面向行为交互的机器鼠运动规划研究

AL-KHULAQUI MOHAMED
ABDULRAB JABER

2023 年 6 月

面向行为交互的机器鼠运动规划研究

AL-KHULAQUI MOHAMED ABDULRAB JABER

北京理工大学

中国分类号：TP242.6
UDC 分类号：621

面向行为交互的机器鼠运动规划研究

作 者 姓 名　AL-KHULAQUI MOHAMED ABDULRAB JABER
学 院 名 称　机电学院
指 导 教 师　石青教授
答辩委员会主席　宋萍教授
申请学位级别　工学硕士
学 科 专 业　机械工程
学位授予单位　北京理工大学
论文答辩日期　2023 年 06 月

Motion Planning Framework of Robotic Rat for Behavioral Interaction

Candidate Name:	AL-KHULAQUI MOHAMED ABDULRAB JABER
School or Department:	Mechatronics
Faculty Mentor:	Prof. Ping Song
Chair, Thesis Committee:	Prof. Qing Shi
Degree Applied:	Master of Engineering
Major:	Mechanical Engineering
Degree by:	Beijing Institute of Technology
The Date of Defence:	06-2023

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签　　名：　　日　期：

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：① 学校有权保管、并向有关部门送交学位论文的原件与复印件；② 学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③ 学校可允许学位论文被查阅或借阅；④ 学校可以学术交流为目的, 复制赠送和交换学位论文；⑤ 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签　　名：　　日　期：

指导老师签名：　　日　期：

摘要

机器鼠形态仿生、运动模式可控，可替代真实动物开展行为交互，在生物医学领域具有重要的研究价值。在交互过程中实时产生仿鼠运动并稳定、快速跟踪动物是自主行为交互实验的必要条件。目前用于动物行为交互实验的机器人存在着运动数据表示空间不一致、运动表达仿生相似性低和运动控制跟踪能力差的问题。本文提出面向仿生机器鼠 SMuRo 行为交互的数据驱动运动规划方法，实现了交互过程中对实验鼠的实时跟踪与仿鼠机器人的仿生运动生成，通过行为交互实验验证了所提方法的有效性。本文的主要研究内容如下：

首先，针对运动数据表示空间不一致的问题，构建了实验鼠-机器鼠运动数据映射。分析实验鼠和机器鼠结构特征的内在关系，提出了最小化二者脊柱轮廓曲线的欧氏距离的映射方法，引入基于非线性优化的曲线拟合算法，将不同的行为模式下实验鼠运动数据映射到机器鼠关节空间，实现了运动数据的统一描述。

其次，针对运动表达仿生相似性低的问题，利用高斯模型的随机性，在机器鼠关节空间，建立与实验鼠运动数据一致分布的概率模型，生成具备相同运动趋势的轨迹。采用最大似然估计计算概率模型的参数，通过控制经过点与轨迹过渡范围，实现了不同行为模式下机器人仿生运动轨迹的生成。

然后，针对运动控制跟踪能力差的问题，利用 SMuRo 集成的仿鼠视觉感知模块，根据视觉检测结果，提出了最小化视觉中心点与交互目标距离的跟踪方法；通过约束图像到关节速度的雅可比矩阵以及估计机器鼠躯干与基座的偏航角度，实现了机器鼠的整体协调运动控制，完成了快速、稳定的实时目标跟踪。

最后，在仿真和交互实验中均验证了上述方法的有效性。所提出的运动映射算法将机器鼠与实验鼠脊柱轮廓误差降低到 0.5mm 左右，机器鼠执行四种行为模式时运动表达相似度均在 96% 以上，表明了机器鼠具有较高的运动相似性；与实验鼠的交互实验结果表明无论实验鼠是否在机器鼠的视野范围内，机器鼠能够以最快 1.7s 速恢复对其的视觉定位并保持不低于 43s 的主动持续追踪；通过验证概率运动基元与交互行为控制策略的可行性，表明了机器人具备稳定长时间与实验鼠自主行为交互潜力。

关键词：仿生机器人；数据驱动方法；非线性优化；概率动作基元；视觉伺服

Abstract

Robotic rats with bio-mimetic morphology and controllable movement patterns can replace real animals in behavioral interaction experiments and thus, have important research value in the biomedical field. Real-time generation of rat-like motions during interaction and stable and fast tracking of animals are necessary for autonomous behavioral interaction experiments. The current robots used for animal behavioral interaction experiments suffer from spatial inconsistency in motion data representation, low bio-mimetic similarity in motion expression, and poor motion control tracking ability. In this paper, we propose a data-driven motion planning method for the behavioral interaction of the bionic robot mouse SMuRo, and realize real-time tracking of the laboratory rat and bio-mimetic motion generation for the robotic rat during the interaction. Finally, the effectiveness of the method is proved by behavioral interaction experiments. The main contributions of this work is as follows:

First, to address the problem of spatial inconsistency in motion data representation, laboratory rat-to-robotic rat motion data mapping is established. The intrinsic relationship between the structural features of laboratory rats and robotic rats is analyzed, and a mapping method that minimizes the Euclidean distance of the spine contour curves of both is proposed. A curve fitting algorithm based on nonlinear optimization is introduced to map the laboratory rat motion data to the robotic rat joint space under different behavioral patterns, and a unified description of the motion data is realized.

Second, to address the problem of low bio-mimetic similarity of motion expression, the randomness of Gaussian models is used to construct a probabilistic model with a distribution consistent with that of the laboratory rat motion data in order to generate motion trajectories in the joint space of the robotic rat that follow the same patterns. Maximum likelihood estimation is used to calculate the parameters of the probabilistic model, and the generation of continuous motion trajectories of the robot under different behavioral patterns is achieved by controlling the via-points and the transitions of the trajectories.

Then, to address the problem of poor target tracking capability, a tracking method that minimizes the distance between the visual centroid and the interacting target is proposed

based on the visual detection results using the SMuRo integrated rat-like visual perception module. By constraining the Task Jacobian of image-to-joint velocity and estimating the yaw angle between the torso and the base of the robotic rat, the overall coordinated motion of the robotic rat is controlled, and fast and stable real-time target tracking is achieved.

Finally, the effectiveness of the above methods is verified in both simulations and interactive experiments. The motion mapping algorithm reduces the error of spine contour between the robotic rat and the laboratory rat to about 0.5mm, and the similarity of motion expression when the robotic rat performs four behavioral modes is above 96%, which shows that the robotic rat has high motion similarity to the laboratory rat. The results of the interaction experiments with the laboratory rats showed that the robotic rat was able to resume its visual localization as fast as 1.7s and maintain active continuous tracking for no less than 43s, regardless of whether the laboratory rat was within the field of view of the robot.

Key Words: Bio-mimetic Robots; Data-driven methods; Non-linear Optimization; Probabilistic Motion Primitives; Visual Servoing

Table of Contents

摘要 I
Abstract II
Nomenclature X
Chapter 1 Introduction 1
1.1 Motivation 1
1.2 Literature Review 3
1.2.1 Motion Planning of Bio-mimetic Robots 4
1.2.2 Motion Planning for Bio-mimetic Robotic Rats 8
1.2.3 Robot-Animal Behavioral Interaction 10
1.2.4 Existing Problems 12
1.3 Structure of Thesis 13
Chapter 2 Rat-to-Robot Motion Mapping 16
2.1 Overview 16
2.2 Motion Capture Dataset 17
2.2.1 Motion Capture 17
2.2.2 Behavior Classification 18
2.3 Rat-to-Robot Mapping Algorithm 21
2.3.1 Problem Formulation 21
2.3.2 Preprocessing 22
2.3.3 Mapping Process 24
2.3.4 Optimization Problem 28
2.4 Summary 34
Chapter 3 Data-Driven Motion Generation 35
3.1 Overview 35
3.2 Probabilistic Motion Primitives 37
3.2.1 Gaussian Basis Functions 37

3.2.2 Learning from Demonstrations 40
3.3 Properties of Probabilistic Motion Primitives 41
3.3.1 Temporal Modulation 41
3.3.2 Coupled Trajectories 42
3.3.3 Via Points 42
3.3.4 Motion Primitive Operations 43
3.4 Summary 44
Chapter 4 Real-time Rat Tracking 45
4.1 Overview 45
4.2 Robotic Rat Perception System 45
4.2.1 Depth Estimation 47
4.2.2 Laboratory Rat Detection 49
4.3 Laboratory Rat Tracking Algorithm Based on IBVS 51
4.3.1 Target Features Definition 54
4.3.2 Interaction Matrix 55
4.3.3 Target Tracking using the Robotic Rat's Torso 58
4.3.4 Control of the Mobile Base for Tracking 62
4.4 Control Policy for Behavioral Interaction 65
4.4.1 Observation 66
4.4.2 Behavioral Interaction 67
4.5 Summary 67
Chapter 5 Simulation and Experiments 69
5.1 Overview 69
5.2 Rat-to-robot Mapping 69
5.2.1 MO 69
5.2.2 PIN 71
5.2.3 POU 72
5.2.4 SNC 73
5.2.5 Summary 74

5.3 Probabilistic Motion Primitive Trajectories 75
5.4 Rat-Tracking 78
5.4.1 Robotic Platform 78
5.4.2 Simulation & Digital Twin 79
5.4.3 Tracking Experiments 80
5.5 Behavioral Interaction 84
5.6 Summary 85
Conclusions 87
References 90
Publications During Studies 95
Acknowledgement 96

Figures

1.1 Comparsion between traditional experimentation methods and using robotic agents . . . 3
1.2 ETH Zürich's Quadruped Robot ANYmal . . . 4
1.3 Robots Controlled using Gait Planning . . . 5
1.4 Humming Bird . . . 6
1.5 Harvard University's Blueswarm Robots . . . 7
1.6 Quadruped Robotic Rats . . . 8
1.7 Wheeled Robotic Rats . . . 9
1.8 Robots Interacting with Animals . . . 10
1.9 Robots Interacting with Rats . . . 11
1.10 Thesis Structure . . . 14

2.1 Locations of the reflective markers used for motion capture . . . 18
2.2 Setup for laboratory Rat Motion Capture . . . 19
2.3 Behavior Classification based on Clustering . . . 21
2.4 Comparison between Laboratory Rat's body & head and SMuRo's joints. . . . 23
2.5 A Single Frame from the Motion Capture Dataset . . . 24
2.6 Alignment of Motion Data . . . 25
2.7 SMuRo's body Points . . . 26
2.8 Rat contour to Robot Joints Motion Mapping Process . . . 28

3.1 Gaussian Basis Functions as a Linear Combination . . . 39
3.2 Estimating ProMP Model Parameters using MLE . . . 40
3.3 Modulation of Via-Point . . . 43
3.4 Blending Operation of Motion Primitive with 0 . . . 44

4.1 Comparison between Old and New Cameras of the Robotic Rat . . . 46
4.2 Binocular Stereo Camera Model . . . 48
4.3 Extraction of Laboratory Rat Location from Image . . . 50
4.4 Full laboratory Rat Detection Algorithm Based on YOLOv4 and Stereo Cameras . . . 51
4.5 Simplification of Robotic Rat's Model . . . 53
4.6 Feature Error . . . 56
4.7 Effect of Jacobian Constraints on the Robotic Rat's Joints . . . 60
4.8 Full Tracking Range of the Robotic Rat's Torso . . . 63
4.9 Comparison between Base Tracking and Full Body Tracking . . . 64
4.10 Distance control for approaching and withdrawing from laboratory rat . . . 65
4.11 Control Policy for the Robotic Rat . . . 67

5.1 Frames Extracted from MO Mapping Results . . . 70
5.2 Error Distribution of MO Dataset Mapping Results . . . 70
5.3 Frames Extracted from PIN Mapping Results . . . 71

5.4 Error Distribution of PIN Dataset Mapping Results 71
5.5 Frames Extracted from POU Mapping Results 72
5.6 Error Distribution of POU Dataset Mapping Results 72
5.7 Frames Extracted from SNC Mapping Results 73
5.8 Error Distribution of SNC Dataset Mapping Results 73
5.9 Example of High Error Result 74
5.10 Motion Trajectories Generated using ProMP 75
5.11 Target Tracking Simulation Result 79
5.12 Digital Twin Framework 80
5.13 Setup for Conducting Experiments 81
5.14 Basic Tracking Capabilities 81
5.15 Torso & Base Coordinated Tracking 82
5.16 Directional Search Policy 83
5.17 Behavioral Interaction Experiments performing different behaviors 86

Tables

2.1 D-H Parameters of SMuRo 25
2.2 Robotic Rat's body point parameters 26
2.3 Hyperparameters of Optimization Problem 32
2.4 Comparison of Optimization Algorithms 33

4.1 Parameters of Stereo Camera 46
4.2 PID Parameters of Mobile Base Tracking 64

5.1 Motion Capture Dataset 69
5.2 Similarity Analysis of Motion Trajectories 78

Nomenclature

SMuRo	Small-Scale Multi-joint Robotic Rat, 微小型多关节机器鼠
DoF	Degrees of Freedom，自由度
PCA	Principal Component Analysis，主成分分析
CAPTURE	Continuous Appendicular and Postural Tracking Using Retroreflector Embedding，基于反光点嵌入的四肢、姿态持续跟踪
t-SNE	t-Stochastic Neighbor Embedding，基于 t 分布的随机近邻嵌入
KMJ	Key Movement Joints，关键运动关节
GD	Gradient Descent，梯度下降
IPOPT	Interior Point Optimizer，内点法优化器
MP	Motion Primitives，动作基元
ProMP	Probabilistic Motion Primitives，概率动作基元
GBF	Gaussian Basis Functions，高斯基元函数
LfD	Learning from Demonstrations，演示学习
RL	Reinforcement Learning，强化学习
IL	Imitation Learning，模仿学习
CNN	Convolutional Neural Networks，卷积神经网络
FOV	Field of View，视场角
BBox	Bounding Box，边界框
IBVS	Image Based Visual Servoing, 基于图像的视觉伺服
PID	Proportion Integral Differential，比例积分微分

Chapter 1 Introduction

1.1 Motivation

In their long struggle for survival, humans have suffered greatly at the hand of diseases that affect the mind and body, causing mental and physical harm and could even lead to death. For a long time and until now, scientists and doctors have been carefully researching diseases and developing new methods to cure them. However, in order to establish a full understanding of a disease, systemically verify the effectiveness of the cure and study its side effects, extensive research and observation is necessary. Due to the ethical issues with conducting medical tests and research on human subjects, researchers in many industries have opted to use animal models of human diseases to replace humans as the test subjects in these experiments[1].

The behavior animal models exhibit during experimentation is one of the most direct and observable feedback signs about the condition of the animal. Researchers often analyze the test subject's behavioral interaction with normal animals under similar circumstances to diagnose and assess the test subject's condition and the effects the disease/medicine has on its behavior. However, there are many issues that make behavioral interaction experiments between real animal models difficult and unreliable. Since the test subjects cannot be directly controlled, there are too many uncontrollable variables with such experiments, such as low reproducibility, undesired behaviors and lengthy experiment periods. Bio-mimetic Robots that can mimic animals' appearance, movement and behaviors can be used as a replacement for real-rats in certain behavioral experiments to solve these issues.

Using bio-mimetic Robots as a replacement for animal agents in behavioral interaction experiments has the following advantages: (1) Robotic systems are typically designed to closely resemble real animals, which enables them to be readily recognized by target animals as members of the same species. This feature provides a basis for conducting interaction experiments between robots and animals. By creating a sense of familiarity with the robotic system, animals may exhibit more natural behaviors and responses, thereby enhancing the ecological validity of such experiments. (2) Robots provide a significant advantage over live animals due to their ability to follow researcher instructions precisely

and consistently, without variation. This affords scientists a level of control over their experiments that is challenging or unattainable through other means. The ability to conduct experiments with consistent and reproducible conditions can enhance the validity and reliability of research findings.; (3) The bio-mimetic robot can effectively utilize its sensory capabilities, artificial intelligence algorithms, and animal-like motion capabilities to collect data, identify behaviors, and execute responses to specific stimuli in accordance with researcher objectives. By leveraging advanced technologies, such as machine learning and biomimicry, this robotic system can enable researchers to conduct experiments with increased precision and control, while minimizing potential confounding variables. This approach of leveraging Bio-mimetic robots to facilitate animal interactions offers a distinct advantage over the conventional approach of relying on spontaneous occurrence of desired interactions between animal subjects. Thus, conducting research on Bio-mimetic robots equipped with advanced sensory systems and capable of performing animal-like motion, has the potential to provide valuable insights to various fields of study, including biomedicine and animal ethology.

Rattus norvegicus, commonly known as the lab rat, has been used in bio-medical research for more than 150 years and remains the model of choice for studies of physiology, behavior, and complex human disease.[2] The laboratory rat has made invaluable contributions to cardiovascular medicine, neural regeneration, wound healing, diabetes, transplantation, behavioral studies and space motion sickness research. Their brains are larger than mice, and the animals are less timid and more intelligent. Although rats do not ’think’ like humans, some of their brain structure resembles the more primitive elements of human brains, and hence they can be used to model some human behaviors. Rats are also relatively easy to breed and maintain in a laboratory setting, and their short lifespan allows researchers to study the effects of treatments or interventions over multiple generations in a relatively short period of time. As such developing rat-like Bio-mimetic robots for use in behavioral interaction experiments is of great value.

As such, in order for the robotic rat to be able to participate effectively in behavioral interaction experiments, it needs to be capable of rat-like motion. And so, to produce such robot that, the following crucial problems need to be solved: (1) Effective motion planning

algorithms, designed to generate animal-like movements, require a foundation based on data derived from the animal's natural motion. As a result, data-driven methods form an essential component of the motion planning algorithm. By leveraging information obtained through observations of animal behavior, these methods can enable the creation of accurate and realistic models of animal motion, facilitating the development of more sophisticated robotic systems capable of mimicking animal behaviors.(2) Given the inherent variability in the expression of behaviors across individual lab rats, a comprehensive method capable of capturing the specific nuances and variations of each motion is necessary.(3) Given the dynamic and unpredictable nature of the lab rat's behavior, a real-time method that can effectively adapt to sudden changes in the target's position and motion is necessary. This underscores the need for developing sophisticated control and feedback systems capable of quickly detecting and responding to changes in the environment.

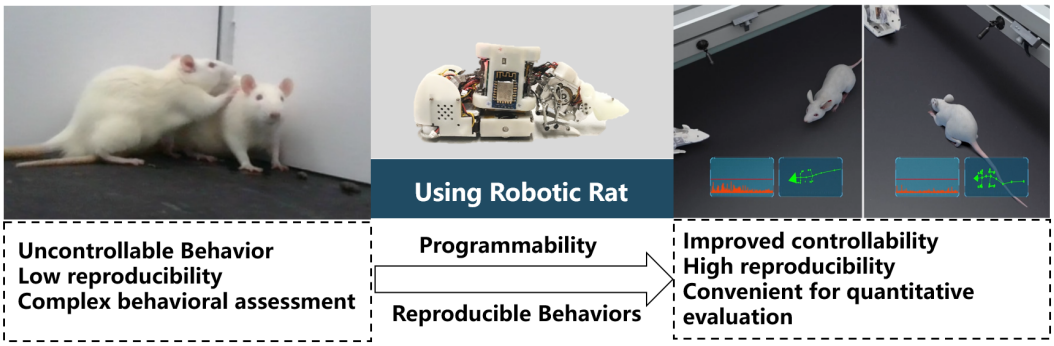


Figure 1.1 Comparsion between traditional experimentation methods and using robotic agents

1.2 Literature Review

In the following section a literature review and current trends in motion planning and behavioral interaction experiments for Bio-mimetic robots are introduced. We start with discussing currently used motion planning methods for Bio-mimetic robots, then discuss the specific case of motion planning for robotic rats, after that we introduce experiments on Robot-Animal behavioral interaction and finally we conclude with a brief discussion on the motion planning methods used in behavioral interaction experiments.

1.2.1 Motion Planning of Bio-mimetic Robots

The term motion planning is used in computational geometry, computer animation, robotics, and computer games. A motion planning algorithm takes a description of these tasks as input and produces the speed and turning commands sent to the agent's actuators. In the field of robotics, motion planning, also known as path planning or navigation, can be defined as the process of breaking down a desired movement task into discrete motions that satisfy movement constraints, safety requirements and possibly optimize some aspect of the robot's movement. Motion planning has emerged as a crucial and productive research area in robotics over the past few decades.

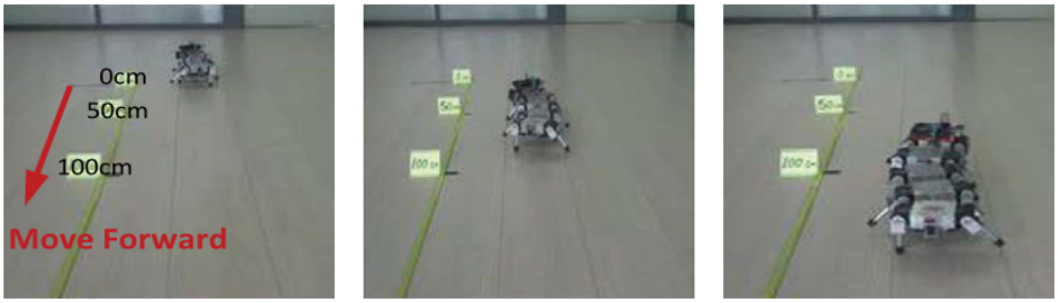


(a) Legged Model (b) Legged-Wheels Model

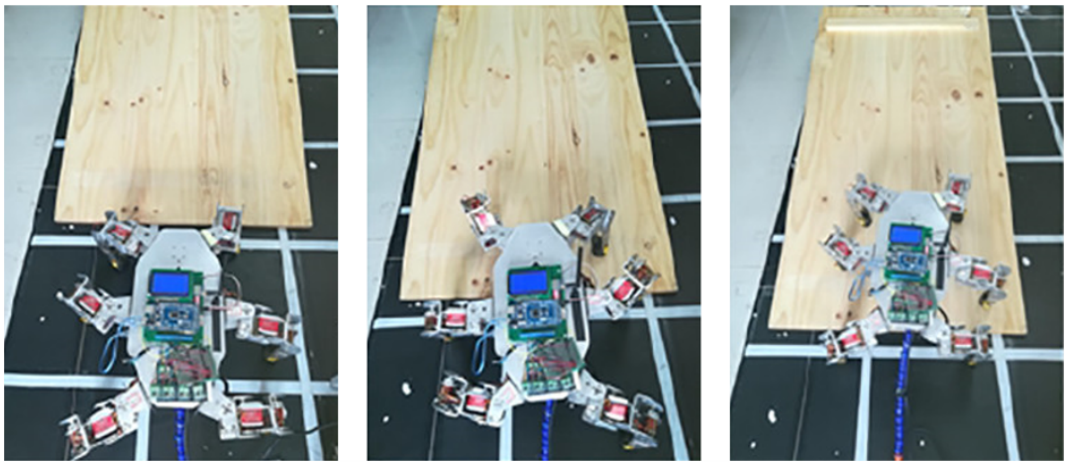
Figure 1.2 ETH Zürich's Quadruped Robot ANYmal

The approaches used for motion planning can differ greatly depending on the structure of the robot. In the literature, extensive work has been done on legged-robots, especially quadrupeds, due to their versatility and the high number of applications[3]. Researchers from ETH Zürich, mainly use Trajectory Optimization for the motion planning of quadruped robot ANYmal[4-6]. Trajectory Optimization is especially useful when there are complex kino-dynamic constraints that must be met for the robot's motion to be stable. In[4] Jan Carius et al, present a complete pipeline of optimization-based motion planning for legged robots. They demonstrated that the full-body dynamics of a quadruped can be efficiently optimized under unilateral constraints with Coulomb friction, possible stick-slip transitions, and automatically emerging contact mode schedule, allowing the robot to navigate stably with slipping motion. Cebe et al proposed a trajectory optimization framework that is

capable of generating dynamically stable base and footstep trajectories for multiple steps, they used a simplified momentum-based task space model for the robot dynamics, allowing computation times that are fast enough for online replanning. This fast planning capability also enables the quadruped to accommodate for drift and environmental changes[7]. Due to the generality of Trajectory Optimization, it can be applied in many motion planning scenarios.



(a) Multi-Legged Robot

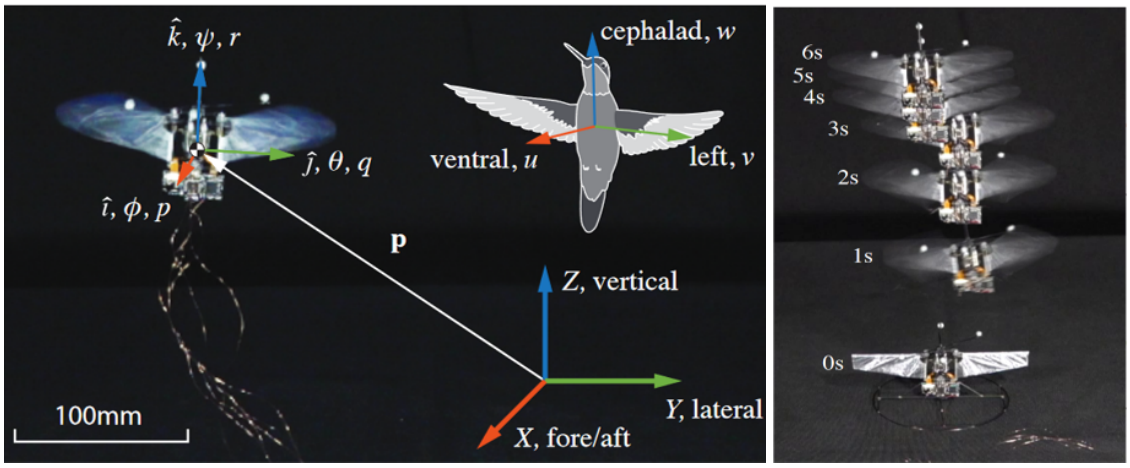


(b) Hexapod

Figure 1.3 Robots Controlled using Gait Planning

Another approach for motion planning in legged-robots is the use of gaits. In the case of robots, gait refers to the sequence of movements that are required to move the robot's legs in a coordinated manner to achieve forward or backward motion, turning, or other movements. Yongchen, et al, proposes an undulatory gait planning method for multi-legged robots with a passive-spine, they used a Finite State Machine to plan the undulatory gait sequence for straight line motion for the 2n-legged robot[8]. Using gait for motion planning has also been

used successfully on hexapod robots to generate stable motion and allow the robot to traverse rugged terrain, in[9] Wang et al. used a central pattern generator (CPG) that computes the landing points of the hexapod's legs for the gait planning, allowing the hexapod to be able to climb slopes of 16 deg. One way of performing gait planning is by taking direct inspiration from nature, by analyzing the target animal's gait patterns and using them as the basis for the gait motion planning algorithm. Li et al. captured the movements of a gecko using motion capture technology and used the gathered data to design a gait motion planner for a gecko-like robot[10].

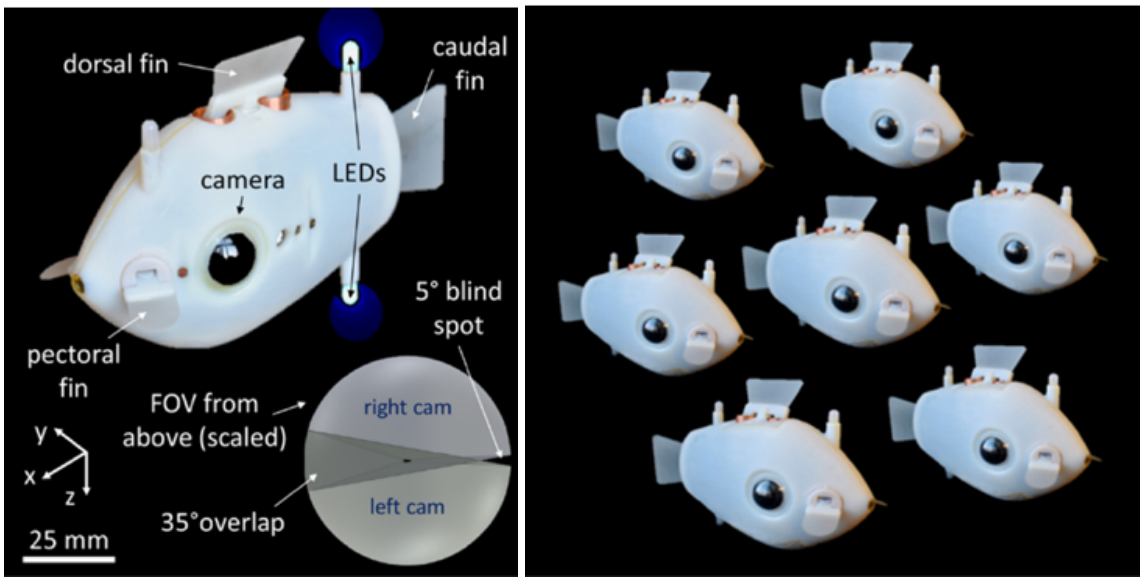


(a) Hummingbird-like Model (b) Hummingbird in Flight

Figure 1.4 Humming Bird

With the rise of neural networks and machine learning methods, reinforcement learning emerged as another method of generating motions. Reinforcement Learning (RL) is a type of machine learning that allows an agent to learn how to behave in an environment by performing actions and receiving rewards or penalties. In the context of motion planning for robots, reinforcement learning can help the robot learn how to move in its environment by trying different actions and receiving feedback in the form of rewards or penalties. Over time, the robot can learn an optimal policy for motion planning that allows it to navigate its environment effectively. Researchers from Purdue University have worked on a hummingbird-like flying robot and successfully managed to replicate the hummingbird's flying mechanism using hybrid approach for motion planning. The unsteady aerodynamics

and the highly nonlinear flight dynamics present challenging control problems for conventional and learning control algorithms such as Reinforcement Learning. Their proposed hybrid control policy combines model-based nonlinear control with model-free reinforcement learning. They used model-based nonlinear control for nominal flight control, as the dynamic model is relatively accurate for these conditions. A model-free reinforcement learning policy trained in simulation was optimized to 'destabilize' the system and maximize the performance during maneuvering[11-13].



(a) Individual Blueswarm Robot (b) Blueswarm School

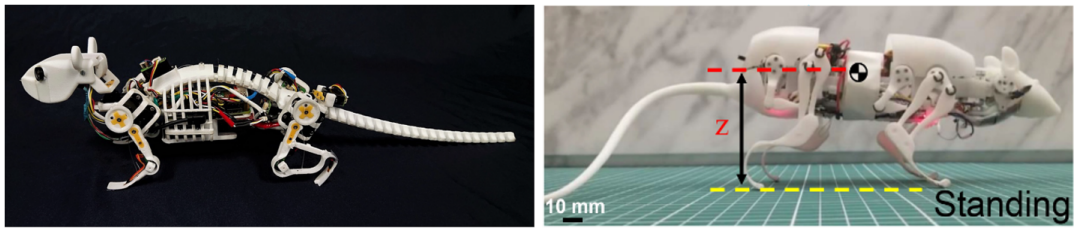
Figure 1.5 Harvard University's Blueswarm Robots

Swarm robotics is an exciting field that draws inspiration from nature, particularly the collective behavior of social insects such as ants and bees. It involves coordinating large groups of simple robots to work together in a decentralized and distributed manner, using swarm intelligence to accomplish complex tasks that would be impossible for individual robots to achieve[14]. One example of this is using swarm robots to explore unknown environments or perform search and rescue operations. Swarm Robots take inspiration from animals that behave as a group and have sophisticated coordination capabilities, such as fish, ants, bees and other. Researchers from Harvard University developed Blueswarm[15-18], a Swarm robotic platform to systematically study self-organized 3D coordination underwater. The robots were only programmed with two fundamental individual-level

capabilities for self-organization: 3D awareness of neighbors' distance and bearing and swift 3D motion response to neighbors. Blueswarm is able to achieve multiple 3D collective behaviors by exploiting biologically inspired coordination techniques that are inherently robust to imperfect knowledge and that enable the emergence of complex and dynamic global behaviors from seemingly simple interactions.

1.2.2 Motion Planning for Bio-mimetic Robotic Rats

Over the years, many researchers have been interested in building rat-inspired robots, the main goal for most researchers was to study and/or replicate certain aspects of the lab rat, such as their anatomy[19], their navigation strategies[20] or their behavioral interaction[21]. Just like with any robot, motion planning is necessary to for the robot to be able to move autonomously, in this section we introduce the commonly used methods in the literature for the motion planning rat-inspired robots.



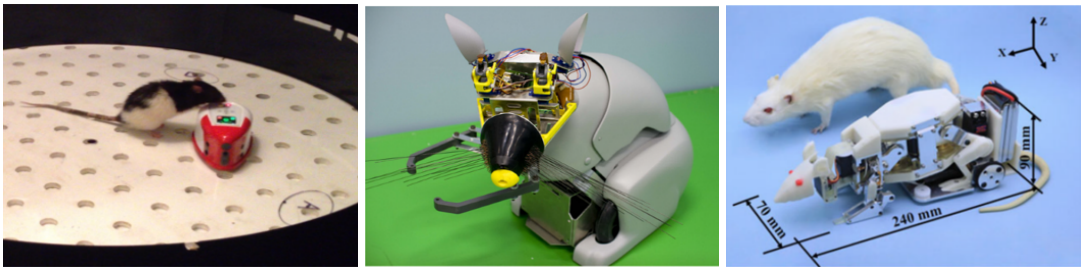
(a) The NeuroRobotic Rat (NeRmo) (b) Small-sized Quadruped Robotic rat (SQuRo)

Figure 1.6 Quadruped Robotic Rats

Depending on the design of the robotic rat, different approaches have been utilized to generate motions and control the robots. The NeuroRobotic Rat (NeRmo) as shown in Figure 1.6(a) developed by the Technical University of Munich is a low-cost, modular bio mimetic quadruped robot, mimicking the actuation and walking behavior of a common mouse[22-24]. As the name suggests, the authors took inspiration from Head Direction Cells (HDC) neural structure found in the limbic system of animals like rats. They demonstrated that this method can generate motion trajectories that allowed the robot to successfully navigate a simple maze-like structure while avoiding collision with walls. Another quadruped rat-like robot is SQuRo, developed by Beijing Institute of Technology as shown in Figure 1.6(b). The authors proposed a control framework for multimodal motion planning, tuned the control parameters through

optimization with consideration to the stability and actuation limits. Their planner uses a simple gait pattern modulator to compute joint trajectories and switch stably between different gaits[25].

Another type of design for rat-inspired robots is a wheeled platform model, sometimes with a multi-link arm attached on top. This is by far the most common design used in the literature, especially for behavioral interaction experiments[21]. Researchers from the Institute of Intelligent Systems and Robotics[20], proposed a biologically-inspired navigation system for the mobile rat-like robot named Psikharpax[26]. They implemented two motion planning strategies in parallel to allow the robot to navigate initially unknown environments. It was shown that the robot can learn to ignore useless strategies very fast (e.g. an exploration expert after the goal was found). Furthermore, the robot learned to associate states with optimal strategies even in more complex cases, when for example a local taxon strategy was combined with a global but coarse path planning strategy. By introducing a simple context switching mechanism, the robot could adapt quickly to changes in the environment.



(a) Intelligent Rat (iRat) (b) Psikharpax (c) Waseda Robotic Rat - 5 (WR-5)

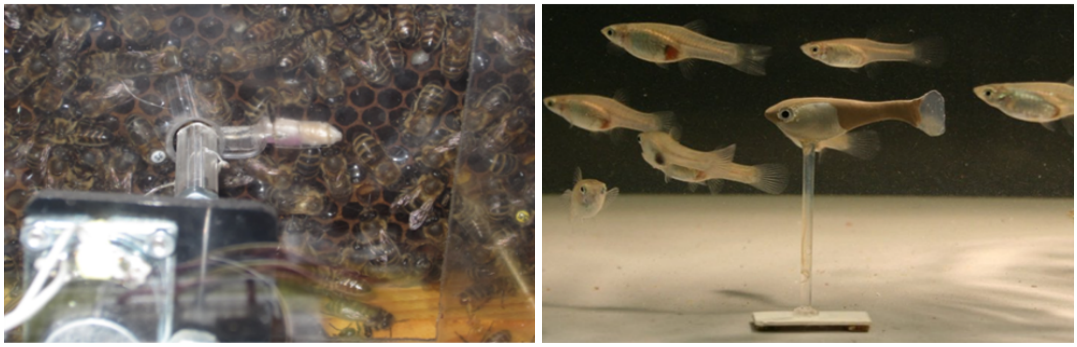
Figure 1.7 Wheeled Robotic Rats

The WR series of robotic rats developed by Beijing Institute of Technology[19,27] with high similarity in appearance and strong motor performance, can achieve typical behaviors such as upright standing, grooming, and sniffing similar to those of lab rats. Among them, the latest generation known as SMuRo, integrates an improved WIFI-based graphical transmission with a digital signal camera improved binocular vision system[28]. The rat-inspired design of the robot enables it to accurately replicate rat movements with a high degree of fidelity. By carefully analyzing the motion of rats, the SMuRo research team has been able to develop algorithms and control systems that allow the robot to emulate the

natural movements of its biological counterpart. Through this data-driven approach, researchers have been able to develop a robot that can replicate the complex and nuanced movements of rats, providing a powerful tool for investigating animal behavior and advancing our understanding of biological systems

1.2.3 Robot-Animal Behavioral Interaction

In recent years, there has been a burgeoning interest in the field of animal-robot interaction, with researchers exploring new ways to design robots that can interact with animals in a constructive and beneficial manner. Through this work, researchers aim to improve animal welfare, deepen our understanding of animal behavior, and develop new tools and techniques for studying and interacting with animals. In order to further research the underlying mechanism of animal behaviors, biomimetic robots are usually placed in a group of animals to explore animal behavior patterns or validate scientific hypotheses in the interaction process[29].



(a) Free University of Berlin's RoboBee Interacting with Bees

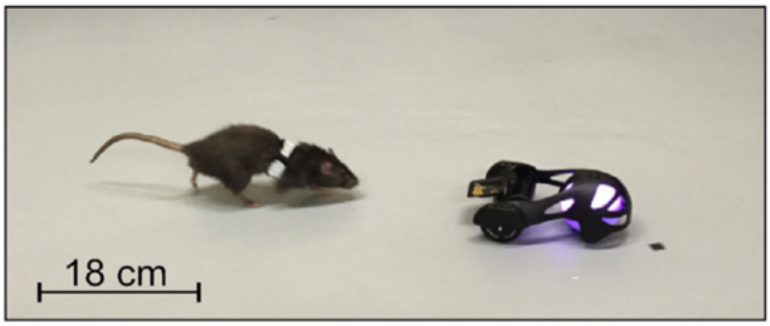
(b) 3D Model of Guppy attached to wheeled platform interacting with Fish

Figure 1.8 Robots Interacting with Animals

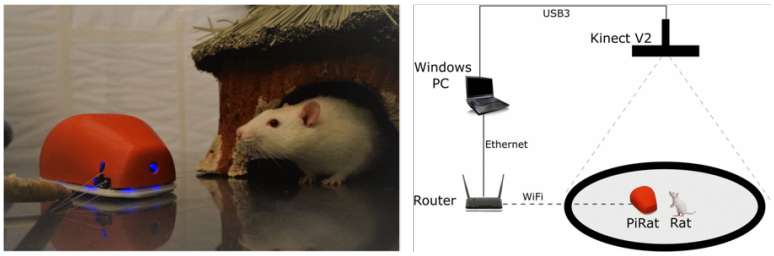
Researchers from the Free University of Berlin conducted robot-animal behavioral experiments on Honeybees using RoboBee, a life-sized honeybee replica[30-31]. They derived a motion model for bees' waggle dance from a large database of natural dance trajectories and have programmed the Robobee such that it can perform these motions[32]. The waggle dance indicates directional information about location of feeders for the bees, they observed that RoboBee's dances elicited similar behavioral responses as observed in natural dances. Not only do bees follow robotic dances, they follow even for longer periods

than with live dancers. The trajectory dance-followers describe in robotic dances is very similar to the one performed in natural dances and may be sustained over dozens of waggle runs. They tracked the flights of some of these bees and all flight paths indicate that directional information was indeed transmitted by the robot[33].

In another experiment, researchers from Leibniz-Institute of Freshwater Ecology and Inland Fisheries used a robotic fish to investigate differences in social responsiveness in the guppy[34]. They used a three-dimensional (3D)-printed guppy-like replica that is attached to a magnetic base aligned with a wheeled robot that is driving below the actual test tank to interact with a group of guppy fish in a test area. They found that guppies followed larger Robofish leaders closer than smaller ones and this pattern was independent of the followers' own body size as well as risk-taking behavior[35].



(a) Sphero Interacting with Rat



(b) PiRat Interacting with Rat

Figure 1.9 Robots Interacting with Rats

There have been behavioral experiments on robot-rat interaction as well. The University of Arizona used a spherical robot, Sphero, to help rats learn and perform spatial memory tasks, primarily using a top camera for location capture and a reflective band wrapped around the middle of the rat's body to provide visible recognition feedback[36] as shown in Figure

1.9(a). The results show that Sphero is able to control the rat's movement through its own motion and generate specific trajectories and speeds, which in turn assist it in recording paths in complex environments.

The University of Queensland, Australia, has proposed an automated framework for observing the robot-rat interaction using a top-fixed camera based on the mouse-like robot Pirat[37] (e.g.,1.9(b). The framework consists of PiRat, a robot behavior model, and a position tracking system for both the robot and the rat. The PiRat was used to track the experimental rat in a fixed area during operation and record the position and distance of both. The tracking experiments show that the rats show different trajectories according to the different behaviors of the robot.

1.2.4 Existing Problems

In the reviewed literature, researchers employed a vast variety of methods to develop motion planning algorithms for bio-mimetic robots. Classical and robust methods such as Trajectory Optimization are used extensively for bio-mimetic robots and Machine Learning algorithms are also a popular choice. However, in order to use motion planning for conducting autonomous behavioral interaction experiments, some existing challenges have to be addressed:

1. One of the primary challenges in motion planning for bio-mimetic robots is the lack of consistency in motion data space representation. In behavioral interaction studies involving robots and rats, researchers have typically not employed animal motion data to control the robots. This is because establishing a direct correspondence between the motion space of laboratory rats and that of robots is challenging, as the robot's motion space typically consists of joint angles and velocities, while the rat's motion space is more complex. Consequently, there is an inconsistency in motion space representation that makes it difficult to directly utilize laboratory rat's motion data to generate motions for the robotic rat. Therefore, addressing this issue is crucial to improving the performance of bio-mimetic robots in interaction studies with animals.

2. Another significant issue in motion planning for bio-mimetic robots is the limited level of bio-mimetic similarity in motion expression. This is primarily due to the use

of simplified and rigid motion patterns in current planning approaches. For instance, robotic rats such as SQuRo rely on gait patterns for motion planning. However, these motion patterns lack the bio-mimetic characteristics of real rats and may fail to capture the subtle variations and nuances in animal movements. Consequently, robotic movements produced using these patterns may appear unnatural and unrecognizable to rats. Addressing this issue is critical to enhancing the bio-mimetic similarity of robotic motions, which could improve their performance in interaction studies with animals.

3. Another significant challenge in motion planning for behavioral interaction is the lack of robust tracking algorithms. For robots to interact effectively with animals, they must maintain continuous visual contact with the animals and track their movements in real-time. However, currently most tracking methods for rats rely on external cameras, such as the one used in PiRat, which limit the range of tracking and can only extract simple behavioral information. Therefore, there is a need to develop more robust and flexible tracking algorithms that can improve the accuracy and reliability of robot-animal interaction studies.

1.3 Structure of Thesis

By simulating and analyzing the interaction between robotic rats and lab rats, researchers can gain valuable insights into the complex social behaviors of rats, including social hierarchy, territoriality, and mating behavior. This can help in understanding the underlying mechanisms of various diseases that affect social behavior, such as autism and schizophrenia. Additionally, the robotic rats can be used to test the effects of new drugs on social behavior, which can lead to the development of new treatments for these disorders. Therefore, such research can have a significant impact on disease mechanism exploration and drug development testing. For a rat-like robot to be able of conducting behavioral interaction experiments with real rats, it has to have the ability to perform rat-like motions and be able to track the position of the laboratory rat in real-time. Hence, the purpose of this research is to produce a data-driven motion planning method for robotic rat to use as a basis for behavioral interaction experiments. This research aims to pursue the following goals:

(1) Produce a dataset of rat-like motions based on real laboratory rats motions and behaviors. The dataset has to be expressed in joint angles such that it can be used to design control sequences for the robotic rat, hence a mapping method is to be developed to convert the motion dataset into a form viable for use as joint commands.

(2) Using Probabilistic Motion Primitive, produce a probabilistic model of different motions that captures the variance in each behavior. The model is used as the basis for generating trajectories used in behavioral interaction experiments.

(3) Using Image-Based Visual Servoing, develop a real-time tracking strategy for laboratory rats, allowing the robotic rat to continuously follow and monitor the laboratory rat during behavioral experiments.

(4) Design a motion planning policy for conducting robot-rat behavioral interaction experiments using data-driven motion generation and real-time tracking.

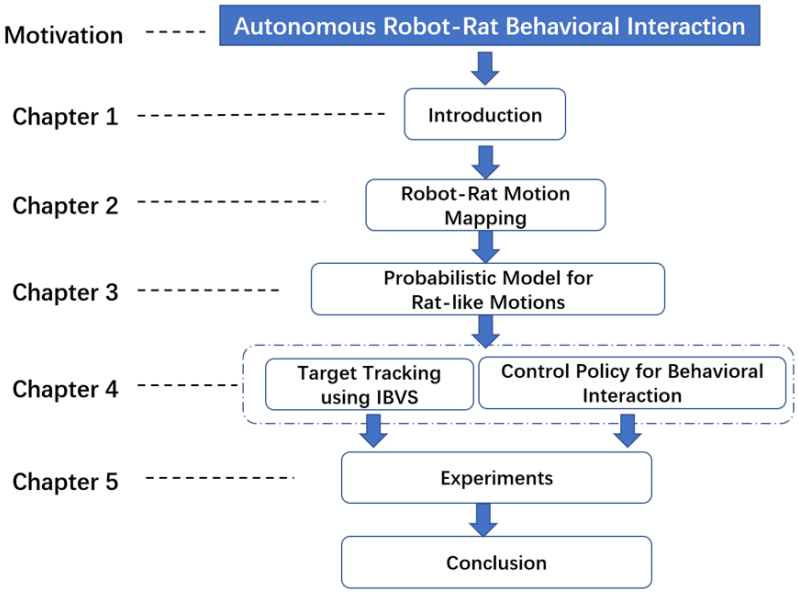


Figure 1.10 Thesis Structure

In Chapter 1, the background and research significance of this work were discussed, and the representative research results in recent years on bio-mimetic robot motion planning, robotic rat motion planning, and robot-animal behavioral interaction experiments were summarized. Combining the research progress and technical means of existing methods, the main problems of current research in the robot-rat interaction scenario are analyzed, and

then the main research contents of this paper are proposed.

In Chapter 2, a mapping algorithm for converting motion capture data of laboratory rat into joint angles of the robotic rat is introduced; the motion data, which is classified into different behaviors, that serve as the basis of this work is first discussed. Then a mapping algorithm using non-linear optimization is introduced in order to make the motion capture data suitable for use on the robotic rat. Finally, the mapping algorithm is used to process the motion data and produce a motion dataset expressed in robotic rat joint angles.

In Chapter 3, a motion primitive library for the robotic rat is produced for use in behavioral interaction experiments. The probabilistic motion primitives (ProMPs) framework, which is used for generating a probabilistic model from motion demonstrations, is first introduced. Then, a number of crucial properties of ProMPs that are essential for generating trajectories for behavioral interaction are derived. Finally, using the motion dataset produced in Chapter 2, the ProMPs framework is utilized to generate a library of motion primitives for the robotic rat, from which trajectories for behavioral interaction are generated.

In Chapter 4, a real-time rat-tracking algorithm for the robotic rat is introduced. First, the robotic rat's lab detection and depth estimation capabilities are introduced. Then, in order to maintain vision of the laboratory rat during behavioral interaction experiments, a rat-tracking algorithm based on Image Based Visual Servoing is developed; the derivation of the algorithm and the control law is first presented, then coordination between the robotic rat's spine and mobile is introduced. Finally, the control policy for conducting a mock behavioral interaction experiment is presented.

In Chapter 5, experiments are conducted on the methods presented in the 2nd, 3rd and 4th Chapters. The results of the Rat-Robot Motion Mapping algorithm is assessed and the resulting motion database is introduced. Then, the motion trajectories generated by Probabilistic Motion Primitive framework are presented and analyzed. Finally, the rat-tracking algorithm is verified in tracking experiments with the laboratory rat.

In the Conclusion, the research results, main innovations and prospects for future work are summarized.

Chapter 2 Rat-to-Robot Motion Mapping

2.1 Overview

One of the key challenges in designing robots that can interact with animals is understanding the animal's behavior and communication. Mimicking animal motions for behavioral interaction is a challenging task for robots. Animals have complex and dynamic movements that are often difficult to replicate, even with the most advanced robotics technology. The challenge is not only to imitate the movements of an animal accurately but also to do so in a way that is safe and non-threatening to the animal. To accurately mimic animal motions, researchers often use motion capture technology, which involves attaching sensors to an animal and recording its movements. These recordings can then be used to develop algorithms that replicate the animal's movements in a robotic system.

During the past few decades, the rapid development of Motion Capture Technology (MoCap) has led to a new surge in data-driven motion planning for robotics. Motion capture systems have been widely used in the entertainment industry to capture human movements for animation and special effects. However, in recent years, these systems have also found application in robotics, particularly in the field of robotic motion planning. Motion capture systems can provide highly accurate and detailed data about human and animal movements, which can be used to develop more realistic and effective robot movements. Using Motion Capture (MoCap) Technology, the movements of the target subject in space can be digitally recorded and stored in a motion capture dataset. The resulting dataset can then be used in training algorithms to study the nature of the movements and learn to produce similar motions[38]. Over the past decade data-driven methods have been employed to produce realistic and human-like motions for use in Sports Education, Video Games, Simulation and VR/AR technologies[39-40]. In the field of robotics, MoCap technology have been mainly used in Learning from Demonstration (LfD) and Imitation methods[41]

One of the main benefits of using motion capture systems in robot motion planning is the ability to accurately capture and replicate a target's movements. By analyzing data from motion capture systems, researchers can identify key features of the targeted movements

and use this information to develop more natural and efficient robot movements. This can be particularly useful in applications where robots need to interact with humans, such as in healthcare or manufacturing settings, or in mimicking animal movements, such as for behavioral interaction.

However, mapping motion capture data to robotic systems is a complex process that involves several challenges. Motion capture data captured by MoCap sensors needs to be processed and translated into robotic commands that can be used to control the robot's motion. This is challenging due to the fact that there is often a non-linear relationship between the motion capture data and the robotic commands required to reproduce that movement. This is because motion capture data does not directly translate into the robotic control signals that drive the robot's actuators. Thus, complex algorithms are required to convert the captured data into the required robotic commands.

In this chapter, the motion capture data that is used as the basis for the development of the motion planning framework is discussed. First, the motion capture system and setup of the environment used in creating the laboratory rat's motion dataset is introduced. Secondly, the classification of the motion capture dataset using Principal Component Analysis into different motion sets corresponding to different behaviors is presented. Finally, a Rat-to-Robot motion mapping algorithm using non-linear optimization is introduced to translate the classified motion capture dataset of the laboratory rat into the robotic rat SMuRo's joint angles.

2.2 Motion Capture Dataset

In this section, the motion capture dataset, which serves as the basis for this work is introduced. Data-driven bio-mimetic motion planning requires using data collected from laboratory rats to generate the motion trajectories. Thus, motion capture technology is employed to produce a dataset of laboratory rat motions to be used for analyzing the rat's behaviors and as a reference for any motion planning and behavioral interaction endeavors.

2.2.1 Motion Capture

In order to capture the motion of a laboratory rat, a specialized system of cameras and sensors is typically employed, which can precisely track the rat's movements in three

dimensions. For this purpose, OptiTrack motion capture camera sensors were utilized. This setup involves fitting the rat with small reflective markers on various points of its body, which can be tracked by the cameras as the rat moves through a controlled environment. The position of the reflective markers were chosen meticulously based on KMJ[42] as shown in Figure 2.1.

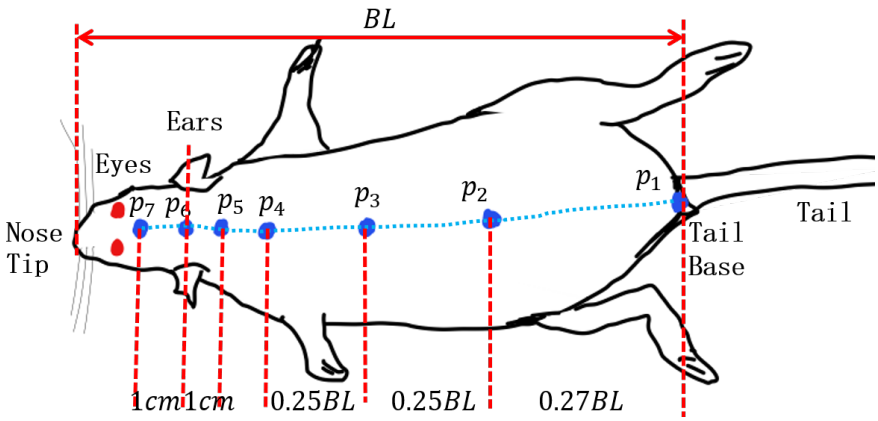


Figure 2.1 Locations of the reflective markers used for motion capture

To ensure accurate tracking, the cameras are strategically placed around the environment to capture the rat's movements from multiple angles. This allows for a more precise reconstruction of the rat's motions in three-dimensional space. The data collected from the cameras is then processed using specialized software, which creates detailed recording of the rat's movements. This valuable information can be utilized for various research applications in fields such as neuroscience, biomechanics, and behavioral analysis. The motion capture setup is described in detail in Figure 2.2

In this work, the laboratory rat's motion capture data is utilized as the basis of a motion generation algorithm for conducting behavioral interaction experiments with a live laboratory-rat in a controlled environment. The motion capture can be used to teach the robotic rat rat-like motions, which will allow the robotic rat to interact more effectively with the laboratory rat.

2.2.2 Behavior Classification

In order to characterize the behavior of the experimental rats more specifically, the state and behavior of the experimental rats based on the positions of the 3D key points in the

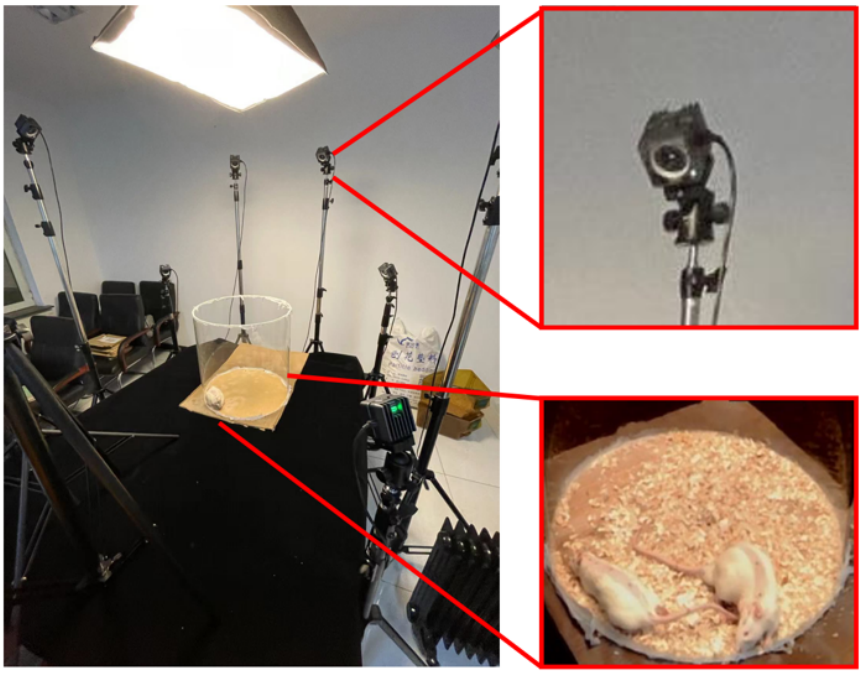


Figure 2.2 Setup for laboratory Rat Motion Capture

world coordinate system obtained from the motion capture. The 3D positions of the seven key points on the line from the head to the spine to the tail are used to describe the state of the experimental rat at the current moment, and the behaviors of the experimental rat during the time period are characterized by the sequence of states in consecutive times. The specific positions and connections of the seven points are based on the definition of Key Movement Joints as described in[42].

As the behavior description process shown in, the key point detection results of successive left and right maps $f_1, f_2 \cdots f_n$ were obtained by 3D reconstruction and positional transformation to obtain the state sequence $S_1, S_2 \cdots S_n$, and the final coordinates were aligned to describe the behavior B_m. However, the data form of the poses composed by the key point linkage is not convenient for quantitative description of the behavior, so exploratory data analysis is needed to extract the features of the poses data for subsequent real-time behavior recognition

To construct the behavior category map of experimental rats, a behavioral mapping method, CAPTURE, was used, which extracts feature quantities based on original data, including vectors, angles, and vector lengths. The principal components of pose features and

dynamic features are extracted based on PCA and wavelet transform, and then the t-SNE algorithm is used for dimensionality reduction and embedding high-dimensional feature data into a two-dimensional graph. Watershed clustering and manual classification are used to group data with similar behavioral features and construct the behavior classification graph.

The process of constructing the behavior category map involves six steps. First, a large amount of 3D key point data containing the rich behaviors of experimental rats is acquired using the motion capture system. Second, the coordinates of the original 3D key points are aligned by translation and rotation. Third, the feature set is defined based on angles and vector coordinates. Fourth, PCA and wavelet transform are used to extract principal components and dynamic features to reduce the dimensionality of the data. Fifth, t-SNE algorithm is used to embed high-dimensional feature data into two-dimensional data. Sixth, watershed clustering and manual classification are used to group data with similar behavioral features and construct the behavior classification graph as shown in Figure 2.3[43].

Four behaviors are considered in this work, they are MO, PIN, POU, SNC, their descriptions are as follows:

1. MO: represents moving, this behavioral pattern encompasses the motions that the laboratory rat exhibits when moving. It is characterized by a lowered head with occasional left-right movement, which is assumed to be sniffing the environment.

2. PIN: represents pinning, this is an interaction behavioral pattern between two rats where one of them mounts the other. This behavior is characterized by a motion of standing up of one rat and laying down on the back of the other rat.

3. POU: represents pouncing, this is an interaction behavioral pattern between two rats that is similar to PIN. This behavior is characterized by one rat raising its torso and lowering it slowly on top of the other rat and is less aggressive than PIN.

4. SNC: represents social nose contact, this is an interaction behavioral pattern where the two rats make contact using their nose, it is characterized with a left-right movement of the head possibly for sniffing the other rat.

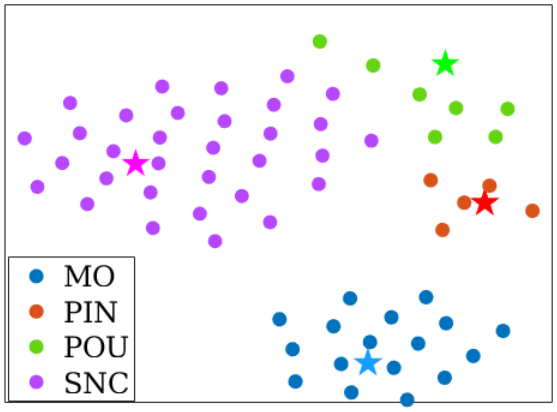


Figure 2.3 Behavior Classification based on Clustering

2.3 Rat-to-Robot Mapping Algorithm

2.3.1 Problem Formulation

Before it is possible to use the classified motion capture dataset in designing a motion controller for the robotic rat, a mapping process is necessary. As shown in Figure 2.4, the motion capture dataset of the laboratory rat is based on the 7 marker points placed on the outer spine and head of the rat. Thus, it cannot be used directly to guide a motion policy for the robotic rat platform, which is controlled through it's joints. In this section a rat-to-robot mapping algorithm using non-linear optimization is introduced. It is employed to find the suitable control sequences of the robotic rat that correspond to the classified behaviors, such that the robotic rat can perform those motions in a rat-like manner and interact with the laboratory rat.

The mapping process is crucial since the motion capture data obtained from the laboratory rat cannot be directly translated into the corresponding joint angles of the robotic rat. Therefore, a conversion from the body points of the laboratory rat to the joint angles of the robotic rat is required to utilize the data for motion planning purposes.

A direct geometric relationship between the laboratory rat's body and the robot's joints is challenging to establish due to several factors. The flexibility and shape-shifting nature of the laboratory rat's spine make it difficult to establish a one-to-one geometric correspondence between each pair of body points. Another factor that makes it challenging

to establish a direct geometric relationship is the variation in size and morphology of different laboratory rats included in the dataset. This variability makes it impossible to use a static conversion method to produce reliable results. In order to establish a dependable correlation between the two values, it is essential to utilize an adaptive approach that can determine a shared representation facilitating comparability. This necessitates the use of a measurable metric capable of evaluating the similarity between the two values. In order to solve this problem the following mapping process is proposed:

1. Starting with the joint angles of the robotic rat denoted as **q**, forward kinematics is performed to obtain the coordinates of each link in the world frame.
2. Subsequently, a set of points that are attached to the robot's links are chosen to effectively represent the robot's body.
3. Finally, the body points are utilized to interpolate a curve, which accurately delineates the contour of the robotic rat.

By following the same interpolation approach on the laboratory rat's body points, the contour of the laboratory rat can also be obtained. With both outlines available, a valid comparison between the postures of the robotic rat and the laboratory rat can be made.

The challenge of correlating the laboratory rat's body and the robotic rat's joints can be expressed as the task of identifying the joint angles denoted as **q**, which enable the outline of the robotic rat to closely match the contour of the laboratory rat during the corresponding movement.

2.3.2 Preprocessing

After behavior classification is performed, the dataset is divided into the 6 different types mentioned above. In this subsection, preprocessing is performed on the classified data to prepare it for Rat-to-Robot motion mapping. First the 7 body points of the dataset was labeled manually, such that the point at the base of the rat (slightly above the tail) is point number 1 and the final point in the head of the rat is point number 7. A filter is then applied to remove erroneous readings from the MoCap sensors. Finally, interpolation of missing frames and

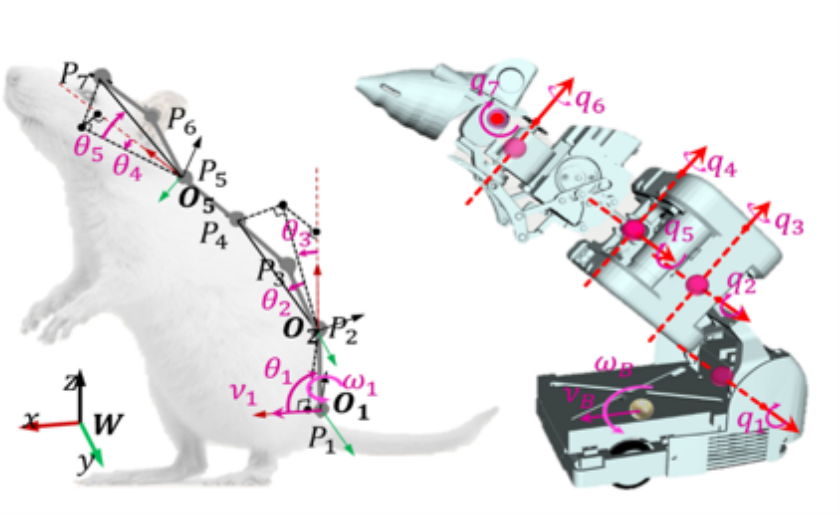


Figure 2.4 Comparison between Laboratory Rat's body & head and SMuRo's joints.

smoothing is employed to increase the quality of the dataset. The laboratory rat's body points in each frame can be represented as a 3x7 matrix in Cartesian coordinates, where each column represents one of the seven points in 3D space as shown in Eq. (2.1).

$$f_i = \begin{bmatrix} P_1 & P_2 & \dots & P_7 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_7 \\ y_1 & y_2 & \dots & y_7 \\ z_1 & z_2 & \dots & z_7 \end{bmatrix} \tag{2.1}$$

Where f_i represents a single frame in the motion capture dataset. However, these points are expressed relative to the world frame of the motion capture system. This is undesirable for rat-to-robot mapping since we are only concerned with the body's shape and not its position relative to the world frame. Hence, alignment of each frame's data would be beneficial for the mapping process as it reduces the complexity of the problem and allows for easier visualization of the relative motion of the body between frames.

The 1st point of the body, which is located above the rat's tail, does not move significantly relative to the rat's body, this makes it a suitable choice as the base point for the alignment. First, the points in each frame are all translated to the base point's coordinate system, such that base point is at the origin. Then a rotation matrix is defined using the vector between the base-point and the 3rd-point of the body. Technically, any other point paired with the base-

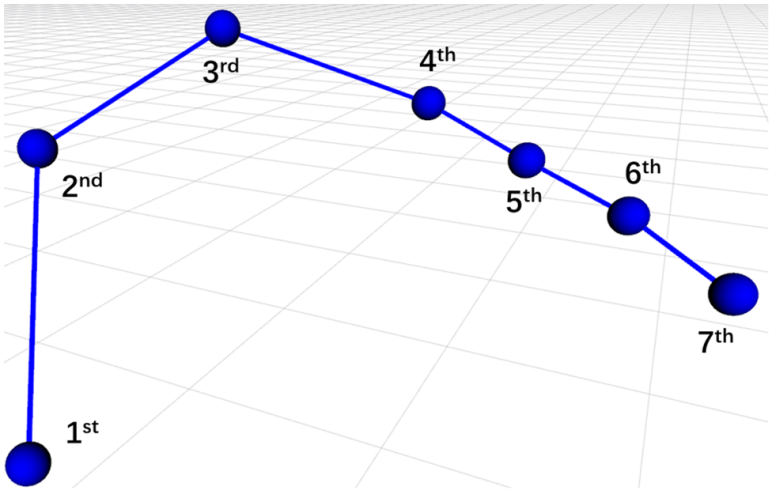


Figure 2.5 A Single Frame from the Motion Capture Dataset

point can be used to define a rotation matrix of the body, as long as the same pair of points is used for all frames. The 3rd point is chosen since its relative position to the base-point does not change dramatically between frames. The rotation matrix is then normalized such that it satisfies $R_i = R_i^{-1} = R_i^T$. Since within the dataset the rat moves on a flat surface, it only exhibits yaw rotations. Therefore, only the rotation matrix for revolutions around the z-axis is needed. Thus, the rotation matrix is of the general form Eq. (2.2). Finally this rotation is applied to all points in the frame completing the alignment process.

$$R_i = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$f_{aligned} = R_i \begin{bmatrix} P_1 - P_1 & P_2 - P_1 & \dots & P_7 - P_1 \end{bmatrix} \tag{2.3}$$

The entire process can described mathematically with Eq. (2.3) and shown in Figure 2.6. This process is applied to the entire classified dataset of the rat's motions, and produces a re-aligned dataset ready for use in rat-to-robot mapping.

2.3.3 Mapping Process

As described above, from the robot's side, the process start with the joint angles of each joint **q**. Given the D-H parameters[44] of the robot's links as shown in Table 2.1[28], the

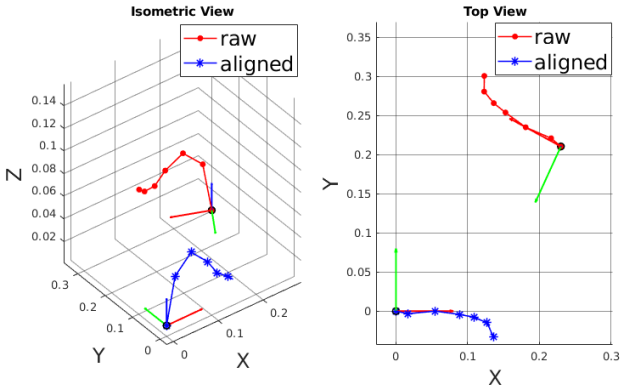


Figure 2.6 Alignment of Motion Data

transformation matrix of each link can be derived as per Eq. (2.4). Forward Kinematics is then computed to obtain the position and orientation of each link with respect to world coordinates. Computation of the transform of the nth link is described in Eq.(2.5).

Table 2.1 D-H Parameters of SMuRo

Link Type	i	a_i	α_i	d_i	θ_i	Joint Description	Inital θ_i
Behavior Link	1	a_1	0	0	$\pi/2$	Base Link	1.57
Behavior Link	2	a_2	0	0	θ_1	Pelvis Pitch	-1.24
Behavior Link	3	0	$-\pi/2$	0	θ_2	Waist Pitch(Lower)	0.33
Behavior Link	4	a_3	0	0	θ_3	Waist Yaw(Lower)	0
Behavior Link	5	0	$\pi/2$	0	θ_4	Waist Pitch(Upper)	0
Behavior Link	6	a_4	$-\pi/2$	0	θ_5	Waist Yaw(Upper)	0
Behavior Link	7	a_5	$\pi/2$	0	θ_6	Head Pitch	0
Behavior Link	J_s	a_6	0	0	θ_7	Head Yaw	0.25
Movement Link(L)	J_l	a_w	0	a_z	θ_{wl}	Left Wheel	0
Movement Link(R)	J_r	a_w	0	a_z	θ_{wr}	Right Wheel	0

$$T_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\alpha_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\alpha_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$T_n^0 = \prod_{i=1}^{n} T_i^{i-1} \tag{2.5}$$

Having computed the robot's position in world coordinates, next the rat's body has to

Table 2.2 Robotic Rat's body point parameters

Body point	Joint Frame p_f	Displacement P^{p_f} (mm)
P_1^{bot}	Base Link	$[-2.3, 0.0, 0.0]^T$
P_2^{bot}	Waist Rod	$[-0.5, 2.5, 0.0]^T$
P_3^{bot}	Waist	$[0.0, 0.0, 2.5]^T$
P_4^{bot}	Chest Rod	$[0.0, 0.0, 2.5]^T$
P_5^{bot}	Neck	$[1.8, 1.0, 0.0]^T$
P_6^{bot}	Neck	$[-2.2, 2.2, 0.0]^T$
P_7^{bot}	Head	$[3.3, 1.5, 0.0]^T$

be defined. In accordance to how seven points were chosen to represent the laboratory rat's body for motion capture, seven points are chosen using the same principles of Key Movement Joints that were used in choosing the marker points of the laboratory rat for the motion capture. Seven points along the body of the robotic rat were chosen to create the robotic rat's body as shown in Figure 2.7.

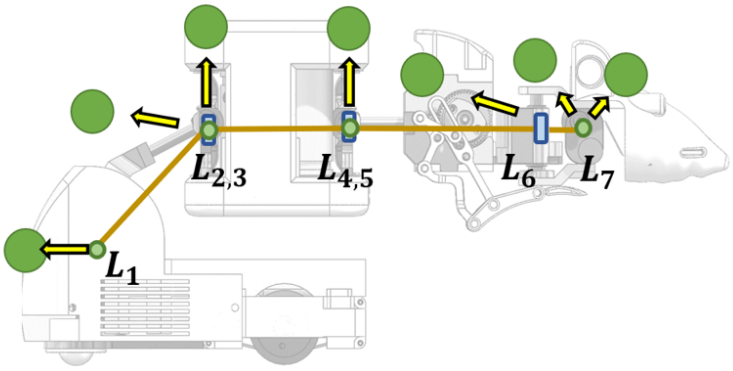


Figure 2.7 SMuRo's body Points

Despite both the laboratory rat and robotic rat possessing seven body points now, a direct comparison between them is not feasible owing to the laboratory rat's body flexibility, enabling its seven body points to move relatively to one another during stretching and expanding. In contrast, the robotic rat is a rigid entity, which cannot exhibit such flexibility. Hence, in order to make a valid comparison between the two, interpolation is employed to produce an contour that spans the entire body. Hermite interpolation is another commonly used curve interpolation method that uses a series of control points and their associated tangents to define the curve. The advantage of Hermite interpolation is that

it can generate smooth and natural curves with precise control over the curve's shape and derivatives. It can also handle sharp corners and cusps in the data set. After evaluating multiple interpolation techniques, Hermite interpolation outperforms others in accurately approximating the contour of the rat. Therefore, it is employed to generate a curve from the body points. Hermite interpolation has the advantage of high accuracy and smoothness for applications that require high accuracy fitting. The equation for calculating the curve for a given m data points $P_1, P_2, \cdots, P_m$ using Hermite interpolation is described in Eq. (2.6)

$$\begin{aligned} H(s) = &\left(2t^3(s) - 3t^2(s) + 1\right) P_{[s\cdot m]} + \left(t^3(s) - 2t^2(s) + t(s)\right) P'_{[s\cdot m]} + \\ &\left(-2t^3(s) + 3t^2(s)\right) P_{[s\cdot m+1]} + \left(t^3(s) - t^2(s)\right) P'_{[s\cdot m+1]} \\ &\text{s.t. } t(s) = t \cdot m - [t \cdot m] \\ &s \in [0, 1] \end{aligned} \tag{2.6}$$

Hermite interpolation is applied to both the body points of the laboratory rat and the robotic rat, generating two comparable 3D curves.

In the equation, $[\cdot]$ represents the rounding operation, P is the set of interpolation nodes, P' is the first-order derivative constraint of the nodes, and H is the generated Hermite curve. The key point position information on the back of the experimental mouse has been provided. Based on the Hermite interpolation, the body contour can be calculated. To apply convex optimization algorithms, it is necessary to calculate the corresponding body contour curve H_s of the robot mouse. According to equation Eq. (2.6), this paper defines 7 nodes on the back of SMuRo based on its mechanical structure, and the definition method is the position offset of each key point relative to the key link. The relationship between the positions of these nodes and the joint positions is:

$$\mathbf{P_s} = R^b_{p_f}(q) P^{p_f} + P^b_{p_f}(q) \tag{2.7}$$

Here, p_f represents the reference coordinates of the defined key points, such as the first key point's position relative to the base link, then its p_f is the Base Link. $R^b_{p_f}(q)$ is the rotation matrix of the reference coordinates relative to the Base Link under specific robot configuration parameters q, P^{p_f} is the offset of the defined key points relative to the

reference coordinates, and $P_{p_f}^{b}(q)$ is the position of the reference coordinates origin relative to the Base Link under specific robot configuration parameters q. Figure 2.7 shows the positions of these points and their relationship to the reference coordinates. Table 2.2 provides the defined variables of the key points of the robotic rat.

The entire conversion process is succinctly presented in Figure 2.8. Now that there are two comparable contours, the mapping problem becomes that of finding the suitable joint angle values **q** such that the curve of the robot's contour matches that of the laboratory rat. In the next section, non-linear optimization is utilized to determine the joint angles required to produce a matching curve.

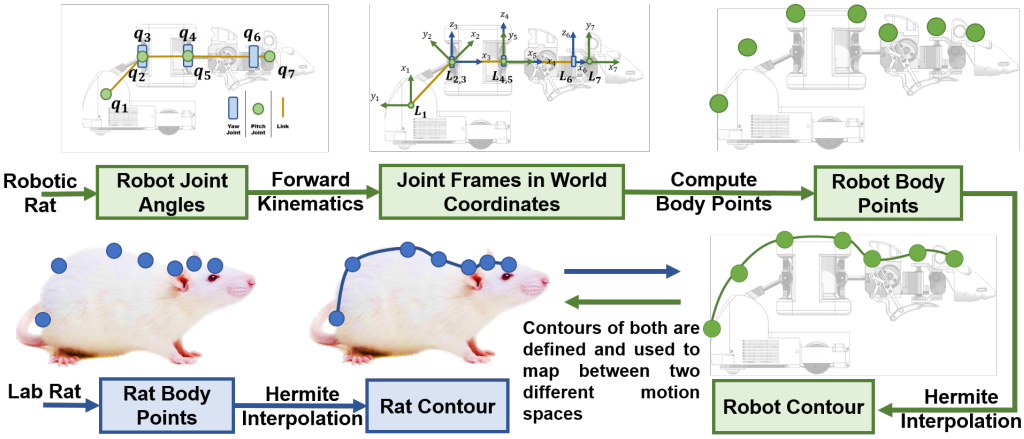


Figure 2.8 Rat contour to Robot Joints Motion Mapping Process

2.3.4 Optimization Problem

Optimization is a powerful tool for curve fitting as it enables us to identify the best possible fit of a mathematical function to a given set of data points. Curve fitting aims to accurately represent the relationship between input and output variables using a mathematical function or set of parameters. In the context of rat-to-robot mapping, curve fitting can be employed to optimize joint angles and fit them to a given curve. This technique aids in achieving the goal of accurately representing the relationship between joint angles and the curvature of the body. In this subsection, the mathematical formulation of a non-linear optimization problem that uses techniques from curve fitting to obtain the optimal joint angles is presented. The above stated problem of mapping is formulated into a non-linear optimization problem where we are minimizing the error between the two contours.

The choice of optimization variables is important because it affects the complexity of the optimization problem and the speed and accuracy of the optimization algorithm. If too many optimization variables are used, the optimization problem may become too complex and difficult to solve. If too few optimization variables are used, the fit may not accurately capture the underlying relationship between the input and output variables. In our case, the robotic rat's body consists of 7 joints, among which 5 are active DoF, joints 2 and 5, as well as joints 3 and 4 are coupled such that they are driven by the same motor and produce the same angle.

$$x = \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{bmatrix} \tag{2.8}$$

Constraints play a crucial role in optimization problems, particularly in joint angle optimization. In such problems, the constraints are typically defined by the upper and lower bounds of the joints, as well as the relative position of the joints. In the case of SMuRo, a joint coupling constraint is also necessary to ensure that the robot exhibits a rat-like shape, due to the interdependent motion of the joints. These coupling constraints can be mathematically described by a set of equations that relate the movements of the different joints. Joint angle bounds are also applied to ensure that the robot's movement is within the feasible range for each joint's actuator. During non-linear optimization, the solver can iteratively adjust the joint movements until the desired robot behavior is achieved, while adhering to the physical limitations of the robot. The constraints of the optimization problem can be mathematically described as shown in Eq. (2.9) and (2.10).

$$b(x) = LowerBound_i \leq x_i \leq UpperBound_i, \quad i = 1, \ldots, 7 \tag{2.9}$$

$$g(x) = \begin{cases} x_2 = x_5 \\ x_3 = x_4 \end{cases} \tag{2.10}$$

Now that the constraints have been defined, next Jacobian is dervied. The Jacobian of the constraints is an important concept in constrained optimization. The Jacobian is a matrix that

contains the partial derivatives of the constraints with respect to the optimization variables. In other words, it describes how the constraints change as we move along each variable. The Jacobian also plays a key role in determining the feasibility of a solution. A solution is feasible if it satisfies all of the constraints. The Jacobian can be used to determine whether a given solution satisfies the constraints, and if not, which constraints are violated. In our case, the jacobian of the form Eq. (2.11).

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.11}$$

Where each row corresponds to a constraint, and each column corresponds to each optimization variable. Thus, the (i, j) element corresponds to the derivative of the ith constraint with respect to the jth variable.

The objective function forms the crux of the optimization problem as it contours the value that requires minimization and has to be chosen judiciously to ensure that the optimization algorithm generates the intended results. For the rat-to-robot mapping problem, the objective function needs to be a metric that gauges the disparity between the two contours that we are attempting to fit and has to be defined in terms of the optimization variables. The Euclidean error, also known as the L2 norm or the Euclidean distance, measures the distance between two points in a n-dimensional space. Euclidean error is a commonly used objective function for curve fitting, which involves minimizing the distance between the actual values and the desired values of the target variable. In curve fitting, the goal is to find a mathematical function that best describes the relationship between the input variables and the output variable. In the context of curve fitting, the Euclidean error measures the distance between the desired values of the target variable and the actual values of the target variable. It is defined as the square root of the sum of the squared differences between the desired point and current point as shown in Eq. (2.12).

$$\text{EuclideanError}(\mathbf{s}^*, \mathbf{s}) = \|\mathbf{s}^* - \mathbf{s}\|_2 = \sqrt{\sum_{i=1}^{3} (s_i^* - s_i)^2} \tag{2.12}$$

After choosing the objective function, it's gradient has to be derived. The gradient is a

vector that points in the direction of the steepest increase in the objective function. In other words, if we move in the direction of the gradient, the objective function will increase at the fastest rate. The gradient is used to update the parameters of a model during optimization by taking steps proportional to its negative. The gradient can be computed analytically for many common objective functions, or it can be approximated numerically using finite differences. Due to the complexity and non-linearity of the objective function, mainly owing to the use of forward kinematics, it is challenging to solve for the analytic gradient. Therefore, approximation of the gradient using central finite differences is employed to compute the gradient of the objective function as shown in Eq. (2.13).

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}, \qquad h \neq 0 \tag{2.13}$$

Where h is the step size, a small non-zero value that determines the distance between the two points used in the central difference approximation. Choosing a larger step size can prevent the algorithm from getting stuck in a local-minima, whereas smaller step sizes allow for a more accurate solution. Therefore the choice of the step size is also a crucial part in defining the optimization problem.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = EuclideanError(contour_{rat}, contour_{bot}) \\ \text{such that} \quad & LowerBound_i \leq x_i \leq UpperBound_i, \quad i = 1, \ldots, 7 \\ & x_2 = x_5 \\ & x_3 = x_4 \end{aligned} \tag{2.14}$$

With the above description, the optimization algorithm is fully defined. Nonetheless, there are other hyperparameters and methods that can be fine-tuned to enhance the optimization algorithm's performance. Hyperparameters play a vital role in determining the optimization algorithm's performance. The selection of hyperparameters can significantly impact the optimization algorithm's speed, accuracy, convergence rate, as well as its performance in terms of accuracy, generalization, and robustness. In the case of rat-to-robot mapping, the hyperparameters depicted in Table 2.3 were employed. These parameters were meticulously chosen through trial and error to ensure the mapping algorithm's robustness

and accuracy.

Table 2.3 Hyperparameters of Optimization Problem

Parameter Name	Description	Chosen value
Max Iterations	Maximum Number of Optimization Iterations	1000
Tolerance	Objective Function Value for Convergence	$3.82e-6$
Acceptable Tol	Acceptable Value for Convergence	$3.82e-3$
Step Size	Step size of each computation of gradient	$3.82e-4$
Weights	Weights for each point on Curve	$0.5 \sim 1.0$
Contour Length	Number of Interpolated Contour Points	100

The initial guess is another crucial component of an optimization problem, particularly for non-linear optimization problems. It refers to the starting point or initial value from which the optimization algorithm starts its search for the optimal solution. The selection of the initial guess can significantly impact the optimization process, as it can affect the algorithm's convergence rate and the final solution's quality. A poor initial guess can lead to slow convergence, being trapped in local minima, or even convergence to an incorrect solution. As the motion capture data is processed frame-by-frame, the laboratory rat's body position and orientation do not change significantly between frames. Therefore, employing the previous frame's result as the initial guess for the current frame's mapping is a good strategy for the problem. Finally the optimization algorithm has to be chosen.

In mathematical optimization, the goal is to find the optimal solution to a problem that satisfies certain constraints. Non-linear optimization is a subset of optimization that deals with problems where the objective function and/or the constraints are non-linear[45-46]. In the context of the rat-to-robot mapping problem, the non-linear nature of the objective function necessitates the use of a non-linear optimization algorithm. There are many optimization algorithms that can be used to solve non-linear optimization problems, each with its own strengths and weaknesses. Some popular optimization algorithms include gradient descent, conjugate gradient, Newton's method, and quasi-Newton methods like BFGS and L-BFGS. These algorithms differ in terms of their speed of convergence, memory requirements, and ability to handle certain types of constraints or objective functions. A comparison of common optimization algorithms is provided in Table 2.4. Considering the requirements of the mapping process, namely the complex objective function and constraints, Interior Point

methods is chosen.

Table 2.4 Comparison of Optimization Algorithms

Algorithm	Advantages	Disadvantages	Constraints	Applications
Gradient Descent	Simple, easy to implement	Slow convergence for large datasets	Unconstrained	Linear regression, logistic regression
Stochastic Gradient Descent	Faster convergence, good for large datasets	Can get stuck in local minima	Unconstrained	Neural networks, Deep Learning
Newton's Method	Faster convergence than GD	Requires Hessian matrix, computationally expensive	Unconstrained	Logistic regression, Support Vector Machines
Conjugate Gradient	Faster convergence than GD and Newton's method	Requires symmetric positive definite matrix	Unconstrained	Large-scale optimization problems
Quasi-Newton	Faster convergence than GD, no need for Hessian matrix	Can be unstable, requires tuning	Unconstrained	Nonlinear optimization problems
Interior Point	Fast convergence, handles constraints efficiently	Computationally expensive	Constrained	Linear and nonlinear programming, quadratic programming

Interior point methods[47] are a class of optimization algorithms that have gained popularity in recent years for solving non-linear optimization problems. Unlike traditional optimization algorithms that work by iteratively improving the solution by taking steps towards the optimal solution, interior point methods take a different approach. Instead of moving towards the boundary of the feasible region, as in traditional methods, interior point methods move towards the interior of the feasible region, hence the name. The main advantage of interior point methods is that they can handle complex constraints such as

inequality constraints, equality constraints, and non-linear constraints. Another reason to choose Interior Point optimization is the use of finite differences for the gradient of the objective function. The use of the gradient in optimization can be challenging when dealing with non-linear objective functions, as the gradient may not always be well-defined or computationally expensive to calculate. In these cases, optimization algorithms such as interior point methods are more suitable. However, interior point methods can be computationally expensive and require a large amount of memory, making them less suitable for some real-time applications. Since the dataset will be processed offline, Interior Point optimization is a suitable choice for the rat-to-robot mapping optimization algorithm.

IPOPT (Interior Point OPTimizer)[48] is a widely used open-source optimization library that implements interior point methods for solving non-linear optimization problems. It was developed by researchers at Carnegie Mellon University, IBM, and other institutions, and is now maintained by the COIN-OR Foundation. IPOPT uses a primal-dual interior point method that exploits the structure of the problem to achieve fast convergence. It also incorporates sophisticated algorithms for handling sparse linear systems, which can arise in large-scale optimization problems.

2.4 Summary

This chapter comprehensively presents the motion capture data that serves as the fundamental basis of motion planning. The discussion commenced with an overview of the motion capture system and data collection setup employed in generating the dataset. Subsequently, t-SNE clustering algorithm was leveraged to categorize the motion capture data into distinct behavioral patterns. Finally, a rat-to-robot mapping algorithm was introduced, which utilized non-linear optimization to establish the correlation between the motion capture data and the robotic rat's joints. Consequently, the motion capture dataset was converted from body points into joint angles, thereby enabling the teaching of the robotic rat rat-like behaviors and movements.

Chapter 3 Data-Driven Motion Generation

3.1 Overview

In order to use the mapped dataset of robotic rat joint angles for generating bio-mimetic motions in behavioral interaction, it is necessary to develop a comprehensive model that accounts for the different behaviors present in the dataset. The raw lab rat motion dataset cannot be directly used for this purpose, as it encompasses a wide variety of motions corresponding to different behaviors, as previously described. This results in a significant variability in each instance of a particular behavior. For example, different instances of the POU behavior may exhibit slightly different joint angles. To capture the variability in the entire dataset of the lab rat's motions, it is necessary to create a comprehensive mathematical representation of the different behaviors and their individual nuances. This involves developing a unified model that incorporates the common features of all the behaviors present in the dataset, while also capturing the subtle differences between each behavior. By constructing such a model, it becomes possible to characterize the dataset and identify the underlying behavioral patterns in the robotic rat's motions. This can enable more effective interaction between the robotic rat and the lab rat, as well as provide insights into the underlying neural and behavioral mechanisms that govern the lab rat's movements.

Learning from demonstrations (LfD)[49] is a technique that allows robots to learn new tasks by observing and imitating the target's motion. This approach to robot learning is particularly useful for tasks that are difficult or impossible to program explicitly, such as playing a musical instrument, cooking a complicated meal, or mimicking the behavior of an animal. The basic idea behind LfD is that an expert performs a task while the robot observes and records the expert's actions. The robot then uses this recorded data to learn how to perform the task itself. This process is also referred to as "Imitation Learning"[50]. One of the key benefits of LfD is that it allows robots to learn tasks quickly and effectively, without requiring a lot of trial and error. This is because the robot can leverage the expert's knowledge and experience, rather than having to figure things out on its own. Additionally, LfD can enable robots to perform tasks that are too complex for traditional programming

methods. There are several different approaches to LfD, including behavior cloning, inverse reinforcement learning, and active learning. Each of these approaches has its strengths and weaknesses, and researchers continue to explore new techniques and refinements.

Motion primitives[51] are basic, reusable motion patterns that can be combined to create complex motion sequences. They are commonly used in robotics, animation, and other fields that involve motion planning. In the context of robotics, motion primitives are often used as building blocks for robot motion planning and control[52]. The idea behind motion primitives is to break down complex movements into simpler, more basic actions that can be combined and sequenced to achieve the desired result. For example, a walking motion could be broken down into individual steps or strides, each of which is a motion primitive. By combining these primitives in a specific sequence, a robot could be programmed to walk forward or backward, turn, or even dance. One of the main advantages of using motion primitives is that they can simplify the process of programming and controlling robots. Rather than having to program every aspect of a robot's behavior from scratch, motion primitives provide a set of pre-defined building blocks that can be easily combined and adjusted to achieve the desired result. Motion primitives can also make it easier to adapt robot behavior to different environments or situations. For example, if a robot is navigating a cluttered environment, it may need to modify its movements to avoid obstacles. By adjusting the sequence or parameters of its motion primitives, the robot could quickly adapt to the new environment without requiring a complete overhaul of its programming. There are several different approaches to defining and using motion primitives in robotics, including dynamic movement primitives (DMPs)[53], probabilistic movement primitives (ProMPs)[54], and Gaussian mixture models (GMMs)[55].

Unlike other MP frameworks, ProMPs models the motion as a probability distribution which enables it to capture the variance in the animal behavioral motion patterns, additionally the ProMP motion representation is enhanced by multiple features and operations that facilitate its use in modeling and manipulating motion trajectories. Thus, ProMPs are chosen for modeling behavioral motions. In this chapter, Probabilistic Motion Primitive (ProMPs)[56] framework is employed to develop a probabilistic model of our lab rat's motion data. The Gaussian Basis Function, which serves as the building block for the

probabilistic model, is first derived. Next, the core concept and mathematical model as well as key properties and operations of ProMPs are derived. Finally, the ProMP model is utilized to produce bio-mimetic motion trajectories of different behaviors for the robotic rat to use during behavioral interaction.

3.2 Probabilistic Motion Primitives

Probabilistic methods such as ProMPs have the advantage of representing data in terms of a probability distribution, which allows the model to naturally account for the variance in the motions. In ProMPs[57], a motion primitive framework based on Gaussian Models is utilized. Gaussian models are a type of statistical model used to describe the probability distribution of a dataset. Gaussian models assume that the data being modeled follows a Gaussian or normal distribution. This distribution is characterized by a bell-shaped curve that is symmetric around the mean of the distribution. The mean represents the central tendency of the data, and the standard deviation represents the spread of the data around the mean. One of the benefits of Gaussian models is that they are easy to interpret and use. They provide a simple, yet powerful way to describe the distribution of a dataset and can be used to make predictions about future observations. Gaussian models are widely used in curve fitting applications due to their ability to accurately represent a wide range of continuous datasets such as trajectories.

3.2.1 Gaussian Basis Functions

Gaussian basis functions are derived from Gaussian models and are widely used in statistics and machine learning. Gaussian Basis functions are commonly used in machine learning, including in the context of learning from demonstration (LfD) techniques. They can be used to encode the demonstrated behavior into a mathematical model that can then be used to replicate the task. This is important in the context of behavioral interaction, since the robotic rat has to replicate predetermined rat-like motions. Gaussian basis functions can be used to represent complex trajectories using a weighted sum of simpler functions, where each simple function is a Gaussian curve centered at a specific point in the input space. The idea is to represent a complex trajectory using a linear combination of simpler Gaussian functions, each of which has a localized effect on the output. This allows for a more efficient and compact representation of the trajectory compared to using a large number of

individual data points. To create Gaussian basis functions, the input space is divided into a set of points, also known as centers c, which are chosen based on some criteria, such as the distribution of the data or the domain knowledge of the problem. The Gaussian basis functions are then defined as a function of the distance between the input value and each center. Specifically, the value of each basis function is calculated as a Gaussian function of the distance between the input value and the center, with the distance being defined using some metric such as Euclidean distance or Mahalanobis distance. The weights of the basis functions are then determined using a learning algorithm. The use of Gaussian basis functions has several advantages. First, they provide a smooth and continuous representation of the underlying function, which is useful for many applications, including control, prediction, and estimation. Second, the number of basis functions can be adjusted to control the complexity of the model, which helps prevent overfitting and improve generalization to unseen data. Finally, Gaussian basis functions can be efficiently computed using standard numerical libraries, making them a popular choice for many machine learning algorithms.

$$p(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \tag{3.1}$$

In order to derive Gaussian Basis Functions from Gaussian models, the first step involves considering the probability density function of a Gaussian distribution with mean μ and standard deviation σ, as expressed by Eq. (3.1). This probability density function can then be utilized to define a Gaussian Basis Function:

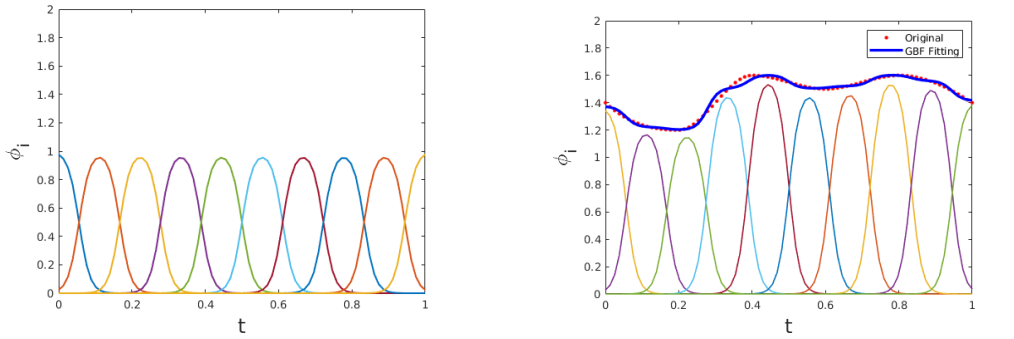
$$\phi_i(t) = \exp\left(-\frac{(t-c_i)^2}{2h}\right) \tag{3.2}$$

Here, c_i represents the center of the Gaussian Basis function, and h represents the standard deviation of the Gaussian distribution or the width of the Basis function. The Gaussian Basis function is centered at c_i and has a maximum value of 1 at this point, which decreases exponentially as t moves away from c_i. By using a set of Gaussian Basis Functions with different centers and standard deviations, a wide variety of functions can be represented as linear combinations of these basis functions. A joint trajectory, such as the ones produced by the Rat-to-Robot mapping, can be expressed using a linear combination

of the Gaussian Basis functions as shown in Eq. (3.3).

$$y(t) = \mathbf{\Phi}(t)\mathbf{w} + \epsilon = \mathbf{q}(t) \tag{3.3}$$

where $\mathbf{\Phi}$ is an $T \times N$ matrix with elements $\Phi_{t,i} = \phi_i(\frac{t}{T})$, and $\boldsymbol{\epsilon}$ is a vector of Gaussian noise terms with mean zero and covariance matrix Σ. The noise terms capture the inherent variability in natural motion and other sources of uncertainty. Given that the trajectory is continuous, c_i is selected from the interval $[0, 1]$ to enable the Gaussian basis functions to span the entire trajectory. By doing so, a probabilistic model that accounts for the variance in the motion of the joint trajectories is created. In the subsequent section, the approach by which the model's parameters are acquired through the demonstration of the ProMP model of Motion Primitives is presented.



(a) A set of Gaussian Basis functions.

(b) An example of trajectory fitting using GBF.

Figure 3.1 Gaussian Basis Functions as a Linear Combination

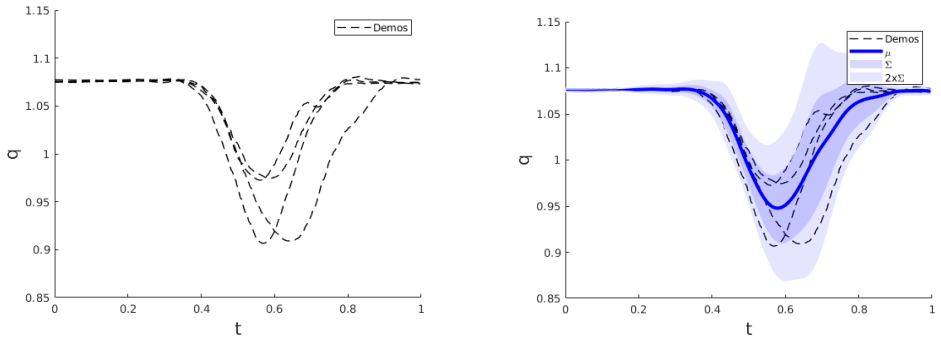
Where h is the width of the basis function, c_i is the center of the ith basis function. It can be seen from Figure 3.1 that the width of the basis function h and the number of basis functions m are crucial parameters of the ProMP model. The width of the basis functions h determines the horizontal extent of the basis function and how much of the trajectory it will represent; a very small value of h means that the length that the basis function covers is narrow, which allows for . The number of basis functions m affects the smoothness of the fitted curve, such that a higher number of basis functions allows for a very precise fit of the desired trajectory. However, a high value of m can also lead to over-fitting of the model, allowing for less variability in the representation of the motion trajectories. Both parameters

have to be tuned in unison in order to ensure that the produced model accurately fits the desired trajectories while also maintaining the variability in representation.

3.2.2 Learning from Demonstrations

A motion primitive can be defined as a generic, reusable, and modular plan that can be adapted and combined to solve a range of tasks. Given a set of demonstration trajectories, where demonstration consisting of states, we want to learn a probabilistic model that captures the underlying structure and variability of the demonstrated motion. The basic idea of Probabilistic Motion Primitives (ProMPs) is to represent a motion primitive as a linear combination of basis functions, where each basis function represents a different component of the motion. The weight vector represents the contribution of each basis function to the motion primitive, and is assumed to be drawn from a Gaussian distribution $\mathbf{w} \sim \mathcal{N}(\mu, \Sigma_w)$, where μ represents the most likely set of weights for generating trajectories, while the covariance matrix $\mathbf{\Sigma}_w$ encodes the uncertainty in the weight estimates. Since a ProMP represents multiple ways to execute an elemental movement, multiple demonstrations are also necessary to learn the parameters μ and $\mathbf{\Sigma}_w$.

To estimate the parameters of the ProMP model, the maximum likelihood estimation is employed. Let $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M$ denote the set of observed trajectories. The corresponding likelihood function can be expressed as Eq. (3.4). The process is shown in Figure 3.2



(a) Demonstration Motion Trajectories (b) Estimated ProMP Model

Figure 3.2 Estimating ProMP Model Parameters using MLE

$$p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M|\mathbf{w}, \mathbf{\Sigma}) = \prod_{m=1}^{M} p(\mathbf{y}_m|\mathbf{w}, \Sigma) \cdot p(\mathbf{w}|\boldsymbol{\mu}, \mathbf{\Sigma}_w) \tag{3.4}$$

Here, $p(\mathbf{y}_m|\mathbf{w}, \Sigma)$ represents the likelihood of observing trajectory $\mathbf{y}_m$ given the weights $\mathbf{w}$ and the noise variance Σ. On the other hand, $p(\mathbf{w}|\boldsymbol{\mu}, \mathbf{\Sigma}_w)$ represents the probability density function of the weights given the mean and covariance.

In summary, the ProMP model represents motion trajectories as linear combinations of Gaussian basis functions, with weights that follow a Gaussian distribution. The parameters of the model are learned using maximum likelihood estimation, which involves finding the most likely set of weights that generate the observed trajectories, while also accounting for uncertainty in the weight estimates.

3.3 Properties of Probabilistic Motion Primitives

After establishing the fundamental aspects of the ProMP model, additional properties and operations of the model need to be derived to enhance its capabilities and allow for further controllability of the model. This section focuses on deriving the essential properties of ProMP Models that are required to generate feasible trajectories. The derivation of these properties is crucial to produce rat-like trajectories from the ProMP Models.

3.3.1 Temporal Modulation

Due to the fact that the motions corresponding to different behaviors take different amounts of time to perform, it is necessary to normalize the length of the motion trajectories such that all behaviors can be expressed using a ProMP model with the same parameters. Thus, all of the behavior motion trajectories are normalized and expressed in the interval $[0, 1]$.

This allows for more convenient processing of the dataset, and production of the probabilistic models. However, motions corresponding to behaviors are time-based, meaning that they are usually performed within a specific amount of time, and cannot be arbitrarily stretched and compressed. Hence, it is also necessary to reverse the normalization when producing motion trajectories as a function of time.

In order to solve this problem, an intermediate phase variable z is introduced to decouple

the movement from the time signal. The phase can be any function monotonically increasing with time $z(t)$. By modifying the rate of the phase variable, the speed of the movement can be modulated. The basis functions $\phi(t)$ now directly depend on the phase instead of time, such that $\phi(t) = \phi(z(t))$.

3.3.2 Coupled Trajectories

The motion primitives for each behavior involve all of the robotic rat's joints moving in coordination to produce the rat-like motion. In order to capture the correlation between the joints' movements while performing a specific behavior, all of the joints' trajectories have to be learned together. Hence, the model and the LfD process are formulated in Matrix format as shown below:

$$\begin{bmatrix} y(t)_1 \\ \vdots \\ y(t)_n \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}(t)_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{\Phi}(t)_n \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \epsilon \\ \vdots \\ \epsilon \end{bmatrix} \tag{3.5}$$

This allows the motion primitive to incorporate the correlation between the joint movements directly and ensures that the produced trajectories mimic the original motion.

3.3.3 Via Points

Introducing via-points to the motion primitive model is a valuable technique to guide the generation of trajectories towards specific locations[58]. Via-points can be incorporated into the probabilistic model as desired observations, whereby the Gaussian distribution of the weight vector is updated accordingly. By applying Bayes' Theorem, the revised distribution that passes through the via-point can be computed as demonstrated in Eq. (3.6).

$$\boldsymbol{\mu}_w^* = \boldsymbol{\mu}_w + \mathbf{\Sigma}_w \mathbf{\Psi} (\mathbf{\Sigma}_y + \mathbf{\Psi}^T \mathbf{\Sigma}_w \mathbf{\Psi})^{-1} (y_t - \mathbf{\Psi}^T \boldsymbol{\mu}_w) \tag{3.6}$$

$$\mathbf{\Sigma}_w^* = \mathbf{\Sigma}_w - \mathbf{\Sigma}_w \mathbf{\Psi} (\mathbf{\Sigma}_y + \mathbf{\Psi}^T \mathbf{\Sigma}_w \mathbf{\Psi})^{-1} \mathbf{\Psi}^T \mathbf{\Sigma}_w \tag{3.7}$$

Even with the conditioning of the ProMP, the model remains within the bounds of the original distribution as shown in Figure 3.3, which indicates that the modulation is also acquired from the original demonstrations and the produced trajectory still follows the same motion patterns as the original model.

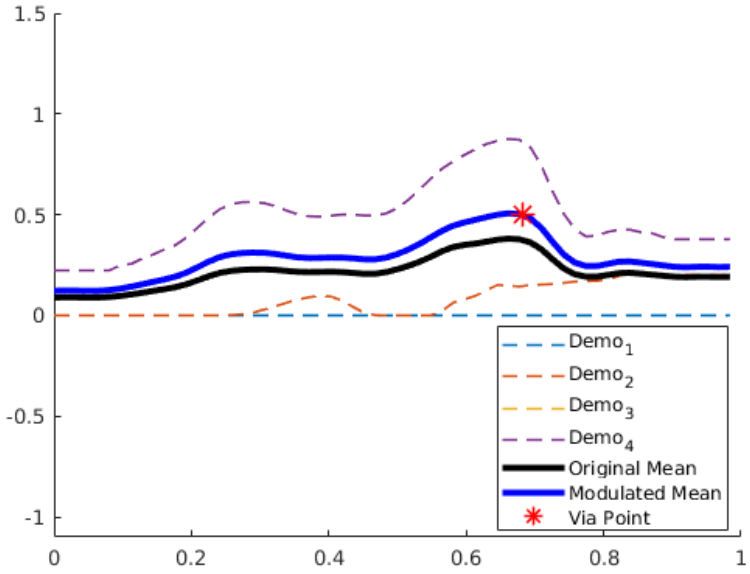


Figure 3.3 Modulation of Via-Point

3.3.4 Motion Primitive Operations

Blending and Combining are fundamental operations in motion primitives that enables the mixing and reuse of motion primitives. In particular, Blending allows for the seamless transition between two different trajectories; it involves smoothly shifting from one motion's trajectory to another, resulting in a continuous motion. An activation function $a(t)$ can be utilized to blend the target primitives. By adjusting the values of the activation function, the activation of a motion primitive can be regulated. To ensure the activation function's output falls within the range of 0 to 1, a Sigmoid function is commonly utilized. Its general form is presented in Eq. (3.8).

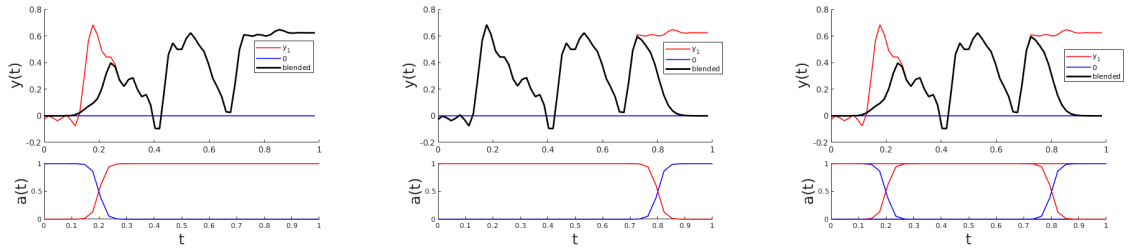
$$a(t) = \frac{1}{1 + e^{-(t+t_s)}} \tag{3.8}$$

Here, t_s denotes the time step at which the transition between the two motion primitives occurs, and s represents the rate of the switch. The outcome of the blending process is a new motion primitive that seamlessly transitions between the two component MPs. This is illustrated in Eq. (3.9).

$$\boldsymbol{\mu_{new}} = a(t)\boldsymbol{\mu_1} + (1 - a(t))\boldsymbol{\mu_2} \tag{3.9}$$

$$\boldsymbol{\Sigma_{new}} = a(t)\boldsymbol{\Sigma_1} + (1 - a(t))\boldsymbol{\Sigma_2} \tag{3.10}$$

Blending is a crucial operation for behavioral interaction since it allows the robot to produce trajectories that transition smoothly from one to another; This aids in the quick transitioning from executing one behavioral motions to another in order to react to the laboratory rat's unpredictable motion patterns.



(a) Blending between 0 and $y_1(t)$ at $0.1s$

(b) Blending between $y_1(t)$ and 0 at $0.9s$

(c) Blending from 0 to $y_1(t)$ then to back 0

Figure 3.4 Blending Operation of Motion Primitive with 0

Combining on the other hand is the process of producing a motion primitive with the properties of two or more motion primitives. It usually involves computing the sum or the product of the motion primitives. The sum of the motion primitives incorporates all the properties of each single primitive, whereas the product produces a motion primitive from the overlap of both primitives. Combining operations however, do not make sense in the context of behavioral motion generation, since it is desired to preserve the characteristics of the original motion.

3.4 Summary

This chapter presented the utilization of the probabilistic motion primitive model to develop a controllable model that captures the variance in the rat's motion. The derivation of the ProMP model and its essential properties were discussed. Subsequently, the model was utilized to construct an MP library from the motion capture data of the laboratory rat.

Chapter 4 Real-time Rat Tracking

4.1 Overview

In the context of robot-animal behavioral interaction, there are several prerequisites that motion planning must meet. Continuous visual tracking of the target animal is one those requirements. Visual tracking is a crucial component of conducting robot-animal behavioral experiments as it allows researchers to accurately monitor and analyze the animal's behavior in response to the robot's movements and stimuli[59]. By using visual tracking techniques, researchers can track the animal's movements and reactions with high precision and accuracy, which is essential for understanding the animal's behavior and preferences. Moreover, visual tracking enables the robot to adjust its behavior and movements in real-time to better interact with the animal. For example, if the animal appears to be scared or anxious, the robot can adjust its movements to become less threatening or back away to give the animal space. This type of real-time adjustment is only possible with accurate visual tracking. Visual tracking also enables researchers to collect large amounts of data on the animal's behavior, which can be analyzed to gain insights into the animal's behavior patterns, preferences, and responses to different stimuli. This data can then be used to improve the design and behavior of the robot, leading to more effective and non-invasive interactions with the animal.

In this chapter, building on the extracted rat-like motion primitives of the previous chapter, a real-time motion planning strategy for robot-rat behavioral interaction is introduced. First, a rat detection algorithm based on YOLOv4 is employed to extract the current position of the laboratory rat in the image. Next, a real-time laboratory rat tracking algorithm based on Visual Servoing is introduced to ensure that the robotic rat can track the laboratory rat's position in real-time. Finally, a behavioral interaction strategy is developed to conduct robot-rat behavioral experiments.

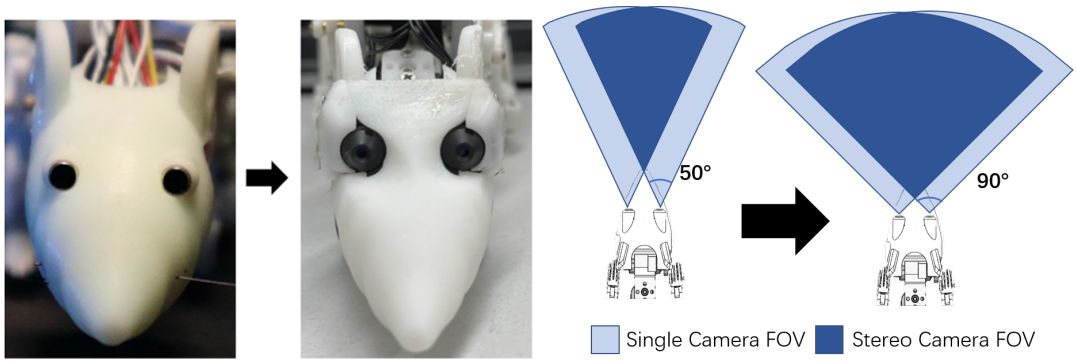
4.2 Robotic Rat Perception System

In the context of robot-animal behavioral interaction, the perception of a robotic rat refers to the ability of the robot to sense and interpret the environment and the animal's behavior using various sensors and algorithms. Perception is a critical component of the

robot's ability to detect and interact with the animal in a meaningful way. Hence, considering the necessity of visual perception in robot environment understanding, target recognition, etc., it is clear that the same rich visual perception is needed to obtain information about the position and behavior of the target laboratory rat in the context of robot-rat autonomous interaction scenario. In order to solve these needs, the robotic rat is equipped with a binocular stereo camera, such that it is capable of detecting and and locating the target laboratory rat. The latest version of the Robotic Rat is equipped with an upgraded stereo cameras with enhanced capabilities. The new cameras have a much higher resolution as well as a larger field of view (FOV) allowing for more accurate detection of the laboratory rat and a wider range of vision for tracking. Detailed Comparison between the old and new camera parameters is presented in Table 4.1 and Figure 4.1.

Table 4.1 Parameters of Stereo Camera

Parameter	Old Camera	New Camera
Dimensions (Length×Width× Height)	$\approx 18mm \times 4mm \times 4mm$	$\approx 12mm \times 12mm \times 12mm$
Resolution	1280×480	2560×720
Stereo Synchronization	Software	Hardware
Horizontal FOV	50°	90°
Operational Voltage	3.3V	3.3V



(a) Old and New Cameras of the Robotic Rat

(b) Comparison of FOV Increase

Figure 4.1 Comparison between Old and New Cameras of the Robotic Rat

4.2.1 Depth Estimation

The binocular perception of animals plays an important role in their own judgment and in understanding information about distant targets. Many animals use two eyes with overlapping visual fields to perceive their surroundings, thus using stereo vision to calculate their own distance to a target and to assist them in more advanced behaviors and assist in higher-level behaviors. Stereo vision has demonstrated a variety of useful functions in animal survival. For example, from distance measurement for navigation to rapid localization for prey capture. Although the vast majority of animals have eyes. Early studies thought that stereo vision was limited to animals with forward-facing eyes, such as primates, carnivores and predators in general. However in recent years, the effectiveness of stereo vision in the lateral eye has been demonstrated in a variety of animals, including rats[60].

Binocular stereo camera systems consist of two cameras that are positioned side-by-side, with each camera capturing an image of the same scene from a slightly different viewpoint. These images are then used to extract depth information about the scene and create a 3D representation. The binocular stereo camera model is a mathematical representation of this process, which can be used to estimate the depth of objects in a scene.

The binocular stereo camera model is based on the principles of triangulation, which involves using the known distance between two points and their corresponding angles to determine the distance to a third point. In the case of a binocular stereo camera system, the two cameras serve as the two points, and the depth of an object in the scene is determined by calculating the angle between the two camera viewpoints and the position of the object in each camera's image.

Although it is possible to perform depth estimation using a monocular camera[61], binocular stereo camera model is more commonly used to estimate the depth information of a 3D scene using two cameras placed at different positions. The basic principle of stereo vision is triangulation, which involves finding the 3D coordinates of a point in space by using the 2D coordinates of the same point in both camera views. The binocular stereo camera model describes the relationship between the 3D world coordinates, the 2D image coordinates, and the camera parameters for each camera.[62]

Let's consider a 3D point in the world coordinate system with coordinates (X, Y, Z) and

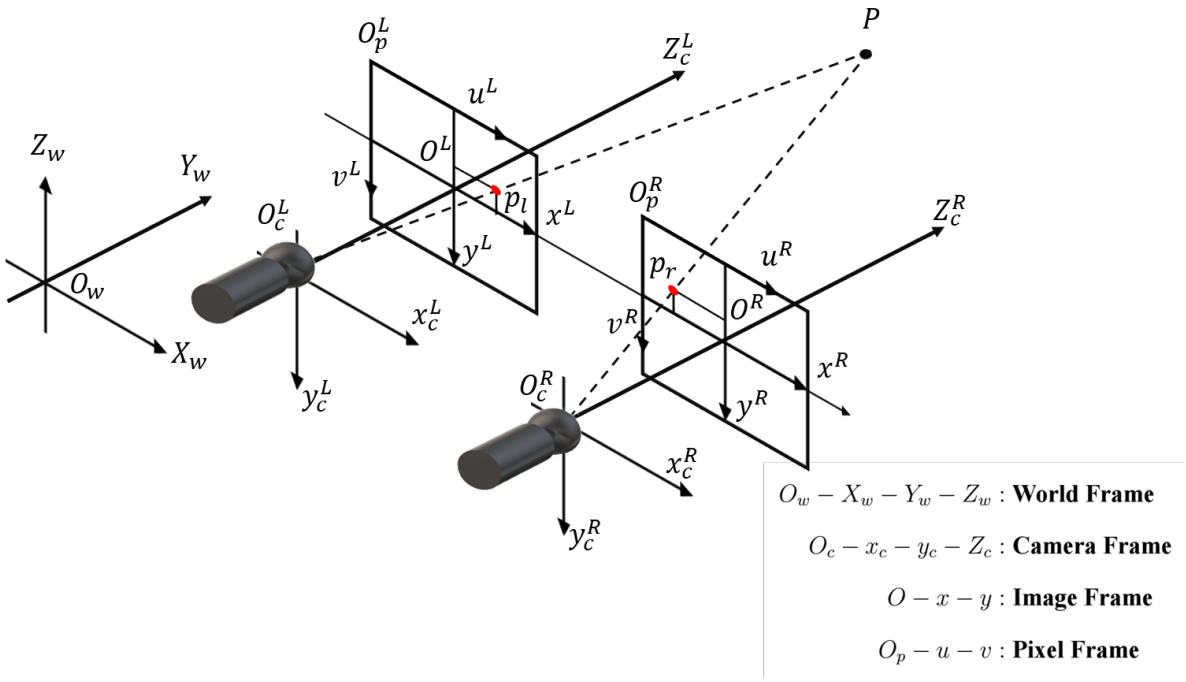


Figure 4.2 Binocular Stereo Camera Model

its corresponding image points in two cameras with image coordinates (x_l, y_l) and (x_r, y_r), respectively. The cameras are assumed to be aligned along the horizontal axis, and the distance between the camera centers is denoted by b. The camera parameters for both cameras include the intrinsic parameters and extrinsic parameters. The intrinsic parameters describe the internal characteristics of the camera, such as the focal length and image sensor size, while the extrinsic parameters describe the position and orientation of the camera relative to the world coordinate system.

The binocular stereo camera model can be represented mathematically as follows:

For the left camera:

$$\begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_l & 0 & c_{x,l} \\ 0 & f_l & c_{y,l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{x,l} \\ r_{21} & r_{22} & r_{23} & t_{y,l} \\ r_{31} & r_{32} & r_{33} & t_{z,l} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.1}$$

For the right camera:

$$\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} f_r & 0 & c_{x,r} \\ 0 & f_r & c_{y,r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{x,r} \\ r_{21} & r_{22} & r_{23} & t_{y,r} \\ r_{31} & r_{32} & r_{33} & t_{z,r} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.2}$$

where (u_l, v_l) and (u_r, v_r) are the image coordinates of the 3D point in the left and right cameras, respectively, and $f_l, f_r, c_{x,l}, c_{x,r}, c_{y,l}, c_{y,r}$ are the intrinsic camera parameters. The extrinsic parameters are represented by the rotation matrix $R = [r_{ij}]$ and the translation vector $T = [t_x, t_y, t_z]$.

The distance between the two camera centers is denoted by b, and the horizontal displacement between the image points in the two cameras is denoted by d. The disparity between the image points is defined as $d = u_l - u_r$. The depth of the 3D point in the world coordinate system can be calculated as follows:

$$Z = \frac{bf_l}{d} \tag{4.3}$$

This equation can be derived by using the similarity of triangles between the 3D point, the left camera center, and the left image point, and the 3D point, the right camera center, and the right image point. The ratio of the distances between the 3D point and the left and right camera centers is equal to the ratio of the distances between the left and right image points and the left camera center. This ratio can be expressed as d/f_l, which gives the depth Z of the 3D point. This model allows the robotic rat to utilize the camera to perform depth estimation of the detected rat in the image.

4.2.2 Laboratory Rat Detection

Being able to detect the laboratory rat in the surrounding environment is a prerequisite for tracking and conducting behavioral interaction. Target detection is a crucial task in various fields, including computer vision, remote sensing, and security. The detection of specific objects or patterns in images or videos is a challenging problem that has been extensively studied over the years. With the emergence of deep learning, new approaches

have been proposed to address this problem, leveraging the power of convolutional neural networks (CNNs)[63] to learn features directly from data. In recent years, significant progress has been made in using deep learning for target detection, achieving state-of-the-art performance in various benchmarks. YOLOv4 is a popular real-time object detection system that was released in April 2020[64]. YOLO stands for ”You Only Look Once,” and it is a type of neural network that can identify and locate objects within an image or video in real-time. YOLOv4 builds upon the previous versions of YOLO[65] with several improvements in terms of accuracy, speed, and performance. It is known for its ability to detect small objects, such as birds, with high accuracy, and its ability to detect objects in crowded scenes with multiple overlapping objects. YOLOv4 is commonly used in various applications, such as self-driving cars, surveillance systems, and robotics. Our team have previously developed a laboratory rat detection model based on YOLOv4[66]. The model is capable of accurately detecting the laboratory rat's position in an image and produces a bounding box to indicate the location of the rat relative to the image frame (i.e. in pixel coordinates) as shown in 4.3. In this section, leveraging the robotic rat capabilities in detecting the laboratory rat in real-time, a rat-tracking algorithm based on Visual Servoing[67] is introduced.

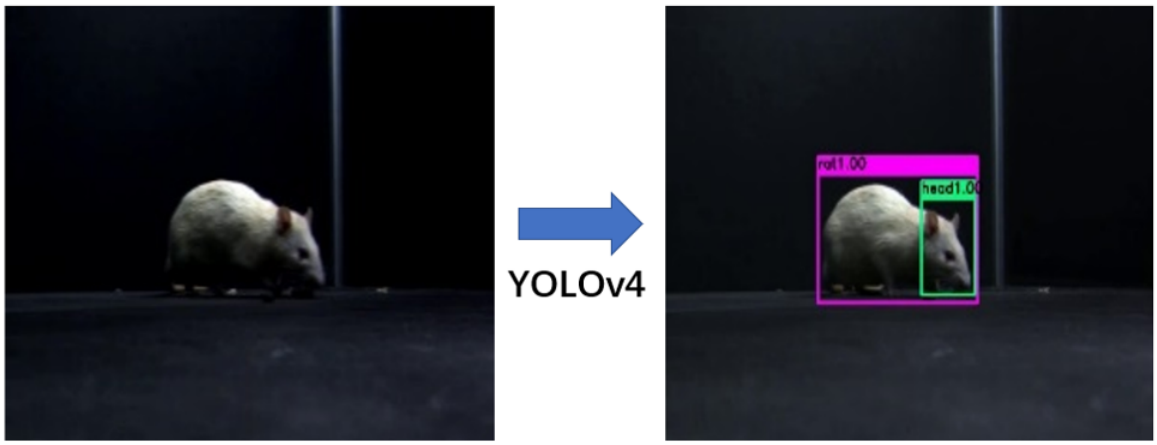


Figure 4.3 Extraction of Laboratory Rat Location from Image

In order to achieve the experimental mouse localization, stereo matching is needed to obtain the parallax according to the analysis in the previous section. Stereo matching can be divided into global matching and local matching in terms of constraint range. Global

matching has high accuracy but slow computation speed, and considering the fast movement of experimental rats and high requirement for real-time, local matching method is used. In turn, local matching is divided into region-based, feature-based and gradient-based matching methods. The ORB algorithm[68], which is the fastest and most accurate method, is selected for feature matching. Therefore, the localization algorithm of the experimental rat was designed based on YOLOv4 target detection, ORB feature point matching, and stereo depth estimation model. The overall localization algorithm flow is shown in Figure 4.4. First, the left and right images are acquired based on the miniature binocular vision system, and the binocular images are transferred to the PC side through WiFi graphical transmission, and the corrected left images are input to the target detector YOLOv4 for detection. Stereo matching based on local feature points is then performed to obtain a sparse parallax map.

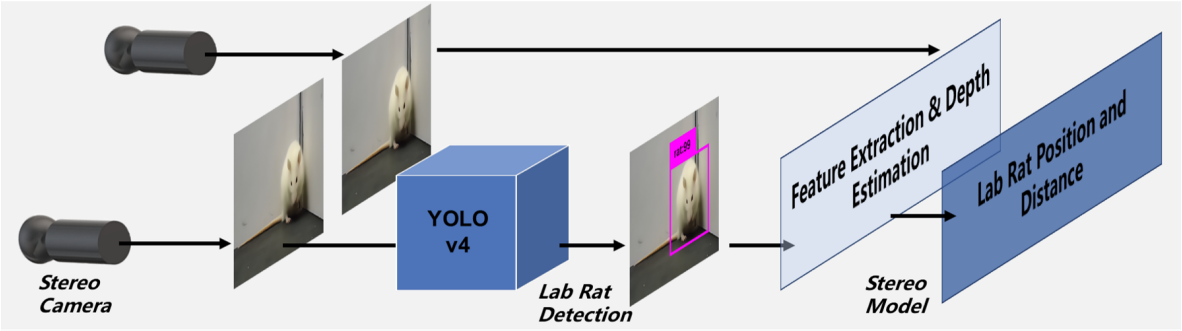


Figure 4.4 Full laboratory Rat Detection Algorithm Based on YOLOv4 and Stereo Cameras

With both target detection and depth estimation serving as the basis, the next section introduces a laboratory rat tracking algorithm.

4.3 Laboratory Rat Tracking Algorithm Based on IBVS

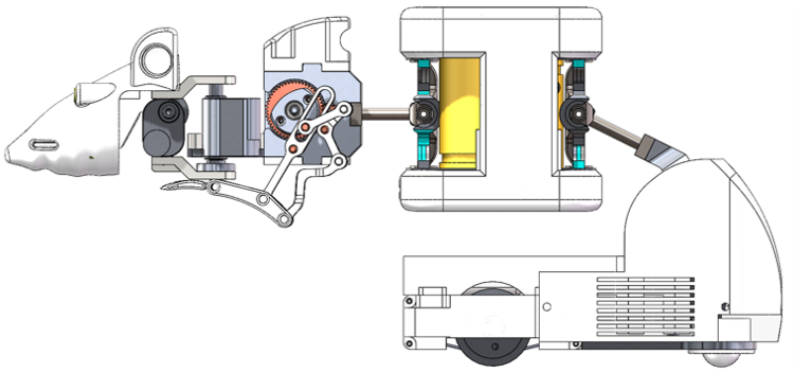
Since the robotic rat is now able to detect the laboratory rat's position in the image returned by the robotic rat's Stereo Camera, it is now possible to develop a rat-tracking algorithm to ensure that the laboratory rat remains in the robotic rat's field of vision for the duration of the behavioral interaction experimentation. In this section, a real-time laboratory rat tracking algorithm for the robotic rat based on visual servoing is introduced. Visual Servoing is a fundamental technique in robotics that has received increasing attention in recent years. It involves using visual feedback to control the motion of a robot, enabling it to navigate and

interact with its environment in a precise and reliable manner. The technique has a wide range of applications, including industrial automation, surveillance, and exploration, and has been used in a variety of robotic platforms, including manipulators, mobile robots, and unmanned aerial vehicles (UAVs)[69].

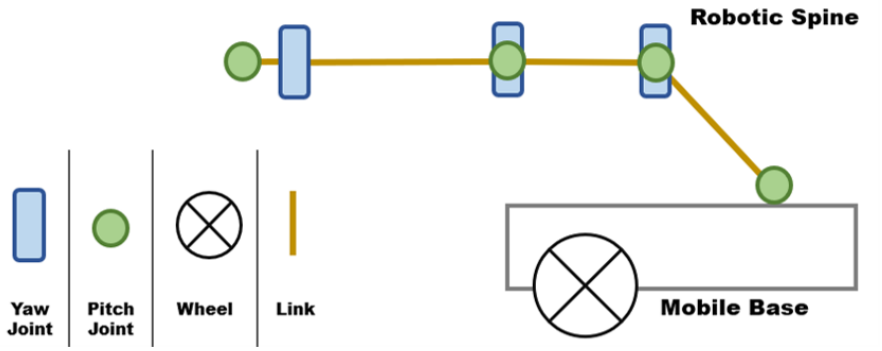
The main idea behind Visual Servoing is to use a camera or other visual sensor to monitor the environment and provide feedback to the robot's control system[70-71]. This feedback can be used to adjust the robot's motion in real-time, allowing it to navigate through complex and dynamic environments. The process involves several steps, including image acquisition, image processing, feature extraction, and control computation. Visual Servoing can be classified into two main categories: image-based visual servoing (IBVS) and pose-based visual servoing (PBVS). In IBVS, the feedback is based on the image features, such as points or lines, and the control input is computed using the geometric relationship between these features and the robot's motion. In PBVS, the feedback is based on the 3D pose of the target object or the camera, and the control input is computed using the difference between the desired and actual poses. Visual Servoing has several advantages over other robotic control techniques. One of the main advantages is its ability to deal with uncertainties and disturbances in the environment, such as occlusions, noise, and changes in lighting conditions. It is also well-suited for applications that require precise and flexible motion control, such as assembly, manipulation, and inspection tasks.

Examining the design of the robotic rat SMuRo, we find that it can viewed as a mobile manipulator[72] as shown in Figure 4.5. Mobile manipulators are robotic systems that combine a mobile platform with a robotic arm, allowing them to move around and interact with their environment. These systems are highly versatile and able to perform complex tasks in unstructured environments. The main challenge in motion planning for mobile manipulators is the coupling between the base and arm motions. The robot must simultaneously plan for both its base motion and its arm motion in order to achieve its goal. This requires coordination between the base and arm controllers, as well as a sophisticated motion planning algorithm that can take into account the constraints of both subsystems.

One common approach to motion planning for mobile manipulators is to use a two-stage process[73]. In the first stage, a motion planner generates a path for the mobile base to



(a) Robotic Rat Model



(b) Simplified Mobile Manipulator Model

Figure 4.5 Simplification of Robotic Rat's Model

navigate to the goal location. In the second stage, the arm planner determines a sequence of arm motions that enable the robot to perform the desired motions along the planned path. The two planners are coordinated so that the arm planner takes into account the motion of the base as it navigates to the goal location. Another approach is to use a unified motion planner that simultaneously plans for both the base and arm motions. This approach can be more computationally efficient than the two-stage approach, as it avoids the need for coordination between separate planners, but it significantly increases the complexity of the planning process. In the case of target tracking, where the camera sensor is attached to the manipulator's end-effector, which is the tool or device attached to the end of the robot arm, the task can be accomplished using an approach similar to the two-stage process. The manipulator would first use its mobile base to navigate to the general location of the target. Once the manipulator is in the vicinity of the target, the camera sensor would be used to detect and track the target. The motion planning algorithm would then determine a sequence

of arm motions that allow the end-effector to maintain the camera's view of the target. This involves adjusting the position and orientation of the end-effector as the target moves, as well as adjusting the position and orientation of the mobile base to maintain the desired view of the target.

Building on the rat-detection results from the previous section, in this section, using the bounding box provided by YOLOv4 as a basis, a rat-tracking algorithm based on IBVS is introduced to ensure that the robotic rat is capable of tracking the laboratory rat in real-time.

4.3.1 Target Features Definition

In an image-based visual servoing (IBVS) tracking algorithm, image features are used to compute the error signal that drives the robot motion. Image features are typically extracted from the camera images and represent specific characteristics of the target that can be used to track its position and orientation relative to the robot. The choice of image features depends on the characteristics of the target being tracked and the capabilities of the camera system. Common image features used in IBVS tracking algorithms include: Points or corners, Edges or contours, Blobs or regions, Features learned by deep learning algorithms. Once the image features are extracted, they are used to compute the control error. The aim of all vision-based servoing schemes is to minimize the control error $e(t)$ which is usually defined as the difference between desired features and current features, a general form of the error is Eq. (4.4).

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), a) - \mathbf{s}^* \tag{4.4}$$

Where the vector $\mathbf{m}(t)$ is a set of image feature measurements, such as the image coordinates of interest points or the image coordinates of the centroid of an object, which are used to compute a vector of k visual features. Whereas a represents additional knowledge about the system, such as camera intrinsic parameters or 3-D models of objects. The vector s^* contains the desired values of the features. For IBVS, s is a set of image features that can be extracted from the camera's images. Since the rat detection algorithm provides a bounding box (BBox), the tracking target feature is defined as the center point of the bounding box Eq. (4.5).

$$s = \mathbf{P}_{BBox} = \begin{bmatrix} x_{BBox} \\ y_{BBox} \\ Z \end{bmatrix} \tag{4.5}$$

$$s^* = \mathbf{P}_* = \begin{bmatrix} x^* \\ y^* \\ Z^* \end{bmatrix} \tag{4.6}$$

Choosing the center of a bounding box of an object in an image as the image feature for Image-based visual servoing (IBVS) is a common approach to track the position of an object in the image. The center of a bounding box can be calculated as the average position of the corner points of the box, and represents the position of the object in the image. Using the center of the bounding box as the image feature simplifies the tracking process, as it reduces the number of features that need to be tracked, and the computation needed to calculate them. Using the center of the bounding box as the image feature has some limitations, as it does not provide information about the orientation of the object in the image. However, since in this tracking task we are not concerned about the orientation of the laboratory rat, it is not necessary to have information about the orientation of the laboratory rat. Having defined the image feature to be tracked, the feature error can be expressed simply as:

$$\mathbf{e}(t) = \begin{bmatrix} x_{BBox} - x^* \\ y_{BBox} - y^* \end{bmatrix} \tag{4.7}$$

4.3.2 Interaction Matrix

In order to relate the error between the image features to the motion of the camera in 3-D space the computation of the Interaction Matrix for the camera is necessary. The Interaction Matrix is an essential component of Image-Based Visual Servoing (IBVS), as it establishes a correlation between the camera velocity and the image velocity. To be more precise, the Interaction Matrix facilitates the mapping between the motion of the camera and the movement of features in the image. This section aims to present the derivation of the

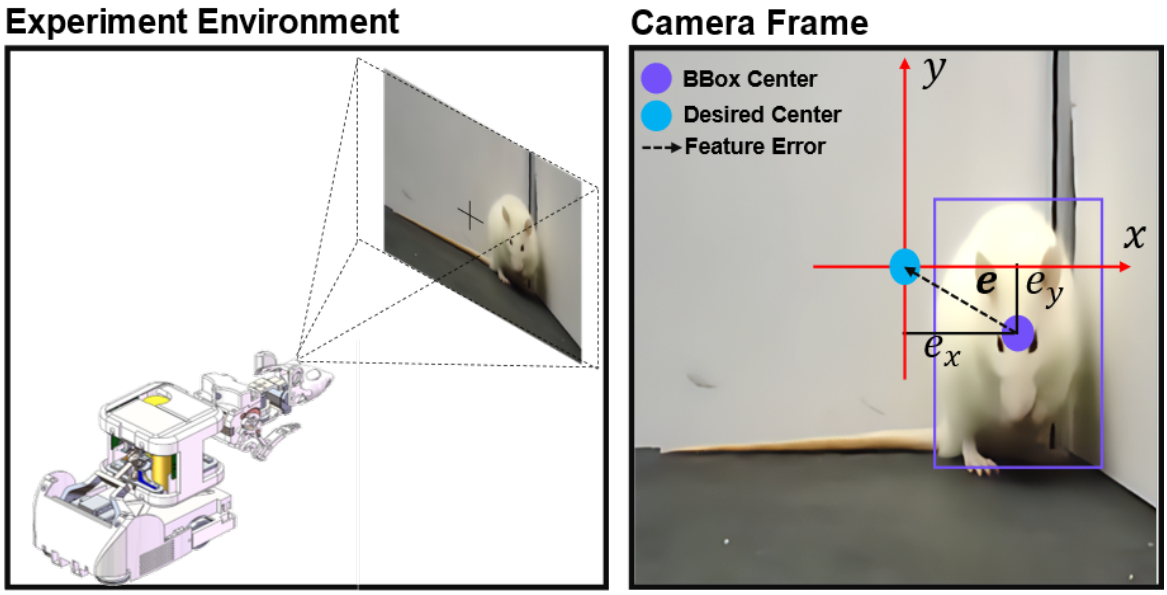


Figure 4.6 Feature Error

Interaction Matrix for a 6 degrees-of-freedom (DoF) camera in a 3-D space attached to the end of a robotic manipulator. In the Camera Frame, if we consider a point with 3-D coordinates $\mathbf{P} = (X, Y, Z)$, it generates a projection in the image plane as a 2-D point with coordinates $\mathbf{p} = (x, y)$, as per the Pinhole camera model Eq. (4.8). As a result, the following relationship can be established:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.8}$$

$$\begin{cases} x = X/Z = (u - c_u)/f_x \\ y = Y/Z = (v - c_v)/f_y \end{cases} \tag{4.9}$$

Where $\mathbf{m} = (u, v)$ denotes the coordinates of the image point measured in pixel units, and $\mathbf{a} = (c_u, c_v, f_x, f_y)$ represents the camera's intrinsic parameters. Specifically, c_u and c_v denote the coordinates of the principal point, whereas f_x and f_y denote the focal length in the x and y directions. With the relationship between the coordinates of the target point in

the camera frame and the coordinates of its projection established, it is possible to determine the correlation between the velocity of the projection and the velocity of the target point by taking the derivative with respect to time.

$$\begin{cases} \dot{x} = & \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = & \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z \end{cases} \tag{4.10}$$

Next, the velocity of the 3-D point to the camera spatial velocity can be related using:

$$\dot{\mathbf{P}} = -\mathbf{v}_c - \omega_c \times \mathbf{P} \Leftrightarrow \begin{cases} \dot{X} = & -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = & -v_y - \omega_z X + \omega_x Z \\ \dot{Z} = & -v_z - \omega_x Y + \omega_y X \end{cases} \tag{4.11}$$

By substituting the spatial velocity Eq. (4.11) into the velocity Eq. (4.10), we obtain:

$$\begin{cases} \dot{x} = & -v_x/Z + xv_z/Z + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ \dot{y} = & -v_y/Z + yv_z/Z + (1+y^2)\omega_w - xy\omega_y + x\omega_z \end{cases} \tag{4.12}$$

Expressing the velocity Eq. (4.12) in a matrix form, we obtain:

$$\dot{\mathbf{p}} = L_x \mathbf{v_c} \tag{4.13}$$

In the above matrix equation, $\mathbf{v_c} = \begin{bmatrix} v_c & \omega_c \end{bmatrix}^T$ represents the spatial velocity of the camera, and L_x denotes the Interaction Matrix of the form:

$$L_x = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & (1+y^2) & -xy & -x \end{bmatrix} \tag{4.14}$$

It should be noted that the interaction matrix presented in Eq. (4.14) includes the Euclidean distance Z, which is the distance between the origin of the camera frame and the target point. Therefore, estimating the value of Z is necessary when using this form of the interaction matrix. Fortunately, this can be accomplished using the stereo camera equipped on the robotic rat, as shown in Figure 4.1.

In Eq. (4.14), the matrix $L_x \in \mathbb{R}^{2\times 6}$ corresponds to a single image feature, where the first and second rows represent the projection of the feature onto the x and y axes of the camera frame, respectively. Each column of L_x represents a component of the spatial velocity of the camera. The first three columns represent the instantaneous linear velocities, denoted by $v_c = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$, while the last three columns represent the instantaneous angular velocity, denoted by $\omega_c = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$. To track multiple features, the interaction matrices of different features can be stacked on top of one another to produce a matrix of the form $L_x = \begin{bmatrix} L_{x1} & L_{x2} & \cdots & L_{xn} \end{bmatrix}^T \in \mathbb{R}^{2k\times 6}$, where k is the number of features being tracked.

By using the Interaction Matrix (4.14), IBVS algorithm can generate control signals to drive the camera motion in order to achieve the desired image motion. This allows IBVS to control the motion of the camera based solely on the image information, without the need for explicit 3D information.

4.3.3 Target Tracking using the Robotic Rat's Torso

The interaction matrix, as derived previously, allows us to compute the required camera motions to track the target feature points and minimize the error. However, to execute these motions, the robotic rat's Jacobian has to be utilized. Specifically, the Jacobian matrix can be leveraged to determine the joint velocities required to achieve the desired camera motions computed from the interaction matrix. By combining the interaction matrix with the Jacobian, the robotic rat's camera situated at the head of the robot can be moved according to the computed motions, thereby facilitating accurate tracking of the target feature points and minimizing error.

The robot Jacobian, $\mathbf{J} \in \mathbb{R}^{6\times 7}$, is a fundamental tool in the analysis and control of robotic manipulators. It is a matrix that maps the velocities of the joints of the manipulator to the velocities of the end-effector. In our case, the end-effector is the robotic rat's left stereo camera. Each row in the Jacobian corresponds to the components of the spatial velocity vector $\mathbf{v_c} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ respectively, and each column corresponds to the joint angles $\mathbf{q}$ of the robot. The Jacobian can be computed using the forward kinematics of the manipulator, which relates the joint angles to the position and orientation of the end-effector. It can also be computed numerically using finite differences

or other numerical methods. In the case of the robotic rat, which has 7 joints, The Jacobian can expressed as shown in Eq. (4.16):

$$\mathbf{v_c} = \mathbf{J}\dot{\mathbf{q}} \tag{4.15}$$

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} & \frac{\partial x}{\partial q_4} & \frac{\partial x}{\partial q_5} & \frac{\partial x}{\partial q_6} & \frac{\partial x}{\partial q_7} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} & \frac{\partial y}{\partial q_4} & \frac{\partial y}{\partial q_5} & \frac{\partial y}{\partial q_6} & \frac{\partial y}{\partial q_7} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} & \frac{\partial z}{\partial q_4} & \frac{\partial z}{\partial q_5} & \frac{\partial z}{\partial q_6} & \frac{\partial z}{\partial q_7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{4.16}$$

where $q_1, q_2, ..., q_n$ are the joint variables of the manipulator, x, y, z are the position of the camera in Cartesian coordinates.

As previously discussed, the Jacobian matrix relates the spatial velocity of a robot's end-effector to the joint velocities of the robot. This relationship enables any constraints on the robot's motion to be applied to the tracking task through the Jacobian. As outlined in Chapter 2, a joint coupling constraint is required between the 3rd and 4th joints. This constraint is achieved by setting their respective columns to be equal and their value is approximated using the average of both columns. In addition, when tracking an object while moving on the ground, the robotic rat is designed to move in a manner similar to a real rat, which means that it never stands up. To achieve this, all pitch joints except for the head's pitch are locked, which is accomplished by setting their corresponding columns to zero. These constraints ensure that the robotic rat exhibits realistic rat-like motion during the tracking process. Therefore, the final form of the Jacobian matrix for the robotic rat is represented by Eq. (4.17).

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} 0 & 0 & \frac{\partial x}{\partial q_{34}} & \frac{\partial x}{\partial q_{34}} & 0 & \frac{\partial x}{\partial q_6} & \frac{\partial x}{\partial q_7} \\ 0 & 0 & \frac{\partial y}{\partial q_{34}} & \frac{\partial y}{\partial q_{34}} & 0 & \frac{\partial y}{\partial q_6} & \frac{\partial y}{\partial q_7} \\ 0 & 0 & \frac{\partial z}{\partial q_{34}} & \frac{\partial z}{\partial q_{34}} & 0 & \frac{\partial z}{\partial q_6} & \frac{\partial z}{\partial q_7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{4.17}$$

Where the 3rd and 4th column are set equal to the average of both columns, ensuring that they are equal. By substituting Eq. (4.15) in Eq. (4.13) we obtain:

$$\dot{\mathbf{p}} = \mathbf{J_e}\dot{\mathbf{q}} \tag{4.18}$$

$$\mathbf{J_e} = \mathbf{L_x}\mathbf{J}, \quad \mathbf{J_e} \in \mathbb{R}^{2k\times 7} \tag{4.19}$$

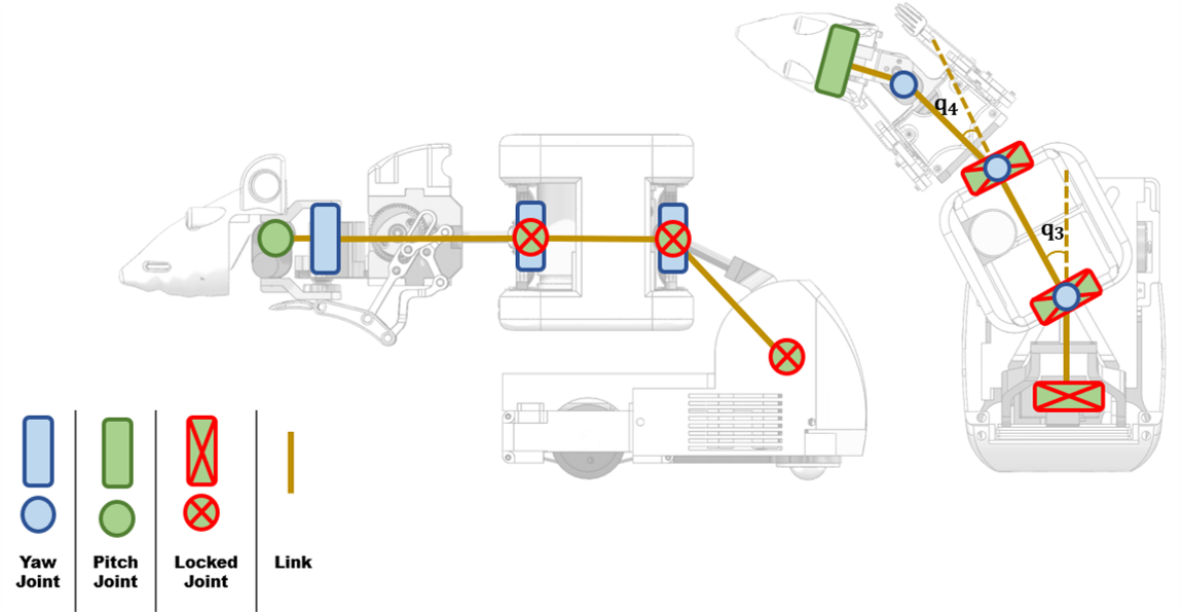


Figure 4.7 Effect of Jacobian Constraints on the Robotic Rat's Joints

Equation Eq. (4.19) is referred to as the Task Jacobian, which is a vital mathematical tool in Image-based Visual Servoing (IBVS). The Task Jacobian describes the relationship between the change in image features and the motion of the robot's joints. Specifically, it

indicates how a small change in the joint configuration can affect the image features. By using the Task Jacobian, the error in image features can be converted into an error in joint angles, which can then be used to update the joint angles. The Task Jacobian plays a crucial role in the success of the visual servoing process since it guides the robot to move its joints in the appropriate direction to minimize the error between the image features and the desired features. By utilizing the Task Jacobian, the feature velocities in the camera frame can be directly related to the motion of the robotic rat's joint velocities.

The Penrose-Moore Pseudo-inverse of a matrix is a generalization of the inverse of a matrix that can be used when the matrix is not invertible or when the system of linear equations represented by the matrix has multiple solutions.The pseudoinverse of a matrix can be computed using the singular value decomposition (SVD) of the matrix. The SVD of a matrix A is given by $A = U\Sigma V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix containing the singular values of A. If A is invertible, then its pseudoinverse is given by $A^+ = (A^T A)^{-1} A^T$. If A is not invertible, then its pseudoinverse is given by $A^+ = V\Sigma^+ U^T$, where Σ^+ is a diagonal matrix containing the reciprocal of the nonzero singular values of A, and zero otherwise. By taking the Penrose-Moore Pseudo-inverse of the Task Jacobian the joint velocities can be computed using the image feature velocity.

$$\dot{\mathbf{q}} = \mathbf{J_e}^{+}\dot{\mathbf{p}} \tag{4.20}$$

The mathematical expression that generates a control signal for a robot's end-effector to track a desired object in an image is known as the control law of an Image-based Visual Servoing (IBVS) algorithm. This expression is essential to the operation of the algorithm and is computed based on the error signal, which is the difference between the current position of the image feature and the desired position. A proportional velocity controller can be derived from the visual servoing tracking task by substituting the feature point in Eq. (4.20) with the feature error described in Eq. (4.7). To this equation, a proportional term λ is added, which is a diagonal matrix that scales the error signal. The resulting control law is expressed in Eq. (4.21).

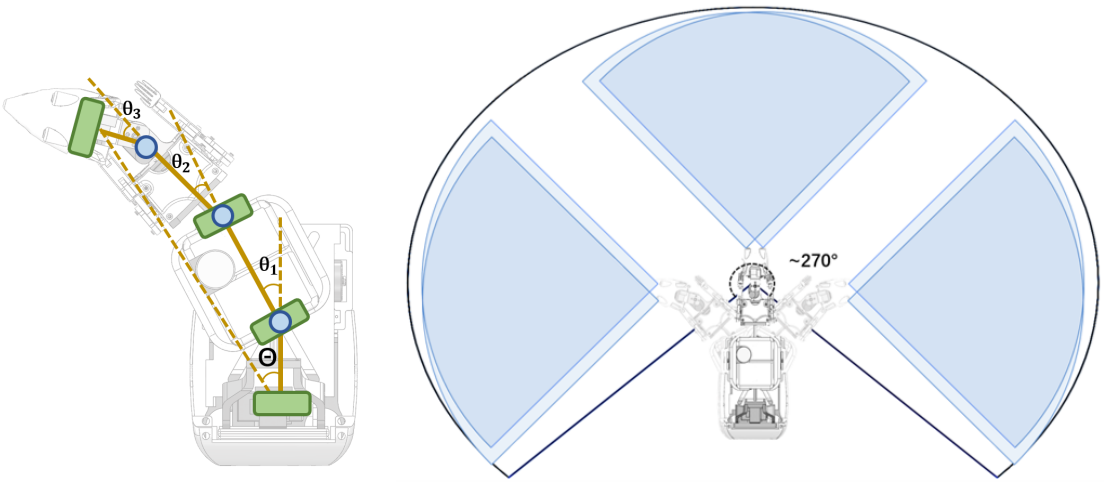
$$\dot{\mathbf{q}} = -\lambda \mathbf{J_e}^{+} \mathbf{e} \tag{4.21}$$

This concludes the derivation and design of the IBVS based rat-tracking algorithm for the robotics rat's torso.

4.3.4 Control of the Mobile Base for Tracking

In this section, the tracking algorithm for the mobile base that enables it to coordinate with the robotic rat's torso is presented. While it is possible to implement a target tracking algorithm using only the robotic rat's mobile base, by locking the torso's movement, this approach has several disadvantages when compared to using both the torso and the base for tracking. When the torso is locked, the robotic rat can follow its target by using its wheels to spin in place towards the target's direction and minimize the error. Although this approach is straightforward to implement, it lacks the flexibility that the torso provides. The torso can move to adjust its viewpoint, enabling the camera to capture objects from different angles. This flexibility is particularly useful in situations where the mobile base cannot move, such as when the target is behind an obstacle. In addition to its flexibility, the torso is also more energy-efficient than relying solely on the mobile base. Since only a part of the robot has to move, less energy is consumed during tracking. Hence, using both the torso and the mobile base for tracking provides a more efficient and flexible solution for target tracking compared to using the mobile base alone.

An additional reason to pursue a combined approach is to achieve rat-like motions, as the laboratory rat typically prioritizes the use of its head to scan the environment during the search and tracking of objects, rather than moving its entire body. Hence, coordinating the robotic torso and the mobile base is more in line with the natural motion patterns of the laboratory rat. Therefore, when designing the control law for the mobile base of the robotic rat, the objective is not solely to track the target laboratory rat directly, but rather to coordinate with the torso movement and enable it to track the laboratory rat while maintaining realistic rat-like motions. This approach ensures that the robotic rat displays natural-looking behavior during the search and tracking of the target object. Coordination with the robotic rat's torso is implemented in two primary aspects. The first is to expand the field of vision of the robotic rat when it reaches



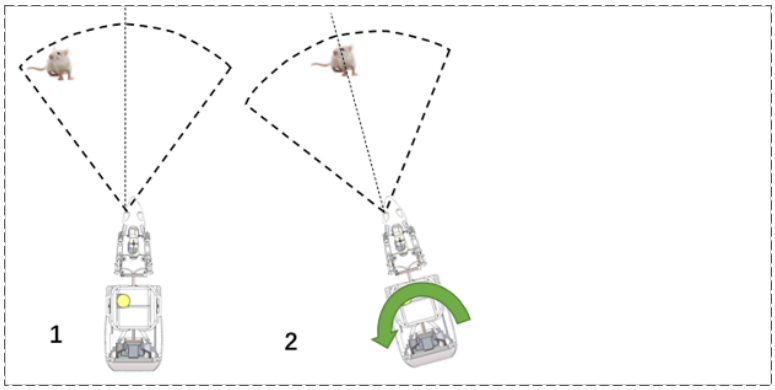
(a) Total Yaw (b) Tracking Range

Figure 4.8 Full Tracking Range of the Robotic Rat's Torso

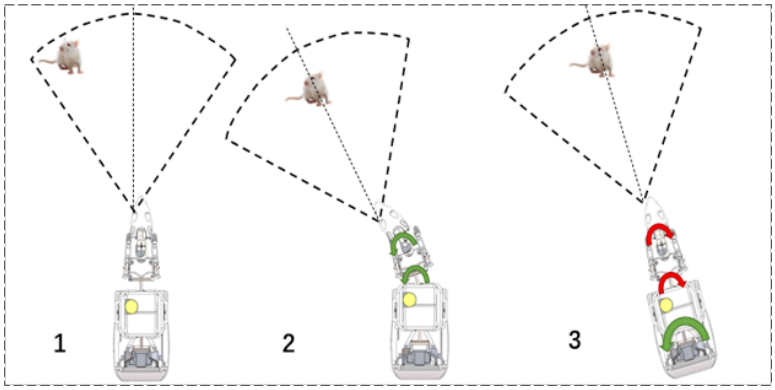
its limit. The second aspect is to rectify the robotic rat's pose, ensuring that the robotic torso returns to its initial configuration, as illustrated in Figure 4.9(b). By coordinating the robotic torso and mobile base, the robotic rat exhibits more natural and realistic behavior, similar to the laboratory rat.

The implementation of both functions involves utilizing a single PID controller[74], where the control error is defined as the total yaw of the robotic rat's torso. The total yaw is calculated as the sum of all the yaw joints, as depicted in Figure 4.8(a). By setting the control error as the total yaw of the robotic rat's torso, the mobile base can align with the torso during the tracking process while simultaneously exhibiting rat-like motion. Furthermore, a tolerance interval is employed to ensure the stability of the system by eliminating oscillations that occur as the error approaches a very small value. This approach enables the system to maintain stability and produce accurate results.

In addition to coordinating with the torso tracking, the mobile base is also required to approach the target laboratory rat during behavioral interactions. The distance between the robotic rat and the laboratory rat is a critical factor to consider in such experiments, as it can affect the behavior of both animals and influence the success of the experiment. The optimal distance between the robotic rat and the laboratory rat is dependent on the specific goals of the experiment and the behavior of the laboratory rat. To control the distance between the



(a) Only Base Tracking



(b) Full Body Tracking

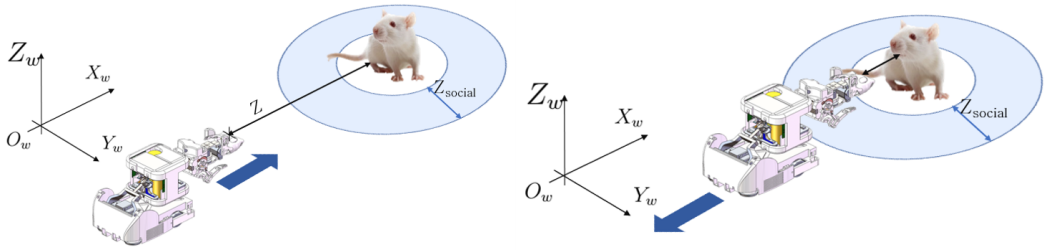
Figure 4.9 Comparison between Base Tracking and Full Body Tracking

robotic rat and the laboratory rat, another PID controller is implemented. The control target is set as the depth from the stereo camera reading Z, allowing the robotic rat to approach the required social distance Z_{social} necessary for performing behavioral interactions. This approach ensures that the robotic rat can maintain a safe and appropriate distance from the laboratory rat, while still facilitating meaningful behavioral interactions.

Table 4.2 PID Parameters of Mobile Base Tracking

Parameter	**Proportional (P)**	**Integral (I)**	**Differential (D)**
Spine Tracking	0.5	0.0	0.01
Distance Tracking	0.2	0.001	0.05

To ensure the effectiveness of the control laws for both rat-tracking and approaching, the PID parameters are tuned using the trial-and-error method. This tuning process involves measuring the response of the system to a step input and adjusting the PID parameters until



(a) Approach (b) Withdraw

Figure 4.10 Distance control for approaching and withdrawing from laboratory rat

the desired response is achieved. By tuning the PID parameters in this way, the control laws can be optimized to achieve accurate and reliable control of the robotic rat's movements. This process of iterative tuning and evaluation is an essential step in developing effective control systems for robotic applications, allowing for the fine-tuning of control parameters to optimize system performance.

4.4 Control Policy for Behavioral Interaction

For the purpose of conducting behavioral interaction experiments, a control policy is designed to guide the robotic rat during the experiment. The policy is divided into four main components, each of which handles a specific aspect of the experiment. These components are: "Search," "Rat-Tracking," "Approach," and "Behavioral Interaction." Together, these components allow the robotic rat to interact with the target laboratory rat in a controlled environment. A detailed description of each component is provided below:

1. **Search**: The "Search" Phase is designed with the purpose of locating the laboratory rat in the surrounding experimentation environment. During "Search" the robotic rat starts scanning its surroundings using its camera by spinning in place until the laboratory rat is detected. The robotic rat also enters the "Search" whenever it suddenly loses track of the laboratory rat. If it has previously detected the laboratory rat, the robotic rat keeps record of the laboratory rat's relative position to itself, such that the search direction is towards the last seen direction of the laboratory rat, thus minimizing the search time.

2. **Rat-Tracking**: The "Rat-Tracking" Phase is concerned with keeping the target

laboratory rat within the robotic rat's field of view. This is accomplished using the Image Based Visual Servoing algorithm presented above. During the "Rat-Tracking" Phase, the robotic rat uses feedback from the stereo camera and computes the necessary motion to ensure that the laboratory rat is approximately situated in the center of the image. If the robotic rat loses vision of the laboratory rat, it switches back to the "Search" phase, using the last detected image of the laboratory rat as a reference for the search direction.

3. **Approach**: The "Approach" Phase is responsible for closing the distance between the robotic rat and the laboratory rat during the experiments. The robotic rat can enter the "Approach" Phase during the "Rat-Tracking" Phase, as long as the laboratory rat is within the robotic rat's field of view. The robotic rat uses the depth reading from the stereo camera to determine the current distance Z between the robotic rat and the laboratory rat. The PID controller mentioned above is then used to drive the robotic rat until the current distance Z equals the reference interaction distance Z_{social} as determined by the researchers.

4. **Behavioral Interaction**: The "Behavioral Interaction" Phase begins when the robotic rat is within the predetermined social distance Z_{social} from the laboratory rat. Once the robotic rat is within the desired distance, behavioral interaction experimentation is conducted by having the robotic rat perform one of the generated Motion Primitive trajectories as introduced in the 3rd Chapter. After executing the desired trajectories, the robotic rat reverses and starts a new cycle.

The above mentioned components are organized into two control policies for experimentation.

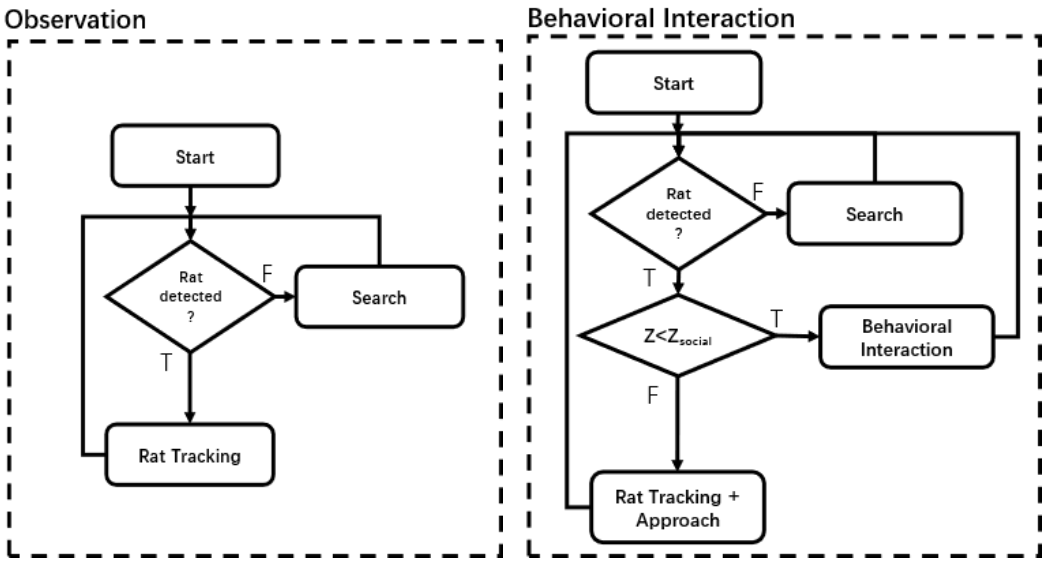
4.4.1 Observation

The Observation policy is mainly used for keeping track of the laboratory rat's location within the experimentation environment. This policy is useful for testing and evaluating the effectiveness of the Rat-Tracking algorithm. It is also valuable for collecting visual data of the laboratory rat from first person view, which could be used in behavior analysis, expanding

the rat detection algorithm etc. The policy begins with the ”Search” phase until the laboratory rat is detected. Then, coordinated visual tracking of the torso and base comes in effect during the ”Rat-Tracking” phase. As long as the robotic rat can maintain visual of the laboratory rat, the robotic rat stays in the ”Rat-Tracking” phase.

4.4.2 Behavioral Interaction

The Behavioral Interaction policy is used to conduct mock robot-animal behavioral experiments between the robotic-rat and the laboratory rat. It expands on the observation policy by entering the ”Approach” phase at the same time as the ”Rat-Tracking” phase if the robotic rat is still not within the behavioral distance from the laboratory rat. Once the robotic rat reaches the social distance of the laboratory rat, it enters the ”Behavioral Interaction” phase and performs the desired behavior’s motion. When the ”Behavioral Interaction” phase is completed. The robotic rat back tracks and begins the ”Search” again. Both policies are described succinctly in Figure 4.11



(a) laboratory Rat Observation Process (b) Behavioral Interaction Process

Figure 4.11 Control Policy for the Robotic Rat

4.5 Summary

In this chapter, the perception system of the robotic rat used for target detection and depth estimation was presented. Additionally, an Image-Based Visual Servoing algorithm is

introduced to enable the robotic rat to track the laboratory rat in real-time as it moves through the environment. The complete derivation of the Rat-Tracking algorithm and coordination between the torso and base was presented. Finally, a Control Policy is presented, which outlines the sequence of steps the robotic rat takes to conduct demonstration behavioral interaction experiments. In the next chapter, a series of experiments are conducted to validate the proposed methods presented in this work.

Chapter 5 Simulation and Experiments

5.1 Overview

In the second chapter, the algorithm for mapping from the laboratory rat's motion capture data to the robotic rat's joint angles was introduced. Subsequently, in the third chapter, the ProMP probabilistic motion primitive framework was discussed and was utilized to produce a motion primitive library for the robotic rat based on the behaviors extracted from the motion capture data. Finally, in the fourth chapter, a real-time rat-tracking algorithm based on Visual Servoing was introduced to allow the robotic rat to maintain visual of the target laboratory rat. In this chapter, the process of validating our methods and conducting experiments is introduced.

5.2 Rat-to-robot Mapping

The optimization problem presented in the 2nd Chapter is implemented using the IPOPT library in C++. Subsequently, the entire classified motion capture dataset is processed frame by frame, inputting the 7 points of each frame and outputting the joint angles corresponding to that frame.

Table 5.1 Motion Capture Dataset

Behavior	Files No.	Avg. Frames No.
MO	18	~ 470
PIN	04	~ 1100
POU	08	~ 850
SNC	54	~ 1180

5.2.1 MO

The MO behavior dataset consisted of 18 files. MO is the behavior that rat exhibits when moving in the environment. It is characterized by the low position of the head for sniffing the environment. The results of the mapping for MO is presented in Figure 5.1. The error of the mapped dataset is shown Figure 5.2, where each graph represents a single file in the dataset, the x-axis of the graphs is the frame number whereas the y-axis represents the mapping error

corresponding to that frame. The final error of each file in the MO dataset is shown in Figure 5.2 . The mean error of the entire MO dataset is $0.701mm$.

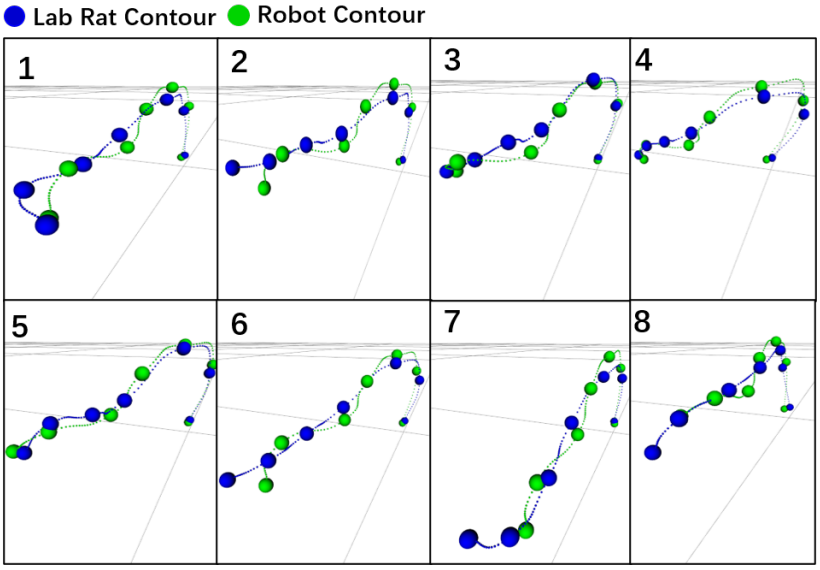


Figure 5.1 Frames Extracted from MO Mapping Results

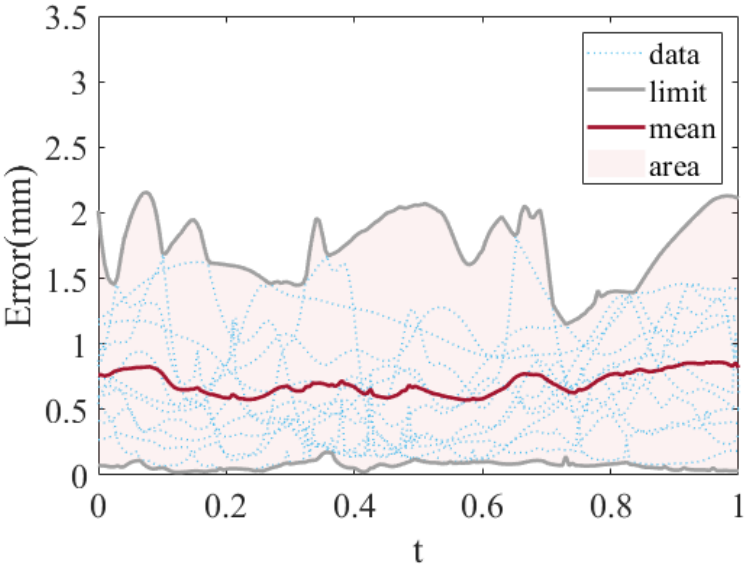


Figure 5.2 Error Distribution of MO Dataset Mapping Results

5.2.2 PIN

The PIN behavior dataset consisted of 4 files. The PIN behavior is defined as pinning the other rat. The results of the mapping for PIN is presented in Figure 5.3. The final error of each file in the PIN dataset is shown in Figure 5.4. The mean error for the entire PIN dataset is $0.509mm$.

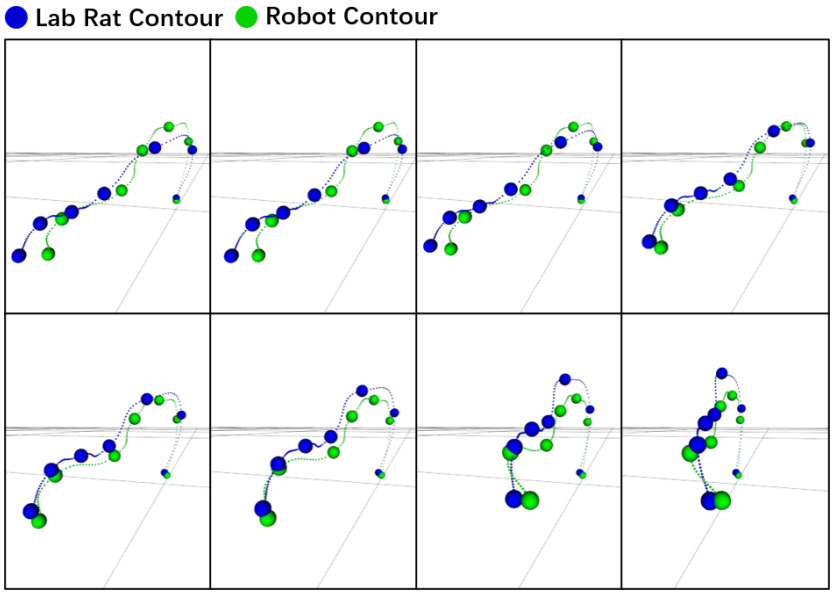


Figure 5.3 Frames Extracted from PIN Mapping Results

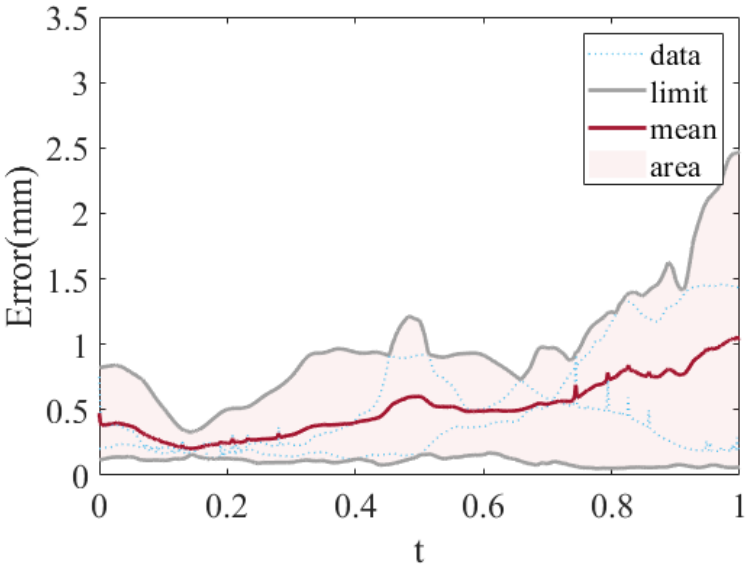


Figure 5.4 Error Distribution of PIN Dataset Mapping Results

5.2.3 POU

The POU behavior dataset consisted of 08 files.The POU behavior is defined as pouncing on the other rat. For this behavior, the rat usually raises its waist in order to lay on top of the other rat. The results of the mapping for POU is presented in Figure 5.5. The final error of each file in the POU dataset is shown in Figure 5.6. The mean error for the entire POU dataset is $0.393mm$.

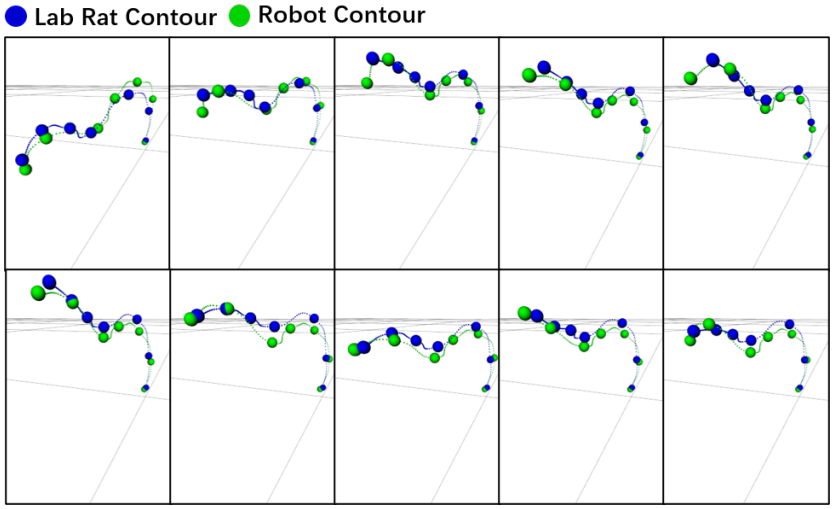


Figure 5.5 Frames Extracted from POU Mapping Results

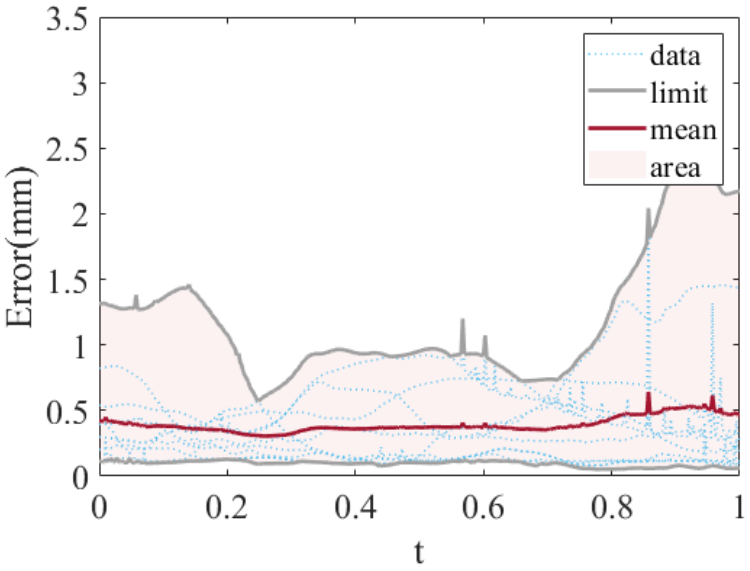


Figure 5.6 Error Distribution of POU Dataset Mapping Results

5.2.4 SNC

The SNC behavior dataset consisted of 16 files. The SNC behavior corresponds to Social Nose Contact, where most of the motion is the head joints. The results of the mapping for SNC is presented in Figure 5.7. The final error of each file in the SNC dataset is shown in Figure 5.8. The mean error for the entire SNC dataset is $0.392mm$.

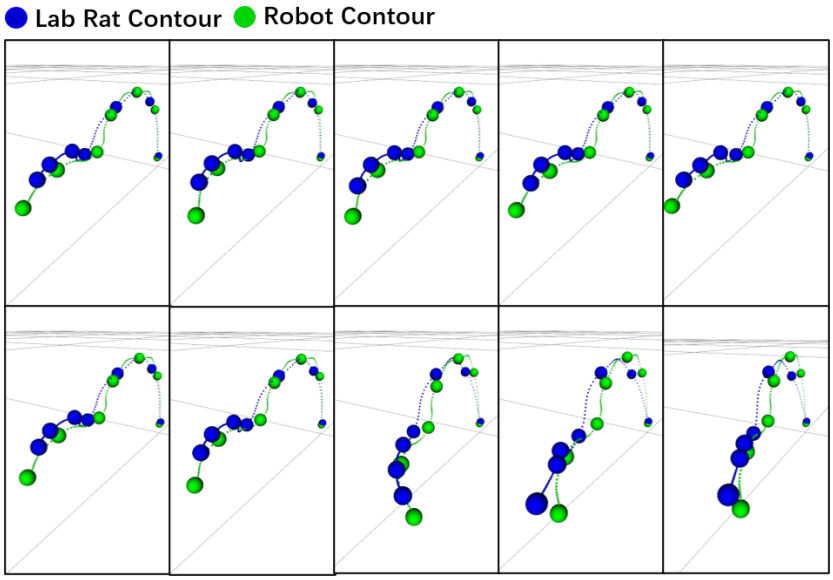


Figure 5.7 Frames Extracted from SNC Mapping Results

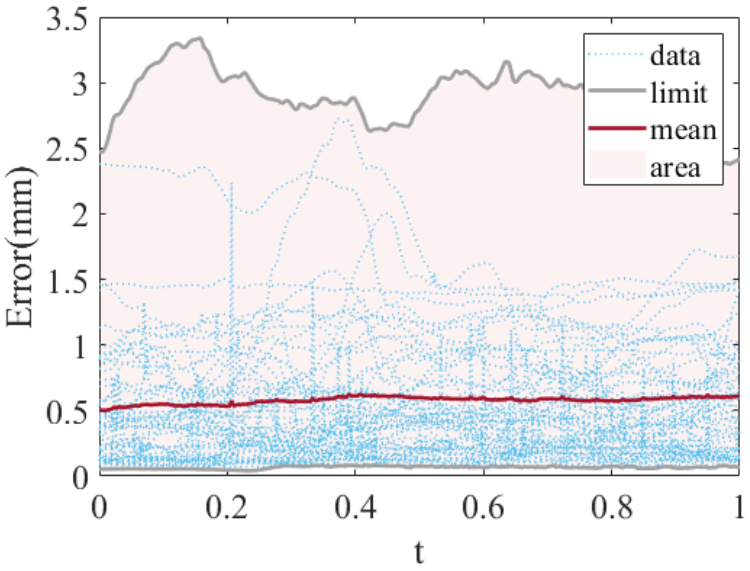
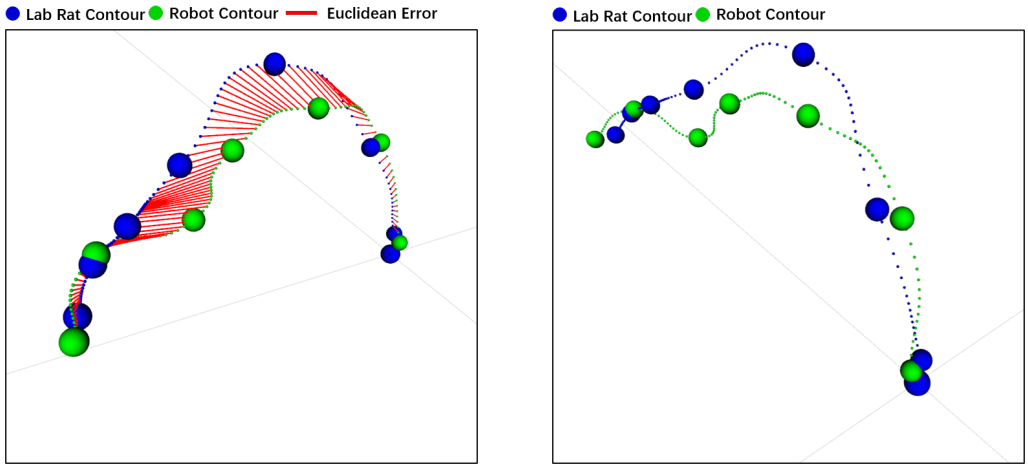


Figure 5.8 Error Distribution of SNC Dataset Mapping Results

5.2.5 Summary

The figures above presented the final error of the objective function for the different motion behavior datasets. By analyzing the final objective function error shown in those figures, we find that for the vast majority of the files in all datasets, the mapping algorithm manages to achieve a mean error of less than $0.5mm$, which is a sufficient amount to ensure likeness in the shapes of the outlines. Part of the reason the error cannot be decreased further in some cases is due to the difference in morphology between the robotic rat and the laboratory rat; the laboratory rat's flexibility allows it to move its joints in ways that the robotic rat cannot, creating a relatively large error even thought the outlines of both the laboratory rat and robotic rat are very similar in shape as shown in Figure 5.9. This shows that the algorithm will produce the best possible match of the laboratory rat's shape under the given circumstances, which is sufficient for the purpose of mimicking the laboratory rat's motions. It can also be observed that the error for certain behaviors is consistent across multiple files, which indicates that the current choice of the algorithm parameters might benefit one behavior over the other.



(a) (b)

Figure 5.9 Example of High Error Result

Overall, the Rat-to-Robot motion mapping algorithm can successfully, and to within an acceptable degree of accuracy, compute the joint angles of the robotic rat corresponding to a given motion by the laboratory rat. Using the mapping algorithm a robotic rat behavioral

motion dataset is produced from the laboratory rat's motion capture data.

5.3 Probabilistic Motion Primitive Trajectories

In this section, the joint trajectories that were generated using the Rat-to-Robot motion mapping algorithm are used to produce ProMP Models of each behavior. For each of the behaviors, the joint trajectory dataset was split into different demonstrations for the algorithm to learn from, as described in the 3rd Chapter. For the purpose of performing behavioral interaction, all of the joint trajectories have been blended with the zero primitive which allows the joints to start and end at a neutral state when conducting the experiments. The joint trajectories corresponding to the different behaviors are all shown in Figure 5.10, note that since the 2nd and 5th joints as well as the 3rd and 4th joints are coupled respectively, hence only one of each pair is shown in the Figure. The thick blue curve in each graph represents the mean of the joint's trajectory and it also represents the general tendency or shape that the trajectory usually takes when performing the corresponding behavior.

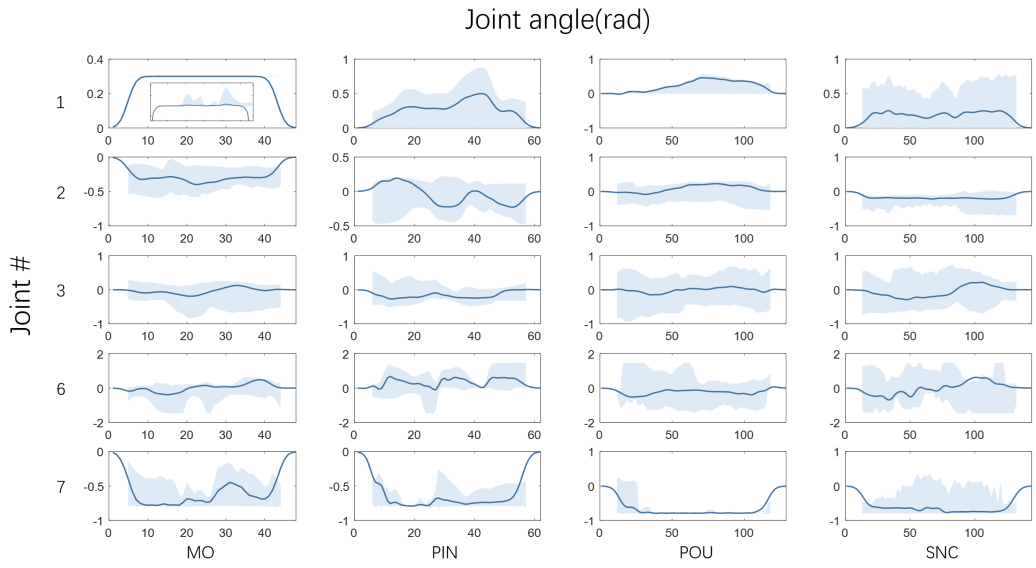


Figure 5.10 Motion Trajectories Generated using ProMP

1. **MO**: The motion trajectories generated by the MO model clearly match the originally described behavior definition and observations of the laboratory rat's behavior during

this motion. While performing the MO behavior, the laboratory rat does not move its lower back pitch joint at all, this is evident by the very low variance in the 1st joint's trajectory. The 2nd/5th joints, which correspond to the waist's pitch, are slightly in the negative. This is accurate as the laboratory rat usually keeps its body on the ground while moving. The 3rd/4th and the 6th joint represent the waist's and head's yaw respectively. It can be observed that all the yaw joints in the MO behavior vary significantly in its values, which is due to the fact that the laboratory rat moves left and right arbitrarily during this behavior. The 7th joint which corresponds to the head's pitch is mainly in the negative, i.e lowering its head. This matches the description of the MO behavior, since the laboratory rat often uses his head to sniff the environment while moving, meaning its head is also usually lowered.

2. **PIN**: The motion trajectories of the PIN model are one of the most dynamic in this motion set. This reflects the dynamic nature of the PIN behavior, which is defined as a laboratory rat pinning another rat on its back. The 1st joint is entirely in the positive, this matches the behavior's description since the rat needs to raise its waist in order to mount the other rat's back. The 2nd/5th pitch joints manifest a rhythmic motion of climb and fall, the climb allows the robot to rise above the other rat, whereas the fall allows it to pin the rat underneath it. The 3rd/4th joints as well as the 6th joint are yaw joints that are inconsequential to this behavior, thus they mainly vary arbitrarily to encompass situations when the rat is not mounting the other rat in a straight line. The 7th head pitch joint is deeply in the negative, this is expected since the rat would have to lower its head while mounting another rat to keep it parallel to the horizontal plane and rest it on top of the other rat.

3. **POU**: The POU behavior corresponds to "pouncing" which is a less dynamic variant of the PIN behavior where instead of laying on the back of the other rat, the rat stands up infront of the other rat and lays on its head. Thus the joint trajectories of the POU ProMP model closely resemble that of the PIN model, albeit with a slight decrease in intensity. The greatest difference between the two trajectories is in the 6th yaw joint, which is inconsequential to the characteristics of this behavior.

4. **SNC**: The motion trajectories of the SNC behavior are very subtle compared to the other behaviors. Since the SNC corresponds to Social Nose Contanct, the joints that are mainly utilized are the 6th and 7th head joints. The 6th joint maintains a left to right oscillation throughout the trajectory which is the most prominent characteristic of SNC. The 7th joint follows a downward tendency to allow the rat's noses to touch. All the remaining joints exhibit stable trajectories that do not contribute much to the behavior.

In order to quantitatively evaluate the similarity of the motions produced by the ProMP model to those of the laboratory rat under the same behavioral patterns we utilize cosine similarity of the rat and the robot's vectors, the detailed evaluation model is expressed as Eq. (5.1)[28].

$$S(t) = [1 - w(t)] \cos \left\langle \overrightarrow{P_t^{rat}}, \overrightarrow{P_t^{bot}} \right\rangle + w(t) \left(1 - \frac{\sum_{i=0}^{7} \sum_{t=0}^{T} (\theta_i^{rat}(t) - \theta_i^{bot}(t))^2}{\sum_{i=0}^{7} \sum_{t=0}^{T} \theta_i^{rat}(t)^2} \right) \tag{5.1}$$

$$\text{s.t. } w(t) \in [0, 1],$$

$$t \in [0, T]$$

where $\cos \left\langle \overrightarrow{P_t^{rat}}, \overrightarrow{P_t^{bot}} \right\rangle$ denotes the cosine distance of the vector between the robot rat and the experimental rat at time t. The latter part denotes the difference in the angle change of each joint between the robot rat and the laboratory rat from time 0 to time t. T is the total time of movement. The second half represents the difference in the angular change of each joint from 0 to t. T is the total time of movement, and the closer the value of S(t) to 1, the higher the similarity between the two. The evaluation model uses the weight transfer function w(t) to combine the transient and dynamic indicators in the motion process, while avoiding the problem of inaccurate evaluation caused by the small amount of dynamic data at the initial moment. w(t) function is chosen as a 5th order polynomial form.

This function allows S(t) to be based primarily on transient metrics at the beginning of the campaign, gradually shifting the weight to more accurate dynamic data metrics as time

Table 5.2 Similarity Analysis of Motion Trajectories

Behavior	Similarity
MO	0.975
PIN	0.963
POU	0.985
SNC	0.983
Avg.	0.976

progresses.

$$w(t) = \frac{10}{T^3}t^3 - \frac{15}{T^4}t^4 + \frac{6}{T^5}t^5 \tag{5.2}$$

By analyzing the results of the similarity evaluation for each set of behaviors, we find that all behavioral motions managed to achieve a very high similarity score. The results of the evaluation is list in Table 5.2.

In conclusion, the motion trajectories generated by the ProMP models succeeded in capturing the essence of each behaviors motion characteristics as well as accounting for the variance that inherently exists in specific demonstrations of each behavior. This has been validated using the similarity evaluation, where the similarity of all the learned behaviors is above 96%. This validates the use of these trajectories in behavioral interaction experimentation between the robotic rat and the laboratory rat.

5.4 Rat-Tracking

5.4.1 Robotic Platform

In order to be able to detect the laboratory rat in real time, real-time image processing of the acquired images is required. With the development of computing vision technology and the development of convolutional neural networks, high-precision recognition algorithms often require high-performance GPUs to perform real-time processing. In addition, the small size of the mouse does not allow it to carry a MCU that can meet the computational power necessary to achieve real-time image processing. Therefore, the real-time binocular images are sent to a server for processing via wireless transmission, and then the feedback The feedback motion control commands are then transmitted back to the robotic rat via

wireless transmission.

5.4.2 Simulation & Digital Twin

In order to verify the effectiveness of the real-time rat-tracking algorithm, a simulation of the tracking task is conducted using RVIZ platform. A URDF[75] of the robotic rat is utilized as the model for testing the algorithm in simulations. In the simulations, a point in 3-D space is used as the target of tracking as a replacement for the position of the laboratory rat. The point moves according to a pre-determined arc centered at the origin of the simulation environment. The simulation was used mainly to test the validity of the Spine Tracking part of the Visual Servoing algorithm and to perform a preliminary tuning of the control law's parameters. The robotic rat manages to track the position of the floating point while maintaining a maximum error of $0.025mm$. The results of the simulation is shown in Figure 5.11.

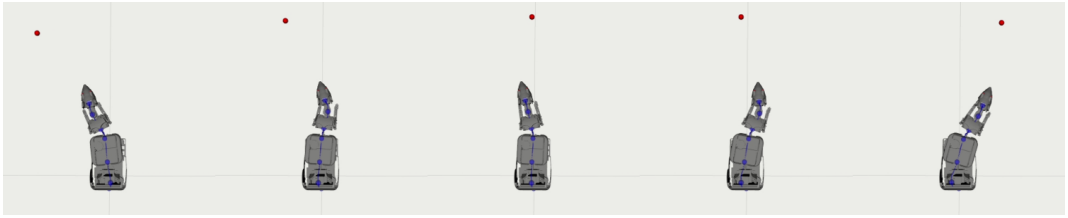
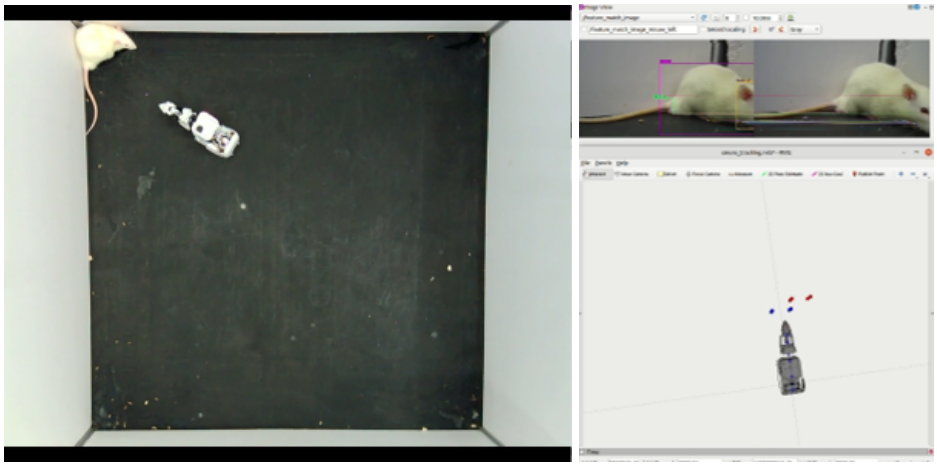
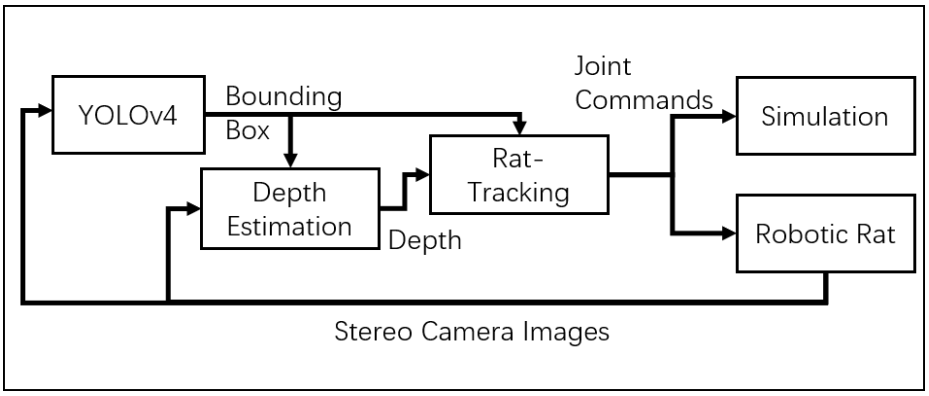


Figure 5.11 Target Tracking Simulation Result

The simulation platform is also set up such that it is possible to produce a digital twin of the Rat-Tracking process by running the simulation platform alongside the robotic rat platform. This is achieved by passing the robotic rat's sensory inputs to the simulation and visualizing the current position of the laboratory rat relative to the image frame of the robotic rat's camera. The tracking control commands are also passed to both the robotic rat's joints and the robot's model in the simulation platform, allowing both platforms to be synchronized and move in unison. The framework of the full system is shown in Figure 5.12



(a) Real Environment (b) Simulation on Server



(c) Software Framework

Figure 5.12 Digital Twin Framework

5.4.3 Tracking Experiments

In order to verify the validity of the Rat-Tracking algorithm, tracking experiments were conducted, within the experimentation setup described in Figure 5.13.

First, the basic tracking capabilities of the robotic rat are tested. The ”Search” phase is mainly used to locate the laboratory rat at the beginning of the experiment or when it leaves the robotic rat’s FOV as shown in Figure 5.14(a), the direction of the searching the first time the robotic rat enters the ”Search” phase is fixed. While conducting behavioral interaction experimentation, the robotic rat will approach the laboratory rat while tracking it as shown in Figure 5.14(b).

In another experiment, the robustness of the tracking algorithm was tested. As shown in Figure 5.15, the robotic rat can maintain constant visual on an active laboratory rat for a

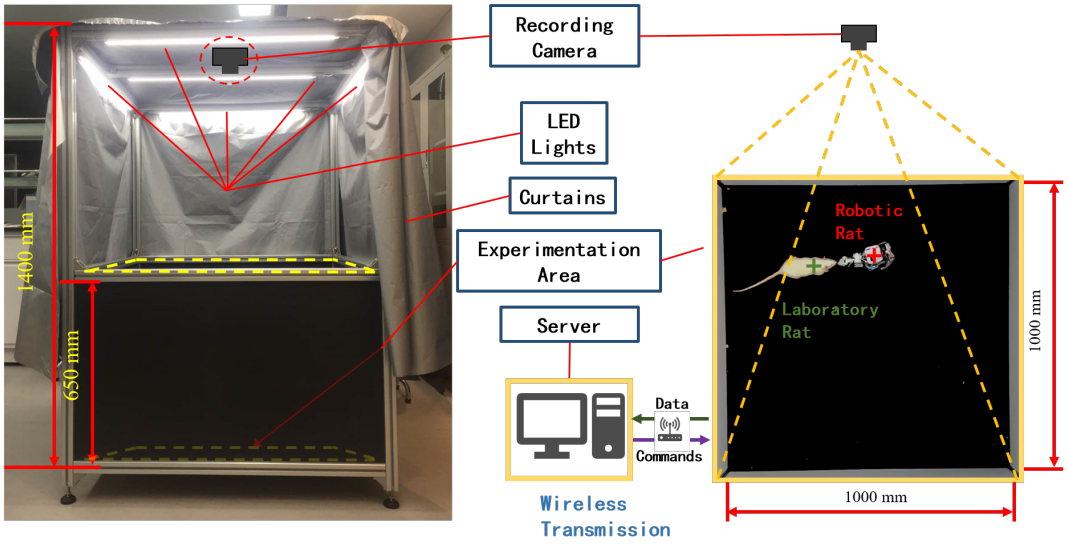
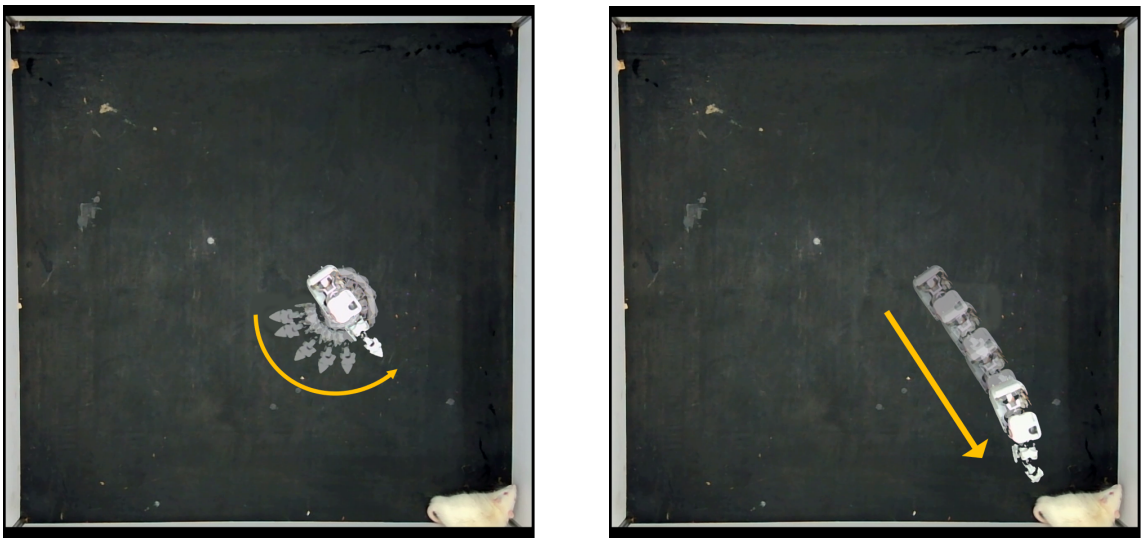


Figure 5.13 Setup for Conducting Experiments



(a) Searching

(b) Approaching

Figure 5.14 Basic Tracking Capabilities

significant amount of time. In the first few frames, the robotic rat twists its spine greatly to keep up with the rat's movements. Soon after, the base begins realigning with the torso as shown in Image c, this allows the spine to return to a neutral state while still maintaining visual on the rat, realizing coordination between torso and base and increasing the stability of tracking. During and after this process the robotic rat continues to track and maintain constant visual of the laboratory rat. During this experiment the robotic rat successfully

tracked an actively moving laboratory rat while maintaining constant visual for a total period of 55 seconds (12 seconds inactive, 43 seconds active) after which the laboratory rat became inactive and stopped moving.

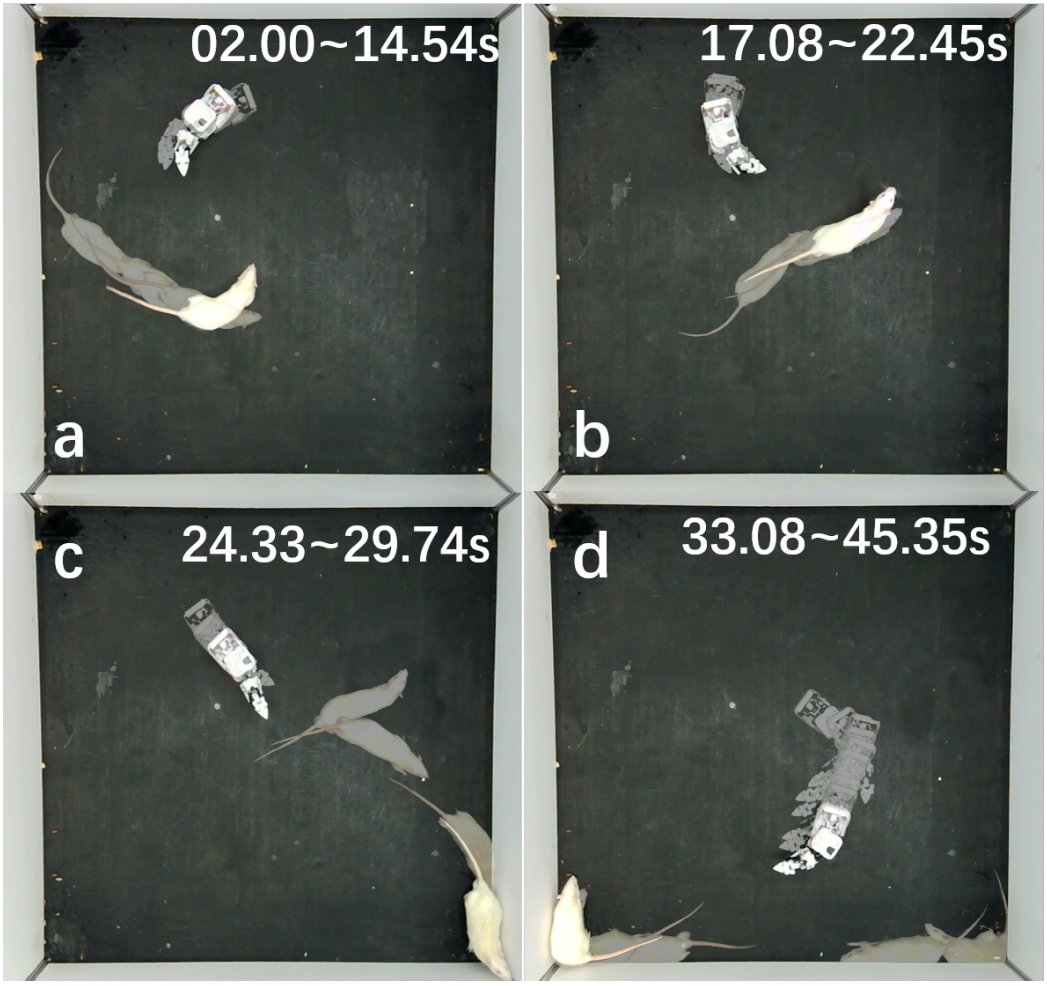


Figure 5.15 Torso & Base Coordinated Tracking

In order to increase the robustness of the tracking algorithm, a Directional Search is used to minimize the time consumed to re-locate the laboratory rat when it leaves the robotic rat's FOV; This is implemented by recording the last known position of the laboratory rat relative to the robotic rat's camera frame, the result is shown in Figure 5.16. By analyzing the experiments, we find that on average the Directional Search consumes around $2.1s$, with $1.7s$ being the least time consumed in re-locating the laboratory rat, effectively decreasing the time spent in this phase as compared with a fixed or random approach.

When compared to the previous One-Step Tracking algorithm, the proposed IBVS

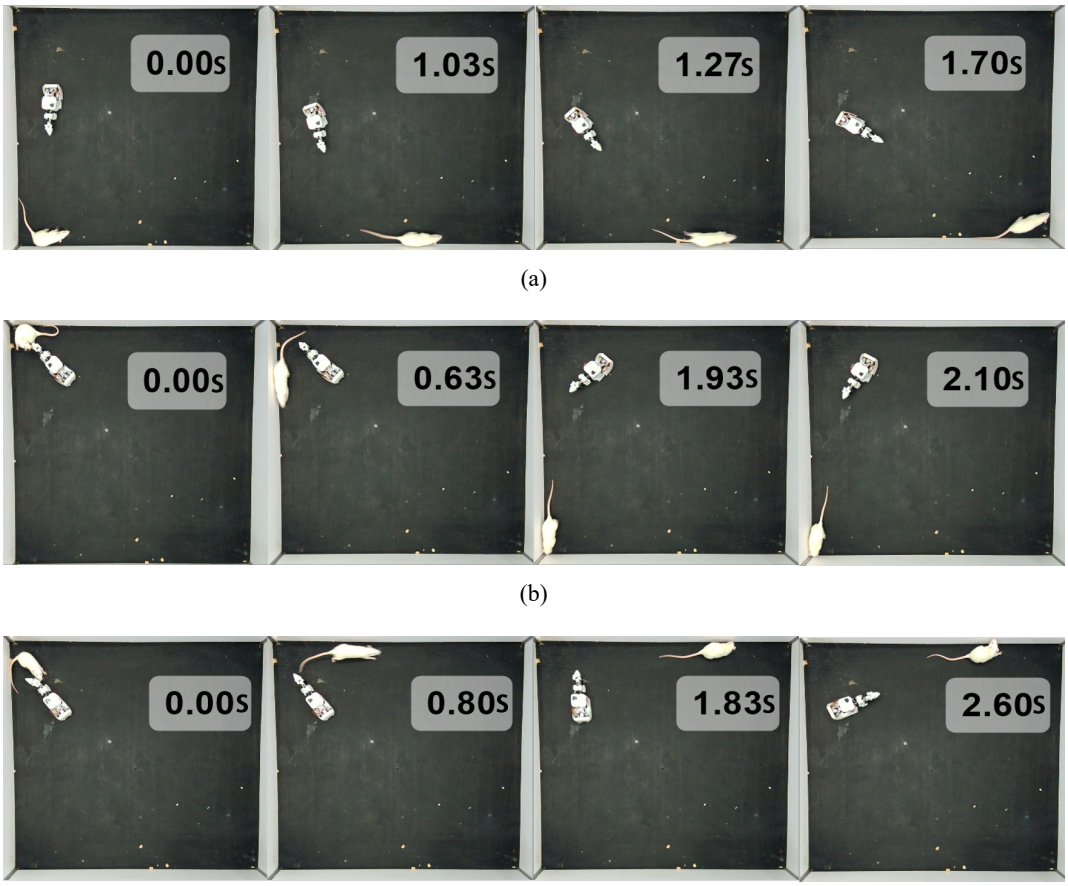


Figure 5.16 Directional Search Policy

tracking displays better stability and smoother tracking trajectories. Additionally, it realizes simultaneous pitch and yaw tracking of the rat's position.

In conclusion, the experiments presented above provide sufficient evidence of the validity of the Image Based Rat-Tracking algorithm presented in this work. Aside from its superior stability compared to the previous methods, it is also more robust to the laboratory rat's movement, can maintain visual of the rat for a longer period of time and can re-locate the rat rapidly when it leaves the robot's FOV.

5.5 Behavioral Interaction

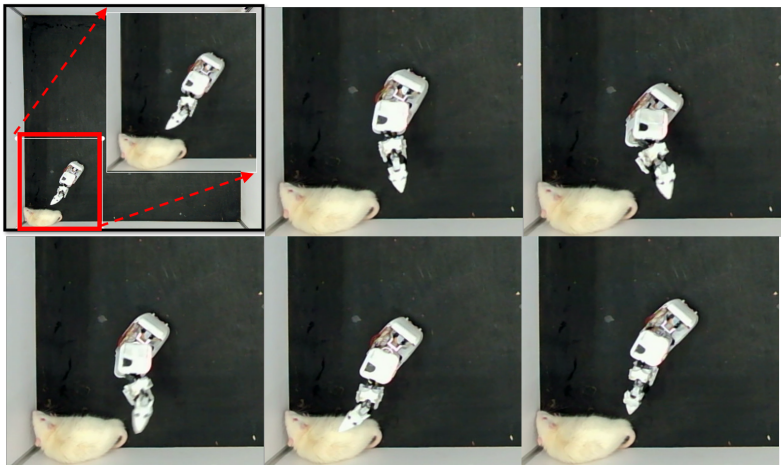
In this section, in order to provide a proof of concept for the motion planning framework, the control policy for behavioral interaction that was introduced in the 4th Chapter is employed alongside the motion trajectories of each behavior that were generated using ProMPs and the Rat-Tracking algorithm to conduct a mock robot-rat behavioral interaction experiment.

In start of the experiment the robotic rat enters the ”Search” phase to locate the laboratory rat, then once the rat is detected, the robotic rat enters the ”Rat-Tracking” and ”Approch” phases this continues until the robotic rat reaches the social distance for interaction Z_{social}, once within the social distance the robotic rat begins performing the desired behavior with the laboratory rat, when the robotic rat finishes executing the behavior's motions it starts back tracking and this process again. Figure 5.17 demonstrates the behavioral interaction conducted with the laboratory rat using this algorithm, namely performing the PIN, POU and SNC behavioral patterns.

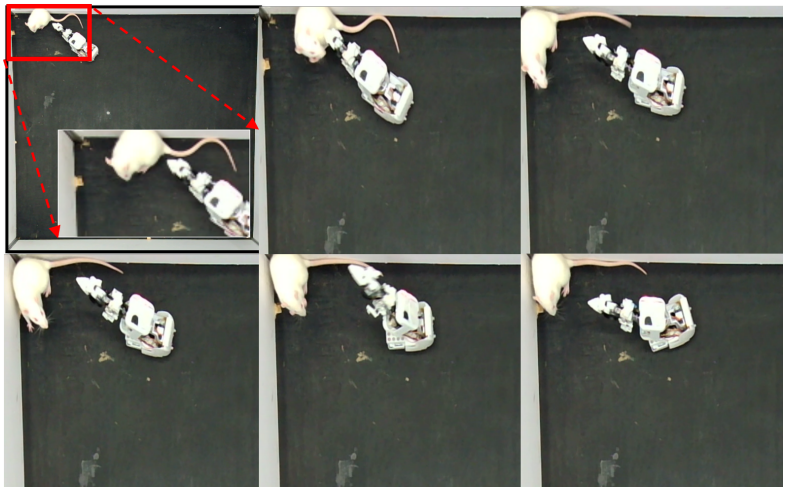
This experiment demonstrates that using the proposed motion planning framework, the robotic rat can successfully locate, track and perform behavioral interaction with the laboratory rat, hence validating this approach for conducting behavioral interaction experiments.

5.6 Summary

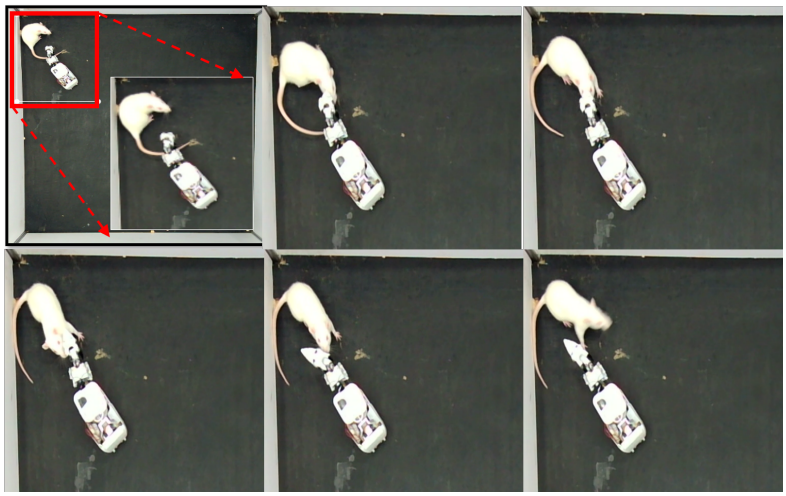
In this chapter, the Rat-to-Robot motion mapping algorithm's results were presented, the algorithm successfully produced joint angle trajectories for the robotic rat that correspond to the laboratory rat's behavioral motions, the algorithm was then used to produce a robotic rat behavior motion dataset. Subsequently, the probabilistic motion primitives framework was utilized to process the joint motion dataset and produce a ProMP model of each behavior within the dataset. Then, the effectiveness of tracking algorithm was verified, first through simulation and then through real experiments using a laboratory rat. Finally, the control policy for behavioral interaction was employed to demonstrate the validity of the proposed methods in robot-rat behavioral interaction experiments.



(a) PIN



(b) POU



(c) SNC

Figure 5.17 Behavioral Interaction Experiments performing different behaviors

Conclusions

The study of interactive behavior between laboratory labs is of great significance for the development of new drugs for psychiatric disorders and the exploration of disease mechanisms. Introducing controllable robotic rats to interact with laboratory labs can solve problems in traditional rat interaction experiments such as uncontrollable behavior, low reproducibility, and long interaction cycles. Thus, in order to realize robot-rat behavioral interaction experiments, this work aimed to address the issues of inconsistency in motion data representation space, insufficient bio-mimetic characteristics in motion representation and poor motion control tracking ability. The current methods used in the literature do not employ a data based approach to motion generation for robotic rats, this leads to motions that are not rat-like in nature and are unsuitable for conducting behavioral experiments. Additionally, the current methods used to implement tracking of the laboratory lab using integrated camera systems are unstable and fail to maintain continuous visual contact with the laboratory rat. To address these issues, this work introduced a Rat-to-Robot motion mapping algorithm to produce motion datasets from motion capture data. The datasets can serve as the basis for data-driven methods that can produce rat-like behavioral motions. Furthermore, the probabilistic motion primitives framework was utilized to generate motion trajectories from the produced dataset. Leveraging the properties of the ProMP model, the produced trajectories can be controlled to satisfy real-time motion planning requirements while conserving their rat-like nature. Then, a Rat-Tracking algorithm using Image Based Visual Servoing was developed to equip the robotic rat with ability of tracking the laboratory lab in an experiment environment. Finally, a control policy for conducting behavioral interaction experiments was presented. This work lays important foundation in the path of achieving autonomous robot-rat behavioral interaction.

Contributions and Innovations of the Work:

1. Rat-to-Robot Motion Mapping Algorithm: A novel Rat-to-Robot motion mapping algorithm was developed to address the inconsistency in motion data representation between laboratory rats and robotic rats. By establishing a mapping process and

utilizing non-linear optimization, this algorithm converts laboratory motion capture data into joint angle representation. It not only enables consistent motion analysis and reproduction but also provides insights into motion space representation mapping for other bio-mimetic robots.

2. Probabilistic Modeling of Bio-mimetic Behavioral Motions: The work employs the probabilistic motion primitive framework to process a joint space motion dataset. This framework generates a probabilistic model of bio-mimetic behavioral motions that accurately captures the patterns observed in laboratory rat motions. It not only contributes to data-driven methods but also provides a ProMP library of rat behavior motions for generating reproducible motion trajectories.

3. Robust Tracking Control Law: A robust tracking control law was designed by establishing the relationship between the position error of the laboratory rat and the joint velocities of the robotic rat, using Image Based Visual Servoing. The control law incorporates coordination between the robotic rat's torso and base, enabling accurate and reliable tracking of the laboratory rat during experiments. This contribution improves tracking capabilities and offers insights into real-time tracking for similar robots.

Future Work

This work laid essential groundwork for conducting future Robot-Rat behavioral interaction experiments and provided new insights for future research prospects. Although, the current

1. In regards to the Rat-Robot motion mapping, the accuracy of the current method can vary depending on which behavior is being processed, this is most likely due to the fact that the laboratory rat's body points can move relative to each other, whereas the robotic rat's body points cannot. Hence, this method can be improved by optimizing the robotic rat's body points position for different behaviors.

2. In regards to motion generation, the ProMP model used in this work provides a versatile tool to learn and manipulate rat-like motions, but is heavily dependent on the quality of the learning data. Thus, the motion generation can improved by using a

more sophisticated method of behavior and motion classification on the data before using it to produce the ProMP models.

3. In regards to Rat-Tracking, the current method can sometimes be fail to keep up with the laboratory rat's movement speed, this could be due to the robot's $40Hz$ control frequency. Therefore, the robotic rat's response time can be greatly improved by increasing its control frequency.

References

[1] Remington G. From mice to men: What can animal models tell us about the relationship between mental health and physical activity?[J]. Mental Health and Physical Activity, 2009, 2(1): 10-15.

[2] Smith J R, Bolton E R, Dwinell M R. The rat: A model used in biomedical research[J]. Methods in Molecular Biology (Clifton, N.J.), 2019, 2018: 1-41.

[3] Chai H, Li Y, Song R, et al. A survey of the development of quadruped robots: Joint configuration, dynamic locomotion control method and mobile manipulation approach[J]. Biomimetic Intelligence and Robotics, 2022, 2(1): 100029.

[4] Carius J, Ranftl R, Koltun V, et al. Trajectory optimization for legged robots with slipping motions[J]. IEEE Robotics and Automation Letters, 2019, 4(3): 3013-3020.

[5] De Viragh Y, Bjelonic M, Bellicoso C D, et al. Trajectory optimization for wheeled-legged quadrupedal robots using linearized ZMP constraints[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1633-1640.

[6] Neunert M, Farshidian F, Winkler A W, et al. Trajectory optimization through contacts and automatic gait discovery for quadrupeds[J]. IEEE Robotics and Automation Letters, 2017, 2(3): 1502-1509.

[7] Cebe O, Tiseo C, Xin G, et al. Online dynamic trajectory optimization and control for a quadruped robot [C]. 2021 IEEE International Conference on Robotics and Automation (ICRA). 2021: 12773-12779.

[8] Tang Y, Zhang G, Ge D, et al. Undulatory gait planning method of multi-legged robot with passive-spine[J]. Biomimetic Intelligence and Robotics, 2022, 2(1): 100028.

[9] Wang B, Zhang K, Yang X, et al. The gait planning of hexapod robot based on CPG with feedback[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 172988142093050.

[10] Li H, Zhu G, Arena P, et al. Bio-inspired locomotion control of gecko-mimic robot: from biological observation to robot control[J]. IEEE Instrumentation & Measurement Magazine, 2022, 25(9): 28-35.

[11] Fei F, Tu Z, Yang Y, et al. Flappy hummingbird: An open source dynamic simulation of flapping wing robots and animals[C]. 2019 International Conference on Robotics and Automation (ICRA). 2019: 9223-9229.

[12] Fei F, Tu Z, Zhang J, et al. Learning extreme hummingbird maneuvers on flapping wing robots[C]. 2019 International Conference on Robotics and Automation (ICRA). 2019: 109-115.

[13] Tu Z, Fei F, Zhang J, et al. Acting is seeing: Navigating tight space using flapping wings[C]. 2019 International Conference on Robotics and Automation (ICRA). 2019: 95-101.

[14] Iglesias A, Gálvez A, Suárez P. Chapter 15 - Swarm Robotics – a Case Study: Bat Robotics[G]. Yang X S. Nature-Inspired Computation and Swarm Intelligence. Academic Press, 2020: 273-302.

[15] Berlinger F, Gauci M, Nagpal R. Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm[J]. Science Robotics, 2021, 6(50): eabd8668.

[16] Berlinger F, Wulkop P, Nagpal R. Self-organized evasive fountain maneuvers with a bioinspired underwater robot collective[C]. 2021 IEEE International Conference on Robotics and Automation (ICRA). 2021: 9204-9211.

[17] Saadat M, Berlinger F, Sheshmani A, et al. Hydrodynamic advantages of in-line schooling[J]. Bioinspiration & Biomimetics, 2021, 16(4): 046002.

[18] Berlinger F, Duduta M, Gloria H, et al. A modular dielectric elastomer actuator to drive miniature autonomous underwater vehicles[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA). 2018: 3429-3435.

[19] Shi Q, Li C, Li K, et al. A modified robotic rat to study rat-like pitch and yaw movements[J]. IEEE/ASME Transactions on Mechatronics, 2018, 23(5): 2448-2458.

[20] Caluwaerts K, Staffa M, N’ Guyen S, et al. A biologically inspired meta-control navigation system for the Psikharpax rat robot[J]. Bioinspiration & Biomimetics, 2012, 7(2): 025009.

[21] Wiles J, Heath S, Ball D, et al. Rat meets iRat[C]. 2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL). 2012: 1-2.

[22] Lucas P, Oota S, Conradt J, et al. Development of the neurorobotic mouse[C]. 2019 IEEE International Conference on Cyborg and Bionic Systems (CBS). 2019: 299-304.

[23] Bing Z, Sewisy A E, Zhuang G, et al. Toward cognitive navigation: Design and implementation of a biologically inspired head direction cell network[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(5): 2147-2158.

[24] Huang Y, Bing Z, Walter F, et al. Enhanced quadruped locomotion of a rat robot based on the lateral flexion of a soft actuated spine[C]. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2022: 2622-2627.

[25] Shi Q, Gao J, Wang S, et al. Development of a small-sized quadruped robotic rat capable of multimodal motions[J]. IEEE Transactions on Robotics, 2022, 38(5): 3027-3043.

[26] Meyer J A, Guillot A, Girard B, et al. The Psikharpax project: towards building an artificial rat[J]. Robotics and Autonomous Systems, 2005, 50(4): 211-223.

[27] 李康. 机器鼠的仿生运动控制研究[D]. 北京理工大学, 2018.

[28] 贾广禄. 仿生机器鼠 SMuRo 设计、运动控制与视觉感知研究[D]. 北京理工大学, 2021.

[29] Gao Z, Shi Q, Fukuda T, et al. An overview of biomimetic robots with animal behaviors[J]. Neurocomputing, 2019, 332: 339-350.

[30] Landgraf T, Oertel M, Rhiel D, et al. A biomimetic honeybee robot for the analysis of the honeybee dance communication system[C]. 2010 IEEE/RSJ International Conference on Intelligent Robots and

Systems. 2010: 3097-3102.

[31] Griparić K, Haus T, Miklić D, et al. A robotic system for researching social integration in honeybees [J]. PLoS ONE, 2017, 12(8): e0181977.

[32] Landgraf T, Rojas R, Nguyen H, et al. Analysis of the waggle dance motion of honeybees for the design of a biomimetic honeybee robot[J]. PLoS ONE, 2011, 6(8): e21354.

[33] Landgraf T, Bierbach D, Kirbach A, et al. Dancing honey bee robot elicits dance-following and recruits foragers[J]. arxiv, 2018.

[34] Bierbach D, Landgraf T, Romanczuk P, et al. Using a robotic fish to investigate individual differences in social responsiveness in the guppy[J]. Royal Society Open Science, 2018, 5(8): 181026.

[35] Bierbach D, Mönck H J, Lukas J, et al. Guppies prefer to follow large (robot) leaders irrespective of own size[J]. Frontiers in Bioengineering and Biotechnology, 2020, 8.

[36] Gianelli S, Harland B, Fellous J M. A new rat-compatible robotic framework for spatial navigation behavioral experiments[J]. Journal of Neuroscience Methods, 2018, 294: 40-50.

[37] Heath S, Ramirez-Brinez C A, Arnold J, et al. PiRat: An autonomous framework for studying social behaviour in rats and robots[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018: 7601-7608.

[38] Menolotto M, Komaris S, Tedesco S, et al. Motion capture technology in industrial applications: A systematic review[J]. Sensors, 2020, 20: 5687.

[39] Putranto J S, Heriyanto J, Kenny, et al. Implementation of virtual reality technology for sports education and training: Systematic literature review[C]. 7th International Conference on Computer Science and Computational Intelligence 2022: vol. 216. 2023: 293-300.

[40] Cipresso P, Giglioli I A C, Raya M A, et al. The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature[J]. Frontiers in Psychology, 2018, 9.

[41] Field M, Stirling D, Naghdy F, et al. Motion capture in robotics review[C]. 2009 IEEE International Conference on Control and Automation. Christchurch, New Zealand: IEEE, 2009: 1697-1702.

[42] Shi Q, Gao Z, Jia G, et al. Implementing rat-like motion for a small-sized biomimetic robot based on extraction of key movement joints[J]. IEEE Transactions on Robotics, 2021, 37(3): 747-762.

[43] 陈晨. 基于机器鼠视觉的动物实时跟踪与行为识别[D]. 北京理工大学, 2022.

[44] Lynch K M, Park F C. Modern robotics: mechanics, planning, and control[M]. Cambridge, UK: Cambridge University Press, 2017. 528 pp. (OCLC: ocn983881868).

[45] Numerical optimization[C]. Nocedal J, Wright S J. Springer Series in Operations Research and Financial Engineering. New York: Springer-Verlag, 1999.

[46] Beck A. Introduction to nonlinear optimization: theory, algorithms, and applications with MATLAB [M]. Philadelphia: Society for Industrial and Applied Mathematics : Mathematical Optimization So-

ciety, 2014. 282 pp.

[47] Nemirovski A S, Todd M J. Interior-point methods for optimization[J]. Acta Numerica, 2008, 17: 191-234.

[48] Wächter A, Biegler L T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming[J]. Mathematical Programming, 2006, 106(1): 25-57.

[49] Ravichandar H, Polydoros A S, Chernova S, et al. Recent advances in robot learning from demonstration[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2020, 3(1): 297-330.

[50] Hussein A, Gaber M M, Elyan E, et al. Imitation learning: A survey of learning methods[J]. ACM Computing Surveys, 2017, 50(2): 21:1-21:35.

[51] Schaal S, Peters J, Nakanishi J, et al. Learning movement primitives[C].: vol. 15. 2003: 561-572.

[52] Li J, Li Z, Li X, et al. Skill learning strategy based on dynamic motion primitives for human–robot cooperative manipulation[J]. IEEE Transactions on Cognitive and Developmental Systems, 2021, 13(1): 105-117.

[53] Saveriano M, Abu-Dakka F J, Kramberger A, et al. Dynamic movement primitives in robotics: A tutorial survey[EB]. (2021-02-07). preprint.

[54] Paraschos A, Daniel C, Peters J, et al. Probabilistic movement primitives[C]. 2013.

[55] Viroli C, McLachlan G J. Deep gaussian mixture models[J]. Statistics and Computing, 2019, 29(1): 43-51.

[56] Loew T, Bandyopadhyay T, Williams J, et al. PROMPT: Probabilistic motion primitives based trajectory planning[C]. Robotics: Science and Systems XVII. Robotics: Science and Systems Foundation, 2021.

[57] Gomez-Gonzalez S, Neumann G, Schölkopf B, et al. Adaptation and robust learning of probabilistic movement primitives[J]. arxiv, 2020. preprint.

[58] Frank F, Paraschos A, van der Smagt P, et al. Constrained probabilistic movement primitives for robot trajectory adaptation[J]. IEEE Transactions on Robotics, 2022, 38(4): 2276-2294.

[59] Giancardo L, Sona D, Huang H, et al. Automatic visual tracking and social behaviour analysis with multiple mice[J]. PLoS ONE, 2013, 8(9): e74557.

[60] Nityananda V, Read J C A. Stereopsis in animals: evolution, function and mechanisms[J]. Journal of Experimental Biology, 2017, 220(14): 2502-2512.

[61] Saxena A, Schulte J, Ng A Y. Depth estimation using monocular and stereo cues[J].

[62] Paragios N, Chen Y, Faugeras O. Handbook of mathematical models in computer vision[M]. Boston, MA: Springer US, 2006.

[63] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77: 354-377.

[64] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection [J]. arxiv, 2020. preprint.

[65] Jiang P, Ergu D, Liu F, et al. A review of YOLO algorithm developments[J]. Procedia Computer Science. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19 2022, 199: 1066-1073.

[66] Chen C, Jia G, Gao Z, et al. A real-time motion detection and object tracking framework for future robot-rat interaction[C]. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic: IEEE, 2021: 7404-7409.

[67] Hutchinson S, Hager G, Corke P. A tutorial on visual servo control[J]. IEEE Transactions on Robotics and Automation, 1996, 12(5): 651-670.

[68] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]. 2011 International Conference on Computer Vision. 2011: 2564-2571.

[69] Serra P, Cunha R, Hamel T, et al. Landing of a quadrotor on a moving target using dynamic image-based visual servo control[J]. IEEE Transactions on Robotics, 2016, 32(6): 1524-1535.

[70] Chaumette F, Hutchinson S. Visual servo control. Part I. Basic Approaches[J]. IEEE Robotics & Automation Magazine, 2006, 13(4): 82-90.

[71] Chaumette F, Hutchinson S. Visual servo control, Part II: Advanced Approaches[J]. 2007.

[72] Hamner B, Koterba S, Shi J, et al. An autonomous mobile manipulator for assembly tasks[J]. Autonomous Robots, 2010, 28(1): 131-149.

[73] Sandakalum T, Ang M H. Motion planning for mobile manipulators—A systematic review[J]. Machines, 2022, 10(2): 97.

[74] Johnson M A, Moradi M H. PID Control[M]. London: Springer-Verlag, 2005.

[75] Vyavahare P, Jayaprakash S, Bharatia K. Construction of URDF model based on open source robot dog using Gazebo and ROS[C]. 2019 Advances in Science and Engineering Technology International Conferences (ASET). 2019: 1-5.

Publications During Studies

[1] Xie, H and Jia, G and **Al-Khulaqui, M**, et al. A Motion Generation Strategy of Robotic Rat Using Imitation Learning for Behavioral Interaction[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 7351-7358.

Acknowledgement

近 2 年的北理研究生生活即将告一段落，回首这段奋斗的日子，有很多关心帮助我的人需要感谢。首先，要感谢我的导师石青教授。本课题的选择、科研工作的开展和实施，以及到最后论文的撰写及答辩，都是在石青教授的悉心指导下完成的。石青教授深厚的学术素养、敏锐的洞察判断能力、勇于开拓的创新精神、严格求实的工作作风、忘我的工作精神给我留下深刻印象，是我学习以及日后工作中永远的标杆。石青教授细心地开导我、鼓励我，让我学会对科研耐心、也对自己有了信心。在此，我向我的导师石青教授致以最崇高的敬意和最衷心的感谢！

其次，感谢我的贾广禄和谢宏钊师兄对我科研工作上的支持与帮助。贾广禄师兄严谨的治学精神、深厚渊博的学术修养、都使我受益匪浅，我将视其为科研中的学习榜样。当我科研遇到困难时，谢宏钊师兄给予我孜孜教诲和无私帮助，教会我如何分析问题，找出问题根源所在，鼓励我继续努力解决问题。再次感谢贾广禄和谢宏钊师兄在生活和学习上给予我的关心与帮助。

还要感谢实验室的所有研究人员。感谢我的师兄高子航、陈晨、刘江、张石、申艺石、张雨来、王若超、周祺杰、于志强、陈勰、陈喆、闫书睿、高俊辉、郭越等。还有和我同届的郭晓雯、权小龙、斯云昊、肖航、唐睿等，以及师弟许毅、金彦周、李罡阳。还想感谢我的同学徐汉鼎、陈柯宏、仪传库、张岭、薛永锐。感谢留学生中心的牛岩婷老师和郑烨老师。感谢各位给我的鼓励与帮助，与你们共同学习使我收获良多，大家无微不至的帮助与关怀让我的科研生活充满快乐与幸福。

感谢我的父母对我不断地支持与鼓励，你们永远是我最坚实又最柔软的后盾。

感谢我的女朋友，在各个方面给我无私的关心、帮助、鼓励和支持，有你在我身边，让我勇于面对任何挑战，让我的生活充满了快乐和幸福。

最后，我要感谢所有在我成长道路上，关心、支持和帮助过我的人们，是你们的默默付出和支持，才成就了我的今天。对在百忙之中抽出时间对我的论文进行评阅和评议的各位专家表示诚挚的谢意！