

电子科技大学
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE



论文题目 基于模仿学习的动态目标
抓取技术研究

专业学位类别 电子信息

学 号 202122100235

作者姓名 卢业康

指导教师 方黎勇 副教授

学 院 航空航天学院

分类号 TP242.6 密级 公开
UDC 注 1 681.5

学 位 论 文

基于模仿学习的动态目标抓取技术研究

(题名和副题名)

卢业康

(作者姓名)

指导教师	方黎勇	副教授
	电子科技大学	成 都
	黄奕勇	研究员
	军事科学院国防科技创新研究院	北 京

申请学位级别	硕士	专业学位类别	电子信息
专业学位领域	人工智能		
提交论文日期	2024 年 4 月 15 日	论文答辩日期	2024 年 5 月 30 日
学位授予单位和日期	电子科技大学	2024 年 6 月	
答辩委员会主席	李波		
评阅人			

注 1: 注明《国际十进分类法 UDC》的类号。

Research on Dynamic Object Grasping Technology Based on Imitation Learning

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline **Electronic Information**

Student ID **202122100235**

Author **Lu Yekang**

Supervisor **Associate Prof.Fang Liyong**

Supervisor **Prof.Huang Yiyong**

School **School of Aeronautics and Astronautic**

摘 要

在机器人领域，动态目标抓取一直是一项极具挑战性的任务，主要的难点集中在动态目标的飞行轨迹预测和机械臂的敏捷运动控制。现有的轨迹预测方法往往依赖于精准的物理模型来构造运动动力学方程或依靠庞大的数据集来训练深度神经网络模型，这些方法在预测效率、预测精度及泛化能力上均存在局限性。另一方面，现有的机械臂运动控制策略在处理复杂运动时，常因模型参数复杂和缺乏稳定性约束等导致轨迹复现精度低，无法保证运动收敛到目标位置的问题。本文针对上述问题，分别在动态目标轨迹预测方法和机械臂的运动控制算法上开展研究，主要工作如下。

首先，针对动态目标轨迹预测中的问题，本文提出了基于神经微分方程网络的动态目标轨迹预测方法。该方法的核心思想是通过神经微分方程来模拟动态目标的运动微分方程，以此捕获物体运动的内在物理规律。为了进一步提高预测的准确度，方法还集成了一个噪声估计网络，该网络负责分析历史预测中的误差，并据此优化未来轨迹的预测结果。实验结果表明，相比于传统时序预测网络模型，该模型的平均终点预测精度提高了 48.94%，数据效率提高了 6 倍。

其次，针对当前机械臂运动控制中遇到的问题，本文提出了一种基于条件变分自编码器与输入凸神经网络构建的稳定变分动态系统（SVDS），通过结合条件变分自编码器的潜在空间连续分布和输入凸神经网络凸性质的优势，保证了机械臂运动轨迹的连续平滑及稳定收敛。在实验中从仿真模拟和实物验证两个层面对本文提出的 SVDS 运动控制算法进行了验证，实验结果表明该方法在保证系统运动连续性和系统稳定性上具有出色的性能。

最后，为了验证本文工作的有效性，设计并实现了一个动态目标抓取原型系统并在真实的物理环境中对抛掷动态物体进行了实验测试。在实验环节中，分别进行了动态目标的接取与抓取实验，其中利用网兜进行动态目标接取的成功率达到了 75%，也成功实现了灵巧手高精度抓取动态目标实验验证。实验结果不仅验证了算法的有效性，而且展现了机器人在动态目标抓取领域应用的潜力。

本文创新性地引入了神经微分方程网络和条件变分自编码器与输入凸神经网络相结合的方法，分别解决了动态目标轨迹预测精度不足和机械臂运动控制稳定性的问题，成功实现了非规则抛掷物体的高精度抓取实验，也为未来在智能制造、自动化物流和服务机器人等领域的应用提供了广阔的前景。

关键词：动态目标抓取，神经微分方程，模仿学习，稳定变分动态系统

ABSTRACT

In the field of robotics, dynamic target grasping has always been a highly challenging task, with the main difficulties centered on the prediction of dynamic targets' flight trajectories and the agile motion control of robotic arms. Existing trajectory prediction methods often rely on precise physical models to construct motion dynamics equations or depend on large datasets to train deep neural network models, both of which have limitations in prediction efficiency, accuracy, and generalization capability. On the other hand, current robotic arm motion control strategies often face issues with trajectory reproduction accuracy due to complex model parameters and a lack of stability constraints, failing to ensure motion convergence to the target location. This paper addresses these issues by researching dynamic target trajectory prediction methods and motion control algorithms for robotic arms, with the main contributions as follows.

Firstly, for dynamic target trajectory prediction, this paper proposes a method based on Neural Ordinary Differential Equation Networks. The core idea is to simulate the motion differential equations of dynamic targets using neural ordinary differential equations, thereby capturing the intrinsic physical laws of object motion. To further improve prediction accuracy, the method integrates a noise estimation network responsible for analyzing errors in historical predictions and optimizing future trajectory predictions accordingly. Experimental results show that compared to traditional time-series prediction network models, this model improves the average endpoint prediction accuracy by 48.94% and increases data efficiency by sixfold.

Secondly, to address issues in current robotic arm motion control, this paper presents a Stable Variational Dynamic System (SVDS) constructed based on Conditional Variational Autoencoders and Input-Convex Neural Networks. By leveraging the advantages of the latent space continuous distribution of Conditional Variational Autoencoders and the convexity of Input-Convex Neural Networks, the system ensures continuous, smooth motion trajectories and stable convergence of the robotic arm. The SVDS motion control algorithm was validated through simulation and physical experiments, demonstrating outstanding performance in ensuring system motion continuity and stability.

Lastly, to verify the effectiveness of the work, a dynamic target grasping prototype system was designed and implemented for experimental testing in a real physical environment. The experiments involved dynamic target catching and grasping, where a success rate of 75% was achieved using a net for dynamic target catching, and high-precision grasping of dynamic targets was successfully demonstrated through dexterous hand experiments. The experimental results not only validated the effectiveness of the algorithms but also showcased the potential of robots in the application of dynamic target grasping.

This paper innovatively introduces neural ordinary differential equation networks and combines conditional variational autoencoders with input convex neural networks to address the challenges of accuracy in dynamic target trajectory prediction and stability in robotic arm motion control. These methods have successfully achieved high-precision grasping of irregularly thrown objects and offer promising prospects for future applications in intelligent manufacturing, automated logistics, and service robotics.

Keywords: Dynamic Object Grasping, Neural Ordinary Differential Equations, Imitation Learning, Stable Variational Dynamic Systems

目 录

第一章 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.2.1 动态目标抓取技术.....	2
1.2.2 动态目标轨迹预测技术.....	3
1.2.3 机械臂敏捷运动控制技术.....	5
1.3 研究目标与内容.....	8
1.4 论文结构安排.....	10
第二章 数据采集及预处理	12
2.1 引言.....	12
2.2 动态目标轨迹数据采集.....	12
2.2.1 轨迹数据采集.....	12
2.2.2 轨迹数据预处理.....	16
2.3 机械臂运动示教数据采集	20
2.3.1 示教数据采集.....	20
2.3.2 示教数据预处理.....	22
2.4 本章小结.....	23
第三章 基于深度神经网络的动态目标轨迹预测	24
3.1 引言.....	24
3.2 动态目标运动分析.....	24
3.2.1 动力学模型分析.....	24
3.2.2 运动轨迹预测问题分析.....	25
3.3 动态目标轨迹预测方法分析	26
3.3.1 牛顿动力学.....	26
3.3.2 最小二乘法拟合	28
3.3.3 ResLSTM 神经网络	28
3.4 基于神经常微分方程网络的动态目标轨迹预测	29
3.4.1 神经常微分方程网络.....	30
3.4.2 模型框架.....	33
3.4.3 估计模块.....	34
3.4.4 补偿模块.....	35
3.5 实验及分析.....	37
3.5.1 数据集构建.....	37
3.5.2 评价指标.....	38

3.5.3 网络训练.....	39
3.5.4 结果分析.....	41
3.6 本章小结.....	44
第四章 基于模仿学习的机械臂敏捷运动控制	45
4.1 引言.....	45
4.2 模仿学习.....	45
4.3 动态系统理论分析与系统构建	47
4.3.1 动态系统理论介绍.....	47
4.3.2 动态系统稳定性分析.....	47
4.3.3 动态系统构建.....	49
4.4 稳定变分动态系统.....	51
4.4.1 稳定变分动态系统结构.....	51
4.4.2 稳定变分动态系统运动控制流程.....	51
4.5 基于条件变分自编码器的动态系统估计项	52
4.6 基于输入凸神经网络的动态系统稳定项	54
4.6.1 输入凸神经网络.....	54
4.6.2 李雅普诺夫函数设计.....	55
4.6.3 稳定性证明.....	56
4.7 实验及分析.....	57
4.7.1 实验设计.....	57
4.7.2 仿真实验.....	58
4.7.3 实物实验.....	64
4.7.4 结果分析.....	67
4.8 本章小结.....	67
第五章 动态目标抓取原型系统设计与实现	68
5.1 引言.....	68
5.2 系统流程.....	68
5.3 硬件系统.....	69
5.3.1 机械臂.....	69
5.3.2 灵巧手.....	70
5.3.3 控制主机.....	71
5.4 软件系统.....	72
5.4.1 软件平台.....	72
5.4.2 软件框架.....	72
5.5 实验及分析.....	73
5.5.1 实验设计.....	73
5.5.2 实验过程.....	74
5.5.3 实验分析.....	77

5.6 本章小结.....	78
第六章 总结与展望	79
6.1 总结.....	79
6.2 展望.....	80
参考文献.....	81
附录 A 手写数据集实验对比图.....	86

图目录

图 1-1 机器人应用场景。(a)协作机器人；(b)手术机器人；(c)物流机器人；(d)工业机器人；	1
图 1-2 KUKA LWR 4+机械臂抓取动态目标	3
图 1-3 本论文的主要研究内容	9
图 1-4 本文章节安排与研究内容	11
图 2-1 数据采集系统。(a)Vicon 红外相机；(b)标定杆；(c)Poe 交换机；(d)标记点； (e)Track 软件；	13
图 2-2 数据采集区域	14
图 2-3 数据采集场景	14
图 2-4 研究对象物体	15
图 2-5 原始轨迹数据	15
图 2-6 预处理后的轨迹数据	18
图 2-7 六类非规则物体	19
图 2-8 预处理后非规则物体轨迹数据	20
图 2-9 示教设备	21
图 2-10 原始示教数据。(a)示教轨迹 1；(b)示教轨迹 2；	21
图 2-11 相对示教轨迹。(a)示教轨迹 1；(b)示教轨迹 2；	23
图 3-1 飞行物体受力情况	24
图 3-2 ResLSTM 网络结构	29
图 3-3 残差网络-神经常微分方程隐藏层对比 ^[45]	31
图 3-4 循环神经网络-神经常微分方程螺旋线重建外推对比 ^[45] 。(a)循环神经网络重建外推效果；(b)神经常微分方程网络重建外推效果；	31
图 3-5 轨迹预测模型框架	33
图 3-6 估计模块结构	34
图 3-7 NoiseNet 网络结构	36
图 3-8 数据划分流程	38
图 3-9 训练损失曲线	40
图 3-10 平均损失曲线	40
图 3-11 领先时间对比	43
图 4-1 示教数据采集。示教过程 1；示教过程 2；	46

图 4-2 稳定变分动态系统 (SVDS) 结构	51
图 4-3 稳定变分动态系统运动控制流程	52
图 4-4 条件变分自编码器(CVAE)结构	53
图 4-5 输入凸神经网络结构	54
图 4-6 平滑 ReLU 激活函数	56
图 4-7 LASA 手写数据集模型损失曲线	59
图 4-8 LASA 复现轨迹	59
图 4-9 LASA 李雅普诺夫函数能量场	60
图 4-10 LASA 速度变化曲线	60
图 4-11 CLF-DM 复现轨迹	61
图 4-12 六维机械臂运动模型损失曲线	62
图 4-13 机械臂复现位置轨迹。(a)三维位置轨迹; (b)X-Y 二维位置轨迹; (c) X-Z 二维位置轨迹;	63
图 4-14 机械臂复现姿态轨迹。(a)三维姿态轨迹; (b)RX-RY 二维姿态轨迹; (c) RX-RZ 二维位置轨迹;	63
图 4-15 机械臂仿真实验	64
图 4-16 静态抓取实验 1	65
图 4-17 静态抓取实验 2	65
图 4-18 动态跟踪实验	66
图 4-19 动态跟踪对接实验	66
图 5-1 动态目标抓取原型系统流程图	68
图 5-2 硬件系统组成与连接	69
图 5-3 UR5e 协作机器人	70
图 5-4 Allegro Hand 灵巧手	70
图 5-5 控制主机	71
图 5-6 动态目标抓取原型系统软件框架	72
图 5-7 运动物体接取 1	74
图 5-8 运动物体接取 2	74
图 5-9 运动物体抓取 1	75
图 5-10 运动物体抓取 2	75
图 5-11 失败案例 1	76
图 5-12 失败案例 2	76

表目录

表 2-1 运动物体初始速度（平均值±标准差）	16
表 3-1 模型超参数设置	39
表 3-2 平均位移误差对比	41
表 3-3 终点平均位移误差对比	41
表 3-4 平均领先时间对比	42
表 3-5 模型执行时间	42
表 4-1 LASA SVDS 超参数	58
表 4-2 模型执行时间	61
表 4-3 UR5e SVDS 超参数	62
表 5-1 UR5e 硬件参数	70
表 5-2 Allegro Hand 硬件参数	71
表 5-3 主机硬件参数	71
表 5-4 实验成功率统计	77

第一章 绪论

1.1 研究背景与意义

随着科技的不断进步，协作机器人已经成为人类社会不可或缺的一部分，它们在工业自动化、家庭服务、自动驾驶汽车以及军事防御等关键领域扮演着越来越重要的角色，机器人的应用场景如图 1-1 所示。这些机器人不仅极大地提高了人们的工作效率，还为人们的日常生活带来了前所未有的便利。特别是在制造行业，人机共享操作空间的协同作业模式有效地提升了生产效率，而在服务行业，机器人的逐渐应用也标志着人机互动进入了一个新的时代。然而，尽管机器人已经在多个领域得到了广泛的应用，但是机器人在灵活操作和智能决策方面的能力仍然远远落后于人类，特别是在高度复杂和动态变化的环境中，机器人的自主抓取能力成为了衡量其智能化水平的重要标准。而动态目标抓取任务作为一项机器人研究领域的综合系统工程不仅对机器人的感知、决策和执行能力提出了更高的要求，也给现有的机器人技术和算法带来了前所未有的挑战。

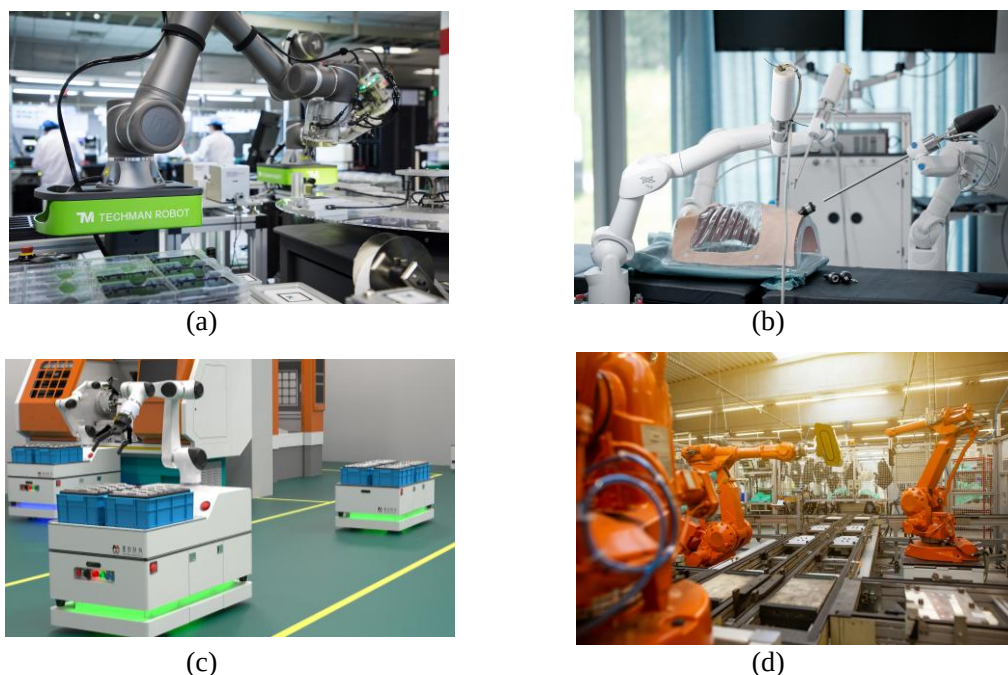


图 1-1 机器人应用场景。(a)协作机器人；(b)手术机器人；(c)物流机器人；
(d)工业机器人；

解决动态目标的实时轨迹预测和机器人的敏捷运动控制问题，不仅可以优化机器人在制造和物流等领域的应用效率，提升生产力，还可以在服务行业和日常

生活中拓展机器人的应用范围，增强其与人互动的能力。通过实现这些研究目标，本研究不仅有望在理论上为机器人的动态目标抓取问题提供新的解决方案，也将实践中推动机器人技术在制造、服务、军事等关键领域的应用和发展。

1.2 国内外研究现状

1.2.1 动态目标抓取技术

动态目标抓取是机器人学和自动化领域的一项复杂的系统工程，要求机器人系统具有高度的敏捷性和对动态目标变化的快速适应能力。这项任务可以分为三个关键的步骤：首先，利用测量数据对动态目标进行高精度的轨迹预测。其次，根据预测的轨迹、当前机器人的姿态和可达空间，计算最佳的捕捉姿态。最后，控制机器人进行快速运动拦截。近年来，多项研究聚焦于轨迹预测算法与运动控制算法以提高捕捉过程的效率和准确性。

Dannemiller 等^[46]在其研究中对人们利用视觉信息来预测飞行球的落点并成功捕捉到球这一现象进行研究，发现人类能够灵活地捕捉到飞行的球体的一个关键因素是其能准确估计球的飞行路径。这项早期的研究不仅对于理解人类的视觉和运动协调机制具有重要意义，也为后来的机器人运动捕捉和人工智能系统的设计提供了宝贵的参考。

之后，Kober 等^[58]提出了运动模板的概念，通过采用机器学习中的监督学习和强化学习方法来构建运动模型。研究表明机器人可以通过观察人类专家的动作或与环境的交互来优化运动模板的参数，经过学习的模型能够执行多种简单任务，且具有一定的适应性和泛化能力。在之后的研究^[59]中 Kober 搭建了一个机器人杂耍系统，通过外部摄像头来定位球的位置，并使用卡尔曼滤波器来预测球的到达位置和到达时间，然后根据目标位置控制机器人用手接住球并抛给其搭档，实现了机器人的抛接球任务。

在对非规则物体的抓取研究中，Kim 等^[12]将抓取任务分解为三个部分，分别通过编程示范方法学习物体动力学模型来进行物体轨迹预测，通过概率方法来寻找可行的抓取配置和通过模仿学习构建动态系统控制机械臂运动。该系统成功在真实世界实验中使用 KUKA LWR 4+机械臂成功抓取了抛出的锤子、网球拍、空瓶、半瓶水和纸盒等物体，如图 1-2 所示。这项研究的成果也对之后的机器人技术领域产生了深远的影响。

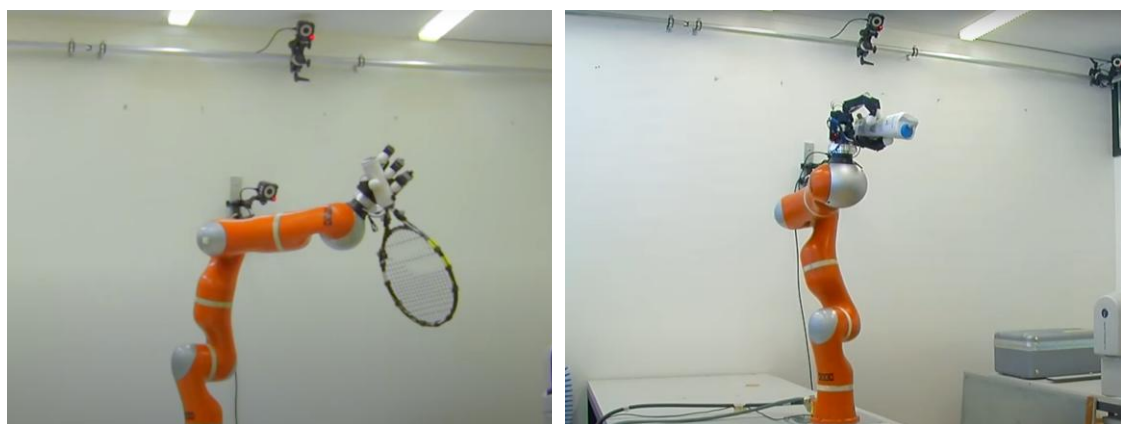


图 1-2 KUKA LWR 4+机械臂抓取动态目标

虽然 Kim 的捕捉实验中取得了成功，但是仍有不少失败案例是由于与机械手发生碰撞导致脱离掉落。Salehian 等^[60]针对这一问题提出了一种柔性捕抓策略。通过让机器人暂时与物体一起移动，让手指有更多时间调整以合适的姿态靠近物体，避免了“硬捕捉”。该方法同样是基于动态系统来控制跟随运动，提升了抓取任务的成功率。

随着研究的深入，人们逐渐意识到，要提高机器人对动态目标的抓取能力，需要从多个维度进行技术创新和突破。从提升感知预测精度、到优化动作规划和执行效率，每一个步骤都是提升整体抓取性能的关键。在此背景下，跨学科的研究方法开始被广泛采用，机器学习、深度学习、控制理论等领域的最新研究成果被逐步引入以解决这一复杂问题。

1.2.2 动态目标轨迹预测技术

动态目标的轨迹预测是完成抓取任务的关键环节，现有的轨迹预测方法主要可以分为两种。一种是基于传统运动学模型的方法，这类方法依靠已建立的物理和数学模型来预测物体的未来位置。第二种方法是基于神经网络的轨迹预测。这种方法利用了深度学习技术，通过训练神经网络来理解和预测物体的运动模式。

传统的轨迹预测方法往往依赖于已建立的物理和数学模型。这些方法通常通过物体过去的运动状态（物体的速度、加速度以及其他动力学参数）来预测未来的轨迹。一些研究^[1-4]忽略空气阻力和其他力对球的作用，将飞行球的轨迹建模为抛物线，然后通过最小二乘优化递归估计球的轨迹。

一些工作^[5-8]也考虑了空气阻力等其他非线性力的影响，采用抛物线形式对球的轨迹进行建模，并结合了扩展卡尔曼滤波器进行在线重新估计，得到了有效的估计结果。

一些基于物理学建模方法^[9,10]在对非规则物体进行轨迹预测前需要采集物体的先验信息,例如质量、质心位置、转动惯量和形状,然后再构建对应的运动学模型用于轨迹预测。然而,想要对形状非规则的物体估计质心位置、转动惯量等尤其困难。此外,基于物理学建模的预测方法的预测精度高度依赖于所构建模型的准确度,如果模型假设与物体的真实运动状态存在差异,例如未能考虑到某些关键的物理效应或环境因素,那么预测的轨迹就可能产生较大的误差。

一些研究^[12-14]采用传统机器学习技术进行离线学习估计运动物体的动力学模型,这些方法在预测物体的平移和旋转方面取得了显著成效。**Kim**等^[11]在其对非规则自由飞行物体动力学的研究中比较了六种非线性回归方法(基于径向基核函数的支持向量回归(SVR)、基于多项式核的支持向量回归(SVR)、高斯混合回归(GMR)、回波状态网络(ESN)、遗传规划(GP)和局部加权投影回归(LWPR))在召回精度、计算成本和对超参数选择的敏感性方面的性能。

尽管这些传统机器学习方法在特定条件下表现出色,但是随着样本量的增加,模型也会变得更加复杂。这种参数膨胀不仅增加了模型训练的计算负担,而且可能导致模型在面对未见过的物体时泛化能力下降。面对传统机器学习方法所存在的局限性,深度学习技术的引入提供了一种新的解决途径。这些深度网络通过层次化的数据表示和特征提取能力,能够有效地从复杂数据中学习抽象和通用的表示,从而提高了对未见过物体的泛化能力。

早在1995年,**Payeur**等^[15]就已经通过训练多个人工神经网络(Artificial Neural Networks, ANN)来估计运动对象的位置、速度和加速度,并在二维和三维空间中进行了轨迹预测仿真。实验证明了神经网络能够用于轨迹预测任务。

随着深度网络的研究深入,研究者针对时序数据的特点提出了循环神经网络(RNN)^[16]和对网络长时间依赖性进行改进后的长短期记忆网络(LSTM)^[17]。**Shi**等^[18]对长短期记忆(LSTM)神经网络在预测飞行轨迹中的应用进行了研究。通过在模型中应用滑动窗口技术来保持序列的连续性并提高预测精度。**陈荣**等^[19]通过将长短期记忆(LSTM)和混合密度网络(MDN)集成构建了双向循环混合密度网络(BiRMDN)的预测模型,显著提高了预测的精度。

张正等^[21]提出了一种基于小波变换的创新框架来对飞行模式进行时频分析以支持轨迹预测,并开发了编码器-解码器神经架构来估计小波分量,重点关注全局飞行趋势和局部运动细节的有效建模,成功实现对飞机飞行轨迹进行预测。

深度神经网络在乒乓球的运动预测中也得到了广泛的应用,**Gonzalez**等^[22,23]通过安装在天花板上的4个RGB摄像机观察乒乓球机器人投掷的乒乓球运动,并记录飞行轨迹,使用变分自动编码器深度神经网络对乒乓球轨迹进行预测。**王一**

雷等^[24]提出一种基于 LSTM 的轨迹预测算法,在获得图像中乒乓球的位置后,通过基于傅里叶变换的速度测量方法和基于 CNN 的旋转方向检测方法分别计算乒乓球的旋转方向和速度,实现对乒乓球的跟踪预测。

面对非规则物体的轨迹预测,雷校东^[25]在其研究中考虑到了物体运动过程中时间和空间上的依赖性,这是传统预测方法往往忽视的重要因素。通过引入注意力机制来分析不同历史时刻的轨迹信息对未来时刻轨迹的影响,建立了融合时空注意力机制的 LSTM 运动物体姿态与位置预测模型,实现了对非规则物体的有效预测,显著提高了预测性能。

Qadeer 等^[26]通过编码器-解码器双向 LSTM 深度神经网络来对工厂中的非规则物体进行飞行轨迹预测。该模型通过摄像头实时观察投掷物体的初始飞行信息来预测投掷物体未来飞行轨迹,在仿真实验中取得了成功。

Yu 等^[44]通过使用神经加速度估计器(NAE)来处理系统的非线性。该估计器使用 LSTM 网络通过观察物体轨迹的一小部分来帮助估计随时间变化的加速度。通过结合可微卡尔曼滤波器端到端地对模型进行训练,提高了预测的精度。

综上所述,复杂的物体其运动过程往往同时伴随着位置的移动及姿态的旋转变化,而大部分关于运动物体轨迹预测的研究往往只考虑物体在运动时质心的位置坐标变化情况,将运动物体等价为一个质点,基于物理模型与机器学习方法在这些情况下精度表现不足。而深度学习方法利用强大的模型拟合能力,能够学习到时序数据中的特征,模型预测精度得到了一定程度的提升,但是由于其对数据的依赖性,导致模型的效果很大程度上却决于数据集的质量。此外,对于使用者来说深度模型更像是一个黑盒子,难以解释。

1.2.3 机械臂敏捷运动控制技术

模仿学习(Imitation Learning, IL),又被称为示教学习(Learning from Demonstration, LFD)。基于模仿学习的机器人运动控制主要分为两种方法:基于时间序列的概率模型和基于状态反馈的动态系统模型。基于时间序列的概率模型方法侧重于分析和理解任务执行过程中的时间序列数据。通过建立概率模型来捕捉人类动作的统计特征用于预测接下来的动作序列。这种方法往往利用隐马尔可夫模型(HMM)、高斯混合模型(GMM)等概率工具来分析时间序列数据生成机器人的动作序列。与基于时间的概率模型不同,基于状态反馈的动态系统模型侧重于从状态空间的角度理解运动控制。基于状态反馈的动态系统模型通过建立一个从当前状态到下一个状态的映射来模拟人类的运动控制策略,而不是直接模仿动作序列。这使得机器人能够在执行任务时更灵活地适应环境变化和任务需求

的变化。动态系统模型通常使用非线性回归、神经网络等技术来构建，能够有效地处理高维数据和复杂的运动控制任务。

最早是 Schaal^[20]提出了模仿学习这一概念，表明机器人可以通过观察和模仿人类行为来学习复杂的运动和技能。研究中探索了不同的模仿学习策略，强调了模仿学习在机器人学中的潜力，尤其是在提高机器人自主性和适应性方面的作用。

Ijspeert 等人^[27]介绍了一种基于吸引子系统的运动学习方法，即动态运动基元 (Dynamical movement primitives, DMP)。通过将运动分解为可学习的动态系统，DMP 提供了一种有效的方式来模仿和泛化人类运动，包括点对点的运动和周期性运动。但 DMP 对时间的依赖性导致其对时间扰动较为敏感，在稳定性和泛化能力上存在着一定限制。

Paraschos 等人^[28]基于 DMP 提出了一种新的运动学习框架——概率运动基元 (Probabilistic Movement Primitives, ProMP)。ProMP 通过对运动轨迹的概率建模，使得机器人能够学习动作的变化性和不确定性，从而更好地适应新的情境和任务。研究中展示了 ProMP 在处理不同运动任务（抓取和投掷）时的灵活性和高效率，突出了其在机器人学习和适应性方面的潜力，但对于超出示教区域的情况，其泛化能力仍受限。

Huang 等人^[29]提出了一种新型的学习框架——核化运动基元 (Kernelized movement primitives, KMP)，通过引入核方法来优化动态运动基元的学习过程。KMP 通过核化技术提高了学习模型的表达能力，允许机器人更加精确地模仿复杂运动，弥补了 DMP 和 ProMP 的不足。该方法特别适用于需要高精度和适应性的机器人应用，如精细操作和人机协作。

Calinon 等人^[30]介绍了一种任务参数化的概率学习模型。该方法首先使用主成分分析 (PCA) 将运动数据投影到通用潜在空间上，进行基于概率的相关性估计，然后使用高斯混合模型或伯努利混合模型对所得信号进行编码。此外，Calinon 还提出了任务参数化高斯混合模型^[31] (Task-Parameterized Gaussian Mixture Model, TP-GMM)。该模型以虚拟弹簧-阻尼系统的形式编码动作，利用多个候选坐标系中的演示学习吸引子路径，通过预测协方差来生成新情况下的吸引子路径，并利用这些预测协方差来估计弹簧-阻尼系统的变化刚度和阻尼。该方法强调了减少对机器人行为的直接干预，从而提高了机器人自主性和适应性。然而，采用 GMM、GPR 等方法生成的轨迹无法保证轨迹稳定收敛至目标点。

瑞士洛桑联邦理工的 Khansari-Zadeh 和 Billard^[36]为了解决 DMP 中的时间相关性问题，提出了动态系统稳定估计器 (Stable Estimator of Dynamical Systems, SEDS) 方法。SEDS 方法通过 GMM 将运动轨迹建模为一阶自治系统（即时不变系

统), 并推导了确保目标全局渐进稳定性的充分条件, 使得机器人能够抵抗运行过程中的时间干扰, 在执行过程中具有可靠稳定性。SEDS 采用了二次型的 Lyapunov 函数来确保系统的稳定性, 对于每条轨迹, 即使存在空间或时间的扰动, 轨迹也能收敛至目标点。这种选择虽然简化了问题的处理, 但二次型 Lyapunov 函数可能过于严格, 无法精确表征示教轨迹。SEDS 牺牲了部分表征精度为代价, 保证了全局渐进稳定性, 被称为动态系统学习中精度和稳定性的两难困境。

为改进 SEDS, 众多学者开始为构建全局稳定且可表征复杂轨迹的 DS 展开研究, 一般可分为以下两步: 1) 基于示教数据灵活构建 Lyapunov 函数; 2) 基于 Lyapunov 稳定性约束求解 DS。

Khansari-Zadeh 和 Billard 从以上的两步对 SEDS 进行了改进, 提出了 SEDSII^[37]。该方法先使用示教数据来学习一个控制李亚普诺夫函数, 然后使用回归方法对运动进行建模, 最后通过解决一个凸优化问题来求解稳定项以确保动态系统的稳定性。此外, 文中还强调了该方法对回归模型选择的适应性, 可以根据任务的类型自由选择合适的回归模型。对于更复杂的运动场景来说, 该模型需要手动去选择合适的回归模型与优化参数。

Neumann 等人^[38]则通过在 DS 与 Lyapunov 函数之间构建一个微分同胚映射 (Diffeomorphic Mapping, DM), 将示教数据映射到隐空间, 在隐空间中利用 SEDS 学习全局稳定的 DS, 再用其逆映射生成原空间的 DS。其对示教数据的表征效果更好, 可表征更复杂的运动。

高霄^[32]则在其研究中通过建立示教数据空间和隐空间之间的微分同胚映射, 并在隐空间中构建一个全局渐近稳定的动态系统, 并设计了 Lyapunov 函数证明了其稳定性, 从而得到示教空间中全局渐进稳定且能精确表征示教数据的动态系统。该方法在保证系统稳定性的同时, 很大程度上克服了模型的精度问题。但是增加了模型复杂度, 导致学习更加耗时, 仅限于离线学习中使用。

在机器人模仿学习领域, 越来越多的研究将轨迹编码为动态系统(DS)以实现精确稳定的运动控制。在现有的学习框架中, 基于神经网络与动态系统策略表现出了相较于传统算法的极大优越性^[33-35]。Urain^[39]提出一种基于标准化流生成稳定的 DS: 模仿流(imitation Flow, iFlow)。使用标准流可以将复杂、非规则的数据分布转换为更简单的数据分布, 从而使学习过程更易于管理, 并且生成的动态系统更可预测和稳定, 提高了模型表达精度。

Rana 等^[40]提出了一种称为“欧几里得流”的方法, 通过微分同胚来学习稳定的动力系统。这项技术本质上简化了数据空间的几何形状, 使其成为欧几里德空

间，从而简化了稳定动态系统的学习。通过采用微分同胚变换，该方法不仅确保学习到的动态系统稳定性，而且可以轻松推广到新场景，具有良好的泛化性。

Dattari 等^[41]通过将模仿学习与对比学习相结合，扩展了学习稳定动态系统的概念，引入了一种新的对比学习损失来训练深度神经网络（DNN），当与模仿学习损失相结合时，可以增强学习运动的全局渐近稳定性。该方法的突出之处在于不限制其函数逼近器的结构，可以使用任意 DNN，并有助于高精度地学习复杂运动。

综上所述，在机械臂的运动控制方面还存在的许多的问题，一是敏捷性与适应性：大部分的传统规划方法设计用于特定场景的离线规划，这限制了它们在实时变化的任务环境中的应用。模仿学习方法通过简化机器人学习流程，提高了机器人的易用性和效率。然而，基于时间序列的概率模型（DMP，ProMP 等）采用开环方式进行轨迹规划，主要侧重于模仿和泛化人类运动。虽然能够精确模仿复杂动作，但它们固有的时间依赖性使得难以适应动态环境或是抵抗时空扰动。SEDS 及其改进方法通过构建动态系统模型来预测机械臂运动，需要对 Lyapunov 函数和动态系统进行深入学习。虽然它们在理论上能提供稳定的轨迹预测，但实现过程中需调整的参数较多，训练耗时长，实现复杂。同时，基于时不变的动态系统（DS）的方法虽然在理论上能够提高在线适应性，但在实际应用中，许多时不变 DS 方法在复现人类动作的精度和稳定性上仍存在不足；二是稳定性与可解释性：基于神经网络的方法因其强大的非线性建模能力被用于复杂运动的学习。然而，这些方法一般训练耗时长，调参困难，且模型稳定性难以保证。基于强化学习的方法通过与环境的交互学习提高了机器人的适应性，它们面临的一个主要问题是仿真到现实的迁移难题。此外，强化学习模型往往复杂且难以解释，这不仅增加了理解的难度，而且缺乏稳定性的保证。

尽管国内外许多研究小组在这些方面已经取得了初步的研究成果，但在动态目标抓取上仍然面临着许多技术难题和实际挑战。这些挑战一方面源于对轨迹预测算法更高的性能要求，另一方面则是对机器人的敏捷运动控制需要。基于这一研究背景，本文旨在探索一种更高效率的机器人抓取框架和算法以应对动态目标抓取任务，提高机器人的自主性和智能化水平。

1.3 研究目标与内容

国内外动态目标抓取任务研究现状表明目前的传统物体轨迹预测方法对非规则物体建模困难，机器学习方法对数据存在依赖性，深度学习需要巨大的数据集来支持模型训练，且难以解释，这些问题都限制了机械臂抓取在工业、人机

协同等场景下的应用。在处理复杂环境下的机器人高动态性运动控制问题时，传统的离线规划策略常常不能满足对动态目标运动实时性的要求，诸如 SEDS 等基于概率方法构建的运动方程在实际应用中不仅难以实现精确的运动轨迹复现，而且还涉及到复杂的参数调整问题，增加了实践应用的难度。

为了系统化地解决这些挑战，本文将以动态目标抓取系统中的动态目标轨迹预测与机械臂的敏捷运动控制作为研究内容，如图 1-3 所示。

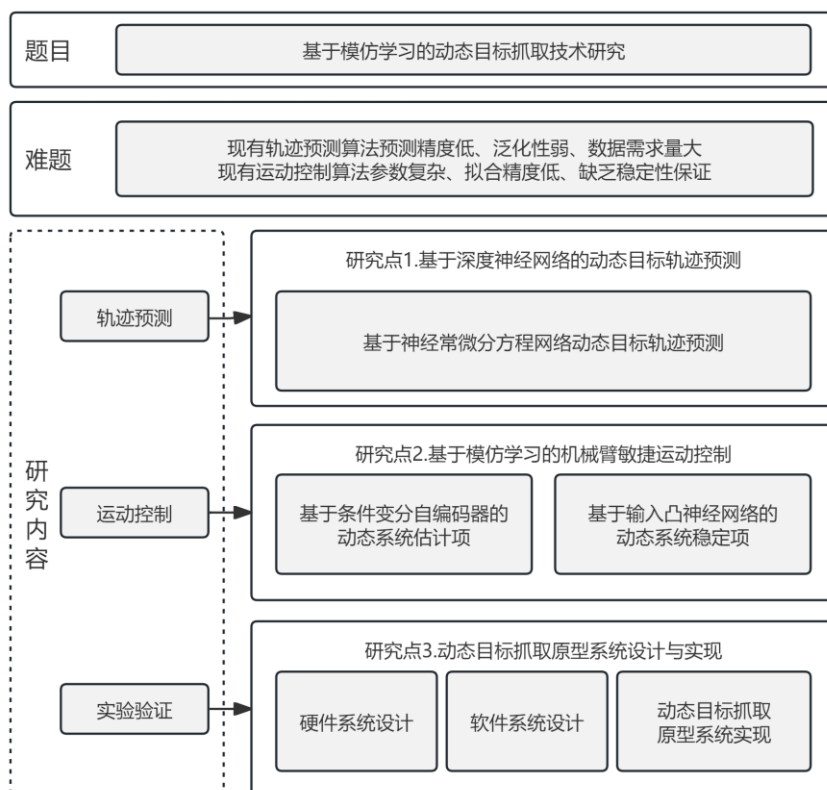


图 1-3 本论文的主要研究内容

针对非规则物体在运动过程中遇到的挑战，如运动学模型的建立困难、模型参数难以辨识以及缺乏泛化性，本研究深入分析了物体在运动中变化的受力情况，基于物体运动的微分方程设计了相应的网络模型用于轨迹预测任务。

针对机械臂运动控制任务遇到的问题，本研究采用从基于模仿学习的角度展开研究，利用条件变分自编码器网络来学习动态系统估计项，通过输入凸神经网络构造李雅普诺夫函数，求解动态系统稳定项，以实现复杂运动轨迹的精确复现与运动稳定性保证。

在机器人操作任务的研究中，虽然仿真环境提供了一个便捷的方式来模拟各种情境，但实物实验是确立方法有效性与实用性的关键步骤。它不仅作为最具说

服力的证据，确认了算法在理论上的成果能够在实际物理世界中得以应用，同时也能暴露那些在仿真中可能未能充分考虑的问题。因此，本研究建立了专门的实验平台，搭建了原型系统，对提出的算法进行实物测试和验证，从而确保所提出的解决方案在解决实际机器人操作任务时的有效性和实用性。

1.4 论文结构安排

根据研究内容，本文分为六个主要章节，具体结构安排如图 1-4 所示。

第一章为绪论。本章首先介绍了研究的背景与意义，突出了机器人技术在当前主要应用领域中的发展现状。接着，从两个不同的角度对现有的动态目标轨迹预测方法及机械臂敏捷运动规划控制的分类、优势与不足进行了分析。最后，明确了本研究面临的问题，阐述了研究目标、具体内容以及本文的整体结构。

第二章为数据采集及预处理。详细介绍了飞行物体轨迹和机械臂运动教学的数据采集和预处理过程。首先使用 Vicon 运动捕捉系统来采集的飞行物体的运动数据。然后使用三次样条插值和卡尔曼滤波等技术来处理数据的不准确性。最后搭建机械臂数据采集平台进行数据采集，将轨迹数据中的绝对位置数据转换为相对位置以增强模型泛化能力。这部分的基础工作构建了强大数据集为进一步的研究做好准备。

第三章为基于深度神经网络的动态目标轨迹预测。本章首先强调了在机械臂捕捉或拦截运动物体过程中，进行有效预测的重要性。随后，深入分析了动态目标的动力学模型，并基于此提出了使用神经微分方程网络进行轨迹预测的新方法，进而详细介绍了预测模型的构建流程，包括网络结构的设计、参数设置等关键技术细节。最后，通过采用网络性能评价指标，全面分析了所提出模型的性能，验证了其在轨迹预测方面的有效性和准确度。

第四章为基于模仿学习的机械臂敏捷运动控制。本章首先介绍了模仿学习的概念，选择模仿学习的原因。继而，深入探讨了基于模仿学习的动态系统理论，包括如何构建动态系统以及如何分析系统的稳定性。然后提出基于条件变分自编码器与凸神经网络联合学习稳定变分动态系统。最后，通过在仿真环境及真实机器人环境中进行的实验，验证了所提算法的有效性，证明了模仿学习在敏捷运动控制领域的潜力和实用性。

第五章为原型系统设计与实现。本章旨在通过搭建实验平台，验证第三、四章提出的轨迹预测和机械臂运动控制算法的实际效果。工作内容包括硬件系统的设计（如机械臂、Vicon 系统和灵巧手平台的搭建）和软件系统的设计（如 ROS

平台的搭建和算法的实现)。最后,通过具体实例验证算法的有效性,并分析实验结果。

第六章为总结与展望。本章先对本论文的所研究的两个方向的主要内容和成果进行总结。其次讨论了在研究过程中遇到的一些挑战和未解决的问题,最后对未来的研究方向提出了展望。

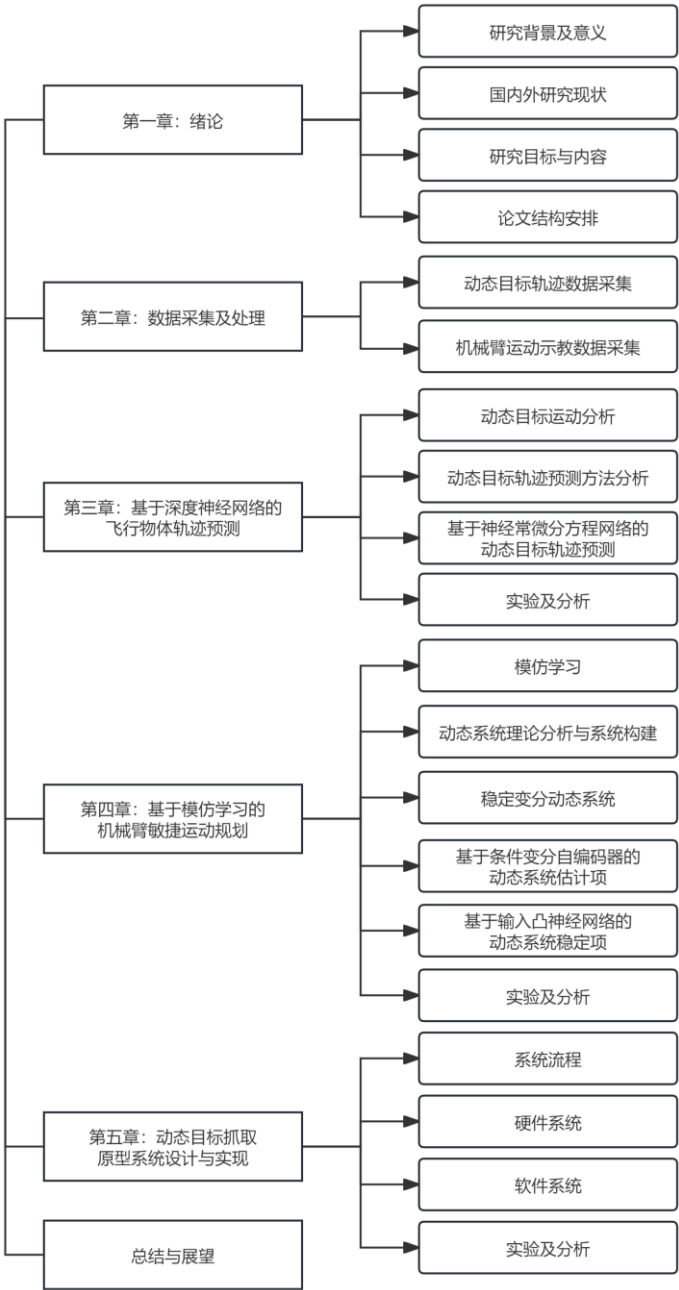


图 1-4 本文章节安排与研究内容

第二章 数据采集及预处理

2.1 引言

在机器人技术领域，尤其是涉及动态目标轨迹预测和机械臂运动模仿学习的研究中，数据采集与处理是实现高精度预测和有效示教的关键环节。高质量的数据不仅是模型学习和优化的基石，而且直接影响了预测准确性和动作执行的精细度。随着深度学习技术的发展，对高质量数据的需求愈发显著。本章将重点介绍如何通过先进的技术手段进行数据准备工作，如通过 Vicon 运动捕捉系统来采集飞行物体运动轨迹，通过示教器示教采集机械臂的运动轨迹数据，并通过三次样条插值、卡尔曼滤波和相对化等技术进行数据预处理工作。这些步骤不仅为研究提供了高质量的数据，也为深度学习模型的训练和验证打下了坚实的基础。

2.2 动态目标轨迹数据采集

在这一小节中，首先介绍飞行物体轨迹数据采集流程，描述了如何利用 Vicon 运动捕捉系统来获取飞行物体在空间中的运动数据。随后，本文将进一步介绍对原始数据进行预处理的必要步骤，包括如何应用三次样条插值和卡尔曼滤波等技术来优化数据。

2.2.1 轨迹数据采集

2.2.1.1 Vicon 运动捕捉系统

本文中使用的 Vicon 运动捕捉系统硬件部分主要包括：

(1) 红外捕捉相机：

实验室配备了 8 个红外捕捉相机(Vicon Vantage V8)，通过实时采集和计算反光标记点的三维坐标进行三维运动捕捉分析。Vicon Vantage V8 相机具有 800 万像素，视场角为 $61.7^{\circ} \times 47^{\circ}$ ，最高像素时最大捕捉频率可达 270 帧，能够同时实现多个自主刚体物体跟踪的动态运动捕捉。

(2) T 型标定杆

通过标定杆对 Vantage 摄像机系统中的光学摄像头进行校准，同时建立运动捕捉系统下的坐标系原点。经过标定后运动捕捉系统即可采集捕捉空间内运动物体的空间坐标。

(3) POE 交换机

红外捕捉相机将处理的反光标记点的三维位置信息通过 POE 交换机传送到主机的 Tracker 软件上进行进一步的处理。

(4) 反光标记球

反光标记球能够反射红外捕捉相机发射出来的 850nm 红外光源，从而能够被捕捉到，只有同时被三个及以上的红外捕捉相机捕捉到才能确定反光标记球的三维位置。

(5) Tracker 软件

Tracker 软件是与 Vicon 运动捕捉系统配套的可视化操作软件，拥有简单易学且功能强大的用户界面，提供低延迟的实时数据。Tracker 3 可以在 1.5ms 的时间内进行数据处理，处理速度为 500 帧/秒以上，是其他系统的五倍。通过 Tracker 对视场区域进行标定，确定坐标系，可以在三维空间中实时显示反光标记点的位姿信息，记录物体的位姿轨迹。



(a) Vicon 红外相机



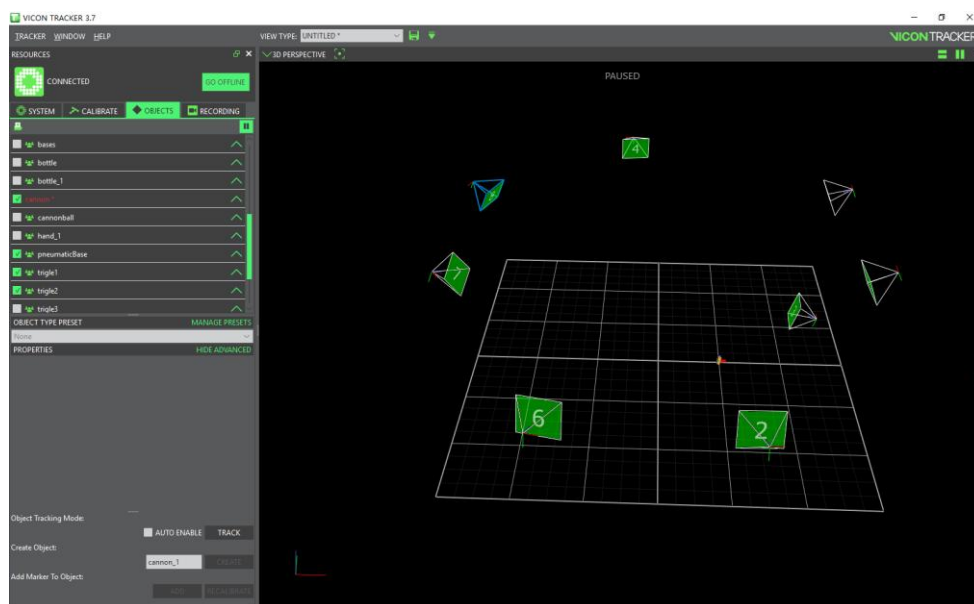
(b) 标定杆



(c) Poe 交换机



(d) 标记点



(e) Track 软件

图 2-1 数据采集系统。(a)Vicon 红外相机；(b)标定杆；(c)Poe 交换机；(d) 标记点；(e)Track 软件；

2.2.1.2 数据采集环境搭建

基于上述的 vicon 动捕系统，本章搭建了一个包含 8 台红外捕捉相机数据采集系统，形成了一个 $6\text{m} \times 6\text{m}$ 的动作捕捉空间，能够以 100Hz 的采集频率对空间中的刚体进行信息采集。为了保证采集到的数据尽可能的准确以及降低丢帧率，选择捕捉空间中心 $3\text{m} \times 2\text{m}$ 区域进行数据采集，如图 2-2 红色区域所示，实际采集场景如图 2-3 所示。

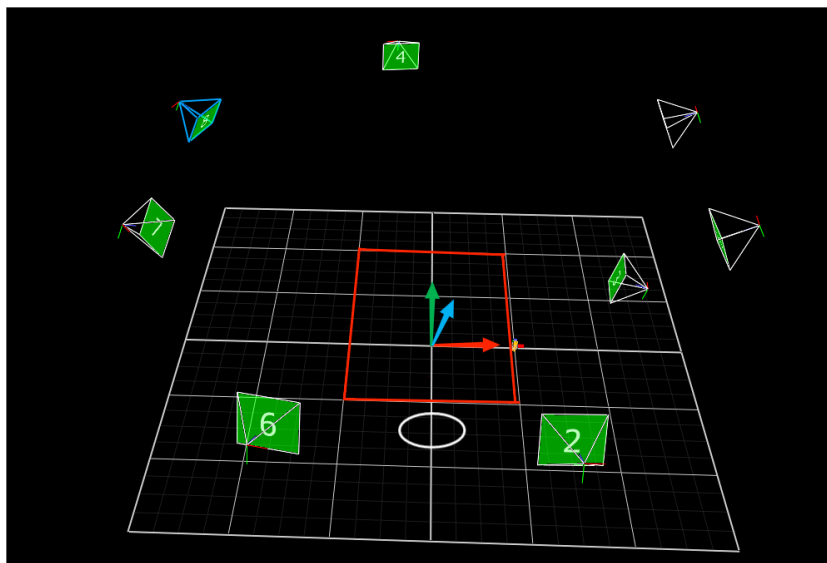


图 2-2 数据采集区域



图 2-3 数据采集场景

飞行物体只有进入红色区域内才会被记录下来。在数据采集过程中，通过模拟人与机器人交互的过程，由人将物体抛出经过捕捉空间的红色区域到达白色的机械臂的工作空间内。其中，人的初始位置、抛出位置、抛出速度、角度均是随机的。

2.2.1.3 飞行物体数据采集

为了模拟日常交互环境，本章选择了三种不同的物体如图 2-4 所示：网球、塑料水瓶和迫击炮弹形状的玩具作为研究对象。



图 2-4 研究对象物体

为了捕捉每个物体的位姿轨迹，首先在物体上贴上了不同数量的反光标记，并在 Tracker 软件中建立刚体模型。在本文的实验中一共采集了 700 个轨迹样本，每个样本以 100Hz 的频率记录，这些数据后续用于模型训练。典型的轨迹数据如图 2-5 所示。

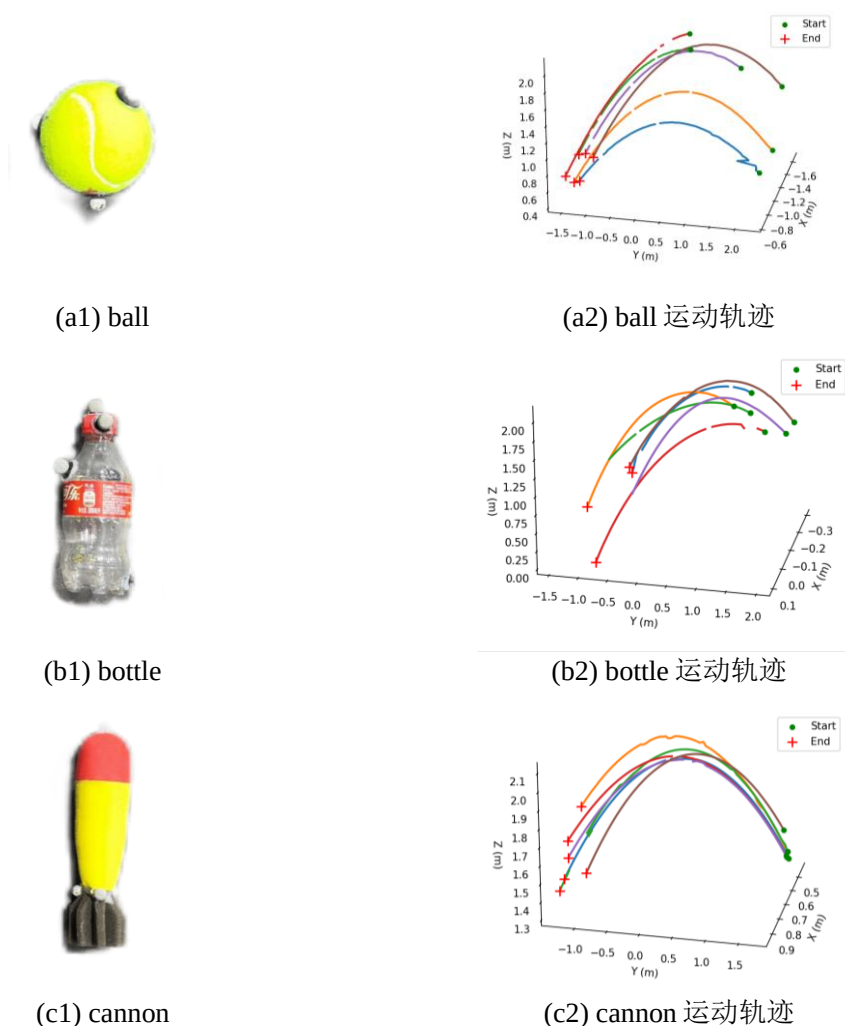


图 2-5 原始轨迹数据

图中展现了每种物体在三维空间中的 6 条轨迹，绿色和红色标记点分别表示轨迹起点和终点，轨迹的形状类似于抛物线。表 2-1 展示了物体抛出的平均速度与标准差。

表 2-1 运动物体初始速度（平均值±标准差）

	初始速度 (m/s)			终点速度 (m/s)		
	x	y	z	x	y	z
ball	0.32 ± 0.49	4.82 ± 1.10	7.72 ± 2.51	0.49 ± 0.26	4.48 ± 0.60	3.92 ± 1.93
bottle	0.03 ± 0.26	4.58 ± 0.43	3.22 ± 0.68	0.29 ± 0.25	3.08 ± 0.72	9.25 ± 1.26
cannon	0.27 ± 0.41	4.03 ± 0.34	3.63 ± 0.38	0.40 ± 0.31	3.25 ± 0.52	4.27 ± 0.74

其中物体抛出速度最高可达 7.72m/s，终点速度最高可达 9.25m/s。通过对速度的统计分析进一步强调了抓取高速运动物体所面临的挑战。

以上数据均是在本地环境中进行采集记录，从图中可以发现数据中不仅存在着显著的噪声影响，还存在着数据丢失的情况。分析表明这些问题是由多种因素造成的，除了由于物体形状非规则导致的非线性力影响外，还包括环境遮挡导致的标记点缺失等。

2.2.2 轨迹数据预处理

为了应对数据中存在的问题，本章采用了数据插值技术来重建缺失的部分轨迹，确保数据的完整性。此外，为了降低噪声对数据质量的影响，进一步应用了卡尔曼滤波算法对数据进行平滑处理。

2.2.2.1 数据插值

针对数据局部缺失的问题，本文采用三次样条插值法对轨迹数据中的缺失数据进行补全。三次样条插值是数值分析中一种广泛使用的方法，旨在通过一组已知的离散数据点，构造一条光滑的连续曲线。该方法的核心在于保证曲线在各段间二阶导数的连续性，以实现高度的光滑过渡。其基本原理是将一系列低阶多项式应用于数据的局部区间，并通过这些多项式在接合点的衔接条件，确保整条曲线的整体光滑性。具体而言，假设有 $n+1$ 个已知数据点 $\{(x_i, y_i)\}_{i=0}^n$ ， $x_0 < x_1 < \dots < x_n$ ，三次样条插值法旨在找到一系列三次多项式函数 $\{S_i(x)\}_{i=0}^{n-1}$ 满足以下条件：

- 1) 对于每个子区间 $[x_i, x_{i+1}]$ ，有一个对应的三次多项式 $S_i(x)$ ；
- 2) 在相邻区间的连接点 x_i ， $S_i(x)$ 和 $S_{i-1}(x)$ 具有相同的一阶和二阶导数值，即 $\forall i \in \{1, 2, \dots, n-1\}$ ，满足 $S'_{i-1}(x_i) = S'_i(x_i)$ 和 $S''_{i-1}(x_i) = S''_i(x_i)$ ；

3) 函数序列 $S_i(x)$ 通过所有给定的数据点，即对于每个 $i \in \{0, 1, \dots, n\}$ 都有 $S_i(x_i) = y_i$ 。

通过三次样条法进行数据插值能够在保证曲线平滑的同时，精确地穿过每个给定的数据点，从而有效地解决数据缺失问题。对于三维空间中的位置轨迹数据，需要对轨迹的每个维度（通常是 X 、 Y 和 Z 坐标）分别执行插值过程。这意味着，对于三维空间中的一系列数据点 $\{(x_i, y_i, z_i)\}_{i=0}^n$ ，插值方法将独立地对 x 坐标、 y 坐标以及 z 坐标对应的值执行三次样条插值。可以将三维轨迹描述为三个函数 $X(t), Y(t), Z(t)$ ，分别对应于三维空间中的 x, y, z 坐标，根据已知数据解算函数，再将缺失时间点作为函数的输入，获得缺失时间的坐标数值。

2.2.2.2 卡尔曼滤波

许多对飞行物体轨迹预测的研究中均应用了滤波算法对数据进行预处理，而在这些算法中卡尔曼滤波的应用尤为广泛。卡尔曼滤波（Kalman Filter，KF）是非常经典的预测追踪算法，能够在系统存在噪声和干扰的情况下进行系统状态的最优估计，广泛应用在导航、制导、控制等相关的领域。同时卡尔曼滤波具有内存占用较小、速度快等优点，非常适合动态变化的系统。在轨迹预测任务中，为了保证提高预测的准确性，需要对实时的轨迹数据进行滤波处理。卡尔曼滤波一般可以分为预测和修正两个步骤，表示如下：

预测阶段：

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \quad (2-1)$$

$$\mathbf{P}_{k+1} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q} \quad (2-2)$$

修正阶段：

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}^T(\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R})^{-1} \quad (2-3)$$

$$\mathbf{x}_k = \mathbf{x}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{x}_k) \quad (2-4)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k \quad (2-5)$$

其中预测阶段 $\mathbf{x}_k = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]$ 代表系统状态变量， $\mathbf{A}$ 为状态转移矩阵， $\mathbf{B}$ 为状态输入矩阵， $\mathbf{u}$ 为输入变量， $\mathbf{P}$ 为先验状态估计协方差矩阵， $\mathbf{Q}$ 为过程噪声协方差矩阵。修正阶段中 $\mathbf{K}$ 为卡尔曼增益， $\mathbf{H}$ 为观察矩阵， $\mathbf{R}$ 为测量噪声协方差矩阵， $\mathbf{z}_k$ 为观测量， $\mathbf{P}_k$ 为后验状态估计协方差矩阵。使用卡尔曼滤波器需要先初始化 $\mathbf{R}$ 、 $\mathbf{P}$ 、 $\mathbf{Q}$ ，每次先进行预测，再进行更新，循环进行，直至最后一个数据被处理完。

2.2.2.3 数据处理

在预处理过程中，本章采用了三次样条插值和卡尔曼滤波两种技术。三次样条插值用于填补数据中的缺失值，保证数据的连续性，而卡尔曼滤波则有效地削弱了噪声的影响。原始轨迹数据集在经过数据插值与卡尔曼滤波预处理后轨迹数据如图 2-6 所示。

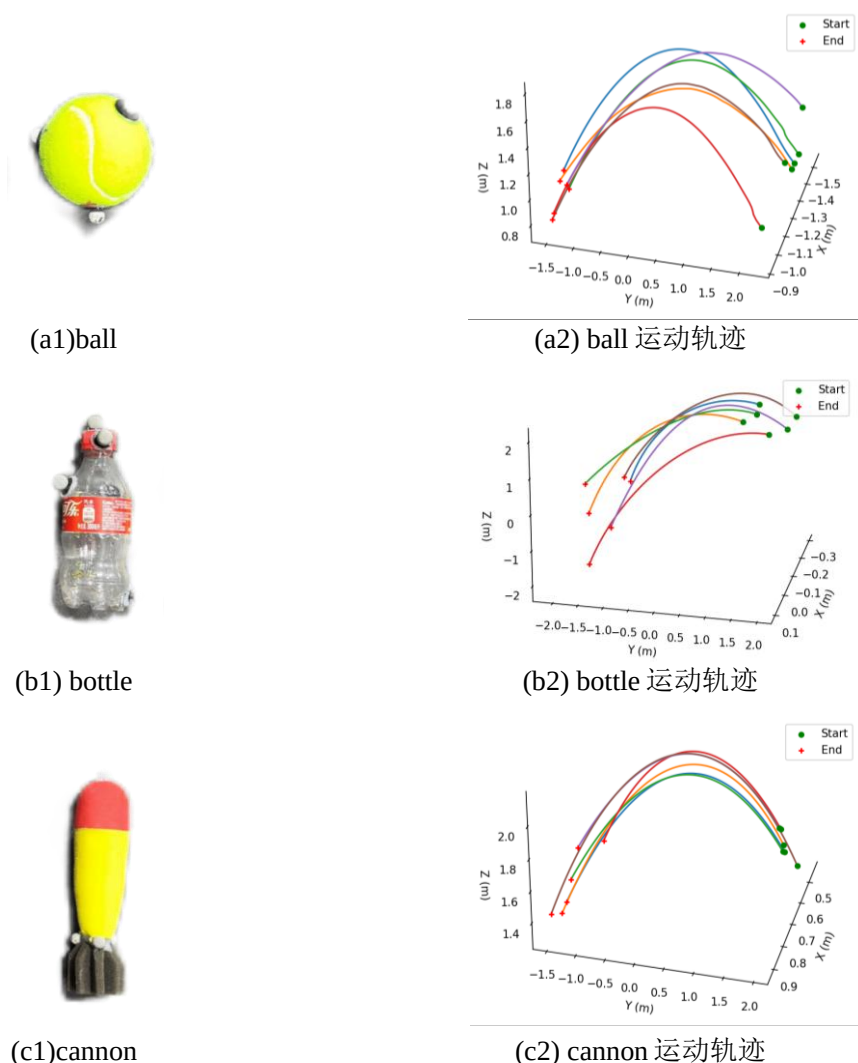


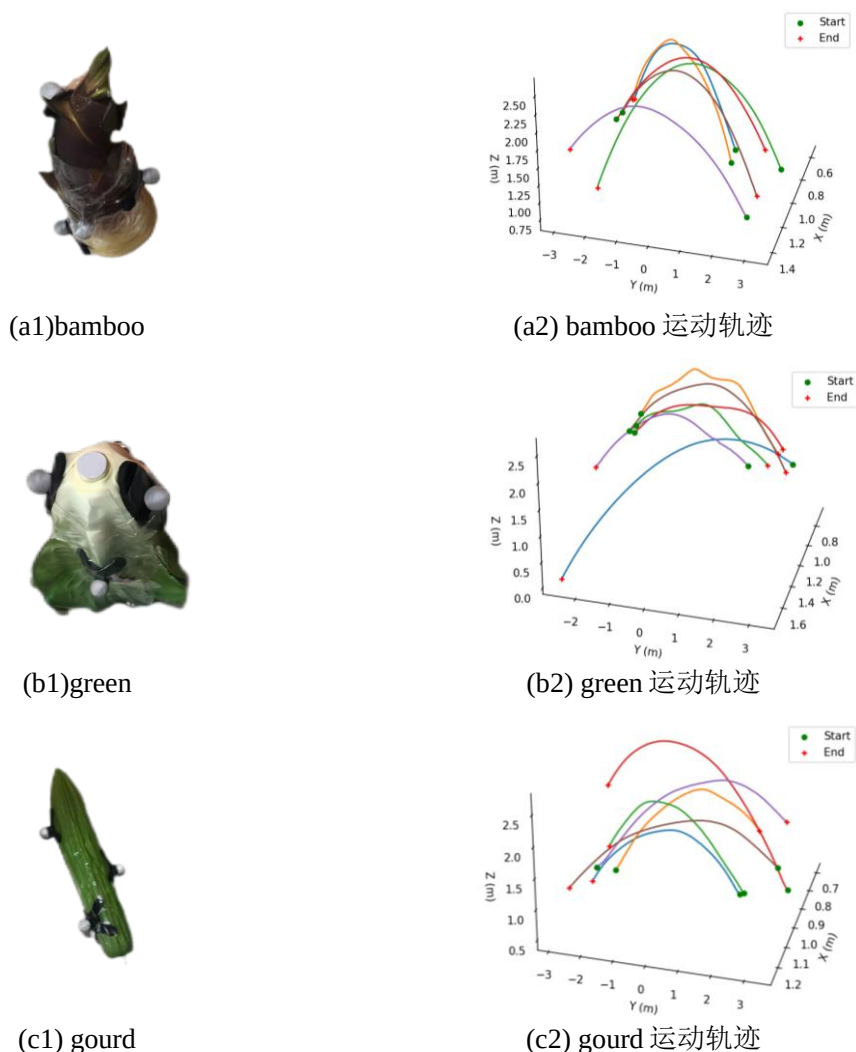
图 2-6 预处理后的轨迹数据

图中左侧是物体的类型，右侧是物体在三维空间中飞行的运动轨迹，从图中可以观察到，经过预处理后的物体运动轨迹数据变得更加平滑，噪声得到有效削减。为了对算法效果进行客观真实的评价，本文在算法研究过程用采用了浙大机器人实验室提供的公开的数据集^[44]作为模型评估的标准数据集。该非规则物体数据集由 1500 条运动轨迹组成，包括 6 类形状各异的物体，如图 2-7 所示。



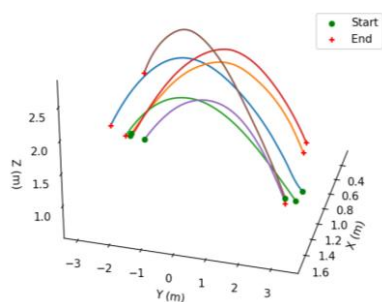
图 2-7 六类非规则物体

包括由聚氨酯材料制成的香蕉，竹子，绿色和苦瓜蔬菜模型，毛绒玩具 Paige 和一个装了半瓶水的瓶子。与本文数据采集过程类似，该数据集将反光标记球贴在物体表面，且尽量保持对称来构建刚体，在数据记录过程中通过 vicon 系统数据集以 120hz 的采集频率记录物体运动过程中的刚体质心的位姿数据。对该数据集进行数据预处理操作后得到了如图 2-8 所示的轨迹数据，图左侧为非规则物体，右侧为其预处理后的运动轨迹。





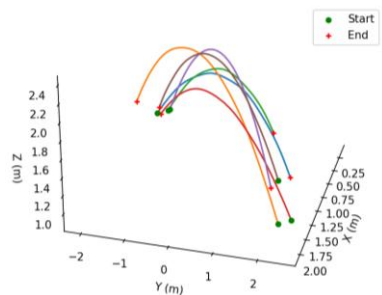
(d1)paige



(d2) paige 运动轨迹



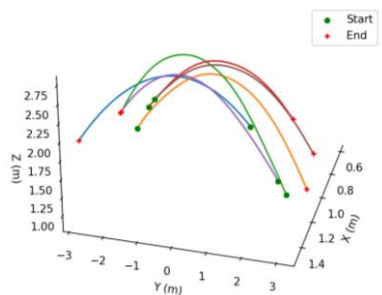
(e1) banana



(e2) banana 轨迹



(f1) bottle(half)



(f2) bottle(half)运动轨迹

图 2-8 预处理后非规则物体轨迹数据

2.3 机械臂运动示教数据采集

本节主要介绍了机械臂运动教学过程中的示教数据采集与预处理工作，这是实现机械臂精确动作重现和后续高效自主学习的关键步骤。

2.3.1 示教数据采集

UR5e 是由丹麦公司 Universal Robots 制造的一款灵活协作机器人，由于其高可靠性与拓展性常被各大实验室用于机器人实验，本文基于 UR5e 机械臂搭建了示教平台进行示教数据的采集（如图 2-9 所示），该平台包括一个六轴机械臂和其示教控制器，操作者可以通过一个直观的图形用户界面（GUI）来编程和操作机器人。



图 2-9 示教设备

在数据采集的过程中需要通过示教器将 UR5e 机械臂设置为重力补偿模式，在重力补偿模式下，机械臂能够感知并抵消由于重力造成的影响，从而允许更精确和轻松的手动操控。在每次记录开始前，需要先将机械臂复位到指定的初始位姿状态，确保每次数据采集的一致性。随后启动轨迹数据记录程序，操作者可以直接手动移动机械臂到期望的位置和姿态，同时机械臂内的传感器将以 100Hz 的频率持续监测并记录关节角度、末端执行器的三维坐标、方向、速度等信息。最终将记录数据保存为表格的数据集形式，其中数据集 D 表示为：

$$D = \{\xi_{t,n}\}_{t=0,n=1}^{T^n,N}, \xi = (\mathbf{x}, \mathbf{v}) \quad (2-6)$$

其中 t 表示记录的时间， n 表示第 n 条轨迹， $\mathbf{x}$ 表示末端 TCP 的三维位置， $\mathbf{v}$ 表示机械臂末端 TCP 的三维轴角姿态。图 2-10 展示了两次的示教所采集的运动轨迹。

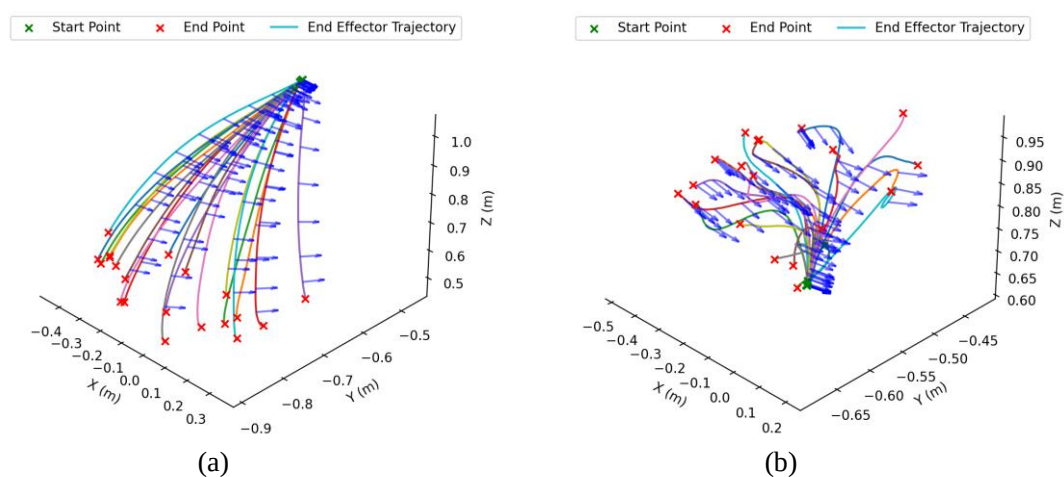


图 2-10 原始示教数据。(a)示教轨迹 1；(b)示教轨迹 2；

其中曲线表示机械臂末端执行器的运动轨迹，曲线上的向量表示轴角姿态，绿色标记代表轨迹的起始位置，由于在每次采集过程中均回到同一个初始位置，所以在图中显示在同一个位置，红色标记代表运动的终点。在示教过程中每次随机改变终点位置，每个示教数据集由 20 条示教轨迹组成。通过这种示教的方式，可以记录机械臂在各个时间点的精确位姿状态，为后续的数据处理和模型学习任务提供了充足的数据源。

2.3.2 示教数据预处理

为了使机械臂运动控制算法能够满足动态场景要求，同时可以将算法快速迁移到新的机器人环境中，本文将采集的机械臂基座标下的绝对位姿数据转换成相对位姿数据，这对于任意的机器人环境均具有普适性。假设数据集有 N 条轨迹数据，每一条轨迹的长度为 T^n ，则相对位置轨迹数据集 P 转换过程如下式。

$$P = \{\mathbf{x}_{t,n} = \bar{\mathbf{x}}_{t,n} - \bar{\mathbf{x}}_{t_{p,n},n}\}_{t=0,n=1}^{T^n,N} \quad (2-7)$$

其中 $\bar{\mathbf{x}}_{t,n}$ 表示绝对位置， $\mathbf{x}_{t,n}$ 表示相对位置。对于欧几里得空间内的位置信息可以直接通过公式(2-7)进行相对数据转换，但是对于姿态数据就不适用。因为轴角表示的姿态难以直接进行运算，相比其他姿态表示方式，四元数仅需 4 个元素就能完整描述空间中的任意旋转，其乘法和旋转操作不涉及矩阵的复杂运算，因而在计算上更为高效。可以先将表示姿态的轴角数据转换成四元数，通过四元数计算相对的旋转量，然后再转换到轴角。假设一个轴角表示为 $\mathbf{v} = (x, y, z)$ ，其模（即向量的长度） $\theta = \|\mathbf{v}\|$ 表示旋转角，轴角分量为 $\mathbf{u} = (u_x, u_y, u_z) = \mathbf{v} / \|\mathbf{v}\|$ ，则轴角转四元数可以表示为：

$$\mathbf{q} = (q_w, q_x, q_y, q_z) = \left(\cos\left(\frac{\theta}{2}\right), u_x \cdot \sin\left(\frac{\theta}{2}\right), u_y \cdot \sin\left(\frac{\theta}{2}\right), u_z \cdot \sin\left(\frac{\theta}{2}\right) \right) \quad (2-8)$$

其中 $\mathbf{q}$ 表示转换的四元数。对整个轴角序列进行转换后得到四元数表示的绝对姿态序列 $\bar{Q} = \{\bar{\mathbf{q}}_{t,n}\}_{t=0,n=1}^{T^n,N}$ ，将四元数绝对序列转换为相对姿态序列过程如下：

$$Q = \{\mathbf{q}_{t,n} = \bar{\mathbf{q}}_{t,n} / \bar{\mathbf{q}}_{t_{p,n},n}\}_{t=0,n=1}^{T^n,N} \quad (2-9)$$

其中 Q 表示为相对姿态序列，与位置的处理方式不同，在四元数中求相对旋转表示为四元数的除法，最后再将四元数序列转换成轴角数据。在经过数据预处理后就得到了如图 2-11 所示的相对运动轨迹。

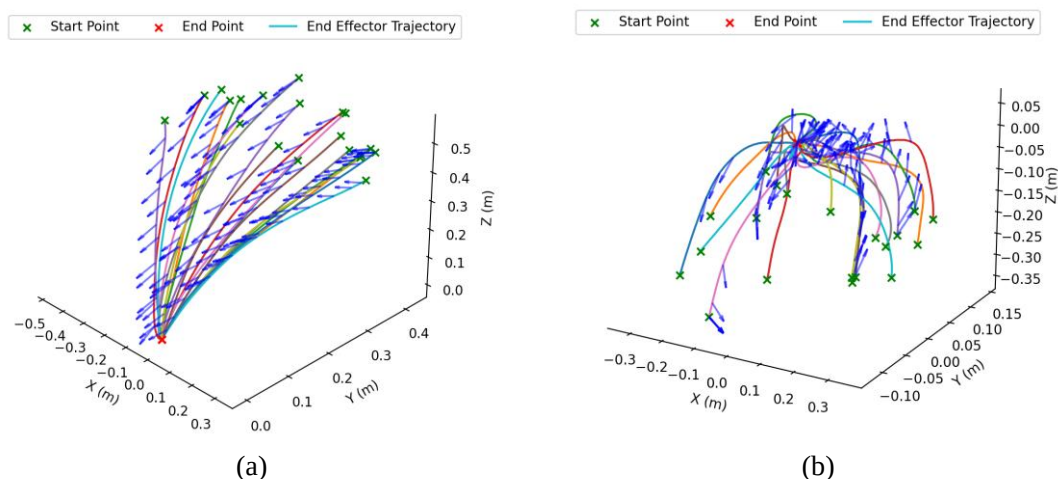


图 2-11 相对示教轨迹。(a)示教轨迹 1；(b)示教轨迹 2；

其中图 2-11(a)为图 2-10(a)经过预处理后的相对轨迹，其中图 2-11(b)为图 2-10(b)经过预处理后的相对轨迹。经过数据预处理后，每条轨迹的起点各不相同而终点均为同一个点，轨迹姿态也是经过类似的处理。鉴于这种相对性的数据处理方式对于任意机械臂都是通用的，采用相对数据进行模型训练能够提高模型的适用性，可以将算法快速迁移到新的应用场景。

2.4 本章小结

本章主要介绍了动态目标轨迹预测与机械臂敏捷运动控制所需要的数据采集及预处理工作。首先是通过人工抛物的方式采集了三种不同物体的飞行运动轨迹，使用三次样条插值法对轨迹缺失坐标进行插补，利用卡尔曼滤波方法对轨迹数据进行滤波得到去噪后的训练数据集。然后搭建了机械臂运动数据采集平台，为了使模型更具泛化性将绝对位姿转换成相对位姿构建数据集。本章节主要完成了相关算法的数据准备与数据预处理工作，为下面的研究做好充足的数据准备。

第三章 基于深度神经网络的动态目标轨迹预测

3.1 引言

动态目标的抓取还面临着许多现实问题，这些问题不仅源于机械硬件的物理限制，如机械臂末端执行器的最大速度和关节的最大加速度限制，还有目标物体的非规则性与动态目标所处环境不确定性，这些都加剧了运动过程中的非线性因素影响。在动捕系统与机械臂的实时交互中，抓取系统将根据新采集的数据不断更新目标的预测位置，迫使机械臂根据新的预测位置重新规划运动路径。动态抓取还需要高速且准确的传感器反馈和数据处理能力，如果运动目标受到传感器设备噪声的影响，有可能导致预测位置偏差较大，这会对机械臂的启动造成延迟。如果预测所耗费时间过长，这有可能导致机械臂运动时间不足，无法实现有效抓取。为了能够及时地抓取运动中的物体，机器人系统必须要在极短时间内精准预测动态目标的未来位置，并据此规划抓取路径。本章将以非规则的飞行物体为研究的动态目标对象，深入研究动态目标轨迹预测方法。

3.2 动态目标运动分析

3.2.1 动力学模型分析

非规则物体在飞行中除了受到重力外还受到空气动力学效应^[47]，包括阻力、升力和马格努斯力等的影响，这些力作用于物体的不同部位，导致其受力情况复杂多变，简要分析其受力情况如图 3-1 所示。

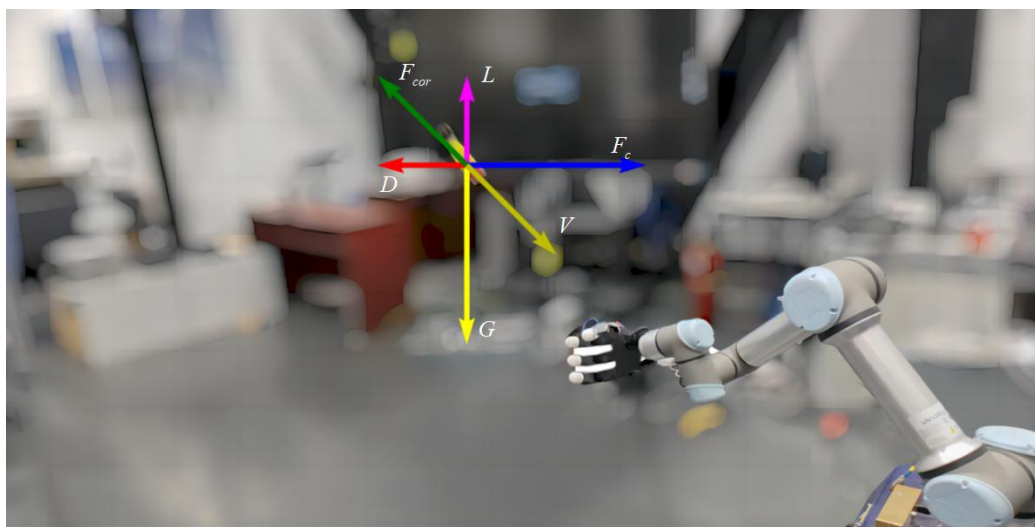


图 3-1 飞行物体受力情况

其中 G 表示重力，这是由地球引力引起的，作用于物体的质心，指向地心； L 表示升力，是物体通过空气移动时，在其上表面产生的力，通常垂直于流动方向； D 表示阻力，是物体在空气中移动时遇到的阻碍力，方向与物体的运动方向相反；物体受力情况表示如下式。

$$G = m \cdot g \quad (3-1)$$

$$L = C_L \cdot \frac{1}{2} \cdot \rho \cdot v^2 \cdot A \quad (3-2)$$

$$D = C_D \cdot \frac{1}{2} \cdot \rho \cdot v^2 \cdot A \quad (3-3)$$

其中 m 是物体的质量， g 是重力加速度， C_L 是升力系数， ρ 是空气密度， v 是物体相对于空气的速度， A 是物体在垂直于速度方向上的投影面积， C_D 是阻力系数。

考虑到旋转对飞行物体受力的影响，需要引入额外的力量来描述这种情况。旋转物体会经历几种额外的力，最重要的是离心力和科里奥利力，这些力对飞行物体的运动和稳定性有显著影响。其受力公式如下：

$$F_c = m \cdot \omega^2 \cdot r \quad (3-4)$$

$$F_{cor} = 2m \cdot v \cdot \omega \cdot \sin(\theta) \quad (3-5)$$

其中 ω 表示角速度， θ 表示物体速度方向与旋转轴之间的角度。 F_c 表示离心力，是当物体沿着曲线路径移动时受到的向外的力，这是由于物体倾向于保持其直线运动状态而产生的； F_{cor} 表示科里奥利力，它垂直于物体的运动方向和旋转轴，科里奥利力的大小与物体的速度和角速度有关；

传统轨迹预测方法在处理非规则物体时常常面临精度低下的问题，这在很大程度上与物体模型的构建和应用密切相关。这些传统方法通常依赖于简化的物理模型和假设，比如忽略空气动力学效应、简化物体的形状或假设线性运动，这样做虽然降低了计算复杂度，但同时也限制了模型的适用范围和精度。

3.2.2 运动轨迹预测问题分析

对于本文中研究的非规则飞行物体运动轨迹而言，要实现对运动轨迹进行高精度的预测主要存在着三个问题：

1) 非线性

非规则形状的复杂运动物体在飞行或运动过程中，往往受到多种非线性因素的影响。这些因素常包括空气动力学效应（涡流和湍流）、内部机械动力的变化（如机械弹性或震动）、以及环境条件的变化（如风速和风向的波动）。这些非线性因素导致运动轨迹预测模型的建立需要复杂的动力学分析和高级数学建模，传统的线性模型很难捕捉这些复杂的动态变化。

2) 适应性

在研究非规则飞行物体运动轨迹预测时，一个常见的挑战是预测模型在特定条件下可能展现出良好的性能，但当遇到新的、未曾经历过的飞行环境时，其性能可能会显著下降。这个问题的核心在于模型泛化能力的局限性，即模型对于训练数据之外的新情况适应和预测的能力。泛化能力受限的原因多样，包括数据集的多样性不足、模型过于复杂导致的过拟合、以及模型未能捕捉到飞行轨迹背后的深层物理和环境规律。

3) 实时性

当物体运动速度过快时，对实时计算能力提出了更高的要求。在高速运动情况下，预测算法需要快速响应，以便实时更新轨迹信息。这意味着计算过程需要极高的效率，以及能够在极短的时间内处理大量数据的能力。

3.3 动态目标轨迹预测方法分析

迄今为止，轨迹预测领域已累积了丰富的研究成果。本文选取并分析了几种具有代表性的预测方法，既包括传统方法，也包括基于深度神经网络的先进方法。具体而言，本章探讨了牛顿动力学（Newton）、最小二乘法拟合（GLS），以及基于循环神经网络 LSTM 的 Seq2Seq 预测网络——ResLSTM。此外，本文还提出了一种基于神经微分方程网络的飞行物体位置预测方法，与前述方法进行了比较分析。

3.3.1 牛顿动力学

如果将动态目标假设为一个投掷的球，在理想条件下，不考虑空气阻力和其他外力，这个物体的动力学可以使用牛顿第二定律和运动学方程来描述。其表达式为：

$$F = ma \quad (3-6)$$

其中 F 是作用在物体上的力， m 是物体的质量， a 是物体的加速度。如果以抛掷时刻为起点，设定物体在水平方向的初始速度为 v_{0x} 和 v_{0y} ，垂直方向的初始速度为 v_{0z} ，则物体的轨迹可以通过以下公式来描述：

$$\begin{aligned} x(t) &= x_0 + v_{0x}t \\ y(t) &= y_0 + v_{0y}t \\ z(t) &= z_0 + v_{0z}t + \frac{1}{2}gt^2 \end{aligned} \quad (3-7)$$

还可以将动力学模型用矩阵的形式进行表示。已知 t 时刻的状态向量为 s_t ，状态向量中包含了完整的动态信息：位置、速度和加速度，则 $t + \Delta t$ 时刻的位置可以表示为：

$$s_{t+\Delta t} = T(\Delta t) \cdot s_t \quad (3-8)$$

其中 $s_t = [x \ y \ z \ v_x \ v_y \ v_z \ a_x \ a_y \ a_z]^T$ ， Δt 表示在下一时刻与当前状态 t 的时间间隔，由于加速度保持不变，转移矩阵可以定义为：

$$T(\Delta t) = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-9)$$

基于运动物体本身物理属性与受力分析的运动学模型方法是早期常见的运动物体轨迹预测算法，牛顿动力学通过直接的力和运动关系导出物体运动动力学模型清晰地给出了预测结果。然而，对于形状各异的物体，其飞行过程常伴随复杂的旋转动作，使得运动学模型呈现高度非线性，难以准确建立精确的物理模型及拟合物体的真实运动轨迹。

3.3.2 最小二乘法拟合

基于最小二乘法（generalized least squares, GLS）进行轨迹预测不直接依赖于牛顿动力学，而是通过统计方法来估算未来的状态，基于过去的点拟合出一个多项式模型，然后用这个模型来预测未来的值。假设多项式的形式为：

$$p(x) = c_0 + c_1 * x + \dots + c_n * x^n \quad (3-10)$$

其中 n 表示多项式的最高项，在本文的研究中假设其最高项为 3。随后最小化实际数据点和多项式预测值之间加权差异的平方和：

$$E = \sum_j w_j^2 * |y_j - p(x_j)|^2 \quad (3-11)$$

其中 y_i 是观测值， $p(x_i)$ 是多项式模型在点 x_i 的预测值， w_i 是对应观测值的权重。为了找到最小化误差平方和的多项式的系数 c ，通常会设置了一个超定的矩阵方程：

$$V(x) * c = w * y \quad (3-12)$$

其中 $V(x)$ 是加权的伪范德蒙德（Weighted Pseudo Vandermonde Matrix）矩阵，它是一个根据多项式的基函数（如 $x^0, x^1, x^2, \dots$ ）构建的矩阵，并且已经与权重 w 结合。

最小二乘法通过最小化加权残差的平方和来估计模型参数，与牛顿动力学模型相比，其更具适应性，通过选择合适的权重，GLS 可以更好地适应数据的统计特性，当数据受到复杂外力影响时这对于动力学建模来说是非常有用的。但是由于其模型的复杂性与对数据的依赖性，在实验数据稀缺或噪声很大的情况下，可能会导致模型拟合不准确。

3.3.3 ResLSTM 神经网络

基于神经网络的轨迹预测方法往往不需要关于运动物体本身的先验属性，只需将重点放在研究轨迹数据序列背后隐藏的特点及规律，通过对数据建模的方式来研究运动物体的姿态及位置预测问题。本节基于深度学习提出了一种名为 ResLSTM 的网络结构用于飞行物体的轨迹预测问题。其结构如图 3-2 所示。

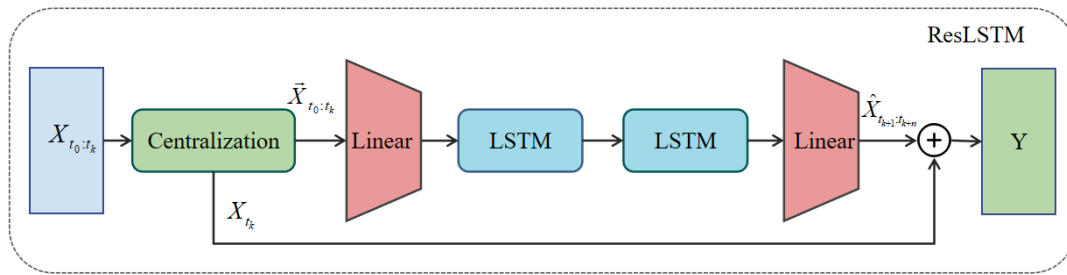


图 3-2 ResLSTM 网络结构

该网络以 seq2seq 结构为骨干，包括一个 Centralization 层，一个线性编码层、一个线性解码层和两个长短期记忆（LSTM）层。在飞行物体轨迹数据采集过程中，由于每次抛出的位置与速度均不一致，轨迹长短不一，在训练过程中无法进行正常的归一化或者标准化。而 Centralization 层通过从每个观测值中减去最后的观测值对输入数据进行中心化，这种处理能够使序列平稳来标准化输入数据，强调最近观察与过去之间的变化或增量过程，其过程如下式所示：

$$\bar{\mathbf{X}}_{t_0:t_k} = \mathbf{X}_{t_0:t_k} - \mathbf{X}_{t_k}, \mathbf{Y} = \hat{\mathbf{X}}_{t_{k+1}:t_{k+n}} + \mathbf{X}_{t_k} \quad (3-13)$$

其中 $\mathbf{X}_{t_0:t_k}$ 表示输入长度为 $k+1$ 的时序数据， $\mathbf{X}_{t_k}$ 代表最新的观测值， $\hat{\mathbf{X}}_{t_{k+1}:t_{k+n}}$ 表示模型的解码输出， $\mathbf{Y}$ 表示输出结果。

线性编码层的主要作用是特征转换，它将输入特征映射到更高维度的空间。第一个 LSTM 层作用是捕获输入序列中的时间依赖性，第二个 LSTM 层则是用于生成输出序列，其输入维度与输出维度一致，然后通过线性解码层将高维输出映射到原始特征空间，最后的结果输出为模型预测结果与最新观测值的残差连接。

神经网络，尤其是深度学习模型，通过从大量数据中学习来发现数据和输出结果之间复杂的非线性关系，可以帮助捕捉时间序列数据中的复杂模式和长期依赖关系，这使得其在轨迹预测精度上远远超越传统的物理分析与统计模型。然而，这也带来了模型训练和解释上的挑战。首先是因为神经网络通常被认为是一个“黑箱”，其输出结果难以用物理过程去解释和理解，其次是对数据的质量和数量有很高的依赖性，数据不足或者质量差可能导致模型表现不佳。

3.4 基于神经常微分方程网络的动态目标轨迹预测

神经微分方程是近些年比较热门的研究，自从神经常微分方程（Neural ODE）^[45]被提出以来，研究者相继提出了基于潜在状态神经常微分方程（Latent ODE）^[62]神经控制微分方程（Neural CDE）、神经随机微分方程（Neural SDE）^[63]等改进

模型以适应更复杂的场景应用。本章基于神经常微分方程构建了动态目标轨迹预测模型，其中神经常微分方程符合飞行物体运动过程中的动力学模型，体现了动态目标的物理特性。本方法突破了传统基于物理属性和受力分析的运动学模型的限制，通过深入分析轨迹数据背后的物理规律来构建预测模型，与同类模型相比具有更高的数据效率。

3.4.1 神经常微分方程网络

在对于时间序列模型的研究中，研究人员经常使用残差网络^[48]建立用于时空依赖学习的深度神经网络。但是，随着残差块数量的增加，内存成本和参数大小将显著增加难以平衡内存消耗，在对于运行速度要求较高的场景中，例如快速运动的飞行物体捕捉场景下，模型部署和应用当中变得非常困难。而神经常微分方程网络（Neural Ordinary Differential Equations, NODE）使用 ODE 求解器直接对网络隐藏状态在时间连续的条件下进行动力学建模，并使用伴随方法(adjoint method)代替传统的反向传播算法，以内存高效的方式计算梯度。

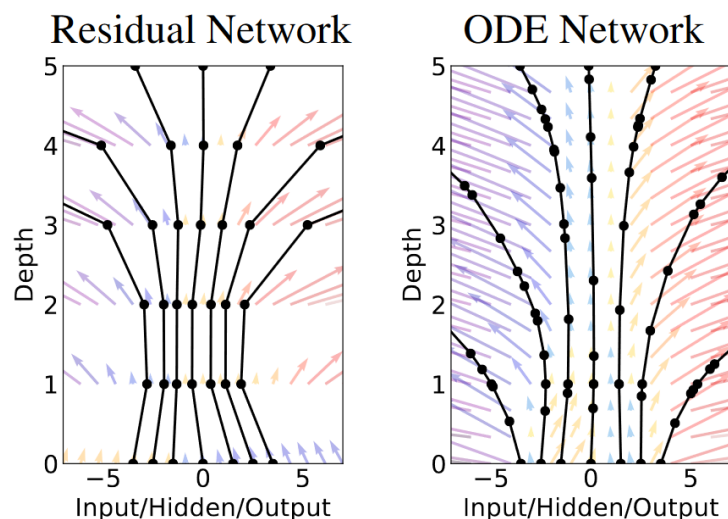
典型的残差网络结构如下式：

$$\mathbf{y}_{j+1} = \mathbf{y}_j + f_{\theta}(\mathbf{j}, \mathbf{y}_j), j \in \{0 \dots T\}, \mathbf{y}_j \in \mathbb{R}^D \quad (3-14)$$

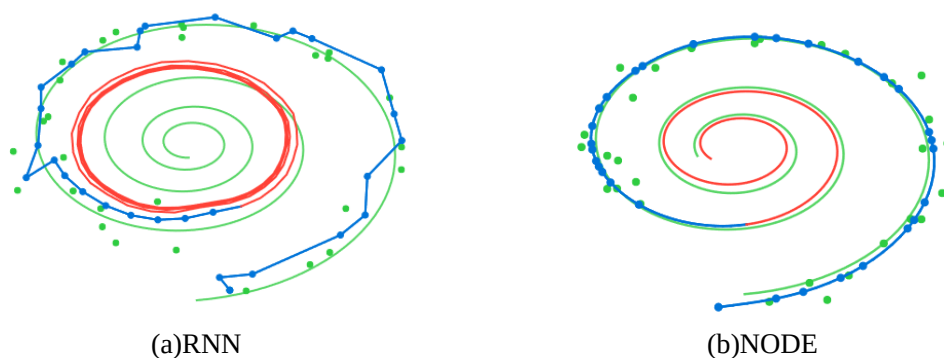
$$\frac{\mathbf{y}(t_{j+1}) - \mathbf{y}(t_j)}{\Delta t} \approx \frac{d\mathbf{y}}{dt}(t_j) = f_{\theta}(t_j, \mathbf{y}(t_j)) \quad (3-15)$$

$$\mathbf{y}(t_{j+1}) = \mathbf{y}(t_j) + \Delta t f_{\theta}(t_j, \mathbf{y}(t_j)) \quad (3-16)$$

其中 $f_{\theta}(\mathbf{j}, \mathbf{y}_j)$ 表示第 j 个残差块。公式(3-15)提供了从连续到离散的过程，通过将连续时间的导数近似为离散时间的差分，其中 Δt 表示连续时间点之间的时间间隔。公式(3-16)则是根据离散时间步长 Δt 和时间 t_j 处的状态导数 $f_{\theta}(t_j, \mathbf{y}(t_j))$ 直接计算下一个时间 t_{j+1} 的状态 $\mathbf{y}(t_{j+1})$ ，这是欧拉方法的一个表达形式，当将 Δt 放入到括号内后可以转化为(3-14)中的形式。因此可以得到残差网络是神经微分方程的连续极限形式。残差网络与神经常微分方程网络中的隐藏层如图 3-3 所示。

图 3-3 残差网络-神经微分方程隐藏层对比^[45]

其中 ResNet 结构使用一层或多层的跳跃连接，图中 Depth 表示网络深度，颜色箭头表示数据从输入到输出的流动和转换。左侧子图表明 ResNet 通过允许梯度流过跳跃连接来帮助缓解深度网络中的梯度消失问题。与具有离散层的传统网络不同，ODE 网络将深度视为连续变量。图右侧是 ODE 模型的概念图，它是一个连续深度模型。由图中可见 ODE 网络中的轨迹（彩色线）更平滑，表明其可以表示更加复杂的变换，而且需要更少的参数。类似的，神经微分方程网络在轨迹重建上也比循环神经网络（RNN）表现更佳。对比结果如下图所示。

图 3-4 循环神经网络-神经微分方程螺旋线重建外推对比^[45]。(a)循环神经网络重建外推效果；(b)神经微分方程网络重建外推效果；

其中，绿色曲线为螺旋线真值，绿色圆点为观测到的离散点，蓝色曲线为重建轨迹，红色曲线为外推轨迹。从图中可以看出，NODE 方法通过学习潜在动力学来重建和外推的螺旋线轨迹更为平滑精确，充分体现了 NODE 方法的高效性。神经微分方程是一种利用神经网络参数化向量场的微分方程，典型的例子是神经

常微分方程 (NODE), NODE 通过以下形式的微分方程来建模数据的连续动态变化。

$$\mathbf{y}(0) = \mathbf{y}_0, \frac{d\mathbf{y}}{dt}(t) = f(t, \mathbf{y}(t), \boldsymbol{\theta}) \quad (3-17)$$

其中 $\mathbf{y}(0)$ 为在时间 $t=0$ 时给定的系统的初始状态, 作为微分方程求解的起点; $\boldsymbol{\theta}$ 表示可学习的网络参数, 通常表示为多层的潜在的神经网络权重参数, 通过这样的微分操作, 将离散化的网络转变为连续, 获得任意时间步的隐藏状态; $f_{\boldsymbol{\theta}}: \mathbb{R} \times \mathbb{R}^{d_1 \times \dots \times d_k} \rightarrow \mathbb{R}^{d_1 \times \dots \times d_k}$ 代表标准的神经网络层, 通常为简单的全连接层, 用来描述系统状态随时间的变化率。

假设要求某一时刻的隐藏状态, 已知初始状态 t_0 的值 $\mathbf{y}(t_0)$, 需要预测未来 t_1 时刻的值, 可以用公式的微分方程进行计算:

$$\mathbf{y}(t_1) = \mathbf{y}(t_0) + \int_{t_0}^{t_1} f(\mathbf{y}(t), t, \boldsymbol{\theta}) dt \quad (3-18)$$

在 NODE 前向传播过程中, ODE 求解器被视为一个黑盒子, 并使用伴随法计算其梯度。伴随法通过向后求解第二个增广的 ODE 来计算梯度, 并且适用于所有的 ODE 求解器。通过伴随法进行后向传播求解, 具有较低的存储成本, 并且可以显式地控制数值误差。

假设前向传播的损失为 L , 那么前向损失可以表示为下式:

$$L(\mathbf{y}(t_1)) = L\left(\mathbf{y}(t_0) + \int_{t_0}^{t_1} f(\mathbf{y}(t), t, \boldsymbol{\theta}) dt\right) = L(\text{ODESolve}(\mathbf{y}(t_0), f, t_0, t_1, \boldsymbol{\theta})) \quad (3-19)$$

在后向传播的过程中^[61], 需要求解 L 关于 $\boldsymbol{\theta}$ 的梯度, 首先需要定义一个名为伴随状态的量: $\mathbf{a}(t) = \partial L / \partial \mathbf{y}(t)$, 其中 $\mathbf{a}(t)$ 表示在 t 的关于隐藏状态 $\mathbf{z}$, $\mathbf{a}(t)$ 是损失 L 对 $\mathbf{y}(t)$ 的导数, 它的动力学由另一个常微分方程给出(3-20), 可以被认为是链式规则的瞬时模拟, 其传播过程如公式(3-21)所示。

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^\top \frac{\partial f(\mathbf{y}(t), t, \boldsymbol{\theta})}{\partial \mathbf{y}} \quad (3-20)$$

$$\mathbf{a}(t_0) = \mathbf{a}(t_1) - \int_{t_1}^{t_0} \mathbf{a}(t) \frac{\partial f(\mathbf{y}(t), t; \boldsymbol{\theta})}{\partial \mathbf{y}(t)} dt \quad (3-21)$$

从上式中, 可以从 t_1 时刻的伴随状态计算到 t_0 时刻的伴随状态。同样地, 也可以通过该伴随状态计算整个梯度, 如公式(3-22)所示。

$$\frac{dL}{d\boldsymbol{\theta}} = \int_{t_1}^{t_0} \mathbf{a}(t)^\top \frac{\partial f(\mathbf{y}(t), t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} dt \quad (3-22)$$

至此，已经通过伴随方法完成了神经常微分方程的后向传播过程。这种方法的优势在于，它只需存储最终时间步的隐藏状态，而无需保存前向传播过程中的任何中间值，如激活函数的输出。因此，其对内存的需求极低，仅为 $O(L)$ 。此外，其计算复杂度与选择的数值解法相关，可以根据任务要求去选择合适的数值方法，在时间效率和计算精度之间做出权衡。在模型训练完成后，可以借助 ODE 求解器，利用新输入数据来预测系统在任意时间点的状态。

相比传统的系统模型，采用伴随方法的 NODE 能显著降低模型的内存消耗成本。将模型的内存消耗成本从 $O(LN)$ 减少到 $O(L)$ ，其中 L 表示模型中 ODE 或者 ResNet 的层数， N 表示每一个 ResNet 中子网络的个数或者 ODE 积分时间内的总时间步数。在 ODE 网络中，由于参数可复用，模型总参数量大约只有使用传统残差网络时的 $1/6$ ，这不仅大幅减少了模型的参数量，还能显著缩短训练时间并提高模型的鲁棒性。

3.4.2 模型框架

本章搭建的动态目标轨迹预测模型框架如图 3-5 所示，该模型主要由滤波模块、估计模块和补偿模块三个部分组成。

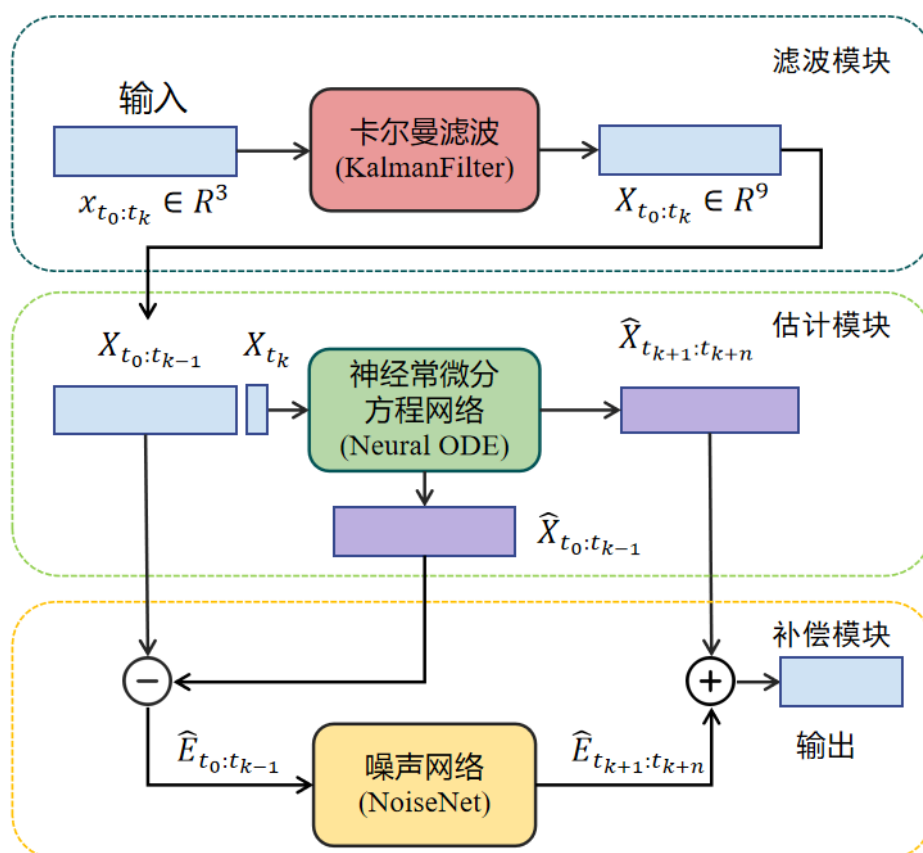


图 3-5 轨迹预测模型框架

滤波模块接收飞行物体的空间位置信息进行数据预处理后得到 9 维输入状态数据（位置、速度、加速度）作为估计模块的输入，然后通过估计模块对轨迹的潜在动力学特征进行学习，通过 NODE Slover 对采样时间点的值进行求解，生成未来的位置轨迹。为了充分利用轨迹信息，提高预测精度，在估计模块后基于编码器-解码器网络构建了补偿模块（NoiseNet）对噪声及其他非线性因素进行补偿，接下来将详细介绍每个模块。

3.4.3 估计模块

利用神经网络来参数化微分方程的解的方法，自然地模拟了时间的连续性，不仅适合处理连续时间数据和系统，同时与其他黑盒子模型不同，神经微分方程通过直接建模系统的动态行为，提供了一种更直观的方式来理解系统的内部机制，具有更好的可解释性。对于非规则的物体来说，其在做空间运动时受到的力具有高阶非线性的特点，导致物体运动时的位置变化具有一定的随机性，估计模块采用了基于 NODE 的网络结构，能够充分利用微分方程的性质去拟合非线性飞行物体动力学，提取轨迹变化趋势背后的特征。估计模块的具体结构如图 3-6 所示。

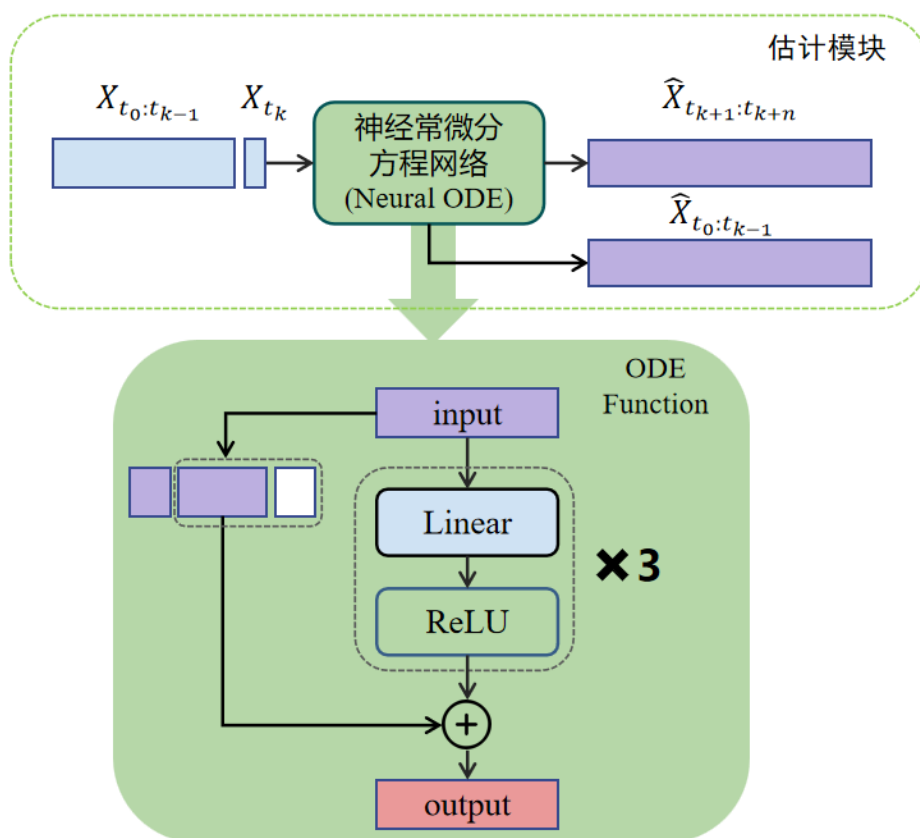


图 3-6 估计模块结构

估计模块的输入为当前观测值 $\mathbf{X}_{t_k}$ ，分别通过时间积分得到当前观测值之前的预测状态 $\hat{\mathbf{X}}_{t_0:t_{k-1}}$ 和之后的预测状态 $\hat{\mathbf{X}}_{t_{k+1}:t_{k+n}}$ 。如下式所示。

$$\begin{cases} \hat{\mathbf{X}}_{t_0:t_{k-1}} = \text{ODESolve}(\mathbf{X}_{t_k}, f_\theta, t_0, \dots, t_{k-1}) \\ \hat{\mathbf{X}}_{t_{k+1}:t_{k+n}} = \text{ODESolve}(\mathbf{X}_{t_k}, f_\theta, t_{k+1}, \dots, t_{k+n}) \end{cases} \quad (3-23)$$

$$f_\theta = \frac{d\mathbf{X}(t)}{dt} = f(\mathbf{X}(t), \theta) \quad (3-24)$$

其中 ODESolve 代表 NODE 求解器， f_θ 为图 3-6 中的 ODE Function 网络模型，其模型的输出是输入的一阶导数。在对飞行物体位置轨迹数据采集过程中，由于每次抛出的起始位置、速度、角度均不一样，即使对于同一个物体而言，也难以用一个简单的动态系统来进行描述。因此本文先通过滤波模块提取出更多的物理状态信息，包括其速度与加速度，其中由于位置的一阶导数为速度，速度的一阶导数为加速度，所以在设计 ODE Function 网络时使用了残差网络的结构，充分利用已存在的信息加快模型的收敛。该模块包含 3 个全连接层，每个全连接层用 Relu 进行正则化操作。网络定义如下：

$$f_\theta(\mathbf{X}_{t_k}) = f_w(\mathbf{X}_{t_k}) + \tilde{\mathbf{X}}_{t_k}, \quad (3-25)$$

其中 $\mathbf{X}_{t_k} = (x_{t_k}, y_{t_k}, z_{t_k}, \dot{x}_{t_k}, \dot{y}_{t_k}, \dot{z}_{t_k}, \ddot{x}_{t_k}, \ddot{y}_{t_k}, \ddot{z}_{t_k})$ 为模型输入， $\tilde{\mathbf{X}}_{t_k} = (\dot{x}_{t_k}, \dot{y}_{t_k}, \dot{z}_{t_k}, \ddot{x}_{t_k}, \ddot{y}_{t_k}, \ddot{z}_{t_k}, 0, 0, 0)$ 为 $\mathbf{X}_{t_k}$ 的估计导数， f_w 为多层的全连接网络与 Relu 正则化。对于那些具有复杂动态行为的系统，使用 NODE 可以利用深度学习与常微分方程来模拟其变化。

具体来说，历史位置轨迹数据在经过滤波模块的卡尔曼滤波后得到 $\mathbf{X}_{t_0:t_k}$ ， $\mathbf{X}_{t_k}$ 是最新的观测状态。由公式(3-17)得到，网络接收初始状态 $\mathbf{h}(0) = \mathbf{X}_{t_k}$ 作为输入，输出结果为输入状态的一阶导数，然后使用 ODE 求解器（Runge-Kutta 方法 [50]）在指定的时间区间内积分微分方程得到给定时间点的结果。

这个求解过程不需要预定义层数，而是根据问题的复杂性和所需的精度动态调整步长。因此利用 ODE 求解器能够实现任意时间点的求解，在时间序列预测任务中能够一次性完成任意时间步长的预测。最后在模型训练过程中通过 adjoint sensitivity method 实现梯度的计算。

3.4.4 补偿模块

仅仅依靠估计模块已经能够完成任意时间的位置预测，但是随着预测时间的延长，小的预测误差可以累积成大的误差，导致预测结果发散。同时由于运动物

体的随机性，在数据处理中难以进行标准化、归一化等操作，可能导致模型无法在长期预测下有效捕捉到时间序列的关键特征和模式。因此针对轨迹预测过程中发散的问题，本文设计了一个补偿模块，或者可以称为噪声估计模块，如图 3-7 所示，其核心是通过利用历史数据与基于当前观测值对历史数据的预测之间的差值来预测观测值之后的差值，将差值与估计模块的预测值进行相加。

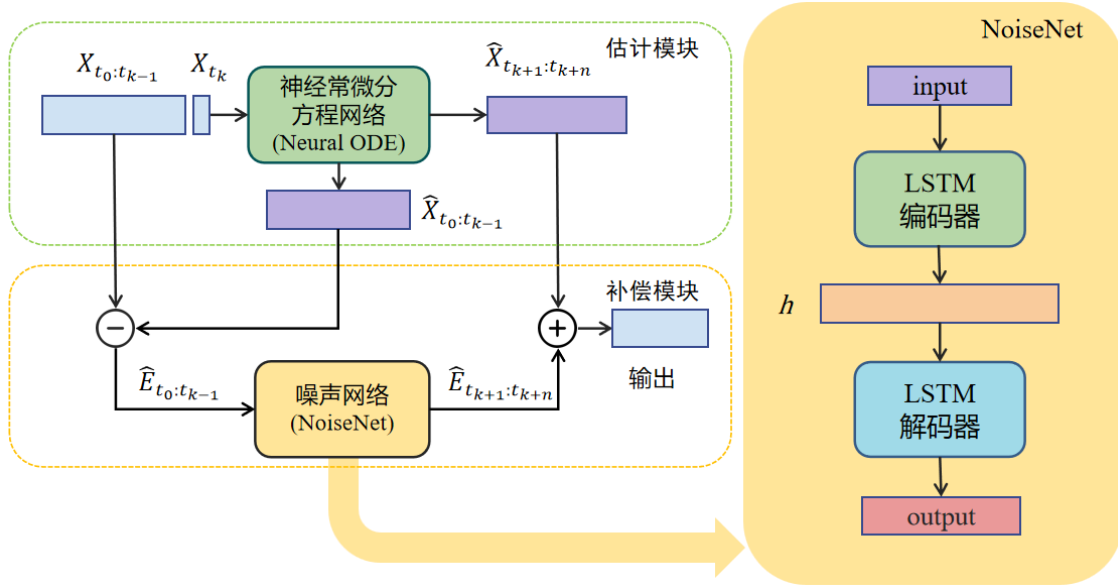


图 3-7 NoiseNet 网络结构

补偿模块算法流程如下：

$$\hat{\mathbf{E}}_{t_0:t_{k-1}} = \mathbf{X}_{t_0:t_{k-1}} - \hat{\mathbf{X}}_{t_0:t_{k-1}} \quad (3-26)$$

$$\hat{\mathbf{E}}_{t_{k+1}:t_{k+n}} = \text{NoiseNet}(\hat{\mathbf{E}}_{t_0:t_{k-1}}) \quad (3-27)$$

其中 *NoiseNet* 是由 LSTM^[49]组成的编码器-解码器网络构成，该结构常常被应用到序列到序列（seq2seq）任务中，该架构通过一个编码器网络捕捉输入序列的信息，并通过一个解码器网络生成输出序列。在该网络中，编码器部分由多个 LSTM 单元组成，其目的是将输入序列转换成固定长度的上下文向量。每个 LSTM 单元接收当前时刻的输入以及前一时刻的隐状态和细胞状态，输出当前时刻的隐状态和细胞状态。隐状态用于传递给下一个 LSTM 单元，而细胞状态负责在单元内部传递重要的信息，帮助网络记住或忘记信息。编码器处理完整个输入序列后，编码过程最后产生的状态被视为对输入序列的压缩表示，包含了序列中的关键信息。

$$h_t^{\text{enc}}, c_t^{\text{enc}} = \text{LSTM}_{\text{enc}}(x_t, h_{t-1}^{\text{enc}}, c_{t-1}^{\text{enc}}) \quad (3-28)$$

其中 x_t 是在时间 t 的输入, LSTM_{enc} 是编码器 LSTM 的一个单元, h_t^{enc} 和 c_t^{enc} 是编码器在每个时间步 t 的隐藏状态和细胞状态。编码器对输入的误差状态 $\hat{E}_{t_0:t_{k-1}}$ 进行编码得到隐藏状态 $h_{t_{k-1}}^{enc}$, 根据预测的序列长度对 $h_{t_{k-1}}^{enc}$ 进行堆叠得到隐藏状态 h 。

解码器同样由多个 LSTM 单元组成, 其接收编码器输出的堆叠隐藏状态 h 来生成输出序列得到高维的隐藏状态 h^{dec} 。解码器的第一个 LSTM 单元接收上下文向量作为输入, 每个 LSTM 单元基于当前的输入、隐状态和细胞状态生成输出, 并更新这些状态以传递给下一个单元。这个过程一直持续到生成全部。

$$h_t^{dec}, c_t^{dec} = \text{LSTM}_{dec}(y_{t-1}, h_{t-1}^{dec}, c_{t-1}^{dec}) \quad (3-29)$$

其中 y_{t-1} 是在前一个时间步生成的输出, LSTM_{dec} 是解码器 LSTM 的一个单元, h_t^{dec} 和 c_t^{dec} 是解码器在每个时间步 t 隐藏状态和细胞状态。解码器最终将所有时刻输出的隐藏状态 h^{dec} 进行降维得到 $\hat{E}_{t_{k+1}:t_{k+n}}$ 。最终, 飞行物体位置预测模型的输出为:

$$output = \hat{E}_{t_{k+1}:t_{k+n}} + \hat{X}_{t_{k+1}:t_{k+n}} \quad (3-30)$$

3.5 实验及分析

3.5.1 数据集构建

本实验的训练数据集为第二章节所提及的 6 种非规则物体轨迹数据。这些轨迹数据由离散的位置坐标点组成, 每个坐标点记录了采样时刻和位置信息, 对于第 i 条轨迹在 t 时刻的空间坐标可以表示为 $P_t^i(x_t, y_t, z_t)$, 在经过卡尔曼滤波后得到九维的状态表示为 $X_t^i(x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \ddot{x}_t, \ddot{y}_t, \ddot{z}_t)$ 。

整个轨迹数据集共包括 6 类物体, 总共 1500 条轨迹。为了解决类别不平衡的问题, 在训练前, 将每类物体的 80% 数据作为训练集, 剩余 20% 用作测试集以评估模型的效果。在训练过程中, 并没有采取任何数据增强或数据扩增的措施。由于轨迹长度的不一致性, 为了在网络模型训练中实现批处理以加快训练速度, 需要对轨迹数据进行分割, 值得注意的是, 数据划分仅适用于训练数据集, 测试数据集长度则保持不变。数据划分流程如图 3-8 所示。

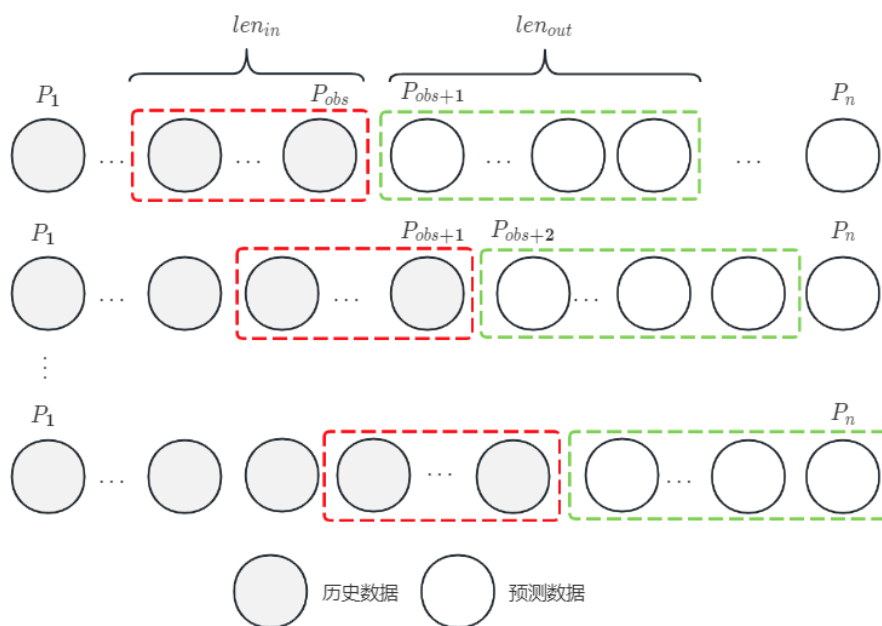


图 3-8 数据划分流程

其中模型的输入长度设为 len_{in} ，见图中红色虚线框，输出长度设为 len_{out} ，见图中绿色虚线框。在数据划分过程中，滑动窗口不断向后移动直到覆盖整个数据轨迹。数据划分将从整个数据集中通过滑动窗口技术提取连续的输入和输出数据序列，以用于网络模型的批次训练。

3.5.2 评价指标

在对位置轨迹预测的研究过程中，有许多常见的评估指标，比如采用真实值与预测值的欧式距离作为评价指标，这种方法能直观地体现预测效果。Kim 等^[11]为不同的算法定义了一个评估标准，称为平均领先时间（Average Leading Time, ALT），即如何在笛卡尔空间中保持 1 厘米精度的持续时间。为了对网络的性能进行多维度客观的评价分析，本文同时采用平均位移误差（Average Displacement Error, ADE）、最终点位移误差（Final Displacement Error, FDE）以及 Average Leading Time 作为评估指标来开展测试。假设某个物体有 m 条测试轨迹，轨迹长度为 L_m ，模型预测长度为 k ，模型输入长度为 $L_m - k$ ，真实轨迹坐标为 $\mathbf{X}_t^i = (x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \ddot{x}_t, \ddot{y}_t, \ddot{z}_t), (0 \leq i \leq N, 0 \leq t \leq M)$ ，预测轨迹数据为 $\hat{\mathbf{X}}_t^i = (\hat{x}_t, \hat{y}_t, \hat{z}_t, \hat{\dot{x}}_t, \hat{\dot{y}}_t, \hat{\dot{z}}_t, \hat{\ddot{x}}_t, \hat{\ddot{y}}_t, \hat{\ddot{z}}_t), (0 \leq i \leq N, 0 \leq t \leq M)$ ，则有以下表示：

平均位移误差是衡量轨迹序列中所有预测值与其对应真实值之间欧氏距离的平均值，用于描述预测轨迹的整体精确度，表示如下式：

$$ADE = \frac{1}{m} \sum_{i=1}^m \frac{1}{k} \sum_{t=L_m-k}^{L_m} \sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2 + (z_t - \hat{z}_t)^2} \quad (3-31)$$

最终点位移误差专注于每个轨迹段终点的预测值与实际值之间的欧氏距离平均值，反映了预测轨迹结束位置的准确性，表示如下式：

$$FDE = \frac{1}{m} \sum_{i=1}^m \sum_{t=L_m}^{L_m} \sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2 + (z_t - \hat{z}_t)^2} \quad (3-32)$$

平均领先时间是衡量系统在保持至少 1 厘米精度的条件下，预测轨迹的平均时间长度，是评价预测模型前瞻性能的一个指标，表示如下式：

$$\sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2 + (z_t - \hat{z}_t)^2} \leq 0.01, t \leq T_{ALT} \quad (3-33)$$

3.5.3 网络训练

在动态目标抓取任务的轨迹预测中，预测网络必须能够在极短时间内对未来的轨迹进行准确预测，以确保机械臂有足够的时间进行响应和运动。为了在时效性和模型效率之间找到合适的平衡点，在经过多次实验后将模型超参数设置如表 3-1 所示。

表 3-1 模型超参数设置

超参数	数值
输入时间长度	30
预测时间长度	30
输入特征维度	9
隐藏层维度	128
Batch_size	8
学习率	0.001

通过对采集的数据进行统计分析发现，目标从开始被抛出到离开记录区域的时间大约为 0.8 秒。实验中将输入时间长度设置为 30 帧，考虑到数据采集的频率为 100Hz，这意味着在物体被抛出 0.3 秒后系统才开始预测目标位置，留给机械臂的运动时间约为 0.5 秒，而这段时间足以让机械臂从起始位置移动到工作空间内的大部分位置。

在给定的超参数设置下，模型在经过 20 个 epoch 训练后，6 种非规则物体训练 loss 下降情况如图 3-9 所示。

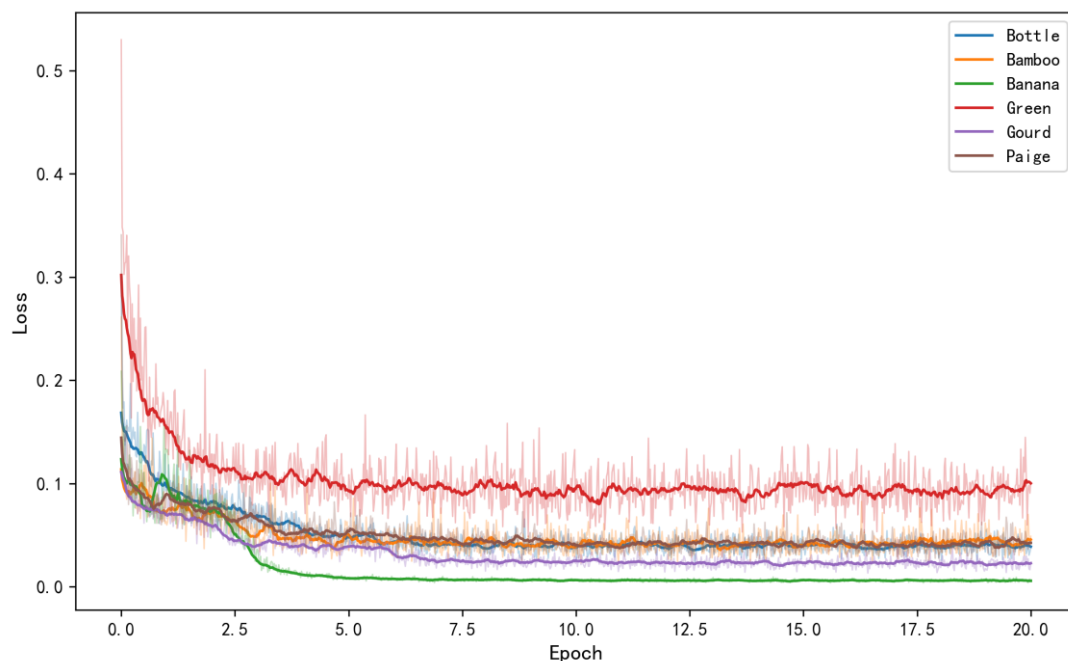


图 3-9 训练损失曲线

一开始曲线的起始点损失较高，随着训练进行，学习速度会在每一个 epoch 后乘一个衰减因子 $\alpha = 0.5$ ，随着损失的减少，学习速度逐渐放缓，损失在后续的训练中趋于平稳，最终达到收敛。图 3-10 展示了训练过程的平均损失曲线。

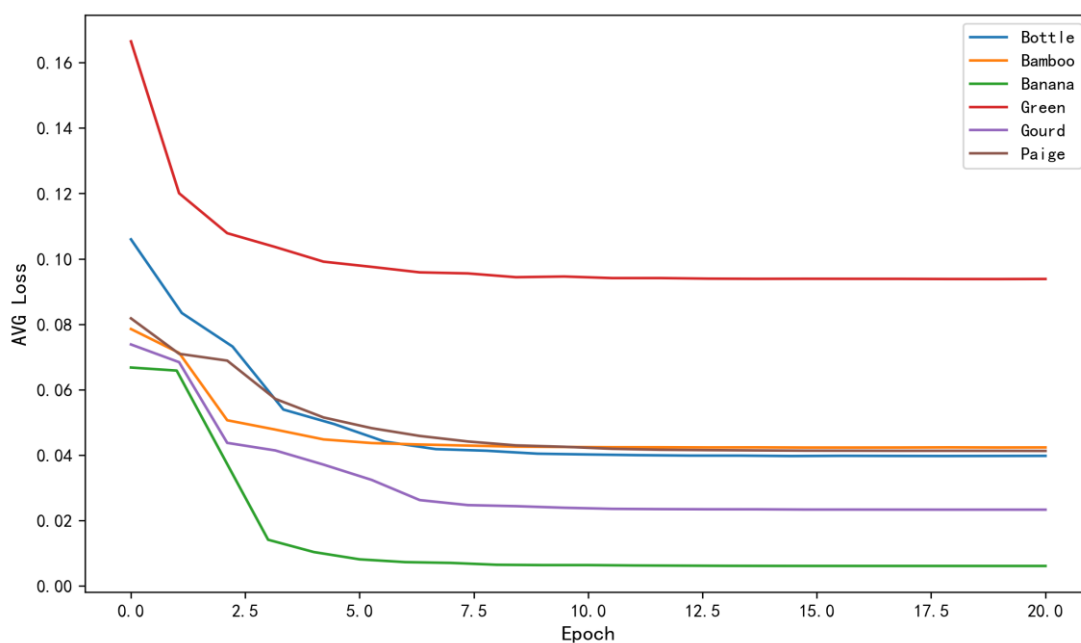


图 3-10 平均损失曲线

通常，平均损失更能平滑地反映模型训练的整体趋势。图中所有类别的平均损失随着时间的推移均显著下降，说明模型对所有物体的预测能力都在提高。

3.5.4 结果分析

完成上述网络的训练后，本研究使用剩余的 20%测试数据对神经微分方程（NODE）模型及其它三个基准模型进行了验证。验证结果包括三个评价指标和一个执行效率的对比。

四种方法轨迹预测的平均位移误差如表 3-2 所示。

表 3-2 平均位移误差对比

	Bottle(half)	Bamboo	Banana	Green	Gourd	Paige
Newton	0.34±0.06	0.49±0.07	0.36±0.10	0.50±0.21	0.51±0.07	0.58±0.12
GLS	0.08±0.03	0.09±0.02	0.09±0.04	0.22±0.11	0.12±0.03	0.20±0.06
NODE	0.02±0.01	0.02±0.01	0.01±0.01	0.06±0.03	0.02±0.01	0.03±0.01
ResLSTM	0.06±0.02	0.07±0.03	0.01±0.01	0.09±0.05	0.04±0.01	0.06±0.03

其中，Newton 方法的误差相对较高，这说明基于传统物理学的预测方法可能不足以捕获复杂的飞行动力学。GLS 方法误差虽然低于 Newton 方法，但仍然高于 ResLSTM 和 NODE 方法，这表明传统的统计方法虽然能够提供一定程度的预测准确性，但仍然无法与基于深度学习的方法相匹敌。NODE 方法在所有物体上均显示出最小的误差，证明了其在精确预测飞行物体轨迹方面的优势。ResLSTM 方法也展现了较低的位移误差，虽然略高于 NODE，但仍然比 Newton 和 GLS 方法好很多，表明深度学习方法在处理此类问题时具有明显的优势。

四种方法轨迹预测的终点平均位移误差统计如表 3-3 所示。

表 3-3 终点平均位移误差对比

	Bottle(half)	Bamboo	Banana	Green	Gourd	Paige
Newton	0.72±0.19	1.25±0.22	0.96±0.36	1.55±0.74	1.41±0.27	1.73±0.43
GLS	0.25±0.1	0.31±0.08	0.30±0.15	0.75±0.4	0.41±0.11	0.68±0.2
NODE	0.06±0.03	0.06±0.03	0.03±0.02	0.17±0.11	0.05±0.02	0.09±0.04
ResLSTM	0.18±0.07	0.20±0.08	0.04±0.03	0.25±0.13	0.10±0.04	0.18±0.08

终点平均位移误差是动态目标的抓取任务中轨迹预测最重要的评估标准。这是因为轨迹预测系统必须要在物体开始移动的瞬间迅速预测出其未来的运动轨迹，而预测的终点精度直接影响到机械臂能否准确抓取到移动中的物体。其结果与平均位移误差类似，其中 NODE 方法效果最优，最高达到了 3cm 的预测精度，在所有类别上的平均预测精度相比 ResLSTM 提高了约为 48.94%。

四种方法维持 1 厘米精度的平均领先时间对比如表 3-4 所示。

表 3-4 平均领先时间对比

	Bottle(half)	Bamboo	Banana	Green	Gourd	Paige
Newton	0.01±0.00	0.01±0.00	0.02±0.00	0.02±0.00	0.01±0.00	0.01±0.00
GLS	0.18±0.04	0.19±0.03	0.15±0.03	0.12±0.03	0.18±0.02	0.13±0.02
NODE	0.30±0.18	0.34±0.16	0.55±0.22	0.18±0.17	0.5±0.22	0.34±0.22
ResLSTM	0.27±0.07	0.31±0.04	0.48±0.15	0.28±0.03	0.34±0.04	0.31±0.05

平均领先时间可以视作模型预测准确性持续时间的指标。其中，Newton 方法的领先时间极短，这表明其预测准确性迅速下降。GLS 方法提供了较长的领先时间，但仍远低于基于深度学习的方法。NODE 方法在多数物体上提供了最长的领先时间，这意味着它能够长时间维持较高的预测精度。ResLSTM 方法也表现出了较长的领先时间，虽然在某些物体中略低于 NODE，但整体上仍表现出色，显示了其泛化能力和长期预测的准确性。

四种方法的执行时间对比如表 3-5 所示。

表 3-5 模型执行时间

模型	数据集数量	执行时间(ms)	频率
Newton	-	1.42e-6	>1000Hz
GLS	-	2.05e-6	>1000Hz
NODE	1200	0.0028	>1000Hz
ResLSTM	7200	0.0016	>1000Hz

在运行时间方面，Newton 和 GLS 算法的执行时间非常短，远远低于 NODE 和 ResLSTM，这表明传统算法在计算效率上具有优势。尽管基于深度神经网络的 NODE 与 ResLSTM 模型的执行时间长于 Newton 和 GLS 方法，但其预测准确性远远优于这两种方法，且均能满足实时性的要求。ResLSTM 模型处理的数据集数量为 7200，是 NODE 模型的 6 倍，虽能保持较好的精度，但是需要对训练数据进行额外的数据增强处理。NODE 模型在训练数据集数量为 1200 的情况下，取得了优于 ResLSTM 的效果，充分体现了基于 NODE 的数据高效性。

为了评估模型泛化性能，本章用每一类物体数据训练出来的模型对其他物体数据进行评估。评估结果如图 3-11 所示。

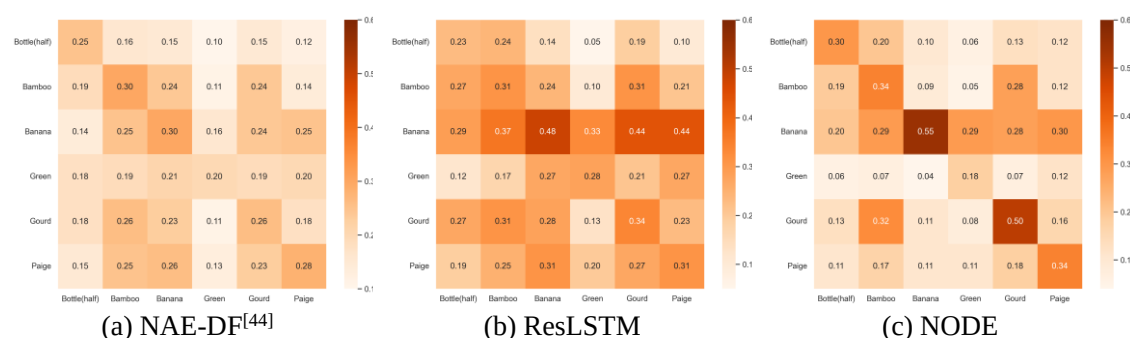


图 3-11 领先时间对比

其中，NAE-DF 方法通过结合神经加速度估计器（NAE）网络与可微分滤波器实时估计物体运动轨迹中的变化加速度，实现预测和测量的最优融合。三种方法领先时间评估结果对比表明本文所提出的 NODE 网络对于物体本身类别的数据验证效果最好，能够在少样本的情况下实现高精度的轨迹预测，但是在泛化性上略差于其余两种方法。

通过综合分析上述图表和表格数据，可以得出以下结论：

（1）深度学习方法的有效性：

与传统物理和统计方法相比，基于深度学习的 NODE 和 ResLSTM 模型在所有评价指标上均展现出了更好的性能。这强调了深度学习技术在理解复杂运动模式和动力学系统方面的强大能力，尤其是在需要从大量数据中自动学习特征和模式的场景下。

（2）NODE 的卓越性能：

在动态目标轨迹预测任务中，基于神经常微分方程（NODE）的模型表现出了显著的优势。它在平均位移误差、最终点位移误差以及平均领先时间等多个评价指标上均超越了其他方法，包括传统的牛顿动力学方法、基于最小二乘法的统计方法以及基于循环神经网络的 ResLSTM 深度学习方法。这证明了 NODE 方法在捕获和预测复杂飞行轨迹动力学方面的有效性和准确性。

（3）NODE 的数据高效性：

尽管 ResLSTM 模型在泛化能力方面表现出色，能够有效地在不同物体之间迁移学习，但它所需的学习样本数量是 NODE 模型的 6 倍。这一特点使得 ResLSTM 在样本收集困难的场景下应用受限。相比之下，NODE 模型由于对物理模型的适应性，能够在更少的样本量上实现有效的学习。

实验结果清楚地表明，在处理复杂和非线性的飞行物体轨迹预测问题时，基于神经网络的方法（NODE 和 ResLSTM）远远超越了传统的牛顿动力学和最小二乘法拟合方法。这些传统方法虽然在特定条件下有效，但在处理高度动态和非线

性系统时，存在着很大的局限性。其中基于深度学习的方法（NODE）利用神经网络模型学习物体运动的内在规律，克服了传统运动学模型在处理复杂运动物体时的局限性，在轨迹预测任务上展现了出色的性能。

3.6 本章小结

本章针对传统预测算法在非规则物体在轨迹预测中遇到的问题，如运动学模型的建立困难、模型参数难以辨识以及缺乏泛化性等，通过深入分析物体在运动中变化的受力情况，提出了基于神经常微分方程网络的轨迹预测方法，通过 NODE 学习潜在的物体运动物理信息估计未来轨迹，并通过噪声估计网络对估计结果进行补偿。在多种非规则物体的轨迹数据集上进行实验验证表明，与传统的牛顿动力学模型、最小二乘法拟合以及基于 ResLSTM 的序列预测模型相比，本文的提出的 NODE 轨迹预测方法在平均位移误差、最终点位移误差以及领先时间等多个评价指标上均取得了最优的预测性能。这些实验结果不仅验证了所提出模型的有效性，也展示了深度学习技术与物理模型相结合的强大潜力。

第四章 基于模仿学习的机械臂敏捷运动控制

4.1 引言

在上一章节中，已经深入讨论了基于深度神经网络的动态目标轨迹预测技术，成功地实现了对飞行物体轨迹的高精度预测。在获取目标位置后，机械臂需要快速而精确地规划并执行抓取动作，这要求高度优化的算法和控制系统以减少延迟并提高执行精度。在人机共享环境中，确保动态抓取操作的安全性对于保护人员和设备至关重要。系统必须能够在紧急情况下迅速做出反应，以避免伤害和损失。为了实现机械臂的敏捷稳定运动控制，本章提出了一种基于模仿学习的方法来构建一个稳定变分动态系统（SVDS）。通过引入模仿学习，机器人可以在短时间内快速学习新的运动技能，为机械臂在动态环境中的运动控制提供了新的解决方案。

4.2 模仿学习

与机器人相比，人类能够轻松抓住向他们抛出的物体，在抓取运动物体时展现出卓越的适应性和反应能力，但这仅限于物体运动速度较低且姿态稳定的情况。通常来说，人类的反应时间大约在 0.2 秒到 0.3 秒之间^[51]，这一数值可能会因不同条件而有所变化。对于简单的刺激（如膝跳反射），反应时间大约为 50 毫秒；而对于需要进行决策的复杂刺激，反应时间可能会更长。为了满足高动态性抓取任务的要求，并以最小成本实现任务场景的迁移，本文综合考虑了人体的运动灵活性和机器人的强大计算能力，采用基于模仿学习的方法来学习机械臂的运动控制。

模仿学习通常用于机器人运动技能的学习，又称示教学习或示教编程，是机器学习和人工智能领域的一个重要分支。其核心思想是通过观察并模仿人类的行为或专家的动作来学习特定的任务或技能。这一学习过程通常涉及从单个或少量的示教轨迹中提取关键的运动特征，然后将这些特征应用或泛化到新的情境中，从而使得机器人在面对未知环境或任务时展现出良好的自适应性。

模仿学习的核心思想是减少机器人执行任务时所需的编程工作量，使其能够更自然、更灵活地适应各种环境和任务。在实际操作中，示教者直接引导重力补偿下的机器人进行示教演示。在这个过程中，机器人利用自身装备的传感器、运动学模型和视觉系统来记录各种数据，如关节的角度、机械臂末端的位置和方向、所施加的力和力矩，以及周围环境的状态。之后，这些数据被用于模仿学习算法，

以便机器人能够学习并复现示教过程中的技能。图 4-1 展示了示教者通过控制引导机器人进行示教演示。

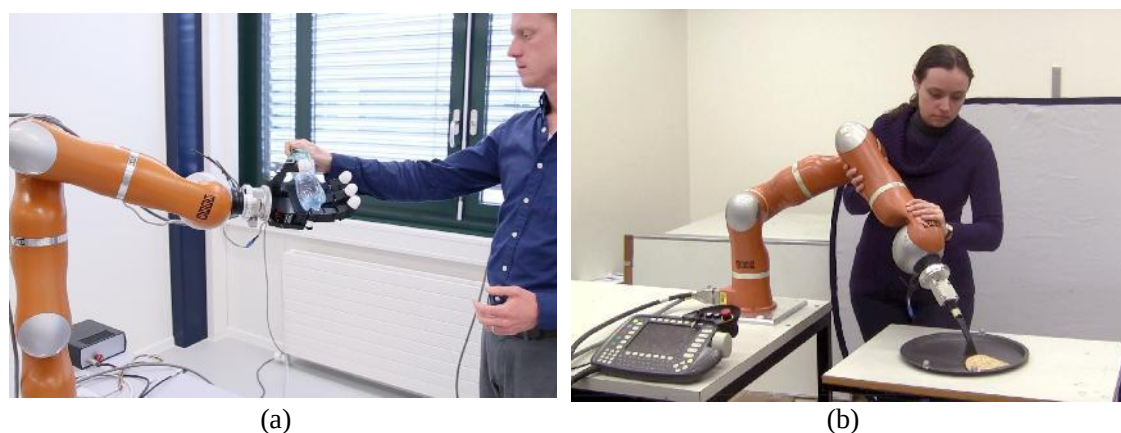


图 4-1 示教数据采集。示教过程 1；示教过程 2；

模仿学习可以分为三个关键阶段，包括示教、模型学习、执行。其中，示教是模仿学习的初始阶段，专家或教师可以通过直接操纵机器人、使用传感器记录人的动作、或是通过虚拟环境进行。在模型学习部分，传统方法通常采用非线性回归算法（如高斯混合模型 GMM，高斯过程回归 GPR，支持向量回归 SVR，局部加权回归 LWR 等）将示教数据建模为一个动态系统。而在深度学习领域，则常通过神经网络，如卷积神经网络（CNNs）或循环神经网络（RNNs），进行轨迹拟合或行为克隆。这一步骤的目的是从示教数据中抽象出一般性规则，使得机器人能够新的情境下重现学习到的行为。最后，机器人将学习到的模型应用于实际任务中，执行所学的动作或行为。这一阶段的成功直接取决于之前两个阶段的效果，尤其是模型学习阶段的准确性和泛化能力。通过示教学习降低了算法搜索空间的复杂度，算法学习时间快，需要的样本数据少，极大地提高了学习效率。

模仿学习也存在着一些问题与挑战，例如：高自由度机器人在人拖动下很难产生协调连续的动作，对于复杂任务难以示教；还有过渡依赖于示教对象和平台，任务完成的准确度直接取决于示教的质量；尽管存在这些挑战，模仿学习仍是机器人学习运动技能的一种有效方法，特别是在快速学习和执行简单任务方面。

在机器人技能学习的任务中，动态系统方法简化了学习算法的设计，将复杂行为分解为更简单动态组件，减少学习任务的复杂性，提高学习效率。通过构建时不变动态系统能够使得机器人运动适应环境变化和内部参数的变动，增强了机器人在实时环境中快速响应的能力。因此在本文的机械臂敏捷运动控制中采用了动态系统的框架，并通过与深度学习结合的方法来提升系统的性能。

4.3 动态系统理论分析与系统构建

4.3.1 动态系统理论介绍

动态系统(Dynamical system, DS)为机器人技能建模提供了一个强大的框架,它特别适用于生成机器人的运动轨迹^[54]。动态系统方法的核心在于其能够基于当前状态和环境的实时观测来生成机器人的下一步动作,而不是预先规划出一条完整的路径。这种方法允许机器人对路径进行实时重规划,并具备对环境干扰进行即时响应的能力。动态系统可以通过一个非线性连续可微函数来定义:

$$\dot{\xi} = f(\xi), f: \mathbb{R}^d \mapsto \mathbb{R}^d \quad (4-1)$$

其中在机器人运动学和动力学中, $f(\xi)$ 是一个非线性连续可微函数,用于编码特定行为, ξ 表示当前的状态(位置、速度、加速度等), $\dot{\xi}$ 表示状态的导数,与输入的维度相等,可以作为系统的控制量。对于一些简单的运动可以采用 GMM、GPR 等机器学习回归算法构建动态系统,但这些回归技术在训练过程没有考虑系统的稳定性问题,所得到的动态系统模型很可能是不稳定的或只能局部收敛。因此,为了构造全局稳定的动态系统,需要对所得到的回归模型进行稳定性分析。

对于动力学不确定的系统,基于学习的方法通过对数据的不确定性进行建模来代替传统的鲁棒控制器。为这种学习算法提供理论上的安全保证是应用基于学习的方法来控制现实世界安全关键系统的核心。这些算法已经促使研究界考虑几种方法,以确保物理系统的安全,其中基于学习的算法被用来寻找控制策略

4.3.2 动态系统稳定性分析

对于动力学不确定的系统,通过学习动态系统的方法能够对数据的不确定性进行建模来代替传统的鲁棒控制器。动态系统通常是基于一组演示轨迹的数据通过使用高斯过程回归、局部加权映射回归算法或混合高斯回归等回归算法来构建。但如果仅仅是构建了回归模型而没有考虑其稳定性,可能会由于系统参数设置不当、非线性效应、外部扰动等因素导致系统出现发散现象,导致系统状态随时间变化,从其初始状态或平衡点远离,向无界增长或复杂、不可预测行为演化,对于实物机器人运动控制来说这是异常危险的。因此,在实际应用场景中必须进行安全稳定设计与分析。在机器人领域内,已有许多研究者提出了各种学习算法来寻找控制策略,以确保物理系统的安全^[65,66]。Lyapunov 在 1892 年提出了两种

方法（称为第一法和第二法）^[55,56]，用于确定由常微分方程描述的动力学系统的稳定性。

Lyapunov 第一法（间接法）包括了利用微分方程显式解进行系统分析，基本思路是：求解系统的微分方程式，根据解的性质或特征方程的根的情况来判据稳定性。系统特征方程为负实数根或负实部的虚根，则系统稳定。对于非线性系统，只能根据在平衡点近似线性化的方程来研究，先将非线性系统线性化，然后计算线性化方程的特征值，判定原非线性系统的稳定性。

对于非线性系统和线性时变系统，求解状态方程通常十分困难，而采用 Lyapunov 第二法不需要求解运动微分方程，也不需要求解特征值，而是通过定义一个叫做李雅普诺夫函数的标量函数 $V(x)$ 来从能量的角度分析系统的稳定性。这个函数通常构造为系统状态的函数，其值反映了系统在特定状态下的能量水平。在 Lyapunov 第二法中，直接根据 $V(x)$ 和其对时间的导数 $\dot{V}(x) = dV(x)/dt$ 的定号性，判断平衡状态处的稳定性、渐近稳定性或不稳定性，而不必直接求出方程的解。为了对动态系统控制策略的稳定性进行分析，本文采用 Lyapunov 第二法对动态系统稳定性进行分析，其中有如下的两个定理：

定理 1 (渐近稳定性) 如果存在一个连续可微的李亚普诺夫候选函数 $V(\xi): \mathbb{R}^d \rightarrow \mathbb{R}$ 满足以下条件，则该系统在原点处的平衡状态是大范围渐近稳定性的：

$$\begin{aligned} V(\xi) \text{ 为正定: } & V(\xi) > 0, V(\xi^*) = 0 \\ \dot{V}(\xi) \text{ 为负定: } & \dot{V}(\xi) < 0, \dot{V}(\xi^*) = 0 \\ \text{当 } \|\xi\| \rightarrow \infty \text{ 时 } & V(\xi) \rightarrow \infty \end{aligned} \quad (4-2)$$

定理 2 (指数稳定性) 在动态系统满足公式(4-2)的基础上，如果这个李亚普诺夫候选函数同时满足以下条件，那么该系统在该平衡点是全局指数稳定的。

$$\begin{aligned} \dot{V}(\xi) &\leq -\alpha(V(\xi)) \\ c_1 \|\xi\|^2 &\leq V(\xi) \leq c_2 \|\xi\|^2 \end{aligned} \quad (4-3)$$

指数稳定性要求 Lyapunov 函数的导数（即随时间的变化率） $\dot{V}(\xi)$ 是负的，并且至少以一定比例 α 减小，其中 α 是正常数。这表示系统的指数稳定性表示系统的能量或者误差会随着时间指数级地减小，指向系统的稳定状态。并且 Lyapunov 函数被系统状态的二范数的平方所界定，其中 c_1, c_2 是正常数，这个条件保证了 Lyapunov 函数在状态向量的零向量附近是正的，并且在状态偏离零向量时增加，这与系统状态的能量或者偏离程度相关。这两个条件一起保证了无论系统开始于状态空间中的哪一点（除了平衡点 ξ^* ）系统状态都将随着时间的推移而向

平衡点指数级地收敛。这意味着系统的平衡点是稳定的，并且任何小的扰动都会随着时间的推移而衰减，最终消失。

4.3.3 动态系统构建

现实中存在某些系统，在没有对系统的进行干预的情况下，无论如何都不能保证系统是渐近稳定的，即系统轨迹是无法到达平衡点的。如果能够找到一个控制输入去改变系统状态的轨迹，从而使得系统轨迹能够到达平衡点，那就能解决这个问题。因此，考虑在原本的系统状态中引入控制输入，以持续减少运动的能量，直到最终收敛到预定的吸引子上。这种方法也叫做控制李雅普诺夫函数 (Control Lyapunov Function, CLF) [67]。CLF 常被用于机器人的在线运动控制，确保在存在动态不确定性、执行器约束和环境中有障碍物的情况下保证系统的稳定性和安全性[68]。CLF 是连续可微且正定的 (即对于所有非零点 ξ 都为正值)，并且对于所有状态 ξ ，都存在一个控制 u 使得函数的时间导数小于零。这意味着对于每个状态 ξ ，我们总能找到一个控制 u 来减少系统的“能量”，理论上这可以使系统能量渐近地趋向于零，从而实现系统的稳定[64]。为了防止出现非期望的收敛行为，本文基于 CLF 构建机器人运动控制的动态系统，因此，动态系统的表达式由初始形式变为包含稳定命令的形式。

$$\dot{\xi} = f(\xi) = \hat{f}(\xi) + u(\xi), \quad u(\xi) \in \mathbb{R}^d \quad (4-4)$$

其中 $\hat{f}(\xi)$ 是动态系统的估计项， $u(\xi)$ 是对估计动态系统估计值产生的稳定项。稳定项应该提供一个合适的修正值，这样在除了目标位置之外，均有一个沿着能量函数梯度 $\nabla V(\xi^0)$ 的负方向上的分量，条件如下：

$$(\nabla_{\xi} V(\xi))^T (\hat{f}(\xi) + u(\xi)) < 0, \quad \forall \xi \in \mathbb{R}^d \setminus \xi^* \quad (4-5)$$

$$(\nabla_{\xi} V(\xi))^T (\hat{f}(\xi) + u(\xi)) = 0, \quad \xi = \xi^* \quad (4-6)$$

由于稳定命令是一个多维向量，因此上式不只有一组解，需要在其中找到最优解。对于除了稳态点 ξ^* 之外的所有状态 ξ ，李雅普诺夫函数的梯度 $\nabla_{\xi} V(\xi)$ 与 $\hat{f}(\xi) + u(\xi)$ 内积是负的。这意味着，系统能量总是沿着稳态点的方向减少，通过添加满足条件稳定项 $u(\xi)$ 能够保证动态系统的渐近稳定性。

由于标准的深度学习技术在训练期间没有考虑动力学模型的稳定性，所以这个系统可能是不稳定的，或者会收敛到一个假的吸引子。因此，本章先基于神经网络从机械臂示教数据集 D 中学习估计系统 $\hat{f}(\xi)$ 。然后引入了同样由神经网络

构建的附加的稳定项 $u(\xi)$ 来确保 $\hat{f}(\xi) + u(\xi)$ 除了在目标点 ξ^* 外，沿 $V(\xi)$ 梯度的负方向有分量，即满足指数稳定性，约束如下式：

$$\nabla V(\xi)^T \hat{f}(\xi) \leq -\alpha V(\xi), \forall \xi \in \mathbb{R}^d \setminus \xi^* \quad (4-7)$$

其中 $\nabla V(\xi) = \partial V(\xi) / \partial \xi$ 。为了使动态系统满足(4-7)，可以构建了一个受约束的优化问题如下式：

$$\begin{aligned} \min_{u(\xi)} & \frac{1}{2} u(\xi)^T u(\xi) \\ \text{s.t.} & \nabla V(\xi)^T (\hat{f}(\xi) + u(\xi)) \leq -\alpha V(\xi) \end{aligned} \quad (4-8)$$

其中优化的目的是为了找到一个控制项 $u(\xi)$ ，这个控制项使得整体系统满足 Lyapunov 函数的导数小于或等于 $-\alpha V(\xi)$ ，这是确保系统稳定性的条件。又因为公式(4-8)给出的优化问题是凸的，意味着它有一个唯一的全局最小值，这个最小值可以解析确定，公式(4-9)提供了解的表达式。

$$\begin{aligned} u(\xi) &= \begin{cases} 0, & \text{if } \nabla V(\xi)^T \hat{f}(\xi) \leq -\alpha V(\xi) \\ -\nabla V(\xi) \frac{\nabla V(\xi)^T \hat{f}(\xi) + \alpha V(\xi)}{\|\nabla V(\xi)\|^2}, & \text{otherwise} \end{cases} \\ &= -\nabla V(\xi) \left(\frac{\text{ReLU}(\nabla V(\xi)^T \hat{f}(\xi) + \alpha V(\xi))}{\|\nabla V(\xi)\|^2} \right) \end{aligned} \quad (4-9)$$

其中 ReLU 激活函数的使用是为了确保当 $\nabla V(\xi)^T \hat{f}(\xi) < -\alpha V(\xi)$ 时 $u(\xi)$ 是非零的，这意味着当系统不满足稳定性条件时，控制项 $u(\xi)$ 将被激活以推动系统朝向满足条件的方向调整。这个优化方案是为了找到一个控制项 $u(\xi)$ ，它在需要时能够确保系统的稳定性，同时尽可能地减少其干预的强度。最后动态系统的构造如下：

$$\dot{\xi} = \hat{f}(\xi) - \nabla V(\xi) \left(\frac{\text{ReLU}(\nabla V(\xi)^T \hat{f}(\xi) + \alpha V(\xi))}{\|\nabla V(\xi)\|^2} \right) \quad (4-10)$$

对所有 $\xi \in \mathbb{R}^d \setminus \{\xi^*\}$ ，如果估计项 $\hat{f}(\xi)$ 符合条件，则不做修改返回。如果它违反了稳定性条件，将对其进行最小化修改，以使得修正后的模型 $f(\xi)$ 刚好满足等式条件。

4.4 稳定变分动态系统

在本章中将结合控制李雅普诺夫函数来进行动态系统的设计，采用神经网络分别构造动态系统中的估计项与稳定项。

4.4.1 稳定变分动态系统结构

基于以上对动态系统构建方法与其稳定性分析后，与之前的工作不同的是，本文不是通过施加预定义的控制函数约束来强制稳定性，而是直接将其整合到动力学模型中通过数据进行学习。本文基于条件变分自编码器网络与输入凸神经网络构建了一个稳定变分动态系统（Stable Variational Dynamical System, SVDS）。其结构如图 4-2 中所示。

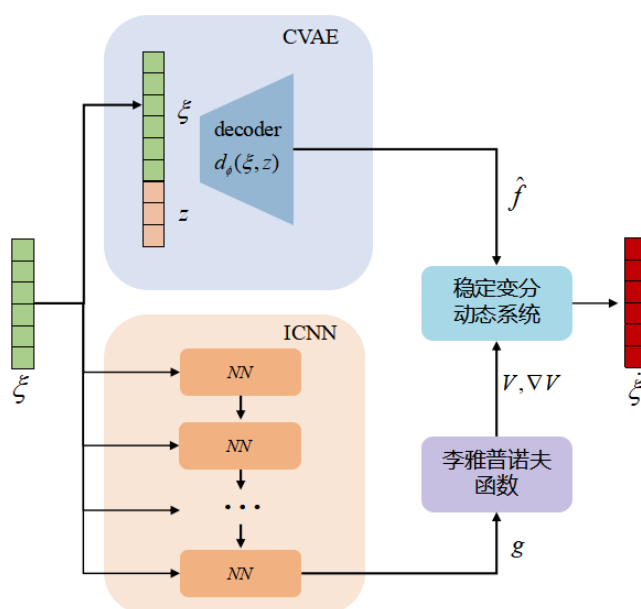


图 4-2 稳定变分动态系统 (SVDS) 结构

该动态系统模型由基于条件变分自编码器网络的动态系统估计项和基于输入凸神经网络的动态系统稳定项构成，其中 ξ 表示模型的输入， z 表示 3 维的正态分布向量， g 表示 ICNN 网络，李雅普诺夫函数 V 是基于 g 设计的，最后结合估计项 $\hat{f}(\xi)$ 由公式(4-10)得到动态系统的输出 $\dot{\xi}$ 。

4.4.2 稳定变分动态系统运动控制流程

机械臂运动控制流程如图 4-3 所示，展示了从示教轨迹到机器人控制器执行的完整流程。

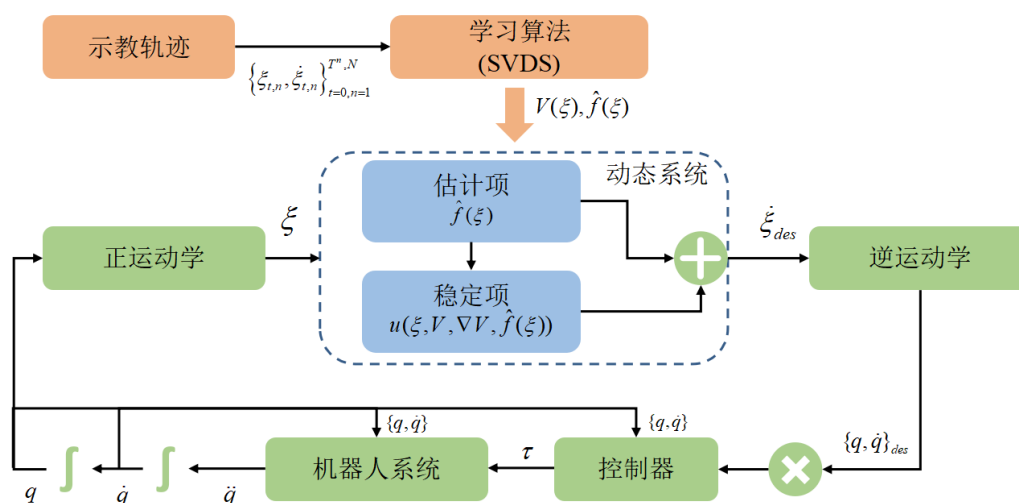


图 4-3 稳定变分动态系统运动控制流程

其中，控制流程可以分为离线学习和在线计算两部分，图中，橙色块代表离线学习模块，蓝色和绿色块代表在线计算。在离线学习阶段，首先进行示教数据的采集，采集到的轨迹数据将用于学习算法中进行模型的训练，该模型能够同时学习动态系统估计项 $\hat{f}(\xi)$ 和稳定项。估计项 $\hat{f}(\xi)$ 主要是对通过回归模型对示教运动进行建模，得到初步的速度估计值。修正项 $u(\xi)$ 则是对估计项的控制，使得生成的运动即使在扰动的情况下也能收敛到期望目标，保证了系统的稳定。

在模型训练完成后，对于每次在线更新，在线估计框架均会使用学习到的模型重新估计动力学 $\hat{f}(\xi)$ 与计算稳定项 $u(\xi)$ 。动态系统以机器人当前状态 ξ_{des} 作为输入，输出期望的速度控制指令 $\dot{\xi}_{des}$ 。逆运动学模块将期望状态从笛卡尔空间转换到关节空间并输出关节位置与关节速度到下级控制器。其中期望的关节角度及其导数记为 $q, \dot{q}_{des}$ 。 $q, \dot{q}_{des}$ 被作为输入提供给低级控制器计算所需电机扭矩 τ 。此控制信号考虑了实际动力学，确保输出信号能够在实际机器人硬件上实现期望的动作。与此同时，还需要考虑逆运动学来确保计算出的期望位置能够被实际机器人系统准确跟随。

4.5 基于条件变分自编码器的动态系统估计项

动态系统的估计项 $\hat{f}(\xi)$ 通常会采用传统的机器学习中的非线性概率回归方法来构建。传统的机器学习方法在面对复杂的运动轨迹时往往需要手动调整模型的参数来提高模型复杂度，难以实现精确的复现。基于深度学习的回归算法具有强大的数据驱动和特征学习能力，能够从大量数据中自动学习复杂的特征表示，而无需手动设计或选择特征，通过多层非线性变换，能够捕获数据中的复杂、抽

象的非线性关系。但是常见的深度学习网络通过堆叠多个全连接层来构建回归模型，这样容易导致模型缺乏鲁棒性和很容易过拟合。本文采用了条件变分自编码器 (Conditions Variational Auto-Encoder, CVAE)^[52]来构建动态系统估计项 $\hat{f}(\xi)$ ，其结合了神经网络的强大的拟合能力与贝叶斯概率模型的泛化能力使得模型性能更优。条件变分自编码器模型结构如图 4-4 所示。

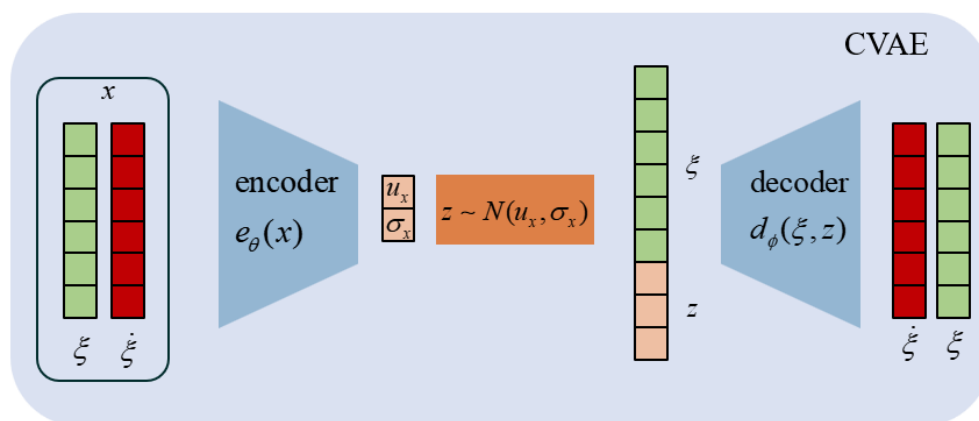


图 4-4 条件变分自编码器(CVAE)结构

条件变分自编码器的结构可以分为三个主要部分：编码器 (Encoder)、解码器 (Decoder)和条件信息。其中，编码器接受 $x = (\xi, \dot{\xi})$ 作为输入。编码器通过处理这些信息，输出潜在空间的均值 u_x 和方差 σ_x 。这些参数定义了一个潜在空间中的一个高斯分布 $N(u_x, \sigma_x)$ ，从中可以采样得到潜在表示。解码器的作用是将潜在空间的表示转换回原始数据空间。在条件变分自编码器中，解码器同时接受潜在表示 z 和条件信息 ξ 作为输入，模型的输出为 $(\xi, \dot{\xi})$ 。在模型训练过程中，条件变分自编码器通过最小化重构误差和潜在空间分布与先验分布之间的 KL 散度来训练，这使得潜在表示符合先验分布（标准正态分布），能够带来更好的泛化能力。在训练过程中以 $x = (\xi, \dot{\xi})$ 作为输入和输出，目的是构建平滑连续的状态与速度的联合空间分布。在 SVDS 运行过程中只需要使用训练后的解码器即可，通过高斯分布随机采样，将机器人状态作为条件信息，通过解码器可以得到当前状态的速度输出。

自动编码器将潜在空间作为一个固定的点作为输入数据的潜在表示，没有考虑输入数据的概率分布。而条件变分自编码器提供了一个概率框架，将输入数据映射到潜在空间的概率分布上。这意味着每个输入数据点被编码为一个分布（高斯分布），从而允许模型捕捉输入数据中的不确定性和变化。随后通过优化变分下界条件变分自编码器能够学习到一个更加平滑和连续的潜在空间，在示教轨迹附近也能生成平滑连续合理的新数据。

4.6 基于输入凸神经网络的动态系统稳定项

4.6.1 输入凸神经网络

在动态系统稳定性分析中，传统的方法通常人工构造李雅普诺夫函数，尤其是对于高维或非线性系统来说，这是一个复杂和充满直觉性的过程。在实际使用中通常将其设计为一个简单的二次型函数 $V(x) = x^T P x$ 。这种简单的设计可能会对运动模型产生过于严格的稳定性约束^[36]，在学习过程中限制模型的精度。而输入凸神经网络^[53]（Input Convex Neural Network, ICNN）提供了一种替代方法，可以用数据驱动的方式来构建李雅普诺夫函数。输入凸神经网络结构如图 4-5 所示。

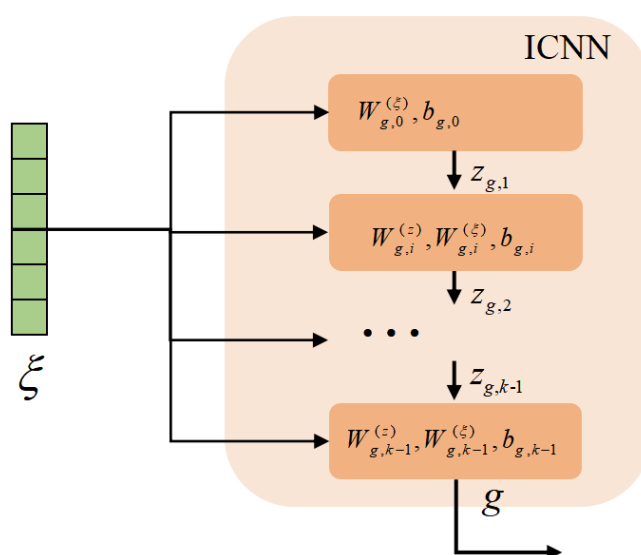


图 4-5 输入凸神经网络结构

输入凸神经网络被设计为保持输入到输出的凸性。这意味着网络的输出关于输入是凸函数，从而可以利用凸优化技术找到全局最优的网络参数。对于输入空间中的任意两点，网络输出的线性组合的值大于或等于这两点输出值的相同线性组合，如果将凸神经网络用函数 $f(x)$ 代表，对于所有 x_1, x_2 在其定义域和 θ 满足 $0 \leq \theta \leq 1$ 都有：

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta)f(x_2) \quad (4-11)$$

为了保持输入到输出的凸性，输入凸神经网络中的权重通常被限制为非负值，网络的偏置项通常也被限制为非负，确保它们不会破坏凸性。对于神经网络而言，非线性是其突出的重点作用，在输入凸神经网络中为了保持凸性的情况下提供非

线性能力的激活函数，选择了凸函数如 ReLU 作为激活函数，在多层结构之间保持前向传递。一个基本的输入凸神经网络层可以表示为：

$$h_{i+1} = \sigma(W_i h_i + b_i) \quad (4-12)$$

其中 h_i 是第 i 层的输出， W_i 是非负权重矩阵， b_i 是非负偏置项， σ 是递增的激活函数。由于每一层的输出都是关于前一层输出的凸函数，整个网络依然保持了凸性。因此，整个网络的输出 $f(x)$ 是关于输入 x 的凸函数。

4.6.2 李雅普诺夫函数设计

在本章中通过搭建了多层的输入凸神经网络（ICNN）来构建一个具有凸性质的李雅普诺夫函数 V 。多层输入凸神经网络 $g(\xi)$ ，其定义如公式(4-13)所示，它强制性地要求 $g(\xi)$ 是关于其输入 ξ 的凸函数。

$$\begin{aligned} z_{g,1} &= \sigma_{g,0} \left(W_{g,0}^{(\xi)} \xi + b_{g,0} \right) \\ z_{g,i+1} &= \sigma_{g,i} \left(W_{g,i}^{(z)} z_{g,i} + W_{g,i}^{(\xi)} \xi + b_{g,i} \right), i=1, \dots, k-1 \\ g(\xi) &= z_{g,k} \end{aligned} \quad (4-13)$$

其中 $z_{g,i}$ 表示第 i 层激活后的输出， $W_{g,i}^{(z)}$ 是将前一层激活 $z_{g,i}$ 映射到后续层的非负权重， $\sigma_{g,i}$ 是凸的、非递减的激活函数。 $W_{g,i}^{(\xi)}, b_{g,i}$ 是 ICNN 中的网络全连接层参数， $W_{g,i}^{(\xi)}$ 是非负权重矩阵， $b_{g,i}$ 是非负偏置项。

ICNN 中的一个突出添加是“透传”层，它将输入 ξ 直接连接到更深层次的隐藏单元，类似于残差神经网络的跨层连接。不同的是，ICNN 中这种连接只有从输入到网络的各层之间，而没有层之间的连接。在传统的前馈网络中，这样的层是不必要的，因为前一个隐藏单元总是可以映射到下一个单元，具有恒等映射。然而，对于 ICNN 来说，权重 $W_{g,i}^{(z)}$ 非负的限制减少了隐藏神经元的作用，使其特征提取能力降低，因此通过额外的“透传”可以使得深层的网络能够学习到更多的特征。即是说， $W_{g,i}^{(z)}$ 权重的非负性约束虽然有些限制性，但由于 $W_{g,i}^{(\xi)}, b_{g,i}$ 的灵活性，网络仍然具有普适的逼近能力。

进一步地，因李亚普诺夫函数 V 应当是正定的，即 $V(\xi) > 0$ 对于 $\xi \neq 0$ ，因此本文将李亚普诺夫函数 V 定义如下式：

$$V(\xi) = \sigma_{g,k} \left(g(\xi) - g(0) \right) + \eta \|\xi\|^2 \quad (4-14)$$

其中 g 是在(4-13)中定义的 ICNN。 η 是一个小的正常数， $\sigma_{g,k}$ 是一个非负的凸非递减函数，且 $\sigma_{g,k}(0) = 0$ 。最后，控制李亚普诺夫函数 V 应该是连续可微的。为了实现这一点，选择采用了一个带有二次区域的平滑 ReLU 激活函数来替代在

公式(4-13)和公式(4-14)中的传统 ReLU 激活函数，如图 4-6 所示，其定义如公式(4-15)所示。

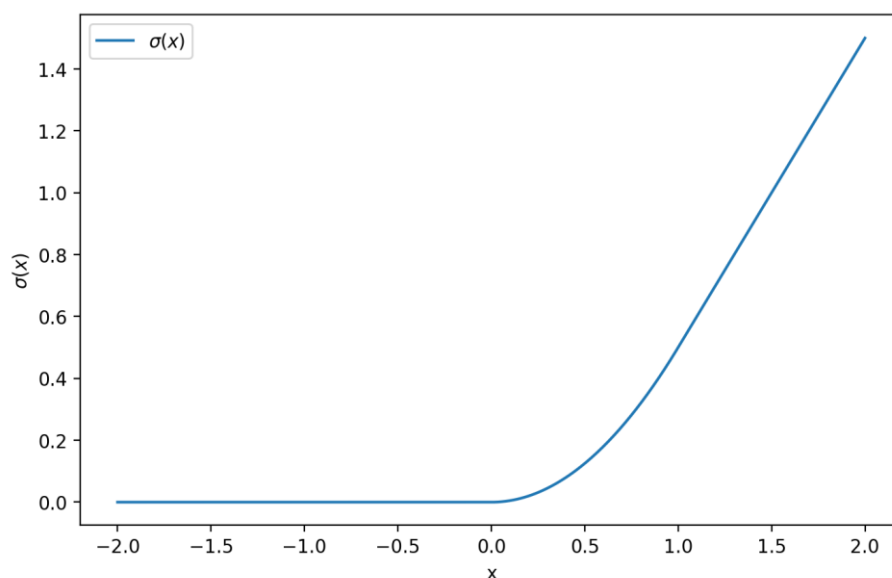


图 4-6 平滑 ReLU 激活函数

$$\sigma(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ \frac{x^2}{2\delta}, & \text{if } 0 < x < \delta \\ x - \frac{\delta}{2}, & \text{otherwise} \end{cases} \quad (4-15)$$

在本章节中，先通过使用 CVAE 来构建动态系统的估计项 $\hat{f}(\xi)$ ，再通过 ICNN 构造 $g(\xi)$ 学习李雅普诺夫函数得到 $V(\xi)$ ，由公式(4-9)得到了动态系统稳定项 $u(\xi)$ 的表达式，最后通过公式(4-10)得到动态系统的输出 $\dot{\xi} = f(\xi) = \hat{f}(\xi) + u(\xi)$ ，至此完成了整个稳定变分动态系统的构建。

4.6.3 稳定性证明

本节将就所设计的稳定变分动态系统 (SVDS) 的稳定性进行证明。假设动态系统产生的运动为 $\xi(t)$ ，李雅普诺夫函数 V 随时间的导数如公式(4-16)所示。

$$-\frac{d}{dt}V(\xi(t)) = \nabla V(\xi(t))^T \frac{d}{dt}\xi(t) \leq -\alpha V(\xi(t)) \quad (4-16)$$

其中 $\nabla V(\xi(t))$ 表示 V 对 $\xi(t)$ 的梯度； $d\xi/dt$ 表示状态 $\xi(t)$ 随时间的变化率； α 是一个正常数，表示系统收敛到平衡点的速度；这个导数是负的，意味着系统

随着时间的推移会向着平衡状态演进。通过对通过积分这个不等式，得到 V 的上界如公式(4-17)所示：

$$V(\xi(t)) \leq V(\xi(0))e^{-\alpha t} \quad (4-17)$$

这表明如果系统开始时是在某个状态，那么随着时间的推移，李雅普诺夫函数将会以 $e^{-\alpha t}$ 的速率指数级减小，即系统能量或其他相关度量会逐渐降低。由于 V 的正定性要求，有以下定义：

$$V(\xi(t)) \geq \eta \|\xi(t)\|^2 \quad (4-18)$$

其中 η 是一个正常数； $\|\xi(t)\|^2$ 为 $\xi(t)$ 的二范数； V 的值随着 $\xi(t)$ 趋向于零而趋向于零。结合公式(4-17)和(4-18)得到了系统状态向量 $\xi(t)$ 的范数的上界，如下式所示。

$$\|\xi(t)\| \leq \sqrt{\frac{V(\xi(t))}{\eta}} \leq \sqrt{\frac{V(\xi(0))}{\eta}} e^{-\frac{\alpha t}{2}} \quad (4-19)$$

然后，考虑到激活函数 σ 是线性的且在原点是二次的， $V(\xi(t))$ 在 $\|\xi(t)\| \rightarrow \infty$ 时是线性的，并在原点附近是二次的，这意味着它可以被 $\xi(t)$ 的二次项上界所约束（如公式）：

$$V(\xi(t)) \leq \rho \|\xi(t)\|^2 \quad (4-20)$$

由式(4-19)与式(4-20)可以推出式(4-21)，显示状态向量的范数指数级收敛到平衡点：

$$\|\xi(t)\| \leq \sqrt{\frac{1}{\eta}} \|\xi(0)\| e^{-\frac{\alpha t}{2}} \quad (4-21)$$

最终，以上的公式分析证明了动态系统指数级地收敛到平衡点。这个定理和其证明过程对于理解和证明复杂动力学系统的稳定性至关重要。

4.7 实验及分析

4.7.1 实验设计

为了验证算法的有效性，本研究首先利用 LASA 手写数据集^[36]对 SVDS 方法进行了二维的动态系统学习，并在仿真中生成了新的手写字母轨迹。随后拓展到更复杂的六维动态系统学习。通过使用第二章中所采集的机器人末端执行器的空

间位置和姿态数据进行模型训练。接着，在仿真平台上对训练好的动态系统模型进行了进一步的验证，以确认其有效性和稳定性。最后进入实物实验阶段，在真实世界条件下进行机器人操作实验。这一阶段包含了两项测试：静态抓取和动态跟踪，以评估机器人在抓取静止物体以及追踪动态目标方面的能力。

4.7.2 仿真实验

基于上一节提出的 SVDS 框架，本部分将基于 LASA 手写数据集在二维空间中进行轨迹生成仿真，再通过 UR5e 机械臂示教数据集学习六维动态系统并进行仿真实验。

4.7.2.1 二维手写字母

LASA 手写数据集（LASA Handwriting Dataset）是由瑞士洛桑联邦理工学院（EPFL）的学习算法与系统实验室（Learning Algorithms and Systems Laboratory, LASA）创建的一个用于模式识别和机器学习研究的手写轨迹数据集。LASA 手写数据集包含了多种手写图样的轨迹数据，用于支持对手写动作识别、轨迹生成以及机器学习算法，被广泛应用于机器人学习方面的研究中。LASA 手写数据集一共包含 30 个不同图样的数据，每个图样包含了 7 条示教轨迹数据，每条轨迹在经过预处理后均是从不同的初始点开始到相同的最终点结束。每条轨迹长度为 1000 个数据点，记录着包括轨迹位置、速度、加速度等信息。

本文基于 SVDS 对每一种图样数据进行了单独训练，其中网络超参数设置如表 4-1 所示：

表 4-1 LASA SVDS 超参数

超参数	数值
输入维度	2
输出维度	2
隐藏层维度	128
潜在空间维度	3
训练轮数	100
学习率	0.005
α	0.1
η	0.1
δ	0.001

在经过 100 个 epoch 的训练后，所有的模型都已收敛，损失曲线如图 4-7 所示，图中展示了选取的 10 中图样的训练损失曲线。

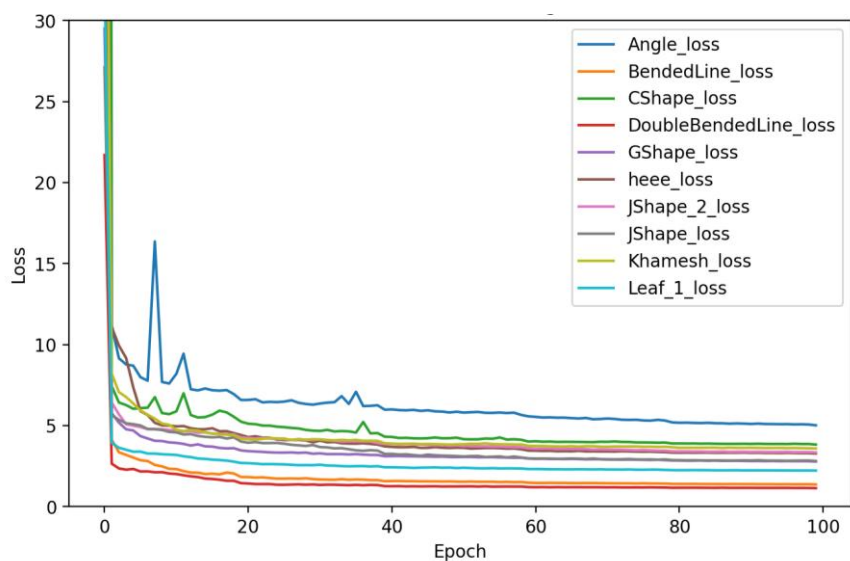


图 4-7 LASA 手写数据集模型损失曲线

模型训练完成后将进行轨迹复现测试。在测试环节中将 30 个图样的 7 条轨迹起始位置作为输入数据送到二维 SVDS 模型中，通过对模型输出的速度进行积分计算出下一时刻的位置，并通过连续迭代生成了完整的规划轨迹。完整的规划轨迹如图 4-8 所示，完整实验结果见附录 A。

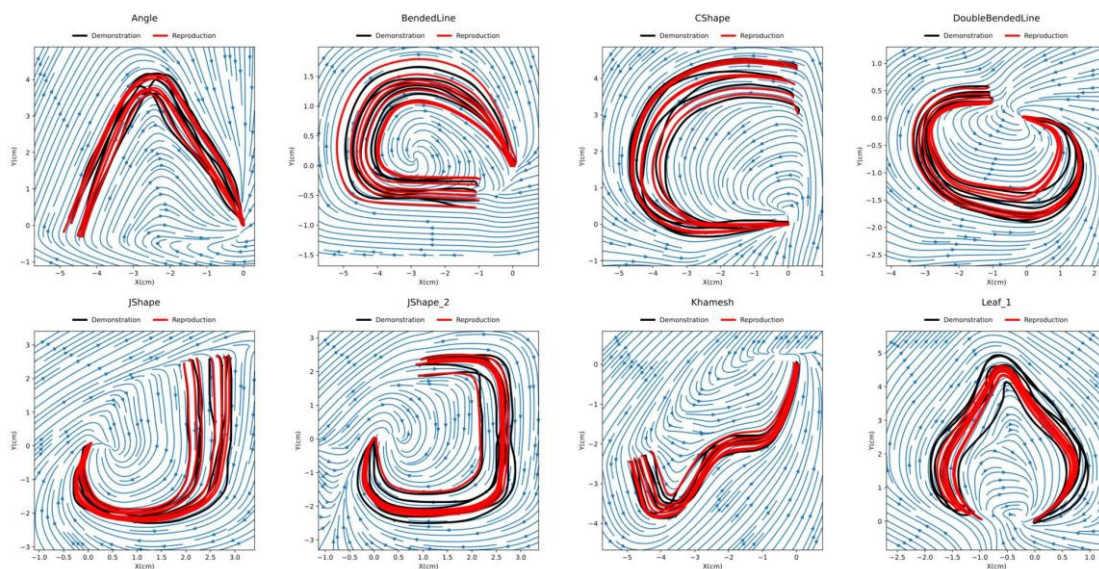


图 4-8 LASA 复现轨迹

其中黑色轨迹表示示教数据，红色曲线是根据示教数据的起始位置点通过网络模型得到速度数据不断积分得到的复现轨迹，蓝色曲线是不同位置速度的方向，如图所示在模型在所有图样上生成的复现轨迹均能顺利收敛到目标点，这表明了学习到的动力系统具有稳定性。

LASA 手写数据集的李雅普诺夫函数能量场如图 4-9 所示，完整实验结果见附录 A。

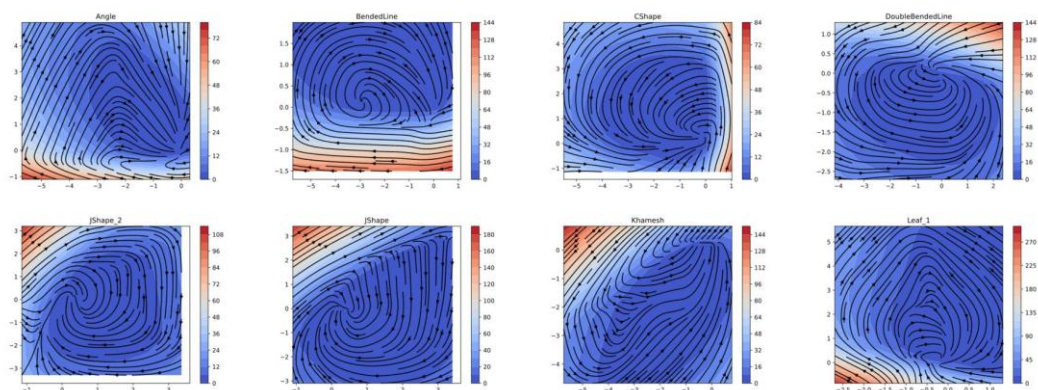


图 4-9 LASA 李雅普诺夫函数能量场

其中图 4-9 展示了动态系统在二维平面的能量场，其中黑色曲线为速度向量的方向，背景中等高线区域为描述运动的李雅普诺夫能量函数，而目标位置始终有着最低的能量，速度场最终收敛于能量最低的目标位置。

LASA 手写数据集的复现轨迹速度变化曲线如图 4-10 所示，完整实验结果见附录 A。

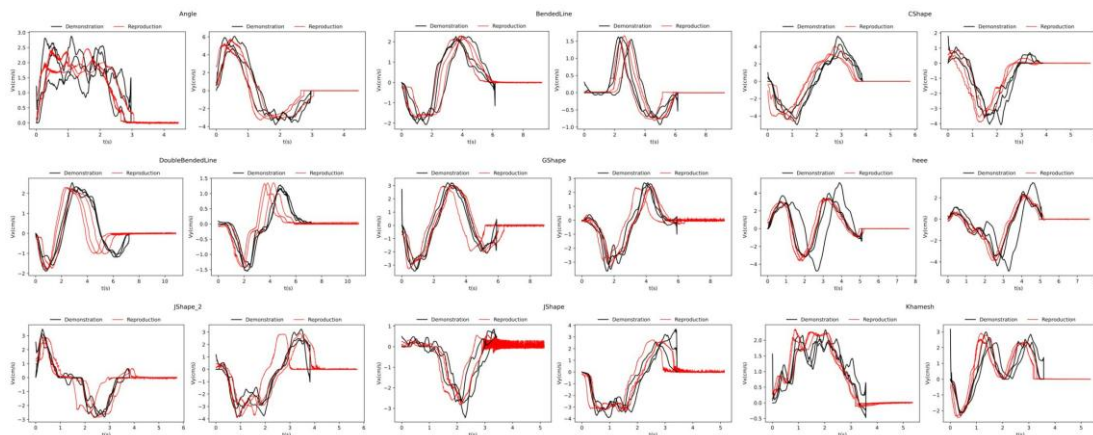


图 4-10 LASA 速度变化曲线

图中展示了 LASA 复现轨迹的 x, y 轴速度变化曲线，其中黑色曲线为示教轨迹的速度变化曲线，红色曲线为复现轨迹速度变化曲线，在大部分图形上复现速度紧密跟随着示教速度，这表明再现的轨迹与演示过程具有相似的速度，意味着它们遵循相似的动力学。最终所有速度均趋于零点，说明复现系统具有良好的稳定性。

为了增强结果的说服力，本文采用 CLF-DM 方法^[37]构建动态系统对 LASA 手写数据集进行测试，测试结果如图 4-11 所示，完整实验结果见附录 A。

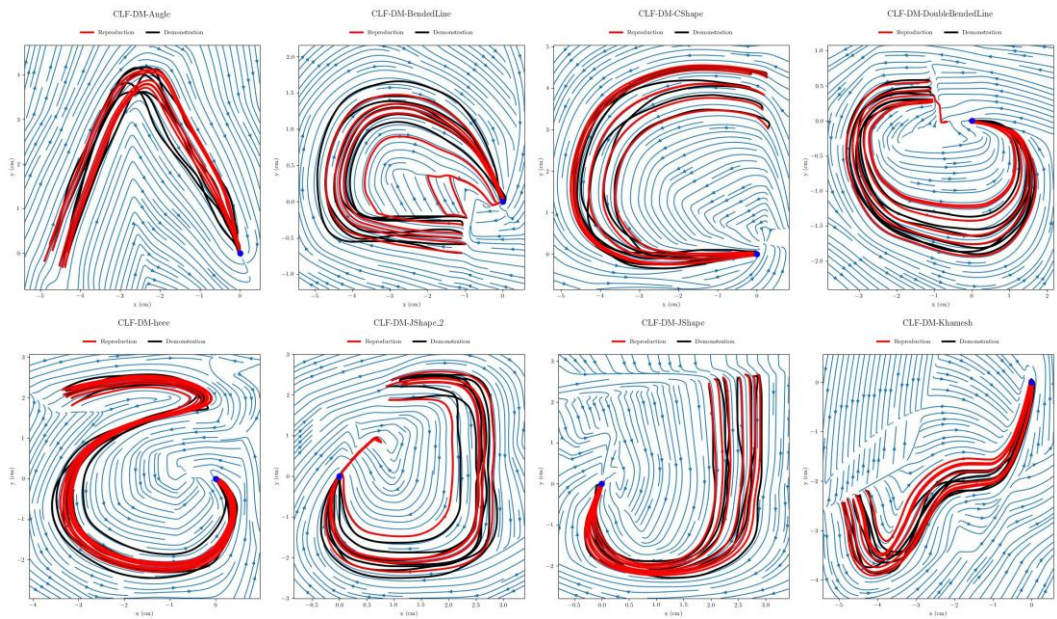


图 4-11 CLF-DM 复现轨迹

图中每幅子图展示了特定图样的动态系统轨迹复现效果，其中红色轨迹代表系统的复现轨迹，而蓝色向量场则表现了整个动态系统的速度向量场。可以观察到在多个案例中，红色轨迹与黑色示教轨迹存在明显偏差，这表明了 CLF-DM 方法的复现精度上的不足，其次，蓝色向量场中的不平滑区域和断层表明动态系统稳定性和鲁棒性存在着问题，在机器人执行过程中，这些不平滑的向量场可能导致运动抖动或不稳定行为。更为关键的是在一些子图中，红色轨迹未能收敛到目标点，显示了 CLF-DM 方法在确保最终状态达到目标点方面的局限性。

相比之下，本文提出的 SVDS 方法在这些测试中表现出更为平滑和连续的速度向量场，以及更好的轨迹收敛性。这不仅验证了 SVDS 在生成平滑运动轨迹方面的有效性，也显示出其在确保动态系统稳定性和鲁棒性方面的优势。

虽然以上的分析集中在了复现精度和动态系统稳定性上，但在实际应用中，执行时间也是一个不可忽视的关键性能指标。表 4-2 对比了两种方法的执行时间。

表 4-2 模型执行时间

模型	执行时间(ms)	频率
CLF-DM	0.9203	1086 Hz
SVDS	4.1312	242 Hz

由于机器学习其模型结构较为简单，CLF-DM 方法在模型训练和执行时间上都明显优于深度学习模型，后者通常包含更多复杂的参数。尽管 SVDS 的执行时间较长，其运行频率为 242Hz，但这个频率已经能够满足机器人抓取任务。综合图 4-11 所复现的轨迹来分析对比结果，虽然深度学习模型在执行速度上可能不如较简单的机器学习模型，但它们在处理复杂任务时的能力更强。

4.7.2.2 六维机械臂运动

在二维 LASA 手写字母数据集上的测试表明，SVDS 在轨迹复现任务上具有较好的性能，既能生成平滑的轨迹又能保证系统的稳定性。本节将进一步在机械臂运动位姿轨迹数据上进行六维的仿真实验。机械臂位姿数据集的超参数与 LASA 手写数据集的超参数设置类似，如表 4-3 所示，仅对模型的输入维度与输出维度进行了修改，同时增加了训练轮数。

表 4-3 UR5e SVDS 超参数

超参数	数值
输入维度	6
输出维度	6
训练轮数	200

训练过程的损失下降曲线如图 4-12 所示。

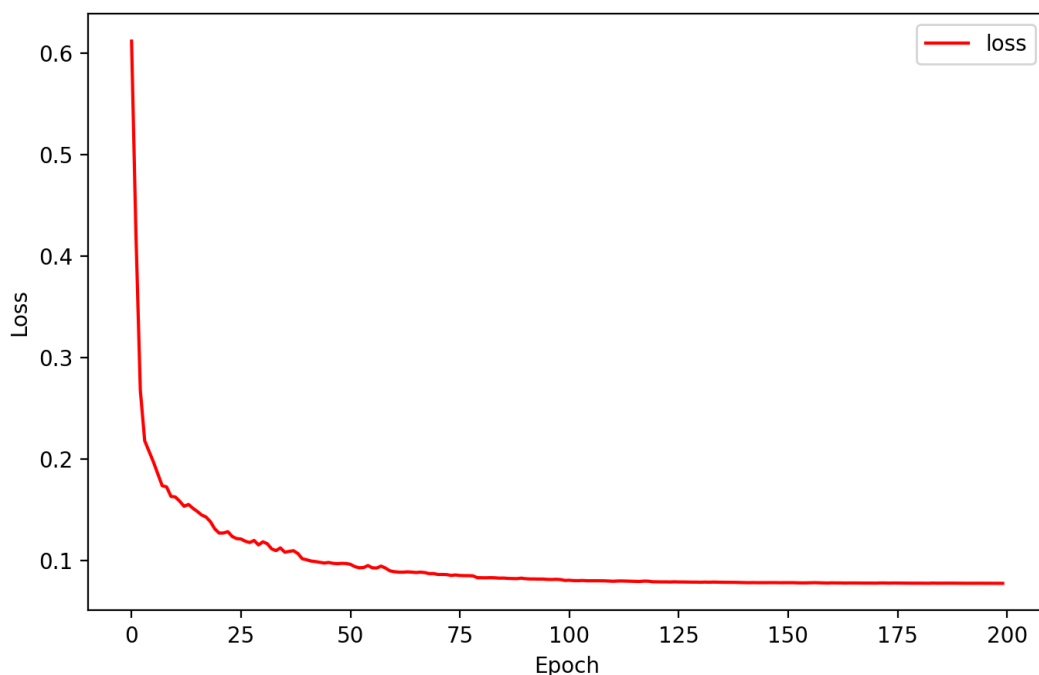


图 4-12 六维机械臂运动模型损失曲线

模型在 200 个 epoch 后成功收敛, 为了更直观地展示机械臂运动过程中位置和姿态的变化轨迹, 分布绘制了机械臂末端运动的三维空间位姿轨迹与二维平面位姿轨迹。如图 4-13 与图 4-14 所示。

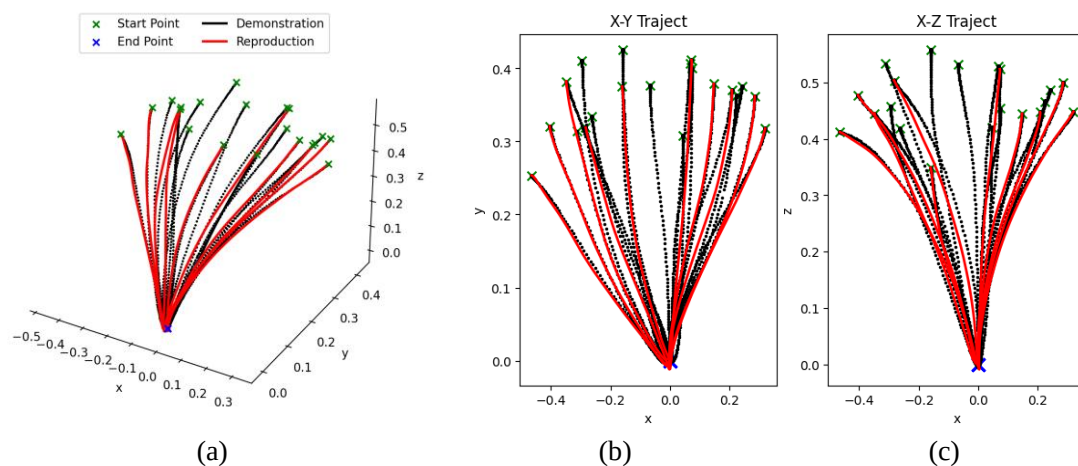


图 4-13 机械臂复现位置轨迹。(a)三维位置轨迹; (b)X-Y 二维位置轨迹;
(c) X-Z 二维位置轨迹;

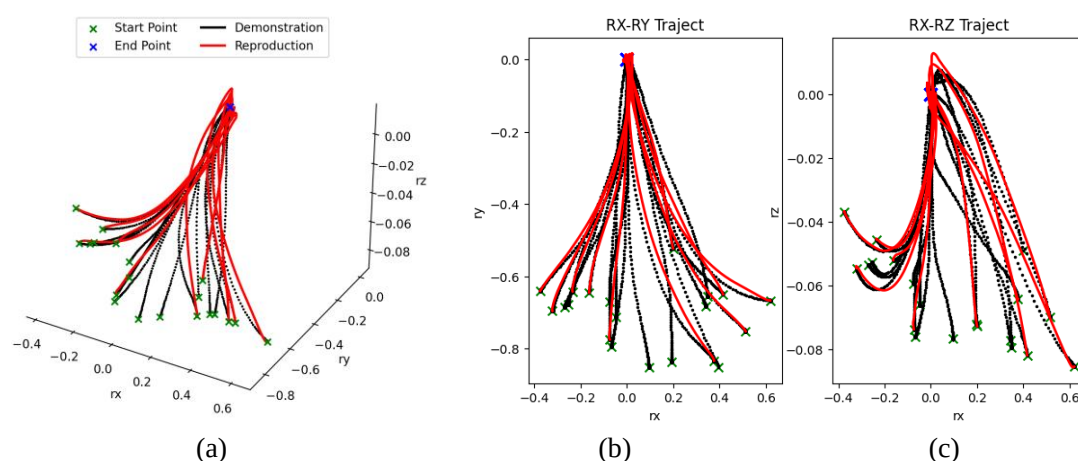


图 4-14 机械臂复现姿态轨迹。(a)三维姿态轨迹; (b)RX-RY 二维姿态轨迹;
(c) RX-RZ 二维位置轨迹;

图 4-13 中黑色曲线表示示教位置轨迹, 红色曲线表示通过起始点位置不断积分得到的复现位置轨迹。图 4-14 显示了机械臂姿态曲线, 其中姿态为轴角表示的三维数据, 其中黑色曲线表示示教姿态轨迹, 红色曲线表示从起始点通过积分得到的复现姿态轨迹。从图中可以看出, SVDS 准确地复现了示教的位姿轨迹与, 且所有运动轨迹均能收敛到目标点位姿。

在完成算法验证之后, 为了保证模型能够成功运行在实物机器人上, 本章通过 ROS 环境对 UR5e 机械臂进行仿真实验, 实验过程如图 4-15 所示。

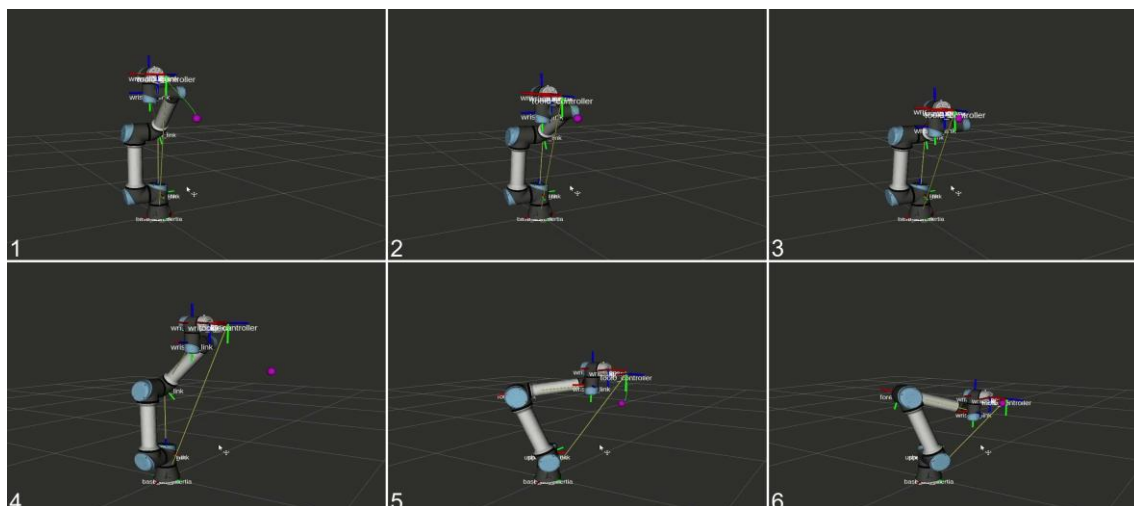


图 4-15 机械臂仿真实验

在仿真实验中，分别随机设置了两个随机目标（紫色小球），子图 1-3 展示了机械臂如何运动到第一个目标，子图 4-6 展示了机械臂如何运动到第二个目标。SVDS 以目标与机械臂末端的相对位姿作为输入，输出当前状态的笛卡尔空间速度指令，通过循环进行速度积分得到完整的规划轨迹（绿色曲线）。通过实验验证，多次改变目标位置，SVDS 均能控制机械臂末端够成功到达目标位置。

4.7.3 实物实验

继仿真实验之后，为了验证在真实机器人上的应用效果，本章设计了两个机器人实物实验：静态抓取和动态抓取实验。

4.7.3.1 静态抓取实验

在实物实验环节，首先进行的是静态抓取实验，如图 4-16 和图 4-17 所示。为了方便灵巧手进行抓取操作，本实验将四个反光标记球贴在瓶盖上方与泡沫基座上以避免在抓取过程中发生遮挡与碰撞。通过 Vicon 动捕系统将矿泉水瓶与泡沫基座构造成一个刚体，将刚体中心移动到矿泉水瓶的中心以获取精准的目标物体位姿。实验中，先随机将矿泉水瓶放置在机械臂可达空间内，机械臂实时获取矿泉水瓶在其基坐标系下的位姿，将其设为目标位姿，将当前位姿与目标位姿进行相对转换后输入 SVDS 规划程序中，得到控制速度，控制机械臂末端到达目标位姿，最后进行抓取动作。

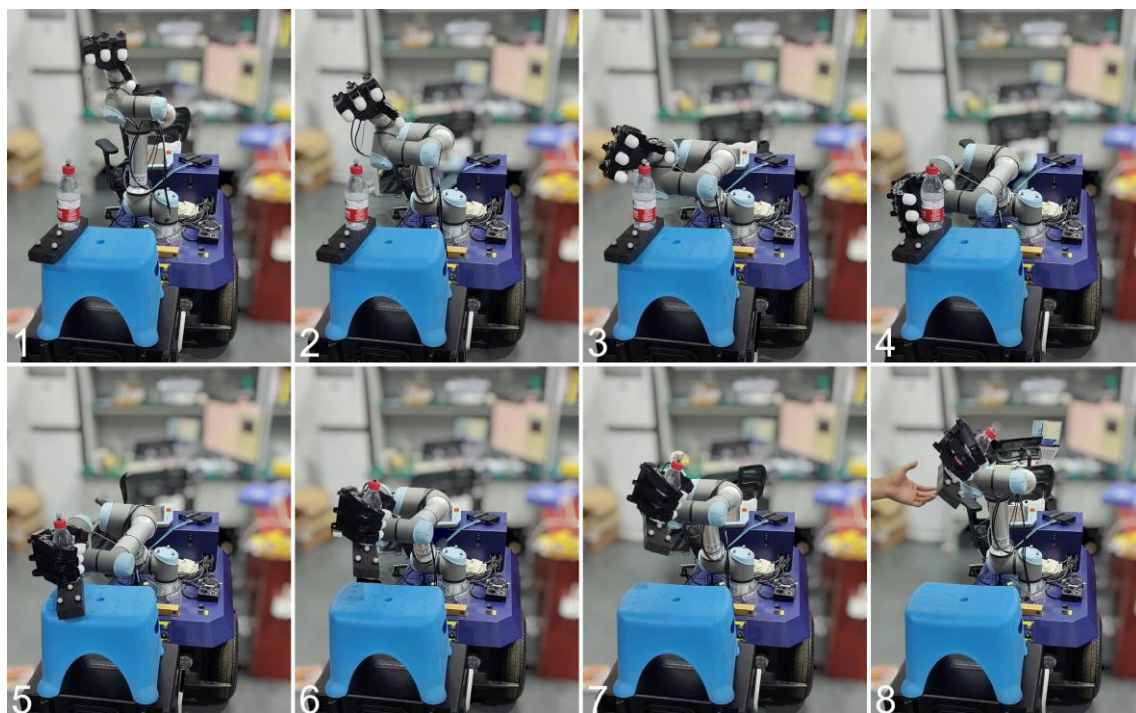


图 4-16 静态抓取实验 1

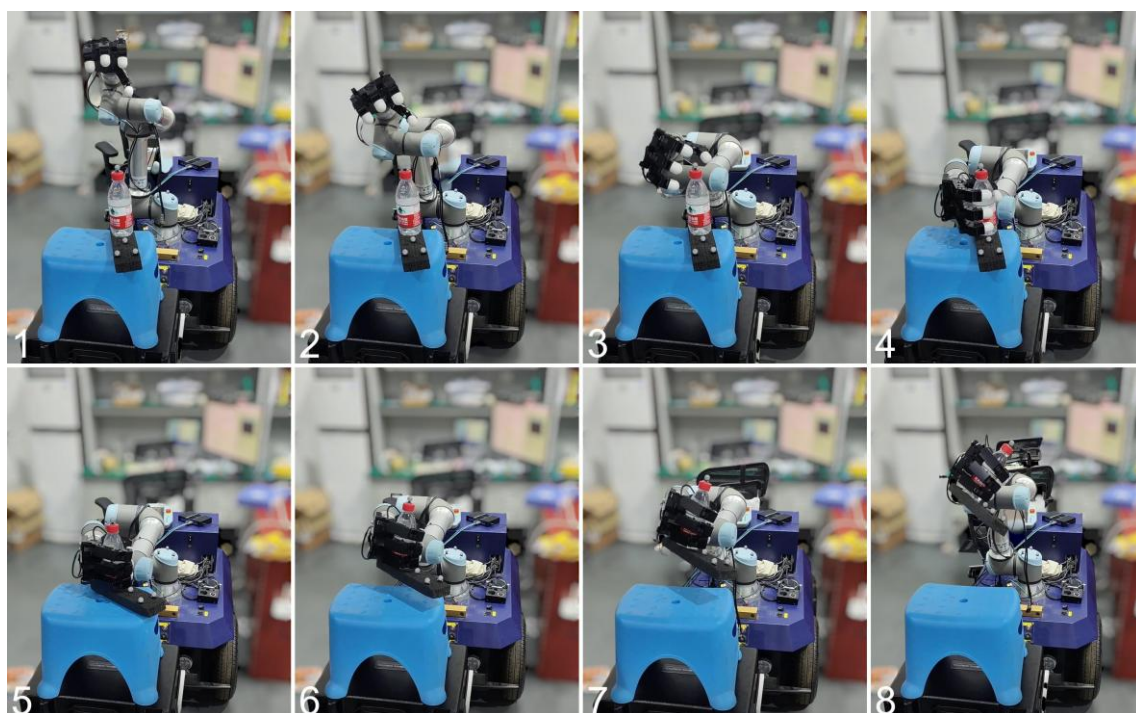


图 4-17 静态抓取实验 2

4.7.3.2 动态跟踪实验

为了测试算法在动态环境下的效果，本章接着进行动态跟踪实验，如图 4-18 所示。在动态跟踪实验中将矿泉水瓶的质心向机械臂基座标移动了 10cm 以实现

跟踪与避免碰撞。在实验中先将矿泉水瓶放置在蓝色凳子上，当机械臂运动到物体质心后再移动物体，在移动过程中不断调整物体的姿态以验证控制算法的有效性。

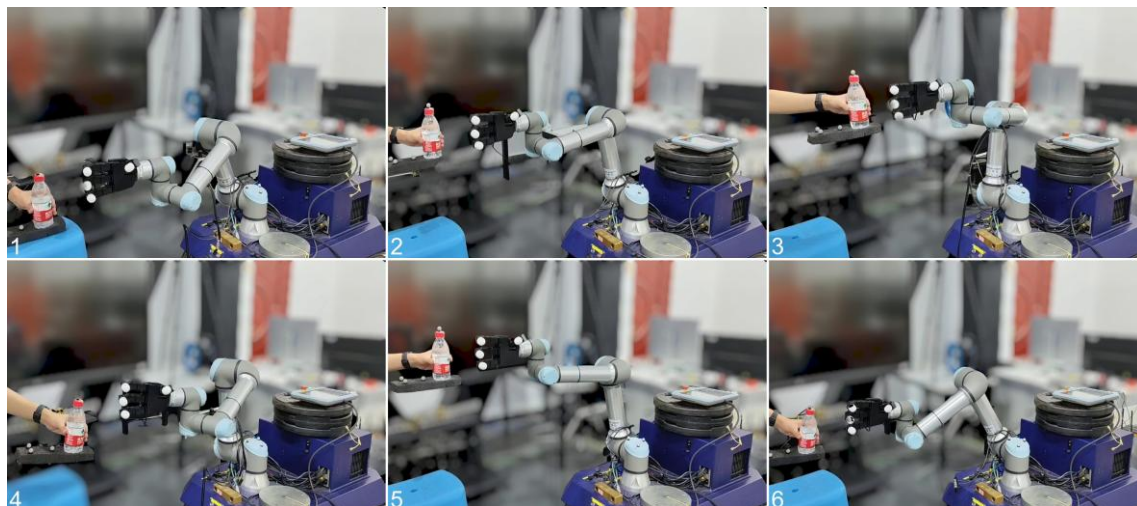


图 4-18 动态跟踪实验

鉴于通过矿泉水瓶的跟踪实验无法直观地体现跟踪的效果，本章进一步设计了一个动态跟踪对接实验来验证跟踪效果。在实验前，先在两台机械臂末端安装上了微型卫星模型并在 Franka 机械臂卫星模型上贴上反光标记点以获取目标的实时位姿。随后预先对 Franka 机械臂进行示教采集路点，在实验中 Franka 机械臂可以根据预设的路点以不同的速度进行移动。在实验中，控制程序通过实时获取 UR5e 机械臂与目标物体的相对位姿进行动态跟踪。实验过程如图 4-19 所示。

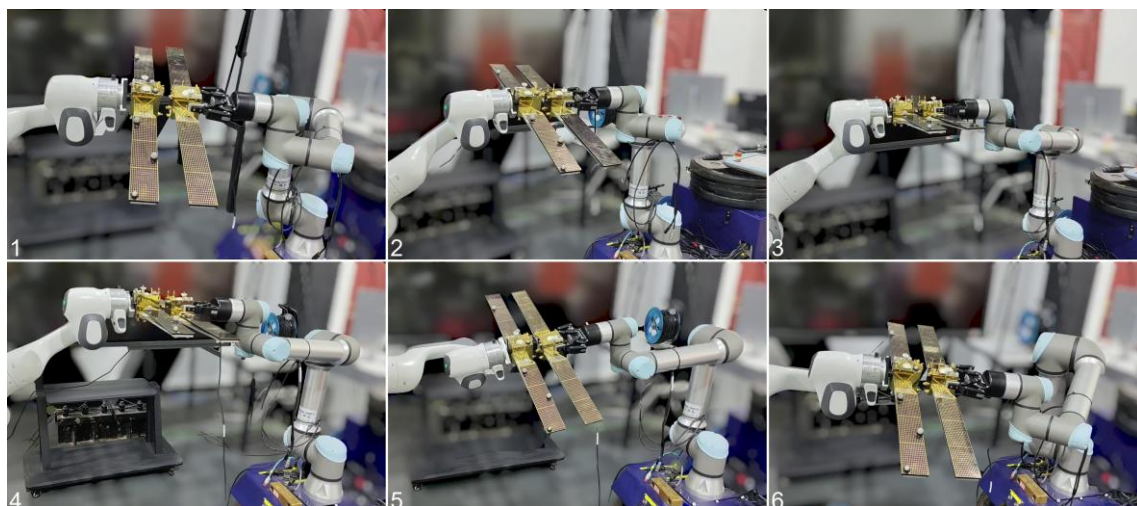


图 4-19 动态跟踪对接实验

4.7.4 结果分析

实验结果表明,所采用的模型在理论上和实际应用中都表现出了高度的有效性和稳定性。在二维动态系统的仿真实验中,本文将 CLF-DM 与提出的 SVDS 复现的轨迹进行对比,充分说明 SVDS 网络模型在捕捉系统动态和学习控制策略方面表现出良好的性能。在六维动态系统的仿真实验中,本文先通过轨迹复现验证了算法的有效性,进而通过在 ROS 中的仿真实验来验证系统的可靠性与稳定性。最后在真实物理机械臂实验中进行了静态抓取与动态跟踪两部分实验,在静态抓取实验中,对于不同的目标位置,动态系统均能产生稳定的控制命令,引导机械臂运动到达目标位置。动态跟踪实验中,模型同样展现出了高效和快速的调整能力,成功跟踪目标运动,实验表明了模型不仅在静态环境下有效,而且能够应对动态变化的目标。

4.8 本章小结

针对基于模仿学习中传统运动控制算法所面临的轨迹复现精度低,缺乏稳定性保证问题,本文提出了一种基于条件变分自编码器与输入凸神经网络构建的稳定变分动态系统(SVDS)。该系统由两部分组成:基于条件变分自编码器构建动态系统估计项和基于凸神经网络的动态系统稳定项。其中条件变分自编码器的潜在空间是连续分布的,保证了估计项的输出连续平滑,而通过利用输入凸神经网络凸性质设计李雅普诺夫函数构建稳定性更是保证了系统的稳定性,确保了机械臂在实际运行中的安全。在本章的实验中从仿真模拟和实物验证两个层面对本章提出的 SVDS 运动控制算法进行了验证,充分证明了本文基于模仿学习的敏捷运动控制算法的有效性。

第五章 动态目标抓取原型系统设计与实现

5.1 引言

机器人抓取是一项富有挑战性的任务，随着机器学习与深度学习的发展，基于强化学习与混合学习的方法逐渐成为了研究的热点，研究者可以借助计算机仿真计算在机器人仿真平台进行算法验证，但将强化学习算法迁移到现实世界的机器人系统中时，会面临一系列的挑战和问题。基于虚拟物理引擎的机器人环境难以精确模拟现实中的物理属性（如摩擦力、重力），模拟环境往往简化或忽略了这些因素，导致在真实环境中应用时效果不佳。现实世界的机器人系统受到硬件性能的限制，包括处理能力、感知系统和执行机构的性能。同时，机器人系统通常需要实时或近实时做出响应，这对学习算法的计算效率提出了高要求。因此本章在现实环境中搭建了动态目标抓取原型系统来验证本文提出的算法效率并且最终实现对飞行物体的抓取。

5.2 系统流程

动态目标抓取原型系统运行流程如图 5-1 所示。

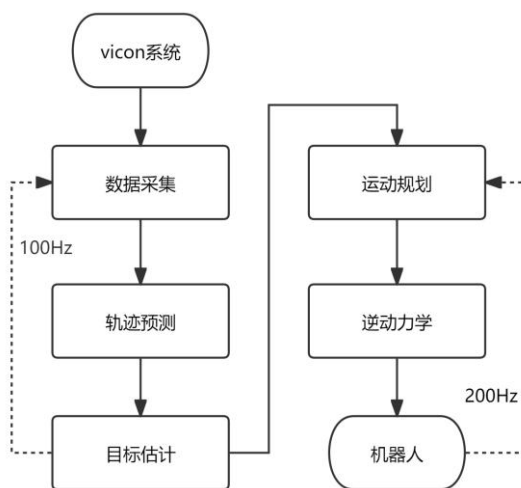


图 5-1 动态目标抓取原型系统流程图

其中，Vicon 动捕系统以 100Hz 的频率向外提供飞行物体的轨迹数据。系统使用这些数据来预测物体的运动轨迹，计算出机器人应在何处捕捉物体。一旦确定了捕捉位置，机器人根据当前机械臂末端位姿与目标位姿状态调用运动控制算

法生成机械臂的速度控制指令。机械臂在接受到控制指令后，通过逆运动学解算得到底层控制力矩，进而控制机械臂运动到目标位姿。其中预测与规划在这个过程中不断进行迭代，预测程序保持着与动捕系统相同的频率，以 100Hz 的频率进行目标位姿的更新，而机械臂的运动控制频率为 200Hz，以确保机器人能够快速、准确地响应。

5.3 硬件系统

动态目标抓取原型系统高性能硬件的使用对于实现高效和精准的操作至关重要，硬件系统如图 5-2 所示。



图 5-2 硬件系统组成与连接

本文搭建的硬件系统包括 Vicon 动捕系统、UR5e 机械臂、AllegroHand 灵巧手、控制主机等。其中除 AllegroHand 与控制主机通过 PCAN 进行通信外，动捕系统、UR5e 均通过交换机与控制主机进行通信。这些高效的硬件协同工作，使得动态目标抓取原型系统能够快速响应环境变化，实时调整抓取策略，从而大幅提高了抓取任务的成功率。

5.3.1 机械臂

本研究以 UR5e 为例开展相关的实验，深入探讨机械臂在动态环境下的应用潜力，如图 5-3 所示。



图 5-3 UR5e 协作机器人

UR5e 是由丹麦公司 Universal Robots 生产的一款中型工业级协作机器人，被设计用于执行精确的自动化任务，如组装、打磨、涂漆、搬运和测试，适用于各种生产环境。得益于其卓越的灵活性、便于编程与集成的特性，以及在人机协作方面的高安全性，UR5e 机器人不仅广泛应用于工业领域，更是备受研究者的欢迎。其硬件参数见表 5-1。

表 5-1 UR5e 硬件参数

参数	描述
有效载荷	5 公斤
工作半径	850 毫米
自由度	6 个自由度
重量	20.6 公斤
重复定位精度	±0.03 毫米

5.3.2 灵巧手

实验中用到的末端执行器为 Allegro Hand 灵巧手，如图 5-4 所示。



图 5-4 Allegro Hand 灵巧手

Allegro Hand 是由韩国的 SimLab 公司开发多指灵巧手，具有四个手指和十六个独立的扭矩控制关节，能够灵活精确地模拟人类手部动作。Allegro Hand 以其

轻质、便携式拟人化设计而闻名，可实现低成本而灵巧的操作，适合研究和工业应用。其硬件参数如表 5-2 所示。

表 5-2 Allegro Hand 硬件参数

参数	描述
自由度	16
力和位置控制	力控
通信协议	PCAN
最大扭矩	0.17 Nm
最大负载	5 Kg
重量	1.08 Kg

5.3.3 控制主机

得益于本文提出算法的高效率，该算法在执行时间上远远低于预设标准，这允许在搭建系统时选择成本效益高的微型 NUC 作为控制主机。如图 5-5 所示，主机参数如表 5-3 所示。



图 5-5 控制主机

表 5-3 主机硬件参数

参数	描述
型号	NUC8i5
CPU	i5-8259U
内存	16GB
网络	1000M
操作系统	Ubuntu22.04

在硬件连接方面，控制主机通过 TCP 网络协议与 UR5e 机械臂进行数据交换，确保了数据通信的稳定性和可靠性。而对于接收 Vicon 动捕系统的数据，控制主机利用 UDP 协议来保证数据传输的速度，使得信息的传递延时保持在一个可控的水平以满足系统对实时性的要求。此外，Allegro Hand 灵巧手则通过 PCAN 协议与控制主机通信，这不仅提升了传输的效率，还确保了传递指令的精确性。

5.4 软件系统

动态目标抓取原型系统的软件系统包括 Ubuntu 系统、机器人操作平台 ROS2，相关的控制程序包括 Vicon 轨迹信息发布程序、轨迹预测程序、运动控制程序、机械臂运动执行程序。

5.4.1 软件平台

Robot Operating System 2 (ROS2) 是一个为机器人软件开发而设计的开源框架，继承并扩展了原始 ROS 平台的功能。ROS2 旨在提供一个统一的软件开发环境，以促进机器人应用的快速开发和部署。它支持多种编程语言，并通过一系列的工具、库和约定促进了高级别的模块化和可重用性。与原始 ROS 相比，ROS2 引入了改进的通信机制，支持实时数据处理和控制。这对于要求高实时性的机器人抓取任务来说，是一个显著的优势，它确保了系统能够及时响应外部事件和内部状态变化。

得益于 ROS2 的高实时性与灵活性，本实验软件平台选择了基于 ubuntu22.04 的 ROS2 (rolling) 版本并使用 python3.8 进行程序开发。通过利用 ROS2 提供的工具和库，极大地促进了软件开发的高效性和模块化。ROS2 丰富的软件包和预构建组件简化了开发流程，使得团队能够快速实现复杂的机器人功能，包括但不限于传感器数据处理、运动控制和执行控制。此外，ROS2 的可扩展性和灵活性也为未来的系统升级和功能扩展提供了便利，确保了研究项目的长期可持续发展。

5.4.2 软件框架

动态目标抓取原型系统基于 ROS2 由 Python 构建，其软件框架如图 5-6 所示，包括三个核心部分：感知预测、运动决策、运动控制（机械臂、灵巧手）。

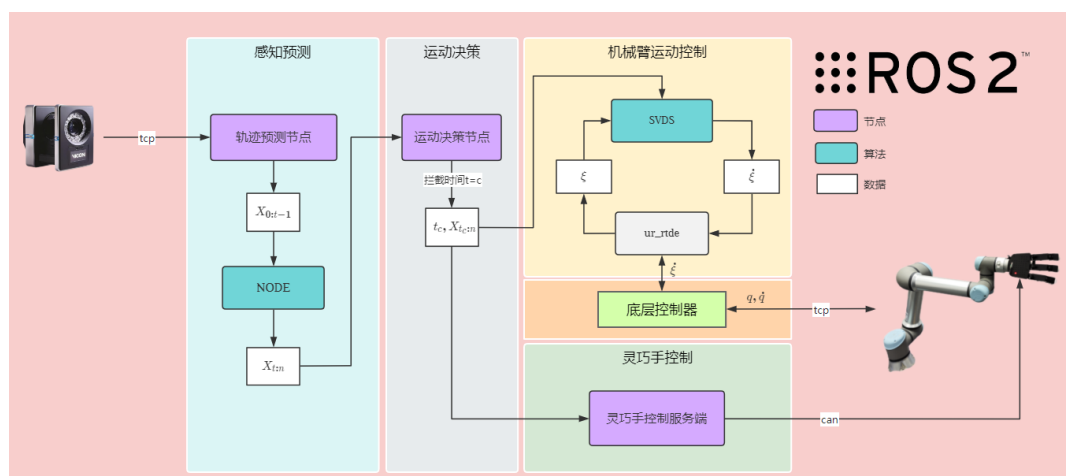


图 5-6 动态目标抓取原型系统软件框架

（1）感知预测模块

此模块的核心功能是预测飞行物体的运动轨迹。有效的轨迹预测对于机器人抓取任务的成功至关重要，因为它直接影响到机器人的响应时间和抓取准确度。轨迹预测模块首先通过 UDP 获取 Vicon 动捕系统的物体实时位置数据，然后通过第三章中提出的基于神经微分方程网络（NODE）预测未来的运动轨迹，最后将轨迹预测的结果以话题（Topic）的形式进行发送。

（2）运动决策模块

基于轨迹预测结果，此模块负责制定机械臂和灵巧手的最佳动作策略。该模块首先通过话题订阅的形式获取轨迹预测的结果，然后计算轨迹与机械臂工作空间内的一块平面的交点，将该交点设为抓取物体的目标位置。同时估计两个时间，一个是目标物体运动到目标位置的时间 T_1 ，另外一个为机械臂末端运行到目标位置的时间 T_2 ，只有当 $T_2 < T_1$ 时机械臂运动才会执行。当 $T_1 = T_{caught}$ ，灵巧手开始闭合，其中 T_{caught} 约为灵巧手从完全张开到半闭合的时间，准确的 T_{caught} 需要经过多次实验进行调整。最后通过话题发布机械臂运动的目标位姿，通过客户端向 Allegro Hand 服务端发送闭合指令。

（3）运动控制模块

最后，运动控制模块负责将决策模块制定的动作策略转化为具体的机器动作。这涉及到对机械臂和灵巧手的精确控制，确保它们能够按照预定的轨迹和动作顺利完成任务。运动控制模块涉及机械臂与灵巧手两部分的硬件控制，其中机械臂在接受到目标位姿后，调用第四章中提出的 SVDS 生成机械臂下一步的末端速度控制指令，然后速度控制指令通过 ur_rtde 库送到底层控制器中执行运动。灵巧手服务端在接受到闭合指令后生成相应的 CAN 控制帧送到灵巧手底层控制器中控制手掌闭合。

动态目标抓取原型系统软件框架通过将抓取任务进行分解为三个部分，将每一部分进行模块化来增加灵活性，保证了系统的模块间的强耦合，通过对每个模块的进一步优化和精细调整，可以提高系统的整体性能和可靠性，以实现高效和稳定的硬件控制。

5.5 实验及分析

5.5.1 实验设计

在完成机器人抓取系统硬件系统与软件系统的设计与实现后，继续通过实物实验来进行相关的系统验证，选择的实验对象需要满足机械手的抓握尺寸要求。

5.5.2 实验过程

(1) 运动物体接取实验

为了验证系统对于环境变化的适应能力，特别是对于快速移动目标的跟踪能力，本章先将抓取任务简化为接取任务，首先在 UR5e 机械臂末端加装一个直径为 20cm 的网兜，在示教器上将末端质心移动到网兜的正中心。图 5-7、图 5-8 为两次运动物体接取实验过程中的截取片段，每张图的左下角标记了记录的顺序，从不同的角度展示了实验过程。

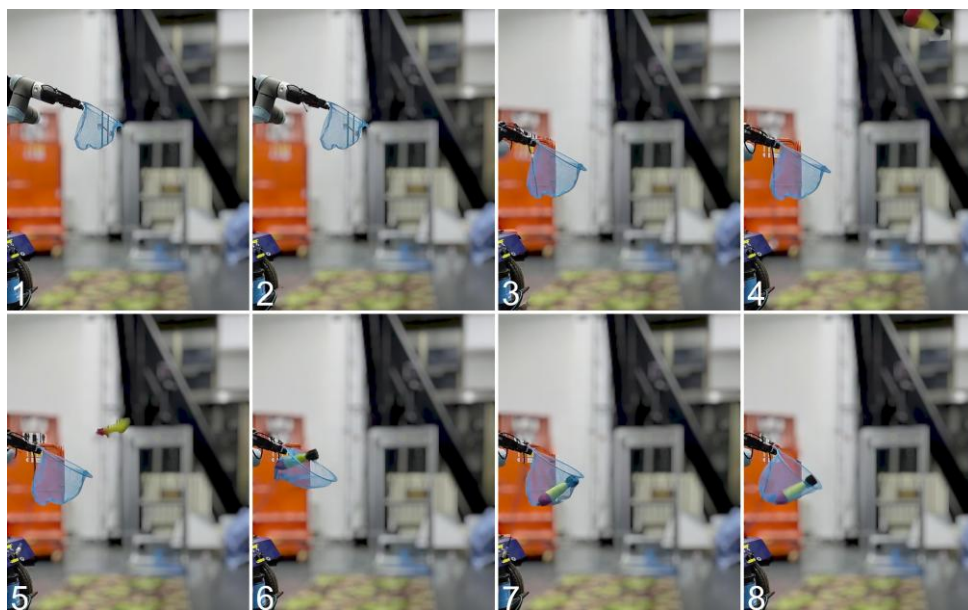


图 5-7 运动物体接取 1

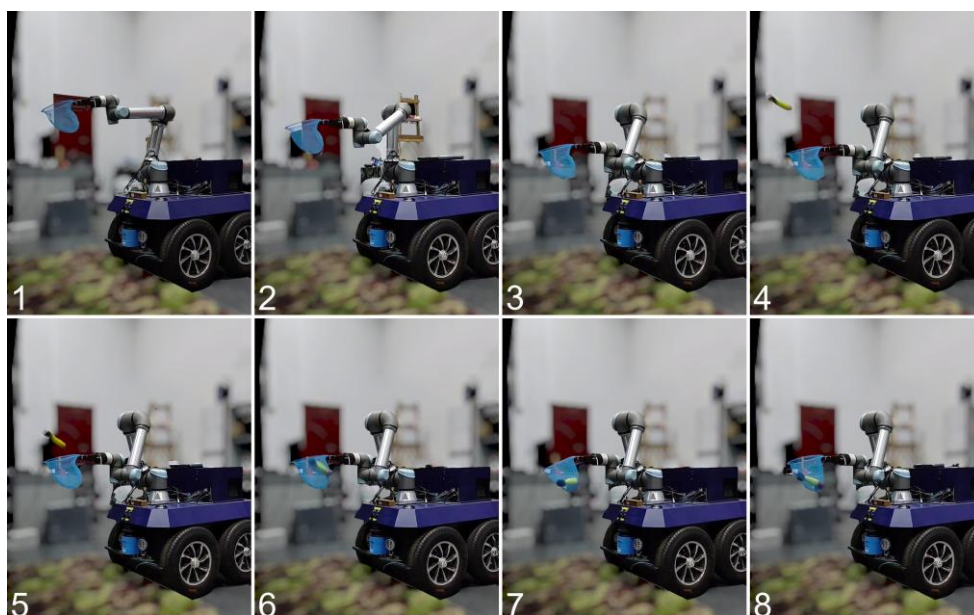


图 5-8 运动物体接取 2

(2) 运动物体抓取实验

为了验证原型系统对快速运动物体轨迹预测的准确性与在高动态环境下运动控制的精确性，本章将通过灵巧手实现的飞行物体的空中抓取。实验中使用 Allegro Hand 成功实现对飞行物体的抓取，图 5-9、图 5-10 展示了两次成功的灵巧手抓取实验过程。

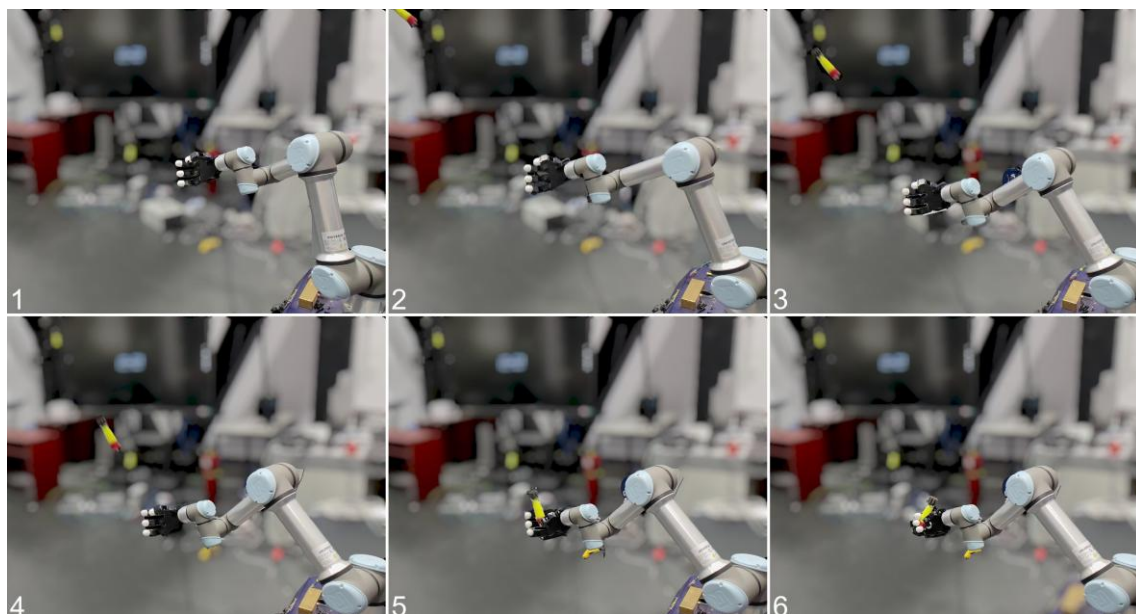


图 5-9 运动物体抓取 1

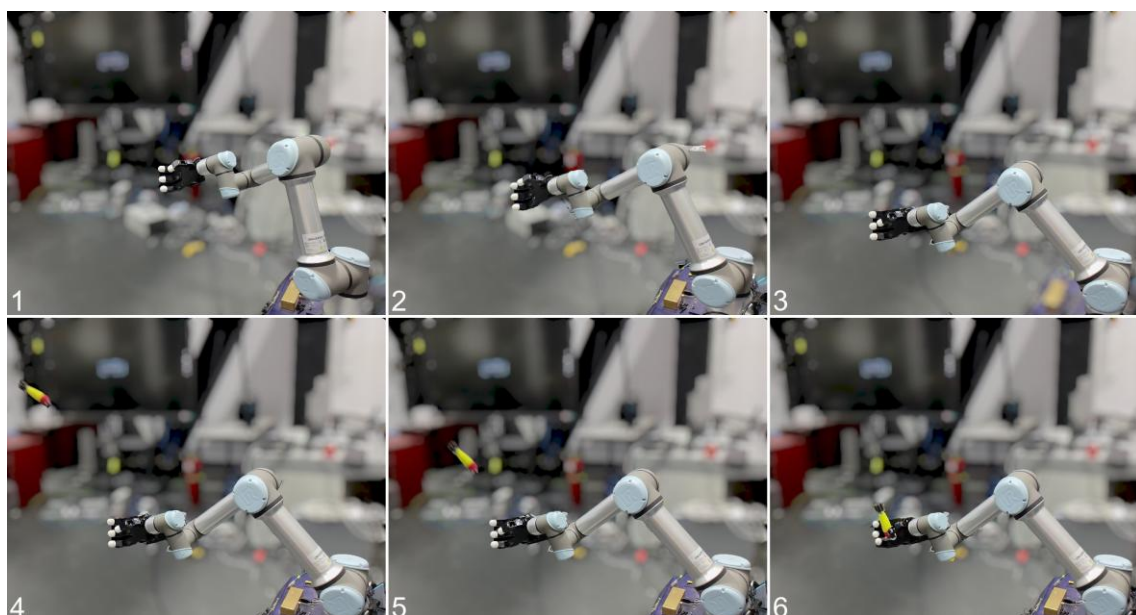


图 5-10 运动物体抓取 2

图中详细记录了机器人接取非规则飞行物体的连续动作。每张图片记录了机器人在不同时间点的位置，显示了从识别目标物体、估算其轨迹，到最终执行接取动作的连贯流程。通过这些连续的图像，可以观察到机器人在面对快速移动物体时的响应速度和准确性。

(3) 运动物体抓取失败实验

在运动物体抓取实验中失败的案例如图 5-11、图 5-12 中所示。

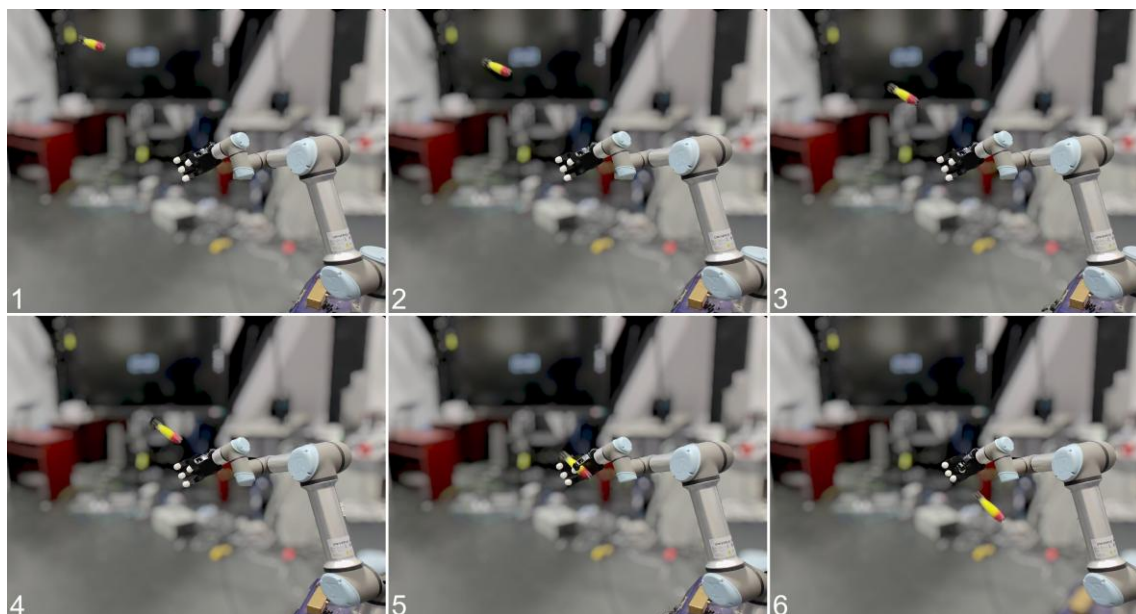


图 5-11 失败案例 1

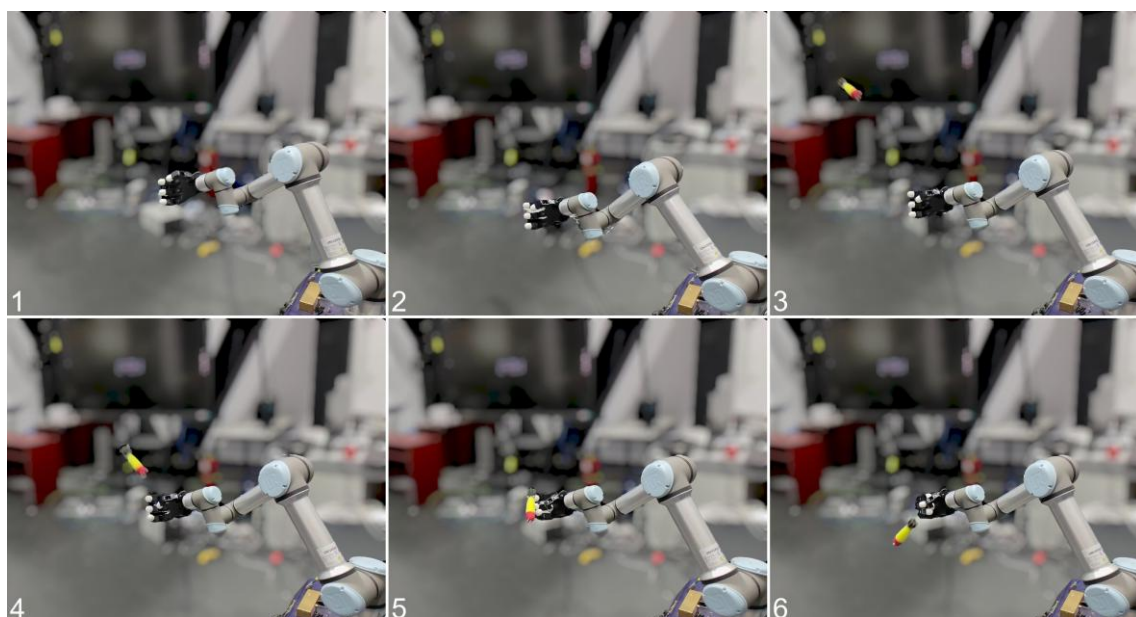


图 5-12 失败案例 2

在失败案例 1 中，由于灵巧手闭合时间估计错误导致运动物体从手掌中穿过，虽然这是一个失败的案例，但是从物体准确穿过灵巧手掌心这一观测证明了飞行物体轨迹预测算法的高精度。失败案例 2 展现了运动物体在与灵巧手指尖碰撞后偏离落地的过程，其中失败的原因可能是两方面的，一方面是飞行物体轨迹预测的误差，另一方面是由于对到达时间的估计错误导致手掌提前闭合。

5.5.3 实验分析

为了对实验进行科学客观的分析，本研究对不同抓取策略的成功率进行了统计，如表 5-4 所示。

表 5-4 实验成功率统计

测试类型	成功次数/实验次数	成功率
运动物体接取实验	15/20	75%
运动物体抓取实验	8/20	40%

经统计物体抛出的平均速度约为 4.2m/s，到达速度约为 5.2m/s，实际抓取频率约为 0.65s。在两个实验的对比中，网兜接取的成功率显著高于灵巧手抓取，分别为 75%和 40%。

分析失败案例的主要原因有以下四点。

（1）接取点距离过远

失败案例中，一部分是因为接取点位于机械臂的初始位置较远处，尤其是靠近机械臂工作空间的外沿，导致机械臂无法迅速到达预定位置。

（2）碰撞导致弹出

在机械臂移动过程中，有物体碰撞到网兜的圆形外沿，并从中心弹出，导致抓取失败。

（3）轨迹数据缺失

物体在飞行过程中产生的轨迹数据出现较多缺失情况，这影响了对其未来位置的精确预测，从而使得机械臂无法正确接取物体。

（4）闭合时间问题

在抓取实验中，一部分失败是因为灵巧手闭合过早，导致物体被弹开；另一部分则是因为闭合太晚，导致物体已经穿过了手掌，灵巧手未能成功抓取。

本实验的分析揭示了不同抓取策略的效率及其局限性。接取实验虽然具有较高的成功率，但在特定条件下（如接取点距离、碰撞、数据缺失等问题）仍然面临挑战。而灵巧手抓取的成功率较低，主要是由于对灵巧手闭合时机控制不当造成的，这要求对灵巧手的控制策略进行更为精细的调整和优化。

5.6 本章小结

为了验证所提出算法的有效性，本章在实际的环境中完成了动态目标抓取原型系统设计与实现。基于动态目标抓取任务的需求搭建硬件系统与软件系统，通过 ROS2 的通信机制实现软硬件的通信连接，确保系统能够及时响应外部事件和内部状态变化，同时能够保持系统灵活性以及可拓展性。通过机器人两阶段实验证明了本文所提出的算法的高效性，能够在真实机器人上完成动态目标抓取任务。

第六章 总结与展望

6.1 总结

本文综合探讨了动态目标抓取任务中的关键挑战，包括动态目标轨迹预测、机械臂的敏捷运动控制，以及动态目标抓取原型系统的构建。为了提升机器人在捕捉非规则物体方面的效率与准确度，将研究重点放在了动态目标轨迹预测与机械臂敏捷运动控制这两项核心技术上。本论文的主要研究内容与贡献总结如下：

（1）由于非规则物体在自由飞行过程中受到质心变化和旋转产生的非线性力影响，传统算法在预测轨迹时表现出明显的局限性。本文采用数据驱动的方法，通过结合现有物理微分方程与深度神经网络来学习飞行物体的复杂动力学模型提出了一种基于神经微分方程网络的动态目标轨迹预测方法。先通过滤波模块对数据进行卡尔曼滤波降噪，再通过 NODE 估计模块产生估计轨迹，最后通过 NoiseNet 补偿模块对估计结果进行准确补偿。即便在样本数据较少的情况下，本文提出的方法也能实现超越先前研究的预测性能，这充分证明了模型对复杂动力学模型的学习和适应能力。

（2）在机械臂的运动控制方面，以往的一些研究算法在敏捷性、适应性、稳定性与可解释性上存在着一些问题。本文针对存在的问题提出了一种基于条件变分自编码器与输入凸神经网络构建的稳定变分动态系统（SVDS）。SVDS 包括一个基于条件变分自编码器的动态系统估计项用于产生平滑的控制输出与一个基于输入凸神经网络的动态系统稳定项用于调节估计项，产生稳定的控制指令。并通过一系列的仿真实验和实物实验充分验证了算法的有效性与稳定性。

（3）为了客观评价本文在动态目标轨迹预测与机械臂敏捷运动控制上所提出算法的性能，本研究在真实世界构建了一个动态目标抓取原型系统。实验结果显示，系统不仅能够准确预测物体的轨迹，还能够通过稳定的运动控制快速响应，成功地完成抓取任务。这些实验不仅验证了算法的有效性，也为未来在更加复杂的环境中提升机器人的自主性提供了坚实的基础。

本文在动态目标抓取任务中引入了几项创新性的技术方法，显著提升了机器人系统的预测精度和运动控制效果。通过采用神经微分方程网络成功模拟了非规则物体在空中的复杂动力学行为，并结合卡尔曼滤波和 NoiseNet 补偿模块显著提升了轨迹预测的准确性，尤其在数据样本较少的情况下表现出色。同时，本文提出的稳定变分动态系统（SVDS）方法通过融合条件变分自编码器与输入凸神经网络，不仅保证了机械臂运动的平滑连续性，还确保了运动控制的稳定性和可

靠性。此外，通过在真实环境中对动态目标抓取原型系统的测试，本研究验证了所提方法的有效性和实用性，为机器人技术在处理更复杂的实际应用任务中的潜在应用提供了坚实基础。

6.2 展望

虽然本文提出的基于深度学习的轨迹预测方法和基于模仿学习的机械臂运动控制方法取得了初步成果，但为进一步提升机器人在动态环境下的物体抓取成功率，还需开展更多研究，主要包括：

（1）姿态的预测：尽管当前研究取得了一定进展，但在飞行物体姿态变换的预测上还有所欠缺。未来的工作需要考虑物体姿态的实时预测，因为在实际的机器人抓取任务中，物体的姿态变化直接影响到机械臂的目标姿态设定和灵巧手的抓取动作。姿态预测的准确性不仅关系到抓取成功的可能性，也是优化机械臂动作策略的关键因素。

（2）抓取策略的优化：当前机械臂的抓取策略主要基于物体移动到预测位置的时间差来设定，这在处理更快速度的物体时可能会出现不适应的情况。因此，需要开发更加智能化的控制算法，动态调整机械臂的抓取时机和动作，以适应物体速度的变化，进一步提高抓取的成功率和效率。

（3）深度学习网络的鲁棒性：尽管本文的方法通过变分编码器从贝叶斯理论的角度改进了模仿学习算法，相较于传统机器学习技术在泛化性方面已有显著提升，但在泛化性上仍存在固有局限性。未来的研究将探索数据增强技术，以提供更多样化的训练样本，同时改进网络结构，进一步提高模型在新场景下的适应能力和鲁棒性。这些措施预期能够有效地扩展模型的应用范围并提高其在动态环境中的表现。

参考文献

- [1] Hong W, Slotine J-J. Experiments in hand-eye coordination using active vision[J]. Experimental Robotics IV, 1997: 130-139.
- [2] Riley M, Atkeson CG. Robot catching: Towards engaging human-humanoid interaction[J]. Auton Robots, 2002, 12(1): 119-128.
- [3] Zhang M, Buehler M. Sensor-based online trajectory generation for smoothly grasping moving objects[C]. Proceedings of the 1994 IEEE International Symposium on Intelligent Control. IEEE, 1994: 141-146.
- [4] Nishiwaki K, Ianno A, Nagashima K, et al. The humanoid saika that catches a thrown ball[C]. Proceedings, 6th IEEE International Workshop on Robot and Human Communication, RO-MAN'97, Sendai, Japan, 1997: 94-99.
- [5] Frese U, Bauml B, Haidacher S, et al. Off-the-shelf vision for a robotic ball catcher[C]. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, 2001, vol. 3: 1623-1629.
- [6] Park GR, Kim K, Kim C, et al. Human-like catching motion of humanoid using evolutionary algorithm(EA)-based imitation learning[C]. Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan, 2009: 809-815.
- [7] Bauml B, Wimbock T, Hirzinger G. Kinematically optimal catching a flying ball with a hand-arm-system[C]. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010: 2592-2599.
- [8] Gold T, Römer R, Völz A, et al. Catching objects with a robot arm using model predictive control[C]. 2022 American Control Conference (ACC), Atlanta, 2022: 1915-1920.
- [9] Jia Y-B, Gardner M, Mu X. Batting an in-flight object to the target[J]. The International Journal of Robotics Research, 2019, 38(4): 451-485.
- [10] Gardner M, Jia Y-B. Pose and motion estimation of free-flying objects: Aerodynamics, constrained filtering, and graph-based feature tracking[J]. IEEE Transactions on Robotics, 2022, 38: 3187-3202.
- [11] Kim S, Billard A. Estimating the non-linear dynamics of free-flying objects[J]. Robotics and Autonomous Systems, 2012, 60(9): 1108-1122.
- [12] Kim S, Shukla A, Billard A. Catching objects in flight[J]. IEEE Transactions on Robotics, 2014, 30(5): 1049-1065.

- [13] Bäuml B, Wimböck T, Hirzinger G. Kinematically optimal catching a flying ball with a hand-arm-system[C]. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010: 2592-2599.
- [14] Barker AL, Brown DE, Martin WN. Bayesian estimation and the Kalman filter[J]. Computers & Mathematics with Applications, 1995, 30(10): 55-77.
- [15] Payeur P, Le-Huy H. Trajectory prediction for moving objects using artificial neural networks[J]. IEEE Transactions on Industrial Electronics, 1995, 42(2): 147-158.
- [16] Williams R J. A learning algorithm for continually running fully recurrent neural networks[J]. Neural Computation, 1989, 1: 256-263.
- [17] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [18] Shi Z, Xu M, Pan Q, et al. LSTM-based flight trajectory prediction[C]. Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018: 1-8.
- [19] Chen R, Chen M, Li W, et al. Predicting future locations of moving objects by recurrent mixture density network[J]. International Journal of Geo-Information, 2020, 9(2): 116.
- [20] Schaal S. Is imitation learning a route to humanoid robots?[J]. Trends in Cognitive Sciences, 1999, 3: 233-242.
- [21] Zhang Z, Guo D, Zhou S, et al. Flight trajectory prediction enabled by time-frequency wavelet transform[J]. Nature Communications, 2023, 14(1): 5258.
- [22] Gomez-Gonzalez S, Prokudin S, Schölkopf B, et al. Real time trajectory prediction using deep conditional generative models[J]. IEEE Robotics and Automation Letters, 2020, 5: 970-976.
- [23] Gomez-Gonzalez S, Nemmour Y, Schölkopf B, et al. Reliable real-time ball tracking for robot table tennis[J]. Robotics, 2019, 8: 90.
- [24] Wang Y, Wang L. Machine vision-based ping pong ball rotation trajectory tracking algorithm[J]. Computational Intelligence and Neuroscience, 2022, 2022: 1-12.
- [25] 雷校东. 基于长短时记忆网络的自由飞行运动物体姿态与位置预测[D]. 郑州: 郑州大学, 2021.
- [26] Qadeer N, Shah JH, Sharif M, et al. Intelligent tracking of mechanically thrown objects by industrial catching robot for automated in-plant logistics 4.0[J]. Sensors, 2022, 22: 2113.
- [27] Ijspeert AJ, Nakanishi J, Hoffmann H, et al. Dynamical movement primitives: learning attractor models for motor behaviors[J]. Neural Computation, 2013, 25(2): 328-373.

- [28] Paraschos A, Daniel C, Peters J, et al. Probabilistic movement primitives[C]. Proceedings of the 26th International Conference on Neural Information Processing Systems, Nevada, USA, 2013: 2616-2624.
- [29] Huang YL, Rozo L, Silverio J, et al. Kernelized movement primitives[J]. The International Journal of Robotics Research, 2019, 38(7): 833-852.
- [30] Calinon S, Guenter F, Billard A. On learning, representing, and generalizing a task in a humanoid robot[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007, 37(2): 286-298.
- [31] Calinon S, Bruno D, Caldwell DG. A task-parameterized probabilistic model with minimal intervention control[C]. Proceedings of the 2014 IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014: 3339-3344.
- [32] 高霄.基于模仿学习的机械臂运动规划与柔顺控制研究[D].武汉: 武汉大学, 2021.
- [33] Y. Zhang, L. Cheng, R. Cao, et al. A neural network based framework for variable impedance skills learning from demonstrations[J]. Robotics and Autonomous Systems, 2023, 160: 104312.
- [34] Manek G, Kolter J Z. Learning stable deep dynamics models[C]. In: Conference on Neural Information Processing Systems, 2019.
- [35] Zhi W, Lai T, Ott L, et al. Learning efficient and robust ordinary differential equations via invertible neural networks[C]. In: International Conference on Machine Learning, PMLR, 2022: 27060-27074.
- [36] Khansari-Zadeh SM, Billard A. Learning stable nonlinear dynamical systems with Gaussian mixture models[J]. IEEE Transactions on Robotics, 2011, 27(5): 943-957.
- [37] Khansari-Zadeh SM, Billard A. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions[J]. Robotics and Autonomous Systems, 2014, 62(6): 752-765.
- [38] Neumann K, Steil JJ. Learning robot motions with stable dynamical systems under diffeomorphic transformations[J]. Robotics and Autonomous Systems, 2015, 70: 1-15.
- [39] Urairi J, Ginesi M, Tateo D, et al. ImitationFlow: learning deep stable stochastic dynamic systems by normalizing flows[C]. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020: 5231-5237.
- [40] Rana M, Li A, Fox D, et al. Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems[C]. In: Proceedings of the 2nd Conference on Learning for Dynamics and Control, PMLR, 2020: 630-639.

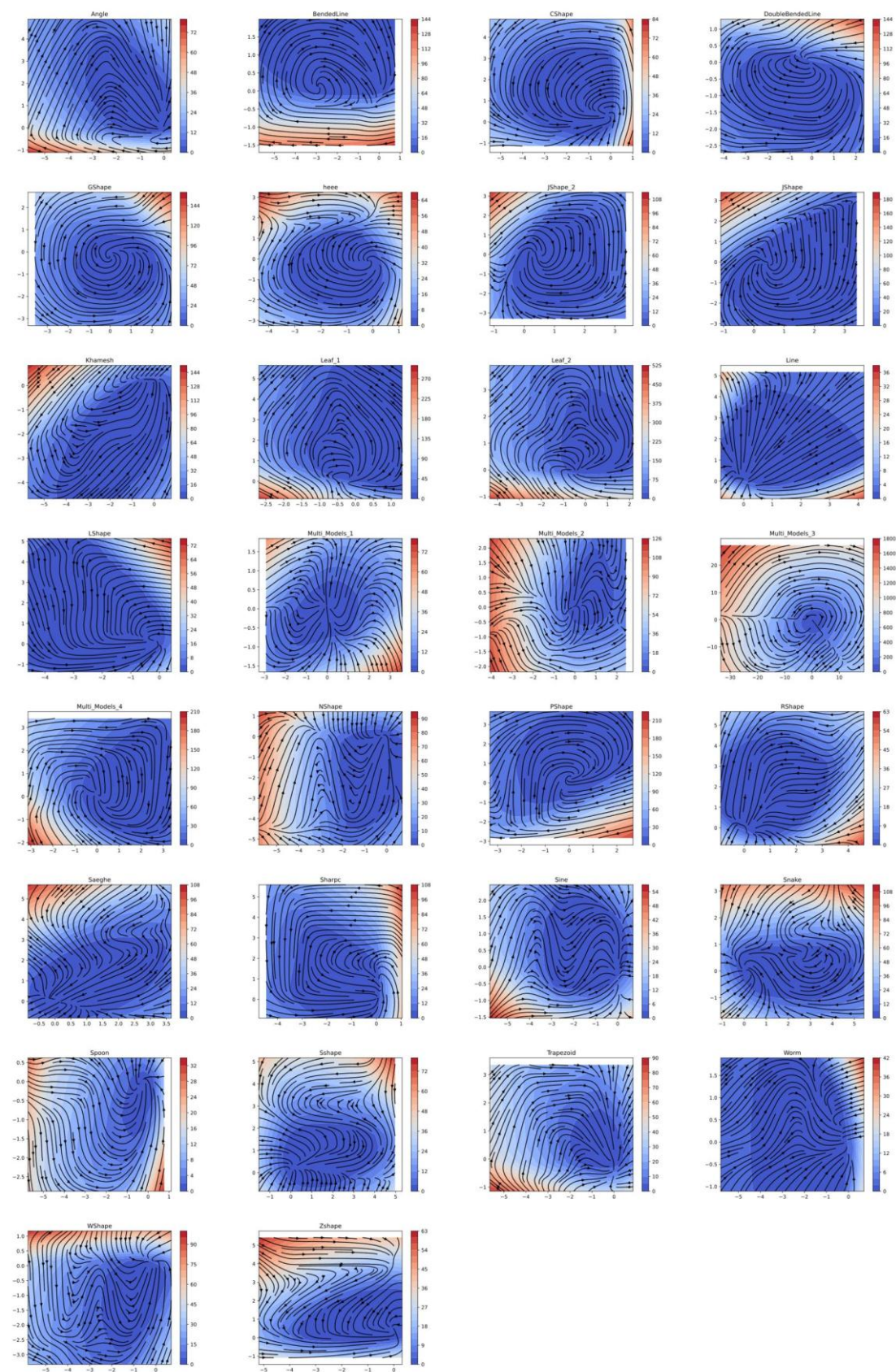
- [41] Pérez-Dattari R, Kober J. Stable motion primitives via imitation and contrastive learning[J]. IEEE Transactions on Robotics, 2023.
- [42] 黄艳龙, 徐德, 谭民. 机器人运动轨迹的模仿学习综述[J]. 自动化学报, 2022, 48(2): 315-334.
- [43] Lemme A, Neumann K, Reinhart RF, et al. Neural learning of vector fields for encoding stable dynamical systems[J]. Neurocomputing, 2014, 141: 3-14.
- [44] Yu H, Guo D, Yin H, et al. Neural motion prediction for in-flight uneven object catching[C]. Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021: 4662-4669.
- [45] Chen RTQ, Rubanova Y, Bettencourt J, et al. Neural ordinary differential equations[J]. Advances in Neural Information Processing Systems, 2018, 31.
- [46] Dannemiller J L, Babler T G, Babler B L, et al. On catching fly balls[J]. Science, 1996, 273(5272): 256-258.
- [47] Anderson JD Jr. Fundamentals of Aerodynamics Sixth Edition[M]. New York: McGraw-Hill Education, 2017.
- [48] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.
- [49] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures[J]. Neural computation, 2019, 31(7): 1235-1270.
- [50] Butcher J C. Numerical methods for ordinary differential equations[M]. John Wiley & Sons, 2016.
- [51] Wong AL, et al. Motor Planning[J]. The Neuroscientist, 2015, 21: 385-398.
- [52] Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models[J]. Advances in neural information processing systems, 2015, 28.
- [53] Amos B, Xu L, Kolter J Z. Input convex neural networks[C]. International Conference on Machine Learning. PMLR, 2017: 146-155.
- [54] Billard A, Hayes G. DRAMA, A connectionist architecture for control and learning in autonomous robots[J]. Adaptive Behavior, 1999, 7(1): 35-63.
- [55] Artstein Z. Stabilization with relaxed controls[J]. Nonlinear Analysis: Theory, Methods & Applications, 1983, 7(11): 1163-1173.
- [56] 夏超英. 现代控制理论[M]. 科学出版社, 2016.
- [57] Chen J, Shen S, Lau HYK. Hitting flying objects with learning from demonstration[C]. 2017 18th International Conference on Advanced Robotics (ICAR), Hong Kong, 2017: 55-60.

- [58] Kober J, Mülling K, Krömer O, et al. Movement templates for learning of hitting and batting[C]. In: Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 2010: 853-858.
- [59] Kober J, Glisson M, Mistry M. Playing catch and juggling with a humanoid robot[C]. In: Proceedings of the 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Osaka, Japan, 2012: 875-881.
- [60] Salehian S, Khoramshahi M, Billard A. A dynamical system approach for softly catching a flying object: theory and experiment[J]. IEEE Transactions on Robotics, 2016, 32: 462-471.
- [61] 李亮.基于神经常微分方程的时序数据分析与预测[D].成都: 电子科技大学, 2021.
- [62] Rubanova Y, Chen R T Q, Duvenaud D K. Latent ordinary differential equations for irregularly-sampled time series[J]. Advances in neural information processing systems, 2019, 32.
- [63] Kidger P, Morrill J, Foster J, et al. Neural controlled differential equations for irregular time series[J]. Advances in Neural Information Processing Systems, 2020, 33: 6696-6707.
- [64] Anand A, Seel K, Gjørsum V, et al. Safe learning for control using control Lyapunov functions and control barrier functions: A review[J]. Procedia Computer Science, 2021, 192: 3987-3997.
- [65] Marvi Z, Kiumarsi B. Safe reinforcement learning: A control barrier function optimization approach[J]. International Journal of Robust and Nonlinear Control, 2021, 31(6): 1923-1940.
- [66] Fan D D, Nguyen J, Thakker R, et al. Bayesian learning-based adaptive control for safety critical systems[C]. International Conference on Robotics and Automation (ICRA), IEEE, 2020: 4093-4099.
- [67] Khalil H K, Grizzle J W. Nonlinear systems[M]. Upper Saddle River, NJ: Prentice Hall, 2002.
- [68] Farzan S, Azimi V, Hu A-P, et al. Adaptive Control of Wire-Borne Underactuated Brachiating Robots Using Control Lyapunov and Barrier Functions[J]. IEEE Transactions on Control Systems Technology, 2022, 30(6): 2598-2614.

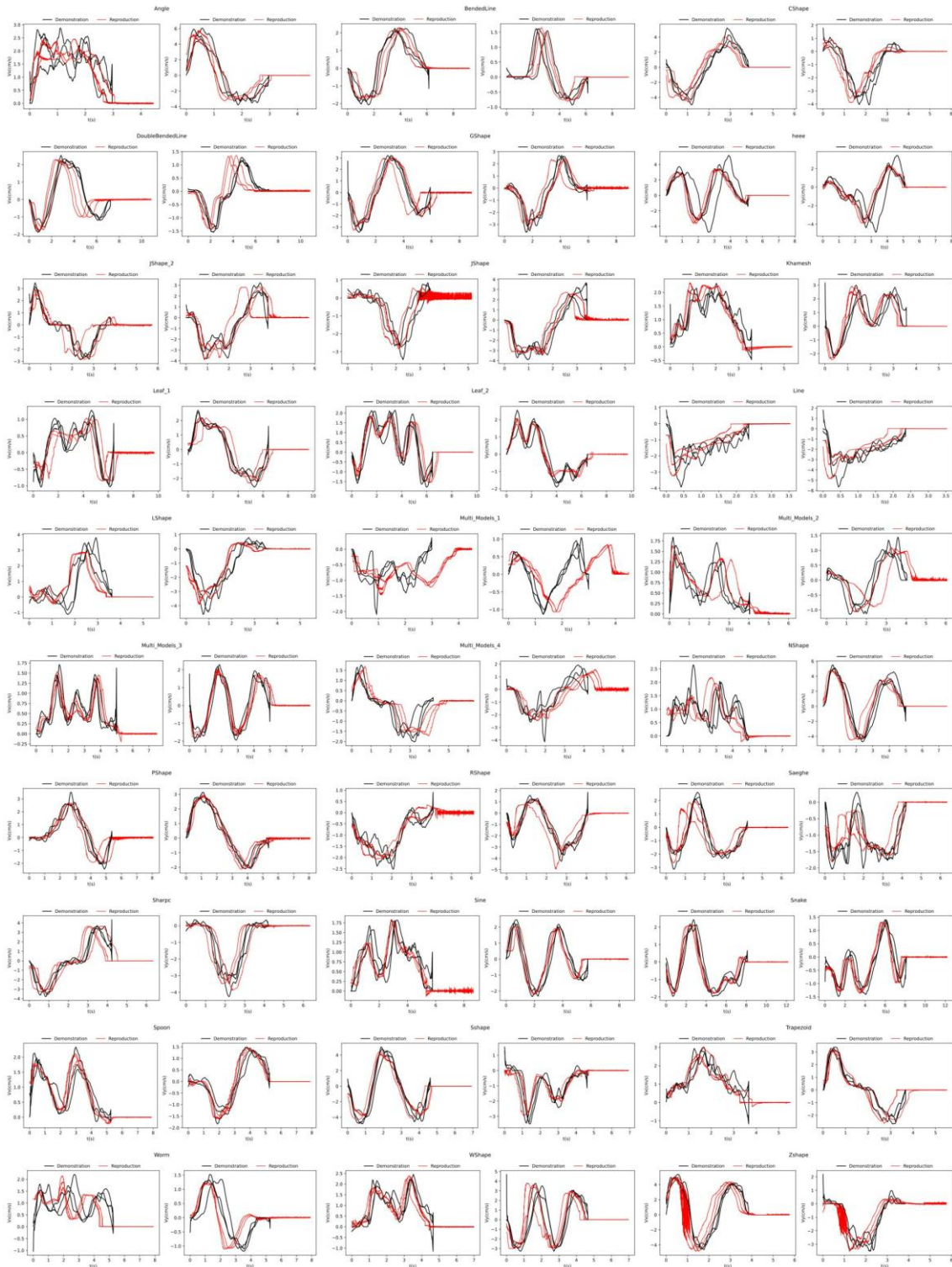
附录 A 手写数据集实验对比图



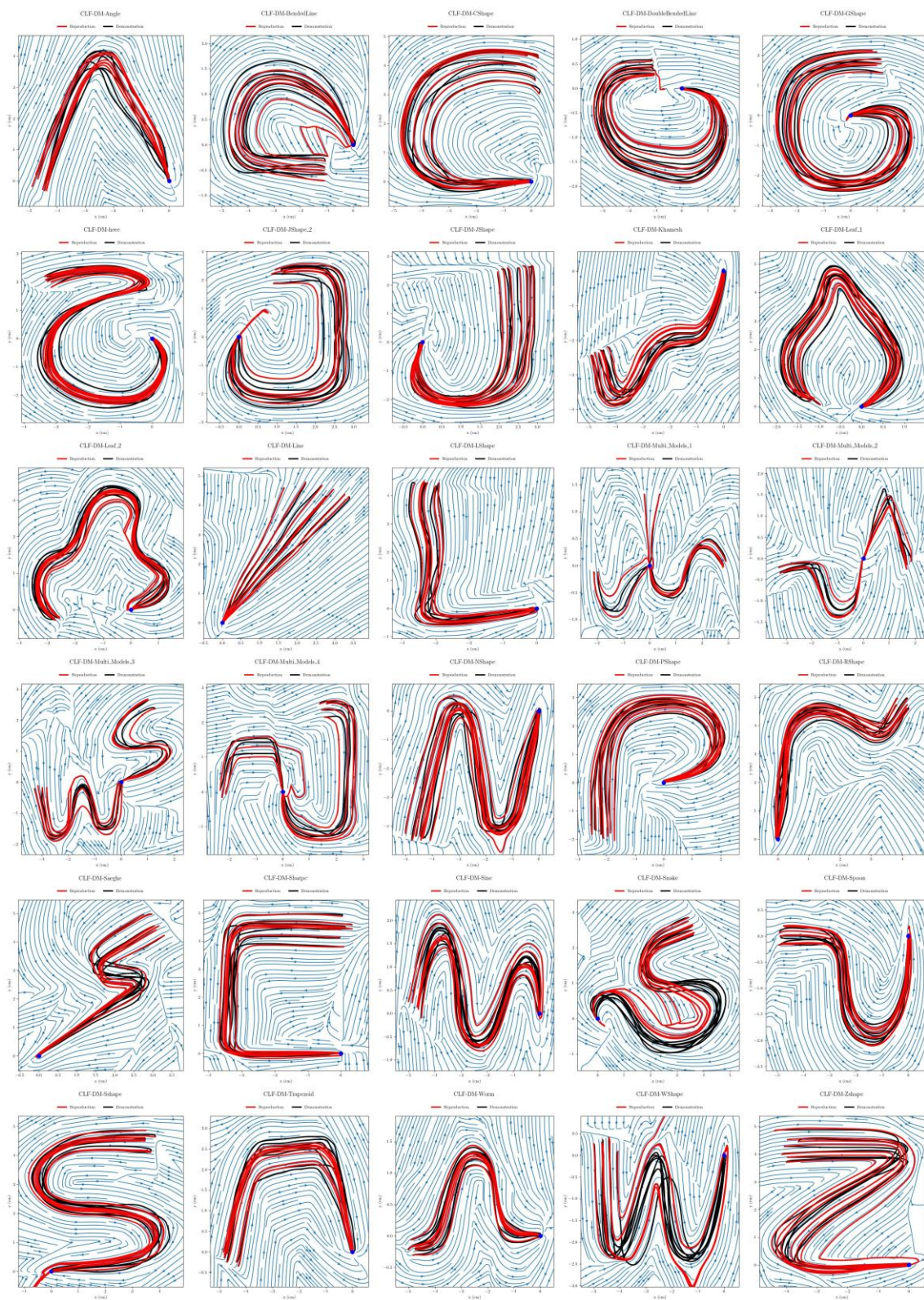
LASA 复现轨迹



LASA 李雅普诺夫函数能量场



LASA 速度变化曲线



CLF-DM 复现轨迹