

申请上海交通大学博士学位论文

基于轨迹库的仿人机器人站立平衡和行走控制

“本文受国家自然科学基金重点项目（60935001）资助”

系 别: 自动化系

学 科: 控制理论与控制工程

作 者: 刘成刚

导 师: 苏剑波 教授

上海交通大学
2011年5月

A Dissertation Submitted to Shanghai Jiao Tong University for the
Degree of Doctor

Control of Standing Balance and Walking of a Humanoid Robot Using a Trajectory Library

CHENGGANG LIU

Supervisor:

Prof. JIANBO SU

DEPARTMENT OF AUTOMATION, SCHOOL OF ELECTRONIC , INFORMATION
AND ELECTRICAL ENGINEERING
SHANGHAI JIAO TONG UNIVERSITY
SHANGHAI, P.R.CHINA

May, 2011

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：_____

日期：_____年____月____日

上海交通大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密 ，在 _____ 年解密后适用本授权书。

本学位论文属于

不保密 。

(请在以上方框内打“√”)

学位论文作者签名: _____

指导教师签名: _____

日 期: _____ 年 _____ 月 _____ 日

日 期: _____ 年 _____ 月 _____ 日

基于轨迹库的仿人机器人站立平衡和行走控制

摘要

仿人机器人可以适应人类的生活和工作环境，代替人类完成各种任务，在很多方面扩展人类的能力，具有广阔的应用前景。仿人机器人运动控制水平的高低是决定其能否走向实用的重要因素。虽然目前在仿人机器人运动控制方面取得了大量可喜的成果，但其在控制的效率、鲁棒性等方面与人类相比仍有很大差距。仿人机器人是目前结构最复杂的机器人之一，因此凭借设计者经验为其设计各种参数化运动控制器不仅非常繁琐，而且很难保证最终得到的控制器性能。最优控制提供了采用优化技术设计控制器的系统方法，但是目前采用最优控制实现的仿人机器人运动控制方法，存在着模型过于简单和计算时间过长等问题。因此，研究新的基于优化的仿人机器人运动控制方法具有重要意义。

对仿人机器人而言，双足行走是其区别于其他类型机器人的主要特征，扰动下的站立平衡控制是运动控制研究的基本问题，因此本文重点研究了基于优化的仿人机器人站立平衡控制和行走控制问题，着眼于提高控制的效率和鲁棒性。所取得的成果主要体现在以下几个方面：

1. 从轨迹优化和构造控制律两个方面，研究了基于轨迹库的控制方法。在轨迹优化方面，研究了多种轨迹优化方法。为了克服单一轨迹优化算法存在的算法收敛依赖于初始轨迹或得到的轨迹违反动力学约束等缺点，提出了一种混合轨迹优化方法。该方法在生成轨迹方面更加鲁棒，能产生更加精确的轨迹，而且能够在生成轨迹的同时得到控制律局部近似模型。另外，研究了轨迹优化中存在的约束处理、振铃现象、稳态值函数近似和局部最优等问题，提出了解决的办法。在构造控制律方面，总结了采用轨迹库实现控制的方法，包括轨迹跟踪法、在线优化法和函数近似法等。

2. 研究了基于优化的仿人机器人站立平衡控制方法。针对实际的机器人控制问题，研究了利用非线性最小二乘方法进行模型物理参数辨识的方法和基于扩展卡尔曼滤波器的状态在线估计方法。针对目前大部分采用参数化控制器实现的方法存在性能局限、采用最优控制实现的方法计算量大且只实现了瞬间扰动下的站立平衡等问题，提出了基于轨迹库的站立平衡控制方法。该方法不仅能够比较快速地得到控制器，而且能够保证最终得到的控制器相对特定优化指

标的性能。在生成轨迹库时，针对如何减少轨迹库中轨迹数目这一关键性问题，提出了初始条件自适应网格上基于控制器预测性能的轨迹库生成方法，在减少轨迹数目的同时，满足控制器的性能要求。以往的研究主要实现了瞬间扰动下的站立平衡控制，而本文同时研究并实现了持续且时变的扰动下的站立平衡控制。与基于线性二次型调节器（LQR）的增益调度控制器进行了对比，仿真和实验结果说明了方法的有效性和性能优势。由于仿人机器人由多个关节和身体部分构成，相同外力作用下站立平衡控制中期望状态的选择并不唯一。为进一步提高控制器的鲁棒性，本文研究了期望状态的选择问题，并提出了确定期望状态的方法。仿真和实验结果说明了方法的有效性。

3. 研究了基于优化的仿人机器人行走控制方法。首先，把仿人机器人的行走控制问题归结为动力学不连续系统周期性任务的最优控制问题，提出一种邻域最优控制方法。采用参数化轨迹优化方法求解周期性稳态轨迹，在推导动力学不连续处值函数的传递关系的基础上，采用微分动态规划得到更加精确的周期性稳态轨迹和其采样状态上的控制律局部近似模型。以此为基础，研究了只考虑单腿支撑的平面七连杆双足行走控制问题。针对现有方法存在的不足，提出了基于轨迹库的行走控制方法。仿真结果验证了方法的鲁棒性和有效性，并与基于轨迹跟踪的行走控制进行了对比，显示了本文方法在鲁棒性、性能方面的优越性。最后，研究了更具一般性的行走控制（考虑双腿支撑阶段的三维七连杆双足行走控制），给出了稳态轨迹优化的初步结果。

关键词： 轨迹优化, 轨迹库, 仿人机器人, 站立平衡, 双足行走

Control of Standing Balance and Walking of a Humanoid Robot Using a Trajectory Library

ABSTRACT

Humanoid robots can adapt themselves well to human living and working environment, assist us in our daily life, and extend our abilities in many aspects. However, the level of motion control of a humanoid robot will determine its success or failure in its applications. Although there are significant successes in this field, compared with humans, these results are still limited in terms of energetic efficiency, control robustness, and so on. A humanoid robot is one of the most complicated robots. It is not only awkward endeavor to manually design parametric controllers for various motion tasks, but also hard to guarantee the resultant controller's performance. Optimal control provides a systematic way to design controllers with optimization techniques. But only applications using very simple models can be found and most of them suffer from the computation time problem. Therefore, it is significant and inevitable to study new optimization-based motion control methods of humanoid robots.

For a humanoid robot, biped walking is an important symbol characterizing it from other kinds of robots, and standing balance control in the presence of perturbations is a fundamental problem. Therefore, this dissertation will focus on optimization-based standing balance control and walking control, with an eye to the efficiency and robustness of the resultant controller. The following contributions are obtained.

1. Control using a trajectory library is studied first from two aspects: trajectory optimization and control law construction. For trajectory optimization, several trajectory optimization algorithms are studied. To overcome the shortages of one single algorithm, such as the algorithm convergence's depending on the initial guess or the resultant trajectory's violating dynamics constraints, a hybrid trajectory optimization approach is proposed. It is more robust in term

of finding solutions and can generate more accurate trajectories and local models of control laws at the same time. Problems with trajectory optimization, such as handling constraints, ringing phenomenon, steady-state value function approximation, and local optima, are addressed and solutions to them are given. For control law construction, methods based on trajectory tracking, online optimization, and function approximation, are studied.

2. Optimization-based standing balance control is studied. For the control of a real robot, physical parameter identification using nonlinear least square method and online state estimation based on the extended Kalman filter are studied. For standing balance control, most of parametric controllers have performance limitations, and most controllers generated by optimal control suffer from the computation time problem and can only handle instantaneous pushes. Therefore, standing balance control using a trajectory library is proposed. It can quickly generate a control law of a good performance for a specified optimization criterion. For the important problem of reducing the number of trajectories in the library, it is proposed to generate a trajectory library on an adaptive grid of initial conditions based on the resultant controller's predicted performance. It keeps the resultant library a reasonable size and also satisfies performance requirements. Most previous work assumes that perturbations are impulsive, but the proposed controller can also handle continuous perturbations that change with time. We compare the proposed method with a Linear Quadratic Regulator (LQR) gain scheduling controller. Simulation and experiment results demonstrate the effectiveness and performance of the proposed method. Since a humanoid robot has many joints and body parts, the desired state for standing balance control is not unique in the presence of continuous external forces. To further improve the robustness of standing balance control, a method to determine the desired state is proposed. Simulation and experiment results demonstrate the effectiveness of the proposed method.

3. Optimization-based biped walking control is studied. First, the biped walking control problem is formulated as an optimal control problem of a periodic task for systems with discontinuous dynamics. Therefore, a neighboring optimal control method is proposed. A parametric trajectory optimization method is proposed to find the steady-state trajectory in a periodic task. The propaga-

tion rule of the value function at dynamics discontinuities is derived, and then DDP (Differential Dynamic Programming) can be used to generate more accurate steady-state trajectory and local models of the control law. Based on this result, we study the control of a planar seven-link biped walking without double-support phase. To overcome the problems of current control methods, walking control using a trajectory library is proposed. Simulation results demonstrate the effectiveness and robustness of the proposed method. The proposed method is also compared with a trajectory tracking controller. Comparison results demonstrate the advantages of the proposed method in terms of robustness and performance. Finally, the control of a 3-D seven-link biped walking with double-support phase is studied and the steady-state trajectory optimization result is given.

KEY WORDS: trajectory optimization, trajectory library, humanoid robot, standing balance, biped walking

目 录

摘要	i
ABSTRACT	iii
目录	vii
第一章 绪论	1
1.1 仿人机器人研究现状	1
1.2 站立平衡控制研究动态	2
1.2.1 参数化站立平衡控制方法	3
1.2.2 基于优化的站立平衡控制方法	4
1.3 双足行走控制研究动态	5
1.3.1 典型的双足行走控制方法	5
1.3.2 基于优化的双足行走控制方法	7
1.3.3 被动行走	8
1.4 基于轨迹和局部模型的控制方法研究现状	9
1.5 本文的主要内容	10
第二章 基于轨迹库的控制方法	15
2.1 最优控制	15
2.1.1 问题描述	15
2.1.2 动态规划	16
2.2 轨迹优化	18
2.2.1 轨迹最优的必要条件	19
2.2.2 非参数化轨迹优化方法	20
2.2.3 参数化轨迹优化方法	22
2.2.4 混合轨迹优化	23

2.2.5	轨迹优化中的若干问题	23
2.3	基于轨迹库的控制方法	29
2.3.1	基于轨迹跟踪的方法	29
2.3.2	基于在线优化的方法	30
2.3.3	基于函数近似的方法	30
2.4	本章小结	33
第三章	基于轨迹库的仿人机器人站立平衡控制	35
3.1	机器人模型	35
3.2	物理参数辨识	36
3.2.1	物理参数辨识方法	37
3.2.2	两连杆倒立摆模型物理参数辨识方法	38
3.2.3	两连杆倒立摆模型物理参数辨识结果	40
3.3	状态在线估计	40
3.3.1	扩展卡尔曼滤波方法	40
3.3.2	状态估计方法 A	42
3.3.3	状态估计方法 B	43
3.4	基于轨迹库的站立平衡控制方法	45
3.4.1	轨迹优化	45
3.4.2	最近邻控制	46
3.4.3	初始条件自适应网格上的轨迹库生成方法	47
3.4.4	仿真和实验结果	49
3.5	本章小结	59
第四章	持续外力作用下站立平衡控制的期望状态研究	63
4.1	期望状态优化方法	64
4.2	期望状态优化结果	65
4.2.1	两连杆倒立摆模型	65
4.2.2	三连杆倒立摆模型	68
4.2.3	四连杆倒立摆模型	69
4.3	仿真和实验结果	71

4.3.1 仿真	71
4.3.2 实验	73
4.4 本章小结	75
第五章 基于轨迹库的仿人机器人行走控制	77
5.1 动力学不连续系统周期性任务的最优控制	77
5.1.1 问题描述	77
5.1.2 稳态轨迹参数化优化方法	79
5.1.3 值函数和控制律局部近似模型	80
5.1.4 最近邻控制	81
5.2 平面七连杆行走控制	81
5.2.1 动力学不连续系统模型	81
5.2.2 轨迹优化	84
5.2.3 轨迹库生成方法	85
5.2.4 仿真结果	85
5.3 三维七连杆行走控制	95
5.4 本章小结	101
第六章 总结与展望	103
6.1 总结	103
6.2 展望	105
附录 A 碰撞模型	107
参考文献	109
致谢	121
攻读学位期间发表的学术论文目录	123

第一章 绪论

1.1 仿人机器人研究现状

仿人机器人是一种外形像人的机器人，是目前公认的最复杂的机器人之一。仿人机器人研究具有悠久的历史，创造像人一样的机器人一直是人类追求的梦想之一 [1–3]。仿人机器人可以适应人类的生活和工作环境，代替人类完成各种任务，在很多方面扩展人类的能力，因此在服务、医疗、教育、娱乐等多个领域有广阔的应用前景。但是因为控制理论和技术水平等的限制，仿人机器人研究进展一直比较缓慢。近年来随着一些高性能仿人机器人（如图 1.1 中的 ASIMO [4]、HRP-2 [5]、Sarcos [6] 等）研制的成功，仿人机器人研究已经成为机器人研究领域的一个热点。

仿人机器人运动控制水平的高低是决定其能否走向实用的重要因素，因此仿人机器人运动控制研究一直是仿人机器人研究领域的热点问题。目前这方面研究主要集中在站立平衡、行走、奔跑、舞蹈、武术和杂技等方面 [1, 3, 7, 8]。虽然目前在仿人机器人运动控制方面取得了大量可喜的成果，但其在控制的效率、鲁棒性等方面与人类相比仍有很大差距。仿人机器人的结构非常复杂，因此凭借设计者经验为其设计各种参数化运动控制器不仅非常繁琐，而且很难保证最终得到的控制器的控制性能。最优控制提供了采用优化技术设计控制器的系统

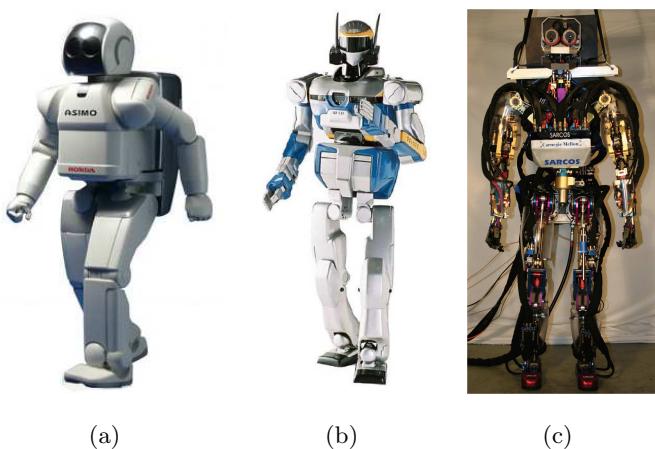


图 1.1 a) ASIMO b) HRP-2 c) Sarcos
Fig. 1.1 a) ASIMO b) HRP-2 c) Sarcos

方法，但是目前采用最优控制实现的仿人机器人运动控制方法，普遍存在着模型过于简单和计算时间过长等问题。随着计算机运算速度的不断提高和存储价格的不断下降，基于数据存储的非参数化控制器受到越来越多的关注。另外，更加先进的优化算法不断涌现，使求解更加复杂的优化问题成为可能。因此，研究基于优化的仿人机器人运动控制方法具有重要的意义。

本文以美国卡耐基-梅隆大学 Sarcos 仿人机器人为主要研究平台，采用优化方法，对仿人机器人运动控制中两个重要问题—站立平衡控制和行走控制，进行了深入的研究。仿人机器人站立平衡是指仿人机器人站立时在受到外界扰动下保持平衡，是仿人机器人运动控制研究的基本问题。双足行走使仿人机器人具备了其他类型机器人不可比拟的运动能力，是仿人机器人区别于其他类型机器人一个最典型特征。对仿人机器人站立平衡和行走控制的研究，不仅能够提高仿人机器人的环境适应能力和运动能力 [35, 53]，还将有助于研制出更加高效的假肢 [9–11]，生成更加生动、自然的动画形象 [12–18]，加深对人类自身运动控制机理的认识 [7, 19–22]。

1.2 站立平衡控制研究动态

仿人机器人站立时脚下的支撑区域比较小，因此踝关节在平衡控制中能够提供的力矩非常有限 [23]。此外，仿人机器人由很多关节和身体部分构成，而且大扰动下上身运动范围比较大，使系统的动力学方程维数高、非线性明显；机器人的关节角、关节角速度和关节力矩均存在一定的限制，进一步增加了控制器设计的难度；而设计对于特定优化指标最优的站立平衡控制更是一种挑战。

仿生机械学领域对站立平衡控制进行了长期而大量的研究，发现了多种不同的站立平衡策略 [22, 27, 28]：当外界扰动比较小时，主要通过脚踝处的力矩控制身体平衡而身体其他关节基本保持不动，如图 1.2 (a) 所示；当扰动比较大或支撑面比较柔软或狭窄时，主要通过快速倾斜上身和快速摆臂等大幅度主动运动来吸收冲击，同时产生更大的地面反作用力以实现身体平衡，如图 1.2 (b) 和 1.2 (c) 所示；当扰动过大而无法在原地保持平衡时，会采用跨步策略实现平衡，如图 1.2 (d) 所示；当扰动为恒定外力时，会利用重力补偿外界扰动，如图 1.2 (e) 所示。人类会根据外界扰动的大小、持续的时间、脚底支撑情况和运动限制采取不同的平衡策略。A. D. Kuo 采用最优控制和状态估计解释人类各种不同平衡策略的选择问题 [20]。S. Park 等人采用全状态反馈增益调度控制解释人类的站立平衡控制，通过优化方法辨识不同扰动下的最佳增益，指出人类的神经系统能够根据生物机械方面的约束调节反馈增益 [21]。仿生机械学对人类

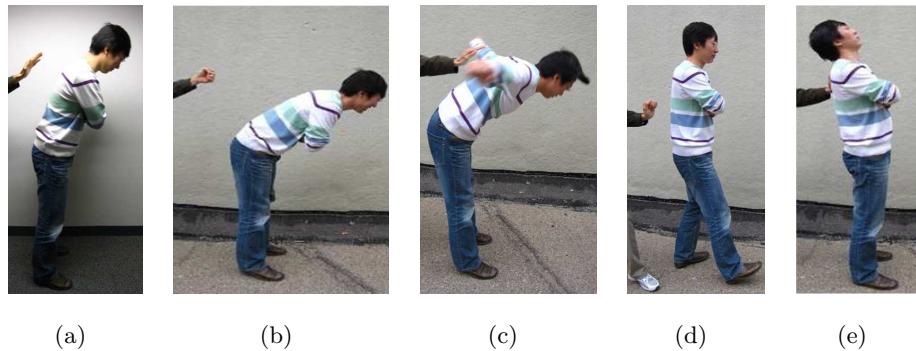


图 1.2 多种平衡策略 a) 脚踝策略 b) 髋策略 c) 摆臂策略 d) 跨步策略 e) 重力补偿策略
Fig. 1.2 Multiple balance strategies a) ankle strategy b) hip strategy c) arm strategy d)
stepping strategy e) gravity compensation strategy

站立平衡控制的研究为仿人机器人站立平衡控制提供了参考。

1.2.1 参数化站立平衡控制方法

为实现仿人机器人的站立平衡控制，目前很多研究将机器人的模型进行简化，然后设计参数化控制器。为了提高控制器的性能，有些研究采用优化手段对控制器的参数进行优化。例如，H. Hemami 和 P. Camana 将机器人建模为由质点和飞轮构成的倒立摆（如图 1.3 (a)），提出了一种非线性反馈平衡控制器 [30]。C. J. Golliday 和 H. Hemami 采用两自由度模型（如图 1.3 (b)），提出了一种线性反馈平衡控制器 [31]。A. Alexandrov 采用平面三连杆倒立摆模型，将站立平衡控制分解为踝关节为主的反馈控制、髋关节为主的反馈控制和膝关节为主的反馈控制，通过数据分析说明人类的平衡控制可以通过这三种典型控制来描述，其稳定性可以通过每个控制器的反馈稳定性来保证 [29]。Acrobot 机器人是一种两连杆机械臂，机器人与底座之间的关节无驱动，只通过两连杆之间关节的驱动实现整个机构的摆起控制 [32, 33]。仿人机器人的质心比较高、支撑面比较小，因此在站立平衡控制中踝关节只能提供非常有限的力矩。当踝关节的力矩饱和时，站立平衡控制与对 Acrobot 机器人的控制非常类似。零力矩点（Zero-moment Point）和质心的位置对维持平衡有着至关重要的影响 [1, 34, 35]。K. Kondak 和 G. Hommel 采用一个 PD 控制器控制质心的位置，在线计算关节期望角加速度实现站立平衡控制 [36]。B. Stephens 在控制质心位置的同时控制机器人的姿态，分别设计质心位置控制和姿态控制，然后将两种控制器集成在一起实现站立平衡控制 [37]。在站立平衡控制中，角动量对抵抗大的外界扰动发挥了很大的作用。文献 [38] 将角动量引入到线性倒立摆模型

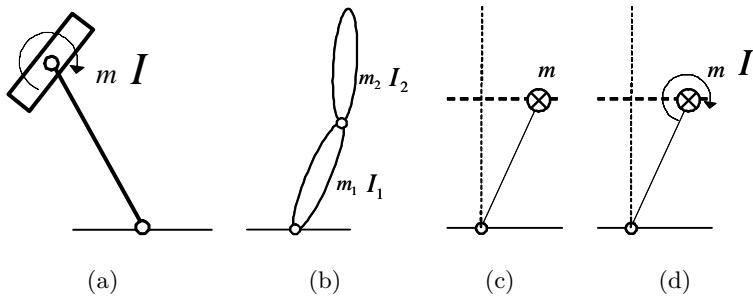


图 1.3 站立平衡采用的各种模型 a) 质点飞轮模型 b) 两自由度模型 c) 线性倒立摆模型 d) 角动量倒立摆模型

Fig. 1.3 Robot models for standing balance control a) The mass and flywheel model b) The two-degree-of-freedom model c) The linear inverted pendulum model d) The angular momentum inducing inverted pendulum mode

(Linear Inverted Pendulum Model) (如图 1.3 (c)), 提出了角动量倒立摆模型 (Angular Momentum inducing inverted Pendulum Mode), 如图 1.3 (d) 所示。T. Komura 等人采用角动量倒立摆模型计算补偿力矩以克服外界扰动引起的角动量 [39]。B. Stephens 首先采用角动量倒立摆模型得到不同控制策略的切换平面, 然后针对每种控制策略设计单独的控制器, 最后将这些控制器集成在一起 [40]。S. Kajita 提出了一种解析动量控制方法 (Resolved Momentum Control) [108], 根据动量参考轨迹和惯性矩阵的伪逆矩阵计算机器人的全身运动轨迹, 实现了踢球和行走中的平衡控制。D. Xing 设计参数化的全状态反馈站立平衡控制器 [44]。为了提高控制器的性能, 研究了控制器参数的优化问题, 得到了不同大小、方向、作用位置的水平外力作用下的最佳控制器参数, 实现了瞬间和持续扰动下的站立平衡控制。

对于仿人机器人的站立平衡控制问题, 虽然目前的方法中有些取得了比较好的效果, 但是这些方法中如何确定控制器的参数仍然比较困难, 而且控制器的性能在很大程度上取决于设计者的经验和水平。探索采用优化手段实现站立平衡控制的方法因此具有非常重要的意义。

1.2.2 基于优化的站立平衡控制方法

最优控制提供了一种利用非线性优化手段设计控制器的系统方法, 不仅能够避免控制器设计中对设计者经验的依赖, 而且能够产生对于特定优化指标最优的控制器 [87]。在这方面, C. G. Atkeson 和 B. Stephens 采用简单的优化指标, 对不同大小的瞬间扰动下的站立平衡控制进行轨迹优化, 产生了和人类反

应相近的各种不同的平衡策略 [41]。但其目的是为了解释人类如何选择站立平衡策略，进而为仿人机器人站立平衡控制器的设计提供指导。其结果是开环控制策略，不能直接用于对仿人机器人的控制。动态规划是求解最优控制的常用方法，但是当应用到仿人机器人控制问题时，往往受到维数问题的限制 [95]。为了加快一般动态规划方法的收敛速度，C. G. Atkeson 和 B. Stephens 采用状态随机采样、轨迹优化和局部模型等多种方法，提出了一种近似动态规划方法，实现了两连杆、三连杆摆起控制和四连杆倒立摆冲击扰动下的平衡控制 [42]。这种近似动态规划方法虽然在一定程度上能够减少动态规划的计算量，但是其计算量和存储量仍然相当可观。对于最优控制的其他方法，Xinjilefu 利用随机规划方法 (stochastic programming) 最小化系统的拉格朗日函数，实现了三维两连杆倒立摆模型瞬间扰动下的站立平衡控制 [43]。D. Xing 采用不同阶次的多项式函数近似控制律函数，然后采用多种优化方法优化多项式参数，实现了多种不同模型的站立平衡控制 [44]。但是这种方法实现的控制器性能不仅决定于控制器的参数，还与基函数的选择有关。该方法的难点在于如何确定表示全局控制律的基函数。

除此之外，计算机动画领域为了生成更加逼真、交互的人物形象，普遍采用了基于物理模型和在线优化的方法 [24, 25]。但是在线优化需要额外的计算时间，一般不适合应用于实时控制。

1.3 双足行走控制研究动态

仿人机器人可以看作是活动基座的多连杆机械臂，其动力学方程呈现高维和非线性的特征。在行走过程中，支撑脚与地面之间的作用力需要满足一定的物理约束 [23, 24]；摆动脚与地面之间的间歇性碰撞，使系统的动力学不连续。此外，摆动脚的高度、机器人的关节角、关节角速度以及关节驱动力矩等存在的限制进一步增加了行走控制的难度，而设计对于特定优化指标最优的行走控制更是一种挑战。

1.3.1 典型的双足行走控制方法

目前行走控制中普遍采用了跟踪提前计算的参考轨迹的方法。在跟踪过程中，采用反馈控制来补偿建模误差、环境变化和外界扰动。

生成参考轨迹最常用的方法是基于零转矩点 (ZMP) 的步态规划 [4, 47, 49, 109]。ZMP 最早是由 Vukobratović 提出，是指地面上一点，来自地面的水平力

矩在该点为零。当 ZMP 点处在支撑点构成的凸包内时，支撑脚能够保持相对地面静止，否则支撑脚可能会相对地面翻转。仿人机器人能否重复规划的步态取决于这个步态对应的 ZMP 能否始终保持在支撑点构成的凸包内。Q. Huang 等采用三次样条插值的方法将髋关节和脚踝的运动轨迹参数化，利用机器人的前向动力学模型计算其他关节的轨迹，在得到所有关节的轨迹后计算对应的 ZMP 轨迹，通过最大化 ZMP 稳定裕度确定最佳的步态参数 [47]。这种方法的缺点在于计算量大，因此不能实时生成步态。基于 ZMP 的轨迹规划的目的是在行走的过程中，始终保持支撑脚相对地面静止，从而保证参考轨迹的物理可实现性。当机器人没有脚或者在行走的过程中有意利用支撑脚相对地面的转动来提高行走效率时，便不适合采用 ZMP 方法规划参考轨迹。

生成参考轨迹另一种常用的方法是基于线性倒立摆模型的步态规划。线性倒立摆模型将机器人的质量集中于一点，同时忽略双腿的质量，通过限制质心在与地面平行的平面内运动，将复杂的多连杆动力学方程简化为线性动力学方程 [48, 50]，如图 1.3 (c) 所示。基于线性倒立摆模型的步态规划方法的优点在于系统的动力学为线性，因此可以采用线性系统控制中许多成熟的方法 [51]。其缺点在于由于忽略了双腿的质量，实际的 ZMP 轨迹和规划的 ZMP 轨迹之间存在偏差，而且偏差会随着运动速度的增加而增大。另外，由于引入了质心运动的约束，使规划得到的行走步态不够自然，行走控制的能量利用效率比较低。

在跟踪参考轨迹方面，ASIMO 机器人采用了地面作用力控制、ZMP 模型控制和脚着陆位置控制克服外界扰动、模型误差、地面不平等因素带来影响 [4]。Q. Huang 等人采用身体姿态控制、实际 ZMP 位置控制和着陆时间控制等，让机器人适应环境的不确定性 [61]。PID 控制、计算转矩控制和滑模控制等反馈控制方法同样已经应用到步态稳定控制中 [62, 63]。R. Tedrake 采用凸优化计算 Lyapunov 函数，评估二次型调节器的吸引域，通过生成大量的二次型调节器将机器人的状态稳定到参考轨迹附近 [64]。J. Morimoto 等人采用鲁棒控制方法计算跟踪参考轨迹所需的最优反馈增益，实现了低转矩的行走控制 [65]。

目前还有大量不需跟踪预先规划的参考轨迹的行走控制方法。例如，早期研究中忽略双腿的质量，将机器人建模成为只有三个自由度的平面倒立摆，通过线性化设计状态反馈控制，实现的双足行走控制 [2]。另外，还有混合零动 [66, 105]、能量控制 [3]、虚拟模型控制 [67]、仿生控制 [72] 和在线轨迹规划 [73, 74] 等方法。

1.3.2 基于优化的双足行走控制方法

虽然在行走控制方面取得了巨大的成功，但是当前的仿人机器人行走控制方法在速度、效率和鲁棒性等方面与人类相比仍有很大差距。以典型的 ASIMO 仿人机器人为例，根据 2007 年本田公布的数据其最高行走速度为 0.75 米/秒，最高奔跑速度为 1.666 米/秒。而人类最高行走速度约为 2.0 米/秒 [45]，2004 年夏季奥运会百米最高奔跑速度约为 10 米/秒，同届马拉松比赛中金牌选手在约 130 分钟内跑完 42.195 公里，平均速度接近 5 米/秒。ASIMO 采用 51.8 伏、重 6 公斤的锂电池，一次充电最长工作时间约为 40 ~ 60 分钟。ASIMO 的行走控制的能量效率只相当于人类行走的十分之一 [46]。而且，目前 ASIMO 还没有展示在自然地面行走的能力。

仿生机械学领域的研究表明行走控制可以解释为最优控制 [106]。最优控制提供了实现更加自然、高效和鲁棒的行走控制的手段 [87]。文献 [68–70] 采用最优控制的常用方法（动态规划方法）实现了行走控制，但是这些方法存在模型过于简单和计算时间过长等问题。

因为优化轨迹比优化全局控制律更容易，所以很多研究采用优化步态轨迹，然后通过反馈跟踪最优步态轨迹的方法。步态轨迹优化通过最小化一定的代价函数，如最小化关节力矩、能量消耗等，得到关节角、关节角速度和关节力矩随时间变化的轨迹。例如，Y. Hasegawa 等人采用分层进化算法在满足 ZMP 约束的前提下生成能量最优的参考步态 [60]。文献 [52, 53] 将关节角、关节角速度和关节力矩作为优化变量直接优化。有些研究则将轨迹进行某种函数近似，如多项式函数近似 [54–57] 和傅立叶展开近似 [58] 等，然后利用常用的非线性优化方法对参数进行优化，如序列二次型优化 (sequential quadratic programming) 方法 [59]。虽然跟踪单一参考轨迹的方法能够快速地实现行走控制，但是所实现的行走控制的稳定性非常有限，即使很小的扰动也可能导致机器人在行走中摔倒。为了提高行走控制的稳定性，文献 [81] 优化具有可调参数的参考轨迹，在行走中根据机器人当前状态在线调节步态参数生成新的最优轨迹，然后采用计算转矩控制方法严格地跟踪生成的轨迹。与跟踪单一参考轨迹的行走控制相比，平面七连杆双足行走控制仿真说明其具有更大的行走控制稳定性。然而，这种方法需要将步态轨迹参数化为几个参数，限制了控制性能的发挥；另外，其在行走控制中需要严格地跟踪在线生成的轨迹，控制效率和对环境的适应能力仍然非常有限。

除了离线生成最优控制器的方法外，R. Tedrake 采用强化学习方法在如图 1.4 所示的 MIT 被动行走机器人上实现了行走控制 [71]。但是，目前通过在



图 1.4 MIT 被动行走机器人

Fig. 1.4 A passive dynamic walker of MIT

线学习方法实现的行走控制采用的模型都比较简单，在仿人机器人上的应用很少。

1.3.3 被动行走

除了上面介绍的行走控制外，还有一类非常特殊的行走控制—被动行走（passive dynamic walking）。被动行走机器人是一种能在重力作用下沿斜面步行而下的简单机构，如图 1.5。被动行走机器人不需要任何驱动和控制，就能产生与人类非常接近的行走。T. McGeer 对被动行走进行了深入的研究，分析了被动行走的机理、稳定性和对参数的敏感程度 [75]。完全的被动行走机器人因为无法补充损失的能量，所以不能在平地上行走。S. Collins 等人研究了三种基于被动行走的主动行走控制，这些机器人只需要加入很少的驱动和简单的控制便能在平地上自然地行走 [46]。

这里简单地分析本文的行走控制与被动行走之间的联系。双足行走机器人的动力学可以描述为如下形式的一组状态方程 [76]:

$$\dot{\mathbf{x}} = a(\mathbf{x}) + b(\mathbf{x})\mathbf{u} \quad (1.1)$$

$$\mathbf{x}^+ = \Delta(\mathbf{x}^-) \quad (1.2)$$

其中 \mathbf{x} 表示由关节角和关节角速度构成的状态变量， \mathbf{u} 表示关节驱动力矩构成的控制变量， \mathbf{x}^- 表示摆动脚与地面碰撞前的状态， \mathbf{x}^+ 表示摆动脚与地面碰撞后的状态，摆动脚与地面之间的碰撞通过碰撞方程 $\Delta(\cdot)$ 描述。状态方程中函数 $a(\cdot)$, $b(\cdot)$ 和 $\Delta(\cdot)$ 只与机器人的构形、动力学参数和地面参数有关。一旦机器人的构形、动力学参数和地面参数确定，则这些函数也随之确定。对于特定的倾

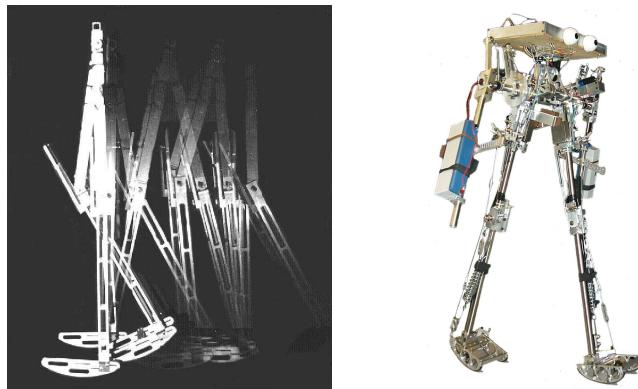


图 1.5 康奈尔大学的被动行走机器人 a) 无驱动 b) 有驱动

Fig. 1.5 Passive dynamic walkers of Cornell University a) Unpowered b) Powered

斜地面和机器人参数,如果状态方程 $\dot{\mathbf{x}} = a(\mathbf{x})$ 存在稳定的极限环,则机器人能够在没有任何驱动的情况下行走,即实现完全的被动行走[75]。如果不存在稳定的极限环,但通过引入一定的控制 $\mathbf{u} = \pi(\mathbf{x})$ 使状态方程 $\dot{\mathbf{x}} = a(\mathbf{x}) + b(\mathbf{x})\pi(\mathbf{x})$ 出现稳定的极限环,则可以实现主动行走。实际的仿人机器人系统的结构非常复杂,在没有控制输入的情况下往往不存在稳定的极限环,因此只能在一定的控制下实现主动行走。在控制时可以优化控制律 $\mathbf{u} = \pi(\mathbf{u})$ 以减小公式(1.1)中的控制量大小,从而充分地利用机器人自身的动力学特性,提高行走控制能量利用效率并实现更加自然的行走,这正是本文研究的重点。

1.4 基于轨迹和局部模型的控制方法研究现状

本文采用轨迹库实现仿人机器人站立平衡和行走控制,目前已有很多研究采用大量的轨迹实现各种类型的控制和在线规划[64, 111–114]。

为了生成交互的动画形象,文献[15, 16, 18]通过运动捕捉(motion capture)将人类关节运动轨迹保存成轨迹库。在生成动画的过程中,利用这些轨迹通过插值和优化等手段在线生成新的动画形象。K. Yin 等人根据当前动量从关节轨迹数据库中选择最合适的平衡恢复动作实现动画人物的平衡控制[110]。

在路径规划方面,概率地图法(Probabilistic Roadmaps PRMs)事先计算并保存随机选择的点之间的路径,查询时只规划查询点与距离最近的网络点之间的路径,然后采用之前保存的路径到达距离目标最近的网络点,最后规划从这网格点到达目标点的路径[77]。这个方法与本文提出的方法有一些相似,都采用了提前计算的轨迹加速在线查询的速度。但主要的不同在于:概率地图法是

一种路径规划算法，而本文提出的方法是反馈控制律；概率地图的结果是双向路径，而在本文控制律作用下的轨迹是单向的。

在机器人控制方面，M. Stolle 和 C. G. Atkeson 采用轨迹库实现了重力感应游戏（marble maze）的控制 [78]。C. G. Atkeson 等人利用轨迹优化加速动态规划的收敛速度，提出了一种近似动态规划方法 [80]。此外，Y. Tassa 等人采用滚动微分动态规划（Receding Horizon DDP）生成参考轨迹库，实现了多连杆仿真鱼的控制 [82]。J. Schierman 通过轨迹优化来处理系统动力学的变化 [83]。强化学习中采用仿真轨迹作为训练数据来近似值函数 [84, 85, 115]。J.J. Murray 等人采用轨迹来估计一组初始状态上的值函数 [86]。仿人机器人行走控制普遍采用了跟踪参考轨迹的方法 [4, 47, 49, 81, 109]。

本文采用微分动态规划得到了值函数的二阶局部近似和最优控制律的一阶局部近似，然后利用这些局部近似模型实现了状态反馈控制 [79, 96]。相似的轨迹优化有 [116]，还包括一些鲁棒轨迹优化方法 [65, 117, 118]。R. Tedrake 等人采用局部有效的线性二次型调节器（LQR）构造全局非线性反馈控制律 [64]。

1.5 本文的主要内容

虽然已有文献对于仿人机器人站立平衡和行走控制进行了大量研究，但仍存在很多问题有待解决：

对于站立平衡控制问题，主要表现在：

- 以往的研究大多集中在参数化控制器上。参数化控制器不仅性能上受到设计者经验和水平的限制，而且其参数的调节也比较困难。
- 以往的研究对指定性能指标下最优的站立平衡控制研究较少。目前采用最优控制实现的站立平衡控制方法生成控制器的计算时间非常可观，而且只能处理瞬间扰动。
- 在人类的工作环境中，持续一段时间的扰动比较常见。但以往的研究主要考虑了瞬间扰动下的站立平衡控制问题，而忽视了持续扰动下的站立平衡控制问题。
- 对于实际的仿人机器人站立平衡控制问题，模型物理参数的辨识、状态的在线估计是不可避免的，但目前实验研究报道较少。

- 由于仿人机器人由多个关节和身体部分构成，相同外力作用下静平衡状态并不唯一。仿人机器人站立平衡控制需要确定某个静平衡状态作为期望状态。期望状态的选取对站立平衡控制的鲁棒性有哪些影响？如何确定最佳的期望状态？目前对这些问题研究很少。

对于行走控制问题，主要表现在：

- 目前的仿人机器人行走控制方法在速度、效率和鲁棒性方面与人类相比仍有很大差距，有很大的提升空间。
- 以往的研究更多地关注于行走控制的稳定性，而对特定优化指标下的最优控制关注较少。目前采用最优控制方法实现的行走控制，存在模型过于简单和生成控制律的计算时间过长等问题。
- 跟踪单条参考轨迹的行走控制方法能够取得的稳定性和效率非常有限，而跟踪多条参考轨迹的行走控制方法，行走效率仍然不高。

在以往研究的基础上，本文研究了基于优化的仿人机器人站立平衡和行走控制问题，着眼于提高控制的效率和鲁棒性。首先研究了多种轨迹优化方法，分析了各自的特点，对轨迹优化中存在的若干问题给出了解决方法。针对采用单一轨迹优化方法存在算法收敛对初始轨迹敏感或结果不能直接用于控制等问题，提出了一种混合轨迹优化方法。总结了基于轨迹库的控制方法，其中重点研究了采用控制律局部近似模型生成状态反馈控制的方法。针对仿人机器人的站立平衡控制问题，研究了实际机器人控制中的物理参数辨识、状态在线估计问题。针对利用轨迹库实现仿人机器人控制中如何减少轨迹数量这一关键性问题，提出了状态空间自适应网格上基于控制器预测性能的轨迹库生成方法。在此基础上，提出了基于轨迹库的站立平衡方法，实现了瞬间和持续扰动下的站立平衡控制。为进一步提高站立平衡控制的鲁棒性，研究了期望状态对控制鲁棒性的影响，提出了期望状态优化方法。针对仿人机器人行走控制问题，首先研究了更为一般的动力学不连续系统周期性任务的最优控制问题，解决了周期性稳态轨迹的优化、动力学不连续处值函数的传递关系等问题。在此基础上，研究了2D和3D七连杆双足机器人轨迹优化、轨迹库生成，针对现有方法存在的不足，提出了基于轨迹库的行走控制方法。

本论文内容安排如下：

第一章介绍了仿人机器人的研究现状和研究意义；概括了仿人机器人站立平衡控制和行走控制研究的难点、意义和现有方法存在的问题；总结了基于轨迹库和局部模型的控制方法。

第二章首先介绍了最优控制及常用解法—动态规划方法和其改进方法，并以单摆摆起控制为例加以说明。接着，研究了轨迹优化方法。将轨迹优化方法分为两类：参数化轨迹优化和非参数化轨迹优化，分别介绍了各自的方法并分析了各自的特点。随后研究了轨迹优化中存在的约束的处理、振铃现象、稳态值函数近似和局部最优等问题，给出了解决方法。针对参数化轨迹优化得到的轨迹不能完全满足动力学约束、不能直接用于控制，而微分动态规划的收敛对初始轨迹敏感等问题，提出了参数化轨迹优化和微分动态规划相结合的轨迹优化方法。最后，总结了基于轨迹库的控制方法，其中重点研究了采用控制律局部近似模型生成状态反馈控制的方法。

第三章针对仿人机器人的站立平衡控制问题，首先介绍了机器人模型。针对站立平衡控制中遇到的物理参数的辨识问题，给出了非线性最小二乘物理参数辨识的结果；针对状态的在线估计问题，研究了基于扩展卡尔曼滤波的状态估计方法。将仿人机器人的站立平衡控制问题转化为无限时域的最优控制问题，给出了单步代价函数和约束条件。为了达到性能要求并减少轨迹库中轨迹的数量，提出了状态空间自适应网格上基于控制器预测性能的轨迹库生成方法，在此基础上提出了基于轨迹库的站立平衡控制方法，实现了瞬间和持续外界扰动下的站立平衡控制。与基于线性二次型调节器（LQR）的增益调度控制器和采用最优增益的全状态反馈控制器进行了比较，仿真和实验说明方法的有效性和性能优势。

第四章针对仿人机器人站立平衡控制中期望状态的选择并不唯一这一问题，研究了期望状态的选择对站立平衡控制鲁棒性的影响，并提出了期望状态的优化方法。该方法与基于轨迹库的站立平衡控制方法相结合，以实现持续扰动下更加鲁棒的站立平衡控制。仿真和实验验证了方法的有效性。

第五章针对仿人机器人的行走控制问题，首先将行走控制问题归结为力学不连续系统周期性任务的最优控制问题，针对目前采用动态规划方法实现中存在的“维数”问题，提出一种邻域最优控制方法。研究了控制中周期性稳态轨迹的优化问题、推导了力学不连续处值函数的传递关系。在此基础上，研究了平面七连杆机器人的行走控制问题，给出了代价函数和约束条件，研究了轨迹优化的方法，并进一步提出了基于轨迹库的行走控制方法。仿真和与基于轨迹跟踪的行走控制的对比结果说明方法的有效性和性能优势。最后，研究了更具

一般意义的三维行走控制问题，给出了轨迹优化的结果。

第六章对全文工作的总结和下一步工作的展望。

第二章 基于轨迹库的控制方法

本章从轨迹优化和构造控制律两个方面，研究基于轨迹库的控制方法，作为后续研究的基础。首先介绍最优控制及常用解法—动态规划方法和其改进方法。动态规划方法是一种全局规划方法，其结果给出了闭环形式的全局状态反馈控制律。但是，动态规划在应用中存在“维数灾”问题，限制了其在仿人机器人运动控制中的应用。轨迹优化是一种在初始轨迹附近的局部优化方法，轨迹优化受维数的影响比较小，但其结果只给出了开环形式的控制律。按照轨迹规划时是否利用了优化问题本身的结构特点，轨迹优化方法可分为非参数化方法和参数化方法。非参数化方法曾经是轨迹优化最有效的方法，但是随着非线性优化方法的发展和计算机速度和存储能力的提高，参数优化轨迹优化方法引起越来越多的重视，已逐渐成为一种流行的方法。本章将研究几种典型的非参数化轨迹优化方法（打靶法、微分动态规划方法等）和一种参数化轨迹优化方法。针对不同轨迹优化算法的特点，本章将提出一种混合的方法，使最终得到的轨迹不仅完全满足系统动力学约束，而且算法收敛对初始轨迹依赖较小。另外，本章还将研究轨迹优化中存在的约束的处理、振铃现象和轨迹平滑、稳态值函数的近似和局部最优等问题，提出解决的方法。轨迹优化只给出了最优控制的开环解，一般不能直接用于反馈控制，因此本章最后将研究采用大量轨迹实现反馈控制的方法，并分析了每种方法各自的特点。

2.1 最优控制

2.1.1 问题描述

本文研究可以采用离散状态方程描述的时不变非线性动力学系统，

$$\mathbf{x}(i+1) = F(\mathbf{x}(i), \mathbf{u}(i)), \quad (2.1)$$

其中 $\mathbf{x}(i) \in \mathcal{R}^n$ 为连续的状态变量， $\mathbf{u}(i) \in \mathcal{R}^m$ 为连续的控制变量， i 为时间索引。无限时域的最优控制问题可以描述为已知 $\mathbf{x}(0)$ ，在满足状态方程 (2.1) 的约束下，求解 $\mathbf{u}(0, \dots, \infty)$ 使总代价函数

$$J = \sum_{i=0}^{\infty} L(\mathbf{x}(i), \mathbf{u}(i)) \quad (2.2)$$

最小，其中 $L(\mathbf{x}, \mathbf{u})$ 为时不变单步代价函数。

在动态规划中引入了值函数，某个状态的值函数是以这个状态为初始状态采用最优控制得到的最小代价

$$V(\mathbf{x}_0) = \min_{\mathbf{u}(0, \dots, \infty)} \sum_{i=0}^{\infty} L(\mathbf{x}(i), \mathbf{u}(i)) \quad (2.3)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.4)$$

在无限时域的最优控制问题中，为使值函数有限，需要使单步代价函数在期望状态处的取值为零。而对于周期性任务控制，因为不存在期望状态，所以需要引入折扣因子以保证值函数有限，

$$V(\mathbf{x}_0) = \min_{\mathbf{u}(0, \dots, \infty)} \sum_{i=0}^{\infty} \lambda^i L(\mathbf{x}(i), \mathbf{u}(i)) \quad (2.5)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.6)$$

其中 $0 < \lambda < 1$ 为折扣因子。

2.1.2 动态规划

动态规划 (Dynamic Programming) 提供了一种求解全局最优状态反馈控制律， $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$ ，的方法，其中 \mathbf{x} 是任意状态， \mathbf{u} 为该状态应采取的最优控制量 [92, 93]。

求解动态规划常用的方法是“值迭代”法。值迭代过程中反复运用 Bellman 方程， $V(\mathbf{x}) = \min_{\mathbf{u}} (L(\mathbf{x}, \mathbf{u}) + V(F(\mathbf{x}, \mathbf{u})))$ ，更新所有采样状态上的值函数直至值函数收敛。得到值函数以后便可以得到整个状态空间上的最优状态反馈控制律， $\mathbf{u}^*(\mathbf{x}) = \arg \min_{\mathbf{u}} (L(\mathbf{x}, \mathbf{u}) + V(F(\mathbf{x}, \mathbf{u})))$ 。如果状态变量为连续的值，通常将其离散化为均匀的网格，值函数和控制律采用状态空间上的均匀网格来表示，网格内部的点通过插值来计算。

如果将状态变量的量化级数表示为 R_x ，控制变量的量化级数表示为 R_u ，状态变量的维数表示为 d_x ，控制变量的维数表示为 d_u ，那么动态规划的计算量将正比于 $R_x^{d_x} R_u^{d_u}$ ，而存储量正比于 $R_x^{d_x}$ 。动态规划方法的计算量和存储量随着状态变量维数的增长而指数增长的问题称为“维数灾”问题，这个问题限制了其在实际中的应用。

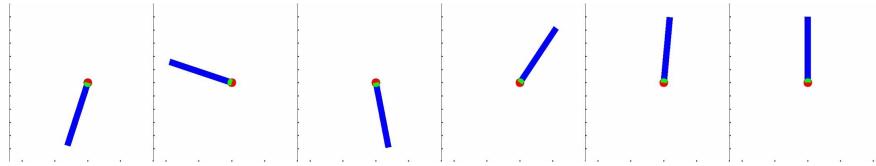


图 2.1 单摆摆起控制
Fig. 2.1 One-link pendulum swingup control

为了加快动态规划的求解速度, Larson 提出了一种改进方法 [93]。该方法在更新值函数时采用了多步优化:

$$V(\mathbf{x}_0) = \min_{\mathbf{u}(0, N-1)} \left(V(\mathbf{x}(N)) + \sum_{i=0}^{N-1} L(\mathbf{x}(i), \mathbf{u}(i)) \right) \quad N > 1 \quad (2.7)$$

其中 N 的确定依据是使 $\mathbf{x}(N)$ 处于当前正在更新的状态 $\mathbf{x}(0)$ 所在的网格以外。在进行多步优化时, 可以采用某种轨迹优化方法来完成。Larson 采用多步优化的目的在于使轨迹片段末尾处的状态远离当前的采样状态, 从而使得采样状态的值函数更新计算不依赖于采样状态自身的值函数, 进而加快动态规划的收敛速度。

在此基础上, 文献 [42, 94] 总结了多种基于轨迹的动态规划方法, 并从以下几个方面提出了进一步改进: 1) 采用局部模型近似值函数和控制律; 2) 同样采用轨迹片段对值函数运用 Bellman 方程进行更新, 所不同的是 N 的确定依据是使 $\mathbf{x}(N)$ 到达目标状态附近; 3) 由近似的控制律或者近似的值函数生成初始轨迹片段, 然后采用一种轨迹优化对得到的轨迹片段进行重新优化; 4) 值函数和控制律的局部近似模型作为轨迹优化的副产品, 在轨迹优化的同时得到; 5) 在优化的过程中, 局部模型之间不断的交换信息以保证结果处处满足 Bellman 方程, 从而使最终得到的值函数和控制律满足全局最优; 6) 采用轨迹优化对每次查询得到的值函数和控制律的预测进行重新优化; 7) 尝试了多种自适应网格表示控制律和值函数。

以上改进方法虽然加快了动态规划的收敛速度, 但是为了得到与全局最优一致的结果, 其计算量仍然相当可观。

下面以单摆摆起控制为实例, 对动态规划的值函数和最优反馈控制律进行更加形象地说明。单摆摆起控制是采用安装在基座上的电机施加一定的控制力矩, 将单摆从任意状态摆动到竖立静止的状态, 如图 2.1 所示。设单摆为均匀细

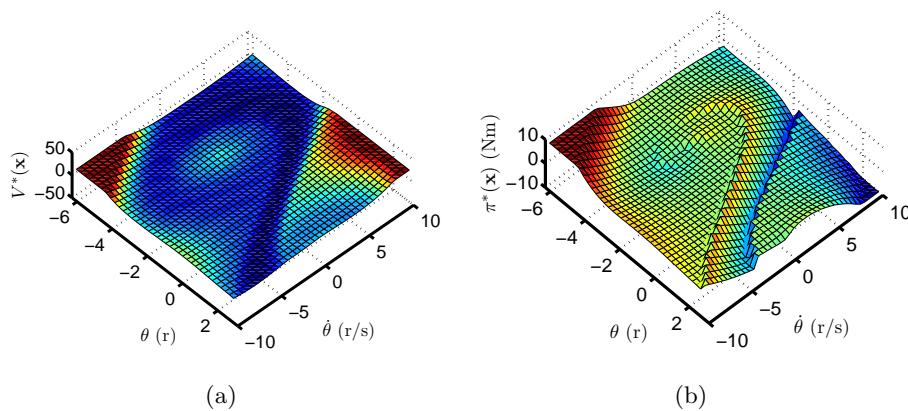


图 2.2 动态规划得到的单摆摆起控制结果 a) 值函数 b) 最优控制律

Fig. 2.2 Results given by dynamic programming for one-link pendulum swingup control
 a) The value function b) The control law

杆，其长度为 1 米，质量为 1kg，那么其对应的离散状态方程为

$$\begin{aligned}\mathbf{x}(i+1) &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(i) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} [1.5g \sin(\theta(i)) + 3u(i)] \\ &= F(\mathbf{x}(i), u(i)),\end{aligned}\quad (2.8)$$

其中 $\mathbf{x} \triangleq (\theta, \dot{\theta})^\top$ 为状态变量, θ 和 $\dot{\theta}$ 分别为关节角和关节角速度, u 为施加的控制力矩, T 为仿真步长 (0.01 秒), g 为重力加速度常数。单摆摆起控制的难点在于: 对于任意初始状态, 使整个控制过程对于指定的单步代价函数,

$$L(\mathbf{x}, \mathbf{u}) = 0.1\theta^2 T + u^2 T, \quad (2.9)$$

的总代价最小。对于无限时域的最优控制，图 2.2 显示了通过动态规划得到的值函数， V^* ，和最优控制律， π^* ，其中期望状态为 $x_d = (0, 0)^\top$ 。

2.2 轨迹优化

轨迹优化是在满足特定约束的前提下寻找使评价函数最小（或最大）的轨迹。轨迹优化最初源于变分法 (calculus of variations)，与最优控制有紧密的联系。轨迹优化将时间连续的无限维最优控制问题转化为时间离散的有限维非线性优化问题。最优控制得到的是闭环形式的状态反馈控制律，而轨迹优化的解是最优控制序列和与之对应的状态轨迹，一般是开环形式的控制律或者局部闭环形式的控制律。

轨迹优化算法根据是否利用了优化问题本身的结构特点可分为非参数化方法和参数化方法：非参数化方法通过前向积分系统的动力学方程从而利用了问题自身的结构特点（例如各种打靶法、梯度方法和微分动态规划法等）；参数化方法将动力学方程作为某种形式的约束，从而把迹优化问题转化为参数优化问题（例如直接配点法和伪谱方法等）。直接打靶法 (direct shooting method) 将每个时刻上的控制变量作为优化变量，然后显式前向积分得到状态轨迹，将轨迹优化问题转化为参数优化问题 [89]。微分动态规划方法 (DDP) 利用值函数、状态方程和单步代价函数的二阶展开，通过类似于 Newton-Raphson 迭代最小化评价函数得到最优轨迹。直接配点法 (direct collocation) 通过函数近似，如多项式函数或样条函数，将状态轨迹和控制轨迹参数化，将函数近似的参数作为优化变量从而将轨迹优化问题转化为一般的非线性优化问题，然后通过序列二次规划 (sequential quadratic programming) 求解 [52, 54–57]，或者将系统的状态方程作为约束，将所有时刻上的状态变量和控制变量作为优化变量，对评价函数进行直接优化 [52, 53]。伪谱方法 (pseudospectral method) 通过采用全局多项式函数近似状态轨迹和控制轨迹，系统状态方程约束由正交配置法近似，从而将原问题参数化。两个主要的伪谱方法是勒让德伪谱方法 (the Legendre pseudospectral method) [90] 和切比雪夫伪谱方法 (the Chebyshev pseudospectral method) [91]。

2.2.1 轨迹最优的必要条件

首先研究轨迹最优的必要条件。定义离散哈密尔顿函数

$$H(i) = L(\mathbf{x}(i), \mathbf{u}(i)) + \lambda(i+1)^T F(\mathbf{x}(i), \mathbf{u}(i)), \quad (2.10)$$

其中 $\lambda(i+1) \in \mathcal{R}^n$ 称为协状态 (co-state)。

轨迹最优的必要条件是满足离散欧拉-拉格朗日方程 (Euler-Lagrange equations) [95]：

$$\mathbf{x}(i+1) = F(\mathbf{x}(i), \mathbf{u}(i)) \quad (2.11)$$

$$\lambda(i) = H_x(i)^T \equiv L_x(\mathbf{x}(i), \mathbf{u}(i))^T + F_x(\mathbf{x}(i), \mathbf{u}(i))^T \lambda(i+1) \quad (2.12)$$

$$H_u(i) \equiv L_u(\mathbf{x}(i), \mathbf{u}(i)) + \lambda(i+1)^T F_u(\mathbf{x}(i), \mathbf{u}(i)) = 0 \quad (2.13)$$

边界条件为

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.14)$$

$$\lambda(N) = V_x(\mathbf{x}(N))^T, \quad (2.15)$$

其中 \mathbf{x}_0 为指定的初始状态。

公式 (2.13) 是 Pontryagin 最大值原理的简化形式, 更一般的形式是

$$\mathbf{u}(i) = \arg \min_{\mathbf{u}(i) \in \mathcal{U}} H(i), \quad (2.16)$$

其中 \mathcal{U} 表示控制变量可行集。

公式 (2.11) 和 (2.12) 通过公式 (2.13) 相互耦合, 边界条件分别位于状态轨迹起始位置和协状态轨迹的末尾位置。轨迹最优的必要条件说明轨迹优化的难点在于求解一个两点边值问题。

2.2.2 非参数化轨迹优化方法

2.2.2.1 打靶法和多目标打靶法

轨迹优化问题本质上是求解两点边值问题, 因此可以采用打靶法或者多目标打靶法来求解。

前向打靶法的基本过程是采用一条初始的控制轨迹 $\mathbf{u}(0, \dots, N-1)$ 生成一条初始的状态轨迹 $\mathbf{x}(0, \dots, N)$, 从而可以计算 $\lambda(0)$ 。通过公式 (2.12) 和 (2.12) 分别求解 $\lambda(i+1)$ 和 $\mathbf{u}(i)$ 。计算末端边值误差 $e_f \triangleq \lambda(N) - V_x(\mathbf{x}(N))^T$ 。采用非线性优化方法求解 $\lambda(0)$ 使得一定精度下满足 $e_f = 0$ 。

打靶法能够得到非常精确的最优轨迹, 且结果满足极值必要条件。但最大的问题在于“如何开始”, 即需要一条非常接近最优轨迹的初始轨迹才能保证算法收敛。另外打靶法对于保守系统通常是可行的, 但是对于非保守系统一般不可行。原因是对于非保守系统, 状态的前向积分和协状态的反向积分都是不稳定的。为了避免这个问题, 可以先通过其他方法(如下面介绍的梯度下降法)得到比较满意的初始控制轨迹和状态轨迹, 然后再采用打靶法。

多目标打靶法将轨迹分成很多轨迹片段, 在每条轨迹片段上反复采用打靶法和插值方法, 直到所有断点满足连续条件。多目标打靶法比打靶法要鲁棒, 可以对很长的轨迹进行优化。

2.2.2.2 微分动态规划

微分动态规划为了加快轨迹优化收敛的速度采用值函数的二阶近似, 在轨迹更新时采用 Newton-Raphson 迭代 [79, 96]。微分动态规划方法在得到最优轨迹的同时能够生成轨迹采样状态上值函数的二阶和控制律的一阶局部近似。

微分动态规划采用了沿初始轨迹的值函数、动力学方程和单步代价函数的二阶泰勒展开。设轨迹上一点 $(\mathbf{x}(i), \mathbf{u}(i))$, 则值函数的二阶泰勒近似为 $V(i) \approx V_0(i) + V_x(i)\hat{\mathbf{x}} + \frac{1}{2}\hat{\mathbf{x}}^T V_{xx}(i)\hat{\mathbf{x}}$, 其中 $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x}(i)$, 状态方程的局部二阶泰勒近似 $\mathbf{x}(i+1) = F(\mathbf{x}(i), \mathbf{u}(i)) \approx F_0(i) + F_x(i)\hat{\mathbf{x}} + F_u(i)\hat{\mathbf{u}} + \frac{1}{2}\hat{\mathbf{x}}^T F_{xx}(i)\hat{\mathbf{x}} + \hat{\mathbf{x}}^T F_{xu}(i)\hat{\mathbf{u}} + \frac{1}{2}\hat{\mathbf{u}}^T F_{uu}(i)\hat{\mathbf{u}}$, 其中 $\hat{\mathbf{u}} = \mathbf{u} - \mathbf{u}(i)$, 单步代价函数的局部二阶泰勒近似为 $L(i) \approx L_0(i) + L_x(i)\hat{\mathbf{x}} + L_u(i)\hat{\mathbf{u}} + \frac{1}{2}\hat{\mathbf{x}}^T L_{xx}(i)\hat{\mathbf{x}} + \hat{\mathbf{x}}^T L_{xu}(i)\hat{\mathbf{u}} + \frac{1}{2}\hat{\mathbf{u}}^T L_{uu}(i)\hat{\mathbf{u}}$ 。

定义一个中间函数: $Z(\mathbf{x}, \mathbf{u}) = L(\mathbf{x}, \mathbf{u}) + V(F(\mathbf{x}, \mathbf{u}))$ 。给定一条轨迹, DDP 沿轨迹反向计算轨迹上的值函数、其一阶和二阶偏微分:

$$Z_x(i) = L_x(i) + V_x(i)F_x(i) \quad (2.17)$$

$$Z_u(i) = L_u(i) + V_x(i)F_u(i) \quad (2.18)$$

$$Z_{xx}(i) = L_{xx}(i) + F_x(i)^T V_{xx}(i)F_x(i) + V_x(i)F_{xx}(i) \quad (2.19)$$

$$Z_{uu}(i) = L_{uu}(i) + F_u(i)^T V_{xx}(i)F_u(i) + V_x(i)F_{uu}(i) \quad (2.20)$$

$$Z_{ux}(i) = L_{ux}(i) + F_u(i)^T V_{xx}(i)F_x(i) + V_x(i)F_{ux}(i) \quad (2.21)$$

$$\mathbf{K}(i) = Z_{uu}(i)^{-1} Z_{ux}(i) \quad (2.22)$$

$$\Delta \mathbf{u}(i) = Z_{uu}(i)^{-1} Z_u(i) \quad (2.23)$$

$$V_x(i-1) = Z_x(i) - Z_u(i)\mathbf{K}(i) \quad (2.24)$$

$$V_{xx}(i-1) = Z_{xx}(i) - Z_{xu}(i)\mathbf{K}(i), \quad (2.25)$$

对于周期性任务, 无限时域的轨迹优化需要考虑折扣因子, 文献 [97] 推导了沿轨迹反向计算轨迹上的值函数、其一阶和二阶偏微分的递推关系:

$$Z_x(i) = L_x(i) + \lambda V_x(i)F_x(i) \quad (2.26)$$

$$Z_u(i) = L_u(i) + \lambda V_x(i)F_u(i) \quad (2.27)$$

$$Z_{xx}(i) = L_{xx}(i) + \lambda F_x(i)^T V_{xx}(i)F_x(i) + \lambda V_x(i)F_{xx}(i) \quad (2.28)$$

$$Z_{uu}(i) = L_{uu}(i) + \lambda F_u(i)^T V_{xx}(i)F_u(i) + \lambda V_x(i)F_{uu}(i) \quad (2.29)$$

$$Z_{ux}(i) = L_{ux}(i) + \lambda F_u(i)^T V_{xx}(i)F_x(i) + \lambda V_x(i)F_{ux}(i) \quad (2.30)$$

$$\mathbf{K}(i) = Z_{uu}(i)^{-1} Z_{ux}(i) \quad (2.31)$$

$$\Delta \mathbf{u}(i) = Z_{uu}(i)^{-1} Z_u(i) \quad (2.32)$$

$$V_x(i-1) = Z_x(i) - Z_u(i)\mathbf{K}(i) \quad (2.33)$$

$$V_{xx}(i-1) = Z_{xx}(i) - Z_{xu}(i)\mathbf{K}(i), \quad (2.34)$$

其中 $0 < \lambda < 1$ 为折扣因子。

微分动态规划的过程概括为：

- 采用初始控制序列 $\mathbf{u}(0, \dots, N-1)$ 和系统的状态方程 (2.1) 生成状态轨迹 $\mathbf{x}(0, \dots, N)$, 并计算总代价

$$J = V(\mathbf{x}(N)) + \sum_{i=0}^{N-1} L(\mathbf{x}(i), \mathbf{u}(i)) \quad (2.35)$$

- 沿当前轨迹按照公式 (2.17)-(2.25) 或公式 (2.26)-(2.34) 反向计算 $V_x(N, \dots, 0)$ 、 $V_{xx}(N, \dots, 0)$ 、 $\mathbf{K}(N, \dots, 0)$ 和 $\Delta\mathbf{u}(N, \dots, 0)$ 。
- 对于某个 $\epsilon > 0$, 应用下面的反馈控制律更新轨迹并计算新轨迹的总代价 J^{new} 。
- $\mathbf{u}^{new}(i) = \mathbf{u}(i) - \epsilon \Delta\mathbf{u}(i) - \mathbf{K}(i)(\mathbf{x}^{new}(i) - \mathbf{x}(i)), \quad (2.36)$
- 如果 $J^{new} < J$ 返回 2, 3 重复, 否则减小 $\epsilon = \epsilon/2$, 重复 3。
- 以上过程不断进行, 直至总代价无法继续减小或已经达到了所需的精度。

在迭代的过程中, 如果 Z_{uu} 接近奇异, 可以采用类似于 Levenberg-Marquardt 的方法。

2.2.3 参数化轨迹优化方法

微分动态规划在轨迹更新过程中沿当前轨迹计算值函数、状态方程和单步代价函数的一阶和二阶偏微分, 其轨迹更新利用了 Newton-Raphson 迭代, 因此需要当前轨迹是“凸”轨迹以保证其快速收敛。对于非线性很强的系统, 初始轨迹的好坏对微分动态规划方法能否收敛到局部最优的轨迹上有较大的影响。参数化轨迹优化方法将最优控制问题转化为一般的非线性优化问题, 然后通过常规的非线性优化方法求解。当对最终解了解甚少时, 参数化轨迹优化方法和微分动态规划方法相比在寻找更好的轨迹方面更加鲁棒。

一种参数化轨迹优化方法将所有采样时刻上的状态变量和控制变量作为优化变量, $\mathbf{y} \triangleq \{\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(N), \mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(N-1)\}$ 。将每个时刻上的状态方程作为路径约束,

$$\mathbf{0} \leq \begin{bmatrix} \mathbf{x}(1) - F(\mathbf{x}(0), \mathbf{u}(0)) \\ \mathbf{x}(2) - F(\mathbf{x}(1), \mathbf{u}(1)) \\ \vdots \\ \mathbf{x}(N) - F(\mathbf{x}(N-1), \mathbf{u}(N-1)) \end{bmatrix} \leq \mathbf{0} \quad (2.37)$$

这样一个最优控制问题便转化为一个一般的非线性优化问题,

$$\min_{\mathbf{y}} V(\mathbf{x}(N)) + \sum_{i=0}^{N-1} L(\mathbf{x}(i), \mathbf{u}(i)) \quad (2.38)$$

可以采用常规的非线性优化方法求解, 例如基于 SQP 算法的 SNOPT [59]。这种方法看似简单, 实际使用时却非常鲁棒, 逐渐成为轨迹优化中比较一种流行的方法 [88, 107]。

2.2.4 混合轨迹优化

参数化方法的优点在于可以大大地减少数值计算不稳定带来的问题, 由比较差的初始轨迹得到满意的结果。其缺点在于其结果往往不能严格地满足系统的动力学约束, 因此一般不能直接用于机器人控制。

非参数化方法因为在轨迹优化的过程中利用了动力学积分, 所以能够生成完全满足系统动力学的轨迹; 而且有些非参数化方法, 如微分动态规划方法, 能够在生成轨迹的同时得到其他有用的信息(如值函数和控制律的局部近似模型)。其缺点在于受初始轨迹好坏的影响比较大, 即当初始轨迹远离实际的最优轨迹时, 可能导致算法无法收敛。

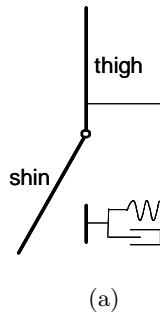
根据不同轨迹优化算法的特点, 我们提出了一种混合轨迹优化方法: 在进行轨迹优化时, 首先采用微分动态规划方法(DDP)对轨迹进行优化, 如果无法收敛到最优轨迹, 则利用参数化轨迹优化方法(SNOPT)为其生成初始轨迹再利用 DDP 进一步优化。采用 SNOPT 进行轨迹优化时, 为了避免陷入同样的局部极小, 可以采用不同的初始轨迹, 例如采用初始状态和期望状态的连线作为其初始轨迹。最终得到完全满足系统动力学约束、能够直接用于机器人控制的高分辨率轨迹, 同时得到轨迹采样状态上控制律的局部近似模型。

2.2.5 轨迹优化中的若干问题

2.2.5.1 约束的处理

轨迹优化的过程中, 除了动力学约束(2.1), 还可能存在状态变量和控制变量约束, 如

$$\mathbf{b}(\mathbf{x}(i), \mathbf{u}(i), i) \leq 0, \quad 0 \leq i < N \quad (2.39)$$



(a)

图 2.3 通过弹簧阻尼模型满足膝关节角约束

Fig. 2.3 A spring damper is used to satisfy the constraint on the knee angle.

或

$$\mathbf{x}_L \leq \mathbf{x}(i) \leq \mathbf{x}_U \quad 0 \leq i \leq N \quad (2.40)$$

$$\mathbf{u}_L \leq \mathbf{u}(i) \leq \mathbf{u}_U \quad 0 \leq i < N \quad (2.41)$$

对于参数化轨迹优化方法, 公式 (2.39) 可以转化为优化变量 \mathbf{y} 的路径约束的一部分

$$c_L \leq c(\mathbf{y}) \leq c_U \quad (2.42)$$

状态变量和控制变量上的约束 (2.40) 和 (2.41) 转化为优化变量的约束,

$$\mathbf{y}_L \leq \mathbf{y} \leq \mathbf{y}_U, \quad (2.43)$$

其中

$$\mathbf{y}_L \triangleq \left\{ \underbrace{\mathbf{x}_L, \dots, \mathbf{x}_L}_N, \underbrace{\mathbf{u}_L, \dots, \mathbf{u}_L}_{N-1} \right\} \quad (2.44)$$

和相应的 \mathbf{y}_U 。

微分动态规划方法主要用于求解无约束的轨迹优化问题, 对于有约束的轨迹优化需要做特殊的处理。对于控制变量上的约束公式 (2.41), 可以在更新控制变量时通过限制控制变量的取值来实现。对于状态变量上的约束, 公式 (2.39) 和 (2.40), 一种简单的方法是可以通过修改代价函数来近似。该方法在违反约束时在代价函数中引入二次型惩罚项。但是该方法存在以下缺点: 1) 只能近似地满足约束; 2) 因为修改了代价函数, 可能引入额外的局部最小值; 3) 较大的惩罚权重会降低算法收敛的速度。

处理状态变量约束的另一种方法是作用力场法。作用力场法在状态快违反约束时, 引入作用力将状态拉回到可行集。例如在站立平衡和行走控制中, 膝

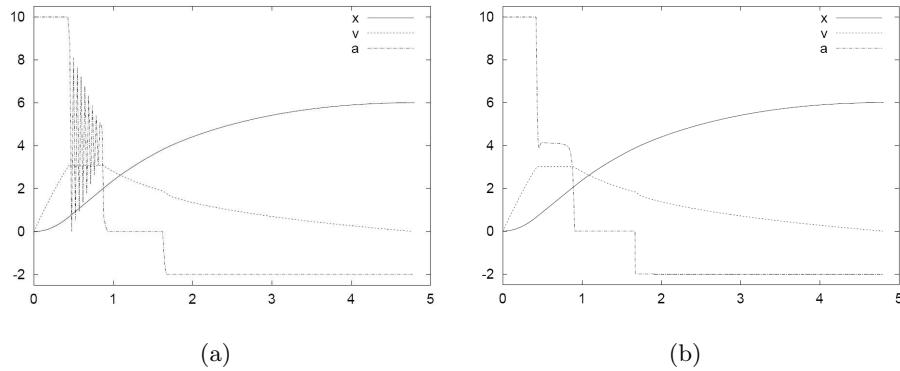


图 2.4 振铃现象和轨迹平滑 [98]

Fig. 2.4 Ringing phenomenon and trajectory smoothing [98]

关节的转动角度是受限制的，表现为膝关节的转角约束。在优化的过程中，可以通过加入弹簧阻尼模型对膝关节转角进行限制，如图 2.3 所示。如果采用作用力场法来满足状态约束，需要注意仿真步长的选取。如果仿真步长过大引起系统不稳定进而影响优化算法收敛的速度。

2.2.5.2 振铃现象和轨迹平滑

在轨迹优化过程中，尤其是参数化轨迹优化过程中，可能会出现“振铃”现象 [98]。图 2.4 (a) 显示了位置 x 、速度 v 和加速度 a 随时间的变化，加速度在中间值附近的时候出现了振铃现象。振铃出现的地方，控制量会快速切换，这是实际控制中不希望出现的。为了消除振铃现象，可以在代价函数中增加控制量切换的代价，实现一阶平滑 $\sum_{i=0}^{N-2} (\mathbf{u}(i+1) - \mathbf{u}(i))^2$ 或者二阶平滑 $\sum_{i=1}^{N-2} (\mathbf{u}(i+1) + \mathbf{u}(i-1) - 2\mathbf{u}(i))^2$ 。图 2.4 (b) 为图 2.4 (a) 中的轨迹经过一阶平滑后的结果。

2.2.5.3 稳态值函数近似

为了采用有限时域 (N 为有限值) 的轨迹优化得到无限时域 ($N \rightarrow \infty$) 的最优解，在轨迹优化前需要得到末端状态处的稳态值函数， $V(\mathbf{x}(N))$ 。

在镇定控制中，期望状态为系统的稳定状态，且单步代价函数在期望状态处的取值为零。为了得到期望状态附近的稳态值函数的近似，可以将系统在期望状态线性化，然后求解 Riccati 矩阵方程

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} = 0, \quad (2.45)$$

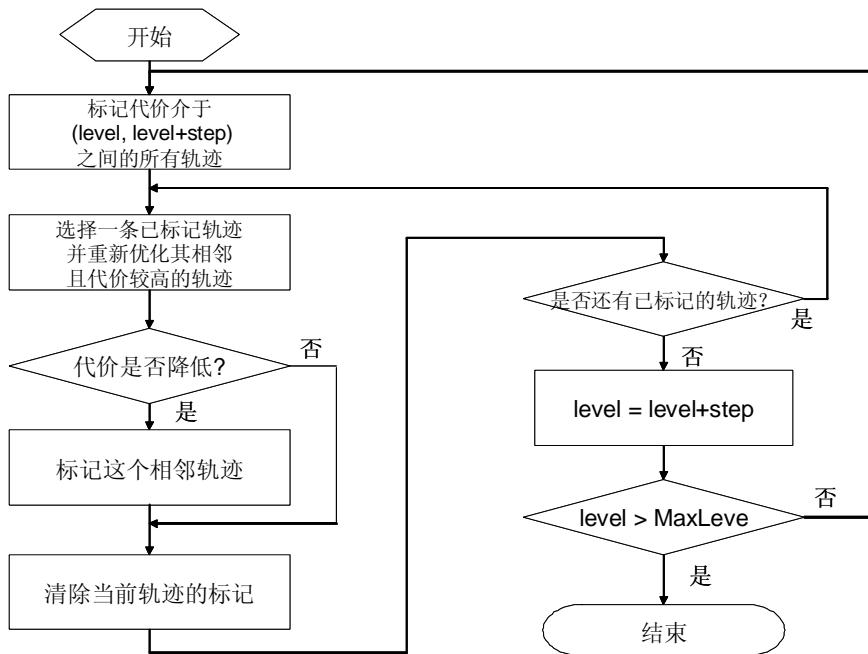


Fig. 2.5 The flowchart of trajectory re-optimization

其中 $\mathbf{A} \triangleq F_x$ 和 $\mathbf{B} \triangleq F_u$ 分别为状态方程 F 关于期望状态和期望控制的偏微分。
期望状态处稳态值函数可以近似为

$$V(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_d)^T \mathbf{P}(\mathbf{x} - \mathbf{x}_d), \quad (2.46)$$

其中 \mathbf{x} 表示期望状态附近的状态， \mathbf{x}_d 表示期望状态。

本文采用微分动态规划得到稳态值函数近似。对于镇定控制，期望状态为不动点 (fixed point)。对于周期性任务控制，周期性稳态轨迹为极限环 (limit cycle)。计算不动点或极限环上的稳态值函数的过程是通过公式 (2.17)-(2.25) 或公式 (2.26)-(2.34) 沿不动点或极限环反复更新值函数，直至值函数稳定不变或达到指定的精度。值函数更新时， V_x 和 V_{xx} 初值的选取并不重要，可以初始化为零。

2.2.5.4 局部最优和改善方法

轨迹优化是局部优化，即最终得到的最优轨迹只是初始轨迹邻域内的最优轨迹。为了尽可能得到全局最优的轨迹，可以在得到了最优轨迹以后，对所有得到的最优轨迹进行重新优化。重新优化的流程如图 2.5 所示，在重新优化的过程中，以代价比较低的轨迹作为初始轨迹，对其相邻的代价比较高的轨迹进行

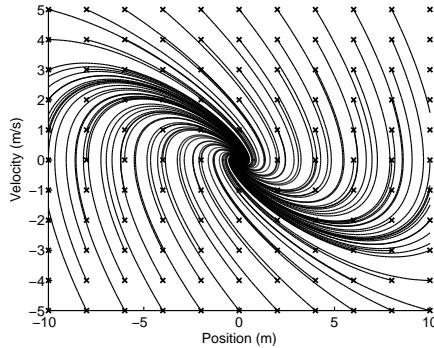


图 2.6 线性系统轨迹优化
Fig. 2.6 Trajectory optimization for a linear system

重新优化。这个过程从代价最低的轨迹开始 ($level = 0$)，并逐渐提高重新优化的轨迹代价范围，($level, level + step$)。如果某条轨迹经过重新优化后代价有所下降，那么这条轨迹相邻的轨迹需要进行重新优化。逐渐提高代价水平以扩大优化的范围，直到所有轨迹重新优化完毕。

下面以一个线性系统为例，分析轨迹优化中的局部最优问题。控制任务为通过控制一个水平力将水平地面上质量为 1kg 的箱子从某个位置移动到坐标原点并最终停止在那里。控制的难点在于单步代价函数惩罚当前位置与期望位置的误差及所施加控制的平方和。单步代价函数定义为 $L(\mathbf{x}, \mathbf{u}) \triangleq \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}$ ，其中 $\mathbf{x} \triangleq (p, v)^T$ 为系统的状态， p 为当前位置， v 为当前速度， $\mathbf{Q} = \text{diag}(1, 0)$ ， $\mathbf{R} = 1$ ， \mathbf{u} 为控制变量（作用在箱子上的水平力），期望状态为 $(0, 0)^T$ 。

如果忽略地面摩擦，则系统的离散状态方程为

$$\begin{aligned} \mathbf{x}(i+1) &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(i) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \mathbf{u}(i), \\ &= \mathbf{A}\mathbf{x}(i) + \mathbf{B}\mathbf{u}(i) \end{aligned} \quad (2.47)$$

其中 T 为仿真步长。

对于无限时域的轨迹优化，单步代价函数在期望状态处的取值为零，因此可以不考虑折扣因子。将状态空间离散化为均匀网格，将网格点上的状态作为初始状态，采用初始状态和期望状态的连线作为初始轨迹，然后采用参数化轨迹优化方法优化轨迹。图 2.6 为轨迹优化的结果。对于线性系统，其代价函数为凸函数，且约束为线性，所以轨迹优化为凸优化，轨迹优化可以得到全局最优的结果。

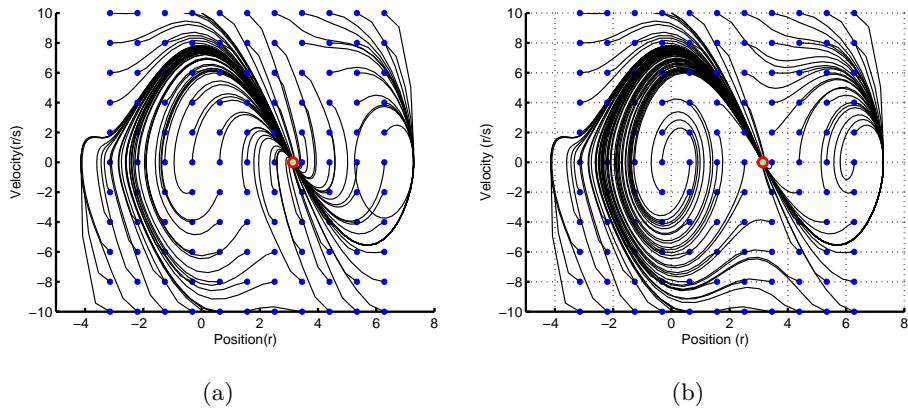


图 2.7 单摆摆起控制轨迹优化 a) 局部最优 b) 全局最优

Fig. 2.7 Trajectory optimization for one-link pendulum swingup control
a) local optima
b) global optima

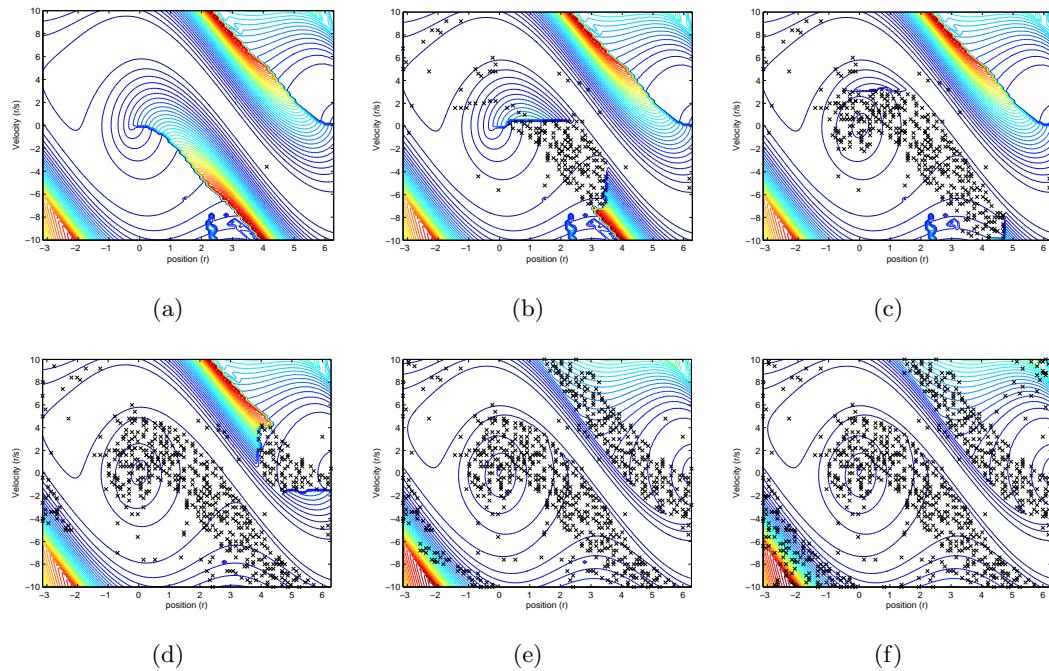


图 2.8 单摆摆起控制轨迹重新优化

Fig. 2.8 Trajectory re-optimization for one-link pendulum swingup control

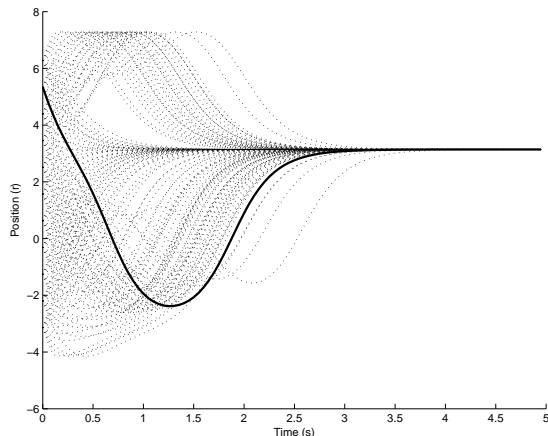


图 2.9 基于轨迹跟踪的单摆摆起控制

Fig. 2.9 One link swing-up control by trajectory tracking.

下面以单摆摆起任务为例，分析非线性系统的轨迹优化中的局部最优问题。单摆摆起控制的状态方程为方程 (2.8)，所采用的单步代价函数为方程 (2.9)。同样将状态空间离散化为均匀的网格，将网格点上的状态作为初始状态。轨迹优化中为了生成初始轨迹，将系统在期望状态附近线性化并设计线性二次型调节器，将线性二次型调节器响应轨迹作为初始轨迹，然后采用参数化方法进行轨迹优化。虽然本问题中代价函数是凸函数，但约束为比较复杂的非线性函数，整个优化不是凸优化，因此参数化轨迹优化的结果不能保证与全局最优一致。图 2.7 (a) 为轨迹优化的结果，其中目标状态为 $(\pi, 0)$ 。图 2.7 (b) 为采用动态规划得到的最优轨迹。通过对比发现图 2.7 (a) 中的部分结果为局部最优。为减少局部最优的轨迹，可以对所有得到的轨迹进行重新优化。图 2.8 中显示了重新优化过程中值函数下降的状态及此时对应的值函数的等高线。通过对得到的轨迹进行重新优化，最终得到了与图 2.7 (b) 一致的结果。

2.3 基于轨迹库的控制方法

2.3.1 基于轨迹跟踪的方法

基于轨迹库实现控制最简单的方法是根据当前状态选择并跟踪某条轨迹。实现的步骤是：1) 离线生成大量最优轨迹；2) 在控制的过程中，根据当前状态， x ，选择某条起始状态， $x^j(0)$ ，最接近的轨迹作为参考轨迹，或者几条起始状态最接近的轨迹通过插值在线生成从当前状态开始的轨迹作为参考轨迹，然后采用一定的反馈控制跟踪参考轨迹。例如，对于单摆摆起控制，事先以一些可

能的状态作为起始状态生成最优轨迹，在控制中根据当前状态选择起始状态最近接的轨迹，然后采用 $\mathbf{u}(t) = \mathbf{u}^j(t) - \mathbf{K}(\mathbf{x}(t) - \mathbf{x}^j(t))$ 跟踪这条轨迹，其中 $\mathbf{u}(t)$ 代表当前的控制量， $\mathbf{u}^j(t)$ 为选定的轨迹对应的控制， \mathbf{K} 为反馈增益矩阵， $\mathbf{x}(t)$ 为当前状态， $\mathbf{x}^j(t)$ 为选定的轨迹上对应的参考状态。图 2.9 给出了这种控制的一个例子。

然而这种方法需要事先保持大量的轨迹，因此在机器人控制中很少采用。而且仿人机器人属于一类欠驱动系统，即使采用高增益有时也很难跟踪选定的轨迹。

2.3.2 基于在线优化的方法

随着计算机运算速度的提高，在线优化开始应用于实时控制的场合 [101, 102]。尽管如此，如何提高在线优化的速度仍然是需要解决的问题。基于在线优化的方法利用事先计算并保存的最优轨迹加快在线优化的速度。在线优化时根据当前状态直接采用起始状态最接近的轨迹作为初始轨迹，然后进一步优化；或者根据当前状态采用起始状态最接近的几条轨迹通过插值得到初始轨迹，然后进一步优化。在每次在线优化结束后，可以将所生成的最优轨迹保存到轨迹库，以避免未来重复运算。

该方法的优点在于轨迹库比较小，离线生成轨迹库所需的计算时间比较短。但其缺点在于在线优化需要额外的计算开销，通常不适合实时控制的场合。

2.3.3 基于函数近似的方法

轨迹优化的结果建立了状态、控制随时间变化的关系，即 $\mathbf{x}(t)$ 和 $\mathbf{u}(t)$ 。对于无限时域的最优控制，其最优控制律成为与时间无关的函数，因此轨迹优化的结果通过采样时间这个参数建立了采样状态和控制之间的对应关系，即 $\{\mathbf{u}(t), \mathbf{x}(t)\} \rightarrow \mathbf{u}(\mathbf{x})$ 。有了轨迹采样状态和最优控制的对应关系，便可以利用各种非线性函数的近似方法近似控制律函数，如插值法和多种监督学习方法等。

一种通过插值实现状态反馈的方法首先将状态离散化为均匀的网格，然后以网格点上的状态作为初始状态生成最优轨迹，最后只保存最优轨迹起始状态和对应的控制量。在控制的规程中，通过线性插值得到任意状态对应的控制。图 2.10 显示采用这种方法得到状态反馈控制律的两个例子。

微分动态规划方法在生成最优轨迹的同时能够得到控制律在轨迹采样状态上的局部近似模型

$$\mathbf{u} = \mathbf{u}^p - \mathbf{K}(\mathbf{x} - \mathbf{x}^p), \quad (2.48)$$

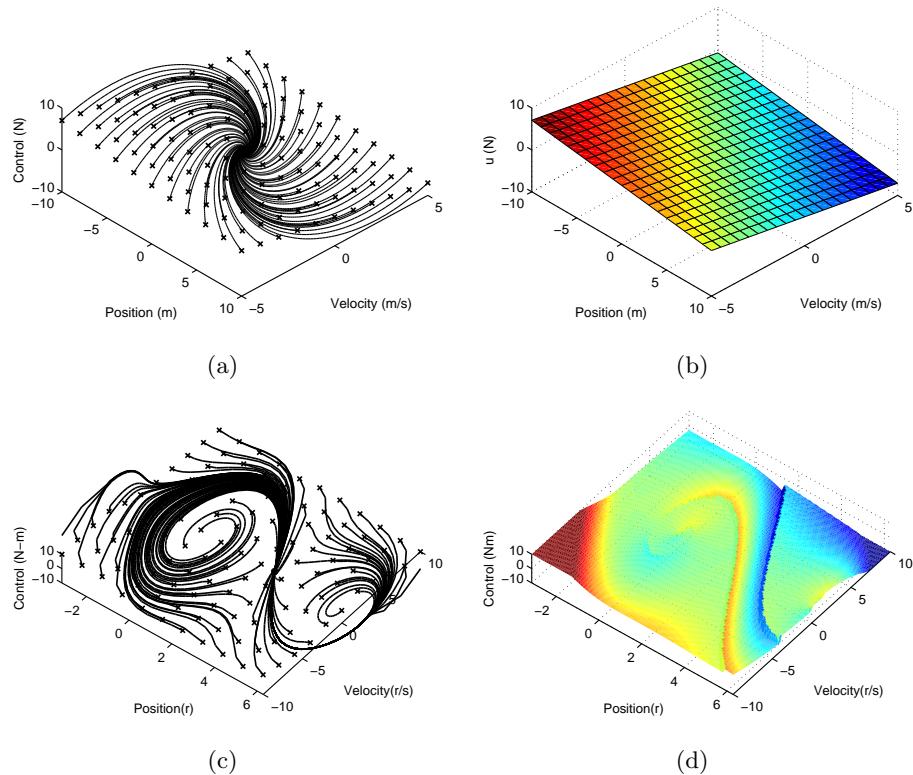


图 2.10 a) 线性系统控制的最优轨迹 b) 插值得到的线性系统控制律 c) 非线性系统控制的最优轨迹 d) 插值得到的非线性系统控制律

Fig. 2.10 a) Optimal trajectories of a linear system control b) Control law by interpolation for the linear system control c) Optimal trajectories of a nonlinear system control d) Control law by interpolation for the nonlinear system control

其中 \mathbf{x} 为状态变量, \mathbf{u} 为控制变量, \mathbf{x}^p 为局部模型对应的采样状态, \mathbf{u}^p 为对应的最优控制轨迹。在控制过程中可以利用这些控制律的局部近似模型预测当前状态对应的控制, 从而可以实现轨迹邻域内的最优状态反馈控制。

一种方法是利用 K 个距离最近的控制律局部模型的预测, \mathbf{u}_i , 将他们的归一化加权平均和作为当前控制的最佳预测:

$$\mathbf{u} = \frac{\sum_{i=1}^K \exp(-d_i) \mathbf{u}_i}{\sum_{i=1}^K \exp(-d_i)}, \quad (2.49)$$

其中 d_k 表示局部模型与当前状态的距离。

一种最简单的方法是采用最近的局部模型来预测当前的控制

$$\mathbf{u} = \bar{\mathbf{u}} - \bar{\mathbf{K}}(\mathbf{x} - \bar{\mathbf{x}}), \quad (2.50)$$

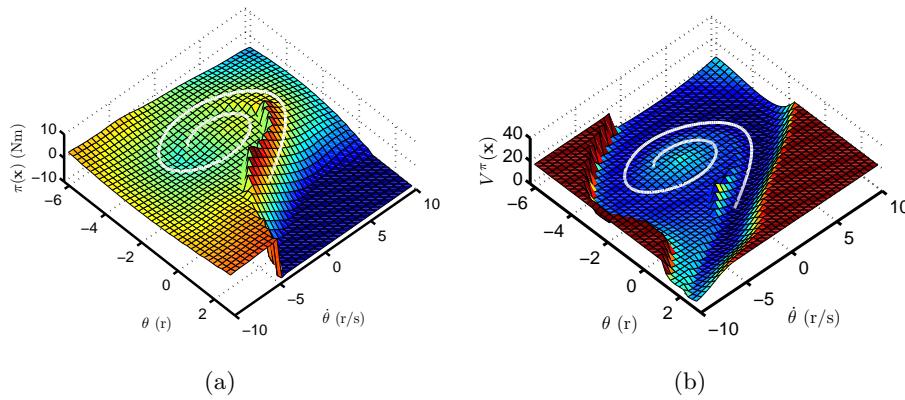


图 2.11 单摆摆起邻域最优控制 a) 控制律 b) 对应的代价

Fig. 2.11 Neighboring optimal control for one-link pendulum swingup a) The control law
b) Its cost

其中 \bar{u} , \bar{x} 和 \bar{K} 分别是距离最近的局部模型的控制、状态和增益矩阵。为了提高最近邻搜索的效率，可以将局部模型保存为 kd-tree [99]。

与其他方法相比，采用控制律局部近似模型实现反馈控制的优点在于：控制律局部近似模型可以在轨迹优化的同时得到；在线查询的计算量小，适合实时控制；不需要事先确定表示控制律函数的基函数，控制律函数的表示更加灵活；构造状态反馈控制律所需的轨迹数目较少。例如，对于一个线性系统，其稳态最优控制律也为线性

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}(\mathbf{x} - \mathbf{x}_d), \quad (2.51)$$

其中 \mathbf{R} , \mathbf{B} , \mathbf{P} 的定义见公式 (2.45)，因此其状态反馈控制律可以只通过一个控制律局部近似模型描述。

对于一个非线性系统，控制律通常为非线性，因此需要很多控制律局部近似模型描述。例如，对于单摆摆起控制，图 2.11 中显示了采用一条最优轨迹所有采样状态上的控制律局部近似模型，利用一阶最近邻控制实现的邻域最优控制和其对应的代价。我们将得到控制律代价与最优控制的代价进行了比较，其中全局最优控制可以通过动态规划方法得到。图 2.12 显示了对比结果。对比结果说明当状态靠近最优轨迹时，邻域最优控制的代价与全局最优控制的代价接近；当状态远离最优轨迹时，邻域最优控制的性能开始变差。尽管如此只采用一条轨迹采样状态上控制律的局部模型构造的控制器，仍然能够在轨迹邻域很大状态空间上得到和全局最优控制器相近的性能。这个结果表明我们有可能利用少量的最优轨迹实现较大状态空间上的控制。

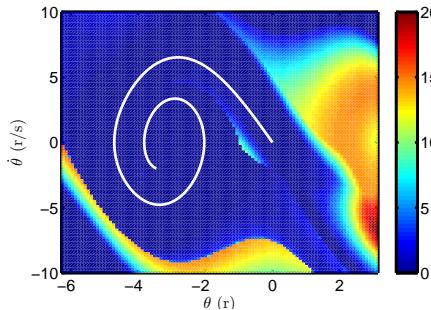


图 2.12 单摆摆起邻域最优控制的代价与全局最优控制的代价之间的误差 (白线为所采用的最优轨迹)

Fig. 2.12 The cost deviation of neighboring optimal control from the global optimal control for one-link pendulum swingup. The white line is the optimal trajectory used.

2.4 本章小结

动态规划提供了一种生成全局状态反馈的方法，但是在实际应用中存在“维数灾”问题。对于仿人机器人的控制问题，控制问题的维数通常比较高，限制了动态规划方法的应用。轨迹规划作为一种局部优化方法受维数问题的影响比较小，能够更快地得到满足特定初始条件的开环控制结果。轨迹优化方法可分为两类：参数化轨迹优化和非参数化轨迹优化。参数化轨迹优化算法的收敛对初始轨迹比较鲁棒，但得到的轨迹往往不能完全满足动力学约束，一般不能直接用于控制；而非参数化方法生成的轨迹能够完全满足动力学约束，但算法的收敛与否对初始轨迹的好坏比较敏感。因此本章提出了一种混合轨迹优化方法，以弥补每种方法各自的不足。这种方法将参数化轨迹优化和微分动态规划相结合，不仅能够得到可直接用于机器人控制的轨迹，而且能够得到轨迹采样状态上控制律和值函数的局部近似模型。本章还研究了轨迹优化中存在的约束的处理、振铃现象、稳态值函数近似和局部最优等问题。最后，总结了采用轨迹库实现控制的方法，包括轨迹跟踪法、在线优化法和函数近似法。与其他方法相比，采用控制律局部近似模型实现反馈控制的优点在于：控制律局部近似模型可以在轨迹优化的同时得到；在线查询的计算量小，适合实时控制；不需要事先确定表示控制律函数的基函数，控制律函数的表示更加灵活；构造状态反馈控制律所需的轨迹数目较少等。

第三章 基于轨迹库的仿人机器人站立平衡控制

站立平衡是指仿人机器人站立时在受到外界扰动下保持平衡，是仿人机器人运动控制研究的基本问题。对站立平衡控制的研究不仅有利于提高仿人机器人的环境适应能力，还有助于加深人类对自身平衡控制的理解 [7]。仿人机器人站立平衡控制的难点在于：1) 仿人机器人由很多关节和身体部分构成，而且大扰动下上身运动范围比较大，使系统动力学维数高、非线性明显；2) 仿人机器人站立时脚下的支撑区域比较小，踝关节在平衡中能够提供的力矩非常有限，而且地面作用力存在其他约束 [23]；3) 机器人关节角、关节角速度和关节力矩存在的一定的限制，进一步增加了控制器设计的难度；4) 设计对于特定优化指标最优的站立平衡控制更是一种挑战。

虽然目前对仿人机器人站立平衡控制方面进行了大量研究，但仍存在许多问题尚待解决。例如，这些研究大多集中在凭设计者经验设计的参数化控制器上。参数化控制器不仅性能上受到设计者水平、经验的制约，而且控制器参数的调节也比较困难。以往的研究对指定性能指标下最优的站立平衡控制研究较少。目前采用最优控制实现的站立平衡控制方法生成控制器的计算时间非常可观，而且只能处理瞬间扰动。以往的站立平衡控制研究主要考虑了瞬间扰动下的站立平衡问题，而忽视了持续扰动下的站立平衡控制问题。对于实际的仿人机器人站立平衡控制问题，模型的物理参数辨识、状态在线估计是不可避免的，但实际研究报道较少。

针对这些问题，本章研究基于优化的仿人机器人站立平衡控制方法。针对实际的机器人控制问题，将研究两连杆倒立摆模型物理参数辨识问题、基于扩展卡尔曼滤波的状态估计方法及滤波器参数对状态估计的影响。针对现有站立平衡控制方法存在的不足，提出基于轨迹库的仿人机器人站立平衡控制方法，

3.1 机器人模型

本文研究了前向平面内的站立平衡问题，将仿人机器人建模为多连杆刚体系统，外界扰动建模为作用在上身某处的恒定的水平力，力的大小为 p ，位置 r 为作用点与髋关节的距离。机器人的双脚虽然没有固定在地面上，但如果在控制的过程中始终保持相对地面静止，则机器人可以建模为多连杆倒立摆。本文采用两连杆倒立摆模型和四连杆倒立摆模型，如图 3.1(a) 和 3.1(b) 所示。为

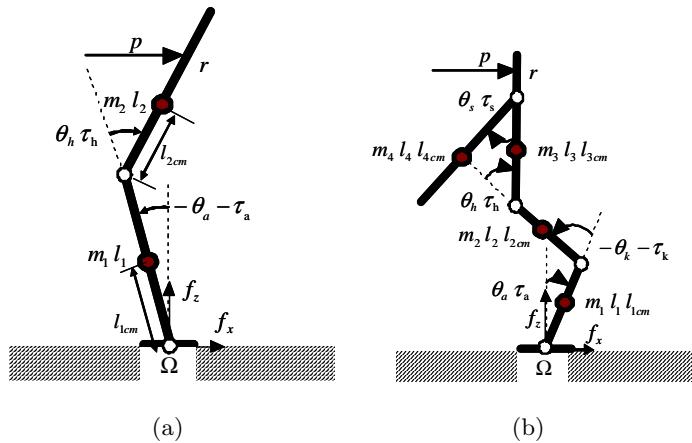


图 3.1 a) 两连杆倒立摆模型 b) 四连杆倒立摆模型

Fig. 3.1 a) Two-link inverted pendulum model b) Four-link inverted pendulum model

为了描述方便, 定义两连杆倒立摆模型关节角向量为 $\mathbf{q} \triangleq (\theta_a, \theta_h)^\top$, 控制向量为 $\mathbf{u} \triangleq (\tau_a, \tau_h)^\top$, 四连杆倒立摆模型关节角向量为 $\mathbf{q} \triangleq (\theta_a, \theta_k, \theta_h, \theta_s)^\top$, 控制向量为 $\mathbf{u} \triangleq (\tau_a, \tau_k, \tau_h, \tau_s)^\top$, 其中下标 a 表示脚踝关节, k 表示膝关节, h 表示髋关节, s 表示肩关节。

采用拉格朗日方法, 或者直接利用商业软件, 如 SDFAST, 得到机器人的动力学方程

$$\ddot{\mathbf{q}} = a(\mathbf{q}, \dot{\mathbf{q}}) + b(\mathbf{q})\mathbf{u} + c(\mathbf{q}, r)p \quad (3.1)$$

可以写成状态方程的形式

$$\dot{\mathbf{x}} \triangleq \frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = f(\mathbf{x}, \mathbf{u}, p, r) \quad (3.2)$$

其中 $\mathbf{x}^\top \triangleq (\mathbf{q}^\top, \dot{\mathbf{q}}^\top)$ 。

3.2 物理参数辨识

在仿真和生成控制器的过程中, 需要利用机器人的动力学模型, 因此需要确定机器人的运动学参数和动力学参数。运动学参数包括连杆长度和关节位置。动力学参数包括连杆的质量、惯量矩、惯性积和质心位置。运动学参数通常可以通过测量直接得到, 精度比较高。机器人的动力学参数可以通过机器人机械结构设计数据计算得到, 实际当中可以利用机械辅助设计软件根据各个部件的几何尺寸和所采用的材料计算。但是辅助设计软件在建模的过程中引入了很多几何简化, 没有考虑很多复杂的动态效应, 如关节摩擦等, 因此计算得到的数

据通常不够精确。机器人在安装以后往往会附加其他的部件，如分布在机器人身体各处的传感器、驱动机器人的输油管和电缆等。这些部件的加入会进一步改变机器人的动力学参数。因此在实验之前必须通过辨识得到机器人的物理参数。本节介绍机器人物理参数辨识的方法和实验结果。

3.2.1 物理参数辨识方法

假测量的次数为 P ，有 N_{par} 个需要辨识的物理参数，构成 $N_{par} \times 1$ 的向量 $\phi \triangleq (\phi_1, \phi_2, \dots, \phi_{N_{par}})^\top$ 。这些参数可能在机器人的模型中呈现线性或者非线性：

$$\text{线性模型: } \mathbf{y}^l = \mathbf{A}^l(\mathbf{x}^l)\phi \quad (3.3)$$

$$\text{非线性模型: } \mathbf{y}^l = \mathbf{f}(\mathbf{x}^l, \phi) \quad (3.4)$$

其中 $\mathbf{y}^l \triangleq (y_1^l, y_2^l, \dots, y_M^l)^\top$ 为 $M \times 1$ 的输出向量， $\mathbf{x}^l \triangleq (x_1^l, x_2^l, \dots, x_N^l)^\top$ 为 $N \times 1$ 的输入向量， $\mathbf{f} \triangleq (f_1, f_2, \dots, f_M)^\top$ 为 $M \times 1$ 的非线性函数，上标代表第 $l = 1, 2, \dots, P$ 次测量。

对于线性模型，将 P 次测量的方程放在一起组成

$$\mathbf{y} = \mathbf{A}\phi, \quad (3.5)$$

其中 $\mathbf{y} \triangleq \{\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^P\}$ 为 $MP \times 1$ 所有输出构成的向量， $\mathbf{A} = \{\mathbf{A}^1, \mathbf{A}^2, \dots, \mathbf{A}^P\}$ 为 $MP \times N_{par}$ 的矩阵。最小二乘参数估计

$$\phi = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{y}, \quad (3.6)$$

其中 \mathbf{A} 称为回归矩阵。

对于非线性模型，可以采用优化的方法，比较典型的如高斯—牛顿方法，进行参数估计。将 \mathbf{x}^l 带入公式 (3.4) 中的非线性函数，并将其表示为 \mathbf{f}^l 。通过最小化预测输出和实际测量输出之间的误差来估计参数 ϕ ，

$$\min_{\phi} \sum_{l=1}^P (\mathbf{y}^l - \mathbf{f}^l)^\top \mathbf{W} (\mathbf{y}^l - \mathbf{f}^l), \quad (3.7)$$

其中 \mathbf{W} 是加权矩阵。

因为仿人机器人特有的结构布局，使得接近某些部分的运动受到限制。同时因为某些关节没有安装力/力矩传感器，导致有些连杆的参数可能无法辨识。对于非线性的参数估计，参数的初值对最终的结果会有很大影响。

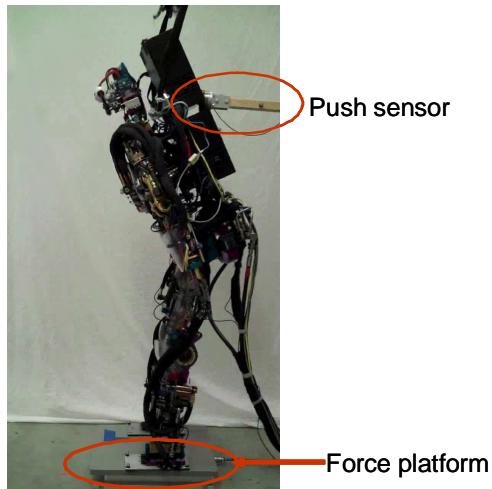


图 3.2 物理参数辨识实验配置

Fig. 3.2 The experiment configuration for physical parameter identification

3.2.2 两连杆倒立摆模型物理参数辨识方法

实验中采用两连杆倒立摆模型设计控制器。通常情况下，三维空间中的一个连杆有 10 个需要估计的动力学参数：质量，质心位置向量，3 个惯量矩和 3 个惯性积。所以三维空间中一个具有 n 个连杆的机器人需要估计 $10n$ 个动力学参数。对于平面两连杆模型，每个连杆有 4 个需要估计的动力学参数：质量，质心位置向量和一个惯性矩。在实验的过程中有些液压管连接到实验中的仿人机器人，这些管路对机器人产生了一定的外力。我们假设这些外力在实验中保持不变，并将外力的大小和作用位置作为需要辨识的参数。这样对于平面两连杆倒立摆模型，需要辨识的参数共 12 个。

定义需要估计的参数为

$$\phi \triangleq (m_1, m_2, x_{cm1}, z_{cm1}, x_{cm2}, z_{cm2}, MoI_1, MoI_2, x^{ext}, z^{ext}, f_x^{ext}, f_z^{ext})^T, \quad (3.8)$$

其中 m_1 和 m_2 为各连杆的质量、 x_{cm1} , z_{cm1} , x_{cm2} 和 z_{cm2} 为质心位置、 MoI_1 和 MoI_2 为各连杆的惯量矩、 f_x^{ext} 和 f_z^{ext} 为外力的水平和垂直分量大小、 x^{ext} 和 z^{ext} 为外力作用点的位置。两连杆倒立摆模型物理参数辨识的实验环境配置如图 3.2 所示。实验中关节角， \mathbf{q} ，可以通过测量直接得到，而关节角速度， $\dot{\mathbf{q}}$ ，和关节角加速度， $\ddot{\mathbf{q}}$ ，则可以采用数值方法通过关节角测量值来构造。因为是离线构造角速度和角加速度，所以可以采用双向滤波来实现。输入向量为 $\mathbf{x} = \{\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, p, r\}$ ，其中 p 和 r 分别为推力的大小和推力作用点离髋关节的距离，实验中分别通过推力测量装置直接测量和手工直接测量。输出定义为

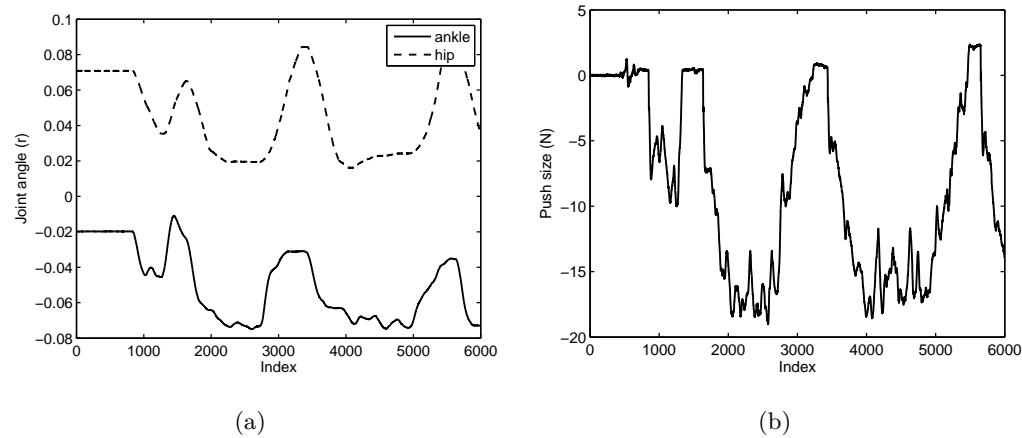


图 3.3 物理参数辨识 a) 关节角测量值 b) 推力大小测量值

Fig. 3.3 Physical parameter identification a) Measurements of the joint angles b)
Measurement of the push size

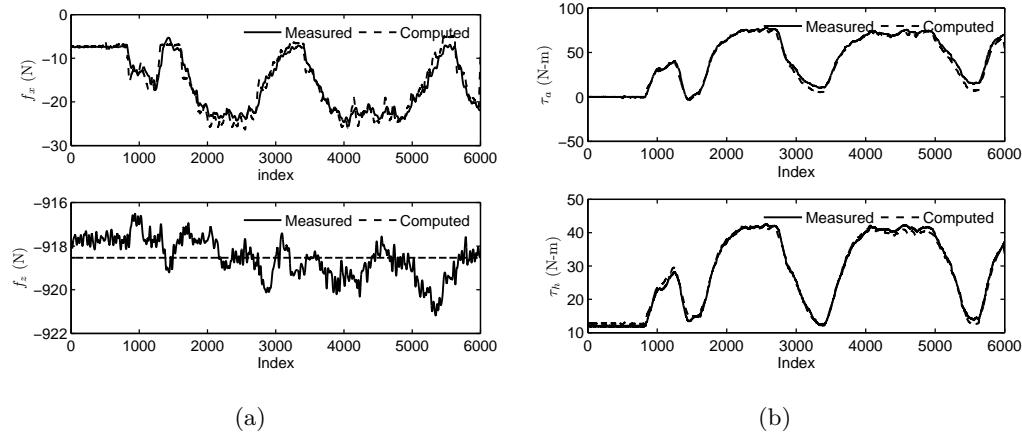


图 3.4 物理参数辨识输出拟合结果 a) 地面作用力 b) 关节力矩

Fig. 3.4 Output fitting results of physical parameter identification a) Ground reaction forces b) Joint torques

$\mathbf{y} = \{\tau_a, \tau_h, f_x, f_z\}$, 其中 τ_a 和 τ_h 为踝关节和髋关节力矩, 通过关节处的力矩传感直接测量。 f_x 和 f_z 为地面作用力水平分量和垂直分量, 通过机器人脚下的测力平台直接测量。 \mathbf{f} 采用 SDFAST 软件进行逆动力学分析得到。在实验中, 通过推力测量装置产生不同大小和持续时间的推力, 同时记录关节角、地面作用力、推力大小和位置的测量值, 然后通过非线性最小二乘方法辨识物理参数。

3.2.3 两连杆倒立摆模型物理参数辨识结果

根据各输出的取值范围,选取 $\mathbf{W} = \text{diag}(1, 1, 1, 0.01^2)$ 。根据机器人的设计手册和实际测量,选取

$$\phi_0 = \{56.40, 37.36, 0.14, 0.56, -0.07, 0.62, 1.24, 1.73, 0.01, 0 - 8.49, 3.81\}$$

作为参数估计的初值。图 3.3 为辨识中所用的关节角和推力的测量值。通过辨识得到的物理参数为

$$\phi = \{56.06, 38, 0.14, 0.57, -0.06, 0.61, 1.24, 1.73, -0.01, 0.12, -7.33, 4.11\},$$

图 3.4 为采用这个物理参数对输出拟合的结果。

3.3 状态在线估计

本文提出的基于轨迹库的站立平衡控制需要利用全状态信息,但实际的机器人缺少关节角速度传感,而且外力的大小和位置不容易测量,所以在控制中必须在线估计当前的状态,本节研究了采用扩展卡尔曼滤波器在线估计系统状态的方法。

3.3.1 扩展卡尔曼滤波方法

因为要估计的系统为非线性,所以采用了扩展卡尔曼滤波方法。扩展卡尔曼滤波是卡尔曼滤波对非线性系统的扩展。扩展卡尔曼滤波在状态估计时将非线性系统在当前状态附近线性化,在估计的过程中采用了一种反馈形式:在某一时刻对状态进行估计,在得到测量值后进行反馈。正因为如此,扩展卡尔曼滤波器分成了两个部分:时间更新方程和测量更新方程。时间更新方程负责根据系统当前状态和误差协方差估计向前得到状态的先验估计。测量更新方程实现反馈,即采用新的测量值和状态的先验估计得到状态的后验估计。

将关节角、关节角速度和推力的大小和位置一起构成一个新的状态变量, $\mathbf{x}^e \triangleq (\mathbf{x}^\top, p, r)^\top$, 得到系统的状态转移模型

$$\mathbf{x}^e(i+1) = \begin{bmatrix} F(\mathbf{x}(i), \mathbf{u}(i), p, r) \\ p \\ r \end{bmatrix} + \mathbf{w} \quad (3.9)$$

$$\triangleq F^e(\mathbf{x}^e(i), \mathbf{u}(i)) + \mathbf{w} \quad \mathbf{w} \sim N(0, \mathbf{S}) \quad (3.10)$$

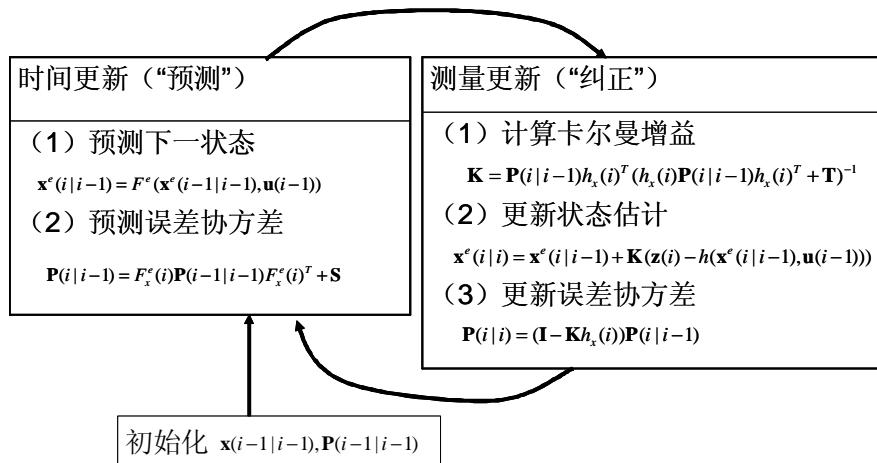


图 3.5 扩展卡尔曼滤波器工作流程

Fig. 3.5 A complete picture of the operation of the extended Kalman filter

观测模型为：

$$\mathbf{z}(i) = h(\mathbf{x}(i), \mathbf{u}(i)) + \mathbf{v} \quad \mathbf{v} \sim N(0, \mathbf{T}), \quad (3.11)$$

其中 F 是连续时间状态方程 (3.2) 对应的离散形式， \mathbf{w} 和 \mathbf{v} 为互不相关的噪声项， \mathbf{S} 和 \mathbf{T} 分别为过程噪声协方差矩阵和测量噪声协方差矩阵。

扩展卡尔曼滤波在测量之前预测下一状态：

$$\mathbf{x}^e(i|i-1) = F^e(\mathbf{x}^e(i-1|i-1), \mathbf{u}(i-1)) \quad (3.12)$$

$$\mathbf{P}(i|i-1) = F_x^e(i)\mathbf{P}(i-1|i-1)F_x^e(i)^T + \mathbf{S} \quad (3.13)$$

在得到新的测量值以后更新状态估计：

$$\mathbf{K} = \mathbf{P}(i|i-1)h_x(i)^T(h_x(i)\mathbf{P}(i|i-1)h_x(i)^T + \mathbf{T})^{-1} \quad (3.14)$$

$$\mathbf{x}^e(i|i) = \mathbf{x}^e(i|i-1) + \mathbf{K}(\mathbf{z}(i) - h(\mathbf{x}^e(i|i-1), \mathbf{u}(i))) \quad (3.15)$$

$$\mathbf{P}(i|i) = (\mathbf{I} - \mathbf{K}h_x(i))\mathbf{P}(i|i-1), \quad (3.16)$$

其中上标 $n|m$ 表示基于时刻 m (包括时刻 m) 之前的测量得到的时刻 n 的估计， \mathbf{K} 为卡尔曼增益矩阵， \mathbf{P} 为状态估计协方差矩阵，

$$F_x^e(i) = \frac{\partial F^e}{\partial \mathbf{x}^e} \Big|_{\mathbf{x}^e(i-1|i-1), \mathbf{u}(i-1)} \quad (3.17)$$

$$h_x(i) = \frac{\partial h}{\partial \mathbf{x}^e} \Big|_{\mathbf{x}^e(i-1|i-1), \mathbf{u}(i-1)} \quad (3.18)$$

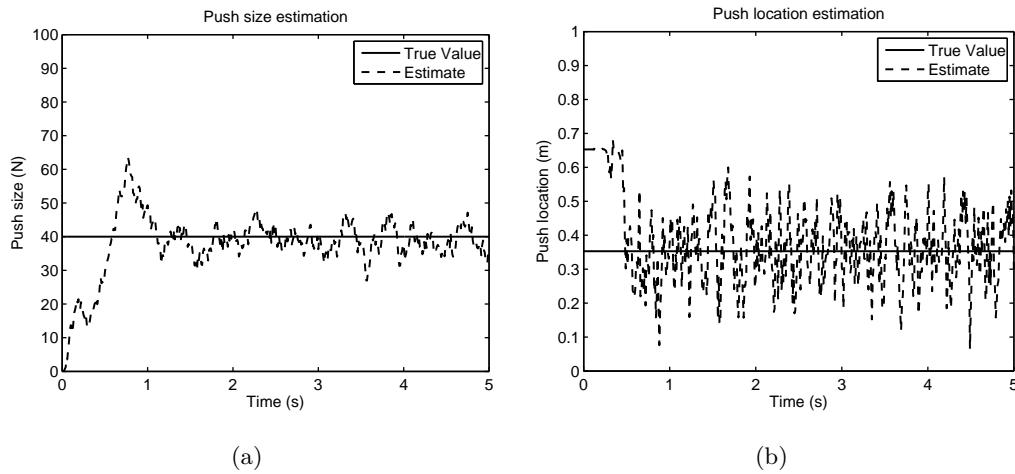


图 3.6 采用关节角测量值的状态估计结果 a) 推力大小估计 b) 推力位置估计

Fig. 3.6 State estimation using the joint angle measurements a) Push size estimation b)
Push location estimation

图 3.5 显示了完整的扩展卡尔曼滤波流程。

与卡尔曼滤波器不同，扩展卡尔曼滤波并不是最优估计。而且如果状态的最初估计与真实状态相差较大，或者系统建模不正确，也会导致扩展卡尔曼滤波器发散。在实际应用中，测量噪声协方差矩阵可以通过实际测量得到。因为需要估计的状态往往不能直接测量，所以过程噪声协方差的选取相对比较复杂。

3.3.2 状态估计方法 A

实验中关节角比较容易测量，所以我们首先研究了只利用关节角测量值对当前状态进行估计的方法，此时采用的观测量为 $\mathbf{z} \triangleq (\bar{\theta}_a, \bar{\theta}_h)^T$ ，其中 $\bar{\theta}_a$ 和 $\bar{\theta}_h$ 为踝关节和髋关节的关节角测量值，此时，雅可比矩阵为常数矩阵， $h_x = [I_{2 \times 2} \quad 0_{2 \times 4}]$ 。

选取 10^{-4} 作为关节角和关节角速度的过程协方差，选取 $s_p = 4$ 作为推力大小的过程噪声协方差， $s_r = 0.01$ 作为推力位置的过程噪声协方差，选取 10^{-4} 作为所有测量的噪声协方差，图 3.6 为推力大小和推力位置的估计结果。虽然推力的大小和推力的位置的估计值最终收敛到实际值，但是收敛所需要的时间比较长，例如推力大小估计的收敛时间接近 1.5 秒。仿真结果说明只采用关节角测量进行状态估计的效果不太理想。

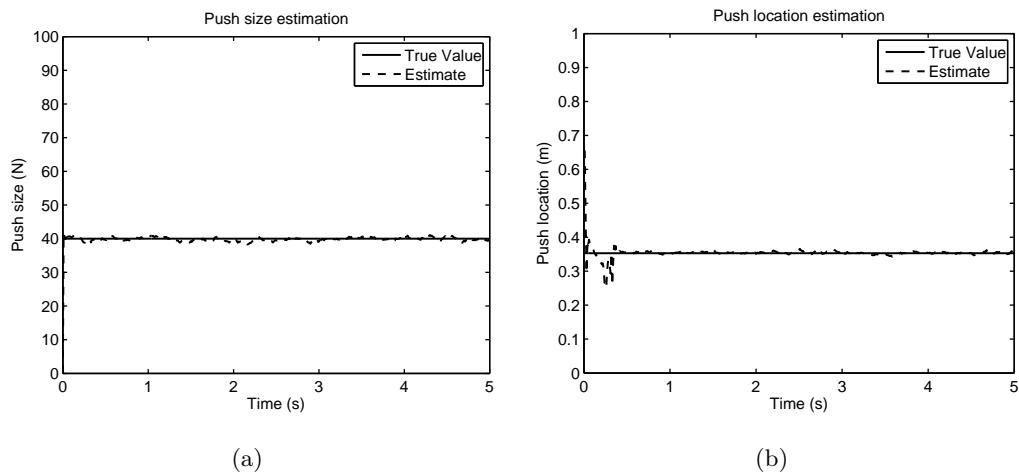


图 3.7 采用关节角和地面作用力测量值的状态估计结果 a) 推力大小估计 b) 推力位置估计

Fig. 3.7 State estimation using the measurements of joint angles and ground reaction forces a) Push size estimation b) Push location estimation

3.3.3 状态估计方法 B

为了提高状态估计的收敛速度，尝试通过增加观测变量的方法。将机器人放置在一个力测量平台上，采用力测量平台测量地面作用力。采用新的观测变量， $\mathbf{z} \triangleq (\bar{\theta}_a, \bar{\theta}_h, \bar{f}_x, \bar{f}_z)^T$ ，其中 $\bar{\theta}_a$ 和 $\bar{\theta}_h$ 为踝关节和髋关节的关节角测量值， \bar{f}_x 和 \bar{f}_z 为地面作用力水平和竖直分量的测量值。此时观测模型成为：

$$\begin{aligned} \mathbf{z} = & \begin{bmatrix} \theta_a \\ \theta_h \\ f_x(\mathbf{x}, \mathbf{u}, p, r) \\ f_z(\mathbf{x}, \mathbf{u}, p, r) \end{bmatrix} + \mathbf{v} \\ \triangleq & h(\mathbf{x}^e, \mathbf{u}) + \mathbf{v} \quad \mathbf{v} \sim N(0, \mathbf{T}) \end{aligned} \quad (3.19)$$

其中 $f_x(\mathbf{x}, \mathbf{u}, p, r)$ 和 $f_z(\mathbf{x}, \mathbf{u}, p, r)$ 采用 SDFAST 软件的逆动力学分析计算。

仿真中仍然采用 10^{-4} 作为所有测量的噪声协方差，采用 10^{-4} 作为关节角和关节角速度的过程噪声协方差，采用 $s_p = 4$ 作为推力大小过程噪声协方差， $s_r = 0.1$ 作为推力位置过程噪声协方差，图 3.7 为推力估计的结果。与采用关节角测量值的推力估计相比，引入地面作用力测量值以后明显的提高了推力估计的收敛速度。

下面研究过程噪声协方差的选取对推力估计的影响。选取 $s_p = 0.1$ 作为推力大小过程噪声协方差， $s_r = 0.01$ 作为推力位置过程噪声协方差， 10^{-4} 作为

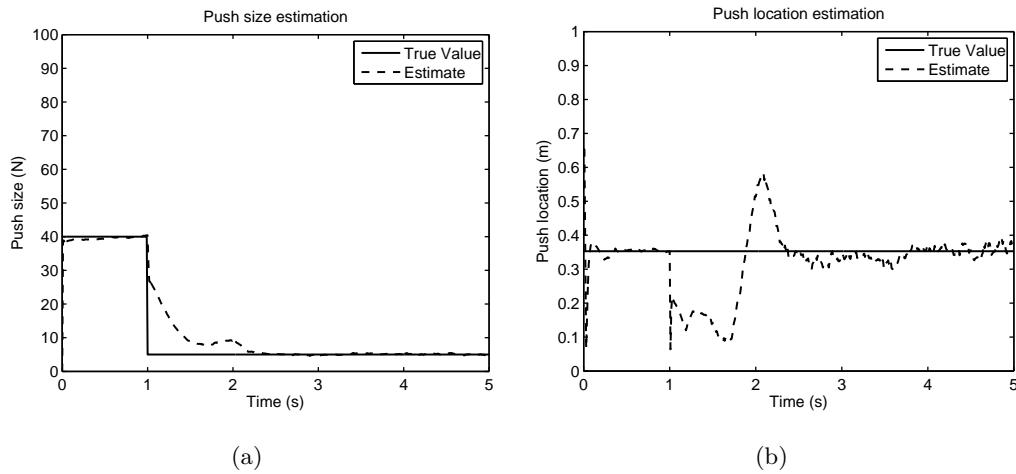
图 3.8 $s_p = 0.1$ 和 $s_r = 0.01$ 状态估计结果 a) 推力大小估计 b) 推力位置估计

Fig. 3.8 State estimation using $s_p = 0.1$ and $s_r = 0.01$ a) Push size estimation b) Push location estimation

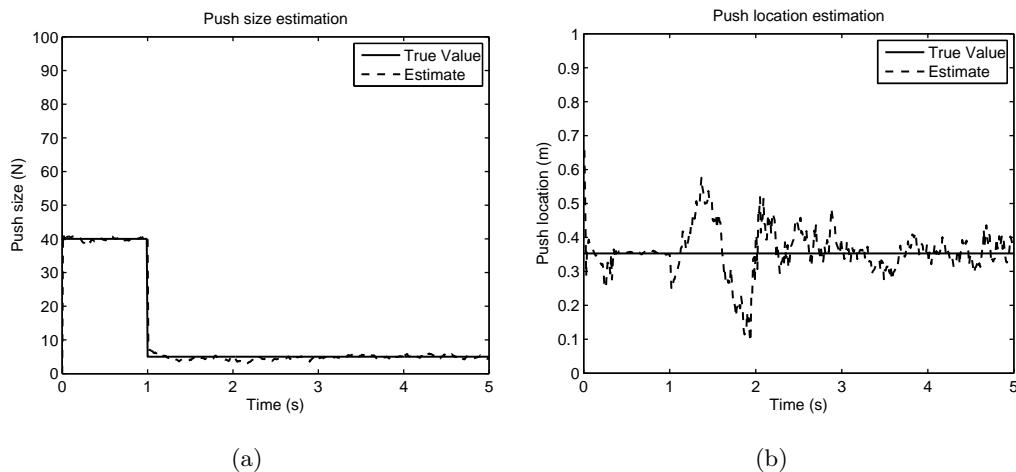
图 3.9 $s_p = 1$ 和 $s_r = 0.01$ 状态估计结果 a) 推力大小估计 b) 推力位置估计

Fig. 3.9 State estimation using $s_p = 1$ and $s_r = 0.01$ a) Push size estimation b) Push location estimation

关节角和关节角速度的过程协方差，对变化的推力进行估计。图 3.8 为推力大小和位置的估计结果。而选取 $s_p = 1, s_r = 0.01, 10^{-4}$ 作为关节角和关节角速度的过程协方差，对同样的推力进行估计。图 3.9 为推力大小和位置的估计的结果。以上的仿真结果说明采用比较小的过程协方差能够减小估计结果中的噪声，但同时会降低状态估计收敛的速度。权衡噪声和收敛速度，最终选取 $S = \text{diag}(0.01^2, 0.01^2, 0.01^2, 0.01^2, 1, 0.01)$ 和 $T = \text{diag}(0.01^2, 0.01^2, 0.01^2, 0.01^2)$ 。

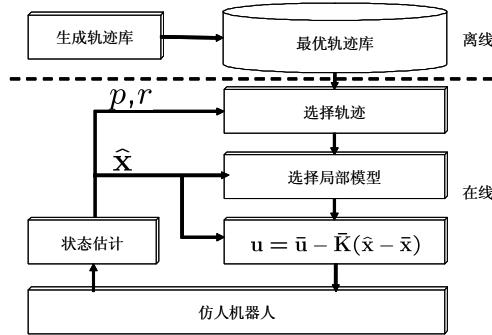


图 3.10 基于轨迹库的站立平衡控制结构

Fig. 3.10 Standing balance control architecture using a trajectory library

3.4 基于轨迹库的站立平衡控制方法

针对现有方法存在的问题，本章提出了基于轨迹库的站立平衡控制方法。采用轨迹优化针对站立平衡中可能遇到的各种不同情况，如推力的不同大小和不同作用位置，生成最优轨迹。在控制的过程中，采用最近邻控制实现最优轨迹邻域内的最优的控制。为了得到更大状态空间上的状态反馈控制器，生成大量的最优轨迹。为满足控制性能，同时减少轨迹库中轨迹的数目，提出了初始条件自适应网格上的轨迹库生成方法。

基于轨迹库的站立平衡控制结构如图 3.10。首先采用轨迹优化离线地生成最优轨迹和最优控制律的局部近似模型构成轨迹库；在控制中，根据当前推力的大小 p 和作用位置 r 的估计 (\hat{p}, \hat{r}) 从轨迹库中选择最接近的轨迹，再根据系统当前状态的估计 \hat{x} 选择轨迹采样状态上最接近的局部近似模型并由此计算当前的控制 u ，从而实现状态反馈控制。

3.4.1 轨迹优化

单步代价函数定义为

$$L(\mathbf{x}, \mathbf{u}, p, r) \triangleq \tilde{\mathbf{x}}^T \mathbf{Q} \tilde{\mathbf{x}} T + \tilde{\mathbf{u}}^T \mathbf{R} \tilde{\mathbf{u}} T, \quad (3.20)$$

其中 $\tilde{\mathbf{x}} \triangleq \mathbf{x} - \mathbf{x}_d(p, r)$, $\tilde{\mathbf{u}} \triangleq \mathbf{u} - \mathbf{u}_d(p, r)$, T 为积分步长, \mathbf{Q} 和 \mathbf{R} 是相应维数的加权矩阵, \mathbf{x}_d 和 \mathbf{u}_d 分别是大小为 p 、作用位置为 r 的外力作用下的期望状态和期望驱动力矩。

在轨迹规划时，需要考虑站立平衡的动力学约束和机器人自身的限制。为了保持支撑脚相对地面静止，在控制过程中需要始终满足脚与地面接触的物

理约束。为防止支撑脚相对地面打滑，地面作用力必须始终处于摩擦力锥体内 [24]:

$$\left| \frac{f_x}{f_z} \right| \leq \mu, \quad (3.21)$$

其中 f_x 为水平方向的摩擦力， f_z 为垂直方向的地面作用力， μ 为脚与地面之间的干摩擦系数。为防止支撑脚意外翻转，需要满足 [23]:

$$-\frac{\tau_a}{f_z} \in \Omega, \quad (3.22)$$

其中 τ_a 为支撑脚踝关节驱动力矩， Ω 为支撑脚的支撑区域。因此公式 (3.21) 和 (3.22) 一起构成关于轨迹优化中 \mathbf{x} 和 \mathbf{u} 的路径约束。除此之外将机器人自身关节角和关节角速度的限制作为状态变量约束，将关节驱动力矩限制作为控制变量约束。

3.4.2 最近邻控制

微分动态规划 (Differential Dynamic Programming) 在生成最优轨迹的同时得到轨迹邻域内的状态反馈控制律:

$$\mathbf{u}(i) = \mathbf{u}^*(i) - \mathbf{K}(i)(\mathbf{x}(i) - \mathbf{x}^*(i)), \quad (3.23)$$

其中 $\mathbf{x}^*(i)$ 和 $\mathbf{u}^*(i)$ 分别为最优轨迹的采样状态和对应的控制量，

$$\mathbf{K}(i) = -\frac{\partial \mathbf{u}^*}{\partial \mathbf{x}} \Big|_{\mathbf{x}^*(i)}, \quad (3.24)$$

其中 $\frac{\partial \mathbf{u}^*}{\partial \mathbf{x}} \Big|_{\mathbf{x}^*(i)}$ 为最优状态反馈控制律在采样状态处的梯度。

对于无限时域的最优控制，值函数和最优控制律是关于状态的稳态函数。我们将站立平衡控制问题转化为无限时域的最优控制问题，因此公式 (3.23) 将成为采样状态上控制律不变局部近似模型。为利用这些局部近似模型实现状态反馈控制，最简单的方法是利用最接近的控制律局部近似模型预测当前状态对应的最优控制。为此，首先定义跟推力有关的距离，

$$(\Delta p, \Delta r) \mathbf{D}_1 (\Delta p, \Delta r)^T, \quad (3.25)$$

然后定义跟状态有关的距离，

$$\Delta \mathbf{x}^T \mathbf{D}_2 \Delta \mathbf{x}, \quad (3.26)$$

表 3.1 站立平衡中可能出现的情况

Table 3.1 Possible conditions during standing balance control

	q	\dot{q}	p
瞬间扰动后	0	\times	0
持续扰动	0	0	\times
持续扰动忽然后	\times	0	0
一般情况	\times	\times	\times

其中 Δp , Δr , Δx 分别表示当前外力的大小、位置和状态与轨迹库中保存的局部模型对应量之间的误差。

最近邻控制表示为

$$\mathbf{u} = \bar{\mathbf{u}} - \bar{\mathbf{K}}(\mathbf{x} - \bar{\mathbf{x}}), \quad (3.27)$$

其中 $\bar{\mathbf{u}}$, $\bar{\mathbf{x}}$ 和 $\bar{\mathbf{K}}$ 分别是距离当前状态最近的局部近似模型的控制、状态和增益矩阵。

3.4.3 初始条件自适应网格上的轨迹库生成方法

基于一条最优轨迹的控制器当状态远离这条轨迹时的性能会比较差。为了生成更大状态空间上的控制律，需要采用多条最优轨迹构成的轨迹库来实现。为在满足控制性能的同时减少轨迹库中轨迹的数目，本文提出了初始条件自适应网格上的轨迹库生成方法。

在生成轨迹库之前，我们首先研究了站立平衡控制中可能出现的情况（表 3.1，其中“ \times ”代表非零）。第一种情况，外力和关节角均为零而关节角速度非零，对应静止直立站立时受到瞬间扰动的情况；第二种情况，关节角和关节角速度为零而外力大小非零，对应静止直立站立时受到恒定外力作用瞬间的情况；第三种情况，关节角速度和外力大小为零而关节角非零，对应机器人在恒定外力作用下稳定后外力忽然消失瞬间的情况；第四种情况，关节角、关节角速度和外力的大小均不为零，对应站立平衡中最一般的情况。考虑到实际中出现比较多的情况，在生成控制器时目前只考虑了第二种情况（静止直立时受到大小和位置不同的恒定外力作用瞬间）中所有可能的 $\mathbf{x}(0)$, p , r , 将其作为轨迹优化的初始条件。为了提高控制器在其他情况下的性能，可以考虑更多的可能的初始条件，但这样做会增加控制器优化的时间。

图 3.11 为生成轨迹库的流程。首先以 $\mathbf{x}(0) = \mathbf{0}$, $p = 0$ 和 $r = 0$ 为初始条件优化轨迹，将得到的最优轨迹和控制律在轨迹采样状态上的局部近似模型保

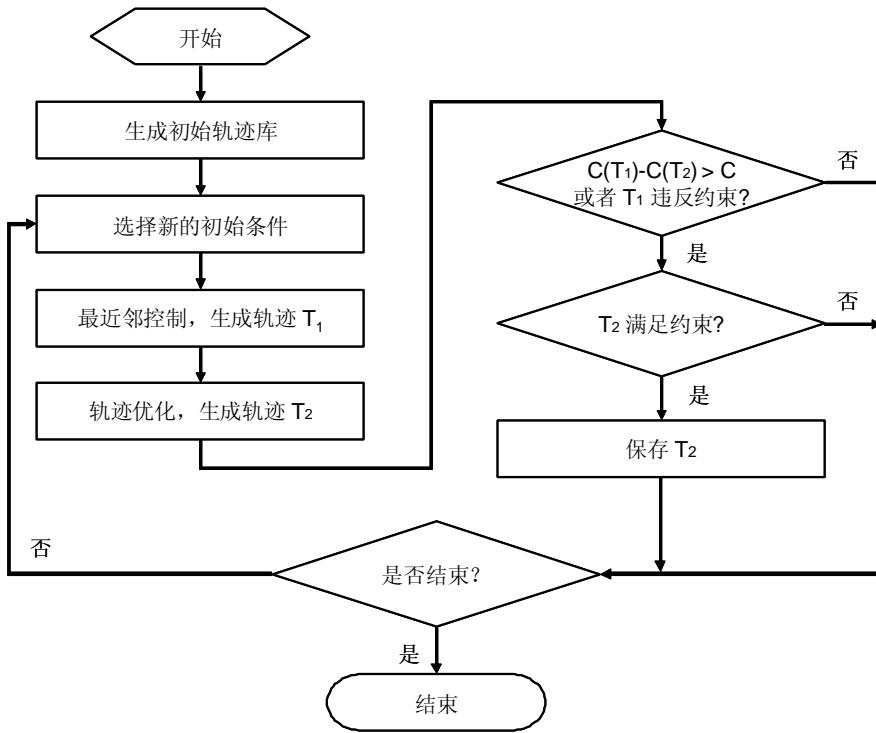


图 3.11 自适应网格上的轨迹库生成流程

Fig. 3.11 The flowchart of the trajectory library generation on an adaptive grid

存为初始轨迹库。此时的最近邻控制实际上构成了线性二次型调节器 (LQR)。然后在已保存的初始条件附近选择一个新的初始条件, $\{\mathbf{x}(0), p, r\}$, 应用当前轨迹库和最近邻控制, 在长度为 N 的时间域上生成新的轨迹, 表示为 T_1 。同样以 $\{\mathbf{x}(0), p, r\}$ 作为初始条件在长度为 N 的时间域上进行轨迹优化, 将得到的最优轨迹表示为 T_2 。比较两条轨迹的总代价 $C(T_1)$ 和 $C(T_2)$ 。如果两者之间的误差大于某个性能阈值 $C(T_1) - C(T_2) > C$ 或者轨迹 T_1 违反了约束, 并且如果 T_2 满足约束, 则将轨迹 T_2 连同它采样状态上的控制律局部近似模型保存进轨迹库。否则判断是否还有其他需要考虑的初始条件, 如果还有需要考虑的初始条件, 则继续上面的过程, 否则停止。

为了在生成轨迹库时选择一个新的初始条件, 我们将初始条件 $\{\mathbf{x}(0), p, r\}$ 离散化为均匀的网格, 然后从网格的某个容易优化的节点 (如节点 $\{0, 0, 0\}$) 开始, 然后逐渐考虑更多可能的初始条件。具体实现的方法是建立一个等候列表, 如图 3.12 所示。在选择等候列表最前面的内容作为初始条件的同时将其网格上临近的且需要考虑的节点追加到等候列表的后面。如果在优化 T_2 时无法得到满足约束的轨迹, 则当前节点及其相邻节点不加入等候列表。不断地处理等候

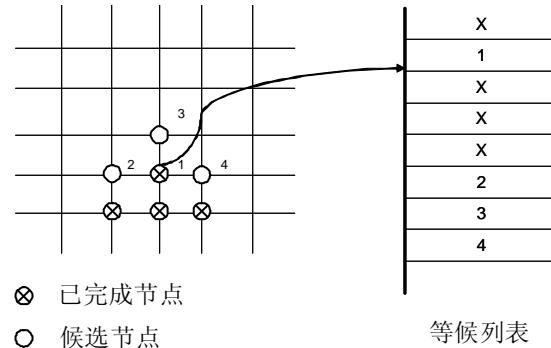


图 3.12 轨迹优化初始条件的选择

Fig. 3.12 The initial condition selection for trajectory optimization

表 3.2 两连杆倒立摆模型物理参数

Table 3.2 Physical parameters of the two-link inverted pendulum model

	连杆 1	连杆 2
质量 (kg)	19.474	29.492
角惯量 ($\text{kg}\cdot\text{m}^2$)	0.696	1.03356
长度 l (m)	0.661	0.653
质心位置 l_{cm} (m)	0.430	0.141

列表中的节点，直至等候列表为空。

参数化轨迹优化和微分动态规划都是局部优化方法，因此不能保证生成的轨迹为全局最优。如果不能保证所有的轨迹是全局最优，就不能保证来自不同轨迹的控制律局部近似模型来自同一个全局控制律。为了采用来自不同轨迹的控制律局部近似模型构造全局控制律，在生成轨迹库时应该保持轨迹之间的一致性，本文采用了下列手段：1) 从容易优化的初始条件开始生成初始轨迹库，然后考虑更大的状态空间；2) 在生成轨迹的过程中，采用基于当前轨迹库的控制器为起始于临近状态的轨迹优化生成初始轨迹；3) 采用小节 2.2.5.4 中的方法重新优化得到的轨迹库。如果局部近似模型描述的值函数处处满足 Bellman 方程，那么其对应的控制律将成为全局最优控制律 [92, 100]。

3.4.4 仿真和实验结果

3.4.4.1 仿真结果

我们将推力大小和推力位置离散化为均匀的网格，然后在自适应网格上生

表 3.3 四连杆倒立摆模型物理参数

Table 3.3 Physical parameters of the four-link inverted pendulum model

	连杆 1	连杆 2	连杆 3	连杆 4
质量 (kg)	5.78	13.69	20.95	8.54
角惯量 ($\text{kg} \cdot \text{m}^2$)	0.038	0.17	0.46	0.48
长度 l (m)	0.33	0.33	0.653	0.55
质心位置 l_{cm} (m)	0.19	0.20	0.18	0.34

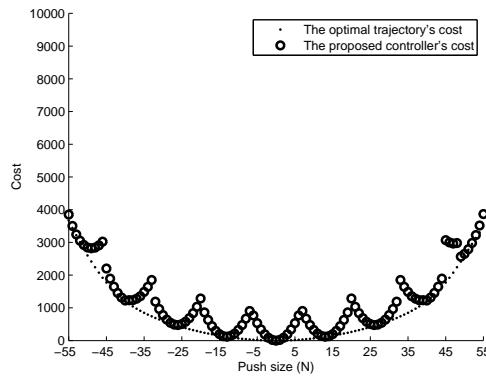


图 3.13 本文提出的控制器代价与最优轨迹代价的比较

Fig. 3.13 Cost comparison of the proposed controller with the optimal trajectory

成轨迹库。在轨迹优化时初始关节角和关节角速度全部设为零，推力大小 p 的量化步长取为 2 牛顿，推力位置 r 的量化步长取为 0.1 米。对于两连杆倒立摆模型，采用表 3.2 中的模型参数，根据变量的变化范围选取 $\mathbf{D}_1 = \text{diag}(10^2, 10^6)$, $\mathbf{D}_2 = \text{diag}(1, 1, 0.01, 0.01)$, $\mathbf{Q} = \mathbf{I}_4$, $\mathbf{R} = \mathbf{I}_2$ ，采用关节力矩最小的静平衡状态作为期望状态。因为 SNOPT 生成的轨迹作为 DDP 的初始轨迹，所以 SNOPT 采用 $N = 100$, $T = 0.05$ 秒；DDP 采用 $N = 500$, $T = 0.01$ 秒。因为时间步

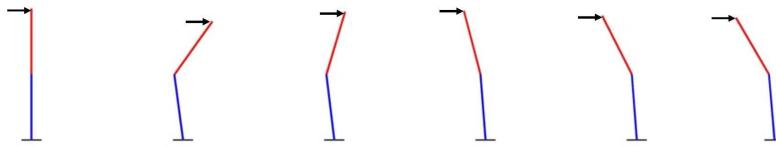


图 3.14 仿真机器人对作用在头部恒定的 42 牛顿前向推力的响应（截图时间间隔为 0.3 秒）。

Fig. 3.14 Our simulated robot responding to a constant forward push at the head of 42 Newtons. The frames are taken in intervals of 0.3 seconds.

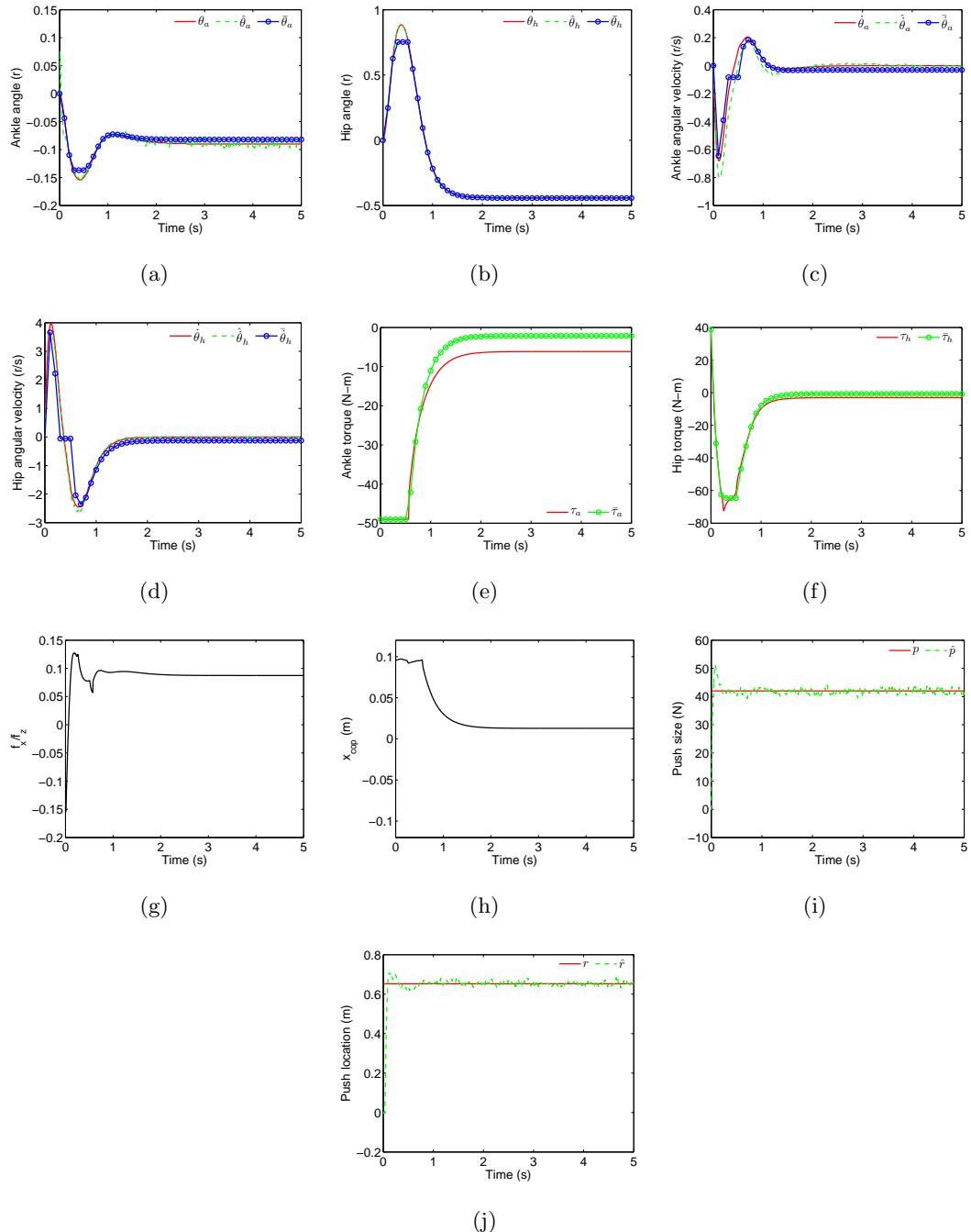


图 3.15 对作用在头部恒定的 42 牛顿前向推力的响应和状态估计结果

Fig. 3.15 Responses to a 42 Newtons forward push at the head and the state estimation results

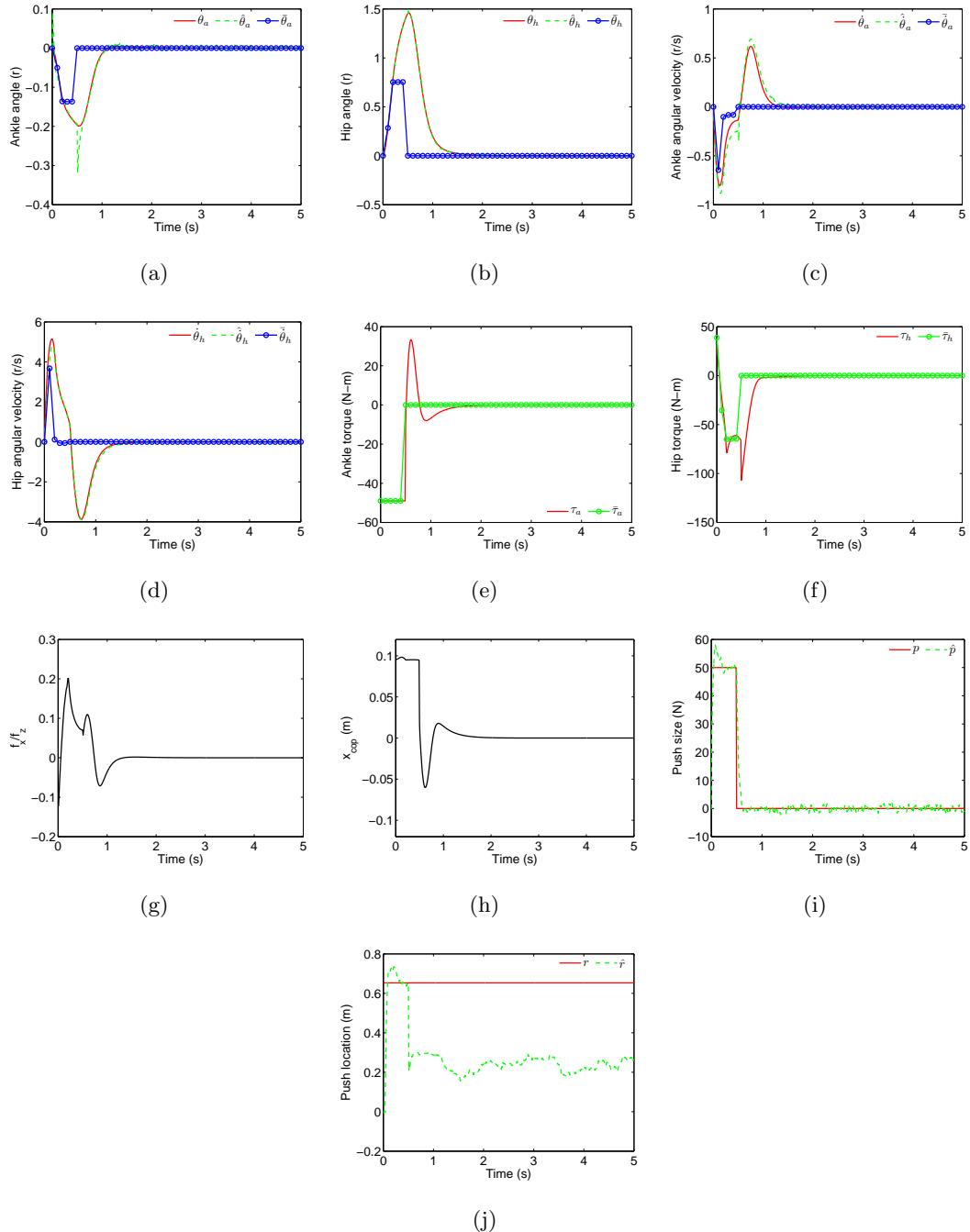


图 3.16 仿真机器人对作用在头部持续 0.5 秒的 50 牛顿前向推力的响应和状态估计结果
Fig. 3.16 Our simulated robot responding to a forward push at the head of 50 Newtons lasting 0.5 seconds and the state estimation results

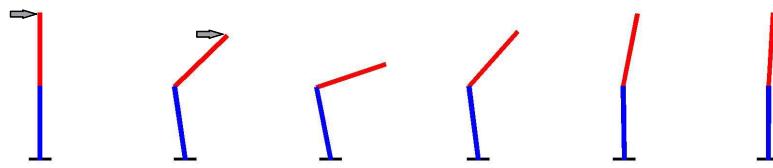


图 3.17 仿真中机器人对作用时间为 0.5 秒的 50 牛顿的前向推力的响应（截图时间间隔为 0.25 秒）

Fig. 3.17 Our simulated robot responding to a push of 50 Newtons lasting 0.5 seconds
(the configuration of the robot is drawn every 0.25 seconds)

长不同，SNOPT 生成的轨迹在用作 DDP 的初始轨迹之前需要进行插值。在 Intel(R) Xeon(TM) 3.20GHz 双核 CPU 和 2G 内存的工作站上，SNOPT 需要 6 秒而 DDP 需要 1 秒来优化一条轨迹，总共需要约 1 小时生成完整的轨迹库。对于给定的性能阈值 (1000)，最终生成的轨迹库中含有 21 条最优轨迹 (10500 个局部模型)。对于四连杆倒立摆模型，采用表 3.3 中的模型参数。SNOPT 需要 13 秒而 DDP 需要 30 秒来优化一条轨迹，总共需要约 3.5 小时生成轨迹库。最终生成的轨迹库包含 42 条最优轨迹 (21000 个局部模型)。

采用两连杆模型，对基于这个轨迹库的控制器的代价和最优轨迹的代价进行了比较，如图 3.13 所示。如果轨迹库中存在和测试推力大小、位置相近的轨迹，则本文提出的控制的代价与跟踪最优轨迹的代价相近，否则本文提出的控制器的性能会有所下降。但是，因为轨迹库是基于控制器的预测性能生成的，所以本文提出的控制器的性能下降能够被限制在有限的范围内。

本文比较了所提出的站立平衡控制器的代价与最优轨迹的代价。对作用在上身 0.35 米处，间隔 1 牛顿的不同推力进行测试，控制代价的比较结果如图 3.13 所示。如果轨迹库中存在和测试推力大小、位置相近的轨迹，则本文提出的控制器的代价与最优轨迹的代价相近，否则本文提出的控制器的性能将有所下降。但因为轨迹库是基于控制器的预测性能生成的，所以控制器性能的下降能够被限制在有限的范围内。

应用基于轨迹库的控制器，对不同类型的推力进行了仿真。仿真结果中 θ_a , $\dot{\theta}_a$, $\ddot{\theta}_a$, τ_a 和 τ_h 表示机器人的关节角、关节角速度和关节力矩的真实值， $\hat{\theta}_a$, $\hat{\dot{\theta}}_a$, $\hat{\ddot{\theta}}_a$ 表示关节角、关节角速度的估计值， p 和 r 表示推力的大小和位置的真实值， \hat{p} 和 \hat{r} 表示其估计值。 $\bar{\theta}_a$, $\bar{\theta}_h$, $\bar{\dot{\theta}}_a$, $\bar{\dot{\theta}}_h$, $\bar{\tau}_a$ 和 $\bar{\tau}_h$ 表示所采用局部近似模型的关节角、关节角速度和关节力矩。

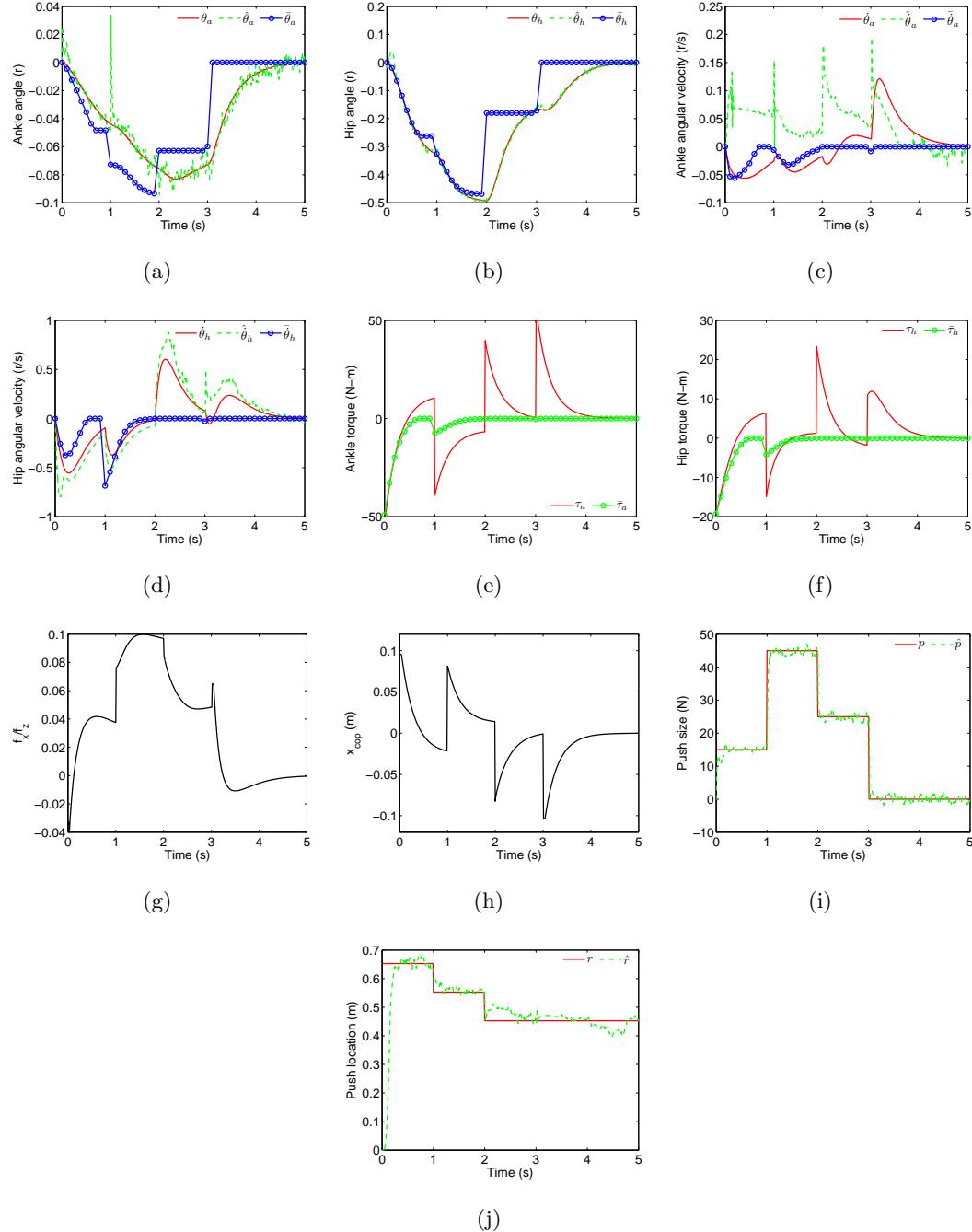


图 3.18 仿真机器人对作用在头部随机推力的响应和状态估计结果

Fig. 3.18 Our simulated robot responding to a random push sequence at the head and the state estimation results

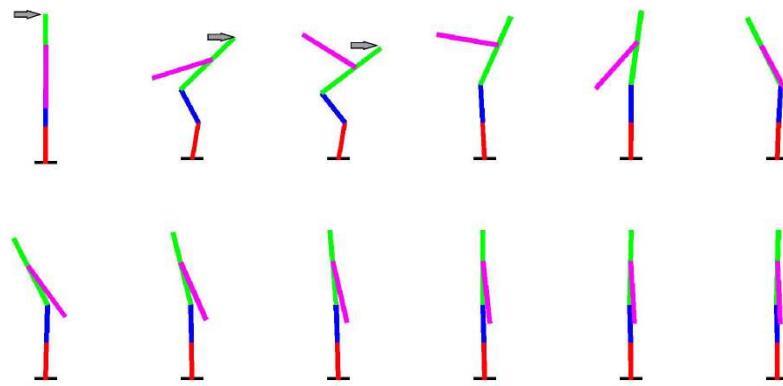


图 3.19 仿真中机器人对作用在头部持续 0.5 秒 50 牛顿推力的响应（截图时间间隔为 0.25 秒）

Fig. 3.19 Our simulated robot responding to a push at the head of 50 Newtons lasting for 0.5 seconds (the configuration of the robot is drawn every 0.25 seconds)

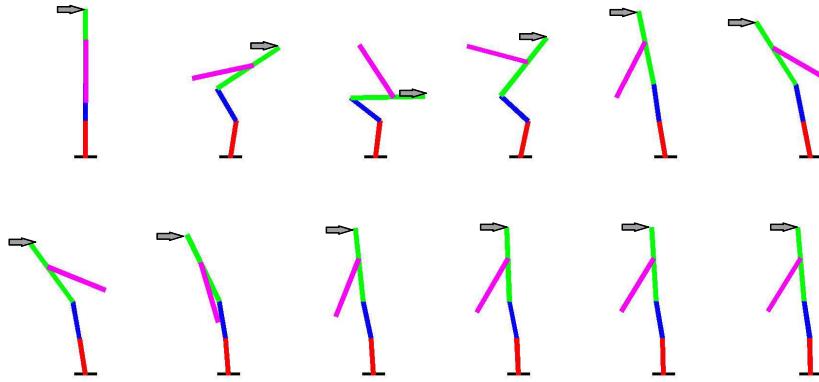


图 3.20 仿真中机器人对作用在头部的 55 牛顿恒定推力的响应（截图时间间隔为 0.25 秒）

Fig. 3.20 Our simulated robot responding to a constant push at the head of 55 Newtons (the configuration of the robot is drawn every 0.25 seconds)

图 3.14 为仿真机器人对作用在头部恒定的 42 牛顿的前向推力的姿态响应。机器人向前倾斜，然后倒向推力一边，用重力平衡外界推力。图 3.15 为机器人的关节角、关节角速度、关节力矩、地面作用力响应和推力估计结果。轨迹库中没有和这个推力完全相同的最优轨迹，在控制中实际采用了大小为 39.5 牛顿作用在同一位置的最优轨迹。图 3.15(g)和图 3.15(h) 说明最终得到的控制器响应能够满足支撑脚与地面之间的物理约束。图 3.15(i) 和 3.15(j) 说明推力大小和

位置估计能够迅速地逼近真实值。

图 3.17 为仿真机器人对大小为 50 牛顿、持续时间为 0.5 秒、作用在头部的水平推力的姿态响应。平衡控制过程中机器人向前倾斜，最终恢复直立。图 3.16 为机器人关节角、关节角速度、关节力矩、地面作用力响应和推力估计结果。图 3.16 (j) 中需要说明的是当推力的大小为零时，推力位置的估计没有实际意义。图 3.16 (g) 和 3.16 (h) 说明在控制过程中能够满足支撑脚与地面之间的物理约束，图 3.16 (i) 和 3.16 (j) 说明推力估计值仍然能够快速逼近真实值。

控制器的鲁棒性通过大小和位置随机变化的推力加以验证。图 3.18 为大小分别为 15、45 和 25 牛顿，作用位置变化的推力下的控制器响应。对于推力的大小和位置均不在轨迹库中的测试推力，本文提出的控制器仍然能够保持机器人平衡。

采用四连杆倒立摆模型，对作用在头部，大小为 50 牛顿，持续时间为 0.5 秒的推力下的站立平衡控制进行仿真。图 3.19 为机器人姿态响应。机器人下蹲、向前弯曲、摆臂并最终在外界扰动消失以后恢复直立。图 3.20 为仿真机器人对作用在头部，大小为 55 牛顿的恒定外力的响应。仿真机器人下蹲、向前弯曲、摆臂并最终向后倚向外力。采用两连杆倒立摆模型的仿真机器人能处理的作用在头部的最大恒定推力为 47 牛顿，而采用四连杆倒立摆模型的仿真机器人能处理的最大值为 55 牛顿，这一结果说明连杆数目的增加能够提高机器人的抗扰动能力。

本文同时设计了基于 LQR 的增益调度控制器 (LQR-based gain scheduling)，并进行了对比。LQR 增益调度控制器将系统在期望状态附近线性化，然后设计 LQR 控制器，在控制过程中根据当前推力的估计选择合适的 LQR 控制器。采用两连杆倒立摆模型，对于作用在头部的恒定推力，LQR 增益调度控制器能够处理的最大值为 36 牛顿，而本文提出的控制器能够处理的最大值为 47 牛顿。对于大小分别为 35, 30 和 20 牛顿，作用在头部的恒定推力，LQR 增益调度控制器的代价分别为 3565, 1272 和 496，而本文提出的控制器对应的代价为 2080, 1255 和 496。本文提出的控制器在能够处理的最大推力和代价方面优于基于 LQR 的增益调度控制器。

另外，本文设计的控制器与最优增益的全状态反馈控制器进行了对比。全状态反馈控制器的形式为

$$\tau_a = -k_1 \Delta \theta_a - k_2 \Delta \theta_h - k_3 \dot{\theta}_a - k_4 \dot{\theta}_h \quad (3.28)$$

$$\tau_h = -k_5 \Delta \theta_a - k_6 \Delta \theta_h - k_7 \dot{\theta}_a - k_8 \dot{\theta}_h, \quad (3.29)$$

其中 $\Delta\theta_a := \theta_a - \theta_{a,d}$, $\Delta\theta_h := \theta_h - \theta_{h,d}$ 分别表示当前关节角与期望关节角之间的偏差, $\theta_{a,d}$ 和 $\theta_{h,d}$ 分别为指定推力下的踝关节和髋关节的期望关节角。针对指定的推力和静止站立的初始状态, 采用与公式 3.20 相同的单步代价函数, 对公式 3.28 和 3.29 构成的全状态反馈中的反馈增益 $k_1 \sim k_8$ 进行优化, 得到指定推力下的最优反馈增益。对于作用在机器人头部, 大小分别为 20, 30 和 35 牛顿的前向水平推力, 最优增益的全状态反馈控制器的代价分别为 492, 1216 和 1999, 而本文提出的控制器能取得的最小代价分别为 492, 1106 和 1497。对比结果说明, 与最优增益的全状态反馈控制器相比基于轨迹的控制器能够取得更小的代价。

3.4.4.2 实验结果

我们在卡耐基-梅隆大学的 Sarcos 仿人机器人上对提出的站立平衡控制方法进行了实验。Sarcos 机器人身高约为 1.6 米, 体重约为 95 公斤。利用液压驱动器和关节处的应力传感器形成的反馈, 实现了对各关节力矩的直接控制 [6]。

实验中我们采用关节上的电位计测量关节角度、采用自行开发的推力传感器测量推力的大小、手工测量推力作用点的位置、采用机器人脚下的测力平台测量地面作用力、采用关节处应力传感器测量关节处的实际力矩。实验中通过高增益 PD 控制固定其他关节, 在两条腿之间增加协调控制使机器人近似于平面两连杆倒立摆模型:

$$\tau_{a,l} = \tau_a/2 + K_p^a(\hat{\theta}_a - \theta_{a,l}) \quad (3.30)$$

$$\tau_{h,l} = \tau_h/2 + K_p^h(\hat{\theta}_h - \theta_{h,l}) \quad (3.31)$$

$$\tau_{a,r} = \tau_a/2 + K_p^a(\hat{\theta}_a - \theta_{a,r}) \quad (3.32)$$

$$\tau_{h,r} = \tau_h/2 + K_p^h(\hat{\theta}_h - \theta_{h,r}) \quad (3.33)$$

其中 $\tau_{a,l}$ 、 $\tau_{h,l}$ 、 $\tau_{a,r}$ 和 $\tau_{h,r}$ 分别为左侧和右侧的踝关节驱动力矩和髋关节驱动力矩; $\theta_{a,l}$ 、 $\theta_{h,l}$ 、 $\theta_{a,r}$ 和 $\theta_{h,r}$ 分别为左侧和右侧踝关节角和髋关节角测量值。 τ_a 和 τ_h 是控制器期望输出力矩, $\hat{\theta}_a$ 和 $\hat{\theta}_h$ 分别为踝关节角和髋关节角估计值, K_p^h 和 K_p^a 为协调控制增益。因为关节上没有角速度传感器, 所以实验中通过扩展卡尔曼滤波器 (Extended Kalman Filter) 在线估计关节角、关节角速度、推力的大小和作用点位置。

实验中在机器人上身中部附近 (离地面约 1.2 米) 施加大小、方向、持续时间变化的水平推力。图 3.21 为其中一次实验的视频截图, 图 3.22 为测量得到的推力大小、机器人的关节角和关节力矩响应, 图 3.23 为推力大小和位置的测量

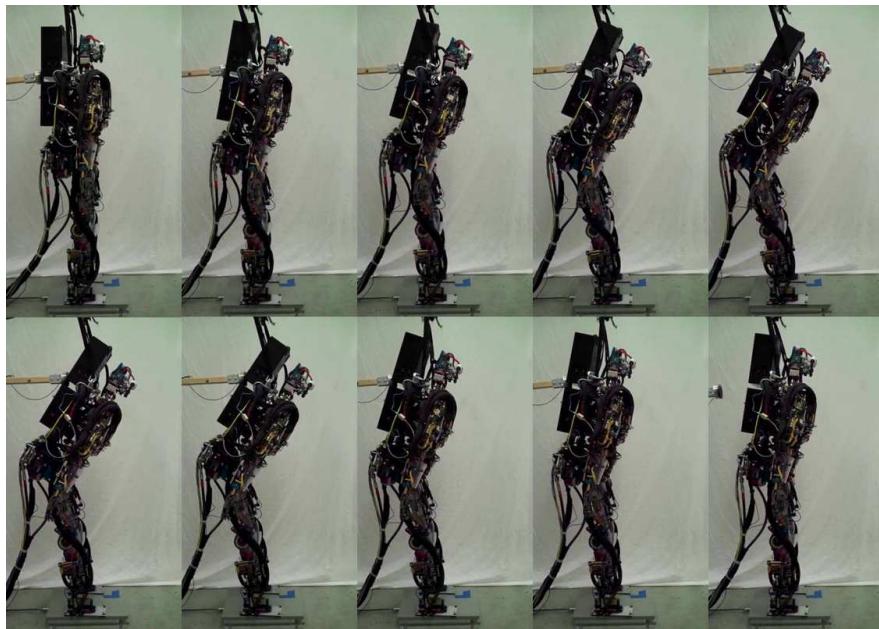


图 3.21 机器人站立时对外界推力的响应 (视频截图的时间间隔为 0.5 秒)

Fig. 3.21 Our robot responding to a push while standing (video frames are shown every half second)

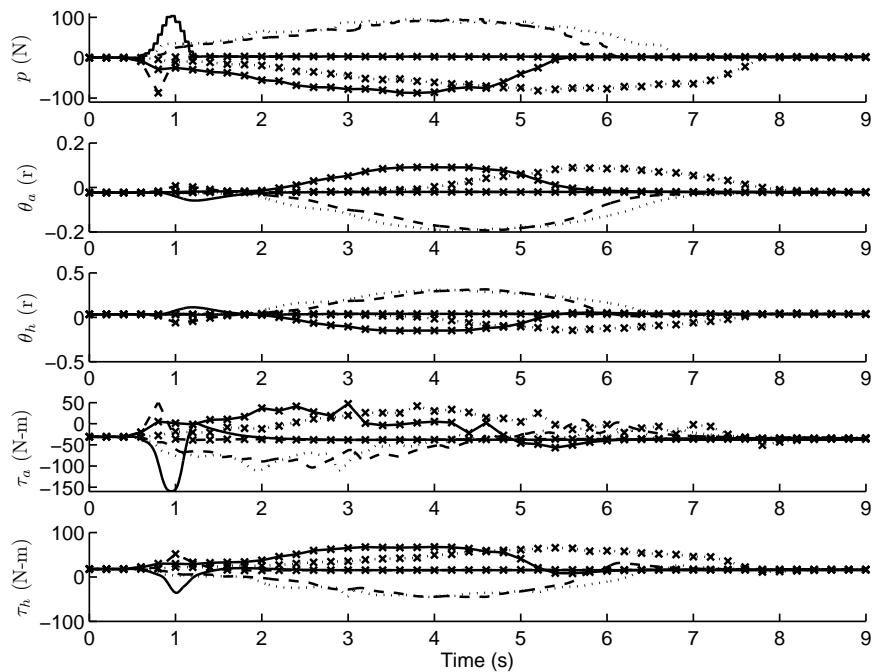


图 3.22 机器人站立时对不同推力的响应。三个前向推力 (正的) 和大小相近的后向推力。
Fig. 3.22 Our robot responding to pushes while standing. Three pushes are in the forward direction (positive forces) and a similar range of pushes are in the backward direction.

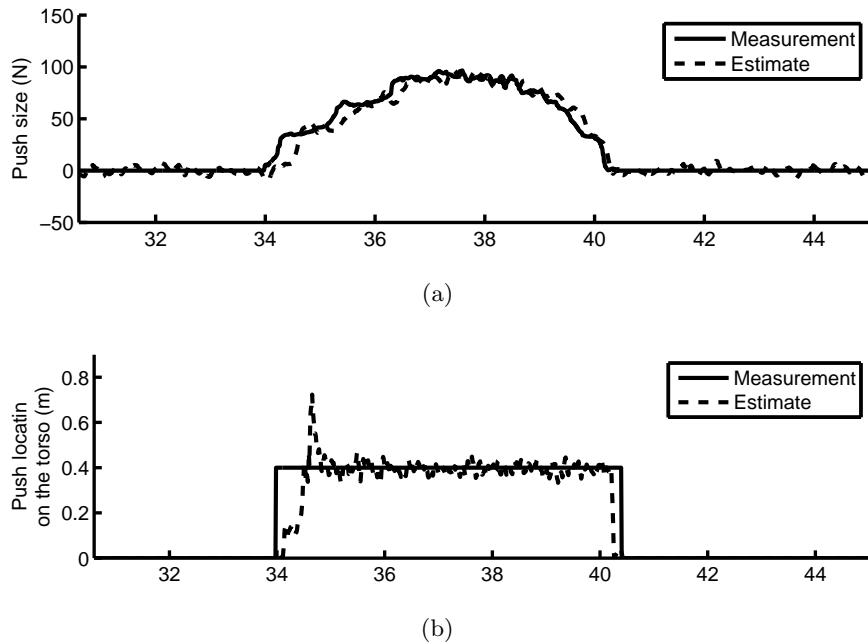


图 3.23 a) 推力大小估计 b) 推力位置估计

Fig. 3.23 a) Push size estimation b) Push location estimation

值和估计值。在不同大小的推力作用下，所提出的控制器均能使机器人保持站立平衡。

为了进一步验证控制器，将实验中测量到的推力用于动力学仿真，然后对实验结果和仿真结果进行比较。图 3.24 为关节角、关节角速度和关节力矩的比较结果，图 3.25 为标准化的地面对作用力对比结果。结果表明实验与仿真基本吻合。

3.5 本章小结

在实际的机器人控制中，物理参数的辨识是不可避免的，利用非线性最小二乘法和一定的实验配置能够辨识出系统的参数。另外，因为某些状态和扰动是不能直接测量的，仿真和实验中利用基于扩展卡尔曼滤波器的状态估计方法能够在线估计出系统的状态和扰动。针对目前参数化控制器存在性能局限、采用动态规划实现控制器存在计算量大且只实现了瞬间扰动下的站立平衡等问题，提出了基于轨迹库的站立平衡控制方法。该方法不仅能够比较快速地得到控制器，而且能够保证最终得到的控制器对特定优化指标的性能。在生成轨迹库的过程中，针对如何减少轨迹数目这一关键性问题，提出了初始条件自适应

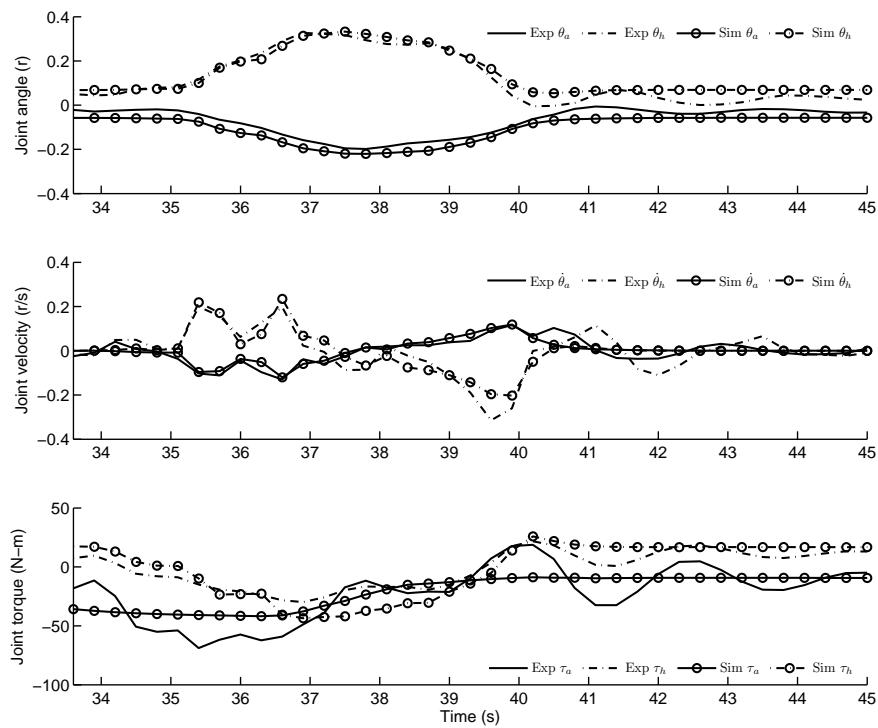


图 3.24 站立平衡控制仿真和实验结果对比

Fig. 3.24 The comparison between simulation and experiment results of standing balance control

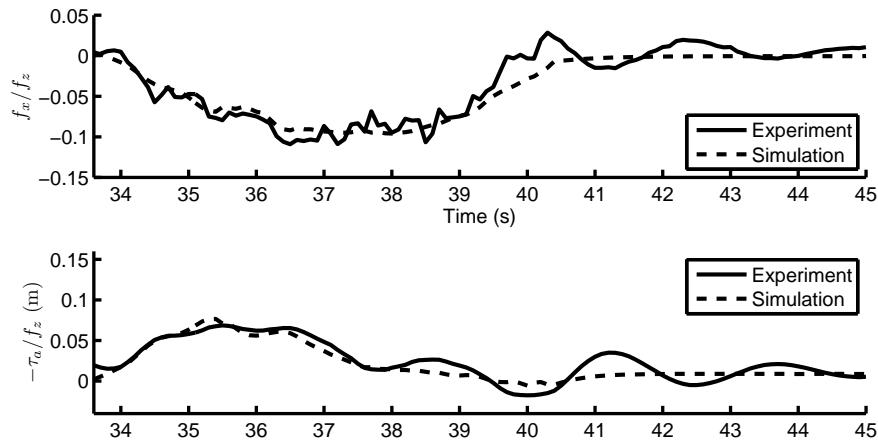
图 3.25 标准化的地面作用力仿真和实验结果对比，其中脚支撑区域为 $(-0.04, 0.16)$ 米。

Fig. 3.25 The comparison between simulation and experiment results of the normalized ground reaction forces, where the foot support region is $(-0.04, 0.16)$ meters.

网格上基于控制器预测性能的轨迹库生成方法。该方法不仅能够保证生成的轨迹之间的一致性，从而可以利用来自不同轨迹的控制律局部近似模型实现状态反馈控制律，而且能够在满足控制器性能要求的同时，减少轨迹库中轨迹数目。利用本文提出的方法，实现了瞬间和持续且时变的扰动下的站立平衡控制，仿真结果验证了方法的有效性。与基于线性二次调节器（LQR）的增益调度控制器和采用最优增益的全状态反馈控制的对比结果说明了该方法的性能优势。在 Sarcos 仿人机器人上的实验结果进一步验证了本文所提出的方法的有效性。

第四章 持续外力作用下站立平衡控制的期望状态研究

仿人机器人站立平衡控制需要选择某个静平衡状态作为期望状态，然后将机器人稳定到这个选定的期望状态。以往的研究将扰动假设为瞬间的冲击，直接选取静止站立为期望状态 [30, 31, 37, 40, 41, 108]；或者将扰动假设为恒定的外力，选取关节力矩最小的静平衡状态作为控制中的期望状态 [44, 103]。由于仿人机器人由多个关节和身体部分构成，相同外力作用下静平衡状态并不唯一。对于持续扰动下的站立平衡控制，期望状态的选取对在外力消失以后机器人恢复直立的难易程度有很大的影响，可以通过一个简单实例加以说明。机器人在地面上站立时，如果支撑脚没有相对地面的运动，可以将整个机器人建模为多连杆倒立摆。对于两连杆倒立摆模型（图 4.1 (a)），当某个大小恒定的外力作用在机器人的头部时，存在多种可能的静平衡状态。图 4.1 (b) 为一种静态平衡姿态，此时踝关节和髋关节的力矩均为零，机器人保持这种姿态的代价最小，但是此时质心在地面的投影在脚支撑区域以外。如果外力忽然消失，机器人摔倒的风险很大。图 4.1 (c) 为另一种静态平衡姿态，此时机器人踝关节和髋关节的力矩均不为零，但是此时质心在地面的投影在脚支撑区域以内，外力忽然消失后机器人更容易恢复直立平衡。

本章在第三章提出的基于轨迹库的站立平衡控制方法 [103] 的基础上，进一步研究了持续外力作用下期望状态的选取问题。从最优控制出发，同时考虑了扰动下保持某个期望状态的代价和扰动消失后从这个状态恢复直立的代价，提出了确定最佳期望状态的方法。采用最佳期望状态实现了更加鲁棒的站立平衡控制。

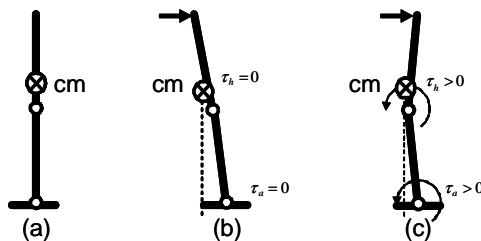


图 4.1 持续推力下的两连杆倒立摆模型可能的静平衡状态

Fig. 4.1 Possible static equilibrium state of the two-link model in the presence of a continuous push

4.1 期望状态优化方法

为了确定持续外力下最佳的期望状态，假设机器人已经稳定到某个静平衡状态 \mathbf{x}_d ，外力可能继续存在的时间为 t_p 。定义一个代价函数

$$J(\mathbf{x}_d, \mathbf{u}_d, t_p) \triangleq \int_0^{t_p} \mathbf{u}_d^\top \mathbf{R} \mathbf{u}_d dt + \int_{t_p}^{\infty} L(\mathbf{x}, \mathbf{u}, 0, r) dt, \quad (4.1)$$

其中等式右侧第一项和第二项分别描述了维持某个期望状态， \mathbf{x}_d ，的代价和外力消失以后从这个状态恢复直立的代价，其中 \mathbf{u}_d 表示 \mathbf{x}_d 对应的期望控制量， $\mathbf{x}(t_p) = \mathbf{x}_d$ 。站立平衡控制的单步代价函数 L 由公式 (3.20) 给出。在没有外力的情况下， $\mathbf{x}_d = \mathbf{0}$ 和 $\mathbf{u}_d = \mathbf{0}$ ，因此 $L(\mathbf{x}, \mathbf{u}, 0, r) = \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}$ 。

因为 \mathbf{u}_d 为机器人处于期望状态 \mathbf{x}_d 时对应的控制，所以在保持期望状态时一直不变。因此公式 (4.1) 右侧第一项可以进一步表示为

$$\int_0^{t_p} \mathbf{u}_d^\top \mathbf{R} \mathbf{u}_d dt = \mathbf{u}_d^\top \mathbf{R} \mathbf{u}_d t_p \quad (4.2)$$

如果扰动消失后采用最优控制，则公式 (4.1) 右侧第二项将成为 \mathbf{x}_d 的函数，即

$$V(\mathbf{x}_d, 0, r) = \min_{\mathbf{u}} \int_{t_p}^{\infty} L(\mathbf{x}, \mathbf{u}, 0, r) dt \quad (4.3)$$

$V(\mathbf{x}_d, 0, r)$ 表示没有外力时以 \mathbf{x}_d 为初始状态采用最优控制能取得的最小代价，可以通过动态规划 (Dynamic Programming) 的“值迭代”方法得到 [93]。这里将系统在原点附近线性化， $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ ，然后求解 Riccati 方程：

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P} = 0 \quad (4.4)$$

从而得到值函数在原点附近的二阶近似：

$$V(\mathbf{x}, 0, r) \approx \frac{1}{2} \mathbf{x}^\top \mathbf{P} \mathbf{x} \quad (4.5)$$

如果 t_p 是随机变量且满足 $\gamma(t_p)$ 概率分布函数，则代价函数的数学期望值为

$$\begin{aligned} E(J) &= \int_0^{\infty} \gamma(t_p) J(\mathbf{x}_d, \mathbf{u}_d, t_p) dt_p \\ &\approx E(t_p) \mathbf{u}_d^\top \mathbf{R} \mathbf{u}_d + \frac{1}{2} \mathbf{x}_d^\top \mathbf{P} \mathbf{x}_d \end{aligned} \quad (4.6)$$

最后，采用最小化代价函数的数学期望值得到最佳期望状态：

$$\min_{\mathbf{x}_d} E(J) \quad (4.7)$$

同时需要满足下面的约束：

$$f(\mathbf{x}_d, \mathbf{u}_d, p, r) = 0 \quad (4.8)$$

$$\mathbf{x}_L \leq \mathbf{x}_d \leq \mathbf{x}_U \quad (4.9)$$

$$\mathbf{u}_L \leq \mathbf{u}_d \leq \mathbf{u}_U, \quad (4.10)$$

我们采用能够基于 SQP 方法的 SNOPT 优化软件包对期望状态进行优化 [59]。

4.2 期望状态优化结果

4.2.1 两连杆倒立摆模型

对于两连杆倒立摆，腿为连杆 1，上身为连杆 2。连杆 1 和连杆 2 的质量均为 35 公斤，长度均为 1 米的均匀细杆。采用前后对称的脚，前后长度均为 0.1 米。脚踝关节和髋关节的转动范围分别为 $-0.8 \sim 0.4$ 和 $-0.1 \sim 1.5$ 弧度。为了保持脚相对地面静止，脚踝的力矩范围是 $-70 \sim 70$ N·m。髋关节的力矩范围均为 $-500 \sim 500$ N·m。在下面的结果中，将指向右侧的外力定义为正。

首先研究机器人的最佳期望状态随外力大小的变化规律。所有外力为作用在机器人头部（离地面 2 米）的水平外力，并选取 $\mathbf{Q} = \mathbf{I}_4$, $\mathbf{R} = \mathbf{I}_2$, $E(t_p) = 10$ 。图 4.2 为最佳期望状态对应的关节角、关节力矩和质心的地面投影位置随外力大小的变化，图 4.3 为对应的机器人姿态。当外力比较小时，随着外力的增大脚踝力矩接近于线性增加，而关节角基本保持为零，质心的地面投影基本保持在脚的中央。当脚踝力矩达到最大值后，最佳姿态开始向外力相反方向倾斜，质心的地面投影随之向外力相反方向移动，从而利用重力补偿外力。

下面研究机器人的最佳期望状态随外力预计持续时间的变化。所有外力均设定为 60N、作用在机器人头部（离地面 2 米）的水平外力。图 4.4 显示了最佳站立姿势对应的关节角、关节力矩和质心的地面投影位置随 $E(t_p)$ 的变化。图 4.5 为对应的机器人姿态。当外力预计持续时间比较短时，最佳期望状态采取的关节力矩均比较大，此时质心的地面投影比较靠近支撑区域的中心；当外力预计持续时间比较长时，最佳期望状态采取的关节力矩比较小，此时质心的地面投影开始远离支撑区域的中心。两个极端的例子是外力预计持续时间为零和正无穷，如图 4.5 (a) 和 4.5 (c) 所示。前一种情况对应假设外力随时可能消

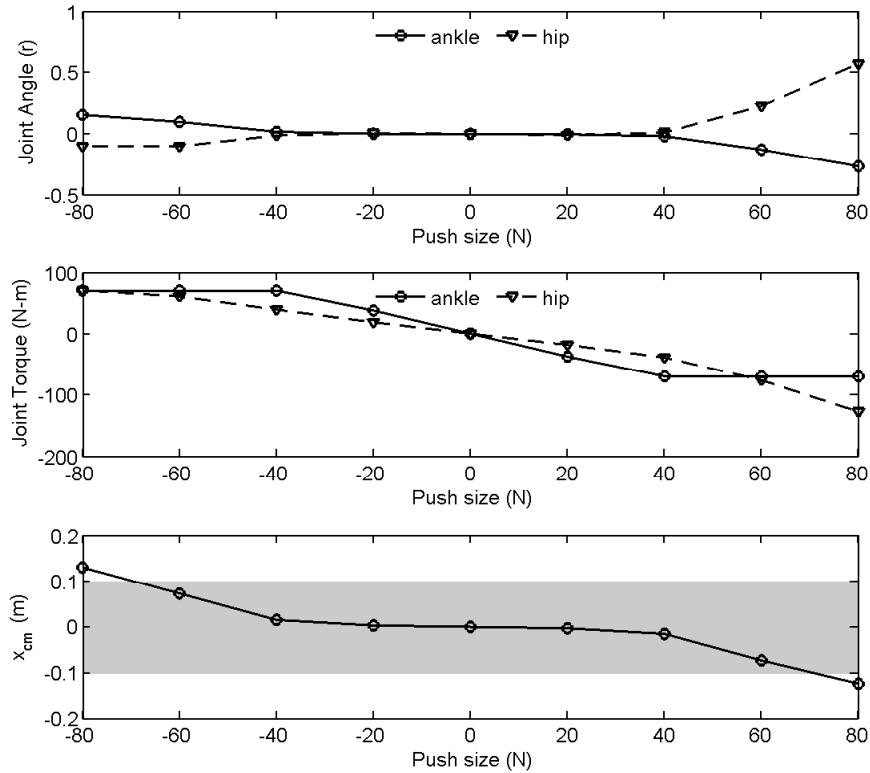


图 4.2 两连杆机器人期望状态随外力大小的变化 (灰色区域表示脚的支撑区域)

Fig. 4.2 The desired state of the two-link robot changes with the size of external force.
(the gray region denotes the foot-support region).

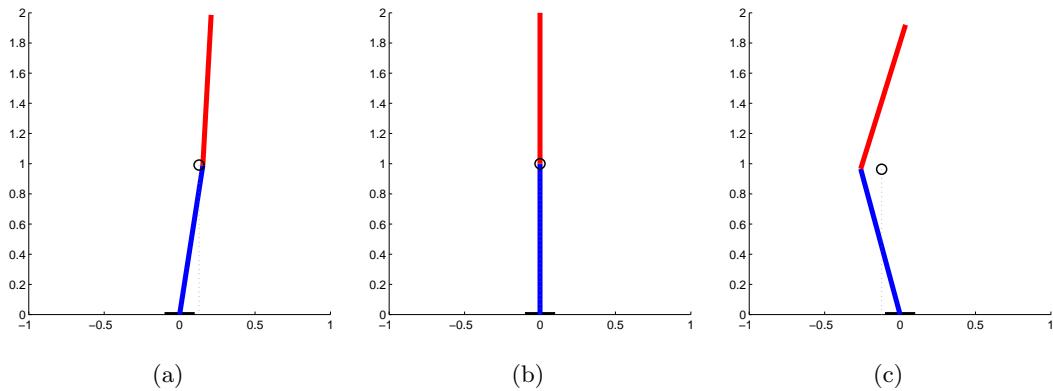


图 4.3 大小分别为 -80, 0 和 +80 牛顿的外力下的期望状态

Fig. 4.3 The desired states for external forces of -80, 0, and +80 Newtons.

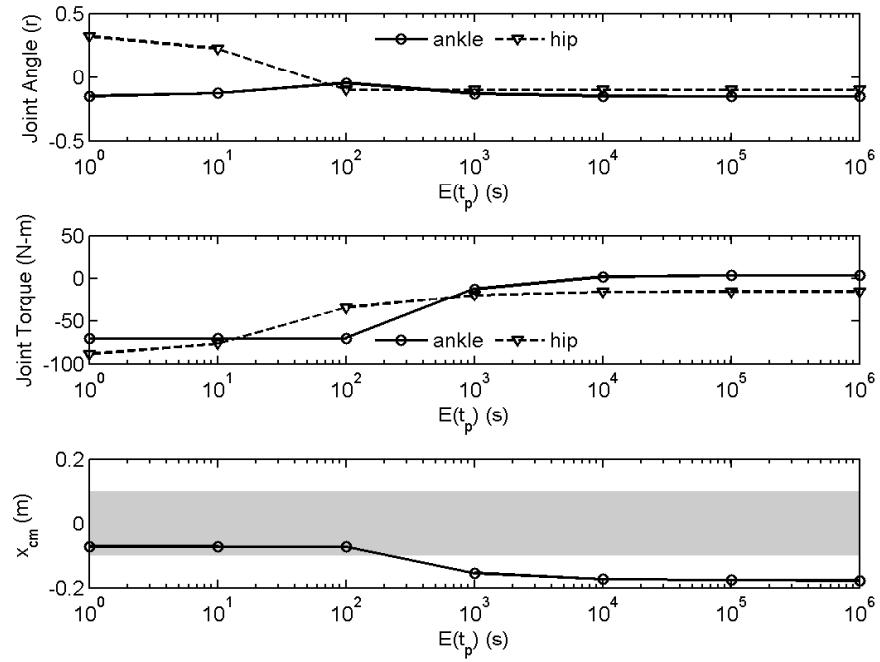
图 4.4 两连杆机器人期望状态随外力 $E(t_p)$ 的变化 (灰色区域表示脚的支撑区域)

Fig. 4.4 The desired state of the two-link robot changes with $E(t_p)$ of external force (the gray region denotes the foot-support region).

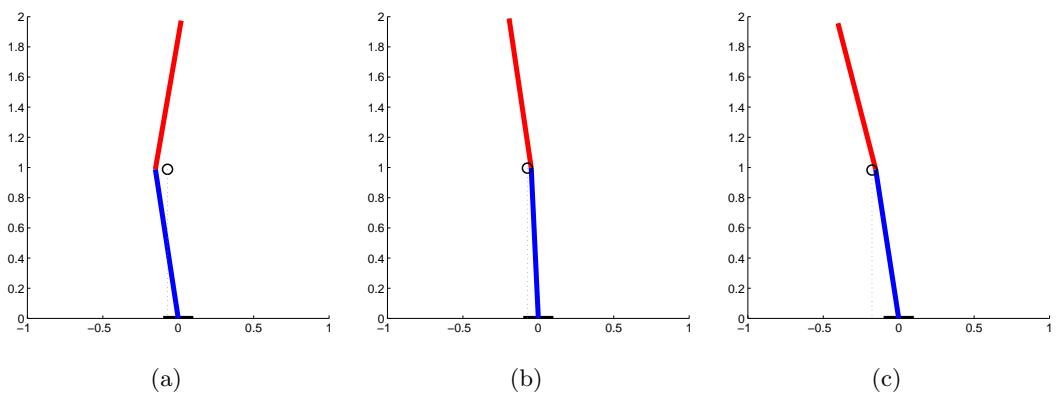
图 4.5 $E(t_p)$ 分别为 0, 10^2 和 $+\infty$ 时两连杆机器人的期望状态

Fig. 4.5 The desired states of the two-link robot for $E(t_p)$ of 0, 10^2 , and $+\infty$.

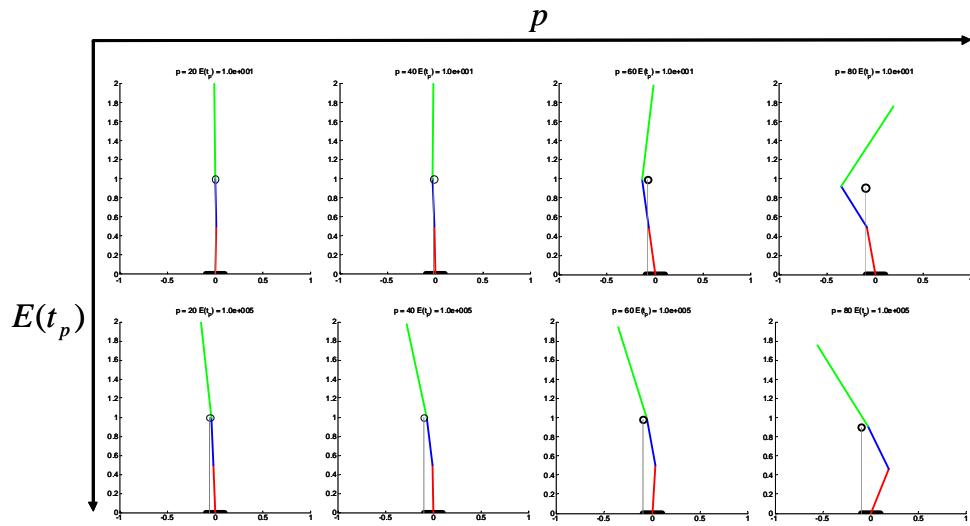


图 4.6 三连杆机器人期望状态优化结果随外力大小 p 和期望持续时间 $E(t_p)$ 的变化。外力指向右侧。

Fig. 4.6 The desired state of the three-link robot changes with size p and expected lasting time $E(t_p)$ of external forces pointing to the right.

失的情况，采用这个姿态对机器人最安全，因为扰动消失后恢复直立的代价最小；后一种情况对应假设扰动为恒定外力的情况，采用这个姿态对机器人最危险，因为质心的地面投影离开了脚支撑区域，从这个状态恢复直立比前一种情况更加困难。

4.2.2 三连杆倒立摆模型

本文同时研究了三连杆机器人，其中小腿为连杆1，大腿为连杆2，上身为连杆3。连杆1和2的质量均为17.5公斤，长度为0.5米，连杆3的质量为35公斤，长度为1米，所有连杆为均匀细杆。脚踝、膝盖和腰关节的转动范围分别是 $-0.8 \sim 0.4$, $-2.5 \sim 0.01$ 和 $-0.1 \sim 1.5$ 弧度。膝关节和腰关节的力矩范围均为 $-500 \sim 500$ N·m。为了防止脚相对地面翻转，脚踝的力矩限制在 $-70 \sim +70$ N·m之间。优化时选取 $\mathbf{Q} = \mathbf{I}_6$ 和 $\mathbf{R} = \mathbf{I}_3$ 。

不同大小、不同期望持续时间向右的外力的期望状态优化结果如图 4.6。不同大小、不同期望持续时间向左的外力的期望状态优化结果如图 4.7。以上结果表明：当外力的持续时间期望值比较小时，最佳姿势尽量保持质心在地面的投影靠近支撑区域的中心，此时机器人下蹲比较明显。当外力持续时间期望值比较大时，最佳姿势的质心在地面的投影向外力相反方向移动更加明显，此时机

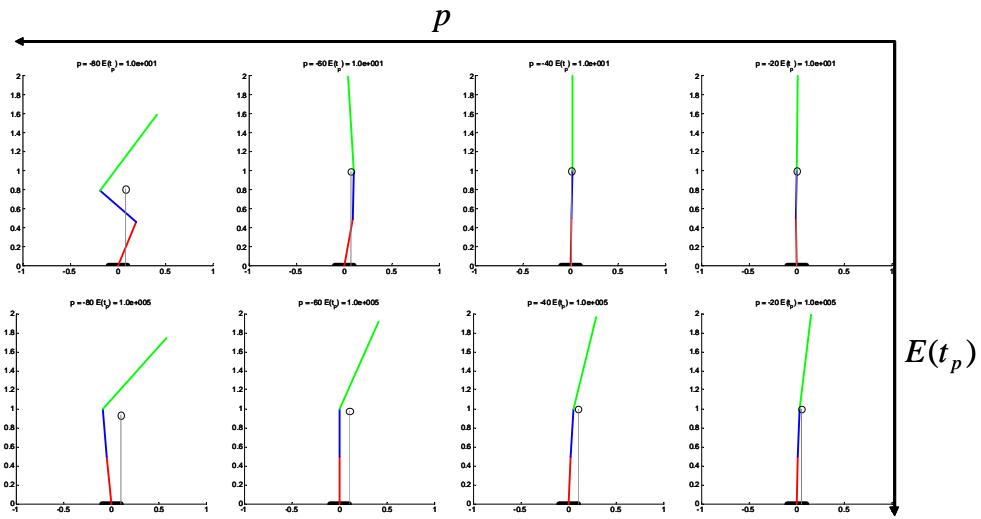


图 4.7 三连杆机器人期望状态优化结果随外力大小 p 和期望持续时间 $E(t_p)$ 的变化。外力指向左侧。

Fig. 4.7 The desired state of the three-link robot changes with size p and expected lasting time $E(t_p)$ of external forces pointing to the left.

器人整体倾斜更加明显。

4.2.3 四连杆倒立摆模型

我们同时研究了四连杆机器人，其中小腿为连杆 1，大腿为连杆 2，上身为连杆 3，手臂为连杆 4。连杆 1 和 2 的质量均为 17.5 千克，长度为 0.5 米，连杆 3 的质量为 26.25 千克，长度为 1 米，连杆 4 的质量为 8.75 千克，长度为 1 米。所有连杆为均匀细杆。脚踝、膝盖、髋关节和肩关节的转动范围分别是 $-0.8 \sim 0.4$, $-2.5 \sim 0.01$, $-0.1 \sim 1.5$ 和 $-2.5 \sim 0.5$ 弧度。为了保持脚相对地面静止，脚踝的力矩范围是 $-70 \sim 70$ N-m。膝关节和髋关节的力矩范围均为 $-500 \sim 500$ N-m，肩关节力矩范围为 $-250 \sim 250$ N-m。

选取 $\mathbf{Q} = \mathbf{I}_8$, $\mathbf{R} = \mathbf{I}_4$, $E(t_p) = 10$, 所有外力为作用在机器人头部（离地面 2 米）的水平外力。图 4.8 显示了最佳期望状态的关节角、关节力矩和质心的地面投影位置随外力大小的变化，图 4.9 为对应的机器人姿态。当外力比较小时，随着外力的增大脚踝力矩接近于线性增加，关节角基本保持为零，质心的地面投影基本保持在脚的中央。当脚踝力矩达到最大值后，最佳姿态开始向外力相反方向倾斜，质心的地面投影也随之向外力相反方向移动，从而利用重力补偿外力。

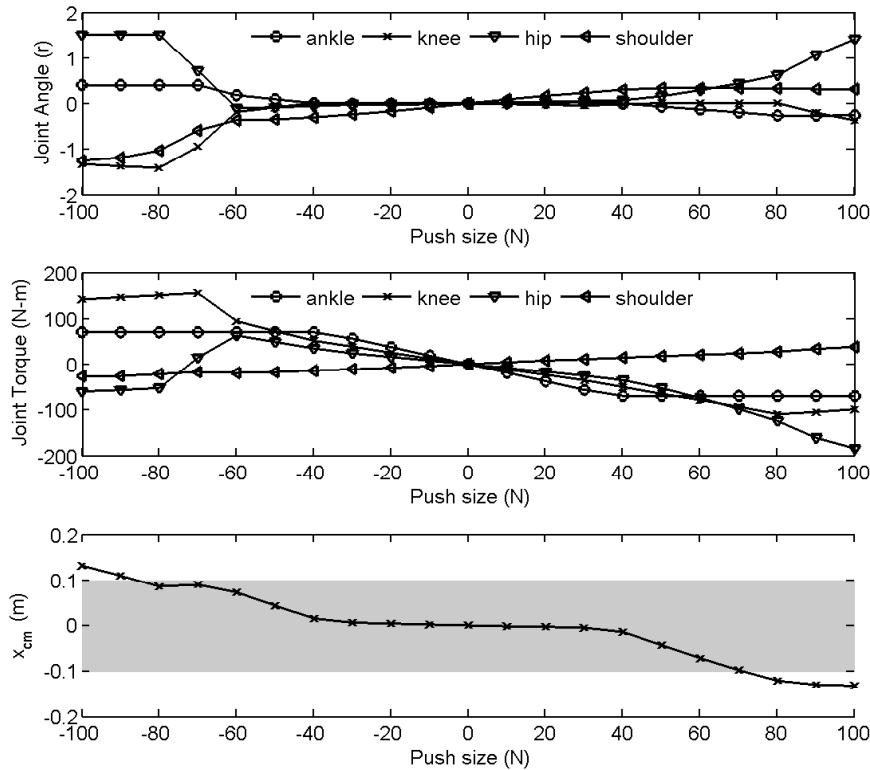


图 4.8 四连杆机器人期望状态随外力大小的变化 (灰色区域表示脚的支撑区域)
 Fig. 4.8 The desired state of the four-link robot changes with the size of external force
 (the gray region denotes the foot-support region).

我们进一步研究四连杆机器人最佳期望状态随外力预计持续时间的变化。设定外力为 60N、作用在机器人头部(离地面 2 米)的水平外力, 图 4.10 显示了四连杆仿真机器人最佳期望状态对应的关节角、关节力矩和质心的地面投影位置随外力持续时间数学期望值的变化, 图 4.11 为对应的机器人姿态。当外力预计持续时间比较短时, 最佳期望状态采取的关节力矩均比较大, 但此时质心的地面投影比较靠近支撑区域的中心, 因此扰动消失后更容易恢复直立; 当外力预计持续时间比较长时, 最佳期望状态采取的关节力矩均比较小, 但此时质心的地面投影开始远离支撑区域的中心, 因此在扰动消失后恢复直立变得更加困难。

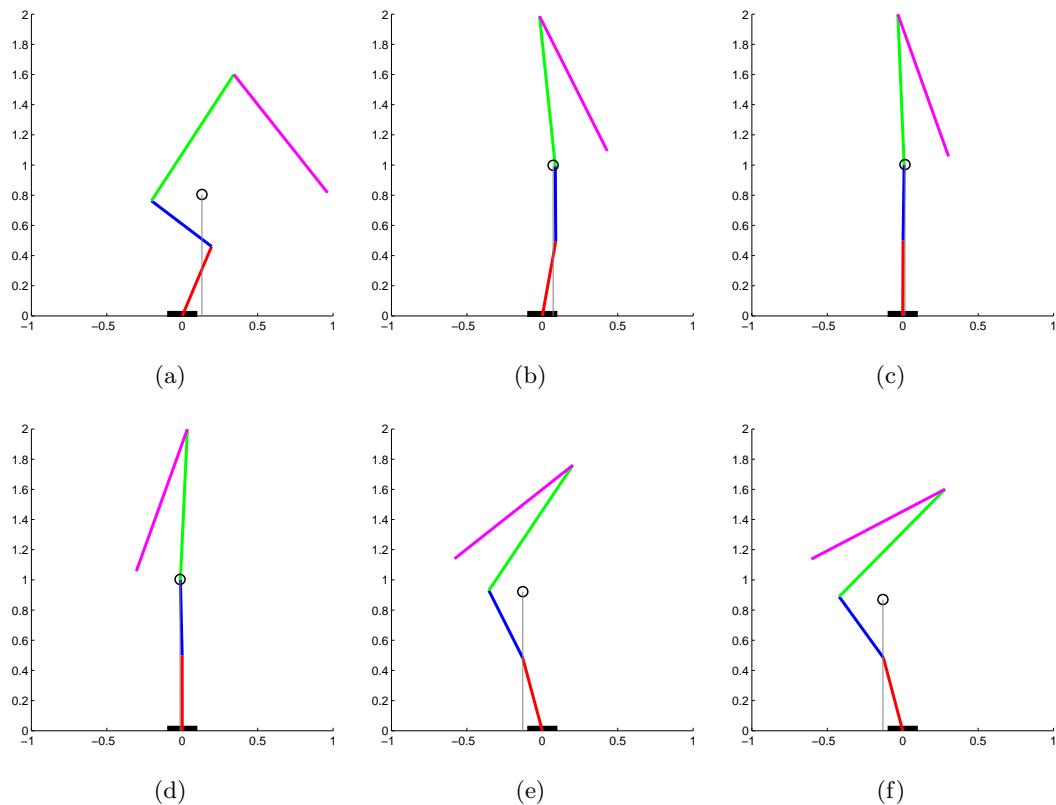
图 4.9 大小分别为 -100 、 -60 、 -40 、 40 、 90 和 100 牛顿的外力下机器人的期望状态

Fig. 4.9 The robot's desired state for external forces of -100 , -60 , -40 , 40 , 90 , and 100 Newtons.

4.3 仿真和实验结果

4.3.1 仿真

我们采用 $E(t_p) = +\infty$ 确定期望状态设计基于轨迹库的站立平衡控制。此时期望状态对应的关节力矩最小，因此在恒定外力下保持这个状态的代价最小。采用 40 牛顿、作用在机器人的头部（距地面约为 2 米）、持续时间为 5 秒的恒定外力作为扰动，图 4.12 为站立平衡控制的关节角、关节角速度、关节力矩和质心的地面投影的响应。在外力作用下（ $0 \sim 5$ 秒），机器人能够稳定到选定的期望状态，此时机器人质心在地面的投影在脚支撑区域（ $-0.1 \sim 0.1$ 米）以外，如图 4.12 (d) 所示。当外力忽然消失（5 秒时），机器人将会摔倒。这是因为在确定期望状态时，没有考虑外力消失后恢复直立的代价，所以机器人倾斜过于严重，一旦外力消失，机器人很难恢复直立。

接着我们采用 $E(t_p) = 0$ 确定期望状态设计基于轨迹库的站立平衡控制。

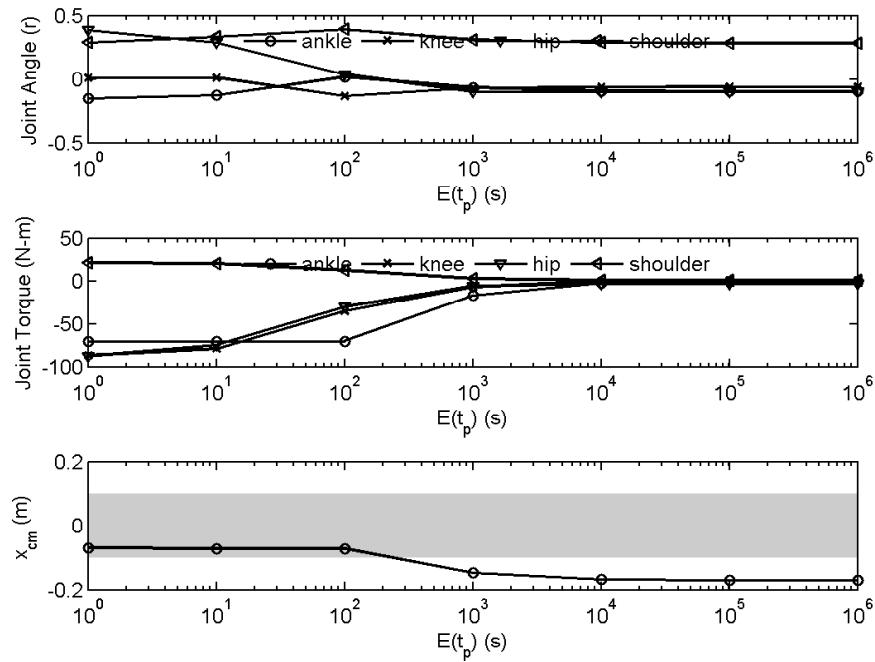
图 4.10 四连杆机器人期望状态随外力 $E(t_p)$ 变化的规律 (灰色区域表示脚的支撑区域)

Fig. 4.10 The desired state of a four-link robot changes with $E(t_p)$ of external force (the gray region denotes the foot-support region).

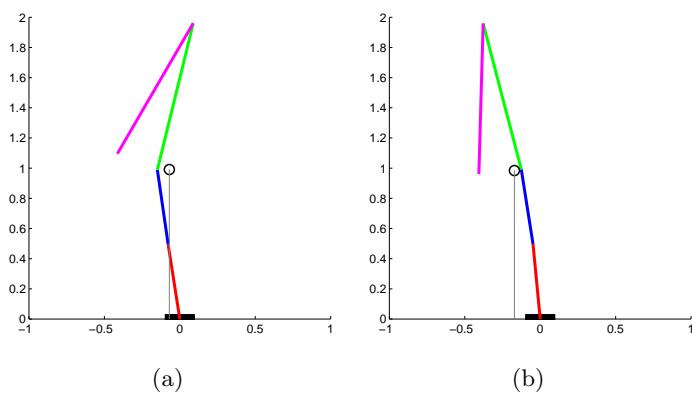
图 4.11 $E(t_p)$ 分别为 0 和 $+\infty$ 的机器人期望状态

Fig. 4.11 The robot's desired state for $E(t_p)$ of 0 and $+\infty$.

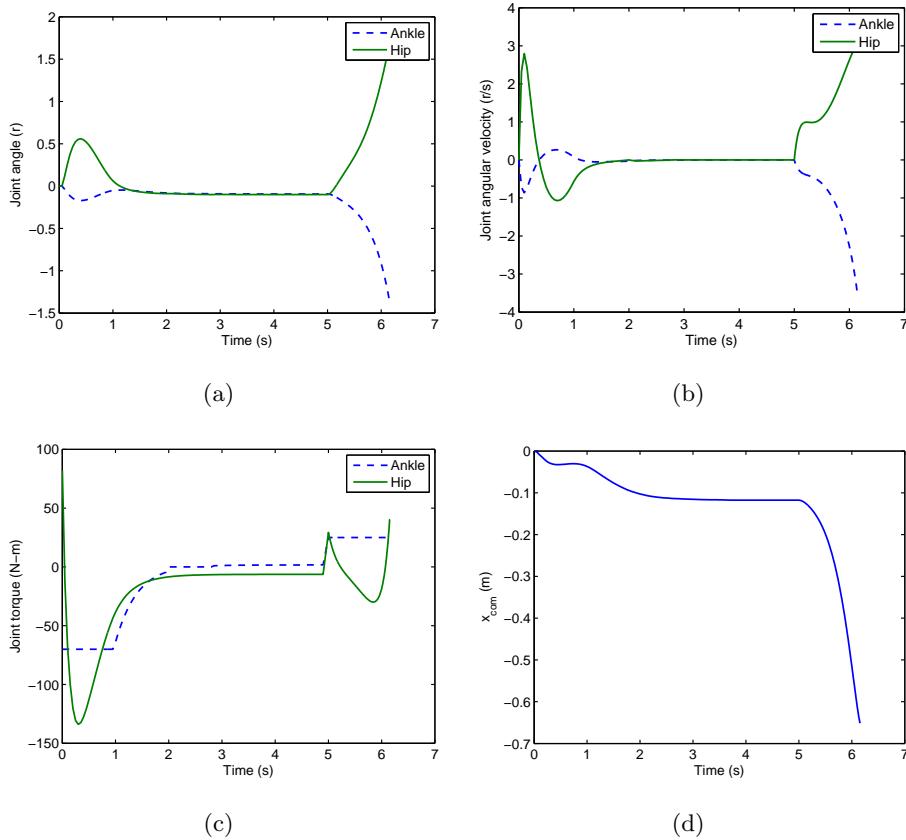
图 4.12 采用 $E(t_p) = +\infty$ 设计的站立平衡控制器在仿真中的响应

Fig. 4.12 The responses of the standing balance controller using $E(t_p) = +\infty$ in a simulation.

此时在扰动消失后机器人从这个期望状态恢复直立的代价最小。同样采用 40 牛顿、作用在机器人的头部（距地面约为 2 米）、持续时间为 5 秒的恒定外力下作为扰动，图 4.13 为站立平衡控制的关节角、关节角速度、关节力矩和质心的地面投影的响应。在外力作用下（0 ~ 5 秒），机器人能够稳定到选定的期望状态。与前面的仿真不同的是，此时机器人质心的地面投影始终处于脚支撑区域以内，如图 4.13 (d) 所示。当扰动忽然消失（5 秒时），机器人仍能够恢复直立。

4.3.2 实验

我们在卡耐基-梅隆大学的 Sarcos 仿人机器人上进行了实验。采用基于轨迹库的控制方法实现站立平衡控制，并利用本节提出的方法确定其中的期望状态。实验中通过扩展卡尔曼滤波器（Extended Kalman Filter）在线估计关节角、关节角速度、推力的大小和作用点位置，

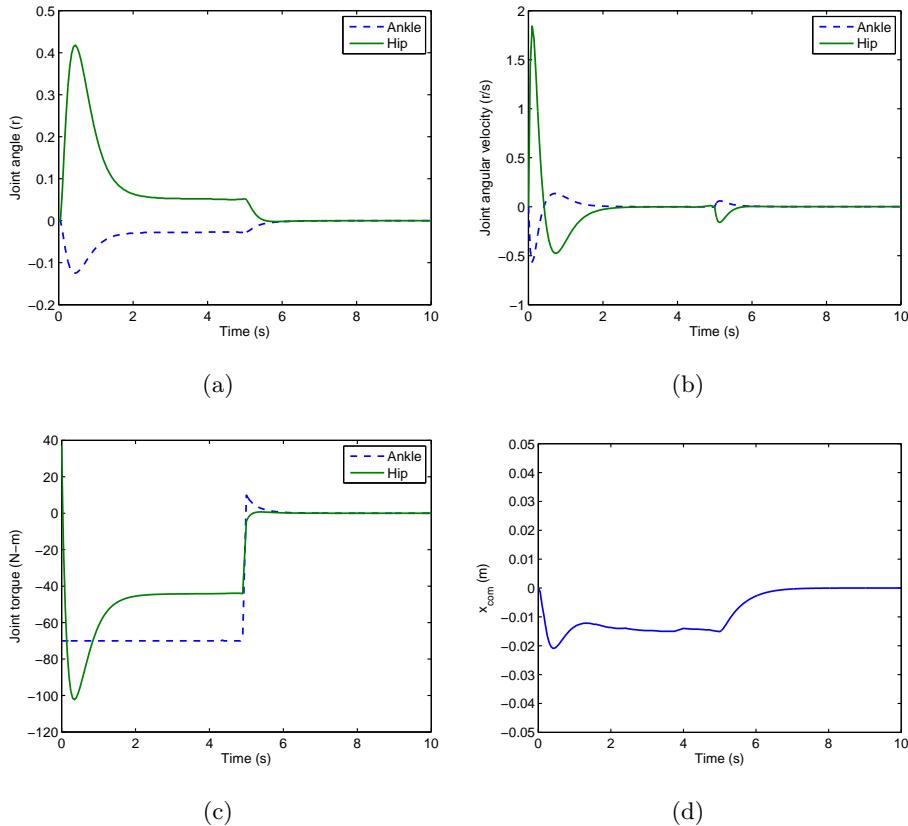
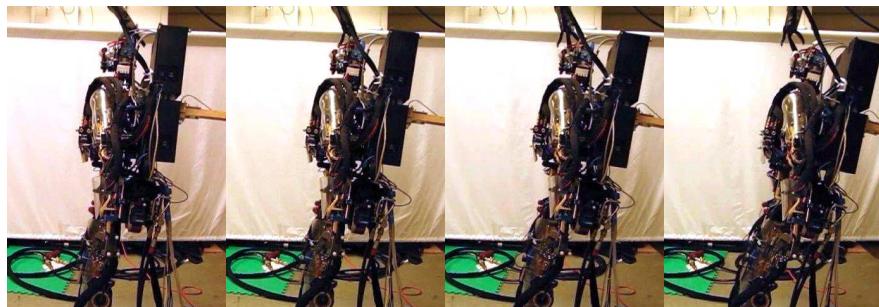
图 4.13 仿真中采用 $E(t_p) = 0$ 设计的站立平衡控制器响应Fig. 4.13 The responses of the standing balance controller using $E(t_p) = 0$ in a simulation.

图 4.14 采用最小化关节力矩期望状态的控制器对持续外力消失后的响应

Fig. 4.14 Our robot responding to the disappearance of a continuous push using the desired state which minimizes the joint torques.

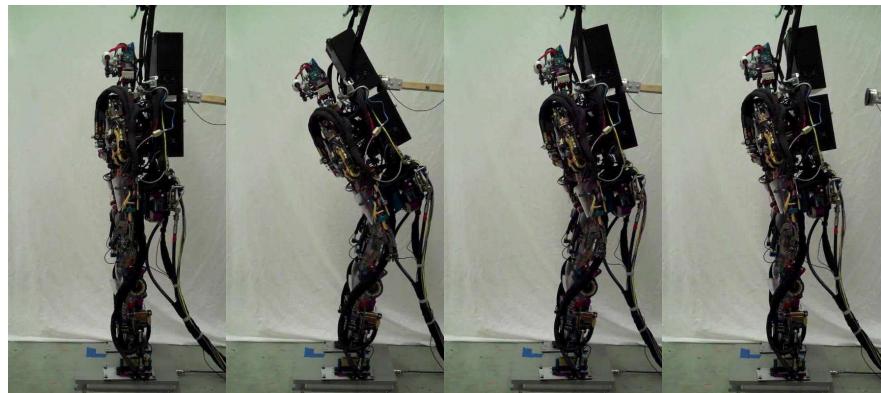


图 4.15 采用最佳期望状态的控制器对持续外力消失后的响应

Fig. 4.15 Our robot responding to the disappearance of a continuous push using the best desired state.

我们采用 $E(t_p) = +\infty$ 确定期望状态设计基于轨迹库的站立平衡控制。此时的期望状态采用的关节力矩最小。图 4.14 为机器人对缓慢增大然后忽然消失的外力的响应。随着外力的逐渐增大，机器人整体向外力相反方向倾斜，完全利用重力补偿外力。当外力忽然消失后，机器人因无法恢复直立而摔倒。

接着我们采用 $E(t_p) = 0$ 确定期望状态设计基于轨迹库的站立平衡控制。图 4.15 为机器人对缓慢增大然后忽然消失的外力的响应。随着外力的增大，机器人的双腿逐渐向外力相反方向倾斜，上身则向外力方向倾斜，当外力忽然消失，机器人仍然能够恢复平衡。

4.4 本章小结

针对持续扰动下的站立平衡控制问题，本章提出了一种确定期望状态的方法。本文的研究表明：如果强调站立平衡的鲁棒性，应该假设外力随时可能消失，即采用 $E(t_p) = 0$ 确定站立平衡的期望状态；如果外界扰动为恒定外力，应采用 $E(t_p) = +\infty$ 确定期望状态；持续外力作用下的最佳期望状态不仅决定于外力的大小，还决定于外力预计持续的时间。这一研究结果对于仿人机器人站立平衡的研究将会提出新的参考。该方法与基于轨迹库的站立平衡控制方法相结合实现了更加鲁棒的控制，仿真和实验说明方法的有效性。

第五章 基于轨迹库的仿人机器人行走控制

双足行走使仿人机器人具备了其他类型机器人不可比拟的运动能力，是仿人机器人区别于其他类型机器人的一个最典型特征。行走控制是仿人机器人运动控制研究的重要内容，对行走控制的研究不仅有利于充分发挥仿人机器人的运动能力，还有助于设计更加高效的假肢 [9]。行走控制的难点在于：1) 仿人机器人由很多关节和身体部分构成，行走过程中身体各部分的运动范围比较大，使系统动力学维数高、非线性明显；2) 在行走过程中支撑脚与地面之间的作用力必须满足一定的物理约束 [23, 24]；3) 行走时脚与地面之间的间歇性碰撞，使系统的动力学不连续；4) 在行走的过程中摆动脚的高度、机器人关节角、关节角速度以及关节驱动力矩的限制，进一步增加了行走控制的难度；5) 设计对于特定优化指标最优的行走控制更是一种挑战。

虽然在行走控制方面取得了巨大的成功，但是目前的行走控制方法在速度、效率和鲁棒性方面与人类相比仍有很大差距。针对目前仿人机器人行走控制方法存在的不足，本章将研究基于优化的行走控制方法。首先将行走控制问题归结为动力学不连续系统周期性任务的最优控制问题，提出一种领域最优控制方法。在此基础上，研究忽略双腿支撑的七连杆平面行走控制问题，提出基于轨迹库的控制方法。最后，研究更加具有一般性的行走控制（考虑双腿支撑的三维七连杆行走控制）。

5.1 动力学不连续系统周期性任务的最优控制

5.1.1 问题描述

一类动力学不连续系统的周期性运动，其动力学可以通过一组常微分方程

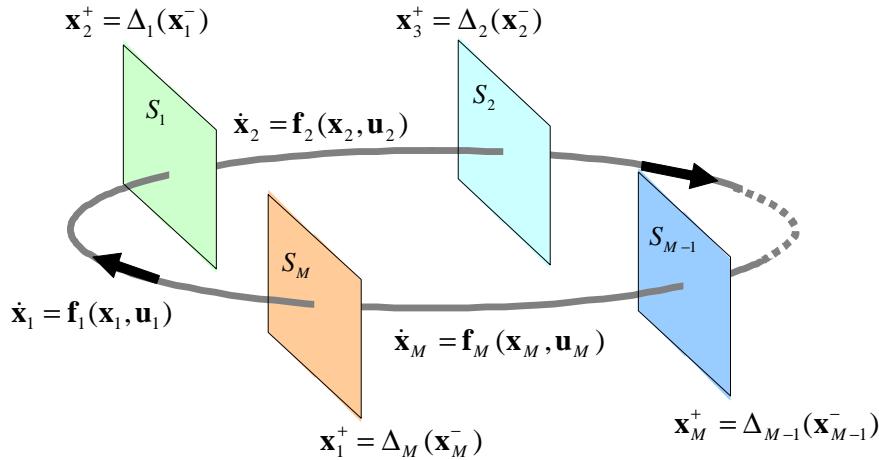


图 5.1 状态空间中动力学不连续系统的周期性运动

Fig. 5.1 The periodic motion of a system with dynamic discontinuities in state space.

和离散映射构成的混合方程描述:

$$\begin{cases} \dot{x}_1 = f_1(x_1, u_1) \\ x_2^+ = \Delta_1(x_1^-) & \text{if } x_1^- \in \mathcal{S}_1 \\ \dot{x}_2 = f_2(x_2, u_2) \\ x_3^+ = \Delta_2(x_2^-) & \text{if } x_2^- \in \mathcal{S}_2 \\ \dots \\ \dot{x}_M = f_M(x_M, u_M) \\ x_1^+ = \Delta_M(x_{M-1}^-) & \text{if } x_{M-1}^- \in \mathcal{S}_M \end{cases} \quad (5.1)$$

其中 M 一个周期内通过动力学不连续的次数, x_i , u_i 和 f_i 分别为通过第 i 次动力学不连续之前的状态变量、控制变量和状态方程, x_i^- 为即将通过第 i 次动力学不连续前的状态, x_{i+1}^+ 为刚刚通过第 i 次动力学不连续后的状态, $\Delta_i(\cdot)$ 为第 i 次动力学不连续处的状态映射方程, \mathcal{S}_i 为第 i 次动力学不连续发生时的状态集合。动力学系统按照常微分方程 f_i 演化, 直至系统的状态进入 \mathcal{S}_i 集合。动力学不连续处的状态方程 Δ_i 将动力学不连续发生前的状态映射到新的动力学不连续发生后的状态, 动力学系统从这个新的状态开始按新的状态方程 f_{i+1} 演化直至状态落入下一动力学不连续 \mathcal{S}_{i+1} 集合, 如图 5.1 所示。

对于周期性任务无限时域最优控制问题, 代价函数需要包含折扣因

予, $e^{-\beta t}$, 以保证值函数有限:

$$J \triangleq \int_0^\infty e^{-\beta t} c(\mathbf{x}(t), \mathbf{u}(t)) dt, \quad (5.2)$$

其中 $\beta > 0$ 为折扣率, $c(\cdot, \cdot)$ 为代价率函数。

5.1.2 稳态轨迹参数化优化方法

对于周期性任务, 无限时域最优控制中的稳态轨迹为周期性轨迹。对于周期为 T 的周期轨迹, 无限时域上的代价函数为

$$\begin{aligned} J &= \int_0^\infty e^{-\beta t} c(\mathbf{x}(t), \mathbf{u}(t)) dt \\ &= (1 + e^{-\beta T} + e^{-\beta 2T} + \dots) \int_0^T e^{-\beta t} c(\mathbf{x}(t), \mathbf{u}(t)) dt, \end{aligned} \quad (5.3)$$

其中幂级数收敛

$$1 + e^{-\beta T} + e^{-\beta 2T} + \dots \rightarrow \frac{1}{1 - e^{-\beta T}} \quad (5.4)$$

因此

$$J = \frac{1}{1 - e^{-\beta T}} \int_0^T e^{-\beta t} c(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (5.5)$$

若一个周期内动力学不连续的次数为 M , 并采用 $0 = t_0 < t_1 \dots < t_{M-1} < t_M = T$ 表示通过动力学不连续处的时间, 则周期轨迹的代价为

$$\begin{aligned} J &= \frac{1}{1 - e^{-\beta T}} \int_0^T e^{-\beta t} c(\mathbf{x}(t), \mathbf{u}(t)) dt \\ &= \frac{1}{1 - e^{-\beta T}} \left[\int_0^{t_1} e^{-\beta t} c_1(\mathbf{x}_1(t), \mathbf{u}_1(t)) dt + e^{-\beta t_1} \psi_1(\mathbf{x}_1(t_1)) \right. \\ &\quad + \int_{t_1}^{t_2} e^{-\beta t} c_2(\mathbf{x}_2(t), \mathbf{u}_2(t)) dt + e^{-\beta t_2} \psi_2(\mathbf{x}_2(t_2)) + \dots \\ &\quad \left. + \int_{t_{M-1}}^T e^{-\beta t} c_M(\mathbf{x}_M(t), \mathbf{u}_M(t)) dt + e^{-\beta T} \psi_M(\mathbf{x}_M(T)) \right], \end{aligned} \quad (5.6)$$

其中 c_i 为 $[t_{i-1}, t_i]$ 区间的代价率函数, ψ_i 表示通过第 i 次动力学不连续处额外的惩罚函数。

为采用参数化轨迹优化方法计算无限时域最优控制中的稳态轨迹, 将每个时间段 $[t_{i-1}, t_i]$ 均匀的分成 N_i 等分, 然后采用 Runge-Kutta 方将每个时间段上

的状态方程 $\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i)$ 离散化为 $\mathbf{x}_i(k+1) = F_i(\mathbf{x}_i(k), \mathbf{u}_i(k))$ [88]。离散化以后代价函数成为

$$J = \frac{1}{1 - e^{-\beta T}} \sum_{i=1}^M \left(e^{-\beta t_i} \psi_i(\mathbf{x}_i(N_i)) + \sum_{k=0}^{N_i-1} e^{-\beta(t_{i-1} + \Delta T_i k)} c_i(\mathbf{x}_i(k), \mathbf{u}_i(k)) \Delta T_i \right), \quad (5.7)$$

其中 $\Delta T_i = (t_i - t_{i-1})/N_i$, $\mathbf{x}_i(k)$ 和 $\mathbf{u}_i(k)$ 为状态变量和控制变量在 $t_{i-1} + \Delta T_i k$ 时刻的值。将 t_1, \dots, t_M 和每个时间点上的状态变量和控制变量作为优化变量,

$$\begin{aligned} \mathbf{y} \triangleq & \{t_1, \dots, t_M, \mathbf{x}_1(1), \dots, \mathbf{x}_1(N_1), \dots, \mathbf{x}_M(1), \dots, \mathbf{x}_M(N_M) \\ & \mathbf{u}_1(0), \dots, \mathbf{u}_1(N_1 - 1), \dots, \mathbf{u}_M(0), \dots, \mathbf{u}_M(N_M - 1)\} \end{aligned} \quad (5.8)$$

将每个时间点上的离散状态方程和动力学不连续处的动力学方程作为约束

$$\begin{aligned} 0 \leq \mathbf{x}_i(k+1) - F_i(\mathbf{x}_i(k), \mathbf{u}_i(k)) \leq 0 \quad i = 1, 2, \dots, M; \\ k = 0, 1, \dots, N_i - 1 \end{aligned} \quad (5.9)$$

$$0 \leq \mathbf{x}_{i+1}(0) - \Delta_i(\mathbf{x}_i(N_i)) \leq 0 \quad i = 1, 2, \dots, M - 1 \quad (5.10)$$

$$0 \leq \mathbf{x}_1(0) - \Delta_M(\mathbf{x}_M(N_M)) \leq 0 \quad (5.11)$$

同时考虑状态变量和控制变量可能的路径约束

$$b_L \leq b(\mathbf{x}_i(k), \mathbf{u}_i(k)) \leq b_U \quad i = 1, 2, \dots, M; k = 0, 1, \dots, N_i - 1 \quad (5.12)$$

和

$$\mathbf{x}_i(N_i) \in \mathcal{S}_i \quad i = 1, 2, \dots, M \quad (5.13)$$

这样将求解稳态轨迹问题转化为一个一般的非线性优化问题, 从而可以通过 SNOPT 来求解 [59]。

5.1.3 值函数和控制律局部近似模型

首先推导动力学不连续处值函数的传递关系, 根据值函数的传递关系:

$$V_i(\mathbf{x}_i(N_i)) = \begin{cases} \psi_i(\mathbf{x}_i(N_i)) + V_{i+1}(\Delta_i(\mathbf{x}_i(N_i))) & i = 1, 2, \dots, M - 1 \\ \psi_i(\mathbf{x}_i(N_i)) + V_1(\Delta_i(\mathbf{x}_i(N_i))) & i = M \end{cases} \quad (5.14)$$

如果动力学不连续处的状态函数 Δ_i 光滑可微, 那么

$$\frac{\partial V_i}{\partial \mathbf{x}_i} = \begin{cases} \frac{\partial \psi_i}{\partial \mathbf{x}_i} + \frac{\partial V_{i+1}}{\partial \mathbf{x}_{i+1}} \frac{\partial \Delta_i}{\partial \mathbf{x}_i} & i = 1, 2, \dots, M-1 \\ \frac{\partial \psi_i}{\partial \mathbf{x}_i} + \frac{\partial V_1}{\partial \mathbf{x}_1} \frac{\partial \Delta_i}{\partial \mathbf{x}_i} & i = M \end{cases} \quad (5.15)$$

$$\frac{\partial^2 V_i}{\partial \mathbf{x}_i^2} = \begin{cases} \frac{\partial^2 \psi_i}{\partial \mathbf{x}_i^2} + \left(\frac{\partial \Delta_i}{\partial \mathbf{x}_i} \right)^T \frac{\partial^2 V_{i+1}}{\partial \mathbf{x}_{i+1}^2} \frac{\partial \Delta_i}{\partial \mathbf{x}_i} + \frac{\partial V_{i+1}}{\partial \mathbf{x}_{i+1}} \frac{\partial^2 \Delta_i}{\partial \mathbf{x}_i^2} & i = 1, 2, \dots, M-1 \\ \frac{\partial^2 \psi_i}{\partial \mathbf{x}_i^2} + \left(\frac{\partial \Delta_i}{\partial \mathbf{x}_i} \right)^T \frac{\partial^2 V_1}{\partial \mathbf{x}_1^2} \frac{\partial \Delta_i}{\partial \mathbf{x}_i} + \frac{\partial V_1}{\partial \mathbf{x}_1} \frac{\partial^2 \Delta_i}{\partial \mathbf{x}_i^2} & i = M \end{cases} \quad (5.16)$$

在得到了动力学不连续处值函数的递推关系以后, 便可以采用微分动态规划方法对得到的稳态轨迹进一步优化, 在轨迹优化的同时得到值函数和控制律的局部近似模型。通过这些局部近似模型和最近邻控制便可以构造稳态轨迹邻域内的状态反馈控制。

5.1.4 最近邻控制

微分动态规划在优化轨迹的同时得到最优控制律的局部近似模型, $\mathbf{u} = \mathbf{u}^p - \mathbf{K}^p(\mathbf{x} - \mathbf{x}^p)$, 其中 \mathbf{x} 为当前状态, \mathbf{x}^p , \mathbf{u}^p 和 \mathbf{K}^p 为轨迹上的采样状态及其对应的最优控制和反馈增益矩阵。对于无穷时域的最优控制问题, 这些局部近似模型为时不变的局部近似模型。

为了采用这些时不变的局部近似模型实现状态反馈控制, 最简单的方法是采用与当前状态最接近的局部近似模型预测当前状态对应的最优控制。为此, 首先定义与状态有关的欧式距离:

$$\Delta \mathbf{x}^T \mathbf{D} \Delta \mathbf{x}, \quad (5.17)$$

其中 \mathbf{D} 为距离加权矩阵, $\Delta \mathbf{x} \triangleq \mathbf{x} - \mathbf{x}^p$ 表示当前状态 \mathbf{x} 与最优轨迹采样状态 \mathbf{x}^p 之间的误差。因此, 当前状态对应的最优控制可利用最接近的局部近似模型近似为

$$\mathbf{u} = \mathbf{u}(\mathbf{x}) = \bar{\mathbf{u}} - \mathbf{K}(\mathbf{x} - \bar{\mathbf{x}}), \quad (5.18)$$

其中 $\bar{\mathbf{x}}$ 为距离当前最近的采样状态, $\bar{\mathbf{u}}$ 和 \mathbf{K} 为对应最优控制和增益矩阵。

5.2 平面七连杆行走控制

5.2.1 动力学不连续系统模型

我们首先研究了平面七连杆双足行走控制。平面七连杆双足行走机器人(图 5.2)由上身、两个带膝关节的腿和双脚构成。忽略双脚的质量, 身体其

表 5.1 双足机器人仿真物理参数

Table 5.1 Physical parameters of the simulated biped robot

	小腿	大腿	上身
质量 (kg)	6.90	5.68	50.00
角惯量 ($\text{kg}\cdot\text{m}^2$)	0.15	0.10	1.50
长度 (m)	0.38	0.39	0.80
质心位置 l_{cm} (m)	0.24	0.19	0.29

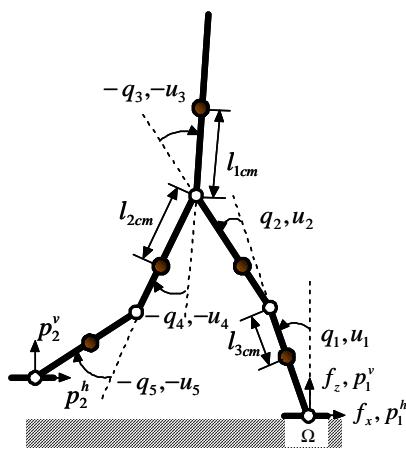


图 5.2 平面七连杆双足机器人简化模型

Fig. 5.2 Simplified structure of a planar 7-link biped robot

他各部分均为具有一定质量的刚体，通过转动关节相互连接。仿真中使用表 5.1 列出的 Sarcos 机器人运动学和动力学参数。Sarcos 机器人通过每个关节的液压驱动元件和应力传感器实现基于力矩的控制 [6]。首先研究不考虑双腿支撑的行走控制，一个行走周期由单腿支撑和摆动脚和地面的瞬间碰撞构成。单腿支撑可以通过一个常微分方程描述，而摆动脚和地面的瞬间碰撞可以通过碰撞模型描述。

单腿支撑模型

单腿支撑阶段，如果始终保持双脚与地面平行且支撑脚相对地面静止，那么整个系统具有 5 个自由度。不管哪条腿作为支撑腿都可以采用相同的单腿支撑模型，只是在碰撞发生以后需要交换坐标。动力学方程可以通过拉格朗日方

法得到，其结果是标准的二阶系统

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q})\mathbf{u}, \quad (5.19)$$

其中 $\mathbf{q} = (\theta_1, \theta_2, \dots, \theta_5)^\top$, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{5 \times 5}$ 为惯性矩阵, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^5$ 为向心力、科氏力和重力向量, $\mathbf{B}(\mathbf{q})$ 决定关节力矩 $\mathbf{u} \triangleq (u_1, \dots, u_5)^\top$ 对模型的作用。这个二阶系统可以表达成状态方程的形式

$$\dot{\mathbf{x}} = \frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{q})(-\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u}) \end{bmatrix} = f(\mathbf{x}, \mathbf{u}), \quad 0 < t < T \quad (5.20)$$

其中 $\mathbf{x} \triangleq (\mathbf{q}^\top, \dot{\mathbf{q}}^\top)^\top$, T 是单腿支撑阶段的时间长度。我们直接利用商业软件 SDFAST 得到状态方程。

碰撞模型

摆动脚和地面之间的碰撞建模为刚体之间的碰撞。刚体碰撞的假设有：

1. 关节为理想的转动关节，即只有完全弹性形变 [104].
2. 相对于冲击力的大小，向心力、科氏力、重力和关节力矩可以忽略。
3. 碰撞发生的瞬间，因碰撞引起的冲击力在瞬间改变关节角速度而不改变关节角。
4. 摆动脚接触地面以后没有反弹和相对地面的滑动。

碰撞模型产生了平滑的映射 (附录 A):

$$\mathbf{x}^+ = \Delta(\mathbf{x}^-), \quad \mathbf{x}^- \in \mathcal{S} \quad (5.21)$$

其中 \mathbf{x}^- 为碰撞即将发生前的状态值, \mathbf{x}^+ 为碰撞刚刚发生后的状态值, 对于水平地面 $\mathcal{S} = \{\mathbf{x} | p_2^v(\mathbf{x}) = 0, \dot{p}_2^v(\mathbf{x}) < 0\}$ (p_2^v 的定义见图 5.2)。函数 Δ 描述了因碰撞而引起的关节角速度的瞬间变化和因为支撑脚变化引起的坐标变化。

动力学不连续系统模型

双足机器人动力学可以描述为动力学不连续的非线性系统。机器人按照常微分方程 (5.20) 演化直至发生碰撞。碰撞引起关节角速度瞬间变化, 支撑脚变化引起坐标瞬间变化, 由此产生一个新的起始状态, 如图 5.3 所示。不考虑双腿支撑时, 一个行走周期由一个单腿支撑阶段和摆动脚和地面的碰撞构成, 因此一个周期内通过动力学不连续的次数 $M = 1$ 。

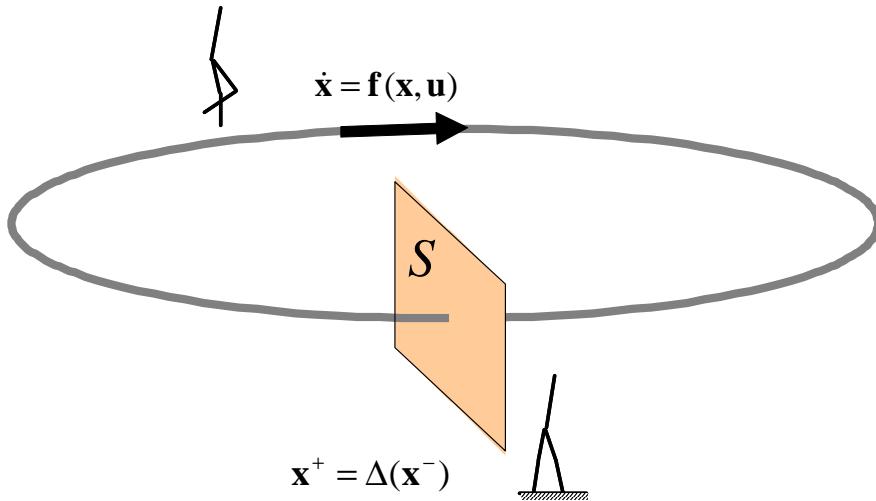


图 5.3 平面双足机器人行走混合动力学模型

Fig. 5.3 A hybrid dynamic model for a planar biped robot walking.

5.2.2 轨迹优化

我们将行走控制问题转化为无限时域的最优控制问题。为产生更加自然、高效和鲁棒的行走控制，代价函数需要通过多次尝试来确定。代价率函数中考虑了关节力矩、行走速度和地面作用力， $c(\mathbf{x}, \mathbf{u}) = \mathbf{u}^\top \mathbf{W}_u \mathbf{u} + w_v(v_h(\mathbf{x}) - v_d)^2 + w_r f_x(\mathbf{x})^2$ ，其中 \mathbf{W}_u ， w_v 和 w_r 是代价权重， v_d 和 $v_h(\mathbf{x})$ 分别为期望行走速度和髋关节水平移动速度。行走过程中需要考虑摆动脚的高度，因此当摆动脚高度低于 h_d 时引入代价 $w_c(p_2^v - h_d)^2$ ，其中 w_c 为代价权重， p_2^v 为摆动脚的高度。行走过程中为减少上身的运动，在代价函数中增加 $w_t \dot{q}_t^2$ ，其中 w_t 为代价权重， $q_t = q_1 + q_2 + q_3$ 为上身相对于竖直方向的倾角。

为了保持支撑脚相对地面静止，在控制过程中需要始终满足支撑脚与地面接触的物理约束。为防止支撑脚相对地面打滑，地面作用力必须始终处于摩擦力锥体内 [24]：

$$\left| \frac{f_x}{f_z} \right| \leq \mu, \quad (5.22)$$

其中 f_x 为水平方向的摩擦力， f_z 为垂直方向的地面作用力， μ 为脚与地面之间的干摩擦系数。为防止支撑脚相对地面的意外翻转，需要满足 [23]：

$$-\frac{u_1}{f_z} \in \Omega, \quad (5.23)$$

其中 u_1 为支撑脚踝关节驱动力矩， Ω 为支撑脚的支撑区域。地面作用力可以通过逆运动学方程表示为状态变量和控制变量的函数， $f_x(\mathbf{x}, \mathbf{u})$ 和 $f_z(\mathbf{x}, \mathbf{u})$ ，在实

表 5.2 轨迹优化采用的参数

Table 5.2 Parameters used by trajectory optimization

\mathbf{W}_u	diag($10^{-2}, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}$)	w_v	10^2	w_r	10^{-6}	h_d	0.05
w_I	10^{-1}	w_c	10^4	β	0.5	w_t	10

现时通过 SDFAST 软件得到。在轨迹优化时通过惩罚 u_1 和 f_x 的大小以满足公式 (5.22) 和 (5.23)。除此之外将机器人自身存在的关节角和关节角速度的限制作为状态变量约束，将关节驱动力矩限制作为控制变量约束。

5.2.3 轨迹库生成方法

采用稳态轨迹采样状态上的控制律局部近似模型，只能实现稳态轨迹邻域内的局部状态反馈控制。当状态远离稳态轨迹时，局部控制律的性能会有所下降。为了有效地控制更大的状态空间，可以采用大量非周期轨迹片段构成轨迹库来覆盖所有可能的状态空间。采用最近邻控制时，同时采用这些轨迹片段采样状态上的控制律局部模型，从而实现更大状态空间上的反馈控制 [78, 82, 97, 103]。

为产生轨迹片段，选取已有轨迹附近的一个状态作为初始状态，采用已有轨迹上的局部模型和最近邻控制生成初始轨迹，然后采用微分动态规划方法对初始轨迹进行优化，在得到最优轨迹的同时，生成新的控制律局部近似模型。

为了确定生成轨迹片段的初始状态，本文可以采用“失败测试”法，即在仿真机器人行走过程中，不同时间、不同位置施加不同的瞬间扰动，如果仿真机器人在瞬间扰动后摔倒，则以此瞬间扰动后的状态作为初始状态生成新的轨迹片段。

仿真中公式 (5.17) 中的距离加权矩阵取对角阵，其中关节角和关节角速度对应的元素分别为 1 和 10^{-4} 。为了防止支撑脚相对地面翻转从而满足模型假设，仿真中对由公式(5.18) 计算得到的踝关节驱动力矩在应用前加以限制 $u_1/f_z \in \Omega$ 。

5.2.4 仿真结果

在求解稳态轨迹时，参数化轨迹优化采用 $N = 20$ ，微分动态规划采用 $N = 100$ 。采用表 5.2 中的参数。对于期望行走速度为 0.5m/s 的行走控制，图

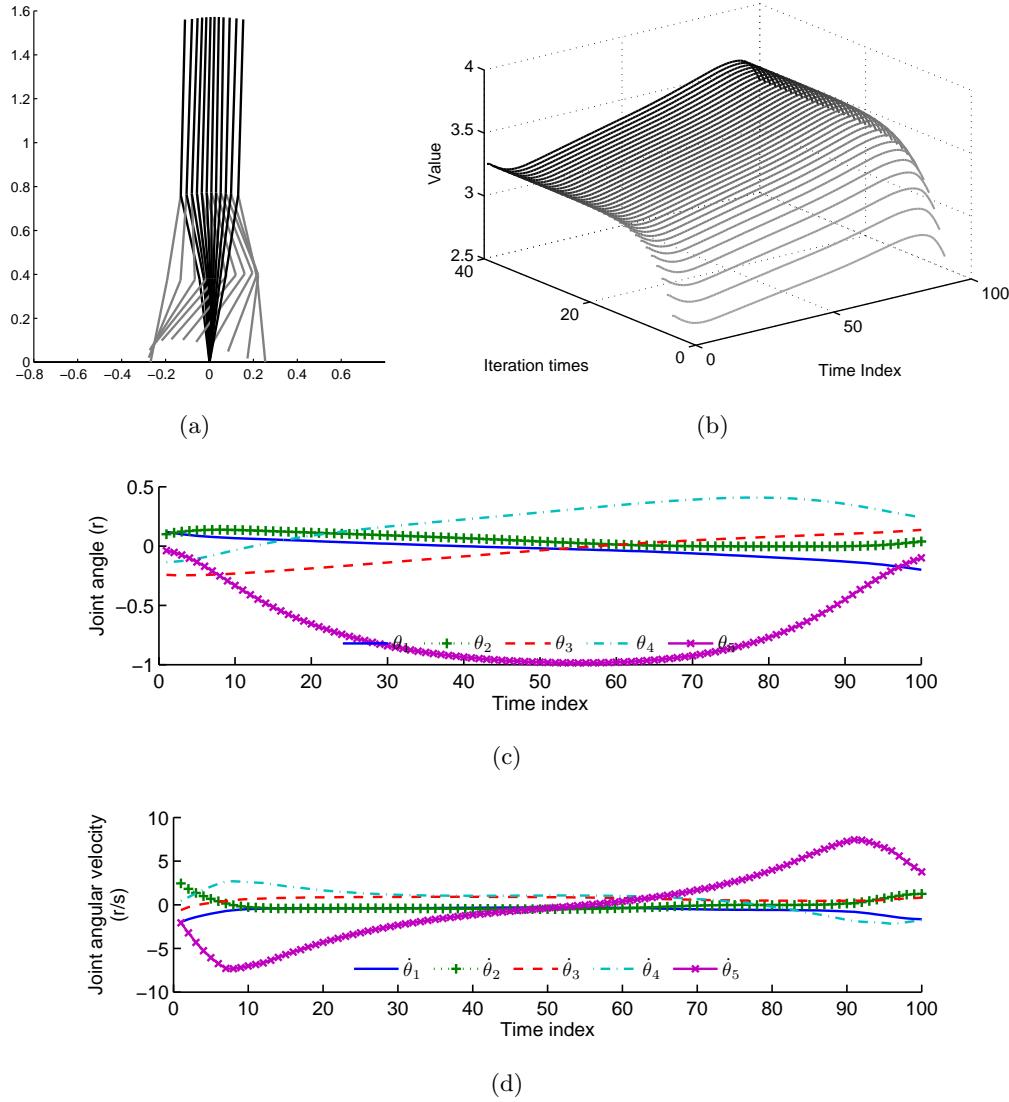


图 5.4 期望行走速度为 0.5m/s 的稳态轨迹 a) 采用稳态轨迹的机器人姿态 (截图时间间隔为 50 毫秒) b) 每次迭代后稳态轨迹上的值函数 c) 对应的各关节角轨迹 d) 对应的各关节角速度轨迹

Fig. 5.4 The steady-state trajectory at 0.5 m/s desired walking speed a) The corresponding configuration of the robot motion (the configuration of the robot is drawn every 50 ms) b) The evolution of the value function on the steady-state trajectory after each iteration c) The corresponding joint angle trajectories d) The corresponding joint angular velocity trajectories

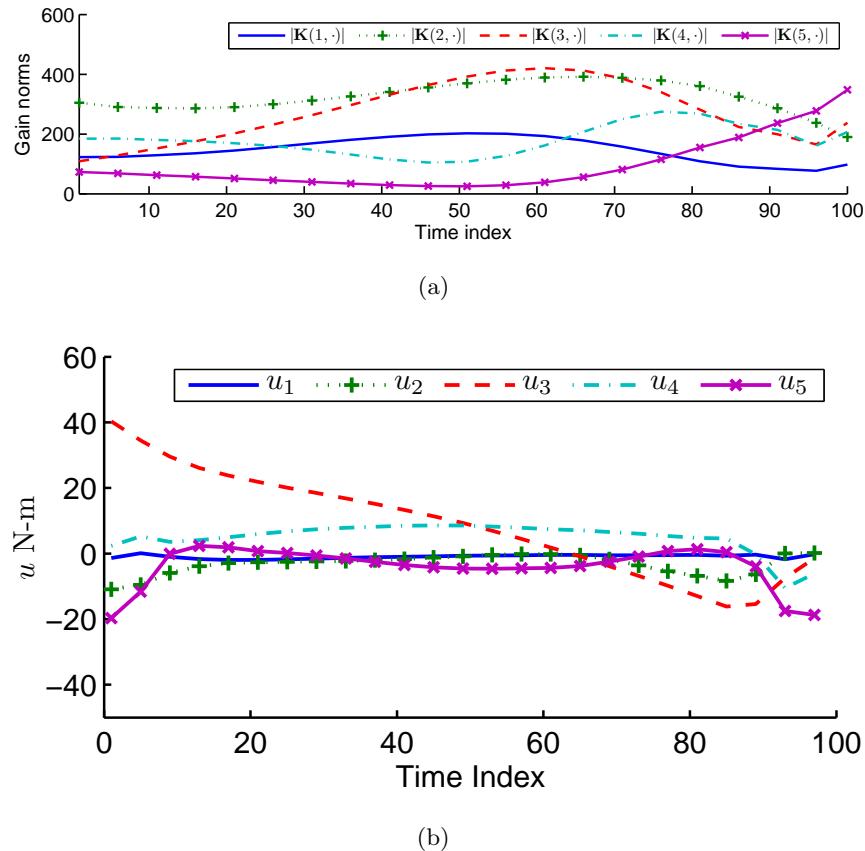


图 5.5 期望行走速度为 0.5m/s 时的稳态轨迹 a) 每个控制量对应的增益向量的模 b) 对应的各关节力矩

Fig. 5.5 The steady-state trajectory at 0.5 m/s desired walking speed a) The norms of the feedback gains of each control b) The corresponding joint torques

5.4 为稳态轨迹优化的结果。通过选择合适的代价权重 w_t 和 w_c ，机器人在行走过程中上身倾斜运动很小，摆动脚抬起足够的安全高度。图 5.5 显示了最优轨迹采样状态上的增益矩阵和关节力矩随时间的变化，其中 u_1 , u_2 , u_3 为支撑腿的踝关节、膝关节和髋关节力矩， u_4 和 u_5 为摆动腿髋关节和膝关节力矩。控制量对应的增益向量 $\mathbf{K}(i, \cdot)$ 为 $u_i = \bar{u}_i - \mathbf{K}(i, \cdot)(\mathbf{x} - \bar{\mathbf{x}})$ 。轨迹优化的结果中，最优轨迹采用的支撑腿踝关节力矩一直很小，其他关节力矩变化比较平缓。

在下面的仿真中，为了反映 10 维状态空间上的机器人运动，将髋关节与某只脚连线构成虚拟腿，然后考察虚拟腿相对垂直方向的运动，如图 5.6。在得到了稳态轨迹后，以不同时间作用在胯关节处 -4Ns 和 10Ns 冲击扰动后的状态为初始状态，通过轨迹优化得到 12 条最优轨迹片段，由这些轨迹片段和稳态轨迹及采样状态上的控制律局部近似模型一起构成轨迹库，如图 5.7(a) 所示。

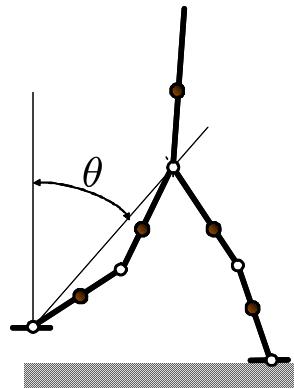


图 5.6 采用虚拟腿相对垂直方向的运动反映机器人在10维状态空间上的运动。

Fig 5.6 The motion of a virtual leg w.r.t. the vertical direction is used to reflect the motion of the robot in 10D state space.

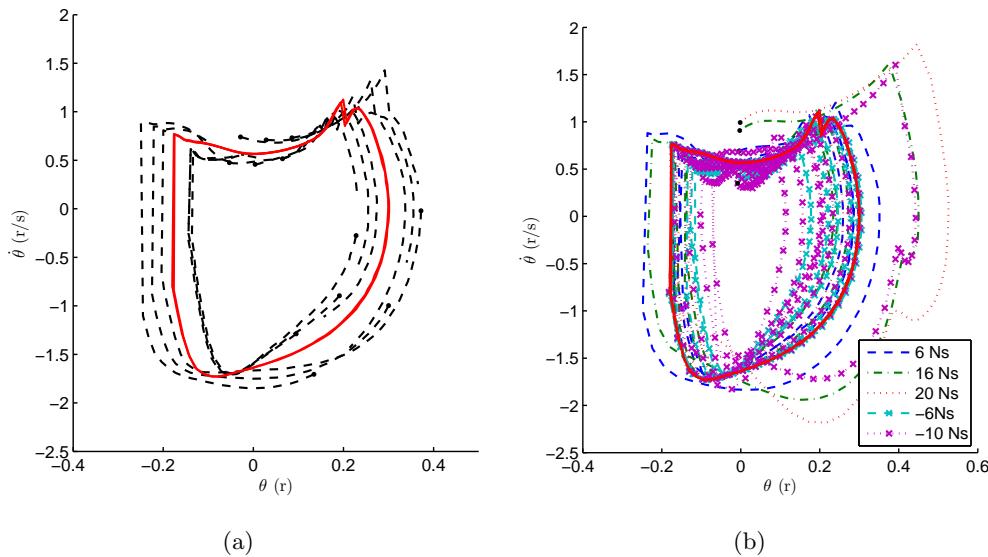


图 5.7 行走控制轨迹库的例子及行走中对瞬间扰动的响应。相图中的不连续是因为冲击扰动或脚与地面的碰撞引起的。a) 轨迹库的例子。虚线为最优轨迹片段, 实线为周期性稳态轨迹。b) 不同大小的瞬间扰动的响应

Fig. 5.7 A example trajectory library and responses to impulsive perturbations during walk. the discontinuities are caused by impulsive perturbations or foot-ground collisions. a) An example trajectory library. The dashed lines are optimal trajectory segments and the solid line is the steady-state trajectory. b) Responses to impulsive perturbations of different sizes.

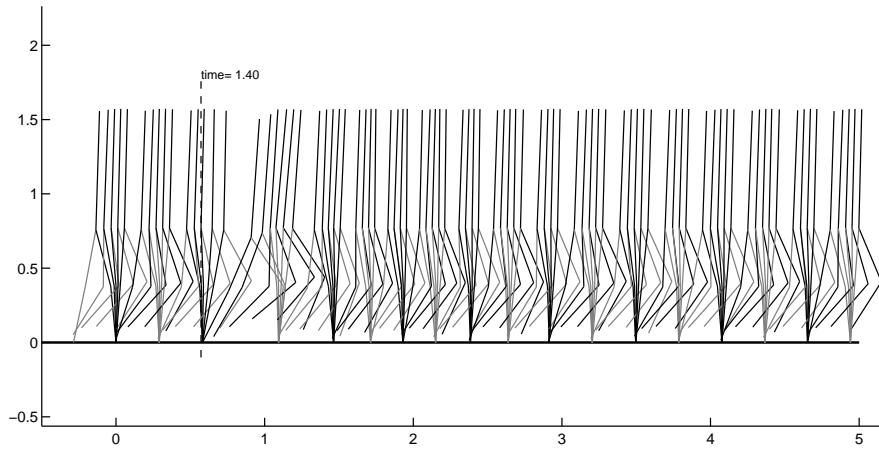


图 5.8 行走中在 1.4 秒时受到 16Ns 扰动的姿态响应

Fig. 5.8 The robot's configuration responding to a 16 Newton-seconds perturbation at 1.4 seconds during walk.

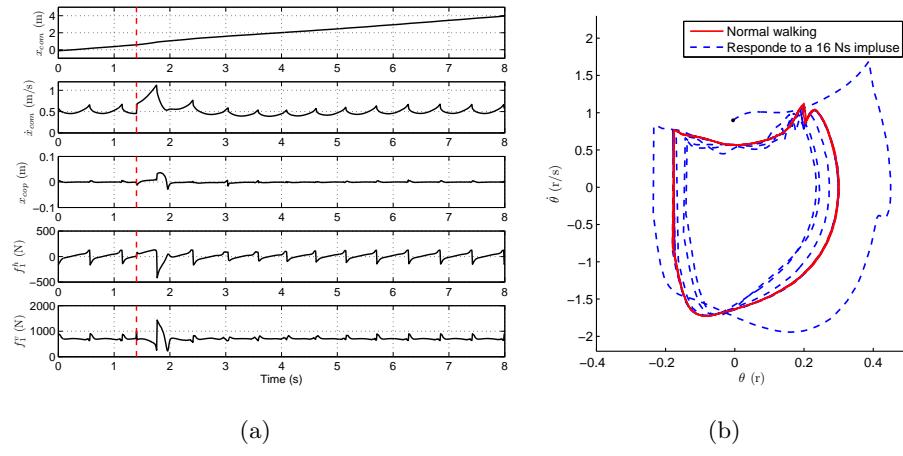


图 5.9 行走中在 1.4 秒时受到 16Ns 瞬间扰动的响应。a) 质心的水平位置和速度、CoP 的水平位置和地面作用力。红色垂线代表扰动发生的时间。b) 虚拟腿相对垂直方向运动的相图。

Fig. 5.9 Responses to a 16 Newton-seconds impulsive perturbation at 1.4 seconds during walk. a) The horizontal position and velocity of the center of mass, the horizontal position of CoP, and the ground reaction forces. The vertical red lines denote when the perturbation happens. b) The phase portrait of the motion of one virtual leg w.r.t. the vertical direction.

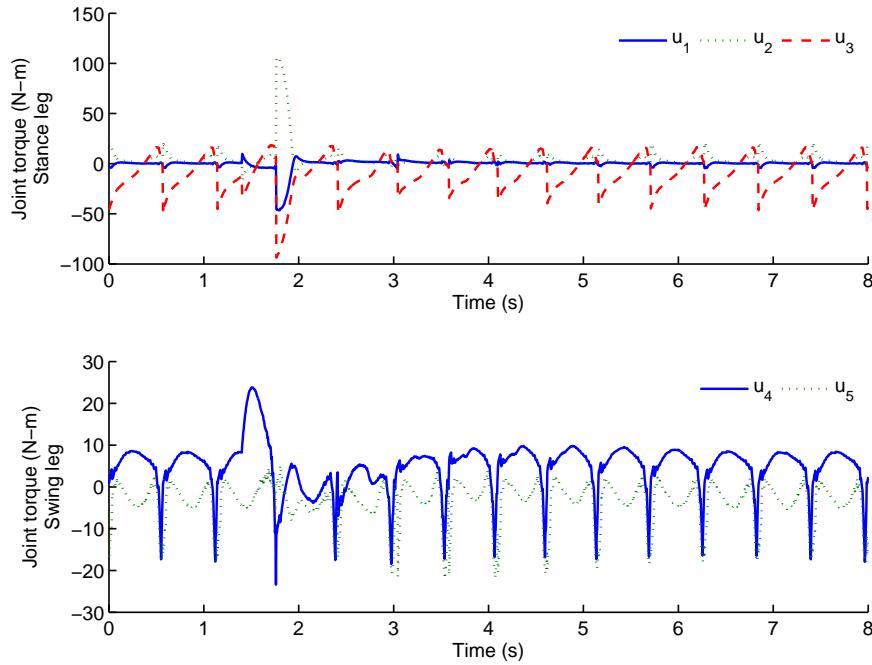


图 5.10 行走中在1.4秒时受到 16Ns 扰动的关节力矩响应。

Fig. 5.10 Responses of the joint torques to a 16 Newton-seconds perturbation at 1.4 seconds during walk.

下面通过行走控制对冲击扰动的响应，验证本文提出的行走控制方法的鲁棒性。将机器人的状态初始化到稳态轨迹上并以 0.5m/s 的速度行走，在跨关节施加 -10Ns , -6Ns , 6Ns , 16Ns 和 20Ns 的冲击扰动，以虚拟腿相对垂直方向的运动研究控制器的响应。仿真结果如图 5.7(b)所示，对于不同大小的冲击扰动本文提出的控制器均能使机器人稳定到稳态轨迹上。其中对于 16Ns 的瞬间扰动，图 5.8 为机器人的姿态响应，其中姿态响应截图的时间间隔为 0.1 秒。在冲击发生后，机器人快速向前迈步并逐渐恢复正常的速度。机器人质心的水平位置和速度、CoP 的水平位置和地面作用力响应如图 5.9 所示。仿真结果说明：行走控制始终满足地面作用力约束；冲击扰动瞬间改变了质心的水平速度，但是机器人能够逐渐恢复正常的速度并保持期望的速度 (0.5 m/s)。图 5.10 为各关节力矩响应，冲击发生时支撑腿膝关节和踝关节力矩迅速增加，起到了减速的作用。

进一步验证行走控制的鲁棒性，进行了在倾斜度为 5° 上升斜面上的行走仿真。图 5.11 为机器人在斜面上行走的姿态响应，这个为在水平地面上行走设计的控制器同样能够在斜面上行走。图 5.12 显示了此时质心的水平位置、速度，

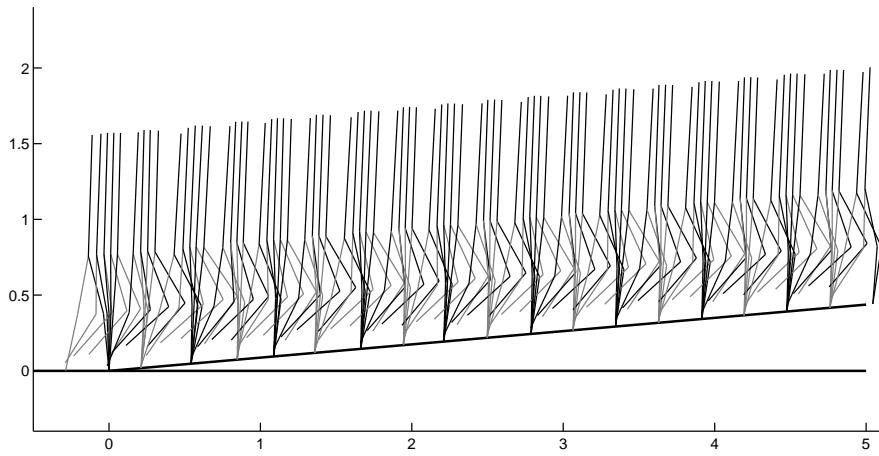
图 5.11 机器人在 5° 上升斜面上行走时的姿态

Fig. 5.11 The robot's configuration of walking on an incline of 5 degrees.

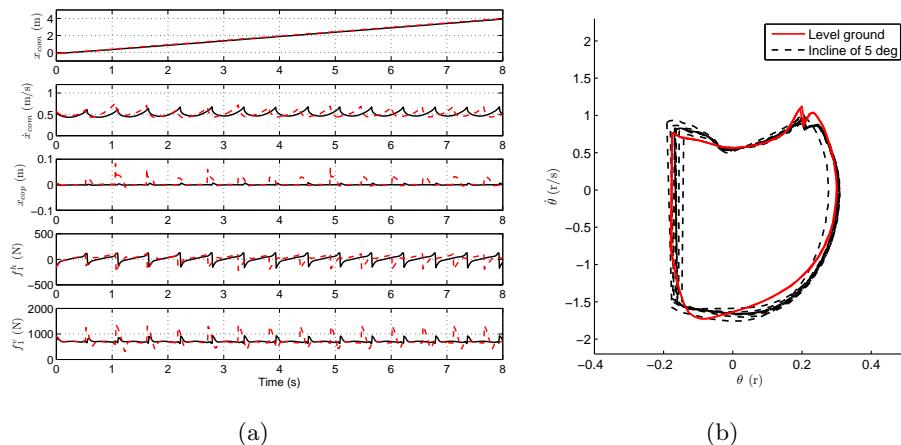
图 5.12 在倾斜度为 5° 上升斜面上行走 a) 质心的水平位置和速度、CoP 的水平位置和地面作用力。b) 虚拟腿相对垂直方向运动的相图。

Fig. 5.12 Walking on an incline of 5 degrees a) The horizontal position and velocity of the center of mass, the horizontal position of CoP, and the ground reaction forces. b) The corresponding phase portrait of the motion of one virtual leg w.r.t. the vertical direction.

CoP 的水平位置和地面作用力。在斜面上行走时，机器人的速度仍然能够保持期望速度 (0.5m/s)，不过虚拟腿运动的相图说明此时机器人的状态已收敛到一个新的极限环上。图 5.13 显示了各关节的力矩，在上升斜面上行走时支撑腿各关节的力矩有所增加。

对持续一段时间的外界扰动，图 5.14 为机器人对持续时间为 2 秒、大小为 10 牛顿的前向推力的姿态响应（推力的作用时间为 $3 \sim 5$ 秒）。图 5.15 显示了

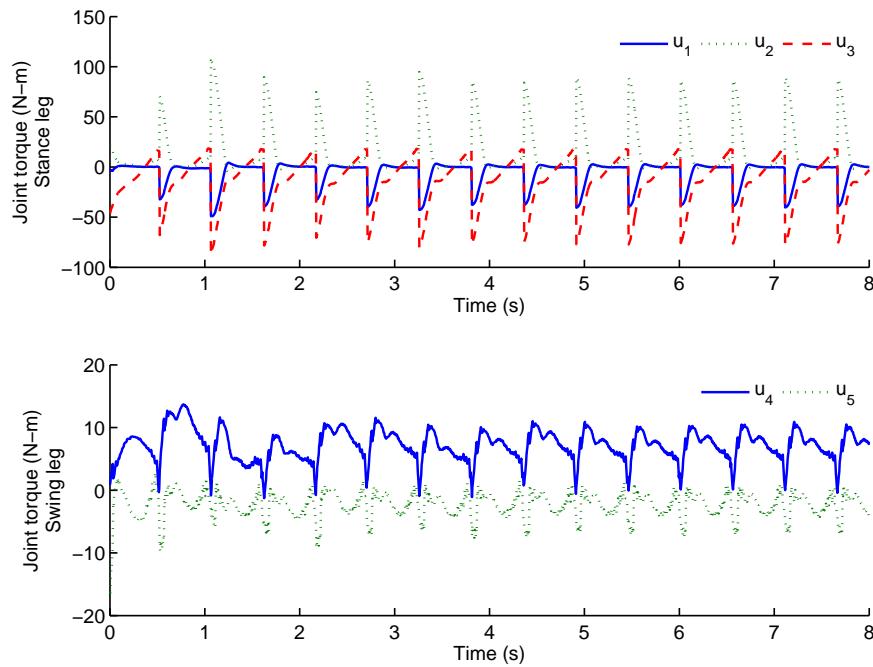


图 5.13 5° 斜面上行走时的关节力矩响应

Fig. 5.13 Responses of the joint torques during walking on an incline of 5 degrees

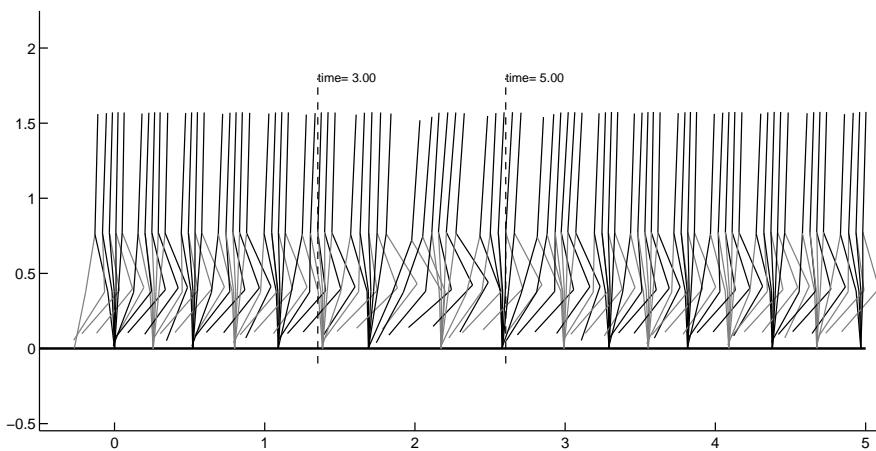


图 5.14 机器人在持续时间为2秒、大小为10牛顿的前向推力下的行走姿态（推力的作用时间为 3 ~ 5 秒）。

Fig. 5.14 The robot's configuration of walking under a continuous forward push of 10 Newtons lasting 2 seconds. The push is applied during 3 and 5 seconds.

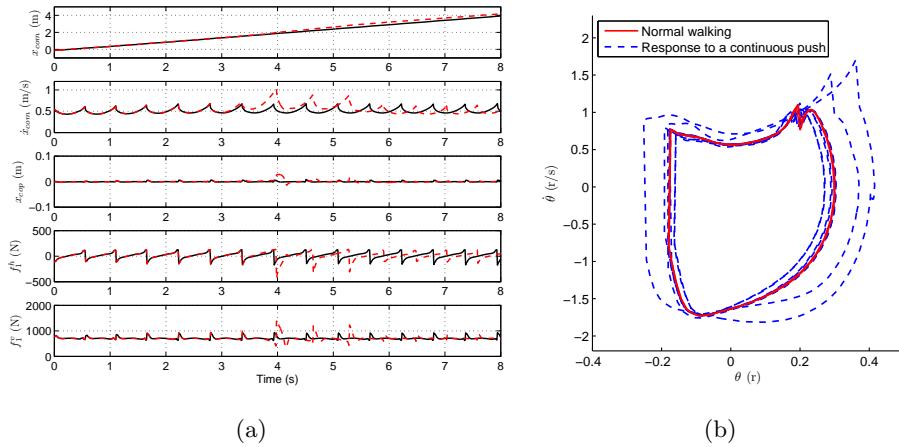


图 5.15 机器人在持续时间为2秒、大小为10牛顿的前向推力下的行走控制响应（推力的作用时间为3~5秒）。实线为正常行走的响应。a) 质心的水平位置和速度、CoP的水平位置和地面作用力。b) 虚拟腿相对垂直方向运动的相图。

Fig. 5.15 Responses to a continuous forward push of 10 Newtons lasting 2 seconds as dashed lines (the push is applied during 3 to 5 seconds). The solid lines are responses of normal walking. a) The horizontal position and velocity of the center of mass, the horizontal position of the CoP, and the ground reaction forces. b) The phase portrait of the motion of one virtual leg w.r.t. the vertical direction.

机器人质心的水平位置和速度，CoP的水平位置和地面作用力。在外力作用下，机器人的实际行走速度比期望行走速度快。虚拟腿运动的相图说明当扰动消失后，控制器能够将机器人重新稳定到期望的稳态轨迹上。

在系统建模的规程中模型误差是不可避免的，因此本文进一步考察了所提出的控制方法对模型误差的鲁棒性。图 5.16 为上身质量增加30%后正常行走的响应。图 5.16(a) 说明机器人依然能够行走，行走过程中地面作用力的垂直分量有所增加。在这个仿真中，机器人的行走速度基本保持不变。图 5.16(b) 说明模型误差导致机器人收敛到与周期性稳态轨迹不同的周期性轨迹上。

本文采用解析的脚-地面接触模型（刚体碰撞模型）并设计行走控制器，因此进一步研究了得到的控制器对其他地面接触模型的鲁棒性，如常用的弹簧-阻尼模型。在弹簧-阻尼模型中，将支撑脚与地面之间的关节建模为3自由度的平面关节，当某条腿的末端处于地面以下时，其受到的地面作用力通过下面的公

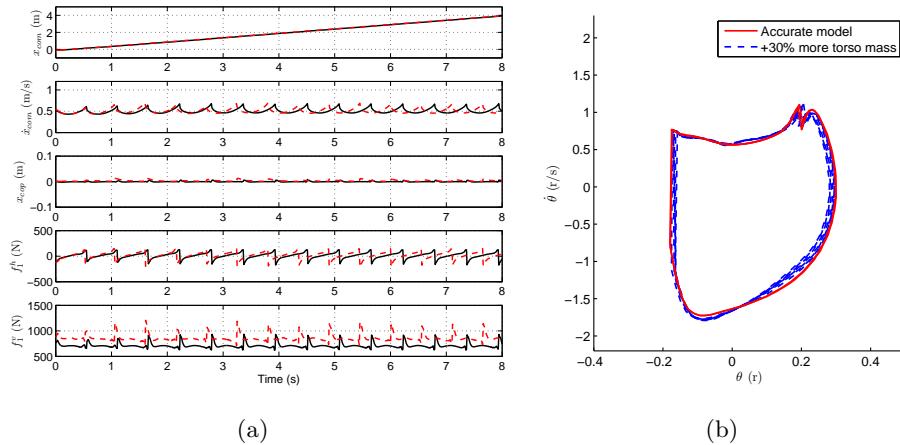


图 5.16 上身质量增加 30% 后的响应 (虚线)。实线表示采用精确模型时的行走控制响应。a) 质心的水平位置和速度、CoP 的水平位置和地面作用力。b) 一条虚拟腿运动相图。

Fig. 5.16 Responses to a model error of 30% more torso mass (dashed lines). The solid lines are responses of walking control with an accurate model. a) The horizontal position and velocity of the center of mass, the horizontal position of CoP, and the ground reaction forces. b) The phase portrait of the motion of one virtual leg.

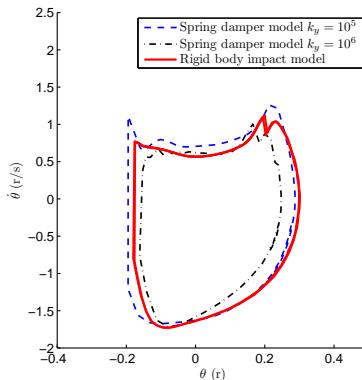


图 5.17 采用不同的脚-地面接触模型产生的虚拟腿运动相图

Fig. 5.17 Phase portraits of one virtual leg of normal walking using different foot-ground contact models.

式计算

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = - \begin{bmatrix} k_x & 0 & c_x & 0 \\ 0 & k_y & 0 & c_y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \dot{x} \\ \dot{y} \end{bmatrix}, \quad (5.24)$$

其中 f_x 和 f_y 分别为地面作用力的水平和垂直分量, k_x 和 k_y 为弹簧常数, c_x

和 c_y 为阻尼常数, Δx 和 Δy 分别是这条腿的末端在地面上滑移的距离和陷入地面的深度, \dot{x} 和 \dot{y} 为腿末端的水平和垂直运动速度。采用 $k_x = 10^4$, $c_x = 10^3$, $c_y = 10^3$ 和不同的 k_y 对行走控制进行仿真。仿真结果(图 5.17)说明本文得到的控制器对不同参数的弹簧-阻尼地面模型仍然有效。

基于轨迹库的行走控制器与基于轨迹跟踪的行走控制器进行了比较。为了能够进行比较,两种控制器都只采用了稳态轨迹一条轨迹。常见的基于轨迹跟踪的行走控制所有关节采用单独的 PD 加前馈控制跟踪参考轨迹, $\tau_j(t) = \tau_j^r(t) - k_{pj}[\theta_j(t) - \theta_j^r(t)] - k_{dj}[\dot{\theta}_j(t) - \dot{\theta}_j^r(t)]$, 其中 $\tau_j(t)$, $\theta_j(t)$ 和 $\dot{\theta}_j(t)$ 分别为第 j 个关节的控制力矩、转角和角速度, $\tau_j^r(t)$, $\theta_j^r(t)$ 和 $\dot{\theta}_j^r(t)$ 为参考轨迹上相应的值, 增益 k_{pj} 和 k_{dj} 通过参数优化得到。我们比较了两种控制器 6 秒时间内代价。让仿真机器人从参考轨迹上某一状态开始行走, 在没有扰动的情况下, 后者的代价为 3.661, 而前者的代价为 2.951。这个结果说明, 即使将机器人的状态初始化为参考轨迹上的某一状态然后跟踪这条参考轨迹, 基于轨迹跟踪的行走控制的代价也远大于本文提出的方法。究其原因, 在轨迹跟踪的过程中, 跟踪误差是不可避免的。基于轨迹跟踪的行走控制采用固定增益跟踪参考轨迹, 而本文提出的控制器根据当前状态在线计算最优控制, 因此本文提出的控制方法能够得到代价更小的响应。在跟踪单一轨迹方面, 本文提出的方法已经显示出性能的优势, 因此与跟踪多条参考轨迹的行走控制(如 [81])相比, 本文提出的行走控制方法也将同样具有性能的优势。让仿真机器人从参考轨迹上某一状态开始行走, 在行走过程中施加 5Ns 的扰动, 后者的代价为 21.679, 而前者为 3.697。这个结果进一步说明, 与基于轨迹跟踪的行走控制相比, 本文所提出的控制方法在性能方面具有优越性。

5.3 三维七连杆行走控制

在二维行走控制的基础上, 本文进一步研究了考虑双腿支撑的三维行走控制问题。三维双足行走机器人如图 5.18 所示。三维行走中一个行走周期包括左腿单腿支撑、双腿支撑(左腿)、右腿单腿支撑和双腿支撑(右腿)。三维行走的

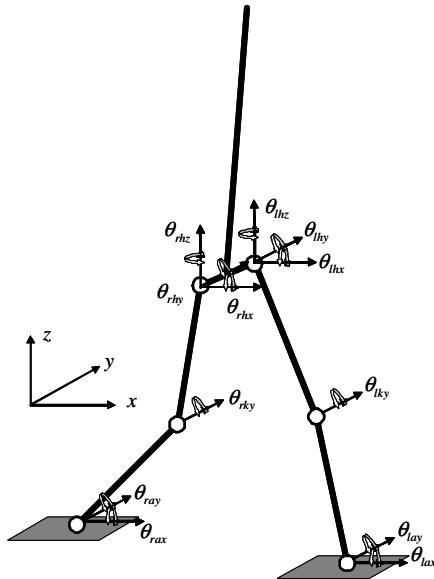


图 5.18 三维七连杆双足机器人简化模型

Fig. 5.18 Simplified structure of a 3D 7-link biped robot

动力学方程同样可以通过混合动力学模型描述:

$$\begin{cases} \dot{\mathbf{x}}_{lss} = \mathbf{f}_{lss}(\mathbf{x}_{lss}, \mathbf{u}_{lss}) \\ \mathbf{x}_{lds}^+ = \Delta_1(\mathbf{x}_{lss}^-) & \text{if } \mathbf{x}_{lss}^- \in \mathcal{S}_1 \\ \dot{\mathbf{x}}_{lds} = \mathbf{f}_{lds}(\mathbf{x}_{lds}, \mathbf{u}_{lds}) \\ \mathbf{x}_{rss}^+ = \Delta_2(\mathbf{x}_{lds}^-) & \text{if } \mathbf{x}_{lds}^- \in \mathcal{S}_2 \\ \dot{\mathbf{x}}_{rss} = \mathbf{f}_{rss}(\mathbf{x}_{rss}, \mathbf{u}_{rss}) \\ \mathbf{x}_{rds}^+ = \Delta_3(\mathbf{x}_{rss}^-) & \text{if } \mathbf{x}_{rss}^- \in \mathcal{S}_3 \\ \dot{\mathbf{x}}_{rds} = \mathbf{f}_{rds}(\mathbf{x}_{rds}, \mathbf{u}_{rds}) \\ \mathbf{x}_{lss}^+ = \Delta_4(\mathbf{x}_{rds}^-) & \text{if } \mathbf{x}_{rds}^- \in \mathcal{S}_4 \end{cases} \quad (5.25)$$

其中 lss 、 lds 、 rss 和 rds 分别代表左腿单腿支撑、双腿支撑（左腿）、右腿单腿支撑和双腿支撑（右腿）。虽然仿真机器人没有脚，但在行走的过程中可以在踝关节施加一定的转矩，因此 $\mathbf{x}_{lss} \in \mathcal{R}^{20}$, $\mathbf{u}_{lss} \in \mathcal{R}^{10}$, $\mathbf{x}_{lds} \in \mathcal{R}^{12}$, $\mathbf{u}_{lds} \in \mathcal{R}^6$, $\mathbf{x}_{rss} \in$

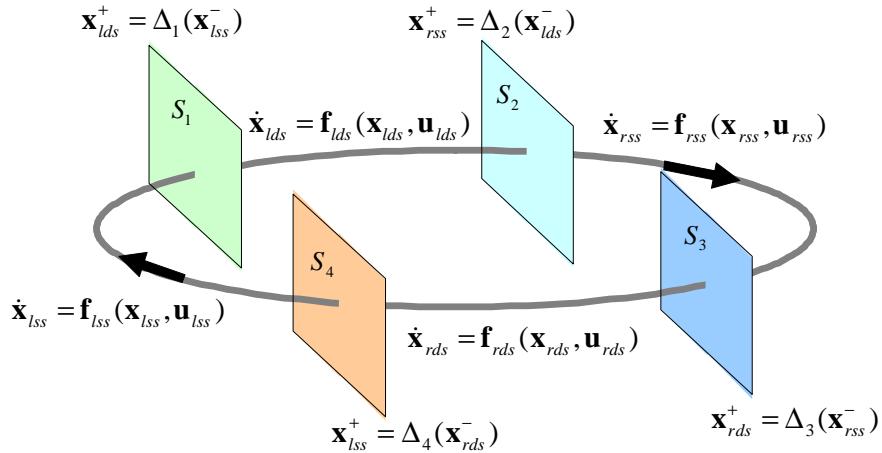


图 5.19 三维行走混合动力学模型
Fig. 5.19 A hybrid dynamic model of 3D walking

\mathcal{R}^{20} , $\mathbf{u}_{rss} \in \mathcal{R}^{10}$, $\mathbf{x}_{rds} \in \mathcal{R}^{12}$, $\mathbf{u}_{rds} \in \mathcal{R}^6$ 。对于水平地面,

$$\mathcal{S}_1 = \{\mathbf{x}_{lss} | p_r^v(\mathbf{x}_{lss}) = 0, \dot{p}_r^v(\mathbf{x}_{lss}) < 0\} \quad (5.26)$$

$$\mathcal{S}_2 = \{\mathbf{x}_{lds} | p_l^v(\mathbf{x}_{lds}) = 0, \dot{p}_l^v(\mathbf{x}_{lds}) > 0\} \quad (5.27)$$

$$\mathcal{S}_3 = \{\mathbf{x}_{rss} | p_l^v(\mathbf{x}_{rss}) = 0, \dot{p}_l^v(\mathbf{x}_{rss}) < 0\} \quad (5.28)$$

$$\mathcal{S}_4 = \{\mathbf{x}_{rds} | p_r^v(\mathbf{x}_{rds}) = 0, \dot{p}_r^v(\mathbf{x}_{rds}) > 0\}, \quad (5.29)$$

其中 p_l^v 和 p_r^v 分别为左脚和右脚离地面的高度。三维行走可描述为力学不连续系统的周期性运动, 其混合动力学模型如图 5.19 所示。

为了优化稳态轨迹, 将单腿支撑阶段离散化为 $N_{lss} = N_{rss} = 40$, 将双腿支撑阶段离散化为 $N_{lds} = N_{rds} = 10$ 。为了加快优化收敛的速度, 将每个阶段的时间设为固定值、所有阶段的 $\Delta T_i = 0.01$ 秒。优化的代价率函数需要进行反复调节, 为减小关节力矩, $\sum_{i=1}^r (10\tau_{iay}^2 + 10\tau_{iay}^2 + 0.1\tau_{iky}^2 + 0.1\tau_{ihx}^2 + 0.1\tau_{ihy}^2 + 10\tau_{ihz}^2)$; 为减小地面摇首 (yaw) 转矩, $0.01\tau_z^2$; 为减小上身摇摆, $10\omega_{tx}^2 + 10\omega_{ty}^2 + \omega_{tz}^2$; 为减小摆动脚旋转, $100\omega_{ax}^2 + 100\omega_{ay}^2 + 100\omega_{az}^2$; 为减小上身运动, $10(v_{tx} - v_d)^2 + 10v_{ty}^2 + 0.1v_{tz}^2$, $v_d = 0.5$ 为期望行走速度, 同时还包括摇摆脚轨迹跟踪的代价等。取折扣率 $\beta = 5$, 采用参数化轨迹优化方法优化稳态轨迹, 图 5.20 为稳态轨迹优化结果, 图 5.21 为对应的各关节角, 图 5.22 为各关节角速度, 图 5.23 为各关节力矩, 图 5.24 为行走过程中地面作用力。优化产生的步态与人类的自然行走非常类似。

在得到周期性稳态轨迹以后, 采用 5.1.3 提出的方法计算力学不连续处的

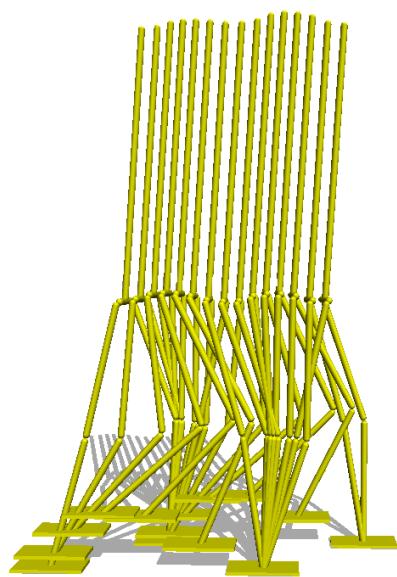


图 5.20 采用期望行走速度 0.5 米/秒的稳态轨迹的机器人姿态 (截图时间间隔为 50 毫秒)。

Fig. 5.20 The configuration of the robot motion using the steady-state trajectory at 0.5 m/s desired walking speed (it is drawn every 50 ms).

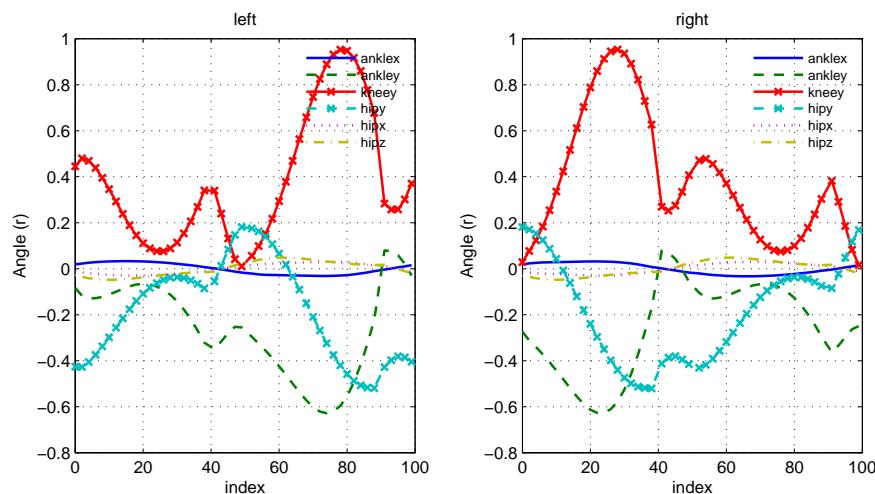


图 5.21 期望行走速度 0.5 米/秒的稳态轨迹对应的关节角

Fig. 5.21 Joint angles of the steady-state trajectory at 0.5 m/s desired walking speed

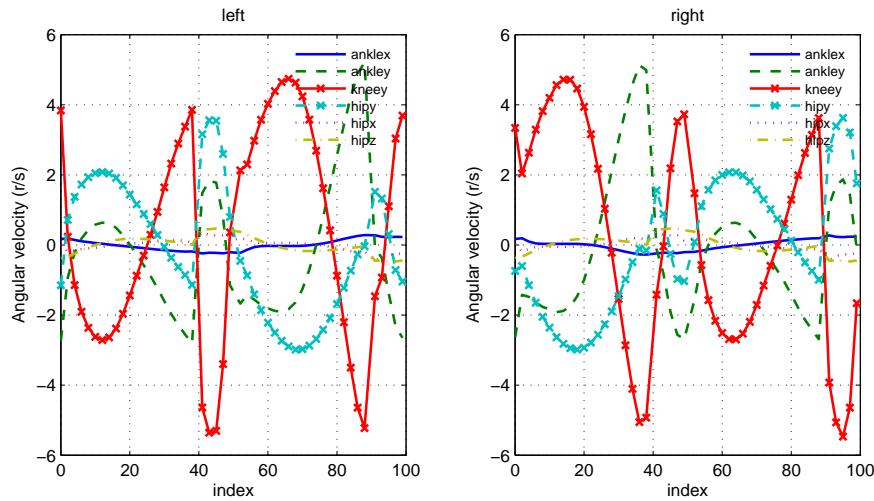


图 5.22 期望行走速度 0.5 米/秒的稳态轨迹对应的关节角速度

Fig. 5.22 Joint angular velocities of the steady-state trajectory at 0.5 m/s desired walking speed

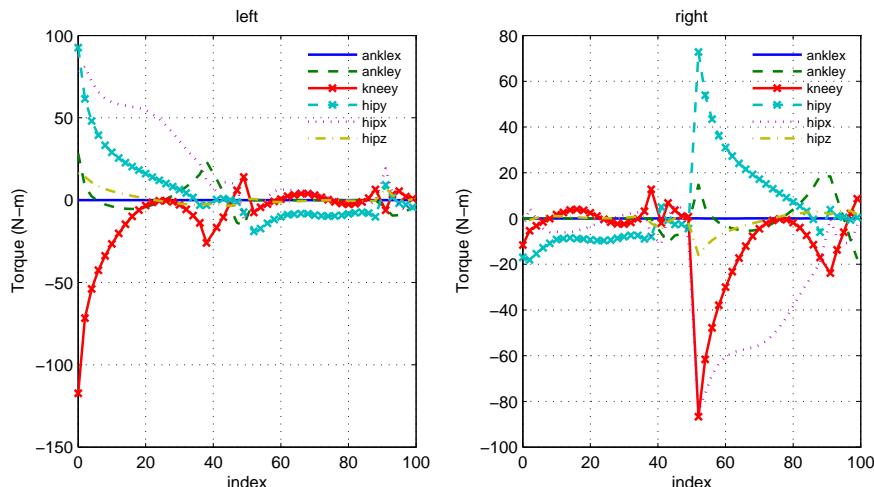


图 5.23 期望行走速度 0.5 米/秒的稳态轨迹对应的关节力矩

Fig. 5.23 Joint torques of the steady-state trajectory at 0.5 m/s desired walking speed

值函数传递关系，并利用微分动态规划得到周期性稳态轨迹采样状态上控制率的局部近似模型。利用这些局部近似模型和最近邻控制，实现了三维七连杆机器人周期性稳态轨迹邻域内的最优行走控制。求得的局部近似模型中增益矩阵各元素的值如图 5.25 所示。稳态轨迹采样状态处值函数的梯度各元素的值如图 5.26 所示。单脚支撑阶段采样状态值函数的梯度在接近单足-双足支撑转换时会出现比较明显的变化，结果导致增益矩阵各元素此时同样出现比较明显的变化。

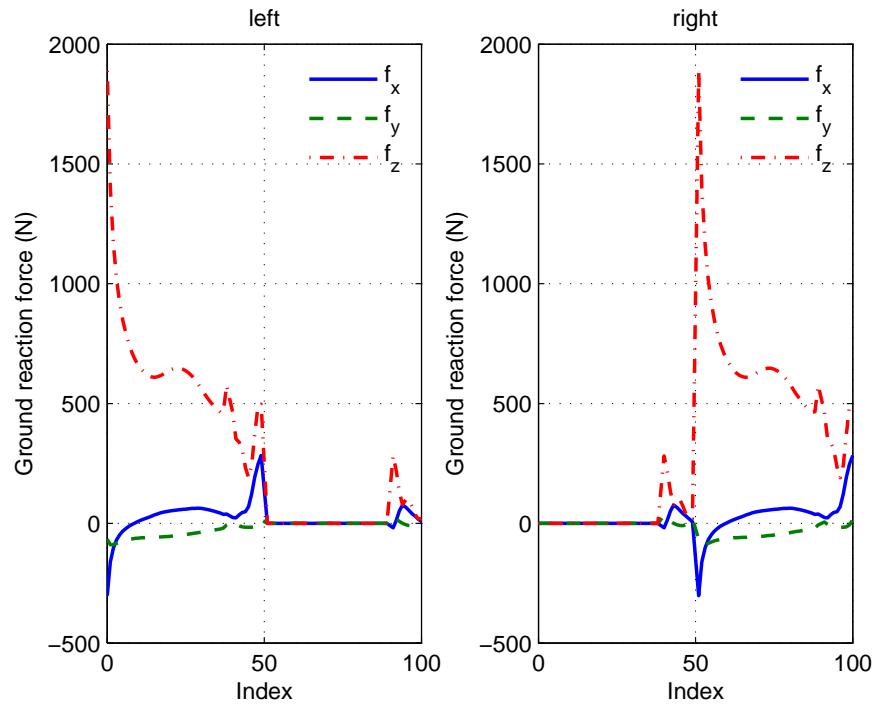


图 5.24 期望行走速度 0.5 米/秒的稳态轨迹对应的地面作用力

Fig. 5.24 The ground reaction forces of the steady-state trajectory at 0.5 m/s desired walking speed

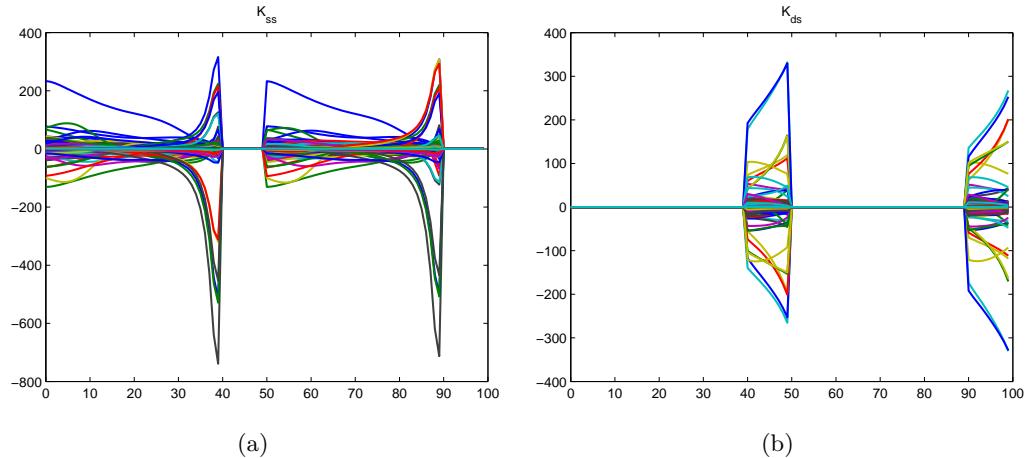


图 5.25 稳态轨迹采样状态处的控制律局部近似模型的增益矩阵 a) 单脚支撑阶段 b) 双脚支撑阶段

Fig. 5.25 The gain matrices of the local models of the control law on the sampled states of the steady-state trajectory. a) Single support phase. b) Double support phase.

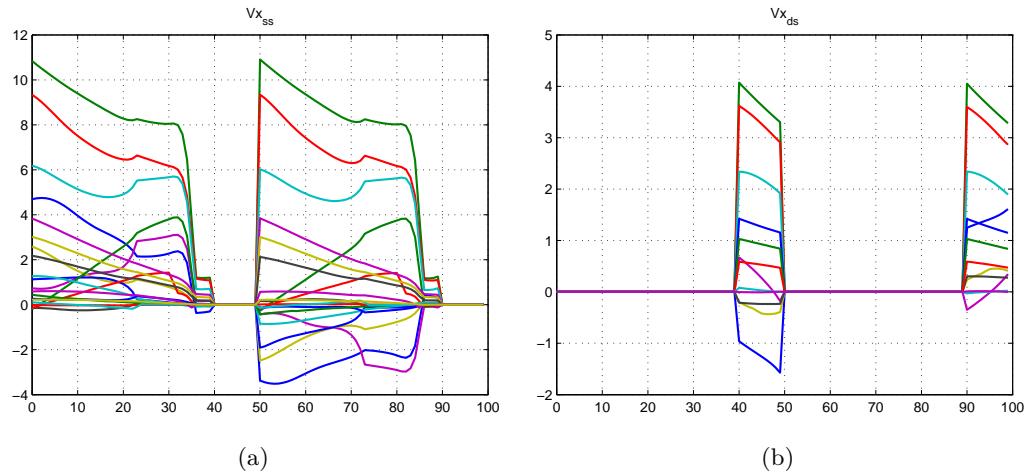


图 5.26 稳态轨迹采样状态处值函数的梯度 a) 单脚支撑阶段 b) 双脚支撑阶段

Fig. 5.26 The gradients of the value function on the sampled states of the steady-state trajectory. a) Single support phase. b) Double support phase.

分析其原因可能是因为采用刚体碰撞模型和固定单足支撑时间长度造成的。

5.4 本章小结

本章首先研究了具有一般意义的动力学不连续系统的周期性任务的最优控制问题。针对目前采用动态规划方法实现中存在的问题，提出一种邻域最优控制方法。通过将问题转化为无限时域的最优控制问题，采用参数化轨迹优化方法得到周期性任务中稳态轨迹。在推导动力学不连续处值函数的递推关系的基础上，采用微分动态规划得到更加精确的稳态轨迹和轨迹邻域内控制律时不不变局部近似模型。对于行走控制，首先研究了忽略双腿支撑的平面七连杆行走控制。摆动脚与地面之间的间歇性碰撞，使系统的动力学不连续。因此将双足行走转化为动力学不连续系统的周期性任务，将行走控制转化为无限时域的最优控制问题，采用参数化轨迹优化方法和微分动态规划方法生成稳态轨迹和轨迹采样状态上最优控制律的局部近似模型，为了得到更大状态空间上的状态反馈控制，采用微分动态规划生成大量轨迹片段，与稳态轨迹一起构成轨迹库，从而实现了基于轨迹库的行走控制。通过瞬间扰动下的行走控制、斜面上的行走控制和从静止开始行走的仿真说明方法的可行性和鲁棒性。与基于轨迹跟踪的行走控制进行了对比，结果说明基于轨迹库的行走控制方法能产生更小的驱动转矩、具有更大的鲁棒性。

本章提出的基于轨迹库的行走控制与跟踪参考轨迹的行走控制相比：1) 前

者采用的不是一条轨迹，而是大量轨迹。在控制的过程中，根据当前状态选择不同的参考轨迹；2) 前者采用控制律的局部近似模型的目的不是为了使系统跟踪某条选中的参考轨迹，而是近似最优控制；3) 因为得到的是近似最优控制，所以前者在轨迹跟踪和轨迹切换方面具有性能的优势。本文提出的方法与基于动态规划的方法相比：1) 轨迹优化受状态维数的影响比较小，能够对维数较高的机器人全模型进行优化；2) 轨迹优化能够更快速地得到控制律，而不需要像动态规划方法那样只有先得到值函数才能得到控制律。

第六章 总结与展望

6.1 总结

仿人机器人可以适应人类的生活和工作环境，代替人类完成各种任务，在很多方面扩展人类的能力，因此将在很多领域有广阔的应用前景。仿人机器人运动控制水平的高低是决定其能否走向实用的重要因素。虽然这方面已经取得了大量成果，但是目前的运动控制方法在效率、鲁棒性等方面与人类相比仍有很大差距，存在很大的改善余地。仿人机器人结构非常复杂，因此凭借设计者经验为其设计各种参数化运动控制器不仅非常繁琐，而且很难保证最终得到的控制器性能；而目前采用最优控制方法实现的仿人机器人运动控制器，普遍存在模型过于简单和计算时间过长等问题。随着计算机运算速度的不断提高和存储价格的不断下降，基于数据存储的非参数化控制器受到越来越多地关注。此外，更加先进的优化算法不断涌现，使求解更加复杂的优化问题成为可能。研究新的基于优化的仿人机器人运动控制方法因此具有重要的意义。

对仿人机器人而言，扰动下的站立平衡控制是运动控制研究的基本问题，双足行走是其区别于其他类型机器人的重要特征，因此本文重点研究了基于优化的仿人机器人站立平衡控制和行走控制问题，着眼于提高控制的效率和鲁棒性。所取得的成果和结论主要体现在以下几个方面：

1. 从轨迹优化和构造控制律两个方面，研究了基于轨迹库的控制方法。在轨迹优化方面，研究了多种轨迹优化方法，分析了其各自的特点。在此基础上，提出了参数化轨迹优化方法和 DDP（微分动态规划）结合的混合轨迹优化方法。既解决了参数化轨迹优化方法存在的结果不精确和不能严格满足动力学约束问题，又解决了 DDP 方法收敛对初始轨迹敏感的问题，得到了可直接用于控制的最优轨迹和轨迹采样状态上控制律的局部近似模型。另外，研究了轨迹优化中存在的约束处理、振铃现象、稳态值函数近似和局部最优等问题，提出了解决的办法。在构造控制律方面，总结了采用轨迹库实现控制的方法，包括轨迹跟踪法、在线优化法和函数近似法，分析了其各自的特点。与其他方法相比，采用控制律局部近似模型实现反馈控制的优点在于：控制律局部近似模型可以在轨迹优化的同时得到；在线查询的计算量小，适合实时控制；不需要事先确定表示控制律函数的基函数，控制律函数的表示更加灵活；构造状态反馈控制律所需的轨迹数目较少等。

2. 研究了基于优化的仿人机器人站立平衡控制。首先研究了实际机器人控制中遇到的模型物理参数辨识和状态在线估计问题。实验中利用非线性最小二乘方法能够有效地辨识出两连杆倒立摆模型的物理参数，基于扩展卡尔曼滤波的状态估计方法在仿真和实验中能有效地在线估计出系统的状态。针对目前参数化控制器存在性能局限和参数调节困难、采用最优控制方法实现的控制器计算时间过长且只实现了瞬间扰动下的站立平衡等问题，提出了基于轨迹库的站立平衡控制方法。在生成轨迹库时，针对如何减少轨迹库中轨迹数目这一关键性问题，提出了初始条件自适应网格上基于控制器预测性能的轨迹库生成方法。该方法不仅保证了生成的轨迹之间的一致性，从而可以利用来自不同轨迹的控制律局部近似模型实现状态反馈控制，而且能够在满足控制器性能要求的同时，减少轨迹库中轨迹数量。利用本文提出的方法，研究了平面两连杆和平面四连杆倒立摆模型的站立平衡控制，实现了瞬间和持续且时变的扰动下的站立平衡控制，仿真结果验证了方法的有效性。我们还设计了基于线性二次型调节器（LQR）的增益调度控制器，对比结果说明所提出方法更加鲁棒且性能更加优越。我们进一步在 Sarcos 仿人机器人上实现了所提出的站立平衡控制方法，实验结果进一步验证了基于轨迹库的站立平衡控制方法的有效性。为进一步提高控制器的鲁棒性，研究了期望状态的选择对控制器鲁棒性的影响，并提出了确定期望状态的方法。研究表明，持续外力作用下站立平衡控制的最佳站立姿态不仅决定于外力的大小，还决定于外力可能持续的时间。仿真和实验结果说明以最佳的站立姿态作为站立平衡控制中的期望状态能够进一步提高控制的鲁棒性。

3. 研究了基于优化的仿人机器人行走控制问题。首先，研究了更具一般意义的动力学不连续系统的周期性任务的最优控制问题。针对目前采用动态规划方法实现中存在的“维数”问题，提出了一种邻域最优控制方法。采用参数化轨迹优化方法求解周期性稳态轨迹，在推导动力学不连续处值函数的传递关系的基础上，采用微分动态规划得到更加精确的周期性稳态轨迹和其采样状态上的控制律局部近似模型。在此基础上，研究了只考虑单腿支撑的平面七连杆双足行走控制问题。针对现有方法存在控制效率低、鲁棒性差、应用模型过于简单等问题，提出了基于轨迹库的行走控制方法。该方法首先将行走控制转化为动力学不连续系统的周期性任务无限时域的最优控制问题，然后利用参数化轨迹优化方法和 DDP 生成了控制中的稳态轨迹和轨迹采样状态上的控制律局部近似模型，进而实现稳态轨迹邻域内的反馈控制。最后生成大量的轨迹片段实现更大状态空间上的反馈控制。通过瞬间扰动下的、斜面上的、持续扰动下的、上

身质量增加 30% 的和不同的脚-地面接触模型的行走控制仿真验证方法的有效性和鲁棒性。本文提出的方法能够实现状态空间为 10 维的非线性系统的最优控制，而同样的问题采用动态规划方法目前几乎是不可能的。另外，将提出的方法与基于轨迹跟踪的行走控制进行了对比，对比结果显示了基于轨迹库的行走控制方法在鲁棒性、性能方面的优越性。最后，研究了更具一般性的行走控制（考虑双腿支撑阶段的三维七连杆双足行走控制），给出了稳态轨迹优化和微分动态规划的初步结果。对于状态空间是 20 维的非线性系统的控制问题，参数化轨迹优化方法仍能给出满意的结果。

6.2 展望

1. 参数化轨迹优化和微分动态规划方法都属于局部优化方法，本文通过由易到难逐渐生成整个轨迹库、保持相邻轨迹之间的一致性、在值函数不连续处增加局部近似模型和对生成的轨迹库周期地重新优化等方法，这样能够使得得到的轨迹不断地接近全局最优，但仍不能保证结果是全局最优。因为本文的主要目的是如何尽快地生成较好的控制器，所以全局最优并不是最重要的。但是某些情况下全局最优是控制器设计追求的目标，因此探索能够实现全局最优控制同时避免出现“维数灾”问题的方法是未来研究的方向之一。
2. 考虑到侧向平面双脚的支撑区域大于前向平面，侧向平面内的站立平衡比前向平面更容易，因此本文主要研究了前向平面内的站立平衡控制。对于三维站立平衡，如果支撑脚始终不离开地而且外界扰动不引起摇首转矩或角动量，那么可以将机器人的动力学分解为前向平面和侧向平面，然后采用本文提出的方法分别设计两个平面内的站立平衡控制器。对于一般的三维站立平衡问题，则需要考虑外界扰动引起的摇首转矩或角动量以及前向平面和侧向平面动力学的耦合等问题，这增加了状态空间的维数和轨迹优化的难度。采用基于轨迹库的控制方法实现三维站立平衡并进行实验将作为下一步的工作之一。
3. 本文给出了三维行走控制中周期性稳态轨迹优化的初步结果。对于更大状态空间上的三维行走控制，则需要采用微分动态规划在周期性稳态轨迹附近生成更多的轨迹片段构成轨迹库来实现。三维行走控制仿真和实验是下一步的工作。
4. 目前行走控制轨迹优化的代价函数的形式比较复杂，参数的选择是根据反复尝试得到。仿真中发现代价函数参数的选择对最终生成的行走控制器的鲁棒性有很大影响，因此如何确定代价函数的形式和参数是值得研究的问题。这个问题属于“逆优化”问题，解决这个问题的一种思路是通过捕捉到的大量的

人类关节运动轨迹通过“逆优化”得到最合适代价函数。如何确定代价函数的形式和参数是未来研究的方向之一。

5. 仿人机器人实际控制中，模型误差和传感器测量噪声是不可避免的。基于轨迹的控制方法“过于乐观”，即其完全信任动力学模型生成最优控制，所以得到的控制律在实际应用时可能会因为增益过大而导致系统不稳定。因此研究考虑模型误差、测量噪声等不利因素的影响，生成更加鲁棒的控制器的方法具有更加实际的意义。

6. 本文采用基于轨迹库的方法实现了仿人机器人两类典型任务：扰动下的站立平衡控制和双足行走控制。但这两类任务并不是孤立的：扰动下的站立平衡控制可以看作是期望速度为零的行走控制。如果将扰动下的站立平衡控制看作是期望速度为零的行走控制，那么可以很自然地产生站立平衡中的跨步策略和其他各种策略。研究采用轨迹库实现更加一般、多任务的仿人机器人运动控制，将是下一步的工作之一。

7. 作为一种仿人机器人的运动控制方法，本文提出的方法同样适于计算机动画领域，可以用来生成逼真的动画形象。

附录 A 碰撞模型

对于平面七连杆机器人，如果在行走过程中始终保持双脚与地面平行，那么系统具有 7-DOF。将其广义坐标定义为 $\mathbf{q}_e \triangleq (\theta_1, \dots, q_5, p_1^h, p_1^v)^\top$ ，其中 p_1^h 和 p_1^v 分别代表支撑踝关节的水平和垂直位置，如图 A.1 所示。机器人的动力学因此可以表示为

$$\mathbf{M}_e(\mathbf{q}_e)\ddot{\mathbf{q}}_e + \mathbf{h}_e(\mathbf{q}_e, \dot{\mathbf{q}}_e) = \mathbf{B}_e \mathbf{u}_e + \mathbf{J}^\top \delta \mathbf{F} \quad (\text{A.1})$$

其中 $\mathbf{u}_e \triangleq (u_1, \dots, u_5, f_1^h, f_1^v)^\top$, $\delta \mathbf{F} \triangleq (f_2^h, f_2^v)^\top$, f_1^h , f_1^v , f_2^h 和 f_2^v 分别为碰撞时的地面反作用力（如图 A.1 所示）。 \mathbf{J} 代表摆动脚踝关节和广义坐标之间的雅可比矩阵

$$\mathbf{J} \triangleq \frac{\partial \mathbf{p}_2}{\partial \mathbf{q}_e} \quad (\text{A.2})$$

其中 $\mathbf{p}_2 \triangleq (p_2^h, p_2^v)^\top$ 。将摆动腿和地面之间的碰撞建模为刚体之间的瞬间碰撞，并且假设向心力、科氏力、重力和关节转矩相对碰撞引起的冲击力可以忽略。碰撞的冲击力将引起关节角速度的瞬间变化，而关节角基本保持不变。对公式 (A.1) 两边在碰撞发生时积分，得到

$$\mathbf{M}_e(\mathbf{q}_e)(\dot{\mathbf{q}}_e^+ - \dot{\mathbf{q}}_e^-) = \mathbf{J}^\top \mathbf{F}, \quad (\text{A.3})$$

其中 $\dot{\mathbf{q}}_e^+$ 是碰撞之后的速度， $\dot{\mathbf{q}}_e^-$ 是碰撞之前的速度， $\mathbf{F} \in \mathbb{R}^2$ 是地面反作用力

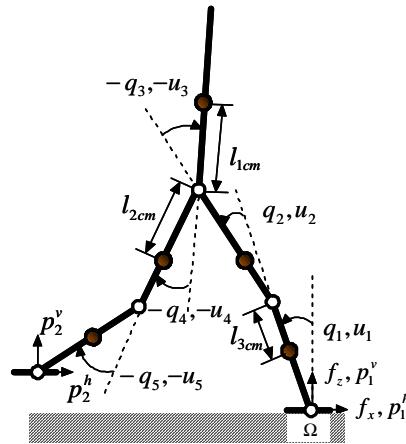


图 A.1 双足机器人仿真简化模型

Fig. A.1 Simplified structure of the simulated biped robot.

引起的冲量。假设摆动脚在碰撞后相对地面没有发生反弹和滑动，得到

$$\dot{\mathbf{p}}_2 = \mathbf{J}\mathbf{q}_e^+ = 0 \quad (\text{A.4})$$

由公式 (A.3) 和 (A.4) 构成方程组

$$\begin{bmatrix} \mathbf{M}_e(\mathbf{q}_e) & -\mathbf{J}^\top \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}_e^+ \\ \mathbf{F} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_e(\mathbf{q}_e)\mathbf{q}_e^- \\ 0 \end{bmatrix} \quad (\text{A.5})$$

这个线性代数方程组的解分别是碰撞以后的关节角速度 $\dot{\mathbf{q}}^+$ 和碰撞引起的冲量 \mathbf{F} 。最终得到的碰撞模型是

$$\mathbf{x}^+ = \begin{bmatrix} \mathbf{E} & 0 \\ 0 & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{q}^- \\ \dot{\mathbf{q}}^+ \end{bmatrix} = \Delta(\mathbf{x}^-), \quad (\text{A.6})$$

其中 \mathbf{x}^- 和 \mathbf{x}^+ 分别是碰撞之前和碰撞之后的状态， \mathbf{E} 为常数矩阵，描述了因支撑脚变化引起的坐标变换关系。

参考文献

- [1] Vukobratović M., Borovac B., Surla D., *et al.*, *Biped Locomotion: Dynamics, Stability, Control and Application*, Springer, 1990.
- [2] Gubina F., Hemami H., and McGhee R.B., “On the dynamic stability of biped locomotion”, *IEEE Transactions on Biomedical Engineering*, 1974, **21**(2), 102–108.
- [3] Raibert M.H., “Legged robots”, *Communications of the ACM*, 1986, **29**(6), 499–514.
- [4] Hirai K., Hirose M., Haikawa Y., *et al.*, “The development of Honda humanoid robot”, *Proceedings of IEEE International Conference on Robotics and Automation*, 1998, 1321–1326.
- [5] Kaneko K., Kanehiro F., Kajita S., *et al.*, “Design of prototype humanoid robotics platform for HRP”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2002, 2431–2436.
- [6] Bentivegna D.C., Atkeson C.G., and Kim J.Y., “Compliant control of a hydraulic humanoid joint”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2007, 483–489.
- [7] Atkeson C.G., Hale J.G., Pollick F., *et al.*, “Using humanoid robots to study human behavior”, *IEEE Intelligent Systems*, 2000, **15**(4), 46–56.
- [8] Pratt G.A., “Legged robots at MIT: what’s new since Raibert?”, *IEEE Robotics & Automation Magazine*, 2000, **7**(3), 15–19.
- [9] Au S.K., Berniker M., and Herr H., “Powered ankle-foot prosthesis to assist level-ground and stair-descent gaits”, *Neural Networks*, 2008, **21**, 654–666.
- [10] Popović M. and Englehart A., “Angular momentum primitives for human walking: biomechanics and control”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2004, 1685–1691.

- [11] Wu Q. and Sabet N., “An experimental study of passive dynamic walking”, *Robotica*, 2004, **22**(3), 251–262.
- [12] Yin K., Coros S., Beaudoin P., *et al.*, “Continuation methods for adapting simulated skills”, *Proceedings of ACM SIGGRAPH*, 2008, 1–7.
- [13] Yin K., Loken K., and van de Panne M., “SIMBICON: simple biped locomotion control”, *ACM Transactions on Graphics*, 2007, **26**(3), 105–110.
- [14] Raibert M.H. and Hodgins J.K., “Animation of dynamic legged locomotion”, *ACM SIGGRAPH Computer Graphics*, 1991, **25**(4), 349–358.
- [15] Lee J., Chai J., Reitsma P.S.A., *et al.*, “Interactive control of avatars animated with human motion data”, *ACM Transactions on Graphics*, 2002, **21**(3), 491–500.
- [16] Safanova A., Hodgins J.K., and Pollard N.S., “Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces”, *ACM Transactions on Graphics*, 2004, **23**(3), 514–521.
- [17] Anderson S., Atkeson C.G., and Hodgins J.K., “Coordinating feet in bipedal balance”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2006, 624–628.
- [18] Safanova A. and Hodgins J.K., “Construction and optimal search of interpolated motion graphs”, *ACM Transactions on Graphics*, 2007, **26**(3), 106.
- [19] Kuo A.D., “Energetics of actively powered locomotion using the simplest walking model”, *Journal of Biomechanical Engineering*, 2002, **124**(1), 113–120.
- [20] Kuo A.D., “An optimal control model for analyzing human postural balance”, *IEEE Transactions on Biomedical Engineering*, 1995, **42**(1), 87–101.
- [21] Park S., Horak F., and Kuo A.D., “Postural feedback responses scale with biomechanical constraints in human standing”, *Experimental Brain Research*, 2004, **154**(4), 417–427.

- [22] Hemami H., Barin K., and Pai Y.C., “Quantitative analysis of the ankle strategy under translational platform disturbance”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2006, **14**(4), 470–480.
- [23] Goswami A., “Postural stability of biped robots and the foot-rotation indicator (FRI) point”, *International Journal of Robotics Research*, 1999, **18**(6), 523–533.
- [24] Abe Y., da Silva M., and Popović J., “Multiobjective control with friction contacts”, *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007, 249–258.
- [25] Macchietto A., Zordan V., and Shelton C.R., “Momentum control for balance”, *ACM Transactions on Graphics*, 2009, **28**(3), 1–8.
- [26] Kudoh S., Komura T., and Ikeuchi K., “The dynamic postural adjustment with the quadratic programming method”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2002, 2563–2568.
- [27] Nenchev D.N. and Nishi A., “Ankle and hip strategies for balance recovery of a biped subjected to an impact”, *Robotica*, 2008, **26**(5), 643–653.
- [28] Runge C.F., Shupert C.L., Horak F.B., *et al.*, “Ankle and hip postural strategies defined by joint torques”, *Gait and Posture*, 1999, **10**, 161–170.
- [29] Alexandrov A., Frolov A., Horak F., *et al.*, “Feedback equilibrium control during human standing”, *Biological Cybernetics*, 2005, **93**, 309–322.
- [30] Hemami H. and Camana P., “Nonlinear feedback in simple locomotion systems”, *IEEE Transactions on Automatic Control*, 1976, **21**(6), 855–860.
- [31] Golliday C.J. and Hemami H., “Postural stability of the two-degree-of-freedom biped by general linear feedback”, *IEEE Transactions on Automatic Control*, 1976, **21**(1), 74–79.
- [32] Xin X., Mita T., and Kaneda M., “The posture control of a two-link free flying acrobot with initial angular momentum”, *IEEE Transactions on Automatic Control*, 2004, **49**(7), 1201–1206.

- [33] Boone G., “Minimum-time control of the acrobot”, *Proceedings of IEEE International Conference on Robotics and Automation*, 1997, 3281–3287.
- [34] Napoleon, Shigeki N., and Sampei M., “Balance control analysis of humanoid robot based on ZMP feedback control”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2002, 2437–2442.
- [35] Hofmann A., *Robust Execution of Bipedal Walking Tasks From Biomechanical Principles*, Ph.D. thesis, Massachusetts Institute of Technology, 2006.
- [36] Kondak K. and Hommel G., “Control and online computation of stable movement for biped robots”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2003, 874–879.
- [37] Stephens B., “Integral control of humanoid balance”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2007, 4020–4027.
- [38] Kudoh S. and Komura T., “ C^2 continuous gait-pattern generation for biped robots”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2003, 1135–1140.
- [39] Komura T., Leung H., Kudoh S., et al., “A feedback controller for biped humanoids that can counteract large perturbations during gait”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, 1989–1995.
- [40] Stephens B., “Humanoid push recovery”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2007, 589–595.
- [41] Atkeson C.G. and Stephens B., “Multiple balance strategies from one optimization criterion”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2007, 57–64.
- [42] Atkeson C.G. and Stephens B., “Random sampling of states in dynamic programming”, *IEEE Transactions on Systems, Man & Cybernetics: Part B*, 2008, **38**(4), 924–929.

- [43] Xinjilefu X., Hayward V., and Michalska H., “Stabilization of the spatial double inverted pendulum using stochastic programming seen as a model of standing postural control”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2009, 367–372.
- [44] Xing D., Atkeson C.G., Su J., et al., “Gain scheduled control of perturbed standing balance”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2010, 4063–4068.
- [45] Alexander R.M., *Optima for Animals*, revised edition edition, Princeton University Press, 1996.
- [46] Collins S., Ruina A., Tedrake R., et al., “Efficient bipedal robots based on passive dynamic walkers”, *Science*, 2005, **307**(5712), 1082–1085.
- [47] Huang Q., Yokoi K., Kajita S., et al., “Planning walking patterns for a biped robot”, *IEEE Transactions on Robotics and Automation*, 2001, **17**(3), 280–289.
- [48] Kajita S., Kanehiro F., Kaneko K., et al., “The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2001, 239–246.
- [49] Kajita S., Matsumoto O., and Saigo M., “Real-time 3D walking pattern generation for a biped robot with telescopic legs”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2001, 2299–2306.
- [50] Kajita S. and Tani K., “Experimental study of biped dynamic walking”, *IEEE Control Systems Magazine*, 1996, **16**(1), 13–19.
- [51] Kajita S., Kanehiro F., Kaneko K., et al., “Biped walking pattern generation by using preview control of zero-moment point”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2003, 1620–1626.
- [52] Roussel L., Canudas C., and Goswami A., “Generation of energy optimal complete gait cycles for biped robots”, *Proceedings of IEEE International Conference on Robotics and Automation*, 1998, 2036–2041.

- [53] Hardt M., *Multibody Dynamical Algorithms, Numerical Optimal Control, with Detailed Studies in the Control of Jet Engine Compressors and Biped Walking*, Ph.D. thesis, University of California, 1999.
- [54] Djoudi D., Chevallereau C., and Aoustin Y., “Optimal reference motions for walking of a biped robot”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, 2002–2007.
- [55] Bessonnet G., Seguin P., and Sardain P., “A parametric optimization approach to walking pattern synthesis”, *International Journal of Robotics Research*, 2005, **24**(7), 523–536.
- [56] Saidouni T. and Bessonnet G., “Generating globally optimised sagittal gait cycles of a biped robot”, *Robotica*, 2003, **21**(2), 199–210.
- [57] Chevallereau C. and Aoustin Y., “Optimal reference trajectories for walking and running of a biped robot”, *Robotica*, 2001, **19**(5), 557–569.
- [58] Roussel L., Canudas C., and Goswami A., “Comparative study of methods for energy-optimal gait generation for biped robots”, *Proceedings of IEEE International Conference on Informatics and Control*, 1997, 1205–1212.
- [59] Gill P.E., Murray W., and Saunders M.A., “SNOPT: an SQP algorithm for large-scale constrained optimization”, *SIAM Journal on Optimization*, 1997, **12**, 979–1006.
- [60] Hasegawa Y., Arakawa T., and Fukuda T., “Trajectory generation for biped locomotion robot”, *Mechatronics*, 2000, **10**, 67–89.
- [61] Huang Q. and Nakamura Y., “Sensory reflex control for humanoid walking”, *IEEE Transactions on Robotics*, 2005, **21**(5), 977–984.
- [62] Park J. and Kim K., “Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control”, *Proceedings of IEEE International Conference on Robotics and Automation*, 1998, 3528–3533.
- [63] Nikkhah M., Ashrafiou H., and Fahimi F., “Robust control of underactuated bipeds using sliding modes”, *Robotica*, 2007, **25**(3), 367–374.

- [64] Tedrake R., Manchester I., Tobenkin M., *et al.*, “LQR-trees: feedback motion planning via sums-of-squares verification”, *International Journal of Robotics Research*, 2010, **29**(8), 1038–1052.
- [65] Morimoto J., Zeglin G., and Atkeson C.G., “Minmax differential dynamic programming: application to a biped walking robot”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2003, 1927–1932.
- [66] Westervelt E.R., Grizzle J.W., and Koditschek D.E., “Hybrid zero dynamics of planar biped walkers”, *IEEE Transactions on Automatic Control*, 2003, **48**(1), 42–56.
- [67] Pratt J., Chew C.M., Torres A., *et al.*, “Virtual model control: an intuitive approach for bipedal locomotion”, *International Journal of Robotics Research*, 2001, **20**(2), 129–143.
- [68] Whitman E. and Atkeson C.G., “Control of a walking biped using a combination of simple policies”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2009, 520–527.
- [69] Mandersloot T., Wisse M., and Atkeson C.G., “Controlling velocity in bipedal walking: a dynamic programming approach”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2006, 124–130.
- [70] Stilman M., Atkeson C.G., Kuffner J., *et al.*, “Dynamic programming in reduced dimensional spaces: dynamic planning for robust biped locomotion”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, 2399–2404.
- [71] Tedrake, R. L., *Applied Optimal Control for Dynamically Stable Legged Locomotion*, Ph.D. thesis, Massachusetts Institute of Technology, 2004.
- [72] Matsubara T., Morimoto J., Nakanishi J., *et al.*, “Learning CPG-based biped locomotion with a policy gradient method”, *Robotics & Autonomous Systems*, 2006, **54**(11), 911.

- [73] Dimitrov D., Wieber P.B., Ferreau H.J., *et al.*, “On the implementation of model predictive control for on-line walking pattern generation”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2008, 2685–2690.
- [74] Wieber P.B., “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations”, *Proceedings of IEEE International Conference on Humanoid Robots*, 2006, 137–142.
- [75] McGeer T., “Passive dynamic walking”, *International Journal of Robotics Research*, 1990, **9**(2), 62.
- [76] Chevallereau B.C., Abba G., Aoustin Y., *et al.*, “RABBIT: a testbed for advanced control theory”, *IEEE Control Systems Magazine*, 2003, **23**(5), 57–79.
- [77] Kavraki L., Svestka P., Latombe J.C., *et al.*, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”, *IEEE Transactions on Robotics and Automation*, 1996, **12**(4), 566–580.
- [78] Stolle M. and Atkeson C.G., “Policies based on trajectory libraries”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2006, 3344–3349.
- [79] Jacobson D.H. and Mayne D.Q., *Differential Dynamic Programming*, volume 24 *Modern Analytic and Computational Methods in Science and Mathematics*, Elsevier, 1970.
- [80] Atkeson C.G., “Using local trajectory optimizers to speed up global optimization in dynamic programming”, *Advances in Neural Information Processing Systems*, 1994, 663–670.
- [81] Wieber P.B. and Chevallereau C., “Online adaptation of reference trajectories for the control of walking systems”, *Robotics and Autonomous Systems*, 2006, **54**(7), 559–566.

- [82] Tassa Y., Erez T., and Smart W., “Receding horizon differential dynamic programming”, *Advances in Neural Information Processing Systems*, 2008, 1465–1472.
- [83] Schierman J., Ward D., Hull J., *et al.*, “Integrated adaptive guidance and control for re-entry vehicles with flight test results”, *Journal of Guidance, Control, and Dynamics*, 2004, **27**(6), 975–988.
- [84] Boyan J.A. and Moore A.W., “Generalization in reinforcement learning: safely approximating the value function”, *Advances in Neural Information Processing Systems*, 1995, 369–376.
- [85] Tsitsiklis J.N. and Roy V.B., “Regression methods for pricing complex American-style options”, *IEEE Transactions on Neural Networks*, 2001, **12**, 694–703.
- [86] Murray J.J., Cox C., Lendaris G., *et al.*, “Adaptive dynamic programming”, *IEEE Transactions on Systems, Man & Cybernetics: Part C*, 2002, **32**(2), 140–153.
- [87] Bryson A.E. and Ho Y.C., *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere Pub. Co., 1975.
- [88] Betts J.T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, 2001.
- [89] Hargraves C.R. and Paris S.W., “Direct trajectory optimization using nonlinear programming and collocation”, *Journal of Guidance, Control, and Dynamics*, 1987, **10**(4), 338–342.
- [90] Elnagar G., Kazemi M., and Razzaghi M., “The pseudospectral Legendre method for discretizing optimal control problems”, *IEEE Transactions on Automatic Control*, 1995, **40**(10), 1793 –1796.
- [91] Vlassenbroeck J. and Van Dooren R., “A Chebyshev technique for solving nonlinear optimal control problems”, *IEEE Transactions on Automatic Control*, 1988, **33**(4), 333–340.

- [92] Bellman R., *Dynamic Programming*, Dover Publications, 2003.
- [93] Larson R.E., *State Increment Dynamic Programming*, American Elsevier Pub. Co., 1968.
- [94] Atkeson C.G. and Liu C., “Trajectory-based dynamic programming”, submitted to Springer Lecture Notes in Electrical Engineering.
- [95] Bryson A.E., *Dynamic Optimization*, 1st edition, Addison Wesley Longman, 1999.
- [96] Dyer P. and McReynolds S.R., *The Computation and Theory of Optimal Control*, Academic Press, 1970.
- [97] Atkeson C.G. and Morimoto J., “Nonparametric representation of policies and value functions: a trajectory-based approach”, *Advances in Neural Information Processing Systems*, 2003, 1643–1650.
- [98] Vanderbei R.J., “Case studies in trajectory optimization: trains, planes, and other pastimes”, *Optimization and Engineering*, 2001, **2**(2), 215–243.
- [99] Indyk P., *Nearest Neighbors in High-dimensional Spaces*, 2nd edition, CRC Press, 2004.
- [100] Bertsekas D.P., *Dynamic Programming and Optimal Control*, Athena Scientific, 1995.
- [101] Azevedo C., Poignet P., and Espiau B., “Moving horizon control for biped robots without reference trajectory”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2002, 2762–2767.
- [102] Takeuchi H., “Real time optimization for robot control using receding horizon control with equal constraint”, *Journal of Robotic Systems*, 2003, **20**(1), 3–13.
- [103] Liu C. and Atkeson C.G., “Standing balance control using a trajectory library”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2009, 3031–3036.

- [104] Moore M., Wilhelms J., Graphics C., *et al.*, “Collision detection and response for computer animation”, *Computer Graphics*, 1988, **22**(4), 289–298.
- [105] Chevallereau C., Grizzle J.W., and Shih C.L., “Asymptotically stable walking of a five-link underactuated 3-D bipedal robot”, *IEEE Transactions on Robotics*, 2009, **25**(1), 37–50.
- [106] Anderson F.C. and Pandy M.G., “Dynamic optimization of human walking”, *Journal of Biomechanical Engineering*, 2001, **123**(5), 381.
- [107] Betts J.T., “Survey of numerical methods for trajectory optimization”, *Journal of Guidance, Control, and Dynamics*, 1998, **21**(2), 193–207.
- [108] Kajita S., Kanehiro F., Kaneko K., *et al.*, “Resolved momentum control: humanoid motion planning based on the linear and angular momentum”, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2003, 1644–1650.
- [109] Chevallereau C., Formal’sky A., and Djoudi D., “Tracking a joint path for the walk of an underactuated biped”, *Robotica*, 2004, **22**(1), 15–28.
- [110] Yin K., Pai D.K., and van De Panne M., “Data-driven interactive balancing behaviors”, *Proceedings of Pacific Graphics*, 2005.
- [111] Grossman R.L., Valsamis D., and Qin X., “Persistent stores and hybrid systems”, *Proceedings of IEEE International Conference on Decision and Control*, 1993, 2298–2302.
- [112] Frazzoli E., Dahleh M., and Feron E., “Maneuver-based motion planning for nonlinear systems with symmetries”, *IEEE Transactions on Robotics*, 2005, **21**(6), 1077–1091.
- [113] Ramamoorthy S. and Kuipers B., “Qualitative hybrid control of dynamic bipedal walking”, *Robotics: Science and Systems*, 2006, 89–96.
- [114] Stolle M., Tappeiner H., Chestnutt J., *et al.*, “Transfer of policies based on trajectory libraries”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2007, 2981–2986.

- [115] Sutton R.S. and Barto A.G., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
- [116] Todorov E. and Tassa Y., “Iterative local dynamic programming”, *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2009, 90–95.
- [117] Altamimi A., Abu-Khalaf M., and Lewis F.L., “Adaptive critic designs for discrete-time zero-sum games with application to H-infinity control”, *IEEE Transactions on Systems, Man & Cybernetics: Part B*, 2007, **37**(1), 240–247.
- [118] Altamimi A., Lewis F.L., and Abu-Khalaf M., “Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control”, *Automatica*, 2007, **43**, 473-481.

致 谢

执笔至此，博士生涯将告一段落。诸多感慨，此刻却化为平静。

人生之大幸使我能够受教于两位杰出的老师—苏剑波教授和Christopher G. Atkeson教授。苏剑波教授同时也是我的硕士研究生导师，在近10年的相处中，老师渊博的学识、勤奋的工作态度、活跃的学术思想和严谨的工作作风永远是我学习的榜样。求学路上，老师的鼓励，给了我极大的鼓舞；老师的鞭策，更让我受益匪浅。从老师身上我得到的不仅是知识，更重要的是做事的方法、做人的原则、做研究的品位和精益求精的态度，这些将使我受用终身。在卡耐基-梅隆大学求学期间，得到了Christopher G. Atkeson教授的悉心指导。他为我开启了通往仿人机器人运动控制研究之门，教会了我看待问题独特的视角。Atkeson教授的敬业、勤奋与对科学事业的追求给我留下了深刻的印象。

论文完成之际，感谢仿人组邢登鹏博士、陈健、刘飞、张怡、杨斌、于佳和刘凯硕士的鼎力支持和帮助；同时感谢实验室其他博士：雷旭升、高振东、周玮、谢文龙、彭文东、杨飞、王永雄、钟建冬等在平时生活和研究过程中给予的诸多帮助；感谢刘艳涛、赵玥和骆超在本论文修改中提供的帮助；感谢实验室其他所有成员，在大家庭般融洽的气氛中，我度过了快乐的博士生涯。

在卡耐基-梅隆大学求学期间，田渊栋博士和曹劲硕士在生活和学习上给予了极大帮助；Sarcos实验室的Benjamin Stephens博士在实验过程中给予了极大帮助；Bumjoo Lee博士共享了部分代码；Eric C. Whitman博士和Sid Sanan博士等人给予了诸多启发；在Human Sensing Seminar上的交流开阔了我的眼界并让我感受到研究的无穷乐趣，在此表示衷心的感谢！

最后感谢上海交通大学A0603291班的全体同学，有你们的陪伴我度过了难忘的博士生涯。

谨以此文献给我挚爱的妻子、女儿和父母及亲人们！

攻读学位期间发表的学术论文目录

- [1] Chenggang Liu and Jianbo Su, “Basic behavior acquisition with multisensor integration for a robot system”, *Advanced Robotics*, 2009, **23**(12-13), 1561—1578. (SCI source)
- [2] Chenggang Liu, Christopher G. Atkeson, and Jianbo Su, “Neighboring optimal control for periodic tasks for systems with discontinuous dynamics”, *SCIENCE CHINA Information Sciences*, 2011, **54**(3): 653-663. (SCI source)
- [3] Chenggang Liu and Jianbo Su, “Basic behavior acquisition based on multisensor integration of a robot head”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2008, 3094—3099. (EI source)
- [4] Chenggang Liu and Christopher G. Atkeson, “Standing balance control using a trajectory library”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2009, 3031—3036. (EI source)
- [5] Chenggang Liu and Christopher G. Atkeson, “Standing balance control using a trajectory library”, *the 4th annual meeting of Dynamic Walking*, 2009.
- [6] Chenggang Liu and Jianbo Su, “Biped walking control using offline and online optimization”, *the 30th Chinese Control Conference*, 2011. Accepted (EI Source)
- [7] 刘成刚, 苏剑波,“持续扰动下的仿人机器人站立平衡控制”, 中国控制会议, 2011. 录用 (EI源)
- [8] Chenggang Liu, Christopher G. Atkeson, and Jianbo Su, “Biped walking control using a trajectory library”, *Robotica*, 2010. Revised (SCI source)
- [9] Chenggang Liu and Christopher G. Atkeson, “Control using a trajectory library: application to standing balance control”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*. 2011. Submitted (SCI source)

- [10] Chenggang Liu and Jianbo Su, “Implementation of a trajectory library approach to controlling humanoid standing balance”, *IEEE International Conference on Systems, Man, and Cybernetics*, 2011. Submitted (EI source)
- [11] Christopher G. Atkeson and Chenggang Liu, “Trajectory-based dynamic programming”, Springer Lecture Notes in Electrical Engineering, 2010. Submitted (SCI source)
- [12] 刘成刚, 苏剑波, “基于轨迹库的仿人机器人站立平衡控制方法”, 自动化学报, 2011. 已投 (EI源)

专利

1. 人工关节电缆驱动器, 苏剑波, 刘成刚, 中国专利授权 ZL 200710038296.6
2. 分布式仿人机器人通用控制系统, 苏剑波, 杨斌, 刘成刚, 中国专利授权 ZL 200910054322
3. 仿人机器人分布式双总线运动控制系统, 刘成刚, 苏剑波, 张怡, 中国专利公开 CN101293350