

机器人操作器控制

——理论与实践（第二版）

中文译本

原著作者：

Frank L. Lewis
Darren M. Dawson
Chaouki T. Abdallah

翻译：Kimi、Gemini

校对：[FrankJIE09](#)

<https://github.com/FrankJIE09/Robot-Manipulator-Control-CN>

2026 年 4 月 10 日

前言与系列介绍

0.1 系列介绍

许多关于控制工程的教科书已经出版，描述了控制系统的新技术，或以新的、更好的方式从数学上阐述现有方法，以解决实际工程师面临的日益复杂的问题。然而，这些书籍很少涉及控制工程的应用方面。这个新系列的目的是纠正这种情况。

该系列将强调应用问题，而不仅仅是控制工程的数学。它将提供既介绍新技术又介绍成熟技术的教材，同时提供这些方法应用于解决现实世界问题的详细示例。作者将来自学术界和相关应用领域。

在电气、机械（包括航空航天）和化学工程等成熟领域，已经有许多控制技术应用的精彩示例。我们只需环顾当今高度自动化的社会，就能看到制造业中先进机器人技术的应用；空中和地面运输系统中自动控制和导航系统的使用；家用消费品中智能控制系统日益增多的使用；以及对家庭消费者和工业的可靠供水、供气和供电。然而，目前有许多具有挑战性的问题，如果能更广泛地应用控制方法学和控制系统化的应用基础，将会受益匪浅。

本系列推出的书籍将借鉴学术界和应用领域的专业知识，不仅可作为学术界推荐的课程教材，也可作为应用领域从业者的手册。《非线性控制系统》是 Dekker 控制工程系列中的另一部杰出作品。

0.2 前言

“机器人”一词由捷克剧作家卡雷尔·恰佩克（Karel Capek）在他 1920 年的戏剧《罗素姆的万能机器人》（Rossum's Universal Robots）中首次引入。捷克语中的“robota”一词简单地意味着“工作”。尽管有如此实用的起源，科幻小说作家和早期好莱坞电影却给了我们一种关于机器人的浪漫概念。这些机器的人形特性似乎给机器人的概念引入了某种人类寻找自身身份的元素。

“自动化”一词于 20 世纪 40 年代在福特汽车公司引入，是“自动动机”的缩写。“自动化”这一单一术语将两个概念结合在一起：为机械执行任务而设计的专用机器人机器的概念，以及指挥它们的自动控制系统的概念。

自动控制系统的历史有着深厚的根源。希腊人和阿拉伯人的大多数反馈控制器用于调节水钟以准确报时；这些在 14 世纪瑞士发明机械钟后被淘汰。自动控制系统直到三百年后的工业革命时期，随着需要先进控制器的复杂机器的出现，才真正兴起；我们特别想到的是风力磨坊和蒸汽机。另一方面，尽管由其他人（如 T.Newcomen 于 1712 年）发明，但蒸汽机的荣誉通常归于詹姆斯·瓦特，他在 1769 年制造了他的发动机，将机械创新与允许自动调节的控制系统结合在一起。也就是说，现代复杂机器除非配备合适的控制系统，否则是没有用的。

瓦特于 1788 年设计的离心飞球调速器提供了恒速控制器，使蒸汽机能够在工业中高效使用。飞球调速器的运动即使对未经训练的眼睛也是清晰可见的，其原理对许多人来说似乎具有异国情调，体现了新时代的精神。因此，调速器很快在欧洲各地引起了轰动。

主从遥操作机制于 20 世纪 40 年代中期在橡树岭和阿贡国家实验室用于放射性物质的远程处理。第一台商用机器人于 20 世纪 50 年代末由 Unimation 公司推出（几乎与 1957 年的斯普特尼克同时——因此太空时代和机器人时代同时开始）。像飞球调速器一样，机器人操作器的运动对未经训练的眼睛也是显而易见的，因此机器人设备的潜力可以激发想象力。然而，20 世纪 60 年代对工业和非结构化环境中自主机器人自动化的高期望通常未能实现。这是因为今天的机器人学与 1712 年纽科门工作后不久蒸汽机所处的阶段相同。

机器人学是一个跨学科领域，涉及物理学、机械设计、静力学和动力学、电子学、控制理论、传感器、视觉、信号处理、计算机编程、人工智能（AI）和制造等多种学科。各种专家研究机器人学的各种有限方面，但很少有工程师能够同时面对所有这些领域。这进一步增加了机器人学的浪漫化性质，例如，对于控制理论家来说，他对 AI 有着堂吉诃德式的幻想。

我们可以将机器人学分为五个主要领域：运动控制、传感器和视觉、规划和协调、人工智能和决策，以及人机接口。没有良好的控制系统，机器人设备是无用的。机器人手臂及其控制系统可以被封装为通用数据抽象；也就是说，机器人加控制器被视为与外部世界交互的单一实体或“代理”。

机器人代理的能力由运动和力施加能力的机械精度、手臂的自由度数量、抓手的可操控程度、传感器以及控制器的复杂性和可靠性决定。机器人手臂的输入仅仅是电机电流和电压，或液压或气动压力；然而，机器人加控制器代理的输入可以是期望的运动轨迹，或期望的施加力。因此，控制系统将机器人在抽象层次上提升了一个层次。

本书旨在提供对串联连杆机器人手臂控制系统的深入研究。它是我们 1993 年书籍的修订和扩展版本。增加了关于商用机器人操作器和设备、神经网络智能控制以及在真实机器人系统上实现先进控制器的新章节。第 1 章通过描述现有的机器人及其局限性和能力、传感器和控制器，将本书置于现有商业机器人系统的背景中。

我们希望本书既适合控制工程师也适合机器人学家。因此，附录 A 提供了机器人

运动学和雅可比矩阵的背景知识，第 2 章提供了控制理论和数学概念的背景知识。本书的初衷是作为研究生第二门机器人课程的教材，但鉴于背景材料，在 UTA 它被用作电气工程研究生的第一年课程。该课程也被列为本科课程的一部分，本科生很快就掌握了这些内容。

第 3 章介绍了作为控制设计基础所需的机器人动力学方程。附录 C 和全书示例中给出了一些常见手臂的动力学。第 4 章涵盖了计算力矩控制这一重要主题，它提供了重要的见解，同时也在统一框架中汇集了几种经典和现代机器人控制方案。

鲁棒控制和自适应控制分别在第 5 章和第 6 章中以并行方式介绍，以突出这两种在不确定性和干扰面前的控制方法的相似之处和不同之处。第 7 章讨论了一些先进技术，包括学习控制和具有柔性关节耦合的手臂。

基于生物系统的现代智能控制技术已经解决了复杂系统控制中的许多问题，包括未知的非参数化动力学和未知干扰、间隙、摩擦和死区。因此，我们在第 8 章增加了一章关于神经网络控制系统的内容。机器人只有在其与外部环境接触时才有用，因此第 9 章讨论了力控制问题。

成功控制器设计验证的关键是计算机仿真。因此，我们在全书中举例说明了受控非线性系统的计算机仿真过程。附录 B 提供了仿真软件。MATLAB 等商业软件包使机器人控制系统的仿真变得非常容易。

设计了机器人控制系统后，有必要实现它；鉴于今天的微处理器和数字信号处理器，从计算机仿真到实现只是一小步，因为仿真所需的控制器子程序（包含在本书中）与实际手臂实现所需的微处理器中的子程序几乎相同。事实上，第 10 章展示了在真实机器人系统上实现本书开发的先进控制器的技术。

所有基本信息和控制设计算法都以表格形式显示在书中。这连同书前的示例列表和表格列表，为学生、学者或实际工程师提供了方便的参考。

我们感谢 Milagro Design 的 Wei Cheng 负责本书的 LaTeX 排版和图表制作，以及将第一版的内容扫描成电子格式。

中文译本说明

本中文译本由 AI 辅助翻译生成，并经人工校对。翻译团队信息如下：

- **翻译：** Kimi、Gemini
- **校对：** [FrankJIE09](#)

相关链接：

- GitHub 仓库：<https://github.com/FrankJIE09/Robot-Manipulator-Control-CN>
- 在线阅读：<https://frankjie09.github.io/Robot-Manipulator-Control-CN/>

- SCARA 交互式可视化:https://frankjie09.github.io/Robot-Manipulator-Control-CN/figures/SCARA/SCARA_Type_I_vs_II_Comparison.html

本译本仅供学习交流使用，版权归原作者及原出版社所有。

F.L. Lewis, 德克萨斯州阿灵顿
D.M. Dawson, 南卡罗来纳州克莱姆森
C.T. Abdallah, 新墨西哥州阿尔伯克基

目录

Chapter 1

商用机器人操作器

1.1 引言

本章通过概述商用机器人操作器、传感器和控制器，为本书奠定基础。我们要强调的是，如果一个人希望获得高性能的柔性机器人工作单元，那么就有必要为机器人操作器设计如本书所述的先进控制系统。

在研究机器人控制、规划、传感器和人机接口的高级技术时，了解现有商用系统是很重要的。这允许人们在现有技术的背景下开发新技术，从而能够在现有机器人系统上实现这些新技术。

全国制造商协会的一份报告 [NAM 1998] 指出，20 世纪 90 年代美国商业制造业成功的两个最重要驱动因素是可重构制造工作单元和工厂中的局域网。在本章中，我们将讨论柔性机器人工作单元、商用机器人构型、商用机器人控制器、到互联网的信息集成以及机器人工作单元传感器。有关这些主题的更多信息可以在《机械工程手册》 [Lewis 1998] 和《计算机科学工程手册》 [Lewis and Fitzgerald 1997] 中找到。

1.1.1 柔性机器人工作单元

在工厂自动化和其他领域，曾经普遍使用围绕传送带或其他运输系统建造的固定布局，其中每个机器人执行特定任务。这些装配线具有不同的工作站，每个工作站执行专用功能。机器人已被用于工作站级别执行装配、钻孔、表面处理、焊接、码垛等操作。在装配线中，零件由运输系统顺序地路由到工作站。此类系统安装成本非常高，需要一组工程专家来设计和编程，并且随着需求的变化极难修改或重新编程。在今天的高混合低批量（HMLV）制造场景中，这些特性宣告了这种刚性过时设计的死亡。

在装配线中，机器人被限制在放入刚性顺序系统中。机器人是具有多种功能的通用机器，通过将它们用作柔性机器人工作单元的基础，其潜力可以显著增加，如图 1.1.1 所示的 UTA 自动化与机器人测试单元。在柔性机器人工作单元中，机器人用于零件搬运、装配和其他加工操作。通过重新编程机器人，可以改变工作单元的整个功能。工作

单元设计旨在充分利用机器人的工作空间，并且铣床、钻床、振动零件送料机等组件被放置在机器人的工作空间内，以便由机器人提供服务。与装配线相反，物理布局并不预先确定固定的操作或作业顺序。因此，随着产品需求的变化，只需要在软件中重新编程工作单元即可。工作单元非常适合制造业和其他领域中新兴的 HMLV 条件。

机器人工作单元的日益普及已经将重点从硬件设计转移开，并将新的重点放在包括规划、协调和控制 (PC&C) 功能的创新软件技术和架构上。需要对机器人控制器进行大量研究，以使机器人具备现代柔性工作单元所需的灵活性、精度和功能。本书的其余部分详细介绍了此类先进控制技术。

1.2 商用机器人构型与类型

本节中的大部分信息由 Mick Fitzgerald 准备，他当时是 UTA 自动化与机器人研究所 (ARRI) 的经理。

机器人是高度可靠、可信和技术先进的工厂设备。世界上大多数机器人由使用可靠的现成组件技术的成熟公司提供。所有商用工业机器人都有两个物理上分离的基本元素——操作器手臂和控制器。大多数商用机器人的基本架构基本相同，由串行连杆运动机器上的数字伺服控制电动机驱动组成，通常不超过六个轴（自由度）。所有机器人都配有专有控制器。几乎所有的机器人应用都需要工程师和技术人员进行大量的设计和实施工作。使每个机器人独特的是如何将组件组合在一起以实现性能，从而产生具有竞争力的产品。工业机器人应用中最重要考虑集中在两个问题上：操作和集成。

1.2.1 操作器性能

运动学结构、轴驱动机构设计和实时运动控制的综合效果决定了主要操作性能特征：可达性和灵巧性、负载能力、快速性和精度。在根据制造商公布的性能规格做出决策和进行比较时必须谨慎，因为测量和报告这些规格的方法在行业内并不统一。通常使用运动测试、仿真或其他分析技术来验证每个应用的性能。

可达性通过测量机器人运动所描述的工作空间范围来表征，而**灵巧性**通过各个关节的角位移来表征。一些机器人在其可达性边界内会有无法使用的空间，如死区、奇异姿态和腕部缠绕姿态。

负载重量由所有工业机器人的制造商指定。一些制造商还指定了旋转腕轴的惯性负载。通常给出的负载是在极端速度和可达条件下。所有工具、工件、电缆和软管的重量和惯性都必须作为负载的一部分包括在内。

快速性在确定吞吐量方面至关重要，但很难从公布的机器人规格中确定。大多数制造商会指定单个关节或特定运动学工具点的最大速度。然而，工作周期中的平均速度是感兴趣的快速性特征。

精度通常通过测量重复性来表征。几乎所有机器人都指定了静态位置重复性。很少指定准确度，但它可能比重复性大四倍。动态精度，或在连续路径上跟踪位置、速度和加速度的重复性和准确度，通常不被指定。

1.2.2 常见运动学构型

所有常见的商用工业机器人都是串行连杆操作器，通常不超过六个运动学耦合轴。按照惯例，运动轴按从基座到腕部的顺序编号。前三个轴负责机器人的空间定位运动；它们的构型决定了机器人可以定位的空间形状。运动链中的任何后续轴通常提供旋转运动以定向机器人手臂的末端，被称为腕部轴。在机器人腕部中，三个轴通常相交，以三维定向的形式产生真正的独立定位。附录 A 提供了球形机器人腕部机构的运动学分析。请注意，在我们的三维空间中，完全独立的空间定位需要三个自由度，完全独立的定向定位也需要三个自由度。

机器人轴可以在其驱动连杆中产生两种主要类型的运动——旋转或移动。旋转关节是拟人化的（例如像人类关节），而移动关节能够像汽车收音机天线一样伸缩。通常根据其前三个轴的方向和类型对机器人进行分类是非常有用的。有四种非常常见的商用机器人构型：**关节型**、**I 型 SCARA**、**II 型 SCARA** 和**笛卡尔型**。另外两种构型，**圆柱型**和**球形**，现在已不那么常见。

附录 C 包含了一些常见机器人操作器的动力学，用于本书的控制仿真。

关节型手臂。商用关节型手臂的种类繁多，大多数有六个轴（图 1.2.1）。所有这些机器人的轴都是旋转的。第二和第三轴共面并协同工作以在垂直平面内产生运动。基座中的第一轴是垂直的并旋转手臂以扫出大的工作体积。已经设计了各种类型的驱动机构，以允许腕部和前臂驱动电机和齿轮箱安装在靠近第一和第二旋转轴的位置，从而最小化手臂的延伸质量。当需要五个或更多自由度时，设计良好的关节型手臂的工作空间效率（即相对于手臂大小的快速灵巧可达程度）是其他手臂构型无法比拟的。关节型手臂性能的一个主要限制因素是第二轴必须工作以提升随后的手臂结构和负载。从历史上看，关节型手臂未能达到与其他手臂构型一样高的精度，因为所有轴都有关节角度位置误差，这些误差乘以连杆半径并在整个手臂上累积。

I 型 SCARA。I 型 SCARA（选择性柔顺装配机器人手臂）手臂使用两个平行旋转关节在水平平面内产生运动。手臂结构承载重量，但第一和第二轴不做提升。I 型 SCARA 的第三轴通过添加垂直或 z 轴提供工作体积。第四旋转轴将添加绕 z 轴的旋转以控制水平平面内的方向。这种类型的机器人很少发现超过四个轴。I 型 SCARA 广泛用于电子元件和设备的装配，并广泛用于小型和中型机械装配的装配。

II 型 SCARA。II 型 SCARA 也是一种四轴构型，与 I 型不同之处在于第一轴是一个长的垂直移动 z 行程，提升两个平行旋转轴及其连杆。对于在更长距离（超过约三英尺）上快速移动更重负载（超过约 75 磅），II 型 SCARA 构型比 I 型更有效。

注：读者可以通过[交互式可视化工具](#)对比 I 型和 II 型 SCARA 机器人的运

动学差异。

笛卡尔坐标机器人。笛卡尔坐标机器人使用正交移动轴，通常称为 x 、 y 和 z ，通过其矩形工作空间平移其末端执行器或负载。可以添加一个、两个或三个旋转腕轴用于定向。商业机器人公司提供几种类型的笛卡尔坐标机器人，工作空间大小从几立方英寸到数万立方英尺，负载能力可达数百磅。门式机器人具有高架桥结构，是最常见的笛卡尔型，非常适合需要在大面积和/或大负载上提供服务的物料搬运应用。它们在弧焊、水射流切割和大型复杂精密零件的检查等应用中特别有用。

模块化笛卡尔机器人也可从几家商业来源获得。每个模块是一个自包含的完全功能的单轴执行器；这些模块可以定制组装用于特殊用途应用。

球型和圆柱坐标机器人。球型坐标机器人的前两个轴是旋转的且相互正交，第三轴提供移动径向延伸。结果是具有球型工作体积的自然球坐标系统。圆柱坐标机器人的第一轴是旋转基座旋转。第二和第三是移动的，产生自然的圆柱运动。球型和圆柱型机器人的商业型号最初非常常见，在机器看护和物料搬运应用中很受欢迎。数百台仍在使用，但现在只有少数商用型号。这两种构型使用下降的原因归因于使用移动连杆进行径向延伸/缩回运动的问题；实心臂需要间隙才能完全缩回。

并联连杆操作器。对于某些特殊用途应用，并联连杆机器人比串联连杆机器人更适合。这些机器人通常有三个或六个并联连杆，每个连杆连接到一个固定基座和一个移动工作平台。如图 1.2.7 所示。通过适当的设计，六连杆并联连杆操作器可以使工作平台具有六自由度运动。军用 Link 训练器是一种大型并联连杆机器人，移动飞行员座椅。这些机器人比串联连杆机器人具有更大的刚度和精度，在串联连杆机器人中，每个连杆的定位误差随着从基座向外移动而累积。因此，轻质并联连杆机器人能够精确地移动大负载。这些机器人已被用于加工和表面精加工精密工业和航空航天部件，如舱壁和飞行器外壳。

并联连杆机器人是一个闭链运动学系统，因此相对难以分析 [Liu and Lewis 1993]。这些机器人的控制系统设计问题更加困难。

注：读者可以通过[交互式可视化工具](#)探索六自由度并联连杆操作器（Stewart 平台）的运动特性。

1.2.3 商用机器人的驱动类型

绝大多数商用工业机器人使用带减速传动的电动伺服电机驱动。交流和直流电机都很流行。现在有一些用于喷漆应用的伺服液压关节型手臂机器人。很少发现带伺服气动驱动轴的机器人。使用所有类型的机械传动，但趋势是向低间隙和零间隙型驱动发展。一些机器人使用直接驱动方法来消除与其他驱动相关的惯性和机械间隙放大。关节角度位置传感器是实时伺服级控制所需的，通常被认为是传动系统的重要部分。较少提供速度反馈传感器。

注：读者可以通过[交互式可视化工具](#)对比传统齿轮传动与直接驱动方式的差异，观察间隙对控制精度的影响。

1.3 商用机器人控制器

商用机器人控制器是专门的多处理器计算系统，提供四个基本过程，允许将机器人集成到自动化系统中：运动轨迹生成和跟踪、运动/过程集成和排序、人机集成以及信息集成。

1.3.1 运动轨迹生成和跟踪

工业机器人运动生成有两个重要的与控制器相关的方面。一个是可编程的操作程度，另一个是执行受控编程运动的能力。每个机器人系统的独特方面其实时间伺服级运动控制。实时控制的细节通常不会向用户透露，出于安全和专有信息保密的原因。每个机器人控制器通过其操作系统程序，将来自更高级协调器的数字数据通过精确计算和高速分配和通信转换为协调的手臂运动，各个轴运动命令由各个关节伺服控制器执行。大多数商用机器人控制器以 16 毫秒的采样周期运行。实时运动控制器 invariably 使用经典独立关节比例-积分-微分 (PID) 控制或 PID 的简单改进。这使得商用控制器适用于点对点运动，但大多数不适用于跟踪连续位置/速度轨迹或施加规定的力，即使有大量的编程工作。

注：读者可以通过[交互式可视化工具](#)调节 PID 参数 (K_p 、 K_i 、 K_d)，实时观察对关节控制性能的影响。

最近，出现了更先进的控制器。Adept Windows 系列的自动化控制器集成了机器人学、运动控制、机器视觉、力传感和制造逻辑在一个与 Windows 98 & Windows NT/2000 兼容的单一控制平台上。Adept 运动控制器可以配置为控制其他机器人和定制机构，并且是各种 OEM 系统的标准配置。

1.3.2 运动/过程集成和排序

运动/过程集成涉及将操作器运动与过程传感器或其他过程控制器设备协调。最基本的过程集成是通过离散数字输入/输出 (I/O)。例如，机器人控制器外部的机器控制器可能会发送一位信号，表示它已准备好被机器人装载。机器人控制器必须能够读取数字信号并使用该信号执行逻辑操作 (if then、wait until、do until 等)。也就是说，一些机器人控制器内置了一些可编程逻辑控制器 (PLC) 功能。与传感器 (例如视觉) 的协调也通常提供。

1.3.3 人机集成

控制器的人机接口对于机器人系统的快速设置和编程至关重要。大多数机器人控制器有两种类型的人机接口可用：计算机式 CRT/键盘终端用于离线和编辑程序代码，以及示教器，这是便携式手动输入终端，用于通过触摸键或操纵杆以遥操作方式命令运动。示教器通常是定位机器人最有效的方式，控制器中的存储器可以在回放模式下播放示教位置以执行运动轨迹。通过实践，人类操作员可以快速示教一系列在回放模式下链接在一起的点。目前，大多数机器人应用依赖于编程阶段人类专业知识的集成，以成功规划和协调机器人运动。这些接口机制在无阻碍的工作空间中有效，在编程和执行之间没有变化。它们不允许在执行期间人机接口或适应变化的环境。

更 recent 的高级机器人接口技术基于行为编程，其中各种特定行为以低级别编程到机器人控制器中（例如拾取零件、插入机器卡盘）。然后由更高级别的机器主管根据人类操作员的规定对这些行为进行排序并指定其特定运动参数。这种方法在 [Mireles and Lewis 2001] 中使用。

1.3.4 信息集成

随着向增加灵活性和敏捷性的趋势影响机器人学，信息集成变得越来越重要。许多商用机器人控制器现在通过通信端口（例如 RS-232）采用集成 PC 接口支持信息集成功能，或者通过直接连接到机器人控制器数据总线。最近的集成努力使得可以将机器人工作单元连接到互联网，以允许远程站点监控和控制。有许多技术可以实现这一点，其中最方便的是 LabVIEW 6.1，它不需要用 Java 编程。

1.4 传感器

本节中的大部分信息由 Kok-Meng Lee 准备 [Lewis 1998]。传感器和执行器 [Tzou and Fukuda 1992] 作为换能器发挥作用，高级工作单元规划、协调和控制系统通过它与组成工作单元的硬件组件接口。传感器是一个至关重要的元素，因为它们将物理设备的状态转换为适合输入到工作单元 PC&C 控制系统的信号；不合适的传感器可能引入错误，无论 PC&C 系统多么复杂或昂贵，都无法使正确操作成为可能，而传感器的创新选择可以使控制和协调问题变得更加容易。

传感器有许多不同类型和许多不同的用途。考虑到与生物系统的类比，**本体感受器**是设备内部的传感器，产生关于该设备内部状态的信息（例如机器人手臂关节角度传感器）。**外感受器**产生关于设备外部其他硬件的信息。传感器产生模拟或数字输出；数字传感器通常提供关于机器或资源状态的信息（夹持器打开或关闭、机器已装载、作业完成）。传感器产生 PC&C 层次结构所有级别所需的输出，包括用于：

- 伺服级反馈控制（通常是模拟本体感受器）

- 过程监控和协调（通常是数字外感受器或零件检查传感器如视觉）
- 故障和安全监控（通常是数字——例如接触传感器、气动压力损失传感器）
- 质量控制检查（通常是视觉或扫描激光）。

传感器输出数据通常必须进行处理，以将其转换为对 PC&C 目的有意义的形式。传感器加所需的信号处理显示为虚拟传感器。它作为数据抽象发挥作用——一组数据加上对该数据的操作（例如相机、加帧采集卡、加信号处理算法如图像增强、边缘检测、分割等）。一些传感器，包括伺服级反馈控制所需的本体感受器，是其主机设备的组成部分，因此传感器数据的处理和数据的使用发生在该设备内；然后，传感器数据在伺服控制级别或机器协调级别合并。其他传感器，通常是视觉系统，在复杂性上与机器人操作器相当，并由作业协调器协调，作业协调器将其视为有价值的共享资源，其使用由某种优先级分配（例如分派）方案分配给需要它们的作业。所谓的主动感知提出了一个有趣的协调问题，例如，机器人可能持有扫描相机，相机有效地接管运动协调问题，指挥机器人向何处移动以在后续图像中实现熵的最大减少（信息增加）。

1.4.1 传感器类型

本节从操作角度概述传感器。有关功能和物理原理的更多信息可以在 [Fraden 1993]、[Fu et al. 1987]、[Snyder 1985] 中找到。

触觉传感器。触觉传感器依靠与外部物体的物理接触。数字传感器如限位开关、微动开关和真空设备提供关于是否发生接触的二进制信息。有传感器可以检测滑移的开始。模拟传感器如弹簧加载杆提供更多信息。基于橡胶状碳或硅基弹性体并嵌入电气或机械组件的触觉传感器可以提供关于零件几何形状、位置等的非常详细的信息。弹性体可以包含电阻或电容元件，其电气特性随着弹性体压缩而变化。基于 LSI 技术的设计可以生产触觉网格垫，例如在单个垫上有 64×64 的”力点”。此类传感器产生具有类似相机数字图像属性的”触觉图像”，需要类似的数据处理。额外的触觉传感器属于随后讨论的”力传感器”分类。

接近和距离传感器。非接触接近传感器包括基于霍尔效应的设备或基于电磁效应的感应设备，可以检测约 5 毫米范围内的铁磁材料。此类传感器通常是数字的，产生关于物体是否在附近的二进制信息。基于电容的传感器检测约 5 毫米范围内的任何附近固体或液体。光学和超声传感器具有更长的范围。

距离传感器包括飞行时间测距设备如声纳和激光。商用 Polaroid 声纳提供约 1 英寸的精度，可达 5 英尺，角扇区精度约 15 度。对于移动机器人导航应用中的 360 度覆盖，可以使用扫描声纳和环形安装的多声纳。声纳通常有噪声，有虚假读数，需要低通滤波和其他旨在降低虚警率的数据处理。更昂贵的激光测距仪在距离上极其精确，具有很高的角分辨率。

位置、速度和加速度传感器。线性位置测量设备包括线性电位计和刚刚讨论的声纳和激光测距仪。线性速度传感器可以是基于激光或声纳的多普勒效应设备。

关节角度位置和速度本体感受器是机器人手臂伺服控制驱动轴的重要组成部分。角度位置传感器包括使用直流电压的电位计和使用交流电压的旋转变压器，精度为 15 分钟。光学编码器可以使用数字技术提供极高的精度。增量光学编码器使用三个光学传感器和单个交替不透明/透明区域环，如图 1.4.1(a) 所示，以提供相对于参考点的角位置信息和角速度信息；商用设备每转可能有 1200 个槽。更昂贵的绝对光学编码器，如图 1.4.1(b) 所示，有 n 个同心环的交替不透明/透明区域，需要 n 个光学传感器。如果采用格雷码，它们提供更高的精度并最小化与数据读取和传输相关的误差，其中连续扇区之间只有一位变化。精度为 $360^\circ/2^n$ ，商用设备具有 $n \approx 12$ 。

注：读者可以通过[交互式可视化工具](#)观察增量式与绝对式光学编码器的工作原理，理解格雷码的防误读机制。

陀螺仪如果补偿与漂移相关的重复性问题，具有良好的精度。定向陀螺的精度约为 1.5 度。垂直陀螺的精度为 0.5 度，可用于测量多轴运动（例如俯仰和横滚）。速率陀螺直接测量速度，阈值约为 0.05 度/秒。

注：读者可以通过[交互式可视化工具](#)对比三种陀螺仪（定向/垂直/速率）的精度差异和漂移特性。

有各种基于应变计（下一段）、陀螺仪或晶体特性的加速度计可用。商用设备可用于测量沿三个轴的加速度。一种流行的新技术涉及微机电系统 (MEMS)，它要么是表面要么是体微加工设备。MEMS 加速度计非常小、便宜、鲁棒且精确。MEMS 传感器特别已用于汽车行业 [Eddy 1998]。

力和扭矩传感器。有各种扭矩传感器可用，尽管通常不需要；例如，机器人手臂关节的内部扭矩可以从电机电枢电流计算。钻削工具上的扭矩传感器可以指示工具何时变钝。

线性力可以使用负载传感器或应变计测量。应变计是一种弹性传感器，其电阻是施加应变或变形的函数。压电效应，即在施加力时产生电压，也可用于力传感。其他力传感技术基于真空二极管、石英晶体（其共振频率随施加力变化）等。

机器人手臂力-扭矩腕部传感器在灵巧操作任务中非常有用。商用设备可以沿三个垂直轴测量力和扭矩，提供关于与腕部接触的笛卡尔力矢量 F 的完整信息。标准变换允许计算其他坐标中的力和扭矩。六轴力-扭矩传感器相当昂贵。

光电传感器。有各种光电传感器可用，有些基于光纤原理。这些响应速度约为 50 微秒，范围可达约 45 毫米，可用于检测零件和标签、扫描光学条形码、确认分拣任务中的零件通过等。

其他传感器。有各种传感器可用于测量压力、温度、流体流量等。这些在焊接等某些过程的闭环伺服控制应用中以及作业协调和/或安全中断程序中非常有用。

1.4.2 传感器数据处理

在任何传感器用于机器人工作单元之前，必须对其进行校准。根据传感器的不同，这可能涉及大量的实验、计算和安装后的调整工作。制造商通常提供校准程序，尽管在某些情况下，包括视觉，此类程序可能不明显，需要参考已发表的科学文献。系统修改后可能需要耗时的重新校准。

特别是对于更复杂的传感器如光学编码器，需要大量的传感器信号调理和处理。这可能包括信号放大、噪声抑制、数据从模拟到数字或从数字到模拟的转换等。制造商通常为这些目的提供硬件，应被视为机器人工作单元设计的传感器包的一部分。传感器及其信号处理硬件和软件算法可被视为数据抽象，被称为“虚拟传感器”。

如果需要解决信号处理问题，使用有限状态机 (FSM) 设计通常非常有用。来自增量光学编码器的典型信号如图 1.4.2(a) 所示；用于将其解码为角位置的 FSM 如图 1.4.2(b) 所示。FSM 非常容易直接转换为逻辑门形式的硬件。用于声纳排序的 FSM 如图 1.4.3(a) 所示；从该 FSM 派生的声纳驱动硬件如图 1.4.3(b) 所示。

一个特定问题是从角位置测量获得角速度。很多时候，位置测量只是简单地差分，使用小采样周期来计算速度。如果信号中有任何噪声，这肯定会导致问题。几乎总是需要使用低通滤波函数，其中速度样本 v_k 从位置测量样本 p_k 使用以下公式计算：

$$v_k = \frac{p_k - p_{k-1}}{T} + \alpha(v_{k-1} - v_k)$$

其中 T 是采样周期， α 是小滤波系数。需要类似的方法来计算加速度。

1.4.3 视觉系统、相机和照明

典型的商用视觉系统符合 20 世纪 50 年代的 RS-170 标准，因此帧通过帧采集卡以每秒 30 帧的速率采集。图像是扫描的；在一个流行的美国标准中，每个完整的扫描或帧由 525 条线组成，其中 480 条包含图像信息。这个采样率和这种数量级的图像分辨率足以满足大多数应用，除了基于视觉的机器人手臂伺服。机器人视觉系统相机通常是电视相机——固态电荷耦合器件 (CCD)，对低于 350 纳米 (紫外线) 到 1100 纳米 (近红外) 的光波长响应，峰值响应约为 800 纳米，或电荷注入器件 (CID)，提供类似的光谱响应，峰值响应约为 650 纳米。线扫描 CCD 相机和面积扫描 CCD 相机都可用，分辨率范围在 256 到 2048 元素之间。中分辨率面积扫描相机产生 256×256 的图像，尽管现在已有 1024×1024 的高分辨率设备。线扫描相机适用于零件在相机前移动的应用，例如在传送带上。帧采集器通常支持多个相机，常见数量为四个，可支持黑白或彩色图像。

如果听任自然，机器人工作单元的照明可能会导致操作中的严重问题。常见问题包括低对比度图像、镜面反射、阴影和无关细节。这些问题可以通过过于复杂的图像处理来纠正，但所有这些问题都可以通过在工作单元设计阶段对细节进行适当关注来避免。照明技术包括光谱滤波、选择合适的照明源光谱特性、漫射照明技术、背光 (产生易于处理的轮廓)、结构光 (提供额外的深度信息并简化物体检测和解释) 以及定向照明。

参考文献

- [Decelle 1988] Decelle, L.S., "Design of a Robotic Workstation For Component Insertions," AT&T Technical Journal, March/April 1988, Volume 67, Issue 2. pp 15–22.
- [Eddy 1998] Eddy, D.S., and D.R.Sparks, "Application of MEMS technology in automotive sensors and actuators," Proc. IEEE, vol. 86, no. 8, pp. 1747–1755, Aug. 1998.
- [Fraden 1993] Fraden, J. AIP Handbook Of Modern Sensors, Physics, Design, and Applications, American Institute of Physics. 1993.
- [Fu et al. 1987] Fu, K.S., R.C.Gonzalez, and C.S.G.Lee, *Robotics*, McGraw-Hill, New York, 1987.
- [Jamshidi et al. 1992] Jamshidi, M., Lumia, R., Mullins, J., and Shahinpoor, M., 1992. *Robotics and Manufacturing: Recent Trends in Research, Education, and Applications*, Vol. 4, ASME Press, New York.
- [Lewis and Fitzgerald 1997] Lewis, F.L., M.Fitzgerald, and K.Liu "Robotics," in *The Computer Science and Engineering Handbook*, Allen B.Tucker, Jr. ed., Chapter 33, CRC Press, 1997.
- [Lewis 1998] Lewis, F.L., "Robotics," in *Handbook of Mechanical Engineering*, F.Kreith ed., chapter 14, CRC Press, 1998.
- [Liu and Lewis 1993] Liu, K., F.L.Lewis, G.Lebret, and D.Taylor, "The singularities and dynamics of a Stewart Platform Manipulator," J. Intelligent and Robotic Systems, vol. 8, pp. 287–308, 1993.
- [Mireles and Lewis 2001] Mireles, J., and F.L.Lewis, "Intelligent Material Handling: Development and implementation of a matrix-based discrete-event controller," IEEE Trans. Industrial Electronics, vol. 48, no. 6, pp. 1087–1097, Dec. 2001.
- [NAM 1998] National Assoc. Manufacturers Report, "Technology on the Factory Floor III," ed. P.M.Swadimass, NAM Pub. Center, 1–800–637–3005, Aug. 1998.
- [Pugh 1983] Pugh, A., ed., 1983. *Robotic Technology*, IEE Control Engineering Series 23, Perginus, London.
- [Snyder 1985] Snyder, W.E., 1985, *Industrial Robots: Computer Interfacing and Control*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA.
- [Tzou and Fukuda 1992] Tzou, H.S., and Fukuda, T., *Precision Sensors, Actuators, and Systems*, Kluwer Academic, 1992.

Chapter 2

控制理论导论

第二章：控制理论导论

在本章中，我们回顾了用于机器人控制的各种控制理论概念。我们首先回顾了线性和非线性系统的状态空间描述，并介绍了后续所需的稳定性概念。本章旨在介绍现代控制概念，但即使是具有控制理论背景的读者也可能希望查阅它以了解符号和方便性。

2.1 引言

机器人机械臂的控制是一个成熟但富有成效的研究、开发和制造领域。工业机器人本质上是定位和搬运设备。因此，一个有用的机器人是能够控制其运动以及机器人与环境之间的相互作用力和力矩的机器人。本书关注机器人机械臂的控制方面。控制通常需要数学模型的可用性以及某种作用于模型的智能。机器人的数学模型是从控制其运动的基本物理定律中获得的。另一方面，智能需要感知能力以及作用于和响应感知变量的手段。这些机器人的作用和反应是控制器设计的结果。

在本章中，我们回顾了本书所需的控制理论概念。所有证明都被省略，但参考了提供更多证明的更专业书籍和论文。一旦在第 2 章中描述了机器人动力学的满意模型，本章介绍的自动控制概念可用于修改机器人对不同刺激的作用和反应。因此，后续章节将讨论将控制原理应用于机器人方程。所使用的特定控制器将取决于数学模型的复杂性、手头的应用、可用资源以及许多其他标准。

我们在第 2.2 节开始本章，回顾线性的连续时间和离散时间系统的状态空间描述。在第 2.3 节中介绍了非线性系统的类似回顾。非线性系统的平衡在第 2.4 节中回顾，而向量空间的概念在第 2.5 节中介绍。稳定性理论在第 2.6 节中介绍，这是本章的主要内容。在第 2.7 节中介绍了李雅普诺夫稳定性结果，而输入-输出稳定性概念在第 2.8 节中介绍。在第 2.9 节中编制了高级稳定性概念，以使后续发展更加简洁。在第 2.10 节中，我们回顾了一些有用的定理和引理。在第 2.11 节中，我们从状态空间的角度回顾了基

本的线性控制器设计，本章在第 2.12 节中结束。

2.2 线性状态变量系统

本书中考虑的机器人等许多物理系统由微分或差分方程描述。这些描述方程通常从基本物理定律中获得，为系统的分析和控制提供了起点。当然，有些系统非常复杂，没有可用的描述微分（或差分）方程。我们在本书中不考虑这些系统。

在本节中，我们研究物理系统的状态空间模型，这些系统是线性的。我们将自己限制为由常微分方程描述的系统，这将导致有限维状态空间。需要偏微分方程（导致无限维系统）来研究柔性机器人机械臂，但这些在本教材中不予考虑。我们强调，本章的材料旨在作为这些主题的快速介绍，并不全面。读者可参考 [Kailath 1980]、[Antsaklis and Michel 1997] 以了解线性控制系统的更严格介绍。

2.2.1 连续时间系统

在深入研究连续时间系统的数学描述之前，我们先明确几个重要的基本概念：

[校对者注：基本概念通俗解释]

- **连续时间 vs 离散时间**：连续时间系统的状态随时间连续变化，在每一时刻都有定义（如真实的物理世界）；而离散时间系统只在特定的采样时刻有定义（如计算机程序）。机器人手臂的平滑运动是连续的，而数字控制器的输出是离散的。
- **动态 vs 非动态**：动态系统的当前输出取决于过去的输入和状态（有“记忆”），如机器人关节的位置取决于之前的历史运动；非动态系统的输出只取决于当前输入（无记忆、即时响应），如纯比例控制器 $u(t) = K_p e(t)$ 。
- **时不变 vs 时变**：时不变系统的特性不随时间改变（“脾气”稳定），如理想的刚性机械臂；时变系统的参数会随时间变化（如齿轮磨损导致间隙增大、负载变化导致惯量改变）。

连续时间系统被称为线性的，如果它遵循叠加原理，即如果输出 $y_1(t)$ 由输入 $u_1(t)$ 产生，输出 $y_2(t)$ 由输入 $u_2(t)$ 产生，那么由 $u(t) = \alpha_1 u_1(t) + \alpha_2 u_2(t)$ 产生的输出由 $y(t) = \alpha_1 y_1(t) + \alpha_2 y_2(t)$ 给出，其中 α_1 和 α_2 是标量常数。线性、单输入/单输出 (SISO)、连续时间、时不变系统由线性、标量、常系数常微分方程描述，例如：

$$\sum_{i=0}^n a_i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^n b_i \frac{d^i u(t)}{dt^i} \quad (2.1)$$

其中 $a_i, b_i, i = 0, \dots, n$ 是标量常数， $y(t)$ 是标量输出， $u(t)$ 是标量输入。此外，在时间 t_0 给定初始条件：

$$y(t_0), \frac{dy(t_0)}{dt}, \dots, \frac{d^{n-1}y(t_0)}{dt^{n-1}} \quad (2.2)$$

注意，输入 $u(t)$ 的导数次数最多与输出 $y(t)$ 相同。否则，系统被称为非动态的。

状态空间实现

系统的状态被定义为一组充分的变量，当在时间 t_0 指定这些变量以及输入 $u(t), t \geq t_0$ 时，足以完全确定系统在 $t \geq t_0$ 时的行为 [Kailath 1980]。状态向量然后包含确定系统中任何信号的未来行为所需的所有必要变量。根据定义，这样的状态向量 $x(t)$ 不是唯一的，这一特性稍后将加以利用。事实上，如果 x 是状态向量，那么任何 $\tilde{x}(t) = Tx(t)$ 也是状态向量，其中 T 是任何 $n \times n$ 可逆矩阵。对于在(??)中描述的连续时间系统，可以选择以下状态向量：

$$x_i(t) = \frac{d^{i-1}y(t)}{dt^{i-1}}, \quad i = 1, 2, \dots, n \quad (2.3)$$

其中 $x_1 = y, x_2 = \dot{y}, \dots, x_n = y^{(n-1)}$ 。输入-输出方程然后简化为：

$$y(t) = b_0x_1(t) + b_1x_2(t) + \dots + b_{n-1}x_n(t) + u(t) \quad (2.4)$$

(??)和(??)的更紧凑表述由下式给出：

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (2.5)$$

[校对者注：符号含义说明]

状态空间方程中各符号的物理含义：

- **x (状态向量)**：系统内部的状态变量，包含描述系统行为所需的全部信息。如机器人关节的角度 x_1 、角速度 x_2 、角加速度 x_3 等。 x_1 是位置， $x_2 = \dot{x}_1$ 是速度， $x_3 = \dot{x}_2$ 是加速度。
- **u (输入/控制量)**：外部施加的控制作用，如电机转矩、电压、力等。这是我们可以主动调节的量。
- **y (输出/观测量)**：通过传感器实际测量得到的量，如编码器读数（角度）、陀螺仪读数（角速度）。 y 可以是 x 的任意组合，取决于传感器的类型和安装位置。

以机器人单关节为例：若只用编码器测角度，则 $y = x_1$ ；若同时用陀螺仪测角速度，则输出方程包含 x_1 和 x_2 两项。

其中

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & -\frac{a_2}{a_n} & \cdots & -\frac{a_{n-1}}{a_n} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (2.6)$$

这种特定的状态空间表示被称为可控规范型 [Kailath 1980]、[Antsaklis and Michel 1997]。一般来说，线性、时不变、连续时间系统将具有多个输入和一个以上的输出。事

实上, $u(t)$ 是 $m \times 1$ 向量, $y(t)$ 是 $p \times 1$ 向量。将 $u(t)$ 与 $y(t)$ 关联的微分方程此处不介绍, 但多输入/多输出 (MIMO) 系统的状态空间表示变为:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (2.7)$$

其中 A 是 $n \times n$, B 是 $n \times m$, C 是 $p \times n$, D 是 $p \times m$ 。有关 A, B, C 和 D 的具体形式, 读者再次参考 [Kailath 1980]、[Antsaklis and Michel 1997]。(??)的框图如图 2.2.1a 所示。注意, 状态的最小数量等于找到微分方程组的唯一解所需的初始条件的数量。

示例 2.1 (双重积分器). 考虑由以下方程描述的 SISO 系统:

$$\ddot{y}(t) = u(t)$$

这个系统被称为双重积分器, 代表了由牛顿定律描述的各种各样的物理系统。为了获得状态空间描述, 设:

$$x_1 = y, \quad x_2 = \dot{y}$$

所以

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u, \quad y = x_1$$

示例 2.2 (双平台系统). 考虑图 2.2.2 所示的 MIMO 机械系统, 它代表了一个用于将实验与外部扰动隔离的双平台系统。系统有两个输入: 导致地面移动的 u_2 和导致平台 m_1 移动的 u_1 。系统还有 2 个输出, 即平台 m_1 的运动 y_1 和平台 m_2 的运动 y_2 。实验将在平台 m_1 的顶部进行, 因此, 人们希望最小化 y_1 的大小。系统包含弹簧 k_1 、 k_2 和阻尼器 c_1 、 c_2 , 使用牛顿第二定律获得描述该系统的微分方程:

$$m_1 \ddot{y}_1 + c_1(\dot{y}_1 - \dot{y}_2) + k_1(y_1 - y_2) + u_1 = 0 \quad (2.8)$$

$$m_2 \ddot{y}_2 + c_1(\dot{y}_2 - \dot{y}_1) + c_2 \dot{y}_2 + k_1(y_2 - y_1) + k_2(y_2 - u_2) = 0 \quad (2.9)$$

可以通过选择以下状态获得该系统的状态空间表述:

$$x_1 = y_1, \quad x_2 = \dot{y}_1, \quad x_3 = y_2, \quad x_4 = \dot{y}_2$$

状态方程为:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k_1}{m_1}(x_1 - x_3) - \frac{c_1}{m_1}(x_2 - x_4) - \frac{u_1}{m_1} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{k_1}{m_2}(x_3 - x_1) - \frac{k_2}{m_2}(x_3 - u_2) - \frac{c_1}{m_2}(x_4 - x_2) - \frac{c_2}{m_2}x_4 \end{aligned}$$

注: 读者可以通过[交互式可视化工具](#)观察双平台弹簧-质量-阻尼系统的动态响应, 调节参数 (质量、弹簧刚度、阻尼系数) 并观察外部激励对两个平台的影响。

传递函数

线性、时不变、连续时间系统的另一种等价表示由其传递函数给出，它将系统的输入 $u(t)$ 与其输出 $y(t)$ 在拉普拉斯变量 s 中或在频域中关联起来。重要的是要注意，传递函数描述没有关于系统初始条件的信息，因此，除非所有初始条件都为零，否则它不会为特定输入提供唯一输出 [Antsaklis and Michel 1997]、[Kailath 1980]。然而，传递函数形式在实践中很重要，因为许多工程师熟悉频域规范。此外，许多系统的识别可以在频域中有效地进行 [Ljung 1999]。因此，人们必须能够在状态空间（或现代）描述和传递函数（或经典）描述之间进行转换。

让我们考虑由(??)描述的系统并取其拉普拉斯变换：

$$\begin{aligned} sX(s) &= AX(s) + BU(s) + x(0) \\ Y(s) &= CX(s) + DU(s) \end{aligned} \quad (2.10)$$

其中 $X(s), U(s)$ 和 $Y(s)$ 分别是 $x(t), u(t)$ 和 $y(t)$ 的拉普拉斯变换。初始状态向量为 $x(0)$ 。通过在(??)中的两个方程之间消去 $X(s)$ ，我们找到以下关系：

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) + C(sI - A)^{-1}x(0) \quad (2.11)$$

如前所述，当 $x(0) = 0$ 时，传递函数作为输入 $U(s)$ 和输出 $Y(s)$ 之间的关系获得，即：

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) \quad (2.12)$$

这个特定线性、时不变系统的传递函数由下式给出：

$$P(s) = C(sI - A)^{-1}B + D \quad (2.13)$$

使得（见图 2.2.1）：

$$Y(s) = P(s)U(s) \quad (2.14)$$

示例 2.3 (双重积分器的传递函数). 考虑示例 2.2.1 的系统。很容易看出传递函数是：

$$P(s) = \frac{1}{s^2}$$

2.2.2 离散时间系统

在离散时间情况下，使用差分方程描述系统如下：

$$\sum_{i=0}^n a_i y(k+i) = \sum_{i=0}^n b_i u(k+i) \quad (2.15)$$

其中 $a_i, b_i, i = 0, \dots, n$ 是标量常数， $y(k)$ 是输出， $u(k)$ 是时间 k 的输入。注意，时间 $k+n$ 的输出依赖于时间 $k+n$ 的输入，但不依赖于后面的输入；否则，系统将是非因果的。

状态空间表示

以与连续时间情况类似的方式，定义以下状态向量：

$$x_i(k) = y(k + i - 1), \quad i = 1, 2, \dots, n \quad (2.16)$$

输入-输出方程然后简化为：

$$y(k) = b_0 x_1(k) + b_1 x_2(k) + \dots + b_{n-1} x_n(k) + u(k) \quad (2.17)$$

(??)和(??)的更紧凑表述由下式给出：

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (2.18)$$

其中

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & -\frac{a_2}{a_n} & \dots & -\frac{a_{n-1}}{a_n} \end{bmatrix} \quad (2.19)$$

MIMO 情况与连续时间情况类似，由下式给出：

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (2.20)$$

其中 A 是 $n \times n$ ， B 是 $n \times m$ ， C 是 $p \times n$ ， D 是 $p \times m$ 。

在许多实际情况下，例如在机器人的控制中，系统是连续时间系统，但控制器使用数字硬件实现。这将要求设计者在连续和离散时间系统之间进行转换。有许多不同的方法来“离散化”连续时间系统，其中一些在第 3 章中讨论。对这一控制问题的这一非常重要方面感兴趣的读者可参考 [Åström and Wittenmark 1996]、[Franklin et al. 1997]。

示例 2.4 (离散时间双重积分器). 回忆示例 2.2.1 中提出的双重积分器或牛顿系统模型。微分方程的一个离散时间版本由以下差分方程给出：

$$y(k+2) - 2y(k+1) + y(k) = T^2 u(k)$$

其中 T 是以秒为单位的采样周期。如果我们选择 $x_1(k) = y(k)$ 和 $x_2(k) = x_1(k+1)$ ，我们得到状态空间描述：

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= 2x_2(k) - x_1(k) + T^2 u(k) \\ y(k) &= x_1(k) \end{aligned}$$

传递函数表示

以与连续时间情况类似的方式，由(??)给出的线性、时不变、离散时间系统可以在 Z 变换域中由其传递函数 $P(z)$ 描述，从输入 $U(z)$ 到输出 $Y(z)$ ，使得：

$$Y(z) = P(z)U(z)$$

其中

$$P(z) = C(zI - A)^{-1}B + D$$

注意，Z 变换用于离散时间情况，而拉普拉斯变换用于连续时间情况。

示例 2.5 (离散时间双重积分器的传递函数). 示例 2.2.4 的传递函数由下式给出：

$$P(z) = \frac{T^2}{z^2 - 2z + 1}$$

2.3 非线性状态变量系统

在许多情况下，潜在的物理行为可能无法使用线性状态变量方程来描述。这就是机器人机械臂的情况，其中不同连杆之间的相互作用由非线性微分方程描述，如第 3 章所示。状态变量表述仍然能够处理这些系统，而传递函数和频域方法则失效。在本节中，我们处理前一节的非线性变体，并强调 [Khalil 2001]、[Vidyasagar 1992] 和 [Verhulst 1997]、[LaSale and Lefschetz 1961]、[Hahn 1967] 中研究的非线性系统的经典方法。

2.3.1 连续时间系统

非线性、标量、连续时间、时不变系统由非线性、标量、常系数微分方程描述，例如：

$$f(y, \dot{y}, \dots, y^{(n)}, u, \dot{u}, \dots, u^{(m)}) = 0 \quad (2.21)$$

其中 $y(t)$ 是输出， $u(t)$ 是所考虑系统的输入。与线性情况一样，我们通过其分量定义状态向量 x 如下：

$$x_1 = y, \quad x_2 = \dot{y}, \quad \dots, \quad x_n = y^{(n-1)} \quad (2.22)$$

输出方程然后简化为：

$$y(t) = x_1(t) \quad (2.23)$$

(??)和(??)的更紧凑表述由下式给出：

$$\begin{aligned} \dot{x} &= f(x, U) \\ y &= cx \end{aligned} \quad (2.24)$$

其中

$$U(t) = [u(t) \quad \dot{u}(t) \quad \cdots \quad u^{(n-1)}(t)]^T$$

和

$$c = [1 \quad 0 \quad 0 \quad \cdots \quad 0] \quad (2.25)$$

示例 2.6 (非线性系统). 我们提出 2 个示例来说明这些概念:

1. 考虑阻尼摆方程

$$\ddot{y} + \dot{y} + \sin(y) = 0$$

通过选择 $x_1 = y, x_2 = \dot{y}$ 获得状态空间描述, 导致:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -x_2 - \sin(x_1)$$

$y(t)$ 的时间历程如图 2.3.1 所示。

2. 一个经典的非线性系统是 Van der Pol 振荡器, 其描述为:

$$\ddot{y} + (y^2 - 1)\dot{y} + y = 0$$

或在状态空间中, 其中:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -(x_1^2 - 1)x_2 - x_1$$

$y(t)$ 的时间历程和相平面图 (即 x_2 对 x_1) 如图 2.3.2 所示。

示例 2.7 (刚性机器人动力学). 刚性机器人由以下方程描述:

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) = \tau$$

其中 $M(q)$ 是 $n \times n$ 惯性矩阵, q 及其导数是 $n \times 1$ 广义坐标向量, $V(q, \dot{q}), G(q)$ 和 τ 分别是包含速度相关力矩、重力力矩和输入力矩的 $n \times 1$ 向量。

在此示例中, 我们将集中于将 n 个耦合微分方程写入状态空间形式。事实上, 设状态向量 x 为:

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

输入向量为 $u = \tau$, 假设输出向量为 $y = q$ 。由于刚性机器人的一些特殊性质 (见第 2 章), 矩阵 $M(q)$ 已知是可逆的, 因此:

$$\ddot{q} = M^{-1}(q)[\tau - V_m(q, \dot{q})\dot{q} - G(q)]$$

或

$$\dot{x} = \begin{bmatrix} \dot{q} \\ M^{-1}(q)[\tau - V_m(q, \dot{q})\dot{q} - G(q)] \end{bmatrix}$$

其中

$$f(x) = \begin{bmatrix} x_2 \\ M^{-1}(x_1)[-V_m(x_1, x_2)x_2 - G(x_1)] \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ M^{-1}(x_1) \end{bmatrix}$$

2.3.2 离散时间系统

非线性、标量、离散时间、时不变系统由非线性、标量、常系数差分方程描述，例如：

$$f(y(k), y(k+1), \dots, y(k+n), u(k), u(k+1), \dots, u(k+n)) = 0 \quad (2.26)$$

其中 $y(\cdot)$ 和 u 如前所定义。状态变量的简单选择将导致：

$$x_i(k) = y(k+i-1), \quad i = 1, 2, \dots, n \quad (2.27)$$

或更紧凑地：

$$\begin{aligned} x(k+1) &= f(x(k), U(k)) \\ y(k) &= cx(k) \end{aligned} \quad (2.28)$$

其中 $U(k)$ 和 c 的定义类似于(??)中给出的那些。

示例 2.8 (逻辑斯蒂方程). 考虑标量系统

$$y(k+1) = y(k) + u(k)y(k)(1 - y(k))$$

这导致状态空间表示：

$$x(k+1) = x(k) + u(k)x(k)(1 - x(k)), \quad y(k) = x(k)$$

我们不强调离散非线性系统的研究，因为机器人由微分方程描述。然而，如第 4 章所讨论的，机器人控制器通常使用数字控制器实现。因此，能够像 [Åström and Wittenmark 1995]、[Franklin et al. 1997] 中讨论的那样在连续和离散时间描述的非线性动态系统之间进行转换将是有利的。

2.4 非线性系统与平衡点

在本节中，我们集中于由(??)描述的系统，附加要求是 $u(t)$ 被指定为状态 $x(t)$ 的函数，即：

$$\dot{x}(t) = f[t, x(t)], \quad x \in \mathbb{R}^n \quad (2.29)$$

这将使我们能够集中于分析问题。我们需要几个定义，我们现在介绍它们。

定义 2.1 (自治系统). 如果 $f[t, x(t)]$ 不明确依赖于时间 t ，即：

$$\dot{x}(t) = f[x(t)]$$

则系统(??)被称为自治的。

示例 2.9 (非自治系统). 在示例 2.3.1 中介绍的 2 个系统都是自治的, 而由以下方程描述的系统:

$$\dot{x} = \frac{x}{t}$$

不是自治的。

定义 2.2 (平衡点). 向量 $x_e \in \mathbb{R}^n$ 是(??)在时刻 t_0 的固定点或平衡点, 如果:

$$f[t_0, x_e] = 0$$

示例 2.10 (自治系统的平衡点). 由以下方程描述的系统:

$$\dot{x} = Ax$$

是自治的, 并且它在 \mathbb{R}^n 的原点处有一个平衡点。

注意:

- 如果系统是自治的, 那么时间 t_0 处的平衡点也是所有其他时间的平衡点。
- 如果 x_e 是非自治系统(??)在时间 t_0 的平衡点, 那么 x_e 是(??)对于所有 $t_1 \geq t_0$ 的平衡点。

示例 2.11 (非自治系统的平衡点). 考虑系统

$$\dot{x} = x - t$$

这是非自治的。它没有平衡点。虽然看起来在时间 $t_0 = 1$ 它有 2 个平衡点 $x_{e1} = -1$ 和 $x_{e2} = 1$ 。然而, 这些不是时间 $t \geq 1$ 的平衡点, 因为在时间 $t \geq 1$ 平衡的条件不成立。

一些书籍也使用术语”静止点”或”奇点”来表示平衡点。

示例 2.12 (阻尼摆). 回忆示例 2.3.1 中的摆, 让我们尝试找到它的平衡点。首先注意系统是自治的, 因此我们不需要指定特定的时间 t_0 , 然后注意, 当两者都满足时摆处于平衡:

$$x_2 = 0 \quad \text{和} \quad \sin(x_1) = 0$$

因此平衡点位于:

$$x_e = \begin{bmatrix} n\pi \\ 0 \end{bmatrix}, \quad n = 0, \pm 1, \pm 2, \dots$$

显然, 当摆以零速度垂直向上或垂直向下悬挂时, 摆处于平衡。

定义 2.3 (孤立平衡点). 如果存在 x_e 的邻域 N , 除了 x_e 本身外不包含其他平衡点, 则(??)在时间 t_0 的平衡点 x_e 被称为孤立的。

示例 2.13 (摆的平衡点). 摆的平衡点是孤立的。另一方面, 由 $\dot{x} = 0$ 描述的系统具有 \mathbb{R} 中的任何点作为其平衡点, 因此其平衡点都不是孤立的。

2.5 向量空间、范数与内积

在本节中，我们将讨论非线性微分方程及其解的一些性质。我们将需要许多概念，如向量空间和范数，我们将简要介绍它们。读者可参考 [Boyd and Barratt]、[Desoer and Vidyasagar 1975]、[Khalil 2001] 进行证明和详细说明。

2.5.1 线性向量空间

在大多数应用中，我们需要处理（线性）实向量和复向量空间，它们随后被定义。

定义 2.4 (线性向量空间). 一个实线性向量空间（相应地，复线性向量空间）是一个集合 V ，配备 2 个二元运算：加法 (+) 和标量乘法 (\cdot)，使得：

1. $x + y \in V$ ，如果 $x \in V$ 和 $y \in V$
2. $x + y = y + x$
3. $(x + y) + z = x + (y + z)$
4. 存在零元素 $0 \in V$ ，使得 $x + 0 = x$ ，对所有 $x \in V$
5. 对每个 $x \in V$ ，存在唯一的 $y = -x$ ，使得 $x + y = 0$
6. $\alpha \cdot x \in V$ ，如果 $x \in V$ 和 α 是实数（相应地，复数）
7. $\alpha \cdot (x + y) = \alpha \cdot x + \alpha \cdot y$
8. $(\alpha + \beta) \cdot x = \alpha \cdot x + \beta \cdot x$
9. $(\alpha\beta) \cdot x = \alpha \cdot (\beta \cdot x)$
10. 对每个 $x \in V$ ，我们有 $1 \cdot x = x$ ，其中 1 是 \mathbb{R} （相应地， \mathbb{C} ）中的单位元

示例 2.14 (向量空间). 以下是具有相关标量域的线性向量空间： \mathbb{R}^n 带有 \mathbb{R} ，和 \mathbb{C}^n 带有 \mathbb{C} 。

定义 2.5 (子空间). 向量空间 V 的子集 M 如果它本身是线性向量空间，则它是一个子空间。 M 成为子空间的一个必要条件是它包含零向量。

我们可以为向量空间配备许多函数。其中之一是内积，它将 V 中的两个向量带到 \mathbb{R} 或 \mathbb{C} 中的标量，另一个是向量的范数，它将 V 中的向量带到 \mathbb{R} 中的正值。以下部分讨论向量的范数，然后是内积的部分。

2.5.2 信号和系统的范数

范数是距离和长度概念的推广。由于稳定性理论通常关注某些向量和矩阵的大小，我们在这里简要介绍本书中将使用的一些范数。我们将首先考虑定义在向量空间 X 上的向量的范数，具有相关的实数标量域 \mathbb{R} ，然后引入矩阵诱导范数、函数范数，最后是系统诱导范数或算子增益。

向量范数

我们通过回顾最熟悉的赋范空间开始讨论范数，即具有常数条目的向量空间。在下文中， $|a|$ 表示实数 a 的绝对值，或者如果 a 是复数则表示 a 的幅值。

定义 2.6 (向量范数). 向量 x 的范数 $\|\cdot\|$ 是定义在向量空间 X 上的实值函数，使得：

1. $\|x\| > 0$ 对所有 $x \in X$ ，当且仅当 $x = 0$ 时 $\|x\| = 0$
2. $\|\alpha x\| = |\alpha| \cdot \|x\|$ 对所有 $x \in X$ 和任何标量 α
3. $\|x + y\| \leq \|x\| + \|y\|$ 对所有 $x, y \in X$

示例 2.15 (向量范数 (1)). 以下是 $X = \mathbb{R}^n$ 中的常见范数，其中 \mathbb{R}^n 是具有实分量的 $n \times 1$ 向量的集合。

1. 1-范数: $\|x\|_1 = \sum_{i=1}^n |x_i|$
2. 2-范数: $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ ，也称为欧几里得范数
3. p-范数: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$
4. ∞ -范数: $\|x\|_\infty = \max_i |x_i|$

示例 2.16 (向量范数 (2)). 考虑向量

$$x = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

那么， $\|x\|_1 = 3$ ， $\|x\|_2 = \sqrt{5}$ ， $\|x\|_\infty = 2$ 。

我们现在给出 \mathbb{R}^n 中向量范数的一个重要性质，它将在后续中 useful。

引理 2.1 (范数等价性). 设 $\|x\|_a$ 和 $\|x\|_b$ 是向量 $x \in \mathbb{R}^n$ 的任意两个范数。那么存在有限的正常数 k_1 和 k_2 ，使得：

$$k_1 \|x\|_a \leq \|x\|_b \leq k_2 \|x\|_a$$

引理中的两个范数被称为等价的，这个特定性质对于 \mathbb{R}^n 上的任何两个范数都成立。

矩阵范数

在系统应用中，特定向量 x 可以由矩阵 A 操作以获得另一个向量 $y = Ax$ 。为了关联 x 和 Ax 的大小，我们定义诱导矩阵范数如下。

定义 2.7 (诱导矩阵范数). 设 $\|x\|$ 是给定的 $x \in \mathbb{R}^n$ 的范数。那么每个 $m \times n$ 矩阵 A 都有一个诱导范数，由下式定义：

$$\|A\|_i = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|$$

其中 \sup 表示上确界。

检查所提出的范数是否满足定义 2.5.3 的条件总是必要的。新定义的矩阵范数也可以证明满足：

$$\|AB\|_i \leq \|A\|_i \|B\|_i$$

对所有 $n \times m$ 矩阵 A 和所有 $m \times p$ 矩阵 B 。

函数范数

接下来，我们回顾时间相关函数和函数向量的范数。这些构成一类重要的信号，将在控制机器人时遇到。

定义 2.8 (函数范数). 设 $f(\cdot) : [0, \infty) \rightarrow \mathbb{R}$ 是一致连续函数。函数 f 如果对于任何 $\epsilon > 0$ ，存在 $\delta(\epsilon)$ 使得当 $|t_1 - t_2| < \delta$ 时 $|f(t_1) - f(t_2)| < \epsilon$ ，则称 f 是一致连续的。

那么，如果对于 $p \in [1, \infty)$ ：

$$\int_0^{\infty} |f(t)|^p dt < \infty$$

f 被称为属于 L_p 。

如果 f 是有界的，则称 f 属于 L_∞ ，即如果：

$$\sup_{t \geq 0} |f(t)| < \infty$$

其中 $\sup f(t)$ 表示 $f(t)$ 的上确界，即大于或等于 $f(t)$ 最大值的最小数。 L_1 表示具有有限绝对面积的信号集合，而 L_2 表示具有有限总能量的信号集合。

系统范数

接下来我们想研究多输入多输出 (MIMO) 系统对多维信号的影响。换句话说，时变向量 $u(t)$ 通过 MIMO 系统 H 时会发生什么？设 H 是一个具有 m 输入和 l 输出的系统，因此其对输入 $u(t)$ 的输出由下式给出：

$$y(t) = (Hu)(t)$$

如果 u 属于 L_p 时 Hu 属于 L_p , 并且存在有限常数 $\gamma > 0$ 和 b 使得:

$$\|y\|_p \leq \gamma \|u\|_p + b$$

则我们说 H 是 L_p 稳定的。

如果 $p = \infty$, 系统被称为有界输入有界输出 (BIBO) 稳定。

定义 2.9 (L_p 增益). 系统 H 的 L_p 增益记为 $\gamma_p(H)$, 是使得存在有限 b 来验证该方程的最小 γ 。

因此, 增益 γ_p 表征了输入信号通过系统时的放大。以下引理表征了线性系统的增益, 可在 [Boyd and Barratt] 中找到。

引理 2.2 (线性系统的增益). 给定线性系统 H , 使得输入 $u(t)$ 产生输出 $y(t) = (h*u)(t) = \int_0^t h(t-\tau)u(\tau)d\tau$, 并假设 H 是 BIBO 稳定的, 那么:

1. $\gamma_\infty(H) = \int_0^\infty |h(t)|dt$
2. $\gamma_2(H) = \sup_\omega |H(j\omega)|$
3. $\gamma_1(H)$ 不提供简单的解析解, 但可以通过线性规划有效地计算

2.5.3 内积

内积是向量空间中两个向量之间的运算, 它将使我们能够定义几何概念, 如正交性和傅里叶级数等。以下定义了内积。

定义 2.10 (内积). 在向量空间 V 上定义的内积是一个函数 $\langle \cdot, \cdot \rangle$, 从 V 到 F , 其中 F 是 \mathbb{R} 或 \mathbb{C} , 使得对所有 $x, y, z \in V$:

1. $\langle x, y \rangle = \langle y, x \rangle^*$, 其中 $\langle \cdot, \cdot \rangle^*$ 表示复共轭
2. $\langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle$
3. $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$, 对所有标量 α
4. $\langle x, x \rangle \geq 0$, 其中仅当 $x = 0_V$ 时取等号

我们可以为任何内积定义一个范数:

$$\|x\| = \sqrt{\langle x, x \rangle}$$

因此, 范数是一个更一般的概念: 一个向量空间可以有一个与之相关的范数, 但没有内积。然而, 反过来则不然。

2.5.4 矩阵性质

一些矩阵性质在研究动态系统的稳定性中起着重要作用。本书所需的性质收集在本节中。我们将假设读者熟悉基本的线性代数。

定义 2.11 (矩阵定性). 本定义中的所有矩阵都是实方阵。

- 正定: 如果对所有 $x \in \mathbb{R}^n, x \neq 0$, 有 $x^T A x > 0$, 则实 $n \times n$ 矩阵 A 是正定的。
- 半正定: 如果对所有 $x \in \mathbb{R}^n$, 有 $x^T A x \geq 0$, 则实 $n \times n$ 矩阵 A 是半正定的。
- 负定: 如果对所有 $x \in \mathbb{R}^n, x \neq 0$, 有 $x^T A x < 0$, 则实 $n \times n$ 矩阵 A 是负定的。
- 半负定: 如果对所有 $x \in \mathbb{R}^n$, 有 $x^T A x \leq 0$, 则实 $n \times n$ 矩阵 A 是半负定的。
- 不定: 如果对于某些 $x \in \mathbb{R}^n$, 有 $x^T A x > 0$, 而对于其他 $x \in \mathbb{R}^n$, 有 $x^T A x < 0$, 则 A 是不定的。

定理 2.1 (特征值测试). 设 $A = [a_{ij}]$ 是对称的 $n \times n$ 实矩阵。因此, A 的所有特征值都是实数。那么我们有以下结论:

- 正定: 如果所有特征值都是正的, 则实 $n \times n$ 矩阵 A 是正定的。
- 半正定: 如果所有特征值都是非负的, 则实 $n \times n$ 矩阵 A 是半正定的。
- 负定: 如果所有特征值都是负的, 则实 $n \times n$ 矩阵 A 是负定的。
- 半负定: 如果所有特征值都是非正的, 则实 $n \times n$ 矩阵 A 是半负定的。
- 不定: 如果某些特征值是正的而某些是负的, 则实 $n \times n$ 矩阵 A 是不定的。

定理 2.2 (Rayleigh-Ritz). 设 A 是实对称的 $n \times n$ 正定矩阵。设 λ_{\min} 是最小特征值, λ_{\max} 是 A 的最大特征值。那么, 对任何 $x \in \mathbb{R}^n$:

$$\lambda_{\min} x^T x \leq x^T A x \leq \lambda_{\max} x^T x$$

定理 2.3 (Gerschgorin 圆盘定理). 设 $A = [a_{ij}]$ 是对称的 $n \times n$ 实矩阵。假设:

$$r_i = \sum_{j \neq i} |a_{ij}|$$

如果所有对角元素都是正的, 即 $a_{ii} > 0$, 则矩阵 A 是正定的。

2.6 稳定性理论

我们研究的第一个稳定性概念涉及自由系统的行为，或等效地，具有给定输入的强制系统的行为。换句话说，我们研究系统平衡点对系统初始条件变化的稳定性。然而，在这样做之前，我们回顾一些基本定义。这些定义将以连续非线性系统的形式陈述，理解离散非线性系统允许类似结果，而线性系统只是非线性系统的特例。

设 x_e 是自由连续时间（可能是时变）非线性系统的平衡点（或固定）状态：

$$\dot{x} = f(x, t), \quad x \in \mathbb{R}^n \quad (2.30)$$

即 $f(x_e, t) = 0$ ，其中 x, f 是 $n \times 1$ 向量。

我们将首先回顾平衡点 x_e 的稳定性，理解 $x(t)$ 状态的稳定性总是可以通过稍后讨论的变量平移来获得。我们使用的稳定性定义可在 [Khalil 2001]、[Vidyasagar 1992] 中找到。

定义 2.12 (稳定性定义). 在本定义的所有部分中， x_e 是时间 t_0 处的平衡点， $\|\cdot\|$ 表示先前定义的任何函数范数。

1. 稳定性：如果在时刻 t_0 足够接近 x_e 开始的状态将始终在以后的时间保持在 x_e 附近，则 x_e 在李雅普诺夫意义下是稳定的 (SL)。更精确地说，如果对任何给定的 $\epsilon > 0$ ，存在正数 $\delta(\epsilon, t_0)$ ，使得如果：

$$\|x_0 - x_e\| < \delta$$

则：

$$\|x(t) - x_e\| < \epsilon, \quad \forall t \geq t_0$$

x_e 在李雅普诺夫意义下是稳定的，如果它对于任何给定的 t_0 都是稳定的。

2. 不稳定性：如果无论状态多接近 x_e 开始，它都不会在以后的时间被限制在 x_e 的附近，则 x_e 在李雅普诺夫意义下是不稳定的 (UL)。换句话说，如果它在 t_0 不稳定，则 x_e 是不稳定的。
3. 收敛性：如果从接近 x_e 开始的状态最终收敛到 x_e ，则 x_e 在时间 t_0 是收敛的 (C)。换句话说，如果对任何正数 ϵ ，存在正数 $\delta_1(t_0)$ 和正数 $T(\epsilon_1, x_0, t_0)$ ，使得如果：

$$\|x_0 - x_e\| < \delta_1$$

则：

$$\|x(t) - x_e\| < \epsilon, \quad \forall t \geq t_0 + T$$

x_e 是收敛的，如果它对于任何 t_0 都是收敛的。

4. 渐近稳定性: 如果从充分接近 x_e 开始的状态将保持在附近并最终收敛到它, 则 x_e 在时间 t_0 是渐近稳定的 (AS)。更准确地说, 如果 x_e 在时间 t_0 既是收敛的又是稳定的, 则它在时间 t_0 是渐近稳定的。如果它对任何 t_0 都是渐近稳定的, 则 x_e 是渐近稳定的。
5. 全局渐近稳定性: 如果任何初始状态将保持在 x_e 附近并最终收敛到它, 则 x_e 是全局渐近稳定的 (GAS)。换句话说, 如果它在 t_0 是稳定的, 并且如果每个 $x(t)$ 随着时间趋于无穷大收敛到 x_e , 则 x_e 是全局渐近稳定的。如果它对任何 t_0 都是全局渐近稳定的, 则 x_e 是全局渐近稳定的, 并且系统在这种情况下被称为全局渐近稳定的, 因为它只能有一个平衡点 x_e 。

注意, 稳定性和渐近稳定性是局部概念, 因为如果初始扰动 δ 太大, 随后的状态 $x(t)$ 可能会任意远离 x_e 。因此, 存在以 x_e 为中心的区域:

$$\|x - x_e\| < r$$

使得对从该区域开始的任何状态都会产生稳定性和渐近稳定性, 但对从该区域外开始的状态则不会。该区域被称为 x_e 的吸引域。如果吸引域是 \mathbb{R}^n , 则平衡点 x_e 是全局渐近稳定的。

还应注意, 所有先前的稳定性定义都依赖于初始时间 t_0 , 因此吸引域可能随变化的初始时间而变化。如果系统(??)独立于时间 (或自治), 则定义??中的稳定性概念实际上独立于 t_0 , 并且它们将等同于接下来定义的稳定性的概念。另一方面, 即使系统(??)是时间相关的, 我们也希望其稳定性性质不依赖于 t_0 , 因为这将为我们提供所需的鲁棒性程度。这使我们定义均匀稳定性概念 [Khalil 2001]。

定义 2.13 (均匀稳定性). 在本定义的所有部分中, x_e 是时间 t_0 处的平衡点。

1. 均匀稳定性: 如果定义??中的 $\delta(\epsilon, t_0)$ 独立于 t_0 , 则 x_e 在 $[t_0, \infty)$ 上是均匀稳定的 (US)。
2. 均匀收敛性: 如果定义??中的 $\delta_1(t_0)$ 和 $T(\epsilon_1, x_0, t_0)$ 可以独立于 t_0 选择, 则 x_e 在 $[t_0, \infty)$ 上是均匀收敛的 (UC)。
3. 均匀渐近稳定性: 如果 x_e 既是均匀稳定的又是均匀收敛的, 则它在 $[t_0, \infty)$ 上是均匀渐近稳定的 (UAS)。
4. 全局均匀渐近稳定性: 如果 x_e 是均匀稳定的和均匀收敛的, 则它是全局均匀渐近稳定的 (GUAS)。
5. 全局指数稳定性: 如果对所有 $x_0 \in \mathbb{R}^n$, 存在 $\alpha > 0$ 和 $\beta \geq 0$ 使得:

$$\|x(t)\| \leq \beta \|x_0\| e^{-\alpha(t-t_0)}$$

则 x_e 是全局指数稳定的 (GES)。

注意，GES 意味着 GUAS。

在许多情况下，只需要状态大小方面的界限。这是比李雅普诺夫稳定性更不严格的要求。研究下面有界性定义与定义??中李雅普诺夫稳定性定义之间的细微差别是有益的。

定义 2.14 (有界性). 1. 有界性: 如果从接近 x_e 开始的状态永远不会离得太远, 则 x_e 在时间 t_0 是有界的 (B)。换句话说, 如果对于每个 $\delta > 0$, 使得 $\|x_0 - x_e\| < \delta$, 存在正数 $\epsilon(r, t_0) < \infty$, 使得对所有 $t \geq t_0$:

$$\|x(t) - x_e\| < \epsilon$$

如果它对于任何 t_0 都是有界的, 则 x_e 是有界的。

2. 均匀有界性: 如果 $\epsilon(r, t_0)$ 可以独立于 t_0 , 则 x_e 在 $[t_0, \infty)$ 上是均匀有界的 (UB)。
3. 均匀最终有界性: 如果从接近 x_e 开始的状态最终会变得有界, 则 x_e 被称为均匀最终有界的 (UUB)。更准确地说, 如果对任何 $\delta > 0, \epsilon > 0$, 存在有限时间 $T(\epsilon, \delta)$, 使得每当 $\|x_0 - x_e\| < \delta$ 时, 以下满足:

$$\|x(t) - x_e\| < \epsilon, \quad \forall t \geq T(\epsilon, \delta)$$

4. 全局均匀最终有界性: 如果对 $\epsilon > 0$, 存在有限时间 $T(\epsilon)$ 使得:

$$\|x(t) - x_e\| < \epsilon, \quad \forall t \geq T(\epsilon)$$

则 x_e 被称为全局均匀最终有界的 (GUUB)。

2.7 李雅普诺夫稳定性定理

李雅普诺夫稳定性理论涉及由微分方程描述的非强制非线性系统的行为:

$$\dot{x} = f(t, x), \quad x \in \mathbb{R}^n \quad (2.31)$$

其中不失一般性, 原点是(?)的平衡点。读者可能认为这样的理论是不必要的, 因为我们在前一节的示例中所要做的就是求解微分方程, 并研究状态向量范数的时间演化。至少有两个原因说明为什么需要李雅普诺夫理论。第一个是李雅普诺夫理论将允许我们确定特定平衡点的稳定性, 而无需实际求解微分方程。这, 如任何非线性微分方程的学生所熟知的那样, 是一个巨大的节省。第二个相关的原因是, 李雅普诺夫理论为我们提供了稳定性问题的定性结果, 可用于设计非线性动力系统的稳定控制器。

我们将首先假设满足(?)具有唯一解的任何必要条件 [Khalil 2001]、[Vidyasagar 1992]。对应于 $x(t_0) = x_0$ 的唯一解是 $x(t, t_0, x_0)$, 将简单地记为 $x(t)$ 。在我们实际引入李雅普诺夫定理之前, 我们回顾某些函数类, 这些函数将简化李雅普诺夫定理的陈述。

2.7.1 K 类函数

考虑连续函数 $\alpha: \mathbb{R} \rightarrow \mathbb{R}$ 。

定义 2.15 (K 类函数). 如果满足以下条件, 我们说 α 属于 K 类:

1. $\alpha(0) = 0$
2. 对所有 $x > 0$, $\alpha(x) > 0$
3. α 是非递减的, 即对所有 $x_1 > x_2$, 有 $\alpha(x_1) \geq \alpha(x_2)$

示例 2.17 (K 类函数). 函数 $\alpha(x) = x^2$ 是一个 K 类函数。函数 $\alpha(x) = x^2 + 1$ 不是 K 类函数, 因为条件 (1) 不成立。另一方面, $\alpha(x) = -x^2$ 不是 K 类函数, 因为条件 (2) 和 (3) 不成立。

定义 2.16 (正定函数). 在以下内容中, $\mathbb{R}_+ = [0, \infty)$ 。

1. 局部正定: 如果存在 K 类函数 $\alpha(\cdot)$ 和 \mathbb{R}^n 原点的邻域 N 使得对所有 $t \geq 0$ 和所有 $x \in N$:

$$V(t, x) \geq \alpha(\|x\|)$$

则连续函数 $V: \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ 是局部正定的 (l.p.d)。

2. 正定: 如果 $N = \mathbb{R}^n$, 则函数 V 被称为正定的 (p.d)。
3. 负定和局部负定: 如果 $-V$ 是 (局部) 正定的, 我们说 V 是 (局部) 负定的 (n.d)。

定义 2.17 (递减函数). 在以下内容中:

1. 局部递减: 如果存在 K 类函数 $\beta(\cdot)$ 和 \mathbb{R}^n 原点的邻域 N 使得对所有 $t \geq 0$ 和所有 $x \in N$:

$$V(t, x) \leq \beta(\|x\|)$$

则连续函数 $V: \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ 是局部递减的。

2. 递减: 如果 $N = \mathbb{R}^n$, 我们说 V 是递减的。

定义 2.18 (沿轨迹的导数). 给定连续可微函数 $V: \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ 以及微分方程组(?), V 沿(??)的导数被定义为函数 $\dot{V}: \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$, 由下式给出:

$$\dot{V}(t, x) = \frac{\partial V(t, x)}{\partial t} + \frac{\partial V(t, x)}{\partial x} f(t, x)$$

2.7.2 李雅普诺夫定理

我们现在准备陈述李雅普诺夫定理，我们将它们分组在定理??中。证明见 [Khalil 2001]、[Vidyasagar 1992]。

定理 2.4 (李雅普诺夫稳定性定理). 给定非线性系统:

$$\dot{x} = f(t, x), \quad x \in \mathbb{R}^n$$

在原点处有平衡点，即 $f(t, 0) = 0$ ，并设 N 是大小为 ϵ 的原点的邻域，即:

$$N = \{x \in \mathbb{R}^n; \|x\| \leq \epsilon\}$$

其中，**正定**是指 $V(t, 0) = 0$ 且对所有 $x \neq 0$ 有 $V(t, x) > 0$ ；**半负定**是指对所有 x 有 $\dot{V}(t, x) \leq 0$ ；**负定**是指 $\dot{V}(t, 0) = 0$ 且对所有 $x \neq 0$ 有 $\dot{V}(t, x) < 0$ ；**递减**是指存在 K 类函数 α 使得 $V(t, x) \leq \alpha(\|x\|)$ ；**径向无界**是指当 $\|x\| \rightarrow \infty$ 时 $V(t, x) \rightarrow \infty$ 。 \dot{V} 表示 V 沿系统轨迹的全导数： $\dot{V} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x)$ 。

那么:

1. 稳定性: 如果存在标量函数 $V(t, x)$ 具有连续偏导数，使得对 $x \in N$:

(a) $V(t, x)$ 是正定的

(b) \dot{V} 是半负定的

则原点在李雅普诺夫意义下是稳定的。

2. 均匀稳定性: 如果除了 (a) 和 (b) 之外， $V(t, x)$ 对 $x \in N$ 是递减的，则原点是均匀稳定的。

3. 渐近稳定性: 如果 $V(t, x)$ 满足 (a) 并且对 $x \in N$ 是负定的，则原点是渐近稳定的。

4. 全局渐近稳定性: 如果 $V(t, x)$ 验证 (a)，并且对所有 $x \in \mathbb{R}^n$ (即 $N = \mathbb{R}^n$) $\dot{V}(t, x)$ 是负定的，则原点是全局渐近稳定的。

5. 均匀渐近稳定性: 如果 $V(t, x)$ 满足 (a)， $V(t, x)$ 是递减的，并且 $\dot{V}(t, x)$ 对 $x \in N$ 是负定的，则原点是 UAS。

6. 全局均匀渐近稳定性: 如果 $N = \mathbb{R}^n$ ，并且如果 $V(t, x)$ 满足 (a)， $V(t, x)$ 是递减的， $\dot{V}(t, x)$ 是负定的，并且 $V(t, x)$ 是径向无界的 (即如果时间趋于一致，它趋于无穷大当 $\|x\| \rightarrow \infty$)，则原点是 GUAS。

7. 指数稳定性: 如果存在正常数 α, β, γ 使得:

$$\alpha \|x\|^2 \leq V(t, x) \leq \beta \|x\|^2$$

$$\dot{V}(t, x) \leq -\gamma \|x\|^2$$

则原点是指数稳定的。

8. 全局指数稳定性: 如果它对所有 $x \in \mathbb{R}^n$ 是指数稳定的, 则原点是全局指数稳定的。

定理中的函数 $V(t, x)$ 被称为李雅普诺夫函数。注意, 该定理提供了原点稳定性的充分条件, 无法为特定系统提供李雅普诺夫函数候选并不表示原点的稳定性。

注: 读者可以通过[交互式可视化工具](#)直观理解李雅普诺夫稳定性定理中的各种稳定性类型 (稳定、渐近稳定、指数稳定、全局渐近稳定、不稳定), 观察不同条件下的状态轨迹演变。

2.7.3 自治情况

假设开环系统不是自治的, 即不明确依赖于 t , 那么可以获得与时间无关的李雅普诺夫函数候选 $V(x)$, 并且正定条件大大简化, 如下所述。

引理 2.3. 如果 $V(0) = 0$ 并且对所有 $x \neq 0$, $V(x) > 0$, 则时间不变的连续函数 $V(x)$ 是正定的。如果上述在邻域中成立, 则它是局部正定的。

注意, 条件 $V(0) = 0$ 不是必需的, 并且只要 $V(0)$ 有上界, 李雅普诺夫结果就无需修改而成立。

利用上述简化, 李雅普诺夫结果成立, 只是均匀和常规稳定性结果之间没有区别。

定理 2.5 (LaSalle 定理). 给定自治非线性系统:

$$\dot{x} = f(x)$$

并设原点是平衡点。那么:

1. 渐近稳定性: 假设已经找到了一个李雅普诺夫函数 $V(x)$, 使得对 $x \neq 0$, $V(x) > 0$ 并且 $\dot{V} \leq 0$ 。那么, 当且仅当 $\dot{V} = 0$ 仅在 $x = 0$ 处时, 原点是渐近稳定的。
2. 全局渐近稳定性: 如果上述 $\dot{V} = 0$ 并且 $V(x)$ 是径向无界的, 则原点是 GAS。

2.7.4 线性时不变情况

在所考虑的系统是线性且时不变的情况下，李雅普诺夫理论发展得很好，并且李雅普诺夫函数的选择很简单。事实上，在这种情况下，定义 2.6.1 中的各种稳定性概念是相同的。李雅普诺夫理论然后提供了稳定性的必要和充分条件，如本节所讨论的。证明参考 [Khalil 2001]。

定理 2.6. 给定线性时不变系统：

$$\dot{x} = Ax$$

当且仅当存在方程的正定解 P 时，系统是稳定的：

$$A^T P + PA = -Q$$

其中 Q 是任意正定矩阵。

注意，在最后这个定理中获得了整个系统的稳定性，因为在这种情况下，原点是唯一的平衡点，其稳定性等同于系统的稳定性。此外，没有提到暗示什么类型的稳定性，因为在线性、时不变系统的非常特殊情况下，所有稳定性概念都是等价的 [Khalil 2001]。还要注意，这个结果等同于测试 A 的所有特征值是否具有负实部 [Kailath 1980]。

定理 2.7 (Krasovskii 定理). 考虑自治非线性系统 $\dot{x} = f(x)$ ，原点是平衡点。设 $A(x) = \partial f / \partial x$ 。那么，原点为 AS 的一个充分条件是存在 2 个对称正定矩阵 P 和 Q ，使得对所有 $x \neq 0$ ，矩阵：

$$F(x) = A(x)^T P + PA(x) + Q$$

在某个以原点为中心的球 B 中是正定的。函数 $V(x) = f(x)^T P f(x)$ 然后是系统的李雅普诺夫函数。如果 $B = \mathbb{R}^n$ 并且如果 $V(x)$ 是径向无界的，则系统是 GAS。

2.8 输入/输出稳定性

当处理非线性系统时，李雅普诺夫意义下的稳定性并不一定意味着有界输入将导致有界输出。这个事实在下一个示例中显示。

然而，我们需要讨论有界输入将导致有界输出的条件 [Boyd and Barratt]、[Desoer and Vidyasagar 1975]。这实际上在讨论系统诱导范数时已经提出（见定义 2.5.7），当前的讨论应作为这些概念与李雅普诺夫稳定性的对比。

定义 2.19 (BIBO 稳定性). 如果对于任何 $\|u(t)\| \leq M < \infty$ ，存在有限 $\gamma > 0$ 和 b 使得：

$$\|y(t)\| \leq \gamma M + b$$

则动态系统(??)-(??)是有界输入有界输出 (BIBO) 稳定的。

注意，BIBO 稳定性意味着所有平衡状态的均匀有界性。

2.9 高级稳定性结果

在本节中，我们回顾一些高级稳定性概念。这些结果将用于显示使用鲁棒或自适应控制器时系统的闭环稳定性。如果读者只对实现这些控制器感兴趣，可以跳过本节。另一方面，任何对设计新控制器感兴趣的人都应该了解这里介绍的结果。

2.9.1 被动系统

给定图 2.9.1 所示的非线性系统，我们感兴趣的是仅基于输入-输出测量来研究这种系统的稳定性。受网络理论中能量概念的启发，例如：

$$[\text{储存的能量}] = [\text{外部功率输入}] + [\text{内部功率产生}]$$

人们可以研究所有类型系统的内部稳定性。一般来说，外部功率输入是输入努力或流 u 与输出流或努力 y 的标量积 $y^T u$ 。最后一个方程然后采取以下形式：

$$\dot{V}(x) = y^T u - g(t) \quad (2.32)$$

在许多情况下（例如孤立系统）， $g(t) = 0$ ，可以使用储存的能量：

$$V(x) = \int_0^t y^T(\tau) u(\tau) d\tau \quad (2.33)$$

作为李雅普诺夫函数候选。

非线性系统的被动性定义如下 [Narendra and Taylor 1973]、[Ortega et al. 1998]。

定义 2.20 (被动性). 考虑图 2.9.1 所示的系统，并假设它具有相同数量的输入和输出，即 $u(t)$ 和 $y(t)$ 具有相同的维数。

1. 被动性：如果对所有有限 $T > 0$ 和一些 $\gamma > -\infty$ ：

$$\int_0^T y^T(t) u(t) dt \geq -\gamma$$

则系统被称为被动的。

2. 严格被动性：如果存在 $\delta > 0$ 和 $\gamma > -\infty$ 使得对所有有限 $T > 0$ ：

$$\int_0^T y^T(t) u(t) dt \geq \delta \int_0^T \|u(t)\|^2 dt - \gamma$$

则系统被称为严格被动的。

被动系统实际上是一个不产生能量的系统。

2.9.2 正实系统

如果所考虑的系统是线性且时不变的，则被动性等同于正实性，可以在频域中测试 [Narendra and Taylor 1973]。事实上，让我们描述正实系统并讨论它们的一些性质。考虑多输入多输出线性时不变系统：

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

其中 x 是 n 向量， u 是 m 向量， y 是 p 向量， A, B, C 和 D 具有适当的维数。相应的传递函数矩阵是：

$$P(s) = C(sI - A)^{-1}B + D$$

我们将假设系统具有相等数量的输入和输出，即 $p = m$ 。

定义 2.21 (正实). 如果满足以下条件，则 $m \times m$ 矩阵 $T(s)$ 的正则有理函数（不恒为零）是正实的或 PR：

1. $T(s)$ 的所有元素在区域 $\text{Re}(s) > 0$ 中没有极点
2. $T(s)$ 在 $j\omega$ 轴上的任何极点都是简单的，具有正定留数
3. 矩阵 $\text{He}[T(s)]$ 对 $\text{Re}(s) > 0$ 是半正定的

定义 2.22 (严格正实). 如果满足以下条件，则 $m \times m$ 矩阵 $T(s)$ 的正则有理函数（不恒为零）是严格正实的或 SPR：

1. $T(s)$ 的所有元素在区域 $\text{Re}(s) \geq 0$ 中没有极点
2. 矩阵 $\text{He}[T(s)]$ 对 $\text{Re}(s) > 0$ 是正定的

2.9.3 Lure 问题

考虑以下反馈互连系统：

$$\begin{aligned}\dot{x} &= Ax + bu \\ y &= c^T x + du \\ u &= -\phi(t, y)\end{aligned}\tag{2.34}$$

其中 ϕ 在两个参数中都是连续的。Lure 随后陈述了以下绝对稳定性问题，后来被称为 Lure 问题：假设上述方程描述的系统给定，其中：

1. A 的所有特征值都具有负实部，或者 A 在原点处有一个特征值，而其余的特征值在开左半平面 (OLHP) 中

2. (A, b) 是可控的
3. (c, A) 是可观测的
4. 非线性 $\phi(., .)$ 满足:

$$(a) \phi(t, 0) = 0; t \geq 0$$

$$(b) 0 \leq y\phi(t, y) \leq ky^2; y \in \mathbb{R}$$

那么, 找到线性系统 (A, B, C, D) 上的条件, 使得 $x = 0$ 是闭环系统的 GAS 平衡点。

2.9.4 MKY 引理

以下引理是 Meyer-Kalman-Yakubovich (MKY) 引理的版本, 出现在 [Narendra and Taylor 1973]、[Khalil 2001] 等地, 在设计机器人的自适应控制器时将非常有用。

引理 2.4 (Meyer-Kalman-Yakubovitch). 设系统(??)且 $D = 0$ 是可控的。那么传递函数 $c(sI - A)^{-1}b$ 是 SPR 当且仅当:

1. 对任何对称正定 Q , 存在对称正定 P 解李雅普诺夫方程:

$$A^T P + PA = -Q$$

2. 矩阵 B 和 C 满足:

$$C = B^T P$$

MKY 引理给出了传递矩阵具有一定鲁棒性的条件。注意, 条件取决于输入和输出矩阵, 因此特定系统可能对某些输入/输出对的选择是 SPR, 而对其他选择则不是 SPR。

2.10 有用的定理与引理

考虑图 2.9.2 所示的框图。标记为 H_1 和 H_2 的块代表两个系统 (线性或非线性), 它们对输入 e_1 和 e_2 进行如下操作:

$$y_1 = H_1 e_1 = H_1(u_1 - y_2)$$

$$y_2 = H_2 e_2 = H_2(u_2 + y_1)$$

2.10.1 小增益定理

定理 2.8 (小增益定理). 设 $H_1 : L_{pe} \rightarrow L_{pe}$ 和 $H_2 : L_{pe} \rightarrow L_{pe}$ 。因此 H_1 和 H_2 对所有 $T \in [0, \infty)$ 满足不等式:

$$\|y_1\|_T \leq \gamma_1 \|e_1\|_T + \beta_1$$

$$\|y_2\|_T \leq \gamma_2 \|e_2\|_T + \beta_2$$

并且假设 $\gamma_1 \gamma_2 < 1$ 。如果:

$$u_1, u_2 \in L_p$$

则 $e_1, y_2 \in L_p$ 和 $y_1, e_2 \in L_p$ 。

基本上, 小增益定理指出, 如果环增益小于 1, 则两个系统的反馈互连是 BIBO 稳定的。换句话说, 如果一个信号遍历反馈环并在幅度上减小, 则闭环系统不会变得不稳定。

2.10.2 总稳定性定理

定理 2.9 (总稳定性). 考虑由下式描述的状态空间系统:

$$\dot{x} = Ax + f(t, x) + g(t, x)$$

其中:

1. 系统 $\dot{x} = Ax$ 是指数稳定的, 即存在某些 $a > 0$ 和 $K \geq 1$, 使得 $\|e^{At}\| \leq Ke^{-at}$
2. $f(t, 0) = 0$, 即原点是 $f(t, x)$ 的平衡点
3. $\|f(t, x_1) - f(t, x_2)\| \leq \beta_1 \|x_1 - x_2\|$, 对某些 $\beta_1 > 0$
4. $\|g(t, x)\| \leq \beta_2 r$, 对某些 $\beta_2 > 0$
5. $\|g(t, x_1) - g(t, x_2)\| \leq \beta_2 \|x_1 - x_2\|$
6. $\|x_0\| \leq r/K$

那么, 存在(??)的唯一解 $x(t)$ 并且:

$$\|x(t)\| \leq k_1 e^{-\alpha t} \|x(0)\| + k_2 r$$

总稳定性定理将用于设计控制器, 使系统的线性部分指数稳定。实际上, 这个定理保证如果系统的线性部分“非常”稳定(指数稳定), 则有界非线性的去稳定效应可能不足以使系统去稳定, 状态将保持有界。

2.10.3 Bellman-Gronwall 引理

引理 2.5 (Bellman-Gronwall). 设 $x(\cdot), a(\cdot), b(\cdot) : [0, \infty) \rightarrow [0, \infty)$, 且 $T \geq 0$. 假设对所有 $t \in [0, T]$, 以下不等式成立:

$$x(t) \leq \int_0^t a(\tau)x(\tau)d\tau + b(t)$$

那么对所有 $t \in [0, T]$:

$$x(t) \leq \int_0^t a(\tau)b(\tau) \exp\left(\int_\tau^t a(\sigma)d\sigma\right) d\tau + b(t)$$

如果 $b(t)$ 进一步是常数, 则以下成立:

$$x(t) \leq b \exp\left(\int_0^t a(\tau)d\tau\right)$$

2.10.4 Barbalat 引理

引理 2.6 (Barbalat). 设 $f(t)$ 是 t 的可微函数。

- 第一版本: 如果 $\dot{f}(t)$ 是一致连续的并且 $\lim_{t \rightarrow \infty} f(t) = k < \infty$, 则 $\lim_{t \rightarrow \infty} \dot{f}(t) = 0$ 。
- 第二版本: 如果 $f(t) \geq 0$, $\dot{f}(t) \leq 0$ 并且 \ddot{f} 有界, 则 $\lim_{t \rightarrow \infty} \dot{f}(t) = 0$ 。

2.11 线性控制器设计

控制设计的目的是使机器人以可预测和期望的方式响应一组输入信号。在本节中, 我们回顾如何设计线性控制器, 以使机器人-控制器组合的行为是可接受的。机器人控制器的第一个要求显然是其稳定性。因此, 控制器的第一个功能是稳定机器人在空间中移动时的状态。事实上, 我们现在可以提出我们希望解决的问题:

线性控制设计问题: 考虑一个 LTI 系统。设计一个对输出 $y(t)$ 进行操作而不进行微分并生成输入 $u(t)$ 的反馈控制器, 使得系统从任何初始状态 $x(0)$ 在有限时间内达到指定的期望最终状态 x_d 。这个问题有 2 个部分:

1. 可控性: 如果 $y = x$, 问题可以解决吗?
2. 可观测性: 如果是这样, 我如何从 y 得到 x ?

在下一节中, 我们介绍这些概念并找到测试, 使我们能够回答所提出的问题。

定义 2.23 (可控性). 如果存在输入 $u(t)$, 对 $0 \leq t \leq t_1$, 使得 $x(t_1) = 0$, 对某些有限 t_1 , 则状态 x_0 是可控的。如果所有 $x_0 \in \mathbb{R}^n$ 都是可控的, 则系统本身被称为可控的。

定理 2.10 (可控性测试). LTI 系统完全可控的必要和充分条件是 $n \times nm$ 矩阵:

$$C = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$$

是满秩的, 即 $\text{rank}C = n$ 。

接下来, 我们展示可控性足以通过放置 A 矩阵的特征值来稳定系统(??)。

定理 2.11. 设 (A, b) 是可控的。那么, 存在常数增益矩阵 K , 使得 $u = -Kx + v$ 将把 $A - bK$ 的特征值放置在 s 平面中的任何位置。

在单输入情况下, 可以使用 Ackermann 公式找到放置特征值所需的状态反馈增益 (参见例如 [Antsaklis and Michel 1997])。假设期望的闭环特征值被指定为方程的根:

$$\phi(s) = s^n + d_{n-1}s^{n-1} + \dots + d_1s + d_0 = 0 \quad (2.35)$$

那么, 状态反馈控制器由下式给出:

$$K = [0 \quad 0 \quad \dots \quad 0 \quad 1]C^{-1}\phi(A) \quad (2.36)$$

其中 C 是(??)的可控性矩阵, $\phi(A)$ 是通过在 A 处求值 $\phi(s)$ 获得的矩阵, 即:

$$\phi(A) = A^n + d_{n-1}A^{n-1} + \dots + d_1A + d_0I \quad (2.37)$$

定义 2.24 (可观测性). 如果知道 $u(t), y(t); 0 \leq t \leq t_1$ 足以唯一确定 x_0 , 则状态 $x(0) = x_0$ 被称为可观测的。注意一旦 $x(0)$ 已知, $t \geq 0$ 的 $x(t)$ 也就知道了。如果每个初始状态都是可观测的, 则系统被称为完全可观测的。

定理 2.12 (可观测性测试). LTI 系统完全可观测的必要和充分条件是 $np \times n$ 矩阵:

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

是满秩的。

通过结合可观测性和可控性的概念, 我们可以设计解决线性控制设计问题的补偿器。事实上, 以下定理总结了线性控制设计。

定理 2.13. 当且仅当状态空间实现既是可观测的又是可控的, 线性控制设计问题对系统(??)是可解的。

注: 读者可以通过[交互式可视化工具](#)实时调节 PID 参数, 观察 K_p 、 K_i 、 K_d 对机器人关节跟踪性能的影响。

2.12 总结与注记

本章旨在回顾将在本书其余部分中使用的各种控制理论概念。重点是包含足够的材料，以便控制理论背景很少或没有的读者能够跟随机器人控制器的设计。经常进行简化以集中于本书中研究的系统，即机械系统和由拉格朗日-欧拉方程描述的系统。特别是，双重积分器系统被详细说明，因为它将来自在机械机械臂上应用初步非线性反馈。在第3章中，我们介绍了刚性机器人机械臂的动力学描述，在随后的章节中，本章回顾的控制概念将在这些机械臂上实现。

这里回顾的大多数结果的更多细节可在 [Kailath 1980]、[Antsaklis and Michel 1997] 关于线性系统情况的书籍中找到，以及在 [Khalil 2001]、[Vidyasagar 1992] 关于非线性系统情况的书籍中找到。

参考文献

- Anderson et al. 1986 B.D.O.Anderson and R.Bitmead and C.R.Johnson, Jr. and P.V.Kokotovic and R.L.Kosut and I.M.Y.Mareels and L.Praly and B.D.Riedle. "Stability of Adaptive Systems: Passivity and Averaging Analysis". MIT Press, Cambridge, MA, 1986.
- Antsaklis and Michel 1997 P.Antsaklis and A.N.Michel. "Linear Systems". McGraw Hill, New York, 1997.
- Åström and Wittenmark 1995 K.Åström and B.Wittenmark. "Adaptive Control". Addison-Wesley, Reading, MA, 2nd edition, 1995.
- Åström and Wittenmark 1996 K.Åström and B.Wittenmark. "Computer-Controlled Systems: Theory and Design", Prentice-Hall, Englewood Cliffs, NJ, 1996.
- Boyd and Barratt S.Boyd and G.Barratt. "Linear Controller Design: Limits of Performance", Prentice-Hall, Englewood Cliffs, NJ, 1991.
- Desoer and Vidyasagar 1975 C.Desoer and M.Vidyasagar. "Feedback Systems: Input-Output Properties", Academic Press, New York, 1975.
- Franklin et al. 1997 G.Franklin and D.Powell and M.Workman. "Digital Control of Dynamic Systems", Addison-Wesley, Reading, MA, 1997.
- Gu 1988 G.Gu. "Stabilizability Conditions for Multivariable Uncertain systems via Output Feedback", IEEE Trans. Autom. Control, vol. AC-35, pp. 988-992, 1988.
- Hahn 1967 W.Hahn. "Theory and Applications of Lyapunov's Direct Method". Prentice-Hall, Englewood Cliffs, NJ, 1967.

- and Johnson 1991 R.Horn and C.Johnson. "Topics in Matrix Analysis". Cambridge University Press, New York, 1991.
- Kailath 1980 T.Kailath. "Linear Systems". Prentice-Hall, Englewood Cliffs, NJ, 1980.
- Khalil 2001 H.Khalil. "Nonlinear Systems". Prentice-Hall, Englewood Cliffs, NJ, 2001.
- and Lefschetz 1961 P.LaSale and S.Lefschetz. "Stability by Lyapunov's Direct Method". Academic Press, New York, 1961.
- Ljung 1999 L.Ljung. "System Identification-Theory for the User". Prentice-Hall, Englewood Cliffs, NJ, 1999.
- ra and Taylor 1973 K.S.Narendra and J.H.Taylor. "Frequency Domain Criteria for Absolute Stability". Academic Press, New York, 1973.
- ga and Spong 1988 R.Ortega and M.Spong. "Adaptive Motion Control of Rigid Robots: A Tutorial". Proc. IEEE Conf. Dec. and Cont., pp. 1575–1584, 1988.
- Ortega et al. 1998 R.Ortega and A.Loria and P.J.Nicklasson and H.Sira-Ramirez. "Passivity-based Control of Euler-Lagrange Systems". Springer-Verlag, Berlin, 1998.
- y and Bodson 1989 S.Sastry and M.Bodson. "Adaptive Control: Stability, Convergence, and Robustness". Prentice-Hall, Englewood Cliffs, NJ, 1989.
- Vidyasagar 1992 M.Vidyasagar. "Nonlinear Systems Analysis". Prentice-Hall, Englewood Cliffs, NJ, 1992.

Chapter 3

机器人动力学

本章为机器人机械臂控制研究提供必要的背景知识。机械臂动力学方程采用二阶微分方程形式和多种状态变量形式推导。介绍动力学的一些重要性质。展示如何包含驱动机械臂的电机的动力学，这些电机可能是电动或液压马达。

3.1 引言

机器人学是一个涉及众多不同学科的复杂领域，如物理学、材料性质、静力学与动力学、电子学、控制理论、视觉、信号处理、计算机编程和制造技术。本书的主要兴趣是机器人机械臂的控制。本章的目的是研究控制机器人所需的动力学方程。

对于那些需要控制理论背景的读者，提供了第二章。对于那些需要机器人机械臂基础知识背景的读者，在附录 A 中考察了机器人机械臂的几何结构，涵盖了基本机械臂构型、运动学和逆运动学。在那里还回顾了机械臂雅可比矩阵，这对于在笛卡尔坐标系或工作空间坐标系中的控制至关重要，期望的机械臂轨迹通常首先在这些坐标系中指定。

机器人动力学在 3.2 节中推导。推导中使用了拉格朗日力学。在 3.3 节中，我们回顾了机械臂动力学方程的一些基本性质，这些性质对于后续章节中推导机器人控制方案至关重要。这些性质总结在表 3.3.1 中，在整本书中都会引用。

3.2 节中的机械臂动力学采用二阶向量微分方程形式。在 3.4 节中，展示了将这种表述转换为状态变量描述的几种方法。状态变量描述是一阶向量微分方程，对于开发许多机械臂控制方案非常有用。本节使用了反馈线性化技术和哈密顿力学。

3.2 节中的机器人机械臂动力学以关节空间坐标给出。在 3.5 节中，展示了一种非常通用的方法，用于以任何期望的坐标（包括笛卡尔坐标或工作空间坐标，以及相机坐标系或参考系的坐标）获得机械臂动力学描述。

在 3.6 节中，分析了执行移动机器人机械臂连杆所需工作的电动或液压驱动器。展示了如何将驱动器的动力学模型并入机械臂动力学中，以提供机械臂加驱动器系统的完整动力学描述。这最终使我们能够进入下一章讨论机器人机械臂控制设计。

3.2 拉格朗日-欧拉动力学

为了控制设计目的，需要一个揭示系统动力学行为的数学模型。因此，在本节中推导机器人机械臂的动力学方程。我们的方法是推导机械臂的动能和势能，然后使用拉格朗日运动方程。

在本节中，我们忽略驱动机器人机械臂的电动或液压电机的动力学；驱动器动力学在 3.6 节中讨论。

3.2.1 力、惯量与能量

让我们回顾物理学中的一些基本概念，这将使我们能够更好地理解机械臂动力学 [Marion 1965]。在本小节中，我们使用粗体表示向量，普通字体表示其大小。

质量为 m 的物体以角速度 ω 绕半径为 r 的点旋转时，其向心力为

$$F_{cen} = -m\omega \times (\omega \times r) = m\omega^2 r \quad (3.1)$$

见图 3.2.1。线速度为

$$v = \omega \times r \quad (3.2)$$

在这种情况下意味着简单地 $v = \omega r$ 。

想象一个球体（即地球）以其中心为轴以角速度 ω_0 旋转。见图 3.2.2。在球面上以速度 v 移动的质量为 m 的物体所受的科里奥利力为

$$F_{cor} = -2m\omega_0 \times v \quad (3.3)$$

使用右手螺旋定则（即如果手指从 ω_0 旋转到 v ，拇指指向 $\omega_0 \times v$ 的方向），我们看到，在图中，科里奥利力使 m 向右偏转。

在低气压天气系统中，气团向低压中心移动。科里奥利力负责将气团向右偏转，从而引起称为气旋流的逆时针环流。结果是飓风的旋转运动。对图 3.2.2 的简要检查表明，在南半球， F_{cor} 使移动的物体向左偏转，因此低压系统将具有顺时针风向运动。

由于 $v = \omega \times r$ 和 $\omega \times (\omega \times r) = \omega(\omega \cdot r) - r(\omega \cdot \omega)$ ，我们可以写出

$$F_{cen} = -m\omega \times (\omega \times r) = m\omega^2 r \quad (3.4)$$

重要的是要注意，向心力涉及单个角速度的平方，而科里奥利力涉及两个不同角速度的乘积。

以线速度 v 移动的质量的动能为

$$K = \frac{1}{2}mv^2 \quad (3.5)$$

图 3.2.1 中质量的旋转动能为

$$K_{rot} = \frac{1}{2}I\omega^2 \quad (3.6)$$

其中转动惯量为

$$I = \int_V \rho(r)r^2 dr \quad (3.7)$$

$\rho(r)$ 是体积中半径 r 处的质量分布。在所示的简单情况下, m 是质点, 这变为

$$I = mr^2 \quad (3.8)$$

因此,

$$K_{rot} = \frac{1}{2}mr^2\omega^2 \quad (3.9)$$

质量为 m 的物体在重力加速度为常数 g 的引力场中高度 h 处的势能为

$$P = mgh \quad (3.10)$$

对应零势能的原点可以任意选择, 因为只有势能差在物理力方面才有意义。

以速度 v 移动的质量 m 的动量为

$$p = mv \quad (3.11)$$

质量 m 相对于原点的角动量为

$$P_{ang} = r \times p \quad (3.12)$$

力 F 相对于同一原点的扭矩或力矩定义为

$$N = r \times F \quad (3.13)$$

3.2.2 拉格朗日运动方程

对于保守系统, 拉格朗日运动方程由下式给出 [Marion 1965]

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \quad (3.14)$$

其中 q 是广义坐标 q_i 的 n 维向量, τ 是广义力 τ_i 的 n 维向量, 拉格朗日量是动能与势能之差

$$L = K - P \quad (3.15)$$

在我们的使用中, q 将是关节变量向量, 由关节角 θ_i (以度或弧度表示) 和关节偏移 d_i (以米为单位) 组成。那么 τ 是一个向量, 其分量为 n_i , 对应关节角的扭矩 (牛顿-米), 以及 f_i , 对应关节偏移的力 (牛顿)。注意我们用大写字母表示 τ 的分量。

我们将使用拉格朗日方程推导一般机器人机械臂动力学。让我们先通过考虑一些例子来感受一下这个过程。

例 3.2-1: 二连杆极坐标机械臂的动力学

二连杆平面旋转/平移 (RP) 机械臂的运动学在例 A.2-3 中给出。为了确定其动力学, 请检查图 3.2.3, 其中关节变量和关节速度向量为

$$q = \begin{bmatrix} \theta \\ r \end{bmatrix}, \quad \dot{q} = \begin{bmatrix} \dot{\theta} \\ \dot{r} \end{bmatrix} \quad (3.16)$$

相应的广义力向量为

$$\tau = \begin{bmatrix} n \\ f \end{bmatrix} \quad (3.17)$$

其中 n 为扭矩, f 为力。扭矩 n 和力 f 可由电机或液压驱动器提供。我们在 3.6 节讨论驱动器的动力学。

为了确定机械臂动力学, 我们现在必须计算拉格朗日方程所需的量。

a. 动能和势能

由于角运动 $\dot{\theta}$ 和线运动 \dot{r} 产生的总动能为

$$K = \frac{1}{2}m_1r^2\dot{\theta}^2 + \frac{1}{2}m_2(\dot{r}^2 + r^2\dot{\theta}^2) \quad (3.18)$$

势能为

$$P = m_1gr \sin \theta + m_2gr \sin \theta \quad (3.19)$$

b. 拉格朗日方程

拉格朗日量为

$$L = K - P = \frac{1}{2}m_1r^2\dot{\theta}^2 + \frac{1}{2}m_2(\dot{r}^2 + r^2\dot{\theta}^2) - m_1gr \sin \theta - m_2gr \sin \theta \quad (3.20)$$

现在我们得到

$$\frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} (m_1 + m_2)r^2\dot{\theta} \\ m_2\dot{r} \end{bmatrix} \quad (3.21)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} (m_1 + m_2)r^2\ddot{\theta} + 2(m_1 + m_2)r\dot{r}\dot{\theta} \\ m_2\ddot{r} \end{bmatrix} \quad (3.22)$$

$$\frac{\partial L}{\partial q} = \begin{bmatrix} -(m_1 + m_2)gr \cos \theta + (m_1 + m_2)r\dot{\theta}^2 \\ (m_1 + m_2)\dot{\theta}^2r - (m_1 + m_2)g \sin \theta \end{bmatrix} \quad (3.23)$$

因此, (3.2.14) 表明机械臂动力学方程为

$$(m_1 + m_2)r^2\ddot{\theta} + 2(m_1 + m_2)r\dot{r}\dot{\theta} + (m_1 + m_2)gr \cos \theta = n \quad (3.24)$$

$$m_2\ddot{r} - (m_1 + m_2)r\dot{\theta}^2 + (m_1 + m_2)g \sin \theta = f \quad (3.25)$$

这是一组耦合的非线性微分方程, 描述了给定控制输入扭矩 $n(t)$ 和力 $f(t)$ 时的运动 $q(t) = [\theta(t) \ r(t)]^T$ 。我们将在第四章展示如何通过计算机仿真确定给定控制输入 $n(t)$ 和 $f(t)$ 时的 $q(t)$ 。

根据我们对力和惯量的讨论，很容易识别动力学方程中的各项。每个方程中的第一项是涉及质量和惯量的加速度项。(9) 中的第二项是科里奥利项，而 (10) 中的第二项是向心项。第三项是重力项。

c. 机械臂动力学

通过使用向量，机械臂方程可以写成方便的形式。事实上，注意

$$\begin{bmatrix} (m_1 + m_2)r^2 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{r} \end{bmatrix} + \begin{bmatrix} 2(m_1 + m_2)r\dot{r}\dot{\theta} \\ -(m_1 + m_2)r\dot{\theta}^2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2)gr \cos \theta \\ (m_1 + m_2)g \sin \theta \end{bmatrix} = \begin{bmatrix} n \\ f \end{bmatrix} \quad (3.26)$$

我们将此向量方程符号化为

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (3.27)$$

注意，确实，惯量矩阵 $M(q)$ 是 q 的函数（即 θ 和 r 的函数），科里奥利/向心向量 $V(q, \dot{q})$ 是 q 和 \dot{q} 的函数，重力向量 $G(q)$ 是 q 的函数。

例 3.2-2: 二连杆平面肘形机械臂的动力学

在例 A.2-2 中给出了二连杆平面 RR 机械臂的运动学。为了确定其动力学，请检查图 3.2.4，其中我们假设连杆质量集中在连杆末端。关节变量为

$$q = [\theta_1 \ \theta_2]^T \quad (3.28)$$

广义力向量为

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (3.29)$$

其中 τ_1 和 τ_2 是由驱动器提供的扭矩。

a. 动能和势能

对于连杆 1，动能和势能为

$$K_1 = \frac{1}{2}m_1a_1^2\dot{\theta}_1^2 \quad (3.30)$$

$$P_1 = m_1ga_1 \sin \theta_1 \quad (3.31)$$

对于连杆 2，我们有

$$x_2 = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \quad (3.32)$$

$$y_2 = a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \quad (3.33)$$

$$\dot{x}_2 = -a_1 \sin \theta_1 \dot{\theta}_1 - a_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \quad (3.34)$$

$$\dot{y}_2 = a_1 \cos \theta_1 \dot{\theta}_1 + a_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \quad (3.35)$$

因此，速度平方为

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = a_1^2\dot{\theta}_1^2 + a_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + 2a_1a_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_1 \cos \theta_2 \quad (3.36)$$

因此，连杆 2 的动能为

$$K_2 = \frac{1}{2}m_2v_2^2 = \frac{1}{2}m_2a_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2a_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2a_1a_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_1 \cos \theta_2 \quad (3.37)$$

连杆 2 的势能为

$$P_2 = m_2gy_2 = m_2g[a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2)] \quad (3.38)$$

b. 拉格朗日方程

整个机械臂的拉格朗日量为

$$L = K - P = K_1 + K_2 - P_1 - P_2 \quad (3.39)$$

$$= \frac{1}{2}(m_1 + m_2)a_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2a_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (3.40)$$

$$+ m_2a_1a_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_1 \cos \theta_2 - (m_1 + m_2)ga_1 \sin \theta_1 \quad (3.41)$$

$$- m_2ga_2 \sin(\theta_1 + \theta_2) \quad (3.42)$$

(3.2.14) 所需的项为

$$\frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} (m_1 + m_2)a_1^2\dot{\theta}_1 + m_2a_2^2(\dot{\theta}_1 + \dot{\theta}_2) + m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \\ m_2a_2^2(\dot{\theta}_1 + \dot{\theta}_2) + m_2a_1a_2\dot{\theta}_1 \cos \theta_2 \end{bmatrix} \quad (3.43)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} (m_1 + m_2)a_1^2\ddot{\theta}_1 + m_2a_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2a_1a_2(2\ddot{\theta}_1 + \ddot{\theta}_2) \cos \theta_2 - m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \sin \theta_2 \\ m_2a_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2a_1a_2\ddot{\theta}_1 \cos \theta_2 - m_2a_1a_2\dot{\theta}_1\dot{\theta}_2 \sin \theta_2 \end{bmatrix} \quad (3.44)$$

$$\frac{\partial L}{\partial q} = \begin{bmatrix} -(m_1 + m_2)ga_1 \cos \theta_1 - m_2ga_2 \cos(\theta_1 + \theta_2) \\ -m_2a_1a_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_1 \sin \theta_2 - m_2ga_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.45)$$

最后，根据拉格朗日方程，机械臂动力学由两个耦合的非线性微分方程给出

$$[(m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2]\ddot{\theta}_1 + [m_2a_2^2 + m_2a_1a_2 \cos \theta_2]\ddot{\theta}_2 \quad (3.46)$$

$$- m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \sin \theta_2 + (m_1 + m_2)ga_1 \cos \theta_1 + m_2ga_2 \cos(\theta_1 + \theta_2) = \tau_1$$

$$[m_2a_2^2 + m_2a_1a_2 \cos \theta_2]\ddot{\theta}_1 + m_2a_2^2\ddot{\theta}_2 + m_2a_1a_2\dot{\theta}_1^2 \sin \theta_2 + m_2ga_2 \cos(\theta_1 + \theta_2) = \tau_2$$

c. 机械臂动力学

将机械臂动力学写成向量形式得到

$$\begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2 & m_2a_2^2 + m_2a_1a_2 \cos \theta_2 \\ m_2a_2^2 + m_2a_1a_2 \cos \theta_2 & m_2a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (3.47)$$

$$+ \begin{bmatrix} -m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \sin \theta_2 \\ m_2a_1a_2\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2)ga_1 \cos \theta_1 + m_2ga_2 \cos(\theta_1 + \theta_2) \\ m_2ga_2 \cos(\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (3.48)$$

其中

$$M(q) = \begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2 & m_2a_2^2 + m_2a_1a_2 \cos \theta_2 \\ m_2a_2^2 + m_2a_1a_2 \cos \theta_2 & m_2a_2^2 \end{bmatrix} \quad (3.49)$$

这些机械臂动力学采用标准形式

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (3.50)$$

其中 $M(q)$ 为惯量矩阵, $V(q, \dot{q})$ 为科里奥利/向心向量, $G(q)$ 为重力向量。注意 $M(q)$ 是对称的。

习题 3.2-3: 三连杆圆柱形机械臂的动力学

我们在例 A.2-1 中研究三连杆圆柱形机械臂的运动学。在图 3.2.5 中, 关节变量向量为

$$q = [\theta \ h \ r]^T \quad (3.51)$$

证明机械臂动力学由下式给出

$$\begin{bmatrix} J + m_1r^2 + m_2r^2 & 0 & 0 \\ 0 & m_1 + m_2 & 0 \\ 0 & 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{h} \\ \ddot{r} \end{bmatrix} + \begin{bmatrix} 2(m_1 + m_2)r\dot{r}\dot{\theta} \\ 0 \\ -(m_1 + m_2)r\dot{\theta}^2 \end{bmatrix} + \begin{bmatrix} 0 \\ (m_1 + m_2)g \\ 0 \end{bmatrix} = \begin{bmatrix} n \\ f_h \\ f_r \end{bmatrix} \quad (3.52)$$

其中 J 是基座连杆的惯量, 力向量为

$$\tau = \begin{bmatrix} n \\ f_h \\ f_r \end{bmatrix} \quad (3.53)$$

3.2.3 推导操作器动力学

我们在几个例子中展示了如何应用拉格朗日方程计算任何给定机器人机械臂的动力学方程。在例子中, 我们发现的动力学总是具有特殊形式

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (3.54)$$

其中 q 为关节变量向量, τ 为广义力/扭矩向量。在本小节中, 我们推导一般机器人机械臂的动力学。它们将采用这种相同的形式。

为了获得一般机器人机械臂动力学方程, 我们确定机械臂动能和势能, 然后拉格朗日量, 最后代入拉格朗日方程 (3.2.14) 获得最终结果 [Paul 1981, Lee et al. 1983, Asada and Slotine 1986, Spong and Vidyasagar 1989]。

机械臂动能

给定连杆 i 上以相对于连接到该连杆的坐标系 i 的坐标 ${}^i r$ 表示的一个点, 该点在基坐标系中的坐标为

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = T_i {}^i r \quad (3.55)$$

其中 T_i 是附录 A 中定义的 4×4 齐次变换矩阵。注意 T_i 是关节变量 q_1, q_2, \dots, q_i 的函数。因此, 该点在基坐标系中的速度为

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 0 \end{bmatrix} = \sum_{j=1}^i \frac{\partial T_i}{\partial q_j} \dot{q}_j {}^i r \quad (3.56)$$

由于当 $j > i$ 时, $\partial T_i / \partial q_j = 0$, 我们可以将上限求和替换为 n , 即连杆数。如果已知机械臂矩阵 T_i , 则可以计算 4×4 矩阵 $\partial T_i / \partial q_j$ 。

以速度 $v = [v_x \ v_y \ v_z]^T$ 运动的无穷小质量 dm 的动能为

$$dK_i = \frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)dm = \frac{1}{2}\text{trace}(\dot{T}_i {}^i r {}^i r^T \dot{T}_i^T)dm \quad (3.57)$$

因此, 连杆 i 的总动能为

$$K_i = \int_{\text{link } i} dK_i = \frac{1}{2}\text{trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \dot{q}_j \left(\int_{\text{link } i} {}^i r {}^i r^T dm \right) \frac{\partial T_i^T}{\partial q_k} \dot{q}_k \right] \quad (3.58)$$

将 dK_i 的表达式代入, 我们可以将积分移到求和内。然后, 为连杆 i 定义 4×4 伪惯量矩阵为

$$M_i = \int_{\text{link } i} {}^i r {}^i r^T dm \quad (3.59)$$

我们可以将连杆 i 的动能写为

$$K_i = \frac{1}{2} \sum_{j=1}^i \sum_{k=1}^i \text{trace} \left(\frac{\partial T_i}{\partial q_j} M_i \frac{\partial T_i^T}{\partial q_k} \right) \dot{q}_j \dot{q}_k = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (3.60)$$

让我们继续寻找机械臂总动能之前简要讨论伪惯量矩阵。令

$${}^i r = [x \ y \ z \ 1]^T \quad (3.61)$$

为无穷小质量 dm 在坐标系 i 中的坐标。那么, 展开 (3.2.20) 得到

$$M_i = \begin{bmatrix} \int x^2 dm & \int xy dm & \int xz dm & \int x dm \\ \int xy dm & \int y^2 dm & \int yz dm & \int y dm \\ \int xz dm & \int yz dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix} \quad (3.62)$$

其中积分在连杆 i 的体积上进行。这是一个对每个连杆计算一次的常数矩阵。它取决于连杆 i 的几何形状和质量分布。实际上，根据连杆 i 的惯性矩

$$I_{xx} = \int (y^2 + z^2) dm \quad (3.63)$$

$$I_{yy} = \int (x^2 + z^2) dm \quad (3.64)$$

$$I_{zz} = \int (x^2 + y^2) dm \quad (3.65)$$

惯性积

$$I_{xy} = \int xy dm \quad (3.66)$$

$$I_{xz} = \int xz dm \quad (3.67)$$

$$I_{yz} = \int yz dm \quad (3.68)$$

和一阶矩

$$mx = \int x dm = mx_c \quad (3.69)$$

$$my = \int y dm = my_c \quad (3.70)$$

$$mz = \int z dm = mz_c \quad (3.71)$$

其中 m 为连杆 i 的总质量，

$$x_c = \frac{1}{m} \int x dm \quad (3.72)$$

是连杆 i 质心在坐标系 i 中的坐标，我们可以写出

$$M_i = \begin{bmatrix} \frac{1}{2}(-I_{xx} + I_{yy} + I_{zz}) & I_{xy} & I_{xz} & mx_c \\ I_{xy} & \frac{1}{2}(I_{xx} - I_{yy} + I_{zz}) & I_{yz} & my_c \\ I_{xz} & I_{yz} & \frac{1}{2}(I_{xx} + I_{yy} - I_{zz}) & mz_c \\ mx_c & my_c & mz_c & m \end{bmatrix} \quad (3.73)$$

这些量要么在机械臂制造商的规格表中列出，要么可以从那里列出的量计算得出。回到我们的推导，机械臂总动能可以写为

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{trace} \left(\frac{\partial T_i}{\partial q_j} M_i \frac{\partial T_i^T}{\partial q_k} \right) \dot{q}_j \dot{q}_k \quad (3.74)$$

由于矩阵和的迹是各个迹的和，我们可以交换求和和迹算子得到

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n m_{jk} \dot{q}_j \dot{q}_k = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (3.75)$$

或

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (3.76)$$

其中 $n \times n$ 机械臂惯量矩阵 $M(q)$ 的元素定义为

$$m_{jk} = \sum_{i=\max(j,k)}^n \text{trace} \left(\frac{\partial T_i}{\partial q_j} M_i \frac{\partial T_i^T}{\partial q_k} \right) \quad (3.77)$$

由于当 $j > i$ 时, $\partial T_i / \partial q_j = 0$, 我们可以更有效地将其写为

$$m_{jk} = \sum_{i=\max(j,k)}^n \text{trace} \left(\frac{\partial T_i}{\partial q_j} M_i \frac{\partial T_i^T}{\partial q_k} \right) \quad (3.78)$$

方程 (3.2.29) 正是我们一直在寻求的; 它提供了以已知量和关节变量 q 表示的机械臂动能的方便表达式。由于 $m_{jk} = m_{kj}$, 惯量矩阵 $M(q)$ 是对称的。由于动能是正的, 只有当广义速度为零时才消失, 惯量矩阵 $M(q)$ 也是正定的。注意动能取决于 q 和 \dot{q} 。

机械臂势能

如果连杆 i 具有质量 m_i 和以其坐标系 i 的坐标表示的质心, 该连杆的势能为

$$P_i = -m_i g^T T_i^i r_i \quad (3.79)$$

其中重力向量以基坐标表示为

$$g = \begin{bmatrix} g_x \\ g_y \\ g_z \\ 0 \end{bmatrix} \quad (3.80)$$

如果机械臂是水平的, 在海平面上, 基座 z 轴垂直向上, 那么

$$g = \begin{bmatrix} 0 \\ 0 \\ -9.8 \\ 0 \end{bmatrix} \text{ m/s}^2 \quad (3.81)$$

因此, 机械臂总势能为

$$P = \sum_{i=1}^n P_i = - \sum_{i=1}^n m_i g^T T_i^i r_i \quad (3.82)$$

注意 P 仅取决于关节变量 q , 而不取决于关节速度 \dot{q} 。

注意 $-m_i g^T T_i^i$ 是连杆 i 伪惯量矩阵 M_i 的最后一列, 我们可以写出

$$P = - \sum_{i=1}^n m_i g^T T_i^i r_i \quad (3.83)$$

其中 e_4 是 4×4 单位矩阵的最后一列 (即 $e_4 = [0 \ 0 \ 0 \ 1]^T$)。

拉格朗日方程

机械臂拉格朗日量为

$$L = K - P = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q) \quad (3.84)$$

动能是关节速度向量的二次函数, 势能独立于 \dot{q} 是一个基本性质。

拉格朗日方程 (3.2.14) 现在所需的项由下式给出

$$\frac{\partial L}{\partial \dot{q}} = M(q) \dot{q} \quad (3.85)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = M(q) \ddot{q} + \dot{M}(q) \dot{q} \quad (3.86)$$

$$\frac{\partial L}{\partial q} = \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M(q) \dot{q}) - \frac{\partial P(q)}{\partial q} \quad (3.87)$$

因此, 机械臂动力学方程为

$$M(q) \ddot{q} + \dot{M}(q) \dot{q} - \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M(q) \dot{q}) + \frac{\partial P(q)}{\partial q} = \tau \quad (3.88)$$

定义科里奥利/向心向量

$$V(q, \dot{q}) = \dot{M}(q) \dot{q} - \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M(q) \dot{q}) \quad (3.89)$$

和重力向量

$$G(q) = \frac{\partial P(q)}{\partial q} \quad (3.90)$$

我们可以写出

$$M(q) \ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (3.91)$$

这正是我们一直在寻求的机器人动力学方程的最终形式。

$M(q)$ 元素相对于旋转关节变量 $q_i = \theta_i$ 的单位是 $\text{kg} \cdot \text{m}^2$ 。 $M(q)$ 元素相对于平移关节变量 $q_i = d_i$ 的单位是千克。 $V(q, \dot{q})$ 和 $G(q)$ 元素相对于旋转关节变量的单位是 $\text{kg} \cdot \text{m}^2 / \text{s}^2$ 。 $V(q, \dot{q})$ 和 $G(q)$ 元素相对于平移关节变量的单位是 $\text{kg} \cdot \text{m} / \text{s}^2$ 。

3.3 机器人方程的结构与性质

在本节中, 我们研究动力学机械臂方程的详细结构和性质, 因为这种结构应该反映在控制律的形式中。如果在设计阶段结合机械臂的已知性质, 控制器会更简单更有效。

实际上, 机器人机械臂总是受摩擦和干扰的影响。因此, 我们将我们刚刚推导出的机械臂模型推广, 将机械臂动力学写为

$$M(q) \ddot{q} + V(q, \dot{q}) + F_v \dot{q} + F_d(\dot{q}) + G(q) + \tau_d = \tau \quad (3.92)$$

其中 q 为关节变量 n 维向量, τ 为广义力的 n 维向量。 $M(q)$ 是惯量矩阵, $V(q, \dot{q})$ 是科里奥利/向心向量, $G(q)$ 是重力向量。我们添加了一个摩擦项

$$F(\dot{q}) = F_v \dot{q} + F_d(\dot{q}) \quad (3.93)$$

其中 F_v 是粘性摩擦系数矩阵, F_d 是动态摩擦项。还添加了一个干扰 τ_d , 它可以表示, 例如, 任何建模不准确的动力学。

摩擦不是一个容易建模的项, 实际上, 可能是机械臂动力学模型中最难描述的项。关于摩擦的更多讨论可以在 [Schilling 1990] 中找到。

我们有时将机械臂动力学写为

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (3.94)$$

其中

$$N(q, \dot{q}) = V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d \quad (3.95)$$

表示非线性项。

让我们检查机器人动力学方程中每一项的结构和性质。这项研究将为我们提供很大的洞察力, 我们在后续章节中推导机器人控制方案时会用到。我们在表 3.3.1 中总结了发现的性质。当我们发展每个性质时, 参考例 3.2.1 到 3.2.3 以验证这些性质确实在那里成立是值得的。在本节末尾的例 3.3.1 中, 我们说明了二连杆平面肘形机械臂的几个性质。

表 3.1: 机器人方程及其性质

机器人方程	
动力学	$M(q)\ddot{q} + V(q, \dot{q}) + F_v \dot{q} + F_d(\dot{q}) + G(q) + \tau_d = \tau$
状态空间	$\dot{x} = Ax + Bu$, 其中 $u = \tau$, $x = [q^T \dot{q}^T]^T$
性质	
惯量矩阵	对称、正定的有界矩阵
科里奥利/向心	包含 \dot{q} 的二次项
摩擦	$F(\dot{q}) = F_v \dot{q} + F_d(\dot{q})$
干扰	$\ \tau_d\ \leq d$
线性参数化	$M(q)\ddot{q} + V(q, \dot{q}) + F_v \dot{q} + F_d(\dot{q}) + G(q) = W(q, \dot{q}, \ddot{q})\phi$
斜对称	$S(q, \dot{q}) = \dot{M}(q) - 2V_m(q, \dot{q})$ 是斜对称的
无源性	$\int_0^T \dot{q}^T \tau dt \geq -\gamma^2$
能量守恒	$\dot{K} = \dot{q}^T (\tau - G)$

3.3.1 惯量矩阵性质

正如我们所见, $M(q)$ 是对称且正定的。实际上, 机械臂动能为

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (3.96)$$

下一小节给出了 M 的一些表达式。

$M(q)$ 的另一个重要性质是它上下有界。即,

$$\mu_1 I \leq M(q) \leq \mu_2 I \quad (3.97)$$

其中 μ_1 和 μ_2 是可以为任何给定机械臂计算的标量 (见例 3.3.1)。当我们说 $\mu_1 I \leq M(q)$, 例如, 我们的意思是 $(M(q) - \mu_1 I)$ 是半正定的。即,

$$x^T (M - \mu_1 I) x \geq 0 \quad (3.98)$$

对所有 $x \in R^n$ 成立。

同样, 惯量矩阵的逆是有界的, 因为

$$\frac{1}{\mu_2} I \leq M^{-1}(q) \leq \frac{1}{\mu_1} I \quad (3.99)$$

如果机械臂是旋转关节的, 界限 μ_1 和 μ_2 是常数, 因为 q 只通过 \sin 和 \cos 项出现在 $M(q)$ 中, 它们的大小被 1 限制 (见例 3.2.2 和 3.3.1)。另一方面, 如果机械臂有平移关节, 那么 μ_1 和 μ_2 可能是 q 的标量函数。见例 3.2.1, 其中 $M(q)$ 被 $\mu_2 = mr^2$ (如果 $r > 1$) 上界限制。

惯量矩阵的有界性也可以表示为

$$m_1 \leq \|M(q)\| \leq m_2 \quad (3.100)$$

其中任何诱导矩阵范数可以用来定义正标量 m_1 和 m_2 。

3.3.2 科里奥利/向心项性质

一瞥 (3.2.42) 就会发现一个问题, 如果不理解, 会使机器人动力学的研究变得混乱。这个 $V(q, \dot{q})$ 项的简化将需要对矩阵 [即 $M(q)$] 关于 n 维向量 q 求导。然而, 这样的导数不是矩阵, 而是三阶张量——即, 它们必须用三个指标表示, 而不是两个。有几种方法可以解决这个问题, 涉及几个新量的定义。

$V(q, \dot{q})$ 的克罗内克积分析

让我们首先从克罗内克积 [Brewer 1978] 的角度检查项 $V(q, \dot{q})$, 克罗内克积对两个矩阵 A 、 B 定义为

$$A \otimes B = [a_{ij} B] \quad (3.101)$$

其中 A 有元素 a_{ij} , $[a_{ij}B]$ 表示由 $p \times q$ 块 $a_{ij}B$ 组成的 $np \times mq$ 块矩阵。因此, 对于 A 、 B 、 C 、 D , 我们有

$$(A \otimes B)(C \otimes D) = AC \otimes BD \quad (3.102)$$

对于矩阵 $A(q)$ 、 $B(q)$, 定义矩阵导数为

$$\frac{\partial A}{\partial q} = \begin{bmatrix} \frac{\partial A}{\partial q_1} & \cdots & \frac{\partial A}{\partial q_n} \end{bmatrix} \quad (3.103)$$

然后我们可以证明乘积法则

$$\frac{\partial}{\partial q}(AB) = \frac{\partial A}{\partial q}(I_n \otimes B) + A \frac{\partial B}{\partial q} \quad (3.104)$$

其中 I_n 是 $n \times n$ 单位矩阵。

现在我们可以检查科里奥利/向心向量 $V(q, \dot{q})$ [Koditschek 1984, Gu and Loh 1988]。在 (3.2.42) 上使用 (3.3.11) 两次, 我们可以获得

$$V(q, \dot{q}) = \frac{dM}{dt} \dot{q} - \frac{1}{2}(I_n \otimes \dot{q}^T) \frac{\partial M}{\partial q} \dot{q} \quad (3.105)$$

或

$$V(q, \dot{q}) = [\dot{M}(q) - \frac{1}{2}U(q)] \dot{q} \quad (3.106)$$

其中

$$U(q) = \begin{bmatrix} \frac{\partial M}{\partial q_1} & \cdots & \frac{\partial M}{\partial q_n} \end{bmatrix} \quad (3.107)$$

$$\dot{M} = \sum_{j=1}^n \frac{\partial M}{\partial q_j} \dot{q}_j \quad (3.108)$$

矩阵系数由下式给出

$$\frac{\partial M}{\partial q_j} = \begin{bmatrix} \frac{\partial m_{1j}}{\partial q_1} & \cdots & \frac{\partial m_{1j}}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial m_{nj}}{\partial q_1} & \cdots & \frac{\partial m_{nj}}{\partial q_n} \end{bmatrix} \quad (3.109)$$

为了找到 V_{m1} 的等效表达式, 注意

$$\frac{\partial}{\partial q}(\dot{q}^T M \dot{q}) = \frac{\partial}{\partial q}(\dot{q}^T M) \dot{q} + (I_n \otimes \dot{q}^T) \frac{\partial M^T}{\partial q} \dot{q} \quad (3.110)$$

这可以写为

$$\frac{\partial}{\partial q}(\dot{q}^T M \dot{q}) = \left[\frac{\partial}{\partial q}(\dot{q}^T M) + (I_n \otimes \dot{q}^T) \frac{\partial M}{\partial q} \right] \dot{q} \quad (3.111)$$

或

$$\frac{\partial}{\partial q}(\dot{q}^T M \dot{q}) = \left[\dot{q}^T \frac{\partial M}{\partial q} + (I_n \otimes \dot{q}^T) \frac{\partial M}{\partial q} \right] \dot{q} \quad (3.112)$$

(注意 $\partial M/\partial q_i$ 是对称的。) 因此,

$$V_{m1}(q, \dot{q}) = \frac{1}{2} \left[\dot{M}(q) + (I_n \otimes \dot{q}^T) \frac{\partial M}{\partial q} \right] \dot{q} \quad (3.113)$$

当使用适当的定义时, 我们可以写出

$$V_{m1}(q, \dot{q}) = \frac{1}{2} [\dot{M}(q) + U^T(q)] \dot{q} \quad (3.114)$$

也可以写出 (见习题)

$$V_{m2}(q, \dot{q}) = \frac{1}{2} [\dot{M}(q) + U(q)] \dot{q} \quad (3.115)$$

由于 $V_v(\dot{q})$ 在 \dot{q} 中是线性的, 因此 $V(q, \dot{q})$ 在 \dot{q} 中是二次的。事实上, 可以证明 (见习题)

$$V(q, \dot{q}) = \begin{bmatrix} \dot{q}^T V_1(q) \dot{q} \\ \vdots \\ \dot{q}^T V_n(q) \dot{q} \end{bmatrix} \quad (3.116)$$

对于 $V_i(q)$ 的适当定义 [Craig 1988]。确实, $V_i(q)$ 是对称的 $n \times n$ 矩阵。

由于 $V(q, \dot{q})$ 在 \dot{q} 中是二次的, 它可以被 \dot{q} 的二次函数上界限制。即,

$$\|V(q, \dot{q})\| \leq v_b(q) \|\dot{q}\|^2 \quad (3.117)$$

其中 $v_b(q)$ 是已知的标量函数, $\|\cdot\|$ 是任何适当的范数。对于旋转机械臂, v_b 是与 q 无关的常数。见例 3.2.2 和 3.3.1, 其中 \dot{q} 中的二次项乘以 $\sin \theta_2$, 其大小被 1 限制。另一方面, 对于具有平移关节的机械臂, $v_b(q)$ 可能是 q 的函数; 见例 3.2.1 和 3.2.3, 其中 $V(q, \dot{q})$ 有一个在 \dot{q} 中的二次项乘以 r 。

为了帮助确定给定机器人机械臂的 $v_b(q)$, 注意 $V(q, \dot{q}) = V_v(\dot{q})\dot{q}$, 因此

$$\|V(q, \dot{q})\| \leq \|V_v(q)\| \cdot \|\dot{q}\| \quad (3.118)$$

其中 $\|V_v(q)\|$ 在 (3.3.23) 中定义。因此, 对于旋转机械臂

$$v_b = \sup_q \|V_v(q)\| \quad (3.119)$$

我们可以注意到 $\dot{q}^2 \equiv \dot{q} \otimes \dot{q}$, 因此

$$\dot{q}^2 = \begin{bmatrix} \dot{q}_1^2 & \dot{q}_1 \dot{q}_2 & \cdots & \dot{q}_1 \dot{q}_n & \dot{q}_2 \dot{q}_1 & \cdots & \dot{q}_n^2 \end{bmatrix}^T \quad (3.120)$$

这是一个由 \dot{q} 分量所有可能乘积组成的 n^2 维向量。这允许我们证明

$$\|\dot{q}\|^2 = \|\dot{q}^2\| \quad (3.121)$$

在这个证明中，我们还需要恒等式

$$(A \otimes B)^T = A^T \otimes B^T \quad (3.122)$$

对任何矩阵 A 和 B 成立。注意： $(A \otimes B)^T = A^T \otimes B^T$ 不成立。

现在我们可以使用这些各种恒等式来证明

$$V_m(q, \dot{q}) = V_{m1}(q) + V_{v1}(q)(I_n \otimes \dot{q}) = V_{m2}(q) - V_{v2}(q)(I_n \otimes \dot{q}) \quad (3.123)$$

因此

$$V(q, \dot{q}) = V_m(q)\dot{q} = V_{m1}(q, \dot{q})\dot{q} = V_{m2}(q, \dot{q})\dot{q} \quad (3.124)$$

其中

$$V_{m1}(q) = \frac{1}{2}[\dot{M}(q) + U^T(q)] \quad (3.125)$$

$$V_{m2}(q) = \frac{1}{2}[\dot{M}(q) + U(q)] \quad (3.126)$$

注意，一般来说， $V_{m1} \neq V_{m2}$ 。

就 M 和 U 而言，机械臂动力学可以写为

$$M(q)\ddot{q} + V_m(q)\dot{q} + G(q) = \tau \quad (3.127)$$

在这一点上，我们可以证明一个恒等式，它在构建高级控制方案时非常有用。我们称之为斜对称性质；它表明 $M(q)$ 的导数和科里奥利向量以非常特殊的方式相关。事实上，

$$\dot{M} - 2V_m = -(\dot{M} - 2V_m)^T \quad (3.128)$$

由于矩阵减去其转置总是斜对称的，这是一个重要的恒等式。如果使用 V_{m1} 代替 V_{m2} ，这也成立。

重要的是要注意，(3.3.33) 中的第一个等式成立是因为 \dot{q} 乘以 $\dot{M} - 2V_m$ 。即， $\dot{q}^T(\dot{M} - 2V_m)\dot{q} = 0$ ，因此 $(\dot{M} - 2V_m)$ 本身不一定是斜对称的。然而，可以定义一个矩阵 V_m ，使得

$$V(q, \dot{q}) = V_m(q, \dot{q})\dot{q} \quad (3.129)$$

和

$$S(q, \dot{q}) \equiv \dot{M}(q) - 2V_m(q, \dot{q}) \quad (3.130)$$

是斜对称的，因此对所有 $x \in R^n$ ， $x^T S x = 0$ 。确实，根据 (3.3.13) 和 (3.3.27)，我们可以定义

$$V_m(q, \dot{q}) = \frac{1}{2}[\dot{M}(q) - U^T(q)] \quad (3.131)$$

因为那么斜对称矩阵只不过是

$$S = \dot{M} - 2V_m = U - U^T \quad (3.132)$$

这个 V_m 是在几个现代自适应和鲁棒控制算法中使用的标准定义，也是我们在本书其余部分要使用的定义。因此，我们将机械臂方程写为

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) = \tau \quad (3.133)$$

或

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (3.134)$$

注意，可以将 $V(q, \dot{q})$ 分成其科里奥利和向心分量为

$$V(q, \dot{q}) = V_{cor}(q) + V_{cen}(q, \dot{q}) \quad (3.135)$$

其中

$$V_{cor}(q) = \dot{M}(q)\dot{q} - V_{cen}(q, \dot{q}) \quad (3.136)$$

和 V_{cen} 是 (3.3.26) 中所有平方项 \dot{q}_i^2 被移除的 \dot{q}^2 [Craig 1988] (见习题)。

$V(q, \dot{q})$ 的分量分析

科里奥利/向心向量的克罗内克积分析的替代方法是基于 $V(q, \dot{q})$ 标量分量的分析，这提供了额外的洞察力。

就惯量矩阵 $M(q)$ 的分量 $m_{kj}(q)$ 而言，我们可以将 (3.2.38)-(3.2.40) 分量写为

$$\frac{\partial K}{\partial \dot{q}_k} = \sum_j m_{kj} \dot{q}_j \quad (3.137)$$

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{q}_k} = \sum_j m_{kj} \ddot{q}_j + \sum_{i,j} \frac{\partial m_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j \quad (3.138)$$

$$\frac{\partial K}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial m_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j \quad (3.139)$$

其中所有求和都是在关节数 n 上进行的。现在，拉格朗日方程表明机械臂动力学分量表示为

$$\sum_j m_{kj} \ddot{q}_j + \sum_{i,j} \left[\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial m_{ij}}{\partial q_k} \right] \dot{q}_i \dot{q}_j + \frac{\partial P}{\partial q_k} = \tau_k \quad (3.140)$$

其中 n 是关节数。

通过交换求和顺序并利用对称性，

$$\sum_{i,j} \frac{\partial m_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j = \frac{1}{2} \sum_{i,j} \left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} \right] \dot{q}_i \dot{q}_j \quad (3.141)$$

因此，我们可以定义

$$v_{ijk} = \frac{1}{2} \left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right] \quad (3.142)$$

并将机械臂动力学写为

$$\sum_j m_{kj} \ddot{q}_j + \sum_{i,j} v_{ijk} \dot{q}_i \dot{q}_j + \frac{\partial P}{\partial q_k} = \tau_k \quad (3.143)$$

v_{ijk} 的循环对称性使我们能够推导出这个方程中对应于科里奥利/向心向量 $V(q, \dot{q})$ 的第二项的重要性质。量 v_{ijk} 被称为(第一类)克里斯托费尔符号 [Borisenko and Tarapov 1968]。

(3.3.36) 中定义的矩阵 V_m 的分量 v_{kj} 由下式给出

$$v_{kj} = \sum_i v_{ijk} \dot{q}_i \quad (3.144)$$

3.3.3 重力、摩擦与干扰性质

重力项 $G(q)$ 的性质

根据 (3.2.43) 和 (3.2.36),

$$G(q) = \frac{\partial P}{\partial q} = - \sum_{i=1}^n m_i g^T \frac{\partial T_{i i} r_i}{\partial q} \quad (3.145)$$

因此使用 (3.3.11) 两次揭示

$$G(q) = - \sum_{i=1}^n m_i (I_n \otimes g^T) \frac{\partial T_{i i} r_i}{\partial q} \quad (3.146)$$

可以为任何给定机器人机械臂导出重力项的界限。因此

$$\|G(q)\| \leq g_b(q) \quad (3.147)$$

其中 $\|\cdot\|$ 是任何适当的向量范数, g_b 是可以为任何给定机械臂确定的标量函数 (见例 3.3.1)。对于旋转机械臂, g_b 是与关节向量 q 无关的常数, 但对于具有平移连杆的机械臂, g_b 可能取决于 q 。见 3.2 节和例 3.3.1 中的例子来验证这些说法。

摩擦项 $F(\dot{q})$ 的性质

机械臂方程 (3.3.1) 中的摩擦形式为

$$F(\dot{q}) = F_v \dot{q} + F_d(\dot{q}) \quad (3.148)$$

其中 F_v 是粘性摩擦系数矩阵, F_d 是动态摩擦项。摩擦系数是为给定机械臂最难确定的参数之一, 实际上, (3.3.52) 仅代表其影响的近似数学模型。更多讨论见 [Craig 1988, Schilling 1990]。

由于摩擦是局部效应, 我们可以假设 $F(\dot{q})$ 在关节之间是解耦的, 因此

$$F(\dot{q}) = \begin{bmatrix} f_1(\dot{q}_1) \\ \vdots \\ f_n(\dot{q}_n) \end{bmatrix} \quad (3.149)$$

其中 $f_i(\cdot)$ 是可以为任何给定机械臂确定的已知标量函数。我们为将来使用定义了 $\text{vec}\{\cdot\}$ 函数。

粘性摩擦通常可以假设具有形式

$$F_v \dot{q} = \text{diag}\{v_i\} \dot{q} \quad (3.150)$$

其中 v_i 是已知常数系数。那么 $F_v = \text{diag}\{v_i\}$ ，一个对角元素为 v_i 的对角矩阵。动态摩擦通常可以假设具有形式

$$F_d(\dot{q}) = \text{diag}\{k_i\} \text{sgn}(\dot{q}) \quad (3.151)$$

其中 k_i 是已知常数系数，符号函数对标量 x 定义为

$$\text{sgn}(x) = \begin{cases} +1, & x > 0 \\ -1, & x < 0 \\ 0, & x = 0 \end{cases} \quad (3.152)$$

那么

$$F_d(\dot{q}) = K_d \text{sgn}(\dot{q}) \quad (3.153)$$

其中 $K_d = \text{diag}\{k_i\}$ 是动态摩擦系数矩阵，向量 x 的符号函数定义为

$$\text{sgn}(x) = \text{vec}\{\text{sgn}(x_i)\} = \begin{bmatrix} \text{sgn}(x_1) \\ \vdots \\ \text{sgn}(x_n) \end{bmatrix} \quad (3.154)$$

摩擦项的界限可以假设具有形式

$$\|F_v \dot{q}\| \leq v \|\dot{q}\|, \quad \|F_d(\dot{q})\| \leq k \quad (3.155)$$

其中 v 和 k 对特定机械臂是已知的， $\|\cdot\|$ 是合适的范数。

可以包含在 $F(\dot{q})$ 中的另一个摩擦项是静摩擦，其分量具有形式

$$F_{si} = k_{si} \text{sgn}(\dot{q}_i), \quad |\dot{q}_i| < \epsilon \quad (3.156)$$

其中 k_{si} 是关节 i 的静摩擦系数， ϵ 是一个小的正参数。我们通常忽略这个项。

干扰项 τ_d 的性质

机械臂方程 (3.3.1) 有一个干扰项 τ_d ，它可以表示建模不准确的动力学等。我们假设它是有界的，因此

$$\|\tau_d\| \leq d \quad (3.157)$$

其中 d 是可以为给定机械臂计算的标量常数， $\|\cdot\|$ 是任何合适的范数。

3.3.4 参数线性

机器人动力学方程享有的最后一个性质将在第五章中对我们非常有用。即，它在参数中是线性的，这个性质首先在 [Craig 1988] 的自适应控制中被利用。这很重要，因为一些或所有参数可能是未知的；因此动力学在未知项中是线性的。

这个性质可以表示为

$$M(q)\ddot{q} + V(q, \dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) = W(q, \dot{q}, \ddot{q})\phi \quad (3.158)$$

其中 ϕ 是参数向量， $W(q, \dot{q}, \ddot{q})$ 是取决于关节变量、关节速度和关节加速度的机器人函数矩阵。这个矩阵可以为任何给定机器人机械臂计算，因此是已知的。见例 3.3.1。注意干扰 τ_d 不包含在这个方程中。

例 3.3-1: 二连杆平面肘形机械臂的结构和界限

二连杆平面机械臂的动力学在例 3.2.2 中给出。我们现在应该计算本节中定义的结构矩阵，以及表 3.3.1 中需要的界限。摩擦界限是直接的，因此我们在这里不提及它们。动力学矩阵为

$$M(q) = \begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2 & m_2a_2^2 + m_2a_1a_2 \cos \theta_2 \\ m_2a_2^2 + m_2a_1a_2 \cos \theta_2 & m_2a_2^2 \end{bmatrix} \quad (3.159)$$

$$V(q, \dot{q}) = \begin{bmatrix} -m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \sin \theta_2 \\ m_2a_1a_2\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} \quad (3.160)$$

$$G(q) = \begin{bmatrix} (m_1 + m_2)ga_1 \cos \theta_1 + m_2ga_2 \cos(\theta_1 + \theta_2) \\ m_2ga_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.161)$$

在表 3.3.1 中选择合适的范数并不总是简单的。在后续章节中要开发的控制算法中，我们证明了就某种范数而言的适当性能，这通常可以是任何期望的范数。为了实现控制器，必须选择特定的范数并评估界限。这个选择通常简单地取决于哪种范数使得可以评估表中的界限。例如，选择向量的 2-范数需要评估 $M(q)$ 的最大奇异值，这是一个非常困难的任务。

选择向量的 ∞ -范数意味着在每个采样时间确定 $V(q(t), \dot{q}(t))$ 中具有最大幅度的元素。这需要决策逻辑，范数可能不是连续的。因此，让我们在这个例子中使用 1-范数。相应的矩阵诱导范数则是最大绝对列和（第 2 章）。

a. 惯量矩阵的界限

μ_1 和 μ_2 的评估相当于确定 $M(q)$ 在所有 q 上的最小和最大特征值。这不是一件容易的事，需要求解一些二次方程，尽管使用 Mathematica 或 Maple 等软件可以相当容易地进行。因此，让我们找到 m_1 和 m_2 。

$M(q)$ 的诱导 1-范数是最大绝对列和。在确定这个范数的界限时，重要的是要考虑关节角允许的运动范围。为了说明，假设 θ_1 和 θ_2 被限制在 $\pm\pi/2$ 以内。那么 1-范数总

是以第 1 列表示为

$$\|M(q)\|_1 = |(m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2| + |m_2a_2^2 + m_2a_1a_2 \cos \theta_2| \quad (3.162)$$

这被下式上界限制

$$m_2 = (m_1 + m_2)a_1^2 + 2m_2a_2^2 + 3m_2a_1a_2 \quad (3.163)$$

和下界

$$m_1 = (m_1 + m_2)a_1^2 + m_2a_2^2 - 2m_2a_1a_2 \quad (3.164)$$

由于机械臂是旋转的, 且 $\cos \theta_2$ 上下有界, M_2 和 M_1 是常数。重要的是要注意, 如果机械臂是旋转/平移 (RP) 的, 因此关节变量为 (θ_1, a_2) , 界限是 q 的函数。

b. 科里奥利和重力项的界限

使用

$$V(q, \dot{q}) = \begin{bmatrix} -m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \sin \theta_2 \\ m_2a_1a_2\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} \quad (3.165)$$

找到科里奥利/向心向量的界限 v_b , 因此 $v_b = m_2a_1a_2$ 。

同样, 对于重力界限,

$$G(q) = \begin{bmatrix} (m_1 + m_2)ga_1 \cos \theta_1 + m_2ga_2 \cos(\theta_1 + \theta_2) \\ m_2ga_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.166)$$

注意如果机械臂是 RP, 那么 v_b 和 g_b 是 q 的函数。

c. 科里奥利/向心结构矩阵

我们现在列出本节讨论的 $V(q, \dot{q})$ 的各种结构矩阵。它们的计算留作练习 (见习题)。

科里奥利/向心矩阵:

$$V_{m1}(q, \dot{q}) = \begin{bmatrix} -m_2a_1a_2\dot{\theta}_2 \sin \theta_2 & -m_2a_1a_2(\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 \\ m_2a_1a_2\dot{\theta}_1 \sin \theta_2 & 0 \end{bmatrix} \quad (3.167)$$

$$V_{m2}(q, \dot{q}) = \begin{bmatrix} -m_2a_1a_2\dot{\theta}_2 \sin \theta_2 & m_2a_1a_2\dot{\theta}_1 \sin \theta_2 \\ -m_2a_1a_2(\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 & 0 \end{bmatrix} \quad (3.168)$$

斜对称矩阵 $S(q, \dot{q})$:

$$S = \begin{bmatrix} 0 & -m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 \\ m_2a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 & 0 \end{bmatrix} \quad (3.169)$$

对称矩阵 $V_1(q)$ 、 $V_2(q)$:

$$V_1(q) = \begin{bmatrix} 0 & -m_2a_1a_2 \sin \theta_2 \\ -m_2a_1a_2 \sin \theta_2 & -m_2a_1a_2 \sin \theta_2 \end{bmatrix} \quad (3.170)$$

$$V_2(q) = \begin{bmatrix} m_2a_1a_2 \sin \theta_2 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.171)$$

位置/速度分解矩阵:

$$V_p(q) = \begin{bmatrix} 0 & -m_2 a_1 a_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 \\ m_2 a_1 a_2 \dot{\theta}_1 \sin \theta_2 & 0 \end{bmatrix} \quad (3.172)$$

$$V_v(\dot{q}) = \begin{bmatrix} -m_2 a_1 a_2 \dot{\theta}_2 \sin \theta_2 & -m_2 a_1 a_2 \dot{\theta}_2 \sin \theta_2 \\ 0 & 0 \end{bmatrix} \quad (3.173)$$

d. 机器人函数参数矩阵 W

机器人动力学在参数中是线性的。对于自适应控制的目的, 应该选择表 3.3.1 中的参数向量 ϕ , 使其包含未知参数。

包括摩擦在内的动力学可以写为

$$\tau = W\phi \quad (3.174)$$

其中

$$\phi = \begin{bmatrix} m_1 \\ m_2 \\ v_1 \\ k_1 \\ v_2 \\ k_2 \end{bmatrix} \quad (3.175)$$

第二质量 m_2 包括负载的质量。这个和摩擦系数通常是未知的。因此, 选择

$$\phi = \begin{bmatrix} m_1 a_1^2 + m_2 a_1^2 \\ m_2 a_2^2 \\ m_2 a_1 a_2 \\ m_1 a_1 + m_2 a_1 \\ m_2 a_2 \\ v_1 \\ v_2 \\ k_1 \\ k_2 \end{bmatrix} \quad (3.176)$$

那么已知机器人函数的矩阵 $W(q, \dot{q}, \ddot{q})$ 变为

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & \dot{\theta}_1 & \text{sgn}(\dot{\theta}_1) & 0 & 0 \\ 0 & w_{22} & w_{23} & 0 & w_{25} & 0 & 0 & \dot{\theta}_2 & \text{sgn}(\dot{\theta}_2) \end{bmatrix} \quad (3.177)$$

其中

$$w_{11} = \ddot{\theta}_1 \quad (3.178)$$

$$w_{12} = \ddot{\theta}_1 + \ddot{\theta}_2 \quad (3.179)$$

$$w_{13} = (2\ddot{\theta}_1 + \ddot{\theta}_2) \cos \theta_2 - (2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2 \quad (3.180)$$

$$w_{14} = g \cos \theta_1 \quad (3.181)$$

$$w_{15} = g \cos(\theta_1 + \theta_2) \quad (3.182)$$

$$w_{22} = \ddot{\theta}_1 + \ddot{\theta}_2 \quad (3.183)$$

$$w_{23} = \ddot{\theta}_1 \cos \theta_2 + \dot{\theta}_1^2 \sin \theta_2 \quad (3.184)$$

$$w_{25} = g \cos(\theta_1 + \theta_2) \quad (3.185)$$

读者应该验证，有了这些定义，动力学可以表示为 $\tau = W\phi$ 。矩阵 W 由测量的关节位置及其速度和加速度计算得出。

3.3.5 无源性与能量守恒

表 3.3.1 中给出的机械臂动力学的“牛顿”形式掩盖了一些我们应该在这里探讨的重要物理性质 [Koditschek 1984]、[Ortega and Spong 1988]、[Johansson 1990]、[Slotine and Li 1987]、[Slotine 1988]。注意动力学可以用斜对称矩阵 $S(q, \dot{q})$ 写为

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) = \tau \quad (3.186)$$

其中摩擦和 τ_d 被忽略。现在，以 K 为动能，

$$\dot{K} = \frac{1}{2}\dot{q}^T \dot{M}\dot{q} + \dot{q}^T M\ddot{q} \quad (3.187)$$

因此 (3.3.63) 产生

$$\dot{K} = \frac{1}{2}\dot{q}^T (\dot{M} - 2V_m)\dot{q} + \dot{q}^T (\tau - G) \quad (3.188)$$

或

$$\dot{K} = \frac{1}{2}\dot{q}^T S\dot{q} + \dot{q}^T (\tau - G) \quad (3.189)$$

这是能量守恒的表述，右边表示来自净外力的功率输入。 $S(q, \dot{q})$ 的斜对称性只不过是一个陈述，即虚拟力 $S(q, \dot{q})$ 不做功。外力所做的功由下式给出

$$K(t) - K(0) = \int_0^t \dot{q}^T (\tau - G) dt - \frac{1}{2} \int_0^t \dot{q}^T S \dot{q} dt = \int_0^t \dot{q}^T (\tau - G) dt \quad (3.190)$$

此时回忆机械臂从 $\tau(t)$ 到 \dot{q} 的无源性 (第 1.5 节)，它仅仅说明机械臂不能创造能量。从控制的角度来看，无源系统不能变得不稳定。一些流行的控制方案 (例如，标准计算扭矩，第 3.4 节) 的问题是它们破坏了无源性性质，如果系统参数不完全已知或存

在干扰, 可能导致不稳定。基于无源性的设计确保闭环系统是无源的 (见第 4.3 节、上面引用的参考文献和 [Anderson 1989])。

这个分析不包括摩擦项。无论 $f(\dot{q})$ 的形式如何, 一个合理的假设是摩擦是耗散的, 因此 $f_i(x)$ 仅位于第一和第三象限。这等价于

$$\dot{q}^T F(\dot{q}) \geq 0 \quad (3.191)$$

在这一假设下, 摩擦不会破坏机械臂的无源性。然后可以简单地修改 (3.3.63) 设计的控制器以包含摩擦 [Slotine 1988]。摩擦的耗散性质允许人们将系统的带宽增加到经典极限之外。

3.4 状态变量表示与反馈线性化

机器人机械臂动力学方程在表 3.3.1 中为

$$M(q)\ddot{q} + V(q, \dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) + \tau_d = \tau \quad (3.192)$$

其中 $q(t) \in R^n$ 为关节变量向量, $\tau(t)$ 为控制输入。 $M(q)$ 是惯量矩阵, $V(q, \dot{q})$ 是科里奥利/向心向量, $F_v(\dot{q})$ 是粘性摩擦, $F_d(\dot{q})$ 是动态摩擦, $G(q)$ 是重力, τ_d 是干扰。这些项满足表 3.3.1 中所示的性质。我们也可以将动力学写为

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (3.193)$$

其中非线性项表示为

$$N(q, \dot{q}) = V(q, \dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) + \tau_d \quad (3.194)$$

在本节中, 我们打算展示机械臂动力学方程的一些等价表述。

第 2 章讨论的非线性状态变量表示具有许多从控制角度来看有用的性质。函数 $u(t)$ 是控制输入, $x(t)$ 是状态向量, 描述能量在系统中如何存储。我们在这里展示如何将 (3.4.1) 放入这种形式。在第 4 章中, 我们展示如何使用这种非线性状态变量形式使用计算机仿真机器人臂的行为。在整本书中, 我们将反复使用状态空间表述进行控制设计, 无论是非线性形式还是线性形式

$$\dot{x} = Ax + Bu \quad (3.195)$$

在本节中, 我们还介绍了非线性机器人方程反馈线性化的一般方法, 它涉及以系统的方式重新定义变量, 以产生关于我们感兴趣的动态变量的线性状态方程。这个变量可以是, 例如, 关节变量 $q(t)$ 、笛卡尔位置, 或相机参考系中的位置。

3.4.1 哈密顿公式

机械臂方程是使用拉格朗日力学推导的。在这里，让我们使用哈密顿力学 [Marion 1965] 推导机械臂动力学的状态变量表述 [Arimoto and Miyazaki 1984]、[Gu and Loh 1985]。为了简单起见，让我们忽略摩擦项和干扰 τ_d ；它们可以很容易地在我们的推导结束时添加。

在 3.2 节中，我们将机械臂拉格朗日量表示为

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q) \quad (3.196)$$

其中 $q(t) \in R^n$ 为关节变量， K 为动能， P 为势能， $M(q)$ 为机械臂惯量矩阵。定义广义动量为

$$p = \frac{\partial L}{\partial \dot{q}} = M(q) \dot{q} \quad (3.197)$$

那么我们有

$$\dot{q} = M^{-1}(q)p \quad (3.198)$$

以 $p(t)$ 表示的动能为

$$K(q, p) = \frac{1}{2} p^T M^{-1}(q) p \quad (3.199)$$

值得注意

$$\frac{\partial K}{\partial q} = -\frac{1}{2} (I_n \otimes p^T) \frac{\partial M^{-1}}{\partial q} p = \frac{1}{2} (I_n \otimes p^T) M^{-1} \frac{\partial M}{\partial q} M^{-1} p \quad (3.200)$$

通过定义机械臂哈密顿量为

$$H(q, p) = K(q, p) + P(q) \quad (3.201)$$

哈密顿运动方程为

$$\dot{q} = \frac{\partial H}{\partial p} = M^{-1}(q)p \quad (3.202)$$

$$\dot{p} = -\frac{\partial H}{\partial q} + \tau = -\frac{\partial K}{\partial q} - \frac{\partial P}{\partial q} + \tau \quad (3.203)$$

注意

$$\frac{\partial P}{\partial q} = G(q) \quad (3.204)$$

评估 (3.4.13) 得到

$$\dot{p} = \frac{1}{2} (I_n \otimes p^T) M^{-1} \frac{\partial M}{\partial q} M^{-1} p - G(q) + \tau \quad (3.205)$$

可以表示为 (见习题)

$$\dot{p} = -V_m(q, \dot{q}) \dot{q} - G(q) + \tau \quad (3.206)$$

其中 $G(q)$ 是重力向量, \otimes 是克罗内克积 (见第 3.3 节)。

将状态向量定义为 $x \in R^{2n}$

$$x = \begin{bmatrix} q \\ p \end{bmatrix} \quad (3.207)$$

我们看到机械臂动力学可以表示为

$$\dot{x} = \begin{bmatrix} 0 & I_n \\ 0 & -V_m M^{-1} \end{bmatrix} x + \begin{bmatrix} 0 \\ I_n \end{bmatrix} u - \begin{bmatrix} 0 \\ G \end{bmatrix} \quad (3.208)$$

其中控制输入定义为

$$u = \tau \quad (3.209)$$

这是形式为 (3.4.4) 的非线性状态方程。重要的是要注意, 这个动力学方程在控制输入 u 中是线性的, 它激发广义动量 $p(t)$ 的每个分量。

这个哈密顿状态空间表述在 [Arimoto and Miyazaki 1984] 中使用李雅普诺夫方法推导 PID 控制律, 在 [Gu and Loh 1985] 中推导轨迹跟踪控制。

3.4.2 位置/速度公式

通过将位置/速度状态 $x \in R^{2n}$ 定义为

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (3.210)$$

可以获得机械臂动力学的替代状态空间表述。

为了简单起见, 忽略干扰 τ_d 和摩擦 $F_v + F_d(\dot{q})$, 注意根据 (3.4.2), 我们可以写出

$$\ddot{q} = M^{-1}(q)[\tau - N(q, \dot{q})] \quad (3.211)$$

现在, 我们可以直接写出位置/速度状态空间表示

$$\dot{x} = \begin{bmatrix} \dot{q} \\ -M^{-1}N \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \tau = f(x) + B(x)\tau \quad (3.212)$$

这是 (3.4.4) 的形式, $u(t) = \tau(t)$ 。

也可以写出一个形式为 (3.4.5) 的替代线性状态方程

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u \quad (3.213)$$

其中控制输入定义为

$$u = M^{-1}(q)[\tau - N(q, \dot{q})] \quad (3.214)$$

这两个位置/速度状态空间表述在后续章节中都将证明是有用的。

3.4.3 反馈线性化

现在让我们开发一种通用的方法，用于确定机械臂动力学 (3.4.1)-(3.4.2) 的线性状态空间表示。该技术涉及一种线性化变换，它消除机械臂的非线性。它是 [Hunt et al. 1983, Gilbert and Ha 1984] 中反馈线性化技术的简化版本。另见 [Kreutz 1989]。

机器人动力学由 (3.4.2) 给出， $q \in R^n$ 。让我们定义一种通用类型的输出

$$y = h(q) - s(t) \quad (3.215)$$

其中 $h(q)$ 是关节变量 $q \in R^n$ 的通用预定函数， $s(t)$ 是通用预定时间函数。那么控制问题将是选择关节扭矩和力输入 $\tau(t)$ 以使输出 $y(t)$ 变为零。

$h(q)$ 和 $s(t)$ 的选择基于我们心中的控制目标。例如，如果 $h(q) = -q$ 和 $s(t) = q_d(t)$ ，我们希望机械臂遵循的期望关节空间轨迹，那么 $y(t) = q_d(t) - q(t) = e(t)$ ，关节空间跟踪误差。在这种情况下强制 $y(t)$ 为零将使关节变量 $q(t)$ 跟踪它们的期望值 $q_d(t)$ ，导致机械臂轨迹跟踪。

作为另一个例子， $h(q)$ 可以表示笛卡尔空间跟踪误差， y_d 为位置误差， $e_0 \in R^3$ 为方向误差。将 $y(t)$ 控制为零然后将导致直接在笛卡尔空间中的轨迹跟踪，毕竟，期望运动通常是在那里指定的。

最后， $-h(q)$ 可以表示到相机坐标系的非线性变换， $s(t)$ 是该坐标系中的期望轨迹。那么 $y(t)$ 是相机坐标系跟踪误差。强制 $y(t)$ 为零将产生相机空间中的跟踪运动。

反馈线性化变换

为了确定用于机器人控制器设计的线性状态变量模型，让我们简单地对输出 $y(t)$ 求导两次以获得

$$\dot{y} = \frac{\partial h}{\partial q} \dot{q} - \dot{s} = J\dot{q} - \dot{s} \quad (3.216)$$

$$\ddot{y} = J\ddot{q} + \dot{J}\dot{q} - \ddot{s} \quad (3.217)$$

其中我们定义了雅可比矩阵

$$J(q) \equiv \frac{\partial h}{\partial q} \quad (3.218)$$

如果 $y \in R^p$ ，雅可比矩阵是 $p \times n$ 形式的矩阵

$$J = \begin{bmatrix} \frac{\partial h_1}{\partial q_1} & \cdots & \frac{\partial h_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial q_1} & \cdots & \frac{\partial h_p}{\partial q_n} \end{bmatrix} \quad (3.219)$$

给定函数 $h(q)$ ，计算与 $h(q)$ 相关的雅可比矩阵 $J(q)$ 是直接的。在特殊情况下，如果 \dot{q} 表示笛卡尔速度， $J(q)$ 是附录 A 中讨论的机械臂雅可比矩阵。那么，如果所有关节都是旋转的， J 的单位是长度单位。

根据 (3.4.2)，

$$\ddot{q} = M^{-1}(\tau - N) \quad (3.220)$$

因此 (3.4.26) 产生

$$\ddot{y} = JM^{-1}(\tau - N) + \dot{J}\dot{q} - \ddot{s} \quad (3.221)$$

定义控制输入函数

$$u = JM^{-1}(\tau - N) + \dot{J}\dot{q} - \ddot{s} \quad (3.222)$$

和干扰函数

$$w = JM^{-1}\tau_d \quad (3.223)$$

现在我们可以将状态 $x(t) \in R^{2p}$ 定义为

$$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \quad (3.224)$$

并将机器人动力学写为

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ -w \end{bmatrix} = Ax + Bu + \begin{bmatrix} 0 \\ -w \end{bmatrix} \quad (3.225)$$

这是形式为

$$\dot{x} = Ax + Bu + \begin{bmatrix} 0 \\ -w \end{bmatrix} \quad (3.226)$$

的线性状态空间系统由控制输入 $u(t)$ 和干扰 $v(t)$ 驱动。由于 A 和 B 的特殊形式，该系统被称为布鲁诺夫斯基标准形式 (第 2 章)。读者应该确定可控性矩阵以验证它总是从 $u(t)$ 可控。

方程 (3.4.31) 被称为机器人动力学方程的线性化变换。我们可以反转这个变换以获得

$$\tau = MJ^+(u - \dot{J}\dot{q} + \ddot{s}) + N + J^+w \quad (3.227)$$

其中 J^+ 是雅可比矩阵 $J(q)$ 的摩尔-彭若斯逆 [Rao and Mitra 1971]。如果 $J(q)$ 是方阵 (即 $p = n$) 且非奇异, 那么 $J^+(q) = J^{-1}(q)$, 我们可以写出

$$\tau = MJ^{-1}(u - \dot{J}\dot{q} + \ddot{s}) + N + J^{-1}w \quad (3.228)$$

正如我们将在第 4 章看到的, 反馈线性化提供了一种强大的控制设计技术。事实上, 如果我们选择 $u(t)$ 使得 (3.4.34) 稳定 (例如, 一种可能性是 PD 反馈 $u = -K_v\dot{y} - K_p y$), 那么由 (3.4.36) 定义的控制输入扭矩 $\tau(t)$ 使机器人臂以 $y(t)$ 变为零的方式移动。

在特殊情况 $y(t) = q(t)$, 那么 $J = I$, (3.4.34) 简化为线性位置/速度形式 (3.4.22)。

3.5 笛卡尔动力学与其他动力学

在 3.2 节中，我们以 $q(t)$ 的时间行为推导了机器人动力学。根据表 3.3.1，

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (3.229)$$

或

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d = \tau \quad (3.230)$$

其中非线性项为

$$N(q, \dot{q}) = V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d \quad (3.231)$$

我们称之为以关节空间形式表述的机械臂动力学，或简称为关节空间动力学。

3.5.1 笛卡尔机械臂动力学

拥有除关节变量 $q(t)$ 以外的变量的动力学发展描述通常是有用的。因此，定义

$$y = h(q) \quad (3.232)$$

其中 $h(q)$ 通常是来自运动学的非线性变换。虽然 $y(t)$ 可以是任何感兴趣的变量，让我们在这里将其视为末端执行器的笛卡尔或任务空间位置（即在基坐标中末端执行器的位置和方向）。

从关节空间动力学到笛卡尔动力学的推导类似于 3.4 节中的反馈线性化。将 (3.5.4) 求导两次得到

$$\dot{y} = \frac{\partial h}{\partial q} \dot{q} = J\dot{q} \quad (3.233)$$

$$\ddot{y} = J\ddot{q} + \dot{J}\dot{q} \quad (3.234)$$

其中雅可比矩阵为

$$J = \frac{\partial h}{\partial q} \quad (3.235)$$

笛卡尔速度向量为 $\dot{y} = [v^T \ \omega^T]^T$ ，其中 $v \in R^3$ 是线速度， $\omega \in R^3$ 是角速度。让我们假设连杆数为 $n = 6$ ，因此 J 是方阵。还假设我们远离工作空间奇异点，因此 $|J| \neq 0$ ，根据 (3.5.6)，我们可以写出

$$\ddot{q} = J^{-1}\ddot{y} - J^{-1}\dot{J}\dot{q} \quad (3.236)$$

这是“逆加速度”变换。将其代入 (3.5.2) 得到

$$MJ^{-1}\ddot{y} - MJ^{-1}\dot{J}\dot{q} + N = \tau \quad (3.237)$$

现在回忆力变换 $\tau = J^T F$ ，其中 F 是笛卡尔力向量（见附录 A），我们有

$$MJ^{-1}\ddot{y} - MJ^{-1}\dot{J}J^{-1}\dot{y} + N = J^T F \quad (3.238)$$

这可以写为

$$M'\ddot{y} + N' = F \quad (3.239)$$

其中我们定义了笛卡尔惯量矩阵、非线性项和干扰为

$$M'(q) = J^{-T} M(q) J^{-1} \quad (3.240)$$

$$N'(q, \dot{q}) = J^{-T} [N(q, \dot{q}) - M(q) J^{-1} \dot{J} \dot{q}] \quad (3.241)$$

$$\tau'_d = J^{-T} \tau_d \quad (3.242)$$

方程 (3.5.9)-(3.5.10) 给出了机器人机械臂的笛卡尔或工作空间动力学。

注意, M' 、 N' 和 τ'_d 取决于 q 和 \dot{q} , 因此严格来说, 笛卡尔动力学并没有完全以 y 给出。然而, 给定 $y(t)$, 我们可以使用逆运动学确定 $q(t)$, 因此 M' 、 N' 、 τ'_d 可以作为 y 和 \dot{y} 的函数使用计算机子程序计算。

3.5.2 笛卡尔动力学的结构与性质

重要的是要认识到, 只要 J 是非奇异的, 表 3.3.1 中列出的关节空间动力学的所有性质都适用于笛卡尔动力学 [Slotine and Li 1987]。特别要注意, M' 是对称且正定的。对于旋转机械臂, 雅可比矩阵具有长度单位且是有界的。在这种情况下, M' 上下有界。

定义

$$V' = J^{-T} (V - M J^{-1} \dot{J} J^{-1} \dot{y}) = J^{-T} (V - M J^{-1} \dot{J} \dot{q}) \quad (3.243)$$

它遵循

$$V' = V'_m \dot{y} \quad (3.244)$$

其中

$$V'_m = J^{-T} V_m J^{-1} - J^{-T} M J^{-1} \dot{J} J^{-1} \quad (3.245)$$

其中 V_m 在第 3.3 节中定义。

很容易证明

$$S' \equiv \dot{M}' - 2V'_m \quad (3.246)$$

是斜对称的。确实, 使用恒等式

$$\frac{dJ^{-1}}{dt} = -J^{-1} \dot{J} J^{-1} \quad (3.247)$$

我们可以看到

$$S' = J^{-T} S J^{-1} \quad (3.248)$$

这是斜对称的, 因为 S 是斜对称的。

笛卡尔动力学中的摩擦项为

$$F' = J^{-T} F(\dot{q}) = J^{-T} F(J^{-1} \dot{y}) \equiv F'(\dot{y}) \quad (3.249)$$

它们满足表 3.3.1 中那样的界限。注意在笛卡尔坐标中，摩擦效应不是解耦的（例如， $J^{-T}F_v J^{-1}$ 不是对角矩阵）。笛卡尔重力向量

$$G' = J^{-T}G(q) \quad (3.250)$$

是有界的。

参数线性性质成立并表示为

$$M'\ddot{y} + V' + F' + G' = W'(y, \dot{y}, \ddot{y})\phi \quad (3.251)$$

其中已知笛卡尔机器人函数为

$$W' = J^{-T}W \quad (3.252)$$

ϕ 是机械臂参数向量。

例 3.5-1: 三连杆圆柱形机械臂的笛卡尔动力学

让我们展示如何将例 3.2.3 中找到的关节空间动力学转换为笛卡尔动力学。从例 A.3-1，机械臂雅可比矩阵为

$$J = \begin{bmatrix} -r \sin \theta & 0 & \cos \theta \\ r \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \end{bmatrix} \quad (3.253)$$

因此其逆为

$$J^{-1} = \begin{bmatrix} -\frac{\sin \theta}{r} & \frac{\cos \theta}{r} & 0 \\ 0 & 0 & 1 \\ \cos \theta & \sin \theta & 0 \end{bmatrix} \quad (3.254)$$

从例 3.2.3，机械臂惯量矩阵为

$$M = \begin{bmatrix} J + m_1 r^2 + m_2 r^2 & 0 & 0 \\ 0 & m_1 + m_2 & 0 \\ 0 & 0 & m_2 \end{bmatrix} \quad (3.255)$$

应用 (3.5.11) 得到（验证！）

$$M' = \begin{bmatrix} \frac{J + m_1 r^2 + m_2 r^2}{r^2} & 0 & 0 \\ 0 & m_1 + m_2 & 0 \\ 0 & 0 & m_2 \end{bmatrix} \quad (3.256)$$

其中 $J' = \frac{J}{r^2} + m_1 + m_2$ 。

以类似的方式，可以计算 N' 。

3.6 执行器动力学

我们已经在关节空间坐标和笛卡尔坐标中讨论了刚性机器人机械臂的动力学。然而，机器人需要执行器来移动它；这些通常是电动或液压马达。因此，现在需要将执行器动力学添加到机械臂动力学中，以获得机械臂加执行器系统的完整动力学描述。关于执行器和传感器的好参考文献由 [de Silva 1989] 提供。

3.6.1 带执行器的机械臂动力学

我们将考虑电动执行器的情况，假设电机是电枢控制的。液压执行器由类似的方程描述。在本小节中，我们假设电枢电感可以忽略。

n -连杆机器人机械臂的方程来自表 3.3.1:

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d = \tau \quad (3.257)$$

其中 $q \in R^n$ 是机械臂关节变量。驱动连杆的电枢控制直流电机的动力学由 n 个解耦方程给出

$$J_M \ddot{q}_M + B_M \dot{q}_M + F_M + \tau = K_M v \quad (3.258)$$

其中 $\dot{q}_M = \text{vec}\{\dot{q}_{Mi}\}$, q_{Mi} 是第 i 个转子位置角, $\text{vec}\{\alpha_i\}$ 表示分量为 α_i 的向量。控制输入是电机电压向量 $v \in R^n$ 。

执行器系数矩阵都是常数，由下式给出

$$J_M = \text{diag}\{J_{Mi}\}, \quad B_M = \text{diag}\{B_{Mi}\}, \quad K_M = \text{diag}\{K_{Mi}R_{ai}^{-1}\} \quad (3.259)$$

其中第 i 个电机具有惯量 J_{Mi} 、转子阻尼常数 B_{Mi} 、反电动势常数 K_{bi} 、扭矩常数 K_{Mi} 和电枢电阻 R_{ai} 。

从第 i 个电机到第 i 个机械臂连杆的耦合齿轮比为 r_i ，我们定义使得

$$q_i = r_i q_{Mi} \quad \text{或} \quad q = R q_M \quad (3.260)$$

如果第 i 个关节是旋转的，那么 r_i 是小于 1 的无量纲常数。如果 q_i 是平移的，那么 r_i 具有 m/rad 单位。

执行器摩擦向量由下式给出

$$F_M = \text{vec}\{F_{Mi}\} \quad (3.261)$$

其中 F_{Mi} 是第 i 个转子的摩擦。

注意大写的“M”表示电机常数和变量，而 V_m 是用克里斯托费尔符号定义的机械臂科里奥利/向心向量。

使用 (3.6.4) 在 (3.6.2) 中消去 q_M ，然后代入 (3.6.1) 中的 τ ，得到以关节变量表示的动力学

$$(M + R^2 J_M)\ddot{q} + (V + R^2 B_M)\dot{q} + (F + R F_M) + G + \tau_d = R K_M v \quad (3.262)$$

或，通过适当定义符号，

$$M'(q)\ddot{q} + V'(q, \dot{q}) + F'(\dot{q}) + G(q) + \tau_d' = K'v \quad (3.263)$$

完整机械臂加执行器动力学的性质

完整动力学 (3.6.6) 与机器人动力学 (3.6.1) 具有相同的形式。很容易验证完整机械臂加执行器动力学享有与表 3.3.1 中列出的机械臂动力学相同的性质（见习题）。特别地， V' 是 M' 和斜对称矩阵之间差的一半，所有有界性假设都成立，参数线性成立。因此，在我们设计控制器时，我们可以假设执行器已经包含在表 3.3.1 中的机械臂方程中。

独立关节动力学

在许多商业机器人机械臂中，齿轮比 r_i 非常小，在执行器/连杆耦合中提供很大的扭矩优势。这对机器人机械臂控制器的设计有重要的影响，大大简化了设计。

为了探索这一点，让我们按分量写出完整动力学

$$(J_{M_i} r_i^2 + m_{ii})\ddot{q}_i + (B_{M_i} r_i^2 + v_i)\dot{q}_i + k_i \text{sgn}(\dot{q}_i) + d_i = k_i' v_i \quad (3.264)$$

其中 $B \equiv \text{diag}\{B_i\}$ ， d_i 是由下式给出的干扰

$$d_i = \sum_{j \neq i} m_{ij} \ddot{q}_j + \sum_{j,k} V_{jki} \dot{q}_j \dot{q}_k + g_i + f_i(\dot{q}_i) \quad (3.265)$$

其中 m_{ij} 是 M' 的非对角元素， V_{jki} 是第 i 个连杆摩擦的张量分量， G_i 是第 i 个重力分量。

这个方程表明，如果 r_i 很小，机械臂动力学近似由 n 个具有常数系数的解耦二阶方程给出。关节耦合和重力的动力学效应仅作为干扰项乘以 r_i^2 出现。也就是说，机器人控制设计实际上只是控制执行器动力学的问题。

不幸的是，现代高性能任务使科里奥利和向心项变大，因此 d_i 不小。此外，现代高性能机械臂具有接近统一的齿轮比（例如，直驱机械臂），因此非线性必须在任何认真的控制设计中考虑。

3.6.2 三阶机械臂加执行器动力学

有时在控制设计中使用完整机器人机械臂的替代模型 [Tarn et al. 1991]。它是一个三阶微分方程，当电机电枢电感不可忽略时应该使用。

当电枢电感 L_i 不可忽略时，我们必须使用电枢控制直流电机方程代替 (3.6.2)

$$L \frac{dI}{dt} + RI + K_b' \dot{q}_M = v \quad (3.266)$$

$$J_M \ddot{q}_M + B_M \dot{q}_M + F_M + \tau = K_T I \quad (3.267)$$

其中 $I \in R^n$ 是电枢电流向量,

$$L = \text{diag}\{L_i\}, \quad R = \text{diag}\{R_i\}, \quad K_T = \text{diag}\{K_{T_i}\}, \quad K'_b = \text{diag}\{K_{b_i} r_i^{-1}\} \quad (3.268)$$

重要的是要注意, T 是电机电气时间常数矩阵。在前一小节中, 假设这些时间常数与电机机械时间常数相比可以忽略不计。

为了确定机械臂加直流电机执行器的整体动力学, 在 (3.6.1) 和 (3.6.10) 之间消去 τ 以获得 I 的表达式。然后, 求导以显式暴露 \dot{I} 。将这些表达式代入 (3.6.9) (见习题) 以获得形式为

$$L' M'(q) \ddot{q} + D(q, \dot{q}, \ddot{q}) = v \quad (3.269)$$

的动力学。

系数矩阵 D 由下式给出

$$D(q) = T M'(q) \quad (3.270)$$

因此当 L_i 很小时可以忽略不计。

3.6.3 具有关节柔性的动力学

我们假设执行器和机器人连杆之间的耦合通过具有齿轮比 r_i 的刚性齿轮系提供。在实际实践中, 耦合存在齿隙和齿轮系柔性或弹性。这里我们在机械臂动力学模型中包含关节的柔性, 为了简单起见, 假设 $r_i = 1$ 。

这并不难做到。确实, 假设耦合柔性被建模为刚性弹簧。那么在方程 (3.6.1)、(3.6.2) 中提到的扭矩只不过是

$$\tau = B_s(\dot{q}_M - \dot{q}) + K_s(q_M - q) \quad (3.271)$$

其中 $B_s = \text{diag}\{b_{s_i}\}$, $K_s = \text{diag}\{k_{s_i}\}$, b_{s_i} 和 k_{s_i} 是第 i 个齿轮系的阻尼和弹簧常数。因此动力学方程变为

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + B_s(\dot{q} - \dot{q}_M) + K_s(q - q_M) = 0 \quad (3.272)$$

$$J_M \ddot{q}_M + B_M \dot{q}_M + F_M - B_s(\dot{q}_M - \dot{q}) - K_s(q_M - q) = K_M v \quad (3.273)$$

这些方程的结构与表 3.3.1 中描述的刚性关节机械臂非常不同。我们在第 6 章讨论具有关节柔性的机器人机械臂的控制 (见 [Spong 1987])。下一个例子展示了在控制柔性关节机器人时可能出现的问题。

例 3.6-1: 带柔性耦合轴的直流电机

为了关注关节柔性的影响, 让我们检查单个电枢控制的直流电机通过具有显著柔性的轴耦合到负载。电气和机械子系统如图 3.6.1 所示。

电机电气方程为

$$L \frac{di}{dt} + Ri + K_b \omega_m = u \quad (3.274)$$

其中 $i(t)$ 、 $u(t)$ 分别是电枢电流和电压。反电动势为 $e_b = K_b \omega_m$ 。

柔性轴施加的相互作用力由下式给出 $f = b(\omega_m - \omega_L) + k(\theta_m - \theta_L)$ ，其中轴阻尼和弹簧常数分别表示为 b 和 k 。因此运动方程可以写为

$$J_m \dot{\omega}_m + b_m \omega_m + b(\omega_m - \omega_L) + k(\theta_m - \theta_L) = K_T i \quad (3.275)$$

$$J_L \dot{\omega}_L - b(\omega_m - \omega_L) - k(\theta_m - \theta_L) = 0 \quad (3.276)$$

其中下标 m 和 L 分别指电机参数和负载参数。假设负载惯量 J_L 是常数。其余符号的定义可以从前面的文本中推断。

为了将这些方程放入状态空间形式，将状态定义为

$$x = \begin{bmatrix} \theta_L \\ \theta_m \\ \omega_L \\ \omega_m \\ i \end{bmatrix} \quad (3.277)$$

其中 ω_m 、 ω_L 是电机和负载的角速度。那么

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{k}{J_L} & \frac{k}{J_L} & -\frac{b}{J_L} & \frac{b}{J_L} & 0 \\ \frac{k}{J_m} & -\frac{k}{J_m} & \frac{b}{J_m} & -\frac{(b+b_m)}{J_m} & \frac{K_T}{J_m} \\ 0 & 0 & 0 & -\frac{K_b}{L} & -\frac{R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u \quad (3.278)$$

a. 刚性耦合轴

如果耦合轴没有柔性， $\omega_m = \omega_L = \omega$ ，状态方程简化为（见习题）

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{(b_m+b_L)}{J} & \frac{K_T}{J} \\ 0 & -\frac{K_b}{L} & -\frac{R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u \quad (3.279)$$

其中 $x = [i \ \omega]^T$ ， $J = J_m + J_L$ 。将输出定义为电机速度得到

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x \quad (3.280)$$

传递函数计算为

$$H(s) = \frac{K_T}{(Ls + R)[(J_m + J_L)s + (b_m + b_L)] + K_T K_b} \quad (3.281)$$

使用参数值 $J_m = J_L = 0.1 \text{ kg}\cdot\text{m}^2$, $L = 0.5 \text{ H}$, $b_m = 0.2 \text{ N}\cdot\text{m}/\text{rad}/\text{s}$, 和 $R = 5 \Omega$ 得到

$$H(s) = \frac{20}{(s + 2.3)(s + 8.7)} \quad (3.282)$$

因此有两个实极点在 $s = -2.3$, $s = -8.7$ 。

使用附录 B 中的程序 TRESP 进行仿真 (见第 3.3 节) 产生图 3.6.2 中所示的 ω 阶跃响应。

b. 非常柔性的耦合轴

对应于非常柔性轴的耦合轴参数 $k = 2 \text{ N}\cdot\text{m}/\text{rad}$ 和 $b = 0.2 \text{ N}\cdot\text{m}/\text{rad}/\text{s}$ 。使用这些值, 可以使用 PC-MATLAB 等软件获得两个传递函数

$$H_m(s) = \frac{20s(s^2 + 3s + 44)}{(s + 10)[(s + 3.4)^2 + 5.6^2]} \quad (3.283)$$

$$H_L(s) = \frac{40(s + 10)}{(s + 10)[(s + 3.4)^2 + 5.6^2]} \quad (3.284)$$

轴柔性模式有极点 $s = -3.4 \pm j5.6$, 因此阻尼比为 $\zeta = 0.52$, 自然频率为 $\omega = 6.55 \text{ rad}/\text{s}$ 。注意系统是边际稳定的, 有一个极点在 $s = 0$ 。由于极零点抵消, 它是 BIBO 稳定的。

程序 TRESP 产生图 3.6.3 中所示的阶跃响应。有几点值得注意。最初, 电机速度 ω_m 比图 3.6.2 中上升得更快, 因为轴柔性意味着只有转子转动惯量 J_m 最初影响速度。然后, 当负载 J_L 通过轴耦合回电机时, ω_m 的增加速率减慢。还要注意, 负载速度 ω_L 由于轴中的柔性表现出大约 0.1 s 的延迟。

极其有趣的是要注意, 轴柔性具有加速最慢电机实极点的效果 [比较 (8) 和 (9)], 因此 ω_L 比刚性轴情况下更快地接近其稳态值。这是由于柔性轴的“甩动”作用。

轴动力学使得在没有某种专门设计的控制器的情况下, 控制 θ_L (在机器人机械臂中对应于关节角 q_i) 非常困难。

3.7 总结

在本章中, 我们为机器人控制系统的研究奠定了基础。在 3.2 节中使用拉格朗日力学, 我们推导了一些将在整本书中用于演示设计的机械臂动力学。我们为任何串联连杆机械臂提供了通用机器人机械臂动力学的表达式。

在 3.3 节中, 我们研究了机器人动力学的性质, 如有界性、参数线性和斜对称性, 这些在控制设计中是必需的。表 3.3.1 给出了我们发现的总结。我们使用了一种克罗内克积方法, 该方法对机器人方程中各项之间的关系提供了很好的洞察。

现代控制系统设计中的一个重要形式是状态变量形式。在 3.4 节中, 我们推导了机械臂动力学的几种状态空间形式, 为后续章节中提供的几种设计技术奠定了基础。状态形式在机器人控制器的计算机仿真中也是有用的, 如我们在第 4 节中看到的那样。

笛卡尔形式的动力学在 3.5 节中给出。驱动机器人机械臂连杆的执行器的动力学在 3.6 节中分析并包含。

3.8 参考文献

- Anderson 1989 Anderson, R.J., "Passive computed torque algorithms for robots," Proc. IEEE Conf. Decision Control, pp. 1638–1644, Dec. 1989.
- and Miyazaki 1984 Arimoto, S., and F.Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capability," Proc. First Int. Symp., pp. 783–799, MIT, Cambridge, MA, 1984.
- a and Slotine 1986 Asada, H., and J.-J.E.Slotine, *Robot Analysis and Control*, New York: Wiley, 1986.
- and Tarapov 1968 Borisenko, A.I., and I.E.Tarapov, *Vector and Tensor Analysis with Applications*. Englewood Cliffs, NJ: Prentice Hall, 1968.
- Brewer 1978 Brewer, J.W., "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. CAS-25, no. 9, pp. 772–781, Sept. 1978.
- Craig 1988 Craig, J.J., *Adaptive Control of Mechanical Manipulators*. Reading, MA: Addison-Wesley, 1988.
- de Silva 1989 de Silva, C.W., *Control Sensors and Actuators*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- ilbert and Ha 1984 Gilbert, E.G., and I.J.Ha, "An approach to nonlinear feedback control with applications to robotics," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-14, no. 6, pp. 879–884, Nov./Dec. 1984.
- Gu and Loh 1985 Gu, Y.-L., and N.K.Loh, "Dynamic model for industrial robots based on a compact Lagrangian formulation," Proc. IEEE Conf. Decision Control, pp. 1497–1501, 1985.
- Gu and Loh 1988 Gu, Y-L., and N.K.Loh, "Dynamic modeling and control by utilizing an imaginary robot model," *IEEE J. Robot. Autom.*, vol. 4, no. 5, pp. 532–534, Oct. 1988.
- Hunt et al. 1983 Hunt, L.R., R.Su, and G.Meyer, "Global transformations of nonlinear systems," *IEEE Trans. Autom. Control*, vol. AC-28, no. 1, pp. 24–31, Jan. 1983.
- Johansson 1990 Johansson, R., "Quadratic optimization of motion coordination and control," *IEEE Trans. Autom. Control*, vol. 35, no. 11, pp. 1197–1208, Nov. 1990.

- Koditschek 1984 Koditschek, D., "Natural motion for robot arms," Proc. IEEE Conf. Decision Control, pp. 733–735, Dec. 1984.
- Kreutz 1989 Kreutz, K., "On manipulator control by exact linearization," *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 763–767, July 1989.
- Lee et al. 1983 Lee, C.S.G., R.C.Gonzalez, and K.S.Fu, *Tutorial on Robotics*. New York: IEEE Press, 1983.
- Marion 1965 Marion, J.B., *Classical Dynamics*. New York: Academic Press, 1965.
- Ortega and Spong 1988 Ortega, R., and Spong, M.W., "Adaptive motion control of rigid robots: a tutorial," Proc. IEEE Conf. Decision Control, pp. 1575–1584, Dec. 1988.
- Paul 1981 Paul, R.P., *Robot Manipulators*. Cambridge, MA: MIT Press, 1981.
- Rao and Mitra 1971 Rao, C.R., and S.K.Mitra, *Generalized Inverse of Matrices and Its Applications*. New York: Wiley, 1971.
- Schilling 1990 Schilling, R.J., *Fundamentals of Robotics*. Englewood Cliffs, NJ: Prentice Hall, 1990.
- Slotine 1988 Slotine, J.-J.E., "Putting physics in control: the example of robotics," *IEEE Control Syst. Mag.*, pp. 12–17, Dec. 1988.
- Slotine and Li 1987 Slotine, J.-J.E., and W.Li, "Adaptive strategies in constrained manipulation," Proc. IEEE Conf. Robot. Autom., pp. 595–601, 1987.
- Spong 1987 Spong, M.W., "Modeling and control of elastic joint robots," *J. Dyn. Syst. Meas. Control*, vol. 109, pp. 310–319, Dec. 1987.
- Spong and Vidyasagar 1989 Spong, M.W., and M.Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.
- Tarn et al. 1991 Tarn, T.-J., A.K.Bejczy, X.Yun, and Z.Li, "Effect of motor dynamics on nonlinear feedback robot arm control," *IEEE Trans. Robot. Autom.*, vol. 7, no. 1, pp. 114–122, Feb. 1991.

3.9 习题

第 3.2 节

3.2-1 动力学。求例 A.2-4 中球腕的动力学。

3.2-2 **从推导方程的动力学**。在例 3.2.2 中，我们从基本原理找到了二连杆平面肘形机械臂的动力学。在本题中，从该例子中动能和势能的表达式开始，并：

(a) 将 K 写成形式 (3.2.29) 以确定 $M(q)$ 。

(b) 使用 (3.2.42) 和 (3.2.43) 确定 $V(q, \dot{q})$ 和 $G(q)$ 。

3.2-3 **从推导方程的动力学**。对例 3.2-3 中的三连杆机械臂重复问题 3.2-2。

第 3.3 节

3.3-1 通过找到 $V_{p1}(q)$ 和 $V_{v1}(q)$ 证明 (3.3.22)。

3.3-2 通过找到矩阵 $V_i(q)$ 证明 (3.3.23)。

3.3-3 证明 (3.3.27)。

3.3-4 **科里奥利项**。在 (3.3.40) 中找到 $V_{cor}(q)$ 和 $V_{cen}(q)$ 。

3.3-5 **科里奥利项**。证明动力学方程中的科里奥利/向心项可以表示为 $V(q, \dot{q}) = \text{vec}\{V(q, \dot{q})\}$ ，其中

$$V = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \text{tr} \left(\frac{\partial T_i}{\partial q_j} M_i \frac{\partial T_i^T}{\partial q_k} \right) \dot{q}_j \dot{q}_k$$

其中

$$\frac{\partial T_i}{\partial q_j} = T_i Q_j$$

和附录 A 中定义的 T_i 。将其与第 3.2 节中定义的 V_{m1} 、 V_{m2} 、 V_m 进行比较。

3.3-6 **界限与结构**。详细推导例 3.3.1 中的结果。

3.3-7 **界限与结构**。为例 3.2.1 中的二连杆极坐标机械臂推导界限和结构矩阵。使用：

(a) 1-范数。

(b) 2-范数。

(c) ∞ -范数。

3.3-8 **界限与结构**。对习题 3.2.3 中的三连杆圆柱形机械臂重复问题 3.3-7。

3.3-9 **使用 2-范数的界限**。使用 2-范数推导例 3.3.1 中二连杆平面肘形机械臂的界限。

第 3.4 节

3.4-1 证明 (3.4.15)。

3.4-2 **哈密顿状态公式**。证明 (3.4.15) 等价于

$$\dot{p} = -V_m^T(q, \dot{q})\dot{q} - G(q) + \tau$$

其中斜对称矩阵在第 3.3 节中定义。

3.4-3 **哈密顿状态公式**。使用 (3.4.17) 为例 3.2.1 中的二连杆极坐标机械臂推导哈密顿状态变量公式。

3.4-4 **哈密顿状态公式**]。为例 3.2.2 中的二连杆平面肘形机械臂重复问题 3.4-3。

第 3.5 节

3.5-1 **笛卡尔动力学**。完成例 3.5.1, 计算笛卡尔坐标中的非线性项。

3.5-2 **笛卡尔动力学**。求例 3.2.1 中二连杆极坐标机械臂的笛卡尔动力学。

3.5-3 **笛卡尔动力学**。求例 3.2.2 中二连杆平面肘形机械臂的笛卡尔动力学。

第 3.6 节

3.6-1 **执行器动力学**。验证机械臂加执行器动力学 (3.6.6) 具有表 3.3.1 中列出的性质。

3.6-2 **执行器动力学**。推导三阶动力学 (3.6.12), 提供 τ 、 N 的显式表达式。验证当 L_i 可忽略时它们简化为 (3.6.5)。

3.6-3 **柔性耦合轴**。验证例 3.6.1 中刚性轴情况的状态方程。

Chapter 4

计算力矩控制

4.1 引言

控制机器人的一个基本问题是使操作臂跟随预先规划的期望轨迹。在机器人执行任何有用的工作之前，我们必须在正确的时刻将其定位在正确的位置。本章讨论计算力矩控制，它产生一系列易于理解的控制方案，在实践中通常效果很好。这些方案涉及将控制设计问题分解为内环设计和外环设计。

在 4.4 节中，我们将提供基于 PID 控制的独立关节设计的经典操作臂控制方案的联系。在 4.6 节中，我们将展示如何结合计算力矩控制使用一些现代设计技术。因此，本章可以被视为连接几年前机器人控制中使用的经典设计技术与本书其余部分中需要的现代设计技术之间的桥梁，以在不确定环境中获得高性能。

这里我们假设机器人在自由空间中移动，不与环境接触。接触会导致力的产生。力控制问题在第 7 章中讨论。我们还将假设在本章中机器人是一个已知的刚体系统，因此基于一个相当已知的模型设计控制器。在存在不确定性或未知参数（例如摩擦、负载质量）的情况下进行控制需要精细的方法。这个问题在第 4 章使用鲁棒控制处理，在第 5 章使用自适应控制处理。

实际的机器人操作臂可能在其连杆中具有柔性，或在其齿轮中具有顺应性（关节柔性）。在第 6 章中，我们涵盖了带关节柔性的控制的一些方面。

在我们可以控制机器人手臂之前，有必要知道执行任务的期望路径。与路径规划问题相关的有许多问题，例如避免障碍物以及确保规划的路径不需要超过执行器的电压和力矩限制。为了将控制问题简化为基本组成部分，在本章中，我们假设最终的控制目标是使机器人沿着规定的期望轨迹移动。我们不关心实际的轨迹规划问题；但是，我们确实展示了如何重建一个连续的期望路径，该路径从给定期望点表（末端执行器应通过这些点）中得出。这个连续路径生成问题在 4.2 节中介绍。

在大多数情况下，机器人控制器在微处理器上实现，特别是考虑到现代控制方案的复杂性质。因此，在 4.5 节中，我们说明了机器人控制器数字实现的一些概念。

贯穿始终，我们演示如何在计算机上模拟机器人控制器。在实际在真实机器人操作臂上实现之前，应该这样做以验证任何提议的控制方案的有效性。

4.2 路径生成

在本书中，我们假设给定了一条规定路径 $q_d(t)$ ，机器人手臂应该跟随它。我们设计控制方案使操作臂跟随这个期望路径或轨迹。轨迹规划涉及找到规定路径，通常被视为一个单独的设计问题，涉及碰撞避免、执行器饱和等考虑。参见 [Lee et al. 1983]。

我们不涵盖轨迹规划。但是，我们确实涵盖轨迹生成的两个方面。首先，我们展示了如何将给定的规定路径从笛卡尔空间转换到关节空间。然后，给定一个期望点表（末端执行器应通过这些点），我们展示了如何重建一个连续的期望轨迹。

4.2.1 将笛卡尔轨迹转换到关节空间

在机器人应用中，期望任务通常在工作空间或笛卡尔空间中指定，因为在这里操作臂的运动很容易相对于外部环境和工作件进行描述。然而，轨迹跟随控制在关节空间中很容易执行，因为这里是臂动力学更容易表述的地方。

因此，重要的是能够找到给定期望笛卡尔轨迹的期望关节空间轨迹 $q_d(t)$ 。这通过使用逆运动学来完成，如下面示例所示。该示例说明了笛卡尔到关节空间轨迹的映射可能不是唯一的——也就是说，几个关节空间轨迹可能对末端执行器产生相同的笛卡尔轨迹。

示例 4.2-1: 将规定笛卡尔轨迹映射到关节空间

在示例 A.3-5 中推导出了图 4.2.1 所示的两连杆平面机器人手臂的逆运动学。让我们使用它们将路径从笛卡尔空间转换到关节空间。

假设我们希望两连杆臂在 (x, y) 平面中跟随给定的工作空间或笛卡尔轨迹

$$p(t) = (x(t), y(t)) \quad (4.1)$$

它是时间 t 的函数。由于手臂由控制其角度 θ_1, θ_2 的执行器移动，因此方便将指定的笛卡尔轨迹 $(x(t), y(t))$ 转换为关节空间轨迹 $(\theta_1(t), \theta_2(t))$ 以进行控制。

这可以通过使用逆运动学变换来实现

$$r^2 = x^2 + y^2 \quad (4.2)$$

$$C = \cos \theta_2 = \frac{r^2 - a_1^2 - a_2^2}{2a_1a_2} \quad (4.3)$$

$$D = \pm \sqrt{1 - C^2} = \sin \theta_2 \quad (4.4)$$

$$\theta_2 = (D, C) \quad (4.5)$$

$$\theta_1 = (y, x) - (a_2 \sin \theta_2, a_1 + a_2 \cos \theta_2) \quad (4.6)$$

假设臂的末端应重复追踪图 4.2.2 所示的圆形工作空间路径 $p(t)$ ，其描述为

$$x(t) = 2 + \frac{1}{2} \cos t \quad (4.7)$$

$$y(t) = 1 + \frac{1}{2} \sin t \quad (4.8)$$

通过在每个时间 t 在逆运动学方程中使用这些表达式，我们得到图 4.2.3 所示的所需关节空间轨迹 $q(t) = (\theta_1(t), \theta_2(t))$ ，这些轨迹产生了末端执行器的圆形笛卡尔运动（使用 $a_1 = 2, a_2 = 2$ ）。

我们为”肘部向下”配置计算了关节变量。在 (4) 中选择相反的符号会得到”肘部向上”的关节空间轨迹，产生相同的笛卡尔轨迹。

4.2.2 多项式路径插值

假设已经为操作臂运动确定了期望轨迹，无论是在笛卡尔空间还是使用逆运动学在关节空间。为方便起见，我们使用关节空间变量 $q(t)$ 表示。将整个轨迹存储在计算机内存中是不可能的，很少有实用有用的轨迹具有简单的闭合形式表达式。因此，通常的做法是将一系列点 $q_i(t_k)$ 存储在计算机内存中，对于每个关节变量 i ，这些点代表变量在离散时间 t_k 的期望值。因此 $q(t_k)$ 是 \mathbb{R}^n 中的一个点，关节变量应在时间 t_k 通过该点。我们称这些为经过点。

大多数机器人控制方案需要连续的期望轨迹。为了将经过点表 $q_i(t_k)$ 转换为连续的期望轨迹 $q_d(t)$ ，我们可以使用许多选项。让我们在这里讨论多项式插值。

假设经过点在时间上是均匀间隔的，并将采样周期定义为

$$T = t_{k+1} - t_k \quad (4.9)$$

对于平滑运动，在每个时间间隔 $[t_k, t_{k+1}]$ 上，我们要求期望位置 $q_d(t)$ 和速度 $\dot{q}_d(t)$ 与制表的经过点匹配。这产生了边界条件

$$\begin{cases} q_d(t_k) = q(t_k), & q_d(t_{k+1}) = q(t_{k+1}) \\ \dot{q}_d(t_k) = \dot{q}(t_k), & \dot{q}_d(t_{k+1}) = \dot{q}(t_{k+1}) \end{cases} \quad (4.10)$$

为了匹配这些边界条件，有必要在每个时间间隔 $[t_k, t_{k+1}]$ 上使用三次插值多项式

$$q_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (4.11)$$

它有四个自由变量。那么

$$\dot{q}_d(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (4.12)$$

$$\ddot{q}_d(t) = 2a_2 + 6a_3 t \quad (4.13)$$

因此，加速度在每个采样周期上是线性的。

很容易求解在每个时间间隔 $[t_k, t_{k+1}]$ 上保证边界条件匹配的系数。实际上，我们看到

$$\begin{bmatrix} 1 & t_k & t_k^2 & t_k^3 \\ 0 & 1 & 2t_k & 3t_k^2 \\ 1 & t_{k+1} & t_{k+1}^2 & t_{k+1}^3 \\ 0 & 1 & 2t_{k+1} & 3t_{k+1}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q(t_k) \\ \dot{q}(t_k) \\ q(t_{k+1}) \\ \dot{q}(t_{k+1}) \end{bmatrix} \quad (4.14)$$

这被求解以获得每个时间间隔 $[t_k, t_{k+1}]$ 上所需的插值系数。

请注意，这种技术需要以表格形式存储每个采样点的期望位置和速度。一种变体使用更高阶的多项式来确保在每个采样时间 t_k 处的位置、速度和加速度连续。

虽然我们已经使用了关节变量符号 $q(t)$ ，但应该强调的是，轨迹插值也可以在笛卡尔空间中进行。

4.2.3 带抛物线混合的线性函数

使用三次插值多项式，加速度在每个采样周期上是线性的。然而，在许多实际应用中，有很好的理由坚持在每个采样周期内保持恒定加速度。例如，任何真实的机器人都有一个执行器可以提供的力矩的上限。对于线性系统（想想牛顿定律），这转化为恒定加速度的上限。因此，恒定加速度不太可能导致执行器饱和。除此之外，大多数工业机器人控制器被编程为在每个采样周期上使用恒定加速度。

恒定加速度曲线如图 4.2.4(a) 所示。相关的速度和位置曲线如图 4.2.4(b) 和 4.2.4(c) 所示。位置轨迹由三部分组成：二次或抛物线初始部分、线性中间部分和抛物线最终部分。因此，让我们讨论使用带抛物线混合的线性函数 (LFPB) 的经过点插值。

位置轨迹从抛物线切换到线性的时间被称为混合时间 t_b 。应为每个关节变量 i 指定位置 $q_{di}(t)$ 。图 4.2.4(c) 中的轨迹可以写成关节 i 的形式

$$q_{di}(t) = \begin{cases} a_{i0} + a_{i1}t + a_{i2}t^2 & 0 \leq t < t_b \\ b_{i0} + b_{i1}t & t_b \leq t < t_f - t_b \\ c_{i0} + c_{i1}t + c_{i2}t^2 & t_f - t_b \leq t \leq t_f \end{cases} \quad (4.15)$$

系数 v_i 可以被解释为关节变量 i 允许的最大速度。设计参数是 v_i 和 t_b 。

很容易求解在每个时间间隔 $[t_k, t_{k+1}]$ 上确保满足边界条件 (4.2.2) 的系数。结果是

$$\begin{cases} a_{i0} = q_{ik}, & a_{i1} = v_i/t_b, \\ a_{i2} = -v_i/(2t_b) \\ b_{i0} = q_{ik} + v_it_b/2, & b_{i1} = v_i \\ c_{i0} = q_{i,k+1} - v_it_f^2/(2t_b), & c_{i1} = v_it_f/t_b, \\ c_{i2} = -v_i/(2t_b) \end{cases} \quad (4.16)$$

4.2.4 最小时间轨迹

有一类重要的特殊 LFPB 轨迹。假设加速度受到最大值 a_M 的限制，并且希望机器人手臂以最小时间从一个位置到达另一个位置。为简单起见，假设初始和最终速度等于零。一般情况在 [Lewis 1986a] 中涵盖（参见习题）。

最小时间轨迹如图 4.2.5 所示。为了在最小时间 t_f 内将关节变量 i 从静止位置 $q_0 = q_i(t_0)$ 驱动到期望的最终静止位置 $q_f = q_i(t_f)$ ，最大加速度 a_M 应施加到切换时间 t_s ，之后时间应施加最大减速度 $-a_M$ 直到 t_f 。注意 t_s 和 t_f 都取决于 q_0 和 q_f 。我们可以写成

$$q_i(t_s) = q_0 + \frac{1}{2}a_M(t_s - t_0)^2 \quad (4.17)$$

那么速度方程产生

$$v_i(t_s) = a_M(t_s - t_0) = a_M(t_f - t_s) \quad (4.18)$$

或

$$t_s = \frac{t_f + t_0}{2} \quad (4.19)$$

也就是说，从最大加速度到最大减速度的切换发生在半时间点。现在对位置方程进行简单的操作产生

$$q_f = q_i(t_s) + v_i(t_s)(t_f - t_s) - \frac{1}{2}a_M(t_f - t_s)^2 \quad (4.20)$$

因此 (4.2.10) 产生

$$t_f = t_0 + 2\sqrt{\frac{q_f - q_0}{a_M}} \quad (4.21)$$

不幸的是，使用恒定最大加速度计算的最小时间轨迹与机器人技术没有直接相关性。这是由于实际的操作臂具有 τ_m 的力矩限制这一事实。由于机器人方程（见表 3.3.1）是非线性的，这与加速度的上限恒定没有相关性。例如，机器人手臂在其完全伸展和完全缩回位置具有不同的最大加速度。有关更多讨论，请参见 [Kahn and Roth 1971]、[Chen 1989]、[Geering 1986]、[Gourdeau and Schwartz 1989]、[Jayasuriya and Suh 1985]、[Kim and Shin 1985]、[Shin and McKay 1985]。

4.3 机器人系统的计算机仿真

在实际在臂上实现之前，在数字计算机上仿真提议的操作臂控制方案非常重要。我们在这里展示如何对机器人系统进行这样的计算机仿真。由于大多数机器人控制器实际上以数字方式实现（第 4.5 节），我们还展示了如何仿真数字机器人手臂控制器。

4.3.1 机器人动力学仿真

有许多用于仿真非线性动力学系统的软件包，包括 SIMNON [Åström and Wittenmark 1984]、MATLAB 等。为方便起见，我们在附录 B 中包含了一些对连续和数字控制相当有用的仿真程序。

所有仿真程序都要求用户编写类似的子程序。使用积分程序（如 Runge-Kutta）的时间响应仿真器都需要计算给定当前状态的状态导数。在 3.4 节中，我们展示了如何将机器人手臂方程

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (4.22)$$

以非线性状态空间形式表示

$$\dot{x} = f(x, u) \quad (4.23)$$

其中 $x(t)$ 是状态， $u(t)$ 是输入。

定义状态为 $x = [q^T \dot{q}^T]^T$ ，我们可以写出隐式形式

$$\begin{bmatrix} I & 0 \\ 0 & M(q) \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & I \\ 0 & -N(q, \dot{q}) \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} \tau + \begin{bmatrix} 0 \\ I \end{bmatrix} \tau_d \quad (4.24)$$

其中 τ 是臂控制力矩，由控制器提供， τ_d 是干扰力矩。我们说这是一种隐式形式，因为左侧的系数矩阵意味着 \ddot{q} 不是以右侧的显式形式给出的。

给定 $x(t)$ ，有必要为积分程序提供一个子程序来计算 $\dot{x}(t)$ 。一种求解 \ddot{q} 的方法是求逆 $M(q)$ 。然而，由于潜在的数值问题，这不推荐。让我们将 (4.3.3) 表示为

$$\dot{x} = g(x, u) \quad (4.25)$$

请注意，在这种情况下， $u(t)$ 是由控制力 $\tau(t)$ 和干扰 $\tau_d(t)$ 组成的向量。

附录 B 中给出了一个简单的时间响应程序 TRESP。给定一个子程序 $F(\text{time}, x, xp)$ ，它使用 (4.3.4) 根据 $x(t)$ 和 $u(t)$ 计算 \dot{x} ；它使用 Runge-Kutta 积分器来计算状态轨迹 $x(t)$ 。为了在子程序 $F(t, x, xp)$ 中求解 \dot{x} ，我们建议计算 $M(q)$ 和 $N(q, \dot{q})$ ，然后使用最小二乘技术求解

$$\ddot{q} = M^{-1}(q)[-N(q, \dot{q}) + \tau] \quad (4.26)$$

[i.e., (4.3.3) 的底部部分]，这比 $M(q)$ 的逆数值上更稳定。最小二乘方程求解器在商业上很容易获得，例如在 [IMSL]、[LINPACK] 等处。对于更简单的臂，可以解析地求逆 $M(q)$ 。

贯穿全书，我们使用各种控制方案说明臂动力学的仿真。

4.3.2 数字机器人控制器的仿真

虽然大多数机器人控制器是在连续时间中设计的，但它们在实际的机器人上以数字方式实现。也就是说，控制信号仅在离散的时间瞬间使用微处理器更新。我们在 4.5 节

中讨论数字机器人手臂控制器的实现。因此，在实际实现之前验证提议的控制器是否按预期运行是非常可取的，使用其数字化或离散形式对其进行仿真。

数字控制方案如图 4.3.1 所示。要控制的工厂或系统是连续时间系统， $K(z)$ 是动态数字控制器，其中 z 是 Z 变换变量（即， z^{-1} 代表单位时间延迟）。数字控制器 $K(z)$ 使用 DSP 中的软件代码实现。参考输入 $r(t)$ 是 $y(t)$ 应该跟随的期望轨迹， e_k 是（离散）跟踪误差。

采样器与采样周期 T 是一个模数 (A/D) 转换器，它获取软件控制器 $K(z)$ 所需的输出 $y(t)$ 的样本 $y_k = y(kT)$ 。 $y(t)$ 的定义可以根据控制方案而变化。例如，在机器人控制中， $y(t)$ 可能表示由 $q(t)$ 和 $\dot{q}(t)$ 组成的 $2n$ 向量。

保持设备是图中的一个数模 (D/A) 转换器，它将软件控制器 $K(z)$ 计算的离散控制样本 u_k 转换为工厂所需的连续时间控制 $u(t)$ 。它是一种数据重建设备。零阶保持 (ZOH) 的输入 u_k 和输出 $u(t)$ 如图 4.3.2 所示。注意 $u(kT) = u_k$ ， T 为采样周期，因此 $u(t)$ 从右侧是连续的。也就是说， $u(t)$ 在时刻 kT 更新。ZOH 通常用于控制目的，而不是其他高阶设备如一阶保持，因为大多数商用 DSP 都有内置的 ZOH。

一旦设计了控制器，重要的是在使用数字计算机实现之前对其进行仿真，以确定闭环响应是否合适。这在机器人技术中尤其正确，因为数字控制器通常是通过设计连续时间控制器找到的，然后使用 Euler 方法等近似技术进行数字化。也就是说，对于非线性系统，控制器离散化方案通常不是精确的。这会导致性能下降。为了验证控制器性能是否合适，仿真应在所有时间提供响应，包括采样之间的时间。

为了仿真数字控制器，我们可以使用图 4.3.3 所示的方案。在那里，连续工厂动力学包含在子程序 $F(t, x, xp)$ 中；它们使用 Runge-Kutta 积分器进行积分。该图假设 ZOH；因此控制输入 $u(t)$ 在每个时间 kT 更新为 u_k ，然后保持恒定直到时间 $(k+1)T$ 。注意涉及两个时间间隔；采样周期 T 和 Runge-Kutta 积分周期 $T_R \ll T$ 。 T_R 应被选为 T 的整数除数。

这种仿真技术将工厂状态 $x(t)$ 提供为时间的连续函数，即使在采样瞬间之间的值也是如此 [事实上，它以 T_R 的倍数提供 $x(t)$]。这对于在实际在工厂上实现数字控制器之前验证闭环系统的可接受的样本间行为是至关重要的。

附录 B 中的程序 TRESP 可用于实现图 4.3.3。它以模块化的方式编写，可应用于广泛的情况。我们将在随后的几个示例中说明其用于数字控制的用途。使用 SIMNON 等仿真软件非常相似。

我们在 4.5 节中讨论数字机器人手臂控制器的实现。[Lewis 1992] 中给出了关于数字控制、仿真和 DSP 控制器实现的详细讨论。

4.4 计算力矩控制

多年来提出了许多机器人控制方案。事实上，大多数都可以被视为计算力矩控制器类的特殊情况。同时，计算力矩是非线性系统反馈线性化的特殊应用，这在现代系统理论中很流行 [Hunt et al. 1983]、[Gilbert and Ha 1984]。事实上，对机器人控制方案进行分类的一种方法是将其分为“类计算力矩”或“非类计算力矩”。类计算力矩控制出现在鲁棒控制、自适应控制、学习控制等中。

在本章的剩余部分，我们探索这类机器人控制器，它包括广泛的设计。计算力矩控制使我们能够方便地推导出非常有效的机器人控制器，同时提供一个框架，将经典独立关节控制和一些现代设计技术结合在一起，并为本书的其余部分奠定基础。在本节末尾的表 4.4.1 中总结了不同类型的类计算力矩控制器。我们将看到，许多数字机器人控制器也是类计算力矩控制器（第 4.5 节）。

4.4.1 内前馈环推导

机器人手臂动力学为

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (4.27)$$

或

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau - \tau_d \quad (4.28)$$

其中关节变量 $q(t)$ 、控制力矩 τ 和干扰 $\tau_d(t)$ 。如果此方程包括电机执行器动力学（第 3.6 节），则 $\tau(t)$ 是输入电压。

假设已根据 4.2 节的讨论为臂运动选择了期望轨迹 $q_d(t)$ 。为了确保关节变量的轨迹跟踪，定义输出或跟踪误差为

$$e(t) = q_d(t) - q(t) \quad (4.29)$$

为了演示输入 $\tau(t)$ 对跟踪误差的影响，微分两次以获得

$$\ddot{e} = \ddot{q}_d - \ddot{q} \quad (4.30)$$

现在求解 (4.4.2) 中的 \ddot{q} 并将其代入最后一个方程得到

$$\ddot{e} = \ddot{q}_d - M^{-1}(q)[-N(q, \dot{q}) + \tau - \tau_d] \quad (4.31)$$

定义控制输入函数

$$u = M^{-1}(q)[-N(q, \dot{q}) + \tau] - \ddot{q}_d \quad (4.32)$$

和干扰函数

$$w = M^{-1}(q)\tau_d \quad (4.33)$$

我们可以按如下方式定义状态 $x = [e^T \dot{e}^T]^T$ 并写出跟踪误差动力学

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} w \quad (4.34)$$

这是 Brunovsky 规范形中的线性误差系统，由 n 对双积分器 $1/s^2$ 组成，每个关节一个。它由控制输入 $u(t)$ 和干扰 $w(t)$ 驱动。注意，此推导是 3.4 节中一般反馈线性化程序的特殊情况。

反馈线性化变换 (4.4.5) 可以反转以得到

$$\tau = M(q)(\ddot{q}_d - u) + N(q, \dot{q}) \quad (4.35)$$

我们称之为计算力矩控制律。这些操作的重要性如下。从 (4.4.1) 到 (4.4.8) 没有状态空间变换。因此，如果我们选择一个稳定 (4.4.8) 的控制 $u(t)$ ，使得 $e(t)$ 趋于零，那么由 $\tau(t)$ (4.4.9) 给出的非线性控制输入将导致机器人臂 (4.4.1) 中的轨迹跟随。事实上，将 (4.4.9) 代入 (4.4.2) 得到

$$M\ddot{q} + N = M(\ddot{q}_d - u) + N + \tau_d \quad (4.36)$$

或

$$\ddot{e} = u + M^{-1}\tau_d = u + w \quad (4.37)$$

这正是 (4.4.8)。

图 4.4.1 展示了计算力矩控制方案，显示了内环和外环。重要的是要注意，它由一个内非线性环和一个外控制信号 $u(t)$ 组成。我们将看到选择 $u(t)$ 的几种方法。由于 $u(t)$ 将取决于 $q(t)$ 和 $\dot{q}(t)$ ，外环将是一个反馈环。一般来说，我们可以选择一个动态补偿器 $H(s)$ ，使得

$$U(s) = H(s)E(s) \quad (4.38)$$

$H(s)$ 可以被选择用于良好的闭环行为。根据 (4.4.10)，闭环误差系统然后具有传递函数

$$T(s) = s^2I - H(s) \quad (4.39)$$

重要的是要意识到计算力矩取决于机器人动力学的反演，实际上有时被称为逆动力学控制。事实上，(4.4.9) 显示了通过用 $\ddot{q}_d - u$ 替换 \ddot{q} 来计算 $\tau(t)$ ；也就是说，通过求解机器人逆动力学问题。与系统反演相关的所有警告，包括当系统具有非最小相位零点时产生的问题，都在这里适用。（注意在线性情况下，系统零点是反演的极点。这种非最小相位概念推广到非线性系统。）幸运的是，对于我们来说，刚性臂动力学是最小相位的。

出于实现目的，有几种方法可以计算 (4.4.9)。应避免在每个采样时间进行形式矩阵乘法。在某些情况下，表达式可以解析地推导出来。计算力矩 $\tau(t)$ 的一个好方法是使用高效的 Newton-Euler 逆动力学公式 [Craig 1985]，用 $\ddot{q}_d - u$ 代替 $\ddot{q}(t)$ 。

可以使用许多方法选择外环信号 $u(t)$ ，包括鲁棒和自适应控制技术。在本章的剩余部分，我们探索 $u(t)$ 的一些选择以及计算力矩控制的一些变体。

4.4.2 PD 外环设计

选择辅助控制信号 $u(t)$ 的一种方法是作为比例加微分 (PD) 反馈

$$u = -K_v \dot{e} - K_p e \quad (4.40)$$

然后整体机器人臂输入变为

$$\tau_c = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \quad (4.41)$$

此控制器如图 4.4.6 所示, 其中 $K_i = 0$ 。闭环误差动力学为

$$\ddot{e} + K_v \dot{e} + K_p e = w \quad (4.42)$$

或以状态空间形式

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} w \quad (4.43)$$

闭环特征多项式为

$$\Delta_c(s) = |s^2 I + sK_v + K_p| \quad (4.44)$$

PD 增益选择。通常取 $n \times n$ 增益矩阵为对角矩阵, 使得

$$K_v = \text{diag}\{k_{vi}\}, \quad K_p = \text{diag}\{k_{pi}\} \quad (4.45)$$

那么

$$\Delta_c(s) = \prod_{i=1}^n (s^2 + sk_{vi} + k_{pi}) \quad (4.46)$$

只要所有的 k_{vi} 和 k_{pi} 都为正, 误差系统就是渐近稳定的。因此, 只要干扰 $w(t)$ 有界, 误差 $e(t)$ 就是有界的。与此相关, 检查 (4.4.6) 并回忆表 3.3.1, M^{-1} 是有上界的。因此, $w(t)$ 的有界性等同于 $\tau_d(t)$ 的有界性。

重要的是要注意, 虽然选择对角 PD 增益矩阵导致在外环级别解耦控制, 但它不会导致解耦的关节控制策略。这是因为内环中的乘以 $M(q)$ 和加上非线性前馈项 $N(q, \dot{q})$ 使得信号 $u(t)$ 在所有关节之间混杂。因此, 计算任何一个给定关节的控制 $\tau(t)$ 通常需要所有关节位置 $q(t)$ 和速度 $\dot{q}(t)$ 的信息。

二阶特征多项式的标准形式为

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (4.47)$$

其中 ζ 是阻尼比, ω_n 是自然频率。因此, 可以通过如下选择 PD 增益来实现误差 $e(t)$ 的每个分量中的期望性能

$$k_{pi} = \omega_{ni}^2, \quad k_{vi} = 2\zeta_i \omega_{ni} \quad (4.48)$$

其中 ζ_i, ω_{ni} 是关节误差 i 的期望阻尼比和自然频率。使末端附近的响应比靠近基座处更快可能是有用的，因为必须移动的质量更重。

机器人表现出超调是不希望的，因为如果例如期望轨迹在工件的表面终止，这可能导致冲击。因此，PD 增益通常选择为临界阻尼 $\zeta = 1$ 。在这种情况下

$$k_{pi} = \omega_{ni}^2, \quad k_{vi} = 2\omega_{ni} \quad (4.49)$$

自然频率的选择。自然频率 ω_n 控制每个误差分量中的响应速度。对于快速响应，它应该很大，并且根据性能目标进行选择。因此，在选择 ω_n 时应考虑期望轨迹。

现在讨论选择中的一些其他因素。[Paul 1981] 给出了 ω_n 选择上的一些上限。尽管大多数工业机器人的连杆是重的，但它们可能有一些柔性。假设连杆 i 的第一柔性或共振模式的频率为

$$\omega_r = \sqrt{k_r/J} \quad (4.50)$$

其中 J 是连杆惯量， k_r 是连杆刚度。那么，为了避免激发共振模式，我们应该选择 $\omega_n < \omega_r/2$ 。当然，连杆惯量 J 随臂构型变化，因此其最大值可用于计算 ω_r 。

ω_n 上的另一个上限由执行器饱和考虑提供。如果 PD 增益太大，力矩 $\tau(t)$ 可能达到其上限。

如下从误差有界性考虑提供了 PD 增益选择的一些更多感觉。(4.4.15) 中闭环误差系统的传递函数为

$$H(s) = (s^2 I + sK_v + K_p)^{-1} \quad (4.51)$$

或如果 K_v 和 K_p 是对角的

$$H(s) = \text{diag} \left\{ \frac{1}{s^2 + sk_{vi} + k_{pi}} \right\} \quad (4.52)$$

$$w = M^{-1}\tau_d + \ddot{q}_d \quad (4.53)$$

我们假设干扰和 M^{-1} 是有界的（表 3.3.1），使得

$$\|w\|_2 \leq \|M^{-1}\|_2 \|\tau_d\|_2 + \|\ddot{q}_d\|_2 \leq \bar{m} \|\tau_d\|_2 + \|\ddot{q}_d\|_2 \leq d \quad (4.54)$$

其中对于给定的机器人臂， m 和 d 是已知的。因此

$$\|e\|_2 \leq \|H(s)\|_2 \|w\|_2 \leq \|H(s)\|_2 d \quad (4.55)$$

$$\|\dot{e}\|_2 \leq \|sH(s)\|_2 d \quad (4.56)$$

现在选择 L_2 范数，算子增益 $\|H(s)\|_2$ 是 $H(s)$ 的 Bode 幅度图的最大值。对于临界阻尼系统

$$|H_i(j\omega)| = \frac{1}{\sqrt{(\omega^2 - \omega_{ni}^2)^2 + (2\zeta_i \omega_{ni} \omega)^2}} \quad (4.57)$$

因此

$$\|H(s)\|_2 = \max_i \frac{1}{2\zeta_i \omega_{ni}^2} \quad (4.58)$$

此外 (参见习题)

$$\|sH(s)\|_2 = \max_i \frac{1}{2\zeta_i \omega_{ni}} \quad (4.59)$$

所以

$$\|e\|_2 \leq \frac{d}{\omega_{ni}^2}, \quad \|\dot{e}\|_2 \leq \frac{d}{\omega_{ni}} \quad (4.60)$$

因此, 在临界阻尼的情况下, 位置误差随 k_{pi} 减小, 速度误差随 k_{vi} 减小。

示例 4.4-1: PD 计算力矩控制的仿真

a. 计算力矩控制律

在示例 3.2.2 中, 我们找到了图 4.2.1 所示的两连杆平面肘臂的动力学

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2c_2 & m_2a_2^2 + m_2a_1a_2c_2 \\ m_2a_2^2 + m_2a_1a_2c_2 & m_2a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -m_2a_1a_2s_2\dot{\theta}_2^2 - 2m_2a_1a_2s_2\dot{\theta}_1\dot{\theta}_2 \\ m_2a_1a_2s_2\dot{\theta}_1^2 \end{bmatrix} \quad (4.61)$$

这些在标准形式中

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (4.62)$$

取连杆质量为 1 kg, 长度为 1 m。

PD 计算力矩控制律给出为

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + N(q, \dot{q}) \quad (4.63)$$

其中跟踪误差定义为

$$e = q_d - q \quad (4.64)$$

b. 期望轨迹

让期望轨迹 $q_d(t)$ 的分量为

$$\theta_{1d} = g_1 \sin(2\pi t/T) \quad (4.65)$$

$$\theta_{2d} = g_2 \cos(2\pi t/T) \quad (4.66)$$

周期 $T = 2$ s, 幅值 $g_i = 0.1$ rad ≈ 6 deg。为了良好的跟踪, 选择闭环系统的时间常数为 0.1 s。对于临界阻尼, 这意味着 $K_v = \text{diag}\{k_v\}$, $K_p = \text{diag}\{k_p\}$, 其中

$$k_p = \omega_n^2 = 100, \quad k_v = 2\omega_n = 20 \quad (4.67)$$

重要的是要意识到控制器参数 (如 PD 增益) 的选择取决于性能目标——在这种情况下, 是期望轨迹的周期。

c. 计算机仿真

让我们使用附录 B 中的程序 TRESP 仿真计算力矩控制器。使用 MATLAB 和 SIMNON 等商业包进行仿真非常相似。

TRESP 所需的子程序如图 4.4.2 所示。它们是

4.4.3 PID 外环设计

为了提高性能，可以使用 PID 外环控制

$$u = -K_v \dot{e} - K_p e - K_i \int e t \quad (4.68)$$

然后整体机器人臂输入变为

$$\tau_c = M(q) \left(\ddot{q}_d + K_v \dot{e} + K_p e + K_i \int e t \right) + N(q, \dot{q}) \quad (4.69)$$

此控制器如图 4.4.6 所示。闭环误差动力学为

$$\ddot{e} + K_v \dot{e} + K_p e + K_i \int e t = w \quad (4.70)$$

或以状态空间形式，通过定义

$$\dot{\varepsilon} = e \quad (4.71)$$

我们有

$$\frac{d}{dt} \begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_v \end{bmatrix} \begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} w \quad (4.72)$$

积分项的作用是消除稳态误差。然而，积分项可能导致执行器饱和引起的积分器饱和问题。

4.4.4 计算力矩类控制器

有许多计算力矩控制器的变体。在本小节中，我们总结了一些重要的变体。

考虑近似计算力矩控制器

$$\tau_c = \hat{M}(\ddot{q}_d - u) + \hat{N} \quad (4.73)$$

其中 \hat{M} 和 \hat{N} 是 M 和 N 的近似值，可能在实现中使用以节省计算量。然后闭环系统变为

$$(I - \Delta)\ddot{e} + (I - \Delta)(K_v \dot{e} + K_p e) + \delta = w \quad (4.74)$$

其中

$$\Delta = I - \hat{M}^{-1}M, \quad \delta = \hat{M}^{-1}(N - \hat{N}) \quad (4.75)$$

和干扰为

$$w = \hat{M}^{-1}\tau_d \quad (4.76)$$

如果使用精确计算力矩控制使得 $\Delta = 0, \delta = 0$ ，这会简化为误差系统 (4.4.10)。否则，误差系统由期望加速度和非线性项不匹配 $\delta(t)$ 驱动。因此，跟踪误差永远不会精确为零。此外，辅助控制 $u(t)$ 乘以 $(I - \Delta)$ ，这可能导致一个非常困难的控制问题。

使用外环 PD 反馈，使得 $u(t) = -K_v\dot{e} - K_p e$ ，得到误差系统

$$\ddot{e} + (I - \Delta)K_v\dot{e} + (I - \Delta)K_p e + d = w \quad (4.77)$$

其中 $d(t) = \delta + \Delta\ddot{q}_d$ 。即使 K_v 和 K_p 被选择为左侧具有良好的稳定性，这种系统的行为也不是显而易见的。存在两种问题：首先是干扰项 $d(t)$ ，其次是误差及其导数的函数 $\Delta(K_v\dot{e} + K_p e)$ 。

4.4.5 PD-Plus-Gravity 控制器

计算力矩族中一个有用的控制器是 PD-plus-gravity 控制器，当 $\hat{M} = I, \hat{N} = G(q) - \ddot{q}_d$ 时产生，其中 $G(q)$ 是操作臂动力学的重力项。然后，为 $u(t)$ 选择 PD 反馈产生

$$\tau_c = -K_v\dot{e} - K_p e + G(q) \quad (4.78)$$

此控制律在 [Arimoto and Miyazaki 1984]、[Schilling 1990] 中进行了处理。它比精确计算力矩控制器实现起来简单得多。

当臂静止时，动力学 (4.4.1) 中唯一的非零项是重力 $G(q)$ 、干扰 τ_d 和控制力矩 τ 。PD-gravity 控制器 τ_c 包含 $G(q)$ ，因此我们应该期望对设定点跟踪有良好的性能，即当给定恒定的 q_d 使得 $\ddot{q}_d = 0$ 时。下一个结果将这一点形式化。它依赖于 Lyapunov 证明 (第 1 章)，这种证明在本书中将始终有用，特别是利用表 3.3.1 中的斜对称性质。因此，理解此证明中的步骤非常重要。

定理 4.1. 假设在臂动力学 (4.4.1) 中使用 PD-gravity 控制且 $\tau_d = 0, \ddot{q}_d = 0$ 。则稳态跟踪误差 $e = q_d - q$ 为零。

证明. 1. 闭环系统

忽略摩擦，机器人动力学由下式给出

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) = \tau \quad (4.79)$$

当 $\ddot{q}_d = 0$ 时，提出的控制律 (4.4.49) 产生闭环动力学

$$M\ddot{q} + V_m\dot{q} + K_v\dot{q} + K_p e = 0 \quad (4.80)$$

2. Lyapunov 函数

现在选择 Lyapunov 函数

$$V = \frac{1}{2} \dot{q}^T M \dot{q} + \frac{1}{2} e^T K_p e \quad (4.81)$$

并微分得到

$$\dot{V} = \frac{1}{2} \dot{q}^T \dot{M} \dot{q} + \dot{q}^T M \ddot{q} + \dot{e}^T K_p e \quad (4.82)$$

代入闭环动力学 (2) 产生

$$\dot{V} = \frac{1}{2} \dot{q}^T (\dot{M} - 2V_m) \dot{q} - \dot{q}^T K_v \dot{q} \quad (4.83)$$

现在, 第一项的斜对称性给出

$$\dot{V} = -\dot{q}^T K_v \dot{q} \leq 0 \quad (4.84)$$

状态为 $x = [e^T \dot{q}^T]^T$, 因此 V 是正定的但只是负半定的。因此, 我们展示了 Lyapunov 意义上的稳定性, 即误差和关节速度都是有界的。

3. LaSalle 扩展的渐近稳定性

可以使用 Barbalat 引理和 LaSalle 扩展的变体 [Slotine and Li 1991] (第 2 章) 来演示系统的渐近稳定性。因此, 有必要演示包含在集合 $\dot{V} = 0$ 中的唯一不变集是原点。

由于 V 的下界为零且非正, 因此 V 趋近于一个有限极限, 可以写成

$$V(\infty) - V(0) = \int_0^\infty \dot{V} t < \infty \quad (4.85)$$

我们现在调用 Barbalat 引理来表明 \dot{V} 趋于零。为此, 有必要演示 \dot{V} 的一致连续性。我们看到

$$\ddot{V} = -2\dot{q}^T K_v \ddot{q} \quad (4.86)$$

所展示的稳定性表明 \dot{q} 和 q 是有界的, 因此 (2) 和 $M^{-1}(q)$ 的有界性 (见表 3.3.1) 揭示 \ddot{q} 是有界的。因此, \ddot{V} 是有界的, 因此 \dot{V} 是一致连续的。这保证了根据 Barbalat 引理, \dot{V} 趋于零。

现在很明显 \dot{q} 趋于零。还需要表明跟踪误差 $e(t)$ 消失。注意当 $\dot{q} = 0$ 时, (2) 揭示

$$K_p e = 0 \quad (4.87)$$

因此, 非零的 $e(t)$ 导致非零的 \ddot{q} , 因此 $\dot{q} \neq 0$, 这是一个矛盾。因此, 包含在 $\{x(t) | \dot{V} = 0\}$ 中的唯一不变集是 $x(t) = 0$ 。这最终表明 $e(t)$ 和 \dot{q} 都消失, 并结束了证明。□

示例 4.4-3: PD-Gravity 控制器的仿真

在示例 4.4.1 中，我们在双连杆平面操作臂上仿真了精确的计算力矩控制律。在这里，让我们仿真 PD-gravity 控制器。我们将采用相同的臂参数和期望轨迹。

假设每个连杆的 PD 增益相同，双连杆臂的 PD-gravity 控制律为

$$\tau_c = -K_v \dot{e} - K_p e + G(q) \quad (4.88)$$

这很容易通过注释掉几行代码并在图 4.4.2 的子程序 CTL(x) 中进行一些其他修改来仿真。

对于临界阻尼，PD 增益选择为

$$k_p = \omega_n^2, \quad k_v = 2\omega_n \quad (4.89)$$

ω_n 几个值的 PD-gravity 控制器仿真结果如图 4.4.9 所示。随着 ω_n 的增加，因此 PD 增益也增加，跟踪性能提高。然而，无论 PD 增益多大，跟踪误差永远不会精确为零，而是以零为中心的一个球内有界，其半径随着增益变大而减小。 $\omega_n = 50$ 的性能，对应于 $k_p = 2500, k_v = 100$ ，对于许多应用来说会相当合适。

非常重要的是要注意误差的直流值等于零。可以考虑将重力项作为计算力矩控制律的“直流部分”。如果将它们包括在计算力矩中，将没有误差偏移。

相关的控制输入力矩如图 4.4.10 所示。非常有趣的是，力矩对于更高的增益更小。这与流行的观点相反，后者假设控制力矩总是随着 PD 增益的增加而增加。这是由于更大的增益提供更好的性能，因此跟踪误差更小这一事实。

鉴于图 4.4.9(a) 中的误差具有不同的幅值，将内连杆的 PD 增益比外连杆更大可能更合理。

4.4.6 经典关节控制

通过选择 (4.4.43) 中的

$$\hat{M} = I, \quad \hat{N} = 0 \quad (4.90)$$

可以获得在实践中经常给出良好结果的简单控制器。

然后有结果

$$\tau_c = -K_v \dot{e} - K_p e \quad (4.91)$$

通过选择仅依赖于关节变量 i 的 u_i ，这描述了 n 个解耦的独立关节控制器，被称为独立关节控制。在实际机器人手臂上的实现很容易，因为关节 i 的控制输入仅依赖于局部测量的变量，而不依赖于其他关节的变量。此外，计算简单，不涉及求解复杂的非线性机器人逆动力学。

在机器人技术的早期，独立关节控制很受欢迎 [Paul 1981]，因为它允许使用单输入/单输出 (SISO) 经典技术对闭环系统进行解耦分析。它也被称为经典关节控制。即使在今天，也有一些研究者强烈主张这种控制方案始终适合实际实现，并且现代非线性控制方案对于工业机器人应用来说过于复杂。

现在进行独立关节控制的传统分析。它提供了与经典控制概念的联系，对于机器人控制设计人员理解很重要。参见 [Franklin et al. 1986] 了解经典控制理论参考。

具有电动执行器的机器人臂的简化动力学模型可以写成 (第 3.6 节)

$$\left(J_{mi} + \frac{m_{ii}}{r_i^2} \right) \ddot{\theta}_i + \left(B_{mi} + \frac{k_{bmi} k_{mi}}{R_i} \right) \dot{\theta}_i + \frac{d_i}{r_i} = \frac{k_{mi}}{R_i} v_i \quad (4.92)$$

其中 J_m 是执行器电机惯量， B_m 是转子阻尼常数， k_m 是力矩常数， k_b 是反电动势常数， R 是电枢电阻， r_i 是关节 i 的齿轮比。电机角度表示为 $\theta_i(t)$ 。 $M(q)$ 的对角元素的常数部分表示为 m_{ii} 。这些元素的时间变化部分，以及 $M(q)$ 的非对角元素、非线性项 $N(q, \dot{q})$ 和任何干扰 τ_d 都集中在一起成为干扰 $d_i(t)$ 。因此 $d_i(t)$ 包含所有其他关节对关节 i 的影响。控制输入是电机电枢电压 $v_i(t)$ 。

请注意，在此方程中主要出现电机参数。事实上，如果齿轮比很小，甚至可以忽略 m_{ii} 。由于这个原因，如果齿轮比很小，机器人臂控制问题实际上就简化为控制执行器电机的问题。

让我们将操作臂关节 i 的这种简化线性时不变模型表示为

$$J\ddot{\theta}_i + B\dot{\theta}_i + rd_i = kv_i \quad (4.93)$$

其中常数 k_m/R 已被并入 J 和 B 的定义中。

根据表 3.3.1 中的性质，干扰 $d(t)$ 是有界的，尽管通常不是由常数界定。界限可以是 $q(t)$, $\dot{q}(t)$ 甚至 $\ddot{q}(t)$ 的函数。这通常在经典分析中被忽略。然而， $d(t)$ 的影响通过乘以齿轮比 r 而有所减轻。在经典关节控制设计中忽略了关节柔性 (第 6 章) 的影响。[为了与经典符号保持一致，本节中 $u(t)$ 的定义与我们以前的用法相比有一个符号变化。]

现在，让我们考虑控制输入 $u(t)$ 的一些选择。

PD 控制。 选择 PD 控制律

$$u = k_v \dot{e} + k_p e \quad (4.94)$$

其中 $e(t) = \theta_{di}(t) - \theta_i(t)$ 是电机角度 i 的跟踪误差，得到图 4.4.11 所示的闭环系统。回想一下，电机角度与关节变量通过 $q_i = r_i \theta_i$ 相关联。因此可以从 $q_{di}(t)$ 计算 $\theta_d(t)$ 。这个 PD 控制器很容易在实际的臂上实现，只需要很少的计算机能力。

使用经典控制中的 Mason 定理，发现设定点跟踪的闭环传递函数为 (参见习题)

$$\frac{\Theta_i}{\Theta_{di}} = \frac{k_p s + k_p}{J s^3 + (B + k_v) s^2 + k_v s + k_p} \quad (4.95)$$

具有闭环特征多项式

$$\Delta_c(s) = Js^3 + (B + k_v)s^2 + k_v s + k_p \quad (4.96)$$

可以选择 PD 增益以获得合适的自然频率和阻尼比，正如我们所见。

在稳态时，对干扰 $d(t)$ 的唯一非零贡献是被忽略的重力 $G(q)$ 。表 3.3.1 显示，对于给定的机器人臂，重力向量由已知值 g_b 界定。因此，在稳态时，使用 PD 控制的关节 i 的稳态跟踪误差由

$$|e_i| \leq \frac{r_i g_b}{k_p} \quad (4.97)$$

界定。因此，对于指定最终值 q_d 的设定跟踪，具有大 k_p 的 PD 控制可能是合适的。

事实上，下一个结果表明，即使对于跟随期望轨迹，而不仅仅是设定跟踪，PD 控制通常也非常合适。它在 [Dawson 1990] 中被证明。

定理 4.2. 如果将 PD 控制律 (4.4.54) 应用于每个关节且 $e(0) = 0, \dot{e}(0) = 0$ ，则位置和速度跟踪误差在球内有界，其半径随着 $k_v \rightarrow \infty$ 大致以 $1/\sqrt{k_v}$ 减小。

这个结果证实了那些坚持认为 PD 控制通常足够用于实际应用的人的观点。

当然，问题是 k_v 不能在达到执行器力矩限制的情况下无限增加。本书中将讨论的其他方案允许良好的轨迹跟随而不需要如此大的力矩。

在 [Paul 1981] 中讨论了修改 PD 控制律以获得更好性能的几种方法。这些包括重力补偿 [这恰好产生控制器 (4.4.49)] 和加速度前馈，这相当于使用

$$u = k_v \dot{e} + k_p e + \ddot{\theta}_d \quad (4.98)$$

对于每个关节。还提到了关节耦合控制，这相当于在 $u(t)$ 中重新添加 $M(q)$ 和 $N(q, \dot{q})$ 中描述关节之间相互作用的一些被忽略的项。因此，这种修正涉及在近似计算力矩控制 (4.4.43) 中使用更好的 \hat{M} 和 \hat{N} 估计。

PID 控制。 已经看到 PD 独立关节控制通常适合跟踪控制。然而，在稳态时存在由于重力引起的残余误差 (4.4.57)。这可以使用 PID 独立关节控制律消除

$$u = k_v \dot{e} + k_p e + k_i \int e t \quad (4.99)$$

对于每个关节。

现在设定跟踪 $\ddot{\theta}_d = 0$ 的传递函数为

$$\frac{\Theta_i}{\Theta_{di}} = \frac{k_i s + k_p s + k_i}{Js^3 + (B + k_v)s^2 + k_p s + k_i} \quad (4.100)$$

具有闭环特征多项式

$$\Delta_c(s) = Js^3 + (B + k_v)s^2 + k_p s + k_i \quad (4.101)$$

现在终值定理显示，对于设定跟踪，稳态误差为零。Routh 检验表明，对于稳定性，需要

$$k_i < \frac{(B + k_v)k_p}{J} \quad (4.102)$$

在控制律中使用积分项时，重要的是要注意由于执行器饱和限制引起的积分器饱和的可能性，这在本节前面的“PID 外环设计”中进行了讨论。

示例 4.4-4: 经典关节控制和力矩饱和限制

在示例 4.4.1 中，我们为双连杆平面肘臂仿真了精确的计算力矩控制律。在这里，我们希望在同一臂上显示使用 PD 和 PID 经典关节控制的结果（具有相同的期望轨迹）。我们还对演示力矩饱和限制的影响感兴趣。为了突出效果而没有无关的细节，我们将只使用臂动力学，没有执行器动力学或齿轮比。

这些可以很容易地通过在图 4.4.2 中的子程序 `arm(x, xp)` 中进行一些轻微的修改来包括。

a. PD 独立关节控制

对于双连杆臂，PD 独立关节控制就是

$$\tau_c = -K_v \dot{e} - K_p e \quad (4.103)$$

其中跟踪误差 $e(t) = q_d(t) - q(t)$ 。这个控制律非常简单直接，甚至不需要使用 DSP 就可以在实际的臂上实现。它比示例 4.4.1 中的控制简单得多。

在两连杆臂上使用 PD 控制器的结果如图 4.4.13 所示，对应于几个闭环自然频率 ω_n 的值。回想一下，临界阻尼的 PD 增益为

$$k_p = \omega_n^2, \quad k_v = 2\omega_n \quad (4.104)$$

图 (a) 部分中的低增益结果 ($k_p = 100, k_v = 20$) 非常糟糕（注意刻度）。然而，(c) 部分中的高增益结果要好得多。更高的增益将导致进一步的跟踪误差减小。

重要的是要注意，在所有情况下，跟踪误差都有一个直流分量。这是由于重力项被忽略的事实，应该与示例 4.4.3 的结果形成对比。

因此在控制律中加入重力项显著提高了跟踪性能。

b. PID 独立关节控制

双连杆臂的 PID 独立关节控制为

$$\tau_c = -K_v \dot{e} - K_p e - K_i \int e t \quad (4.105)$$

这个简单的定律很容易通过向图 4.4.2 中的子程序 `arm(x, xp)` 添加两个状态 $x(5)$ 和 $x(6)$ 来实现，以考虑积分器（参见示例 4.4.2）。

图 4.4.13(c) 中的仿真被重复，但现在添加 $k_i = 1000$ 的积分增益。结果如图 4.4.15 所示。在正弦运动的几次迭代之后，跟踪误差的直流值变为零，因此性能大大提高。也就是说，添加积分项可以补偿控制律中忽略的重力项。然而，误差收敛到零直流值需要一段时间，结果仍然不如示例 4.4.3 中使用相同 PD 增益的 PD-gravity 控制器。另一方面，积分项可以拒绝除重力之外的其他项（例如，某些类型的摩擦）。

c. 执行器饱和限制

我们讨论了本例 (b) 部分前面“PID 外环设计”中由于执行器饱和限制引起的积分器饱和。在这里，我们想演示其有害影响。

本例 (b) 部分中的控制力矩类似于图 4.4.14(c) 中的力矩；除了时间零附近的一些疯狂动作外，它表现得相当好，最大正偏移为 78 N-m。

假设现在有力矩限制 $\tau_{max} = 35$ N-m, $\tau_{min} = -35$ N-m。这很容易通过在图 4.4.2 中的子程序 CTL(x) 中添加几行代码来仿真

$$\text{if } \tau(1) > 35 \text{ then } \tau(1) = 35 \quad (4.106)$$

等等。

这些限制的仿真结果如图 4.4.16 所示，非常糟糕。力矩如图 4.4.17 所示。

在 4.5 节中，我们展示了如何使用数字控制器中的抗饱和保护来改善执行器饱和的影响。PD 控制的仿真结果与 PID 控制的结果相比不那么糟糕，因为如果没有积分器存在，饱和限制的影响较小。

4.5 数字机器人控制

许多机器人控制方案是复杂的，涉及评估非线性项的大量计算。因此，它们在数字信号处理器 (DSP) 上作为数字控制律实现。某些控制方案可以通过使用简单的近似来以离散形式实现。

4.5.1 基于近似连续控制器的设计

机器人控制器设计通常按以下方式进行。首先，在连续时间中设计控制律以获得良好的性能。然后，使用某种近似技术将连续控制器数字化以在 DSP 上实现。

考虑连续时间控制器

$$u_c(t) = -K_v \dot{e} - K_p e - K_i \int e t \quad (4.107)$$

这可以以传递函数形式表示为

$$U_c(s) = -K(s)E(s) \quad (4.108)$$

其中

$$K(s) = K_v s + K_p + \frac{K_i}{s} \quad (4.109)$$

为了获得数字控制器，我们可以使用 Euler 近似 $s \approx (z - 1)/T$ ，其中 T 是采样周期。这产生

$$K(z) = K_v \frac{z - 1}{T} + K_p + \frac{K_i T}{z - 1} \quad (4.110)$$

然而，更常用的近似是双线性变换 (BLT)

$$s \approx \frac{2}{T} \frac{z-1}{z+1} \quad (4.111)$$

这对应于使用梯形规则近似积分。在此映射下，稳定连续系统的极点被映射到稳定离散系统的极点。

4.5.2 执行器饱和和积分器抗饱和补偿

执行器饱和导致的积分器饱和是机器人控制器外环 PID 回路中可能发生的问题。在示例 4.4.4 中，我们看到了积分器饱和的有害影响。使用数字实现可以很容易地实现抗饱和保护 [Åström and Wittenmark 1984]、[Lewis 1992]。

假设控制器以传递函数形式给出

$$R(z^{-1})v_k = T(z^{-1})r_k - S(z^{-1})w_k \quad (4.112)$$

其中 z^{-1} 在时间域中被解释为单位延迟 T 秒。

通过向两边添加 $A_0(z^{-1})u_k$ 可以获得具有抗饱和补偿的调节器

$$A_0(z^{-1})v_k = T(z^{-1})r_k - S(z^{-1})w_k \quad (4.113)$$

$$u_k = \text{sat}(v_k) \quad (4.114)$$

其中 $\text{sat}(\cdot)$ 是饱和函数。

当 $A_0(z)$ 的所有 n 个极点都在原点时，会发生特殊情况。然后调节器显示死区行为；在 n 个时间步之后，其状态保持限制在依赖于 w_k 和 u_k 的值的容易计算的值（参见习题）。

示例 4.5-1：数字机器人计算力矩控制器的仿真

在示例 4.4.1 中，我们仿真了双连杆平面肘臂的连续时间 PD 计算力矩 (CT) 控制器。在此示例中，演示了该控制器离散化的一些问题。有两个目标。首先，我们展示了采样对 CT 控制器的影响。然后设计了一个仅使用编码器关节位置测量的实际数字控制器，重建计算控制律所需的速度。

a. 数字 CT 控制器的采样周期影响

在示例 4.4.1 中，给出了 TRESP (附录 B) 所需的子程序，以仿真连续时间 PD CT 控制器。为了仿真 PD CT 控制器的数字化版本，有必要将控制器子程序 [在该示例中称为 CTL(x)] 从连续动力学中移除并将其放入子程序 DIG(IK, T, x) 中，该子程序由 TRESP 调用。这与图 4.5.1 和 4.3 节中给出的数字控制器仿真技术保持一致。

图 4.5.4 中显示了 TRESP 所需的 DIG(IK, TD, x) 和 F(t, x, xp) 子程序。用于轨迹生成的子程序 SYSINP(IT, x, t) 和包含臂动力学的 ARM(x, xp) 与示例 4.4.1

中的相同。

现在使用不同的采样周期 T 运行程序 TRESP 得到图 4.5.5 所示的跟踪误差图。所有图的 Runge-Kutta 积分周期为 5 ms。图表显示，对于 $T = 5$ ms，误差非常小，实际上与示例 4.4.1 中的误差非常相似。然而，随着采样周期 T 的增加，跟踪性能恶化。

关节 2 的力矩输入如图 4.5.6 所示。对于小 T ，它实际上与示例 4.4.1 中的连续 CT 控制器相同。当 $T = 0.1$ s 时，会发生非常有趣的现象。根据图 4.5.6(c) 中的图，存在极限环，一种非线性振荡形式。众所周知，采样可以在闭环系统中引起这种非线性效应 [Lewis 1992]。根据图 4.5.5(d)，极限环在跟踪误差中反映为关于 $e(t) = 0$ 的周期性振荡。极限环的出现与采样系统中可观测性的丧失密切相关。

b. 仅使用位置测量的数字控制器

在部分 a 中，我们假设关节位置和速度都可用作测量。在实际情况下，只有关节位置可用，通常来自光学编码器测量。因此，在这里，我们希望设计一个现实的数字 CT 控制器，重建速度。

图 4.5.7 中的子程序仅使用位置测量，使用导数滤波器 (4.5.7) 估计关节速度。产生的跟踪误差如图 4.5.8 所示。性能与部分 a 中使用速度测量的控制器相当，除了更大的初始误差瞬变。滤波器极点的值 v 取为 0.1。

一些实现细节值得注意。首先，数字控制器的初始化方式非常重要。请注意在第一次迭代 (IK=0) 中将速度估计置零的代码行。删除这些行是一个很好的练习 (参见习题)。

其次，可能会认为找到速度误差 $\dot{e}_k = \dot{e}(kT)$ 的合理过程是找到关节速度 \dot{q}_k 的估计 v_k ，然后使用

$$\dot{e}_k = \dot{q}_{dk} - v_k \quad (4.115)$$

这两个缺点。首先，它需要在内存中存储期望速度和期望轨迹。其次，它不起作用。

相反，有必要使用两个导数滤波器，一个用于估计关节速度，一个用于提供速度误差 \dot{e}_k 的估计。尝试使用 $\dot{q}_{dk} - v_k$ 的仿真很好地说明了这一点。

请注意，速度估计也用于代替 \dot{q} 计算非线性项中的值。

4.6 最优外环设计

在 4.4 节中，我们讨论了计算力矩控制，展示了如何使用涉及逆操作臂动力学的精确技术以及通过各种近似方法选择内控制环。我们还讨论了设计外线性反馈（跟踪）环的几种方案。本节讨论的结果总结在表 4.4.1 中。在本节中，我们打算介绍用于选择外反馈环的现代控制最优技术。现代最优设计在存在干扰和未建模动力学的环境下产生改

进的鲁棒性。

4.6.1 线性二次最优控制

首先，有必要回顾现代线性二次 (LQ) 设计。假设我们给出了状态空间形式的线性时不变系统

$$\dot{x} = Ax + Bu \quad (4.116)$$

具有 n 维状态 $x(t)$ 和 m 维输入 $u(t)$ 。希望计算状态反馈增益 K

$$u = -Kx \quad (4.117)$$

使得闭环系统

$$\dot{x} = (A - BK)x \quad (4.118)$$

是渐近稳定的。此外，我们不想使用太多控制能量来稳定系统，因为在许多现代系统（例如汽车、航天器）中，燃料或能量是有限的。

这是一个复杂的多变量设计问题，因为反馈增益矩阵 K 的维度为 $m \times n$ 。经典控制方法可能涉及，例如，执行 mn 根轨迹设计以一次关闭一个反馈环。另一方面，现代控制技术通过求解一些标准矩阵设计方程可以找到保证稳定性的解决方案。这种方法同时关闭所有 nm 反馈环并保证良好的增益和相位裕度。

使用现代控制理论找到反馈矩阵如下。首先，定义形式为的二次性能指标 (PI)

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (4.119)$$

其中 Q 是对称半正定 $n \times n$ 矩阵（表示为 $Q \geq 0$ ）， R 是对称正定 $m \times m$ 矩阵（ $R > 0$ ）。也就是说， R 的所有特征值都大于零， Q 的所有特征值都大于或等于零。 Q 被称为状态加权矩阵， R 是控制加权矩阵。这些矩阵是设计参数，由工程师选择，例如，取决于闭环时间响应的期望形式。

最优 LQ 反馈增益 K 是使 PIJ 最小化的增益。动机如下。PI 中的二次项 $x^T Q x$ 和 $u^T R u$ 是广义能量函数（例如，电容器中的能量为 $\frac{1}{2} C v^2$ ，运动的动能为 $\frac{1}{2} m v^2$ ）。假设在闭环系统 (4.6.3) 中最小化 J 。这意味着 $x^T(t) Q x(t) + u^T(t) R u(t)$ 的无限积分是有限的，因此这个时间的函数随着 t 变大而趋于零。然而

$$x^T Q x = \|\sqrt{Q}x\|^2, \quad u^T R u = \|\sqrt{R}u\|^2 \quad (4.120)$$

其中矩阵的平方根定义为 $\sqrt{Q}\sqrt{Q} = Q$ 。由于这些范数随着 t 消失且 $|R| \neq 0$ ，函数 $\|\sqrt{Q}x\|$ 和 $u(t)$ 都趋于零。在假设 \sqrt{Q}, A 是可观测的 [Kailath 1980] 的情况下，如果 $y(t)$ 趋于零， $x(t)$ 也趋于零。

因此，最优增益 K 保证在闭环系统 (4.6.3) 中所有信号都随时间趋于零。也就是说， K 稳定 $(A - BK)$ 。

在现代控制理论中, 确定最优 K 很容易, 是标准结果 (参见, 例如 [Lewis 1986a]、[Lewis 1986b])。只需通过求解矩阵设计方程就可以找到最优反馈增益

$$K = R^{-1}B^T P \quad (4.121)$$

$$0 = A^T P + PA - PBR^{-1}B^T P + Q \quad (4.122)$$

其中 P 是依赖于最优增益的辅助设计对称 $n \times n$ 矩阵。第二个是被称为 Riccati 方程的非线性矩阵二次方程; 使用 MATLAB、[MATRIXx 1989] 等软件设计包中的标准例程, 很容易求解该方程以获得辅助矩阵 P 。

下一个结果在现代控制理论中非常重要, 形式化了刚才给出的稳定性讨论。

定理 4.3. 设 \sqrt{Q} , A 可观测且 (A, B) 可控。则:

[label=()]存在 Riccati 方程的唯一正定解 P 。闭环系统 $(A - BK)$ 是渐近稳定的。闭环系统具有无限增益裕度和 60° 相位裕度。

可控性在第 1 章中讨论过。可观测性大致意味着所有系统模式在 PI 中都有独立的影响, 因此如果 J 有界, 所有模式都独立地随 t 趋于零。验证这些属性很容易。如果可控性矩阵

$$U = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \quad (4.123)$$

具有满秩 n , 则系统是可控的。如果可观测性矩阵

$$V = \begin{bmatrix} \sqrt{Q} \\ \sqrt{Q}A \\ \vdots \\ \sqrt{Q}A^{n-1} \end{bmatrix} \quad (4.124)$$

具有满秩 n , 则系统 (A, C) 是可观测的。MATLAB, 例如, 为这些测试提供例程。因此, 可以选择状态加权矩阵 Q 以满足可观测性要求。

该定理使这种现代设计方法非常强大。无论输入和状态的数量如何, 只要假设成立, 就可以找到稳定系统的反馈增益。

4.6.2 线性二次计算力矩设计

现在我们将这些结果应用于机器人操作臂动力学的控制

$$\ddot{e} = u + w \quad (4.125)$$

根据 4.4 节, 计算力矩控制律

$$\tau_c = M(q)(\ddot{q}_d - u) + N(q, \dot{q}) \quad (4.126)$$

产生误差系统

$$\ddot{e} = u + M^{-1}\tau_d \quad (4.127)$$

我们可以将其写成

$$\dot{x} = Ax + Bu \quad (4.128)$$

状态定义为

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (4.129)$$

现在, 选择外环 PD 反馈

$$u = -Kx = -\begin{bmatrix} K_p & K_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = -K_p e - K_v \dot{e} \quad (4.130)$$

为了找到稳定增益 K , 选择 PI 中的设计参数 Q 为

$$Q = \begin{bmatrix} Q_p & 0 \\ 0 & Q_v \end{bmatrix} \quad (4.131)$$

使得位置和速度误差被独立加权。然后, 由于代表 n 个解耦牛顿定律 (即双积分器) 系统的简单 A 和 B 矩阵形式, Riccati 方程的解很容易找到 (参见习题)。使用该解在 (4.6.5) 中产生最优稳定增益的公式

$$K_p = R^{-1/2} \sqrt{Q_p}, \quad K_v = \sqrt{2K_p} \quad (4.132)$$

此 LQ 方法揭示了 PD 增益与某些设计参数 Q 和 R 之间的关系, 这些参数决定了闭环系统中的总能量。特别要注意的是, 闭环系统中 $x(t)$ 和 $u(t)$ 的相对幅值可以权衡。事实上, 如果 R 相对于 Q_p 和 Q_v 较大, 则 PI(4.6.4) 中的控制力被状态加权更重。然后最优控制将通过选择较小的控制增益来使 $u(t)$ 更小; 因此响应时间将增加。另一方面, 选择较小的 R 将增加 PD 增益并使误差更快消失。

如果 Q_p, Q_v 和 R 是对角的, 那么 PD 增益 K_p, K_v 也是对角的。具有非对角 Q_p, Q_v 和 R 的 LQ 方法提供了关节之间耦合的外反馈环的可能性, 有时可以改善性能。LQ 设计的另一个重要特征是定理中提到的保证鲁棒性。这在近似计算力矩设计中非常有用, 其中

$$\tau_c = \hat{M}(\ddot{q}_d - u) + \hat{N} \quad (4.133)$$

其中 \hat{M} 和 \hat{N} 可以是 $M(q)$ 和 $N(q, \dot{q})$ 的简化版本。这种控制器的性能可以预期超过使用任意选择 K_p 和 K_v 设计的控制器。

重要的是要注意, 这种 LQ 设计导致 $e(t), \dot{e}(t)$ 和 $u(t)$ 方面的最小闭环能量。然而, 机器人手臂的实际控制输入为

$$\tau = M(q)(\ddot{q}_d - u) + N(q, \dot{q}) \quad (4.134)$$

尽管使用这种方法没有最小化 $\tau(t)$ 中的能量，但我们可以使用一些范数不等式来写入

$$\|\tau\| \leq \|M(q)\|\|\ddot{q}_d\| + \|M(q)\|\|u\| + \|N(q, \dot{q})\| \quad (4.135)$$

因此，保持小的 $\|u(t)\|$ 可能预期会使 $\|\tau\|$ 更小。

我们已经使用计算力矩（即反馈线性化）方法推导了 LQ 控制器。产生基于相同 Riccati 方程的设计的另一种方法是采用完整的非线性臂动力学并找到 Hamilton-Jacobi-Bellman 方程的近似（即时不变）解 [Luo and Saridis 1985]; [Luo et al. 1986]。

4.7 笛卡尔控制

我们已经看到了如何使机器人操作臂跟踪期望的关节空间轨迹 $q_d(t)$ 。然而，在任何实际应用中，机器人手臂的期望轨迹都是在工作空间或笛卡尔坐标中给出的。一系列重要的论文处理了分解运动操作臂控制 [Whitney 1969]; [Luh et al. 1980]、[Wu and Paul 1982]。在那里，关节运动被分解到笛卡尔坐标中，其中指定了控制目标。

有几种笛卡尔机器人控制方法。例如，可以：

3. 在 3.5 节中使用笛卡尔动力学进行控制设计（参见习题）。
2. 使用逆运动学将期望的笛卡尔轨迹 $y_d(t)$ 转换为关节空间轨迹 $q_d(t)$ 。然后使用关节空间计算力矩控制方案表 4.4.1。
3. 使用笛卡尔计算力矩控制。

4.7.1 笛卡尔计算力矩控制

这种方法从关节空间动力学开始

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (4.136)$$

笛卡尔误差定义为

$$e_y = y_d - y \quad (4.137)$$

其中 $y_d(t)$ 是期望的笛卡尔轨迹， $y(t)$ 是末端执行器笛卡尔位置。

末端执行器笛卡尔位置可以用 $y(t)$ 以多种形式指定，包括： 4×4 臂 T 矩阵；使用欧拉角的 6 向量；或使用四元数的 7 向量。

笛卡尔误差可以指定为 6 向量

$$e_y = \begin{bmatrix} e_p \\ e_o \end{bmatrix} \quad (4.138)$$

其中 $e_p(t)$ 是位置误差， $e_o(t)$ 是方向误差。

假设 $y = h(q)$, 其中 $h(q)$ 是从 $q(t)$ 到 $y(t)$ 的变换。那么相关的 Jacobian 为 $J = \partial h / \partial q$ 且 $\dot{y} = J\dot{q}$ 。

相对于 $e_y(t)$ 的计算力矩控制由下式给出

$$\tau = M(q)J^{-1}(\ddot{y}_d - \dot{J}\dot{q} - u) + N(q, \dot{q}) \quad (4.139)$$

产生误差系统

$$\ddot{e}_y = u + w \quad (4.140)$$

具有干扰

$$w = JM^{-1}\tau_d - \dot{J}\dot{q} \quad (4.141)$$

我们称 (4.7.7) 为笛卡尔计算力矩控制律。

外环控制 $u(t)$ 可以使用已经提到的用于关节空间计算力矩控制的任何技术进行选择 (参见表 4.4.1)。对于 PD 控制, 完整的控制律为

$$\tau_c = M(q)J^{-1}(\ddot{y}_d - \dot{J}\dot{q} + K_v\dot{e}_y + K_p e_y) + N(q, \dot{q}) \quad (4.142)$$

笛卡尔计算力矩控制的一个缺点是需要计算逆 Jacobian。为了避免在每个采样周期求逆 Jacobian, 我们可以使用近似笛卡尔计算力矩控制器

$$\tau_c = \hat{M}\hat{J}^{-1}(\ddot{y}_d - \dot{J}\dot{q} + K_v\dot{e}_y + K_p e_y) + \hat{N} \quad (4.143)$$

通过设置 $\hat{M}\hat{J}^{-1} = I, \hat{N} = G(q)$ 获得此控制律的特殊情况, 产生笛卡尔 PD-gravity 控制器

$$\tau_c = -K_v\dot{e}_y - K_p e_y + G(q) \quad (4.144)$$

4.7.2 笛卡尔误差计算

可以从使用臂 T 矩阵表示的臂运动学测量的关节变量计算实际笛卡尔位置

$$y = T = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.145)$$

并且可以类似地将期望笛卡尔位置表示为 $y_d = T_d$ 。然后可以如下计算适合计算力矩控制的笛卡尔位置误差和速度误差。定义

$$\dot{y} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (4.146)$$

其中 $v(t), v_d(t)$ 是实际和期望的线速度, $\omega(t), \omega_d(t)$ 是实际和期望的角速度。然后

$$\dot{e}_y = \begin{bmatrix} \dot{e}_p \\ \dot{e}_o \end{bmatrix} = \begin{bmatrix} v_d - v \\ \omega_d - \omega \end{bmatrix} \quad (4.147)$$

很容易计算，因为 $\dot{q}(t)$ 可以使用 $\dot{q} = J^{-1}\dot{y}$ 从测量的关节变量确定。

线性位置误差简单地由

$$e_p = p_d - p \quad (4.148)$$

给出。可以获得用于反馈目的合适方向误差 $e_o(t)$ 更困难，但可以定义如下。

分别表示 T 和 T_d 的旋转变换部分为 $R(t), R_d(t)$ 。方向误差可以用将 $R(t)$ 带入 $R_d(t)$ 的 Euler 轴 $k(t)$ 的 $\phi(t)$ 弧度旋转表示。事实上，可以定义 3 向量

$$e_o = k \sin \phi \quad (4.149)$$

其中可以假设 $e_o(t)$ 很小。使用此定义，可以证明 $e_o(t)$ 从 T 和 T_d 使用

$$e_o = \frac{1}{2}(n \times n_d + o \times o_d + a \times a_d) \quad (4.150)$$

找到。整体笛卡尔误差现在由 $e_y = [e_p^T e_o^T]^T$ 给出。

注：读者可以通过[交互式可视化工具](#)对比传统 PID 控制与计算力矩控制的性能差异，观察非线性补偿对轨迹跟踪精度的影响。

4.8 总结

在本章中，我们展示了如何生成定义机器人末端执行器运动的平滑轨迹，该轨迹通过一组指定的点。然后我们涵盖了重要的计算力矩控制器类，它包含许多类型的机器人控制算法。经典和现代控制算法都由该类描述，因此计算力矩提供了旧运动控制算法与本书其余部分更现代算法之间的桥梁。

作为计算力矩算法的特殊类型，我们提到了 PD 控制、PID 控制、PD-plus-gravity、经典关节控制和数字控制。大多数机器人控制算法都是数字实现的，计算力矩为分析数字化和采样周期大小的影响提供了严格的框架。这是通过将数字控制视为近似计算力矩定律并研究误差系统来实现的。

我们展示了现代线性二次外环设计的一些方面，并以笛卡尔控制的讨论结束。

参考文献

- [label=[0], leftmargin=*]Arimoto, S., and F. Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capability," *Proc. First Int. Symp.*, pp. 783–799, MIT, 1984. Åström, K.J., and B. Wittenmark, *Computer Controlled Systems*. Englewood Cliffs, NJ: Prentice Hall, 1984. Bobrow, J.E., S. Dubowsky, and J.S. Gibson, "On the optimal control of robotic manipulators with actuator constraints," *Proc. Am. Control Conf.*, pp. 782–787, June 1983. Chen, Y.-C., "On the structure of the time-optimal controls for robotic manipulators," *IEEE Trans. Autom. Control*, vol. 34, no. 1, pp. 115–116, Jan. 1989. Dawson, D.M., "Uncertainties in the control of robot manipulators," Ph.D. thesis, School of Electrical Engineering, Georgia Institute of Technology, Mar. 1990. Franklin, G.F., J.D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Reading, MA: Addison-Wesley, 1986. Geering, H.P., L. Guzzella, S.A.R. Hepner, and C.H. Onder, "Time-optimal motions of robots in assembly tasks," *IEEE Trans. Autom. Control*, vol. AC-31, no. 6, pp. 512–518, June 1986. Gilbert, E.G., and I.J. Ha, "An approach to nonlinear feedback control with applications to robotics," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-14, no. 6, pp. 879–884, Nov./Dec. 1984. Gourdeau, R., and H.M. Schwartz, "Optimal control of a robot manipulator using a weighted time-energy cost function," *Proc. IEEE Conf. Decision Control*, pp. 1628–1631, Dec. 1989. Hunt, L.R., R. Su, and G. Meyer, "Global transformations of nonlinear systems," *IEEE Trans. Autom. Control*, vol. AC-28, no. 1, pp. 24–31, Jan. 1983. Jayasuriya, S., and M.-S. Suh, "Sub-optimal control strategies for manipulators with actuator constraints: the near minimum-time problem," *Proc. Am. Control Conf.*, pp. 61–62, June 1985. Johansson, R., "Quadratic optimization of motion coordination and control," *IEEE Trans. Autom. Control*, vol. 35, no. 11, pp. 1197–1208, Nov. 1990. Kahn, M.E., and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chains," *Trans. ASME J. Dyn. Syst. Meas. Control*, pp. 164–172, Sept. 1971. Kailath, T., *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1980. Kim, B.K., and K.G. Shin, "Suboptimal control of industrial manipulators with a weighted minimum time-fuel criterion," *IEEE Trans. Autom. Control*, vol. AC-30, no. 1, pp. 1–10, Jan. 1985. LaSalle, J., and S. Lefschetz, *Stability by Liapunov's Direct Method*. New York: Academic Press, 1961. Lee, C.S.G., and M.H. Chen, "A suboptimal control design for mechanical manipulators," *Proc. Am. Control Conf.*, pp. 1056–1061, June 1983. Lewis, F.L., *Optimal Control*. New York: Wiley, 1986 (a). Lewis, F.L., *Optimal Estimation*. New York:

Wiley, 1986 (b). Lewis, F.L., *Applied Optimal Control and Estimation*. Englewood Cliffs, NJ: Prentice Hall, 1992. Luh, J.Y.S., M.W. Walker, and R.P.C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. Autom. Control*, vol. AC-25, no. 3, pp. 195–200, June 1980. Luo, G.L., and G.N. Saridis, "L-Q design of PID controllers for robot arms," *IEEE J. Robot. Autom.*, vol. RA-1, no. 3, pp. 152–159, Sept. 1985. Paul, R.P., *Robot Manipulators*. Cambridge, MA: MIT Press, 1981. Schilling, R.J., *Fundamentals of Robotics*. Englewood Cliffs, NJ: Prentice Hall, 1990. Shin, K.G., and N.D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Autom. Control*, vol. AC-30, no. 6, pp. 531–541, June 1985. Slotine, J.-J.E., and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall, 1991. Whitney, D.E., "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man Machine Syst.*, vol. MMS-10, no. 2, pp. 47–53, June 1969. Wu, C.-H., and R.P. Paul, "Resolved motion force control of robot manipulator," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-12, no. 3, pp. 266–275, June 1982.

习题

4.2 节

[label=4.2-0] **最小时间控制**。当初始和最终速度不为零时，推导最小时间控制切换时间 t_s [cf. (4.2.11)]。 **多项式路径插值**。希望将单个关节从 $q(0) = 0, \dot{q}(0) = 0$ 通过点 $q(1) = 5, \dot{q}(1) = 40$ 移动到最终位置/速度 $q(2) = 10, \dot{q}(2) = 0$ 。确定此两个间隔路径所需的三次插值多项式。绘制生成的路径并验证它满足 $q(t)$ 和 $\dot{q}(t)$ 的指定要求。 **LFPB**。使用 LFPB 重复习题 4.2-2。 **加速度匹配的多项式路径**。推导在通过点匹配位置、速度和加速度所需的插值多项式。

4.3 节

[label=4.3-0, resume] **柔性耦合系统仿真**。使用计算机仿真重现示例 3.6.1 中带柔性耦合轴的电机的仿真结果。 **非线性系统仿真**。Van der Pol 振荡器是一个具有一些有趣特性的非线性系统。仿真初始条件 $x_1(0) = 0.1, x_2(0) = 0.1$ 下的动力学。使用参数值 $\alpha = 0.1$ ，然后 $\alpha = 0.8$ 。绘制 $x_1(t)$ 和 $x_2(t)$ ，以及相平面中的 $x_2(t)$ 与 $x_1(t)$ 。

4.4 节

[label=4.4-0, resume] 证明 (4.4.32)。 **PD 计算力矩仿真**。使用各种 PD 增益值重复示例 4.4.1。 **经典关节控制**。证明 (4.4.55)、(4.4.57)、(4.4.60) 和 (4.4.62)。 **带负载不确定性的 PD 计算力矩**。在示例 4.4.1 中，假设 m_2 在 $t = 5$ s 时从 1 kg 变为 2 kg。CT 控制器仍使用 $m_2 = 1$ 的值。使用仿真绘制误差时间历史。 **带负载不确定性的 PID 计算力矩**。使用 PID 外环重复习题 4.4-4。 **带摩擦不确定性的 PD 计算力矩**。将形式为 $F(q, \dot{q}) = F_v \dot{q} + K_d \text{sgn}(\dot{q})$ 的摩擦添加到臂动力学中，但不添加到 CT 控制器中。仿真不同 PD 增益下的性能。 **带摩擦不确定性的 PID 计算力矩**。使用 PID 外环重复习题 4.4-6。 **带执行器动力学的 PD 计算力矩**。[label=()]

20. (a) 为具有执行器动力学的双连杆平面肘臂设计 CT 控制律。取连杆质量和长度为 1 kg, 1 m。取电机参数为 $J_m = 0.1$ kg-m², $b_m = 0.2$ N-m/rad/s, $R = 5\Omega$, 设置齿轮比 $r = 0.1$ 。
 - (b) 对各种 PD 增益值仿真控制器。
9. **带忽略执行器动力学的 PD 计算力矩**。假设 CT 控制器仅使用臂动力学设计，忽略了执行器动力学。使用示例 4.4.1 中的 PD 增益。对各种齿轮比 r 值仿真控制

律。

10. **仅使用执行器动力学的 PD 计算力矩。**仅使用执行器动力学设计 CT 控制器，不使用臂动力学。对各种齿轮比 r 值仿真控制律。
11. **带执行器动力学的经典关节控制。**像习题 4.4-8 那样包含执行器动力学重复示例 4.4.4。
12. **带柔性耦合的 PD 计算力矩。**将示例 3.6.1 中的柔性轴与示例 4.4.1 中的双连杆臂结合起来。
13. **带近似 CT 控制的误差动力学。**考虑示例 3.2.1 中的双连杆极坐标臂，具有形式为 $F(\dot{q}) = F_v \dot{q} + K_d \text{sgn}(\dot{q})$ 的摩擦。对以下情况找到误差动力学 (4.4.44):

[label=()]摩擦未包含在 CT 控制律中。在 CT 控制律中负载质量 m_2 不完全已知。使用 PD-gravity CT。使用 PD 经典关节控制，没有非线性项。

4.5 节

[label=4.5-0, resume]数字控制仿真。[label=()]

- (a) 使用周期为 1 s 而不是 2 s 的期望轨迹重复示例 4.5.1。
 - (b) 通过删除图 4.5.7 中将初始速度估计置零的行重新进行仿真。
 - (c) 尝试使用示例中方程 (1) 给出的从 \dot{q}_d 和 v_k 计算 \dot{e}_k 的替代技术仿真数字 CT 控制器。
2. **数字控制仿真。**将示例 4.4.3 中的 PD-gravity CT 控制器转换为数字控制器。
 3. **数字控制的误差动力学。**使用形式 (4.5.3) 的数字控制找到表 4.4.1 中的误差系统。
 4. **抗饱和和保护。**在示例 4.4.4 中的机器人控制器上实现抗饱和和保护。

4.6 节

[label=4.6-0, resume]最优 LQ 外环 PD 增益。验证 (4.6.15)。使用 LQ 外环设计的鲁棒控制。使用从 LQ 设计找到的 PD 增益重做示例 4.4.3 中的 PD-gravity 仿真。

4.7 节

[label=4.7-0, resume]直接笛卡尔计算力矩设计。从 3.5 节中的笛卡尔动力学开始并设计计算力矩控制器。**近似笛卡尔计算力矩**。推导与控制律 (4.7.11) 相关的误差系统动力学。**笛卡尔 PD-Plus-Gravity 控制**。使用笛卡尔计算力矩控制重复示例 4.4.3，其中轨迹在工作空间坐标中给出。

Chapter 5

机器人操作器的鲁棒控制

引言

在本章中，我们讨论当机器人的动力学模型存在不确定性时的控制问题。这种情况可能源于机器人携带未知负载，或者精确计算机器人动力学的代价过高。本章的鲁棒控制器是通过第 4 章设计的控制器进行改进而获得的。

5.1 引言

不确定系统的控制通常采用自适应控制或鲁棒控制两种方法之一来实现。在自适应方法中，设计者尝试让控制器“学习”系统的不确定参数，如果设计得当，最终将成为该系统的“最优”控制器。在鲁棒控制方法中，控制器具有固定结构，能够为一类包含当前被控对象的系统提供“可接受”的性能。一般而言，自适应方法适用于更广泛的不确定性范围，但鲁棒控制器更易于实现，且不需要时间来“调整”控制器以适应特定的被控对象。在本章中，我们回顾用于控制机器人运动的各种鲁棒控制设计。自适应控制方法将在第 6 章讨论。

本章介绍的鲁棒控制方法可用于分析第 4 章讨论的机器人制造商使用的简单控制器的性能，并提出改进和修改建议。事实上，我们能够根据对机器人动力学知识的固有缺乏程度，确定第 4 章简单 PID 控制器的适用范围。

本章设计的控制器可以使用输入-输出稳定性工具或状态空间工具进行分析。在输入-输出方法中，使用小增益定理或被动性定理来证明受控机器人的稳定性。在状态空间方法中，大多数设计使用基于李雅普诺夫的论证来证明稳定性。有关这两种方法的概述，请参见第 2 章。

考虑第 3 章给出的机器人动力学：

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau - \tau_d \quad (5.1)$$

假设关节空间中的期望轨迹由时间函数 $q_d(t)$ 给出。如无歧义，我们将省略时间依赖性。设 q_d 、 \dot{q}_d 、 \ddot{q}_d 和 τ_d 为时间的有界函数。类似于第 4 章，我们假设轨迹误差 e 有两个分量：

$$e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} q_d - q \\ \dot{q}_d - \dot{q} \end{bmatrix} \quad (5.2)$$

本章的控制器假设可以测量 q 和 \dot{q} 。如第 3.5 节所述，可以测量除 q 和 \dot{q} 以外的变量。第 4.7 节的笛卡尔计算力矩控制器提供了一个直接跟踪笛卡尔轨迹的设置。我们将讨论限制在关节测量的情况，理解到可以通过首先获得相应的关节轨迹，然后应用本章的方法来跟踪另一个坐标系中的期望轨迹。

然而，我们可以假设 q 和 \dot{q} 的测量 p 和 \dot{p} 受到有界噪声的污染，即：

$$\|w_i\| \leq c_i \quad (5.3)$$

其中 $\|w_i\| \leq c_i$ 。

在第 5.2 节中，我们讨论第 4.4 节的计算力矩类控制器，并研究其鲁棒性。该节分为使用李雅普诺夫稳定性推导鲁棒性的控制器，以及鲁棒性依赖于输入-输出稳定性的控制器。在第 5.3 节中介绍了不一定从计算力矩控制器导出的非线性控制器，包括利用机器人动力学被动性的控制器，以及其他变结构方法和饱和控制器。最后，在第 5.4 节中，我们回顾通过显式或隐式修改机器人动力学来增强控制器鲁棒性的方法。

5.2 反馈线性化控制器

本节设计的控制器可以作为第 3 章反馈线性化（或计算力矩）控制器的改进。它们基本上第 4.4 节的计算力矩类控制器。我们研究静态和动态反馈设计，并比较文献中不同的控制器。请注意，第 4.4 节开始了一项研究，其中一些在那里引入的控制器将在本章再次出现。这里的重点是将文献中分散的许多控制器联系起来，并给它们一个共同的理论依据。

为简化起见，我们假设式 (??) 中 $\ddot{q}_d = 0$ ，式 (??) 中 $w_i = 0$ ，尽管可以轻松考虑有界的 \ddot{q}_d 和 w_i 的影响，并在大多数示例中加以考虑。类似于第 4 章，机器人的动力学被转换为线性系统

$$u = \ddot{q} \quad (5.4)$$

和

$$\dot{e} = Ae + B(\nu - \ddot{q}_d) \quad (5.5)$$

导出非线性计算力矩控制器

$$\tau = M(q)\nu + N(q, \dot{q}) \quad (5.6)$$

由于 $M(q)$ 的可逆性，这给出以下闭环系统：

$$\ddot{e} = u \quad (5.7)$$

这可以用传递函数描述：

$$G(s) = \frac{1}{s^2} I \quad (5.8)$$

然后问题简化为找到线性控制 u 来实现期望的闭环性能；即，在

$$U(s) = C(s)E(s) \quad (5.9)$$

或

$$u = C(s)e \quad (5.10)$$

中找到 F 、 G 、 H 和 J 。注意上述概念表明 $u(t)$ 是当输入 $e(t)$ 施加时系统 $C(s)$ 的输出。还请注意，从式 (??) 可以看出，机器人的不同关节是去耦的，因此在这一层次上，可以设计 n 个独立的 SISO 控制器来控制机器人的 n 个关节。

不幸的是，控制律 (??) 通常由于复杂性或 $M(q)$ 和 $N(q, \dot{q})$ 中存在的 uncertainty 以及 \ddot{q}_d 和 w_i 的存在而无法实现。相反，应用式 (??) 中的 $\hat{\tau}$ ，其中 \hat{M} 和 \hat{N} 是 M 和 N 的估计值：

$$\tau = \hat{M}(q)\nu + \hat{N}(q, \dot{q}) \quad (5.11)$$

这反过来将在线性模型中重新引入一些耦合，并导致 (图 5.2.1)：

$$\dot{e} = Ae + B(u - \eta) \quad (5.12)$$

其中

$$\eta = M^{-1}[\Delta M\nu + \Delta N] + w_1 + M^{-1}\tau_d \quad (5.13)$$

$\Delta M = M - \hat{M}$ ， $\Delta N = N - \hat{N}$ ，因此 η 在 $\hat{M} = M$ 和 $\hat{N} = N$ 时为零。然而，一般而言，向量 η 是 e 和 u 的非线性函数，不能视为外部干扰。它代表由建模不确定性、参数变化、外部干扰、摩擦项甚至可能是测量噪声引起的全局线性化误差动力学的内部干扰 [Spong and Vidyasagar 1987]。

大多数商用机器人实际上使用式 (??) 中的控制器进行控制，其中选择 $\hat{M} = I$ 和 $\hat{N} = 0$ 。例如，参见 [Luh 1983] 和第 4.4 节。选择 $\hat{M} = I$ 是由驱动机器人连杆的强大电

机以及用于将电机输出扭矩提高到可接受水平同时降低速度的减速机构来验证的。选择 $\hat{N} = 0$ 是通过防止不同电机驱动其连杆过快，从而限制科里奥利和向心力扭矩来验证的。这种商用控制器被称为“非基于模型的控制”，自机器人学早期以来一直在使用。

对更高性能的追求正在促使研究人员和制造商使用直接驱动机器人，并尝试使用功率较小但效率更高的电机以更高的速度移动它们 [Asada and Youcef-Toumi 1987]。这一新方向增加了对更鲁棒控制器（如下面描述的）的需求。

本节的方法围绕设计线性控制器 $C(s)$ 展开，使得图 5.2.1 中的完整闭环系统以某种合适的方式（例如，一致最终有界、全局渐近稳定、 L_p 稳定等）对于给定类别的非线性扰动 η 是稳定的。换句话说，选择式 (??) 中的 $C(s)$ ，使得式 (??) 中的误差 $e(t)$ 以某种期望的方式稳定。

$$\ddot{e} = u - \eta \quad (5.14)$$

在使用这种方法时，通常对旋转关节机器人做出合理的假设 (??)-(??) [Spong and Vidyasagar 1987]。在下文中， μ_1 、 μ_2 、 a 、 α_0 、 α_1 和 α_2 是依赖于不确定性大小的非负有限常数。

$$\mu_1 I \leq M^{-1}(q) \leq \mu_2 I \quad (5.15)$$

$$\|M^{-1}\Delta M\| \leq a \leq 1 \quad (5.16)$$

$$\|\eta\| \leq \alpha_0 + \alpha_1 \|e\| + \alpha_2 \|e\|^2 \quad (5.17)$$

回顾不等式 (??) 在第 3.3 节中引入，并注意上述不等式中使用的范数可以是 ∞ 范数或 2 范数，取决于应用。还注意，对于具有棱柱关节的机器人，边界 μ_i 和 α_i 是 q 的标量函数，并且 $\mu_1 \approx \mu_2$ 使得 $a = (\mu_2 - \mu_1)/(\mu_2 + \mu_1)$ [Spong and Vidyasagar 1987]。最后，(??) 是第 3.3 节讨论的科里奥利和向心项性质的结果。

我们将给出反馈线性化方法的不同代表性设计，从使用李雅普诺夫稳定性理论研究行为的控制器开始。

5.2.1 李雅普诺夫设计

静态反馈补偿器已被广泛使用，始于 [Freund 1982] 和 [Tarn et al. 1984] 的工作。考虑式 (4.4.13) 中引入的控制器：

$$u = -K_p e - K_v \dot{e} \quad (5.18)$$

使得

$$\dot{e} = A_c e + B\eta \quad (5.19)$$

可以看出, 通过将 A_c 的极点足够远地放置在左半平面, 可以保证在存在 η 的情况下闭环系统的鲁棒稳定性。这在 [Arimoto and Miyazaki 1985] 中对于 $\ddot{q}_d = 0$ 的情况被证明是正确的, 如定理 4.4.1 和例 4.4.3 中所述。对于轨迹跟踪问题, 假设 $\ddot{q}_d \neq 0$, 在 [Dawson et al. 1990] 中也证明了这一点, 如定理 4.4.2 中所述。

使用李雅普诺夫稳定性概念设计的鲁棒控制器有很多, 选择李雅普诺夫函数候选和设计增益 K 以确保李雅普诺夫函数候选沿式 (??) 轨迹下降的方法也有很多种。然而, 为了减少渐近轨迹误差, 可能需要过大的增益 (参见例 4.4.3)。因此, 我们选择使用被动性定理和增益矩阵 K 的选择, 使闭环系统的线性部分成为严格正实 (SPR) 系统。如第 2.11 节所述, 可以选择一个输出使闭环系统成为 SPR 系统; 因此允许对 $M(q)$ 的知识存在较大的被动不确定性。事实上, 可以使用状态反馈控制器来定义适当的输出 Ke , 使得输入-输出闭环线性系统 $K(sI - A + BK)^{-1}B$ 是严格正实的 (SPR)。考虑以下闭环线性系统:

$$\dot{e} = (A - BK)e + B\eta \quad (5.20)$$

$$z = Ke \quad (5.21)$$

$$r = K\dot{e} = Ke + Ke \quad (5.22)$$

然后可以使用定理 2.11.5 证明, 如果以下条件成立, 则该系统是 SPR 的:

$$K_v = 2aI, \quad K_p = 4a^2I \quad (5.23)$$

选择

$$\Lambda = 2aI \quad (5.24)$$

使得

$$r = \dot{e} + \Lambda e \quad (5.25)$$

是以下李雅普诺夫方程的正定解

$$A_c^T P + P A_c = -Q \quad (5.26)$$

和

$$K = B^T P \quad (5.27)$$

下一个定理给出了轨迹误差一致有界的充分条件。

定理 5.1. 定理 5.2-1: 如果满足以下条件, 式 (??) 给出的闭环系统将是一致有界的:

$$\hat{M} \geq \mu_2 I \quad (5.28)$$

和

$$K_v = 2aI, \quad K_p = 4aI \quad (5.29)$$

3. 证明. 考虑式 (??) 给出的闭环系统, 使用控制器 (??), 并选择以下李雅普诺夫函数候选:

$$V = e^T P e + \frac{1}{2} e^T K_v e + \delta(e) \quad (5.30)$$

其中 $\delta(e)$ 是对应于 SPR 系统 (??) 的李雅普诺夫函数。那么如果 $\hat{M} \geq 0$, 我们有 $V > 0$ 。当 $\hat{M} \geq \mu_2 I$ 时满足此条件。然后微分得到

$$\dot{V} = -e^T Q e + 2\delta^T(e)\eta + e^T K_v \dot{e} \quad (5.31)$$

为了保证 $\dot{V} < 0$, 回忆边界 (??)-(??), 并写出

$$\dot{V} \leq -\lambda_{\min}(Q)\|e\|^2 + 2\alpha_0\|e\| + 2\alpha_1\|e\|^2 + 2\alpha_2\|e\|^3 \quad (5.32)$$

其中 α_i 是常数。注意可以从 (3) 中提出因子 $\|e\|$ 而不影响方程的符号确定性。然后如果满足以下条件, 误差的均匀有界性就得到保证:

$$\lambda_{\min}(Q) > 2\alpha_1 + 2\alpha_2\|e\| \quad (5.33)$$

这可以通过以下条件保证:

$$a > \alpha_1 + \alpha_2\|e\| \quad (5.34)$$

或

$$K_v > 2(\alpha_1 + \alpha_2\|e\|)I, \quad K_p > 4(\alpha_1 + \alpha_2\|e\|)^2 I \quad (5.35)$$

误差将被一个随着 a 增大而趋于零的项所限制 (有关详细信息, 请参见定理 2.10.3 及其在 [Dawson et al. 1990] 中的证明)。这一分析允许 ΔM 任意大, 只要 $\hat{M} \geq \mu_2 I$,

如下例所示。事实上，如果 N 是已知的，由于在这种情况下 $\Delta N = 0$ ，从被动性定理可以确保全局渐近稳定性。控制器总结在表 5.2.1 中。

研究 (??) 并尝试理解其各项的贡献是有启发性的。以下选择将有助于满足 (??)：

1. 对应于大 a 的大增益 K_p 和 K_v 。
2. 对 N 的良好了解，这转化为小的 α_i 。
3. 大的 μ_1 或大的惯性矩阵 $M(q)$ 。
4. 具有小 c 的轨迹，即小的期望加速度 \ddot{q}_d 。

□

以下示例说明了条件 (??) 的充分性以及更大增益 K_p 和 K_v 的影响。

示例 5.1 (静态控制器 (李雅普诺夫设计))。例 5.2-1：在本章的所有示例中，我们使用第 3 章例 3.2.2 中首次描述的双连杆旋转关节机器人，其动力学在这里重复：

$$\tau = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) \quad (5.36)$$

其中

$$M(q) = \begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos q_2 & m_2a_2^2 + m_2a_1a_2 \cos q_2 \\ m_2a_2^2 + m_2a_1a_2 \cos q_2 & m_2a_2^2 \end{bmatrix} \quad (5.37)$$

参数 $m_1 = 1\text{kg}$, $m_2 = 1\text{kg}$, $a_1 = 1\text{m}$, $a_2 = 1\text{m}$, $g = 9.8 \text{ m/s}^2$ 。让本章所有示例中使用的期望轨迹由下式描述：

$$q_d = \begin{bmatrix} \sin t \\ \cos t \end{bmatrix} \quad (5.38)$$

然后 $\ddot{q}_d = \begin{bmatrix} -\sin t \\ -\cos t \end{bmatrix}$ 。可以证明

$$\mu_1 = 0.5, \quad \mu_2 = 3.5 \quad (5.39)$$

让 $\Delta M = 0$ 然后

$$\alpha_0 = 9.8, \quad \alpha_1 = 0, \quad \alpha_2 = 0 \quad (5.40)$$

或

$$\|\eta\| \leq 9.8 \quad (5.41)$$

然后使用 $\hat{M} = 6I$ 和 $a = 172$ 来满足 (??)。事实上, 这些值将导致比实际需要的更大的控制器增益。相反, 假设我们让 $\hat{M} = 6I$, $\hat{N} = 0$, 并且

$$K_p = 50I, \quad K_v = 25I \quad (5.42)$$

注意这基本上是一个计算力矩类的 PD 控制器。机器人轨迹的仿真如图 5.2.2 所示。我们还从 $e(0) = 0$ 开始我们的仿真。增加增益的效果如图 5.2.3 所示, 对应于控制器

$$K_p = 225I, \quad K_v = 30I \quad (5.43)$$

注意, 至少最初, 高增益情况需要更大的扭矩 (比较图 5.2.2c 和 5.2.3c), 但误差幅值的大小通过消耗更多努力而大大降低。

这些静态控制器的一致有界性还有其他证明。特别是, [Dawson et al. 1990] 中的结果提供了用控制器增益表示的 e 界限的显式表达式。为了简洁起见并展示不同的设计, 我们选择在本节中限制我们的发展到一个控制器。

如第 4.4 节所讨论的, 即使使用精确的计算力矩控制器, 如果存在干扰, 也可能存在残余的稳态误差。解决这个问题的常见方法 (并且可以消除恒定干扰) 是引入积分反馈, 如第 4.4 节所做。这样的控制器可以再次在鲁棒控制器框架中使用, 如果避免积分器饱和问题, 将导致类似的改进 (参见第 4.4 节)。

在下一节中, 我们使用输入-输出稳定性方法来显示类似于这里设计的静态控制器的稳定性, 并设计更一般的动态补偿器。

5.2.2 输入-输出设计

在本节中, 我们将展示使用输入-输出方法验证轨迹误差稳定性的设计。特别是, 我们展示了误差的 L_∞ 和 L_2 稳定性的控制器。我们将本节分为处理先前描述的静态控制器的小节, 以及处理更一般的动态控制器的子节。

静态控制器

这些控制器具有与第 4.4 节和前一节中描述的相同的结构。不同之处在于这里我们使用输入-输出概念而不是李雅普诺夫理论暗示的状态空间方法来显示误差的稳定性。在 [Craig 1988] 中, 使用静态控制器展示了误差信号的有界性。(??)-(??) 中使用的范数然后是 L_∞ 范数。该控制器的开发始于假设 (??)、(??) 和 (??) 到

$$\|\eta\| \leq \alpha_0 + \alpha_1 \|\dot{e}\| + \alpha_2 \|\dot{e}\|^2 \quad (5.44)$$

的修改。这个假设是合理的, 因为 N 由重力和速度相关项组成, 可以独立于位置误差 e 进行限制 [参见 (??)]。我们还假设 $\ddot{q}_d = 0$ 。让我们然后选择状态反馈控制器 (??), 这里为了方便重复:

$$u = -K_p e - K_v \dot{e} \quad (5.45)$$

相应的输入-输出微分方程

$$\ddot{e} + K_v \dot{e} + K_p e = \eta \quad (5.46)$$

该方程的框图描述在图 5.2.4 中给出。考虑从 η (作为独立输入) 到 e 的传递函数:

$$E(s) = P_{11}(s)\eta(s) + P_{12}(s)\dot{\eta}(s) \quad (5.47)$$

或

$$E(s) = \frac{1}{s^2 + K_v s + K_p} \eta(s) \quad (5.48)$$

可以看出, K_v 和 K_p 都是对角矩阵, $K_v = 2\sqrt{K_p}$, 在每个关节实现临界阻尼响应 [参见 (4.4.22)、(4.4.30) 和 (3.3.32)]。 $P_{11}(s)$ 和 $P_{12}(s)$ 的无限算子增益是 (参见引理 2.5.2 和例 2.5.8):

$$\|P_{11}\|_\infty = \frac{1}{\lambda_{\min}^2(K_p)}, \quad \|P_{12}\|_\infty = \frac{1}{\lambda_{\min}(K_p)} \quad (5.49)$$

其中

$$\eta = M^{-1}[\Delta M \ddot{q}_d + \Delta N] + w_1 + M^{-1}\tau_d \quad (5.50)$$

然后考虑以下不等式:

$$\|e\|_\infty \leq \|P_{11}\|_\infty \|\eta\|_\infty + \|P_{12}\|_\infty \|\dot{\eta}\|_\infty \quad (5.51)$$

使用 (??)-(??), 我们有

$$\|\eta\|_\infty \leq \alpha_0 + \alpha_1 \|e\|_\infty + \alpha_2 \|e\|_\infty^2 \quad (5.52)$$

以下定理给出了与定理??平行的误差有界性的充分条件。

定理 5.2. 定理 5.2-2: 假设

$$\hat{M} \geq \mu_1 I \quad (5.53)$$

和

$$K_v = 2\sqrt{K_p} \quad (5.54)$$

那么如果满足以下条件，误差的 L_∞ 稳定性就得到保证：

$$\frac{a}{\mu_1} < 1 \quad (5.55)$$

证明. 上述条件是通过对小增益定理应用于闭环系统而得到的，假设 $e(0) = 0$ ，使得二次项 $\|e\|^2$ 很小。有关详细信息，请参见 [Craig 1988]。

注意 (2) 简化为

$$\frac{a}{\mu_1} < 1 \quad (5.56)$$

并进一步简化为

$$\frac{\mu_2 - \mu_1}{\mu_2 + \mu_1} < 1 \quad (5.57)$$

让我们研究上述不等式以确定各项的影响。以下观察有助于满足 (??)。

1. 大的 k_v 有助于满足稳定性条件。注意：这也意味着大的 k_p 。
2. 对 N 的良好了解，这将转化为小的 α_i 。
3. 大的 μ_1 或大的惯性矩阵 $M(q)$ 。
4. 具有小 \ddot{q}_d 的轨迹。
5. 其惯性矩阵 $M(q)$ 在其工作空间中变化不大的机器人（即 $\mu_1 \approx \mu_2$ ），使得 a 很小。注意，小的 a 需要至少在 (??) 中满足 $a < 1$ 。这将转化为严重的要求，即矩阵 \hat{M} 应在机器人的所有配置中接近惯性矩阵 $M(q)$ 。

控制器总结在表 5.2.2 中。

这些观察类似于在不等式 (6) 之后所做的观察，并在下一个示例中说明。□

示例 5.2 (静态控制器 (输入-输出设计))。例 5.2-2：考虑非线性控制器 (??)，其中

$$\hat{M} = \begin{bmatrix} 6 & 1 \\ 1 & 2 \end{bmatrix}, \quad \hat{N} = 0 \quad (5.58)$$

因此，

$$\|M^{-1}\Delta M\|_\infty \leq a = 0.72 \quad (5.59)$$

如果 $k_v > 720$ ，则满足条件 (1)。这当然是一个大界限，可以通过选择更好的 \hat{M} 来改进。对于 $k_p = 225$ 和 $k_v = 30$ 的闭环行为仿真如图 5.2.5 所示。误差幅值比使用可比较控制努力的例 5.2.1 的 PD 控制器所实现的误差要小得多。这种改进是以了解式 (1) 中的惯性矩阵 $M(q)$ 为代价的。

动态控制器

到目前为止讨论的控制器是静态控制器，因为它们没有存储先前状态信息的机制。在第 4 章和本章中，这些控制器只能对当前位置和速度误差进行操作。在本节中，我们提出了三种方法来展示基于反馈线性化方法的动态控制器的鲁棒性。前两种方法是一自由度 (DOF) 反馈补偿器，而最后一种是二自由度补偿器。

一自由度设计

第一类动态控制器称为一自由度控制器，因为它们只能在机器人的测量输出上操作。换句话说，这些是将测量信号通过动态系统反馈到输入之前的控制器。它们应与没有使用动态反馈的静态控制器，以及接下来考虑的二自由度控制器形成对比。

在 [Spong and Vidyasagar 1987] 中，使用因式分解方法设计了一类动态线性补偿器 $C(s)$ ，由稳定传递矩阵 $Q(s)$ 参数化，保证线性系统 (??) 的解 $e(t)$ 是有界的。实际的因式分解方法设计超出了本书的范围，但适用于机器人学的方法论的相当一般的代表性在例 5.2.3 中给出。在 [Spong and Vidyasagar 1987] 中，实际上假设 ΔN 的界限是线性的 [即，在式??中 $\alpha_2 = 0$]，然后找到了标称植物的所有 L_∞ 稳定补偿器的族。虽然 [Spong and Vidyasagar 1987] 中处理了噪声测量的情况，但为了简单起见，我们将自己限制在无噪声情况下。让我们首先回忆系统 (??)，同时抑制 s 依赖性，

$$\ddot{e} = u - M^{-1}[\Delta M u - \Delta N] - w_1 - M^{-1}\tau_d \quad (5.60)$$

并定义

$$u = M^{-1}[\Delta M u - \Delta N] - w_1 - M^{-1}\tau_d \quad (5.61)$$

令 P_i 、 $i = 1, 2$ 的算子增益由下式给出

$$\gamma_1 = \|P_1\|, \quad \gamma_2 = \|P_2\| \quad (5.62)$$

注意 $\gamma_1 = \max(\gamma_{11}, \gamma_{12})$ 。参见引理 2.5.2 和例 2.5.8。然后考虑

$$\ddot{e} = P_2[u - P_1\eta] \quad (5.63)$$

下一个定理给出了轨迹误差 BIBO 稳定性的充分条件。

定理 5.3. 定理 5.2-3: 如果满足以下条件，闭环系统 (??) 的 BIBO 稳定性将得到保证:

$$\gamma_1\gamma_2 < 1 \quad (5.64)$$

事实上，轨迹误差受以下限制:

$$\|e\|_\infty \leq \frac{\gamma_2}{1 - \gamma_1\gamma_2} \|\eta\|_\infty \quad (5.65)$$

证明. 通过小增益定理. 有关详细信息, 请参见 [Spong and Vidyasagar 1987].

如果我们仔细研究 (1), 我们注意到如果满足以下条件, 它将被满足:

1. μ_1 很大或 $M(q)$ 很大。
2. 对 N 的良好了解, 导致小的 α_1 。
3. 由于补偿器 C 的大增益, 导致小的 γ_1 。
4. γ_2 接近 1, 这也可以通过大增益补偿器 C 获得。

注意, 在极限情况下, 随着 $C(s)$ 的增益变得无限大, γ_1 趋于零. 这将条件 (1) 转换为

$$\|\eta\|_\infty < \infty \quad (5.66)$$

还可以从 (??)-(??) 看出, 增加 $C(s)$ 的增益 k 将减小 γ_1 , 因此减小 $\|e\|_\infty$. 现在可以通过选择参数 $Q(s)$ 来满足其他设计标准, 例如抑制 η 的影响, 来获得特定的补偿器. 例如, 可以通过选择 $C(s) = -K$ 来恢复 Craig 的补偿器, 使得控制努力由下式给出

$$u = Ke \quad (5.67)$$

然后注意, 如果 $\alpha_2 = 0$ 和 $\gamma_2 = \gamma_{11}k_p + \gamma_{12}k_v$, 条件 (??) 和 (1) 是相同的. 还可以从 (2) 看出, 较小的 d 导致较小的跟踪误差. 事实上, 如果 $e(0) = 0$ 和 $\alpha_0 = 0$, 可以证明误差的渐近稳定性. 最后, 注意有界干扰的存在将使误差 e 的界限更大, 但不会影响稳定性条件 (1). 该控制器总结在表 5.2.3 中.

因式分解方法给出了所有一自由度稳定补偿器 $C(s)$ 的族. 设计方法在接下来的示例中针对双连杆机器人进行说明. \square

示例 5.3 (动态控制器 (输入-输出设计)). 例 5.2-3: 让例 4.2.1 中的 $G_v(s)$ 因式分解为

$$G_v(s) = N(s)D^{-1}(s) = \tilde{D}^{-1}(s)\tilde{N}(s) \quad (5.68)$$

其中 $N(s)$ 、 $D(s)$ 、 $\tilde{N}(s)$ 和 $\tilde{D}(s)$ 是稳定有理函数的矩阵. 然后我们可以找到

$$N(s) = \tilde{N}(s) = \frac{1}{(s+1)^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.69)$$

和

$$D(s) = \frac{s^2}{(s+1)^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \tilde{D}(s) \quad (5.70)$$

接下来我们求解 Bezout 恒等式 [Vidyasagar 1985] 对于 $X(s)$ 和 $Y(s)$, 它们也是稳定有理函数,

$$Y(s)D(s) + X(s)N(s) = I \quad (5.71)$$

得到

$$X(s) = \frac{2s+1}{(s+1)^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Y(s) = \frac{(s+1)^2}{(s+1)^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.72)$$

然后所有稳定控制器由下式给出

$$C(s) = [Y(s) - Q(s)\tilde{N}(s)]^{-1}[X(s) + Q(s)\tilde{D}(s)] \quad (5.73)$$

其中 $Q(s)$ 是稳定有理函数, 否则是任意的。一个选择当然是让 $Q(s) = 0$, 这导致”中心解”[Vidyasagar 1985]

$$C(s) = Y^{-1}(s)X(s) = \frac{2s+1}{(s+1)^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.74)$$

还可以选择 $Q(s)$ 来满足所需的性能。特别是, [Spong and Vidyasagar 1987] 中提出了以下 $Q(s)$ 的选择:

$$Q(s) = \frac{k}{s(s+1)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.75)$$

这导致以下控制器:

$$C(s) = \frac{(2s+1)(s+1) + k}{s(s+1)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.76)$$

或

$$u = C(s)e \quad (5.77)$$

其中

$$\dot{x} = -x + ke \quad (5.78)$$

和

$$u = (2+k)x + (1+2k)e \quad (5.79)$$

随着 k 的增加, 控制器的干扰抑制性能增强, 但代价是如从 $C(s)$ 表达式中看到的更高增益。该控制器对于 $k = 225$ 的仿真如图 5.2.6 所示。以下观察是有序的: 轨迹误

差比以前任何控制器都小，而扭矩努力是可比较的。此外，控制器的复杂性是可接受的，因为机器人的动力学没有用于实现控制。

如 [Craig 1988] 中讨论并在定理??中提出的，包括更合理的二次界限不会破坏 [Spong and Vidyasagar 1987] 的 L_∞ 稳定性结果。然而，如 [Becker and Grimm 1988] 中所示，除非重新构建问题并做出更多假设，否则不能保证误差的 L_2 稳定性。实际上，误差将是有界的，但可能没有有限能量。特别是，对于 L_2 稳定性成立，不再容忍噪声测量。接下来我们展示了适用于类似于定理??中描述的动态补偿器但不要求 $\alpha_2 = 0$ 的 L_∞ 稳定性结果的扩展。

定理 5.4. 定理 5.2-4: 如果满足以下条件，式 (??) 的误差系统是 L_∞ 有界的:

$$\ddot{q}_d = 0 \quad (5.80)$$

和

$$\gamma_1 \gamma_2 < 1 - a \quad (5.81)$$

证明. 小增益定理的扩展。有关详细信息，请参见 [Becker and Grimm 1988]。

对 (1) 的研究揭示了以下所需特征将有助于满足不等式:

1. 由于大的 $M(q)$ 导致大的 μ_1 。
2. 小的 γ_1 和接近 1 的 γ_2 ，这将来自大增益补偿器 C 。
3. 小的 α_i ，这将来自对 N 的良好了解。
4. 由于小的 \ddot{q}_d 导致的小的 c 。

注意，如果 $\gamma_1 = \max(\gamma_{11}, \gamma_{12})$ 和 $\gamma_2 = k = \gamma_{11}k_p + \gamma_{12}k_v$ ，则恢复了 Craig 在定理??中的条件。

另一方面，假设 $\ddot{q}_d = 0$ 和 $\alpha_2 = 0$ ，如果满足以下条件，则 [Becker and Grimm 1988] 中证明了 e 的 L_2 稳定性:

$$\gamma_1 \gamma_2 < 1 - a \quad (5.82)$$

其中 γ_i 如引理 2.5.2 中所示。该控制器总结在表 5.2.4 中。 \square

示例 5.4 (动态一自由度设计). 例 5.2-4: 使用与例 5.2.3 相同的线性控制器，但假设 N 中的二次速度项是已知的，使得 $\alpha_2 = 0$; 即

$$\|\eta\| \leq \alpha_0 + \alpha_1 \|e\| \quad (5.83)$$

结果显示在图 5.2.7 中, 参数 $k = 150$ 。它们与图 5.2.6 中例 5.2.3 的结果看起来并没有显著不同。这是由于速度项在此特定应用中的贡献确实可以忽略不计。然而, 这些项在更快的轨迹中将做出更重要的贡献。

二自由度设计

众所周知, 二自由度结构是最通用的线性控制器结构。二自由度设计允许我们同时指定对指令输入的期望响应并保证闭环系统的鲁棒性。该设计在第 2 章例 2.11.4 中简要讨论。它与本章的其他设计在精神上不同, 因为它依赖于经典的频域 SISO 概念。一般结构如图 5.2.8 所示。在 [Sugie et al. 1988] 中设计并仿真了一个二自由度鲁棒控制器, 将在此呈现。让植物由式 (??) 给出, 并考虑以下因式分解:

$$G(s) = N(s)D^{-1}(s) \quad (5.84)$$

其中

$$D(s) = s^2, \quad N(s) = I \quad (5.85)$$

下一个结果呈现了一个二自由度补偿器, 它将 (在 L_∞ 意义上) 鲁棒地稳定误差系统。

定理 5.5. 定理 5.2-5: 考虑图 5.2.8 的二自由度结构。让 $K_1(s)$ 是稳定系统, $K_2(s)$ 是稳定 $G(s)$ 的补偿器。然后控制器

$$u = s^2 K_1 v + K_2 (K_1 v - q) \quad (5.86)$$

将导致闭环系统

$$q = K_1 v \quad (5.87)$$

并且闭环误差系统 (??) 将是 L_∞ 稳定的。

证明. 通过简单的框图操作, 可以证明闭环系统是 $q = K_1 v$ 。实际的鲁棒性分析涉及很多内容, 将省略, 但特定设计及其鲁棒性在下一个示例中讨论。

从 (2) 可以看出, $K_1(s)$ 用于获得期望的闭环传递函数。然后它应该是稳定的, 为了保证零稳态误差, 我们选择 $v = q_d$ 并确保直流增益 $K_1(0) = 1$ 。最后, 我们希望 $K_1(s)$ 是精确适当的 (即, 零相对阶)。另一方面, $K_2(s)$ 将保证闭环系统的鲁棒性。因此, $K_2(s)$ 应该稳定 $G(s)$ 并提供合适的稳定裕度。它应该包含一个积分项以实现静态精度。其相对阶可以是 -1, 因为 q 和 \dot{q} 都是可用的。□

示例 5.5 (动态二自由度设计). 例 5.2-5: 考虑图 5.2.8 的二自由度结构, 其中

$$K_1(s) = \frac{\omega_1^2}{s^2 + 2\zeta\omega_1 s + \omega_1^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.88)$$

然后一个二自由度调节器由下式描述

$$K_2(s) = \frac{a_1 s + \omega_2}{s(s + b_2) + b_1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.89)$$

其中 $s^3 + b_2 s^2 + b_1 s + 1$ 是稳定多项式, w_1 、 w_2 、 a_1 、 b_1 和 b_2 是设计参数。注意 $K_2(s)$ 是 PID 控制器, 并且由 $K_1(s)$ 给出的闭环系统是自然频率为 w_1 和阻尼比为 ζ 的二阶系统。还请注意, 机器人的输入变为

$$\tau = M(q)\nu + N(q, \dot{q}) \quad (5.90)$$

或

$$\nu = K_2(s)(K_1(s)q_d - q) + K_1(s)q_d \quad (5.91)$$

我们可以立即看到关节位置向量 q 通过 PID 控制器 K_2 进行滤波。因此, 除非测量 \dot{q} , 否则需要 q 的微分。非线性闭环系统的行为如图 5.2.9 所示, 当 $\zeta = 1$ 和 $\eta = 0$, $a_1 = 2$, $b_1 = b_2 = 10$, $w_1 = 8$ 和 $w_2 = 12$ 时。可以看出, 最初扭矩努力和轨迹误差都太大。为了理解该控制器的行为, 考虑极限情况下的控制器 τ (即, 当时间趋于无穷大时)。 $K_1 q_d$ 的输出已稳定到其最终值 q_d , 因此控制器 (3) 变得等效于 PID 补偿器 [参见第 4 章, 方程 (4.4.35)]。似乎 K_1 和 K_2 的不同结构是有保证的, 因为在此期间, 二自由度控制器性能相当差。这是示例的特征, 而不是二自由度方法论固有的流程。事实上, 这种结构在 [Sugie et al. 1988] 的一组跟踪情况下比一自由度 PID 补偿器表现出更好的性能。鼓励读者完成本章末尾与此设计相关的问题, 以便比较一自由度和二自由度设计的性能。

我们已经展示了一大批或多或少基于计算力矩的控制器。我们使用不同的稳定性论证表明, 计算力矩结构是固有鲁棒的, 并且通过在外环线性补偿器上增加增益, 位置和速度误差在范数意义上趋于减小。这类补偿器是迄今为止机器人制造商使用的最常见的结构, 也是最易于实现和研究的。还有更多适合这种结构的补偿器, 同时适用于一些经典控制应用。可以用超前-滞后补偿器代替 PD 和 PID 补偿器。当只有位置测量可用时, 这些特别有吸引力。这种设计在 [Chen 1989] 的离散时间情况下讨论。在非线性观测器领域也有一些工作直接与这个问题相关 [Canudas de Wit and Fixot 1991]。我们参考第 2.11 节中的可观测性讨论。我们还建议本章末尾的一些问题, 讨论反馈线性化设计的进一步修改。

另一方面, 还存在其他类型的控制器, 尽管不那么普遍, 但仍然构成一类非常重要的机器人补偿器。这些是不直接依赖机器人反馈线性化的非线性控制器。相反, 它们可

从其动力学的被动性获得，甚至可以绕过机器人的任何特殊结构特性。这些控制器接下来讨论。

5.3 非线性控制器

存在一类不是计算力矩类控制器的机器人控制器。这些控制器直接由机器人方程获得，而不使用反馈线性化过程。相反，这些控制器可以依赖机器人的其他性质（如其拉格朗日-欧拉描述的被动性），或者甚至可以不考虑机器人的物理特性而获得。一般而言，这些控制器可以写成计算力矩控制器加上一个辅助的非线性控制器。非线性控制项独立于计算力矩项引入不同关节之间的耦合。换句话说，即使计算力矩控制器是一个简单的PID，非线性项也会将所有关节耦合在一起，如定理??和??中所示。

5.3.1 直接无源控制器

首先，我们呈现直接依赖刚性机器人被动结构的控制器，如方程(??)中所述，其中 $\dot{M}(q) - 2V_m(q, \dot{q})$ 通过适当选择 $V_m(q, \dot{q})$ 是斜对称的，如第 3.3 节所述。

基于被动性特性，如果能将 \dot{q} 到 τ 的闭环与被动系统（以及 L_2 有界输入）闭合，如图 5.3.1 所示，则闭环系统将使用被动性定理渐近稳定。注意输入 u_2 提供了一个额外的自由度来满足某些性能标准。换句话说，通过选择不同的 L_2 有界 u_2 ，我们可能能够获得更好的轨迹跟踪或噪声免疫。该结构将显示 \dot{q} 的渐近稳定性，但只有 e 的李雅普诺夫稳定性。另一方面，如果能显示映射 τ 到新向量 r 的系统的被动性， r 是 e 的滤波版本，则关闭 $-r$ 和 τ 之间环路的控制器将保证 e 和 \dot{q} 的渐近稳定性。这种间接使用被动性特性在 [Ortega and Spong 1988] 中说明，将首先讨论。

让控制器由式(??)给出，其中 $F(s)$ 是严格适当、稳定的有理函数， K_r 是正定矩阵，

$$\tau = M(q)\dot{r} + V_m(q, \dot{q})r + G(q) + K_r r \quad (5.92)$$

其中 $r = F^{-1}(s)e$ 。将式(??)代入式(??)并假设无摩擦 [即 $F(\dot{q}) = 0$]，我们得到

$$M(q)\dot{r} + V_m(q, \dot{q})r + K_r r = 0 \quad (5.93)$$

然后可以证明 e 和 \dot{q} 都是渐近稳定的。事实上，选择以下李雅普诺夫函数：

$$V = \frac{1}{2}r^T M(q)r + \frac{1}{2}e^T K_p e \quad (5.94)$$

然后

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2}r^T \dot{M}(q)r + e^T K_p \dot{e} \quad (5.95)$$

从式 (??) 代入 $M(q)\dot{r}$, 我们得到

$$\dot{V} = -r^T K_r r \leq 0 \quad (5.96)$$

因此, r 是渐近稳定的, 可用于证明 e 和 \dot{e} 都是渐近稳定的 [Slotine 1988]。这种方法用于自适应控制文献中设计无源控制器 [Ortega and Spong 1988], 但当 M 、 V_m 和 G 不完全已知时, 其在鲁棒控制器设计中的修改并不明显。这种修改将在变结构设计中给出, 但首先, 我们呈现一个简单的控制器来说明无源补偿器的鲁棒性。

定理 5.6. 定理 5.3-1: 考虑控制律 (1)

$$\tau = M(q)\ddot{q}_d + V_m(q, \dot{q})\dot{q}_d + G(q) + K_p e + K_v \dot{e} + u_2 \quad (5.97)$$

其中 $F(s)$ 是 SPR 传递函数, 由设计者选择, 外部输入 u_2 在 L_2 范数中有界。则 \dot{q} 是渐近稳定的, e 是李雅普诺夫稳定的。

证明. 使用上述控制律, 从图 5.3.1 得到,

$$M(q)\dot{r} + V_m(q, \dot{q})r + K_r r = u_2 \quad (5.98)$$

通过适当选择 $F(s)$ 和 u_2 , 可以应用被动性定理并推断 \dot{q} 和 r 在 L_2 范数中有界, 并且由于 $F(s)^{-1}$ 是 SPR 的 (是 SPR 函数的逆), 推断 \dot{q} 是渐近稳定的, 因为

$$\dot{q} = F(s)^{-1}e \quad (5.99)$$

这将意味着位置误差 e 是有界的但不是渐近稳定的, 在时变轨迹 \dot{q}_d 的情况下。然而, 在设定点跟踪情况下 (即 $\dot{q}_d = 0$), 并且具有重力预补偿, 可以使用 LaSalle 定理推断 e 的渐近稳定性。只要 $F(s)$ 是 SPR 的并且 u_2 是 L_2 有界的, 闭环系统的鲁棒性就得到保证, 无论机器人参数的确切值如何。□

控制器总结在表 5.3.1 中。

示例 5.6 (无源控制器). 例 5.3-1: 我们选择一个 SPR 传递函数

$$F(s) = \frac{1}{s + \lambda}, \quad \lambda > 0 \quad (5.100)$$

并且我们让 $u_2 = \dot{q}_d$ 。注意, 到目前为止使用的期望轨迹违反了 u_2 应该是 L_2 稳定的假设。然而, 从 $e(0) = \dot{e}(0) = 0$ 开始的该控制器的轨迹如图 5.3.2 所示, 对于正弦轨迹表现相当好。当然, 由于被动性定理提供了稳定性的充分条件, 这个示例并不与先前的定理矛盾, 而只是指出了其保守性。

注意定理??的补偿器是第 4 章和前一节 PD 补偿器的推广。在 [Anderson 1989] 中，使用网络理论概念证明了，即使在不存在接触力的情况下，计算力矩类控制器也不是无源的，并且因此可能在存在不确定性的情况下引起不稳定。他对问题的解决方案包括使用比例微分 (PD) 控制器，其具有可变增益 $K_1(q)$ 和 $K_2(q)$ ，依赖于惯性矩阵 $M(q)$ ，即，

$$\tau = K_1(q)e + K_2(q)\dot{e} \quad (5.101)$$

即使 $M(q)$ 不完全已知，闭环误差的稳定性也通过与反馈律的机器人被动性得到保证。这种方法的优点是，现在可以容纳接触力和更大的不确定性。其主要缺点是，尽管保证了鲁棒稳定性，但闭环性能依赖于对 $M(q)$ 的了解，需要其奇异值来找到 K_p 和 K_v 。我们不会讨论这种特定设计，并将感兴趣的读者推荐给 [Anderson 1989]。

5.3.2 变结构控制器

在本节中，我们将使用变结构 (VSS) 控制器的设计分组。VSS 理论已应用于许多非线性过程的控制 [DeCarlo et al. 1988]。这种方法的主要特征之一是只需要将误差驱动到”切换面”，之后系统处于”滑动模式”，不会受到任何建模不确定性和/或干扰的影响。

这些控制器在应用于机器人时有两个主要批评。第一个是，通过忽略机器人的物理特性，这些控制器必然不会比利用拉格朗日-欧拉方程结构的控制器表现得更好（如果不是更差的话）。另一个批评涉及变结构控制器通常相关的”颤振”问题。我们将在本节后面讨论第二个问题，并通过承认，虽然变结构理论的最初应用确实忽略了机器人的物理特性，但后来（如此处讨论的）由 [Slotine 1985] 和 [Chen et al. 1990] 设计的控制器弥补了这个问题来回答第一个问题。

变结构理论在机器人控制中的首次应用似乎是在 [Young 1978] 中，其中使用以下控制器解决了设定点调节问题 ($\dot{q}_d = 0$):

$$\tau_i = \begin{cases} \tau_i^+ & \text{if } r_i > 0 \\ \tau_i^- & \text{if } r_i < 0 \end{cases} \quad (5.102)$$

其中 $i = 1, \dots, n$ 对于 n 连杆机器人， r_i 是切换平面，

$$r_i = \dot{e}_i + \lambda_i e_i, \quad \lambda_i > 0 \quad (5.103)$$

然后使用滑动表面 r_1, r_2, \dots, r_n 的层次结构和给定机器人模型不确定性的界限，可以找到 τ^+ 和 τ^- ，以便将误差信号驱动到切换表面的交点，之后误差将”滑动”到零。该控制器通过将系统强制进入滑动模式来消除关节的非线性耦合。此后设计了其他 VSS 机器人控制器。不幸的是，对于大多数这些方案，从式 (??) 看到的控制努力沿 $r_i = 0$

是不连续的，因此会产生颤振，这可能会激发未建模的高频动力学。此外，这些控制器不利用机器人的物理特性，因此不如利用它们的控制器有效。

为了解决这个问题，[Slotine 1985] 中的原始 VSS 控制器被修改，如下一个定理所述。让我们首先定义一些变量来简化定理的陈述。让

$$\begin{aligned} s &= \dot{e} + \Lambda e \\ \dot{q}_r &= \dot{q}_d - \Lambda e \\ r &= \dot{q} - \dot{q}_r = s \end{aligned} \quad (5.104)$$

其中

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad \lambda_i > 0 \quad (5.105)$$

定理 5.7. 定理 5.3-2: 考虑控制器

$$\tau = \hat{M}(q)\ddot{q}_r + \hat{V}_m(q, \dot{q})\dot{q}_r + \hat{G}(q) - K \text{sgn}(s) \quad (5.106)$$

其中

$$K = \text{diag}(k_1, k_2, \dots, k_n), \quad k_i > 0 \quad (5.107)$$

和

$$\text{sgn}(s) = [\text{sgn}(s_1), \text{sgn}(s_2), \dots, \text{sgn}(s_n)]^T \quad (5.108)$$

误差在有限时间内到达表面

$$s = \dot{e} + \Lambda e = 0 \quad (5.109)$$

此外，一旦在表面上， $q(t)$ 将指数快速趋向 $q_d(t)$ 。

证明. 考虑李雅普诺夫函数候选

$$V = \frac{1}{2} s^T M(q) s \quad (5.110)$$

然后

$$\dot{V} = s^T M(q) \dot{s} + \frac{1}{2} s^T \dot{M}(q) s \quad (5.111)$$

代入 $M(q)\ddot{q}$ 并使用 $\dot{M}(q) - 2V_m(q, \dot{q})$ 的斜对称性质，我们得到

$$\dot{V} = s^T [\tau - M(q)\ddot{q}_r - V_m(q, \dot{q})\dot{q}_r - G(q)] \quad (5.112)$$

然后，如果我们使用由下式给出的控制器

$$\tau = \hat{M}(q)\ddot{q}_r + \hat{V}_m(q, \dot{q})\dot{q}_r + \hat{G}(q) - K\text{sgn}(s) \quad (5.113)$$

我们得到

$$\dot{V} = -s^T K\text{sgn}(s) + s^T [\Delta M(q)\ddot{q}_r + \Delta V_m(q, \dot{q})\dot{q}_r + \Delta G(q)] \quad (5.114)$$

其中

$$\Delta M = \hat{M} - M, \quad \Delta V_m = \hat{V}_m - V_m, \quad \Delta G = \hat{G} - G \quad (5.115)$$

然后选择

$$k_i \geq \|\Delta M(q)\ddot{q}_r + \Delta V_m(q, \dot{q})\dot{q}_r + \Delta G(q)\|_i + \beta_i \quad (5.116)$$

其中 $\beta_i > 0$ 。因此，

$$\dot{V} \leq -\sum_{i=1}^n \beta_i |s_i| < 0 \quad (5.117)$$

这意味着 $r = 0$ 在有限时间内到达。此外，一旦在滑动模式下， e 指数快速收敛到零。□

控制器总结在表 5.3.2 中。

示例 5.7 (VSS 控制器 1). 例 5.3-2: 考虑双连杆机器人并选择 $K = 10I$ 和 $\Lambda = 5I$ 。关节 1 的轨迹误差在图 5.3.3 中给出。注意，由于控制器的无限快速切换，颤振行为变得明显。问题的常见补救措施是通过在扭矩计算中使用 $\text{sat}(s/\phi)$ ， $\phi > 0$ 而不是 $\text{sgn}(s)$ 来牺牲渐近稳定性。饱和函数如图 5.3.4a 所示，将导致误差一致最终有界。这种控制器的行为如图 5.3.5 所示，其中 $\Lambda = 5I$ ， $K = 75I$ ，和 $\phi = 0.001$ 。注意，在这种情况下我们能够大大增加增益，这导致更小的误差轨迹。

最近，在 [Chen et al. 1990] 中引入了另一个 VSS 控制器。我们详细描述该控制器，以给出一个不同 VSS 方法的风味，该方法将利用机器人的动力学。该控制器所需的假设如下列出。

$$\begin{aligned} |M_{ij}(q) - \hat{M}_{ij}(q)| &\leq \bar{M}_{ij} \\ |V_{ij}(q, \dot{q}) - \hat{V}_{ij}(q, \dot{q})| &\leq \bar{V}_{ij} \\ |G_i(q) - \hat{G}_i(q)| &\leq \bar{G}_i \end{aligned} \quad (5.118)$$

注意这些界限与式 (??)-(??) 中给出的界限在精神上不同。当前界限通常更有用，因为它们依赖于惯性矩阵 M 和速度相关扭矩 N 的每个元素的不确定性。

定理 5.8. 定理 5.3-3: 如果输入扭矩由下式给出, 则误差 e 和 \dot{e} 是渐近稳定的

$$\tau = \hat{M}(q)\ddot{q}_r + \hat{V}_m(q, \dot{q})\dot{q}_r + \hat{G}(q) + K\text{sgn}(s) \quad (5.119)$$

其中

$$s = \dot{e} + \Lambda e \quad (5.120)$$

$$\ddot{q}_r = \ddot{q}_d - \Lambda \dot{e} \quad (5.121)$$

$$K = \text{diag}(k_1, k_2, \dots, k_n) \quad (5.122)$$

证明. 选择与定理??中相同的李雅普诺夫函数:

$$V = \frac{1}{2}s^T M(q)s \quad (5.123)$$

然后, 微分并代入 τ , 我们得到

$$\dot{V} = -s^T K\text{sgn}(s) + s^T [\Delta M(q)\ddot{q}_r + \Delta V_m(q, \dot{q})\dot{q}_r + \Delta G(q)] \quad (5.124)$$

使用 (4)-(??) 可以证明这是

$$\dot{V} \leq -\sum_{i=1}^n \beta_i |s_i| < 0 \quad (5.125)$$

因此, 表面 $r(t) = 0$ 在有限时间内到达, 之后如定理??中所讨论的, 误差的指数稳定性结果。□

该控制器总结在表 5.3.3 中。

示例 5.8 (VSS 控制器 2 (饱和)). 例 5.3-3: 双连杆机器人使用该控制器控制, 具有以下设计参数:

$$K = 50I, \quad \Lambda = 5I \quad (5.126)$$

关节 1 的产生的误差如图 5.3.5 所示。由于示例 5.3.2 中遇到的相同颤振问题, 我们选择具有饱和函数和与示例 5.3.2 中相同参数的控制器。绘制的如图 5.3.7 所示。

如从图 5.3.3 和 5.3.5 中看到的, 定理??和??的算法虽然使用了机器人的物理特性, 但由于存在 $\text{sgn}(r)$ 项, 遭受了变结构控制中常见的颤振问题。该问题的常见补救措施是通过在扭矩计算中使用 $\text{sat}(r/\phi)$, $\phi > 0$ 代替 $\text{sgn}(r)$ 来牺牲渐近稳定性, 如示例 5.3.2 和 5.3.3 中所做的那样。我们建议使用项 $\tanh(gs)$, 其中 \tanh 是双曲正切, g 是一个增

益参数，用于调整原点周围 \tanh 的斜率，如图 5.3.7b 所示。这个项是连续可微的，并且对于大的 s 是 $\text{sgn}(s)$ 的良好近似，将导致一致最终有界误差，通过调整 g ，我们能够获得与 $\text{sgn}(s)$ 控制器类似的性能，而没有颤振行为。下一个示例说明了该控制器，并将其与通常的饱和控制器进行比较。

示例 5.9 (VSS 控制器 2 (双曲正切)). 例 5.3-4: 将示例 5.3.2 的饱和控制器行为与 $g = 3$ 的双曲正切修改进行对比，如图 5.3.8 所示。应该对比例 5.3.3 和图 5.3.9 中显示的行为进行类似的比较，其中也使用 $g = 3$ 。

5.3.3 饱和型控制器

在本节中，我们介绍了利用辅助饱和信号来补偿机器人动力学中存在的控制器的控制器，如式 (??) 所给，其中 $V_m(q, \dot{q})$ 在第 3 章中定义。

$$Z(q, \dot{q}) = V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d \quad (5.127)$$

或

$$Z(q, \dot{q}) = N(q, \dot{q}) + \tau_d \quad (5.128)$$

因此， $Z(q, \dot{q})$ 是一个表示摩擦、重力和有界扭矩干扰的 n 维向量。本节中引入的控制器是鲁棒的，因为它们是基于不确定性界限而不是参数实际值设计的。需要以下界限，可以在物理上证明。式 (??) 中的 β_i 是正标量常数，轨迹误差 e 如前所定义。

$$\|Z(q, \dot{q})\| \leq \beta_0 + \beta_1\|e\| + \beta_2\|e\|^2 \quad (5.129)$$

$$\|Z(q, \dot{q}) - Z(q, \dot{q}_d)\| \leq \beta_3 + \beta_4\|e\| + \beta_5\|e\|^2 \quad (5.130)$$

该类的一个代表在 [Spong et al. 1987] 中开发，如下所示。

定理 5.9. 定理 5.3-4: 使用控制器，轨迹误差 e 是一致最终有界 (UUB) 的

$$\tau = \hat{M}(q)\ddot{q}_d + \hat{V}_m(q, \dot{q})\dot{q}_d + \hat{G}(q) + K_v\dot{e} + K_p e + v_r \quad (5.131)$$

其中

$$v_r = \frac{\beta_0 + \beta_1\|e\| + \beta_2\|e\|^2}{1 - a} \text{sgn}(e) \quad (5.132)$$

和

$$\|v_r\| \leq \frac{\beta_0 + \beta_1\|e\| + \beta_2\|e\|^2}{1 - a} \quad (5.133)$$

$$a = \|\hat{M}^{-1}M - I\| \quad (5.134)$$

$$\hat{M} \geq \mu_0 I \quad (5.135)$$

证明. 选择李雅普诺夫函数

$$V = \frac{1}{2}e^T K_p e + \frac{1}{2}\dot{e}^T M(q)\dot{e} \quad (5.136)$$

并按照定理 2.10.4 进行。有关详细信息，请参见 [Spong et al. 1987]。□

注意，在上述方程中，矩阵 B 如式 (??) 中定义， α_i 如式 (??) 中定义，矩阵 P 是式 (??) 中李雅普诺夫方程的对称、正定解，其中 Q 是对称正定矩阵， A_c 在式 (??) 中给出。

$$A_c^T P + P A_c = -Q \quad (5.137)$$

特别是， Q 的选择

$$Q = \begin{bmatrix} K_p & 0 \\ 0 & M(q) \end{bmatrix} \quad (5.138)$$

导致以下 P ：

$$P = \begin{bmatrix} K_p + \frac{1}{2}K_v & \frac{1}{2}M(q) \\ \frac{1}{2}M(q) & M(q) \end{bmatrix} \quad (5.139)$$

因此，式 (??) 中 P 的表达式可以用于式 (2) 中 v_r 的表达式。该设计总结在表 5.3.4 中。

示例 5.10 (饱和控制器 1). 例 5.3-5: 在仿真该控制器时选择了以下设计参数：

$$\hat{M} = 6I, \quad \hat{V}_m = 0, \quad \hat{G} = 0 \quad (5.140)$$

和

$$K_p = 50I, \quad K_v = 25I, \quad e(0) = \dot{e}(0) = 0 \quad (5.141)$$

双连杆机器人跟踪与图 5.3.10 所示相同的轨迹。注意，尽管轨迹误差似乎正在发散，但它们确实最终有界，并且可以通过长时间运行仿真来证明这一点。

仔细检查 Spong 在定理??中的控制器，可以清楚地看出 v_r 通过 p 依赖于伺服增益 K_p 和 K_v 。这可能会模糊调整伺服增益的效果，可以通过如 [Dawson et al. 1990] 中所述的方法来避免。

定理 5.10. 定理 5.3-5: 使用控制器, 轨迹误差 e 是一致最终有界 (UUB) 的

$$\tau = \hat{M}(q)\ddot{q}_r + \hat{V}_m(q, \dot{q})\dot{q}_r + \hat{G}(q) + K_v s + K_p e + v_r \quad (5.142)$$

其中

$$s = \dot{e} + \Lambda e, \quad \dot{q}_r = \dot{q}_d - \Lambda e \quad (5.143)$$

和

$$v_r = \frac{\beta_0 + \beta_1 \|s\| + \beta_2 \|s\|^2}{1 - a} \text{sgn}(s) \quad (5.144)$$

其中 β_i 是正标量。

证明. 我们再次选择

$$V = \frac{1}{2} s^T M(q) s + e^T K_p e \quad (5.145)$$

和

$$\dot{V} = -s^T K_v s - e^T \Lambda^T K_p e + s^T v_r \quad (5.146)$$

并按照定理 2.10.4 进行。有关详细信息, 请参见 [Dawson et al. 1990]。

注意 p 不再包含伺服增益, 因此可以调整 K_p 和 K_v 而不会影响辅助控制 v_r 。如 [Dawson et al. 1990] 中所示, 如果初始误差 $e(0) = 0$ 并通过选择 $K_v = 2K_p = k_v I$, 则跟踪误差可以由以下限制, 这显示了控制参数对跟踪误差的直接影响:

$$\|e\| \leq \frac{\beta_0}{k_v} \quad (5.147)$$

最后, 注意如果 $e(0) = \dot{e}(0) = 0$, 可以推断 $e(t)$ 的一致有界性。该控制器在表 5.3.5 中给出。□

示例 5.11 (饱和控制器 2). 例 5.3-6: 在该示例中, 让

$$\hat{M} = 6I, \quad \hat{V}_m = 0, \quad \hat{G} = 0 \quad (5.148)$$

和

$$K_p = 100I, \quad K_v = 20I, \quad \Lambda = 5I \quad (5.149)$$

该仿真的结果呈现在图 5.3.11 中。经过长时间仿真运行后, 可以看出误差最终是有界的。

注意最后两个控制器虽然使用连续项 v_r ，但可以使用示例 5.3.4 中引入的双曲正切项进行修改。事实上，这些类型的控制器还有许多扩展。一个特别简单的扩展使用机器人未知物理参数的线性动力学，在 (3.3.62) 中给出，其中 $p = \|W\|_{\max}$ 。在这种表述中， W 是一个包含时变但已知项的矩阵，而 θ 是动态公式 (3.3.59) 中未知但恒定项的向量，这里重复：

$$\tau = W(q, \dot{q}, \ddot{q})\theta \quad (5.150)$$

有关此表述及其使用的更多信息在第 6 章中呈现。

5.4 动力学重新设计

在本节中，我们提出了另外两种设计鲁棒控制器的方法。第一种从机器人的机械设计开始，并建议设计机器人使其动力学简单且解耦。然后通过消除其原因来解决鲁棒控制器问题。第二种方法可以重新构建为先前讨论的方法之一，但它提出了一种看待问题的如此新颖的方式，我们决定将其单独包括。

5.4.1 解耦设计

在前面的章节中已经表明，控制器的复杂性直接依赖于机器人动力学的复杂性。因此，设计具有简单动力学使其控制更加容易的机器人是有意义的。这种方法在 [Asada and Youcef-Toumi 1987] 中得到提倡。事实上，它表明某些机器人结构将具有解耦的动力学结构，产生一组解耦的 n 个 SISO 非线性系统，这比一个 MIMO 非线性系统更容易控制。解耦是通过修改臂的尺寸和质量特性以消除速度相关项并解耦惯性矩阵来实现的。这种机器人的说明性示例在下一个示例中给出。

示例 5.12 (解耦设计). 例 5.4-1: 考虑图 5.4.1 中描述的机器人。这种机构被称为五连杆机构，其动力学描述在以下条件成立时：

$$m_1 = m_2, \quad a_1 = a_2, \quad I_1 = I_2 \quad (5.151)$$

由下式给出

$$\tau = M\ddot{q} + G(q) \quad (5.152)$$

其中

$$M = \begin{bmatrix} m_{11} & 0 \\ 0 & m_{22} \end{bmatrix} \quad (5.153)$$

注意惯性矩阵是解耦的且与位置无关。由下式给出的控制器

$$\tau = M(\ddot{q}_d + K_v\dot{e} + K_p e) + G(q) \quad (5.154)$$

具有正定的 K_p 和 K_v 足以实现任何期望的性能水平。特别是，考虑以下情况：

$$K_p = \omega_n^2 I, \quad K_v = 2\zeta\omega_n I \quad (5.155)$$

闭环误差系统是稳定的二阶线性系统，具有正定增益。

一些标准机器人结构也可以通过设计解耦。已经进行了研究以部分或完全解耦多达六个连杆的机器人。感兴趣的读者可以参考 [Yang and Tzeng 1986]、[Asada and Youcef-Toumi 1987] 和 [Kazerooni 1989] 以获取有关此主题的良好讨论。

5.4.2 虚拟机器人概念

解耦设计替代方案非常有用，如果控制工程师可以在早期阶段访问或修改机器人设计。然而，更合理的假设是机器人在控制律实际实施之前已经构建以满足许多机械要求。因此，动力学重新设计很困难，如果不是不可能的话。虚拟机器人概念作为另一种鲁棒设计方法论提出 [Gu and Loh 1988]。该方法的开发描述如下。考虑由下式给出的机器人输出函数

$$y = f(q) \quad (5.156)$$

使得

$$\dot{y} = J(q)\dot{q} \quad (5.157)$$

和

$$\ddot{y} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (5.158)$$

广义输出 y 可以表示机器人末端执行器的坐标或轨迹关节误差 $q_d - q$ 。虚拟机器人概念试图通过控制一个“虚拟”机器人来简化实际机器人的控制律设计，该机器人接近实际机器人。该控制器的选择被证明实现了关节位置由向量 y 的分量描述的虚拟机器人的全局稳定性。

该方法从将 $M(q)$ 分解如下开始：

$$M(q) = \hat{M}(q) + \tilde{M}(q) \quad (5.159)$$

然后使用控制器

$$\tau = \hat{M}(q)\nu + N(q, \dot{q}) \quad (5.160)$$

$$\nu = \ddot{y}_d + K_v \dot{e}_y + K_p e_y \quad (5.161)$$

由于 $\tilde{M}(q)$ 是未知的，然而，实际的 \hat{M} 不可用。然后产生的控制器更简单，可以应用于物理机器人以导致可接受的，如果不是最优的行为。

以下定理说明了保证误差有界性的控制器。

定理 5.11. 定理 5.4-1: 让

$$\|M^{-1}\tilde{M}\| \leq a < 1 \quad (5.162)$$

$$\ddot{y}_d = 0 \quad (5.163)$$

闭环系统的 L_∞ 稳定性如果满足以下条件则得到保证:

$$\|K_p\| \geq \frac{\alpha_0}{1-a} \quad (5.164)$$

证明. 这是定理??的直接结果。有关更多详细信息和说明性示例，请参见 [Gu and Loh 1988]。□

该控制器在表 5.4.1 中显示。

5.5 总结

回顾了刚性机器人鲁棒运动控制器的设计。识别并解释了三种主要设计。所有控制器对于一定范围的不确定参数都是鲁棒的，并将保证位置跟踪误差的有界性。在存在干扰扭矩的情况下，有界误差是可实现的最佳结果。选择哪种鲁棒控制方法的问题很难解析回答，但建议了以下指导方针。当线性性能规格（超调百分比、阻尼比等）可用时，线性多变量方法很有用。一自由度动态补偿器表现相当好，对机器人动力学几乎没有了解。然而，它们可能导致高增益控制律以试图实现鲁棒性。被动控制器易于实现，但不提供易于量化的性能测量。改进的变结构控制器在使用机器人的物理特性时似乎表现良好，而不会产生过度的扭矩努力。饱和控制器在可以容忍短暂瞬态误差时最有用，但最终误差将必须是有界的。

本章的一个共同线索是高增益控制器将保证闭环系统的鲁棒性的事实。然而，挑战是在不需要过大扭矩的情况下保证机器人的鲁棒稳定性。当存在非零初始误差或干扰时，运动控制器的鲁棒性也通过一些示例进行了验证，并在本章末尾的问题中讨论。值得注意的是，尽管机器人的动力学是高度非线性的，但大多数成功的控制器都利用了它们的物理特性和非常特殊的结构。在下一章中，我们在机器人动力学描述不确定的情况下描述自适应控制器的设计。

参考文献

参考文献

- [Abdallah et al. 1991] C.T. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi. “Survey of Robust Control for Rigid Robots”. *IEEE Control Syst. Mag.*, vol. 11, number 2, pp. 24–30, 1991.
- [Anderson 1989] R.J. Anderson. “A Network Approach to Force Control in Robotics and Teleoperation”. Ph.D. Thesis, Department of Electrical & Computer Engineering, University of Illinois at Urbana-Champaign, 1989.
- [Arimoto and Miyazaki 1985] S. Arimoto and F. Miyazaki. “Stability and Robustness of PID Feedback Control for Robot Manipulators of Sensory Capability”. *Proc. Third Int. Symp. Robot. Res.*, Gouvieux, France. July, 1985.
- [Asada and Youcef-Toumi 1987] H. Asada and K. Youcef-Toumi. “Direct-Drive Robots: Theory and Practice”. MIT Press, Cambridge, MA, 1987.
- [Becker and Grimm 1988] N. Becker and W.M. Grimm. “On L2 and L8 Stability Approaches for the Robust Control of Robot Manipulators”. *IEEE Trans. Autom. Control*, vol. 33, number 1, pp. 118–122, January, 1988.
- [Canudas de Wit and Fixot 1991] C. Canudas de Wit and N. Fixot. “Robot Control via Robust Estimated State Feedback”. *IEEE Trans. Autom. Control*, vol. 36, number 12, pp. 1497–1501, December, 1991.
- [Chen 1989] Y. Chen. “Replacing a PID Controller by a Lag-Lead Compensator for a Robot: A Frequency Response Approach”. *IEEE Trans. Robot. Autom.* vol. 5, number 2, pp. 174–182, April, 1989.
- [Chen et al. 1990] Y-F. Chen, T. Mita, and S. Wahui. “A New and Simple Algorithm for Sliding Mode Control of Robot Arms”. *IEEE Trans. Autom. Control*, vol. 35, number 7, pp. 828–829, 1990.

- [Corless 1989] M. Corless. “Tracking Controllers for Uncertain Systems: Application to a Manutec R3 Robot”. *J. Dyn. Syst. Meas. Control* vol. 111, pp. 609–618, December, 1989.
- [Craig 1988] J.J. Craig. “Adaptive Control of Mechanical Manipulators”. Addison-Wesley, Reading, MA, 1988.
- [Dawson et al. 1990] D.M. Dawson et al. “Robust Control for the Tracking of Robot Motion”. *Int. J. Control*, vol. 52, number 3, pp. 581–595, 1990.
- [DeCarlo et al. 1988] R.A. DeCarlo, S.H. Zak, and G.P. Matthews. “Variable Structure Control of Nonlinear Multivariable Systems”. *IEEE Proc.* vol. 76, number 3, pp. 212–232, March, 1988.
- [Freund 1982] E. Freund. “Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators”. *Int. J. Robot. Res.* vol. 1, number 1, pp. 65–78, 1982.
- [Gu and Loh 1988] Y-L. Gu and N.K. Loh. “Dynamic Modeling and Control by Utilizing an Imaginary Robot Model”. *IEEE Trans. Rob. Aut.* vol. 4, number 5, 1988.
- [Kazerooni 1989] H. Kazerooni. “Design and Analysis of a Statically Balanced Direct-Drive Manipulator”. *IEEE Control Syst. Mag.* vol. 9, number 2, pp. 30–34, February, 1989.
- [Luh 1983] J.Y.S. Luh. “Conventional Controller Design for Industrial Robots: A Tutorial”. *IEEE Trans. Syst. Man Cybern.* vol. 13, pp. 298–316, May-June, 1983.
- [Ortega and Spong 1988] R. Ortega and M.W. Spong. “Adaptive Motion Control of Rigid Robots: A Tutorial”. *Proc. IEEE Conf. Decision Control*, Austin, TX, December, 1988.
- [Slotine 1985] J-J.E. Slotine. “The Robust Control of Robot Manipulators”. *Int. J. Rob. Res.* vol. 4, number 4, pp. 49–64, 1985.
- [Slotine 1988] J-J.E. Slotine. “Putting Physics Back in Control: The Example of Robotics”. *IEEE Contr. Syst. Mag.* vol. 8, number 7, pp. 12–17, December, 1988.
- [Spong and Vidyasagar 1987] M.W. Spong and M. Vidyasagar. “Robust Linear Compensator Design for Nonlinear Robotic Control”. *IEEE Trans. Robot. Autom.* vol. 3, number 4, pp. 345–351, August, 1987.

- [Spong et al. 1987] M.W. Spong, J.S. Thorp, and J.M. Kleinwaks. “Robust Microprocessor Control of Robot Manipulators”. *Automatica*. vol. 23, number 3, pp. 373–379, 1987.
- [Sugie et al. 1988] T. Sugie et al. “Robust Controller Design for Robot Manipulators”. *Trans. ASME Dyn. Syst. Meas. Control*, vol. 110, number 1, pp. 94–96, March, 1988.
- [Tarn et al. 1984] T.J. Tarn, A.K. Bejczy, A. Isidori, and Y. Chen. “Nonlinear Feedback in Robot Arm Control”. *Proc. IEEE Conf. Decision Control*, Las Vegas, NV. December, 1984.
- [Utkin 1977] V.I. Utkin. “Variable Structure Systems with Sliding Modes”. *IEEE Trans. Autom. Control* vol. 22, pp. 212–222, April, 1977.
- [Vidyasagar 1985] M. Vidyasagar. “Control Systems Synthesis: A Factorization Approach”. MIT Press, Cambridge, MA. 1985.
- [Yang and Tzeng 1986] D.C-H. Yang and S.W. Tzeng. “Simplification and Linearization of Manipulator Dynamics by the Design of Inertia Distribution”. *Int. J. Rob. Res.* vol. 5, number 3, pp. 120–128, 1986.
- [Yeung and Chen 1988] K.S. Yeung and Y.P. Chen. “A New Controller Design for Manipulators Using the Theory of Variable Structure Systems”. *IEEE Trans. Autom. Control*, vol. 33, number 2, pp. 200–206, February, 1988.
- [Young 1978] K-K.D. Young. “Controller Design for a Manipulator Using Theory of Variable Structure Systems”. *IEEE Trans. Syst. Man Cybern.* vol. 8, number 2, pp. 210–218, February, 1978.

习题

第 5.2 节

5.2–1 我们考虑图 5.5.1 所示的三轴 SCARA 机器人，其中所有连杆都假定为质量 m_i 和长度 a_i 的细均匀杆， $i = 1, 2, 3$ 。然后动力学由下式给出：

请找到式 (??)–(??) 中定义的 α 、 β_i 和 μ_i ，假设 $m_{12}(q) = m_{21}(q) = 0$ 并且

$$\mu_1 \leq \|M(q)\| \leq \mu_2 \quad (5.165)$$

5.2-2 为问题 5.2-1 中的机器人选择期望的设定点 $q_{1d} = 45$ 度, $q_{2d} = 90$ 度, 和 $\tau_d = 0$ 。设计一个如式 (??)-(??) 中描述的 SPR 控制器。此外, 找到一个满足定理??的 a 值。

5.2-3 对于问题 5.2-1 中的机器人, 让 $q_{1d} = 10 \sin t$, $q_{2d} = 10 \cos t$, 和 $\tau_d = 0$ 。

1. 设计一个如式 (??)-(??) 中描述的 SPR 控制器。

2. 现在让期望轨迹为 $q_{1d} = \sin t$, $q_{2d} = \cos t$, 和 $\tau_d = 0$ 。研究减小 q_d 对 a 和轨迹误差的影响。

5.2-4 为问题 5.2-1 中的机器人和问题 5.2-3(b) 中的轨迹选择满足条件 (??) 的 k_v 值。如果 m_1 、 m_2 、 m_3 增加到 20, k_v 和轨迹误差会发生什么?

5.2-5 为问题 5.2-1 中的机器人和问题 5.2-3(a) 中的轨迹设计类似于例 5.2.3 的动态控制器。比较 $k = 10$ 和 $k = 50$ 的性能。

5.2-6 找到满足问题 5.2-5 中设计的不等式 (1) 的 k 值。如果速度和重力项都可用于反馈 (即 $\alpha_i = 0$), 您的条件会发生什么?

5.2-7 考虑问题 5.2-1 中的机器人和问题 5.2-2 中的设定点。此外, 假设所有 $\alpha_i = 0$, 使得速度和重力项都可用于反馈。找到满足 (??) 的增益 k 并实现产生的控制器。

5.2-8 用问题 5.2-3(a) 中的轨迹重复问题 5.2-7。

5.2-9 为问题 5.2-1 中的机器人设计类似于例 5.2.5 中的控制器, 以跟踪问题 5.2-3(a) 中的轨迹。选择该示例中使用的相同参数, 并将产生的行为与您选择的一组参数进行比较。

第 5.3 节

5.3-1 为问题 5.2-1 中的机器人设计类似于例 5.3.1 中的控制器, 以跟踪问题 5.2-3(a) 中的轨迹。选择该示例中使用的相同参数, 并将产生的行为与您选择的一组参数进行比较。

5.3-2 考虑问题 5.2-1 中具有问题 5.2-2 中期望设定点的机器人。设计如定理??中描述的变结构控制器。您可能希望从示例 5.3.2 中的参数值开始您的设计。

5.3-3 用问题 5.2-3(a) 中描述的轨迹重复问题 5.3-2。

5.3-4 考虑问题 5.2-1 中具有问题 5.2-2 中期望设定点的机器人。设计如定理??中描述的变结构控制器。您可能希望从示例 5.3.3 中的参数值开始您的设计。

5.3-5 用问题 5.2-3(a) 中描述的轨迹重复问题 5.3-4。

5.3-6 考虑问题 5.2-1 中具有问题 5.2-2 中期望设定点的机器人。设计如定理??中描述的饱和型控制器。您可能希望从示例 5.3.5 中的参数值开始您的设计。

5.3-7 用问题 5.2-3(a) 中描述的轨迹重复问题 5.3-6。

5.3-8 考虑问题 5.2-1 中具有问题 5.2-2 中期望设定点的机器人。设计如定理??中描述的饱和型控制器。您可能希望从示例 5.3.6 中的参数值开始您的设计。

5.3-9 用问题 5.2-3(a) 中描述的轨迹重复问题 5.3-8。

Chapter 6

机器人操作器的自适应控制

本章概述

在本章中，我们将通过将未知常参数与机器人动力学方程中的已知函数分离，来构建自适应控制器。本章将详细讨论每种自适应控制策略的稳定性类型，以激励控制器的构建。本章还将讨论参数误差收敛、持续激励和鲁棒性等相关问题。

6.1 引言

设计确保渐近轨迹跟踪的刚性机器人机械臂自适应控制律的问题已经引起研究人员多年的关注。开发有效的自适应控制器代表着向高速/高精度机器人应用迈出的重要一步。即使在结构良好的工业设施中，机器人也可能面临描述抓取负载动力学特性的参数不确定性（例如，未知的惯性矩）。由于这些参数难以计算或测量，它们限制了机器人准确操纵大尺寸和大重量物体的潜力。最近已经认识到，在高速应用中，传统方法的准确性受到参数不确定性的严重影响。

为了补偿这种参数不确定性，许多研究人员提出了用于机器人机械臂控制的自适应策略。与第 5 章中讨论的鲁棒控制策略相比，自适应方法的一个优势是，携带未知负载的机械臂的精度会随着时间的推移而提高，因为自适应机制持续从跟踪误差中提取信息。因此，自适应控制器能够在负载变化的情况下提供一致的性能。

直到最近，自适应控制结果才包含跟踪误差全局收敛的严格证明。现在既然已经建立了全局收敛自适应控制律的存在性，就很难证明基于近似模型、局部线性化技术或慢时变假设的控制方案的合理性。在控制文献中，对于什么构成自适应控制算法似乎也没有普遍共识；因此，在本章中，讨论将仅限于在控制律中显式包含参数估计的控制方案。

6.2 计算力矩方法的自适应控制

在第 3 章中, 我们讨论了使用计算力矩控制律来控制由以下方程给出的机器人机械臂动力学:

$$\tau = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (6.1)$$

这种激励实际上非常简单。具体来说, 如果存在一些不想处理的非线性动力学, 可以通过直接抵消非线性来将非线性控制问题转化为线性控制问题。控制线性系统有丰富的可用知识; 因此, 如果机器人模型的精确知识可用, 就不需要复杂的非线性控制技术。

近似计算力矩控制器

当然, 在现实中, 由于与模型公式相关的许多问题, 我们永远无法获得机器人模型的精确知识。机器人应用中不允许精确模型抵消的两个常见不确定性是负载扰动导致的未知连杆质量和未知摩擦系数。处理这些类型参数不确定性的一种方法是在第 3 章给出的计算力矩控制器中使用未知参数的固定估计值代替实际参数。这种近似计算力矩控制器的形式为:

$$\tau = \hat{M}(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + \hat{V}_m(q, \dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q}) \quad (6.2)$$

其中上标“ $\hat{\cdot}$ ”表示用参数估计值代替未知实际参数的估计动力学, K_v 和 K_p 是控制增益矩阵, q_d 用于表示期望轨迹, 跟踪误差 e 定义为:

$$e = q_d - q \quad (6.3)$$

例 6.2-1: 近似计算力矩控制器

我们希望为图 6.2.1 所示的两连杆机械臂 (参见第 2 章中两连杆旋转机器人机械臂动力学) 设计和仿真近似计算力矩控制器。假设摩擦可忽略不计, 连杆长度完全已知, 质量 m_1 和 m_2 已知分别在 0.8 ± 0.05 kg 和 2.3 ± 0.1 kg 范围内, 可能的近似计算力矩控制器可以写为:

$$\tau_1 = \hat{m}_1 l_1^2 (\ddot{q}_{d1} + k_{v1} \dot{e}_1 + k_{p1} e_1) \quad (6.4)$$

$$+ \hat{m}_2 [l_1^2 (\ddot{q}_{d1} + k_{v1} \dot{e}_1 + k_{p1} e_1) + l_1 l_2 C_2 (\ddot{q}_{d1} + k_{v1} \dot{e}_1 + k_{p1} e_1) + l_1 l_2 S_2 (\dot{q}_1 \dot{q}_2) \quad (6.5)$$

$$+ l_1 l_2 S_2 \dot{q}_2 (\dot{q}_{d1} + \dot{q}_{d2}) + l_2^2 (\ddot{q}_{d1} + \ddot{q}_{d2} + k_{v2} (\dot{e}_1 + \dot{e}_2) + k_{p2} (e_1 + e_2))] + g \hat{m}_1 l_1 C_1 + g \hat{m}_2 (l_1 C_1 + l_2 C_{12}) \quad (6.6)$$

其中 $l_1 = l_2 = 1$ m, g 是重力常数。我们选择 $\hat{m}_1 = 0.85$ kg 和 $\hat{m}_2 = 2.2$ kg, 因为假设实际值未知。将上述控制律代入两连杆机器人动力学后, 我们可以形成误差系统:

$$\dot{x} = Ax + W(q, \dot{q}, \ddot{q})\tilde{\varphi} \quad (6.7)$$

其中 $\hat{M}^{-1}(q)$ 是用 \hat{m}_1 和 \hat{m}_2 代替 m_1 和 m_2 的惯性矩阵 $M(q)$ 的逆。矩阵 $W(q, \dot{q}, \ddot{q})$, 有时称为回归矩阵 [Craig 1985], 是一个 2×2 矩阵, 由下式给出:

$$W(\cdot) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \quad (6.8)$$

其中

$$W_{11} = l_1^2 \ddot{q}_1 + gl_1 C_1 \quad (6.9)$$

$$W_{12} = l_1^2 \ddot{q}_1 + l_1 l_2 C_2 (2\ddot{q}_1 + \ddot{q}_2) + l_1^2 \ddot{q}_2 + l_1 l_2 S_2 \dot{q}_2^2 - 2l_1 l_2 S_2 \dot{q}_1 \dot{q}_2 + g(l_1 C_1 + l_2 C_{12}) \quad (6.10)$$

向量 $\tilde{\varphi}$ (称为参数误差向量) 是一个 2×1 向量, 由下式给出:

$$\tilde{\varphi} = \begin{bmatrix} \tilde{m}_1 \\ \tilde{m}_2 \end{bmatrix} \quad (6.11)$$

其中

$$\tilde{m}_1 = m_1 - \hat{m}_1 \quad (6.12)$$

$$\tilde{m}_2 = m_2 - \hat{m}_2 \quad (6.13)$$

相关的跟踪误差 2×1 向量和 (3) 中的 2×2 增益矩阵由下式给出:

$$x = \begin{bmatrix} e_1 \\ e_2 \\ \dot{e}_1 \\ \dot{e}_2 \end{bmatrix}, \quad A = \begin{bmatrix} O_2 & I_2 \\ -K_p & -K_v \end{bmatrix} \quad (6.14)$$

由于误差系统 (3) 的右侧由 q 、 \dot{q} 和 \ddot{q} 的非线性函数组成, 我们无法使用误差系统 (3) 用标准线性控制方法选择 K_v 和 K_p 的值。对于近似计算力矩控制器, 仿真可用于选择适当的 K_v 和 K_p 值。为近似计算力矩控制器选择 K_p 和 K_v 值的一个好起点是使用计算力矩方案为给定期望阻尼比和自然频率指定的相同值。

对于 $m_1 = 0.8 \text{ kg}$ 和 $m_2 = 2.3 \text{ kg}$, 在 $q(0) = \dot{q}(0) = 0$ 、控制器增益设置为:

$$K_p = K_v = \text{diag}\{15, 15\} \quad (6.15)$$

和期望轨迹为:

$$q_{d1} = q_{d2} = \sin t \quad (6.16)$$

的条件下, 对近似计算力矩控制器 (1)-(2) 进行了仿真。跟踪误差如图 6.2.2 所示。正如这些图所示, 跟踪误差保持有界而不是趋于零。这种有界跟踪误差性能的原因是误差系统 (3) 不断地被 (3) 右侧的动力学激励。

自适应计算力矩控制器

对于未知的参数量, 如连杆质量或摩擦系数, 近似计算力矩控制器只是用固定估计值代替未知参数。从例 6.2-1 中的方程 (3) 可以清楚地看到, 如果参数误差向量等于零, 跟踪误差可以被证明是渐近稳定的。对于许多应用, 我们不能假设等于零。例如, 在机器人末端执行器携带未知负载质量的情况下, 总会存在与负载质量相关的未知参数量出现在参数误差向量中。

自适应控制策略可以从启发式推理中得到激励, 即如果随着机械臂的运动调整参数估计而不是始终是固定量, 则可以预期更好的跟踪性能。也就是说, 基于自适应更新规则是机器人配置和跟踪误差的函数的推理, 尝试改变我们的参数估计是合理的。问题就变成了: 如何构建自适应控制策略, 以及这个自适应更新规则如何影响跟踪误差的稳定性? 这两个问题的答案是, 自适应更新规则是从跟踪误差系统的稳定性分析中构建的。也就是说, 我们通过同时构建自适应更新规则和分析跟踪误差系统的稳定性来确保跟踪误差系统的稳定性。

我们将研究的第一个自适应控制策略是 [Craig 1985] 中概述的方法。自适应计算力矩控制器与近似计算力矩控制器 (6.2.2) 相同, 但增加了用于调整参数估计的自适应更新规则。这种自适应控制器基于机器人模型中参数线性出现的特性 (见第 2 章)。也就是说, 机器人动力学 (6.2.1) 可以写成以下形式:

$$\tau = W(q, \dot{q}, \ddot{q})\varphi \quad (6.17)$$

其中 $W(q, \dot{q}, \ddot{q})$ 是 $n \times r$ 已知时间函数矩阵, φ 是 $r \times 1$ 未知常参数向量。这一特性对 Craig 提出的自适应控制类型至关重要, 因为它说明了未知参数与已知时间函数的分离。机器人动力学可以以这种形式分离的原因是, 以向量形式 φ 表达的机器人动力学对参数是线性的。这种未知参数与已知时间函数的分离将用于自适应更新规则的构建以及跟踪误差系统的稳定性分析。

研究自适应计算力矩控制器的第一步是形成跟踪误差系统。注意, 通过使用 (6.2.3), 我们可以将 (6.2.1) 给出的机器人动力学方程写为:

$$\tau = W(q, \dot{q}, \ddot{q})\varphi \quad (6.18)$$

根据 [Craig 1985], 自适应计算力矩控制器由下式给出:

$$\tau = \hat{M}(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + \hat{V}_m(q, \dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q}) \quad (6.19)$$

从跟踪误差的定义可以很容易地看出 (6.2.5) 如何写成:

$$\tau = \hat{M}(q)(\ddot{e} + K_v \dot{e} + K_p e) + \hat{W}(q, \dot{q}, \ddot{q})\hat{\varphi} \quad (6.20)$$

通过利用 (6.2.3), (6.2.6) 可以写为:

$$\tau = \hat{M}(q)(\ddot{e} + K_v \dot{e} + K_p e) + W(q, \dot{q}, \ddot{q})\hat{\varphi} \quad (6.21)$$

其中 $\hat{\varphi}$ 是 $n \times 1$ 向量, 用于表示未知常参数的时变估计。将 (6.2.7) 代入 (6.2.4), 我们可以形成跟踪误差系统:

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{M}^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\varphi} \quad (6.22)$$

其中参数误差为:

$$\tilde{\varphi} = \varphi - \hat{\varphi} \quad (6.23)$$

为了方便, 将 (6.2.8) 重写为状态空间形式:

$$\dot{x} = Ax + B\hat{M}^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\varphi} \quad (6.24)$$

其中跟踪误差向量为:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad A = \begin{bmatrix} O_n & I_n \\ -K_p & -K_v \end{bmatrix}, \quad B = \begin{bmatrix} O_n \\ I_n \end{bmatrix} \quad (6.25)$$

其中 I_n 是 $n \times n$ 单位矩阵, O_n 是 $n \times n$ 零矩阵。

现在跟踪误差系统已经形成, 我们使用 Lyapunov 稳定性分析 (见第 1 章) 来证明通过正确选择自适应更新律, 跟踪误差向量 e 是渐近稳定的。我们首先选择正定的 Lyapunov 类函数:

$$V = \frac{1}{2}x^T Px + \frac{1}{2}\tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (6.26)$$

其中 P 是 $2n \times 2n$ 正定、常数、对称矩阵, Γ 是对角、正定 $r \times r$ 矩阵。也就是说, Γ 可以写成:

$$\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_r\} \quad (6.27)$$

其中 γ_i 是正标量常数。

对时间微分 (6.2.11) 得到:

$$\dot{V} = \frac{1}{2}x^T(PA + A^T P)x + \tilde{\varphi}^T \Gamma^{-1} (B^T P x + \dot{\tilde{\varphi}}) \quad (6.28)$$

在 (6.2.12) 中, 我们使用了 $\Gamma = \Gamma^T$ 这一事实。现在, 将 e 从 (6.2.10) 代入 (6.2.12) 得到:

$$\dot{V} = -\frac{1}{2}x^T Q x + \tilde{\varphi}^T \left(\Gamma^{-1} \dot{\tilde{\varphi}} + W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-T}(q) B^T P x \right) \quad (6.29)$$

合并 (6.2.14) 中的项并使用标量转置性质得到:

$$\dot{V} = -\frac{1}{2}x^T Q x + \tilde{\varphi}^T \left(\Gamma^{-1} \dot{\tilde{\varphi}} + W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-1}(q) B^T P x \right) \quad (6.30)$$

其中 Q 是满足 Lyapunov 方程的正定、对称矩阵:

$$A^T P + P A = -Q \quad (6.31)$$

从第 1 章我们注意到, 为了稳定性, 总是希望 \dot{V} 至少为负半定; 因此, 自适应更新规则的选择变得显而易见。具体来说, 通过代入:

$$\dot{\tilde{\varphi}} = \Gamma W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-1}(q) B^T P x \quad (6.32)$$

(6.2.15) 变为:

$$\dot{V} = -\frac{1}{2}x^T Q x \quad (6.33)$$

为了确定明确的稳定性类型, 我们必须进行进一步分析; 然而, 首先注意 (6.2.17) 给出了参数估计向量的自适应更新规则, 因为 $\dot{\tilde{\varphi}} = -\dot{\hat{\varphi}}$ 。也就是说, 通过回忆实际未知参数是常数, 我们可以将 (6.2.9) 代入 (6.2.17) 以获得参数估计向量 $\hat{\varphi}$ 的自适应更新规则:

$$\dot{\hat{\varphi}} = \Gamma W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-1}(q) B^T P x \quad (6.34)$$

在详细说明跟踪误差的稳定性之前, 我们注意到 Craig 修改了 (6.2.19) 中的自适应更新规则, 以防止稳定性分析中的循环论证 [Craig 1985]。具体来说, 参数估计被迫保持在某个已知区域内。也就是说, 如果参数估计漂出已知区域, 它们将被重置到已知区域内。通过在“软件”中重置参数估计, 我们保证参数估计保持有界。

我们现在详细说明跟踪误差的稳定性。由于 \dot{V} 是负半定的, V 以零为下界, V 在时间段 $[0, \infty)$ 上保持上有界; 此外:

$$\lim_{t \rightarrow \infty} V(x, \tilde{\varphi}, t) = V_\infty \leq V(0) \quad (6.35)$$

其中 V_∞ 是正标量常数。由于 V 是上有界的，从 (6.2.11) 中给出的 V 定义可以明显看出 e 和 $\tilde{\varphi}$ 是有界的，这也意味着 q 、 \dot{q} 和 $\hat{\varphi}$ 是有界的。注意我们已经假设 \ddot{q}_d 是有界的，我们将始终假设期望轨迹及其前两阶导数是有界的。

现在，从机器人方程 (6.2.1) 中，很明显：

$$\ddot{q} = M^{-1}(q)[\tau - V_m(q, \dot{q})\dot{q} - G(q) - F(\dot{q})] \quad (6.36)$$

因此， \ddot{q} 是有界的，因为 τ 仅依赖于有界量 q 、 \dot{q} 和 $\hat{\varphi}$ 。如果 \ddot{q} 是有界的，(6.2.10) 表明 \dot{x} 是有界的。由于 \dot{x} 是有界的，我们可以从 (6.2.18) 中说明 \dot{V} 是有界的。因此，由于 V 以零为下界， \dot{V} 是负半定的， \dot{V} 是有界的，根据 Barbalat 引理（见第 1 章）：

$$\lim_{t \rightarrow \infty} \dot{V} = 0 \quad (6.37)$$

这意味着根据 Rayleigh-Ritz 定理（见第 1 章）：

$$\lim_{t \rightarrow \infty} e = 0 \quad (6.38)$$

(6.2.22) 中给出的信息告诉我们跟踪误差向量 e 是渐近稳定的。但参数误差 $\tilde{\varphi}$ 呢？它是否也收敛到零？从我们的分析中，我们只能说，如果 $\hat{M}^{-1}(q)$ 存在，参数误差保持有界。事实上，这对 (6.2.19) 中给出的参数更新律提出了限制。也就是说，我们必须使用前面讨论的参数重置方法来确保差的参数估计不会导致 $\hat{M}^{-1}(q)$ 的逆爆炸。在 [Craig 1985] 中，概述了一种确保参数估计和 $\hat{M}^{-1}(q)$ 保持有界的可能方法；此外，他展示了这种方法如何不干扰 (6.2.22) 所描述的稳定性结果。我们不会讨论这种参数估计重置方法和由此产生的稳定性证明，因为我们将在本章后面展示 Slotine 和 Li 如何使用更明智的 Lyapunov 函数选择来消除重置参数估计的需要，同时消除回归矩阵 $W(q, \dot{q}, \ddot{q})$ 中加速度测量的需要！

表 6.1: 自适应计算力矩控制器

控制器	
扭矩控制	$\tau = \hat{M}(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + \hat{V}_m(q, \dot{q})\dot{q} + \hat{G}(q)$
自适应更新规则	$\dot{\hat{\varphi}} = \Gamma W^T(q, \dot{q}, \ddot{q})\hat{M}^{-1}(q)B^T P x$
信号定义	$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$, $A = \begin{bmatrix} O_n & I_n \\ -K_p & -K_v \end{bmatrix}$, $B = \begin{bmatrix} O_n \\ I_n \end{bmatrix}$
增益条件	K_p, K_v 是正定对角矩阵
Lyapunov 方程	$A^T P + P A = -Q$ (Q 是正定的)
自适应增益	Γ 是正定的对角矩阵
稳定性结果	e 渐近稳定, $\tilde{\varphi}$ 有界

例 6.2-2: 自适应计算力矩控制器

我们希望为图 6.2.1 所示的两连杆机械臂设计和仿真表 6.2.1 中给出的自适应计算力矩控制器。假设摩擦可忽略不计且连杆长度完全已知，自适应计算力矩控制器可以写成与例 6.2-1 中给出的相同形式，但我们需要找到 \hat{m}_1 和 \hat{m}_2 的更新规则。也就是说，我们使用例 6.2-1 中的方程 (1) 和 (2) 进行关节扭矩控制，然后根据表 6.2.1 制定 \hat{m}_1 和 \hat{m}_2 的更新规则。

为简单起见，在本例中我们选择伺服增益为：

$$K_p = k_p I_n, \quad K_v = k_v I_n \quad (6.39)$$

其中 k_v 和 k_p 是正标量常数， I_n 在这种情况下是 2×2 单位矩阵。我们提议表 6.2.1 中的矩阵 P 选择为：

$$P = \begin{bmatrix} k_p I_n + \frac{k_v}{2} I_n & \frac{1}{2} I_n \\ \frac{1}{2} I_n & I_n \end{bmatrix} \quad (6.40)$$

注意 P 是对称的，如果 k_v 选择大于 1，它是正定的（参见第 1 章中的 Gerschgorin 定理）。为了验证我们对 P 的选择是否给出正定的 Q ，执行矩阵运算：

$$Q = -(A^T P + P A) \quad (6.41)$$

$$Q = \begin{bmatrix} \frac{k_v}{2} I_n & O_n \\ O_n & k_v I_n \end{bmatrix} \quad (6.42)$$

由于我们已经限制 $k_v > 1$ ，可以验证 Q 是正定的、对称矩阵。我们在这里注意到，为一般 Lyapunov 方法找到正定、对称的 P 和 Q 并不总是一项容易的任务。

现在我们有适当的 P ，我们可以制定表 6.2.1 中给出的自适应更新规则。相关的参数估计向量为：

$$\hat{\varphi} = \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \end{bmatrix} \quad (6.43)$$

带有更新规则：

$$\dot{\hat{m}}_1 = \gamma_1 (P_{12} W_{11} + P_{14} W_{21}) \dot{e}_1 + \gamma_1 (P_{13} W_{11} + P_{14} W_{21}) \dot{e}_2 \quad (6.44)$$

和

$$\dot{\hat{m}}_2 = \gamma_2 (P_{12} W_{12} + P_{14} W_{22}) \dot{e}_1 + \gamma_2 (P_{13} W_{12} + P_{14} W_{22}) \dot{e}_2 \quad (6.45)$$

其中 P_{ij} 在 (2) 中定义，形成回归矩阵 W_{ij} 的量在例 6.2.1 中找到。

对于 $m_1 = 0.8 \text{ kg}$ 和 $m_2 = 2.3 \text{ kg}$ ，在 $k_v = 50$ 、 $k_p = 125$ 、 $\gamma_1 = 500$ 、 $\gamma_2 = 500$ 、 $\hat{m}_1(0) = 0.85$ 、 $\hat{m}_2(0) = 2.2$ 以及例 6.2.1 中使用的相同期望轨迹和初始关节条件下，对

自适应计算力矩控制器进行了仿真。跟踪误差和质量估计如图 6.2.4 所示。如图所示，跟踪误差趋于零，参数估计保持有界，正如理论所预测的那样。

6.3 惯量相关方法的自适应控制

在第 6.2 节中，我们展示了自适应控制如何用于补偿参数不确定性。这引导 Craig 开发了自适应计算力矩控制器。我们还给出了实现自适应计算力矩控制器所需的两个限制 [即，需要测量 \ddot{q} 并确保 $\hat{M}^{-1}(q)$ 存在]。这两个限制可能相当麻烦。例如，大多数工业机器人只有位置和速度传感器，由于通常不希望对速度进行微分，我们必须添加额外的昂贵传感器来测量 \ddot{q} 。此外，如果机械臂举起相对于机械臂重量的大未知负载，确保 $\hat{M}^{-1}(q)$ 存在可能极其困难。

在研究人员重新审视自适应计算力矩控制器的结构后，他们开始怀疑是否在设计自适应控制方案时使用了所有关于机器人机械臂的可用信息。也就是说，是否正在利用机械操纵器固有的所有特性？在 [Arimoto 和 Miyazaki 1986] 中，提出了一种带重力补偿的比例-微分 (PD) 反馈控制器。应该注意，该控制器不是反馈线性化方法的产物。相反，该控制器在稳定性分析中使用了惯量相关的 Lyapunov 函数，该函数利用了机械操纵器固有的物理特性。重新审视自适应计算力矩控制器后，可以看出，稳定性分析中使用的 Lyapunov 函数不是惯量相关的，而是有些任意的。

PD 加重力控制器的检验

一种激励 PD 加重力控制器 [Arimoto 和 Miyazaki 1986] 的方法 [Slotine 1988] 是将机械臂动力学写成能量守恒形式：

$$\frac{d}{dt} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] = \dot{q}^T (\tau - G(q) - F(\dot{q})) \quad (6.46)$$

其中 (6.3.1) 的左侧是机械臂动能的导数，(6.3.1) 的右侧表示来自执行器的功率减去由于重力和摩擦而耗散的功率。注意，科里奥利和向心项在 (6.3.1) 中得到了解释，因为这些项与惯性矩阵的时间导数有关。

假设我们现在想要为能量守恒形式 (6.3.1) 中给出的系统设计恒定点控制器（即， $\dot{q}_d = 0$ ）。首先，我们选择惯量相关的 Lyapunov 函数：

$$V = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} e^T K_p e \quad (6.47)$$

由于 Lyapunov 函数可以从启发式上理解为能量函数，这个 Lyapunov 函数似乎相当合理。也就是说，Lyapunov 函数由机器人机械臂系统的动能和额外的能量阻尼项 $\frac{1}{2} e^T K_p e$ 组成。这个能量阻尼项可以被理解为使用物理弹簧使机械臂表现更好。

对时间微分 (6.3.2) 得到：

$$\dot{V} = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} - e^T K_p \dot{q} \quad (6.48)$$

将 (6.3.1) 代入 (6.3.3), 我们有:

$$\dot{V} = \dot{q}^T (\tau - G(q) - F(\dot{q})) - e^T K_p \dot{q} \quad (6.49)$$

因为我们正在解决定点控制问题。由于希望 \dot{V} 至少为负半定, 控制:

$$\tau = K_p e - K_v \dot{q} + G(q) \quad (6.50)$$

受到 (6.3.4) 形式的激励。也就是说, 将 (6.3.5) 代入 (6.3.4) 得到:

$$\dot{V} = -\dot{q}^T K_v \dot{q} - \dot{q}^T F(\dot{q}) \quad (6.51)$$

上面的分析说明 V 除了在 $\dot{q} = 0$ 时外都在减小。我们现在使用这些信息来说明达到了期望的设定点 q_d 。也就是说, 如果 $\dot{q} = 0$, 则 $\ddot{q} = 0$, 且 $\dot{e} = 0$; 因此对于 $\dot{q} = \ddot{q} = 0$, 我们可以利用 (6.2.1) 和 (6.3.5) 以 $K_p e = 0$ 的形式写出闭环系统, 这意味着 $e = 0$ 。从第 1 章中, 现在可以使用 LaSalle 定理来证明跟踪误差 e 是渐近稳定的。

自适应惯量相关控制器

尽管 (6.3.5) 中给出的控制器利用了能量守恒特性, 但它有两个缺点。首先, 控制器仅仅确保机械臂达到期望的设定点。一般来说, 机器人控制设计者必须确保机械臂跟踪期望的时变轨迹。其次, 控制器需要机器人机械臂模型中任何相关参数的精确知识, 因为控制律 (6.3.5) 中包含了重力和摩擦项。

在 [Slotine 和 Li 1985(a)] 中, 利用 (6.3.1) 中给出的能量守恒公式为轨迹跟踪问题设计了自适应控制器。这种控制器可以以与我们处理 PD 加重力控制器大致相同的方式得到激励。换句话说, 我们使用稳定性分析来帮助我们找到自适应控制器。由于我们正在设计自适应轨迹跟踪控制器, 我们应该选择是跟踪误差和参数误差函数的 Lyapunov 函数。Slotine 选择了惯量相关的 Lyapunov 类函数:

$$V = \frac{1}{2} r^T M(q) r + \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (6.52)$$

其中

$$r = \dot{e} + \Lambda e \quad (6.53)$$

其中 Γ 如 (6.2.11) 中定义, Λ 定义为正定的对角矩阵, 使得:

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (6.54)$$

$\tilde{\varphi}$ 如 (6.2.9) 中定义。(6.3.8) 中给出的辅助信号 $r(t)$ 可以被认为是滤波后的跟踪误差。

对时间微分 (6.3.7) 后, 我们有:

$$\dot{V} = r^T M(q) \dot{r} + \frac{1}{2} r^T \dot{M}(q) r + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \quad (6.55)$$

从 (6.3.9) 中很明显我们必须用变量 r 替代; 因此, 我们必须用 r 写出机器人方程。使用 (6.2.1), 机器人动力学可以重写为:

$$\tau = M(q) \dot{r} + V_m(q, \dot{q}) r + Y(\cdot) \varphi \quad (6.56)$$

其中

$$Y(\cdot) \varphi = V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + M(q)(\ddot{q}_d + \Lambda \dot{e}) + G(q) + F(\dot{q}) \quad (6.57)$$

$Y(\cdot)$ 是 $n \times r$ 已知时间函数矩阵。这是与自适应计算力矩控制器公式中使用的相同类型的参数分离; 然而, 注意 $Y(\cdot)$ 不是关节加速度的函数!

将 (6.3.10) 代入 (6.3.9) 得到:

$$\dot{V} = r^T (\tau - V_m(q, \dot{q}) r - Y(\cdot) \varphi) + \frac{1}{2} r^T \dot{M}(q) r + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \quad (6.58)$$

应用斜对称性质 (参见第 2 章), 我们可以将 (6.3.12) 写成:

$$\dot{V} = r^T (\tau - Y(\cdot) \varphi) + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \quad (6.59)$$

再次, 稳定性分析引导我们选择扭矩控制器和自适应更新规则。也就是说, 如果我们选择扭矩控制为:

$$\tau = Y(\cdot) \hat{\varphi} + K_v r \quad (6.60)$$

(6.3.13) 变为:

$$\dot{V} = -r^T K_v r + \tilde{\varphi}^T (\Gamma^{-1} \dot{\tilde{\varphi}} + Y^T(\cdot) r) \quad (6.61)$$

通过选择自适应更新规则为:

$$\dot{\tilde{\varphi}} = \Gamma Y^T(\cdot) r \quad (6.62)$$

(6.3.15) 变为:

$$\dot{V} = -r^T K_v r \quad (6.63)$$

我们现在详细说明跟踪误差的稳定性类型。首先，由于 (6.3.17) 中的 \dot{V} 是负半定的，我们可以说 (6.3.7) 中的 V 是上有界的。利用 V 是上有界的， $M(q)$ 是正定矩阵（参见第 2 章）这一事实，我们可以说 r 和 $\hat{\varphi}$ 是有界的。从 (6.3.8) 中给出的 r 的定义，我们可以使用标准线性控制论证来说明 e 和 \dot{e} （因此 q 和 \dot{q} ）是有界的。由于 e 、 \dot{e} 、 r 和 $\hat{\varphi}$ 是有界的，我们可以使用 (6.3.10) 和 (6.3.14) 来证明 \dot{r} （因此 \ddot{e} ，通过对 (6.3.17) 微分得到）是有界的。其次，注意由于 $M(q)$ 是下有界的，我们可以说 (6.3.7) 中给出的 V 是下有界的。由于 \dot{V} 是下有界的， \dot{V} 是负半定的， \ddot{V} 是有界的，我们可以使用 Barbalat 引理（参见第 1 章）来说明：

$$\lim_{t \rightarrow \infty} r = 0 \quad (6.64)$$

这意味着根据 Rayleigh-Ritz 定理（参见第 1 章）：

$$\lim_{t \rightarrow \infty} e = 0 \quad (6.65)$$

注意 (6.3.8) 是一个由“输入” r 驱动的稳定一阶微分方程；因此，根据标准线性控制论证和 (6.3.19)，我们可以写出：

$$\lim_{t \rightarrow \infty} \dot{e} = 0 \quad (6.66)$$

这一结果告诉我们跟踪误差 e 和 \dot{e} 是渐近稳定的。同样，从上面的分析中，我们只能说参数误差保持有界。稍后，我们将展示在期望轨迹的某些条件下，参数误差也收敛到零。

表 6.3.1 总结了上述导出的自适应控制器，图 6.3.1 描绘了该控制器。

浏览表 6.3.1 后，我们可以看到，关于自适应计算力矩控制器的限制已被消除。具体来说，对于自适应惯量相关控制器，我们不需要加速度测量或任何临时调整参数估计来确保 $\hat{M}^{-1}(q)$ 存在。

表 6.2: 自适应惯量相关控制器

控制器	
扭矩控制	$\tau = Y(\cdot)\hat{\varphi} + K_v r$
自适应更新规则	$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot)r$
辅助信号	$r = \dot{e} + \Lambda e$
增益条件	K_v, Λ, Γ 是正定对角矩阵
信号定义	$e = q_d - q$
稳定性结果	e, \dot{e} 渐近稳定, $\hat{\varphi}$ 有界

例 6.3-1: 自适应惯量相关控制器

我们希望为图 6.2.1 所示的两连杆机械臂设计和仿真表 6.3.1 中给出的自适应惯量相关控制器。假设摩擦可忽略不计且连杆长度完全已知，自适应惯量相关扭矩控制器可以写为：

$$\tau_1 = Y_{11}\hat{m}_1 + Y_{12}\hat{m}_2 + k_{v1}r_1 \quad (6.67)$$

和

$$\tau_2 = Y_{21}\hat{m}_1 + Y_{22}\hat{m}_2 + k_{v2}r_2 \quad (6.68)$$

在扭矩控制表达式中，回归矩阵 $Y(\cdot)$ 由下式给出：

$$Y(\cdot) = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \quad (6.69)$$

其中

$$Y_{11} = l_1^2(\ddot{q}_{d1} + \lambda_1\dot{e}_1) + gl_1C_1 \quad (6.70)$$

$$Y_{12} = l_1^2(\ddot{q}_{d1} + \lambda_1\dot{e}_1) + l_1l_2C_2(2\ddot{q}_{d1} + 2\lambda_1\dot{e}_1 + \ddot{q}_{d2} + \lambda_2\dot{e}_2) + l_2^2(\ddot{q}_{d1} + \lambda_1\dot{e}_1 + \ddot{q}_{d2} + \lambda_2\dot{e}_2) \quad (6.71)$$

$$+ l_1l_2S_2(\dot{q}_2(\dot{q}_{d1} + \lambda_1e_1) - (\dot{q}_1 + \dot{q}_2)(\dot{q}_{d1} + \lambda_1e_1 + \dot{q}_{d2} + \lambda_2e_2)) + g(l_1C_1 + l_2C_{12}) \quad (6.72)$$

按照表 6.3.1 中给出的方式制定自适应更新规则，相关的参数估计向量为：

$$\hat{\varphi} = \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \end{bmatrix} \quad (6.73)$$

带有自适应更新规则：

$$\dot{\hat{m}}_1 = \gamma_1(Y_{11}r_1 + Y_{21}r_2) \quad (6.74)$$

和

$$\dot{\hat{m}}_2 = \gamma_2(Y_{12}r_1 + Y_{22}r_2) \quad (6.75)$$

对于 $m_1 = 0.8 \text{ kg}$ 和 $m_2 = 2.3 \text{ kg}$ ，在 $k_{v1} = k_{v2} = 10$ 、 $\lambda_1 = \lambda_2 = 2.5$ 、 $\gamma_1 = \gamma_2 = 20$ 、 $\hat{m}_1(0) = 0$ 、 $\hat{m}_2(0) = 0$ 以及例 6.2.1 中给出的相同期望轨迹和初始关节条件下，对自适应惯量相关控制器进行了仿真。跟踪误差和质量估计如图 6.3.2 所示。如图所示，跟踪误差是渐近稳定的，参数估计保持有界。

注：读者可以通过[交互式可视化工具](#)观察自适应控制中参数估计的在线收敛过程，理解自适应增益对收敛速度的影响。

6.4 基于无源性的自适应控制器

近年来,许多作者开发了在扭矩控制律或自适应更新规则方面不同的自适应控制方案。为了统一其中一些方法,基于无源性方法开发了一般的自适应控制策略(参见 [Ortega 和 Spong 1988] 和 [Brogliato 等 1990])。在本节中,我们说明如何使用无源性方法来开发一类用于机器人机械臂控制的扭矩控制律和自适应更新规则。

无源自适应控制器

首先,我们定义一个类似于为自适应惯量相关控制器定义的辅助滤波跟踪误差变量。也就是说,我们定义跟踪变量为:

$$r(s) = H(s)e(s) \quad (6.76)$$

其中

$$H(s) = [sI + K(s)]^{-1} \quad (6.77)$$

s 是 Laplace 变换变量。在 (6.4.2) 中, $n \times n$ 增益矩阵 $K(s)$ 的选择使得 $H(s)$ 是严格真、稳定的传递函数矩阵。稍后在本节中分析我们稍后提出的自适应控制器的稳定性时,这一限制的原因将变得清晰。

与前几节一样,我们的自适应控制策略一直集中在将已知时间函数与未知常参数分离的能力上。因此,我们使用 (6.4.1) 和 (6.4.2) 中给出的表达式来定义:

$$Z(\cdot)\varphi = M(q)\dot{r} + V_m(q, \dot{q})r - \tau + M(q)K(s)e + V_m(q, \dot{q})K(s)e \quad (6.78)$$

其中在这种控制公式中, $Z(\cdot)$ 是已知的 $n \times r$ 回归矩阵。[注意 (6.4.3) 中的标准符号滥用,其中 $K(s)e$ 用于表示 $K(s)$ 与 $e(t)$ 卷积的逆 Laplace 变换。]重要的是要注意,可以选择 $K(s)$ 使得 $Z(\cdot)$ 和 r 不依赖于 \ddot{q} 的测量。实际上,如果选择 $K(s)$ 使得 $H(s)$ 的相对度为 1 [Kailath 1980], $Z(\cdot)$ 和 r 将不依赖于 \ddot{q} 。

本节中给出的自适应控制公式称为无源性方法,因为 $-r \rightarrow Z(\cdot)$ 的映射被构造为无源映射。也就是说,我们构造一个自适应更新规则使得:

$$\int_0^t -r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma \geq -\beta \quad (6.79)$$

对所有时间和某个正标量常数 β 都满足。这一无源性概念用于分析误差系统的稳定性,正如我们将展示的。然而,现在让我们说明 (6.4.4) 在生成自适应更新规则中的使用。

例 6.4-1: 无源性自适应更新规则

让我们证明自适应更新规则:

$$\dot{\hat{\varphi}} = -\Gamma Z^T(\cdot)r \quad (6.80)$$

满足 (6.4.4) 给出的不等式。注意 Γ 如 (6.2.11) 中定义。

首先将 (1) 写成以下形式：

$$\dot{\tilde{\varphi}} = \Gamma Z^T(\cdot)r \quad (6.81)$$

其中我们使用了 Γ 是对角矩阵这一事实。将 (2) 代入 (6.4.4) 得到：

$$\int_0^t -r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma = \int_0^t \tilde{\varphi}^T(\sigma)\Gamma^{-1}\dot{\tilde{\varphi}}(\sigma)d\sigma \quad (6.82)$$

由于 Γ 是常数矩阵，我们可以使用乘积规则将 (3) 重写为：

$$\int_0^t -r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma = \frac{1}{2} \int_0^t \frac{d}{dt} [\tilde{\varphi}^T(\sigma)\Gamma^{-1}\tilde{\varphi}(\sigma)] d\sigma \quad (6.83)$$

或

$$\int_0^t -r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma = \frac{1}{2}\tilde{\varphi}^T(t)\Gamma^{-1}\tilde{\varphi}(t) - \frac{1}{2}\tilde{\varphi}^T(0)\Gamma^{-1}\tilde{\varphi}(0) \quad (6.84)$$

从 (5) 中现在很明显，如果 β 选择为：

$$\beta = \frac{1}{2}\tilde{\varphi}^T(0)\Gamma^{-1}\tilde{\varphi}(0) \quad (6.85)$$

那么 (6.4.4) 中给出的无源性积分对于 (1) 中给出的自适应更新规则是满足的。

现在我们已经了解了如何使用无源性积分 (6.4.4) 来生成自适应更新规则，我们使用无源性的概念来分析一类自适应控制器的稳定性。对于这类自适应控制器，扭矩控制由下式给出：

$$\tau = Z(\cdot)\hat{\varphi} + K_v r \quad (6.86)$$

注意，通过为 $K(s)$ 选择不同的传递函数矩阵，可以从 (6.4.5) 生成许多类型的扭矩控制器。也就是说，对于不同的 $K(s)$ ，我们有不同类型的反馈，因为反馈项 $K_v r$ 会相应地改变。

为了形成误差系统，将机器人动力学 (6.2.1) 用跟踪误差变量 r 和回归矩阵 $Z(\cdot)$ 重写为：

$$M(q)\dot{r} + V_m(q, \dot{q})r = Z(\cdot)\tilde{\varphi} - K_v r \quad (6.87)$$

将 (6.4.5) 代入 (6.4.6) 得到跟踪误差系统：

$$M(q)\dot{r} + V_m(q, \dot{q})r + K_v r = Z(\cdot)\tilde{\varphi} \quad (6.88)$$

为了分析该系统的稳定性, 我们使用 Lyapunov 类函数:

$$V = \frac{1}{2}r^T M(q)r + \beta - \int_0^t r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma \quad (6.89)$$

[Ortega 和 Spong 1988]。注意 V 是正的, 因为参数估计更新规则被构造为保证 (6.4.4)。也就是说, 如果满足 (6.4.4), 那么:

$$\beta - \int_0^t r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma \geq 0 \quad (6.90)$$

因此, V 是正标量函数。对时间微分 (6.4.8) 得到:

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2}r^T \dot{M}(q)r - r^T Z(\cdot)\hat{\varphi} \quad (6.91)$$

将 (6.4.7) 代入 (6.4.10) 得到:

$$\dot{V} = -r^T K_v r + r^T Z(\cdot)\tilde{\varphi} - r^T Z(\cdot)\hat{\varphi} \quad (6.92)$$

利用斜对称性质 (参见第 2 章), 可以将上式写成:

$$\dot{V} = -r^T K_v r - r^T Z(\cdot)\hat{\varphi} \quad (6.93)$$

我们现在详细说明跟踪误差的稳定性类型。首先从 (6.4.12) 可以看出, 我们可以对 \dot{V} 放置新的上界:

$$\dot{V} \leq -\lambda_{\min}\{K_v\}\|r\|^2 - r^T Z(\cdot)\hat{\varphi} \quad (6.94)$$

上式也可以写成:

$$\dot{V} \leq -\lambda_{\min}\{K_v\}\|r\|^2 - \frac{d}{dt} \left(\int_0^t r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma \right) \quad (6.95)$$

将 (6.4.14) 乘以-1 并积分左侧得到:

$$-\int_0^t \dot{V}(\sigma)d\sigma \geq \lambda_{\min}\{K_v\} \int_0^t \|r(\sigma)\|^2 d\sigma + \int_0^t r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma \quad (6.96)$$

由于 \dot{V} 如 (6.4.12) 所描述的是负半定的, 我们可以说 V 是一个非增函数, 上界为 $V(0)$ 。通过回忆 $M(q)$ 是下有界的, 如惯性矩阵的正定性所描述的 (参见第 2 章), 我们可以说 (6.4.8) 中给出的 V 以零为下界。由于 V 是非增的, 上界为 $V(0)$, 下界为 0, 我们可以将 (6.4.15) 写成:

$$V(0) - V(\infty) \geq \lambda_{\min}\{K_v\} \int_0^t \|r(\sigma)\|^2 d\sigma + \int_0^t r^T(\sigma)Z(\sigma)\hat{\varphi}(\sigma)d\sigma \quad (6.97)$$

或

$$V(0) + \beta \geq \lambda_{\min}\{K_v\} \int_0^t \|r(\sigma)\|^2 d\sigma \quad (6.98)$$

(6.4.17) 给出的边界告诉我们 $r \in L_2^n$ (参见第 1 章), 这意味着滤波后的跟踪 r 以 (6.4.17) 给出的“特殊”方式有界。

为了为位置跟踪误差 e 建立稳定性结果, 我们建立位置跟踪误差与滤波跟踪误差 r 之间的传递函数关系。从 (6.4.1) 我们可以说明:

$$e(s) = H(s)r(s) \quad (6.99)$$

其中 $H(s)$ 如 (6.4.2) 中定义。由于 $H(s)$ 是严格真、渐近稳定的传递函数矩阵, $r \in L_2^n$, 我们可以使用第 1 章中的定理 1.4.7 来说明:

$$\lim_{t \rightarrow \infty} e = 0 \quad (6.100)$$

上面的结果告诉我们位置跟踪误差是渐近稳定的。根据上面的理论发展, 我们只能说速度跟踪误差是有界的。

表 6.3: 无源性自适应控制器类

控制器	
扭矩控制	$\tau = Z(\cdot)\hat{\varphi} + K_v r$
无源性条件	$\int_0^t -r^T Z \hat{\varphi} d\sigma \geq -\beta$
跟踪变量	$r(s) = H(s)e(s)$
传递函数	$H(s) = [sI + K(s)]^{-1}$
增益条件	K_v 正定, $H(s)$ 严格真、稳定
稳定性结果	e 渐近稳定, \dot{e} 有界

无源性方法给出了一类扭矩控制律。我们用一些例子来说明这一概念, 这些例子统一了自适应控制中的一些研究。

例 6.4-2: 自适应惯量相关控制器的无源性

在这个例子中, 我们展示了如何使用无源性概念导出自适应惯量相关控制器。首先, 注意通过在表 6.4.1 中定义:

$$K(s) = \Lambda \quad (6.101)$$

和

$$H(s) = [sI + \Lambda]^{-1} \quad (6.102)$$

我们得到扭矩控制律:

$$\tau = Z(\cdot)\hat{\varphi} + K_v r \quad (6.103)$$

其中

$$Z(\cdot)\hat{\varphi} = \hat{M}(q)(\ddot{q}_d + \Lambda\dot{e}) + \hat{V}_m(q, \dot{q})(\dot{q}_d + \Lambda e) + \hat{G}(q) + \hat{F}(\dot{q}) \quad (6.104)$$

这对应于 (6.3.11) 中给出的定义；因此，使用 (2)，我们得到了表 6.3.1 中给出的自适应惯量相关扭矩控制器。

最后一项要检查的是自适应惯量相关更新规则是否满足表 6.4.1 中给出的无源性积分。从表 6.3.1 中，自适应惯量相关更新规则可以写成：

$$\dot{\hat{\varphi}} = \Gamma Z^T(\cdot)r \quad (6.105)$$

对于 (1) 中给出的 $K(s)$ 选择。重新检查例 6.4-1，现在很明显我们已经用无源性方法导出了自适应惯量相关控制器。

例 6.4-3: PID 扭矩控制律

在例 6.4-2 中，扭矩控制律被证明为：

$$\tau = \hat{M}(q)\ddot{q}_d + \hat{V}_m(q, \dot{q})\dot{q} + \hat{G}(q) + K_v r + K_p e \quad (6.106)$$

该扭矩控制器是比例-微分 (PD) 反馈控制器，因为我们在反馈部分使用了 e 和 \dot{e} 。现在希望找到表 6.4.1 中 $K(s)$ 的选择，以给出形式为：

$$\tau = \hat{M}(q)\ddot{q}_d + \hat{V}_m(q, \dot{q})\dot{q} + \hat{G}(q) + K_p e + K_i \int e dt + K_v \dot{e} \quad (6.107)$$

的比例-积分-微分 (PID) 型扭矩控制器。

使用表 6.4.1，我们可以选择：

$$K(s) = \Lambda + \frac{1}{s}\Psi \quad (6.108)$$

其中 Λ 和 Ψ 是对角正定矩阵。现在我们必须验证 $H(s)$ 是否确实是严格真、稳定的传递函数矩阵。对于这种 $K(s)$ 选择，我们可以写出：

$$H(s) = \text{diag} \left[\frac{s}{s^2 + s\lambda_1 + \psi_1}, \dots, \frac{s}{s^2 + s\lambda_n + \psi_n} \right] \quad (6.109)$$

由于 λ_i 和 ψ_i 是正的， $H(s)$ 是严格真、稳定的传递函数矩阵。

一般自适应更新规则

如前所述, 表 6.4.1 中概述的自适应控制方案允许通过确保所提出的更新满足 (6.4.4) 中给出的无源性积分来制定不同的自适应更新律。Landau 提出了满足无源性积分的一般更新规则:

$$\dot{\hat{\phi}} = -\Gamma Z^T(\cdot)r - F_p Z^T(\cdot)r - \int_0^t F_I(t-\sigma)Z^T(\sigma)r(\sigma)d\sigma \quad (6.110)$$

其中 F_p 是 $r \times r$ 正定常数矩阵, $F_I(t)$ 是其 Laplace 变换是具有 $s = 0$ 处极点的正实传递函数矩阵的 $r \times r$ 正定矩阵核 [Landau 1979]。

通过利用这一一般更新律, 可以设计多种类型的自适应。我们需要记住的只是 $F_I(t)$ 和 F_p 必须满足的条件。(6.4.20) 直接得出的一个可能自适应方案是比例 + 积分 (PI) 自适应方案。PI 更新律与 (6.4.20) 给出的相同, 其中:

$$F_I(t) = K_I \quad (6.111)$$

K_I 是对角常数正定矩阵。在 [Landau 1979] 中指出, 关于自适应模型跟随, PI 自适应比积分自适应显示出显著的改进。因此, 这种自适应类型可能有益于机器人机械臂的跟踪控制。

6.5 持续激励

对于前几节中提出的自适应控制器, 跟踪误差已被证明是渐近稳定的; 然而, 关于参数误差只能说它是有界的。一般来说, 只有当回归矩阵满足某些条件时, 自适应控制系统中才会发生参数识别。具体来说, 几位研究人员 [Morgan 和 Narendra 1977]、[Anderson 1977] 研究了类似于我们在本章中提出的自适应控制系统的渐近稳定性。例如, 如果回归矩阵 $Y(\cdot)$ 满足:

$$\int_t^{t+\rho} Y^T(q, \dot{q})Y(q, \dot{q})d\sigma \geq \alpha I_r \quad (6.112)$$

对所有 $t \geq 0$, 其中 α 、 β 和 ρ 都是正标量。此外, 由于跟踪误差是渐近稳定的, 我们可以将 (6.5.1) 重写为:

$$\int_t^{t+\rho} Y^T(q_d, \dot{q}_d)Y(q_d, \dot{q}_d)d\sigma \geq \alpha I_r \quad (6.113)$$

其中参数 q 和 \dot{q} 已分别被 q_d 和 \dot{q}_d 替代。

(6.5.2) 中给出的条件告诉我们, 如果 $Y(\cdot)$ 在 ρ 给出的区间上变化足够大, 使得整个 r 维参数空间被跨越, 我们就知道参数误差收敛到零。这相当于对期望轨迹的条件, 使得所有参数在经过足够的学习间隔后都会被识别。这一条件有助于制定期望轨迹, 以

确保识别诸如摩擦系数或负载质量之类的参数。我们现在用一些例子来说明持续激励轨迹的含义。

例 6.5-1: 单连杆机器人机械臂缺乏持续激励

我们希望研究图 6.5.1 所示单连杆机器人机械臂的持续激励条件。该机器人机械臂的动力学取为:

$$\tau = m\ddot{q} + b\dot{q} \quad (6.114)$$

其中项 b 用于表示表示动态摩擦系数的正标量。我们假设该机器人机械臂在不受重力影响的平面内, m 和 b 是未知的正常数。

a. 自适应控制器

通过使用表 6.3.1, 动力学 (1) 的自适应惯量相关控制器可以证明由下式给出:

$$\tau = Y_{11}\hat{m} + Y_{12}\hat{b} + k_v r \quad (6.115)$$

在扭矩控制表达式中, 回归矩阵 $Y(\cdot)$ 由下式给出:

$$Y(\cdot) = [Y_{11}, Y_{12}] \quad (6.116)$$

其中

$$Y_{11} = \ddot{q}_d + \lambda \dot{e} \quad (6.117)$$

$$Y_{12} = \dot{q} \quad (6.118)$$

相应的参数估计向量为:

$$\hat{\varphi} = \begin{bmatrix} \hat{m} \\ \hat{b} \end{bmatrix} \quad (6.119)$$

我们可以按照表 6.3.1 制定自适应更新规则为:

$$\dot{\hat{m}} = \gamma_1 Y_{11} r \quad (6.120)$$

和

$$\dot{\hat{b}} = \gamma_2 Y_{12} r \quad (6.121)$$

b. 持续激励

对于这个自适应控制器, 希望从分析上证明 $q_d = 1 - e^{-2t}$ 不是持续激励的。从 (6.5.2) 中, 这个例子的持续激励条件的被积函数由下式给出:

$$Y^T(\cdot)Y(\cdot) = \begin{bmatrix} (\ddot{q}_d + \lambda\dot{q}_d)^2 & \dot{q}_d(\ddot{q}_d + \lambda\dot{q}_d) \\ \dot{q}_d(\ddot{q}_d + \lambda\dot{q}_d) & \dot{q}_d^2 \end{bmatrix} \quad (6.122)$$

或

$$Y^T(\cdot)Y(\cdot) = \begin{bmatrix} 4e^{-4t} & 2e^{-2t}(1 - e^{-2t}) \\ 2e^{-2t}(1 - e^{-2t}) & (1 - e^{-2t})^2 \end{bmatrix} \quad (6.123)$$

将 $Y^T(\cdot)Y(\cdot)$ 的第一列乘以 2 并加到第二列得到：

$$R_{YY} = \begin{bmatrix} 4e^{-4t} & 2e^{-2t} \\ 2e^{-2t}(1 - e^{-2t}) & 1 \end{bmatrix} \quad (6.124)$$

由于矩阵 R_{YY} 与矩阵 $Y^T(\cdot)Y(\cdot)$ 具有相同的值域空间，我们可以从 (7) 中看出，矩阵 $Y^T(\cdot)Y(\cdot)$ 的值域空间总是一维的；因此，持续激励条件对于 $q_d = 1 - e^{-2t}$ 不成立。

例 6.5-2：单连杆机械臂的持续激励

a. 单频期望轨迹

使用例 6.5-1 中的自适应控制器，我们想通过仿真证明 $q_d = \sin t$ 不是持续激励的。对于 $m = 1 \text{ kg}$ 和 $b = 1 \text{ N-m-s}$ ，在：

$$k_v = 10, \quad \lambda = 5, \quad \gamma_1 = \gamma_2 = 10 \quad (6.125)$$

$$\hat{m}(0) = \hat{b}(0) = 0 \quad (6.126)$$

的初始条件下，对自适应惯量相关控制器进行了仿真。跟踪误差和参数估计如图 6.5.2 所示。从图中可以看出，正如预期的那样，跟踪误差是渐近稳定的，参数估计保持有界。注意 \hat{m} 和 \hat{b} 不趋于零，因为期望轨迹不是持续激励的。

b. 多频期望轨迹

使用例 6.5-1 中的自适应控制器，我们希望仿真证明 $q_d = \sin t + \cos 3t$ 是持续激励的。例 6.5-1 中给出的自适应控制器应在与本例 (a) 部分相同的条件下仿真，只是改变期望轨迹。跟踪误差和参数误差如图 6.5.3 所示。如图所示，跟踪误差是渐近稳定的，参数估计保持有界。注意 \hat{b} 和 \hat{m} 分别收敛到 b 和 m 的精确值。这是因为期望轨迹是持续激励的。

6.6 复合自适应控制器

在自适应计算力矩和自适应惯量相关控制策略中，我们已经证明跟踪误差是渐近稳定的，参数误差是有界的。然后说明如果持续激励条件成立，参数误差收敛到零。在某些机器人应用中，利用持续激励轨迹可能是不切实际的；因此，我们有动力重新设计自适应控制策略以实现参数识别。

在本节中，我们展示了如何修改第 6.3 节中给出的自适应控制器，以确保跟踪误差和参数误差都渐近收敛 [Slotine 和 Li 1985(b)]。参数误差的渐近收敛被证明在滤波回归矩阵的条件下成立。这个条件通常称为无限积分条件，比持续激励条件限制更少。

设计新自适应控制器的过程可以概述如下。首先，从扭矩测量形成滤波回归矩阵。其次，展示如何使用这个滤波回归矩阵来制定未知参数的最小二乘估计器。最后，修改表 6.3.1 中的自适应更新规则以包含额外的最小二乘估计器项。

扭矩滤波

我们现在展示如何从扭矩测量形成的回归矩阵可以滤波以消除加速度测量的需要。从第 6.2 节我们可以以以下形式写出机器人方程 (6.2.1)：

$$\tau = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = W(q, \dot{q}, \ddot{q})\varphi \quad (6.127)$$

(6.6.1) 中的中间表达式写成以下形式：

$$\tau = h(q, \dot{q}, \ddot{q}) + g(q, \dot{q}) \quad (6.128)$$

其中

$$h(q, \dot{q}, \ddot{q}) = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} \quad (6.129)$$

和

$$g(q, \dot{q}) = G(q) + F(\dot{q}) \quad (6.130)$$

以 (6.6.2) 给出的形式写出机器人方程的原因是，这个方程现在以允许 \ddot{q} 被滤波掉或移除的方式分离。也就是说，通过滤波 (6.6.2)，我们有：

$$\tau_f = f * h + g_f \quad (6.131)$$

其中 f 是线性稳定、严格真滤波器的脉冲响应，* 用于表示卷积运算。例如，我们可以使用由以下给出的一阶滤波器：

$$f(s) = \frac{a}{s + a} \quad (6.132)$$

其中 a 是正标量常数。通过使用卷积的性质：

$$f * h = \dot{f} * (\dot{q} * M(q)) + f * [V_m(q, \dot{q})\dot{q}] \quad (6.133)$$

我们可以将 (6.6.5) 重写为：

$$\tau_f = \dot{f} * (M(q)\dot{q}) + f * [V_m(q, \dot{q})\dot{q}] + g_f \quad (6.134)$$

在代入 h 和 g 的表达式后, 注意 \ddot{q} 已被滤波掉。也就是说, 滤波扭矩的显式表达式由下式给出:

$$\tau_f = \tilde{f}(q, \dot{q})\varphi \quad (6.135)$$

其中 \tilde{f} 是适当、稳定滤波器的脉冲响应; 例如,

$$\tilde{f}(s) = \frac{as}{s+a} \quad (6.136)$$

根据线性, 未知参数仍然可以与 (6.6.9) 分离。也就是说, (6.6.9) 可以重写为:

$$\tau_f = W_f(\cdot)\varphi \quad (6.137)$$

其中 $W_f(\cdot)$ 是 $n \times r$ 滤波回归矩阵, φ 是 $r \times 1$ 未知参数向量。我们现在用一个例子来说明如何使用扭矩滤波来消除加速度测量的需要。

例 6.6-1: 单连杆机器人机械臂的扭矩滤波

使用例 6.5-1 中给出的单连杆机器人机械臂动力学, 希望找到 (6.6.11) 中给出的滤波回归矩阵 $W_f(q, \dot{q})$, 其中线性滤波器由下式给出:

$$f(s) = \frac{1}{s+1} \quad (6.138)$$

或, 在时域中,

$$f(t) = e^{-t} \quad (6.139)$$

使用 (6.6.9), 单连杆机械臂的滤波扭矩表达式由下式给出:

$$\tau_f = \tilde{f} * (m\dot{q}) + f * (b\dot{q}) \quad (6.140)$$

其中

$$\tilde{f}(s) = \frac{s}{s+1} \quad (6.141)$$

(3) 中的表达式用于将已知函数与未知常数分离成以下形式:

$$\tau_f = W_f(\cdot)\varphi \quad (6.142)$$

其中滤波回归矩阵和参数向量为:

$$W_f(\cdot) = [f * \dot{q}, f * \dot{q}] \quad (6.143)$$

和

$$\varphi = \begin{bmatrix} m \\ b \end{bmatrix} \quad (6.144)$$

最小二乘估计

最小二乘估计方法已在许多类型的参数识别方案中使用 [Astrom 和 Wittenmark 1989]。事实证明，即使期望轨迹不是持续激励的，这种类型的估计方法也能提取最大量的参数信息。这是为机器人机械臂设计自适应控制系统时要认识到的一个重要事实，因为在许多机器人应用中，持续激励条件将不成立。因此，最小二乘估计为机器人机械臂的自适应控制器设计提供了一个有吸引力的解决方案。

我们现在展示如何使用最小二乘估计方法来生成自适应更新规则。首先，定义最小二乘更新规则：

$$\dot{\hat{\varphi}} = -PW_f^T(\cdot)\tau_f \quad (6.145)$$

其中

$$\dot{P} = -PW_f^T(\cdot)W_f(\cdot)P \quad (6.146)$$

P 是 $r \times r$ 时变对称矩阵。

通过这种最小二乘估计方法，如果“无限积分”条件成立，参数误差收敛到零。具体来说，如果：

$$\lim_{t \rightarrow \infty} \lambda_{\min} \left\{ \int_0^t W_f^T(\cdot)W_f(\cdot)d\sigma \right\} = \infty \quad (6.147)$$

成立，那么：

$$\lim_{t \rightarrow \infty} \tilde{\varphi} = 0 \quad (6.148)$$

如 [Slotine 和 Li 1985(b)] 中指出的，这一无限积分条件比持续激励条件限制更少。也就是说，在实际机器人应用中，(6.6.18) 通常可以被验证。

例 6.6-2：单连杆机器人机械臂的最小二乘估计器

使用例 6.5-1 中给出的单连杆机器人机械臂动力学，希望找到 (6.6.16) 和 (6.6.17) 中给出的最小二乘估计器。由于未知参数的数量为两个，定义矩阵 P 为：

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (6.149)$$

利用例 6.6-1 中的滤波回归矩阵，我们有：

$$PW_f^T(\cdot) = \begin{bmatrix} p_1(\dot{f} * \dot{q}) + p_2(f * \dot{q}) \\ p_2(\dot{f} * \dot{q}) + p_3(f * \dot{q}) \end{bmatrix} \quad (6.150)$$

其中

$$W_f(\cdot) = [\dot{f} * \dot{q}, f * \dot{q}] \quad (6.151)$$

使用 (6.6.17), 很容易看出矩阵 P 应该以下列方式更新:

$$\dot{p}_1 = -[p_1(\dot{f} * \dot{q}) + p_2(f * \dot{q})]^2 \quad (6.152)$$

$$\dot{p}_2 = -[p_1(\dot{f} * \dot{q}) + p_2(f * \dot{q})][p_2(\dot{f} * \dot{q}) + p_3(f * \dot{q})] \quad (6.153)$$

和

$$\dot{p}_3 = -[p_2(\dot{f} * \dot{q}) + p_3(f * \dot{q})]^2 \quad (6.154)$$

现在使用 (6.6.16), 参数更新规则为:

$$\dot{m} = -[p_1(\dot{f} * \dot{q}) + p_2(f * \dot{q})]\tau_f \quad (6.155)$$

和

$$\dot{b} = -[p_2(\dot{f} * \dot{q}) + p_3(f * \dot{q})]\tau_f \quad (6.156)$$

其中, 从 (6.6.16) 中, τ_f 由下式给出:

$$\tau_f = m(\dot{f} * \dot{q}) + b(f * \dot{q}) \quad (6.157)$$

复合自适应控制器

复合自适应控制器与表 6.3.1 中给出的控制器相同, 只是自适应更新规则的修改。这一修改由下式给出:

$$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot)r - PW_f^T(\cdot)\tau_f \quad (6.158)$$

为了证明跟踪误差和参数误差都收敛到零, 从 Lyapunov 类函数开始:

$$V = \frac{1}{2}r^T M(q)r + \frac{1}{2}\tilde{\varphi}^T P^{-1}\tilde{\varphi} \quad (6.159)$$

对时间微分 (6.6.29) 得到:

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2}r^T \dot{M}(q)r + \tilde{\varphi}^T P^{-1}\dot{\tilde{\varphi}} + \frac{1}{2}\tilde{\varphi}^T \dot{P}^{-1}\tilde{\varphi} \quad (6.160)$$

从表 6.3.1 中给出的控制律和第 6.3 节中的发展, 我们可以形成跟踪误差系统:

$$M(q)\dot{r} + V_m(q, \dot{q})r = Y(\cdot)\tilde{\varphi} - K_v r \quad (6.161)$$

将 (6.6.31) 代入 (6.6.30) 得到:

$$\dot{V} = -r^T K_v r + \tilde{\varphi}^T (Y^T(\cdot)r + P^{-1}\dot{\tilde{\varphi}} + \frac{1}{2}\dot{P}^{-1}\tilde{\varphi}) \quad (6.162)$$

在将 (6.6.21) 中的 \dot{P}^{-1} 、(6.6.28) 中的和 (6.6.15) 中的 τ_f 代入 (6.6.32) 后, 我们有:

$$\dot{V} = -r^T K_v r - \frac{1}{2}\tilde{\varphi}^T W_f^T(\cdot)W_f(\cdot)\tilde{\varphi} \quad (6.163)$$

我们现在详细说明跟踪误差和参数误差的稳定性类型。首先, 由于 (6.6.33) 中的 \dot{V} 至少是负半定的, 形式为:

$$\dot{V} \leq -r^T K_v r \quad (6.164)$$

我们可以说 (6.6.29) 中的 V 是有界的。由于 V 是有界的, $M(q)$ 是正定矩阵, P^{-1} 满足 (6.6.27) 给出的条件, 我们可以说 r 和 $\tilde{\varphi}$ 是有界的。此外, 从 (6.3.8) 中给出的 r 的定义, 我们可以使用标准线性控制论证来说明 e 和 \dot{e} (因此 q 和 \dot{q}) 是有界的。我们现在可以使用第 6.4 节中提出的相同论证来说明 \dot{r} 是有界的。给定 \dot{r} 是有界的, 我们可以通过建立位置跟踪误差与滤波跟踪误差 r 之间的传递函数关系来确定位置跟踪误差的稳定性结果。从 (6.3.8) 中, 我们可以说明:

$$e(s) = G(s)r(s) \quad (6.165)$$

其中 s 是 Laplace 变换变量,

$$G(s) = [sI + \Lambda]^{-1} \quad (6.166)$$

I 是 $n \times n$ 单位矩阵, Λ 是 $n \times n$ 正定矩阵。由于 $G(s)$ 是严格真、渐近稳定的传递函数, $r \in L_2^n$, 我们可以使用第 1 章中的定理 1.4.7 来说明:

$$\lim_{t \rightarrow \infty} e = 0 \quad (6.167)$$

其次, 由于 \dot{V} 至少是负半定的, 我们知道 V 必须是非增的, 因此以 $V(0)$ 为上界。此外, 通过无限积分假设, 我们在 (6.6.27) 中得出结论:

$$\lim_{t \rightarrow \infty} \lambda_{\min}\{P^{-1}\} = \infty \quad (6.168)$$

由于 (6.6.29) 中给出的 V 中的项 $\frac{1}{2}\tilde{\varphi}^T P^{-1}\tilde{\varphi}$ 以 $V(0)$ 为上界, 我们可以看到为了使 (6.6.38) 成立, 我们必须有:

$$\lim_{t \rightarrow \infty} \tilde{\varphi} = 0 \quad (6.169)$$

因此, 根据上面的论证, 位置跟踪误差和参数误差对于表 6.6.1 中概述的复合自适应控制器都是渐近稳定的。根据上面的理论发展, 我们只能说速度跟踪误差是有界的; 然而, 可以调用 Barbalat 引理来说明速度跟踪误差也是渐近稳定的。

表 6.4: 复合自适应控制器

控制器	
扭矩控制	$\tau = Y(\cdot)\hat{\varphi} + K_v r$
自适应更新规则	$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot)r - P W_f^T(\cdot)\tau_f$
辅助信号	$r = \dot{e} + \Lambda e$
滤波器	$f(s) = \frac{a}{s+a}$
增益条件	K_v, Λ, Γ 正定对角, $a > 0$
稳定性结果	$e, \tilde{\varphi}$ 渐近稳定, \dot{e} 有界

例 6.6-3: 单连杆机器人机械臂的复合自适应控制器

希望为图 6.5.1 所示的单连杆机器人机械臂制定复合自适应控制器。扭矩控制律与例 6.5-1 中的方程 (2) 给出的相同; 因此, 需要做的只是复合自适应更新规则的制定。从表 6.6.1 中, 复合参数更新规则为:

$$\dot{\hat{m}} = \gamma_1 Y_{11} r - (p_1 W_{f11} + p_2 W_{f12}) \tau_f \quad (6.170)$$

和

$$\dot{\hat{b}} = \gamma_2 Y_{12} r - (p_2 W_{f11} + p_3 W_{f12}) \tau_f \quad (6.171)$$

其中 r 在 (6.3.8) 中定义, Y_{11}, Y_{12} 在例 6.5-1 中定义, $p_1, p_2, p_3, W_{f11}, W_{f12}, \tau_f$ 在例 6.6-2 中定义。

6.7 自适应控制器的鲁棒性

所有讨论的自适应控制方案都确保机器人机械臂动力学的期望参考轨迹的渐近跟踪; 然而, 在现实中, 我们知道在任何机电系统中总会有干扰。考虑某种干扰效应的一种简单方法是在机械臂动力学方程中添加有界干扰项。有了这个附加干扰项, 机器人方程变为:

$$\tau = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + T_d \quad (6.172)$$

其中 T_d 是 $n \times 1$ 向量, 表示附加干扰。

应用自适应惯量相关控制策略并忽略 (6.7.1) 中的 T_d 项, 得到表 6.3.1 的自适应控制方案。然而, 如果我们重新检查第 6.3 节中对自适应惯量相关控制器的稳定性分析, 我们可以看到有界干扰项给我们提供了跟踪误差的不同类型的稳定性结果。具体来说, 在 (6.7.1) 中添加了有界干扰项后, (6.3.7) 中 Lyapunov 函数的导数变为:

$$\dot{V} = -r^T K_v r + r^T T_d \quad (6.173)$$

从 (6.7.2) 中很明显, \dot{V} 不能再被认为是负半定的。从我们之前使用 Lyapunov 稳定性理论的经验来看, 希望 \dot{V} 是“负的”; 因此, 找到 (6.7.2) 中 \dot{V} 为负的区域是有利的。通过使用 Rayleigh-Ritz 定理 (参见第 1 章), 我们可以将 (6.7.2) 写成:

$$\dot{V} \leq -\lambda_{\min}\{K_v\}\|r\|^2 + \|r\|\|T_d\| \quad (6.174)$$

从 (6.7.3) 中, 可以获得 \dot{V} 为负的充分条件。也就是说, 如果:

$$\|r\| > \frac{\|T_d\|}{\lambda_{\min}\{K_v\}} \quad (6.175)$$

\dot{V} 将为负。

如果满足 (6.7.4), \dot{V} 为负, V 将减小。如果 V 减小, 那么从 (6.3.7) 中给出的 Lyapunov 函数定义, r 必须最终减小。然而, 如果 r 减小使得:

$$\|r\| \leq \frac{\|T_d\|}{\lambda_{\min}\{K_v\}} \quad (6.176)$$

那么 \dot{V} 可能为正, 这意味着 V 将开始增加。如果 V 开始增加, 我们通过检查两种可能性来深入了解问题。一, V 的增加导致 r 增加, 使得 (6.7.4) 被满足。这意味着 V 将开始减小, 因此 r 将最终减小。如果 r 以这种方式持续增加和减小, 那么 r 和 $\tilde{\varphi}$ 都保持有界。另一种可能性是 V 的增加导致 $\tilde{\varphi}$ 增加, 而 r 保持足够小使得 (6.7.5) 被满足。在这种情况下, V 保持为正; 因此, $\tilde{\varphi}$ 可以继续增加。如果 V 以这种方式继续增加, r 是有界的; 然而, $\tilde{\varphi}$ 因此 $\hat{\varphi}$ 都变得无界。

上面的论证揭示了自适应惯量相关控制律中的参数估计在存在有界干扰时可能变得不稳定。也就是说, 在假设机器人模型由 (6.7.1) 给出的情况下, 参数估计可能发散。如果参数估计变得太大, 我们可以从表 6.3.1 中看到输入扭矩将开始增长并可能使关节电机饱和; 因此, 修改自适应控制器以消除扭矩饱和的可能性是可取的。

例 6.7-1: 干扰对自适应控制的影响

在本例中, 我们仿真例 6.3-1 中给出的相同自适应控制器, 具有相同的控制参数、初始条件和期望轨迹; 然而, 我们在两连杆机械臂动力学中添加了由以下给出的干扰项:

$$T_{d1} = 2, \quad T_{d2} = 1 \quad (6.177)$$

跟踪误差和参数估计如图 6.7.1 所示。从图中可以看出，对于 (1) 中给出的干扰，参数估计不会变得无界；然而，跟踪误差不再是渐近稳定的。

基于扭矩的干扰抑制方法

为了抑制机器人模型中的附加干扰项，我们说明了如果扭矩控制修改为：

$$\tau = \tau_a + K_d \text{sgn}(r) \quad (6.178)$$

其中 τ_a 是表 6.3.1 中给出的扭矩控制，

$$K_d = \text{diag}\{k_{d1}, k_{d2}, \dots, k_{dn}\} \quad (6.179)$$

$\text{sgn}(\cdot)$ 用于表示符号函数， k_d 是满足：

$$k_{di} > |T_{di}|_{\max} \quad (6.180)$$

的标量常数， T_{di} 表示 $n \times 1$ 向量 T_d 的第 i 个分量 [Slotine 和 Li 1985(c)]。

应用 (6.7.6) 中给出的干扰抑制控制器，(6.3.7) 中 Lyapunov 函数的导数变为：

$$\dot{V} = -r^T K_v r + r^T T_d - r^T K_d \text{sgn}(r) \quad (6.181)$$

通过注意：

$$|r^T T_d| \leq \sum_{i=1}^n |r_i| |T_{di}|_{\max} \quad (6.182)$$

和

$$r^T K_d \text{sgn}(r) = \sum_{i=1}^n k_{di} |r_i| \quad (6.183)$$

(6.7.9) 可以写成：

$$\dot{V} \leq -r^T K_v r - \sum_{i=1}^n (k_{di} - |T_{di}|_{\max}) |r_i| \quad (6.184)$$

通过利用 (6.7.8)，我们可以将 (6.7.11) 写成：

$$\dot{V} \leq -r^T K_v r \quad (6.185)$$

第 6.6 节中的相同论证可以用来证明位置跟踪误差是渐近稳定的，而速度跟踪误差和参数估计是有界的。

例 6.7-2: 两连杆机器人机械臂的干扰抑制

在本例中，我们仿真 (6.7.6) 中给出的改进自适应控制器，具有与例 6.3-1 中相同的控制参数、初始条件和期望轨迹，以及例 6.7-1 中给出的干扰。改进的扭矩控制器由下式给出：

$$\tau_1 = \tau_{a1} + k_d \text{sgn}(r_1) \quad (6.186)$$

和

$$\tau_2 = \tau_{a2} + k_d \text{sgn}(r_2) \quad (6.187)$$

其中 τ_{a1} 和 τ_{a2} 是例 6.3-1 中给出的相同自适应扭矩控制器， λ_1 和 λ_2 是例 6.3-1 中定义的共同标量常数，和

$$k_d = 3 \quad (6.188)$$

注意 k_d 已被选择为满足 (6.7.8) 中给出的条件，并且更新律与例 6.3-1 中给出的相同。跟踪误差和参数估计如图 6.7.2 所示。从图中我们注意到参数估计保持有界；此外，即使在存在干扰的情况下，跟踪误差现在也是渐近稳定的。重要的是要注意，对于上述基于扭矩的干扰抑制方法的理论发展，我们只保证速度跟踪误差是有界的。

基于估计器的干扰抑制方法

在 [Reed 和 Ioannou 1988] 中，引入了 σ -修改 [Ioannou 和 Tsakalis 1985] 的修改版本到自适应惯量相关控制算法 [Slotine 和 Li 1985(a)]，以补偿机器人模型中的未建模动力学和有界干扰。在这种方法中，扭矩控制与表 6.3.1 中给出的相同；然而，自适应更新规则修改为：

$$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot) r - \sigma_s \Gamma \hat{\varphi} \quad (6.189)$$

其中

$$\sigma_s = \begin{cases} 0 & \text{if } \|\hat{\varphi}\| \leq \varphi_0 \\ \sigma_0 \left(\frac{\|\hat{\varphi}\|}{\varphi_0} - 1 \right) & \text{if } \varphi_0 \leq \|\hat{\varphi}\| \leq 2\varphi_0 \\ \sigma_0 & \text{if } \|\hat{\varphi}\| \geq 2\varphi_0 \end{cases} \quad (6.190)$$

和

$$\sigma_0 \geq \frac{\|T_d\|^2}{\varphi_0^2 \lambda_{\min}\{K_v\} \lambda_{\min}\{\Gamma\}} \quad (6.191)$$

通过更新规则 (6.7.13), Reed 证明了跟踪误差可以被限制在一个残差集合内, 所有闭环信号都是有界的。

正如我们所展示的, 有界干扰可能导致参数估计变得不稳定。(6.7.13) 中给出的更新规则旨在通过在线调节参数估计的大小来解决这个问题。这是通过标量设计常数 φ_0 完成的。也就是说, 通过检查参数估计的大小与 φ_0 , 使用这个新的更新规则, 参数估计被迫保持有界。可以清楚地看到, 如果 $\|\hat{\varphi}\| < \varphi_0$, 更新规则与表 6.3.1 中给出的相同。换句话说, 如果参数估计没有变得太大, 控制器与自适应惯量相关控制器相同。另一方面, 如果参数估计变得太大, 修改自适应更新规则以确保参数估计保持有界。我们现在讨论这个 σ -修改如何完成这一任务。

为了激励 (6.7.13) 中给出的更新规则的制定, 检查 $\|\hat{\varphi}\| > \varphi_0$ 的情况。这是更新规则的稳定部分。也就是说, 如果参数估计变得太大, 更新规则切换为:

$$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot)r - \sigma_0 \Gamma \hat{\varphi} \quad (6.192)$$

或就参数误差而言,

$$\dot{\tilde{\varphi}} = -\Gamma Y^T(\cdot)r - \sigma_0 \Gamma \tilde{\varphi} + \sigma_0 \Gamma \varphi \quad (6.193)$$

我们现在重新检查第 6.3 节中对自适应惯量相关控制器的稳定性分析, 其中参数更新规则由 (6.7.16) 给出。具体来说, 在 (6.7.1) 中添加了有界干扰项后, (6.3.7) 中 Lyapunov 函数的导数变为:

$$\dot{V} = -r^T K_v r - \sigma_0 \tilde{\varphi}^T \Gamma \tilde{\varphi} + \sigma_0 \tilde{\varphi}^T \Gamma \varphi + r^T T_d \quad (6.194)$$

或

$$\dot{V} = -x^T K_s x + x^T h \quad (6.195)$$

其中

$$x = \begin{bmatrix} r \\ \tilde{\varphi} \end{bmatrix}, \quad K_s = \begin{bmatrix} K_v & O_{n \times r} \\ O_{r \times n} & \sigma_0 \Gamma \end{bmatrix}, \quad h = \begin{bmatrix} T_d \\ \sigma_0 \Gamma \varphi \end{bmatrix} \quad (6.196)$$

通过使用 Rayleigh-Ritz 定理 (参见第 1 章), 我们可以将 (6.7.19) 写成:

$$\dot{V} \leq -\lambda_{\min}\{K_s\} \|x\|^2 + \|x\| \|h\| \quad (6.197)$$

从 (6.7.21) 中, 如果:

$$\|x\| > \frac{\|h\|}{\lambda_{\min}\{K_s\}} \quad (6.198)$$

\dot{V} 将为负。

重要的是要注意 (6.7.22) 的右侧是常数；因此，如果满足 (6.7.22)， \dot{V} 为负，这导致 V 减小。如果 V 减小，那么从 (6.3.7) 中给出的 Lyapunov 函数定义， x 必须最终减小。然而，如果 x 减小使得：

$$\|x\| \leq \frac{\|h\|}{\lambda_{\min}\{K_s\}} \quad (6.199)$$

那么 \dot{V} 可能为正，这意味着 V 将开始增加。 V 的增加导致 x 增加，使得 (6.7.22) 被满足。这意味着 V 现在开始再次减小，因此 x 最终减小。这个论证说明了 x 是如何有界的。如果 x 是有界的，那么从 (6.7.20) 中， r 和 $\dot{\varphi}$ 是有界的。由于 r 是有界的，标准线性控制论证可以用来证明 e 和 \dot{e} 是有界的。

关于自适应更新规则 (6.7.13) 中的区域 $\varphi_0 \leq \|\hat{\varphi}\| \leq 2\varphi_0$ ，现在讨论最后一点。自适应更新规则的这一部分用于确保自适应惯量相关更新规则和 (6.7.16) 中给出的更新规则的稳定部分之间存在平滑过渡。也就是说，这确保了我们在参数估计中获得任何不连续性，这可能导致输入扭矩中的大不连续性。输入扭矩信号中的大不连续性是不可取的，因为这种类型的信号可能导致机器人机械臂剧烈抖动。

6.8 总结

在本章中，给出了刚性机器人几种最新自适应控制结果的描述。意图是为机器人机械臂不断增长的自适应控制结果列表提供一些视角。诸如瞬态行为、数字实现和对未建模动力学的鲁棒性等研究领域无疑将在未来得到解决。一个仍有待调查的问题是比较不同伺服和自适应律的优缺点。

关于机器人机械臂的其他研究人员的一些优秀自适应控制工作在 [Ortega 和 Spong 1988] 中概述。由于这是一个研究如此充分的领域，本章中可用空间有限，我们对任何被遗漏的人表示歉意。

参考文献

1. [Anderson 1977] Anderson, B., “Exponential stability of linear equations arising in adaptive identification,” *IEEE Trans. Autom. Control*, Feb. 1977, pp. 83–88.
2. [Arimoto and Miyazaki 1986] Arimoto, S., and F. Miyazaki, “Stability and robustness of PD feedback control with gravity compensation for robot manipulators,”

- Robot. Theory Pract.*, DSC Vol. 3, pp. 67–72, ASME Winter Annual Meeting, Dec. 1986.
3. [Åström and Wittenmark 1989] K. Åström and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.
 4. [Brogliato et al. 1990] Brogliato, B., I. Landau, and R. Lozano-Leal, “Adaptive motion control of robot manipulators: a unified approach based on passitivity,” *Proc. IEEE Am. Controls Conf*, pp. 2259–2264, San Diego, CA, May 1990.
 5. [Craig 1985] Craig, J., *Adaptive Control of Mechanical Manipulators*. Reading, MA.: Addison Wesley, 1985.
 6. [Ioannou and Tsakalis 1985] Ioannou, P., and K. Tsakalis, “A robust direct adaptive controller,” *IEEE Trans. Autom. Control*, vol. AC-31, no. 11, pp. 1033–1043, 1985.
 7. [Kailath 1980] Kailath, T, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1980.
 8. [Landau 1979] Landau, Y., *Adaptive Control: The Model Reference Approach*. New York: Marcel Dekker, 1979.
 9. [Li and Slotine 1987] Li, W., and J. Slotine, “Parameter estimation strategies for robotic applications,” ASME Winter Annual Meeting, Boston, 1987.
 10. [Morgan and Narendra 1977] Morgan, A., and K. Narendra, “On the uniform asymptotic stability of certain linear nonautonomous differential equations,” *SIAM J. Control Optim.*, 1977, p. 15.
 11. [Ortega and Spong 1988] Ortega, R., and M. Spong, “Adaptive motion control of rigid robots: a tutorial,” *Proc. IEEE Conf Decision Control*, Austin, TX, 1988.
 12. [Reed and Ioannou 1988] Reed, J., and P. Ioannou, “Instability analysis and robust adaptive control of robotic manipulators,” *Proc. IEEE Conf Decision Control*, Austin, TX, 1988.
 13. [Slotine 1988] Slotine, J., (1988), “Putting physics in control: the example of robotics,” *Control Syst. Mag.*, Dec. 1988, Vol. 8, pp 12–15.
 14. [Slotine and Li 1985(a)] Slotine, J., and W. Li, “Theoretical issues in adaptive control,” 5th Yale Workshop on Applications of Adaptive Systems Theory, Yale University, New Haven, CT, 1985(a).

15. [Slotine and Li 1985(b)] Slotine, J., and W. Li, "Adaptive robot control: a new perspective," *Proc. IEEE Conf Decision Control*, Los Angeles, 1985(b).
16. [Slotine and Li 1985(c)] Slotine, J., and W. Li, "Adaptive strategies in constrained manipulation," *Proc. IEEE Int. Conf Robot. Autom.*, Raleigh, NC, Mar. 1985(c), pp. 595–601.
17. [Vidyasagar 1978] Vidyasagar, M., *Nonlinear Systems Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1978.

习题

第 6.2 节

- 6.2-1 为第 2 章中给出的两连杆极坐标机器人机械臂设计和仿真表 6.2.1 中给出的自适应计算力矩控制器。
- 6.2-2 找到与例 6.2-2 中给出的不同的正定、对称矩阵 P 和 Q ，满足：

$$A^T P + P A = -Q$$

其中

$$A = \begin{bmatrix} O_n & I_n \\ -K_p & -K_v \end{bmatrix}$$

K_p 、 K_v 是对角正定矩阵， O_n 、 I_n 分别表示 $n \times n$ 零矩阵和 $n \times n$ 单位矩阵。

- 6.2-3 用习题 6.2-2 中找到的 P 和 Q 重做习题 6.2-1，并报告跟踪误差性能的差异。

第 6.3 节

- 6.3-1 为第 2 章中给出的两连杆极坐标机器人机械臂设计和仿真表 6.3.1 中给出的自适应惯量相关控制器。
- 6.3-2 对于习题 6.3-1 中给出的仿真，用控制参数（即 Λ 、 K_v 、 Γ ）的不同值运行几次仿真，并报告对跟踪误差性能的影响。
- 6.3-3 列举表 6.3.1 中给出的自适应控制器相对于表 6.2.1 中给出的自适应控制器的优点。
- 6.3-4 如 (6.3.8) 中给出的，滤波跟踪误差定义为：

$$r = \dot{e} + \Lambda e$$

其中 Λ 是正定对角矩阵。证明如果

$$\lim_{t \rightarrow \infty} r = 0$$

那么

$$\lim_{t \rightarrow \infty} e = 0 \quad \text{和} \quad \lim_{t \rightarrow \infty} \dot{e} = 0$$

第 6.4 节

6.4-1 用例 6.4-3 中给出的 PID 伺服律和例 6.4-1 中给出的自适应律为例 6.3-1 中给出的两连杆旋转机械臂设计和仿真自适应控制器。报告与例 6.3-1 中给出的任何差异。

6.4-2 用方程 (6.4.20) 和 (6.4.21) 给出的比例 + 积分自适应律重做习题 6.4-1。

第 6.5 节

6.5-1 从分析上证明例 6.5-2 中的 $q_d = \sin t$ 不是持续激励的。

第 6.6 节

6.6-1 证明

$$\frac{d}{dt}[P^{-1}] = -P^{-1}\dot{P}P^{-1}$$

6.6-2 仿真例 6.6-3 中给出的复合自适应控制器，并报告对跟踪误差的 $P(0)$ 和 a （即用于滤波回归矩阵的滤波器极点）不同值的影响。

6.6-3 展示如何使用第 1 章中给出的 Barbalat 引理在复合自适应控制器的证明中得到

$$\lim_{t \rightarrow \infty} \dot{e} = 0$$

第 6.7 节

6.7-1 将附加有界干扰：

$$T_d = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

添加到第 2 章中给出的两连杆极坐标机器人机械臂动力学，重做习题 6.3-1。

6.7-2 用习题 6.7-1 中给出的附加有界干扰和项

$$K_d \text{sgn}(r)$$

添加到自适应控制器，重做习题 6.3-1。用 k_d 的不同值运行几次仿真，并报告对跟踪误差性能的影响。

Chapter 7

先进控制技术

第七章 先进控制技术

本章讨论机器人机械臂跟踪控制的一些先进控制技术。本章开发的控制器涉及计算问题和执行器动力学效应。本章提出的分析概念和控制开发通常比前几章提出的更为复杂；因此，强烈建议在研究新材料之前先学习前面的章节。

7.1 引言

随着过去几年来机器人控制研究的进展，许多机器人控制研究人员开始关注实现问题。也就是说，实现方面的考虑，如减少在线计算和执行器动力学效应，正在促使研究人员重新思考以前机器人控制器的理论开发，以便解决这些考虑。不断重新设计以前的控制开发以符合实现限制，是以前机器人控制研究进展的进行方式。利用这种迫使理论开发满足实现限制的概念，我们说明了研究人员如何开始着手解决诸如减少在线计算和补偿执行器动力学效应等问题。

7.2 减少在线计算的机器人控制器

在本节中，我们研究了由 Sadegh 及其合作者 [Sadegh 和 Horowitz 1990]、[Sadegh 等 1990] 设计的机器人控制器。我们将这些控制器与相关工作分开，因为这项工作解决了极其相关的在线控制器计算实现问题。具体而言，这种自适应控制器减少了在线计算，而不是其他控制技术，如第 6 章中提出的自适应控制器。在自适应控制器研究的发展之后，还提出了一种“重复”控制器。这种重复控制器也减少了在线计算。

期望补偿自适应律

第 6 章中自适应控制器的一个缺点是，用作前馈补偿的回归矩阵（例如，自适应惯量相关控制器中的矩阵 $Y(\cdot)$ ）必须在线计算。回归矩阵必须在线计算，因为它取决于关

节位置和速度的测量值（即 q 和 \dot{q} ）。对于例 6.3.1 中给出的简单二连杆机器人控制器，很明显 $Y(\cdot)$ 的在线计算在计算上是密集的。可以想象，如果人们希望控制具有许多自由度的机器人机械臂，回归矩阵的在线计算可能在计算上非常密集。

为了消除对回归矩阵在线计算的需要，我们现在研究期望补偿自适应律（DCAL）[Sadegh 和 Horowitz 1990]。DCAL 通过用期望关节位置和速度（即 q_d 和 \dot{q}_d ）替换 q 和 \dot{q} ，消除了对回归矩阵在线计算的需要。也就是说，DCAL 回归矩阵仅取决于期望轨迹信息；因此，DCAL 回归矩阵可以预先离线计算。当然，这种回归矩阵的修改迫使我们重新检查自适应控制设计和相应的稳定性分析。

为了本节中的控制设计，我们假设机器人机械臂是具有以下动力学的旋转机械臂：

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F_d\dot{q} = \tau \quad (7.1)$$

其中 F_d 是 $n \times n$ 正定对角矩阵，用于表示摩擦的动态系数，所有其他量如第 3 章中定义。如其他章节所述，我们将关节跟踪误差定义为

$$e = q_d - q \quad (7.2)$$

如第 6 章所述，机器人机械臂的自适应控制涉及将已知时间函数与未知常数参数分离。例如，回想自适应惯量相关控制器的这种参数与时间函数分离为

$$M(q)\ddot{q}_r + V_m(q, \dot{q})\dot{q}_r + G(q) + F_d\dot{q}_r = Y(\cdot)\varphi \quad (7.3)$$

其中 $Y(\cdot)$ 是 $n \times r$ 回归矩阵，仅取决于实际和期望轨迹的已知时间函数， φ 是 $r \times 1$ 未知常数参数向量。（注意表 6.3.1 中定义的 Λ 取为单位矩阵。）

在 DCAL 中，参数与时间函数的这种分离为

$$M(q_d)\ddot{q}_r + V_m(q_d, \dot{q}_d)\dot{q}_r + G(q_d) + F_d\dot{q}_r = Y_d(\cdot)\varphi \quad (7.4)$$

其中 $Y_d(\cdot)$ 是 $n \times r$ 回归矩阵，仅取决于期望轨迹的已知函数。注意，如果我们将 q_d 和 \dot{q}_d 分别代入 q 和 \dot{q} 到式(7.3)中，式(7.3)给出的回归矩阵公式与式(7.4)给出的公式等价。

利用式(7.4)给出的回归矩阵公式，DCAL 表示为

$$\tau = Y_d(\cdot)\hat{\varphi} + k_v r + k_p e + k_a \|e\|^2 r \quad (7.5)$$

其中 k_v 、 k_p 、 k_a 是标量常数控制增益， $\hat{\varphi}$ 是 $r \times 1$ 参数估计向量，滤波跟踪误差定义为

$$r = \dot{e} + e \quad (7.6)$$

相应的 DCAL 参数自适应更新律为

$$\dot{\hat{\varphi}} = \Gamma Y_d^T(\cdot) r \quad (7.7)$$

其中 Γ 是 $r \times r$ 正定、对角、常数自适应增益矩阵，参数误差定义为

$$\tilde{\varphi} = \varphi - \hat{\varphi} \quad (7.8)$$

注意，式(??)给出的 DCAL 与第 6 章讨论的自适应控制器非常相似，除了式(??)中的项 $k_a \|e\|^2 r$ 。事实证明，这个附加项用于补偿式(??)和(??)中给出的 $Y(\cdot)\varphi$ 和 $Y_d(\cdot)\varphi$ 之间的差异。如 [Sadegh 和 Horowitz 1990] 所示，实际回归矩阵与期望回归矩阵公式之间的这种差异可以量化为

$$\|Y(\cdot)\varphi - Y_d(\cdot)\varphi\| \leq \zeta_1 \|e\| + \zeta_2 \|e\| \|r\| + \zeta_3 \|r\| + \zeta_4 \quad (7.9)$$

其中

$$\tilde{Y} = Y(\cdot)\varphi - Y_d(\cdot)\varphi \quad (7.10)$$

而 ζ_1 、 ζ_2 、 ζ_3 和 ζ_4 是正的边界常数，取决于期望轨迹和特定机器人配置的物理特性（即连杆质量、连杆长度、摩擦系数等）。

为了分析式(??)给出的控制器的稳定性，我们必须形成相应的误差系统。首先，我们将式(??)用式(??)和(??)中定义的 $Y(\cdot)\varphi$ 和 r 重写。也就是说，我们有

$$M(q)\dot{r} + V_m(q, \dot{q})r + Y(\cdot)\varphi = \tau \quad (7.11)$$

在式(??)的右侧添加和减去项 $Y_d(\cdot)\varphi$ 得到

$$M(q)\dot{r} + V_m(q, \dot{q})r + Y_d(\cdot)\varphi + \tilde{Y} = \tau \quad (7.12)$$

其中 \tilde{Y} 在式(??)中定义。将式(??)给出的控制代入式(??)得到误差系统

$$M(q)\dot{r} = -V_m(q, \dot{q})r - k_v r - k_p e - k_a \|e\|^2 r - Y_d(\cdot)\tilde{\varphi} - \tilde{Y} \quad (7.13)$$

其中 $\tilde{\varphi}$ 在式(??)中定义。

我们现在用 Lyapunov-like 函数分析式(??)给出的误差系统的稳定性

$$V = \frac{1}{2} r^T M(q)r + \frac{1}{2} k_p e^T e + \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (7.14)$$

对式(??)关于时间求导得到

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2} r^T \dot{M}(q)r + k_p e^T \dot{e} - \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \quad (7.15)$$

因为标量量可以转置。将式(??)代入式(??)得到

$$\begin{aligned} \dot{V} = & -r^T V_m(q, \dot{q})r - k_v r^T r - k_p r^T e - k_a \|e\|^2 r^T r \\ & - r^T Y_d(\cdot)\tilde{\varphi} - r^T \tilde{Y} + \frac{1}{2} r^T \dot{M}(q)r + k_p e^T \dot{e} - \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \end{aligned} \quad (7.16)$$

通过利用斜对称性质（见第 3 章）和式(??)中的更新律，很容易看出式(??)中的第二行等于零，因此，通过调用式(??)中给出的 r 的定义，式(??)简化为

$$\dot{V} = -k_v r^T r - k_p e^T e - k_a \|e\|^2 r^T r - r^T \tilde{Y} \quad (7.17)$$

从式(??)，我们可以以以下方式放置 \dot{V} 的上界：

$$\dot{V} \leq -k_v \|r\|^2 - k_p \|e\|^2 - k_a \|e\|^2 \|r\|^2 + \|r\| \|\tilde{Y}\| \quad (7.18)$$

通过在式(??)中代入式(??)可以得到 \dot{V} 的新上界

$$\dot{V} \leq -k_v \|r\|^2 - k_p \|e\|^2 - k_a \|e\|^2 \|r\|^2 + \zeta_1 \|e\| \|r\| + \zeta_2 \|e\| \|r\|^2 + \zeta_3 \|r\|^2 + \zeta_4 \|r\| \quad (7.19)$$

通过重新排列式(??)的第二行，它可以写成

$$\dot{V} \leq -k_v \|r\|^2 - k_p \|e\|^2 - k_a \|e\|^2 \|r\|^2 + \zeta_1 \|e\| \|r\| + \zeta_2 \|e\| \|r\|^2 + \zeta_3 \|r\|^2 + \frac{1}{2} \|r\|^2 + \frac{1}{2} \zeta_4^2 \quad (7.20)$$

在式(??)中收集公共项后，它可以重写为

$$\dot{V} \leq -(k_v - \zeta_3 - \frac{1}{2}) \|r\|^2 - k_p \|e\|^2 - (k_a \|e\| - \zeta_1) \|e\| \|r\|^2 + \zeta_2 \|e\| \|r\|^2 + \frac{1}{2} \zeta_4^2 \quad (7.21)$$

通过注意到，如果控制增益 k_a 按照

$$k_a > \frac{\zeta_1 + \zeta_2}{\|e\|} \quad (7.22)$$

调整，我们可以看到式(??)第二行上的所有项都将为负；因此，我们可以获得 \dot{V} 的新上界。

$$\dot{V} \leq -(k_v - \zeta_3 - \frac{1}{2}) \|r\|^2 - k_p \|e\|^2 + \frac{1}{2} \zeta_4^2 \quad (7.23)$$

通过以矩阵形式重写式(??)

$$\dot{V} \leq - \begin{bmatrix} \|e\| & \|r\| \end{bmatrix} \begin{bmatrix} k_p & 0 \\ 0 & k_v - \zeta_3 - \frac{1}{2} \end{bmatrix} \begin{bmatrix} \|e\| \\ \|r\| \end{bmatrix} + \frac{1}{2} \zeta_4^2 \quad (7.24)$$

其中

$$Q_0 = \begin{bmatrix} k_p & 0 \\ 0 & k_v - \zeta_3 - \frac{1}{2} \end{bmatrix}$$

我们可以建立 k_p 和 k_v 的充分条件，使得式(??)中的矩阵为正定。具体而言，通过使用 Gerschgorin 定理（见第 2 章），我们可以看到，如果

$$k_p > 0 \quad (7.25)$$

和

$$k_v > \zeta_3 + \frac{1}{2} \quad (7.26)$$

式(??)中定义的矩阵将为正定；因此， \dot{V} 将为半负定。

我们现在详细说明跟踪误差的稳定性类型。首先，由于 \dot{V} 是半负定的，我们可以说明 V 是上界有界的。利用 V 是上界有界的事实，我们可以说明 e 、 \dot{e} 、 r 和 $\hat{\varphi}$ 是有界的。由于 e 、 \dot{e} 、 r 和 $\hat{\varphi}$ 是有界的，我们可以使用式(??)来表明 \dot{r} 以及因此式(??)中的 \dot{V} 是有界的。其次，注意由于 $M(q)$ 如惯性矩阵的正定性质所阐述的那样下界有界（见第 3 章），我们可以说明式(??)中给出的 V 下界有界。由于 V 下界有界， \dot{V} 是半负定的，且 \ddot{V} 是有界的，我们可以使用 Barbalat 引理（见第 2 章）来说明

$$\lim_{t \rightarrow \infty} \dot{V} = 0$$

因此，从上面的论证和式(??)，我们知道

$$\lim_{t \rightarrow \infty} \|e\|, \|r\| = 0 \tag{7.27}$$

从式(??)，我们也可以确定速度跟踪误差的稳定性结果。具体而言，从式(??)，注意 r 被定义为变量 e 的稳定一阶微分方程；因此，通过标准线性控制论证，我们可以写成

$$\lim_{t \rightarrow \infty} \|\dot{e}\| = 0 \tag{7.28}$$

这个结果告诉我们，如果控制器增益根据式(??)、(??)和(??)选择，跟踪误差 e 和 \dot{e} 是渐近稳定的。从上面的分析，关于参数误差我们所能说的就是它保持有界。刚刚推导的自适应控制器总结在表 7.2.1 中，并在图 7.2.1 中描绘。

表 7.2.1 DCAL 控制器

控制律	$\tau = Y_d(\cdot)\hat{\varphi} + k_v r + k_p e + k_a \ e\ ^2 r$
滤波跟踪误差	$r = \dot{e} + e$
参数更新律	$\dot{\hat{\varphi}} = \Gamma Y_d^T(\cdot) r$

浏览表 7.2.1 后，我们可以看到，与自适应惯量相关控制器相比，DCAL 具有明显的减少在线计算的优势。具体而言，回归矩阵 $Y_d(\cdot)$ 仅取决于期望轨迹；因此，回归矩阵可以离线计算。我们现在提出一个示例来说明如何使用表 7.2.1 为机器人机械臂设计自适应控制器。

示例 7.2-1：二连杆机械臂的 DCAL

我们希望为图 6.2.1 中给出的二连杆机械臂设计和仿真表 7.2.1 中给出的 DCAL。(该机器人臂的动力学在第 3 章中给出。) 假设摩擦可忽略不计且连杆长度精确知道为每根 1 米长，DCAL 可以写成

$$\tau_1 = Y_{d11}\hat{\varphi}_1 + Y_{d12}\hat{\varphi}_2 + k_v r_1 + k_p e_1 + k_a \|e\|^2 r_1 \tag{7.29}$$

和

$$\tau_2 = Y_{d21}\hat{\varphi}_1 + Y_{d22}\hat{\varphi}_2 + k_v r_2 + k_p e_2 + k_a \|e\|^2 r_2 \tag{7.30}$$

在控制扭矩的表达式中，回归矩阵 $Y_d(\cdot)$ 由

$$Y_d = \begin{bmatrix} \ddot{q}_{d1} + \dot{e}_1 & Y_{d12} \\ 0 & Y_{d22} \end{bmatrix} \quad (7.31)$$

给出其中

$$Y_{d12} = \ddot{q}_{d1} + \dot{e}_1 + \ddot{q}_{d2} + \dot{e}_2 + (\dot{q}_{d1} + \dot{e}_1)(\dot{q}_{d2} + \dot{e}_2)$$

$$Y_{d22} = \ddot{q}_{d1} + \dot{e}_1 + \ddot{q}_{d2} + \dot{e}_2$$

按照表 7.2.1 给出的公式制定自适应更新规则，相关参数估计向量为

$$\hat{\varphi} = \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \end{bmatrix}$$

带有自适应更新规则

$$\dot{\hat{m}}_1 = \gamma_1(Y_{d11}r_1 + Y_{d21}r_2) \quad (7.32)$$

和

$$\dot{\hat{m}}_2 = \gamma_2(Y_{d12}r_1 + Y_{d22}r_2) \quad (7.33)$$

对于 $m_1 = 0.8 \text{ kg}$ 和 $m_2 = 2.3 \text{ kg}$ ，DCAL 在以下条件下进行了仿真

$$k_v = k_p = k_a = 50, \quad \gamma_1 = \gamma_2 = 20$$

和

$$q_{d1} = q_{d2} = \sin t, \quad \dot{q}_{d1} = \dot{q}_{d2} = \cos t$$

$$\hat{m}_1(0) = 0, \quad \hat{m}_2(0) = 0$$

跟踪误差和质量估计在图 7.2.2 中描绘。如图所示，跟踪误差是渐近稳定的，参数估计保持有界。

重复控制律

在许多工业应用中，机器人机械臂被用来重复执行相同的任务。例如，机器人可能被要求在装配线上反复喷涂相同的零件。可以想象，这种喷涂操作的期望轨迹将是周期函数。也就是说，在为喷涂第一个零件生成期望轨迹之后（即第一个“试验”），相同的期望轨迹应该以重复的方式为喷涂下一个零件而遵循（即下一个“试验”）。

如果控制策略不考虑重复操作的性质，沿第一条轨迹犯的错误将从试验到试验重复。因此，人们有动机设计一种控制器，利用当前试验中的跟踪误差测量来改善下一次试验的跟踪性能。这些类型的控制器通常被称为“学习”或重复控制器。“学习控制器”一词用于强调控制器试图学习机械臂动力学的可重复部分。

为了推动重复控制律 (RCL) [Sadegh 等 1990] 的设计, 我们注意到由

$$u_d(t) = M(q_d)\ddot{q}_d + V_m(q_d, \dot{q}_d)\dot{q}_d + G(q_d) + F_d\dot{q}_d \quad (7.34)$$

给出的动力学如果期望轨迹是周期性的, 则是可重复的。也就是说, 即使式(??)中可能存在未知的常数参数, 由 $n \times 1$ 向量 $u_d(t)$ 表示的信号将是周期性的或可重复的。因此, 在随后的讨论中, 我们假设期望轨迹的周期为 T 。期望轨迹的这种周期性假设使我们能够写成

$$u_d(t) = u_d(t - T) \quad (7.35)$$

因为 $u_d(t)$ 表示的动力学仅取决于周期性量。

利用式(??)给出的动力学的可重复性, RCL 表示为

$$\tau = \hat{u}_d(t) + k_v r + k_p e + k_a \|e\|^2 r \quad (7.36)$$

其中 $n \times 1$ 向量 $\hat{u}_d(t)$ 是用于补偿可重复动力学 $u_d(t)$ 的学习项, 所有其他量与 DCAL 中定义的相同。学习项 $\hat{u}_d(t)$ 通过以下学习更新规则从试验到试验更新

$$\hat{u}_d(t) = \hat{u}_d(t - T) + k_L r \quad (7.37)$$

其中 k_L 是正的标量控制增益。

如自适应控制开发中类似地进行, 我们将用式(??)给出的学习更新规则用学习误差表示, 学习误差定义为

$$\tilde{u}_d(t) = u_d(t) - \hat{u}_d(t) \quad (7.38)$$

具体而言, 将式(??)乘以-1, 然后在式(??)的两边加上 $u_d(t)$ 得到

$$\tilde{u}_d(t) = \tilde{u}_d(t - T) - k_L r \quad (7.39)$$

通过利用式(??)给出的周期性假设, 我们可以将式(??)写成

$$u_d(t) - \hat{u}_d(t) = u_d(t - T) - \hat{u}_d(t - T) - k_L r \quad (7.40)$$

这给出了学习误差更新规则

$$\tilde{u}_d(t) = \tilde{u}_d(t - T) - k_L r \quad (7.41)$$

其中 $\tilde{u}_d(t - T)$ 根据式(??)定义。

在分析式(??)中给出的控制器的稳定性之前, 我们将形成相应的误差系统。首先, 我们将式(??)用式(??)中定义的 r 重写。也就是说, 我们有

$$M(q)\dot{r} + V_m(q, \dot{q})r + u_a(t) = \tau \quad (7.42)$$

其中 $n \times 1$ 向量 $u_a(t)$ 用于表示“实际机械臂动力学”

$$u_a(t) = M(q)(\ddot{q}_d + \dot{e}) + V_m(q, \dot{q})(\dot{q}_d + e) + G(q) + F_d(\dot{q}_d + e) \quad (7.43)$$

在式(??)的右侧添加和减去项 $u_d(t)$ 得到

$$M(q)\dot{r} + V_m(q, \dot{q})r + u_d(t) + \tilde{u} = \tau \quad (7.44)$$

其中 \tilde{u} 定义为

$$\tilde{u} = u_a(t) - u_d(t) \quad (7.45)$$

如 [Sadegh 和 Horowitz 1990] 中类似所示, 实际机械臂动力学 (即 $u_a(t)$) 与可重复机械臂动力学 (即 $u_d(t)$) 之间的这种差异可以量化为

$$\|\tilde{u}\| \leq \zeta_1 \|e\| + \zeta_2 \|e\| \|r\| + \zeta_3 \|r\| + \zeta_4 \quad (7.46)$$

其中 ζ_1 、 ζ_2 、 ζ_3 和 ζ_4 是正的边界常数, 取决于期望轨迹和特定机器人配置的物理特性 (即连杆质量、连杆长度、摩擦系数等)。

形成误差系统的最后一步是将式(??)给出的控制代入式(??)得到

$$M(q)\dot{r} = -V_m(q, \dot{q})r - k_v r - k_p e - k_a \|e\|^2 r + \tilde{u}_d(t) - \tilde{u} \quad (7.47)$$

我们现在用 Lyapunov-like 函数分析式(??)给出的误差系统的稳定性

$$V = \frac{1}{2} r^T M(q) r + \frac{1}{2} k_p e^T e + \frac{1}{2k_L} \int_{t-T}^t \tilde{u}_d^T(\sigma) \tilde{u}_d(\sigma) d\sigma \quad (7.48)$$

对式(??)关于时间求导得到

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2} r^T \dot{M}(q)r + k_p e^T \dot{e} + \frac{1}{2k_L} [\tilde{u}_d^T(t) \tilde{u}_d(t) - \tilde{u}_d^T(t-T) \tilde{u}_d(t-T)] \quad (7.49)$$

将式(??)给出的误差系统代入式(??)得到

$$\begin{aligned} \dot{V} = & -r^T V_m(q, \dot{q})r - k_v r^T r - k_p r^T e - k_a \|e\|^2 r^T r \\ & + r^T \tilde{u}_d(t) - r^T \tilde{u} + \frac{1}{2} r^T \dot{M}(q)r + k_p e^T \dot{e} + \frac{1}{2k_L} [\tilde{u}_d^T(t) \tilde{u}_d(t) - \tilde{u}_d^T(t-T) \tilde{u}_d(t-T)] \end{aligned} \quad (7.50)$$

通过利用斜对称性质和学习误差更新律式(??), 很容易表明式(??)中的第二行等于

$$-\frac{1}{2k_L} [\tilde{u}_d(t) + k_L r]^T [\tilde{u}_d(t) + k_L r]$$

因此, 通过调用式(??)中给出的 r 的定义, 式(??)简化为

$$\dot{V} = -k_v r^T r - k_p e^T e - k_a \|e\|^2 r^T r + r^T \tilde{u} \quad (7.51)$$

从式(?)我们可以以以下方式放置 \dot{V} 的上界:

$$\dot{V} \leq -k_v \|r\|^2 - k_p \|e\|^2 - k_a \|e\|^2 \|r\|^2 + \|r\| \|\tilde{u}\| \quad (7.52)$$

其余稳定性论证是前一节对 DCAL 提出的稳定性论证的修改。具体而言, 我们首先注意到式(?)和(?)几乎相同, 因为式(?)中的 \tilde{Y} 和式(?)中的 \tilde{u} 由相同的标量函数界定。追溯 DCAL 稳定性论证的步骤后, 我们可以看到控制器增益 k_a 和 k_p 仍应按照式(?)和(?)分别调整。然而, 控制器增益 k_v 与控制器增益 k_L 一起调整以满足

$$k_v > \zeta_3 + \frac{1}{2} + k_L \zeta_2^2 \quad (7.53)$$

其中 ζ_1 、 ζ_2 、 ζ_3 和 ζ_4 在式(?)中定义。如果控制器增益根据式(?), (?), 和(?)调整, 则从 DCAL 的分析开发 (即式(?)到(?)) 和式(?), 我们可以在 \dot{V} 上放置新的上界:

$$\dot{V} \leq -\lambda_3 \|x_0\|^2 + \frac{1}{2} \zeta_4^2 \quad (7.54)$$

其中 λ_3 是正的标量常数, 由 $\lambda_{\min}\{Q_0\}$ 给出。

我们现在详细说明跟踪误差的稳定性类型。首先注意, 从式(?), 我们可以在 \dot{V} 上放置新的上界:

$$\dot{V} \leq -\lambda_3 \|x_0\|^2 + \frac{1}{2} \zeta_4^2 \quad (7.55)$$

这意味着

$$\int_0^\infty \|x_0\|^2 dt \leq \frac{1}{\lambda_3} [V(0) - V(\infty)] + \frac{1}{2\lambda_3} \zeta_4^2 \int_0^\infty dt \quad (7.56)$$

将式(?)乘以-1 并对式(?)的左侧积分得到

$$-\int_0^\infty \|x_0\|^2 dt \geq \frac{1}{\lambda_3} [V(\infty) - V(0)] - \frac{1}{2\lambda_3} \zeta_4^2 \int_0^\infty dt \quad (7.57)$$

由于 \dot{V} 如式(?)所阐述的那样是半负定的, 我们可以说明 V 是一个非递增函数, 因此由上界 $V(0)$ 界定。通过回忆, 如惯性矩阵的正定性所阐述的那样, $M(q)$ 是下界有界的, 我们可以说明式(?)中给出的 V 由零下界有界。由于 V 是非递增的, 由上界 $V(0)$ 界定, 由下界零界定, 我们可以将式(?)写成

$$\int_0^\infty \|x_0\|^2 dt \leq \frac{1}{\lambda_3} V(0) + \frac{1}{2\lambda_3} \zeta_4^2 \int_0^\infty dt \quad (7.58)$$

或

$$x_0 \in L_2^n \quad (7.59)$$

式(?)给出的界限告诉我们 $x_0 \in L_2^n$ (见第 2 章), 这意味着滤波跟踪误差 r 以式(?)给出的特殊方式有界。

为了建立位置跟踪误差 e 的稳定性结果, 我们建立位置跟踪误差与滤波跟踪误差 r 之间的传递函数关系。从式(?), 我们可以说明

$$e(s) = G(s)r(s) \quad (7.60)$$

其中 s 是拉普拉斯变换变量,

$$G(s) = (sI + I)^{-1} \quad (7.61)$$

而 I 是 $n \times n$ 单位矩阵。由于 $G(s)$ 是严格适当的、渐近稳定的传递函数且 $r \in L_2^n$, 我们可以使用第 2 章中的定理 2.4.7 来说明

$$\lim_{t \rightarrow \infty} \|e\| = 0 \quad (7.62)$$

因此, 如果控制器增益根据式(??)、(??)和(??)选择, 位置跟踪误差 e 是渐近稳定的。根据本节提出的理论开发, 关于速度跟踪误差 \dot{e} 我们只能说它是有界的。应该注意的是, 如果式(??)中的学习估计 $\hat{u}_d(t)$ 人为地防止增长, 我们可以得出速度跟踪误差是渐近稳定的结论 [Sadegh 等 1990]。这种修改的稳定性证明是第 6 章中提出的自适应控制证明的直接应用。

本节研究的重复控制器总结在表 7.2.2 中, 并在图 7.2.3 中描绘。浏览表 7.2.2 后, 我们可以看到 RCL 与需要形成回归型矩阵的自适应控制器相比, 需要关于被控机器人的很少信息。RCL 的另一个明显优势是它需要很少的在线计算。我们现在提出一个示例来说明如何使用表 7.2.2 为机器人机械臂设计重复控制器。

示例 7.2-2: 二连杆机械臂的 RCL

我们希望为图 6.2.1 中给出的二连杆机械臂设计和仿真表 7.2.2 中给出的 RCL。(该机器人臂的动力学在第 3 章中给出。) 从表 7.2.2, RCL 可以写成

$$\tau_1 = \hat{u}_{d1} + k_v r_1 + k_p e_1 + k_a \|e\|^2 r_1 \quad (7.63)$$

和

$$\tau_2 = \hat{u}_{d2} + k_v r_2 + k_p e_2 + k_a \|e\|^2 r_2 \quad (7.64)$$

按照表 7.2.2 给出的公式制定学习更新规则得到

$$\hat{u}_{d1}(t) = \hat{u}_{d1}(t - T) + k_L r_1 \quad (7.65)$$

和

$$\hat{u}_{d2}(t) = \hat{u}_{d2}(t - T) + k_L r_2 \quad (7.66)$$

对于 $m_1 = 0.8$ kg, $m_2 = 2.3$ kg, 连杆长度为 1 m, RCL 在以下条件下进行了仿真

$$k_a = k_v = k_p = k_L = 50, \quad T = 2\pi$$

和

$$q_{d1} = \sin t, \quad q_{d2} = \cos t$$

$$\hat{u}_{d1}(t) = 0, \quad \hat{u}_{d2}(t) = 0, \quad \text{对于 } -2\pi \leq t < 0$$

位置和速度跟踪误差在图 7.2.4 中描绘。如图所示, 位置和速度跟踪误差都是渐近稳定的; 然而, 根据本小节的理论开发, 我们只保证位置跟踪误差是渐近稳定的。

7.3 自适应鲁棒控制

在第 6 章中，我们讨论了自适应控制器用于机器人机械臂跟踪控制的用途。自适应控制器的有吸引力的特征之一是，控制实现不需要诸如负载质量或摩擦系数之类的未知常数参数的先验知识。自适应控制器的两个缺点是需要大量的在线计算，以及缺乏对附加有界扰动的鲁棒性。

在第 5 章中，我们讨论了鲁棒控制器用于机器人机械臂控制的用途。鲁棒控制器的两个有吸引力的特征是在线计算保持在最低限度，以及它们对附加有界扰动的固有鲁棒性。鲁棒控制方法的一个缺点是需要不确定性的先验已知界限。通常，不确定性界限的计算可能是一个相当繁琐的过程，因为此计算涉及为机器人机械臂的每个连杆找到质量和摩擦相关常数的最大值。鲁棒控制方法的另一个缺点是，即使在不存在附加有界扰动的情况下，我们也不能保证跟踪误差的渐近稳定性。通常，至少获得跟踪误差的“理论”渐近稳定性结果将是可取的。

在本节中，为机器人机械臂的跟踪控制开发了自适应鲁棒控制器。自适应鲁棒控制器可以被认为是结合了自适应控制器和鲁棒控制器的最佳品质。这种控制方法具有减少在线计算（与自适应控制方法相比）、对附加有界扰动的鲁棒性、无需系统不确定性的先验知识以及渐近跟踪误差性能的优点。

为了本节中的控制设计，我们假设机器人机械臂是具有以下动力学的旋转机械臂：

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F_d\dot{q} + T_d = \tau \quad (7.67)$$

其中 F_d 是用于表示摩擦动态系数的 $n \times n$ 正定对角矩阵， T_d 是表示未知有界扰动的 $n \times 1$ 向量，所有其他量如第 3 章中定义。

自适应鲁棒控制器与第 5 章中讨论的鲁棒控制策略非常相似，因为使用辅助控制器来“界定”不确定性。回想第 5 章，鲁棒控制器通过使用由跟踪误差范数和正边界常数组成的标量函数来界定不确定性。例如，假设由

$$w = M(q)\ddot{q}_r + V_m(q, \dot{q})\dot{q}_r + G(q) + F_d\dot{q}_r + T_d \quad (7.68)$$

给出的动力学表示给定机器人控制器的不确定性。也就是说，式(??)给出的动力学是不确定的，因为负载质量、摩擦系数和扰动不确切知道。然而，假设可以使用正的标量函数 ρ 来界定不确定性，如下所示：

$$\|w\| \leq \rho \quad (7.69)$$

如 [Dawson 等 1990] 中所述，机器人机械臂的物理特性可用于表明式(??)给出的动力学可以界定为

$$\rho = \delta_0 + \delta_1\|e\| + \delta_2\|\dot{e}\| \quad (7.70)$$

其中

$$\theta = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{bmatrix} \quad (7.71)$$

而 δ_0 、 δ_1 和 δ_2 是正的边界常数，基于最大可能的负载质量、连杆质量、摩擦系数、扰动等。

通常，第 5 章中提出的鲁棒控制器要求式(??)中定义的边界常数事先制定。本节将开发的自适应鲁棒控制器在机械臂运动时在线”学习”这些边界常数。也就是说，在控制实现中，我们不需要边界常数的知识；相反，我们只需要式(??)中定义的边界常数的存在。

与 [Corless 和 Leitmann 1983] 中提出的一般开发类似，自适应鲁棒控制器具有以下形式

$$\tau = K_v r + v_R \quad (7.72)$$

其中 K_v 是 $n \times n$ 对角、正定矩阵， r （滤波跟踪误差）如式(??)中定义，而 v_R 是表示辅助控制器的 $n \times 1$ 向量。式(??)中的辅助控制器 v_R 由

$$v_R = \frac{r \hat{\rho}^2}{\hat{\rho} \|r\| + \varepsilon} \quad (7.73)$$

定义其中

$$\dot{\varepsilon} = -k_\varepsilon \varepsilon \quad (7.74)$$

k_ε 是正的标量控制常数， $\hat{\rho}$ 是定义为

$$\hat{\rho} = S \hat{\theta} \quad (7.75)$$

的标量函数而 $\hat{\theta}$ 是边界常数 δ_0 、 δ_1 和 δ_2 的相应边界估计，在式(??)中定义。用””表示的边界估计基于自适应更新规则在线更改。在给出更新规则之前，我们将式(??)写成更方便的形式

$$\hat{\rho} = \hat{\theta}^T S^T \quad (7.76)$$

其中

$$S = \begin{bmatrix} 1 & \|e\| & \|r\| \end{bmatrix}$$

式(??)中给出的实际边界函数 ρ 也可以写成矩阵形式

$$\rho = \theta^T S^T \quad (7.77)$$

其中

$$\theta = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{bmatrix}$$

注意自适应方法中回归矩阵公式（见第 6 章）与式(??)给出的公式之间的相似性。具体而言， 1×3 矩阵 S 类似于“回归矩阵”，而 3×1 向量 $\hat{\theta}$ 类似于“参数估计向量”。

式(??)中定义的边界估计通过以下关系在线更新

$$\dot{\hat{\theta}} = \gamma S^T \|r\| \quad (7.78)$$

其中 r 在式(??)中定义， S 在式(??)中定义， γ 是正的标量控制常数。为方便起见，我们还注意到，由于式(??)中定义的 δ_0 、 δ_1 和 δ_2 是常数，式(??)可以写成

$$\dot{\hat{\theta}} = -\gamma S^T \|r\| \quad (7.79)$$

因为我们将定义 θ 和 $\hat{\theta}$ 之间的差为

$$\tilde{\theta} = \theta - \hat{\theta} \quad (7.80)$$

我们现在将注意力转向分析式(??)中给出的控制器的相应误差系统的稳定性。将控制器式(??)代入机器人方程式(??)得到误差系统

$$M(q)\dot{r} = -V_m(q, \dot{q})r - K_v r - v_R + w \quad (7.81)$$

其中 w 在式(??)中定义。

我们现在用 Lyapunov-like 函数分析式(??)给出的误差系统的稳定性

$$V = \frac{1}{2} r^T M(q) r + \frac{1}{2} \tilde{\theta}^T \gamma^{-1} \tilde{\theta} + \frac{1}{2k_\varepsilon} \varepsilon^2 \quad (7.82)$$

对式(??)关于时间求导得到

$$\dot{V} = r^T M(q)\dot{r} + \frac{1}{2} r^T \dot{M}(q)r + \tilde{\theta}^T \gamma^{-1} \dot{\tilde{\theta}} + \frac{1}{k_\varepsilon} \varepsilon \dot{\varepsilon} \quad (7.83)$$

因为标量量可以转置。将式(??)和(??)代入式(??)得到

$$\begin{aligned} \dot{V} = & -r^T V_m(q, \dot{q})r - r^T K_v r - r^T v_R + r^T w \\ & + \frac{1}{2} r^T \dot{M}(q)r - \tilde{\theta}^T S^T \|r\| - \varepsilon \end{aligned} \quad (7.84)$$

通过利用斜对称性质，很容易看出式(??)中的第二行等于零。从式(??)，我们可以使用式(??)和(??)以以下方式放置 \dot{V} 的上界：

$$\dot{V} \leq -r^T K_v r - \rho \|r\| + r^T w - \tilde{\theta}^T S^T \|r\| - \varepsilon \quad (7.85)$$

将式(??)、(??)、(??)和(??)代入式(??)，我们得到

$$\dot{V} \leq -r^T K_v r - \frac{\|r\|^2 \hat{\theta}^T S^T}{\hat{\theta}^T S^T \|r\| + \varepsilon} + \|r\| \theta^T S^T - \|r\| \hat{\theta}^T S^T + \|r\| \hat{\theta}^T S^T - \varepsilon \quad (7.86)$$

它可以写成

$$\dot{V} \leq -r^T K_v r - \|r\| \hat{\theta}^T S^T \left[\frac{\|r\|}{\hat{\theta}^T S^T \|r\| + \varepsilon} - 1 \right] - \varepsilon \quad (7.87)$$

为式(??)中的最后两项获得公共分母使我们能够将式(??)写成

$$\dot{V} \leq -r^T K_v r - \frac{\|r\| \hat{\theta}^T S^T (-\varepsilon)}{\hat{\theta}^T S^T \|r\| + \varepsilon} - \varepsilon \quad (7.88)$$

由于式(??)中最后两项之和总是小于零，我们可以在 \dot{V} 上放置新的上界：

$$\dot{V} \leq -r^T K_v r \quad (7.89)$$

我们现在详细说明跟踪误差的稳定性类型。首先，注意从式(??)，我们可以在 \dot{V} 上放置新的上界：

$$\dot{V} \leq -\lambda_{\min}\{K_v\} \|r\|^2 \quad (7.90)$$

如前一节对 RCL 所示，我们可以使用式(??)来表明所有信号都是有界的，且 $r \in L_2^n$ (见第 2 章)。按照 RCL 稳定性分析，我们可以使用式(??)来表明位置跟踪误差 e 与滤波跟踪误差 r 通过传递函数关系相关

$$e(s) = G(s)r(s) \quad (7.91)$$

其中 s 是拉普拉斯变换变量， $G(s)$ 是严格适当的、渐近稳定的传递函数。因此，我们可以使用第 2 章中的定理 2.4.7 来说明

$$\lim_{t \rightarrow \infty} \|e\| = 0 \quad (7.92)$$

上面的结果告诉我们位置跟踪误差 e 是渐近稳定的。根据本节提出的理论开发，我们只能说明速度跟踪误差 \dot{e} 和边界估计 $\hat{\theta}$ 是有界的。应该注意的是，在 [Corless 和 Leitmann 1983] 中，提出了更复杂的理论开发，证明速度跟踪误差是渐近稳定的。然而，为简洁起见，这些附加信息留给读者自行研究。

本节推导的自适应鲁棒控制器总结在表 7.3.1 中，并在图 7.3.1 中描绘。我们现在提出一个示例来说明如何使用表 7.3.1 为机器人机械臂设计自适应鲁棒控制器。

表 7.3.1 自适应鲁棒控制器

控制律	$\tau = K_v r + \frac{r \hat{\rho}^2}{\hat{\rho} \ r\ + \varepsilon}$
辅助控制	$\hat{\rho} = S \hat{\theta} = \hat{\delta}_0 + \hat{\delta}_1 \ e\ + \hat{\delta}_2 \ r\ $
参数更新	$\dot{\hat{\theta}} = \gamma S^T \ r\ $

示例 7.3-1: 二连杆机械臂的自适应鲁棒控制器

我们希望为图 6.2.1 中给出的二连杆机械臂设计和仿真表 7.3.1 中给出的自适应鲁棒控制器。(该机器人臂的动力学在第 3 章中给出。)为了建模摩擦和扰动,将以下动力学

$$F_d \dot{q} = \begin{bmatrix} 3\dot{q}_1 \\ 3\dot{q}_2 \end{bmatrix} \quad (7.93)$$

和

$$T_d = \begin{bmatrix} 0.5 \sin t \\ 0.2 \cos t \end{bmatrix} \quad (7.94)$$

添加到二连杆机器人模型中的相应位置。

我们现在可以使用表 7.3.1 来制定自适应鲁棒控制器为

$$\tau_1 = k_v r_1 + \frac{r_1 \hat{\rho}^2}{\hat{\rho} \|r_1\| + \varepsilon} \quad (7.95)$$

和

$$\tau_2 = k_v r_2 + \frac{r_2 \hat{\rho}^2}{\hat{\rho} \|r_2\| + \varepsilon} \quad (7.96)$$

在上面控制扭矩的表达式中,边界函数 $\hat{\rho}$ 由

$$\hat{\rho} = \hat{\delta}_0 + \hat{\delta}_1 \|e\| + \hat{\delta}_2 \|r\| \quad (7.97)$$

给出其中 $\hat{\delta}_0$ 、 $\hat{\delta}_1$ 和 $\hat{\delta}_2$ 是自适应估计。从表 7.3.1,相关边界估计按以下方式更新

$$\dot{\hat{\delta}}_0 = \gamma \|r\|, \quad \dot{\hat{\delta}}_1 = \gamma \|e\| \|r\|, \quad \dot{\hat{\delta}}_2 = \gamma \|r\|^2 \quad (7.98)$$

对于 $m_1 = 0.8 \text{ kg}$, $m_2 = 2.3 \text{ kg}$, 每根连杆长度为 1 m , 自适应鲁棒控制器在以下条件下进行了仿真:控制参数、初始条件和期望轨迹由

$$k_v = 50, \quad \gamma = 10, \quad k_\varepsilon = 0.001, \quad \varepsilon(0) = 1$$

和

$$q_{d1} = \sin t, \quad q_{d2} = \cos t, \quad \hat{\delta}_0(0) = 0, \quad \hat{\delta}_1(0) = 0, \quad \hat{\delta}_2(0) = 0$$

给出

跟踪误差和质量估计在图 7.3.2 中描绘。如图所示,位置和速度跟踪误差都是渐近稳定的,边界估计保持有界。应该注意的是,从本节给出的理论开发,我们只保证位置跟踪误差是渐近稳定的,而所有其他信号保持有界。

7.4 执行器动力学补偿

在本书中,我们讨论了在“扭矩输入级别”设计的控制器。也就是说,与执行器相关的任何动力学都被忽略了。提出这一点的原因不是要贬低先前讨论的控制开发,因为

这项研究涉及解决一个非常困难的问题，即在不确定性存在的情况下高度非线性系统的全局跟踪控制。我们提到以前方法中的这一不足是为了强调这样一个事实：根据许多机器人控制研究人员的观点，现在是时候开始在控制综合中包括执行器动力学的效应了。最近，一些研究人员假设执行器的有害效应正在阻止机器人机械臂的高速运动/力控制 [Eppinger 和 Seering 1987]。

在本节中，我们说明了如何使用系统方法来补偿电气效应和关节柔性形式的执行器动力学。使用这种方法和精确模型知识的假设，开发了产生连杆跟踪误差全局渐近稳定性结果的控制器。尽管我们假设精确知道模型，但重要的是要认识到，在某些情况下，可能可以制定自适应和鲁棒非线性跟踪控制器来补偿“不确定性”。在存在执行器动力学的情况下补偿不确定系统目前正在研究中 [Ghorbel 和 Spong 1990]。

电气动力学

在本小节中，我们说明了如何合成“校正”控制器 [Kokotovic 等 1986]，以确保尽管电机增加整体系统动力学的电气动力学，连杆跟踪也是渐近的。“校正控制器”一词用于强调控制器校正电气动力学。本小节中研究的机器人将被称为刚连杆电气驱动 (RLED) 机器人。为简单起见，我们假设执行器是直流 (dc) 电机；然而，通过一些修改，以下分析可用于更复杂的电机，如开关磁阻电机 [Taylor 1989]。

RLED 机器人的模型 [Tarn 等 1991] 采用

$$\dot{e} = A_0 e + B \eta_E \quad (7.99)$$

$$\dot{\eta}_E = J^{-1} K_T I - \dot{u}_L \quad (7.100)$$

其中

$$\bar{M}(q) = M(q) + J \quad (7.101)$$

$M(q)$ 是 $n \times n$ 连杆惯性矩阵， $N(q, \dot{q})$ 是包含向心力、科里奥利力、重力、阻尼和摩擦项的 $n \times 1$ 向量， J 是用于表示执行器惯性的 $n \times n$ 常数、对角、正定矩阵， $I(t)$ 是用于表示每个执行器中电流的 $n \times 1$ 向量， K_T 是用于表示扭矩和电流之间转换的常数对角 $n \times n$ 矩阵， L_a 是用于表示电气电感的 $n \times n$ 常数正定对角矩阵， $R(q, \dot{q})$ 是用于表示电气电阻和电机反电动势的 $n \times 1$ 向量， $u_E(t)$ 是用于表示输入电机电压的 $n \times 1$ 控制向量。

在本书中，重点放在利用机器人机械臂的物理特性来帮助我们进行稳定性分析。遵循这一传统，我们注意到式(??)中定义的复合惯性矩阵 $\bar{M}(q)$ 是对称的、正定的，并且作为 q 的函数一致有界；因此，我们可以对任何 $n \times 1$ 向量 x 说明

$$\frac{1}{m_1} \|\bar{M}^{-1}(q)\|_{i_2} \leq 1 \quad (7.102)$$

其中 m_1 是正的标量常数，取决于特定机器人的质量特性（见第 3 章）。从式(??)，也可以建立

$$m_1 \leq \lambda_{\min}\{\bar{M}(q)\} \quad (7.103)$$

其中 $\|\cdot\|_{i2}$ 用于表示诱导 2-范数 (见第 2 章)。

如多次讨论的那样, 我们对连杆跟踪误差的性能感兴趣。为避免混淆, 我们重申跟踪误差定义为

$$e = q_d - q \quad (7.104)$$

其中 q_d 表示期望连杆轨迹。我们将假设 q_d 及其第一、第二和第三导数都是作为时间函数有界的。我们还假设式(?)左侧连杆动力学的第一导数存在。对期望轨迹和连杆动力学的这些“平滑性”假设确保稍后开发的控制器保持有界。

控制目标将是尽管有电气动力学, 也要获得渐近连杆跟踪。为了实现这一目标, 我们首先用式(?)给出的跟踪误差将式(?)重写为

$$\dot{e} = A_0 e + B \eta_E \quad (7.105)$$

式(?)给出的误差系统也可以用状态空间形式写成

$$\dot{e} = A_0 e + B \eta_E \quad (7.106)$$

其中

$$A_0 = \begin{bmatrix} -I_{n \times n} & I_{n \times n} \\ -K_{Lp} & -K_{Lv} \end{bmatrix}, \quad B = \begin{bmatrix} O_{n \times n} \\ \bar{M}^{-1}(q) \end{bmatrix}$$

$O_{n \times n}$ 是 $n \times n$ 零矩阵, $I_{n \times n}$ 是 $n \times n$ 单位矩阵。

如人们可以清楚地看到, 式(?)中没有控制输入; 因此, 我们将在式(?)的右侧添加和减去项 $B\bar{M}^{-1}(q)u_L$ 得到

$$\dot{e} = A_L e + B(\bar{M}^{-1}(q)u_L - \eta_E) \quad (7.107)$$

其中 u_L 是表示“虚构” $n \times 1$ 控制输入的 $n \times 1$ 向量。事实证明, 如果电气动力学不存在, 控制器 u_L 是计算扭矩控制器, 确保渐近连杆跟踪误差。正如我们稍后将看到的, 虚构控制器 u_L 实际上嵌入在整体控制策略中, 该策略设计在电压控制输入 u_E 处。

继续误差系统开发, 我们为 RLED 机器人定义 u_L 为计算扭矩控制器

$$u_L = \bar{M}(q)(\ddot{q}_d + K_{Lv}\dot{e} + K_{Lp}e) + N(q, \dot{q}) \quad (7.108)$$

其中 K_{Lv} 和 K_{Lp} 定义为 $n \times n$ 正定对角矩阵。仅将式(?)的第一个 u_L 项代入式(?)得到连杆跟踪误差系统

$$\dot{e} = A_L e + B \eta_E \quad (7.109)$$

其中

$$A_L = \begin{bmatrix} -I_{n \times n} & I_{n \times n} \\ -K_{Lp} & -K_{Lv} \end{bmatrix}, \quad \eta_E = \bar{M}^{-1}(q)(u_L - K_T I) \quad (7.110)$$

关于式(?)给出的连杆跟踪误差系统, 如果 η_E 可以保证在所有时间为零, 我们可以很容易地表明跟踪误差是渐近稳定的, 因为式(?)中定义的 A_L 具有稳定的特征值。因此, 可以将控制目标视为迫使”扰动” η_E 趋近于零。

为了设计 η_E 的控制律, 我们必须首先建立其动态特性。从式(?), η_E 关于时间的导数为

$$\dot{\eta}_E = \bar{M}^{-1}(q)\dot{u}_L - \dot{u}_L \quad (7.111)$$

为了获得 η_E 的动态特性, 我们从式(?)代入 \dot{I} 到式(?)得到

$$\dot{\eta}_E = J^{-1}K_T I - \dot{u}_L \quad (7.112)$$

我们现在可以使用式(?)设计输入 u_E 的控制律以迫使 η_E 趋近于零。 η_E 应该趋近于零的事实推动了校正控制律

$$u_E = L_a K_T^{-1} J(\ddot{u}_L + K_{Ep}\eta_E) + K_T^{-1} J R(\dot{q}, I) + u_L \quad (7.113)$$

其中 K_{Ep} 定义为 $n \times n$ 正定对角矩阵。将式(?)代入式(?)得到

$$\dot{\eta}_E = -K_{Ep}\eta_E \quad (7.114)$$

式(?)和(?)给出的动力学方程可以被认为代表整体闭环动力学的两个相互连接的子系统。如人们所期望的, 确定整体闭环系统的稳定性类型将是可取的。为了确定式(?)和(?)的稳定性类型, 我们将利用 Lyapunov 函数

$$V = x_0^T P_L x_0 + \eta_E^T P_E \eta_E \quad (7.115)$$

其中

$$P_L = \frac{1}{2} \begin{bmatrix} K_{Lp} + \frac{1}{2}K_{Lv} & \frac{1}{2}I_{n \times n} \\ \frac{1}{2}I_{n \times n} & I_{n \times n} \end{bmatrix}, \quad P_E = \frac{1}{2}J$$

如果满足

$$\lambda_{\min}\{K_{Lv}\} > 1 \quad (7.116)$$

给出的充分条件, 则由 Gerschgorin 定理 (见第 2 章), 很明显 P_L 是正定矩阵, 因此 V 是 Lyapunov 函数。式(?)给出的条件简单地意味着最小速度控制器增益应大于 1。

对式(?)关于时间求导得到

$$\dot{V} = \dot{x}_0^T P_L x_0 + x_0^T P_L \dot{x}_0 + \dot{\eta}_E^T P_E \eta_E + \eta_E^T P_E \dot{\eta}_E \quad (7.117)$$

将式(?)和(?)代入式(?)得到

$$\dot{V} = -x_0^T Q_L x_0 - \eta_E^T Q_E \eta_E + 2x_0^T P_L B \eta_E \quad (7.118)$$

其中

$$Q_L = -[A_L^T P_L + P_L A_L], \quad Q_E = K_{Ep} J$$

注意, 如果式(?)给出的充分条件成立, 很明显式(?)中的矩阵 Q_L 是正定的。利用 Q_L 是正定的事实使我们能够在式(?)中给出的 \dot{V} 上放置上界。这个上界由

$$\dot{V} \leq -\lambda_{\min}\{Q_L\}\|x_0\|^2 - \lambda_{\min}\{Q_E\}\|\eta_E\|^2 + 2\|P_L B\|\|x_0\|\|\eta_E\| \quad (7.119)$$

给出其中

$$\|P_L B\| \leq \frac{1}{2\sqrt{m_1}}$$

其中 m_1 在式(?)中定义。

为了确定渐近稳定性的控制器增益的充分条件, 我们将式(?)写成矩阵形式

$$\dot{V} \leq - \begin{bmatrix} \|x_0\| & \|\eta_E\| \end{bmatrix} Q \begin{bmatrix} \|x_0\| \\ \|\eta_E\| \end{bmatrix} \quad (7.120)$$

其中

$$Q = \begin{bmatrix} \lambda_{\min}\{Q_L\} & -\frac{1}{2\sqrt{m_1}} \\ -\frac{1}{2\sqrt{m_1}} & \lambda_{\min}\{Q_E\} \end{bmatrix}$$

由 Gerschgorin 定理, 如果满足

$$\lambda_{\min}\{Q_E\} > \frac{1}{2\sqrt{m_1}} \quad (7.121)$$

给出的充分条件, 则式(?)中定义的矩阵 Q 将为正定。因此, 如果控制器增益满足式(?)和(?)给出的条件, 我们可以使用标准 Lyapunov 稳定性论证 (见第 2 章) 来说明式(?)中定义的向量 x_0 , 因此 $\|e\|$ 、 \dot{e} 和 \ddot{e} 都是渐近稳定的。很容易表明, 如果满足

$$\lambda_{\min}\{K_{Ep}\} > \frac{1}{m_1} \quad (7.122)$$

给出的充分条件, 式(?)和(?)给出的条件总是满足的。

应该注意的是, 式(?)给出的控制取决于 \dot{u}_L 、 \ddot{u}_L 和 I 的测量。起初人们可能会想说这个控制器需要测量 \ddot{q} 和 I ; 然而, 由于我们假设精确知道式(?)和(?)给出的动力学模型, 我们可以使用这些信息来消除测量 \ddot{q} 的需要。也就是说, 通过对式(?)关于时间求导, \dot{u}_L 可以写成

$$\dot{u}_L = \dot{M}(q)(\ddot{q}_d + K_{Lv}\dot{e} + K_{Lp}e) + \bar{M}(q)(\ddot{q}_d + K_{Lv}\ddot{e} + K_{Lp}\dot{e}) + \dot{N}(q, \dot{q}) \quad (7.123)$$

其中 \ddot{q} 从式(?)中找到为

$$\ddot{q} = \bar{M}^{-1}(q)[K_T I - N(q, \dot{q})]$$

在式(?)中代入 \ddot{q} 后, \dot{u}_L 将仅取决于 q 、 \dot{q} 和 I 的测量。在控制输入 u_E 处实现的实际控制可以通过进行适当的代入找到。也就是说, 式(?)给出的校正控制可以写成

$$u_E = L_a K_T^{-1} J(\ddot{u}_L + K_{Ep}\eta_E) + K_T^{-1} J R(\dot{q}, I) + u_L \quad (7.124)$$

其中 \dot{u}_L 将由式(??)给出。

在仔细检查式(??)中给出的 \dot{u}_L 和 \ddot{u}_L 的函数依赖性后，现在很明显为什么我们假设期望轨迹和连杆动力学足够平滑。具体而言，从式(??)我们可以看到，校正控制器需要期望轨迹的第一、第二和第三时间导数有界，同时需要连杆动力学的第一导数存在。对期望轨迹和连杆动力学的这些假设确保控制输入保持有界。

上面推导的校正控制器总结在表 7.4.1 中，并在图 7.4.1 中描绘。我们现在提出一个示例来说明如何使用表 7.4.1 为 RLED 机器人设计校正控制器。

示例 7.4-1: 单连杆 RLED 机械臂的校正控制器

我们希望使用表 7.4.1 为图 7.4.2 中给出的单连杆电机驱动机器人臂设计和仿真校正控制器。系统的动力学采用

$$\bar{m}\ddot{q} + f_d\dot{q} + mgL \sin q = K_T I \quad (7.125)$$

和

$$L_a \dot{I} + RI + k_b \dot{q} = u_E \quad (7.126)$$

其中 $m = 1\text{kg}$, $K_T = 2\text{ N/A}$, $k_b = 0.3\text{ V-S}$, $f_d = 3\text{ kg-m/s}$, $L = 1\text{ m}$, $L_a = 0.1\text{ H}$, $R = 5\ \Omega$, g 是重力系数, $J = 0.2\text{ kg-m}^2$, I 是电机电流, u_E 是电机输入电压。

假设式 (1) 和 (2) 给出的模型精确知道，我们可以使用表 7.4.1 来制定校正控制器

$$u_E = \frac{L_a}{K_T}(\ddot{u}_L + K_{Ep}\eta_E) + \frac{R}{K_T}(\dot{u}_L - u_L) + \frac{k_b}{K_T}\dot{q} + u_L \quad (7.127)$$

其中

$$u_L = \bar{m}(\ddot{q}_d + K_{Lv}\dot{e} + K_{Lp}e) + f_d\dot{q} + mgL \sin q \quad (7.128)$$

其中 \ddot{u}_L 从式 (1) 中找到为

$$\ddot{u}_L = \frac{1}{\bar{m}}(\ddot{u}_L + K_{Ep}\eta_E) \quad (7.129)$$

校正控制器在以下条件下进行了仿真：控制参数、初始条件和期望轨迹由

$$K_{Lp} = K_{Lv} = K_{Ep} = 5, \quad q(0) = \dot{q}(0) = I(0) = 0$$

和

$$q_d = \sin t$$

给出

跟踪误差和控制电压在图 7.4.3 中描绘。如图所示，跟踪误差是渐近稳定的。

关节柔性

在本小节中，我们说明了如何合成“校正”控制器，以确保尽管驱动或齿轮会增加整体系统动力学的关节柔性，连杆跟踪也是渐近的。“校正控制器”一词用于强调控制

器校正用于表示关节柔性效应的动力学。本小节中研究的机器人将被称为刚连杆柔性关节 (RLFJ) 机器人。

RLFJ 机器人的模型 [Spong 1987] 采用

$$\ddot{q} = \bar{M}^{-1}(q)[K(q_m - q) - N(q, \dot{q})] \quad (7.130)$$

$$\ddot{q}_m = J^{-1}[u_F - B_m(q_m, \dot{q}_m) - K(q_m - q)] \quad (7.131)$$

其中

$$\bar{M}(q) = M(q) + J, \quad B_m(q_m, \dot{q}_m) = F_m \dot{q}_m \quad (7.132)$$

$q_m(t)$ 是表示电机位移的 $n \times 1$ 向量, K 是常数、对角、正定 $n \times n$ 关节柔性矩阵, $N(q, \dot{q})$ 是表示电机阻尼和柔性效应的 $n \times 1$ 向量, $u_F(t)$ 是用于表示输入扭矩的 $n \times 1$ 控制向量, 所有其他量如前一子节中定义。关于式(??)中给出的刚连杆模型, 我们从第 3 章注意到, 对于任何 $n \times 1$ 向量 x

$$\frac{1}{m_1} \leq \|\bar{M}^{-1}(q)\|_{i2} \leq \frac{1}{m_2} \quad (7.133)$$

其中 m_1 是正的标量常数。

如前一子节, 我们对式(??)中定义的连杆跟踪误差的性能感兴趣。对于 RLFJ 机器人的控制, 我们将假设 q_d 及其第一、第二、第三和第四导数都是作为时间函数有界的。我们还假设式(??)左侧连杆动力学的第一和第二导数存在。对期望轨迹和连杆动力学的这些“平滑性”假设确保稍后开发的控制器保持有界。

遵循前面各节给出的相同分析开发, 我们用式(??)给出的跟踪误差将式(??)写成状态空间形式

$$\dot{e} = A_0 e + B(u_L - K(q_m - q)) \quad (7.134)$$

其中 A_0 、 e 和 B 如式(??)中定义。同样, 由于式(??)中没有控制输入; 我们在式(??)的右侧添加和减去项 $B\bar{M}^{-1}(q)u_L$ 得到

$$\dot{e} = A_L e + B\bar{M}^{-1}(q)\eta_F \quad (7.135)$$

其中 u_L 再次用于表示虚构的 $n \times 1$ 控制输入。如前所述, 虚构控制器 u_L 将嵌入在整体控制策略中, 该策略设计在控制输入 u_F 处。

继续误差系统开发, 我们为 RLFJ 机器人定义 u_L 为计算扭矩控制器

$$u_L = \bar{M}(q)(\ddot{q}_d + K_{Lv}\dot{e} + K_{Lp}e) + N(q, \dot{q}) \quad (7.136)$$

其中 K_{Lv} 和 K_{Lp} 如式(??)中定义。将式(??)代入式(??)得到连杆跟踪误差系统

$$\dot{e} = A_L e + B\eta_F \quad (7.137)$$

其中 A_L 如式(??)中定义,

$$\eta_F = \bar{M}^{-1}(q)(u_L - K(q_m - q)) \quad (7.138)$$

以式(??)中给出的方式定义 η_F 的原因在于式(??)中给出的动力学是二阶动力学。也就是说, 由于执行器动力学是二阶的, 我们迫使 η_F 及其导数 (即 $\dot{\eta}_F$) 为零以确保连杆跟踪误差 e 趋近于零。

为了设计 η_F 的控制律, 我们必须首先建立其动态特性。从式(??), η_F 的导数为

$$\dot{\eta}_F = \bar{M}^{-1}(q)(\dot{u}_L - K(\dot{q}_m - \dot{q})) + \dot{\bar{M}}^{-1}(q)(u_L - K(q_m - q)) \quad (7.139)$$

为了获得 η_F 的动态特性, 我们从式(??)代入 \ddot{q}_m 到式(??)得到

$$\dot{\eta}_F = \bar{M}^{-1}(q)\dot{u}_L - \bar{M}^{-1}(q)KJ^{-1}[u_F - B_m(q_m, \dot{q}_m)] + \bar{M}^{-1}(q)K(q_m - q) + \dot{\bar{M}}^{-1}(q)\bar{M}(q)\eta_F \quad (7.140)$$

我们现在可以使用式(??)设计输入 u_F 的控制律以迫使 η_F 趋近于零。 η_F 应该趋近于零的事实推动了控制律

$$u_F = J\bar{M}^{-1}(q)[\dot{u}_L + K_{Fv}\eta_F + \bar{M}(q)K_{Fp}\eta_F + \dot{\bar{M}}^{-1}(q)\bar{M}(q)\eta_F] + B_m(q_m, \dot{q}_m) \quad (7.141)$$

其中 K_{Fv} 和 K_{Fp} 定义为 $n \times n$ 正定对角矩阵。将式(??)代入式(??)得到

$$\dot{\eta}_F = -K_{Fv}\eta_F - K_{Fp}\eta_F \quad (7.142)$$

其中

$$\eta_F = \bar{M}^{-1}(q)(u_L - K(q_m - q))$$

式(??)和(??)给出的动力学方程可以被认为代表整体闭环动力学的两个相互连接的子系统。为了确定闭环动力学的稳定性类型, 我们将利用 Lyapunov 函数

$$V = x_1^T P_L x_1 + \eta_F^T P_F \eta_F \quad (7.143)$$

其中 P_L 在式(??)中定义且

$$P_F = \frac{1}{2}KJ$$

如果满足

$$\lambda_{\min}\{K_{Lv}\} > 1 + \frac{1}{\lambda_{\min}\{K_{Fp}\}} \quad (7.144)$$

给出的充分条件, 则由 Gerschgorin 定理很明显式(??)中的矩阵 P_L 和 P_F 是正定矩阵, 因此 V 是 Lyapunov 函数。

对式(??)关于时间求导得到

$$\dot{V} = \dot{x}_1^T P_L x_1 + x_1^T P_L \dot{x}_1 + \dot{\eta}_F^T P_F \eta_F + \eta_F^T P_F \dot{\eta}_F \quad (7.145)$$

将式(??)和(??)代入式(??)得到

$$\dot{V} = -x_1^T Q_L x_1 - \eta_F^T Q_F \eta_F + 2x_1^T P_L B \eta_F \quad (7.146)$$

其中 Q_L 在式(??)中定义且

$$Q_F = K_{Fv} K J + K_{Fp} K J \quad (7.147)$$

注意, 如果式(??)给出的充分条件成立, 很明显式(??)中的矩阵 Q_L 和 Q_F 是正定矩阵。利用 Q_L 和 Q_F 是正定的事实使我们能够在式(??)中给出的 \dot{V} 上放置上界。这个上界由

$$\dot{V} \leq -\lambda_{\min}\{Q_L\}\|x_1\|^2 - \lambda_{\min}\{Q_F\}\|\eta_F\|^2 + \frac{1}{\sqrt{m_1}}\|x_1\|\|\eta_F\| \quad (7.148)$$

给出其中

$$m_1 = \lambda_{\min}\{\bar{M}(q)\}$$

其中 m_1 在式(??)中定义。

为了确定渐近稳定性的控制器增益的充分条件, 我们将式(??)写成矩阵形式

$$\dot{V} \leq - \begin{bmatrix} \|x_1\| & \|\eta_F\| \end{bmatrix} Q \begin{bmatrix} \|x_1\| \\ \|\eta_F\| \end{bmatrix} \quad (7.149)$$

其中

$$Q = \begin{bmatrix} \lambda_{\min}\{Q_L\} & -\frac{1}{2\sqrt{m_1}} \\ -\frac{1}{2\sqrt{m_1}} & \lambda_{\min}\{Q_F\} \end{bmatrix}$$

由 Gerschgorin 定理, 如果满足

$$\lambda_{\min}\{Q_F\} > \frac{1}{4m_1} \quad (7.150)$$

给出的充分条件, 则式(??)中定义的矩阵 Q 将为正定。因此, 如果控制器增益满足式(??)和(??)给出的条件, 我们可以使用标准 Lyapunov 稳定性论证来说明式(??)中定义的向量 x_1 , 因此 $\|e\|$ 、 \dot{e} 和 \ddot{e} 都是渐近稳定的。很容易表明, 如果满足

$$\lambda_{\min}\{K_{Fp}\} > \frac{1}{m_1} \quad (7.151)$$

给出的充分条件, 式(??)和(??)给出的条件总是满足的。

应该注意的是, 式(??)给出的校正控制器取决于 \dot{u}_L 、 \ddot{u}_L 和 \dot{q}_m 的测量。同样, 似乎这个控制需要测量 \ddot{q} 及其导数; 然而, 由于我们假设精确知道动力学模型, 我们可以使用这些信息来消除测量 \ddot{q} 及其导数的需要。也就是说, \dot{u}_L 可以写成

$$\dot{u}_L = \dot{\bar{M}}(q)(\ddot{q}_d + K_{Lv}\dot{e} + K_{Lp}e) + \bar{M}(q)(\ddot{q}_d + K_{Lv}\ddot{e} + K_{Lp}\dot{e}) + \dot{N}(q, \dot{q}) \quad (7.152)$$

其中 \ddot{q} 从式(??)中找到为

$$\ddot{q} = \bar{M}^{-1}(q)[K(q_m - q) - N(q, \dot{q})] \quad (7.153)$$

将式(??)代入式(??), 我们看到 u_L 可以作为 q 、 \dot{q} 、 q_m 和 t 的函数写出 (即 t), 因为期望轨迹可以明确地写成时间的函数。我们可以使用以下方程阐述这种函数依赖性

$$u_L = f(q, \dot{q}, q_m, t) \quad (7.154)$$

其中 $f(\cdot)$ 是式(??)右侧给出的 $n \times 1$ 向量。我们可以通过对式(??)关于时间求导获得 \ddot{u}_L 的函数依赖性

$$\ddot{u}_L = \dot{f}(q, \dot{q}, q_m, \dot{q}_m, t) \quad (7.155)$$

其中 $\dot{f}(\cdot)$ 是 $n \times 1$ 向量。注意式(??)可以再次用于消除测量 \ddot{q} 的需要; 因此, 式(??)和(??)中给出的 \dot{u}_L 和 \ddot{u}_L 的表达式仅取决于 q 、 \dot{q} 、 q_m 和 \dot{q}_m 的测量。在控制输入 u_F 处实现的实际控制可以通过进行适当的代入找到。也就是说, 式(??)给出的校正控制可以写成

$$u_F = J\bar{M}^{-1}(q)[\dot{u}_L + K_{Fv}\eta_F + \bar{M}(q)K_{Fp}\eta_F + \dot{\bar{M}}^{-1}(q)\bar{M}(q)\eta_F] + B_m(q_m, \dot{q}_m) \quad (7.156)$$

其中 \dot{u}_L 和 \ddot{u}_L 将分别由式(??)和(??)给出。

在仔细检查式(??)和(??)中给出的 \dot{u}_L 和 \ddot{u}_L 的函数依赖性后, 很明显为什么我们假设期望轨迹和连杆动力学足够平滑。具体而言, 从式(??)和(??)我们可以看到, 式(??)中给出的校正控制器将需要期望轨迹的第一、第二、第三和第四时间导数有界, 同时需要连杆动力学的第一和第二导数存在。对期望轨迹和连杆动力学的这些假设确保控制输入保持有界。

上面推导的校正控制器总结在表 7.4.2 中, 并在图 7.4.4 中描绘。我们现在提出一个示例来说明如何使用表 7.4.2 为 RLFJ 机器人设计校正控制器。

示例 7.4-2: 单连杆 RLFJ 机械臂的校正控制器

我们希望使用表 7.4.2 为图 7.4.5 中给出的单连杆柔性关节机器人臂设计和仿真校正控制器。系统的动力学采用

$$\bar{m}\ddot{q} + f_d\dot{q} + mgL \sin q = K(q_m - q) \quad (7.157)$$

和

$$J\ddot{q}_m + B\dot{q}_m + K(q_m - q) = u_F \quad (7.158)$$

其中 $m = 1 \text{ kg}$, $K = 10 \text{ N}$, $B = 5 \text{ kg}\cdot\text{m/s}$, $f_d = 3 \text{ kg}\cdot\text{m/s}$, $L = 1 \text{ m}$, g 是重力系数, $J = 0.2 \text{ kg}\cdot\text{m}^2$, q_m 是以弧度测量的电机位移, u_F 是输入扭矩。

假设式 (1) 和 (2) 给出的模型精确知道, 我们可以使用表 7.4.2 来制定校正控制器

$$u_F = J\bar{m}^{-1}[\ddot{u}_L + K_{Fv}\eta_F + \bar{m}K_{Fp}\eta_F] + B\dot{q}_m \quad (7.159)$$

其中

$$\eta_F = \bar{m}^{-1}(u_L - K(q_m - q)) \quad (7.160)$$

其中 \dot{u}_L 从式 (1) 中找到为

$$\dot{u}_L = \bar{m}(\ddot{q}_d + K_{Lv}\ddot{e} + K_{Lp}\dot{e}) + f_d\ddot{q} + mgL \cos q \cdot \dot{q} \quad (7.161)$$

\ddot{q} 从式 (1) 中找到为

$$\ddot{q} = \frac{1}{\bar{m}}[K(q_m - q) - f_d\dot{q} - mgL \sin q] \quad (7.162)$$

为获得式 (3) 中给出的 \ddot{u}_L 的表达式, 我们将式 (5) 代入式 (4) 得到

$$\dot{u}_L = \bar{m}(\ddot{q}_d + K_{Lv}\ddot{e} + K_{Lp}\dot{e}) + f_d\frac{1}{\bar{m}}[K(q_m - q) - f_d\dot{q} - mgL \sin q] + mgL \cos q \cdot \dot{q} \quad (7.163)$$

对式 (6) 关于时间求导得到

$$\ddot{u}_L = \bar{m}(q_d^{(4)} + K_{Lv}\ddot{e} + K_{Lp}\dot{e}) + f_d\ddot{q} + mgL(\cos q \cdot \ddot{q} - \sin q \cdot \dot{q}^2) \quad (7.164)$$

其中 \ddot{q} 从式 (5) 中找到。

校正控制器在以下条件下进行了仿真: 控制参数、初始条件和期望轨迹由

$$K_{Lp} = K_{Lv} = 5, \quad K_{Fp} = K_{Fv} = 10$$

和

$$q_d = \sin t, \quad q(0) = \dot{q}(0) = q_m(0) = \dot{q}_m(0) = 0$$

给出

跟踪误差和控制扭矩在图 7.4.6 中描绘。如图所示, 跟踪误差是渐近稳定的。

7.5 总结

在本章中, 给出了机器人机械臂控制的几种更先进控制技术的描述。本章的意图是研究减少在线计算的控制器和补偿执行器动力学的控制器。一些当前的研究问题涉及力控制器与高级运动控制器的集成以及相应的数字实现。

参考文献

- [1] Corless, M., and G. Leitmann, "Adaptive control of systems containing uncertain functions and unknown functions with uncertain bounds," *J. Optim. Theory Appl.*, Jan. 1983.
- [2] Dawson, D.M., Z. Qu, F.L. Lewis, and J.F. Dorsey, "Robust control for the tracking of robot motion," *Int. J. Control*, vol. 52, pp. 581-595, 1990.

- [3] Eppinger, S., and W. Seering, "Introduction to Dynamic models for robot force control," *IEEE Control Syst. Mag.*, vol. 7, no. 2, pp. 48–52, Apr. 1987.
- [4] Ghorbel, F., and M. Spong, "Stability analysis of adaptively controlled flexible joint robots," *Proc. IEEE Conf. Decision Control*, Honolulu, pp 2538–2544, 1990.
- [5] Kokotovic, P.V., H. Khalil, and J. O'Reilly, *Singular Perturbation Methods in Control Analysis and Design*. New York: Academic Press, 1986.
- [6] Sadegh, N., and R. Horowitz, "Stability and robustness analysis of a class of adaptive controllers for robotic manipulators," *Int. J. Robot. Res.*, vol. 9, no. 3, pp. 74–92, June 1990.
- [7] Sadegh, N., R. Horowitz, W. Kao, and M. Tomizuka, "A unified approach to the design of adaptive and repetitive controllers for robotic manipulators," *Trans. ASME*, vol. 112, pp. 618–629, Dec. 1990.
- [8] Spong, M., "Modeling and control of elastic joint robots," *J. Dyn. Syst., Meas. Control*, vol. 109, pp. 310–319, Dec. 1987.
- [9] Tarn, T., A. Bejczy, X. Yun, and Z. Li, "Effect of motor dynamics on nonlinear feedback robot arm control," *IEEE Trans. Robot. Autom.*, vol. 7, pp. 114–122, Feb. 1991.
- [10] Taylor, D., "Composite control of direct-drive robots," *Proc. IEEE Conf. Decision Control*, pp. 1670–1675, Dec. 1989.

习题

第 7.2 节

7.2-1 如果滤波跟踪误差定义为

$$r = \dot{e} + \Lambda e$$

其中 Λ 是正定对角矩阵, 说明如何修改 DCAL 稳定性分析。

7.2-2 为第 3 章中给出的二连杆极坐标机器人臂设计和仿真表 7.2.1 中给出的 DCAL 控制器。(忽略该机器人有棱柱形连杆的事实。)

7.2-3 DCAL 稳定性分析 (以及因此控制器本身) 能否修改为考虑问题 7.2-2 中给出的棱柱形连杆机器人? 如果可以, 说明如何修改。

7.2-4 如果学习项 $\hat{u}_d(t)$ 被迫保持在先验界限内

$$\hat{u}_{d\min} \leq \hat{u}_{di}(t) \leq \hat{u}_{d\max}$$

其中下标 i 用于表示 $n \times 1$ 向量 $\hat{u}_d(t)$ 的第 i 个分量, $\hat{u}_{d\min}$ 、 $\hat{u}_{d\max}$ 是标量常数, 说明如何修改 RCL 稳定性分析以表明速度跟踪误差是渐近稳定的。

7.2-5 为第 3 章中给出的二连杆极坐标机器人臂设计和仿真表 7.2.2 中给出的 RCL 控制器。(忽略该机器人有棱柱形连杆的事实。)

7.2-6 RCL 稳定性分析 (以及因此控制器本身) 能否修改为考虑问题 7.2-5 中给出的棱柱形连杆机器人? 如果可以, 说明如何修改。

第 7.3 节

7.3-1 为第 3 章中给出的二连杆极坐标机器人臂设计和仿真表 7.3.1 中给出的自适应鲁棒控制器。(忽略该机器人有棱柱形连杆的事实。)

7.3-2 自适应鲁棒控制器稳定性分析 (以及因此控制器本身) 能否修改为考虑问题 7.3-1 中给出的棱柱形连杆? 如果可以, 说明如何修改。

7.3-3 说明如何使用 Barbalat 引理 (见第 2 章) 修改自适应鲁棒控制器的稳定性分析以保证速度跟踪误差也是渐近稳定的。

第 7.4 节

7.4-1 对于常数、对称、正定、 $n \times n$ 矩阵 A , 表明

$$(a) \|A\|_{i2} = \lambda_{\max}\{A\}$$

$$(b) \lambda_{\min}\{A^{-1}\} = \frac{1}{\lambda_{\max}\{A\}}$$

7.4-2 为第 3 章中给出的二连杆旋转机器人臂设计和仿真表 7.4.1 中给出的 RLED 校正控制器。假设两个电机都可以建模为示例 7.4-1 中给出的电机。

7.4-3 对于常数、对称、正定、 $n \times n$ 矩阵 A , 表明

$$A^{-1} = \begin{bmatrix} O_{n \times n} & I_{n \times n} \\ -A_{21} & -A_{22} \end{bmatrix}$$

其中 $O_{n \times n}$ 是 $n \times n$ 零矩阵。

7.4-4 为第 3 章中给出的二连杆旋转机器人臂设计和仿真表 7.4.2 中给出的 RLFJ 校正控制器。假设两个关节都可以建模为类似于示例 7.4-2 中给出的关节。

Chapter 8

机器人的神经网络控制

本章给出了使用神经网络进行控制器设计的框架。这些结构具有通用逼近性质，使其能够在未知系统的反馈控制中使用，而无需系统参数线性化的要求或寻找回归矩阵。本章给出了能够保证闭环稳定性和有界权重的反馈控制拓扑结构和权重调整算法。

8.1 引言

近年来，在设计模仿生物系统功能的反馈控制系统方面进行了大量工作 [?, ?]。最近人们对“通用无模型控制器”产生了极大兴趣，这种控制器不需要被控对象的数学模型，而是模仿生物过程的功能，在线学习它们正在控制的系统，从而自动提高性能。相关技术包括模糊逻辑控制——它模仿语言和推理功能，以及人工神经网络——它基于相互连接节点的生物神经元结构。神经网络 (NN) 在分类、模式识别以及数字信号处理和其他领域的开环应用方面取得了巨大成功。严格的分析已经展示了如何选择神经网络拓扑结构和权重，例如，用于区分指定的示例模式。到目前为止，神经网络在开环应用中的理论和应用已经得到了很好的理解，因此神经网络已成为信号处理人员和计算机科学家工具箱中的重要工具。

关于使用神经网络对未知对象进行反馈控制，已有大量文献。直到 20 世纪 90 年代，设计和分析技术都是临时性的，没有可重复的设计算法或稳定性和保证性能的证明。尽管如此，文献中出现的仿真结果仍显示了良好的性能。大多数早期方法使用标准的反向传播权重调整 [?], 因为当时适用于闭环控制目的的严格调整算法推导尚不可用。许多神经网络设计技术模仿自适应控制方法，而自适应控制方法已有严格的分析结果可用 [?, ?, ?], 提出了基于间接辨识的控制、逆动力学控制、串并联技术等神经网络反馈控制拓扑结构。

按照 Maxwell、Lyapunov、A.N.Whitehead、von Bertalanffy 等人从事控制系统理论研究的哲学，要解决这些问题，必须从被控系统的已有知识开始。Narendra [?, ?] 和其他人 [?, ?, ?] 通过研究闭环系统中神经网络的动力学行为，包括计算反向传播调整所

需的梯度，开创了严格的神经网络控制应用。其他为闭环神经网络控制器提供严格分析和设计技术的团队包括：[?] 展示了如何在反馈控制中使用径向基函数神经网络；[?, ?] 提供了基于死区方法的神经网络调整算法；[?, ?] 使用了投影方法；[?, ?] 使用了 e-mod 方法；[?] 为离散时间系统提供了神经网络控制器；[?] 使用动态神经网络进行反馈控制。

本章给出了一种基于几篇博士论文和一系列已发表工作的神经网络控制器设计和分析方法，包括 [?, ?]。这里讨论的控制结构是多环控制器，其中一些环路包含神经网络，外部是跟踪单位增益 PD 环路。有可重复的设计算法和系统性能保证，包括小跟踪误差和有界神经网络权重。研究表明，随着对被控系统不确定性的增加或性能要求的提高，神经网络控制器需要越来越多的结构。更深入的阐述见于 [?]，其中还展示了与模糊逻辑系统的关系。

8.2 神经网络背景

本节提供了反馈控制所需的神经网络结构背景知识。更多细节请参见 [?, ?, ?]。使神经网络对反馈控制有用的两个关键特征是其通用逼近性质和学习能力，后者源于其权重是可调参数，可以更新以提高控制器性能。通用逼近性质是使非线性网络结构比自适应机器人控制器更适合机器人控制的主要特征，后者通常依赖于确定回归矩阵，而这又要求可调参数线性化 (LIP)。

8.2.1 多层神经网络

一个多层神经网络如图 8.2.1 所示。该神经网络有两层可调权重，这里称为“两层”网络。该神经网络没有内部反馈连接，因此称为前馈网络；没有内部动态，因此称为静态网络。神经网络输出 \mathbf{y} 是一个具有 m 个分量的向量，由回忆方程根据具有 n 个分量的输入向量 \mathbf{x} 确定：

$$y_i = \sum_{j=1}^L w_{ij} \sigma \left(\sum_{k=1}^n v_{jk} x_k + \bar{v}_j \right) + \bar{w}_i, \quad i = 1, \dots, m \quad (8.1)$$

其中 $\sigma(\cdot)$ 是激活函数， L 是隐层神经元数量。第一层互联权重用 v_{jk} 表示，第二层互联权重用 w_{ij} 表示。阈值偏移用 \bar{v}_j 、 \bar{w}_i 表示。

有许多不同的激活函数 $\sigma(\cdot)$ 在广泛使用。对于使用多层神经网络的反馈控制，要求 $\sigma(\cdot)$ 足够光滑，使得至少其一阶导数存在。合适的选择包括 sigmoid 函数

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (8.2)$$

双曲正切函数

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (8.3)$$

以及其他逻辑曲线型函数，还有高斯函数和其他各种函数。

通过将所有神经网络权重 v_{jk} 、 w_{ij} 收集到权重矩阵 \mathbf{V}^T 、 \mathbf{W}^T 中，神经网络回忆方程可以用向量形式写成：

$$\mathbf{y} = \mathbf{W}^T \sigma(\mathbf{V}^T \mathbf{x}) \quad (8.4)$$

阈值包含在权重矩阵 \mathbf{W}^T 、 \mathbf{V}^T 的第一列中；为了适应这一点，向量 \mathbf{x} 和 $\sigma(\cdot)$ 需要通过在前面放置一个“1”来增广（例如 $\mathbf{x} = [1, x_1, x_2, \dots, x_n]^T$ ）。在这个方程中，为了表示式(??)，如果 $\sigma(\cdot)$ 取为从 \mathbb{R}^L 到 \mathbb{R}^L 的对角函数，即对于向量 $\mathbf{z} = [z_1, z_2, \dots, z_L]^T \in \mathbb{R}^L$ 有 $\sigma(\mathbf{z}) = \text{diag}\{\sigma(z_j)\}$ ，则具有足够的通用性。

注意，回忆方程在第一层权重和阈值 \mathbf{V} 方面是非线性的。

通用函数逼近性质。神经网络满足许多重要性质。对于反馈控制目的而言，主要关注的是通用函数逼近性质 [?]。设 $f(\mathbf{x})$ 是一个从 \mathbb{R}^n 到 \mathbb{R}^m 的一般光滑函数。那么可以证明，只要 \mathbf{x} 被限制在 \mathbb{R}^n 的紧集 S 内，就存在权重和阈值，使得

$$f(\mathbf{x}) = \mathbf{W}^T \sigma(\mathbf{V}^T \mathbf{x}) + \varepsilon \quad (8.5)$$

对于一定数量的隐层神经元 L 。这对于一大类激活函数都成立，包括刚才提到的那些。值 ε 称为神经网络函数逼近误差，它通常随着网络规模 L 的增加而减小。事实上，对于任何选择的正数 ε_M ，可以找到前馈神经网络使得

$$\|\varepsilon\| < \varepsilon_M \quad (8.6)$$

对于 S 中所有 \mathbf{x} 。为给定精度 ε_M 选择网络规模 L 的问题对于一般无结构全连接神经网络仍是一个开放问题。对于 CMAC、FL 网络和其他结构化非线性网络，这个问题可以随后解决。

最佳逼近给定非线性函数 $f(\mathbf{x})$ 的理想神经网络权重矩阵 \mathbf{W} 、 \mathbf{V} 难以确定。事实上，它们甚至可能不唯一。然而，对于控制目的，人们只需要知道，对于指定的 ε_M 值，存在某些理想逼近神经网络权重。然后，可以用下式给出 $f(\mathbf{x})$ 的估计

$$\hat{f}(\mathbf{x}) = \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T \mathbf{x}) \quad (8.7)$$

其中 $\hat{\mathbf{W}}$ 、 $\hat{\mathbf{V}}$ 是理想神经网络权重的估计，由随后详细介绍的在线权重调整算法提供。注意，所有“带帽”量都是已知的，因为它们可以用可测量信号和当前神经网络权重估计来计算。

存在使逼近性质成立的理想权重的假设非常类似于自适应控制中的各种类似假设 [?, ?]，包括 Erzberger 假设和参数线性化。非常重要的区别在于，在神经网络情况下，通用逼近性质总是成立的，而在自适应控制中，这些假设在实践中往往不成立，因此它们意味着对被控系统形式的限制。

注：读者可以通过[交互式可视化工具](#)观察神经网络学习非线性函数的过程，理解隐藏层神经元数量、学习率对逼近精度的影响。

克服 NLIP 问题。多层神经网络在权重 \mathbf{V} 方面是非线性的，因此产生保证闭环反馈系统中稳定性和有界权重的权重调整算法直到几年前才被发现。如果采取正确和严格的方法，NLIP 问题很容易克服。一种方法是以下对泰勒级数的适当使用 [?]

定义函数估计误差

$$\tilde{f} = f - \hat{f} = f - \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T \mathbf{x}) \quad (8.8)$$

权重估计误差

$$\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}, \quad \tilde{\mathbf{V}} = \mathbf{V} - \hat{\mathbf{V}} \quad (8.9)$$

和隐层输出误差

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(\mathbf{V}^T \mathbf{x}) - \sigma(\hat{\mathbf{V}}^T \mathbf{x}) \quad (8.10)$$

对于任何 \mathbf{z} ，可以写出泰勒级数

$$\sigma(\mathbf{z}) = \sigma(\hat{\mathbf{z}}) + \sigma'(\hat{\mathbf{z}})\tilde{\mathbf{z}} + O(\tilde{\mathbf{z}})^2 \quad (8.11)$$

其中 σ' 是雅可比矩阵，最后一项表示 $\tilde{\mathbf{z}}^2$ 阶的项。因此

$$\tilde{\sigma} = \sigma'(\hat{\mathbf{V}}^T \mathbf{x})\tilde{\mathbf{V}}^T \mathbf{x} + O(\tilde{\mathbf{V}}^T \mathbf{x})^2 \quad (8.12)$$

这个关键方程允许将函数估计误差写成

$$\tilde{f} = \tilde{\mathbf{W}}^T [\hat{\sigma} - \hat{\sigma}' \hat{\mathbf{V}}^T \mathbf{x}] + \hat{\mathbf{W}}^T \hat{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} + w \quad (8.13)$$

其中 w 包含高阶项。第一项有 $\tilde{\mathbf{W}}$ 乘以已知量（方括号内），第二项有 $\tilde{\mathbf{V}}$ 乘以已知量。当随后在闭环误差动力学中使用时，这种形式允许导出保证闭环稳定性的 \mathbf{V} 和 \mathbf{W} 的调整律。

8.2.2 线性参数神经网络

如果式 (??) 中的第一层权重和阈值 \mathbf{V} 固定，只调整第二层权重和阈值 \mathbf{W} ，那么神经网络只有一层可调权重。可以定义固定函数 $\phi(\mathbf{x}) = \sigma(\mathbf{V}^T \mathbf{x})$ ，使得这种单层神经网络具有回忆方程

$$y = \mathbf{W}^T \phi(\mathbf{x}) \quad (8.14)$$

其中 $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^L$ ， L 是隐层神经元数量。该神经网络在可调参数 \mathbf{W} 方面是线性的，因此更容易调整。事实上，可以使用标准的自适应控制证明来导出合适的调整算法。

虽然是 LIP，但这些神经网络相比要求确定回归矩阵的标准自适应控制方法仍有巨大优势，因为如果正确选择 $\phi(\cdot)$ 函数，它们满足通用逼近性质。注意，自适应控制器在系统参数方面是线性的，而单层神经网络在神经网络权重方面是线性的。单层神经网络相比双层神经网络的一个优势是，对于前者存在关于为指定逼近精度选择隐层神经元数量 L 的确定结果。LIP 网络的一个缺点是 Barron [?] 已经证明了逼近精度的下界。

功能连接基神经网络。如果 $\sigma(\cdot)$ 不是对角矩阵，而是允许 $\phi(\cdot)$ 是从 \mathbb{R}^n 到 \mathbb{R}^L 的一般函数，则获得更大的通用性。这称为功能连接神经网络 (FLNN) [?]。对于 LIP 神经网络，函数逼近性质通常不成立。然而，只要将激活函数 $\phi(\cdot)$ 选择为基，单层神经网络仍然可以逼近函数，基必须在 \mathbb{R}^n 的紧单连通集 S 上满足以下两个要求：

1. S 上的常数函数可以表示为式 (??) 的形式，对于有限数量 L 的隐层神经元。
2. 式 (??) 的函数值域在从 S 到 \mathbb{R}^m 的连续函数空间中是稠密的，对于可数 L 。

如果 $\phi(\cdot)$ 提供基，那么从 \mathbb{R}^n 到 \mathbb{R}^m 的光滑函数 $f(\mathbf{x})$ 可以在 \mathbb{R}^n 的紧集 S 上逼近为

$$f(\mathbf{x}) = \mathbf{W}^T \phi(\mathbf{x}) + \varepsilon \quad (8.15)$$

对于某些理想权重和阈值 \mathbf{W} 和某些隐层神经元数量 L 。事实上，对于任何选择的正数 ε_M ，可以找到前馈神经网络使得 $\|\varepsilon\| < \varepsilon_M$ 对于 S 中所有 \mathbf{x} 。

$f(\mathbf{x})$ 的逼近由下式给出

$$\hat{f}(\mathbf{x}) = \hat{\mathbf{W}}^T \phi(\mathbf{x}) \quad (8.16)$$

其中 $\hat{\mathbf{W}}$ 是由调整算法给出的神经网络权重和阈值估计。对于 LIP 神经网络，函数估计误差由下式给出

$$\tilde{f} = \tilde{\mathbf{W}}^T \phi(\mathbf{x}) + \varepsilon \quad (8.17)$$

当随后在误差动力学中使用，这直接产生了保证闭环稳定性的 \mathbf{W} 的调整算法。

Barron [?] 已经证明，对于所有 LIP 逼近器存在一个基本下界，因此 ε 被 $1/L^{2/n}$ 阶的项下界所限制。因此，随着神经网络输入 n 的增加，增加 L 以提高逼近精度变得不那么有效。如下所示，这对于使用单层神经网络设计的自适应控制器不是一个主要限制。这个下界问题在 NLIP 多层非线性网络中不存在。

我们经常用 $\sigma(\cdot)$ 代替 $\phi(\cdot)$ ，理解对于 LIP 网络，这个激活函数向量不是对角矩阵，而是从 \mathbb{R}^L 到 \mathbb{R}^L 的一般函数。

高斯或径向基函数 (RBF) 网络。在各种结构化非线性网络中，包括径向基函数、CMAC 和模糊逻辑网络，激活函数基集的选择大大简化。这里将展示这种结构化非线性网络设计的关键在于比标准方程 (??) 中允许的更一般的神经网络阈值集，以及它们的适当选择。

常用的神经网络激活函数是高斯或径向基函数 (RBF) [?]，当 \mathbf{x} 为标量时由下式给出

$$\sigma(x) = \exp\left(-\frac{(x - \bar{x})^2}{2p}\right) \quad (8.18)$$

其中 \bar{x} 是均值， p 是方差。RBF 神经网络可以写成式 (??) 的形式，但相比通常的 sigmoid 神经网络有一个优势，即 n 维高斯函数从概率论、卡尔曼滤波和其他地方得到了很好的理解，因此 n 维 RBF 易于概念化。

第 j 个激活函数可以写成

$$\sigma_j(\mathbf{x}) = \exp [-(\mathbf{x} - \bar{\mathbf{x}}_j)^T \mathbf{P}_j^{-1} (\mathbf{x} - \bar{\mathbf{x}}_j)] \quad (8.19)$$

其中 $\mathbf{P}_j = \text{diag}\{p_{jk}\}$ 。定义激活函数向量为 $\sigma(\mathbf{x}) \equiv [\sigma_1(\mathbf{x}), \sigma_2(\mathbf{x}), \dots, \sigma_L(\mathbf{x})]^T$ 。如果协方差矩阵是对角的, 使得 $\mathbf{P}_j = \text{diag}\{p_{jk}\}$, 那么式 (??) 变为可分离的, 可以分解为分量形式

$$\sigma_j(\mathbf{x}) = \prod_{k=1}^n \exp \left[-\frac{(x_k - \bar{x}_{jk})^2}{2p_{jk}} \right] = \prod_{k=1}^n \sigma_{jk}(x_k) \quad (8.20)$$

其中 $\sigma_{jk}(x_k)$ 是 $\sigma_j(\mathbf{x})$ 的第 k 个分量。因此, n 维激活函数是 n 个标量函数的乘积。注意, 这个方程具有式 (??) 中激活函数的形式, 但有更一般的阈值, 因为在每个隐层神经元 j 处, \mathbf{x} 的每个不同分量都需要一个阈值; 即图 8.2.1 中每个隐层神经元的阈值是一个向量。RBF 方差 p_{jk} 和偏移 \bar{x}_{jk} 通常在 RBF 神经网络设计中选择并固定; 通常只调整输出层权重 \mathbf{W}^T 。因此, RBF 神经网络是一种特殊的 FLNN (式 (??)) (其中 $\phi(\mathbf{x}) = \sigma(\mathbf{x})$)。

图 8.2.2 显示了 $\mathbf{x} \in \mathbb{R}^2$ 情况下的可分离高斯函数。在这个图中, 所有方差 p_{jk} 相同, 均值 \bar{x}_{jk} 以特殊方式选择, 在 2 维网格的节点点上间隔激活函数。为了形成在区域 $\{-1 < x_1 \leq 1, -1 < x_2 \leq 1\}$ 上逼近函数的 RBF 神经网络, 这里选择 $L = 5 \times 5 = 25$ 个隐层神经元, 沿 x_1 有 5 个单元, 沿 x_2 有 5 个单元。其中九个神经元具有 2 维高斯激活函数, 而沿边界的需要图示的”单侧”激活函数。

8.3 使用静态神经网络的跟踪控制

本节讨论使用非线性网络和假设全状态变量反馈的反馈跟踪控制设计。更多细节请参见 [?] 和其他列出的参考文献。如果全状态反馈可用, 则静态前馈神经网络足以进行控制。这里提供了保证闭环稳定性和有界权重的控制拓扑和网络调整算法。所讨论的技术适用于一般非线性网络, 包括神经网络和模糊逻辑系统 [?, ?], 因此缩写 NN 此后可理解为指”非线性网络”。所得的多环控制拓扑具有外部 PD 跟踪环路和内部神经网络反馈线性化或动作生成环路。发现反向传播调整通常不够, 需要修改的调整算法来保证闭环性能。

许多工业机械系统以及汽车、飞机和航天器具有拉格朗日形式的动力学, 其典型代表是刚性机器人系统。因此, 将考虑拉格朗日机器人动力学 [?, ?]。这里介绍的神经网络控制技术也可应用于其他未知系统, 包括某些重要的非线性系统类别 [?]

8.3.1 机器人手臂动力学和误差系统

刚性拉格朗日系统的动力学, 包括机器人手臂, 具有一些重要的物理、结构和无源性性质 [?, ?, ?, ?], 使得在其控制中使用神经网络非常自然。在设计任何控制器时都应考虑这些性质——事实上, 它们为神经网络控制器的严格设计算法提供了基础。

n 连杆刚性（即无柔性连杆或高频关节/电机动力学）机器人操作臂的动力学可以用拉格朗日形式表示为

$$M(\mathbf{q})\ddot{\mathbf{q}} + V_m(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) + F(\dot{\mathbf{q}}) + \boldsymbol{\tau}_d = \boldsymbol{\tau} \quad (8.21)$$

其中 $\mathbf{q}(t) \in \mathbb{R}^n$ 是关节变量向量，其元素是机器人手臂关节角度或连杆伸长。 $M(\mathbf{q})$ 是惯性矩阵， $V_m(\mathbf{q}, \dot{\mathbf{q}})$ 是科里奥利/向心矩阵， $G(\mathbf{q})$ 是重力向量， $F(\dot{\mathbf{q}})$ 是摩擦力。有界的未知扰动（包括例如非结构化未建模动力学）用 $\boldsymbol{\tau}_d$ 表示，控制输入力矩是 $\boldsymbol{\tau}(t)$ 。机器人动力学具有以下标准性质：

性质 1: $M(\mathbf{q})$ 是正定对称矩阵，有界于 $m_1\mathbf{I} \leq M(\mathbf{q}) \leq m_2\mathbf{I}$ ，其中 m_1 、 m_2 是正常数。

性质 2: 矩阵 $V_m(\mathbf{q}, \dot{\mathbf{q}})$ 的范数被 $v_b(\|\mathbf{q}\|)\|\dot{\mathbf{q}}\|$ 所限制，对于某个函数 $v_b(\mathbf{q})$ 。

性质 3: 矩阵 $\dot{M} - 2V_m$ 是斜对称的。这等价于内部力不做功的事实。

性质 4: 未知扰动满足 $\|\boldsymbol{\tau}_d\| \leq d_B$ ，其中 d_B 是正常数。

给定期望的手臂轨迹 $\mathbf{q}_d(t) \in \mathbb{R}^n$ ，跟踪误差为

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t) \quad (8.22)$$

滤波跟踪误差为

$$\mathbf{r}(t) = \dot{\mathbf{e}}(t) + \boldsymbol{\Lambda}\mathbf{e}(t) \quad (8.23)$$

其中 $\boldsymbol{\Lambda}$ 是对称正定设计参数矩阵，通常选择为对角矩阵。如果找到控制器使得 $\mathbf{r}(t)$ 有界，则 $\mathbf{e}(t)$ 也有界；事实上 $\|\mathbf{e}\| \leq \frac{\|\mathbf{r}\|}{\sigma_{\min}(\boldsymbol{\Lambda})}$ ，其中 $\sigma_{\min}(\boldsymbol{\Lambda})$ 是 $\boldsymbol{\Lambda}$ 的最小奇异值。

对 $\mathbf{r}(t)$ 求导并使用式 (??)，可以用滤波跟踪误差的形式写出手臂动力学

$$M\dot{\mathbf{r}} = -V_m\mathbf{r} - \boldsymbol{\tau} + \mathbf{f}(\mathbf{x}) \quad (8.24)$$

其中非线性机器人函数为

$$\mathbf{f}(\mathbf{x}) = M(\mathbf{q})(\ddot{\mathbf{q}}_d + \boldsymbol{\Lambda}\dot{\mathbf{e}}) + V_m(\mathbf{q}, \dot{\mathbf{q}})(\dot{\mathbf{q}}_d + \boldsymbol{\Lambda}\mathbf{e}) + G(\mathbf{q}) + F(\dot{\mathbf{q}}) \quad (8.25)$$

计算 $\mathbf{f}(\mathbf{x})$ 所需的向量 \mathbf{x} 可以定义为，例如

$$\mathbf{x} = \left[\mathbf{e}^T \quad \dot{\mathbf{e}}^T \quad \mathbf{q}_d^T \quad \dot{\mathbf{q}}_d^T \quad \ddot{\mathbf{q}}_d^T \right]^T \quad (8.26)$$

这是可测量的。函数 $\mathbf{f}(\mathbf{x})$ 包含潜在未知的机器人参数，包括负载质量和复杂的摩擦形式。

用于轨迹跟踪的合适控制输入由计算力矩型控制给出

$$\boldsymbol{\tau} = \hat{\mathbf{f}}(\mathbf{x}) + K_v\mathbf{r} - \mathbf{v}(t) \quad (8.27)$$

其中 K_v 是增益矩阵, 通常选择为对角矩阵, $\hat{\mathbf{f}}(\mathbf{x})$ 是由某种方式提供的机器人函数 $\mathbf{f}(\mathbf{x})$ 的估计。鲁棒化信号 $\mathbf{v}(t)$ 需要补偿未建模非结构化扰动。使用这个控制, 闭环误差动力学为

$$M\dot{\mathbf{r}} = -(V_m + K_v)\mathbf{r} + \hat{\mathbf{f}} + \boldsymbol{\tau}_d + \mathbf{v} \quad (8.28)$$

在计算控制信号时, 估计 $\hat{\mathbf{f}}$ 可以由几种技术提供, 包括自适应控制或神经或模糊网络。辅助控制信号 $\mathbf{v}(t)$ 可以由几种技术选择, 包括滑模方法和其他鲁棒控制方法。

假设期望轨迹有界, 使得

$$\begin{bmatrix} \mathbf{q}_d \\ \dot{\mathbf{q}}_d \\ \ddot{\mathbf{q}}_d \end{bmatrix} \leq \mathbf{q}_B \quad (8.29)$$

其中 \mathbf{q}_B 是已知标量界。容易证明对于每个时刻 t , $\mathbf{x}(t)$ 被限制为

$$\|\mathbf{x}\| \leq c_0 + c_1\|\mathbf{r}\| + c_2\|\mathbf{r}\|^2 \equiv q_B \quad (8.30)$$

对于可计算的正常数 c_0 、 c_1 、 c_2 。

8.3.2 自适应控制

机器人学中的标准自适应控制技术 [?, ?, ?, ?] 假设非线性机器人函数在可调参数方面是线性的, 使得

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\phi(\mathbf{x}) + \varepsilon \quad (8.31)$$

其中 \mathbf{W} 是未知参数向量, $\phi(\mathbf{x})$ 是已知回归矩阵。控制选择为

$$\boldsymbol{\tau} = \hat{\mathbf{W}}^T\phi(\mathbf{x}) + K_v\mathbf{r} - \mathbf{v}(t) \quad (8.32)$$

并为参数估计 $\hat{\mathbf{W}}$ 确定调整算法。这由 LIP 假设促进。如果逼近误差 ε 非零, 则必须修改调整算法以保证有界的参数估计。可以使用各种鲁棒化技术, 包括 σ -mod [?], e-modification [?] 或死区技术 [?]. 现在已经存在不需要 LIP 或确定回归矩阵的自适应控制技术 [?]. 将这些与这里将要讨论的神经网络控制器的复杂性进行比较是有趣的。

8.3.3 神经网络反馈跟踪控制器

神经网络将用于式 (??) 中的控制 $\boldsymbol{\tau}(t)$, 为未知机器人函数 $\mathbf{f}(\mathbf{x})$ 提供估计 $\hat{\mathbf{f}}$ 。神经网络逼近性质保证总是存在神经网络能在给定精度 ε_N 内完成这一点。对于 2 层 NLIP 神经网络, 逼近为

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\sigma(\mathbf{V}^T\mathbf{x}) + \varepsilon \quad (8.33)$$

而对于 1 层 LIP 神经网络, 它是

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\phi(\mathbf{x}) + \varepsilon \quad (8.34)$$

其中 $\phi(\cdot)$ 选择为基。

神经网络控制器的结构出现在图 8.3.1 中，其中 $\mathbf{e} \equiv [\mathbf{e}^T, \dot{\mathbf{e}}^T]^T$ 。为 $\mathbf{f}(\mathbf{x})$ 提供估计的神经网络出现在内部控制环路中，外部有 PD 项 $K_v \mathbf{r}$ 提供的跟踪环路。在控制理论术语中，内部环路是反馈线性化控制器 [?]，而在计算机学术语中，它是动作生成环路 [?, ?, ?]。这种多环智能控制结构自然地机器人控制概念中导出，不是临时性的。因此，它不受关于合适神经网络控制拓扑的哲学讨论的影响，包括关于前馈与反馈、直接 versus 间接等的常见讨论。值得注意的是，图中的静态前馈神经网络通过在周围闭合反馈环路变成了动态神经网络（参见 [?]

神经网络相比 LIP 自适应控制的优势。 每个机器人系统都有其自己的回归矩阵，因此必须为每个系统确定不同的 $\phi(\mathbf{x})$ 。这个回归矩阵通常很复杂，确定它可能很耗时。人们注意到，回归矩阵有效地为特定给定系统提供了函数逼近的基。另一方面，非线性网络的通用逼近性质表明，神经网络为所有足够光滑的系统提供了基。因此，神经网络可以用于逼近所有刚性机器人系统的光滑 $\mathbf{f}(\mathbf{x})$ ，有效地允许为一类系统设计单个可凋控制器。无需确定回归矩阵，因为相同的神经网络激活函数足以应对整个对象类。

即使是 LIP 的 1 层神经网络也有这个优势。注意，1 层神经网络在系统参数方面不是线性的，而是在神经网络权重方面是线性的。即使是 LIP 神经网络也具有通用逼近性质，因此不需要回归矩阵。

8.3.4 初始跟踪误差和初始神经网络权重

现在需要确定如何调整神经网络权重以产生保证的闭环稳定性。本节考虑神经网络控制器设计的几种情况。所有情况都需要以下构造。

由于神经网络逼近性质在紧集上成立，必须如下定义允许的初始条件集。让神经网络逼近性质对于式 (??) 中给出的函数 $\mathbf{f}(\mathbf{x})$ 在式 (??) 中具有给定精度 ε_N ，对于半径 $\mathbf{b}_x > \mathbf{q}_B$ 的球内所有 \mathbf{x} 成立。定义允许的初始跟踪误差集为

$$S_r = \{\mathbf{r}(0) : \|\mathbf{r}(0)\| \leq \frac{(\mathbf{b}_x - \mathbf{q}_B)}{c_0}\} \quad (8.35)$$

注意，神经网络的逼近精度决定了 S_r 的大小。对于更大的神经网络（即更多隐层单元）， ε_N 在更大半径 \mathbf{b}_x 下较小。因此，允许的初始条件集 S_r 更大。另一方面，更活跃的期望轨迹（例如包含更高频率分量）导致更大的加速度 $\ddot{\mathbf{q}}_d(t)$ ，产生更大的界 \mathbf{q}_B ，从而减小 S_r 。值得注意的是 S_r 依赖于 PD 设计比 Λ —— c_0 和 c_2 都依赖于 Λ 。

我们初始条件要求的一个关键特征是其独立于神经网络初始权重。这与文献中其他技术形成鲜明对比，在那些技术中，稳定性证明依赖于选择某些初始稳定神经网络权重，这是非常困难的。

8.4 线性参数神经网络的调参算法

假设现在使用 LIP FLNN 根据式 (??)逼近非线性机器人函数 (??), 其中 $\|\varepsilon\| \leq \varepsilon_M$ 在紧集上, 理想逼近权重 \mathbf{W} 是常数, $\phi(\mathbf{x})$ 选择为基。 $\mathbf{f}(\mathbf{x})$ 的估计由式 (??)给出。那么控制律 (??)变为

$$\boldsymbol{\tau} = \hat{\mathbf{W}}^T \phi(\mathbf{x}) + K_v \mathbf{r} - \mathbf{v}(t) \quad (8.36)$$

闭环滤波误差动力学 (??)为

$$M\dot{\mathbf{r}} = -(V_m + K_v)\mathbf{r} + \tilde{\mathbf{W}}^T \phi(\mathbf{x}) + \boldsymbol{\tau}_d + \varepsilon + \mathbf{v} \quad (8.37)$$

下一个定理导出了用 e-mod 项增强的神经网络控制器的调整律, 如 [?] 中所述。

现在必须假设理想权重 \mathbf{W} 是常数且有界, 使得

$$\|\mathbf{W}\|_F \leq W_B \quad (8.38)$$

其中界 W_B 已知。在 [?] 中, 展示了如何使用标准自适应鲁棒技术来避免假设界 W_B 已知。

定理 8.4-1: (神经网络权重调整算法)。

设期望轨迹 $\mathbf{q}_d(t)$ 被 \mathbf{q}_B 所界, 初始跟踪误差 $\mathbf{r}(0)$ 在 S_r 中。设神经网络重构误差界 ε_N 和扰动界 d_B 是常数。假设理想神经网络目标权重被 W_B 所界, 如式 (??)。设机器人手臂的控制输入由式 (??)给出, $\mathbf{v}(t) = 0$, 增益满足

$$K_{v \min} > \frac{(W_B + \frac{\varepsilon_N + d_B}{\|\phi\|})^2}{4} \quad (8.39)$$

设权重调整为

$$\dot{\hat{\mathbf{W}}} = \mathbf{F} \phi(\mathbf{x}) \mathbf{r}^T - \kappa \mathbf{F} \|\mathbf{r}\| \hat{\mathbf{W}} \quad (8.40)$$

其中 $\mathbf{F} = \mathbf{F}^T > 0$, $\kappa > 0$ 是小设计参数。那么滤波跟踪误差 $\mathbf{r}(t)$ 和神经网络权重估计 $\hat{\mathbf{W}}(t)$ 是一致最终有界 (UUB) 的, 实际界分别由式 (??)、(??)的右侧给出。而且, 通过增加跟踪增益 K_v , 跟踪误差可以变得任意小。

证明:

让神经网络逼近性质 (??)对于式 (??)中给出的函数 $\mathbf{f}(\mathbf{x})$ 在紧集 $\|\mathbf{x}\| \leq \mathbf{b}_x$ 上具有给定精度 ε_N , 其中 $\mathbf{b}_x > \mathbf{q}_B$ 。设 $\mathbf{r}(0) \in S_r$ 。那么在时刻 0 逼近性质成立。

定义 Lyapunov 函数候选

$$L = \frac{1}{2} \mathbf{r}^T M(\mathbf{q}) \mathbf{r} + \frac{1}{2} \text{tr}\{\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \tilde{\mathbf{W}}\} \quad (8.41)$$

求导得

$$\dot{L} = \mathbf{r}^T M \dot{\mathbf{r}} + \frac{1}{2} \mathbf{r}^T \dot{M} \mathbf{r} + \text{tr}\{\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \dot{\tilde{\mathbf{W}}}\} \quad (8.42)$$

代入式 (??) 得

$$\dot{L} = -\mathbf{r}^T(V_m + K_v)\mathbf{r} + \frac{1}{2}\mathbf{r}^T\dot{M}\mathbf{r} + \text{tr}\{\tilde{\mathbf{W}}^T(\mathbf{F}^{-1}\dot{\tilde{\mathbf{W}}} + \phi\mathbf{r}^T)\} + \mathbf{r}^T(\boldsymbol{\tau}_d + \boldsymbol{\varepsilon}) \quad (8.43)$$

然后, 使用调整规则 (??) 得

$$\dot{L} = -\mathbf{r}^TK_v\mathbf{r} + \kappa\|\mathbf{r}\|\text{tr}\{\tilde{\mathbf{W}}^T(\mathbf{W} - \tilde{\mathbf{W}})\} + \mathbf{r}^T(\boldsymbol{\tau}_d + \boldsymbol{\varepsilon}) \quad (8.44)$$

由于

$$\text{tr}\{\tilde{\mathbf{W}}^T(\mathbf{W} - \tilde{\mathbf{W}})\} = \langle \tilde{\mathbf{W}}, \mathbf{W} \rangle - \|\tilde{\mathbf{W}}\|_F^2 \quad (8.45)$$

$$\leq \|\tilde{\mathbf{W}}\|_F\|\mathbf{W}\|_F - \|\tilde{\mathbf{W}}\|_F^2 \quad (8.46)$$

得到

$$\dot{L} \leq -K_{v\min}\|\mathbf{r}\|^2 + \kappa\|\mathbf{r}\|\|\tilde{\mathbf{W}}\|_F(W_B - \|\tilde{\mathbf{W}}\|_F) + \|\mathbf{r}\|(d_B + \varepsilon_M) \quad (8.47)$$

这在花括号中的项为正时是负的。完成平方得

$$\dot{L} \leq -\|\mathbf{r}\| \left[K_{v\min}\|\mathbf{r}\| - \kappa W_B\|\tilde{\mathbf{W}}\|_F + \kappa\|\tilde{\mathbf{W}}\|_F^2 - (d_B + \varepsilon_M) \right] \quad (8.48)$$

当以下条件时保证为正

$$\|\mathbf{r}\| \geq \frac{\kappa W_B^2/4 + (d_B + \varepsilon_M)}{K_{v\min}} \equiv b_r \quad (8.49)$$

或

$$\|\tilde{\mathbf{W}}\|_F \geq W_B/2 + \sqrt{\kappa W_B^2/4 + (d_B + \varepsilon_M)/\kappa} \equiv b_W \quad (8.50)$$

因此, \dot{L} 在紧集外是负的。根据增益选择式 (??) 确保由 $\|\mathbf{r}\| \leq b_r$ 定义的紧集包含在 S_r 中, 因此逼近性质在整个过程中成立。这证明了 $\|\mathbf{r}\|$ 和 $\|\mathbf{W}\|_F$ 的 UUB。

这个证明类似于 [?], 但修改为反映神经网络只在紧集上逼近的事实。在这个结果中, 现在需要理想权重 \mathbf{W} 的界 W_B , 它出现在 PD 增益条件 (??) 中。现在发现了跟踪误差 $\mathbf{r}(t)$ 和权重 $\mathbf{W}(t)$ 的界。注意, 使用修改的权重调整算法建立 \mathbf{W} 的界不需要持续激励 (PE) 条件。

神经网络权重调整算法 (??) 由反向传播时间项加上一个新的第二项组成。 κ 项 (称为 e-modification) 的重要性在于它在 $\|\tilde{\mathbf{W}}\|_F$ 中添加了二次项, 使得可以证明 \dot{L} 在 $(\|\mathbf{r}\|, \|\tilde{\mathbf{W}}\|_F)$ 平面上的紧集外为负 [?]. e-mod 项使调整律对未建模动力学具有鲁棒性, 因此不需要 PE 条件。在 [?] 中, 使用投影算法来保持神经网络权重有界。在 [?, ?] 中, 采用了死区技术。

权重初始化和在线调整。 本文导出的神经网络控制方案中没有预先的离线学习阶段。权重简单地初始化为零, 因为那时图 8.3.1 显示控制器只是一个 PD 控制器。机器人学文献中的标准结果 [?] 表明, 如果 K_v 足够大, PD 控制器给出有界误差。因此, 闭

环系统保持稳定，直到神经网络开始学习。权重在系统跟踪期望轨迹时实时在线调整。随着神经网络学习 $f(\mathbf{x})$ ，跟踪性能提高。这比其他神经网络控制技术有重大改进，在那些技术中必须找到某些初始稳定权重，对于复杂非线性系统通常是不可能的。

跟踪误差和神经网络权重估计误差的界。式 (??) 的右侧可以视为跟踪误差的实际界，即 $\mathbf{r}(t)$ 永远不会偏离它太远。重要的是从这个方程注意到，跟踪误差随着神经网络重构误差 ε_N 和机器人扰动 d_B 而增加，但通过选择大增益 K_v ，可以获得任意小的跟踪误差。尽管 Barron 对 ε 有下界限制，也是如此。

注意，神经网络权重 $\hat{\mathbf{W}}$ 不能保证接近理想未知权重 \mathbf{W} ，后者给出 $f(\mathbf{x})$ 的良好逼近。然而，正如证明所保证的，只要 $\hat{\mathbf{W}} - \mathbf{W}$ 有界，这就无关紧要。这保证了有界控制输入，从而可以实现跟踪目标。

8.5 非线性参数神经网络的调参算法

现在假设使用 2 层神经网络根据式 (??) 逼近机器人函数，其中 $\|\varepsilon\| \leq \varepsilon_M$ 在紧集上，理想逼近权重 \mathbf{W} 、 \mathbf{V} 是常数。那么控制律 (??) 变为

$$\boldsymbol{\tau} = \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T \mathbf{x}) + K_v \mathbf{r} - \mathbf{v}(t) \quad (8.51)$$

提出的神经网络控制结构如图 8.3.1 所示。

NLIP 神经网络控制器比 1 层 FLNN 强大得多，因为不需要选择激活函数基。实际上，权重 \mathbf{V} 在线调整以自动提供合适的基。控制在第一层权重 \mathbf{V} 方面是非线性的，这在导出合适的权重调整律时带来复杂性。辅助信号 $\mathbf{v}(t)$ 是随后详细介绍的函数，它提供面对 NLIP 问题产生的泰勒级数中高阶项的鲁棒性。

使用结果 (??) 正确处理 NLIP 问题，闭环误差动力学 (??) 可以写成

$$M\dot{\mathbf{r}} = -(V_m + K_v)\mathbf{r} + \tilde{\mathbf{W}}^T \hat{\sigma}' \hat{\mathbf{V}}^T \mathbf{x} + \tilde{\mathbf{W}}^T \hat{\sigma} + \mathbf{w} + \boldsymbol{\tau}_d + \mathbf{v} \quad (8.52)$$

其中扰动项为

$$\mathbf{w} = \tilde{\mathbf{W}}^T \hat{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} + \mathbf{W}^T O(\tilde{\mathbf{V}}^T \mathbf{x})^2 \quad (8.53)$$

定义

$$\mathbf{Z} \equiv \begin{bmatrix} \mathbf{W} & 0 \\ 0 & \mathbf{V} \end{bmatrix}, \quad \hat{\mathbf{Z}} \equiv \begin{bmatrix} \hat{\mathbf{W}} & 0 \\ 0 & \hat{\mathbf{V}} \end{bmatrix}, \quad \tilde{\mathbf{Z}} \equiv \mathbf{Z} - \hat{\mathbf{Z}} \quad (8.54)$$

和 \mathbf{Z} 、 $\hat{\mathbf{Z}}$ 等价。不难证明

$$\|\mathbf{w}\| \leq C_0 + C_1 \|\tilde{\mathbf{Z}}\|_F + C_2 \|\tilde{\mathbf{Z}}\|_F \|\mathbf{r}\| \quad (8.55)$$

其中 C_i 是已知正常数。

假设理想权重有界，使得

$$\|\mathbf{Z}\|_F \leq Z_B \quad (8.56)$$

其中 Z_B 已知。(可以使用标准自适应鲁棒控制技术来放宽 Z_B 已知的假设 [?].)

下一个定理给出了本章最强大的神经网络控制器。

定理 8.5-1: (多层神经网络控制器的增强反向传播权重调整)。

设期望轨迹 $\mathbf{q}_d(t)$ 被 \mathbf{q}_B 所界, 初始跟踪误差 $\mathbf{r}(0)$ 在 S_r 中。设理想目标神经网络权重被已知 Z_B 所界。取机器人动力学 (??) 的控制输入为式 (??), PD 增益满足

$$K_{v \min} > \frac{(C_0 + C_1 Z_B + C_2 Z_B^2/4)(C_3 + Z_B)}{4} \quad (8.57)$$

其中 C_3 在证明中定义。设鲁棒化项为

$$\mathbf{v}(t) = -K_z(\|\hat{\mathbf{Z}}\|_F + Z_B)\mathbf{r} \quad (8.58)$$

增益为

$$K_z > C_2 \quad (8.59)$$

设神经网络权重调整为

$$\dot{\hat{\mathbf{W}}} = \mathbf{F}\hat{\sigma}'\hat{\mathbf{V}}^T\mathbf{x}\mathbf{r}^T - \mathbf{F}\hat{\sigma}'\tilde{\mathbf{V}}^T\mathbf{x}\mathbf{r}^T - \kappa\|\mathbf{r}\|\mathbf{F}\hat{\mathbf{W}} \quad (8.60)$$

$$\dot{\hat{\mathbf{V}}} = \mathbf{G}\mathbf{x}(\hat{\sigma}'^T\hat{\mathbf{W}}\mathbf{r})^T - \kappa\|\mathbf{r}\|\mathbf{G}\hat{\mathbf{V}} \quad (8.61)$$

其中任意常数矩阵 $\mathbf{F} = \mathbf{F}^T > 0$, $\mathbf{G} = \mathbf{G}^T > 0$, $\kappa > 0$ 是小标量设计参数。那么滤波跟踪误差 $\mathbf{r}(t)$ 和神经网络权重估计 $\hat{\mathbf{V}}$ 、 $\hat{\mathbf{W}}$ 是 UUB 的, 界分别由式 (??) 和 (??) 具体给出。而且, 通过增加式 (??) 中的增益 K_v , 跟踪误差可以保持任意小。

证明:

让神经网络逼近性质 (??) 对于式 (??) 中给出的函数 $\mathbf{f}(\mathbf{x})$ 在紧集 $\|\mathbf{x}\| \leq \mathbf{b}_x$ 上具有给定精度 ε_N , 其中 $\mathbf{b}_x > \mathbf{q}_B$ 。设 $\mathbf{r}(0) \in S_r$ 。那么在时刻 0 逼近性质成立。

定义 Lyapunov 函数候选

$$L = \frac{1}{2}\mathbf{r}^T M(\mathbf{q})\mathbf{r} + \frac{1}{2}\text{tr}\{\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \tilde{\mathbf{W}}\} + \frac{1}{2}\text{tr}\{\tilde{\mathbf{V}}^T \mathbf{G}^{-1} \tilde{\mathbf{V}}\} \quad (8.62)$$

求导得

$$\dot{L} = \mathbf{r}^T M\dot{\mathbf{r}} + \frac{1}{2}\mathbf{r}^T \dot{M}\mathbf{r} + \text{tr}\{\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \dot{\tilde{\mathbf{W}}}\} + \text{tr}\{\tilde{\mathbf{V}}^T \mathbf{G}^{-1} \dot{\tilde{\mathbf{V}}}\} \quad (8.63)$$

从误差系统 (??) 代入得

$$\begin{aligned} \dot{L} = & -\mathbf{r}^T K_v \mathbf{r} + \frac{1}{2}\mathbf{r}^T (\dot{M} - 2V_m)\mathbf{r} - \mathbf{r}^T \mathbf{v} + \mathbf{r}^T (\mathbf{w} + \boldsymbol{\tau}_d) \\ & + \text{tr}\{\tilde{\mathbf{W}}^T (\mathbf{F}^{-1} \dot{\tilde{\mathbf{W}}} + \hat{\sigma}'\mathbf{r}^T - \hat{\sigma}'\hat{\mathbf{V}}^T\mathbf{x}\mathbf{r}^T)\} \\ & + \text{tr}\{\tilde{\mathbf{V}}^T (\mathbf{G}^{-1} \dot{\tilde{\mathbf{V}}} + \mathbf{x}\mathbf{r}^T \hat{\mathbf{W}}^T \hat{\sigma}')\} \end{aligned} \quad (8.64)$$

调整规则给出

$$\dot{L} = -\mathbf{r}^T K_v \mathbf{r} + \kappa \|\mathbf{r}\| \text{tr}\{\tilde{\mathbf{Z}}^T (\mathbf{Z} - \tilde{\mathbf{Z}})\} + \mathbf{r}^T (\mathbf{w} + \boldsymbol{\tau}_d + \boldsymbol{\varepsilon}) \quad (8.65)$$

由于

$$\text{tr}\{\tilde{\mathbf{Z}}^T (\mathbf{Z} - \tilde{\mathbf{Z}})\} = \langle \tilde{\mathbf{Z}}, \mathbf{Z} \rangle - \|\tilde{\mathbf{Z}}\|_F^2 \quad (8.66)$$

$$\leq \|\tilde{\mathbf{Z}}\|_F \|\mathbf{Z}\|_F - \|\tilde{\mathbf{Z}}\|_F^2 \quad (8.67)$$

得到

$$\dot{L} \leq -K_{v\min} \|\mathbf{r}\|^2 - K_z \|\mathbf{r}\|^2 (\|\hat{\mathbf{Z}}\|_F + Z_B) + \kappa \|\mathbf{r}\| \|\tilde{\mathbf{Z}}\|_F (Z_B - \|\tilde{\mathbf{Z}}\|_F) + \|\mathbf{r}\| (C_0 + C_1 \|\tilde{\mathbf{Z}}\|_F + C_2 \|\tilde{\mathbf{Z}}\|_F \|\mathbf{r}\|) \quad (8.68)$$

其中 $K_{v\min}$ 是 K_v 的最小奇异值，最后一个不等式由于式 (??) 成立。

\dot{L} 在花括号中的项为正时是负的。定义 $C_3 = Z_B + C_1/\kappa$ 并完成平方得

$$\dot{L} \leq -\|\mathbf{r}\| \left[K_{v\min} \|\mathbf{r}\| - \kappa C_3^2/4 - C_0 - \frac{(C_3 + Z_B/2)^2}{K_z - C_2} \right] \quad (8.69)$$

当以下条件时保证为正

$$\|\mathbf{r}\| \geq \frac{\kappa C_3^2/4 + C_0}{K_{v\min}} + \frac{(C_3 + Z_B/2)^2}{K_{v\min}(K_z - C_2)} \equiv b_r \quad (8.70)$$

或

$$\|\tilde{\mathbf{Z}}\|_F \geq C_3/2 + \sqrt{C_3^2/4 + C_0/\kappa + (C_3 + Z_B/2)^2/[\kappa(K_z - C_2)]} \equiv b_Z \quad (8.71)$$

因此， \dot{L} 在紧集外是负的。根据 LaSalle 扩展 [?], 这证明了在该集合内控制保持有效时 $\|\mathbf{r}\|$ 和 $\|\mathbf{Z}\|_F$ 的 UUB。然而，PD 增益条件 (??) 表明由 $\|\mathbf{r}\| \leq b_r$ 定义的紧集包含在 S_r 中，因此逼近性质在整个过程中成立。

调整算法 (??)、(??) 中的第一项不过是跟踪误差通过时间的反向传播。最后一项是 Narendra e-mod，式 (??) 中的第二项是由于 NLIP 而产生的新型二阶项。鲁棒化项 $\mathbf{v}(t)$ 也是由于 NLIP 问题而需要的。

定理 8.4.1 之后的评论在这里也是适用的。具体来说，没有预先离线调整阶段，神经网络权重调整在实时控制动作的同时在线发生。权重初始化在证明中不是问题，无需提供初始稳定权重；权重简单地初始化使得神经网络输出等于零，因为那时 PD 环路保持系统稳定，直到神经网络开始学习。然而，实际实验 [?] 表明，适当初始化权重很重要。一个好的选择是随机选择 $\hat{\mathbf{V}}(0)$ ， $\hat{\mathbf{W}}(0)$ 等于零。

在实践中不需要计算常数 c_0 、 c_1 、 c_2 、 C_0 、 C_1 、 C_2 、 Z_B 或确定 S_r 。简单地选择大的神经网络规模 L 和 PD 增益 K_v ，然后执行仿真。然后用不同的 L 和 K_v 重复仿真。基于两次仿真之间行为的差异，可以修改 L 和 K_v 。

直接反向传播权重调整。反向传播算法（通常以离散时间形式）在文献中被无数次提出用于闭环控制。可以证明，如果神经网络逼近误差 ε 为零，扰动 $\tau_d(t)$ 为零，且泰勒级数 (??) 中没有高阶项，那么直接反向传播算法

$$\dot{\hat{W}} = F\hat{\sigma}'\hat{V}^T \mathbf{x}\mathbf{r}^T, \quad \dot{\hat{V}} = G\mathbf{x}(\hat{\sigma}'^T \hat{W}\mathbf{r})^T \quad (8.72)$$

可以用于闭环控制代替式 (??)、(??)。需要使用 PE 来证明使用直接反向传播时神经网络权重是有界的。提到的条件非常严格，在实践中不会发生；它们本质上要求机器人手臂是线性的（例如单连杆）。而且，在神经网络中不容易保证 PE 条件。

8.5.1 神经网络控制器的无源性性质

本文中使用的神经网络是静态前馈网络，但由于它们出现在闭环反馈中并使用微分方程调整，它们是动态系统。因此，可以讨论这些神经网络的无源性。一般来说，不能保证神经网络是被动的。然而，本文中的神经网络控制器具有一些重要的无源性性质，从而产生鲁棒的闭环性能。

系统 $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, $\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u})$ 如果验证所谓的功率形式等式 [?]

$$\dot{L} = \mathbf{u}^T \mathbf{y} - g(\mathbf{x}, \mathbf{u}) \quad (8.73)$$

对于某个下有界 $L(t)$ 和某个 $g(t) \geq 0$ ，则称为被动的。即

$$\int_0^T \mathbf{u}^T \mathbf{y} dt \geq \int_0^T g(t) dt - \gamma_0^2 \quad (8.74)$$

对于所有 $T \geq 0$ 和某个 $\gamma_0^2 \geq 0$ 。如果系统是被动的，且此外

$$\int_0^T \mathbf{u}^T \mathbf{y} dt \geq \int_0^T g(t) dt - \gamma_0^2 \quad (8.75)$$

其中 $g(t) > 0$ ，则系统称为耗散的。

如果 $g(t)$ 是具有有界系数的 $\|\mathbf{x}\|$ 的二次函数，则发生一种特殊的耗散性。我们称之为状态严格无源性 (SSP)；那么，内部状态的范数以输送到系统的功率为界。令人惊讶的是，SSP 概念在文献中没有得到广泛使用 [?, ?]，尽管参见 [?] 其中定义了输入和输出严格无源性。SSP 结果在研究神经网络控制器的无源性性质中是关键性的，并允许在没有持续激励假设的情况下得出某些内部有界性质。

机器人跟踪误差动力学的无源性

本文中的误差动力学（例如 (??)）具有形式

$$M\dot{\mathbf{r}} = -(V_m + K_v)\mathbf{r} + \boldsymbol{\xi}_0 \quad (8.76)$$

其中 $\mathbf{r}(t)$ 是滤波跟踪误差， $\boldsymbol{\xi}_0(t)$ 是适当定义的。该系统满足以下强无源性性质。

定理 8.5-2: (机器人误差动力学的 SSP)。

动力学 (??) 从 $\xi_0(t)$ 到 $r(t)$ 是状态严格被动系统。

证明:

取非负函数

$$L = \frac{1}{2} r^T M(q) r \quad (8.77)$$

所以, 使用 (??), 得到

$$\dot{L} = -r^T (V_m + K_v) r + r^T \xi_0 \quad (8.78)$$

因此, 斜对称性质产生功率形式

$$\dot{L} = r^T \xi_0 - r^T K_v r \quad (8.79)$$

这是输送到系统的功率减去 $\|r\|^2$ 中的二次项, 验证了状态严格无源性。

2 层神经网络控制器的无源性性质

闭环系统如图 8.3.1 所示, 其中神经网络出现在内部反馈线性化环路中。2 层神经网络控制器的误差动力学由式 (??) 给出。闭环误差系统如图 8.5.1 所示, 信号 ξ_1 定义为

$$\xi_1 = \tilde{W}^T \hat{\sigma}' \hat{V}^T x + \tilde{W}^T \hat{\sigma} + w + \tau_d \quad (8.80)$$

注意神经网络在误差动力学中的作用, 其中它被分解为两个有效块, 出现在典型的反馈配置中。

我们现在揭示直接反向传播调整 (??) 产生的无源性性质。为了证明这个算法, 使用与 (??) 不同的误差动力学形式, 使得在图 8.5.1 中 $\xi_1 = \tilde{W}^T \sigma(V^T x) + w + \tau_d$ 。

定理 8.5-3: (反向传播神经网络调整算法的无源性)。

简单反向传播权重调整算法 (??) 使从 $-\hat{W}^T \sigma(V^T x)$ 到 $\hat{\sigma}' \hat{V}^T x$ 的映射和从 $r(t)$ 到 $\hat{W}^T \hat{\sigma}' \hat{V}^T x$ 的映射都是被动映射。

证明:

相对于 W 、 V 的动力学为

$$\dot{\tilde{W}} = -F \hat{\sigma}' \hat{V}^T x r^T, \quad \dot{\tilde{V}} = -G x (\hat{\sigma}' \hat{V}^T x) r^T \quad (8.81)$$

1. 选择非负函数

$$L = \frac{1}{2} \text{tr}\{\tilde{W}^T F^{-1} \tilde{W}\} \quad (8.82)$$

并沿轨迹 (1) 评估 \dot{L} 得

$$\dot{L} = \text{tr}\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\} = -\text{tr}\{\tilde{W}^T \hat{\sigma}' \hat{V}^T x r^T\} = \langle -\hat{W}^T \sigma(V^T x), \hat{\sigma}' \hat{V}^T x \rangle \quad (8.83)$$

这是功率形式 (??)。

2. 选择非负函数

$$L = \frac{1}{2} \text{tr}\{\tilde{\mathbf{V}}^T \mathbf{G}^{-1} \tilde{\mathbf{V}}\} \quad (8.84)$$

并沿轨迹 (2) 评估得

$$\dot{L} = \text{tr}\{\dot{\tilde{\mathbf{V}}}^T \mathbf{G}^{-1} \tilde{\mathbf{V}}\} = -\text{tr}\{\tilde{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T \hat{\mathbf{W}}^T \dot{\sigma}'\} = \langle \mathbf{r}, \hat{\mathbf{W}}^T \dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x} \rangle \quad (8.85)$$

这是功率形式。

因此, 图 8.5.1 中的机器人误差系统是状态严格被动的 (SSP), 权重误差块是被动的; 这保证了闭环系统的无源性。使用无源性定理 [?], 现在可以得出, 只要外部输入有界, 每个块的输入/输出信号都是有界的。

不幸的是, 虽然是被动的, 但闭环系统不能保证是 SSP, 因此当扰动非零时, 这不产生权重块内部状态 (即 \mathbf{W} 、 \mathbf{V}) 的有界性, 除非这些块是可观测的, 即持续激励 (PE)。

下一个结果显示为什么使用定理 8.5.1 中给出的修改权重更新算法不需要 PE 条件。

定理 8.5-4: (增强反向传播神经网络调整算法的 SSP)。

定理 8.5.1 中的修改权重调整算法使从 $-\hat{\mathbf{W}}^T \sigma(\mathbf{V}^T \mathbf{x})$ 到 $\dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x}$ 的映射和从 $\mathbf{r}(t)$ 到 $\hat{\mathbf{W}}^T \dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x}$ 的映射都是状态严格被动 (SSP) 映射。

证明:

使用定理 8.5.1 中的调整算法, 相对于 \mathbf{W} 、 \mathbf{V} 的动力学由下式给出

$$\dot{\tilde{\mathbf{W}}} = -\mathbf{F} \dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T + \mathbf{F} \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T + \kappa \|\mathbf{r}\| \mathbf{F} \tilde{\mathbf{W}} \quad (8.86)$$

$$\dot{\tilde{\mathbf{V}}} = -\mathbf{G} \mathbf{x} \mathbf{r}^T \hat{\mathbf{W}}^T \dot{\sigma}' + \kappa \|\mathbf{r}\| \mathbf{G} \tilde{\mathbf{V}} \quad (8.87)$$

1. 选择非负函数

$$L = \frac{1}{2} \text{tr}\{\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \tilde{\mathbf{W}}\} \quad (8.88)$$

并评估 \dot{L} 得

$$\dot{L} = \text{tr}\{\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \dot{\tilde{\mathbf{W}}}\} \quad (8.89)$$

$$= \text{tr}\{\tilde{\mathbf{W}}^T (-\dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T + \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T + \kappa \|\mathbf{r}\| \hat{\mathbf{W}})\} \quad (8.90)$$

$$= \langle -\hat{\mathbf{W}}^T \sigma(\mathbf{V}^T \mathbf{x}), \dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x} \rangle + \text{tr}\{\tilde{\mathbf{W}}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T\} + \kappa \|\mathbf{r}\| \text{tr}\{\tilde{\mathbf{W}}^T (\mathbf{W} - \tilde{\mathbf{W}})\} \quad (8.91)$$

由于

$$\text{tr}\{\tilde{\mathbf{W}}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \mathbf{r}^T\} = \langle \mathbf{r}, \tilde{\mathbf{W}}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \rangle = \langle \mathbf{r}, \hat{\mathbf{W}}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \rangle - \langle \mathbf{r}, \mathbf{W}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \rangle \quad (8.92)$$

$$= \langle \mathbf{r}, \hat{\mathbf{W}}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \rangle - \langle \mathbf{r}, \hat{\mathbf{W}}^T \dot{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \rangle = 0 \quad (8.93)$$

得到

$$\dot{L} = \langle -\hat{\mathbf{W}}^T \sigma(\mathbf{V}^T \mathbf{x}), \dot{\sigma}' \hat{\mathbf{V}}^T \mathbf{x} \rangle + \kappa \|\mathbf{r}\| \text{tr}\{\tilde{\mathbf{W}}^T \mathbf{W}\} - \kappa \|\mathbf{r}\| \|\tilde{\mathbf{W}}\|_F^2 \quad (8.94)$$

这是功率形式，最后一项在 $\|\tilde{\mathbf{W}}\|_F$ 中是二次的。

2. 选择非负函数

$$L = \frac{1}{2} \text{tr}\{\tilde{\mathbf{V}}^T \mathbf{G}^{-1} \tilde{\mathbf{V}}\} \quad (8.95)$$

并评估 \dot{L} 得

$$\dot{L} = \text{tr}\{\tilde{\mathbf{V}}^T \mathbf{G}^{-1} \dot{\tilde{\mathbf{V}}}\} = \langle \mathbf{r}, \hat{\mathbf{W}}^T \hat{\sigma}' \tilde{\mathbf{V}}^T \mathbf{x} \rangle + \kappa \|\mathbf{r}\| \text{tr}\{\tilde{\mathbf{V}}^T \mathbf{V}\} - \kappa \|\mathbf{r}\| \|\tilde{\mathbf{V}}\|_F^2 \quad (8.96)$$

这是功率形式，最后一项在 $\|\tilde{\mathbf{V}}\|_F$ 中是二次的。

机器人动力学和权重调整块都是 SSP，这保证了闭环系统的 SSP，因此内部状态的范数以输送到每个块的功率为界。那么，输入/输出信号的有界性保证了在没有可观测性或 PE 要求的情况下状态有界性。

被动神经网络和鲁棒神经网络的定义。如果调整在误差形式中保证权重调整子系统的无源性，则我们将动态调整的神经网络定义为被动的。那么，需要额外的 PE 条件来保证权重的有界性。这个 PE 条件通常以神经网络隐层输出为条件。如果调整在误差形式中保证权重调整子系统的 SSP，则我们将动态调整的神经网络定义为鲁棒的。那么，不需要额外的 PE 条件来保证权重的有界性。注意 (1) 此外还需要开环对象误差系统的 SSP 以保证跟踪稳定性，(2) 神经网络无源性性质依赖于所使用的权重调整算法。

1 层神经网络控制器的无源性性质

类似地，FLNN 控制器调整算法使系统被动，因此需要额外的 PE 条件来验证神经网络权重的内部稳定性。另一方面，定理 8.4.1 中的增强调整算法产生 SSP，因此不需要 PE。

8.6 小结

为一般工业拉格朗日运动系统（以刚性机器人手臂为特征）给出了神经网络控制器设计算法。设计程序基于严格的非线性推导和稳定性证明，产生具有神经网络在某些环路中的多环智能控制结构。给出了不需要复杂初始化程序或任何离线学习阶段、实时在线工作、提供保证跟踪和有界神经网络权重和控制信号的神经网络权重调整算法。这里给出的神经网络控制器是无模型控制器，因为它们对规定类别的任何系统都有效，无需大量建模和初步分析来寻找“回归矩阵”。与标准机器人自适应控制器不同，它们不需要参数线性化（另见 [?]，它不需要 LIP）。适当的设计允许神经网络控制器避免诸如持续激励和确定性等价等要求。本文定义和研究了神经网络无源性和鲁棒性性质。

参考文献

- [1] K.J.Åström and B.Wittenmark, *Adaptive Control*, Addison Wesley, Reading, MA, 1989.
- [2] Barron, A.R., “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Trans. Info. Theory*, vol. 39, no. 3, pp. 930–945, May 1993.
- [3] F.-C.Chen and H.K.Khalil, “Adaptive control of nonlinear systems using neural networks,” *Int. J. Control*, vol. 55, no. 6, pp. 1299–1317, 1992.
- [4] F.-C.Chen and C.-C.Liu, “Adaptively controlling nonlinear continuous-time systems using multilayer neural networks,” *IEEE Trans. Automat. Control*, vol. 39, no. 6, pp. 1306–1310, June 1994.
- [5] R.Colbaugh, H.Seraji, and K.Glass, “A new class of adaptive controllers for robot trajectory tracking,” *J. Robotic Systems*, vol. 11, no. 8, pp. 761–772, 1994.
- [6] Craig, J.J., *Adaptive Control of Mechanical Manipulators*, Reading, MA: Addison-Wesley, 1988.
- [7] Dawson, D.M., Z.Qu, F.L.Lewis, and J.F.Dorsey, “Robust control for the tracking of robot motion,” *Int. J. Control*, vol. 52, no. 3, pp. 581–595, 1990.
- [8] Goodwin, C.G., and K.S.Sin, *Adaptive Filtering, Prediction, and Control*, Prentice-Hall, New Jersey, 1984.
- [9] D.Gorinevsky, “On the persistency of excitation in radial basis function network identification of nonlinear systems,” *IEEE Trans. Neural Networks*, vol. 6, no. 5, pp. 1237–1244, Sept. 1995.
- [10] L.B.Gutierrez and F.L.Lewis, “Implementation of a neural net tracking controller for a single flexible link: comparison with PD and PID controllers,” *IEEE Trans. Industrial Electronics*, to appear, April 1998.

- [11] S.Haykin, *Neural Networks*, IEEE Press, New York, 1994.
- [12] K.Hornik, M.Stinchcombe, and H.White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
- [13] P.A.Ioannou and P.V.Kokotovic, "Instability analysis and improvement of robustness of adaptive control," *Automatica*, vol. 20, no. 5, pp. 583–594, 1984.
- [14] Y.H.Kim and F.L.Lewis, *High-Level Feedback Control with Neural Networks*, World Scientific, Singapore, 1998.
- [15] Kreisselmeier, G., and B.Anderson, "Robust model reference adaptive control," *IEEE Trans. Automat. Control*, vol. AC-31, no. 2, pp. 127–133, Feb. 1986.
- [16] Y.D.Landau, *Adaptive Control*, Marcel Dekker, Basel, 1979.
- [17] F.L.Lewis, "Nonlinear Network Structures for Feedback Control", *Asian Journal of Control*, vol. 1, no. 4, pp.205–228, December 1999.
- [18] F.L.Lewis, C.T.Abdallah, and D.M.Dawson, *Control of Robot Manipulators*, New York: Macmillan, 1993.
- [19] F.L.Lewis, S.Jagannathan, and A.Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis, London, 1999.
- [20] F.L.Lewis and Y.H.Kim, "Neural Networks for Feedback Control," *Encyclopedia of Electrical and Electronics Engineering*, ed. J.G.Webster, New York: Wiley, 1998. To appear.
- [21] Lewis, F.L., K.Liu, and A.Yesildirek, "Neural net robot controller with guaranteed tracking performance," *Proc. Int. Symp. Intelligent Control*, pp. 225–231, Chicago., Aug. 1993.
- [22] Lewis, F.L., K.Liu, and A.Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.
- [23] F.L.Lewis, K.Liu, and S.Commuri, "Neural networks and fuzzy logic systems for robot control," in *Fuzzy Logic and Neural Network Applications*, ed. F.Wang, World Scientific Pub., to appear, 1997.
- [24] F.L.Lewis, A.Yesildirek, and K.Liu, "Multilayer neural net robot controller: structure and stability proofs," *IEEE Trans. Neural Networks*, vol. 7, no. 2, pp. 1–12, Mar. 1996.

- [25] W.T.Miller, R.S.Sutton, P.J.Werbos, ed., *Neural Networks for Control*, Cambridge: MIT Press, 1991.
- [26] K.S.Narendra, "Adaptive Control Using Neural Networks," *Neural Networks for Control*, pp 115–142. ed. W.T.Miller, R.S.Sutton, P.J.Werbos, Cambridge: MIT Press, 1991.
- [27] K.S.Narendra and A.M.Annaswamy, "A new adaptive law for robust adaptive control without persistent excitation," *IEEE Trans. Automat. Control*, vol. 32, pp. 134–145, Feb., 1987.
- [28] Narendra, K.S., and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 252–262, Mar. 1991.
- [29] M.M.Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Automat. Control*, vol. 41, no. 3, pp. 447–451, Mar. 1996.
- [30] M.M.Polycarpou and P.A.Ioannou, "Identification and control using neural network models: design and stability analysis," Tech. Report 91–09–01, Dept. Elect. Eng. Sys., Univ. S. Cal., Sept. 1991.
- [31] G.A.Rovithakis and M.A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 400–412, Mar. 1994.
- [32] N.Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 982–988, Nov. 1993.
- [33] R.M.Sanner and J.-J.E.Slotine, "Stable adaptive control and recursive identification using radial gaussian networks," *Proc. IEEE Conf. Decision and Control*, Brighton, 1991.
- [34] Seron, M.M., D.J.Hill, and A.L.Fradkov, "Adaptive passification of nonlinear systems," *Proc. IEEE Conf. Decision and Control*, pp. 190–195, Dec. 1994.
- [35] Slotine, J.-J.E., "Putting physics in control the example of robotics," *IEEE Control Systems Magazine*, pp. 12–17, Dec. 1988.
- [36] J.-J.Slotine and W.Li, *Applied Nonlinear Control*, Prentice Hall, New Jersey, 1991.
- [37] Spong, M.W., and M.Vidyasagar, *Robot Dynamics and Control*, New York: Wiley, 1989.

-
- [38] L.-X.Wang, *A course in fuzzy systems and control*, Prentice-Hall, New Jersey, 1997.
- [39] P.J.Werbos, "Back propagation: past and future," *Proc. 1988 Int. Conf. Neural Nets*, vol. 1, pp. I343-I353, 1989.
- [40] P.J.Werbos, "Neurocontrol and supervised learning: an overview and evaluation," in *Handbook of Intelligent Control*, ed. D.A. White and D.A.Sofge, New York: Van Nostrand Reinhold, 1992.
- [41] D.A.White and D.A.Sofge, ed. *Handbook of Intelligent Control*, New York: Van Nostrand Reinhold, 1992.

Chapter 9

力控制

本章研究位置/力控制器的基本原理。涵盖的主题包括刚度控制、混合位置/力控制、混合阻抗控制和降维状态位置/力控制。重点放在控制器的开发、稳定性和实现问题上。

9.1 引言

对于机器人机械臂执行的任务，例如移动载荷或喷涂物体，位置控制器能够提供足够的性能，因为这些类型的任务仅要求机器人跟随期望轨迹。然而，在磨削或装配任务中，机器人机械臂与环境接触；因此，在机器人机械臂和环境之间会产生相互作用力。因此，这些相互作用力以及末端执行器的位置都必须被控制。

为了说明同时使用力和位置控制的必要性，考虑控制机械臂在黑板上书写句子的问题。为了形成句子中的字母，我们当然必须控制末端执行器的位置，或者等价地，控制粉笔的位置。正如任何在黑板上书写过的人都知道的，按压黑板的力也必须被控制。也就是说，按压太轻会导致字母不易辨认，而按压太重会导致粉笔折断。这个例子清楚地说明了许多机器人应用将需要指定期望位置轨迹和期望力轨迹。在本章中，我们介绍一些通用的控制策略，不仅控制机器人末端执行器的位置，还控制末端执行器施加在环境上的力。需要注意的是，在本章中，我们假设控制器发出的期望速度和力轨迹与环境模型一致 [Lipkin and Duffy 1988]。如果不是这种情况，可以修改期望速度和力轨迹以使其与环境模型一致。有兴趣的读者可以参考 [Lipkin and Duffy 1988] 了解关于这种修改或期望轨迹的“运动静力学滤波”的信息。

9.2 刚度控制

由于工业过程中涉及的第一批机器人机械臂需要执行位置任务（例如，喷涂），机器人机械臂被制造得非常刚性。这种刚性设计允许机器人控制设计者通过使用简单的控制律获得合理的位置精度。正如人们所预期的，力控制应用（例如，磨削或打磨）用这

种“刚性”机器人极其难以完成。因此，如果机器人机械臂的“刚度”可以被控制，力控制应用将更容易实现。在本节中，针对简单的单自由度示例阐述了刚度控制的概念。然后修改机器人机械臂方程以考虑施加在环境上的力。使用这个新模型，刚度控制概念 [Salisbury and Craig 1980] 被推广到 n 连杆机器人机械臂。

9.2.1 单自由度机械臂的刚度控制

为了阐述刚度控制的概念，考虑图 9.2.1 所示系统的力控制问题。这里质量为 m 的机械臂被假设与环境接触，环境位于静态位置 x_e 。控制问题是确定输入力（即 τ ），使机械臂移动到期望的恒定位置（即 x_d ）。在这个系统中，我们还假设如果机械臂的位置（即 x ）大于 x_e ，施加在环境上的力（即 f ）由下式给出：

$$f = k_e(x - x_e), \quad (9.1)$$

其中 k_e 是一个正常数，用于表示环境刚度。也就是说，我们假设环境刚度可以被建模为弹簧常数为 k_e 的线性弹簧。基于这个假设，我们可以将单自由度系统可视化为图 9.2.2 所示的质量-弹簧图。

假设重力和摩擦可以忽略不计，图 9.2.2 所示系统的运动方程由下式给出：

$$m\ddot{x} = \tau - f \quad (9.2)$$

式 (9.2) 的系统框图由图 9.2.3 给出。注意，在图 9.2.3 中，我们使用变量 s 表示拉普拉斯变换变量。

式 (9.2) 的动力学形式启发了简单的 PD 控制律：

$$\tau = -k_v\dot{x} + k_p(x_d - x) \quad (9.3)$$

其中 k_v 和 k_p 是正标量控制增益。将式 (9.3) 代入式 (9.2) 后，我们得到闭环系统：

$$m\ddot{x} + k_v\dot{x} + k_px + k_e(x - x_e) = k_px_d \quad (9.4)$$

这可以用图 9.2.4 所示的框图表示。从图 9.2.4 我们知道闭环系统是稳定的，因为我们已经定义常数 m 、 k_v 、 k_p 和 k_e 为正数。也就是说，我们可以证明传递函数

$$\frac{s}{k_p}$$

的极点在开左半 s 平面。

为了研究式 (9.4) 给出的 PD 控制如何控制施加在环境上的力，我们检查系统在稳态条件下的情况。因为 x_d 和 x_e 是常数，可以从式 (9.4) 找到 x 的拉普拉斯变换为

$$x(s) = \frac{k_px_d + k_ex_e}{ms^2 + k_vs + k_p + k_e} \quad (9.5)$$

因此，稳态机械臂位置（即 x_{ss} ）可以很容易地证明为

$$x_{ss} = \frac{k_p x_d + k_e x_e}{k_p + k_e} \quad (9.6)$$

现在可以使用稳态机械臂位置来计算施加在环境上的稳态力（即 f ）。具体来说，将式 (9.6) 代入式 (9.5)，稳态力由下式给出：

$$f_{ss} = \frac{k_p k_e}{k_p + k_e} (x_d - x_e) \quad (9.7)$$

正如人们所预期的，环境弹簧常数通常被认为很大，因为在大多数机器人力控制应用中，机器人推动的是几乎刚性的表面。因此，如果我们假设 $k_e \gg k_p$ ，我们可以将式 (9.7) 中的稳态力近似为

$$f_{ss} \approx k_p (x_d - x_e) \quad (9.8)$$

从上面的讨论中，我们可以看到式 (9.8) 给出的位置控制策略确实会对环境施加一个力。具体来说，这个力是通过命令一个稍微进入接触面内部的期望轨迹而产生的。在试图消除位置误差时，位置控制器在表面上施加了一个稳态力。从式 (9.8) 给出的近似稳态力来看，位置增益（即 k_p ）可以被看作是代表机械臂的期望“刚度”。也就是说，机械臂可以被可视化为一个弹簧，弹簧常数为 k_p ，对环境施加力。因此，“刚度控制”这个术语经常与式 (9.8) 给出的 PD 控制相关联，因为机械臂的刚度可以通过调整 k_p 来设置。

9.2.2 雅可比矩阵和环境力

在将刚度控制器推广到 n 连杆机器人机械臂之前，我们必须定义一些关于机器人对环境施加的力的符号。正如 [Spong and Vidyasagar 1989] 中所解释的，力通常通过雅可比矩阵转换到关节空间。在本章中，我们根据为特定机器人应用定义的任务空间坐标系来定义雅可比矩阵。也就是说，对于某个应用，我们可能希望末端执行器沿一组特定方向施加力，而沿其他方向移动。这个概念由图 9.2.5 说明，图中描述了一个沿倾斜表面移动的机械臂。任务空间坐标系由方向 u 和 v 给出，因为我们希望沿 v 方向沿表面移动末端执行器，同时沿 u 方向对表面施加法向力。通常从这种推理中为大多数机器人应用定义适当的任务空间坐标系。

按照这个逻辑，让 x 为由下式定义的 $n \times 1$ 任务空间向量

$$x = h(q) \quad (9.9)$$

其中 $h(q)$ 从机械臂运动学和关节空间与任务空间之间的适当关系中找到。 x 的导数定义为

$$\dot{x} = J(q)\dot{q} \quad (9.10)$$

其中 $n \times n$ 任务空间雅可比矩阵 $J(q)$ [Spong and Vidyasagar 1989] 定义为

$$J(q) = \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} \frac{\partial h(q)}{\partial q} \quad (9.11)$$

其中单位矩阵 I 、零矩阵 0 和变换矩阵 T 的维度取决于所选的任务空间坐标系。变换矩阵 T 通常在将关节速度转换为与末端执行器方向相关的横滚、俯仰和偏航角的导数时使用。为简洁起见，我们假设本章讨论的机器人机械臂是非冗余的，并且始终处于非奇异构型；因此，雅可比矩阵是一个非奇异方阵。

使用任务空间坐标概念，我们现在检查如何修改机器人方程以用于力控制目的。如果机械臂的末端执行器与环境接触，末端执行器和环境之间将存在力相互作用。如果相互作用力在关节空间中测量，机械臂动力学方程可以写成

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau - \tau_e \quad (9.12)$$

其中 τ_e 是关节空间坐标中的 $n \times 1$ 向量，表示施加在环境上的力。式 (9.12) 给出的动力学方程是有意义的，因为如果机械臂不运动（即 $\dot{q} = \ddot{q} = 0$ ），则式 (9.12) 简化为

$$G(q) = \tau - \tau_e \quad (9.13)$$

也就是说，如果机器人机械臂处于静态操作，执行器力等于施加在环境上的力加上承受重力所需的力。注意，在式 (9.13) 中，我们假设可以忽略静摩擦。

关节空间表示施加在环境上的力不是机器人文献中的标准符号；相反，机器人机械臂方程通常由下式给出

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f = \tau \quad (9.14)$$

其中 f 是任务空间中的接触力和力矩的 $n \times 1$ 向量。

为了理解式 (9.14) 的来源，将式 (9.12) 和式 (9.14) 的右侧相等，得到

$$\tau_e = J^T(q)f \quad (9.15)$$

这个方程可以通过使用能量守恒论证很容易地证明为真。具体来说，通过能量守恒，我们知道

$$\dot{q}^T \tau_e = \dot{x}^T f \quad (9.16)$$

将式 (9.15) 代入式 (9.16) 得到

$$\dot{q}^T \tau_e = (J(q)\dot{q})^T f = \dot{q}^T J^T(q)f$$

由于上述关系必须对所有 q 都成立，我们可以看到式 (9.15) 显然代表一个真命题。为了阐明开发雅可比矩阵和任务空间公式的过程，现在讨论两个示例。

示例 9.2-1: 倾斜表面的任务空间公式

我们希望找到图 9.2.5 中所示笛卡尔机械臂系统（即两个关节都是棱柱形）的机械臂动力学，并将施加在表面上的力分解为法向力和切向力。首先，当机器人不受表面约束时，可以轻松确定动力学部分。在移除表面和相互作用力 f_1 和 f_2 后，机械臂动力学可以表示为

$$M\ddot{q} + F(q) = \tau \quad (9.17)$$

其中

$$M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$F(q)$ 是 2×1 向量 $\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$ ，它模拟了如第 2 章所讨论的摩擦。

为了考虑相互作用力，让 x 为由下式定义的 2×1 任务空间向量

$$x = \begin{bmatrix} u \\ v \end{bmatrix} \quad (9.18)$$

其中 u 和 v 定义了一个固定坐标系，使得 u 表示到表面的法向距离， v 表示沿表面的切向距离。如式 (??) 所示，任务空间坐标可以用关节空间坐标表示为

$$x = h(q) \quad (9.19)$$

其中 $h(q)$ 从问题的几何中找到为

$$h(q) = \frac{1}{\sqrt{2}} \begin{bmatrix} q_1 - q_2 \\ q_1 + q_2 \end{bmatrix} \quad (9.20)$$

任务空间雅可比矩阵从式 (??) 中找到，利用 T 是单位矩阵的事实，因为我们不需要关心任何末端执行器方向角。也就是说， $J(q)$ 由下式给出

$$J(q) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (9.21)$$

根据式 (??)，机器人机械臂方程由下式给出

$$M\ddot{q} + G + F(q) + J^T f = \tau \quad (9.22)$$

其中

$$G = \begin{bmatrix} m_1 g \\ 0 \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

重要的是要认识到法向力（即 f_1 ）和切向力（即 f_2 ）是按 (2) 给出的任务空间坐标系的方向绘制的（见图 9.2.5）。

示例 9.2-2: 椭圆表面的任务空间公式

我们希望找到图 9.2.6 中所示笛卡尔机械臂系统的机械臂动力学, 并将施加在表面上的力分解为法向力和切向力。动力学部分与示例 9.2.1 相同; 然而, 由于环境表面的变化, 必须定义一个新的任务空间坐标系。具体来说, 让 x 为由下式定义的 2×1 任务空间向量

$$x = \begin{bmatrix} u \\ v \end{bmatrix} \quad (9.23)$$

其中 u 和 v 定义了一个旋转坐标系, 使得 u 表示到表面的法向距离, v 表示沿表面的切向距离。如式 (??) 所示, 任务空间坐标可以用关节空间坐标表示为

$$x = h(q) \quad (9.24)$$

其中 $h(q)$ 被找到为

$$h(q) = \begin{bmatrix} u \cdot q_1 \\ v \cdot q_2 \end{bmatrix} \quad (9.25)$$

u 、 v 、 q_1 和 q_2 是在 (3) 中给出的点积符号中使用的适当定义的单位向量。单位向量 q_1 和 q_2 由 q_1 和 q_2 给出的固定坐标集来定义。这些单位向量定义为

$$q_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad q_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (9.26)$$

为了找到单位向量 u 和 v , 我们首先使用表面函数

$$\frac{1}{4}q_1^2 + q_2^2 = 1 \quad (9.27)$$

用一个变量 (即 q_2) 参数化表面如下:

$$q_1 = 2\sqrt{1 - q_2^2} \quad (9.28)$$

(6) 关于 q_2 的偏导数除以向量长度得到一个单位向量 (即 v), 它始终与表面相切。也就是说, v 由下式给出

$$v = \frac{1}{\sqrt{1 + 4q_2^2/(1 - q_2^2)}} \begin{bmatrix} -2q_2/\sqrt{1 - q_2^2} \\ 1 \end{bmatrix} \quad (9.29)$$

其中

$$\sqrt{1 + \frac{4q_2^2}{1 - q_2^2}} = \sqrt{\frac{1 + 3q_2^2}{1 - q_2^2}}$$

通过再次使用 (5), v 的表达式可以简化为

$$v = \frac{1}{\sqrt{1 + 3q_2^2}} \begin{bmatrix} -q_1 \\ 2q_2 \end{bmatrix} \quad (9.30)$$

其中

$$q_1 = 2\sqrt{1 - q_2^2}$$

因为向量 u 和 v 必须正交 (即 $u \cdot v = 0$), 通过 (8) 和问题的几何, 我们知道

$$u = \frac{1}{\sqrt{1 + 3q_2^2}} \begin{bmatrix} 2q_2 \\ q_1/2 \end{bmatrix} \quad (9.31)$$

将 (4)、(8) 和 (9) 代入 (3) 得到

$$h(q) = \frac{1}{\sqrt{1 + 3q_2^2}} \begin{bmatrix} 2q_2 & q_1/2 \\ -q_1 & 2q_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad (9.32)$$

任务空间雅可比矩阵从式 (??) 中找到, 利用 T 是单位矩阵的事实。也就是说, $J(q)$ 由下式给出

$$J(q) = \frac{1}{\sqrt{1 + 3q_2^2}} \begin{bmatrix} 1 + 3q_2^2 & 2q_2 \\ 3q_1q_2/2 & 2 - 3q_2^2 \end{bmatrix} \quad (9.33)$$

其中

$$q_1 = 2\sqrt{1 - q_2^2}$$

根据式 (??), 机器人机械臂方程由下式给出

$$M\ddot{q} + G + F(q) + J^T f = \tau \quad (9.34)$$

其中 τ 、 M 、 q 、 G 、 f 和 $F(q)$ 如示例 9.2.1 所定义。重要的是要注意法向力 (即 f_1) 和切向力 (即 f_2) 是按 (1) 给出的任务空间坐标系的方向绘制的 (见图 9.2.6)。

9.2.3 N 连杆机械臂的刚度控制

既然我们已经得到了包含环境相互作用力的机器人机械臂动力学形式, 就可以建立 n 连杆机器人机械臂的刚度控制器。如前所述, 施加在环境上的力定义为

$$f = K_e(x - x_e) \quad (9.35)$$

其中 K_e 是一个 $n \times n$ 对角、半正定常数矩阵, 用于表示环境刚度, x_e 是任务空间中测量的 $n \times 1$ 向量, 用于表示环境的静态位置。注意, 如果机械臂在特定任务空间方向不受约束, 则矩阵 K_e 的相应对角元素被假定为零。此外, 在刚度控制公式中, 环境表面摩擦通常被忽略。

多维刚度控制器是 PD 型控制器

$$\tau = J^T(q)(-K_v\dot{x} + K_p(x_d - x)) + G(q) \quad (9.36)$$

其中 K_v 和 K_p 是 $n \times n$ 对角、常数、正定矩阵, 任务空间跟踪误差定义为

$$\tilde{x} = x_d - x$$

如前所述, x_d 用于表示我们希望将机器人机械臂移动到的期望恒定末端执行器位置; 然而, x_d 现在是一个 $n \times 1$ 向量。将式 (??) 和式 (??) 代入式 (??) 得到闭环动力学

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F(\dot{q}) + J^T(q)K_e(x - x_e) = J^T(q)(-K_v\dot{x} + K_p(x_d - x)) \quad (9.37)$$

为了分析式 (??) 给出的系统的稳定性, 我们利用类李雅普诺夫函数

$$V = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}(x_d - x)^T K_p(x_d - x) + \frac{1}{2}(x_e - x)^T K_e(x_e - x) \quad (9.38)$$

将式 (??) 关于时间微分并利用式 (??) 得到

$$\dot{V} = \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{M}(q)\dot{q} - \dot{x}^T K_p(x_d - x) + \dot{x}^T K_e(x_e - x) \quad (9.39)$$

注意, 在式 (??) 中, 我们使用了 x_e 和 x_d 是常数的事实, 并且标量函数或对角矩阵的转置等于该函数或矩阵。将式 (??) 代入式 (??) 并利用式 (??) 得到

$$\dot{V} = -\dot{q}^T F(\dot{q}) - \dot{q}^T J^T(q)K_v J(q)\dot{q} \quad (9.40)$$

将斜对称性质 (见第 2 章) 应用于式 (??) 得到

$$\dot{V} = -\dot{q}^T F(\dot{q}) - \dot{q}^T J^T(q)K_v J(q)\dot{q} \quad (9.41)$$

它是非正的。由于矩阵 $J(q)$ 以及因此 $J^T(q)K_v J(q)$ 是非奇异的, \dot{V} 只能在 $\dot{q} = 0$ 且因此 $\ddot{q} = 0$ 的轨迹上保持为零 (见第 1 章的拉萨尔定理)。将 $\dot{q} = 0$ 和 $\ddot{q} = 0$ 代入式 (??) 并利用式 (??) 得到

$$J^T(q)K_e(x - x_e) = J^T(q)K_p(x_d - x) \quad (9.42)$$

或等价地

$$x_{iss} = \frac{K_{pi}x_{di} + K_{ei}x_{ei}}{K_{pi} + K_{ei}} \quad (9.43)$$

其中下标 i 用于表示向量 x 、 x_d 、 x_e 的第 i 个分量以及矩阵 K_p 和 K_e 的第 i 个对角元素。

上面的稳定性分析可以解释为意味着当任务空间坐标由式 (??) 给出时, 机器人机械臂将停止移动。也就是说, 末端执行器的最终位置或稳态位置由式 (??) 给出, 这在单自由度情况下等价于由式 (??) 给出。为了获得施加在环境上的稳态力的第 i 个分量, 我们将式 (??) 代入式 (??) 的第 i 个分量, 得到

$$f_{iss} = \frac{K_{pi}K_{ei}}{K_{pi} + K_{ei}}(x_{di} - x_{ei}) \quad (9.44)$$

因此末端执行器施加在环境上的稳态力由式 (??) 给出, 这在单自由度情况下等价于由式 (??) 给出。如在单自由度情况下一样, 我们假设对于要力控制的任务空间方向, K_{ei} 远大于 K_{pi} 。也就是说, 式 (??) 中的稳态力可以近似为

$$f_{iss} \approx K_{pi}(x_{di} - x_{ei}) \quad (9.45)$$

因此, K_{pi} 可以被解释为在这些任务空间方向上指定机械臂的刚度。

如果机械臂在任务空间方向上不受约束, 则相应的刚度常数 K_{ei} 等于零。将 $K_{ei} = 0$ 代入式 (??) 得到

$$x_{iss} = x_{di} \quad (9.46)$$

这意味着对于不受约束的任务空间方向, 我们获得了设定点控制; 因此, 在稳态下达到期望的位置设定点。刚度控制器以及相应的稳定性结果都在表 9.2.1 中总结。

示例 9.2-3: 笛卡尔机械臂的刚度控制器

我们希望为图 9.2.5 中给出的机器人机械臂系统设计并模拟一个刚度控制器。控制目标是将末端执行器移动到期望的最终位置 $v_d = 3$ m, 同时施加 2 N 的最终期望法向力。我们忽略表面摩擦 (即 f_2) 和关节摩擦, 并假设法向力 (即 f_1) 满足关系

$$f_1 = k_e(u - u_e) \quad (9.47)$$

其中 $u = \frac{1}{\sqrt{2}}(q_1 - q_2)$ 和 $k_e = 1000$ N/m。机器人连杆质量被假定为单位质量, 初始末端执行器位置由下式给出

$$v = \frac{1}{\sqrt{2}}(q_1 + q_2) = 0 \quad (9.48)$$

为了完成控制目标, 表 9.2.1 中的刚度控制器由下式给出

$$\tau = J^T(-K_v\dot{x} + K_p\tilde{x}) + G \quad (9.49)$$

其中 τ 、 J 、 G 和 x 如示例 9.2.1 所定义, u_d 定义为期望法向位置, 增益矩阵 K_v 和 K_p 取为 $K_v = k_v I$ 和 $K_p = k_p I$ 。对于这个示例, 我们选择 $k_v = k_p = 10$, 这将保证 $k_p \ll k_e$, 如刚度控制公式所要求的那样。为了满足控制目标 $f_{d1} = 2$ N, 我们利用式 (??) 来确定期望法向位置。具体来说, 将 f_{d1} 、 k_e 和 u_e 的值代入

$$f_{d1} = k_p(u_d - u_e) \quad (9.50)$$

得到

$$2 = 10\left(u_d - \frac{1}{\sqrt{2}}\right)$$

$$u_d = 0.3414$$

图 9.2.7 所示的机械臂系统 (图 9.2.5) 的刚度控制器 (3) 的仿真结果表明, 期望的切向位置和法向力在约 4 秒内达到。

9.3 混合位置/力控制

第??节给出的刚度控制器的一个主要缺点是它只能用于设定点控制; 换句话说, 期望的末端执行器机械臂位置和施加在环境上的期望力必须是常数。在许多机器人应用

中，例如磨削，末端执行器必须跟踪沿物体表面的期望位置轨迹，同时跟踪施加到物体表面的期望力轨迹。在这种类型的应用中，刚度控制器将不能充分执行；因此，必须使用另一种控制方法。

所谓的混合位置/力控制器 [Chae et al. 1988] 和 [Raibert and Craig 1981] 可用于同时跟踪位置和力轨迹。混合位置/力控制器的基本概念是通过任务空间公式将位置和力控制问题解耦为子任务。正如我们所见，任务空间公式在确定哪些方向应该受力或位置控制方面很有价值。也就是说，位置和力控制子任务很容易从任务空间公式中确定。在确定控制子任务后，可以分别开发位置控制器和力控制器。

9.3.1 笛卡尔二连杆臂的混合位置/力控制

为了说明这种混合位置/力控制的概念，考虑图 9.3.1 给出的机器人机械臂系统。对于这个应用，沿表面的位置和施加在表面上的法向力都应该被控制；因此，必须确定哪些变量应该受力控制，哪些应该受位置控制。

根据第??节给出的任务空间概念，图 9.3.1 给出的机械臂系统的任务空间公式为

$$x = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad (9.51)$$

任务空间雅可比矩阵由下式给出

$$J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

如式 (??) 所示，对于这个问题的任务空间和关节空间是等价的；因此，我们将在整个问题中将关节变量称为任务空间变量。

为了为机械臂系统设计位置/力控制器，我们必须首先确定式 (??) 给出的任务空间公式的动力学方程。使用这个任务空间公式并忽略关节摩擦，可以证明机械臂动力学为

$$M\ddot{q} + G + f = \tau \quad (9.52)$$

其中 τ 、 M 、 G 和 f 如示例 9.2.1 所定义。式 (??) 表示的矩阵形式中的两个动力学方程是

$$m_1\ddot{q}_1 + m_1g + f_1 = \tau_1 \quad (9.53)$$

和

$$m_2\ddot{q}_2 + f_2 = \tau_2 \quad (9.54)$$

在制定混合位置/力控制器时，我们为式 (??) 和式 (??) 给出的动力学设计了单独的控制。如图 9.3.1 所示，沿任务空间方向 q_2 的位置应该受位置控制；因此，我们应该使用式 (??) 来设计位置控制器。[这很明显，因为式 (??) 给出的动力学不包含任务空间变量 q_2 。]

因为我们正在设计一个位置控制器来跟踪期望轨迹，我们将定义“切向空间”跟踪误差为

$$e = q_{d2} - q_2 \quad (9.55)$$

其中 q_{d2} 表示沿表面或切向表面的期望位置轨迹。位置控制器将是计算力矩控制器（见第 3 章）

$$\tau_2 = m_2(a_M + k_{Tv}\dot{e} + k_{Tp}e) + f_2 \quad (9.56)$$

其中

$$a_M = \ddot{q}_{d2} \quad (9.57)$$

k_{Tv} 和 k_{Tp} 是正控制增益。将式 (??) 代入式 (??) 得到位置跟踪误差系统

$$\ddot{e} + k_{Tv}\dot{e} + k_{Tp}e = 0 \quad (9.58)$$

利用 k_{Tv} 和 k_{Tp} 是正数的事实，我们可以对式 (??) 应用标准线性控制结果，得到

$$\lim_{t \rightarrow \infty} e = 0$$

因此，式 (??) 给出的控制器保证了渐近位置跟踪。注意，位置控制器需要测量关节位置、关节速度和表面摩擦力；因此，从实现的角度来看，位置控制器中需要力测量。

式 (??) 给出的位置控制器将确保沿环境表面的良好位置跟踪；然而，我们还希望控制施加在环境上的力。对于图 9.3.1 给出的机械臂系统，任务空间方向垂直于表面的是 q_1 ；因此，我们将假设在这个方向上环境可以被建模为弹簧。具体来说，施加在环境上的法向力 f_1 由下式给出

$$f_1 = k_e(q_1 - q_e) \quad (9.59)$$

其中 k_e 表示环境刚度， $q_e = 3$ 。将式 (??) 关于时间的二阶导数得到表达式

$$\ddot{f}_1 = k_e\ddot{q}_1 \quad (9.60)$$

其中法向任务空间加速度用设法向力的二阶导数来书写。将式 (??) 代入式 (??) 得到力动力学方程

$$\frac{m_1}{k_e}\ddot{f}_1 + f_1 = \tau_1 - m_1g \quad (9.61)$$

我们现在可以使用式 (??) 给出的力动力学方程来设计一个力控制器来跟踪期望的力轨迹。首先，定义力跟踪误差为

$$e_f = f_{d1} - f_1 \quad (9.62)$$

其中 f_{d1} 表示要施加在环境上的期望法向力。与位置控制器类似，力控制器将是计算力矩控制器

$$\tau_1 = m_1g + \frac{m_1}{k_e}(a_N + k_{Nv}\dot{e}_f + k_{Np}e_f) + f_1 \quad (9.63)$$

其中

$$a_N = \ddot{f}_{d1} \quad (9.64)$$

k_{Nv} 和 k_{Np} 是正控制增益。将式 (??) 代入式 (??) 得到力跟踪误差系统

$$\ddot{e}_f + k_{Nv}\dot{e}_f + k_{Np}e_f = 0 \quad (9.65)$$

利用式 (??) 中 k_{Nv} 和 k_{Np} 是正数的事实, 得到

$$\lim_{t \rightarrow \infty} e_f = 0$$

因此, 式 (??) 给出的控制器保证了渐近力跟踪。重要的是要认识到力控制器需要测量法向力及其导数。因为力导数通常不用于测量, 所以由式 (??) 制造, 即

$$\dot{f}_1 = k_e \dot{q}_1 \quad (9.66)$$

因此, 环境刚度和法向任务空间速度用于模拟力的导数。

9.3.2 N 连杆机械臂的混合位置/力控制

上一节给出的混合位置/力控制器可以通过使用任务空间公式概念轻松扩展到多自由度情况。具体来说, 可以开发一个反馈线性化控制, 使机器人机械臂方程全局线性化, 然后开发线性控制器来跟踪期望的力和位置轨迹。

首先, 控制设计者选择一个任务空间公式

$$x = h(q) \quad (9.67)$$

使得法向和切向表面运动如第??节所讨论的那样被分解。式 (??) 给出的机器人动力学然后用任务空间加速度表示, 通过对式 (??) 关于时间求二阶导数得到

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (9.68)$$

其中 $J(q)$ 是式 (??) 中定义的任务空间雅可比矩阵。求解式 (??) 中的 \ddot{q} 得到

$$\ddot{q} = J^{-1}(q)(\ddot{x} - \dot{J}(q)\dot{q}) \quad (9.69)$$

将式 (??) 代入式 (??) 得到

$$\ddot{x} = J(q)M^{-1}(q)(\tau - V_m(q, \dot{q})\dot{q} - G(q) - F(\dot{q})) + \dot{J}(q)\dot{q} - J(q)M^{-1}(q)J^T(q)f \quad (9.70)$$

式 (??) 给出的动力学对应的反馈线性化控制由下式给出

$$\tau = M(q)J^{-1}(q)(\ddot{x} - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f \quad (9.71)$$

其中 a 是一个 $n \times 1$ 向量, 用于表示线性位置和力控制策略, 稍后将讨论。将式 (??) 代入式 (??) 后, 我们有

$$\ddot{x} = a \quad (9.72)$$

从式 (??) 中, 我们可以看到任务空间运动已经被全局线性化和解耦; 因此, 我们可以为每个任务空间自由度独立设计位置和力控制器。具体来说, 可以为表示切向运动的任务空间变量设计线性位置控制器; 此外, 可以为表示法向力的任务空间变量设计线性力控制器。

因为式 (??) 给出的动力学在任务空间中已经被解耦, 我们将 x 的切向空间分量定义为 x_{Ti} , 其中下标 T 用于表示切向空间, 下标 i 用于表示 x_T 的第 i 个分量。根据这个符号, 式 (??) 的切向空间分量由下式给出

$$\ddot{x}_{Ti} = a_{Ti} \quad (9.73)$$

其中 a_{Ti} 是第 i 个线性切向空间位置控制器。为了反馈控制的目的, 我们定义切向空间跟踪误差为

$$e_{Ti} = x_{Tdi} - x_{Ti} \quad (9.74)$$

其中 x_{Tdi} 表示沿环境表面切向的第 i 个期望位置轨迹。如上一节所示, 相应的线性控制器由下式给出

$$a_{Ti} = \ddot{x}_{Tdi} + k_{Tvi}\dot{e}_{Ti} + k_{Tpi}e_{Ti} \quad (9.75)$$

k_{Tvi} 和 k_{Tpi} 是第 i 个正控制增益。将式 (??) 代入式 (??) 得到位置跟踪误差系统

$$\ddot{e}_{Ti} + k_{Tvi}\dot{e}_{Ti} + k_{Tpi}e_{Ti} = 0 \quad (9.76)$$

利用式 (??) 中 k_{Tvi} 和 k_{Tpi} 是正数的事实, 得到

$$\lim_{t \rightarrow \infty} e_{Ti} = 0$$

因此, 保证了渐近位置跟踪。

为了力控制的目的, 我们将 x 的法向空间分量定义为 x_{Nj} , 其中下标 N 用于表示法向空间, 下标 j 用于表示 x_N 的第 j 个分量。根据这个符号, 式 (??) 的法向空间分量由下式给出

$$\ddot{x}_{Nj} = a_{Nj} \quad (9.77)$$

其中 a_{Nj} 是第 j 个线性法向空间力控制器。如上一节所示, 我们假设环境可以被建模为弹簧。具体来说, 施加在环境上的法向力 f_{Nj} 由下式给出

$$f_{Nj} = k_{ej}(x_{Nj} - x_{ej}) \quad (9.78)$$

其中 k_{ej} 是环境刚度的第 j 个分量, x_{ej} 用于表示法向空间 x_{Nj} 方向上的环境静态位置。

如对上一小节中的单自由度机器人所做的那样，在开发力控制器之前，我们必须制定力动力学。对式 (??) 关于时间求二阶导数得到表达式

$$\ddot{f}_{Nj} = k_{ej}\ddot{x}_{Nj} \quad (9.79)$$

其中法向任务空间加速度用第 j 个法向力的二阶导数来书写。将式 (??) 代入式 (??) 得到力动力学

$$\frac{1}{k_{ej}}\ddot{f}_{Nj} = a_{Nj} \quad (9.80)$$

为了反馈控制的目的，我们定义力跟踪误差为

$$e_{fj} = f_{Ndj} - f_{Nj} \quad (9.81)$$

其中 f_{Ndj} 表示要垂直施加在环境上的期望力的第 j 个分量。如上一节所示，相应的线性控制器由下式给出

$$a_{Nj} = \frac{1}{k_{ej}}(\ddot{f}_{Ndj} + k_{Nvj}\dot{e}_{fj} + k_{Npj}e_{fj}) \quad (9.82)$$

k_{Nvj} 和 k_{Npj} 是第 j 个正控制增益。将式 (??) 代入式 (??) 得到力跟踪误差系统

$$\ddot{e}_{fj} + k_{Nvj}\dot{e}_{fj} + k_{Npj}e_{fj} = 0 \quad (9.83)$$

利用式 (??) 中 k_{Nvj} 和 k_{Npj} 是正数的事实，得到

$$\lim_{t \rightarrow \infty} e_{fj} = 0$$

因此，保证了渐近力跟踪。

混合位置/力控制器以及相应的稳定性结果都在表 9.3.1 中总结。

9.4 混合阻抗控制

阻抗控制基于这样的概念：控制器应用于调节机器人机械臂运动与施加在环境上的力之间的动态行为 [Hogan 1987]，而不是分别考虑运动和力控制问题。使用这个概念，控制设计者指定机械臂运动和施加在环境上的力之间的期望动态行为。这种期望行为有时被称为目标阻抗，因为它用于表示运动和力之间的欧姆定律类型关系。

9.4.1 环境建模

正如 [Hogan 1987] 中所指出的，环境模型对任何力控制策略都是至关重要的。在前面讨论的力控制策略中，环境简单地被建模为弹簧；然而，可以想象，简单的弹簧模型可能不足以描述所有类型的环境。为了对多种类型的环境进行分类，我们使用线性传递函数关系

$$f(s) = Z_e(s)\dot{x}(s) \quad (9.84)$$

其中变量 s 是拉普拉斯变换变量, f 表示施加在环境上的力, \dot{x} 表示机械臂在环境接触点的速度, $Z_e(s)$ 表示环境阻抗。目前, 所有量都假定为标量函数; 然而, 在本节末尾, 我们将阻抗控制推广到多维情况。

量 $Z_e(s)$ 被称为阻抗, 因为式 (??) 表示运动和力之间的欧姆定律类型关系。如在电路理论中, 环境阻抗可以分为不同的类别。为了进一步讨论阻抗控制, 我们现在给出三种常用的用于分类环境阻抗的类别。

定义 9.1. 当且仅当 $|Z(0)| = 0$ 时, 阻抗是惯性的。

惯性环境的说明由图 9.4.1a 给出。该图描绘了机器人机械臂以速度 \dot{x} 移动质量为 h 的载荷。相应的相互作用力由下式给出

$$f = h\ddot{x}$$

因此, 利用式 (??) 得到惯性环境阻抗为

$$Z_e(s) = hs \quad (9.85)$$

我们可以通过应用定义??轻松验证这个阻抗确实是惯性的。

定义 9.2. 当且仅当 $|Z(0)| = c$ 时, 阻抗是电阻性的, 其中 $0 < c < \infty$ 。

电阻性环境的说明由图 9.4.1b 给出。该图描绘了机器人机械臂以速度 \dot{x} 穿过液体介质。液体介质被假定为具有阻尼系数 b 。相应的相互作用力由下式给出

$$f = b\dot{x}$$

因此, 利用式 (??) 得到电阻性环境阻抗为

$$Z_e(s) = b \quad (9.86)$$

我们可以通过应用定义??轻松验证这个阻抗确实是电阻性的。

定义 9.3. 当且仅当 $|Z(0)| = \infty$ 时, 阻抗是电容性的。

电容性环境的说明由图 9.4.1c 给出。该图描绘了机器人机械臂以速度 \dot{x} 推动质量为 h 的物体。物体被假定为具有阻尼系数 b 和弹簧常数 k 。相应的相互作用力由下式给出

$$f = b\dot{x} + k \int \dot{x} dt$$

因此, 利用式 (??) 得到电容性环境阻抗为

$$Z_e(s) = b + \frac{k}{s} \quad (9.87)$$

我们可以通过应用定义??轻松验证这个阻抗确实是电容性的。

9.4.2 位置和力控制模型

正如我们在前一节中看到的，环境可以被建模为由式 (??) 中的力/速度关系定义的阻抗。对于我们的阻抗控制公式，我们将假设环境阻抗是惯性的、电阻性的或电容性的。问题就变成了：如何为给定的环境阻抗设计控制器？解决方案是通过制定机械臂阻抗模型 $Z_m(s)$ [Anderson and Spong 1988] 来获得的。换句话说，在对环境建模之后选择机械臂阻抗（或目标阻抗）。选择机械臂阻抗的标准与机械臂的动态性能有关。也就是说，选择机械臂阻抗使得对阶跃输入（可能是力或速度命令）的稳态误差为零。正如我们将展示的，如果机械臂阻抗是环境阻抗的对偶，就可以实现这个性能标准。

在完全阐述对偶概念之前，必须制定位置和力控制的模型。对于位置控制 [Anderson and Spong 1988]，力与速度之间的关系由下式建模

$$f = Z_m(s)(\dot{x}_d - \dot{x}) \quad (9.88)$$

其中 \dot{x}_d 表示机械臂在环境接触点的输入速度， $Z_m(s)$ 表示机械臂阻抗。

如我们随后将展示的，机械臂阻抗 $Z_m(s)$ 被选择为通过对阶跃输入利用 x 和 \dot{x}_d 之间的动态关系来“归零”稳态。为了确定 x 和 \dot{x}_d 之间的动态关系，我们将式 (??) 与式 (??) 组合，得到图 9.4.2 给出的位置控制框图。我们可以使用这个框图来说明对偶概念。具体来说，我们检查稳态速度误差

$$E_{ss} = \lim_{s \rightarrow 0} s(\dot{x}_d - \dot{x}) \quad (9.89)$$

其中 $\dot{x}_d(s) = 1/s$ 对于阶跃速度输入。利用图 9.4.2，我们可以很容易地证明式 (??) 可以简化为

$$E_{ss} = \lim_{s \rightarrow 0} \frac{Z_e(s)}{Z_e(s) + Z_m(s)} \quad (9.90)$$

对于式 (??) 中的 E_{ss} 等于零， $Z_e(s)$ 必须是非电容性阻抗 [即 $Z_e(s)$ 必须是惯性或电阻性的]， $Z_m(s)$ 必须是非惯性阻抗。也就是说，如果惯性环境用非惯性机械臂阻抗进行位置控制，而电阻性环境用电容性机械臂阻抗进行位置控制，则对速度阶跃输入可以实现零稳态误差。上述术语“对偶”用于强调惯性环境阻抗可以用电容性机械臂阻抗进行位置控制。

从上面的发展来看，很明显电容性环境不能被位置控制并保持零稳态误差规范。然而，随后我们将展示电容性环境可以在保持零稳态误差规范的同时被力控制。关于力控制，力与速度之间的动态关系被建模为 [Anderson and Spong 1988]

$$\dot{x} = Z_m^{-1}(s)(f_d - f) \quad (9.91)$$

其中 f_d 用于表示在环境接触点施加的输入力。

如我们随后将展示的，机械臂阻抗 $Z_m(s)$ 被选择为通过对阶跃输入利用 f 和 f_d 之间的动态关系来“归零”稳态。为了确定 f 和 f_d 之间的动态关系，我们将式 (??) 与式

(??) 组合，得到图 9.4.3 给出的力控制框图。我们可以使用这个框图来说明对偶概念。具体来说，我们检查稳态力误差

$$E_{ss} = \lim_{s \rightarrow 0} s(f_d - f) \quad (9.92)$$

其中 $f_d(s) = 1/s$ 对于阶跃力输入。利用图 9.4.3，我们可以很容易地证明式 (??) 可以简化为

$$E_{ss} = \lim_{s \rightarrow 0} \frac{Z_m(s)}{Z_e(s) + Z_m(s)} \quad (9.93)$$

对于式 (??) 中的 E_{ss} 等于零， $Z_e(s)$ 必须是非惯性阻抗， $Z_m(s)$ 必须是非电容性阻抗。也就是说，如果电容性环境用非电容性机械臂阻抗进行力控制，而电阻性环境用惯性机械臂阻抗进行力控制，则对力阶跃输入可以实现零稳态误差。术语“对偶”用于强调电容性环境阻抗可以用惯性机械臂阻抗进行力控制。

上面的讨论可以由以下对偶原理总结。

对偶原理 电容性环境用非电容性机械臂阻抗进行力控制，惯性环境用非惯性机械臂阻抗进行位置控制，电阻性环境用惯性机械臂阻抗进行力控制或用电容性机械臂阻抗进行位置控制。

注：读者可以通过[交互式可视化工具](#)理解阻抗控制的概念，调节虚拟刚度、阻尼和质量参数，观察机器人对外力的顺应响应。

9.4.3 阻抗控制公式

既然我们已经展示了如何基于这个模型将环境和机械臂建模为阻抗，我们就基于这个模型开发一个“阻抗”控制器。为了利用阻抗控制方法，控制设计者选择一个任务空间公式

$$x = h(q) \quad (9.94)$$

用于特定的位置/力控制应用。如第??节所示，我们可以证明力矩控制

$$\tau = M(q)J^{-1}(q)(a - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f \quad (9.95)$$

得到线性方程组

$$\ddot{x} = a \quad (9.96)$$

其中 a 是一个 $n \times 1$ 向量，用于表示阻抗位置和力控制策略。

如式 (??) 所阐述的，任务空间运动已经被全局线性化；因此，我们可以为每个任务空间自由度设计单独的位置和力控制器。也就是说，在每个由分量 x_k 表示的任务空间方向上，分配 x_k 与相应环境力 f_k 之间的环境阻抗关系。基于这种环境阻抗的分配，对偶原理用于确定 a 的相应元素应该是力控制器还是位置控制器。在做出这个决定后，然后使用式 (??) 和式 (??) 来获得 a 的特定位置和力控制分量。

作为将位置控制设计与力控制设计分离的手段，我们使用式 (9.96) 来定义在任务空间方向上受位置控制的方程为

$$\ddot{x}_{pi} = a_{pi} \quad (9.97)$$

其中下标 i 表示第 i 个受位置控制的任务空间变量，下标 p 表示位置控制。与受位置控制的任务空间方向相关的环境力用 f_{pi} 表示。

假设零初始条件，式 (9.97) 的拉普拉斯变换可以写成

$$s\dot{x}_{pi} = a_{pi} \quad (9.98)$$

从式 (9.98) 给出的位置控制模型，我们也可以将式 (9.97) 的左侧写成

$$f_{pi} = Z_{pmi}(s)(\dot{x}_{pdi} - \dot{x}_{pi}) \quad (9.99)$$

其中 Z_{pmi} 是第 i 个受位置控制的机械臂阻抗。因此，将式 (9.99) 和式 (9.98) 相等得到第 i 个位置控制器

$$a_{pi} = s\dot{x}_{pdi} - sZ_{pmi}^{-1}(s)L^{-1}\{f_{pi}\} \quad (9.100)$$

其中 L^{-1} 用于表示逆拉普拉斯变换操作。

继续分离位置控制和力控制设计，我们使用式 (9.99) 来定义在任务空间方向上受力控制的方程为

$$\ddot{x}_{fj} = a_{fj} \quad (9.101)$$

其中下标 j 表示第 j 个受力控制的任务空间变量，下标 f 表示力控制。与受力控制的任务空间方向相关的环境力用 f_{fj} 表示。

假设零初始条件，式 (9.101) 的拉普拉斯变换可以写成：

$$s\dot{x}_{fj} = a_{fj} \quad (9.102)$$

从式 (9.102) 给出的力控制模型，式 (9.101) 的左侧也可以写成

$$sZ_{fmj}^{-1}(s)(f_{fdj} - f_{fj}) = a_{fj} \quad (9.103)$$

或

$$s\dot{x}_{fj} = Z_{fmj}^{-1}(s)(f_{fdj} - f_{fj}) \quad (9.104)$$

其中 Z_{fmj} 是第 j 个受力控制的机械臂阻抗。因此，将式 (9.103) 和式 (9.104) 相等得到第 j 个力控制器

整体的“混合”阻抗控制策略通过将式 (9.99) 与式 (9.103) 和式 (9.104) 一起使用来获得。这种混合阻抗控制策略在表 9.4.1 中总结。注意，高级控制器将用于选择要受位置或力控制的任务空间分量，然后分配适当的机械臂阻抗。如对偶原理所阐述的，式 (9.99) 中的机械臂阻抗 Z_{pmi} 被分配为非电容性的，式 (9.103) 中的机械臂阻抗 Z_{fmj} 被分配为非惯性的。

9.5 降维状态位置/力控制

如果一个刚性机械臂受到刚性环境的约束，自由度就会降低，因为机械臂末端执行器不能穿过环境移动；因此，关于位置的一个或多个自由度会丢失。因此，当机械臂末端执行器接触环境约束时，末端执行器和环境之间会产生相互作用力（有时称为约束力）。这种“减少”位置自由度同时发展约束力的过程使人相信应该根据这一自然现象设计位置/力控制器。

最近，研究人员开始制定一个理论框架 [McClamroch and Wang 1988]、[Kankaanranta and Koivo 1988]，利用动力学中的经典结果将约束力的影响纳入机器人机械臂模型。假设应该为接触刚性约束的刚性机械臂设计控制器的理由是，在大多数力控制应用中，环境比机械臂刚硬得多。因此，假设机械臂是刚性的而环境是柔性的似乎不合理。

9.5.1 完整约束对机械臂动力学的影响

对于本节给出的控制器，我们假设环境约束是完整且无摩擦的。也就是说，我们假设存在一个关节空间坐标中的约束函数 $\psi(q)$ （一个 $p \times 1$ 向量函数）满足

$$\psi(q) = 0 \quad (9.105)$$

式 (??) 给出的关系说明环境约束是完整的。假设约束函数的维度小于关节变量的数量（即 $p < n$ ）。对于特定问题，从机器人运动学和环境配置中找到函数 $\psi(q)$ 。为了阐明完整概念，我们现在给出一个示例。

示例 9.5-1: 完整约束

我们希望为图 9.2-5 和 9.2-6 中给出的机器人/环境配置制定约束函数 $\psi(q)$ 。在这两种配置中，关节空间的维度为 2，约束函数的维度为 1（即，约束是一维表面）。对于图 9.2.5 给出的机械臂系统，约束函数由下式给出

$$\frac{1}{\sqrt{2}}(q_1 - q_2) - \frac{1}{\sqrt{2}} = 0$$

对于图 9.2.6 给出的机械臂系统，约束函数由下式给出

$$\frac{1}{4}q_1^2 + q_2^2 - 1 = 0$$

对于具有完整且无摩擦约束的通用 n 连杆机器人机械臂，约束机器人动力学可以写成形式

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + A^T(q)\lambda = \tau \quad (9.106)$$

其中 λ 是一个 $p \times 1$ 向量，表示与约束相关的广义力乘数，约束雅可比矩阵 $A(q)$ 是一个 $p \times n$ 矩阵，定义为

$$A(q) = \frac{\partial \psi(q)}{\partial q} \quad (9.107)$$

如 [Kankaanranta and Koivo 1988] 中所述, 我们假设 $A^T(q)$ 的 p 列在关节空间上线性独立。此外, 注意力变量 λ 独立于 q 和 \dot{q} 。

为了激励式 (??) 的来源, 我们重新检查拉格朗日方程

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = \tau \quad (9.108)$$

其中受约束机器人机械臂的修正拉格朗日量由下式给出

$$L = K - P - \lambda^T \psi(q) \quad (9.109)$$

注意, 式 (??) 中给出的拉格朗日量实际上与第 2 章中给出的拉格朗日量相同, 因为

$$\psi(q) = 0$$

如式 (??) 所要求的。

从式 (??) 和式 (??) 中, 我们可以看到机器人机械臂动力学方程的结构对于受约束的机械臂与对于不受约束的机械臂相同, 除了拉格朗日方程中替换 $\psi(q) = 0$ 所产生的任何新项之外。具体来说, 式 (??) 给出的约束机器人机械臂方程是通过将式 (??) 代入式 (??) 并利用恒等式

$$\frac{\partial}{\partial q} (\lambda^T \psi(q)) = A^T(q) \lambda$$

得到的。

在我们开发降维状态位置/力控制器之前, 我们简要地从启发式的角度解释位置和力变量如何被降维。首先, 注意对于式 (??) 给出的约束机器人动力学, 变量 λ 用于表示应该被控制的约束力。注意, λ 的维度是 p , 根据定义, 小于 n 。在前面给出的环境力公式中, 力的维度被假定为等于 n 。在这种方法中, 我们能够减少必须控制的力的数量, 因为我们最初假设约束表面是无摩擦的。当然, 在现实中, 表面摩擦将存在于大多数机器人力控制应用中; 然而, 它通常被视为干扰, 因为这样的摩擦是所施加的法向接触力的函数。

其次, 我们检查式 (??) 给出的机器人运动的位置约束。因为根据假设 $\psi(q) = 0$, 我们可以对式 (??) 关于时间求导得到

$$A(q) \dot{q} = 0 \quad (9.110)$$

其中 $A(q)$ 在式 (??) 中定义。式 (??) 给出的表达式给出了运动学速度约束的简洁形式。从式 (??) 和式 (??) 给出的位置和速度约束, 我们可以说明机械臂动力学属于在 R^{2n} 中由 C 定义的不变流形

$$C = \{(q, \dot{q}) : \psi(q) = 0, A(q) \dot{q} = 0\}$$

也就是说, 机器人机械臂的运动保持在由 C 定义的流形上。如 [McClamroch and Wang 1988] 中所述, 流形 C 在 R^{2n} 上是奇异的; 因此, 我们可以降低运动动力学的阶数。

9.5.2 降维建模和控制

由单个 n 关节非冗余机械臂受刚性环境约束组成的机器人系统具有 $n - p$ 个运动自由度。注意，然而，如式 (??) 给出的机械臂关节变量模型包含 n 个位置变量 (q)，结合 p 个力变量 (λ)，使控制变量（即状态）的总数 $n + p$ 超过控制输入的数量 n 。在本节中，我们展示了如何使用变量变换来降低动力学模型的状态，从而将控制变量的数量从 $n + p$ 减少到 n 。

降维模型可以通过用另一组独立坐标（称为约束空间坐标）表示式 (??) 给出的机械臂动力学来获得，这组坐标将用 x 表示。预期 x 是一个 $(n - p) \times 1$ 关节空间坐标向量。也就是说， x 是 q 的一个子集。

通过假设存在一个将约束空间向量 x [一个 $(n - p) \times 1$ 向量] 与关节空间向量 q 相关联的 $n \times 1$ 向量函数 $g(x)$ 来降低约束机器人模型。该函数由下式给出

$$q = g(x) \quad (9.111)$$

其中函数 $g(x)$ 必须被选择为使得

$$A(g(x))\Sigma(x) = 0 \quad (9.112)$$

并且 $n \times (n - p)$ 雅可比矩阵 $\Sigma(x)$ 定义为

$$\Sigma(x) = \frac{\partial g(x)}{\partial x} \quad (9.113)$$

在由 x 给出的约束空间运动中包含 $n - p$ 个独立行。

即使约束空间向量 x 的维度小于关节空间向量 q 的维度，通常也能够找到式 (??) 中给出的从 x 到 q 的函数映射。对于特定问题，式 (??) 给出的完整方程和机器人运动学的代数关系用于找到 $g(x)$ 。还要注意， $g(x)$ 的选择是非唯一的，并且式 (??) 中给出的 $g(x)$ 条件以及 $\Sigma(x)$ 行的条件是相互关联的。具体来说， $\Sigma(x)$ 包含 $n - p$ 个独立行的条件确保了解耦模型表示 $n - p$ 个独立方程。式 (??) 中给出的 $g(x)$ 条件用于确保由 λ 表示的力可以与由 x 表示的约束空间运动解耦。应该注意的是，由于约束被假定为完整的，并且矩阵 $A^T(q)$ 被假定为具有 p 个线性独立的列，我们可以证明上述两个条件始终成立。读者可以参考 [McClamroch and Wang 1988] 了解详情。

为了获得用约束空间坐标表示的降维动力学，我们首先对式 (??) 关于时间求导得到

$$\dot{q} = \Sigma(x)\dot{x} \quad (9.114)$$

其中 $\Sigma(x)$ 在式 (??) 中定义。对式 (??) 关于时间求导得到

$$\ddot{q} = \Sigma(x)\ddot{x} + \dot{\Sigma}(x)\dot{x} \quad (9.115)$$

在将式 (??)、式 (??) 和式 (??) 中的 q 、 \dot{q} 和 \ddot{q} 分别代入式 (??) 后，我们得到降维模型

$$M(x)\Sigma(x)\ddot{x} + N(x, \dot{x}) + A^T(x)\lambda = \tau \quad (9.116)$$

其中

$$N(x, \dot{x}) = (V_m(x, \dot{x})\Sigma(x) + M(x)\dot{\Sigma}(x))\dot{x} + G(x) + F(x)$$

式 (??) 给出的降维模型的意义通过用 $\Sigma^T(x)$ 前乘式 (??) 来说明

$$M^*\ddot{x} + N^* = \Sigma^T\tau \quad (9.117)$$

其中

$$M^* = \Sigma^T M \Sigma, \quad N^* = \Sigma^T N$$

注意, 式 (??) 中的接触力已被式 (??) 的结果移除, 式 (??) 确保 $A\Sigma = 0$ 。式 (??) 给出的模型是有用的, 因为控制机械臂在约束表面上运动的动力学已经从 n 个微分方程减少到 $n-p$ 个微分方程; 此外, 运动已经与接触力解耦。这两个结果在后续位置/力控制器的设计和稳定性分析中很重要。

在介绍降维状态位置/力控制器之前, 我们给出一些关于位置/力跟踪问题的定义。首先, 约束位置跟踪误差定义为

$$e = x_d - x \quad (9.118)$$

我们假设期望的约束空间轨迹及其前两个导数, 分别用 x_d 、 \dot{x}_d 和 \ddot{x}_d 表示, 都是有界函数。我们还假设期望的力乘数轨迹 λ_d 是一个已知的界函数, 由此定义了相应的力乘数跟踪误差变量

$$e_\lambda = \lambda_d - \lambda \quad (9.119)$$

降维状态位置/力控制器是一个反馈线性化控制器; 因此, 需要精确了解机器人动力学。降维状态控制器 [McClamroch and Wang 1988] 是

$$\tau = M\Sigma(\ddot{x}_d + K_v\dot{e} + K_p e) + N + A^T(\lambda_d + K_f e_\lambda) \quad (9.120)$$

其中 K_v 和 K_p 是对角、正定的 $(n-p) \times (n-p)$ 矩阵, K_f 是对角、正定的 $p \times p$ 矩阵。

为了确定位置误差和力跟踪误差的稳定性类型, 我们将式 (??) 代入式 (??) 得到

$$M\Sigma(\ddot{e} + K_v\dot{e} + K_p e) + A^T(e_\lambda + K_f e_\lambda) = 0 \quad (9.121)$$

用 Σ^T 前乘式 (??) 得到

$$M^*(\ddot{e} + K_v\dot{e} + K_p e) = 0 \quad (9.122)$$

作为式 (??) 的结果。因为 $\Sigma^T(x)$ 在约束空间运动中包含 $n-p$ 个独立列, $M(x)$ 是一个正定对称矩阵, M^* 是一个正定对称矩阵。因此, 我们可以用 M^{*-1} 前乘式 (??) 得到

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \quad (9.123)$$

因为 K_v 和 K_p 是对角正定矩阵, 可以对式 (??) 应用标准线性控制论证, 得到

$$\lim_{t \rightarrow \infty} e = 0 \quad (9.124)$$

为了获得力跟踪误差的稳定性结果，我们将式 (??) 代入式 (??) 得到

$$A^T(I + K_f)e_\lambda = 0 \quad (9.125)$$

其中式 (??) 中的 I 是 $p \times p$ 单位矩阵。因为 $A^T(x)$ 的 p 列被假定为线性独立的，复合矩阵 $(I + K_f)$ 是一个正定对角矩阵，我们可以将式 (??) 写成

$$\lim_{t \rightarrow \infty} e_\lambda = 0 \quad (9.126)$$

如式 (??) 和式 (??) 所阐述的，式 (??) 给出的降维状态位置/力控制器对约束位置误差和力跟踪误差都产生了渐近稳定性结果。降维状态位置/力控制策略在表 9.5.1 中总结。

9.6 总结

在本章中，给出了几种用于刚性机器人的位置/力控制策略。目的是按时间顺序记录力控制的发展。一些新的研究领域，如执行器动力学、传感器动力学、关节柔性、机械臂动态不确定性、表面不确定性和环境影响对位置/力控制器性能的影响，现在正在被许多研究人员研究。在解决了其中一些问题之后，可以期望看到机器人机械臂更频繁地用于力控制应用。

参考文献

- [1] Anderson, R., and M. Spong, “Hybrid impedance control of robotic manipulators,” *J. Robot. Autom.*, vol. 4, no. 5, pp. 549–556, Oct. 1988.
- [2] Chae, A., C. Atkeson, and J. Hollerbach, *Model-Based Control of a Robot Manipulator*. Cambridge, MA: MIT Press, 1988.
- [3] Hogan, N., “Stable execution of contact tasks using impedance control,” *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 595–601, Raleigh NC, Mar. 1987.
- [4] Kankaanranta, R., and H. Koivo, “Dynamics and simulation of compliant motion of a manipulator,” *IEEE Trans. Robot. Autom.*, vol. 4, pp. 163–173, Apr. 1988.
- [5] Lipkin, H., and J. Duffy, “Hybrid twist and wrench control for a robot manipulator,” *Trans. ASME J. Mech. Transmissions Autom. Design*, vol. 110, pp. 138–144, June 1988.
- [6] McClamroch, N., and D. Wang, “Feedback stabilization and tracking of constrained robots,” *IEEE Trans. Autom. Control*, vol. 33, no. 5, pp. 419–426, May 1988.
- [7] Raibert, M., and J. Craig, “Hybrid position/force control of manipulators,” *J. Dyn. Syst. Meas. Control*, vol. 102, pp. 126–132, June 1981.
- [8] Salisbury, J., and J. Craig, “Active stiffness control of manipulator in Cartesian coordinates,” *Proc. 19th IEEE Conf. Decision Control*, Dec. 1980.
- [9] Spong, M., and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.

习题

第 9.2 节

9.2-1 对于图 9.2.6 中给出的机械臂，使用新的表面函数找到施加在环境表面上的法向力和切向力相关的任务空间雅可比矩阵

$$\frac{1}{4}q_1^2 + q_2^2 = 1$$

9.2-2 为图 9.2.5 中给出的机器人机械臂系统设计并模拟一个刚度控制器，表面函数由下式给出

$$\frac{1}{2}q_1 - q_2 = 5$$

控制目标是将末端执行器移动到 $q_{d2} = 0.3$ m 的期望最终位置，同时施加 2 N 的最终期望法向力。忽略表面和关节摩擦，并假设法向力满足关系

$$f_1 = k_e(u - u_e)$$

其中 $k_e = 100$ N/m， u_e 是到表面的静态法向距离。机器人连杆质量可以假定为单位质量，而初始末端执行器位置被假定为由下式给出

$$v = \frac{1}{\sqrt{2}}(q_1 + q_2) = 0$$

第 9.3 节

9.3-1 由于混合位置/力控制策略的位置和力误差系统是线性的，讨论如何选择控制器增益来改变机械臂末端执行器的性能。

9.3-2 为图 9.2.5 中给出的机器人机械臂系统设计并模拟一个混合位置/力控制器，表面函数由下式给出

$$\frac{1}{\sqrt{2}}q_1 + \frac{1}{\sqrt{2}}q_2 - 5 = 0$$

控制目标是使末端执行器沿表面轨迹移动

$$v_d = \sin(t) \text{ m}$$

同时施加期望法向力

$$f_{d1} = 1 + e^{-t} \text{ N}$$

忽略表面和关节摩擦，并假设法向力满足关系

$$f_1 = k_e(u - u_e)$$

其中 $k_e = 1000 \text{ N/m}$, u_e 是到表面的静态法向距离。机器人连杆质量可以假定为单位质量, 而初始末端执行器位置被假定为由下式给出

$$u = \frac{1}{\sqrt{2}}(q_1 - q_2) = 0$$

9.3-3 解释为什么混合位置/力控制策略实际上是一种位置控制策略。

第 9.4 节

9.4-1 假设对于力或位置控制, 机械臂阻抗被选择为

$$Z_m(s) = h + Z_r(s)$$

其中 h 是正常数, $Z_r(s)$ 是一个适当稳定的传递函数。使用上述机械臂阻抗, 证明混合阻抗控制器只需要测量关节位置、关节速度和接触力。

9.4-2 为图 9.2.5 中给出的机器人机械臂系统设计一个混合阻抗控制器, 表面函数由下式给出

$$\frac{1}{\sqrt{2}}q_1 + \frac{1}{\sqrt{2}}q_2 - \frac{3}{\sqrt{2}} = 0$$

忽略表面和关节摩擦, 并假设法向力和切向力满足关系

$$f_1 = k_e(u - u_e) + b_e \dot{u}$$

其中 $k_e = 10 \text{ N/m}$, $b_e = 1 \text{ N-s/m}$, u_e 是到表面的静态法向距离。

9.4-3 明确展示如何使用表 9.4.1 找到示例 9.4.1 中的方程 (6)。

第 9.5 节

9.5-1 为图 9.2.6 中给出的机器人机械臂系统设计一个降维状态控制器。

9.5-2 对于在问题 9.5-1 中开发的控制器, 展示法向力 (即 f_1) 如何与力乘数 (即 λ) 相关, 以及降维运动变量 (即 x) 如何与沿表面的运动 (即 v) 相关。

Chapter 10

机器人控制实现与软件

机器人操控器研究的多样性以及相关的硬件/软件系统的复杂要求，一直是软件开发者面临的挑战。因此，许多先前开发的软件控制平台复杂、昂贵且对用户不太友好。尽管以前的几个平台被设计为提供基于开放架构的系统，但很少被重复使用。为了解决以往的缺点，本章描述了一个名为**机器人平台 (Robotic Platform)** 的软件控制平台的设计和实现。机器人平台是一个用于机器人操控器应用的开放式软件开发平台。它包括硬件接口、伺服控制、轨迹生成、3D 仿真、图形用户界面和数学库。与分布式解决方案不同，机器人平台在单一硬件平台上、使用单一编程语言 (C++) 和单一操作系统实现所有这些组件，同时保证硬实时性能。这种设计产生了一个开放架构，它不那么复杂、更容易使用和扩展，并建立在以下最先进的技术和通用组件之上，以进一步提高简单性和可靠性：i) PC 技术，ii) QNX 实时操作系统，iii) Open Inventor 和 STL 库，iv) 面向对象设计，v) 易于使用、先前开发的低级控制环境。

10.1 引言

由于软件控制平台的构建需要覆盖广泛学科领域的构建模块 (见图 10.1.1)，因此创建一个可以被研究人员用于不同应用的通用平台是非常理想的。考虑到机器人应用的多样性和研究领域，这项任务变得更加具有挑战性，因为机器人操控器供应商通常向用户专用的硬件组件上运行的专有机人控制语言。由于缺乏专有机人控制语言的灵活性和性能，以前的一些工作侧重于在常用编程语言之上构建机器人控制库 (RCCL [?] 和 ARCL [?] 就是这种库的例子)。

虽然以前的许多软件方法通过使用通用编程语言实现了新的灵活性和性能水平，但在 80 年代和 90 年代初期开发的许多软件控制平台由于当时操作系统和硬件组件的限制而本质上是复杂的。也就是说，大多数操作系统不支持实时编程，因此促进了 RCI [?] 和 Chimera [?] 等项目的发展。此外，过程式编程语言如“C”在复杂项目的可重用性方面往往达到极限。此外，那个时代硬件组件的有限性能迫使系统开发者利用分布式架构

(参见 [?], [?] 和 [?]), 集成了专有硬件和软件的混合体。

除了分布式系统的明显优势(例如,更大的可扩展性和更强的计算能力)之外,还有几个缺点。具体来说,分布式架构显著增加了软件的复杂性。此外,网络连接上的硬实时行为通常需要昂贵的专有硬件。总的来说,硬件成本也更高;此外,用户必须熟悉不同的硬件架构和操作系统。尽管许多软件控制平台试图灵活、可重构和开放,但这些平台很少被重复使用(与非机器人平台如 Open Inventor 或 Corba 相比)。显然,工程师认为从头开始开发他们的应用程序更快、更容易。事实上,正如许多研究人员所指出的,过去安装、学习和修改先前开发的软件控制平台的学习曲线非常陡峭。

在过去十年中,计算领域发生了许多创新。具体来说,面向对象软件设计 [?] 的出现促进了更复杂项目的管理,同时也促进了代码重用和灵活性。例如,RIPE [?], MMROC+ [?], OSCAR [?] 和 ZERO++ [?] 等机器人控制库在机器人编程中利用了面向对象技术。工程界还见证了用于 PC 的类 Unix 实时操作系统的 proliferation [?], 这有利于用专有硬件组件替换实时控制 [?]

同样,在硬件领域,工程界见证了高速、低成本 PC、快速 3D 图形视频板和经济型运动控制卡的出现。因此,PC 平台现在提供了多功能功能,从而使复杂的软件架构和专有硬件组件变得多余。由于这些技术进步,QMotor 机器人工具包(QMotor RTK) [?] 被开发出来。QMotor RTK 是最早将实时操控器控制和图形用户界面(GUI)集成在单个 PC 上的软件控制平台之一。尽管 QMotor RTK 有许多优点,但由于其所基于的实时操作系统(即 QNX4)的限制,它缺少一些功能。例如,它缺乏 3D 仿真能力,并且不支持线程的使用。

在 1990 年代末,领先的实时软件开发公司 QSSL 发布了 QNX6 [?](即 QNX6/Neutrino 操作系统),对 QNX4 进行了重大升级。这种基于微内核的操作系统可能是目前最先进的实时操作系统;此外,它还有用于软件开发和多媒体应用的附加组件。QNX6 的新特性极大地促进了 QMotor RTK 升级版本的潜在能力,事实上,QMotor RTK 被完全转换为一个名为机器人平台 [?] 的新软件控制平台。本章的其余部分通过检查机器人平台的优点、设计和操作,提供其广泛的概述。需要强调的是”广泛”这个词,因为在本书的一章中详细解释如何为机器人操控器开发软件控制平台实际上是不可能的。尽管如此,希望本章能让读者更好地了解相关的复杂性。

10.2 工具与技术

鉴于最近的技术进步,下一代软件控制平台应该将伺服控制回路、轨迹生成、任务级程序、GUI 程序和 3D 仿真集成在统一的软件架构中,这似乎几乎是显而易见的。也就是说,下一代软件控制平台应该利用单一硬件平台(例如 PC)、单一操作系统(例如 QNX6 实时操作系统 [?])和单一编程语言(例如 C++ [?])。这种下一代软件控制平台的优点如下:

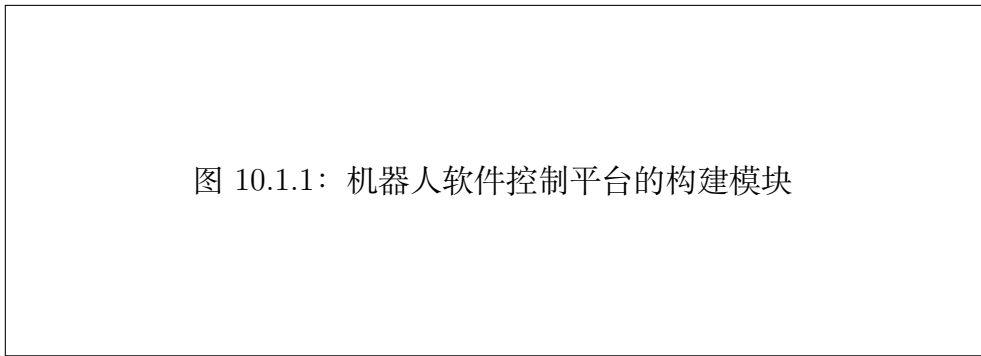


图 10.1.1: 机器人软件控制平台的构建模块

图 10.1: 机器人软件控制平台的构建模块

简单性。统一的非分布式架构比分布式异构架构要小得多、简单得多。它更容易安装、理解和扩展。简单性对于激励代码在不同应用中的重用至关重要。

性能。统一的非分布式架构还确保高性能和硬实时行为，因为进程间通信的开销最小。

各级灵活性。平台的每个组件都对扩展和修改开放。许多过去的平台在某些级别上利用了开放架构，但其他级别是在专有硬件上实现的，因此无法修改。

成本。机器人平台比分布式平台需要更少的硬件组件。基本上，一台带有一个或多个输入/输出 (I/O) 板的 PC 就足够了。此外，与其他硬件平台相比，PC 硬件非常经济实惠。

为了更详细地解释这些优点并说明机器人平台如何降低软件开发和复杂性，现在检查用于开发机器人平台的通用工具和技术。从硬件引擎即标准桌面 PC 开始讨论似乎是合乎逻辑的。由于其在消费市场的普及，标准桌面 PC 在过去 20 年中发生了巨大的变化。过去，只有昂贵的工作站提供了控制机器人系统所需的处理能力，而 PC 已经赶上甚至超过了工作站的性能 [?]。与 UNIX 工作站相比，基于 PC 的系统允许更多的硬件和软件组件种类。此外，这些组件和 PC 本身通常比其 UNIX 对应物便宜；此外，PC 是一个众所周知、大多数软件开发人员都感到舒适的平台。

任何软件控制平台的第二个最重要的部分是操作系统。由于可靠性对于机器人操作极其重要，因此选择了 QSSL 的 QNX6 实时操作系统 [?]。QNX6 的编程接口符合 POSIX 标准，是第一个结合以下特性的平台：

硬实时。QNX6 提供硬实时响应（例如，它可以在一定频率下执行控制循环而不会落后）。实时功能集成在微内核中。

自主开发。软件开发和执行在同一环境和同一 PC 上进行。

快速 3D 图形。QNX6 提供硬件加速 3D 支持，这是机器人工作单元图形仿真所需的。

线程。QNX6 支持进程的并发执行，即线程。

设备驱动架构。在 QNX6 下开发设备驱动程序非常容易，这是集成不同机器人硬件所需的。

成本。QNX6 也非常经济实惠，因为它对非商业用途免费，并且在低成本的标准 PC 上运行。

任何软件控制平台的第三个最重要的部分或设计方面是编程理念。对于开发机器人操控器控制软件，面向对象编程比过程式编程有三个主要优点。首先，它提供了语言结构，使得编程界面更加容易。例如，矩阵乘法可以用简单的"*"表示，类似于 MATLAB 编程。其次，面向对象编程允许系统架构非常灵活但又非常简单。也就是说，系统的组件（类）可以具有内置的默认行为和默认设置。使用这种默认行为和/或默认设置可以让程序员减少代码大小。然而，程序员仍然可以为特定应用程序覆盖默认行为。最后，面向对象编程通过促进独立于特定实现的组件的开发来支持泛型编程（例如，通用类"Manipulator"可以与不同类型的操控器一起工作）。

所有这些优点都基于面向对象编程的一般概念，如抽象、封装、多态性和继承 [?]。正如本章后面将要展示的，这些面向对象设计概念产生了一个直观的系统架构。正如人们所期望的，选择的语言是 C++，因为它提供了面向对象概念的整个范围，同时保持高性能 [?]（例如，可以使用模板和内联函数在机器人平台中创建高度通用和优化的代码）。

当使用强大的面向对象库如 STL [?] 时，C++ 和面向对象编程的能力可以进一步增强。STL 提供了标准数据结构（例如，列表、向量等）和算法。这些标准数据结构和算法简化了编程工作，因为这些工具在常见编程任务中经常被需要。STL 的数据结构非常通用，因为它们可以与不同的数据类型一起使用（即模板编程）。

软件控制平台的最后一个设计方面与机器人专家如何与系统交互有关。具体来说，机器人专家如何为仿真目的设置对象的动画，以及如何实现和调整控制算法。对于仿真动画，使用了 Open Inventor [?]。Open Inventor 由 Silicon Graphics 开发，是一个用于创建 3D 图形和动画的面向对象 C++ 库。Open Inventor 最大限度地减少了开发工作，因为它能够加载以虚拟现实建模语言（VRML）格式创建的 3D 模型 [?]。有各种软件包可以促进 3D VRML 模型的构建。这些包可用于快速创建机器人组件的图形表示。机器人平台利用 Open Inventor 的内置功能来设置这些组件的动画。Open Inventor 还提供了高级功能，如对象拾取，这在操作界面编程中非常有用。

对于控制实现，使用了 QMotor [?]，因为它允许用户建立实时控制循环、记录数据、调整控制参数和绘制信号。由于 QMotor 支持面向对象编程，QMotor 控制程序可以直接集成到机器人平台的类设计中。

10.3 机器人平台的设计

10.3.1 概述

机器人平台的每个组件（例如，示教器、轨迹生成器等）都由一个 C++ 类建模。C++ 类定义结合了与该组件相关的数据和函数。例如，类"Puma560"包含与 Puma 560

机器人相关的数据（例如，当前关节位置）以及与 Puma 560 机器人相关的函数（例如，启用机械臂电源）。因此，机器人平台的设计源于以有意义和直观的方式将数据和函数分组到多个类中。类还可以通过从基类派生来使用另一个类的功能和数据的一部分。机器人平台的类包括 GUI 组件和用于图形仿真的 3D 模型。这些类型的组件传统上出现在不同的程序中（例如，参见 RCCL 机器人仿真器 [?])。然而，通过将它们包含在同一个类中，可以实现用户界面、3D 建模和其他功能部分之间的紧密集成。

为了为不同的应用程序提供新功能，用户创建新类。通常，新类是从一个已存在的类派生出来的，以最小化编码工作。这个过程称为继承，它极大地促进了代码重用并消除了冗余。

为了说明类如何从彼此派生，使用了类层次结构图。机器人平台的主类层次结构图如图 10.3.1 所示。每个箭头从派生类指向父类。列出的类越靠右，它必须变得越具体。列出的类越靠左，它必须变得越通用。机器人平台的类可以分为以下几类（注意，下面讨论的一些类没有显示在图 10.3.1 中，因为它们的类层次结构将在后面讨论）：

核心类。类 `RoboticObject`、`FunctionalObject` 和 `PhysicalObject` 通过定义所有机器人对象的总体行为来构建机器人平台的核心。这些类本身可以作为任何机器人系统的基础。

通用机器人类。从核心类派生的是些通用机器人类。这些类（例如，`ServoControl`、`Manipulator` 和 `Gripper`）不能被实例化。相反，这些类作为基类实现通用功能，同时向程序员呈现通用接口（即，这些类可用于创建独立于特定硬件或特定算法的程序）。

特定机器人类。从通用机器人类派生的是实现特定硬件的类（例如，类 `Puma560` 实现 Puma 560 机器人）或特定功能组件（例如，类 `DefaultPositionControl` 实现比例积分微分（PID）位置控制器）。

ControlProgram 类。这个类是所有实时控制循环的基础。

工具类。工具类管理常见任务（例如，管理配置文件）。

对象管理器。对象管理器监督系统中存在的所有对象和类。

数学库。数学库由用于矩阵、向量、变换、滤波器、微分器和积分器的类组成。

操控器模型库。操控器模型库由对操控器的运动学和动力学行为进行建模的类组成。

在机器人控制程序中，用户从类实例化对象。实例化对象时，为对象保留内存，对象初始化自身。用户可以根据需要从同一类创建任意数量的对象。例如，通过简单地创建两个 `Puma560` 类的对象，就可以操作两个 Puma 机器人。一旦创建了对象，用户就可以使用它们的功能。对象管理器维护所有当前存在对象的列表。通过对象管理器，可以在多个对象上启动功能（例如，关闭所有对象）。场景查看器（Scene Viewer）是机器人平台的默认 GUI。它包含查看机器人工作单元 3D 场景和列出所有对象的窗口。整体运行时架构如图 10.3.2 所示。

在机器人系统中，不同的组件彼此相关。为了反映这一事实，在对象之间建立了对



图 10.3.1: 机器人平台的类层次结构

图 10.2: 机器人平台的类层次结构



图 10.3.2: 机器人平台的运行时架构

图 10.3: 机器人平台的运行时架构

象关系。例如，对象可以指定它们之间的物理连接。对象关系通过 C++ 指针实现到相关对象。示例场景中的对象关系如图 10.3.3 所示。

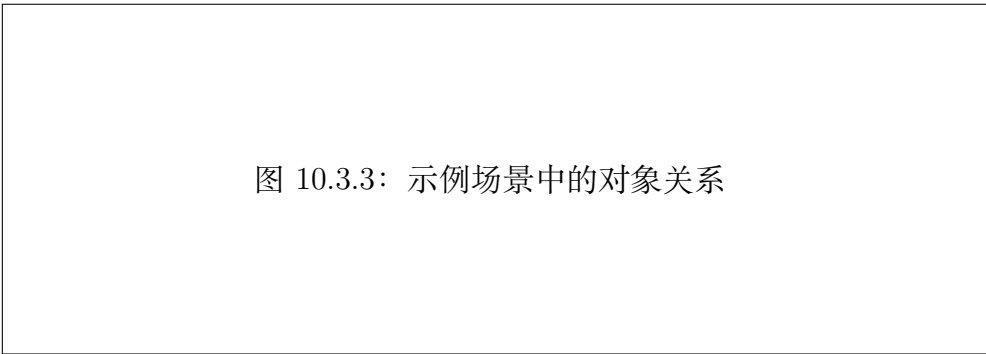


图 10.3.3: 示例场景中的对象关系

图 10.4: 示例场景中的对象关系

10.3.2 核心类

类 `RoboticObject` 是所有类的基类。它定义了一个通用接口（即，可以与机器人平台的所有类一起使用的一组函数）。例如，程序可以使用 `startShutdown()` 函数来指示 `Puma560` 类或 `Gripper` 类的对象关闭。具体来说，以下通用功能在类 `RoboticObject` 中定义（另见图 10.3.4）：

错误处理。每个对象必须指示其错误状态。

交互式命令。每个对象可以定义一组交互式命令（例如，“打开夹爪”），这些命令将显示在场景查看器的对象弹出菜单中。

配置管理。每个对象可以使用全局配置文件进行设置。此外，每个对象还可以有一个本地对象配置文件。

关闭行为。每个对象都能够关闭自身。

GUI 控制面板。每个对象都能够创建一个控制面板。

消息处理器。每个对象都有一个可以解释自定义消息的消息处理器。

线程管理。每个对象指示其操作是否需要额外的线程，并且还提供执行这些额外线程的函数。

注意，实际功能通常在派生类中实现。然而，类 `RoboticObject` 也实现了简单的默认功能。此功能通过让从类 `RoboticObject` 派生的所有类在需要时接管此默认功能的能力来支持代码重用和简单性。表 10.3.1 显示了所有函数及其默认行为。

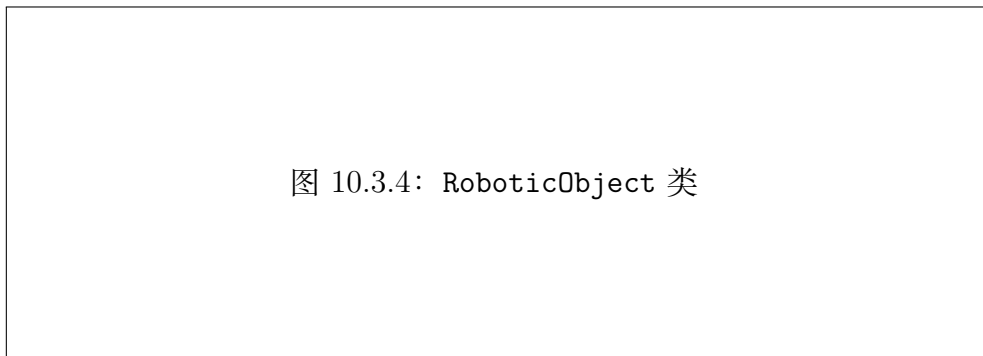


图 10.3.4: RoboticObject 类

图 10.5: RoboticObject 类

类 `PhysicalObject` 从类 `RoboticObject` 派生。它是代表物理对象（例如，操纵器、传感器、夹爪等）的所有类的基类。它为这些类定义了一个通用接口，如图 10.3.5 和表 10.3.2 所示。具体来说，以下通用功能在 `PhysicalObject` 中定义：

3D 可视化。每个物理对象可以创建其 Open Inventor 3D 模型。场景查看器循环遍历所有物理对象以创建整个 3D 场景。

对象连接。物理对象可以将另一个对象指定为安装位置。通过使用这种对象关系，场景查看器在正确的位置绘制对象（例如，绘制安装在机器人操纵器末端的夹爪）。

位置和方向。位置/方向信息指定对象在场景中的绝对位置（或安装位置，如果指定了对象连接）。

仿真模式。每个物理对象可以锁定在仿真模式中。也就是说，对象不执行任何硬件 I/O（即，其行为只是仿真的）。

类 `FunctionalObject` 不包含任何功能；但是，它用于构建其他类。例如，功能类（如类 `TrajectoryGenerator`）是从类 `FunctionalObject` 派生的。

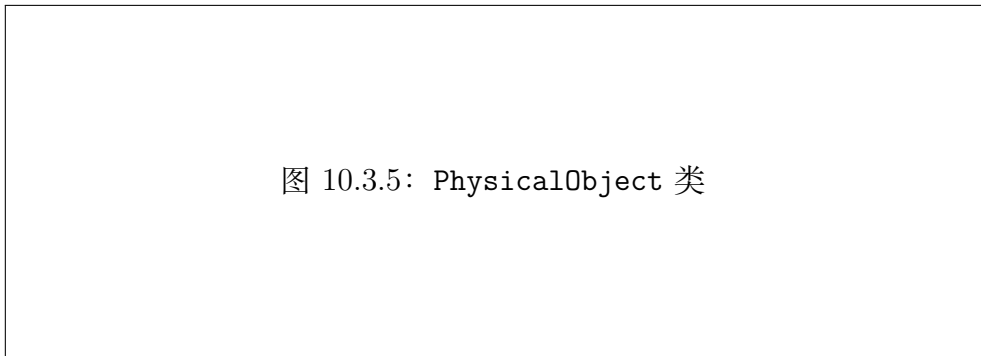


图 10.3.5: PhysicalObject 类

图 10.6: PhysicalObject 类

10.3.3 机器人控制类

任何机器人工作单元的核心组件是操控器。Manipulator 类是一个通用类，定义了具有任意数量关节的操控器的通用功能。从 Manipulator 类派生的是 DefaultManipulator 类，它包含操控器的默认实现。DefaultManipulator 类是一个模板类（即，它以关节数量作为参数）。表 10.3.3 说明了通用编程接口和默认实现。从 DefaultManipulator 类派生的是实现特定操控器类型的类。目前，支持三种操控器：Puma 560 机器人、IMI 机器人 [?] 以及 Barrett 全臂操控器（WAM） [?]（包括 4 连杆和 7 连杆配置）。关于这些机器人操控器特定控制实现的更多信息可以在 [?] 和 [?] 中找到。

为了仿真操控器，需要它们的动力学模型。此外，对于笛卡尔控制，需要正向/反向运动学和雅可比矩阵的计算。所有这些函数都位于 ManipulatorModel 类中。ManipulatorModel 类的类层次结构和通用接口如图 10.3.6 所示。

DefaultManipulator 类创建实时控制循环，但它不包含伺服控制算法。伺服控制算法包含在 ServoControl 类或其派生类的单独对象中。ServoControl 类是一个通用类，定义了伺服控制的通用属性和功能。也就是说，伺服控制从操控器对象获取当前位置，并计算实现控制任务所需的输出控制信号（例如，将操控器伺服到期望的设定点）。

从 ServoControl 类派生的是 DefaultPositionControl 类，它在关节空间和笛卡尔空间中包含具有摩擦补偿的 PID 位置控制。DefaultManipulator 类中的控制循环首先确定操控器的当前位置，然后调用 ServoControl 类的 calculate() 函数。伺服控制作为单独对象的优点是用户可以在操控器移动时切换多个伺服控制算法。

轨迹生成也在一个单独的类中执行。TrajectoryGenerator 类定义了通用轨迹生成器的接口。轨迹生成器是创建连续设定点流并将其转发给伺服控制对象的任何对象。轨迹生成器必须实现自己的循环，以便轨迹生成器可以独立运行，并以与伺服控制中使用的频率不同的频率运行。伺服控制调用轨迹生成器的 getCurrentSetpoint() 函数来确定当前期望位置。还可以定义多个轨迹生成器并在操控器移动时在它们之间切换。

QueueTrajectoryGenerator 类从 TrajectoryGenerator 类派生，以实现通用接口和创建沿经由点和目标点的轨迹的轨迹生成器的通用功能（即，实现了运动队列管理

功能)。从 `QueueTrajectoryGenerator` 类派生的是 `DefaultTrajectoryGenerator` 类。这个类是轨迹生成器的具体实现，可以在关节空间和笛卡尔空间中进行插值，包括在经由点处两段运动之间的路径混合。表 10.3.4 显示了默认轨迹生成器的接口。

`Puma560` 或 `WAM` 等类自动创建默认轨迹生成器和默认伺服控制，以方便用户。但是，如果需要，用户可以切换到不同的轨迹生成器或伺服控制。图 10.3.7 说明了在典型场景中操控器对象、伺服控制对象和轨迹生成器对象如何协同工作。

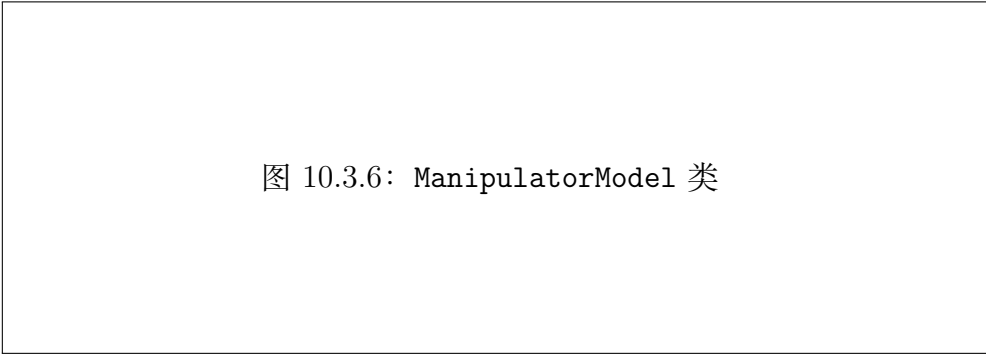


图 10.3.6: ManipulatorModel 类

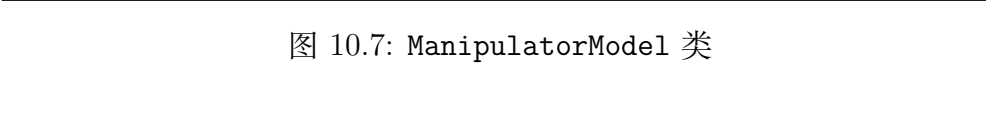


图 10.7: ManipulatorModel 类

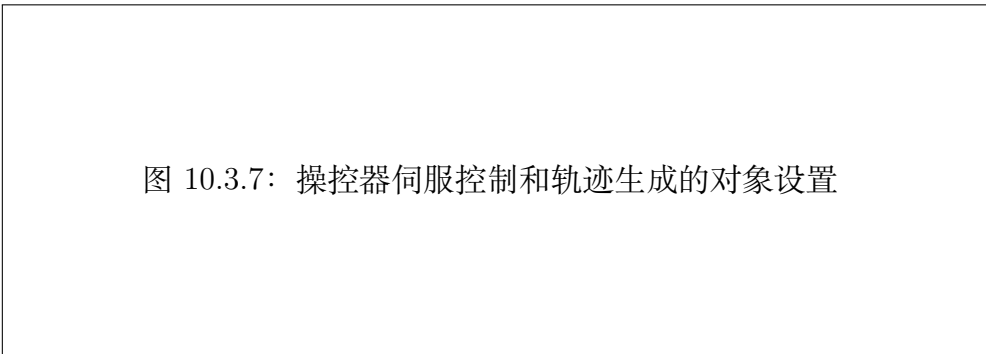


图 10.3.7: 操控器伺服控制和轨迹生成的对象设置

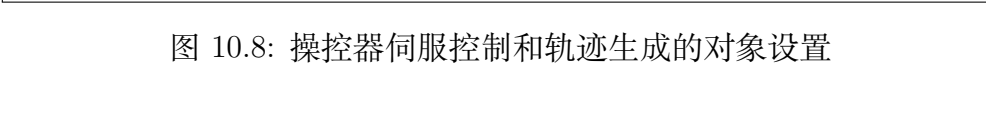


图 10.8: 操控器伺服控制和轨迹生成的对象设置

10.3.4 外部设备类

除了与特定操控器相关的类之外，还有一些其他类可用于机器人应用。具体来说，这些类使机器人操控器能够通过夹爪、机器手等工具与其他物理对象进行接口。这些外部设备类如下：

夹爪 (Gripper)。`Gripper` 类是夹爪的通用接口。基本上，它定义了 `open()`、`close()` 和 `relax()` 函数。

默认夹爪 (DefaultGripper)。`DefaultGripper` 类假设两个数字输出线控制夹爪，一个数字线打开夹爪，一个关闭它。

Barrett 手 (BarrettHand)。这是一个操作 Barrett 手 [?] 的类。这个类允许用户移动三个手指中的每一个到特定位置，并控制展开运动。

力/力矩传感器 (ForceTorqueSensor)。这个类定义了力/力矩传感器的通用接口。也就是说，它定义了读取力和力矩的函数。

ATI 力传感器 (AtiFTSensor)。这个类是 ATI 工业自动化 Gamma 30/100 力/力矩传感器的具体实现。

工具更换器 (ToolChanger)。ToolChanger 类是工具更换器的通用接口。基本上，它定义了 lock()、unlock() 和 relax() 函数。

默认工具更换器 (DefaultToolChanger)。DefaultToolChanger 类假设一个数字输出线控制工具更换器的锁定功能，另一个线控制解锁功能。

静态对象 (StaticObject)。StaticObject 类的对象除了指定其 3D 模型外没有其他功能。它可以用于向 3D 场景添加桌子或工件等对象。

10.3.5 工具类

与任何复杂的软件系统一样，需要实用程序用于通信目的和向用户提供系统相关信息。为此，在机器人平台中使用了几个单独的通用工具类。这些工具类如下：

状态 (Status)。Status 类提供了堆叠状态消息的功能。它向用户返回详细的状态报告。每个类都有一个 Status 类的数据成员，指示对象的状态。

客户端 (Client)。Client 类简化了向单独任务或线程发送消息。

服务器 (Server)。Server 类是 Client 类的对应类，简化了接收消息和回复消息。

配置 (Config)。Config 类处理 ASCII 格式的配置文件。

IO 板客户端 (IOBoardClient)。IOBoardClient 类允许与 I/O 板通信（例如，提供编码器读数、写入 D/A 值、读取 A/D 值、设置和读取数字输入/输出）。

锁 (Lock)。Lock 类在两个线程之间提供互斥。

10.3.6 配置管理

机器人平台利用一个全局配置文件，由对象管理器 and 对象解析以确定系统配置。默认情况下，这个文件称为 rp.cfg。对象管理器在由 shell 变量 RP_CONFIG_PATH 指定的路径中搜索此文件。通过使用选项 -config <filename> 启动机器人平台程序，可以指定不同的文件名和路径作为配置文件。现在讨论配置文件的格式。对于每个对象，配置文件列出括号中的对象名称、对象的类名称以及一些附加设置（见图 10.3.8）。表 10.3.5 列出了可能的对象设置及其由 RoboticObject 和 PhysicalObject 类定义的相关成员函数。派生类可用于定义附加设置。

10.3.7 对象管理器

对象管理器实现为 ObjectManager 类，处理系统中的所有对象。每次创建新对象时，它都会向对象管理器注册自己。同样，每次销毁对象时，它都会从对象管理器维护

图 10.3.8: 示例全局配置文件

图 10.9: 示例全局配置文件

的对象列表中删除。对象管理器包含遍历此列表的必要功能，因此允许对多个对象执行操作。例如，场景查看器检索从 `PhysicalObject` 类派生的所有对象的列表。此列表检索操作允许场景查看器渲染每个对象，从而渲染整个 3D 场景。

对象管理器还维护机器人平台中使用的所有类的列表。此功能对于促进通用编程至关重要。具体来说，通用代码仅利用提供通用接口的组件；因此，当使用具有不同实现的组件时，通用代码不需要更改。也就是说，相同的代码可用于许多具体实现，因此有助于代码重用（例如，通用轨迹生成器可与不同类型的操控器一起使用）。幸运的是，C++ 利用类继承和虚函数来促进通用编程。为了说明这个概念，请考虑图 10.3.9 中所示的类层次结构的一部分。

图 10.3.9: 通用 Manipulator 类及其派生类

图 10.10: 通用 Manipulator 类及其派生类

以下通用代码片段演示了如何编写操控器独立程序，使其可以与 Puma 机器人、WAM、IMI 或将来可能添加的任何机器人一起工作：

Listing 10.1: 通用操控器编程示例

```
Manipulator *manipulator;  
ObjectManager om;  
  
manipulator = om.createDerivedObject<Manipulator>("leader");  
// 根据全局配置文件中 "leader" 下指定的内容
```

```
// 创建Puma、IMI或WAM对象

// 现在，我们可以进行通用操作
manipulator->enableArmPower();
Transform t = manipulator->getCurrentCartesianPosition();
```

为了在 C++ 中利用通用编程，上述代码创建派生类的对象（例如，Puma 类或 WAM 类的对象），并通过指向通用基类的指针操作对象（即，Manipulator *manipulator）。对象管理器的 createDerivedObject() 函数用于创建 Puma560、Barrett Arm、WAM 或 IMI 类的对象。为了创建正确的对象，createDerivedObject() 函数在全局配置文件中查找对象名称（见图 10.3.8）。然后，createDerivedObject() 函数从配置文件中读取对象的类名称，并创建此类的一个对象。因此，当使用现有的通用程序时，只需更改全局配置文件中的类名称即可切换到不同的操控器类型。

10.3.8 并发/通信模型

与机器人操控器系统软件开发相关的固有问题之一是并发问题。也就是说，虽然对于许多工程系统来说作为单个任务运行就足够了，但机器人系统需要组件（例如，伺服控制）与其他组件（例如，轨迹生成器）并发执行。机器人平台通过在一个 PC 上运行所有并发任务来解决这个问题。应该注意的是，由于 QNX6 支持对称多处理，因此可以将任务分布在多个处理器上。

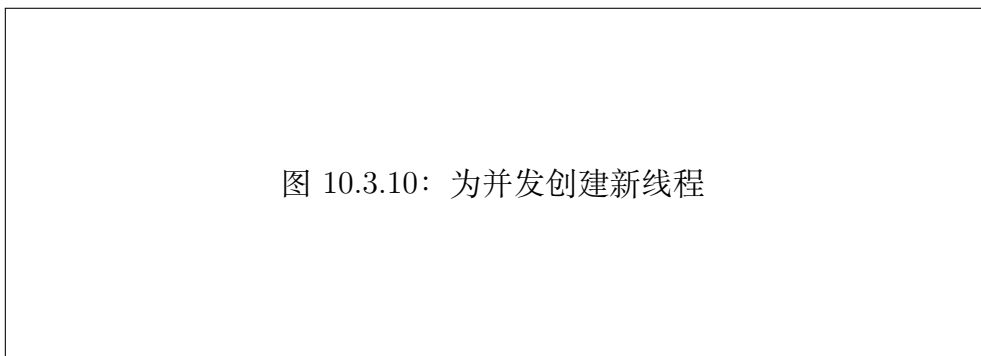


图 10.3.10: 为并发创建新线程

图 10.11: 为并发创建新线程

机器人平台的前身 QMotor RTK [?] 执行多个程序以在单个处理器上实现并发。虽然这个概念促进了模块化，但管理多个程序的启动和终止是不方便的。相比之下，使用机器人平台的应用程序被编译并链接到单个程序中。如果需要并发执行，此程序会生成线程。一旦程序终止，所有线程自动终止。图 10.3.10 说明了用户程序如何生成多个线程的示例。在程序启动时，只有线程 1 正在执行。在机器人平台库初始化时，创建了一个执行 3D 场景查看器的新线程。然后，用户使用操控器类的新对象。此操控器对象的创建会自动生成用于伺服控制循环的第三个线程。因此，第一个线程可以继续计算操控

器的目标点，而伺服控制循环和场景查看器在后台运行。为了确保时间关键任务的实时行为，线程以不同的优先级运行（例如，伺服控制循环以高优先级 27 运行）。

由于线程访问相同的地址空间，因此可以通过使用此地址空间轻松执行线程之间的通信。但是，同步此访问以避免数据结构的损坏非常重要。通过使用 Lock 类为在多个线程之间共享的数据访问提供互斥来实现同步。

10.3.9 绘图与控制调整能力

调整低级控制器和轨迹规划算法以及调试的问题在许多以前开发的软件控制平台中通常没有被解决。然而，鉴于准确定位诸如机器人操控器之类的多自由度系统的复杂性，这是一个极其重要的问题。为了解决这个问题，机器人平台利用 QMotor 进行低级算法开发。QMotor [?] 是一个用于实现和调整控制策略的综合环境。QMotor 包括：i) 用于硬件访问的客户端/服务器架构，ii) 用于创建控制程序的 C++ 库，iii) 用于控制参数调整、数据记录和绘图的自定义 GUI。

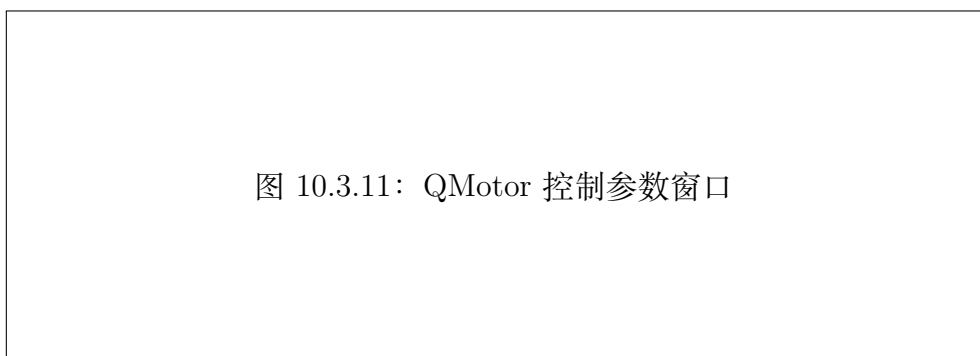


图 10.3.11: QMotor 控制参数窗口

图 10.12: QMotor 控制参数窗口

为了与硬件通信，QMotor 使用在后台运行并以固定速率执行硬件 I/O 的硬件服务器。有不同的 I/O 板服务器可用（例如，ServoToGo 板、Quanser MultiQ 板和 ATI 力/力矩传感器接口板）。硬件服务器的使用提供了抽象的客户端/服务器通信接口，使得客户端可以使用不同的服务器执行相同的通用操作。因此，可以通过简单地启动不同的服务器快速重新配置系统以使用不同的 I/O 板。对于编写控制程序，QMotor 提供了一个定义 ControlProgram 类的库。为了实现实时控制循环，用户从 ControlProgram 类派生一个特定类，并定义执行控制计算和必要内务处理的几个函数。一旦实现了控制程序并编译完成，用户可以启动 QMotor GUI，加载控制程序，启动它，并从控制参数窗口（见图 10.3.11）调整控制策略。此外，用户可以打开多个绘图窗口（见图 10.3.12）并设置记录参数。

为了将 QMotor 用于机器人平台，DefaultManipulator 等类是从 ControlProgram 类派生的。因此，这些类继承了控制程序的功能（即，实时执行、数据记录和与 GUI 通信）。如果从 ControlProgram 类派生一个类，基类 RoboticObject 会自动创建一个新

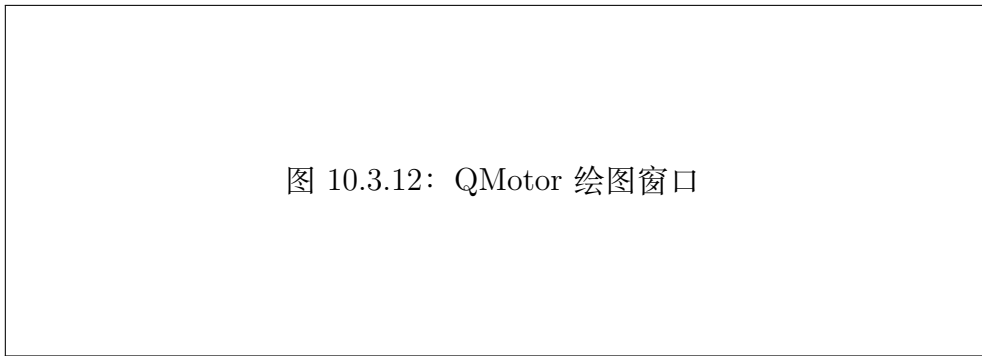


图 10.3.12: QMotor 绘图窗口

图 10.13: QMotor 绘图窗口

的执行线程，在后台运行控制循环。QMotor 最初设计为一次加载和调整一个控制程序。也就是说，要切换到不同的控制程序，必须停止并卸载当前的控制程序。机器人平台设计为同时启动和运行多个控制循环；因此，有必要升级 QMotor。具体来说，QMotor 现在可以将自己附加到正在执行的控制程序，而不是加载和启动控制程序。此 QMotor 修改现在允许机器人平台中的以下操作场景。用户首先启动控制程序。此控制程序创建几个执行实时控制循环的线程。然后，用户可以启动 QMotor GUI 并将其附加到任何控制程序。一旦 QMotor GUI 附加到控制程序，就可以更改控制参数并绘制数据。

10.3.10 数学库

过去的机器人控制库经常引入它们自己特定的机器人类型数据。大多数这些数据类型基于向量或矩阵（例如，齐次变换是一个 4×4 矩阵）。因此，使用通用的 C++ 矩阵库并在其上定义机器人类型更为可行。大多数用于 C++ 的矩阵库使用动态内存分配，这冒着失去确定性实时响应的风险 [?]。因此，使用动态内存分配的库的大缺点是它们不能在机器人平台的大部分中使用。为了克服这个问题，为机器人平台开发了特殊的实时矩阵类，使用模板表示矩阵大小。这意味着矩阵大小在编译时已知，不需要动态内存分配。除了适用于实时应用之外，此解决方案还产生高度优化的代码。也就是说，由于使用模板和内联函数，矩阵类可以像直接编程一样快。具体来说，通过优化实现，两个 2×2 矩阵的乘法 $C = A * B$ 与编写以下代码一样快：

Listing 10.2: 矩阵乘法优化示例

```
c11=a11 * b11 + a12 * b21;
c12=a11 * b12 + a12 * b22;
// ... 等等
```

另一个优点是编译器可以在编译时检查正确的矩阵大小（例如，在编译期间检测到两个大小不兼容的矩阵的矩阵乘法）。图 10.3.13、表 10.3.6 和表 10.3.7 说明了矩阵类的类层次结构、类型类和数据类型。MatrixBase、VectorBase、Matrix、ColumnVector 和 RowVector 类以元素的数据类型为参数。默认的元素数据类型是 double，这是机器

人平台的标准浮点数据类型。`MatrixBase` 和 `VectorBase` 类是纯虚基类，允许操作未知大小的矩阵和向量（在通用操控器编程期间需要未知大小的矩阵和向量）。

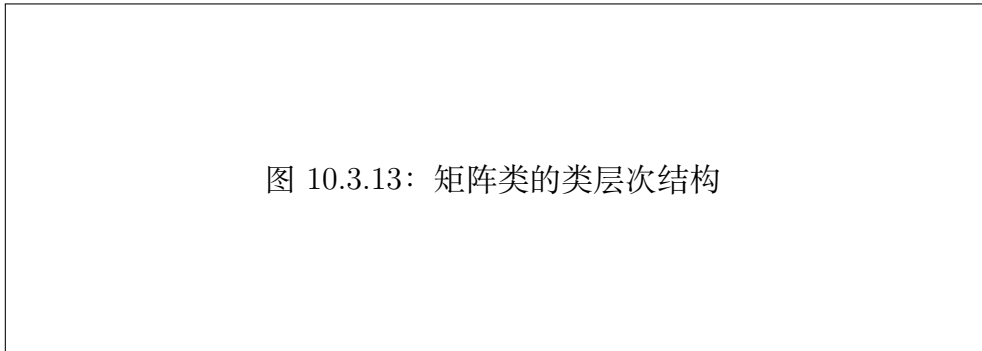


图 10.3.13: 矩阵类的类层次结构

图 10.14: 矩阵类的类层次结构

除了矩阵、向量和变换类之外，`ButterworthFilter`、`Differentiator` 和 `Integrator` 类也是数学库的一部分。`ButterworthFilter` 类用于低通滤波。`Differentiator` 类通过反向差分法加低通滤波实现数值微分。最后，`Integrator` 类利用梯形法进行数值积分。用户可以从 `Integrator` 类派生一个特定类以实现更高级的积分方法。两个类都以数据类型为参数（即，它们适用于标量、向量和矩阵）。

10.3.11 错误管理和前端 GUI

每个对象负责维护适当的错误状态。如果发生致命错误，任何对象都可以请求对象管理器关闭系统。例如，当控制力矩超过其限制时可能就是这种情况。对于此类系统关闭，对象管理器遍历系统中的所有对象并调用它们的 `startShutdown()` 函数。然后，对象管理器等待所有对象完成其关闭。对象关闭的完成由 `isShutdownComplete()` 函数指示。

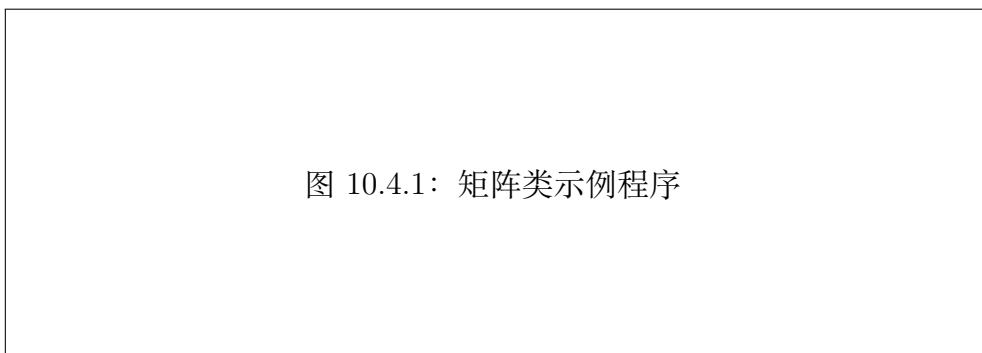


图 10.4.1: 矩阵类示例程序

图 10.15: 矩阵类示例程序

机器人平台的前端 GUI 组件是用 C++ GUI 类库 `QWidgets++` 开发的。`QWidgets++` 允许在 GUI 程序中使用面向对象技术。GUI 由三部分组成：

- **场景查看器 (Scene Viewer)**: 是默认的监控 GUI, 在每个机器人平台程序启动时自动打开。
- **控制面板**: 每个类都可以有自己的控制面板。控制面板从场景查看器打开。
- **实用程序**: 几个实用程序 (例如, 手动移动程序和示教器程序) 有 GUI。

机器人平台前端 GUI 的操作在下一节中进一步解释。

10.4 机器人平台的操作

10.4.1 场景查看器和控制面板

当执行在机器人平台内部运行的程序时, 场景查看器窗口打开。场景查看器窗口显示整个 3D 场景, 并允许用户打开一个显示当前运行对象列表的窗口 (见图 10.4.2)。为了创建 3D 场景, 场景查看器循环遍历从 `PhysicalObject` 类派生的所有对象, 并调用 `get3DModel()` 函数以获取该对象的 Open Inventor 3D 数据。然后, 场景查看器使用对象连接关系 (由 `PhysicalObject` 类的 `setConnection()` 函数指定) 在正确的位置显示 3D 对象 (例如, 显示安装在机器人操控器末端的夹爪)。此外, 场景查看器用所有对象的当前状态连续更新 3D 场景 (例如, 它使用机器人操控器的当前关节位置在正确的位置显示机器人关节)。因此, 场景查看器窗口中渲染的 3D 场景始终代表实际硬件的当前状态 (在仿真模式下, 表示硬件的仿真状态)。为了选择最佳观看位置, 用户使用鼠标在 3D 场景中导航。用户可以旋转视图、前后移动, 还可以存储/恢复期望的观看位置。此外, 用户可以隐藏场景中的对象以提高帧率。可以同时打开许多不同的场景查看器窗口, 从不同的观看位置同时查看 3D 场景。用户还可以打开对象查看器窗口以显示当前正在实例化的所有对象的列表。



图 10.4.2: 场景查看器和对象列表窗口

图 10.16: 场景查看器和对象列表窗口

每个对象还有一个单独的弹出菜单 (见图 10.4.3)。如果用户: i) 在场景查看器渲染区域中右键单击对象, 或 ii) 在对象查看器窗口中右键单击条目, 则会出现此弹出菜单。

与弹出菜单相关的选项可用于在渲染区域中隐藏对象并选择线框或实体显示。此外，弹出菜单显示在对象的特定类中定义的交互式命令。例如，夹爪对象有额外的菜单项来打开、关闭和放松夹爪。最后，用户可以从对象弹出菜单中打开控制面板。图 10.4.5 显示了 Barrett WAM 的控制面板作为示例。每个控制面板出现在一个新窗口中。

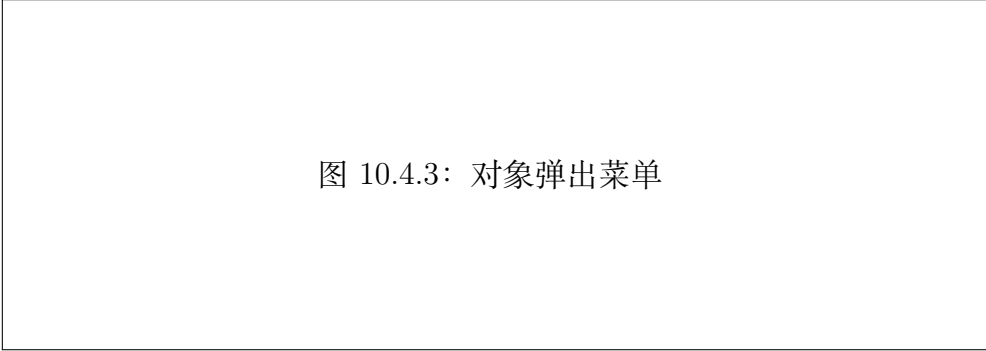


图 10.4.3: 对象弹出菜单

图 10.17: 对象弹出菜单



图 10.4.5: WAM 类的控制面板

图 10.18: WAM 类的控制面板

10.4.2 用于移动机器人的实用程序

为了促进低级算法开发，必须创建用于移动机器人的实用程序。例如，手动移动实用程序（见图 10.4.6）是一个让用户能够测试机器人操控器低级伺服控制的程序。具体来说，手动移动实用程序包含每个关节的滑块。当用户可以用鼠标移动滑块时，选定的机器人操控器关节立即跟随命令的用户输入。

示教器（见图 10.5.1）实现了一个低级零重力控制器，允许用户将机器人操控器关节物理定位在任何位置。一旦用户将机器人操控器关节移动到期望的目标位置，此位置可以存储在机器人操控器位置列表中。示教器利用机器人平台的轨迹生成器将机器人操控器移动到存储的位置。这样，用户可以使机器人操控器循环通过所有或部分存储的位置。

图 10.4.6: 手动移动实用程序

图 10.19: 手动移动实用程序

图 10.4.5: 示教器

图 10.20: 示教器

10.4.3 编写、编译、链接和启动控制程序

控制程序首先被编译，然后链接到机器人平台库。整个机器人平台（即，所有类和场景查看器）都包含在一个库中。如前所述，机器人平台可以通过添加新类轻松扩展。如果扩展特定于某个控制程序，则可以将类添加到相关控制程序的代码中。如果扩展在不同的控制程序中使用，则将此新功能添加到机器人平台库可能更方便。为了反映对预先存在的编译和链接程序的扩展，机器人平台库被构造为动态库（即，程序在启动时加载库）。因此，在用新功能扩展库之后，示教器等程序将利用新功能（例如，示教器将能够操作新操控器类型）。

图 10.4.6 显示了一个简单的拾取和放置操作示例控制程序的清单。每个机器人平台控制程序必须首先调用 `RoboticPlatform::init()`。此函数初始化机器人平台并启动场景查看器。命令行参数传递给 `RoboticPlatform::init()`，以便任何机器人平台程序都可以用某些默认命令行选项启动（见表 10.4.1）。此外，这些选项也可以通过使用表 10.4.1 第三列中列出的函数从 C++ 代码控制。调用 `RoboticPlatform::init()` 后，用户的程序创建相关机器人任务所需的所有对象（即，创建夹爪对象、Puma 560 对象和轨迹生成器对象）。如果在命令行中指定了 `-wait` 选项，则在用户程序执行 `RoboticPlatform::start()` 函数之前不会启动控制循环。此函数等待用户按下启动按钮。此功能是必需的，因为它使用户能够在启动控制循环之前启动 QMotor GUI、设置控制参数和选择记录模式。示例程序的最后部分利用轨迹生成器对象和夹爪对象将机

器人移动到工件，关闭夹爪，拾取工件，并将其放置在目标位置。

表 10.1: 机器人平台程序的默认命令行选项

选项	描述	C++ 等效
-config <file>	指定配置文件	setConfigFile()
-wait	等待用户启动	setWaitForStart()
-simulation	强制仿真模式	setSimulationMode()
-log	启用数据记录	setLogging()

Listing 10.3: 简单的拾取和放置程序

```
// 简单的拾取和放置操作
#include "RoboticPlatform.hpp"

void main(int argc, char *argv[])
{
    RoboticPlatform::init(argc, argv);

    Puma560 puma; // 创建Puma560对象
    DefaultGripper gripper; // 创建夹爪对象

    DefaultTrajectoryGenerator tragen; // 创建轨迹生成器

    RoboticPlatform::start(); // 等待用户启动

    // 移动到工件位置
    tragen.moveTo(0.3, 0.2, 0.1);
    gripper.close(); // 关闭夹爪

    // 拾取工件
    tragen.moveTo(0.3, 0.2, 0.15);

    // 移动到目标位置
    tragen.moveTo(0.5, 0.4, 0.1);
    gripper.open(); // 打开夹爪
}
```

10.5 编程示例

10.5.1 仿真与实现的比较

一个非常有趣的选项是将轨迹生成器创建的设定点转发给两个操控器。通过这种方式，可以比较两个具有相同运动学的操控器的运动，或者可以将真实操控器的行为与动态仿真进行比较。后一种应用在图 10.5.1 的程序中实现。首先，创建两个 Puma560 类的对象，并将其中一个对象配置为仿真模式。然后，两个对象都连接到同一个轨迹生成器以接收相同的设定点。

Listing 10.4: 将相同轨迹转发给两个机器人的示例

```
#include "RoboticPlatform.hpp"

void main(int argc, char *argv[])
{
    RoboticPlatform::init(argc, argv);

    // 创建两个Puma560对象
    Puma560 realRobot;
    Puma560 simulatedRobot;

    // 将第二个对象配置为仿真模式
    simulatedRobot.setSimulationMode(true);

    // 创建轨迹生成器
    DefaultTrajectoryGenerator tragen;

    // 将两个机器人连接到同一个轨迹生成器
    realRobot.setTrajectoryGenerator(&tragen);
    simulatedRobot.setTrajectoryGenerator(&tragen);

    // 移动到几个位置进行比较
    for(int i = 0; i < 5; i++) {
        tragen.moveTo(i*0.1, i*0.1, i*0.05);
    }
}
```

10.5.2 虚拟墙壁

这是如何创建自定义伺服控制的一个很好的示例。它还演示了操控器模型函数和数学库的使用。虚拟墙壁是操控器工作空间中的虚拟平面，一旦操控器移动到墙壁中就会产生反作用力。给定具有平面方程的平面（使用齐次坐标）：

$$n^T \cdot x + d_0 = 0 \quad (10.1)$$

其中 n 是平面的法向量， d_0 是距原点的距离。如果 $x_{EndEffector}$ 是当前末端执行器位置，则

$$d = n^T \cdot x_{EndEffector} + d_0 \quad (10.2)$$

使用控制律

$$\tau = J^T \cdot (-K_d \cdot \dot{x} - K_p \cdot n \cdot d) \quad (10.3)$$

创建了一个阻止机器人移动到墙壁中的关节力矩。

要实现新的伺服控制算法，需要从 `ServoControl` 类派生一个类（见图 10.5.2 中的第 (1) 行）。上述虚拟墙壁函数在计算控制输出的 `calculate()` 函数中实现。在 `main()` 函数中，像往常一样创建机器人对象。此外，还创建了虚拟墙壁伺服控制类的对象（见图 10.5.2 中的第 (2) 行）。最后，指示机器人对象使用新的伺服控制而不是默认位置控制，并启用重力补偿以允许机器人被推动（见图 10.5.2 中的第 (3) 行）。

Listing 10.5: 虚拟墙壁示例

```
#include "RoboticPlatform.hpp"

// (1) 从 ServoControl 派生新类
class VirtualWallServo : public ServoControl
{
private:
    Vector<3> n;           // 平面法向量
    double d0;           // 距原点的距离
    double Kp, Kd;       // 增益

public:
    VirtualWallServo() {
        n(0) = 1.0; n(1) = 0.0; n(2) = 0.0;
        d0 = 0.5;
    }
};
```

```
    Kp = 100.0; Kd = 10.0;
}

void calculate() {
    // 获取当前末端执行器位置
    Transform T = manipulator->getCurrentCartesianPosition();
    Vector<3> x = T.getPosition();

    // 计算到平面的距离
    double d = dot(n, x) + d0;

    // 计算雅可比矩阵
    Matrix<6,6> J = manipulator->getJacobian();

    // 计算末端执行器速度
    Vector<6> xd = J * manipulator->getJointVelocities();

    // 虚拟墙壁控制律
    Vector<3> F;
    if(d < 0) { // 墙壁内
        F = -Kd * xd.getSubVector(0,3) - Kp * n * d;
    } else {
        F = zeros(3);
    }

    // 计算关节力矩
    Vector<6> tau = J.transpose() * F;

    // 输出控制力矩
    output = tau;
}
};

void main(int argc, char *argv[])
{
    RoboticPlatform::init(argc, argv);
```

```
Puma560 puma;  
  
// (2) 创建虚拟墙壁伺服控制对象  
VirtualWallServo servo;  
  
// (3) 设置新的伺服控制并启用重力补偿  
puma.setServoControl(&servo);  
puma.enableGravityCompensation();  
  
RoboticPlatform::start();  
}
```

10.6 总结

机器人平台是一个支持实现广泛机器人应用的软件框架。与过去基于分布式架构的软件控制平台不同，机器人平台呈现了一个统一的、非分布式的面向对象架构。也就是说，基于 PC 技术和实时操作系统 QNX6，所有非实时和实时组件都集成在一个 C++ 库中。机器人平台的架构提供了设备、控制策略、轨迹生成和 GUI 组件的高效集成和可扩展性。此外，使用机器人平台实现的基于软件的控制系统价格低廉且提供高性能。机器人平台还建立在 QMotor 控制环境之上，用于数据记录、控制参数调整和实时绘图。实时数学库简化了操作并提供了一个易于使用的编程接口。内置的 GUI 组件如场景查看器和控制面板使机器人平台的操作变得容易，并为不熟悉 C++ 编程的用户提供了快速的上手时间。

参考文献

- [1] J. Lloyd, M. Parker and R. McClain, “Extending the RCCL Programming Environment to Multiple Robots and Processors”, Proc. IEEE International Conference on Robotics & Automation, 1988, pp. 465–469.
- [2] P. Corke and R. Kirkham, “The ARCL Robot Programming System”, Proc. of the International Conference on Robots for Competitive Industries, Brisbane, Australia, pp. 484–493.
- [3] J. Lloyd, M. Parker and G. Holder, “Real Time Control Under UNIX for RCCL”, Proceedings of the 3rd International Symposium on Robotics and Manufacturing (ISRAM '90).
- [4] D. B. Stewart, D. E. Schmitz, and P. K. Khosla, “CHIMERA II: A Real-Time UNIX-Compatible Multiprocessor Operating System for Sensor-based Control Applications”, Technical Report CMU-RI-TR-89-24, Robotics Institute, Carnegie Mellon University, September, 1989.
- [5] B. Stroustrup, “What is ‘Object-Oriented Programming’?”, Proc. 1st European Software Festival. February, 1991.
- [6] D. J. Miller and R. C. Lennox, “An Object-Oriented Environment for Robot System Architectures”, IEEE Control Systems Magazine, Feb. 1991, pp. 14–23.
- [7] C. Zielinski, “Object-Oriented Robot Programming”, Robotica, Vol.15, 1997, pp. 41–48.
- [8] Chetan Kapoor, “A Reusable Operational Software Architecture for Advanced Robotics”, Ph.D. thesis, University of Texas at Austin, December 1996.
- [9] C. Pelich & F. M. Wahl, “A Programming Environment for a Multiprocessor-Net Based Robot Control Unit”, Proc. 10th International Conference on High Performance Computing, Ottawa, Canada, 1996.

- [10] QSSL, Corporate Headquarters, 175 Terence Matthews Crescent, Kanata, Ontario K2M 1W8 Canada, Tel: +1 800-676-0566 or +1 613-591-0931, Fax: +1 613-591-3579, Email: info@qnx.com, <http://qnx.com>.
- [11] N. Costescu, D. M. Dawson, and M. Loffler, "QMotor 2.0—A PC Based Real-Time Multitasking Graphical Control Environment", *IEEE Control Systems Magazine*, Vol. 19 Number 3, Jun., 1999, pp. 68–76.
- [12] M. Loffler, D. Dawson, E. Zergeroglu, and N. Costescu, "Object-Oriented Techniques in Robot Manipulator Control Software Development", *Proc. of the American Control Conference*, Arlington, VA, June 2001, pp. 4520–4525.
- [13] M. Loffler, A. Bhargava, V. Chitrakaran, and D. Dawson, "Design and Implementation of the Robotic Platform", *Proc. of the IEEE Conference on Control Applications*, Mexico City, Mexico, Sept., 2001, pp. 357–362.
- [14] B. Stroustrup, "An Overview of the C++ Programming Language", *Handbook of Object Technology*, CRC Press. 1998. ISBN 0-8493-3135-8.
- [15] D. R. Musser, A. Saini, "STL Tutorial and Reference Guide", Addison-Wesley, 1996. ISBN 0-201-63398-1.
- [16] Josie Wernecke, "The Inventor Mentor", Addison-Wesley, ISBN 0-201-62495-8.
- [17] Hartman, Jed and Josie Wernecke, "The VRML 2.0 Handbook", Addison-Wesley, Reading Massachusetts, 1996.
- [18] M. Loffler, N. Costescu, and D. Dawson, "QMotor 3.0 and the QMotor Robotic Toolkit—An Advanced PC-Based Real-Time Control Platform", *IEEE Control Systems Magazine*, Vol. 22, No. 3, pp. 12–26, June, 2002.
- [19] "Direct Drive Manipulator Research and Development Package, Operations Manual", Integrated Motion Inc., Berkeley, CA, 1992.
- [20] Barrett Technologies, 139 Main St, Kendall Square, Cambridge, MA 02142, <http://www.barretttechnology.com/robot>.
- [21] N. Costescu, M. Loffler, E. Zergeroglu, and D. M. Dawson, "QRobot—A Multitasking PC Based Robot Control System", *Microcomputer Applications Journal Special Issue on Robotics*, Vol. 18, No. 1, pp. 13–22, 1999.

附录 A

机器人运动学与雅可比矩阵回顾

前言

此处我们回顾机器人学入门课程中所需的基本知识，包括机器人运动学、臂雅可比矩阵以及笛卡尔位置描述方法。本回顾内容详尽，因为具有系统理论与控制背景但尚未接触过这些内容的读者可能需要详细参考。文中给出了多个示例，这些示例将在全书的设计与仿真中使用。

A.1 A.1 基本操作器几何

本节介绍一些基本的手臂几何结构。机器人手臂或操作器由一系列通过臂连杆在空间上分隔的关节组成。关节是手臂运动发生的地方（类似于人类的手腕和肘部），而连杆具有固定结构（类似于人类的前臂）。因此，连杆在关节之间维持固定的关系。[尽管连杆可能是柔性的（即可能弯曲），但在此忽略柔性效应。]

关节可以由电机或液压执行器驱动。机器人关节有两种类型，涉及两种运动方式：

- **旋转关节 (R)**：允许绕旋转轴旋转运动，例如人类肘部。
- **移动关节 (P)**：允许伸展或伸缩运动，例如伸缩式汽车天线。

移动连杆没有拟人化的类比。操作器的关节变量是关节的可变参数。对于旋转关节，变量是角度，记为 θ ；对于移动关节，它是长度，记为 d 。

图 A.1.1 展示了一些基本的臂几何结构。

RRR 铰接臂（图 A.1.1a）类似于人类手臂；**PPP 笛卡尔臂**（图 A.1.1e）与操作器工作空间中使用的坐标紧密相关，通常使用笛卡尔坐标 (x, y, z) 描述要执行的任务。工作空间是末端执行器在机器人执行所有可能运动时扫过的总体积。

旋转关节的关节轴是旋转 θ 发生的轴。(旋转方向使用右手螺旋定则确定: 如果右手弯曲的手指指示旋转方向, 拇指指示旋转轴的方向。) 对于移动关节, 它是伸缩动作 d 发生的轴。臂关节轴的相对方向决定了其基本属性。

图 A.1.1c 所示的 RRP 操作器称为 **SCARA** (选择性柔性装配机器人手臂)。它与图 A.1.1b 所示的 RRP 球形臂结构非常不同, 因为其关节轴全部平行。另一方面, 球形臂的关节轴交于一点。

工业实例:

- RRR 臂: PUMA 和 Cincinnati-Milacron T3 735 操作器
- 球形 RRP 臂: 斯坦福操作器
- SCARA RRP 臂: AdeptOne
- RPP 臂: GMF M-100
- PPP 臂: Cincinnati-Milacron T3 龙门机器人

许多工业机器人是**串联连杆操作器**, 因为它们由一系列通过驱动关节连接的连杆组成。基座称为连杆 0, 最后一个连杆由工具或末端执行器终止。许多机器人有六个关节, 对应于在三维空间中获得末端执行器的任意位置和姿态所需的六个自由度。

如 PUMA 560 这样的臂有六个旋转关节。在这样的臂中, 关节可以分为两组, 每组三个关节。前三个关节可用于将末端执行器放置在三维工作空间内的任意位置。后三个关节可用于在该位置获得末端执行器的任意姿态。在 PUMA 560 中, 关节 4、5 和 6 的轴交于一点且相互正交。这使得末端执行器定向变得方便。后三个关节称为**腕部机构** (见示例 A.2.4)。

A.2 A.2 机器人运动学

本节回顾机器人操作器的运动学, 包括臂 A 矩阵、齐次变换、 T 矩阵、正向和逆向运动学, 以及关节空间和笛卡尔坐标。文中给出了多个说明性示例。

A.2.1 A 矩阵

给定关节变量的值, 能够指定连杆相对于彼此的位置非常重要。这通过使用操作器运动学方程来实现。

我们可以为每个连杆 i 关联一个固定在该连杆上的坐标系 (x_i, y_i, z_i) (见图 A.2.1)。一个标准且一致的方法是 Denavit-Hartenberg (D-H) 表示法。固定在连杆 0 (即操作器基座) 上的框架称为基座坐标系或惯性坐标系。

坐标系 $i-1$ 与坐标系 i 之间的关系由变换矩阵给出：

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.2.1})$$

连杆 i 的 A 矩阵中的大多数参数是固定的：

- α_i : 连杆 i 的扭转角
- a_i : 连杆 i 的长度

这些参数在臂制造商的规格表中给出。

关节参数为：

- θ_i : 关节角度
- d_i : 关节偏移量

如果关节 i 是旋转关节，则关节变量为 θ_i ， d_i 为规格表中给出的常数。如果连杆是移动关节，则 d_i 是关节变量， θ_i 为规格表中提供的常数。对于移动关节，参数 a_i 定义为零，因为连杆长度是变量，由 d_i 描述。

根据 D-H 约定，对于旋转关节，旋转 θ_i 发生在 z_{i-1} 轴周围。对于移动关节， d_i 沿 z_{i-1} 轴发生。因此，连杆坐标系被认为是附加在连杆的外端。

A 矩阵 A_i 仅是单个变量的函数，即关节变量 θ_i 或 d_i ，因为 A_i 中的所有其他参数对于特定关节是固定的。如果操作器有 n 个连杆，则关节变量向量 q 是由单个关节变量组成的 n 向量。因此， q 通常是角度 θ_i 和长度 d_i 的组合。例如，对于 RRP 臂：

$$q = [\theta_1 \quad \theta_2 \quad d_3]^T \quad (\text{A.1})$$

q 的分量记为 q_i ；即，一般关节变量 q_i 可以根据情况表示角度 θ_i 或长度 d_i 。

A.2.2 齐次变换

A 矩阵是以下形式的齐次变换矩阵：

$$A_i = \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix} \quad (\text{A.2.2})$$

其中 R_i 是旋转矩阵， p_i 是平移向量。因此，如果 ${}^i r$ 是相对于连杆 i 的坐标系描述的点，则同一点相对于连杆 $i-1$ 的框架的坐标 ${}^{i-1} r$ 由下式给出：

$$\begin{bmatrix} {}^{i-1} r \\ 1 \end{bmatrix} = A_i \begin{bmatrix} {}^i r \\ 1 \end{bmatrix} \quad (\text{A.2.3})$$

齐次变换是一个 4×4 矩阵，因此它可以描述旋转和平移；因此，在给定坐标系中描述位置的向量是 4 向量。它们的形式为：

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \quad (\text{A.2.4})$$

其中 (x, y, z) 是帧 i 中的点的坐标。

A.2.3 臂 T 矩阵

要获得相对于基座（即连杆 0）框架的点的坐标，我们可以使用矩阵：

$$T_i = A_1 A_2 \cdots A_i \quad (\text{A.2.7})$$

然后，给定相对于连杆 i 的附加框架表达的点的坐标 ${}^i r$ ，同一点在基座框架中的坐标由下式给出：

$$\begin{bmatrix} {}^0 r \\ 1 \end{bmatrix} = T_i \begin{bmatrix} {}^i r \\ 1 \end{bmatrix} \quad (\text{A.2.8})$$

我们称 T_i 为运动学变换链。我们定义臂 T 矩阵为：

$$T = T_n = A_1 A_2 \cdots A_n \quad (\text{A.2.9})$$

其中 n 是操作器中的连杆数。然后，如果 ${}^n r$ 是相对于最后一个连杆的坐标，则该点的基座坐标为：

$$\begin{bmatrix} {}^0 r \\ 1 \end{bmatrix} = T \begin{bmatrix} {}^n r \\ 1 \end{bmatrix} \quad (\text{A.2.10})$$

这是一个重要的关系，因为 ${}^n r$ ——第 n 帧中物体的坐标——可以表示物体相对于工具或末端执行器的位置。因此，这对于指定要执行的任务非常重要。

A.2.4 正向运动学

末端执行器相对于操作器基座框架的位置和姿态通过评估臂 T 矩阵给出。传统上，这个齐次变换表示为：

$$T = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \quad (\text{A.2.11})$$

因此，末端执行器参考框架轴的方向由旋转矩阵 $R = [n \ o \ a]$ 相对于基座坐标描述，末端执行器框架的原点在基座坐标中的位置为 p 。

3 向量 n 、 o 、 a 和 p 定义如下：

- a (approach 向量)：末端执行器的接近向量
- o (orientation 向量)：指定手部方向的方向向量，从指尖到指尖

- n (normal 向量): 选用来使用 $n = o \times a$ 完成右手坐标系定义的向量
- p : 指定 (n, o, a) 框架原点相对于基座框架的位置向量

因此, (n, o, a) 是附加在末端执行器上的 (x, y, z) 笛卡尔坐标系相对于基座的坐标。

关节空间与笛卡尔空间:

- q 是末端执行器位置和姿态的**关节空间描述**
- (n, o, a, p) 是**笛卡尔或任务空间描述**

机器人臂运动学问题如下: 给定关节变量 q , 找到操作器末端的笛卡尔位置和姿态。因此, 运动学问题涉及将给定关节变量转换为基座坐标中表达的末端执行器笛卡尔位置和姿态。

注: 读者可以通过[交互式可视化工具](#)探索正运动学与逆运动学的关系, 观察关节角度与末端位置的映射, 以及多解性问题。

A.3 A.3 操作器雅可比矩阵

给定从关节变量 $q(t) \in \mathbb{R}^n$ 到 $y \in \mathbb{R}^p$ 的一般非线性变换:

$$y = h(q) \quad (\text{A.3.1})$$

我们定义与 $h(q)$ 相关的雅可比矩阵为:

$$J(q) \equiv \frac{\partial h(q)}{\partial q} \quad (\text{A.3.2})$$

正如我们刚刚看到的, 雅可比矩阵在反馈线性化中很有用, 因此在机器人操作器控制中也很重要。它还是我们在坐标系之间转换速度、加速度和力的手段。

A.3.1 速度和加速度的变换

由于:

$$\dot{y} = \frac{\partial h}{\partial q} \dot{q} = J(q) \dot{q} \quad (\text{A.3.3})$$

雅可比矩阵允许我们将速度从关节空间转换到“ y 空间”。让我们讨论 $y(t)$ 是笛卡尔速度的特殊情况。则 $J(q)$ 称为**操作器雅可比矩阵**。

通常将广义笛卡尔速度定义为:

$$\dot{y} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{A.3.4})$$

其中 $v = [v_x v_y v_z]^T$ 是线速度, $\omega = [\omega_x \omega_y \omega_z]^T$ 是角速度。例如, ω_x 表示绕 x 轴的角速度。因此, y 有六个分量, 臂雅可比 J 是一个 $6 \times n$ 矩阵, n 是操作器中的关节数。如果 $n = 6$, 雅可比是方阵。

使用 (A.3.3), 我们可以获得微分运动变换的表达式。令 $dq = [dq_1 \cdots dq_n]^T$ 为关节空间中的微分运动, dq_i ——如果关节 i 是旋转关节, 则为小旋转; 如果关节 i 是移动关节, 则为小线位移。令:

$$dy = \begin{bmatrix} dx & dy & dz & \delta x & \delta y & \delta z \end{bmatrix}^T \quad (\text{A.3.5})$$

在同一笛卡尔坐标中描述相同的微分运动, $[dx \ dy \ dz]^T$ 是微分线运动, $[\delta x \ \delta y \ \delta z]^T$ 表示微分旋转。

根据 (A.3.3), 其中 $\dot{y} = dy/dt$ 和 $\dot{q} = dq/dt$, 我们看到:

$$\dot{y} = J \dot{q} \quad (\text{A.3.6})$$

其中 J 是关联关节空间和笛卡尔空间的雅可比矩阵。

加速度的变换通过对 (A.3.3) 微分得到:

$$\ddot{y} = \dot{J} \dot{q} + J \ddot{q} \quad (\text{A.3.7})$$

A.3.2 力的变换

为了发现静态力在坐标系之间的变换, 考虑以下情况。关节空间中施加的广义力/扭矩产生的微分运动 dq 所产生的虚功为:

$$\delta W = \tau^T dq \quad (\text{A.3.8})$$

其中 τ 是臂控制力/扭矩的 n 向量, $dq = [dq_1 \cdots dq_n]^T$ 是关节变量的微分变化。如果在另一个坐标系中力的描述是 F , 位置的描述是 y , 我们也必须有:

$$\delta W = F^T dy \quad (\text{A.3.9})$$

现在考虑 (A.3.6), 我们可以写成:

$$\tau^T dq = F^T J dq = (J^T F)^T dq \quad (\text{A.2})$$

因此, 从力到扭矩的变换由下式给出:

$$\tau = J^T(q) F \quad (\text{A.3.10})$$

当 $y(t)$ 是笛卡尔位置时, 我们定义笛卡尔广义力为 6 向量:

$$F = \begin{bmatrix} f_c \\ \tau_c \end{bmatrix} \quad (\text{A.3.11})$$

其中 $f_c = [f_x f_y f_z]^T$ 是笛卡尔力 3 向量, $\tau_c = [\tau_x \tau_y \tau_z]^T$ 是表示笛卡尔扭矩的 3 向量。例如, τ_x 表示绕 x 轴施加的扭矩。

如果臂有六个连杆且雅可比是非奇异的, 则从广义扭矩到广义力的变换由下式给出:

$$F = J^{-T}(q)\tau \quad (\text{A.3.12})$$

雅可比的奇异性通常发生在操作器工作空间的极端位置。

注: 读者可以通过[交互式可视化工具](#)观察雅可比行列式随位形的变化, 理解奇异位形的物理意义及其对运动控制的影响。

A.3.3 笛卡尔位置描述

现在有必要面对机器人学中一个令人困惑的问题。考虑方程 (A.3.2)。不幸的是, 当讨论从关节空间到笛卡尔空间的变换 $h(q)$ 时, 这个方程只能被解释为符号的便利, 而不是严格的数学公式。原因虽然广义笛卡尔速度 (A.3.4) 和加速度以及广义笛卡尔力 (A.3.11) 是合法的 6 向量, 但在方便地指定广义笛卡尔位置 $y(t)$ 方面存在问题。

将广义笛卡尔位置表示为 (n, o, a, p) : 在我们的背景下, 必须在基座坐标中指定末端执行器框架原点的位置及其姿态。使用 3 向量 $p(t) = [p_x p_y p_z]^T$ 在基座坐标 (x, y, z) 中指定末端执行器框架的原点很容易。然而, 指定姿态却不那么容易。这是因为需要多于三个独立变量才能在相对于另一个框架唯一地指定一个框架的姿态。

应该提到的是, 欧拉角和滚转-俯仰-偏航等约定只涉及三个变量。然而, 它们不能用于唯一地指定一个框架相对于另一个框架的绝对姿态, 而只能用于姿态的相对变化。

在我们的工作中, 我们通过使用 (n, o, a, p) 方法 (见 A.2 节) 在基座坐标中指定末端执行器的笛卡尔位置 $y(t)$ 。在那里, 我们将广义笛卡尔位置定义为:

$$y(t) \equiv T(t) \quad (\text{A.3.13})$$

其中 $T(t)$ 是臂 T 矩阵, (n, o, a) 是描述末端执行器姿态所需的向量, $p = [p_x p_y p_z]^T$ 是指定末端执行器框架原点的笛卡尔描述位置部分。

A.3.4 计算臂雅可比

最后, 我们可以展示如何给定关节变量 q_i 计算臂雅可比。程序如下:

给定 q_i , 计算 A.2 节中定义的矩阵 T_i :

$$T_i = A_1 A_2 \cdots A_i = \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix} \quad (\text{A.3.29})$$

其对应的旋转矩阵将记为:

$$R_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix} \quad (\text{A.3.30})$$

定义 $T_0 = I$, $R_0 = I$, 以及臂 T 矩阵:

$$T = T_n = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \quad (\text{A.3.31})$$

向量 z_i 表示基座坐标中框架 i 的 z 轴。向量 p 表示基座坐标中连杆框架 n (末端执行器框架) 原点的位置。雅可比使用向量 p 和 z_i 如下计算。

广义笛卡尔速度是 $\dot{y} = [v^T \omega^T]^T$ 。因此, 我们可以将雅可比矩阵分成线性部分和定向部分, 写成:

$$J(q) = \begin{bmatrix} J_p(q) \\ J_o(q) \end{bmatrix} \quad (\text{A.3.32})$$

其中 $J_p(q)$ 是 $J(q)$ 的前三行, $J_o(q)$ 是其最后三行。

首先, 考虑线性位置雅可比 $J_p(q)$ 的计算。鉴于广义笛卡尔位置 y 的线性部分只是 p , 我们可以写成:

$$v = \dot{p} = \frac{\partial p}{\partial q} \dot{q} = J_p(q) \dot{q} \quad (\text{A.3.33})$$

因此, $J_p(q)$ 由下式给出:

$$J_p(q) = \frac{\partial p}{\partial q} = \begin{bmatrix} \frac{\partial p}{\partial q_1} & \cdots & \frac{\partial p}{\partial q_n} \end{bmatrix} \quad (\text{A.3.34})$$

现在转向雅可比的定向部分 $J_o(q)$ 。正如线速度一样, 只要在同一坐标框架中表示, 就可以添加角速度。让我们将臂中连杆的单个角速度相加, 以获得末端执行器的角速度。移动关节对末端执行器的角速度没有贡献。

对于旋转关节, 关节旋转 $q_i = \theta_i$ 发生在关节轴 z_{i-1} 周围 (见 A.2 节, 特别是图 A.2.1)。因此, 关节变量 i 的角速度由 $\omega_i = z_{i-1} \dot{q}_i$ 给出。要添加所有连杆的效果, 有必要将 z_{i-1} 表示在公共框架中; 我们选择基座坐标。然而, R_{i-1} 的最后一列正是基座坐标中的 z_{i-1} 。因此, 我们可以写成:

$$J_o = \begin{bmatrix} z_0 \sigma_1 & z_1 \sigma_2 & \cdots & z_{n-1} \sigma_n \end{bmatrix} \quad (\text{A.3.35})$$

其中 $z_0 = [0 \ 0 \ 1]^T$, 选择参数 σ_i 如果 q_i 是移动关节则为 0, 如果是旋转关节则为 1。因此:

$$\sigma_i = \begin{cases} 0, & \text{如果关节 } i \text{ 是移动关节} \\ 1, & \text{如果关节 } i \text{ 是旋转关节} \end{cases} \quad (\text{A.3.36})$$

完整的雅可比现在通过将 $J_p(q)$ 堆叠在 $J_o(q)$ 上给出。现在应该清楚, 臂雅可比必须通过对关节向量 q 使用大量计算来找到。因此, 每当在机器人臂控制方案中需要雅可比时, 应该使用计算机子程序计算。

重新做示例 A.3.1 和 A.3.2 以确定完整的雅可比, 包括角部分, 是一个很好的练习 (见习题)。

参考文献

[Craig 1989] Craig, J.J., Introduction to Robotics. Reading, MA: Addison-Wesley, 1989.

[Ickes 1970] Ickes, B.P., "A new method for performing digital control system altitude computations using quaternions," AIAA J., vol. 8, no. 1, pp. 13–17, Jan. 1970.

[IMSL] IMSL, Library Contents Document, 8th ed. Houston, TX: International Mathematical and Statistical Libraries.

[MATMAN 1986] MATMAN, Symbolic Matrix Manipulation, M.R. Driels, Pembroke, MA: Kern International, 1986.

[Paul 1981] Paul, R.P., Robot Manipulators, Cambridge, MA: MIT Press, 1981.

[Schilling 1990] Schilling, R.J., Fundamentals of Robotics, Englewood Cliffs, NJ: Prentice Hall, 1990.

[Sheppard 1978] Sheppard, S.W., "Quaternion from rotation matrix," Eng. Notes, pp. 223–224, May/June 1978.

[Spong and Vidyasagar 1989] Spong, M.W., and M. Vidyasagar, Robot Dynamics and Control. New York: Wiley, 1989.

[Stevens and Lewis 1992] Stevens, B.L., and F.L. Lewis, Aircraft Modeling, Dynamics, and Control. New York: Wiley, 1992.

[Yuan 1988] Yuan, J.S.-C., "Closed-loop manipulator control using quaternion feedback," IEEE J. Robot. Autom., vol. 4, no. 4, pp. 434–440, Aug. 1988.

附录 B

控制器仿真软件

附录 B

控制器仿真软件

理解控制系统设计和性能的直观感受的一种极好方法是进行计算机仿真。从概念上讲，从仿真到实际实现只是短短的一步，因为当今数字信号处理器上使用的子程序与仿真中使用的子程序非常相似。本附录包含本书中用于机器人控制器仿真的软件。

有一些优秀的软件包可用于系统设计和仿真，我们应该了解并使用它们。例如 MATLAB、MATRIX、Program CC、SIMNON 等。然而，在学习阶段使用自己的软件是非常有益的，特别是当涉及到数字控制时，有时不清楚这些软件包中具体发生了什么。

对于连续系统的时间响应仿真，Runge-Kutta 积分器工作得很好。在图 B.1.1 中展示了程序 TRESP，它使用四阶 Runge-Kutta 子程序实现第 3.3 节讨论的仿真过程。它以状态变量形式积分线性或非线性系统：

$$\dot{x} = f(x, u, t) \quad (\text{B.1})$$

在第 2.4 节中，我们展示了如何将机器人动力学转换为这种形式。注意，MATLAB 的函数 ode23 和 ode45 是自适应步长的 Runge-Kutta 积分器，它们需要相同形式的动力学方程。

TRESP 需要一个子程序 F(time, x, xp)，它从当前状态 $x(t)$ 和控制输入 $u(t)$ 计算 \dot{x} （记为 xp，或“x prime”）。控制 $u(t)$ 和任何形式的输出

$$y = h(x, u, t) \quad (\text{B.2})$$

被放入 COMMON 存储区（例如， $u(t)$ 可以在 F(time, x, xp) 外部计算，而 $y(t)$ 在 TRESP 中需要用于绘图）。在本书各章节的示例中给出了使用 TRESP 的示例。

COMMON 中的参数数组 PAR() 使得使用不同参数值 (如 PD 增益) 进行连续运行变得容易。如果需要, 可以向连续动力学注入时间延迟 (例如使用环形缓冲区)。

重要的是要认识到以下几点。为了将 $x(kT_R)$ 更新到 $x((k+1)T_R)$, Runge-Kutta 积分器在每个 Runge-Kutta 积分周期 T_R 内调用子程序 F(time, x, xp) 四次。在这四次调用期间, 控制输入应保持恒定值 $u(kT_R)$ 。使用子程序 SYSINP 计算 $u(kT_R)$ 可以实现这一点。

对于数字控制仿真, TRESP 需要子程序 DIG(IK, T, x), 它包含离散控制器方程; 它在每个采样周期 T 中调用一次。时间 T_R 应选择为 T 的整数除数。每个采样周期内通常需要 5 到 10 个 Runge-Kutta 周期。

该程序还允许数字滤波 (例如, 用于从关节位置编码器测量重建速度估计)。注意, 对于数字控制目的, 在调用 Runge-Kutta 程序之前调用子程序 DIG, 而对于数字滤波, 在调用 Runge-Kutta 之后调用 DIG。

对于某些系统, 图中的 Runge-Kutta 积分器可能无法工作; 此时可以使用自适应步长的 Runge-Kutta 程序 (例如 Runge-Kutta-Fehlberg) [Press et al. 1986]。(注: 此处给出的程序适用于本书中的所有示例。)

参考文献

[Press et al. 1986] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recipes*. New York: Cambridge University Press, 1986.

附录 C

常见机器人手臂的动力学

附录 C

常见机器人机械臂的动力学

在本附录中，我们给出一些常见机器人机械臂的动力学。假设机器人动力学由下式给出：

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (\text{C.1.1})$$

其中矩阵 $M(q)$ 是对称且正定的，其元素为 $m_{ij}(q)$ ，即

$$M(q) = [m_{ij}(q)], \quad i, j = 1, \dots, n \quad (\text{C.1})$$

而 $N(q, \dot{q})$ 是一个 $n \times 1$ 向量，其元素为 n_i ，即

$$N(q, \dot{q}) = [n_i], \quad i = 1, \dots, n \quad (\text{C.2})$$

特别要注意，在 n_i 的表达式中通过重力常数 $g = 9.8$ 米/秒² 来识别重力项。我们还将采用以下符号表示：

- 连杆 i 的长度为 L_i ，单位为米
- 连杆 i 的质量为 m_i ，单位为千克
- 连杆 i 关于轴 u 的质量惯性矩为 I_{uui} ，单位为 $\text{kg}\cdot\text{m}^2$
- $S_i = \sin q_i$, $C_i = \cos q_i$
- $S_{ij} = \sin(q_i + q_j)$, $C_{ij} = \cos(q_i + q_j)$

- $S_{ijk} = \sin(q_i + q_j + q_k)$, $C_{ijk} = \cos(q_i + q_j + q_k)$
- $SS_i = \sin^2 q_i$, $CC_i = \cos^2 q_i$, $CS_i = \cos q_i \sin q_i$
- $SS_{ij} = \sin^2(q_i + q_j)$

C.1 SCARA 机械臂

我们考虑的第一个机器人是通用 SCARA 构型机器人，如图 C.1.1 所示。这些方程适用于 AdeptOne 和 AdeptTwo 机器人。动力学包括前四个自由度，以符号形式给出如下：

$$m_{11} = m_1 L_1^2 + I_{zz1} + m_2(L_1^2 + L_2^2 + 2L_1 L_2 C_2) + I_{zz2} + m_3(L_1^2 + L_2^2 + 2L_1 L_2 C_2) \quad (\text{C.3})$$

$$m_{12} = m_2(L_2^2 + L_1 L_2 C_2) + I_{zz2} + m_3(L_2^2 + L_1 L_2 C_2) \quad (\text{C.4})$$

$$m_{13} = 0 \quad (\text{C.5})$$

$$m_{14} = 0 \quad (\text{C.6})$$

$$m_{22} = m_2 L_2^2 + I_{zz2} + m_3 L_2^2 \quad (\text{C.7})$$

$$m_{23} = 0 \quad (\text{C.8})$$

$$m_{24} = 0 \quad (\text{C.9})$$

$$m_{33} = m_3 \quad (\text{C.10})$$

$$m_{34} = 0 \quad (\text{C.11})$$

$$m_{44} = I_{zz4} \quad (\text{C.12})$$

$$n_1 = -m_2 L_1 L_2 S_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) - m_3 L_1 L_2 S_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \quad (\text{C.13})$$

$$n_2 = m_2 L_1 L_2 S_2 \dot{q}_1^2 + m_3 L_1 L_2 S_2 \dot{q}_1^2 \quad (\text{C.14})$$

$$n_3 = -m_3 g \quad (\text{C.15})$$

$$n_4 = 0 \quad (\text{C.16})$$

图 C.1.1: SCARA 机械臂

图 C.1: SCARA 机械臂示意图

C.2 Stanford 机械臂

如图 C.2.1 所示的 Stanford 机械臂具有以下动力学方程 [Bejczy 1974], [Paul 1981]:

$$m_{11} = I_{xx2}S_2^2 + I_{yy2}C_2^2 + m_2d_2^2C_2^2 + m_3q_3^2S_2^2 + I_{xx3}S_2^2 + I_{yy3}C_2^2 \quad (C.17)$$

$$m_{12} = -m_3q_3C_2 \quad (C.18)$$

$$m_{13} = -m_3d_2S_2 \quad (C.19)$$

$$m_{22} = m_3q_3^2 + I_{zz2} + I_{zz3} \quad (C.20)$$

$$m_{23} = 0 \quad (C.21)$$

$$m_{33} = m_3 \quad (C.22)$$

$$n_1 = (I_{yy2} - I_{xx2} + m_3q_3^2 + I_{yy3} - I_{xx3})CS_2\dot{q}_1\dot{q}_2 + m_3q_3^2S_2C_2\dot{q}_1\dot{q}_3 \quad (C.23)$$

$$+ (I_{yy3} - I_{xx3} + m_3q_3^2)CS_2\dot{q}_1\dot{q}_2 \quad (C.24)$$

$$n_2 = -(I_{yy2} - I_{xx2} + m_3q_3^2 + I_{yy3} - I_{xx3})CS_2\dot{q}_1^2/2 - m_3q_3^2S_2C_2\dot{q}_1^2/2 \quad (C.25)$$

$$+ m_3q_3S_2\dot{q}_2\dot{q}_3 - m_3q_3C_2^2\dot{q}_1^2 - m_3q_3\dot{q}_2^2/2 + m_3gC_2 \quad (C.26)$$

$$n_3 = -m_3q_3S_2^2\dot{q}_1^2 - m_3d_2C_2\dot{q}_2^2 + m_3q_3\dot{q}_2^2 - m_3gS_2 \quad (C.27)$$

图 C.2.1: Stanford 机械臂

图 C.2: Stanford 机械臂示意图

C.3 PUMA 560 机械臂

PUMA 560 如图 C.3.1 所示。对于这种特定的结构，可以进行许多简化以获得以下动力学方程，这些方程出现在 [Armstrong et al. 1986] 中。

$$m_{11} = I_{xx1} + I_{xx2}S_2^2 + I_{yy2}C_2^2 + m_2d_2^2C_2^2 + m_3(L_2S_2 + d_3C_2)^2 \quad (C.28)$$

$$+ I_{xx3}S_{23}^2 + I_{yy3}C_{23}^2 + m_3d_3^2S_{23}^2 + m_4(L_2S_2 + L_3S_{23})^2 \quad (C.29)$$

$$m_{12} = m_3d_2d_3S_3 + m_4d_2L_3S_3 \quad (C.30)$$

$$m_{13} = -m_3d_3^2 - m_3L_2d_3C_3 - m_4L_2L_3C_3 - I_{zz3} \quad (C.31)$$

$$m_{22} = m_2d_2^2 + m_3d_3^2 + m_3L_2^2 + m_4L_2^2 + m_4L_3^2 + 2m_3L_2d_3C_3 + 2m_4L_2L_3C_3 + I_{zz2} + I_{zz3} \quad (C.32)$$

$$m_{23} = m_3d_3^2 + m_4L_3^2 + m_3L_2d_3C_3 + m_4L_2L_3C_3 + I_{zz3} \quad (C.33)$$

$$m_{33} = m_3d_3^2 + m_4L_3^2 + I_{zz3} \quad (C.34)$$

$$n_1 = [(I_{yy2} - I_{xx2})C_2S_2 + m_2d_2^2C_2S_2 + m_3(L_2S_2 + d_3C_2)(L_2C_2 - d_3S_2)] \quad (C.35)$$

$$+ (I_{yy3} - I_{xx3})C_{23}S_{23} + m_3d_3^2C_{23}S_{23} + m_4(L_2S_2 + L_3S_{23})(L_2C_2 + L_3C_{23}](2\dot{q}_1\dot{q}_2) \quad (C.36)$$

$$+ [(I_{yy3} - I_{xx3})C_{23}S_{23} + m_3d_3^2C_{23}S_{23} - m_3(L_2S_2 + d_3C_2)d_3S_{23}] \quad (C.37)$$

$$+ m_4(L_2S_2 + L_3S_{23})L_3C_{23}](2\dot{q}_1\dot{q}_3) \quad (C.38)$$

$$n_2 = [(I_{xx2} - I_{yy2})C_2S_2 - m_2d_2^2C_2S_2 - m_3(L_2S_2 + d_3C_2)(L_2C_2 - d_3S_2)] \quad (C.39)$$

$$+ (I_{xx3} - I_{yy3})C_{23}S_{23} - m_3d_3^2C_{23}S_{23} - m_4(L_2S_2 + L_3S_{23})(L_2C_2 + L_3C_{23}]\dot{q}_1^2 \quad (C.40)$$

$$+ [-m_3L_2d_3S_3 - m_4L_2L_3S_3](2\dot{q}_2\dot{q}_3 + \dot{q}_3^2) \quad (C.41)$$

$$+ [m_3d_3C_{23} + m_4L_3C_{23}]g \quad (C.42)$$

$$n_3 = [(I_{xx3} - I_{yy3})C_{23}S_{23} - m_3d_3^2C_{23}S_{23} + m_3(L_2S_2 + d_3C_2)d_3S_{23}] \quad (C.43)$$

$$- m_4(L_2S_2 + L_3S_{23})L_3C_{23}]\dot{q}_1^2 \quad (C.44)$$

$$+ [m_3L_2d_3S_3 + m_4L_2L_3S_3]\dot{q}_2^2 \quad (C.45)$$

$$+ [m_3d_3S_{23} + m_4L_3S_{23}]g \quad (C.46)$$

图 C.3.1: PUMA 560 机械臂

图 C.3: PUMA 560 机械臂示意图

参考文献

[Armstrong et al. 1986] Armstrong, B., O. Khatib, and J. Burdick, “The explicit dynamic model and inertial parameters of the PUMA 560 arm,” *Proc. 1986 IEEE Conf. Robot. Autom.*, pp. 510–518, San Francisco, Apr. 7–10, 1986.

[Bejczy 1974] Bejczy, A.K., “Robot arm dynamics and control,” NASA-JPL Technical Memorandum 33–669, 1974.

[Paul 1981] Paul, R.P., *Robot Manipulators: Mathematics, Programming and Control*. Cambridge, MA: MIT Press, 1981.