

高等学校自动化专业教材

系统建模与辨识

王秀峰 卢桂章 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书比较全面、系统地介绍目前在不同领域中常用的有效建模与辨识方法。主要内容包括:线性系统的辨识,多变量线性系统的辨识,线性系统的非参数表示和辨识,非线性系统的辨识,时间序列建模,房室模型(多用于医学、生物工程中)的辨识,神经网络模型的辨识,模糊系统的建模与辨识,遗传算法及其在辨识中的应用,辨识的实施等。各种方法都给出具体的计算步骤或框图,并结合实例或仿真例子给予说明,尽量使读者易学会用。

本书为天津市高校“十五”规划教材,可作为高等学校自动化、系统工程、经济管理、应用数学等专业的高年级本科生和研究生的教材或参考书,也可作为有关科技工作者、工程技术和管理人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

系统建模与辨识/王秀峰等编著. —北京:电子工业出版社,2004.8

高等学校自动化专业教材

ISBN 7-121-00066-0

I. 系… II. 王… III. 系统建模—高等学校—教材 IV. N945.12

中国版本图书馆 CIP 数据核字(2004)第 062968 号

责任编辑:凌 毅

印 刷:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×980 1/16 印张:16.5 字数:410 千字

印 次:2004 年 8 月第 1 次印刷

印 数:5 000 册 定价:22.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

在中国大百科全书“自动控制与系统工程”卷中对“模型”的解释是：对于现实的事物、现象、过程或系统的简化描述，或其部分属性的模仿。模型不存在一种统一的分类方式，常见的有物理模型、数学模型和仿真模型等。

随着科学技术不断进步和社会经济的不断发展，为了提高决策的科学性和准确性，定量分析的重要性日益显现出来，这就要求对所研究对象的各种量之间建立定量的关系——数学模型。对不同的领域、对象和不同的目的，数学模型的形式和内容也有很大的不同。科学实验中依据物理规律、化学定律等建立精确机理模型比较多，生产过程中根据生产原理和实验数据相结合的半机理模型用得比较普遍，而在社会经济、管理、金融等领域内更多的是用实际数据与经济理论结合而产生的模型。总之，不同的应用需要建立不同类型的模型，建立不同类型的模型需要不同的方法。

建立模型的过程称为“建模”(Modelling)。由于模型在科学技术中的重要作用，而模型所要描述的又是涉及到各个领域的多种多样的过程。因此，在建模过程中也自然会遇到许多困难和问题。比如，有些过程的内在运动机理尚未认识清楚，只能从外在的现象去认识，并在此基础上建模；对有些过程的内在规律虽然有所认识但又过于复杂，因此必须进行合理的简化，并在此基础上建模；有些问题的目的(如预报)并不在于了解过程的内在规律而只关心其外在综合表现的变化，因此，只需基于这种需求建模。总之，建模是一项十分重要而复杂的工作，需要从实际出发，“去粗取精，去伪存真，由表及里”，才能建立一个真正合用的模型。

在建模过程中，除了运用已有的知识对过程本身的规律进行分析外，实验数据是非常重要的依据，即使是分析过程本身的规律，往往也需要实验数据的检验。通过实验数据建模，称为“辨识”(Identification)，它是建模的一条十分重要的途径，这也是本书的主要内容。建模是更广泛的概念，建模与辨识是密不可分的两个范畴，只是在实际工作中，辨识主要依据数据，因此往往用到的算法多，工作量大，似乎是主角，实际上两者是不可或缺、相互依存的。

本书不可能广泛地涉及到建模的所有领域，只是从系统分析、仿真、预测、控制这几个最常用的方面讨论建模与辨识，并以数学模型的辨识为主线，因为这已经是现代控制工程的一个公认的分支。

作者从20世纪70年代末开始从事“建模与辨识”的教学，20多年来科学技术获得了极大的发展，新的科技成果对这一学科方向无疑要产生影响：一方面由于发展需要，建模的对象日趋复杂，一些新的内容、新方法、新理论不断地融入教学内容，如神经网络模型的辨识、模糊系统的建模、遗传算法的应用等；另一方面，一些经典的内容，如线性系统模型辨识、一般非线性系统模型辨识至今仍在辨识中占有重要的地位。本书从这两个侧面选择了作者认为最主要的、最有代表性的内容。

从多年的教学和科研的实践中，作者深深感到“建模与辨识”在科技发展中日益发挥的重要作用，除了在控制工程中的广泛应用外，在经济系统、金融系统、生物医学工程系统、生态系统及其他社会人文系统等中，模型分析的方法都得到普遍的运用，而且越来越受到重视，因此建模与辨识自然也备受关注。

学习“建模与辨识”这门课程，作者认为应该本着“清楚了解背景知识，灵活分析运用方法，

实事求是检验结果”的精神把建模辨识和应用结合起来,全面准确地掌握方法和应用。

本书作为教材,力求把基本概念、基本理论和基本方法讲清楚,并尽量结合实例分析给出方法的具体步骤和应用领域,使读者易学会用。另外,对不同的学时和专业要求,在内容上可以做适当的选择(目录中用 * 号标出)。如第 6 章多变量线性系统辨识本书讨论得比较细致,不一定作为普遍的要求;时间序列模型的辨识可能在经济及其他预测系统用得较多,房室模型则在生物、生态等系统用得较多,等等,这些都可以供内容选择时考虑。

作者期望这本教材对建模与辨识方法的学习有所帮助,从而将通过模型(特别是数学模型)分析系统的思想和方法能为更多的领域运用,同时,也衷心地希望读者在使用的过程中对本书的错误和不足提出批评和建议。

本书作为天津市高校“十五”规划教材出版,得到了天津市教委和南开大学教务处的大力支持和资助;编著过程中研究室的广大师生给予了许多具体帮助;电子工业出版社凌毅编辑为本书的出版做了大量工作,在此特向他们表示衷心的感谢!

作 者

2004 年 5 月于南开大学

目 录

第 1 章 引论	(1)
1.1 建模与系统辨识概述	(1)
1.1.1 系统辨识研究的对象	(1)
1.1.2 系统辨识	(2)
1.1.3 系统辨识的目的	(2)
1.1.4 辨识中的先验知识	(4)
1.1.5 先验知识的获得	(6)
1.1.6 系统辨识的基本步骤	(6)
1.2 数学模型	(7)
1.2.1 概述	(7)
1.2.2 线性系统的 4 种数学模型	(8)
1.3 本书的指导思想和布局	(10)
第 2 章 线性静态模型的辨识	(12)
2.1 问题的提出	(12)
2.2 最小二乘法 (LS)	(13)
2.2.1 最小二乘估计	(13)
2.2.2 最小二乘估计的性质	(15)
2.2.3 逐步回归方法	(16)
2.3 病态方程的求解方法	(17)
2.3.1 病态对参数估计的影响	(17)
2.3.2 条件数	(18)
2.3.3 病态方程的求解方法	(18)
2.4 模型参数的最大似然估计 (ML)	(23)
2.4.1 最大似然准则	(23)
2.4.2 最大似然估计	(24)
2.4.3 松弛算法	(25)
习题	(25)
第 3 章 离散线性动态模型的最小二乘估计	(27)
3.1 问题的提法及一次完成最小二乘估计	(27)
3.2 最小二乘估计的递推算法 (RLS)	(28)
3.2.1 递推最小二乘法	(29)
3.2.2 初始值的选择	(30)
3.2.3 计算步骤及举例	(31)
3.3 时变系统的实时算法	(33)
3.3.1 渐消记忆 (指数窗) 的递推算法	(33)
3.3.2 限定记忆 (固定窗) 的递推算法	(34)
3.3.3 变遗忘因子的实时算法	(35)
3.4 递推平方根算法	(37)

3.5	最大似然估计(ML)	(40)
	习题	(40)
第4章	相关(有色)噪声情形的辨识算法	(42)
4.1	辅助变量法	(42)
4.2	增广最小二乘法(ELS)	(44)
4.2.1	增广最小二乘法	(44)
4.2.2	改进的增广最小二乘法	(44)
4.3	最大似然法(ML)	(47)
4.4	闭环系统的辨识	(49)
4.4.1	问题的提出	(49)
4.4.2	可辨识性	(50)
4.4.3	闭环条件下的最小二乘估计	(51)
	习题	(53)
第5章	模型阶的辨识	(54)
5.1	单变量线性系统阶的辨识	(54)
5.1.1	损失函数检验法	(54)
5.1.2	F 检验法	(54)
5.1.3	赤池信息准则(AIC 准则)	(56)
5.2	阶与参数同时辨识的递推算法	(57)
5.2.1	辨识阶次的基本思想和方法	(57)
5.2.2	阶的递推辨识算法	(59)
5.2.3	几点说明	(61)
5.3	仿真研究	(61)
5.3.1	辨识方法的仿真研究	(62)
5.3.2	对模型适用性的仿真研究	(62)
5.3.3	控制系统设计中的计算机仿真研究	(63)
	习题	(63)
* 第6章	多变量线性系统的辨识	(64)
6.1	不变量、适宜选择路线及规范形	(64)
6.1.1	代数等价系统	(65)
6.1.2	适宜选择路线与不变量	(66)
6.1.3	适宜选择路线与规范形	(69)
6.2	输入/输出方程	(74)
6.2.1	输入/输出方程一般形式	(74)
6.2.2	PCF 规范形对应的输入/输出方程	(76)
6.3	PCF 规范形的辨识	(79)
6.3.1	结构确定及参数辨识	(80)
6.3.2	B^* 和 D^* 的实现算法	(87)
	习题	(93)
第7章	线性系统的非参数表示和辨识	(94)
7.1	线性系统的非参数表示	(94)
7.1.1	脉冲响应函数	(94)
7.1.2	Markov 参数(Hankel 模型)	(95)
7.2	估计脉冲响应函数的相关方法	(97)

7.2.1	相关方法的基本原理	(97)
7.2.2	伪随机二位式信号(M 序列)	(98)
7.2.3	用 M 序列做输入信号时脉冲响应函数的估计	(101)
7.2.4	估计 $h(t)$ 的具体步骤与实施	(102)
习题	(105)
第 8 章	非线性系统辨识	(106)
8.1	引言.....	(106)
8.2	单纯形搜索法.....	(107)
8.2.1	问题的提法.....	(107)
8.2.2	单纯形搜索法.....	(107)
8.3	迭代算法的基本原理.....	(111)
8.3.1	迭代算法的一般步骤.....	(111)
8.3.2	可接受方向.....	(111)
8.4	牛顿—拉夫森算法.....	(112)
8.5	麦夸特方法.....	(113)
* 8.6	数据处理的分组方法(GMDH)	(117)
8.6.1	背景.....	(117)
8.6.2	一般模型结构及基本原则.....	(118)
8.6.3	基本的 GMDH 方法	(119)
8.6.4	变量的预选择.....	(120)
8.6.5	数据的分组和部分实现检验准则.....	(120)
8.6.6	选择层——中间变量的选择.....	(122)
8.6.7	部分实现的形式.....	(122)
8.6.8	GMDH 方法总结及应用	(122)
8.7	NARMAX 模型的辨识	(123)
8.7.1	引言.....	(123)
8.7.2	非线性动态系统的描述.....	(123)
8.7.3	“新息—贡献”准则与矩阵求逆定理.....	(125)
8.7.4	NARMAX 模型的递推辨识算法	(126)
8.7.5	小结.....	(129)
习题	(129)
第 9 章	房室模型的辨识	(130)
9.1	问题的提出.....	(130)
9.2	房室模型的建模.....	(131)
9.2.1	房室.....	(131)
9.2.2	物质转移速度.....	(131)
9.2.3	房室模型.....	(132)
9.2.4	房室模型分类.....	(132)
9.2.5	房室模型建模示例.....	(133)
9.3	参数估计.....	(138)
9.3.1	问题.....	(138)
9.3.2	参数估计算法.....	(138)
9.3.3	参数估计中遇到的几个问题.....	(142)
9.4	可辨识性问题.....	(143)

9.4.1	问题	(143)
9.4.2	一房室模型	(144)
9.4.3	二房室模型	(144)
9.4.4	三房室模型	(145)
9.4.5	多房室模型的可辨识性问题	(146)
9.5	应用实例	(148)
	习题	(153)
第 10 章	时间序列的建模与辨识	(154)
10.1	引言	(154)
10.1.1	模型形式	(155)
10.1.2	格林函数	(156)
10.1.3	稳定性	(158)
10.2	模型的参数估计	(160)
10.2.1	AR(n)模型的参数估计	(160)
10.2.2	ARMA(n, m)模型的参数估计	(160)
10.2.3	初值的求法	(161)
10.3	模型阶的确定	(164)
10.4	确定性的趋向和季节性;非平稳序列	(170)
	习题	(173)
第 11 章	神经网络模型	(174)
11.1	引言	(174)
11.2	神经组织的基本特征和人工神经元	(174)
11.2.1	神经组织的基本特征	(174)
11.2.2	人工神经元的 M-P 模型	(175)
11.3	多层前馈神经网络模型	(177)
11.3.1	前馈神经网络模型的结构	(177)
11.3.2	确定网络模型权值问题的数学描述	(178)
11.3.3	BP 算法	(179)
11.3.4	神经网络的几个有关概念	(180)
11.4	神经网络在辨识中的应用	(181)
11.5	径向基函数网络及其应用	(185)
11.5.1	径向基函数网络的结构	(186)
11.5.2	RBF 网络的辨识	(187)
11.5.3	用 RBF 网络建模实例	(189)
	习题	(192)
第 12 章	模糊建模与辨识	(194)
12.1	模糊集合的基本概念	(194)
12.1.1	模糊集合及其表示	(195)
12.1.2	模糊集的运算	(197)
12.1.3	常用的隶属函数	(198)
12.1.4	模糊逻辑关系的格式	(200)
12.1.5	推理算法	(200)
12.2	基于 T-S 模型的模糊辨识	(201)
12.2.1	结论参数的辨识	(202)

12.2.2	前提参数的辨识	(203)
12.2.3	前提变量的选择	(204)
12.3	应用实例	(207)
12.4	小结	(209)
	习题	(210)
第 13 章	遗传算法及应用简介	(211)
13.1	引言	(211)
13.1.1	遗传算法的基本思想	(211)
13.1.2	基本遗传算法的工作步骤	(211)
13.2	遗传算法的计算机实现	(213)
13.3	遗传算法的工作过程举例	(216)
13.4	实数编码遗传算法	(224)
13.4.1	实数编码遗传算法及其在神经网络训练中的应用	(225)
13.4.2	结果与讨论	(227)
13.5	关于遗传算法的几点说明	(229)
13.5.1	模式定理	(229)
13.5.2	GA 算法的收敛性分析	(229)
13.5.3	GA 是一种优化算法的方法论	(230)
	习题	(230)
第 14 章	辨识的实施	(231)
14.1	辨识的实验设计	(231)
14.1.1	问题的提出	(231)
14.1.2	设计准则	(231)
14.1.3	输入信号的设计	(233)
14.1.4	采样区间的设计	(236)
14.2	模型适用性检验	(238)
14.2.1	通过先验知识的检验	(238)
14.2.2	通过数据的检验	(238)
14.2.3	通过对预测误差序列的检验	(238)
14.2.4	应用实例	(239)
	习题	(240)
附录 A	矩阵运算的两个结果	(241)
附录 B	矩阵微分的几个结果	(242)
附录 C	伪随机二位式序列	(244)
附录 D	正态伪随机数	(245)
参考文献	(246)

第 1 章 引 论

1.1 建模与系统辨识概述

本节里对建模与系统辨识的一些一般性问题,诸如什么是系统辨识、系统辨识的研究对象、系统辨识与建模的关系、系统辨识能解决什么问题,以及系统辨识执行过程中的一般概况等进行一些讨论,作为全书的一个引子。

1.1.1 系统辨识研究的对象

将研究的对象看成是一个系统,从整体的行为上进行研究这是现代科学技术中的一种十分重要的方法论。所谓“系统”,按通常的意义去理解,就是按某种相互依赖关系联系在一起的客体的集合。

这里研究的“对象”是抽象的,重要的是其输入、输出关系,因此我们只需去处理“因果关系”,这样就可以将许多不同的问题进行统一的处理。

对一个系统,我们只着眼于其整体行为,而不一定去细究产生这些行为的内在机制。当然,这种整体行为的研究又可以启发人们对内在结构的进行更深入的认识。对系统的性能与行为进行分析、综合、控制、决策……这些就是系统科学所要研究的问题。

不论是对系统进行分析、综合、控制还是决策,都要了解或者为了达到某种目的而去改变系统的各个变量之间的因果关系。为了实现这些目的,就需要对这些因果关系给以定量的描述。将这些关系用数学关系式定量地表示出来,就是系统的数学模型。因此,只有建立合适的数学模型,才有可能对系统进行有效的定量研究。

这里要特别指出的是,数学模型的主要功能是用来模拟系统的行为,而不是去描述系统的实际结构,正如可以用一个电子系统去模拟人的心血管系统、一个假肢可以完成人的肢体功能而在结构上却完全不同一样,数学模型也有类似的特点。至于究竟要建立什么样的模型,还是要取决于建立模型的实际背景,即有多少可利用的先验知识、有什么可采用的建模方法、所建立的模型要达到什么目的等。

在工程系统中研究的对象可以是待设计的系统,也可以是一个已运行的现有的系统。

在本书中讨论的对象都是数学模型,“模型”这个概念应用已经十分广泛,有些模型远不是一般数学表达式所能表示的,这些模型本书中都不予讨论。

建立系统的数学模型,简称“建模”(Modelling),它是研究系统的重要手段。建模是一个很广泛的概念,凡是用一个数学模型去表示系统的某种因果关系都属于建模的范畴,建模涵盖了将一个实际系统模型化的过程。至于如何去实现这一过程,可以用各种不同的手段与方法,可以通过对系统本身的运动规律的分析,可以根据已有的关于系统的知识和经验,也可以通过对实验数据的处理,当然也可以同时兼用几种方法。

在科学研究和工程实践中,实验和观测总是人们研究一个系统的最重要手段之一。实验的结果是输入和输出的数据,通过这些数据去建立数学模型就是系统辨识(有时简称为辨识)。

1.1.2 系统辨识

系统辨识可定义为:根据输入与输出数据在指定的一类系统中选择一个系统,这个系统和所研究的实际系统等价。

从这个定义出发,辨识首先要给定一类系统 $\bar{S} = \{s\}$,至于对一个具体的辨识问题如何选择 \bar{S} ,就必须依赖于对该系统及其所处环境的知识;其次,还需要有一类输入信号 u 及与之相对应的输出信号,也就是给出取得输入、输出数据的手段与途径;最后还需要选定一个判断真实系统与辨识得到的系统是否等价的准则。

以上 3 个方面都有各种不同的形式与内容,这样就形成了多种多样的辨识问题。由于处理的问题不同,也就带来了处理方法上的千变万化。

具体实施一个辨识问题,大致要经过以下的步骤:

- ① 模型类的选择;
- ② 实验设计;
- ③ 参数估计;
- ④ 模型核验与确认。

模型类的选择主要取决于对所辨识的对象已有的知识及建模的目的,这些对辨识而言均属于先验知识,当然现在也有讨论结构辨识的,但这只是在模型类选定之后辨识其中的一些与结构有关的参数而已。

针对所选定的模型类,选择实验条件,设计实验去取得辨识所需要的输入、输出数据。

在取到实验数据后,就要去确定模型,参数估计是确定模型的主要工作,如果有待辨识的结构参数(如模型的阶或其他的结构参数)就首先确定结构参数,然后再估计其他参数。

在确定了一个模型之后,就必须进行核验,考察所得到的模型对辨识的目的是否合用。如果不合用,则必须重新辨识(包括重新取得数据);如果是合用的,就可以将辨识得到的模型确定为最终结果。

以上这些问题就是本书所要讨论的内容。

1.1.3 系统辨识的目的

弄清楚系统辨识的目的是什么,对我们正确地运用辨识的工具是有重要意义的,系统辨识至少有以下几方面的作用。

1. 估计某些具有特定物理解释的参数

这是辨识用于很大一类对象,特别是用于非工业对象所要解决的问题。在这一类问题中,系统的分析或评价往往依赖于几个主要的参数,而这些参数又是不能直接测量的,这就需要通过一定可以直接测量的实验数据,用系统辨识的方法建立模型,从而估计到这些参数,为系统的分析和评价提供依据。

【例 1.1】 地下水流模型

一个地下水的积水层的水流量可用下面的模型来描述

$$S \frac{\partial h}{\partial t} = \operatorname{div} T \cdot \operatorname{grad} h + q$$

式中, h 是水位, q 是输入流速, S 是存储系数, T 是输送能力, S 和 T 是土质特征的函数。

辨识的目的是估计 S 和 T , 因为它们对积水层的性质提供了一个实际的观察。例如, 一个区域的 T 值高, 就意味着能够提供较大的水量。

【例 1.2】肾血流量的估计

这里考虑的是研究肾功能的生理系统, 肾是排泄器官, 血液经过血管进入肾脏, 通过肾小球、肾小管与血液进行物质交换, 最后形成尿排出。

肾能将血液中的物质清除出去的能力是肾的主要功能, 这样的生理系统可以描述为: 系统中物质运动的关系如图 1.1 所示, 不同区域间物质转移的速度与该区域物质的浓度 x_i 成正比, 速度常数为 k_{ij} , 这样的系统可以根据物质平衡定律写出数学模型为

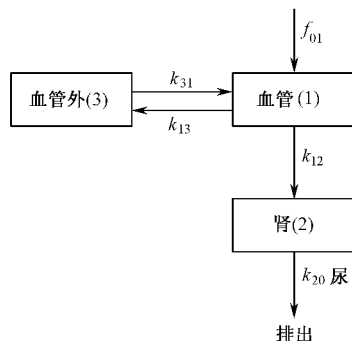


图 1.1 肾功能系统

$$\begin{cases} \frac{dx_1}{dt} = -(k_{13} + k_{12})x_1 + k_{31}x_3 + f_{01} \\ \frac{dx_2}{dt} = k_{12}x_1 - k_{20}x_2 \\ \frac{dx_3}{dt} = k_{13}x_1 - k_{31}x_3 \end{cases}$$

当系统的输入是一脉冲时, $f_{01} = 0$, 初始状态是 $x_1(0) = 1, x_2(0) = x_3(0) = 0$ 。研究肾功能的主要参数是 k_{12} , 通过它可以直接算出肾血浆流量, 通过系统辨识可得到参数 k_{12} 。

类似的例子在生物系统中是很多的, 通过系统辨识, 可以估计一些无法直接测量到的生理参数, 关于本例中的系统将在第 9 章详细讨论。

2. 得到仿真模型

为了研究不同的控制策略或待设计变量对所研究的系统的影响或分析对系统进行某些扰动后可能产生的影响, 这样的工作最好是对系统本身进行实验研究, 但这往往是很难做到的, 原因是多方面的, 比如:

- ① 实验费用可能太昂贵;
- ② 系统可能是不稳定的, 实验将破坏系统的平衡, 可能会造成危险;
- ③ 系统的时间常数很大, 使实验进行得很慢, 以至于效果不明显;
- ④ 系统不可能在设计控制系统时运行, 如所研究的是一个待设计的系统。

在这样的情况下, 只能寻求另外的途径, 途径之一就是建立一个描述系统的数学模型, 用这个模型的运算结果来模拟真实系统的行为。仿真技术已经是现代工程技术中的一种十分重要的、不可或缺的手段, 下面举例来说明。

【例 1.3】生物甲烷化过程

生物甲烷化是一个绝氧发酵过程, 由于细菌的分解作用而产生甲烷这样的一个生物反应, 可用下述非线性状态模型来描述

$$\begin{cases} \frac{dx}{dt} = -(k + g)x + \frac{\mu s}{k + s} \\ \frac{ds}{dt} = \frac{\mu s}{\alpha(k + c)} - gs + pq \end{cases}$$

输出变量是

$$q = \alpha^* \frac{\mu s}{k + s}$$

式中, x 为产生甲烷的细菌的浓度; s 为作用物的浓度; q 为产生的气体; g 为稀释速度; u 为在流体中的酸浓度; k 为细菌的衰减系数; p 为在反应中氢化的酸的未知部分; μ 为给定的细菌生长速度; α 和 α^* 为生产系数。

这个模型是根据生物化学的知识得到的, 其中一些参数是通过测量数据估计而得到的。也就是说, 这个模型就是辨识的结果, 利用这个模型可以寻求最优稳态值, 并且可以研究各种起动方法, 而这些研究几乎不可能在系统自身上完成, 因为目前技术水平很难保持过程处于稳定。此外, 生物甲烷化反应的起动过程需要几个星期的时间, 而利用这个模型在计算机上仿真就非常快。

凡是所研究的系统是一个待设计的系统时, 无论是进行分析还是设计, 只有进行仿真。例如, 核反应的控制系统就是如此, 这时要求仿真所用的模型能真实反映系统的特性。对于设计参数, 则要求能正确地符合其本身的物理意义。

3. 预测

在许多应用中, 要求用现在和过去系统可测量的输入、输出数据实时预测系统输出未来的演变。例如, 最常见的气象预报、洪水预报, 工程系统中的状态预测, 经济学领域中市场价格预测, 环境系统中污染情况的预测……, 总之, 预测问题涉及到许多领域, 预测问题的研究有着广泛的应用前景。

建立预测模型主要是依据预测误差最小的原则, 一般来说, 只要能有较小的预测误差, 就是一个好的预测模型, 对模型的结构及参数上很少有其他要求, 这就使得建立一个预测模型有可能会比建立其他的模型要简单一些。

例如, 常常用输入、输出的一个线性函数去预测未来的输出, 甚至当系统是时变和非线性时也是如此。只要对预测模型的系数以适当的方式不断进行修正, 就可能获得很好的预测效果。

4. 控制

要对一个系统设计控制器, 就需知道系统的传递函数或状态方程。对于一个实际系统, 传递函数或状态方程一般要通过建模和辨识来得到。在这些应用中, 建立模型是为了控制, 特别是在工业应用中, 目的就是要设计一个控制器。随着设计方法和采用的控制策略的不同, 对模型也将提出不同的要求。

以上 4 个方面是建模和辨识最常见的应用, 明确应用的目的对模型的选择是很重要的, 因为这样就知道辨识会带来什么问题及需要用什么样的先验知识。当然, 有时同一问题中可以同时有几个目标, 在这种场合下建立什么样的模型及采用什么方法就需要具体分析。

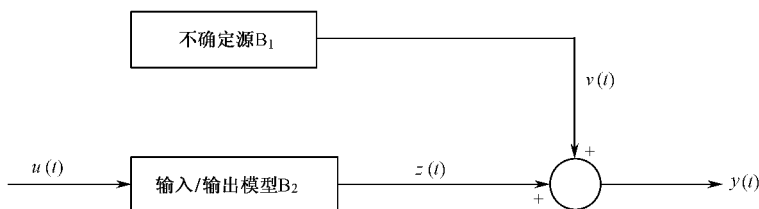
1.1.4 辨识中的先验知识

无论是出于哪一种应用目的, 完全从输入和输出的数据去确定模型总是很困难的。要确定模型, 如何利用有关系统的先验知识是十分重要的, 特别是模型类的选择往往要用到有关系统的先验知识, 下面就这个问题做进一步的讨论。

1. 从先验知识如何选择一个模型类

在建模和辨识中, 存在一个客观的现实系统, 一般地说, 对它的完全准确的描述并不清楚。

尽管如此,还是假定研究者感兴趣的那部分客观现实有一个精确的描述,这个描述(它是研究者对现实的近似)就称为是“真实系统”。为了使这一概念具体化,假定真实系统如图 1.2 所示。



研究者的先验知识就是关于 B_1 和 B_2 的知识,极端情况可以没有任何先验知识。

建模和辨识的任务是选择一个模型类并估计参数,选择模型类是整个建模和辨识过程中最困难的部分之一,为了清楚起见,将模型分成两类。

(1) 预测模型

$$\hat{y}(t+1 | \theta) = f_p(U^t, Y^t, \theta) \quad (1.1)$$

用所有到 t 时刻为止的输入、输出数据去预测将来的输出 $y(t+1)$ 。

(2) 仿真模型

$$\hat{y}(t+1 | \theta) = f_s(U^t, \theta) \quad (1.2)$$

仅用输入数据预测 $y(t+1)$ 。

注意:这两类模型的参数都可以用预测误差最小的方法来辨识。

2. 先验知识在各类模型的辨识中的作用

(1) 预测模型

其目的是构造一个模型,从输入、输出数据来得到预测 $\hat{y}(t+1)$ 。这样的模型不需要多少真实系统的知识,就可以辨识并对其性能进行评价。这种评价的准则和辨识准则是一致的。当然,有关 B_1 和 B_2 的任何知识,将有助于选择模型集,而且将会改进预测性能。

(2) 仿真模型

这里的问题是构造一个模型能模仿真实系统的行为,甚至在与收集数据时不同的环境中也能做到这一点。这样的模型要求就比较高。当然,可以考虑先辨识一个预测模型(这样的模型要求较少的先验知识),然后将预测模型中的 Y^t 用其预测值 \hat{Y}^t 代替,于是模型就转化为仿真模型(见式(1.2))。但是,这种做法是很危险的。因为一个最好的预测模型并不总是一个最好的仿真模型。出现这种现象的原因是预测模型的选择准则与辨识准则是一致的,即使预测误差的某个函数最小。而对仿真模型来说,模型选择的准则与辨识准则不一致,辨识准则不能判别一个仿真模型的好坏,这时只有用先验知识或大量的仿真研究才能确认一个仿真模型的好坏。

(3) 估计具有物理解释的参数

选择模型是为了估计某些参数,而这些参数是不能直接测量的,选择模型类的准则是估计值 $\hat{\theta}$ 必须尽可能接近真值 θ 。这里又出现模型选择的准则与辨识准则不一致的情况。因此,参数和模型结构关系的先验知识是本质的,研究者必须知道或假定一个“真实系统”(至少是包含有关参数的部分真实系统);另一方面,模型集必须选得使 $\hat{\theta}_N$ 收敛到真值 θ 。因此,它必须包含这个真实系统并且必须是可辨识的。粗略地说,这意味着所选的模型对某一 θ 必须有同

真实系统同样的输入、输出关系,这里输入理解为包含确定性输入和随机输入。

可以通过预测误差去讨论参数估计的收敛性,并用估计误差协方差阵去讨论估计的精确度,当然这仅在真实系统的结构等事先已知时才有意义,如果关于 θ 的某些先验知识可利用,则可以提高估计的精度。

(4) 控制

对多数的控制应用,如最小方差控制等,要求有一个好的预测,甚至模型类是否包含真实系统都是不重要的。然而,如果目的是去模拟控制策略,则必须要有一个仿真模型。

总之,系统辨识必须提出所追求的目标及关于真实系统可用的先验知识,追求的目标决定最少需要多少先验知识,先验知识是十分重要的,它是建模的重要依据。因此,工作中要十分重视先验知识,要重视正确的先验知识的搜集和利用。

1.1.5 先验知识的获得

先验知识有两种不同的含义。其一是有关模型结构、参数或者数据的实际知识或信息,而这些知识或信息是研究者已知的或是从数据中收集到的;其二是研究者在从事辨识工作之前所做的假定(这样的先验假定往往是为了保证可辨识性而要求的),这种假定是关于模型结构、参数或数据的某些限定或约定。下面分别进行讨论。

1. 关于模型结构的先验知识

模型结构可以从本质上不同的两条途径来得到:一条是从过程本身的运动(物理的、化学的等)规律来得到的,通过这种途径建模,一般称为理论建模;另一条途径是通过对数据分析得到的,如判断线性、时变性、噪声的相关性等。

2. 关于数据的先验知识

关于数据的先验知识有两种:一种是对数据限制,例如流量必须是正的,控制输入有一定的幅度限制;另一种限制是要求噪声服从某种事先指定的分布,最常见的是要服从正态分布,因为许多分析、估计算法都是以正态噪声为基础的。如果数据不满足要求,就要对数据进行处理,如做变换、取对数就是常用的一种。

3. 关于参数的先验知识

这种先验知识有 3 种形式:①给出参数的区域或者给出少数决定参数变化范围(常常是高维的)的约束变量(常常是低维的);②提供参数的先验估计,这种估计可以作为估计算法的初值;③给定或假定参数的先验分布密度,这种分布将有助于改进估计的精度。

1.1.6 系统辨识的基本步骤

一个实际的辨识过程究竟应该如何去做呢?辨识的主要步骤可以用如图 1.3 所示的流程图来表示。

先验知识和建模的目的前面已经讨论,这里对其他部分做一简单的说明。

1. 实验设计

安排实验的目的是为了获得输入和输出数据。设计实验原则是所获得的数据(对研究目

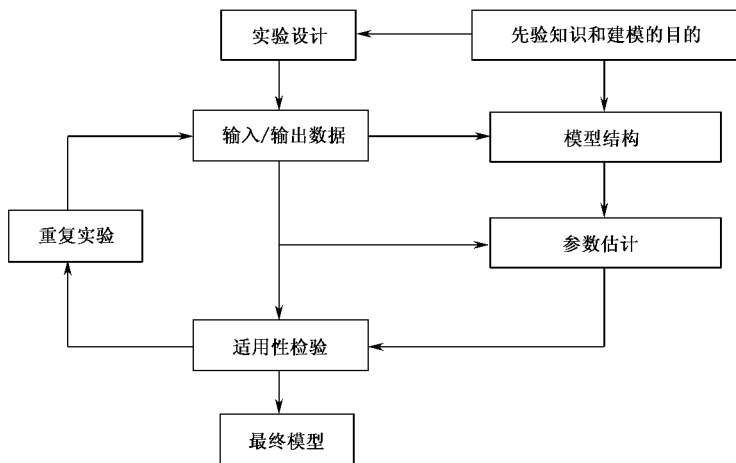


图 1.3 辨识过程框图

的来说)包含的信息要尽可能地多。当然,还要考虑到实现的可能性及实验费用等。

实验设计中要考虑的问题包括:在系统输入允许的情况下,如何选择输入信号、采样速度、实验期限及做哪些预备性实验等。

2. 模型结构

除了单变量线性系统的阶、多变量线性系统的结构不变量(见第 6 章)和一些逼近算法的近似模型外,模型的结构主要是依靠先验知识来决定的。

3. 参数估计

模型结构确定后,辨识主要就是通过实验数据去估计未知参数。因此,参数估计是辨识工作中非常重要的内容。一般来说,参数估计是一个具体的辨识过程中工作量最大的部分,也是以后各章讨论的主要内容之一。

4. 模型适用性的检验

一个模型被辨识出来之后,检验模型是否达到建模的目的的是一个很重要的问题。解决这个问题目前还没有什么系统的方法,只是随着问题的不同而提出具体的解决办法。

1.2 数学模型

1.2.1 概述

分类就是根据事物的特征将具有某种共性的对象归为同一类。这样归并的目的当然可以是多种多样的。一般来说,分类要有助于某种研究目的的实现,对数学模型的分类也是如此。从不同的目的出发,就可能得到不同的分类,如常见的线性与非线性、集中参数与分布参数、定常与时变、确定性与随机、参数与非参数、连续与离散、单变量与多变量、模糊与非模糊、神经网络模型与符号逻辑模型等。

这些分类都是正确的,但又都是不全面的,是否合适取决于研究的目的和观察的角度。在本书中,选择了几种最常用的模型,针对这些模型来讨论辨识问题,它们之中有的是针对一类模型的,如非线性、非参数等;也有的是针对某一特殊模型,如房室模型。我们这里不详细讨论数学模型的各种分类,因为这样做对后面的讨论并不重要。

线性系统是目前研究得最多也是研究成果最丰富的一类系统。由于这一类系统在实际应用中的重要性,所以从许多不同的角度对它进行研究,因此也就有多种不同的描述方式,下面对这个问题进行简要的讨论。

1.2.2 线性系统的 4 种数学模型

描述线性系统常用的有 4 种类型的数学模型:传递函数(阵),时域输入/输出的关系,状态方程和脉冲响应函数(阵)。下面分别进行讨论。

1. 描述系统的各种变量

【例 1.4】 用电枢电流控制直流电机的转角和角速度系统,系统如图 1.4 所示。

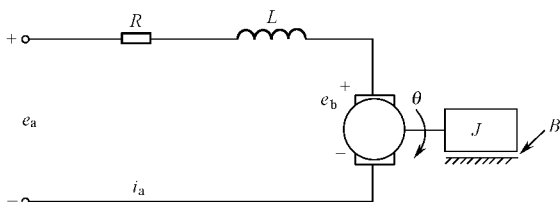


图 1.4 电枢电流控制系统

电机有一惯性负载和一黏性摩擦负载,假如电机输出力矩是与电枢电流成正比,比例常数是 K_T ,及电动势与角速度成正比,比例常数是 K_g ,即 $e_b = K_g \dot{\theta}$ 。图中各量是: J 为惯性负载; B 为黏性摩擦系数; R 为电枢回路电阻; L 为电枢回路电感; e_a 为输入电压; e_b 为电枢反电动势; i_a 为电枢电流。

这个系统的力学部分,用牛顿第二定律有

$$J\ddot{\theta} = -B\dot{\theta} + K_T i_a \quad (1.3)$$

对电枢电流用基尔霍夫定律有

$$L \frac{di_a}{dt} + R i_a + e_b = e_a \quad (1.4)$$

其中 $e_b = K_g \dot{\theta}$ 。

假定 i_a 和 θ 是可检测的。这样一来,描述这个系统的变量有 $i_a, \theta, \dot{\theta}, e_b$, 这些变量在系统中的地位和作用是不同的。 e_a 是能由外界控制并作用于系统的,称为输入变量或控制变量, i_a, θ 是输出变量,输入、输出变量构成了系统的外部特征。但是对一个系统而言,仅研究其外部特征又是不够的。令

$$f_1 = -B/J, f_2 = K_T/J, f_3 = -K_g/L, f_4 = -R/L, c = 1/L$$

$$x_1 = \theta, x_2 = \dot{\theta}, x_3 = i_a, u = e_a$$

则系统的运动方程可写成

$$\begin{cases} \frac{dx_1}{dt} = x_2 \\ \frac{dx_2}{dt} = f_1 x_2 + f_2 x_3 \\ \frac{dx_3}{dt} = f_3 x_2 + f_4 x_3 + cu \end{cases} \quad (1.5)$$

如果已经知道了系统的输入、输出变量的值,即 $x_1(t_0), x_3(t_0), u(t_0)$ 及 $t_0 < t \leq T$ 时间内的输入 $u(t)$, 并不能完全决定某个时刻 t 的输出变量值 $x_1(t)$ 和 $x_3(t)$, 也就是不能完全决定系统的运动。如果将研究范围扩大到除 x_1, x_3, u 以外, 包括 x_2 , 则系统的运动就完全决定了。

x_1, x_2, x_3 称为状态变量, 它们的一般定义是: $\mathbf{x}^T(t) = (x_1(t), \dots, x_n(t))$ 是描述系统的一组变量。如果在 $t=t_0$ 时, $x(t_0)$ 已知, 且对 $t \geq t_0$, 输入变量 $u(t)$ 已知, 则 $x(t)$ 惟一确定, 具有这种性质的 $x(t)$ 称为状态向(变)量。

需要注意的是: 对于一个给定的系统, 状态变量可以是不惟一的。在例 1.4 中, 除 $\theta, \dot{\theta}, \ddot{\theta}$ 外, 也可以选 $\theta, \dot{\theta}, \ddot{\theta}$ 作为状态变量, 但是输入、输出变量由系统决定, 不能任意选择, 状态变量可能有一部分就是输出变量, 也可能全部都是输出变量, 用状态变量来描述的系统特性是内部特性。

按照使用的变量及描述方式, 线性系统有 4 种常用的数学模型。

2. 状态方程

表示状态变量随时间演变规律的方程式就是状态方程, 一般的形式是

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (1.6)$$

式中, $\mathbf{x}(t)$ 是 n 维状态向量, $\mathbf{u}(t)$ 是 m 维输入(控制)向量, $\mathbf{A}(t)$ 和 $\mathbf{B}(t)$ 分别是 $n \times n$ 和 $n \times m$ 维时变矩阵。对于定常线性系统, $\mathbf{A}(t), \mathbf{B}(t)$ 均为常数矩阵 \mathbf{A}, \mathbf{B} , 则式(1.6)可变为

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1.7)$$

由于状态变量不一定是能检测的, 所以为了与输入、输出数据相联系, 在研究状态方程的同时, 还需要考虑输出方程。 $\mathbf{y}(t)$ 是 r 维输出向量, \mathbf{C} 是 $r \times n$ 维矩阵, 则

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (1.8)$$

如果对输入 $\mathbf{u}(t)$ 存在瞬时响应, 则输出方程变为

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (1.9)$$

对于离散时间系统, 与式(1.6)和式(1.8)相对应的模型为

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (1.10)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (1.11)$$

如果系统本身含有噪声源且存在观测噪声, 则式(1.10)和式(1.11)可变成下列随机状态方程

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \quad (1.12)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{v}(k) \quad (1.13)$$

其中, $\mathbf{w}(k), \mathbf{v}(k)$ 均为随机噪声序列。

对于连续时间的随机线性系统, 要描述其状态的运动就需要用到随机微分方程的知识, 由于本书不讨论这类模型的辨识, 所以就不再做进一步的讨论。

由于状态变量不是惟一的, 因而状态方程也就不惟一。这一点就给建模和辨识带来了一些问题, 因为辨识所针对的模型是要有确定的含义, 在讨论状态方程的辨识时, 首先要讨论辨识什么样的状态方程, 这是一个相当麻烦的问题。辨识所依据的是输入、输出数据, 一般来说, 直接得到的是输入/输出关系, 而输入/输出关系并不一定能惟一决定状态方程的形式, 详细的讨论见第 6 章。

3. 时域输入/输出方程

如果着眼于系统的外部特征,即系统的输入/输出关系,则可用输入/输出方程。在例 1.4 中,式(1.3)和式(1.4)就是输入/输出方程,它是关于输出 $i_a \cdot \theta$ 的微分方程。对于离散时间系统,输入/输出方程则为一个差分方程。

4. 传递函数(阵)

传递函数是大家所熟知的,它是输入/输出关系的频域表示,它的定义是

$$G(s) = \frac{\bar{Y}(s)}{\bar{U}(s)}$$

式中, $\bar{Y}(s)$, $\bar{U}(s)$ 分别是输出、输入的 Laplace 变换。关于传递函数,在任何一本关于经典控制理论的书中都有详细的讨论,本书重点不在于讨论传递函数的辨识,所以就不做更详细的讨论。

很明显,已知状态方程式(1.7)和式(1.9),很容易直接得到相应的传递函数阵为

$$G(s) = C^T (sI - A)^{-1} B + D \quad (1.14)$$

反过来,如果已知系统的传递函数阵(或时域的输入/输出方程),如何找出一个状态空间模型(包括选择适当的状态变量并写出状态方程)使得由它决定的传递函数阵(或输入/输出方程)就是所给定的。这是一个复杂的问题,因为输入/输出关系只给出系统的外部信息,所以它不能惟一决定状态方程。如何解决这个问题呢?这就是代数系统理论的核心问题——实现问题。由于本书的篇幅及内容的限制,实现问题在后面章节只能稍有涉及,但不可能做详细讨论。

5. 系统的非参数表示——脉冲响应函数

作为叠加原理的一个结果,定常线性系统的动态行为可以用脉冲响应函数 $h(t)$ 来描述,即输入/输出关系在零初始条件下可以用一卷积分

$$y(t) = \int_0^\infty h(\tau) u(t - \tau) d\tau \quad (1.15)$$

来表示,如果输入是一脉冲函数 $u(t) = \delta(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases}$, 则

$$y(t) = \int_0^\infty h(\tau) \delta(t - \tau) d\tau = h(t) \quad (1.16)$$

式(1.15)和式(1.16)对脉冲响应函数的物理意义做了解释,即 t 时刻输入一个脉冲时系统的响应。由于物理的可实现性,当 $t < 0$ 时, $h(t) \equiv 0$, 即在激励之前没有响应。

1.3 本书的指导思想和布局

正如前面所讨论的,建模和辨识是紧密相关的两个领域,其核心问题就是建立系统的数学模型。当然这是一个困难的问题,因为任何实际的系统都是一个客观的实体,它表现现实世界中的某种活动,与这种活动相联系的数学模型究竟应该是什么?这是至今在理论上未能得到完全回答的问题。尽管如此,今天在系统科学的各个领域和控制工程中广泛地使用着各种各样的数学模型,或者是根据实际的需要用输入、输出数据得到的模型或者是两者结合而得到的模型,这些模型都是在一定程度和一定范围内对真实系统的一个近似描述,建模和辨识就是为

建立这种近似模型提供各种方法及对所得到的结果进行必要的理论分析。

本书始终贯穿的一个主导思想就是如何建立数学模型,我们十分重视辨识中利用先验知识,充分利用先验知识所提供的信息来给出模型的结构。但是,由于建模所涉及的领域是十分广泛的,针对不同的问题就要用到不同方面的知识,所以在较短的篇幅中是不可能进行全面细致的讨论,在本书中将通过几个典型的实例对建模的思路与过程进行剖析。这样做希望能对其他的建模工作有所启发。

另一方面,对各类模型的参数估计是目前在文献中讨论得最多的问题,也是成果最丰富的部分,因此,这部分内容在本书中也占有相当大的比重,所讨论的参数估计方法都是在辨识的工作中常用的,对这些算法也给出了必要的理论分析。

在这里还有一些看法要提出来请读者注意,目前有一些书籍和教科书把辨识的重点完全放在离散时间的线性差分方程模型及其参数的递推估计上,当然这部分内容已经研究得比较清楚,结果比较深入,讨论这些问题当然是无可非议的。但是,把辨识仅限于此或主要限于此就不一定妥当了。建模的范围很广,模型的种类很多,应用模型的目的是多种多样的。线性差分方程仅是其中一种,而且这种模型的应用范围有很大的局限性。一般来说,它只适合于做预测模型,所以仅限于此就有失辨识之真谛。此外,有一种习惯,往往是将建模和辨识归入控制理论,顾名思义主要应用范围是控制工程,特别是工业装置的控制,可事实上在工业装置的控制中,辨识的应用并不是很广泛,除了在校正控制系统中的预测模型的参数用递推估计外,典型的应用就不很多,但是在非工业系统中,近年来辨识却得到了广泛的应用。如社会经济系统、生物医学系统、生态系统、环境系统等,建模的目的有用于系统分析预测,有用于估计特殊物理意义的参数,也有用于系统的调控。在这些特殊复杂系统的问题中,给建模和辨识提出了许多有兴趣的课题,如理论建模、可辨识性、适用于特殊情形的辨识方法……,这些都有力地促进了建模和辨识的发展。所以,工程系统以外的其他领域的辨识必须给予充分重视。

本书的布局是这样考虑的。第1章概述建模和系统辨识的基本思路和概况,因为是概述,细节上不可能谈得很深入,所以在对辨识的许多具体内容有所了解以后再来读这一部分,可能会帮助读者抓住问题的本质。第2~5章是线性模型的辨识,第6章是多变量线性系统的辨识,包括结构和参数的辨识,给出了整套算法,在随后的几章中讨论非参数模型、非线性模型、时间序列模型,以及一类特殊的模型——房室模型的辨识。第11~13章讨论近年来发展非常迅速的软智能方法(如神经网络、模糊系统、遗传算法等)在建模和辨识中的应用,通过这些内容可以看出辨识应用(特别是非工业应用)的广泛性,并讨论各种行之有效的辨识方法,同时在这几章中也通过一些实例剖析建模的一般过程。最后一章是系统辨识的实施,介绍辨识的实际工作过程。

第2章 线性静态模型的辨识

2.1 问题的提出

线性静态模型,也称稳态模型,因为它考虑的是在同一瞬时过程的输入和过程响应之间的关系,模型不涉及这些变量随时间的演变。所讨论的变量可以有动态背景,即是一个动态过程的瞬时采样值,也可以没有任何动态背景。在数理统计学中,这样的模型称为线性回归模型。对这类模型的分析 and 参数估计是数理统计所讨论的问题之一。

在过程控制中,即使是连续生产过程,操作经常是处于稳定状态,因此在这种情况下只需要考虑输入变量和输出变量之间的稳态关系就行了。也就是说,不考虑过程由一个稳定状态变到另一个稳定状态的过程,而只是简单地将同一时刻的输入值和输出值对应起来。

另一类问题本身就是静态的,如轧钢机的辊缝预设定值,它根据轧机的参数、钢环的数据及要求的出口厚度事先计算出各机架的辊缝;又如炼钢炉的静态装料模型也是如此。在社会经济系统的研究中,建立各个变量之间的关系,提供分析、决策使用的也常常是用线性稳态模型。因为一个经济系统的过渡过程的含义经常是不清楚的,所以这种稳态模型常常是合适的。

目前大家所关注的分级递阶系统、高级系统的主要功能是判断决策,而完成这种功能是长周期的,所以经常采用的也是稳态模型。目前正兴起的现代管理中的计划、规划、决策等许多情况,绝大多数也都是采用稳态模型。

此外,在各种研究或工程问题中,常常会出现这样的情形,需要研究的量之间的关系是不清楚的,也就是难以从过程本身运动规律的分析来得到模型,因此只能从实验数据去建立经验模型,而这种经验模型最常用的就是线性稳态模型(因为缺少先验知识,最简单的办法就是假定模型是线性的)。

对于单输出的线性稳态系统来说,先验知识就是已知有 n 个输入变量 $u_i (i=1,2,\dots,n)$ 都是可测的,一个响应 y , 已知 y 和 u_i 的关系是线性的。当然,在实际应用中,有许多情况是因为缺乏其他的先验知识而将此关系简单地处理为线性的。模型的一般形式为

$$y = a_0 + a_1 u_1 + \dots + a_n u_n \quad (2.1)$$

对这样的模型的辨识问题就是通过输入、输出测量数据去估计参数 a_0, a_1, \dots, a_n 。

用来估计参数的数据记为

$$\{u_1(i), \dots, u_n(i); y(i)\} \quad i = 1, \dots, N \quad (2.2)$$

其中每一组数据 $\{u_1(i), \dots, u_n(i); y(i)\}$ 表示在某一个稳态条件下第 i 次采样取得的输入、输出数据, N 是数据(组)的总数,称为数据长度。注意:这些数据之间没有时间上的相依关系,这是与动态系统的观测数据有本质区别的地方。

由于所确定模型结构的近似性及数据的量测误差,所以对于每一组数据都不可能精确地满足式(2.1)。因此实际模型结构应该是

$$y(k) = a_0 + a_1 u_1(k) + \dots + a_n u_n(k) + e(k) \quad k = 1, 2, \dots, N \quad (2.3)$$

其中, $e(k)$ 称为模型的残差(或误差),模型残差体现了数据测量的误差、模型结构误差及其他

原因造成的模型偏差的总和,一般用一随机序列来表示,通常对 $e(k)$ 做如下假定:

- ① $e(k)$ 是零均值、白噪声序列;
- ② 具有相同的方差 σ^2 ;
- ③ 服从高斯(正态)分布。

这些假定将直接影响到参数估计量的性质。

现在的问题是如何通过实验数据去得到 a_0, a_1, \dots, a_n 的估计值。由于这种模型通常是有误差的,所以只能在统计意义下来估计这些参数,最常用的方法就是最小二乘法(Least Squares Method)。

最小二乘法就是求出使实际观测值与模型计算值之差的平方和达到极小的参数值作为所要的估计值。如果由实验取得了数据,如式(2.2)所示,则误差平方和记为

$$\begin{aligned} J(a_0, a_1, \dots, a_n) &= \sum_{k=1}^N e^2(k) \\ &= \sum_{k=1}^N [y(k) - a_0 - a_1 u_1(k) - \dots - a_n u_n(k)]^2 \end{aligned} \quad (2.4)$$

现在的问题就是要求参数的估计值 $\hat{a}_0, \hat{a}_1, \dots, \hat{a}_n$, 使得

$$J(\hat{a}_0, \hat{a}_1, \dots, \hat{a}_n) = \min \quad (2.5)$$

其中, $\{\hat{a}_i\}$ 称为参数 $\{a_i\}$ 的最小二乘估计(Least Squares Estimation)。

2.2 最小二乘法(LS)

2.2.1 最小二乘估计

从上一节讨论知道,求解最小二乘估计就是要使残差平方和 J 达到最小, J 是参数 a_0, a_1, \dots, a_n 的二次函数,因此可以按多元函数求极值的方法求解。

为了推导简便,将问题用向量形式表示。令

$$\begin{aligned} \boldsymbol{\phi}_k &= (1, u_1(k), \dots, u_n(k))^T \\ \boldsymbol{\theta} &= (a_0, a_1, \dots, a_n)^T \end{aligned}$$

则式(2.3)可写为

$$y(k) = \boldsymbol{\phi}_k^T \boldsymbol{\theta} + e(k) \quad k = 1, 2, \dots, N \quad (2.6)$$

再令

$$\begin{aligned} \mathbf{y}_N &= (y(1), \dots, y(N))^T, \mathbf{e}_N = (e(1), \dots, e(N))^T \\ \boldsymbol{\Phi}_N &= \begin{bmatrix} \boldsymbol{\phi}_1^T \\ \boldsymbol{\phi}_2^T \\ \vdots \\ \boldsymbol{\phi}_N^T \end{bmatrix} = \begin{bmatrix} 1 & u_1(1) & \cdots & u_n(1) \\ 1 & u_1(2) & \cdots & u_n(2) \\ \vdots & \vdots & \cdots & \vdots \\ 1 & u_1(N) & \cdots & u_n(N) \end{bmatrix} \end{aligned}$$

则式(2.6)等价于

$$\mathbf{y}_N = \boldsymbol{\Phi}_N \boldsymbol{\theta} + \mathbf{e}_N \quad (2.7)$$

残差平方和 J 可表示为

$$J(\boldsymbol{\theta}) = (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) \quad (2.8)$$

用多元函数求极值的方法(见附录 B), 令

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\boldsymbol{\Phi}_N^T(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta}) = 0$$

得方程组

$$\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N\boldsymbol{\theta} = \boldsymbol{\Phi}_N^T\mathbf{y}_N \quad (2.9)$$

方程组(2.9)称为正规方程组。解此方程组即可得到参数 $\boldsymbol{\theta} = (a_0, a_1, \dots, a_n)^T$ 的估计。

假定 $\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N$ 是满秩的,于是最小二乘估计解的向量形式为

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N)^{-1}\boldsymbol{\Phi}_N^T\mathbf{y}_N \quad (2.10)$$

这种形式对进一步讨论最小二乘估计算法及它的统计性质分析都是十分有用的。但在具体求解一个模型参数的最小二乘估计时,仍然是直接解正规方程组(2.9)。

注 1:在实际辨识过程中,如果数据的量级差别很大,为了提高估计精度,常常先对输入、输出数据实行正规化处理,即进行如下变换

$$\begin{cases} u'_i(k) = \frac{u_i(k) - \bar{u}_i}{\sqrt{s_{ii}}} & i = 1, \dots, n, k = 1, \dots, N \\ y'(k) = \frac{y(k) - \bar{y}}{\sqrt{s_{yy}}} & k = 1, \dots, N \end{cases}$$

其中

$$\begin{aligned} \bar{u}_i &= \frac{1}{N} \sum_{k=1}^N u_i(k), & s_{ii} &= \frac{1}{N} \sum_{k=1}^N [u_i(k) - \bar{u}_i(k)]^2, i = 1, 2, \dots, n \\ \bar{y} &= \frac{1}{N} \sum_{k=1}^N y(k), & s_{yy} &= \frac{1}{N} \sum_{k=1}^N [y(k) - \bar{y}]^2 \end{aligned}$$

这时模型结构式(2.3)变为

$$\begin{aligned} y'(k) &= a'_1 u'_1(k) + \dots + a'_n u'_n(k) + e'(k) \\ a_0 &= \bar{y} - a_1 \bar{u}_1 - \dots - a_n \bar{u}_n \end{aligned} \quad (2.11)$$

若令

$$\begin{aligned} \boldsymbol{\phi}'_k &= (u'_1(k), \dots, u'_n(k))^T, \boldsymbol{\theta}' = (a'_1, \dots, a'_n)^T \\ \boldsymbol{\Phi}'_N &= (\boldsymbol{\phi}'_1, \boldsymbol{\phi}'_2, \dots, \boldsymbol{\phi}'_N)^T \\ \mathbf{y}'_N &= (y'(1), \dots, y'(N))^T, \mathbf{e}_N = (e(1), \dots, e(N))^T \end{aligned}$$

参数 $\boldsymbol{\theta}'$ 的最小二乘估计由解如下正规方程得到

$$\boldsymbol{\Phi}'_N{}^T\boldsymbol{\Phi}'_N\boldsymbol{\theta}' = \boldsymbol{\Phi}'_N{}^T\mathbf{y}'_N \quad (2.12)$$

$$\hat{\boldsymbol{\theta}}' = (\boldsymbol{\Phi}'_N{}^T\boldsymbol{\Phi}'_N)^{-1}\boldsymbol{\Phi}'_N{}^T\mathbf{y}'_N$$

原模型式(2.6)的参数 $\boldsymbol{\theta}$ 的估计值与正规化后参数 $\boldsymbol{\theta}'$ 估计值之间满足如下关系

$$a_i = \frac{\sqrt{s_{yy}}}{\sqrt{s_{ii}}} \cdot a'_i \quad i = 1, \dots, n, \quad a_0 = \bar{y} - a_1 \bar{u}_1 - \dots - a_n \bar{u}_n \quad (2.13)$$

注 2:从上面的简单讨论可以看到,模型结构式(2.3)中的常数项 a_0 并不影响对辨识方法的讨论。因此,在辨识方法的研究中常常忽略常数项 a_0 ,即采用如下方程式

$$y = a_1 u_1 + a_2 u_2 + \dots + a_n u_n + e$$

当然,在对实际过程建模时,模型中是否包含常数项是由过程本身所决定的。

【例 2.1】 假定模型的形式为

$$y = a_1 u_1 + a_2 u_2 + e$$

取得 3 组实验数据为

$$y(k): 1, 1, 3; \quad x_1(k): 1, 2, 1; \quad x_2(k): 0, -1, 2$$

求参数 a_1, a_2 的最小二乘估计。

解 该例中 $\theta = (a_1, a_2)^T, y = (1, 1, 3)^T, \Phi = \begin{pmatrix} 1 & 0 \\ 2 & -1 \\ 1 & 2 \end{pmatrix}$, 所以

$$\Phi^T \Phi = \begin{pmatrix} 6 & 0 \\ 0 & 5 \end{pmatrix}, \quad \Phi^T y = \begin{pmatrix} 6 \\ 5 \end{pmatrix}$$

解正则方程组

$$\Phi^T \Phi \theta = \Phi^T y$$

即

$$\begin{pmatrix} 6 & 0 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \end{pmatrix}$$

得参数的最小二乘估计为: $a_1 = 1, a_2 = 1$ 。

2.2.2 最小二乘估计的性质

若假定残差序列 $\{e(k)\}$ 是零均值、方差为 δ^2 的白噪声序列, 即 $Ee(k) = 0, Ee(i)e(j) = \delta^2 \delta_{ij}, \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$, 这里 $E(\cdot)$ 表示取数学期望。则最小二乘估计具有如下性质。

(1) 参数估计误差是残差 e 的线性函数。

$$\begin{aligned} \tilde{\theta} &= \hat{\theta} - \theta = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T y_N - \theta \\ &= (\Phi_N^T \Phi_N)^{-1} \Phi_N^T (\Phi_N \theta + e_N) - \theta = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T e_N \end{aligned}$$

(2) $\hat{\theta}$ 是 θ 的无偏估计, 即 $E\hat{\theta} = \theta$ 。

由性质(1)及对残差性质的假定有: $E\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Ee_N = 0$ 。

(3) $\hat{\theta}$ 的方差矩阵 $D_{\hat{\theta}} = \sigma^2 (\Phi_N^T \Phi_N)^{-1}$ 。

由性质(1)及对残差性质的假定有

$$D_{\hat{\theta}} = E\tilde{\theta}\tilde{\theta}^T = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Ee_N e_N^T \Phi_N (\Phi_N^T \Phi_N)^{-1} = \sigma^2 (\Phi_N^T \Phi_N)^{-1}$$

(4) $\hat{\theta}$ 的均方误差为

$$E\tilde{\theta}^T \tilde{\theta} = \sigma^2 \left(\frac{1}{\lambda_1} + \cdots + \frac{1}{\lambda_n} \right)$$

其中, $\lambda_i (i=1, 2, \dots, n)$ 是 $\Phi_N^T \Phi_N$ 的 n 个特征根。

证明 由于 $\Phi_N^T \Phi_N$ 是对称正定阵, 故存在正交矩阵 $P, PP^T = I$, 使得 $P^T \Phi_N^T \Phi_N P =$

$$\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}, \text{ 其中 } \lambda_1, \dots, \lambda_n \text{ 是 } \Phi_N^T \Phi_N \text{ 的特征根。由性质(3)知}$$

$$E\tilde{\theta}^T \tilde{\theta} = \sigma^2 \text{tr}(\Phi_N^T \Phi_N)^{-1}$$

其中, $\text{tr} A$ 表示矩阵 A 的迹, 即主对角线元素之和。它具有性质: $\text{tr} AB = \text{tr} BA$, 只要 AB 和 BA 都有意义。由于 $PP^T = I$, 即 $P^{-1} = P^T, P = P^{-T}$, 则有

$$\text{tr}(\Phi_N^T \Phi_N)^{-1} = \text{tr}[(\Phi_N^T \Phi_N)^{-1} PP^T] = \text{tr}[P^T \Phi_N^T \Phi_N P]^{-1} = \left(\frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \cdots + \frac{1}{\lambda_n} \right)$$

本性质得证。

(5) $\hat{\theta}$ 是 θ 的方差最小的线性无偏估计。证明从略,本性质及下面性质的证明请参阅文献(周概容,1984),这里只给出结果。

(6) 噪声方差 σ^2 的一个无偏估计是

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{N-n} J(\hat{\theta}) \\ &= \frac{1}{N-n} \sum_{k=1}^N [y(k) - \hat{a}_1 u_1(k) - \cdots - \hat{a}_n u_n(k)]^2\end{aligned}$$

(7) 输出偏差

$$\tilde{y}_N = y_N - \Phi_N \hat{\theta} = [I - \Phi_N^T (\Phi_N^T \Phi_N)^{-1} \Phi_N^T] e_N$$

(8) $E\tilde{y}_N = 0, E\tilde{y}_N \tilde{y}_N^T = \sigma^2 [I - \Phi_N (\Phi_N^T \Phi_N)^{-1} \Phi_N^T]$ 。

以上给出的性质,对随机序列 $e(k)$ 的概率分布形式未做假设。换句话说,以上得到的结论不依赖分布的具体形式。如果给 $e(k)$ 加上正态分布条件,除了仍具有上述性质外,还具有如下性质:

(9) ① $\hat{\theta}$ 服从 n 维正态分布 $N(\theta, \sigma^2 (\Phi_N^T \Phi_N)^{-1})$;

② $\hat{\theta}$ 与 $e_N = y_N - \Phi_N^T \hat{\theta}$ 独立;

③ $(N-n)\hat{\sigma}^2/\sigma^2$ 服从于自由度为 $(N-n)$ 的 χ^2 分布。

(10) ① $\hat{\theta}_i$ 服从正态分布 $N(\theta_i, \sigma^2 p_{ii}), i=1, \dots, n$ 。其中 $\hat{\theta}_i$ 是参数向量 θ 的第 i 个分量; p_{ii} 为矩阵 $(\Phi_N^T \Phi_N)^{-1}$ 的对角线上第 i 个元素。

② $\hat{\theta}_i$ 与 σ^2 独立。

③ $\theta_i (i=1, 2, \dots, n)$ 的置信区间为: $(\hat{\theta}_i - t_\alpha \hat{\sigma} \sqrt{p_{ii}}, \hat{\theta}_i + t_\alpha \hat{\sigma} \sqrt{p_{ii}})$, 其中 t_α 是自由度为 $(N-n)$ 的 t 分布之 α 水平的双侧分位数。

2.2.3 逐步回归方法

通常在建立稳态模型时,总是在所有有影响的变量中选一些变量作为自变量,并事先选定模型的形式(线性的或非线性的),然后再来确定模型的参数。这样确定的模型一般存在两个问题:一是所选的变量是否合适?也就是说,重要变量是否包含在内、影响小的变量是否排除在外?另一个问题是模型的形式是否合适?

要解决这些问题,首先就要解决什么样的变量是“重要”的变量。当模型中增加一个变量,残差平方和就减少,如果这种减少是显著的,则该变量的影响就是大的(重要的),反之影响就是小(不重要)的。当增加了新的变量后,原来模型中的变量也可能变成不重要的了。因此,为了恰当地选择变量,同时又尽可能地减少计算量,可以考虑以下做法:将变量一个一个地加到模型中去,每加入一个新变量都要检验它是否重要,同时也检验原有的变量是否变成不重要的了,这样一步一步地进行,直到全部的变量都被考察过,就得到了经过筛选的变量和最后的模型。这样的方法就称为逐步回归算法。这是在建立稳态线性模型时常用的一种方法,具体的算法可参看相关文献(卢桂章,1981)。

最近的研究已经将逐步回归思想推广到线性参数的非线性模型情况。线性逐步回归仅是它的一个特例。而且给出了更有效的准则和更简单的计算方法(王秀峰,苏育红,1992;王秀

峰,李波,1993),我们将在 8.6 节中介绍。

2.3 病态方程的求解方法

上一节已经指出,求解模型参数的最小二乘估计,实际上就是直接求解正规方程

$$\Phi_N^T \Phi_N \theta = \Phi_N^T y_N \quad (2.14)$$

如果矩阵 $\Phi_N^T \Phi_N$ 满秩时,最小二乘估计有惟一解。但在很多实际问题中,常常有这种情况:由于所选的输入变量存在较大相关性(这种现象在数理统计学中称为多重共线性),而使得矩阵 Φ_N 的列向量之间存在近似线性关系,或称为非严格的线性关系。这表现为矩阵 $\Phi_N^T \Phi_N$ 虽然满秩,但其行列式的值很小,这时正规方程式(2.14)的解变得对数据特别灵敏,这种情况称为“病态”,相应地正规方程称为病态方程。当然“病态”轻重不同,对解的影响也不同。关于“病态”的轻重如何度量,病态对参数估计会产生什么影响,以及如何改善参数估计等将是本节要讨论的问题。

2.3.1 病态对参数估计的影响

先看一个简单例子。

【例 2.2】考虑方程

$$Ax = b$$

其中, $A = \begin{bmatrix} 98 & 99 \\ 99 & 100 \end{bmatrix}$, $b = \begin{bmatrix} 197 \\ 199 \end{bmatrix}$, 则解 $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 。

若 b 中元素有万分之一的误差,例如取

$$b' = \begin{bmatrix} 197.0197 \\ 199 \end{bmatrix} \text{ 或 } b'' = \begin{bmatrix} 197 \\ 199.0199 \end{bmatrix}$$

则解变为

$$x' = \begin{bmatrix} -0.9700 \\ 2.9503 \end{bmatrix} \text{ 或 } x'' = \begin{bmatrix} 2.9701 \\ -0.9502 \end{bmatrix}$$

这时解与原解的误差高达 200%。显然“病态”使解产生不稳定,一旦数据有误差,解可能产生更大误差。

对一般线性方程

$$Ax = b \quad (2.15)$$

如果 b 有一个扰动 δb ,其解也有一个扰动 δx ,即

$$A(x + \delta x) = b + \delta b$$

$$A\delta x = \delta b$$

因此有 $\|\delta x\| \leq \|A^{-1}\| \|\delta b\|$,这表明了 δb 对解 x 的影响,式中 $\|\cdot\|$ 表示任一种范数。用相对变化可能对这种影响看得更清楚。由方程式(2.15)可得: $\|b\| \leq \|A\| \|x\|$,即 $\|x\| \geq \|b\| / \|A\|$,于是有

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|\delta b\|}{\|b\|}$$

令 $P(A) = \|A\| \|A^{-1}\|$,它反映了方程式(2.15)右端常数项波动对解的影响。类似可以导出(斯图尔特,1980)

$$\frac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{P(\mathbf{A}) \frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|}}{1 - P(\mathbf{A}) \frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|}}$$

$$\frac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{P(\mathbf{A}) \frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|}}{1 - P(\mathbf{A}) \frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|}} \left[\frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\delta \mathbf{b}\|}{\|\mathbf{b}\|} \right]$$

其中, $\Delta \mathbf{A}$ 表示方程组系数矩阵 \mathbf{A} 的扰动。

上面这些式子都说明了数据扰动对解的影响。如果矩阵是病态的, 其行列式值接近于零, 则 $\|\mathbf{A}^{-1}\|$ 会很大, 这时数据扰动会对解产生较大影响。

用 2.2 节给出的最小二乘估计的性质来分析病态对解的影响, 可得到更直观的说明。若 $|\Phi_N^T \Phi_N|$ 很小, 由矩阵逆的定义可知 $(\Phi_N^T \Phi_N)^{-1}$ 将会很大, 由 2.2 节性质(3)知, 参数估计具有很大方差, 这说明解对数据非常灵敏, 极不稳定, 这样得到的解实际上是无法应用的。另外, 由于

$$|\Phi_N^T \Phi_N| = \lambda_1 \lambda_2 \cdots \lambda_n$$

这里 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$ 是 $\Phi_N^T \Phi_N$ 的 n 个特征根。当 $|\Phi_N^T \Phi_N|$ 很小时, 至少有一个特征根接近于零, 由 2.2 节性质(4)则有, 参数估计 $\hat{\theta}$ 的均方误差 $E\tilde{\theta}^T \tilde{\theta} = \sigma^2 \left(\frac{1}{\lambda_1} + \cdots + \frac{1}{\lambda_n} \right)$ 将趋于无穷大, 其解失去实用价值。

从以上分析可见, “病态”对求解参数估计影响是很大的。那么如何判断方程是否为病态? 如何衡量“病态”的轻重? 是“病入膏肓”还是“伤风感冒”的小病? 下面给出度量标准。

2.3.2 条件数

衡量病态(或多重共线)程度的量称为条件数。下面给出常用的两种条件数的定义。

(1) $P(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$ 定义为线性方程组 $\mathbf{Ax} = \mathbf{b}$ 的(病态)条件数。 $P(\mathbf{A})$ 越小, 方程组越良好。可以证明(斯图尔特, 1980): $P(\mathbf{A}^T \mathbf{A}) = P^2(\mathbf{A})$ 。由此可见, 正规方程式(2.14)可使病态加重。

(2) $K(\mathbf{A}) = |\lambda_1| / |\lambda_n|$ (即 \mathbf{A} 的最大特征根/ \mathbf{A} 的最小特征根, $\lambda_i > 0, i = 1, \cdots, n$) 定义为线性方程组 $\mathbf{Ax} = \mathbf{b}$ 的条件数(叶其孝, 1993)。一般认为: 当 $K(\mathbf{A}) \leq 100$ 时, 方程组不存在病态; $100 \leq K(\mathbf{A}) \leq 1000$ 时, 存在较重病态; 当 $K(\mathbf{A}) \geq 1000$ 时, 存在严重病态。

2.3.3 病态方程的求解方法

最小二乘估计是一种无偏估计。但在病态情况下, 估计方差太大, 很不稳定。为了改进估计, 减少均方误差, 下面给出两种估计方法。

1. Householder 变换求解法

为了减轻方程的“病情”, 该方法不是求解正规方程式(2.14), 而是直接解方程组 $\Phi_N \theta = \mathbf{y}_N$ 。它的基本思想: 寻找一个正交变换阵 \mathbf{P} , $\mathbf{P}^T \mathbf{P} = \mathbf{I}$, 使得

$$\mathbf{P} \Phi_N = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{P} \mathbf{y}_N = \begin{pmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{pmatrix} \quad (2.16)$$

其中 \mathbf{R} 为 $n \times n$ 的上三角阵; $\boldsymbol{\eta}_1$ 为 n 维向量, 而最小二乘估计应使

$$\begin{aligned}
J(\boldsymbol{\theta}) &= (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) \\
&= (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T \mathbf{P}^T \mathbf{P} (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) \\
&= (\boldsymbol{\eta}_1 - \mathbf{R} \boldsymbol{\theta})^T (\boldsymbol{\eta}_1 - \mathbf{R} \boldsymbol{\theta}) + \boldsymbol{\eta}_2^T \boldsymbol{\eta}_2
\end{aligned} \tag{2.17}$$

达到最小。由于右端第二项与 $\boldsymbol{\theta}$ 无关,所以只要使

$$\mathbf{R} \boldsymbol{\theta} = \boldsymbol{\eta}_1 \tag{2.18}$$

即求出参数估计 $\hat{\boldsymbol{\theta}}$ 。这样,求解最小二乘估计,即求解正规方程式(2.14)的问题化为解方程组(2.18)的问题了,而方程组(2.18)不会加重方程病态。

现在的问题是如何得到满足式(2.16)的正交变换矩阵,Householder 变换将能完成这项任务。

考虑如下的 Householder 变换

$$\mathbf{H} = \mathbf{I} - 2\mathbf{s}\mathbf{s}^T \tag{2.19}$$

其中, \mathbf{s} 为任意单位长度的向量。

由式(2.19)定义的变换矩阵的性质可归纳为以下几条引理。

引理 2.1 \mathbf{H} 具有下列性质:

- ① \mathbf{H} 是对称的, 即 $\mathbf{H}^T = \mathbf{H}$;
- ② \mathbf{H} 是正交矩阵, 即 $\mathbf{H}^T \mathbf{H} = \mathbf{I}$ 。

证明 ① $\mathbf{H}^T = (\mathbf{I} - 2\mathbf{s}\mathbf{s}^T)^T = \mathbf{I} - 2\mathbf{s}\mathbf{s}^T = \mathbf{H}$

② $\mathbf{H}^T \mathbf{H} = (\mathbf{I} - 2\mathbf{s}\mathbf{s}^T)^T (\mathbf{I} - 2\mathbf{s}\mathbf{s}^T) = \mathbf{I} - 4\mathbf{s}\mathbf{s}^T + 4\mathbf{s}\mathbf{s}^T \mathbf{s}\mathbf{s}^T = \mathbf{I}$ (因为 $\mathbf{s}^T \mathbf{s} = 1$)

引理 2.2 Householder 变换矩阵的积是正交的, 即若

$$\mathbf{P} = \mathbf{H}_n \mathbf{H}_{n-1} \cdots \mathbf{H}_1$$

其中, $\mathbf{H}_n, \mathbf{H}_{n-1}, \cdots, \mathbf{H}_1$ 都是 Householder 变换矩阵, 则 $\mathbf{P}^T \mathbf{P} = \mathbf{I}$ 。

证明 $\mathbf{P}^T \mathbf{P} = (\mathbf{H}_n \mathbf{H}_{n-1} \cdots \mathbf{H}_1)^T (\mathbf{H}_n \mathbf{H}_{n-1} \cdots \mathbf{H}_1)$
 $= \mathbf{H}_1^T \mathbf{H}_2^T \cdots \mathbf{H}_n^T \mathbf{H}_n \mathbf{H}_{n-1} \cdots \mathbf{H}_1$
 $= \mathbf{I}$

引理 2.3 任一已知向量 \mathbf{x} , 存在一个 Householder 变换 \mathbf{H} , 使得

$$\mathbf{H}\mathbf{x} = \begin{pmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{2.20}$$

其中, $\lambda = (\mathbf{x}^T \mathbf{x})^{\frac{1}{2}}$ 。

证明 如果式(2.20)成立, 则由 \mathbf{H} 的对称、正交性有

$$\mathbf{x} = \mathbf{H} \begin{pmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

而 $\mathbf{H} = \mathbf{I} - 2\mathbf{s}\mathbf{s}^T$, 故得

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = (\mathbf{I} - 2\mathbf{s}\mathbf{s}^T)^T \begin{pmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} (1 - 2s_1^2)\lambda \\ -2s_1 s_2 \lambda \\ \vdots \\ -2s_1 s_n \lambda \end{pmatrix}$$

式中, s_i 是向量 s 的第 i 个分量。取

$$s_1 = \sqrt{\frac{1}{2} \left(1 - \frac{x_1}{\lambda} \right)}$$

$$s_i = -x_i / (2s_1\lambda), \quad i = 2, \dots, n$$

容易验证, 这样取得的向量 $s = (s_1, s_2, \dots, s_n)^\top$ 是单位向量, 所构造的 H 矩阵满足要求。即有

$$Hx = \begin{pmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{pmatrix}, x^\top H^\top Hx = x^\top x = \lambda^2, \text{ 即 } \lambda = \sqrt{x^\top x}$$

引理 2.4 对任意 $N \times n$ 矩阵 $\Phi (N \geq n)$, 存在正交矩阵 P , 使得

$$P\Phi = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

其中, R 为 $n \times n$ 的上三角矩阵, P 是 n 个 Householder 变换之积。

证明 对给定矩阵 Φ , 根据引理 2.3, 对 Φ 的第一列可以找到一个 Householder 变换 $H_1 = I - 2s_1 s_1^\top$, 使得

$$H_1 \Phi = \begin{pmatrix} \lambda_1 & \varphi'_{12} & \cdots & \varphi'_{1n} \\ 0 & & & \\ \vdots & \vdots & & \vdots \\ 0 & \varphi'_{N2} & \cdots & \varphi'_{Nn} \end{pmatrix}$$

类似地, 对第二列构造 $H_2 = I - 2s_2 s_2^\top$, 其中 s_2 的第一个分量为零, 其第二个至第 N 个分量用 Φ 的第二列中第二至第 N 个元素构成向量 x , 并按引理 2.3 给出的方法计算得到。这样使得

$$H_2 H_1 \Phi = \begin{pmatrix} \lambda_1 & \varphi''_{12} & \varphi''_{13} & \cdots & \varphi''_{1n} \\ 0 & \lambda_2 & \varphi''_{23} & \cdots & \varphi''_{2n} \\ 0 & 0 & & & \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \varphi''_{N3} & \cdots & \varphi''_{Nn} \end{pmatrix}$$

如此继续下去, 即可得到 $P = H_n H_{n-1} \cdots H_1$ 满足要求。

由上面的这些引理就可得到最后所要的结果。

定理 2.1 求解误差平方和为

$$J(\theta) = (y_N - \Phi_N \theta)^\top (y_N - \Phi_N \theta)$$

的最小二乘估计问题, 等价于求解方程组

$$R\theta = \eta_1 \quad (2.21)$$

其中 R 为上三角矩阵。且

$$P\Phi_N = \begin{pmatrix} R \\ 0 \end{pmatrix} \quad (2.22)$$

$$Py_N = \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \quad (2.23)$$

其中 $P = H_n H_{n-1} \cdots H_1$, $H_i (i = 1, 2, \dots, n)$ 是 Householder 变换。则误差平方和为

$$J(\theta) = \eta_2^\top \eta_2 \quad (2.24)$$

由引理 2.4 及本节开始的分析易得出此结论,证明从略。

利用上面的定理,可将求最小二乘估计的过程变为先用 Householder 变换将 Φ_N 变成上三角阵,然后再解式(2.21),从而得到最小二乘估计 $\hat{\theta}$ 。由于 R 是上三角阵,所以只需用简单的回代法即可得到解。关于求 $P\Phi_N$ 和 py_N 的算法框图如图 2.1 所示。

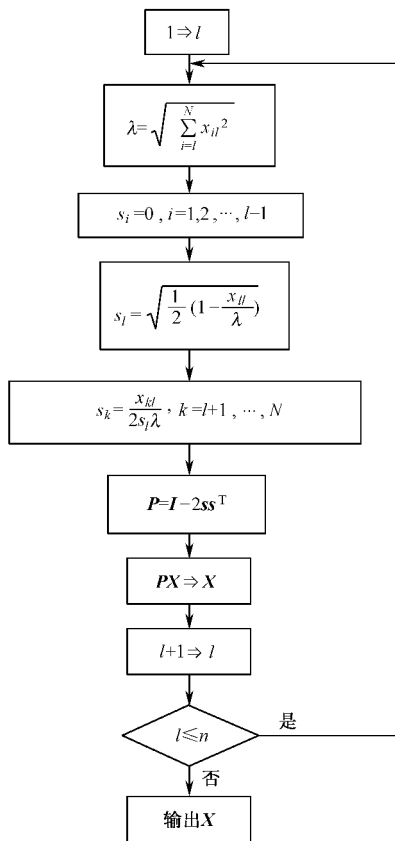


图 2.1 Householder 变换计算框图

注:图 2.1 中矩阵 X

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} & x_{1n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} & x_{Nn+1} \end{pmatrix} = (\Phi : y_N) \quad (2.25)$$

【例 2.3】 考虑下列数据: $N=5, n=3$

$$\Phi_N = \begin{bmatrix} 2 & 1 & -3 \\ 1 & -3 & 4 \\ 5 & -1 & -2 \\ 0 & 1 & 6 \\ 6 & -5 & 0 \end{bmatrix}, \quad y_N = \begin{bmatrix} 4 \\ -4 \\ 2 \\ 2 \\ -4 \end{bmatrix}$$

用 Householder 变换计算结果如下(每一步所得 X):

第一步

8.124039	-4.431293	-1.477098	-1.230915
0.000000	-2.113119	3.751324	-3.145839
-0.000000	3.434405	-3.243381	6.270805
0.000000	1.000000	6.000000	2.000000
0.000000	0.321286	-1.492056	1.124966

第二步

8.124039	-4.431293	-1.477098	-1.230915
0.000000	4.166969	-3.250672	7.330376
-0.000000	-0.000000	0.585816	0.541655
0.000000	-0.000000	7.114952	0.331836
0.000000	0.000000	-1.133838	0.589008

第三步

8.124039	-4.431293	-1.477098	-1.230915
-0.000000	4.166969	-3.250672	7.330376
-0.000000	-0.000000	7.228507	0.278131
0.000000	-0.000000	0.000000	0.614096
0.000000	0.000000	-0.000000	0.544027

从而解得: $a_1=0.831, a_2=1.789, a_3=0.038$ 。

2. 主要成分估计方法(叶其孝,1993;孙文爽,1994)

由于 $\Phi_N^T \Phi_N$ 是对称正定阵,总是存在一个正交矩阵 $P, PP^T=I$,使

$$P^T \Phi_N^T \Phi_N P = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

其中 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 为 $\Phi_N^T \Phi_N$ 的 n 个特征根。这时正规方程(2.14)可化为

$$P^T \Phi_N^T \Phi_N P P^T \theta = P^T \Phi_N^T y_N \quad (2.26)$$

记 $\beta = P^T \theta = (\beta_1, \beta_2, \dots, \beta_n)^T$, 方程式(2.26)变为

$$\begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_r & \\ & & & & \lambda_{r+1} & \\ & & & & & \ddots \\ & & & & & & \lambda_n \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_r \\ \beta_{r+1} \\ \vdots \\ \beta_n \end{pmatrix} = P^T \Phi_N^T y_N \triangleq \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \\ b_{r+1} \\ \vdots \\ b_n \end{pmatrix} \quad (2.27)$$

若 $\Phi_N^T \Phi_N$ 不是病态,则由式(2.27)可解得 $\beta_i = b_i / \lambda_i (i=1, 2, \dots, n)$, 而原正规方程参数 $\theta = P\beta$ (因为 $PP^T=I$)。

当 $\Phi_N^T \Phi_N$ 为病态时,某些特征根近似为零,不妨设

$$\lambda_{r+1} = \cdots = \lambda_n = 0$$

这时可令

$$\begin{aligned}\beta_{r+1} &= \cdots = \beta_n = 0 \\ \beta_i &= b_i / \lambda_i, i = 1, 2, \cdots, r\end{aligned}$$

将正交矩阵 P 对应于 r 也相应地分成两块: $P = (P_1 : P_2)$, 则

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_r \end{pmatrix} = P_1 \begin{pmatrix} b_1 / \lambda_1 \\ b_2 / \lambda_2 \\ \vdots \\ b_r / \lambda_r \end{pmatrix}, \theta_{r+1} = \theta_{r+2} = \cdots = \theta_n = 0$$

这样得到的估计称做主成分估计。

正交变换实质上是将 Φ_N 中的列正交化。特征根的大小,反映数据所提供的信息量的多少,主成分估计保留信息量大的变量,去掉信息量小的变量,克服由于数据的相关性带来的求解困难,与最小二乘估计相比,具有更好的性质。

剩下的问题是在做主成分估计时, r 究竟取多少? 在主成分估计中,把比值

$$(\lambda_1 + \cdots + \lambda_r) / (\lambda_1 + \cdots + \lambda_n)$$

称做前 r 个特征根的累计贡献率。一般经验中选择 r , 使其累计贡献率达到 80% 以上。

经验表明,用 Householder 变换求最小二乘解一般比解正规方程精度要高。特别在病态较重时,这一优点尤为突出。但当病态严重时,解正规方程的方法无法求解,Householder 方法可得到众多解中的一个解,但往往不是最小模解,即可能存在其他解,而模比它还小。这里“模”是指“解向量的模”,一般是指解向量的长度。主成分估计法可得到惟一的最小模解。应当指出,病态方程求解问题有许多专门讨论的文章和书籍,提出了不少其他有效的方法,如奇异值分解(斯图尔特,1980)和岭估计法(叶其孝,1993)就是常用的两种。

2.4 模型参数的最大似然估计(ML)

2.4.1 最大似然准则

由于系统受随机噪声干扰,所以系统的输出 y 是随机的。对于给定的参数向量 θ , 输出 y 的分布密度记做 $P(y; \theta)$ 。若 $y(1), \cdots, y(N)$ 是独立的观测,则它们的联合分布密度是

$$L(y(1), \cdots, y(N); \theta) = \prod_{k=1}^N P(y(k); \theta) \quad (2.28)$$

或

$$L(y_N; \theta) = \prod_{k=1}^N P(y(k); \theta)$$

L 称为似然函数(Likelihood Function)。根据数理统计中最大似然原则(Wilks, 1962)可知,使得 L 达到最大的参数估计 $\hat{\theta}$ 称为最大似然估计(Maximum Likelihood Estimator),即 $\hat{\theta} = \hat{\theta}(y_N)$, 使

$$L(y_N; \hat{\theta}) = \max_{\theta} L(y_N; \theta)$$

仍然考虑模型结构式(2.3),假定噪声 $e(k)$ 服从正态分布 $N(0, \sigma^2)$, 于是

$$p(y(k); \theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2} [y(k) - a_1 u_1(k) - \cdots - a_n u_n(k)]^2\right\}$$

所以似然函数是

$$L(y_N; \theta) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \exp\left\{-\frac{1}{2\sigma^2} (y_N - \Phi_N \theta)^T (y_N - \Phi_N \theta)\right\}$$

如果 $e(k)$ 不一定是白噪声序列, 假定 y_N 的联合分布仍然是正态分布, 协方差阵是 Σ , 则似然函数是

$$L(y_N; \theta) = (2\pi |\Sigma|)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2} (y_N - \Phi_N \theta)^T \Sigma^{-1} (y_N - \Phi_N \theta)\right\} \quad (2.29)$$

现在的问题是求 $\hat{\theta}$, 使 $L(y_N; \hat{\theta})$ 达到最大。

2.4.2 最大似然估计

(1) 考虑一般情形的似然函数式(2.29), 则参数 θ 的最大似然估计就是加权的最小二乘估计, 即

$$\hat{\theta} = (\Phi_N^T \Sigma^{-1} \Phi_N)^{-1} \Phi_N^T \Sigma^{-1} y_N \quad (2.30)$$

这个结论是明显的, 因为使式(2.29)达到最大等价于使

$$J(\theta) = (y_N - \Phi_N \theta)^T \Sigma^{-1} (y_N - \Phi_N \theta)$$

达到最小。

(2) 考虑特殊情况 $\Sigma = \sigma^2 I$, 而且 σ^2 未知, 则最大似然估计是

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T y_N \quad (2.31)$$

$$\hat{\sigma}^2 = \frac{1}{N} (y_N - \Phi_N \theta)^T (y_N - \Phi_N \theta) \quad (2.32)$$

这一结论通过直接求似然函数的极值点可得。值得注意的是, 式(2.32)与 2.2 节由性质(6)给出的 $\hat{\sigma}^2$ 估计是有别的, 所以式(2.32)给出的 $\hat{\sigma}^2$ 估计是有偏的。但是 N 充分大时, 两者的差别也就很微小了。从式(2.31)不难看出, 对于噪声为零均值、正态白噪声的情况, 参数 θ 的最大似然估计和最小二乘估计是一样的。

(3) y_N 的联合分布是 $N(\Phi_N \theta, \Sigma)$, 则 θ, Σ 的最大似然估计是

$$\hat{\theta} = (\Phi_N^T \hat{\Sigma}^{-1} \Phi_N)^{-1} \Phi_N^T \hat{\Sigma}^{-1} y_N \quad (2.33)$$

$$\hat{\Sigma} = \frac{1}{N} (y_N - \Phi_N \theta)(y_N - \Phi_N \theta)^T \quad (2.34)$$

从似然函数

$$L(y_N; \theta, \Sigma) = (2\pi |\Sigma|)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2} (y_N - \Phi_N \theta)^T \Sigma^{-1} (y_N - \Phi_N \theta)\right\}$$

出发, 分别对参数 θ 和 Σ 求导数并令其为零, 即可得上面结果。但由于似然函数是密度函数(从而 >0), 且对数函数 \log 是单调函数, 所以 $\log L$ 和 L 有相同的极大值点, 又因为极大化 $\log L$ 比极大化 L 简单, 所以常用对数似然函数 $\log L$ 代替似然函数 L 求解。对 $\log L$ 分别求导, 并令其为零得

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log L(\mathbf{y}_N; \boldsymbol{\theta}, \boldsymbol{\Sigma}) = \boldsymbol{\Phi}_N^T \boldsymbol{\Sigma}^{-1} (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) = 0$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \log L(\mathbf{y}_N; \boldsymbol{\theta}, \boldsymbol{\Sigma}) = -\frac{N}{2} \boldsymbol{\Sigma}^{-1} + \frac{1}{2} \boldsymbol{\Sigma}^{-1} (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T \boldsymbol{\Sigma}^{-1} = 0$$

于是得到式(2.33)和式(2.34)。

注意在上面的计算中,用到下面的矩阵微分的结果

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \log |\boldsymbol{\Sigma}| = (\boldsymbol{\Sigma}^{-1})^T$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \text{tr} \mathbf{A} \boldsymbol{\Sigma}^{-1} \mathbf{B} = -(\boldsymbol{\Sigma}^{-1} \mathbf{B} \mathbf{A} \boldsymbol{\Sigma}^{-1})^T$$

这两个结果的证明见附录 B 的(5)和(6)。

另一方面,式(2.33)和式(2.34)互相之间是有交叉的,因此,如何能计算出正确的结果是必然考虑的问题。

2.4.3 松弛算法

由式(2.33)和式(2.34)给出的最大似然估计没有一个简单的解法,下面给出一个松弛算法,其步骤如下:

- ① 给出 $\boldsymbol{\Sigma}$ 以某个初始猜测值,例如取 $\boldsymbol{\Sigma} = \mathbf{I}$;
- ② 用给定的 $\boldsymbol{\Sigma}$ 从式(2.33)得到 $\boldsymbol{\theta}$ 的估计值 $\hat{\boldsymbol{\theta}}$;
- ③ 用求得的 $\hat{\boldsymbol{\theta}}$,从式(2.34)得到 $\boldsymbol{\Sigma}$ 的估计值 $\hat{\boldsymbol{\Sigma}}$;
- ④ 如果收敛条件满足则停机,否则从②开始重复。

这种松弛算法的总体收敛性并没有得到证明,但是在实际计算中得到了很好的效果。

总之,最大似然估计对高斯噪声的情形给出同时估计 $\boldsymbol{\theta}$ 和 $\boldsymbol{\Sigma}$ 的算法,由数理统计的理论知道最大似然估计有着很好的渐近性质。

习 题

1. 何为静态模型?试举出实际应用中的 3 个例子。
2. 证明式(2.13)的正确性。
3. 假定输入、输出数据满足如下方程组

$$\begin{cases} y_1 = \theta_1 + e_1 \\ y_2 = 2\theta_1 - \theta_2 + e_2 \\ y_3 = \theta_1 - 2\theta_2 + e_3 \end{cases}$$

其中 $Ee_i = 0, Ee_i^2 = \sigma^2, Ee_i e_j = 0, i \neq j; i = 1, 2, 3$ 。

- (1) 求 θ_1, θ_2 最小二乘估计。
- (2) 求估计量 $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2)^T$ 的协方差阵。
- (3) 求 σ^2 的无偏估计。
4. 证明引理 2.3 中所给出的向量 \mathbf{s} 是单位长度的向量。

5. 分别用 Householder 变换和高斯消去法求解 2.3 节的例 2.2(编程计算)。

6. 用向量形式直接推导性能指标为

$$J(\boldsymbol{\theta}) = (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T \boldsymbol{\Sigma} (\mathbf{y}_N - \boldsymbol{\Phi}_N^T \boldsymbol{\theta})$$

($\boldsymbol{\Sigma}$ 为已知正定加权矩阵)的加权最小二乘估计为

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}_N^T \boldsymbol{\Sigma} \boldsymbol{\Phi}_N)^{-1} \boldsymbol{\Phi}_N^T \boldsymbol{\Sigma} \mathbf{y}_N$$

7. 证明式(2.33)和式(2.34)。

第3章 离散线性动态模型的最小二乘估计

3.1 问题的提法及一次完成最小二乘估计

考虑单输入、单输出系统,如图 3.1 所示。

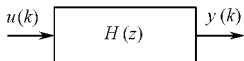


图 3.1 单输入、单输出系统

经 z 变换后,其输入/输出关系为

$$\frac{Y(z)}{U(z)} = H(z) = \frac{b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}} \quad (3.1)$$

其中, $z = e^{sT}$, T 为采样周期, $m \leq n$ 。为了书写方便,不失一般性,在下面的讨论中,取 $m = n$,若 $m < n$,只要取 $b_{m+1} = b_{m+2} = \cdots = b_n = 0$ 即可。这样,式(3.1)也可以写成如下差分方程的形式

$$y(k) + \sum_{i=1}^n a_i y(k-i) = \sum_{i=1}^n b_i u(k-i) \quad (3.2)$$

其中

$$y(k) \triangleq y(kT), \quad u(k) \triangleq u(kT), \quad k = 1, 2, \cdots$$

下面的问题是根据输入、输出数据来确定未知参数 $a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n$ 。

但是,在实际的测量中,量测到的数据总是有误差的。这包含了测量噪声、模型误差等。因此,实际测得的输入、输出数据之间的关系应修正为

$$y(k) + \sum_{i=1}^n a_i y(k-i) = \sum_{i=1}^n b_i u(k-i) + e(k) \quad (3.3)$$

其中, $e(k)$ 称为模型误差或残差。

若令

$$\boldsymbol{\varphi}_k^T = [-y(n+k-1), \cdots, -y(k), u(n+k-1), \cdots, u(k)] \quad (3.4)$$

$$\boldsymbol{\theta}^T = [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n] \quad (3.5)$$

则式(3.3)可以写成向量形式

$$y(n+k) = \boldsymbol{\varphi}_k^T \boldsymbol{\theta} + e(n+k) \quad (3.6)$$

这时,对于给定的阶次 n ,基于模型式(3.2)的最小二乘估计问题可以表述如下:通过对系统量测得到输入、输出量测数据 $\{u(k), y(k); k=1, 2, \cdots, n+N\}$,利用这些数据,按照最小残差平方和准则来决定参数向量 $\boldsymbol{\theta}$ 的估计。换句话说,确定参数向量 $\boldsymbol{\theta}$,使得残差平方和

$$J = \sum_{k=n+1}^{n+N} [y(k) - \boldsymbol{\varphi}_{k-n}^T \boldsymbol{\theta}]^2 \quad (3.7)$$

达到极小。指标 J 可以写成向量形式

$$J = \mathbf{e}_N^T \mathbf{e}_N \quad (3.8)$$

其中

$$e_N = y_N - \Phi_N \theta = \begin{bmatrix} e(n+1) \\ e(n+2) \\ \vdots \\ e(n+N) \end{bmatrix} \quad (3.9)$$

$$\Phi_N = \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_N^T \end{bmatrix} = \begin{bmatrix} -y(n) & \cdots & -y(1) & u(n) & \cdots & u(1) \\ -y(n+1) & \cdots & -y(2) & u(n+1) & \cdots & u(2) \\ \cdots & & \cdots & \cdots & \cdots & \cdots \\ -y(n+N-1) & \cdots & -y(N) & u(n+N-1) & \cdots & u(N) \end{bmatrix}$$

$$y_N = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(n+N) \end{bmatrix} \quad (3.10)$$

将式(3.9)代入式(3.8),然后关于 θ 求偏导数并令其为零得

$$(\Phi_N^T \Phi_N) \theta = \Phi_N^T y_N \quad (3.11)$$

称式(3.11)为正则方程。

如果 $\Phi_N^T \Phi_N$ 可逆,则存在惟一解

$$\hat{\theta}_N = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T y_N \quad (3.12)$$

称 $\hat{\theta}_N$ 为 θ 的最小二乘估计。

性能指标式(3.8)意味着对所有数据的地位都是平等的。但是在一些情况下,某些数据比较重要,应当特别重视等,这就需要对它们区别对待,在性能指标中体现为对残差的不同加权值,加权越大表示对相应的数据越重视。这时,性能指标变为残差的加权平方和

$$J = e_N^T W_N e_N \quad (3.13)$$

其中, W_N 为正定矩阵。

使性能指标式(3.13)达到极小的估计称为加权最小二乘估计。类似地推导,可得加权最小二乘估计为

$$\hat{\theta}_N = (\Phi_N^T W_N \Phi_N)^{-1} \Phi_N^T W_N y_N \quad (3.14)$$

最简单的情况是 W_N 为对角阵,即

$$W_N = \text{diag}(w_1 w_2 \cdots w_N), w_i > 0, i = 1, \cdots, N \quad (3.15)$$

从上面的推导可以看出,只要取得充分多的输入、输出数据,解正则方程式(3.11)(或式(3.14))就可以得到参数向量 θ 的最小二乘(或加权最小二乘)估计。

3.2 最小二乘估计的递推算法(RLS)

上一节给出的最小二乘估计式(3.12)和加权最小二乘估计式(3.14)是基于全部数据一次处理得到的,因此称此算法是一次完成最小二乘法。对于线性定常系统来说,数据越多,估计量的精确度就越高。但是,若采用一次完成的算法就需要存储大量数据,这不仅要占用计算机的大量内存,而且随着数据的增加,计算时间也越来越长。另外,一次完成算法也不能满足在线辨识的要求。为了解决这些矛盾,下面给出最小二乘估计的递推算法。

3.2.1 递推最小二乘法

递推算法的基本思想是：每取得一个新数据，就根据新数据对原估计量进行修正，从而得到（改善的）新估计量，而不是将新数据加到老数据中重新计算。为得到递推算法，先来看一下增加新数据后最小二乘估计有何变化。

假定根据 $(n+N)$ 次采样数据得到的最小二乘估计为 $\hat{\theta}_N$ ，第 $(n+N+1)$ 次采样后，得到的估计为 $\hat{\theta}_{N+1}$ ，那么，由式(3.14)知

$$\hat{\theta}_{N+1} = (\Phi_{N+1}^T W_{N+1} \Phi_{N+1})^{-1} \Phi_{N+1}^T W_{N+1} y_{N+1} \quad (3.16)$$

其中

$$\Phi_{N+1} = \begin{bmatrix} \Phi_N \\ \varphi_{N+1}^T \end{bmatrix} \quad (3.17)$$

$$y_{N+1} = \begin{bmatrix} y_N \\ y(n+N+1) \end{bmatrix} \quad (3.18)$$

$$W_{N+1} = \begin{bmatrix} W_N & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & w_{N+1}^{-1} \end{bmatrix} \quad (3.19)$$

这里 $\Phi_N, y_N, \varphi_{N+1}, W_N$ 的定义与3.1节中相同。

下面推导 $\hat{\theta}_N$ 和 $\hat{\theta}_{N+1}$ 之间的关系。令

$$P_k = (\Phi_k^T W_k \Phi_k)^{-1}$$

则

$$\begin{aligned} P_{N+1} &= [\Phi_{N+1}^T W_{N+1} \Phi_{N+1}]^{-1} = [\Phi_N^T W_N \Phi_N + \varphi_{N+1} w_{N+1} \varphi_{N+1}^T]^{-1} \\ &= [P_N^{-1} + \varphi_{N+1} w_{N+1} \varphi_{N+1}^T]^{-1} \end{aligned} \quad (3.20)$$

$$\begin{aligned} \hat{\theta}_{N+1} &= P_{N+1} \Phi_{N+1}^T W_{N+1} y_{N+1} \\ &= P_{N+1} [\Phi_N^T W_N y_N + \varphi_{N+1} w_{N+1} y(n+N+1)] \\ &= P_{N+1} [P_N^{-1} \hat{\theta}_N + \varphi_{N+1} w_{N+1} y(n+N+1)] \end{aligned} \quad (3.21)$$

其中最后一个等号用到式(3.14)及 P_k 的定义。

由式(3.20)解得 P_N^{-1} 并代入式(3.21)得

$$\hat{\theta}_{N+1} = \hat{\theta}_N + P_{N+1} \varphi_{N+1} w_{N+1} [y(n+N+1) - \varphi_{N+1}^T \hat{\theta}_N] \quad (3.22)$$

显然，利用式(3.22)即可递推估计参数。然而，不难发现，若采用式(3.22)，则在每一次递推估计时都需要进行矩阵求逆运算（为求得 P_{N+1} ），这当然是所不希望的。借助于矩阵求逆公式

$$(A^{-1} + H R^{-1} H^T)^{-1} = A - A H (H^T A H + R)^{-1} H^T A$$

（证明见附录A）可以弥补这一缺陷。利用上述的矩阵求逆公式，式(3.20)可简化为

$$P_{N+1} = P_N - P_N \varphi_{N+1} (\varphi_{N+1}^T P_N \varphi_{N+1} + w_{N+1}^{-1})^{-1} \varphi_{N+1}^T P_N$$

或

$$P_{N+1} = P_N - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N}{\varphi_{N+1}^T P_N \varphi_{N+1} + w_{N+1}^{-1}} \quad (3.23)$$

将式(3.23)代入式(3.22)，并注意到 $\varphi_{N+1}^T P_N \varphi_{N+1}$ 是标量，则式(3.22)可以简化为

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{P_N \varphi_{N+1}}{\varphi_{N+1}^T P_N \varphi_{N+1} + w_{N+1}^{-1}} [y(n+N+1) - \varphi_{N+1}^T \hat{\theta}_N] \quad (3.24)$$

式(3.23)和式(3.24)构成了加权最小二乘估计的递推算法。特别地,如果把每次的误差都平等对待,即取加权矩阵 W 为单位阵,则式(3.24)和式(3.23)变为

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{P_N \varphi_{N+1}}{\varphi_{N+1}^T P_N \varphi_{N+1} + 1} [y(n+N+1) - \varphi_{N+1}^T \hat{\theta}_N] \quad (3.25)$$

$$P_{N+1} = P_N - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N}{\varphi_{N+1}^T P_N \varphi_{N+1} + 1} \quad (3.26)$$

式(3.25)和式(3.26)构成了通常最小二乘估计的递推公式。

递推公式式(3.25)有着明显的直观意义。 $\varphi_{N+1}^T \hat{\theta}_N$ 是在 $(n+N)$ 次量测基础上对第 $(n+N+1)$ 次输出 $y(n+N+1)$ 的预测,当实际测得 $y(n+N+1)$ 后,它与预测估计之间有一偏差

$$y(n+N+1) - \varphi_{N+1}^T \hat{\theta}_N$$

称为预测偏差。这个偏差一方面说明实际过程与原模型之间存在误差,另一方面也体现了新的量测对参数带来的新信息。因此,新的参数估计值应根据这一偏差对原估计值 $\hat{\theta}$ 进行修正。

修正的幅度,即增益向量 $K = \frac{P_N \varphi_{N+1}}{\varphi_{N+1}^T P_N \varphi_{N+1} + 1}$ 是根据使误差平方和最小的原则来确定的。

3.2.2 初始值的选择

初始值的选择在实际中常采用如下两种方法。

(1) 用最初的 $3n$ 组数据首先计算 P_{2n} 和 θ_{2n} , 即

$$\begin{aligned} P_{2n} &= (\Phi_{2n}^T \Phi_{2n})^{-1} \\ \hat{\theta}_{2n} &= P_{2n} \Phi_{2n}^T y_{2n} \end{aligned} \quad (3.27)$$

以此作为初值,从 $(3n+1)$ 次量测开始递推。

(2) 取 $\hat{\theta}_0 = 0, P_0 = \alpha^2 I$, 其中 α 为充分大的实数, I 为单位阵。

第一种取法的优点是初值比较精确,开始递推就可得到较好的估计值。其缺点是需要求解方程组。

第二种取法的优点是简单,不需要先解方程组,便于应用。而且可以证明,从此初值出发,经过 $2n$ (待估参数的个数) 步递推后得到的 P_{2n} 和 θ_{2n} , 当 $\alpha \rightarrow \infty$ 时,趋近于由式(3.27)计算出的值。其缺点是在递推的最初几步,参数估计值误差较大。因此,将此方法应用于在线辨识时,最初的若干步辨识的结果是不能应用的。所以,在应用于自校正控制系统时,往往需要首先递推充分长的步数后,才能将辨识的参数直接用于控制律的计算,否则会影响控制品质。

下面对第二种初始值取法的正确性给出简单说明。

按照第二种取法得

$$\theta_0^* = 0, \quad P_0^* = \begin{bmatrix} \alpha^2 & & & \\ & \alpha^2 & & \\ & & \ddots & \\ & & & \alpha^2 \end{bmatrix}_{2n}$$

当取得 $(n+1)$ 次观测后,按照式(3.20)和式(3.21) (取 $W \equiv I$) 得

$$\begin{aligned} P_1^* &= [P_0^{*-1} + \varphi_1 \varphi_1^T]^{-1} \\ \theta_1^* &= P_1^* \varphi_1 y(n+1) \end{aligned}$$

再取第 $(n+2)$ 次观测,再按照式(3.20)和式(3.21)得

$$\mathbf{P}_2^* = [\mathbf{P}_1^{*-1} + \boldsymbol{\varphi}_2 \boldsymbol{\varphi}_2^T]^{-1} = [(\mathbf{P}_0^{*-1} + \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T) + \boldsymbol{\varphi}_2 \boldsymbol{\varphi}_2^T]^{-1}$$

$$\boldsymbol{\theta}_2^* = \mathbf{P}_2^* [\boldsymbol{\Phi}_1^T \mathbf{y}_1 + \boldsymbol{\varphi}_2 y(n+2)] = \mathbf{P}_2^* [\boldsymbol{\Phi}_2^T \mathbf{y}_2]$$

这样一直取到第 $2n$ 次观测,有

$$\mathbf{P}_{2n}^* = [\mathbf{P}_0^{*-1} + \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T + \boldsymbol{\varphi}_2 \boldsymbol{\varphi}_2^T + \cdots + \boldsymbol{\varphi}_{2n} \boldsymbol{\varphi}_{2n}^T]^{-1} = [\mathbf{P}_0^{*-1} + \boldsymbol{\Phi}_{2n}^T \boldsymbol{\Phi}_{2n}]^{-1}$$

$$\boldsymbol{\theta}_{2n}^* = \mathbf{P}_{2n}^* [\boldsymbol{\Phi}_{2n}^T \mathbf{y}_{2n}]$$

这样,当 $\alpha \rightarrow \infty$ 时, $\mathbf{P}_0^{*-1} \rightarrow 0$,故此有

$$\mathbf{P}_{2n}^* \rightarrow \mathbf{P}_{2n} = (\boldsymbol{\Phi}_{2n}^T \boldsymbol{\Phi}_{2n})^{-1}$$

$$\boldsymbol{\theta}_{2n}^* \rightarrow \mathbf{P}_{2n} \boldsymbol{\Phi}_{2n}^T \mathbf{y}_{2n} = \hat{\boldsymbol{\theta}}_{2n}$$

3.2.3 计算步骤及举例

由前面的讨论,如图 3.2 所示给出了基于式(3.25)和式(3.26)的递推算法框图。下面以二阶线性模型为例说明计算过程。

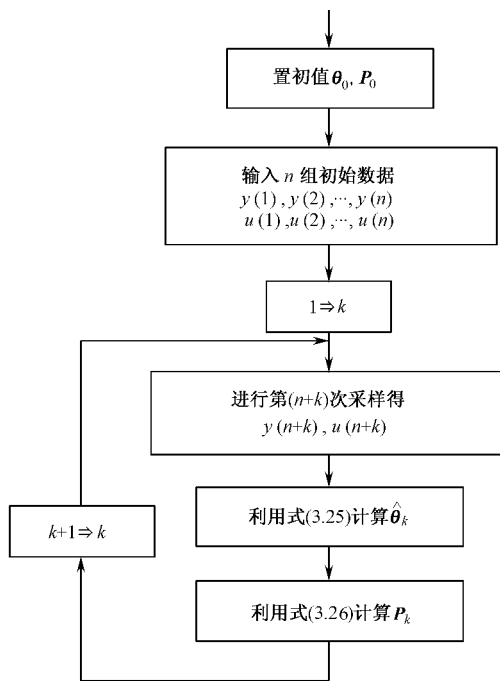


图 3.2 递推最小二乘法框图

【例 3.1】 考虑二阶线性模型

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) + e(k)$$

利用递推公式(3.25)和式(3.26)求参数估计,其中 $n=2$ 。

解 记

$$\begin{cases} \boldsymbol{\varphi}_k^T = [y(k+1), y(k), u(k+1), u(k)] \triangleq [\boldsymbol{\varphi}_k(1), \boldsymbol{\varphi}_k(2), \boldsymbol{\varphi}_k(3), \boldsymbol{\varphi}_k(4)] \\ \boldsymbol{\theta}^T = [a_1, a_2, b_1, b_2] \triangleq [\boldsymbol{\theta}(1), \boldsymbol{\theta}(2), \boldsymbol{\theta}(3), \boldsymbol{\theta}(4)] \\ \mathbf{P}_K = [\mathbf{P}_k(i, j)]_{4 \times 4} \end{cases}$$

若令

$$\begin{cases} \boldsymbol{\psi}(i) = \sum_{j=1}^4 \mathbf{P}_N(i, j) \boldsymbol{\varphi}_{N+1}(j), i = 1, \dots, 4 \\ q = 1 + \boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} = 1 + \sum_{j=1}^4 \boldsymbol{\varphi}_{N+1}(j) \boldsymbol{\psi}(j) \\ z = \boldsymbol{\varphi}_{N+1}^T \hat{\boldsymbol{\theta}}_N = \sum_{j=1}^4 \boldsymbol{\varphi}_{N+1}(j) \hat{\boldsymbol{\theta}}(j) \end{cases}$$

则有

$$\begin{cases} \hat{\boldsymbol{\theta}}_{N+1}(i) = \hat{\boldsymbol{\theta}}_N(i) + \frac{\boldsymbol{\psi}(i)}{q} [y(n+N+1) - z] \\ \mathbf{P}_{N+1}(i, j) = \mathbf{P}_N(i, j) - \frac{\boldsymbol{\psi}(i) \boldsymbol{\psi}(j)}{q} \end{cases}$$

初值取为

$$\hat{\boldsymbol{\theta}}_0 = 0, \mathbf{P}_0 = \begin{pmatrix} \alpha^2 & 0 & \cdots & 0 \\ 0 & \alpha^2 & \ddots & \vdots \\ \vdots & \ddots & \alpha^2 & 0 \\ 0 & \cdots & 0 & \alpha^2 \end{pmatrix}$$

α^2 是充分大的正数(一般取 $\alpha^2 > 10^3$), 从第 3 次观测开始递推。从上面的计算过程可以看出, 计算所需要的存储量是:

	参数个数为 4	参数个数为 l
\mathbf{P}_N	$4 \times 5/2 = 10$	$l \times (l+1)/2$
$\boldsymbol{\varphi}$	4	l
$\boldsymbol{\psi}$	4	l
$\hat{\boldsymbol{\theta}}$	4	l
z	1	1(工作单元)
q	1	1(工作单元)
总计	24	$l \times (l+1)/2 + 3l + 2$

由此可见所需存储是相当少的, 而且每次递推的计算量也非常小。

在上例中, 取 $a_1 = 1.4, a_2 = -0.54, b_1 = 0.6, b_2 = -0.46$, 以斜坡函数 $u(k) = 0.1k (k = 1, 2, \dots, 30)$ 作为输入, 产生 30 个数据。然后用递推最小二乘辨识参数, 只经过 6 步就得到相当好的结果。其结果如下:

第 5 步

$$\begin{aligned} a_1 &= 1.392961, \\ a_2 &= -0.5367436, \\ b_1 &= 0.5991358, \\ b_2 &= -0.4542816. \end{aligned}$$

第 6 步

$$a_1 = 1.399749,$$

$$a_2 = -0.5399043,$$

$$b_1 = 0.5999376,$$

$$b_2 = -0.4599445.$$

3.3 时变系统的实时算法

通常的最小二乘法,无论是一次完成算法还是递推算法都是按照最小化性能指标

$$J = \sum_{k=n+1}^{n+N} (y(k) - \boldsymbol{\varphi}_{k-n}^T \boldsymbol{\theta})^2 = (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) \triangleq \mathbf{e}_N^T \mathbf{e}_N$$

来确定估计量。这种算法的特点是每一个数据在计算中所处的地位是完全平等的。如果实际系统是线性定常的,那么这种处理方式是合理的。但如果实际系统为时变情况(即其参数是随时间变化的)时,或者当动态模型作为某种非线性系统的局部近似时,这种不管数据的新老一律平等的做法显然就不太合适。这是因为系统是不断变化的,越新的数据越能反映当前系统的状况,或者说越能带来当前参数的更多信息,因此也就越应该得到重视;而老数据更多地反映了系统变化以前的状况,因此不应该与新数据平等对待。基于这种考虑,为了适应时变系统的辨识,对数据采取渐消记忆或有限记忆的方式,即随着数据的不断积累,将老数据逐渐地遗忘掉,或只记忆最近的有限数据。下面分别讨论基于此两种思想的算法。

3.3.1 渐消记忆(指数窗)的递推算法

性能指标考虑为

$$J = \sum_{k=n+1}^{n+N} \rho^{n+N-k} [y(k) - \boldsymbol{\varphi}_{k-n}^T \boldsymbol{\theta}]^2, \quad 0 < \rho < 1 \quad (3.28)$$

由于权参数 $\rho < 1$, 所以该性能指标的直观意义是:越老的数据,加权越小,即越不被重视;越新的数据,加权越大,越被重视,最新的一次采样数据加权为 1。实际上,性能指标式(3.28)相当于性能指标式(3.13)中加权矩阵 \mathbf{W}_N 取为

$$\mathbf{W}_N = \text{diag}(\rho_{N-1} \rho_{N-2} \cdots \rho_1) \quad (3.29)$$

而

$$\mathbf{W}_{N+1} = \begin{bmatrix} \rho \mathbf{W}_N & \cdots \\ \cdots & \rho \mathbf{1} \end{bmatrix} \quad (3.30)$$

将上式代入式(3.20)和式(3.22),并利用矩阵求逆公式可得实时递推最小二乘算法

$$\hat{\boldsymbol{\theta}}_{N+1} = \hat{\boldsymbol{\theta}}_N + \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1}}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + \rho} [y(n+N+1) - \boldsymbol{\varphi}_{N+1}^T \hat{\boldsymbol{\theta}}_N] \quad (3.31)$$

$$\mathbf{P}_{N+1} = \frac{1}{\rho} \left[\mathbf{P}_N - \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1} \boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + \rho} \right] \quad (3.32)$$

其中 $0 < \rho < 1$ 称为遗忘因子。选择 ρ 的不同值就可得到不同的遗忘效果。 ρ 越小,遗忘速度越快,或者说记忆越短。在实际应用时需根据系统的特性适当选取,如下公式可做参考

$$\rho = 1 - \frac{1}{N} \quad (3.33)$$

其中 N 为渐近记忆长度。

上述算法是按照渐消记忆方式得到的,因此称为渐消记忆法。又因为遗忘老数据是按照指数规律遗忘的,因此也有人称此方法为指数窗法。

另外,由于遗忘因子 ρ 相对于 $\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1}$ 是很小的,又 ρ 一般取值接近于 1,因此在实际应用中,常将式(3.32)和式(3.33)简化为

$$\hat{\boldsymbol{\theta}}_{N+1} = \hat{\boldsymbol{\theta}}_N + \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1}}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + 1} [y(n+N+1) - \boldsymbol{\varphi}_{N+1}^T \hat{\boldsymbol{\theta}}_N] \quad (3.32)'$$

$$\mathbf{P}_{N+1} = \frac{1}{\rho} \left[\mathbf{P}_N - \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1} \boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + 1} \right] \quad (3.33)'$$

除上述实时算法外,较常用的另一种实时算法是限定记忆法,下面讨论该种算法。

3.3.2 限定记忆(固定窗)的递推算法

现在讨论关于对时变系统辨识的另一种数据信息处理方式——限定记忆方法,即每次估计只是用最近的 $(n+N)$ ($N \geq 2n+1$, 即 N 大于或等于需要估计的参数个数) 个数据,每增加一个新数据,就去掉一个最老的数据。

假定数据总数为 $(n+N)$,仍沿用前面的记号,参数的最小二乘估计为

$$\hat{\boldsymbol{\theta}}_N = (\boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N)^{-1} \boldsymbol{\Phi}_N^T \mathbf{y}_N \quad (3.34)$$

按照限定记忆方式,取得新观测数据 $y(n+N+1)$ 后,去掉 $y(1)$,然后形成新的

$$\boldsymbol{\Phi}_{N+1}^* = \begin{bmatrix} \boldsymbol{\varphi}_2^T \\ \boldsymbol{\varphi}_3^T \\ \vdots \\ \boldsymbol{\varphi}_{N+1}^T \end{bmatrix}, \quad \mathbf{y}_{N+1}^* = \begin{bmatrix} y(n+2) \\ y(n+3) \\ \vdots \\ y(n+N+1) \end{bmatrix} \quad (3.35)$$

注意:这里 $\boldsymbol{\Phi}_{N+1}^*$ 和 \mathbf{y}_{N+1}^* 的下标 $(N+1)$ 不再表示其维数,只是表示在第 $(n+N+1)$ 次采样后去掉 $\boldsymbol{\Phi}_N^*$ 的第一行 $\boldsymbol{\varphi}_1^T$,增加最后一行 $\boldsymbol{\varphi}_{N+1}^T$ 形成的新矩阵 $\boldsymbol{\Phi}_{N+1}^*$,以及由 \mathbf{y}_N^* 去掉第一个元素 $y(n+1)$ 、增加最后一个元素 $y(n+N+1)$ 得到的新向量 \mathbf{y}_{N+1}^* ,它们的维数都是不变的($\boldsymbol{\Phi}_N^*$ 为 $N \times 2n$ 维, \mathbf{y}_N^* 为 N 维),而不像前面所介绍的递推算法那样,随着数据增加而增加其行数。

这里 $\boldsymbol{\Phi}^*$, \mathbf{y}^* 也就是正则方程式(3.11)中的 $\boldsymbol{\Phi}$ 和 \mathbf{y} 。于是得到最小二乘估计

$$\hat{\boldsymbol{\theta}}_{N+1} = (\boldsymbol{\Phi}_{N+1}^{*T} \boldsymbol{\Phi}_{N+1}^*)^{-1} \boldsymbol{\Phi}_{N+1}^{*T} \mathbf{y}_{N+1}^* \quad (3.36)$$

将式(3.35)代入式(3.36)得

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{N+1} &= (\boldsymbol{\varphi}_2^T + \cdots + \boldsymbol{\varphi}_{N+1}^T)^{-1} \cdot [\boldsymbol{\varphi}_2 y(n+2) + \cdots + \boldsymbol{\varphi}_{N+1} y(n+N+1)] \\ &= (\boldsymbol{\varphi}_1^T + \boldsymbol{\varphi}_2^T + \cdots + \boldsymbol{\varphi}_{N+1}^T - \boldsymbol{\varphi}_1^T)^{-1} \cdot [\boldsymbol{\varphi}_1 y(n+1) + \boldsymbol{\varphi}_2 y(n+2) \\ &\quad + \cdots + \boldsymbol{\varphi}_{N+1} y(n+N+1) - \boldsymbol{\varphi}_1 y(n+1)] \\ &= [\boldsymbol{\Phi}_N^{*T} \boldsymbol{\Phi}_N^* + \boldsymbol{\varphi}_{N+1}^T \boldsymbol{\varphi}_{N+1} - \boldsymbol{\varphi}_1^T \boldsymbol{\varphi}_1]^{-1} \cdot [\boldsymbol{\Phi}_N^{*T} \mathbf{y}_N^* + \boldsymbol{\varphi}_{N+1} y(n+N+1) - \boldsymbol{\varphi}_1 y(n+1)] \end{aligned} \quad (3.37)$$

于是有

$$\begin{aligned} &[\boldsymbol{\Phi}_N^{*T} \boldsymbol{\Phi}_N^* + \boldsymbol{\varphi}_{N+1}^T \boldsymbol{\varphi}_{N+1} - \boldsymbol{\varphi}_1^T \boldsymbol{\varphi}_1] \hat{\boldsymbol{\theta}}_{N+1} \\ &= \boldsymbol{\Phi}_N^{*T} \mathbf{y}_N^* + \boldsymbol{\varphi}_{N+1} y(n+N+1) - \boldsymbol{\varphi}_1 y(n+1) \end{aligned} \quad (3.38)$$

记

$$\boldsymbol{\varphi}_{N+1}^T \hat{\boldsymbol{\theta}}_N \triangleq \hat{y}(n+N+1) \quad (3.39)$$

$$\boldsymbol{\varphi}_1^T \hat{\boldsymbol{\theta}}_N \triangleq \hat{y}(n+1) \quad (3.40)$$

分别表示对实际输出量测 $y(n+N+1)$, $y(n+1)$ 的预测。由式(3.39)、式(3.40)及式(3.36)

式还可以得到另一个表达式

$$\begin{aligned}
& (\Phi_N^{*T} \Phi_N^* + \varphi_{N+1} \varphi_{N+1}^T - \varphi_1 \varphi_1^T) \hat{\theta}_N \\
&= \Phi_N^{*T} \Phi_N^* \hat{\theta}_N + \varphi_{N+1} \varphi_{N+1}^T \hat{\theta}_N - \varphi_1 \varphi_1^T \hat{\theta}_N \\
&= \Phi_N^{*T} y_N^* + \varphi_{N+1} \hat{y}(n+N+1) - \varphi_1 \hat{y}(n+1)
\end{aligned} \quad (3.41)$$

式(3.38)减去式(3.41)得

$$\begin{aligned}
& (\Phi_N^{*T} \Phi_N^* + \varphi_{N+1} \varphi_{N+1}^T - \varphi_1 \varphi_1^T) (\hat{\theta}_{N+1} - \hat{\theta}_N) \\
&= \varphi_{N+1} \Delta y(n+N+1) - \varphi_1 \Delta y(n+1)
\end{aligned} \quad (3.42)$$

其中

$$\Delta y(n+N+1) \triangleq y(n+N+1) - \hat{y}(n+N+1) \quad (3.43)$$

$$\Delta y(n+1) \triangleq y(n+1) - \hat{y}(n+1) \quad (3.44)$$

分别表示对 $y(n+N+1)$, $y(n+1)$ 的预测误差, 所以有

$$\hat{\theta}_{N+1} = \hat{\theta}_N + (\Phi_N^{*T} \Phi_N^* + \varphi_{N+1} \varphi_{N+1}^T - \varphi_1 \varphi_1^T)^{-1} \cdot [\varphi_{N+1} \Delta y(n+N+1) - \varphi_1 \Delta y(n+1)] \quad (3.45)$$

令

$$Q_{K+1} \triangleq (\Phi_K^{*T} \Phi_K^* + \varphi_{K+1} \varphi_{K+1}^T)^{-1}, \quad K \geq N \quad (3.46)$$

$$P_K \triangleq (\Phi_K^{*T} \Phi_K^*)^{-1}, \quad K \geq N \quad (3.47)$$

$$\begin{aligned}
P_{K+1} &= (\Phi_{K+1}^{*T} \Phi_{K+1}^*)^{-1} \\
&= (\Phi_K^{*T} \Phi_K^* + \varphi_{K+1} \varphi_{K+1}^T - \varphi_{K-N+1} \varphi_{K-N+1}^T)^{-1} \\
&= (Q_{K+1}^{-1} - \varphi_{K-N+1} \varphi_{K-N+1}^T)^{-1}, \quad K \geq N
\end{aligned} \quad (3.48)$$

对 Q_{K+1} , P_{K+1} 分别用矩阵求逆公式得

$$Q_{K+1} = (P_K^{-1} + \varphi_{K+1} \varphi_{K+1}^T)^{-1} = P_K - \frac{P_K \varphi_{K+1} \varphi_{K+1}^T P_K}{\varphi_{K+1}^T P_K \varphi_{K+1} + 1} \quad (3.49)$$

$$P_{K+1} = Q_{K+1} - \frac{Q_{K+1} \varphi_{K-N+1} \varphi_{K-N+1}^T Q_{K+1}}{\varphi_{K-N+1}^T Q_{K+1} \varphi_{K-N+1} + 1} \quad (3.50)$$

$$\hat{\theta}_{K+1} = \hat{\theta}_K + P_{K+1} [\varphi_{K+1} \Delta y(n+K+1) - \varphi_{K-N+1} \Delta y(n+K-N+1)] \quad (3.51)$$

用式(3.49)~式(3.51), 从 $K=N$ 开始递推, $P_N = (\Phi_N^{*T} \Phi_N^*)^{-1}$ 和 $\hat{\theta}_N$ 可以用一次完成最小二乘或者递推最小二乘对 $(n+N)$ 个数据计算得到。在此基础上不再增加记忆长度, 每次只保留 $(n+N)$ 个数据, 参数估计按照上面的方式递推, 计算路线简图如图 3.3 所示。

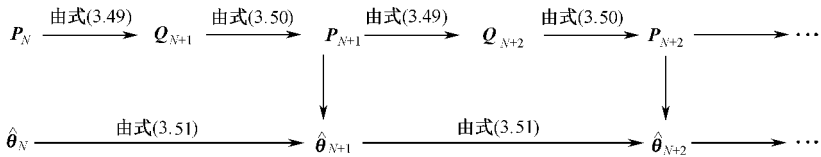


图 3.3 限定记忆的递推算法路线图

3.3.3 变遗忘因子的实时算法

前面讨论的实时算法, 对于变化平稳的时变系统来说, 只要适当选取遗忘因子或者记忆长度, 可以得到相当好的辨识效果。但是, 在实际中还有另外一些时变系统, 它们的动态特性不

是按照基本相同的规律变化,而是有时变化快,有时变化慢,有时还可能有突然的变化。对于这类系统,用选定遗忘因子或记忆长度的方法很难得到满意的结果。这是因为,如果选用较小的遗忘因子或较短的记忆长度,在系统较稳定的时候,从数据中得到的信息就很少,这将导致参数估计的误差协方差按指数增大,这将对干扰非常灵敏;如果遗忘因子选得较大($\rho \approx 1$),能记忆很远的过去的的数据,这样对系统的突然变化就反应不灵敏,因而不能很好地跟踪系统的变化。所以,对于动态特性变化较大的系统,需另想良策。

这里将要介绍的变遗忘因子的辨识算法就是针对上述情况提出来的。其基本思想是随着系统动态特性的变化自动调整遗忘因子。当系统有突然变化时,则选择较小的遗忘因子来提高灵敏度,直至调整到参数的估计误差较小;当系统趋于稳定时,选择较大的遗忘因子,增长记忆长度。总之,通过修正遗忘因子 ρ ,使得记忆的数据始终保持系统的一定信息量。

由于矩阵 Φ_N 是由 $(n+N)$ 次采样得到的所有输入、输出数据构成的,所以它包含了系统的动态信息,因此称 $\Phi_N^T \Phi_N$ 为信息矩阵。其信息量为先验误差的加权平方和,记为 \sum 。它有如下递推公式(Fortescue, T. R 等, 1981)

$$\sum_k = \rho_k \sum_{k-1} + [1 - \Phi_k^T K_k] \epsilon_k^2 \quad (3.52)$$

其中

$$K_k = \frac{P_{k-1} \Phi_k}{\Phi_k^T P_{k-1} \Phi_k + 1} \quad (3.53)$$

$$\epsilon_k = y(n+k) - \Phi_k^T \hat{\theta}_{k-1} \quad (3.54)$$

ρ_k 为遗忘因子,其他符号同前。选择遗忘因子 ρ 的目标是使信息矩阵的信息保持不变,即使得

$$\sum_k = \sum_{k-1} = \dots = \sum_0$$

这就是说,在每一步忘掉的信息量相当于在最近一次量测中得到的新信息量,总信息量保持不变。基于上述思想,取变遗忘因子 ρ_k ,使得满足

$$\sum_0 = \rho_k \sum_0 + [1 - \Phi_k^T K_k] \epsilon_k^2$$

即取

$$\rho_k = 1 - (1 - \Phi_k^T K_k) \epsilon_k^2 / \sum_0$$

于是得到变遗忘因子的实时算法如下:

- (1) 给定初值 θ_0, ρ_0 和 \sum_0 ;
- (2) 输入 n 组数据 $y(1), \dots, y(n); u(1), \dots, u(n)$;
- (3) 置 $k=1$;
- (4) 进行第 $(n+k)$ 次采样,得 $y(n+k), u(n+k)$;
- (5) 计算预测误差 $\epsilon_k = y(n+k) - \Phi_k^T \hat{\theta}_{k-1}$;
- (6) 计算增益 $K_k = P_{k-1} \Phi_k / (\Phi_k^T P_{k-1} \Phi_k + 1)$;
- (7) 修正估计 $\hat{\theta}_k = \hat{\theta}_{k-1} + K_k \epsilon_k$;
- (8) 计算遗忘因子: $\rho_k = 1 - (1 - \Phi_k^T K_k) \epsilon_k^2 / \sum_0$, 当 $\rho_k \leq \rho_{\min}$ 时,置 $\rho_k = \rho_{\min}$;
- (9) 计算协方差 $P_k = (1 - K_k \Phi_k^T) P_{k-1} / \rho_k$;
- (10) $k+1 \Rightarrow k$, 返回(4)。

在上述步骤中, θ_0 和 P_0 的选取同递推最小二乘法。 \sum_0 可取为 $\sigma_0^2 N_0$, 这里 σ_0^2 是量测噪声

方差,它可以根据实际过程的知识得到,或者通过实测数据用估计方法得到。 N_0 是渐近记忆长度。另外,在第(8)步中,对算出的遗忘因子进行了检验,以防止由于计算误差或者偶尔的干扰造成 ρ 变得过小。

3.4 递推平方根算法

在3.2节所讨论的递推最小二乘法具有计算量小、存储少的优点,因此得到了广泛的应用。但是,在实际的在线应用过程中,往往出现这种情况:由于计算机的字长是有限的(尤其是微型控制机),因而存在舍入误差,较长时间地应用递推算法引起误差的积累,这样就逐渐降低递推算法的精度,从而破坏估计误差协方差阵 P_k 的正定性,以至于在控制系统中引起发散现象。

为了减少这种误差及其积累对辨识造成的影响,应设法尽量减少对计算机所要求的有效数字的位数,于是递推平方根算法就因此而产生。

递推平方根算法的基本思想是:利用协方差阵 P_k 的正定性,将其分解为互为转置的上三角阵和下三角阵的乘积,即 $P_k = S_k S_k^T$, S_k 为上三角阵。然后给出 S_k 的递推公式(从而降低了对字长的要求),并与参数向量 $\hat{\theta}_k$ 的递推公式相结合构成递推平方根算法。下面简单地介绍其推导过程。

式(3.26)可改写为

$$P_k = P_{k-1} - P_{k-1} \phi_k [\phi_k^T P_{k-1} \phi_k + 1]^{-1} \phi_k^T P_{k-1} \quad (3.55)$$

引进分解式 $P_j = S_j S_j^T$,则上式变为

$$P_k = S_{k-1} S_{k-1}^T - S_{k-1} S_{k-1}^T \phi_k [\phi_k^T S_{k-1} S_{k-1}^T \phi_k + 1]^{-1} \phi_k^T S_{k-1} S_{k-1}^T$$

令

$$f_k = S_{k-1}^T \phi_k, \quad g_k = [f_k^T f_k + 1]^{-1} \quad (3.56)$$

则

$$P_k = S_{k-1} [I - f_k g_k f_k^T] S_{k-1}^T \quad (3.57)$$

再将 $I - f_k g_k f_k^T$ 做如下分解

$$I - f_k g_k f_k^T = [I - g_k \alpha_k f_k f_k^T] [I - g_k \alpha_k f_k f_k^T] \quad (3.58)$$

其中 α_k 为待定参数。将上式展开并整理得

$$I - f_k g_k f_k^T = I - f_k g_k [2\alpha_k - g_k \alpha_k^2 f_k^T f_k] f_k^T$$

比较上式两端可知,若等号成立, α_k 应满足

$$2\alpha_k - g_k \alpha_k^2 f_k^T f_k = 1$$

而

$$g_k f_k^T f_k = [f_k^T f_k + 1]^{-1} f_k^T f_k = 1 - [f_k^T f_k + 1]^{-1} = 1 - g_k$$

故有 $2\alpha_k - (1 - g_k) \alpha_k^2 = 1$ 。解此方程得

$$\alpha_k = \frac{1 \pm \sqrt{g_k}}{1 - g_k}$$

由于 $0 < g_k < 1$,若取 $0 < \alpha_k < 1$,故得

$$\alpha_k = \frac{1 - \sqrt{g_k}}{1 - g_k} = \frac{1}{1 + \sqrt{g_k}} \quad (3.59)$$

取 α_k 如式(3.59)给出, 因此有

$$P_k = S_{k-1} [I - g_k \alpha_k f_k f_k^T] [I - g_k \alpha_k f_k f_k^T] S_{k-1}^T = S_k S_k^T$$

其中

$$S_k = S_{k-1} [I - g_k \alpha_k f_k f_k^T] \quad (3.60)$$

令

$$K_k = P_{k-1} \varphi_k [\varphi_k^T P_{k-1} \varphi_k + 1]^{-1} = S_{k-1} f_k g_k \quad (3.61)$$

则式(3.25)变为

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k [y(n+k) - \varphi_k^T \hat{\theta}_{k-1}] \quad (3.62)$$

式(3.56)、式(3.59)~式(3.62)构成了平方根算法的递推算法。递推初始值为: $\hat{\theta}_0 = 0$, $S_0 = \alpha^2 I$, 其中 α^2 为充分大的正数。其算法框图如图 3.4 所示。

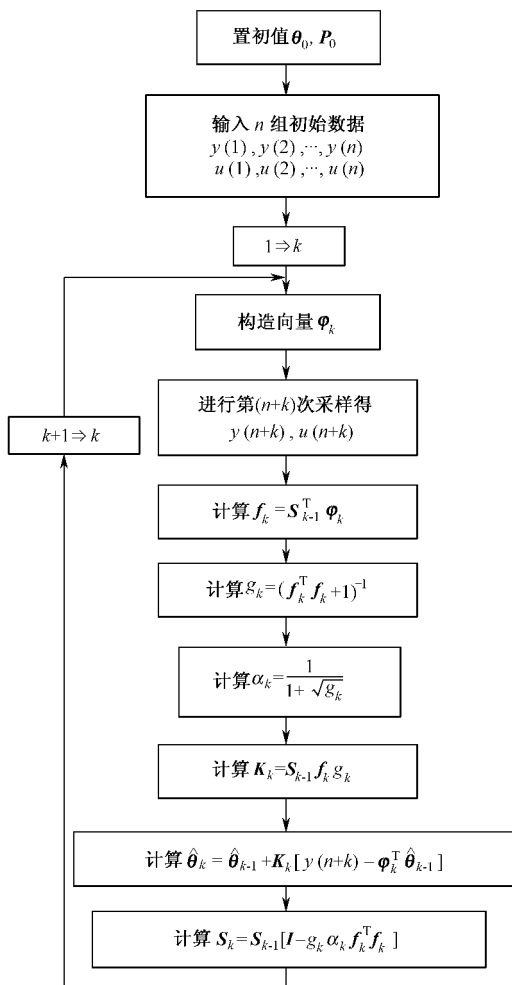


图 3.4 递推平方根法计算框图

递推平方根算法既保持了递推最小二乘法简单易行的优点, 又能提高精度, 因此在在线辨

识和自校正控制系统中得到了广泛的应用。

【例 3.2】 用伪随机二位式序列作为输入,由如下模型产生 200 组数据

$$y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-1) + 0.5u(k-2) + \lambda e(k)$$

其中, $e(k)$ 是均值为零、方差为 1 的正态伪随机数。

解 分别用递推最小二乘和平方根法做辨识。从辨识结果我们发现,当 $\lambda=0$ 时,两种方法一样;当 $\lambda=1$ 时,平方根法优于最小二乘法,其结果见表 3.1 (其中 $\hat{\theta}_0=0, P_0=10^6 I, S_0=10^3 I$)。

表 3.1 例 3.2 辨识结果表

算 法	递 推 次 数	辨 识 结 果				$\left[\frac{1}{100} \sum_{k=3}^{103} \epsilon^2(k) \right]^{1/2}$
		a_1	a_2	b_1	b_2	
递推最小二乘	100	-1.415549	0.6406413	0.8963724	0.4671259	1.039221
递推平方根法	100	-1.460463	0.6697421	0.9630480	0.4874953	1.028073
真 值		-1.5	0.7	1.0	0.5	$\lambda=1$

注 1:本章前面所讲的方法都可以简单地推广到多变量情形。

注 2:对于动态系统,由于关于系统的动态特性很难精确把握,所以给出的模型结构很难准确;或者参数太多;或者采样间隔太小;或者数据有明显的趋势性变化,这些都容易造成用于解最小二乘估计的正则方程呈病态状况。在 2.3 节给出的病态方程的求解方法也适用于动态系统。

注 3:对于动态系统所采用的最小二乘估计为

$$\hat{\theta}_N = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T \Phi_N$$

形式上与静态情况一样,但是,它与静态情况有本质的不同,主要是这里 Φ_N 中包含输出。因此,估计量 $\hat{\theta}_N$ 不再有那样强的性质。对动态系统的最小二乘估计主要的性质是:在相当宽的条件下,主要是当残差序列 $\{e(k)\}$ 为白噪声时,参数的最小二乘估计是相容的。即当 $N \rightarrow \infty$ 时, $\hat{\theta}_N$ 以概率收敛到真值 Q , 即 $\hat{\theta}_N \xrightarrow{p} \theta$ 。当然 $\hat{\theta}_N$ 也是渐近无偏的(Ljung, L, 1976)。

注 4:卡尔曼滤波器(卢桂章, 1982)除了用来做动态系统的状态估计外,也可以用于系统参数的在线辨识。特别是对于时变参数,卡尔曼滤波是一种比较有效的辨识方法。

假定未知时变参数向量记为 θ , 它的变化过程可以描述为

$$\theta(k+1) = \theta(k) + w(k) \tag{3.63}$$

这里

$$\begin{aligned} \hat{\theta}^T(k) &= (a_1(k), \dots, a_n(k), b_1(k), \dots, b_n(k)) \\ w^T(k) &= (w_1(k), w_2(k), \dots, w_{2n}(k)) \end{aligned}$$

其中, $\{w_i(k)\} (i=1, \dots, 2n)$ 表示参数中的噪声部分。假设它们是零均值、正态白噪声序列, 且与 $\{e(k)\}$ 独立。系统模型为

$$y(k) = \Phi_{k-1}^T \theta(k) + e(k) \tag{3.64}$$

这样,式(3.63)和式(3.64)就构成以 $\theta(k)$ 为状态、以 $y(k)$ 为输出的状态空间模型,直接应用卡尔曼滤波公式即可得到 $\theta(k)$ 的估计。

3.5 最大似然估计(ML)

在 2.4 中给出了静态模型参数的最大似然估计,对于线性动态模型,也可以完全类似地得到。为了书写方便,将系统模型改写为如下形式

$$\mathbf{A}(Z^{-1})\mathbf{y}(k) = \mathbf{B}(Z^{-1})\mathbf{u}(k) + \mathbf{e}(k) \quad (3.65)$$

其中

$$\begin{aligned} \mathbf{A}(Z^{-1}) &= 1 + a_1 Z^{-1} + \cdots + a_n Z^{-n} \\ \mathbf{B}(Z^{-1}) &= b_1 Z^{-1} + b_2 Z^{-2} + \cdots + b_n Z^{-n} \end{aligned} \quad (3.66)$$

称为移位多项式; Z^{-1} 称为后移算子,即 $Z^{-i}\mathbf{y}(k) = \mathbf{y}(k-i)$, $Z^{-i}\mathbf{u}(k) = \mathbf{u}(k-i)$ 。 $\{\mathbf{e}(k)\}$ 为白噪声序列,且与输入 $\{\mathbf{u}(k)\}$ 无关。

其向量形式

$$\mathbf{y}_N = \Phi_N \boldsymbol{\theta} + \mathbf{e}_N \quad (3.67)$$

其中符号同式(3.5)和式(3.10)。

如果 $\{\mathbf{e}(k)\}$ 是零均值、方差为 σ^2 的正态白噪声序列,且与 $\{\mathbf{u}(k)\}$ 无关时,那么似然函数为

$$\begin{aligned} L(\mathbf{y}_N, \boldsymbol{\theta}, \sigma^2) &= (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}_N - \Phi_N \boldsymbol{\theta})^T (\mathbf{y}_N - \Phi_N \boldsymbol{\theta})\right\} \\ &= (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{k=1}^N e^2(k)\right\} \end{aligned} \quad (3.68)$$

取对数得

$$\log L(\mathbf{y}_N, \boldsymbol{\theta}, \sigma^2) = -\frac{N}{2} \log 2\pi - \frac{N}{2} \log \sigma^2 - \frac{1}{2} \frac{(\mathbf{y}_N - \Phi_N \boldsymbol{\theta})^T (\mathbf{y}_N - \Phi_N \boldsymbol{\theta})}{\sigma^2} \quad (3.69)$$

上式分别对未知参数 $\boldsymbol{\theta}$ 和 σ^2 求偏导数,并令其等于零。得

$$\frac{\partial \log(\mathbf{y}_N, \boldsymbol{\theta}, \sigma^2)}{\partial \boldsymbol{\theta}} = \frac{1}{\sigma^2} (\Phi_N^T \mathbf{y}_N - \Phi_N^T \Phi_N \boldsymbol{\theta}) = 0 \quad (3.70)$$

$$\frac{\partial \log(\mathbf{y}_N, \boldsymbol{\theta}, \sigma^2)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{(\mathbf{y}_N - \Phi_N \boldsymbol{\theta})^T (\mathbf{y}_N - \Phi_N \boldsymbol{\theta})}{2\sigma^4} = 0 \quad (3.71)$$

由式(3.70)可解得参数 $\boldsymbol{\theta}$ 的最大似然估计

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T \mathbf{y}_N \quad (3.72)$$

由式(3.71)可以解出噪声方差 σ^2 的最大似然估计

$$\hat{\sigma}_{\text{ML}}^2 = \frac{1}{N} (\mathbf{y}_N - \Phi_N \hat{\boldsymbol{\theta}})^T (\mathbf{y}_N - \Phi_N \hat{\boldsymbol{\theta}}) = \frac{1}{N} \sum_{k=1}^N e(k)^2 \quad (3.73)$$

其中

$$e(k) = \mathbf{y}(n+k) - \boldsymbol{\varphi}_k^T \hat{\boldsymbol{\theta}}_{\text{ML}} \quad (3.74)$$

由式(3.72)可见,对于噪声为零均值、正态白噪声的情况,最大似然估计和最小二乘估计是相同的。

习 题

1. 递推最小二乘法与一次完成最小二乘法是否等价?递推最小二乘法的优点是什么?

2. 写出多变量情况的最小二乘递推公式。

3. 利用矩阵求逆公式,推导公式(3.22)和式(3.23)。

4. 已知模型

$$y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-1) + 0.5u(k-2) + e(k)$$

其中, $e(k) \sim N(0,1)$ 分布,用伪随机二位式序列作为输入,由此模型产生 100 组数,分别

用递推最小二乘法和平方根递推算法估计参数,并比较平均残差平方和 $J = \frac{1}{N} \sum_{k=n+1}^{n+N} e^2(k)$ 。

5. 推导渐消记忆的递推平方根算法。

6. 考虑模型

$$(I) \quad y(k) + 0.8y(k-1) = 0.5u(k-1) + e(k)$$

$$(II) \quad y(k) + 0.6y(k-1) = 0.3u(k-1) + e(k)$$

其中, $\{e[k]\}$ 为零均值、标准差为 0.05 的正态伪随机序列,用伪随机二位式序列作为输入,先由模型(I)产生采样数据,到第 200 次采样后变为由模型(II)产生采样数据,也取 200 次,共取 400 对数据。用这组数据分别用递推最小二乘、带遗忘因子的最小二乘($\rho=0.98$)及变遗忘因子的最小二乘($\rho_{\min}=0.5, \Sigma$ 取为 2.5)估计参数,并画图比较其结果。

第 4 章 相关(有色)噪声情形的辨识算法

前几章讨论了最小二乘估计的一系列算法。当残差序列 $\{e(k)\}$ 为白噪声(即不相关随机序列)时,最小二乘估计量具有无偏性和相容性。但是当残差序列为相关噪声(即相关序列)时,这种优良性质便遭到了破坏。下面通过具体计算,简单地分析一下优良性遭到破坏的原因。

考虑模型

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_1 u(k-1) + \cdots + b_n u(k-n) + \varepsilon(k)$$

其中, $\{\varepsilon(k)\}$ 为残差序列。由式(3.12)可知,参数向量 θ 的最小二乘估计为

$$\begin{aligned}\hat{\theta}_N &= (\Phi_N^T \Phi_N)^{-1} \Phi_N^T y_N = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T (\Phi_N \theta + \varepsilon_N) \\ &= \theta + (\Phi_N^T \Phi_N)^{-1} \Phi_N^T \varepsilon_N\end{aligned}\quad (4.1)$$

其中, $\varepsilon_N^T = (\varepsilon(n+1), \varepsilon(n+2), \dots, \varepsilon(n+N))$ 。其他符号与 3.1 节相同。

再由 Φ_N 的定义,式(4.1)可写成

$$\hat{\theta}_N = \theta + \left[\frac{1}{N} \sum_{k=1}^N \varphi_k \varphi_k^T \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^N \varphi_k \varepsilon(n+k) \right] \quad (4.2)$$

假定 $\{u(k)\}, \{\varepsilon(k)\}$ 是平稳、遍历的(克拉美,1966),则有

$$\frac{1}{N} \sum_{k=1}^N \varphi_k \varphi_k^T \xrightarrow{p} R_{\varphi\varphi} \triangleq E \varphi_k \varphi_k^T \quad (4.3)$$

$$\frac{1}{N} \sum_{k=1}^N \varphi_k \varepsilon(n+k) \xrightarrow{p} R_{\varphi\varepsilon} \triangleq E \varphi_k \varepsilon(n+k) \quad (4.4)$$

其中记号 $a \xrightarrow{p} b$, 表示 a 以概率收敛到 b ; $E(\cdot)$ 表示 (\cdot) 的数学期望; R_{xy} 表示 x, y 的相关函数(阵)。因此有

$$\hat{\theta}_N \xrightarrow{p} \theta + R_{\varphi\varphi}^{-1} R_{\varphi\varepsilon} \quad (4.5)$$

当 $\{\varepsilon(k)\}$ 为零均值、不相关序列时,由于 φ_k 只与 $\varepsilon(n+k-1), \dots, \varepsilon(k)$ 相关,与 $\varepsilon(n+k)$ 无关,所以 $R_{\varphi\varepsilon} = 0$,从而有 $\hat{\theta}_N \xrightarrow{p} \theta$ 。

但若 $\{\varepsilon(k)\}$ 为相关序列时,即存在 $t > 0$,使得 $R_{\varepsilon\varepsilon}(t) \neq 0$,则 $R_{\varphi\varepsilon} \neq 0$,从而 $\hat{\theta}_N$ 不再以概率趋于 θ 。由此可见,当噪声序列相关时,最小二乘估计的优良性质就不一定保持了。因此需要考虑其他方法来得到好的估计,辅助变量法就是其中一种。下面将从直观分析出发导出辅助变量算法。

4.1 辅助变量法

从式(4.5)可知, $\hat{\theta}_N$ 不收敛到真值 θ 的一个原因是 $R_{\varphi\varepsilon} \neq 0$ 。由此可见,如果存在与 Φ_N 同样维数的矩阵 Z_N ,使得

$$\frac{1}{N} Z_N^T \Phi_N \xrightarrow{p} R_{z\varphi} \text{ 正定} \quad (4.6)$$

且

$$\frac{1}{N} \mathbf{Z}_N^T \boldsymbol{\varepsilon}_N \xrightarrow{p} \mathbf{R}_{z\varepsilon} = 0 \quad (4.7)$$

则由式(4.2)知

$$\hat{\boldsymbol{\theta}}_N = (\mathbf{Z}_N^T \boldsymbol{\Phi}_N)^{-1} \mathbf{Z}_N^T \mathbf{y}_N \xrightarrow{p} \boldsymbol{\theta} \quad (4.8)$$

我们把满足式(4.6)和式(4.7)的矩阵 \mathbf{Z}_N 称为辅助变量阵,它的元素称为辅助变量。问题是如何构造辅助变量阵 \mathbf{Z}_N ,使其满足式(4.6)和式(4.7)。从直观分析可知,式(4.7)意味着辅助变量与噪声序列 $\{\varepsilon(k)\}$ 不相关,式(4.6)则意味着辅助变量与 $\{u(k)\}, \{y(k)\}$ 强相关。因此,一个很自然的想法是用下面的辅助模型来产生辅助变量。

辅助模型为

$$\begin{aligned} \hat{y}(k) + \hat{\alpha}_1 \hat{y}(k-1) + \cdots + \hat{\alpha}_n \hat{y}(k-n) \\ = \hat{\beta}_1 u(k-1) + \cdots + \hat{\beta}_n u(k-n) \end{aligned} \quad (4.9)$$

其中,参数 $\hat{\boldsymbol{\beta}}^T = (\hat{\alpha}_1, \cdots, \hat{\alpha}_n, \hat{\beta}_1, \cdots, \hat{\beta}_n)$ 是通过原始数据得到的最小二乘估计; $\hat{y}(\cdot)$ 是模型的输出。

令

$$\begin{aligned} \hat{\boldsymbol{\phi}}_k^T &= [-\hat{y}(n+k-1), \cdots, -\hat{y}(k), u(n+k-1), \cdots, u(k)] \\ \mathbf{Z}_N^T &= [\hat{\boldsymbol{\phi}}_1, \hat{\boldsymbol{\phi}}_2, \cdots, \hat{\boldsymbol{\phi}}_N] \end{aligned}$$

则 \mathbf{Z}_N 即可作为辅助变量阵。这样得到的 \mathbf{Z}_N 是否满足式(4.6)和式(4.7)一般来说很难验证。但是当 $u(k)$ 能精确地量测且与 $\varepsilon(k)$ 无关时,满足式(4.7)则是显然的。

综合上述,辅助变量法的计算步骤可归纳如下:

(1) 用最小二乘法辨识模型

$$y(k) + \alpha_1 y(k-1) + \cdots + \alpha_n y(k-n) = \beta_1 u(k-1) + \cdots + \beta_n u(k-n)$$

得参数向量估计 $\hat{\boldsymbol{\beta}}$;

(2) 用模型式(4.9)计算模型输出 $\{\hat{y}(k)\}$;

(3) 用 $\{y(k)\}, \{u(k)\}$ 构造 $\boldsymbol{\Phi}_N$, 用 $\{\hat{y}(k)\}, \{u(k)\}$ 构造 \mathbf{Z}_N , 用 $\{y(k)\}$ 构造 \mathbf{y}_N ;

(4) 解方程组

$$\mathbf{Z}_N^T \boldsymbol{\Phi}_N \boldsymbol{\theta} = \mathbf{Z}_N^T \mathbf{y}_N$$

得到原系统的参数估计 $\boldsymbol{\theta}$ 。

与最小二乘法进行完全类似的推导,可以得到递推算法

$$\begin{aligned} \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \frac{\mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1}}{1 + \boldsymbol{\phi}_{k+1}^T \mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1}} [y(n+k+1) - \boldsymbol{\phi}_{k+1}^T \hat{\boldsymbol{\theta}}_k] \\ \mathbf{P}_{k+1} &= \mathbf{P}_k + \frac{\mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1} \boldsymbol{\phi}_{k+1}^T \mathbf{P}_k}{1 + \boldsymbol{\phi}_{k+1}^T \mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1}} \end{aligned}$$

其中

$$\begin{aligned} \hat{\boldsymbol{\phi}}_{k+1}^T &= [-\hat{y}(n+k), \cdots, \hat{y}(k+1), u(n+k), \cdots, u(k+1)] \\ \boldsymbol{\phi}_{k+1}^T &= [-y(n+k), \cdots, y(k+1), u(n+k), \cdots, u(k+1)] \\ \hat{y}(n+k) &= \hat{\boldsymbol{\phi}}_k^T \hat{\boldsymbol{\theta}}_k, \quad k = 1, 2, \cdots \end{aligned}$$

递推初始值取为: $\hat{\boldsymbol{\theta}}_0 = 0, \mathbf{P}_0 = \alpha^2 \mathbf{I}, \alpha$ 为充分大的正数; $\hat{y}(1) = \hat{y}(2) = \cdots = \hat{y}(n) = 0$ 或为实测值。

提醒读者注意,在递推算法中,每次替换 $\hat{\boldsymbol{\phi}}$ 中的一个元素,参数 $\hat{\boldsymbol{\theta}}$ 也逐次变化,因此,它与

一次完成算法是不等价的。另外,辅助变量的选取也不是惟一的。

与通常递推最小二乘一样,递推算法可以通过在矩阵 \mathbf{P} 的递推公式中增加遗忘因子,变为适应算法。

4.2 增广最小二乘法 (ELS)

4.2.1 增广最小二乘法

增广最小二乘法,也称为增广矩阵法(Extended Matrix Method)。

考虑系统模型为

$$\begin{aligned} y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) \\ = b_1 u(k-1) + \cdots + b_n u(k-n) + e(k) + c_1 e(k-1) + \cdots + c_n e(k-n) \end{aligned} \quad (4.10)$$

其中 $\{e(k)\}$ 为零均值的正态白噪声序列。

如果 $\{e(k)\}$ 是能够量测的,则可将 $e(k)$ 视为确定性输入扩充到向量中去,这时模型式(4.10)可写为

$$y(k) = \boldsymbol{\phi}_{k-n}^T \boldsymbol{\theta} + e(k)$$

其中

$$\begin{aligned} \boldsymbol{\phi}_{k-n}^T &= [-y(k-1), \cdots, -y(k-n), u(k-1), \cdots, u(k-n), e(k-1), \cdots, e(k-n)] \\ \boldsymbol{\theta}^T &= [a_1, \cdots, a_n, b_1, \cdots, b_n, c_1, \cdots, c_n] \end{aligned}$$

利用最小二乘法的结果,立即得到参数向量 $\boldsymbol{\theta}$ 的估计算法及其递推公式。

但是 $e(k)$ 是不可测的。一个很自然的想法是将 $e(k)$ 用其估计来代替。这时 $\boldsymbol{\phi}_{k-n}$ 变为 $\hat{\boldsymbol{\phi}}_{k-n}$, 这里

$$\hat{\boldsymbol{\phi}}_{k-n}^T \triangleq [-y(k-1), \cdots, -y(k-n), u(k-1), \cdots, u(k-n), \hat{e}(k-1), \cdots, \hat{e}(k-n)] \quad (4.11)$$

$$\hat{e}(k) = y(k) - \hat{\boldsymbol{\phi}}_{k-n}^T \hat{\boldsymbol{\theta}}_{k-1} \quad (4.12)$$

其中 $\hat{\boldsymbol{\theta}}_{k-1}$ 是参数 $\boldsymbol{\theta}$ 的第 $(k-1)$ 次估计值,其递推辨识算法为

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \frac{\mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1}}{\hat{\boldsymbol{\phi}}_{k+1}^T \mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1} + 1} [y(n+k+1) - \hat{\boldsymbol{\phi}}_{k+1}^T \hat{\boldsymbol{\theta}}_k] \quad (4.13)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1} \hat{\boldsymbol{\phi}}_{k+1}^T \mathbf{P}_k}{\hat{\boldsymbol{\phi}}_{k+1}^T \mathbf{P}_k \hat{\boldsymbol{\phi}}_{k+1} + 1} \quad (4.14)$$

增广最小二乘法在许多方面获得了成功的应用,但对一些特殊情况可能不收敛,Ljung (1975) 给出了不收敛的反例。同时提出了增广最小二乘法的改进算法,称为改进的增广最小二乘法。下面给出该算法的简单推导。

4.2.2 改进的增广最小二乘法

为了描述方便,将系统模型式(4.10)改写为如下形式

$$\mathbf{A}(Z^{-1})y(k) = \mathbf{B}(Z^{-1})u(k) + \mathbf{C}(Z^{-1})e(k) \quad (4.15)$$

其中

$$\begin{cases} \mathbf{A}(Z^{-1}) = 1 + a_1 Z^{-1} + \cdots + a_n Z^{-n} \\ \mathbf{B}(Z^{-1}) = b_1 Z^{-1} + \cdots + b_n Z^{-n}, & b_1 \neq 0 \\ \mathbf{C}(Z^{-1}) = 1 + c_1 Z^{-1} + \cdots + c_n Z^{-n} \end{cases} \quad (4.16)$$

称为移位多项式; Z^{-1} 称为后移算子, 即 $Z^{-i}y(k) = y(k-i)$, $Z^{-i}u(k) = u(k-i)$, $Z^{-i}e(k) = e(k-i)$ 。 $\{e(k)\}$ 为零均值、方差为 σ^2 的正态白噪声序列, 且与输入 $u(k)$ 无关。

仍记参数向量 $\boldsymbol{\theta}^T = (a_1, \dots, a_n, b_1, \dots, b_n, c_1, \dots, c_n)$, 我们的目标是求参数 $\boldsymbol{\theta}$ 的估计, 使得性能指标

$$J_N(\boldsymbol{\theta}) = \sum_{k=1}^N e_k^2(\boldsymbol{\theta}) \quad (4.17)$$

达到极小, 其中

$$e_k(\boldsymbol{\theta}) = \mathbf{C}^{-1}(Z^{-1})[\mathbf{A}(Z^{-1})y(k) - \mathbf{B}(Z^{-1})u(k)] \quad (4.18)$$

注意, 这里性能指标式(4.17)形式上似乎也像通常的最小二乘法中“误差平方和”一样, 但实际上有很大不同, 这里 $e_k(\boldsymbol{\theta})$ 不再是参数的线性函数, 因此不能再简单地用求导数找极值的方法来导出参数的估计和递推算法。但以前的方法给了我们启发, 我们首先将 $e_k(\boldsymbol{\theta})$ 通过泰勒展开将其线性化, 进而将 $J_N(\boldsymbol{\theta})$ 近似表示为参数向量的二次函数, 然后采用配方的方法得到使 $J_N(\boldsymbol{\theta})$ 达到极小的参数估计。估计算法是以递推形式给出的, 因此也称为改进增广最小二乘的递推算法。

假定在 N 次采样后得到参数估计 $\hat{\boldsymbol{\theta}}_N$, 将 $e_k(\boldsymbol{\theta})$ 在 $\hat{\boldsymbol{\theta}}_N$ 附近应用泰勒展开, 并取一阶近似得

$$e_k(\boldsymbol{\theta}) = \hat{e}(k) + \left(\frac{\partial \hat{e}(k)}{\partial \boldsymbol{\theta}} \right) (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N) \quad (4.19)$$

其中

$$\hat{e}(k) = e_k(\hat{\boldsymbol{\theta}}_N), \quad \frac{\partial \hat{e}(k)}{\partial \boldsymbol{\theta}} = \left. \frac{\partial e_k(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_N} \quad (4.20)$$

则 $\frac{\partial \hat{e}(k)}{\partial \boldsymbol{\theta}}$ 满足

$$\frac{\partial \hat{e}(k)}{\partial a_i} = [\hat{\mathbf{C}}(Z^{-1})]^{-1} y(k-i) \triangleq y_f(k-i) \quad (4.21)$$

$$\frac{\partial \hat{e}(k)}{\partial b_i} = -[\hat{\mathbf{C}}(Z^{-1})]^{-1} u(k-i) \triangleq -u_f(k-i) \quad (4.22)$$

$$\frac{\partial \hat{e}(k)}{\partial c_i} = -[\hat{\mathbf{C}}(Z^{-1})]^{-1} \hat{e}(k-i) \triangleq -\hat{e}_f(k-i) \quad (4.23)$$

$$i = 1, 2, \dots, n$$

这时

$$\begin{aligned} J_N(\boldsymbol{\theta}) &= \sum_{k=1}^N e_k^2(\boldsymbol{\theta}) \\ &\approx \sum_{k=1}^N [\hat{e}(k) + \frac{\partial \hat{e}(k)}{\partial \boldsymbol{\theta}}^T (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N)]^2 \\ &= (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N)^T \mathbf{P}_N^{-1} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N) + \eta_N \end{aligned} \quad (4.24)$$

其中, $\mathbf{P}_N^{-1} = \sum_{k=1}^N \frac{\partial \hat{e}(k)}{\partial \boldsymbol{\theta}} \left(\frac{\partial \hat{e}(k)}{\partial \boldsymbol{\theta}} \right)^T$ 为对称正定矩阵, η_N 为余项。

由式(4.19)和式(4.24)得

$$\begin{aligned} J_{N+1}(\boldsymbol{\theta}) &= \sum_{k=1}^{N+1} e_k^2(\boldsymbol{\theta}) \\ &= J_N(\boldsymbol{\theta}) + e_{N+1}^2(\boldsymbol{\theta}) \\ &\approx \Delta^T \mathbf{P}_N^{-1} \Delta + \eta_N + [\hat{e}(N+1) - \boldsymbol{\phi}_{N+1}^T \Delta]^2 \end{aligned}$$

$$= \Delta^T (\mathbf{P}_N^{-1} + \boldsymbol{\varphi}_{N+1} \boldsymbol{\varphi}_{N+1}^T) \Delta - 2\hat{e}(N+1) \boldsymbol{\varphi}_{N+1}^T \Delta + \hat{e}^2(N+1) + \eta_N \quad (4.25)$$

其中

$$\Delta = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N, \quad \boldsymbol{\varphi}_{N+1} = -\frac{\partial \hat{e}(N+1)}{\partial \boldsymbol{\theta}} \quad (4.26)$$

将式(4.25)配成完全平方得

$$J_{N+1}(\boldsymbol{\theta}) \approx (\Delta - \gamma_{N+1})^T \mathbf{P}_{N+1}^{-1} (\Delta - \gamma_{N+1}) + \eta_{N+1} \quad (4.27)$$

其中

$$\mathbf{P}_{N+1}^{-1} = \mathbf{P}_N^{-1} + \boldsymbol{\varphi}_{N+1} \boldsymbol{\varphi}_{N+1}^T \quad (4.28)$$

$$\boldsymbol{\gamma}_{N+1} = \mathbf{P}_{N+1} \boldsymbol{\varphi}_{N+1} \hat{e}_{(N+1)} \quad (4.29)$$

$$\eta_{N+1} = \eta_N - \boldsymbol{\gamma}_{N+1}^T \mathbf{P}_{N+1}^{-1} \boldsymbol{\gamma}_{N+1} + \hat{e}_{(N+1)}^2 \quad (4.30)$$

对式(4.28)运用矩阵求逆公式(见附录 A)

$$(\mathbf{A}^{-1} + \mathbf{H} \mathbf{R}^{-1} \mathbf{H}^T)^{-1} = \mathbf{A} - \mathbf{A} \mathbf{H} (\mathbf{H}^T \mathbf{A} \mathbf{H} + \mathbf{R})^{-1} \mathbf{H}^T \mathbf{A}$$

并把结果代入式(4.29)得

$$\mathbf{P}_{N+1} = \mathbf{P}_N - \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1} \boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + 1} \quad (4.31)$$

$$\boldsymbol{\gamma}_{N+1} = \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1}}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + 1} \hat{e}(N+1)$$

由式(4.27)可见,当 $\Delta = \boldsymbol{\gamma}_{N+1}$ 时,即取

$$\hat{\boldsymbol{\theta}}_{N+1} = \hat{\boldsymbol{\theta}}_N + \frac{\mathbf{P}_N \boldsymbol{\varphi}_{N+1}}{\boldsymbol{\varphi}_{N+1}^T \mathbf{P}_N \boldsymbol{\varphi}_{N+1} + 1} \hat{e}(N+1) \quad (4.32)$$

$J_{N+1}(\boldsymbol{\theta})$ 达到极小值。这里

$$\hat{e}(k) \approx y(k) - \hat{\boldsymbol{\varphi}}_k^T \hat{\boldsymbol{\theta}}_{k-1} \quad (4.33)$$

其中

$$\hat{\boldsymbol{\varphi}}_k^T = [-y(k-1), \dots, -y(k-n), u(k-1), \dots, u(k-n), \hat{e}(k-1), \dots, \hat{e}(k-n)]$$

$$\hat{\boldsymbol{\theta}}^T = [\hat{a}_1, \dots, \hat{a}_n, \hat{b}_1, \dots, \hat{b}_n, \hat{c}_1, \dots, \hat{c}_n]$$

最后,将式(4.21)~式(4.23)代入式(4.26)得

$$\boldsymbol{\varphi}_k^T = [-y_f(k-1), \dots, -y_f(k-n), u_f(k-1), \dots, u_f(k-n), \hat{e}_f(k-1), \dots, \hat{e}_f(k-n)] \quad (4.34)$$

容易证明(留做习题)关于 $\boldsymbol{\varphi}$ 有如下的递推关系

$$\boldsymbol{\varphi}_{k+1} = \begin{bmatrix} -\hat{C}_1 & -\hat{C}_2 & \cdots & -\hat{C}_n & & & & \\ 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & 0 & & & & \\ & & & -\hat{C}_1 & -\hat{C}_2 & \cdots & -\hat{C}_n & \\ & & & 1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & 0 & \\ & & & & & & -\hat{C}_1 & -\hat{C}_2 & \cdots & -\hat{C}_n \\ & & & & & & 1 & & & \\ & & & & & & & \ddots & & \\ & & & & & & & & 1 & 0 \end{bmatrix} \boldsymbol{\varphi}_k + \begin{bmatrix} -y(k) \\ 0 \\ \vdots \\ 0 \\ u(k) \\ 0 \\ \vdots \\ 0 \\ \hat{e}(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.35)$$

由式(4.31)~式(4.35)就构成了改进增广最小二乘的递推算法,具体步骤如下:

(1) 置初值: $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}, \mathbf{P}_0 = \alpha^2 \mathbf{I}, \alpha$ 为充分大正数, \mathbf{I} 为单位矩阵,

$$y(i) = y_f(i), u(i) = u_f(i), \hat{e}(i) = \hat{e}_f(i) = 0, i = 0, -1, \dots, -n+1;$$

构造 $\hat{\boldsymbol{\phi}}_1$ 和 $\boldsymbol{\phi}_1$;

(2) 置 $k=1$;

(3) 进行第 k 次采样得 $y(k), u(k)$;

(4) 计算

$$\begin{aligned}\hat{e}(k) &= y(k) - \hat{\boldsymbol{\phi}}_k^T \hat{\boldsymbol{\theta}}_{k-1} \\ \hat{\boldsymbol{\theta}}_k &= \hat{\boldsymbol{\theta}}_{k-1} + \frac{\mathbf{P}_{k-1} \boldsymbol{\phi}_k}{\boldsymbol{\phi}_k^T \mathbf{P}_{k-1} \boldsymbol{\phi}_k + 1} \hat{e}(k) \\ \mathbf{P}_k &= \mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T \mathbf{P}_{k-1}}{\boldsymbol{\phi}_k^T \mathbf{P}_{k-1} \boldsymbol{\phi}_k + 1}\end{aligned}$$

(5) 利用式(4.35),由 $\boldsymbol{\phi}_k, y(k), u(k), \hat{e}(k)$ 计算 $\boldsymbol{\phi}_{k+1}$ 。

(6) $k+1 \Rightarrow k$, 并转步骤(3)。

现有文献已经证明,改进的增广最小二乘法比增广最小二乘法有更好的收敛性,而且以概率 1 收敛到估计准则的一个局部极小值,这是目前所能期望的较好的一种方法。应该说明的是,性能指标式(4.17)可以基于最大似然函数的分析得到,因此上述递推算法也有人称其为近似最大似然递推算法。

4.3 最大似然法(ML)

最大似然法是对含有相关噪声数据得到无偏估计的另一种有效方法。

现在我们仍考虑模型式(4.15)。该式实际上表示噪声 $\{\varepsilon(k)\}$ 是相关的,且可表成如下形式

$$\varepsilon(k) = \sum_{i=1}^n c_i \varepsilon(k-i) + e(k) \quad (4.36)$$

其中, $\{e(i)\}$ 为零均值、正态白噪声序列,且与输入 $\{u(k)\}$ 无关,显然 $\varepsilon(n)$ 仍服从正态分布。

令

$$\begin{aligned}\mathbf{R} &= E(\boldsymbol{\varepsilon}_N \boldsymbol{\varepsilon}_N^T) \\ \boldsymbol{\varepsilon}_N^T &= [\varepsilon(n+1), \dots, \varepsilon(n+N)]\end{aligned}$$

这时,对数似然函数为

$$\log L(\mathbf{y}_N; \boldsymbol{\theta}) = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log(\det \mathbf{R}) - \frac{1}{2} (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta})^T \mathbf{R}^{-1} (\mathbf{y}_N - \boldsymbol{\Phi}_N \boldsymbol{\theta}) \quad (4.37)$$

值得注意的是,虽然式(4.37)与式(3.69)形式上很相似,但这里似然函数(通过 \mathbf{R}^{-1} 和 $\det \mathbf{R}$)依赖于 c_i ,实际上它对参数的依赖是非线性的,所以不能像式(3.72)那样得到显式解。只有通过迭代法(如牛顿法、梯度法,请参见第8章)求其数值解。

记

$$\boldsymbol{\theta}^T = (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n, c_1, c_2, \dots, c_n)$$

$$\boldsymbol{\phi}_{k-n}^T = [-y(k-1), \dots, -y(k-n), u(k-1), \dots, u(k-n), e(k-1), \dots, e(k-n)]$$

则具体计算步骤如下:

(1) 用经典的辨识方法(如最小二乘法)估计 a_i, b_i 的初值,同时给出 c_i 的初始估计,也就得到了 θ 的初始估计值 $\hat{\theta}$;置 $k=1$ 。

(2) 计算残差序列

$$e(k) = y(k) - \phi_{k-n}^T \hat{\theta}$$

(3) 由残差序列计算偏导数

$$\begin{aligned}\frac{\partial e(k)}{\partial a_i} &= y(k-i) + \sum_{j=1}^n c_j \frac{\partial e(k-j)}{\partial a_i} \\ \frac{\partial e(k)}{\partial b_i} &= -u(k-i) - \sum_{j=1}^n c_j \frac{\partial e(k-j)}{\partial b_i} \\ \frac{\partial e(k)}{\partial c_i} &= -e(k-i) - \sum_{j=1}^n c_j \frac{\partial e(k-j)}{\partial c_i}\end{aligned}\quad (4.38)$$

(4) 计算

$$\begin{aligned}\hat{\sigma}_k^2 &= \frac{1}{N} \sum_{j=1}^N e^2(j) \\ \sum_{j=1}^N e(j) \frac{\partial e(j)}{\partial \theta}\end{aligned}\quad (4.39)$$

$$\sum_{j=1}^N \left[\frac{\partial e(j)}{\partial \theta} \right] \left[\frac{\partial e(j)}{\partial \theta} \right]^T \quad (4.40)$$

(5) 计算参数向量的改变量

$$\delta_\theta = - \left[\sum_{j=1}^N \left(\frac{\partial e(j)}{\partial \theta} \right) \left(\frac{\partial e(j)}{\partial \theta} \right)^T + \sum_{j=1}^N e(j) \frac{\partial^2 e(j)}{\partial \theta^T \partial \theta} \right]^{-1} \sum_{j=1}^N e(j) \frac{\partial e(j)}{\partial \theta} \quad (4.41)$$

(6) 确定新的参数向量

$$\hat{\theta}_{k+1} = \theta + \delta_\theta \quad (4.42)$$

(7) 若 $\frac{\sigma_{k+1}^2 - \sigma_k^2}{\sigma_k^2} < 10^{-4}$ 则停止;否则,置 $k=k+1$,转步骤(2)。

步骤(7)意味着如果残差平方和的减少不到 0.01% 时计算结束,此准则是基于残差的统计检验而定的(Astrom and Eykhoff, 1977)。

上述方法在实际中得到成功的应用,尤其在噪声水平相当高的情况下,也得到了满意结果,不足之处是不能在线使用。正如 4.2 节最后指出的,从似然函数出发,与 4.2 节完全类似地推导,同样可导出所谓改进增广最小二乘递推算法,只是名称改为递推近似最大似然法,从而弥补最大似然法不能在线应用的这一缺点。

【例 4.1】 考虑系统模型为

$$(1 - 1.5Z^{-1} + 0.7Z^{-2})y(k) = (Z^{-1} + 0.5Z^{-2})u(k) + \frac{1}{1 - Z^{-1} + 0.2Z^{-2}}e(k)$$

式中 $\{u(k)\}, \{e(k)\}$ 均为零均值正态同分布不相关序列, $\{u(k)\}$ 的标准差 σ 取为 1, $\{e(k)\}$ 的标准差 σ 分别取 0.4, 1.8, 7.2, 在各种情况下,分别用递推增广最小二乘和递推最大似然法进行参数估计,对于样本长度 $N=500$,估计结果在表 4.1 中列出。结果表明,在噪声水平较高时,递推最大似然估计似乎好一些。

【例 4.2】 考虑系统模型为

$$(1 - 1.5Z^{-1} + 0.7Z^{-2})y(k) = (Z^{-1} + 0.5Z^{-2})u(k) + (1 - Z^{-1} + 0.2Z^{-2})\lambda e(k)$$

$\{u(k)\}, \{e(k)\}$ 的性质同上例,取 $\lambda=1, \sigma=0.4, N=500$ 用递推最大似然法、递推增广最小二乘法及离散最大似然法 3 种方法,参数估计结果在表 4.2 中列出。结果表明,递推最大似然估计性能居中。

表 4.1 例 4.1 辨识结果

参数	真值	$\sigma=0.4$		$\sigma=1.8$		$\sigma=7.2$	
		RELS	RML	RELS	RML	RELS	RML
a_1	-1.5	-1.500	-1.517	-1.025	-1.761	-1.010	-1.831
a_2	0.7	0.677	0.702	0.269	0.876	0.308	0.932
b_1	1.0	1.026	1.040	0.073	1.199	-0.910	1.551
b_2	0.5	0.507	0.510	0.742	0.432	-0.407	0.294
c_1	-1	-1.024	-0.948	-0.511	-0.716	-0.700	-0.661
c_2	0.2	0.253	0.167	-0.174	0.238	0.038	0.287
σ		/	0.428	/	1.982		8.014

表 4.2 例 4.2 辨识结果

参数	真值	RELS	RML	ML
a_1	-1.5	-1.489	-1.498	-1.500
a_2	0.7	0.691	0.699	0.700
b_1	1.0	1.036	1.091	1.020
b_2	0.5	0.468	0.495	0.470
c_1	-0.1	-0.896	-0.901	-1.010
c_2	0.20	0.716	0.212	0.200
σ	0.4			0.400

4.4 闭环系统的辨识

4.4.1 问题的提出

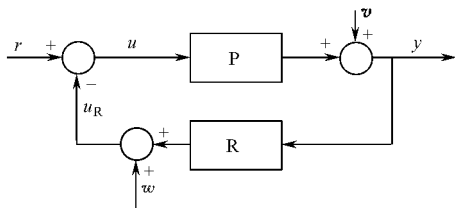
前面所讲的辨识方法,对于在开环(即系统没有反馈)情况下取得的数据,都能得到良好的辨识结果。但是有许多实际系统是不允许开环试验的,因为开环可能使系统不稳定,甚至有危险(例如一些化工系统就是如此);另外,一些系统反馈是本身固有的,不能拆除(如生物系统,经济系统等);还有一些系统是必须在闭环条件下在线辨识的(如自校正控制系统)。那么,自然会提出:在闭环条件下,前面所介绍的辨识方法还灵不灵? 闭环条件下的辨识有什么特点? 这是大家所关心的问题。

为了回答上述问题,我们先分析一下闭环情况(如图 4.1 所示)与开环情况所取得的数据有什么不同。首先,正常工作着的闭环系统是稳定的,对于反馈产生的不同输入可能获得几乎相同的输出,因此,闭环系统的输入、输出数据所提供的信息比开环的少。更重要的是,在闭环操作下,输入不再与输出及其所受的干扰独立,而是与输出噪声相关,这就给辨识造成了更大的困难,以至于产生参数不可辨识的情况,从下面的简单例子可以清楚地看到这一点。

【例 4.3】 考虑一阶模型

$$y(k) + ay(k-1) = bu(k-1) + e(k) \quad (4.43)$$

其中, $\{e(k)\}$ 为零均值、方差为 σ^2 的白噪声序列。



P—装置; r —外部输入; R—调节器; v —正向通道噪声; w —反馈通道噪声

图 4.1 闭环系统示意图

采用线性调节器

$$u(k) = cy(k) \quad (4.44)$$

估计参数 a, b 的性能指标采用最小二乘准则

$$J(a, b) = \sum_{k=1}^N [y(k+1) + ay(k) - bu(k)]^2 \quad (4.45)$$

由式(4.44), 对任意实数 λ , 均有

$$\lambda[-u(k) + cy(k)] = 0 \quad (4.46)$$

所以有

$$\begin{aligned} J(a, b) &= \sum_{k=1}^N [y(k+1) + ay(k) - bu(k) + \lambda(-u(k) + cy(k))]^2 \\ &= \sum_{k=1}^N [y(k+1) + (a + \lambda c)y(k) - (b + \lambda)u(k)]^2 \\ &= J(a + \lambda c, b + \lambda) \end{aligned} \quad (4.47)$$

式(4.47)意味着若某 a_0, b_0 使性能指标式(4.45)达到极小, 则对任意 $\lambda, a_0 + \lambda c$ 和 $b_0 + \lambda$ 也使式(4.45)达到极小, 或者说有无穷多个参数使性能指标式(4.45)达到极小。换句话说, 使性能指标式(4.45)达到极小的参数不惟一。在这种意义上讲, 参数是不可辨识的。

4.4.2 可辨识性

从上面给出的例子中可以看出, 对一个系统, 并不是随意使用一组数据都能将参数惟一地估计出来, 即将系统的数学模型辨识出来, 这样就产生了系统是否可辨识的问题。所谓不可辨识有两种情形, 一种情形就像在例 4.3 指出的那样, 既使对式(4.43)所描述的这样一个非常简单的系统, 由于数据是在(由式(4.44)构成)有反馈的条件下取得的, 所以就不能得到惟一的参数估计值。换句话说, 这种辨识的不惟一性是由于给定的实验条件(从而决定了取得数据的途径)造成的, 这种情况在实际中是经常遇到的, 这种由于检测手段及其他实验条件的限制而得不到必要的数据, 从而产生的系统不可辨识的问题, 称为先验不可辨识问题。因为这时造成系统不可辨识的原因并不是由于数据有误差, 而是由于系统及输入、输出的结构造成的。另一种情形, 由于测量数据有误差而造成估计的不精确, 利用有限的数数据不可能得到精确的估计值, 因此就要考虑极限情形, 也就是说当数据不断增加, 估计值是否收敛到真正的参数值。这当然与系统的性质、数据(实验条件)和所用的辨识方法有关, 由于这些原因而产生的不可辨识的问

题称为后验不可辨识问题。

当然,这两种可辨识性的讨论都是重要的,尤其是先验可辨识性。因为先验不可辨识的问题是根本不能解的,需要进一步研究模型的结构或考虑改进实验条件,而这一点在一些实际的辨识问题中是不容易解决的。对于后验不可辨识的问题,或者仍然能得到一个模型,或者通过改进方法得到一个模型,这样的模型有时还是可以用的。这里不对一般可辨识性的问题进行讨论,有兴趣的读者可参看 Soderstrom(1975)等人的文献。这里主要讨论在控制工程和社会经济、生物系统等领域最常见的一种情况,即在闭环条件下,离散线性系统的辨识问题。Goodwin 等(1977)曾对闭环条件下可辨识性问题进行过理论探讨,并给出了可辨识的充分条件,简单地说可以概括为两个:一个充分条件是当反馈调节器的阶次大于或等于正向系统的阶次时,闭环系统是可辨识的。这个条件实际上是不太容易满足的,因为实际上用的调节器阶次一般不可能太高。为了解决这个矛盾,他又提出另一个充分条件,即通过几个调节器互相切换,取得闭环系统数据,当所有调节器的传递数矩阵与单位阵构成的矩阵秩等于输出和输入维数之和时,则可以辨识。在一些实际系统中,通过切换调节器取得闭环系统数据,进而实现闭环系统的辨识,一般是容易实现的。对于在闭环条件下采用最小二乘法进行辨识的情况,Wellstead 和 Edmunds(1975)给出了更具体、更易判断的条件。下面将介绍他们的结果。

4.4.3 闭环条件下的最小二乘估计

从例 4.3 不难看出,为了使参数可辨识,最方便的办法就是将其中某个参数事先取定,不参加辨识。上例中, a 和 b 取定任何一个,则另一个显然惟一确定。这一思想在校正控制系统中得到应用。

应该指出,例 4.3 只是说明在控制律为式(4.44)的情况下,闭环系统的参数是不可辨识的。如果将反馈控制律改为

$$u(k) = cy(k-1) \quad (4.48)$$

或

$$u(k) = c_1 y(k-1) + c_2 y(k) \quad (4.49)$$

等,则都可得到惟一解。这说明在闭环操作下并非总是不可辨识的,只要满足一定的条件,闭环系统仍可辨识,这为闭环系统的辨识指明了光明前景。下面介绍由 W-E 给出的闭环条件下,可采用最小二乘辨识的充分条件(Wellstead P E and Edmunds,1975)。

考虑单输入、单输出系统,其正向通道数学模型为

$$y(k) = \sum_{i=1}^{n_a} a_i y(k-i) = \sum_{j=q}^{n_b} b_j u(k-j) + \epsilon(k) \quad (4.50)$$

其中, $y(k)$ 为输出, $u(k)$ 为输入。

反馈调节器模型为

$$u(k) = \sum_{i=1}^{n_d} d_i u(k-i) + \sum_{j=p}^{n_c} c_j y(k-j) + e(k) \quad (4.51)$$

其中 $\{\epsilon(k)\}, \{e(k)\}$ 为互不相关的零均值正态白噪声序列。对此闭环系统 W-E 指出:如果①模型式(4.50)的阶已知;②反馈调节器式(4.51)是已知的(或) $\epsilon(k)$ 不存在;③反馈通道噪声存在 $e(k)$,则可直接采用最小二乘法辨识系统模型式(4.50)的参数。

方程式(4.50)可以写成如下形式

$$y(k) = \boldsymbol{\varphi}_k^T \boldsymbol{\theta} + \epsilon(k) \quad (4.52)$$

其中

$$\boldsymbol{\varphi}_k^T = [-y(k-1), \dots, -y(k-n_a), u(k-q), \dots, u(k-n_b)] \quad (4.53)$$

$$\boldsymbol{\theta}^T = (a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}) \quad (4.54)$$

当取得 $(n+N)$ 次采样后(这里 $n=\max\{n_a, n_b\}$; $N \geq n_a + n_b - q + 1$),可得

$$\mathbf{y}_N = \boldsymbol{\Phi}_N \boldsymbol{\theta} + \boldsymbol{\varepsilon}_N$$

其中

$$\mathbf{y}_N^T = (y(n+1), \dots, y(n+N))$$

$$\boldsymbol{\Phi}_N^T = (\boldsymbol{\varphi}_{n+1}, \dots, \boldsymbol{\varphi}_{n+N})$$

$$\boldsymbol{\varepsilon}_N^T = (\varepsilon(n+1), \dots, \varepsilon(n+N))$$

如果 $(\boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N)^{-1}$ 存在,则模型式(4.50)的最小二乘解为

$$\hat{\boldsymbol{\theta}}_N = (\boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N)^{-1} \boldsymbol{\Phi}_N^T \mathbf{y}_N \quad (4.55)$$

文献对上述估计的惟一性和收敛性问题做了进一步讨论,这里将其思路及一些结果做一简单介绍。

如果 $(\boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N)^{-1}$ 存在,则估计误差为(见 2.1 节中的讨论)

$$\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N = (\boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N)^{-1} \boldsymbol{\Phi}_N^T \boldsymbol{\varepsilon}_N \quad (4.56)$$

由大数定律得

$$P\lim(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_N) = \left[P\lim\left(\frac{1}{N} \boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N\right)^{-1} \right] \left[P\lim\left(\frac{1}{N} \boldsymbol{\Phi}_N^T \boldsymbol{\varepsilon}_N\right) \right]$$

其中符号 $P\lim(\quad)$ 表示以概率收敛。

由此可见,相容性要求

$$P\lim\left[\frac{1}{N} \boldsymbol{\Phi}_N^T \boldsymbol{\varepsilon}_N\right] = 0$$

如果 $\varepsilon(k)$ 不存在,此条件显然满足,如果 $\varepsilon(k)$ 存在,可以证明:如果反馈调节器已知,只要有一个回路有滞后,或者说,回路不存在瞬时传递通道(即 $p > 0$ 或 $q > 0$),并且系统是闭环稳定的, $\frac{1}{N}(\boldsymbol{\Phi}_N^T \boldsymbol{\varepsilon}_N)$ 的均值和方差都趋于零(当 $N \rightarrow \infty$ 时),从而最小二乘估计是相容的。

惟一性取决于 $\frac{1}{N} \boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N$ 的极限。可以证明: $e(k)$ 存在时, $\frac{1}{N} \boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N$ 一定是非奇异的。若 $e(k)$ 不存在,但只要反馈调节器多项式的阶满足

$$\max\{n_d - n_b + p, n_c - n_a + q\} > 0 \quad (4.57)$$

则最小二乘估计仍是惟一的。

现在利用上述条件再分析一下例 4.3。例 4.3 中所用的反馈律无噪声,即 $e(k)$ 不存在。对于反馈律式(4.44), $n_a=1, n_b=1, q=1, n_d=0, n_c=0, p=0$, 它不满足式(4.57), 所以不满足惟一性的充分条件。而对于式(4.48)和式(4.49), 它们都满足式(4.57), 所以可得到惟一的参数估计。下面的仿真例子也说明了上述结果的正确性。

【例 4.4】 仿真系统的模型为

$$y(k) + 1.4y(k-1) + 0.45y(k-2) = u(k-1) + 0.7u(k-2) + e(k) \quad (4.58)$$

其中 $\{e(k)\}$ 是零均值、单位方差的正态白噪声序列。

在不同的闭环条件下,用递推最小二乘辨识结果(递推 2000 步),见表 4.3。

从上面的仿真结果可以看出:对于第一种情况,由于具有非零的反馈噪声,所以参数估计是惟一的,又因为它有一个回路有单位滞后(式(4.58)中 $q > 0$),因此参数估计也是相容的。

对于第二种情况,虽然没有反馈噪声,但调节器的阶满足式(4.57),因此系统的参数仍然是可辨识的,但是结果比第一种情况差一些。对于第三种情况,调节器的阶虽然不满足式(4.57),但它具有非零的反馈噪声干扰,因此系统仍是可辨识的。对于第四种情况,调节器既不满足式(4.57),又不存在反馈噪声干扰,因此不满足可辨识条件。

表 4.3 例 4.3 辨识结果

反馈调节器	真 值	a_1	a_2	a_3	a_4
	估计值	1.4	0.45	1.0	0.7
$u(k)=0.33y(k)+0.033y(k-1)-0.4y(k-2)$ 有非零反馈干扰		1.398	0.448	1.003	0.696
$u(k)=0.33y(k)+0.033y(k-1)-0.4y(k-2)$ 无反馈干扰		1.371	0.420	1.004	0.765
$u(k)=y(k)+0.2y(k-1)$ 有非零反馈干扰		1.390	0.443	1.020	0.714
$u(k)=y(k)+0.2y(k-1)$ 无反馈干扰		0.308	0.216	-0.027	-0.333

从上面的简单分析可以看出,对于实际系统,闭环可辨识的条件并不是很难满足的。

习 题

1. 考虑模型

$$y(k) = ay(k-1) + bu(k) + \varepsilon(k)$$

其中, $|a| < 1$, $\{\varepsilon(k)\}$ 和 $\{u(k)\}$ 是零均值的广义平稳过程, (四价矩有界), $\{u(k)\}$ 与 $\{\varepsilon(k)\}$ 独立, 证明:

取

$$Z_N = \begin{pmatrix} u(1) & u(2) \\ u(2) & u(3) \\ \vdots & \vdots \\ u(N) & u(N+1) \end{pmatrix}$$

为辅助变量阵, 则

$$\begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = (Z_N^T \Phi_N)^{-1} Z_N^T y_N \xrightarrow{p} \begin{pmatrix} a \\ b \end{pmatrix}$$

其中 Φ_N, y_N 由式(3.10)定义。

- 证明式(4.21)~式(4.23)。
- 证明式(4.35)。
- 证明式(4.38)。
- 考虑带有反馈控制 $u(k)=gu(k)$ 的一阶系统

$$y(k) = -ay(k-1) + bu(k-1) + \varepsilon(k)$$

其中, $\varepsilon(k)$ 为白噪声, 说明当 a, b 中之一固定时, 则 a, b 是可辨识的。

- 验证式(4.48)、式(4.49)是否满足式(4.57)。
- 编程实现例 4.4, 并验证其结果。

第 5 章 模型阶的辨识

前面几章所给出的方法都是在假定线性模型的阶已知的条件下讨论的。但实际上,在多数情况下系统的阶次是不知道的,而阶次正确与否是能否正确描述系统的一个关键问题。因此,模型阶次的辨识是系统辨识的重要组成部分。模型阶次的辨识是一个比较麻烦的问题,目前虽有不少辨识阶次的方法,但是要准确地确定阶次还需要经过反复试验,最终通过实践来确认。

目前,确定模型的阶次主要有两种途径:一种是通过系统本身运动的机理分析及实验来确定;另一种是通过系统的输入、输出数据用辨识的方法来确定。本章仅对单变量线性系统介绍几种常用的辨识阶次的简单方法。至于多变量线性系统结构的辨识,由于其结构描述的特殊性,其辨识方法将在第 6 章专门介绍。

5.1 单变量线性系统阶的辨识

5.1.1 损失函数检验法

考虑单输入、单输出线性系统

$$\begin{aligned} y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) \\ = b_1 u(k-1) + \cdots + b_n u(k-n) + e(k) \end{aligned} \quad (5.1)$$

损失函数为残差平方和

$$J(n) = \sum_{k=n+1}^{n+N} e^2(k)$$

显然,对不同阶次 n 的模型, $J(n)$ 是不同的。当阶次越接近于真实阶次 n_0 时, $J(n)$ 就应该越小,而 n 超过真实阶次 n_0 时, $J(n)$ 应该接近于常值(或因计算误差而使 $J(n)$ 略大)。基于这种考虑,可给出如下一个粗略的估计阶次的方法。

对于 $n=1, 2, \cdots$, 逐次估计参数并计算损失函数 $J(n)$ 的值。开始 $J(n)$ 应逐渐下降, 当对某个 n_0 , $J(n_0+1)$ 和 $J(n_0)$ 没有明显的不同时, 就取 n_0 作为阶的估计值, 即 $\hat{n} = n_0$ 。实际上可以用作图法来确定, 即画出损失函数依赖于阶次 n 的曲线图, 然后由图确定对阶次的估计。下面是对一个热交电器确定阶次的实际例子, 计算得到的损失函数曲线如图 5.1 所示。

由图 5.1 给出的结果可以看出, $\hat{n}=3$ 。

5.1.2 F 检验法

上面给出的方法虽比较简单, 但比较粗糙。因为在许多实际问题中, 由于所讨论问题的损失函数的值与其数量级有很大关系, 很难给出一个相差多大才算有显著差异的一般标准, 因此单从损失函数的绝对数值上往往很难判定它们之间是否有显著的差异, 通常需要借助于统计检验的方法来判别。这里介绍一种最常用的判别方法: F 检验法。

对于阶次为 n 的模型(5.1), 共含 $2n$ 个参数, 即 $a_1, \cdots, a_n; b_1, \cdots, b_n$ 。设 $n_1 < n_2$ 是模型的

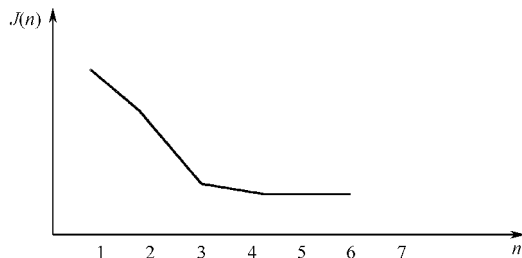


图 5.1 热交电器的损失函数曲线

两个不同阶次。当阶次从 n_1 增加到 n_2 时,损失函数从 $J(n_1)$ 降至 $J(n_2)$ 。现在的问题是说明 $J(n)$ 的变化是否显著?如果显著,就说明 n_2 比 n_1 更接近真实阶次 n_0 ;如果不显著,则说明 n_1 已经接近真实阶次 n_0 。为了检验损失函数是否有显著差异,我们构造统计量 F

$$F = \frac{J(n_1) - J(n_2)}{J(n_2)} \cdot \frac{N - 2n_2}{2(n_2 - n_1)} \quad (5.2)$$

其中 N 为样本长度, $2n_1, 2n_2$ 分别为模型阶次是 n_1, n_2 时参数的数目。可以证明(Åström K J, 1968);当 $n_2 > n_1 \geq n_0$ 时,对于充分大的 N ,统计量 F 渐近服从于自由度为 $(2(n_2 - n_1), N - 2n_2)$ 的 F 分布: $F(2(n_2 - n_1), N - 2n_2)$ 。这时,确定阶的问题可以化为如下的假设检验问题:

$$H_0: n_2 > n_1 \geq n_0$$

利用统计量 F 来检验 H_0 。

给定置信度 α ,查 F 分布表得 F_α (否定域临界值)。若 $F < F_\alpha$,则在置信度 α 下接受原假设 H_0 ,说明 $J(n_1)$ 与 $J(n_2)$ 无显著差异,即 n_1 可确定为系统的阶;若 $F \geq F_\alpha$ (拒绝原假设 H_0),则说明 $J(n_1)$ 和 $J(n_2)$ 有显著差异,即 n_2 更接近于真实阶。

利用上述结果,可以用逐次递增阶的方法(即 $n_1 = 1, 2, \dots; n_2 = n_1 + 1$),通过 F 检验来确定较好的阶次,每增加一次,就进行一次 F 检验,直到 $J(n_1)$ 和 $J(n_2)$ 无显著差别为止。这时 $n_2 - n_1 = 1$,从而统计量 F 可简化为

$$F = \frac{J(n_1) - J(n_2)}{J(n_2)} \cdot \frac{N - 2n^2}{2} \quad (5.3)$$

下面通过一个数值例子来说明其用法。

取置信度 $\alpha = 5\%$,查表得 $F_{0.05}(2, 100) = 3.09$ 。根据 100 组输入、输出数据,用最小二乘法分别对阶次为 1, 2, $\dots, 7$ 的模型进行辨识,辨识的结果及损失函数 $J(n)$ 和统计量 F 的值都列在表 5.1 中。

表 5.1 F 检验法辨识结果

n	a	b	$J(n)$	F
1	-0.643	1.018	592.65	
2	-1.015 0.377	1.086 -0.520	469.64	12.57
3	-1.118 0.624 -0.178	1.115 -0.660 0.263	389.50	9.67

n	a	b	$J(n)$	F
4	-1.157	1.085	323.24	9.43
	0.756	-0.733		
	-0.412	0.409		
	0.187	-0.146		
5	-1.185	1.080	299.85	3.51
	0.814	0.745		
	-0.518	0.475		
	0.349	-0.252		
6	-0.117	0.123	293.25	0.99
	-1.195	1.079		
	0.339	-0.751		
	-0.555	0.487		
7	0.410	-0.290	287.31	0.89
	-0.208	0.183		
	-0.061	-0.080		

从表 5.1 可以看出：

(1) 当模型的阶由 1 增加到 7 时, $J(n)$ 的值逐次下降；

(2) 当模型的阶由 4 增加到 5 时, $F=3.51>3.09$ 。因此, $J(n)$ 的减少是显著的。而当阶次由 5 增加到 6 时, 统计量 F 的值为 $0.99(<3.09)$, $J(n)$ 的减少不显著。因此模型阶应确定为 5。

5.1.3 赤池信息准则(AIC 准则)

日本学者赤池(Akaike, 1974)从极大似然法的信息论解释出发, 给出了一个用基本信息量定阶准则。即

$$\text{AIC}(p) = -2\log L(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta}) + 2p \quad (5.4)$$

其中, $\log L(\cdot)$ 表示对数似然函数, $\boldsymbol{\theta}$ 是参数向量的极大似然估计, \mathbf{u}, \mathbf{y} 分别表示输入和输出向量, p 是模型中可独立调整的参数个数。

使 $\text{AIC}(p)$ 达到最小的阶次作为模型的最佳阶次。下面对此准则给以粗略直观的解释。

设系统的真实的阶为 n_0 , 对应的独立可调参数个数为 p_0 。我们从 $n=1$ 开始计算 $\text{AIC}(p)$ 的值, 开始时, p 的值小, $-2\log L(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta})$ 的数值大, 在 $\text{AIC}(p)$ 中, $-2\log L(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta})$ 起主要作用。随着 n 的增大接近于 n_0 , $-2\log L(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta})$ 的值逐渐变小, 而 $2p$ 的值随着 p 的增大而增大, 从而在 $\text{AIC}(p)$ 中起主导作用。因此, $\text{AIC}(p)$ 在 p_0 点形成一个最小值, 图 5.2 给出一个直观解释。

如果我们考虑模型仍为式(5.1), 并假定 $\{e(k)\}$ 是零均值、方差为 σ^2 的独立正态随机序列, 且与输入 \mathbf{u} 无关。这时似然函数可表示为

$$L(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta}) = (2\pi\sigma^2)^{-N/2} \cdot \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})^T(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})\right\} \quad (5.5)$$

其中, $\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta}$ 和 N 如同第 4 章所定义。这时

$$-2\log L(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta}) = N\log(2\pi) + N\log\sigma^2 + \frac{1}{\sigma^2}(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})^T(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})$$

如果 σ^2 已知, 则 AIC 准则等价于

$$\text{AIC}(p) = \frac{1}{\sigma^2}(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})^T(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta}) + 2p \quad (5.6)$$

或者

$$\text{AIC}(p) = (\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})^T(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta}) + 2p\sigma^2 \quad (5.7)$$

当 $e(k)$ 为正态分布时, $\boldsymbol{\theta}$ 的极大似然估计和最小二乘估计是一致的。

$(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta})^T(\mathbf{y}_N - \boldsymbol{\Phi}_N\boldsymbol{\theta}) \triangleq J_N(\boldsymbol{\theta})$ 为模型残差平方和。再用 σ^2 的极大似然估计 $\hat{\sigma}^2 = J_N(\boldsymbol{\theta})/N$ 代替 σ^2 , 这时式(5.7)变为

$$\text{AIC}(p) = J_N(\boldsymbol{\theta}) + 2pJ_N(\boldsymbol{\theta})/N$$

或

$$\text{AIC}(p) = (1 + 2p/N)J_N(\boldsymbol{\theta}) \quad (5.8)$$

因此, 当用最小二乘估计模型参数时, 可通过使式(5.8)的值达到最小来确定线性系统模型的阶。

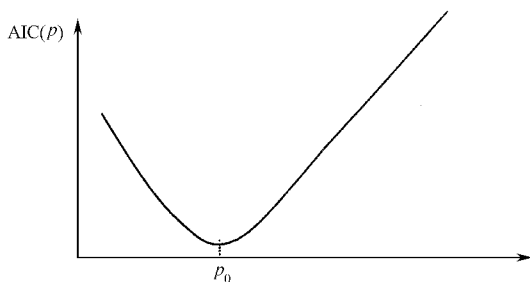


图 5.2 AIC(p)示意图

5.2 阶与参数同时辨识的递推算法

前面介绍的阶的辨识方法, 都是将阶的确定和参数辨识分开进行的。即先指定一个阶次, 再进行参数辨识, 然后再计算判断阶的准则, 逐次进行, 但每增加一次阶都需重新计算, 极大地增加了计算量。本节所介绍的方法是在辨识模型阶的过程中, 同时得到相应的参数估计, 而且低阶计算时所得的结果, 都可直接用来计算更高阶情况, 极大地简化了计算过程, 且无多余和重复的运算, 是一种模型阶次和参数估计同时确定的递推算法(王秀峰、卢桂章, 1981)。

5.2.1 辨识阶次的基本思想和方法

考虑模型

$$y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_1 u(k-1) + \cdots + b_n u(k-n) \quad (5.9)$$

对于 $k = n+1, n+2, \dots, n+N$ 得到

$$\begin{cases} y(n+1) = a_1 y(n) + \cdots + a_n y(1) + b_1 u(n) + \cdots + b_n u(1) \\ y(n+2) = a_1 y(n+1) + \cdots + a_n y(2) + b_1 u(n+1) + \cdots + b_n u(2) \\ \vdots \\ y(n+N) = a_1 y(n+N-1) + \cdots + a_n y(N) + b_1 u(n+N-1) + \cdots + b_n u(N) \end{cases} \quad (5.10)$$

若令

$$\begin{aligned} y_i &= [y(i), y(i+1), \dots, y(i+N-1)]^T \\ u_i &= [u(i), u(i+1), \dots, u(i+N-1)]^T \\ \theta &= [a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n]^T \end{aligned} \quad (5.11)$$

则式(5.9)就意味着有

$$\begin{aligned} y_{n+1} &= a_1 y_n + \dots + a_n y_1 + b_1 u_n + \dots + b_n u_1 \\ &= [y_n, \dots, y_1, u_n, \dots, u_1] \theta \end{aligned} \quad (5.12)$$

式(5.12)表明,如果系统的阶为 n , 则 y_{n+1} 可表示成 $y_n, \dots, y_1, u_n, \dots, u_1$ 的线性组合,其组合系数就是系统模型的参数,这就是本节将给出的辨识系统阶次方法的基础。

基于上述的分析,为了描述清楚,我们把取得的数据首先排成如下矩阵

$$\begin{aligned} H &= \begin{bmatrix} y(1) & y(2) & \cdots & \vdots & u(1) & u(2) & \cdots \\ y(2) & y(3) & \cdots & \vdots & u(2) & u(3) & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \\ y(N) & y(N+1) & \cdots & \vdots & u(N) & u(N+1) & \cdots \end{bmatrix} \\ &= [y_1 y_2 \cdots \vdots u_1 u_2 \cdots] = [Y : U] \end{aligned} \quad (5.13)$$

然后按如下顺序选择线性无关向量

$$y_1, u_1, y_2, u_2, y_3, u_3, \dots \quad (5.14)$$

当选到某个向量 y_{n_0} 与前面所选出的向量线性相关时,则得到模型的阶次 $n = n_0 - 1$ 。注意,为了保证线性无关向量的充分选择,向量长度 N 必须充分大(至少大于模型所包含参数的个数,模型式(5.12)中所包含参数的个数是 $2n$)。

通过上面的讨论,将确定阶的问题化为向量相关性的判别问题,而实际中判断相关性并无可实施的手段。所以下面将相关性判别问题化为容易计算的判别矩阵是否满秩的问题。

记 $\Phi(k, l)$ 为按式(5.14)给出的顺序选出的向量,且按选出的先后次序排列所构成的矩阵,其中 k, l 分别表示在矩阵 Φ 中包含了矩阵 Y 的前 k 个和矩阵 U 的前 l 个向量。例如

$$\begin{aligned} \Phi(1, 0) &= (y_1) \\ \Phi(1, 1) &= (y_1 u_1) \\ \Phi(2, 1) &= (y_1 u_1 y_2) \\ &\vdots \end{aligned}$$

这样,确定阶的问题就化为判断矩阵系列 $\Phi(1, 1), \Phi(2, 1), \dots$, 是否满秩的问题。但是,我们注意到, Φ 一般并不是方阵,这给判别带来了困难。但可利用下面众所周知的事实来克服这一困难。事实是:

对任意矩阵 D

$$\text{rank}(D) = \text{rank}(D^T D) \quad (5.15)$$

令 $S(k, l) = \Phi^T(k, l) \Phi(k, l)$, 则确定阶次的问题就变为逐个判断方阵序列 $\{S(k, l)\}$ 是否满秩的问题。即依次判断 $S(1, 1), S(2, 1), S(3, 2), \dots$, 是否满秩。当发现 $S(n, n)$ 满秩, 而 $S(n+1, n)$ 不满秩时,则可断言 n 为系统的阶。这时可接着进行参数估计

$$\theta = S^{-1}(n, n) \Phi^T(n, n) y_{n+1} \quad (5.16)$$

这里

$$\theta^T = (a_n, b_n, a_{n-1}, b_{n-1}, \dots, a_1, b_1) \quad (5.17)$$

用上述方法确定阶时,对于一个二阶系统需进行 4 次判断,对于 3 阶系统需进行 6 次判

断。每次判断实际上是求其行列式的值,且每次判断彼此没有联系,这需要大量的、重复的计算,计算量依然很大。就计算量而言,这与 5.1 节所介绍的辨识阶的方法没有多大区别,但这种新的思路为给出递推算法奠定了基础。下面给出定阶的递推方法。

5.2.2 阶的递推辨识算法

为了得到递推算法,引入如下引理和定理。

引理 5.1 设矩阵

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{matrix} \} n_1 \\ \} n_2 \end{matrix}, \text{ 且 } \det \mathbf{A}_{11} \neq 0 \quad (5.18)$$

则

$$(1) \mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \mathbf{P}^{-1} \mathbf{A}_{21} \mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1} \mathbf{A}_{12} \mathbf{P}^{-1} \\ -\mathbf{P}^{-1} \mathbf{A}_{21} \mathbf{A}_{11}^{-1} & \mathbf{P}^{-1} \end{bmatrix} \quad (5.19)$$

$$(2) \det \mathbf{A} = \det \mathbf{A}_{11} \cdot \det \mathbf{P}$$

其中, $\mathbf{P} = \mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12}$ 。

证明 见附录 A。

定理 5.1 设 x_1, x_2, \dots, x_n, y 是同维向量, 记

$$\Phi_n = (x_1, x_2, \dots, x_n)$$

$$\Phi_{n+1} = (x_1, x_2, \dots, x_n, y) = (\Phi_n, y)$$

$$S_n = \Phi_n^T \Phi_n, S_{n+1} = \Phi_{n+1}^T \Phi_{n+1}$$

若 S_n 和 S_{n+1} 皆可逆, 则

$$S_{n+1}^{-1} = \begin{bmatrix} S_n & \Phi_n^T y \\ y^T \Phi_n & y^T y \end{bmatrix}^{-1} = \begin{bmatrix} S_n^{-1} + \theta \theta^T / p & -\theta / p \\ -\theta^T / p & 1/p \end{bmatrix} \quad (5.20)$$

其中, θ 是在最小二乘意义下 y 由 x_1, x_2, \dots, x_n 线性表示的组合系数, p 是残差平方和。

证明 由引理 5.1 得

$$S_{n+1}^{-1} = \begin{bmatrix} S_n^{-1} + S_n^{-1} \Phi_n^T y P^{-1} y^T \Phi_n S_n^{-1} & -S_n^{-1} \Phi_n^T y P^{-1} \\ -P^{-1} y^T \Phi_n S_n^{-1} & P^{-1} \end{bmatrix} \quad (5.21)$$

这里

$$P = y^T y - y^T \Phi_n S_n^{-1} \Phi_n^T y \quad (5.22)$$

令

$$\theta = S_n^{-1} \Phi_n^T y \quad (5.23)$$

注意到 Φ_n 和 S_n 的定义, θ 即是方程 $\Phi_n \theta = y$ 的最小二乘解, 也就是由 x_1, x_2, \dots, x_n 表示 y 的线性组合系数。由于

$$(I - \Phi_n S_n^{-1} \Phi_n^T)^T (I - \Phi_n S_n^{-1} \Phi_n^T) = I - \Phi_n S_n^{-1} \Phi_n^T$$

而

$$\begin{aligned} (I - \Phi_n S_n^{-1} \Phi_n^T) y &= y - \Phi_n S_n^{-1} \Phi_n^T y \\ &= y - \Phi_n \theta \\ &\triangleq e \end{aligned} \quad (5.24)$$

故有

$$p = y^T y - y^T \Phi_n S_n^{-1} \Phi_n^T y = y^T (I - \Phi_n S_n^{-1} \Phi_n^T) y$$

$$\begin{aligned}
&= [(I - \Phi_n S_n^{-1} \Phi_n^T) y]^T [(I - \Phi_n S_n^{-1} \Phi_n^T) y] \\
&= e^T e
\end{aligned} \tag{5.25}$$

p 即是残差平方和。

将式(5.23)、式(5.25)代入式(5.21),即得式(5.20)。

利用引理 5.1 和定理 5.1 可得到定阶的递推算法。为了叙述方便,我们把第 i 步得到的矩阵 $\Phi(k, l)$ 记为 Φ_i , 用 Φ_{i+1} 表示下一步将得到的矩阵 Φ , 那么, 若 $\Phi_i = \Phi(k, l)$, 则

$$\Phi_{i+1} = [\Phi(k, l) y] \tag{5.26}$$

其中

$$y = \begin{cases} y_{k+1}, & \text{当 } k = l \\ u_{l+1}, & \text{当 } k > l \end{cases} \tag{5.27}$$

令

$$S_i = \Phi_i^T \Phi_i$$

则

$$S_{i+1} = \Phi_{i+1}^T \Phi_{i+1} = [\Phi_i y]^T [\Phi_i y] = \begin{bmatrix} \Phi_i^T \Phi_i & \Phi_i^T y \\ y^T \Phi_i & y^T y \end{bmatrix} \tag{5.28}$$

若 S_i 满秩, 即 S_i^{-1} 存在, 则由引理 5.1 得

$$\det S_{i+1} = \det S_i \cdot \det [y^T y - y^T \Phi_i S_i^{-1} \Phi_i^T y] \tag{5.29}$$

如果 $\det S_{i+1} = 0$ (当且仅当 $\det [y^T y - y^T \Phi_i S_i^{-1} \Phi_i^T y] = 0$), 则可得阶次估计 $n = k - 1$ 。同时得到参数的最小二乘估计

$$\theta_i = S_i^{-1} \Phi_i y \tag{5.30}$$

如果 $\det S_{i+1} \neq 0$, 则由式(5.20)得

$$S_{i+1}^{-1} = \begin{bmatrix} S_i & \Phi_i^T y \\ y^T \Phi_i & y^T y \end{bmatrix}^{-1} = \begin{bmatrix} S_i^{-1} + \theta_i \theta_i^T / p & -\theta_i / p \\ -\theta_i^T / p & 1/p \end{bmatrix} \tag{5.31}$$

其中

$$p = [y^T y - y^T \Phi_i S_i^{-1} \Phi_i^T y] = (y - \Phi_i \theta)^T (y - \Phi_i \theta) \tag{5.32}$$

从上面的讨论可以看出, 从 $S_1 = \Phi^T(1, 0) \Phi(1, 0)$ 出发, 按式(5.27)给出的规则 (或者说按照式(5.14)给出的顺序) 逐次增加向量, 直到某个 n , 使得 $\det S_{n+1} = 0$, 则确定了阶次 n 和相应的参数 θ 。具体计算步骤如下:

- (1) 置 $k=1; l=0; i=1$; 指定阈值 ϵ 。
- (2) 置 $\Phi_i = y_1; S_i^{-1} = (y_1^T y_1)^{-1}$ 。
- (3) 取向量 y ; 如果 $k > l$, 则置 $y = u_{l+1}; l = l + 1$, 并继续步骤(4)。如果 $l = k$, 则置 $y = y_{k+1}; k = k + 1$, 并继续步骤(4)。

(4) 计算

$$\begin{aligned}
\theta &= S_i^{-1} \Phi_i^T y \\
e &= y - \Phi_i \theta \\
p &= e^T e
\end{aligned}$$

(5) 如果 $k=l$, 则转至步骤(7)。

(6) 如果 $|p| < \epsilon$, 转至步骤(8)。

(7) 构造

$$\begin{aligned}\Phi_{i+1} &= [\Phi_i, \mathbf{y}] \\ \beta &= \theta/p \\ S_{i+1}^{-1} &= \begin{bmatrix} S_i^{-1} + \beta\beta^T & -\beta \\ -\beta^T & p^{-1} \end{bmatrix}\end{aligned}$$

置 $i=i+1$, 转至步骤(3)。

(8) 置 $n=k-1, \hat{\theta}=\theta$; 输出模型形式及相应参数, 停止计算。

5.2.3 几点说明

(1) 注意到算法中 p 是标量, 所以整个计算过程无矩阵求逆运算, 因此计算量非常小。

(2) 在具体计算步骤中, 不是直接通过计算矩阵 S 的行列式 $\det S$ 的值是否为零来判别矩阵 S 的相关性, 而是判别 p 的值。因为 $\det S_{i+1} = p \det S_i$ (由式(5.29)), 若 $\det S_i \neq 0$, 则 $\det S_{i+1} = 0 \Leftrightarrow p = 0$ 。更重要的是, 从数值计算角度考虑, 用行列式的绝对数值的大小往往很难判断其相关性, 这是因为其行列式的值强烈地依赖于数值的量级和向量长度, 而从其值的相对变化, 或者说其值是否有陡降变化来判断更为合理。这里用 p 值判断就是突出了行列式值的相对变化 ($p = \det S_{i+1} / \det S_i$), 同时这也可抵消数值计算带来的一些误差。当然, 在数值计算中, p 不可能严格为零, 所以给一阈值 ϵ , 但 ϵ 的值需经多次试验确定。

(3) p 的值不仅取决于矩阵 Φ 中向量的相关性, 而且也依赖于向量的长度和量级。因此 ϵ 的选取需经过试验。但由定理 5.1 知, 这里 p 实际上是残差平方和, 因此可用本章 5.1 节中所介绍的方法来确定阶, 以避免选择阈值 ϵ 的困难。

(4) 正如(3)中所指出的, 实际计算表明, p 的值不仅依赖于计算误差, 还依赖于数值的量级和向量的长度。为了克服量级的影响, 常常首先将矩阵 H 中的向量“标准化”, 即对 H 中所有向量做变换

$$\bar{\mathbf{y}}(k) = \frac{\mathbf{y}(k)}{\max_{\mathbf{y}(j) \in H} \|\mathbf{y}(j)\|}$$

其中, $\|\mathbf{y}(j)\|$ 表示向量 $\mathbf{y}(j)$ 的欧几里德范数; $\mathbf{y}(k)$ 表示矩阵 H 中 (包括输入 u 和输出 y) 的所有向量。然后再按原步骤进行辨识, 这样不仅可以提高精度, 而且得到的参数估计仍不变。

事实上, 令

$$a = \max_{i,j} \{ \|\mathbf{y}(i)\|, \|\mathbf{u}(j)\| \}, \mathbf{y}^*(k) = \frac{1}{a} \mathbf{y}(k), \mathbf{u}^*(k) = \frac{1}{a} \mathbf{u}(k)$$

则 $\Phi_i^* = \frac{1}{a} \Phi_i$, 因此

$$\hat{\theta}_i^* = (\Phi_i^{*T} \Phi_i^*)^{-1} \Phi_i^{*T} \mathbf{y}^* = \left(\frac{1}{a^2} \Phi_i^T \Phi_i \right)^{-1} \frac{1}{a} \Phi_i^T \frac{1}{a} \mathbf{y} = (\Phi_i^T \Phi_i)^{-1} \Phi_i^T \mathbf{y} = \hat{\theta}$$

5.3 仿真研究

从实际系统测得一批输入、输出数据, 用任意一种辨识方法 (只要是有效的), 总能得到一个数学模型。但所得到的模型是否能正确地描述系统的动态特性呢? 如果用不同的方法得到不同的模型, 或者用不同的数据得到不同的模型, 那么, 哪一个模型能更好地描述系统的动态特性呢? 这些问题在实际建模过程中经常遇到。在系统设计中经常碰到的另一类问题, 即是:

在大多数的情况下,所设计的系统不能立即到实际装置上进行试验,况且有些系统在设计时,实际装置是根本没有的。一个设计好或设计中的方案,一方面不可能轻易地进行装置试验,另一方面又必须要从实验的效果来考察和修改方案,这个矛盾如何解决呢?解决的方法之一是进行计算机仿真研究。下面简单介绍一下对于上述问题如何进行计算机仿真研究。

5.3.1 辨识方法的仿真研究

这里所要解决的问题是比较不同辨识方法或者考察新辨识方法的有效性。

首先用一个已知模型来表示被辨识的系统。例如

$$y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_1 u(k-1) + \cdots + b_m u(k-m) \quad (5.33)$$

再构造序列 $\{u(k)\}$ (常用伪随机二位式序列,见附录 C) 作为系统的输入,然后用式 (5.33) 计算得到系统的输出 $\{y(k)\}$ 。

将数据 $\{u(k)\}, \{y(k)\}$ 作为从实际系统采样得到的数据,可以用不同(或新提出)的辨识方法来估计系统的阶和参数。由于系统的真实阶和参数是已知的,因此从辨识的结果可以粗略地比较各种方法的优劣。

在进行仿真研究时,为了使输出更符合实际情况,需要在模型上增加随机噪声项。这时式 (5.33) 变为

$$y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_1 u(k-1) + \cdots + b_m u(k-m) + \lambda e(k) \quad (5.34)$$

这里 $e(k)$ 通常采用独立正态伪随机序列(其产生方法见附录 D)。 λ 是大于零的常数,其值的大小体现了噪声的强弱。

进一步地讲,我们还可以构造相关噪声,比如, $\varepsilon(k) = e(k) + c_1 e(k-1) + \cdots + c_r e(k-r)$, 将 $\varepsilon(k)$ 加到模型式 (5.33) 上。

用这些模型产生数据,然后根据这些数据用不同方法进行辨识,进而比较不同辨识方法的效果。

综上所述,辨识方法的仿真研究可归纳为如下步骤:

- (1) 给出数学模型(最好是具有代表性的数学模型);
- (2) 构造输入序列(见附录 C);
- (3) 构造正态伪随机序列(见附录 D);
- (4) 由数学模型产生数据;
- (5) 根据产生的数据进行辨识;
- (6) 比较辨识方法的性能,如辨识精度、鲁棒性等。

上述研究过程,其输入序列也可以是由调节器产生的,并且可以与被辨识的系统构成闭环系统,用来考察闭环系统辨识的一些特点,如同在 4.3 节中给出的例子那样。另外,用于仿真的数学模型,也可以根据问题的需要构造各种情况。例如,时变参数、变结构等来考察辨识方法的性能。

5.3.2 对模型适用性的仿真研究

这里要解决的问题是考察已知模型是否基本上符合实际系统,或者在几个模型中选择一个与实际系统符合较好的模型。这里所谓符合较好,是指模型的输出与实际系统的输出(在相同的输入下)之间的误差在工程上是可以接受的。

对此问题,一个简单的方法是首先对实际系统进行长期的观察,取得若干组包括各种情况

的数据。然后将取得的数据代入已知模型,计算出模型相应的输出,通过对实际输出和模型输出的比较,找出一个与实测各组数据都能较好地符合的模型。此外,还需对此模型构造一些特殊的输入(如边界输入、不合理输入等)来考察其输出,如果其输出是基本符合实际经验的,则初步选定此模型为适用模型。

当然,模型的最终确定还需要在实际中经过长时间的考验和修正。

5.3.3 控制系统设计中的计算机仿真研究

这里要解决的问题是通过仿真,研究给出理想的设计方案。它的基本思想是:首先用辨识或其他方法得到的数学模型来表示实际装置,这个模型可以是状态方程,也可以是差分方程及其他形式,将通过这个模型计算得到的输出作为实际过程的输出。这样我们就可以设计各种输入(控制器),加上各种约束,通过模型计算得到系统的响应。由此来分析控制系统的性能,调整、修正设计。直到最后得到一个满意的设计方案为止。

从上面的简单介绍不难看出,仿真研究是一种很有价值的手段。因为其费用少、周期短、见效快,可以在短时间内完成大量的工作,因此这种研究受到了普遍的重视。值得指出的是,当今计算机仿真已发展成为一门独立的、活跃的、具有广泛应用领域的学科,是认识客观世界的一种重要手段,在军事、生物、社会、经济等许多领域的研究和应用中起着重要作用,而且在当今的科学研究中越来越显示出其重要地位。

习 题

1. 证明引理 5.1。
2. 用如下模型产生仿真数据(200 个)

$$y(k)=1.5y(k-1)-0.7y(k-2)+u(k-1)+0.5u(k-2)+e(k)$$

其中 $e(k)$ 为正态白噪声。分别用 F 检验、AIC 准则与阶和参数同时辨识的递推算法在计算机上实现模型的辨识。

3. 将阶和参数同时辨识递推算法中的第(6)步改为用 F 检验判断。

* 第 6 章 多变量线性系统的辨识

多变量线性系统是指有若干个输入和若干个输出的线性系统,如图 6.1 所示。

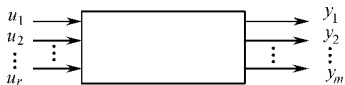


图 6.1 多变量线性系统

线性系统的辨识主要有两部分:结构辨识和参数估计,对线性模型来说,最主要的先验知识就是“模型是线性的”。在已知线性的前提下,其结构可以通过输入、输出数据来确定,对于单变量系统,模型的结构就是阶,定阶的方法已经在第 5 章中讨论过。对于多变量系统,情形就要复杂得多,对于一个已知输入、输出关系的系统可以有多种不同形式的模型,模型的结构就不是用一个参数所能描述的,因为可能的模型有无穷多个,所以要逐个地研究它们的结构是不可能的。因此,必须对多变量线性系统的结构进行深入研究,Luenberger(1967)较早地研究了线性系统的规范形结构,但是它对描述结构的参数不完全,所以没有完全解决问题。Popov(1972)首先提出了用一组独立的、完备的不变量描述结构,这组不变量完全刻画了线性系统结构的本质。Guidorzi(1975)针对 Popov 所提出的不变量找出了规范形的输入、输出关系,从而给出了通过输入、输出数据确定结构指标的具体方法,这是多变量线性系统辨识的一个重大进展。随后,Bingulas 和 Farias(1976)提出了另一种形式的规范形及辨识方法,王治铭,卢桂章(1982)证明了这两种规范形实质上是一样的,并给出了输入、输出方程的一般形式。以后又有许多不同的算法提出来(Kotta,1979;Besger et al,1980),王秀峰和卢桂章(1981)提出了辨识结构指标的递推算法,改进了 Guidorzi 的算法。

除了 Popov,Bingulas 等提出的规范形外,还有没有其他的规范形?决定规范形的又是什么呢?如果还有其他的规范形,辨识方法有没有区别?针对不同的规范形,具体的辨识方法是什么?这些问题在卢桂章和王秀峰(1984)的文章中做了回答。

多变量线性系统还可以有其他的描述方式,如一般的非规范形的输入、输出方程、传递函数阵、非参数的脉冲响应阵(Markov 参数)等,对于这些模型形式,以及它们之间的相互关系,可参见 Hajdasinski(1980)。对于非规范形的输入、输出方程,只要通过简单的系统分解,把多变量线性系统分解为若干个子系统,而每个子系统是一个多输入单输出的系统,可按照一般单变量系统进行辨识,关于 Markov 参数的辨识将在第 7 章中讨论。

本章将讨论规范的状态方程及对应的输入、输出方程的辨识。为了使线性系统的描述具有惟一性,首先从各种可能的形式中在代数等价意义下找出所谓的规范形式。这种规范形式的结构和参数都具有不变性,而这组不变量取决于某些向量的相关性,这些将是本章要讨论的问题之一,在此基础给出一套完整(包括状态实现)的辨识方法。

6.1 不变量、适宜选择路线及规范形

前面已经指出,对于给定的多变量线性系统,状态方程的表示是不惟一的,我们的目的就

是要找出系统在代数等价意义下不变的、具有规范形式的状态方程。那么什么是在代数等价意义下不变？不变的量是什么？它们是否惟一？它们是由什么决定的？系统的状态方程表示的规范形式是什么？对上述问题，本节将做简单的讨论。

下面首先给出代数等价的定义。

6.1.1 代数等价系统

考虑线性多变量系统

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}\quad (6.1)$$

其中, $x(k)$ 为 n 维状态向量, $u(k)$ 是 r 维输入向量, $y(k)$ 是 m 维输出向量。 A, B, C, D 分别为 $n \times n, n \times r, m \times n, m \times r$ 维矩阵。

将系统式(6.1)简记为 $\sigma(A, B, C, D)$ (以后的讨论都采用类似记号, 或简记为 σ)。假定 (A, B) 是完全能控对, (C, A) 是完全能观对, 且 $m \leq n$, 如果对状态变量做非奇异线性变换, 即令 $z(k) = Tx(k)$, T 为非奇异常数矩阵, 则在新状态变量下, 系统表示为

$$\sigma^*(A^*, B^*, C^*, D) = \sigma^*(TAT^{-1}, TB, CT^{-1}, D)$$

事实上, 由 $z(k) = Tx(k)$ 即 $x(k) = T^{-1}z(k)$, 代入式(6.1)得

$$\begin{aligned}T^{-1}z(k+1) &= AT^{-1}z(k) + Bu(k) \\ y(k) &= CT^{-1}z(k) + Du(k)\end{aligned}$$

即

$$\begin{aligned}z(k+1) &= TAT^{-1}z(k) + TBu(k) \\ y(k) &= CT^{-1}z(k) + Du(k)\end{aligned}$$

或简记为

$$\begin{aligned}z(k+1) &= A^*z(k) + B^*u(k) \\ y(k) &= C^*z(k) + D^*u(k)\end{aligned}\quad (6.2)$$

其中

$$A^* = TAT^{-1}, B^* = TB, C^* = CT^{-1}, D^* = D \quad (6.3)$$

另一方面, σ^* 和 σ 可视为两个线性系统, 很容易验证, 通过可逆线性变换得到的线性系统 σ^* 与 σ 具有同样的传递函数阵, 也就是说, 具有相同的输入、输出关系, 即

$$C(ZI - A)^{-1}B + D = C^*(ZI - A^*)^{-1}B^* + D^*$$

式中, Z 是 z 变换算子, I 是单位矩阵。

事实上, 用式(6.3)直接代入得

$$\begin{aligned}C^*(ZI - A^*)^{-1}B^* + D^* &= CT^{-1}(ZI - TAT^{-1})^{-1}TB + D \\ &= CT^{-1}(T(ZI - AT^{-1})^{-1}TB + D) \\ &= C(ZI - A)^{-1}B + D\end{aligned}$$

从式(6.3)可看到, 线性系统 σ^* 中的 A^* 是由 σ 中 A 经过代数的相似变换得到, 而且 σ 和 σ^* 具有相同的输入、输出关系。这就使我们对满足式(6.3)的两个线性系统备感兴趣, 因此引入如下代数等价定义。

定义 6.1 两个线性系统 $\sigma(A, B, C, D)$ 和 $\sigma^*(A^*, B^*, C^*, D^*)$, 如果存在非奇异线性变换矩阵 T , 使得式(6.3)成立, 则称线性系统 σ 和 σ^* 是代数等价的。

由此得到: 如果系统是完全能控、完全能观的, 代数等价的线性系统具有同样的传递函数

阵,也就是具有同样的输入、输出关系。反之结论也对,即具有相同传递函数阵的两个线性系统一定是代数等价的。但对不完全能控、能观的系统,结论未必正确。

6.1.2 适宜选择路线与不变量

如上所述,代数等价的一类系统描述了同样的输入、输出关系,要得到系统不变的最简单的表示,就需要知道在代数等价意义下不变的量是什么?如何确定这些不变量?由式(6.3)知道,代数等价的系统中,矩阵 D 是不变的,因此,在讨论不变量及规范形时,为了简单,先将其略去,在讨论辨识方法时再将其添上。

考虑线性系统 $\sigma(A, B, C)$, 即

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (6.4)$$

其中符号的定义同式(6.1)。假定 (C, A) 是完全能观的,即对应于系统 σ 的能观性矩阵

$$\mathbf{F} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \quad (6.5)$$

矩阵 \mathbf{F} 的秩为 n 。

为了简单,在下面的讨论中我们还假定矩阵 \mathbf{C} 的秩为行数 m (否则,将系统中多余的输出去掉,使新的矩阵 \mathbf{C} 满足其秩等于行数)。

定义 6.2 集合 $\{1, 2, \dots, mn\}$ 的任一个包含 n 个元素的子集 S 称为是一个选择。

若将能观性矩阵 \mathbf{F} 的行 $\mathbf{c}_i^T \mathbf{A}^j$ (\mathbf{c}_i^T 为矩阵 \mathbf{C} 的第 i 行), 用数偶 (i, j) 表示, 并按 (i, j) 的顺序 $(1, 0), (2, 0), \dots, (m, 0), \dots, (1, n-1), (2, n-1), \dots, (m, n-1)$ 编为 1 至 mn 号, 则一个选择 S 就对应于选择出能观矩阵 \mathbf{F} 的 n 个行。因此, 在后面的讨论中, 对一个具体系统而言, 对选择与由选择得到 \mathbf{F} 中的 n 个行向量不加区别的都称为选择。

由定义 6.2 可知, 所谓两个选择相等, 就是两个子集相等, 也就是两个子集包含的元素相同。

定义 6.3 选择 S 如果满足: 若 $(i, j) \in S$, 则 $(i, j-1) \in S$, 那么称 S 为适宜选择。

对于我们的问题, 所关心的是这样的适宜选择 S : 对于给定的系统, 能观性矩阵 \mathbf{F} 中按 S 选出的行向量是线性无关的, 或者说按 S 选出的 n 个行向量构成的矩阵是可逆的。不加声明, 下面所提到的适宜选择均指满足此要求的适宜选择。

显然, 对于完全能观的系统, 满足上述要求的适宜选择一定存在。事实上, 我们记 \mathbf{c}_i^T ($i = 1, 2, 3, \dots, m$) 为矩阵 \mathbf{C} 的第 i 个行向量, 按如下顺序

$$\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_m^T, \mathbf{c}_1^T \mathbf{A}, \dots, \mathbf{c}_m^T \mathbf{A}, \dots, \mathbf{c}_1^T \mathbf{A}^{n-1}, \dots, \mathbf{c}_m^T \mathbf{A}^{n-1}$$

自左至右挑选线性无关(行)向量, 向量 $\mathbf{c}_i^T \mathbf{A}^j$ 被选出, 当且仅当它与前面所选出的向量线性无关。由于能观矩阵 \mathbf{F} 的秩为 n , 所以一定能选出 n 个线性无关的向量。由选择的顺序, 显然这 n 个线性无关向量对应一个适宜选择。

定义 6.4 对能观系统 σ , 令 $\nu_i(S)$ ($i = 1, \dots, m$) 表示按适宜选择 S 选出的形如 $\mathbf{c}_i^T \mathbf{A}^j$ ($j \geq 0$) 的向量个数, 则

$$\sum_{i=1}^m \nu_i(S) = n \quad (6.6)$$

$\{\nu_1(S), \nu_2(S), \dots, \nu_m(S)\}$ 称为系统 σ 关于适宜选择 S 的克隆尼克(Kronecker)指标。进一步的讨论可知,克隆尼克指标确定了系统的结构,所以也称之为结构指标。

由结构指标的定义,很显然,它是依赖于选择的,下面用一个简单例子来加以说明。

【例 6.1】设能观系统 $\sigma(A, B, C)$, 其中

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

那么,能观性矩阵 F 为

$$F = \begin{pmatrix} c_1^T \\ c_2^T \\ c_1^T A \\ c_2^T A \\ c_1^T A^2 \\ c_2^T A^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

由定义可知选择

$$S_1: (c_1^T, c_1^T A, c_1^T A^2) = \{(1 \ 0 \ 0), (1 \ 0 \ 1), (2 \ 1 \ 1)\}$$

$$S_2: (c_1^T, c_2^T, c_1^T A) = \{(1 \ 0 \ 0), (0 \ 1 \ 0), (1 \ 0 \ 1)\}$$

$$S_3: (c_1^T, c_2^T, c_2^T A) = \{(1 \ 0 \ 0), (0 \ 1 \ 0), (0 \ 0 \ 1)\}$$

$$S_4: (c_2^T, c_2^T A, c_2^T A^2) = \{(0 \ 1 \ 0), (0 \ 0 \ 1), (1 \ 1 \ 0)\}$$

都是适宜选择。在这 4 种选择下,结构指标分别为

$$\nu_1(S_1) = 3, \quad \nu_2(S_1) = 0$$

$$\nu_1(S_2) = 2, \quad \nu_2(S_2) = 1$$

$$\nu_1(S_3) = 1, \quad \nu_2(S_3) = 2$$

$$\nu_1(S_4) = 0, \quad \nu_2(S_4) = 3$$

从例 6.1 可以看出,对同一系统,对于不同的选择 S ,相应的结构指标一般也不一样(此例中,如果忽略输出的顺序,则由 S_1 和 S_4 , S_2 和 S_3 得到的结构指标可视为一样的)。但对固定的选择, $\{\nu_i(S)\}$ 则惟一确定。因此,当在某一选择下讨论问题时,或者不必要强调选择时,简记 $\nu_i(S)$ 为 ν_i 。

定理 6.1 设线性系统 $\sigma(A, B, C)$ 是完全可观测的,那么一定存在一组结构指标 $\{\nu_i\}$ 及惟一确定的一组有序的实数组 $\{\alpha_{ij,k}\}$, 使得

$$c_i^T A^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k} c_j^T A^{k-1} \quad (i = 1, 2, \dots, m) \quad (6.7)$$

证明 由于系统 σ 是完全可观测的,所以一定存在一个适宜选择 S ,使得按适宜选择 S 所选出的 n 个向量

$$c_1^T, c_1^T A, \dots, c_1^T A^{\nu_1-1}, \dots, c_m^T, c_m^T A, \dots, c_m^T A^{\nu_m-1} \quad (6.8)$$

是线性无关的。由定义 6.4 知, $\{\nu_i, i=1, \dots, m\}$ 就是由 S 确定的系统 σ 的结构指标。所以向量组式(6.8)就构成了 n 维线性空间的一组基底,因此,任一 n 维向量都可由这组向量线性表示,从而对任意 i ($i=1, 2, \dots, m$) 都有

$$c_i^T A^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k} c_j^T A^{k-1}$$

其次,由于向量组式(6.8)构成了 n 维线性空间的一组基底,所以上述表示是惟一的,即有序实数组 $\{\alpha_{ij,k}\}$ 惟一确定。

应当指出,在式(6.7)中,实数组 $\{\alpha_{ij,k}, i, j=1, \dots, m; k=1, \dots, \nu_j\}$ 中的某些参数可能恒为零。至于其中哪些参数恒为零,依赖于适宜选择,这将在 6.1.3 节中讨论。

定理 6.1 指出,对于可观系统,总可找到一组 $\{\nu_j\}$ 及 $\{\alpha_{ij,k}\}$,使得式(6.7)成立,那么这些量在代数等价意义下是否变化呢?回答是否定的。

定理 6.2 设系统 $\sigma(A, B, C)$ 和 $\sigma^*(A^*, B^*, C^*)$ 是两个代数等价的线性系统,且都是完全能观测的。则对任意适宜选择 S ,如果系统 σ 所选出的向量组 $\{c_i^T A^j\}$ 是线性无关组,则关于 σ^* 所选出的向量组 $\{c_i^{*T} A^{*j}\}$ 也是线性无关组,而且所确定的结构指标 $\{\nu_j\}$ 和有序实数组 $\{\alpha_{ij,k}\}$ (由定理 6.1 所确定)是一样的。

证明 由假设知,系统 σ 和 σ^* 代数等价,即存在可逆矩阵 T ,使得 $A^* = TAT^{-1}, B^* = TB, C^* = CT^{-1}$,因此,系统 σ^* 的能观矩阵 F^* 和系统 σ 的能观矩阵 F 之间满足如下关系

$$F^* = \begin{bmatrix} C^* \\ C^* A^* \\ \vdots \\ C^* A^{*n-1} \end{bmatrix} = \begin{bmatrix} CT^{-1} \\ CAT^{-1} \\ \vdots \\ CA^{n-1} T^{-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} T^{-1} = FT^{-1} \quad (6.9)$$

式(6.9)说明,系统 σ 的能观矩阵 F 的行向量与系统 σ^* 的能观矩阵 F^* 的行向量具有相同的相关性,因此,同一适宜选择所确定的向量组具有相同的相关性。显然,由结构指标的定义,对于系统 σ 和 σ^* 由同一适宜选择所确定的结构指标 $\{\nu_i\}$ 也是一样的。

下面再证明由定理 6.1 所确定的有序实数组 $\{\alpha_{ij,k}\}$ 也相同。假定系统 σ 和 σ^* 由定理 6.1 所确定的有序实数组分别为 $\{\alpha_{ij,k}\}$ 和 $\{\alpha_{ij,k}^*\}$,即对任意 $i(i=1, 2, \dots, m)$ 有

$$c_i^T A^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k} c_j^T A^{k-1} \quad (6.10)$$

$$c_i^{*T} A^{*\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k}^* c_j^{*T} A^{*k-1} \quad (6.11)$$

由于 σ 和 σ^* 代数等价,故式(6.11)可写为

$$(c_i^T T^{-1})(TAT^{-1})^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k}^* (c_j^T T^{-1})(TAT^{-1})^{k-1}$$

再利用等式: $(TAT^{-1})^k = TA^k T^{-1}$, k 为整数,上式可简化为

$$c_i^T A^{\nu_i} T^{-1} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k}^* c_j^T A^{k-1} T^{-1}$$

上式两端再右乘以矩阵 T 得

$$c_i^T A^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k}^* c_j^T A^{k-1} \quad (6.12)$$

比较式(6.10)和式(6.12),并注意到向量组 $\{c_j^T A^{k-1}\} (j=1, 2, \dots, m; k=1, 2, \dots, \nu_j)$ 是 n 维线性空间的基底,因此, $c_i^T A^{\nu_i}$ 的表示惟一,从而有

$$\alpha_{ij,k} = \alpha_{ij,k}^*, i, j = 1, 2, \dots, m, k = 1, 2, \dots, \nu_j$$

于是定理得证。

由上述定理可知,凡是代数等价的线性系统都存在共同的结构指标 $\{\nu_i\}$ 和参数集 $\{\alpha_{ij,k}\}$,

换句话说,由适宜选择所确定的变量组 $\{\nu_i, \alpha_{ij,k}\}$ 在代数等价意义下是不变的,因此称之为不变量组。可以证明(Popov,1972),不变量组是独立的不变量完备组。即 $\{\nu_i, \alpha_{ij,k}\}$ 完全可以确定一个系统, $\{\nu_i, \alpha_{ij,k}\}$ 的不同取值代表不同(等价类)的系统,并且对任何一个系统都能找到与之对应的 $\{\nu_i, \alpha_{ij,k}\}$ 。所以,辨识系统就是要辨识这组参数,进而得到系统表示的规范形式。

6.1.3 适宜选择路线与规范形

1. 适宜选择路线

6.1.2 节中已经指出,多变量线性系统的辨识就是辨识结构指标 $\{\nu_i\}$ 及相应参数 $\{\alpha_{ij,k}\}$ 。然而从前面的讨论知道,这些量都是由满足一定要求的适宜选择确定的。但是从辨识的角度考虑,对于待辨识的系统来说,这样的适宜选择并不能事先知道,它只能是辨识的结果。所以,不可能从给定的适宜选择出发去找不变量和规范形。因此,在辨识中需找出一个与适宜选择作用相当的概念,即所谓适宜选择路线。

定义 6.5 集合 $\{1, 2, \dots, mn\}$ 的任一排列称为是一个选择路线。

对于给定的完全能观系统 $\sigma(A, B, C)$ 而言,任一选择路线对应于向量组

$$c_1^T, c_1^T A_1, \dots, c_1^T A^{n-1}, c_2^T, c_2^T A, \dots, c_2^T A^{n-1}, \dots, c_m^T, c_m^T A, \dots, c_m^T A^{n-1} \quad (6.13)$$

的一个排列。因此,对于给定的系统,按照给定的选择路线去选择无关向量,即可得到一个选择,按照不同的选择路线,一般得到不同选择。

定义 6.6 一个选择路线 L , 如果它对应向量组式 (6.13) 的一个排列总保持 $c_i^T A^j$ 在 $c_i^T A^{j+1}$ 的前面, 对一切 i, j 成立, 则称这个选择路线为适宜选择路线。由适宜选择路线所决定的选择称为适宜选择。

【例 6.2】对于式 (6.4) 所描述的系统

$$c_1^T, c_2^T, \dots, c_m^T, c_1^T A, \dots, c_m^T A, \dots, c_1^T A^{n-1}, \dots, c_m^T A^{n-1} \quad (6.14)$$

$$c_1^T, c_1^T A_1, \dots, c_1^T A^{n-1}, c_2^T, c_2^T A, \dots, c_2^T A^{n-1}, \dots, c_m^T, c_m^T A, \dots, c_m^T A^{n-1} \quad (6.15)$$

$$c_1^T, c_2^T, \dots, c_m^T, c_m^T A, c_{m-1}^T A, \dots, c_1^T A, \dots, c_m^T A^{n-1}, \dots, c_1^T A^{n-1} \quad (6.16)$$

$$c_2^T, c_2^T A, \dots, c_2^T A^{n-1}, c_3^T, c_3^T A, \dots, c_3^T A^{n-1}, \dots, c_1^T, \dots, c_1^T A^{n-1} \quad (6.17)$$

...

都是适宜选择路线。可见,适宜选择路线有许多种,当然相应地适宜选择也有许多种。但是,辨识中最常用的、具有代表性的就是由式 (6.14) 和式 (6.15) 所确定的适宜选择路线。

式 (6.14) 所确定的适宜选择路线称为波波夫选择路线 (Popov, 1972), 记为 PL, 由式 (6.15) 所确定的适宜选择路线称为自然选择路线, 记为 NL。

对于给定的系统, 规定了适宜选择路线, 就意味着按照规定的路线以这样的原则选择线性无关的向量: 某向量被选出, 当且仅当它与前面所选出的向量线性无关。因此, 一旦选择路线给定, 则适宜选择惟一确定, 从而不变量和规范形也相应确定。下面, 仅对 PL 适宜选择路线下的不变量和规范形做详细讨论。类似的思路可以应用于在 NL 选择路线下不变量和规范形的讨论, 这里不做介绍, 有兴趣的读者可参见卢桂章、王秀峰 (1982) 的文章。

为了叙述方便, 我们记在 PL 选择路线下得到的规范形称为 PCF 规范形。

2. PCF 规范形

仍考虑由式 (6.4) 描述的线性系统

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$

假设按 PL(由式(6.14)定义)选择路线得到的适宜选择 S_p 为

$$S_p: \mathbf{c}_1^T, \mathbf{c}_1^T \mathbf{A}, \dots, \mathbf{c}_1^T \mathbf{A}^{\nu_1-1}, \mathbf{c}_2^T, \mathbf{c}_2^T \mathbf{A}, \dots, \mathbf{c}_2^T \mathbf{A}^{\nu_2-1}, \dots, \mathbf{c}_m^T, \mathbf{c}_m^T \mathbf{A}, \dots, \mathbf{c}_m^T \mathbf{A}^{\nu_m-1} \quad (6.18)$$

由定理 6.1 有

$$\mathbf{c}_i^T \mathbf{A}^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_j} \alpha_{ij,k} \mathbf{c}_j^T \mathbf{A}^{k-1}, i = 1, 2, \dots, m$$

但是,由 PL 的具体顺序可知,上式中参数 $\alpha_{ij,k}$ 满足

$$\alpha_{ij,k} \equiv 0 \begin{cases} k > \min\{\nu_{i+1}, \nu_j\} & \text{当 } j < i \text{ 时} \\ k > \min\{\nu_i, \nu_j\} & \text{当 } j \geq i \text{ 时} \end{cases} \quad (6.19)$$

事实上,由选择路线 PL

$$\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_m^T, \dots, \mathbf{c}_1^T \mathbf{A}^{\nu_1-1}, \dots, \mathbf{c}_{i-1}^T \mathbf{A}^{\nu_{i-1}-1}, \mathbf{c}_i^T \mathbf{A}^{\nu_i-1}, \mathbf{c}_{i+1}^T \mathbf{A}^{\nu_{i+1}-1}, \dots, \mathbf{c}_m^T \mathbf{A}^{\nu_m-1}$$

$$\mathbf{c}_1^T \mathbf{A}^{\nu_i}, \dots, \mathbf{c}_{i-1}^T \mathbf{A}^{\nu_i}, \mathbf{c}_i^T \mathbf{A}^{\nu_i}, \mathbf{c}_{i+1}^T \mathbf{A}^{\nu_i}, \dots, \mathbf{c}_m^T \mathbf{A}^{\nu_i}, \dots, \mathbf{c}_1^T \mathbf{A}^{n-1}, \dots, \mathbf{c}_m^T \mathbf{A}^{n-1}$$

以及所得到的选择式(6.18)可知, $\mathbf{c}_i^T \mathbf{A}^{\nu_i}$ 一定可由这前面所选出的向量线性表示,而它前面所选出的向量显然满足

$$\mathbf{c}_j^T \mathbf{A}^{k-1} : \begin{cases} 1 \leq k \leq \min\{\nu_{i+1}, \nu_j\} & \text{当 } j < i \text{ 时} \\ 1 \leq k \leq \min\{\nu_i, \nu_j\} & \text{当 } j \geq i \text{ 时} \end{cases}$$

这与式(6.19)意义是一样的。因此,对于适宜选择路线 PL,式(6.7)变为

$$\mathbf{c}_i^T \mathbf{A}^{\nu_i} = \sum_{j=1}^{i-1} \sum_{k=1}^{\min\{\nu_{i+1}, \nu_j\}} \alpha_{ij,k} \mathbf{c}_j^T \mathbf{A}^{k-1} + \sum_{j=i}^m \sum_{k=1}^{\min\{\nu_i, \nu_j\}} \alpha_{ij,k} \mathbf{c}_j^T \mathbf{A}^{k-1} \quad (6.20)$$

引进记号

$$\nu_{ij} = \begin{cases} \min\{\nu_{i+1}, \nu_j\} & \text{当 } j < i \text{ 时} \\ \min\{\nu_i, \nu_j\} & \text{当 } j \geq i \text{ 时} \end{cases} \quad (6.21)$$

则式(6.20)可简化为

$$\mathbf{c}_i^T \mathbf{A}^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_{ij}} \alpha_{ij,k} \mathbf{c}_j^T \mathbf{A}^{k-1} \quad i = 1, 2, \dots, m \quad (6.22)$$

如果由适宜选择 S_p 得到的行向量构造变换阵

$$\mathbf{T} = \begin{pmatrix} \mathbf{c}_1^T \\ \mathbf{c}_1^T \mathbf{A} \\ \vdots \\ \mathbf{c}_1^T \mathbf{A}^{\nu_1-1} \\ \vdots \\ \mathbf{c}_m^T \\ \mathbf{c}_m^T \mathbf{A} \\ \vdots \\ \mathbf{c}_m^T \mathbf{A}^{\nu_m-1} \end{pmatrix} \quad (6.23)$$

则有如下结论。

定理 6.3 对完全能观系统式(6.4)做状态变换: $\mathbf{x}^* = \mathbf{T}\mathbf{x}$, 得到

$$\mathbf{x}^*(k+1) = \mathbf{A}^* \mathbf{x}^*(k) + \mathbf{B}^* \mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}^* \mathbf{x}^*(k)$$

则 $(A^*, B^*, C^*) = (TAT^{-1}, TB, CT^{-1})$ 有如下结构

$$A^* = (A_{ij}^*) \quad (6.24)$$

其中

$$A_{ii}^* = \begin{pmatrix} 0 & 1 & & \\ \vdots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 \\ \alpha_{ii,1} & \alpha_{ii,2} & \cdots & \alpha_{ii,\nu_i} \end{pmatrix} \quad A_{ij}^* = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \\ \alpha_{ij,1} & \alpha_{ij,2} & \vdots & \alpha_{ij,\nu_{ij}} \end{pmatrix}$$

$i, j = 1, 2, \cdots, m$

$$C^* = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ & \cdots & & & \cdots & \\ & & & 1 & 0 & \cdots & 0 \end{pmatrix}$$

$\uparrow \qquad \qquad \uparrow \qquad \qquad \uparrow$
 $1 \qquad \qquad \nu_1 + 1, \quad \sum_{j=1}^{m-1} \nu_j + 1$

(6.25)

B^* 没有特殊的规范形式。

证明 由于 $\nu_1 + \nu_2 + \cdots + \nu_m = n$, 所以变换矩阵 T 是可逆方阵, 从而有

$$TT^{-1} = \begin{pmatrix} c_1^T T^{-1} \\ \vdots \\ c_1^T A^{\nu_1-1} T^{-1} \\ \vdots \\ c_m^T T^{-1} \\ \vdots \\ c_m^T A^{\nu_m-1} T^{-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \\ & & \ddots & \ddots \\ & & & \ddots & 0 \\ 0 & \cdots & & 0 & 1 \end{pmatrix}_{n \times n}$$

即

$$\begin{aligned} c_1^T T^{-1} &= (10\cdots\cdots 0) \\ c_1^T A T^{-1} &= (010\cdots\cdots 0) \\ &\vdots \\ c_m^T A^{\nu_m-1} T^{-1} &= (0\cdots\cdots 01) \end{aligned} \quad (6.26)$$

一般地有

$$c_j^T A^{k-1} T^{-1} = (0 \cdots 0 \ 1 \ 0 \cdots 0) \quad \begin{matrix} j = 1, 2, \cdots, m \\ k = 1, 2, \cdots, \nu_j \end{matrix}$$

\uparrow
 $\sum_{i=0}^{j-1} \nu_i + k \qquad \nu_0 \equiv 0$

(6.27)

再由式(6.22)知, 对任意 $i (i=1, 2, \cdots, m)$ 有

$$c_i^T A^{\nu_i} = \sum_{j=1}^m \sum_{k=1}^{\nu_{ij}} \alpha_{ij,k} c_j^T A^{k-1}$$

上式等号两端同右乘以矩阵 T^{-1} , 并将式(6.27)代入得

$$c_i^T A^{\nu_i} T^{-1} = \sum_{j=1}^m \sum_{k=1}^{\nu_{ij}} \alpha_{ij,k} c_j^T A^{k-1} T^{-1}$$

$$= (\alpha_{i1,1} \alpha_{i1,2} \cdots \alpha_{i1,\nu_{i1}} 0 \cdots 0 : \cdots : \alpha_{ii,1} \alpha_{ii,2} \cdots \alpha_{ii,\nu_i} : \cdots : \alpha_{im,1} \alpha_{im,2} \cdots \alpha_{im,\nu_{im}} 0 \cdots 0) \quad (6.28)$$

另一方面,由定理条件可知

$$A^* = TAT^{-1} = \begin{pmatrix} c_1^T A T^{-1} \\ \vdots \\ c_1^T A^{\nu_1-1} T^{-1} \\ c_1^T A^{\nu_1} T^{-1} \\ \vdots \\ c_m^T A T^{-1} \\ \vdots \\ c_m^T A^{\nu_m-1} T^{-1} \\ c_m^T A^{\nu_m} T^{-1} \end{pmatrix}$$

将式(6.27)及式(6.28)直接代入上式,可见 A^* 即为定理所要求的形式。

对于 C^* , 同样利用式(6.27)直接得到

$$C^* = CT^{-1} = \begin{pmatrix} c_1^T T^{-1} \\ c_2^T T^{-1} \\ \vdots \\ c_m^T T^{-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & \\ 0 & \cdots & & 0 & 1 & 0 \cdots 0 \end{pmatrix}$$

$$\begin{matrix} \uparrow & \uparrow & \uparrow \\ 1 & \nu_1 + 1 & \nu_1 + \cdots + \nu_{m-1} + 1 \end{matrix}$$

因此,定理得证。

由式(6.24)和式(6.25)表征的规范形,我们称为 PCF 规范形,这即是 Goodwin(1979)定义的 SSF2 型和 Guidorzi(1975)所用的规范形。

当然,用这组不变量还可以构造许多不同形式的能观规范形,其中包括 Bingulac 和 Farias (1976)所用的规范形

$$\tilde{A}^* = \left(\begin{array}{cccccccc} 0 & \cdots & 0 & 1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & & \\ \alpha_{11} & \alpha_{12} & \cdots & \cdots & & \alpha_{1q_1} & & \\ & & & 1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & & \\ \alpha_{21} & \alpha_{22} & \cdots & \cdots & & \cdots & \alpha_{2q_2} & \\ & & & \cdots & & \cdots & & \ddots \\ & & & & & & 1 & \\ \alpha_{m1} & \alpha_{m1} & \cdots & \cdots & & \cdots & \cdots & 1 \end{array} \right) \begin{matrix} \leftarrow l_1 \\ \leftarrow l_2 \\ \vdots \\ \leftarrow l_p = n \end{matrix}$$

$$\tilde{\mathbf{C}}^* = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots & \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

这里 l_i, q_i 由结构指标 $\{\nu_i\}$ 惟一确定, 且满足: $l_i = q_i - m + i$, 以及所谓能观 II 型

$$\tilde{\mathbf{A}}^* = (\tilde{\mathbf{A}}_{ij})$$

$$\tilde{\mathbf{A}}_{ii} = \begin{pmatrix} 0 & \cdots & 0 & \alpha_{ii,1} \\ 1 & & & \alpha_{ii,1} \\ & \ddots & & \\ & & 1 & \alpha_{ii,1} \end{pmatrix}_{\nu_i \times \nu_i}$$

$$\tilde{\mathbf{A}}_{ij} = \begin{pmatrix} 0 & \cdots & 0 & \alpha_{ij,1} \\ & & & \alpha_{ij,2} \\ & & & \vdots \\ & \ddots & & \alpha_{ij,\nu_{ij}} \\ & & 0 & \\ & & \vdots & \\ 0 & \cdots & 0 & 0 \end{pmatrix}_{\nu_i \times \nu_j}$$

当 $i < j$ 时; 当 $i > j$ 时, 为 $\nu_j \times \nu_i$ 维矩阵。

$$\tilde{\mathbf{C}}^* = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ \cdots & & \cdots & \cdots & & \\ 0 & \cdots & \cdots & \cdots & & 0 & 1 \end{pmatrix}$$

$\uparrow \qquad \qquad \uparrow \qquad \qquad \uparrow$
 $\nu_1 \qquad \nu_1 + \nu_2 \qquad \nu_1 + \cdots + \nu_m = n$

当然还可以得到其他形式的规范形。这些规范形表面上看形式各异, 其实都是同一组不变量所决定的, 因此, 本质上是一样的。所以在 PL 选择路线下, 规范形的辨识中只考虑 PCF 形的辨识就可以了。

上面导出了 PCF 规范形式, 顺便指出它的一个简单性质: 对于规范 PCF, 存在一个适宜选择 S , 它决定的向量可构成了一个 n 阶单位阵。

事实上, 因为 $\mathbf{A}^* = \mathbf{TAT}^{-1}$, $\mathbf{C}^* = \mathbf{CT}^{-1}$, 由式 (6.9) 可知, 关于系统 $\sigma(\mathbf{A}, \mathbf{B}, \mathbf{C})$ 的适宜选择 S_p 也必为系统 $\sigma^*(\mathbf{A}^*, \mathbf{B}^*, \mathbf{C}^*)$ 的适宜选择, 而由 S_p 所决定的向量构成矩阵 \mathbf{T}^* 为

$$\mathbf{T}^* = \begin{pmatrix} \mathbf{c}_1^{*\top} \\ \vdots \\ \mathbf{c}_1^{*\top} \mathbf{A}^{*\nu_1-1} \\ \vdots \\ \mathbf{c}_m^{*\top} \\ \vdots \\ \mathbf{c}_m^{*\top} \mathbf{A}^{*\nu_m-1} \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1^\top \\ \vdots \\ \mathbf{c}_1^\top \mathbf{A}^{\nu_1-1} \\ \vdots \\ \mathbf{c}_m^\top \\ \vdots \\ \mathbf{c}_m^\top \mathbf{A}^{\nu_m-1} \end{pmatrix} \quad \mathbf{T}^{-1} = \mathbf{T}\mathbf{T}^{-1} = \mathbf{I}_{n \times n}$$

值得指出的是, 如果选择路线改为式 (6.15), 而不是式 (6.14), 则可导出另外一组不变量及其规范形。这两种选择路线下的规范形式分别是由两组独立不变量完备组所确定的, 因此, 它们

本质上是不同的。

6.2 输入/输出方程

我们的目的是要辨识状态方程的规范形式,从前面的讨论已经知道,只要辨识出结构指标及相应参数即可。但对辨识而言,除了一些先验知识外,可利用的信息只有输入、输出数据。因此,要得到结构指标和参数的估计,就要求出与规范形对应的输入、输出方程。

另外,从 6.1 节可以看到,状态方程模型的规范形是与“选择”有关的,但选择这个概念在数学推导中很难应用。因此,为了导出规范形所对应的输入、输出描述,我们引进与“选择”相应的数学描述——选择矩阵的概念。

定义 6.7 称 $m \times n$ 维($m \leq n$)矩阵 $E = (e_{ij})$ 为选择矩阵,如果 E 满足条件:

- ① $e_{ij} = 0$ 或 1, 对一切 i, j ;
- ② 对每一个 i , 有一个且只有 j_i , 满足 $e_{ij_i} = 1$;
- ③ 对每一个 j , 至多有一个 i_j , 满足 $e_{i_j j} = 1$ 。

上述定义即是说,选择矩阵的元素只能为 0 或 1; 每一行一定有一个元素,且只有一个元素为 1; 每一列最多有一个元素为 1。

【例 6.3】 矩阵

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

都是选择矩阵,而

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

都不是选择矩阵,前两个矩阵不满足条件②,最后一个不满足条件③。

从选择矩阵的定义可以看出,选择矩阵的行实际上都是某单位矩阵 I 的行。由选择矩阵的定义及矩阵的乘法法则可知,如果用 $m \times n$ 维的选择矩阵 E 作用于 $n \times p$ 维矩阵 G 得到 \tilde{G} , 即 $EG = \tilde{G}$, 实际上就是从矩阵 $G_{n \times p}$ 中选出 m 行,重新构成一个 $m \times p$ 维矩阵 \tilde{G} 。换句话说, $m \times p$ 维矩阵 \tilde{G} 的行都是从 $n \times p$ 维矩阵 G 的行中选出的。由此易知,对于式(6.1)描述的完全能观测系统 $\sigma(A, B, C, D)$ 的任一适宜选择 S , 如果用 S 所含的行向量构成矩阵(仍记为 S), 则一定对应一个 $n \times m, n$ 维选择矩阵 E , 使得

$$S = EF \quad (6.29)$$

其中, F 为系统 σ 的能观性矩阵(由式(6.5)所定义)。因此,在 6.1 节推导能观规范形过程中所构造的变换矩阵都可由相应的选择矩阵作用于系统的能观性矩阵得到。

有了选择矩阵的概念,就可以比较方便地推导出规范形对应的输入、输出描述。

6.2.1 输入/输出方程一般形式

设系统的状态方程为

$$x(k+1) = Ax(k) + Bu(k) \quad (6.30)$$

$$y(k) = Cx(k) + Du(k) \quad (6.31)$$

由式(6.30)、式(6.31)又可直接得到

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)$$

$$\begin{aligned}\mathbf{y}(k+1) &= \mathbf{C}\mathbf{x}(k+1) + \mathbf{D}\mathbf{u}(k+1) \\ &= \mathbf{C}[\mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)] + \mathbf{D}\mathbf{u}(k+1) \\ &= \mathbf{C}\mathbf{A}\mathbf{x}(k) + \mathbf{C}\mathbf{B}\mathbf{u}(k) + \mathbf{D}\mathbf{u}(k+1) \\ \mathbf{y}(k+2) &= \mathbf{C}\mathbf{x}(k+2) + \mathbf{D}\mathbf{u}(k+2) \\ &= \mathbf{C}[\mathbf{A}\mathbf{x}(k+1) + \mathbf{B}\mathbf{u}(k+1)] + \mathbf{D}\mathbf{u}(k+2) \\ &= \mathbf{C}\mathbf{A}[\mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)] + \mathbf{C}\mathbf{B}\mathbf{u}(k+1) + \mathbf{D}\mathbf{u}(k+2) \\ &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k) + \mathbf{C}\mathbf{A}\mathbf{B}\mathbf{u}(k) + \mathbf{C}\mathbf{B}\mathbf{u}(k+1) + \mathbf{D}\mathbf{u}(k+2) \\ &\dots\end{aligned}$$

$$\mathbf{y}(k+n-1) = \mathbf{C}\mathbf{A}^{n-1}\mathbf{x}(k) + \mathbf{C}\mathbf{A}^{n-2}\mathbf{B}\mathbf{u}(k) + \dots + \mathbf{D}\mathbf{u}(k+n-1)$$

上述各式可统一表示为如下形式

$$\bar{\mathbf{y}}(k) = \mathbf{F}\mathbf{x}(k) + \mathbf{R}\bar{\mathbf{u}}(k) \quad (6.32)$$

其中

$$\bar{\mathbf{y}}(k) = \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \\ \vdots \\ \mathbf{y}(k+n-1) \end{bmatrix}_{n \times m} \quad \bar{\mathbf{u}}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+n) \end{bmatrix}_{(n+1)r} \quad (6.33)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix}_{n \times m \times n} \quad \mathbf{R} = \begin{bmatrix} \mathbf{D} & 0 & \dots & \dots & 0 \\ \mathbf{C}\mathbf{B} & \mathbf{D} & 0 & \dots & 0 \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \mathbf{D} & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{n-2}\mathbf{B} & \dots & \mathbf{C}\mathbf{B} & \mathbf{D} & \dots & 0 \end{bmatrix}_{n \times m \times (n+1)r} \quad (6.34)$$

假定由某选择得到的行向量构成的变换矩阵为 \mathbf{T} , 使得式(6.30)和式(6.31)变成某种规范形式。由上面的讨论可知, 一定存在一个选择阵 \mathbf{E} , 使得 $\mathbf{T} = \mathbf{E}\mathbf{F}$ 。

将式(6.32)两端左乘以 \mathbf{E} , 然后再左乘以 \mathbf{T}^{-1} 并移项得

$$\mathbf{x}(k) = \mathbf{T}^{-1}\mathbf{E}\bar{\mathbf{y}}(k) - \mathbf{T}^{-1}\mathbf{E}\mathbf{R}\bar{\mathbf{u}}(k) \quad (6.35)$$

再将式(6.35)代入式(6.30)得

$$\mathbf{T}^{-1}\mathbf{E}\bar{\mathbf{y}}(k+1) - \mathbf{T}^{-1}\mathbf{E}\mathbf{R}\bar{\mathbf{u}}(k+1) = \mathbf{A}[\mathbf{T}^{-1}\mathbf{E}\bar{\mathbf{y}}(k) - \mathbf{T}^{-1}\mathbf{E}\mathbf{R}\bar{\mathbf{u}}(k)] + \mathbf{B}\mathbf{u}(k)$$

上式两端再左乘以 \mathbf{T} 并重新整理得到

$$\mathbf{E}\bar{\mathbf{y}}(k+1) = \mathbf{A}^*\mathbf{E}\bar{\mathbf{y}}(k) - \mathbf{A}^*\mathbf{E}\mathbf{R}\bar{\mathbf{u}}(k) + \mathbf{E}\mathbf{R}\bar{\mathbf{u}}(k+1) + \mathbf{B}^*\mathbf{u}(k) \quad (6.36)$$

其中 $\mathbf{A}^* = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$, $\mathbf{B}^* = \mathbf{T}\mathbf{B}$ 。

若令

$$\mathbf{B}^*\mathbf{u}(k) + \mathbf{E}\mathbf{R}\bar{\mathbf{u}}(k+1) \triangleq \tilde{\mathbf{E}}\bar{\mathbf{u}}(k) \quad (6.37)$$

则由于 $\mathbf{B}^*\mathbf{u}(k) = \mathbf{T}\mathbf{B}\mathbf{u}(k) = \mathbf{E}\mathbf{F}\mathbf{B}\mathbf{u}(k)$, 并注意到 \mathbf{R} , \mathbf{F} 和 $\bar{\mathbf{u}}(k+1)$ 的定义, 所以有

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{C}\mathbf{B} & \mathbf{D} & 0 & \dots & 0 \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \mathbf{D} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \mathbf{C}\mathbf{A}^{n-1}\mathbf{B} & \dots & \dots & \mathbf{C}\mathbf{B} & \mathbf{D} \end{bmatrix} \quad (6.38)$$

于是式(6.36)简化为

$$\bar{E}\bar{y}(k+1) = A^* \bar{E}\bar{y}(k) + \bar{Q}\bar{u}(k) \quad (6.39)$$

其中

$$Q = \tilde{E}\tilde{R} - A^*ER \quad (6.40)$$

式(6.39)和式(6.40)就是规范状态方程所对应的输入、输出方程的一般形式。

综合上述可得如下结论:

定理 6.4 对完全能观系统 $\sigma(A, B, C, D)$, 如果用可逆变换阵 $T=EF$ (E 为选择矩阵; F 为 σ 的能观性矩阵) 得到能观规范形 $\sigma^*(A^*, B^*, C^*, D^*)$, 其中 $A^* = TAT^{-1}$, $B^* = TB$, $C^* = CT^{-1}$, $D^* = D$, 则 σ^* 对应的输入、输出方程为

$$\bar{E}\bar{y}(k+1) = A^* \bar{E}\bar{y}(k) + \bar{Q}\bar{u}(k)$$

$$Q = \tilde{E}\tilde{R} - A^*ER$$

其中所有符号皆为前面所定义。

反之, 如果系统是完全能控完全能观的, 则 $\sigma^*(A^*, B^*, C^*, D^*)$ 就是其最小实现。

6.2.2 PCF 规范形对应的输入/输出方程

由定理 6.4 可知, 若要得到 PCF 规范形对应的输入、输出方程, 只要将其相应的 A^* , Q 和选择阵 E 直接代入式(6.39)和式(6.40)即可得到。

由得到 PCF 规范形所用的变换阵 T 的构成可知, 相应的选择阵 E 为

$$E = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{pmatrix}_{n \times m \cdot n} \quad E_j = \begin{pmatrix} e_j \\ e_{j+m} \\ \vdots \\ e_{j+(v_j-1)m} \end{pmatrix} \quad (6.41)$$

其中, e_j 是单位阵 $I_{m \cdot n}$ 的第 j 行。

故输入、输出方程为

$$\bar{E}\bar{y}(k+1) = A^* \bar{E}\bar{y}(k) + \bar{Q}\bar{u}(k) \quad (6.42)$$

$$Q = \tilde{E}\tilde{R} - A^*ER \quad (6.43)$$

其中, $\bar{y}(k)$ 和 $\bar{u}(k)$ 、 R 、 \tilde{R} 分别由式(6.33)、式(6.34)和式(6.38)所定义。

为了便于实际应用, 根据 σ^* 和 E 的特殊结构, 还可以将式(6.42)和式(6.43)进一步简化为更直观的通常差分方程形式。

记

$$B^* \triangleq \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{s_1} \\ b_{s_1+1} \\ \vdots \\ b_{s_2} \\ \vdots \\ b_{s_m} \end{pmatrix} \quad D^* \triangleq \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \quad (6.44)$$

式中, $S_j = \sum_{i=1}^j \nu_i, S_0 \equiv 0$, b_i 和 d_i 分别为矩阵 B^* 和 D^* 的第 i 行。

由 B^* 和 E 的定义显然有

$$B^* = TB = EFB = E \cdot \begin{pmatrix} CB \\ CAB \\ \vdots \\ CA^{n-1}B \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$E \cdot \begin{pmatrix} D \\ CB \\ \vdots \\ CA^{n-2}B \end{pmatrix} = \begin{pmatrix} d_1 \\ b_1 \\ \vdots \\ b_{S_1-1} \\ d_2 \\ b_{S_1+1} \\ \vdots \\ b_{S_2-1} \\ \vdots \\ b_{S_m-1} \end{pmatrix}, E \cdot \begin{pmatrix} 0 \\ D \\ CB \\ \vdots \\ CA^{n-3}B \end{pmatrix} = \begin{pmatrix} 0 \\ d_1 \\ b_1 \\ \vdots \\ b_{S_1-2} \\ 0 \\ d_2 \\ \vdots \\ b_{S_2-2} \\ \vdots \\ b_{S_m-2} \end{pmatrix}, \dots$$

于是

$$E\tilde{R} = E \cdot \begin{pmatrix} CB & D & & & \\ CAB & CB & D & & \\ \vdots & & \ddots & \ddots & \\ CA^{n-1} & \dots & & CB & D \end{pmatrix}$$

$$= \begin{pmatrix} b_1 & d_1 & & & & & \\ \vdots & \ddots & \ddots & & & & \\ b_{S_1} & \dots & b_1 & d_1 & 0 & \dots & 0 \\ b_{S_1+1} & d_2 & & & & & \\ \vdots & \ddots & \ddots & & & & \\ b_{S_2} & \dots & b_{S_1+1} & d_2 & 0 & \dots & 0 \\ \vdots & & \vdots & & & & \\ b_{S_{m-1}+1} & d_m & & & & & \\ \vdots & \ddots & \ddots & & & & \\ b_{S_m} & \dots & b_{S_{m-1}+1} & d_m & 0 & \dots & 0 \end{pmatrix} \quad (6.45)$$

$$ER = E \cdot \begin{pmatrix} D & 0 & \dots & \dots & \dots & \dots & 0 \\ CB & D & 0 & \dots & \dots & \dots & 0 \\ CAB & CB & D & \vdots & & & \\ \vdots & & \ddots & \ddots & & & \\ CA^{n-2}B & \dots & CB & D & 0 & \dots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} d_1 & & & & & & & \\ b_1 & d_1 & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ d_{s_1-1} & \cdots & b_1 & d_1 & 0 & \cdots & 0 & \\ d_2 & & & & & & & \\ b_{s_1+1} & d_2 & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ b_{s_2-1} & \cdots & b_{s_1+1} & d_2 & 0 & \cdots & 0 & \\ \vdots & & & & & & & \\ d_m & & & & & & & \\ b_{s_{m-1}+1} & d_m & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ b_{s_m-1} & \cdots & d_{s_{m-1}+1} & d_m & 0 & \cdots & 0 & \end{pmatrix} \quad (6.46)$$

注意到 A^* (由式 6.24) 定义) 的特殊结构, 直接进行矩阵乘法得

$$A^* ER = \begin{pmatrix} b_1 & d_1 & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ b_{s_1} & \cdots & b_1 & d_1 & 0 & \cdots & 0 & \\ g_{11} & g_{12} & & & & & & \\ b_{s_1+1} & d_2 & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ b_{s_2-1} & \cdots & b_{s_1+1} & d_2 & 0 & \cdots & 0 & \\ g_{21} & g_{22} & & & & & & \\ \vdots & & \vdots & & & & & \\ b_{s_{m-1}+1} & d_m & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ b_{s_m-1} & \cdots & b_{s_{m-1}+1} & d_m & 0 & \cdots & 0 & \\ g_{m1} & g_{m2} & \cdots & & & & & \end{pmatrix} \quad (6.47)$$

其中 g_{ij} 为 r 维行向量, 由下式所确定

$$g_{ij} = \begin{cases} \sum_{l=1}^m \alpha_{il,j} d_l + \sum_{l=1}^m \sum_{t=j+1}^{\nu_{il}} \alpha_{il,t} b_{s_{l-1}+t-j} & j = 1, 2, \dots, \nu_i \end{cases} \quad (6.48)$$

$$g_{ij} = \begin{cases} \sum_{l=1}^{i-1} \alpha_{il, \nu_i+1} d_l & j = \nu_i + 1 \end{cases} \quad (6.49)$$

$$g_{ij} = \begin{cases} 0 & j > \nu_i + 1 \end{cases} \quad (6.50)$$

$$i = 1, 2, \dots, m, j = 1, 2, \dots, n$$

将式(6.45)和式(6.47)代入式(6.43)可知, Q 中只有 m 行不恒为零, 这 m 行就是第 s_1, s_2, \dots, s_m 行。把这 m 行按原顺序重新构成新矩阵 Q^* , 即 $Q^* = (q_{i,j}^*) (i=1, 2, \dots, m, j=1, 2, \dots, n)$ 。则由式(6.45)、式(6.47)及 g_{ij} 的定义式(6.48)~式(6.50)易得

$$q_{ij}^* = b_{s_i-j+1} - \sum_{l=1}^m \alpha_{il,j} d_l - \sum_{l=1}^m \sum_{t=j+1}^{\nu_{il}} \alpha_{il,t} b_{s_{l-1}+t-j} \quad (6.51)$$

$$i = 1, 2, \dots, m \quad j = 1, 2, \dots, \nu_i$$

$$q_{\nu_i+1}^* = d_i - \sum_{l=1}^{i-1} \alpha_{il,\nu_i+1} d_l \quad i = 1, 2, \dots, m \quad (6.52)$$

$$q_{ij}^* = 0 \quad j > \nu_i + 1 \quad i = 1, 2, \dots, m \quad (6.53)$$

其中, q_{ij}^* 是 r 维行向量, b_i, d_i 分别为矩阵 B^* 和 D^* 的第 i 个行向量。

另一方面, 由 E 和 A^* 的特殊形式, 直接运算可知, 向量 $E\bar{y}(k+1) - A^*E\bar{y}(k)$ 也只有 m 个元素不恒为零, 而且也恰好为第 s_1, s_2, \dots, s_m 个元素, 且第 i 个非零元素为

$$y_i(k + \nu_i) - \sum_{l=1}^m \sum_{j=1}^{\nu_{il}} \alpha_{il,j} y_l(k + j - 1) \quad (6.54)$$

其中 $y_i(\cdot)$ 为向量 $y(\cdot)$ 的第 i 个分量。

因此, 在向量等式 (6.42) 中, 只有第 s_1, s_2, \dots, s_m 行为非平凡行。于是式 (6.42) 和式 (6.43) 可简化为

$$\begin{aligned} y_i(k + \nu_i) &= \sum_{l=1}^m \sum_{j=1}^{\nu_{il}} \alpha_{il,j} y_l(k + j - 1) + \sum_{j=1}^{\nu_i+1} q_{ij}^* u(k + j - 1) \\ &= \sum_{l=1}^m \sum_{j=1}^{\nu_{il}} \alpha_{il,j} y_l(k + j - 1) + \sum_{j=1}^{\nu_i+1} \sum_{l=1}^r q_{ij,l} u_l(k + j - 1) \end{aligned} \quad (6.55)$$

$$i = 1, 2, \dots, m$$

式中, $q_{ij,l}$ 为 r 维行向量 q_{ij}^* 的第 l 个分量; $u_l(\cdot)$ 为 r 维输入向量 $u(\cdot)$ 的第 l 个分量; $y_l(\cdot)$ 为 m 维输出向量 $y(\cdot)$ 的第 l 个分量 (上式第一个等号利用了 $q_{ij}^* = 0, j > \nu_i + 1$)。

式 (6.55) 即是 PCF 规范形所对应的输入、输出差分方程, 这是我们所熟悉的便于应用的一种形式。

很显然, 当系统的量测方程不包含控制项时 (即系统模型中不含 D), 即对于系统 $\sigma^*(A^*, B^*, C^*)$, 则式 (6.51)、式 (6.52) 和式 (6.53) 可简化为

$$q_{ij}^* = \begin{cases} b_{s_i-j+1} - \sum_{l=1}^m \sum_{t=j+1}^{\nu_{il}} \alpha_{il,t} b_{s_{l-1}+t-j} & j = 1, 2, \dots, \nu_i \\ 0 & j \geq \nu_i + 1, \quad i = 1, 2, \dots, m \end{cases} \quad (6.56)$$

因此, 输入、输出方程式 (6.55) 相应地变为

$$y_i(k + \nu_i) = \sum_{l=1}^m \sum_{j=1}^{\nu_{il}} \alpha_{il,j} y_l(k + j - 1) + \sum_{j=1}^{\nu_i} \sum_{l=1}^r q_{ij,l} u_l(k + j - 1) \quad (6.57)$$

其中, $\nu_{ii} = \nu_i, i = 1, 2, \dots, m$ 。

6.3 PCF 规范形的辨识

关于 PCF 规范形辨识的基本步骤分为两步: 首先利用输入、输出方程式 (6.55) 辨识系统的结构指标及参数, 然后再求状态实现——PCF 规范形。在具体讨论结构和参数辨识及实现方法时, 又分系统方程中包含 D 和不包含 D 两种情况, 重点讨论方程中不包含 D 的情况。对于包含 D 的情况, 仅指出与前者区别及相应的方法即可。

6.3.1 结构确定及参数辨识

考虑 PCF 规范形(不包含 D)

$$\mathbf{x}(k+1) = \mathbf{A}^* \mathbf{x}(k) + \mathbf{B}^* \mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}^* \mathbf{x}(k)$$

其中 $\mathbf{A}^*, \mathbf{C}^*$ 分别由式(6.24)、式(6.25)所定义。

由式(6.57)知,第 i 个子系统的输入、输出方程为

$$y_i(k + \nu_i) = \sum_{l=1}^m \sum_{j=1}^{\nu_{il}} \alpha_{il,j} y_l(k + j - 1) + \sum_{l=1}^r \sum_{j=1}^{\nu_{il}} q_{ij,l} u_l(k + j - 1) \quad (6.58)$$

其中 $\nu_{ii} = \nu_i$ 。

由此可见,如果第 i 个子系统的结构指标为 ν_i ,则输出 $y_i(k + \nu_i)$ 可由 k 至 $k + \nu_i - 1$ 时刻的输入 $u(\cdot)$ 输出 $y(\cdot)$, 以及 $(k + \nu_i)$ 时刻的输出 $y_j(k + \nu_i) (j < i)$ 线性表出,这就是辨识结构指标 $\{\nu_i\}$ 的根据。

从式(6.58)及辨识所依据的原理不难发现,下面所介绍的辨识方法实际上就是 5.2 节所介绍方法在多变量线性系统辨识中的推广,只是符号描述复杂一些而已。

1. 用相关方法确定结构指标(Guidorzi, 1975)

假如取得输入(r 维)、输出(m 维)数据 $\mathbf{u}(k), \mathbf{u}(k+1), \mathbf{u}(k+2), \dots, \mathbf{u}(k+N), \dots, \mathbf{y}(k), \mathbf{y}(k+1), \dots, \mathbf{y}(k+N), \dots$, 将其排成如下形式的矩阵

$$\mathbf{H} = \begin{bmatrix} y_1(k) & y_1(k+1) & \cdots & y_m(k) & y_m(k+1) & \cdots & u_1(k) & u_1(k+1) & \cdots & u_r(k) & u_r(k+1) & \cdots \\ y_1(k+1) & y_1(k+2) & \cdots & y_m(k+1) & y_m(k+2) & \cdots & u_1(k+1) & u_1(k+2) & \cdots & u_r(k+1) & u_r(k+2) & \cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots \\ y_1(k+N) & y_1(k+N+1) & \cdots & y_m(k+N) & y_m(k+N+1) & \cdots & u_1(k+N) & u_1(k+N+1) & \cdots & u_r(k+N) & u_r(k+N+1) & \cdots \end{bmatrix}$$

$$\triangleq [\bar{y}_1(k) \bar{y}_1(k+1) \cdots \bar{y}_m(k) \bar{y}_m(k+1) \cdots \bar{u}(k) \bar{u}_1(k+1) \cdots \bar{u}_r(k) \bar{u}_r(k+1) \cdots] \quad (6.59)$$

为了叙述方便,把输入和输出的每个分量组成的矩阵称为矩阵 \mathbf{H} 的一个子矩阵。利用上面的记号,则式(6.58)就意味着 $\bar{y}_i(k + \nu_i)$ 是 $\bar{y}_1(k + \nu_i), \dots, \bar{y}_{i-1}(k + \nu_i), \bar{y}_i(k + \nu_i - 1), \dots, \bar{y}_m(k + \nu_i - 1), \dots, \bar{y}_1(k), \dots, \bar{y}_m(k), \bar{u}_1(k + \nu_i - 1), \dots, \bar{u}_r(k + \nu_i - 1), \dots, \bar{u}_1(k), \dots, \bar{u}_r(k)$ 的线性组合。因此,可以利用这个事实,通过适当的方式选择输入、输出 $\bar{u}_i(\cdot), \bar{y}_i(\cdot)$ 就可以确定出结构指标 $\nu_i, i = 1, 2, \dots, m$ 。

在式(6.59)中,按如下顺序选择线性无关向量

$$\begin{aligned} & \bar{y}_1(k), \quad \bar{y}_2(k), \quad \cdots, \quad \bar{y}_m(k), \quad \bar{u}_1(k), \quad \bar{u}_2(k), \quad \cdots, \quad \bar{u}_r(k), \\ & \bar{y}_1(k+1), \bar{y}_2(k+1), \cdots, \bar{y}_m(k+1) \quad \bar{u}_1(k+1), \bar{u}_2(k+1), \cdots, \bar{u}_r(k+1), \bar{y}_1(k+2) \cdots \end{aligned} \quad (6.60)$$

当找到某向量 $\bar{y}_i(k + \nu_i)$ 与前面所选出的向量线性相关时(这时 $\bar{y}_i(k + \nu_i)$ 所在的子阵中其他剩余的向量也与所选出的向量线性相关),这就得到了 ν_i 。自然, $\bar{y}_i(k + \nu_i)$ 也不被选出。继续按式(6.60)给定的顺序选择,直到关于输出的每个子阵都找到这样的向量,选择结束。分别从关于输出的每个子阵中所选出的向量个数就是所求的结构指标 $\nu_1, \nu_2, \dots, \nu_m$ 。这最后一点,可由

选择顺序式(6.60)及式(6.58)直接可以看出。

注意:为了充分保证无关向量的选择,向量 $\bar{\mathbf{u}}_i(\cdot)$, $\bar{\mathbf{y}}_i(\cdot)$ 的长度必须充分大,即

$$N+1 > n + r\nu_M$$

其中

$$\nu_M = \max\{\nu_i\}, n = \nu_1 + \nu_2 + \cdots + \nu_m \quad (6.61)$$

由上面分析指出,确定结构指标的问题,实际上变为判断向量组是否相关的问题。为了判断向量的相关性,引入记号 $\mathbf{R}(\delta_1, \delta_2, \cdots, \delta_{m+r})$,此记号表示从矩阵 \mathbf{H} 的第一个子阵中取出前 δ_1 个列向量,……,从第 $m+r$ 个子阵中取出前 δ_{m+r} 个列向量构成的矩阵。即

$$\begin{aligned} \mathbf{R}(\delta_1, \cdots, \delta_m, \delta_{m+1}, \cdots, \delta_{m+r}) \\ = [\bar{\mathbf{y}}_1(k) \cdots \bar{\mathbf{y}}_1(k + \delta_1 - 1) \cdots \bar{\mathbf{y}}_m(k) \cdots \bar{\mathbf{y}}_m(k + \delta_m - 1) \\ \bar{\mathbf{u}}_1(k) \cdots \bar{\mathbf{u}}_1(k + \delta_{m+1} - 1) \cdots \bar{\mathbf{u}}_m(k) \cdots \bar{\mathbf{u}}_m(k + \delta_{m+r} - 1)] \end{aligned} \quad (6.62)$$

那么,结构指标可能通过对 $\delta_i (i=1, 2, \cdots, m+r)$ 的不同取值来判断 \mathbf{R} 的列相关性来确定。为了给出方便的判别法,利用如下众所周知的事实:对任何矩阵 \mathbf{D} , $\text{rank}(\mathbf{D}) = \text{rank}(\mathbf{D}^T \mathbf{D})$ 。

定义

$$\Phi(\delta_1, \delta_2, \cdots, \delta_{m+r}) = \mathbf{R}^T(\delta_1, \delta_2, \cdots, \delta_{m+r}) \mathbf{R}(\delta_1, \delta_2, \cdots, \delta_{m+r}) \quad (6.63)$$

这时,矩阵 $\mathbf{R}(\delta_1, \delta_2, \cdots, \delta_{m+r})$ 列相关充分必要矩阵 $\Phi(\delta_1, \delta_2, \cdots, \delta_{m+r})$ 的行列式为零。因此,确定结构指标问题转变为依次考察矩阵序列

$$\begin{aligned} \Phi(1, 1, \cdots, 1), \Phi(2, 1, \cdots, 1), \Phi(2, 2, 1, \cdots, 1), \Phi(2, 2, \cdots, 2), \\ \Phi(3, 2, \cdots, 2) \cdots \end{aligned}$$

是否满秩的问题。当发现 $\Phi(\delta_1, \cdots, \delta_i, \delta_{i+1}, \cdots, \delta_{r+m})$ 非奇异($i \leq m$),而 $\Phi(\delta_1, \cdots, \delta_i + 1, \delta_{i+1}, \cdots, \delta_{r+m})$ 奇异时,这说明矩阵 \mathbf{H} 中第 i 个子阵的第 $\delta_i + 1$ 个向量与前面所选出的向量线性相关。由上面的分析及式(6.58)可见,第 i 个结构指标 $\nu_i = \delta_i$,并且 $\nu_{ij} = \delta_j (j=1, 2, \cdots, m, j \neq i)$ 。如此一直继续下去,就可得到全部 $\nu_i (i=1, 2, \cdots, m)$ 。

结构指标确定后,我们可用最小二乘法进行参数估计

$$\hat{\boldsymbol{\theta}}_i = \Phi_i^{-1} \mathbf{R}_i^T \bar{\mathbf{y}}_i(k + \nu_i) \quad (6.64)$$

其中

$$\begin{aligned} \hat{\boldsymbol{\theta}}_i^T = [\alpha_{i1,1} \alpha_{i1,2} \cdots \alpha_{i1,\nu_{i1}} : \cdots : \alpha_{i\bar{i},1} \alpha_{i\bar{i},2} \cdots \alpha_{i\bar{i},\nu_{i\bar{i}}} : \cdots : \alpha_{im,1} \alpha_{im,2} \cdots \alpha_{im,\nu_{im}} \\ : q_{i1,1} q_{i2,1}, \cdots q_{i\nu_i,1} : \cdots : q_{i1,r} q_{i2,r} \cdots q_{i\nu_i,r}] \end{aligned} \quad (6.65)$$

$$\mathbf{R}_i = \mathbf{R}(\nu_{i1}, \nu_{i2}, \cdots, \nu_{i\bar{i}-1}, \nu_{i\bar{i}+1}, \cdots, \nu_{im}, \nu_i, \cdots, \nu_i) \quad (6.66)$$

$$\Phi_i = \mathbf{R}_i^T \mathbf{R}_i \quad (6.67)$$

为了说明计算步骤,下面给出一个简单的数值例子。

【例 6.3】假定从单输入、双输出的系统取得如下数据,以此辨识系统的结构和参数。

$k:$	0	1	2	3	4	5	6	7	8	9	10	11	12
$u(k):$	1	2	4	5	-5	-12	15	50	10	-60	30	0	0
$y_1(k):$	0	0	1	0	2	5	-3	5	-25	56	-3	-27	40
$y_2(k):$	0	0	0	2	-1	4	-9	10	8	-7	10	-4	5

解 第一步：结构参数的确定

取 $N=8$, 依据上面的数据构造矩阵 H

$$H = \begin{bmatrix} \bar{y}_1(0) & \bar{y}_1(1) & \bar{y}_1(2) & \cdots & \bar{y}_2(0) & \bar{y}_2(1) & \bar{y}_2(2) & \cdots & \bar{u}(0) & \bar{u}(1) & \bar{u}(2) & \cdots \end{bmatrix}$$

$$H = \begin{pmatrix} 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 1 & 2 & 4 & \cdots \\ 0 & 1 & 0 & \cdots & 0 & 0 & 2 & \cdots & 2 & 4 & 5 & \cdots \\ 1 & 0 & 2 & \cdots & 0 & 2 & -1 & \cdots & 4 & 5 & -5 & \cdots \\ 0 & 2 & 5 & \cdots & 2 & -1 & 4 & \cdots & 5 & -5 & -12 & \cdots \\ 2 & 5 & -3 & \cdots & -1 & 4 & -9 & \cdots & -5 & -12 & 15 & \cdots \\ 5 & -3 & 5 & \cdots & 4 & -9 & 10 & \cdots & -12 & 15 & 50 & \cdots \\ -3 & 5 & -25 & \cdots & -9 & 10 & 8 & \cdots & 15 & 50 & 10 & \cdots \\ 5 & -25 & 56 & \cdots & 10 & 8 & -7 & \cdots & 50 & 10 & -60 & \cdots \\ -25 & 56 & -3 & \cdots & 8 & -7 & 10 & \cdots & 10 & -60 & 30 & \cdots \end{pmatrix}$$

再构造矩阵序列

$$\Phi(1,1,1), \Phi(2,1,1), \Phi(2,2,1), \Phi(2,2,2), \Phi(3,2,2) \cdots$$

然后, 依次判别以上矩阵是否满秩, 经过 5 次判别其行列式的值, 得到第一个奇异阵为 $\Phi(3,3,2)$, 所以 $\nu_2 = 3 - 1 = 2, \nu_{21} = 3$; 在此基础上 (这时 $\Phi(\cdot, \cdot, \cdot)$ 中第二个标号应固定为 2), 继续判别 $\Phi(3,2,3), \Phi(4,2,3), \cdots$, 得到第二个奇异矩阵为 $\Phi(4,2,3)$, 从而 $\nu_1 = 4 - 1 = 3, \nu_{12} = 2$ 。由于 $m=2$, 所以结构指标完全确定。

第二步：参数估计

由式 (6.58) 第一个子系统的输入、输出差分方程为

$$y_1(k+3) = \alpha_{11,1}y_1(k) + \alpha_{11,2}y_1(k+1) + \alpha_{11,3}y_1(k+2) + \alpha_{12,1}y_2(k) + \alpha_{12,2}y_2(k+1) + q_{11,1}u(k) + q_{12,1}u(k+1) + q_{13,1}u(k+2)$$

参数的最小二乘估计为

$$\begin{aligned} \hat{\theta}_1 &= (\alpha_{11,1}, \alpha_{11,2}, \alpha_{11,3}, \alpha_{12,1}, \alpha_{12,2}, q_{11,1}, q_{12,1}, q_{13,1})^T \\ &= \Phi_1^{-1} R_1^T \bar{y}_1(4) \\ &= (1 \ 0 \ -1 \ -1 \ 1 \ 1 \ 0 \ 0)^T \end{aligned}$$

其中 $R_1 = R(3,2,3), \Phi_1 = R_1^T R_1$ 。

第二个子系统的差分方程为

$$y_2(k+2) = \alpha_{21,1}y_1(k) + \alpha_{21,2}y_1(k+1) + \alpha_{21,3}y_1(k+2) + \alpha_{22,1}y_2(k) + \alpha_{22,2}y_2(k+1) + q_{21,1}u(k) + q_{22,1}u(k+1)$$

参数的最小乘估计为

$$\begin{aligned} \hat{\theta}_2 &= (\alpha_{21,1}, \alpha_{21,2}, \alpha_{21,3}, \alpha_{22,1}, \alpha_{22,2}, q_{21,1}, q_{22,1})^T \\ &= \Phi_2^{-1} R_2^T \bar{y}_2(3) \\ &= (1 \ 0 \ -1 \ 1 \ -2 \ 1 \ 0 \ 0)^T \end{aligned}$$

其中 $R_2 = R(3,2,2), \Phi_2 = R_2^T R_2$ 。

值得注意的是, 在实际应用上述方法确定结构指标时, 是判断一系列矩阵 Φ 的行列式是否为零。正如在 5.2 节所分析的那样, 由于数据的不精确性、计算的舍入误差及非严格的线性相关性, 既使在确定性情况下, Φ 的行列式也不会严格为零。因此, 需给出一些其他判别方法 (见 5.2 节中的说明)。另外, 也常常采用作图方法, 看行列式值是否有陡降变化来决定其结构指标。

解决上述问题的另一种考虑是用矩阵 Φ 的最小特征根是否接近于零来判别。由线性代数的理论知道, 矩阵行列式的值等于其所有特征根之积, 从而得到行列式值等于零的充分必要条件为: 有一个特征值为零; 又矩阵 Φ 是半正定的, 所以其特征值皆大于或等于零。因此, 可用其最小特征值 λ_{\min} 来判别矩阵 Φ 的奇异性。实践表明, λ_{\min} 的值随矩阵 Φ 的阶的增加呈现单调下降的趋势(这与向量增加而相关性增强是一致的), 而向量组变为相关时, λ_{\min} 的值也明显表现由平坦下降变为陡峭下降, 所以, 可用作图方法确定其结构指标 (Goidorzi 1975)。

2. 递推算法

Guidorzi 方法的最大优点是不需要对系统的结构进行试探性的检验, 由输入、输出数据直接确定。它的缺点是计算量太大, 每确定一个性能指标就需要多次判断矩阵的奇异性, 这就花费了大量的时间。为了克服这个缺点, 如 5.2 节完全类似地推导, 可给出结构确定及参数辨识的递推算法。

为了得到递推算法, 需重新构造矩阵 R , 令 $R'(\delta_1, \delta_2, \dots, \delta_{m+r})$ 表示按式 (6.60) 指定的顺序选出的向量且按先后次序排列构成的矩阵。 δ_i 只是表示在矩阵 R' 中包含了从矩阵 H 的第 i 个子阵中选出的前 δ_i 个列向量 ($i=1, 2, \dots, m+r$)。

例如

$$\begin{aligned} R'(1, 1, \dots, 1) &= [\bar{y}_1(k) \bar{y}_2(k) \dots \bar{y}_m(k) \bar{u}_1(k) \bar{u}_2(k) \dots \bar{u}_r(k)] \\ R'(2, 1, \dots, 1) &= [\bar{y}_1(k) \bar{y}_2(k) \dots \bar{y}_m(k) \bar{u}_1(k) \bar{u}_2(k) \dots \bar{u}_r(k) \bar{y}_1(k+1)] \\ &\triangleq [R'(1, 1, \dots, 1) \quad \bar{y}_1(k+1)] \\ &\dots \end{aligned}$$

一般地

$$\begin{aligned} R'(\mu_1, \dots, \mu_i + 1, \mu_{i+1}, \mu_{m+r}) \\ = [R'(\mu_1, \dots, \mu_i, \mu_{i+1}, \dots, \mu_{m+r}) \quad \bar{y}_i(k + \mu_i)] \quad i \leq m \end{aligned} \quad (6.68)$$

为了方便起见, 在下面的递推算法中, 我们将采用如下的简化记号, 令 R_i 表示递推第 i 步时的矩阵 R' , 即

$$R_i = R'(\mu_1, \mu_2, \dots, \mu_{m+r})$$

而用 R_{i+1} 表示下一步的矩阵 R' , 即

$$R_{i+1} = [R'(\mu_1, \mu_2, \dots, \mu_{m+r}) \quad \bar{y}_{j, \mu_j}] \triangleq [R_i \quad \bar{y}_{j, \mu_j}] \quad (6.69)$$

其中

$$\bar{y}_{j, \mu_j} = \begin{cases} \bar{y}_j(k + \mu_j) & \text{当 } j = 1, 2, \dots, m \text{ 时} \\ \bar{u}_{j-m}(k + \mu_j) & \text{当 } j = m+1, \dots, m+r \text{ 时} \end{cases} \quad (6.70)$$

注意, 这里的 R_i 与式 (6.62) 定义的 R_i 其列向量的排列顺序是不一样的。

令

$$\begin{aligned} \Phi_i &= R_i^T R_i \\ \Phi_{i+1} &= R_{i+1}^T R_{i+1} = [R_i \quad \bar{y}_{j, \mu_j}]^T [R_i \quad \bar{y}_{j, \mu_j}] \\ &= \begin{bmatrix} R_i^T R_i & R_i^T \bar{y}_{j, \mu_j} \\ \bar{y}_{j, \mu_j}^T R_i & \bar{y}_{j, \mu_j}^T \bar{y}_{j, \mu_j} \end{bmatrix} \end{aligned} \quad (6.71)$$

如 5.2 节的推导一样, 利用引理 5.1 和定理 5.1 的结果, 可得到结构及参数辨识的递推算法。从 $\Phi_1 = [R'(1, 0, \dots, 0)]^T \cdot R'(1, 0, \dots, 0)$ 出发, 就可以递推地确定全部结构指标及所有

参数。当 $\bar{y}_j(k + \mu_j)$ 添加到矩阵 R' 中,使得 $\det(R'^T R') = 0$ 时,就得到 $\nu_j = \mu_j (1 \leq j \leq m)$, $\nu_{jk} (k=1, 2, \dots, m)$ 及

$$\hat{\theta}_j = \Phi_i^{-1} R_i^T \bar{y}_j(k + \mu_j) \quad (6.72)$$

$$\hat{\theta}_j^T = [\alpha_{j1,1} \alpha_{j2,1} \dots \alpha_{jm,1} : q_{j1,1} q_{j1,2} \dots q_{j1,r} : \dots] \quad (6.73)$$

然后去掉 $\bar{y}_j(k + \mu_j)$,再按照式(6.60)给定的顺序将其他向量添加到 R' 中,继续以上步骤,直到全部确定 ν_1, \dots, ν_m 及 $\hat{\theta}_1, \dots, \hat{\theta}_m$ 为止。

提醒读者注意,这里得到的参数排列顺序(如式(6.73)所示)与式(6.65)给出的参数排列顺序是不一样的。

为了更确切地描述算法的具体计算步骤,特引入如下记号:

$y(t, l)$ ——表示矩阵 H 中第 t 个子阵的第 l 列;

$KK(i)$ ——表示从第 i 个子阵中取出的向量个数($i=1, 2, \dots, m+r$);

$\{I_0(i); i=1, 2, \dots, m+r\}$ 是一个标志数组。当 $I_0(i)=1$ 时,表示 ν_i 已确定(不能再从第 i 个子阵中取向量);当 $I_0(i)=0$ 时,表示 ν_i 还未确定,可以从第 i 个子阵中取向量。

利用上述记号给出具体计算步骤如下:

- (1) 置 $i=1; t=2; l=1; KK(1)=1; KK(2)=1;$
 $KK(3)=\dots=KK(m+r)=0; I_0(1)=\dots=I_0(m+r)=0;$
 $R_1 = y(1, 1); \Phi_1^{-1} = 1/y^T(1, 1)y(1, 1)。$
- (2) 取向量 $y(t, l)。$
- (3) 计算

$$\hat{\theta}_t = \Phi_i^{-1} R_i^T y(t, l)$$

$$\eta = y(t, l) - R_i \hat{\theta}_t$$

$$p = \eta^T \eta$$

如果 $t > m$, 则转步骤(5), 否则继续。

- (4) 如果 $|p| < \varepsilon$, 则 $\hat{\theta}_t$ 即为所求参数; $KK(t)-1 \Rightarrow KK(t), \nu_i = KK(t), \nu_{ij} = KK(j), j=1, 2, \dots, m, j \neq t$; 置 $I_0(t)=1$; 转步骤(6)。

- (5) 构造 $R_{i+1} = [R_i \ y(t, l)]$

$$\text{计算 } \Phi_{i+1}^{-1} = \begin{bmatrix} \Phi_i^{-1} + \hat{\theta}_t \hat{\theta}_t^T / p & -\hat{\theta}_t / p \\ -\hat{\theta}_t^T / p & 1/p \end{bmatrix}$$

- (6) 如果 $\sum_{j=1}^m I_0(j) < m$, 则 $i+1 \Rightarrow i$; 并按顺序: $k=t+1, t+2, \dots, r+m, 1, 2, \dots, t$, 考

查 $I_0(k)$, 当有某 $I_0(j)=0$, 则 $j \Rightarrow t$; 并置 $KK(t)+1 \Rightarrow KK(t) \Rightarrow l$, 转步骤(2)。如果 $\sum_{j=1}^m I_0(j) = m$, 则结束。

根据以上步骤, 给出计算框图, 如图 6.2 所示。

应该指出, 在 5.2 节中对算法的几点说明, 在这里仍适用, 不再重述。

采用多变量线性系统辨识的递推算法, 确定结构指标的整个过程中不需直接矩阵求逆和求行列式值的运算, 而且同时得到结构指标和参数估计, 大大减少运算时间, 简单易行。就拿例 6.3 来说, 对这样一个单输入、双输出的简单系统, 用非递推的方法确定其结构就需计算 7

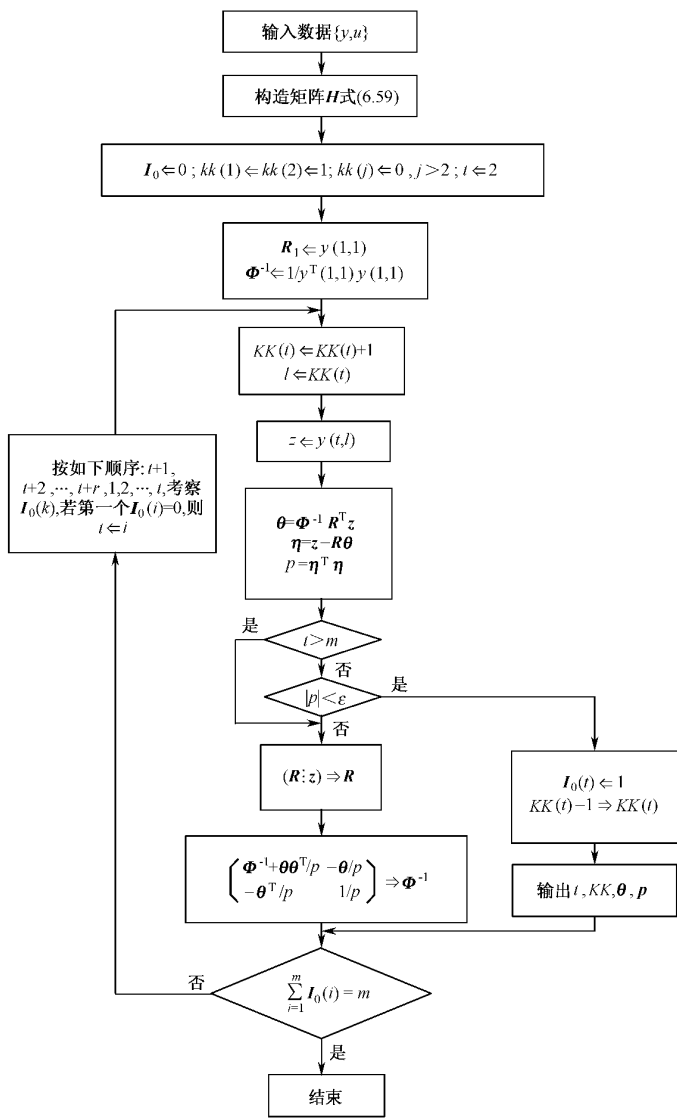


图 6.2 多变量线性系统递推辨识算法框图

次行列式的值,其最高阶次达到 9 阶,对于变量较多的系统,递推算法更显示其优越性。

3. 数据包含随机噪声的情况

在这里只考虑在量测噪声的情况,这是实际中最常见的情形。假定 $y_j^*(k)$ 是实测的含有噪声的数据,即

$$y_j^*(k) = y_j(k) + \epsilon_j(k)$$

$$u_j^*(k) = u_j(k) + \eta_j(k)$$

其中 $\epsilon_j(k)$, $\eta_j(k)$ 是与 $y_j(k)$, $u_j(k)$ 不相关的零均值的随机噪声,在工程上一般总可以认为它是平稳遍历的。这种情况下,最小二乘估计将是有偏的。但是,只要已知噪声的统计特性,就

可以将这一偏差补偿掉,得到参数的相容估计量。

另一个要解决的问题是如何从有噪声的数据中得到结构辨识所必须的信息 Φ_i 。由于对噪声特性所做的假定,有(Guidorzi, 1975)

$$P \lim_{N \rightarrow \infty} \frac{1}{N} \Phi_i^* = \lim_{N \rightarrow \infty} \frac{1}{N} \Phi_i + Q_i$$

其中, Q_i 是噪声向量

$$[\epsilon_1(k), \epsilon_1(k+1), \dots, \epsilon_1(k+\nu_{i1}) : \dots : \eta_r(k), \eta_r(k+1), \dots, \eta_r(k+\nu_i-1)]^T$$

的协方差矩阵,同时有

$$P \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{R}_i^{*T} \mathbf{y}_i^*(k+\nu_i) = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{R}_i^T \mathbf{y}_i(k+\nu_i) + q_i$$

其中

$$q_i = \lim_{N \rightarrow \infty} \frac{1}{N} \left[\sum_{j=k}^N \epsilon_1(j) \epsilon_1(j+\nu_i) \dots \sum_{j=k}^N \epsilon_1(j+\nu_{i1}-1) \epsilon_1(j+\nu_i), \dots, \right. \\ \left. \sum_{j=k}^N \eta_r(j+\nu_i-1) \epsilon_1(j+\nu_i) \right]$$

如果已知噪声的统计特性,即 Q_i, q_i 已知,就可以得到参数的相容估计量,即

$$\hat{\Phi}_i = \Phi_i^* - N Q_i$$

$$\hat{\mathbf{R}}_i^T \hat{\mathbf{y}}_i(k+\nu_i) = \mathbf{R}_i^T \mathbf{y}_i^*(k+\nu_i) - N q_i$$

经过这样补偿后的最小二乘估计为

$$\hat{\theta}_i = \hat{\Phi}_i^{-1} \hat{\mathbf{R}}_i^T \hat{\mathbf{y}}_i(k+\nu_i)$$

将是参数的相容估计量。结构确定可利用矩阵 $\hat{\Phi}(\mu_1, \dots, \mu_{m+r})$ 。以上事实的详细推证可参见 Bonivento 等(1971)的文章。

下面考虑实际中最便于应用的情形。即假定各子系统的噪声互不相关、零均值且有相同方差 σ^2 , 即 $Q_i = \sigma^2 I, q_i = 0$ 。在这种情况下,上面讨论的关系式可简化为

$$P \lim_{N \rightarrow \infty} \frac{1}{N} \Phi_i^* = \lim_{N \rightarrow \infty} \frac{1}{N} \Phi_i + \sigma^2 I$$

$$P \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{R}_i^{*T} \mathbf{y}_i^*(k+\nu_i) = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{R}_i^T \mathbf{y}_i(k+\nu_i)$$

所得到的参数估计应为

$$\hat{\theta}_i = (\Phi_i^* - N\sigma^2 I)^{-1} \mathbf{R}_i^{*T} \mathbf{y}_i^*(k+\nu_i)$$

因此,对有噪声的情况,只需从有噪声的 Φ_i^* 中减去 $N\sigma^2 I$,就可代替无噪声情况的 Φ_i 去进行结构辨识,这时有关的递推算法只需做微小的改动即可同样进行。具体讨论如下:

考察 Φ_{i+1} 与 Φ_i 的关系(这里为了书写方便,将“*”号省略掉,以下计算凡是用到的数据皆指有噪声的数据)。假定 Φ_i 已补偿过,则

$$\Phi_{i+1} = \begin{bmatrix} \Phi_i & \mathbf{R}_i^T \bar{\mathbf{y}}_{j\mu_j} \\ \bar{\mathbf{y}}_{j,\mu_j}^T \mathbf{R}_i & \bar{\mathbf{y}}_{j\mu_j}^T \bar{\mathbf{y}}_{j\mu_j} - N\sigma^2 \end{bmatrix}$$

因此,行列式的递推算法公式中相应修改为

$$\det \Phi_{i+1} = \det \Phi_i \det [\bar{\mathbf{y}}_{j\mu_j}^T (I - \mathbf{R}_i \Phi_i^{-1} \mathbf{R}_i^T) \bar{\mathbf{y}}_{j,\mu_j} - N\sigma^2]$$

在结构辨识中,对 $\det \Phi_i$ 的考察方法不变。

在矩阵求递公式中, 只要用

$$p^{-1} = \bar{y}_{j, \mu_j} (I - R_i \Phi_i^{-1} R_i^T) \bar{y}_{j, \mu_j} - N \sigma^2$$

代替矩阵求递公式中的 p , 其他皆不变。将上面修改过的递推公式替换递推算法中的有关步骤, 就得到适用于含噪声数据的递推算法。

下面提出几点注意事项:

① 如果各系统的随机干扰不是互不相关的, 只要 Q_i 已知, 同样可以按上面办法对 Φ_i 进行补偿, 只是补偿不是一个常值, 而是将 Q_i 的有关部分补偿进去。

② 当对所有输入、输出数据进行标度变换后, 假定同方差在实际上是合理的。 σ^2 可通过统计估计得到, 也可以对某个假定的较大的阶 μ , 求 $\Phi(\mu, \mu, \dots, \mu)$ 的最小特征值得到 (Guidorzi, 1975)。

③ 上面的讨论是在样本量 N 充分大时所做的近似。

4. 状态方程中包含矩阵 D 时的辨识

上面所讨论的结构及参数辨识都是在状态方程中不含 D 的情况下进行的, 它表征了系统的输出不直接包含输入的情况。在有些情况下, 输出不仅包含状态的信息, 而且含有输入的信息, 反应到数学描述上就是量测方程中包含 D 。下面讨论在 PCF 规范形包含 D 情况下的系统辨识。

考虑系统 $\sigma^*(A^*, B^*, C^*, D^*)$, 由 6.2 节的讨论知道, 它对应的输入、输出方程为式 (6.55), 注意到式 (6.55) 与不包 D 时输入、输出方程式 (6.57) 形式上的区别仅仅是 $u(\cdot)$ 的变化范围不同。但这形式上的不同反应了输入、输出之间关系产生了变化。由方程式 (6.55) 可以看出, 这时如果第 i 个子系统的结构指标为 ν_i , 则输出 $y_i(k + \nu_i)$ 可由 k 至 $k + \nu_i$ 时刻的输入 $u(\cdot)$ 和输出 $y_j(\cdot)$ ($j < i$) 及 k 至 $k + \nu_i - 1$ 时刻的输出 $y_j(\cdot)$ ($j \geq i$) 线性表示。

因此, 对于包含 D 的 PCF 规范形, 在确定结构指标时, 选择线性无关向量的顺序应改为

$$\begin{aligned} & \bar{u}_1(k), \bar{u}_2(k), \dots, \bar{u}_r(k), \bar{y}_1(k), \bar{y}_2(k), \dots, \bar{y}_m(k), \\ & \bar{u}_1(k+1), \bar{u}_2(k+1), \dots, \bar{u}_r(k+1), \bar{y}_1(k+1), \bar{y}_2(k+1), \dots, \bar{y}_m(k+1) \dots \end{aligned} \quad (6.74)$$

所以, 只要对本节前面所给出的算法中的记号, 如 R_i, Φ_i 等, 按照式 (6.74) 所给的顺序进行相应的改变, 不含 D 情况的辨识算法完全适用于含 D 的情况, 但这时参数向量的排列次序相应地变为

$$\begin{aligned} \theta_i^T = & (q_{i1,1} q_{i1,2} \dots q_{i1,r} : \alpha_{i1,1} \alpha_{i2,1} \dots \alpha_{im,1} : \\ & q_{i2,1} q_{i2,2} \dots q_{i2,r} : \alpha_{i1,2} \alpha_{i2,2} \dots \alpha_{im,2} : \dots) \end{aligned} \quad (6.75)$$

6.3.2 B^* 和 D^* 的实现算法

假定结构指标 $\{\nu_i\}$ 和参数 $\{\alpha_{ij,k}\}, \{q_{ij,k}\}$ 已经得到。那么, PCF 规范形中 A^*, C^* 可根据式 (6.24) 和式 (6.25) 直接写出。问题是如何从已知参数 $\{\alpha_{ij,k}\}$ 和 $\{q_{ij,k}\}$ 确定出 B^* 和 D^* 。

当系统的量测方程不含 D 时, 利用式 (6.56), 只要通过简单地回代便可求得 B^* 。具体地说, 在式 (6.56) 中令

$$\begin{array}{llll} i = 1 & j = \nu_1 & \text{可解得} & b_1 \\ i = 2 & j = \nu_2 & \text{可解得} & b_{S_1+1} \\ \vdots & \vdots & \vdots & \vdots \\ i = m & j = \nu_m & \text{可解得} & b_{S_{m-1}+1} \end{array}$$

再令

$$\begin{array}{llll} i=1 & j=\nu_1-1 & \text{解得} & b_2 \\ i=2 & j=\nu_2-1 & \text{解得} & b_{s_1+2} \\ \vdots & \vdots & & \vdots \\ i=m & j=\nu_m-1 & \text{解得} & b_{s_{m-1}+2} \end{array}$$

直到

$$\begin{array}{llll} i=1 & j=1 & \text{解得} & b_{s_1} \\ i=2 & j=1 & \text{解得} & b_{s_2} \\ \vdots & \vdots & & \vdots \\ i=m & j=1 & \text{解得} & b_{s_m} \end{array}$$

于是得到

$$\mathbf{B}^* = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

最后,确定初始状态。在式(6.35)中,令 $k=0$ 得

$$\mathbf{x}(0) = \mathbf{T}^{-1} \mathbf{E}^* \bar{\mathbf{y}}(0) - \mathbf{T}^{-1} \mathbf{E}^* \mathbf{R} \bar{\mathbf{u}}(0)$$

由 6.1 节最后指出的事实,对 PCF 规范形,变换矩阵 \mathbf{T} 是一个单位矩阵,所以有

$$\mathbf{x}(0) = \mathbf{E}^* \bar{\mathbf{y}}(0) - \mathbf{E}^* \mathbf{R} \bar{\mathbf{u}}(0) \quad (6.76)$$

根据数据 $\bar{\mathbf{y}}(0)$, $\bar{\mathbf{u}}(0)$ 和辨识得到的 \mathbf{R} 及 \mathbf{E}^* 的具体形式就得到初始状态 $\mathbf{x}(0)$ 。

【例 6.4】利用例 6.3 辨识的结果,求系统的状态实现。

由例 6.3 辨识得到的结构指标 $\{\nu_i\}$ 及参数分别为

$$\begin{aligned} \hat{\boldsymbol{\theta}}_1^T &= (\alpha_{11,1} \quad \alpha_{11,2} \quad \alpha_{11,3} \quad \vdots \quad \alpha_{12,1} \quad \alpha_{12,2} \quad \vdots \quad q_{11,1} \quad q_{12,1} \quad q_{13,1}) \\ &= (1 \quad 0 \quad -1 \quad \quad -1 \quad 1 \quad \quad 1 \quad 0 \quad 0) \\ \hat{\boldsymbol{\theta}}_2^T &= (\alpha_{21,1} \quad \alpha_{21,2} \quad \alpha_{21,3} \quad \vdots \quad \alpha_{22,1} \quad \alpha_{22,2} \quad \vdots \quad q_{21,1} \quad q_{22,1}) \\ &= (-1 \quad 0 \quad -1 \quad -1 \quad -2 \quad 1 \quad 0) \\ \nu_1 &= 3, \nu_2 = 2 \end{aligned}$$

由式(6.24)和式(6.25)即得

$$\mathbf{A}^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 1 & -2 \end{pmatrix} \quad \mathbf{C}^* = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

由式(6.56)得

$$\begin{aligned} q_{1\nu_1} &= q_{13,1} = b_1 = 0 \\ q_{2\nu_2} &= q_{22,1} = b_4 - \alpha_{21,3} b_1 \\ 0 &= b_4 \\ q_{1\nu_1-1} &= q_{12,1} = b_2 - \alpha_{11,3} b_1 \end{aligned}$$

$$\begin{aligned}
0 &= b_2 \\
q_{2\nu_2-1} &= q_{21,1} = b_5 - \alpha_{21,2}b_1 - \alpha_{21,3}b_2 - \alpha_{22,2}b_4 \\
1 &= b_5 \\
q_{11} &= q_{11,1} = b_3 - \alpha_{11,2}b_1 - \alpha_{11,3}b_2 - \alpha_{12,2}b_4 \\
1 &= b_3
\end{aligned}$$

从而得

$$\mathbf{B}^* = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

下面确定状态初始值。

因为 $\nu_1=3, \nu_2=2$, 所以选择矩阵为

$$\mathbf{E} = \begin{pmatrix} \mathbf{e}_1^T \\ \mathbf{e}_3^T \\ \mathbf{e}_5^T \\ \mathbf{e}_2^T \\ \mathbf{e}_4^T \end{pmatrix}$$

其中, \mathbf{e}_i 为单位阵 \mathbf{I}_5 的第 i 个列向量。

由如下原始数据

$$\begin{aligned}
\bar{\mathbf{y}}^T(0) &= (y_1(0), y_2(0), y_1(1), y_2(1), \cdots, y_1(4), y_2(4)) \\
&= (0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 2 \ 2 \ -1) \\
\bar{\mathbf{u}}^T(0) &= (u(0), u(1), \cdots, u(5)) = (1 \ 2 \ 4 \ 5 \ -5 \ -12)
\end{aligned}$$

和上述选择矩阵 \mathbf{E} , 并由式(6.46)知

$$\mathbf{E}\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ b_1 & & & & & \\ b_2 & b_1 & & & & \\ 0 & & & & & \\ b_4 & & & & & \end{pmatrix}$$

又 $b_1=b_2=b_4=0$, 代入式(6.76)得

$$\mathbf{x}(0) = \mathbf{E}\bar{\mathbf{y}}(0) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

当系统的量测方程包含 \mathbf{D} 时, 首先利用式(6.52)求得 \mathbf{D} , 具体地说, 在式(6.52)中令 $i=1, 2, \cdots, m$, 即可求得 d_1, d_2, \cdots, d_m , 从而得

$$D^* = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix}$$

然后利用式(6.51)求得 B^* , 具体步骤与不含 D 的情况一样。

【例 6.5】 考虑两输入、两输出的系统, 假定其结构指示及参数辨识结果如下面给出, 求其状态实现。

结构指示及参数辨识结果分别为: $\nu_1 = 2, \nu_2 = 1$;

$$\begin{aligned} \hat{\theta}_1^T &= (q_{11,1} \quad q_{11,2} \quad \vdots \quad \alpha_{11,1} \quad \alpha_{12,1} \quad \vdots \quad q_{12,1} \quad q_{12,2} \quad \vdots \quad \alpha_{11,2} \quad \vdots \quad q_{13,1} \quad q_{13,2}) \\ &= (3 \quad 0 \quad -1 \quad 2 \quad -2 \quad -4 \quad 1 \quad 1 \quad 2) \end{aligned}$$

$$\begin{aligned} \hat{\theta}_2^T &= (q_{21,1} \quad q_{21,2} \quad \vdots \quad \alpha_{21,1} \quad \alpha_{22,1} \quad \vdots \quad q_{22,1} \quad q_{22,2} \quad \vdots \quad \alpha_{21,2}) \\ &= (1 \quad -2 \quad 0 \quad 1 \quad -2 \quad 1 \quad 1) \end{aligned}$$

由式(6.24)和式(6.25)直接得到

$$A^* = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 1 & 2 \\ 0 & 1 & 1 \end{pmatrix} \quad C^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

由式(6.52)得

$$d_1 = q_{1\nu_1+1} = q_{13} = (q_{13,1} \quad q_{13,2}) = (1 \quad 2)$$

$$d_2 = q_{22} + \alpha_{21,2}d_1 = (-1 \quad 3)$$

再利用式(6.51), 令 $i=1, j=\nu_1=2$ 得

$$b_1 = q_{12} + \alpha_{11,2}d_1 = (-1 \quad -2)$$

令 $i=2, j=\nu_2=1$ 得

$$b_3 = q_{21} + \alpha_{21,1}d_1 + \alpha_{22,1}d_2 + \alpha_{21,2}b_1 = (-1 \quad -1)$$

令 $i=1, j=\nu_1-1=1$ 得

$$b_2 = q_{11} + \alpha_{11,1}d_1 + \alpha_{12,1}d_2 + \alpha_{11,2}b_1 = (-1 \quad 2)$$

从而得

$$D^* = \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix} \quad B^* = \begin{pmatrix} -1 & -2 \\ -1 & 2 \\ -1 & -1 \end{pmatrix}$$

【例 6.6】(模拟例子) 考虑具有如下参数的两输入、两输出系统 $\sigma(A, B, C)$, 其中

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -0.1 & 0.65 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0.67 & 1.67 & -0.25 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 2 \\ 0.25 & 0.8 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

用两个伪随机二位式序列作为输入, 利用上述系统产生 100 组数据, 然后用 6.3.1 节中给出的递推算法进行辨识。在算法中取向量的维数 $N=50$, 判别阈值 $\epsilon=10^{-3}$, 用 p 值的相对变化(即本次与上次的 p 值之比)作为判别准则, 计算结果如下:

$\Phi(k_1, k_2, k_3, k_4)$	(1 0 0 0)	(1 1 0 0)	(1 1 1 0)	(1 1 1 1)	(2 1 1 1)
$P_{\text{本}}/P_{\text{上}}$	0.1209676×10^4	0.5253327×10^2	0.6851036×10^{-2}	0.8866832×10^{-1}	0.4447516

(2 2 1 1)	(2 2 2 1)	(2 2 2 2)	(3 2 2 2)	(2 3 2 2)
0.1873646×10^2	0.1815529×10^{-1}	0.4320396×10	0.2337194×10^{-6}	0.2689188×10^{-3}

于是得到： $\nu_1=2, \nu_2=2$ 。

得到的参数估计为

$\alpha_{11,1}$ -0.100034	$\alpha_{12,1}$ $-0.9516277 \times 10^{-6}$	$q_{11,1}$ 0.2499941	$q_{11,2}$ -0.5000054	$\alpha_{11,2}$ 0.6500012
$\alpha_{12,1}$ 0.138183×10^{-5}	$q_{12,1}$ 0.4380384×10^{-5}	$q_{12,2}$ 0.199999×10		

$\alpha_{21,1}$ 0.6700959	$\alpha_{22,1}$ -0.2499688	$q_{21,1}$ 0.1000180×10	$q_{21,2}$ -0.2339886×10	$\alpha_{21,2}$ 0.1669988×10
$\alpha_{22,2}$ 0.9999569	$q_{22,1}$ $-0.1320780 \times 10^{-3}$	$q_{22,2}$ 0.3541227×10^{-4}		

状态实现为

$$A=\left[\begin{array}{cccc}0 & 1 & 0 & 0 \\-0.1000034 & 0.6500012 & -0.9516277 \times 10^{-6} & 0.1318183 \times 10^{-5} \\0 & 0 & 0 & 1 \\0.6700595 & 0.1669988 \times 10 & -0.2499688 & 0.9999569\end{array}\right]$$

$$B=\left[\begin{array}{cc}0.4380383 \times 10^{-5} & 0.1999999 \times 10 \\0.2499969 & 0.799963 \\-0.1320780 \times 10^{-3} & 0.3541227 \times 10^{-4} \\0.1000055 \times 10 & 0.1000123 \times 10\end{array}\right]$$

$$C=\left[\begin{array}{cccc}1 & 0 & 0 & 0 \\0 & 0 & 1 & 0\end{array}\right]$$

从以上模拟结果可见,递推算法的精度是非常理想的。

【例 6.7】 考虑系统方程为

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{Ax}(k) + \boldsymbol{Bu}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{Cx}(k) + \boldsymbol{Du}(k) \end{aligned}$$

其中

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -0.06 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 3.0 & -2.0 & -0.4 & 0.3 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 & 2 \\ -1 & 1 \\ 0 & 3 \\ 4 & 1 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0.1 & 0.4 \\ 2 & -0.8 \end{pmatrix}$$

用两个伪随机二位式序列作为输入,由上述系统产生 100 组数据,向量维数 N 取为 60, ε 取为 10^{-3} ,用 6.3.1 节中给出的递推算法辨识,其结果如下:

$\Phi(k_1, k_2, k_3, k_4)$	(0 0 1 0)	(0 0 1 1)	(1 0 1 1)	(1 1 1 1)
p	0.8000000×10^2	0.7595000×10^2	0.5327900×10^3	0.7869322×10^4

(1 1 2 1)	(1 1 2 2)	(2 1 2 2)	(2 2 2 2)	(2 2 3 2)
0.7993917×10^2	0.7370923×10^2	0.1597515×10^3	0.3859128×10^4	0.7842469×10^2

(2 2 3 3)	(3 2 3 3)	(2 3 3 3)		
0.6568777×10^2	0.2748941×10^{-5}	0.2708320×10^{-4}		

于是得到: $\nu_1 = 2, \nu_2 = 2$ 。

结构指标确定的同时,也得到参数估计如下:

$q_{11,1}$	$q_{11,2}$	$\alpha_{11,1}$	$\alpha_{12,1}$	$q_{12,1}$
-0.14940×10	0.24000×10^{-1}	-0.60000×10^{-1}	-0.52126×10^{-8}	0.95000
$q_{12,2}$	$\alpha_{11,2}$	$\alpha_{12,2}$	$q_{13,1}$	$q_{13,2}$
0.18000×10	0.50000	0.80428×10^{-8}	0.10000	0.40000

$q_{21,1}$	$q_{21,2}$	$\alpha_{21,1}$	$\alpha_{22,1}$	$q_{22,1}$
0.65000×10	0.25800×10	0.30000×10	-0.40000	-0.40000
$q_{22,2}$	$\alpha_{21,2}$	$\alpha_{22,2}$	$q_{23,1}$	$q_{23,2}$
0.40400×10	-0.20000×10	0.30000	0.20000×10	-0.80000

最后得到状态实现

$$\mathbf{A} = \begin{pmatrix} 0 & 1.0 & 0 & 0 \\ -0.06 & 0.5 & -0.52126 \times 10^{-8} & 0.80428 \times 10^{-8} \\ 0 & 0 & 0 & 1.0 \\ 3.0 & -2.0 & -0.4 & 0.3 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1.0 & 2.0 \\ -1.0 & 1.0 \\ -11921 \times 10^{-6} & 0.3 \\ 4.0 & 1.0 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0.1 & 0.4 \\ 2.0 & -0.8 \end{pmatrix}$$

辨识结果与原系统惊人地一致,一方面因为模拟数据没有随机噪声,另一方面在计算时采用双精度计算,而输出时采用单精度从而略去了尾数的缘故。

大量仿真例子表明,对于噪声不很大的多变量线性系统,用 6.3.1 节给出的递推辨识算法,不仅计算量很小,而且可以得到较理想的辨识结果。

习 题

1. 设多变量线性系统能观矩阵对 (A, C) 分别为

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 2 & 3 & 4 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- (1) 写出能观性矩阵。
- (2) 在由式(6.14)和式(6.15)规定的选择路线下,分别给出适宜选择,并指出结构指标。
- (3) 写出变为 PCF 规范形的变换矩阵和相应地选择矩阵。
- (4) 设矩阵

$$B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

写出多变量线性系统 $\sigma(A, B, C)$ 相应地输入、输出差分方程。

2. 已知一个两输入、两输出线性系统,由辨识得到的结果为:

结构指标: $\nu_1 = 2, \nu_2 = 2$;

参数向量分别为:

$$\begin{aligned} \hat{\theta}_1^T &= (q_{11,1} \quad q_{11,2} \quad \vdots \quad \alpha_{11,1} \quad \alpha_{12,1} \quad \vdots \quad q_{12,1} \quad q_{12,2} \quad \vdots \quad \alpha_{11,2} \quad \vdots \quad q_{13,1} \quad q_{13,2}) \\ &= (1 \quad 0 \quad -1 \quad 2 \quad -1 \quad -2 \quad 1 \quad 1 \quad 2) \end{aligned}$$

$$\begin{aligned} \hat{\theta}_2^T &= (q_{21,1} \quad q_{21,2} \quad \vdots \quad \alpha_{21,1} \quad \alpha_{22,1} \quad \vdots \quad q_{22,1} \quad q_{22,2} \quad \vdots \quad \alpha_{21,2}) \\ &= (-1 \quad -3 \quad 0 \quad 1 \quad -3 \quad 1 \quad 1) \end{aligned}$$

求 PCF 规范形 $\sigma(A^*, B^*, C^*, D^*)$ 的实现。

3. 用计算机实现由 6.3.1 节给出的递推辨识算法和 6.3.2 节给出的实现算法,并计算例 6.3 和例 6.6。

第 7 章 线性系统的非参数表示和辨识

7.1 线性系统的非参数表示

研究一个系统的输入量 $u(t)$ 和输出量 $y(u)$ 的关系,或者说建立它们之间的模型,是建模和辨识所要研究的一个主要问题。

7.1.1 脉冲响应函数

对于一个连续线性动态系统,输入、输出关系可以用一个微分方程表示

$$y^{(n)} + a_1 y^{(n-1)} + \cdots + a_{n-1} \dot{y} + a_n y = b_0 u^{(n)} + \cdots + b_{n-1} \dot{u} + b_n u + c \quad (7.1)$$

初始条件 $y_0^{(i)}, i=0, \cdots, n-1$ 系数 a_i, b_i 不依赖 u 或 y , 这是一个有限维的系统,它可以用有限个参数来决定,因此这种输入、输出模型称为参数模型,对于离散时间系统,就为在第 3 章和第 4 章中所讨论的模型。

线性系统有一个重要的性质就是叠加原理成立,这个原理的意思是:如果 y_1 是由输入 u_1 产生的输出, y_2 是由输入 u_2 产生的输出,那么, $\alpha y_1 + \beta y_2$ 便是由 $\alpha u_1 + \beta u_2$ 产生的输出。

还要注意到在上述微分方程中,输出 y 与输入 u 的关系是隐式的,即要通过解微分方程才能得到 $y(t)$,而在系统的分析与控制中,常常对它们之间的显式关系感兴趣,这样的关系可以表示为

$$y(t) = V[u(t)]_{-\infty}^t$$

其中,算子 V 可以是一个泛函。

例如, $\ddot{y} + a_1 \dot{y} + a_2 y = \dot{u} + b_2 u$, 则它的解可以表示为

$$y(t) = V[u(t)]_{-\infty}^t - \int_{-\infty}^t h(t-\tau) u(\tau) d\tau$$

这是一个有记忆的系统,也就是说,系统保存所有的历史纪录, $h(t)$ 可以用微分方程的解来确定,如可用 Laplace 变换得到

$$h(t) = L^{-1} \left[\frac{s + b_2}{s^2 + a_1 s + a_2} \right]$$

其中, L^{-1} 表示逆 Laplace 变换。

按照叠加原理,线性系统的动态行为都可以表示为

$$y(t) = \int_{-\infty}^t h(t-\tau) u(\tau) d\tau \quad (7.2)$$

其中, $h(t)$ 称为脉冲响应函数,它是在零初始条件下,假定 $u(t)$ 是一个脉冲函数 $\delta(t)$ 时式 (7.1) 的解。

式 (7.2) 也可写成

$$y(t) = \int_0^{\infty} h(\tau) u(t-\tau) d\tau \quad (7.3)$$

这样表示的显式的输入、输出关系因为是依赖于一个泛函,所以本质上是无穷维的,不是

由有限个参数来决定的,也就是不需要确定系统的阶,用这种表示有可能得到精确度很高的模型,式(7.2)也称为输入、输出的卷积表示。

只要有了 $h(t)$,就完全决定了系统的输入、输出关系,系统的性质也完全由 $h(t)$ 来决定,如系统的稳定性等。

7.1.2 Markov 参数(Hankel 模型)

对一个给定的离散时间多变量线性系统

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (7.4)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (7.5)$$

其中, $\mathbf{x}(k)$ 是 n 维状态向量, $\mathbf{u}(k)$ 是 r 维输入向量, $\mathbf{y}(k)$ 是 m 维输出向量,由式(7.4)、式(7.5)可决定输入、输出关系

$$\mathbf{x}(N) = \mathbf{A}^N \mathbf{x}(0) + \sum_{k=0}^{N-1} \mathbf{A}^{N-k-1} \mathbf{B} \mathbf{u}(k) \quad (7.6)$$

$$\mathbf{y}(N) = \mathbf{C} \mathbf{A}^N \mathbf{x}(0) + \mathbf{C} \sum_{k=0}^{N-1} \mathbf{A}^{N-k-1} \mathbf{B} \mathbf{u}(k) \quad (7.7)$$

系数矩阵 $\mathbf{M}_k = \mathbf{C} \mathbf{A}^k \mathbf{B}$, $k=0, 1, 2, \dots$, 称为实现 $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ 的 k 阶 Markov 参数

$$\bar{\mathbf{Y}} = \mathbf{M}_H \bar{\mathbf{U}} + \mathbf{H}_K \boldsymbol{\beta}, \mathbf{u}(i) = 0, i < 0 \quad (7.8)$$

称为 Hankel 模型,其中

$$\bar{\mathbf{Y}} = \begin{bmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ \mathbf{y}(3) \\ \vdots \end{bmatrix} \quad \bar{\mathbf{U}} = \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \mathbf{u}(2) \\ \vdots \end{bmatrix} \quad (7.9)$$

$$\mathbf{M}_H = \begin{bmatrix} \mathbf{M}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots \\ \mathbf{M}_1 & \mathbf{M}_0 & \mathbf{0} & \cdots & \mathbf{0} & \cdots \\ \mathbf{M}_2 & \mathbf{M}_1 & \mathbf{M}_0 & \cdots & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & & & \end{bmatrix} \quad (7.10)$$

$$\mathbf{H}_K = \begin{bmatrix} \mathbf{M}_0 & \mathbf{M}_1 & \cdots & \mathbf{M}_{k-1} & \cdots \\ \mathbf{M}_1 & \mathbf{M}_2 & \cdots & \mathbf{M}_k & \cdots \\ \mathbf{M}_2 & \mathbf{M}_3 & \cdots & \mathbf{M}_{k+1} & \cdots \\ \vdots & \vdots & & \vdots & \end{bmatrix} \quad (7.11)$$

$\boldsymbol{\beta}$ 是适当维数包含初始条件的分块向量, \mathbf{H}_K 称为 Hankel 矩阵。

很明显,从状态方程式(7.4)、式(7.5)完全可以确定全部 Markov 参数,反过来,如果已知 Markov 参数 \mathbf{M}_k , $k=0, 1, 2, \dots$, 能不能决定状态方程呢? 这个问题可以用下面两个结果来回答。

1. Markov 参数的性质

(1) Markov 参数 \mathbf{M}_k , $k=0, 1, 2, \dots$, 具有有限维实现 $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ 的充分必要条件是: 存在一个整数 r 和常数 a_i ($i=1, \dots, r$), 使得

$$\mathbf{M}_{r+j} = \sum_{i=1}^r a_i \mathbf{M}_{r+j-i} \quad \text{一切 } j \geq 0 \quad (7.12)$$

式中, r 是状态转移矩阵 A 的最小多项式的阶, r 称为可实现指标。

(2) 如果 Markov 参数 $M_k, k=0, 1, 2, \dots$, 具有有限维实现 $\{A, B, C\}$, 可实现指标 r 对这个实现的状态空间的最小维数 n_0 满足

$$\text{rank } |H_r| = n_0 \quad (7.13)$$

$|M_i|_{q \times p}$ 矩阵

$$n_0 \leq r \times \min(p, q) \quad (7.14)$$

并且

$$H_r = \begin{bmatrix} M_0 & M_1 & \cdots & M_{r-1} \\ M_1 & M_2 & \cdots & M_r \\ \cdots & \cdots & & \\ M_{r-1} & M_r & \cdots & M_{2r-2} \end{bmatrix} \quad (7.15)$$

由此可看出, 对一个可实现的 Markov 参数序列, 独立的 Markov 参数只有有限多个。由状态方程可以很方便地得到传递函数阵和 Markov 参数; 反过来, 从传递函数阵或 Markov 参数序列, 要得到状态方程就必须要通过比较复杂的实现算法。

作为非参数模型辨识, 就是从输入、输出数据得到 Markov 参数, 因而也就得到输入、输出关系, 再通过实现算法, 就可以得到状态方程, 不过, 要注意这样得到的状态方程只是最小实现, 但不一定是规范型。

2. 几种模型之间的关系

(1) 假定传递函数阵是 $G(s)$, 而 $U(s)$ 是输入向量的 Laplace 变换, 则输出向量 $y(t)$ (零初始条件) 为

$$y(t) = L^{-1}\{G(s)U(s)\} \quad (7.16)$$

同时, 用卷积表示为

$$y(t) = \int_0^\infty H(t-\tau)U(\tau)d\tau \quad (7.17)$$

其中

$$H(t) = L^{-1}\{G(s)\} \quad (7.18)$$

就是脉冲响应函数阵, 如果在第 i 个输入处给一个脉冲输入 $\delta_i(t)$, 则式 (7.17) 变成

$$\begin{aligned} y(t) &= \int_0^\infty \begin{bmatrix} H_{11}(t-\tau) & H_{12}(t-\tau) & \cdots & H_{1r}(t-\tau) \\ H_{21}(t-\tau) & H_{22}(t-\tau) & \cdots & H_{2r}(t-\tau) \\ \cdots & \cdots & & \\ H_{m1}(t-\tau) & H_{m2}(t-\tau) & \cdots & H_{mr}(t-\tau) \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ \delta_i(\tau) \\ 0 \\ \vdots \\ 0 \end{bmatrix} d\tau \\ &= \int_0^\infty \begin{bmatrix} H_{1i}(t-\tau) \\ H_{2i}(t-\tau) \\ \vdots \\ H_{mi}(t-\tau) \end{bmatrix} \delta_i(\tau) d\tau = H_i(t) \end{aligned} \quad (7.19)$$

(2) 对离散时间系统

$$G(z) = C(zI - A)^{-1}B \quad (7.20)$$

脉冲响应矩阵 $H(z)$ 可以展开成

$$H(z) = \sum_{i=0}^{\infty} \mathbf{M}_i z^{-(i+1)} \quad (7.21)$$

其中, \mathbf{M}_i 是 Markov 参数。另一方面, 有

$$\mathbf{y}(k) = \mathbf{C}\mathbf{A}^k\mathbf{x}(0) + \sum_{i=0}^{k-1} \mathbf{M}_i \mathbf{u}(k-i-1) = \mathbf{M}_k \boldsymbol{\beta}_0 + \sum_{i=0}^{k-1} \mathbf{M}_i \mathbf{u}(k-i-1) \quad (7.22)$$

其中, $\mathbf{B}\boldsymbol{\beta}_0 = \mathbf{x}_0$ 。

(3) 若已知系统的脉冲响应函数 $h(t)$, 并假定输入是正弦波信号

$$u(t) = A e^{j\omega t} \quad (7.23)$$

其中, A 是正弦波的幅度, $s = \sigma + j\omega$, ω 是正弦波的频率 (rad/s), j 是虚数单位, 将此输入信号代入式 (7.3) 得

$$y(t) = A \int_0^{\infty} h(\tau) e^{s(t-\tau)} d\tau = A e^{st} \int_0^{\infty} h(\tau) e^{-s\tau} d\tau$$

通过输入、输出信号的 Laplace 变换之比, 得到传递函数为

$$G(s) = \int_0^{\infty} h(\tau) e^{-s\tau} d\tau \quad (7.24)$$

当 $s = j\omega$, 即得到系统的稳态频率响应 $G(j\omega)$ 为

$$G(j\omega) = \int_0^{\infty} h(\tau) e^{-j\omega\tau} d\tau \quad (7.25)$$

只有当脉冲响应函数是代数函数时, 才能得到准确的频率响应, 否则只有用估计得到的脉冲响应函数的值去求近似的频率响应。

7.2 估计脉冲响应函数的相关方法

本节要介绍的是用脉冲输入及其响应来辨识脉冲响应函数, 因为真正的脉冲输入是物理上不可实现的, 所以必须产生一种有类似性质的输入信号, 为此, 在辨识方法和输入信号的构造上都需要进一步做工作。

7.2.1 相关方法的基本原理

对于单变量线性系统, 用脉冲响应函数即可描述系统的输入、输出关系, 因此只需估计出 $h(t)$ 就可以了。如果输入一个真正的脉冲函数 $\delta(t)$ 函数, 则 $\delta(t)$ 的响应就是脉冲响应函数 $h(t)$ 。即

$$h(t) = \int_0^{\infty} h(\tau) \delta(t-\tau) d\tau$$

但遗憾的是, $\delta(t)$ 是物理上不可实现的, 因此为了要得到 $h(t)$, 还需寻找其他方法, 最常用的就是相关方法。设系统的传递函数是 $G(s)$, 即

$$\bar{Y}(s) = G(s)U(s) \quad (7.26)$$

由卷积表示式 (7.3) 知

$$L[h(t)] = G(s) \quad (7.27)$$

在式 (7.26) 两端同乘以 $U(s)$, 得到

$$\bar{Y}(s)U(s) = G(s)U^2(s) \quad (7.28)$$

令

$$\phi_{uy}(t) = \int_0^{\infty} y(t-\tau)u(\tau)d\tau$$

$$\phi_{uu}(t) = \int_0^{\infty} u(t-\tau)u(\tau)d\tau$$

则有

$$L[\phi_{uy}(t)] = \bar{Y}(s)U(s)$$

$$L[\phi_{uu}(t)] = U^2(s)$$

所以根据式(7.27)和式(7.28),应用卷积定理即可得到

$$\phi_{uy}(t) = \int_0^{\infty} h(t)\phi_{uu}(t-\tau)d\tau \quad (7.29)$$

事实上, $\phi_{uy}(t)$ 是输入 $u(t)$ 和输出 $y(t)$ 的互相关函数, $\phi_{uu}(t)$ 是输入 $u(t)$ 的自相关函数, 式(7.29)是辨识 $h(t)$ 的基本关系式, 通常称为 Wiener-Hopf 方程。从方程可以看出, 我们用的输入信号的自相关函数 $\phi_{uu}(\tau)$ 是 $\delta(t)$ 函数, 则得到的互相关函数 $\phi_{uy}(t)$ 就是所要求的脉冲函数 $h(t)$, 问题就在于如何寻找这样的输入信号。

7.2.2 伪随机二位式信号(M 序列)

从前面的讨论可以知道, 现在要找自相关函数是 $\delta(t)$ 函数的信号。我们知道白噪声就是具有这种性质的信号, 但是连续时间白噪声信号又是物理上不可实现的, 因此在这里引进一种周期信号, 它和白噪声具有某种类似的性质, 即自相关函数近似于一个 $\delta(t)$ 函数, 这种信号就是伪随机二位式序列——M 序列(PRBS)。

1. M 序列的产生

这种信号是通过一个移位寄存器来产生的。如图 7.1 所示是一个 4 级移位寄存器。

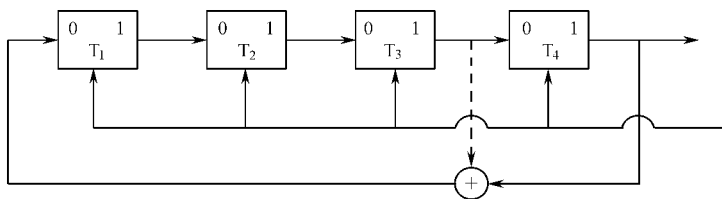


图 7.1 移位寄存器示意图

对某一初始状态, 如(1,1,1,1)将第 3 级与第 4 级的输出进行模 2 加, 然后反馈到输入端, 则此各级寄存器的状态变化如下:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
2	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1
3	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1

只可能有 15 种不同的状态, 从任何一级输出即可得到一个二位式的信号, 例如从第二级输出即可得到: 110001001101011。因为这一信号是 4 级移位寄存器可以产生的信号中周

期最长的,所以称为最长周期序列,简称 M 序列。

用这样的方法得到序列是周期性的,其周期大小与寄存器级数及反馈回路所选的级数有关,例如在上面的例中,反馈信号从第 2 级和第 4 级引出,则得到:

1	1	0	0	1	1	1	1
2	1	1	0	0	1	1	1
3	1	1	1	0	0	1	1
4	1	1	1	1	0	0	1

周期为 7,因为 n 级的寄存器,总共可能有 2^n 种不同的状态,除去全是“0”的状态之外,最多可能有 $2^n - 1$ 种,所以最大周期为 $N = 2^n - 1$,周期为 N 的序列就是最大长度序列,即 M 序列。

究竟应该如何选取寄存器和反馈位置才能得到合用的 M 序列呢?下面做一点简单的讨论,将一个 n 位寄存器与一个 n 次多项式

$$f(x) = 1 + c_1x + \cdots + c_nx^n$$

相对应,其中 $c_i = 0$ 或 $1, i = 1, \cdots, n, x = 0$ 或 1 ,由此多项式可以产生一个周期序列 $G(f)$, $f(x)$ 称为 $G(f)$ 的特征多项式,例如前面所举的例中

$$f(x) = 1 + c_3x^3 + c_4x^4$$

要使 $f(x)$ 产生 M 序列,则 $f(x)$ 必须是不可约的并且是 $x^n + 1$ 的因子。要把这样的多项式都找出来是很麻烦的,但该工作已经有人做了,并且将满足条件的多项式列成表可供使用时查找。现摘录一部分,供使用时参考,见附录 C。

2. M 序列的性质

下面给出 M 序列的一些重要性质:

(1) 对 n 级移位寄存器所产生的 M 序列的长度为 N 位

$$N = 2^n - 1$$

(2) 在长度为 N 的 M 序列中,包含 $(N+1)/2$ 个逻辑 1 状态和 $(N-1)/2$ 个逻辑 0 状态。

(3) 将一个 M 序列延迟 r 位再与原序列进行模 2 加,所得到的序列是和原序列一样的 M 序列,只是延迟了 q 位,且

$$1 \leq r, q \leq N - 1$$

(4) 如果逻辑状态 1 对应于幅值为 $+a$ 的信号,逻辑状态 0 对应于幅值为 $-a$ 的信号,则 M 序列的过零(即由一种状态转换为另一种状态)的数目为 $(N-1)/2$ 或 $(N+1)/2$ 。

(5) 如上的两个 M 序列相乘 $x(t)x(t-\tau)$,在每一时刻,若两个序列的逻辑状态相同(即两个序列同为 1 和同为 0),则乘积为 $+a^2$;若逻辑状态不同,则乘积为 $-a^2$ 。这一结果和模 2 加一样。所以, $x(t)x(t-\tau)$ 仍然是 M 序列。

以上几个重要性质对正确使用 M 序列是非常有用的。

3. M 序列的自相关函数

讨论 M 序列的目的就是要用 M 序列去代替实际上不能实现的白噪声序列。所以,重要的是要看其自相关函数是否能近似为一个 $\delta(t)$ 函数,下面就来找出 M 序列的自相关函数。

将 M 序列记为 $u(t)$,其周期为 T ,自相关函数为

$$\phi_{uu}(\tau) = \frac{1}{T} \int_0^T u(t)u(t-\tau)dt$$

如果移位的时序脉冲的周期是 Δt , 则上式可写为

$$\phi_{uu}(\tau) = \frac{1}{N} \sum_{i=1}^N u(i\Delta t)u(i\Delta t - \tau) \quad (7.30)$$

显然

$$\phi_{uu}(0) = \frac{N}{N} a^2 = a^2$$

其中, a 是信号的幅值。再看当 $\tau = k\Delta t$ 时, 因为 M 序列与其一个移位序列相加(模 2)仍然是其自身的一个移位序列, 注意到 M 序列的性质(2), 有

$$\phi_{uu}(\tau) = \frac{1}{N} \{\text{逻辑相同数} - \text{逻辑不同数}\} a^2$$

由性质(5)知, $u(t)u(t-\tau)$ 仍然是一个 M 序列, 所以也有 $\frac{1}{2}(N+1)$ 个“1”和 $\frac{1}{2}(N-1)$ 个“0”, 即有 $\frac{1}{2}(N+1)$ 个“逻辑不同”和 $\frac{1}{2}(N-1)$ 个“逻辑相同”, 故

$$\phi_{uu}(t) = \frac{a^2}{N} \left[\frac{1}{2}(N-1) - \frac{1}{2}(N+1) \right] = -\frac{a^2}{N}$$

由于 M 序列在 $k\Delta t$ 与 $(k+1)\Delta t$ 之间是常数, 故自相关函数在两个相邻的 k 值之间是一条直线。

当 τ 值的变化范围在 $0 < \tau < \Delta t$ 时, 由于 M 序列具有 $\frac{1}{2}(N+1)$ 个过零点, 于是在每个时间间隔 Δt 内, 以 $\phi_{uu}(0)$ 中移去一块面积 $a^2 T / (N\Delta t)$ 而代之以同样的一块面积 $-a^2 T / (N\Delta t)$, 故自相关函数为

$$\phi_{uu}(\tau) = \phi_{uu}(0) - \frac{N+1}{2} \frac{a^2 \tau}{N\Delta t} - \frac{N+1}{2} \frac{a^2 \tau}{N\Delta t} = a^2 \left[1 - \frac{\tau(N+1)}{N\Delta t} \right]$$

当 $(N-1)\Delta t < \tau < N\Delta t$ 时, 也可类似得到

$$\phi_{uu}(\tau) = a^2 \left[1 - \frac{(N\Delta t - \tau)(N+1)}{N\Delta t} \right]$$

最后得到 M 序列的自相关函数为

$$\phi_{uu}(\tau) = \begin{cases} a^2 \left[1 - \frac{\tau(N+1)}{N\Delta t} \right] & 0 \leq \tau < \Delta t \\ -\frac{a^2}{N} & \Delta t \leq \tau \leq (N-1)\Delta t \\ a^2 \left[1 - \frac{(N\Delta t - \tau)(N+1)}{N\Delta t} \right] & (N-1)\Delta t < \tau < N\Delta t \end{cases} \quad (7.31)$$

4. 当 M 序列为输入时的输入、输出互相关函数

根据 Wiener-Hopf 方程

$$\phi_{uy}(\tau) = \int_0^\infty h(s)\phi_{uu}(\tau-s)ds$$

对一个渐近稳定的系统, $h(s) = 0, s > \tau$, 故得

$$\phi_{uy}(\tau) = \int_0^\tau h(s)\phi_{uu}(\tau-s)ds$$

将 $\phi_{xx}(\tau)$ 代入得

$$\phi_{uy}(\tau) = \frac{a^2(N+1)}{N} \int_0^T h(s) \bar{\phi}_{uu}(\tau-s) ds - \frac{a^2}{N} \int_0^T h(s) ds \quad (7.32)$$

其中, $\bar{\phi}_{uu}(\tau)$ 是一个高度为 1 个单位、底部宽度是 $2\Delta t$ 的三角形, 即

$$\bar{\phi}_{uu}(\tau) = \begin{cases} \frac{\tau + \Delta t}{\Delta t} & -\Delta t \leq \tau \leq 0 \\ \frac{\Delta t - \tau}{\Delta t} & 0 \leq \tau \leq \Delta t \end{cases}$$

将脉冲响应函数 $h(s)$ 在 $s=\tau$ 处展开, 并做一阶近似得到

$$\phi_{uy}(\tau) = A_0 \frac{a^2(N+1)}{N} h(\tau) - \frac{a^2}{N} \int_0^T h(s) ds$$

其中

$$A_0 = \int_0^T \bar{\phi}_{uu}(\tau-s) ds = - \int_T^{-T} \bar{\phi}_{uu}(s) ds$$

当 $\tau \geq \Delta t$ 时

$$A_0 = - \left[\int_{\Delta t}^0 \frac{\Delta t - s}{s} ds + \int_0^{-\Delta t} \frac{s + \Delta t}{s} ds \right] = \Delta T$$

当 $\tau = 0$ 时

$$A_0 = - \int_0^{-\Delta t} \frac{s + \Delta t}{\Delta t} ds = \frac{\Delta t}{2}$$

当 $0 < \tau < \Delta t$ 时

$$A_0 = \left(\int_{-\Delta t}^0 + \int_0^\tau \right) \bar{\phi}_{xx}(s) ds = \frac{\Delta t}{2} + \tau \left(1 - \frac{\tau}{2\Delta t} \right)$$

所以

$$\begin{aligned} \phi_{uy}(\tau) &= \frac{a^2(N+1)\Delta t}{N} h(\tau) - \frac{a^2}{N} \int_0^T h(s) ds & \tau \geq \Delta t \\ \phi_{uy}(0) &= \frac{a^2(N+1)\Delta t}{2N} h(0) - \frac{a^2}{N} \int_0^T h(s) ds & \tau = 0 \\ \phi_{uy}(\tau) &= \frac{a^2(N+1)}{N} \left[\frac{\Delta t}{2} + \tau \left(1 - \frac{\tau}{2\Delta t} \right) \right] h(\tau) - \frac{a^2}{N} \int_0^T h(s) ds & 0 < \tau < \Delta t \end{aligned} \quad (7.33)$$

这是估计脉冲响应函数的重要结果。

7.2.3 用 M 序列做输入信号时脉冲响应函数的估计

这里考虑采用数值计算的情形, 假定采样时间满足 $(N-1)\Delta t > T_s$, T_s 是系统的时间常数, 并且采样周期与 M 序列 $u(t)$ 的时序脉冲同步, 于是 Wiener-Hopf 方程可表示为离散形式

$$\phi_{uy}(\tau) = \sum_{s=0}^{N-1} h(s) \phi_{uu}[(\tau-s)\Delta t] \quad (7.34)$$

根据遍历性定理, ϕ_{uy}, ϕ_{uu} 可表示为 (为了方便起见, 令 $\Delta t = 1$)

$$\phi_{uu}(\tau-s) = \frac{1}{N} \sum_{t=0}^{N-1} u(t-s)u(t-\tau) \quad (7.35)$$

$$\phi_{uy}(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} y(t)u(t-\tau) \quad (7.36)$$

将式(7.34)离散化得

$$\phi_{uy}(\tau) = \frac{(N+1)a^2}{N}h(\tau) - \frac{a^2}{N}\sum_{t=0}^{N-1}h(t) \quad (7.37)$$

而 M 序列的自相关函数是式(7.31)

$$\phi_{uu}(\tau) = \begin{cases} a^2 & \tau = kN, k = 0, 1, 2, \dots \\ -\frac{a^2}{N} & \text{其他} \end{cases}$$

由式(7.37)有

$$h(\tau) = \frac{N}{(N+1)a^2}\phi_{uy}(\tau) + \frac{1}{N+1}\sum_{t=0}^{N-1}h(t) \quad (7.38)$$

对式(7.37)两端求和得

$$\sum_{t=0}^{N-1}\phi_{uy}(t) = \frac{(N+1)a^2}{N}\sum_{t=0}^{N-1}h(t) - a^2\sum_{t=0}^{N-1}h(t) = \frac{a^2}{N}\sum_{t=0}^{N-1}h(t)$$

所以

$$\sum_{t=0}^{N-1}h(t) = \frac{N}{a^2}\sum_{t=0}^{N-1}\phi_{uy}(t) \quad (7.39)$$

将式(7.39)代入式(7.38)得

$$\begin{aligned} h(\tau) &= \frac{N}{(N+1)a^2}\phi_{uy}(\tau) + \frac{N}{(N+1)a^2}\sum_{t=0}^{N-1}\phi_{uy}(t) \\ &= \frac{N}{(N+1)a^2}\left[\phi_{uy}(\tau) + \sum_{t=0}^{N-1}\phi_{uy}(t)\right] \end{aligned} \quad (7.40)$$

由式(7.40)知脉冲响应函数 $h(t)$ 与 $\phi_{uy}(t)$ 是线性关系,而系数是已知的,式(7.40)就是用 M 序列估计脉冲响应函数的基本公式。

7.2.4 估计 $h(t)$ 的具体步骤与实施

1. 算法

利用 M 序列作为输入去估计脉冲响应函数 $h(t)$ 的算法如下:

- (1) 选择参数:采样间隔 Δt , 信号幅度 a , 信号周期 N ;
- (2) 产生伪随机码;
- (3) 计算互相关函数 $\phi_{uy}(t)$, $t=0, \dots, (N-1)\Delta t$;
- (4) 用式(7.40)估计 $h(t)$ 。

2. 具体实施要点

(1) 要选好一些基本参数,首先可以通过简单的实验,估计系统的时间常数 T_s 和截止频率 f ,确定截止频率可用输入不同频率的低幅值方波序列逐步增加频率,直到输出不响应为止,这时的频率就可以作为截止频率的估计值。

选择好这些参数后,就可以来选择伪随机码,它的周期必须大于系统的时间常数 T_s ,即

$$(N-1)\Delta t > T_s$$

采样频率 $\frac{1}{\Delta t}$ 必须小于截止频率 f ,一般 Δt 约为 $\frac{1}{f}$ 的 2~5 倍。

例如,一个系统的时间常数是 15min,而 $\frac{1}{f}$ 估计为 3s,所以采样周期(也是时序脉冲的宽度)可取为 10s,伪随机码周期可取 $2^8 - 1 = 127$,则

$$127 \times 10 = 1270\text{s} > 900\text{s} = 15\text{min}$$

(2) 信号幅度原则上应该小些,这样可以不影响正常操作,但是信号太弱又可能使检测受影响。因此,可以在实际系统允许的条件下取较大的幅度,在幅度提高受限制时,可加大采样周期,但是,采样周期增大,伪随机信号的自相关函数与 $\delta(t)$ 函数相差就更大了。此时要权衡得失,适当选择这些参数。

(3) 伪随机信号除了上面所介绍的 M 序列外,还可以用其他方法产生,例如一种较好的信号——逆重复序列,它的产生方法是:用 $2N$ 个序列码与 $2N$ 个方波按模 2 加法相加,便得到逆重复序列。例如,当 $N=7$ 时,逆重复序列为

$$\begin{array}{cccccccccccccccc} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

这样的序列具有如下性质:

- ① 逆重复序列的周期是原序列的 2 倍;
- ② 如果将逻辑 1 看做是电平 $+a$,逻辑 0 看做是 $-a$,则在序列中,两个电平出现的次数相等,因此整个周期的平均值为零。
- ③ 序列的前半周期与后半周期是逆重复的。
- ④ 序列与原序列不相关,即互相关函数为零。
- ⑤ 序列的自相关函数与原序列的自相关函数 $\phi_{uu}(t)$ 有如下关系

$$\phi_{uu}(\tau) = \begin{cases} \phi_{uu}(\tau), & \tau \text{ 为偶数} \\ -\phi_{uu}(\tau), & \tau \text{ 为奇数} \end{cases} \quad (7.41)$$

其自相关数如图 7.2 所示。

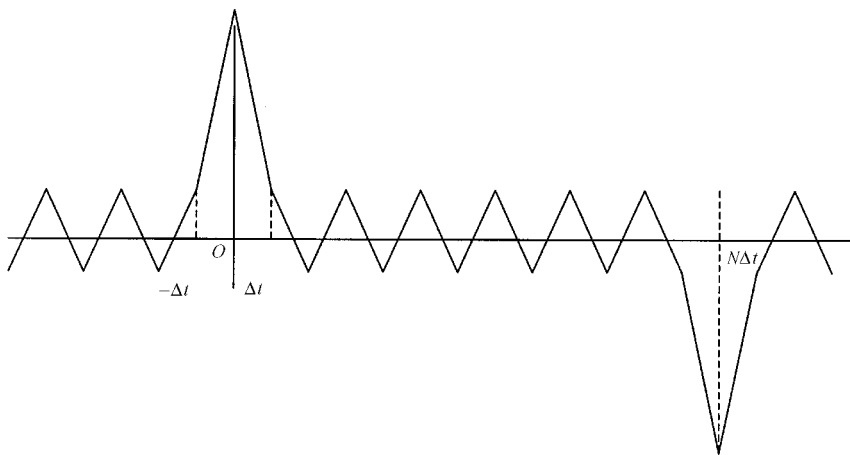


图 7.2 逆重复序列的自相关函数

用逆重复序列 L 作为输入信号可得

$$h(t) = \frac{N}{(N+1)a^2} \phi_{ly}(t) \quad (7.42)$$

这时不像式(7.40)那样有一常数项,所以比 M 序列更为简单。

(4) 用伪随机信号作为输入信号辨识脉冲响应函数具有不少优点,主要表现为:

- ① 伪随机信号近似于白噪声,近似程度是可以控制的;
- ② 伪随机信号本身没有统计误差,能准确地重复;
- ③ 伪随机信号实现延迟很简单,便于计算相关函数;
- ④ 用伪随机信号可缩短测量时间;
- ⑤ 便于数字计算。

另一方面,在应用伪随机信号时,也需要注意被辨识系统的特性,当系统的时间常数较大,截止频率较低,特别是实际系统不允许控制动作频繁改变,同时影响系统的其他因素又比较复杂时,用伪随机信号的效果就可能不好。因为采样周期长(有时达几分钟到十几分钟),整个实验时间就要很长。可是信号的幅度又不允许太大,在实验过程中其他因素就可能变化,所以很难得到单纯输入信号的影响。这就影响到实验精度和脉冲响应函数的真实性。在化工、炼油等大型装置和流程中,常常会出现这种情况,在使用时应该很好地分析,以便选择合适的方法。

(5)部分可用于产生 M 序列的多项式

下面给出了部分不可约的原始多项式,它可供产生 M 序列时应用。一个 M 序列用两个号码表示为 (i, j) 。 i 表示移位计存器的级数 n ,它决定 M 序列总的位数 N ; j 表示移位寄存器中通过模 2 门反馈的位数。

例如 $(6, 2)$,表示总位数 $63(2^6 - 1)$ 位,它由 6 级位移寄存器产生,其中第 2 位通过模 2 门进行反馈。相应的不可约的、原始的多项式记为 $(n, 0, p, q, r, \dots)$,多项式即为: $1 \oplus \dots \oplus x^p \oplus x^q \oplus x^r \oplus \dots \oplus x^n = 0$,其 M 序列具有特征多项式

$$f(x) = 1 \oplus x^5 \oplus x^6$$

记为 $(6, 5, 0)$ 。

下面给出部分可用于产生 M 序列的多项式:

M 序列编号	多项式
(3, 2)	(3, 1, 0)
(4, 2)	(4, 1, 0)
(5, 2)	(5, 2, 0)
(5, 4)	(5, 3, 2, 1, 0)
(6, 2)	(6, 5, 0)
(6, 4)	(6, 4, 3, 1, 0)
(7, 2)	(7, 1, 0)
(7, 4)	(7, 5, 2, 1, 0)
(7, 6)	(7, 6, 5, 4, 3, 2, 0)
(8, 4)	(8, 7, 2, 1, 0)
(8, 6)	(8, 6, 4, 3, 2, 1, 0)
(9, 2)	(9, 4, 0)
(9, 4)	(9, 6, 4, 3, 0)
(9, 6)	(9, 6, 5, 4, 2, 1, 0)
(9, 8)	(9, 8, 7, 6, 5, 4, 3, 1, 0)

(10,2)

(10,3,0)

(10,6)

(10,6,5,3,2,1,0)

(10,8)

(10,9,6,5,4,3,2,1,0)

习 题

1. 证明 7.1.2 节给出的 Markov 参数的重要性质(1)和(2)。
2. 写出由输入、输出数据估计脉冲响应函数的算法。
3. 请回答下列问题：
 - (1) $\delta(t)$ 函数作为输入有什么特殊作用？
 - (2) 为什么 $\delta(t)$ 函数物理上不可实现？
 - (3) 用什么方法可近似地替代 $\delta(t)$ 函数作为输入？
4. 什么是估计脉冲响应函数的相关方法？表述其基本原理和实现方法。
5. 估计脉冲响应函数可能应用的是什么类型的输入、输出数据？试举出两种常用的此类输入数据。
6. 证明卷积定理式(7.29)。
7. 试构造两种移位寄存器,使其产生 63 位 M 序列。
8. 证明 M 序列的性质(2)。
9. 证明 M 序列的性质(3)。
10. 画出采样周期为 Δt 的 7 位 M 序列的自相关函数。

第 8 章 非线性系统辨识

8.1 引言

非线性是研究、分析系统时常常会遇到的一种现象。非线性系统的行为是十分丰富的,可以表现为跳跃、滞后、极限环、分岔及混沌运动等线性系统不可能出现的现象。

非线性现象从整体上是很复杂的,通常在具体研究这些现象时,为了简化问题,常常把研究的范围限制在系统的局部性质,这样就可以用泰勒展开式中的一次项来近似地描述。但这仅表示了系统的线性性质,而非线性性质包含在高次项中。所以为了要研究非线性的整体行为,就要用与研究线性系统不同的方法。而为了要对非线性系统进行正确的分析和综合,首先就要正确地描述系统的非线性性质,也就是要建立系统的非线性模型。

非线性系统辨识中最困难的问题之一是模型结构的确定。因为非线性的含义十分广泛,在先验知识不足时,要选择完全确定的模型形式很不容易,要把一个系统的非线性行为的定性性质都了解清楚也很困难,所以要建立一个完全“准确”的非线性模型是不现实的。另一方面,在各种应用中(如控制系统的设计、各种预报等),只要给出适当的输入、输出表示,至于这种表示是否惟一并不是本质的。在这种情况下,建模问题主要取决于可供选择的模型类(即可表示为哪一类非线性模型)及由输入、输出数据确定的模型的可辨识性。

选择非线性模型的结构,除了根据对象的机理(如物理、化学的原理、定律等)或经验确定模型的结构外,主要有两种途径:一种是根据已有的先验知识,从各种典型非线性模型类中去选择一个合适的结构,如指数模型、对数模型、双线性模型、Hammerstein 模型及其他特殊类型的模型;另一种途径是当模型的先验知识十分缺乏时,无法确定模型的特殊结构,则可用某种一般形式的结构去逼近非线性系统。目前常用的是用 Volterra 型泛函级数及其离散形式 Kolmogorov-Gabor 多项式,或用神经网络模型等。关于非线性系统的典型结构及辨识方法可参看文献(Haber R and Unbehauen H, 1990)。

非线性模型的辨识和其他模型的辨识一样,包括结构辨识和参数估计两部分,这两部分在研究中有时是分开进行的,有时又结合在一起。

本章先讨论已知结构的非线性系统

$$\frac{dx}{dt} = f(x, \theta), x \in R^n, \theta \in R^m \quad (8.1)$$

的参数估计问题。其中 f 是已知函数, θ 是待估参数向量。如果 f 具有特殊的形式,就可能有一些特殊的方法,这里讨论的是一般形式。由于通常所取得的数据大多数是在离散时间上的检测点,所以对模型(8.1)的参数估计问题往往归结为一个非线性函数的拟合问题。从初始值 $x(t_0) = x_0$ 出发可以得到模型的解,从而可以得到系统的理论输出(按模型计算得到的) $\phi(x_0, \theta, t)$, ϕ 是 θ 的非线性函数,将这样计算得到的输出与实测输出数据进行比较,按误差平方和最小的准则,使

$$J = \sum_{k=1}^N [y(t_k) - \phi(x_0, \theta, t_k)]^2 \quad (8.2)$$

达到最小。

由于采用的是误差平方和最小准则,所以称此问题为非线性最小二乘问题。一般地说,这是稳态非线性模型的参数估计问题,本章的前 5 节将讨论这一问题的解法。值得注意的是,解这一问题是以函数 f 的形式已知为前提的。

本章的最后两节将讨论非线性系统辨识的逼近方法,其中包括 GMDH 方法和用 Kolmogorov-Gabor 多项式逼近的 NARMAX 模型的辨识方法。本书接下来的几章实际上都是用不同的观点、不同的方法,讨论不同领域的非线性系统辨识问题。

8.2 单纯形搜索法

8.2.1 问题的提法

用理论建模的方法,即根据系统本身的运动规律来决定模型的结构,所得到的模型大部分为形如式(8.1)的非线性模型,当然也可能是稳态模型,即

$$g(y, u, \theta) = 0 \quad (8.3)$$

$$\text{其中} \quad g = (g_1, g_2, \dots, g_n)^T \quad (8.4)$$

y, u 分别为输出、输入向量, θ 是参数向量。只有在式(8.3)能解出 y , 即式(8.3)可写成

$$y = f(u, \theta) \quad (8.5)$$

时,才能真正预测 y 。

显然,式(8.5)是稳态模型。对于动态模型式(8.1),实际上在估计参数时也将其归结为

$$y = \phi(t_0, u, \theta) \quad (8.6)$$

式(8.6)对同一时刻的输入、输出变量 u, y 成立。

总之,对于已知模型形式的非线性系统的参数估计,不论系统是动态的还是静态的,都可归纳为模型式(8.5)的参数估计问题。对模型式(8.5)的参数估计的基本方法是解这一非线性最小二乘问题。

假定被辨识的系统满足式(8.5),对系统进行若干次试验,得到输入及相应的输出观测值为 $\{(u(i), y(i)), i=1, 2, \dots, N\}$, 每次实验的模型偏差是

$$e(i) = y(i) - f[u(i), \theta], i = 1, 2, \dots, N \quad (8.7)$$

取目标函数为误差平方和,即取

$$J = \sum_{k=1}^N e^T(k) e(k) \quad (8.8)$$

现在的问题是要求参数 θ 的估计 $\hat{\theta}$, 使得

$$J(\hat{\theta}) = \min_{\theta} \{J(\theta)\}$$

值得注意的是,这里 f (对 θ 而言)是非线性函数,所以求式(8.8)的极值问题就不能像在线性系统那样归结为求解正则方程的问题了。求解这一非线性最小二乘问题一般有两类方法,一类是直接搜索法,另一类是迭代算法。下面先介绍一种常用的搜索法。

8.2.2 单纯形搜索法

搜索法总的思路是:同时选择若干个参数点,分别计算它们的目标函数值,并比较目标函数值的大小,选出对应的目标函数值为最小的参数点,去掉对应的目标函数值为最大的参数

点,然后再选一批新的点,与原有的点进行比较,再选使目标函数最小的参数点,直到选不出更好的参数值为止。

搜索法的种类很多,其不同点在于:每次比较选几个点,每次比较完后保留几个点进行下一次比较;新参加比较的点怎么找、如何选;如何判断最好的点已经选到等。

这里将讨论的是“单纯形搜索法”,方法的梗概是:对 n 维参数向量 θ ,先选出 $n+1$ 个参数点,每一个点是一个 n 维空间的向量(它表示参数向量 θ 的一个估计)。以这些点的顶点构成 n 维空间的一个单纯形(我们在这里取正多面体),并计算各个参数点的目标函数值。然后将目标函数值最大的那个点剔除,再按某种方式补一新的参数点,构成一个新的单纯形。再进行新的比较,直到单纯形的各顶点对应的目标函数值都没有明显的差别为止。计算步骤详述如下:

(1) 对包含 n 个参数的模型,选 $n+1$ 个参数点,即

$$\theta_i = (\theta_{i_1}, \dots, \theta_{i_n})^T, i = 1, \dots, n+1 \tag{8.9}$$

并以这 $n+1$ 个点作为顶点构成一个单纯形。为了方便起见,不妨使这 $n+1$ 个点构成一个正多面体。例如, $n=2$,则选 3 个点构成一个正三角形; $n=3$,则选 4 个参数点构成一个正四面体。对参数没有先验知识可参考时,可将最初的单纯形的一个顶点取做原点,由解析几何的知识容易算出 $n+1$ 个顶点的坐标,如表 8.1 所示。对 $n+1$ 个参数点即可算得 $n+1$ 个目标函数值。

表 8.1 初始单纯形顶点坐标

顶点	坐 标				
I	θ_{i_1}	θ_{i_2}	\dots	$\theta_{i_{n-1}}$	θ_{i_n}
1	0	0	\dots	0	0
2	π_1	π		π	π
3	π	π_1		π	π
\vdots	\dots		\dots	\dots	\dots
n	π	π		π_1	π
$n+1$	π	π		π	π_1

表中

$$\pi_1 = a/(\sqrt{2}n) \cdot (\sqrt{n+1} + n - 1) \tag{8.10}$$

$$\pi = a/(\sqrt{2}n) \cdot (\sqrt{n+1} - 1) \tag{8.11}$$

式中, a 是单纯形的边长。

注: 当顶点不在原点时,如在 (a_1, \dots, a_n) (如果已知参数初始值时就是此情况),则表 8.1 中第一个参数点改为 (a_1, \dots, a_n) ,其他参数点的第 j 个元素相应地加上 a_j 即可。

(2) 对 $n+1$ 个参数点分别计算其目标函数值,并记

$$J_M = \max_i(J_i), J_l = \min_i(J_i)$$

J_M 和 J_l 对应的参数点分别记为 θ_M 和 θ_l ,显然 θ_M 是这 $n+1$ 个参数点中最坏的一个。注意,除去 θ_M 之外,其余的 n 个参数点为顶点仍然可以构成一个单纯形,这个单纯形的中心(其坐标为各顶点坐标的平均)记为 c 。现剔除最坏的点 θ_M ,代之以一个新的参数点 θ^* , θ^* 是将 θ_M 反射而得到的,由式(8.12)计算。图 8.1 给出了示意图。

$$\theta^* = (1 + r_r)c - r_r\theta_M \tag{8.12}$$

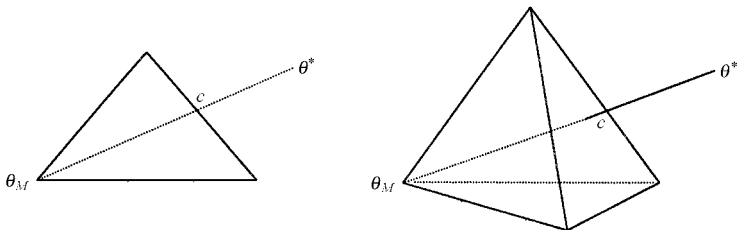


图 8.1 单纯形的反射

式中, r_r 称为反射系数, 是正常数, 通常取为 1。记 $J^* = J(\theta^*)$, 这时可能产生 3 种情况:

① 若 $J_l < J^* < J_{i_0}$ (θ_{i_0} 为不等于 θ_M 的某一点)。即反射后得到的点有所改进, 则用 θ^* 代替 θ_M , 得到一个新的单纯形, 转步骤(3)。

② 若 $J^* < J_l$, 即反射后得到一个最好的点。意味着按此反射方向改进幅度很大, 希望能再进一步改进, 于是将 θ^* 开拓到 θ^{**} 。

$$\theta^{**} = (1 + r_c)\theta^* - r_c c \quad (8.13)$$

式中, r_c 称为开拓系数 (如可取为 2), 并计算 $J^{**} = J(\theta^{**})$, 若 $J^{**} < J^*$, 则用 θ^{**} 代替 θ_M , 转步骤(3); 若 $J^{**} > J^*$, 即开拓失败, 则仍用 θ^* 代替 θ_M , 转步骤(3)。

③ 若 $J^* > J_l$, $i \neq M$, 即用 θ^* 代替 θ_M 后, θ^* 仍然是最坏的, 即沿反射方向是不利的, 所以应向回收缩, 再找出一个新的参数点

$$\theta^{**} = r_c \theta_M + (1 - r_c) c \quad (8.14)$$

式中, r_c 称为收缩系数, $0 \leq r_c \leq 1$ (通常可取为 0.5), 若 $J^{**} < J_{i_0}$ (θ_{i_0} 为除 θ_M 外的某一点), 则用 θ^{**} 代替 θ_M , 转步骤(3); 如果经过收缩后的 θ^{**} 仍然是最坏的, 即 $J(\theta^{**}) > J_i$ ($i \neq M$), 这时则将所有的 θ_i 用 $0.5(\theta_i + \theta_l)$ 代替, 转步骤(2)。

(3) 当

$$\sum_{i=1}^{n+1} (J_i - \bar{J})^2 / (N+1) < \epsilon \quad (8.15)$$

停止搜索, 这里 \bar{J} 是 J_i 的平均值。也就是说, 当 J 的均方误差小于 ϵ (充分小的正数) 时, 停止搜索。否则转步骤(2)。

这一搜索方法的程序框架如图 8.2 所示。

【例 8.1】 化学反应动力学常数估计

在化学反应动力学中, 描述反映速度常用的模型是

$$r = aK e^{-E/(RT)} C \quad (8.16)$$

式中, r 是反映速度, a 是催化剂活性, T 是反映温度, C 是反应物浓度。以上均属可检测的输入、输出变量。 R 是气体常数 (已知); K 是频率因子, E 是活化能, 它们是待估计参数。用非线性最小二乘来估计参数, 即要求估计值 K 和 E , 使

$$J = \sum_{i=1}^N [r_i - a_i K e^{-E/(RT_i)} C_i]^2 \quad (8.17)$$

达到最小。

下面用单纯形搜索法解此问题。

(1) 根据经验, 取初始单纯形的一个顶点

$$(K, E) = (0.16 \times 10^{10}, 1600)$$

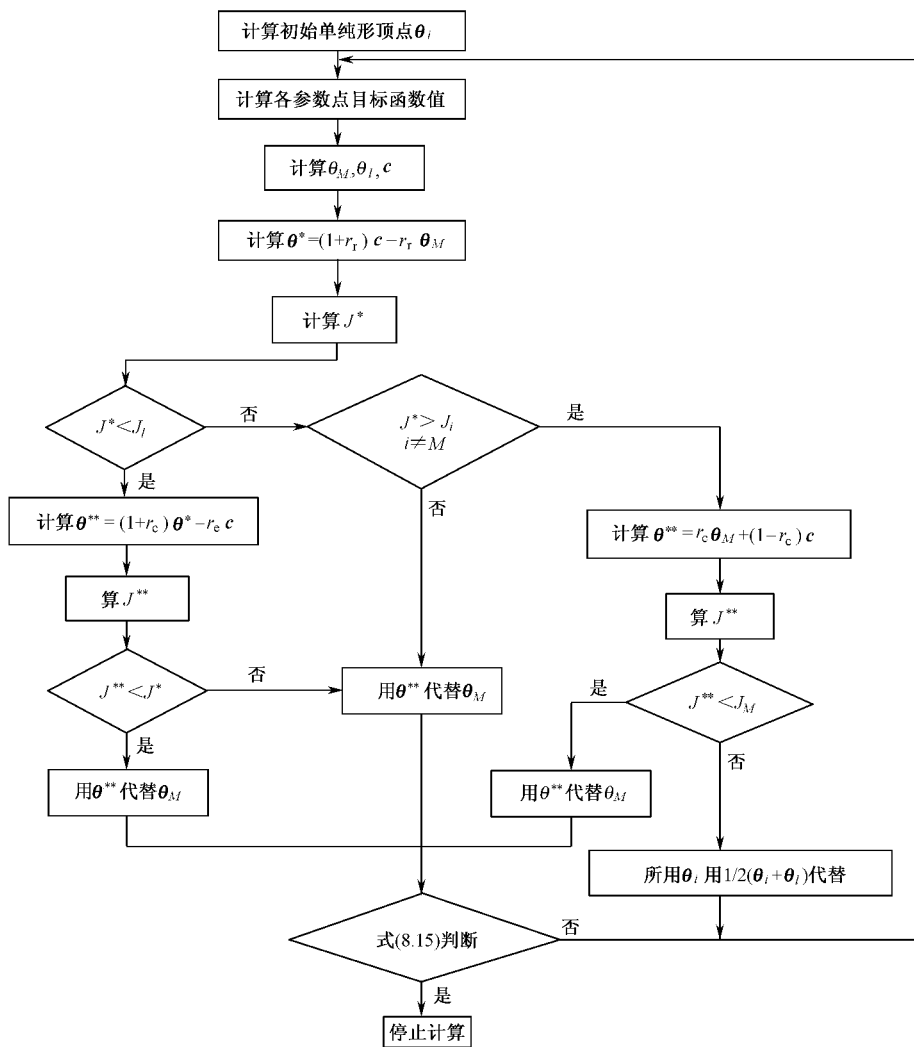


图 8.2 单纯形搜索计算框

取边长 $a=200, r_r=1, r_e=2, r_c=0.5$ 。经过 229 次搜索得到最后结果为：

$$J^* = 861.43, K^* = 0.808 \times 10^{11}, E^* = 19261$$

(2) 如果不用任何先验信息,即完全按表 8.1 决定初始单纯形,即取 3 个参数点为

	K	E
1	0	0
2	193.2	51.8
3	51.8	193.2

在其他条件相同时,需经过 488 次搜索才能得到上述结果。

(3) 初始单纯形的顶点离最优值越近,则搜索越快。

通过上面例题的计算,对算法本身提出几点值得注意的地方供使用该方法时参考。

① 这一搜索法按计算步骤而言,收敛速度并不算快。但是由于它每一步搜索除计算目标函数外,只是一些很简单的代数运算,所以实际上还是很快的。

② 它的主要优点是不需要精确的初始值,甚至在没有任何关于参数的知识时,可按表 8.1 求出初始单纯形顶点的坐标就可以算出结果。这一点是迭代算法所做不到的。

③ 搜索了多步以后,收敛速度显著变慢,再重新进行第二次搜索是有效的。这实际上是重新调整单纯形的边长和步长,有助于收敛。

④ 各参数的数量级应调整到大致相同,否则在搜索中会发生某些参数变动、而另一些参数几乎不变的情况,以致使搜索范围难以扩大。在例 8.1 中,实际计算时将 K 乘以 10^{-6} ,使得两个参数的数量级一致。这样处理在计算过程中证明是必要的。

8.3 迭代算法的基本原理

这里要讨论的问题仍然是求参数估计值 $\hat{\theta}$,使目标函数式(8.8)达到最小。采用的方法是迭代算法,即从某一初始猜测值 $\theta^{(0)}$ 出发,然后沿着使目标函数 $J(\theta)$ 减小的方向产生一系列的参数点 $\theta^{(1)}, \theta^{(2)}, \dots$,这个序列收敛到 θ 。这样,对充分大的 N 就可用 $\theta^{(N)}$ 作为参数 θ 的估计值 $\hat{\theta}$ 。

下面给出迭代算法的一般步骤。

8.3.1 迭代算法的一般步骤

由 $\theta^{(i-1)}$ 计算 $\theta^{(i)}$ 的过程称为是第 i 次迭代,并记

$$J_i = J[\theta^{(i)}], i = 0, 1, 2, \dots \quad (8.18)$$

如果 $J_i < J_{i-1}$,则称第 i 次迭代是可接受的,或者说 $\theta^{(i)}$ 是可接受的。如果迭代过程的每一次都是可接受的,则整个迭代算法称为是可接受的。我们这里仅考虑可接受的方法。

迭代算法的一般步骤是:

- (1) 给出初始猜测值 $\theta^{(0)}$,并置迭代次数 $i=1$;
- (2) 确定一个向量 $\mathbf{v}^{(i)}$ 作为第 i 次的迭代方向;
- (3) 确定一个标量 $\rho^{(i)}$,使得

$$\theta^{(i)} = \theta^{(i-1)} + \rho^{(i)} \mathbf{v}^{(i)} \quad (8.19)$$

是可接受的。这时也称 $\mathbf{v}^{(i)}$ 是可接受方向。

(4) 检查停机规则是否满足。如果不满足,则置 $i=i+1$,再转向步骤(2)开始重复。如果满足,则取 $\theta^{(i)}$ 为 $\hat{\theta}$ 。

从上面迭代算法的一般步骤不难看出,可接受的方向是迭代算法的关键。究竟什么方向是可接受的?如何寻找可接受方向?这将是下面要讨论的问题。

8.3.2 可接受方向

考虑第 i 次迭代。假定沿某个方向 \mathbf{v} 修正 $\theta^{(i-1)}$,于是有

$$\theta(\rho) = \theta^{(i-1)} + \rho \mathbf{v}, \rho \geq 0 \quad (8.20)$$

记

$$\Psi_b(\rho) \equiv J[\theta(\rho)] = J[\theta^{(i-1)} + \rho \mathbf{v}] \quad (8.21)$$

$$\frac{d\Psi_b(\rho)}{d\rho} = \left(\frac{\partial J}{\partial \theta}\right)^T \frac{\partial \theta}{\partial \rho} = \left(\frac{\partial J}{\partial \theta}\right)^T \mathbf{v} \quad (8.22)$$

式中, $\frac{\partial J}{\partial \theta}$ 是 $J(\theta)$ 的梯度向量, 记为 $\mathbf{Q}(\theta)$ 。记 $\frac{\partial J}{\partial \theta}$ 在 $\theta^{(i)}$ 处取值的梯度向量为 \mathbf{Q}_i , 于是有

$$\Psi_b \triangleq \left. \frac{d\Psi_b(\rho)}{d\rho} \right|_{\rho=0} = \mathbf{Q}_{i-1}^T \mathbf{v} \quad (8.23)$$

这里假定 $\mathbf{Q}_{i-1} \neq 0$ 。

如果 $\Psi_b < 0$, 则对充分小的 ρ , $\rho\mathbf{v}$ 是可接受的, 这时称 \mathbf{v} 是可接受方向; 如果 $\Psi_b \geq 0$, 则不存在 $\rho > 0$ 使得 $\rho\mathbf{v}$ 是可接受的。关于可接受方向有如下定理。

定理 8.1 一个方向 \mathbf{v} 是可接受的, 当且仅当存在一个正定矩阵 \mathbf{R} , 使得

$$\mathbf{v} = -\mathbf{R}\mathbf{Q}_{i-1} \quad (8.24)$$

证明 充分性: 设 \mathbf{R} 是正定矩阵, \mathbf{v} 满足式(8.24), 则由式(8.23)有

$$\Psi_b = \mathbf{Q}_{i-1}^T \mathbf{v} = -\mathbf{Q}_{i-1}^T \mathbf{R}\mathbf{Q}_{i-1} < 0$$

所以 \mathbf{v} 是可接受方向。

必要性: 假定 \mathbf{v} 是可接受方向, 即满足 $\Psi_b = \mathbf{Q}_{i-1}^T \mathbf{v} < 0$ 。取

$$\mathbf{R} = [\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}^T / (\mathbf{Q}_{i-1}^T \mathbf{Q}_{i-1})] - \mathbf{v}\mathbf{v}^T / (\mathbf{v}^T \mathbf{Q}_{i-1})$$

则直接验算可知: $\mathbf{R}\mathbf{Q}_{i-1} = -\mathbf{v}$, 因此只需证明 \mathbf{R} 是正定矩阵即可。因为 $\mathbf{v}\mathbf{v}^T$ 非负定, 而 $\mathbf{v}^T \mathbf{Q}_{i-1} < 0$, 所以 \mathbf{R} 表达式的第三项是非负定的。现在再来看前两项。对任意非零向量 \mathbf{x} , 有

$$\begin{aligned} \mathbf{x}^T [\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}^T / (\mathbf{Q}_{i-1}^T \mathbf{Q}_{i-1})] \mathbf{x} &= \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{Q}_{i-1} \mathbf{Q}_{i-1}^T \mathbf{x} / (\mathbf{Q}_{i-1}^T \mathbf{Q}_{i-1}) \\ &= \sum_j x_j^2 - \left(\sum_j x_j q_{i-1,j} \right)^2 / \sum_j q_{i-1,j}^2 \end{aligned}$$

式中, $q_{i-1,j}$ 是向量 \mathbf{Q}_{i-1} 的第 j 个分量。

根据 Cauchy 不等式知

$$\left(\sum_j x_j q_{i-1,j} \right)^2 \leq \sum_j x_j^2 \cdot \sum_j q_{i-1,j}^2$$

而等号仅当 \mathbf{x} 与 \mathbf{Q}_{i-1} 成比例时才成立。如 \mathbf{x} 与 \mathbf{Q}_{i-1} 不成比例, 则显然有

$$\mathbf{x}^T [\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}^T / (\mathbf{Q}_{i-1}^T \mathbf{Q}_{i-1})] \mathbf{x} > \sum_j x_j^2 - \sum_j x_j^2 \cdot \sum_j q_{i-1,j}^2 / \sum_j q_{i-1,j}^2 = 0$$

从而有 $\mathbf{x}^T \mathbf{R} \mathbf{x} > 0$ 。如果 \mathbf{x} 与 \mathbf{Q}_{i-1} 成比例, 即有 $\mathbf{x} = c\mathbf{Q}_{i-1}$, c 为常数, 这时上式等于零。对这种情形再看第三项: $\mathbf{x}^T \mathbf{v}\mathbf{v}^T \mathbf{x} / (\mathbf{v}^T \mathbf{Q}_{i-1}) = c^2 \mathbf{v}^T \mathbf{Q}_{i-1} < 0$, 所以仍有 $\mathbf{x}^T \mathbf{R} \mathbf{x} > 0$ 。综合上述讨论可知, 对任意非零向量 \mathbf{x} , 皆有 $\mathbf{x}^T \mathbf{R} \mathbf{x} > 0$, 即证明了矩阵 \mathbf{R} 是正定的。

按照这个定理, 可接受迭代算法的第 i 次迭代的结果应是

$$\theta^{(i)} = \theta^{(i-1)} - \rho_i \mathbf{R}_i \mathbf{Q}_{i-1} \quad (8.25)$$

不同的方法只是选择不同的正定矩阵 \mathbf{R}_i 和步长 ρ_i 。

应该指出的是, 一个可接受的算法只能保证目标函数值是下降的, 但是否能收敛到最小值呢? 另外, 即使目标函数能收敛到最小值, 但能否保证参数也能收敛到真值呢? 这要讨论方法的收敛性问题, 这方面的进一步讨论可参看 Band(1975)。

8.4 牛顿—拉夫森算法

在讨论非线性最小二乘时, 目标函数 $J(\theta)$ 关于 θ 是非线性的。

考虑在 $\theta^{(i)}$ 附近将 $J(\theta)$ 展成二次函数, 并令

$$\phi_i(\theta) = J(\theta^{(i)}) + Q_i^T(\theta - \theta^{(i)}) + \frac{1}{2}(\theta - \theta^{(i)})^T H_i(\theta - \theta^{(i)}) \quad (8.26)$$

其中, $Q_i = \left. \frac{\partial J(\theta)}{\partial \theta} \right|_{\theta^{(i)}}$, H_i 是由 $J(\theta)$ 的二阶偏导数组成的黑森(Hessian)矩阵, 它的元素是 $H_i(k, l) = \partial^2 J / (\partial \theta_k \partial \theta_l)$ 在点 $\theta^{(i)}$ 的值, 这里 θ_k 是参数向量 θ 的第 k 个分量。

令

$$\frac{\partial \phi_i}{\partial \theta} = Q_i + H_i(\theta - \theta^{(i)}) = 0$$

如果 H_i 是满秩的, 则有解

$$\theta^{(i+1)} = \theta^{(i)} - H_i^{-1} Q_i \quad (8.27)$$

式(8.27)所示的迭代算法称为牛顿—拉夫森(Newton-Raphson)方法。它相当于在一般的迭代格式式(8.25)中取 $\rho_i = 1$, $R_i = H_i^{-1}$, 但在实际应用时, 为了提高收敛速度, 也经常增加一个标量因子 ρ , 即式(8.27)变为

$$\theta^{(i+1)} = \theta^{(i)} - \rho H_i^{-1} Q_i$$

牛顿—拉夫森法的计算步骤如下:

- (1) 猜测初始值 $\theta^{(0)}$; 置 $i=0$, 并设步长 ρ ;
- (2) 计算梯度向量 Q_i 和矩阵 H_i ;
- (3) 解方程 $H_i v = -Q_i$, 得 v_i ;
- (4) 计算新的参数估计值: $\theta^{(i+1)} = \theta^{(i)} - \rho v_i$;
- (5) 如果满足要求则结束, 并取 $\hat{\theta} = \theta^{(i+1)}$; 否则置 $i=i+1$, 转步骤(2)继续。

如果 $J(\theta)$ 是二次函数, 即 $J(\theta)$ 与 $\phi(\theta)$ 是一致的, 则只要 H_i 是正定的, 一次迭代即可达到最小值点, 因此这一方法有收敛快的优点。但是它也存在缺点, 主要是除非参数 θ 在 $J(\theta)$ 的最小值附近, 不一定能保证 $H(\theta)$ 是正定的, 另外, H 的计算也太复杂。为了克服这些缺点, 提出了各种改进方法, 这里只讨论一种麦夸特(Marquardt)方法。

8.5 麦夸特方法

这里讨论的问题仍然是非线性最小二乘估计问题, 目标函数

$$J = \sum_{i=1}^N (y(i) - f[u(i), \theta])^2 \quad (8.28)$$

为了克服对 θ 的交度非线性, 牛顿—拉夫森方法是将目标函数 J 展开成二次函数, 而这里考虑的办法是将 f 展开成线性函数(在 $\theta^{(i)}$ 附近), 即

$$f(u, \theta) = f(u, \theta^{(i)}) + \frac{\partial f^T}{\partial \theta^{(i)}}(\theta - \theta^{(i)})$$

其中

$$\frac{\partial f}{\partial \theta^{(i)}} \triangleq \left. \frac{\partial f}{\partial \theta} \right|_{\theta^{(i)}}$$

这样, 目标函数式(8.28)可化为(仍用符号 J)

$$J = \sum_{k=1}^N \left[y(k) - f(u(k), \theta^{(i)}) - \frac{\partial f^T}{\partial \theta^{(i)}}(\theta - \theta^{(i)}) \right]^2$$

这时

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -2 \sum_{k=1}^N \frac{\partial f}{\partial \boldsymbol{\theta}^{(i)}} \left[\mathbf{y}(k) - f(\mathbf{u}(k), \boldsymbol{\theta}^{(i)}) - \frac{\partial f^T}{\partial \boldsymbol{\theta}^{(i)}} (\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}) \right]$$

令 $\frac{\partial J}{\partial \boldsymbol{\theta}} = 0$, 可解得

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{(i)} + \left[\sum_{k=1}^N \left(\frac{\partial f}{\partial \boldsymbol{\theta}^{(i)}} \frac{\partial f^T}{\partial \boldsymbol{\theta}^{(i)}} \right) \right]^{-1} \sum_{k=1}^N \frac{\partial f}{\partial \boldsymbol{\theta}^{(i)}} \times [\mathbf{y}(k) - f(\mathbf{u}(k), \boldsymbol{\theta}^{(i)})] \quad (8.29)$$

令

$$A_i = \sum_{k=1}^N \left[\frac{\partial f}{\partial \boldsymbol{\theta}^{(i)}} \frac{\partial f^T}{\partial \boldsymbol{\theta}^{(i)}} \right] \quad (8.30)$$

而

$$\sum_{k=1}^N \frac{\partial f}{\partial \boldsymbol{\theta}^{(i)}} [\mathbf{y}(k) - f(\mathbf{u}(k), \boldsymbol{\theta}^{(i)})] = -\mathbf{Q}_i \quad (8.31)$$

注意, 这里的 \mathbf{Q}_i 就是梯度方向 (由式 (8.28) 直接可得), 因此, 由定理 8.1 可知, 要使所得到的方向是可接受的, 只要矩阵 $A_i^{-1} \triangleq \mathbf{R}_i$ 是正定矩阵, 但是这一点并不是总能做到的。为了保证方向 $\mathbf{v} = -\mathbf{R}_i \mathbf{Q}_i$ 是可接受的, 就需对 A_i 做一些修正。办法是将 A_i 修正为 $A_i + \lambda \mathbf{P}_i$, 其中 $\lambda > 0$, \mathbf{P}_i 是一个正定矩阵。容易证明, 只要 λ 充分大, $A_i + \lambda \mathbf{P}_i$ 就一定是正定矩阵。

最简单地取 \mathbf{P}_i 为单位阵, 即 $\mathbf{P}_i = \mathbf{I}$, 于是 $\mathbf{R}_i = (A_i + \lambda \mathbf{I})^{-1}$, 取 $\rho_i = 1$, 即有

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \mathbf{R}_i \mathbf{Q}_i \quad (8.32)$$

正如上面指出的, 只要 λ 充分大, 就能使方向 $-\mathbf{R}_i \mathbf{Q}_i$ 是可接受的; 另一方面, 当 λ 很小时, 方向接近于 $-\mathbf{A}_i^{-1} \mathbf{Q}_i$ (高斯方向)。前者实际上是梯度方向, 可接受但收敛较慢, 后者如果收敛则速度较快。因此, 关键在于选择 λ , 使在保证 $-\mathbf{R}_i \mathbf{Q}_i$ 可接受的前提下 λ 尽可能小。针对这一问题提出下述麦夸特方法。

令 λ_i 是第 i 次迭代所确定的 λ , 则第 i 次迭代用 λ_i 所确定的参数的校正值是

$$\mathbf{v} = (A_i + \lambda_i \mathbf{I})^{-1} \mathbf{Q}_i \quad (8.33)$$

并由此得到第 $i+1$ 次迭代的参数值为

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \mathbf{v}$$

在迭代的每一步适当调整 λ_i , 以保证算法收敛。其具体步骤如下:

(1) 猜测初始值 $\boldsymbol{\theta}^{(1)}$, 计算 $J_1 = J(\boldsymbol{\theta}^{(1)})$; 置 $\lambda_0 = 1, i = 1$;

(2) 计算 A_i, \mathbf{Q}_i ;

(3) 计算 $\mathbf{v} = -(A_i + (\lambda_{i-1}/10)\mathbf{I})^{-1} \mathbf{Q}_i, \hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(i)} + \mathbf{v}, \hat{J} = J(\hat{\boldsymbol{\theta}})$;

(4) 若 $\hat{J} < J_i$, 则置 $\boldsymbol{\theta}^{(i+1)} = \hat{\boldsymbol{\theta}}$, 然后转步骤(6); 否则继续;

(5) 用 $10 \cdot \lambda_{i-1}$ 代替 λ_{i-1} 并转步骤(3);

(6) 若 $\max_{1 \leq j \leq n} \left\{ \frac{\Delta \theta_j^{(i)}}{\theta_j^{(i)}} \right\} < \epsilon$, 其中 $\theta_j^{(i)}$ 是 $\boldsymbol{\theta}^{(i)}$ 的第 j 个分量, $\Delta \theta_j^{(i)} = \theta_j^{(i+1)} - \theta_j^{(i)}$, 则迭代停止, 取

$\boldsymbol{\theta}^{(i+1)}$ 为所要的结果。否则置 $i = i+1$, 转步骤(2)。

对上述算法作几点说明:

① 在上述算法中, 每一次求 \mathbf{v} 时, 实际上是解方程组

$$(A_i + \lambda_i \mathbf{I}) \mathbf{v} = \mathbf{Q}_i \quad (8.34)$$

② 为了改善数值计算的状况, 可对式 (8.34) 中的量进行标度变换。记

$$\mathbf{A}_i^* \triangleq (\mathbf{A}_i + \lambda_i \mathbf{I}) , \mathbf{Q}^* = (q_1, \dots, q_n)^T$$

对 $\{a_{ij}\}$ 和 $\{q_i\}$ 进行标度变换, 并记

$$a_{ij}^* = a_{ij} / (\sqrt{a_{ii}} \sqrt{a_{jj}}) \quad (8.35)$$

$$q_i^* = q_i / \sqrt{a_{ii}} \quad (8.36)$$

这样式(8.34)变为

$$\mathbf{A}_i^* \mathbf{v}^* = \mathbf{Q}_i^* \quad (8.37)$$

将所求得的 \mathbf{v}^* 经过反标度变换即得

$$\mathbf{v}_j = \mathbf{v}_j^* / \sqrt{a_{jj}} \quad j = 1, \dots, n \quad (8.38)$$

(3) 在实际编程计算时, 需对最大迭代次数及 λ 的最大值给予阈值, 以免造成死循环。

麦夸特算法框图如图 8.3 所示。下面举例说明它的应用。

【例 8.2】 问题同例 8.1, 用麦夸特方法估计反应动力学常数 K 和 E 。

$$\frac{\partial f}{\partial K} = a \cdot e^{-E/(RT)} C$$

$$\frac{\partial f}{\partial E} = -aK/(RT) e^{-E/(RT)} C$$

$$\frac{\partial f[\mathbf{u}(i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} = \begin{bmatrix} a(i) e^{-E/(RT(i))} C(i) \\ -Ka(i)/(RT(i)) e^{-E/(RT(i))} C(i) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \sum_{i=1}^N a^2(i) e^{-2E/(RT(i))} C^2(i) & -\sum_{i=1}^N \frac{a^2(i)K}{RT(i)} e^{-2E/(RT(i))} C^2(i) \\ -\sum_{i=1}^N \frac{a^2(i)K}{RT(i)} e^{-2E/(RT(i))} C^2(i) & \sum_{i=1}^N a^2(i) e^{-2E/(RT(i))} \left(\frac{K}{RT(i)} \right) \end{bmatrix}$$

计算结果如表 8.2 和表 8.3 所示。

表 8.2 初始参数取 $(10^{11}, E^{(0)})$ 的计算结果

$E^{(0)}$	$J^{(0)}$	J_{\min}	迭代次数	\hat{K}	\hat{E}
15000	1.8×10^9	861.43	218	0.808×10^{11}	19261
16000	1.9×10^8	861.43	99	同上	同上
17000	1.9×10^7	861.43	8	同上	同上
18000	1.5×10^6	861.43	7	同上	同上
19000	43423	861.43	6	同上	同上
20000	2137.8	861.43	6	同上	同上

当 $E^{(0)}$ 小于 15000 或大于 21000 时就收敛到局部极值或不收敛。

表 8.3 初始参数取 $(K^{(0)}, 17000)$ 的计算结果

$K^{(0)}$	$J^{(0)}$	J_{\min}	迭代次数	\hat{K}	\hat{E}
10^1	2.24×10^{11}	861.43	40	0.808×10^{11}	19261
10^{20}	2.24×10^{25}	861.43	12	同上	同上
10^{30}	2.24×10^{35}	861.43	45	同上	同上

从以上结果可以看出, 只要初始参数选得合适, 麦夸特方法收敛还是很快的。其次, 由于

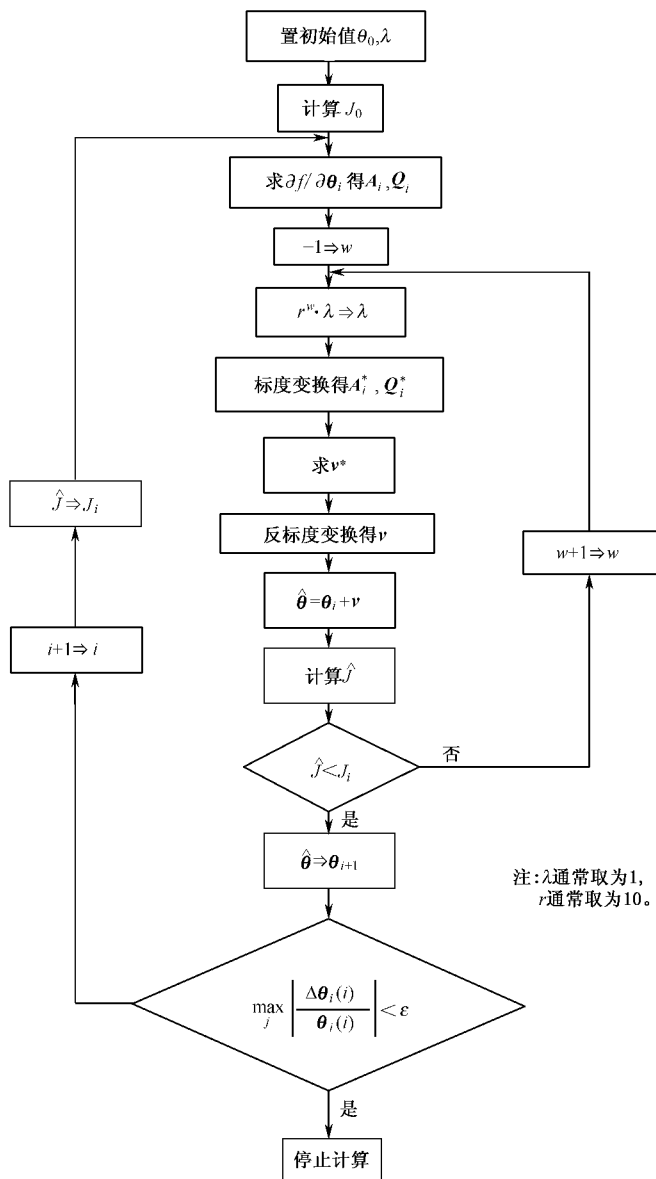


图 8.3 麦夸特方法计算框图

参数 E 属非线性参数, 所以它的收敛范围也比较窄。 K 是线性的, 所以它的收敛范围是很宽的。

本章所讨论的是已知结构的非线性模型的参数估计, 实质上是非线性的最小二乘法。所讨论的算法都属于优化算法, 因此, 有关优化的算法在这里都可以使用。另外, 这里讨论的模型虽然是静态的, 但是连续、动态模型的参数估计也往往归结为解非线性最小二乘估计问题。

* 8.6 数据处理的分组方法(GMDH)

8.6.1 背景

本章前几节所介绍的非线性系统辨识方法,由于模型结构已知,实际上转换成了非线性参数估计问题。但在许多实际系统中,如社会系统、经济系统、生物系统和环保系统等,甚至在许多工业系统中,其非线性模型结构是很难事先给定的,而且随着科学技术和社会生产的不断发展,需要进行研究和处理的系统也变得越来越复杂,应用现在普遍采用的模型类型和建模方法难于解决更为复杂的现实问题。

自从 Wiener 的控制论问世以来,从研究方向和方法论上都开始突破原有的狭窄范畴,朝着将生物的、心理的、物理的、社会经济的各个领域贯通起来的方向发展。从对不同系统的研究中借鉴、移植新的概念、思想和方法已成为促进边缘科学发展的重要动力。

GMDH(Group Method of Data Handling) 是对复杂系统的一种启发式自组织建模方法。它是依瓦连柯教授(Ivakhnenko,1970)借助生物控制论中的自组织原理,用启发式的方法提出的一套建模方法,因此也有人将该方法列入自组织神经网络范畴。这种方法的思路完全不同于前面介绍的其他方法,为了能清楚地理解这种方法,先介绍一些有关生物学的背景是必要的。

神经网络的活动是一种典型的自组织化行动,如图 8.4 所示(关于神经网络模型较详细的内容将在第 11 章中介绍,这里仅对与 GMDH 方法有关的内容做一简单说明)。感受器 S 接收到刺激 x_1, \dots, x_m 后,通过随机组合将超过一定阈值的信号到达 A_1 感受器,然后再由 A_1 通过随机组合线路将超过一定阈值的信号传达到下一个感受器 A_2 ,这样一级一级地传送下去,直到最后的感受器 R。这就是外界刺激在神经网络中传送的一个基本模式,这样传送刺激的结果取决于信号进行组合的方式与强度,以及通过每一感受器的阈值。

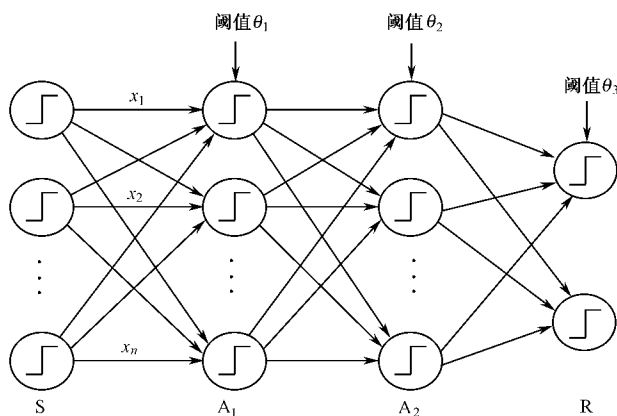


图 8.4 神经网络活动模型示意图

其次,再来讨论品种改良的假说。品种改良的一般过程是通过培育得到新的品种,在新的品种中挑选优良者进行培育,再得到新一代的品种,再挑选优良者继续培育,这样经过若干代培育挑选,就可以得到真正优良的新品种。这样一个过程有以下一些共同点。

① 每一代选用一定比例的品种留做培养下一代用。这个比例如果安排得不恰当,就可能

使品种改良推迟,甚至反而使新品种的品质变差。

② 在一代内完成品种改良是不可能的,需要经过数代才能完成。

③ 长期的品种改良可能会带来植物的退化。

④ 品种改良的问题越是复杂,越要经过很多代的淘汰。

以上这些原则不能用数学进行严格的证明,可将其称为“假说”。

在这些原则的启发下,可以考虑一个分层筛选、层间进行组合并且部分淘汰的多层信息处理方式来得到的模型,这就是 GMDH 方法的启发式自组织的原理。

8.6.2 一般模型结构及基本原则

多变量回归(包括动态的 ARMA 模型)所研究的问题是要建立对象的输入、输出关系,而在回归分析中,总是人为地给了许多事先的限制,如线性,即使是非线性也是事先规定了模型的形式(这些形式是事先主观决定的),这里采用的是与上述思路完全不同的方法来研究同样的问题,即用近似的模型去拟合输入、输出数据。

设系统的 m 个输入变量为 x_1, \dots, x_m , 输出为 y , 最终模型

$$y = f(x_1, x_2, \dots, x_m)$$

称为系统的“完全实现”。

先来看一个单输入单输出系统,输出是 $y(t)$, 输入是 $x(t)$, 根据已有的知识知道它们之间的关系是非线性的。

一般地,任意的非线性函数 f 可展成输入 x 的 Volterra 级数(Roy R A and Sherman J A, 1967), 即

$$\begin{aligned} y(t) = & \int_0^t h_1(\tau_1)x(t-\tau_1)d\tau_1 + \int_0^t \int_0^t h_2(\tau_1, \tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \\ & \int_0^t \int_0^t \int_0^t h_3(\tau_1, \tau_2, \tau_3)x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)d\tau_1 d\tau_2 d\tau_3 + \dots \end{aligned} \quad (8.39)$$

将式(8.39)变成离散型多项式,称为多变量非线性多项式,即

$$y(t) = \sum_{i=1}^N a_i x_i + \sum_{i=1}^N \sum_{j=1}^N a_{ij} x_i x_j + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N a_{ijk} x_i x_j x_k + \dots \quad (8.40)$$

其中, $x_i = x(t-i\tau)$ 是输入 $x(t)$ 的 i 步滞后,这就是把 $x(t)$ 和自身的慢变化过程看成是一多变量输入组,这样就得到前述所谓的非线性函数的 Kolmogorov-Gavor 多项式,简称 K-G 多项式,也称之为非线性函数的一个“完全实现”。

然而,这样的 K-G 多项式完全实现,随着多项式阶次的增加,项数也以等比级数增加,由输入、输出数据应该估计的系数的数目也变得很大。例如,输入变量为 10, 若多项式阶次为 3, 那么 K-G 多项式的参数数目为

$$a_i : 10, \quad a_{ij} : 55, \quad a_{ijk} : 220$$

共 285 个参数;如多项式阶次增加到 10, 那么参数总数将达 10 万个。为了估计这 10 万个参数,需要有相当可观数量的输入、输出数据。不仅计算量非常大,更重要的是解高阶方程组会带来数值计算上的一系列难以克服的问题。

为了克服这些困难,而且要得到完全的实现式(8.40),按照启发式自组织原则,可以考虑由一系列的“部分实现”来达到。所谓“部分实现”,就是由任意两个输入变量构成的函数。即

$$y_1 = G(x_1, x_2)$$

$$y_2 = G(x_3, x_4)$$

...

$$y_s = G(x_{m-1}, x_m)$$

在第一层用输入、输出数据拟合得到各部分实现,再用上一层的输出作为下一层的输入,做出下一层的部分实现

$$z_1 = G(y_1, y_3)$$

$$z_2 = G(y_3, y_4)$$

每一层保留一部分好的部分实现,舍去一部分坏的部分实现。这样一层一层地选下去,直到找出近似程度满意的部分实现为止,最后就以这个最优的部分实现作为完全实现。这样每一层的部分实现就相当于产生新的品种,将其淘汰掉一部分,再按神经网络的活动方式产生下一层的新品种,经过一层(代)一层(代)的选择,最后得到所要的优良品种。由于在实际计算时,每次只是计算部分实现,所以数据量和计算量的矛盾都很容易解决。

为了实现上述目标,需要解决以下几个问题:

- (1) 确定输入变量的个数;
- (2) 确定部分实现 G 的具体结构形式;
- (3) 如何使用数据;
- (4) 每一层如何选择部分实现(选择规则);
- (5) 如何确定最优部分实现已经选出(停止规则)。

近十几年来,各国学者主要是围绕上述的几个问题对 GMDH 方法进行改进和研究的。

8.6.3 基本的 GMDH 方法

考虑 n 个输入变量为 x_1, \dots, x_n (也可以是同一输入的不同时间的值),输出变量为 y ,描述输入、输出关系的“完全实现”是

$$y = f(x_1, x_2, \dots, x_n)$$

基本的 GMDH 方法一般取二阶多项式作为“部分实现” G ,即取

$$G(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j \quad (8.41)$$

对所有产生的中间变量,按平方误差准则进行选择。误差低于某一阈值的留下,淘汰掉其他的,对留下的变量 y_i ,再产生第二层中间变量

$$z_j = G(y_i, y_k)$$

再从 z_j 中淘汰掉一部分变量,用选出的变量继续生成第三层的中间变量。这样继续进行,直到只剩下一个变量或者到某个事先规定的阶次时停止,得到的最后模型也就是“完全实现”。

现在用一个很简单的例子来说明上述过程。设第一层有 4 个输入 x_1, x_2, x_3, x_4 ,由变量组合 $(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4), (x_3, x_4)$ 组成了 6 个部分实现,它们的输出就是 6 个部分实现 y_1, \dots, y_6 。经过选择,如果只有 3 个,比如 y_1, y_2, y_3 保留,则以 y_1, y_2, y_3 作为下一层的输入可组成 3 个部分实现 $(y_1, y_2), (y_1, y_3), (y_2, y_3)$,它们的输出分别是 z_1, z_2, z_3 ,都是 x_i 的 4 阶多项式,再继续选下去,直到满足要求为止。

从上面的过程可以看出,不管完全实现如何复杂,但每个部分实现只需估计 6 个参数,计算非常简单;而如果直接拟合一个 4 个变量的 4 阶多项式,就需要同时估计 70 个参数,或者说要求一个 70×70 矩阵的逆,这个计算量是相当可观的。

GMDH 算法的示意图如图 8.5 所示。图中, G 是部分多项式; y_{ij} 是由部分多项式模型得到的输出; x_{ij} 是中间变量。

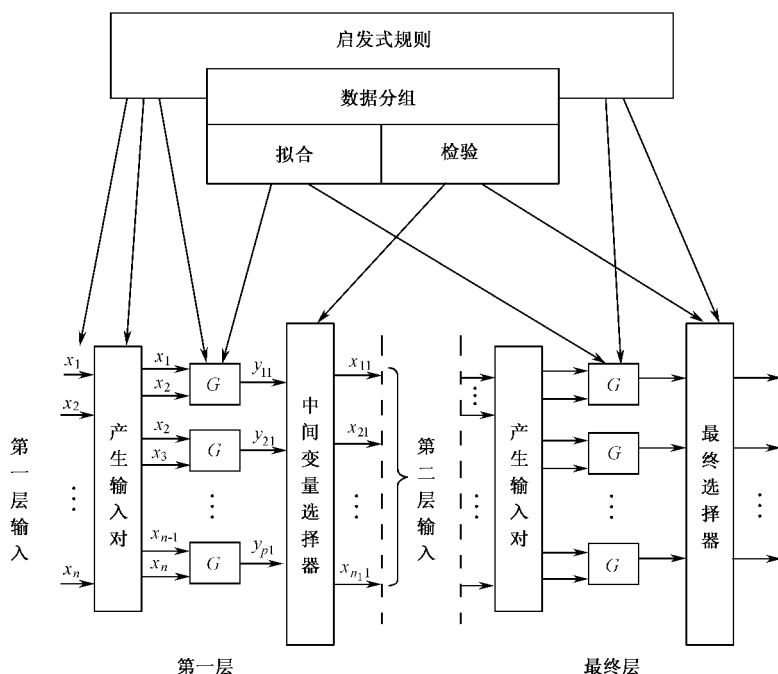


图 8.5 基本 GMDH 方法示意图

基本 GMDH 算法可归纳为以下基本步骤：

- (1) 对最初的输入变量进行预备性的选择；
- (2) 产生以两个变量为基础的“部分实现”；
- (3) 为了决定部分实现的系数和对中间变量的正确选择，将数据分为拟合组和检验组，给出数据分组的原则和方法；
- (4) 按图 8.5 所示的多层结构给出各层的阈值或检验准则，使得在算法中能自动选择各中间变量；
- (5) 最终得到完全实现。

8.6.4 变量的预选择

输入变量的选择对模型的优劣无疑是有很大影响的。对一般的物理、化学及其他工业生产过程，输入变量是清楚的，而且定量化也很容易的。但是，在其他的系统中，特别是社会经济系统、环境系统等，变量之间的关系往往受人为因素的影响，如受政策的影响等。一般来说，得到理论模型是不大可能的，从定量化上也有很多困难。要研究这样的系统，选择输入变量是十分重要的。例如，在经济系统中需要找出影响大的变量，舍弃那些关系不密切的变量，同时还需要确定输入变量对输出影响的滞后时间。常用的方法是用相关分析法选出与输出相关度大的变量。最近研究表明，用“新息—贡献”准则更为合理，该准则将在 8.7 节介绍。

8.6.5 数据的分组和部分实现检验准则

在 GMDH 算法中，原则上将使用的数据分成 3 组：①拟合（估计）组，用来估计每一部分实现的系数；②检验组，用来在每一层检验，比较选择模型；③评估组，在得到模型后用来评价

模型的优良性。

评估组只起评价作用而对模型的选择无关,所以下面就不再讨论了。重点放在讨论拟合组和检验组如何来分,特别在数据较少时(这在 GMDH 方法所处理的问题中是经常遇到的)。下面讨论两种效果较好的且较常用的分组方法。

1. DUPLEX——将数据分开的一种方法

这种方法的基本思想是使两组数据基本上覆盖相同的区域且具有相同的统计特性。

具体的做法是将数据先进行正规化(无量纲化),然后计算每个(组)数据之间的欧氏距离,将距离最大的两个(组)数据分配到拟合组中;然后将与拟合组距离最大的数据分配到检验组中,将与检验组距离最大的数据分配到拟合组中,这样一直进行到所有数据都被分配到这两个组之一中去为止。

究竟要多少数据才能分开呢?为了使在检验时误差平方和有一定的自由度,所以数据不能太少,一个经验规则是数据总数 $N > 2p + 25$, p 是参数的最大数目。

这种做法的缺点是可能带来估计方差的增大(因为数据少了)。在数据太少的情况下,可采用第 2 种方法。

2. PESS——预测误差平方和法

PESS 的拟合组和检验组都用整个数据去定义。用一个系统的方法去改变分配到两个组的数据,如用 $N-1$ 个数据点为拟合集,用一个点为检验集,每一个数据都做一次检验集,重复 N 次。对于部分实现,定义预测误差平方和为

$$\text{PESS} = \sum_{t=1}^N [y(t) - \hat{y}^*(t)]^2 \quad (8.42)$$

其中

$$\begin{aligned} \hat{y}^*(t) = & a_{0t} + a_{1t}x_i(t) + a_{2t}x_j(t) + a_{3t}x_i^2(t) \\ & + a_{4t}x_j^2(t) + a_{5t}x_i(t)x_j(t), t = 1, 2, \dots, N \end{aligned}$$

是通过除第 t 次量测外的所有数据用多元回归得到的部分多项式对第 t 次输出 $y(t)$ 的估计值。 N 是数据长度。

式(8.42)可以简化为

$$\text{PESS} = \sum_{t=1}^N \left(\frac{y(t) - \hat{y}(t)}{1 - \mathbf{x}_t^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_t} \right)^2 \quad (8.43)$$

式中, $\hat{y}(t)$ 是用全部数据通过多元回归得到的对第 t 次输出的估计值。

$$\begin{aligned} \mathbf{x}_t^T = & [1, x_i(t), x_j(t), x_i^2(t), x_j^2(t), x_i(t)x_j(t)], t = 1, 2, \dots, N \\ \mathbf{X}^T = & [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \end{aligned} \quad (8.44)$$

将 GMDH 方法用于单变量时间序列分析时,只要令序列 $y(t)$ 的采样滞后 $y(t-i)$ 作为第一层的输入 x_i ,则式(8.43)可以写为

$$\text{PESS} = \sum_{t=1}^N \left(\frac{y(t) - \hat{y}(t)}{1 - \mathbf{y}_t^T (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{y}_t} \right)^2 \quad (8.45)$$

式中

$$\begin{aligned} \mathbf{y}_t^T = & [1, y(t-i), y(t-j), y(t-i)^2, y(t-j)^2, y(t-i)y(t-j)] \\ \mathbf{Y}^T = & [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N] \end{aligned}$$

8.6.6 选择层——中间变量的选择

选择层是 GMDH 算法中的重要组成部分,它的任务是要决定究竟有多少部分实现(中间变量)可以为下一层构造新的部分实现之用。如果舍弃变量太多,则造成信息不全,使最终模型精度较差;如果保留了重复信息的变量,则不仅使模型复杂,而且使高层模型产生多重共线性,从而产生数值计算问题和模型不稳定。因此,对一个选择层首要的任务是确定检验部分实现的合适的准则,其次是要给出检验的阈值。目前常用的准则有相关准则、AIC 准则和预测误差平方和准则等。选择准则的目标都是试图尽可能避免人为因素对选择变量的影响。8.7 节中介绍的“新息—贡献”准则也许是较好的准则之一(Wang Xiufeng and Li Bo, 1994)。

8.6.7 部分实现的形式

这里讨论的辨识问题的目的是要建立输入变量的多项式模型来逼近一个非线性系统。之所以要引进部分实现,是为了将一个复杂的问题进行简化,通过一层一层地筛选部分实现(也就是“品种改良”的思想)来逼近一个优良的完全实现,即最终模型。

原则上说,部分实现的形式可以是多种多样的。但是,尽管形式可以不同,却都必须保证“简单”这一基本要求。从这一点出发,通常用的部分实现是两个变量的二次完全多项式(式(8.3))。但是这并非是最优选择,文献(Gallo P etc, 1985)给出了另外的部分实现形式。文献(Wang Xiufeng and Liu Dan, 1990)给出了自动选择部分实现的方法,该方法可以作为 8.7 节将要讨论的 NARMAX 模型辨识算法的一个特例。

8.6.8 GMDH 方法总结及应用

在一个很长的时期内,GMDH 方法作为非线性系统辨识的一种重要方法一直受到人们的重视。它吸取了神经网络活动和品种改良过程的精髓,提炼出分层筛选、部分实现这样一种算法结构,再加上对变量(包括中间变量)的选择、数值的优化及合理的使用数据,就有可能使模型的精度达到很高。

GMDH 方法有 3 个基本的优点。

(1) 在数据很少时(这在环境、生态、生理等系统中是常会出现的),如果不用部分多项式而直接计算完全多项式是不可能的。

(2) 在数据很多时,完全多项式的正规方程的系数矩阵几乎总是病态的,但是在小的各种各样部分多项式中总是可以选到好的。

(3) 在完全的多项式中并不是所有的项都是有用的,也就是说,某些中间变量可能是有害的,去掉它反而会提高模型精度,这样的变量在多层优化的选择中就会被淘汰掉。

GMDH 方法已普遍地用于非线性系统的建模与预报中(Ichikawa A, 1978; Kei Hare etc, 1990; 沈心焯等, 1992)。随着提出问题的背景不同,算法的各个部分都在不断出现新的思路和方法(Ivachnenko AG and Muller JA, 1992; 王秀峰, 刘丹, 1990),使其适用范围更加广大,效果也更加明显。

【例 8.3】 河流水质预测模型

河流水质是由许多因素决定的,而各种因素之间又有很强的相互影响,所以很难用一个模型表示水质。要决定流入和流出的污染物的物质平衡是困难的,这些流入和流出的污染物受农业灌溉、城市用水、支流和地表水流入的影响,而这些影响对不同的河流和同一河流的不同

地段都是不一样的。即使这些问题都解决了,要得到水质模型仍然困难。因为收集数据是非常困难的,费用也很高。因此水质模型必须满足:①结构简单且具有相当的可塑性(柔性);②可用少量的数据进行预测。

GMDH 方法十分合适这样的问题。

考虑水质的问题,数据总是受限制的。因为总不可能有足够的数据可用,特别是在日期上连续的数据。这里只能用每周一次或每月一次的数据。水质的指标有很多,这里只以其中之一 BOD(生化氧需求量)的预测问题为例来说明 GMDH 方法的应用。将近 3 年共 70 个数据作为拟合组,共挑选了下面的 14 个变量: x_1 = 某观测站的 COD(化学氧需求量); $x_2 = x_1^2$; x_3 = 该站的径流量; $x_4 = x_1 x_3$; $x_5 = x_3$; $x_6 = 1/x_4^2$; x_7 = 预测点的 COD; $x_8 = x_7^2$; x_9 = 在预测点的 COD 装料; $x_7 x_{10}$; x_{10} = 预测点的径流量; $x_{11} = x_{10}^2$; $x_{12} = 1/x_{10}$; $x_{13} = x_9/x_4$; $x_{14} = x_{10}/x_3$; y = 预测点的 BOD。

计算结果如下(某预测点):

部分实现:

$$y_1 = 22.1 + 0.55x_4 - 3.20x_5 + 0.10x_5^2 - 0.01x_4x_5$$

$$y_5 = 2.67 + 0.55x_4 - 0.02x_{13}$$

完全实现:

$$y = 2.25 - 0.01y_1 + 1.53y_5 - 0.23y_5^2 + 0.20y_1y_5$$

注意上面表达式中缺少的项是因为系数太小而略去。

上面得到的模型可以很好地进行水质指标 BOD 的预测。但是并不能说明模型的机制。因为在不同的观测站和预测点所得到的模型都可能是不一样的。

8.7 NARMAX 模型的辨识

8.7.1 引言

上一节所介绍的 GMDH 方法将一个复杂问题化为一个多层简单问题处理,因此很容易实现,而且可以得到较好的系统描述,所以被广泛地应用。但是,任何一个方法都不可能是十全十美的,在克服某些缺点的同时,也往往会带来另外的问题。GMDH 方法用分层筛选方法可以将一个复杂问题简单化,但由于它将上一层的输出作为下一层的输入,因此造成模型结构柔性较差,从而得到的最终模型往往不是最简单的,甚至对一个简单的非线性过程也可能得到一个复杂的模型(Kortmann M. and Unbehauen H., 1988)。因此,20 世纪 80 年代以来,对于不太复杂的非线性系统,基于 NARMAX (Non-linear Auto Regressive Moving Average model with exogenous inputs) 模型描述的非线性系统的辨识又得到了人们的重视(Billings S A etc 1983; Chen S etc, 1989; Haber R etc, 1990)。

8.7.2 非线性动态系统的描述

一个 r 个输入、 s 个输出的非线性动态系统可以用 s 个单输出、多输入非线性差分方程来描述。而 k 时刻的输出可以用 k 时刻以前一段时间的输入和输出及 k 时刻的输入来表示。记为

$$\mathbf{x}^T(t) = \{u_1(k), \dots, u_1(k-n_1); u_2(k), \dots, u_2(k-n_2); \dots; u_r(k), \dots, u_r(k-n_r); \\ y_1(k-1), \dots, y_1(k-m_1); \dots; y_s(k-1), \dots, y_s(k-m_s)\}$$

$$\underline{\Delta} (x_1, x_2, \dots, x_p) \quad (8.46)$$

式中, $u_i, n_i (i=1, \dots, r)$ 分别表示输入及相应的滞后次数; $y_i, m_i (i=1, 2, \dots, s)$ 分别表示输出和相应的阶次。 x 的维数(即 x 中的项数) p 为

$$p = r + \sum_{i=1}^r n_i + \sum_{i=1}^s m_i \quad (8.47)$$

考虑非线性动态系统的第 t 个输出有如下关系

$$\begin{aligned} y_t(k) = & a_0 + \sum_{i=1}^p a_i x_i(k) + \sum_{i=1}^p \sum_{j=i}^p a_{ij} x_i(k) x_j(k) \\ & + \sum_{i=1}^p \sum_{j=i}^p \sum_{l=j}^p a_{ijl} x_i(k) x_j(k) x_l(k) + \dots \end{aligned} \quad (8.48)$$

式(8.48)称为具有外部输入的非线性自回归滑动平均模型,记为 NARMAX,该模型被广泛地用来作为非线性动态系统的完全描述。若系统是稳定的,一定存在一个充分高的阶次,比如,预计系统的实际阶次不高于 q ,则下述模型可作为系统的一般描述。即

$$\begin{aligned} y(k) = & a_0 + \sum_{i=1}^p a_i x_i(k) + \sum_{i=1}^p \sum_{j=i}^p a_{ij} a_i(k) x_j k + \dots \\ & + \underbrace{\sum_{i=1}^p \dots \sum_{l=v}^p a_{i\dots l} x_i(k) \dots k_v(k) x_l(k)}_q \end{aligned} \quad (8.49)$$

模型式(8.49)中所含的项数为

$$n = (p+q)! / (p! q!) \quad (8.50)$$

为了描述简单,将式(8.49)等号右端的各项分别记为

$$\begin{aligned} \nu_0(k) &= 1 \\ \nu_1(k) &= x_1(k) \\ &\vdots \\ \nu_p(k) &= x_p(k) \\ \nu_{p+1}(k) &= x_1^2(k) \\ \nu_{p+2}(k) &= x_1(k) x_2(k) \\ &\vdots \\ \nu_{n-1}(k) &= x_p^q(k) \end{aligned} \quad (8.51)$$

并记: $\mathbf{v}^T(k) = [\nu_0(k), \nu_1(k), \dots, \nu_{n-1}(k)]$, 相应于式(8.49)的系数向量记为 $\boldsymbol{\theta}^T = [a_0, a_1, \dots, a_{n-1}]$, 则模型式(8.49)的向量形式为

$$y(k) = \mathbf{v}^T(k) \boldsymbol{\theta} \quad (8.52)$$

模型式(8.49)(或式(8.52))关于参数是线性的。似乎可用最小二乘法估计其参数,但实际上是不可行的。这不仅是因为参数太多使计算非常复杂,更重要的是由于许多不同项中存在着重叠信息(因而是有相关性)而使方程变为病态,从而产生数值计算问题,因此该非线性模型的结构确定是非常关键的问题。这里所谓确定非线性模型的结构,即是确定模型的阶次、滞后和重要的项,使模型既简单又能正确地描述系统。这就需要寻求一种合适准则和方法,使重要的项一定被选入模型,而多余的项一定排除在外。“新息—贡献”准则就是一种很有效的准则,借助于该准则可以得到满意的结果,下面将介绍这一准则。

8.7.3 “新息—贡献”准则与矩阵求逆定理

假定取得 N 组数据: $\{y(k); u_i(k), i = 1, \dots, r; k = 1, \dots, N\}$, 用实测数据构造向量

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}, v_0 = \begin{bmatrix} v_0(1) \\ v_0(2) \\ \vdots \\ v_0(N) \end{bmatrix}, \dots, v_{n-1} = \begin{bmatrix} v_{n-1}(1) \\ v_{n-1}(2) \\ \vdots \\ v_{n-1}(N) \end{bmatrix} \quad (8.53)$$

模型式(8.49)(或式(8.52))意味着向量 y 可以表示为向量 v_0, \dots, v_{n-1} 的线性组合,即

$$y = a_0 v_0 + a_1 v_1 + \dots + a_{n-1} v_{n-1} \quad (8.54)$$

但正如上面指出的那样,式(8.54)的右端项 v_0, \dots, v_{n-1} 只是给出了一个范围,确定模型结构和参数估计是要从 v_0, \dots, v_{n-1} 中选择出尽可能少的项来近似表示输出 y ,这是模型结构简单性和模型拟合高精度的合理统一。为实现这一目标,必须寻求一种有效的准则和方法,使重要项一定被选入模型,信息重复项一定排除在外,首要的问题是如何评价项的重要性,为此引入如下“新息—贡献”准则(王秀峰,劳育红,1992)。

假定模型中已选入的项是 v_1, v_2, \dots, v_k , 记矩阵 $R_k = [v_1, \dots, v_k]$, 对剩余的每一项 v , (在最小二乘意义下)它能够由已选入模型的 k 项线性表示的部分为

$$u = R_k (R_k^T R_k)^{-1} R_k^T v \quad (8.55)$$

换句话说, u 表明了 v 所含信息中与已选项重复的部分, 那么 v 所带来的新的信息——“新息”为

$$e = v - u \quad (8.56)$$

y 上 e 的投影模的平方为

$$Q = (y^T e)^2 / (e^T e) \quad (8.57)$$

Q 的大小反映了 v 对于描述 y 的作用的大小,称之为对描述 y 的“贡献”。

考虑某一项是否重要,关键要看它与已选入模型的项相比带来多少“新息”,以及“新息”对描述 y 的“贡献”是否显著,这就是所谓“新息—贡献”准则。应该指出,用“新息—贡献”准则比单纯用“新息”要好,因为“新息”可能包含了噪声,“贡献”更体现了对模型描述系统的作用。

在本节所介绍的算法中,对模型精度影响的显著性判别采用了通常的 F 检验(见 5.1 节)和 BIC 准则

$$\text{BIC} = N \ln S_k^2 + k \ln N \quad (8.58)$$

式中, k 为已选入模型的项数; $S_k^2 = e^T e / N$, 其中, $e = [e(1), \dots, e(N)]^T$ 是模型包含 k 项时所得的残差向量。

为了导出辨识的递推算法,这里我们用到定理 5.1,现重述如下(为了统一起见,我们记它为定理 8.2):

定理 8.2 设 x_1, \dots, x_n, x_{n+1} 是一组同维向量,令

$$\begin{aligned} X_n &= [x_1, x_2, \dots, x_n] \\ X_{n+1} &= [X_n, x_{n+1}] = [x_1, \dots, x_n, x_{n+1}] \\ \Phi_n &= X_n^T X_n, \Phi_{n+1} = X_{n+1}^T X_{n+1} \end{aligned}$$

如果 Φ_n 和 Φ_{n+1} 可逆,则有

$$\Phi_{n+1}^{-1} = \begin{bmatrix} \Phi_n^{-1} + \theta \theta^T / \sigma & -\theta / \sigma \\ -\theta^T / \sigma & 1 / \sigma \end{bmatrix} \quad (8.59)$$

式中, θ 是在最小二乘意义下 x_{n+1} 由 x_1, \dots, x_n 线性表示的组合系数, σ 是误差平方和。

这里给出该定理的两个推论。

推论 8.1 Φ_n, Φ_{n+1} 的定义与定理 8.1 中相同, 若

$$\Phi_{n+1}^{-1} = \begin{bmatrix} \Phi'_n & \alpha \\ \alpha^\top & a \end{bmatrix} \quad (8.60)$$

则

$$\Phi_n^{-1} = \Phi'_n - \alpha^\top \alpha / a \quad (8.61)$$

其中, Φ'_n 是 $n \times n$ 矩阵, α 是 n 维向量, a 为标量。

证明 比较式(8.59)和式(8.60)的相应部分易得

$$\Phi'_n = \Phi_n^{-1} - \theta \theta^\top / \sigma = \theta / \sigma, \alpha = 1 / \sigma$$

所以有

$$\Phi_n^{-1} = \Phi'_n - \theta \theta^\top / \sigma = \Phi'_n - \alpha \alpha^\top / a$$

推论 8.2 向量 x_1, \dots, x_n , 矩阵 Φ_n, X_n 的定义同定理 8.1, 令 $\tilde{X}_i = [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$, $\tilde{\Phi}_i = \tilde{X}_i^\top \tilde{X}_i$, 则

$$\tilde{\Phi}_i^{-1} = \Phi'_{n-1} - \alpha' (\alpha')^\top / a'$$

其中 Φ'_{n-1}, α', a' 由下式确定

$$\Phi'^{-1}_{n-1} = \begin{bmatrix} \Phi'_{n-1} & \alpha' \\ (\alpha')^\top & a' \end{bmatrix} \quad (8.62)$$

这里 Φ'^{-1}_{n-1} 是将 Φ_n^{-1} 的第 i 行移至最后一行, 再将第 i 列移至最后一列而得到的矩阵。

证明 令 $X(i) = [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, x_i]$, $\Phi(i) = X^\top(i) X(i)$, $P = P_{i,i+1} \cdot P_{i+1,i+2} \cdot \dots \cdot P_{n-1,n}$, 这里

$$P_{j,j+1} = \begin{bmatrix} 1 & & & & & & \\ & \dots & & & & & \\ & & 1 & & & & \\ & & & 0 & 1 & & \\ & & & 1 & 0 & & \\ & & & & & 1 & \\ & & & & & & \dots \\ & & & & & & & 1 \end{bmatrix} \begin{array}{l} \leftarrow \text{第 } j \text{ 行} \\ \leftarrow \text{第 } j+1 \text{ 行} \end{array}$$

是第 j 行和第 $j+1$ 行互换的初等变换矩阵。显然有: $X(i) = X_n P$, $\Phi(i) = P^\top \Phi_n P$ 。因此, $\Phi^{-1}(i) = P^{-1} \Phi_n^{-1} (P^\top)^{-1}$, 由于 $P^{-1} = P^\top$, 于是有

$$\Phi^{-1}(i) = P^\top \Phi_n^{-1} P = \Phi'^{-1}_{n-1} = \begin{bmatrix} \Phi'_{n-1} & \alpha' \\ (\alpha')^\top & a' \end{bmatrix}$$

由推论 8.1 可得: $\tilde{\Phi}_i^{-1} = \Phi'_{n-1} - \alpha' (\alpha')^\top / a'$ 。

定理 8.1 给出了由 $n \times n$ 维矩阵 Φ_n 的逆求 $(n+1) \times (n+1)$ 维矩阵 Φ_{n+1} 的逆矩阵的递推算法; 而推论 8.1, 8.2 则给出了已知 $(n+1) \times (n+1)$ 维矩阵 Φ_{n+1} 的逆求 $n \times n$ 维矩阵 Φ_n 的逆的递推算法。

8.7.4 NARMAX 模型的递推辨识算法

这里给出的辨识算法是模型结构和参数同时确定的递推算法。算法的基本思想是: 首先

在所有备选项中选出使“新息—贡献”值最大的项进入模型。以后的每一步都在剩余项中选出使“新息—贡献”值 Q 取最大值的项,若在统计意义下该项对提高拟合精度是显著的,则被选入模型;同时,若由于新增加的项使得某些已选入模型的项变得不重要了,则将不重要的项剔除。然后再从剩余项中选出使 Q 取最大值的项,重复上述步骤,直到既无可选入项又无可剔除项为止,计算结束。

利用 8.7.3 节的定理和推论中给出的矩阵运算技巧,下面给出辨识的递推算法,其具体计算步骤如下:

(1) 给定输入变量的个数 r 和各自的滞后阶次 $n_i, i=1, \dots, r$; 输出变量的个数 s 和各自的阶次 $m_i, i=1, \dots, s$; 模型的最高幂次 q 和数据长度 N ; 给出 F 检验临界值 F_α 。按式(8.50)计算总项数 n ; 按式(8.53)构造向量 $\mathbf{y}, \mathbf{v}_0, \dots, \mathbf{v}_{n-1}$ 。

(2) 置 $k=1, \mathbf{R}_k = [\mathbf{v}_0]; \Phi_k^{-1} = (\mathbf{R}_k^T \mathbf{R}_k)^{-1} = 1/\mathbf{v}_0^T \mathbf{v}_0 = 1/N; \boldsymbol{\theta}_k = \bar{\mathbf{y}} = \frac{1}{N} \sum_{t=1}^N y(t); \mathbf{e}_k = \mathbf{y} - \bar{\mathbf{y}};$
 $S_k^2 = (\mathbf{e}_k^T \mathbf{e}_k)/N, \text{BIC}_k = N \ln S_k^2 + k \ln N, f = \mathbf{e}_k^T \mathbf{e}_k$ 。

(3) 对剩余的 $n-k$ 个向量,记为 $\tilde{\mathbf{v}}_i, i=1, \dots, n-k$, 分别计算

$$\tilde{\boldsymbol{\theta}}_i = \Phi_k^{-1} \mathbf{R}_k^T \tilde{\mathbf{v}}_i; \tilde{\mathbf{e}}_i = \tilde{\mathbf{v}}_i - \mathbf{R}_k \tilde{\boldsymbol{\theta}}_i, Q = (\mathbf{y}^T \tilde{\mathbf{e}}_i)^2 / (\tilde{\mathbf{e}}_i^T \tilde{\mathbf{e}}_i), i = 1, 2, \dots, n-k$$

(4) 选出使 Q_i 取最大值的项记为 \mathbf{v}_{\max} , 相应地 $\tilde{\boldsymbol{\theta}}$ 和 $\tilde{\mathbf{e}}_i$ 记为 $\tilde{\boldsymbol{\theta}}_{\max}$ 和 $\tilde{\mathbf{e}}_{\max}$; 构造矩阵 $\mathbf{R}_{k+1} = [\mathbf{R}_k, \mathbf{v}_{\max}]$, 计算 $\sigma_{\max} = \tilde{\mathbf{e}}_{\max}^T \tilde{\mathbf{e}}_{\max}$ 和

$$\Phi_{k+1}^{-1} = [\mathbf{R}_{k+1}^T \mathbf{R}_{k+1}]^{-1} = \begin{bmatrix} \Phi_k + \tilde{\boldsymbol{\theta}}_{\max} \tilde{\boldsymbol{\theta}}_{\max}^T / \sigma_{\max} & -\tilde{\boldsymbol{\theta}}_{\max} / \sigma_{\max} \\ -\tilde{\boldsymbol{\theta}}_{\max}^T / \sigma_{\max} & 1 / \sigma_{\max} \end{bmatrix}$$

(5) 计算将 \mathbf{v}_{\max} 加入模型后的参数估计、误差和信息准则值

$$\boldsymbol{\theta}_{k+1} = \Phi_{k+1}^{-1} \mathbf{R}_{k+1}^T \mathbf{y}, \mathbf{e}_{k+1} = \mathbf{y} - \mathbf{R}_{k+1} \boldsymbol{\theta}_{k+1}$$

$$f_{k+1} = \mathbf{e}_{k+1}^T \mathbf{e}_{k+1}, F = \frac{f_k - f_{k+1}}{f_{k+1}} (N - k - 1)$$

$$S_{k+1}^2 = f_{k+1} / N, \text{BIC}_{k+1} = N \ln S_{k+1}^2 + (k+1) \ln N$$

(6) 检验: 如果 $\text{BIC}_{k+1} < \text{BIC}_k$, 且 $F \geq F_\alpha$, 则置 $k = k+1$, 并将 \mathbf{v}_{\max} 选入模型, 继续步骤(7); 其他情况下 ($\text{BIC}_{k+1} \geq \text{BIC}_k$ 或者 $F < F_\alpha$), 则不将 \mathbf{v}_{\max} 选入模型, 算法结束。 $\boldsymbol{\theta}_k$ 为最终模型的参数, $\boldsymbol{\theta}_k$ 及对应的项构成了最终模型。

(7) 对前面已选入模型的各项, 分别计算将其从 k 项中去掉后的信息准则值。计算方法如下: 将 Φ_k^{-1} 的第 i 行移至最后一行、第 i 列移至最后一列得到矩阵记为

$$\Phi_{k,i}^{-1} = \begin{bmatrix} \Phi & \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^T & a \end{bmatrix}$$

然后计算

$$\Phi_{k-1,i}^{-1} = \Phi - \boldsymbol{\alpha} \boldsymbol{\alpha}^T / a, \boldsymbol{\theta}_{k-1,i} = \Phi_{k-1,i}^{-1} \mathbf{R}_{k-1,i}^T \mathbf{y}, \mathbf{e}_{k-1,i} = \mathbf{y} - \mathbf{R}_{k-1,i} \boldsymbol{\theta}_{k-1,i}$$

$$f_{k-1,i} = \mathbf{e}_{k-1,i}^T \mathbf{e}_{k-1,i}, F_i = \frac{f_{k-1,i} - f_k}{f_k} (N - k)$$

$$S_{k-1,i}^2 = f_{k-1,i} / N, \text{BIC}_{k-1,i} = N \ln S_{k-1,i}^2 + (k-1) \ln N, i = 1, \dots, k$$

这里 $\mathbf{R}_{k-1,i}$ 是去掉 $N \times k$ 维矩阵 \mathbf{R}_k 中的第 i 列向量后得到的 $N \times (k-1)$ 维矩阵, 即

$$\Phi_{k-1,i}^{-1} = [\mathbf{R}_{k-1,i}^T, \mathbf{R}_{k-1,i}]^{-1}$$

(8) 检验第(7)步中计算出的准则值。将 F_i 和 $\text{BIC}_{k-1,i}, i = 1, \dots, k$ 中的最小值分别记为 F_{\min} 和 BIC_{\min} 。如果 $\text{BIC}_{\min} \leq \text{BIC}_k$ 或 $F_{\min} < F_\alpha$, 则将对应的项从模型中剔除, 并置 $k = k-1$;

$R_k=R_{k,\min},\Phi_k^{-1}=\Phi_{k,\min}^{-1},\theta_k=\theta_{k,\min}$,然后转向第(7)步。如果所有项都选入了模型,则算法结束。其他情况转第(3)步。

上述计算步骤中,第(1),(2)步用于置初值,第(3)~(6)步用于选变量,第(7),(8)步用于剔除变量。

值得说明的是,上述算法的描述(符号)似乎很复杂,但其描述都很明确具体,每个符号都有确切的含义,只要了解了算法的基本思路,即本节开始时对算法的概述,就很容易理解算法中具体步骤的意义。另外,该算法的每一步计算结果都为下一步直接应用,因此整个算法的计算量是很小的,而且很容易计算机实现,是一种较为理想的辨识方法。

【例 8.4】 仿真例子

考虑由如下模型描述的非线性系统

$$y(k)=0.2y(k-1)+0.3y^2(k-1)+0.2u(k-1)+0.6u^2(k-1)$$

以幅度为±0.9 的伪随机二位式序列作为输入,用零值、正态白噪声{e(k)}对输出 y 进行干扰。信噪比((y^Ty)^{1/2}/(e^Te)^{1/2})约为 100,产生 100 组数据对未知模型辨识。

假定该非线性模型为 2 阶 3 次幂,即 n₁=m₁=2,q=3。这样总项数为 35。按本节所给的递推算法逐步进行,模型中各项是按如下次序确定的:(0)计算常数项(即输出均值);(1)选入 u²(k-1);(2)选入 u(k-1);(3)选入 y(k-1);(4)选入 y²(k-1),并剔除常数项。下一个预选项是 u(k-1)·y²(k-1),由于其 F 值为 1.387762,小于临界值 F_a(=F_{0.05}≈3.9),故未被选入模型,到此算法结束。详细过程见表 8.4。

表 8.4 例 8.4 计算过程及结果

	Selected term	F-test	BIC	Investigated terms	F-test	BIC
0	\bar{y}		-327.48			
	Model: $y(k)=0.236$					
1	$u^2(k-1)$	138.81	-411.1	\bar{y} $u^2(k-1)$	24.35 138.81	-393.52 -327.48
	Model: $y(k)=0.087+0.539u^2(k-1)$					
2	$u(k-1)$	160.13	-503.98	\bar{y} $u^2(k-1)$ $u(k-1)$	53.20 361.55 160.13	-464.86 -353.25 -411.1
	Model: $y(k)=0.08+0.591u^2(k-1)+0.193u(k-1)$					
3	$y(k-1)$	4276.39	-881.25	\bar{y} $u^2(k-1)$ $u(k-1)$ $y(k-1)$	48.69 16536.19 7625.36 4276.39	-844.83 -370.38 -447.11 -503.98
	Model: $y(k)=-0.02+0.596u^2(k-1)+0.199u(k-1)+0.387y(k-1)$					

	Selected term	F-test	BIC	Investigated terms	F-test	BIC
4	$y^2(k-1)$	513.50	-1062.4	$* \bar{y}$	0.23	-1066.72
				$u^2(k-1)$	103939.73	-367.10
				$u(k-1)$	48665.79	-442.88
				$y(k-1)$	560.93	-873.74
				$y^2(k-1)$	513.50	-881.25
				$u^2(k-1)$	16622.27	-361.25
				$u(k-1)$	49208.57	-447.42
				$y(k-1)$	1153.58	-814.95
				$y^2(k-1)$	830.8	-844.83
	Model: $y(k)=0.603u^2(k-1)+0.1996u(k-1)+0.1985y(k-1)+0.298y^2(k+1)$					
5	$u(k-1)$ $y^2(k-1)$	1.39	-1063.8			
Final model: $y(k)=0.603u^2(k-1)+0.1996u(k-1)+0.1985y(k-1)+0.298y^2(k+1)$						

注:带*的项为被剔除项,这里 $F_a(=F_{0.05} \approx 3.9)$ 。

8.7.5 小结

本节给出的关于 NARMAX 模型的辨识方法,由于引入了“新息—贡献”准则,所以不仅避免了信息的重复,而且从根本上简化了计算;该方法采用了可添、可删的双向回归递推算法,可自动地选择非线性模型的结构,并同时得到参数估计,避免了大量重复计算,而且辨识结果与非线性模型项的排列顺序无关,从而保证了最终模型的惟一性和最优性。但这里给出的辨识方法,理论上只是针对确定性系统讨论的,因此仅适用于确定性或噪声很小的非线性系统辨识,当输出噪声较大时,参数估计将是有偏的。对于系统输出噪声较大时的系统辨识请参考有关文献(王秀峰,李波(1993)和 Korenberg M etc(1988))。

习 题

1. 牛顿—拉夫森方法和麦夸特方法有什么区别与联系?各有什么缺点?
2. 推导麦夸特方法关于多元函数如 $y = f[u_i, \theta]$, 其中 $y = (y_1, \dots, y_n)^T$, $f = (f_1, \dots, f_n)^T$ 的计算公式(8.30)和式(8.31)。
3. 证明式(8.38)。
4. 用如下模型产生数据

$$y(k) = 0.5e^{-0.1u_1(k)} + 100u_2(k) + 0.01e(k)$$

其中 u_1, u_2 为不同周期、幅值不同的伪随机二位式序列, $e(k)$ 为 $N(0, 1)$ 正态伪随机序列。然后分别用计算机实现单纯形搜索法和麦夸特方法对该模型进行辨识。

5. 将 8.7 节的辨识算法用于 GMDH 中的部分实现——二次多项式的辨识,给出具体步骤,并在计算机上实现。
6. 在计算机上实现例 8.4 的仿真例子。
7. 试说明 NARMAX 一般模型中式(8.49)为何会有信息重复项?
8. 试说明 8.7 节给出的递推算法中,为何选项上的项还可能被剔除?
9. 5.2 节中给出的递推算法为什么没有剔除的步骤?

第9章 房室模型的辨识

9.1 问题的提出

在这一章中将讨论一类特殊模型的辨识问题,这一类模型我们称为房室模型(Compartment model)。提出这一类模型主要是研究各种动力学模型时提出来的,特别是在药物动力学和各种生理系统动力学的研究中,房室模型起着十分重要的作用。此外,在生态系统和化学动力学等的研究中也用到房室模型。

房室(Compartment)是指包含某一物质的具有一定体积的空间,并假定任何时刻,该物质都均匀分布于该空间内,房室与房室之间及房室与外界环境都可能物质交换,并遵守物质守恒定律。

由若干个这样的房室组成的系统称为房室系统,描述这类系统的模型就是房室模型。

房室是一种理想化的情形,最初是为了研究药物在有机体内的分布而引入的。按照房室的概念,可以将所研究的系统分成若干个相互可以进行物质交换的房室,然后根据物质守恒定律,可得出系统中物质运动规律,即系统中物质随时间变化的分布规律。如果开始建立的模型是符合实际的,那么通过对物质运动规律的研究就可以得到许多信息,而这些信息仅通过实验手段是难以得到的,例如,各个房室之间物质的传输速度;某房室中的物质达到最高浓度的时间等。这样,通过房室模型的研究,就可以为这些系统的深入研究定量地提供重要的信息,可以更好地对系统进行分析、预测及其他方面的定量研究。

用房室模型研究生理动力学和药物动力学是系统辨识重要的应用之一。在药物动力学的研究中,提出的大量问题都可以用房室模型进行分析研究,如说明药物在体内的运动,即吸收分布和消除的规律,以及这些规律与药理作用的关系。在临床治疗上,根据房室模型的计算还可以对药物的疗效评价及毒性分析给出重要的依据。下面将给出实际的例子来说明它的应用。

房室模型的研究包括建模和辨识两个部分,建模就是要从实际研究的对象出发,根据研究的目的,经过分析、提炼、归并、简化,将所研究的系统划分为若干个房室(房室的个数正好是模型的阶),找出房室之间及房室与外界环境之间的相互关系,将这些关系弄清楚之后,数学模型很容易就写出来了。要做到这一点,主要是靠对研究对象本身的知识及对所研究问题的本质、深入地分析。例如,要研究在血液中两种物质的转化,就要将血液中的这两种物质看做是两个房室,虽然现实的空间是一个,但是需要抽象地提炼出两个房室。在化学动力学的研究中有类似的问题,研究参加反应的各种物质之间的相互转化关系,虽然反应器只有一个,但是却要被抽象成若干个房室。要最后确定模型还需要对未知部分进行辨识,对房室模型来说,主要是估计参数。一般地说,物质转化的模型用一组微分方程来表示的,所以辨识主要是通过实验数据估计微分方程的系数。对于房室模型,如生理、药理、生态等系统,要取得数据往往比较困难,并不是所有房室的状态都能得到,甚至取得数据的方式也要受到许多限制。例如,在生理系统中普遍采用的放射性同位素示踪测量方法,这类实验的输入方式(一般只能注射一定量的药

物,即脉冲输入)受到很大的限制,在输出数据的检测上也受到检测点和检测时间的限制。因此,实验数据在检测部位和数据量上都受到限制。这样一来,对房室模型来说,能否用所能得到的数据估计出全部想得到的参数,就成了一个突出的问题,即系统的可辨识性问题。在数据不足时,就只有降低要求,进一步简化模型。

9.2 房室模型的建模

建模是研究房室模型的首要步骤,而完成建模这一步主要是依靠对被研究系统的工作原理的了解和可能的实验条件,并不需要借助多少数学方法。这里通过几个例子来说明建模的基本思路。为了能掌握建模的基本方法,先将有关房室模型的几个基本概念进行一些更细致的讨论。

9.2.1 房室

房室是指含有某一物质的特定空间,该物质在任何时候都均匀分布于整个空间,并满足物质守恒定律。一个房室系统是指由若干个房室组成的系统,房室与房室之间有物质转移,房室与环境之间也有物质转移。描述一个房室系统的物质运动的数学方程就是房室模型。

这里所讨论的物质转移是广义的,可以是物质的流动,也可以是物质的转换,总之,只要符合物质守恒定律就行,所以这种转移含义是十分广泛的,它能概括相当广泛的现象。

房室模型是对现实系统的一种概括、归并和简化,它可以和一个现实系统同构,也可以不和任何一个具体的现实系统等同。例如,研究在血液中两种物质的相互转化,就把血液看做是两个房室;又如研究在同一个反应器中多种物质的相互转化,就要将反应器看做是多个房室。另一方面,也可以将含有不同物质的空间看做同一房室,如在研究药物动力学时,对于能迅速分布于血液及其他体液之间的药物,就可以将身体的这些部分看做是一个房室。如果血液及血液附近组织(周缘组织)的药物浓度有明显不同,就要将系统看做是由两个房室组成的,即血液作为一个房室,周缘组织作为另一个房室。所以,即使是研究生理系统,模型所定义的房室也不一定完全具有生理学或解剖学的特征。

总之,在对一个实际的系统建模时,究竟用几个房室,把哪些部分看做是一个房室是要根据应用研究的目的及对系统的具体分析,有时还要通过对实验数据拟合情况的比较来选择合适的房室模型的结构,特别是对实际系统的内部机制不完全清楚时更是如此。

9.2.2 物质转移速度

房室模型有一个重要的特征,就是在房室之间有物质的转移。房室之间物质的转移速度是房室模型研究的主要内容之一。

考虑一个 n 房室系统, $x_i(t)$ 表示在 t 时刻第 i 个房室中的物质含量,称为是系统的第 i 个状态,用 f_{ij} 表示从第 i 房室转移到第 j 房室的物质的转移速度,物质转移也可以是双向的(见图 9.1)。

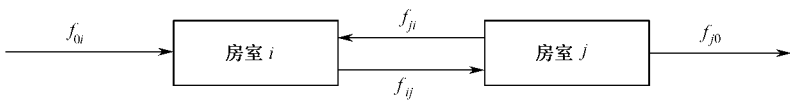


图 9.1 房室间的转移速度

房室也可以和环境间有物质转移,环境用 0 来标记,因此房室 i 和环境间的转移速度记为 f_{0i} 和 f_{i0} 。

有的物质转移速度是可以通过实验手段进行直接测量的,但是有许多转移速度是无法直接测量的,而这些转移速度却往往是分析系统或控制系统中十分重要的量,所以估计转移速度是研究房室模型中心问题之一,解决的办法是用辨识的方法通过可以量测到的实验数据去估计转移速度。

转移速度本身也可以具有不同的性质,例如,一种最简单的情形是 f_{ij} 只与 $x_i(t)$ 有关且是线性的,而与其他量无关。有时候情况就可能会复杂一些,如 f_{ij} 可以同时与 $x_i(t)$ 和 $x_j(t)$ 有关; f_{ij} 也可以是随机的; f_{ij} 与状态的关系也可以是复杂的非线性关系等,这些都是在房室模型的建模中需要解决的问题。

9.2.3 房室模型

当我们将研究的目的及系统本身的特性进行充分地研究之后,对系统进行合理的抽象、归并、简化之后得到一个合理的房室结构。换句话说,主要解决下面几个问题。

① 将系统分成若干个房室,对房室的实际意义、房室中的物质、房室内物质是否可以看做均匀分布、房室间的物质转移是否服从物质守恒定律等问题均有了肯定的结论,也就是系统可以用房室系统来描述。

② 确定房室的个数,这将决定模型的阶次。

③ 转移速度 f_{ij} 的其他性质,如对时间的依赖关系、对状态的依赖关系等。

解决了以上问题,就可以根据物质守恒定律写出房室系统的数学模型——房室模型。

设系统可分成 n 个房室,分别用 $1, 2, \dots, n$ 来标记,用 0 标记环境。对任何一个房室,根据物质守恒定律,即转移进该房室的物质减去转移出的物质等于房室内物质的变化,可得到

$$\frac{dx_i}{dt} = f_{0i} + \sum_{\substack{j=1 \\ j \neq i}}^n (f_{ji} - f_{ij}) - f_{i0} \quad i = 1, 2, \dots, n, 0 \leq t < \infty \quad (9.1)$$

初始条件为 $x_i(0) = x_{i0}, i = 1, 2, \dots, n$ 。

这样得到的 n 阶微分方程组就是房室模型。当然,为了具体描述系统的动态性能,还需要给出转移速度对状态的依赖关系。根据转移速度的不同形式,可以得到不同类型的房室模型。

在多数应用中,都考虑线性房室模型,也就是说,转移速度 f_{ij} 与状态 $x_i(t)$ 成正比,比例常数为 k_{ij} ,称为转移速度常数,即

$$f_{ij} = k_{ij}x_i, i = 0, 1, \dots, n, j = 1, \dots, n, i \neq j \quad (9.2)$$

于是房室模型式(9.1)变为

$$\frac{dx_i}{dt} = f_{0i} + \sum_{j=1}^n (k_{ji}x_j - k_{ij}x_i) - k_{i0}x_i \quad i = 1, 2, \dots, n, 0 \leq t < \infty \quad (9.3)$$

初始条件为 $x_i(0) = x_{i0}, i = 1, 2, \dots, n$ 。

在不同的应用领域中,转移速度还可以具有不同的特性,我们将在下面再做一些简短的讨论。

9.2.4 房室模型分类

本节按照转移速度的特性对房室模型做一个粗略的分类。

式(9.1)是一般性的模型,为了实际应用的需要,必须要具体找出转移速度与状态间的确定

切关系,式(9.3)所表示的为最简单、最常用的线性房室模型,根据转移速度与状态 x_1, \dots, x_n 之间的不同关系,除线性模型外,还可将模型分成如下几种不同的类型。

1. 非线性房室模型

当房室之间的转移速度与状态是非线性关系时,就得到非线性房室模型。非线性关系当然没有一般性表示,只能针对具体问题给出具体表达式,例如

$$f_{ij} = k_{ij} f_i(x_i), i = 0, 1, \dots, n, j = 1, 2, \dots, n, j \neq i$$

式中

$$f_i(x_i) > 0, x_i > 0, i = 1, 2, \dots, n$$

这样就可以得到房室模型

$$\frac{dx_i}{dt} = f_{0i} + \sum_{\substack{j=1 \\ j \neq i}}^n [k_{ji} f_j(x_j) - k_{ij} f_i(x_i)] - k_{i0} f_i(x_i) \quad (9.4)$$

$$0 \leq t < \infty, x_i(0) = x_{i0}, i = 1, 2, \dots, n$$

另一种非线性房室模型是转移速度常数依赖于所有的状态,即

$$k_{ij} = k_{ij}(x_1, \dots, x_n), i = 0, 1, \dots, n, j = 1, 2, \dots, n \quad j \neq i \quad (9.5)$$

于是可以得到房室模型为

$$\frac{dx_i}{dt} = f_{0i} + \sum_{\substack{j=1 \\ j \neq i}}^n [k_{ji}(x_1, \dots, x_n) x_j - k_{ij}(x_1, \dots, x_n) x_i] - k_{i0}(x_1, \dots, x_n) x_i \quad (9.6)$$

$$0 \leq t < \infty, x_i(0) = x_{i0}, i = 1, 2, \dots, n$$

当然还可以有其他形式的非线性房室模型。

2. 时变线性房室模型

转移速度常数是时变的,于是得到时变的房室模型为

$$\frac{dx_i}{dt} = f_{0i} + \sum_{\substack{j=1 \\ j \neq i}}^n [k_{ji}(t) x_j - k_{ij}(t) x_i] - k_{i0}(t) x_i \quad (9.7)$$

$$0 \leq t < \infty, x_i(0) = x_{i0}, i = 1, 2, \dots, n$$

3. 随机房室模型

一个房室系统包含的随机性(这里的随机性不是指数据中包含的随机噪声)大致有两种情形。

(1) 房室包含的物质微粒数的随机性

当房室中所感兴趣的物质是以微粒的形式出现,而微粒的数目又非常小时,房室之间的微粒运动的随机性就不能不考虑了。

(2) 将速度常数处理为随机变量

以上的分类只是一种粗略的分法,可供在建模时参考,实际的建模过程一定要从分析实际问题的角度出发,而不要受这些类型的约束。

9.2.5 房室模型建模示例

下面要讨论几个建模的具体例子,通过这些例子可以了解从分析实际对象出发直到最后

建立符合需要的模型的全过程。要分析实际对象,就不可避免地涉及到与该实际系统有关的一些知识,如生理、化学……,要详细地讨论这些知识,当然超出了本书的范围,所以在遇到这些情况时,读者只要接受有关的结论而不必去深究其细节,注意力集中在如何利用这些知识去建立模型。

【例 9.1】胆红素代谢的随机房室模型

用静脉快速注射同位素标记的胆红素作为示踪物来研究胆红素在人体内的代谢,由生理学知识可知,该药在注射后 4h 内主要在血液和肝脏内,这样可以把胆红素代谢过程描述成两房室系统(见图 9.2)。

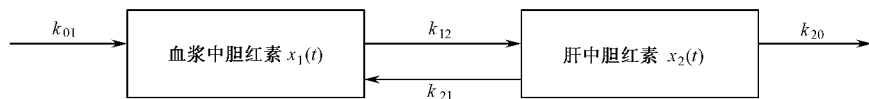


图 9.2 胆红素代谢的房室模型

如果考虑到正常人、溶血病人或先天性溶血性黄疸等各类疾病患者中个体之间的差异,参数本身随时间会有变化,再加上其他的影响因素,我们可以考虑把速度看做是随机变量,这就是随机房室模型。

【例 9.2】肾清除率的房室模型

这里要研究的是肾功能,表征肾功能的一个重要指标是肾清除率。所谓肾清除率,就是指肾如果在单位时间内能将 C 毫升血浆中含有的某种物质全部清除去,则清除率就是 C 毫升/分。肾是排泄器官,大约由 200 万个肾单位组成,血液进入肾脏以后,血液中的物质可以通过细胞膜的渗透作用而进入肾小球,而这些物质又通过肾小管的吸收,一部分回到血液中,一部分进入尿道,通过膀胱排泄出去。

为了研究肾清除率,可以寻找从血液中进入肾后不再被肾小管吸收回血液的某种物质,该物质绝大部分被排泄出去。例如,邻碘马尿酸就是这样的物质。将它用同位素(如 I^{131})标记以后,静脉注射进入血液,然后在肾脏记录放射性的量,就可以观察到邻碘马尿酸通过肾的清除过程。由于邻碘马尿酸能通过肾小球的过滤而又不被肾小管吸收,所以计数得到的量就是其清除率,这个指标是研究肾功能的一个重要指标。建立肾功能模型的重要目的之一就是要估计肾的清除率。

为了达到这个研究目的,要对这一过程进行描述,也就是要建立肾功能的模型。这一特定的模型并不意味着那是人体的真实的实体表示,而只是体现系统的本质特征,并且尽可能用简单的方式。

研究肾清除率的最简单的房室模型是两房室,一个表示血液中的示踪物,另一个是从血液中出去的示踪物,用这样的模型来预测血液房室的示踪物,得到的结论是示踪物应按指数减少,这与实际不符,这说明所考虑的模型不正确(过于简单)。

再扩大到 3 个房室,即一个中心房室(血液)、一个由可与中心房室有物质转移的器官(血管等)构成的周缘房室,以及一个与中心房室进行单向转移(排泄)的末端房室(肾),其结构如图 9.3 所示。

由中心房室向末端房室的清除速度就是所要估计的参数。在研究这一系统时,从血管快速注射(脉冲输入)用同位素 I^{131} 标记的药物,然后在末端房室(肾)记录示踪物的变化,这就是通常同位素检查中的肾图。由于实验手段的限制,所得到的测量值除了主要是末端房室放射性外,还

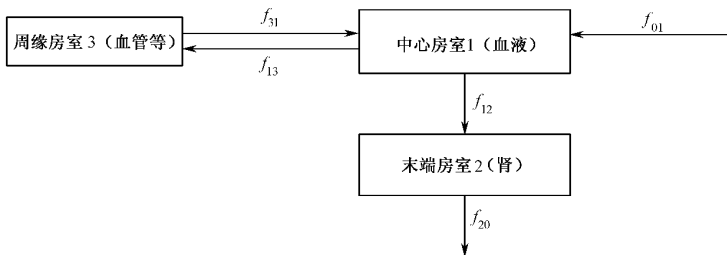


图 9.3 肾功能系统的房室系统

有一部分是中心房室和边缘房室的放射性。一般来说,它是由末端房室放射性的 50% 和其他两个房室放射性的各 5% 混合在一起的结果。这些情况在数据处理时应具体考虑。

假定 3 个房室中示踪物的量分别是 Q_1, Q_2, Q_3 (mg); 房室间的清除速度分别为 f_{12}, f_{13}, f_{31} (ml/min); 离开第 i 房室进入第 j 房室的示踪物的速度 (mg/min) 等于 f_{ij} (ml/min) 乘以示踪物浓度 (mg/ml) 即 $f_{ij} \frac{Q_i}{V_i}$; V_i 是第 i 房室的体积; 房室模型可以写为 (注意系统在初始时刻输入示踪物 Q_0 后输入量就是零)

$$\frac{dQ_1}{dt} = f_{31} \frac{Q_3}{V_3} - f_{13} \frac{Q_1}{V_1} - f_{12} \frac{Q_1}{V_1}$$

注射的示踪物总量为 Q_0 , 并令 $\frac{Q_1}{Q_0} = x_1, \frac{Q_2}{Q_0} = x_2, \frac{Q_3}{Q_0} = x_3$, 以及

$$\frac{f_{12}}{V_1} = k_{12}, \frac{f_{13}}{V_1} = k_{13}, \frac{f_{31}}{V_3} = k_{31} \quad (1/\text{min})$$

k_{ij} 为清除速度, 即每分钟从 i 房室清除到 j 房室的无量纲体积。模型就变为

$$\begin{aligned} \frac{dx_1}{dt} &= k_{31}x_3 - k_{13}x_1 - k_{12}x_1 \\ \frac{dx_2}{dt} &= k_{12}x_1 - k_{20}x_2 \\ \frac{dx_3}{dt} &= k_{13}x_1 - k_{31}x_3 \end{aligned} \quad (9.8)$$

初始值

$$x_1(0) = 1, x_2(0) = x_3(0) = 0 \quad (9.9)$$

观测值

$$y(t) = x_2(t)$$

由此即可通过观测得到的数据去估计参数 $k_{13}, k_{31}, k_{12}, k_{20}$ 。

【例 9.3】水生生物的生态系统的房室模型

生态系统与人类生活紧密相关。当前对生态系统的研究已成为引人注目的课题, 生态系统的数学模型已经有了许多研究成果。利用数学模型可以对生态系统中能量转换和物质循环进行定量的研究。例如, 研究某一种或几种生物种群数量的消长对生态系统平衡的影响、研究生态系统在人工干预下产生的演变等。这些都可以为自然资源的合理开发利用和优化管理提供可靠的科学依据。

生态系统是由生物生长、消亡、能量转换, 以及与周围环境的关系构成的系统。下面以某水域的水生生物的生态模型为例来说明建模的过程。房室模型是描述生态系统的一种重要手

段。现在讨论的系统概括地说有植物、食草类动物和肉食类动物。植物靠太阳能及其他营养而生长,食草类动物以植物为营养而又是肉食动物的食物,这样 3 种生物及周围环境构成了一个生态平衡系统,如何来描述这个系统呢?

水生生物系统可以分成 3 个房室:房室 P 是水生植物,是营养的生产者;房室 H 是食草类动物;房室 C 是食肉类动物, H 和 C 是营养消费者。用 P_p 表示供给 P 的太阳能及其他营养素的单位时间变化率, G_p 表示供给 H 的牧食率; G_h 表示供给 C 的捕食率, m_p , m_h , m_c 分别表示 P , H 和 C 的沉积、呼吸、排泄和死亡率。

这样的系统结构如图 9.4 所示。

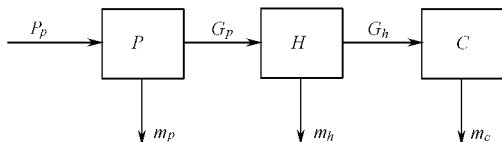


图 9.4 某水域水生生物生态的房室系统

由此可以写出该系统的房室模型为

$$\begin{aligned}\frac{dP}{dt} &= P[P_p(1 - \sigma_p) - m_p - G_p H / (1 + \sigma_p P)] \\ \frac{dH}{dt} &= H[G_p P / (1 + \sigma_p P) - m_h - G_h C / (1 + \sigma_h H)] \quad 0 \leq t \leq T \\ \frac{dC}{dt} &= C[G_h H / (1 + \sigma_h H) - m_c] \\ P(0) &= P_0 \quad H(0) = H_0 \quad C(0) = C_0\end{aligned}$$

式中, $P_p(1 - \sigma_p)$ 表示 P 的自然增长率; $1/(1 + \sigma_p P)$ 表示牧食系数; $G_p P / (1 + \sigma_p P)$ 是 H 的自然增长率; $1/(1 + \sigma_h H)$ 表示被捕食物系数; $G_h H / (1 + \sigma_h H)$ 是 C 的自然增长率。

σ_p, σ_h 是待定常数, m_p, m_h 和 m_c 也可以看做是常数,可以由实验来确定, P_p, G_p 和 G_h 是时变参数(随季节性变化),一般是不容易测定的,研究模型就是要定量地估计这些参数。

从这个例子中可以看出,房室模型对描述生态系统的变化规律是很合适的,同时也看出参数 P_p, G_p 和 G_h 都有明确的生态学意义且是时变的,所以参数的估计也是比较复杂的。

【例 9.4】胆红素代谢过程建模

为了更深入研究胆红素代谢过程,对图 9.3 所示的房室模型可以进一步划分出更多的房室。在研究中,从一只胳膊上注射 ^{14}C 标记的游离胆红素,在另一只胳膊上取血样进行检验,从生理学的知识得知,胆红素代谢路线如图 9.5 所示。

根据这个结构,应该用 6 个房室去描述胆红素代谢动力学房室模型,如图 9.6 所示。

这样的结构是否合适呢?还需要用后验知识去检验,即由输入、输出数据去检验。上面所得到的模型从生理学的角度似乎是合适的,但是由于实验手段的限制,只能得到系统的脉冲响应,用这样的数据是不能将 6 个房室的参数全部估计出来的,也就是说,这个结构太复杂。为了能得到一个适用的模型,需要将模型约简。从对实验数据(见图 9.7)的分析可知,对大部分数据可以用一个指数函数去拟合(即一房室模型),但是将该曲线外推到零时间的截距,即血液体积的估计值比预计的要小,所以加上第二个指数项,也就是用二房室模型,这样就得到了约简后的结构。对二房室模型有两种假设的结构,即略去图 9.6 中的房室 1,5 和 6,并将房室 2 和 3 合并得到如图 9.8(a)所示的结构,或将之与 4 合并得到如图 9-8(b)所示的结构。

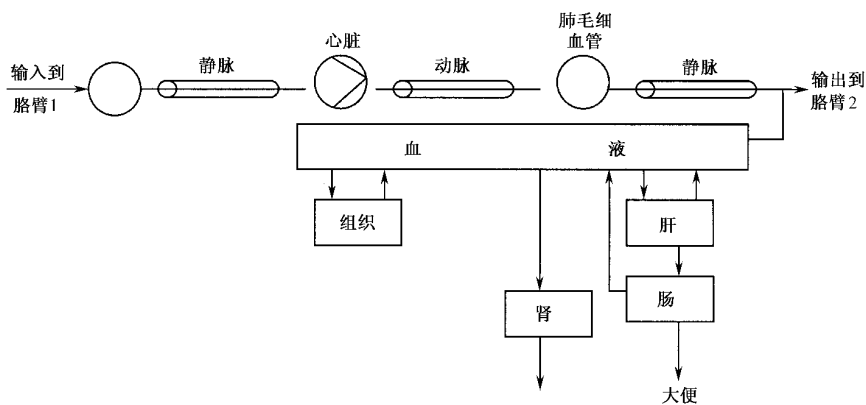


图 9.5 胆红素代谢路线模型

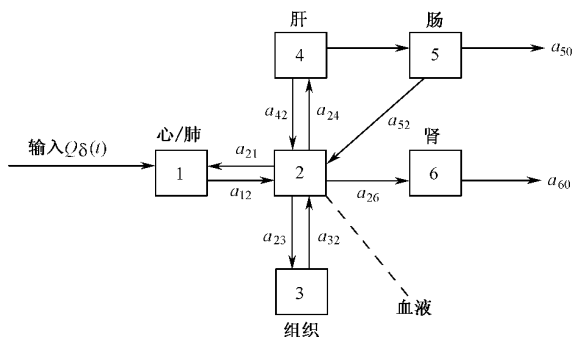


图 9.6 胆红素代谢房室模型

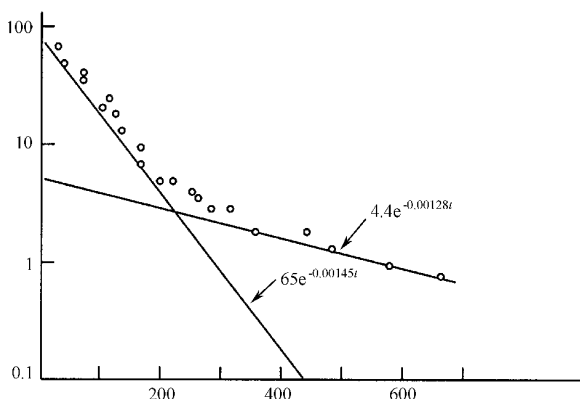


图 9.7 血浆中胆红素清除率曲线

这是通过系统的方法和生理学方法相结合而对模型的结构进行约简而得到的房室模型，也就是用先验知识和后验知识相结合确定模型的结构。如图 9.8(a)所示就是例 9.1 中的模型，正如例 9.1 中所指出的，如果考虑到正常人、溶血病人或先天性溶血黄疸等各类疾病患者中个体之间的差异，参数本身随时间可以有变化，再加上其他的影响因素，可以把速度常数看做是随机变量，这样就构成了胆红素代谢过程的随机房室模型。

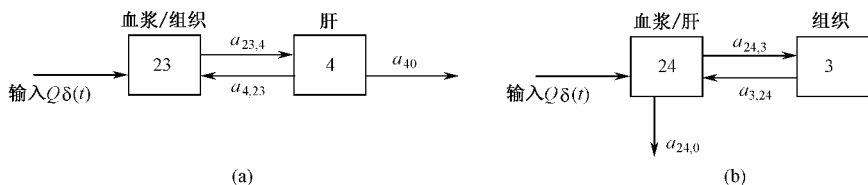


图 9.8 约简的胆红素代谢房室模型

9.3 参数估计

9.3.1 问题

建立一个房室模型,就意味着确定它的结构,其中包括模型的阶及转移速度的形式,这时还可能包含有若干未知参数需要确定,这些参数一般是转移速度常数(对线性房室模型)或其他确定转移速度的参数(对非线性房室模型)。这些参数必须通过输入、输出数据来进行估计,只有当这些参数都被估计出来,房室模型才最终确定。

在房室模型的参数估计中,有一个需要特别注意的问题,即可辨识性问题。可辨识性问题是指在房室系统中,往往受实验条件的限制而不能得到足够的输入、输出数据,这样就会造成部分参数不能惟一确定,此时要讨论估计的惟一性和可能取到的输入、输出数据之间的关系,这对房室模型的参数估计来说是十分重要的。另一方面,这种讨论也有助于如何安排和改进实验,使得能得到惟一的参数估计值。关于可辨识性的问题,将在下一节进行讨论,这一节主要讨论参数估计方法。

房室模型一般都表现为连续时间模型,而实验数据绝大多数都是在离散时间上取得的,所以这里的参数估计问题实质上是一个用离散时间数据去估计连续时间模型的参数估计问题。

9.3.2 参数估计算法

房室模型参数估计的一般思路是:首先根据初始条件对某些给定的参数值解模型,得到状态的时间函数,然后再计算出模型的输出,并将在采样时间点上计算出的模型的输出与实验数据相比较,按照某种优化准则来调整参数,直到最后确定参数的最优值。由于从模型解出的状态的时间函数往往是参数的非线性函数,所以参数优化计算也将是非线性的,最常用的就是非线性最小二乘准则。

设已经建立的房室模型是式(9.3),即

$$\begin{aligned} \frac{dx_i}{dt} &= f_{0i} + \sum_{j=1}^n (k_{ji}x_j - k_{ij}x_i) - k_{i0}x_i, 0 \leq t \leq \infty \\ x_i(0) &= x_{i0}, i = 1, 2, \dots, n \end{aligned} \quad (9.10)$$

根据初始条件可解出 $x_i(t)$, $i = 1, 2, \dots, n$, 然后由 $x_i(t)$ 算出模型输出 $y^*(t)$ 。一般来说, $y^*(t)$ 是参数 k_{ij} 的非线性函数,于是可以按非线性最小二乘准则,使

$$Q = \sum_{j=1}^N [y^*(t_j) - y(t_j)]^2 \quad (9.11)$$

达到最小来确定参数 k_{ij} , 式中 t_j 是采样时刻, $y(t_j)$ 是采样值。为此,可采取非线性最小二乘法的各种算法,如搜索算法或迭代算法(见第8章)。

要注意通过非线性最小二乘法能够直接确定的参数不一定是 k_{ij} , 也可能是 k_{ij} 的一些组合。至于通过所确定的参数(可能有部分 k_{ij} 和其他的参数)能不能惟一地全部确定 k_{ij} , 还要视问题的具体条件而定。

综上所述, 房室模型的参数估计主要包括模型求解和非线性最小二乘两部分。为了求解模型, 必须对系统的输入认真加以研究, 如果环境(外界)对系统的输入 $f_{0i} (i=1, 2, \dots, n)$ 是一般形式的, 要得到解就比较麻烦, 在不得已的情况下, 只好通过数值解去处理。在房室模型的应用中, 最常用的输入形式有两种: 脉冲输入(相当于快速注射)和阶跃输入(相当于点滴)。对于这两种特定的输入, 都可以用求模型的解析解的办法, 使问题的计算大大简化。

对于输入是脉冲的情形, 输入的影响只体现在初始状态上, 即 x_{i0} 就是在初始时刻外界输入系统(如注射)的物质浓度, 而模型式(9.1)中的 f_{i0} 就全为零, 于是模型变为一个常系数齐次微分方程组。因此, 只需求出系数矩阵的特征值, 就可以结合初始条件得出模型的解析解。对于输入是阶跃函数的情形, 就要稍微复杂一些, 但阶跃函数是一种特殊的输入, 可以采用拉氏变换方法来得到方程的解析解。

下面通过具体的实例来详细说明这一过程。

1. 对于脉冲输入的情形

首先考虑图 9.3 所示的 3 个房室模型, 这种情形得到房室模型式(9.8), 即

$$\begin{aligned}\frac{dx_1}{dt} &= -(k_{13} + k_{12})x_1 + k_{31}x_3 \\ \frac{dx_2}{dt} &= k_{12}x_1 - k_{20}x_2 \\ \frac{dx_3}{dt} &= k_{13}x_1 - k_{31}x_3\end{aligned}$$

初始条件为

$$\begin{aligned}x_1(0) &= 1 \\ x_2(0) &= x_3(0) = 0\end{aligned}$$

这样一来输入 Q_0 就成了模型的初值。

如果所采集的实验数据是通过肾图所描记的肾中示踪物的消失过程, 设在 $t_1 < t_2 < \dots < t_N$ 时刻记录了实验数据 $y(t) = x_2(t)$

$$y(t_1), y(t_2), \dots, y(t_N)$$

于是可以用非线性最小二乘估计法来得到速度常数 k_{ij} 的估计量。

由于在非线性最小二乘准则式(9.11)中, $y^*(t)$ 是通过求解房室模型得到的, 但是不一定能得到显含参数 k_{ij} 的解, 因此, 应用搜索法解非线性最小二乘估计可能更方便。

从式(9.8)知方程组的系数矩阵为

$$\mathbf{A} = \begin{pmatrix} -(k_{13} + k_{12}) & 0 & k_{31} \\ k_{12} & -k_{20} & 0 \\ k_{13} & 0 & -k_{31} \end{pmatrix}$$

其特征方程是

$$|\lambda \mathbf{I} - \mathbf{A}| = \begin{vmatrix} \lambda + (k_{13} + k_{12}) & 0 & -k_{31} \\ -k_{12} & \lambda + k_{20} & 0 \\ -k_{13} & 0 & \lambda + k_{31} \end{vmatrix}$$

$$= (\lambda + k_{20})(\lambda^2 + (k_{13} + k_{12} + k_{31})\lambda + k_{13}k_{12}) = 0 \quad (9.12)$$

求得特征根是 $-k_{20}, \lambda_1, \lambda_2$ 。

根据特征方程的特殊形式 ($-k_{ij} > 0$) 和 λ_1, λ_2 为相异实根, 这里可以找到 3 个特解, 分别为

$$\mathbf{x}^{(1)} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (9.13)$$

$$\mathbf{x}^{(2)} = \begin{pmatrix} \frac{k_{31} + \lambda_1}{k_{13}} \\ \frac{k_{12}(k_{31} + \lambda_1)}{\lambda_1 k_{13}} \\ 1 \end{pmatrix} e^{\lambda_1 t} = \begin{pmatrix} \beta_1^{(2)} \\ \beta_2^{(2)} \\ 1 \end{pmatrix} e^{\lambda_1 t} \quad (9.14)$$

$$\mathbf{x}^{(3)} = \begin{pmatrix} \frac{k_{31} + \lambda_2}{k_{13}} \\ \frac{k_{12}(k_{31} + \lambda_2)}{\lambda_2 k_{13}} \\ 1 \end{pmatrix} e^{\lambda_2 t} = \begin{pmatrix} \beta_1^{(3)} \\ \beta_2^{(3)} \\ 1 \end{pmatrix} e^{\lambda_2 t} \quad (9.15)$$

于是得到通解

$$\begin{aligned} x_1(t) &= c_1 \beta_1^{(2)} e^{\lambda_1 t} + c_2 \beta_1^{(3)} e^{\lambda_2 t} \\ x_2(t) &= c_3 + c_1 \beta_2^{(2)} e^{\lambda_1 t} + c_2 \beta_2^{(3)} e^{\lambda_2 t} \\ x_3(t) &= c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} \end{aligned} \quad (9.16)$$

再利用初始条件式 (9.9) 来确定常数 c_1, c_2, c_3 得

$$\begin{aligned} c_1 - c_3 &= \frac{k_{13} x_1(0)}{\lambda_1 - \lambda_2} \\ c_2 &= \frac{k_{12} k_{31} x_1(0)}{\lambda_1 \lambda_2} \end{aligned} \quad (9.17)$$

2. 当输入是阶跃函数的情形

这里考虑一个药物动力学的例子。

【例 9.5】研究某种新药用于人体的药物动力学

所谓药物动力学, 就是要研究药物在血液中和在体内各器官组织中的浓度分布过程和排泄过程。

对完全由肾脏通过尿排泄的药物, 一般可以考虑用只有中心房室和排泄的两房室模型, 并假定转移速度是线性的, 如图 9.9 所示。

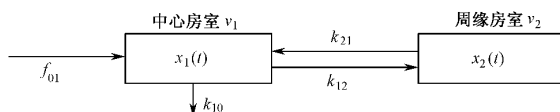


图 9.9 药物两房室分布模型

为了观察其疗效, 希望求出此药的药物动力学参数 (即 k_{ij}), 以及血液 (中心房室) 及周缘房室中的药物浓度随时间变化的曲线。

实验设计如下: 从初始时刻开始, 匀速静脉点滴 1h 后停药。滴注总量为 2g, 滴药停止后,

测量受试者各时刻血液中的药物浓度,取得实验数据,这里模型为

$$\begin{aligned}\frac{dx_1}{dt} &= -(k_{10} + k_{12})x_1 + k_{21}x_2 + f_{01} \\ \frac{dx_2}{dt} &= k_{12}x_1 - k_{21}x_2\end{aligned}\quad (9.18)$$

其中

$$\begin{aligned}f_{01} &= 2 \times 10^6 \times [I(t) - I(t-1)] (\mu\text{g}) \\ I(t) &= \begin{cases} 0 & t \geq 0 \\ 1 & t < 0 \end{cases}\end{aligned}\quad (9.19)$$

因为输入已不再是脉冲式的,所以 f_{01} 不能为 0,这样就要求用 Laplace 变换的方法,首先对式(9.18)两端取拉氏变换得

$$\begin{aligned}sX_1(s) &= -(k_{10} + k_{12})X_1(s) + k_{21}X_2(s) + \frac{Q_0}{s\tau}(1 - e^{-s\tau}) \\ sX_2(s) &= k_{12}X_1(s) - k_{21}X_2(s)\end{aligned}$$

式中, $Q_0 = 2 \times 10^6 \mu\text{g}$, τ 是输入持续时间。

于是可解得 $X_2(s) = \frac{k_{12}}{s+k_{21}}X_1(s)$,代入第一式可得

$$X_1(s) = \frac{Q_0(1 - e^{-s\tau})(s + k_{21})}{\tau s[s^2 + (k_{21} + k_{12} + k_{10})s + k_{21}k_{10}]}\quad (9.20)$$

其次,将式(9.20)分解成部分分式,因为 k_{21}, k_{12}, k_{10} 均为正实数,所以分母中方括号内的二项式必可分解为 $(s+\alpha)(s+\beta)$ 的形式,其中, $\alpha > \beta > 0$, 均为实数,可得

$$\begin{aligned}\alpha\beta &= k_{21}k_{10} \\ \alpha + \beta &= k_{21} + k_{12} + k_{10}\end{aligned}$$

可用待定系数法求出各部分式的系数,即有

$$\frac{s + k_{21}}{s(s^2 + (k_{12} + k_{21} + k_{10})s + k_{21}k_{10})} = \frac{A}{s} + \frac{B}{s + \alpha} + \frac{C}{s + \beta}$$

其中

$$\begin{aligned}A + B + C &= 0 \\ A\alpha + A\beta + B\beta + C\alpha &= 1 \\ A\alpha\beta &= k_{21}\end{aligned}$$

可解得

$$\begin{aligned}A &= \frac{k_{21}}{\alpha\beta} \\ B &= \frac{k_{21} - \alpha}{\alpha(\alpha - \beta)} \\ C &= \frac{\beta - k_{21}}{\beta(\alpha - \beta)}\end{aligned}$$

故得

$$\begin{aligned}X_1(s) &= \frac{Q_0(1 - e^{-s\tau})(s + k_{21})}{\tau s(s + \alpha)(s + \beta)} \\ &= \frac{Q_0}{\tau}(1 - e^{-s\tau})\left(\frac{k_{21}}{\alpha\beta s} + \frac{k_{21} - \alpha}{\alpha(\alpha - \beta)(s + \alpha)} - \frac{\beta - k_{21}}{\beta(\alpha - \beta)(s + \beta)}\right)\end{aligned}\quad (9.21)$$

最后对式(9.21)逐项取反拉氏变换得

$$\begin{aligned}
 x_1(t) = & \frac{Q_0 k_{21}}{\tau \alpha \beta} I(t) + \frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta)} e^{-\alpha t} - \frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta)} e^{-\beta t} - \\
 & \frac{Q_0 k_{21}}{\tau \alpha \beta} I(t - \tau) - \frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta)} e^{-\alpha(t-\tau)} I(t - \tau) + \\
 & \frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta)} e^{-\beta(t-\tau)} I(t - \tau)
 \end{aligned} \quad (9.22)$$

注意到

$$\begin{aligned}
 L^{-1} \left(\frac{Q_0 k_{21}}{\tau \alpha \beta s} \right) &= \frac{Q_0 k_{21}}{\tau \alpha \beta} I(t) \\
 L^{-1} \left(\frac{Q_0 k_{21}}{\tau \alpha \beta s} e^{-\tau s} \right) &= \frac{Q_0 k_{21}}{\tau \alpha \beta} I(t - \tau) \\
 L^{-1} \left(\frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta) (s + \alpha)} \right) &= \frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta)} e^{-\alpha t} \\
 L^{-1} \left(\frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta) (s + \alpha)} e^{-\tau s} \right) &= \frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta)} e^{-\alpha(t-\tau)} I(t - \tau) \\
 L^{-1} \left(\frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta) (s + \beta)} \right) &= \frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta)} e^{-\beta t} \\
 L^{-1} \left(\frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta) (s + \beta)} e^{-\tau s} \right) &= \frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta)} e^{-\beta(t-\tau)} I(t - \tau)
 \end{aligned}$$

做时间变换 $t' = t - \tau$, 即以滴注停止时为计算时间的起点, 为清楚起见, 下式中仍以 t 表示 t' , 即

$$\begin{aligned}
 x_1(t) = & \frac{Q_0 (k_{21} - \alpha)}{\tau \alpha (\alpha - \beta)} (e^{-\alpha(t+\tau)} - e^{-\alpha t}) + \frac{Q_0 (\beta - k_{21})}{\tau \beta (\alpha - \beta)} (e^{-\beta(t+\tau)} - e^{-\beta t}) \\
 = & R e^{-\alpha t} + S e^{-\beta t} + T
 \end{aligned} \quad (9.23)$$

其中

$$\begin{aligned}
 R &= \frac{Q_0 (k_{21} - \alpha) (e^{-\alpha \tau} - 1)}{\tau \alpha (\alpha - \beta)} \\
 S &= \frac{Q_0 (\beta - k_{21}) (e^{-\beta \tau} - 1)}{\tau \beta (\alpha - \beta)} \\
 T &= \frac{k_{21} f_{01}}{\tau \alpha \beta}
 \end{aligned}$$

用相同的方法可以得到在停药后, 周缘组织液中药量随时间的变化曲线。

以上通过两个实例讨论了如何对两种特定的输入求解房室模型。在房室模型求解的问题解决之后, 可以用非线性最小二乘法估计转移速度。

9.3.3 参数估计中遇到的几个问题

下面对在参数估计中会遇到的几个问题做一点简要的讨论。

1. 关于数据

房室模型辨识中所用的数据是多种多样的, 主要取决于在所研究的系统中采用的测量手段。例如, 在生理系统中, 用体外测量手段 (如肾功能系统) 就可以得到一条连续描记的曲线

(见图 9.10);对需要测量血液内物质浓度的系统,如药物动力学系统,就只能在一些离散的时间点上取到数据;又如在环境、生态等系统中,由于测量困难或周期过长,往往只能得到很少的数据,对这些数据的使用要注意数据处理,不要将过大的测量误差带到参数估计中来。

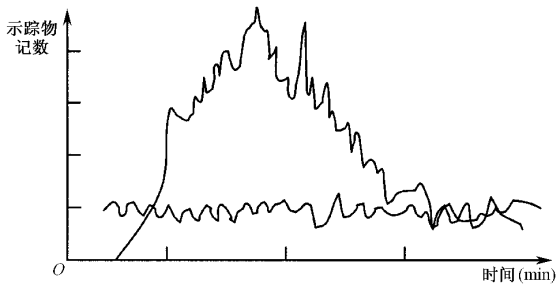


图 9.10 肾图

2. 求参数的近似初值

在估计房室模型的参数时,可以采用非线性最小二乘法,如果用迭代法求解,就需要有较好的参数估计的初值,即使用搜索法,有一个较好的初始估计值也有利于计算,所以,如何得到较好的初始参数估计值是房室模型辨识中值得注意的一个问题,下面讨论一种常用的简便方法。

求解房室模型得到的浓度曲线是由指数函数叠加而成的,这样就可以在半对数纸上对参数进行近似地估计,仍然以血药浓度估计为例,设实例数据如下:

采样次数 i :	1	2	3	4
采样时间:	滴药停止时	停药 1h	停药 2h	停药 3h
$x_1(t_i)/V_1$:	237.94	86.22	40.44	26.41

而 $x_1(t) = Re^{-\alpha t} + Se^{-\beta t} + T$, 即 $x_1(t)$ 是由两条指数曲线相叠加而成的 ($\alpha > \beta$), 然后可用目测求出 R, S, α, β , 具体做法见后面 9.5 节“应用实例”。

9.4 可辨识性问题

9.4.1 问题

可辨识性问题是系统辨识的一个重要问题,也是一个普遍存在的问题,房室模型由于试验条件和测量手段的限制,这个问题就显得格外突出,这里就房室模型来讨论可辨识性问题,其思路是有一般性意义的,对其他类型的模型也是有效的。

在房室模型的辨识中(在其他模型的辨识中也存在着类似的问题),由于采集数据受到客观条件的限制,因此可能由于所得到的信息不足而不能惟一确定所要估计的参数。例如,在生理系统中并不是任何一个房室中示踪物消失的速度都能检测,同样在药物动力学或化学反应动力学,以及环境、生态等的研究中都有类似的问题存在,这个问题在多房室模型的辨识中尤为突出。

另一方面,能够人为从外界施加给系统的刺激也是受限制的,多数情况下,只能有脉冲式和阶跃式两种。由于系统不能得到充分的激励,就可能造成有些参数不能够得到惟一的估计值。从数据不能得到惟一的参数的估计值,就称参数是不可辨识的。如果出现了这样的情况,

就需要重新考虑实验的安排或者改变模型的结构。

在辨识中还可能出现另一种“可辨识性”问题,即由于数据中包含随机噪声,使得估计值偏离参数的“真值”。这两种可辨识性问题当然是不一样的,前者是先验可辨识性问题,即能否得到惟一解,而后者是要解决如何从噪声数据中得到参数真值,这个问题称为后验可辨识性问题,这个问题已经在第 4 章中讨论过了,这里只讨论先验可辨识性问题。

下面先讨论几种简单情形,然后再讨论几个复杂的多房室例子。

9.4.2 一房室模型

一房室系统如图 9.11 所示,模型为

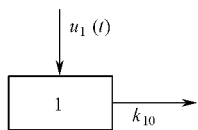


图 9.11 一房室结构

$$\frac{dx_1}{dt} = -k_{10}x_1(t) + u_1(t) \quad (9.24)$$

式中, $u_1(t)$ 是输入, $x_1(t)$ 是房室 1 的状态,例如,向人体内注射的某种药物, k_{10} 是房室对环境的转移速度。如果考虑的是一个脉冲输入,例如向血管中的快速注射,则

$$u_1(t) = u_1 \delta(t) \quad (9.25)$$

模型的解是

$$x_1(t) = u_1 e^{-k_{10}t} \quad (9.26)$$

假定可以测量房室 1 的响应,即测量方程是

$$y_1(t) = h_1 x_1(t) \quad (9.27)$$

h_1 是一增益,可以是已知的也可以是未知的。

如果测量的量是浓度,则 $h_1 = \frac{1}{V_1}$, V_1 是房室 1 的体积,对快速注射的响应将是

$$y_1(t) = k_1 e^{-\lambda_1 t} \quad (9.28)$$

通过实验数据即可估计得到 k_1 和 λ_1 , 于是

$$k_{10} = \lambda_1$$

$$V_1 = \frac{u_1}{k_1}$$

所以在这种情况下,一房室模型的参数都是可辨识的。

9.4.3 二房室模型

二房室系统结构如图 9.12 所示。

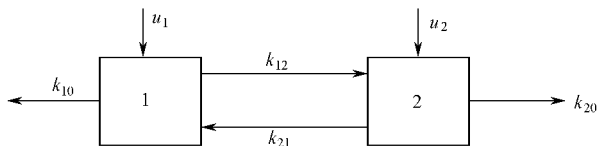


图 9.12 二房室系统结构

模型为

$$\begin{aligned}\frac{dx_1}{dt} &= -(k_{10} + k_{12})x_1 + k_{21}x_2 + u_1 \\ \frac{dx_2}{dt} &= k_{12}x_1 - (k_{20} + k_{21})x_2 + u_2\end{aligned}\quad (9.29)$$

测量方程为

$$\begin{aligned}y_1(t) &= h_1 x_1(t) \\ y_2(t) &= h_2 x_2(t)\end{aligned}\quad (9.30)$$

式中, $x_1(t)$, $x_2(t)$ 分别是房室 1 和房室 2 的状态, $y_1(t)$, $y_2(t)$ 分别是它们的测量值。最常用的试验是仅对房室 1 进行脉冲扰动, 同时也仅在房室 1 测量响应 (即 $u_2 = h_2 = 0$) 这种情形, 系统是不可辨识的。

如果 $h_2 = 0$, 但房室 2 也可以被扰动, 有 3 种情形:

- ① 如果 $u_1(t)$ 和 $u_2(t)$ 都是脉冲, 则系统是不可辨识的;
- ② 如果 $u_1(t)$ 是非脉冲量, $u_2(t)$ 是一个脉冲, 只要 h_1 已知, 则系统是可辨识的;
- ③ 如果 $u_1(t)$ 是一个脉冲, $u_2(t)$ 是非脉冲量, 即使 h_1 是未知的, 模型也是可辨识的。

此外, 当 $u_2 = h_2 = 0$ 时, 如果 k_{10} 或 k_{20} 是零, 则模型是可辨识的。

9.4.4 三房室模型

这里以两个常用的三房室模型为例来讨论可辨识性, 首先考虑一个三房室模型, 其结构如图 9.13 所示。

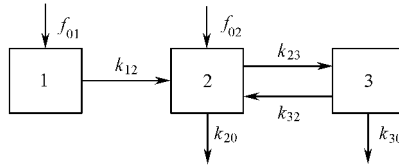


图 9.13 三房室结构

假定测量的输出是

$$\begin{aligned}y_1(t) &= h_1 x_2(t) \\ y_2(t) &= h_2 \int_0^t k_{20} x_2(\tau) d\tau\end{aligned}\quad (9.31)$$

$y_2(t)$ 是到 t 时刻为止, 房室 2 向外界环境转移的物质总量, h_1, h_2 是测量增益, 可以是已知的也可以是未知的, 房室模型是

$$\begin{aligned}\frac{dx_1}{dt} &= f_{01} - k_{12}x_1 \\ \frac{dx_2}{dt} &= f_{02} + k_{12}x_1 - (k_{23} + k_{20})x_2 + k_{32}x_3 \\ \frac{dx_3}{dt} &= k_{23}x_2 - (k_{32} + k_{30})x_3\end{aligned}\quad (9.32)$$

假设 f_{01} 是一大剂量脉冲量 $Q_1 \delta(t)$, f_{02} 是 0, 并设在 $t=0$ 时所有房室都是空的, 则关于模型的可辨识性有下列结果:

- ① 如果 Q_1, h_1 和 h_2 是已知的, 则 5 个速度常数都惟一可确定;
- ② 如果 Q_1 未知, 而 h_1 和 h_2 已知, 则速度常数 k_{20} 是惟一可辨识的, 但是对速度常数 k_{12} ,

k_{23}, k_{30} 和 k_{32} 有 3 组可能的值；

③ 如果 Q_1 和 h_1 是已知的, 但 h_2 未知, 则速度常数 k_{12} 是惟一可辨识的, 而 k_{20}, k_{23}, k_{30} 和 k_{32} 有无穷多组解；

④ 如果 Q_1 和 h_2 是已知的, 但 h_1 是未知的, 则所有 5 个速度常数都有无穷多组解。

这里分析的只是一般的情形, 三房室模型还有许多可能的情况, 每一种情况的可辨识性问题都需要具体分析。以例 9.2 讨论过的肾功能系统为例, 这一系统输出方程是

$$h(t) = h_1 x_2(t)$$

所以要讨论它的可辨识性问题, 就需要分析 $x_2(t)$ 与要估计参数之间的关系, 从所取的 3 个特解的第 2 个分量可以看出, 解只与 k_{12}/k_{13} 有关, 所以从估计中也只能得到这个比值的估计值, 要估计出每一个参数就需要再估计出其中的一个参数。

9.4.5 多房室模型的可辨识性问题

多房室系统一般都比较复杂, 参数是否可以辨识与模型的结构有关, 很难得到一般性的结论, 对具体的系统需要具体分析。下面举一个 6 房室系统的例子来讨论可辨识性问题的思路。

【例 9.6】甲状腺激素代谢的 6 房室模型

这里讨论的是一个经典的例子, 它研究的是甲状腺激素代谢系统, 系统结构如图 9.14 所示。建模的详细过程涉及到甲状腺生理的有关知识, 不在这里详细讨论。

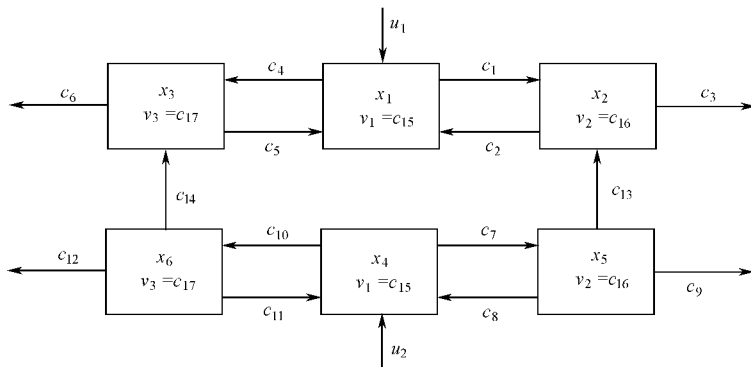


图 9.14 6 房室系统结构

图 9.14 所示给出的结构对应的房室模型是式(9.33)。

$$\begin{aligned} \frac{dx_1}{dt} &= [u_1 + c_5 x_3 + c_2 x_2 - c_1 x_1 - c_4 r_1] / c_{15} \\ \frac{dx_2}{dt} &= [c_1 x_1 + c_{13} x_5 - (c_2 + c_3) x_2] / c_{16} \\ \frac{dx_3}{dt} &= [c_4 r_1 + c_{14} x_6 - (c_5 + c_6) x_3] / c_{17} \\ \frac{dx_4}{dt} &= [u_2 + c_8 x_5 + c_{11} x_6 - c_7 x_4 - c_{10} r_2] / c_{15} \\ \frac{dx_5}{dt} &= [c_7 x_4 - (c_8 + c_9 + c_{13}) x_5] / c_{16} \\ \frac{dx_6}{dt} &= [c_{10} r_2 - (c_{11} + c_{12} + c_{14}) x_6] / c_{17} \end{aligned} \quad (9.33)$$

式中, $r_1 = r_1(x_1, x_4, k)$, $r_2 = r_2(x_1, x_4, k)$ 是已知的非线性函数, 能得到的观测值是

$$\begin{aligned} y_1 &= x_1 + v_1 \\ y_2 &= x_4 + v_2 \end{aligned} \quad (9.34)$$

初始条件 $x_1(t_0)$, $x_4(t_0)$ 及 $r_1(t_0)$, $r_2(t_0)$ 可以直接测量, 其余的初始条件将从稳态方程解出

$$\begin{aligned} x_2(t_0) &= \frac{c_1(c_8 + c_9 + c_{13})x_1(t_0) + c_7c_{13}x_4(t_0)}{(c_2 + c_3)(c_8 + c_9 + c_{13})} \\ x_3(t_0) &= \frac{c_4(c_{11} + c_{12} + c_{14})x_1(t_0) + c_{10}c_{14}r_2(t_0)}{(c_5 + c_6)(c_{11} + c_{12} + c_{14})} \\ x_5(t_0) &= \frac{c_7x_4(t_0)}{c_8 + c_9 + c_{17}} \\ x_6(t_0) &= \frac{c_{10}r_2(t_0)}{c_{11} + c_{12} + c_{14}} \end{aligned} \quad (9.35)$$

在小扰动情况下, 可用近似 $\Delta r_1 = F_1 \Delta x_1$, $\Delta r_2 = F_2 \Delta x_4$, 所以将模型线性化得到

$$\frac{d\Delta \mathbf{x}}{dt} = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & 0 & 0 \\ a_4 & a_5 & 0 & 0 & a_6 & 0 \\ a_7 & 0 & a_8 & 0 & 0 & a_9 \\ 0 & 0 & 0 & a_{10} & a_{11} & a_{12} \\ 0 & 0 & 0 & a_{13} & a_{14} & 0 \\ 0 & 0 & 0 & a_{15} & 0 & a_{16} \end{bmatrix} \Delta \mathbf{x} + \begin{bmatrix} 1/c_{15} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1/c_{15} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \Delta \mathbf{u} \quad (9.36)$$

式中, 参数 a_1, \dots, a_{16} 与原始模型参数的关系是

$$\begin{aligned} a_1 &= -(c_1 + c_4 F_1) c_{15} & a_2 &= c_2 / c_{15} \\ a_3 &= c_5 / c_{15} & a_4 &= c_1 / c_{16} \\ a_5 &= -(c_2 + c_3) / c_{16} & a_6 &= c_{13} / c_{16} \\ a_7 &= c_4 F_1 / c_{17} & a_8 &= -(c_5 + c_6) / c_{17} \\ a_9 &= c_{14} / c_{17} & a_{10} &= -(c_7 + c_{10} F_2) / c_{15} \\ a_{11} &= c_8 / c_{15} & a_{12} &= c_{11} / c_{15} \\ a_{13} &= c_7 / c_{16} & a_{14} &= -(c_8 + c_9 + c_{13}) / c_{16} \\ a_{15} &= c_{10} F_2 / c_{17} & a_{16} &= -(c_{11} + c_{12} + c_{14}) / c_{17} \end{aligned}$$

输入方程 $\Delta y_1 = \Delta x_1$, $\Delta y_2 = \Delta x_4$, 则

$$\frac{d\Delta \mathbf{x}}{dt} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ 0 & \mathbf{A}_4 \end{bmatrix} \Delta \mathbf{x} + \begin{bmatrix} \Delta u_1 \\ 0 \\ \Delta u_2 \\ 0 \\ 0 \end{bmatrix} \quad (9.37)$$

根据线性化模型式(9.36), 系统可分为 3 个相互分离的子系统 A_{11} , A_{12} , A_{22} , 它们的传递函数分别为

$$H_{11}(s) = \frac{k_{11}(s - a_5)(s - a_8)}{s^3 + \beta_3 s^2 + \beta_2 s + \beta_1} \quad (9.38)$$

$$H_{22}(s) = \frac{k_{11}(s - a_{14})(s - a_{16})}{s^3 + \beta_6 s^2 + \beta_5 s + \beta_4} \quad (9.39)$$

$$H_{12}(s) = \frac{k_{11}k_{12}(s^2 + \beta_7 s + k')}{(s^3 + \beta_6 s^2 + \beta_5 s + \beta_4)(s^3 + \beta_3 s^2 + \beta_2 s + \beta_1)} \quad (9.40)$$

式中

$$k_{11} = 1/c_{15}$$

$$k_{12} = a_2 a_6 a_{13} + a_3 a_9 a_{15}$$

$$\beta_1 = a_2 a_4 a_8 + a_5 (a_3 a_7 - a_1 a_8)$$

$$\beta_2 = a_1 (a_5 + a_8) + a_5 a_8 - a_2 a_4 - a_3 a_7$$

$$\beta_3 = -(a_1 + a_5 + a_8)$$

$$\beta_4 = a_{11} a_{13} a_{16} + a_{14} (a_{12} a_{15} - a_{10} a_{16})$$

$$\beta_5 = a_{10} (a_{14} + a_{16}) - a_{11} a_{13} - a_{12} a_{15} + a_{14} a_{16}$$

$$\beta_6 = -(a_{10} + a_{14} + a_{16})$$

$$\beta_7 = -[a_2 a_6 a_{13} (a_8 + a_{16}) + a_3 a_9 a_{15} (a_5 + a_{14})]/k_{12}$$

$$k' = \frac{a_5 a_{14} (\beta_7 + a_8 + a_{16}) - a_8 a_{16} (a_5 + a_{14} + \beta_7)}{(a_3 + a_{16}) - (a_5 + a_{14})}$$

从模型可以看出, $\Delta x_4, \Delta x_5, \Delta x_6$ 完全可以独立进行辨识。如果 Δu_1 是扰动输入, $\Delta u_2 = 0$, $\Delta x_4(t_0) = 0$, Δy_1 是输出, 则子系统 A_{11} 可以独立进行辨识。

现在可设计两种实验。

实验 A: 同时输入两个充分小的扰动 $\Delta u_{1A}, \Delta u_{2A}$, 在时刻 t_{0A} 开始, 同时测量输出 $\Delta y_{1A}, \Delta y_{2A}$;

实验 B: 输入充分小的扰动 Δu_{2B} 并且测量输出 $\Delta y_{1B}(t)$ 。

则子系统 A_{11} 可以从实验 A 来辨识, 子系统 A_{12} 和子系统 A_{22} 可以从实验 B 来辨识, 这样一来, 原来 17 个参数仅有 13 个组合 $k_{11}, k_{12}, a_5, a_8, a_{14}, a_{16}, \beta_1, \dots, \beta_7$ 可以辨识, 所以要辨识所有的参数, 还需要其他的信息。

从这个例子可以看出, 并不是有了模型就可以估计出参数, 而是要对模型进行分析, 有针对性地设计实验, 另外, 例子也表明了研究可辨识性的重要性。

9.5 应用实例

在这一节我们讨论几个利用房室模型的应用实例。

【例 9.7】药物动力学研究

本研究的目的是比较两种药物在人体内的药物动力学特征, 这两种药是国产新药呋苄青霉素和进口羧苄青霉素。其目的是要比较两种药的体内血药浓度的动态分布和排泄规律, 即通过比较它们的药物动力学参数来分析比较两种药的药效和毒性。

要正确全面评价一种药物, 需要将其体内外抑菌浓度的高低和在人体内的药物动力学特征结合起来考虑。

1. 建模

为了要分析某种药物在人体内各器官的分布及其排泄的规律, 就要对药物代谢的系统给以恰当的描述, 并在此基础上进行分析。

药物代谢是一个生理过程。这里要比较两种药物在静脉注射后在血液及其他器官中的分布,以及它们之间的转移规律,在药物注入血液后,要使药物在血液及其他器官中达到均匀分布,需要有一个过程。虽然这个时间难以准确掌握,然而人体内的血液循环的流速很快,在研究药效时,完全可以忽略,因此可以近似地满足房室模型的条件。

进一步就要决定在这个系统中究竟应该包含多少个房室,这一点将取决于药物的性质。羧苄青霉素是完全由肾脏通过尿排泄的药物,对这样的药物一般可考虑用两房室模型,即只有中心房室和排泄(周缘房室)(见图 9. 9)。

至于呋苄青霉素,由实验得知此药有肾外排泄途径,只有 40% 的药物由肾脏通过尿排泄,一般要用带有胃肠回路的多房室模型(见图 9. 15)。

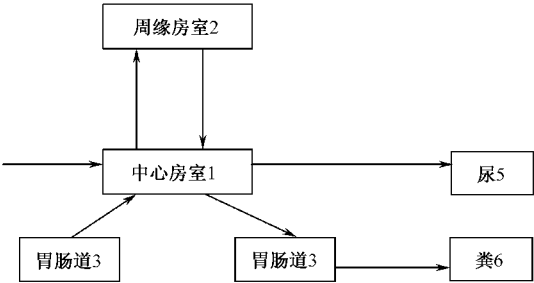


图 9. 15 药物代谢的多房室模型

两种模型计算后,发现两房室模型只在实验曲线的最后部分(远低于临床有效药物浓度)与实验值符合较差,可是我们研究的目的在于比较药物的疗效,而两房模型要简单得多,所以还是采用了两房室模型。

此外,在研究药物在体内分布的药物动力学时,一般认为药物在各房室间的转换是通过房室间的生物膜相互转换渗透的,渗透系数一般是常数,此常数就是速度常数 k_{ij} ,它反映了药物的特性及不同个体渗透的不同性质。

2. 实验

用呋苄青霉素和羧苄青霉素以同样的剂量、同样的输入方式在不同时间对同一组正常人进行试验,然后采血样分析血药浓度曲线,实验用匀速静脉点滴 1h,滴注药量为 2g,滴药停止后测量受试者的各时刻血药浓度,得实验数据如下:

时间 t_i	滴药停止时	停药 1h	停药 2h	停药 3h
$c_i(t_i) = \frac{x_1(t_i)}{v_i}$	237.94	86.22	40.44	26.41

3. 房室模型

输入

$$f_{01} = 2 \times 10^6 [I(t) - I(t - 1)] \mu\text{g}$$

$$I(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

按物质守恒原理得到数学模型为

$$\begin{cases} \frac{dx_1}{dt} = -(k_{10} + k_{12})x_1 + k_{21}x_2 + f_{01} \\ \frac{dx_2}{dt} = k_{12}x_1 - k_{21}x_2 \end{cases}$$

初始条件 $x_1(0)=0$, $x_2(0)=0$; 输出是房室 1 的药物浓度; $c_1(t)$; 目的是要估计 k_{ij} 。

4. 可辨识性

上一节已经提到一般两房室模型, 只扰动房室 1 并且只检测房室 1, 则模型是不可辨识的。但是对这里所讨论的特殊情形(周缘房室对外没有输入也没有输出), 这种情形参数是可以辨识出来的, 具体分析见下面的参数估计过程。

5. 参数估计

模型的解可以用 9.4 节中所讨论过的拉氏变换的解法来得到, 即

$$x_1(t) = \frac{x_1(0)(k_{21} - \lambda_1)(e^{-\lambda_1 t} - 1)}{\tau\lambda_1(\lambda_1 - \lambda_2)}e^{-\lambda_1 t} - \frac{x_1(0)(\lambda_2 - k_{21})(e^{-\lambda_2 t} - 1)}{\tau\lambda_2(\lambda_1 - \lambda_2)}e^{-\lambda_2 t}$$

$$\triangleq (\alpha e^{-\lambda_1 t} + \beta e^{-\lambda_2 t})\nu_1$$

用实验数据 $(t_i, x_1(t_i))$: $(0, 237.94)$; $(1, 86.22)$; $(2, 40.44)$; $(3, 26.41)$ 根据最小二乘准则

$$Q = \sum_{i=1}^4 [\alpha e^{-\lambda_1 t_i} + \beta e^{-\lambda_2 t_i} - c_1(t_i)]^2$$

然后可用非线性最小二乘法求取估计值, 例如可用迭代算法或搜索算法。

6. 参数估计初值

为了得到最后的估计值, 各种算法往往都需要有一组参数初值, $x_1(t)$ 是指数函数叠加而成的, 可以在半对数纸上由实验曲线粗略估计出参数的初值(见图 9.16)。

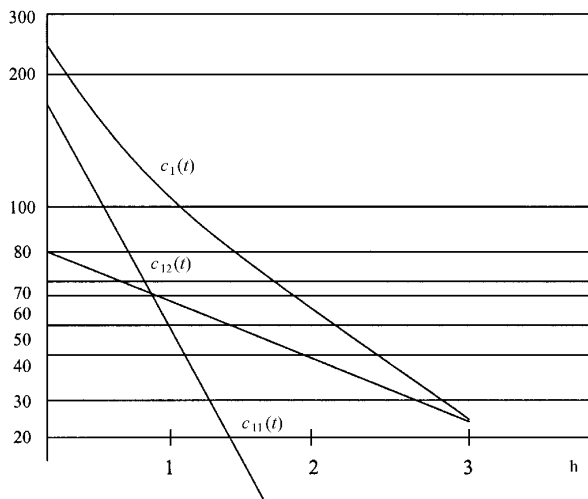


图 9.16 用目测法求参数初值

图中, $c_1(t)$ 是两条曲线 $c_{11}(t) = \alpha e^{-\lambda_1 t}$ 及 $c_{12}(t) = \beta e^{-\lambda_2 t}$ ($\lambda_1 > \lambda_2$) 的和, 当 t 充分大时, $\alpha e^{-\lambda_1 t}$ 趋近于 0, 因此 c_1 接近于 $\beta e^{-\lambda_2 t}$, 可以在 $c_1(t)$ 的尾部接近于直线的地方按此斜率画出一条直线 $c_{12}(t)$, $c_{12}(t)$ 即 $\beta e^{-\lambda_2 t}$, 在此直线上任意读出两个点的坐标, 即可定出 β 和 λ_2 。如果 $t=0$ 时, 读出 $c_{12}(t) = 80$; $t=1$ 时 $c_{12}(t) = 54$, 则有 $\beta = 80, 54 = 80e^{-\lambda_2}$, 得 $\lambda_2 = 0.393$, 则可定出 $c_{12}(t) = 80e^{-0.393t}$, 然后有 $c_{11}(t) = c_1(t) - c_{12}(t) = \alpha e^{-\lambda_1 t}$, 逐点求出 $c_{11}(t)$, 在半对数图上描出, 应为一直线, 如图 9.16 所示。由图上读出 $t=0$ 时, $c_{11}(t) = 157.94$; $t=1$ 时 $c_{11}(t) = 32.22$, 即 $32.22 = 157.94e^{-\lambda_1}$, 得 $\lambda_1 = 1.596$, 由此估计出

$$\begin{aligned}\lambda_1^0 &= 1.596 & \lambda_2^0 &= 0.393 \\ \alpha^0 &= 157.94 & \beta^0 &= 80\end{aligned}$$

7. 结果

由参数估计初值, 从 $\alpha^0, \beta^0, \lambda_1^0, \lambda_2^0$ 出发, 应用非线性最小二乘估计可得到最终估计值 $\alpha, \beta, \lambda_1, \lambda_2$, 但是我们要估计的是转移速度常数 k_{ij} , 这些速度常数与所得到的估计值之间的关系是

$$\begin{aligned}\lambda_1 \lambda_2 &= k_{21} k_{10} \\ \lambda_1 + \lambda_2 &= k_{12} + k_{21} + k_{10}\end{aligned}$$

同时由式(9.23)有

$$\begin{aligned}\alpha &= \frac{(k_{12} - \lambda_1)(e^{-\lambda_1 t} - 1)}{v_1 \tau \lambda_1 (\lambda_1 - \lambda_2)} x_1(0) \\ \beta &= \frac{(\lambda_2 - k_{12})(e^{-\lambda_2 t} - 1)}{v_1 \tau \lambda_2 (\lambda_1 - \lambda_2)} x_1(0)\end{aligned}$$

式中, $x_1(0) = 10^6 \mu\text{g}$, $\tau = 1$, 再用得到的估计值 $\alpha, \beta, \lambda_1, \lambda_2$, 即可解得

$$\begin{aligned}k_{21} &= 0.6928 \\ v_1 &= 4876.96 \\ k_{10} &= 0.9457 \\ k_{12} &= 0.3416\end{aligned}$$

现在还有最后一个参数 v_2 无法直接由实验曲线求出, 为了求出 v_2 , 需对模型做进一步的分析。根据图 9.9 所示的模型, 如果长时间不停止地以匀速 k_{01} 从静脉注入药物, 时间充分长以后, 体内药浓度及药量处于稳态平衡, 在此平衡状态下, 必有 $c_1(t) = c_2(t) = \text{常数}$ 。 $\frac{x_0}{\tau} = k_{01}$, 在 $\tau \rightarrow \infty$ 时有 ($t' \rightarrow 0$)

$$\begin{aligned}x_1(\infty) &= -\frac{k_{01}(k_{12} - \lambda_1)}{\lambda_1(\lambda_1 - \lambda_2)} - \frac{k_{01}(\lambda_2 - k_{12})}{\lambda_2(\lambda_1 - \lambda_2)} = \frac{k_{01}k_{12}}{\lambda_1\lambda_2} \\ c_1(\infty) &= k_{01}k_{12}/(\lambda_1\lambda_2v_1) = c_2(\infty)\end{aligned}$$

同时

$$\begin{aligned}x_2(\infty) &= \frac{k_{01}k_{21}}{\lambda_1(\lambda_1 - \lambda_2)} + \frac{k_{01}k_{21}}{\lambda_2(\lambda_1 - \lambda_2)} = \frac{k_{01}k_{21}}{\lambda_1\lambda_2} \\ c_2(\infty) &= \frac{k_{01}k_{21}}{\lambda_1\lambda_2v_2} = \frac{k_{01}k_{12}}{\lambda_1\lambda_2v_1} \\ v_2 &= \frac{k_{21}}{k_{12}}v_1 = 2404\end{aligned}$$

这样就求出了全部参数, 由此即可按模型算出药物在周缘组织中的浓度随时间变化的曲线。

通过模型的分析可知,味苳青霉素在血液中的平均浓度是羧苳青霉素的 3.01 倍,这和临床实验的结果符合得很好。

【例 9.8】 甲状腺激素代谢的房室模型的参数估计

在讨论房室模型的可辨识性问题时,在例 9.5 中已经看到了甲状腺激素代谢的 6 房室模型,这个模型研究的是甲状腺激素 T_3, T_4 的代谢规律, T_3, T_4 是人体内具有重要生理作用的激素。甲状腺激素的代谢过程是很复杂的。由于它们都是微量物质,再加上其他实验手段的限制,除了血液中的 T_3, T_4 外,它们是很难测量的。要研究 T_3, T_4 的生成、传输及转换的规律,利用房室模型是很有效的。

T_3, T_4 代谢的房室模型如图 9.14 所示,房室 1,4 分别表示血液中的 T_3 和 T_4 ,由甲状腺体分泌的 T_3, T_4 进入血液后,通过血液以不同的速度分布和清除到各个血管外的空间中去,其中有以肝为代表的速度快的空间,模型中用房室 2 和 5 表示;有以血管外肌肉组织为代表的速度慢的空间,模型中用房室 3 和 6 表示。房室 1,2,3 是关于 T_3 的,房室 4,5,6 是关于 T_4 的,同时一部分 T_4 可以转换为 T_3 ,这样的模型是半生理同构的,模型参数的生理意义都是十分清楚的。

从式(9.35)、式(9.36)可知,整个系统实际上可以分成 3 个子系统,这 3 个子系统的传递函数如上一节的式(9.37)~(9.39)所示,为了进行辨识,设计了如下两个实验。

实验 A:用 I^{131} 标记的 T_3 和 I^{125} 标记的 T_4 同时在一时刻快速注入血管,并同时记录血液中的 T_3 和 T_4 (通过静脉取血样化验),这样的输入就近似为脉冲输入,所得到的数据就是脉冲响应。如图 9.17 所示。图中所画的是 3 个正常人的数据平均值。

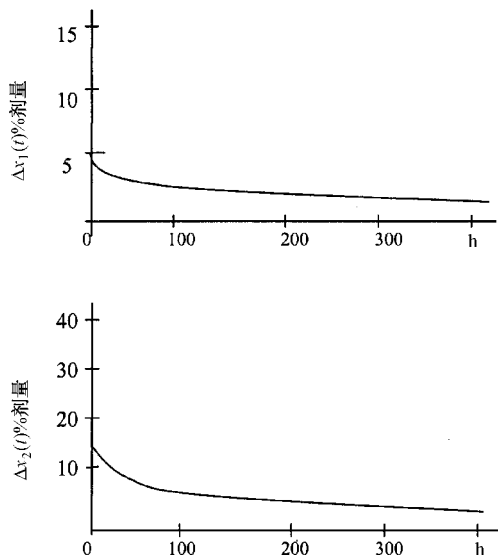


图 9.17 实验 A 的脉冲响应曲线

实验 B:单输入 I^{125} 标记的 T_4 ,记录血液中的 T_3 ,用这样的实验来估计 a_6 和 a_9 。

先用实验 A 来估计子系统 A_1 和 A_4 ,估计算法可用第 8 章讨论的非线性最小二乘算法。这样就可以得到不同参数的 13 个组合: $\nu_p (=c_{15}), a_1, a_5, a_8, a_{10}, a_{14}, a_{16}, a_3 a_7, a_2 a_4, a_{12} a_{15}, a_{11} a_{13}, a_2 a_6 a_{13}, m (a_3 a_9 a_{15}) + d$ (可估计出 m, d)。

然后利用动态模型的参数可以估计若干稳态参数(因为人体一般是处于稳态的,只有在外

界注射入 T_3, T_4 后才破坏了平衡,出现了动态过程)。

SR_3 : 甲状腺分泌的 T_3 ($\mu\text{mol} / \text{h}$)。

SR_4 : 甲状腺分泌的 T_4 ($\mu\text{mol} / \text{h}$)。

TR_{3PT} : 从血液到组织的 T_3 的稳态流速 (nmol / h)。

TR_{4PT} : 从血液到组织的 T_4 的稳态流速 (nmol / h)。

TR_{3TP} : 从组织到血液的 T_3 的稳态流速 (nmol / h)。

TR_{4TP} : 从组织到血液的 T_4 的稳态流速 (nmol / h)。

CR_{4-3} : 血管外 T_4 到 T_3 的转换速度 (nmol / h)。

DR_3 : 从组织中来的 T_3 , 不可逆的 (nmol / h)。

DR_4 : 从组织中来的 T_4 , 不可逆的 (nmol / h)。

以上这几个参数都是不能直接测量到的,而又具有十分重要的生理意义,对甲状腺代谢系统的生理研究有重要意义。

从这几个例子可以看出,系统方法的突出优点是用系统的观点去研究问题,从而可以得到单纯从外部的观察所难以得到的信息,从而使得问题的研究达到新的深度。

习 题

1. 考虑如图 9.18 的二房室系统

如果可测量的输出量是 $x_2(t)$, 转移系数 a_{ij} 是状态的线性函数。试辨识该模型并讨论可辨识性:

- (1) 写出系统的房室模型;
- (2) 求解 $x_1(t), x_2(t)$;
- (3) 讨论 a_{12}, a_{21}, a_{10} 的可辨识性。

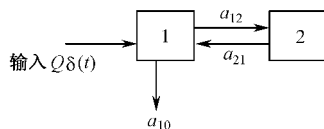


图 9.18 习题 1 图

2. 考虑如图 9.19 所示的二房室系统

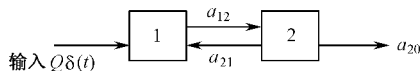


图 9.19 习题 2 图

如果可测量的输出量是 $x_1(t)$, 试讨论如上题(1),(2)及 a_{12}, a_{21}, a_{20} 的可辨识性。

3. 考虑如图 9.12 所示的二房室系统,证明在下列几种情况下系统是不可辨识的:

- (1) 当 $u_2 = h_2 = 0, u_1(t) = c\delta(t)$;
- (2) $u_1(t), u_2(t)$ 都是脉冲量。

4. 考虑如图 9.13 所示的三房室系统,假定测量的输出量是

$$y_1(t) = h_1 x_2(t)$$

$$y_2(t) = h_2 x_3(t)$$

h_1, h_2 已知,试讨论速度常数 $k_{12}, k_{23}, k_{20}, k_{30}, k_{32}$ 的可辨识性。

5. 证明 9.4.4 节三房室系统的可辨识性结果 ①~④。

6. 考虑如图 9.13 所示的三房室系统。如果 f_{02} 是一大剂量脉冲量, f_{01} 是 0, 假定 $t=0$ 时所有房室都是空的,试讨论模型的可辨识性。

第 10 章 时间序列的建模与辨识

10.1 引言

借助系统有关的先验知识进行建模(主要是给出模型结构),以及通过输入、输出数据对系统进行辨识(得到系统的输入/输出关系)是前几章所讨论的主要问题。但是,还有一类很重要的系统不能完全用那些方法来处理。例如,在生物系统中的各种生物电(如心电、脑电、肌电……),在社会经济系统中的市场价格,对某种产品的需求量,航空公司的订票数,天文学中的太阳黑子数,机械学中的机床震动,社会上的某种疾病的发病率等。这些量有两个共同的特点:首先,将这些量看做一个系统的输出,则系统的输入往往是不清楚的,也就是说,因果关系中的“因”不清楚,能被我们观测到的只是“果”(当然,“因”在客观上是存在的,只是由于我们对系统的知识缺乏或是系统本身过于复杂使得原因不易清楚);其次,这些量随时间变化的观测值之间是相关的,也就是说,任何时刻的观测值都受过去观测值的影响。这种相关性是我们能对这些量进行研究的基础,因为没有相关性的序列(白噪声)是不能预测的。

随时间顺序排列的一个观测序列称为时间序列(当然,不一定是真正的时间,可以是其他量,如长度、次数等),或者说是一个随机序列(这里要讨论的都是等间隔观测值)。

由于时间序列只有输出观测,所以很难直接用前几章的方法来进行建模和辨识。应该指出的是,这里所说的“很难直接用前几章的方法来进行建模和辨识”是指一般情况,但在处理具体问题,也可以把当前的输出作为过去输出的函数,即将过去的输出作为输入、当前的输出作为输出,构成一个形式上的输入/输出系统,从而可用前几章的方法来进行建模和辨识,这种处理思路在近年来的研究中是经常采用的。但是,时间序列毕竟有其自身的特点,应从其本身的特点寻找新的方法。为此,首先要明确研究时间序列的目的,一般来说,研究时间序列的主要目的是要通过对它自身相关性分析来预测其未来的行为。解决这一问题的主要工具是其观测值之间的(自)相关函数。传统的办法是在时域估计其相关函数或在频域估计其功率谱,有时候甚至只用简单的波形(定性)分析方法来分析系统,如最初在心电、脑电波的分析中主要还是依靠这种手段。当然这样做也有一定的效果,但是对时间序列的信息分析总是不完善的。另外,由于有限的长度数据带来的估计误差,以及由相关函数(或功率谱密度)去进行预测的复杂性而使得时间序列的分析变得困难。

本节主要讲述另外一种途径——模型法。它的主要思路是将时间序列看成是某一系统的输出,而系统的输入是白噪声,如图 10.1 所示,只要能找出这个系统模型,预测问题就迎刃而解了。

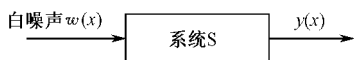


图 10.1 时间序列的系统模型

系统 S 的建模就是随机序列中的谱分解问题,理论上已经证明在一定条件下这一问题

可以解的(Åström,1971)。

Box and Jenking (1976), Wu(吴贤铭)and Pandit(1980)等对系统 S 的建模问题做了很详细的讨论,近年来这方面的文章和应用都很多,本章仅对系统 S 建模问题中的几个基本问题做一些讨论,为进一步研究建模奠定一定的基础。

值得注意的是图 10.1 所示的系统并不是产生时间序列 $y(t)$ 的真实系统,而是为了对 $y(\cdot)$ 进行分析和预测而构造的系统(当然不排除有一些时间序列真是由白噪声刺激某个系统而产生的响应),并用 $y(\cdot)$ 的观测数据去拟合而得到的,所以这样得到的模型除了进行预测外,如果要用于其他目的时,如分析某些参数的物理意义,就必须结合具体系统研究。

10.1.1 模型形式

假定观测得到的序列 $\{y(k)\}$ 是平稳的时间序列,不妨假定其均值为零(否则,由于其均值 \bar{y} 为常数,只要从所有观测数据 $\{y(k)\}$ 中均减去 \bar{y} 即可)。

在一定条件下,平稳时间序列可看做是以白噪声为输入的线性定常随机系统的响应,这一点是本章分析的基础。下面的讨论就是围绕如何确定线性定常系统,系统的示意图如图 10.1 所示,其中 $\{w(k)\}$ 为零均值、白噪声序列。系统的传递函数 $G(Z)$ 为

$$G(Z) = \frac{1 - \theta_1 Z^{-1} - \theta_2 Z^{-2} - \cdots - \theta_m Z^{-m}}{1 - \phi_1 Z^{-1} - \phi_2 Z^{-2} - \cdots - \phi_n Z^{-n}} \quad (10.1)$$

式中, Z^{-1} 是向后移位算子: $Z^{-1}y(k) = y(k-1), \dots, Z^{-m}y(k) = y(k-m)$, 这时 $y(k)$ 满足如下

$$y(k) - \phi_1 y(k-1) - \cdots - \phi_n y(k-n) = w(k) - \theta_1 w(k-1) - \cdots - \theta_m w(k-m) \quad (10.2)$$

式(10.2)等号的左端是自回归部分,阶次为 n ;等号的右端是滑动平均(也称为移动平均)部分,阶次为 m ,所以称式(10.2)为 (n, m) 阶的自回归滑动平均(Auto Regressive Moving Average)模型,记为 $ARMA(n, m)$ 。 ϕ_i 称为自回归参数, θ_i 称为滑动平均参数。注意,为了与时间序列专著符号一致,这里采用的符号都是时间序列专著中常用的符号,这似乎与前几章的符号不一致,但这样处理更有利于参考有关时间序列方面的书籍和文章。

当 $\theta_i = 0, i = 1, 2, \dots, m$ 时,式(10.2)变为

$$y(k) - \phi_1 y(k-1) - \cdots - \phi_n y(k-n) = w(k) \quad (10.3)$$

此式称为 n 阶自回归模型,记为 $AR(n)$ 。

当式(10.2)中所有的 $\phi_i = 0, i = 1, 2, \dots, n$ 时,式(10.2)变为

$$y(k) = w(k) - \phi_1 w(k-1) - \cdots - \phi_m w(k-m) \quad (10.4)$$

称之为 m 阶滑动平均模型,记为 $MA(m)$ 。

本章所讨论的时间序列的建模就是指用动态数据来拟合 $ARMA$ 模型的。

为了讨论方便,引进 n 阶自回归算子

$$\begin{aligned} \Phi(Z^{-1}) &= 1 - \phi_1 Z^{-1} - \phi_2 Z^{-2} - \cdots - \phi_n Z^{-n} \\ &= 1 - \sum_{k=1}^n \phi_k Z^{-k} \end{aligned} \quad (10.5)$$

及 m 阶滑动平均算子

$$\begin{aligned} \Theta(Z^{-1}) &= 1 - \theta_1 Z^{-1} - \cdots - \theta_m Z^{-m} \\ &= 1 - \sum_{k=1}^m \theta_k Z^{-k} \end{aligned} \quad (10.6)$$

这时式(10.2)简写为

$$\Phi(Z^{-1})y(k) = \Theta(Z^{-1})w(k) \quad (10.7)$$

从系统分析的观点看, $\Phi(Z^{-1})=0$ 的根为系统的极点, 而 $\Theta(Z^{-1})=0$ 的根为系统的零点。为了保证系统是稳定的, 需限制 $\Phi(Z^{-1})=0$ 的根在单位圆内, 即 $|Z|<1$; 而为了保证系统是最小相位系统, 需要限制 $\Theta(Z^{-1})=0$ 的根在单位圆内, 这时也称系统是可逆的, 即现在的输出 $y(k)$ 可分解为以往输出 $y(k)$ 的线性组合。

10.1.2 格林函数

为了讨论模型的一些性质, 引入格林函数的概念。首先看简单的 AR(1)模型

$$y(k) - \phi_1 y(k-1) = w(k) \quad (10.8)$$

或

$$(1 - \phi_1 Z^{-1})y(k) = w(k)$$

或

$$\begin{aligned} y(k) &= (1 - \phi_1 Z^{-1})^{-1} w(k) \\ &= \sum_{j=0}^{\infty} (\phi_1 Z^{-1})^j w(k) \\ &= \sum_{j=0}^{\infty} \phi_1^j w(k-j) \end{aligned} \quad (10.9)$$

方程式(10.8)是一阶差分方程, 左端是齐次的, 右端是“驱动函数”。式(10.9)给出方程式(10.8)的解, 它是驱动函数值的线性组合, 解的实质部分是这个展开式的系数 ϕ_1^j , 称之为“格林函数”。

如果以 G_j 记格林函数, 则对 AR(1)模型有

$$G_j = \phi_1^j \quad (10.10)$$

式(10.9)可写为

$$y(k) = \sum_{j=0}^{\infty} G_j w(k-j) = \sum_{j=-\infty}^k G_{k-j} w(j) \quad (10.11)$$

式中, $G_0=1$ 。

作为系统的动态特性, 格林函数可以有如下两种解释: 在展开式(10.11)中, G_j 是 j 个时间单位以前加于系统的刺激 $w(k-j)$ 对现在响应的权重, G_j 表示了系统对 $w(k-j)$ 记忆的强度。在 AR(1)系统中, ϕ_1 越大, 在确定的 j 值下, 对刺激 $w(k-j)$ 记忆越明显, 而且越能记住更远的过去。例如, $\phi_1=0.9$, 则 $G_6=0.9^6=0.532$, 表示对 $w(k-6)$ 的记忆是相当强的; 若 $\phi_1=0.4$, 则权 $G_6=0.4^6=0.004$, 表示对 $w(k-6)$ 的记忆非常弱, 几乎可以忘记。

对格林函数的第二种解释是它表示系统对任意刺激 $w(k)$ 的动态响应衰减的快慢特征。换句话说, 如果有单个 $w(k)$ 加入系统, 格林函数便决定了系统返回到它的平衡位置(一般取为0)的快慢程度, 显然, 如果 ϕ_1 值越小, 系统响应的衰减越快。

另外, 用格林函数给出的 $y(k)$ 的展开式(10.11)也可看做 $y(k)$ 的一个“正交分解”, 对随机变量而言, 独立或不相关与正交性是一样的。如果 $w(k)$ 为“轴”, 则格林函数 G_j 便是表示 $y(k)$ 为无限维空间中正交向量 $G_j w(k-j)$ 的和。

对于 ARMA(2,1)模型的格林函数, 下面给出两种形式: 隐式和显式。

1. 隐式

ARMA(2,1)模型为

$$y(k) - \phi_1 y(k-1) - \phi_2 y(k-2) = w(k) - \theta_1 w(k-1) \quad (10.12)$$

或

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2})y(k) = (1 - \theta_1 Z^{-1})w(k) \quad (10.13)$$

用 $y(k) = \sum_{j=0}^{\infty} G_j w(k-j) = (\sum_{j=0}^{\infty} G_j Z^{-j})w(k)$ 代入方程式(10.13)得

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2})(\sum_{j=0}^{\infty} G_j Z^{-j})w(k) = (1 - \theta_1 Z^{-1})w(k)$$

由于 $\{w(k)\}$ 是正交的, 故有

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2})(G_0 + G_1 Z^{-1} + G_2 Z^{-2} + \cdots) = (1 - \theta_1 Z^{-1})$$

由 Z 的同次幂系数相等得

$$\begin{aligned} G_0 &= 1 \\ G_1 - \phi_1 G_0 &= -\theta_1 \Rightarrow G_1 = \phi_1 - \theta_1 \\ G_2 - \phi_1 G_1 - \phi_2 G_0 &= 0 \Rightarrow G_2 = \phi_1^2 - \phi_1 \theta_1 + \phi_2 \\ G_j &= \phi_1 G_{j-1} + \phi_2 G_{j-2}, j \geq 2 \end{aligned} \quad (10.14)$$

即

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2})G_j = 0, j \geq 2 \quad (10.15)$$

由此可见, 若已知模型参数值, 格林函数很容易求得。例如, $\phi_1 = 1.3, \phi_2 = -0.4, \theta_1 = 0.4$, 则

$$\begin{aligned} G_0 &= 1 \\ G_1 &= \phi_1 - \theta_1 = 0.9 \\ G_2 &= \phi_1 G_1 + \phi_2 G_0 = 1.3 \times 0.9 - 0.4 = 0.77 \\ G_3 &= \phi_1 G_2 + \phi_2 G_1 = 1 - 0.36 = 0.64 \end{aligned}$$

应当注意, 格林函数 G_j 所满足的齐次方程式(10.15)正是模型式(10.13)的左端(自回归方程), 依上述方法不难证明, 对任意 ARMA(n, m)模型也有

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2} - \cdots - \phi_n Z^{-n})G_j = 0 \quad j \geq \max(n, m+1)$$

2. 显式

假设 ARMA(2,1)模型分解为

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2}) = (1 - \lambda_1 Z^{-1})(1 - \lambda_2 Z^{-1})$$

那么, $\lambda_1 + \lambda_2 = \phi_1, \lambda_1 \lambda_2 = -\phi_2$ 。其中 λ_1, λ_2 是二阶差分方程给出的特征方程 $\lambda^2 - \phi_1 \lambda - \phi_2 = 0$ 的两个根, 即

$$\lambda_1, \lambda_2 = \frac{1}{2}(\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2})$$

当 $\phi_1^2 + 4\phi_2 > 0$ 时

$$\begin{aligned} y(k) &= \frac{(1 - \theta_1 Z^{-1})w(k)}{(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2})} \\ &= \frac{1 - \theta_1 Z^{-1}}{(1 - \lambda_1 Z^{-1})(1 - \lambda_2 Z^{-1})}w(k) \end{aligned}$$

$$\begin{aligned}
&= \left[\frac{1 - \frac{\theta_1}{\lambda_1}}{1 - \frac{\lambda_2}{\lambda_1}} \cdot \frac{1}{1 - \lambda_1 Z^{-1}} + \frac{1 - \frac{\theta_1}{\lambda_2}}{1 - \frac{\lambda_1}{\lambda_2}} \cdot \frac{1}{1 - \lambda_2 Z^{-1}} \right] w(k) \\
&= \left[\frac{\lambda_1 - \theta_1}{\lambda_1 - \lambda_2} \cdot \frac{1}{1 - \lambda_1 Z^{-1}} + \frac{\lambda_2 - \theta_1}{\lambda_2 - \lambda_1} \cdot \frac{1}{1 - \lambda_2 Z^{-1}} \right] w(k) \\
&= \sum_{j=0}^{\infty} \left[\left(\frac{\lambda_1 - \theta_1}{\lambda_1 - \lambda_2} \right) \lambda_1^j + \left(\frac{\lambda_2 - \theta_1}{\lambda_2 - \lambda_1} \right) \lambda_2^j \right] w(k-j)
\end{aligned}$$

于是格林函数为

$$G_j = \left(\frac{\lambda_1 - \theta_1}{\lambda_1 - \lambda_2} \right) \lambda_1^j + \left(\frac{\lambda_2 - \theta_1}{\lambda_2 - \lambda_1} \right) \lambda_2^j \quad (10.16)$$

在上例中,特征根 $\begin{cases} \lambda_1 = \frac{1}{2}(1.3 + \sqrt{1.69 - 0.16}) = 0.8 \\ \lambda_2 = \frac{1}{2}(1.3 - \sqrt{1.69 - 0.16}) = 0.5 \end{cases}$, 格林函数为

$$\begin{aligned}
G_j &= \frac{0.8 - 0.4}{0.8 - 0.5} \times (0.8)^j + \frac{0.5 - 0.4}{0.5 - 0.8} \times (0.5)^j \\
&= \frac{0.4}{0.3} \times (0.8)^j - \frac{0.1}{0.3} \times (0.5)^j \\
G_0 &= 1, \quad G_1 = 0.9, \quad G_2 = 0.77, \quad \dots
\end{aligned}$$

与隐式结果完全相同。

当 $\phi_1^2 + 4\phi_2 < 0$ 时, 即 λ_1, λ_2 为两个共轭复根时, 复根可写为: $\lambda = a + jb = \gamma e^{j\theta}$, 则格林函数为

$$\begin{aligned}
G_j &= \gamma^j \left[\cos j\theta + \frac{\phi_1 - 2\theta_1}{\sqrt{-(\phi_1^2 + 4\phi_2)}} \sin j\theta \right] \\
&= \gamma^j A \sin(\beta + j\theta)
\end{aligned} \quad (10.17)$$

式中

$$\begin{aligned}
\gamma &= \sqrt{a^2 + b^2} = \sqrt{-\phi_2}, \quad \beta = \arctan \left[\frac{\sqrt{-(\phi_1^2 + 4\phi_2)}}{\phi_1 - 2\theta_1} \right] \\
\theta &= \arctan \left[\frac{\sqrt{-(\phi_1^2 + 4\phi_2)}}{\phi_1} \right], \quad A = \sqrt{1 + \left[\frac{\phi_1 - 2\theta_1}{\sqrt{-(\phi_1^2 + 4\phi_2)}} \right]^2}
\end{aligned}$$

10.1.3 稳定性

当仅有一个 $w(k)$ 加入系统时, 如果在充分长时间后, 系统逐渐恢复到其平衡位置 (一般取为 0), 则称该系统是渐近稳定的。如果在充分长时间后, 系统并不回到平衡位置, 但输出响应始终不超过 $w(k)$, 则称系统是稳定的。如果无论多么小的单个的 $w(k)$ 加入系统, 当时间充分长后, 输出响应总能超过任意给定的界限, 则称该系统是不稳定的。

可以利用格林函数及特征根来讨论系统的稳定性。例如, 对 ARMA(2, 1) 模型

$$(1 - \phi_1 Z^{-1} - \phi_2 Z^{-2})y(k) = (1 - \theta_1 Z^{-1})w(k)$$

如果 λ_1, λ_2 为两个不同的特征根, 格林函数为

$$G_j = \frac{\lambda_1 - \theta_1}{\lambda_1 - \lambda_2} \lambda_1^j + \frac{\lambda_2 - \theta_1}{\lambda_2 - \lambda_1} \lambda_2^j$$

(1) $|\lambda_1| < 1, |\lambda_2| < 1$, 当 $j \rightarrow \infty$ 时, $G_j \rightarrow 0$, 系统渐近稳定。

(2) $|\lambda_1| > 1$ 或 $|\lambda_2| > 1$, 或二者皆大于 1, 当 $j \rightarrow \infty$ 时, $G_j \rightarrow \infty$, 系统不稳定。

(3) $|\lambda_1| = 1, |\lambda_2| < 1$ (或 $|\lambda_2| = 1, |\lambda_1| < 1$), 当 $j \rightarrow \infty$ 时, $G_j \rightarrow \frac{\lambda_1 - \theta_1}{\lambda_1 - \lambda_2}$ ($G_j \rightarrow \frac{\lambda_2 - \theta_1}{\lambda_2 - \lambda_1}$), 系

统稳定但不渐近稳定。

(4) $|\lambda_1| = |\lambda_2| = 1$, 且 $\lambda_1 = -\lambda_2$, 这时格林函数为

$$G_j = \left(\frac{1 - \theta_1}{2}\right)(1)^j + \left(\frac{1 + \theta_1}{2}\right)(-1)^j \quad (10.18)$$

故当 $j \rightarrow \infty$ 时, $G_j \rightarrow 1$ 或 $-\phi_1$, 所以系统稳定而不渐近稳定。

(5) $\lambda_1 = \lambda_2 = 1$, 这时基于两根不等的格林函数式(10.16)不能用了, 但可直接求得

$$\begin{aligned} y(k) &= \frac{1 - \theta_1 Z^{-1}}{(1 - Z^{-1})(1 - Z^{-1})} w(k) \\ &= (1 + Z^{-1} + Z^{-2} + \cdots)(1 + Z^{-1} + Z^{-2} + \cdots)(1 - \theta_1 Z^{-1})w(k) \\ &= (1 + 2Z^{-1} + 3Z^{-2} + 4Z^{-3} + \cdots)(1 - \theta_1 Z^{-1})w(k) \\ &= \sum_{j=0}^{\infty} [(j+1) - \theta_1 j] w(k-j) \\ &= \sum_{j=0}^{\infty} [(1 - \theta_1)j + 1] w(k-j) \end{aligned} \quad (10.19)$$

故当 $j \rightarrow \infty$ 时, $G_j = (1 - \theta_1)j + 1 \rightarrow \infty$, 所以系统不稳定。

综合上述: ARMA(2,1)或 AR(2)的稳定条件可写成

$$|\lambda_1| \leq 1, |\lambda_2| \leq 1$$

而若 $\lambda_1 = \lambda_2$ 时, 则 $|\lambda_1| = |\lambda_2| < 1$ 。

顺便指出, 对于 ARMA(2,1)模型, 显然有

$$\phi_1 = \lambda_1 + \lambda_2, \phi_2 = -\lambda_1 \lambda_2$$

所以, 当 $|\lambda_1| < 1, |\lambda_2| < 1$ 时, 则有

$$|\phi_2| < 1$$

即

$$-1 < \phi_2 < 1$$

另外, 由于

$$\lambda_1(1 - \lambda_2) < (1 - \lambda_2)$$

或

$$(\lambda_1 + \lambda_2) - \lambda_1 \lambda_2 < 1$$

得

$$\phi_1 + \phi_2 < 1$$

而且

$$-(1 + \lambda_2) < \lambda_1(1 + \lambda_2)$$

或

$$-\lambda_1 \lambda_2 - (\lambda_1 + \lambda_2) < 1$$

得

$$\phi_2 - \phi_1 < 1$$

因此, ARMA(2,1)系统渐近稳定的条件可为

$$\phi_1 + \phi_2 < 1$$

$$\phi_2 - \phi_1 < 1$$

$$|\phi_2| < 1$$

$$(10.20)$$

类似地推导, 可得更一般的结论(S M Wu 和 S M Pandit, 1979)。对 ARMA(n, m)渐近稳定性条件为 $|\lambda_k| < 1, k=1, 2, \dots, n, G_j \rightarrow 0$; 稳定条件为 $|\lambda_k| \leq 1, k=1, 2, \dots, n, G_j$ 有界, 当 $\lambda_i =$

$\lambda_j, i \neq j$ 时, 为 $|\lambda_i| = |\lambda_j| < 1$ 。

应当指出, 格林函数和稳定性分析与本书的主要目的(建模与辨识)并无直接关系, 只是因为它们是序列分析与系统分析中的基本概念和基本方法, 才略给予介绍, 如果不做系统分析, 略去此部分并不影响本章给出的建模方法。

10.2 模型的参数估计

10.2.1 AR(n) 模型的参数估计

假定得到了 N 个观点数据 $y(1), y(2), \dots, y(N)$ (以后不加声明, 所说的“数据”, 均指的是减去了平均值 \bar{y} 以后的数据)。

AR(n) 模型为

$$y(k) = \phi_1 y(k-1) + \dots + \phi_n y(k-n) + w(k) \quad (10.21)$$

显然可用最小二乘法估计参数。

令

$$\mathbf{y} = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} y(n) & y(n-1) & \dots & y(1) \\ y(n+1) & y(n) & \dots & y(2) \\ \dots & \dots & \dots & \dots \\ y(N-1) & y(N-2) & \dots & y(N-n) \end{bmatrix} \quad \boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix}$$

最小二乘法估计为

$$\hat{\boldsymbol{\phi}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (10.22)$$

而 $w(k)$ 的方差为

$$\sigma_w^2 = \frac{1}{N - n} \sum_{k=n+1}^N (y(k) - \phi_1 y(k-1) - \dots - \phi_n y(k-n))^2 \quad (10.23)$$

10.2.2 ARMA(n, m) 模型的参数估计

对于 AR(n) 模型可用最小二乘法估计参数, 但对 ARMA(n, m) 就不可以, 这是因为当模型中出现滑动平均项时(即使是 MA(1) 模型), 回归方程就是非线性的了, 如 ARMA(2, 1) 模型。

$$y(k) = \phi_1 y(k-1) + \phi_2 y(k-2) - \theta_1 w(k-1) + w(k) \quad (10.24)$$

由此得到

$$w(k-1) = y(k-1) - \phi_1 y(k-2) - \phi_2 y(k-3) + \theta_1 w(k-2)$$

$$w(k-2) = y(k-2) - \phi_1 y(k-3) - \phi_2 y(k-4) + \theta_1 w(k-3)$$

...

代入式(10.24)有

$$y(k) = \phi_1 y(k-1) + \phi_2 y(k-2) - \theta_1 w(k-1) + w(k)$$

$$= \phi_1 y(k-1) + \phi_2 y(k-2) - \theta_1 [y(k-1) - \phi_1 y(k-2)$$

$$- \phi_2 y(k-3) + \theta_1 w(k-2)] + w(k)$$

$$= (\phi_1 - \theta_1) y(k-1) + (\phi_2 + \theta_1 \phi_1) y(k-2) + \theta_1 \phi_2 y(k-3) - \theta_1^2 w(k-2) + w(k)$$

$$\begin{aligned}
&= (\phi_1 - \theta_1)y(k-1) + (\phi_2 + \theta_1\phi_1)y(k-2) + \theta_1\phi_2y(k-3) \\
&\quad - \theta_1^2[y(k-2) - \phi_1y(k-3) - \phi_2y(k-4) + \theta_1w(k-3)] + w(k) \\
&\quad \dots
\end{aligned} \tag{10.25}$$

因此,对参数的估计需用非线性最小二乘法。具体算法见第8章。

正如前面所看到的,非线性最小二乘法一般从被估计参数的某个初始推测值出发逐步求出所要的参数,所用到的 $w(k)$ 由下式计算

$$w(k) = y(k) - \phi_1y(k-1) - \dots - \phi_ny(k-n) + \theta_1w(k-1) + \dots + \theta_mw(k-m) \tag{10.26}$$

其中, $k=n+1, n+2, \dots, N$, 当 $k \leq n$ 时, $w(k)=0$ (它的期望值), 而其中参数是由最近一次迭代得到的估计值。

初始的推测值是很重要的, 初始值不好, 对一些算法来说, 不仅影响收敛速度, 甚至可能使系统不收敛。

10.2.3 初值的求法

对于 ARMA(n, m) 模型, 求自回归参数 $\phi_1, \phi_2, \dots, \phi_n$ 的初始值, 最简单、但比较粗糙的方法是略去滑动平均参数, 而通常按线性最小二乘法得到 ϕ_i 的估计。此方法虽然粗糙, 但有时也得到出乎意外好的近似。尽管如此, 这毕竟不是一般性的, 而且对滑动平均参数再进行估计时仍是比较复杂的。这里介绍求参数初值的两种普遍有效的方法。

1. 逆函数法

ARMA(n, m) 模型可以写为

$$(1 - \phi_1Z^{-1} - \phi_2Z^{-2} - \dots - \phi_nZ^{-n})y(k) = (1 - \theta_1Z^{-1} - \dots - \theta_mZ^{-m})w(k) \tag{10.27}$$

若系统为可逆的(最小相位), 即多项式 $\lambda^m - \theta_1\lambda^{m-1} - \dots - \theta_m = 0$ 的根在单位圆内。由式(10.27)可以解得 $w(k)$ 为

$$\begin{aligned}
w(k) &= \frac{1 - \phi_1Z^{-1} - \phi_2Z^{-2} - \dots - \phi_nZ^{-n}}{1 - \theta_1Z^{-1} - \theta_2Z^{-2} - \dots - \theta_mZ^{-m}}y(k) \\
&\triangleq (1 - r_1Z^{-1} - r_2Z^{-2} - \dots)y(k)
\end{aligned} \tag{10.28}$$

将式(10.28)代入式(10.27)得恒等式为

$$\begin{aligned}
&(1 - \phi_1Z^{-1} - \phi_2Z^{-2} - \dots - \phi_nZ^{-n})y(k) \\
&= (1 - \theta_1Z^{-1} - \dots - \theta_mZ^{-m})(1 - r_1Z^{-1} - r_2Z^{-2} - \dots)y(k)
\end{aligned}$$

由 Z 的同次幂对应系数相等得

$$\begin{aligned}
\phi_1 &= \theta_1 + r_1 \\
\phi_2 &= \theta_2 - \theta_1r_1 + r_2 \\
\phi_3 &= \theta_3 - \theta_1r_2 - \theta_2r_1 + r_3 \\
&\dots \\
\phi_j &= \theta_j - \theta_1r_{j-1} - \theta_2r_{j-2} - \dots - \theta_{j-1}r_1 + r_j
\end{aligned} \tag{10.29}$$

对于 ARMA(n, m) 模型, 当 $j > m$ 时, $\theta_j = 0$, 而当 $j > n$ 时, $\phi_j = 0$, 特别当 $j > \max(n, m)$ 时, 由式(10.29)的最后一个等式得

$$(1 - \theta_1Z^{-1} - \dots - \theta_mZ^{-m})\gamma_j = 0 \tag{10.30}$$

上式中 $\gamma_{j-k} \equiv Z^{-k} r_j$ 。显然,如果 $\{r_j\}$ 的值已知,则可利用式(10.30)求得滑动平均参数的初值,再由式(10.29)求得自回归参数的初始值。

由此得到利用逆函数求 ARMA(n, m)模型参数初始值的步骤如下。

(1) 估计 $\{r_j\}$

取充分大的正整数 $p(p \geq \max(n, m) + m)$,考虑自回归模型 $AR(p)$

$$w(k) = (1 - r_1 Z^{-1} - r_2 Z^{-2} - \cdots - r_p Z^{-p})y(k) \quad (10.31)$$

用最小二乘法求出 r_j 的估计 \hat{r}_j 。

注意:式(10.31)是有限项,不严格与式(10.28)相等,但由于系统是可逆的,当 j 充分大时, $r_j \rightarrow 0$ 。因此,只要 p 充分大,利用 $AR(p)$ 可得到 r_j 相当好的估计,如果 p 值较小,则可能造成较大的误差。

(2) 求 θ_j 的估计

在式(10.30)中,令 $j = n+1, n+2, \dots, n+m$,得方程组(这里假定 $n \geq m$,若 $m > n$,则令 $j = m+1, \dots, m+n$,下式中 m 和 n 的位置互换即可)

$$\begin{pmatrix} r_n & r_{n-1} & \cdots & r_{n-m+1} \\ r_{n+1} & r_n & \cdots & r_{n-m+2} \\ \cdots & & \cdots & \cdots \\ r_{n+m-1} & r_{n+m-2} & \cdots & r_n \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{pmatrix} = \begin{pmatrix} r_{n+1} \\ r_{n+2} \\ \vdots \\ r_{n+m} \end{pmatrix} \quad (10.32)$$

用 \hat{r}_i 代替式(10.32)中 r_j ,并解此方程组得 θ_j 的估计 $\hat{\theta}_j$ 。

(3) 校验参数 $\hat{\theta}_i$

由式(10.32)得到的估计值,未必满足可逆性条件。为了判断是否满足此条件,首先按下式求根 λ_i

$$\lambda^m - \theta_1 \lambda^{m-1} - \cdots - \theta_m = (\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_m) \quad (10.33)$$

如果 $|\lambda_i| > 1$,则以 $\frac{1}{\lambda_i}$ 代替 λ_i ,并且用下式(由多项式根与系数的关系)重新求 θ_j 估计

$$\hat{\theta}_j = (-1)^{j+1} \sum_{\substack{i_1 < i_2 < \cdots < i_j \\ i_1, i_2, \dots, i_j = 1}}^m \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_j} \quad j = 1, 2, \dots, m \quad (10.34)$$

(4) 求 ϕ_j 的估计

由式(10.29)中的前 n 个方程求 ϕ_j 的估计

$$\begin{pmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \vdots \\ \hat{\phi}_n \end{pmatrix} = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_n \end{pmatrix} + \begin{pmatrix} 1 & & & & \\ -\hat{r}_1 & & & & \\ \cdots & & & & \\ -\hat{r}_{n-1} & -\hat{r}_{n-2} & \cdots & -\hat{r}_1 & 1 \end{pmatrix} \begin{pmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_n \end{pmatrix} \quad (10.35)$$

其中,当 $j > m$ 时, $\theta_i = 0$ 。

为了具体说明上述步骤,下面给出用太阳黑子数据(见表 10.1)求 ARMA(2,1)模型参数初值的例子。

表 10.1 1749~1924 年每年太阳黑子数

60.9	63.4	47.7	47.3	30.7	12.2	9.6	10.6	32.4	47.6
50.0	62.9	59.5	61.2	45.1	36.4	20.9	11.4	37.5	69.9
101.1	100.8	81.6	66.5	34.8	30.4	7.9	19.8	92.5	154.4
115.9	81.8	65.1	36.5	32.8	10.2	24.1	32.9	132.6	130.9
114.1	89.9	66.6	60.0	46.9	41.0	21.3	16.0	6.4	4.1
6.8	14.5	34.6	45.0	43.1	47.5	42.2	20.1	10.1	8.1
2.5	9.0	1.4	5.0	12.2	13.9	35.4	45.8	41.1	30.4
23.9	15.7	6.6	4.0	1.8	8.5	16.6	36.3	49.7	62.5
67.0	71.0	47.9	27.5	8.5	13.2	56.9	121.5	138.3	103.2
85.8	69.2	36.5	24.2	10.7	15.0	40.1	61.5	98.5	124.3
93.9	66.5	64.5	54.2	39.0	20.6	6.7	41.3	22.8	54.8
93.8	99.7	77.3	59.1	44.0	47.0	30.5	16.3	7.3	37.3
93.9	139.1	111.3	101.7	66.3	44.7	17.1	11.3	12.3	3.4
6.0	33.3	54.3	59.7	63.7	63.5	52.2	25.4	13.1	6.5
6.3	7.1	35.6	70.3	84.9	78.0	64.0	1.5	26.2	36.7
12.1	9.5	2.7	5.0	24.4	42.0	63.5	53.8	62.0	48.5
113.9	13.6	5.7	3.6	1.4	9.6	47.4	57.1	103.9	50.6
69.6	37.6	26.1	14.2	5.8	16.7				

【例 10.1】 考虑 ARMA(2,1)模型

这里 $n=2, m=1$, 取 $p=\max(n,m)+m=3$ 。

首先用最小二乘法拟合 AR(3)模型, 求出 r_j 。AR(3)模型可写为

$$y(k) = r_1 y(k-1) + r_2 y(k-2) + r_3 y(k-3) + w(k) \quad (10.36)$$

按线性最小二乘法有

$$\mathbf{y} = \begin{pmatrix} y_4 \\ y_5 \\ \vdots \\ y_N \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} y_3 & y_2 & y_1 \\ y_4 & y_3 & y_2 \\ \vdots & \vdots & \vdots \\ y_{N-1} & y_{N-2} & y_{N-3} \end{pmatrix}$$

$$\begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{pmatrix} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \begin{pmatrix} 1.27 \\ -0.50 \\ -0.11 \end{pmatrix}$$

残差平方和为 42203.19。

由于 $m=1$, 所以由式(10.32)得 $\hat{r}_2 \hat{\theta}_1 = \hat{r}_3$, 即

$$\hat{\theta}_1 = \frac{\hat{r}_3}{\hat{r}_2} = \frac{-0.11}{-0.50} = 0.22$$

由于 $\lambda - \hat{\theta}_1 = 0$ 的根为 $\lambda = 0.22 < 1$, 所以 θ_1 满足可逆条件。再利用式(10.35)求 ϕ_1, ϕ_2 的估计

$$\hat{\phi}_1 = \hat{r}_1 + \hat{\theta}_1 = 1.27 + 0.22 = 1.49$$

$$\hat{\phi}_2 = \hat{r}_2 - \hat{r}_1 \hat{\theta}_1 + \hat{\theta}_2 = -0.50 - 1.27 \times 0.22 + 0 = -0.78$$

因为对于 ARMA(2,1)模型, $\theta_2 = 0$, 故得 ARMA(2,1)模型的初始值为

$$\hat{\phi}_1 = 1.49$$

$$\hat{\phi}_2 = -0.78$$

$$\hat{\theta}_1 = 0.22$$

2. 两次最小二乘法

所考虑的模型仍为式(10.27)。像逆函数法一样,首先对充分大的 p 拟合模型

$$w(k) = (1 - r_1 Z^{-1} - r_2 Z^{-2} - \cdots - r_p Z^{-p})y(k) \quad (10.37)$$

得到 r_j 的最小二乘估计 \hat{r}_j 。

然后用 \hat{r}_j 代替式(10.37)中 r_j ,并由式(10.37)计算 \hat{w} ,将这些估计值看做输入,将观测值 y 作为输出,再用最小二乘法拟合 ARMA(n, m)模型,即

$$\begin{aligned} (1 - a_1 Z^{-1} - a_2 Z^{-2} - \cdots - a_n Z^{-n})y(k) \\ = (b_0 - b_1 Z^{-1} - \cdots - b_m Z^{-m})w(k) \end{aligned} \quad (10.38)$$

或

$$= b_0 \left(1 - \frac{b_1}{b_0} Z^{-1} - \cdots - \frac{b_m}{b_0} Z^{-m} \right) w(k)$$

考虑到放大倍数 b_0 可以归到噪声强度 σ_w 中,故式(10.38)右端首项系数应取 1,与式(10.27)比较可知

$$\begin{cases} \hat{\phi}_i = \hat{a}_i & i = 1, 2, \dots, n \\ \hat{\theta}_i = \hat{b}_i / b_0 & i = 1, 2, \dots, m \end{cases} \quad (10.39)$$

具体步骤如下:

- (1)用最小二乘法拟合式(10.37)(使 $w(k)$ 平方和最小);
- (2)用 \hat{r}_j 代替 r_j ,由式(10.37)计算 $w(k)$, $k = p+1, \dots, N$;
- (3)用最小二乘拟合式(10.38);
- (4)求 $\hat{\phi}_i = \hat{a}_i, \hat{\theta}_i = \hat{b}_i / b_0$ 。

实践表明,采用此种方法得到的初估计,比逆函数法往往更接近终值。另外,两次最小二乘法当模型的阶增加或减少时,还可以用 5.2 节给出的递推算法,简单易行。

10.3 模型阶的确定

要建立 ARMA(n, m)模型,首先要知道 n, m 是多少,但估计 n 和 m 并不是容易的,除非 m 为零。一般地,为了要得到 n 和 m ,只有逐次进行模型的比较,如要求一个 ARMA(5,3)模型,就需要比较多达 22 个模型。当阶数更高时,计算量就会变得非常大。为了简化这一过程, Wu 和 Pandit 提出了一种新的策略,即总是用 ARMA($n, n-1$)模型,这样就把定阶时需要比较搜索的范围大大缩小,如确定一个 ARMA(5,3)模型,只需要比较 5 个模型即可。问题是这样的策略是否合理?事实上,对一个平稳随机序列,总可以找到一个 ARMA($n, n-1$)模型满足事先要求的精度,详细的讨论读者可参见文献 Wu 和 Pandit (1980)。这一点是这里讨论模型定阶的主要基础。

从这个做法再一次可以看出,时间序列建模的结果用于它自身的预测是完全合适的,但由于模型本身的结构(阶)是靠拟合精度来决定的,所以并不能保证它能满足其他的实际约束,因

此用于其他目的时必须小心。

现在来讨论模型的阶 n 如何确定,确定阶 n 的原则像在 5.1 节中介绍的一样,仍然是根据残差平方和有无显著变化来决定的。从低阶到高阶拟合模型,如果更高阶的残差平方和有显著的减少,则意味着现有的模型不适用;另一方面,如果现有的模型是适用的,而我们试图用更高阶的模型,则残差平方和减少不显著,甚至可能略有增加。同时,更高阶的模型中新增加参数的绝对值也将很小(其置信区间包含零),这意味着新增加参数不必要,也即更高阶模型不必要,残差平方和变化的显著性判别用 F 检验(见 5.1 节)。在实际建模过程中,为了减少非线性拟合的次数,尽快地确定出适用的模型,上述确定阶的办法可简化为在低阶模型 ARMA($2n, 2n-1$)和高阶模型 ARMA($2n+2, 2n+1$)之间做出选择。

具体地说,ARMA($2n, 2n-1$)模型对 ARMA($2n+2, 2n+1$)模型检验:记 $J(n)$ 为 ARMA($2n, 2n-1$)模型的残差($w(k)$)平方和; $J(n+1)$ 为 ARMA($2n+2, 2n+1$)模型的残差($w(k)$)平方和; N 为观测数据的数目; p_n, p_{n+1} 分别为 ARMA($2n, 2n-1$)模型和 ARMA($2n+2, 2n+1$)模型中所包含的参数个数,那么, $p_n = 4n-1, p_{n+1} = 4n+3$ 。则统计量

$$F \triangleq \frac{J(n) - J(n+1)}{J(n+1)} \cdot \frac{N - p_{n+1}}{p_{n+1} - p_n} = \frac{J(n) - J(n+1)}{J(n+1)} \cdot \frac{N - 4n - 3}{4}$$

渐近服从自由度为 4 和 $(N-4n-3)$ 的 F 分布。

给定置信度 α (如取 5%),由 F 分布表查得(或 5.1 节中给出的近似公式计算)相应于 α 的 $F(4, N-4n-3)$ 的临界值 F_α 。

若 $F \geq F_\alpha$,则认为模型 ARMA($2n+2, 2n+1$)比 ARMA($2n, 2n-1$)的残差平方和有显著的改善,因此 ARMA($2n, 2n-1$)模型不适用。

若 $F \leq F_\alpha$,则认为模型 ARMA($2n+2, 2n+1$)与 ARMA($2n, 2n-1$)的残差平方和无显著变化,因此 ARMA($2n, 2n-1$)即为适用模型。

有了上述准则,下面就可以从 $n=1$ 开始逐次增加来拟合 ARMA($2n, 2n-1$)模型,每当 n 增加 1 时,就用 F 检验判断残差平方和有无显著变化,若有显著变化,则继续增加 n 值,直到在给定的置信度下无显著变化的 n_0 为止,并认为 ARMA($2n_0, 2n_0-1$)模型是适用的。如果 ϕ_{2n_0} 和(或) θ_{2n_0-1} 的绝对值很小或置信区间包含零,还可拟合参数较少的模型并用 F 检验,直到得到更简单的适用模型。

值得注意的是,上面给出的定阶过程,每一步都要重新计算初值和用非线性最小二乘拟合模型,而非线性最小二乘拟合的计算量相当大,因此应设法减少非线性最小二乘拟合的次数。鉴于两次最小二乘求得的初值与终值相差较小,所以可以先用两次最小二乘粗略确定(比较原则仍用 F 检验)ARMA 模型的阶和参数,作为“初估模型”,然后再用非线性最小二乘逐次拟合,直至得到理想的适用模型。

另外,寻求“初估模型”,可以采用 5.2 节所给的递推算法,只要将其中的判别准则改为 F 检验即可。实践表明,采用先求初估模型的方法,使整个建模过程大大简化,明显地缩短了建模时间。

综合上述,整个建模方案可归纳为如下步骤:

(1) 用两次最小二乘递推确定“初估模型”,具体地说,取充分大的数 p ,拟合模型式(10.37),再用拟合得到的参数值,由式(10.37)估算 $\{w(k)\}$,然后以 $\{w(k)\}$ 为输入、 $y(k)$ 为输出,用 5.2 节给的递推算法(将其中判别准则改为 F 检验)确定“初估模型”ARMA($2n_0, 2n_0-1$)。

(2) 拟合(均指非线性最小二乘拟合)ARMA($2n, 2n-1$)模型, $n = n_0, n_0+1, \dots, n$ 每增加 1,

用 F 准则检验残差平方和的改善情况,当从 $ARMA(2n, 2n-1)$ 到 $ARMA(2n+2, 2n+1)$ 残差变化不显著(在预定的置信度下,如 5%)时就停止,并选定 $ARMA(2n, 2n-1)$ 模型。

(3) 检查 ϕ_{2n} 、 θ_{2n-1} 的值,如果它们的值不太小或者它们的置信区间不包含零,则 $ARMA(2n, 2n-1)$ 为适用的模型。

(4) 如果 ϕ_{2n} 和(或) θ_{2n-1} 很小且置信区间包含零,则拟合 $ARMA(2n-1, 2n-2)$ 模型,并且用 F 准则将它与 $ARMA(2n, 2n-1)$ 模型做检验,如果不显著,则再拟合 $ARMA(2n-1, m)$, $m < 2n-2$,并用 F 检验,直到得出具有较少参数的适用模型为止。

(5) 如果用 F 准则检验 $ARMA(2n-1, 2n-2)$ 和 $ARMA(2n, 2n-1)$ 是显著的,则依步骤(4)的方法确定适用的 $ARMA(2n, m)$ 模型($m < 2n-1$)。

(6) 如有必要,可以拟合所希望的模型形式,诸如 $AR(n)$ 或 $MA(m)$, 等等。对阶数增加用 F 准则做检验,直到取得不显著的值为止。

上述步骤的简单框图如图 10.2 所示。

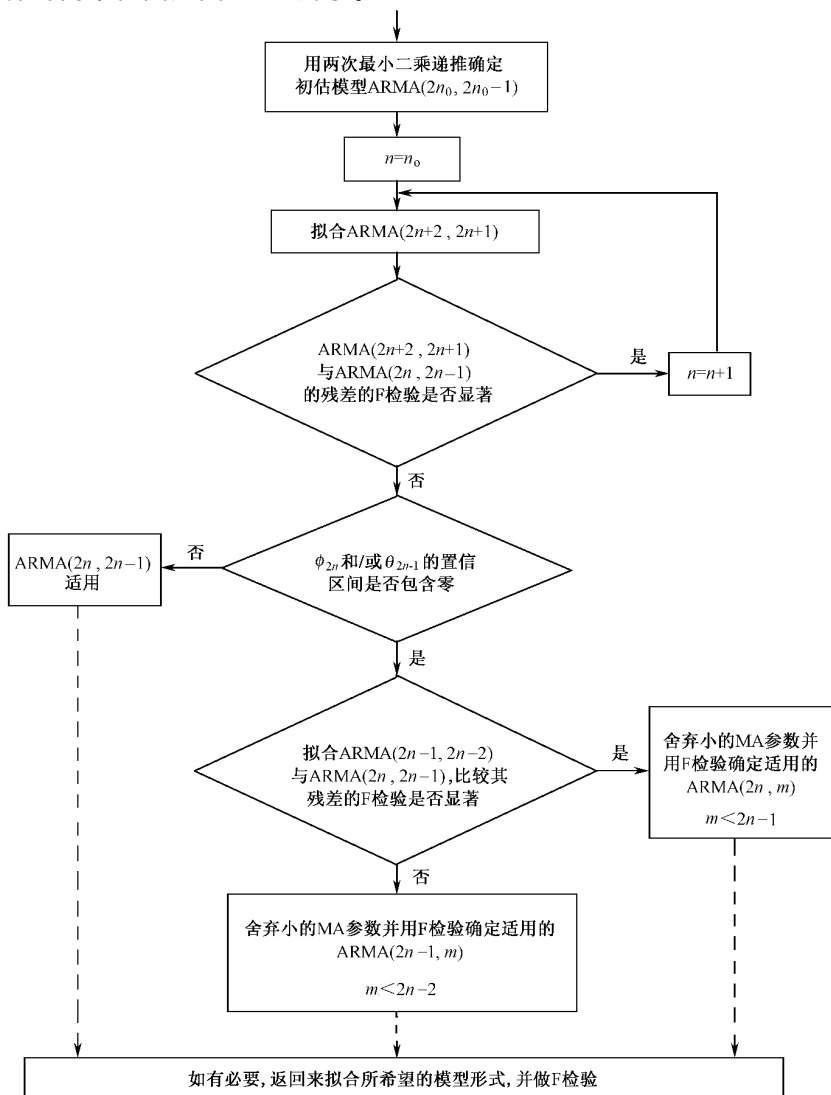


图 10.2 时间序列建模步骤流程框图

注意:当不用“初估模型”时,图 10.2 中第一框改为“拟合 ARMA(2,1)”,第二框改为“ $n=1$ ”。

【例 10.2】对太阳黑子数据的建模

表 10.1 所示的是 1749~1924 年按每年太阳黑子数目所表示的太阳内部活动记录,据此建立太阳黑子活动模型。

为寻找适合太阳黑子数据的模型,下面首先拟合 ARMA(2,1)和 ARMA(4,3)模型,具体步骤如下。

1. ARMA(2,1)模型

(1) 初值 $\phi_1 = 1.49, \phi_2 = -0.78, \theta_1 = 0.22$

对应于 AR(3)的残差平方和为 40843.66。

(2) 终值 95%置信区间

$\phi_1 = 1.41$ (1.25 ~ 1.57)

$\phi_2 = -0.71$ (-0.85 ~ -0.57)

$\theta_1 = 0.136$ (-0.09 ~ 0.36)

残差平方和为 40788.00。

2. ARMA(4,3)模型

(1) 初值 $\phi_1 = 0.7967, \phi_2 = -0.2231, \phi_3 = 0.2768, \phi_4 = -0.3785,$

$\theta_1 = -0.4845, \theta_2 = -0.3391, \theta_3 = 0.2652$

对应于 AR(7)模型的参差平方和为 39595.75。

(2) 终值 95%置信区间

$\phi_1 = 1.266$ (-0.291 ~ 2.823)

$\phi_2 = -0.837$ (-3.05 ~ 1.379)

$\phi_3 = 0.392$ (-1.616 ~ 2.400)

$\phi_4 = -0.254$ (-1.277 ~ 0.769)

$\theta_1 = -0.035$ (-1.590 ~ 1.520)

$\theta_2 = -0.336$ (-1.730 ~ 1.058)

$\theta_3 = 0.112$ (-0.145 ~ 0.369)

残差平方和为 39624.58。

ARMA(4,3)模型的残差平方和比 ARMA(2,1)的小。F 检验法

$$J_1 = 40788.00 \quad J_2 = 39624.58$$

$$p_2 - p_1 = 4, \quad n + m + 1 = 8 (\text{包含平均值 } \bar{y}), \quad N = 176$$

$$F = \frac{J_1 - J_2}{J_2} \cdot \frac{N - 8}{4} = 1.23$$

设 $\alpha = 0.05$, 得

$$F_{\alpha}(4, \infty) = 2.37$$

由于 $F < F_{\alpha}$, 所以适用模型的阶不需超过 ARMA(2,1)模型。

3. AR(2)模型

检查 ARMA(2,1)的参数 ϕ_2, θ_1 , 发现 θ_1 的绝对值较小, 并且它的置信区间包含零, 如果去

掉滑动平均模型参数 θ_1 , 那么拟合的纯自回归模型 AR(2)为

$$\begin{aligned}\phi_1 &= 1.32 & (1.210 \sim 1.430) \\ \phi_2 &= -0.63 & (-0.764 \sim -0.496)\end{aligned}$$

残差平方和为 41173.35。

$$\begin{aligned}J_1 &= 41173.35 & J_2 &= 40788.00 \\ p_2 - p_1 &= 1 & p_2 + 1 &= 4, & N &= 176 \\ F &= \frac{41173.35 - 40788.00}{40788.00} \times \frac{176 - 4}{1} = 1.62\end{aligned}$$

$F_{0.05}(1, \infty) = 3.48$, 由于 $F < F_\alpha$, 又 AR(2) 参数 ϕ_2 的置信区间不包含 0, 因此, 对太阳黑子活动数据来说, AR(2) 模型即 ARMA(2, 0) 可以认为是适用的模型, 辨识过程结果如表 10.2 所示。

表 10.2 太阳黑子活动的辨识结果

参数	ARMA 的阶数			
	(2, 1)	(4, 3)	(1, 0)	(2, 0)
ϕ_1	1.41 ± 0.16	1.27 ± 1.16	0.81 ± 0.09	1.32 ± 0.11
ϕ_2	-0.71 ± 0.14	-0.84 ± 2.22		-0.63 ± 0.14
ϕ_3		0.39 ± 2.01		
ϕ_4		-0.25 ± 1.02		
θ_1	0.14 ± 0.23	-0.04 ± 1.56		
θ_2		-0.34 ± 1.39		
θ_3		0.11 ± 0.26		
\bar{y}	44.8 ± 6.6	44.4 ± 6.8	44.7 ± 5.4	44.7 ± 7.4
残差平方和	40788.00	39624.58	76577.86	41173.35

【例 10.3】 用高斯法求参数估计

假定取得数据为

k	1	2	3	4	5	6	7	8	9	10
$y(k)$	0.2	0.4	0.5	0.42	0.3	0.35	0.21	0.12	-0.06	-0.13

用此数据拟合 MA(1) 模型。正如在 10.2 节指出的, 这时估计需用非线性最小二乘法。现在用高斯算法来解这一问题。

MA(1) 模型可以写为

$$w(k) = y(k) + \theta w(k - 1)$$

取 θ 的初始推测值为 $\theta^0 = -0.8$, 则 $w(k)$ 可用下式

$$w(k, \theta^0) = y(k) - 0.8w(k - 1)$$

取 $w(0) = 0$, 递推地算出

$$\begin{aligned}w(1, \theta^0) &= y(1) - 0.8w(0) = 0.2 - 0.8 \times 0 = 0.2 \\ w(2, \theta^0) &= y(2) - 0.8w(1) = 0.4 - 0.8 \times 0.2 = 0.24 \\ w(3, \theta^0) &= 0.308 \\ w(4, \theta^0) &= 0.1736\end{aligned}$$

$$w(5, \theta^0) = 0.1611$$

$$w(6, \theta^0) = 0.22$$

$$w(7, \theta^0) = 0.0331$$

$$w(8, \theta^0) = 0.0935$$

$$w(9, \theta^0) = -0.1348$$

$$w(10, \theta^0) = -0.0222$$

取 $\Delta\theta = -0.008$, 则因 $\theta = \theta^0 + \Delta\theta = -0.8 - 0.008 = -0.808$, 可得

$$w(k) = y(k) - 0.808 w(k-1)$$

从而对于 $\theta = \theta^0 + \Delta\theta = -0.808$, 得

$$w(1, \theta^0 + \Delta\theta) = y(1) - 0.808w(0) = 0.2 - 0.808 \times 0 = 0.2$$

$$w(2, \theta^0 + \Delta\theta) = y(2) - 0.808w(1) = 0.4 - 0.808 \times 0.2 = 0.2384$$

$$w(3, \theta^0 + \Delta\theta) = 0.3074$$

$$w(4, \theta^0 + \Delta\theta) = 0.1716$$

$$w(5, \theta^0 + \Delta\theta) = 0.1613$$

$$w(6, \theta^0 + \Delta\theta) = 0.2197$$

$$w(7, \theta^0 + \Delta\theta) = 0.0325$$

$$w(8, \theta^0 + \Delta\theta) = 0.0937$$

$$w(9, \theta^0 + \Delta\theta) = -0.1357$$

$$w(10, \theta^0 + \Delta\theta) = -0.0203$$

$w(k)$ 对 θ 的偏导数可近似表示为

$$\frac{\partial w(k)}{\partial \theta} = \frac{w(k, \theta^0 + \Delta\theta) - w(k, \theta^0)}{\Delta\theta}$$

因而, 在 θ^0 点的偏导数为

$$\frac{\partial w(1)}{\partial \theta} = \frac{w(1, \theta^0 + \Delta\theta) - w(1, \theta^0)}{\Delta\theta} = \frac{0.2 - 0.2}{-0.008} = 0$$

$$\frac{\partial w(2)}{\partial \theta} = \frac{w(2, \theta^0 + \Delta\theta) - w(2, \theta^0)}{\Delta\theta} = \frac{0.2384 - 0.24}{-0.008} = 0.2$$

类似可得

$$\frac{\partial w(3)}{\partial \theta} = 0.075, \quad \frac{\partial w(4)}{\partial \theta} = 0.025, \quad \frac{\partial w(5)}{\partial \theta} = -0.0241, \quad \frac{\partial w(6)}{\partial \theta} = 0.1806$$

$$\frac{\partial w(7)}{\partial \theta} = 0.0752, \quad \frac{\partial w(8)}{\partial \theta} = -0.0276, \quad \frac{\partial w(9)}{\partial \theta} = 0.1158, \quad \frac{\partial w(10)}{\partial \theta} = -0.2375$$

用泰勒级数展开法在 θ^0 处将模型线性化, 并且只保留一阶项, 结果为

$$w(k) = w(k, \theta^0) + \left. \frac{\partial w(k)}{\partial \theta} \right|_{\theta=\theta^0} (\theta - \theta^0) + \varepsilon(k)$$

其中, $\left. \frac{\partial w(k)}{\partial \theta} \right|_{\theta=\theta^0}$ 表示 $w(k)$ 对 θ 的偏导数在 $\theta = \theta^0$ 处的值, $\varepsilon(k)$ 表示被略去的高阶导数项。

由于 $w(k)$ 是零均值的白噪声序列, 上式可以简写为

$$-w(k, \theta^0) = \left. \frac{\partial w(k)}{\partial \theta} \right|_{\theta=\theta^0} (\theta - \theta^0) + \varepsilon^*(k)$$

其中, $\varepsilon^*(k) = -w(k) + \varepsilon(k)$ 。

令

$$\mathbf{y} = \begin{bmatrix} -w(1, \theta^0) \\ -w(2, \theta^0) \\ -w(3, \theta^0) \\ -w(4, \theta^0) \\ -w(5, \theta^0) \\ -w(6, \theta^0) \\ -w(7, \theta^0) \\ -w(8, \theta^0) \\ -w(9, \theta^0) \\ -w(10, \theta^0) \end{bmatrix} = \begin{bmatrix} -0.2000 \\ -0.2400 \\ -0.3080 \\ -0.1736 \\ -0.1611 \\ -0.2200 \\ -0.0331 \\ -0.0935 \\ 0.1348 \\ 0.0222 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} \frac{\partial w(1)}{\partial \theta} \\ \frac{\partial w(2)}{\partial \theta} \\ \frac{\partial w(3)}{\partial \theta} \\ \frac{\partial w(4)}{\partial \theta} \\ \frac{\partial w(5)}{\partial \theta} \\ \frac{\partial w(6)}{\partial \theta} \\ \frac{\partial w(7)}{\partial \theta} \\ \frac{\partial w(8)}{\partial \theta} \\ \frac{\partial w(9)}{\partial \theta} \\ \frac{\partial w(10)}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.2000 \\ 0.0750 \\ 0.2500 \\ -0.0241 \\ 0.1806 \\ 0.0752 \\ -0.0276 \\ 0.1158 \\ -0.2375 \end{bmatrix}$$

则用最小二乘法得到

$$\theta - \theta^0 = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = -0.0296$$

故

$$\theta = \theta^0 - 0.0296 = -0.8 - 0.0296 = -0.8296$$

再以 θ 为新的初始值进行迭代,直到收敛为止。

10.4 确定性的趋向和季节性:非平稳序列

前面所讨论的建模方法,是对平稳序列给出的,即基于这样的假定:序列均值和(自)相关函数与时间原点无关。在一些实际问题中,如果知道系统含有确定性分量或得到的动态数据具有明显的确定性趋向和周期性,这时时间序列的统计特性表现为与时间原点有关,则称此序列为非平稳序列。对这样的数据如何建模呢?一个很自然的做法是通过差分或变换将非平稳序列变为平稳的,然后按平稳序列的方法建模(Box 和 Jenkins, 1970)。但是,用此方法建模的一个很大缺点是:它完全掩盖了数据本来存在趋向,而以一个不相干的模型来表示它们,这就完全失去了数据提供的极为重要的信息及其物理解释,这当然是不希望的,因此,建议用另外的方法,其基本思想如下:

把模型分解为两部分——确定性部分和随机性部分。确定性部分表示序列的均值,是用与时间原点有关的确定性函数(如多项式、指数或正弦)来表示序列的非平稳趋向;随机性部分表示序列的均值为零的平稳随机成分。首先在误差是不相关的假定下,单独拟合确定性模型,并对所得到的残差序列进行分析检验,如果残差序列没有非平稳趋向,则对残差序列(平稳序列)拟合 ARMA 模型,这个模型粗略地表示随机模型。然后把确定性的和随机性的模型合并

在一起构成完整模型,并且将分别得到的确定性和随机性模型参数的估计值作为完整模型参数的初始值,再拟合数据,求出完整模型参数的最终估计值,从而得到非平稳时间序列的适用模型。可以证明,很多确定性趋向都可由线性趋向、指数趋向和周期趋向,以及它们的叠加来表示。下面仅以化学反应的建模为例来说明非平稳序列的建模步骤。

【例 10.4】 用一个温度阶跃系统来研究化学反应中浓度变化的衰减过程

在一个容器中,电火花突然使温度升高,破坏了溶液浓度的平衡状态,经过一段时间后,浓度又逐渐恢复到平衡状态。在此过程中,对反应溶液的浓度变化进行量测,采样间隔为0.02s,共记录了100个数据,记录结果如图10.3所示。

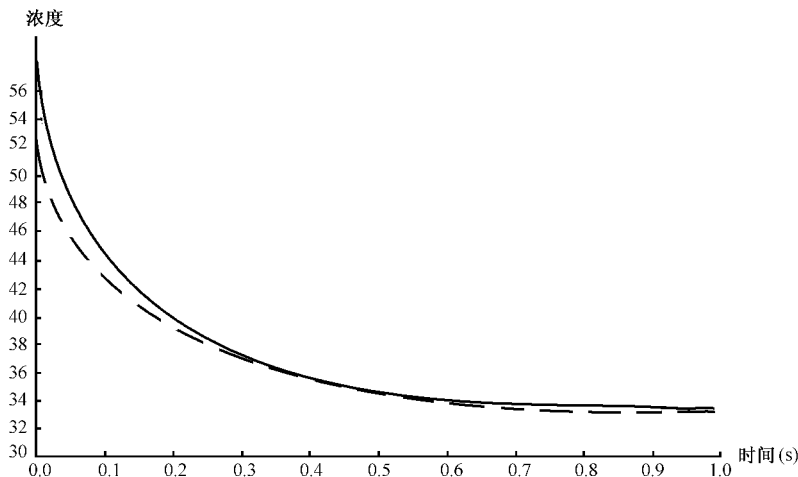


图 10.3 化学衰减数据

从图10.3可以看出,数据有明显的指数趋向,首先用一个指数项来拟合确定性部分,模型为

$$y(t) = a_0 + a_1 e^{-\frac{t}{\tau}} + \varepsilon(t) \quad (10.40)$$

其中, $(a_0 + a_1)$ 表示初始浓度; a_0 表示最后平衡状态的浓度; τ 为衰减时间常数。

由于式(10.40)是非线性的,所以也需根据数据先求出参数的初始值。由图10.3可以初步确定: $a_0 = 32$; $a_0 + a_1 = 56$, $a_1 = 24$ 。

当 $t = \tau$ 时, $y(\tau) = a_0 + a_1 e^{-1} = 32 + 24 \times 0.37 \approx 41$, 从图10.3上可以查出,当 $y(\tau) = 41$ 时, $t = 0.25s = \tau$, 以这些值作为 a_0, a_1, τ 的初始值,用非线性最小二乘法得出以下结果

$$\hat{a}_0 = 31.343 \pm 0.278$$

$$\hat{a}_1 = 22.575 \pm 0.329$$

$$\hat{\tau}_0 = 0.3226 \pm 0.0134$$

残差平方和为:58.05。拟合的结果在图10.3中以实线表示。残差 $\varepsilon(k)$ 的自相关函数在图10.4中示出。为了检查残差的平稳性,除了从残差图形分析外,还需研究残差的样本自相关函数,由图10.4可见,样本自相关函数是逐渐衰减的,这说明残差序列是平稳的。所以对残差序列(随机部分)拟合 ARMA 模型,拟合的结果是一个 AR(1)模型,即

$$x(t) = \phi_1 x(t-1) + w(t) \quad (10.41)$$

式中, $\phi_1 = 0.60$ 。

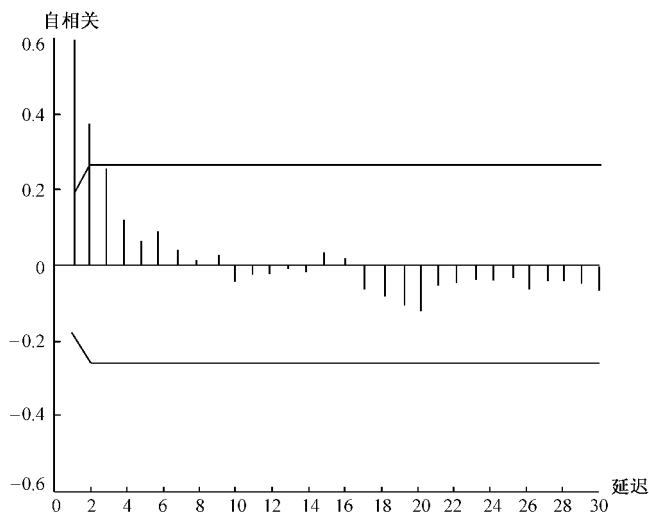


图 10.4 确定性模型的残差序列自相关图形

完整模型为

$$y(t) = a_0 + a_1 e^{-\frac{t}{\tau}} + x(t) \quad (10.42)$$

$$x(t) = \phi_1 x(t-1) + w(t)$$

将分离模型求得的参数值作为完整模型参数的初始值,并用非线性最小二乘法得到式(10.42)中参数的最终估计值为

$$\hat{a}_0 = 30.545 \pm 0.671$$

$$\hat{a}_1 = 21.686 \pm 0.769$$

$$\hat{\tau} = 0.3789 \pm 0.379$$

$$\hat{\phi}_1 = 0.6172 \pm 0.0689$$

残差平方和为 32.49。

最后,再检查残差 $w(t)$ 的相关性。 $w(t)$ 的自相关函数示于图 10.5 中。由图可见,自相关函数值都不超过 $\frac{2}{\sqrt{100}} = 0.2$, 因此可以认为 $w(t)$ 为白噪声,所以模型式(10.42)是适用的。完整模型拟合结果在图 10.43 中由虚线示出。

对于其有线性趋向的随机序列,可用与上例完全类似的步骤建模。但对于具有季节性趋向的序列,情况比较复杂,尤其在经济、商业、市场价格等领域中,观测得到的数据往往除了基本周期(如年、月、周等)外,还有谐波分量。因此,若使残差变为平稳,往往需要逐次对残差图形和自相关函数的图形进行分析判断,消去几种趋向才有可能。由于篇幅所限,这里不再详述,参见吴贤铭教授(Wu S M and S M Pandit, 1980)和 Montgomery D 等人(1990)的文章。

值得注意的是,具有周期性变化的数据,如果在各数据上周期也有变化(称之为随机周期性),对这样的序列仍可以认为是平稳的,从而仍可用 10.3 节给出的建模步骤,若用非平稳序列的方法建模,则不能得到满意结果。只有当数据的周期趋向有规律地重复出现,必须作为确定性趋向时,才用非平稳序列的建模方法。

最后指出,本章所介绍的时间序列建模方法,只是一种最基本的方法,对不同的问题、不同

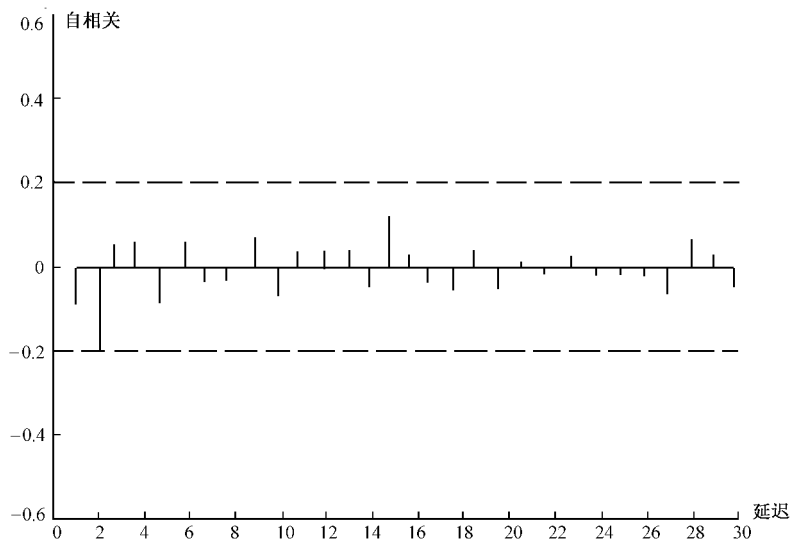


图 10.5 完整模型的残差序列自相关图形

的要求,可以有许多的方法。就本质而言,时间序列预测模型的基本依据是事物发展本身存在一个延续发展过程,因此,模型的一般形式(对单变量)可以写为

$$x(t) = f(x(t-1), x(t-2), \dots, x(t-n), t) + w(t)$$

如果形式上将 $x(t)$ 作为输出,将 $x(t-1), x(t-2), \dots, x(t-n)$ 作为输入,则也可以用前面几章所介绍的有关辨识的方法来建模。对非线性系统,神经网络模型是采用较多的模型之一(Plummer E a, 2000)。

另外,最近的研究表明:混沌(Chaos)是非线性系统中普遍存在的现象,在经济、金融系统中普遍存在着混沌现象(Benhabib J, 1992)。因此,混沌时间序列建模方法的研究也是时间序列建模研究的一个重要领域,已有许多文章和书籍可以参考。

习 题

1. 以 AR(1)模型为例,解释格林函数的物理意义。
2. 以表 10.1 给出的数据,分别以逆函数法和两次最小二乘法求出 ARMA(2, 1)模型参数的初始值(编程计算实现)。
3. 用 5.2 节给出的递推算法,给出实现 ARMA(n, m)模型寻求初估模型的具体步骤,并用计算机实现对例 10.2 的计算。
4. 编程实现用高斯法求解非线性最小二乘估计,并具体计算例 10.3。
5. 编程实现对例 10.4 计算。

第 11 章 神经网络模型

11.1 引言

20 世纪 80 年代,人工神经网络(Artificial Neural Networks, ANN)的研究取得了重大的进展,有关的理论和方法已经使其发展成为一门介于物理学、数学、计算机科学和神经生物学之间的交叉学科;它在诸如过程控制、模式识别、函数逼近、动态建模、数据挖掘,以及模拟某些生物体神经活动的过程等领域,在具有重大理论及实际意义的许多问题上都取得了成功,近几年来将人工神经网络用于系统辨识和控制中的研究成了热门话题。有些学者甚至预言,在 21 世纪的科学发展中,人工神经网络将占有更重要的地位。

本章作为讨论数学模型的一部分内容,简单地介绍如何利用神经模型进行辨识和建模问题;同时,通过这一介绍,也试图使读者对于人工神经网络作为人脑活动的数学模型有所了解。

在叙述本章主要内容之前,有必要强调以下几点:

(1) 术语“神经网络模型”或“人工神经网络”,显然来自对人脑的模拟,然而就本章讨论的内容而言,实际上与人脑的高级神经活动还有相当大的距离。

(2) 本章所要介绍的,仅限于建模中最常用的神经网络模型的数学描述,整个过程可以在计算机上实现,完全不涉及人工神经网络的硬件实现途径。

(3) 本章仅仅是一个初步介绍,“初步”两字的含义是多方面的:它不仅意味着从最基本的内容、最基本的结果、最基本的方法开始,也表明叙述的范围是很有限的;即使对辨识问题,也远未涵盖全部的内容。此外,文中完全不涉及问题的理论方面。更全面的叙述请参考文献(胡学仁,1993;焦李成,1993;Hertz J A 等,1988)

11.2 神经组织的基本特征和人工神经元

如前所述,人工神经网络来源于对人脑实际神经网络的模拟,它是由若干层、每层有若干个神经细胞、按某种方式相互连接构成的网络,从神经生理学的观点来看,这一模拟是极端简化的。但是,尽管这些模型忽略了单个神经细胞的许多细节,但就整体而言,的确已经捕捉到了生物神经活动的某些基本特点。为了说明这一点,先简略地考察一下生物神经组织的结构和功能。

11.2.1 神经组织的基本特征

如图 11.1 所示是一个神经细胞的示意图。为了行文方便,下文有时又称神经细胞为神经元。细胞核所在部分为细胞体,从细胞体树状延伸出许多神经纤维,其中最长的一条称为轴突,它的末端化为许多细小的分枝,称为神经末梢;从细胞体出发的其他树状分枝称为树突。一个神经细胞通过轴突与其他细胞的树突相连,神经末梢与树突接触的界面称为突触。

就功能而言,细胞体可以视为一个基本的初等信号处理器,轴突是信号的输出通路,树突

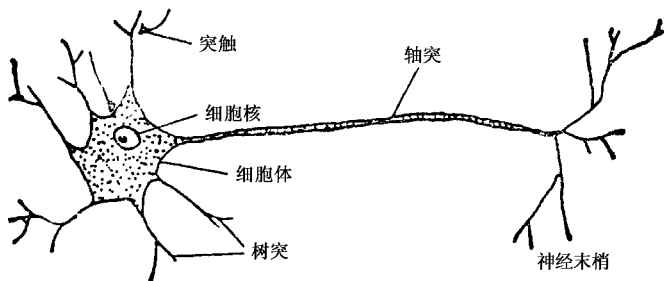


图 11.1 神经细胞示意图

是信号的输入通路。信号从一个神经细胞经过突触传递到另一个细胞,是一个相当复杂的生物化学过程,它可以产生两种不同的效果:或者使接收信号一方细胞体内的电位升高,或者使之降低。当细胞体内电位达到某一阈值时,则接受信号的细胞被激发,它也会通过轴突,传出一个有固定强度和持续时间的脉冲信号,此时称该细胞处于激发态;当细胞体内电位低于阈值时,则不产生任何电位输出,细胞处于抑制状态。由处于激发态的神经细胞所产生的脉冲信号,通过神经末梢,传向处于下游的每个与之相连的神经元,但是对不同的下游神经元,信号所引起的电位变化是不同的。这一点可以表述为:不同的神经元之间有不同的作用强度,或称连接强度。

神经组织还有一个重要特点,即神经元之间的信号传输的效能不是一成不变的。如果神经元 A 不断地向神经元 B 传送信号, B 在接到信号后又不断地被激发,那么由 A 发出的同样强度的信号,对 B 电位变化的影响不断加强,或者说, A 和 B 之间的连接强度将随时间变化而增加。这一性质进一步可以表述为:两种神经元之间的连接强度,随其激发与抑制行为相关性的时间平均值正比变化。这就是生物学上所说的 Hebb 律,它表明神经系统是有某种可塑性的。

11.2.2 人工神经元的 M-P 模型

由上述可以看出,为了构造一个模拟生物神经组织的人工神经网络,必须给出如下 3 方面的要素:① 对单个人工神经元给出某种形式的定义;② 决定网络的层数、各层中的神经元的数量,以及彼此间的连接方式,或者说,定义网络的结构;③ 给出一种方法,决定神经元与神经元之间的连接强度,使网络具有某种预定的功能。

下面就来介绍由 McCulloch 和 Pitts 在 1943 年所提出的历史上第一个人工神经元模型,或称之为 M-P 模型。这一模型形式上表示为

$$S_i(t+1) = f\left(\sum_{j=1}^n w_{ij} S_j(t) - \mu_i\right)$$

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{其他} \end{cases} \quad (11.1)$$

其中, t 表示时间,它只取离散值,两个相继时刻的间隔均为一个时间单位。 $S_i(t)$ 表示第 i 个神经元在 t 时刻的状态; $S_i(t) = 1$,表示处于激发态; $S_i(t) = 0$,表示处于抑制态。容易看出,由于 $f(x)$ 的函数形式,每个神经元只能有两个状态,因此, M-P 人工神经元模型实际上代表了一个双稳态元件。 w_{ij} 是一个实数,刻画第 j 个神经元到第 i 个神经元的连接强度,称之为权,其值可正可负,分别表示神经元 j 对神经元 i 的作用是激发还是抑制。特别规定 $w_{ii} = 0$,即神经元对自身没有作用。 $\sum_{j=1}^n w_{ij} S_j(t)$ 表示第 i 个神经元在 t 时刻所接收到的所有信号的线性叠

加, μ_i 是神经元 i 的阈值, 当神经元 i 的输入信号的加权和超过 μ_i 时, 神经元 i 才被激发。应当指出, 在这一模型中, μ_i 这一项是非本质的, 因为可以考虑在模型中增加一个永远处于激发态, 即永远取值为 1 的神经元, 将 $-\mu_i$ 取为该神经元到神经元 i 的权, 那么阈值项便可归并到和号中去了。如图 11.2 给出了 M-P 模型的一个简单图示。

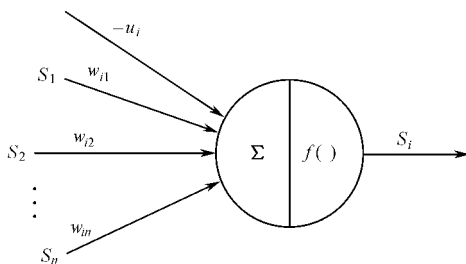


图 11.2 M-P 模型示意图

M-P 神经元是最简单的神经元模型。实际的生物神经元远比它复杂得多。但是, 重要之点在于, M-P 模型已经捕捉到了神经细胞的一个最基本特征, 即输入与输出之间的非线性关系, 这当然不是指其具体的函数形式。在很多情况下, 尤其在对非线性系统建立神经网络模型时, 由于实际过程变化常常是连续的, 因此, 考虑具有连续输出的神经元模型不仅更符合实际, 描述范围更大, 而且在具体计算时也会带来很大方便。对 M-P 模型, 这方面最常用的一个推广是令 S_i 的状态可取 $[0, 1]$ 区间的连续值, 以常规的 S 形函数

$$g(x) = [1 + \exp(-2\beta x)]^{-1}$$

$$x = \sum_{j=1}^n w_{ij} S_j(t), \quad 0 < g(x) < 1 \quad (11.2)$$

代替阶跃函数 $f(x)$, 其中 β 是一个正参数。可以看出, β 越大, $g(x)$ 的形式越接近 $f(x)$, 它们之间的差别在于 $g(x)$ 可在区间 $(0, 1)$ 中连续取值且是可微的。S 形函数在实际中得到广泛的应用, 但因为它的输出均为正值, 对于描述实际系统受到极大的限制, 因此实际中也常用双曲正切函数 $h(x)$, 来取代 S 形函数, 双曲正切函数的输出值可为正或负, 即

$$h(x) = \frac{1 - e^{-\beta x}}{1 + e^{-\beta x}}, \quad -1 < h(x) < 1 \quad (11.3)$$

式中 β 是一个正参数。

二值函数 $f(x)$ 、S 形函数 $g(x)$ 和双曲正切函数 $h(x)$ 的图形分别如图 11.3(a), (b), (c) 所示。

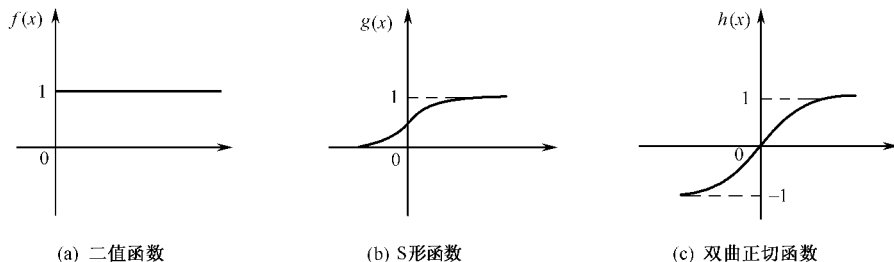


图 11.3 神经元中 3 种常用的变换(激发)函数

以上给出了单个神经元的形式定义。如前所述,要建立一个神经网络模型,除了定义人工神经元外,还必须确定网络的结构和神经元与神经元之间的连接强度值,即权值。

11.3 多层前馈神经网络模型

上一节给出了单个神经元的模型,若建立神经网络模型,还需确定网络的结构和参数。网络模型的结构包括:① 网络的层数和每层神经元的数目;② 神经元之间的连接方式;③ 非线性神经元的非线性函数。网络中的参数即是神经元之间连接的权值和神经元的阈值。

11.3.1 前馈神经网络模型的结构

考虑多输入、多输出的非线性系统,如图 11.4 所示。

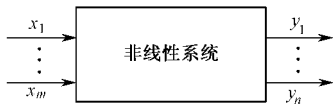


图 11.4 多变量非线性系统示意图

可以形式地表示为

$$y = f(x) \tag{11.4}$$

其中, $y = (y_1, y_2, \dots, y_n)^T$ 为输出向量, $x = (x_1, x_2, \dots, x_m)^T$ 为输入向量, f 为未知非线性函数。根据 Cyhenko G (1989) 和 K Funahashi (1989) 的研究结果:只要网络结构充分大(这里指隐含神经元的数目充分大),通过精心选择网络参数,可以以任意精度逼近任意连续函数(这个结论对只含一个隐含层的神经网络仍正确)。因此,模型式(11.4)可由如图 11.5 所示的 3 层前馈神经网络(Feed forward Neural Networks, FNN)来逼近。

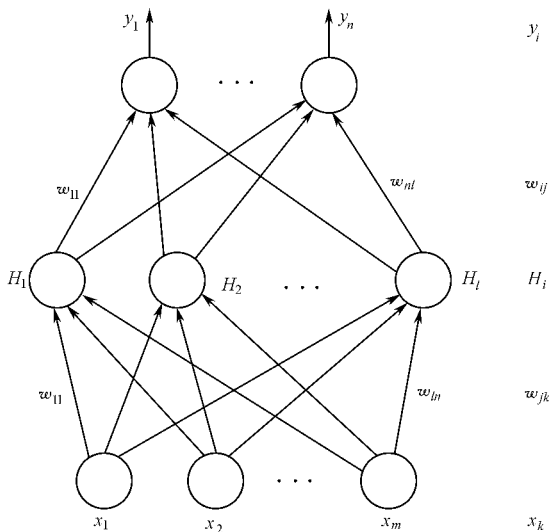


图 11.5 多层前馈神经网络示意图

式中,神经元状态取 $(0,1)$ (或 $(-1,1)$) 中的连续值,由所选的 S 形函数(或双曲正切函数)所决

定。图 11.5 中最下面的一层,称为输入层,用于输入已知量测值;中间一层称为处理层或隐含层(隐单元层);最上面一层称为输出层,用于输出与每一组输入数据相对应的输出信息。任何一个中间层神经元(或简称为单元)接受所有输入单元传来的信号,并把处理后的结果传向每一个输出单元,供输出层再次加工。同层的神经元彼此不连接,输入与输出单元之间也没有直接连接。这种结构通常称为 3 层前馈神经网络。也有些文献将这样的网络称为两层前馈网络,其理由是只有中间层及输出层的单元才对信号进行处理,输入层的单元对输入数据没有任何加工,故不计算在层数之内。但本节中仍称为其 3 层神经网络。

从上面网络模型的结构框架不难看出,输入层和输出层的单元数完全由问题本身确定。确定网络的结构问题简化为确定隐含单元数目的问题。关于如何适当选取隐含单元数目的问题,有一些文献进行了探讨,但实际应用中,通过经验或试验来决定或许是最好的途径。试验的方法常用的有两种思路:一种是从适当小的单元数开始,逐个增加隐含单元,直到再增加单元对网络的精度无显著影响为止;另一种思路是与之相反,先取一个充分大的隐含单元数,逐次删除连接权值最小的单元,直到删除该单元明显影响网络模型的精度为止。这种增或删过程的每一步也都需要确定连接权值,这样才能得到一个网络模型,才能进一步考察模型的精度,才能判断该单元对模型精度的影响,只不过隐含单元的数目是事先主观确定的。

由以上可以看出,无论用何种方法确定网络模型的结构,或者结构已确定,建立网络模型的一个关键问题,就是确定网络模型的参数,即确定连接权值(阈值也作为连接权)。

11.3.2 确定网络模型权值问题的数学描述

确定网络模型权值问题,其基本思路与其他类型模型的参数估计一样,首先取得系统的一组实测输入、输出数据,在建立神经网络模型中称此为训练样本,对应于已知输入的输出称为理想输出;再给出一种确定网络模型参数的方法,即确定连接权值,使得模型的输出与理想输出的偏差达到极小。

为了使问题表达清楚,首先考察一下信息在神经网络模型中是如何传播的。

假设取得了 N 个样本 $\{y(t), x(t); t=1, \dots, N\}$, 其中 y 是 n 维向量, x 是 m 维向量。当第 t 个样本 $x(t) = (x_1(t), x_2(t), \dots, x_m(t))$ 的原始数据输入网络时,网络相应地输出记为 $\hat{y}(t) = (\hat{y}_1(t), \hat{y}_2(t), \dots, \hat{y}_n(t))$; 隐含单元的状态记为 $H_j(t)$ ($j = 1, \dots, l$); 输入单元取值为 $x_k(t)$ ($k = 1, \dots, m$)。请注意,为了叙述清楚,这里约定下标 i, j, k 依次对应于输出层、中间层和输入层。在这一约定下,从中间层到输出层的权记为 w_{ij} , 即表示中间层第 j 个单元向输出层第 i 单元的连接权; 从输入层到中间层的权记为 w_{jk} , 表示输入层第 k 个单元向中间层第 j 个单元的连接权。以上说明已在图 11.5 中标出。如果 w_{ij}, w_{jk} 均已给定,那末对任何一组确定的输入 (x_1, \dots, x_m) , 网络中所有单元的取值都不难确定。事实上,对第 t 个输入样本,隐含单元 j 的输入是

$$h_j(t) = \sum_k w_{jk} x_k(t) \quad (11.5)$$

相应的输出状态是

$$H_j(t) = g(h_j(t)) = g\left(\sum_k w_{jk} x_k(t)\right) \quad (11.6)$$

输出单元 i 所接收到的叠加信号是

$$h_i(t) = \sum_j w_{ij} H_j(t) = \sum_j w_{ij} g\left(\sum_k w_{jk} x_k(t)\right) \quad (11.7)$$

网络的最终(单元 i) 输出是

$$\hat{y}_i(t) = g(h_i(t)) = g\left(\sum_j w_{ij} H_j(t)\right) = g\left(\sum_j w_{ij} g\left(\sum_k w_{jk} x_k(t)\right)\right) \quad (11.8)$$

注意,这里没有考虑阈值,正如前面已经说明的那样,这一点是无关紧要的。还应当指出的是,对任何一组确定的输入,输出是所有权 $\{w_{ij}, w_{jk}\}$ 的函数。确定权值的目标是确定适当的权 $w = \{w_{ij}, w_{jk}\}$,使得网络的输出与系统的实际输出(理想输出)误差最小。这归结为一个最优化问题,即确定适当的权 w ,使

$$E(w) = \frac{1}{2} \sum_{i,t} [y_i(t) - \hat{y}_i(t)]^2 = \frac{1}{2} \sum_{i,t} [y_i(t) - g\left(\sum_j w_{ij} g\left(\sum_k w_{jk} x_k(t)\right)\right)]^2 \quad (11.9)$$

达到极小。

显然,对每一个变量 w_{ij} 或 w_{jk} 而言, $E(w)$ 是一个连续可微的非线性函数。对于多层前馈网络,如何求得一组恰当的权值,使网络具有特定的功能,在很长一段时间内曾是使研究者感到困难的问题。直到 1985 年,美国加州大学的一个研究小组提出了所谓的向后传播算法,称为 BP(Back Propagation)算法,才使问题有了重大进展,从而促使人工神经网络的研究得以迅猛发展,下面介绍这一算法。

11.3.3 BP 算法

BP 算法是一种梯度法,也称为最速下降法,这是一种迭代算法,基本步骤如在 6.1 节中所述。从一个初始点 w_0 出发,计算在 w_0 点的负梯度方向 $-\nabla E(w_0)$,只要 $\nabla E(w_0) \neq 0$,就可沿该方向移动一小段距离,达到一个新的点 $w_1 = w_0 - \eta \nabla E(w_0)$, η 是一个正参数,只要 η 足够小,定能保证 $E(w_1) < E(w_0)$ 。不断重复这一过程,一定能达到 $E(w)$ 的一个极小点。就本质而言,这就是 BP 算法的全部内容。然而对人工神经网络而言,这一算法的具体形式是非常重要的,下面给出这一具体的表达形式。

对于隐单元到输出单元的权 w_{ij} 而言,最速下降法给出的每一步的修正量是

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \sum_t [y_i(t) - \hat{y}_i(t)] \dot{g}(h_i(t)) H_j(t) = \eta \sum_t \delta_i(t) H_j(t) \quad (11.10)$$

这里令

$$\delta_i(t) = \dot{g}(h_i(t)) [y_i(t) - \hat{y}_i(t)] \quad (11.11)$$

对输入单元到隐含单元的权

$$\begin{aligned} \Delta w_{jk} &= -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial H_j(t)} \frac{\partial H_j(t)}{\partial w_{jk}} \\ &= \eta \sum_{t,i} [y_i(t) - \hat{y}_i(t)] \dot{g}(h_i(t)) w_{ij} \dot{g}(h_j(t)) x_k(t) \\ &= \eta \sum_{t,i} \delta_i(t) w_{ij} \dot{g}(h_j(t)) x_k(t) \\ &= \eta \sum_t \delta_j(t) x_k(t) \end{aligned} \quad (11.12)$$

式中

$$\delta_j(t) = \dot{g}(h_j(t)) \sum_i \delta_i(t) w_{ij} \quad (11.13)$$

由式(11.10)和式(11.12)可以看出,所有权修正量都有如下形式

$$\Delta w_{pq} = \eta \sum_t \delta_p(t) v_q(t) \quad (11.14)$$

其中,指标 p 对应于两个单元中输出信号的一端, q 对应于输入信号的一端, v 或者代表 H 或者代表 x 。从形式上来看,这一修正是“局部”的,可以看做是 Hebb 律的一种表现形式。还应注意, $\delta_i(t)$ 由实际输出与理想输出的差及 $h_i(t)$ 决定,而 $\delta_j(t)$ 则需依赖 $\delta_i(t)$ 算出,因此,这一算法才称为向后传播算法。稍加分析还可知道,利用由式(11.10)~式(11.13)所给出的计算安排,较之不考虑 $\delta_p(t)$ 的向后传播,直接计算所有含 g' 的原表达式(11.12),极大地降低了计算量。这组关系式称为广义 δ 法则,它们不难推广到一般的多层网络上去。

利用这一迭代算法,最终生成在一定精度内满足要求的 $\{w_{ij}, w_{jk}\}$ 的过程,称为人工神经网络的学习(或训练)过程。可以看出,这里所提供的学习机制是元与元之间权的不断调整,学习样本中任何一个样本所提供的信息,最终将包含在网络的每一个权中。参数 η 的大小则反映了学习效率。

为了更有效地应用 BP 算法,进行如下补充说明:

① 在式(11.10)与式(11.13)中, Δw_{jk} 表示为与所有样本 t 有关的求和计算。实际上,还可以每次只考虑输入一个样本所造成的修正,然后按照随机选取的顺序,将所有样本逐个输入,不断重复这一过程,直至收敛到一个满意的解为止。

② 在如上的算法中,利用实际输出与理想输出差的平方和作为度量 $\{w_{ij}, w_{jk}\}$ 优劣的标准,这并不是惟一的度量方式,完全可以从其他的函数形式出发,导出相应的算法。

③ 在如上的讨论中使用的是最速下降法,可采用其他的非线性优化方法,如在第 8 章中给出的方法,以及将在第 13 章介绍的遗传算法也可以用于权值训练。为了加速算法的收敛速度,还可以考虑各种不同的修正方式和不同算法的结合。

④ BP 算法的出现,虽然对人工神经网络的发展起了重要推动作用,但是这一算法仍有很多问题。对于一个大的网络系统,BP 算法的工作量仍然是十分可观的,这主要在于算法的收敛速度很慢,这一速度与 $g(x)$ 的形式、参数 β, η 及初值的选取均有关系。更为值得注意的是,这里所讨论的既然是非线性函数的优化,那么它就无法逃脱该类问题的共同困难:BP 算法所求的解,只能保证是依赖于初值选取的局部极小点。为克服这一缺陷,可以考虑改进方法,如模拟退火算法、遗传算法等(将在第 13 章中介绍),或从多个随机选定的初值点出发,进行多次计算,但这些方法都不可避免地加大了工作量。

⑤ 为了提高学习算法的收敛速度,可将递推最小二乘的思想引入,形成 RLS 的学习算法(谭永红,1994),这可提高神经网络的学习速度。

11.3.4 神经网络的几个有关概念

1. 监督下学习与无监督学习

把利用 BP 算法确定连接强度即确定权值的过程称为“学习过程”。这种学习的特点是,对任何一个输入样本,其相应地理想输出是已知的,因而从网络模型所产生的输出与理想输出之间的差异,我们可以清楚地知道网络正确与否,故此把这一类学习称为在教师监督下的学习,对实际系统建立神经网络模型多采用此种方式;与此不同的是,有些情况下学习是无监督的,例如,试图把一组样品按其本身特点分类,所要划分的类别是事先未知的,需要网络自身通过学习来决定。换句话说,用于学习的训练数据集中,只有输入而没有目标输出,在训练过程中,神经网络自动地将输入数据的特征提取出来,并将其分成若干类。经过训练好的网络能够识别训练数据集以外的新的数据类别,并获得相应输出。在这样的学习过程中,对每一输入所

产生的输出也就无所谓对错,因此 BP 算法是不适用的。关于无监督学习的情况可参考专门介绍神经网络的书籍(如焦李成,1993)。

2. 有竞争的学习

在讨论用于分类、故障诊断等问题的网络模型中,尽管我们所希望的理想输出是只有一个输出单元为 1,其他全为 0,但实际输出并不如此。如果我们设想另一种输出模式:对应任何一组输入,在所有输出单元中,只允许有一个处于激发态,即取值为 1,其他输出单元均被抑制,即取值为 0。一种形象的说法是,对应任何一组输入,要求所有的输出单元彼此竞争,惟一的胜利者赢得一切,失败者一无所获。形成这样一种输出机制的网络学习过程,称为有竞争的学习。这些内容可参考 Hertz J A 等(1991)和 Thomas F 等(1989)的文章。

11.4 神经网络在辨识中的应用

由于人工神经网络具有很强的学习、自适应、自组织、函数逼近和大规模并行处理等能力,因而在系统辨识、模式识别、信号处理、优化和控制中得到广泛的应用。尤其是对高度非线性系统的建模、预测和控制方面,作为一种新的、令人感兴趣的工具起了非常重要的作用。下面给出几个对实际系统建立人工神经网络模型的实例。

【例 11.1】 缩聚反应器的建模(徐耀玲、戴汝为,1991)

如图 11.6 所示是某石化总公司涤纶厂的间歇式缩聚反应示意图。反应完成与否是根据反应釜中物料是否达到要求的特性黏度值来判断的。目前,国内外大部分缩聚反应器没有直接测量黏度的仪表,而是根据搅拌电机的功率来间接推测黏度值。在理想的情况下,即环境(电压、电流、电网频率等)、工艺参数(投料量、温度、转数等)稳定时,搅拌功率和黏度值有固定的对应关系。但实际环境和工艺参数往往会有变化,这就给根据功率推测黏度带来了困难和误差。目前工厂是根据经验对功率进行修正,当修正后的功率值达到设定值时,就可以出料。经验的修正公式如下:

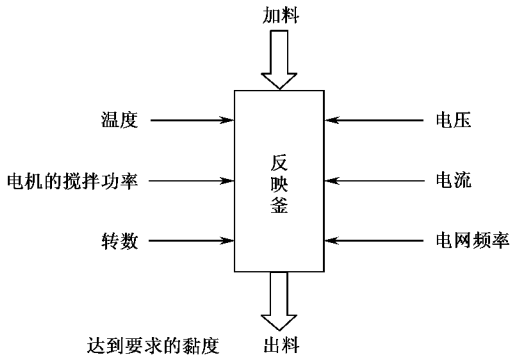


图 11.6 间歇式缩聚反应示意图

- (1) 物料补正: $\pm 30\text{kg}$, 修正: $\pm 0.1\text{kw}$
- (2) 温度补正: $\pm 1^\circ\text{C}$, 修正: $\pm 0.1\text{kw}$
- (3) 黏度补正: ± 0.002 , 修正: $\pm 0.1\text{kw}$
- (4) 电压补正: $\pm 10\text{V}$, 修正: $\pm 0.4\text{kw}$

根据从现场采集的 49 组数据,用神经网络来建立此对象的数学模型,即寻找电机转数 n 、电网频率 f 、电压 V 、投料量 Q 、电流 I 、温度 T 、黏度 IV 与电机搅拌功率 W 的关系

$$W = g(n, f, V, Q, I, T, IV)$$

为此,构造一个 3 层前馈网络,输入节点、隐含节点、输出节点分别为 7 (即函数 g 中的自变量个数)、5 和 1 (即电机搅拌功率 W),选择 24 组较有代表性的实测数据作为网络学习 (训练) 样本 (考虑到模型的检验,所以这里只用了一半数据作为学习样本),送入网络中学习,得到模型 $P_{24\text{mod}}$,其学习精度为 129W,即模型输出与学习样本输出误差小于 129W。

将 49 组数据全部送入 $P_{24\text{mod}}$ 检验,可知模型的输出与实际测量值拟合程度较好。25 组非学习样本中大多数也能满足模型的误差要求。模型的外延性较好,仅个别点偏差较大。模型输出与实测数据、经验规则与实测数据的比较分别如图 11.7 和图 11.8 所示。

该网络模型在实际中得到了成功的应用。

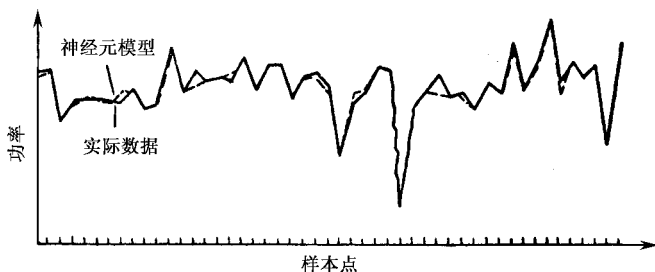


图 11.7 神经网络模型的输出与实测值的比较

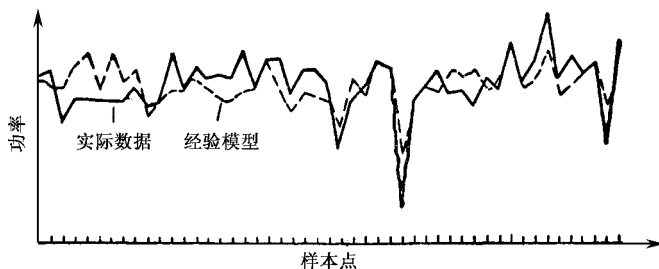


图 11.8 经验模型的输出与实测值的比较

【例 11.2】神经网络在电机建模中的应用

直接转矩控制(DTC)是工业应用交流电机的最简单的转矩控制,有着广泛的应用,DTC 的关键部件是状态选择器。这里考虑的是一个用直接转矩控制的逆变器馈电式交流电机。开关状态选择器的输入是电磁转矩误差(E_{te})、定子磁通误差(E_{ϕ})和磁通角(θ)。其输出是逆相开关状态(u_1, u_2, u_3)。目的是要用神经网络来模拟该开关状态选择器,即对开关状态选择器建立神经网络模型。图 11.9 给出了开关状态选择器及训练神经网络的结构图。

图 11.10 给出了神经网络的结构。其中隐含层中的 5 个神经元和输出层的 3 个神经元均采用 S 形函数。

在图 11.10 中, $r_1 = E_{te}$, $r_2 = E_{\phi}$, $r_3 = \theta$ 。

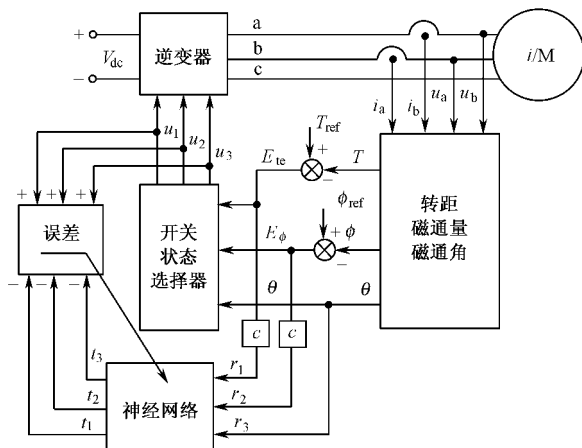


图 11.9 神经网络训练结构示意图

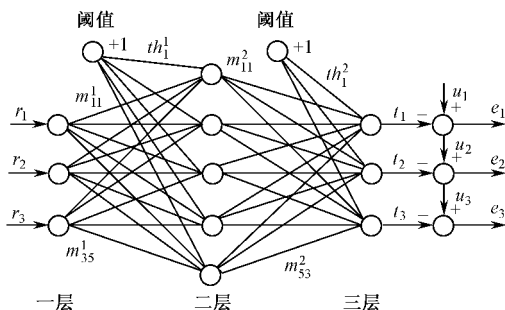


图 11.10 神经网络结构

目标是:寻求一组权值和阈值,使性能指标

$$E(w) = \frac{1}{2} \sum_{i=1}^N e^T(i) \Delta e(i) \quad (11.15)$$

达到极小。其中, $e = (e_1, e_2, e_3)^T$ 是 DTC 的状态选择器输出与神经网络输出之间的误差, Δ 为半正定矩阵, 这里取为单位阵, N 是训练样本的数目 (本例采用 500)。由于 DTC 的状态选择器只依赖于电磁转矩误差、定子磁通误差和磁通角, 而不依赖于交流电机的参数, 所以可以离线训练。这意味着神经网络一旦被训练成功, 则交流电机在任意改变参数的情况下, 神经网络也无需重新训练。

从图 11.10 不难看出, 该神经网络含有 38 个参数 (包括权值和阈值), 采用 BP 算法, 用 500 个样本训练。将训练好的神经网络模型在具有 $12N \cdot m$ 转矩和 90V 定子磁通量的 3HP 交流电机上进行了实验。一般地, 先用具有 20kHz 开关频率的 DTC 实验, 前 0.5s 的转矩、定子电流和转子速度如图 11.11 所示。然后在同一电机上用神经网络代替状态选择器, 得到的结果如图 11.12 所示。

从图中可以看出, 神经网络的实验结果虽不很理想, 但是在工程还是可接受的, 究其原因, 可能是由于 BP 算法没能找到全局最优解, 而是陷于局部最优所致。在第 13 章中, 用遗传算

法训练网络得到了理想的结果,可能在一定程度上验证了此种猜想。

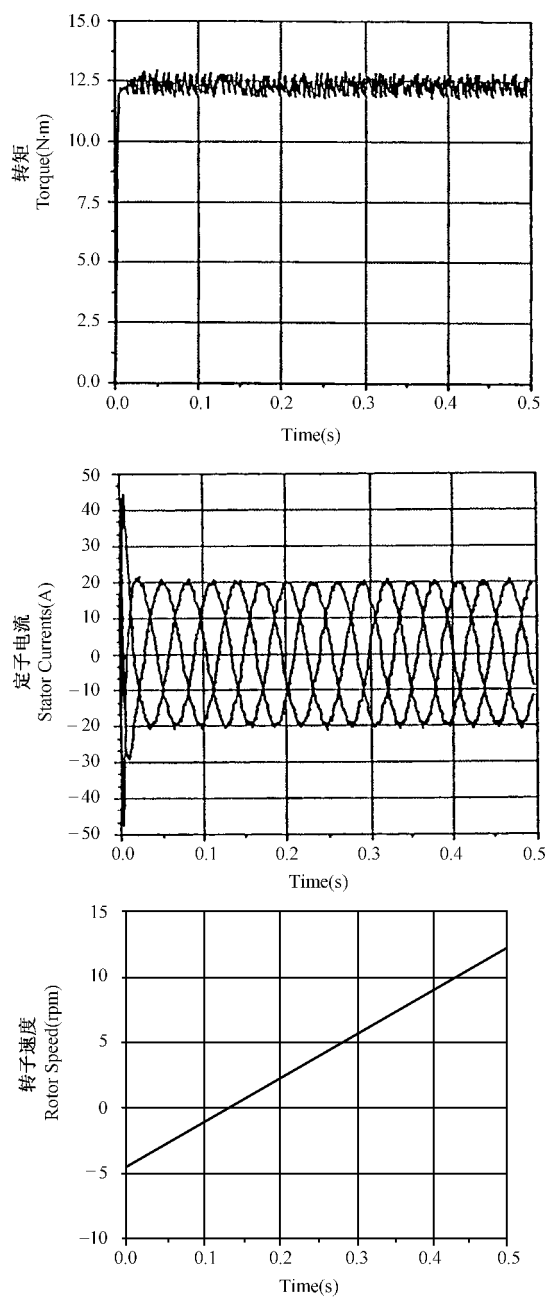


图 11.11 DTC 的转矩、电流和转速的曲线图

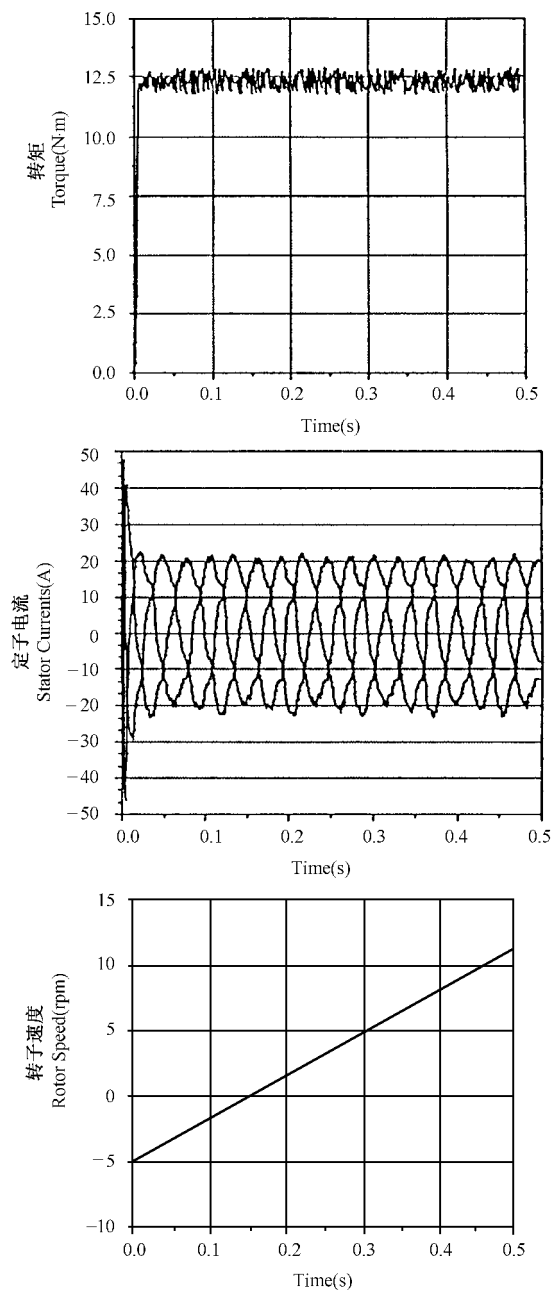


图 11.12 用神经网络的转矩、电流和转速的曲线图

11.5 径向基函数网络及其应用

前馈神经网络以极强的函数逼近能力在信号处理、建模和控制中得到广泛的应用。但它有一个明显的缺点——对参数高度非线性,因此学习的算法必须用非线性最优化技术,而且当采用梯度算法时,常常会使参数估计陷入局部极小。另外,由于计算的复杂性,也不易在线学

习,所得到的模型本质上是非线性静态映射。因此,一般前馈神经网络模型难以应用于时变系统。在大脑皮层、视觉皮层的局部调节和交叠接受区域结构启发下,借鉴生物接受区域的知识,另一种网络——被称为径向基函数网络(Radial Basis Function,RBF)提出了。该网络的函数逼近功能与前馈神经网络相近,但结构简单,对参数是线性的,可以在线学习。因此,在信号处理、建模与预测中得到广泛应用(Powell M J D,1987; Chen S 等,1988)。

11.5.1 径向基函数网络的结构

为了不失一般性,我们考虑 m 个输入、单输出的 RBF 网络(对多输出的 RBF 网络,总可以分解为一组单输出的 RBF 网络),其结构如图 11.13 所示。

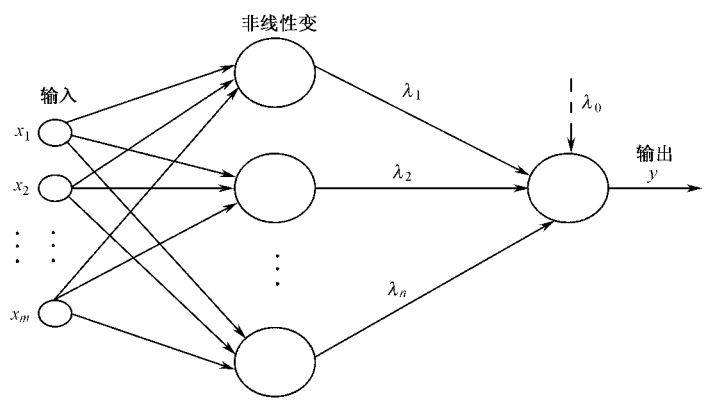


图 11.13 RBF 网络结构

BRF 实现的非线性映射关系为 $f: R^m \rightarrow R$

$$y = f(\mathbf{x}) = \lambda_0 + \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (11.16)$$

式中, $\mathbf{x} \in R^m$ 是输入向量; $\phi(\cdot): R^+ \rightarrow R$ 是给定非线性函数; $\|\cdot\|$ 表示欧几里德(Eudidean)范数, $\lambda_i (0 \leq i \leq n)$ 是权值; $\mathbf{c}_i \in R^m; 1 \leq i \leq n$, 是 RBF 网络的中心点, n 是中心点的个数。

RBF 网络可以看做是固定隐含层节点的中心点和非线性性质、参数线性的特殊神经网络。

隐含层起到非线性转换作用,没有可调参数,而输出层则实现其转换后的线性组合,需要调节的参数仅是线性组合的系数。RBF 网络若要精确地匹配 3 层神经网络,必须精心选择中心点、非线性函数和参数。理论与实践表明,在 RBF 网络中,非线性函数 $\phi(\cdot)$ 的选择对于网络性能的影响并不很重要,重要的是中心点的选择(其个数即是隐含层的单元数),而且只要 RBF 网络的隐含层适当选取,RBF 网络具有和通常神经网络类似的近似特性(Broomhead D S and D Lowe, 1988)。由于其具有很好的函数拟合性能,网络的结构又较简单,且关于参数是线性的,所以 RBF 网络越来越得到广泛的应用。

在 RBF 网络中常用的非线性函数 $\phi(\cdot)$ 有:

薄板样条(thin-plate-spline)函数

$$\phi(\gamma) = \gamma^2 \log(\gamma) \quad (11.17)$$

高斯函数

$$\phi(\gamma) = \exp\left(-\frac{\gamma^2}{\beta^2}\right) \quad (11.18)$$

以及

$$\phi(\gamma) = (\gamma^2 + \beta^2)^{\frac{1}{2}} \quad (11.19)$$

$$\phi(\gamma) = (\gamma^2 + \beta^2)^{-\frac{1}{2}} \quad (11.20)$$

式中, β 是一实常数。

值得指出的是, 式(11.17)和式(11.18)是两种非常典型的选择, 但其表征的非线性却完全不同: 当 $\gamma \rightarrow \infty$ 时, 式(11.17)中 $\phi(\gamma) \rightarrow \infty$; 而式(11.18)中 $\phi(\gamma) \rightarrow 0$ 。尽管如此, RBF 网络都给出了很好的逼近能力(Powell M J D, 1987)。

在非线性函数 $\phi(\cdot)$ 取定的情况下, 确定 RBF 的结构, 就是决定中心点的个数和中心点的值。

11.5.2 RBF 网络的辨识

对于 RBF 网络的辨识, 就是辨识中心点 $\{c_i\}$ 和权值 λ_i 。应该指出的是, 这里所说的中心点, 是指输入向量空间的中心点(向量), 但实际上, 我们只能从输入样本点中去选择。最初的一些研究常常简单地从输入样本点中任意选取若干个为中心点, 这样构造的 RBF 网络模型要么性能不好, 要么维数太大。另外, 由于某些中心点太相近, 数据常出现病态情况, 即近似线性相关, 造成数值计算困难。后来较多地采用聚类的方法实现中心点的选择, 即通过聚类的方法得到聚类中心作为中心点, 然后再用最小二乘法估计式(11.16)的参数 $\{\lambda_i\}$ 。但这种选择中心点的方法只依据输入样本点本身, 而没有直接与拟合精度联系, 因此这样选出的中心点所构造的 RBF 网络常常拟合精度不高, 或经过反复多次才能得到较理想模型。Chen S 等(1991)提出了一种基于正交最小二乘的选择方法, 较好地克服了数据病态情况。

1. 正交最小二乘方法

假定已取得到 N 组样本: $\{y(k), x(k) = (x_1(k), \dots, x_m(k))^T; k=1, \dots, N\}$, 选中心点就是从 $\{x(k), k=1, \dots, N\}$ 中去选择。对于充分多的中心点 $\{c_i\} (i=1, \dots, n)$ 和选定的非线性函数 $\phi(\cdot)$, 记 $p_i = \phi(\|x(t) - c_i\|)$, $p_i(k) = p_i(x(k)) = \phi(\|x(k) - c_i\|)$, $k=1, 2, \dots, N$ 。显然, $p_i(k)$ 是 $x(k)$ 的函数。这时 RBF 网络式(11.16)可看做线性回归模型的特殊形式, 即

$$y(k) = \lambda_0 + \sum_{i=1}^n \lambda_i p_i(k) + e(k)$$

上式中 λ_0 不是本质的, 只要将 λ_0 作为 $p_0(k) \equiv 1$ 对应的参数, 则其即可写入和号 \sum 中。因此, 为了方便, 又不失一般性, 常将上式写为

$$y(k) = \sum_{i=1}^n \lambda_i p_i(k) + e(k) \quad (11.21)$$

其中, $y(k)$ 是输出, λ_i 是参数, $p_i(k)$ 是回归项, $e(k)$ 是误差, 与回归项 $p_i(k)$ 不相关。

由式(11.21)不难看出, 选择中心点的问题等价于选择式(11.21)中回归项的问题。一般来说, 全部候选项的数目非常大(最多等于全部样本点的个数), 而建模只需要其中很少的重要项就够了。如何把那些能满足描述系统的较少的重要回归项选出来一直是研究者追求的目标, 向前回归的正交最小二乘法在一定意义下能满足这一要求。下面具体介绍这一方法。

对于 $k=1, 2, \dots, N$, 由式(11.21)可得

$$\begin{aligned}
y(1) &= \lambda_1 p_1(1) + \cdots + \lambda_n p_n(1) + e(1) \\
y(2) &= \lambda_1 p_1(2) + \cdots + \lambda_n p_n(2) + e(2) \\
&\vdots \\
y(N) &= \lambda_1 p_1(N) + \cdots + \lambda_n p_n(N) + e(N)
\end{aligned}$$

进一步简化为向量形式

$$\mathbf{y} = \lambda_1 \mathbf{p}_1(N) + \cdots + \lambda_n \mathbf{p}_n(N) + \mathbf{e} = \mathbf{P}\boldsymbol{\theta} + \mathbf{e} \quad (11.22)$$

式中, $\mathbf{y} = (y(1), \cdots, y(N))^T$, $\mathbf{P} = [\mathbf{p}_1 \cdots \mathbf{p}_n]$, $\mathbf{p}_i = [p_i(1), \cdots, p_i(N)]^T, i = 1, \cdots, n, \boldsymbol{\theta} = (\lambda_1, \cdots, \lambda_n)^T, \mathbf{e} = (e(1), \cdots, e(N))^T$ 。

从式(11.22)可以看出, $\{\mathbf{p}_i\}$ 形成了一组基向量, $\mathbf{P}\boldsymbol{\theta}$ 是 \mathbf{y} 在由这组基向量展成的空间上的投影。用能量的观点, 换句话说, $\mathbf{P}\boldsymbol{\theta}$ 的平方描述了期望输出 \mathbf{y} 能够由回归项计算的那部分能量。但不同回归项之间一般是相关联的, 每个回归项对输出能量的贡献如何并不清楚。因此, 用正交最小二乘法将这组基向量变换成正交基向量, 这样可以计算每个基向量对期望输出能量的贡献, 以便进一步选出贡献大的回归项, 构成新基向量, 即确定了中心点。

将式(11.22)中矩阵 \mathbf{P} 进行正交分解得

$$\mathbf{P} = \mathbf{V} * \mathbf{A} \quad (11.23)$$

其中

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1n} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2n} \\ 0 & 0 & 1 & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \alpha_{n-1n} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \quad (11.24)$$

\mathbf{V} 是 $N \times n$ 的列正交矩阵。

记 $\mathbf{g} = \mathbf{A}\boldsymbol{\theta} = (g_1, \cdots, g_n)^T$, 则由式(11.22)可得辅助模型

$$\mathbf{y} = \mathbf{V}\mathbf{g} + \mathbf{e} \quad (11.25)$$

最小二乘估计为

$$\hat{\mathbf{g}} = (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T \mathbf{y}$$

即

$$\hat{g}_i = \frac{\sum_{k=1}^N v_i(k) y(k)}{\sum_{k=1}^N v_i^2(k)}, \quad i = 1, 2, \cdots, n \quad (11.26)$$

式中, $v_i(k)$ 为矩阵 \mathbf{V} 的第 i 个列向量 \mathbf{v}_i 的第 k 个元素。具体地, 正交变换可由 Gram-Schmidt 方法实现

$$\begin{cases} \mathbf{v}_1 = \mathbf{p}_1 \\ \alpha_{ij} = \frac{\mathbf{v}_i^T \mathbf{p}_j}{\mathbf{v}_i^T \mathbf{v}_i}, \quad 1 \leq i < j \\ \mathbf{v}_j = \mathbf{p}_j - \sum_{i=1}^{j-1} \alpha_{ij} \mathbf{v}_i, \quad j = 2, \cdots, n \end{cases} \quad (11.27)$$

模型式(11.22)中参数 $\boldsymbol{\theta}$ 的估计 $\hat{\boldsymbol{\theta}}$ 满足 $\mathbf{A}\hat{\boldsymbol{\theta}} = \hat{\mathbf{g}}$ 。

将式(11.25)展开得

$$\mathbf{y} = \sum_{i=1}^n g_i \mathbf{v}_i + \mathbf{e} \quad (11.28)$$

由于 $\{\mathbf{v}_i\}$ 是正交的, 因此可定义

$$[\text{err}]_i = \frac{g_i^2 \mathbf{v}_i^T \mathbf{v}_i}{\mathbf{y}^T \mathbf{y}}, \quad 1 \leq i \leq n$$

为基向量 \mathbf{v}_i 对描述输出 \mathbf{y} 所做的贡献, 故可以作为选择回归项的一个准则。

综合上述, 用正交最小二乘法选择中心点的步骤归纳如下:

(1) 对 $1 \leq i \leq n$ 计算

$$\begin{aligned} \mathbf{v}_1^{(i)} &= \mathbf{p}_i \\ g_1^{(i)} &= \frac{(\mathbf{v}_1^{(i)})^T \mathbf{y}}{(\mathbf{v}_1^{(i)})^T \mathbf{v}_1^{(i)}} \\ [\text{err}]_1^{(i)} &= \frac{(g_1^{(i)})^2 ((\mathbf{v}_1^{(i)})^T \mathbf{v}_1^{(i)})}{\mathbf{y}^T \mathbf{y}} \end{aligned}$$

(2) 取 $[\text{err}]_1^{(i_1)} = \max\{[\text{err}]_1^{(i)}, 1 \leq i \leq n\}$, 选择 $\mathbf{v}_1 = \mathbf{v}_1^{(i_1)} = \mathbf{p}_{i_1}$ 。

(3) 置 $k=2$ 。

(4) 对于 $1 \leq i \leq n, i \neq i_1, \dots, i \neq i_{k-1}$, 计算

$$\begin{aligned} \alpha_{jk}^{(i)} &= \frac{\mathbf{v}_j^T \mathbf{p}_i}{\mathbf{v}_j^T \mathbf{v}_j}, \quad 1 \leq j < k \\ \mathbf{v}_k^{(i)} &= \mathbf{p}_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} \mathbf{v}_j \\ g_k^{(i)} &= \frac{(\mathbf{v}_k^{(i)})^T \mathbf{y}}{(\mathbf{v}_k^{(i)})^T \mathbf{v}_k^{(i)}} \\ [\text{err}]_k^{(i)} &= \frac{(g_k^{(i)})^2 ((\mathbf{v}_k^{(i)})^T \mathbf{v}_k^{(i)})}{\mathbf{y}^T \mathbf{y}} \end{aligned}$$

取 $[\text{err}]_k^{(i_k)} = \max\{[\text{err}]_k^{(i)}, 1 \leq i \leq k, i \neq i_1, \dots, i \neq i_{k-1}\}$ 。选择 $\mathbf{v}_k = \mathbf{v}_k^{(i_k)} = \mathbf{p}_{i_k} - \sum_{j=1}^{k-1} \alpha_{jk} \mathbf{v}_j$, 其中, $\alpha_{jk} = \alpha_{jk}^{(i_k)}, 1 \leq j < k$ 。

(5) 如果 $1 - \sum_{j=1}^k [\text{err}]_j^{i_j} < \rho$ (其中 $0 < \rho < 1$ 是选定的容许误差), 则停止选择, 中心点 (即回归项) 即选出; 否则置 $k = k+1$, 转步骤 (4)。

从以上步骤不难看出, 向前回归正交最小二乘法的具体实施过程是: 首先将样本点构造回归项 \mathbf{p}_i , 通过正交选出一个贡献最大的项, 然后将其余项与之正交化, 再在剩余正交化项中选择贡献最大项, 再正交化, 再选择, 逐项增加, 直到满足允许误差为止。

正交最小二乘法有效地避免了由于病态而带来的数值计算的麻烦, 该方法是只添不删的向前回归的方法, 向量的选择是基于正交化的。对于数据多的情况, 计算是相当麻烦的。从式 (11.22) 不难看出, 这里选择重要回归项问题, 完全可以用 8.7 节所介绍的可添可删的双向回归递推算算法来实现。另外, 将 8.7 节中新息计算公式展开不难发现, 新息的计算与正交化计算是一致的, 因此采用 8.7 节的方法, 不仅可有效地避免病态, 可以得到最简形式的模型, 而且大大减化计算。

11.5.3 用 RBF 网络建模实例

【例 11.3】用 RBF 网络对前东德月失业人数时间序列建模

图 11.14 显出了前东德从 1948 年 1 月至 1980 年 12 月每月的失业人数(这 396 个数据可在 Subba Rao and M M Gabr(1984)的文献附录中找到)。

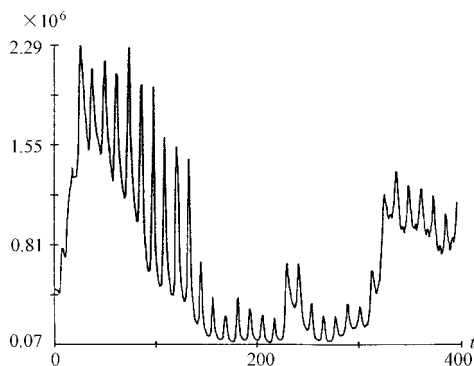


图1.14 前东德(1948.1~1980.12)月失业人数图

目的是要用 RBF 网络

$$\hat{y}(t) = f_r(\mathbf{x}(t)) \quad (11.29)$$

作为输出 $y(t)$ 的一步预测器,其中 RBF 网络的输入为

$$\mathbf{x}(t) = [y(t-1), \dots, y(t-n_y)]^T \quad (11.30)$$

非线性函数 $\phi(\cdot)$ 选为薄板样条函数(见式(11.17))。由于数据存在季节性(12 个月一周期)和线性趋势,故取延迟 n_y 为 13。RBF 的中心点就是从数据点 $\{x(t)\}$ 中选择,而且按上升顺序从全部 400 个候选项中选择。用前面介绍的正交最小二乘选择,取 $\rho=0.003$,共选出 55 个中心点,最终得到 RBF 网络为

$$\hat{y}(t) = \hat{\lambda}_0 + \sum_{i=1}^{55} \hat{\lambda}_i \phi(\|\mathbf{x}(t) - \mathbf{c}_i\|)$$

其中

$$\|\mathbf{x}(t) - \mathbf{c}_i\|^2 = \sum_{j=1}^{13} (y(t-j) - c_{ij})^2, 1 \leq i \leq 55$$

式中, $\mathbf{c}_i = (c_{i1}, \dots, c_{i13})^T$ 是选出的中心点。

$\hat{y}(k)$ 作为 $y(k)$ 的向前一步预测,预测误差 $\epsilon(t) = y(t) - \hat{y}(t)$ 如图 11.15 所示。

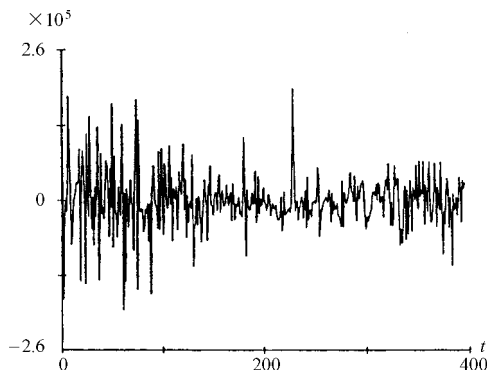


图 11.15 用 RBF 网络的一步预测误差图

【例 11.4】 用 RBF 网络对青岛海尔股市日平均价格建模

取青岛海尔股市 1995 年一年的日平均价格(223 个)作为时间序列,建立 RBF 网络模型作为一步预测器

$$\hat{y}(t) = f_r(x(t))$$

式中, $x(t)=[y(t-1), \cdots, y(t-n_y)]^T$ 作为网络的输入,非线性变换中 $\phi(\cdot)$ 取为高斯函数(见式(11.18))且 $\beta^2=2$ 。由于数据的波动性,取 $n_y=30$ 。建模过程是将数据分成两部分:前半年的 111 个数据作为训练集,用来建立 RBF 模型;后半年的 112 个数据作为测试集,用于测试模型的预测效果。取 $\rho=0.01$,共选出 10 个中心点,最终的 RBF 模型为

$$\hat{y}(t) = \hat{\lambda} + \sum_{i=1}^{10} \hat{\lambda}_i \phi(\|x(t) - c_i\|) \tag{11.31}$$

式中, $\|x(t) - c_i\|^2 = \sum_{j=1}^{30} (y(t-j) - c_{ij})^2, 1 \leq i \leq 10, c_i = (c_{i1}, \cdots, c_{i30})^T$ 为选出的中心点。

用同样的数据,用 8.4 节给出的可添可删的双向回归递推算法,则选出 7 个中心点(选择过程如表 11.1 所示)。最终的 RBF 模型为

$$\hat{y}(t) = \hat{\lambda}_0 + \sum_{i=1}^7 \hat{\lambda}_i \phi(\|x(t) - c_i\|) \tag{11.32}$$

两种算法得到的 RBF 模型的预测值 $\hat{y}(t)$ 与实测值的比较如图 11.16 所示。

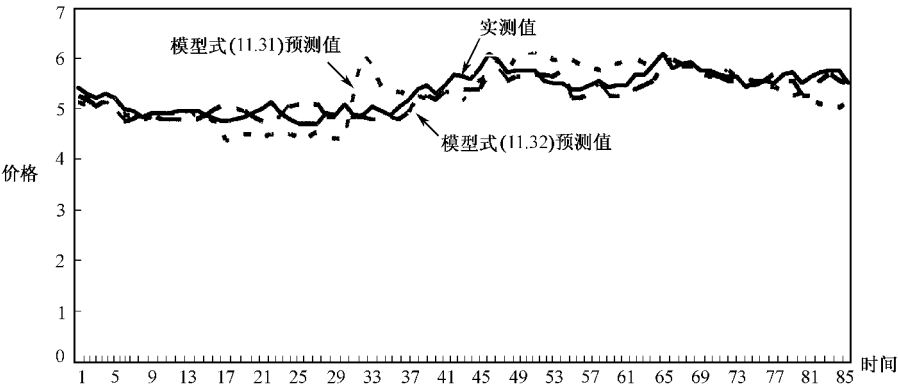


图 11.16 实测值、模型式(11.31)预测值和模型式(11.32)预测值的比较

从仿真结果可以看出,8.7 节给出的可添可删的双向回归算法用于 RBF 网络的辨识,不仅可以得到更简练的 RBF 模型,而且有较高的精度。

表 11.1 用双向回归算法建立 RBF 模型的选项过程

选项步数	各种指标值
第 1 步: 所选项为第 52 项	$Q_{\max}=4.623716$ $BIC=-125.463959, AIC=-128.098938$ 辨识参数结果:5.013499
第 2 步: 所选项为第 7 项	$Q_{\max}=1.163401$ $BIC=-155.620956, AIC=-159.573425$ 辨识参数结果:5.213508 -1.097033

选项步数	各种指标值
第 3 步： 所选项为第 63 项	$Q_{\max}=2.744194$ $BIC=-163.213058$ ， $AIC=-168.483017$ 辨识参数结果：5.173243 -1.321164 -0.995433
第 4 步： 所选项为第 61 项	$Q_{\max}=0.469988$ $BIC=-197.953033$ ， $AIC=-204.540482$ 辨识参数结果：5.225351 -0.725160 -1.094618 2.489680
第 5 步： 所选项为第 39 项	$Q_{\max}=0.373078$ $BIC=-202.970108$ ， $AIC=-210.875031$ 辨识参数结果：5.225351 -0.725160 -1.094618 2.489680 -1.857632
第 6 步： 所选项为第 12 项	$Q_{\max}=0.312002$ $BIC=-205.559647$ ， $AIC=-214.782074$ 辨识参数结果：5.193958 -0.714862 -1.653229 2.543853 -1.836714 -0.451124
第 7 步： 所选项为第 21 项	$Q_{\max}=0.136344$ $BIC=-210.435074$ ， $AIC=-220.974976$ 辨识参数结果：5.261152 -0.778815 -1.897703 2.494489 -1.848765 -0.551544 0.999892

习 题

1. 建立人工神经网络模型的 3 方面要素是什么？

2. 设非线性函数为 S 形函数 $f(x) = \frac{1}{1 + e^{-\beta x}}$, β 为常数, 求 $\frac{df(x)}{dx}$ 。

3. 非线性函数仍为上题中的 S 形函数, 给出用 BP 算法修正权值的具体表达式。

4. 用 C 语言编写一套计算机程序, 用于执行 BP 学习算法。

5. 设非线性系统为 $y(k) = \frac{y(k-1)y(k-2)u(k-1)[y(k-3)-1] + u(k-2)}{1 + y^2(k-1) + y^2(k-2)}$

其中, $u(k) = \sin(2\pi k/250)$, 设神经网络的结构为 5-20-1 前馈神经网络, 输入为 $y(k-1)$, $y(k-2)$, $y(k-3)$, $u(k-1)$, $u(k-2)$, 用 C 语言编程, 用 BP 算法学习, 并探讨步长选取对学习速度的影响。

6. 用 C 语言编一套正交最小二乘学习的计算机程序, 用于辨识 RBF 网络模型。

7. 给出将 NARMAX 模型的递推辨识算法用于 RBF 网络辨识的具体步骤。

8. 假设 RBF 网络模型的中心点及非线性函数 $\phi(\cdot)$ 已确定, 即结构已确定, 给出用递推最小二乘估计权值的递推公式。

第 12 章 模糊建模与辨识

前面所讲的建模和辨识方法,都是建立在对象的所谓精确数学模型。然而,随着社会、经济的迅速发展,对科学管理、预测、决策和控制深度和广度方面要求的不断提高,以及科学研究的不断深入,所研究的对象越来越复杂,影响因素也越来越多,甚至存在一些难以精确描述的人为因素,且存在着大量严重非线性、时变等非常复杂的现象,很难建立精确的数学模型。因此,在复杂的大系统中,复杂性与精确性形成了尖锐的矛盾。正如 L A Zadeh(1965,1975)所指出的:当系统日益复杂时,人们对它的精确而又有意义的描述能力将相应地降低,以至于达到精确性与有意义成为两个几乎互相排斥特征的地步。要想确切地描述复杂现象的任何现实的物理状态,事实上是不可能的。为了对整个问题描述有意义,我们必须在准确与简明之间取得平衡。模糊集合(Fuzzy Sets)的提出,为用简单的方法对复杂系统做出合乎实际的处理,提供了有力的数学工具。

基于模糊集合理论,用系统的输入、输出量测数据来辨识系统模糊模型,是系统建模与辨识的又一思路,而且是对复杂系统建模与辨识的一种有效途径。尽管这种模糊模型与通常的精确模型相比,似乎显得比较粗糙,但它同样也能对复杂系统的基本特征给出严格的定量描述。因此,在复杂系统的建模、预测、控制与决策中得到了广泛的应用,并且越来越显示出其重要作用。另外,在实际的一些复杂过程控制中,也经常会出现这种情况,一些控制器很难控制的问题,操作人员和工程师凭其经验和知识却能得到满意的控制效果,但操作人员却不能够用语言确切地描述出在什么情况下采用什么动作的完整过程。因此,给出通过数据对人的控制行为进行模糊建模也是很有意义的另一类问题。这不仅可将人的有效控制模型化、规范化,从而便于推广应用,而且当认为人的控制不是最优时,还可以通过模型进行优化。

总之,无论是要模拟操作人员的控制行为,还是要建立基于模型的控制;无论是要对复杂系统的行为进行预测,还是要为达到某个目标而进行决策,对系统的模糊建模都是非常有意义和必要的。

近年来,由 Takagi 和 Sugeno (Takagi T and Sugeno M,1988)提出的模糊模型(简称为 T-S 模型)得到了广泛的应用。本章以 T-S 模型为对象,介绍模糊建模的一些基本概念和方法,为进一步研究和应用模糊建模打下初步基础。为了下面叙述方便,这里先简单地介绍与模糊建模有关的一些模糊集合的基本知识和概念。

12.1 模糊集合的基本概念

我们知道,一个概念通常从两个方面去表达:一方面是概念所包含的那些区别于其他概念的全体本质属性,称为概念的内涵;另一方面是符合某概念的对象的整体,就是概念的外延。比如“人”这个概念的外延就是世界上所有人的全体,而内涵就是人区别于其他动物的那些本质属性的全体,如“能制造和使用工具”,“具有抽象概括、推理和思维的能力”等。从集合论的角度看,内涵就是集合的定义,而外延则是组成该集合的所有元素。

在康托(George Cantor)创立的经典集合论中,一事物要么属于某集合,要么不属于某集

合,二者必居其一,没有模棱两可的情况。但在现实世界中,在人们的思维中,有许多事物的概念是模糊的,即一个对象是否属于某个概念难以确定,这种由于概念的外延的模糊而造成的不确定性称为模糊性。如在讨论人的年龄时,那么“青年”、“中年”、“老年”都没有明确的外延;或考虑人的身高时,那么“高个子”、“中等身材”、“矮个子”也没有明确的外延;再如以某炉温为对象时,那么“高温”、“中温”、“低温”等也没有明确的外延,等等,所以诸如此类的概念都是模糊概念。

模糊概念不能用经典集合论加以描述,这是因为它不能绝对地区别“属于”或“不属于”,也就是说,所研究对象集合中的元素符合概念的程度不能像在经典集合论中用特征函数描述的那样,要么是“0”,要么是“1”,而是介于 0 和 1 之间的一个实数。Zadeh 教授在论文《模糊集合论》中(Zadeh L A,1965)提出的“隶属函数”(Membership Function)的概念突破了经典集合论中属于或不属于的绝对关系,恰当地描述了对对象差异的中间过渡这种模糊性。下面仅对本章所用到的有关模糊集的一些基本概念做简单介绍。

12.1.1 模糊集合及其表示

设 U 为某对象所有元素的集合,称为论域,可以是连续的或离散的; u 表示集合 U 的元素,通常记集合 $U = \{u\}$ 。

1. 模糊集合(Fuzzy Sets)

论域 U 到闭区间 $[0,1]$ 的任意一个映射 μ_F ,即 $\mu_F:U \rightarrow [0,1], u \rightarrow \mu_F(u)$,都确定 U 的一个模糊子集 F , μ_F 称为模糊子集的隶属函数, $\mu_F(u)$ 称为 u 对于 F 的隶属度(Grade of Membership)。隶属度也可记为 $F(u)$ 。在不混淆的情况下,模糊子集也称为模糊集合。

上述定义表明,论域 U 上的模糊子集 F 可由隶属函数 $\mu_F(u)$ 来表征, $\mu_F(u)$ 的取值范围为闭区间 $[0,1]$ 。 $\mu_F(u)$ 的大小反映了 u 属于模糊子集 F 的程度或等级。 $\mu_F(u)$ 的值接近于 1,表示 u 属于 F 的程度很高; $\mu_F(u)$ 的值接近于 0,表示 u 属于 F 的程度很低。可见,模糊子集完全可由隶属函数来描述。

当 $\mu_F(u)$ 的值域等于 $\{0,1\}$ 时, $\mu_F(u)$ 蜕化成经典子集的特征函数,模糊子集 F 便蜕化成一个经典子集。由此不难看出,经典集合是模糊集合的特殊情况,模糊集合是经典集合概念的推广。

2. 模糊集合的表示

(1) 当论域 U 为有限离散集 $\{u_1, u_2, \dots, u_n\}$ 时

通常有 3 种表示方式。

① Zadeh 表示法

$$F = \frac{F(u_1)}{u_1} + \frac{F(u_2)}{u_2} + \dots + \frac{F(u_n)}{u_n}$$

式中, $\frac{F(u_i)}{u_i}$ 并不表示“分数”,而是表示论域中的元素 u_i 与其隶属度 $F(u_i)$ 之间的对应关系;“+”号也不表示求和,而是表示模糊集合在论域 U 上的整体。

【例 12.1】 在由整数 $1, 2, 3, \dots, 10$ 组成的论域中,即 $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$,讨论“几个”这一模糊概念。根据经验,可以定量地给出它们的隶属函数。模糊子集“几个”可

表示为:

$$F = 0.0/1 + 0.0/2 + 0.3/3 + 0.7/4 + 1.0/5 + 1.0/6 + 0.7/7 + 0.3/8 + 0.0/9 + 0.0/10$$

上式表明,5个、6个的隶属度为1,说明用“几个”表示5个、6个的可能性最大;而4个、7个对于“几个”这个模糊概念的隶属度为0.7;通常不采用“几个”来表示1个、2个或9个、10个,因此它们的隶属度为零。

在论域 U 中,所有满足 $\mu_F(u) > 0$ 的元素 u 构成的模糊子集,称为模糊集 F 的支集(也称为 F 的台)。实际上,若某元素的隶属度为零,即它不属于这个集合。因此,为了表达简单,通常用支集来表达一个模糊集合,如模糊集合“几个”可表示为

$$F = 0.3/3 + 0.7/4 + 1.0/5 + 1.0/6 + 0.7/7 + 0.3/8$$

② 序偶表示法

将论域中元素 u_i 与其隶属度 $F(u_i)$ 构成序偶来表示 F ,则 F 可表示为

$$F = \{(u_1, F(u_1)), (u_2, F(u_2)), \dots, (u_n, F(u_n))\}$$

采用序偶表示法,例12.1中的 F 可写为

$$F = \{(3, 0.3), (4, 0.7), (5, 1.0), (6, 1.0), (7, 0.7), (8, 0.3)\}$$

此种表示法隶属度为零的项可不写入。

③ 向量表示法

用此种表示法,隶属度为零的项不能省略: $F = \{F(u_1), F(u_2), \dots, F(u_n)\}$ 。

采用向量表示法,例12.1中的 F 可表示为 $F = \{0.0, 0.0, 0.3, 0.7, 1.0, 1.0, 0.7, 0.3, 0.0, 0.0\}$ 。

有时也将上述3种方法结合起来表示为: $F = \left(\frac{F(u_1)}{u_1}, \frac{F(u_2)}{u_2}, \dots, \frac{F(u_n)}{u_n} \right)$,则例12.1可表

示为 $F = \left(\frac{0.3}{3}, \frac{0.7}{4}, \frac{1.0}{5}, \frac{1.0}{6}, \frac{0.7}{7}, \frac{0.3}{8} \right)$,这里也舍弃了隶属度为零的项。

(2) 当论域 U 为有限连续域时

模糊集合 F 表示为

$$F = \int_U \frac{\mu_F(u)}{u}$$

同样, $\frac{\mu_F(u)}{u}$ 并不表示“分数”,而表示论域中的元素 u 与隶属度 $\mu_F(u)$ 之间的对应关系;符号“ \int ”既不表示“积分”,也不表示“求和”,而是表示论域 U 上的元素 u 与隶属度 $\mu_F(u)$ 对应关系的一个总括。

【例12.2】以年龄为论域,取 $U = [0, 200]$ ，“年青” Y 模糊集合的隶属函数为

$$\mu_Y(u) = \begin{cases} 1 & 0 \leq u \leq 25 \\ \left[1 + \left(\frac{u-25}{5} \right)^2 \right]^{-1} & 25 < u \leq 200 \end{cases}$$

其隶属函数曲线如图12.1所示。

“年青” Y 模糊集合可写为

$$Y = \int_{0 \leq u \leq 25} \frac{1.0}{u} + \int_{25 < u \leq 200} \frac{\left[1 + \left(\frac{u-25}{5} \right)^2 \right]^{-1}}{u}$$

如果给定 $u=60$,则可由隶属函数计算出属于“年青”这个模糊集的隶属度 $\mu_Y(60) = 0.02$,这说明60岁不应属于“年青”之列。

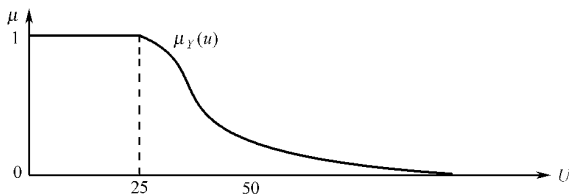


图 12.1 “青年”隶属函数曲线

在给定的论域上,可以有多个模糊子集,如在上述年龄论域上还可以定义“年老”、“中年”等模糊子集。

12.1.2 模糊集的运算

1. 模糊集的并、交、补运算

设 A 和 B 为论域 U 中的两个模糊集,其隶属函数分别为 $\mu_A(u)$ 和 $\mu_B(u)$,则对于所有的 $u \in U$,存在下列运算。

(1) A 与 B 的并(逻辑“或”),记为 $A \cup B$,其隶属函数定义为

$$\mu_{A \cup B}(u) = \mu_A(u) \vee \mu_B(u) = \max\{\mu_A(u), \mu_B(u)\}$$

(2) A 与 B 的交(逻辑“与”),记为 $A \cap B$,其隶属函数定义为

$$\mu_{A \cap B}(u) = \mu_A(u) \wedge \mu_B(u) = \min\{\mu_A(u), \mu_B(u)\}$$

(3) A 的补(逻辑“非”),记为 A^c ,其隶属函数定义为

$$\mu_{A^c}(u) = 1 - \mu_A(u)$$

【例 12.3】 设论域 $U = \{x_1, x_2, x_3, x_4\}$, A, B 是该论域上的两个模糊子集,已知

$$A = 0.3/x_1 + 0.5/x_2 + 0.7/x_3 + 0.4/x_4$$

$$B = 0.5/x_1 + 1.0/x_2 + 0.8/x_3$$

利用模糊集的运算可得

$$A^c = \frac{0.7}{x_1} + \frac{0.5}{x_2} + \frac{0.3}{x_3} + \frac{0.6}{x_4}$$

$$B^c = \frac{0.5}{x_1} + \frac{0.2}{x_3} + \frac{1.0}{x_4}$$

$$\begin{aligned} A \cup B &= \frac{0.3 \vee 0.5}{x_1} + \frac{0.5 \vee 1.0}{x_2} + \frac{0.7 \vee 0.8}{x_3} + \frac{0.4 \vee 0.0}{x_4} \\ &= \frac{0.5}{x_1} + \frac{1.0}{x_2} + \frac{0.8}{x_3} + \frac{0.4}{x_4} \end{aligned}$$

$$\begin{aligned} A \cap B &= \frac{0.3 \wedge 0.5}{x_1} + \frac{0.5 \wedge 1.0}{x_2} + \frac{0.7 \wedge 0.8}{x_3} + \frac{0.4 \wedge 0.0}{x_4} \\ &= \frac{0.3}{x_1} + \frac{0.5}{x_2} + \frac{0.7}{x_3} \end{aligned}$$

2. 模糊集的代数运算

(1) 代数积:称 $A \cdot B$ 为模糊集合 A 和 B 的代数积, $A \cdot B$ 的隶属函数为

$$\mu_{A \cdot B} = \mu_A \cdot \mu_B$$

(2) 代数和: 称 $A+B$ 为模糊集合 A 和 B 的代数和, $A+B$ 的隶属函数为

$$\mu_{A+B} = \begin{cases} \mu_A + \mu_B & \mu_A + \mu_B \leq 1 \\ 1 & \mu_A + \mu_B > 1 \end{cases}$$

(3) 环和: 称 $A \oplus B$ 为模糊集合 A 和 B 的环和, $A \oplus B$ 的隶属函数为

$$\mu_{A \oplus B} = \mu_A + \mu_B - \mu_{A \cdot B}$$

【例 12.4】论域及模糊集 A, B 仍为例 12.3 所定义, 则

$$A \cdot B = 0.15/x_1 + 0.5/x_2 + 0.56/x_3$$

$$A + B = 0.8/x_1 + 1.0/x_2 + 1.0/x_3 + 0.4/x_4$$

$$A \oplus B = 0.65/x_1 + 1.0/x_2 + 0.94/x_3 + 0.4/x_4$$

值得指出的是, 模糊集合运算的许多性质, 如幂等律、交换律、结合律等都与经典集合相同。但模糊集合不满足互补律, 即 $A \cup B^c \neq \Omega, A \cap A^c \neq \emptyset$ 。这是因为模糊子集 A 没有明确的边界, A^c 也无明确的边界。这个性质虽然给理论计算带来了麻烦, 但正是这一点, 使模糊集合更能客观地反应实际情况, 因为在大量实际问题中, 存在着许多模棱两可的情况, 人们限定为界限清楚的集合, 会使问题过于简单, 甚至失去意义。

12.1.3 常用的隶属函数

隶属函数是对模糊概念的定量描述。正确地确定隶属函数, 是运用模糊集理论解决实际问题的基础。但客观世界的模糊概念可以说是无穷无尽的, 要想找到能准确反映模糊概念的隶属函数的统一模式是不可能的。一般只能根据实际问题 and 应用目的, 通过经验、专家或用统计方法具体确定或选定。下面给出几种常用的隶属函数。

(1) 正态型

$$\mu(x) = e^{-\left(\frac{x-a}{b}\right)^2}, b > 0$$

其分布曲线如图 12.2 所示。

(2) Γ 型

$$\mu(x) = \begin{cases} 0 & x < 0 \\ \left(\frac{x}{\lambda\nu}\right)^\nu \cdot e^{-\frac{x}{\lambda}} & x \geq 0 \end{cases}$$

其中, $\lambda > 0, \nu > 0$ 。当 $x = \lambda\nu$ 时, 隶属度为 1, 其分布曲线如图 12.3 所示。

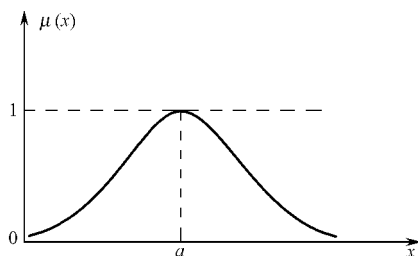


图 12.2 正态型分布曲线

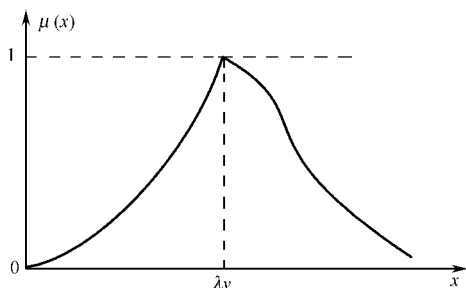


图 12.3 Γ 型分布曲线

(3) 戒上型

$$\mu(x) = \begin{cases} 1 & x \leq c \\ \frac{1}{1 + [a(x-c)]^b} & x > c \end{cases}$$

其中, $a > 0, b > 0$ 。其分布曲线如图 12.4 所示。

当 $a = 0.2, b = 2, c = 25$ 时, 即为“年青”的隶属函数; 当 $a = 0.1, b = 4, c = 15^\circ\text{C}$ 时, 可作为“不热”的隶属函数。

(4) 戒下型

$$\mu(x) = \begin{cases} 0 & x < c \\ \frac{1}{1 + [a(x-c)]^b} & x \geq c \end{cases}$$

其中, $a > 0, b < 0$ 。其分布曲线如图 12.5 所示。

当 $a = 0.2, b = -2, c = 50$ 时, 即为“年老”的隶属函数; 当 $a = 0.5, b = -2, c = 10^\circ\text{C}$ 时, 可作为“不冷”的隶属函数。

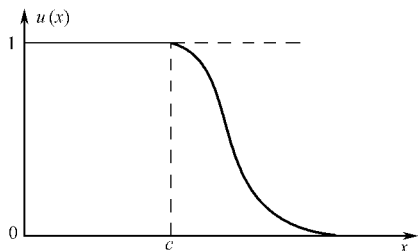


图 12.4 戒上型分布曲线

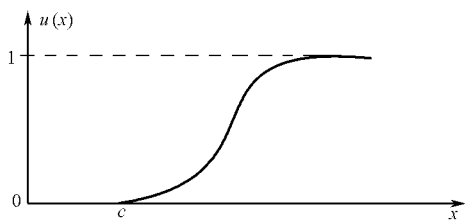


图 12.5 戒下型分布曲线

(5) 三角型

$$\mu(x) = \begin{cases} 1 - \frac{|x-c|}{\gamma} & |x-c| \leq \gamma, \gamma > 0 \\ 0 & \text{其他} \end{cases}$$

其分布曲线如图 12.6 所示。

(6) 梯型

$$\mu(x) = \begin{cases} 1 - \frac{c_1 - x}{\gamma} & c_1 - \gamma \leq x < c_1 \\ 1 & c_1 \leq x \leq c_2 \\ 1 - \frac{x - c_2}{\gamma} & c_2 < x \leq c_2 + \gamma \\ 0 & \text{其他} \end{cases} \quad \gamma > 0$$

其分布曲线如图 12.7 所示。

现实世界中的隶属函数形式可以有多种多样, 只能根据实际问题而确定或选定。

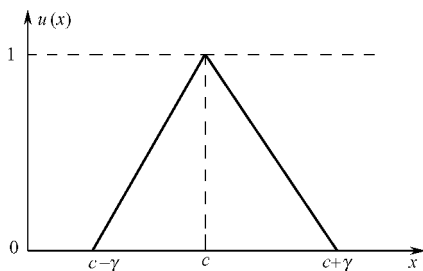


图 12.6 三角型分布曲线

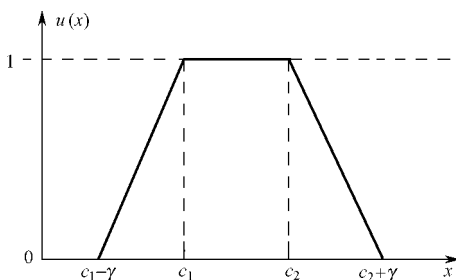


图 12.7 梯形分布曲线

12.1.4 模糊逻辑关系的格式

模糊逻辑关系(也称为模糊规则) R 的一般格式如下

$$R: \text{If } f(x_1 \text{ is } A_1, \dots, x_m \text{ is } A_m) \text{ then } y = g(x_1, x_2, \dots, x_m)$$

其中, y 称为结论变量, x_1, \dots, x_m 称为前提变量, A_1, \dots, A_m 是一组模糊子集。 f 表示连接前提命题的逻辑函数。 g 是当 x_1, \dots, x_m 满足前提时 y 的值函数。

【例 12.5】 $R: \text{If } x_1 \text{ is small and } x_2 \text{ is big then } y = x_1 + x_2 + 2x_3$ 。

该关系表明,如果 x_1 是小,并且 x_2 是大,那么 y 的值就等于 x_1, x_2 和 $2x_3$ 的和。其中 x_3 在前提中没有出现,说明对任意的 x_3 ,关系都成立,即 x_3 是绝对的。

在 T-S 模型中,前提中只用连接词“and”,结论中的值函数 g 采用线性函数,即逻辑关系只采用如下形式

$$R: \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \cdots \text{and } x_m \text{ is } A_m \text{ then } y = p_0 + p_1 x_1 + \cdots + p_m x_m$$

其中,模糊子集 A_1, \dots, A_m 的隶属函数中的参数称为前提参数。输出表达式中的 p_m 称为结论参数。

12.1.5 推理算法

这里采用的推理算法(也称为模糊判决,Defuzzification)是重心法。假定有上面格式的 n 个逻辑关系 $R^i (i=1, \dots, n)$,当给定 $x_1 = x_1^0, \dots, x_m = x_m^0$ 时,最终输出 y 的值由如下步骤确定。

(1) 对每一个逻辑关系 R^i ,计算结论 y^i 的值

$$y^i = g^i(x_1^0, \dots, x_m^0) = p_0^i + p_1^i x_1^0 + \cdots + p_m^i x_m^0$$

(2) 计算前提命题 $y = y^i$ 的“真值”,记为 $|y = y^i|$,即

$$\begin{aligned} |y = y^i| &= |x_1^0 \text{ is } A_1^i \text{ and } \cdots \text{ and } x_m^0 \text{ is } A_m^i| \\ &= A_1^i(x_1^0) \wedge \cdots \wedge A_m^i(x_m^0) \\ &= \min\{A_1^i(x_1^0), \dots, A_m^i(x_m^0)\} \end{aligned}$$

其中, $|x^0 \text{ is } A| = A(x^0)$,即 x^0 关于模糊集 A 的隶属度。

(3) 由 n 个逻辑关系推断出的最终输出 y 是以 $|y - y^i|$ 为权重的所有 y^i 的加权平均值,即

$$y = \frac{\sum_{i=1}^n |y - y^i| \times y^i}{\sum_{i=1}^n |y - y^i|}$$

【例 12.6】 假定有如下 3 个逻辑关系(模糊规则)

R^1 : If x_1 is $small_1$ and x_2 is $small_2$ then $y = x_1 + x_2$;

R^2 : If x_1 is big_1 then $y = 2x_1$;

R^3 : If x_2 is big_2 then $y = 3x_2$ 。

给定 $x_1=12, x_2=5$ 时, 每个模糊规则的推理过程由表 12.1 给出。

表 12.1 模糊规则及其模糊推理过程

规则	前提	结论	真 值
	If x_1x_2	Then y	
R^1		$y = 12 + 5$ $= 17$	$0.25 \wedge 0.375$ $= 0.25$
R^2		$y = 2 \times 12$ $= 24$	0.2
R^3		$y = 3 \times 5$ $= 15$	0.375

在表 12.1 中,“前提”一栏显示了规则前提模糊集的隶属函数“small”和“big”;“结论”一栏中显示了每个结论函数 g^i 计算出的 y^i 的值;“真值”栏显示了 $|y - y^i|$ 的真值。例如

$$\begin{aligned}
 |y - y^1| &= |x_1^0 = small_1| \wedge |x_2^0 = small_2| = small_1(x_1^0) \wedge small_2(x_2^0) \\
 &= small_1(12) \wedge small_2(5) = 0.25 \wedge 0.375 = 0.25
 \end{aligned}$$

由 3 条规则得到的最终输出值 y 为

$$y = \frac{0.25 \times 17 + 0.2 \times 24 + 0.375 \times 15}{0.25 + 0.2 + 0.375} = 17.8$$

12.2 基于 T-S 模型的模糊辨识

由日本高木(Takagi)和杉野(Sugeno)教授提出的模糊模型(简称 T-S 模型)(Takagi T and Sugeno M, 1985)是基于一组语言规则描述的模型,每一规则的具体格式如在 12.1.4 节所述,即前提中的逻辑关系只用“and”连接,且结论函数是线性的。具体地说,系统的模糊模型可用如下逻辑关系(模糊规则)描述

$$\begin{aligned}
 R^1: & \text{If } x_1 \text{ is } A_1^1, \dots, x_m \text{ is } A_m^1 \text{ then } y = p_0^1 + p_1^1 x_1 + \dots + p_m^1 x_m \\
 R^2: & \text{If } x_1 \text{ is } A_1^2, \dots, x_m \text{ is } A_m^2 \text{ then } y = p_0^2 + p_1^2 x_1 + \dots + p_m^2 x_m \\
 & \dots
 \end{aligned} \tag{12.1}$$

$$R^n: \text{If } x_1 \text{ is } A_1^n, \dots, x_m \text{ is } A_m^n \text{ then } y = p_0^n + p_1^n x_1 + \dots + p_m^n x_m$$

如果这几个规则已知,则可由这几个规则用 12.1.5 节给出的推理算法计算出系统的最终输出 y 。

该模型的辨识问题,就是通过目标系统的输入、输出数据确定:

- (1) x_1, x_2, \dots, x_m , 即组成逻辑关系的前提变量;
- (2) A_1, A_2, \dots, A_m , 即前提中模糊集的隶属函数,简称为前提参数;

(3) p_0, p_1, \dots, p_m , 即结论参数。

使得模型的输出 \hat{y} 与目标系统的实际输出 y 的误差平方和, 即

$$J = \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \quad (12.2)$$

达到最小。其中, $\hat{y}(k)$ 表示由辨识得到的模糊模型计算出的第 k 个采样时刻的输出值, $y(k)$ 为目标系统第 k 个采样的实际输出值, N 为总采样次数。

在 T-S 模糊模型辨识中, 前提参数的辨识和结论参数的辨识是分开进行的, 而且在具体步骤上又常是互为前提、交替进行的。辨识算法的基本流程如图 12.8 所示。

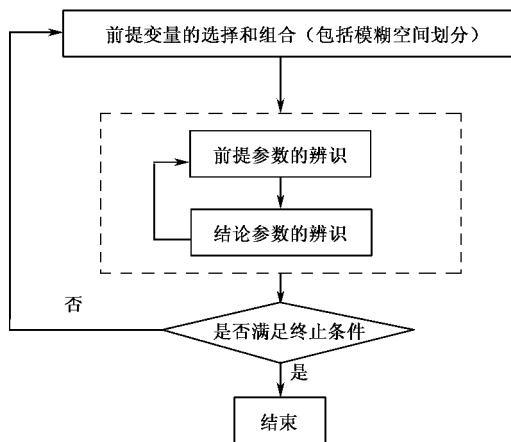


图 12.8 模糊辨识基本流程

在图 12.8 中, 之所以称为前提参数的辨识, 是因为在模糊模型中通常假定模糊集的隶属函数的形式是事先给定的, 只需确定其参数。在 T-S 模型中, 通常假定模糊集的隶属函数是线性的, 因此隶属函数取决于两个参数, 即其隶属度为最高“1”和最低“0”的两个参数。当然, 对于其他形式的隶属函数, 只要形式确定, 也只需确定其参数即可。例如, 12.1.3 节的正态型隶属函数, 只要确定其均值 a 和标准差 b 这两个参数即可。另外, 图 12.8 中的终止条件常采用如下某项指标:

- ① 性能指标(式(12.2))小于预定值;
- ② 规则数目超过或达到预定数;
- ③ 相邻两个的性能指标的相对变化小于给定的阈值。

下面分别介绍各部分的具体辨识算法。

12.2.1 结论参数的辨识

假定系统模型由式(12.1)描述, 即由如下一组模糊关系描述

$$R^1: \text{If } x_1 \text{ is } A_1^1, \dots, x_m \text{ is } A_m^1 \text{ then } y = p_0^1 + p_1^1 x_1 + \dots + p_m^1 x_m \quad (12.3)$$

...

$$R^n: \text{If } x_1 \text{ is } A_1^n, \dots, x_m \text{ is } A_m^n \text{ then } y = p_0^n + p_1^n x_1 + \dots + p_m^n x_m$$

假定前提条件已知, 即前提变量和隶属函数都已知, 下面的任务是辨识结论中的参数 p_j^i , $i=1, 2, \dots, n, j=0, 1, \dots, m$ 。显然, 模型式(12.3)对应于输入 (x_1, x_2, \dots, x_m) 的输出 y 可由下式计算

$$y = \frac{\sum_{i=1}^n (A_1^i(x_1) \wedge \cdots \wedge A_m^i(x_m)) \cdot (p_0^i + p_1^i x_1 + \cdots + p_m^i x_m)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \cdots \wedge A_m^i(x_m))} \quad (12.4)$$

记

$$\beta_i = \frac{A_1^i(x_1) \wedge \cdots \wedge A_m^i(x_m)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \cdots \wedge A_m^i(x_m))} \quad (12.5)$$

则式(12.4)可简化为

$$\begin{aligned} y &= \sum_{i=1}^n \beta_i (p_0^i + p_1^i x_1 + \cdots + p_m^i x_m) \\ &= \sum_{i=1}^n (p_0^i \beta_i + p_1^i x_1 \beta_i + \cdots + p_m^i x_m \beta_i) \end{aligned} \quad (12.6)$$

令

$$\boldsymbol{\theta}^T = (p_0^1, \cdots, p_0^n, p_1^1, \cdots, p_1^n, \cdots, p_m^1, \cdots, p_m^n) \quad (12.7)$$

$$\boldsymbol{\phi}^T = (\beta_1, \cdots, \beta_n, x_{11}\beta_1, \cdots, x_{1n}\beta_n, \cdots, x_{m1}\beta_1, \cdots, x_{mn}\beta_n) \quad (12.8)$$

则式(12.6)可简化为

$$y = \boldsymbol{\phi}^T \boldsymbol{\theta}$$

假定取得了 N 组数据(注意, N 需不小于 $n \cdot (m+1)$), 因为 $\boldsymbol{\theta}$ 中包含了 $n \cdot (m+1)$ 个数): $\{x_{1j}, x_{2j}, \cdots, x_{mj}; y_j; j=1, 2, \cdots, N\}$, 令

$$\boldsymbol{\phi}_k^T = (\beta_1, \cdots, \beta_n, x_{1k}\beta_1, \cdots, x_{1n}\beta_n, \cdots, x_{mk}\beta_1, \cdots, x_{nk}\beta_n) \quad (12.9)$$

其中 β_i 由式(12.5)计算。则

$$y_k = \boldsymbol{\phi}_k^T \boldsymbol{\theta} + e_k \quad k = 1, \cdots, N \quad (12.10)$$

其中, e_k 是误差, 使误差平方和最小可得 $\boldsymbol{\theta}$ 的最小二乘估计。即

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}_N^T \boldsymbol{\Phi}_N)^{-1} \boldsymbol{\Phi}_N^T \mathbf{y}_N \quad (12.11)$$

其中

$$\boldsymbol{\Phi}_N = \begin{bmatrix} \boldsymbol{\phi}_1^T \\ \boldsymbol{\phi}_2^T \\ \vdots \\ \boldsymbol{\phi}_N^T \end{bmatrix}, \mathbf{y}_N = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

显然, 式(12.11)可用如下递推算法(见 3.2 节)

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \frac{\mathbf{P}_k \boldsymbol{\phi}_{k+1}}{\boldsymbol{\phi}_{k+1}^T \mathbf{P}_k \boldsymbol{\phi}_{k+1} + 1} [y_{k+1} - \boldsymbol{\phi}_{k+1}^T \hat{\boldsymbol{\theta}}_k]$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P}_k \boldsymbol{\phi}_{k+1} \boldsymbol{\phi}_{k+1}^T \mathbf{P}_k}{\boldsymbol{\phi}_{k+1}^T \mathbf{P}_k \boldsymbol{\phi}_{k+1} + 1}$$

$$\hat{\boldsymbol{\theta}} = 0, \mathbf{P}_0 = \alpha I (\alpha \text{ 是一个充分大的正数})$$

12.2.2 前提参数的辨识

这里假定前提变量已经选定, 而且模糊子空间划分的个数已确定, 确定模糊子空间的问题

就是确定前提中的隶属函数,也即是确定使性能指标式(12.2)达到极小的隶属函数中的参数,该过程称为“前提参数辨识”。具体算法如下:

- (1) 给定初始前提参数;
- (2) 用 12.2.1 节给出的算法辨识结论参数;
- (3) 假定结论参数已知(计算出),使性能指标最小,求前提参数(这里每个模糊集都取决于两个参数),这是一个非线性规划问题,可采用梯度法、遗传算法等非线性辨识的一些方法。
- (4) 转步骤(2),直到达到预先要求。

12.2.3 前提变量的选择

正像前面所指出的,不是所有前提变量都出现在每个规则的前提中。因此,选择前提变量的算法应解决两个问题:一个是前提变量的选择(这关系着输入空间的划分),另一个问题是划分的数目,整个问题是一个组合问题。所以一般情况下,似乎不存在理论上的可行方法。下面给出一种启发式的搜索方法。

假定欲建立一个 m 个输入 (x_1, x_2, \dots, x_m) 、单输出系统的模糊模型,具体步骤如下:

- (1) 把 x_1 的范围划分为两个模糊子空间“big”和“small”,而不划分其他变量 x_2, \dots, x_m ,这就意味着只有 x_1 出现在逻辑关系的前提中。由两个逻辑关系组成的模型为

$$\text{If } x_1 \text{ is big}_1 \text{ then } y = p_{10}^1 + p_{11}^1 x_1 + \dots + p_{1m}^1 x_m$$

$$\text{If } x_1 \text{ is small}_1 \text{ then } y = p_{10}^2 + p_{11}^2 x_1 + \dots + p_{1m}^2 x_m$$

该模型记为模型 1-1。同样地,划分 x_2 ,而不划分其他变量 x_1, x_3, \dots, x_m ,得到的模型记为模型 1-2。这样一直继续下去,可以得到 m 个模型,每个模型都由两个逻辑关系组成。模型 1- i 的形式为

$$\text{If } x_i \text{ is big}_i \text{ then } y = p_{i0}^1 + p_{i1}^1 x_1 + \dots + p_{im}^1 x_m$$

$$\text{If } x_i \text{ is small}_i \text{ then } y = p_{i0}^2 + p_{i1}^2 x_1 + \dots + p_{im}^2 x_m$$

- (2) 用前面给出的方法辨识每个模型的前提参数和结论参数,并计算其性能指标。从 m 个模型中选出性能指标最小的模型,称为“稳态模型”。

- (3) 假如由步骤(2)得到的稳态模型是模型 1- i (只有 x_i 出现在前提中),那么将 x_i 与每个 $x_j (j=1, \dots, m)$ 分别组合,并把每个变量都划分为两个模糊子空间。对于组合 x_i-x_j ,则 x_i 被划分为 4 个子空间,如“big”,“medium big”,“medium small”,“small”。这样,又得到了 m 个模型,这些模型分别记为模型 2- $j, j=1, 2, \dots, m$ 。每个模型都由 2×2 个逻辑关系组成。再从中选出性能指标最小的稳态模型,不妨设为模型 2- k 。

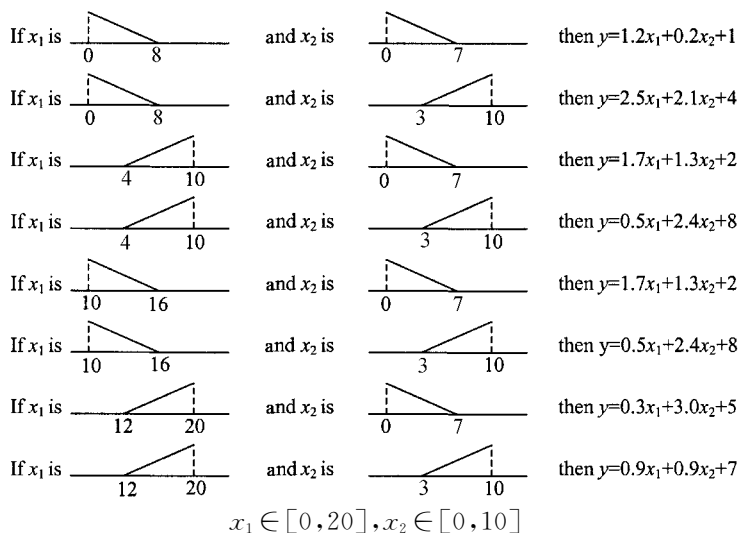
- (4) 将每一个变量与稳态模型 2- k 中的前提变量组合,重复步骤(3)。

- (5) 如果满足下列任一个条件,搜索停止。

- 稳态模型的性能指标小于预定值;
- 稳态模型的逻辑关系的个数超过了预定个数。

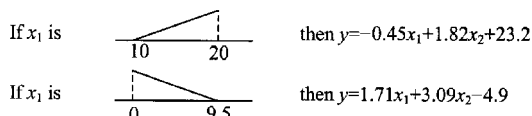
下面用一个仿真例子进一步说明模糊辨识的过程。

【例 12.7】 考虑一个两输入、单输出的模糊系统,系统模型如下

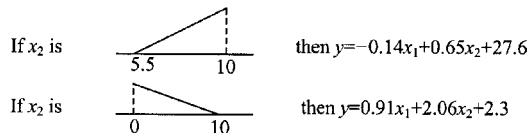


由该模型产生 440 个数据,并在输出中添加了噪声,噪声的标准差是输出的 2%。用这些数据辨识该系统的过程如下:

(1) 我们从两个模型开始,每个模型由两个逻辑关系组成模型 1-1:(划分 x_1)(性能指标为 2.55)

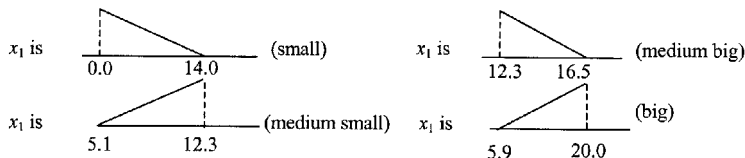


模型 1-2:(划分 x_2)(性能指标为 3.73)

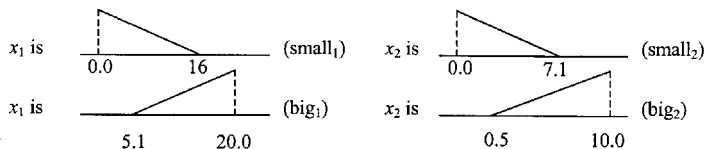


可以发现,在第一步中,模型 1-1 为稳态模型,因为它的性能指标最小。

(2) 进一步划分输入空间。在模型 2-1 中, x_1 被分为 4 个子空间,而 x_2 没有划分。

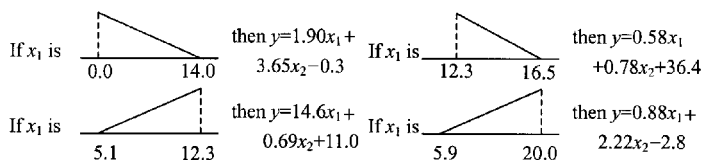


在模型 2-2 中, x_1, x_2 都被重新划分为两个子空间。即

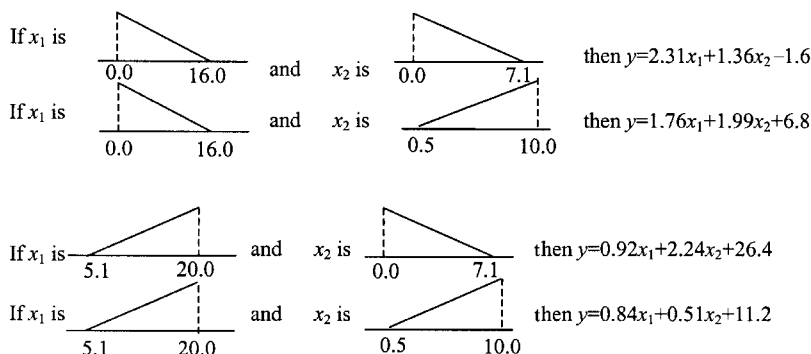


注意,这里对 x_1 的划分与模型 1-1 中的划分是不同的,这是因为关于 x_1 的模糊集和关于 x_2 的模糊集一起进行了优化。继续辨识得到:

模型 2-1:(性能指标为 1.91)



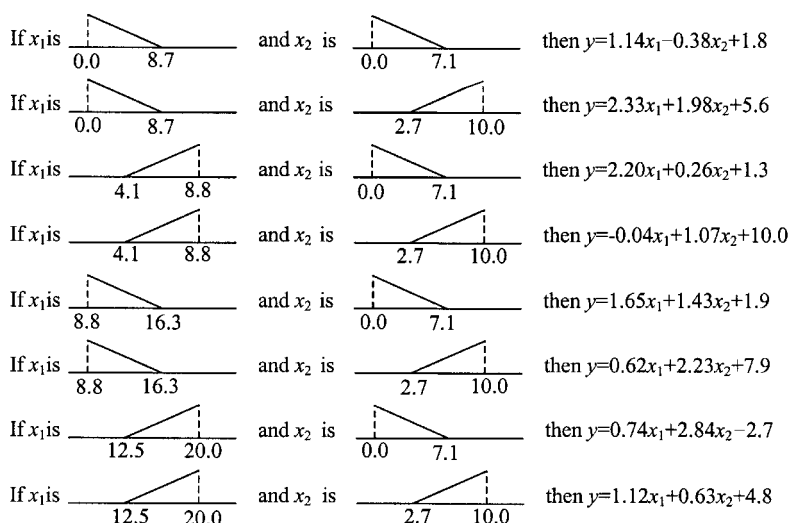
模型 2-2:(性能指标为 1.40)



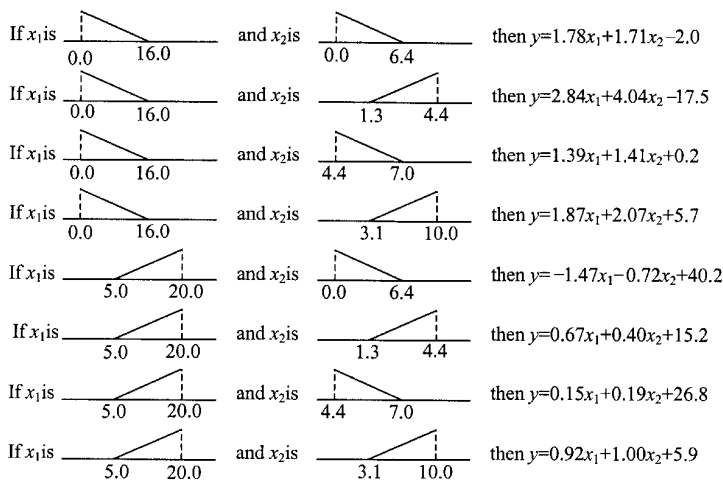
由此可见,模型 2-2 是第二步的稳态模型。

(3) 由模型 2-2 扩展的两个模型及性能指标如下。

模型 3-1:(x_1 分为 4 个子空间, x_2 分为 2 个子空间,性能指标为 1.08)



模型 3-2:(x_1 分为 2 个子空间, x_2 分为 4 个子空间,性能指标为 1.33)



显然模型 3-1 是稳定模型。

从最后结果不难看出,该方法可以得到与原系统几乎相同的前提参数。

应当指出,本节所介绍的 T-S 模糊模型的辨识算法,实际上假定了模型中每个逻辑关系结论函数的结构都是固定为所有输入变量的线性函数,即 $y=p_0+p_1x_1+\cdots+p_mx_m$ 的形式。但对一个实际系统而言,对某些关系式这种结构可能不是最好的,严格地说,对结论函数的结构也有辨识问题。事实上,结论函数的一般结构不仅可以假设为线性形式,而且还可以假定为非线性多项式等其他形式,从而可以用在 8.7 节所介绍的关于 NARMAX 模型辨识的双向回归递推辨识方法,同时辨识结论函数的结构和系数,进而得到最优模型。

另外,在实际应用中,充分利用系统的先验知识,对于简化建模过程、提高模型精度是很重要的。例如,在很多情况下,根据经验确定前提变量和模糊空间的划分,或者知道某些参数的大致范围,这将使辨识难度大大下降,而且容易得到更符合实际情况的模型。

12.3 应用实例

【例 12.8】 模糊建模在炼钢中的应用(Takagi T and Sugemo M,1985)

一般炼钢过程由以下 4 步组成:

- (1) 铁矿在高炉中熔化、去硫后,由鱼雷吊斗把熔化的铁(所谓的铁水(hot pig))转入炼钢炉;
- (2) 在炼钢炉中,首先向铁水中添加废料、铁矿和石灰,然后吹氧进行脱碳和除磷,之后再加入各种合金来调节成品钢的成分;
- (3) 取走漂浮的矿渣,进一步精炼,再次调整钢的成分含量;
- (4) 铸成钢锭,最后切割成合适的形状。

整个炼钢过程,除了第 4 步的最后半步外,每一步都是凭操作人员的经验来控制的。因为构建一个过程模型实在太困难了。炼钢厂可以生产成分不同的各种钢,尤其是对钢中的锰含量要求各有不同。不少人试图建立以可测量为输入、以成品钢中的锰含量为输出的数学模型,试验几乎都没有成功。

为了达到控制的目的,受经验丰富的操作人员的控制方式所启发,试着建立了一个控制器模糊模型。操作人员的经验可以告诉我们哪些可测变量应该选为模型的前提变量,这样得到

的模糊模型可能是他们的知识(经验)的提炼。因为模型的形式是“If...then”的形式,恰符合操作人员的思维和控制方式。一旦建立了模型,我们可以从模型得到模糊控制规则,这时,操作人员的控制行为也容易被定量解释,同时,也可通过操作人员的丰富经验来修正模糊控制规则。实际上,凡是操作人员起着关键作用的任何控制过程,都可以按此思路建立模糊模型。

根据操作人员的经验和研究,从过程涉及的 8 个变量中选择了如下 3 个变量作为炼钢炉模型的输入变量。它们分别是

$$HP(\text{输入铁的铁水含量})=\frac{\text{铁水}}{\text{输入铁}}=\frac{\text{铁水}}{\text{铁水}+\text{废料}}(\%)$$

$$MA(\text{输入铁中的锰合金含量})=\frac{\text{锰合金}}{\text{输入铁}}\times 10^{-1}(\%)$$

SG——漂浮的矿渣的状态

$$\text{输出变量为}\quad \Delta Mn=Mn_2-Mn_1\text{ (锰含量的增量)}$$

其中, Mn_1 是输入铁中锰的初始含量(比率), Mn_2 是成品钢中锰的最终含量(比率)。SG 可以量测,经验丰富的操作人员也可定量地观测出来,如软、中或硬。因此,只将 SG 放进前提中调节输入、输出关系,而在结论中不使用它。

从一个月的实际操作数据中取出 61 个用于炼钢过程的辨识,又选了 20 个数据作为测试数据,用来检查所得模型的有效性。部分数据由表 12.2 列出。

表 12.2 炼钢过程辨识的部分数据

HP	ΔMn	SG	MA
93.90	1.11	0.30	14.00
94.00	14.52	-0.08	135.00
93.10	5.22	-0.05	54.00
93.60	5.36	-0.43	53.00
85.60	13.56	0.11	129.00
85.80	11.55	0.12	106.00
86.40	11.88	-0.06	113.00
93.10	13.85	0.16	19.00
88.50	12.27	-0.02	029.00
93.00	11.69	-0.03	027.00

用本章给出的辨识算法,MA,SG 都被划分为两个子空间,而 HP 没被划分。最后得到由 4 个逻辑关系构成的模糊模型,如图 12.9 所示。表 12.3 中列出了前提变量、模型的性能指标及辨识过程的实际输出和模型输出的相关系数。

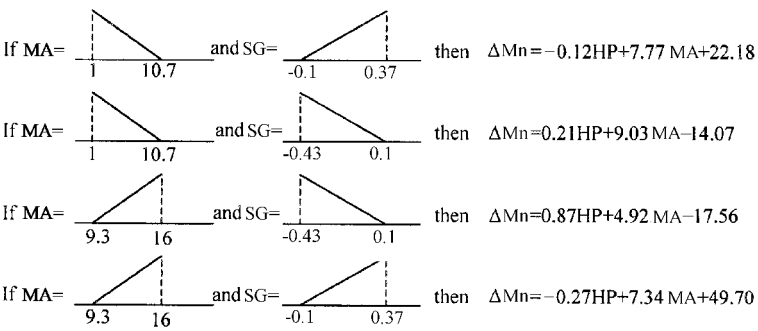


图 12.9 转炉控制模型

表 12.3 前提变量、性能指标及相关系数表

选出的前提变量		性能指标	相关系数
步骤 1	HP	5.20	0.98975
	MA(稳态)	4.87	0.99105
	SG	5.54	0.98840
步骤 2	MA-HP	4.17	0.99344
	MA-MA	4.23	0.99234
	MA-SG(稳态)	4.06	0.99378

从模型可以看出,这里每个前提变量只划分成了两个模糊子空间,这可能是由于数据太少的缘故。

用 20 个测试数据对模糊模型与通常的静态模型(由线性回归得到)

$$\Delta M_n = -0.24HP + 8.64MA + 45.6$$

进行了比较,所得的性能指标分别为:模糊模型为 7.15,静态模型为 7.77。总的效果要比静态模型好。

另外,从数据中得到的矿渣的状态——SG 的模糊划分,与操作人员结论相当一致,这对控制提供了重要信息。

下面将介绍如何将模型用于炼钢炉的控制。

在通常的控制中,只要给定了理想的输出 ΔM_n^* ,操作人员就可以凭其经验给出必要的输入 MA。为了简单起见,就用 MA 代表设计的控制器。沿着此思路,可以把理想输出作为模型的输出,由模型反解出输入 MA,用以控制。试验表明这种思路是可行的。

假定理想输出为 ΔM_n^* ,系统的实际输出记为 ΔM_n 。从实际生产记录的输入、输出数据得到的人工控制的平均相对误差为

$$AC_{\text{ope}} = \frac{1}{N} \sum_N (|\Delta M_n^* - \Delta M_n| / \Delta M_n^*) = 6.7\%$$

将实际输出 ΔM_n 作为理想输出(之所以用实际输出,是因为可得到与实际生产状况相关的各种数据,便于用模型计算),由模型反解计算出相应的系统输入 MA,再将该输入代入模型计算出模型的输出 $\Delta \hat{M}_n$,由模型得到的平均相对误差为

$$AC_{\text{mod}} = \frac{1}{N} \sum_N (|\Delta M_n - \Delta \hat{M}_n| / \Delta M_n) = 4.7\%$$

由此可见,由模型给出的操作完全可以比操作人员现在的控制结果需好。当然,也可以通过模型,由数据 $(\Delta M_n, HP, SG) \rightarrow MA$ 直接得到模糊控制律。

12.4 小 结

本章介绍的 T-S 模糊模型是描述复杂系统的一种很重要的形式。它的形式虽然简单,但它能描述高度非线性等一些非常复杂的现象。该模型包括了两种知识:一种是由模糊规则集 “If...then” 表示的定性知识;另一种是由局部模型(结论函数)所表示的定量知识。而且把定性和定量知识有机结合起来,特别适合于分层管理及多级控制的复杂系统的描述。例如,一个

两级控制系统,上层管理级处理高层计划,包括了系统的定性知识,很难表示为数学模型,因此可以基于规则及隶属函数来描述高层的定性知识,而底层一般执行基本的控制调节任务,因此可用数学模型定量描述。

又如,在石油化工生产过程中,原料油可能是不同产地的,如大庆油、胜利油、大港油等。由于各地油品的组成成分不同,因此生产中所采用的最优控制参数也不同。但这些原料油通常是混合使用的,如何设定最优控制参数问题是一个很困难的问题。如果采用 T-S 模型,用规则描述不同油品的划分,而用数学模型来定量描述控制参数的最优设定,则最优控制参数的自动设定问题将会得到满意的解决。

总之,T-S 模型可以作为很广的一类复杂系统的描述,因此得到了广泛的应用。但在将其用于复杂系统控制设计时,由于实际上它所表示的不是真正的动态模型,而且也无法用一般控制论知识来设计反馈控制器,因此,限制了它的实际应用。对此,Cao 等人推广了 T-S 模型(Cao S G and Rees N W,1995;Cao S G,Rees N W and Feng G,1997),他们用一组局部动态系统模型来描述每个子系统的动态特性,然后用隶属函数连接局部子系统模型,进而形成了一个全局动态系统,并给出了一类模糊动态系统的完整的辨识算法,在复杂控制系统设计中得到了广泛应用。详细介绍可参见有关文献。

习 题

1. 什么是概念模糊性?试说明模糊性所描述的不确定性与随机性所描述的不确定性有何不同?试举例说明。
2. 模糊集有哪几种表示方法?试举出 3 个不同表示的模糊集合。
3. 用图 12.4 所示的“不热”隶属函数和图 12.5 所示的“不冷”隶属函数,求出“温度适中”的隶属函数。
4. T-S 模型的模糊逻辑关系格式是什么?
5. 用 T-S 模糊模型描述系统有何特点?
6. 已知模型如例 12.7 中模型 2-2 所示,取 $x_1=5, x_2=6$,计算模型的最终输出 y 。
7. T-S 模型辨识的基本流程图是什么?
8. 用 C 语言编程,实现例 12.7 的辨识过程。

第 13 章 遗传算法及应用简介

13.1 引言

13.1.1 遗传算法的基本思想

遗传算法(Genetic Algorithm,GA)是由美国密西根(Michigan)大学,John H Holland 教授等创立的一类仿生型的优化算法(Holland J H,1975)。遗传算法的思想源于自然遗传。为了说明其基本思想,下面先看一个老鼠的例子。在任意给定的时刻,有一群老鼠,其中有一些老鼠比较机灵、敏捷,这些机灵、敏捷的老鼠不易被猫抓住,于是这些老鼠多数就会生存下来,当然一些反应迟钝的老鼠也可能活下去,但这仅仅是它们运气好而已。老鼠开始繁殖,这导致了老鼠的基因物质更好地结合,迟钝的和敏捷的、敏捷的和敏捷的、机灵的和笨的等相互结合,结果小鼠群将会比初始群体的老鼠更敏捷、更机灵。因为它们的双亲一般都是敏捷机灵的,都躲过了猫的追捕。这样一代一代传下去,每一代都比其父辈更机敏,其他生物也有类似过程。这就是自然选择中的“优胜劣汰、适者生存”的自然法则。

遗传算法就是对这种自然过程的模拟,将其抽象、简化,形成形式上的一种优化方法。它把问题的一个可能解表示为一个个体(Individual),若干个个体组成群体(Population,称为种群),每个个体用一个染色体(Chromosome)表示(注意,这里把一个个体和染色体等同起来,即每个个体只有一个染色体,这与实际生物有很大区别,比如每个人有 23 对(46 个)染色体),染色体由基本单元——基因(Gene)构成。一个基因控制一个或几个特征(如模型中的某个参数)的遗传,一定特征的基因被定位在染色体的一定位置上。通过在染色体上进行的遗传、进化、优胜劣汰,得到最优染色体,即最优解。简略地说,遗传算法是以达尔文的生物进化论和孟德尔遗传变异理论为基础,模拟生物界进化过程,自适应、启发式、全局优化的搜索算法。遗传算法瞄准的是那些复杂问题,它依赖于概率,又不同于随机算法,把启发性指导与随机算法有机结合,使其更具有鲁棒性;它的另一个特点是始终保持一组(而不是一个)可能的结果,从而保证可在全局范围内搜索,得到全局最优解,而不需要被优化的对象具有连续、可导等先验条件。总之,遗传算法在思路突破了以往最优化方法的框架,以其极强的解决问题的能力和广泛的实用性渗透到科研和国民经济的各个领域,如工程优化、神经网络、机器学习、机器人轨迹设计、故障诊断、作业调度、最优控制、模型辨识、模式识别、电路设计等领域,都得到了成功的应用。近几年来,其应用领域远远突破了控制工程的范畴,在生物、环保、社会、经济、金融、市场贸易等领域也引起了广泛的注意并得到成功的应用。本章只简单地介绍遗传算法的基本概念、工作步骤和一些应用,为进一步学习和研究打下最基本的基础。

13.1.2 基本遗传算法的工作步骤

基本遗传算法(Simple Genetic Algorithm,SGA)的结构和进化过程的结构一样,也是一种迭代结构。在迭代值为 t 时(常称为第 t 代),遗传算法保留一组可能的结果(一组染色体向量——第 t 代群体),记为 $P(t) = (x_1^t, \dots, x_n^t)$,每个染色体向量 x_i^t 都是问题的一个可能的结果,即一个可

能的解,每一个结果都赋以适应值。然后,通过选择更多的适应值高的个体,组成新的群体(即第 $t+1$ 次迭代),这个新群体的一些成员是经过交叉和变异后产生的新结果。交叉是指按一定概率通过交换双亲染色体的相应片段,把两个双亲染色体的特征结合起来,从而形成两个相似的后代。例如,双亲染色体是一个 5 维向量 $(a_1 \ b_1 \ c_1 \ d_1 \ e_1)$ 和 $(a_2 \ b_2 \ c_2 \ d_2 \ e_2)$,那么从第二个基因后交叉,得到两个后代染色体为 $(a_1 \ b_1 \ c_2 \ d_2 \ e_2)$ 和 $(a_2 \ b_2 \ c_1 \ d_1 \ e_1)$ 。交叉操作直观上就是在不同可能结果中交换信息。变异是指按一定的变异概率在选定的一个染色体上改变一个或多个基因(在基本遗传算法中,即 0 变为 1 或 1 变为 0)。变异操作直观上可以理解为引入新结果的辅助手段。因此,作为模拟生物进化过程的遗传算法,在解一个具体问题时应包括如下 5 个要素:

- 问题可能解的遗传表示(可能解表示为字符串,作为染色体(Chromosome));
- 建立可能解的初始群体(Population,种群);
- 评价函数(模拟环境),用以评价每个染色体所代表的个体的优劣;
- 遗传操作(Operators)(如 选择、复制、交叉、变异),用以改变后代染色体的结构(模拟基因重组);
- 遗传算法中的各种参数(如群体的大小、交叉概率、变异概率等)、模拟繁殖。

遗传算法的工作过程和具体步骤如下:

- (1) 初始化染色体群体;
- (2) 评价群体中每个染色体,得到每个染色体的适应值;
- (3) 根据适应值选择父母染色体;
- (4) 对父母染色体实施遗传操作产生后代(新染色体);
- (5) 对新染色体评价,并按照某种原则替换父母染色体,形成新一代群体;
- (6) 如果满足终止条件,则停止计算,并返回最好的染色体(问题的解);否则,再从步骤(3)开始继续。

以上算法可表示为如图 13.1 所示的流程图。

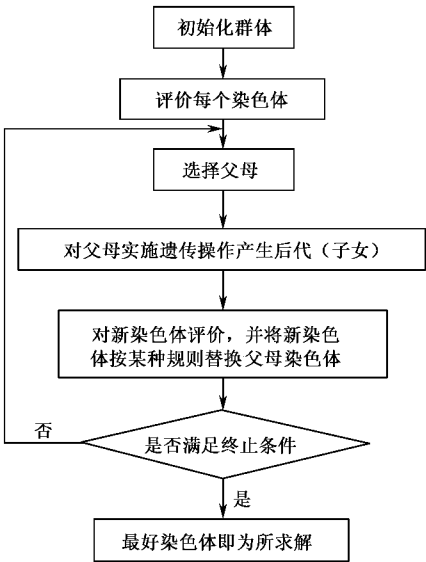


图 13.1 遗传算法流程图

13.2 遗传算法的计算机实现

前面给出了基本遗传算法的结构、工作过程和步骤,本节将分别介绍每一工作步骤是如何具体实现的。

1. 染色体的表示

为了表述更清楚,现结合下面的辨识问题具体说明。

假定已知模型结构为

$$y(k) = f(u(k), \theta) + e(k)$$

其中, $\theta = (a_1, a_2, \dots, a_n)^T$, $a_i \in D_i = [L_i, U_i]$, $i=1, 2, \dots, n$ 是待估参数。

已取得数据 $\{y(k), u(k), k=1, 2, \dots, N\}$, 求 θ , 使得如下目标函数(或称性能指标)

$$J(\theta) = \sum_{k=1}^N e(k)^2$$

达到最小。

在基本的遗传算法中,染色体采用二进制编码,即染色体表示为一串只含“0”和“1”的比特串。该问题中,是将参数向量 $\theta = (a_1, a_2, \dots, a_n)^T$ 表示为形如(1 0 0 1...0)的 0 和 1 的字符串,即先将每个参数 a_i 表示为 $\{0, 1\}$ 串,然后将其合并在一起构成参数向量的二进制编码,对应于问题的一个解。假定实变量 a_i 的值域为 $D_i = [L_i, U_i]$, 其编码是长度为 m 的 $\{0, 1\}$ 串: $[b_{m-1}, b_{m-2}, \dots, b_0]$ (其中 b_i 为 0 或 1), 则从二进制字符串 $[b_{m-1}, \dots, b_0]$ 映射到取值于区间 D_i 上的实数 a_i , 由如下两步完成:

(1) 将二进制字符串 $(b_{m-1}, b_{m-2}, \dots, b_0)$, 转换成十进制数

$$(b_{m-1}, b_{m-2}, \dots, b_0)_2 = \left(\sum_{i=0}^{m-1} b_i \cdot 2^i \right)_{10} = a'_i$$

(2) 得到相应实数

$$a_i = L_i + \frac{(U_i - L_i)}{2^m - 1} a'_i \quad (13.1)$$

这里编码长度 m 的大小依赖于变量的值和要求的精度。 m 是满足

$$2^{m-1} - 1 \leq (U_i - L_i) \times 10^k \leq 2^m - 1 \quad (13.2)$$

的最小整数,其中 k 是要求精度达到的小数点后的位数。

如 $x \in [-1, 2]$, 定义域长度为 3, 要求小数点后保留 6 位小数。则按精度要求需要将区间 $[-1, 2]$ 至少分成 3×10^6 个等长的区间段,这意味着一个实数的二进制向量需用 22 位。

$$2097152 = 2^{21} \leq 3 \times 10^6 \leq 2^{22} = 4194304$$

如果一个染色体是由 n 个实数的二进制表示构成的染色体,每一个实数的二进制表示的

长度分别为 $m_i (i=1, 2, \dots, n)$, 则该染色体的长度 $m = \sum_{i=1}^n m_i$ 。

本例中一个染色体代表了参数向量 θ 的一个可行解。

2. 初始化种群

随机赋予参数向量 θ 以 0, 1 串作为一个染色体。如果种群的规模为 pop , 则上述过程重

复 pop 次,即得到初始种群。

3. 评价函数与适应值 (Evaluation Function and Fitness)

评价函数起着模拟生存环境的作用,用来评价每个染色体的优劣。每个染色体的评价值为该染色体的适应值,一般适应值越大越好。评价函数与问题的目标有密切关系,在进化过程中起着启发指导性作用。例如,本例中评价函数可取为: $E(\theta)=10/(1+J(\theta))$,使目标函数值 $J(\theta)$ 越小的染色体,其适应值越高,其中 10 起放大倍数的作用,不是本质性的。

4. 父母的选择

在遗传算法中,选择父母的指导思想是试图模拟生物界的优胜劣汰、适者生存的自然法则。从总体而言,是使得群体中适应值越高的个体越有更多的生存和繁殖机会。已有许多可实施的选择方案,通常采用较多的是像轮盘赌(Roulette Wheel)式的随机选择,赌盘是按每个染色体的适应值来刻度的,即按每个染色体的适应值占总适应值的比例来划分赌盘,如图13.2所示。因此,此种选择方式也称为比例选择。在计算机上实现赌盘式选择的具体步骤如下:

- (1) 计算每个染色体 v_i 的适应值 f_i ;
- (2) 计算群体的总适应值 $F = \sum f_i$;
- (3) 计算每个染色体 v_i 的概率 $p_i = f_i/F$;
- (4) 计算每个染色体 v_j 的累积概率 $q_j = \sum_{i=1}^j p_i$ 。

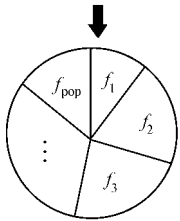


图 13.2 赌盘

选择过程是旋转 pop (群体的规模,也称容量)次赌盘,每次选出一个染色体作为父母。计算机上实现如下:

- ① 在 $[0,1]$ 区间产生一个随机浮点数 r ;
- ② 若 $r < q_1$,则选第一个染色体 v_1 ;否则选第 i 个染色体 v_i , v_i 满足 $q_{i-1} < r \leq q_i$ 。

显然,某些染色体可能不只被选一次,最佳染色体可能被选多次,一般染色体保持平衡,最差的可能被淘汰,这符合自然法则。

5. 交叉 (Crossover)

现在对选择出的新群体进行重组交叉。如前所说,遗传算法的一个系统参数就是交叉概率 p_c ,它决定了希望交叉的染色体的数量,最多可能有 $p_c \times pop$ 个染色体进行交叉操作。这里仅给出单点交叉过程。单点交叉按如下程序进行:

- (1) 对群体中的每个染色体,在 $[0,1]$ 区间产生一个随机数 r ,若 $r < p_c$,则该染色体被选出待进行交叉操作;
- (2) 将被选出的全部染色体自由配对,并对每对染色体在 $[1,m-1]$ (这里 m 是染色体的长度) 区间产生一个整型随机数作为单点交叉的交叉点;
- (3) 双亲染色体中位于交叉点以后的成分相互交换,产生两个子女。

如图 13.3 给出了一个单点交叉操作的说明。

交叉方式还有许多种,常用的有:两点交叉,即对选出待交叉的双亲染色体随机产生两个交叉点,交换双亲两个交叉点之间的元素;均匀交叉 (Uniform Crossover),即对选出待交叉的双亲染色体的每一个元素产生一个 $\{0,1\}$ 随机数,若为“1”,则交换双亲对应元素,否则不交换。

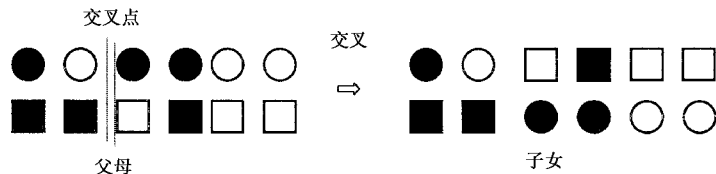


图 13.3 单点交叉

还有多点交叉,换位交叉等。

6. 变异 (Mutation)

群体经交叉操作后,再进行变异操作。基本遗传算法中的变异操作是按“比特”位(基因)进行的。每个“比特”位有相等的机会变异,即由 0 变为 1,或相反。如图 13.4 所示为变异的例子。

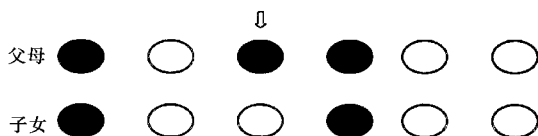


图 13.4 变异

每个“比特”位是否变异取决于变异概率 p_m ,它是遗传算法的另一个参数。它决定了所希望被变异比特的数目。整个种群可能有 $p_m \times m \times pop$ 个基因被变异。

计算机实现过程是:对每一个染色体的每一位基因产生一个 $[0,1]$ 区间上的随机数 r ,如果 $r < p_m$,则对该基因实施变异操作,否则不进行变异。

应该指出的是,前面所介绍的编码、遗传操作等只是对以二进制编码的遗传算法而言的。对于一般的遗传算法,编码方式可以是整数的、实数的、混合的、逻辑的等。根据不同问题可设计不同的编码,同时,问题不同也可设计不同的遗传操作,虽也称“编码”、“交叉”、“变异”,但具体实现过程可能大不相同,这一点从后面的例子中可以看到。

7. 参数的选择

参数的选择视问题而定,一般都需要通过试探、调整。对于二进制编码的遗传算法,种群规模一般取 $10 \sim 50$,种群规模越大,搜索到全局最优的可能性越大,但计算量也越大;在二进制编码的遗传算法中,交叉是进化的主要手段,因此交叉(概)率一般取得比较大,一般在 $0.5 \sim 0.9$ 之间;变异起着引进新结果和保持多样性的作用,它是对基因实施的一种操作,由于在二进制编码的遗传算法中,基因的数目很多,所以变异(概)率一般取得比较小,一般在 $0.001 \sim 0.01$ 之间,变异率取得越大,搜索到全局最优的可能性越大,但收敛越慢。

8. 终止条件

常用如下准则:

- (1) 规定最大遗传代数。
- (2) 评价值的方差小于规定阈值,即

$$\frac{1}{pop} \sum_{i=1}^{pop} (f_i - \bar{f})^2 \leq \epsilon, \quad \bar{f} = \frac{1}{pop} \sum_{i=1}^{pop} f_i$$

其中, ϵ 是预先设定的阈值。

(3) 最好的染色体连续若干次不变。

13.3 遗传算法的工作过程举例

下面通过一个仿真例子来说明基本遗传算法的整个工作过程。

【例 13.1】 求使函数

$$f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$$

达到最大的 x_1, x_2 的值。其中, $-3.0 \leq x_1 \leq 12.1$, $4.1 \leq x_2 \leq 5.8$ 。其函数图如图 13.5 所示。

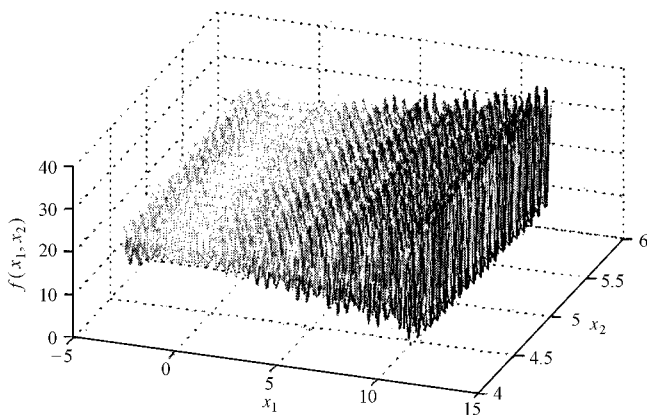


图 13.5 函数图形

用基本遗传算法求解该问题。假定采用的种群规模 $pop=20$, 交叉概率 $p_c=0.25$, 变异概率 $p_m=0.01$ 。下面将给出用遗传算法求解的详细过程。首先介绍如何对解进行编码, 即确定解的染色体表示。

1. 解的染色体表示

假定要求精度为小数点后 4 位, 变量 x_1 的值域长度为: $12.1 - (-3) = 15.1$, 对精度的要求意味着区间 $[-3.0, 12.1]$ 至少应该分成 15.1×10000 个等长的小区间。这即是说染色体的第一部分 (x_1 部分) 需用 18 位: $2^{17} < 151000 \leq 2^{18}$ 。变量 x_2 的值域长度为 1.7, 精度要求意味着区间 $[4.1, 5.8]$ 至少应该分成 1.7×10000 个等长的小区间。这即是说染色体的第二部分 (x_2 部分) 需用 15 位: $2^{14} < 17000 \leq 2^{15}$ 。于是染色体 (解向量) 的总长度 $m=18+15=33$, 前 18 位为 x_1 的编码, 后 15 位 (19~33) 是 x_2 的编码。

例如, 染色体为 (010001001011010000111110010100010), 前 18 位 010001001011010000 表示为

$$\begin{aligned} x_1 &= -3.0 + \text{decimal}(010001001011010000)_2 \cdot \frac{12.1 - (-3.0)}{2^{18} - 1} \\ &= -3.0 + 70352 \times \frac{15.1}{262143} \\ &= -3.0 + 4.052426 = 1.052426 \end{aligned}$$

后 15 位 111110010100010 表示为

$$\begin{aligned}
x_2 &= 4.1 + \text{decimal}(111110010100010)_2 \cdot \frac{5.8 - 4.1}{2^{15} - 1} \\
&= 4.1 + 31906 \cdot \frac{1.7}{32767} \\
&= 4.1 + 1.655330 = 5.755330
\end{aligned}$$

于是染色体(010001001011010000111110010100010)对应于 $(x_1, x_2) = (1.052426, 5.755330)$ 。该染色体的适应值为 $f(1.052426, 5.755330) = 20.252640$ 。

2. 初始化种群

染色体表示确定后,为了用遗传算法优化函数 f ,首先构造一个规模为20的初始种群,即每个染色体的所有元素均已随机初始化。假设初始化的种群为

```

v1 = (100110100000001111111010011011111)
v2 = (111000100100110111001010100011010)
v3 = (000010000011001000001010111011101)
v4 = (100011000101101001111000001110010)
v5 = (000111011001010011010111111000101)
v6 = (00010100001001010100101011111011)
v7 = (00100010000011010111101101111011)
v8 = (100001100001110100010110101100111)
v9 = (010000000101100010110000001111100)
v10 = (000001111000110000011010000111011)
v11 = (011001111110110101100001101111000)
v12 = (110100010111101101000101010000000)
v13 = (111011111010001000110000001000110)
v14 = (010010011000001010100111100101001)
v15 = (111011101101110000100011111011110)
v16 = (110011110000011111100001101001011)
v17 = (011010111111001111010001101111101)
v18 = (011101000000001110100111110101101)
v19 = (000101010011111111110000110001100)
v20 = (101110010110011110011000101111110)

```

3. 评价每个染色体

评价每个染色体时,先对每个染色体解码得 (x_1, x_2) ,再计算其适应值。于是得到

```

eval(v1) = f(6.084492, 5.652242) = 26.019600
eval(v2) = f(10.348434, 4.380264) = 7.580015
eval(v3) = f(-2.516603, 4.390381) = 19.526329
eval(v4) = f(5.278638, 5.593460) = 17.406725
eval(v5) = f(-1.255173, 4.734458) = 25.341160

```

$$\begin{aligned}
\text{eval}(\mathbf{v}_6) &= f(-1.811725, 4.391937) = 18.100417 \\
\text{eval}(\mathbf{v}_7) &= f(-0.991471, 5.680258) = 16.020812 \\
\text{eval}(\mathbf{v}_8) &= f(4.910618, 4.703018) = 17.959701 \\
\text{eval}(\mathbf{v}_9) &= f(0.795406, 5.381472) = 16.127799 \\
\text{eval}(\mathbf{v}_{10}) &= f(-2.554851, 4.793707) = 21.278435 \\
\text{eval}(\mathbf{v}_{11}) &= f(3.130078, 4.996097) = 23.410669 \\
\text{eval}(\mathbf{v}_{12}) &= f(9.356179, 4.239457) = 15.011619 \\
\text{eval}(\mathbf{v}_{13}) &= f(11.134646, 5.378671) = 27.316702 \\
\text{eval}(\mathbf{v}_{14}) &= f(1.335944, 5.151378) = 19.876294 \\
\text{eval}(\mathbf{v}_{15}) &= f(11.089025, 5.054515) = 30.060205 \\
\text{eval}(\mathbf{v}_{16}) &= f(9.211598, 4.993762) = 23.867227 \\
\text{eval}(\mathbf{v}_{17}) &= f(3.367514, 4.571343) = 13.696165 \\
\text{eval}(\mathbf{v}_{18}) &= f(3.843020, 5.158226) = 15.414128 \\
\text{eval}(\mathbf{v}_{19}) &= f(-1.746635, 5.395584) = 20.095903 \\
\text{eval}(\mathbf{v}_{20}) &= f(7.935998, 4.757338) = 13.666919
\end{aligned}$$

从以上评价价值不难看出, \mathbf{v}_{15} 是最好的一个染色体, \mathbf{v}_2 是最差的一个染色体。

4. 选择

为了选择, 先构造一个赌盘(Roulette Wheel)。

种群的总适应值为

$$F = \sum_{i=1}^{20} \text{eval}(\mathbf{v}_i) = 387.776822$$

每个染色体 $\mathbf{v}_i (i=1, 2, \dots, 20)$ 的选择概率 p_i 为

$$\begin{aligned}
p_1 &= \text{eval}(\mathbf{v}_1)/F = 0.067099 & p_2 &= \text{eval}(\mathbf{v}_2)/F = 0.019547 \\
p_3 &= \text{eval}(\mathbf{v}_3)/F = 0.050355 & p_4 &= \text{eval}(\mathbf{v}_4)/F = 0.044889 \\
p_5 &= \text{eval}(\mathbf{v}_5)/F = 0.065350 & p_6 &= \text{eval}(\mathbf{v}_6)/F = 0.046677 \\
p_7 &= \text{eval}(\mathbf{v}_7)/F = 0.041315 & p_8 &= \text{eval}(\mathbf{v}_8)/F = 0.046315 \\
p_9 &= \text{eval}(\mathbf{v}_9)/F = 0.041590 & p_{10} &= \text{eval}(\mathbf{v}_{10})/F = 0.054873 \\
p_{11} &= \text{eval}(\mathbf{v}_{11})/F = 0.060372 & p_{12} &= \text{eval}(\mathbf{v}_{12})/F = 0.038712 \\
p_{13} &= \text{eval}(\mathbf{v}_{13})/F = 0.070444 & p_{14} &= \text{eval}(\mathbf{v}_{14})/F = 0.051257 \\
p_{15} &= \text{eval}(\mathbf{v}_{15})/F = 0.077519 & p_{16} &= \text{eval}(\mathbf{v}_{16})/F = 0.061549 \\
p_{17} &= \text{eval}(\mathbf{v}_{17})/F = 0.035320 & p_{18} &= \text{eval}(\mathbf{v}_{18})/F = 0.039750 \\
p_{19} &= \text{eval}(\mathbf{v}_{19})/F = 0.051823 & p_{20} &= \text{eval}(\mathbf{v}_{20})/F = 0.035244
\end{aligned}$$

每个染色体 $\mathbf{v}_i (i=1, 2, \dots, 20)$ 的累计概率 q_i 为

$$\begin{aligned}
q_1 &= 0.067099 & q_2 &= 0.086647 & q_3 &= 0.137001 & q_4 &= 0.181890 \\
q_5 &= 0.247240 & q_6 &= 0.293917 & q_7 &= 0.335232 & q_8 &= 0.381546 \\
q_9 &= 0.423137 & q_{10} &= 0.478009 & q_{11} &= 0.538381 & q_{12} &= 0.577093 \\
q_{13} &= 0.647537 & q_{14} &= 0.698794 & q_{15} &= 0.776314 & q_{16} &= 0.837863 \\
q_{17} &= 0.873182 & q_{18} &= 0.912932 & q_{19} &= 0.964756 & q_{20} &= 1.000000
\end{aligned}$$

旋转赌盘 20 次(即随机产生 20 个 $[0,1]$ 区间的随机数),每次选出一个染色体进入新种群。假定从 $[0,1]$ 区间随机产生的 20 个随机数为

0.513870 0.175741 0.308652 0.534534 0.947628
0.171736 0.702231 0.226431 0.494773 0.424720
0.703899 0.389647 0.277226 0.348071 0.983437
0.005398 0.765682 0.646473 0.767139 0.780237

第一个数 $r = 0.513870$ 大于 q_{10} 且小于 q_{11} , 这意味着 \mathbf{v}_{11} 被选入新种群, 第二个数 $r = 0.175741$ 大于 q_3 且小于 q_4 , 这意味着 \mathbf{v}_4 被选入新种群, 依次类推, 最后由选出的染色体构成的新种群如下

$\mathbf{v}'_1 = (011001111110110101100001101111000)(\mathbf{v}_{11})$
 $\mathbf{v}'_2 = (100011000101101001111000001110010)(\mathbf{v}_4)$
 $\mathbf{v}'_3 = (00100010000011010111101101111011)(\mathbf{v}_7)$
 $\mathbf{v}'_4 = (011001111110110101100001101111000)(\mathbf{v}_{11})$
 $\mathbf{v}'_5 = (000101010011111111110000110001100)(\mathbf{v}_{19})$
 $\mathbf{v}'_6 = (100011000101101001111000001110010)(\mathbf{v}_4)$
 $\mathbf{v}'_7 = (111011101101110000100011111011110)(\mathbf{v}_{15})$
 $\mathbf{v}'_8 = (000111011001010011010111111000101)(\mathbf{v}_5)$
 $\mathbf{v}'_9 = (011001111110110101100001101111000)(\mathbf{v}_{11})$
 $\mathbf{v}'_{10} = (000010000011001000001010111011101)(\mathbf{v}_3)$
 $\mathbf{v}'_{11} = (111011101101110000100011111011110)(\mathbf{v}_{15})$
 $\mathbf{v}'_{12} = (010000000101100010110000001111100)(\mathbf{v}_9)$
 $\mathbf{v}'_{13} = (000101000010010101001010111111011)(\mathbf{v}_6)$
 $\mathbf{v}'_{14} = (10000110000111010001011010110011)(\mathbf{v}_8)$
 $\mathbf{v}'_{15} = (101110010110011110011000101111110)(\mathbf{v}_{20})$
 $\mathbf{v}'_{16} = (100110100000001111111010011011111)(\mathbf{v}_1)$
 $\mathbf{v}'_{17} = (00000111100011000001101000011101)(\mathbf{v}_{10})$
 $\mathbf{v}'_{18} = (111011111010001000110000001000110)(\mathbf{v}_{13})$
 $\mathbf{v}'_{19} = (111011101101110000100011111011110)(\mathbf{v}_{15})$
 $\mathbf{v}'_{20} = (11001111000001111110000110100101)(\mathbf{v}_{16})$

5. 交叉

现在对新种群的个体(向量 \mathbf{v}')进行交叉(重组)操作。交叉概率 $p_c = 0.25$, 即期望(平均)有 25%(即 5/20)的染色体进行交叉。按如下方式进行: 对新种群的每个染色体, 产生一个 $[0,1]$ 区间的随机数 r , 如果 $r < 0.25$, 则该染色体被选为进行交叉的染色体。

假定产生的随机数序列为

0.822951	0.151932	0.625477	0.314685	0.346901
0.917204	0.509760	0.401154	0.606758	0.785401
0.031523	0.869921	0.166525	0.674520	0.758400
0.581893	0.389248	0.200232	0.355635	0.826927

这意味着染色体 v'_2, v'_{11}, v'_{13} 和 v'_{18} 被选出进行交叉。这里选出的染色体的数目是偶数, 所以很容易配对。如果选出的染色体的个数是奇数, 则可随机增加一个或去掉一个。现在对选出的染色体配对: 前两个 (即 v'_2 和 v'_{11}) 和后两个 (即 v'_{13} 和 v'_{18}) 两两成对。对每一对染色体在区间 $[1, 32]$ 上 (33 是整个染色体的长度) 随机产生一个整数 pos 作为交叉点。

对第一对染色体产生的 $pos=9$

$v'_2 = (100011000 \mid 101101001111000001110010)$

$v'_{11} = (111011101 \mid 101110000100011111011110)$

交换双亲染色体第 9 个元素以后的成分产生后代, 并代替其双亲得

$v''_2 = (100011000 \mid 101110000100011111011110)$

$v''_{11} = (111011101 \mid 101101001111000001110010)$

第二对染色体产生的随机数 $pos=20$

$v'_{13} = (00010100001001010100 \mid 1010111111011)$

$v'_{18} = (11101111101000100011 \mid 0000001000110)$

交换双亲染色体第 20 个元素以后的成分产生后代为

$v''_{13} = (00010100001001010100 \mid 0000001000110)$

$v''_{18} = (11101111101000100011 \mid 1010111111011)$

种群现在变为

$v'_1 = (011001111110110101100001101111000)$

$v''_2 = (100011000101110000100011111011110)$

$v'_3 = (001000100000110101111011011111011)$

$v'_4 = (011001111110110101100001101111000)$

$v'_5 = (000101010011111111110000110001100)$

$v'_6 = (1000110001011010011111000001110010)$

$v'_7 = (111011101101110000100011111011110)$

$v'_8 = (000111011001010011010111111000101)$

$v'_9 = (011001111110110101100001101111000)$

$v'_{10} = (000010000011001000001010111011101)$

$v''_{11} = (1110111011011010011111000001110010)$

$v'_{12} = (010000000101100010110000001111100)$

$v'_{13} = (000101000010010101000000001000110)$

$v'_{14} = (100001100001110100010110101100111)$

$v'_{15} = (101110010110011110011000101111110)$

$v'_{16} = (100110100000001111111010011011111)$

$v'_{17} = (000001111000110000011010000111011)$

$v''_{18} = (111011111010001000111010111111011)$

$$\begin{aligned} \mathbf{v}'_{19} &= (111011101101110000100011111011110) \\ \mathbf{v}'_{20} &= (110011110000011111100001101001011) \end{aligned}$$

6. 变异

接下来是按比特位进行变异操作。变异概率 $p_m = 0.01$, 这意味着期望有(平均)1%的比特位变异。整个种群共有 $pop \times m$ (种群规模 \times 染色体长度) $= 20 \times 33 = 660$ 比特位, 我们期望有 6.6 个比特位变异, 种群中每一个比特位都有相同的机会变异。对每个比特位在 $[0, 1]$ 区间上产生一个随机数 r , 如果 $r < 0.01$, 则变异该比特位。

本例中, 产生 660 个随机数, 其中有 5 个小于 0.01, 其比特位和相应的随机数如表 13.1 所示。

表 13.1 比特位和相应的随机数表

随机数	0.000213	0.009945	0.008809	0.005425	0.002836
比特位	112	349	418	429	602

将比特位转换为对应染色体的编号及所在染色体内的基因位, 如表 13.2 所示。

表 13.2 染色体的编号及基因位表

比特位	112	349	418	429	602
染色体编号	4	11	13	13	19
基因位	13	19	22	33	8

从表 13.2 可以看出, 只有 4 个染色体受变异操作的影响, 第 13 个染色体有两个比特位变异。

最后得到的种群如下, 其中 \mathbf{v}_i 是新染色体

$$\begin{aligned} \mathbf{v}_1 &= (011001111110110101100001101111000) \\ \mathbf{v}_2 &= (100011000101110000100011111011110) \\ \mathbf{v}_3 &= (001000100000110101111011011111011) \\ \mathbf{v}_4 &= (011001111110010101100001101111000) \\ \mathbf{v}_5 &= (000101010011111111110000110001100) \\ \mathbf{v}_6 &= (100011000101101001111000001110010) \\ \mathbf{v}_7 &= (111011101101110000100011111011110) \\ \mathbf{v}_8 &= (000111011001010011010111111000101) \\ \mathbf{v}_9 &= (011001111110110101100001101111000) \\ \mathbf{v}_{10} &= (000010000011001000001010111011101) \\ \mathbf{v}_{11} &= (111011101101101001011000001110010) \\ \mathbf{v}_{12} &= (010000000101100010110000001111100) \\ \mathbf{v}_{13} &= (000101000010010101000100001000111) \\ \mathbf{v}_{14} &= (100001100001110100010110101100111) \end{aligned}$$

$$\begin{aligned}
\mathbf{v}_{15} &= (101110010110011110011000101111110) \\
\mathbf{v}_{16} &= (100110100000001111111010011011111) \\
\mathbf{v}_{17} &= (0000011111000110000011010000111011) \\
\mathbf{v}_{18} &= (11101111101000100011101011111011) \\
\mathbf{v}_{19} &= (111011111101110000100011111011110) \\
\mathbf{v}_{20} &= (110011110000011111100001101001011)
\end{aligned}$$

到目前为止,我们完成了整个遗传的一次迭代(即一代进化)。为了检验进化结果,对新种群的每个染色体进行评价,即对每个染色体进行解码为相应的实数值(x_1, x_2),并计算其相应的函数值。

$$\begin{aligned}
\text{eval}(\mathbf{v}_1) &= f(3.130078, 4.996097) = 23.410669 \\
\text{eval}(\mathbf{v}_2) &= f(5.279042, 5.054515) = 18.201083 \\
\text{eval}(\mathbf{v}_3) &= f(-0.991471, 5.680258) = 16.020812 \\
\text{eval}(\mathbf{v}_4) &= f(3.128235, 4.996097) = 23.412613 \\
\text{eval}(\mathbf{v}_5) &= f(-1.746635, 5.395584) = 20.095903 \\
\text{eval}(\mathbf{v}_6) &= f(5.278638, 5.593460) = 17.406725 \\
\text{eval}(\mathbf{v}_7) &= f(11.089025, 5.054515) = 30.060205 \\
\text{eval}(\mathbf{v}_8) &= f(-1.255173, 4.734458) = 25.341160 \\
\text{eval}(\mathbf{v}_9) &= f(3.130078, 4.996097) = 23.410669 \\
\text{eval}(\mathbf{v}_{10}) &= f(-2.516603, 4.39038) = 19.526329 \\
\text{eval}(\mathbf{v}_{11}) &= f(11.088621, 4.743434) = 33.351874 \\
\text{eval}(\mathbf{v}_{12}) &= f(0.795406, 5.381472) = 16.127799 \\
\text{eval}(\mathbf{v}_{13}) &= f(-1.811725, 4.209937) = 22.692462 \\
\text{eval}(\mathbf{v}_{14}) &= f(4.910618, 4.703018) = 17.959701 \\
\text{eval}(\mathbf{v}_{15}) &= f(7.935998, 4.757338) = 13.666916 \\
\text{eval}(\mathbf{v}_{16}) &= f(6.084492, 5.652242) = 26.019600 \\
\text{eval}(\mathbf{v}_{17}) &= f(-2.554851, 4.793707) = 21.278435 \\
\text{eval}(\mathbf{v}_{18}) &= f(11.134646, 5.666976) = 27.591064 \\
\text{eval}(\mathbf{v}_{19}) &= f(11.059532, 5.054515) = 27.608441 \\
\text{eval}(\mathbf{v}_{20}) &= f(9.211598, 4.993762) = 23.867227
\end{aligned}$$

注意,新种群总的适应值 F 是 447.049688,比上一代种群的总适应值(387.776822)高,而且最好染色体(\mathbf{v}_{11})的适应值(33.351874)也比上一代最好染色体(\mathbf{v}_{15})的适应值(30.060205)要高。继续不断进行选择、遗传操作、评价等迭代过程,经过 1000 次迭代后,得到的种群如下

$$\begin{aligned}
\mathbf{v}_1 &= (111011110110011011100101010111011) \\
\mathbf{v}_2 &= (111001100110000100010101010111000) \\
\mathbf{v}_3 &= (111011110111011011100101010111011) \\
\mathbf{v}_4 &= (111001100010000110000101010111001) \\
\mathbf{v}_5 &= (111011110111011011100101010111011) \\
\mathbf{v}_6 &= (111001100110000100000100010100001) \\
\mathbf{v}_7 &= (110101100010010010001100010110000)
\end{aligned}$$

$\mathbf{v}_8 = (111101100010001010001101010010001)$
 $\mathbf{v}_9 = (111001100010010010001100010110001)$
 $\mathbf{v}_{10} = (111011110111011011100101010111011)$
 $\mathbf{v}_{11} = (110101100000010010001100010110000)$
 $\mathbf{v}_{12} = (110101100010010010001100010110001)$
 $\mathbf{v}_{13} = (111011110111011011100101010111011)$
 $\mathbf{v}_{14} = (111001100110000100000101010111011)$
 $\mathbf{v}_{15} = (111001101010111001010100110110001)$
 $\mathbf{v}_{16} = (111001100110000101000100010100001)$
 $\mathbf{v}_{17} = (111001100110000100000101010111011)$
 $\mathbf{v}_{18} = (111001100110000100000101010111001)$
 $\mathbf{v}_{19} = (111101100010001010001110000010001)$
 $\mathbf{v}_{20} = (111001100110000100000101010111001)$

相应的适应值为

$\text{eval}(\mathbf{v}_1) = f(11.120940, 5.092514) = 30.298543$
 $\text{eval}(\mathbf{v}_2) = f(10.588756, 4.667358) = 26.869724$
 $\text{eval}(\mathbf{v}_3) = f(11.104627, 5.092514) = 30.316575$
 $\text{eval}(\mathbf{v}_4) = f(10.574125, 4.242410) = 31.933120$
 $\text{eval}(\mathbf{v}_5) = f(11.124627, 5.092514) = 30.316575$
 $\text{eval}(\mathbf{v}_6) = f(10.588756, 4.214603) = 34.356125$
 $\text{eval}(\mathbf{v}_7) = f(9.631066, 4.42788) = 35.458636$
 $\text{eval}(\mathbf{v}_8) = f(11.518106, 4.452835) = 23.309078$
 $\text{eval}(\mathbf{v}_9) = f(10.574816, 4.427933) = 34.393820$
 $\text{eval}(\mathbf{v}_{10}) = f(11.124627, 5.092514) = 30.316575$
 $\text{eval}(\mathbf{v}_{11}) = f(9.623693, 4.42788) = 35.477938$
 $\text{eval}(\mathbf{v}_{12}) = f(9.631066, 4.427933) = 35.456066$
 $\text{eval}(\mathbf{v}_{13}) = f(11.124627, 5.092514) = 30.316575$
 $\text{eval}(\mathbf{v}_{14}) = f(10.588756, 4.242514) = 32.932098$
 $\text{eval}(\mathbf{v}_{15}) = f(10.606555, 4.653714) = 30.746768$
 $\text{eval}(\mathbf{v}_{16}) = f(10.588814, 4.214603) = 34.359545$
 $\text{eval}(\mathbf{v}_{17}) = f(10.588756, 4.242514) = 32.932098$
 $\text{eval}(\mathbf{v}_{18}) = f(10.588756, 4.242410) = 32.956664$
 $\text{eval}(\mathbf{v}_{19}) = f(11.518106, 4.472757) = 19.669670$
 $\text{eval}(\mathbf{v}_{20}) = f(10.588756, 4.242410) = 32.956664$

通过若干次迭代(本例是 1000 次),得到最好的染色体,即为相应于问题的解。然而,如果仔细观察整个运行过程,就会发现,在早的一些代中,某些染色体的适应值比 1000 代后最好的染色体的适应值(35.477938)还好。例如,396 代中最好染色体的适应值为 38.827553,但由于随机选择和操作的原因,使最好的染色体没有成活下来,这对于优化来说,显然是不愿出现的。为了选出整个遗传过程的最佳染色体,一个通常最常用的方法,即所谓“精英策略”,即把当代

最好的染色体保留在某个特殊地方,并直接进入下一代,这样做,遗传算法可以给出整个进化过程的最佳解。但这样做的结果又可能出现另外的问题,即染色体过早地集中在“精英”附近,实际上没有达到全局最优解(得到的只是局部最优),这称为早熟收敛(Premature Convergence)。如何使算法既收敛快又避免早熟收敛,是遗传算法研究的重要问题之一。

【例 13.2】用遗传算法训练神经网络

问题仍如例 11.2 所述。神经网络的结构如图 11.10 所示。这里用二进制编码的基本遗传算法训练该神经网络。

1. 染色体的编码

将图 11.10 所示的神经网络的所有权值和阈值并在一起构成一个 n 维向量 w (这里权值和阈值共有 38 个,即 $n=38$)。

$$w = [th_1^2 m_{11}^2 \cdots m_{51}^2 \cdots th_5^1 m_{15}^1 \cdots m_{35}^1] \triangleq [\omega_1 \omega_2 \cdots \omega_{38}]$$

用二进制编码,即将参数向量 w 表示为只含“0”和“1”的字符串。本例要求精度为小数点后 4 位,每个权值的值域为 $[-100, 100]$,因此,每一个权值需用 21 位 $\{0, 1\}$ 串表示。因为 $2^{20} < 200 \times 10000 \leq 2^{21}$; 一个染色体的长度 $m = 38 \times 21 = 798$,每 21 位对应一个实数权 ω_i 的值,其相应权值的实数值由式(13.1)确定。

2. 评价函数取为

$$\text{eval}(\mathbf{v}) = 100 / (1 + E(\mathbf{w}))$$

其中染色体 \mathbf{v} 表示解码后的实数向量 w , $E(w)$ 由式(11.15)定义,式中 100 只起放大倍数作用,不是本质的。

3. 遗传操作

单点交叉、两点交叉和比特位变异。

4. 采用的参数

种群规模 $pop = 30$; 交叉概率 $p_c = 0.8$; 变异概率 $p_m = 0.001$ 。经过 5 次运行,每次迭代 10000 次,得到最好染色体的适应值为 5.287648。

训练好的神经网络,同样在例 11.2 给出的条件下进行实际实验,得到了比 BP 算法更好的结果。其输出图形将在图 13.6 中与实数编码遗传算法的结果一起示出。

13.4 实数编码遗传算法

二进制编码的遗传算法,其编码方式具有普遍意义,且一旦完成编码,则以后的遗传操作等运行程序几乎与实际问题的空间再无关系。因此,算法具有通用性,也便于理论分析,所以得到广泛的应用。但用二进制编码解决实际问题时,也存在明显的缺点,例如,由于编码长度所限和局部调整功能较弱造成精度不够高等。尤其是在应用于解多位、高精度的数学问题时,要达到要求精度,计算量相当大,甚至无法达到。例如,有 100 个在 $[-500, 500]$ 之间的变量,当精度要求达到小数点后 6 位时,用二进制编码表示的解向量的长度是 3000,由此产生的搜索空间将达 10^{1000} (Michalewicz Z, 1994),对这样的问题,二进制编码的遗传算法则无能为力。

而实数编码不受精度和值域的影响,即对任意精度要求和值域都不会增加编码长度和计算量。另外,实数编码有明显的物理意义,更接近实际问题空间,可根据实际问题需要和先验知识构造一些更有效的算子,以达到求最优解的目的。同时,用实数编码更便于处理一些非平凡约束的最优问题,这一点用二进制编码是很难实现的。

由于实数编码有诸多优点,因此在大量遗传、进化算法的研究与应用中,实数编码越来越受到重视。实践也表明,实数编码一般比二进制编码会得到更好的结果。

下面结合神经网络训练问题,介绍实数编码遗传算法常用的一些遗传操作,这些操作具有普遍意义。至于算法的运行过程,与基本遗传算法一样,也包括3个基本步骤:适应值计算、选择和繁殖(详见例13.1)。但实数编码的遗传算法用来繁殖的一些遗传操作与基本遗传算法大不相同,这是因为它们是在不同空间中进行的,基本遗传算法是在二进制空间,而实数编码遗传算法是在实数空间进行的。在实数空间中,解的编码为实数向量,为了提高算法的微调和寻优功能,一些操作是非均匀的。然而,由于这些遗传操作与二进制编码时的遗传操作直观上很相似,所以人们仍采用标准的操作分类,即变异和交叉。下面仍以训练神经网络为例,介绍实数编码遗传算法。

13.4.1 实数编码遗传算法及其在神经网络训练中的应用

这里不对实数编码遗传算法做全面系统的介绍,只是结合一个具体例子简单介绍一下实数编码遗传算法常用的基本遗传操作和实现方法。(王秀峰等,1998)

所考虑的问题仍如例11.2所述,神经网络的结构如图11.10所示。这里用实数编码的遗传算法训练该神经网络。

1. 染色体的编码

在图11.10所示的神经网络中,取所有权值和阈值组成 n 维(这里 $n=38$)实数向量

$$w = (th_1^2 m_{11}^2 \cdots m_{51}^2 \cdots th_5^1 m_{15}^1 \cdots m_{35}^1) \triangleq (w_1, w_2, \dots, w_{38}) \quad (13.3)$$

为染色体。换句话说,每个染色体的编码是与解向量同样长度的实数向量,每个元素的初始值都选在所指定的域内。

2. 评价函数

对于染色体 w 的评价函数取为

$$\text{eval}(w) = \frac{100}{1 + E(w)} \quad (13.4)$$

式中,染色体向量 w 是由式(13.3)所定义的实数向量, $E(w)$ 如式(11.15)所定义。

常用的遗传操作介绍如下。

3. 变异组

(1) 均匀变异(Uniform Mutation, UM)

它的定义与基本遗传算法类似,设 $w' = (w_1, w_2, \dots, w_n)$ 是一染色体,其中每一个元素 w_k 都有相等的机会变异,经变异后的向量记为 $w'^{+1} = (w_1, w_2, \dots, w'_k, \dots, w_n)$, $1 \leq k \leq n$,其中 w'_k 是由相应域 $[L_k, U_k]$ 中产生的随机数。简单地说,均匀变异就是用在相应域随机产生的实数替换染色体中被变异的实数。当然,每个元素是否被变异,也与基本遗传算法一样,由其对应

的变异概率所决定。

(2) 非均匀变异(Non-Uniform Mutation, NUM)

所谓“非均匀”是指操作的作用依赖于种群的代龄(即迭代次数)。其定义如下: 设 $w' = (w_1, w_2, \dots, w_n)$ 是一个染色体, 其中元素 w_k 被选为变异, $w_k \in [L_k, U_k]$, 变异结果为

$$w'^{t+1} = (w_1, w_2, \dots, w'_k, \dots, w_n), k \in \{1, 2, \dots, n\}$$

其中

$$w'_k = \begin{cases} w_k + \Delta(t, U_k - w_k), & \text{如果随机数取 } 0 \\ w_k - \Delta(t, w_k - L_k), & \text{如果随机数取 } 1 \end{cases}$$

这里函数 $\Delta(t, y)$ 返回 $[0, y]$ 区间的一个值, 而且随 t 值的增加, 接近零的概率也增加, 常取如下函数

$$\Delta(t, y) = y(1 - r^{(1-t/T)^b})$$

式中, r 是 $[0, 1]$ 区间的随机数, t 是种群的当时代龄, T 是最大代龄, b 是参数, 它决定了非均匀的程度, 通常取 $b=1$ 。

从 $\Delta(t, y)$ 的表达式可以看出, t 越小, 变异的作用越大, 随 t 值(代数)的增加, 变异的作用逐渐减少, 当 $t=T$ 时, 变异不起作用。

(3) 非均匀算术变异(Non-Uniform Arithmetical Mutation, MUAM)

这是一种具有很好局部调整功能的操作, 其定义如下: 设 $w' = (w_1, w_2, \dots, w_n)$ 是一个染色体, 其中元素 w_k 被选为经受该变异, 其结果为

$$w'^{t+1} = (w_1, w_2, \dots, w'_k, \dots, w_n)$$

式中, $w'_k = lw_k + (1-\alpha)\text{random}$, 这里 random 是从区间 $[L_k, U_k]$ 中产生的随机数, $w_k \in [L_k, U_k]$, α 是一个变量, 如取 $\alpha = b \cdot \frac{t}{T}$, 其中 $b \in [0, 1]$ 是参数, 用于调整变异的程度, b 越大, 变异的作用越小, 一般取 $0.8 \sim 0.9$; t 是种群当时的代龄, T 是最大代龄。显然有 $\alpha \in [0, b]$, 而且当 t 趋于 T 时, α 趋于 b 。由此可见, 随着种群代龄的增加, 该变异的作用也随之减少, 而 b 越接近于 1, 减少越快。

4. 交叉组

单点交叉形式上与基本遗传算法完全一样, 不再重述。另外, 被交叉染色体的选取, 以及各种交叉点的产生都与基本遗传算法产生的方法完全一样, 所以也不再重述。

(1) 两点交叉(Two-Points Crossover, TPC)

设 $w'_1 = (w_1^1, w_2^1, \dots, w_n^1)$ 和 $w'_2 = (w_1^2, w_2^2, \dots, w_n^2)$ 是被选出交叉的两个染色体, 随即选出的两个交叉点为第 i 个元素和第 j 个元素, 则交叉产生的后代是

$$w_1^{t+1} = (w_1^1, \dots, w_{i-1}^1, w_i^2, \dots, w_j^2, w_{j+1}^1, \dots, w_n^1)$$

$$w_2^{t+1} = (w_1^2, \dots, w_{i-1}^2, w_i^1, \dots, w_j^1, w_{j+1}^2, \dots, w_n^2)$$

即双亲的第 i 个元素至第 j 个元素之间的元素互相交换。而 i 和 j 是由 $[1, n]$ 区间上随机产生的两个整数。

(2) 两点算术交叉(Two-Points Arithmetical Crossover, TPAC)

设 $w'_1 = (w_1^1, w_2^1, \dots, w_n^1)$ 和 $w'_2 = (w_1^2, w_2^2, \dots, w_n^2)$ 是两个待交叉的染色体, 在第 i 点至第 j 点实施两点算术交叉, 所产生的后代为

$$w_1^{t+1} = (w_1^1, \dots, w_i', \dots, w_j', w_{j+1}^1, \dots, w_n^1)$$

$$w_2^{t+1} = (w_1^2, \dots, w_i'', \dots, w_j'', w_{j+1}^2, \dots, w_n^2)$$

其中向量 w_1^{t+1} 中从 w_i' 至 w_j' 间的每个元素 $w_k' (i \leq k \leq j)$ 由如下线性组合产生: $w_k' = \alpha w_k^1 + (1-\alpha)w_k^2$; 向量 w_2^{t+1} 中的从 w_i'' 到 w_j'' 间的每个元素 $w_k'' (i \leq k \leq j)$ 由式 $w_k'' = \alpha w_k^2 + (1-\alpha)w_k^1$ 确定。这里 w_k^1 和 w_k^2 分别是向量 w_1^t 和 w_2^t 中的相应元素。

该操作中,若 α 取常量,则称其为均匀算术交叉;如果 α 随种群的代龄而变(如在非均匀算术变异中的取法),则称其为非均匀算术交叉。

5. 参数的选择

在基本的遗传算法实现中,变异扮演着次要的角色。因为在基本遗传算法实现中,采用二进制编码,变异是用来挽回损失的比特值。由于二进制数据只有 0 和 1 两种状态,数据损失是很小的,所以其实现采用很小的变异概率。而用实数向量编码,种群中每个数据对应于向量中一个权值,所以它不可能充满权的整个数据域,因此,为了保证种群的多样性(个体间互异)和引入新的最佳解,变异是非常必要的,一般变异概率取值比二进制编码时要大。在本例中,取变异概率 $p_m = 0.01$;交叉概率 p_c 为 $0.8 \sim 0.9$,种群规模为 100。另外,父母染色体的选择是采用赌盘式(比例)选择来实现的。

13.4.2 结果与讨论

1. 结果

为了比较不同操作效果,训练该神经网络采用了由不同的操作组合构成的遗传算法的 4 种实现:GA1 由 TPC 和 UM 构成;GA2 由 TPC 和 NUM 构成;GA3 由 TPAC 和 NUM 构成;GA4 由 TPAC 和 NUAM 构成。对每种实现和每组参数都进行了 5 次随机运行并记录最好的结果。值得指出的是,对每种实现的多次运行结果之间标准差是很小的,几乎可以忽略不计。4 种算法得到的结果都比基本 GA 给出的结果要好,其中 GA4 的结果最好。如表 13.3 所示。

表 13.3 4 种算法的结果比较

算 法	基本 GA	GA1	GA2	GA3	GA4
最好适合值	5.287648	6.350414	6.555747	6.704008	6.741626

用于仿真的是 3HP 交流电机,用通常的具有 20 kHz 开关频率的 DTC 仿真,前 0.5s 转矩、定子电流和转子速度如图 13.6(a)所示。采用由 GA4 所训练的神经网络代替通常的状态选择器仿真得到的结果如图 13.6(b)所示,基本遗传算法的结果如图 13.6(c)所示。从图中不难看出,图 13.6(a)与图 13.6(b)几乎一样。

2. 讨论

① 用实数编码时,为了满足问题的约束(如所有的权值都必须在预定的域内),遗传操作需精心设计(在二进制表示中不存在这个问题,遗传操作的设计非常简单)。例如,本节设计操作时应用了凸空间的性质:即对任意两点 $w_i, w_j \in [L, U]$, 则其线性组合

$$\alpha w_i + (1-\alpha)w_j \in [L, U] \quad \alpha \in [0, 1]$$

② 在用实数编码时,为了产生后代,如果只用通常的交叉,即只交换两个染色体的部分元

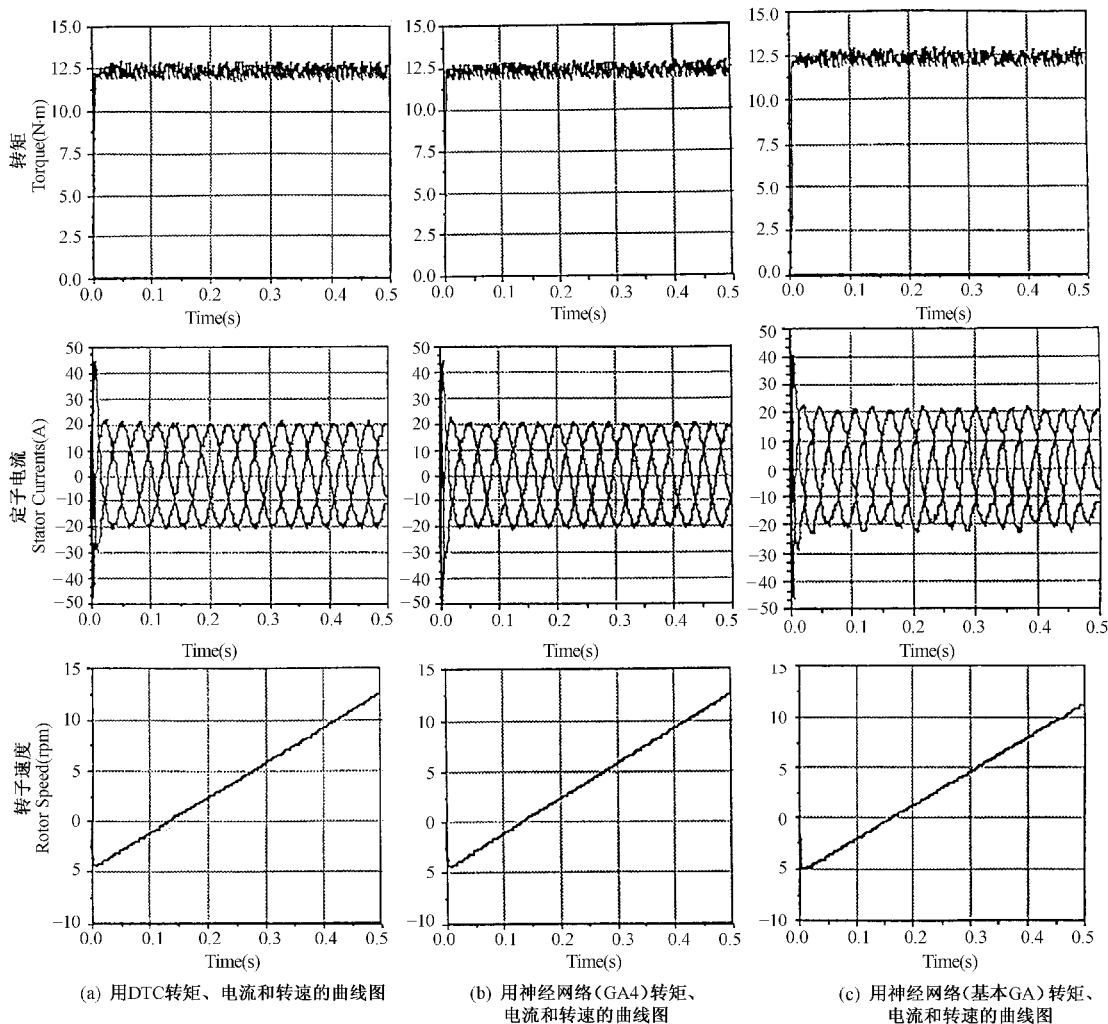


图 13.6 神经网络控制与实际电机控制的比较

素而相对位置不变,这实际上只是对参数组的合理组合起作用,而对参数本身并不起作用。在这种情况下,参数的搜索仅靠变异,如果变异概率取得太小,由于群体的容量有限,早熟收敛是不可避免的。因此,对于实数编码情况,为了避免早熟收敛,除了加大变异概率外,构造多种操作、使之不断产生新成员,从而保持种群的多样性是必要的。本节所介绍的非均匀算术交叉操作就是较有效的一种操作。相对于实数编码,二进制编码的交叉操作即包括了参数组合理组合的调整(当交叉点在两个参数编码之间时),又包含了对参数的优化搜索(当交叉点在某个参数的编码之中时),因此,交叉在进化中起着重要作用,所以交叉概率一般取值比较大,而变异概率取值很小。

③ NUM 和 NUAM 这两种操作都具有很好的微调功能。它们在开始时都是在空间中均匀搜索,而在接近收敛的步骤中越趋于局部调整。所不同的是,用 NUM 常常使解的某些元素位于搜索空间的边界,而 NUAM 则不然。

13.5 关于遗传算法的几点说明

13.5.1 模式定理

前几节主要是介绍 GA 是如何工作的,但它为什么能工作没有说明。这里简单介绍一下 Holland(1975)的模式(Schema)定理(Holland,1975; Goldberg,1989; Davis,1991),它在一定程度上解释了遗传算法的机理、数学特性,以及为什么有很强的计算能力。模式定理奠定了经典 GA 的理论基础。

(1) 模式(Schema)

基于三值字符集 $\{0,1,*\}$ 所产生的能描述具有某些结构相似性的 $\{0,1\}$ 字符串集的字串称为模式。其中符号“ $*$ ”表示可取 0 或 1 的任一值。

例如,长度为 5 的字符串,模式 $(*0001)$ 可匹配如下两个 $\{0,1\}$ 串: $\{(00001),(10001)\}$;模式 $(*001*)$ 可匹配如下 4 个字符串: $\{(00010),(10010),(00011),(10011)\}$ 。

(2) 模式的阶(Schema Order)

模式 H 中所有确定基因位的个数称为模式 H 的阶,记为 $O(H)$ 。

例如, $H_1=(011*1)$, $H_2=(**1*0)$, $H_3=(1****)$, 则 $O(H_1)=4$, $O(H_2)=2$, $O(H_3)=1$ 。模式的阶越高,确定性越高,其代表的样本数越少。

(3) 模式的定义矩(Defining Length)

模式 H 中第一个确定位置与最后一个确定位置之间的距离,称为模式的定义矩,记为 $\delta(H)$ 。例如, $\delta(H_1)=5-1=4$, $\delta(H_2)=5-3=2$, $\delta(H_3)=0$ 。

模式定理:在遗传算子选择、交叉和变异的作用下,具有低阶、短定义矩,以及平均适应值高于种群平均适应值的模式的生存数量,在子代中将以指数级增长。

建筑模块假说(Building Block Hypothesis):低阶、短定义矩、高平均适应值的模式(称为建筑模块),在遗传算子的作用下重组,能生成高阶、长矩、高平均适应值的模式,最终生成全局最优解。

模式定理保证了较优的模式样本呈指数级增长,从而满足了寻求最优解的必要条件,即存在找到最优解的可能性。建筑模块假说则指出了 GA 具备寻找最优解的能力。

13.5.2 GA 算法的收敛性分析

Goldberg 等人(Goldberg D E and Segrest P,1987;Rudolph,1994)用马尔柯夫链的理论对基本 GA 进行了收敛性分析,进一步从理论上分析了遗传算法的收敛性质。

(1) 马尔柯夫链的有关基本概念。

有限马尔柯夫链描述的是有限状态空间 $\Phi(|\Phi|(\text{基})=n)$ 的概率轨迹。例如,设状态空间共有 n 个状态,在第 t 步,从状态 $i \in \Phi$ 转移到状态 $j \in \Phi$ 的概率 $p_{ij}(t)$ 称为转移概率。如果转移概率 $p_{ij}(t)$ 与 t 无关,即对任意 i, j 和任意时刻 t_1, t_2 , 都有 $p_{ij}(t_1) = p_{ij}(t_2)$, 则称该马尔柯夫链是奇次的。

有限奇次马尔柯夫链可用转移概率矩阵 $P=[p_{ij}]$ 来表示, $p_{ij} \in (0,1)$, 且对任意 $i \in \Phi$, $\sum_{j \in \Phi} p_{ij} = 1$ 。

(2) 基本 GA 的马尔柯夫链分析

在基本 GA 中,采用二进制编码。如果种群的规模一定,不妨设为 n ,染色体的长度一定,不妨设为 l ,则状态空间的状态个数 $\Phi = 2^{l \cdot n}$ 。因为长度为 l 的串最多可以表示 2^l 个不同的点,每个状态中含有 n 个这样的点。不同的组合构成了 $2^l \cdot 2^l \cdot \dots \cdot 2^l = 2^{l \cdot n}$ 种可能状态。例如,串的长度为 2,种群规模为 2 的情况,共有 16 个状态。即

状态	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
X_1	00	00	00	00	01	01	01	01	10	10	10	10	11	11	11	11
X_2	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11

由遗传操作引起的基因改变的概率用状态转移矩阵 P 表示为: $P = C \cdot M \cdot S$, 其中 C, M, S 分别表示交叉、变异和选择的概率转移矩阵。这样就把遗传算法的收敛性问题转化为马尔柯夫链的收敛性问题来讨论。比较典型的分析结果是:基本遗传算法不能保证收敛到全局最优解,而具有“精英策略”的 GA 可以保证全局收敛。

13.5.3 GA 是一种优化算法的方法论

实际上遗传算法是一种优化算法的方法论,它提供的只是一种算法框架(Michalewicz, 1994)。其框架的每一步都可以有多种不同的实现方法,从而可构成多种不同的实现方案。基本遗传算法只是遗传算法最早实现的一种简便而典型的格式。换言之,在基本遗传算法中的各个步骤,即从编码、评价、选择、交叉、变异,到参数选择等各步中,都可根据不同的实际问题 and 理论观点,采用各种替代方案和改进方式来提高遗传算法的求解效率,以便更好地解决各种复杂的实际问题,有很广阔的创造空间。本章只是对遗传算法的一个简单介绍,只能作为遗传算法在建模中应用的一个入门基础。另外,GA 除了进一步扩大应用领域外,还有很多如带约束的最优化问题,多峰函数优化问题,并行计算问题,与神经网络、模糊逻辑等其他方法融合的问题,以及一些理论问题还需进一步研究。

习 题

1. 遗传算法的基本要素是什么?
2. 简述遗传算法的基本工作步骤。
3. 将二进制编码转换为相应的实数的步骤是什么?
4. 设变量 $x_1 \in [-5, 5], x_2 \in [1, 3]$, 要求精度为小数点后 6 位, 对 x_1, x_2 进行二进制编码, 试问染色体的长度应为多少?
5. 详细给出用计算机实现赌盘式(比例)选择的具体步骤。
6. 详细给出在基本 GA 中,用计算机实现单点交叉和均匀交叉的具体步骤。
7. 详细给出在基本 GA 中,用计算机实现变异的步骤。
8. 用 C 语言编程,分别用二进制编码和实数编码解例 13.1。
9. 用 C 语言编程,分别用二进制编码和实数编码实现对神经网络的训练。
10. 试举例说明变异操作二进制编码比实数编码搜索能力弱。
11. 试举例说明交叉操作二进制编码比实数编码搜索能力强。

第 14 章 辨识的实施

14.1 辨识的实验设计

本将要讨论在具体进行一个实际的系统辨识课题时会遇到的一些问题。这些问题在实际工作中往往是很重要的,但是到目前为止,对这些问题解决得并不是很好,真正适用的结果不多,理论分析也不甚完善,多数还是一些经验性的结果。尽管如此,这些结果有时候对于实际的辨识工作却是很有用的,下面将对这些问题做一些简单地讨论。

14.1.1 问题的提出

前几章中所处理的问题都是从已知数据出发对某种类型的模型进行结构或参数的辨识,在那些问题中,假定数据已经取到,至于数据是如何取来的是不讨论的,只是假定它们是离散采样值。

如果把前几章所做的工作做一个简单地概括,则可以说它们都是按照不同的要求、用不同的方式从数据中提取信息,例如模型的阶的估计是提取关于阶次 n 的信息,多变量线性系统结构是提取关于不变量 v_1, v_2, \dots, v_m 的信息,各种参数估计方法则是提取关于参数的信息。无疑地,这些信息都是由数据所带来的,因此数据是一切工作的基础。于是很自然会提出这样的问题:能不能通过改变取得数据的方式使得数据中能包含更多的信息,换句话说,取得更好的数据呢?

为了解决这个问题,首先要弄清楚什么数据是“好”的?也就是要给出一个判断数据中包含信息多少的标准。

当然,在实际中并不是都能按照最优的标准去设计取得数据的实验,有时候还需要讨论为了取得必要的信息至少要满足的条件,这一点在实际中可能更为重要。

在实际设计中,还必须考虑到实验条件所允许的限制,也就是说,必须满足一些约束条件,实际中可能遇到的一些典型的约束是:

- (1) 对输入和输出或中间变量的幅度限制;
- (2) 对输入和输出或中间变量的功率限制;
- (3) 实验所能使用的全部时间;
- (4) 采样总个数;
- (5) 最大采样速度;
- (6) 设备的其他限制。

这些条件究竟哪一个是重要的,要根据实际情况来决定,同时这些约束之间也是互相关联的。

14.1.2 设计准则

为了比较不同的实验,就必须对实验的“优良性”给出一个度量,这里提供的一种度量是与参

数估计的精度有关的,很自然地,一个好的实验提供的数据应该得到一个比较精确的参数估计。

我们知道参数估计量是一个随机变量或随机向量,因此它的精确度当然是用它们的方差或协方差矩阵来衡量。

在数理统计学中有下面的重要结果:对一次完成的采样,参数估计 $\hat{\theta}$ 的协方差阵有下界,即最好的估计量(有效估计量)的协方差阵达到下界。即

$$\text{COV}\hat{\theta} \geqslant \mathbf{M}^{-1} \quad (14.1)$$

其中

$$\mathbf{M} = E_{\theta} \left\{ \left[\frac{\partial \log p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \cdot \left[\frac{\partial \log p(\mathbf{x}; \boldsymbol{\theta})^T}{\partial \boldsymbol{\theta}} \right] \right\} \quad (14.2)$$

$E_{\theta}(\cdot)$ 表示关于 $\boldsymbol{\theta}$ 的数学期望, $P(\mathbf{x}; \boldsymbol{\theta})$ 是分布密度函数。

式(14.1)称为 Cramer-Rao 不等式(Wilks, 1962)。根据这个不等式就可能提出这样的度量:在某种意义下,使 \mathbf{M} 尽可能“小”。一般来说, \mathbf{M} 是一个矩阵,为了便于比较,可取 \mathbf{M} 的一个标量函数,即

$$J = E\phi(\mathbf{M}) \quad (14.3)$$

但是要注意到, \mathbf{M} 是与参数 $\boldsymbol{\theta}$ 有关的,因此要得到一个标量指标,就需要对 $\boldsymbol{\theta}$ 的某个先验分布求期望,这种对先验知识的依赖性必须注意的。

为了将式(14.3)进一步具体化,可以用一个标量函数 $S(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ 来衡量真实参数 $\boldsymbol{\theta}$ 和估计量 $\hat{\boldsymbol{\theta}}$ 之间的“差别”,从而通过它的值来表征估计量 $\hat{\boldsymbol{\theta}}$ 的好坏,因此可以考虑准则

$$J' = E(S(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})) \quad (14.4)$$

这里的数学期望是对样本和参数取的,当然这样的准则是很复杂的,下面进行一些简化得

$$\begin{aligned} J' &= E(S(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})) = E_{\theta} E_{x/\theta}(S(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(x))) \\ &\approx E_{\theta} E_{x/\theta}(S(\boldsymbol{\theta}, \boldsymbol{\theta})) + \frac{\partial S}{\partial \hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T + \frac{1}{2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \frac{\partial^2 S}{\partial \hat{\boldsymbol{\theta}}^2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ &= E_{\theta}(S(\boldsymbol{\theta}, \boldsymbol{\theta})) + \frac{1}{2} t_r \frac{\partial^2 S}{\partial \hat{\boldsymbol{\theta}}^2}(\mathbf{M}^{-1}) \end{aligned}$$

因此它就等价于

$$J'' = E_{\theta}(t_r \mathbf{W} \mathbf{M}^{-1}) \quad (14.5)$$

也就是 \mathbf{M}^{-1} 的加权和, \mathbf{W} 是加权矩阵。

另一个常用的准则是

$$J = -\log \det \mathbf{M} \quad (14.6)$$

当然还有其他准则,但比较能用的还是上述两个准则。

用于系统辨识的实验设计主要包括以下问题:

- (1) 输入信号的设计;
- (2) 采样区间的设计;
- (3) 为了补偿由于采样带来的信息损失而设计的预采样滤波器等。

虽然它的重要性已被认识到,但由于问题是很复杂的,至今真正有价值的结果并不多,特别是能付诸实际应用的成果就更少了,下面只能就一些简单的情形来说明解决这些问题的基本思路,讨论限于上述(1),(2)两类问题。

14.1.3 输入信号的设计

1. 持续激励输入信号

为了辨识一个动态系统的能控和能观测部分(即可通过输入、输出辨识的一部分),输入信号必须满足一定的条件,最少的一个要求是在观测周期内系统的动态必须被输入信号持续激励,这就引出了所谓持续激励输入信号,下面考虑两种情形。

首先讨论用相关分析确定离散时间脉冲响应 $h(t)$ (见第 7 章)的情况。在用这种方法时,要利用的基本关系式是(见第 7 章)

$$\varphi_{uy}(\tau) = \sum_{t=0}^{\infty} h(t) \varphi_{uu}(\tau - t) \quad (14.7)$$

式中, $\varphi_{uy}(\tau)$ 是输入 u 与输出 y 的互相关函数, $\varphi_{uu}(\tau)$ 是输入的自相关函数。对于充分大的 l , 式(14.7)可以近似表示为: $\varphi_{uy}(\tau) = \sum_{t=0}^l h(t) \varphi_{uu}(\tau - t)$, 利用这一基本输入、输出关系, 可得到下面的代数方程组

$$\begin{bmatrix} \varphi_{uy}(-p+l) \\ \vdots \\ \varphi_{uy}(-1) \\ \varphi_{uy}(0) \\ \varphi_{uy}(1) \\ \vdots \\ \varphi_{uy}(N) \end{bmatrix} = \begin{bmatrix} \varphi_{uu}(-p+l) & \cdots & \varphi_{uu}(-p) \\ \vdots & & \vdots \\ \varphi_{uu}(-1) & \cdots & \varphi_{uu}(-1-l) \\ \varphi_{uu}(0) & \cdots & \varphi_{uu}(-l) \\ \varphi_{uu}(1) & \cdots & \varphi_{uu}(1-l) \\ \vdots & & \vdots \\ \varphi_{uu}(N) & \cdots & \varphi_{uu}(N-l) \end{bmatrix} \cdot \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(l) \end{bmatrix} \quad (14.8)$$

或记为

$$\hat{\phi}_{uy} = \hat{\Phi}_{uu} h \quad (14.9)$$

对 $N=l, p=l$, 则 $l+1$ 个未知 $h(t)$ 可以从式(14.8)的 $l+1$ 个方程直接计算得到, 即

$$h = \hat{\Phi}_{uu}^{-1} \hat{\phi}_{uy} \quad (14.10)$$

对这种情形, 对输入信号的持续激励的要求是逆矩阵 $\hat{\Phi}_{uu}^{-1}$ 存在, 即

$$\det \hat{\Phi}_{uu} \neq 0 \quad (14.11)$$

由于这时 $\hat{\Phi}_{uu}$ 是对称矩阵, 所以式(14.11)成立的一个充分条件是

$$\varphi_{uu}(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} u(k) u(k+\tau) \quad (14.12)$$

存在。

下面再考虑离散线性系统的参数估计情形。在这种情形下, 模型是线性差分方程(见第 3, 4 章)

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_1 u(k-1) + \cdots + b_n u(k-n) + e(k) \quad (14.13)$$

如果用一次完成的最小二乘估计, 则有

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T y \quad (14.14)$$

其中, 参数向量是

$$\boldsymbol{\theta}^T = (-a_1, \dots, -a_n, b_1, \dots, b_n)$$

Φ 和 y 的结构见第 3 章。

矩阵 $\Phi^T \Phi$ 中的元素为 $\sum_{k=0}^N y(k-i)y(k-j)$, $\sum_{k=0}^N y(k-i)u(k-j)$ 和 $\sum_{k=0}^N u(k-i)u(k-j)$, 它们各自都被 N 除后, 分别是 $y(k)$ 的自相关函数 $\varphi_{yy}(i-j)$ 的估计 (记为 $\varphi_{yy}^N(i-j)$)、 $y(k)$ 与 $u(k)$ 的互相关函数估计 (记为 $\varphi_{uy}^N(i-j)$) 和 $u(k)$ 的自相关函数估计 (记为 $\varphi_{uu}^N(i-j)$)。于是得到

$$\begin{aligned} \frac{1}{N} \Phi^T \Phi &= \begin{bmatrix} \varphi_{yy}^N(0) & \varphi_{yy}^N(1) & \cdots & \varphi_{yy}^N(n-1) & -\varphi_{uy}^N(0) & \cdots & -\varphi_{uy}^N(n-1) \\ & \varphi_{yy}^N(0) & \cdots & \varphi_{yy}^N(n-2) & -\varphi_{uy}^N(-1) & \cdots & -\varphi_{uy}^N(n-2) \\ & & & \vdots & \vdots & & \\ & & & \varphi_{yy}^N(0) & -\varphi_{uy}^N(-n+1) & \cdots & -\varphi_{uy}^N(0) \\ \cdots & \cdots & \cdots & \cdots & \varphi_{uu}^N(0) & \cdots & \varphi_{uu}^N(n-1) \\ & & & & & \ddots & \\ & & & & & & \varphi_{uu}^N(0) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} = \mathbf{H} \end{aligned} \quad (14.15)$$

为了解方程 (14.15), 必须 $(\Phi^T \Phi)$ 是非奇异的。因此

$$\det \mathbf{H} \neq 0 \quad (14.16)$$

这时有两种可能的情形: $\det \mathbf{H} > 0$ 或 $\det \mathbf{H} < 0$, 而从自相关函数的性质可知

$$\begin{aligned} H_{11} &= \varphi_{uu}^N(0) > 0 \\ H_{22} &= \begin{vmatrix} \varphi_{uu}^N(0) & \varphi_{uu}^N(1) \\ \varphi_{uu}^N(-1) & \varphi_{uu}^N(0) \end{vmatrix} = (\varphi_{uu}^N(0))^2 - (\varphi_{uu}^N(1))^2 > 0 \end{aligned} \quad (14.17)$$

因为 H_{11} 和 H_{22} 都大于 0, 所以按照二次型定号的判别法, \mathbf{H} 不可能是负定的。因此, 如果式 (14.16) 满足, 则 \mathbf{H} 必须是正定的, 这就意味着 $\det \mathbf{H} > 0$, 并且所有顺序主子式的行列式 $\det H_1 > 0$ 。因此持续激励的条件是

$$\det \mathbf{H}_{22} > 0 \quad (14.18)$$

或者矩阵

$$\mathbf{H}_{22} = (\varphi_{uu}^N(i-j), i, j = 1, 2, \dots) \quad (14.19)$$

必须是正定的。

因此

$$\bar{u} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} u(k) \quad (14.20)$$

$$\varphi_{uu}(\tau) = \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} (u(k) - \bar{u})(u(k+\tau) - \bar{u}) \quad (14.21)$$

必须存在。

以上是对不同的估计问题得出输入持续激励的条件。

2. 最优输入信号

上面讨论的持续激励输入信号是对输入信号的最少的要求, 现在要讨论的是要对某个准

则来得到“最优”的输入信号,这种讨论当然就要复杂得多。

以单输入、单输出的滑动平均模型为例来说明解决问题的思路,考虑模型

$$y(k) = b_1 u(k-1) + \cdots + b_n u(k-n) + e(k) \quad (14.22)$$

式中, $\{e(k)\}$ 是 Gauss 白噪声序列,并具有相同的方差 σ^2 ,被估计的参数是

$$\boldsymbol{\theta}^T = (b_1, \cdots, b_n, \sigma) \quad (14.23)$$

从数理统计的一般知识可知,在这种情形下, Cramer-Rao 下界中的信息矩阵 \mathbf{M} 为

$$\mathbf{M} = \begin{bmatrix} \frac{1}{\sigma^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} & 0 \\ 0 & \frac{N}{2} \sigma^4 \end{bmatrix} \quad (14.24)$$

其中, N 是数据的个数

$$\boldsymbol{\Phi} = \begin{bmatrix} u(0) & u(-1) & \cdots & u(1-n) \\ u(1) & u(0) & \cdots & u(2-n) \\ \vdots & \vdots & & \vdots \\ u(N-1) & u(N-2) & \cdots & u(N-n) \end{bmatrix} \quad (14.25)$$

定义平均信息矩阵

$$\bar{\mathbf{M}} = \frac{1}{N} \mathbf{M} \quad (14.26)$$

即

$$\bar{\mathbf{M}} = \begin{bmatrix} \frac{1}{\sigma^2} \hat{\mathbf{R}} & 0 \\ 0 & \frac{1}{2} \sigma^4 \end{bmatrix} \quad (14.27)$$

其中

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{k=1}^N \begin{bmatrix} u^2(k-1) & \cdots & u(k-1)u(k-n) \\ \vdots & & \vdots \\ u(k-1)u(k-n) & \cdots & u^2(k-n) \end{bmatrix} \quad (14.28)$$

现在的问题是选择输入序列 $\{u(1), \cdots, u(N+n-1)\}$, 使得准则

$$J = -\log \det \bar{\mathbf{M}} \quad (14.29)$$

达到最小。对输入的约束考虑如下的功率限制

$$\frac{1}{N} \sum_{k=1}^N u^2(k-i) = 1, i = 1, \cdots, n \quad (14.30)$$

在这些条件下有下面的结论:在约束式(14.30)之下,当 $\hat{\mathbf{R}}$ 的非对角线元素为 0 时,准则式(14.29)达到极小。

要证明这一点,只需用到下面的事实:对角线元素全为 1 的正定矩阵 \mathbf{M} 的行列式,当 $\mathbf{M} = \mathbf{I}$ 时达到极大。

于是,最优输入信号必须满足

$$\frac{1}{N} \sum_{k=1}^N u(k-i)u(k-j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (14.31)$$

式(14.31)可粗略地理解为输入信号 u 在不同时刻是不相关的。

剩下的问题是如何找出输入序列 $u(1), \cdots, u(N+n-1)$ 满足式(14.31)。这个问题就不

在这里详细地讨论,只是做一简单的说明:当 N 比较大时,式(14.31)是可以由白噪声序列来实现的。因为 N 充分大时,式(14.31)右端接近自相关函数 $\varphi_{uu}(\tau)$,而白噪声的自相关函数 $\varphi_{uu}(\tau)$ 是满足式(14.31)的。对有限值 N ,伪随机二位式序列可以非常近似地满足式(14.31)(见第7章的讨论)。对某些 N 值,可以找到严格满足式(14.31)的二位式输入序列,它比用伪随机二位式序列输入所估计出来的方差还要小(Goodwin and Payne, 1977)。

输入信号的设计是一个重要的问题,但是由于问题比较复杂,所以除了一些较简单的情形外,还没有更多适用的结果,特别是最优设计困难就更多了。

14.1.4 采样区间的设计

1. 一般性讨论

对离散时间模型的辨识,在实验开始以前必须适当选择采样区间(间隔),因为一旦采样信号被存储起来,就不可能再回去缩短采样区间了。当然,用删去对应数据的办法,采样区间可以加倍、3倍等。采样区间的选择主要依赖于:

- 最终应用的采样区间(如数字控制);
- 所要得到的模型的精度;
- 如果采样区间 T 太小,可能引起数值计算的问题。

下面对这几个问题做一些简短的讨论。

对于单输入、单输出离散线性系统模型式(14.13),表14.1中给出了不同采样间隔参数辨识的结果。

表 14.1 不同采样区间的参数辨识结果

$T(s)$	1	4	8	16
a_1	-2.48824	-1.4086	-0.83771	-0.30842
a_2	2.05387	0.70409	0.19667	0.02200
a_3	-0.56203	-0.09978	-0.00995	-0.00010
b_0	0	0	0.06525	0.37590
b_1	0.00462	0.06525	0.25598	0.32992
b_2	0.00169	0.04793	-0.02850	0.00767
b_3	-0.00273	-0.00750	-0.00074	-0.00001
d	4	2	1	1
$\sum b_i$	0.00358	0.10568	0.34899	0.71348
$1 + \sum a_i$	0.00358	0.10568	0.34899	0.71348

对一个小采样区间,如 $T = 1s$ 时,有 $|b_i| \ll |a_i|$ 以及 $|\sum b_i| = |1 + \sum a_i| \ll |a_i|$,这样参数的很小误差可能在模型的输入、输出特性上有显著影响,因为参数 b_i 在小数点以后第4,5位的值都会影响到 $\sum b_i$ 。反过来说,数据的微小误差也将影响参数大的变化。

另一方面,采样区间如果选得太大,动态性质的描述就变得不精确了,对 $T = 8s$,模型实际上退化成二阶,因为 $|a_3^8| \ll |1 + \sum a_i^8|$, $|b_3^8| \ll |\sum b_i^2|$,而对 $T = 16s$,模型甚至可以退化为一阶。

(1) 如果辨识的最终目的是设计一个数字控制系统,采样区间必须按照数字控制系统的

性能来选择,许多因素都可能影响到采样区间的选择,如控制性能、控制算法的类型、干扰的特性、其他硬件的性能等。

(2) 从上面的例子说明,由于所得到的模型的精确性问题,采样时间不能选得太小,也不能选得太大。

(3) 如果采样时间选得太小,数据的相关程度明显地增加,就会使得求参数估计值时出现病态方程组,产生计算上的困难。

(4) 上面只是对采样时间对辨识结果的影响做了一些简短的讨论。尽管采样时间对辨识结果有明显的影 响,但是在实际应用中也不是十分严格的,也就是说,还是允许在一定的范围内变化的。

一个经验的规则是根据系统的过渡时间来确定采样区间,令 T_{95} 是系统过渡时间的 95%, 则一般取采样区间 T , 使得

$$T_{95}/T \approx 5, \dots, 15$$

也就是说,在 T_{95} 这段时间内进行 5~15 次采样。

当然这是经验的办法,为了更精确地决定采样时间,也可以按照某一准则来确定“最优”的采样区间。

2. 最优采样区间的设计

用于系统辨识的实验设计,不仅能设计输入信号,而且也能设计使得某个准则达到极小的采样区间。

现在考虑的问题是:

- (1) 用 N 个样本来估计连续时间线性系统中的参数;
- (2) 在时间 t_1, t_2, \dots, t_N 上被采样,这里采样区间 $\Delta_k = t_{k+1} - t_k$ 不必是相等的;
- (3) 可以调整的实验条件是
 - ① 系统输入;
 - ② 采样时刻 t_1, \dots, t_N ;
 - ③ 预采样滤波器(假定是线性定常的)。

根据具体条件,如模型的形式、噪声特性的假定等,可将信息矩阵 M 表示成输入序列 $(u(0), u(1), \dots, u(N-1))$ 、采样间隔 $(\Delta_0, \Delta_1, \dots, \Delta_{N-1})$, 以及预采样滤波器的函数,然后将 $\log \det M$ 关于这些设计变量极小化。这样做在原则上是可能的,但是,即使对简单情形,这样做计算也是很复杂的,在这里也就不做详细讨论。对于一般的应用,按照经验法确定采样区间也可以了。

总之,对系统辨识而言,经过适当设计的实验所提供的数据将会提供更多的有用信息,在这些设计中,有一些是为了能够完成辨识而必不可少的,如持续激励信号,另一些是为了提高辨识精度(这种精度是用使某个准则极小来体现的)而需要进行的设计。对一个实际的辨识课题来说,前者是重要的,是必不可少的;而后者能提高辨识精度,但是却要进行较复杂的设计工作,在一些实验费用昂贵、时间短、要求精度高的问题(如飞机动态参数的辨识)中,考虑这样的设计是合适的。

值得说明的是:辨识与控制对系统的要求是矛盾的。辨识要求控制信号持续激励,把系统充分扰动起来,这样得到的数据才能充分代表系统的全部特征;而控制则是要求输入使系统平稳,而抑制扰动。

14.2 模型适用性检验

通过辨识得到的模型,虽然在选定的模型类中是最好的(与实际系统符合程度最好),但是并不等于所得到的模型是适用的,也就是说满足建模目的需要。对模型的适用性进行检验是辨识的重要环节之一,只有通过适用性检验的模型才是最终的模型。造成模型不适用的原因主要来自3个方面:模型类(模型的结构)选择不当;实验数据误差过大;辨识算法存在问题(如没有考虑必要的约束)。

14.2.1 通过先验知识的检验

通过先验知识检验是一种重要的检验方法。因为有些模型从数据的拟合上看不出问题,但是根据对模型已有的知识却可以断定模型是否适用,特别是在估计一些有特定物理意义的参数时,这种检验特别重要。如在第9章中,用肾功能模型估计肾血流量时,对所估计参数的大致范围是知道的,如果偏离太大就不行了。此外,还可以用特殊情形的数据(如肾切除病人的数据)检验模型是否正确。又如在第8章中估计化学反应动力学参数时,通过非线性最小二乘得到的模型,反应级数是负的,从已有的知识知道这是不合理的。虽然所得到的模型对数据的拟合误差最小,但是不能用。

这里也告诉我们,在建模和辨识中,明确建模的目的并且尽可能多地了解系统的有关知识是十分重要的。

14.2.2 通过数据的检验

用数据进行适用性检验有两种:在同一模型类中进行比较和在不同模型类中进行比较。

① 通常在得到一个模型后,总是希望用另一组数据(不参加辨识的)去验证模型的精度,看看是否是接受的。这组数据的作用类似于GMDH方法中的检验组,如果这种检验是不成功的,则可能存在两个问题:辨识所用数据缺乏代表性,也就是说,在一个很窄的变化范围内取得的数据,包含的信息量不足;或所选的模型类不合适。

② 用同一组数据对不同的模型类进行比较。这种比较常用的有两种方法,一种是统计检验的方法,如用F检验或似然比检验(Astrom and Eykhoff, 1970);另一种是在误差平方和上加惩罚项,这种做法的典型例子是赤池的AIC准则(见5.1节),在这个准则中,附加的惩罚项 $2p$ 是模型所包含的参数个数。

除了以上两种数据检验方法外,还常采用一些已知意义的特殊的数据(如边界的数据、极端的数据等)来检验,看它是否符合已知的物理意义。

14.2.3 通过对预测误差序列的检验

如果用数据估计得到模型及参数 $\hat{\theta}$,用这个模型预测系统的输出 y 得 $\hat{y}(t, \hat{\theta}, Z^{t-1})$,其中 Z^{t-1} 是到 $t-1$ 时刻为止的输入、输出数据,则可得预测误差为

$$\varepsilon(t, \hat{\theta}) = y(t) - \hat{y}(t, \hat{\theta}, Z^{t-1}) \quad (14.32)$$

这就是所谓“新息”序列。在十分广泛的条件下,“新息”序列应为一独立的随机序列,所以可以通过检验新息序列是否是白噪声(白性检验)来检验所得到的模型是否合适。

模型的适用性检验是一个十分重要的问题,是完成一个辨识过程必不可少的组成部分。这种检验既有客观标准,又带有很大的主观性。到目前为止,还没有很多方法(严格的数学方法)去进行客观的检验,甚至“适用性”概念本身也还是主观的,所以多数情形还是就具体问题去寻找具体的解决办法。

在许多辨识课题的研究中,都要涉及到模型的适用性研究,但是对这个问题的专题研究并不多,有兴趣的读者可能参见 Ljung(1981)。

14.2.4 应用实例

【例 14.1】葡萄糖消失模型的适用性检验

葡萄糖的利用是人体的一种重要的代谢功能,在对这种代谢功能进行量化时,模型的作用是很大的(如估计葡萄糖的利用、胰岛素的分泌等)。到目前为止,至少已经有 15 个以上关于血浆葡萄糖调节系统的模型,对这些模型如何进行评价呢?可以通过检验模型的适用性来选择最合适的模型。

下面共考察 7 个常用的模型,具体结构如图 14.1 所示。

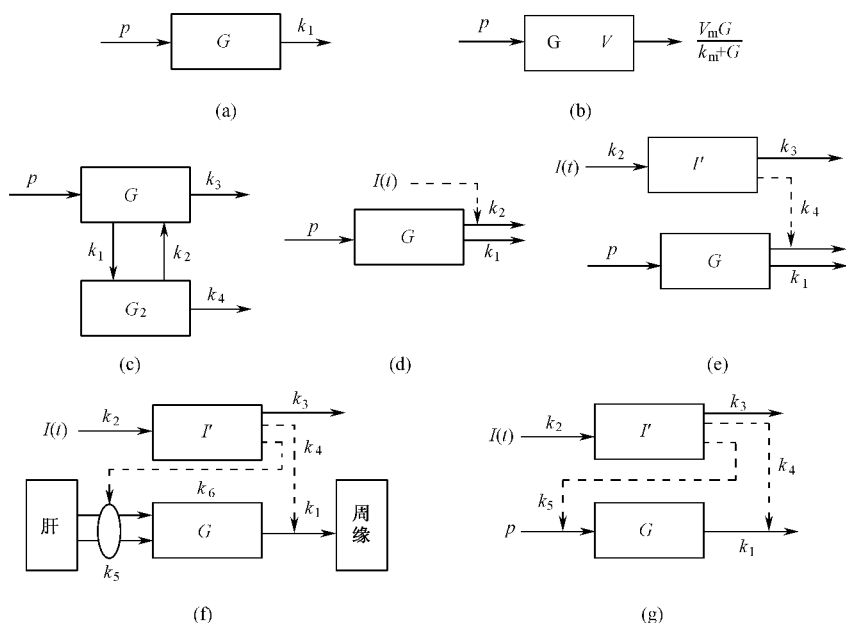


图 14.1 血浆葡萄糖调节模型

在图 14.1 中, G 表示血浆葡萄糖浓度; G_2 表示血浆外葡萄糖浓度; I 表示对血浆胰岛素浓度的基础值的偏差; I' 表示血浆外胰岛素浓度的偏差; p 表示肝生产的葡萄糖; k 表示分流动速度; V 表示葡萄糖体积; V_m, k_m 为 Michaelrs-Menten 参数。

数学模型是

$$\text{模型 (a)} \quad \frac{dG}{dt} = -k_1 G + p \quad G(0) = G_{ss}$$

$$\text{模型 (b)} \quad \frac{dG}{dt} = -\frac{1}{V} \frac{V_m G}{k_m + G} + p \quad G(0) = G_{ss}$$

$$\text{模型 (c)} \quad \frac{dG}{dt} = -(k_1 + k_3)G + k_2 G_2 + p \quad G(0) = G_{ss}$$

$$\frac{dG}{dt} = k_1 G - (k_2 + k_4) G_2 \quad G_2(0) = 0$$

$$\text{模型(d)} \quad \frac{dG}{dt} = -k_1 G - k_2 I + p \quad G(0) = G_{ss}$$

$$\text{模型(e)} \quad \frac{dG}{dt} = -k_1 G - k_4 I' + p \quad G(0) = G_{ss}$$

$$\frac{dI'}{dt} = -k_3 I' + k_2 I \quad I'(0) = 0$$

$$\text{模型(f)} \quad \frac{dG}{dt} = -[(k_1 + k_2) - (k_4 + k_6) I'] G + B_0 \quad G(0) = G_{ss}$$

$$\frac{dI'}{dt} = -k_3 I' + k_2 I \quad I'(0) = 0$$

式中, B_0 是在 $G=0$ 时肝产生的葡萄糖的推断值。

$$\text{模型(g)} \quad \frac{dG}{dt} = -[(k_1 + k_4 I')] G + \frac{p}{k_5 + I} \quad G(0) = G_{ss}$$

$$\frac{dI'}{dt} = -\frac{k_2}{k_3} I' + I \quad I'(0) = 0$$

按前面讨论的原则, 利用房室模型的参数估计对这 7 个模型进行检验, 分析结果如下:

模 型	先验可 辨识性	后验可 辨识性	参数值 物理可实现性	拟合精度	可接受性
(a)	整体可辨识	可	可	不可	不
(b)	整体可辨识	不可	不考虑	不考虑	不
(c)	整体可辨识	不确定	不可	不考虑	不
(d)	整体可辨识	可	不确定	不可	不
(e)	整体可辨识	可	可	可	可
(f)	整体可辨识	可	可	可	可
(g)	局部可辨识可	不可	不考虑	不考虑	不

注: 前一项检验被否定, 后面的检验就不再考虑。

从分析的结果看出, 只有模型(e)和(f)是可接受的。最后再从生理学的角度来考察这两个模型: 模型(f)包含了胰岛素对肝生产葡萄糖的抑制, 表现了纯肝的葡萄糖平衡。而在模型(e)中, 假定肝葡萄糖是常产率, 因此(f)与葡萄糖代谢的已知生理更为一致, 所以最后认为模型(f)是适用的。值得注意的是, 这样的选择将常用的模型(a)和(d)淘汰了。

从这个例子可以看出, 模型适用性检验是一个很具体的问题, 但是还缺乏一般理论的指导。

习 题

1. 系统辨识的实验设计主要包括哪些内容?
2. 为了辨识, 输入信号必须满足的条件是什么? 辨识中常用的输入序列有哪些?
3. 辨识和控制对输入序列要求有何不同?
4. 采样区间的大小对参数辨识有什么影响?
5. 确定采样区间的经验规则是什么?
6. 造成辨识得到的模型不适用的原因主要来自哪些方面?
7. 通常用哪些方法检验模型的实用性?

附录 粤瑶矩阵运算的两个结果

有理矩阵求逆引理

摇摇若

孕^韻 越孕^韻 垣匀^韻 砸匀^韻 (粵^韻)

则

孕越孕 原孕匀(匀孕匀垣砸)愿孕匀孕 (粤趣)

摇摇证明摇等式(粤)两端先同左乘以孕,而后右乘以孕匀得

孕_員勻越孕_員垣孕_員砸_員勻孕_員勻
越孕_員砸_員(勻_裁孕_員勻垣砸_員)

再用(匀孕匀垣) 匀孕 右乘以上式两端得

孕勻裁勻韻垣韻勻韻越韻孕韻勻韻 (粵韻)

摇摇由式(粵)得：孕^原原孕^原越匀^原匀^裁，所以

孕^原孕^原孕^原越孕^原孕^原孕^原孕^原越孕^原孕^原孕^原孕^原 (粵韻)

摇摇将式(粤獾)代入式(粤獾)后移项得

孕越孕_员 原孕_员匀(匀_栽孕_员匀垣砸)_原匀_栽孕_员

圆爱矩阵分块求逆运算

考虑如下矩阵

粤越 $\left(\begin{smallmatrix} \text{粤} & \text{粤} \\ \text{越} & \text{越} \end{smallmatrix} \right)$ 火 火 摇苗 摇苗 \neq 园
 摇越 摇越 火 火 摇苗 摇苗 园

则

(員 粵 越) 原 粵 垣 粵 園 粵 韻 原 粵 孕 原
原 孕 粵 粵 孕

(圓) 齒讓越齒讓_云 · 齒讓_云

其中,孕越^粵原^粵粵^原粵^粵

摇摇证明摇因为

$$\begin{aligned} & \text{粤越} \left(\begin{smallmatrix} \text{粤} & \text{粤} \\ \text{方} & \text{方} \\ \text{原} & \text{原} \end{smallmatrix} \right) \text{越} \left(\begin{smallmatrix} \text{限} & \text{园} \\ \text{方} & \text{方} \\ \text{原} & \text{原} \end{smallmatrix} \right) \left(\begin{smallmatrix} \text{粤} & \text{粤} \\ \text{方} & \text{方} \\ \text{原} & \text{原} \end{smallmatrix} \right) \\ & \text{越} \left(\begin{smallmatrix} \text{限} & \text{园} \\ \text{方} & \text{方} \\ \text{原} & \text{原} \end{smallmatrix} \right) \left(\begin{smallmatrix} \text{园} & \text{园} \\ \text{方} & \text{方} \\ \text{原} & \text{原} \end{smallmatrix} \right) \left(\begin{smallmatrix} \text{限} & \text{粤} & \text{粤} \\ \text{方} & \text{方} & \text{方} \\ \text{原} & \text{原} & \text{原} \end{smallmatrix} \right) \triangleq \text{粤} \cdot \text{粤} \cdot \text{粤} \end{aligned}$$

又

粵越(隕園) 粵越(隕園) 粵越(隕原粵粵)

所以,利用 $\frac{a}{b} > \frac{c}{d}$ 直接可得结论(员);利用 $\frac{a}{b} > \frac{c}{d}$ 可得结论(圆)。

附录 用矩阵微分的几个结果

本附录约定：

(员) $\frac{\partial \phi}{\partial \theta}$ 表示一个列向量, 其中 ϕ 是标量函数, θ 是向量, $\frac{\partial \phi}{\partial \theta_{\text{蚤}}}$ 是 $\frac{\partial \phi}{\partial \theta}$ 的第 蚤 行分量。

(圆) θ, β 都是向量, $\frac{\partial \phi}{\partial \theta}$ 表示以 $\frac{\partial \phi_{\text{蚤}}}{\partial \theta_{\text{躁}}}$ 为第 蚤 行第 躁 列的矩阵, $\beta_{\text{蚤}}, \theta_{\text{躁}}$ 分别为 β 的第 蚤 个分量和 θ 的第 躁 个分量。

(猿) ϕ 是标量, Φ 是矩阵, $\frac{\partial \phi}{\partial \Phi}$ 表示以 $\frac{\partial \phi}{\partial \Phi_{\text{蚤躁}}}$ 为第 蚤 行第 躁 列元素的矩阵, 其中 $\Phi_{\text{蚤躁}}$ 表示矩阵 Φ 的第 蚤 行第 躁 列元素。

有以下结果：

(员) 若 $\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}^m, \theta \in \mathbb{R}^k$, 则

$$\frac{\partial \beta}{\partial \theta} = \frac{\partial \beta}{\partial \theta} \alpha$$

证明因为

$$\frac{\partial \beta}{\partial \theta} = \frac{\partial \beta}{\partial \theta} \alpha = \frac{\partial \beta}{\partial \theta} \alpha$$

所以

$$\frac{\partial \beta}{\partial \theta} = \begin{pmatrix} \frac{\partial \beta}{\partial \theta_1} \\ \vdots \\ \frac{\partial \beta}{\partial \theta_k} \end{pmatrix} = \frac{\partial \beta}{\partial \theta} \alpha$$

(圆) 若 Φ 为 $n \times m$ 维常数矩阵, $\theta \in \mathbb{R}^k$, 则有

$$\frac{\partial (\Phi \theta)}{\partial \theta} = \Phi$$

这是因为

$$\frac{\partial (\Phi \theta)}{\partial \theta} = \frac{\partial \Phi \theta}{\partial \theta} = \Phi$$

(猿) 若 Φ 为 $n \times m$ 维矩阵, 利用 (员) 和 (圆) 的结果, 则有

$$\frac{\partial \Phi}{\partial \theta} = \frac{\partial \Phi}{\partial \theta} \Phi$$

(源) 设 Φ 是 $n \times n$ 阶可逆方阵, 它的元素是实数 θ 的函数, 则

$$\frac{\partial \Phi^{-1}}{\partial \theta} = -\Phi^{-1} \frac{\partial \Phi}{\partial \theta} \Phi^{-1}$$

证明因为

$$\text{园越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{垣} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}$$

故有

$$\frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越原} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}$$

摇摇(缘) 对任意非奇异矩阵 越,有

$$\frac{\partial}{\partial \text{越}} \text{越早} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}$$

摇摇证明摇摇因为

$$\text{越}^{\text{原}} \text{越葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}$$

所以

$$\text{越}(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}) \text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \cdot \text{陨}$$

摇摇将 越(葬)的第 蚤行第 蚤元素取对数得

$$\text{越早} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越} \text{越早} \sum_{\text{噪}} \text{越}(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})$$

于是

$$\frac{\partial \text{越早} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}}{\partial \text{越}} \text{越} \frac{\partial \text{越早} \sum_{\text{噪}} \text{越}(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})}{\partial \text{越}} \text{越} \frac{\text{员}}{\sum_{\text{噪}} \text{越}(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})} \frac{\partial \sum_{\text{噪}} \text{越}(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})}{\partial \text{越}} \text{越} \frac{\text{员}}{\text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}}(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})$$

因为(葬)不依赖于 越

摇摇故有

$$\frac{\partial \text{越早} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}}{\partial \text{越}} \text{越} \frac{(\text{葬} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})^{\text{栽}}}{\text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}} \text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}$$

摇摇(远) 如果 宰为常数矩阵 则有

$$\frac{\partial}{\partial \text{越}} (\text{越宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}) \text{越原} (\frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})$$

摇摇证明摇摇由于 越越^{\text{原}} 越员所以

$$\frac{\partial \text{越}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{垣} \frac{\partial \text{越}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越园}$$

从而有

$$\frac{\partial \text{越}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越原} \frac{\partial \text{越}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}$$

式中, 藻为单位矩阵 陨的第 蚤个列向量。

于是

$$\frac{\partial}{\partial \text{越}} (\text{越宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}) \text{越} \text{越宰} \frac{\partial \text{越}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越原} \text{越宰} \frac{\partial \text{越}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{越原} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{藻}$$

$$\text{越原} (\frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})$$

故有

$$\frac{\partial}{\partial \text{越}} (\text{越宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}}) \text{越原} (\frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}} \text{宰} \frac{\partial \text{越云}^{\text{原}}}{\partial \text{越}})^{\text{栽}}$$

摇摇注意 ,上面的证明中 ,用到了如下众所周知的事实 对任意矩阵 粤,月,都有

$$\text{越} \text{越}^{\text{原}} \text{越} \text{越}^{\text{原}}$$

· 圆原

只要矩阵 \mathbf{A} 与 \mathbf{B} 是可交换的。

附录 阅读正态伪随机数

产生 $N(0,1)$ 独立正态分布的伪随机序列 $\{\omega_{灶}\}$,分两步进行。

第一步 :用同余法产生 $(0,1)$ 区间上的均匀分布伪随机数 $\{\xi_{灶}\}$ 。

考虑同余方程

$$\xi_{灶+1} \equiv \alpha \xi_{灶} \pmod{m} \quad (阅读)$$

$$\xi_{灶} \in [0, 1) \quad (阅读)$$

其中 m 为正整数 , α 为大于零的常数 ,式(阅读)的意思是 $\xi_{灶+1}$ 取为 $\alpha \xi_{灶}$ 除以 m 所得的余数部分。因此 ,取正整数 m 后 ,按式(阅读)式可递推得到 ξ_1, ξ_2, \dots 。显然 $0 \leq \xi_{灶} \leq 1$,令

$$\xi_{灶} \equiv \xi_{灶} \pmod{m}, \dots$$

则 $\{\xi_{灶}\}$ 就是 $(0,1)$ 区间上均匀分布的伪随机序列。

第二步 :根据 $\{\xi_{灶}\}$ 用对数法产生 $N(0,1)$ 独立正态伪随机序列。

其公式为

$$\omega_{灶+1} \equiv (\xi_{灶+1} - \xi_{灶})^{1/2} \pmod{m} \quad (阅读)$$

$$\omega_{灶+2} \equiv (\xi_{灶+2} - \xi_{灶+1})^{1/2} \pmod{m} \quad (阅读)$$

摇摇根据式(阅读)和式(阅读)式由一对 $\xi_{灶}$ 产生一对 $\omega_{灶}$ 。这样得到 $\{\omega_{灶}\}$ 即是相互独立的 $N(0,1)$ 正态伪随机序列。

注意 ,为了不使 $\xi_{灶}$ 退化 , α , m 可按如下公式选取 : $\alpha \equiv 2^{k-1} \pmod{m}$,其中 $[k]$ 表示不大于 k 的最大整数 , $k \geq 1$,其中 k 为计算机的字长 , m 为进位制。例如 ,计算机为二进制 ,字长为 k ,则取 $\alpha \equiv 2^{k-1} \pmod{m}$, m 可取为 2^k 。

当欲产生分布为 $N(\mu, \sigma^2)$ 的正态分布伪随机序列时 ,式(阅读)和式(阅读)改为

$$\omega_{灶+1} \equiv \sigma(\xi_{灶+1} - \xi_{灶})^{1/2} \pmod{m} \quad (阅读)$$

$$\omega_{灶+2} \equiv \sigma(\xi_{灶+2} - \xi_{灶+1})^{1/2} \pmod{m} \quad (阅读)$$

参考文献

[illegible][illegible]

猿月藻藻藻藻分分別別體體系系菜菜苗苗精精製製乳乳藻藻蝦蝦燥燥霜霜藻藻各各種種完完援援孕孕登登藥藥社社孕孕登登藥藥社社故故登登藥藥贈
孕孕釋釋員員總總

源月籍東韵匀葵苗粤允葵苗菓庠藩樑杜撰貴賊親親趙趙華解棟迨華刺信國國粵建寧粵
 曠宅鄧妹志劫匪洋粵曠曠樑曠粵兇曠院曠獠

缘月保里戴悦蔡苗砸月器巢里另博毛剿稽登某通翻鼎杜零蛋剿良忘器典建豪籍巢宅受巢
剥晒巢零曹悦染巢泽 巢弱

[illegible][illegible]

愿瑶月增土砾云禁苗运砾郎渠烟赠名外赠美圣渠茎藻堆圣群赠圣叶渠青则藻助自渠藻圣曾藻葵京
精藻叶渠藻圣精圣叶名藻藻西之藻葵月渠藻叶藻(源)渠藻-渠原渠藻

[illegible]

元遥月燿吐砾禁崔孕孕剿操吐播策埔建建金增寒再霁霁电霁霁策埔宅堡霁，圆霁援员霄并疑圆
许陀之赠诗援虞猿园尿园

同音字

𧈧 𧈨 𧈩 𧈪 𧈫 𧈬 𧈭 𧈮 𧈯 𧈰 𧈱 𧈲 𧈳 𧈴 𧈵 𧈶 𧈷 𧈸 𧈹 𧈺 𧈻 𧈼 𧈽 𧈾 𧈿 𧉀 𧉁 𧉂 𧉃 𧉄 𧉅 𧉆 𧉇 𧉈 𧉉 𧉊 𧉋 𧉌 𧉍 𧉎 𧉏 𧉐 𧉑 𧉒 𧉓 𧉔 𧉕 𧉖 𧉗 𧉘 𧉙 𧉚 𧉛 𧉜 𧉝 𧉞 𧉟 𧉠 𧉡 𧉢 𧉣 𧉤 𧉥 𧉦 𧉧 𧉨 𧉩 𧉪 𧉫 𧉬 𧉭 𧉮 𧉯 𧉰 𧉱 𧉲 𧉳 𧉴 𧉵 𧉶 𧉷 𧉸 𧉹 𧉺 𧉻 𧉼 𧉽 𧉾 𧉿 𧊀 𧊁 𧊂 𧊃 𧊄 𧊅 𧊆 𧊇 𧊈 𧊉 𧊊 𧊋 𧊌 𧊍 𧊎 𧊏 𧊐 𧊑 𧊒 𧊓 𧊔 𧊕 𧊖 𧊗 𧊘 𧊙 𧊚 𧊛 𧊜 𧊝 𧊞 𧊟 𧊠 𧊡 𧊢 𧊣 𧊤 𧊥 𧊦 𧊧 𧊨 𧊩 𧊪 𧊫 𧊬 𧊭 𧊮 𧊯 𧊰 𧊱 𧊲 𧊳 𧊴 𧊵 𧊶 𧊷 𧊸 𧊹 𧊺 𧊻 𧊼 𧊽 𧊾 𧊿 𧋀 𧋁 𧋂 𧋃 𧋄 𧋅 𧋆 𧋇 𧋈 𧋉 𧋊 𧋋 𧋌 𧋍 𧋎 𧋏 𧋐 𧋑 𧋒 𧋓 𧋔 𧋕 𧋖 𧋗 𧋘 𧋙 𧋚 𧋛 𧋜 𧋝 𧋞 𧋟 𧋠 𧋡 𧋢 𧋣 𧋤 𧋥 𧋦 𧋧 𧋨 𧋩 𧋪 𧋫 𧋬 𧋭 𧋮 𧋯 𧋰 𧋱 𧋲 𧋳 𧋴 𧋵 𧋶 𧋷 𧋸 𧋹 𧋺 𧋻 𧋼 𧋽 𧋾 𧋿 𧌀 𧌁 𧌂 𧌃 𧌄 𧌅 𧌆 𧌇 𧌈 𧌉 𧌊 𧌋 𧌌 𧌍 𧌎 𧌏 𧌐 𧌑 𧌒 𧌓 𧌔 𧌕 𧌖 𧌗 𧌘 𧌙 𧌚 𧌛 𧌜 𧌝 𧌞 𧌟 𧌠 𧌡 𧌢 𧌣 𧌤 𧌥 𧌦 𧌧 𧌨 𧌩 𧌪 𧌫 𧌬 𧌭 𧌮 𧌯 𧌰 𧌱 𧌲 𧌳 𧌴 𧌵 𧌶 𧌷 𧌸 𧌹 𧌺 𧌻 𧌼 𧌽 𧌾 𧌿 𧍀 𧍁 𧍂 𧍃 𧍄 𧍅 𧍆 𧍇 𧍈 𧍉 𧍊 𧍋 𧍌 𧍍 𧍎 𧍏 𧍐 𧍑 𧍒 𧍓 𧍔 𧍕 𧍖 𧍗 𧍘 𧍙 𧍚 𧍛 𧍜 𧍝 𧍞 𧍟 𧍠 𧍡 𧍢 𧍣 𧍤 𧍥 𧍦 𧍧 𧍨 𧍩 𧍪 𧍫 𧍬 𧍭 𧍮 𧍯 𧍰 𧍱 𧍲 𧍳 𧍴 𧍵 𧍶 𧍷 𧍸 𧍹 𧍺 𧍻 𧍼 𧍽 𧍾 𧍿 𧎀 𧎁 𧎂 𧎃 𧎄 𧎅 𧎆 𧎇 𧎈 𧎉 𧎊 𧎋 𧎌 𧎍 𧎎 𧎏 𧎐 𧎑 𧎒 𧎓 𧎔 𧎕 𧎖 𧎗 𧎘 𧎙 𧎚 𧎛 𧎜 𧎝 𧎞 𧎟 𧎠 𧎡 𧎢 𧎣 𧎤 𧎥 𧎦 𧎧 𧎨 𧎩 𧎪 𧎫 𧎬 𧎭 𧎮 𧎯 𧎰 𧎱 𧎲 𧎳 𧎴 𧎵 𧎶 𧎷 𧎸 𧎹 𧎺 𧎻 𧎼 𧎽 𧎾 𧎿 𧏀 𧏁 𧏂 𧏃 𧏄 𧏅 𧏆 𧏇 𧏈 𧏉 𧏊 𧏋 𧏌 𧏍 𧏎 𧏏 𧏐 𧏑 𧏒 𧏓 𧏔 𧏕 𧏖 𧏗 𧏘 𧏙 𧏚 𧏛 𧏜 𧏝 𧏞 𧏟 𧏠 𧏡 𧏢 𧏣 𧏤 𧏥 𧏦 𧏧 𧏨 𧏩 𧏪 𧏫 𧏬 𧏭 𧏮 𧏯 𧏰 𧏱 𧏲 𧏳 𧏴 𧏵 𧏶 𧏷 𧏸 𧏹 𧏺 𧏻 𧏼 𧏽 𧏾 𧏿 𧐀 𧐁 𧐂 𧐃 𧐄 𧐅 𧐆 𧐇 𧐈 𧐉 𧐊 𧐋 𧐌 𧐍 𧐎 𧐏 𧐐 𧐑 𧐒 𧐓 𧐔 𧐕 𧐖 𧐗 𧐘 𧐙 𧐚 𧐛 𧐜 𧐝 𧐞 𧐟 𧐠 𧐡 𧐢 𧐣 𧐤 𧐥 𧐦 𧐧 𧐨 𧐩 𧐪 𧐫 𧐬 𧐭 𧐮 𧐯 𧐰 𧐱 𧐲 𧐳 𧐴 𧐵 𧐶 𧐷 𧐸 𧐹 𧐺 𧐻 𧐼 𧐽 𧐾 𧐿 𧑀 𧑁 𧑂 𧑃 𧑄 𧑅 𧑆 𧑇 𧑈 𧑉 𧑊 𧑋 𧑌 𧑍 𧑎 𧑏 𧑐 𧑑 𧑒 𧑓 𧑔 𧑕 𧑖 𧑗 𧑘 𧑙 𧑚 𧑛 𧑜 𧑝 𧑞 𧑟 𧑠 𧑡 𧑢 𧑣 𧑤 𧑥 𧑦 𧑧 𧑨 𧑩 𧑪 𧑫 𧑬 𧑭 𧑮 𧑯 𧑰 𧑱 𧑲 𧑳 𧑴 𧑵 𧑶 𧑷 𧑸 𧑹 𧑺 𧑻 𧑼 𧑽 𧑾 𧑿 𧒀 𧒁 𧒂 𧒃 𧒄 𧒅 𧒆 𧒇 𧒈 𧒉 𧒊 𧒋 𧒌 𧒍 𧒎 𧒏 𧒐 𧒑 𧒒 𧒓 𧒔 𧒕 𧒖 𧒗 𧒘 𧒙 𧒚 𧒛 𧒜 𧒝 𧒞 𧒟 𧒠 𧒡 𧒢 𧒣 𧒤 𧒥 𧒦 𧒧 𧒨 𧒩 𧒪 𧒫 𧒬 𧒭 𧒮 𧒯 𧒰 𧒱 𧒲 𧒳 𧒴 𧒵 𧒶 𧒷 𧒸 𧒹 𧒺 𧒻 𧒼 𧒽 𧒾 𧒿 𧓀 𧓁 𧓂 𧓃 𧓄 𧓅 𧓆 𧓇 𧓈 𧓉 𧓊 𧓋 𧓌 𧓍 𧓎 𧓏 𧓐 𧓑 𧓒 𧓓 𧓔 𧓕 𧓖 𧓗 𧓘 𧓙 𧓚 𧓛 𧓜 𧓝 𧓞 𧓟 𧓠 𧓡 𧓢 𧓣 𧓤 𧓥 𧓦 𧓧 𧓨 𧓩 𧓪 𧓫 𧓬 𧓭 𧓮 𧓯 𧓰 𧓱 𧓲 𧓳 𧓴 𧓵 𧓶 𧓷 𧓸 𧓹 𧓺 𧓻 𧓼 𧓽 𧓾 𧓿 𧔀 𧔁 𧔂 𧔃 𧔄 𧔅 𧔆 𧔇 𧔈 𧔉 𧔊 𧔋 𧔌 𧔍 𧔎 𧔏 𧔐 𧔑 𧔒 𧔓 𧔔 𧔕 𧔖 𧔗 𧔘 𧔙 𧔚 𧔛 𧔜 𧔝 𧔞 𧔟 𧔠 𧔡 𧔢 𧔣 𧔤 𧔥 𧔦 𧔧 𧔨 𧔩 𧔪 𧔫 𧔬 𧔭 𧔮 𧔯 𧔰 𧔱 𧔲 𧔳 𧔴 𧔵 𧔶 𧔷 𧔸 𧔹 𧔺 𧔻 𧔼 𧔽 𧔾 𧔿 𧕀 𧕁 𧕂 𧕃 𧕄 𧕅 𧕆 𧕇 𧕈 𧕉 𧕊 𧕋 𧕌 𧕍 𧕎 𧕏 𧕐 𧕑 𧕒 𧕓 𧕔 𧕕 𧕖 𧕗 𧕘

[illegible][illegible][illegible]

员 瑶悦 袁 霖 阅 宰 掇 陈 藻 滔 肇 董 争 原 泽 李 麟 藻 走 蔡 聚 出 恧 爆 寿 博 老 藻 剿 翠 孝 曲 挂 邑 翟 皂 藻 援
召 肆 舞 晃 警 兑 乞 冉 冉 书 援 昆 恆 茄

[illegible]

孙广振,王劲松援深圳股市混沌现象和辨识及其讨论援数量经济技术经济研究,灾
(员) 远源~ 远苑,灾源

谭永红援多层前向神经网络的 砸杂训练算法及其在辨识中的应用援控制理论及应用,
灾源猿缘~ 缘源,灾源

王士同援神经模糊系统与应用援北京: 航空航天大学出版社, 灾苑

王秀峰,卢桂章援多变量线性系统的递推算法援自动化学报,灾源苑源~ 源源,灾源

王秀峰,刘丹援非线性系统辨识—— 砸杂学习的一种新算法及其应用援自动化学报,
灾源远源~ 猿园,灾源

王秀峰,劳育红援非线性动态系统模型结构确定和参数估计新算法援自动化学报,
灾源愿源~ 猿缘,灾源

王秀峰,李波援随机非线性系统辨识的正交优选算法援自动化学报,灾猿猿~ 猿缘,
灾源

徐耀玲,戴汝为援人工神经网络在系统辨识中的应用援自动化学报,灾源猿猿~ 猿
~ 怨源,灾源

叶庆凯,王肇明援优化与最优控制中的计算方法援北京: 科学出版社, 灾源

张文修,梁怡援遗传算法的数学基础援西安: 西安交通大学出版社, 灾源

赵似兰等援生物系统的房室建模与辨识援北京: 科学出版社, 灾苑

周概容援概率论与数理统计援北京: 高等教育出版社, 灾源