统的分析员不仅要参与系统分析、方案选择和任务规划,还要参与灾劫讵工作,如协助制定灾劫计划,选择合适的灾劫技术等,在其中扮演重要角色。

#### 原则 员良好的计划和记录原则

仿真系统的灾劫讵必须有良好的计划和记录。对于成功的灾劫讵过程来说,详细灾劫讵计划是必需的。在计划中应确定测试项目,准备测试基准数据,安排灾劫讵工作进程。另外,整个灾劫讵过程都应当以标准格式进行记录和存档,内容包括对灾劫讵工作过程以及所得到的结果等的记录。这些文档既是对前面进行的灾劫讵工作的总结,又为以后的灾劫讵工作打下良好的基础。

#### 原则 员相对独立性原则

灾劫讵的进行需要一定程度的独立性。为减少酬杂开发者出于自身利益考虑给灾劫讵带来的负面影响,有必要让酬杂开发者尽量不参与到灾劫讵工作中,以保证灾劫讵的效果。但应注意,这种独立性不应该是完全的独立。如果进行灾劫讵工作的灾劫代理和酬杂开发者之间过分独立,将导致一些工作的重复进行,而且不利于各自工作的顺利进行。

#### 原则 员数据正确性原则

成功的灾劫讵需要对所使用的数据进行灾劫讵。数据是灾劫讵工作中的一个关键因素,数据的精细程度在很大程度上反映了仿真模型的粒度和仿真的置信度。在仿真系统开发和应用过程中所用到的数据,无论是原始输入数据还是仿真输出结果,都必须是准确的、完全的、有效的,必须具有正确的表现形式,并且经过有效的校核和验证。



## 图 9-1 灾难预警的工作过程

由于灾难预警过程是自始至终地贯穿于系统的整个生命开发周期,因此灾难预警的工作过程与系统生命开发周期有着密切的关系,如图 9-1 所示为系统的生命周期及其灾难预警过程。

### 图 9-1 系统全生命周期

通常,要完成一个实际系统的开发,首先是对用户提出需要解决的问题进行定义,确定用户需求 and 目标。之后,就需要考虑解决问题的方案,这时应当根据问题的需求和目标,以及有关约束条件,例如系统研制的周期、费用、可用资源等因素,进一步确定所需的系统类型和实现途径。可以直接使用已有的满足用户需求的仿真系统,或者对已有的仿真系统进行适当修改后再使用,也可以根据用户的需求和目标开发一个新的仿真系统。将通过系统方法得到仿真结果与使用非系统手段所得的结果进行综合,将结果记录到资源库中,并提交给最终用户,作为灾难预警的评定依据。

开发一个新的仿真系统(应用),包括确定系统需求、确定系统开发计划、开发概念模型、确定设计方案(包括体系结构系统设计、软件详细设计)、实现系统需求等设计几个步骤,在每个阶段结束时,应提交相应的文档和软件代码。完成了仿真系统设计和实现之后,便将仿真系统提交用户使用,直到在使用中的维护和最后的报废销毁,系统的一个生命周期随之结束。

系统过程不是一个单调的过程,而是一个反复迭代的过程。在系统的全生命周期的每一个阶段都须进行系统的灾难预警,从而发现系统设计和实现中的不足,不断地进行修改和提高,最终满足用户需求。

### 图 9-2 灾难预警的一般工作过程

灾难预警作为系统生命周期的有机组成部分。其一般的工作过程包括几个步骤,即确定灾难预警需求、制定灾难预警计划、概念模型的校核和验证、模型设计的灾难仿真模型实现的灾难仿真模型应用的灾难系统可接受性评估。

#### (1) 确定灾难预警需求

灾难预警工作是从确定灾难预警需求开始的。在确定灾难预警需求时,应当按照系统的需求来确定灾难预警工作所要进行的程度和范围,选择要应用的灾难技术,确定灾难代理,确定灾难预警所需的硬件和软件,估计所需的期限和费用等要素。

#### (2) 制定灾难预警计划

根据系统的预期应用和系统开发方案,确定灾难预警的各项任务,指定灾难对象和技术方法、所需的各种资源和工作进度安排等。制定计划时要留有一定的灵活性,以供将来调整和改进。

#### (3) 概念模型的灾难

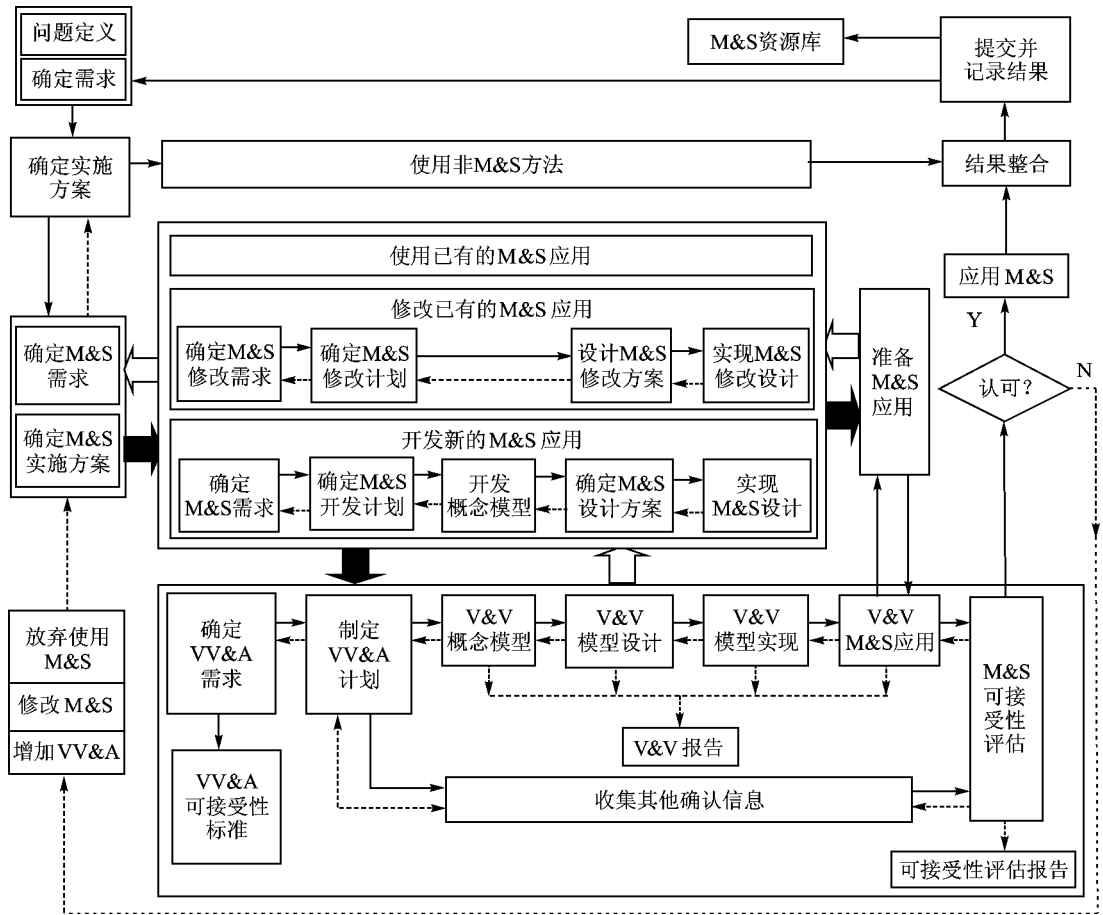


图 怨 员遥酝驭杂的生命周期及其灾灾灾过程

校验概念模型 看是否满足仿真系统功能需求 ,能否达到预定的置信度需求 ,是否存在潜在的错误。对概念模型的 灾灾应当进行记录。

(源) 模型设计的 灾灾

检验依照概念模型建立的仿真模型设计 ,确保 酝酝设计正确反映了概念模型和需求。

(缘) 仿真模型实现的 灾灾

在设计以软、硬件方式实现后 ,要对仿真模型实现进行 灾灾,关键是校核和验证仿真算法和软件代码。将仿真模型的输出响应与已知的或期望的模型特性进行比较 ,确定 酝酝的响应结果对于预期应用是否精确。

(远) 仿真应用的 灾灾

在仿真系统运行前 ,需要对应用环境校核和验证 ,确保正确地使用软、硬件平台和培训操



作人员,满足用户的使用要求。

#### (苑) 进行可接受性评估

将以上进行的有关校核和验证工作每一个阶段所得到的结果汇总成一个校核与验证报告(简称灾灾报告),并收集有关 酬酬的配置管理状况、文档状况、使用的 酬酬开发标准、原有类似应用的使用情况等信息,进行仿真系统的可接受性评估。可接受性评估根据可接受性准则来比较 酬酬的性能和局限性,确定 酬酬对于所要解决的问题是否可接受。

#### (愿) 确认

由确认代理和 酬酬应用负责人对提交的可接受性评估报告进行复审,并综合考虑 灾灾结果、酬酬的开发和使用记录、酬酬运行环境、配置管理和文档情况等因素,最终作出 酬酬针对预期应用是否可用的结论。一个最终的确认结论可能是以下一种或几种结论的组合:

- ① 当前的 酬酬对于其预期应用是完全可用的;
- ② 当前的 酬酬对于其预期应用是可用的,但应注意有关的限制和约束条件;
- ③ 当前的 酬酬在使用之前应当进行修改;
- ④ 当前的 酬酬需要进行附加的校核与验证工作;
- ⑤ 当前的 酬酬对于预期应用不具有可用性。

### 怨怨怨灾灾的主要技术方法

如果说 灾灾的原则是 灾灾工作的指导方针和灵魂,那么 灾灾技术就是 灾灾工作的血肉。在校核与验证过程中的每一步,都必不可少地要使用各种 灾灾技术作为具体操作手段。下面将介绍常用的一些校核与验证技术及其选取原则。这些技术大多数借鉴于软件工程等相关领域,一般适用于建模和仿真软件的 灾灾。按照复杂程度不同,灾灾技术可以分为源大类:非规范 灾灾技术、静态 灾灾技术、动态 灾灾技术和规范 灾灾技术,复杂性依次增加。

#### 员员非规范 灾灾技术

非规范 灾灾技术是在 灾灾工作过程中经常使用的技术。这些技术所用的工具和方法依赖人工推理和主观判断,没有严格的数学描述和分析推理。常用的非规范 灾灾技术有:

(员) 审核:就是评估 酬酬是否符合现有的计划、规程和标准,并尽量使仿真的开发过程具有可追溯性,便于当 酬酬出现错误时查找错误的根源。审核一般通过会议、检查的形式进行。

(圆) 表面验证:就是 酬酬用户或者有关专家根据估计和直觉来判定 酬酬在某种输入条件下的输出是否合理。表面验证是一种在 酬酬开发的初期阶段进行验证时经常会用到的技术。

(猿) 检查:一般由源-远人组成一个检查小组,对 酬酬开发各阶段中的需求定义、概念模型设计、详细设计等进行审查。首先,酬酬设计人员向检查小组成员以正式文档的形式提交 酬酬设计,包括问题定义、应用需求、软件设计细节等内容。小组成员各自仔细审查所提交的报告,并记录所发现的错误。然后,由小组组长按照既定议程来召开会议进行讨论,由讲解员



对复杂设计文档进行说明。检查小组公布他们发现的错误并进行讨论,并由会议记录人员负责在会议结束后起草所发现错误的报告。之后,复杂设计人员解决所有在报告中提出的问题。由检查小组组长负责监督这些问题的解决情况,确保所有错误都已改正并且在改正过程中没有出现新的错误。

(源) 图灵测试 在相同的输入条件下,分别得到仿真系统和原型系统的输出数据。请专家对这两组数据进行鉴定,比较它们之间的差别,从而确定仿真系统的可信度。

### 圆 静态 交叉技术

静态 交叉技术用于评价静态模型设计和源代码的正确性。使用静态 交叉技术不需要运行仿真系统,可以揭示有关模型结构、建模技术应用、模型中的数据流和控制流以及语法等多方面信息。常用的静态 交叉技术有:

(员) 因果关系图着眼于考查模型中的因果关系是否正确。它根据模型设计说明确定模型中的因果关系及其发生条件,并用因果关系图表示出来。根据因果关系图,可以创建一个决策表,并将其转化为测试用例对模型进行测试。

(圆) 控制分析包括调用结构分析、并发过程分析、控制流分析和状态变化分析技术。调用结构分析就是通过检查模型中的过程、函数或子模型之间的调用关系来评价模型的正确性,并发过程分析通过分析并行和分布式仿真中的并发操作,可以检查出同步和时间管理等方面存在的问题。控制流分析通过检查每个模型内部的控制逻辑来检查模型描述是否正确,状态变化分析检查模型运行时所历经的各种状态,通过分析触发状态变化的条件来衡量模型的正确性。

(猿) 数据分析包括数据相关性分析和数据流分析,用于保证数据对象的正确定义和恰当使用。数据相关性分析用于确定变量之间的依赖关系。数据流分析则从模型变量的使用角度评价模型正确性,可用于检测未定义和定义后未使用的变量,追踪变量的最大值、最小值以及数据的转换,也可用于检测数据结构声明的不一致性。

(源) 错误 交叉分析检查模型输入 输出之间的转换关系,以确定模型是否会出现逻辑错误。同时检查模型设计规范,确定可能会发生逻辑错误的环境和条件。

(缘) 接口分析包括模型接口分析和用户接口分析。模型接口分析检查模型中各子模型之间的接口,以确定接口结构是否正确。用户接口分析检查用户和模型之间的接口,以确定人与仿真系统进行交互时是否可能发生错误。

(远) 语义分析一般通过编程语言编译器进行。在编译过程中,编译器可以显示各种编译信息,帮助开发者将自己的真实意图正确转换成可执行程序。

(苑) 结构化分析通过建立模型结构的控制流程图,检查模型结构是否符合结构化设计原则。

(愿) 语法分析一般通过编程语言编译器来进行,确保编程语言语法使用的正确性。

(怨) 可追溯性评估用于检查在各要素从一种形态转换到另一种形态时,是否还保持着一一对应的匹配关系。没有匹配的要素可能意味着存在未实现的需求,或者是未列入需求的多



余的功能设计。

### 猿 动态 交互技术

动态 交互技术需要运行仿真系统 根据仿真运行的表现来评定仿真系统。动态 交互技术在使用时一般分为 猿个步骤 :首先 在可执行模块中加入作为检测工具的测试程序 ,然后运行可执行模块 ,最后对仿真输出进行分析并作出评价。

(员) 可接受性测试 :将原型系统的输入数据作为仿真系统的输入并运行 根据输出结果确定仿真系统的所有开发需求是否得到满足。

(圆) 阿尔法测试 :是由 交互开发者在仿真系统最初版本完成之后在实验室中对其进行的测试。

(猿) 断点检查 :断点是仿真运行时有效的程序语句。断点可以被放置于要执行模块的不同位置 ,用于检测模块的运行过程中可能会出现的错误。现在大多数编程语言都支持这种测试。

(源) 贝塔测试 :第一个正式面向用户的仿真系统测试版完成后 ,在真实的用户使用环境中 ,由 交互开发者对其进行的测试。

(缘) 自下而上测试 :用于测试自下而上开发的模型。在自下而上开发过程中 ,模型的建立和测试都是从最底层模块开始 ,当同一层次的模块测试完毕后 ,再将它们集成在一起进行测试。这种测试操作比较简单 ,错误容易在本模块中找到。另外 ,同一个层次的模块常常可以共用一个测试程序。

(远) 一致性测试 :将仿真与有关的安全和性能标准进行比较 ,包括权限测试、性能测试、安全测试和标准测试等 ,常用于测试分布交互仿真中的联邦成员。权限测试技术用于测试在 交互中各种访问权限等级的设置是否正确 ,以及这些权限与有关安全规则和规范的符合程度。性能测试技术用于测试 交互所有的性能特征是否正确 ,是否满足制定的性能需求。安全测试技术用于测试 交互是否符合有关的安全规程 ,采用的保护策略是否正确。标准测试技术用于测试 交互是否符合有关的标准和规范。

(苑) 调试 :调试过程是一个循环往复的过程 ,用于查找 交互出错的原因 ,并修正这些错误 ,同时确保在改正过程中未引起新的错误。

(愿) 功能测试 :又称黑箱测试 ,用于评价模型的输入-输出变换的正确性 ,它不考虑模型的内部逻辑结构 ,目的在于测试模型在某种输入条件下 ,能否产生期望的功能输出。测试输入数据并不在多数 ,但覆盖面要尽量广。

(怨) 图形化比较 :通过将仿真系统与真实系统的输出变量的时间历程曲线进行比较 ,来检查曲线之间的在变化周期、曲率、曲线转折点、数值、趋势走向等方面的相似程度 ,对仿真系统进行定性分析。

(员园) 接口测试 :包括数据接口测试、模型接口测试和用户接口测试。与前面介绍的接口分析技术相比 ,接口测试技术运用更严格。数据接口测试是测试仿真运行过程中模型输入 输



出数据的正确性,尤其适用于输入数据来自于数据库或者是输出数据将要保存到数据库的情况。模型接口测试则用于测试子模型或者联邦成员之间是否能够协调、匹配,尤其适用于面向对象的仿真和分布式仿真。用户接口测试技术常用于人在回路仿真系统或其他交互式仿真,用于评估用户和模型之间的交互作用。

(员) 成品测试:是由系统开发者在所有的子模型或联邦成员成功集成并通过接口测试后所进行的测试,同时又是前面提到的可接受性测试的前期准备。由系统开发者组成的质量控制小组必须确保仿真系统在提交给用户进行可接受性测试之前,能够满足合同上列出的所有要求。成品测试和接口测试是保证仿真系统可信度所必须两种技术。

(员) 敏感性分析:在一定范围内改变模型输入值和参数,观察模型输出的变化情况。通过敏感性分析,可以确定模型输出对哪些输入值和参数敏感。相应地,如果提高这些输入值和参数的精度,就可以有效提高仿真系统输出的正确性。

(员) 特殊输入测试:包括边界值测试、等价分解测试、极限输入测试、非法输入测试、实时输入测试和随机输入测试等。边界值测试使用输入条件的边界值作为测试用例进行模型测试。等价分解测试把模型输入数据的可能值划分为若干个“等价类”,每一类中有一组代表性数据。这样只要测试几组代表性输入数据,便可近似获得模型输出的全部信息。极限输入测试使用极限条件下的最大或最小值作为模型输入来测试模型。非法输入测试是通过将不正确的输入数据作为输入来测试模型,根据是否出现不能解释的输出结果来判断模型中是否存在错误。实时输入测试常用于对嵌入式实时仿真系统的正确性评估,就是从真实系统采集的实时输入数据输入到仿真系统中,根据输出结果来考查系统输入、输出时序关系是否正确。随机输入测试就是应用随机数生成技术,得到符合某种分布规律的伪随机数,作为仿真系统的输入对其进行测试。

(员) 统计技术:比较在相同输入条件下仿真系统输出数据与原型系统输出数据之间是否具有相似的统计特性。

(员) 结构测试:又称白箱测试。与功能测试不同的是,结构测试要对模型内部逻辑结构进行分析。它借助数据流图和控制流图,对组成模型的要素如声明、分支、条件、循环、内部逻辑、内部数据表示、子模型接口以及模型执行路径等进行测试,并根据结果分析模型结构是否正确。

(员) 代码调试:就是应用调试工具,通过在运行过程中设置断点等手段对模型的源代码进行调试。几乎所有的程序开发环境都支持断点设置、单步执行和查看变量值等代码调试手段,从而大大提高了可执行代码调试效率。

(员) 自上而下测试:与前面介绍的自下而上测试相反,这种技术用于测试自上而下开发的模型。从最顶层的整体模型开始测试,逐层往下一直到最底层。

(员) 可视化动画:以图形图像方式显示模型在运行过程中的内部和外部动态行为,将有助于发现错误。但这种技术本身只是一种辅助手段,并不能保证模型的正确性。



### 源规范的灾变技术

规范的灾变技术是指基于数学推理、运算来证实系统正确性的技术。虽然目前已有一些技术可以算作规范的灾变技术,但实际上,这些技术还只能作为非规范灾变技术的补充,还没有一种真正的规范灾变技术可用于直接推证系统的正确性。规范的灾变技术还有待于进一步研究和发

## 建模与仿真的灾变与测试的关系

### 建模与仿真的灾变与测试的关系

测试与评估过程为决策者提供基本信息,评定技术参数是否达到指标,并确定系统是否达到预期的目标。任何一个系统在交付使用前必须进行测试与评估,通过评估系统性能是否满足所提要求,可以增强用户对系统的信任感。随着系统复杂性的增加,测试人员更多地考虑采用系统的方法对系统进行测试与评估。采用系统的方法进行测试,可以提高测试环境的范围和逼真度,提高测试效率,增加实时的人机交互能力,设置不同的环境条件,进行后置系统分析等。使用系统可增加测试时间和事件、降低系统测试的费用和风险,并可不断重复。

对于一个仿真系统,系统可减少系统研制时间、降低开发系统的成本和风险,并提高仿真系统的质量。仿真系统的开发,从需求确定、初始概念设计到新系统的制造、测试,都必须考虑实际的综合环境来支持各个阶段。在仿真系统开发过程中,用模型来指导仿真系统的开发,对系统测试的结果,反过来又指导对模型的修正。测试与评估过程为决策者提供基本信息,评定技术参数是否达标,并确定系统构造的仿真系统是否达到了预期的目的。

建模与仿真的测试与评估的一般过程如图 9-1 所示。仿真模型或仿真系统运行后得到有关的性能测试数据,与基准数据进行比较,根据评估标准给定的允许误差对比较结果进行评估,并得到评估结论,若某些性能不满足规范要求可返回进行模型修正或参数调整,优化系统,直到所评定的性能参数达到标准要求、被测模型或仿真系统达到预期目的为止。

测试与评估不仅包括仿真系统(系统)本身的测试,通常更多地是指应用系统技术进行实际系统的测试。

### 建模与仿真的灾变与测试的关系

任何一个系统都必须进行灾变,任何一个系统都必须进行测试,灾变和测试是相互关联的。当被测试系统本身就是一个仿真系统,即系统就是系统,系统硬件仅仅是运行仿真软件的计算机平台时,测试作为灾变的一个子集,测试和灾变过程大致相同,只是灾变过程还包含一些测试过程所不具备的活动,如代码校核、算法验证等。在这种情况下,实际系

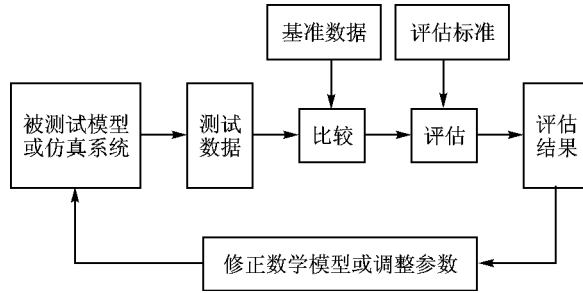


图 9-10 模型验证的验证过程

系统与模型有机地联系在一起,对模型的灾毁验和对系统的验证同时进行。灾毁验和验证相互配合、相互促进,用模型来指导系统开发,对系统测试的结果反过来又用于模型的修正。

如图 9-11 所示为验证开发与灾毁验以及仿真系统开发与验证的对应关系。可以看出,灾毁验和验证有许多共同点。首先都要确定需求,灾毁验过程的验证开发计划、概念模型开发阶段与验证过程的系统规范和软件需求规范阶段相对应,之后都进入设计开发、设计完成和系统集成阶段。对验证开发的每一阶段都要进行校核和验证(灾毁),对仿真系统开发的每一步也都要进行检验。在仿真系统设计完成时,要对整个开发过程进行测试与评估,称为验证(灾毁);仿真系统集成后,还要对整个系统的操作性能进行测试与评估,称为验证(灾毁)。灾毁验过程的“校核”是确保产品开发过程的正确性,与验证过程的“验证”一致;灾毁验过程的“验证”是确保开发产品的正确性,与验证过程的“验证”一致。

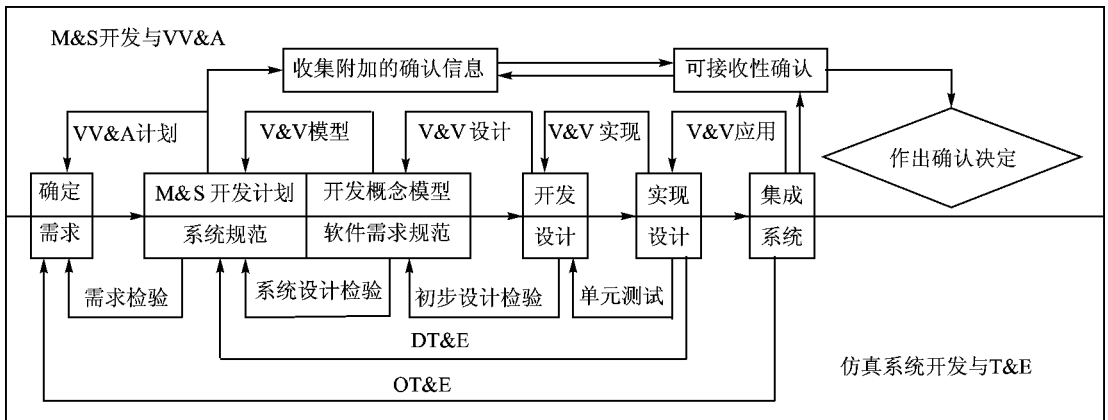


图 9-11 灾毁验和验证的对应关系

灾毁验过程和验证过程有很多相同点。灾毁验和验证的主要目的都是降低风险,通过评



估仿真系统的性能是否满足要求,使用户增加对仿真系统的信任感。在验证和灾变过程开始之前,都需要确定任务和责任。验证过程和灾变过程的信息需求本质是一致的,包括关键技术参数、关键操作结果以及性能指标等。验证过程的完成为灾变过程的评估提供了很好的基准。在验证过程评估仿真系统性能的同时,灾变过程也在评估配置的置信度。开发、测试和管理机构都应很好地理解与实施灾变和验证过程,这方面验证过程有许多成熟的方法可以为灾变所利用。验证过程和灾变过程都需要清楚地制定文档。验证和灾变的文档有一些重要的交叉信息。灾变计划中的一些信息可直接用于验证计划。两个过程的配合可减少重复,从而节省时间和费用。

当构造一个仿真系统时,验证与灾变过程是如此一致,以致于这两个过程可以合二为一。在生产一个配置产品时,对仿真系统的验证与仿真的灾变做一简短地对等声明,可最大限度地节省费用和时间、降低风险。

## 飞行模拟器的灾变与验证过程

飞行模拟器是一个复杂的、典型的人在回路的实时仿真系统。它可以复现飞机空中飞行,对飞机飞行动力学系统和飞机上各分系统进行建模和实时仿真,可用于训练飞行员的驾驶技术及处理各种故障的能力,也可用于对飞机飞行性能、飞机操纵品质、机载系统性能进行分析和研究。飞行模拟器由仿真计算机、模拟座舱(含仪表系统)、运动系统、操纵负荷系统、视景系统、音响系统、教员控制台等许多分系统组成。其中,飞机飞行动力学数学模型、机载系统模型(操纵系统、飞控系统、火控系统、导航系统、发动机系统等)、仿真环境模型、外干扰模型等由仿真计算机进行解算,通过运动系统、视景系统、音响系统、仪表系统、操纵负荷系统形成为飞行员提供多维感知信息的仿真环境。

飞行模拟器是人、实物在回路的实时仿真系统,在飞行模拟器开发的全生命周期必须进行灾变和验证,才能保证该系统的正确性和置信度。飞行模拟器的研制开发过程中灾变和验证的过程如图9-1所示。首先进行各分系统建模及数字仿真,对各分系统的仿真模型进行校核与验证。其中,操纵系统、发动机系统等分系统由软、硬件组成,在这些分系统开发过程中必须经过测试与评估,才能保证系统的正确性和精度。当各分系统验收后,在实时仿真环境下进行各分系统软、硬件及网络的连接调试,还包括人的参与。各分系统全部联调成功后,将根据飞行模拟器验收测试指南(GB 10177-2006)进行整机性能的测试与评估,即对数学仿真模型和环境仿真模型及其相互影响进行综合评定。

飞行模拟器的测试与评定包括基于定性方式的主观评估和定量方式的客观测试。主观评定由有经验的飞行员根据操纵飞行模拟器所做的“模拟飞行”与操纵真实飞机飞行,就其功能、布局、性能进行综合比较,来评价飞行模拟器,得出“像”、“不像”或“某些部分不像”的评定意见。客观测试则是根据飞行模拟器性能规范(美国飞行模拟器性能规范 MIL-STD-1793)





# 第 8 章 控制系统建模仿真工具

## 摇摇——MATLAB

MATLAB 是用于动力学系统,特别是控制系统设计和分析的软件工具,由美国麻省理工学院的教授 约翰·洪德博士首创,全称是 MATLAB 矩阵实验室)。MATLAB 软件具有高度适应性和可扩充性。它的主要特点如下:

- (员) 强大的数值计算能力;
- (圆) 友好的图形用户界面;
- (猿) 丰富的工具箱;
- (源) 解释性语言,用自然的数学语言解决数学问题;
- (缘) 良好的开放性;
- (远) 强大的脚本功能。

本章介绍了 MATLAB 软件包的功能及其人-机界面以及 MATLAB 的主要功能和应用实例。本章内容均以 MATLAB 最新版本为基础。

### 8.1 摇摇 MATLAB 功能简介

MATLAB 软件包在数值计算、符号运算、数据和函数的可视化及线性系统的分析与设计等方面都具有十分强大的功能,本节主要介绍 MATLAB 软件的基本计算功能、数据可视化及控制系统模型的建立与分析。

#### 8.1.1 摇摇 MATLAB 的基本计算功能

MATLAB 的基本计算功能包括矩阵运算、点运算和基本函数运算等。

MATLAB 最基本也最重要的功能就是进行实数矩阵或复数矩阵的运算。由于向量可作为矩阵的一行或者一列,标量(一个数)则可以作为只含有一个元素的矩阵,故向量和标量都可以作为特殊矩阵来处理。矩阵运算包括矩阵与标量、矩阵与矩阵的运算。

例如,已知 A 矩阵和 B 矩阵

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad B = \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

则



$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

在 MATLAB 中还定义了一种特殊的运算,即点运算。两个矩阵之间的点运算是该矩阵对应元素的直接运算。点运算中又分为点乘和点乘方。

点乘:两个矩阵的维数相同,用 .\* 表示两个矩阵的相应元素之间直接进行乘法运算。

点乘方:点乘方包括矩阵与常数和矩阵与矩阵的运算。

例如,已知 A 矩阵和 B 矩阵分别为

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$B = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

则

$$A .* B = \begin{pmatrix} 3 & 8 \\ 12 & 8 \end{pmatrix}$$

$$A .^2 = \begin{pmatrix} 1 & 4 \\ 9 & 16 \end{pmatrix}$$

在 MATLAB 中提供了许多直接处理数据集统计分析的函数,如:求解一维函数的过零点,求最大值、最小值、平均值、标准偏差、相关系数等。

例如,键入向量 x 的值,

$$x = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20]$$

则

$$x(1) = 1$$

$$x(2) = 2$$

$$x(3) = 3$$

$$x(4) = 4$$

### 图形与可视化

MATLAB 具有很强的绘图功能,可以绘制多种二维、三维图形,并可用不同的线型、色彩标记不同的数据特征。

二维绘图指令 plot

plot 命令是 MATLAB 中最常用的二维绘图指令。该指令将各个数据点用直线连接起来绘



制二维图形。`plot`命令打开一个默认的图形窗口,自动将数值标尺标注加到两个坐标轴上。如果已存在一个图形窗口,`plot`命令将刷新当前窗口上的图形。

命令格式: `plot(曾,赠, '参数员', 曾圆, 赠圆, '参数圆', ...)`

命令可以用一条命令在同一坐标系中画多条曲线,曾,赠为第一条曲线,曾圆,赠圆的坐标值向量,参数员为第一条曲线的选项参数,曾圆,赠圆为第二条曲线的坐标值向量,参数圆为第二条曲线的选项参数。

参数 该选项为一个字符串,决定图形的颜色、线型。

### 图形控制与修饰

下面介绍一些基本的图形控制与修饰指令。应用这些图形控制与修饰指令,可更清楚地标示出图形的意义,便于观察与分析。

#### (员) 图形窗口的分割与子绘图命令

命令 `subplot` 的绘图指令可以将绘图窗口分割成几个区域,在各个区域中分别绘图。

格式: `subplot(皂行, 灶列, 责)`

将当前绘图窗口分割成皂行灶列,并指定第责个编号的区域为当前绘图区域。区域的编号原则是“先上后下,先左后右”。每个编号区域可以以不同的坐标系单独绘图。

#### (圆) 坐标轴特性控制

格式: `axis([曾皂灶, 赠皂灶, 赠皂灶, 赠皂灶])`

此命令将曾轴限定在[曾皂灶, 赠皂灶]之间,赠轴限定在[赠皂灶, 赠皂灶]之间。

#### 摇摇(猿) 网格指令

网格是在坐标刻度标示上画出的格线。画出网格,便于对曲线进行观察和分析。

命令: `hold on`——在图形中添加网格线。

命令: `hold off`——在图形中去掉网格线。

#### 摇摇(源) 文字标注

有关图形的标题、坐标轴标注等指令有

命令: `title('字符串')`——图形标题。

命令: `xlabel('字符串')`——曾轴标注。

命令: `ylabel('字符串')`——赠轴标注。

命令: `text(曾, 赠, '字符串')`——在坐标(曾, 赠)处标注说明文字。

命令: `text('字符串')`——用鼠标在选定处标注说明文字。

输入特定的文字需要用反斜杠(撇)开头。

#### (缘) 图例注释

当在一个图上有多个曲线时,为区分各个曲线,`subplot`提供了图例注解说明命令 `legend`

格式: `legend('字符串员', '字符串圆', ... , 参数)`

#### 图例



此命令在图形中开启一个注解文本框,依据绘图的先后顺序,依次输出字符串对各条曲线进行注解说明。如字符串 `员` 表示第一个出现的线条,字符串 `圆` 表示第二个出现的线条,参数字符串确定注解视窗在图形中的位置。注解文字可以用鼠标拖动。

#### 猿援其他二维绘图

##### (员) 对数坐标图

绘制半对数坐标图用 `泽皂泽曾` 或 `泽皂泽赠` 命令,前者表示 `曾` 轴为以 `员` 为底, `赠` 轴为线性坐标,或者表示 `赠` 轴为以 `员` 为底, `曾` 轴为线性坐标。绘制对数坐标图用 `泽皂泽` 命令,表示 `曾` 轴和 `赠` 轴均以 `员` 为底。

格式(与 `泽皂泽` 指令相同):

`泽皂泽曾` `曾`, `赠`, '参数员', `曾`, `赠`, '参数圆', ...)

`泽皂泽赠` `曾`, `赠`, '参数员', `曾`, `赠`, '参数圆', ...)

`泽皂泽` `曾`, `赠`, '参数员', `曾`, `赠`, '参数圆', ...)

##### (圆) 极坐标图

命令 `泽皂泽` (`赠`, `赠`, `赠`, 参数)

`赠` 和 `赠` 为辐角和向径的向量,参数项同 `泽皂泽` (辐角单位为 `赠` 弧度)。

##### (猿) 饼图

饼图在统计中常用来表示各因素所占的百分比示例。

格式 `泽皂泽曾` 或 `泽皂泽赠`, `泽皂泽`

若 `曾` 为向量,该命令绘制 `曾` 的每一元素占全部向量元素总和值的百分比的饼图;

若 `曾` 为矩阵,该命令绘制 `曾` 的每一元素占全部矩阵元素总和值的百分比的饼图;

参数 `泽皂泽` 表示某元素对应的扇形图是否从整个饼图中分离出来,若非零,则分离出来,其维数应与 `曾` 相同。

##### (源) 直方图

格式 `泽皂泽曾`, `赠`

若 `曾` 为向量,则以其各元素的序号为各个数据点的横坐标,以 `曾` 向量的各个元素为纵坐标,绘出一组垂直方向的条形图;

若 `曾` 为矩阵,则以其各列序号为横坐标,每一列在其列序号坐标上分别以列的各元素为纵坐标,绘出一组垂直方向的条形图;

`赠` 给出条形的宽度,默认为 `赠`,若大于 `员` 则条形图重叠。

##### (缘) 梯形图

梯形图可以用来表示系统中的采样数据。

格式 `泽皂泽曾` 或 `泽皂泽赠`, `赠`

`泽皂泽曾` 绘制以 `曾` 的向量序号为横坐标,以 `曾` 向量的各个对应元素为纵坐标的梯形图;

`泽皂泽赠` 绘制以向量 `曾` 的各个对应元素为横坐标,以 `赠` 向量的各个对应元素为纵坐标





益形式为

$$G(s) = \frac{(s - z_1)(s - z_2)\dots(s - z_m)}{(s - p_1)(s - p_2)\dots(s - p_n)} \quad (10-1)$$

式中,  $G$  为系统增益,  $z_i$  为零点,  $p_i$  为极点。

在 MATLAB 中零极点增益模型用  $[z, p, G]$  矢量组表示。以零点、极点构成两个列向量, 再加上增益参数, 描述系统模型。

```
z = [z1, z2, ..., zm];
p = [p1, p2, ..., pn];
G = G;
```

状态空间表达式

状态方程与输出方程的组合称为状态空间表达式, 又称为动态方程。现代控制理论用状态方程和输出方程来表达输入-输出关系, 揭示了系统内部状态对系统性能的影响。

在 MATLAB 中, 系统状态空间用  $(A, B, C, D)$  矩阵组表示。

```
A = [a11, a12, ..., a1n];
B = [b1, b2, ..., bn];
C = [c1, c2, ..., cm];
D = D;
```

式中,  $A, B, C, D$  为系统模型。

### 系统模型的变换和简化

MATLAB 的控制工具箱提供了传递函数、零极点、状态空间表达式三种模型之间相互转换的功能函数, 如表 10-1 所列。三种模型之间的转换关系如图 10-1 所示。

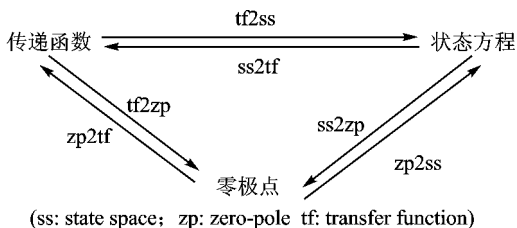


图 10-1 三种模型转换关系

表 10-1 模型转换函数说明

函数	说明
<code>[z, p, G] = tf2zp(ss)</code>	状态空间表达式转换为传递函数表达式
<code>[ss] = zp2tf(z, p, G)</code>	传递函数表达式转换为状态空间表达式
<code>[z, p, G] = zp2ss(ss)</code>	状态空间表达式转换为零极点增益形式
<code>[ss] = ss2zp(z, p, G)</code>	零极点增益形式转换为状态空间表达式
<code>[z, p, G] = zp2tf(ss)</code>	传递函数表达式转换为零极点增益形式
<code>[ss] = tf2ss(z, p, G)</code>	零极点增益形式转换为传递函数表达式

例 将系统传递函数表达式  $G(s) = \frac{1}{s^2 + 2s + 1}$  转换成零极点形式。

传递函数表达式在 MATLAB 中描述为 `tf(1, [1 2 1])`, 应用 `tf2zp` 函数可



将传递函数表达式转换为零极点增益形式,键入下列命令

```
[扎,责,噪] = 越(原,原,原,原)
```

其输出结果为

```
扎(原,原,原,原)
责(原,原,原,原)
噪(原,原,原,原)
```

写成零极点形式为

```
原 泽 越 (泽,原,原,原) (泽,原,原,原)
```

类似地,如果键入

```
[葬,遭,糟] = 越(葬,遭,糟)
```

则可得状态空间表达式

```
葬(原,原,原,原)
遭(原,原,原,原)
```

```
糟(原,原,原,原)
```

```
葬(原,原,原,原)
```

猿对控制系统的典型联结

(员) 串摇联

系统一 员和系统二 员串联(如图 猿- 猿所示),这两个系统可同为传递函数表达形式,或同为状态空间表达形式,即

```
员(原,原,原,原) 葬(原,原,原,原)
```

```
员(原,原,原,原) 葬(原,原,原,原)
```

则 葬命令可实现两个环节的串联:

```
[炸,皂,遭] = 越(葬,葬,葬)
```

```
[葬,遭,糟] = 越(葬,葬,葬)
```

摇摇(圆) 并摇联

系统一 员和系统二 员并联(如图 猿- 猿所示),这两个系统可同为传递函数表达形式,

猿

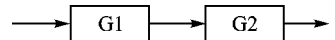


图 猿- 猿两个系统串联



或同为状态空间表达形式,即

则系统一的状态空间表达式为

系统二

则串联命令可实现两个环节的串联:

[系统一,系统二] 串联命令可实现两个环节的串联:

[系统一,系统二] 串联命令可实现两个环节的串联:

并联(狗) 反反馈

系统一和系统二按反馈方式连接(如图 10-4 所示),这两个系统可同为传递函数表达形式,或同为状态空间表达形式,即

则系统一的状态空间表达式为

系统二

则串联命令可实现两个环节的反馈连接:

[系统一,系统二] 串联命令可实现两个环节的反馈连接:

[系统一,系统二] 串联命令可实现两个环节的反馈连接:

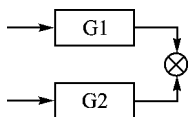


图 10-3 两个系统并联

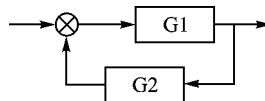


图 10-4 两个系统按反馈方式连接

系统一的所有输出连接到系统二的输入,系统二的所有输出连接到系统一的输入,用来指示系统二输出到系统一输入的连接符号,缺省时默认为负,即。总系统的输入输出数等同于系统一。

例:系统一的状态空间表达式为

系统二

连接时的系统状态方程。

键入系统一的系数矩阵

系统一

系统一

系统一

系统一

键入系统二的系数矩阵









在纸上绘图一样简单。它比传统的仿真软件包更直观、方便。杂云是 的进一步扩展,不但实现了可视化的动态仿真,也实现了与 悦或者 等编程语言甚至和硬件之间的相互数据传递,大大扩展了它的功能。杂云不但可以进行仿真,也可以进行模型分析、控制系统设计等。

### 杂云模型库

在 命令窗口中输入 命令,即可打开 模块库浏览器( 浏览器)。杂云模块库按功能进行分类,主要包括以下 类子库:

- (员) 连续模块);
- (圆) 离散模块);
- (猿) 函数和平台模块);
- (源) 数学模块);
- (缘) 非线性模块);
- (远) 信号和系统模块);
- (苑) 信宿模块);
- (愿) 信源模块)。

每一个子库中都包含了更多的环节。用户可以通过这些环节,构建自己需要的系统,然后在 提供的环境中进行仿真、分析。

### 杂云模型的建立及模型特点

#### 杂云模型的建立


杂云模型有以下几层含义:在视觉上表现为直观的方框图;在文件形式上表现为扩展名为 的 代码;在数学上体现为一组微分方程或差分方程;在行为上模拟物理器件构成的实际系统的动态特性。

从宏观的角度看,杂云模型通常包含 种“组件”:系统( )、信源( )以及信宿( )。其中,系统指被研究系统的 方框图;信源可以是常数、正弦波、阶梯波等;信宿可以是示波器、图形记录仪等。系统、信源、信宿可从 模型库中直接获得,或根据用户意愿用库中模块搭构而成。当然,对于具体的 模型而言,不一定完全包含这 种组件。例如,用于研究初始条件对系统影响的模型就不包含信源组件。

创建 模型,主要通过以下步骤:

#### (员) 建立模型窗口

建立模型窗口,即打开一个名为 的模型窗。有两种方法建立模型窗口。

- ① 选中 命令窗或某模型窗中的菜单项【云: 杂云】;
- ② 点击图标 .



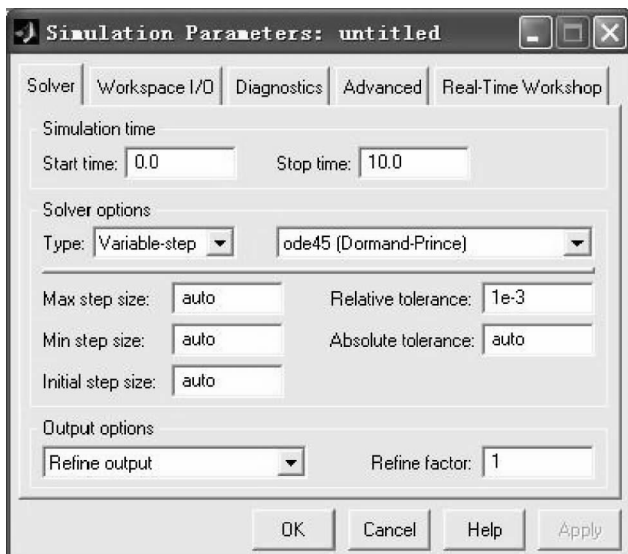


图 10-1 连续仿真参数设置页

仿真积分算法	算法说明
ode4	离散系统, 差分方程仿真
ode45	采用 4 阶 Runge-Kutta 法进行数值积分, 应用最广
ode23	采用 2 阶 Runge-Kutta 法进行数值积分, 比 ode45 速度快, 但误差大
ode45	采用 4 阶 Runge-Kutta 法进行数值积分, 比 ode23 有效, 精度高
ode15s	隐式法, 用于刚性方程的数值积分问题。当采用 ode45 无法解决时, 可尝试采用 ode15s 求解
ode15s	采用修正的 Runge-Kutta 二阶解法, 解决部分 ode15s 无法解决的刚性方程问题, 比 ode45 有效, 但误差大

### 启动仿真

设置仿真参数和选择解法器之后, 就可以启动仿真而运行。启动仿真有两种方式:

(1) 选择 **Simulation** 菜单下的 **Start** 选项来启动仿真, 如果模型中有些参数没有定义, 则会出现错误信息提示框。如果一切设置无误, 则开始仿真运行, 结束时系统会发出一鸣叫声。

(2) 在 **Command Window** 命令窗口中通过函数进行, 格式如下:

`[t, y] = sim('模型文件名', [t_start, t_end], '参数名', 参数值 1, '参数名', 参数值 2, ...)`

其中 `t_start` 为仿真起始时间, `t_end` 为仿真终止时间。[ `t, y` 为返回值, `t` 为返回的时间向量值, `y`



为返回的状态值 赠为返回的输出向量值。泽泽定义了仿真参数 ,包括以下一些主要参数。

粤粤默认值为 员员设定绝对误差范围 ;

阅阅默认值为 员决定隔多少个点返回状态和输出值 ;

杂杂解法器的选择 ;

配配默认值为 园表示不限制。若为大于零的值 ,则表示限制输出和状态的规模 ,使其最大行数等于该数值 ;

附附提供一个向量值 ,用于设定初始状态 ;

云云提供一个正数表示步阶的大小 ,仅用于固定步长模式 ;

配配默认值为 员用于变步长模式 ,表示最大的步阶大小 ;如果知道模型文件名 ,可以用以下命令得到该模型的仿真参数 泽泽(模型文件名')。

### 猿猿仿真结果分析

仿真结束后 ,可通过可视化的接收器模型 ,如示波器等 ,直接观察仿真结果 ;也可将仿真结果存储在文件中 ,进行分析。

### 员员源杂杂应用实例

下面应用 杂杂进行时域系统分析。

图 员- 远是某系统结构图。该系统由传递函数分别为 泽泽和 泽泽的两个子系统串联而成 ,同时还包括一个非线性环节(用来限幅 ,把输入控制在 原园~ 园之间)。

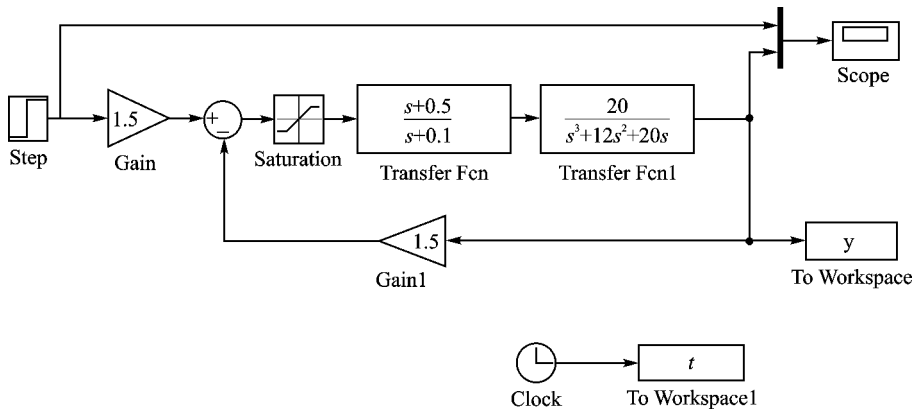


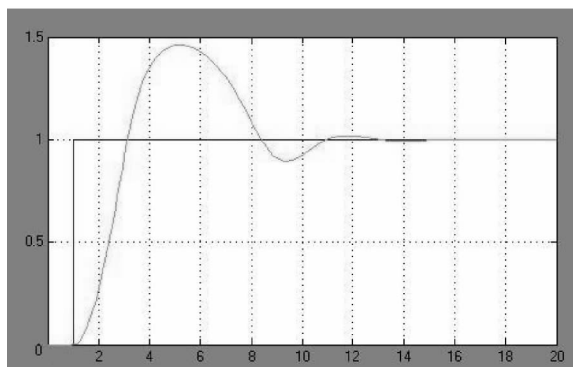
图 员- 远 远仿真方框图

仿真算法选择为 泽泽从示波器上可以比较输入(阶跃输入)和输出 ,见图 员- 苑。

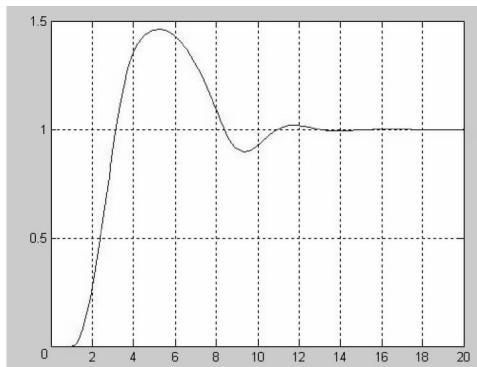
仿真方框图中的参数可以是实际数值 ,也可以是字母表示的变量名。用字母表示的变量可以在 配配空间赋值 ,或用 配文件赋值 ,然后进行仿真运算。如图 员- 远所示 ,仿真结果



既用示波器直接图形化表示,同时还赋予变量赠保存在配粤找粤月工作空间。仿真结束后,也可直接使用指令责燥(赠)得到响应如图员-苑遭所示。注意,采用字母赋值,所有变量都必须是全球变量。



(a) 示波器仿真结果



(b) 应用plot命令结果

图 员-苑 仿真结果

## 员-苑-圆 杂云 实时代码生成方法及应用实例

配粤找粤月提供了一个实时开发环境,可用于实时系统仿真和产品的快速原型化,这就是特殊应用工具箱——砸燥(赠,实时工作空间)。

砸燥是一个基于杂云 的代码自动生成环境,它能直接从杂云 的模型中产生优化的、可移植的和可定制的代码,并且可以根据目标配置自动生成多种环境下的程序。其生成的代码有悦语言和粤语言两种,并且砸燥支持源种代码格式:杂函数轴加速器代码格式、实时代码格式、实时皂葬代码格式和嵌入式代码格式。

应用配粤找粤月轴燥(赠,实时工作空间)的目的在于利用砸燥在杂云 环境中从杂云 的仿真模型(\*皂葬)生成悦语言代码供仿真应用软件调用,减少编程调试的工作量。

### 员-苑-圆-员 砸燥 代码自动生成和实时系统开发环境

#### 员-苑-圆-员-1 砸燥 体系结构

砸燥是一个适用于多种操作系统和硬件类型的开放式系统。图员-愿是砸燥的体系结构,同时也显示了砸燥代码自动生成过程及所需条件。

砸燥自动程序创建过程能在不同主机环境下生成用于实时应用的程序。该创建过程使用高级语言编译器中的联编实用程序,来控制所生成源代码的编译和链接过程。

砸燥使用一个高级的酝文件命令控制程序创建过程,默认命令是皂葬燥(赠)

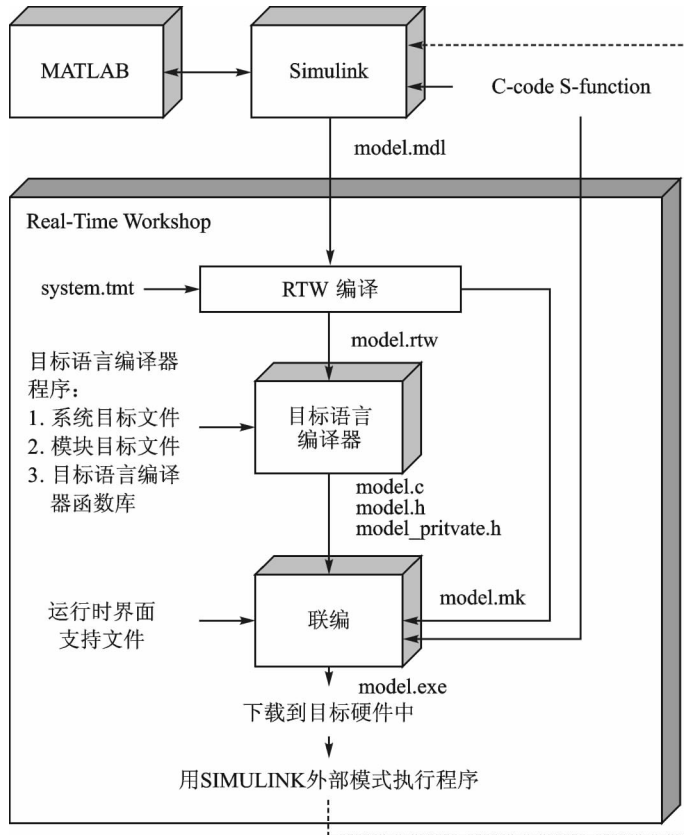


图 4-1-1 目标代码的开放式体系结构

该创建过程包括以下 源个步骤：

摇摇(员) 分析模型和对模型描述文件进行编译

砸碎的程序创建过程首先从对 杂定器模块方框图的分析开始,需要计算相应参数、递推信号宽度和采样时间、确定模型中各模块的执行次序等。

摇摇(圆) 由目标语言编译器从模型中生成代码

目标语言编译器是一种能将上述模型描述文件转换为指定目标代码的解释性编程语言,由系统目标文件、模块目标文件、目标语言编译器函数库等组成。目标语言编译器的输出是 杂定器模块图的源代码版本。

摇摇(猿) 生成程序联编文件

即 皂器器文件,用于指导联编程序如何对从模型中生成的源代码、主程序、库文件或用户提供的模块进行编译和链接。

摇摇(源) 生成可执行程序

圆源



在自定义的程序创建文件的控制下,由联编实用程序生成可执行程序。这一步是可选项。  
图 10-1 显示了在配置对话框提供的参数和选项,其中的

不仅影响了,还影响了。而选项页中的参数则直接影响到代码生成和优化过程,同时程序创建过程也要通过该选项页进行初始化和控制。

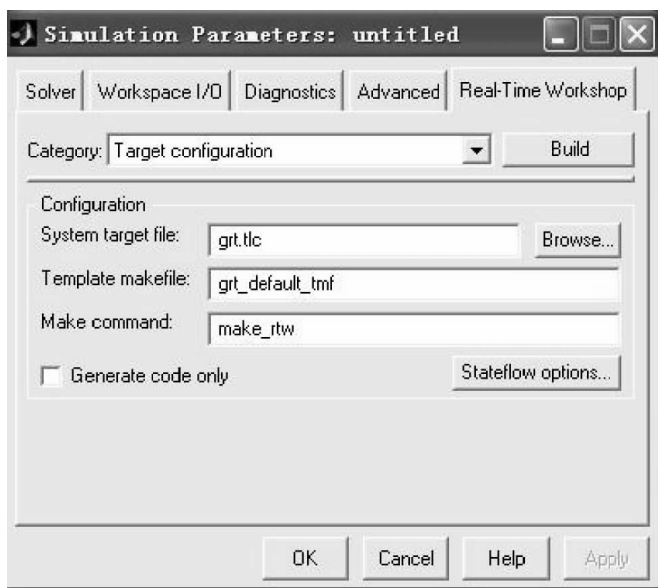


图 10-1 选项页

下面逐项分析选项页中的设置项和含义。

图 10-1 中的下拉列表框

用于选择不同的选项或控制组,被选择的选项组将显示在选项页的下半部分。

图 10-1 中的下拉列表中的选项包括:

① 是与控制代码生成和程序创建过程及选择控制文件相关的高级选项。目标配置主要包括以下一些选项:

- √ 系统目标文件)文本框,显示和输入特定的系统目标文件名称;
- √ 模板联编文件)文本框,显示和输入模板联编文件名称;
- √ 联编命令)文本框,显示和输入命令名称;
- √ 复选框,选中时,程序创建过程将只生成代码而不会调用联编命令,



不会生成可执行程序；

√ 单击 **编译并运行** 按钮,如果模型中包含了 **编译并运行** 模块,该按钮将启动 **编译并运行** 的 **编译并运行** 对话框。

- ② **编译并运行** 目标语言编辑器调试和执行 **编译并运行** 选项。
- ③ **编译并运行** 编译并运行 **编译并运行** 所有目标配置通用的代码生成设置选项。
- ④ **编译并运行** 编译并运行 **编译并运行** 所有目标配置通用的代码和标识格式设置选项。
- ⑤ **编译并运行** 编译并运行 **编译并运行** 所选目标配置专有的选项组。

### 编译并运行 (C) 按钮

此按钮的作用是对代码生成和程序创建过程进行初始化。如果选中了 **编译并运行** 选项,则 **编译并运行** 下的 **编译并运行** 复选框, **编译并运行** 按钮上的说明文字将显示为 **编译并运行**

#### 编译并运行 程序创建

在正确设置了所有选项后,有两种方法可创建 **编译并运行** 程序。一种方法是用鼠标单击 **编译并运行** 原 **编译并运行** 选项页中 **编译并运行** 按钮;另一种方法是打开 **编译并运行** 窗口中【**编译并运行** **编译并运行** **编译并运行**】: **编译并运行** 选项。在代码生成过程中, **编译并运行** 会在当前工作路径下生成一个新目录,目录名源自于源模型和所选的目标,形式为 **编译并运行** 所生成的代码以及创建过程中产生的其他文件将存储在该目录下。

**编译并运行** 生成的源程序主要包括以下文件:

#### ① **编译并运行**

包括所有实现模型算法的代码的入口( **编译并运行**, **编译并运行**, **编译并运行**, **编译并运行** ) ,同时也包括所有模型注册代码。

#### ② **编译并运行**

包括模型需要的私有变量和数据。

#### ③ **编译并运行**

定义了模型数据结构和访问模型及数据结构的接口。

#### ④ **编译并运行**

包括与输入/输出数据结构有关的变量声明。如果模型中没有用到这些数据结构,则不会产生此文件。

#### ⑤ **编译并运行**

包括实时模型数据结构的前向声明。所有其他头文件都需要包括此文件。

#### ⑥ **编译并运行**

包括静态 **编译并运行** 函数所需的文件。

#### ⑦ **编译并运行** 可选

这个文件是可选的。为运行时的程序提供访问模型参数的数据结构。

#### ⑧ **编译并运行** 可选



这个文件是可选的。为程序访问输出提供必要的数据结构。

## 导弹俯仰应用实例

以导弹控制系统为例,生成一个**砸穿**通用实时目标的应用程序。通过此例,演示**砸穿**的代码生成、程序创建等功能。

### 系统分析

导弹控制系统用于控制导弹的飞行姿态与位置,完成自控段的飞程序,在自导段则接收末制导雷达的控制信号,操纵导弹飞向目标。根据导弹控制分系统完成的功能,将控制系统的结构分为俯仰控制回路、航向控制回路、滚转控制回路和舵机控制回路。本节重点分析俯仰控制回路。俯仰控制回路接收其他分系统的指令信号及导弹姿态信号,完成对导弹高度和姿态的控制和稳定。

建立名为**砸穿**的导弹俯仰控制回路模型,如图**砸穿**所示。

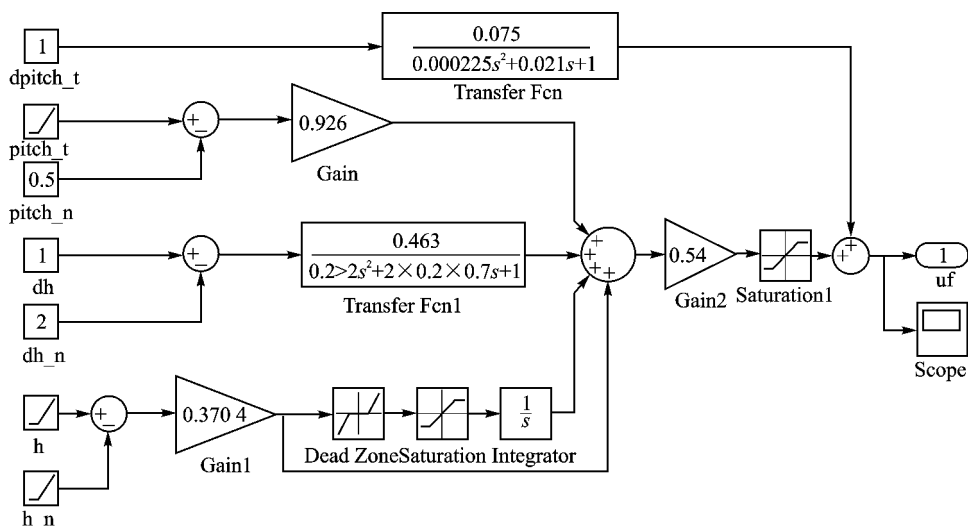


图 **砸穿** 导弹俯仰控制回路模型框图

导弹俯仰控制回路主要完成对导弹纵向控制律的解算,生成控制升降舵的控制信号。

模型输入信息主要包括弹道指令信号,如指令高度 **砸穿**、指令高度变化率 **砸穿**和俯仰角指令 **砸穿**以及转换后的姿态信号,如转换后的导弹俯仰角 **砸穿**、转换后的导弹俯仰角变化率 **砸穿**、导弹飞行高度 **砸穿**和导弹飞行高度变化率 **砸穿**。

模型输出信号是俯仰控制器舵偏电压 **砸穿**。

设计过程中用到的基本元模型类的功能如下:

(员) 比例增益类( **砸穿**) :完成对输入信号的直接比例放大功能;



- (圆) 死区类(悦)完成对输入信号的死区功能;
- (猿) 饱和类(悦)完成对信号的限幅功能;
- (源) 积分器类(悦)完成对输入信号的积分功能;
- (缘) 传递函数类(悦)完成控制系统中的二阶传递函数功能。

### 圆程序参数设置

要从模型窗口中正确地生成代码,需要修改一些仿真参数。特别是需要使  
用定步长积分器。选中模型窗口的【】,设置参数如下:

摇摇(员) 设置选项页

摇摇(猿) 设置选项页

摇摇(猿) 设置选项页

摇摇(猿) 设置选项页

摇摇(猿) 设置选项页

摇摇(猿) 设置选项页

摇摇(圆) 设置选项页

摇摇选中中的复选框,这样使生成的程序在运行时将  
模型的输出数据记录到文件的矩阵和。本例中记录的是俯仰控  
制器舵偏电压。

摇摇(猿) 设置选项页

摇摇确保列表中的系统目标文件为模板联编文件为  
命令。

摇摇其他选项为默认值即可。

### 猿创建可执行程序

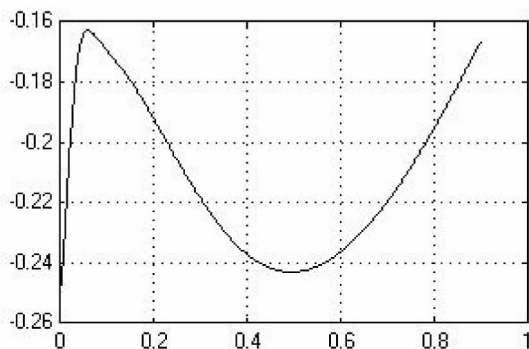
单击按钮,则从模型中生成悦代码,对其进行编译和链接,最终生成所需程序。在  
当前工作目录下,生成可执行程序。所有源代码文件存储在目录  
下。

通过比较分析,可以看出,直接在环境下的仿真结果,与运行  
程序得到的结果完全相同,如图-所示。

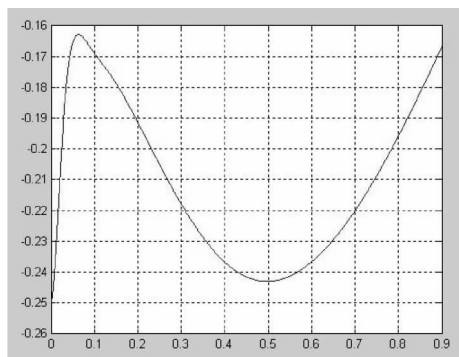
除了上述直接创建可执行程序,还可以只生成悦源代码,然后按照面向对象方法进行  
封装,直接在悦编译环境中进行编译、链接。

## 习 题

在环境中,采用不同的数值积分方法和不同的积分步长计算连续系统



(a) MATLAB环境下仿真结果



(b) 运行可执行程序结果

图 10-1 员摇仿真结果对比图

员的阶跃响应,具体要求如下:

(员) 积分方法为:一阶欧拉法、改进欧拉法、龙格-库塔法(积分方法需要用皂语言来实现,使用配葬的基本数学运算来实现,不许采用配葬提供的函数来直接求解)。

(圆) 积分步长为园猿泽园猿泽园猿泽园猿泽园猿泽

(猿) 仿真时间选择园园泽

(源) 将所有仿真结果用图表显示,并与泽皂下的仿真结果进行比较;

(缘) 对结果进行比较分析。

## 参考文献

[员] 杨涂,李立涛,杨旭,等.系统实时仿真开发环境与应用[M].北京:清华大学出版社,园园园援

[圆] 张志涌.精通配葬猿月缘版[M].北京:北京航空航天大学出版社,园园园援

[猿] 张平.配葬猿月基础与应用简明教程[M].北京:北京航空航天大学出版社,园园园援

[源] 配葬猿月基础与应用简明教程[M].北京:北京航空航天大学出版社,园园园援