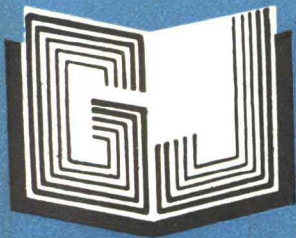


961789

TK730.2
7443



高等学校教材

水轮机调节系统计算机仿真

陕西机械学院 陈嘉谋 编

30.2
3



高等學校教材

水轮机调节系统计算机仿真

陕西机械学院 陈嘉谋 编

水利电力出版社

(京)新登字115号

内 容 提 要

全书共分六章, 内容包括: 连续控制系统计算机仿真基础; 水轮机调节系统各组成部分的数学模型; 水轮机调节系统线性和非线性仿真、大波动过程仿真, 以及与远距离输电系统连接的水轮机调节系统线性仿真等。本书反映了我国在水轮机调节系统仿真领域的研究成果。

本书可作为水利水电工程专业选修课教材、研究生教材, 也可供从事水轮机调节工作的工程技术人员参考。

高等学校教材

水轮机调节系统计算机仿真

陕西机械学院 陈嘉谋 编

水利电力出版社出版

(北京三里河路6号)

新华书店北京发行所发行·各地新华书店经售

北京四季青印刷厂印刷

787×1092毫米 16开本 8.75印张 195千字

1993年6月第一版 1993年6月北京第一次印刷

印数 0001—1536册

ISBN 7-126-01673-3/T·V·819

定价 2.35元

前 言

自70年代以来,我国许多学者致力于水轮机调节系统计算机仿真的研究,近20年来这项技术已得到很大发展,并广泛应用于水电站及水力机组的设计、建造和运行中。有些高等学校已开设了《水轮机调节系统计算机仿真》课程。本教材就是根据水利水电类专业教材编审出版规划,以作者在陕西机械学院讲授该课程时所编写的同名讲义为基础修改而成。

本书力图集中反映我国在水轮机调节系统仿真领域的研究成果,其中也包括作者的研究成果。全书主要论述水轮机调节系统大小波动过程仿真的理论和方法。作为阅读本书的引导,在第一、二章介绍了计算机仿真的理论基础和水轮机调节系统的基本数学模型,本书最后讨论了与远距离输电系统连接时的水轮机调节系统仿真方法。

本书稿承蒙华中理工大学沈宗树教授认真审阅,提出了许多宝贵意见,谨致谢意。

由于本人水平所限,且匆促成书,诚恳欢迎对本书提出意见和批评。

作 者

1991年12月

EA09/10

目 录

前言	
绪论	1
第一章 连续控制系统计算机仿真基础	3
第一节 连续控制系统计算机仿真模型的构造方法	3
第二节 控制系统计算机仿真的数学方法	10
第二章 水轮机调节系统各组成部分数学模型	25
第一节 压力引水系统数学模型	25
第二节 水轮机数学模型	34
第三节 发电机及负载数学模型	38
第四节 水轮机调速器数学模型	39
第三章 水轮机调节系统线性仿真	46
第一节 水轮机调节系统线性仿真数学模型	46
第二节 水轮机调节系统线性仿真程序框图	58
第三节 水轮机调节系统线性仿真的数据准备	62
第四章 水轮机调节系统非线性仿真	67
第一节 水轮机调节系统非线性仿真数学模型	67
第二节 水轮机调节系统非线性仿真程序框图和数据准备	72
第三节 复杂水轮机调节系统非线性仿真	79
第五章 水轮机调节系统大波动过程仿真	92
第一节 水锤特征线解法原理	92
第二节 水轮机调节系统大波动过程仿真 I——利用水锤连锁方程	95
第三节 水轮机调节系统大波动过程仿真 II——利用水锤特征线法	108
第六章 与远距离输电系统连接的水轮机调节系统线性仿真	119
第一节 与无穷大系统并列运行的同步发电机基本方程组	119
第二节 与远距离输电系统连接的水轮机调节系统仿真基本方法	127
参考文献	132

绪 论

一、控制系统计算机仿真

设计或改进一个自动控制系统不但需要进行理论分析计算,通常还需要进行实验研究。然而在实际系统上进行实验研究有时是难以实现的。这首先是出于对系统安全的考虑,其次是实际系统上的实验往往经济代价较大,而且由于实际系统运行状态的限制,使实验条件的变化受到约束,因此在许多情况下就转向模型实验研究。

模型实验研究有两类,一种是用物理模型来模拟实际系统,通常称为“物理模拟”或“物理仿真”。由于建立一个物理模型往往需要耗费较多的财力和物力,周期较长,实验准备工作量可观,而且模型重复利用率低,随着实际系统越来越复杂,有的也难于用物理模型来精确地仿真。因此,现在越来越多地采用“数学仿真”。数学仿真就是将实际系统的运动规律用数学形式表达出来(表达实际系统动态特性的数学表达式便称为系统的数学模型,它们通常是一些微分方程和代数方程的组合),然后用计算机来求解这些方程,便可以分析研究实际系统的性能。由于数学仿真的主要工具是电子计算机,因此又称为“计算机仿真”。

计算机仿真可分为三种:用模拟计算机仿真的,称为“模拟仿真”;用数字计算机进行仿真的,称为“数字仿真”;用数字计算机和模拟计算机联合进行仿真的,称为“混合仿真”。模拟仿真的优点是快速、连续,可以满足实时仿真的需要,但缺点是精度低,一般为千分之几,逻辑功能较差,仿真工作自动程度较低。数字仿真正好相反,精度高,逻辑判断功能强,能适用于采样控制系统,参数修改方便,但速度慢。混合仿真则可以兼顾两者的优点,达到快速、精确的目的。

数字仿真是现在应用最广泛的仿真方法,其主要步骤如下:

(1) 建立实际物理系统的数学模型。它可以根据物理定律或定理,运用推理方法建立;也可以根据实验观测或统计,用归纳方法建立;或者是两者兼而有之。

(2) 将数学模型转换成在计算机上可以实现的仿真模型。

(3) 根据仿真模型编制计算机程序。

(4) 确定实际系统参数和原始数据。

(5) 将计算机程序及数据输入和贮存到计算机中,并通过计算机运行来实现仿真。

现在计算机仿真技术已经广泛应用到各种领域,对于各种新型控制系统的设计、参数匹配、最优控制等,计算机仿真都是一种很有效的手段,并且取得了可信赖的、令人满意的结果。

二、水轮机调节系统计算机仿真

水轮机调节系统是一个复杂的自动控制系统,其动态过程可分为小波动和大波动两类。小波动是指水轮机调节系统受到微小的干扰(负荷或指令信号扰动),系统中各参数的变化都较小,可以认为是在所讨论工况点附近作微小变化,这就可以将调节系统各环节加以

线性化,即可用线性微分方程式来描述各环节及整个系统的动态特征;而大波动是指调节系统受到幅度较大的干扰(负荷变化),系统参数变化剧烈,整个系统已超出了线性范围,因此不能作线性处理,即系统不能按线性系统对待。对小波动,人们关心的是水轮机调节系统工作的稳定性,即关心所产生电能的质量;而对大波动,人们所关心的主要是在负荷突然变化(特别是甩负荷)时水压、转速等各种参数的变化情况。因为水轮机调节系统工作性能的优劣直接关系到水电站和机组运行的安全和生产电能的质量,因此水轮机调节系统的动态过程一直是水电站和水电机组设计人员很关心的一个问题。随着生产技术和设计水平的提高,对水电站工程设计要求也越来越高。有些用人工计算方法已不能满足,用物理模拟方法来分析又费用昂贵,耗时较多,因此采用计算机仿真技术(数字仿真)分析水轮机调节系统各种变化过程便应运而生了。

自70年代以来,我国有许多学者致力于水轮机调节系统计算机仿真的研究,其成果应用于工程上最早的是1974年湖南龙源水电站以调压阀代替调压井的研究中(由华中工学院和湖南省水电勘测设计院完成)。10多年来这项技术已在我国得到很大的发展,在各种型式水电站的甩负荷过程计算、小波动稳定分析、以及抽水蓄能电站的过渡过程计算中得到广泛地采用。特别是水电站调节保证电算,对于大型水电站设计来讲已是不可缺少的了。全国不少研究所、高等学校、设计院都先后开发了许多应用软件。有些高等学校也开设了《水轮机调节系统计算机仿真》这一选修课。本教材就是搜集我国在这方面的研究成果,结合本人的教学科研工作,并参考了国外的发展动向整理而成的。

本教材主要讲述水轮机调节系统小波动过渡过程和大波动过程的计算机数字仿真。小波动过程仿真限于研究系统参数在小范围内变化,将系统当作线性系统看待,其仿真方法对大波动是不适用的。而大波动过程考虑了参数大幅度地、剧烈地变化,考虑了各种非线性因素,因此其方法从本质上讲对小波动仿真也是可以适用的。所以大、小波动过程的仿真是可以统一的,实际工程设计中也已经有了这样做的先例。在水轮机调节系统仿真的具体方法上,可以采用状态方程解法和差分方程解法两种。本书第三章介绍状态方程解法用于线性系统仿真,它可以用来分析调节系统小波动过程。第四章是将状态方程解法应用于非线性系统仿真。它是在线性系统状态方程的基础上,考虑系统中存在的非线性因素而建立起来的非线性仿真模型,也就是在线性模型的基础上加以修正的。它一般适用于偏离线性范围不大的非线性系统仿真,譬如调节系统空载扰动或负载扰动过程仿真。第五章介绍差分方程解法用于调节系统仿真,它适用于非线性仿真,也可用于线性仿真。目前国内大波动过程电算多采用差分方程解法,因此将第五章取名为大波动过程仿真。最后一章介绍与远距离输电系统连接的水轮机调节系统仿真,在这一章中比较详细地分析了电机瞬态过程对水轮机调节系统的影响,可以作为研究水、机、电联合控制系统仿真的基础。

阅读本书之前,应已经掌握《水轮机调节》、《程序设计》的基本内容。为了使读者顺利阅读本书,本书第一、二两章分别介绍了控制系统计算机仿真的基本方法和水轮机调节系统的基本特性,作为讨论水轮机调节系统仿真的基础。但是限于篇幅,这两章介绍比较简单,如果读者需要,可翻阅有关《控制系统计算机仿真》和《水轮机调节》的专门书籍。

第一章 连续控制系统计算机仿真基础

连续控制系统是自动控制系统最基本最常见的型式。本章介绍连续控制系统仿真的基本方法,以后各章讨论水轮机调节系统计算机仿真时可以运用。

第一节 连续控制系统计算机仿真模型的构造方法

一个自动控制系统,其动态特性往往用传递函数和微分方程来表示,较直观的方法也可以用结构框图来表示,针对不同的描述方式,可以用不同的方法构造仿真模型。

一、面向微分方程的仿真模型

设连续控制系统输入量为 $u(t)$, 输出量为 $y(t)$, 它们之间关系可以用下列 n 阶微分方程来描述:

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

式中, $a_1, a_2, \dots, a_{n-1}, a_n$ 及 $b_0, b_1, \dots, b_{m-1}, b_m$ 为常系数, 且 $n \geq m$; $y^{(n)}, y^{(n-1)}, \dots, \dot{y}$ 及 $u^{(m)}, u^{(m-1)}, \dots, \dot{u}$ 为 $y(t)$ 及 $u(t)$ 的各阶导数。

如果控制系统是用传递函数表示, 可写成:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (1-2)$$

式中 $Y(s)$ ——输出量 $y(t)$ 的拉氏变换;

$U(s)$ ——输入量 $u(t)$ 的拉氏变换。

在初始状态为零的条件下(即 $t=0$ 时, y 和 u 及其各阶导数均为零), 式(1-2)和式(1-1)是等价的, 式(1-2)也可以转换成式(1-1)的形式。

当连续控制系统用式(1-1)来描述时, 由于这是一个高阶微分方程, 如果要在数字计算机上实现仿真, 现在通用的办法是将它转化成一阶微分方程组, 即状态方程式。下面讨论两种类型的转换:

1. 不含输入量导数项的 n 阶微分方程

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

对于上述 n 阶微分方程, 可以定义 n 个状态变量 $x_1, x_2, \dots, x_n$, 令

$$x_1 = y$$

$$x_2 = \dot{x}_1 = \dot{y}$$

$$x_3 = \dot{x}_2 = \ddot{y}$$

$$\dots$$

$$x_n = \dot{x}_{n-1} = y^{(n-1)}$$

将上述 n 个定义式代入式 (1-3) 中, 可得:

$$\dot{x}_n = y^{(n)} = -a_1 x_n - a_2 x_{n-1} - \dots - a_{n-1} x_2 - a_n x_1 + u$$

将 $\dot{x}_1, \dots, \dot{x}_n$ 的 n 个一阶微分方程写成矩阵向量形式, 得:

$$\dot{X} = AX + Bu \quad (1-4)$$

$$Y = CX \quad (1-5)$$

式 (1-4)、式 (1-5) 中 A 、 B 、 C 为系数矩阵:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & & \dots & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = [1 \ 0 \ 0 \ \dots \ 0] \quad X = [x_1 \ x_2 \ \dots \ x_n]^T$$

式 (1-4) 为状态方程, 式 (1-5) 为输出方程, 合起来称为状态空间表达式。式 (1-4) 和式 (1-5) 可以在数字计算机上求解。

2. 含有输入量导数项的 n 阶微分方程

$$y^{(n)} + a_1 y^{(n-1)} + \dots + a_{n-1} \dot{y} + a_n y = b_0 u^{(n)} + b_1 u^{(n-1)} + \dots + b_{n-1} \dot{u} + b_n u \quad (1-6)$$

式 (1-6) 和式 (1-1) 的差别在于方程右边输入量 u 的最高阶数为 n 而不是 m , 由于实际系统中 $n > m$, 因此式 (1-6) 包含了式 (1-1)。对于式 (1-6) 可以取以下 n 个变量作为一组状态变量。

$$x_1 = y - \beta_0 u$$

$$x_2 = \dot{x}_1 - \beta_1 u = \dot{y} - \beta_0 \dot{u} - \beta_1 u$$

$$x_3 = \dot{x}_2 - \beta_2 u = \ddot{y} - \beta_0 \ddot{u} - \beta_1 \dot{u} - \beta_2 u$$

$$\vdots$$

$$x_n = \dot{x}_{n-1} - \beta_{n-1} u = y^{(n-1)} - \beta_0 u^{(n-1)} - \beta_1 u^{(n-2)} - \dots - \beta_{n-2} \dot{u} - \beta_{n-1} u$$

上列各式中 $\beta_0, \beta_1, \dots, \beta_n$ 为:

$$\beta_0 = b_0$$

$$\beta_1 = b_1 - a_1 \beta_0$$

$$\beta_2 = b_2 - a_1 \beta_1 - a_2 \beta_0$$

$$\beta_3 = b_3 - a_1 \beta_2 - a_2 \beta_1 - a_3 \beta_0$$

$$\vdots$$

$$\beta_n = b_n - a_1 \beta_{n-1} - \dots - a_{n-1} \beta_1 - a_n \beta_0$$

由此可以得出对应的状态方程和输出方程:

$$\dot{X} = AX + Bu \quad (1-7)$$

$$Y = CX + Du \quad (1-8)$$

式 (1-7)、式 (1-8) 中

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

$$C = [1 \ 0 \ 0 \ \cdots \ 0] \quad D = \beta_0 = b_0$$

$$X = [x_1 \ x_2 \ \cdots \ x_n]^T$$

初始条件 $X(0)$ 可由各状态变量的定义式来求。

对于一个控制系统, 状态变量的选取不是唯一的, 因此上述状态空间表达式也不是唯一的。当采取不同的状态变量时, 状态方程式是不相同的。

二、面向结构框图的仿真模型

实际控制系统分析时, 往往先给出的是系统结构框图。为采用上述方法, 则要先由系统框图求出系统传递函数, 再转换成状态方程, 而且某一环节参数变化时状态方程的系数矩阵 A 、 B 都可能变化, 需重新计算, 这将给仿真工作带来不便。此外, 如果系统中带有非线性环节, 面向微分方程的仿真方法就无法处理。而面向结构框图的仿真方法可以克服上述缺陷。

1. 典型环节的确定

一个控制系统可能由各种各样的环节所组成, 比较常见的环节有比例环节、积分环节、惯性环节、实际微分环节和二阶振荡环节。它们都可以用下列一阶基本环节和二阶基本环节来表示:

$$\text{一阶基本环节} \quad \frac{c + ds}{a + bs} \quad (1-9)$$

$$\text{二阶基本环节} \quad \frac{p_1 + p_2 s + p_3 s^2}{q_1 + q_2 s + q_3 s^2} \quad (1-10)$$

式 (1-9) 中, 当 $b = d = 0$, 为比例环节; 当 $a = d = 0$, 为积分环节; 当 $d = 0$, 为惯性环节; 当 $c = 0$, 为实际微分环节。二阶基本环节〔式 (1-10)〕又可以用两个一阶基本环节来构成, 如图1-1所示。

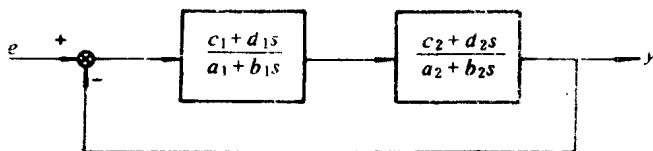


图 1-1 用两个一阶基本环节构成二阶基本环节

设图中 $\frac{c_1 + d_1 s}{a_1 + b_1 s} = G_1(s)$, $\frac{c_2 + d_2 s}{a_2 + b_2 s} = G_2(s)$

如图闭环传递函数:

$$G(s) = \frac{G_1(s)G_2(s)}{1 + G_1(s)G_2(s)} = \frac{c_1 c_2 + (c_1 d_2 + c_2 d_1)s + d_1 d_2 s^2}{(a_1 a_2 + c_1 c_2) + (a_1 b_2 + b_1 a_2 + c_1 d_2 + c_2 d_1)s + (b_1 b_2 + d_1 d_2)s^2}$$

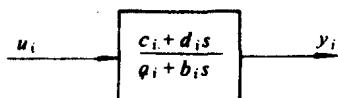
上式中令 $p_1 = c_1 c_2$, $p_2 = c_1 d_2 + c_2 d_1$, $p_3 = d_1 d_2$

$$q_1 = a_1 a_2 + c_1 c_2, \quad q_2 = a_1 b_2 + a_2 b_1 + c_1 d_2 + c_2 d_1, \quad q_3 = b_1 b_2 + d_1 d_2$$

则

$$G(s) = \frac{p_1 + p_2 s + p_3 s^2}{q_1 + q_2 s + q_3 s^2}$$

这就构成了二阶基本环节。因此可以选取图1-2所示一阶环节为典型环节。由这种典型环节组成控制系统, 每个环节的传递函数都可写成:



$$\frac{Y_i(s)}{U_i(s)} = \frac{c_i + d_i s}{a_i + b_i s} \quad (1-11)$$

图 1-2 典型环节传递函数

设控制系统可表示为 n 个典型环节组成, 写成矩阵方程式有:

$$(A + Bs)Y = (C + Ds)U \quad (1-12)$$

式中

$$A = \begin{bmatrix} a_1 & 0 & & \\ & a_2 & & \\ & & \ddots & \\ 0 & & & a_n \end{bmatrix} \quad B = \begin{bmatrix} b_1 & 0 & & \\ & b_2 & & \\ & & \ddots & \\ 0 & & & b_n \end{bmatrix}$$

$$C = \begin{bmatrix} c_1 & 0 & & \\ & c_2 & & \\ & & \ddots & \\ 0 & & & c_n \end{bmatrix} \quad D = \begin{bmatrix} d_1 & 0 & & \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

2. 连接矩阵

为了描述系统, 还要将环节与环节之间的连接关系表示出来, 现以图1-3所示系统为例来说明。

图1-3中环节的输入用符号 u_i 表示, 输出用符号 y_i 来表示, 系统的外加输入信号用 e_i 表示。此外, 比例环节不单独作为一个环节来考虑。

据图1-3可写出下列方程组:

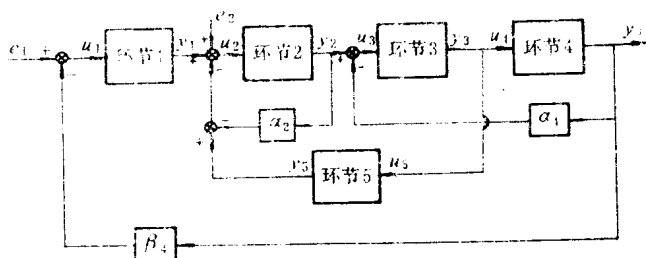


图 1-3 控制系统结构框图

$$\left. \begin{aligned} u_1 &= -\beta_4 y_1 + e_1 \\ u_2 &= y_1 - \alpha_2 y_2 - y_5 + e_2 \\ u_3 &= y_2 - \alpha_4 y_4 \\ u_4 &= y_3 \\ u_5 &= y_3 \end{aligned} \right\} \quad (1-13)$$

写成一般形式有:

$$u_i = \sum_{j=1}^n w_{ij} y_j + e_i \quad (i=1, 2, \dots, n, j=1, 2, \dots, n) \quad (1-14)$$

式中 e_i ——第 i 个环节的系统外加输入量;

u_i ——第 i 个环节的输入量;

y_j ——第 j 个环节的输出量;

n ——环节数;

w_{ij} ——由第 j 个环节连向第 i 个环节的连接关系, 即连接系数。

环节与环节之间的连接关系 w_{ij} 一般有三种:

(1) 环节 j 的输出量 y_j 直接与第 i 个环节相连接, 这时 $w_{ij} = 1$, 如图1-3中 $w_{43} = 1$, $w_{53} = 1$ 。

(2) 环节 j 的输出量与环节 i 没有连接关系, 这时 $w_{ij} = 0$, 如图1-3中 $w_{24} = 0$ 。

(3) 环节 j 的输出量 y_j 经过比例系数加到环节 i 作为输入, 如图1-3中 $w_{14} = -\beta_4$, $w_{34} = -\alpha_4$, $w_{22} = -\alpha_2$ 。

将式 (1-13) 写成矩阵方程形式:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -\beta_4 & 0 \\ 1 & -\alpha_2 & 0 & 0 & -1 \\ 0 & 1 & 0 & -\alpha_4 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \quad (1-15)$$

或写成:

$$U = WY + W_0 E \quad (1-16)$$

式中 U ——环节输入向量;

Y ——环节输出向量;

E ——系统输入向量;

W, W_0 ——连接矩阵。

3. 系统状态方程

根据由典型环节组成的矩阵方程式 (1-12) 和连接矩阵方程 (1-16) 可得:

$$(A + Bs)Y = (C + Ds)(WY + W_0 E) \quad (1-17)$$

经整理有

$$(B - DW)sY = (CW - A)Y + CW_0 E + DW_0 sE$$

令 $Q = B - DW$, $R = CW - A$, $V_1 = CW_0$ 和 $V_2 = DW_0$ 则上式可表示成:

$$QsY = RY + V_1 E + V_2 sE$$

如果 Q 的逆矩阵存在, 则上两式左边乘 Q^{-1} , 得:

$$sY = Q^{-1}RY + Q^{-1}V_1 E + Q^{-1}V_2 sE$$

对于控制系统, 零初始条件一般总是成立的, 上式转变成时域方程 (即求拉氏逆变换) 有:

$$\dot{Y} = Q^{-1}RY + Q^{-1}V_1 E + Q^{-1}V_2 \dot{E} \quad (1-18)$$

这是一个一阶微分方程组, 可以在计算机上求解, 式 (1-18) 就是面向结构框图的仿真模型。

但应看到, 式 (1-18) 右边含有 $\dot{E}$ 项, 即系统输入作用的导数项。如果输入作用信号中有某一项 e_i 是阶跃函数, 那么其导数 $\dot{e}_i \rightarrow \infty$, 则式 (1-18) 无法求解。为了使式 (1-18) 有解, 就应要求 $V_2 = 0$, 即 $DW_0 = 0$, 也就是对于 e_i 所直接作用的那个环节, 应满足 $d_i = 0$ 。所以采用面向结构框图的仿真模型时, 应给予下列限制: 如果对某一环节的外加输入信号是阶跃函数, 则该环节传递函数中 d_i 项应为零。

当满足 $V_2 = 0$ 时, 式 (1-18) 可写成:

$$\dot{Y} = Q^{-1}RY + Q^{-1}V_1 E$$

令 $Q^{-1}R = \tilde{A}$ 和 $Q^{-1}V_1 = \tilde{B}$ 则上式可表示成:

$$\dot{Y} = \tilde{A}Y + \tilde{B}E \quad (1-19)$$

式 (1-19) 就是常用的面向结构框图的仿真模型。

三、典型非线性环节仿真

在控制系统中经常包含有非线性环节, 这里介绍几种常见的非线性环节仿真方法。

1. 饱和非线性环节

饱和非线性环节输出 y_i 与输入 u_i 关系, 如图 1-4(a) 所示。用数学公式描述其关系为:

$$y_i = \begin{cases} Ka & u_i > a \\ Ku_i & -a < u_i < a \\ -Ka & u_i < -a \end{cases} \quad (1-20)$$

根据式 (1-20) 给出其程序框图, 如图 1-4(b) 所示。

2. 死区非线性环节

死区非线性环节输出 y_i 与输入 u_i 关系如图 1-5(a) 所示。用数学描述其关系为:

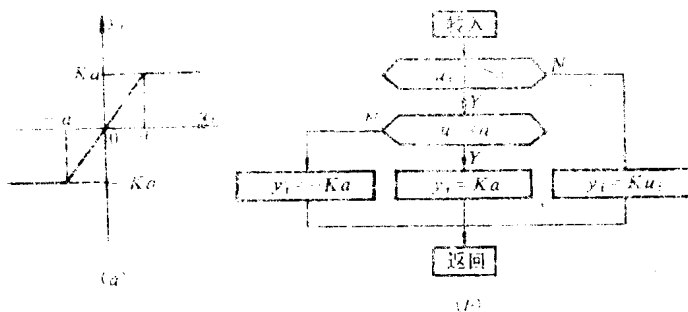


图 1-4 饱和非线性特性及子程序框图
(a) 饱和非线性特性; (b) 饱和非线性子程序框图

$$y_i = \begin{cases} K(u_i - a) & u_i \geq a \\ 0 & -a < u_i < a \\ K(u_i + a) & u_i \leq -a \end{cases} \quad (1-21)$$

式 (1-21) 中 $K = \tan \theta$, 根据上式写出程序框图, 如图 1-5(b) 所示。

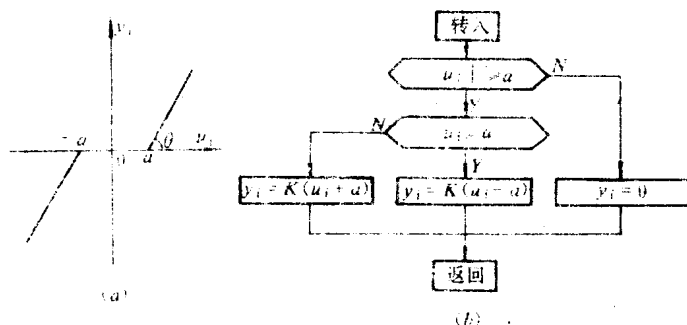


图 1-5 死区非线性特性及子程序框图
(a) 死区非线性特性; (b) 子程序框图

3. 齿轮间隙 (磁滞回环) 非线性环节

齿轮间隙 (磁滞回环) 非线性环节输出 y_i 与输入 u_i 间的关系如图 1-6(a) 所示, 用数学式描述其关系为:

$$y_i = \begin{cases} K(u_i - a) & u_i - u_i^0 > 0 \text{ 且 } y_i^0 \leq K(u_i - a) \\ K(u_i + a) & u_i - u_i^0 < 0 \text{ 且 } y_i^0 \geq K(u_i + a) \\ y_i^0 & \text{如果不满足上述条件} \end{cases} \quad (1-22)$$

式中 $K = \tan \theta$;

u_i, y_i^0 ——上一计算时刻的 u_i 和 y_i 值。

根据式 (1-22) 绘出程序框图如图 1-6(b) 所示。

4. 继电器及具有死区的继电器非线性环节

继电器及具有死区的继电器非线性环节特性, 如图 1-7 所示。图 1-7(a) 是图 1-7(b) 当死区

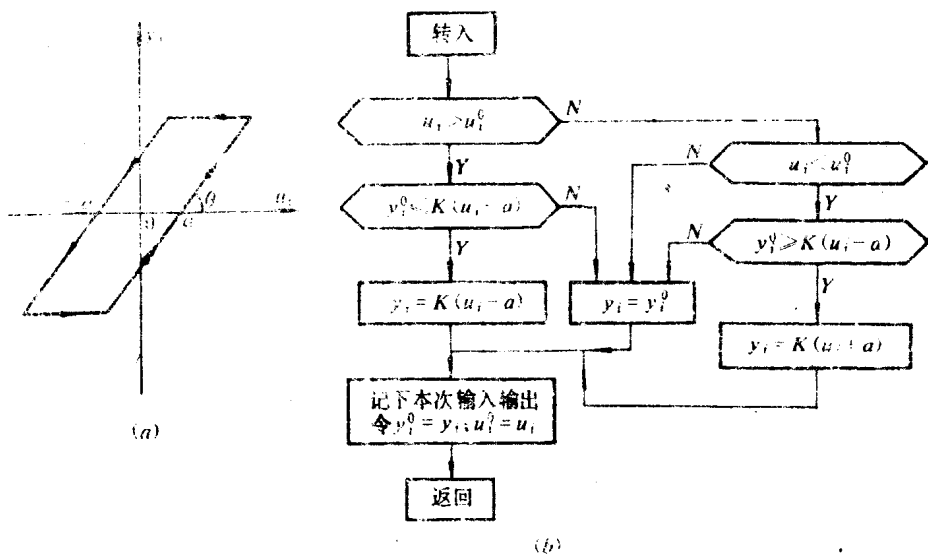


图 1-6 齿轮间隙非线性特性及子程序框图

(a) 非线性特性; (b) 子程序框图

$a = 0$ 时的特例。

用数学式描述其关系为:

$$y_i = \begin{cases} c & u_i \geq a \\ 0 & -a < u_i < a \\ -c & u_i \leq -a \end{cases} \quad (1-23)$$

依据上述关系, 可以绘出程序框图, 如图 1-8 所示。

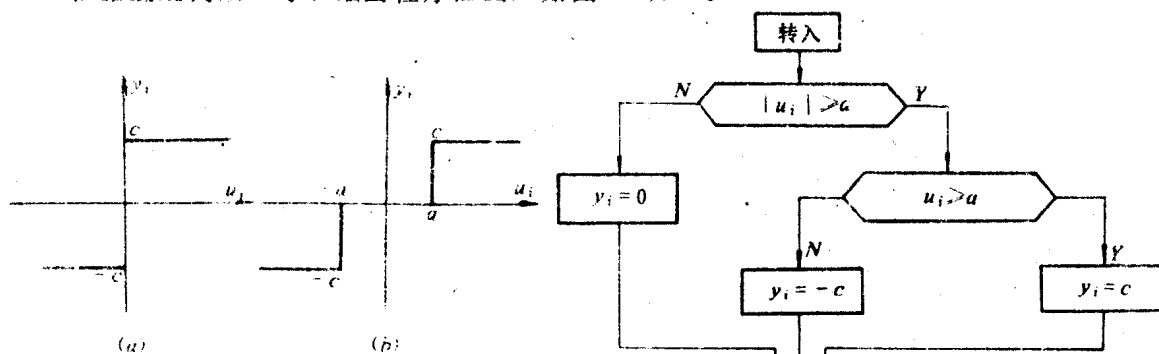


图 1-7 继电及具有死区的继电非线性特性

(a) 继电非线性特性; (b) 具有死区的继电非线性特性

图 1-8 具有死区的继电非线性特性

第二节 控制系统计算机仿真的数学方法

一、一阶微分方程组的数值解

第一节叙述了控制系统可用一阶微分方程组或状态方程式来描述, 状态方程式可认为

就是一阶微分方程组的一种简写形式。因此，控制系统计算机仿真往往就是要利用计算机来求解一阶微分方程组，通常的做法是采用数值积分来近似计算。这里介绍几种常用的数值计算方法。

1. 欧拉法

若微分方程为：
$$y' = f(t, y)$$

初始条件：
$$y(t_0) = y_0 \quad (1-24)$$

欧拉法的基本思路是将微商 df/dt 用差商来代替，即

$$\frac{y_{m+1} - y_m}{t_{m+1} - t_m} = y' = f(t_m, y_m) \quad (1-25)$$

或
$$y_{m+1} = y_m + f(t_m, y_m)h \quad (1-25')$$

式中 $h = t_{m+1} - t_m$ ——每次计算的自变量增值，即步长；

y_m ——第 m 次计算的因变量值；

t_m ——第 m 次计算的自变量值。

利用式 (1-25') 就可以从初始条件 t_0 、 y_0 出发逐步推算到任一步长时的 y 值。欧拉法是以第 m 点 (t_m, y_m) 处的斜率来代替第 m 点至第 $m+1$ 点处变化的斜率，因此其计算存在的误差是显而易见的。为了提高精度，可以考虑以第 m 点和第 $m+1$ 点斜率的平均值来取代欧拉法中所采用的第 m 点斜率，这就是梯形法。

2. 梯形法

同样针对微分方程式 (1-24)，梯形法计算公式为：

$$y_{m+1} = y_m + \frac{1}{2}h [f(t_m, y_m) + f(t_{m+1}, y_{m+1})] \quad (1-26)$$

式中 $f(t_m, y_m)$ 和 $f(t_{m+1}, y_{m+1})$ 为函数 y 在第 m 点和第 $m+1$ 点处的导数。

梯形法显然比欧拉法使用精度有所提高，但由于式 (1-26) 右边含有未知量 y_{m+1} ，不能直接使用该式由 y_m 值一步求出 y_{m+1} 值，因此要利用迭代的方法。但迭代次数增加将使计算工作量成比例地增长，一般采用一次迭代就够了，这时梯形法计算公式可以写成如下形式：

$$y_{m+1}^{(0)} = y_m + hf(t_m, y_m) \quad (1-27)$$

$$y_{m+1} = y_m + \frac{1}{2}h [f(t_m, y_m) + f(t_{m+1}, y_{m+1}^{(0)})] \quad (1-28)$$

式 (1-27)、式 (1-28) 通称为预报——校正公式，其中式 (1-27) 称预报公式，由它预报 y_{m+1} 的一个接近值，式 (1-28) 称为校正公式，由它得出 y_{m+1} 的校正值。

梯形法计算精度虽比欧拉法提高，但在计算精度要求较高时，必须将步长取得很小，这将使计算工作量大大增加，而且累计舍入误差也加大。因此为了提高计算速度和精度，通常采用四阶龙格—库塔法。

3. 四阶龙格—库塔(RG—KT)法

同样针对微分方程式 (1-24)，计算公式为：

$$y_{m+1} = y_m + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \quad (1-29)$$

式中 $k_1 = f(t_m, y_m)$;

$$k_2 = f\left(t_m + \frac{h}{2}, y_m + \frac{1}{2}hk_1\right);$$

$$k_3 = f\left(t_m + \frac{h}{2}, y_m + \frac{1}{2}hk_2\right);$$

$$k_4 = f(t_m + h, y_m + hk_3)。$$

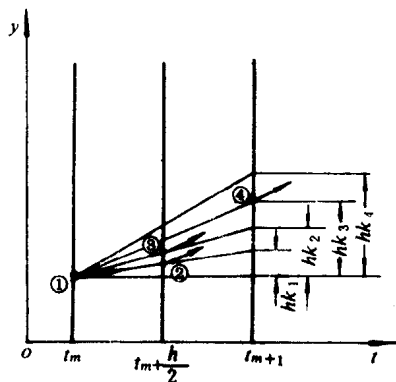


图 1-9 RG—KT法示意图

对照图1-9, 点①处的 y' 值为 $f(t_m, y_m)$, 即式 (1-29) 中的 k_1 , 而 k_2 、 k_3 、 k_4 分别为点②、③、④处的 y' 值, 图1-9中用各点上的箭头表示该点处的斜率 k_i 即 y'_i 值。从式 (1-29) 可看出, 龙格—库塔法是用4个点上斜率的加权平均值来代替梯形法中相邻两点间斜率的平均值, 因此在同样计算步长 h 下, 它可以获得比梯形法更高的计算精度。

上述讨论的是一个一阶微分方程的解法, 如果是由 n 个一阶微分方程组成的微分方程组, 其计算方法如下:

已知 n 个一阶微分方程组:

$$\left. \begin{aligned} y'_1 &= f_1(t, y_1, y_2, \dots, y_n) \\ y'_2 &= f_2(t, y_1, y_2, \dots, y_n) \\ y'_3 &= f_3(t, y_1, y_2, \dots, y_n) \\ &\vdots \\ y'_i &= f_i(t, y_1, y_2, \dots, y_n) \\ &\vdots \\ y'_n &= f_n(t, y_1, y_2, \dots, y_n) \end{aligned} \right\} \quad (1-30)$$

初始条件为:

$$\begin{aligned} y_1(t_0) &= y_{10} \\ y_2(t_0) &= y_{20} \\ &\vdots \\ y_i(t_0) &= y_{i0} \\ &\vdots \\ y_n(t_0) &= y_{n0} \end{aligned}$$

RG—KT法计算公式 (由第 m 步计算第 $m+1$ 步函数值) 为:

$$\begin{aligned} y_{i, m+1} &= y_{i, m} + \frac{h}{6}(k_{i1} + 2k_{i2} + 2k_{i3} + k_{i4}) \\ \text{或} \quad y_{i, m+1} &= y_{i, m} + \frac{1}{3}\left(\frac{h}{2}k_{i1} + hk_{i2} + hk_{i3} + \frac{h}{2}k_{i4}\right) \end{aligned} \quad (1-31)$$