

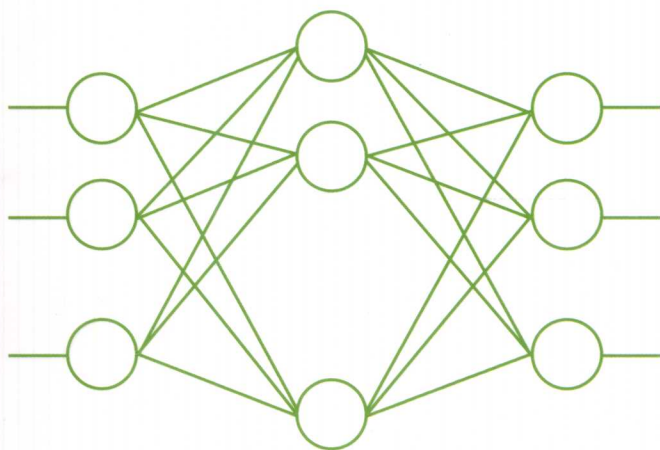
RBF NEURAL NETWORK CONTROL FOR MECHANICAL SYSTEMS

DESIGN, ANALYSIS AND MATLAB SIMULATION

RBF神经网络自适应控制 MATLAB仿真

刘金琨 著

Liu Jinkun



清华大学出版社



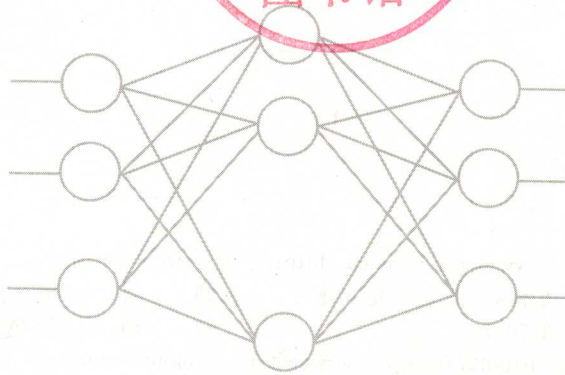
014024864

TP183
130

电子信息与电气工程技术人员丛书 (E&E)

RBF NEURAL NETWORK CONTROL FOR MECHANICAL SYSTEMS
DESIGN, ANALYSIS AND MATLAB SIMULATION

RBF神经网络自适应控制 MATLAB仿真



TP183
130



清华大学出版社
北京

内 容 简 介

本书从 MATLAB 仿真角度,结合典型机械系统控制的实例,系统地介绍神经网络控制的基本理论、基本方法和应用技术,是作者多年来从事控制系统教学和科研工作的结晶,同时融入了国内外同行近年来所取得的新成果。

全书共分 11 章,包括 RBF 网络的设计及分析、基于梯度下降法的 RBF 网络控制、简单的 RBF 网络自适应控制、RBF 网络滑模控制、基于 RBF 网络逼近的自适应控制、基于 RBF 网络的自适应反演控制、RBF 网络数字控制、离散系统的 RBF 网络控制及自适应 RBF 网络观测器的设计。每种控制方法都通过 MATLAB 进行了仿真分析。

本书各部分内容既相互联系又相互独立,读者可根据自己需要选择学习。本书适用于从事生产过程自动化、计算机应用、机械电子和电气自动化领域工作的工程技术人员阅读,也可作为大专院校工业自动化、自动控制、机械电子、自动化仪表、计算机应用等专业的教学参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

RBF 神经网络自适应控制 MATLAB 仿真/刘金琨著. --北京:清华大学出版社,2014

电子信息与电气工程技术丛书

ISBN 978-7-302-34040-9

I. ①R… II. ①刘… III. ①神经网络—自适应控制—研究 ②神经网络—计算机仿真
IV. ①TP183 ②TP391.75

中国版本图书馆 CIP 数据核字(2013)第 234295 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:李建庄

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:20.75

字 数:518千字

版 次:2014年1月第1版

印 次:2014年1月第1次印刷

印 数:1~2000

定 价:69.00元

神经网络控制出现于 20 世纪 80 年代,经历了 30 余年的发展,已形成了一个相对独立的研究分支,成为智能控制系统的一种设计方法,适用于线性与非线性系统、连续与离散系统、确定性与不确定性系统、集中参数与分布参数系统、集中控制与分散控制等。

神经网络具有高度并行的结构、强大的学习能力、连续非线性函数逼近能力、容错能力等优点,极大地促进与拓展了神经网络技术在非线性系统辨识与控制的应用。在实际工业过程中,存在着非线性、未建模动态、不可测噪声以及多环路等问题,这些问题对控制系统设计提出了很大的挑战。

与传统的控制策略相比,神经网络在如下几个方面具有优势:

(1) 神经网络对任意函数具有学习能力,神经网络的自学习能力可避免传统自适应控制理论中占有重要地位的复杂数学分析。

(2) 针对传统控制方法不能解决的高度非线性控制问题,多层神经网络的隐含层神经元采用了激活函数,它具有非线性映射功能,这种映射可以逼近任意非线性函数,为解决非线性控制问题提供了有效的解决途径。

(3) 传统自适应控制方法需要模型先验信息来设计控制方案。由于神经网络的逼近能力,控制器不要具体的模型信息,因此,神经网络控制可以广泛用于解决具有不确定模型的控制问题。

(4) 在神经网络的大规模并行处理架构下,网络的某些节点损坏并不影响整个神经网络的整体性能,有效地提高了控制系统的容错性。

有关神经网络控制理论及其工程应用,近年来已有大量的论文发表。作者多年来一直从事智能控制及应用方面的研究和教学工作,为了促进神经网络控制和自动化技术的进步,反映神经网络控制设计与应用中的最新研究成果,并使广大工程技术人员能了解、掌握和应用这一领域的最新技术,学会用 MATLAB 语言进行 RBF 网络控制器的设计,作者编写了这本书,借以抛砖引玉。

本书是在总结作者多年研究成果的基础上,进一步理论化、系统化、规范化、实用化而成,其特点如下:

(1) RBF 神经网络控制算法简单,内容先进,取材着重于基本概念、基本理论和基本方法。

(2) 针对每种 RBF 网络控制算法给出了完整的 MATLAB 仿真程序,并给出了程序的说明和仿真结果,具有很强的可读性。

(3) 着重从应用领域角度出发,突出理论联系实际,面向广大工程技术人员,具有很强的工程性和实用性。书中有大量应用实例及其结果分析,为读者提供了有益的借鉴。

(4) 所给出的各种 RBF 网络控制算法完整,程序设计结构设计力求简单明了,便于自

学和进一步开发。

本书共分 11 章。第 1 章为绪论,介绍神经网络控制的基本原理及在理论和应用方面的发展状况,并介绍了一种简单的自适应控制设计方法;第 2 章介绍 RBF 网络的设计、影响 RBF 网络的参数及离线建模方法;第 3 章介绍基于梯度下降法的 RBF 网络控制方法,包括 RBF 网络监督控制、RBF 网络模型参考自适应控制和 RBF 网络自校正控制 3 种方法;第 4 章介绍几种简单的 RBF 网络自适应控制的设计和分析方法;第 5 章介绍 RBF 网络滑模控制的设计及分析方法,并引入了一种基于神经网络最小参数学习法的自适应滑模控制方法;第 6 章和第 7 章分别介绍基于 RBF 网络局部逼近的自适应控制方法和基于 RBF 网络整体逼近的自适应控制方法,并以机械手控制为例给出了控制器的设计和分析实例;第 8 章以柔性机械臂的反演控制为例,介绍基于 RBF 网络的自适应反演控制的设计和分析方法;第 9 章介绍 RBF 网络自适应控制的离散化方法;第 10 章介绍离散系统的 RBF 网络控制器设计及离散控制系统的稳定性分析方法;第 11 章介绍自适应 RBF 网络观测器的设计和分析方法。

本书是作者在原有的英文版著作 *Jinkun LIU, RBF Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation* (Tsinghua & Springer Press, 2013) 基础上翻译过来的,并进行了适当的增减。

本书是基于 MATLAB 的 R2011a 环境下开发的,各个章节的内容具有很强的独立性,读者可以结合自己的研究方向深入地进行研究。

作者在研究过程中得到新加坡国立大学葛树志教授的热情支持和宝贵建议,在此表示感谢。研究生袁瑞侠参与了部分翻译和校对工作,在此表示感谢。书中也参考和引用了许多前人的研究成果,已尽量标明。作者对他们的研究工作表示由衷的敬意和感谢。

假如您对算法和仿真程序有疑问,请通过 E-mail 与作者联系。程序下载网址为 <http://ljk.buaa.edu.cn> 或 <http://si.buaa.edu.cn>。由于作者水平有限,书中难免存在一些不足和错误之处,欢迎广大读者批评指正。

作者联系方式: ljk@buaa.edu.cn

刘金琨

2014 年 1 月

于北京航空航天大学

符号说明

$\mathbf{R} \triangleq$ 实数集合

$\mathbf{R}^n \triangleq n$ 维向量实数集合

$\mathbf{R}^{n \times m} \triangleq n \times m$ 维矩阵向量实数集合

$|a| \triangleq$ 标量 a 的绝对值

$\det(\mathbf{A}) \triangleq$ 矩阵 \mathbf{A} 的行列式

$\|\mathbf{x}\| \triangleq$ 矢量 \mathbf{x} 的范数

$\mathbf{A}^T \triangleq$ 矩阵 \mathbf{A} 的转置

$\mathbf{A}^{-1} \triangleq$ 矩阵 \mathbf{A} 的逆

$\mathbf{I} \triangleq$ 单位矩阵

$\mathbf{I}_n \triangleq n \times n$ 维单位矩阵

$\lambda_i(\mathbf{A}) \triangleq \mathbf{A}$ 的第 i 个特征值

$\lambda(\mathbf{A}) \triangleq \mathbf{A}$ 的特征值集合

$\lambda_{\min}(\mathbf{A}) \triangleq$ 矩阵 \mathbf{A} 的最小特征值

$\lambda_{\max}(\mathbf{A}) \triangleq$ 矩阵 \mathbf{A} 的最大特征值

$x_i \triangleq$ 向量 \mathbf{x} 的第 i 个元素

$a_{ij} \triangleq$ 矩阵 \mathbf{A} 的第 ij 元素

$(\hat{\cdot}) \triangleq (\cdot)$ 的估计值

$(\tilde{\cdot}) \triangleq (\cdot) - (\hat{\cdot})$

$\sup \alpha(t) \triangleq \alpha(t)$ 的上界

$\text{diag}[\dots] \triangleq$ 给定元素构成的对角阵

$\mathbf{h} \triangleq$ RBF 网络高斯基函数的输出向量

$\mathbf{c}_j \triangleq$ RBF 网络高斯基函数第 j 个神经元的中心点向量

$b_j \triangleq$ RBF 网络高斯基函数第 j 个神经元的基宽

$\mathbf{W} \triangleq$ RBF 网络的权值矩阵

第 1 章 绪论	1
1.1 神经网络控制	1
1.1.1 神经网络控制的提出	1
1.1.2 神经网络控制概述	2
1.1.3 自适应 RBF 神经网络概述	2
1.2 RBF 神经网络	2
1.3 机器人 RBF 神经网络控制	3
1.4 控制系统 S 函数设计	3
1.4.1 S 函数介绍	3
1.4.2 S 函数基本参数	4
1.4.3 实例	4
1.5 简单自适应控制系统设计实例	5
1.5.1 系统描述	5
1.5.2 自适应控制律设计	6
1.5.3 仿真实例	6
附录 仿真程序	8
参考文献	12
第 2 章 RBF 神经网络设计与仿真	15
2.1 RBF 神经网络算法及仿真	15
2.1.1 RBF 网络算法设计	15
2.1.2 RBF 网络设计实例及 MATLAB 仿真	15
2.2 基于梯度下降法的 RBF 神经网络逼近	18
2.2.1 RBF 网络逼近	18
2.2.2 仿真实例	19
2.3 高斯基函数的参数对 RBF 网络逼近的影响	20
2.4 隐含层节点数对 RBF 网络逼近的影响	22
2.5 RBF 神经网络的建模训练	25
2.5.1 RBF 神经网络训练	25
2.5.2 仿真实例	26
2.6 RBF 神经网络逼近	27
附录 仿真程序	28
参考文献	44

目录

第 3 章 基于梯度下降法的 RBF 神经网络控制	45
3.1 基于 RBF 神经网络的监督控制	45
3.1.1 RBF 监督控制	45
3.1.2 仿真实例	46
3.2 基于 RBF 神经网络的模型参考自适应	47
3.2.1 控制系统设计	47
3.2.2 仿真实例	48
3.3 RBF 自校正控制	49
3.3.1 系统描述	49
3.3.2 RBF 控制算法设计	49
3.3.3 仿真实例	50
附录 仿真程序	51
参考文献	56
第 4 章 自适应 RBF 神经网络控制	57
4.1 基于神经网络逼近的自适应控制	57
4.1.1 系统描述	57
4.1.2 自适应 RBF 控制器设计	57
4.1.3 仿真实例	60
4.2 基于神经网络逼近的未知参数自适应控制	62
4.2.1 系统描述	62
4.2.2 自适应控制设计	63
4.2.3 仿真实例	65
4.3 基于 RBF 神经网络的直接鲁棒自适应控制	67
4.3.1 系统描述	67
4.3.2 理想反馈控制和函数逼近	67
4.3.3 控制器设计及分析	68
4.3.4 仿真实例	70
附录 仿真程序	73
参考文献	92
第 5 章 神经网络滑模控制	93
5.1 经典滑模控制器设计	93

5.2	基于 RBF 神经网络的二阶 SISO 系统的滑模控制	95
5.2.1	系统描述	95
5.2.2	基于 RBF 网络逼近 $f(\cdot)$ 的滑模控制	95
5.2.3	仿真实例	96
5.3	基于 RBF 逼近未知函数 $f(\cdot)$ 和 $g(\cdot)$ 的滑模控制	97
5.3.1	引言	97
5.3.2	仿真实例	99
5.4	基于神经网络最小参数学习法的自适应滑模控制	100
5.4.1	问题描述	101
5.4.2	基于 RBF 网络逼近的自适应控制	101
5.4.3	仿真实例	102
	附录 仿真程序	103
	参考文献	114
第 6 章	基于模型整体逼近的自适应 RBF 控制	116
6.1	基于 RBF 神经网络补偿的机器人自适应控制	116
6.1.1	系统描述	116
6.1.2	RBF 网络逼近	117
6.1.3	RBF 网络控制和自适应律设计及分析	117
6.1.4	仿真实例	120
6.2	基于滑模鲁棒项的 RBF 神经网络机器人控制	124
6.2.1	系统描述	124
6.2.2	RBF 逼近	125
6.2.3	控制律设计及稳定性分析	125
6.2.4	仿真实例	126
6.3	基于 HJI 理论和 RBF 神经网络的鲁棒控制	129
6.3.1	HJI 定理	129
6.3.2	控制器设计及稳定性分析	129
6.3.3	仿真实例	131
	附录 仿真程序	134
	参考文献	163
第 7 章	基于局部逼近的自适应 RBF 控制	165
7.1	基于名义模型的机械臂鲁棒控制	165
7.1.1	系统描述	165

目录

7.1.2	控制器设计	165
7.1.3	稳定性分析	166
7.1.4	仿真实例	167
7.2	基于局部模型逼近的自适应 RBF 机械手控制	168
7.2.1	问题描述	168
7.2.2	控制器设计	169
7.2.3	稳定性分析	170
7.2.4	仿真实例	172
7.3	工作空间机械手的神经网络自适应控制	175
7.3.1	关节角位置与工作空间直角坐标的转换	176
7.3.2	机械手的神经网络建模	176
7.3.3	控制器的设计	177
7.3.4	仿真实例	180
	附录 仿真程序	183
	参考文献	213
第 8 章	基于 RBF 网络的动态面自适应控制	214
8.1	简单动态面控制的设计与分析	214
8.1.1	系统描述	214
8.1.2	动态面控制器的设计	214
8.1.3	动态面控制器的分析	215
8.1.4	仿真实例	216
8.2	飞行器航迹角自适应神经网络动态面控制	218
8.2.1	系统描述	218
8.2.2	自适应神经网络动态面控制设计	219
8.2.3	稳定性分析	221
8.2.4	仿真实例	225
8.3	柔性关节机械手自适应 RBF 网络动态面鲁棒控制	228
8.3.1	问题描述	228
8.3.2	自适应 RBF 网络动态面控制器设计	229
8.3.3	闭环系统稳定性分析	231
8.3.4	仿真验证	234
	附录 仿真程序	237
	参考文献	262
第 9 章	数字 RBF 神经网络控制	264
9.1	自适应 Runge-Kutta-Merson 法	264

9.1.1	引言	264
9.1.2	仿真实例	265
9.2	SISO 系统的数字自适应控制	265
9.2.1	引言	265
9.2.2	仿真实例	266
9.3	两关节机械手的数字自适应 RBF 控制	267
9.3.1	引言	267
9.3.2	仿真实例	267
附录	仿真程序	269
	参考文献	276
第 10 章	离散神经网络控制	277
10.1	引言	277
10.2	一类离散非线性系统的直接 RBF 控制	277
10.2.1	系统描述	277
10.2.2	控制算法设计和稳定性分析	278
10.2.3	仿真实例	280
10.3	一类离散非线性系统的自适应 RBF 控制	284
10.3.1	系统描述	284
10.3.2	经典控制器设计	284
10.3.3	自适应神经网络控制器设计	284
10.3.4	稳定性分析	285
10.3.5	仿真实例	287
附录	仿真程序	291
	参考文献	297
第 11 章	自适应 RBF 观测器设计及滑模控制	299
11.1	自适应 RBF 观测器设计	299
11.1.1	系统描述	299
11.1.2	自适应 RBF 观测器设计	299
11.1.3	仿真实例	301
11.2	基于 RBF 自适应观测器的滑模控制	305
11.2.1	滑模控制器设计	305
11.2.2	仿真实例	306
附录	仿真程序	308
	参考文献	320

1.1 神经网络控制

1.1.1 神经网络控制的提出

自从20世纪40年代提出了基于单神经元模型构建的神经网络计算模型^[1],神经网络在感知学习、模式识别、信号处理、建模技术和系统控制等方面得到了巨大的发展与应用。神经网络具有高度并行的结构、强大的学习能力、连续非线性函数逼近能力、容错能力以及高效实时模拟VSL能力等优点,极大地促进与拓展了神经网络技术在非线性系统辨识与控制的应用^[2]。

在实际工业过程中,存在着非线性、未建模动态、不可测噪声以及多环路等问题,这些问题对在控制系统设计提出了很大的挑战。

经过几十年的发展,基于现代和经典控制理论的控制策略得到了很大的发展。现代控制理论中的自适应和最优控制技术以及经典控制理论主要都是基于线性系统,然而应用这些技术,都需要发展数学建模技术。

与传统的控制策略相比,神经网络在如下几个方面吸引了广大研究人员的注意,概括如下:

(1) 神经网络对任意函数具有学习能力,神经网络的自学习能力可避免在传统自适应控制理论中占有重要地位的复杂数学分析。

(2) 针对传统控制方法不能解决的高度非线性控制问题,多层神经网络的隐含层神经元采用了激活函数,它具有非线性映射功能,这种映射可以逼近任意非线性函数,为解决非线性控制问题提供了有效的解决途径。

(3) 传统自适应控制方法需要模型先验信息来设计控制方案,例如需要建立被控对象的数学模型。由于神经网络的自学习能力,控制器不要许多系统的模型和参数信息,因此,神经网络控制可以广泛用于解决具有不确定模型的控制问题。

(4) 采用神经元芯片或并行硬件,为大规模神经网络并行处理提供了非常快速的多处理技术。

(5) 在神经网络的大规模并行处理架构下,网络的某些节点损坏并不影响整个神经网络的整体性能,有效地提高了控制系统的容错性。

1.1.2 神经网络控制概述

针对 MIMO 模型,例如多关节机器人动力学模型,常规控制器的设计方法一般要求建立最小相位系统的结构和精确的数学模型,在许多情况下模型参数还要求精确已知。

神经网络可以通过前向和反向的动态行为在线学习复杂模型,通过适应环境的变化设计自适应 MIMO 控制器。从理论上讲,一个基于神经网络的控制系统的设计相对简单,因为它不要求有关该模型的任何先验知识。

神经网络的逼近能力已经被许多研究者证明^[3~7],一些研究人员在自适应神经网络控制器引入了逼近能力^[8~14],一些研究小组发展了自适应神经网络控制稳定性分析技术。

针对神经网络控制已经有许多成果发表。例如,一些研究人员针对非线性动态系统的辨识和控制问题,提出了一个统一的框架^[15],针对自适应非线性控制和自适应线性控制理论应用参数化方法进行稳定性分析。文献[8]通过引入 Ge-Lee 算子,针对机器人神经网络控制方法进行了稳定性分析和描述。文献[16~18]中给出了基于李雅普诺夫方法的典型神经网络控制系统稳定性分析方法。

反向传播(BP)神经网络和 RBF 神经网络的普及,大大促进了神经网络控制的发展^[19]。例如,文献[20~28]采用 BP 神经网络进行了控制器的分析和设计。

1.1.3 自适应 RBF 神经网络概述

针对 RBF 通用函数逼近的自适应非线性控制研究引起了国内外学者的广泛关注^[29~37]。RBF 神经网络在系统具有较大不确定性时,能有效地提高控制器的性能。在神经网络控制中,神经网络自适应律可通过 Lyapunov 方法导出,通过自适应权重的调节保证整个闭环系统的稳定性和收敛性。

本书通过许多控制系统设计和仿真实例,来论述当系统受到外界干扰时,采用 RBF 神经网络能够显著改善控制性能。

1.2 RBF 神经网络

在 1990 年,研究人员首次提出了非线性动力系统的人工神经网络自适应控制方法^[38]。此后,多层神经网络(MNN)和径向基函数(RBF)神经网络成功地应用于模式识别和控制中^[39]。

RBF 神经网络于 1988 年提出^[40]。相比多层前馈网络(MFN),RBF 网络由于具有良好的泛化能力,网络结构简单,避免不必要的和冗长的计算而备受关注。关于 RBF 网络的研究表明了 RBF 神经网络能在一个紧凑集和任意精度下,逼近任何非线性函数^[41,42]。目前,已经有许多针对非线性系统的 RBF 神经网络控制研究成果发表^[24,43]。

RBF 神经网络有 3 层:输入层、隐含层和输出层。隐含层的神经元激活函数由径向基

函数构成。隐含层组成的数组运算单元称为隐含层节点。每个隐含层节点包含一个中心向量 c , c 和输入参数向量 x 具有相同维数,二者之间的欧氏距离定义为 $\|x(t) - c_j(t)\|$ 。

隐含层的输出为非线性激活函数 $h_j(t)$ 构成

$$h_j(t) = \exp\left(-\frac{\|x(t) - c_j(t)\|^2}{2b_j^2}\right), \quad j = 1, \dots, m \quad (1.1)$$

其中, b_j 为一个正的标量,表示高斯基函数的宽度; m 是隐含层的节点数量。网络的输出由如下加权函数实现:

$$y_i(t) = \sum_{j=1}^m \omega_{ji} h_j(t), \quad i = 1, \dots, n \quad (1.2)$$

其中, ω 是输出层的权值; n 是输出节点个数; y 是神经网络输出。

1.3 机器人 RBF 神经网络控制

多输入多输出系统(MIMO)模型中的非线性、时变以及模型参数动态变化的控制问题,一直是控制领域的难题。这方面的研究成果已经有很多,如文献[44]针对一类双关节或多关节机械臂的控制问题进行了研究。

机械臂在柔性自动化领域已经成为越来越重要的研究对象,近年来,针对机械臂的控制器设计已取得了许多成果,为了实现精确的轨迹跟踪和良好的控制性能,出现了许多控制方法。

针对机械力臂的控制问题,计算力矩控制方法是最简单的控制方法,但该方法依赖于系统的精确非线性动力学模型。在实际工程中,机械手的有效载荷中可能会发生变化,在其运动期间,该参数无法实现精确预知。为了克服这个问题,许多研究人员引入了自适应控制策略^[45~47]。

自适应控制方法具有不需要未知参数的先验知识的优点,例如不需要有效载荷的质量。对于刚性机械臂系统,神经网络控制技术已经用于此类系统的研究,例如,采用神经网络自适应控制方法,可设计稳定跟踪控制的刚性连杆机械臂系统^[8~14, 48~52]。

针对柔性机械臂系统,出现了许多神经网络自适应控制方法的解决方案^[53~55]。例如,文献[53]针对柔性连杆机械臂关节位置跟踪的奇异摄动技术,设计了一个神经网络控制器,该方法不需要机器人机械臂的先验知识,且不需要离线训练神经网络。文献[54]针对连杆柔性臂尖端位置跟踪控制问题,设计了神经网络自适应控制器,不需要有效载荷质量的先验知识。

1.4 控制系统 S 函数设计

1.4.1 S 函数介绍

S 函数是 Simulink 的重要部分,它为 Simulink 环境下的仿真提供了强有力的拓展能力。S 函数可以用计算机语言来描述动态系统。在控制系统设计中,S 函数可以用来描述控制算法、自适应算法和模型动力学方程。

1.4.2 S函数基本参数

- (1) S函数: 包括 initialization 程序、mdlDerivative 程序、mdlOutput 程序等。
- (2) NumContStates: 连续状态个数。
- (3) NumDiscStates: 离散状态个数。
- (4) NumInputs 和 NumOutputs: 系统的输入和输出个数。
- (5) DirFeedthrough: 表示输入信号是否直接在输出端出现。

例如,形如 $y=k \times u$ 的系统需要输入(即直接反馈),其中 u 是输入, k 是增益, y 是输出,形如等式 $y=x, \dot{x}=u$ 的系统不需要输入(即不存在直接反馈),其中 x 是状态, u 是输入, y 为输出。

(6) NumSampleTimes: Simulink 提供了采样周期选项: 连续采样时间、离散采样时间、变步长采样时间等。对于连续采样时间,输出在很小的步长内改变。

1.4.3 实例

在控制系统设计中,S函数可以用于控制器、自适应律和模型描述。1.5节将会用到如下定义,在此作一个简单的介绍。

1. 模型初始化 Initialization 函数

采用S函数来描述形如式 $m\ddot{x}=u$ 定义的系统,可以看出系统模型是二阶的。如果采用S函数来描述2输入3输出系统,模型初始化参数可写为 $[0.5,0]$,假定模型的输出不直接由输入部分控制,则模型的初始化程序可描述如下:

```
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 2;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 3;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [0.5,0];
str = [];
ts = [0 0];
```

2. 动力学模型描述的 mdlDerivative 函数

在控制系统中,该函数可用于描述动态模型和自适应律等。例如,针对模型 $m\ddot{x}=u$ 的描述程序如下:

```
function sys = mdlDerivatives(t,x,u)
m = 2;
ut = u(2);
```

```
sys(1) = x(2);
sys(2) = 1/m * ut;
```

3. 自适应律的 mdlDerivative 函数

在控制系统中, S 函数的 mdlDerivative 函数用于描述自适应律, 例如自适应律 $\hat{m} = -\gamma v s$ 的程序描述如下:

```
function sys = mdlDerivatives(t,x,u)
xm = u(1);
dxm = u(2);
ddxm = u(3);
x1 = u(4);
dx1 = u(5);

e = x1 - xm;
de = dx1 - dxm;

nmn = 6;
s = de + nmn * e;
v = ddxm - 2 * nmn * de - nmn^2 * e;

gama = 0.5;
sys(1) = - gama * v * s;
```

4. mdlOutput 函数

S 函数的 mdlOutput 函数通常用于描述控制器或模型的输出。例如下面程序就是采用 S 函数 mdlOutput 模块来描述模型的输出:

```
function sys = mdlOutputs(t,x,u)
m = 2;
sys(1) = x(1);
sys(2) = x(2);
sys(3) = m;
```

1.5 简单自适应控制系统设计实例

本节基于文献[56]设计一个简单的自适应系统。

1.5.1 系统描述

系统通过电机控制输入 u 来控制, 动态模型如下:

$$m\ddot{x} = u \quad (1.3)$$

位置命令为 $r(t)$, 则包含 $r(t)$ 的参考模型如下:

$$\ddot{x}_m + \lambda_1 \dot{x}_m + \lambda_2 x_m = \lambda_2 r(t) \quad (1.4)$$

其中, λ_1 和 λ_2 为正实数; $\tilde{x} = x - x_m$ 为跟踪误差。

1.5.2 自适应控制律设计

如果 m 已知,则可设计如下的自适应控制律:

$$u = m(\ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x}) \quad (1.5)$$

其中, λ 是一个严格正实数。

将式(1.5)代入式(1.3)中,则可得一个指数收敛的跟踪误差动态模型:

$$\ddot{\tilde{x}} + 2\lambda\dot{\tilde{x}} + \lambda^2\tilde{x} = 0 \quad (1.6)$$

如果 m 未知,则可设计自适应律如下:

$$u = \hat{m}(\ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x}) \quad (1.7)$$

其中, \hat{m} 是对质量 m 的估计。

定义 $v = \ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x}$, 将式(1.7)代入式(1.3),则

$$m\ddot{x} = \hat{m}(\ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x}) = \hat{m}v$$

定义 $\tilde{m} = \hat{m} - m$, 上式变为

$$m(\ddot{x} - v) = \tilde{m}v \quad (1.8)$$

定义跟踪误差函数 s 为

$$s = \dot{\tilde{x}} + \lambda\tilde{x} \quad (1.9)$$

由式(1.9)可见, s 的收敛性意味着位置跟踪误差 \tilde{x} 和速度跟踪误差 $\dot{\tilde{x}}$ 的收敛性。

由于 $\ddot{x} - v = \ddot{x} - \ddot{x}_m + 2\lambda\dot{\tilde{x}} + \lambda^2\tilde{x} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}} + \lambda(\dot{\tilde{x}} + \lambda\tilde{x}) = \dot{s} + \lambda s$, 则式(1.8)变为

$$m(\dot{s} + \lambda s) = \tilde{m}v \quad (1.10)$$

定义 Lyapunov 函数如下:

$$V = \frac{1}{2} \left(ms^2 + \frac{1}{\gamma} \tilde{m}^2 \right)$$

从而有

$$\dot{V} = ms\dot{s} + \frac{1}{\gamma} \tilde{m}\dot{\tilde{m}} = ms\dot{s} + \frac{1}{\gamma} \tilde{m}\dot{\hat{m}}$$

参数 \hat{m} 的自适应律如下^[56]:

$$\dot{\hat{m}} = -\gamma vs \quad (1.11)$$

考虑式(1.10), 可得

$$\begin{aligned} \dot{V} &= ms\dot{s} + \frac{1}{\gamma} \tilde{m}(-\gamma vs) \\ &= ms\dot{s} - \tilde{m}vs = ms\dot{s} - (m\dot{s} + \lambda ms)s = -\lambda ms^2 \end{aligned}$$

根据 Barbalat 定理, s 收敛于零, 则位置跟踪误差 \tilde{x} 与速度跟踪误差 $\dot{\tilde{x}}$ 都在收敛于零。

1.5.3 仿真实例

在式(1.3)所描述的系统中, 假设真实质量为 $m=2$, 在仿真中, 初始值为 $\hat{m}(0)=0$, 采用式(1.7)与式(1.11)设计的控制律, 设定参数为 $\gamma=0.5, \lambda_1=10, \lambda_2=25, \lambda=6$, 分别设定参考位置为 $r(t)=0, r(t)=\sin(4t)$, 初始条件为 $\dot{x}(0)=\dot{x}_m(0)=0, x(0)=x_m(0)=0.5$ 。