

系
统
仿
真

系 统 仿 真

宋承龄 王章德 编著

國 防 工 業 出 版 社

國

TP15

社

系 统 仿 真

宋承龄 王章德 编著

国防工业出版社

内 容 简 介

本书系统地介绍系统仿真的基本概念、建模方法、分析与致效技术、方差减小技术和仿真语言等方面的知识。

全书共分十章：概述，连续系统仿真，随机变数产生方法，离散事件系统仿真，离散事件仿真策略与结构模型，仿真输出结果分析，模型的致效，方差减小技术，通用离散事件系统仿真语言 GPSS-F77和混合仿真语言GASP IV。本书重点放在离散事件系统仿真方面，内容力求实用。离散事件仿真策略与结构模型一章，系统地介绍了仿真过程的几种调度方法，不但是一般仿真模型流程管理的设计基础，而且也是学习仿真语言的入门知识。

本书可作为高等学校系统工程、管理工程、自动控制等专业本科生系统仿真课的教材或参考书，也可作为研究生初次学习系统仿真课的教材或参考书，还可供有关工程技术人员和科学工作者参考。

系 统 仿 真

宋承龄 王章德 编著

国防工业出版社

(北京市车公庄西路老虎庙七号)

新华书店北京发行所发行 各地新华书店经营

国防工业出版社印刷厂印刷

787×1092 1/16 印张14 323千字

1989年5月第一版 1989年5月第一次印刷 印数：0,001—1,780册

ISBN 7-118-00478-2/TP·61 定价：2.80元

前 言

仿真技术近年来在国内得到了推广和应用。它的重要性已逐渐为人们所认识。随着计算机普及与四化建设的迫切需要, 数字仿真技术必将有一个大的发展。然而, 系统地介绍系统仿真的基本原理和具体应用方法的书籍却不多。我们结合近年来的讲课任务编写了《系统仿真讲义》, 并在此基础上, 写成这本《系统仿真》, 目的是想为学习系统仿真课的同学提供一本体系较完整、内容较实用的数字仿真教材。编写时, 我们认为读者已具有自动控制原理、概率与统计、计算方法和 FORTRAN 语言等方面的知识。

第一章介绍系统仿真的基本概念和实施仿真的一般步骤; 第二章简单地介绍连续系统仿真方法; 第三章介绍随机变数的产生方法; 第四章介绍离散事件系统仿真的基本概念和建模方法; 第五章介绍各类仿真策略和结构模型, 并通过实例具体示出各类仿真策略的实施方法; 第六、第七和第八章介绍系统仿真的辅助技术: 仿真输出结果分析、仿真的致效和方差减小技术; 第九章和第十章分别介绍通用离散事件系统仿真语言 GPSS-F77 和连续-离散事件系统混合仿真语言 GASP IV 的结构、各子程序的功能与调用方式和使用方法。

本书选择介绍 GPSS-F77 和 GASP IV 两种仿真语言的考虑是: 这两种语言都是以 FORTRAN 语言为宿主的, 便于移植、修改与扩充, 便于国内现有计算机资源的利用。

在使用本书时, 如果课程时数较少, 可考虑不用第九、第十章。

本书由宋承龄担任主编, 曹棉章担任主审。

本书的第二、第三和第四章由王章德编写; 第一章由宋承龄和王章德两人合写, 其余部分皆由宋承龄写成。

由于编者水平有限, 不当与错误之处在所难免, 望批评指正。

目 录

第一章 概 述	
§ 1.1 系统仿真的含义与分类	1
§ 1.2 数字仿真的要素	3
§ 1.2.1 系统	4
§ 1.2.2 模型	4
§ 1.3 系统仿真的一般实施步骤	5
§ 1.4 其他	7
§ 1.4.1 蒙特卡洛(Monte Carlo)法	7
§ 1.4.2 仿真法与解析法的区别	7
第二章 连续系统仿真	
§ 2.1 数值积分法	8
§ 2.1.1 欧拉法	8
§ 2.1.2 梯形法(改进的欧拉法)	9
§ 2.1.3 四阶龙格-库塔法	9
§ 2.2 面向微分方程的通用语言仿真程序	10
§ 2.2.1 状态方程模型的仿真程序	10
§ 2.2.2 传递函数模型的仿真程序	13
§ 2.3 连续系统结构图法仿真	17
§ 2.4 连续系统离散相似法仿真	21
§ 2.4.1 连续系统状态方程离散化	22
§ 2.4.2 面向结构图模型离散相似法仿真	25
§ 2.5 非线性系统仿真	30
§ 2.5.1 典型非线性环节仿真模型	30
§ 2.5.2 含有非线性环节系统仿真程序设计	32
习题	34
第三章 随机变数产生方法	
§ 3.1 $[0, 1]$ 均匀分布随机数产生方法	35
§ 3.1.1 乘同余法随机数产生器	36
§ 3.1.2 其他同余法	38
§ 3.2 随机数统计检验	38
§ 3.2.1 分布的均匀性检验	38
§ 3.2.2 独立性检验	39
§ 3.2.3 矩检验	40
§ 3.3 非均匀分布随机数产生方法	41
§ 3.3.1 反变换法	42
§ 3.3.2 舍选法	43
§ 3.3.3 表搜索法	46
§ 3.3.4 正态分布 $N(\mu, \sigma)$ 随机数的产生	48
§ 3.3.5 泊松分布随机数的产生	50
§ 3.3.6 爱尔朗分布随机数的产生	50
§ 3.4 多维随机变数抽样	51
§ 3.4.1 条件密度法	51
§ 3.4.2 正态随机矢量的产生	52
§ 3.5 随机过程的产生	53
§ 3.5.1 一维平稳正态过程的产生	53
§ 3.5.2 n 维正态马尔可夫过程的产生	54
习题	55
第四章 离散事件系统仿真	
§ 4.1 概述	57
§ 4.2 离散事件系统仿真模型的建立方法	58
§ 4.2.1 仿真目的	58
§ 4.2.2 描述系统	59
§ 4.2.3 超级市场仿真模型的建立	61
§ 4.3 流程管理	66
§ 4.3.1 仿真时钟	66
§ 4.3.2 时间进程管理与通道选择	66
§ 4.3.3 同时事件管理	67
§ 4.3.4 排队规则	70
§ 4.4 输出函数的确定	71
§ 4.5 模型的有限自动机化	72
§ 4.6 机修车间仿真实例	75
§ 4.6.1 系统状况与仿真目的	75
§ 4.6.2 系统的描述	75
§ 4.6.3 仿真模型的程序流程图	

与源程序	77
§ 4.6.4 运行结果分析	80
习题	80

第五章 离散事件仿真策略与结构模型

§ 5.1 面向事件结构模型	83
§ 5.1.1 面向事件结构模型清单	83
§ 5.1.2 计算机-用户系统面向事件 结构模型	85
§ 5.2 活动扫描结构模型	88
§ 5.2.1 活动扫描结构模型清单	90
§ 5.2.2 计算机-用户系统活动扫描 结构模型	91
§ 5.3 进程互配结构模型	94
§ 5.3.1 进程互配策略与进程互配 结构模型的特点	94
§ 5.3.2 计算机-用户系统进程互 配结构模型	96
习题	99

第六章 仿真输出结果分析

§ 6.1 概述	100
§ 6.2 重复运行仿真	101
§ 6.3 批均值法	103
§ 6.4 再生法	104
习题	107

第七章 仿真模型的致效

§ 7.1 概述	108
§ 7.2 仿真程序的核实	108
§ 7.3 关于致效的一般观点	109
§ 7.4 致效的三步法	110
§ 7.4.1 研制高外表有效性的模型	110
§ 7.4.2 模型假设的实验检验	110
§ 7.4.3 确定仿真输出数据的代表性	111
§ 7.5 对比统计法	112
§ 7.5.1 验证法	112
§ 7.5.2 置信区间法	113
习题	114

第八章 方差减小技术

§ 8.1 公用随机数法	115
§ 8.2 对偶变量法	117
§ 8.3 控制变量法	119
习题	122

第九章 通用离散事件系统仿真 语言 GPSS-F77

§ 9.1 概述	123
§ 9.2 GPSS-F77 的结构	124
§ 9.2.1 事件管理	125
§ 9.2.2 动元 (Transactions)	126
§ 9.2.3 动元管理	129
§ 9.2.4 动元产生与消除	130
§ 9.2.5 动元状态管理	131
§ 9.2.6 框架 (Chassis)	133
§ 9.3 站与政策	135
§ 9.3.1 GPSS-F77 的政策	135
§ 9.3.2 子程序POLICY	136
§ 9.3.3 优先级动态赋值子程序 DYNVAL	136
§ 9.3.4 状态矢量	137
§ 9.4 设备	138
§ 9.4.1 占用与空出设备	139
§ 9.4.2 抢先	140
§ 9.4.3 设备的多入口	143
§ 9.5 复设备	143
§ 9.5.1 复设备的结构	143
§ 9.5.2 占用和空出复设备	145
§ 9.5.3 复设备中的抢先	147
§ 9.5.4 方案: 服务单元管理	149
§ 9.6 库 (Storages)	150
§ 9.6.1 库的结构	150
§ 9.6.2 非寻址库	151
§ 9.6.3 策略 (Strategies)	153
§ 9.6.4 可寻址库	154
§ 9.6.5 GPSS-F77 的策略	157
§ 9.7 调配动元	159
§ 9.7.1 条件分支	159
§ 9.7.2 条件等待	159

§ 9.7.3 门	161	§ 10.3.1 主程序 MAIN	188
§ 9.7.4 收集站	163	§ 10.3.2 子程序 STATE	189
§ 9.7.5 用户链与起动站	164	§ 10.3.3 子程序 INTLC	190
§ 9.8 族	165	§ 10.3.4 子程序 EVNTS(K)	191
§ 9.8.1 同类作业	166	§ 10.3.5 子程序 SCOND	192
§ 9.8.2 族的数据区	166	§ 10.3.6 子程序 SSAVE	193
§ 9.8.3 亲属的生与死	166	§ 10.3.7 子程序 OUTPUT	193
§ 9.8.4 调配族成员	168	§ 10.3.8 子程序 UERR 与空子程序	194
§ 9.9 GPSS-F77 中的随机变数 产生器	170	§ 10.3.9 GASPⅡ数据与公用区	195
§ 9.9.1 函数 RN	171	§ 10.3.10 数据输入主程序	198
§ 9.9.2 子程序 UNIFORM	171	§ 10.4 GASPⅣ子程序	198
§ 9.9.3 子程序 ERLANG	171	§ 10.4.1 子程序 GASP	200
§ 9.9.4 子程序 GAUSS	172	§ 10.4.2 子程序 DATEIN	201
§ 9.9.5 子程序 LOGNOR	172	§ 10.4.3 函数子程序 KROSS	201
§ 9.9.6 子程序 BOXEXP	173	§ 10.4.4 子程序 COLCT(XX, ICCT)	202
§ 9.9.7 随机变数产生器的使用	173	§ 10.4.5 子程序 TIMST(XX, T, ISTAT)	204
§ 9.10 收集统计	174	§ 10.4.6 子程序 TIMSA(XX, T, ISTAT)	205
§ 9.10.1 仓(BINS)	174	§ 10.4.7 子程序 HISTO(XX, IHIST)	206
§ 9.10.2 频率表	176	§ 10.4.8 子程序 GPLOT(XX, T, JPLOT)	206
§ 9.10.3 系统状态的报告	178	§ 10.4.9 子程序 SUMRY	207
§ 9.10.4 保存与还原系统状态	180	§ 10.4.10 随机变数产生器	208
§ 9.10.5 结束随机仿真	180	§ 10.4.11 其他 GASPⅣ子程序	209
§ 9.11 GPSS-F77 仿真语言的 使用方法	181	§ 10.5 GASPⅣ仿真语言的使 用方法	211
§ 9.12 举例	183	习题	212
习题	185	附录 I GPSS-F77 的主程序框架	212
第十章 混合仿真语言 GASPⅣ		附录 II GPSS-F77 系统变量	215
§ 10.1 概述	186	附录 III GASPⅣ错误信息表	217
§ 10.2 序列存储区	187	参考文献	218
§ 10.3 用户程序	188		

第一章 概 述

§ 1.1 系统仿真的含义与分类

不论是分析一个现有系统，还是设计一个新系统，往往需要进行实验研究。实验、分析计算和决策是科学研究和设计工作的一般方法和步骤。然而，如果直接用真实系统进行实验，往往是不经济或不安全的，有时甚至是做不到或者没有意义的。例如，对于宇航系统、核反应堆控制系统等，直接实验往往很危险，是不允许的，而且费用高昂。对于经济、社会、政治、人口、医学、生态等非工程系统来说，直接做实验几乎是不可能的，或者是时间上得不到预测实效的。因此，实践中出现了用模型代替真实系统做实验的方法。例如，古代在建造大的亭台楼阁或宫殿时，工匠们往往先做个模型进行实验，然后再建造真的；现代的飞机风洞实验和军事训练上的沙盘作业，也都是用模型代替真实系统做实验的例子。所谓系统仿真就是用模型代替真实系统进行实验研究，以获取所需信息的实验方法。系统仿真涉及三个要素：真实系统，模型和实验系统。在飞机的风洞实验中，真实系统是待研究的真飞机或设计图纸上的飞机；模型是气流和小的模型飞机；实验系统是风洞。根据模型和实验系统的性质，系统仿真可分成下列几类：

物理仿真——模型与实验系统皆为物理系统。

模拟机仿真——模型是数学模型，实验系统是模拟计算机。换句话说，就是用数学模型代替真实系统在模拟计算机上做实验。

数字（机）仿真——用数学模型代替真实系统在数字计算机上做实验。

数字机与模拟机混合仿真——用数学模型代替真实系统，在数字机与模拟机混合系统上做实验。

仿真器仿真——用的模型是物理-数学混合模型，实验系统是包括计算机在内的物理系统。

飞机的风洞实验属于物理仿真。模拟机仿真的例子可列举如下：

有一机械系统如图 1-1 所示，在外力 $F(t)$ 作用下，物体 1 做上下运行。为了使运动平稳，需适当选择阻尼系数 B ，为此写出系统的运动方程：

$$M\ddot{x} + B\dot{x} + Kx = F(t) \quad (1-1)$$

式中 M ——物体 1 的质量；

B ——阻尼器 3 的阻尼系数；

K ——弹簧的弹性系数；

x ——物体 1 的位移。

并设初始条件： $x(t)|_{t=0} = x_0$ ， $\dot{x}(t)|_{t=0} = \dot{x}_0$ 。

把模拟计算机的线路按图 1-2 连接，按初始条件启动，连续输入正比于 $F(t)$ 的模

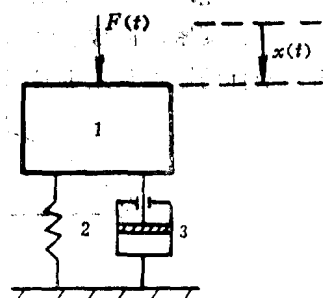


图1-1 质量-弹簧-阻尼系统示意图
1—物体， 2—弹簧， 3—阻尼器。

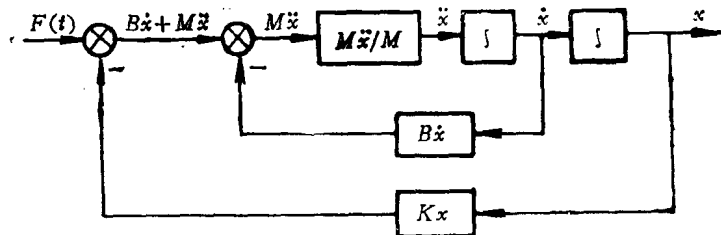


图1-2 图1-1所示系统的仿真框图

拟量, 模拟计算机便可连续输出满足式(1-1)的 $x(t)$ 。根据 $x(t)$ 的变化规律可以看出物体 1 的运动是否平稳。如果不平稳, 则可改变 B 值, 重复上述实验, 直到满足要求为止, 然后按所得 B 值设计制造阻尼器。式 (1-1) 便是所用的数学模型。

如果把式 (1-1) 离散化成差分方程, 编成计算机程序, 输入数字计算机, 求出 $x(t)$ 在各个时刻的值, 便是数字仿真的例子。

仿真器仿真的例子有: 用航线起落训练飞行仿真器 (主要由计算机、座舱、视景信息产生器和指挥台等组成) 和船舶操纵训练仿真器对操纵手进行训练; 用瞄准跟踪仿真器检验被操纵对象的结构参数选择得是否适合操纵手的响应特性等。这后一例子是用于系统的分析与设计的。所以, 仿真器仿真不但可以用于训练, 而且还可以用于系统的分析与设计。

系统仿真虽然古已有之, 但现代的仿真技术却是在计算机问世以后才发展起来的。当然, 数字仿真也是在数字计算机诞生以后才发展起来的。近年来随着数字计算机软、硬件的不断飞速发展, 数字仿真技术发展很快, 并被广泛地应用于各个领域。数字仿真计算机的出现已使全数字实时仿真成为现实。预料, 今后系统仿真技术将向着全数字方向发展。特别是将来随着智能机的问世, 数字仿真也必将发展到一个相应的新阶段。

本书只讨论数字仿真。下边再举一个数字仿真的例子。

设用火炮向地面目标 A 射击, 见图 1-3。由于瞄准误差和火炮射弹的散布, 命中率一般小于 1。其具体值可用实弹射击方法统计计算出来, 也可用数字仿真法统计计算出来。为此, 先建立打靶实验的数学模型。火炮射弹的散布可用二维随机变量 (X', Y') 来表示, 其密度函数 $f(x', y')$ 及特征数在火炮的射表中可查到。而瞄准手的折合成直角坐标的瞄准误差也是二维随机变量, 设为 (X'', Y'') 。它的大量采样可用比火炮瞄

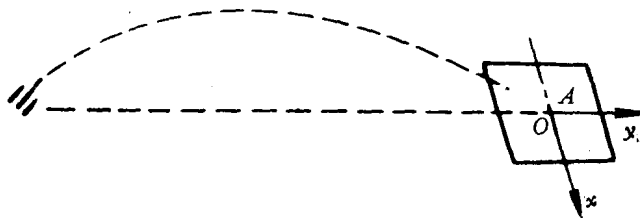


图1-3 火炮射击示意图

准装置更精确的光学仪器测量、转换出来。设根据 (X'', Y'') 的大量采样统计计算出其密度函数为 $P(x'', y'')$, 则可为火炮的打靶实验建立模型如图 1-4 所示。这是个流程图, 广义上, 它也是一种数学模型。

按图 1-4 编写程序, 输入计算机便可得到概率的统计值——频率 N/Z , 同时还可以由显示屏上看到射弹的散布情况。

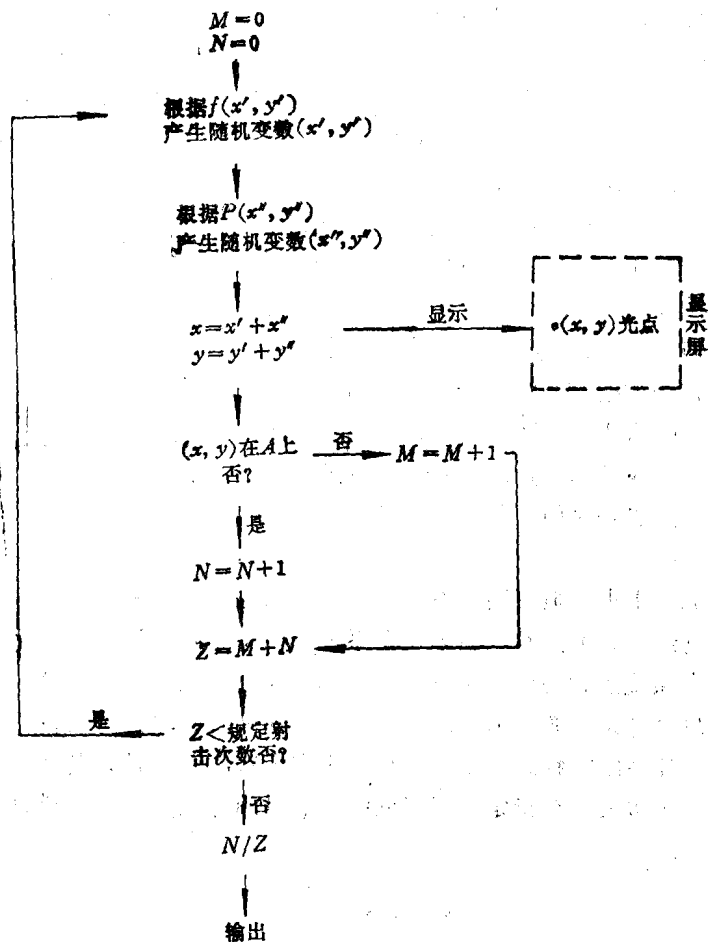


图1-4 打靶实验的数学模型

§ 1.2 数字仿真的要素

由以上所述可知, 数字仿真涉及三要素: 真实系统, 数学模型和数字计算机。它不但涉及这三个要素本身, 而且还要处理它们之间的关系。根据真实系统建立数学模型叫建模。用模型在数字计算机上做实验叫仿真, 见图 1-5。由于本书只讨论数字仿真, 所用模型都是数学模型, 故以后所提到的模型, 如无特殊说明, 皆指数学模型。



图1-5 数字仿真三要素

§ 1.2.1 系 统

系统一词已被广泛应用于各种领域，但它在各个不同领域内的含义是不尽相同的。这里，针对数字仿真几乎可应用于任何对象这个特点，给“系统”一个着眼于概括面宽的定义：系统是人们所关心的客观世界的某一部分，它可以是天然的或人工的，现有的或计划中的。例如，自然科学家可以对花的生长过程或一窝蜜蜂的“社会”组织结构等这样一些现有系统感兴趣；而一位工程师所关心的可能是一个新的人工系统的设计，例如一个新型计算机或一个大城市的运输系统的设计。

通常，我们可以这样说：一个真实系统是（或将变成）一个行为数据产生源。行为数据的基本形式可以是 X 对 T 的曲线，如图1-6所示。其中， X 可以是人们所关心的任何变量，如室温、花瓣数或国家生产总值等。 T 是时间，单位可以是 s （秒）， h （时）， d （日）或 y （年）。

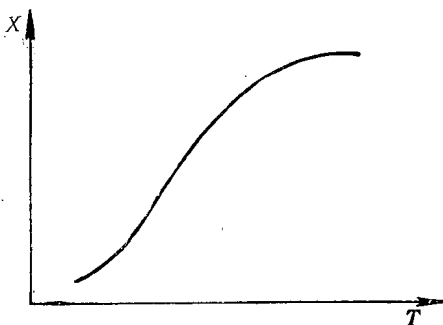


图1-6 行为数据图

系统可分成若干类，但着眼点不同，分类结果也不同。一般按系统状态变化是否连续把系统分成连续与离散两类。如果一个系统的状态是随时间连续变化的，则称为连续系统；如果一个系统的状态变化是离散的，则称为离散系统。由于我们把离散的状态变化叫做事件，故这种系统亦称为离散事件系统。如果时间推移对状态变化起着实质性的影响，则离散事件系统的状态变化必定只发生在一些离散的时间点上，而时间基准却是连续的，即事件可以发生在指定时间区间的任意时间点上。前边谈到的打靶实验便是一个离散事件系统。

§ 1.2.2 模 型

与系统一样，模型一词也被广泛应用于各种领域。同样，它在各个领域内的含义也是不尽相同的。这里，也仅就数字仿真的特点予以定义。所谓模型，其本质是一组说明。根据这些说明，可以产生与真实系统相应的行为数据。说明可以由文字、符号、图形和数学公式（如微分方程、差分方程、状态方程等）组成。例如式（1-1）和图1-4都是模型。

注意：模型本身并不能产生行为数据。它只是向人或计算机提供说明，根据这些说明由人或计算机产生行为数据。为了方便，以后我们把根据模型由人或计算机产生行为数据直接说成模型产生行为数据或模型行为。

用模型行为模仿系统行为便是系统仿真的实质。

模型有各种不同形态。着眼点不同，可以有不同分类。这里仅介绍几种常用分类。

1. 连续模型和离散模型

状态随时间连续变化的模型叫做连续模型。这种模型通常用微分方程或状态方程表示。由于数字计算机是个离散系统，故连续模型必须加以离散化后才能用作数字仿真模型。离散化后的模型称为离散化模型，如用差分方程、离散时间状态方程表示的模型。

状态变化是离散的模型称为离散事件模型。图 1-4 是个离散事件模型。

2. 随机模型和确定性模型

凡含随机变量的模型称为随机模型（或概率模型）。不含随机变量的模型称为确定性模型。同样的输入，确定性模型会给出唯一的输出结果；而随机模型给出的输出结果本身却是随机变量，需要用统计方法研究它们的特征。离散事件模型多半是随机模型（严格说来应是伪随机模型，这以后将会讨论到）。

3. 自激模型和非自激模型

如果真实系统用与外界影响割断连系的模型模仿，则称此模型为自激模型。否则称为非自激模型。非自激模型是具有输入变量的模型。

如何建立模型，由于客观事物的复杂性，不但因情况而异，而且在很大程度上还取决于建模者的知识、经验和技巧。有人认为建模不仅是一种创造性劳动，而且也是一种艺术，不能用一些条条框框定死。但有两点基本要求：

1. 模型要有足够的精度：虽然模型只能是真实系统的一种近似，但其近似程度，应能使反映系统实况的精确程度满足研究目的的要求。

2. 简明：在保证足够精度的前提下，力求模型简单、结构清楚，便于使用。

一般说来，为建立一个模型主要任务有两项：确定模型结构和收集数据。确定模型结构时，要根据研究目的确定系统的边界和系统的有关实体、属性和活动。这里所谓的实体（又称成分），系指系统中的一个有意义的研究对象；属性指实体特征的描述变量；活动是指能够使系统状态产生变化的作用过程。因此，所谓确定有关实体、属性和活动，也就是确定模型的参数和变量性质、种类与数量（例如：是随机变量还是确定性变量，它们的取值范围等），以及变量间的相互关系等。而收集数据的任务是为模型结构的确立提供信息，估计和确定变量的特征（如概率分布等），以及建立变量间的关系。实际上，这两项任务是同一工作的两个侧面，二者是密切相关的。通过收集数据和数据分析，不但可以证实或否定在建模时所作的假定，而且往往还可以弄清楚一些未知的或者不完全清楚的系统特性，从而可以重新改变或修正模型结构，使之能够更可靠地和更精确地反映系统的实况。需要指出的是，对于一个复杂的和内部特性不清楚或不很清楚的系统，建模很难一次成功。往往需要借助于仿真技术，反复实验与修改才能得到令人满意的结果。可见，仿真本身又是建模的一个非常重要的技术手段。

§1.3 系统仿真的一般实施步骤

系统仿真的一般实施步骤示于图 1-7。

1. 调研系统，描述问题

对待研究的真实系统进行调查研究，把要研究的问题尽可能描述清楚。明确研究目的和为达到此目的需要做哪些实验和测量，如有可能，还应给出评估系统有关性能的准则。

2. 收集数据，建立模型

这项工作内容前边已经叙述过。这里再补充一点：在建立仿真模型的同时，尽可能也探索一个能够用解析法求解的简化模型。有了它，仿真工作便有了大致可以参考的指导方向。这可减小仿真工作的盲目性。

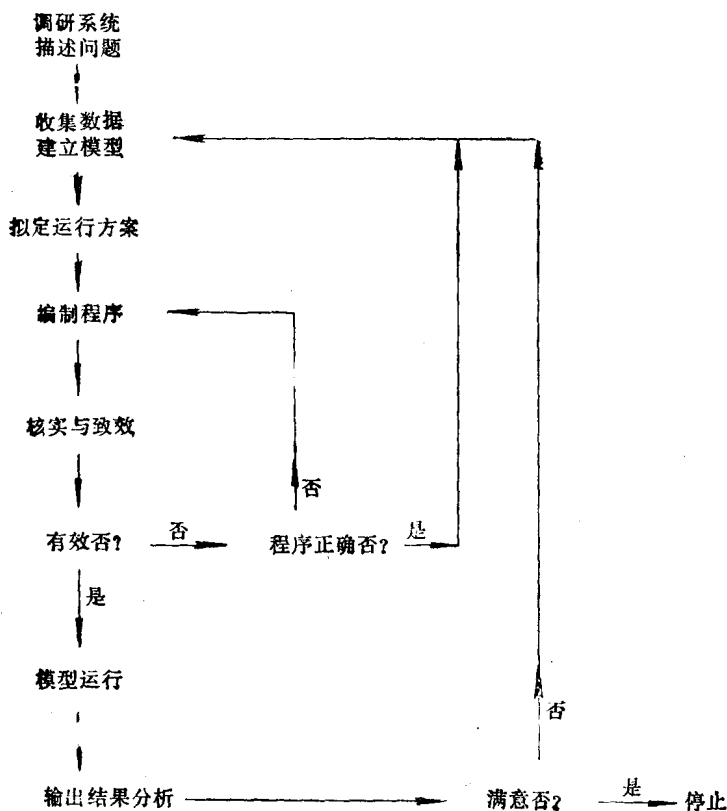


图1-7 仿真实施步骤流程图

3. 拟定运行方案

有了模型便可编程付诸运行。但同一模型可能有多个运行方案，这就需要确定选用哪个方案最省机时，每次运行需要改变哪些参数，初始条件如何预置，独立运行的次数，是否需要采用方差减小技术等。这些工作将有助于估计仿真工作量的大小。

4. 编制程序

绘制仿真程序框图（或流程图）和编写计算机程序。仿真程序既可用通用语言（如 FORTRAN, PASCAL, BASIC 等）编写，也可用专门面向仿真的仿真语言编写。例如用以下仿真语言编写：用于连续系统仿真的 DYNAMO, MIMIC, CSMP 等；用于离散事件系统仿真的 GPSS, GPSS-F, SIMSCRIPT, SIMULA 等；连续-离散混合仿真语言 GSL, CASPIV 等。用仿真语言编写程序可简化程序设计过程，编写出的程序比较简明。编制程序工作通常与拟定运行方案同时进行。

5. 核实与致效

所谓核实就是检查程序是否正确地反映了编程者的意图，进行程序调试。所谓致效就是使模型能有效地代表真实系统。后者是个必须解决但又难以彻底解决的问题。系统仿真的基本原理虽不难，但实施起来却并不容易。并非随意建个模型就能用，而只有致效了的模型才能用。模型的致效是仿真技术中的一个难题。这是因为作为衡量基准的真实系统的行为我们往往并不知道或不完全知道，很多学者正在致力于这一问题的研究。本书将在第七章讨论这个问题。

6. 模型运行

根据所拟定的运行方案，在计算机上进行仿真运行，获取模型行为数据。

模型运行并不是从这一步才开始，实际上从建模之初就不断地需要试运行。特别是程序的核定与模型的致效，需要试运行。如果使用方差减小技术则更需要先导试运行（见第八章）。

7. 输出结果分析

模型往往含有随机变量，这时，其输出结果也是随机的。这就需要用统计技术对正式运行的输出结果进行分析，给出仿真输出示性数的区间估计。如果所得结果令人满意，则结束仿真。否则，应重新建模或修改模型，重复上述过程，直到满意为止。

最后，需要指出：流程图 1-7 虽可直观地表示出系统仿真的各项工作和它们的顺序，但它仅是个示意图，并未把所有情况都反映出来。图中示出的工作顺序也并非一成不变。实际上，有些工作是并行或交错进行的。有的工作则是贯穿整个仿真工作全程的，有的工作则需反复进行。如模型的致效在仿真的全程都得考虑。

§1.4 其 他

§1.4.1 蒙特卡洛 (Monte Carlo) 法

在一些仿真文献中，经常会看到蒙特卡洛法一词。《辞海》中是这样解释的：“蒙特卡洛法又称‘统计试验法’。一种解决数学和物理问题的近似计算方法。将求解的问题置于一定的概率模型之下，用电子计算机来实现统计模拟或抽样，并获得问题的近似解”。有的书上把蒙特卡洛法的定义限制为：用随机模型解决某些时间推移对其不起本质作用的随机性或确定性问题的方法。可见，蒙特卡洛法是一种仿真方法。只不过不管所研究的问题是确定性的或者是随机性的，所建模型必须是概率模型。这种方法或者还只是用来解决静态问题的数字仿真方法。

总之，蒙特卡洛法是包括在系统仿真技术之内的。例如前边提到的打靶问题的仿真，便是一种蒙特卡洛法。

§1.4.2 仿真法与解析法的区别

有很多问题只用解析法便可解决，例如前边提到的火炮射击命中目标的概率，根据概率论，用解析法很容易计算出来。现在用解析法解决问题时，也多用计算机算题。那末这与仿真法又有什么不同呢？用仿真法解决问题时，必须在计算机上模仿出系统行为的全过程。而解析法则不需要。如待解问题是随机性的，则仿真法给出的解是若干现实的统计量，是有一定散布的特解。而解析法给出的解是相对于模型总体的解，是确定性的通解。

解析法虽然有很多优点，但现代很多复杂问题的解决，解析法的成果还不多，因而只好使用仿真法。又因为仿真法在原理上直观，不要求专门的高深数学知识，容易为广大工程技术人员所掌握，所以近年来，随着计算机性能的提高与普及，一些即使是有解析法可用的问题，有时也采用仿真法来解决。

第二章 连续系统仿真

对连续系统仿真,首先要建立描述连续系统的数学模型。一个连续系统的模型通常用微分方程,状态方程,传递函数以及系统的结构图等形式表示。关于这些模型的建立方法,可参阅自动控制理论教材中的有关内容,这里不再重复。

由于本教材重点介绍离散事件系统仿真,所以对于连续系统仿真只是从数值积分法和连续方程离散化(离散相似法)两个方面简要讨论其仿真基本原理,方法和有关程序(通用语言程序)。最后简单介绍一下含有非线性环节的连续系统仿真。

§ 2.1 数值积分法

对于一个连续的数学模型采用数字机仿真,需要将它化成便于在计算机上执行运算的离散形式的模型,如差分方程,离散时间状态方程,离散框图等。下面介绍利用数值积分方法来建立离散形式模型之一——差分方程的方法。

设一阶微分方程及初值

$$\begin{cases} \dot{y}(t) = f[t, y(t)] & a \leq t \leq b \\ y(a) = y_0 \end{cases} \quad (2-1)$$

式(2-1)的解 $y(t)$ 是区间 $[a, b]$ 上连续变量 t 的函数。式(2-1)的数值解法是在若干离散点处,比如

$$a = t_0 < t_1 < \dots < t_n = b$$

计算出解 $y(t)$ 的近似值 $y_0, y_1, \dots, y_n$ 来代替连续变量 $y(t)$ 值。点列 $y_k (k = 0, 1, 2, \dots, n)$ 称式(2-1)在点列 $t_k (k = 0, 1, 2, \dots, n)$ 的数值解,通常取等时间间隔,即 $t_i - t_{i-1} = h (i = 1, 2, \dots, n)$, h 称为步长。可见,数值解法就是首先把一个连续变量问题(2-1),用某种离散化方法(如数值积分法,台劳展开式等)化成离散变量问题——近似的差分方程的初值问题。然后逐步计算出 y_k 。采用不同的离散化方法可得出具有不同精度的微分方程的数值积分方法。关于这方面的问题可参阅计算方法教材中的有关内容,这里不作详细介绍。为了以后讨论方便起见,下面只简要地叙述其中几种常用的数值积分法。

§ 2.1.1 欧拉法

在区间 $[t_k, t_{k+1}]$ 上对式(2-1)求积分得

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f[t, y(t)] dt \quad (2-2)$$

由于上式右端积分中含有未知函数 $f[t, y(t)]$,无法直接得到 $y(t_{k+1})$ 值,我们可以用矩形面积近似代替在该区间上的曲线积分,即

$$\int_{t_k}^{t_{k+1}} f[t, y(t)] dt \approx f[t_k, y(t_k)] \cdot h$$

如果用在 t_k 时刻计算出的近似值 y_k 来代替 $y(t_k)$, 代入式(2-2)右端, 可得

$$y(t_{k+1}) \approx y_k + f(t_k, y_k) h = y_{k+1} \quad k = 0, 1, \dots, n-1 \quad (2-3)$$

式(2-3)为欧拉法计算公式。由于是用一矩形面积代替小区间的曲线积分, 显然精度较低。为了提高计算精度, 可减小步长 h , 这会导致计算次数增加, 这样不仅使计算工作量增加, 而且由于计算机的有限字长而引起的舍入误差, 会因计算次数增加使得累积的舍入误差加大, 故通过减小步长来提高计算精度是有限度的。欧拉法一般用于仿真精度要求不高的地方。

§ 2.1.2 梯形法 (改进的欧拉法)

欧拉法简单, 但精度低, 如果用梯形面积代替每一个小区间的曲线积分, 显然会提高计算精度。梯形法计算公式为

$$y_{k+1} = y_k + \frac{h}{2} [f(t_k, y_k) + f(t_{k+1}, y_{k+1})] \quad k = 0, 1, \dots, n-1 \quad (2-4)$$

上式为含有待求量 y_{k+1} 的方程, 通常解隐含 y_{k+1} 的方程是比较困难的。所以我们首先用简单的欧拉法计算 $y(t_{k+1})$ 的近似值, 用 $y_{k+1}^{(0)}$ 表示, 然后将其代入式(2-4)右端, 计算 y_{k+1} 值。为了提高计算精度, 可反复迭代计算, 于是可得迭代公式如下:

$$\begin{aligned} y_{k+1}^{(0)} &= y_k + f(t_k, y_k) h \\ y_{k+1}^{(1)} &= y_k + \frac{h}{2} [f(t_k, y_k) + f(t_{k+1}, y_{k+1}^{(0)})] \\ &\vdots \\ y_{k+1}^{(i+1)} &= y_k + \frac{h}{2} [f(t_k, y_k) + f(t_{k+1}, y_{k+1}^{(i)})] \end{aligned}$$

直到 $|y_{k+1}^{(i+1)} - y_{k+1}^{(i)}| \leq \varepsilon$ (ε 为给定的允许误差)。如果 $y_{k+1}^{(0)}, y_{k+1}^{(1)}, \dots$, 这个序列是收敛的, 那么就有极限存在, 即 $i \rightarrow \infty$ 时, 该序列趋于某一极限值, 因此可用极限值来作为 y_{k+1} 的值。可以证明, 如果 $\partial f / \partial y$ 有界, 且 h 取得较小, 则上述序列必定收敛。从迭代过程可看出, 每迭代一次, 计算量增加几乎一倍。在实际应用时, 只要 h 取得足够小, 常迭代一次就认为已经求得 y_{k+1} 了。这种迭代一次的计算公式为

$$\begin{cases} y_{k+1}^{(0)} = y_k + h \cdot f(t_k, y_k) \\ y_{k+1} = y_k + h/2 [f(t_k, y_k) + f(t_{k+1}, y_{k+1}^{(0)})] \end{cases} \quad k = 0, 1, \dots, n-1 \quad (2-5)$$

式(2-5)称为改进的欧拉公式, 又称预估-校正公式。式(2-5)中的第一式称预估公式, 第二式称校正公式。

§ 2.1.3 四阶龙格-库塔法

如果要求较高的计算精度, 常采用四阶龙格-库塔法, 这里直接引用其计算公式

$$\begin{cases} y_{k+1} = y_k + h/6 (k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(t_k, y_k) \\ k_2 = f(t_k + h/2, y_k + k_1 h/2) \\ k_3 = f(t_k + h/2, y_k + k_2 h/2) \\ k_4 = f(t_k + h, y_k + k_3 h) \end{cases} \quad (2-6)$$

四阶龙格-库塔法计算精度较高,其截断误差正比于 $0(h^5)$,欧拉法和改进的欧拉法的截断误差分别正比于 $0(h^2)$ 和 $0(h^3)$ 。但是四阶龙格-库塔法计算量较大。

以上介绍的几种数值积分,只是针对一阶微分方程,如果对于高阶微分方程描述的系统,则可将高阶微分方程改化成一组一阶微分方程。每一个一阶微分方程都可用数值积分法计算,这样就可以很容易地将整个系统的动态特性全部计算出来。所以对连续系统仿真,若采用微分方程数值积分法,其基础是一组一阶微分方程或状态方程。

§ 2.2 面向微分方程的通用语言仿真程序

面向微分方程的系统模型通常写成两种形式,一种是直接写成状态方程(或一阶微分方程组),另一种是写成传递函数形式。下面分别介绍这两种模型的仿真程序。

§ 2.2.1 状态方程模型的仿真程序

设系统的状态方程和输出方程为

$$\begin{cases} \dot{X}(t) = AX(t) + Bu(t) \\ y(t) = CX(t) \end{cases} \quad (2-7)$$

式中 $X(t)$ —— n 维列向量;

A —— $n \times n$ 系数矩阵;

B —— $n \times 1$ 控制矩阵(驱动矩阵);

C —— $1 \times n$ 输出矩阵;

$u(t)$ ——输入变量;

$y(t)$ ——输出变量。

式(2-7)为一个单输入单输出的系统状态方程。如果采用四阶龙格-库塔法进行仿真计算,其计算公式可写成

$$\begin{cases} X_{k+1} = X_k + h/6(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = AX_k + Bu(t_k) \\ K_2 = A(X_k + h/2K_1) + Bu(t_k + h/2) \\ K_3 = A(X_k + h/2K_2) + Bu(t_k + h/2) \\ K_4 = A(X_k + hK_3) + Bu(t_k + h) \end{cases} \quad (2-8)$$

应注意的是,这里 K_j ($j = 1, 2, 3, 4$) 为 n 维列向量,即

$$K_j = (k_{1j}, k_{2j}, \dots, k_{nj})^T$$

k_{ij} 表示第 i 个方程 ($i = 1, 2, \dots, n$) 中第 j ($j = 1, 2, 3, 4$) 个 k 系数,例如 $j = 1$ 时, $K_j = K_1$, 而其元素 $k_{1j} = k_{11}$ 表示第一个方程中系数 k_{11} , $k_{2j} = k_{21}$ 表示第二个方程中系数 k_{21} , ...。为了使公式紧凑些,引入一列零矢量 $K_{i0} = 0$ ($i = 1, 2, \dots, n$) 和行矢量 $(h_0, h_1, h_2, h_3) = (0, h/2, h/2, h)$, 于是四组系数计算公式可以合成一个来表示:

$$\begin{aligned} K_j &= A(X_k + K_{j-1}h_{j-1}) + Bu(t_k + h_{j-1}) \\ (j &= 1, 2, 3, 4) \end{aligned}$$

如果已知状态变量的初始值 X_0 和输入函数 $u(t)$, 则可根据式(2-8)递推计算出