

THOMSON
LEARNING
汤姆森学习出版集团



BookWare 系列丛书



机 构 动 态 仿 真

标准分享网 提供下载 WWW.valveinfo.CN 使用 MATLAB[®] 和 SIMULINK[®]



[美] 约翰·F·加德纳

周进雄 张陵 译

西安交通大学出版社

机 构 动 态 仿 真

使用 MATLAB® 和 SIMULINK®

BookWare 系列丛书旨在通过利用 MATLAB® 作为一个“虚拟实验室”提高读者的解题和严谨思考的能力。本丛书中的这些书为鼓励学生积极主动地学习，提供了强有力的软件工具支撑课程的各种活动，并有快捷的电子接入方式进入到相

关的习题和例题，可将读者引入到比单独由课本能深入得多的境地。本丛书用给出一种现实问题的感受、清晰展现解决实际问题的方法，并在理解和应用这些方法上挑战自我，从而为读者所青睐。BookWare 系列丛书主要包括了电气工程方面的课程，并拥有本国和世界范围内的著名作者。

全新版权引进 BookWare 系列丛书

1. 现代通信系统——使用 MATLAB®
2. 反馈控制问题——使用 MATLAB® 及其控制系统工具箱
3. 现代线性系统——使用 MATLAB®
4. 数字信号处理——使用 MATLAB®
5. 自动控制：反馈的力量——使用 MATLAB®
6. 机构动态仿真——使用 MATLAB® 和 SIMULINK®

要了解本丛书的更多信息

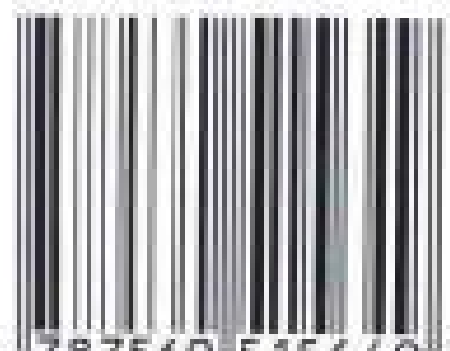
请访问 BookWare Companion Resource Center

网址：<http://www.Brookscole.com/engineering/ee/bookware.htm/>

Brooks/Cole — 重铸工程教育

责任编辑：王培雷 赵群平 封面设计：伍 迪

ISBN 7-5605-1564-9



9 787560 515649 >



ISBN 7-5605-1564-9/O · 187 定价：13.50 元

THOMSON
LEARNING
汤姆森学习出版集团

 Book Ware 系列丛书

机构动态仿真

——使用 MATLAB 和 SIMULINK

Simulations of Machines
Using MATLAB and SIMULINK

[美] 约翰·F·加德纳

周进雄 张陵 译



西安交通大学出版社
Xi'an Jiaotong University Press

内 容 提 要

本书系 Brooks/Cole 出版公司(Thomson Learning 出版集团的下属子公司)2001 年推出的 BookWare 系列丛书(BookWare Companion Series™)之一。该书提出了利用 MATLAB 及 Simulink 工具箱在计算机上解决机构动态建模及仿真问题的分析思路、方法、MATLAB 脚本文件和 Simulink 仿真模块框图以及供学生自主学习研讨的习题。全书包括 9 章内容和一个关于 Simulink 入门辅导的附录,分别是引言和概述、矢量环及矢量链方程、位置问题的求解、运动学的 Simulink 仿真、动力学引论、联立约束法、双连杆平面机器仿真、可变机构仿真、抛石机仿真。

本书可作为已具备 MATLAB 基本知识的机械、能动、土木、汽车、力学、航空航天等专业高年级本科生和研究生相关课程的参考书和补充教材,也可供有关教师、工程技术人员参考使用。

First published by Brooks/Cole, a division of Thomson Learning.

All Rights Reserved.

Authorized Translation/Adaptation of the edition by Thomson Learning and XJUP. No part of this book may be reproduced in any form without the express written permission of Thomson Learning and XJUP.

Simulations of Machines Using MATLAB and Simulink/John F. Gardner)

本书中文简体字版由美国 Thomson Learning 出版集团授权西安交通大学出版社出版发行,未经出版者书面许可,不得以任何方式复制和抄袭本书的任何部分。

版权所有,翻印必究。

图书在版编目(CIP)数据

机构动态仿真:使用 MATLAB 和 Simulink/(美)加德纳(Gardner, J. F.)著;周进雄,张陵译. —西安:西安交通大学出版社,2002. 9
(BookWare 系列丛书)

书名原文:Simulations of Machines; Using MATLAB and Simulink

ISBN 7-5605-1564-9

I. 机… II. ①加… ②周… ③张… III. 计算机
辅助计算—软件包, MATLAB, Simulink
IV. TP391. 75

中国版本图书馆 CIP 数据核字(2002)第 058477 号

*

西安交通大学出版社出版发行

(西安市兴庆南路 25 号 邮政编码:710049 电话:(029)2668315)

陕西省轻工印刷厂印装

各地新华书店经销

*

开本:727 mm×960 mm 1/16 印张:9 字数:155 千字

2002 年 9 月第 1 版 2002 年 9 月第 1 次印刷

印数:0001~5 000 定价:13.50 元

陕版出图字:25—2002—078 号

发行科电话:(029)2668357,2667874

中译本出版者的话

Brooks/ Cole 出版公司(Thomson Learning 出版集团的下属子公司)于 2000 年推出的系列丛书 BookWare Companion Series,大致涵盖的都是电子信息和电气工程类大学本科的主干课程。参加这套丛书编撰工作的大多是一些目前仍活跃在教学第一线的知名教授。组织出版这套丛书的初衷和每一位作者的想法可在“出版者的话”和每本书的作者“前言”中略见其详,这里不再赘述。

概括起来,原丛书出版者的意图大致有二:一是探讨如何将基于计算机的学习工具融于教学;二是给学生提供另外的学习空间,让他们能自主学习。总的目的就是要让学生积极主动地学习,这样会学得更好。由于课程性质和每位作者的想法有些差异,每本书在具体构成方式上会各有不同,但综合来看,这套丛书具有如下特点:

- 不求过细,而强调内容的系统性并注重相互联系。提纲挈领,注意总结,联系实际。因此,一般来说最好用作辅助性教材。

- 用“解说题”、“例题”、“强化题”、“探究题”等这些目的各异的题来演绎和深化概念,联系实际,并在重要概念的深度和广度上作适当延伸。这比“硬灌”会收到更好的效果。

- 全部题都有答案(或题解),如有必要均配有相应的 MATLAB 脚本,便于学生在学习中及时获得反馈。

- 丰富的 MATLAB 脚本资源和可随时进入的在线网站为广大读者提供了很大的学习空间。使他们的主要精力不再耗在编程上,而在探究上;他们可以利用这些脚本资源做他们想做的事;他们可以在计算机仿真上得到初步训练和更大的领悟,并为他们的创新意识和才能的发挥提供无限的空间。

根据我国当前高等学校的具体情况,我们组织出版这套丛书的中译本大致有如下几点考虑:

- 近几年来经国家和各级政府的重点建设和投资,高等学校的计算机硬件条件大为改善, MATLAB 软件已较为普及,有的学校已将宽带接入因特网联到学生宿舍,个人拥有一台 PC 机已不再稀罕,这都为使用这套丛书准备了必要的硬件和软件环境。

• 以 CAI 开始的教学手段现代化改革已持续了十多年,当今多媒体教学、电子教材、网络大学和网络课程等发展势头方兴未艾,其核心问题就是如何有效地利用计算机作为辅助教学工具,改进教学方法。这套丛书提供了一种可供大家借鉴的方式。

• 减少课内学时,精简内容,重组课程体系就是为了给学生留下更多的空间让他们自主学习。这套丛书为他们提供了这方面的素材。

• MATLAB 是一种集成度很高的语言,由于其功能强,使用方便和适用的计算机平台宽而被大家所接受,已在教学、科研和工程实际中广泛采用。这套丛书为学生熟悉、掌握和熟练应用 MATLAB 提供了一个载体,丛书中所给出的各种 MATLAB 脚本和仿真软件是科学研究工作者和工程实践人员不可多得的宝贵资源。

• 对于非全日制大学的那些成人教育、继续教育和自学考试的学生来说,这也是一套很有参考价值的辅助教材。

我社早在 20 世纪 80 年代中期就组织翻译出版了一套“外国教材精选”系列丛书,90 年代又陆续推出了《信号与系统》、《离散时间信号处理》等多本很有影响的外国优秀教材的中译本。这些都取得了良好的经济效益和社会效益。当前在国家教育主管部门正大力倡导在计算机等信息学科积极引进并直接采用外国优秀教材进行教学的时候,我们将一如既往,争取在较短的时间内将这套丛书的中译本奉献给大家。我们感谢广大读者过去对我社组织翻译出版的外国教材的青睐,并继续期望得到你们的支持。如有任何建议和要求,请与我们联系。E-mail: Shadwell.qin@163.com 或 lp-zhao@sohu.com

西安交通大学出版社

2001.8

译者的话

本书作者约翰·F·加德纳(John F. Gardner)教授现为美国博伊西州立大学机械工程系主任。他是机械建模、仿真与控制领域的专家。有关这本书的情况已集中在“中译本出版者的话”中作了介绍,更详细的可参看作者“前言”。

本书的翻译分工如下:第1、2章由张陵翻译,第3章由张红艳翻译,第7章由王学明翻译,第9章由张艳芬翻译,其余各章及附录由周进雄翻译。最后,全书由周进雄、张陵校对统稿。

由于译者水平有限,缺点和错误在所难免,恳请读者及同行批评指正。

周进雄 张陵

2002年4月

于西安交通大学

出版者的话

——关于这套丛书

计算的目的在于它的内涵,而不在数字。

—— R. W. 海明(Hamming), 选自《面向工程师和科学家的数值方法》(Numerical Methods for Engineers and Scientists, McGraw-Hill, Inc.)

正是由于心灵的驱使,我们才推出了这套“BookWare 系列丛书”(BookWare Companion Series™)。

新技术和现代方法正日益融入业已浓缩并具有重要理论的课程,因此,很多教师现在要问:“我们只是将这些最新技术教授给学生呢?还是教会他们如何推理?”我们深信,这两方面不应互相排斥。事实上,本丛书就是基于这样的信念:计算机求解和基本理论是能够互补的。应用恰当,计算结果可以阐明理论,并帮助学生以各种富有意义的方式进行思考、分析和推理。它也能够帮助学生理解新的信息和现有知识之间的关系和联系,以及培养他们解决问题的技巧、感性认识和积极主动思考的能力。为此使命,这套“BookWare 系列丛书”便应运而生。

具体地说,本丛书是专为这样的教师——他们想要把计算机辅助学习融合到课程之中去——而编写的;也是为想比仅从教科书中学到更多知识的学生编写的。教师可以从中发现,功能很强的软件工具是支撑他们课程活动的各种手段,而不会限制他们的应用。学生会发现,可以迅速获得各种相关的习题和例题,并具有电子接入方法。对教师和学生来说均为重要的是这套丛书基于这样的前提:如果学生积极主动学习,他们会学得更好。“BookWare 系列丛书”定会受到学生的青睐,为他们提供一种对现实问题的感受,展示解决实际问题的清晰方法,并在理解和应用这些方法上挑战自我。

为了更好地满足你的需要,我们将继续寻找使这套丛书日臻完善的方法。为达此目的,请加入到我们的“BookWare Companion Resource Center”中来,我中心的网址是:<http://www.brookscole.com/engineering/bookware/>^①

^① 原书提供的网址是 <http://www.brookscole.com/engineering/ee/bookware.html>——译者注。

你推荐的方法都会使这套丛书更加令人满意,在课堂教学中与同仁们分享你如何使用技术的见解,为下一版提出具体的习题或例题,或者将你的想法告诉我们。务请不吝赐教。

我们恭候你的回音,谢谢你的持续不断地支持。

Bill Stenquist	出版人	<i>bill.stenquist@brookscole.com</i>
Heather Woods	责任编辑	<i>heather.woods@brookscole.com</i>
Nathan Wilbur	市场经理	<i>nathan.wilbur@brookscole.com</i>
Christina DeVeto	市场助理	<i>christina.deveto@brookscole.com</i>

前 言

本书主要是为大学生学习机械知识而提供的一本辅助教材。并且,本书也可以作为高年级学生学习动力学的一本人门选修课教材,如同我们在宾夕法尼亚州立大学的做法。本书的观点来源于我和我的同事关于如何在大学生运动学课程中引入动力学知识的讨论。通常,大学生运动学的学习是严格按部就班的,学生被一步步地引导去求解位置、速度和加速度。只是当详尽的运动分析掌握之后才引入力以及动力学的基本概念。在分析力之前就分析物体的运动,这种很明显的矛盾是由机构的约束特性以及某些通常未阐明的假设所产生的,这些假设认为存在一个理想的原动机,能够提供维持某个连杆匀速运动所需任意大小的扭矩或力。真正意义上的约束机构动态仿真被认为超出了大学经典运动学知识的范围。然而,这些看法已经有些过时。

在本书中,我们将利用一类以 MATLAB 及其附加软件 Simulink 为代表的面向运动学的计算机软件包。通过综合闭环矢量方程(描述运动约束)、数值仿真(在加速度已知时计算速度和位移)以及矩阵代数(同时计算加速度和约束反力)等内容,本书将引导学生掌握完成一个机构多功能动态仿真所需的基本步骤。这类仿真是很重要的一种设计工具,同时它可为工程技术人员提供设计轴承和构件尺寸时所需的机构内力。

尽管有些教师可能会把本书作为一本独立使用的教材或者作为有实践经验的工程人员的一本参考书,但本书主要是作为大学机械类课程的一本辅助教材。虽然本书的内容是按章依次介绍的,但各章内容相对独立且其先后顺序并不与前面章节的内容相关,例如第 2 章概述了闭环矢量方程及其导数,如果学生已经对这部分知识很熟悉了,就可以跳过这章内容。类似地,第 3 章论述了使用 MATLAB 求解运动学中的非线性位置问题,由于这一问题已在仿真软件隐含求解,因而该章内容也可以跳过去。

下列表格的内容可指导教师如何利用该书各章的内容辅助这一领域其他教科书的学习。

本书与其他作者的通用运动学教材内容的章节对照表

本书 (章)	Shickley & Uicker (章)	Erdamn & Sandor (章)	Mabie & Reinholtz (章)	Norton (章)	Waldron & Kinzel (章)
2	2,3	3	2	2,4	3
3	2	3	2	4	2,3
4	3,4	3,4	8	6,7	2
6	14,15,16	5	9	10,11	11,12
例子研究	5,17,18	6,8	3,11,12	8,9,13,15	

当我写此书时, MATLAB 和 Simulink 已经升级了好几个版本。写作初期我主要依据 MATLAB 4.3 和 Simulink2.0。当书稿完成时, 我已经开始使用 R11.1 版本的 MATLAB/Simulink 软件了。我曾经试图使编写的程序能够向下兼容, 但写作期间版本的升级已经影响了 Simulink 核心的数值积分算法。因此, 为了获得最佳的仿真效果, 我建议读者使用最新版本的软件。

致谢

我要感激帮助过我的朋友和同事, 从他们那里我得到了许多帮助和支持, 由于人数太多, 这里已无法一一列举。我要感谢 John Lamancusa、Joe Sommer 和 Marty Trethewey, 他们的意见和观点启发了本书的写作思路。加州大学戴维斯分校的 Steve Velinsky 在我公休假期间接待了我, 在这段时间完成了本书的大部分内容。本书 Brooks/Cole 公司的编辑 Jonathan Plant 和 Bill Stenquist 鞭策和鼓励我完成了本书的写作。加利福尼亚工业大学(圣路易斯-奥比斯波)的 Bill Murray 在他的课程教学中试用了本书的初稿, 他和他的学生为本书提出了许多很好的意见和建议。Rensselaer 工学院的 Stephen Derby, 佐治亚理工学院的 Imme Ebert-Uphoff, 衣阿华州立大学的 Greg Luccke, 马奎特大学的 Mark Nagurka 和罗得岛大学的 William J. Palm 也提出了意见和建议。我也要感谢 Dina Berkhoff, Matt Lichter, Jenny Rincon, Will Reutzel 和 Gayathri Vijayakumar, 这几位耐心的学生参与了本书初稿的多次修改, 他们的建议使本书的终稿增色不少。

最后, 我必须感谢我的妻子 Barbara Bowling 和我的女儿 Sarah 和 Beth, 她们的支持、关爱和耐心是我永恒的力量源泉。

约翰·F·加德纳
博伊西, 爱达荷

目 录

中译本出版者的话

译者的话

出版者的话

前 言

第 1 章 引言和综述

- 1.1 为何要对机构进行仿真 (1)
- 1.2 运动学仿真 (2)
- 1.3 机构的动力学仿真 (2)
- 1.4 小结 (3)

第 2 章 矢量环与矢量链方程

- 2.1 引言 (4)
- 2.2 平面矢量 (4)
- 2.3 单个闭环方程 (5)
- 2.4 矢量方程的求导 (6)
- 2.5 其他常见的机构 (10)
- 2.6 矢量链 (10)
 - 2.6.1 两连杆平面机器人 (10)
 - 2.6.2 用矢量链描述任意点的运动 (12)
- 2.7 小结 (14)
- 第 2 章习题 (14)

第 3 章 位置问题的求解概述

- 3.1 概述 (16)
- 3.2 非线性代数方程的数值解法 (16)
- 3.3 四连杆机构的位置问题 (17)

3.4	四连杆机构位置问题的 MATLAB 求解	(18)
3.5	位置解与初始估计	(21)
3.6	小结	(24)
	第 3 章习题	(24)

第 4 章 运动学仿真——Simulink 的使用

4.1	什么是运动学仿真	(25)
4.2	通过运动学仿真求解速度	(25)
4.2.1	曲柄滑块机构的闭环矢量方程	(25)
4.2.2	曲柄滑块机构运动学的 Simulink 仿真	(27)
4.2.3	建立初始条件	(30)
4.2.4	仿真结果	(32)
4.3	通过运动学仿真求解加速度	(34)
4.3.1	考虑加速度的仿真	(34)
4.3.2	运行曲柄滑块机构的仿真	(37)
4.4	相容性检验	(40)
4.5	四连杆机构的运动学仿真	(43)
4.6	小结	(47)
	第 4 章习题	(47)

第 5 章 动力学引论

5.1	概述	(49)
5.2	第 1 步:滑块在斜面上的仿真	(49)
5.3	第 2 步:添加摆	(51)
5.4	第 3 步:矩阵方程的组装	(53)
5.5	第 4 步:建立动态仿真	(53)
5.6	第 5 步:设置初始条件并运行仿真程序	(55)
5.7	小结	(57)
	第 5 章习题	(57)

第 6 章 联立约束法

6.1	引言	(59)
6.2	方法概述	(59)
6.2.1	力方程	(60)

6.2.2	闭环矢量方程	(61)
6.2.3	质心加速度的矢量方程	(61)
6.2.4	动态仿真的实现	(62)
6.3	联立约束法在曲柄滑块机构中的应用	(62)
6.3.1	力方程	(63)
6.3.2	闭环矢量方程	(65)
6.3.3	质心加速度	(65)
6.3.4	系统方程的组装	(65)
6.4	曲柄滑块机构的动态仿真	(65)
6.5	曲柄滑块机构的仿真研究	(70)
6.6	小结	(73)
	第6章习题	(73)

第7章 两连杆平面机器人

7.1	概述	(75)
7.2	矢量方程	(75)
7.3	动力学方程	(76)
7.4	联立约束矩阵	(78)
7.5	动态仿真	(79)
7.6	机器人坐标控制	(82)
7.7	小结	(83)
	第7章习题	(83)

第8章 可变机构的仿真

8.1	间歇机构	(84)
8.2	小结	(89)

第9章 抛石机

9.1	概述	(90)
9.2	矢量环	(92)
9.3	运动方程的建立	(93)
9.4	矩阵方程	(94)
9.5	动态仿真	(94)
9.6	仿真结果	(97)

9.7 小结	(99)
第9章习题	(99)

附录

Simulink 入门辅导	(100)
A.1 启动 Simulink	(100)
A.2 一个简单模型的建立	(101)
A.3 仿真运行	(104)
A.4 仿真运行时间参数	(104)
A.5 初始条件	(106)
A.6 多路信号	(108)
A.7 Simulink 和 MATLAB: 返回数据到工作环境	(110)
A.8 使用 MATLAB plot 命令	(113)
A.9 在 Simulink 中使用 MATLAB 函数	(115)
A.9.1 MATLAB 函数	(115)
A.9.2 从 Simulink 中调用函数	(118)
A.9.3 使用多输入和多输出	(120)
A.10 结束语	(125)

索引

第1章 引言和综述

1.1 为何要对机构进行仿真

计算机仿真一词是指在研究中利用数学模型来获取系统的一些重要特性参数,这些数学模型通常是由以时间为变量的常微分方程来描述,并用数值方法进行计算机仿真求解的。利用计算机仿真可以对整个机械制造系统及过程进行广泛地研究,如从内燃机到原料市场。本书的宗旨是研究如何利用计算机仿真技术进行机构的运动与动力分析。

一般而言,机构设计的目标之一是能够实现某一预先设定的运动轨迹。在研究机械系统的运动规律时,借助于计算机仿真是十分有益的。但是,大多数有关动力学方面的教科书较少涉及到计算机仿真,只是某些需要对研究对象进行运动方程精确描述的高等动力学教材(如拉格朗日力学或哈密尔顿力学)才对计算机仿真有所体现。本书的主线有两条:一是帮助读者使用已有的计算机仿真软件包来求解动力学方程,即动力学仿真;二是帮助读者利用在大学二年级力学课程中所学的牛顿力学方法建立起完整的动力学仿真数学模型。

在本书中,计算机仿真工具将分为两个层次加以介绍。第一,采用数值积分这一计算机仿真的核心技术,从而回避了机械构件的位置问题,而该问题被认为是运动分析(包括位置分析、速度分析和加速度分析)最困难的部分。在这一部分,读者将了解到如何使用 Simulink 仿真工具箱对具有恒定输入速度和可变输入速度两种工况的机构进行仿真和分析,甚至加速度分析。第二,拓展和提高部分:借助文献^①所述的方法,将闭环矢量方程与动力学的牛顿-欧拉公式相组合,构成一组线性方程并写成矩阵形式,然后进行数值积分,计算出仿真时域内每一时刻的加速度和作用力。这种方法所附带的好处就是提供了更为详尽的运动坐标,其数目多于能够唯一确定机构位置所需的坐标个数。通过对坐标之间的相容性检验,就能够对程序调试和数值积分在执行过程中各自产生的误差进行监测。

^① Huang E.J. Intermediate Dynamics. Englewood Cliffs, N.J., Prentice Hall, 1992.

1.2 运动学仿真

在本书中,运动学仿真一词的含义是指利用计算机仿真软件包求解机构在所计时间域内的运动学方程,以此确定机构在连续运动过程中各个构件的速度和加速度。运动学仿真要预先知道机构的几何特性,即各构件的长度。图 1-1 为运动学仿真的流程图,它描述了诸如 MATLAB 仿真工具箱 Simulink 等仿真软件包的仿真步骤和流程。

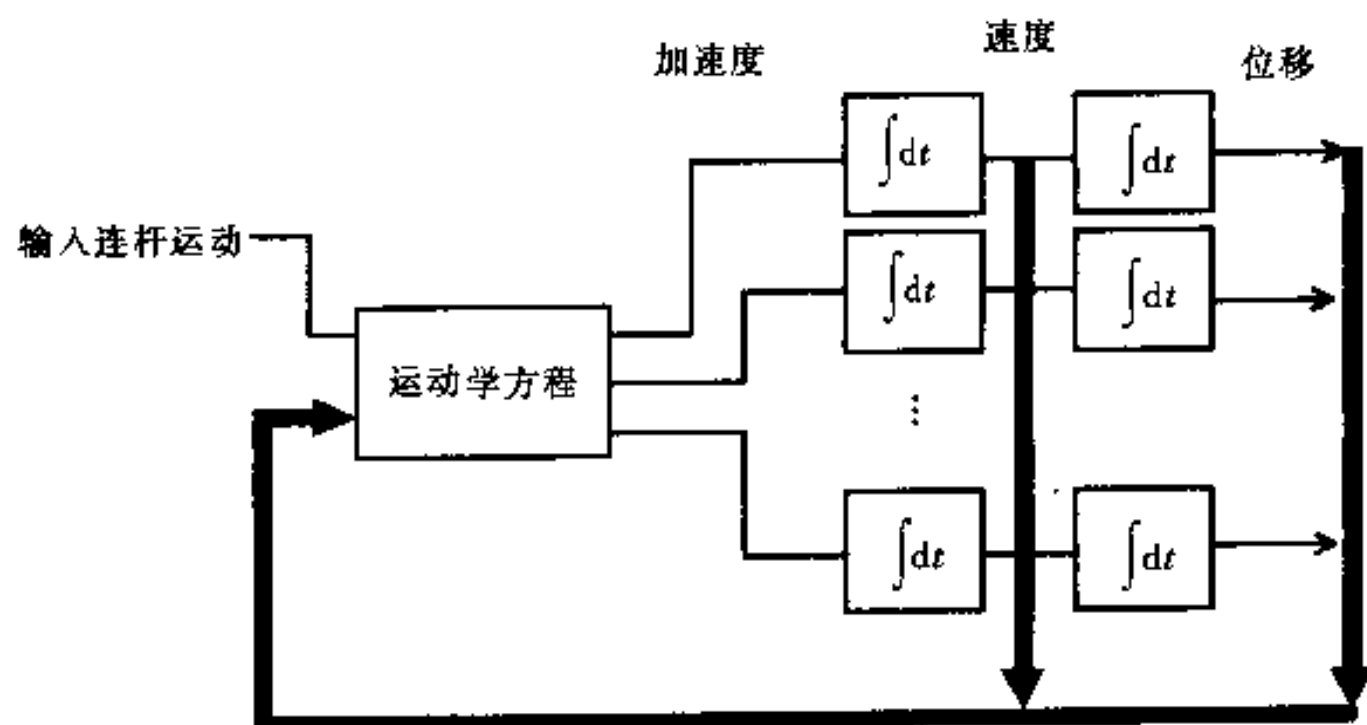


图 1-1 运动学仿真流程框图

在图 1-1 中,标有“运动学方程”的模块为用户自定义程序块,主要用于求解描述机构各构件相互之间加速度关系的运动方程。在第 2 章将要介绍的闭环矢量方程是计算这些运动方程的基础。

运动学仿真方法与传统的运动学问题计算机求解存在着显著的区别。在本书中,仿真软件包中所用的数值积分算法需根据加速度来计算速度和位移。换言之,在该功能模块中必须含有加速度分析和计算。此外,数值积分还应能提供所需的全部速度和位移信息。第 4 章将详尽地讨论这一方法,并以曲柄-滑块机构和四连杆机构为例来阐明其实现过程。

1.3 机构的动力学仿真

如前所述,根据以往的观点,机构的动力学全过程仿真超出了大学低年级学生的动力学教科书范围。实际上,对于含约束系统使用经典的拉格朗日方法进行分析更适合于研究生的教学。另一方面,大学低年级学生对于将牛顿力学用于多刚

体系问题则显得更为熟练。因此,一种更为合理的动力学仿真方法就是应用牛顿力学对每个单独的构件进行力学分析,根据描述机构构件加速度的约束条件建立起力与加速度之间的关系,由此构成关于构件加速度(平动和转动)和约束力的齐次线性代数方程组。再将方程组写成矩阵形式,就可以由仿真软件包求出每一时间的力与加速度。

值得注意的是,该方法与经典的拉格朗日力学或哈密尔顿力学有一点本质上的区别。在经典方法中,运动方程需要进行降阶。换句话说,由于一个四连杆机构只有一个自由度,因而它的动力学表达式是单一的二阶运动微分方程;另一方面,本书描述的计算机仿真对每一个构件进行二阶运动微分方程的求解,因而这些方程的解是彼此相关的。因此,如同本书第4章和第6章所述,这实际上提供了仿真正确性和一致性的外在检验。

图1-2所示的流程框图描述了系统的计算机仿真过程。

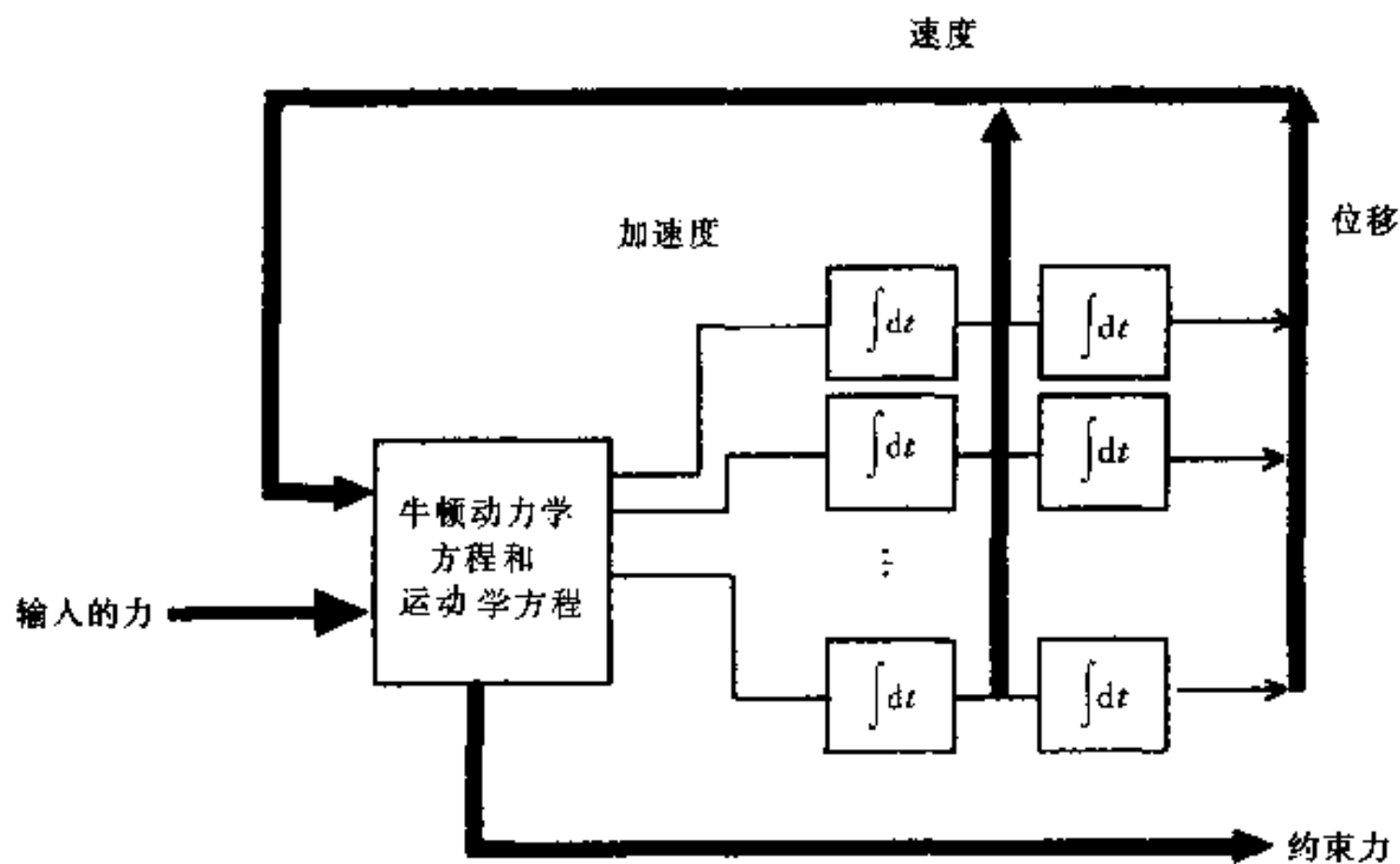


图 1-2 使用联立约束法的机构动力学仿真流程框图

1.4 小结

本书介绍了一种含约束机械系统或机构动力学分析和设计的新方法。本书主要集中阐述利用 MATLAB 仿真工具箱 Simulink 进行动力学仿真,而在第2~第4章和第6章中所介绍的基础理论,对用计算机求解动力学方程的方法而言,这并不是唯一的方法。

第2章 矢量环与矢量链方程

2.1 引言

几乎所有试图用运动学分析程序化的技术,其核心就是闭环矢量方程。该方程是机构各个构件之间连接约束的一个非常简洁而又明了的表达式。闭环矢量方程易于求解,并且是进行机构计算机分析所需采取的第一步。

与此相类似,矢量链是用来确定机构上某一点的位置。这一点并不是机构中随意的某一连接点,对于机构运动学分析而言是十分重要的点,诸如连杆的质心。矢量链也可用于开链机构(仅有一个连接点与固定连杆或地面相连)的运动学分析。

在其他有关的教科书中也能够找到与本章所述相类似的方法。对于复杂杆件以及“旋转矢量”,运动学分析实际上提供了与闭环矢量方程完全相同的数学关系式。依本书作者的观点,这一表达式比其他方法更为适用。

2.2 平面矢量

如同所有的大学工科低年级学生所知,矢量是一个数学概念,它用于表示一个具有大小和方向的物理量。简单地说,一个位移矢量表示了空间任意两点之间的有向距离。为了进行机构分析,机构中每一根连杆都可以表示为一个位移矢量,矢量的起点就是连杆的某一端点,而其另一端点就是矢量的终点。这个位移矢量的大小就是连杆的长度,矢量与 x 轴正向间的夹角就是连杆的夹角(逆时针为正),如图2-1所示。

应当注意到,矢量是用大写黑体字母 \mathbf{R} 来表示的;而矢量的长度是用小写字母 r 来表示,且不是黑体字。这种符号表达方式将贯穿全书。

如图2-1所示,若给定坐标系,则矢量 \mathbf{R} 的 x 分量与 y 分量将可以用矢量的长度 r 和它与 x 轴正向间的夹角来表示,即 $r_x = r\cos(\theta)$ 和 $r_y = r\sin(\theta)$ 。

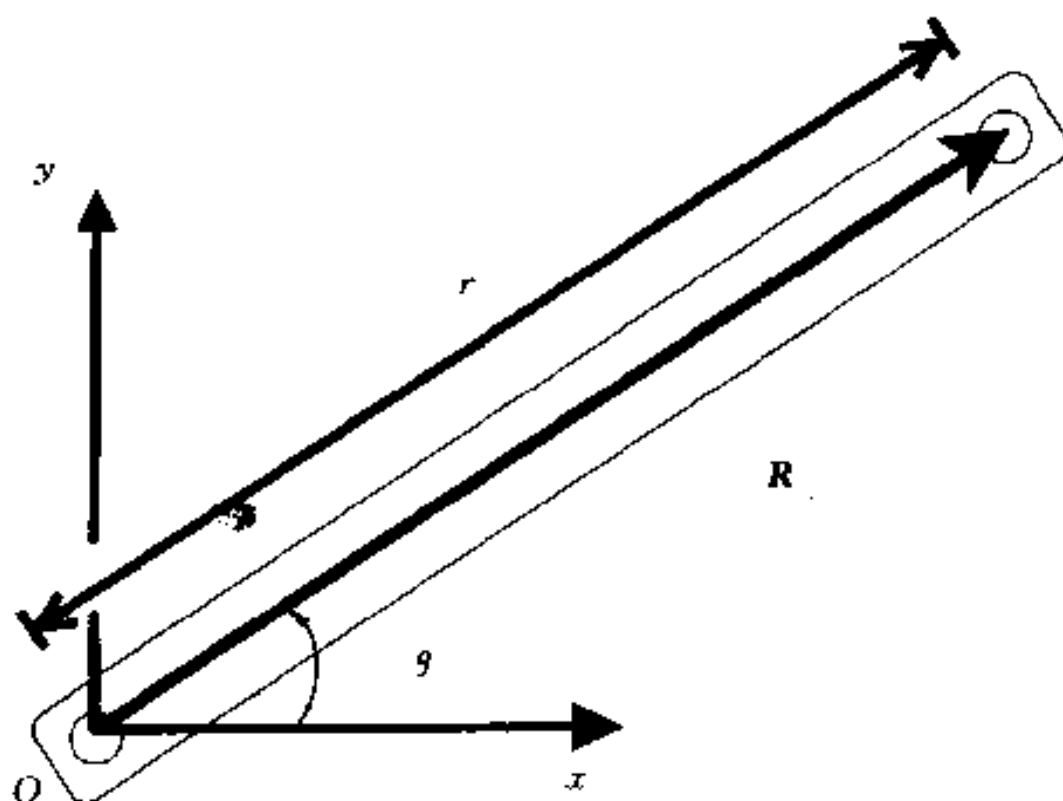


图 2-1 长度为 r 的连杆与坐标系、位移矢量 R 三者相互之间的关系

2.3 单个闭环方程

本节介绍闭环矢量方程的概念。图 2-2 为一个四连杆机构，它是运动学有关教科书中通常列举的例子。在图中，取地面为固定杆件，编号为 1；可动杆件的起始编号为 2。

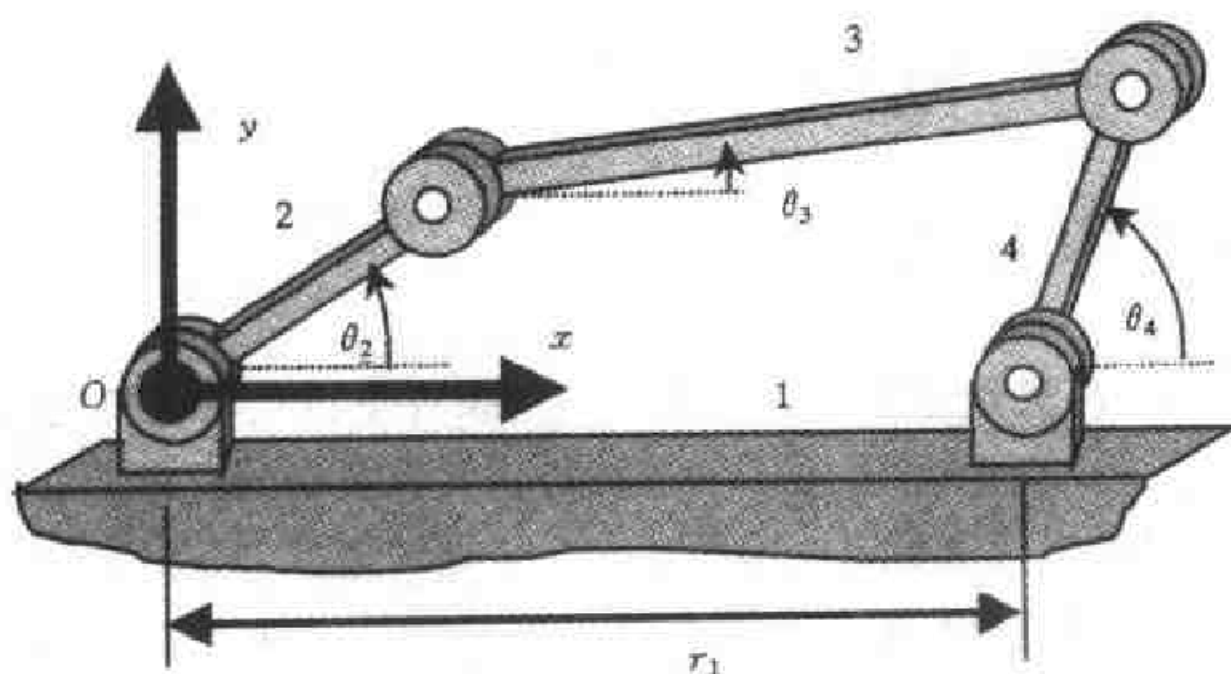


图 2-2 平面四连杆机构(左边基础支座为 x, y 坐标系原点, x 坐标轴与两基础支座连线相重合)

为方便起见,设定机构的 $x-y$ 坐标系如图 2-2 所示,坐标原点按惯例取在左

端固定支座处。如前所述,每一根连杆(包括固定杆件)均可由一位移矢量来表示。因此,图 2-3 仍为一个四连杆机构,只不过用位移矢量取代了连杆。由于在同一坐标系下位移矢量有多种排序表示方法,所以各位移矢量的编号顺序应遵循一定的约定。即,对于任意一组位移矢量,应当构成一个易于正确表达和便于推导的闭环矢量方程。因此,根据矢量加法的要求,矢量 R_2 和 R_3 应当首尾相连,矢量 R_1 和 R_4 亦是如此。注意到矢量 R_3 和 R_4 的矢端都在机构的同一点 B ,因此这表明两对矢量 (R_2, R_3) 和 (R_1, R_4) 各自相加的结果应相等,其数学表达式为

$$R_2 + R_3 = R_1 + R_4 \quad (2-1)$$

方程(2-1)表明,矢量 R_2 和 R_3 相加而得到的位移矢量与矢量 R_1 和 R_4 叠加得到的位移矢量是完全相同的;无论机构运动到何种状态,只要能够保证机构的几何装配条件,则这个闭环矢量方程就一定能够成立。

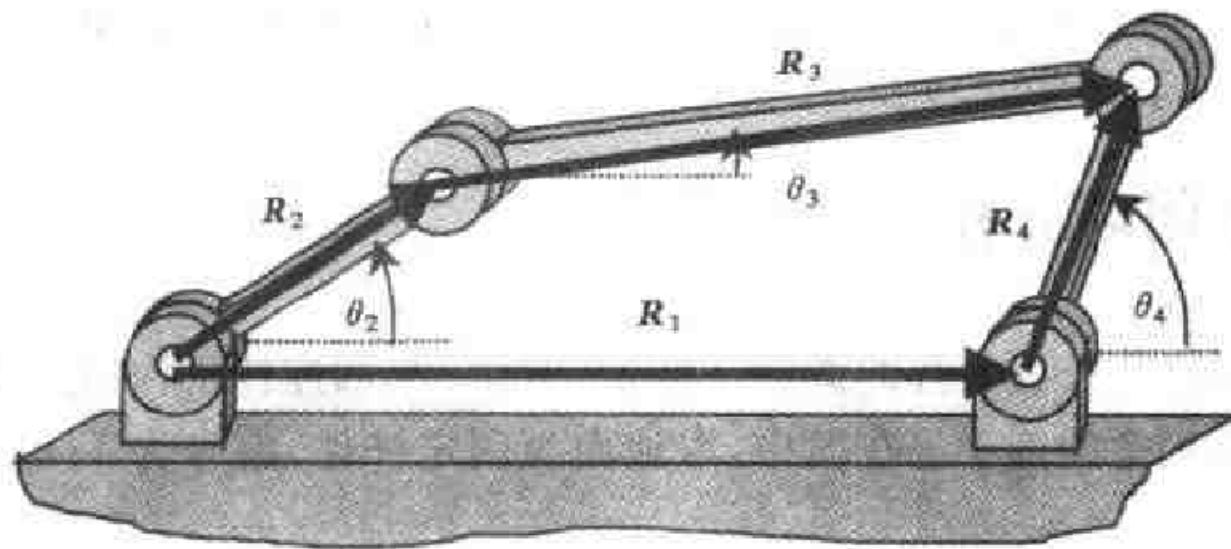


图 2-3 平面四连杆机构的位移矢量表示法(各连杆和相应的位移矢量与水平坐标轴之间的夹角均相同)

2.4 矢量方程的求导

上节推导的闭环矢量方程是针对某种约束条件下机构各连杆之间一个紧凑、简洁的关系表达式。换句话说,由于四连杆机构具有单一的自由度,因此由矢量方程(2-1)就可得知,当任意确定机构的某一个节点转角时,就必然同时限定了其余两个转角。由于运动学分析大部分内容涉及的是机构的速度与加速度计算,因此必然就会遇到对矢量方程(2-1)求时间导数。

显而易见,各个矢量是随时间而变化的。因为,即使各个连杆的长度保持不变,但它们各自的方位(矢量的指向)却是随机构运动而改变的。对于其他一些机构,可能位移矢量的大小和方向均会改变。矢量方程对时间求导数的简单方法是将闭环矢量方程分解成为两个标量表达式;一个沿 x 方向分解,另一个沿 y 方向分解。

根据图 2-1 中关于矢量角度的定义, 标有各矢量夹角的闭环矢量方程几何表达式如图 2-4 所示。必须注意, 若由 x 正向旋转到矢量的矢端为逆时针转动, 则该矢量的角度为正。此外还需注意, 矢量角度的表示应与连杆夹角的表示相一致。

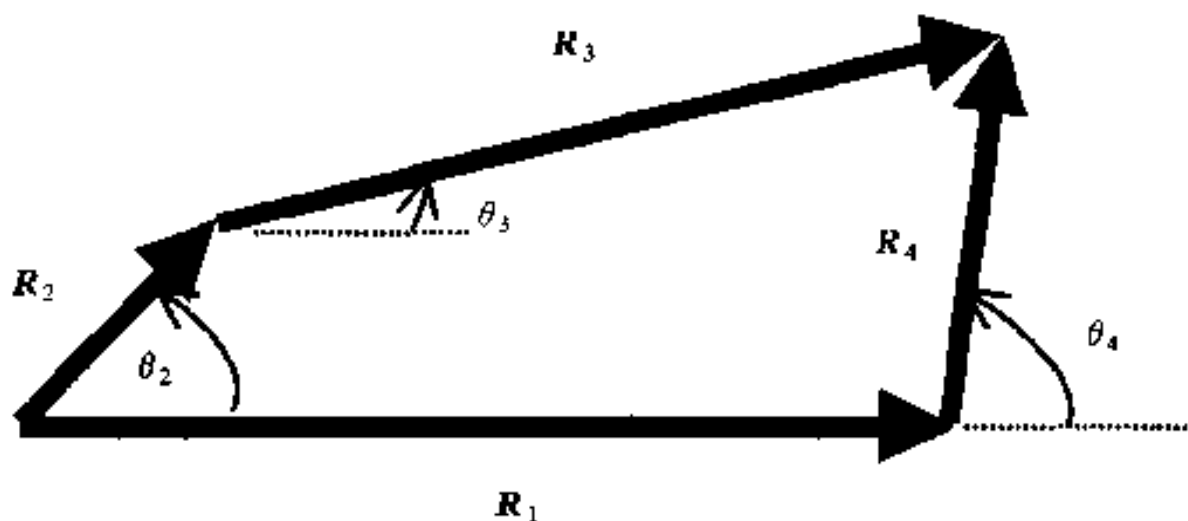


图 2-4 四连杆机构的闭环矢量图(注意矢量的夹角与图 2-3 所示连杆夹角是相同的)

如果上述约定和假设能够严格遵守, 则利用矢量夹角的正旋和余旋函数就可得到闭环矢量方程的两个分量表达式, 即

$$r_2 \cos\theta_2 + r_3 \cos\theta_3 = r_1 \cos\theta_1 - r_4 \cos\theta_4 \quad (2-2)$$

$$r_2 \sin\theta_2 + r_3 \sin\theta_3 = r_1 \sin\theta_1 + r_4 \sin\theta_4 \quad (2-3)$$

稍后再来讨论这些方程。首先, 对于这一特定机构, $r_1 \sim r_4$ 代表了各个连杆的长度, 在整个分析过程中是保持不变的。机构有四个连杆夹角, 它们当中的任一个均可视为恒定并可假定为零, 因为坐标系的定位是任意假定的。基于这一点, 为了简化问题的分析过程, 取 x 坐标轴与某一矢量相重合。在图 2-4 中, 由于矢量 R_1 与 x 坐标轴相重合, 所以 θ_1 夹角恒为零。

由于其他三个矢量夹角均随时间变化, 所以必须考虑对其求时间导数。方程 (2-4) 和 (2-5) 是方程 (2-2) 和 (2-3) 对时间求一阶导数得到的结果。这两个方程确定了四连杆机构中三个可动连杆转动速率之间存在的制约关系。需要指出的是, 方程 (2-4) 和 (2-5) 以及本书随后的内容中, 假设角度 θ_i 对时间的一阶导数用变量 ω_i 表示。与此相类似, 角速度 ω_i 对时间的一阶导数用角加速度 α_i 表示。方程 (2-2) 和 (2-3) 对时间求一阶导数后得

$$-\omega_2 r_2 \sin\theta_2 - \omega_3 r_3 \sin\theta_3 = -\omega_4 r_4 \sin\theta_4 \quad (2-4)$$

$$\omega_2 r_2 \cos\theta_2 + \omega_3 r_3 \cos\theta_3 = \omega_4 r_4 \cos\theta_4 \quad (2-5)$$

在许多分析中, 通常假定某一连杆以匀角速度转动。例如, 假定连杆 2 与一台电机相连, 该电机能够提供足够大的驱动力矩以使得 ω_2 相对保持常量。在此条件下, ω_2 就称之为机构的输入, 并且方程 (2-4) 和 (2-5) 可重新写成

$$-\omega_3 r_3 \sin\theta_3 + \omega_4 r_4 \sin\theta_4 = \omega_2 r_2 \sin\theta_2 \quad (2-6)$$

$$\omega_3 r_3 \cos\theta_3 - \omega_4 r_4 \cos\theta_4 = -\omega_2 r_2 \cos\theta_2 \quad (2-7)$$

在传统的动力学分析中,确定方程(2-6)和(2-7)中的角速度在某时刻的大小是在求解位置问题之后进行的。换言之,在某一时刻所有连杆转角的角度是已知的。第3章将详细讨论位置问题的求解方法,在此仅假定各连杆转角为已知。因此,方程(2-6)和(2-7)可以写成以下矩阵形式

$$\begin{bmatrix} -r_3 \sin\theta_3 & r_4 \sin\theta_4 \\ r_3 \cos\theta_3 & -r_4 \cos\theta_4 \end{bmatrix} \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} \omega_2 r_2 \sin\theta_2 \\ -\omega_2 r_2 \cos\theta_2 \end{bmatrix} \quad (2-8)$$

闭环矢量方程的二阶导数也是十分有用并可容易导出的,只需记住速度方程中的各项是两个时间变量(ω 和 $\cos(\theta)$)的乘积以及求导规则。因此,得到

$$-\alpha_3 r_3 \sin\theta_3 - \omega_3^2 r_3 \cos\theta_3 + \alpha_4 r_4 \sin\theta_4 + \omega_4^2 r_4 \cos\theta_4 = \alpha_2 r_2 \sin\theta_2 + \omega_2^2 r_2 \cos\theta_2 \quad (2-9)$$

$$\alpha_3 r_3 \cos\theta_3 - \omega_3^2 r_3 \sin\theta_3 - \alpha_4 r_4 \cos\theta_4 + \omega_4^2 r_4 \sin\theta_4 = -\alpha_2 r_2 \cos\theta_2 + \omega_2^2 r_2 \sin\theta_2 \quad (2-10)$$

假定连杆2的输入角速度 ω_2 和角加速度 α_2 均为已知,则上述方程组的矩阵形式为

$$\begin{bmatrix} -r_3 \sin\theta_3 & r_4 \sin\theta_4 \\ -r_3 \cos\theta_3 & -r_4 \cos\theta_4 \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} \alpha_2 r_2 \sin\theta_2 + \omega_2^2 r_2 \cos\theta_2 + \omega_3^2 r_3 \cos\theta_3 - \omega_4^2 r_4 \cos\theta_4 \\ -\alpha_2 r_2 \cos\theta_2 + \omega_2^2 r_2 \sin\theta_2 + \omega_3^2 r_3 \sin\theta_3 - \omega_4^2 r_4 \sin\theta_4 \end{bmatrix} \quad (2-11)$$

例 2-1

利用方程(2-8)求解四连杆机构在图2-5所示位置时连杆3、4的角速度。各连杆长度及在图示位置的转角见表2-1。设输入连杆2的角速度为100 rad/s。

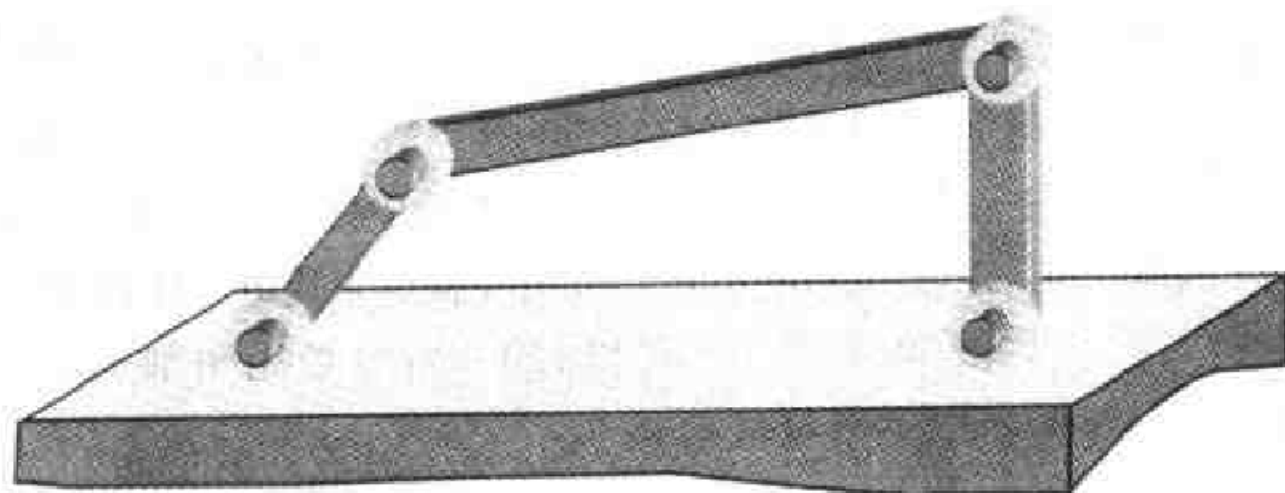


图 2-5 四连杆机构

表 2-1 四连杆机构各连杆的长度和转角

连杆	长度 (r)/cm	转角 (θ)/($^{\circ}$)
1	12.0	0
2	4.0	45
3	10.0	24.65
4	7.0	90.68

使用方程(2-8)对例 2-1 进行求解,使用 MATLAB 编制的程序如下:

```
>> r2=4;
>> r3=10;
>> r4=7;
>> th2=45*pi/180;
>> th3=24.652*pi/180;
>> th4=90.6794*pi/180;
>> %
>> % Note that the angles are in radians
>> %
>> J=[-r3*sin(th3) r4*sin(th4);r3*cos(th3) -r4*cos(th4)]
J =

-4.1711 6.9995
9.0886 0.0830

>> b=[100*r2*sin(th2);-100*r2*cos(th2)]

b =

282.8427
-282.8427

>> om34=inv(J)*b

om34 =

-31.319
21.745
```

所以,机构在图示位置的解是:连杆 3 的角速度为 31.32 rad/s,逆时针;连杆 4 的角速度为 21.74 rad/s,逆时针。

MATLAB 提示:

在 MATLAB 程序设计中,三角函数是用弧度来表示的。由于我们通常习惯于使用度为单位,因此就需要通过对角度乘以 $\pi/180$ 来进行转换,以此得到弧度单位;反之,通过对弧度乘以 $180/\pi$ 来进行转换,以此得到度数单位。MATLAB 通过设定一个恒定的、带有普遍意义的参数 pi(其值等于 π)即可容易地实现这种变换,并且可使程序的计算精度达到最大。

运用传统的多边形矢量法则即可得到这些结论,请读者自行证明。

2.5 其他常见的机构

为了方便地使用闭环矢量方程以及它对时间的导数,表 2-2 列出了几种常见机构的闭环矢量方程以及它对时间的一、二阶导数。有一点值得注意,由于坐标系的放置以及坐标原点的定位是任意的,因而闭环矢量方程的表达并不是唯一的。只要这些方法使用得合理恰当,结果就一定是正确的,并且通过反复推导这些方程,必定能够使得分析能力与数学技能得到大大加强。



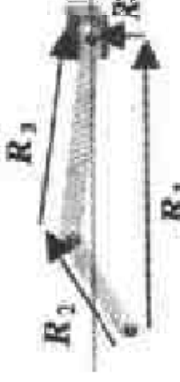

2.6 矢量链

有时,清晰准确地确定机构上除铰支座结点以外的一点或多点的运动是非常重要的。还有的时候,我们需要对具有地面连杆但未形成闭环的机构进行分析。在此情况下,矢量环是不可能实现的,但是通过确定矢量链的方式可以导出类似的关系式。这种矢量链反映了连杆坐标与某些重要坐标之间的关系。在本节中,我们将探讨两个使用矢量链的重要例子,一个是两连杆平面机器人,另一个是四连杆机构中间连杆上任一点位置的确定。

2.6.1 两连杆平面机器人

图 2-6 为一个两连杆平面机器人的示意图。这个例子在机器人学文献中经常能够遇到,它是平面机器人最为简单的形式,由两根连杆和两个转动铰所组成。需注意到,两个铰接点处的转角 θ_2 和 θ_3 仍按照先前的约定,它们是水平线与各自杆轴线形成的逆时针夹角。还需注意到,坐标系原点位于地面(连杆 1)与机器人的第一根连杆(连杆 2)的铰接点处。在该例中,重要的是确定连杆坐标 θ_2 和 θ_3 与端部操作手位置矢量 \mathbf{R}_E 之间的关系。可以看出,从坐标系原点到机器人操作手的矢量是两根连杆的位移矢量之和,即

表 2-2 四种常见机构的闭环矢量方程

机构	矢量方程	速度方程	加速度方程
<p>四连杆机构</p> 	$R_2 + R_3 = R_1 + R_4$	$\begin{bmatrix} -r_3 \sin \theta_3 & r_4 \sin \theta_4 \\ r_3 \cos \theta_3 & -r_4 \cos \theta_4 \end{bmatrix} \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} \omega_2 r_2 \sin \theta_2 \\ -\omega_2 r_2 \cos \theta_2 \end{bmatrix}$	$\begin{bmatrix} a_2 r_2 \sin \theta_2 + r_2 \omega_2^2 \cos \theta_2 + \\ r_3 \omega_3^2 \cos \theta_3 - r_4 \omega_4^2 \cos \theta_4 \\ -a_2 r_2 \cos \theta_2 + r_2 \omega_2^2 \sin \theta_2 + \\ r_3 \omega_3^2 \sin \theta_3 - r_4 \omega_4^2 \sin \theta_4 \end{bmatrix} = \begin{bmatrix} a_3 \\ a_4 \end{bmatrix}$
<p>曲柄滑块机构(滑块运动轴线过曲柄旋转中心)</p> 	$R_2 + R_3 = R_1$	$\begin{bmatrix} 1 & r_3 \sin \theta_3 \\ 0 & -r_3 \cos \theta_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} -\omega_2 r_2 \sin \theta_2 \\ \omega_2 r_2 \cos \theta_2 \end{bmatrix}$	$\begin{bmatrix} 1 & r_3 \sin \theta_3 \\ 0 & -r_3 \cos \theta_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} -a_2 r_2 \sin \theta_2 - r_2 \omega_2^2 \cos \theta_2 - r_3 \omega_3^2 \cos \theta_3 \\ a_2 r_2 \cos \theta_2 - r_2 \omega_2^2 \sin \theta_2 - r_3 \omega_3^2 \sin \theta_3 \end{bmatrix}$
<p>偏置曲柄滑块机构</p> 	$R_2 + R_3 = R_1 + R_4$	$\begin{bmatrix} 1 & r_3 \sin \theta_3 \\ 0 & -r_3 \cos \theta_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} -\omega_2 r_2 \sin \theta_2 \\ \omega_2 r_2 \cos \theta_2 \end{bmatrix}$	$\begin{bmatrix} 1 & r_3 \sin \theta_3 \\ 0 & -r_3 \cos \theta_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} -a_2 r_2 \sin \theta_2 - r_2 \omega_2^2 \cos \theta_2 - r_3 \omega_3^2 \cos \theta_3 \\ a_2 r_2 \cos \theta_2 - r_2 \omega_2^2 \sin \theta_2 - r_3 \omega_3^2 \sin \theta_3 \end{bmatrix}$
<p>急回机构</p> 	$R_1 + R_3 = R_2$	$\begin{bmatrix} \cos \theta_3 & -r_3 \sin \theta_3 \\ \sin \theta_3 & r_3 \cos \theta_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_3 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} -\omega_2 r_2 \sin \theta_2 \\ \omega_2 r_2 \cos \theta_2 \end{bmatrix}$	$\begin{bmatrix} \cos \theta_3 & -r_3 \sin \theta_3 \\ \sin \theta_3 & r_3 \cos \theta_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_3 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} -a_2 r_2 \sin \theta_2 - r_2 \omega_2^2 \cos \theta_2 + \\ 2\dot{\theta}_3 \omega_3 \sin \theta_3 + r_3 \omega_3^2 \cos \theta_3 \\ a_2 r_2 \cos \theta_2 - r_2 \omega_2^2 \sin \theta_2 - \\ 2\dot{\theta}_3 \omega_3 \cos \theta_3 + r_3 \omega_3^2 \sin \theta_3 \end{bmatrix}$

$$\mathbf{R}_E = \mathbf{R}_2 + \mathbf{R}_3 \quad (2-12)$$

如图 2-6 所示, \mathbf{R}_E 矢量与先前定义的位移矢量存在一点重要区别, 即 \mathbf{R}_E 并没有与机构中的任一根连杆固接或随之运动。在此情况下, 该矢量就可简明地表示机器人操作手的位置。如果是这样, 则该矢量的角度无显著意义, 我们将矢量 \mathbf{R}_E 简单写成如下坐标投影形式

$$x_E = r_2 \cos\theta_2 + r_3 \cos\theta_3 \quad (2-13)$$

$$y_E = r_2 \sin\theta_2 + r_3 \sin\theta_3 \quad (2-14)$$

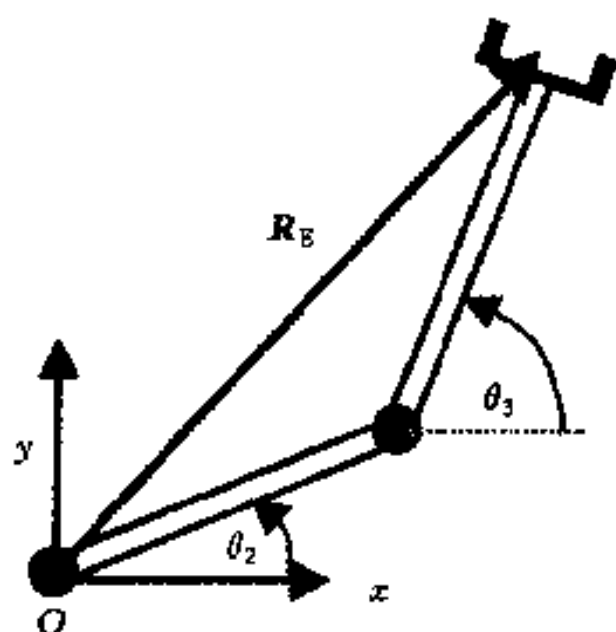


图 2-6 具有两个自由度和一个位移矢量的两连杆操作机构

如前所述, 可以容易地对上面这些标量方程求导数, 然后将其表示成矩阵形式, 最后得到以下雅可比转换矩阵

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = \begin{bmatrix} -r_2 \sin\theta_2 & -r_3 \sin\theta_3 \\ r_2 \cos\theta_2 & r_3 \cos\theta_3 \end{bmatrix} \begin{bmatrix} \omega_2 \\ \omega_3 \end{bmatrix} \quad (2-15)$$

最后应当注意一点, 机器人学文献关于转角的约定与前述有所不同, 这将导致雅可比转换矩阵的形式有所变化。在机器人学中, 使用的是不同于连杆转角的铰接点转角, 它的度量是相对于前一邻近连杆而言。这样做的直接原因就是机器人系统中使用的传感器检测的是相对转角。另一方面, 在运动学文献中通常使用的是绝对转角。

2.6.2 用矢量链描述任意点的运动

在图 2-7 中, 四连杆机构中的连杆 3 上标注了一点 C。有时, 使用矢量来描述这样一点的运动状况是非常重要的。这种方法的一个常用例子就是用来描述“连杆曲线”, 它是四连杆机构在运动过程中其连杆上特定点所形成的轨迹曲线。这一方法的另一普遍用途就是描述连杆转角与某一指定连杆质心运动之间的约束

关系。无论哪一种,原理是相同的。

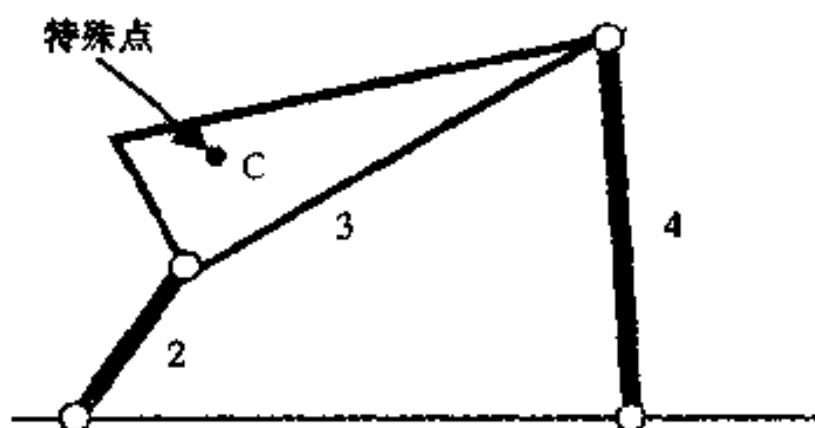


图 2-7 四连杆机构(连杆 3 上标有特殊点 C)

首先,第一步是确定连杆上所需考虑某一特殊点的位移矢量。这个矢量的起点应与该连杆的位移矢量的起点相一致。由图 2-7 可见,曲柄(连杆 2)与连杆(3)之间的顶角是确定的。因此,图 2-8 给出了特殊点 C 的位移矢量。

请注意图 2-8 中所示的两个夹角。如同先前所述,夹角 θ_3 是连杆 3 相对于水平或 x 轴的夹角,夹角 θ_p 是新的位移矢量相对于连杆位移矢量的夹角。该角度是一个常量,并不随连杆的运动而变化。另一方面,新的位移矢量相对于水平轴的绝对夹角是连杆夹角 θ_3 与新位移矢量夹角 θ_p 之和。

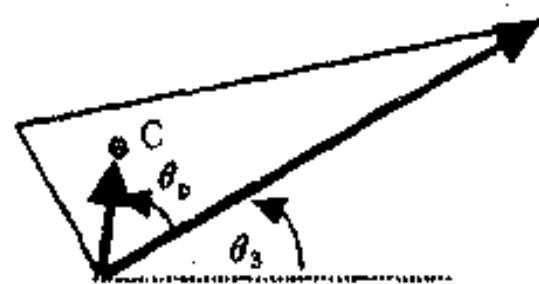


图 2-8 确定特殊点 C 位置的位移矢量

图 2-9 为一个标有上述位移矢量的四连杆机构。由图可见,描述某点位置的矢量方程为

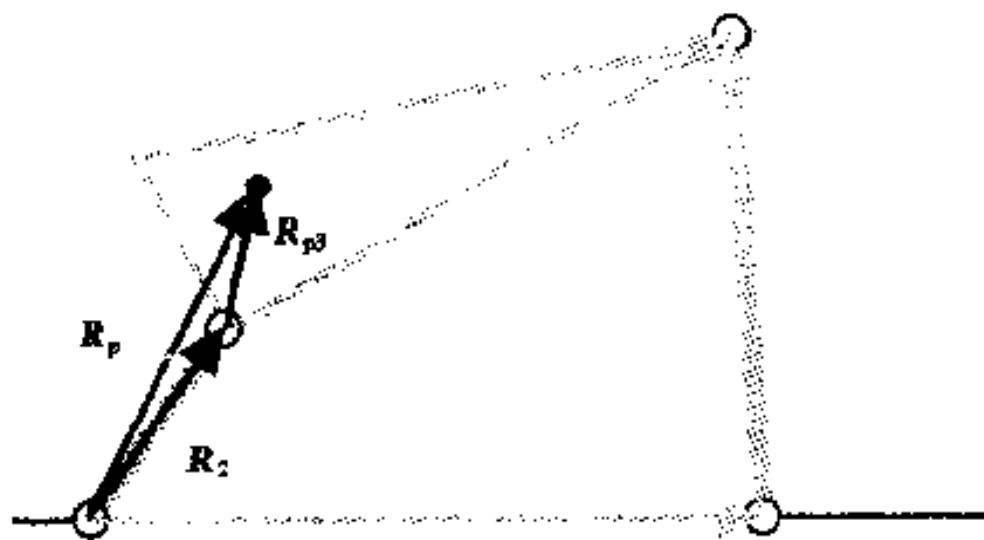


图 2-9 四连杆机构上任意一点的位移矢量

$$\mathbf{R}_p = \mathbf{R}_2 + \mathbf{R}_{p3} \quad (2-16)$$

考虑到在图 2-8 中夹角的定义,因此方程(2-16)可写成以下沿 x 和 y 方向的投影方程

$$x_p = r_2 \cos \theta_2 + r_{p3} \cos(\theta_3 + \theta_{p3}) \quad (2-17)$$

$$y_p = r_2 \sin \theta_2 + r_{p3} \sin(\theta_3 + \theta_{p3}) \quad (2-18)$$

如果该点的速度已知,则对上述方程求导即可得到如下速度表达式。请注意, θ_3 是随时间变化的,而 θ_{p3} 不随时间改变,不必对其求时间导数。

$$\dot{x}_p = -r_2 \omega_2 \sin \theta_2 - r_{p3} \omega_3 \sin(\theta_3 + \theta_{p3}) \quad (2-19)$$

$$\dot{y}_p = r_2 \omega_2 \cos \theta_2 + r_{p3} \omega_3 \cos(\theta_3 + \theta_{p3}) \quad (2-20)$$

类似地,加速度的表达式为

$$\ddot{x}_p = -r_2 \alpha_2 \sin \theta_2 - r_2 \omega_2^2 \cos \theta_2 - r_{p3} \alpha_3 \sin(\theta_3 + \theta_{p3}) - r_{p3} \omega_3^2 \cos(\theta_3 + \theta_{p3}) \quad (2-21)$$

$$\ddot{y}_p = r_2 \alpha_2 \cos \theta_2 - r_2 \omega_2^2 \sin \theta_2 + r_{p3} \alpha_3 \cos(\theta_3 + \theta_{p3}) - r_{p3} \omega_3^2 \sin(\theta_3 + \theta_{p3}) \quad (2-22)$$

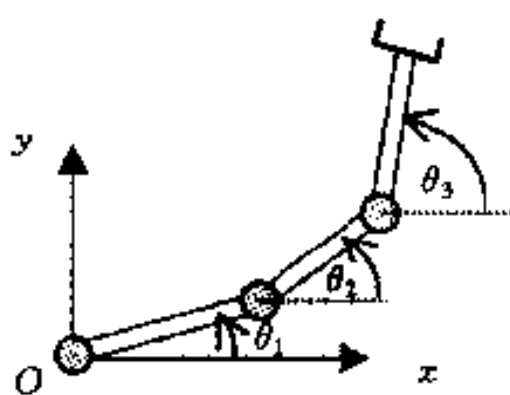
2.7 小结

本章介绍了用于机构分析的闭环矢量方程及其方程的求导。闭环矢量方程是机构运动过程中各连杆之间约束条件的数学表达式。位移矢量是机构各连杆的数学描述,其大小为连杆的长度,夹角与连杆角度相一致。本章所述方法与第4章将要研究的滑移接触类型的约束结点(如曲柄滑块机构)稍有不同。

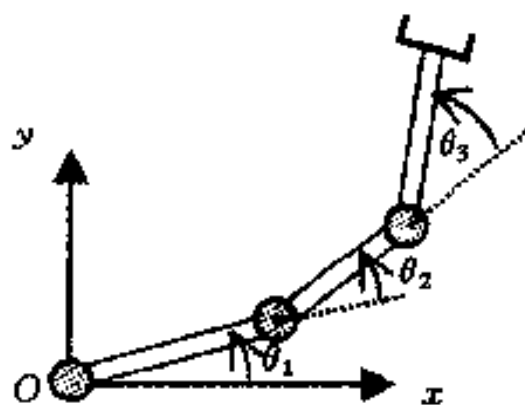
通过将闭环矢量方程沿 x 和 y 方向分解并投影,可易于进行方程的求导,进而得到相关连杆的速度和加速度。这些求导后得到的方程就形成了机构运动学和动力学仿真的基础。

第2章习题

1. 建立三连杆平面机器人的速度方程和加速度方程。



题1图

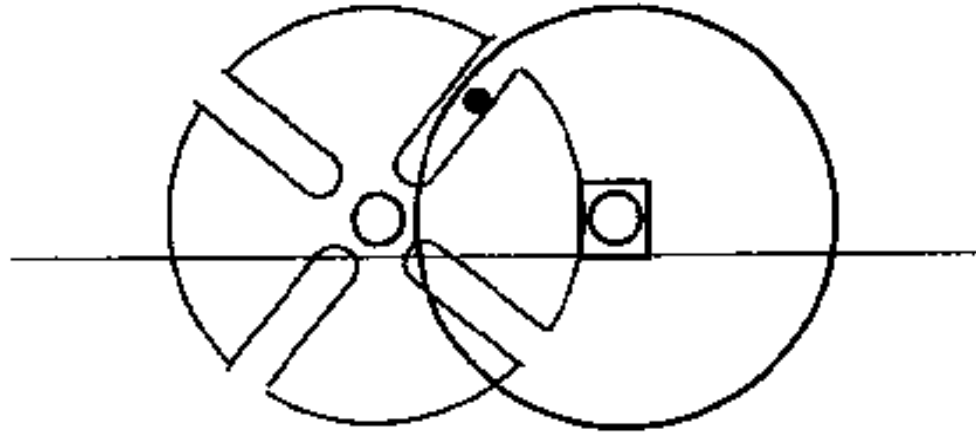


题2图

2. 使用图示相对角速度的概念,建立三连杆平面机器人的速度方程和加速度

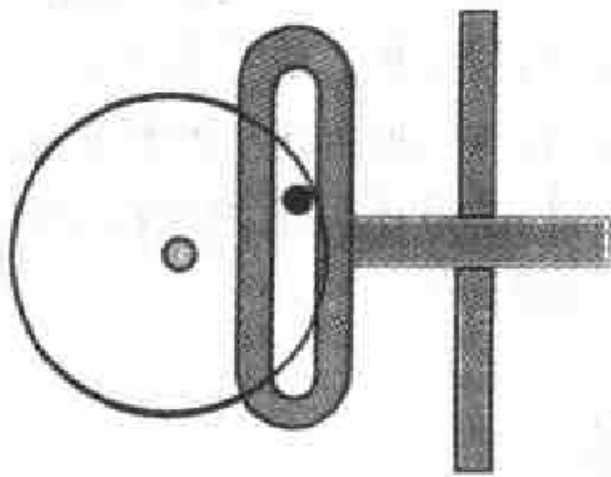
方程。

3. 考虑图示的间歇机构, 请写出该机构的闭环矢量方程并推导其速度和加速度方程。

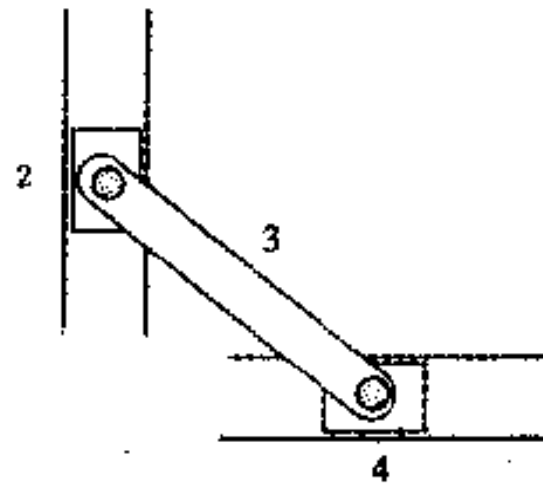


题 3 图

4. 试写出图示机构的闭环矢量方程并推导其速度和加速度方程。



题 4 图



题 5 图

5. 在图示双滑块机构中, 标出恰当的坐标系, 写出闭环矢量方程, 推导出铅直滑块 2 的运动与水平滑块速度之间的关系, 用连杆的角速度表示。

第3章 位置问题的求解概述

3.1 概述

在机构分析过程中,首先要进行位置分析。就单自由度机构而言,需要回答以下问题:若已知机构中某一根连杆的位置(相对于地面),那么在机构中其他连杆的位置应如何确定?如第2章所述,闭环矢量方程可用来求解这类问题。例如:若给定 θ_2 和所有的连杆长度,则 θ_3 和 θ_4 可完全由方程(2-2)和(2-3)求出。然而这组方程是关于 θ_3 和 θ_4 的非线性超越方程,非常难以求解。因此本章介绍了位置分析的数值解法,并讨论了该方法的MATLAB编程实现。应当指出,许多运动学课本都讨论了这种方法,并且MATLAB尤为适合本书所导出方程的数值解法,本章只是为了力求全书的完整性才将其编入。另外,本章所讨论的内容是相对独立的,因此亦可直接跳过本章而并不影响其他章节的阅读。

3.2 非线性代数方程的数值解法

本节将对求解非线性方程的牛顿-辛普森方法做扼要介绍,更详尽的阐述可以在其他文献中找到^①。简单地说,牛顿-辛普森方法是求解非线性方程的一种迭代法,它从某一给定的初始向量开始不断地给以增量直到所得结果“足够接近”精确解。迭代增量是通过非线性方程的级数展开式计算求得,“足够接近”是根据数值精度和工程实际的要求来确定的。

不失一般性考虑,假设一个包含两个未知数的两个方程联立求解问题,条件是方程可根据需要任意展开。此问题可以表述为如下形式:

$$f_1(q_1, q_2) = 0 \quad (3-1)$$

$$f_2(q_1, q_2) = 0 \quad (3-2)$$

其中: q_1, q_2 为待求未知量。第一步,将变量 q_i 表达为解的预估值(由变量 \bar{q}_i 表示)

^① Mable H H, C F Reinholtz. Mechanisms and Mechanics of Machinery. 4th ed, New York: John Wiley & Sons, 1987.

与一个描述预估值与方程解之差的微小修正因子(由变量 Δq_i 表示)之和,即

$$q_i = \bar{q}_i - \Delta q_i \quad (3-3)$$

众所周知,运用泰勒级数可将由式(3-3)所示变量组合而成的非线性函数 $f_1(q_1, q_2)$ 表示为一无穷级数的形式,级数的首项为该函数在预估值 (\bar{q}_1, \bar{q}_2) 处的函数值。因此,第一个函数的展开式如下所示:

$$f_1(q_1, q_2) = f_1(\bar{q}_1, \bar{q}_2) + \left. \frac{\partial f_1}{\partial q_1} \right|_{\bar{q}_1, \bar{q}_2} \Delta q_1 + \left. \frac{\partial f_1}{\partial q_2} \right|_{\bar{q}_1, \bar{q}_2} \Delta q_2 + [\text{高阶项}] \quad (3-4)$$

该级数将在迭代过程中用来逼近原函数。由于高阶项的存在,所以 Δq_i 项为非线性形式。为了使得其只含有线性形式,通常高阶项略而不计。值得注意的是,方程(3-4)中所包含的偏导数均取其在预估值 (\bar{q}_1, \bar{q}_2) 处的值,其目的在于简单易求,而且该值也就是表达式(3-4)中的系数。对第二个函数也按上式展开,即可得到两个函数展开式的矩阵表达形式如下:

$$\begin{bmatrix} f_1(\bar{q}_1, \bar{q}_2) \\ f_2(\bar{q}_1, \bar{q}_2) \end{bmatrix} + \begin{bmatrix} \left. \frac{\partial f_1}{\partial q_1} \right|_{\bar{q}_1, \bar{q}_2} & \left. \frac{\partial f_1}{\partial q_2} \right|_{\bar{q}_1, \bar{q}_2} \\ \left. \frac{\partial f_2}{\partial q_1} \right|_{\bar{q}_1, \bar{q}_2} & \left. \frac{\partial f_2}{\partial q_2} \right|_{\bar{q}_1, \bar{q}_2} \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3-5)$$

方程(3-5)给出了未知量估计值 (\bar{q}_i) 与方程精确解之间差值的计算方法。为求差值 (Δq_i) ,解方程(3-5)得

$$\begin{bmatrix} \Delta q_1 \\ \Delta q_2 \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial q_1} \right|_{\bar{q}_1, \bar{q}_2} & \left. \frac{\partial f_1}{\partial q_2} \right|_{\bar{q}_1, \bar{q}_2} \\ \left. \frac{\partial f_2}{\partial q_1} \right|_{\bar{q}_1, \bar{q}_2} & \left. \frac{\partial f_2}{\partial q_2} \right|_{\bar{q}_1, \bar{q}_2} \end{bmatrix}^{-1} \begin{bmatrix} -f_1(\bar{q}_1, \bar{q}_2) \\ -f_2(\bar{q}_1, \bar{q}_2) \end{bmatrix} \quad (3-6)$$

求解这组非线性方程的流程框图如图 3-1 所示。

3.3 四连杆机构的位置问题

在前一章中,推导得到了四连杆机构的闭环矢量方程以及其在 x 方向和 y 方向的投影方程,如方程(2-2)和(2-3)所示。通常,总可以选取基础连杆(连杆 1)作为参考系,利用这一有利条件,将方程(2-2)与(2-3)重新组合,可得到如下方程:

$$f_1(\theta_3, \theta_4) = r_2 \cos \theta_2 + r_3 \cos \theta_3 - r_1 - r_4 \cos \theta_4 = 0 \quad (3-7)$$

$$f_2(\theta_3, \theta_4) = r_2 \sin \theta_2 + r_3 \sin \theta_3 - r_4 \sin \theta_4 = 0 \quad (3-8)$$

因此位置问题可表述为:对于给定的一组连杆长度和它的值 θ_2 ,寻求适当的 θ_3 与 θ_4 ,使得函数 f_1 和 f_2 等于零。由于 f_1, f_2 是非线性和超越的(超越函数中包含未知量),故线性的矩阵分析方法已不适用,牛顿-辛普森方法正适合于求解这类非线性方程。

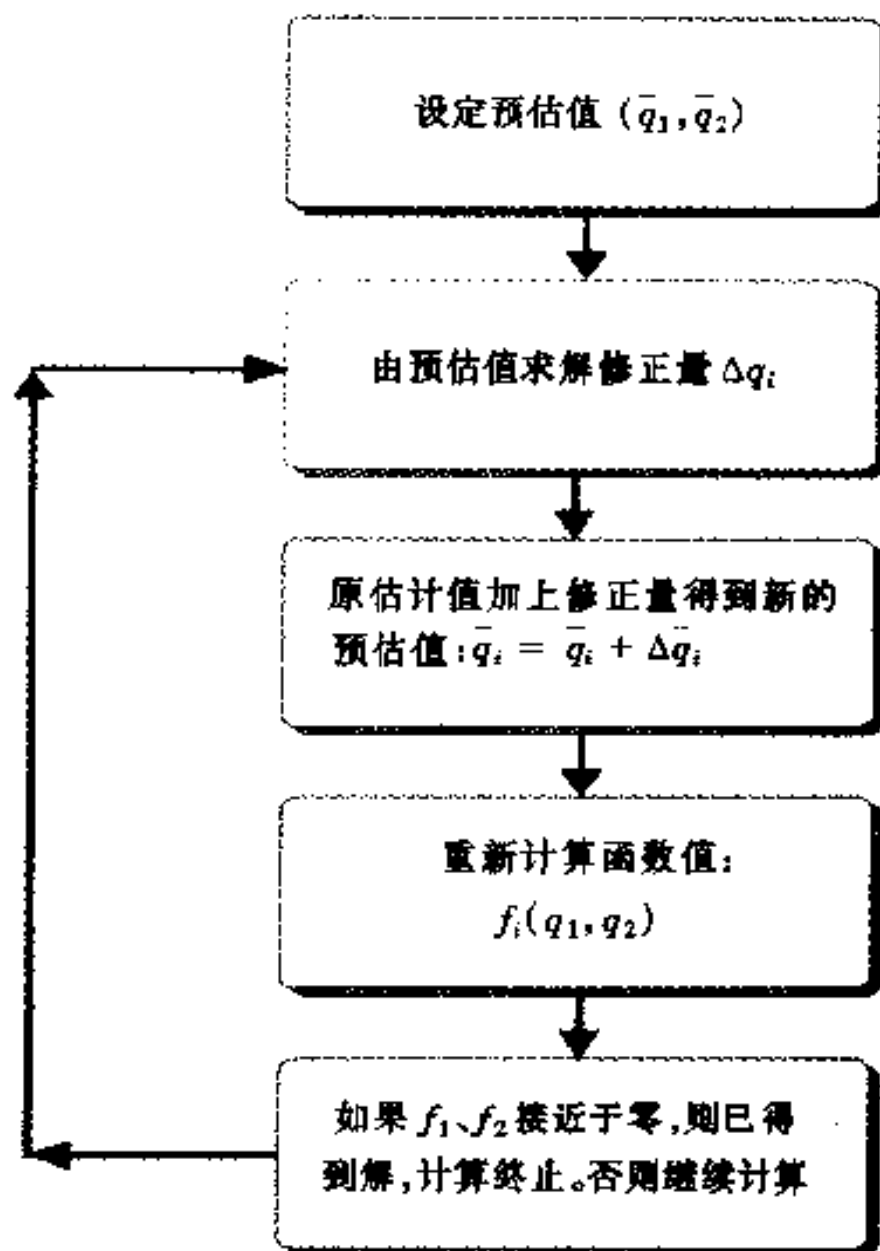


图 3-1 牛顿-辛普森算法求解运动学中位置问题的流程框图

首先,以名义解的形式重新定义变量,认为名义解接近精确解,其间差值由以下修正因子描述:

$$\theta_3 = \bar{\theta}_3 + \Delta\theta_3 \quad (3-9)$$

$$\theta_4 = \bar{\theta}_4 + \Delta\theta_4 \quad (3-10)$$

其中: θ_3 、 θ_4 代表问题的解; $\bar{\theta}_3$ 、 $\bar{\theta}_4$ 为接近解的名义解; $\Delta\theta_3$ 、 $\Delta\theta_4$ 即为修正因子。运用上述泰勒级数,将结果表达为方程(3-6)的形式,可得到如下矩阵方程:

$$\begin{bmatrix} \Delta\theta_3 \\ \Delta\theta_4 \end{bmatrix} = \begin{bmatrix} -r_3 \sin\bar{\theta}_3 & r_4 \sin\bar{\theta}_4 \\ r_3 \cos\bar{\theta}_3 & -r_4 \cos\bar{\theta}_4 \end{bmatrix}^{-1} \begin{bmatrix} -f_1(\bar{\theta}_3, \bar{\theta}_4) \\ -f_2(\bar{\theta}_3, \bar{\theta}_4) \end{bmatrix} \quad (3-11)$$

3.4 四连杆机构位置问题的 MATLAB 求解

MATLAB 运算平台非常适用于求解上述位置问题。以下函数为运用 MATLAB 求解含非线性超越方程的这类具有代表性的位置问题提供了可能。

MATLAB 提示

MATLAB 允许将可能重复执行的一组命令保存在独立的文本文件中。这类文件的扩展名为“.m”，因此被称为“m 文件”(试想程序设计员是否可以选择“.x”作为扩展名!)。一般有两种不同类型的 m 文件，脚本文件只是命令的简单集合，只需在命令提示符下键入文件名，这些命令便如同单独键入一样执行。第二类 m 文件是函数文件，其运行类似于 C 语言中的函数或 FORTRAN 语言中的子程序。这些函数接收参数返回数值，并在执行过程中无法访问除定义为全局变量以外的工作空间中的变量。编辑函数文件时，首行格式如下所示(其中包括关键词“function”(函数)、返回变量、函数)：

```
possol4.m

function [th3, th4]=possol4(th,rs)
%possol4 Position solution of a 4-bar mechanism
%
% Script used to implement Newton-Raphson
% method for solving nonlinear
% position problem of a 4-bar mechanism.
%
%Copyright 2001
%John F. Gardner
%
%
%
%inputs of function
%
%th(1) = theta-2
%th(2) = theta-3-bar (starting guess)
%th(3) = theta-4-bar (starting guess)
%
%rs(1) = r-1
%rs(2) = r-2
%rs(3) = r-3
%rs(4) = r-4
%
th2=th(1);
th3bar=th(2);
th4bar=th(3);
%
% set condition for convergence
%
epsilon=1.0E-6;
%Initialize the f-vector
```

```

%
% Compute the functions as a two-element vector:
%
f=[rs(3)*cos(th3bar)-rs(4)*cos(th4bar)+rs(2)*cos(th2)-
rs(1);rs(3)*sin(th3bar)-rs(4)*sin(th4bar)+rs(2)*sin(th2)];
%
% Repeatedly compute the correction factors
% as per equation (3-11)
%
while norm(f)>epsilon
J=[-rs(3)*sin(th3bar) rs(4)*sin(th4bar);
rs(3)*cos(th3bar) -rs(4)*cos(th4bar)];
dth=inv(J)*(-1.0*f);
th3bar=th3bar+dth(1);
th4bar=th4bar+dth(2);
f=[rs(3)*cos(th3bar)-rs(4)*cos(th4bar)+rs(2)*cos(th2)-
rs(1);rs(3)*sin(th3bar)-rs(4)*sin(th4bar)+rs(2)*sin(th2)];
norm(f)
end;
th3=th3bar;
th4=th4bar;

```

MATLAB 提示

MATLAB 以计算机编码和 20 世纪 70 年代的数值方法为基础,特别适于矩阵运算和求逆。因此,其基本数据结构是矩阵结构,矩阵操作简单易行。这一函数文件通过一行语法简单的语句来求解矩阵公式正表明了 MATLAB 这一特点。

作为例子首先考虑在例 2-1 中出现的四连杆机构。下面是 MATLAB 的一段命令对话,其中函数用来求解未知位置:

```

* rs(1)=12;
* rs(2)=4;
* rs(3)=10;
* rs(4)=7;
* th(1)=0;
* th(2)=45*pi/180;
* th(3)=135*pi/180;
* possol4(th,rs)

```

```

ans =
0.7688    1.6871

```

答案为: $\theta_3 = 44.05^\circ$, $\theta_4 = 96.66^\circ$ 。

MATLAB 提示

浮点数在 MATLAB 内部以多位有效数字表示,但在命令窗中 MATLAB 只显示 5 位有效数字。在某些操作中(例如设置仿真的初始条件)可能要求更高的数值表示精度。可以在命令窗中使用‘FORMAT’命令调整有效数字显示位数。若在命令提示符“>>”下键入‘FORMAT LONG’命令则可以显示完整的有效数字。

3.5 位置解与初始估计

在上一节中,介绍了 MATLAB 函数——只要给定连杆长度,输入角度以及初始估计,即可得到四连杆机构中其他两个连杆的角度。本章将讨论有关这种方法的一些细节问题。

首先应该认识到,许多机构的位置问题有多种解答。例如,若给定不同的连杆长度和输入角度,则机构就会有不同的位置形式,图 3-2 给出了两种不同的四连杆连动装置的组成方案。一般而言,牛顿-辛普森方法所求得的解有赖于所给定的初始估计。适当选择初始估计将保证该算法的正确运行。

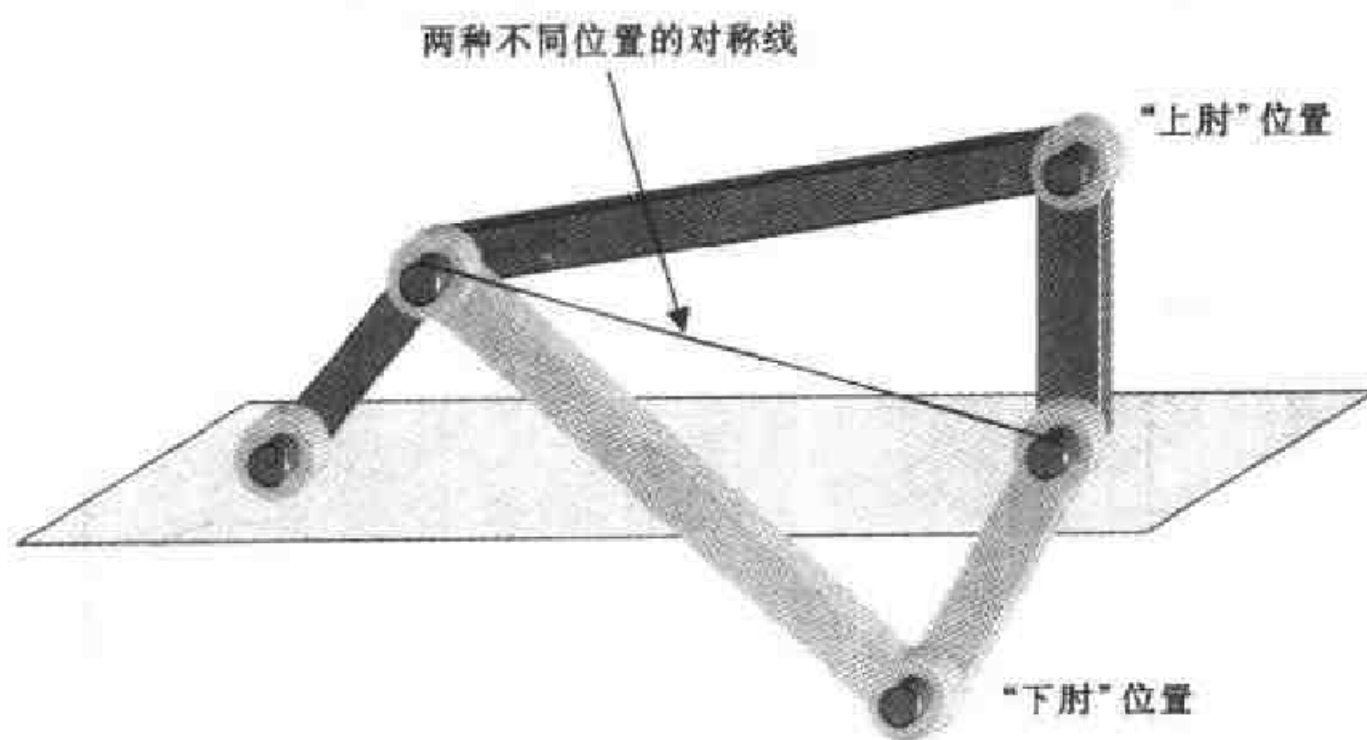


图 3-2 在给定连杆长度及曲柄角度的情况下四连杆机构的两种可能位置形式

确定初始估计的最佳途径就是按比例大致画出连杆草图。目测估计未知角度通常可作为较好的初始估计。此外,通常只需要将曲柄等间距地旋转一周,以此求出若干位置的位置解。MATLAB 的脚本文件易于依次求解这一过程。这时,只需给定第一个位置的初始估计。对于其它位置,其上一位置的位置解必然接近新的位置解,故可作为新位置的初始估计。下面给出实现这一过程的例子。

例 3-1

已知四连杆装置各连杆的长度同例 2-1, 输入连杆(连杆 2)以 5° 为间距, 等间距旋转一周。求各位置的位置解(θ_3, θ_4)。绘出 θ_3, θ_4 随 θ_2 变化的曲线。

如图 3-3 所示, 首先画出装置草图, 当输入角度为 0° 时, 估计出未知角度。从图中可以看出, θ_3 约为 45° , θ_4 约为 100° 。下面在 MATLAB 脚本文件中建立循环, 其中输入角 θ_2 以 0° 开始, 以 5° 为步长, 递增至 360° 。对应每一个 θ_2 的值调用函数 `possol4()` (见前面) 求解未知角。当脚本文件执行完毕, 由 MATLAB 的 'plot' 命令绘出曲线, 如图 3-4 所示。

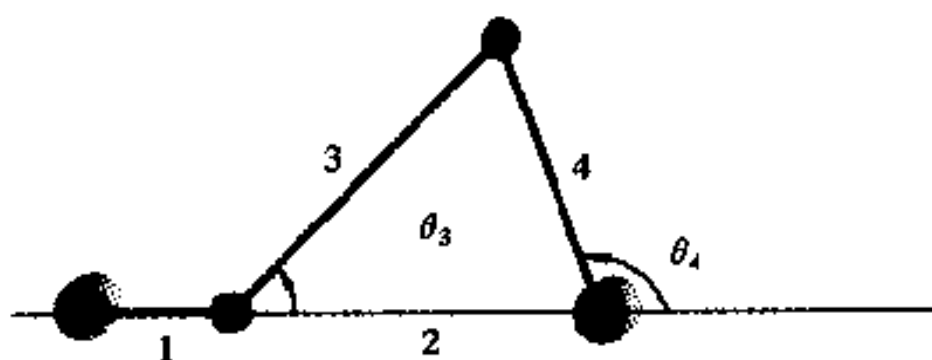


图 3-3 例 3-1 中四连杆机械装置初始结构位置草图

(注意到图中所示连杆 2 角度为 0° , 连杆 3 近似为 45° , 连杆 4 近似为 100°)

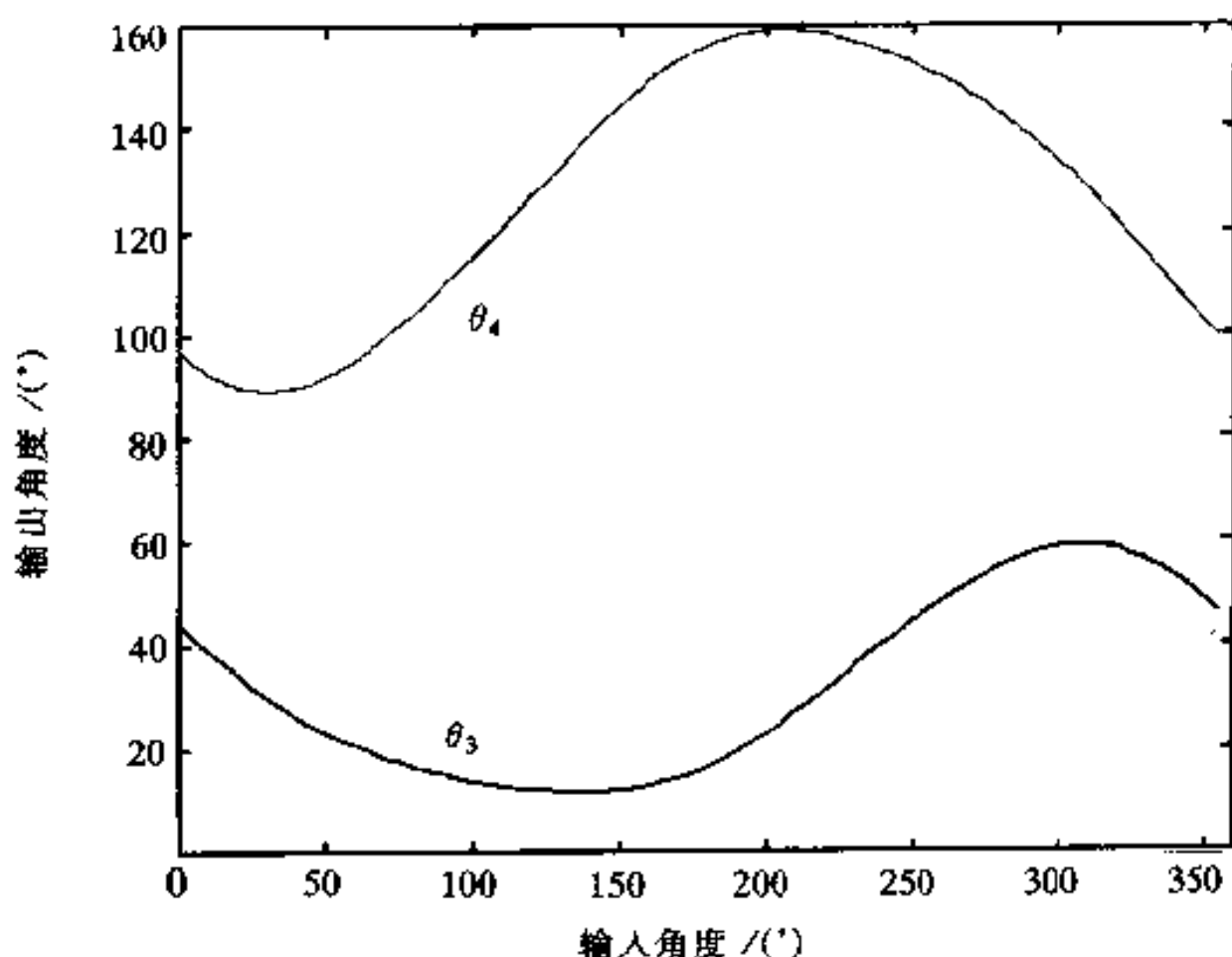


图 3-4 例 3-1 所考虑的情况下四连杆机构中 θ_3 和 θ_4 随输入角度的函数变化规律
(原文图中纵坐标为输入角度有误, 应为输出角度——译者注)

MATLAB 提示

下面将以例子说明 m 文件中脚本文件以及函数文件的应用。此处脚本文件调用了本章前面出现的函数文件。将某一函数编辑为函数文件,并在脚本文件中反复调用,这种联合运用非常行之有效。

```
fourbar.m
%fourbar Sequential position solution of a four-bar linkage
%
%Script file to solve the position solution for 5-degree
%increments
%of input link angle, for a full rotation of the mechanism.
%
%Copyright 2001
%John F. Gardner
%
%
% Handy conversion constant
%
D2R = pi/180.0;
%
% Set up link angles
%
rs(1) = 12.0;
rs(2) = 4.0;
rs(3) = 10.0;
rs(4) = 7.0;
%
% Set up the input and initial guesses
%
th(1) = 0.0;
th(2) = 45 * D2R;
th(3) = 100 * D2R;
%
% define the increment
%
dth = 5 * D2R;
%
for i = 1:72
    ths = possol4(th,rs);
    %Store results in a matrix, in degrees
    angles(i,:) = [th(1)/D2R ths(1)/D2R ths(2)/D2R];
    th(1) = th(1) + dth;
    th(2) = ths(1);
    th(3) = ths(2);
end
```

脚本文件将结果存储于名为 angles 的矩阵中(72 行 3 列),则可由 plot 命令绘图。

```

> plot(angles(:,1),angles(:,2),angles(:,1),angles(:,3))
> axis([0 360 0 160])
> xlabel('Input angles (degrees)')
> ylabel('Solved angles (degrees)')
> text(110,110,'Theta-4')
> text(50,35,'Theta-3')

```

MATLAB 提示

Plot 命令是 MATLAB 运行平台中非常有用的多功能命令之一。MATLAB 的在线帮助详细介绍了 plot 命令的运用,在此仅给出其最简单的格式。Plot 命令的使用要求有两个向量:第一个向量代表 x 轴方向的数据,第二个向量则包含了 y 轴方向的数据。我们往往将许多描述不同物理变量的数据存储在同一矩阵中。其每一列分别代表不同的变量。在这种情况下,可通过 ':' 操作符选择给定列执行 plot 命令。例如:命令 `plot(mydata(:,1),mydata(:,2))` 将绘制 mydata 矩阵前两列一一对应的图形。了解冒号操作符在某些情况下的含义有时是非常有益的,例如在这里可理解为:“所有行”。以这种方式可将 plot 命令表述为:绘制 mydata 矩阵第一列所有行对应于第二列所有行的图形。

3.6 小结

在本章中,由闭环矢量方程描述了机构的位置问题。通过引入 MATLAB 的函数文件以及脚本文件并运用牛顿-辛普森方法解决了四连杆机构位置求解的问题。将所介绍的程序进行简单修改即可适用于不同的其他机械结构。

第 3 章习题

1. 仔细分析图 3-4 所示 θ_3 与 θ_4 。是否可由图总结出连杆 3 和连杆 4 速率的某些规律? 并进行解释。

2. 例 3-1 中分别取不同的连杆长度 r_2 , 运用本文的函数文件进行分析。特别地,逐渐增加长度值直到函数文件无法返回数值解(数值算法不再收敛)。请解释产生这种现象的原因。

3. 修改 m 文件 possol4.m, 使其给出每次求解的迭代次数。用 MATLAB 的 plot 命令绘出迭代次数与相应的曲柄角度的关系曲线,从中可得到什么结论?

4. 改变习题 3 的收敛精度, epsilon 为 $1.0E-9$, 重做习题 3。

第4章 运动学仿真—— Simulink 的使用

4.1 什么是运动学仿真？

在工程分析中，计算机仿真这一术语已经逐渐被用来专指常微分方程的求解。系统动力学的研究领域包括建立和求解描述众多物理系统时变响应的微分方程，已有许多教科书对这一研究领域进行了详尽的论述^①。在本书中，仅考虑由运动分析而导出的一小类微分方程。

运动学仿真这一术语指使用仿真软件包反复求解机构运动约束方程，通过积分获得最终的速度（或加速度），从而确定机构运动的位置（和速度）。这样，一个仿真软件包，例如 Simulink，比通常求解机构运动的方法具有许多优点。最显著的优点是位置问题可以用软件包隐式求解，使用者只需为仿真提供适当的初始条件即可以求解机构在任意时刻的位置问题。

运动学仿真可以用来计算一个机构连杆的速度并画出相应的速度图，也可以如下面将显示的那样，被扩展用来计算加速度。对这两种情况，其出发点都是在第2章中所概述的闭环矢量方程。在本章中，将讨论机构的输入连杆以匀速和加速运动时的运动学仿真过程。

4.2 通过运动学仿真求解速度

4.2.1 曲柄滑块机构的闭环矢量方程

为了说明这种方法，对一个曲柄以匀速旋转若干圈的曲柄滑块机构进行运动学仿真分析。图4-1给出了曲柄滑块机构的示意图。通常，单缸四冲程发动机中

^① Shearer L J, Kulakowski B T, Gardner J F, Dynamic Modelling and Control of Engineering Systems. 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1997.

就有这种机构。

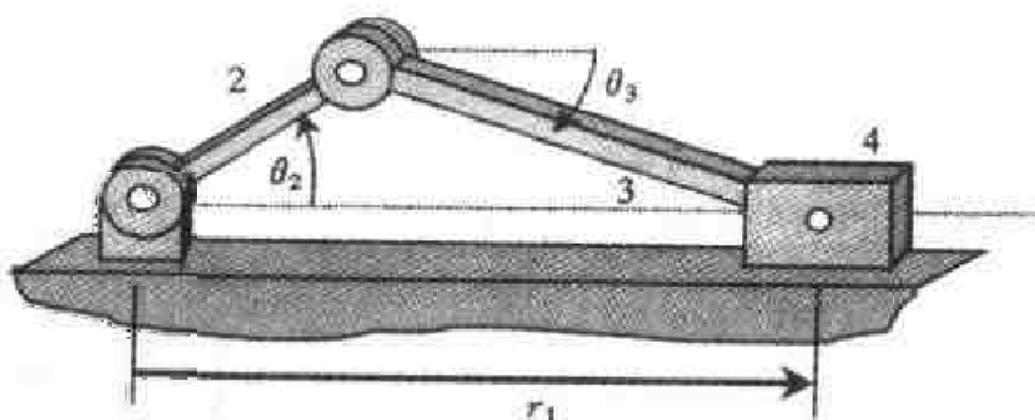


图 4-1 标有连接件编号和位置角度的曲柄滑块机构示意图

虽然此机构仅有一个自由度 (DOF)，但在仿真过程中需要考虑到所有三个可动连杆的运动。仿真以 ω_2 为输入来计算 ω_3 和 \dot{r}_1 (同样的方法已经在例 2-1 中论述过了)。

图 4-2 给出了描述曲柄滑块机构的矢量环。请注意这个矢量环和描述图 2-3 中的四连杆机构的矢量环有重要区别。在四连杆机构的矢量环中, 所有矢量的大小都保持常量, 只是方向随时间变化。而对于曲柄滑块机构, 矢量 \mathbf{R}_1 的大小随时间变化, 而方向保持不变。这一点在计算对时间的导数时是非常重要的。

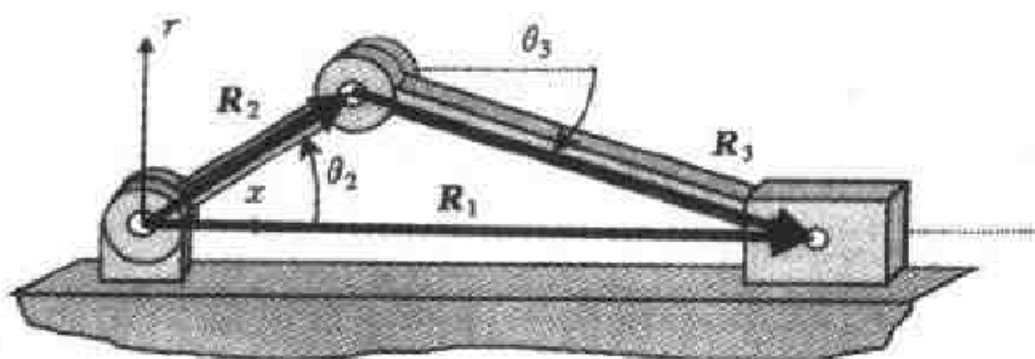


图 4-2 曲柄滑块机构的矢量环

曲柄滑块机构的闭环矢量方程为

$$\mathbf{R}_2 + \mathbf{R}_3 = \mathbf{R}_1 \quad (4-1)$$

将此矢量方程分解到 x 和 y 坐标轴上, 得到

$$r_2 \cos\theta_2 + r_3 \cos\theta_3 = r_1 \quad (4-2)$$

$$r_2 \sin\theta_2 + r_3 \sin\theta_3 = 0 \quad (4-3)$$

将上式对时间求导数, 有

$$-r_2 \omega_2 \sin\theta_2 - r_3 \omega_3 \sin\theta_3 = \dot{r}_1 \quad (4-4)$$

$$r_2 \omega_2 \cos\theta_2 + r_3 \omega_3 \cos\theta_3 = 0 \quad (4-5)$$

其中 \dot{r}_1 是矢量 \mathbf{R}_1 大小的变化率, 也是滑块相对于地面的平移速度。方程 (4-4) 和 (4-5) 可以写成如下的矩阵形式:

$$\begin{bmatrix} r_3 \sin\theta_3 & 1 \\ -r_3 \cos\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \omega_3 \\ \dot{r}_1 \end{bmatrix} = \begin{bmatrix} -r_2 \omega_2 \sin\theta_2 \\ r_2 \omega_2 \cos\theta_2 \end{bmatrix} \quad (4-6)$$

如果曲柄的速度 ω_2 已知, 方程 (4-6) 描述的是曲柄滑块机构的速度问题。

4.2.2 曲柄滑块机构运动学的 Simulink 仿真

当 ω_2 和连杆 2 的所有位置已知时, 方程 (4-6) 可以用来求解 ω_3 和 \dot{r}_1 。如果将 ω_2 视为仿真的输入, 可以用数值积分从速度中计算出 θ_2 、 θ_3 和 r_1 。在面向模块的 Simulink 图形方式中, 实现这一过程需要三个积分模块作为仿真的开始。图 4-3 给出了 Simulink 窗口和这三个积分模块。

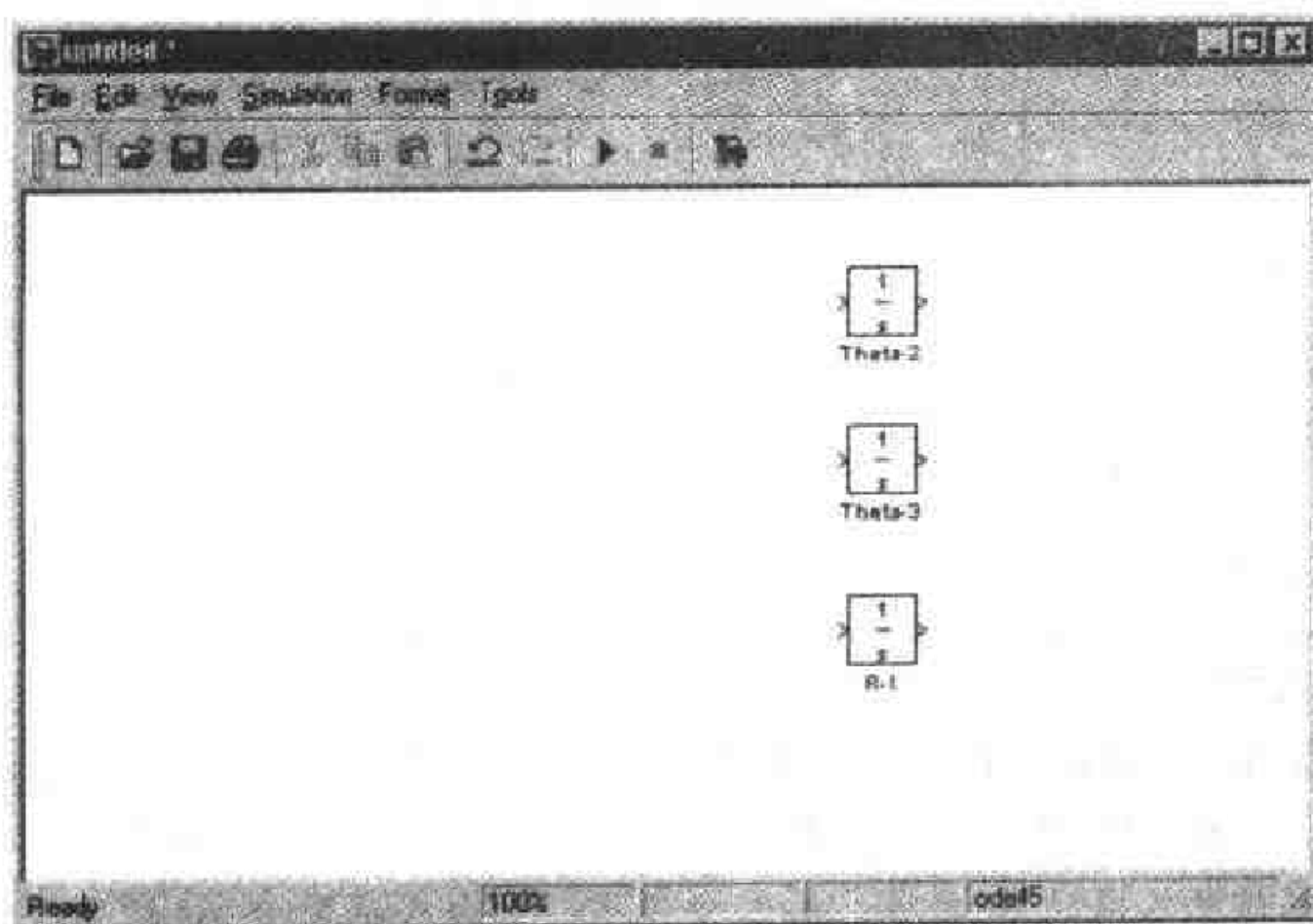


图 4-3 运动学仿真开始时的 Simulink 窗口, 积分模块以输出信号的名称 (即仿真结果) 来标识

要使仿真顺利地进行, 积分器的输入必须和相应的积分器恰当地连接起来。第一个输入的是 ω_2 , 这里假设 ω_2 为常量并作为仿真的输入, 可以使用 SOURCES 库中众多模块中的一个, 此问题选择 constant 模块。另外两个速度可以从闭环矢量方程 (4-6) 中求得。为此, 编写一个 MATLAB 函数来求解该方程, 函数名为 compvel, 存储于文件 compvel.m 中。文件的内容如下:

```

compvel.m
function [x]=compvel(u)
%
%Copyright 2001
%John F. Gardner
%
%
% function to compute the unknown velocities for
% a slider crank with constant crank input
%
% u(1) = omega-2
% u(2) = theta-2
% u(3) = theta-3
%
% Define the geometry
%
r2=1.0;
r3=4.0;
%
a=[r3*sin(u(3)) 1 ;-r3*cos(u(3)) 0];
b=[-r2*u(1)*sin(u(2));r2*u(1)*cos(u(2))];
%
x=inv(a)*b;

```

MATLAB 提示

MATLAB 允许在不同的文件中反复执行一组命令。这些文件都有扩展名“.m”，因此被称为“m 文件”（设想程序设计者如果将扩展名选择为“.x”将有多么麻烦！）。有两类不同的 m 文件。脚本文件是一组命令的集合，这组命令可以在命令提示符键入文件名来执行，这组命令的执行和它们被独立键入一样地执行。第二类 m 文件是函数文件，其执行过程与 C 语言中的函数和 FORTRAN 语言中的子函数执行过程一样。函数接受自变量并返回函数值，在函数执行过程中不能访问环境变量（除非变量被定义为全局变量）。在编写函数文件时，文件首行必须具有上面所示的格式：有 function 关键词、返回变量和函数自变量。

注意到曲柄滑块机构连接杆件的长度 r_2 和 r_3 在函数中给出了定义。可以用 MATLAB 中的 function 块将此函数嵌入到 Simulink 仿真中，function 块可以在“Functions and Tables”库中找到。function 块可以把矢量作为输入和输出，但矢量必须由原始信号“组装”而成。图 4-4 显示了输入矢量（函数中的 u ）用 Mux 块（多路转换器）组装得到。请注意信号连接的顺序和其在函数中使用的顺序完全一样。同样地，用 DeMux 块来分解输出矢量（函数中的 x ）。

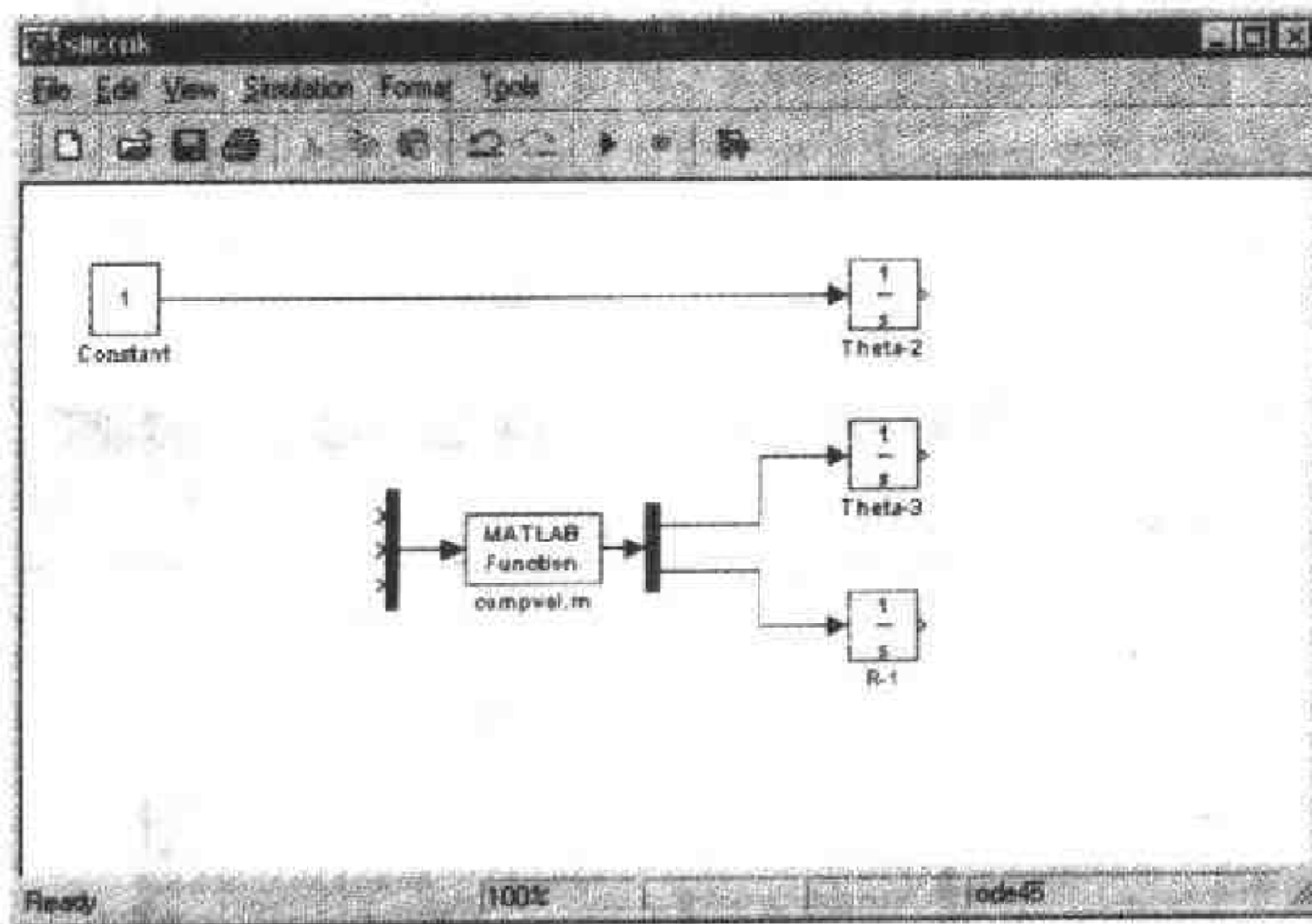


图 4-4 扩展模块图以包含函数 `compvel.m`, 将积分器的输入和函数的输出联系起来

下面, 适当的信号被“联网”到 Mux 块, 结果示于图 4-5 中。

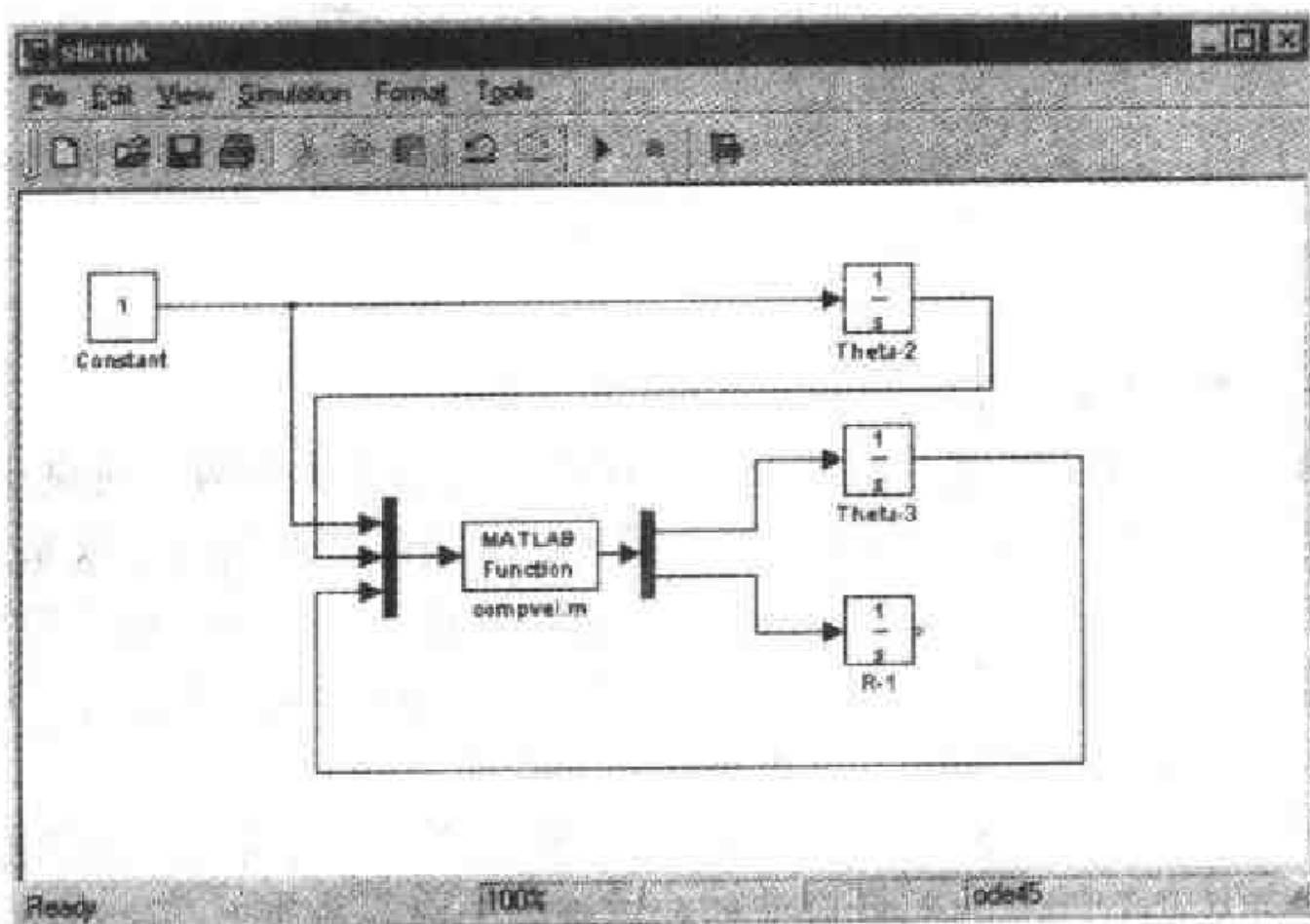


图 4-5 将适当的信号和函数的输入联系起来

为完成仿真, 加入一个 Mux 块, 以此将输出结果集合成一个矩阵。该矩阵在

仿真完成时在 MATLAB 平台中是可见的。值得一提的是,也可以对 Simulink 中的信号线进行标注。双击该线,会打开一个文本框允许你输入信号的名称。这对于解释和文件说明仿真过程非常有用。完成后的仿真过程示于图 4-6 中。图 4-6 中的 clock 块对版本 5. x 或高版本的 MATLAB 是不需要的,在此给出是为了照顾不同版本的完整性。

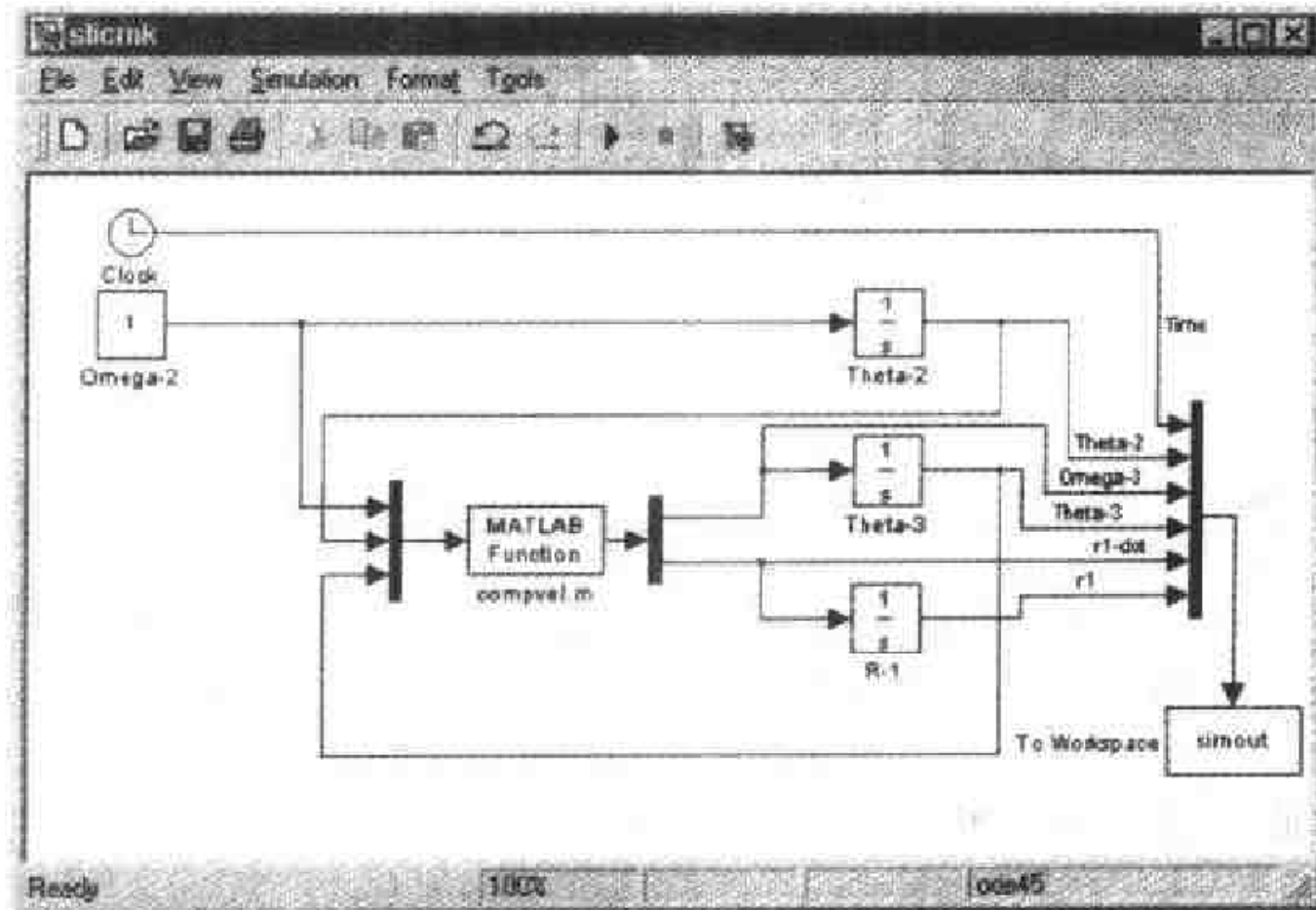


图 4-6 匀速输入时完成的运动学仿真。(角度和速度存于 MATLAB 环境变量 `yout` 中(文件:slicrnk.mdl))

4.2.3 建立初始条件

前面几节说明了如何基于闭环矢量方程建立曲柄滑块机构面向块-图的仿真过程。然而在仿真运行之前,必须为积分器建立适当的初始条件。这是求解任何微分方程的关键一步。而且,这对于运动学仿真显得特别重要。如果使用了不相容的初始条件,将会导致仿真失败。对于图 4-6 中所给出的仿真, θ_2 、 θ_3 和 r_1 的初始条件必须是机构在某个真实位置时的角度和长度。

在这种情况下,初始条件可以通过简单的几何关系求解给出。为了便于求解,假设曲柄的初始位置为 $\theta_2 = 0$, 此时曲柄和连杆处于同一条线上。这个位置在活塞-气缸装置中被称为“上死点(TDC)”。通过进一步观察可知, θ_3 也等于零并且滑块处于距离坐标原点长度为 $r_2 + r_3$ 的位置。表 4-1 给出了一组相容的初始条件。

表 4-1 仿真的初始条件

位置	值
θ_2/rad	0
θ_3/rad	0
$r_1/\text{mm(in)}$	177(5.0)

用下面两种方法中的任何一种可建立起仿真的初始条件。最直接的方法就是双击仿真流程图中的积分器图标,在“Initial condition”栏中输入数值。遗憾的是,这种方法对于更复杂的初始条件会变得非常笨拙,随后几节将说明这一点。另一种方法是为 MATLAB 环境变量设置初始值。下面说明这种方法的两个步骤。

在 MATLAB 命令提示符中键入

```
>> th20 = 0;
>> th30 = 0;
>> r10 = 5.0;
```

双击积分器图标会弹出一个如图 4-7 所示的对话框。注意到变量名在对话框中的“Initial condition”栏中输入。此时对话框中的其他部分暂时不必考虑。

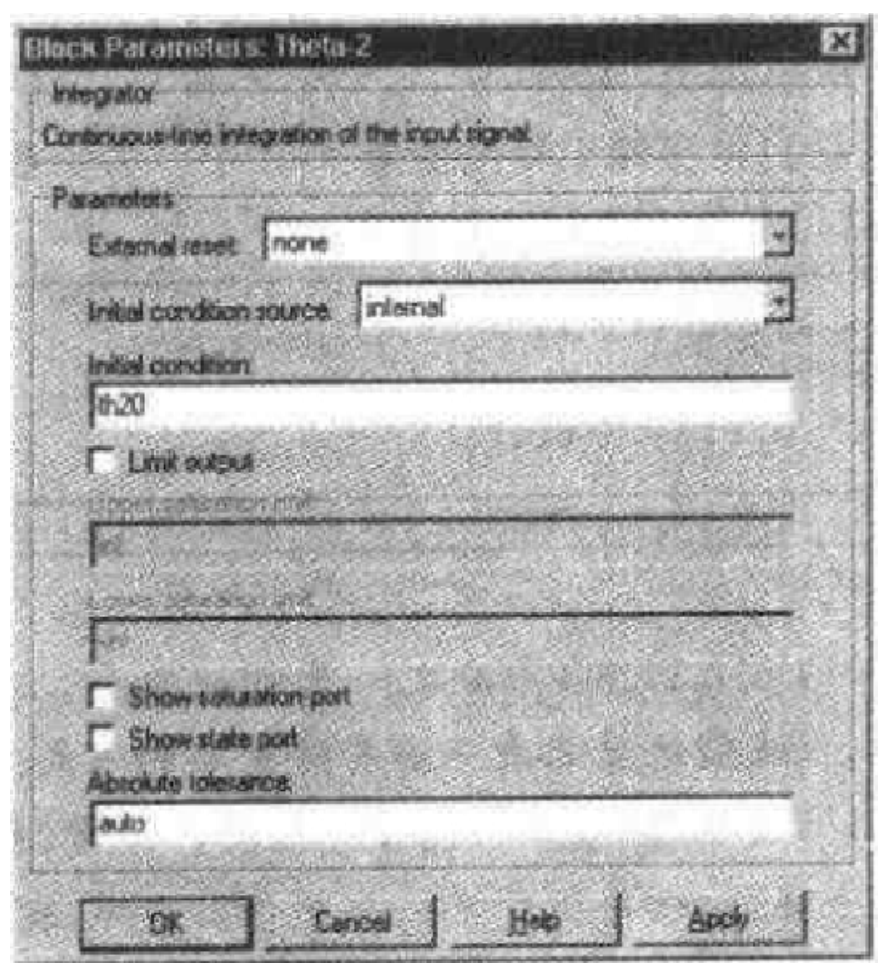


图 4-7 为 Simulink 中的积分器设置初始条件的对话框

用这种非直接方式来建立初始条件的原因有两个方面。首先,这种方法可以用 MATLAB 环境中一系列命令来非常容易地改变初始条件,而无需在每次初始

条件改变时双击每个积分器图标。其次,这种方法可以在 MATLAB 允许的最大精度范围给出 Simulink 的初始条件。对于这里给出的仿真情况,这一点因为初始条件可以精确给出而显得并不重要。然而,如随后将会看到的那样,多数的仿真需要使用复杂的几何关系,初始条件的精度对仿真的正确运行有着重要的影响。

4.2.4 仿真结果

现在用仿真来分析机构的运动。这里假设曲柄以 1 800 r/min(188.5 rad/s)的速度旋转(这是除草机单缸发动机的正常工作速度)。按这种速度,曲柄仅在 0.07 s 之内就旋转了两圈,所以仿真的时间从 0~0.07 s。双击仿真中的 constant 块可以设置曲柄的旋转速度。出现的对话框允许为曲柄旋转速度输入任意值。角速度的单位是 rad/s,注意到这一点是十分重要的。为便于进行分析,以转/分(r/min)为单位的曲柄旋转速度必须转化为 rad/s。

在 5.x 版本的 MATLAB 中,默认的仿真参数(算法、步长、误差限)对本例是适用的。在早些版本的 MATLAB 中,你需要设置最小的步长和大约为 1E-5 量级的误差。注意到仿真没有画图也没有包含 scope 块,所以在仿真运行过程中在屏幕中不会出现任何东西。然而当显示仿真结束时, MATLAB 平台包含了一个新的矩阵 simout。可以用 MATLAB 绘图命令来显示系统的运动。矩阵 simout 有 6 列(Mux 块中的每个变量占一列,行数与仿真中的时间步数相等)。回到图 4-6,我们看到这 6 列依次为:时间, θ_2 , ω_3 , θ_3 , \dot{r} 和 r_1 。注意到在 5.x 版本的 MATLAB 中,时间变量自动存储于环境变量 tout 中。为了显示滑块的位移 r_1 随时间的变化图形,可以用下列 MATLAB 命令:

```
>> plot(simout(:,1),simout(:,6))
>> xlabel('Time (sec)')
>> ylabel('Piston Displacement (in)')
```

MATLAB 提示

最有用、功能最强的 MATLAB 命令之一是 plot 命令。在线帮助功能对 plot 命令进行了详尽叙述,但我们这里只看一看其最简单的形式。plot 命令需要两个变量,第一个代表数据的 x 坐标,第二个包含 y 坐标。通常,描述许多物理变量的数据可以被存储在一个矩阵中,每列对应一个变量。这样,我们可以利用“:”符号来为 plot 命令选定一列。例如,命令 plot(mydata(:,1),mydata(:,2)) 会画出矩阵 mydata 前两列的相互变化规律。有时将诸如此类情况下的冒号操作符理解为“所有行”是很有用的。因此,上述 plot 命令可以重新叙述为:绘出矩阵 mydata 第一列的所有行相对于 mydata 第二列所有行的变化规律。

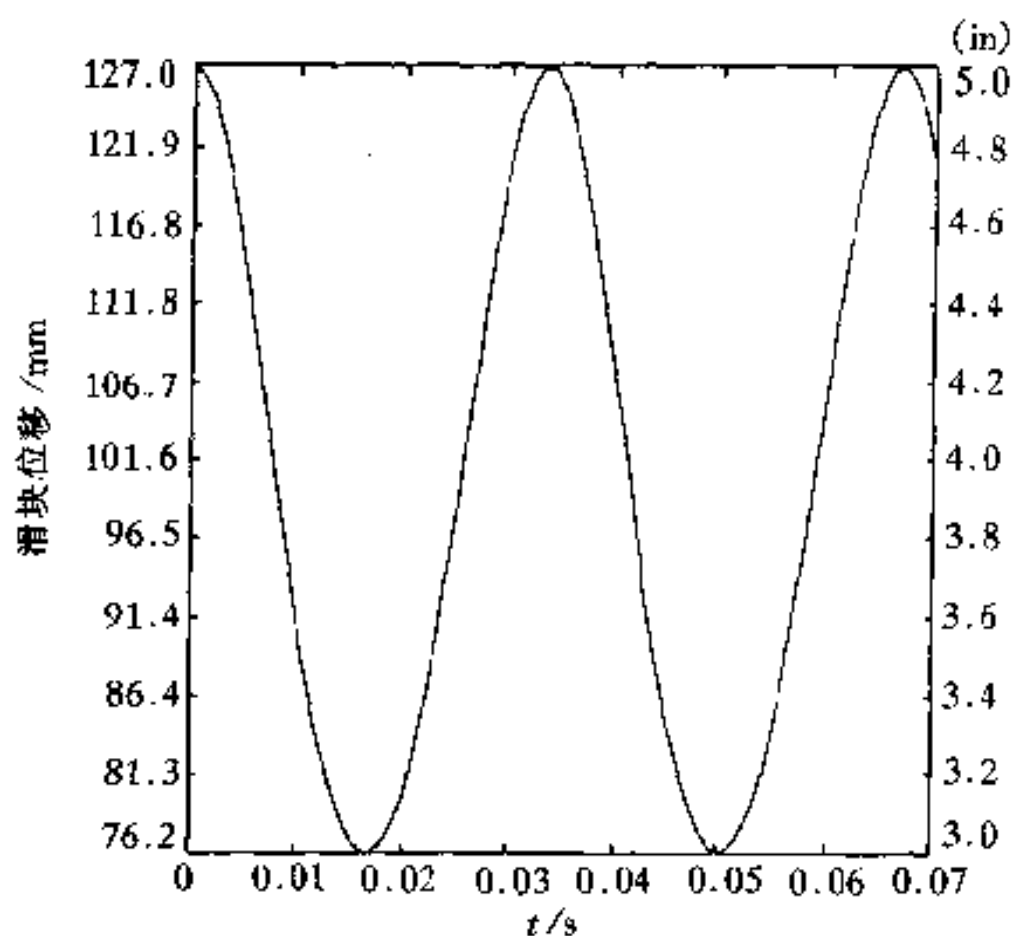


图4-8 曲柄滑块机构滑块位移随时间变化的仿真曲线图*

类似地, 连杆的速度变化规律可以用下面的命令来画出:

```
>> plot(simout(:,1),simout(:,3))
>> xlabel('Time (sec)')
>> ylabel('Connecting Rod Speed (rad/s)')
```

连杆速度仿真曲线图如图4-9所示。

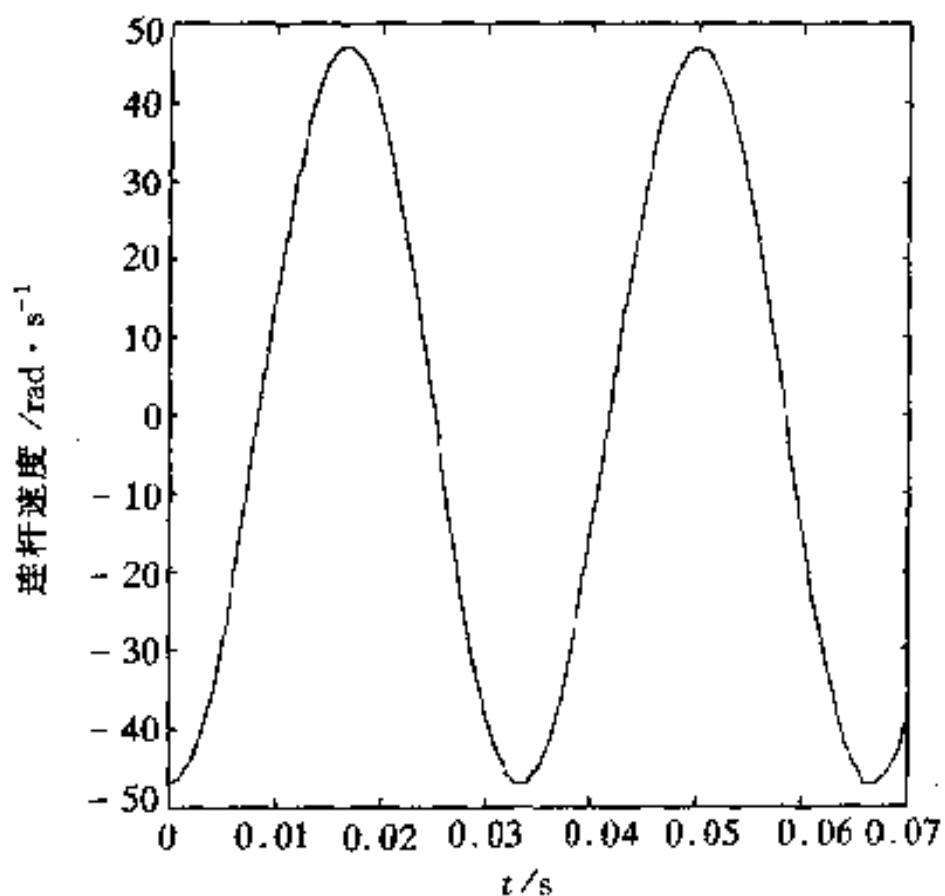


图4-9 曲柄滑块机构的连杆角速度随时间变化的仿真曲线图

* 为保持原版书的完整性,方便读者阅读,将英制单位移至右侧作为对照——出版者注。

最后,看一下滑块的速度也许是十分有趣的。这一点对空气压缩机或活塞式发动机的设计者是非常有用的,因为这些设计人员对活塞和缸内壁之间的摩擦特性非常关心。

```
>> plot(simout(:,1),simout(:,5))
>> xlabel('Time (sec)')
>> ylabel('Piston Speed (in/s)')
```

滑块的速度仿真曲线图如图 4-10 所示。

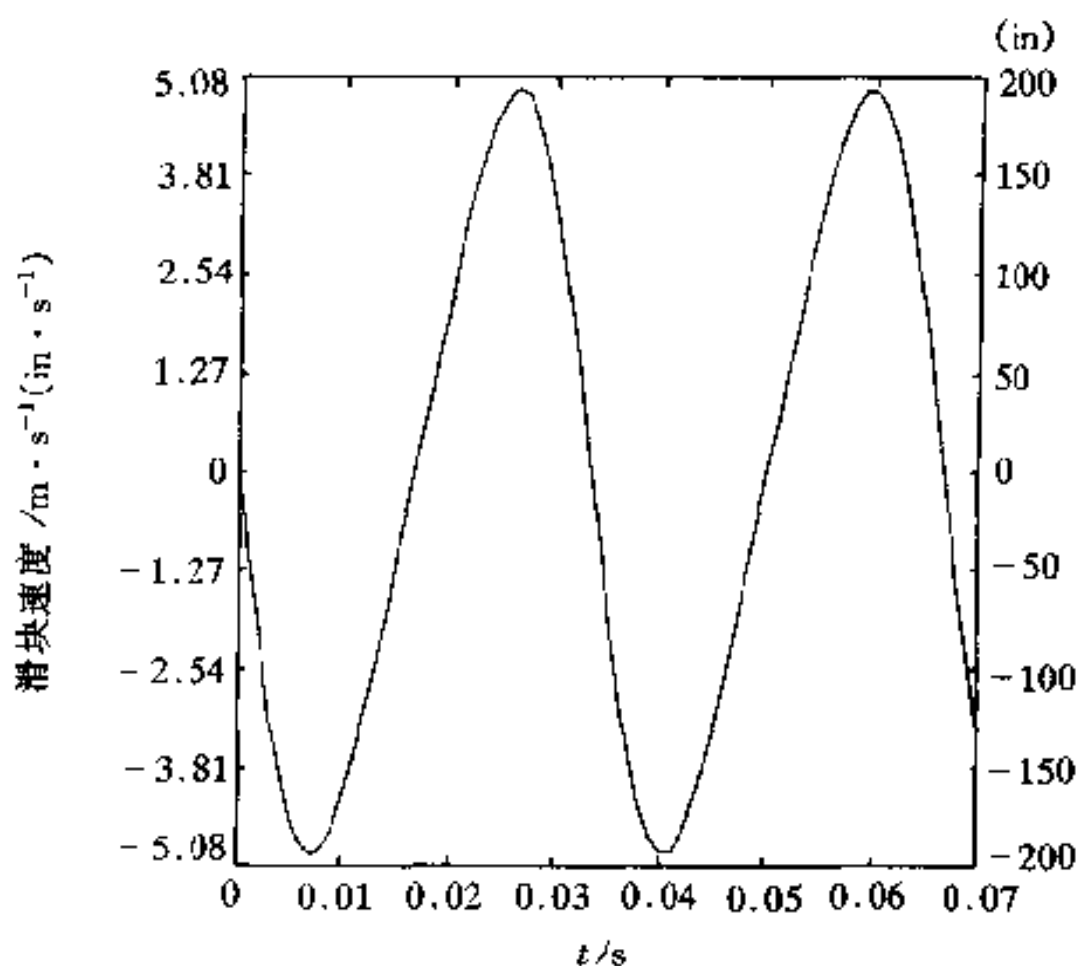


图 4-10 曲柄滑块机构的滑块速度随时间变化的仿真曲线图

4.3 通过运动学仿真求解加速度

4.3.1 考虑加速度的仿真

仍考察曲柄滑块机构的例子,注意到速度方程是通过闭环矢量方程(4-4)和(4-5)求关于时间的一阶导数而得到的。如果对时间求二阶导数,必须要特别注意对时间变量乘积的求导过程。曲柄滑块机构的闭环矢量方程对时间的二阶导数方程由下式给出:

$$-r_2 \dot{\omega}_2 \sin \theta_2 - r_2 \omega_2^2 \cos \theta_2 - r_3 \dot{\omega}_3 \sin \theta_3 - r_3 \omega_3^2 \cos \theta_3 = \ddot{r}_1 \quad (4-7)$$

$$r_2 \dot{\omega}_2 \cos \theta_2 - r_2 \omega_2^2 \sin \theta_2 + r_3 \dot{\omega}_3 \cos \theta_3 - r_3 \omega_3^2 \sin \theta_3 = 0 \quad (4-7)$$

在仿真时,位移 (θ_2 、 θ_3 和 r_1) 被视为已知量(因为它们积分的结果)。类似地,如果将仿真扩展并计入加速度,则速度 (ω_2 、 ω_3 和 \dot{r}_1) 也被视为已知量。在仿真模块集合过程中,输入连杆的角加速度被视为仿真系统的输入量。最后,为了方便起见,以 α_i 表示角加速度。从而,加速度方程可以写成如下紧凑的矩阵形式:

$$\begin{bmatrix} r_3 \sin\theta_3 & 1 \\ -r_3 \cos\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \ddot{r}_1 \end{bmatrix} = \begin{bmatrix} -(r_2 \alpha_2 \sin\theta_2 + r_2 \omega_2^2 \cos\theta_2 + r_3 \omega_3^2 \cos\theta_3) \\ r_2 \alpha_2 \cos\theta_2 - r_2 \omega_2^2 \sin\theta_2 - r_3 \omega_3^2 \sin\theta_3 \end{bmatrix} \quad (4-9)$$

比较方程 (4-9) 和矩阵形式的速度方程 (4-6),这两个方程左端 2×2 的矩阵完全一样。由此可以看出闭环矢量方程导数的这一重要特性。可以将此作为推导新的机构运动方程时验证方程正确性的重要检查方式。

加速度仿真的大部分模块结构和前面集合的速度仿真结构一样。从最后形成的速度仿真结构(图 4-6)开始,去掉联系速度的信号线及输入模块,并将模块作适当的重新布置以便为积分加速度获得速度所需另外的积分器留出位置。图 4-11 给出了修改的第一步。

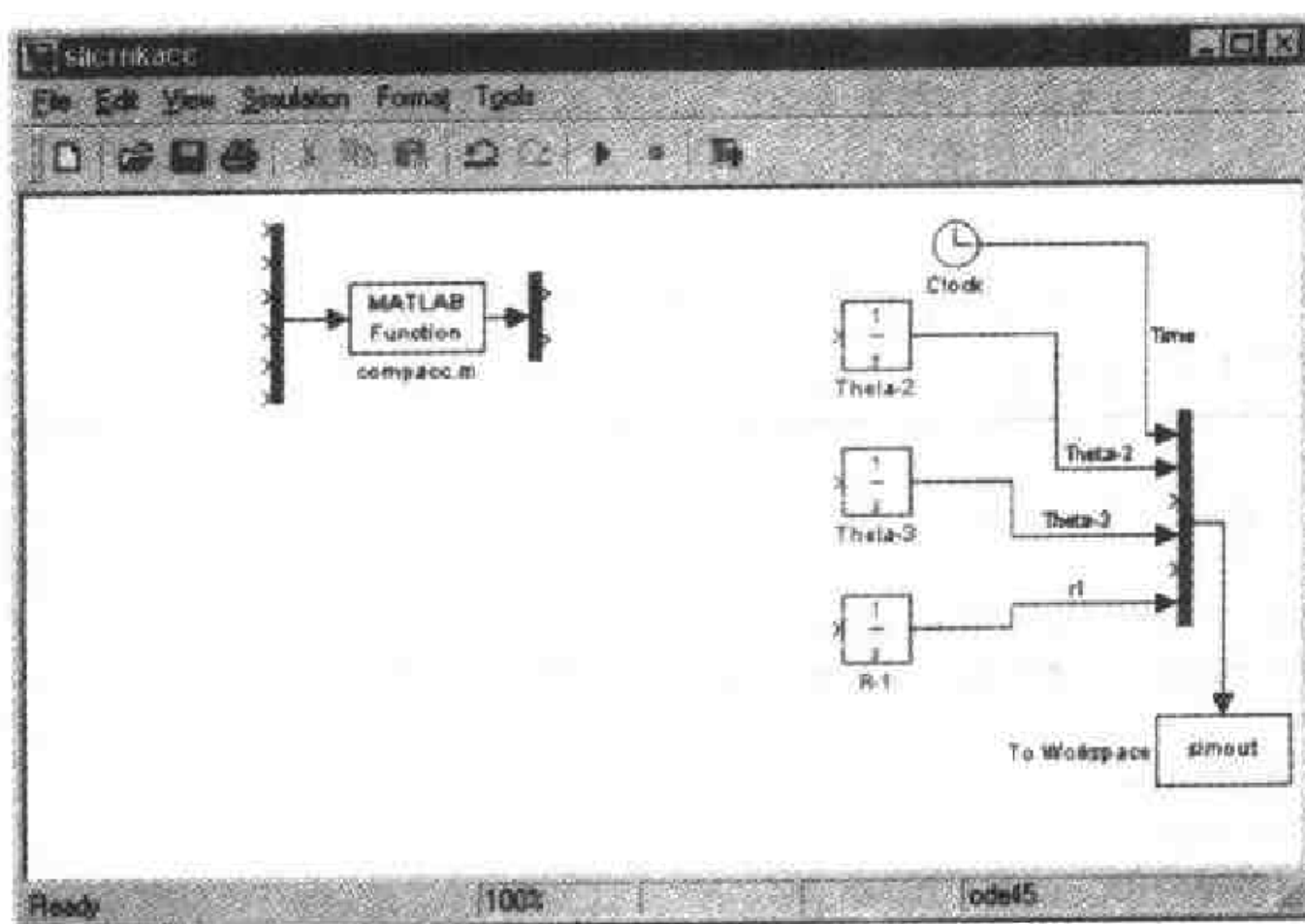


图 4-11 显示曲柄滑块机构从速度扩展到加速度仿真第一步的 Simulink 简图。

求解上面的加速度方程 (4-9) 需要一个新的函数文件 compacc.m。该新函数需要额外的输入 α_2 (仿真输入)、 ω_2 和 ω_3 。Mux 模块被扩展,仿真系统增加了三个新的积分模块。图 4-12 图示了修改后的仿真系统。

输出矢量也相应地要加以改变。变量 simout 现在包括 6 列数据: t 、 θ_2 、 ω_2 、 θ_3 、 \dot{r}_1 和 r_1 。

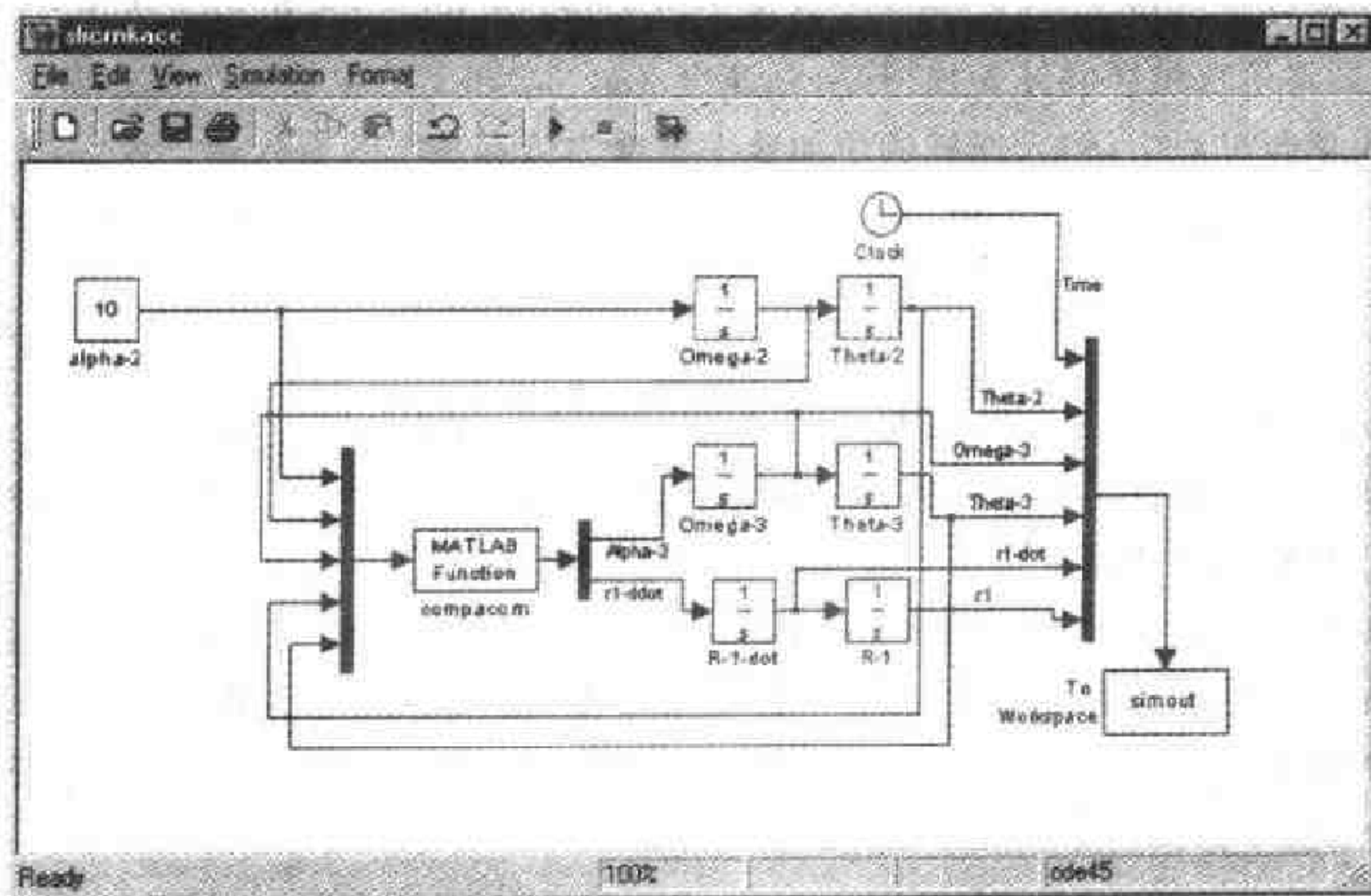


图 4-12 曲柄具有可变转速和加速度的曲柄滑块机构 Simulink 仿真
(文件: slicrnkacc.mdl)

新函数 `compacc.m` 具有和第 4 章中的函数 `compvel.m` 非常类似的形式, 其 MATLAB 程序脚本如下:

```

compacc.m
function [x]=compacc(u)
%
% function to compute the unknown accelerations for
% a slider crank with variable speed crank input
%
% u(1) = alpha-2
% u(2) = omega-2
% u(3) = omega-3
% u(4) = theta-2
% u(5) = theta-3
%
% Define the geometry
%
r2=1.0;
r3=4.0; (cont....)
%
a=[r3*sin(u(5)) 1 ; -r3*cos(u(5)) 0];

```

```

b=[-(r2*u(1)*sin(u(4))+r2*u(2)^2*cos(u(4))+r3*u(3)^2*cos(u(5)));
  r2*u(1)*cos(u(4))-r2*u(2)^2*sin(u(4))-r3*u(3)^2*sin(u(5))];
%
x=inv(a)*b;

```

4.3.2 运行曲柄滑块机构的仿真

前面已经强调指出,必须确保所建立的初始条件的相容性。对于基于速度的仿真,相容性意味着初始条件代表连杆的真实有效位置。对加速度仿真,除了位移的相容性要求外,初始速度也必须表示一个有效的速度组合。将曲柄滑块机构视为一个单自由度系统可以更好地理解这一点。如果指定了曲柄转速(ω_2 的初始速度),那么另外两个速度(ω_3 和 \dot{r}_1)也就通过速度方程(4-6)确定了下来。

这里给出两个例子。第一个例子仿真在匀加速下从零速度状态开始执行,第二个例子在匀速条件下运行。

例 4-1

此仿真算例的初始条件与基于速度的仿真类似,区别只是此仿真算例的初始速度条件均设置为零。输入(α_2)通过一个 Simulink 常量模块设定。表 4-2 给出了仿真的初始条件。

表 4-2 匀加速仿真的初始条件

积分器	初始条件
θ_2/rad	0
θ_3/rad	0
$r_1/\text{mm}(\text{in})$	177(5.0)
$\omega_2/\text{rad} \cdot \text{s}^{-1}$	0
$\omega_3/\text{rad} \cdot \text{s}^{-1}$	0
$\dot{r}_1/\text{mm} \cdot \text{s}^{-1}(\text{in} \cdot \text{s}^{-1})$	0(0,原文误为 in. s)

输入加速度设定为 10 rad/s^2 ,仿真运行时间为 4 s。当 4 s 结束时,曲柄转速应为 40 rad/s (大约 380 r/min),曲柄共旋转 80 rad (大约 13 圈)。图 4-13 给出了滑块所经历的加速过程仿真曲线。注意到图 4-13 给出的周期数和确定的曲柄旋转圈数 13 相吻合。同时注意到加速度本质上肯定不是按正弦规律变化,这很明显地表明曲柄滑块机构中滑块的运动不是简单的简谐运动。

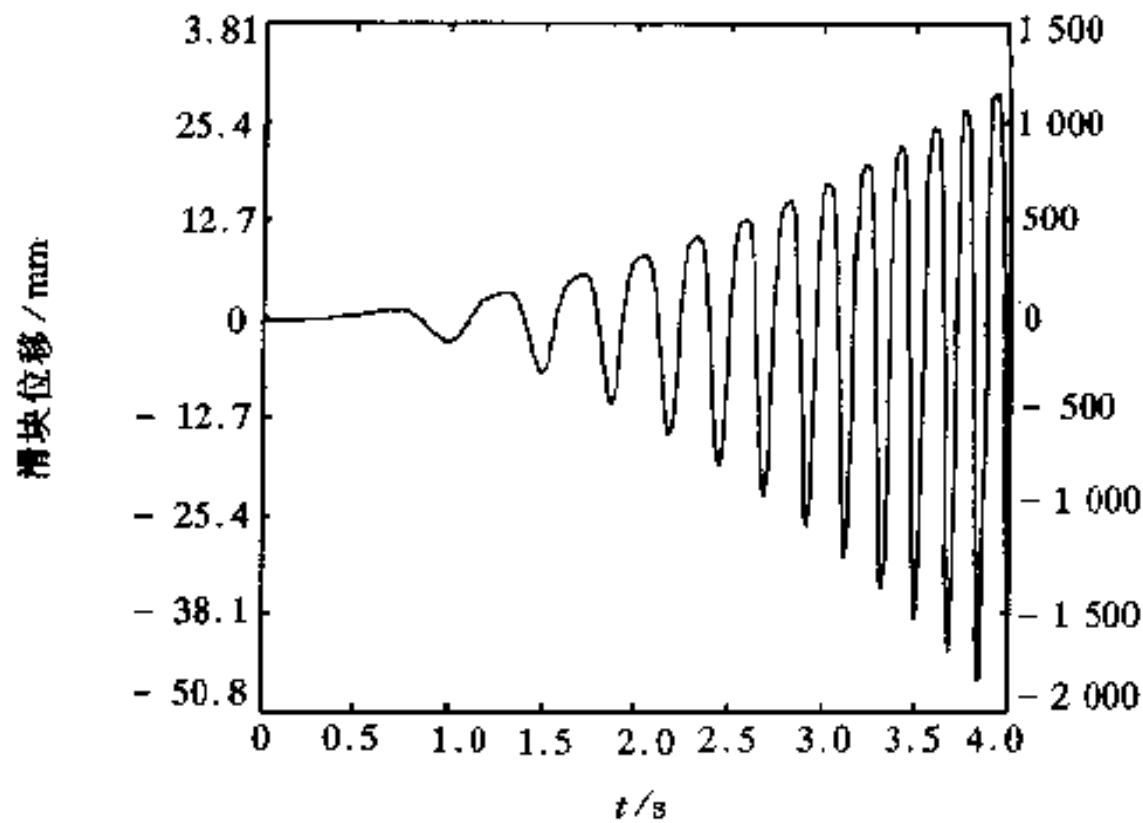


图 4-13 匀加速输入下滑块位移的线性加速过程仿真

例 4-2

匀速仿真初始条件的设置要稍许复杂些,因为必须首先针对机构的初始状态求解一次速度问题,以便为仿真提供必要的初始条件。下面以前面仿真用过的同样的速度 188.5 rad/s 为例来加以说明。很显然, ω_2 积分器的初始条件是确定的 (188.5 rad/s)。为了给出另外两个初始条件,必须求解对应此条件的速度问题 (参见例 2-1)。

```
>> th20=0;
>> th30=0;
>> r10=5.0;
>> om20=188.5;
>> r2=1.0;
>> r3=4.0;
>> a=[r3*sin(th30) 1; -r3*cos(th30) 0 ]

a =

    0 1
   -4 0

>> b=[-r2*om20*sin(th20); r2*om20*cos(th20)]

b =

    0
  188.5000

>> x=inv(a)*b
```

```

x =
-47.1250
0

>> om30=x(1);
>> rldot0=x(2);

```

表 4-3 汇总了仿真的初始条件

表 4-3 匀加速仿真的初始条件

积分器	环境变量	初始条件
θ_2/rad	th20	0
θ_3/rad	th30	0
$r_1/\text{mm}(\text{in})$	th20	177(5.0)
$\omega_2/\text{rad} \cdot \text{s}^{-1}$	om20	188.5
$\omega_3/\text{rad} \cdot \text{s}^{-1}$	om30	-47.125
$\dot{r}_1/\text{mm} \cdot \text{s}^{-1}(\text{in} \cdot \text{s}^{-1})$	rldot0	0(0)

与前面类似,仿真共运行了 0.07 s,图 4-14 给出了整个仿真过程滑块的加速度图。

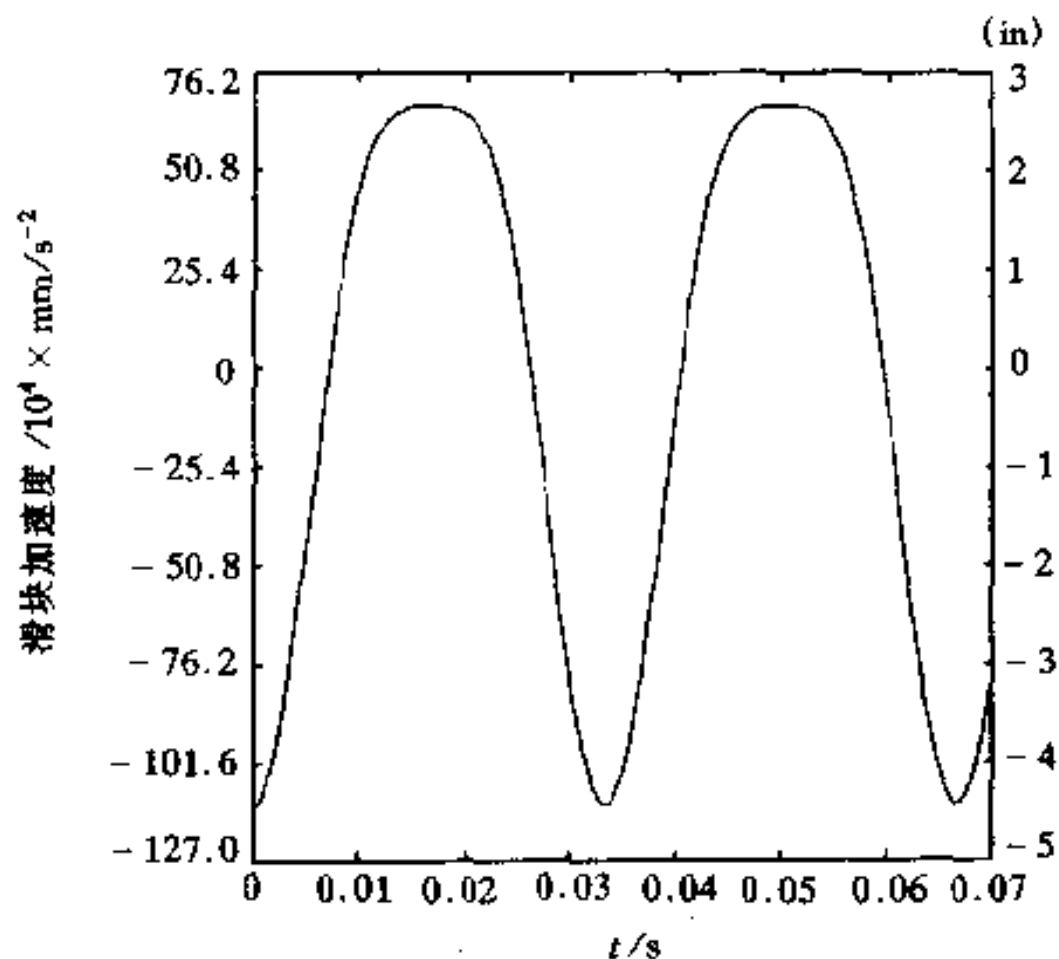


图 4-14 例 4-2 中滑块的加速度历程

对此仿真结果作一些解释是很意义的。此机构的曲柄以 1 800 r/min (小型汽油发动机的典型工作速度) 的转速旋转, 曲柄长度为 25.4 mm (1 英寸), 连杆长度为 101.6 mm (4 英寸)。然而, 滑块的最大加速度峰值超过 $1\ 016/\text{m} \cdot \text{s}^{-2}$ ($40\ 000\ \text{in}/\text{s}^2$), 这对应着质量超过 100 g 的物体的加速度。因此, 滑块所经受的动态力超过其自身重量的 100 倍。显然, 即使对于尺寸和转速均不大的机构, 动态力也是很大的。在后面的章节中, 将引入动态仿真来直接求解机构的内力。

最后, 考察连杆的角速度 ω_3 (图 4-15)。因为此例仅考虑曲柄匀速运动的情况, 仿真结果应当和图 4-9 给出的仅考虑速度的仿真结果一样。正如期望的那样, 这两个图给出了完全一样的结果。

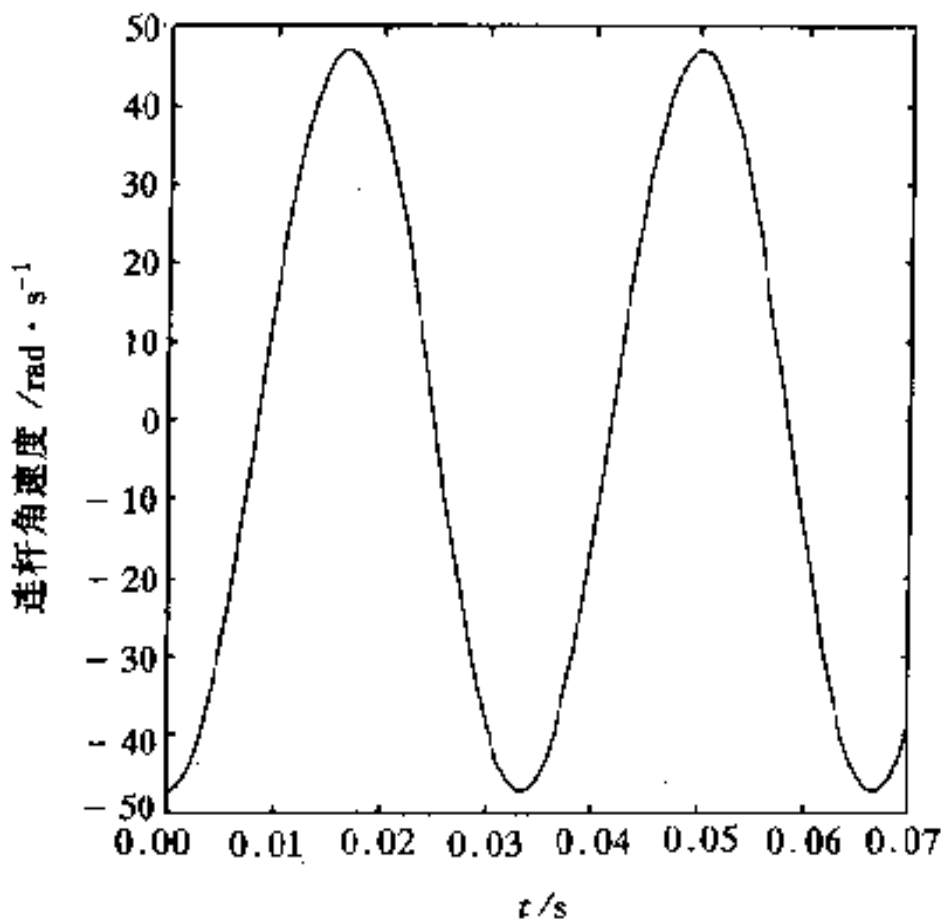


图 4-15 曲柄匀速转动时连杆的角速度随时间变化的曲线

4.4 相容性检验

运动学仿真最令人感兴趣的特性之一就是仿真计算可以得到丰富的数据。例如, 通过对曲柄滑块机构最简单的分析就可以看出该机构是一个单自由度系统, 只需一个坐标就可以完全确定各个构件在任意瞬时的位置。另一方面, 运动学仿真独立地计算出了 θ_2 、 θ_3 和 r_1 的值。当我们认识到闭环矢量方程是求解机构速度和加速度的基础这一事实时, 这一明显的矛盾就可以迎刃而解了。因为, 这些变量的积分结果也必须满足闭环矢量方程。通过检验机构最终位置满足原有闭环矢量方程的程度, 我们就可以得到仿真自身结果的有效性。

为此引入一个新的函数, 将位移变量作为输入, 函数返回的变量代表计算中的

固有误差。对曲柄滑块机构,回想其闭环矢量方程为

$$\mathbf{R}_1 - \mathbf{R}_2 - \mathbf{R}_3 = 0 \quad (4-10)$$

定义误差矢量为

$$\mathbf{E} = \mathbf{R}_1 - \mathbf{R}_2 - \mathbf{R}_3 \quad (4-11)$$

则 E 的模是一个表示仿真有效程度的标量。一种典型情况是,如果在仿真框图的结构上或函数内部出现错误而导致编程中出现错误,初始误差就较大,而且误差很快就会无限制地增长,此时就需要为以后程序调试设置误差指示量。另一方面,如果误差从零开始缓慢增加,但仍然在仿真所需的时间段内(对曲柄滑块机构一般大约为 10 倍周期)达到较大的值,那么认为数值积分不够收敛。为了解决此问题,可以降低 Simulink 的误差限。最后一种情况,如果误差从某个非零值开始稳定增长,则设定的初始条件不是自相容的,需要重新设定。

新函数 comperr 的脚本如下所示:

```
comperr.m
function e=comperr(u)
%
%
% function to compute the displacement error
% inherent in the kinematic simulation
%
% u(1) - theta-2
% u(2) - theta-3
% u(3) - r1
%
r2 = 1.0;
r3 = 4.0;
%
ex = u(3) - r2*cos(u(1)) - r3*cos(u(2));
ey = -r2*sin(u(1)) - r3*sin(u(2));
%
e=norm([ex ey]);
```

此函数可以很容易地被嵌入到速度仿真或加速度仿真系统中,如图 4-16 所示。

当仿真系统修改之后,仿真重新运行。现在的工作环境有两个矩阵 simout 和 error。为了查看在整个运行过程中误差是如何变化的,可以执行以下命令:

```
>> plot(simout(:,1),error)
>> xlabel('Time (sec)')
>> ylabel('RMS Consistency Error (in)')
```

结果如图 4-17 所示。注意到相容性误差接近常量且量值很小。

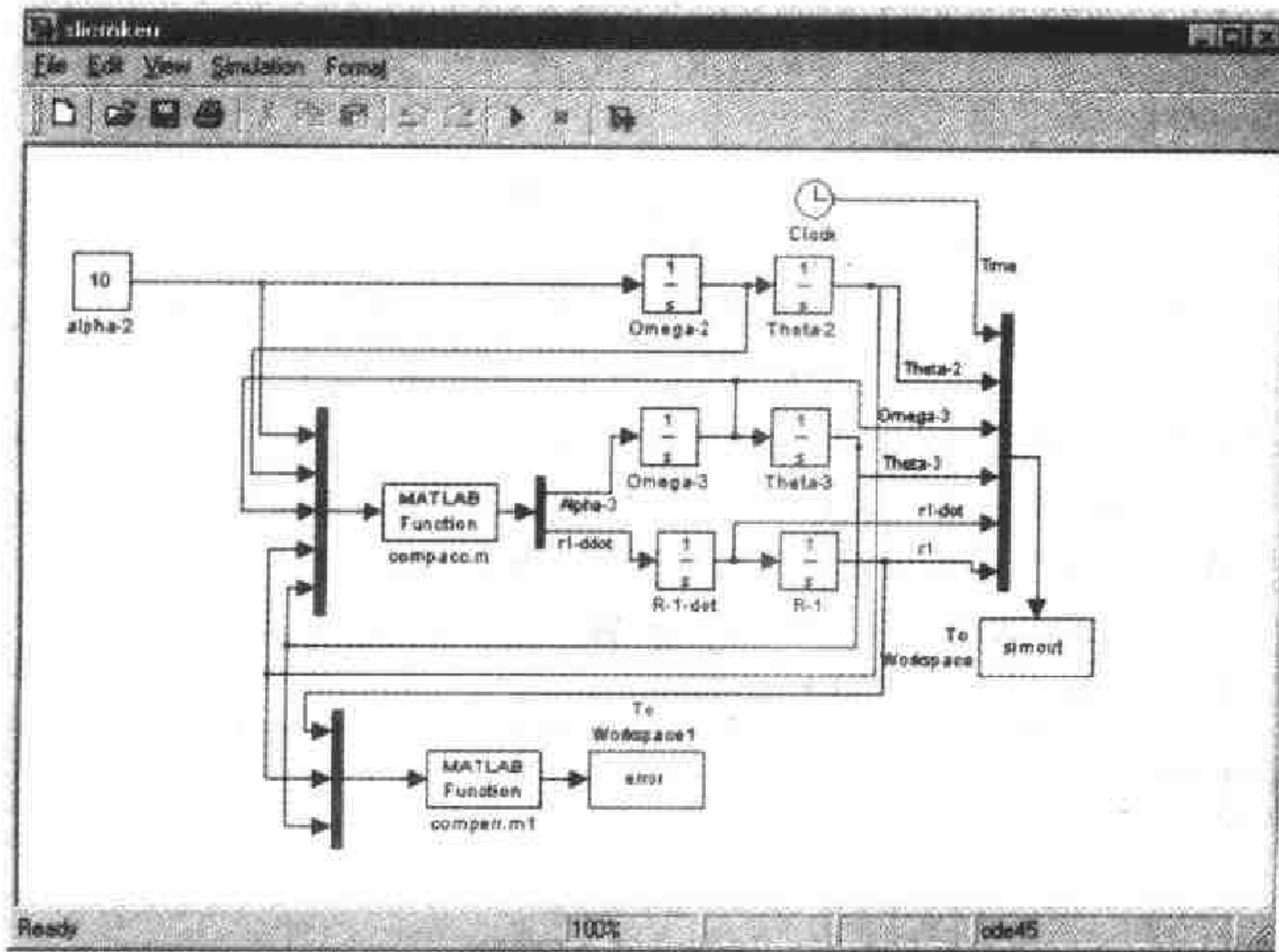


图 4-16 具有相容性检验的曲柄滑块机构仿真框图 (文件: slicrnkerr.mdl)

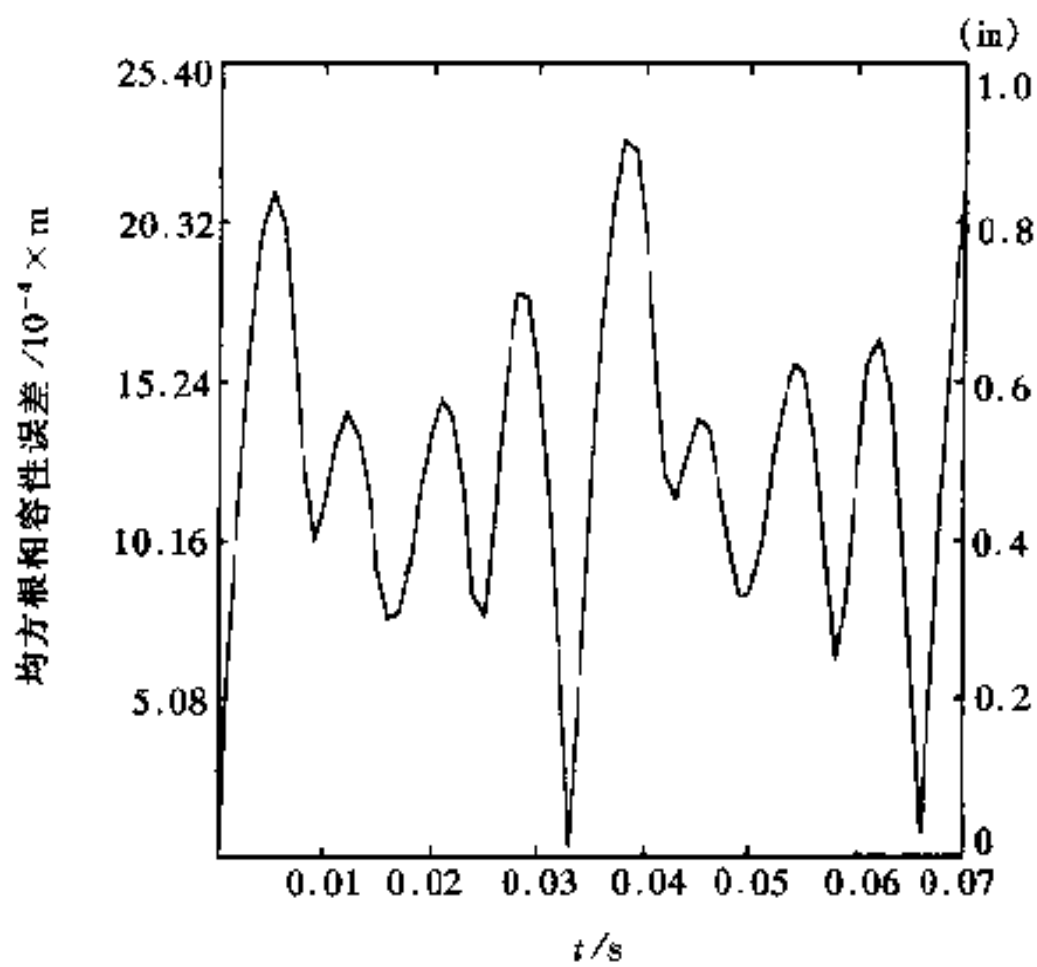


图 4-17 曲柄滑块机构仿真的均方根误差随时间的变化关系图

4.5 四连杆机构的运动学仿真

为了进一步说明运动学仿真的概念,曲柄滑块机构的仿真系统将被修改以用于对四连杆机构进行仿真。这只需对 Simulink 模型作少许修改即可完成。两个函数 compacc.m 和 conchk.m 必须作修改以反映适当的闭环矢量方程。在第二章中已给出了四连杆机构的闭环矢量方程。方程(2-6)和(2-7)表示了方程的 x 和 y 分量的一阶导数。进一步求二阶导数,得到

$$\begin{bmatrix} -r_3 \sin\theta_3 & r_4 \sin\theta_4 \\ r_3 \cos\theta_3 & -r_4 \cos\theta_4 \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} r_2 a_2 \sin\theta_2 + r_2 \omega_2^2 \cos\theta_2 + r_3 \omega_3^2 \cos\theta_3 - r_4 \omega_4^2 \cos\theta_4 \\ -r_2 a_2 \cos\theta_2 + r_2 \omega_2^2 \sin\theta_2 + r_3 \omega_3^2 \sin\theta_3 - r_4 \omega_4^2 \sin\theta_4 \end{bmatrix}$$

新函数 compacc4.m 的 MATLAB 脚本如下:

```
compacc4.m
function [x]=compacc4(u)
%compacc4 computes accelerations of a four-bar mechanism
%
% function used in kinematic simulation of 4-bar
% mechanism.
%
%Copyright 2001
%John F. Gardner
%
% u(1) = alpha-2
% u(2) = omega-2
% u(3) = omega-3
% u(4) = omega-4
%u(5) = theta-2
% u(6) = theta-3
% u(7) = theta-4
%
% Define the geometry
%
r1 = 12.0; r2 = 4.0;
r3 = 10.0; r4 = 7.0;
%
a = [-r3*sin(u(6)) r4*sin(u(7)) ;r3*cos(u(6)) -r4*cos(u(7))];
%
b(1) = r2*u(1)*sin(u(5))+r2*u(2)^2*cos(u(5))+r3*u(3)^2*cos(u(6))-
r4*u(4)^2*cos(u(7));
%
b(2) = -2*u(1)*cos(u(5))+r2*u(2)^2*sin(u(5))+r3*u(3)^2*sin(u(6))-
r4*u(4)^2*sin(u(7));
%
x=inv(a)*b';
```

仿真框图只需作少许修改,具体内容如图 4-18 所示。

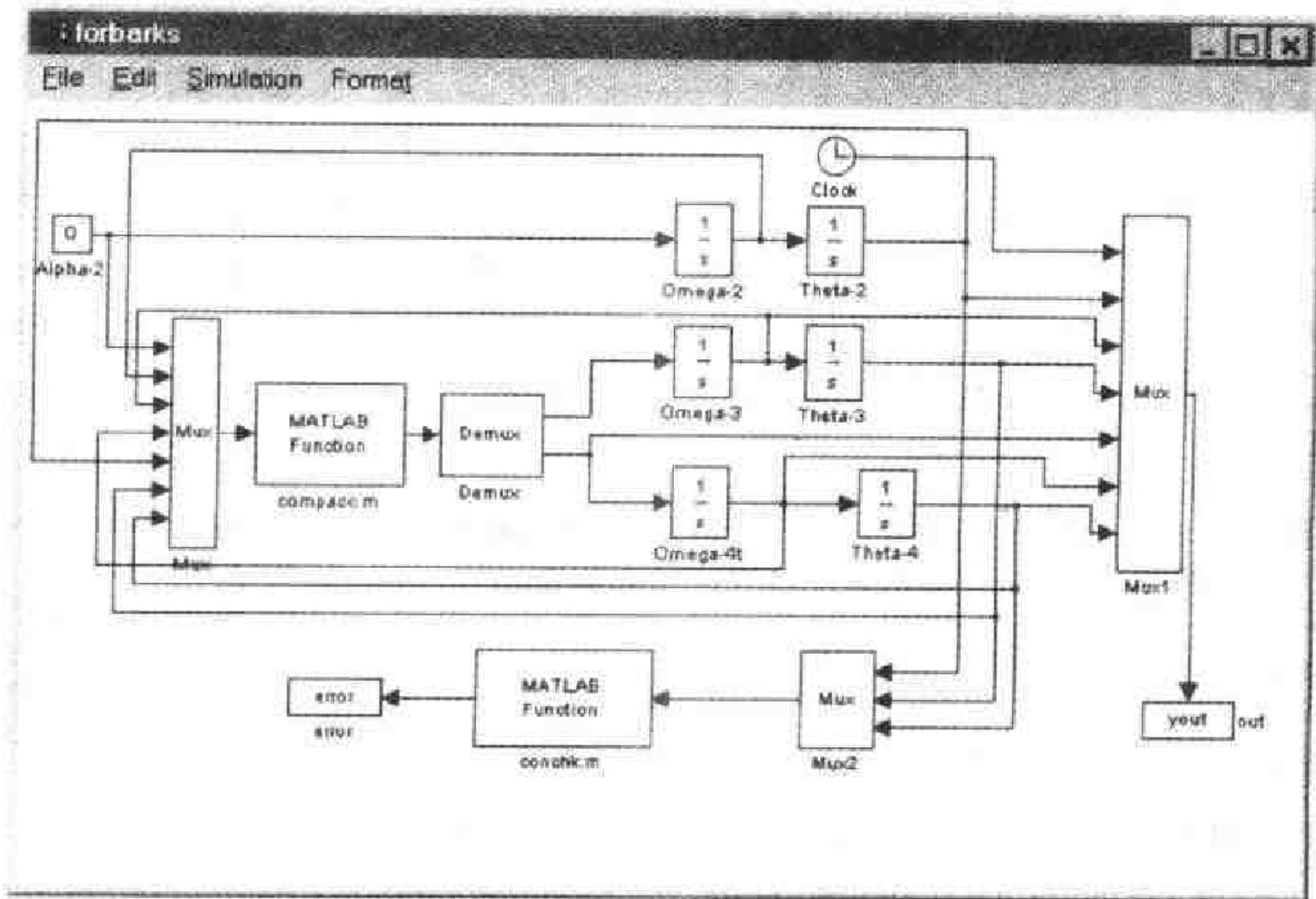


图 4-18 四连杆机构运动学仿真的模块框图(文件:fourbarks.mdl)

运行此仿真,必须提供一组前面已讨论过的相容初始条件。恰巧第三章中已经给出了四连杆机构的位置解,所以速度求解可以用闭环矢量方程很容易地完成。下面的例子说明四连杆机构运动学仿真的应用。

例 4-3

考察具有如下尺寸的四连杆机构连杆 3 和 4 的加速度和速度。

参数	值
r_1/cm	120
r_2/cm	40
r_3/cm	100
r_4/cm	70
$\theta_2/\text{rad}(t=0)$	0
$\omega_2/\text{rad} \cdot \text{s}^{-1}$	-250

首先,在第 3 章中引入的 MATLAB 程序 possol4 将被用于计算对应于该初始位置的另外两个角度(θ_3 和 θ_4),其程序运行结果如下:

```
>> rs = [120 40 100 70]
rs =
 120 40 100 70
>> th=[0 45*pi/180 135*pi/180]

th =

    0    0.7854    2.3562

>> possol4(th,rs)

ans =

    0.7688    1.6871
```

MATLAB 的默认设置是仅显示计算结果的五位有效数字,但它也可以计算得到具有很多位有效数字的结果。由于截断误差的累积特性,这些长精度的仿真对于工程中很少的一部分需要保留尽可能多位有效数字的情况是适用的。为了从 MATLAB 中得到长精度计算结果,要执行下列命令:

```
>> format long
>> ans
ans =

    0.76879354899128    1.68712997846810
```

然后通过剪切和粘贴将这些结果插入到仿真系统中 θ_3 和 θ_4 积分器的初始条件框中。

最后,针对在表中给出的初始条件 ($\theta_2=0$) 和 $\omega_2=-250$ rad/s 求解机构的速度问题。方程 (2-8) 给出了矩阵形式的速度结果。按照例 2-1 中的方法,再次通过 MATLAB 程序求解 ω_3 和 ω_4 ,并作为一组相容的初始条件。在执行完前面给出的一组 MATLAB 命令后, θ_3 和 θ_4 被存储在工作环境中的 ans 矢量中,程序命令及运行结果如下:

```
>> th3=ans(1);
>> th4=ans(2)

>> a=[-rs(3)*sin(th3)  rs(4)*sin(th4)
rs(3)*cos(th3)  -rs(4)*cos(th4)]

a =

-69.52686081652185    69.52686081652185
 71.87499999999999    8.125000000000001
```

```
>> b = [-250*rs(2)*sin(0); 250*rs(2)*cos(0)]
```

```
b =
```

```
    0
 10000
```

```
>> inv(a)*b
```

```
ans =
```

```
 125
 125
```

这些初始条件是很容易被实现的。

若曲柄以 250 rad/s 的转速旋转, 机构在 0.025 s 之内旋转一周。如果仿真运行 0.1 s, 则大约模拟 4 圈。图 4-19 是连杆 3 和 4 的角加速度随时间变化图。在图中可以很明显地看出完整的 4 个周期。

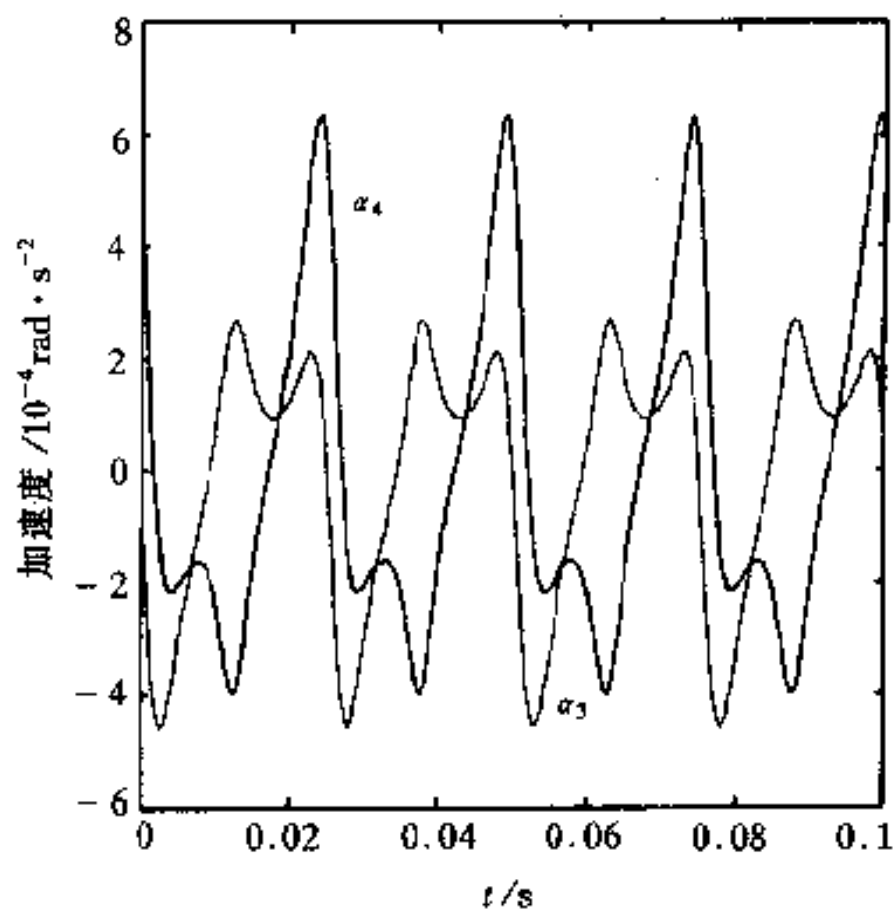


图 4-19 例 4-中的四连杆机构在曲柄匀速转动下连杆 3 和 4 的角速度

类似地, 图 4-20 显示了这两个连杆的角速度仿真结果。

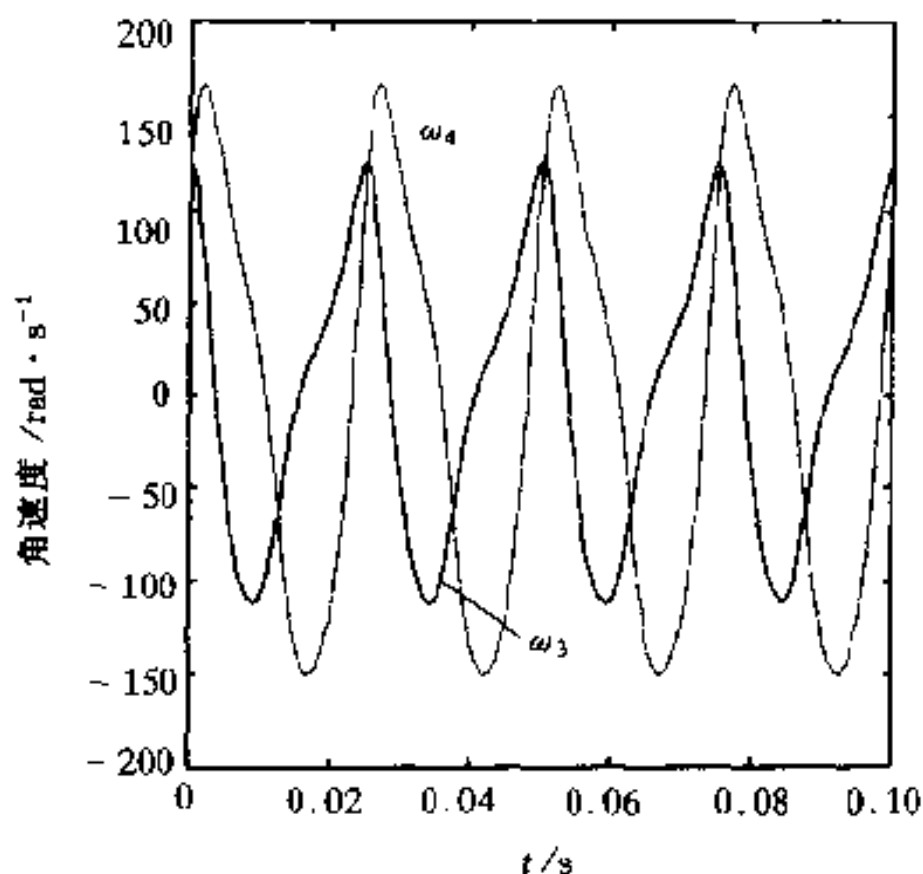


图 4-20 例 4-3 中四连杆机构中连杆 3 和连杆 4 的角速度

4.6 小结

本章引入了运动学仿真的概念。运动学仿真是指系统输入为某个输入连接件的速度或加速度的仿真过程。闭环矢量方程的导出方程被用于求解机构中其余构件的加速度或速度。这些加速度和速度又对时间进行积分从而获得速度和位移。运动学仿真的一个重要方面是为积分器设定适当的初始条件。通常需要工程人员根据现有的办法求解出机构的至少一个位置问题或速度问题。从下一章的内容可以看出,由于很小的截断误差可以被数值积分程序很快地扩大,因而明确这些初始条件应具有精度是非常关键的。

第 4 章习题

1. 某个四连杆机构具有如下的连接长度

$$r_1 = 10.0 \text{ cm}$$

$$r_2 = 3.0 \text{ cm}$$

$$r_3 = 8.0 \text{ cm}$$

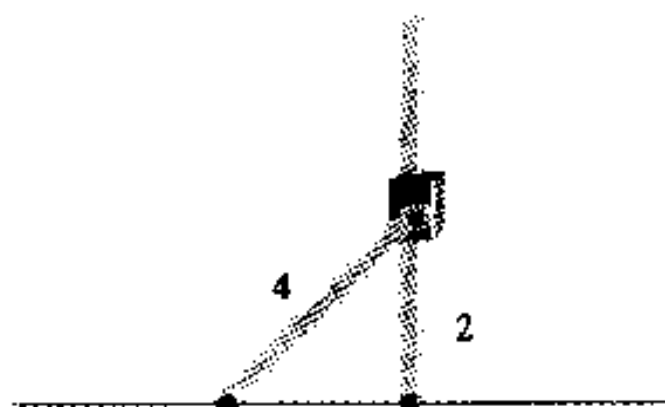
$$r_4 = 6.0 \text{ cm}$$

实现该机构的运动学仿真,并在曲柄(连杆 2)以 6 000 r/min 的匀速旋转下模拟曲柄运行若干圈的过程。通过仿真回答下述问题:

- 仿真的初始条件是什么?
- 连杆 4 的最大角加速度是多少?
- 连杆 2 和 3 之间的铰链的最大速度是多少?
- 连杆 3 和 4 之间的铰链的最大速度是多少?
- 连杆 4 的最大角速度是多少?
- 连杆 4 上中间一点所经历的最大平移速度是多少? 此时加速度的方位如何?

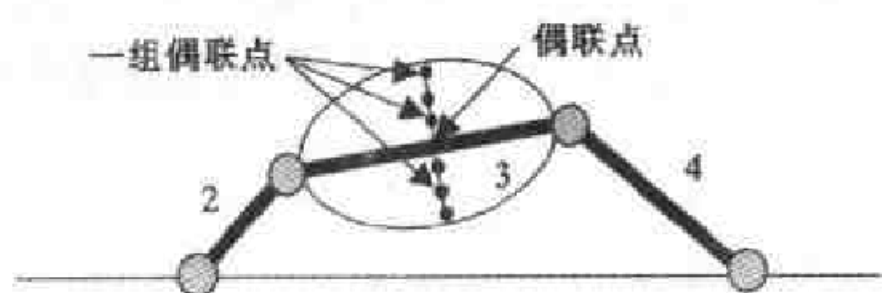
2. 针对题 1 中的四连杆机构, 对曲柄初始速度为 $6\ 000\ \text{r/min}$, 以大小为 $-15\ \text{rad/s}^2$ 匀加速运动的情况进行仿真。

3. 考虑如右图所示的倒置的曲柄滑块机构, 实现机构的运动学仿真。连杆长度 $r_1 = 0.5\ \text{m}$, $r_4 = 0.72\ \text{m}$ 。假设曲柄以 $200\ \text{rad/s}$ 的匀速旋转, 画出曲柄旋转一圈滑块的平移加速度。对应于 r_4 从 $0.5\ \text{m}$ 变化到 $1.0\ \text{m}$ 的五个不同参数值, 画出相应的一组加速度图形。从这些图形中可以得出什么结论?

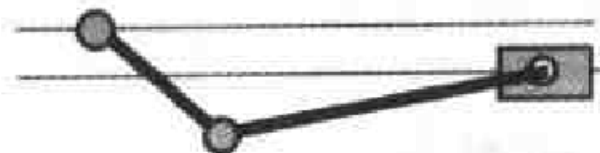


题 3 图

4. “连杆曲线”是随四连杆机构的连接件 3 (连杆) 运动的点所经历的轨迹。修改用于习题 1 中的运动学仿真以跟踪连杆上任意一点的连杆曲线。使用 MATLAB 画出习题 1 中给出的机构的连杆曲线, 偶联点位于连杆的中点。画出过连杆中点与连杆垂直的线上一组偶联点 (见下图) 的连杆曲线。



题 4 图



题 5 图

5. 偏置曲柄滑块机构是一种滑块运动轴线不通过曲柄旋转中心的曲柄滑块机构 (见题 5 图)。导出偏置曲柄滑块机构的闭环矢量方程, 并对闭环矢量方程求适当的导数以便能够实现包括加速度的运动学仿真。偏置曲柄滑块机构的仿真和不偏置的曲柄滑块机构有何不同?

第5章 动力学引论

5.1 概述

在本章里,我们将为本书引入以计算机为基础,且功能强大的约束系统动态仿真打下基础。以一个相对简单的例子为例,我们将建立一个连接有单摆的滑块在斜面上运动的系统(图5-1)的完整动态仿真过程,并考察系统在不同条件下的特性。在这个简单的例子中,我们将引入动态仿真的联立约束法,该方法将运动约束和牛顿-欧拉方程组成一组联立线性方程组,通过求解该方程组得到仿真所需的约束反力和加速度。

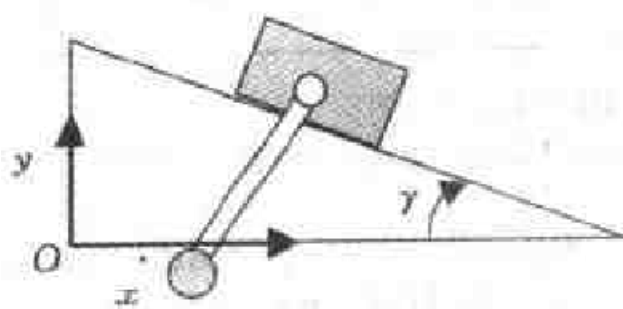


图5-1 连接有单摆的滑块在斜面上的运动,(注意坐标系的原点选在斜面拐角处,原图没有标出角度 γ 应该标出来)

5.2 第1步:滑块在斜面上的仿真

我们依次进行分析。首先从分析滑块在斜面上的运动开始,此时忽略与之相连的摆。图5-2是滑块的受力图,注意摩擦力 F_f ,法向反力 F_N (原文中用 N 表示)和重力 $m_s g$ (原文中质量用 M 表示——译者注)均画在图中。

由滑块的受力图可以立刻导出下面的两个运动方程

$$F_N \sin \gamma - F_f \cos \gamma = m_s \ddot{x}_s \quad (5-1)$$

$$F_N \cos \gamma - m_s g + F_f \sin \gamma = m_s \ddot{y}_s \quad (5-2)$$

其中 (x_s, y_s) 表示滑块质心在参考坐标系中的坐标。

摩擦力 F_f 与运动方向相反,与法向反力成正比,可表示如下

$$F_f = \sigma \mu F_N \quad (5-3)$$

其中

$$\sigma = \begin{cases} +1 & \text{沿斜面向下运动} \\ -1 & \text{沿斜面向上运动} \end{cases}$$

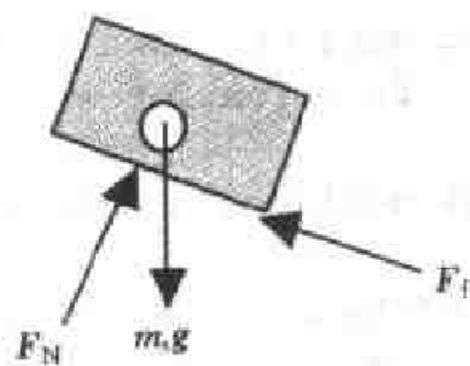


图5-2 单个滑块的受力图

最后,我们注意到两个加速度并不是独立的。斜面为滑块的运动提供了约束,使得其速度必需始终与斜面平行。从数学的角度讲,这可以表述成下面的方程

$$\dot{x}_s \tan \gamma = -\dot{y}_s \quad (5-4)$$

由此我们可以很容易地导出加速度之间的约束

$$\ddot{x}_s \tan \gamma = -\ddot{y}_s \quad (5-5)$$

这些方程(5-1,5-2和5-5)可以被合并形成一个由三个方程构成的含三个未知量的方程组,即

$$\begin{bmatrix} m_s & 0 & \sigma\mu \cos \gamma - \sin \gamma \\ 0 & m_s & -(\sigma\mu \sin \gamma + \cos \gamma) \\ \tan \gamma & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x}_s \\ \ddot{y}_s \\ F_N \end{bmatrix} = \begin{bmatrix} 0 \\ -m_s g \\ 0 \end{bmatrix} \quad (5-6)$$

图 5-3 显示了这个简单系统的仿真模型。计算加速度的函数程序如下:

```

rampslider.m
function out=rampslider(u)
%
% function to compute accelerations of the slider
% on an inclined plane with friction
%
% Author: J.F. Gardner
%
% Copyright 2001
%
% u(1) = x-dot
% u(2) = y-dot
%
m = 1.0; % kg
gamma = 30 *pi/180; % radians
mu = 0.5;
g = 9.8687;
%
% compute the down-ramp velocity to find direction
% of friction force
%%
sdot=u(1)*cos(gamma)-u(2)*sin(gamma);
%
if sdot > 0
    sigma = 1.0;
else
    sigma = -1.0;
end
%
a=[m      0      sigma*mu*cos(gamma)-sin(gamma) ;
   0      m      -(sigma*mu*sin(gamma)+cos(gamma)) ;
   tan(gamma) 1      0];

```

```

%
b=[0;-m*g;0];
%
out=inv(a)*b;

```

从图 5-3 可以看出,这个仿真没有输入,系统的响应完全由初始条件决定。运行这个仿真并不是很有意义,仿真结果仅仅只能证明滑块沿斜面下滑的加速度等于重力加速度沿斜面的分量。而这一结果很容易从基本的物理学知识就可以得到。另一方面,要对图 5-1 所示的滑块-摆系统建立更为复杂的数学模型则是一项更为困难而艰巨的任务。

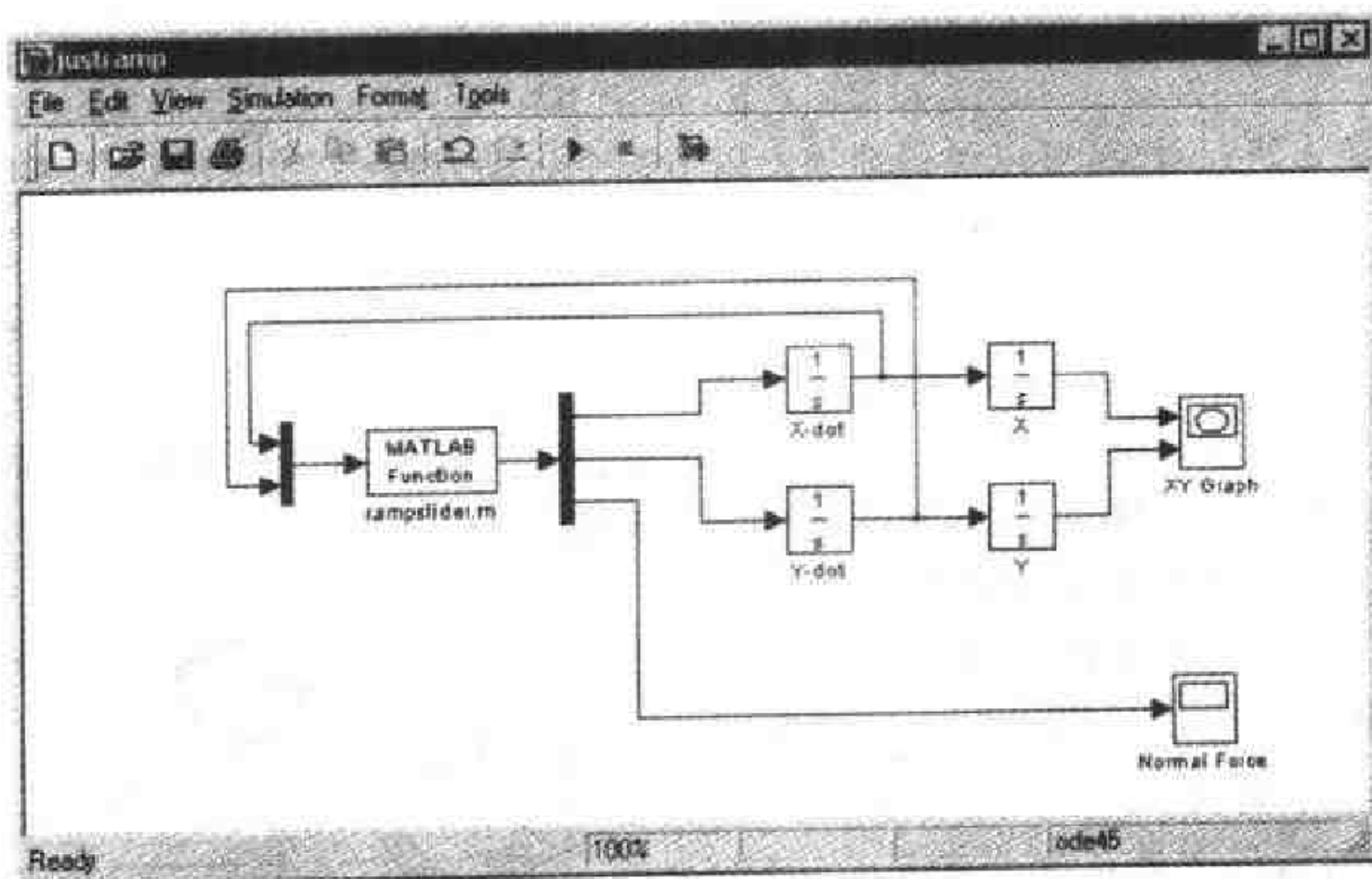


图 5-3 斜面上滑块的 Simulink 仿真模型(图 5-1 所示的滑块-摆系统建模第一步(文件:justramp.mdl))

5.3 第 2 步:添加摆

图 5-4 是滑块的受力图,图中画出了滑块和摆之间的相互作用力,需要注意的是这种表示物体之间相互作用力的习惯用法。 F_{px} 是摆施加给滑块 x 方向的分力。这一习惯用法(注意下标的排列首先是力的来源,然后是力的指向)将在本书的第 6 章中正式引入,并贯穿全书。

图 5-5 画出了摆自身的受力图。注意滑块施加给摆的作用力的下标与摆对滑块作用力的下标相反。

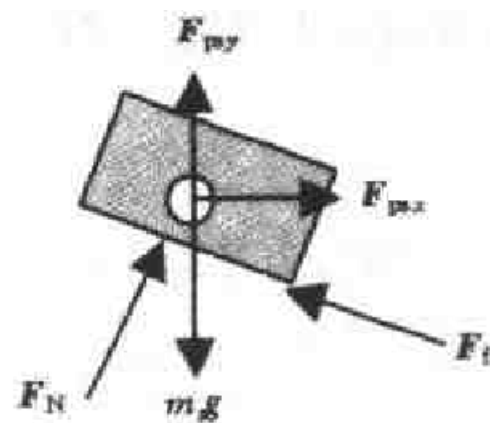


图 5-4 标注有摆作用力的滑块受力图

这时,我们将得到如下的事实:对于两个连接于同一点的刚体 A 和 B, A 作用于 B 的力与 B 作用于 A 的力二者大小相等方向相反。因此,我们可以很容易地得出 $F_{sly} = -F_{psx}$ 以及 $F_{slx} = -F_{psy}$ 。下面给出的运动方程隐含地利用了这些关系。

稍作修改,即可得到反映摆对滑块作用的滑块的运动方程

$$F_N \sin \gamma - \sigma \mu F_N \cos \gamma - F_{slx} - m_s \ddot{x}_s \quad (5-7)$$

$$F_N \cos \gamma - m_s g + \sigma \mu F_N \sin \gamma - F_{sly} = m_s \ddot{y}_s \quad (5-8)$$

以摆为研究对象,可以类似地导出以下三个运动方程

$$F_{spx} = m_p \ddot{x}_p \quad (5-9)$$

$$F_{sly} - m_p g = m_p \ddot{y}_p \quad (5-10)$$

$$r_p \cos \theta_p F_{sly} + r_p \sin \theta_p F_{spx} = I_p \ddot{\theta}_p \quad (5-11)$$

最后一个方程表示所有力对质心的力矩求和。 θ_p 是由 x 轴正向转向摆与滑块连线所形成的夹角。这与图 5-6 给出的位移矢量图是一致的。

由此直接导出下面的闭环矢量方程:

$$\mathbf{R}_1 + \mathbf{R}_3 = \mathbf{R}_p \quad (5-12)$$

x 和 y 方向的投影方程可以很容易地导出。注意到 \mathbf{R}_2 和 \mathbf{R}_p 仅仅是关于 x 和 y 变量的矢量,而与长度和角度变量无关,因为这两个矢量并不与一个真实的连接件相对应。

$$x_s + r_p \cos \theta_p = x_p \quad (5-13)$$

$$y_s + r_p \sin \theta_p = y_p \quad (5-14)$$

对上面的方程求两次导数并重新组合,可以得到关于滑块和摆质心加速度的约束方程如下

$$\ddot{x}_s - \ddot{x}_p - r_p (\sin \theta_p) \ddot{\theta}_p = r_p \omega_p^2 \cos \theta_p \quad (5-15)$$

$$\ddot{y}_s - \ddot{y}_p + r_p (\cos \theta_p) \ddot{\theta}_p = r_p \omega_p^2 \sin \theta_p \quad (5-16)$$

最后,为了内容的完整性,我们重复一下约束方程,该方程反映了滑块仅能沿斜面运动这一事实。

$$\ddot{x}_s \tan \gamma = -\ddot{y}_s \quad (5-17)$$

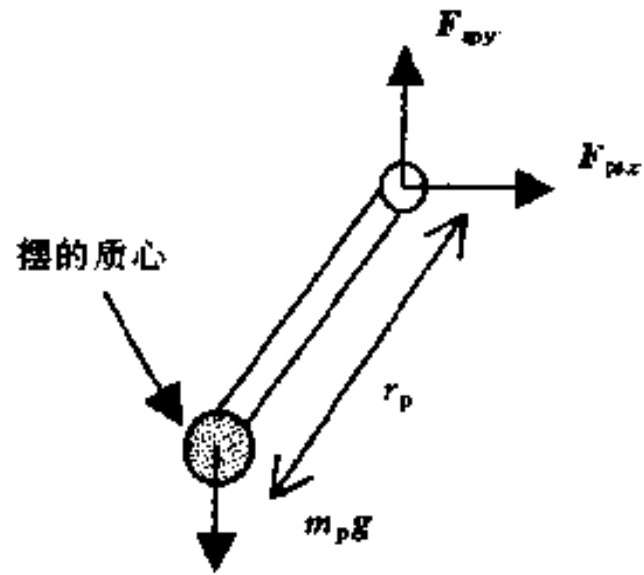


图 5-5 摆的受力图

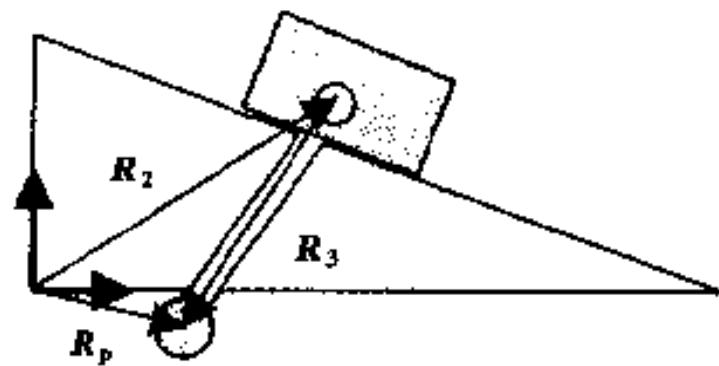


图 5-6 图 5-1 中各运动物体的位移矢量图

5.4 第 3 步:矩阵方程的组装

方程(5-7)至(5-11)以及方程(5-15)至(5-17)形成了一个由 8 个方程所组成的线性方程组,其未知量依次是: \ddot{x}_s , \ddot{y}_s , \ddot{x}_p , \ddot{y}_p , $\ddot{\theta}_p$, N , F_{spx} 和 F_{spx} 。给出该方程组的矩阵形式如下

$$\begin{bmatrix} m_s & 0 & 0 & 0 & 0 & \sigma\mu C_g - S_g & 1 & 0 \\ 0 & m_s & 0 & 0 & 0 & -(\sigma\mu S_g + C_g) & 0 & 1 \\ 0 & 0 & m_p & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & m_p & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & I_p & 0 & -r_p S_p & r_p C_p \\ 1 & 0 & -1 & 0 & -r_p S_p & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & r_p C_p & 0 & 0 & 0 \\ S_g & C_g & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x}_s \\ \ddot{y}_s \\ \ddot{x}_p \\ \ddot{y}_p \\ \ddot{\theta}_p \\ N \\ F_{spx} \\ F_{spx} \end{bmatrix} = \begin{bmatrix} 0 \\ -m_s g \\ 0 \\ -m_p g \\ 0 \\ r_p \omega_p^2 C_p \\ r_p \omega_p^2 S_p \\ 0 \end{bmatrix}$$

(5-18)

(这里 $C_g = \cos\gamma$, $S_g = \sin\gamma$, $C_p = \cos\theta_p$, $S_p = \sin\theta_p$ ——译者注)

请注意,该矩阵方程包含了一些在第 4 章中遇到的矩阵方程的部分信息。此外,运动方程增加了运动约束。如果我们将该矩阵方程嵌入一个加速度积分的仿真模型,那么就可得到速度和位移,进而求出矩阵及方程的右端项。

至此,我们可以同时计算该系统的加速度和约束力。对于这个特定系统的动力学论述可以有更加简洁的表述方式,但上面给出的方程组只要具有大学二年级的矢量力学知识就可以导出。并且对于深入的动态分析和设计都具有至关重要作用的约束反力可以通过以上计算获得。

在下一节,我们建立一个 Simulink 仿真模型来模拟滑块和摆系统的动态特性。

5.5 第 4 步:建立动态仿真

图 5-7 给出了具体实现前面两节所导出的动力学和运动学方程的 Simulink 仿真模型。注意在矩阵方程中的五个未知加速度现在被积分了两次,其结果通过不同的显示模块画在屏幕上。以下的 m 文件被用来建立并求解矩阵方程(注意在函数内给出了系统参数的定义)。

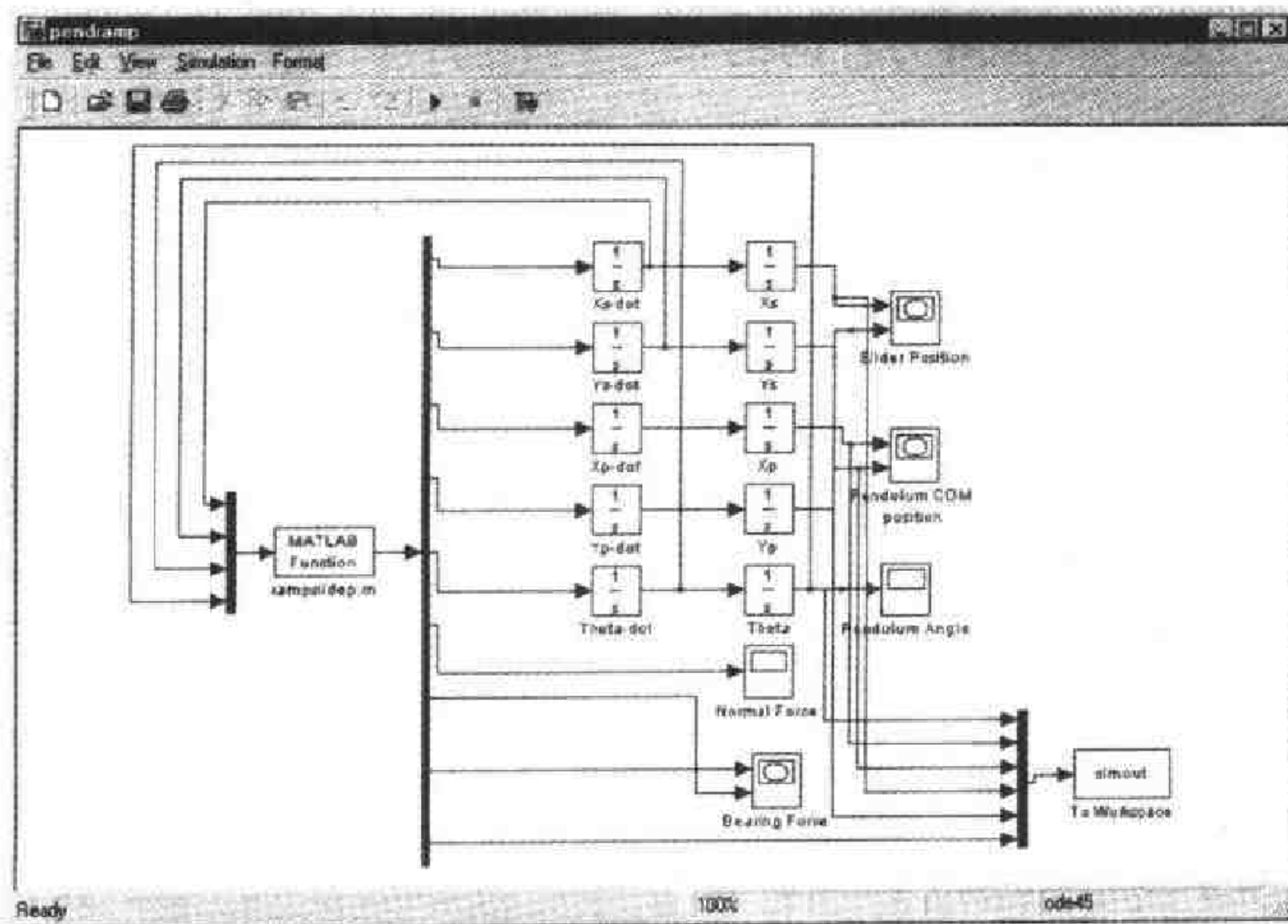


图 5-7 斜面上滑块-54 摆系统的 Simulink 仿真模型(文件: pendramp.mdl)

rampslidep.m

```
function out=rampslidep(u)
```

```
%
```

```
% function to compute accelerations of the slider  
% and pendulum on an inclined plane with friction
```

```
%
```

```
% Author: J.F. Gardner
```

```
%
```

```
% Copyright 2001
```

```
%
```

```
% u(1) = x-dot
```

```
% u(2) = y-dot
```

```
% u(3) = omega-p
```

```
% u(4) = theta-p
```

```
%
```

```
ms = 5.0; % kg
```

```
gamma = 30 *pi/180; % radians
```

```
mu = 0.5;
```

```
g = 9.8687;
```

```
rp = 0.5; % m
```

```

mp = 0.5; % kg
Ip = 0.01; % kg m^2
%
% compute the down-ramp velocity to find direction
% of friction force
%
sdot=u(1)*cos(gamma)-u(2)*sin(gamma);
%
if sdot> 0
    sigma = 1.0;
else
    sigma = -1.0;
end
%
Cp = cos(u(4));
Sp = sin(u(4));
a=zeros(8,8);
a(1,1) = ms; a(1,6) = sigma*mu*cos(gamma)-sin(gamma); a(1,7) = 1;
a(2,2) = ms; a(2,6) = -(sigma*mu*sin(gamma)+cos(gamma)); a(2,8) = 1;
a(3,3) = mp; a(3,7)=-1;
a(4,4) = mp; a(4,8)=-1;
a(5,5) = Ip; a(5,7)=-rp*Sp; a(5,8)=rp*Cp;
a(6,1) = 1; a(6,3) = -1; a(6,5) = -rp*Sp;
a(7,2) = 1; a(7,4)=-1; a(7,5)=rp*Cp;
a(8,1) = sin(gamma); a(8,2)=cos(gamma);
%
b=[0;-ms*g;0;-mp*g;0;rp*u(3)^2*Cp;rp*u(3)^2*Sp;0];
%
out=inv(a)*b;
out=[out; sdot];

```

5.6 第5步:设置初始条件并运行仿真程序

到目前为止,所有的分析都具有一般意义,可以被用于任何希望进行仿真的情况。现在,针对一些具体情况研究以下系统的响应。

为了举例说明,假设滑块-摆系统初始时刻是静止的(所有的速度积分器都具有零初始条件)。图5-8显示了对应这个例子的系统初始位置图。注意到图中滑块的中心在坐标系原点的正上方1 m处,摆处于与铅垂方向夹角 45° 的位置。

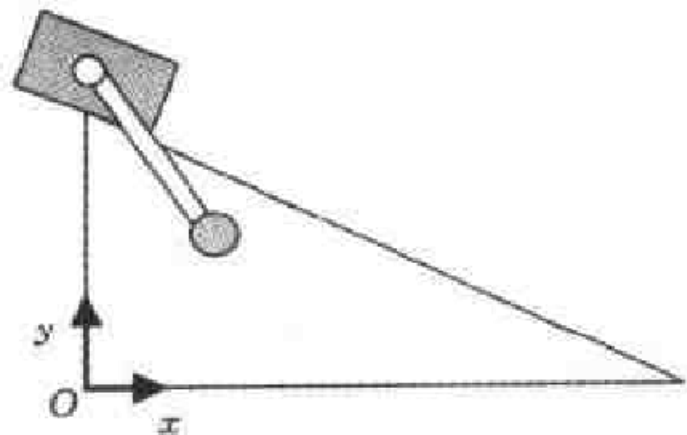


图5-8 滑块-摆系统的初始位置

表 5-1 和表 5-2 汇总了仿真所需的几何和质量参数以及八个初始条件。

表 5-1 质量和几何参数

参数	值	参数	值
m_s/kg	5.0	$\gamma/(\circ)$	30
m_p/kg	0.5	r_p/m	0.5
$I_p/\text{kg}\cdot\text{m}^2$	0.01		

表 5-2 滑块-摆系统仿真的初始条件

变量	初始条件	变量	初始条件
\dot{x}_s	0	x_s	0
\dot{y}_s	0	y_s	1.0
\dot{x}_p	0	x_p	0.353 6
\dot{y}_p	0	y_p	0.646 4
$\dot{\theta}$	0	θ	$-\pi/4$

然后运行仿真,滑块沿斜面向下运动。图 5-9 画出了滑块质心的运动轨迹以及摆来回摆动的轨迹。

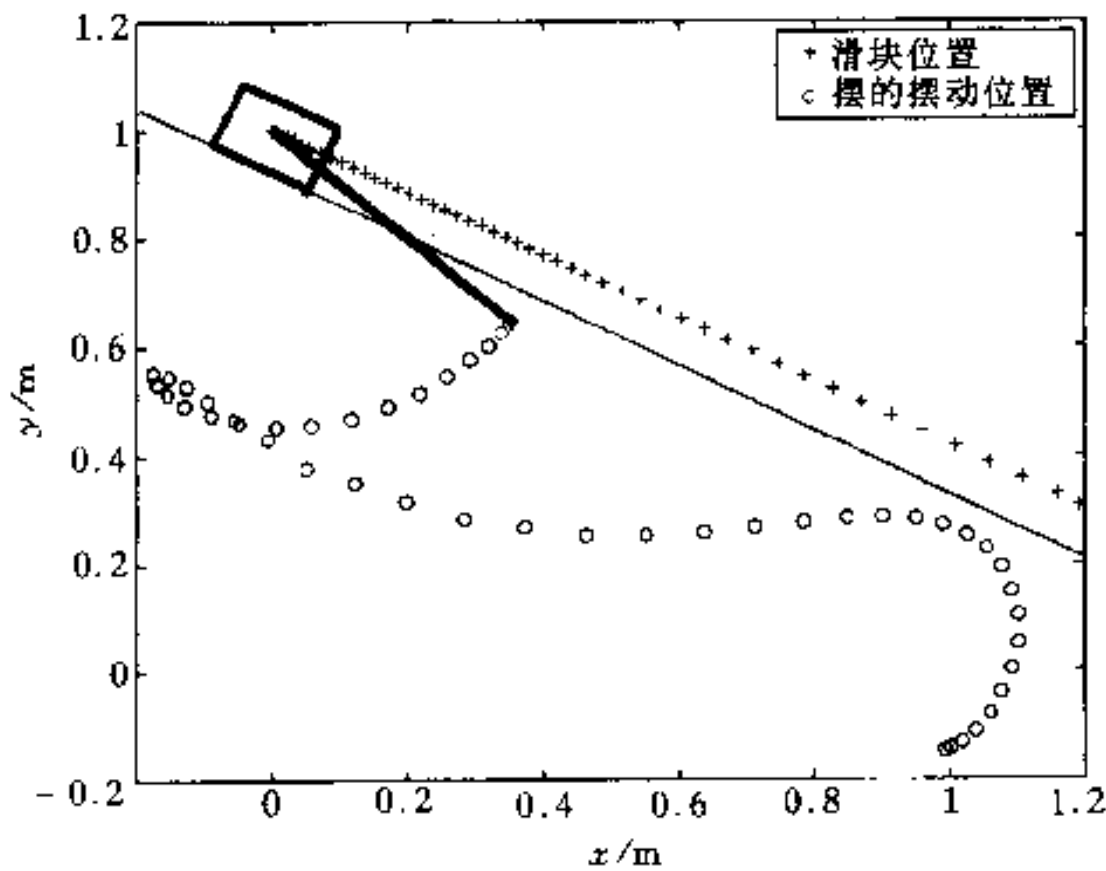


图 5-9 滑块质心以及摆来回摆动的 x 、 y 坐标图(分别用 '+' 和 'o' 表示的滑块质心和摆的运动轨迹)

虽然系统的运动很复杂,但是仿真预测的结果和我们的直觉是吻合的。滑块沿斜面向下直线运动,摆来回振荡并影响滑块的运动。图 5-10 给出了 Scope 模块的图形,显示了滑块和斜面之间法向反力的变化规律。

同样,我们注意到与摆的来回摆动相对应,法向反力也是随时间振荡的,以适应摆的来回摆动。

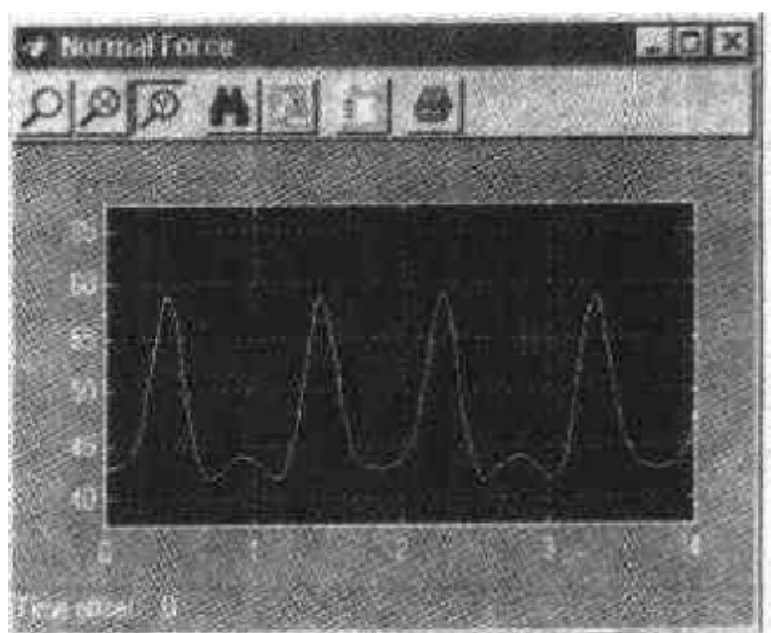


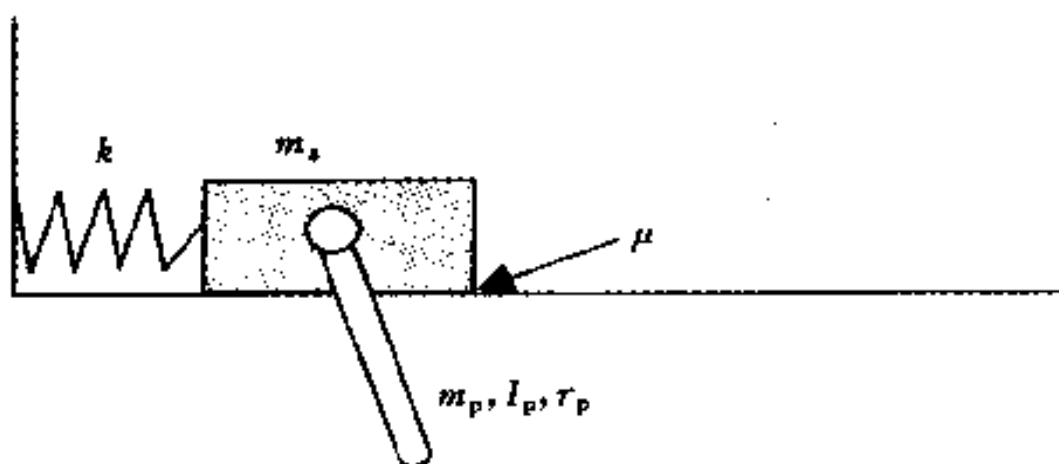
图 5-10 滑块和斜面之间的正压力

5.7 小结

在本章中通过一个简单的例子引入了联立约束法。在下一章中,我们将正式引入这个方法,并将它应用于更复杂的系统。这个方法使得对约束机械系统进行完整的动态仿真并直接计算系统的约束反力成为可能。

第5章习题

1. 考虑图示与水平滑块相连的单摆。弹簧处于原始长度,滑块的 x 坐标是零。对于下表列出的参数值,针对滑块不同的初始位置,画出滑块和摆的响应曲线。针对不同的摩擦系数 μ 重复上述过程。

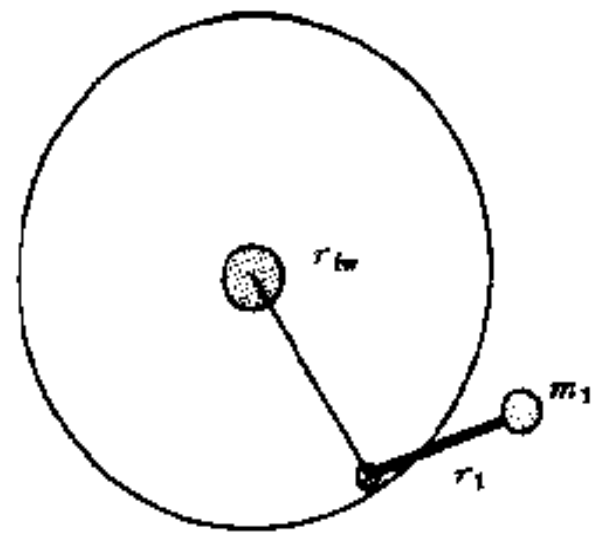


题 1 图

表 1 习题 1 参数值

参数	值	参数	值
$k/\text{kN} \cdot \text{m}^{-1}$	200	m_s/kg	2.2
m_p/kg	0.8	r_p/m	0.5
$I_p/\text{kg} \cdot \text{m}^2$	0.001	m/kg	0.2

2. 一个绕铅垂轴旋转的飞轮, 一个小质量的物体, 通过一个质量可以忽略的连杆连接到飞轮上。建立该系统的动态仿真模型, 当飞轮从静止开始在 $10 \text{ N} \cdot \text{m}$ 的扭矩作用 20 s 的过程中, 画出连杆与飞轮径向连线之间夹角的变化规律。再画出在这些条件下飞轮轴承所承受的约束反力。使用下表所列出的参数。

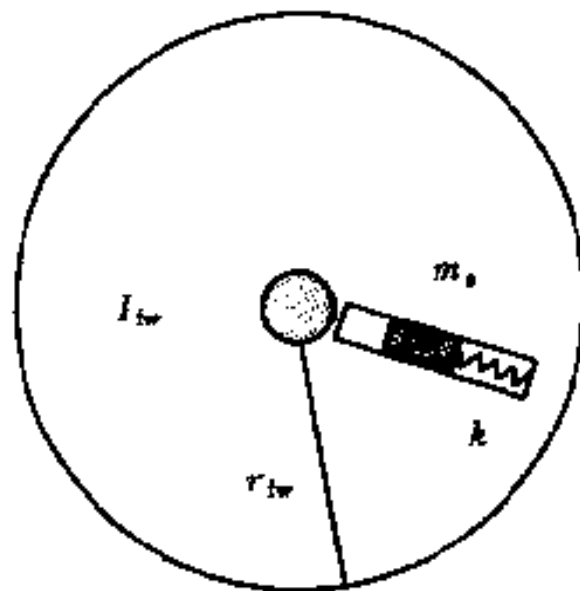


题 2 图

表 2 习题 2 参数值

参数	值	参数	值
r_{fw}/cm	25	$I_{fw}/\text{kg} \cdot \text{m}^2$	0.1
r_1/cm	20	m_1/kg	0.1

3. 考虑如下所示带有滑槽的飞轮。滑块在飞轮槽内运动, 并与一根弹簧相连, 当系统处于静止状态时弹簧处于自由长度。当飞轮以匀加速度 10 rad/s^2 从静止开始旋转时, 计算槽内滑块的相对位移。假设飞轮偏心已被调整至其中心轴。在仿真中使用下表给出的参数。



题 3 图

表 3 习题 3 参数值

参数	值
$I_{fw}/\text{kg} \cdot \text{m}^2$	1.0
r_{fw}/m	2.0
m_s/kg	0.1
$k/\text{N} \cdot \text{m}^{-1}$	10 000

第6章 联立约束法

6.1 引言

截至目前为止,我们尚未讨论外加力、铰链内力和机构的相应运动之间的联系。实际上,大部分运动学教科书都故意模糊这种联系。问题是工程技术人员都熟知的物理学最基本的前提是力产生运动,在计算结果通常不显含力的情况下如何使系统运动的研究成为可能?其答案基于这样一个事实,即机构是一个约束系统,所以相对运动的特性由机构的几何关系预先确定。此外,还有一些对大多数运动学分析几乎总是无需声明的假设。例如,当对某个输入连接件匀速运动的机构(见第4章)进行加速度和速度分析时,总是假定有一个外部装置(例如电机)提供维持匀速运动所需的任意时变扭矩。用通常的矢量多边形方法来预估所需的扭矩是件冗长乏味的事情,并使人容易忘记被分析的运动是由力产生的这一客观事实。在本章中,综合运用第2章和第4章的工作来阐述机构完整的动态仿真过程。和前面一样,将以曲柄滑块机构为例加以讨论。后续的章节将提供别的机构的例子来进一步说明这种方法。

6.2 方法概述

这种被称为联立约束法的方法在机构研究的许多不同领域有广泛的应用。其中,最值得注意的是在机器人领域的应用和 E. J. Haug 所作的工作, E. J. Haug 将该方法推广到三维的一般情况^①。虽然 Haug 给出的方法比这里给出的方法更具有一般性,但由于要给出该方法完整的空间描述需要复杂的符号,因而其所提方法一般认为更适合研究生阅读。

联立约束法建立在第4章中论述的运动学仿真的基础上,闭环矢量方程的二阶导数是仿真的基础。对每个连接件应用力平衡关系将力和加速度联系起来,然

^① Haug E. J., Computer-Aided Kinematics and Dynamics of Mechanical Systems. Englewood Cliffs, NJ; Prentice Hall, 1989

后再应用与闭环矢量方程相类似的方法得到连接件质心的加速度信息。最后,将所有的方程组装形成一个稀疏矩阵并在 MATLAB 中求解,这一步骤亦构成机构的完整的动态仿真的部分。现在将详细给出联立约束法的每个步骤。

6.2.1 力方程

图 6-1 给出了一个在机构中可以找到的典型连接件图,以这个表示连接件的受力图为起点来讨论联立约束法。注意到每个铰链只能承受力(不能承受力矩),因而可以将该力分解为 x 和 y 方向的两个分量。此外,当提到作用于机构内连接件间的力时,遵守以下的习惯表示法:一般地, F_j 表示作用在第 j 个连接杆的力,施力物体为第 i 个连接杆,可以简单地将该习惯用法记为 F_{ij} ,它表示连接杆 i 作用于 j 的力。

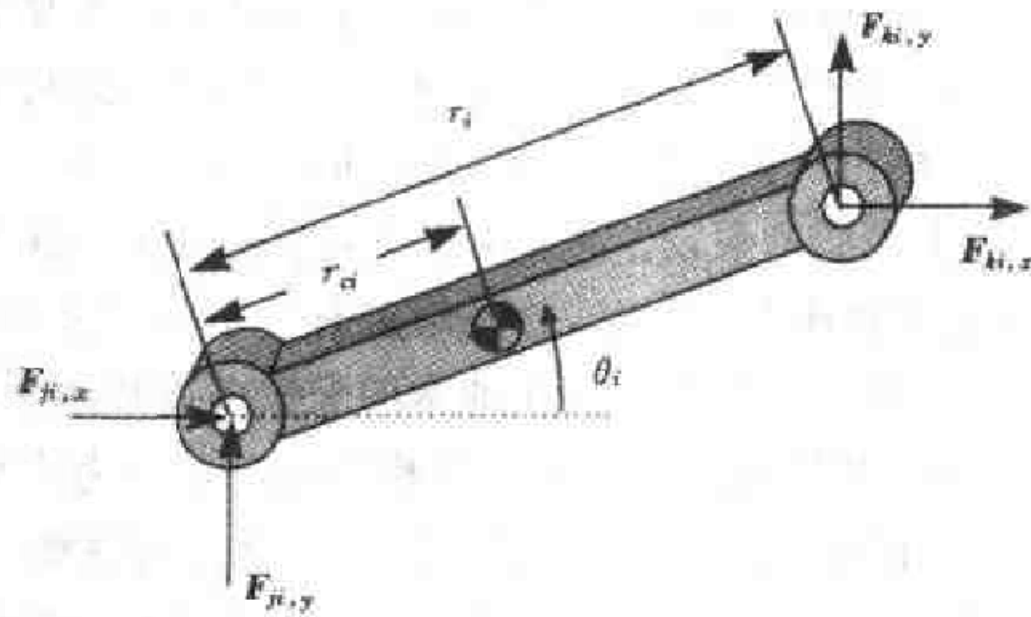


图 6-1 显示其端点所受作用力的典型连接件受力图

这里要说明的关键一点是 F_{ij} 和 F_{ji} 大小相等、方向相反。这一事实隐含在该方法的应用中,以此来减少系统方程的未知数个数。

将牛顿第二定律应用于图 6-1,此连接件的运动方程可以写成如下的形式:

$$F_{j,x} + F_{k,x} = m_i a_{i,x} \quad (6-1)$$

$$F_{j,y} + F_{k,y} = m_i a_{i,y} \quad (6-2)$$

加速度项 $a_{i,x}$ 和 $a_{i,y}$ (原文中加速度用 A 表示) 是后面即将考虑的连接件质心加速度在 x 和 y 方向上的分量。

最后一个动力学方程可以从这个受力图中由力矩之和得到。合力矩与角加速度的关系一般可能比较复杂,但选取恰当的力矩轴可以大大简化二者之间的关系。矩轴最常见的两种选择是选取任意的固定轴或某个通过物体质心 (COM) 的轴。如果两个铰链中其中之一与机架相连,则最好选择通过该铰链的轴为计算力矩之轴,这样可以使力矩表达式较为简单,因为与机架相连的铰链内力不出现在力矩方

程之中。然而,如果没有任何一个铰链与机架相连,则必须用更一般的关系,此时必须选择通过物体质心的轴为矩轴,如图6-2所示。

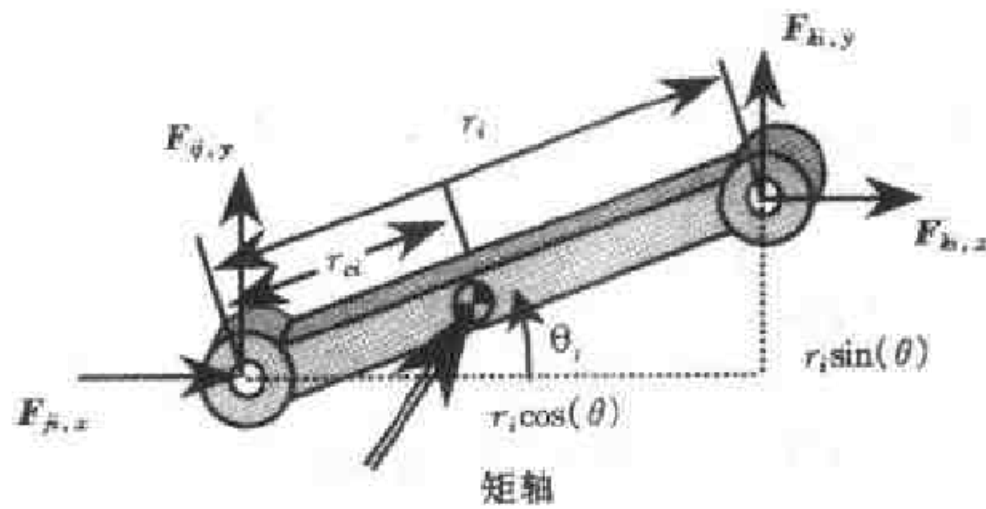


图6-2 力矩方程中各力的力臂

这样,对各个力矩求和即可得到以下方程:

$$F_{q,x} r_a \sin(\theta_i) - F_{q,y} r_a \cos(\theta_i) - F_{k,x} (r_i - r_a) \sin(\theta_i) + F_{k,y} (r_i - r_a) \cos(\theta_i) = I_i \alpha_i \quad (6-3)$$

注意 I_i 是连接杆对质心 c 的转动惯量。通常连接杆相对另外一个轴的转动惯量是已知的。在这种情况下,可以用下面的平行移轴定理来计算相对质心的转动惯量

$$I_{i,\text{new}} = I_i + m_i d^2 \quad (6-4)$$

其中, d 是过质心的轴和转动惯量 I_i 已知的平行轴之间的距离。

6.2.2 闭环矢量方程

在前面几章中已经讨论过闭环矢量方程。因为动态仿真将力和加速度联系起来,所以必须计算闭环矢量方程的二阶导数,可参照第2章最后的表2-2中给出的四连杆机构和曲柄滑块机构的加速度方程。这两个方程表示了由于机构运动约束而产生的连接件加速度之间的固有联系。

6.2.3 质心加速度的矢量方程

图6-3为由两个连接件构成的某个机构的一部分。

在将动力学方程和闭环矢量方程(也被称为约束方程)结合之前,必须提供另外一个关系。注意到在导出一个连接件的运动方程时,引入了质心加速度这个新的运动变量。通常这些加速度并不出现在机构整体的闭环矢量方程中。所以,对每个连接件,必须导出另外一个联系质心加速度和其他运动变量的矢量方程。为了导出两个连接件的质心加速度方程,先写出下面简单的矢量关系

$$\mathbf{a}_{c1} = \ddot{\mathbf{R}}_{c1} \quad (6-5)$$

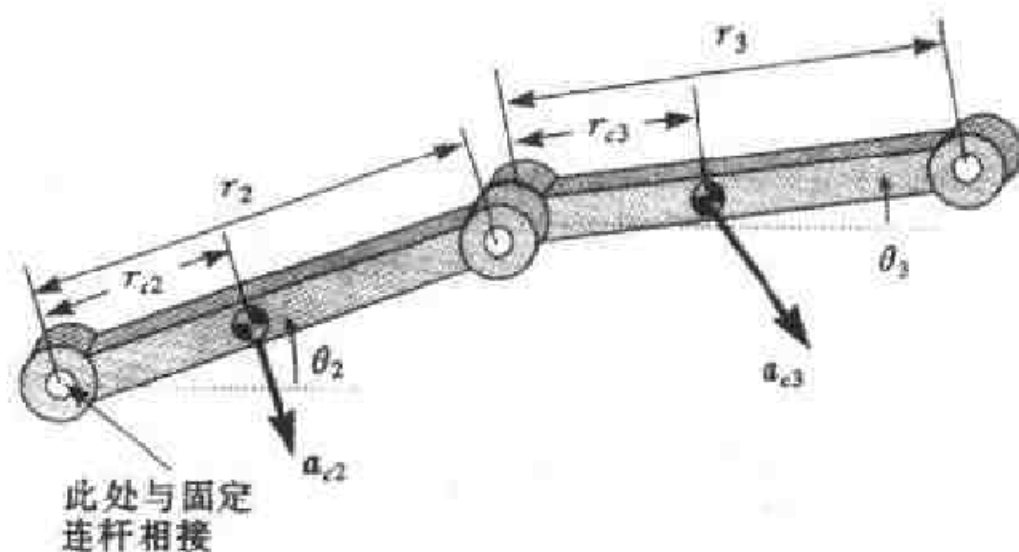


图 6-3 机构局部示意图(说明推导质心加速度的方法)

$$a_{c2} = \ddot{R}_1 + \ddot{R}_{c2} \quad (6-6)$$

和以前讨论的矢量方程一样,这两个方程可以表示成一组 x 和 y 方向的标量方程,即

$$a_{c2,x} = -r_{c2}\alpha_2 \sin\theta_2 - r_{c2}\omega_2^2 \cos\theta_2 \quad (6-7)$$

$$a_{c2,y} = r_{c2}\alpha_2 \cos\theta_2 - r_{c2}\omega_2^2 \sin\theta_2 \quad (6-8)$$

$$a_{c3,x} = -r_2\alpha_2 \sin\theta_2 - r_2\omega_2^2 \cos\theta_2 - r_{c3}\alpha_3 \sin\theta_3 - r_{c3}\omega_3^2 \cos\theta_3 \quad (6-9)$$

$$a_{c3,y} = r_2\alpha_2 \cos\theta_2 - r_2\omega_2^2 \sin\theta_2 + r_{c3}\alpha_3 \cos\theta_3 - r_{c3}\omega_3^2 \sin\theta_3 \quad (6-10)$$

这里作为一个练习留给读者证明。这四个方程与前面的两个方程相互等价。

6.2.4 动态仿真的实现

在前面章节中已经讨论过了对每个连接件应用牛顿-欧拉定理导出的方程,以及对原始的闭环矢量方程求二阶导数获得的方程和通过分析质心加速度而得到的方程,这些方程形成了一组弱耦合的线性形式的力和加速度方程。一般地,对于一个有 n 个连接件的机构,有 $3(n-1)$ 个牛顿-欧拉方程,以及两个闭环矢量方程的标量表达式和 $2(n-1)$ 个质心加速度方程。因此,对于一个四连杆机构,动力学仿真将分析 17 方程,一共 17 个未知数。然而,值得注意的是没有必要采用这种最一般形式的仿真。通常,可以认为质心与某个连接中心重合(例如,若曲柄是某个平衡飞轮的一部分,则质心是静止不动的,其加速度为零)。

下一节将以曲柄滑块机构为例说明这种方法。

6.3 联立约束法在曲柄滑块机构中的应用

为了说明动态仿真的联立约束法,这一节中将给出曲柄滑块机构一个完整的动力学仿真过程。图 6-4 为一个有两个外加力作用的曲柄滑块机构。外加扭矩

M_{12} 作用于曲柄, 而力 F_{ex1} 作用于滑块。在下面几节中, 将分别针对曲柄匀速转动和恒定外力两种情况进行仿真。这样, 曲柄转速 ω_2 被视为输入, 而 M_{12} 变成输出。这样将考虑计算在外力影响下维持曲柄匀速转动所需的扭矩。然后, 再将曲柄转速视为变量, 而扭矩成为仿真系统的输入。

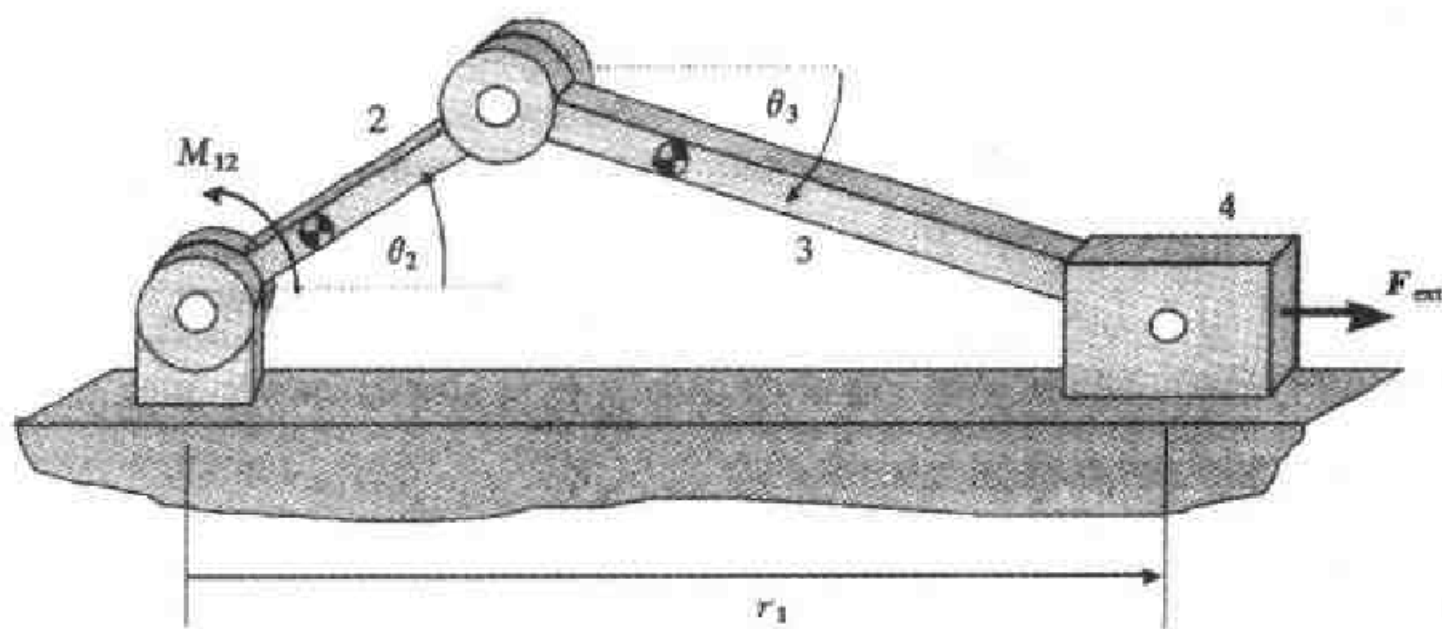


图 6-4 受外加扭矩及力作用的曲柄滑块机构示意图(曲柄匀速转动)

6.3.1 力方程

如前面讨论的那样, 第一步是对每个连接件应用牛顿定律。这一过程可以参阅图 6-5 和 6-6 中给出的受力图。

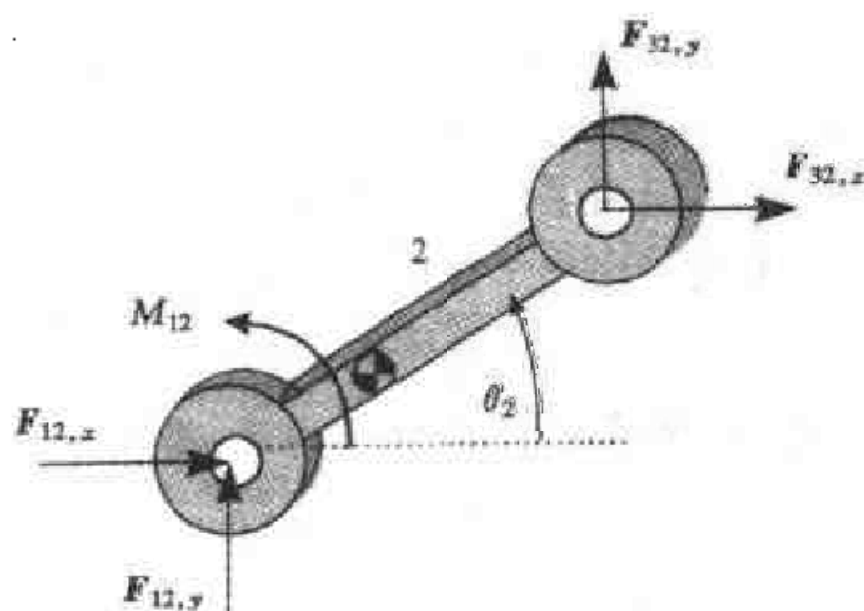


图 6-5 曲柄滑块机构曲柄 2 的受力图

可以很容易地写出杆 2 的运动方程如下

$$F_{12,x} + F_{32,x} = m_2 a_{c2,x} \quad (6-11)$$

$$F_{12,y} + F_{32,y} = m_2 a_{c2,y} \quad (6-12)$$

$$-F_{32,x} r_2 \sin(\theta_2) + F_{32,y} r_2 \cos(\theta_2) + M_{12} = I_2 \alpha_2 \quad (6-13)$$

类似地, 根据连杆 3 的受力图可以写出其运动方程, 但与曲柄 2 的方程有较大差别。注意到前面关于力的符号约定, 因而作用力和反作用力的识别非常容易

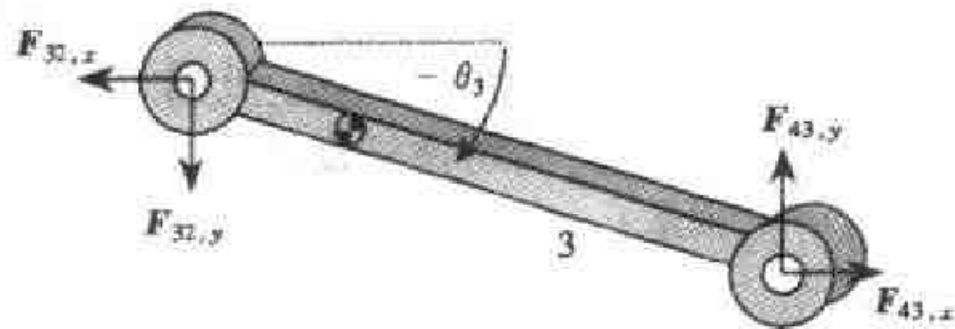


图 6-6 曲柄滑块机构连杆 3 的受力图

(F_{ij} 与 F_{ji} 等值反向), 而且这种关系隐含地表现在图 6-6 及以后的方程中。为了避免引入新的未知力 ($F_{23,x}$ 和 $F_{23,y}$), 将已经引入并画在受力图上的力 ($F_{32,x}$ 和 $F_{32,y}$) 加一负号并用于力和力矩方程中。富于观察力的学生立刻会注意到, 只要利用前面给出的力的符号约定关系, 即可用其他力代替新引入的力, 最终的方程与引入新的力分量所得到的方程完全一样, 即

$$-F_{32,x} + F_{43,x} = m_3 a_{3,x} \quad (6-14)$$

$$-F_{32,y} + F_{43,y} = m_3 a_{3,y} \quad (6-15)$$

$$F_{43,x}(r_3 - r_{c3}) \sin \theta_3 + F_{43,y}(r_3 - r_{c3}) \cos \theta_3 - F_{32,x} r_{c3} \cos \theta_3 + F_{32,y} r_{c3} \sin \theta_3 = I_3 a_3 \quad (6-16)$$

最后, 连接件 4 (即滑块) 上作用的所有力如图 6-7 所示。

注意对滑块 4 没有列力矩方程, 因为在运动分析中滑块被视为一个几何上的点。同时注意到力是滑块所受的摩擦力 $F_{14,x}$ 。目前的分析仅考虑滑块和机架之间无摩擦的情况, 但为了完整性和引导学生进行深入分析, 在图中仍然画出了摩擦力。最后请注意, 滑块和机架之间的支撑面约束限制滑块

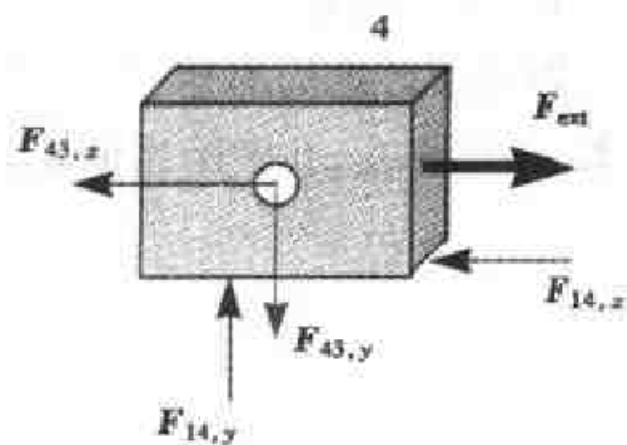


图 6-7 滑块的受力图

只能沿 x 方向运动, 因此 y 方向的加速度分量为零, 滑块 R_4 的运动方程如下

$$F_{34,x} + F_{ext} = m_4 \ddot{r}_1 \quad (6-17)$$

$$F_{34,y} + F_{14,y} = 0 \quad (6-18)$$

现在, 考虑方程 (6-4) ~ (6-18) 中的未知数。经过观察可以发现, 下面这些力是未知量:

$$F_{12,x}, F_{12,y}, F_{32,x}, F_{32,y}, F_{34,x}, F_{34,y}, F_{14,y}, M_{12}$$

另外, 引入如下的加速度:

$$a_{c2,x}, a_{c2,y}, a_{c3,x}, a_{c3,y}, \ddot{r}_1, \alpha_2, \alpha_3$$

最后, 为建立起完整的方程组, 还需从闭环矢量方程导出补充方程。

6.3.2 闭环矢量方程

曲柄滑块机构的闭环矢量方程在前面已经导出,因此为了完整性则再在下面重复一遍,详细的推导过程可以在第4章中找到。

$$\ddot{r}_1 + r_3 \sin\theta_3 \alpha_3 + r_2 \sin\theta_2 \alpha_2 = -r_2 \cos\theta_2 \omega_2^2 - r_3 \cos\theta_3 \omega_3^2 \quad (6-19)$$

$$-r_3 \cos\theta_3 \alpha_3 + r_2 \cos\theta_2 \alpha_2 = r_2 \sin\theta_2 \omega_2^2 - r_3 \sin\theta_3 \omega_3^2 \quad (6-20)$$

6.3.3 质心加速度

方程(6-7)~(6-10)完成了曲柄滑块机构仿真所需的一组方程。其中最后四个方程将机构的运动学状态(连接件位移、速度和加速度)与连接件质心加速度的分量联系起来,而这些质心加速度在力方程中是必须要有的,因此,可得到

$$a_{c2,x} = -r_{c2} \alpha_2 \sin\theta_2 - r_{c2} \omega_2^2 \cos\theta_2 \quad (6-21)$$

$$a_{c2,y} = r_{c2} \alpha_2 \cos\theta_2 - r_{c2} \omega_2^2 \sin\theta_2 \quad (6-22)$$

$$a_{c3,x} = -r_2 \alpha_2 \sin\theta_2 - r_2 \omega_2^2 \cos\theta_2 - r_{c3} \alpha_3 \sin\theta_3 - r_{c3} \omega_3^2 \cos\theta_3 \quad (6-23)$$

$$a_{c3,y} = r_2 \alpha_2 \cos\theta_2 - r_2 \omega_2^2 \sin\theta_2 + r_{c3} \alpha_3 \cos\theta_3 - r_{c3} \omega_3^2 \sin\theta_3 \quad (6-24)$$

6.3.4 系统方程的组装

在进行 Simulink 仿真之前的最后一步是将所有的14个方程组装成矩阵形式。这一过程需要识别输入量,因为只有已知量(系统输入和连接件速度)允许出现在方程的右端。回想本节的开始部分,曾经指出第一个仿真是针对曲柄为匀速旋转的曲柄滑块机构。因此,施加给曲柄的扭矩是未知量并且 α_2 等于零。整个方程组写成矩阵形式如下(见方程(6-25),其中 $C_2 = \cos\theta_2$, $S_2 = \sin\theta_2$, $C_3 = \cos\theta_3$, $S_3 = \sin\theta_3$ ——译者注)。

6.4 曲柄滑块机构的动态仿真

按照和第4章中给出的运动学仿真一样的步骤,可以组装得到机构完整的动力学仿真系统。为了说明这一点,假设输入连接件(曲柄)以100 rad/s的速度匀速旋转。这样需要用五个积分器:一个积分曲柄速度以获得曲柄转角(θ_2);两个积分器积分约束方程求解得到加速度(θ_3 和 \ddot{r}_1);另外两个积分器积分最终的速度来获得位移。

使用一个将所有积分器输出作为自变量的MATLAB函数来求解前一节中导出的矩阵方程(方程(6-25))。此外,允许在滑块上施加外加力。最后,MATLAB函数将进行相容性检验以确保公式推导得正确无误并且积分过程具有足够的精度。

$$\begin{bmatrix}
 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m_2 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m_2 & 0 & 0 \\
 0 & 0 & -r_2 S_2 & r_2 C_2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m_3 & 0 \\
 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m_3 \\
 0 & 0 & -r_{c3} C_3 & r_{c3} S_3 & (r_{c3} - r_3) S_3 & (r_3 - r_{c3}) C_3 & 0 & 0 & 0 & -I_3 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -m_4 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & M & 0 & 0 & 0 & 1 & r_3 S_3 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r_3 C_3 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{c3} S_3 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r_{c3} C_3 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix}
 \begin{bmatrix}
 F_{12,x} \\
 F_{12,y} \\
 F_{32,x} \\
 F_{32,y} \\
 F_{43,x} \\
 F_{43,y} \\
 F_{14,y} \\
 M_{12} \\
 \ddot{r}_1 \\
 \alpha_3 \\
 A_{c2,x} \\
 A_{c2,y} \\
 A_{c3,x} \\
 A_{c3,y}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 -F_{ext} \\
 0 \\
 -r_2 C_2 \omega_2^2 - r_3 C_3 \omega_3^2 \\
 -r_2 S_2 \omega_2^2 - r_3 S_3 \omega_3^2 \\
 -r_{c2} C_2 \omega_2^2 \\
 -r_{c2} S_2 \omega_2^2 \\
 -r_2 C_2 \omega_2^2 - r_{c3} C_3 \omega_3^2 \\
 -r_2 S_2 \omega_2^2 - r_{c3} S_3 \omega_3^2
 \end{bmatrix}
 \tag{6-25}$$

图 6-8 给出了在 Simulink 中一个可能的仿真实现过程。

计算加速度和力的函数在下面的 MATLAB 程序脚本中给出。注意到所有的机构参数定义必须在该函数内部。连接件长度 r_2 和 r_3 是明显的参数,但每个连接件的转动惯量、质量、相对质心的转动惯量以及相对于固接于连接件局部坐标系的质心位置等参数也必须给出定义。在本例中,选取一组典型的单缸除草机上的发动机设计参数。函数的脚本如下:

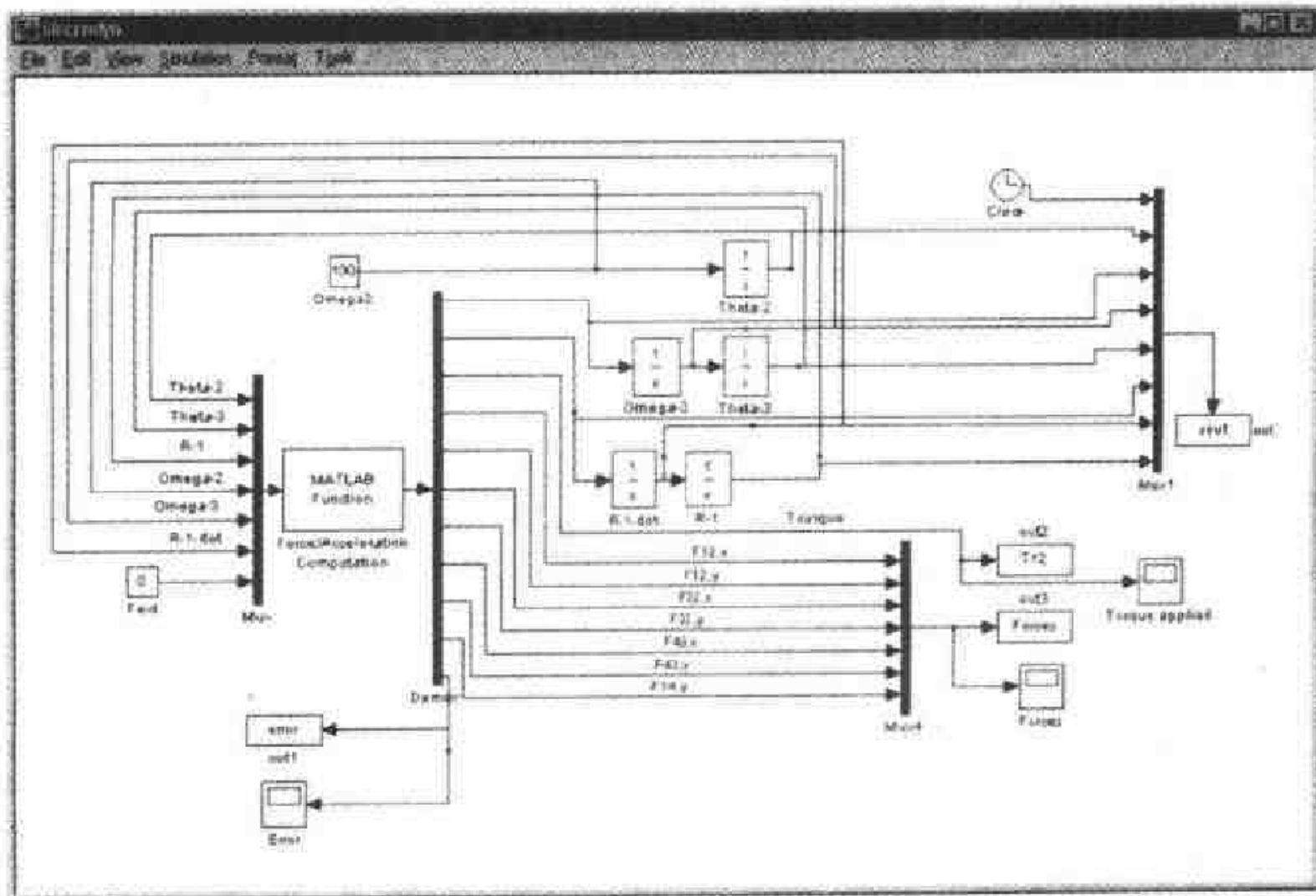


图 6-8 曲柄滑块机构 - 一个完整的 Simulink 动态仿真(文件:slrcrndyn.mdl)

```
slrcrndy.m
function [xout]=slrcrndy(u)
%
% Function to implement the full dynamic simulation
% of a slider crank.
%
% Used with SLRCRNDYN.MDL SIMULINK file
%
%
% BY: J.F. Gardner
%
%
% u(1) = Theta-2
% u(2) = Theta-3
% u(3) = r-1
% u(4) = Omega-2
% u(5) = Omega-3
% u(6) = r-1-dot
% u(7) = F-ext
%
% Define some local variables
%
```

```

r1 = u(3);
r2 = 0.05; % crank length in meters (about 2 inches)
r3 = 0.25; % connecting rod length in meters
rc2 = 0.0; % COM at pivot, implies balanced crank
rc3 = r3/3; % COM about 1/3 of the distance between big end and wrist
pin
C2 = cos(u(1)); S2 = sin(u(1));
C3 = cos(u(2)); S3 = sin(u(2));
w2 = u(4); w3 = u(5);
Fext = u(7);
%
% Define inertial properties
%
M2 = 1.0; % Mass of crank in kg
M3 = 0.2; % Mass of conn. rod in kg
M4 = 0.2; % Mass of piston in kg
I30 = .001; % Mass moment of conn rod about COM in kg-m^2
%
a=zeros(14);
b=zeros(14,1);
%
a(1,1) = 1; a(1,3) = 1; a(1,11) = -M2;
a(2,2) = 1; a(2,4) = 1; a(2,12) = -M2;
a(3,3) = -r2*S2; a(3,4) = r2*C2; a(3,8) = 1;
a(4,3) = -1; a(4,5) = 1; a(4,13) = -M3;
a(5,4) = -1; a(5,6) = 1; a(5,14) = -M3;
a(6,5) = r3*S3; a(6,6) = -r3*C3; a(6,10) = -I30;
a(7,5) = 1; a(7,9) = -M4;
a(8,6) = 1; a(8,7) = 1;
a(9,9) = 1; a(9,10) = r3*S3;
a(10,10) = -r3*C3;
a(11,11) = 1;
a(12,12) = 1;
a(13,10) = rc3*S3; a(13,13) = 1;
a(14,10) = -rc3*C3; a(14,14) = 1;
%
% Set up the RHS vector of equation (5-25)
b(7) = -Fext;
b(9) = -r2*C2*w2^2-r3*C3*w3^2;
b(10) = -r2*S2*w2^2-r3*S3*w3^2;
b(11) = -rc2*C2*w2^2;
b(12) = -rc2*S2*w2^2;
b(13) = -r2*C2*w2^2-rc3*C3*w3^2;
b(14) = -r2*S2*w2^2-rc3*S3*w3^2;
%
% Solve the equations
%
```

```

x = inv(a)*b;
%
% Compute the consistency error
%
error = norm([r1-r2*C2-r3*C3,r2*S2+r3*S3]);
%
% Set up output vector
%
xout(1) = x(10); % Alpha-3
xout(2) = x(9); % r1-double-dot
xout(3) = x(8); % Torque
xout(4) = x(1); % F12x
xout(5) = x(2); % F12y
xout(6) = x(3); % F32x
xout(7) = x(4); % F32y
xout(8) = x(5); % F43x
xout(9) = x(6); % F43y
xout(10) = x(7); % F14y
xout(11) = error; % Consistency Error

```

在仿真运行之前还有一个步骤就是确定五个积分器的初始条件。和运动学仿真一样,相容的初始条件是很关键的。如果选择从 $\theta_2=0$ 开始,那么初始的位置是很简单的($\theta_3=0, r_2=r_2+r_3=0.5$ m)。但速度并不那么简单,它们必须表示当曲柄以 100 rad/s 从零角度开始旋转时整个机构的一个正确可信的解。参考曲柄滑块共线时的速度方程(4-6),可以在命令提示符使用 MATLAB 命令进行如下高精度计算:

```

> r2=0.05;
> r3=0.25;
> om2=100;
> a=[r3*sin(0) 1; -r3*cos(0) 0]
a =
      0    1.0000
 -0.2500    0
> b = [-r2*om2*sin(0); r2*om2*cos(0)];
> vels=inv(a)*b
vels =
    -20
      0

```

这个结果意味着的初始条件是一20 rad/s,而 \dot{r}_1 的初值为 0 (这两个初始条件与初始位置是曲柄滑块机构的上死点这一事实相一致)。

在继续进行之前需要指出的是,计算初始条件的过程可以通过巧妙地选择机构的初始位置而大大简化。在目前的例子中,曲柄滑块机构的上死点(top dead

center, TDC) 是一个非常好的选择, 因为此时曲柄和连接杆(r_2 和 r_3) 都处在零角度位置(假设 x 轴与滑块轴线一致), 并且滑块的速度也是零。因此, 唯一需要计算的值是 ω_3 。尽管使用 MATLAB 求解上面所示情况的速度方程是一个可行的选择, 但富于观察力的工程人员会注意到针对这个特殊状态方程组已经被解耦和简化了。实际上, 可以求出 ω_3 如下封闭形式的解

$$\omega_3 = -\frac{r_2}{r_3}\omega_2$$

6.5 曲柄滑块机构的仿真研究

一个完整的机构动态仿真可以给出远比简单的运动学仿真丰富得多的信息。特别地, 由机构惯性质量的加速度而引起的动态力可以很容易地从动态仿真中计算得到。

让前一节中建立的仿真系统运行 0.1 s 的时间, 这比曲柄旋转整两圈的时间稍短一些。注意到对这种情况只需仿真曲柄旋转一整圈的过程, 是因为仿真是在曲柄匀速旋转的假设之上建立和运行的。认识到这一点对仿真是很重要的, 因为在一个周期之后得到的仿真结果和前面的结果没有任何区别。

图 6-9 给出了为维持曲柄匀速旋转所必须提供的扭矩。有必要对该图作一些进一步的解释。仿真假设曲柄以匀角速度 100 rad/s (大约 950 r/min) 旋转, 但并未提及使曲柄匀速旋转的设备。注意到扭矩是正负交替的, 这意味着在曲柄旋转一圈的过程中必须向机构输入和消除能量。这与我们在建模中没有考虑外加力以及消耗能量的内部摩擦力这一事实是一致的。因此, 在机构运行过程中净能量

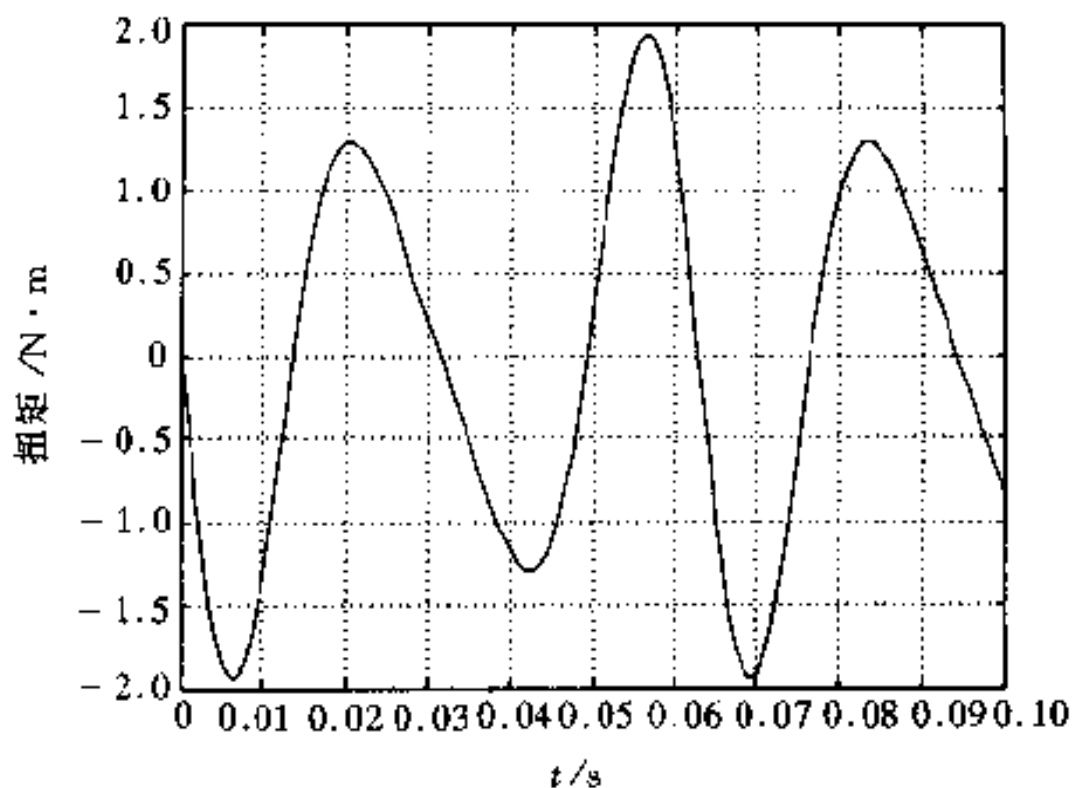


图 6-9 维持曲柄以匀速 100 rad/s 旋转所需的扭矩

传输的平均值必须为零以此满足能量守恒。

在实际应用中,对曲柄匀速转动的假设必须仔细斟酌。如果输入元件具有较大的转动惯量,并且原动机能够提供的扭矩比维持机构运转的要求要大,那么这个假设通常是合理的。然而,我们经常需要将动态仿真模型扩展到包括驱动设备(电动机)在内的简单模型,此时曲柄的转速波动必须予以考虑。

除了输入的扭矩之外,仿真还计算机构内的约束反力。例如 $F_{12,x}$ 和 $F_{12,y}$ 是作用于曲柄和机架连接处支座的 x 和 y 方向的约束反力。如果要对该机构的支座进行设计,这个力是要考虑的重要参数。实际上,因为该机构上没有外加力作用,动态仿真确定的动态力只有支座反力。

图 6-10 是在同一仿真过程中支座反力 x 和 y 方向的分量随时间变化的关系图。注意到这张图很难给出解释,因为通过观察并不清楚支座反力合力的大小随时间的变化规律。但将两个正交分力合成而得到合力的大小和方向是很容易的,用 Simulink 很容易实现这一点。

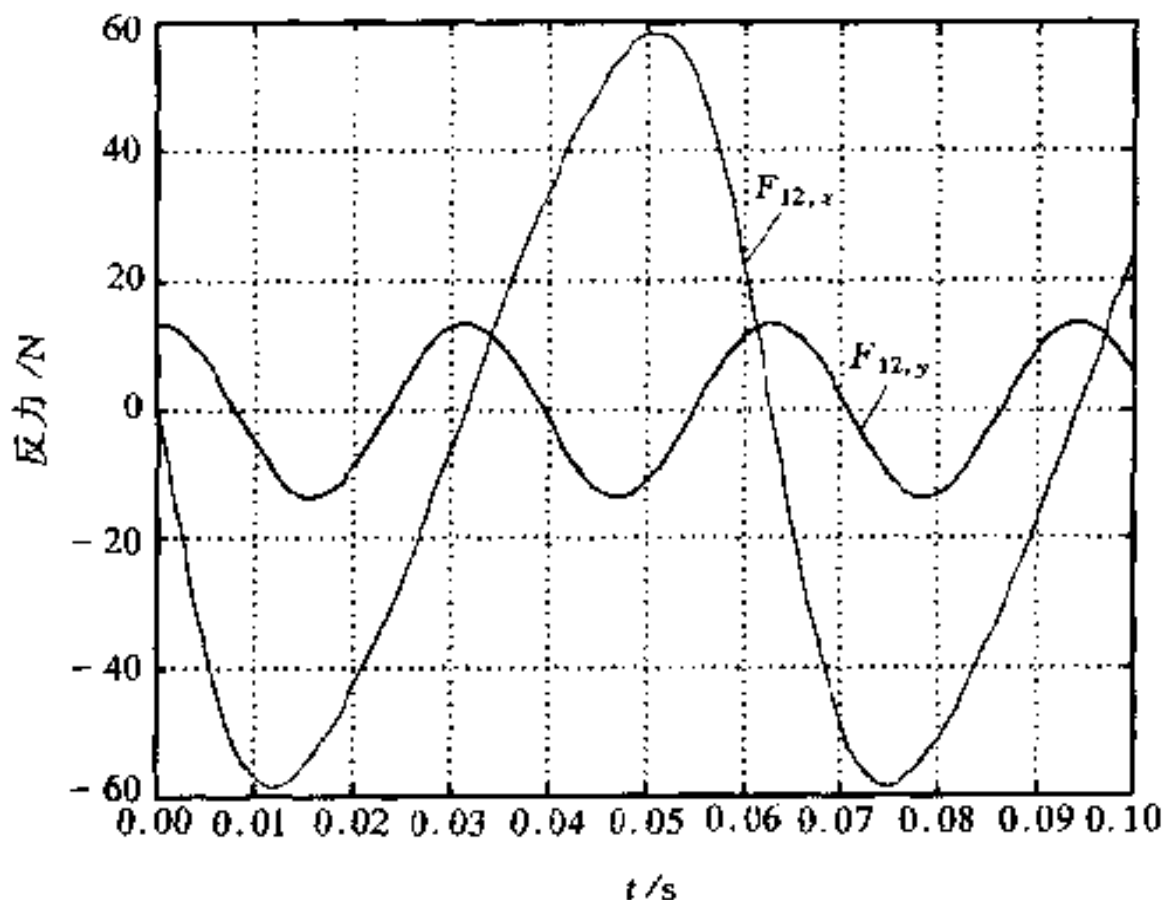


图 6-10 曲柄支撑处 x 和 y 方向的地面反力

图 6-11 显示了能够完成力的合成任务的 Simulink 模块图的一部分。注意 x 和 y 方向的分力可以认为与直角坐标类似,而所需合力的大小和方向是相应的极坐标。Simulink 已经预先定义了可以用来完成这种坐标变换的模块。图 6-11 显示了 Simulink 模块的一部分,该模块可以被进一步扩展以包括坐标变换模块。在 x 和 y 方向的分力已知的情况下,这些模块的应用在于将直角坐标变换为极坐标,以便很好地适应合力大小和方向的计算任务。这个模块在“Simulink Extras”库内,模块名称是“Cartesian to Polar”。

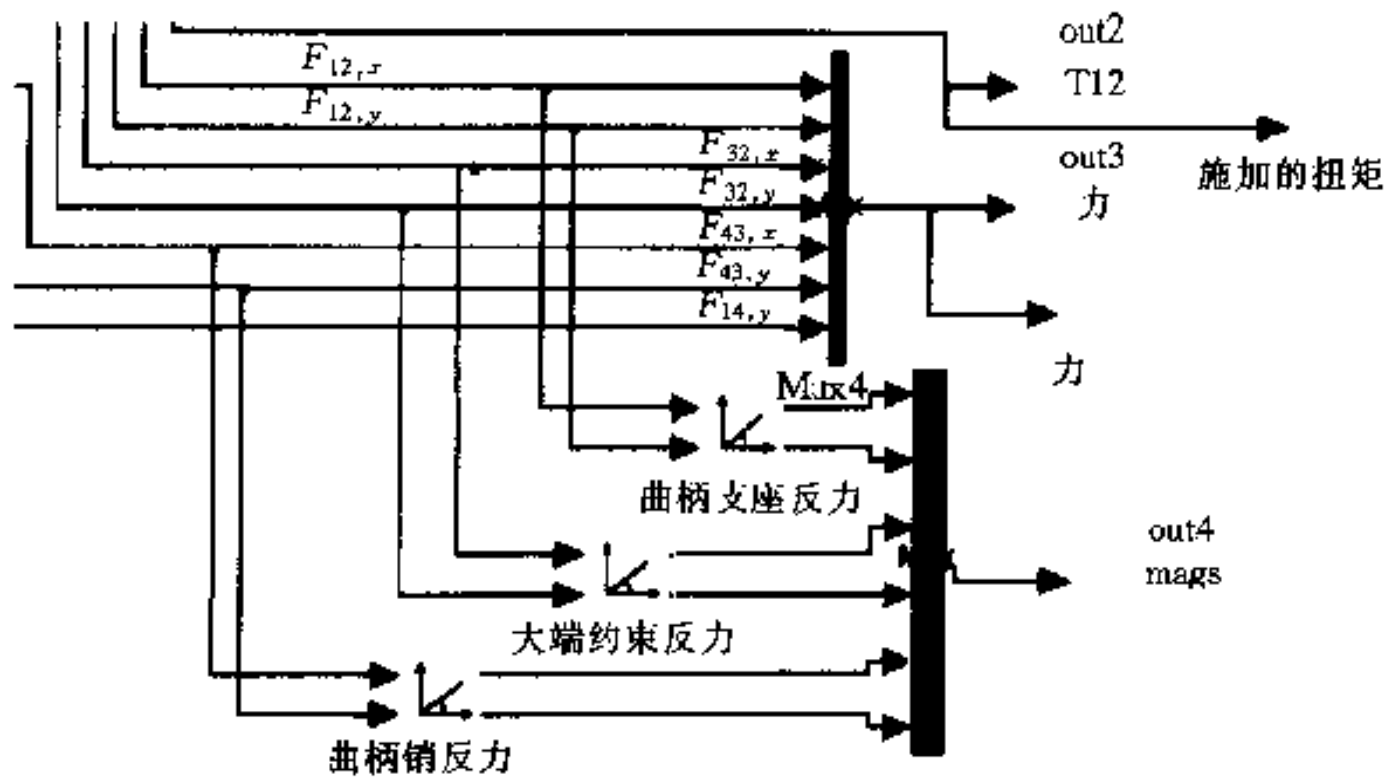


图 6-11 包含将平面坐标变换成极坐标的部分 Simulink 模块

图 6-12 画出了力 F_{12} , F_{23} 和 F_{34} 随时间的变化规律, 图 6-13 给出了以 x 正方向为基准的各个力的方向变化规律。

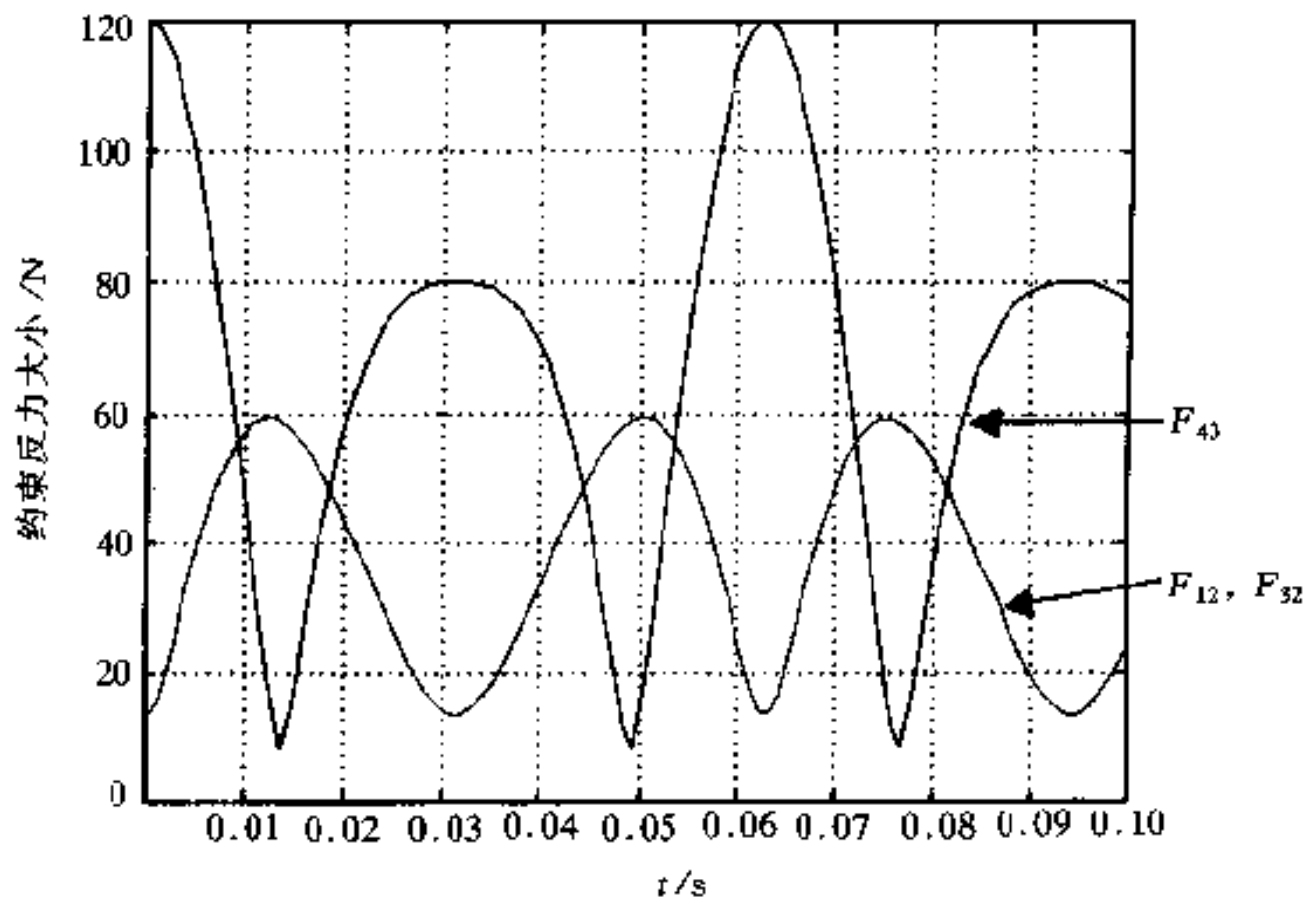


图 6-12 曲柄滑块机构的曲柄支座反力、连杆与曲柄铰接处反力以及滑块销钉处反力的大小随时间变化的关系

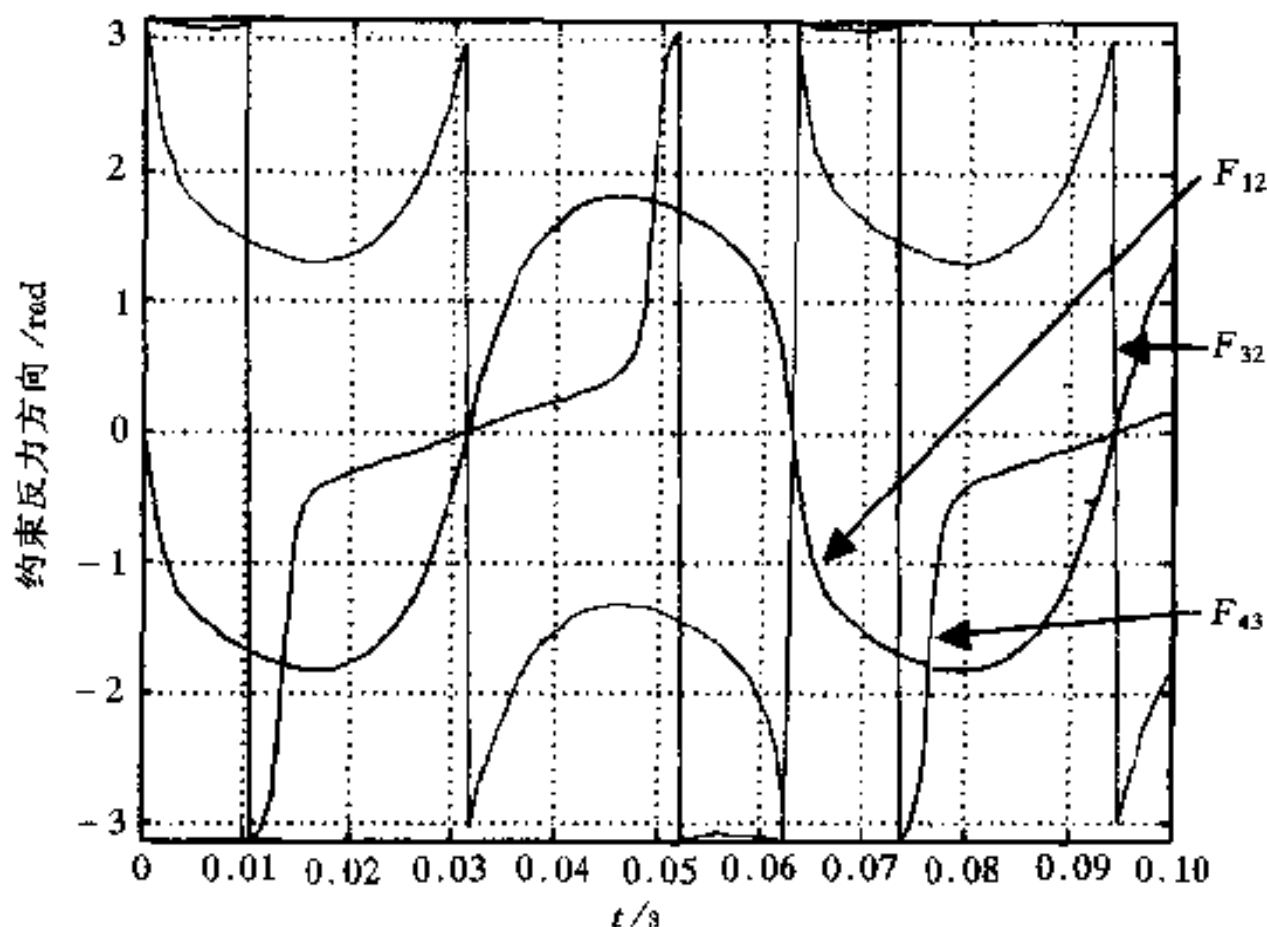


图 6-13 曲柄滑块机构上述三个反力的方向随时间变化的关系

6.6 小结

在本章中,给出了进行约束系统动态仿真的有效工具——联立约束法。动力学仿真与动力学分析相比最主要的优点是对系统的运动无需任何假设。例如,绝大部分的机构动力学分析是基于某个连接件匀速旋转的假设。实际上,作为原动机的系统(例如电动机)和机构的惯性载荷之间总存在着相互作用,当所需的扭矩增大时,电动机的转速将会降低。能够观察到这一结果的唯一途径是采用在本章中论述的动态仿真。

在后面的几章中,将研究几种不同功能的约束系统并将详细讨论两个例子。

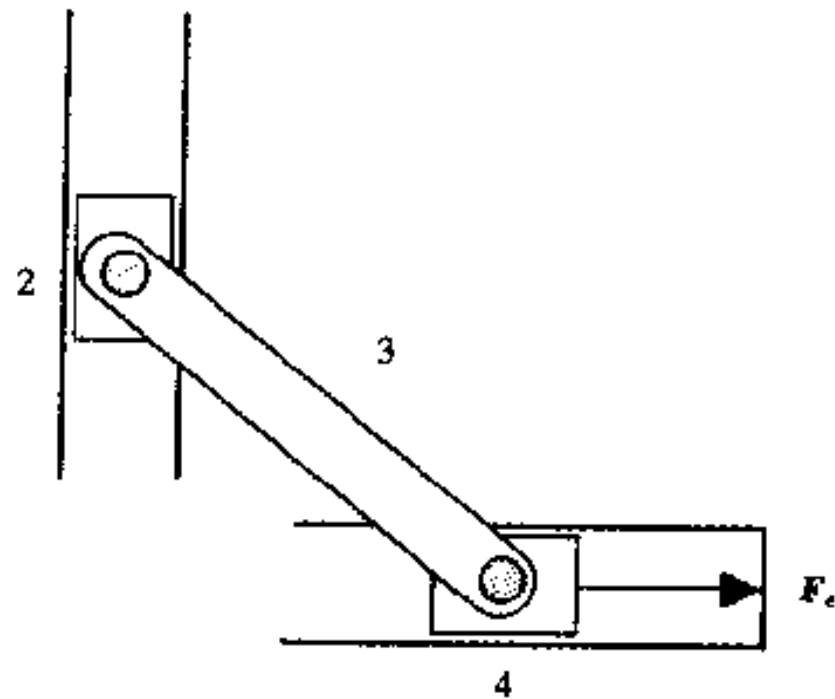
第6章习题

1. 考虑如下所示的双滑块机构

建立以题图1中所示的力 F_e 为输入的动态仿真。使用下表列出的参数,画出若 F_e 以 $10 \sin(5t)$ 的规律变化时滑块处正反力的变化规律。

表1 题1的参数值

参数	值	参数	值
r_3/m	1.0	m_3/kg	1.2
m_2/kg	0.8	m_4/kg	1.5
$I_3/\text{kg} \cdot \text{m}^2$	0.01		

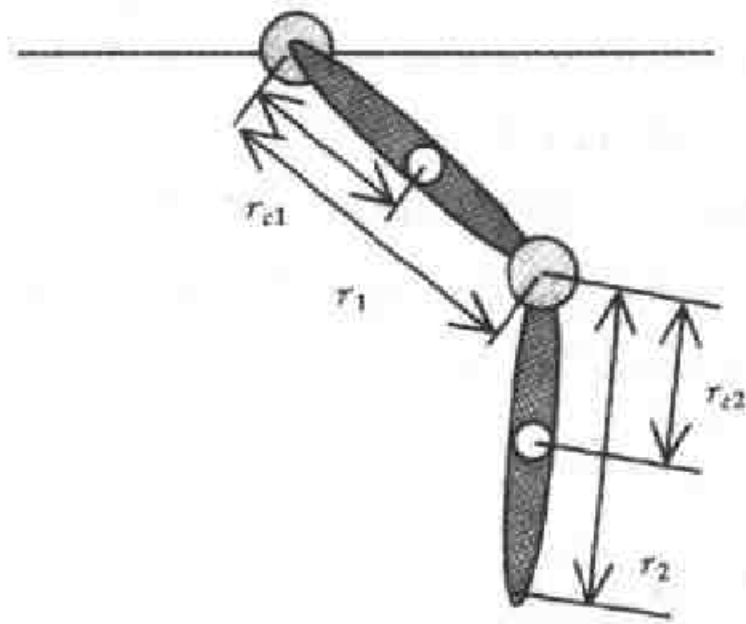


题图 1

2. 复摆由两个或两个以上的摆串连接构成(见题图 2)。复摆是一种非常有趣的机构,它可以表现出混沌行为。在动力学系统中,混沌系统是对初值极端敏感的系统。两组互相非常接近的初值条件有可能导致完全不同的系统行为。采用下表的参数值建立复摆的动态仿真。

表 2 题 2 的参数值

参数	值
r_1/cm	10
r_{c1}/cm	4
r_2/cm	8
r_{c2}/cm	5
m_1/kg	0.1
$I_1/\text{kg} \cdot \text{m}^2$	0.001
m_2/kg	0.08
$I_2/\text{kg} \cdot \text{m}^2$	0.000 6



题图 2

通过仿真,研究系统响应对初始条件的微小变化的灵敏度。

3. 修改在本章内容中给出的曲柄滑块机构仿真以便适应偏置曲柄滑块机构。采用本章给出的惯性和几何参数,研究偏置对滑块正反力的影响。画出当曲柄以 1 800 r/min 的速度匀速旋转时滑块最大正反力随偏置量的变化图,偏置量从 -0.02 m 以 0.005 m 的增量变化至 0.02 m 。

第7章 两连杆平面机器人

7.1 概述

本章将讨论一种在机器人学文献中经常遇到的开链式机构的动力学仿真。平面两连杆机器人是一种简单的两自由度的机械装置,其具有一定的复杂动力特性,图7-1表明了这种装置的相关几何参数。

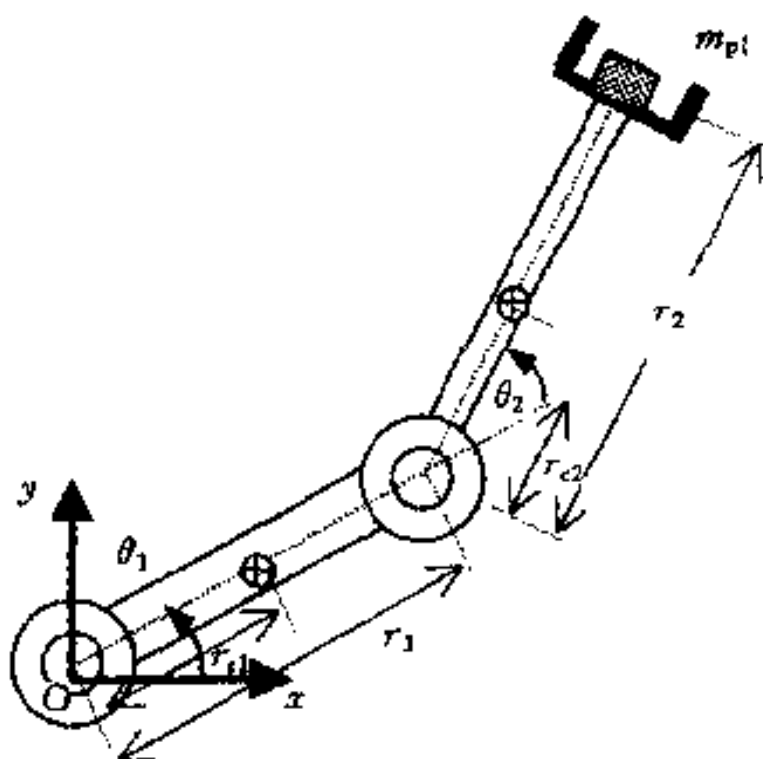


图7-1 两连杆机器人的机构图

本章将导出这种机构的矢量方程和动力学方程。最终的动力学仿真要求具有两个输入参数,即由两个电动机所产生的扭矩。

7.2 矢量方程

显而易见,平面两连杆机器人的矢量方程为

$$\mathbf{R}_p = \mathbf{R}_1 + \mathbf{R}_2 \quad (7-1)$$

注意到方程(7-1)和我们迄今见过的方程形式有所不同,原因在于矢量角定义的方法有所不同。在图7-1中,所给出的角度是相对于前一连杆的方位,而不是相

对于整体坐标系的 x 轴。这种习惯表示法在机器人学中很普遍,其源于安装在机器人手臂上的传感器所测得的是连杆的相对转角,而不是绝对转角。

方程(7-1)中的 R_{pl} 相应于 x 和 y 的标量方程如下所示

$$x_{pl} = r_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2) \quad (7-2)$$

$$y_{pl} = r_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2) \quad (7-3)$$

对上式求导数有

$$\dot{x}_{pl} = -r_1 \omega_1 \sin\theta_1 - r_2 (\omega_1 + \omega_2) \sin(\theta_1 + \theta_2) \quad (7-4)$$

$$\dot{y}_{pl} = r_1 \omega_1 \cos\theta_1 + r_2 (\omega_1 + \omega_2) \cos(\theta_1 + \theta_2) \quad (7-5)$$

将上两式重新整理并写成如下矩阵形式(译者注: $C_{12} = \cos(\theta_1 + \theta_2)$, $S_{12} = \sin(\theta_1 + \theta_2)$)

$$\begin{bmatrix} \dot{x}_{pl} \\ \dot{y}_{pl} \end{bmatrix} = \begin{bmatrix} -r_1 S_1 - r_2 S_{12} & -r_2 S_{12} \\ r_1 C_1 + r_2 C_{12} & r_2 C_{12} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \quad (7-6)$$

在笛卡尔坐标系下这种节点角速度和根部操纵装置速度之间的转换关系在机器人学中是众所周知的,这个矩阵就是人们所熟知的雅可比矩阵,它是一种有效控制算法的基础。对方程(7-6)再次求导,可以推出相应的加速度方程为

$$\ddot{x}_{pl} + (r_1 S_1 + r_2 S_{12})\alpha_1 + r_2 S_{12}\alpha_2 = -[(r_1 C_1 + r_2 C_{12})\omega_1^2 + r_2 C_{12}\omega_2^2 + 2r_2 \omega_1 \omega_2 C_{12}] \quad (7-7)$$

$$\ddot{y}_{pl} - (r_1 C_1 + r_2 C_{12})\alpha_1 - r_2 C_{12}\alpha_2 = -[(r_1 S_1 + r_2 S_{12})\omega_1^2 + r_2 S_{12}\omega_2^2 + 2r_2 \omega_1 \omega_2 S_{12}] \quad (7-8)$$

上述推导出的方程构成了进行动力学仿真的基础,它们表明了有效负荷的加速度与两节点处电动机的角速度和角加速度之间的关系。如同在前面章节中的动力学仿真所看到的那样,确定两连杆质心处的加速度和节点变量之间的关系是十分重要的,而这些关系可推导归纳如下

$$a_{d,x} + r_d S_1 \alpha_1 = -r_d C_1 \omega_1^2 \quad (7-9)$$

$$a_{d,y} - r_d C_1 \alpha_1 = -r_d S_1 \omega_1^2 \quad (7-10)$$

$$a_{c,x} + (r_1 S_1 + r_2 S_{12})\alpha_1 + r_2 S_{12}\alpha_2 = -[(r_1 C_1 + r_2 C_{12})\omega_1^2 + r_2 C_{12}\omega_2^2 + 2r_2 \omega_1 \omega_2 C_{12}] \quad (7-11)$$

$$a_{c,y} + (r_1 C_1 + r_2 C_{12})\alpha_1 - r_2 C_{12}\alpha_2 = -[(r_1 S_1 + r_2 S_{12})\omega_1^2 + r_2 S_{12}\omega_2^2 + 2r_2 \omega_1 \omega_2 S_{12}] \quad (7-12)$$

7.3 动力学方程

按照第6章所讲的方法,分别取两个连杆的隔离体进行分析。图7-2为平面机器人的第一根连杆的受力图。

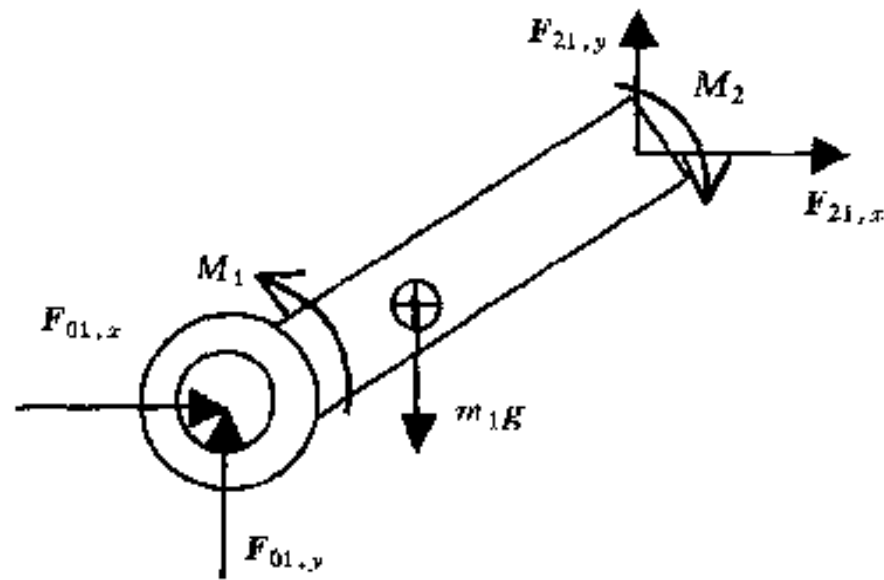


图 7-2 图 7-1 所示的两连杆平面机器人的第一根连杆受力图

从这根连杆可以推导出下面的三个运动方程

$$F_{01,x} + F_{21,x} = m_1 a_{c1,x} \quad (7-13)$$

$$F_{01,y} + F_{21,y} - m_1 g = m_1 a_{c1,y} \quad (7-14)$$

$$M_1 - M_2 - F_{21,x} r_1 S_1 + F_{21,y} r_1 C_1 - m_1 g r_c C_1 = I_1 \alpha_1 \quad (7-15)$$

同样,第二根连杆的受力图如图 7-3 所示。

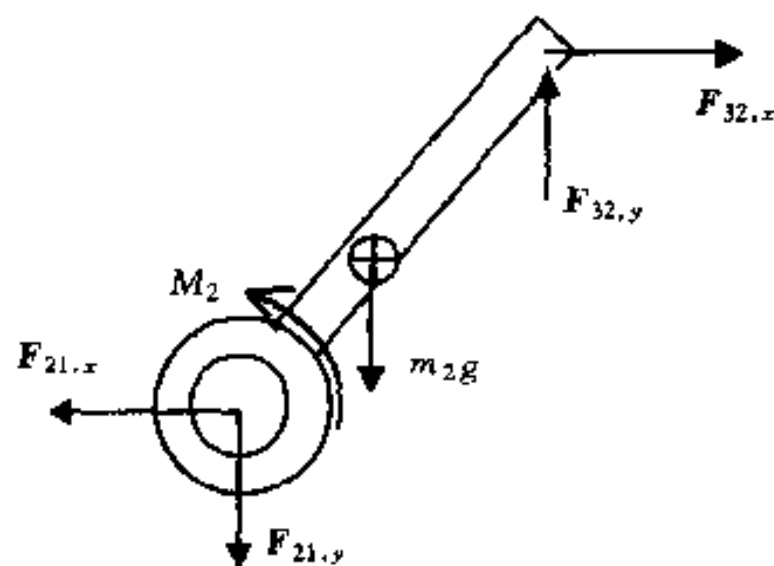


图 7-3 图 7-1 所示两连杆平面机器人的第二根连杆受力图

第二根连杆的三个运动方程如下

$$F_{32,x} - F_{21,x} = m_2 a_{c2,x} \quad (7-16)$$

$$F_{32,y} - F_{21,y} - m_2 g = m_2 a_{c2,y} \quad (7-17)$$

$$M_2 - F_{21,x} r_{c2} S_{12} + F_{21,y} r_{c2} C_{12} - F_{32,x} (r_2 - r_{c2}) S_{12} + F_{32,y} (r_2 - r_{c2}) C_{12} = I_2 \alpha_2 \quad (7-18)$$

最后,研究作用在机器人上的有效载荷。注意到有效载荷的变化是直接与第

二个连杆相联系的,所以可以将其与这个连杆集中在一起考虑,如图 7-3 所示。但是,由于有效载荷的质量是随机器人举起的不同物体而变化的,并且我们所关心的是维持有效载荷所需力的大小,因而再增加几个运动方程对研究问题会更加方便,那样就可以分别跟踪连杆的运动情况。

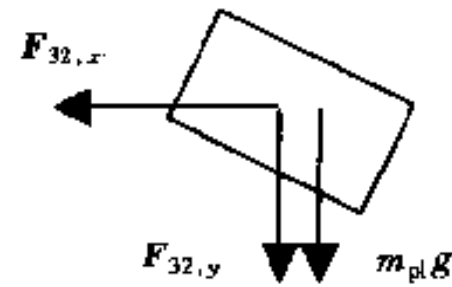
将加载物视为集中质量,可以得到以下两个方程

$$m_{pl}\ddot{x}_{pl} = -F_{32,x} \quad (7-19) \quad \text{图 7-4 加载物受力分析图}$$

$$m_{pl}\ddot{y}_{pl} = -F_{32,y} - m_{pl}g \quad (7-20)$$

综上所述,从矢量方程中得到了六个标量方程,从动力学方程中得到了八个标量方程,如果再将电动机的扭矩考虑进来,就有了下面的 14 个未知量

$$\alpha_1, \alpha_2, a_{c1,x}, a_{c1,y}, a_{c2,x}, a_{c2,y}, a_{pl,x}, a_{pl,y}, F_{01,x}, F_{01,y}, F_{21,x}, F_{21,y}, F_{32,x}, F_{32,y}$$



7.4 联立约束矩阵

联立六个运动约束方程和八个动力学方程,即可得到一个关于 14 个未知量的线性方程组,如下所示:

$$\begin{bmatrix} r_1 s_1 + r_2 s_{12} & r_2 s_{12} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -r_1 c_1 - r_2 c_{12} & -r_2 c_{12} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_1 s_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -r_1 c_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_1 s_1 + r_2 s_{12} & r_2 s_{12} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -r_1 c_1 - r_2 c_{12} & -r_2 c_{12} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -m_1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ I_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_1 s_1 & -r_1 c_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -m_2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -m_2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & I_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 s_{12} & -r_2 c_{12} & (r_2 - r_{c2}) s_{12} & -(r_2 - r_{c2}) c_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & m_{pl} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{pl} & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ A_{c1x} \\ A_{c1y} \\ A_{c2x} \\ A_{c2y} \\ A_{plx} \\ A_{ply} \\ F_{01x} \\ F_{01y} \\ F_{21x} \\ F_{21y} \\ F_{32x} \\ F_{32y} \end{bmatrix}$$

$$= \begin{bmatrix} -[(r_1 C_1 + r_2 C_{12})\omega_1^2 + r_2 C_{12}\omega_2^2 + 2r_2\omega_1\omega_2 C_{12}] \\ -[(r_1 S_1 + r_2 S_{12})\omega_1^2 + r_2 S_{12}\omega_2^2 + 2r_2\omega_1\omega_2 S_{12}] \\ -r_{c1}C_1\omega_1^2 \\ -r_{c1}S_1\omega_1^2 \\ -[(r_1 C_1 + r_{c2}C_{12})\omega_1^2 + r_{c2}C_{12}\omega_2^2 + 2r_{c2}\omega_1\omega_2 C_{12}] \\ -[(r_1 S_1 + r_{c2}S_{12})\omega_1^2 + r_{c2}S_{12}\omega_2^2 + 2r_{c2}\omega_1\omega_2 S_{12}] \\ 0 \\ m_1 g \\ \tau_1 - \tau_2 - m_1 g r_{c1} C_1 \\ 0 \\ m_2 g \\ \tau_2 \\ 0 \\ -m_{pl} g \end{bmatrix} \quad (7-21)$$

这个矩阵方程嵌入到 MATLAB 函数中,随后也可以嵌入到 Simulink 仿真中。

7.5 动态仿真

Simulink 仿真模型按照第 6 章中所描述的方法集合而成,图 7-5 是一个合理

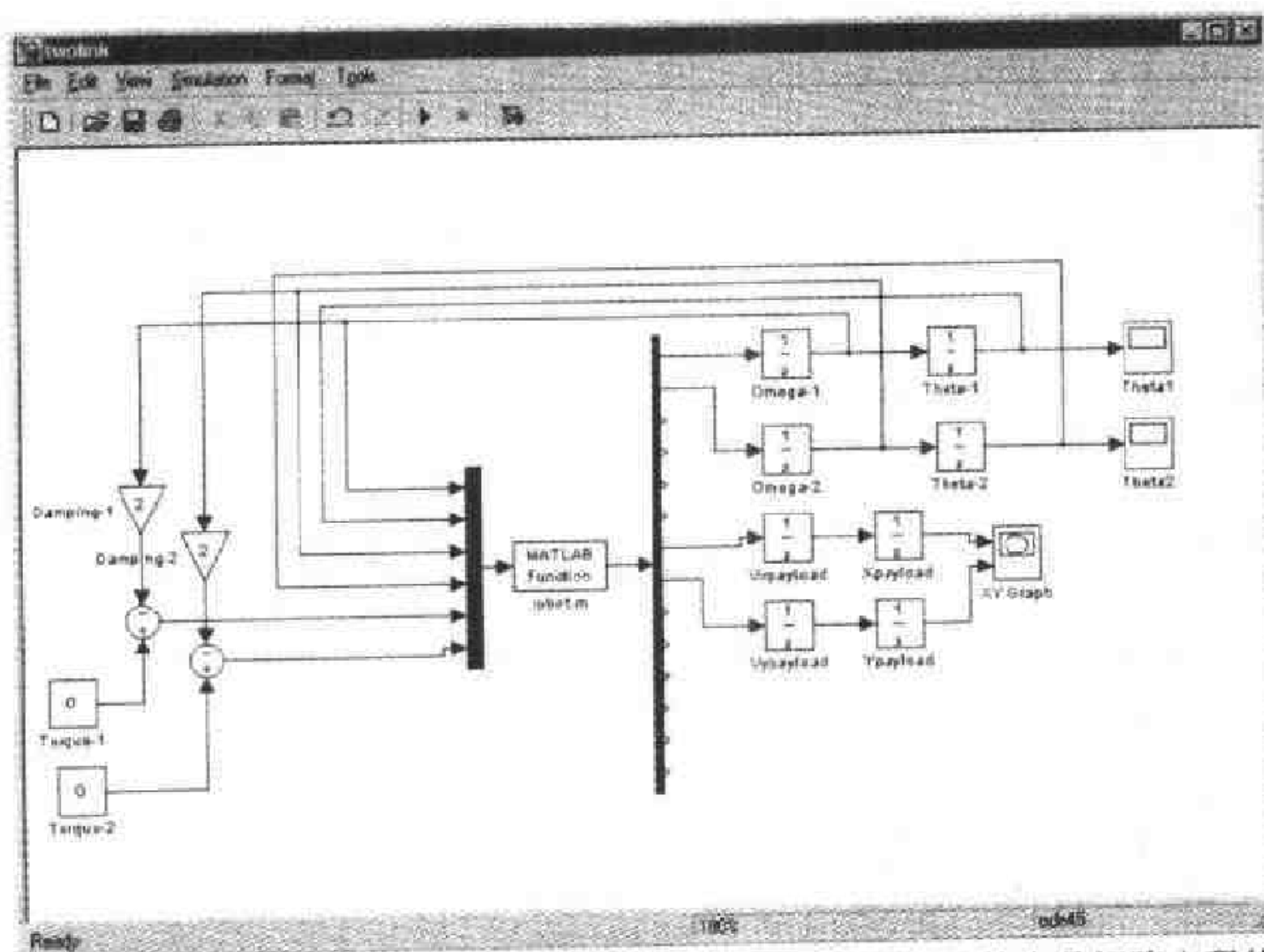


图 7-5 两连杆机器人的 Simulink 仿真模型(DeMux 模块的输出是矩阵方程的 14 个未知数,未连接的输出代表冗余的加速度和约束力(文件: twolink.mdl))

的 Simulink 模型原理框图。值得注意的是,到两个电动机的扭矩被输入到了仿真系统之中,而且轴承摩擦力的简单模型也加了进去。一般说来,摩擦是阻止运动并且和速度有关的力或者力矩。在图 7-5 中,采用了一种线性摩擦力模型。在该模型中,抵抗输入扭矩的摩擦力矩与转速成正比。两个标“Damping”的增益模块表示的是由于轴承和电动机的粘滞阻尼而产生的速度的损失,这些系数的实际值是很难确定的。但是在操纵机器人装置时,某些类型的能量消耗是客观存在的,在模型中缺少对能量消耗的考虑将会导致较大的误差和得到不符合实际的仿真结果。

求解这个 14×14 阶矩阵方程的函数程序如下所示:

```
robot.m
function out=robot(u)
%
% u(1) = omega-1
% u(2) = theta-1
% u(3) = omega-2
% u(4) = theta-2
% u(5) = Torque-1
% u(6) = Torque-2
%
%
g=9.8067;
%
r1 = 1.0; rc1 = 0.5;
r2 = 0.8; rc2 = 0.1;
m1 = 2.5; m2 = 1.8;
I1 = 0.15; I2 = 0.05;
%
mpt=2.0;
%
S1 = sin(u(2)); S12 = sin(u(2)+u(4));
C1 = cos(u(2)); C12 = cos(u(2)+u(4));
%
a=zeros(14,14);
b=zeros(14,1);
%
a(1,1) = r1*S1+r2*S12; a(1,2)=r2*S12; a(1,7) = 1;
a(2,1) = -r1*C1-r2*C12; a(2,2)=-r2*C12; a(2,8) = 1;
a(3,1)=rc1*S1; a(3,3)=1;
a(4,1)=-rc1*C1;a(4,4)=1;
a(5,1)=r1*S1+rc2*S12;a(5,2)=rc2*S12;a(5,5)=1;
a(6,1)=-r1*C1-rc2*C12;a(6,2)=-rc2*C12;a(6,6)=1;
a(7,3)=-m1;a(7,9)=1;a(7,11)=1;
a(8,4)=-m1;a(8,10)=1;a(8,12)=1;
a(9,1)=I1;a(9,11)=r1*S1;a(9,12)=-r1*C1;
```

```

a(10,5)=-m2;a(10,11)=-1;a(10,13)=1;
a(11,6)=-m2;a(11,12)=-1;a(11,14)=1;
a(12,2)=I2;a(12,11)=rc2*S12;a(12,12)=-rc2*C12;a(12,13)=(r2-rc2)*S12;a(12,14)=-rc2*C12;
a(13,7)=m*pl;a(13,13)=1;
a(14,8)=m*pl;a(14,14)=1;
%
%
b(1) = -((r1*C1+r2*C12)*u(1)^2+r2*C12*u(3)^2+2*r2*u(1)*u(3)*C12);
b(2) = -((r1*S1+r2*S12)*u(1)^2+r2*S12*u(3)^2+2*r2*u(1)*u(3)*S12);
b(3) = -rc1*C1*u(1)^2;
b(4) = -rc1*S1*u(1)^2;
b(5) = -((r1*C1+rc2*C12)*u(1)^2+rc2*C12*u(3)^2+2*rc2*u(1)*u(3)*C12);
b(6) = -((r1*S1+rc2*S12)*u(1)^2+rc2*S12*u(3)^2+2*rc2*u(1)*u(3)*S12);
b(8) = m1*g;
b(9) = u(5)-u(6)-m1*g*rc1*C1;
b(11)= m2*g;
b(12)= u(6);
b(14)= -m*pl*g;
%
out=inv(a)*b;

```

为了在仿真中获得较高的可信度,需要做一个简单的实验。回想一下机器人在垂直平面工作时在重力作用下的受力图。因此,如果让机器人从任何初始位置开始运动,将输入的扭矩值设置为零,那么机器人将在自身重力的作用下下落,最后到达两个连杆都在一条铅垂线上的位置。参照图 7-1 中节点角的定义,此时相应的节点角度是 $\theta_1 = -\pi/2$ rad 和 $\theta_2 = 0$ rad。

对于初始条件,选择 $\theta_1 = 0$ 和 $\theta_2 = \pi/2$ rad。这对应于加载物的位置(就是机器人操纵装置的末点位置) $x_{pl} = 1.0$ 和 $y_{pl} = 0.8$ 。如同所有的仿真一样,积分求解器的初始条件必须是相容的。

图 7-6 给出了仿真的结果,其上附加了操纵装置在初始和终了条件下的机构位置略图。数据点表明了加载物体的位置坐标随时间变换的规律。请注意,曲线图中显示的运动轨迹与我们所想到的让机器人在自重作用下下降的运动情形相一致。

最后,为了进一步增加仿真的可信度,增加了一个相容性检验。如果在仿真中所有的工作都顺利进行,那么仿真计算出的冗余信息是相容的。特别地,通过积分求解器计算出的加载物体的 x 和 y 的坐标值应该与通过将 θ_1 和 θ_2 的数值代入方程(7-2)和(7-3)求得的值相等。这四个积分求解器很容易计算出输出值,可以将它们嵌入到一个 Mux 块中,写一个程序来计算误差信息。图 7-7 显示的是 Scope 块监测机器人在自重作用下下落了 10 s 的仿真过程中的计算误差。

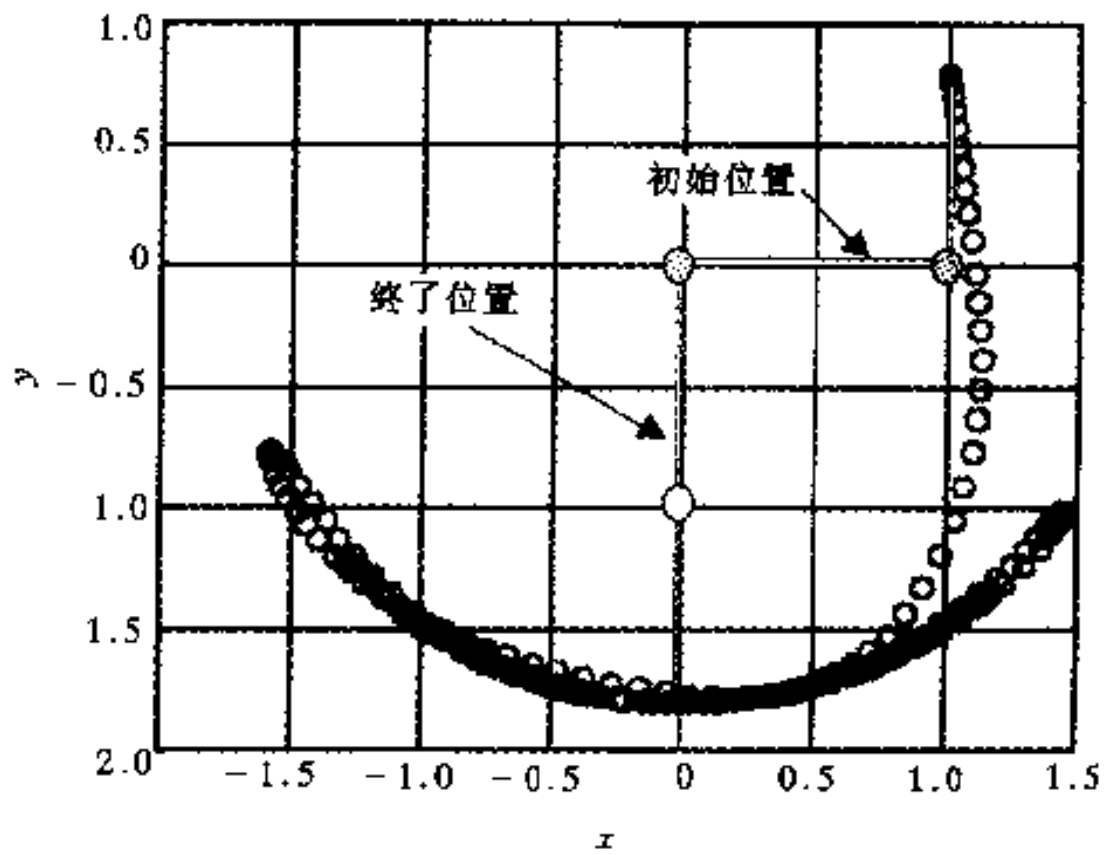


图 7-6 机器人仅在自重作用下的下落过程仿真曲线

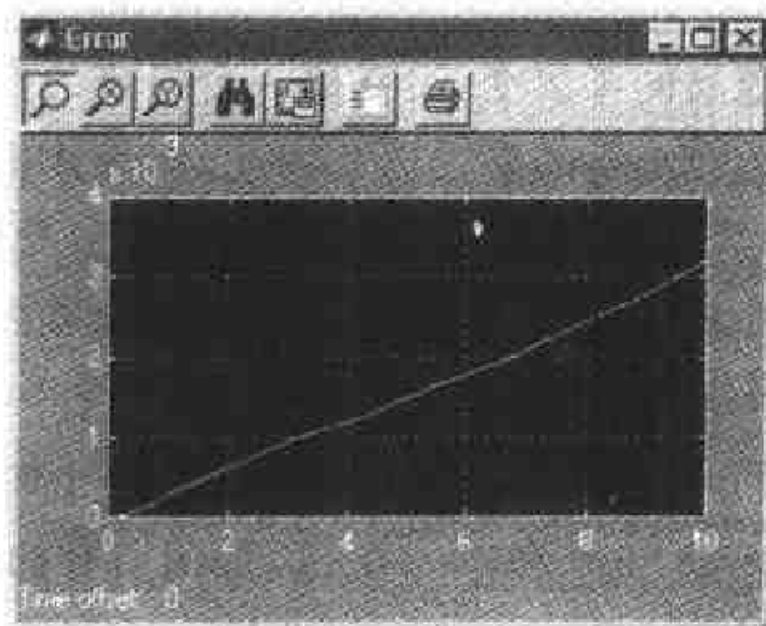


图 7-7 机器人仿真误差输出的矢量的 Scope 模块

注意到这个误差是随着时间的变化而逐渐增加的。它首先精确地从零开始，表明了初始条件设置的比较好，并且在经过 10 s 的仿真过程后，得到的误差值不会超过 10^{-8} m。

7.6 机器人坐标控制

有关机器人控制的详细设计和实现超出了本书的范围。但是，有兴趣的同学可以建立这种仿真，并实现简单的 PD 循环（比例加微分闭环控制）来控制各个节

点电动机的位置(电动机是一种典型的工业化应用产品)。采用“计算扭矩”或者雅可比转置矩阵的更为复杂的仿真方案正在研究中。不管怎么样, Simulink 环境的灵活性可以快速地规范不同的方法,从而来控制和研究它们的效果。

实现机器人操纵装置控制所需的部分仿真留给学生自己去完成。

7.7 小结

在这一章中,得到了一个平面两连杆机器人装置的比较全面的动态模型并在 Simulink 环境下实现了这个模型。这种类型的仿真是很难在更为一般的环境下实施的,这是因为封闭形式的运动方程是非常复杂的,也是很难精确解答的。另外,这种类型的仿真对于了解复杂的多链式机器人操纵装置的动力特性,以及在解决对这些系统实施控制过程中所遇到的困难时是非常有用的。

第7章习题

1. 用本章所讲的动态仿真去研究这种简单的机械系统的运动。尤其是惯性矩阵含有远离对角线的元素的情形。惯性矩阵中远离对角线的元素所隐含的主要意义就是在一个节点处加载将会引起其他节点处的运动。

2. 将本章中平面两连杆机器人的动态仿真扩展到在每个节点电动机上包括简单的 PD(比例微分)控制循环。PD 控制的实质是由电动机产生的扭矩是两个信号的和,这两个信号一个是与期望的节点角度和实际节点角度之间的误差成比例,另外一个与这个误差的导数成比例。这个比例常数,即所熟悉的增益,它对于系统特性有着很大的影响,所以必须对其调整使系统获得较好的性能。用一些时间去调试这些控制循环,以使机器人能够无振荡地从一个节点快速地移动到别的节点。

3. 方程(7-6)显示了将载荷运动与节点运动联系起来的雅可比矩阵。修改仿真模型并写一个适当的 m 文件,此 m 文件将通过明确指定期望的重物速度并利用此关系来控制机器人。

第 8 章 可变机构的仿真

在前面几章中已经处理了单自由度机构的分析和仿真,这些机构的特征是其运动约束不随时间变化。换句话说,连接件的长度始终保持常量并且接头始终互相连接。但是许多有趣的系统,例如有腿的机器人和在接头处具有间隙的机构,必须用更为详细的分析才能够建立适当的仿真模型。在本章中,将研究一个随时间变化的机构及分析这种机构所需的方法。

8.1 间歇机构

图 8-1 是一个间歇机构的示意图。众所周知,间歇机构是一种具有连续旋转驱动源而且能提供间歇运动的机构。右轮以匀速旋转,轮缘附近的销子与左轮的某个槽接合。当销子和槽啮合时,它们象曲柄滑块机构一样工作,当左轮转过 90° 时,销子离开槽,左轮停止转动;当右轮转一圈直到销子和槽再次啮合后,左轮才再次开始运动。

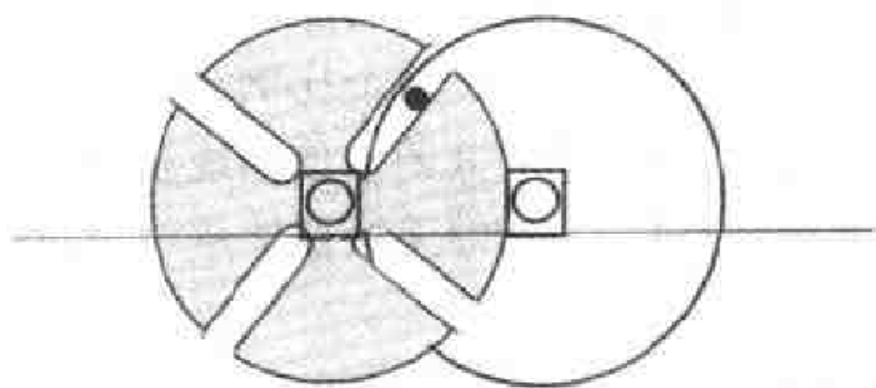


图 8-1 间歇机构示意图

仿真间歇机构的关键是认识到当销子和槽啮合时,机构的运动与倒置的曲柄滑块机构等价;当二者分离时,它们是简单的两个轮子,一个以匀速旋转,另一个静止。因此,若对该机构进行成功地仿真,则要有两组方程——一组描述两个轮子接合在一起的情况,另一组描述二者分离的情况。同时,还需要一组条件以判断在某一时刻哪一组方程起作用。

图 8-2 显示了一个倒置的曲柄滑块机构,其构造形式与图 8-1 给出的间歇

机构类似。图 8-1 中带有销子的右轮可视为连杆 2, 而槽轮可视为连杆 3, 滑块代表槽中的销子。由第 2 章中所列表格, 可以很容易地导出机构的运动方程如下

$$\begin{bmatrix} \cos\theta_3 & -r_3 \sin\theta_3 \\ \sin\theta_3 & r_3 \cos\theta_3 \end{bmatrix} \begin{bmatrix} \ddot{r}_3 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} -a_2 r_2 \sin\theta_2 - r_2 \omega_2^2 \cos\theta_2 + 2\dot{r}_3 \omega_3 \sin\theta_3 + r_3 \omega_3^2 \cos\theta_3 \\ a_2 r_2 \cos\theta_2 - r_2 \omega_2^2 \sin\theta_2 - 2\dot{r}_3 \omega_3 \cos\theta_3 + r_3 \omega_3^2 \sin\theta_3 \end{bmatrix}$$

这组方程可以非常容易地嵌入到第 4 章中所论述的运动学仿真中去。图 8-3 给出了一个可执行的 Simulink 仿真框图。

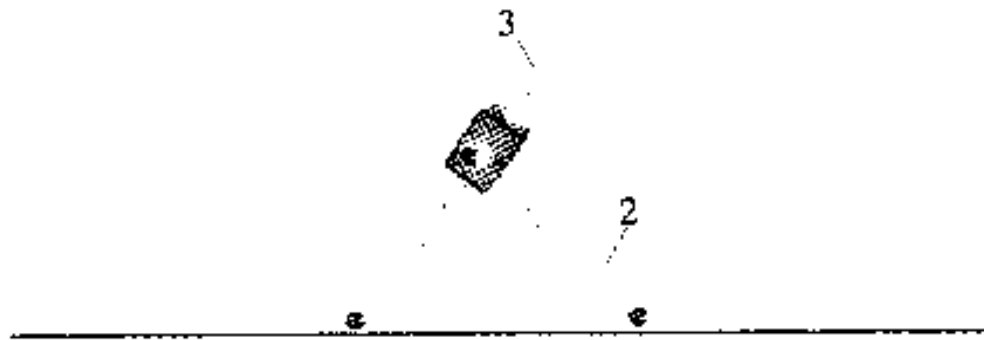


图 8-2 与销子和槽接合时的间歇机构相类似的倒置曲柄滑块机构

从销子和槽接合开始, 被驱动的左轮旋转了 90° , 此时销子离开槽, 系统的运动由另外一组方程控制, 直到右轮旋转一圈销子和下一个槽接合。在这段时间控制系统运动的方程很简单, 可写为

$$\begin{aligned} \alpha_3 &= 0 \\ \ddot{r}_3 &= 0 \end{aligned}$$

在图 8-3 中嵌入的函数可以经过简单地修改来检验机构的相对位置以及使用适当的方程来计算加速度。我们发现, 编写一个监测 θ_2 的值并根据销子是否接合而将输出值设置为 0 或 1 的新函数是非常有用的, 这一点在后面将会看得很清楚。下面给出了这个监测函数 `detect()` 的 MATLAB 脚本:

```
detect.m
function trigg=detect(u)
%
% function to detect engagement of the Geneva Wheel
% It assumes a symmetric arrangement of two wheels of the
% same size and a 90 degree intermittent motion.
%
% The function signals engagement (output goes to 1.0)
% if the driving wheel (right-hand wheel) is between 135
% and 225 degrees
%
```

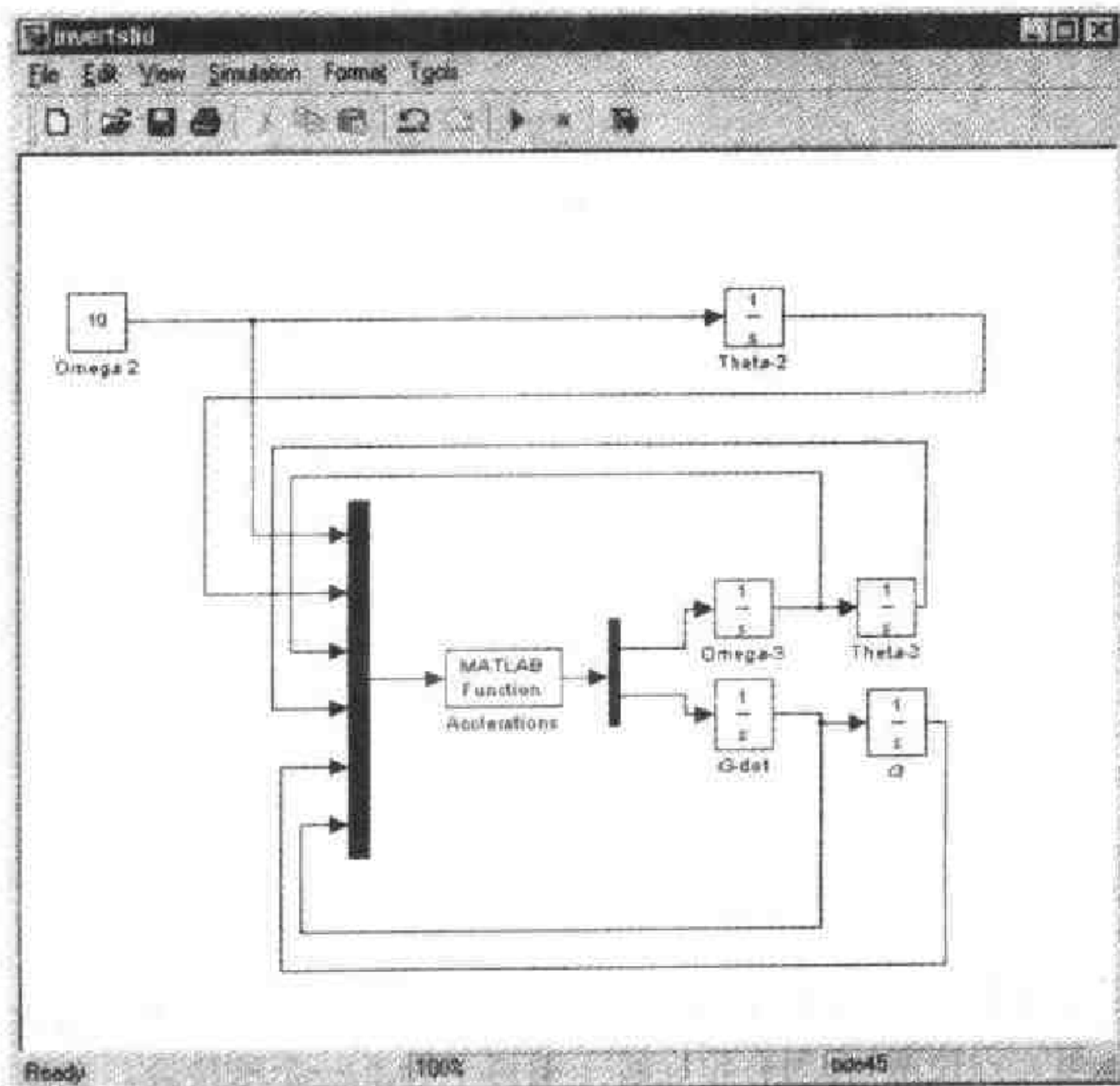


图 8-3 用于倒置曲柄滑块机构运动学仿真的 Simulink 仿真框图
(文件:cnvertslid.mdl)

```
ang=mod(u(1),2*pi);
if (ang>(135*pi/180) & ang<(225*pi/180))
    trigg = 1.0;
else
    trigg = 0.0;
end
```

该函数用来判断销子是否与槽啮合,函数的输出是计算加速度函数的输入。

现在我们基本上完成了间歇机构的运动学仿真。剩下唯一的一件事情是考虑初始条件和在此情况下的积分器输出。对标识 θ_2 和 θ_3 的积分器,其初始条件相对简单。可以假设在仿真初始时刻 $\theta_2 = 0$ 并且 $\omega_3 = 0$,因为当 $\theta_2 = 0$ 时销子没有与槽啮合。然而要给出另外三个积分器的初始条件却遇到了特殊的困难,这一点将在下面进行详尽的讨论。

对与矢量 R_3 相关的两个积分器必须仔细地考虑。当销子和槽啮合时,它们代表销子相对槽的速度和位移。在其他时刻,它们应当为零。当销子开始与槽接合

时,由监测函数产生的信号从0切换到1,同时 r_3 应当是它的最大值(大多数间歇机构的 r_3 与 r_2 相等),而相对速度应当与销子的绝对速度一样,因为销子从切线方向进入槽。销子每次进入槽时,这两个积分器必须重新设置成这些值。一般看来,这一问题的处理似乎是不可能的,但 Simulink 积分器具有一些特殊的性质并能够很好地处理这些情况。积分器可以通过信号的转换(例如监测信号从0变化到1)而加以重新设置。此外,使用者可以选择对话框中确定的值或在另一输入行中出现的值来重新设置积分器。因此,可将标识为 r_3 和“ $r_3 - \text{dot}$ ”的积分器设置成上升信号。对位移积分器,选择使用对话框中确定的值,即初始条件的“内部”值。然而,对于速度积分器,初始条件依赖于驱动轮在接合瞬时的速度。因此,只有在外部特性选定的情况下初始条件才能通过计算一个关于 ω_2 的函数确定下来。

最后,考虑 θ_3 的问题。对于间歇机构,槽轮间歇地运动,每次旋转 90° 。从某个初始条件,例如 $\pi/4$ rad(如图8-1所示),开槽的轮将在带销子的轮旋转完一圈的时候沿顺时针转到 $-\pi/4$ 的位置(注意驱动轮以逆时针方向旋转,所以开槽的轮子为顺时针方向旋转,角度的负号显示了相反的运动方向)。下一圈,带销子的轮将带动开槽的轮子以顺时针从 $-\pi/4$ 转到 $-3\pi/4$ 的位置,其余的依此类推。与此相反,与间歇机构类似的倒置曲柄滑块机构的运动由下面的规律决定:被驱动连杆在接合的一个周期内以逆时针从 $\pi/4$ 运动到 $-\pi/4$ 。如果在我们的仿真中计算得到的转角 θ_3 直接输入到计算加速度函数中(如图8-3所示),则会导致在每4次销子接合中有3次产生错误。为了解决此问题,我们引入另外一个积分器,该积分器将会被重新设置为上升监测信号 $\pi/4$ 。这个在仿真中被标识为 $\theta_3 - \text{hat}$ 的值,将被用于计算加速度的函数中,这样使得倒置曲柄滑块机构在任何时候都是正确的。

最终的 Simulink 简图如图8-4所示。

计算加速度的函数的 MATLAB 脚本如下:

```
genacc.m
```

```
function out=genacc(u)
%
% function to compute accelerations of a geneva wheel
%
% u(1) = omega-2
% u(2) = theta-2
% u(3) = omega-3
% u(4) = theta-3-hat (for equiv inverted slider crank)
% u(5) = r3
% u(6) = r3-dot
% u(7) = engagement trigger
```

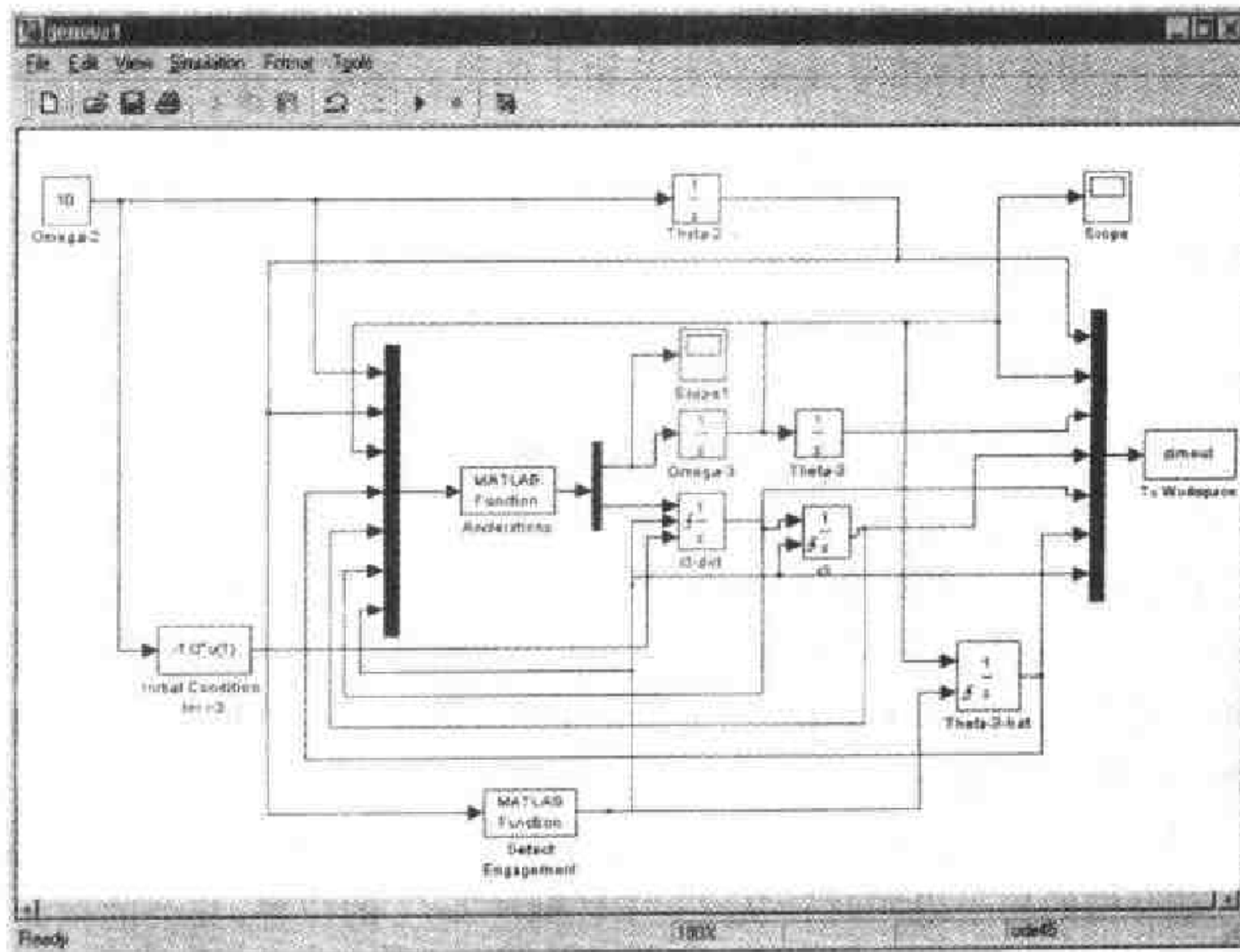


图 8-4 图 8-1 所示间歇机构完整的 Simulink 仿真图(特别地,请注意那些为监测销子是否和机构中键槽接合的函数而重新设置的积分器。(文件:general.mdl))

```

%
r2=1.0;
%
if u(7) == 0
    out(1)=0.0;
    out(2)=0.0;
else
    S2=sin(u(2));
    C2=cos(u(2));
    S3=sin(u(4));
    C3=cos(u(4));
    a=[u(5)*C3 S3 ; -u(5)*S3 C3];
    b=[-r2*u(1)^2*S2-2*u(6)*u(3)*C3+u(5)*u(3)^2*S3;
        -r2*u(1)^2*C2+2*u(6)*u(3)*S3+u(5)*u(3)^2*C3];
    out=inv(a)*b;
end

```

图 8-5 和图 8-6 显示了驱动轮在给定速度 10 rad/s 匀速转动时被动轮的运动。

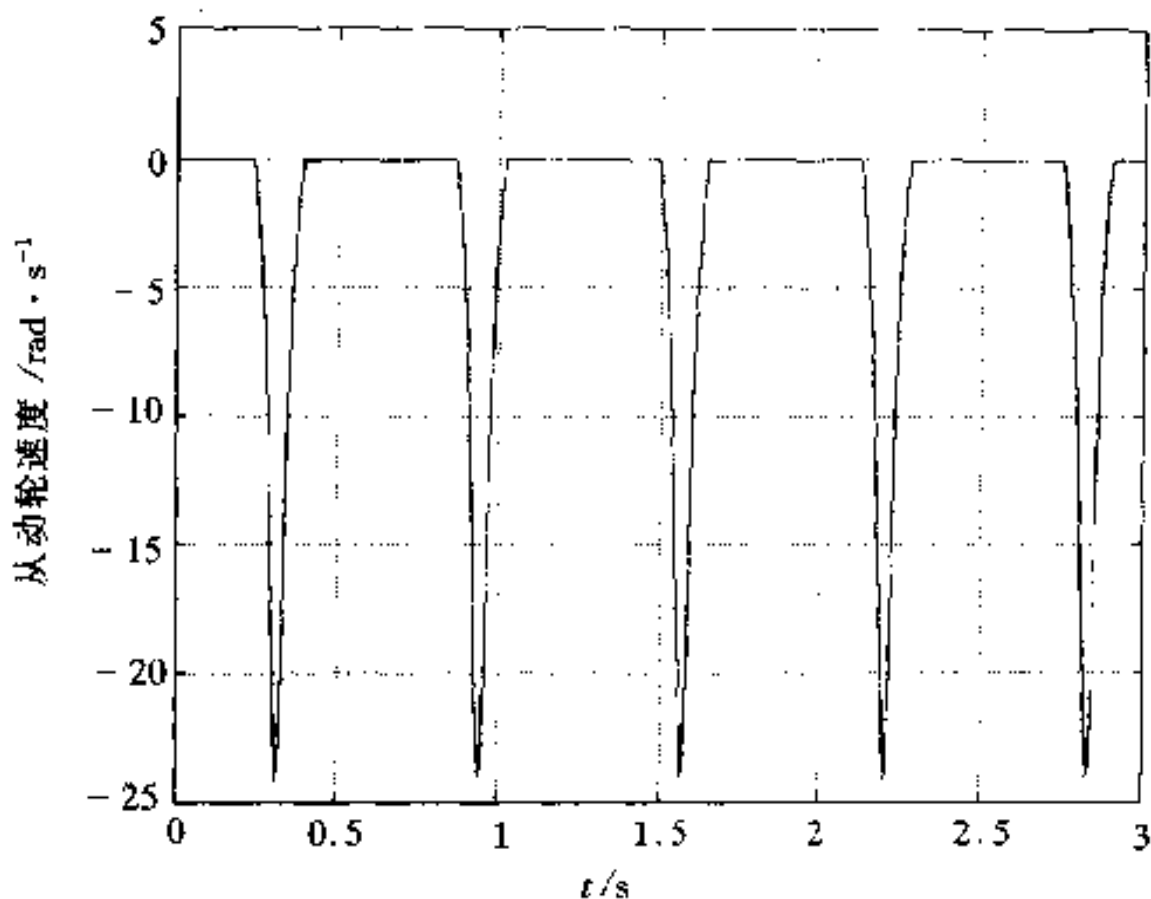


图 8-5 运动学仿真计算得到的间歇机构被动轮角速度

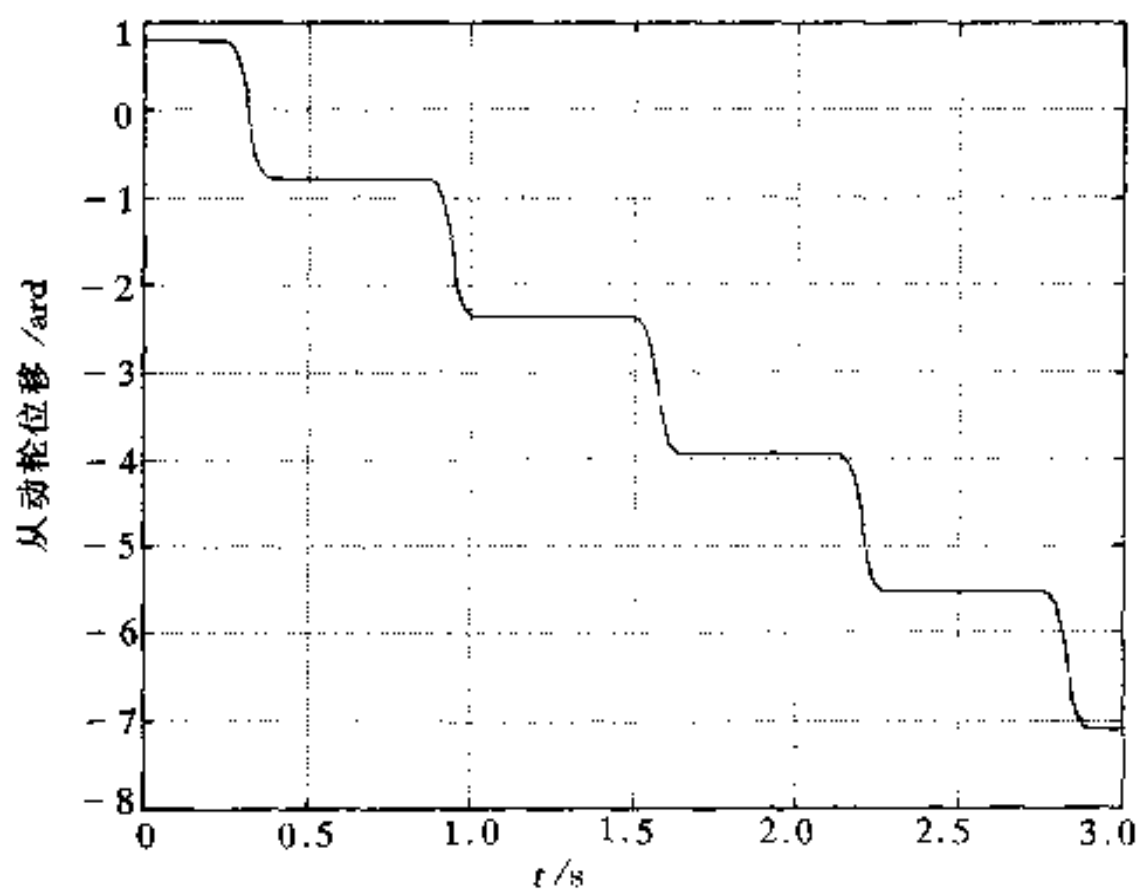


图 8-6 运动学仿真计算得到的间歇机构被动轮的角位移

8.2 小结

在本章中,利用 MATLAB 和 Simulink 的某些高级特性来实现拓扑结构随时间变化的系统运动学仿真。间歇机构是具有双解耦轮子的一种机构,其运动学特征和倒置曲柄滑块机构相类似。本章给出了对这样一类间歇机构进行运动学仿真的方法。

第9章 抛石机

9.1 概述

抛石机是一种很古老的装置,可以追溯到中世纪。其前身是人们熟知的弹弓。和弹弓一样,抛石机也是用来把重物,如石块等抛到更高、更远的地方。在古代,抛石机的主要用途是作为一种攻城武器。当一国军队围攻另一国的城池时,他们就使用抛石机将石块等投掷物扔过城墙,以达到杀伤对方之目的。

图9-1是抛石机的简图。根据不同的比例,我们既可以做成只有桌面大小的东西用来抛掷铅笔、橡皮等;也可做成一庞然大物,用来抛掷汽车等重物^①。

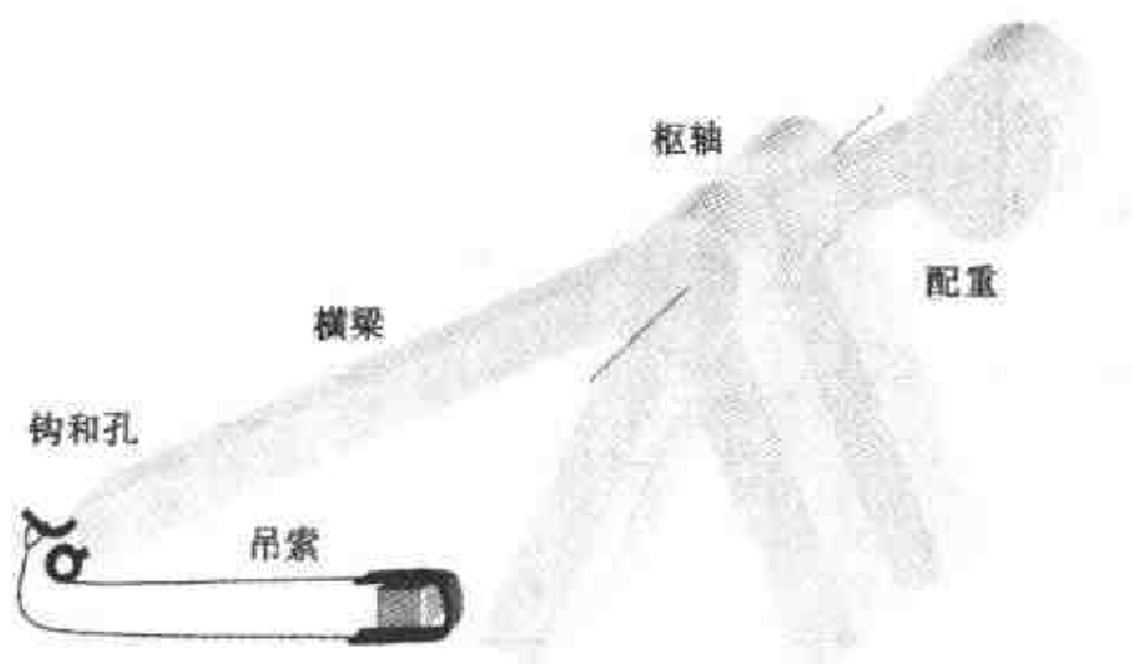


图9-1 抛石机简图

图9-1说明了抛石机已经扣上扳机,准备击发。图9-2是抛石机在使用时的一个顺序简图。一旦释放,配重由于重力而下降,使得横梁顺时针旋转,拉动发射物沿地面水平向左运动。当横梁前端的吊钩拉动吊索离开地面时,发射物运动产生的相当大的水平向左的动量转化为沿切线方向的旋转动量。在运动过程中,发射物不断加速,最终超过横梁,使吊索上的绳环脱离吊钩,这样吊索打开,发射物

^① Chevedden p e., et al. The Trebuchet. Scientific American, 1995, 273 (1): 66~71.

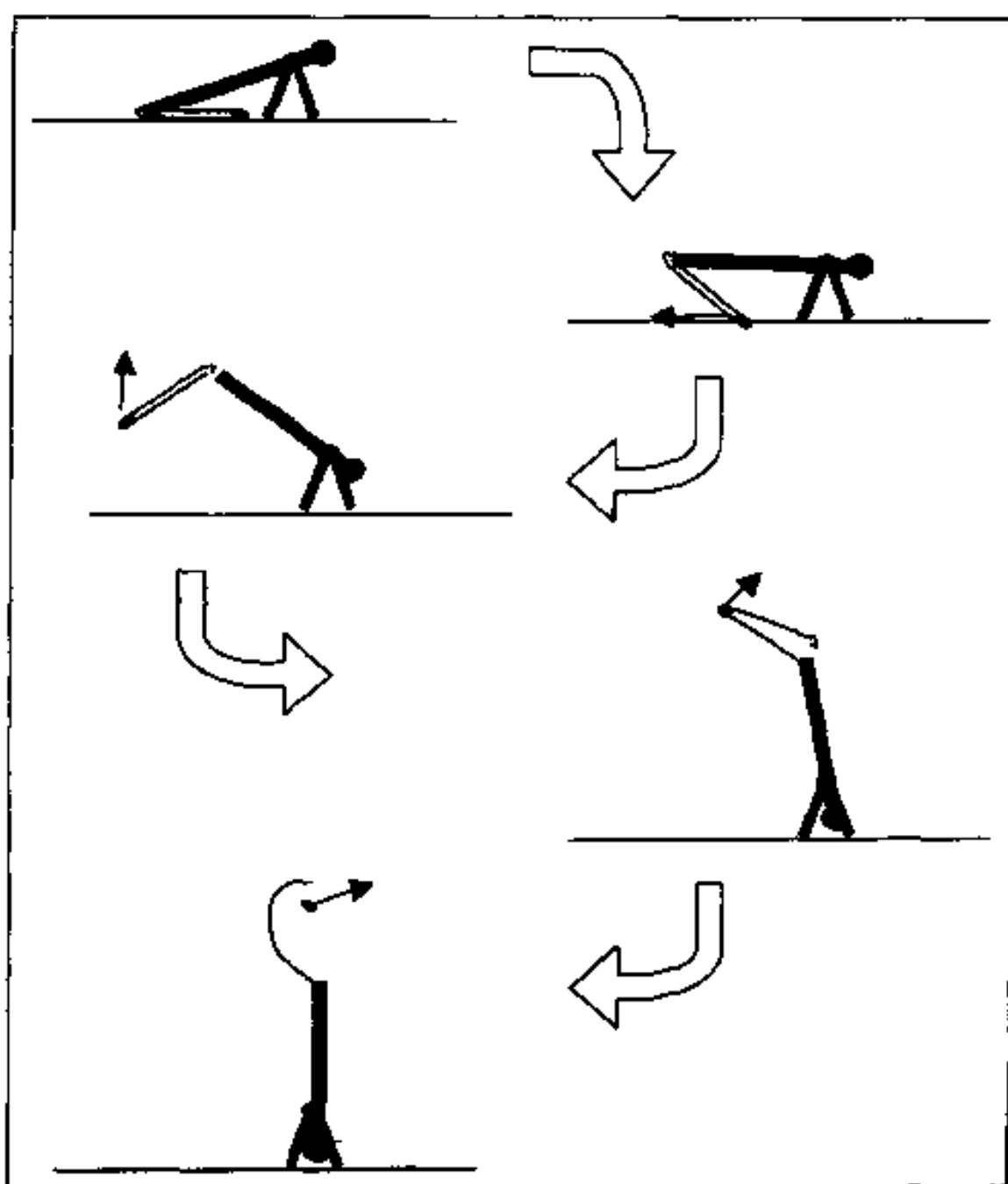


图 9-2 抛石机运动分解过程简图(最上端第一幅图说明抛石机已经准备击发。一旦释放,配重使得横梁旋转,从而带动吊索及环绕的发射物一起运动,最终将发射物抛向右方)

就可以很高的速度向右上方发射出去。中世纪人们就会使用这种设计精巧的装置,在打仗时用它向远距离发射重物。

抛石机是一个非常好的易于动态仿真的机械装置。用传统的运动学方法来分析它是很困难的。这种动力机构本身所产生的速度是不断变化的,它只受到重力的作用,而且我们也很难预测吊索的释放时间。但是,这种装置对其各个不同构件的几何参数和惯性参数具有很高的灵敏度。由于不能实际去制作和实验抛石机,我们只能通过完整的动力分析来分析这种装置。本书中介绍的这种方法适合用 MATLAB 和 Simulink 软件来进行计算机模拟,不久将看到这一点。

9.2 矢量环

抛石机的矢量环非常简单,很容易得到约束方程。图 9-3 给出了抛石机在运动过程中相关的位移矢量。

取抛石机的旋转轴点为坐标原点,所得矢量方程为

$$\mathbf{R}_2 + \mathbf{R}_3 = \mathbf{R}_4 \quad (9-1)$$

在此方程中,我们假定吊索一直处于张拉状态,并且绳环也一直挂在吊钩上,下面我们将讨论当绳环打开,抛射物开始飞出时的情况。注意到横梁的重心一般不与位移矢量方向重合,我们可以连接矢量起始位置和物体中心的连线,设其距离为 r_2 ,符号取负号。

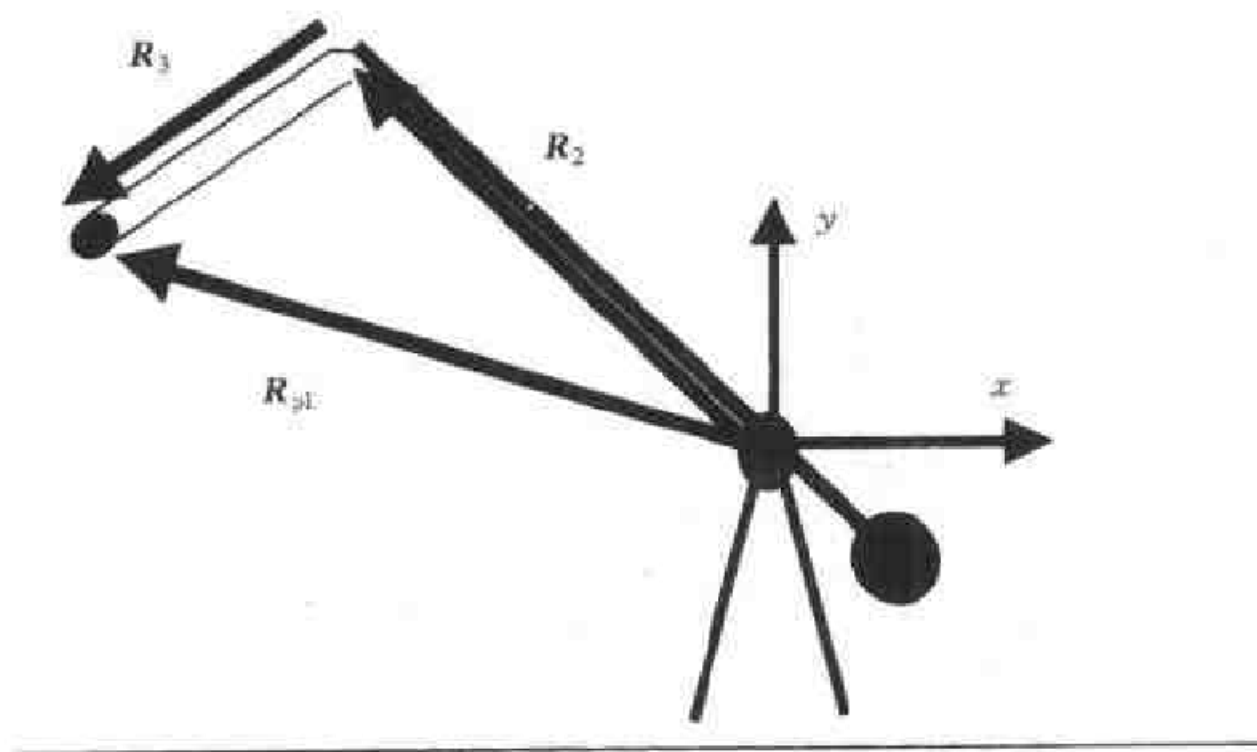


图 9-3 显示描述抛石机位移矢量的矢量图

矢量环方程沿 x 方向、 y 方向的分量为

$$r_2 \cos\theta_2 + r_3 \cos\theta_3 = x_p \quad (9-2)$$

$$r_2 \sin\theta_2 + r_3 \sin\theta_3 = y_p \quad (9-3)$$

对方程关于时间取二阶导数得到加速度方程为

$$\ddot{x}_p + r_2 a_2 \sin\theta_2 + r_3 a_3 \sin\theta_3 = -r_2 \omega_2^2 \cos\theta_2 - r_3 \omega_3^2 \cos\theta_3 \quad (9-4)$$

$$\ddot{y}_p - r_2 a_2 \cos\theta_2 - r_3 a_3 \cos\theta_3 = -r_2 \omega_2^2 \sin\theta_2 - r_3 \omega_3^2 \sin\theta_3 \quad (9-5)$$

因为横梁的重心与旋转轴点(根据它来确定扭矩方程)不重合,我们还需找出横梁重心的加速度与角加速度和速度的关系。此时,如前面的一样,矢量方程就显得不重要了。由 r_{2c} 的含义有

$$r_{2c} \cos\theta_2 = x_{2c} \quad (9-6)$$

$$r_{2c} \sin \theta_2 = y_{2c} \quad (9-7)$$

对上述两式求关于时间的二阶导数,我们就可以得到如下的加速度方程

$$\ddot{x}_{2c} + r_{2c} \alpha_2 \sin \theta_2 = -r_{2c} \omega_2^2 \cos \theta_2 \quad (9-8)$$

$$\ddot{y}_{2c} - r_{2c} \alpha_2 \cos \theta_2 = -r_{2c} \omega_2^2 \sin \theta_2 \quad (9-9)$$

9.3 运动方程的建立

图9-4为横梁的隔离体受力图,作用的外力如图所示。注意到我们在第一象限引入正角 θ_2 ,这样能很容易准确地说明各个力的符号。

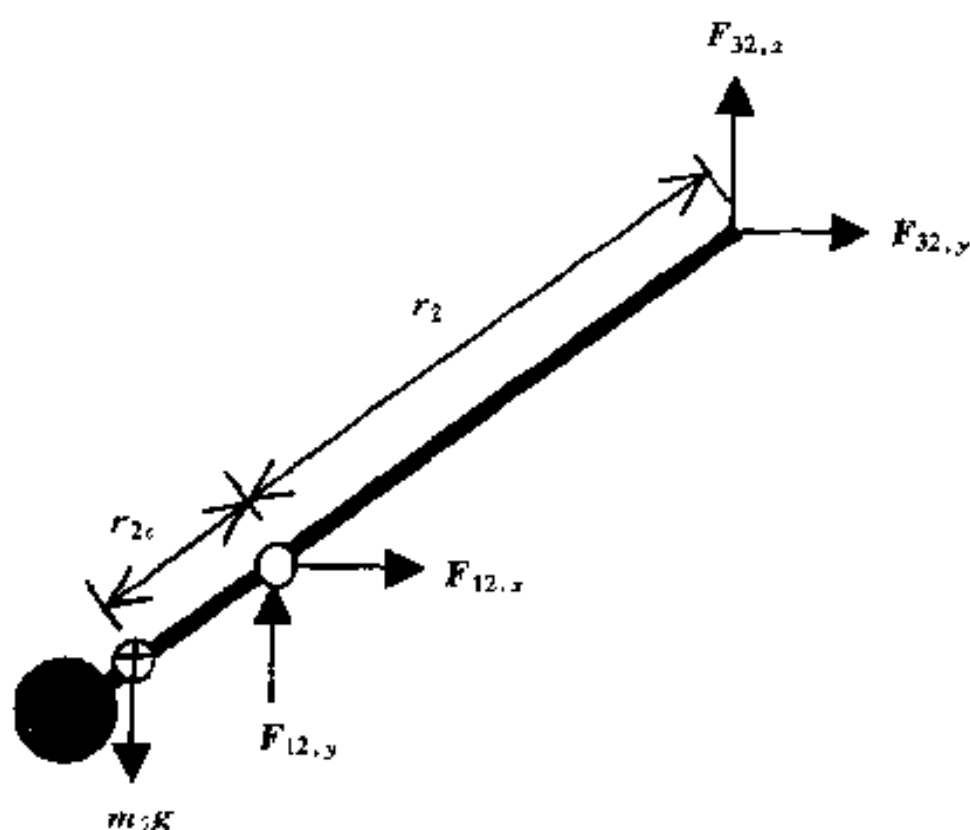


图9-4 抛石机横梁的受力图

下面是横梁的三个运动方程

$$F_{12,x} + F_{32,x} = m_2 \ddot{x}_{2c} \quad (9-10)$$

$$F_{12,y} + F_{32,y} - m_2 g = m_2 \ddot{y}_{2c} \quad (9-11)$$

$$-F_{32,x} r_2 \sin \theta_2 + F_{32,y} r_2 \cos \theta_2 - m_2 g r_{2c} \cos \theta_2 = I_2 \alpha_2 \quad (9-12)$$

同样地,图9-5表示的是吊索的受力分析图。注意到吊索是用粗绳或钢丝绳做成,不是刚性连接,所以不能传递扭矩。而且绳子的质量一般比横梁和抛射物的质量小的多,我们就可以忽略吊索的惯性力,而认为抛射物是作用在吊索末端的一个质点。

我们很容易写出下面三个方程:

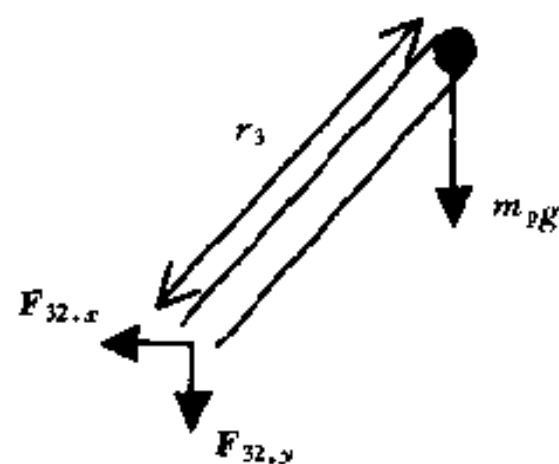


图9-5 吊索的受力图(为了分析的需要,假设吊索处于张拉状态)

$$-F_{32,x} = m_p \ddot{x}_p \quad (9-13)$$

$$-F_{32,y} - m_p g = m_p \ddot{y}_p \quad (9-14)$$

$$-F_{32,x} r_3 \sin\theta_3 + F_{32,y} r_3 \cos\theta_3 = 0 \quad (9-15)$$

方程(9-15)说明吊索不能传递扭矩,可以改写成以下更简单的形式

$$F_{32,y} - F_{32,x} \tan\theta_3 = 0 \quad (9-16)$$

9.4 矩阵方程

在前面的两节里,我们得到了带有 10 个未知变量的 10 个方程(方程(9-4)~(9-15))。下面我们将其转化为矩阵形式,以便运用 MATLAB 的矩阵程序来求解(这里 $C_2 = \cos\theta_2$, $S_2 = \sin\theta_2$, $C_3 = \cos\theta_3$, $S_3 = \sin\theta_3$ ——译者注)

$$\begin{bmatrix} 1 & 0 & r_2 S_2 & r_3 S_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -r_2 C_2 & -r_3 C_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{2c} S_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -r_{2c} C_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 & 0 & -1 & 0 & -1 \\ 0 & 0 & I_2 & 0 & 0 & 0 & 0 & 0 & r_2 S_2 & -r_2 C_2 \\ m_{pl} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & m_{pl} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -S_3 & C_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_p \\ \ddot{y}_p \\ \alpha_2 \\ \alpha_3 \\ \ddot{x}_{2c} \\ \ddot{y}_{2c} \\ F_{12,x} \\ F_{12,y} \\ F_{32,x} \\ F_{32,y} \end{bmatrix} = \begin{bmatrix} -r_2 \omega_2^2 C_2 - r_3 \omega_3^2 C_3 \\ -r_2 \omega_2^2 S_2 - r_3 \omega_3^2 S_3 \\ -r_{2c} \omega_2^2 C_2 \\ -r_{2c} \omega_2^2 S_2 \\ 0 \\ -m_2 g \\ -m_2 g r_{2c} C_2 \\ 0 \\ -m_{pl} g \\ 0 \end{bmatrix} \quad (9-17)$$

9.5 动态仿真

在上一节里,方程(9-17)描述的是在使抛射物产生动量的时间段内抛石机的动力特性。当吊索被释放,抛射物被发射出去后,式(9-17)对研究抛射物的运动也具有指导意义。

这时,我们可以忽略吊索的影响,只计算抛射物和横梁的运动加速度。运动方程如下:

$$\ddot{x}_p = 0 \quad (9-18)$$

$$\ddot{y}_p = -g \quad (9-19)$$

$$\alpha_2 = -m_2 / I_2 g r_{2c} C_2 \quad (9-20)$$

要想测出吊索释放的确切时间是非常困难的,所以我们可以避免这一点,假设通过反复实验和修正,调整横梁末端的吊钩使其在抛射物的速度方向与水平方向成 45° 角时释放吊索,这样将得到抛射的最远距离。

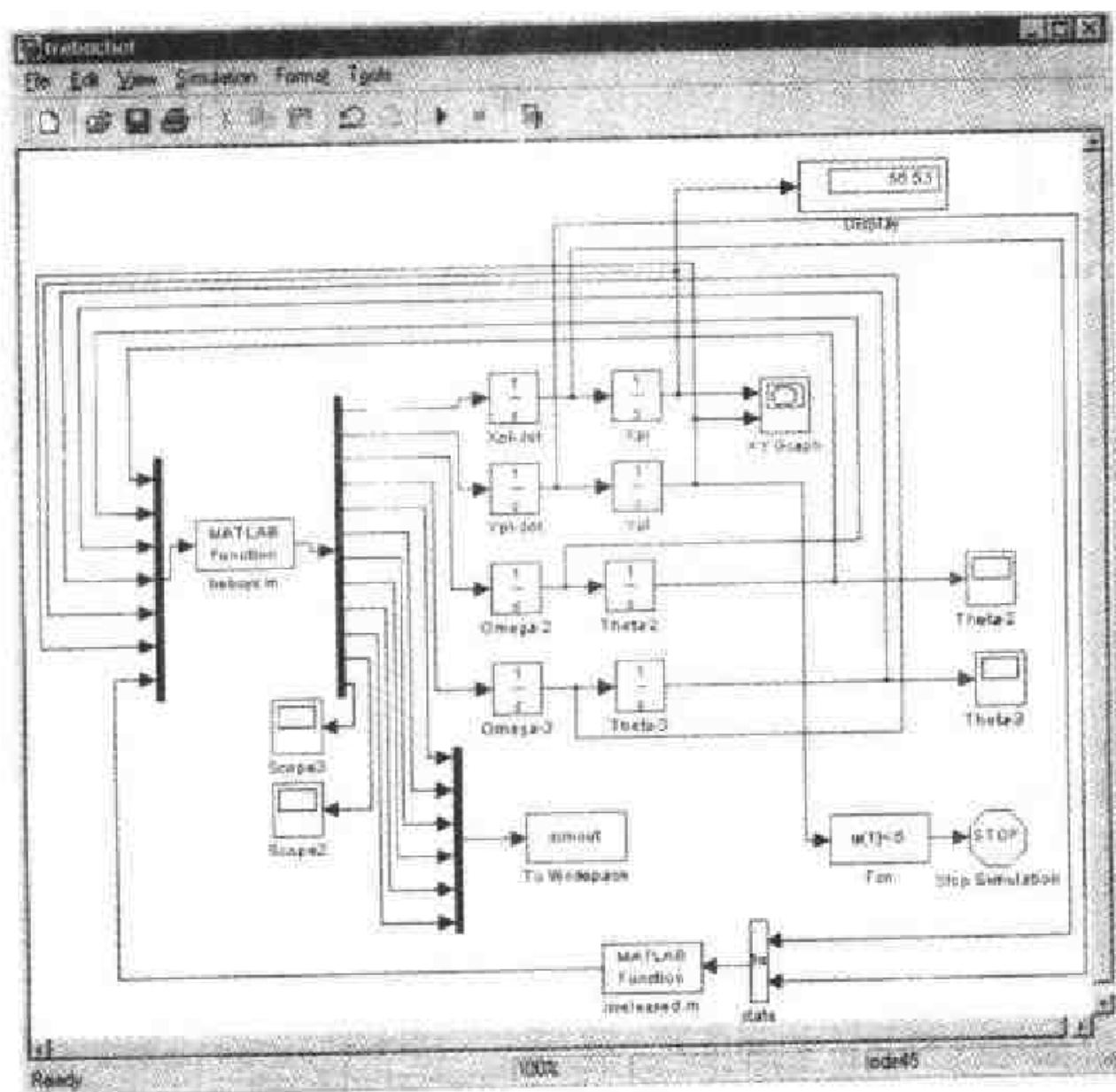


图 9-6 抛石机仿真的 Simulink 实现过程(文件:trebuchet.mdl)

图 9-6 为抛石机的 Simulink 仿真图。在这个仿真中要调用两个函数文件。一个是 `trebsys.m` 函数,用来解方程(9-17)的联立约束矩阵,另一个为 `isreleased.m` 函数,用来设置抛射物体的释放。

通过图 9-6 我们看到这个仿真过程非常繁琐,要想一个个都实现就很困难。这时,把功能相关的块进行分组,并把它们隐藏在相应的子系统里就很有意义,这个内容留给学生自己来做。

仿真图中有两点新东西,第一点是当抛射物块落回地面时,“Stop”块就会插入,自动结束仿真;第二点是“Display”块可以显示抛射物发射出去的距离。“X-Y Graph”块还可以得到抛射物运动的曲线图。

下面列出了计算联立约束的 m 文件程序脚本:

```
trebsys.m
```

```
function [out]=trebsys(u)
%
% u(1) = Theta-2
% u(2) = Omega-2
% u(3) = Theta-3
% u(4) = Omega-3
% u(5) = Xpayload
% u(6) = Ypayload
% u(7) = release switch; 0 implies normal, 1 implies release
%
r2=3.0;
r3=1;
r2c=-0.5;
m2=500.0;
I2=100.0;
mpl=5.;
g=9.81;
%
S2 = sin(u(1));
C2 = cos(u(1));
S3 = sin(u(3));
C3 = cos(u(3));
%
if u(7)==0
a=zeros(10,10);
%
a(1,1)=1;a(1,3)=r2*S2;a(1,4)=r3*S3;
a(2,2)=1;a(2,3)=-r2*C2;a(2,4)=-r3*C3;
a(3,3)=r2c*S2;a(3,5)=1;
a(4,3)=-r2c*C2;a(4,6)=1;
a(5,5)=m2;a(5,7)=-1;a(5,9)=-1;
a(6,6)=m2;a(6,8)=-1;a(6,10)=-1;
a(7,3)=I2;a(7,9)=r2*S2;a(7,10)=-r2*C2;
a(8,1)=mpl;a(8,9)=1;
a(9,2)=mpl;a(9,10)=1;
a(10,9)=-S3;a(10,10)=C3;
b(1)=-r2*u(2)^2*C2-r3*u(4)^2*C3;
b(2)=-r2*u(2)^2*S2-r3*u(4)^2*S3;
b(3)=-r2c*u(2)^2*C2;
b(4)=-r2c*u(2)^2*S2;
b(5)=0;
b(6)=-m2*g;
b(7)=-m2*g*r2c*C2;
b(8)=0;
b(9)=-mpl*g;
b(10)=0;
```

```

%
%
out=inv(a)*b';
%
errx=r2*C2+r3*C3+u(5);
erry=r2*S2+r3*S3-u(6);
out(11)=errx;
out(12)=erry;
else % released
    out(1)=0.0;
    out(2)=-g;
    out(3)=-m2*g*r2c*C2/I2;
    out(4)=0.0;
    out(5:12)=[0 0 0 0 0 0 0 0];
end

```

最后,列出如下的判断抛掷物释放的函数程序:

```

isreleased.m
function sw=isreleased(u)
%
% u(1) = Vx-payload
% u(2) = Vy-payload
%
ang=atan2(u(1),u(2));
if ang > pi/4
    sw=1;% released
else
    sw=0;
end

```

9.6 仿真结果

表9-1给出了一个大型抛石机的一些典型参数。

表9-1 抛石机的惯性参数和几何参数

参数	取值	参数	取值
m_2/kg	500	r_2/m	3.0
m_0/kg	100	r_{c2}/m	-0.5
$I_2/\text{kg} \cdot \text{m}^2$	10	r_3/m	1.0

在抛石机准备击发阶段,各初始状态已知,见表9-2。

表 9-2 抛石机仿真的各初始条件

积分求解器	初值	积分求解器	初值
xpl-dot	0	X_{pl}/m	1.121 32
ypl-dot	0	Y_{pt}/m	-2.121 32
ω_2	0	$\theta_2/(\circ)$	255
ω_3	0	$\theta_3/(\circ)$	0

图 9-7 给出了一个质量为 10 kg 的重物的抛射轨迹

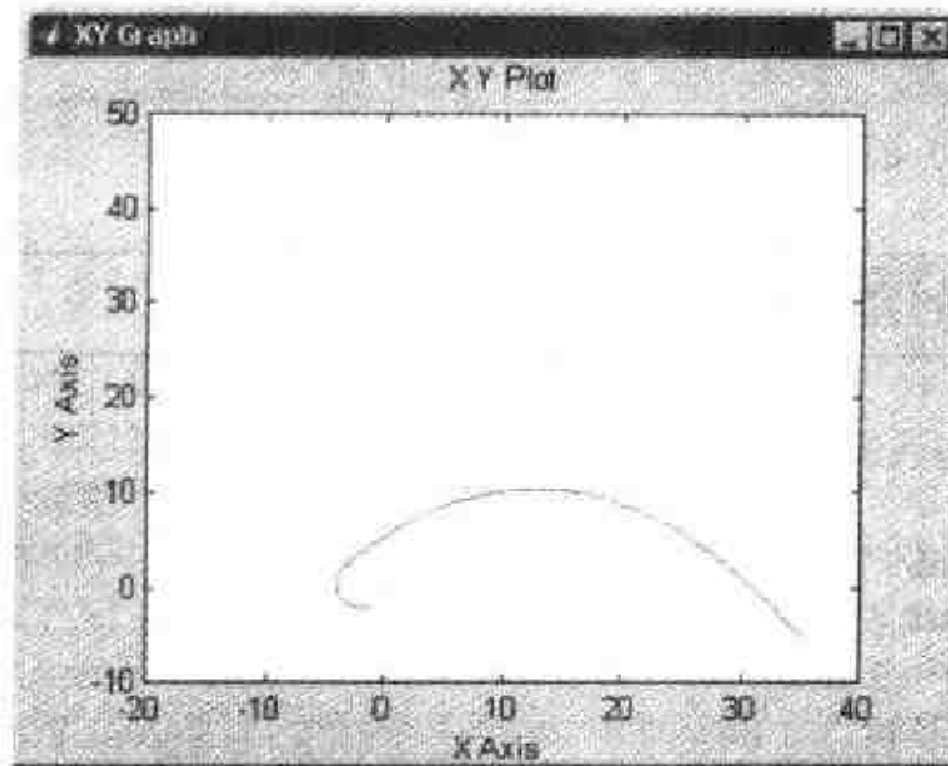


图 9-7 10 kg 的重物的抛射轨迹(注意到抛石机将重物抛射了 35 m 多的距离)

同样地,仿真可以预测一个 1 kg 重的抛射物体的抛射路线,见图 9-8,并且可以计算出抛射物块的飞行距离为 128 m。

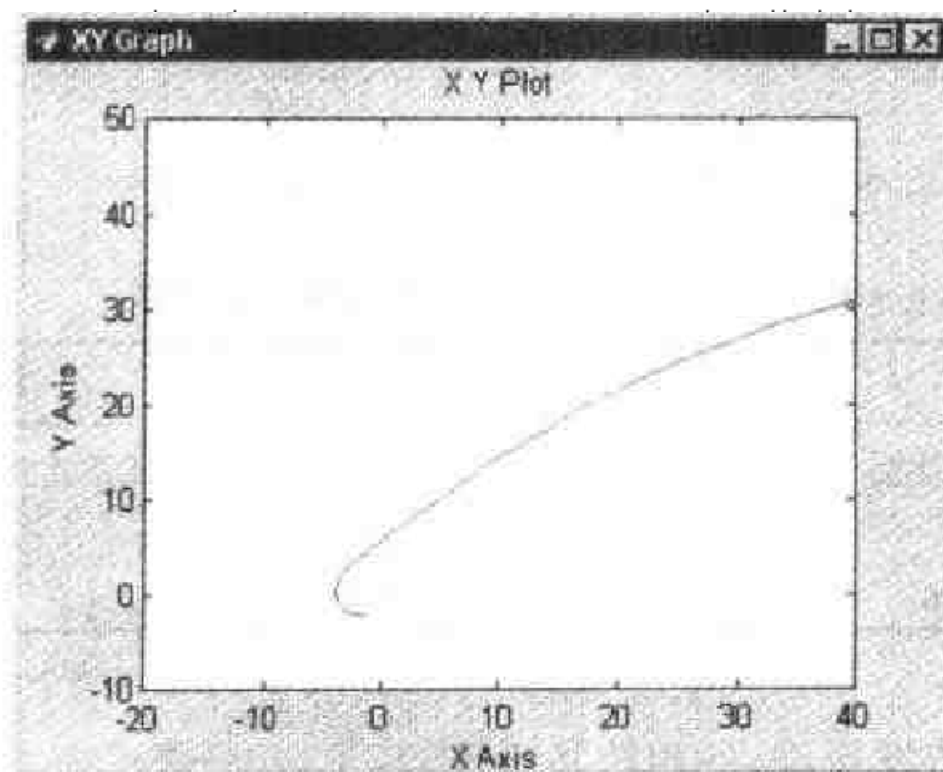


图 9-8 1 kg 重物的抛射轨迹(重物被抛掷 128 m 多的距离)

9.7 小结

抛石机是一种外表结构简单,但运动形式复杂的装置,虽然它只有一个旋转点,但却能表现出非常复杂的动力行为。我们不能用在大学本科的运动学课程中所讲的那些方法来分析这种机器。每一部分的运动速度都是在时刻变化的。整个机械本质上都是动态的,只能进行动态分析。

在本书中,所有的机构仿真都是如此,我们现在讲的这种方法仅仅是个起点,仿真能够(也应该)拓宽成为设计工具或用来进行复杂模型的分析。

本书试图提供一些方法和指导,使同学们和实践者充分运用 MATLAB 或类似的仿真软件较快地进行约束系统的全动态仿真,而不必用复杂的能量方法,该方法限于研究生水平的动力学知识。这种方法只需要掌握基本的工程矢量力学和矩阵分析的基本原理就可以了,希望这种方法在很多工程应用领域对读者有所帮助。

第9章习题

1. 与抛石机打交道的工程人员很快发现对抛石机的基本设计作些修改会改进其性能。修改之一是引入旋转式吊桶以代替固定的配重。这使得横梁的转动惯量比较低,从而提供更高的抛射速度。请修改仿真模型以考虑上述设计改进,并验证这样确实改进了抛石机的性能。

2. 另一个重要的改进是将整个装置安装于轮子上,使得整个装置能够和投掷方向相同的方向在水平面上滚动。这样不仅改进了装置的移动性能,而且也改进了装置的性能。请修改仿真模型以允许基础可以在轮子上水平运动,并通过仿真实验确定基础质量相对于配重的比值,从而确定改进系统性能的最佳参数组合。

附录 Simulink 入门辅导

本入门辅导的目的在于使学生在很短的时间内熟悉 Simulink 的基本知识。Simulink 是一个具有丰富特性和强大数值算法的多功能分析工具。在线帮助文件（通过 MATLAB 帮助命令可以获得这些文件）提供了非常完整的叙述，使用者应当熟悉 MATLAB 的这一特性。

A.1 启动 Simulink

MATLAB 的开始窗口如图 A-1 所示。

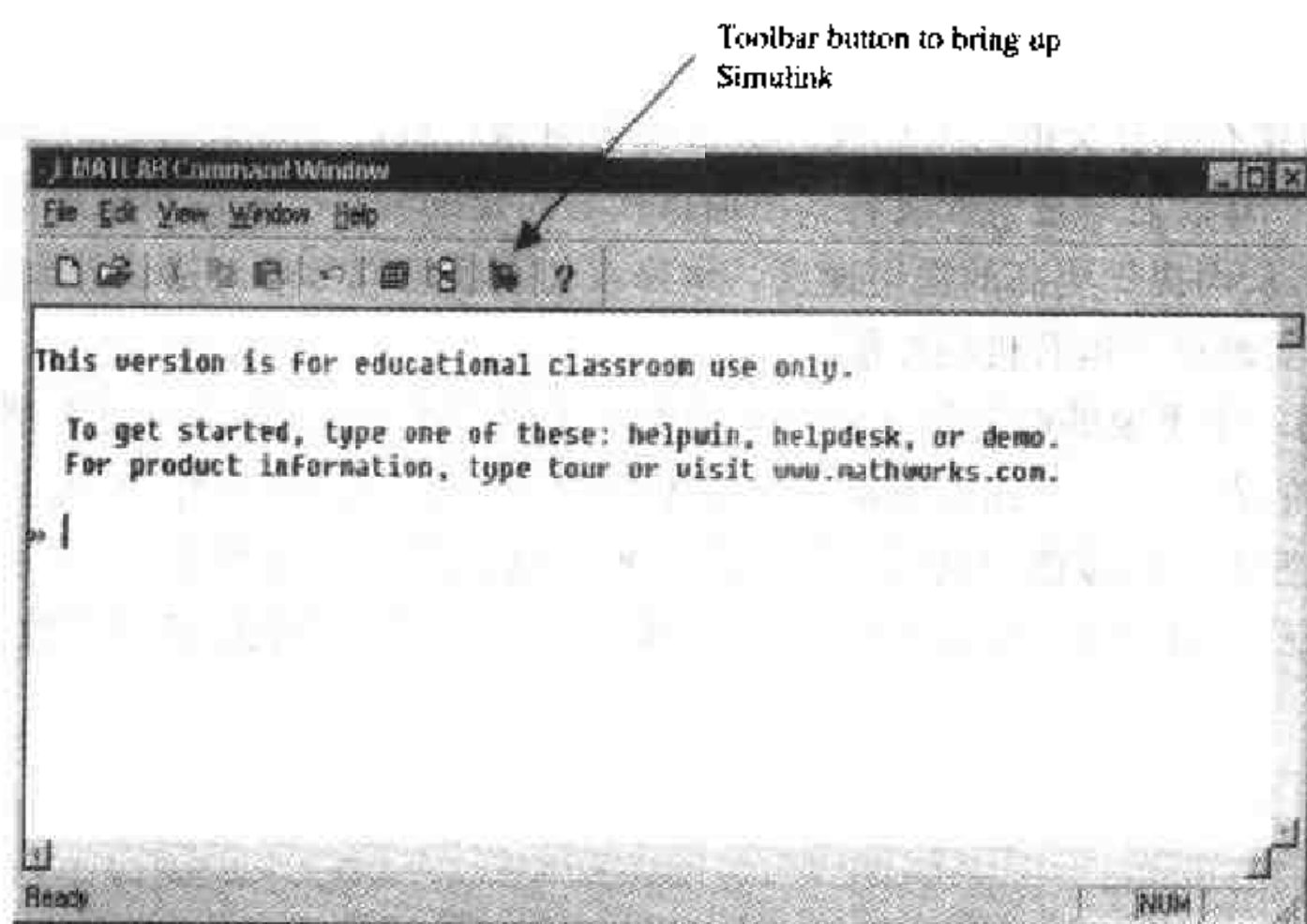


图 A-1 MATLAB 默认的命令窗口

注意在 MATLAB 默认的窗口工具条中的 Simulink 启动按钮。如果该按钮是不可见的，则用户的计算机就没有安装 Simulink，请与用户的系统管理员联系。当用户点击 Simulink 启动按钮，会弹出如图 A-2 所示的窗口。

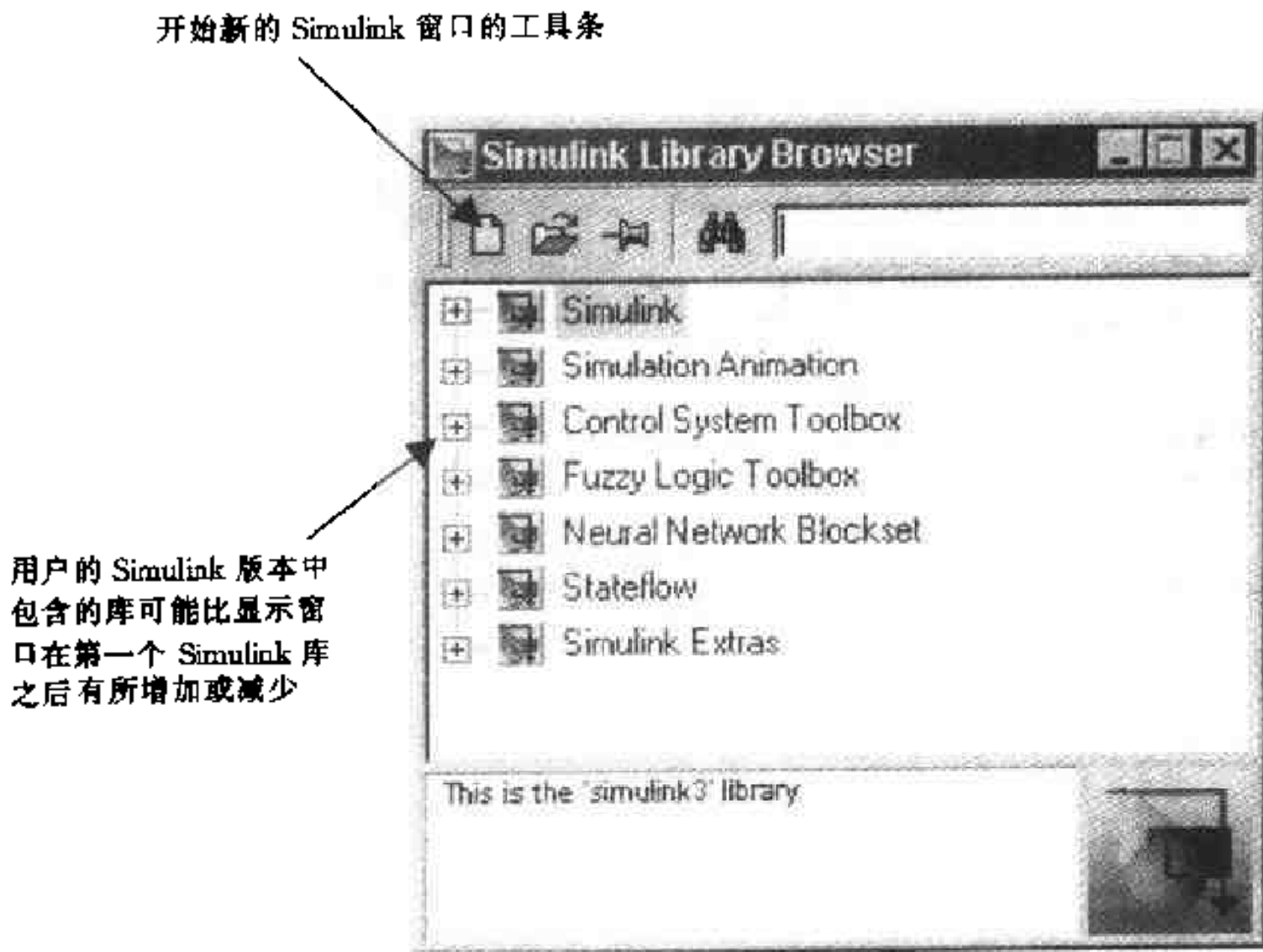


图 A-2 Simulink 库浏览窗口

A.2 一个简单模型的建立

现在,点击“New”图标,并点击 Simulink 窗口中的“+”图标,就展开 Simulink 的模块库,会得到如图 A-3 所示的屏幕窗口。

现在可以准备建立我们的第一个仿真模型。因为计算机仿真的核心是数值积分,所以,我们首先以积分一个简单信号作为仿真建模的起点。仿真结构将包括一个输入(或在 Simulink 术语中称为信号源),两个积分器,一个我们可以观看信号随时间变化规律的窗口(一个“Sinks”)。

现在,回到 Simulink 库浏览窗口(即启动 Simulink 任务时出现的窗口)。展开“Source”库,找到 Step 模块。点击该模块用户会看到在窗口底部出现一个关于该模块的简单介绍。用鼠标点击该模块,并一直按住鼠标将该模块拖到空的窗口。当将该模块拖到窗口时,松开鼠标按钮,在窗口就会出现一个 Step 模块的拷贝(见图 A-4)。

类似地,可以在“Continuous”库下找到 Integrator 模块,并将一个积分器移到窗口中来。在仿真中我们需要两个积分器,但这里仅从库中拖出来一个。一旦在

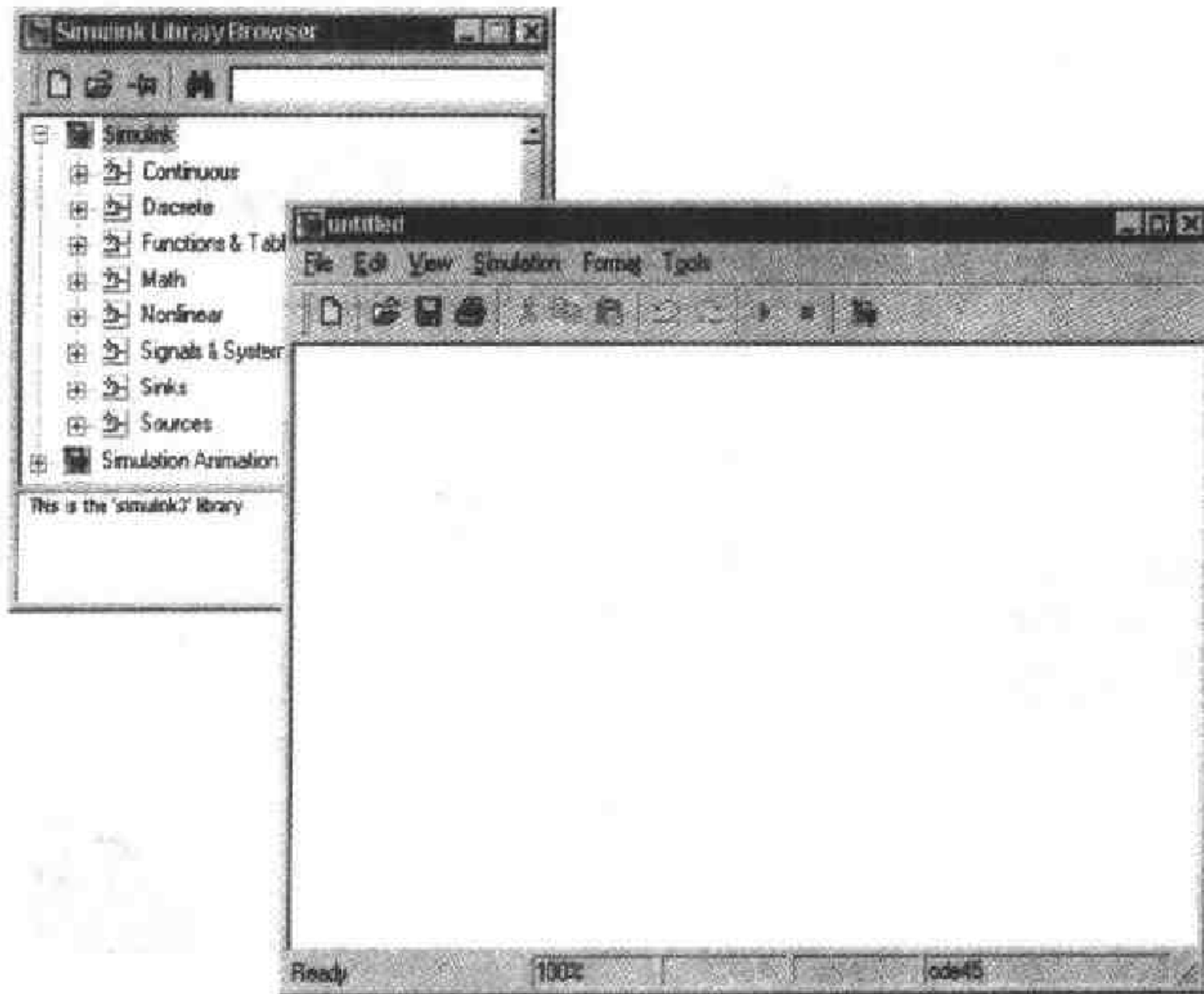


图 A - 3 未加标题的新 Simulink 模型的模型窗口

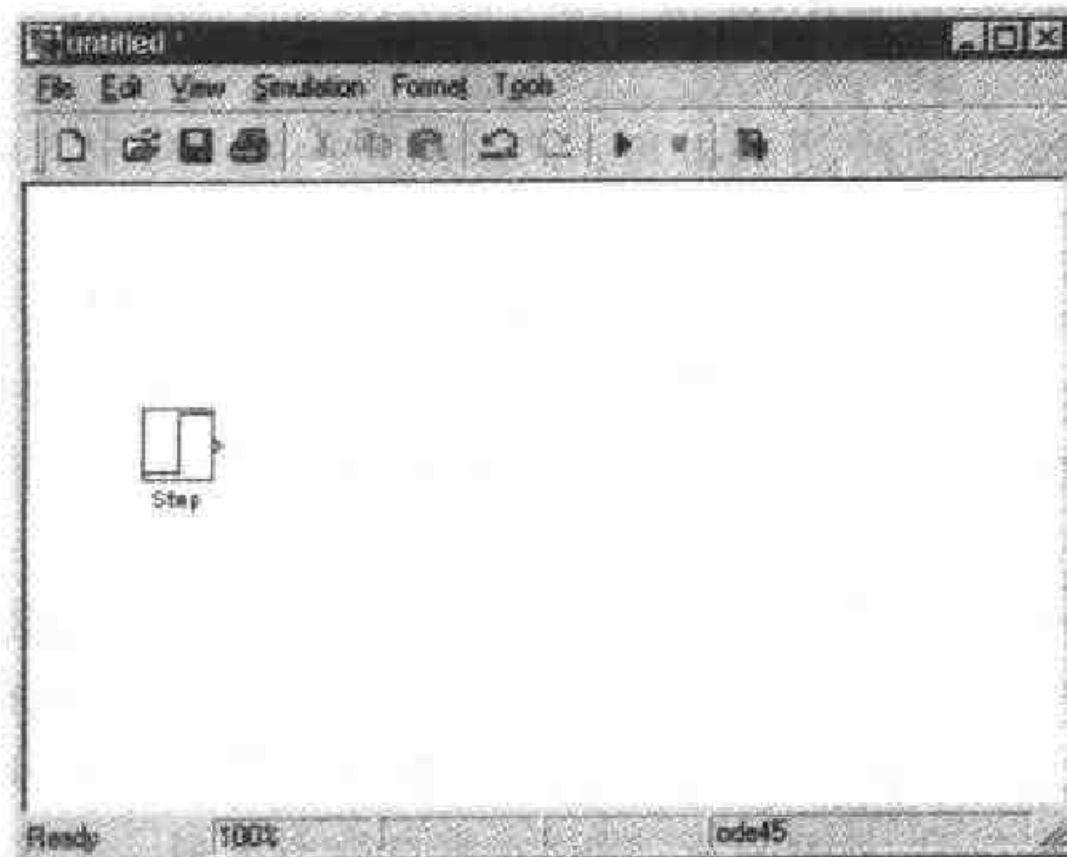


图 A - 4 以将 Step 模块拖到新的模型窗口为起点开始建立 Simulink 模型

Step 函数旁边有了一个积分器,只需将鼠标移到该积分模块并按鼠标右键,在窗

口内将该模块拖到右边一点的位置,然后松开鼠标按钮,就会看到两个积分器。这是 Simulink 非常有用的一个性质:在任意一个模块处按住鼠标右键并将模块拖到一个新的位置,就可以获得该模块的一个拷贝。

从“Sink”库中,找到 Scope 模块并将之拖到窗口。此时,用户的仿真窗口将如图 A-5 所示。

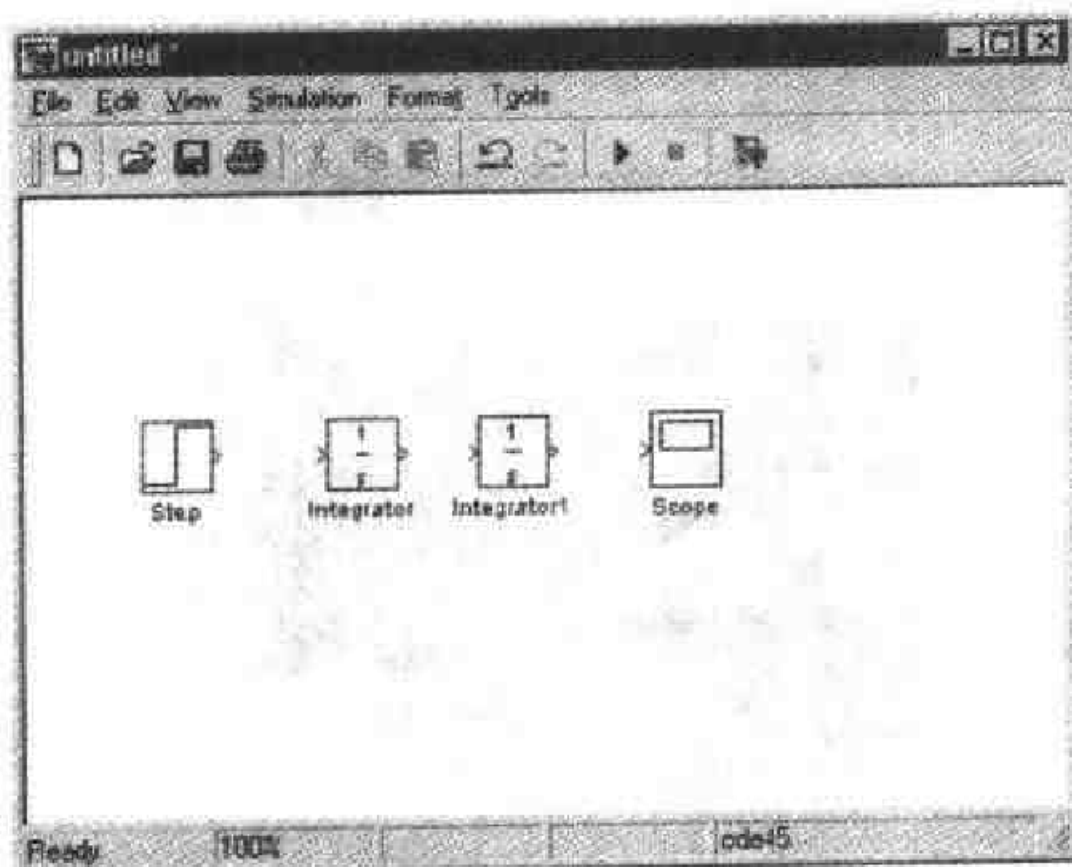


图 A-5 继续建立仿真模型:增加两个 Integrator 模块和一个 Scope 模块

连接这些模块以完成模块图。用鼠标拖着连接线从一个模块移到下一个模块将二者连接起来(与用户在画图程序中画一条线大体类似)。最后,仿真模块如图 A-6 所示。

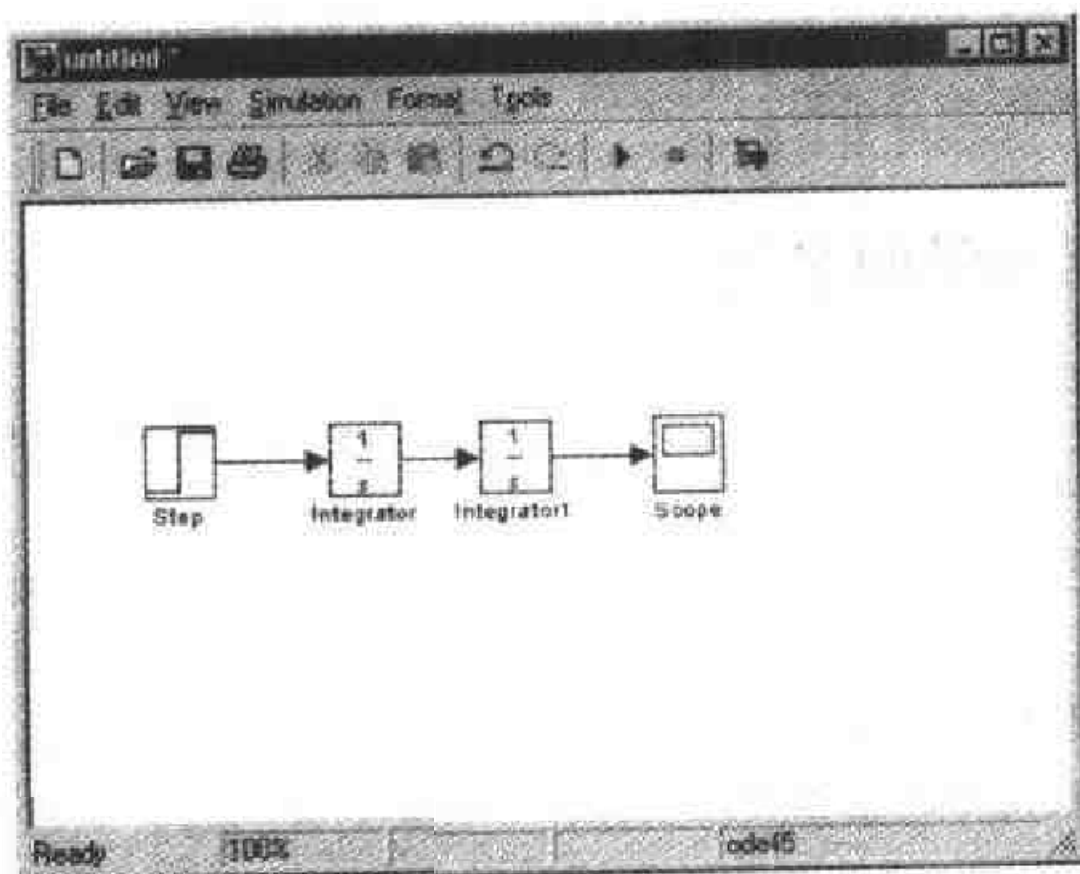


图 A-6 通过画连接线连接模块

A.3 仿真运行

虽然仍有更多的细节需要考虑,但至此我们已经给出了仿真所必需的最少环节。双击 Scope 图标并点击 Simulink 窗口的运行按钮,Simulink 的运行按钮看起来像 CD 播放机上的播放按钮。几秒钟后,仿真结束并发出逐渐减弱的嘟嘟声,显示窗口如图 A-7 所示。

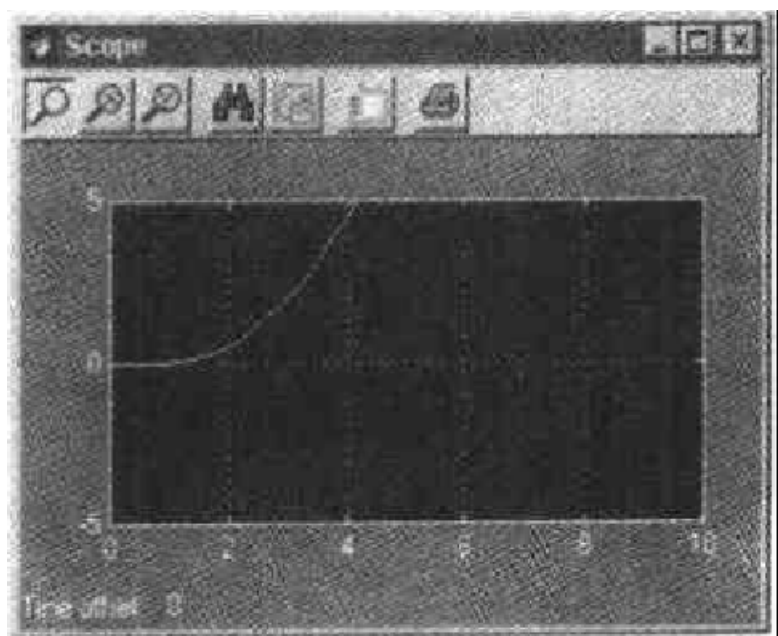


图 A-7 第一个仿真运行的 Scope 模块输出

从此图中可以观察到以下几点:首先,Simulink 默认的仿真运行时间是 0~10 s;其次,第二个积分器的输出信号是一个二次增长的信号,但只有仿真运行 1s 后才开始表现出这种特性;最后,Scope 模块缺省的尺度(± 5)不足以显示整个信号。

现在我们将研究在 Simulink 和 Simulink 模块中不同的设置以及这些不同的设置如何影响仿真的运行结果。

A.4 仿真运行时间参数

在仿真窗口(如图 A-6 所示),点击 Simulation 菜单选择 Parameters,将会出现如图 A-8 的对话框。

注意,在这里用户可以选择仿真的开始和结束时间。除了对那些含有显式时间函数的仿真外,对大多数物理仿真来说,所用的起始时刻实际值和仿真没有关系。通常的作法是让仿真从 $t=0$ s 开始,而不是从某个别的时刻开始。

这个窗口也被用作调整参数以便告诉 Simulink 如何进行数值积分,而这一点恰恰是仿真的核心。Simulink 缺省采用对经典 Runge-Kutta 法改进后的变步长四阶/五阶显式 Runge-Kutta 积分法。该方法在积分过程中调整积分时间步长,使

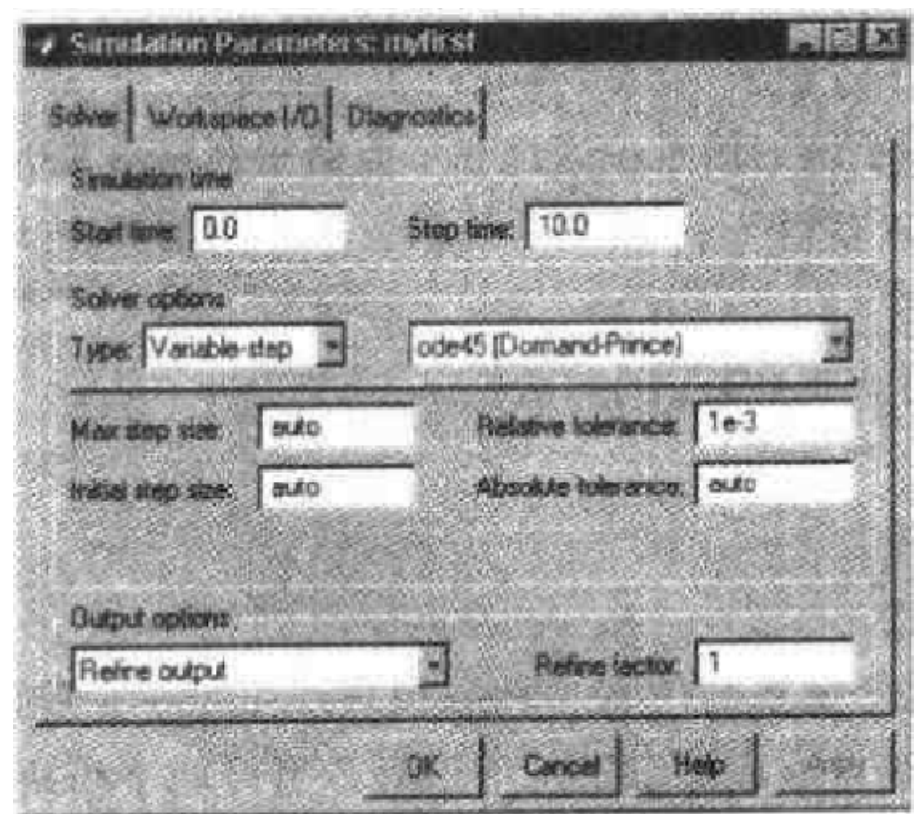


图 A-8 Simulation 对话框——Parameters 菜单选项

得积分误差估计低于某个误差限。Simulink 缺省的相对误差(误差相对于积分输出的比值)限是 0.001(1%或 0.1%)。绝对误差限由算法在积分过程中自动调整。

对于在本书中所讨论的仿真来说,应当非常严格地控制相对误差限,比如令相对误差限达到 $1E-6$ 或 $1E-7$ 。现在在窗口中进行相应的修改。类似地,改变最终的运行时间到 5 s,然后单击 OK 按钮关闭窗口。如果现在运行仿真,将会只显示 5 s 的仿真时间并画出相应的信号。

现在返回到仿真模型,双击“step input”打开该模块,会看到如图 A-9 所示的窗口。

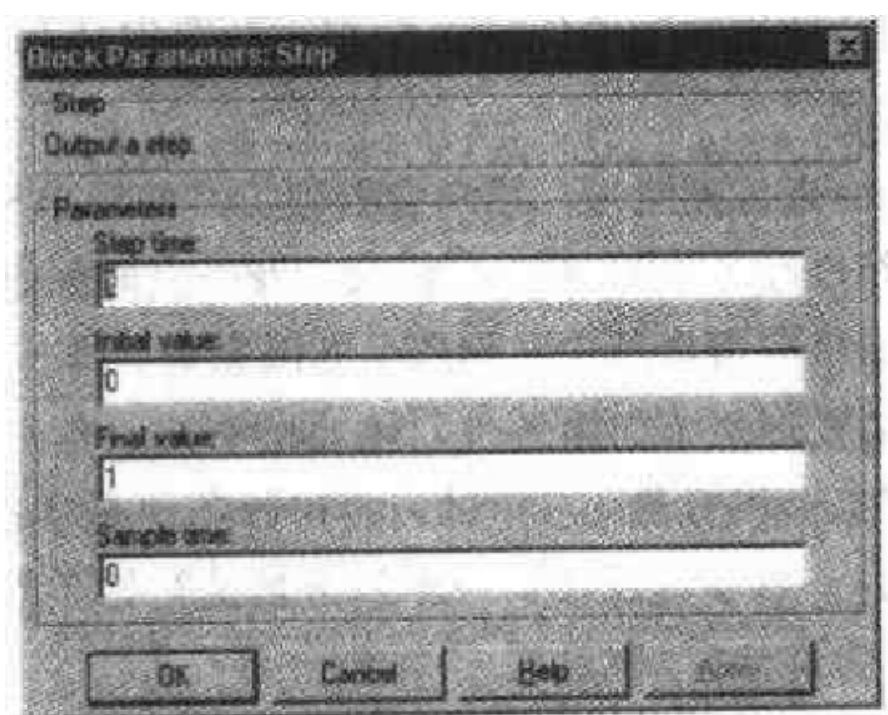


图 A-9 显示 Step 模块选项的对话框

至此我们明白了为什么在仿真开始 1s 后仿真的输出才开始改变,这是因为从 0 到 1 s 积分器接受了零输入。Step 模块产生了在输入到“step time”中的时刻从初始值阶跃变化到终了值的信号。通过在示波器中观察信号的波形或画出信号随时间的变化波形,可以获得 Step 模块的波形。Step 模块的波形是一个阶跃波,模块名称由此得来。

改变阶跃时间为 0.0 s 并关闭对话框。

A.5 初始条件

现在回想一下这个简单的仿真。Step 模块产生了一个阶跃信号,该信号从 0 开始,并当仿真一开始就变成了等于 1 的信号。另外两个模块对该信号进行时间积分,这两个积分的结果被画在模拟示波器的窗口中。从数学上讲,上述过程可以表示为

$$y(t) = \int_0^t \int_0^{\tau} u(\sigma) d\sigma d\tau$$

注意到为了和微积分规则一致,这里采用了虚变量。假设被积函数在积分过程中保持常量 1.0,我们可以对该积分求出下面的解析结果

$$y(t) = \frac{1}{2}t^2 + C_1t + C_2$$

其中: C_1 和 C_2 是积分常数。从数值积分的角度讲,这些积分常数具有重要的物理意义,它们是积分器计算的初始条件。换句话说,它们是信号在 $t=0$ s 时的值。所有的仿真需要使用者给出仿真中使用的积分过程(或积分器)正确的初始条件。Simulink 中的积分器的缺省初始条件是 0.0,但正如我们马上将看到的那样,我们可以修改这些初始条件。

现在运行仿真并检查仿真结果。用户会看到仿真的输出从 0 s 的时刻开始变化,这是由于我们已经修改了“step input”模块。花一点时间研究一下图 A-10 给出的输出结果,这个结果应当和用户刚刚产生的图完全一样。

考察该图,我们能发现两个初始条件对输出的影响。信号在初始时刻的值为零这一事实反映在第二个积分器(右边的)的初始条件是零;信号在 $t=0$ s 时的斜率(或导数)为零这一事实反映在第一个积分器的初始条件也是零。双击右边的积分器并考察弹出的对话框,如图 A-11 所示。

在这段时间内,我们只对初始条件感兴趣。改变该积分器的初始条件为 1,点击 OK,再运行仿真,会出现如图 A-12 所示的图形。

注意到信号现在偏离了一个等于初始条件的量,但信号的斜率仍然是零(因

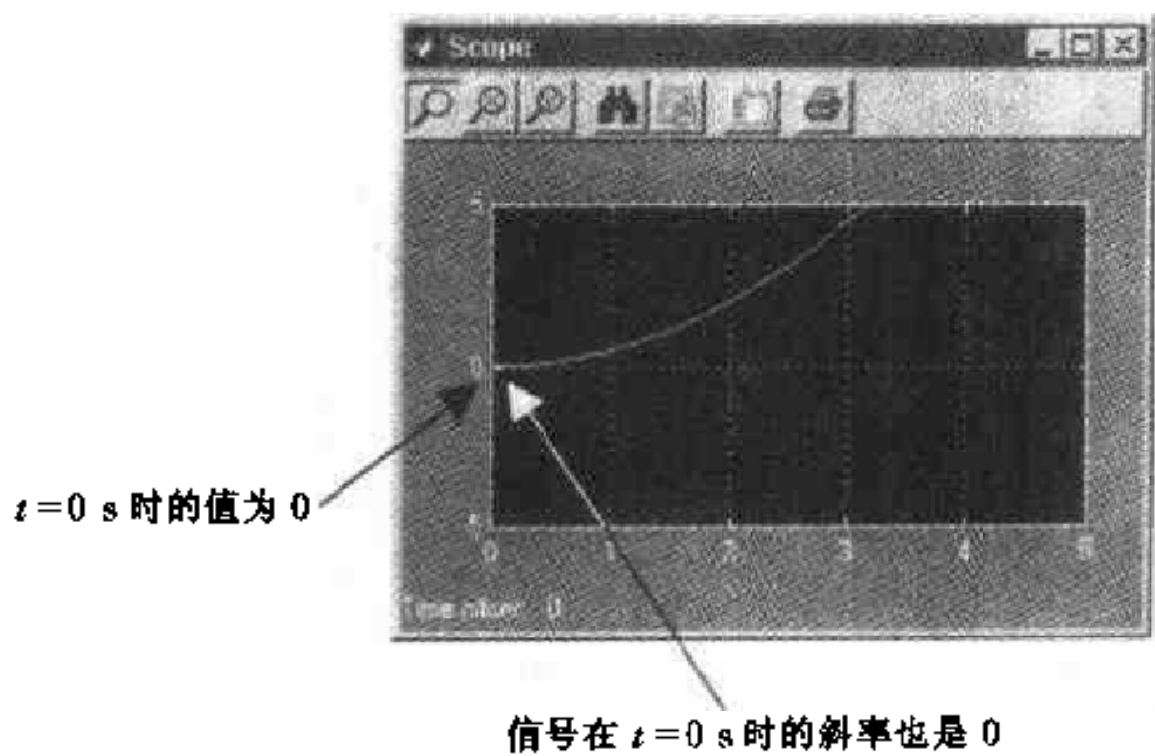


图 A-10 对应零初始条件和阶跃时间 0.0 s 的 Scope 模块仿真结果输出

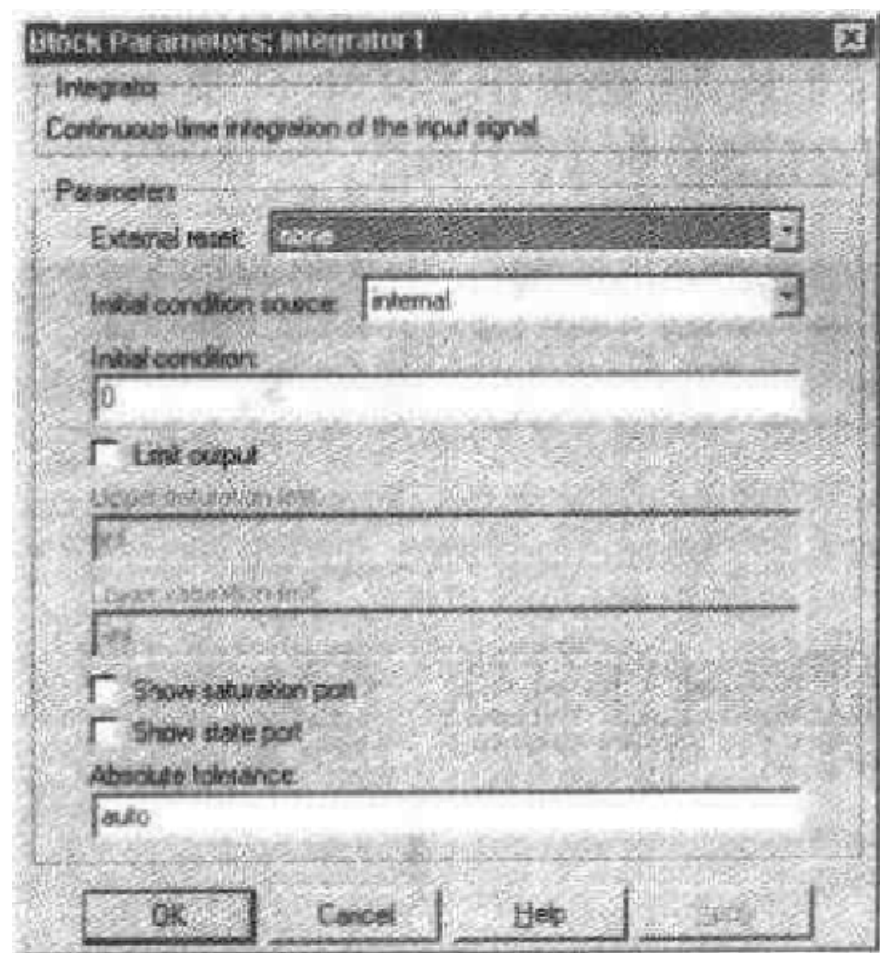


图 A-11 Integrator 模块的对话框

为第一个积分器的初始条件没有改变)。鼓励学生对两个初始条件都进行实验,以确信对初始条件的影响有一个透彻的理解。

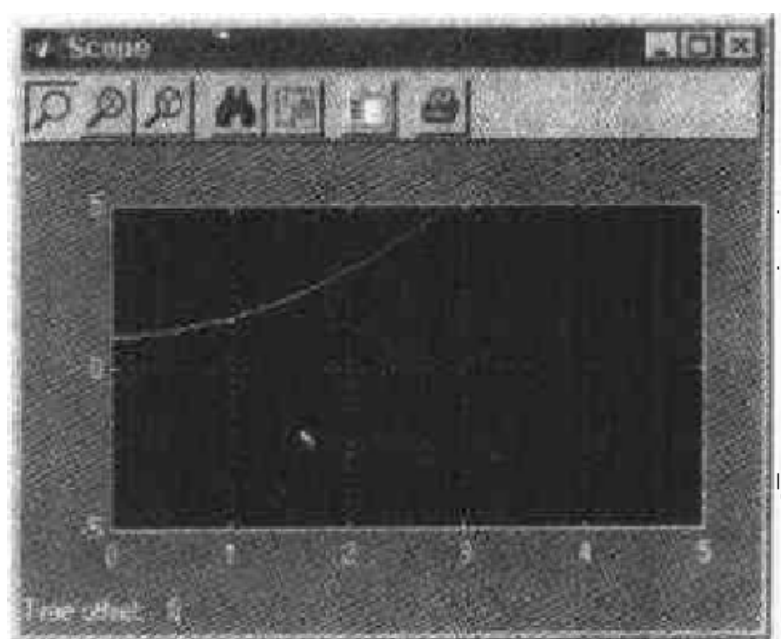


图 A-12 第二个积分器的初始条件被改变后仿真运行的 Scope 模块

A.6 多路信号

现在,我们对前面的仿真再作进一步的修改以显示 Simulink 更多的特性。返回到 Simulink 库浏览窗口,在“Signals & Systems”标题下找到 Mux 模块并将它拖到仿真窗口。选取连接最后一个积分器和 Scope 模块的连接线,点击删除键删除该连接线。现在,用户得到的仿真窗口如图 A-13 所示。

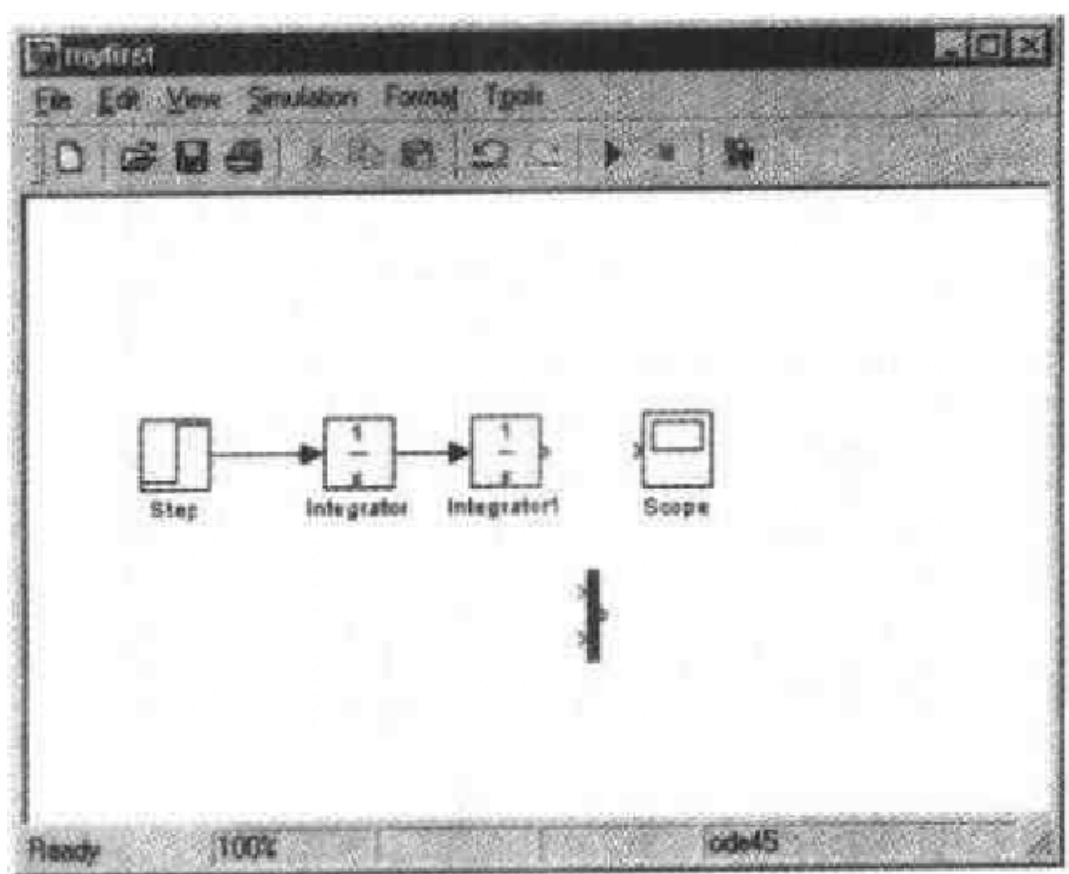


图 A-13 插入 Mux 模块并删除一条连接线

Mux 模块是一个多路转换器,该模块接受多路信号并将它们合并为一路信号,这是我们在许多场合都非常有用的一个特性。用户可以通过双击鼠标利用对

对话框来改变输入信号数。现在就改变输入信号数,将它变成一个三路输入的多路转换器。将仿真模块重新布置,使 Mux 模块位于第二个积分器和 Scope 模块之间,将第二个积分器和 Mux 模块连接并作为第一路输入,将 Mux 模块的输出和 Scope 模块连接起来。这样仿真模型应当如图 A-14 所示。

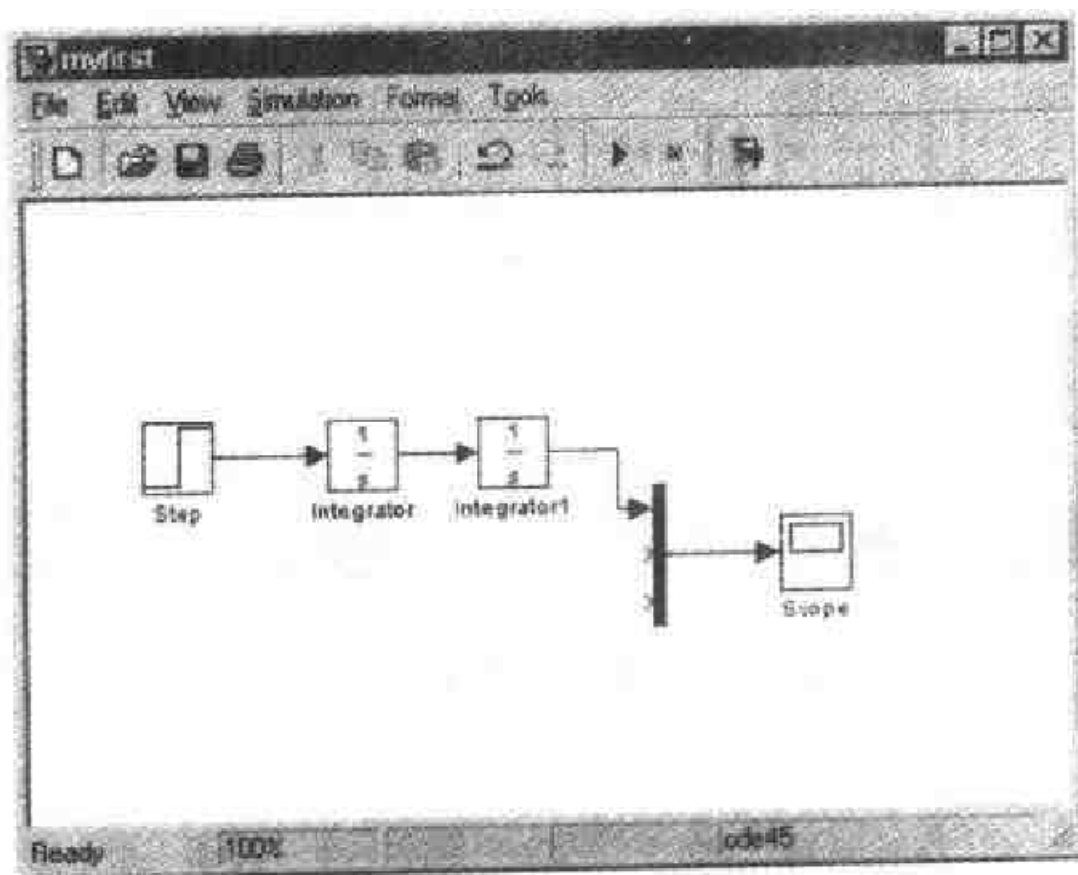


图 A-14 扩展 Mux 模块以便接受三路输入信号并开始连接

为了完成修改,我们将“引出”两个已有的信号并将它们画线连到 Mux 模块的输入端。要引出一个信号线,将光标移到该线并击右键 (Mac 使用者需键入命令),然后如同用户通常那样画出一条新线来。窗口显示在原信号线和引出线 (与交叉非连接线不同) 的接头处有一个很小的实心分线盒。仿真模型修改完成后应如图 A-15 所示。

将初始条件重新设置为零,运行仿真并在 Scope 模块中观察仿真结果 (见图 A-16)。

注意到三条线 (在屏幕上表现为不同的颜色) 表示输入到 Mux 模块的三个信号。水平线是第三个输入,阶跃信号在仿真开始就从 0 变到 1 并在整个仿真过程中保持不变。从零开始当仿真运行 5 s 后数值终止在 5.0 的水平线是 Mux 的第二个输入,也是第一个积分器的输出。稍加考虑即可以看出这是对常量积分的一个恰当的反应。最后,我们注意到第二个积分器的结果和我们见到的最初的仿真结果是完全一样的。

到此为止,鼓励学生调整初始条件、时间步长以及阶跃输入的大小,研究改变这些因素对仿真结果的影响,直到对这些因素的影响有一个深刻的理解。

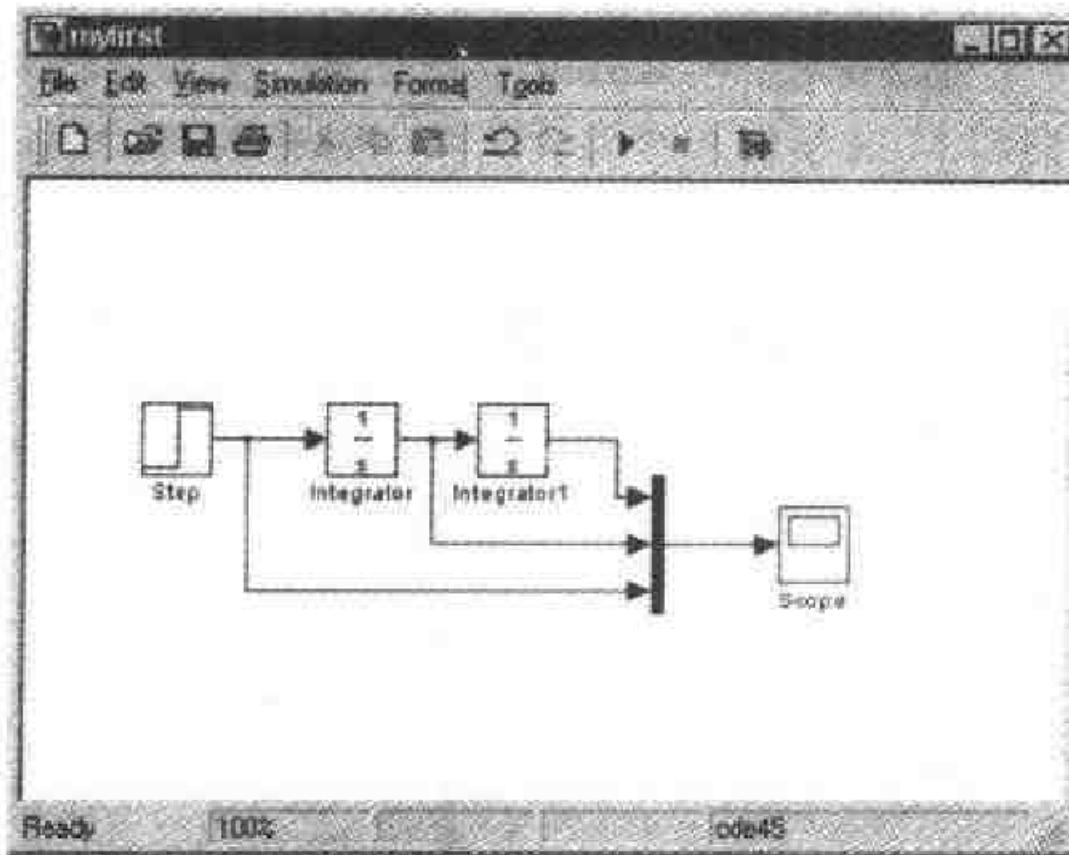


图 A - 15 能够同时观看所有信号修正的仿真结构

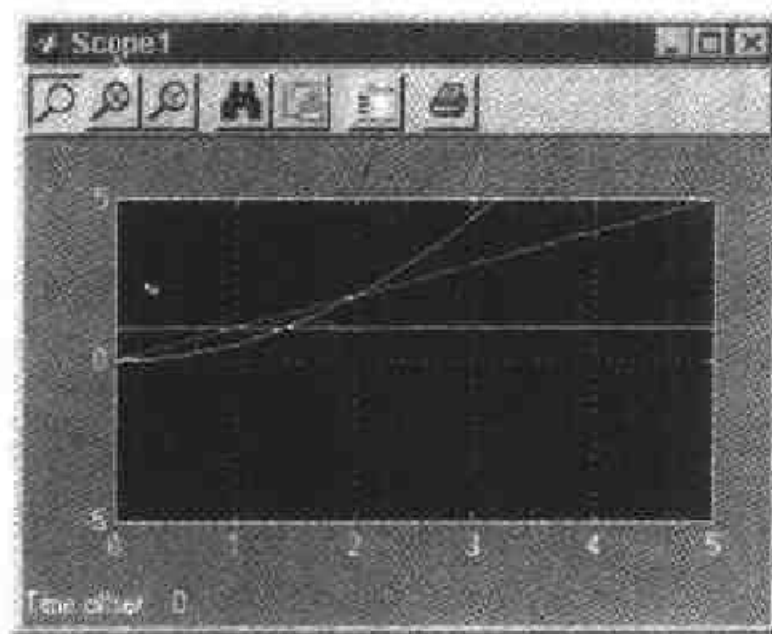


图 A - 16 同时显示三个信号的 Scope 模块

A. 7 Simulink 和 MATLAB:返回数据到工作环境

Simulink 最强大的功能之一在于它是 MATLAB 环境的一部分,并且可以和 MATLAB 环境无缝地交换数据。为了研究 Simulink 和 MATLAB 工作环境可能的联系,在我们的例子中以 To Workspace 模块(也可以在“Sinks”库中找到)代替 Scope 模块,新的仿真结构如图 A - 17 所示。

To Workspace 模块一般将多路信号作为输入,在 MATLAB 环境中生成一个

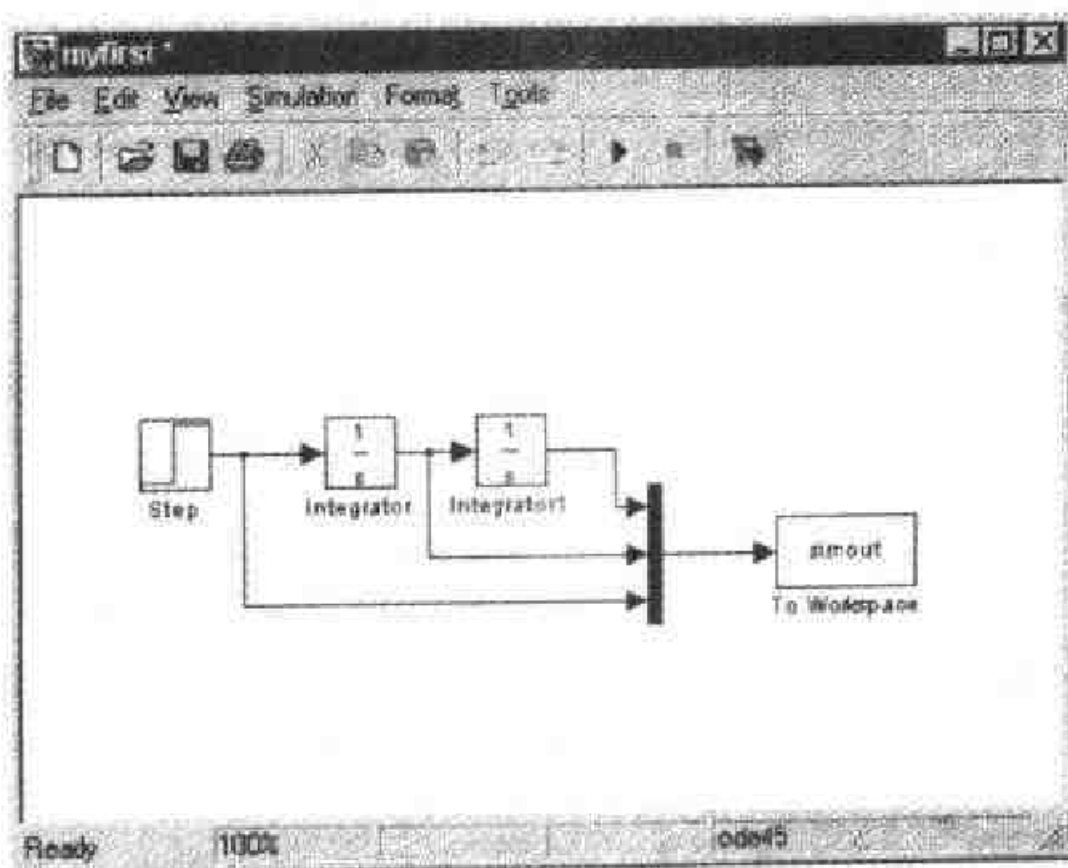


图 A-17 以 To Workspace 模块代替 Scope 模块(可将仿真结果直接存储于 MATLAB 环境(文件:myfirst.mdl))

矩阵。矩阵的每一列(原文该处误为每一行——译者注)代表在给定时间步的输入变量值。变量在矩阵中出现的顺序与它们连接到 Mux 模块的顺序一样。换言之,图 A-17 中第二个积分器的输出与 Mux 模块最上端或第一个输入连接。因此,工作环境矩阵的第一列将是最右端积分器的输出。

现在双击 To Workspace 模块,考察弹出的图 A-18 所示的对话框。

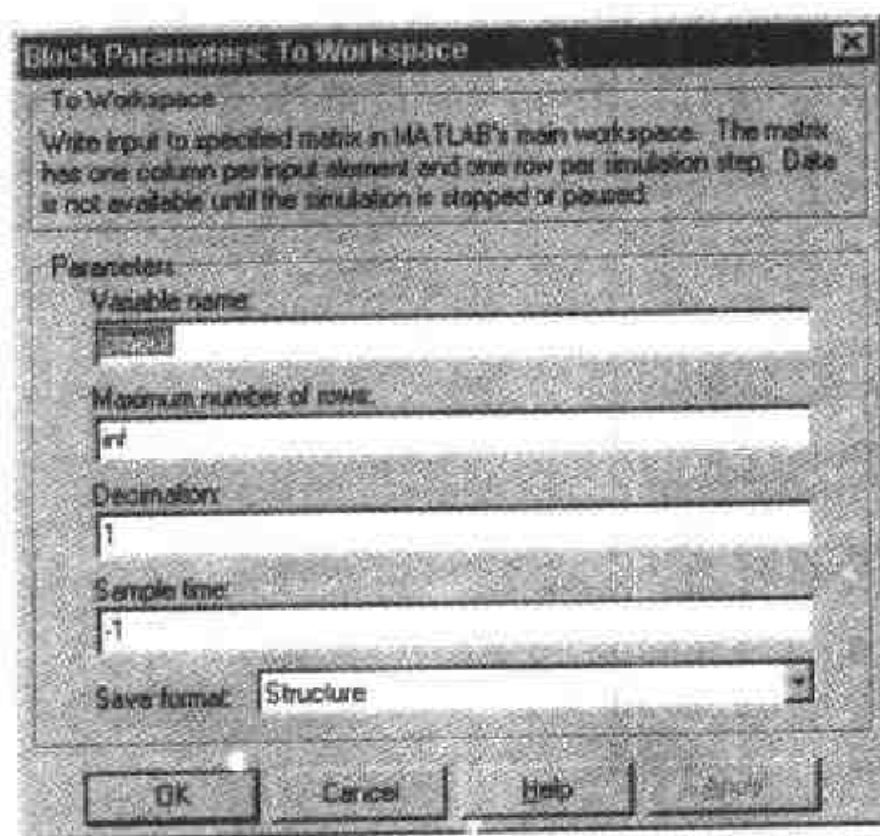


图 A-18 To Workspace 模块的对话框(允许使用者设置存储于工作环境中的变量名)

所起的变量名应当使用户便于识别和管理 MATLAB 环境中的矩阵,任何高质量的 MATLAB 矩阵名都应当这样。现在,仍保留默认的矩阵名 Simout。然而更重要的是需注意对话框底部的“Save format”,默认格式(对 R11 的 MATLAB/Simulink)是“Structure”。这种格式以一种非常紧凑并很完整的数据结构存储数据。遗憾的是,对这些数据进行处理比较烦琐,除非用户掌握的数据结构知识非常全面扎实。对于大多数使用者来说,该对话框中的另外一个选项“Matrix”更为合适。在图 A-18 所示的对话框中作此更改并运行仿真。

因为我们已经删除了 Scope 模块,所以没有对仿真运行过程的显示,但在仿真结果时可以听到轻微的嘟嘟声。现在进入 MATLAB 命令窗口(当用户开始启动 MATLAB 时出现的窗口),键入列出存储在工作环境中的矩阵的命令,whos。

```

> whos
Name          Size          Bytes  Class
simout        53x3           1272  double array
tout          53x1            424  double array
Grand total is 212 elements using 1696 bytes
>

```

MATLAB 从第二个矩阵 tout 开始显示在工作环境中的两个变量(如果用户在执行该命令之前进行其他操作就可能会有更多的变量)。tout 矩阵有 53 行 1 列,是 Simulink 模型产生的默认矩阵,列出了每一个积分步的时间,这一点是非常重要的,因为 Simulink 一般采用变时间步算法,因此数据存储的实际时间在预先是不知道的。

另一个变量 simout,由仿真中的 To Workspace 模块产生。注意该矩阵是 53 行 3 列的矩阵,这与 Mux 具有三个输入变量的事实一致。为了证实这一事实,我们看一下存储在矩阵中的以下数据,最容易的方法是在命令行键入矩阵的名字。

```

> simout
simout =
      0          0    1.0000
 0.0000    0.0000    1.0000
 0.0013    0.0505    1.0000
 0.0051    0.1010    1.0000
 0.0202    0.2010    1.0000
 0.0453    0.3010    1.0000
 0.0804    0.4010    1.0000
 0.1255    0.5010    1.0000

```

如前所述,第一列数据是第二个积分器的输出,是一个二次增长的变量。第二列是积分器之间的信号,是一个线性增长的斜波函数。第三列是输入变量。

A.8 使用 MATLAB plot 命令

plot 命令是 MATLAB 中功能强大并且非常灵活的一个命令。关于该命令的完整详尽的讨论超出了本入门辅导的范围。然而,给出几个适当的例子将是非常有帮助的。下面的命令将画出第二个积分器的输出随时间的变化曲线。

```
>> plot(tout, simout(:,1))
```

这是 plot 命令最简单的形式,对上述语句进行一些说明是必要的。“:”操作符在 MATLAB 中具有特殊的性质。在这里,该操作符选取了第一列的所有元素。理解这一点的最好的方法是将“:”操作符理解为“所有”。因此,上述 plot 命令可以理解为“画出 simout 的第一列的所有行随 tout 的变化曲线”。由该命令得到的图由图 A-19 给出,用下面的命令将该图插入文件中:

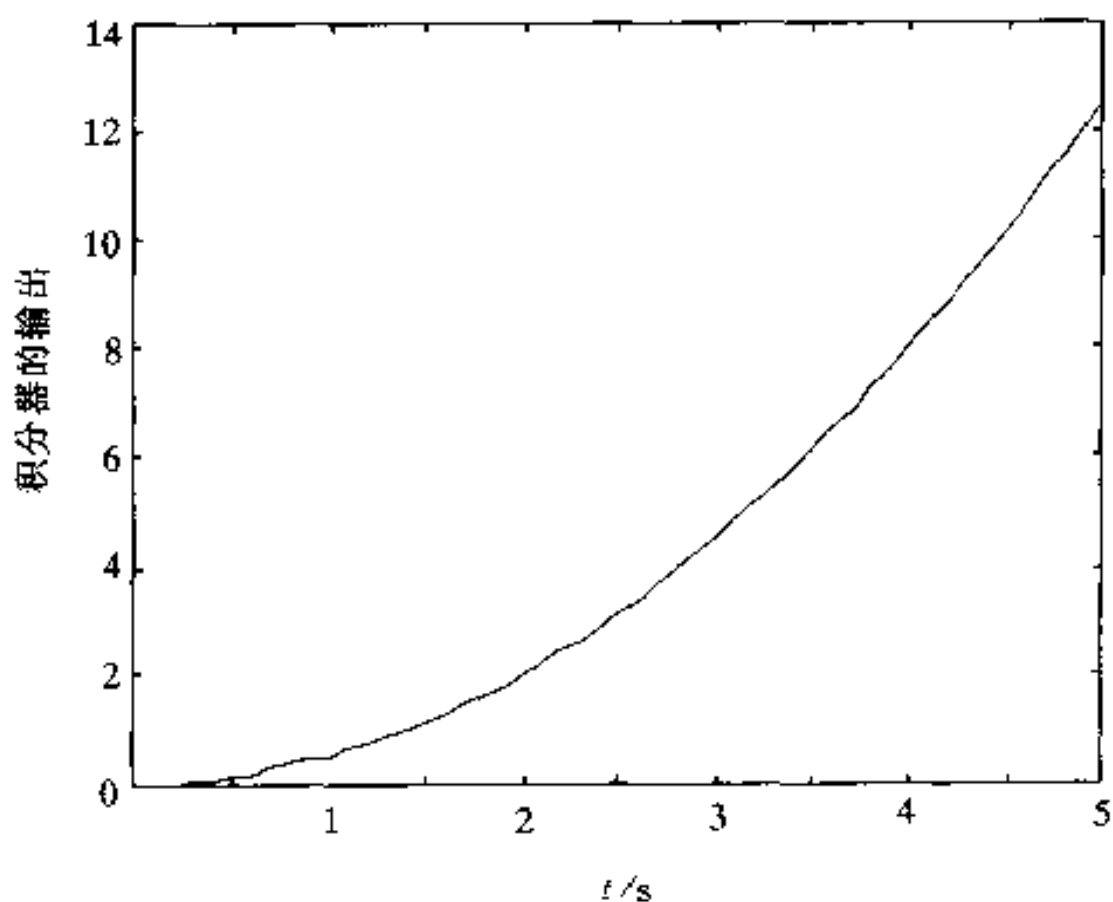


图 A-19 使用 MATLAB 中的 plot 命令画出积分器输出随时间的变化曲线

```
>> print -dmeta
```

该命令将图形窗口以“windows 图元文件”的格式拷贝到剪贴板。也可以用下面的命令得到如图 A-20 所示的多组数据曲线:

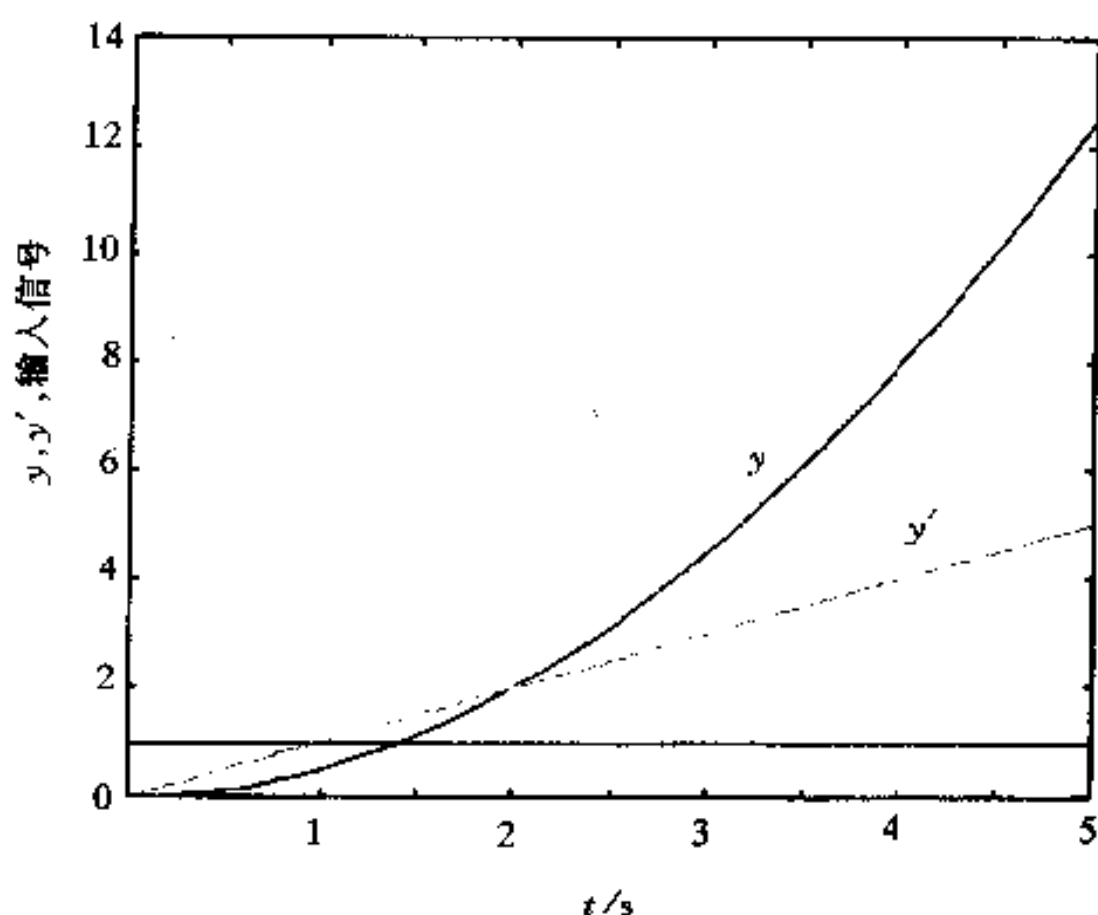


图 A-20 同时画出 simout 矩阵的所有三列随时间的变化曲线

```
>>plot(tout,simout(:,1),tout,simout(:,2),tout,simout(:,3))
```

最后,我们将显示可以使用不同的标志来标识数据(与实线相反)来改变曲线的外观,见图 A-21。

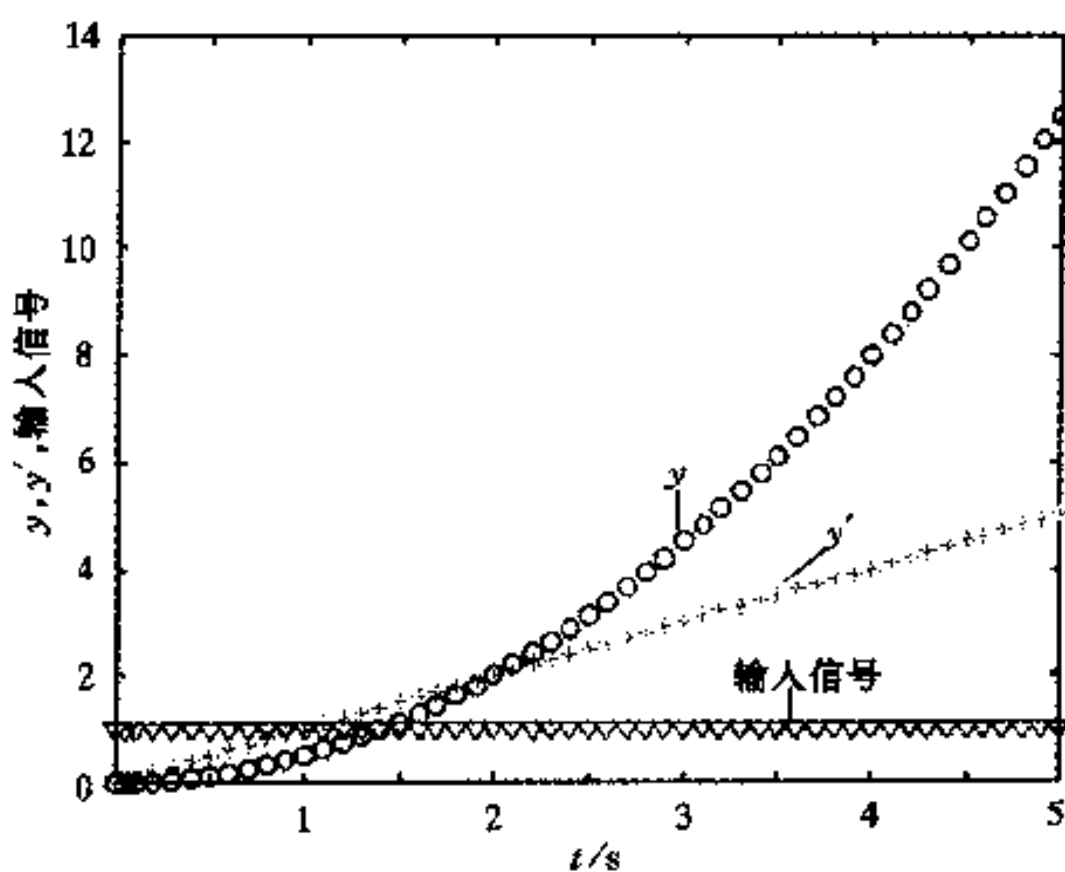


图 A-21 使用 plot 命令以符号标识每个数据点与内插实线形成对比

```
>>plot(tout,simout(:,1),'o',tout,simout(:,2),'+',tout,simout(:,3),'v')
```

关于 plot 命令更多的信息,可以在命令提示符键入 help plot 而得到。

A.9 在 Simulink 中使用 MATLAB 函数

这里给出的方法是利用一个功能强大的特性来建立仿真,即将 MATLAB 环境的可编程能力和 Simulink 所采用的高度可视化方法相结合。MATLAB“Functions & Tables”库中的 Fcn 模块,可以被用来将使用者编写的任何 MATLAB 函数文件插入仿真。这一方法也许最好通过一个例子来加以说明。注意本节所采用的例子——线性弹簧质量阻尼系统的响应,可以用解析法求解或用其他非 MATLAB 函数来进行仿真。然而,我们之所以选择这个例子是因为该例也是众多工程领域所熟知的。

A.9.1 MATLAB 函数

我们从回顾函数的概念开始讨论,因为函数是和 MATLAB 紧密相关的。在 MATLAB 中,使用者可以建立由一系列 MATLAB 操作构成的文本文件并在命令提示符调用它们。这些文件被称为 m 文件,因为其文件的扩展名使用了 .m,可以具有脚本文件或函数的形式。脚本文件是 MATLAB 命令的简单组合,命令的重新执行就像从命令提示符键入这些命令一样,任何存储在 MATLAB 工作环境的变量都可以被脚本文件使用。相反,函数在自己的临时工作环境运行。信息以参数传递给函数,而一旦函数结束则信息返回到工作环境。

作为例子,考虑阶乘函数,该函数定义为如下的非负整数的乘积

$$f(x) = x! = \begin{cases} 1 & x = 0 \\ x \times (x-1) \times (x-2) \times \cdots \times 1 & x > 0 \end{cases}$$

我们现在将建立 MATLAB 函数来计算阶乘。从缺省的 MATLAB 命令提示符,点击新文件的工具条或者从 File 菜单选择 New→M-File。这样会打开一个新的 MATLAB 编辑和调试窗口,该窗口如图 A-22 所示。

现在输入以下这些原代码。注意到当用户输入时编辑器自动识别并用不同的颜色标识出用户的代码。注释(任何出现在 % 符号右边的语句)以绿色显示, MATLAB 关键词以蓝色显示,被怀疑的错误以红色标出。

```
factorial.m
function fact=factorial(u)
%
% factorial function used to demonstrate the use of
% function m-file in MATLAB
```

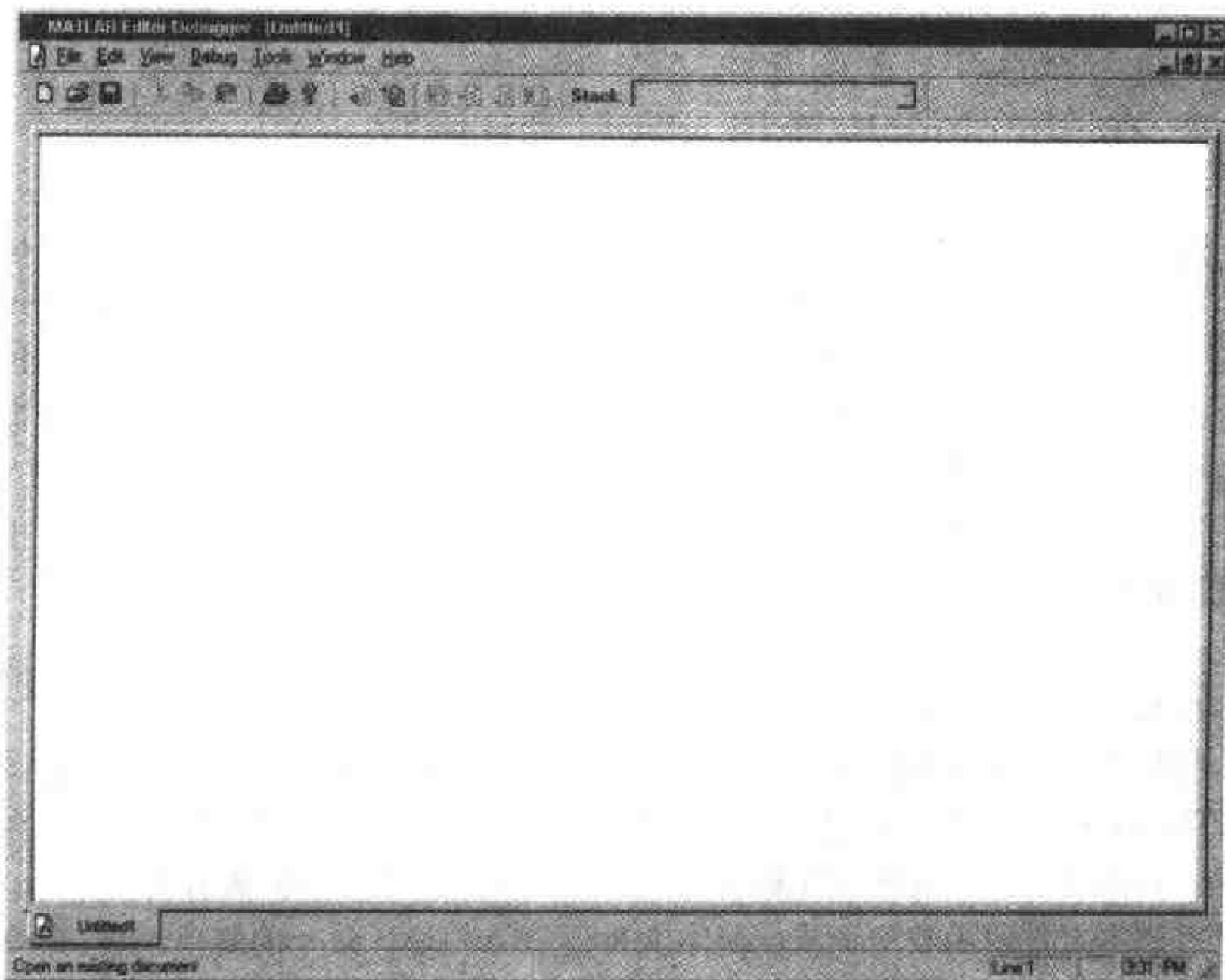


图 A-22 MATLAB 编辑/调试窗口

```

%
%
fact = 1;
if u>0
    for i=1:u
        fact=fact*i;
    end
end
end

```

m 函数最重要的部分并且是和脚本文件相区别的特征是函数首行的函数声明。关键词 `function` 表示该 m 文件是函数。在同一行的赋值语句定义了函数名 (`factorial`)，用来接受传递给函数的参数的哑元名 (`u`)，以及函数为调用语句返回值的变量名 (`fact`)。最后，注意到该文件以 m 文件存储起来，其文件名和函数名一样，即 `factorial.m`。

我们以此为例利用该函数来计算 5 的阶乘，利用阶乘的定义：

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

从 MATLAB 命令提示符键入：

```
> factorial(5)
```

```
ans =  
    120
```

```
>
```

注意到正确的答案,120,被存储于 MATLAB 环境中的一个变量 `ans` 中。下面的形式也是允许的:

```
> xfl=factorial(5)
```

```
xfl =  
    120
```

```
>
```

在此情况下,使用赋值语句调用函数并将该函数的返回值存储于一个新变量, `xfl`。

那些熟悉计算机编程规则的读者可以理解这里给出的赋值过程,但为了更清楚,我们简单地概述如下。赋值语句, `xfl=factorial(5)`, 引导 MATLAB 取值 5 并搜索名为 `factorial` 的函数。如果用户已经在缺省的目录下保存了 `factorial.m`, MATLAB 调用该函数,接受参数值 5,然后从 MATLAB 主工作环境分离出一个内存空间,产生了一个叫作 `u` 的新变量(和函数声明中定义的一样),并把数值 5 存储于该变量中,然后执行函数中的语句。在函数运行结束时,任何存储于局部变量 `fact`(也是在函数声明中给出定义)中的值被返回到工作环境并存储于变量 `xfl` 中。

一个非常重要的细节是在函数内使用的变量名是完全随意的。“外界”无法知道一个特定的函数是如何设计的或者变量是如何定义的。为了证明这一点,用不同的变量名重新编制阶乘函数如下:

```
function res=factorial(udiff)  
%  
% factorial function used to demonstrate the use of  
% function m-file in MATLAB  
%  
%  
res = 1;  
if udiff>0  
    for i=1:udiff  
        res=res*i;  
    end  
end
```

此函数和前面给出的函数完全一样。

A.9.2 从 Simulink 中调用函数

为了显示 m 文件和 Simulink 的结合,我们建立一个和前面例子类似的人为设计的简单例子。假设我们希望建立一个对任意输入积分两次的仿真,但在第一个积分器的输出(称之为 x_1)和第二个积分器的输入(称之为 \dot{x}_2)之间有一个代数函数,函数关系定义如下

$$\dot{x}_2 = 3x_1 - x_1^2$$

为了实现这个仿真,修改现有的仿真结构,在两个积分器之间插入一个 MATLAB 的 Fcn 模块(该模块可以在 Functions & Tables 库中找到)。需要注意的是可以通过选定一个已有的信号连接线并键入删除键来中止该信号的连接。

这样用户所建立的仿真模型应当和图 A-23 给出的类似。

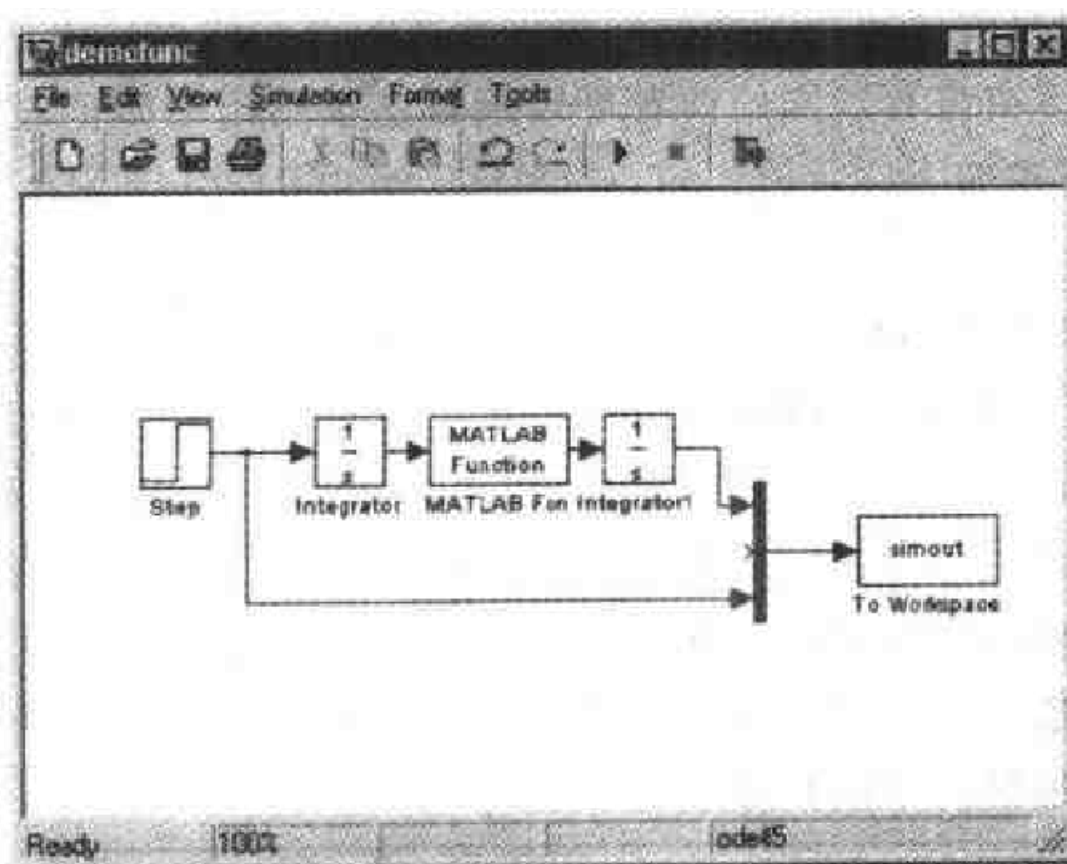


图 A-23 在两个积分器之间插入一个 MATLAB 的 Fcn 模块以显示其应用(文件: demofunc.mdl)

由于现在多了一个需要进行监测的信号(函数的输出),所以我们将 Mux 模块的输入增加到四个,并将函数的输入和输出均连接到 Mux 模块。

现在返回到 m 文件编辑器并打开一个新的文件。我们所需的函数是一个单输入、单输出函数以实现上面给出的多项式函数。注意实际中选择的变量名可以是任意的,但 MATLAB 惯例约定传递进来的参数为变量 u 。

实现这个函数的一个可能的 MATLAB 程序脚本如下:

```
mydemo.m
function x2dot=mydemo(u)
%
% function used to demonstrate the use
% of functions in Simulink
```

```
%
x2dot=3*u - u^2;
```

除注释之外,函数仅有两行:函数声明和实现代数关系的赋值语句。

下面,应当让 Simulink 模型知道该函数。双击 Fcn 模块弹出如图 A-24 所示的对话框。

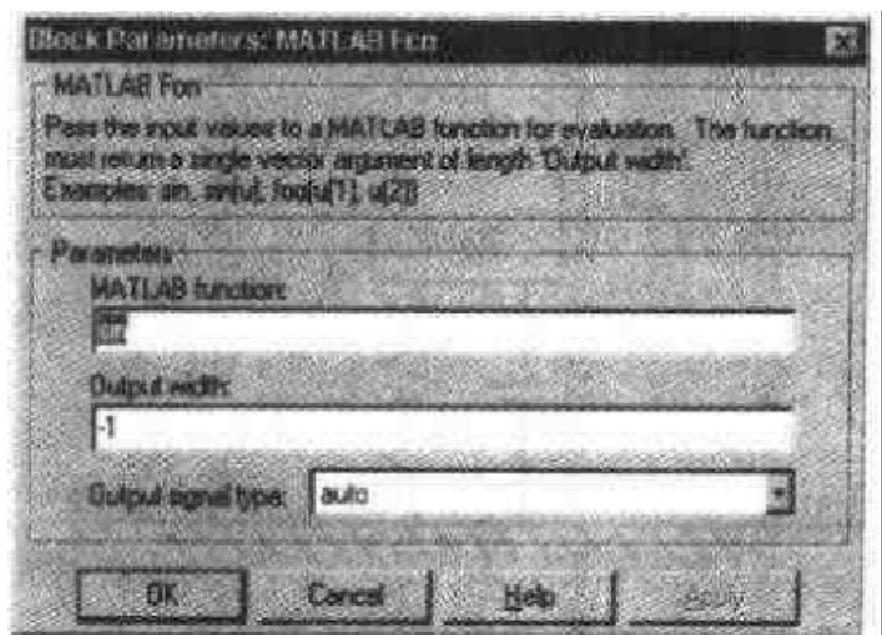


图 A-24 MATLAB Fcn 模块的对话框(可以自定义仿真调用的函数文件)

在 MATLAB function 空格处,输入存储函数的函数文件名(这里为 my-demo)。Output width(输出宽度)指返回的矩阵的大小。这里 1(因为我们返回一个标量)或-1(因为输入和输出具有相同的宽度)都可以。一般选择明确的宽度值,所以用 1。点击 OK 并返回仿真。

结果如图 A-25 所示。

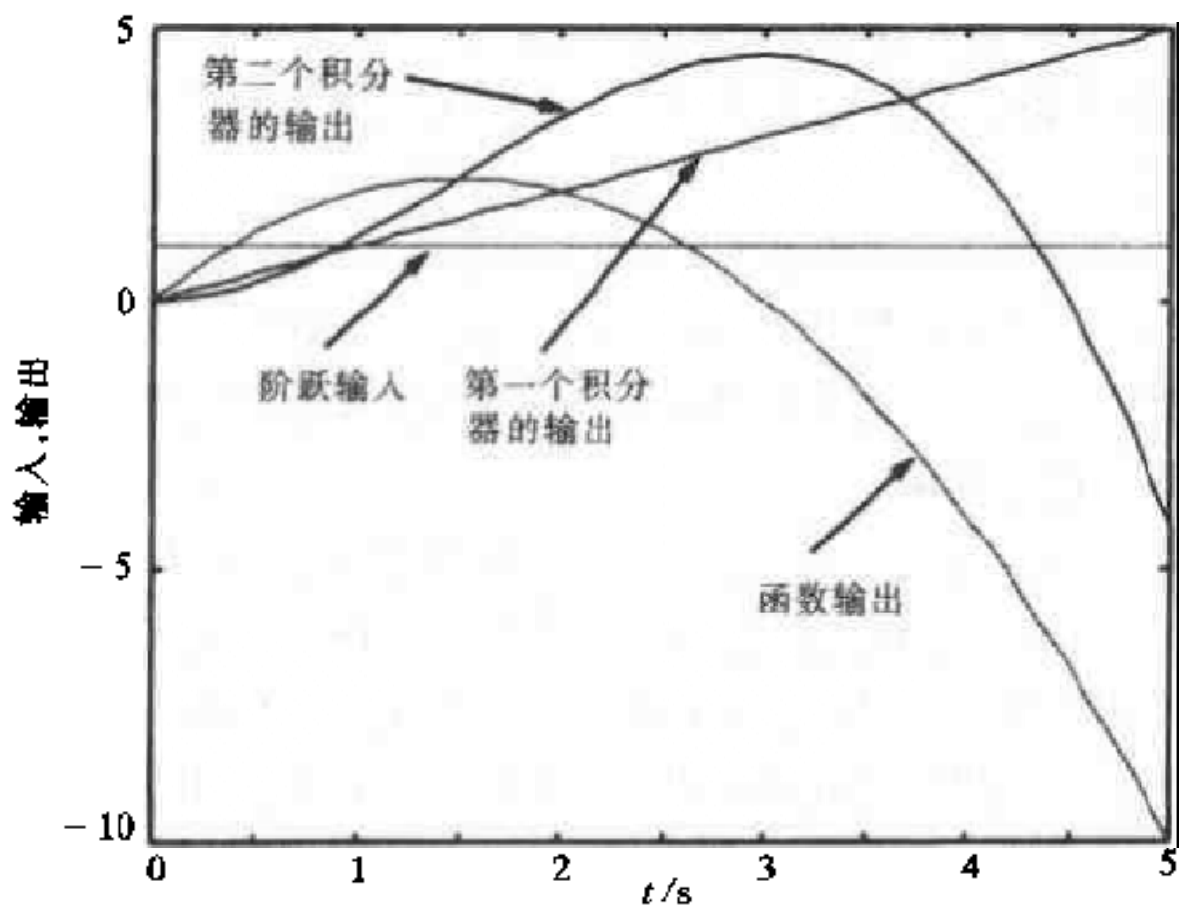


图 A-25 修正后的仿真中各个信号随时间的变化曲线

A.9.3 使用多输入和多输出

尽管前面所给出的例子有点人为设计的意思,但表明了 Simulink 仿真中嵌入 m 文件函数的方法。现在将讨论一个更具物理意义的例子。考虑图 A-26 所示的由集中质量、弹簧和阻尼器组成的系统。

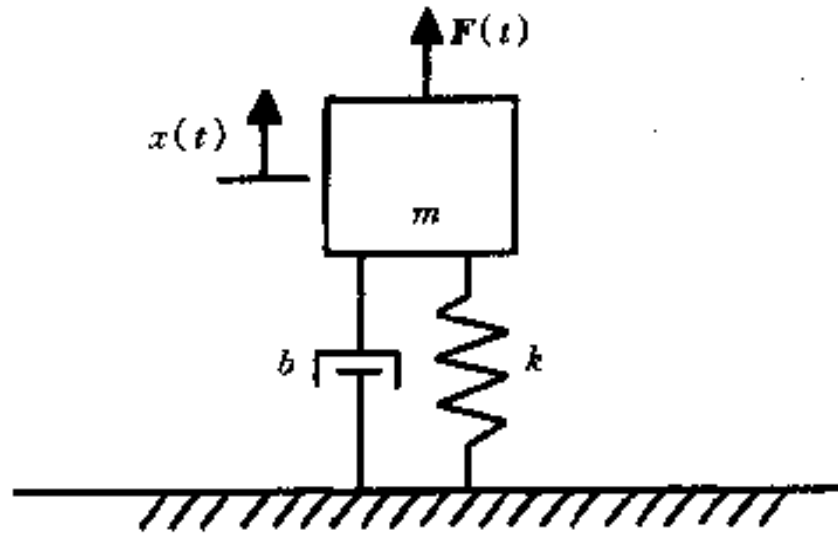


图 A-26 弹簧-质量-阻尼振动系统示意图

可以很容易地导出描述上述系统运动特性的微分方程

$$F(t) = m\ddot{x} + b\dot{x} + kx \quad (\text{A-1})$$

其中变量上的点代表关于时间的导数。

以求解系统的最高阶导数开始系统的仿真,如下所示

$$\ddot{x} = \frac{1}{m}(F(t) - b\dot{x} - kx) \quad (\text{A-2})$$

为了达到示范的目的,我们将对系统在任意输入 $F(t)$ 作用下系统的响应 $x(t)$ 以及阻尼元素所消耗的总能量感兴趣。能量消耗率(或功率)由阻尼力和质量的速度的乘积给出。因此

$$\dot{E}_{\text{diss}} = P_{\text{diss}} = b\dot{x}^2 \quad (\text{A-3})$$

为了得到 $x(t)$,在仿真时需要计算方程 A-1~A-3 给出关于 x 的二阶导数并对它积分两次。利用第一个例子中的两个积分模块,并将这两个积分器以如图 A-27 所示的情况进行连接。

请注意,我们已经为仿真增加了两个新的模块,一个 DeMux 模块和一个 MATLAB Fcn 模块。我们已经见过 Fcn 模块了,而对于 DeMux 模块,可以预言其功能与 Mux 模块相反,即接受一个矢量信号并将其分解为单一信号。DeMux 模块的第一个(上部)输出将假设为加速度,第二个为阻尼器所消耗的功率。方程 A-1~A-3 明确了该函数将需要利用输入的力、质量块的速度和位移。此外,消耗的能量应当被积分以计算出总能量。考虑上述因素,并增加一个 Mux 模块以便在工作环境中存储输出信号,仿真结构可以进一步修改成如图 A-28 所示的形

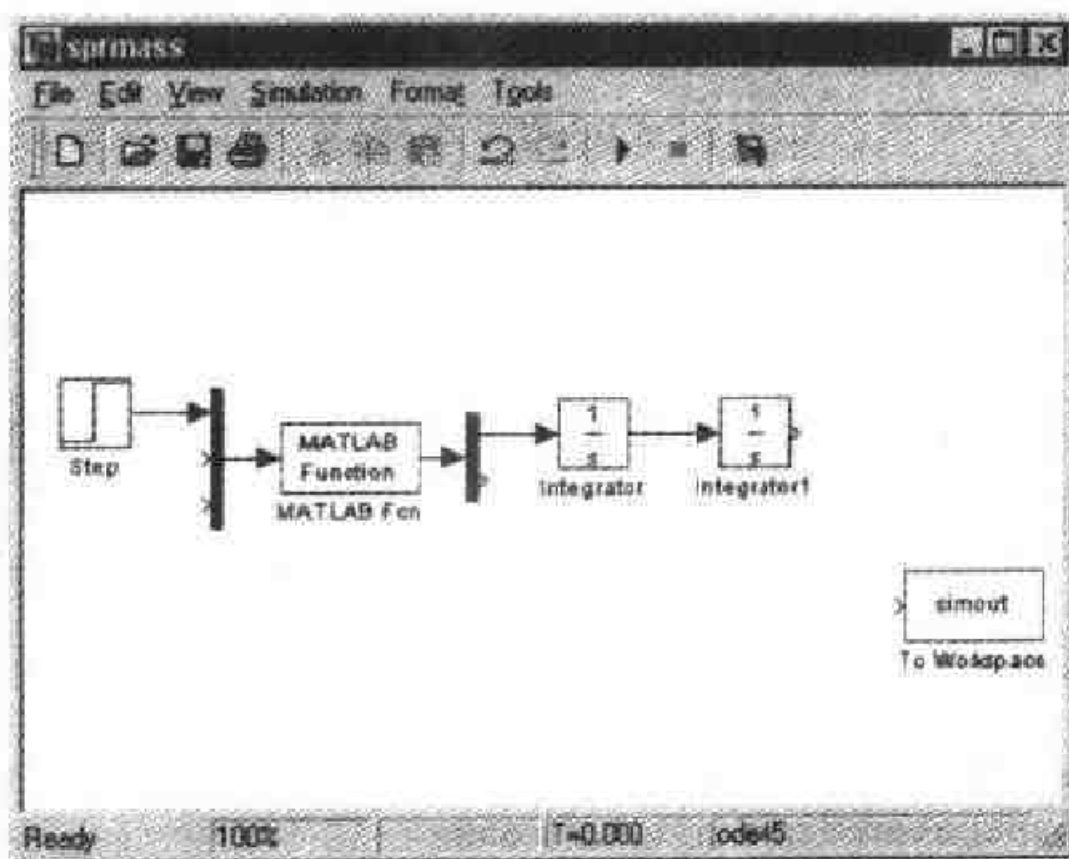


图 A-27 使用 Mux 和 DeMux 模块处理函数模块的多输入和多输出

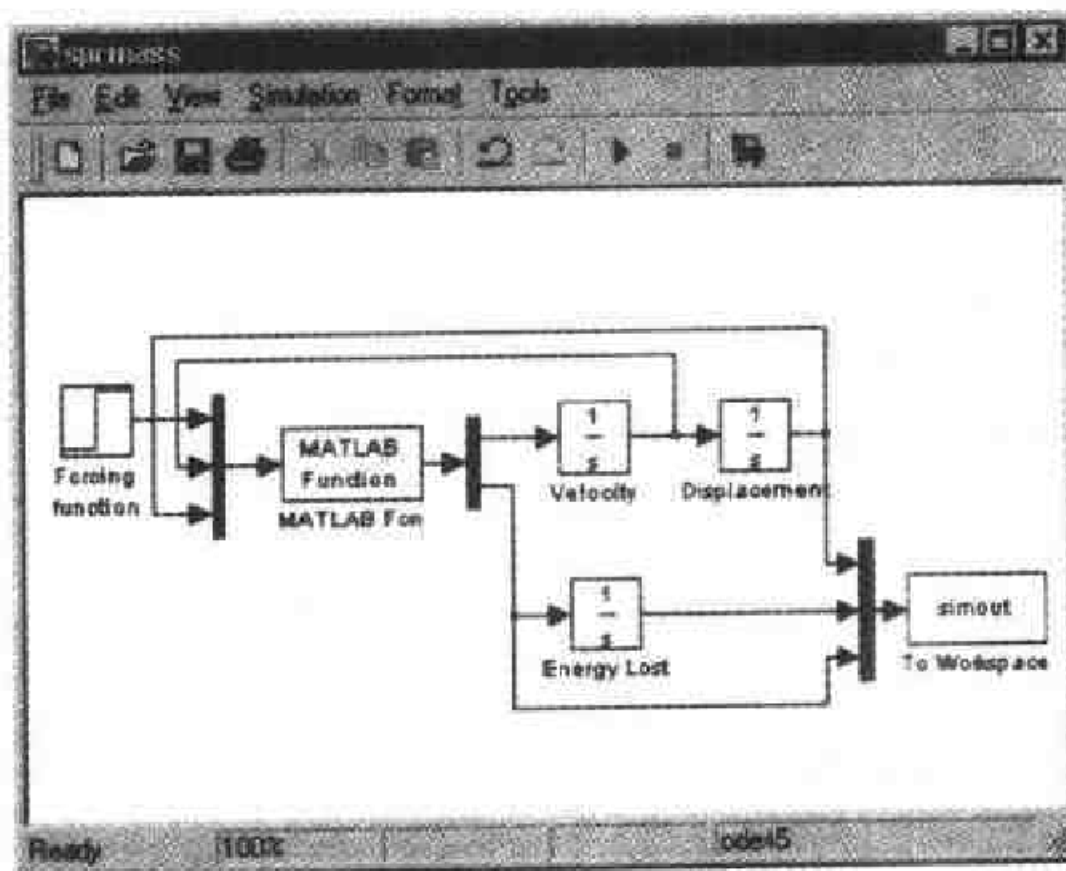


图 A-28 完成仿真结构,连接两个模块并增加一个计算总消耗能量的积分器(文件:srpmass.mdl)

式。

同时注意到模块的标志可以改变以便更好地反映模块的意义。通常,我们以积分器的输出或积分过程的结果标识该积分器。

此时,讨论一下关于仿真结构中循环的使用是十分有必要的。注意到速度和

位移将被用于计算系统的加速度。这似乎有点迂回曲折,但反映在状态空间理论^①里的动力系统基本结构不仅允许而且需要这样作。需要注意的最重要的特点是在循环中间有积分模块。如果构成的循环不包含一些动态模块,那么结果将是一个代数循环,这将导致数值求解困难或者有可能把系统的解排除。一般地,代数循环的出现(由 MATLAB 警告或错误信息显示)表明构造的仿真不好或不正确,这些循环应当视为错误。

按照已经给出的建立函数的步骤,我们可以编制出完成仿真计算所需的函数。该函数的 MATLAB 程序脚本如下:

```
pownacc.m
function out=pownacc(u)
%
% Function used to compute acceleration
% and power dissipated in a spring-mass-
% damper system.
%
% The inputs are assumed to be:
% u(1) = F(t)
% u(2) = v(t)
% u(3) = x(t)
%
% The outputs are:
%
% out(1) = acceleration
% out(2) = power dissipated
%
% Define the system parameters
%
m=1.0;
b=2.0;
k=3.0;
%
% Compute the acceleration
%
out(1) = 1/m * (u(1) - b * u(2) - k * u(3));
%
% Compute the power dissipated
%
out(2) = b * u(2)^2;
```

① Shearer J L, Kulakowski B T, Gardner J F. Dynamic Modelling and Control of Engineering Systems. 2nd ed. Englewood Cliffs, N J: Prentice Hall, 1997.

该函数的几个要素使得它与前面的几个例子不同。首先,输入变量 u 现在是由几个元素构成的向量。在 Simulink 文件中该函数模块的输入是三个信号的多路传输线这一事实表明了这一点。信号连接到多路转换器中的顺序与它们出现在函数中的顺序是一样的。换句话说,最顶端的信号(外加力函数)是 $u(1)$,第二个信号(与速度连接)是 $u(2)$,而最底部的信号(位移)是 $u(3)$ 。函数程序脚本中包含的注释起提示作用以确保信号通讯正确无误,但具体的注释内容取决于使用者。

其次,输出也是一个向量,与输出联系的 DeMux 模块具有和前面一样的功能。在函数中存储于 $out(1)$ 的输出量被置于 DeMux 模块的顶端或第一个信号,依此类推。

最后,我们在程序中使用了大量的注释。注释是任何工程模型必不可少的一部分,恰如其分的注释的重要性无论如何强调都不过分。

该文件已经被存为 `pownacc.m`。返回到 Simulink 文件,那里已经建立起了正确的函数模块。在对话框中应当指出文件名并正确地设置输出宽度(见图 A-29)。

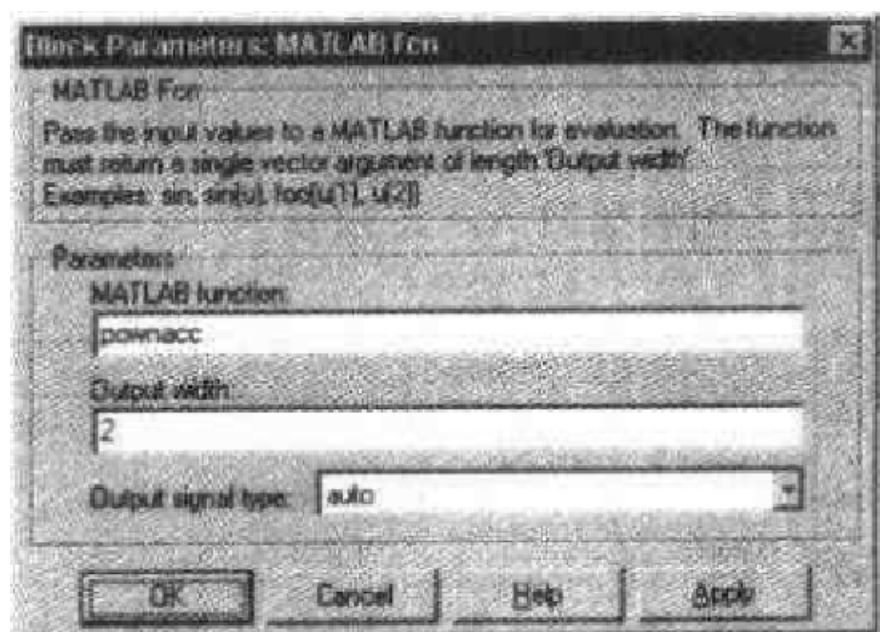


图 A-29 当函数返回一个向量时 MATLAB Fcn 模块的对话框

我们假设所有的初始条件都是零(积分器的缺省值)并运行仿真。一旦运行,结果将存在工作环境中并可以画出图形来。

```

> plot(tout, simout(:,1))
>

```

得到的图形如图 A-30 所示。

在函数的 `m` 文件中所选择的参数(有些随意)导致了一个欠阻尼系统。可以用下面的 `plot` 命令来看功率和消耗的能量。

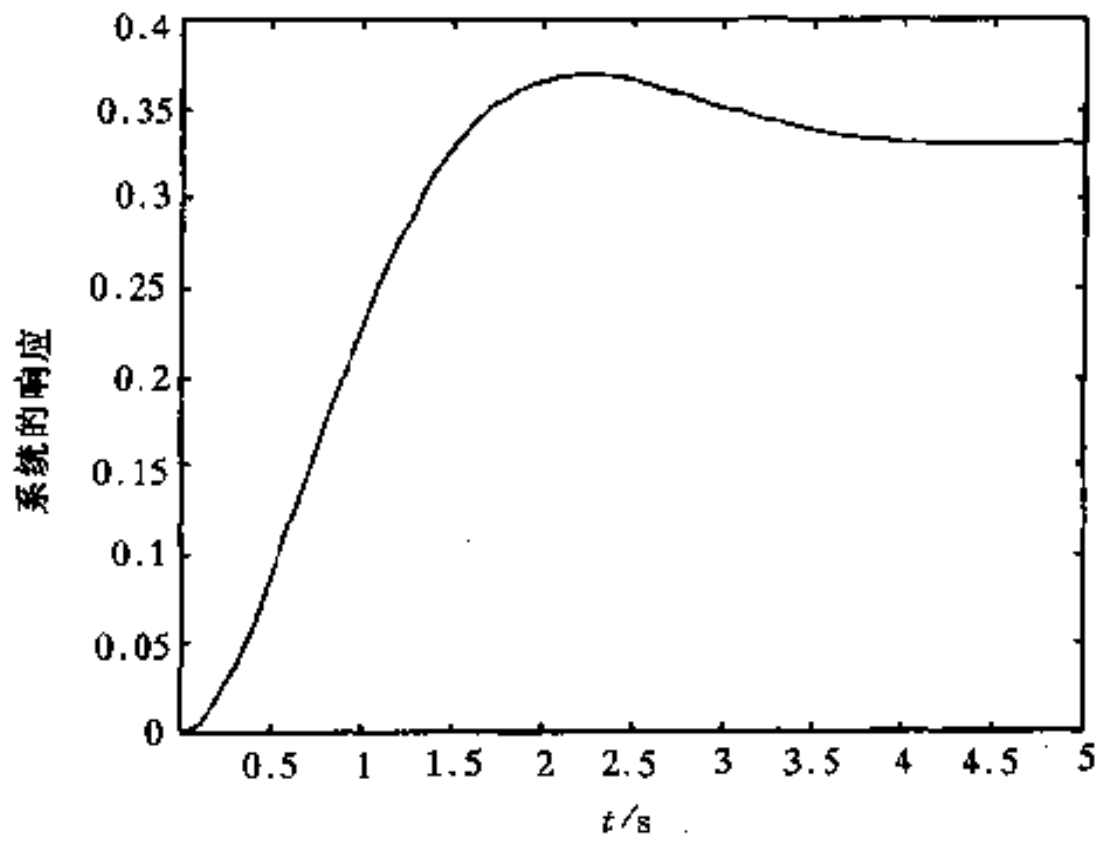


图 A-30 图 A-26 所示系统在阶跃输入下系统的响应曲线

```

> plot(tout, simout(:,2), tout, simout(:,3))
>

```

由此我们看到所画出的图形如图 A-31 所示。

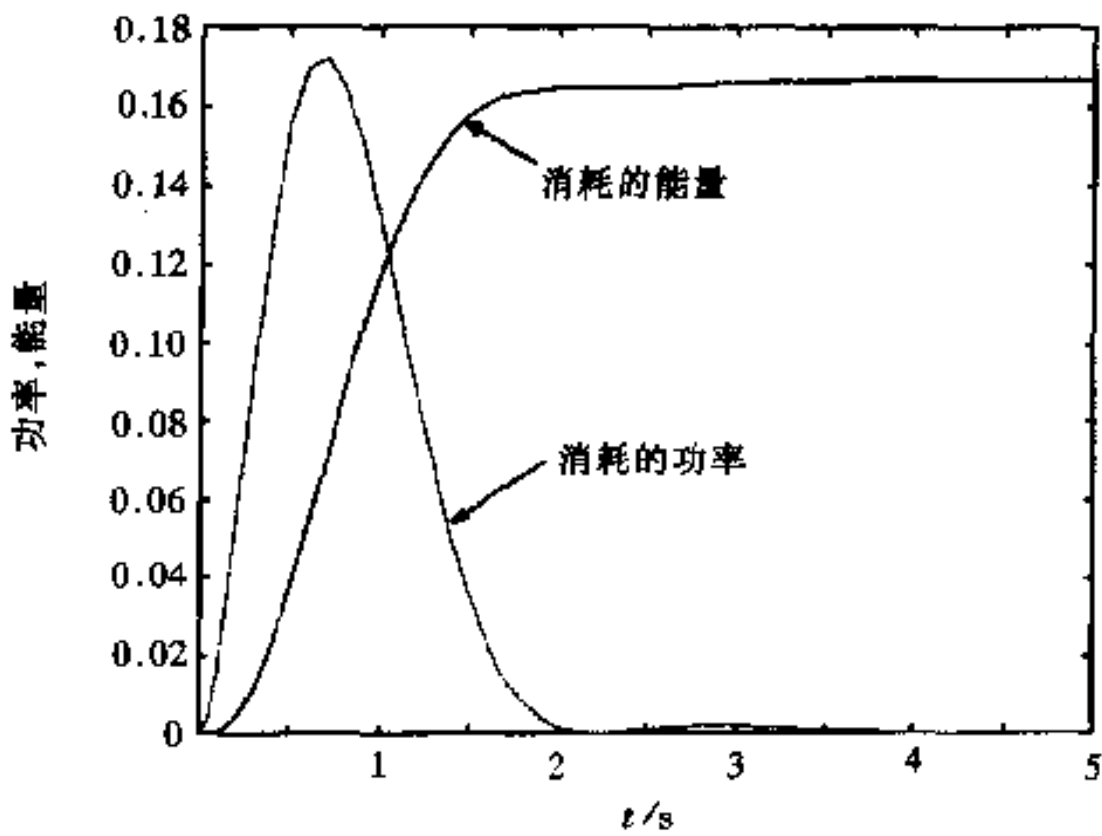


图 A-31 阶跃输入下系统中阻尼器的功率和所消耗的能量

A.10 结束语

本入门辅导仅仅包括了完成本书中的仿真所需要的很少一部分基本知识。无论 MATLAB 还是 Simulink 都具有非常丰富的功能和特色,一个简单的入门辅导不可能深入地介绍任何一个程序。和任何一个复杂的工程软件工具一样,除了反复练习使用软件之外没有更好的途径。在线帮助功能和基于 HTML 的帮助功能提供了计算机上大量的可用文件。Mathworks 的网址(www.mathworks.com)也是一个非常重要的资源,包含了许多使用者提供的程序、问题以及答案。

索引

B

闭环矢量方程, 4~15

Vector loop equations

~的定义, 4~15

of definition

~的导数, 6

derivatives of

四连杆机构的~, 6, 11

for four-bar linkage

快速释放机构的~, 11

for quick-release mechanism

曲柄滑块机构的~, 11

for slider crank

抛石机的~, 92, 93

for trebuchet

双连杆机器人的~, 12, 75, 76

for two-link robot

C

初始条件

Initial conditions

动态仿真的~, 55, 56, 69

of dynamic simulations

四连杆机构的~, 45

four-bar mechanism

间歇机构的~, 86, 87, 88

Geneva mechanism

运动学仿真的~, 30~34

for kinematic simulations

~的重量, 87, 88

resetting

Simulink 的~, 30~34, 108

in Simulink

曲柄滑块机构的~, 55, 56

slider-crank mechanism

两连杆机器人的~, 81

two-link robot

D

动态仿真(见仿真)

Dynamic simulation

多路信号分离器(见 Simulink 模块, 多路信号分离器/DeMux)

Demultiplexor

多路信号转换器(见 Simulink 模块, 多路信号转换器/MUX)

Multiplexor

F

仿真

Simulation

可变机构的~, 84~89

changing mechanisms

计算机~, 1~3

computer

~的定义, 1

definition of

动态~, 2

dynamic

四连杆机构的运动学~, 43~47

four-bar mechanism, kinematic
 间歇机构的~, 86~89
 of Geneva mechanism
 运动学~, 2, 3
 kinematic
 曲柄滑块机构的动态~, 67~70
 of slider-crank mechanism, dynamic
 曲柄滑块机构的运动学~, 27~32,
 34~40
 of slider-crank mechanism, kinematic
 弹簧-质量-阻尼系统的~, 120~124
 of spring-mass-damper
 抛石机的~, 90~99
 of trebuchet
 两连杆机器人的~, 75~83
 of two-link robot

J

机器人, 两连杆, 75~83
 Robot, two-link
 两连杆机器人的动力学方程, 76~79
 dynamic equation
 两连杆机器人的动态仿真, 79, 83
 dynamic simulation of
 两连杆机器人的矢量方程, 10~12,
 14, 75, 76
 vector equations
 计算机仿真(见仿真)
 Computer simulation
 间歇机构, 84~89
 Geneva mechanism
 ~的定义, 84
 definition of
 ~的运动学仿真, 86~89
 kinematic simulation of
 ~的矢量方程, 85
 vector equations of

加速度, 34
 Acceleration
 四连杆机构的~, 11, 7, 8, 41~44
 in four-bar linkage
 倒置曲柄滑块机构的~, 11, 84, 85
 in inverted slider crank
 质心~, 12, 52, 61, 62, 64, 65, 76
 of center of mass
 曲柄滑块机构的~, 34~42
 in slider crank
 矢量环的~, 7, 8
 vector loops

L

力

Force
 ~的分析, 49~58
 analysis
 轴承~, 70, 71
 bearing
 动态~, 49~58, 60, 70, 71
 dynamic
 ~方程, 52~58
 equations
 摩擦~, 49, 52, 57
 friction
 曲柄滑块机构中的~, 63~65
 slider-crank mechanism
 力之和, 49~61
 summation of

力之和, 49, 61

Summation of forces

力矩之和, 61

Summation of moments

M

MATLAB

~函数的调用, 23, 115~119
 calling functions
 ~m 函数, 19, 28, 115, 118
 function m-file
 ~帮助, 125
 help
 ~帮助面板, 125
 helpdesk
 ~矩阵运算, 20
 matrix manipulations
 ~m 文件, 19, 28, 115~118
 m-files
 ~plot 命令, 24, 113
 plot command
 ~绘图数据, 24, 32
 plotting data
 ~脚本文件, 19, 23, 28
 script m-file
 ~脚本和函数, 19, 28, 115~119
 scripts and functions
 ~有效数字, 21
 significant digits
 ~仿真模块, 101
 Simulink blocks
 ~仿真运行时间, 104~106
 Simulink run-time
 ~三角函数, 10
 trigonometric functions
 ~入门辅导, 110~120
 tutorial
 ~whos 命令, 112
 (见 Simulink, Simulink 模块)
 whos

摩擦(见力, 摩擦)

Friction

N

牛顿力学, 1~3, 49~57

Newtonian mechanics

牛顿-辛普森方程, 16~24

Newton-Raphson method

~的定义, 16~18

definition of

四连杆机构位置问题求解的~, 18~21

four-bar positions

~的初值估计, 21~22

initial guesses for

~的 Matlab 实现, 18~24

MATLAB implementation of

~的多重解, 21,

multiple solutions of

P

抛石机, 90~99

Trebuchet

~的描述, 90~92

description of

~的仿真, 90~99

simulation of

~的闭环矢量方程, 92~93

vector loop equation of

PD(比例微分)控制, 82~83

PD(proportional-derivative) control

Q

曲柄滑块机构

Slider-crank mechanism

~的加速度方程, 11, 34~35

acceleration equations

~的质心矢量, 64~66

center-of-mass vectors

~的动力学方程, 66

dynamic equations of

~的动态仿真, 67~70

dynamic simulation of

~的误差检验,40~42
 error checking in simulation
 ~的力方程,62~65
 force equations
 倒置的~,11,48,84~89
 inverted
 ~加速度的运动学仿真,34~40
 kinematic simulation of(acceleration)
 ~速度的运动学仿真,27~34
 kinematic simulation of(velocity)
 偏置的~,48,74
 offset
 ~的仿真研究,69~73
 simulation study of
 ~的闭环矢量方程,11,25~27,64~66
 vector loop equation
 ~的速度方程,11,27
 velocity equations

R

入门辅导

Tutorial

MATLAB的~,110~120
 MATLAB
 Simulink的~,100~125
 Simulink

S

矢量, x 和 y 分量,4

Vectors, x -and y -components

矢量链方程,4

Vector chain equations

任意点的~,12,48
 arbitrary point
 质心的~,61,76
 center of mass

四连杆机构,6,11

Four-bar linkage

~的加速度方程,7~9
 acceleration equations of
 ~的连杆曲线,48
 coupler curves
 ~的运动学仿真,43~47
 kinematic simulation of
 ~的位置求解,18~24
 position solution of
 ~的闭环矢量方程,5~7
 vector loop equations of
 ~的矢量环例子,8~9
 vector loop example
 ~的速度方程,6~8
 velocity equation of

Simulink

从~中调用函数,114~120
 calling functions from
 ~和MATLAB,110~112,114~120
 and MATLAB
 ~的运行,100~104
 running
 ~的运行时间参数,104~106
 run-time parameters
 ~初始条件的设置(见初始条件)
 setting initial conditions
 ~入门辅导,100~104
 tutorial

Simulink 模块 /

Simulink blocks

Clock~,30
 clock
 Constant~,27~29
 constant
 ~的坐标变换,70~72
 coordinate transformation
 多路信号分离~,29~30,121

DeMux / demultiplexor / DeMux
 增益~, 79, 83
 gain
 Integrator~
 Integrator
 ~的 MATLAB 函数, 28~29, 117~119
 MATLAB function
 多路信号转换~/Mux, 29~30, 108~110
 multiplexor / Mux
 ~积分器的重置, 87~89
 reset integrators
 Scope ~, 103~104
 Scope
 Step ~, 101~102, 105~106
 Step
 求和的~, 79~80
 summation
 To Workspace ~, 110~112
 To Workspace

T

泰勒级数展开, 16
 Taylor series expansion

W

位置问题, 16~24
 Position problem
 四连杆机构的~, 18~24
 four-bar
 ~的初始估计, 19
 initial guess
 ~的多重解, 19

multiple solutions of
 ~的求解, 16~24
 solution
 误差检验
 Error checking
 四连杆机构的~, 43~44
 four-bar mechanism
 运动学仿真中的~, 40~42
 in kinematic simulations
 仿真中的~, 40~42, 65~69
 in simulations
 曲柄滑块机构的~, 40~42, 65~69
 slider-crank mechanism
 抛石机的~, 96~97
 trebuchet

X

相容性检验(见误差检验)
 Consistency check

Y

运动学仿真, 2, 25~48
 Kinematic simulation
 ~的定义, 2
 definition of
 四连杆机构的~, 42~47
 four bar
 ~的初始条件, 30~32
 initial conditions for
 曲柄滑块机构的~, 25~42
 slider crank