

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS



论文题目 学习算法构建控障函数：实现双足机器人安全控制的研究

学生姓名 杨 晨 宇

学生学号 516030910537

指导教师 高岳

专 业 IEEE 试点班

学院（系）电子信息与电气工程学院

Submitted in total fulfilment of the requirements for the degree of
Bachelor
in IEEE

Synthesis control barrier functions with learning algorithms: with the applications of bipedal safe walking

CHENYU YANG

Supervisor
Prof. YUE GAO

DEPART OF COMPUTER SCIENNCE, SCHOOL OF ELECTRONIC INFORMATION AND
ELECTRONICAL ENGINEERING
SHANGHAI JIAO TONG UNIVERSITY
SHANGHAI, P.R.CHINA

June. 1st, 2020

上海交通大学

毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（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。

作者签名：_____

日 期：_____年 ____月 ____日

上海交通大学

毕业设计（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“√”)

作者签名：_____

指导教师签名：_____

日 期：_____年 ____月 ____日

日 期：_____年 ____月 ____日

学习算法构建控障函数：实现双足机器人安全控制的研究

摘 要

近期,基于控制障碍函数 Control Barrier function(CBF)的控制方法在很多要求鲁棒性和安全性的任务上都取得了较大的成功。使用带有 CBF 的控制器,被控制的系统可以被证明对于一个安全集合具有前向不变性 (forward invariance)。但是,在足式机器人这样的应用场景中使用 CBF 还有诸多问题,CBF 的应用潜力也没有完全发挥。第一,在双足机器人这类复杂的系统中,对于很多任务,目前的理论方法无法得到可以满足前向不变性的 CBF。因此,科研人员需要手工调试 CBF,并且难以保障前向不变性。第二,目前 CBF 仅仅应用在了比较直观的任务中,但 CBF 的应用不应仅仅局限于此,它仍有被用来确保状态空间中更加抽象的集合的前向不变性的潜力。导致以上两点不足的原因之一,就是目前科研人员仅仅停滞于手工设计 CBF 的形式,而缺少一种自动化寻找 CBF 的方法。本文提出一种基于大量实验的自动化学习 CBF 的算法,并将算法应用到足式机器人步行任务中。使用者将希望系统避免的状态的集合,以及一个可以进行仿真的环境和基本的控制器传给 CBF 学习算法,CBF 学习算法在仿真环境中采集安全集合与不安全集合的样本,并使用经典的分类算法进行拟合,得到 CBF 的函数表达。本文在 pybullet 物理引擎中做了基于双足机器人步行的实验,学习出了可以确保机器人关节极限的 CBF。我们使用 CBF 学习算法应用在步行任务上,得到的 CBF 极大地增加了系统对参考轨迹误差和控制器参数的鲁棒性,甚

至可以与原本不稳定的参考轨迹和控制器参数共同作用，使机器人可以稳定行走。

关键词： 机器学习,SVM, 双足机器人, 状态障碍函数,CBF

Synthesis control barrier functions with learning algorithms: with the applications of bipedal safe walking

ABSTRACT

Recently, Control Barrier Function(CBF) based control techniques have achieved success in several safety-critical applications. Using CBFs as constraints, a controller can be proved to ensure the forward invariance of a "safe set" in the state space. However, problems still exist when using CBFs in a scenario such as legged robot, and the application potential of CBF has not been fully exploited. First, in a highly complexed system such as biped robot, for many tasks, there is no theoretical methods that can derive the CBF with ensurance of forward invariance. Therefore, researchers need to debug CBF manually, with difficulties finding the guarantee of forward invariance. Second, CBFs are only used in more intuitive tasks, such as obstacle avoidance and trajectory guidance. However, CBFs have more potentials than this, It can also be used to ensure forward immutability of more abstract collections in state space. The reason why the second goal is hard to achieve is that at present, researchers can only design CBFs by hand, and lack of an automatic way to find CBF. In this paper, an automatic learning CBF algorithm based on a large number of experiments is proposed and applied to bipedal walking. Users pass the set of states they want the system to avoid, a simulation environment and basic controllers to the proposed CBF learning algorithm.

The learning algorithm collects samples of safe and unsafe sets in simulation environment, and uses classical machine classification learning algorithm to fit the expression of the CBFs. We did experiments based on biped robot walking in pybullet physical engine, and learned CBFs which can ensure the joint limits. And we show that a robust CBF of the robot's stable walking gait against the errors of reference trajectories and controller parameters can be obtained from our learning algorithm.

KEY WORDS: Machine Learning, SVM, Bipedal, Control Barrier Function

目录

第一章 绪论	1
1.1 研究的背景与意义	1
1.2 课题目标与内容	4
1.3 本文结构	5
第二章 双足机器人安全控制相关技术	6
2.1 CBF 机器人控制研究现状	6
2.1.1 CLF、CBF 以及控制器基础原理	6
2.1.2 CBF 在足式机器人控制中的应用	9
2.1.3 CBF 与机器学习的结合	11
2.2 机器人动力学与约束相关推导	12
2.2.1 机器人动力学模型概述	12
2.2.2 机器人系统碰撞模型与约束	13
2.2.3 有约束的动力学的模型	13
2.3 轨迹优化和 Hybrid Zero Dynamics	15
2.3.1 Virtual Constraint 与 Hybrid Zero Dynamics	15
2.3.2 轨迹优化	16
2.4 本章小结	16

第三章 CBF 学习算法的研究和设计	18
3.1 算法思想框架	18
3.1.1 仿真采样	18
3.1.2 CBF 拟合	21
3.1.3 算法总体流程	22
3.2 足式机器人背景下的 CBF 学习算法	22
3.2.1 足式机器人背景下的仿真采样	23
3.2.2 足式机器人环境中的 CBF 拟合	25
3.3 本章小结	26
第四章 实验验证	28
4.1 玩具问题中对算法框架的验证	28
4.1.1 玩具问题问题描述	28
4.1.2 玩具问题具体算法	29
4.1.3 玩具问题实验结果和分析	30
4.2 机器人环境实验设置	31
4.3 学习 CBF 控制机器人关节范围	34
4.3.1 实验设置	35
4.3.2 实验结果和分析	35
4.4 学习 CBF 确保机器人稳定步行	37
4.4.1 实验设置	38
4.4.2 实验结果和分析	39
4.5 本章小结	40

全文总结	43
符号说明	44
参考文献	46
致谢	51

第一章 绪论

1.1 研究的背景与意义

长久以来，足式机器人是工程和科学研究的重要领域。足式机器人具有广阔的应用前景，相比轮式机器人，足式机器人可以克服更多复杂障碍的限制，相比无人机，足式机器人具有更高的能源利用率和续航时间^[1]。所以足式机器人是兼顾了克服复杂地形以及能耗比的极好载具，可以在抢险救灾，物资配送，星际勘探^[2]等领域中发挥重要作用。同时，足式机器人还是各种科学原理与工程技术测试的良好目标，一些足式机器人是典型的欠驱动复杂混合系统^[3]，对其模型的研究与很多有关混沌理论^[4]，控制理论的研究相辅相成。足式机器人的模型复杂，但是其目标——行走或跑动，十分明确且直观，所以足式机器人还是机器学习算法的常见载体^[5, 6]，除了足式机器人的控制算法等软件设计外，更节能^[1]，更廉价^[7]，更坚固^[8]，更灵活^[9]的结构的设计也一直是科研工作者的研究目标。笔者曾听说过一句话并一直铭记在心，“机器人是科学王冠上的明珠。”对足式机器人的研究，意义非同小可。

在各种足式机器人中，双足机器人尤其受到科研人员的关注。尽管实现双足机器人的步行已经不再是梦想或神话，科研人员对于双足机器人步行的研究还远未停歇，人们始终在为设计行走更灵活，姿势更像人的双足机器人而努力。从最早的将机器人的动力学建模成线性倒立摆^[10]，并基于倒立摆的曲线进行控制^[11]，或是使用 Raibert control 的两条腿交替跳跃的双足机器人，到通过设计机器人质心投影的轨迹或者规划特征点 (ZMP, FRI^[12] 等等^[13]) 的方法^[14]，再到近些年来提出的 Hybrid Zero Dynamic 模型^[15] 以及相应的轨迹优化的方法^[1]，或是考虑身体整体需要的力矩从而反解各个关节需要的受力 (Whole Body Control)

[9, 16] 的方法，以及本文主要涉及的基于控制障碍函数 (CBF) 的方法^[17, 18]。各类方法层出不穷，双足机器人的能力也节节攀升，但还和双足机器人真正鲁棒地投入生产生活有一定距离。

近期，基于控制障碍函数 **Control Barrier function(CBF)** 的控制方法在很多要求鲁棒性的任务上都取得了较大的成功（比如足式机器人系统^{[17][19]}，巡航系统^[18]，多智能体系统^[20, 21]，以及无人机系统^[22]）。使用基于 **CBF** 和 **CLF** (**Control Lyapunov Function**) 的控制器框架，被控制的系统可以被证明具有在一个集合内的前向不变性 (**forward invariance**)，即如果系统的初始状态在一个特定集合内，那么系统任意时间的状态也必定属于这一个集合，这个集合也被称作“安全集合” (**safety set**)。所以，上述工作声称只要可以手工找到某一种满足条件的 **CBF**，那么被控系统就可以保证“安全”。对于一个满足函数 B ，如果其满足

$$B(x) \geq 0 \quad \forall x \in \mathcal{C}_0 \quad (1-1)$$

$$\sup_{u \in U} [L_f B(x) + L_g B(x)u + \gamma B(x)] \geq 0, \quad (1-2)$$

那么函数 B 的上水平集就具有前向不变性（详细说明请见2.1节）。比如，用 **CBF** 代表障碍物的位置，无人机和机器人可以实现避障；用 **CBF** 划开人所在的范围，人机协作时操作者的安全就可以得到保障；用 **CBF** 控制与前车的距离，跟车系统也可以得到安全保障。

但上述这些方法的局限在于，对于机器人这类复杂系统，一般的 **CBF** 是在特定的系统假设之下才能具有前向不变性，而真正不依赖假设具有前向不变性的集合很难通过理论分析来求出，更何谈去证明。比如 [19] 中的假设是不考虑机器人系统的电机能够产生的力矩限制，即如果假设系统的执行器可以输出任意大小的控制信号的话，那么系统可以被证明是具有前向不变性的。但是这样的假设往往与真实情况严重不符，因此系统在安全集合内的前向不变性会被破

坏。如 [17] 中明确指出他们并不能确保控制器能找到满足前向不变性的解，只是在实验验证中发现他们的 CBF 设计可行。所以目前 CBF 在双足机器人领域的使用仅仅算是一种新的控制器形式，而确保系统真正安全还需要大量的人工工程。

具体而言，目前 CBF 在实际的使用过程中，其局限体现在如下几个方面。

1. 对于如足式机器人等复杂欠驱动系统，很难实际地证明 CBF 的存在性，即对于一个 CBF，很难证明式(-17)中的“任意”条件。
2. CBF 需要巧妙地设计寻找满足要求的形式。对于一些不能简单地用避障或者位置引导来描述的问题，CBF 应该有的形式变得难以设计。比如想象目标是让机器人的倾角大于 80 度，但是机器人在倾角为 70 度的时候就已经不可能再直立起来了，所以要设计 CBF 的话必须知晓机器人还有能力回复的最大角度。这样对应机器人能力极限寻找 CBF 的复杂性会在机器人步行中更加明显，所以目前 CBF 的应用中，CBF 仅仅能做到一些基于直觉的设置。
3. CBF 仅仅是在控制器中作为辅助约束发挥作用的，而对于步行任务而言，相较控制器的其它成分发挥作用比较次要。在前文提及的足式机器人的研究中，所有机器人步行的核心都是经过轨迹优化得到的步态库，而 CBF 在其中的作用仅仅是如落脚点之类的微调。就算没有 CBF，这些机器人只用按照参考轨迹进行 PD 控制做到稳定行走，CBF 的潜力并没有得到完整地体现。

本文试图采用基于数据和机器学习的方法，不是通过手工在理想的假设中设计 CBF，而是通过大量实验采集数据，结合机器人的动力学模型，拟合出可以在实验环境中力保前向不变性的 CBF。本课题的任务分为两个方面：第一个是探索 CBF 学习算法，第二个是将这个算法应用在足式机器人上。

1.2 课题目标与内容

本课题旨在探索一种基于机器学习的新的解决方案。使用大量仿真环境下的实验代替理想假设下的证明，使用自动的探索和拟合算法代替人的手工设计，同时任务的设计会体现 CBF 的重要性，即通过加入 CBF，使原先无法平稳步行的控制器变得可以平稳步行。

具体的 CBF 学习问题定义为：对于一个机器人系统

$$\dot{x} = f(x) + g(x)u, \quad (1-3)$$

人为设定的避免集合和保护集合 $\mathcal{C}_{avoid} \subset \mathbb{R}^n$, $\mathcal{C}_{protect} \subset \mathbb{R}^n$, $\mathcal{C}_{avoid} \cap \mathcal{C}_{protect} = \emptyset$ 。避免集合就是不想让机器人状态进入的集合，比如与障碍物碰撞，或者跌倒到 45 度。保护集合就是学习得到的 CBF 控制器必须允许的状态的集合，比如机器人的一个稳定步行循环。本课题需要提供一套方法，可以得到在机器人实际运行中具有前向不变性的集合。 $\mathcal{C} \subset \mathbb{R}^n$, 满足

$$\mathcal{C} \cap \mathcal{C}_{avoid} = \emptyset, \mathcal{C}_{protect} \subset \mathcal{C} \quad (1-4)$$

同时，找到定义 \mathcal{C} 的一组 CBF 函数 \mathcal{B}

$$\mathcal{C} = \{x | B_i(x) > 0 \ \forall B_i \in \mathcal{B}\}, \quad (1-5)$$

满足

$$\sup_{u \in U} [L_f B_i(x) + L_g B_i(x)u + \gamma B_i(x)] \geq 0 \quad \forall B_i \in \mathcal{B} \quad (1-6)$$

其中 $\gamma > 0$ 。注意式(1-6)是式(-17)的简单拓展，将对单个 CBF 的条件拓展到对

一组 CBF 上。

举例而言，如果设定 $\mathcal{C}_{protect}$ 为机器人的某一个步行周期， \mathcal{C}_{avoid} 为机器人跌倒的状态，则算法的任务是找到一个前向不变的 **CBF**，对应集合 \mathcal{C} ，让机器人可以在 **CBF** 的控制之内步行¹，同时不会跌倒。

1.3 本文结构

本文第二章介绍课题涉及的相关技术，先着重介绍 **CBF** 的原理、应用、以及目前结合机器学习的相关工作。其次介绍课题中必要的模型以及相关动力学方程，为后文的探讨打下基础。最后简述本课题使用的其它技术，即轨迹优化和 Hybrid Zero Dynamic 思想在设计机器人控制器中的应用。

本文第三章详述课题提出的 **CBF** 学习算法，循序渐进分两步进行。首先介绍本文提出的学习算法的基本原理，相关概念的定义和算法整体流程。之后为了适应足式机器人的大背景，在基本原理上做出近似和改动，详细介绍机器人背景下的 **CBF** 学习算法。

本文第四章为实验部分，主要包含三块内容。第一是玩具问题中测试算法的基本思想；第二是在足式机器人上的简单任务——控制关节在关节限制之内，验证算法的基本可行性；第三是让一个无法形成稳定步行循环的机器人，在习得的 **CBF** 的作用下，可以平稳地步行，验证本文目标：通过学习 **CBF** 挖掘使用 **CBF** 的潜力。

本文将涉及比较多的概念和相对应的符号，为了方便读者阅读，本文最后的附录部分有符号表，整理了所有符号的概念和其被定义的位置。

¹例子中集合 \mathcal{C} 的集合包含着步行所必须的状态，但具体机器人会做什么动作还是会由整个控制器来决定

第二章 双足机器人安全控制相关技术

2.1 CBF 机器人控制研究现状

2.1.1 CLF、CBF 以及控制器基础原理

障碍函数 (barrier function) 在优化和控制领域得到了广泛的关注^[18]。障碍函数具有和李雅普诺夫函数相似的形式，但与之不同的是，障碍函数可以用来表示状态集合的前向不变性。具体而言，对于一个系统状态空间中的集合 \mathcal{C} ，存在障碍函数 B 满足：随着 $x \rightarrow \delta\mathcal{C}, B(x) \rightarrow +\infty$ 。这时如果需要满足系统关于集合 \mathcal{C} 前向不变性，只需一个类似李雅普诺夫的形式：

$$\dot{B} \leq \frac{\gamma}{B}, \quad \gamma > 0 \quad (2-1)$$

上式中 \dot{B} 是障碍函数 B 的时间导数，是系统控制输入 u 的函数。

控制李雅普诺夫函数 (CLF) ^[2] 可以用来寻找满足李雅普诺夫条件的控制输入，从而使系统稳定。其定义为：对于一个系统

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

存在一个正定、无界的函数 V 满足

$$x \neq 0 \implies \inf_{u \in U} \nabla V(x) \cdot f(x, u) < 0 \quad (2-2)$$

即系统总存在输入 u ，满足李雅普诺夫的条件 $\dot{V}(x, u) < 0$ 。使用 CLF 作为控制

器的约束，可以反向求解出满足李雅普诺夫条件的解。

李雅普诺夫函数可以满足系统“李雅普诺夫稳定”，一个更加严格的函数可以保证系统指数稳定^[18] (ES-CLF)。对于系统

$$\dot{x} = f(x) + g(x)u \quad (2-3)$$

如果存在一个连续的函数 $V : X \rightarrow R$:

$$c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2 \quad (2-4)$$

$$\inf_{u \in U} [L_f V(x) + L_g V(x)u + \lambda V(x)] \leq 0, \quad (2-5)$$

其中 $c_1, c_2, \lambda > 0$, $L_f V(x)$ 是 lie derivative,

$$L_f V(x) = \frac{\partial V}{\partial x} f(x), L_g V(x) = \frac{\partial V}{\partial x} g(x) \quad (2-6)$$

那么这个函数是 ES-CLF。注意 ES-CLF 除了收敛性更加严格之外，还假设系统每一时刻的动力学和系统的输入 u 为线性的关系。这样的假设契合大部分系统的实际（比如控制机器人的力矩），同时有助于简化控制器的实现。同样，CBF 也可以有类似指数稳定的形式^{[23][17]}，即：

$$B(x) \geq 0 \quad \forall x \in \mathcal{C}_0 \quad (2-7)$$

$$\sup_{u \in U} [L_f B(x) + L_g B(x)u + \gamma B(x)] \geq 0, \quad (2-8)$$

ES-CBF 和 ES-CLF 可以用来设计二次优化问题作为控制器，

$$\begin{aligned}
 & \underset{\mu, \delta}{\operatorname{argmin}} \quad \mu^T \mu + p\delta^2 \\
 & \text{s.t.} \quad \dot{V}(\eta, \mu) + \lambda V(\eta) \leq \delta \quad (\text{ES-CLF}), \\
 & \quad \quad \dot{B}(x, \mu) + \gamma B(x) \geq 0 \quad (\text{ES-CBF}), \\
 & \quad \quad A_C^\mu(x)\mu \leq b_C^\mu(x) \quad (\text{other constraints})
 \end{aligned} \tag{2-9}$$

其中 η 是系统稳定至 0 的误差, δ 是对于李雅普诺夫条件的松弛, $\lambda, \gamma > 0$ 分别为控制稳定速率的系数。由于 CBF 的定义(2-8), 属于安全集合 \mathcal{C} 中的任意状态 x 都存在满足约束条件的控制输入 μ , 从而在这样的控制器的作用下, 系统可以满足关于安全集合 \mathcal{C} 的前向不变性。除了特殊说明, 后文将用 CBF 代表(2-8)所定义的 ES-CBF, 用 CLF 表示(2-5)定义的 ES-CLF。

Relative degree 的概念经常被应用在 CBF 的设计或分析过程中, 其概念为系统的输入 u 和目标约束的关系, 例如在一个机器人系统中, 目标约束是和机器人位置有关的变量, 控制输入为力矩, 由于力矩和速度导数呈线性关系, 速度和位置导数呈线性关系, 所以 relative degree 为 2。Relative degree 大于 1 的约束条件是无法在 CBF-CLF-QP 中直接使用的, 例如对于系统

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

如果我们限定安全集合 $\mathcal{C} = \{x|x \geq 0\}$, 那么它的 relative degree 就是 2, 使用一阶的障碍函数是无法对系统输入 u 产生约束的。例如令 $B = x$, 我们就有 $L_g B(x) = 0$, 从而式(2-8)中 u 的系数为 0, 不是有效的约束。有很多方法可以得到处理 relative degree 较高的情况下的 CBF^{[19][23]}, 由于本文应用部分所有问题的 relative degree 不超过 2, 所以下面本文把 [23] 中的方法应用在 relative degree

为 2 的问题上。

对于如(2-3)所表示的系统，考虑 **relative degree** 为 2 的目标约束 $B(x)$ 。如果系统满足

$$\ddot{B} + \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} \cdot \begin{bmatrix} B \\ \dot{B} \end{bmatrix} \geq 0, \quad (2-10)$$

$$\text{eigen}\left(\begin{bmatrix} 0 & 1 \\ -k_0 & -k_1 \end{bmatrix}\right) < 0 \quad (2-11)$$

其中

$$\dot{B}(x) = L_f B(x), \ddot{B}(x) = L_{f^2} B + L_g L_f B u,$$

则函数 B 的上水平集具有前向不变性。

2.1.2 CBF 在足式机器人控制中的应用

基于 CBF 的控制技术已经应用在足式机器人的控制中，[19] 用 CBF-CLF-QP 来增加足式机器人的鲁棒性。此工作中添加了五个 CBF，分别是 1. 机器人中心速度；2. 机器人摆动腿最低高度；3. 机器人摆动腿最高高度；4. 机器人支撑腿关节限度 5. 摆动腿的角度。

结合步态库，CBF 还被用来控制机器人在梅花桩上的行走^[24]。在这篇工作中，机器人的摆动腿被限定在两个圆的中间。如2-1所示 其中 CBF 定义的安全集合为图 (b) 中的蓝色区域，图 (b) 中的红线为通过步态库差值得到的目标轨迹，基于步态库的差值无法保证机器人能精确地落脚，所以使用图中逐渐收窄的 CBF 在保证落脚点的精准。

CBF 的思想也可以被拓展到离散的控制中^[17]。如图2-2，CBF 被用来做顶层的机器人轨迹规划，即在机器人落脚时计算下一个落脚位置。从图中可以看

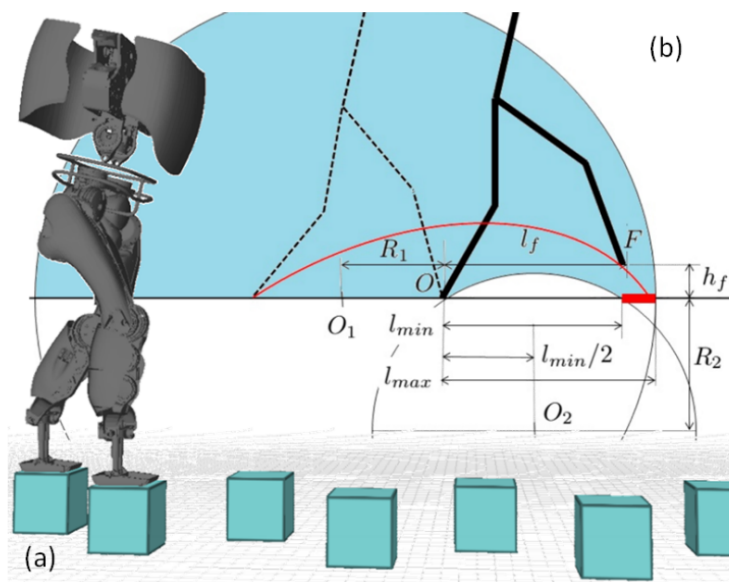


图 2-1 使用 CBF 控制机器人在梅花桩上行走示意图，其中 CBF 定义的安全集合为图 (b) 中的蓝色区域，图 (b) 中的红线为通过步态库差值得到的目标轨迹。（图片来自原文）

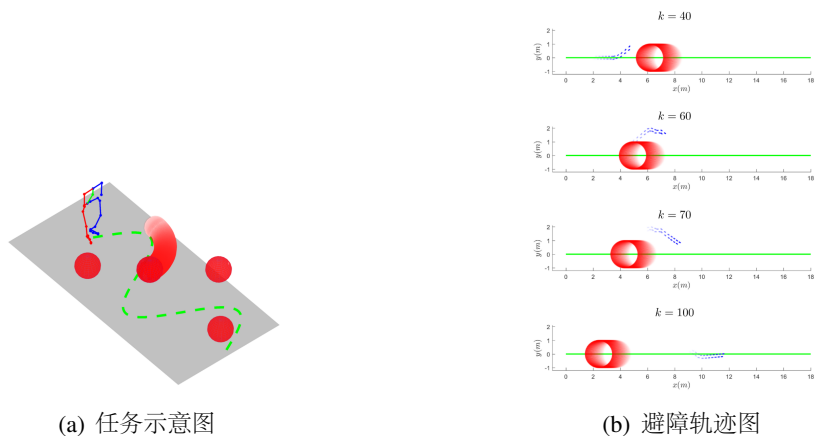


图 2-2 离散 CBF 在足式机器人中的应用^[17] 图中绿色为机器人需要跟踪的目标轨迹，红色为机器人需要躲避的障碍区域，红色的残影表现障碍物的移动 2.2(b)中的蓝色为机器人的落脚点位置，可以看出机器人避开了移动障碍并回到了目标轨迹上。（图片来自原文）

出机器人避开了移动的红色障碍并回到了目标轨迹上。

2.1.3 CBF 与机器学习的结合

CBF 在增强学习领域也得到了应用^[25, 26]。由于相较于传统的最优控制，增强学习领域常用马尔科夫概率模型，所以 CBF 约束的形式也进行了相应的改变^[25]。

$$\begin{aligned}
 (u_t, \epsilon) = \underset{u_t, \epsilon}{\operatorname{argmin}} \quad & \|u_t\|_2 + K_\epsilon \epsilon \\
 \text{s.t.} \quad & p^T f(s_t) + p^T g(s_t) u_t + p^T \mu_d(s_t) - \\
 & k_\delta |p|^T \sigma_d(s_t) + q \geq (1 - \eta) h(s_t) - \epsilon \\
 & a_{low}^i \leq u_t^i \leq a_{high}^i \text{ for } i = 1, \dots, M
 \end{aligned} \tag{2-12}$$

其中系统的模型是 $s_{t+1} = f(s_t) + g(s_t) u_t + d(s_t)$ 增加了最后一项表示不确定性，用一个高斯分布 $\mathcal{N}(\mu_d(s), \sigma_d^2(s))$ 来建模。文中假设 CBF 具有形式: $B = p^T s + q$ 。式(2-12)这个控制器的形式是在式2-9CBF 部分的基础上增加了置信概率的意义以及对 CBF 条件的松弛。文中使用人为设定的 CBF 来确保智能体的探索过程中的安全性，并将 CBF 应用在了倒立摆实验以及跟车实验中。[26] 使用了同样的 CBF 控制原理，但是独特的是，他使用时序逻辑来生成 CBF，确保智能体的行为符合时序逻辑的限制。这篇文章的作者将这一套控制器应用在了制作热狗以及上菜的机器人任务上。

本文中使用的模型，式2-3没有考虑系统建模的不确定性因素。一个精准的系统模型的意义在于，保证在施加 CBF 约束的过程中，控制器可以输出一个精准的前馈控制。但在系统模型仍有未被建模的成分时，神经网络可以用于拟合和预测未被建模的 CBF 的行为^[27]。假设已知有误差的系统模型是 \hat{B} ，用训练神

经网络的方式拟合两个补偿项 $\hat{\mathbf{a}}, \hat{\mathbf{b}}$, 系统的 CBF 控制器可以被表示为

$$\begin{aligned} \mathbf{k}'(\mathbf{x}) = & \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|_2^2 \\ \text{s.t. } & \hat{B}(\mathbf{x}, \mathbf{u}) + \hat{\mathbf{a}}(\mathbf{x})^\top \mathbf{u} + \hat{\mathbf{b}}(\mathbf{x}, \mathbf{u}) \geq -\alpha(B(\mathbf{x})) \end{aligned} \quad (2-13)$$

CBF 的思想也能直接应用于 neural control 中^[28], 文中作者控制机器手抓举物体, 使用 CBF 思想¹ 来确保抓举物体的精确性。限于篇幅, 本文不再赘述。

2.2 机器人动力学与约束相关推导

2.2.1 机器人动力学模型概述

常用的机器人动力学模型是拉格朗日动力学模型 ([29] 第 48.2 章), 模型中的机器人的状态 $q \in \mathbb{R}^n$ 指机器人 n 个自由度的值, 模型描述了当前状态的时间导数和状态的关系。本文推导和实验使用的形式为 (为了和仿真环境的接口保持一致):

$$\mathbf{A}_{rbd}(q)\ddot{q} + \mathbf{g}_{rbd}(q, \dot{q}) = \boldsymbol{\tau} + \mathbf{J}(q)^T \mathbf{F} \quad (2-14)$$

式中的下标 rbd 代表 rigid body dynamics, 为了和本文中其它的动力学模型进行区分。其中 \mathbf{A}_{rbd} 代表机器人的广义质量, \mathbf{g}_{rbd} 代表了机器人当前的加速度, 其中包括科里奥利力的作用以及重力加速度的作用, $\boldsymbol{\tau}$ 代表机器人各个关节的力矩, \mathbf{J}, \mathbf{F} 表示机器人受到的外力作用, \mathbf{F} 是外力的向量, \mathbf{J} 是外力作用点的雅克比矩阵, 定义为作用点的位置 (p_c) 的导数和广义状态的关系

$$\dot{p}_c = \mathbf{J}_c \dot{q} \quad (2-15)$$

¹文中声称是 barrier Lyapunov function (BLF), 但是思想大同小异

2.2.2 机器人系统碰撞模型与约束

本文模型假设每一步机器人的摆动腿与地面碰撞时是完全非弹性碰撞，即碰撞物体的相对速度瞬间变为 0，位置保持原位不动。本文中模型仅考虑机器人支撑脚所受地面的支持力和摩擦力，假设（除了抬起的瞬间）机器人支撑脚和地面没有相对移动和相对速度，即约束支撑脚的速度 \dot{p}_c 和加速度 \ddot{p}_c 如：

$$\dot{p}_c = J_c \dot{q} = 0, \quad \ddot{p}_c = J_c \ddot{q} + \dot{J}_c \dot{q} = 0 \quad (2-16)$$

因此根据机器人的当前状态 (q, \dot{q}) 、控制输入 τ 、接触脚情况 J_c 来求解机器人的运动趋势相当于求解动力学和约束共同作用的方程组^[30] (\ddot{q} 和 F_c 是 $n + m_c$ 个变量，构成总共 $n + m_c$ 行的方程组)

$$\begin{cases} A_{rbd}(q)\ddot{q} + g_{rbd}(q, \dot{q}) = \tau + J_c(q)^T F_c \\ J_c \ddot{q} + \dot{J}_c \dot{q} = 0 \end{cases} \quad (2-17)$$

2.2.3 有约束的动力学的模型

本小节根据 operational space^[9] 的原理推导在前一小节2.2.2描述的接触模型中的系统动力学模型。目的是找到式2-17的显式解，利于将动力学化简为2-3的形式²。

考虑接触点的位置约束，我们有

$$\begin{aligned} \ddot{p}_c &= J_c \ddot{q} + \dot{J}_c \dot{q} \\ &= J_c A_{rbd}^{-1} \tau + J_c A_{rbd}^{-1} J_c^T F_c - J_c A_{rbd}^{-1} g_{rbd} + \dot{J}_c \dot{q} \end{aligned} \quad (2-18)$$

²笔者所知的所有地面接触的动力学的代码实现中^[9, 16, 17]，研究者使用的都是直接在控制器中求解式(2-18)，而笔者推出了式(2-20),(2-21)，并认为使用显式表达会更加方便，并且效果与原来等价，故用之。

以接触位置 p_c 作为 operational space, 定义 $\Lambda = (J_c A_{rbd}^{-1} J_c(q)^T)^{-1}$, $\bar{J}_c^T = \Lambda J_c A_{rbd}^{-1}$, 则有

$$\Lambda \ddot{p}_c = \bar{J}_c^T \tau + F_c - \bar{J}_c^T \mathbf{g}_{rbd} + \Lambda \dot{J}_c \dot{q} \quad (2-19)$$

其中 \bar{J}_c 是 \bar{J} 的动力学伪逆^[9]。

联立方程组2-17以及式(2-19), 利用伪逆的性质, 分解 $\tau = J_c^T \bar{J}_c^T \tau + (I - J_c^T \bar{J}_c^T) \tau$, 带入可得消去反作用力 F 的动力学方程:

$$\mathbf{A}_{rbd} \ddot{q} = (I - J_c^T \bar{J}_c^T)(\tau - \mathbf{g}_{rbd}) - J_c^T \Lambda \dot{J}_c \dot{q} \quad (2-20)$$

并且得到地面作用力 F 的表达式:

$$F = \bar{J}g - \Lambda \dot{J} \dot{q} - \bar{J}^T B \tau \quad (2-21)$$

根据刚体的动力学方程(2-14), 我们可以表示为类似式2-3的形式

$$\dot{x} = \begin{bmatrix} \dot{q} \\ q \end{bmatrix} = \mathbf{A}_{cont} x + \mathbf{B}_{cont} u + \mathbf{g}_{cont} \quad (2-22)$$

其中

$$\mathbf{A}_{cont} = \begin{bmatrix} 0 & I \\ 0 & \bar{J}_c \dot{J}_c \end{bmatrix} \quad (2-23)$$

$$\mathbf{B}_{cont} = \begin{bmatrix} 0 \\ \mathbf{A}_{rbd}^{-1} (I - J_c^T \bar{J}_c^T) \end{bmatrix} \quad (2-24)$$

$$\mathbf{g}_{cont} = \begin{bmatrix} 0 \\ -\mathbf{A}_{rbd}^{-1} (I - J_c^T \bar{J}_c^T) \mathbf{g}_{rbd} \end{bmatrix} \quad (2-25)$$

。

除非特殊说明，后文所指的系统动力学模型都是式2-22的动力学模型。

2.3 轨迹优化和 Hybrid Zero Dynamics

本文使用轨迹优化的方法得到驱动机器人步行的基本的步态^[15]，轨迹优化的概念很广，本文中特指优化 **virtual constraint**，以求在 **virtual constraint** 之下的 **Hybrid zero Dynamics (HZD)** 可以形成稳定循环的方法。下面将对有关概念进行详细介绍。

2.3.1 Virtual Constraint 与 Hybrid Zero Dynamics

在介绍本小节之前，先要介绍参考轨迹，对于大多数机器人其行走的过程是对一条参考轨迹的追踪，即机器人抬腿走出一歩的所有动作，每一时刻每一个驱动器需要的输入，都是预先人为设计的，机器人的任务只是在执行这人为设计的轨迹。如果轨迹是按照时钟设计的，即设定好任何时刻的控制器输出，那么当系统受到扰动的时候，小小的偏移就会积累很大的误差。一种更加科学的方法是使用系统自身的状态作为时钟的表示，从而设计轨迹：

$$\eta = h_0(q) - h_d(\theta(q)) \quad (2-26)$$

其中 $\theta(q) \in \mathbb{R}^n \rightarrow \mathbb{R}$ 是从机器人的状态中提取一个对于每一步单调的维度作为步态的时钟信号，比如身体横坐标 (从这一步开始时刻到目前已经平移的距离)，或者支撑腿和竖直方向的角度。 h_0, h_d 定义了系统状态约束的当前值和目标值，控制的目标就是使 η 稳定到 0。

这样形式的轨迹设计被称为 **virtual constraint**，这样可以使身体各个关节的状态通过一个时钟状态来协调。相比跟踪计时的轨迹，使用 **virtual constraint** 可

以让系统应对扰动更加鲁棒。比如一个人在行走的时候被身后一股强风推了一把，他的动作不是重新回复身体角度，去跟随预定的轨迹而是将腿快速移动到对应当前身体角度的位置，满足他的 **virtual constraint**。

对于一个 n 个自由度， m 个控制输入的系统，我们可以设置 m 个 **virtual constraint**。假设这 m 个 **virtual constraint** 都被控制到 0，那么剩下 $n - m$ 个维度的动力学就被称为 **Zero Dynamics**。又因为机器人的摆动脚触地时的状态转换不是连续的（如2.2.2），所以系统是混合了连续与非连续的混合系统，故被称为 **Hybrid Zero Dynamics**。

2.3.2 轨迹优化

Hybrid Zero Dynamics 可以被用来分析轨迹的稳定性，也可以用来设计机器人的参考轨迹，设计的方法就需要使用轨迹优化。具体而言，轨迹优化的参数就是式(2-26)中的 h_d ，轨迹优化的目标可以包含能耗比，最大力矩等等；轨迹优化的约束 **HZD** 稳定性条件，本文在这里不再赘述，读者可以参考 [15] 去看 **HZD** 更加详细的说明。

2.4 本章小结

本章介绍课题涉及的相关技术。首先先着重介绍 **CBF** 的原理、应用、以及目前结合机器学习的相关工作。本文所使用的 **CBF** 为式2-8所定义，要求对于任意安全集合内的状态，有 $\sup_{u \in U} \gamma B(x) + \dot{B}(x, u) \geq 0$ ，在控制器中添加 $\gamma B + \dot{B} > 0$ 约束，可以使系统满足安全集合的前向不变性。

本文介绍了课题中必要的模型以及相关动力学方程，为后文的探讨打下基础。我们介绍了机器人的碰撞模型，并且求解动力学方程以及地面接触点的约束，最后得到了前向动力学方程 $A_{cont}x + B_{cont}u + g_{cont}$ 。

最后简述本课题使用的其它技术，即轨迹优化和 **HZD** 思想在设计机器人控

制器中的作用。机器人的步态设计可以通过设计 **virtual constraint** 来实现，需要两个关键要素：表示每一步进度的时间信号和参考轨迹。轨迹优化即是寻找一个参考轨迹，可以使系统在 **virtual constraint** 都满足的时候的动力学为一个稳定步态。

第三章 CBF 学习算法的研究和设计

本章旨在介绍 CBF 的学习算法，本章将从最核心的设计思路开始，提出理想条件下学习 CBF 的算法流程，证明相关性质。而后针对机器人环境的复杂性对算法进行近似，介绍在机器人环境中的 CBF 学习算法。

3.1 算法思想框架

本文提出的学习 CBF 算法有两个核心部分，仿真采样和对样本状态分类的学习。仿真采样的目的是找到一组有代表性的状态的集合以及它们的类别（安全或不安全）。对样本的分类则是基本的机器学习问题，从离散的样本中拟合出显式的表达。

3.1.1 仿真采样

仿真采样的目的是找到一组有代表性的状态的集合以及它们的类别（安全或不安全）。安全状态具有如下定义：

定义 3.1 (安全状态). 对于如式2-22定义的系统，和给定的避免集合 \mathcal{C}_{avoid} 。假设系统状态 $x \in \mathbb{R}^n$ ， x 被称为安全状态当且仅当存在策略 $\pi(t), t \in [0, \infty)$ 使 $x(t)$ 一直保持在避免集合之外。即

$$\begin{aligned} \text{safe}(x) &\Leftrightarrow \exists \pi \in \mathbb{R}^+ \rightarrow \mathbb{U} \forall t > 0 \ x_\pi(t) \notin \mathcal{C}_{avoid} \\ \text{where } \dot{x}_\pi(t) &= \mathbf{A}_{cont}x_\pi(t) + \mathbf{B}_{cont}\pi(t) + \mathbf{g}_{cont} \\ x_\pi(0) &= x; \end{aligned} \quad (3-1)$$

所有安全状态组成的集合为安全集合 \mathcal{C}_{safe} ，不安全集合 \mathcal{C}_{danger} 则为安全集

合的补集，包含了避免集合和所有无法避免进入避免集合的状态。比如跟车行为中，碰撞属于避免集合，还未碰撞但高速冲向障碍物回天无术的状态属于不安全集合。

由于系统的复杂性，安全集合是无法通过推导找到显式表达的，所以采样的目的就是通过离散的点来近似安全集合。在安全状态的定义3-1中，直接通过仿真和采样证明 $t \in (0, \infty)$ 的安全性是不现实的，所以下面定义 T -安全状态，代表 T 时间内状态的安全性。

定义 3.2 (T -安全状态). 假设系统状态 $x \in \mathbb{R}^n$, x 被称为安全状态当且仅当存在输入 $u(t), t \in [0, T]$ 使 $x(t)$ 一直保持在避免集合之外。即

$$\begin{aligned} \mathbf{T_safe}(x) &\Leftrightarrow \exists \pi \in [0, T] \rightarrow \mathbb{U} \forall t \in [0, T] x_\pi(t) \notin \mathcal{C}_{avoid} \\ \text{where } \dot{x}_\pi(t) &= \mathbf{A}_{cont}x_\pi(t) + \mathbf{B}_{cont}\pi(t) + \mathbf{g}_{cont} \\ x_\pi(0) &= x; \end{aligned} \quad (3-2)$$

基于 T -安全集合 $\mathcal{C}_{T_safe} = \{x | \mathbf{T_safe}(x)\}$ 的概念¹, $(T + \tau)$ -安全集合可以由 T -安全集合和时长为 τ 的仿真求出。因此，可以迭代地求出 $0, \tau, 2\tau, \dots$ -安全集合直至收敛，所得到的即为安全集合。

定理 3.1. 如果 T -安全集合和 $(T + \tau)$ -安全集合相等 ($\tau > 0$)，则 T -安全集合即为安全集合。

$$\mathcal{C}_{T_safe} = \mathcal{C}_{(T+\tau)_safe} \Rightarrow \mathcal{C}_{T_safe} = \mathcal{C}_{safe} \quad (\tau > 0) \quad (3-3)$$

证明. 首先，因为 $(T + \tau)$ -安全集合就是可以在 τ 时间后保留在 T -安全集合

¹注意 T -安全集合与安全集合性质不同，安全集合一定满足 CBF 条件 (式(2-8))，但 T -安全集合不满足

之中的状态的集合。

$$\mathcal{C}_{(T+\tau)_safe} = \{x | \exists \pi, \forall t \in [0, \tau], x_\pi(0) = x, x_\pi(t) \in \mathcal{C}_{T_safe}\} \quad (3-4)$$

假设 $\mathcal{C}_{(T+k\tau)_safe} = \mathcal{C}_{T_safe}$ 试证 $\mathcal{C}_{(T+(k+1)\tau)_safe} = \mathcal{C}_{T_safe}$, 有

$$\begin{aligned} \mathcal{C}_{(T+(k+1)\tau)_safe} &= \{x | \exists \pi, \forall t \in [0, \tau], x_\pi(0) = x, x_\pi(t) \in \mathcal{C}_{(T+k\tau)_safe}\} \\ &= \{x | \exists \pi, \forall t \in [0, \tau], x_\pi(0) = x, x_\pi(t) \in \mathcal{C}_{T_safe}\} \\ &= \mathcal{C}_{(T+\tau)_safe} = \mathcal{C}_{T_safe} \end{aligned} \quad (3-5)$$

因此通过数学归纳法，定理可证。 □

根据定理3.1，我们就有了迭代寻找安全集合的思路：避免集合可以视作是 T 为 0 的不安全集合， $\mathcal{C}_{avoid} = \mathcal{C}_{0_danger}$ ，从 \mathcal{C}_{0_safe} 开始，迭代进行时间跨度为 τ 的仿真和求解，逐渐扩大可以确保安全的时长 T 。本文提出了一些采样算法，其实质都是在通过采样的方式在 T -安全集合的基础上寻找 $(T + \tau)$ -安全集合。

具体而言，给定表示 T -安全集合的函数表示 $B \in \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathcal{C}_{T_safe} = \{x | B(x) \geq 0\}$, 通过一种随机采样一些状态的集合 X , 以及表示 X_i 安全性的 y_i 。

$$(X, y) = \text{Sample}(B_{T_safe}, \tau) \quad (3-6)$$

$$y_i = \begin{cases} 1 & x \in \mathcal{C}_{(T+\tau)_safe} \\ -1 & x \notin \mathcal{C}_{(T+\tau)_safe} \end{cases} \quad X_i \in \mathbb{R}^n$$

3.1.2 CBF 拟合

CBF 拟合指的是对于上一小节所述的仿真采样环节中得到的 T -安全集合和 T' -不安全集合中的采样, 拟合出一个显式的函数来区分二者。

本文采取类似 SVM (Support Vector Machine) ^[31] 的算法进行分类, 拟合过程可以被建模成如下的优化问题

$$\begin{aligned}
 & \min_{w, b, \xi_i \geq 0} \|w\| + c \|\xi\|_1 \\
 & \text{s.t.} \quad \forall i \quad y_i B(x_i) > 1 - \xi_i \quad (\text{SVM 约束}) \\
 & \quad \forall x \in \{x | B(x_i) \geq 0\} \quad \exists u \in \mathbb{U} \quad \dot{B}(x, u) + \gamma B(x) \geq 0 \quad (\text{CBF 约束}) \\
 & \quad \mathcal{C}_{protected} \subset \{x | B(x) \geq 0\} \quad (\text{保护集合})
 \end{aligned} \tag{3-7}$$

其中采集的数据 x_i, y_i 分别是状态空间中的状态和安全类别 ($y_i = 1$ 代表安全集合, $y_i = -1$ 代表不安全集合)。 \hat{x} 是对于 x 的特征补充, 比如扩充到二次的特征代表二次型的 SVM, B 是特征权重为 w 时的 CBF 函数,

$$B(x) = w^T \hat{x} + b \quad \dot{B}(x, u) = w^T \frac{\partial \hat{x}}{\partial x} (\mathbf{A}_{cont} x + \mathbf{B}_{cont} u + \mathbf{g}_{cont}) \tag{3-8}$$

式3-7所表示的优化问题是标准的 SVM 问题外加 CBF 约束 (见 CBF 定义式2-8), 所以这个优化问题直观地表示寻找可以区分两组集合同时满足 CBF 定义条件的函数。此优化问题中的 CBF 约束要求对于上水平集中的任意 x 都成立, 而这个“任意”无法被直接处理。所以, 下文介绍的具体算法会主要针对这个“任意”条件进行放松或者近似。

3.1.3 算法总体流程

本文提出的 CBF 学习算法的框架便是迭代前文所述的采样和拟合，在一个大循环中执行：1. 基于表示 T -安全集合的函数 B_{T_safe} 进行跨度为 τ 的仿真，得到 $\mathcal{C}_{(T+\tau)_safe}$ 中的采样。2. 基于 $\mathcal{C}_{(T+\tau)_safe}$ 中的采样得到 $(T + \tau)$ -安全集合的函数 $B_{(T+\tau)_safe}$ 。具体算法见1。

Algorithm 1 学习 CBF

Require: $B_{avoid} \in \mathbb{R}^n \rightarrow \mathbb{R}, \mathcal{C}_{protected}, \tau > 0$

$\mathcal{C}_{avoid} = \{x | B_{avoid}(x) < 0\}$

Ensure: $\mathcal{C}_{protected} \cap \mathcal{C}_{avoid} = \emptyset$

$k \leftarrow 0$

$B_{k_safe} \leftarrow B_{avoid}$

DO

$(X, y) \leftarrow \text{Sample}(B_{k_safe}, \tau)$

$B_{(k+1)_safe} \leftarrow \text{FitCBF}(X, y)$

$k \leftarrow k + 1$

WHILE $(\exists i (B_{k-1}(X_i) > 0 \wedge B_k(X_i) < 0))$ {当 $\mathcal{C}_{(k-1)_safe} = \mathcal{C}_{(k)_safe}$ 时停止}

3.2 足式机器人背景下的 CBF 学习算法

由于足式机器人是一个十分复杂而且欠驱动的系统，上一节中所述的算法框架无法实现，所以必须对上一节的算法框架近似。本节将详细描述在足式机器人背景下的 CBF 学习算法。

本文选择线性 CBF 的组合来作为控制器的 CBF 以及要学习的目标形式

$$\mathcal{B} = \{B_1, B_2, \dots, B_m\} \quad B_i(x) = w_i x + b \quad (3-9)$$

在这样的设置下，CBF 的上水平集是空间中的一个凸多面体。算法的过程就是不断增加 B_i ，使得多面体越来越精细的过程。

3.2.1 足式机器人背景下的仿真采样

足式机器人背景下的仿真采样的任务是，给定代表 T -安全集合的函数 B_{T_safe} ，采样得到一组点集 X ，以及 y_i 代表表示 X_i 是否属于 $(T + \tau)$ -安全集合。其中 X 的选取要尽可能有代表性，有助于 $B_{(T+\tau)_safe}$ 的拟合。

本文提出的方法是，首先需要给定一个可以设置 CBF 的控制器 (例如式2-9中的形式) 和仿真环境 $\text{simulate} \in \mathcal{B} \rightarrow \xi$ ，使用控制器进行仿真，仿真时的 CBF 设置为 B_{T_safe} 。由于 B_{T_safe} 不一定具有 CBF 性质 (2-8)，故仿真过程中不能确保所有状态下控制器有解。这些无解的状态 (记为 x_0) 就可以被视作 T -安全集合 \mathcal{C}_{T_safe} 的边界。对于一个处于 T -安全集合边界的状态，我们将状态局部的动力学方程线性化，反推得到一个 $\mathcal{C}_{(t+\tau)_safe}$ 边界上的状态点，作为不安全状态点。同时取仿真轨迹中， x_0 的 τ' 时间前的状态 ($\tau' > \tau$)，作为安全状态点。

具体而言，通过仿真得到一条状态空间中的轨迹 $\xi \in [0, t] \rightarrow \mathbb{R}^n$ ，其中 t 是轨迹 ξ 的时长，轨迹的最后一个状态 $x_0 = \xi(t)$ 由于无法满足 B_{T_safe} 的约束 (式2-9) 而终止。 x_0 处对 B_{T_safe} 和系统的动力学方程进行线性化，定义在 x_0 处的局部的系统离散化动力学方程为，

$$x(t + \tau) = \mathbf{A}_{\tau_discrete}x(t) + \mathbf{B}_{\tau_discrete}u + \mathbf{g}_{\tau_discrete} \quad (3-10)$$

其中²

$$\mathbf{A}_{\tau_discrete} = \exp(\mathbf{A}_{cont}(x_0)\tau) \quad (3-11)$$

$$\mathbf{B}_{\tau_discrete} = \int_0^\tau \exp(\mathbf{A}_{cont}(x_0)t)dt \mathbf{B}_{cont}(x_0) \quad (3-12)$$

$$\mathbf{g}_{\tau_discrete} = \int_0^\tau \exp(\mathbf{A}_{cont}(x_0)t)dt \mathbf{g}_{cont}(x_0, \dot{x}_0) \quad (3-13)$$

² $\mathbf{B}_{\tau_discrete}$ 和 $\mathbf{g}_{\tau_discrete}$ 的积分式虽然复杂，但是有简便的矩阵运算可以求得

定义如下的优化问题来反推 $B_{(T+\tau)_{safe}}$ 处的边界状态。

$$\begin{aligned}
 (x^*, u^*) &= \underset{x, u \in \mathbb{U}}{\operatorname{argmin}} \|x - x_0\|_2^2 \\
 \text{s.t. } &B_{T_{safe}}(x) \geq 0 \\
 &\gamma B_{T_{safe}}(x) + \dot{B}_{T_{safe}}(x, u) \geq 0 \\
 &B_{T_{safe}}(x') \geq 0 \\
 &\gamma B_{T_{safe}}(x') + \dot{B}_{T_{safe}}(x', u) \geq 0
 \end{aligned} \tag{3-14}$$

其中有

$$x' = \mathbf{A}_{\tau_{discrete}}x + \mathbf{B}_{\tau_{discrete}}u + \mathbf{g}_{\tau_{discrete}} \tag{3-15}$$

$$\dot{B}_{T_{safe}}(x, u) = \frac{\partial B_{T_{safe}}}{\partial x_0} \left(\mathbf{A}_{cont}(x_0)x + \mathbf{B}_{cont}(x_0)u + \mathbf{g}_{cont}(x_0, \dot{x}_0) \right) \tag{3-16}$$

优化问题3-14的意思是在 x_0 的附近找到 $\mathcal{C}_{(T+\tau)_{safe}}$ 的边界点。式中的约束表明了优化问题的解 x^* 可以通过 u^* 作用时间 τ 得到状态 x' ，保持 $T + \tau$ 时刻的 x' 仍然处于 $\mathcal{C}_{T_{safe}}$ 内。所以，如果忽略系统动力学线性化所带来的误差，这个优化问题的解是 $\mathcal{C}_{(T+\tau)_{safe}}$ 边界上一点³。由于这个优化问题的形式神似近似点算子^[32]，所以后文中也用 $x^* = \underset{\delta \tilde{\mathcal{C}}_{(T+\tau)_{safe}}}{\operatorname{prox}}(x_0)$ 表示⁴。

由于适当的线性化，读者可以验证优化问题3-14是标准的 QP 问题，目标函数是二次型，约束均为线性，这为采样算法的快速求解提供了方便。

在算法的实际使用时，笔者使用了一个参数 ϵ 来确保优化结果仍处在线性化的区域内 $\|x^* - x_0\|_2 \leq \epsilon$ 。如果 x^* 超出了 x_0 附近 ϵ 的范围，则将 x^* 投影进

³严谨而言，还需要假设式中对 $x^*(0)$ 和 $x^*(\tau)$ 两个端点的约束等价于对这时长为 τ 的轨迹中所有时刻的约束，以及忽略限制 u 在 τ 时间内保持不变所产生的影响

⁴ δ 表示示性函数，如果 x 属于集合 \mathcal{C} ，函数值为 0，反之正无穷。 $\tilde{\mathcal{C}}$ 代表这里使用的只是线性化近似的 \mathcal{C} 集合

其邻域。

足式机器人环境下的采样算法即为重复采样以及优化3-14的操作，具体算法如算法2所示。

Algorithm 2 足式机器人环境下的采样算法

Require: $B_{T_safe} \in \mathbb{R}^n \rightarrow \mathbb{R}, \tau > 0, \tau' > 0, N > 0, \text{simulate} \in B \rightarrow \xi$
 $X_{safe} \leftarrow \emptyset, X_{danger} \leftarrow \emptyset, k \leftarrow 0$
while $k < N$ **do**
 $\xi \leftarrow \text{simulate}(B_{T_safe})$
 $t \leftarrow \text{length}(\xi)$
 $x_0 \leftarrow \text{lastState}(\xi)$
 $x_{danger}^* \leftarrow \text{prox}_{\delta \tilde{\mathcal{C}}_{(T+\tau)_safe}}(x_0)$
 $X_{danger} = X_{danger} \cup \{x_0, x_{danger}^*\}$
 $X_{safe} = X_{safe} \cup \{\xi(t - \tau')\}$
 $k \leftarrow k + 1$
end while

注意，算法中 $\xi(t - \tau')$ 得到的是 $t - \tau'$ 时刻的状态 $x(t - \tau')$ 和控制器输入 $u(t - \tau')$ 。需要记录控制器输入信息的原因是为了方便下一小节3.2.2中拟合的 CBF 更加平滑。

3.2.2 足式机器人环境中的 CBF 拟合

拟合 CBF 中最大的问题就是式3-7中 CBF 约束的任意条件如何表达。本章提出的拟合 CBF 的算法，其实是对 CBF 约束的放宽。也就是求解本章优化问题得到的函数并不是一个满足 CBF 定义式2-8条件的函数，但这个缺陷会由整个 CBF 学习算法的大循环过程所弥补。

本章提出的 CBF 拟合算法是如下优化问题

$$\begin{aligned}
 & \min_{\substack{w \in \mathbb{R}^n, b \in \mathbb{R}, \xi_i \geq 0 \\ u_1, u_2, \dots, u_{n_{safe}} \in \mathbb{U}}} \|w\|_2^2 + \lambda_\xi \|\xi\|_1 + \lambda_{du} \sum_{(x_i, u_i^*) \in X_{safe}} \|u_i - u_i^*\|_2^2 \\
 & \text{s.t. } \forall i \quad y_i B(x_i) > 1 - \xi_i \quad (\text{SVM 约束}) \\
 & \quad \forall (x_i, u_i^*) \in X_{safe} \quad \dot{B}(x_i, u_i) + \gamma B(x_i) \geq 0 \quad (\text{CBF 近似约束}) \\
 & \quad \mathcal{C}_{protected} \subset \{x | B(x) \geq 0\} \quad (\text{保护集合})
 \end{aligned} \tag{3-17}$$

其中 $\lambda_\xi, \lambda_{du}$ 分别是两个正权重, B 和 \dot{B} 的定义如式3-8。这个优化问题和式3-7的区别在于 CBF 近似约束取代了 CBF 约束。CBF 近似约束仅仅作用在安全集合的采样点上, 用对于所有样本满足来代替原本约束中的对所有状态满足。同时, 目标函数的第二项表示 CBF 近似约束还被鼓励使用原轨迹中的 u^* 来满足, 这样的设计意在保证新加的 CBF 函数不会对原始的轨迹产生太大的影响。

这个优化问题是非凸优化, 所以速度和最优性都会受到影响。在实践中求解这个优化问题所消耗的时间也占用了 CBF 学习算法的大部分, 但在下一章节的实验部分说明这个优化问题的解的最优性还比较有保证。

3.3 本章小结

本章提出 CBF 学习算法的思想框架并且定义相应的概念。CBF 学习算法的核心是不断根据 T -安全集合和时间跨度为 τ 的仿真, 得到 $(T+\tau)$ -安全集合。具体而言, 重复采样和拟合两大关键步骤。1. 基于 T -安全集合函数 B_{T_safe} 进行跨度为 τ 的仿真, 得到 $\mathcal{C}_{(T+\tau)_safe}$ 中的采样。2. 基于 $\mathcal{C}_{(T+\tau)_safe}$ 中的采样得到 $(T+\tau)$ -安全集合的函数 $B_{(T+\tau)_safe}$ 。

为了适应足式机器人的大背景, 我们在基本原理上做出近似和改动。在仿真采样步骤中, 我们采用了一个类似近似点算子的操作 $x^* = \underset{\delta \tilde{\mathcal{C}}_{(T+\tau)_safe}}{\text{prox}}(x_0)$, 对

于仿真轨迹中无法满足 CBF 约束的状态, 局部地反推 τ 时间, 得到从 $\tilde{\mathcal{C}}_{(T+\tau)_{safe}}$ 中采样得来的状态。在拟合步骤中, 我们将式3-7中的 CBF 约束放松到了只用对于数据集中的状态点成立。

第四章 实验验证

本文工作有大量的实验支撑，总体而言，本文达到的效果为：找到了可以保障机器人步行不会摔倒的 CBF, 即 CBF 可以与原本不稳定的参考轨迹和控制器参数共同作用，得到机器人稳定的步态。本章将先介绍在玩具问题中对算法框架的验证，让读者对算法的框架有一个更加直观的概念。之后介绍机器人环境中的实验，主要有两个任务，分别是学习限制机器人范围的 CBF，以及学习可以让机器人平稳步行的 CBF。

4.1 玩具问题中对算法框架的验证

4.1.1 玩具问题问题描述

假设一个平面移动的系统，系统的状态空间为 x, y 方向的位置和速度，控制输入为两个方向的加速度

$$\begin{aligned} x &= (p_x, p_y, v_x, v_y) \\ u &= (a_x, a_y) \quad -20 \leq a_x, a_y \leq 20 \end{aligned} \tag{4-1}$$

系统的避免集合是位于原点的单位圆，即

$$\mathcal{C}_{avoid} = \{x | p_x^2 + p_y^2 < 1\} \tag{4-2}$$

控制器使用式2-9所示的控制器，任务是学习一个 CBF, 以确保控制器无论使用何等的 CLF，系统状态都可以保持在避免集合之外。

4.1.2 玩具问题具体算法

本节介绍在玩具问题背景下，3.1中所描述的算法流程的具体实现。

4.1.2.0.1 采样算法 在玩具问题中，我们采用上水平集估计的思想^[33]进行采样。上水平集估计的问题描述为，对于一个十分耗时的函数 $s \in \mathbb{R}^n \rightarrow \mathbb{R}$ ，如何选取数目最少的样本进行测试，尽可能减少计算函数 s 的次数，以求估计出函数的上水平集。由于篇幅限制，本文不再赘述上水平集估计的具体过程和原理，请读者参考^[33]。

上水平集估计的算法可以在玩具问题中被很好的利用，只需定义函数 s 使之描述输入的状态是否在安全集合之中即可，即给定 B_{T_safe} ，我们要求 $B_{(T+\tau)_safe}$ 的上水平集，直接通过 $B_{(T+\tau)_safe}$ 的定义构建如下优化问题作为函数 s ，

$$s(x_0) = \max_{u_1, u_2, \dots, u_k} \min_k (B_{T_safe}(x(k, x_0, \mathbf{u}_k))) \quad (4-3)$$

其中，我们在 $0 - \tau$ 时间段中等分选取 κ 个时刻， $u_1, u_2, \dots, u_\kappa$ 是这些时刻的输入， $x(k, x_0, \mathbf{u}_k)$ 是从 x_0 时刻开始，在 u_1, u_2, \dots, u_k 输入的作用下， k 时刻状态。所以这个优化问题的意思就是，给定当前状态 x ，寻找 τ 时间内它满足函数 B_{T_safe} 的能力的上界。

在这样的设置下，并且假设式4-3描述的非凸优化问题可以找到最优解， $s(x) = 0$ 的等高线即是 $B_{(T+\tau)_safe}$ 的边界。由于求解 $s(x)$ 的时间成本较高，所以套用上水平集估计的方法首先选择最值得求解的状态点 x ，将 $x, y = \text{sign}(s(x))$ 添加进采样的状态集合。

4.1.2.0.2 拟合算法 由于式(3-7)中的 CBF 约束的任意条件难以表达, 式(3-7)被简化成如下的形式:

$$\begin{aligned}
 & \min_{w, b, \xi_i \geq 0} \|w\| + c \|\xi\|_1 \\
 & \text{s.t. } \forall i \quad y_i B(x_i) > 1 - \xi_i \quad (\text{SVM 约束}) \\
 & \forall x \in \left\{ x \left| \argmin_x \sup_{x, B(x) > 0} \dot{B}(x, u) + \gamma B(x) \right. \right\} \sup_{u \in \mathbb{U}} \dot{B}(x, u) + \gamma B(x) \geq 0 \quad (\text{CBF 约束})
 \end{aligned} \tag{4-4}$$

式(4-3)中有两个优化问题, $\min_{w, b, \xi_i \geq 0}$ 和 \argmin_x 在代码实现中各为一个独立的优化问题, 两个优化问题放在一个大循环里面迭代得到最后 CBF 的拟合结果。由于 \mathbb{U} 的形式是无穷范数的约束, 式中的 $\sup_{u \in \mathbb{U}} \dot{B}(x, u) + \gamma B(x)$ 具有解析形式。

4.1.3 玩具问题实验结果和分析

玩具问题中采样算法, 拟合算法, 和仿真结果如图4-1所示。4.1(a), 绿色和红色的点分别是上水平集估计算法认定的安全集合和危险集合中的点, 点处的箭头代表该点的速度方向。蓝色的叉为算法实际测试, 执行 $s(x)$ 的点, 算法测试了很多距离远但是速度方向指向原点的非安全点, 和距离近但背离原点的安全点, 可以看出算法主要测试的是难以判别安全或危险的状态。4.1(b)是使用4-4拟合 CBF 的结果, 同样绿色和红色分别为安全以及危险的点, CBF 使用一个二次函数来参数化, 图中蓝为函数值为 0 的等高线 (速度维度假设为 0) 4.1(c)是使用玩具问题中的 CBF 学习算法习得的 CBF, 最终进行控制的结果, 红色区域表示避免集合, 红色虚线为最终习得的 CBF, 绿色为 CLF 目标轨迹, 蓝色轨迹为实际仿真得到的轨迹。可见在 CBF 的作用下, 系统可以安全避开避免集合。在更多的仿真实验中, 无论 CLF 如何设定, 系统都可以保证安全。

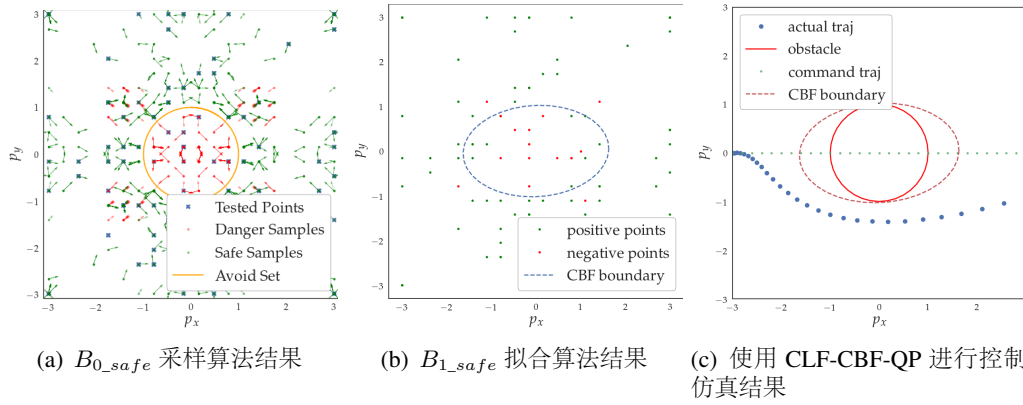


图 4-1 玩具问题中采样算法，拟合算法，和最后仿真的结果。三张图片均为表示 p_x, p_y 的平面图 4.1(a)，绿色和红色的点分别是上水平集估计算法认定的安全集合和危险集合中的点，点处的箭头代表该点的速度方向。蓝色的叉为算法实际测试的点，算法测试了很多距离远但是速度方向指向原点的非安全点，和距离近但背离原点的安全点。4.1(b)，使用 4-4 拟合 CBF 的结果，同样绿色和红色分别为安全以及危险的点，CBF 使用一个二次函数来参数化，图中蓝圈为函数值为 0 的等高线（假设速度为 0）。4.1(c)，使用玩具问题中的 CBF 学习算法习得的 CBF，最终进行控制的结果，红色区域表示避免集合，红色虚线为最终习得的 CBF，绿色为 CLF 目标轨迹，蓝色轨迹为实际仿真得到的轨迹。可见在 CBF 的作用下，系统可以安全避开避免集合。

4.2 机器人环境实验设置

本文的机器人实验使用 `pybullet` 物理引擎进行仿真^[34]，`pybullet` 是一个常见于增强学习任务的仿真引擎，提供了正反动力学仿真，正反运动学计算的功能。环境中设置重力加速度 $9.8m/s$ ，仿真的单步时长为 $1e-3s$ 。`pybullet` 引擎使用的碰撞检测方法是基于 `spring-damper` 的碰撞模型^[29]，即在检测到物体发生碰撞时，法向接触力的大小和两个物体挤压的深度成正比，像弹性系数很大的弹簧一样，外加关于相互速度的阻尼。请读者注意区分 `spring-damper` 模型和前文一直使用的约束和碰撞模型 2-22，在实验中，这些区别直接导致在基于 2-22 可以稳定的机器人步态库无法在基于 `spring-damper` 模型的 `pybullet` 环境中稳定。而本文所学出的 CBF，可以最终维持这些轨迹的稳定。

本文实验所使用的机器人模型叫 `RABBIT`^[15]，是一个平面机器人，如下图所示。`RABBIT` 机器人有七个自由度，分别是身体中心的位置和角度 $p_x \in \mathbb{R}, p_y \in \mathbb{R}$

\mathbb{R}^+ , $r \in [-\pi, \pi)$ 和两条腿的大腿和膝盖处的角度 $q_1, q_3 \in [0, 2\pi)$, $q_2, q_4 \in [-\pi, \pi)$ 。躯干长 0.63 米，大腿小腿各长 0.4 米。左右腿完全对称，虽然平面机器人不分左右，但是后文仍会使用左腿来代指 q_1, q_2 对应的腿，右腿代指另外两个关节对应的腿。机器人有四个驱动器控制着大腿和膝盖 ($q_1 \dots q_4$)，输出力矩范围均是在 $100Nm$ 以内。机器人没有脚掌，足尖处是单点接触地面，这使得机器人成为一个严重欠驱动的系统，为设计机器人的控制算法带来了难度。

所以在机器人背景下，动力学方程2-22中的状态空间表达为

$$x = (p_x, p_y, r, q_1, \dots, q_4, \dot{p}_x, \dots, \dot{q}_4) \quad (4-5)$$

$$u = (u_1, u_2, u_3, u_4) \quad u_i \in [-100, 100] \quad \forall i = 1 \dots 4, \quad (4-6)$$

共十四个状态维度, 四个输入维度。

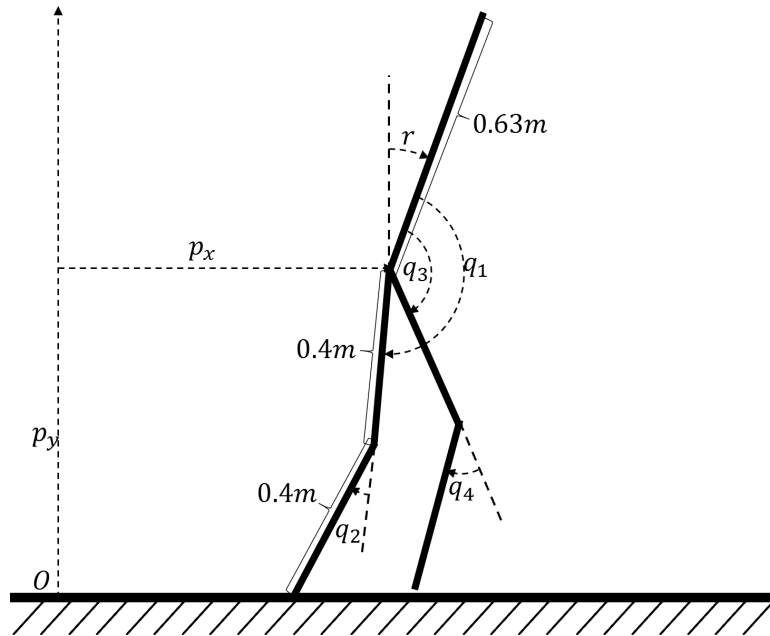


图 4-2 RABBIT 机器人的状态和参数示意图。RABBIT 机器人有七个自由度，分别是身体中心的 p_x, p_y, r 和两条腿的大腿和膝盖处的角度。机器人有四个驱动器控制着大腿和膝盖 ($q_1 \dots q_4$)。机器人没有脚掌，足尖处是单点接触地面。

对于 CBF 而言，系统的状态不仅仅是决定变化方式的变量，还作为用来判断安全性并以此进行控制的特征。因此，系统状态的设计还需要方便 CBF 函数把安全集合包在状态空间中的一个凸集里，所以在进行本文最后一个实验——机器人步行实验时，我们定义机器人关节的状态时，区分摆动腿和支撑腿，并且加入时间信号作为状态，

$$\tilde{x} = (p_x, p_y, r, \tilde{q}_1, \dots, \tilde{q}_4, \theta, \dot{\tilde{q}}_1, \dots, \dot{\tilde{q}}_4, \dot{\theta}) \quad (4-7)$$

其中 \tilde{q}_1, \tilde{q}_2 为支撑腿的大腿和膝盖角度， \tilde{q}_3, \tilde{q}_4 为摆动腿的相应角度¹。 θ 为时钟变量，表示当前步的进度。

本章后续所有机器人背景下的实验中，使用的控制器为：

$$\begin{aligned} u^* &= \underset{u}{\operatorname{argmin}} \quad \|u - \phi(x)\|_2^2 \\ \text{s.t.} \quad & \dot{B}(x, u) + \gamma B(x) \geq 0 \quad (\text{ES-CBF}) \\ & F_z \geq 0 \quad (\text{Contact Force Constraint}), \quad (4-8) \\ & \left\| \begin{bmatrix} F_x \\ F_y \end{bmatrix} \right\|_2 \leq F_z \quad (\text{Friction Constraint}) \end{aligned}$$

其中 $\phi(x)$ 是基于参考轨迹进行 PD 反馈控制得到的参考输入，参考轨迹本质是一个四个参数定义的贝塞尔曲线²，如下图所示。

$\phi(x)$ 输入这一步时间的信号 $p_x - p_{x0}$ ，得到四个关节的参考位置和速度 $(\hat{q}_1, \dots, \hat{q}_4, \dot{\hat{q}}_1, \dots, \dot{\hat{q}}_4)$ 参考输入 u 既与这参考关节的速度和加速度成 PD 反馈关系。

¹这样会使系统在支撑腿落地的刹那不连续，所以系统成为2.3.1所描述的 hybrid dynamic system，本文中我们忽略其对我们进行控制和分析的影响

²参考轨迹来源于轨迹优化，笔者使用共享在 github 上的开源项目^[35]得到的参考轨迹

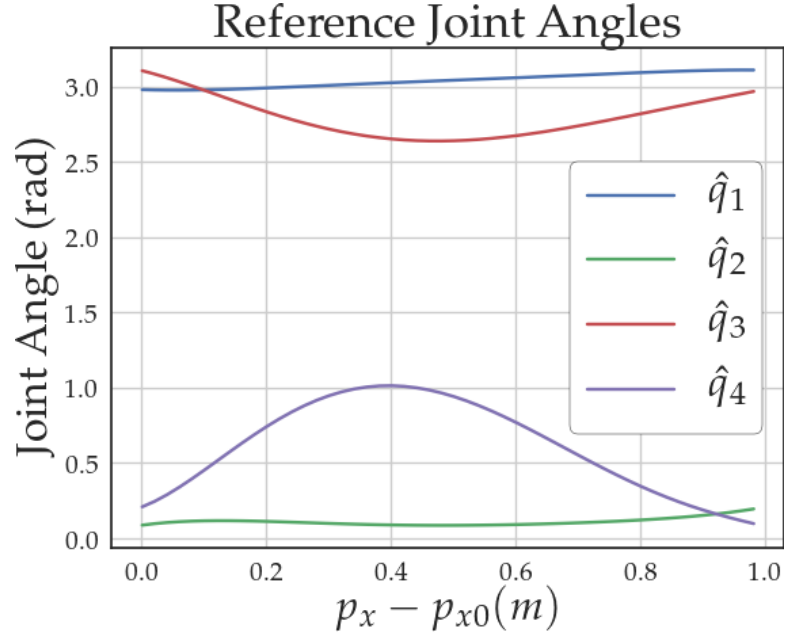


图 4-3 机器人步行参考关节角度示意图，关节的参考轨迹通过轨迹优化计算得到，是一个贝塞尔曲线，横坐标是步态的时钟信号用当前 x 轴坐标和这一步开始的 x 轴坐标的差表示

$$\phi(x) = k_p \left(\begin{bmatrix} \hat{q}_1 \\ \vdots \\ \hat{q}_4 \end{bmatrix} - \begin{bmatrix} q_1 \\ \vdots \\ q_4 \end{bmatrix} \right) + k_d \left(\begin{bmatrix} \hat{\dot{q}}_1 \\ \vdots \\ \hat{\dot{q}}_4 \end{bmatrix} - \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_4 \end{bmatrix} \right) \quad (4-9)$$

4.3 学习 CBF 控制机器人关节范围

本节使用一个简单的小实验来表明机器人背景下 CBF 学习算法的有效性，同时通过低维度的可视化表现 CBF 学习和采样算法背后的原理。

具体任务为：在机器人行走时，通过添加 CBF 约束的方式保证机器人的左腿膝关节 (q_2) 的角度在 0° 到 180° 之间。

4.3.1 实验设置

设置机器人背景下的 CBF 学习算法的避免集合和起始 CBF 函数为

$$\mathcal{C}_{avoid} = \{x | q_2 < 0\} \quad (4-10)$$

$$B_{0_safe}(x) = q_2 \quad (4-11)$$

使用如4-8的控制器，用如图4-3所示的参考轨迹运行算法。实验使用的参数见表4-1:

表 4-1 学习 CBF 控制机器人关节范围实验参数表

Parameter	Value
采样轨迹数	20
τ	0.01
τ'	0.1
γ	100
λ_ξ	100
λ_{du}	100

4.3.2 实验结果和分析

实验结果如下图4-4所示，其中横轴是膝关节的角度，纵轴是膝关节的角速度³。图中的点均是安全集合或者非安全集合中的采样，线代表 CBF 函数值为 0 的等高线。算法初始设置是所有横坐标小于零的状态都属于避免状态 \mathcal{C}_{avoid} (灰线所示)，CBF 采样算法所进行的仿真中，系统发展到黑色的状态 x_{avoid} 时，便无法找到满足式2-9 中 CBF 约束的解了。采样算法将这些无法满足约束的点采集起来，之后使用式3-14的算法反推在局部动力学下的系统极限（红色的点所示）。在采样得到的数据集上进行拟合，得到新的 CBF(蓝色的线所示) 安全状

³虽然图表只有两维，但是这个实验仍然是在 $n = 14$ 维空间中进行的，本节中的图是将状态和 CBF 都投影到了 q_2, \dot{q}_2 二维上得到的，图中在绘制 CBF 等高线时，假设其余的维度是整条轨迹的平均值。

态缩小到了蓝线的上方。用本文提出的 T -安全集合3-1的概念来解释, 这一步算法从灰线代表的 \mathcal{C}_{0_safe} , 采样得到图中黑红绿三色点的集合, 拟合出了 \mathcal{C}_{τ_safe} 所代表的函数。

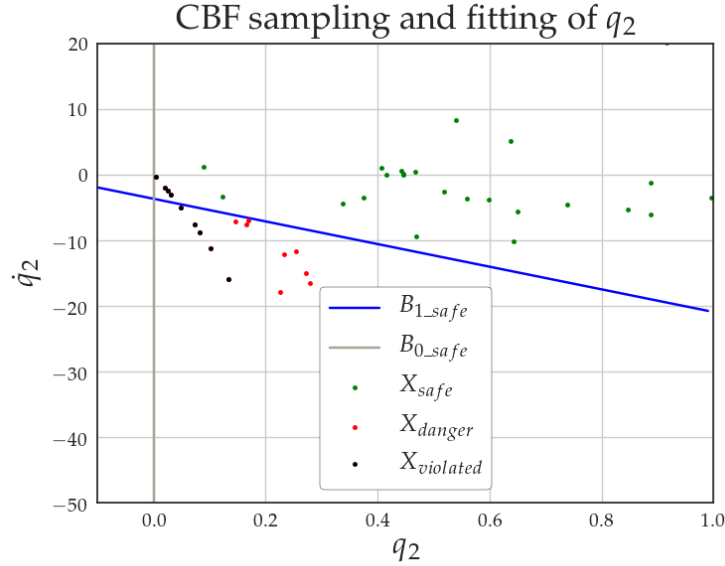


图 4-4 机器人膝关节角度限制 CBF 采样和拟合图。图中横轴是膝关节的角度, 纵轴是膝关节的角速度。图中的点均是安全集合或者非安全集合中的采样, 线代表 CBF 函数值为 0 的等高线。算法初始设置是所有横轴正半轴方向的状态都是安全状态 (灰线), CBF 采样和拟合一轮之后缩小了安全状态的到蓝线的上方。图中红黑色和绿色的点分别是采样得到的非安全集合中的状态和安全集合中的状态, 蓝线分开了安全集和非安全集中的点。非安全集中, 黑色的点为仿真轨迹的结束状态 (违反了 CBF 约束的状态), 旁边的红点是使用 3-14 得到的反推了 $\tau = 0.01s$ 时间的状态, 可见使用 3-14 可以更进一步地推进危险集合。

将学习得到的 CBF 应用在机器人上, 可以发现机器人可以保持不进入避免集合 \mathcal{C}_{avoid} ⁴。如图 4-9 所示, 在没有 CBF 约束时 (图 4.5(b)), 机器人的膝关节会反弯, 进入到横坐标为负的空间中, 在增加了 CBF 约束后, 系统可以一直保持在避免状态之外 (图 4.5(a))。注意由于 x 是十四维的状态, 所以图中求取等高线时假设 x 的其他维度是整条轨迹的平均值, 这个平均值和机器人处在蓝线附近时的状态的误差可以解释图 4.5(a) 中, 机器人轨迹边界和 $B_{1_safe}(x) = 0$ 等高线

⁴ 尽管不进入避免集合, 但是仿真中机器人并没有形成稳定循环 (并不能无限长地步行下去)

之间的间隔。

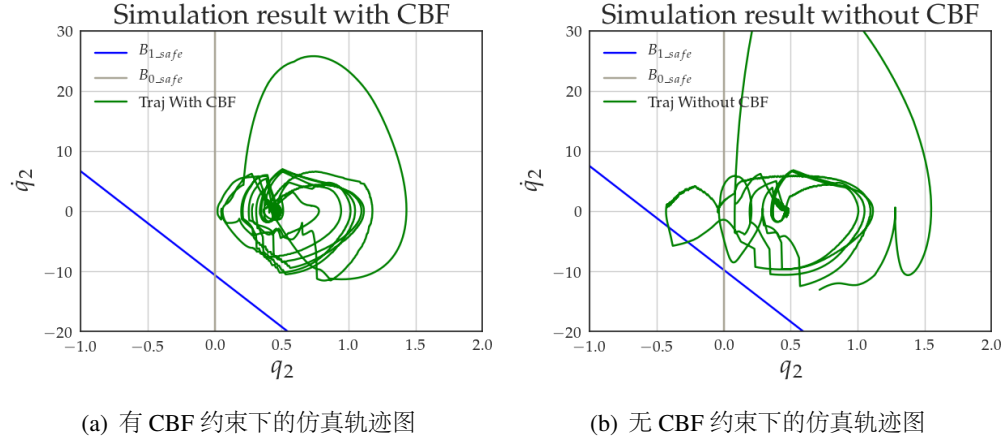


图 4-5 使用 CBF 约束和不使用 CBF 约束的效果对比。图中的绿色线是机器人仿真的整个轨迹中膝关节的相图。可以看到在使用了学习算法得到的 B_{1_safe} 之后 (图4.5(a)), 机器人可以被完全控制在避免集合之外。图中蓝色线是 $B_{1_safe}(x) = 0$ 的等高线。

图4-6展示了在有 CBF 约束的控制器的作用下, $B_{0_safe}(x)$ 和 $B_{1_safe}(x)$ 函数值随时间的关系。其中 $B_{0_safe}(x)$ 的函数值一直大于零, $B_{1_safe}(x)$ 在系统最后机器人跌倒时进入了负半轴, 说明 $B_{1_safe}(x)$ 可以保护 $B_{0_safe}(x)$, 同时 $B_{1_safe}(x)$ 本身还并满足2-8的条件, 因为它对应的上水平集 \mathcal{C}_{1_safe} 的前向不变性在轨迹的最后没有得到保证。再继续学习更多的 CBF 可以保护 $B_{1_safe}(x)$ 同时使整体的安全集合具有前向不变性。

4.4 学习 CBF 确保机器人稳定步行

本节旨在使用足式机器人背景下的 CBF 学习算法, 得到可以使机器人稳定步行的控制器⁵。

⁵机器人稳定步行的测试标准是在仿真环境中连续行走 20 秒且没有异常状况发生

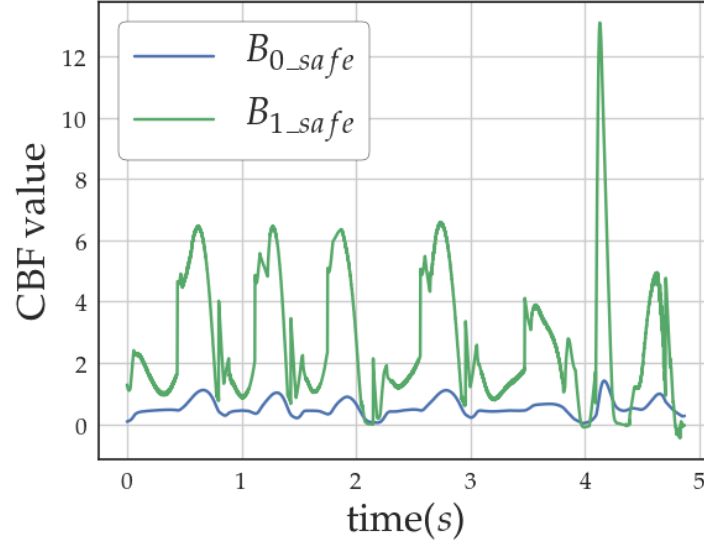


图 4-6 仿真轨迹中 $B_{0_safe}(x)$ 和 $B_{1_safe}(x)$ 函数值随时间的关系。蓝色线是 CBF 学习算法初始的避免集合的函数值 $B_{0_safe}(x)$ ，从图中可以看出其一直保持在正半轴。绿色线是 CBF 学习算法学出的 $CBFB_{1_safe}(x)$ ，从图中可以看出它除了末尾机器人跌倒时的特殊情况外都保持在正半轴。

4.4.1 实验设置

参考^[19] 中的步行机器人 CBF 设置，本文使用了初始的六个 CBF 来定义避免集合 \mathcal{C}_{avoid}

• B_{1_safe}, B_{2_safe} 首先两个约束是上一节 4.3 中对关节范围的约束。在本任务中，我们略微放宽了对膝关节的限制，变为 $q_2, q_4 > -0.1$ 。即

$$B_{1_safe} = q_2 + 0.1 \quad B_{2_safe} = q_4 + 0.1 \quad (4-12)$$

• B_{3_safe}, B_{4_safe} 其次我们定义机器人摔倒的条件，如果机器人的任意一个足尖到身体中心的角度不属于 $[3.0, \frac{5}{4}\pi](\text{rad})$ 之间 (角度小于 3 意味着机器人在后仰，

大于 $\frac{5}{4}\pi$ 意味着机器人向前跌倒), 则视为跌倒, 所以有

$$\begin{aligned} B_{3_safe}(x) &= -(r + q_1 + \frac{1}{2}q_2 - 3)(r + q_1 + \frac{1}{2}q_2 + \frac{5}{4}\pi) \\ B_{4_safe}(x) &= -(r + q_3 + \frac{1}{2}q_4 - 3)(r + q_3 + \frac{1}{2}q_4 + \frac{5}{4}\pi) \end{aligned} \quad (4-13)$$

上式中 $q_1 + \frac{1}{2}q_2$ 的形式是因为机器人大腿小腿等长, 所以足尖和身体中心的连线与大腿的角度为 $\frac{1}{2}q_2$

• B_{5_safe} 对步行速度的约束, 约束要求 $\dot{p}_x > 0.1$ 。

$$B_{5_safe}(x) = \dot{p}_x - 0.1 \quad (4-14)$$

• B_{6_safe} 对躯干角度的约束, 约束要求 $r > 0.017$ 。

$$B_{6_safe}(x) = r - 0.17 \quad (4-15)$$

和学习关节限制的任务一样, 本任务中使用如式(4-8)所示的控制器, 用如图4-3所示的参考轨迹运行算法。实验使用的参数见表4-2, 其中在学习到第二十轮的时候, 我们发现再之后的学习得到的 CBF 变得过于严苛, 因此缩小了向前反推的时长 τ 和 τ' 。

4.4.2 实验结果和分析

使用上述实验设置学习 CBF, 学习 30 轮大致用时三天, 本次实验由于时间原因终止了算法。学习过程中算法迭代次数以及采样轨迹的平均时长的关系如图4-8, 可见在平均意义上, CBF 的引入使得很快就会摔倒的控制器能够坚持更长时间的平稳。在仿真环境中测试得到的 CBF, 机器人可以平稳步行, 效果如图4-7。图中所展现的为开始仿真后 5 秒到 20 秒之间的结果, 这段时间被图中

展示的几个状态等分。图4-9展示了稳定步态各个关节的相轨迹⁶，可以看出机器人的状态收敛到了一个稳定的周期性循环上。

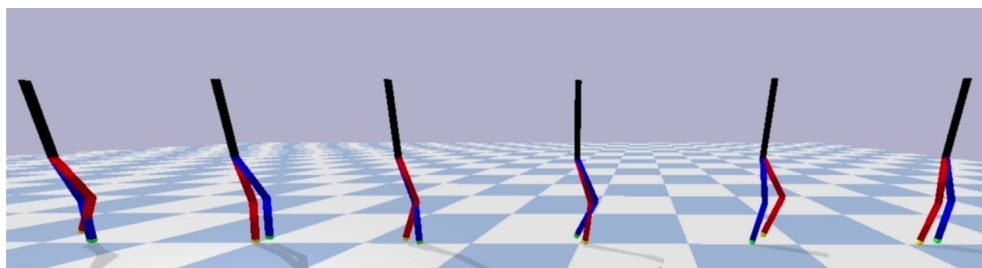


图 4-7 在 CBF 学习算法得到的 CBF 约束下机器步行仿真的结果，图中将几个时刻的机器人状态画在了同一张图中，图中所展现的为开始仿真后 5 秒到 20 秒之间的结果，这段时间被图中展示的几个状态等分。

4.5 本章小结

为了证明算法的有效性，我们首先在玩具问题中对于算法的框架进行了测试，使用基本基于定义的算法实现，配合上水平集估计方法来减少运行算法所需要的代价，我们学到了可以在二维平面避开障碍的 CBF。之后我们在机器人环境中学习了限制机器人关节极限的 CBF，验证我们提出的 CBF 学习机理。从状态空间的图中我们看到仿真轨迹结束状态，通过 prox 算子局部反推得到状态，已经采样算法得到的安全状态的相对位置关系。并且实验中使用我们习得

⁶这里展示的是支撑腿和摆动腿空间的相轨迹， (\hat{q}_1, \hat{q}_2) 为支撑腿的关节， (\hat{q}_3, \hat{q}_4) 为摆动腿的关节

表 4-2 学习 CBF 确保机器人稳定步行实验参数表

Parameter	Value
每轮样本轨迹采集数	100
训练轮数	30
τ (前二十轮)	0.01
τ' (前二十轮)	0.1
τ (二十轮之后)	0.003
τ' (二十轮之后)	0.02
γ	100
λ_ξ	100
λ_{du}	100

Trend of the Mean Length of Traj Samples

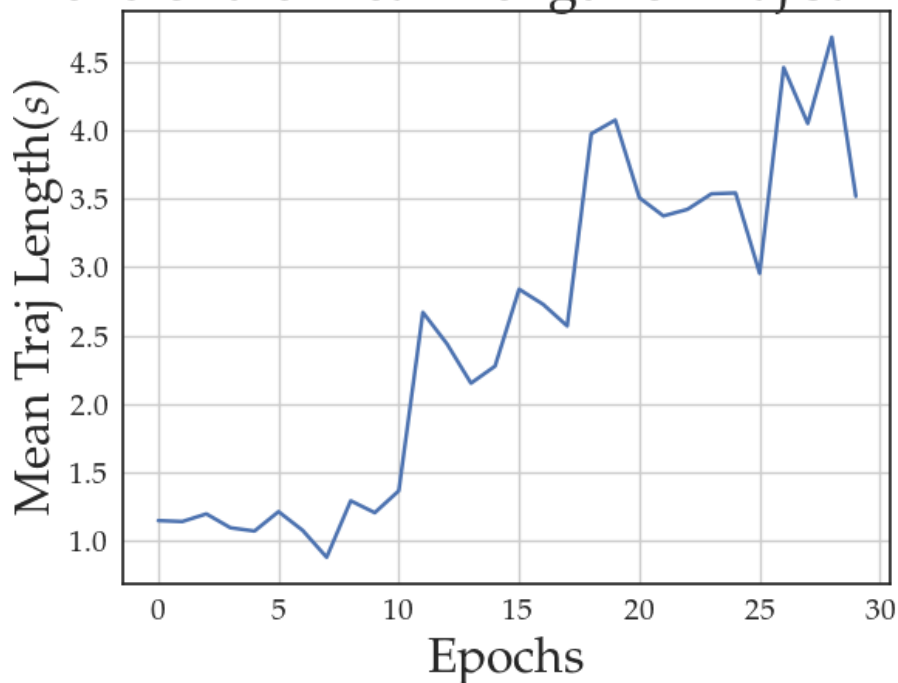


图 4-8 CBF 学习算法迭代次数以及采样轨迹的平均时长的关系，可以看出在 CBF 的作用下，轨迹的平均时长在稳步提升。

的 CBF 的控制器，可以确保系统不进入避免集合。最后我们使用 CBF 学习算法应用在步行任务上，得到的 CBF 极大地增加了系统对参考轨迹误差和控制器参数的鲁棒性，甚至可以与原本不稳定的参考轨迹和控制器参数共同作用，使机器人平稳步行。

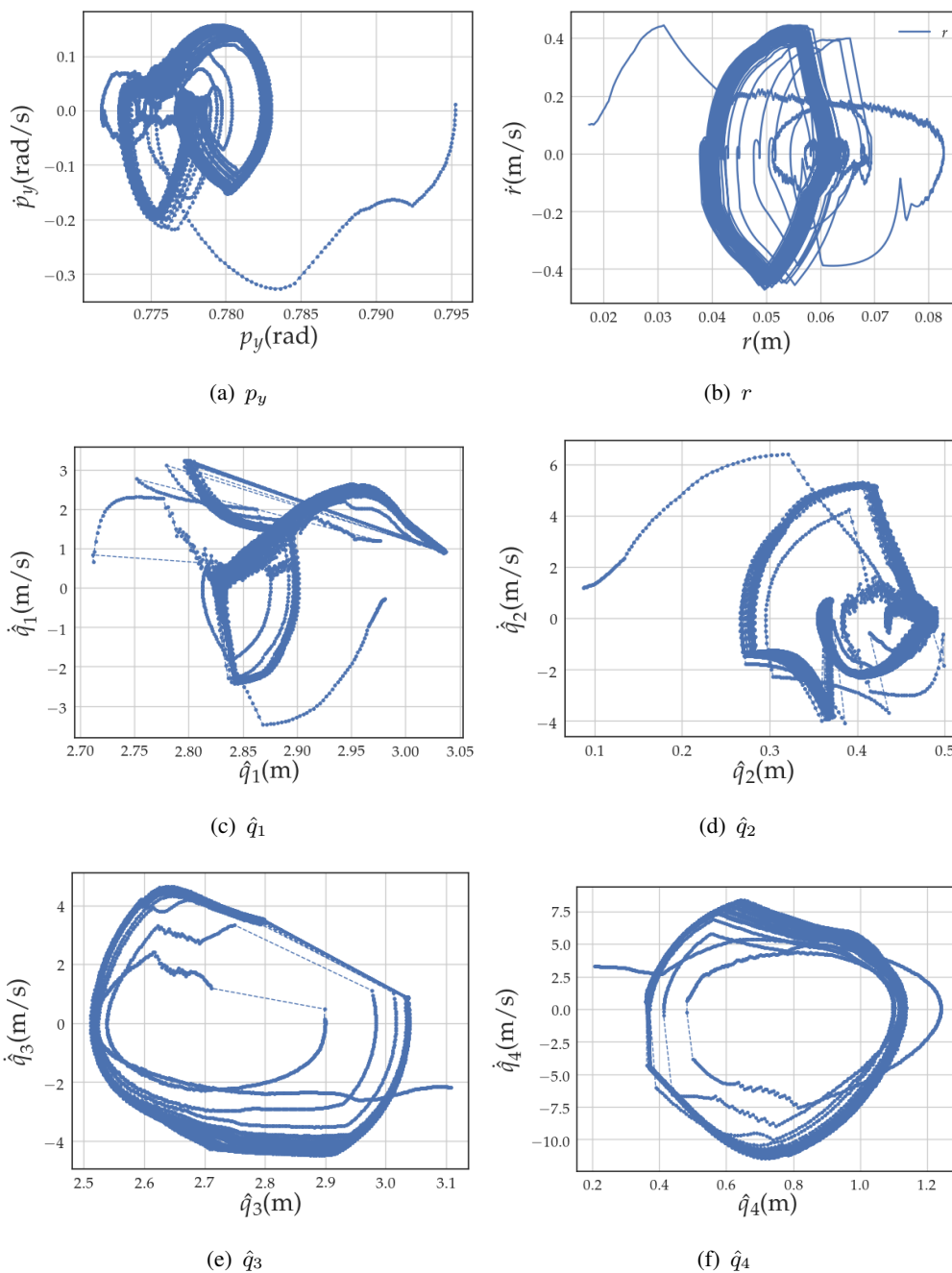


图 4-9 机器人行走前 20 秒各个自由度的相轨迹，图中不连续的点用很细的虚线相连，可以看出机器人的状态收敛到了一个稳定的周期性状态上

全文总结

本文提出了一套双足机器人背景下，基于大量仿真实验的 **CBF** 学习算法。

虽然近期很多使用 **CBF** 的控制器研究在需要安全保障的任务上取得了成功，但是科研人使用的 **CBF** 仍然是手工设计得到的。对于足式机器人一类的复杂欠驱动系统，手工设计 **CBF** 难以验证系统真正的安全性，难以应用到更加抽象的任务中去，例如防止机器人摔倒。

为了填补这一空缺，本文提出了可以自动化地得到避免系统状态进入“避免集合”的 **CBF** 函数的算法。算法主要有采样和拟合两个核心步骤，从给定的避免集合开始，迭代执行两个核心步骤，逐渐扩大确保安全的时间。采样即是通过大量仿真，找到仿真轨迹中最后无法满足 **CBF** 约束的状态，并且反推一定时间作为不安全的状态点。拟合即是使用一个类似 **SVM** 的算法，区分采样过程中得到的安全集合以及不安全集合。在机器人环境中，采样的反推过程是一个寻找离仿真轨迹结束状态最近的安全集合边界点的优化问题的求解过程。拟合过程中原本需要对于任意安全集合中的状态都满足 **CBF** 约束，在机器人环境中，简化到只用对正样本的约束满足即可。

为了证明算法的有效性，我们首先在玩具问题中对于算法的框架进行了测试，学到了可以在二维平面避开障碍的 **CBF**。之后我们在机器人环境中学习了限制机器人关节极限的 **CBF**，验证我们提出的 **CBF** 学习机理。最后我们使用 **CBF** 学习算法应用在步行任务上，得到的 **CBF** 极大地增加了系统对参考轨迹误差和控制器参数的鲁棒性，甚至可以与原本不稳定的参考轨迹和控制器参数共同作用，得到机器人稳定的步态。

符号说明

x 系统状态

n 系统状态维度

m 系统虚拟约束个数, 或 CBF 个数

u 系统输入

π 系统策略 (时间到输入的函数)

η 系统误差 (李雅普诺夫函数或 virtual constraint)

δ 松弛变量 (出现在2-9)

\mathcal{C}_{avoid} 避免集合 (见1.2)

$\mathcal{C}_{protect}$ 保护集合 (见1.2)

B CBF 函数 (见2.1式2-8)

\mathcal{B} CBF 函数 B 的集合 (见1.2)

\mathcal{C} 安全 (具有前向不变性的) 集合

γ CBF 条件中障碍函数的系数 (见2.1式(2-8))

$L_f B$ lie-derivative (见2.1式(2-6))

q 广义坐标 (generalized coordinate)

A_{rbd} 刚体动力学广义质量矩阵 (见2.2.1式(2-14))

g_{rbd} 刚体动力学广义惯性力 (见2.2.1式(2-14))

F 系统所受外力 (见2.2.1式(2-14))

J 雅克比矩阵 (见2.2.1式(2-15))

h_d 参考轨迹 (virtual-constraint) (见2.3.1式(2-26))

\mathcal{C}_{safe} 安全集合 (见定义3-1)

\mathcal{C}_{danger} 不安全集合 (见定义3-1)

\mathcal{C}_{T_safe} T -安全集合 (见定义3-2)

B_{T_safe} T -安全障碍函数 (见式3-6)

τ 采样算法危险集合时间跨度 (见3.1)

τ' 采样算法安全集合时间跨度 (见3.1)

$\lambda_\xi, \lambda_{du}$ 机器人背景下 CBF 拟合算法系数 (见式(3-17))

sample 采样算法 (见式3-6)

prox 机器人背景下反推采样算法 (见式(3-14))
 $\delta\tilde{\mathcal{C}}_{(T+\tau)_safe}$

p_x, p_y, r 机器人中心横纵坐标身体倾角 (见图4-2)

q_1, q_2, q_3, q_4 机器人关节角度 (左腿右腿) (见图4-2)

$\tilde{q}_1, \tilde{q}_2, \tilde{q}_3, \tilde{q}_4$ 机器人关节角度 (支撑腿, 摆动腿) (见图4-2)

θ 机器人步态时间信号 (见图4-2)

参考文献

- [1] SREENATH K, PARK H W, POULAKAKIS I, et al. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL[J]. International Journal of Robotics Research, 2011, 30(9):1170–1193.
- [2] HAUSER K, BRETL T, LATOMBE J C, et al. Motion planning for legged robots on varied terrain[J]. International Journal of Robotics Research, 2008, 27(11-12):1325–1349.
- [3] IQBAL S, ZANG X, ZHU Y, et al. Bifurcations and chaos in passive dynamic walking: A review[J]. Robotics and Autonomous Systems, 2014, 62(6):889–909.
<http://dx.doi.org/10.1016/j.robot.2014.01.006>.
- [4] DENG K, ZHAO M, XU W. Level-ground walking for a bipedal robot with a torso via hip series elastic actuators and its gait bifurcation control[J]. Robotics and Autonomous Systems, 2016, 79:58–71. <http://dx.doi.org/10.1016/j.robot.2016.01.013>.
- [5] XUE BIN P, GLEN B, KANGKANG Y, et al. DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning[J]. ACM Transactions on Graphics, 2017, 36(4). <http://www.cs.ubc.ca/~van/papers/2017-TOG-deepLoco/2017-TOG-deepLoco.pdf>.
- [6] HWANGBO J, LEE J, DOSOVITSKIY A, et al. Learning agile and dynamic motor skills for legged robots[J]. 2019. <http://robotics.sciencemag.org/>.

- [7] BOSWORTH W, KIM S, HOGAN N. The MIT Super Mini Cheetah : A small , low-cost quadrupedal robot for dynamic locomotion[J].
- [8] BLEDT G, POWELL M J, KATZ B, et al. MIT Cheetah 3 : Design and Control of a Robust , Dynamic Quadruped Robot[J]. 2018:2245–2252.
- [9] KIM D, JORGENSEN S J, LEE J, et al. Dynamic Locomotion For Passive-Ankle Biped Robots And Humanoids Using Whole-Body Locomotion Control[M].[S.l.]: [s.n.] , 2019.
- [10] KAJITA S, KANEHIRO F, KANEKO K, et al. The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation[J]. IEEE International Conference on Intelligent Robots and Systems, 2001, 1:239–240.
- [11] ZHANG R, ZHAO M, WANG C L. Standing Push Recovery Based on LIPM Dynamics Control for Biped Humanoid Robot[J]. 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018, 2018:1732–1737.
- [12] GOSWAMI A. Postural stability of biped robots and the foot-rotation indicator (FRI) point[J]. International Journal of Robotics Research, 1999, 18(6):523–533.
- [13] PRATT J, CARFF J, DRAKUNOV S, et al. Capture point: A step toward humanoid push recovery[J]. Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS, 2006:200–207.
- [14] ZHANG X, ZHAO M. Humanoid Robot Gait Planning Based on Virtual Supporting Point[J]. 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018, 2018:588–593.

- [15] CHEVALLEREAU C, ABBA G, AOUSTIN Y, et al. RABBIT: a testbed for advanced control theory[J]. IEEE Control Systems Magazine, 2003, 23(5):57–79.
- [16] KIM D, DI CARLO J, KATZ B, et al. Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control[J]. 2019. <http://arxiv.org/abs/1909.06586>.
- [17] AGRAWAL A, SREENATH K. Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation[J]. Robotics: Science and Systems, 2017, 13.
- [18] AMES A D, XU X, GRIZZLE J W, et al. Control barrier function based quadratic programs with application to Adaptive Cruise Control[J]. arXiv preprint arXiv:1609.06408, 2016(2):1–16.
- [19] HSU S C, XU X, AMES A D. Control barrier function based quadratic programs with application to bipedal robotic walking[J]. Proceedings of the American Control Conference, 2015, 2015-July:4542–4548.
- [20] BORRMANN U, WANG L, AMES A D, et al. Control Barrier Certificates for Safe Swarm Behavior[J]. IFAC-PapersOnLine, 2015, 48(27):68–73.
- [21] WANG L, AMES A D, EGERSTEDT M. Safety barrier certificates for collisions-free multirobot systems[J]. IEEE Transactions on Robotics, 2017, 33(3):661–674.
- [22] WU G, SREENATH K. Safety-critical control of a planar quadrotor[J]. Proceedings of the American Control Conference, 2016, 2016-July:2252–2258.
- [23] NGUYEN Q, SREENATH K. Exponential Control Barrier Functions for Enforcing High Relative-Degree Safety-Critical Constraints[R].[S.l.]: [s.n.] .

- [24] NGUYEN Q, HEREID A, GRIZZLE J W, et al. 3D dynamic walking on stepping stones with control barrier functions[J]. 2016 IEEE 55th Conference on Decision and Control, CDC 2016, 2016:827–834.
- [25] CHENG R, OROSZ G, MURRAY R M, et al. End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:3387–3395.
- [26] LI X, SERLIN Z, YANG G, et al. A formal methods approach to interpretable reinforcement learning for robotic planning[J]. Science Robotics, 2019, 4(37):1–16.
- [27] TAYLOR A, SINGLETARY A, YUE Y, et al. Learning for Safety-Critical Control with Control Barrier Functions[J]. 2019, xxx(1999):1–12. <http://arxiv.org/abs/1912.10099>.
- [28] YANG C, JIANG Y, LI Z, et al. Neural Control of Bimanual Robots with Guaranteed Global Stability and Motion Precision[J]. IEEE Transactions on Industrial Informatics, 2017, 13(3):1162–1171.
- [29] SICILIANO B, KHATIB O. Springer Handbook of Robotics[M]. Berlin, Heidelberg: Springer-Verlag, 2007.
- [30] WESTERVELT E R, GRIZZLE J W, CHEVALLEREAU C, et al. Feedback Control of Dynamic Bipedal Robot Locomotion[M]. New York: CRC Press, 2007.
- [31] CORTES C, VAPNIK V. Support-vector networks[J]. Machine learning, 1995, 20(3):273–297.

- [32] PARIKH N, BOYD S. Proximal Algorithms[J]. Found. Trends Optim., 2014, 1(3):127–239. <https://doi.org/10.1561/24000000003>.
- [33] GOTOVOS A, CASATI N, HITZ G, et al. Active learning for level set estimation[J]. IJCAI International Joint Conference on Artificial Intelligence, 2013:1344–1350.
- [34] COUMANS E, BAI Y. PyBullet, a Python module for physics simulation for games, robotics and machine learning[缺文献类型标志代码].
- [35] FEVREM. TROPIC[缺文献类型标志代码]. <https://github.com/fevrem/TROPIC>. [Online; accessed 27. May 2020].

致 谢

感谢上海交通大学！感谢 IEEE 试点班的培养，为我提供了按照计算机加自动化两个方向自由发展的机会。

首先十分感谢高岳老师的信任和指导，高老师的信任给了我自主发挥的空间，给了我选择这个颇有挑战的题目的勇气，给了我在为了这个课题，不断学习，实验中得到的收获和感悟，让我可以在大学结束时，用这个自我主导，拼尽自我所能的课题来画上圆满的句号。虽然疫情略微阻隔了交流，但高老师总能在我不计可施的时候一语点出我问题的症结，引导我找到解决思路。

其次感谢上海交大的其他老师，尤其是蒋力老师，傅洛伊老师。两位老师开启了我的科研之路，耐心引导我阅读文献，参加组会，以及开展实验，培养了我的实践能力和思辨能力。如果没有他们和我的各位任课老师的培养，我将无法完成本文。另外一提，本文绪论中所提到的“机器人是科学王冠上的明珠”便是蒋老师在一次组会时提到的，我一直念念不忘。

另外感谢伯克利的 Koushil Sreenath 老师，在我学期交换期间，虽然他的足式机器人控制课已经超过了教务处规定的容量，但是他仍然允许我旁听并全程参与他的课，本文中涉及足式机器人的大部分基础知识，包括接触到 CBF，均需要归功于他的课的启蒙。

还需要感谢我的好朋友清华的王嘉曦同学，借着疫情期间在家看网课的机会，他将正在上的清华凸优化课分享给我，并且和我热烈讨论，让我受益匪浅。本文提到的临近点算子便是来自他的分享。

最后，我要感谢我的父母，他们这段时间的支持和陪伴给了我十足的动力。感谢我的女朋友张雨迪在论文撰写过程中给予我陪伴和支持。



SYNTHESIS CONTROL BARRIER FUNCTIONS WITH LEARNING ALGORITHMS: WITH THE APPLICATIONS OF BIPEDAL SAFE WALKING

For a long time, the legged robot is an important field of engineering and scientific research. For its ability of navigation on difficult terrains than wheeled vehicles and its energy efficiency than aircraft. Various methods and techniques are proposed and progress is made about robot walking, running, and performing tasks. However, the safety of these systems is still a concern. Recently, Control Barrier Function(CBF) based control techniques have achieved success in several safety-critical applications. Using CBFs as constraints, a controller can be proved to ensure the forward invariance of a "safe set" in the state space. The forward invariance means, if the initial state of the system is inside of the safety set, then under the control of the controller, the state of the system at any time must still belong to this invariance set. The condition for a barrier function to have that property is

$$B(x) \geq 0 \quad \forall x \in \mathcal{C}_0 \quad (-16)$$

$$\sup_{u \in U} [L_f B(x) + L_g B(x)u + \gamma B(x)] \geq 0, \quad (-17)$$

Where B is the barrier function and x, u are the system state and input. γ is a positive coefficient and $L_f B$ and $L_g B$ are all lie derivatives. Thus, researchers can design CBFs that satisfies the definition. With the CBFs, she can put the CBF constraint in the

controller, The CBF constraint has the form:

$$\dot{B}(x, \mu) + \gamma B(x) \geq 0 \quad (-18)$$

Under the control with a controller with CBF constraint(which is always modeled as a QP problem), the system can be kept safely in the safety set. For example, with CBF representing the location of obstacles, unmanned Aerial Vehicles and robots can avoid obstacles. By using CBF to divide the position of people from the operational space of a robot arm, the human operator's safety can be guaranteed in man-machine cooperation. By using CBF to control the distance from the front car, the following system can also be guaranteed.

However, problems still exist when using CBFs in a scenario such as the legged robot, and the application potential of CBF has not been fully exploited. First, in a highly complex system such as a biped robot, for many tasks, there are no theoretical methods that can get CBFs which ensures forward invariance. Therefore, researchers need to tune and debug CBF(and the parameter gamma) manually, with difficulties finding the guarantee of forwarding invariance. Second, CBFs are only used in more intuitive tasks, such as obstacle avoidance and trajectory guidance. However, CBFs have more potentials than this, it can also be used to ensure forward invariance of more abstract collections in state space, taking avoid falling down as an example. The reason why the second goal is hard to achieve is that at present, researchers can only design CBFs by hand, and lack of an automatic way to analyze the dynamics and find CBF. In this paper, an automatic learning CBF algorithm based on a large number of experiments is proposed and applied to bipedal walking.

The paper proposes an algorithm which solves the following problem: given a set

of states that the user wants the system to avoid(called as avoid set), e.g. states of falling down, a set of states that the user wants to be kept in the safety set(called as protect set), and given a controller and environment where CBFs can be added and tested, the algorithm returns a set of CBFs that guarantees the system will not be developed to the avoid set, and that set contains the protect set. Specifically, the super-level-set of the resulted CBFs form a safety set, which ensures forward invariance under CBF constraints. The safety set has no intersection with the avoided set. So, if the system starts from the safety set and is controlled under the CBF constraints, its state will be kept in the safety set and never gets into the avoid set.

It is difficult to find the safety set analytically and directly in a highly complex system such as the legged robot. To tackle this difficulty, the algorithm is built upon the notion of T -safety set, which is defined as the set of states that can avoid entering the avoid set in a period of T . To decide whether a state is T -safety, the algorithm only needs to find an input policy of the period T , and verify that this policy keeps the state safe. The length of the ensurance of the T -safety set can be enlarged iteratively as the $T + \tau$ safety set can be calculated as the states that can, after a time of τ , stays in the T safety set. We then proved that when the T -safe set and the $T + \tau$ Safe set are equal, then these two sets have converged to the safe set, with a guarantee of forward invariance. With the additive property of T -safety set, the algorithm iteratively expands the safety ensurance period until the safety set converges.

The algorithm consists of two core steps: sampling and fitting. Starting from a given avoidance state set, it iteratively performs the two core steps to gradually extend the time of safety ensurance. The sampling step is to find out the states within the current safe set that cannot meet the CBF constraints through a large number of simulations. The failed states and states calculated from stepping back a period of time from



the failed state are added in the unsafe state sets. In the previously defined notion, the sampling step samples states from the $T + \tau$ danger set, from an optimization problem with horizon length of τ and the T -safety set. At the same time, the states that can ensure safety in a period are added in as safe state sets. The fitting step is to use an algorithm similar to SVM to distinguish the safe state set and the unsafe state set. Besides the SVM form, the fitting optimization problem has to include the CBF constraint. However, the “for any” condition in the CBF constraint is hard to be computationally verified. Thus, the condition is relaxed in the algorithm of the robot environment, and the drawback of the relaxation is remedied by the overall framework to ensure forward invariance.

In the scenario of bipedal walking, we adapted the aforementioned algorithm framework to tackle the complexities of the complex robot system. The CBFs are parameterized as polytopes, i.e. the combination of affine conditions. In each iteration of the algorithm, a new affine condition is added to further limit the safety set. In the sampling step, we first do a batch of simulations with the current CBFs. Because the current CBFs do not ensure forward invariance, there will be states that failed to find an input solution that satisfies the CBF constraint. These states are deemed as states on the boundary of the T -safe set. Then the stepping back process from the T -safe set to the $T + \tau$ safe set is modeled as an optimization problem to find the point closest to the failed state in the T -safety set which can ensure safety in a period of τ . Note, as τ is short, the dynamics of the robot is linearized at the failed state, and the input u in the period of τ is assumed to be a single constant value. Thus, the result of this optimization problem can be regarded as an approximation of a point in $T + \tau$ safety set. As for the fitting step, the original framework needs any states in the safety set to meet CBF constraints. Where in the robot environment, it is simplified to only satisfy the



constraints of positive samples and the states in the protected set. Besides, some extra penalty terms are added in the objective function to make the calculated CBF function do not make drastic changes for the behavior at the states of positive samples. The process is modeled as a non-convex problem which optimizes both the CBF parameter and the solution u to ensure the CBF constraint at each positive state samples.

In order to prove the effectiveness of the algorithm, we first tested the framework of the algorithm in the toy problem, and the resulted CBF can avoid obstacles in the two-dimensional plane. For the algorithm implementation of this experiment, we make use of a technique called super-level-set estimation which tries to find the fewest points to test and provides the most information about deciding the boundary of T -safe set. The experiment was successful in that the learned CBF can avoid collision no matter what the command of the controller is.

Then we learn the CBFs which ensure the robot joint limit in the robot environment, and verified the CBF learning mechanism that we proposed. The core of this task is to learn the relationship between the angle and angular velocity of the robot's capacity. That is, given the current joint angle, what is the largest velocity it can withstand in order to keep the joint angle greater than zero. In the experiment, we can see that the robot successfully learned the relationship and keeps the leg within the limit.

Finally, we apply the CBF learning algorithm to the walking task, and the obtained CBF greatly increases the robustness of the system against the reference trajectory error and controller parameters. The initial avoid set constraints about the joint angle limit, the torso limit, the body angle, and the forward velocity. And the robot's initial reference path can hardly make it walk for more than 5 seconds. The learning algorithm tries to get a set with forward invariance and belongs to the initial safe set. With the algorithm adding CBFs, the average walking length of the robot increased greatly.



At last, the learned CBFs can work with the original unstable reference trajectory and controller parameters to get the stable gait of the robot, which exhibited the potential usefulness of CBF techniques.