Design, modelling and control of a quadruped robot with a bionic spine

Yunlong Lian

Doctor of Philosophy

University of York

School of Physics, Engineering and Technology

September 2024

Abstract

It has always been a challenge to make quadruped robots achieve flexible, fast, and robust movements like quadruped animals. From a mechanical structure perspective, most existing quadruped robots lack a bionic spine, which plays a critical role in locomotion. This thesis investigates the design, implementation, and control of a quadruped robot with a tensegrity spine, called TQbot, to improve gait performance and adaptability in various terrains.

Two bio-inspired spines based on tensegrity structures are developed for quadruped robots. They have three degrees of freedom (DOF) and can move in the roll, pitch, and yaw directions simultaneously, featuring both passive and active components to mimic the vertebrae and surrounding muscles in an animal spine. However, TQbot has a total of 15 degrees of freedom—three in each leg and in the spine joints—which makes the control system and gait generation complicated. Thus, a central pattern generator (CPG) model was designed and implemented to generate coordinated leg-spine gaits. It is enhanced by integrating proprioceptive feedback, enabling TQbot to adapt to uneven terrains.

This work aims to explore how the bio-inspired spine contributes to the gait of quadruped robots and to determine the extent to which it improves gait performance. A series of experiments in both simulations and real-world scenarios were conducted to evaluate the effectiveness of the spine in different gaits, including walking, trotting, pacing, and bounding. The results highlight that the spine improves speed and gait efficiency while helping to maintain stability, particularly in uneven terrain conditions. Furthermore, the experiments revealed that the phase relationship between spinal movement and leg movement is critical for optimal gait performance.

Based on the obtained results, the thesis concludes with a discussion on the impact of the bio-inspired spine on quadruped robots, providing insights into the design of legged robots with advanced locomotion capabilities. Recommendations for further work include refining the spine's control algorithms and extending the model to more complex environments.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my two supervisors, Mark A. Post and Andy M. Tyrrell, for their invaluable guidance, support, and encouragement throughout the course of my research. Their expertise, patience, and insightful feedback have been instrumental in shaping this thesis, and I am deeply grateful for the opportunity to learn from them.

I also would like to express my heartfelt thanks to my parents, whose unwavering love and support have been a constant source of strength. Their belief in me has motivated me to pursue my dreams, and for that I am forever grateful.

To my little sister, Chenyu Lian, thank you for always bringing joy and laughter into my life, even when I was busy or stressed with my work. Your energy, creativity, and funny stories brightened my days and reminded me to take breaks and enjoy the little things. You've been a constant source of happiness, and I'm so lucky to have you as my sister.

To my friends, Tianyuan Wang, Huiting Ge, Junbo Zhao, Tian Gan, Yuyao Wang, Zeliang Zhang, Zhihao Ma and Xinyu Zhang thank you for your companionship, laughter, and encouragement along the way. Whether through late-night conversations, shared challenges, or moments of relaxation, your presence has made this journey more enjoyable and manageable. I am incredibly fortunate to have each of you in my life.

I would also like to thank my friends back in China, Jing Luo, Ping Fang, Yuan Long, Zeyu Xue, Xin Chen, Jiazhi Wang, Yiyang Ji, Linfeng He, Xuan Cheng, Donghao Liu, Luyao Wang, Liting Feng, Fengrui Xie. Although we have been far apart, your constant

care, encouragement messages, and check-ins have meant so much to me. Your support from a distance reminded me that I am never truly alone on this journey and that your friendship has been invaluable in helping me remain positive and focused.a

Finally, to my girlfriend, Menglin Cui, thank you for being my biggest support throughout this journey. Your love, care and patience, especially during the most stressful times of writing this thesis, have meant the world to me. From taking care of me emotionally to preparing delicious meals that kept me going, you have been there every step of the way, offering encouragement and comfort. I could not have done this without you and I am forever grateful for your unwavering support.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

Publications

- Y. Lian, T. Wang, J. Ingham, M. A. Post, and A. Tyrrell, "Cpg-based locomotion control of a quadruped robot with an active spine," in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2023, pp. 177–189

Yunlong Lian

September 2024

Chapter 1

Introduction

1.1 Background and Motivation

Quadrupeds in nature have a limited range of movement in their leg joints, but can easily move on different terrains and exhibit high manoeuvres and flexibility. One of the reasons for this is that they have a flexible spine which plays a vital role in stable gait and agility. Legged robots have always been expected to have such capabilities to move on natural terrains and to perform tasks in harsh environments that humans cannot arrive at or are harmful to humans. Therefore, taking inspiration from biological structures, adding a bionic structure spine to robots and exploring its effect are promising ways for quadruped robots to achieve that.

It has two directions to enable quadruped robots to achieve animals level agility: algorithms and structures. On the algorithm side, model-based methods rely on accurate mathematical models of the robot's body and its environment [2–5]. These methods use pre-defined models to predict the robot's movements and adjust its gait accordingly. While model-based control can provide precise control in known or structured environments, its effectiveness diminishes when facing unpredictable or highly dynamic terrains due to the difficulty of creating accurate models for every situation.

Alternatively, learning-based methods offer a flexible approach by allowing robots to adapt their behaviours based on experiences [6–8]. These algorithms do not require precise models of the robot's body or environment but instead learn to generate appropriate gaits through trial and error. Reinforcement learning, for instance, enables robots to discover efficient locomotion strategies by interacting with their surroundings and optimising their movements over time. Learning-based approaches have shown promise in improving the robot's adaptability to unstructured or unknown environments, although they can be computationally expensive and require significant training data.

In addition to algorithms, the structural design of quadruped robots is another crucial factor in achieving high-performance mobility like animals. Animals have evolved over

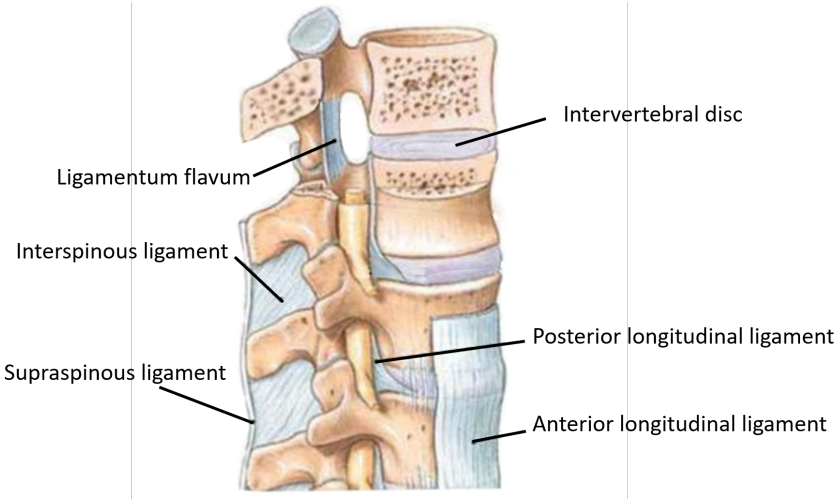


Figure 1.1 Human spine structure. The intervertebral discs connect each vertebra, absorbing shock and allowing flexibility. The ligaments prevent excessive flexion and enhance the stability of the spine.

hundreds of millions of years to have a body structure that is extremely adaptable to nature. By taking inspiration from biological structures, adding a bionic spine to robots and exploring its effect is a promising way for quadruped robots to enhance their mobility. A flexible spine allows for more dynamic movement, better shock absorption, and improved stability when navigating complex terrains. The design of such a spine, combined with advanced control algorithms, provides a pathway for quadruped robots to achieve high levels of agility and stability similar to their biological counterparts.

Biologically, the spine is divided into two components: the passive part and the active part [9]. The passive part consists of bones, ligaments, and tendons, and its deformation depends solely on the applied forces as shown in Figure 1.1. The bones are rigid parts connected by ligaments and intervertebral discs to form a stable and flexible structure. In contrast, the active part is composed of muscles that control the passive spine to produce various movements. Inspired by this, the robot spine may also be composed of passive and active parts to get the animals-level performance.

Over the past decade, there has been significant progress in the development of quadruped robots with various spine designs. Robotic spines have been explored in different contexts, including crawling robots, snake-like robots, salamander-inspired

robots, and modular robots, each contributing to improvements in agility, turning efficiency, and energy efficiency. However, the integration of a flexible spine in quadruped robots remains a relatively underexplored area, particularly when it comes to the coordination of spine and leg movements.

Different spine designs have been proposed for quadruped robots, each offering distinct advantages and trade-offs. The rigid spine, which does not feature movable joints between the robot's torso segments, is simple in structure, low-cost, and easier to control. This design has been adopted by many state-of-the-art quadruped robots, including ANYmal from ETH Zurich, the MIT Cheetah series, and the Spot robot from Boston Dynamics. While these robots have demonstrated impressive performance, the lack of a flexible spine limits their agility and adaptability on complex terrains.

In contrast, passive spines, which use elastic elements to connect the robot's body segments, allow for greater flexibility and energy efficiency. By storing and releasing energy, passive spines can improve high-speed locomotion and reduce the negative effects of external perturbations. However, the uncontrolled elasticity of passive spines can introduce disturbances, making it difficult to precisely control the robot's movements.

Active spines, on the other hand, provide additional degrees of freedom and allow for dynamic adjustments of the robot's spine during locomotion. This enables the robot to perform more complex and agile movements, such as increasing stride length for faster speeds or improving turning efficiency. However, the increased complexity of controlling an active spine, as well as the added weight and energy consumption from additional actuators, presents challenges in designing efficient and lightweight robots.

1.1.1 Exploration, Rescue and Transportation in Extreme Environments

Quadruped robots have the potential to revolutionise exploration, rescue, and transportation tasks in extreme environments. In planetary exploration, for instance, robots

must navigate rugged and unpredictable terrains while collecting valuable data. These missions often require robots to traverse steep slopes, loose soil, and rocky surfaces, where traditional wheeled or tracked robots may struggle. A quadruped robot, equipped with an active spine, can offer enhanced mobility and adaptability, making it well-suited for these challenging tasks.

Similarly, in the aftermath of natural disasters such as earthquakes or floods, robots can play a crucial role in search and rescue missions. Navigating through collapsed structures or hazardous areas, robots can locate survivors and deliver essential supplies without putting human rescuers at risk. The ability of a robot to traverse uneven terrain and maintain stability is critical in such scenarios, and a flexible spine can significantly improve its performance. Additionally, quadruped robots can be used to transport supplies to areas that are difficult to access, providing critical support in disaster relief efforts.

1.1.2 Robots in Extreme Environments

Various types of robots have been developed to operate in extreme environments, each with unique advantages and limitations depending on the task and terrain. Biped robots, for instance, offer human-like mobility, which can be advantageous for working human life scenes, but they often struggle with stability and balance for moving on rough terrains. Quadruped robots, on the other hand, offer greater stability and can traverse more rugged environments but may lack the agility or flexibility needed in certain situations.

Wheeled robots are generally faster and more efficient on flat terrain but face significant challenges when encountering obstacles or steep inclines. Tracked robots, often used in military and industrial applications, can handle rough terrains better than wheeled robots but tend to be slower and less agile.

Modular robots offer a flexible approach, where individual units can be reconfigured to adapt to different tasks and terrains. These robots can potentially address a wide range of challenges by reconfiguring themselves based on the environment. However, the complexity of control systems for modular robots can be a limiting factor.

1.2 Problem Statement

While advances have been made in quadruped robot design, the integration of a flexible, active spine remains an unsolved challenge. The design and control of the spine are crucial for achieving dynamic and efficient gaits, particularly when operating in unstructured or extreme environments. Current robotic systems without a spine limit the potential for dynamic movements, while passive spines introduce unpredictability due to their uncontrolled nature. This research specifically focuses on quadruped robots with novel spine designs, which allow for more dynamic and adaptive movement, potentially improving performance across a range of extreme environments. In order to provide a complete solution, the following issues need to be addressed:

(1) What kind of spine mechanical structure can maintain lightness and compliance while simultaneously providing multiple DOFs?

(2) Which method can parameterise the spine movement and coordinate it with the movement of the legs and generate various gaits by a few parameters?

(3) What phase difference, amplitude and frequency of spinal movements can improve the velocity and stability of gaits on various terrains?

This research addresses these challenges, at least in part, by developing a novel tensegrity spine for a quadruped robot. The spine's flexible structure allows for movement in multiple DOFs, enabling improved leg-spine coordination and dynamic adaptability. The primary focus is to investigate how the bionic spine can improve gait

stability, agility, and speed, particularly when traversing uneven or rapidly changing terrains.

1.3 Contribution

This thesis makes several key contributions to the field of quadruped robotics.

(1) It introduces the design and implementation of two novel bionic spine structures for quadruped robots. These spines, inspired by biological principles, include active and passive segments that offer a balance between flexibility and control. Furthermore, the research presents the first quadruped robot that combines a tensegrity spine with actuated legs, creating a more dynamic and adaptive locomotion system. This robot demonstrates improved performance in simulation, particularly in terms of gait coordination, turning efficiency, speed and stability improvement. The incorporation of a tensegrity spine structure provides unique advantages in terms of weight reduction and mechanical efficiency, making the robot more suitable for tasks in extreme environments.

(2) The thesis proposes a control framework based on a Central Pattern Generator (CPG). The CPG model in the framework is composed of modified phase oscillators, which can independently control of stance and swing phase for a quadruped robot. Moreover, proprioceptive feedback is integrated into the modified CPG model to generate dynamic leg-spine coordinated gait in real-time, allowing the robot to adapt to changes in rough terrain. This model-free approach eliminates the need for precise mathematical modeling of the robot's body, making it more adaptable to real-world conditions.

(3) Thanks to the bionic spine, it can rotate in multiple degrees of freedom simultaneously, a number of experiments in simulation have initially explored the impact of multidegree-of-freedom spinal movements on gaits. TQbot demonstrates improved performance, particularly in terms of improved gait coordination,

turning efficiency, speed, and stability. It provides guidance for the gait design of spine-assisted quadruped robots in the future.

1.4 Thesis Outline

The remainder of the thesis is structured as follows:

Chapter 2 provides a comprehensive survey of existing spinal structures for quadruped robots, CPG models, and the problems that may be encountered when applying reinforcement learning.

Chapter 3 introduces on the mechanical design of TQbot, especially the design of two bionic spines. The basic control methods of legs and spine based on kinematic and inverse kinematics were also described.

Chapter 4 describes the CPG model based on the modified phase oscillator with the internal feedback mechanism. The gait generation method and corresponding parameters are also given. Finally, the proposed CPG model integrating proprioceptive feedback and the proximal policy optimisation algorithm are introduced.

Chapter 5 demonstrates three simulated experiments that were conducted to quantify the performance of the spine in various gaits. The role of the spine on uneven terrain was also evaluated using the proposed CPG model that integrating proprioceptive feedback. Additionally, experiments on physical bionic spines were conducted to prove the effectiveness of the control method and show its properties.

Chapter 6 summarises all the work of this thesis and all results obtained. It also discusses the future research directions related to this work and provides a specific solution for assembling the part composed of tensegrity structures.

Chapter 2

Literature Review

2.1 Introduction

The dissertation presented a completed novel quadruped robot platform, which involved mechanism design, modelling, simulation, control and gait generation for a quadruped robot with a bionic spine. The chapter discusses and reviews the literature for the above domains around the quadruped robots with a spine. The current quadruped robot platform and spine design are reviewed in Section 2.2. Section 2.3 introduced and analysed the different central pattern generators for generating leg-spine coordination gaits. Furthermore, the issues of robot reinforcement learning are reviewed as a method to learn sensory feedback in CPG to generate dynamic gaits.

In summary, the chapter presents a comprehensive background and literature for building and controlling a quadruped robot with a spine. The challenges and issues in designing a bionic spine and generating leg-spine coordination gaits are introduced and analysed. For generating dynamic leg-spine gaits, the advantages and disadvantages of the existing methods are analysed. The challenges of bionic design and control, as well as the issues of leg-spine gait generation mentioned above, can be addressed in part by the contributions in this dissertation.

2.2 State-of-the-art Quadruped Robot Platform

In the recent decade, the robotics field has significantly flourished, leading to the emergence of numerous types of quadruped robots. Many state-of-the-art quadruped robot platforms are built for different purposes. This section reviews those quadruped robot platforms and analyses their performance from the aspect of the structure.

2.2.1 Quadruped Robots without a Spine

Some research institutes and commercial organisations have developed many state-of-the-art quadruped robots that aim to achieve quadruped-like performance in terms

of speed, agility, robustness, energy efficiency, dealing with external disturbance and walking in uncertain environments. Boston Dynamics, Unitree [10] and Ghost Robotics [11] are the commercial companies that have successfully mass-produced quadruped robots and applied them to the real world to some extent.

In addition, universities and research institutions developed excellent quadruped robots, for instance, the Cheetah robot series from the Massachusetts Institute of Technology (MIT) [12–14], ANYmal from the Swiss Federal Institute of Technology in Zurich (ETH) [15], HYQ [16, 17], HYQ2Max [18] from the Italian Institute of Technology (IIT), Minitaur robot from the University of California, Berkeley (UCB) [19], Serval from the Swiss Federal Institute of Technology Lausanne (EPFL) [20]. The focus of their research on quadruped robots varies, encompassing differences in mechanical structures and control methods, and explored diverse and comprehensive approaches to their development. Those representative quadrupedal robot platforms will be introduced here.

BostonDynamics BigDog

(a) BigDog (b) Spot (c) SpotMini

Figure 2.1 Boston Dynamics robots

Most quadruped robots are currently developed with electric actuators due to the high control accuracy, compactness and integration, such as MIT's directly electrically actuated, ETH's serial elastic actuator, Minitaur's direct-drive actuator, and Stanford doggo's quasi-direct-drive(QDD) actuator [21]. However, electric actuators cannot

support high loads compared to hydraulic actuators. Some hydraulic actuators have also been successfully applied on quadruped robot platforms, such as BigDog from Boston dynamics and HyQ [16, 17], HyQ2Max [22] and HyQReal from IIT.

One of the most famous quadruped robots is Bigdog developed by Boston Dynamics, with funding from Defense Advanced Research Projects Agency (DARPA) [2]. It is a hydraulic quadruped platform used to assist soldiers in transporting supplies and equipment in complex environments. BigDog achieved excellent movement ability and load capacity on various rough terrains. Additionally, it has excellent balance and maintains stability even when subjected to pushing or kicking. Each leg has four hydraulic actuators to power joints and a passively compliant foot. The actuator is equipped with precise position and force sensors, as well as aerospace-quality servo-valves, and each foot features force sensors. It is also equipped with an aerospace-grade IMU, and depending on the version, with a stereovision camera or a laser scanner for exteroceptive sensing of the environment. Its subsequent development, Spot and SpotMini, also showed strong adaptability to unstructured terrains [23]. It has had a profound impact on the development of quadrupedal robotics. The BigDog and SpotMini make people see the application prospects of quadruped robots in human life. Unfortunately, the researchers did not publish scientific papers about these two quadruped robots.

IIT HyQ

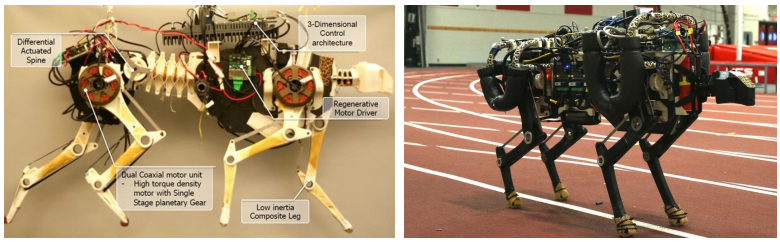
(a) HyQ (b) HyQ2Max (c) HyQReal

Figure 2.2 IIT HyQ

HyQ is a versatile quadruped robot for generating dynamic movement on rough terrains with 3 degrees of freedom in each leg [16, 17]. Its hip abduction/adduction joint is electrically actuated by a Brushless Direct Current (BLDC) motor with harmonic drives that are mounted onto the torso to reduce the leg inertia. The hip and knee joints are actuated by hydraulic cylinders, which aim to provide high torque and velocity. Another advanced robot, HyQ2max, evolved from the earlier HyQ robot [18]. It was designed to be more rugged and powerful with the addition of self-righting capabilities, which HyQ lacks. However, most of the improvements are in materials and hydraulic actuators, with only minor modifications to the mechanical structure. Both legs have similar kinematics with 3 degrees of freedom but HyQ2Max has a larger workspace in hip joints to support self-righting function. HyQReal is the third-generation hydraulic quadruped robot in the HyQ series, developed in collaboration with Moog Inc [24]. HyQReal is an autonomous robot integrating a complete hydraulic power system and a Lithium-Polymer battery within its torso. It's designed on the blueprint of its predecessors, HyQ and HyQ2Max It has the smart hydraulic actuation system to provide HyQReal with a strong load capacity and the ability to output large forces at the joints, making it pull a plane weighing about 3 tons.

MIT Cheetah

The Biomimetic Robotics Lab at MIT developed Cheetah series quadruped robots, which have made many innovations in leg and body design. In 2012, they designed a tendon-bone leg that resembles mammalian musculoskeletal structures to improve the strength and compliance of the leg [26]. In addition, the first generation of Cheetah explored the impact of flexible spine on energy efficiency and high-speed running and achieved a high-speed trotting gait by applying the novel spine [12]. Subsequent Cheetah II uses a spring-loaded inverted pendulum (SLIP) based method to implement a high-speed bounding gait [13]. The controller can generate a stable 3D bounding gait at speeds of up to 6.4 m/s without re-tuning any parameters. However, each leg of



(a) Cheetah 1 [12] (b) Cheetah 2 [13]

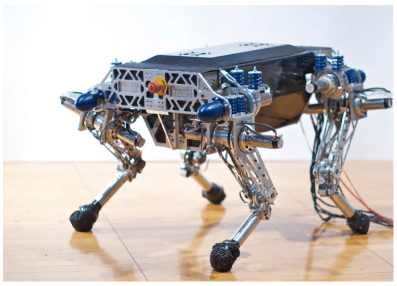
(c) Cheetah 3 [14] (d) Mini Cheetah [25]

Figure 2.3 MIT Cheetah series quadruped robots.

Cheetah II only has two degrees of freedom that limit the generation of other dynamic gaits. The limitation of leg workspace is solved by MIT Cheetah 3, demonstrating a more extensive range of motion space than Cheetah 2. Its modular controller can react quickly to changing terrain without a priori knowledge of the environment [14]. Furthermore, they also designed and developed a low-cost agile quadruped robot, Mini Cheetah, which is the first quadruped robot that can demonstrate a complete backflip on flat ground [25]. Its mechanical design of the low inertia leg had a profound effect on all subsequent quadruped robot development, most of which were designed with the similar kinematic structure of Mini Cheetah.

ETH ANYmal

Another state-of-the-art quadruped robot, ANYmal, was developed by ETH to support humans in harsh industrial environments [15]. They mainly study the mobility of quadruped robots on various rugged terrains, agile motion skills and high robustness. It is a promising quadruped platform to perform dynamics and agile manoeuvres to overcome gaps, stairs and obstacles. Thanks to the different mechanism designs of the



(a) StarlETH

(b) ANYmal a

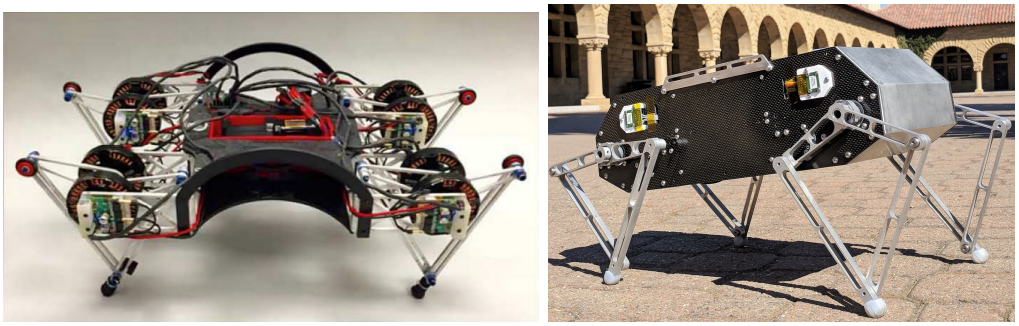
(c) ANYmal b

(d) ANYmal c

Figure 2.4 ETH ANYmal serials quadruped robots

legs, ANYmal shows a larger leg workspace, which lays a solid foundation for showing its excellent mobility and agility. Additionally, a self-developed highly integrated series elastic actuator, ANYdrive, offers high position and torque resolution accuracy. Meanwhile, they also made outstanding contributions to learning-based control methods [8, 27–29]. In 2019, they proposed a novel reinforcement learning-based control method, which enabled the robot to learn a faster trot than the controller previously applied to it and automatically stand up after a fall [8]. The method also addresses the reality gap by training an actuator neural network with real data, allowing the learned controller to be directly applied to physical robots without fine-tuning. Its follow-up research combines exteroceptive perception, such as stereo camera and LiDAR, to enable the robot to walk stably and autonomously in the wild environment even with some sensors having physical faults [29]. They also developed an open-source framework for training thousands of robots in parallel [6].

UCB Minitaur and Stanford Doggo



(a) Minitaur [19] (b) Doggo [21]

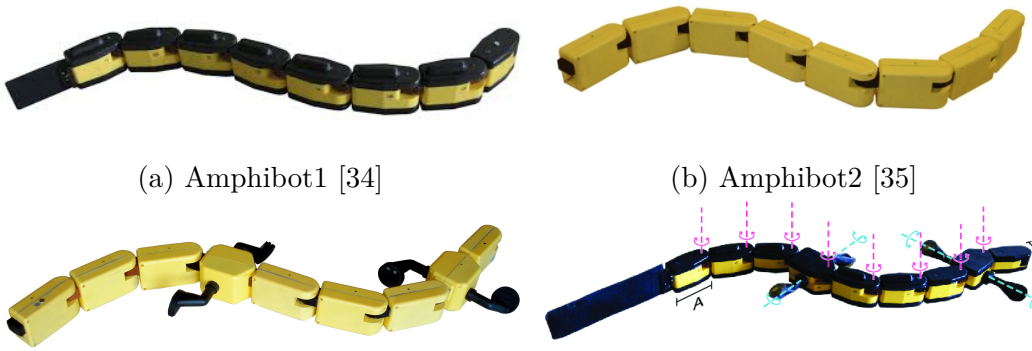
Figure 2.5 Minitaur and Doggo, designed symmetric five-bar legs with 2 DOF

Compared with the bulky and expensive HyQ series, UCB has designed and built a dexterity and cheap quadruped robot, Minitaur [19]. It applied direct-drive driving and designed a symmetric five-bar leg with 2 degrees of freedom. Its simple structure and lightweight body allow it to jump very high and perform high-speed gaits. Thanks to its simple structure and easy modelling, a learnt controller based on a proximal policy optimisation algorithm is successfully applied to Minitaur to perform trotting and galloping gaits in the real world [30]. Furthermore, a learning framework that allows real robots to learn to walk from scratch without modelling and directly in reality has been proposed and applied to the Minitaur. However, the main reason these learning methods can be successfully applied is that the Minitaur has a simple structure and is easy to model. With only two degrees of freedom per leg, the action space defined in reinforcement learning is greatly reduced. Stanford Doggo has leg configurations similar to those of Minitaur but solved the limitation of full revolution in the sagittal plane [21]. They employ a QDD mechanism to balance torque output and control bandwidth. Especially it excels in vertical jumping agility, matching the performance of the best-performing animals and surpassing previous robotic records by 22%. However, both robots share a common shortcoming in terms of mechanical design and degrees of freedom, i.e. each leg has only 2 degrees of freedom and can only

move in the sagittal plane. The low freedom of the legs limits the further development of some flexible movements.

2.2.2 Quadruped Robots with a Spine

Typically, the robot spine design significantly depends on the type of robot and its required functionality. Most robotic spine studies are often found in crawling [31–33], snake [34, 35], salamander [36] and modular robots [37]. For example, a salamander robot with an active spine is able to perform a walking gait on the ground and swimming motion in water. The locomotion that synchronises the spine and legs is determined by coupled phase oscillators [32]. Another biologically inspired sprawling posture robot can perform agile turning behaviours with its active spine [38]. Their controller also achieves precise control and synchronises leg and spine motion by solving the inverse kinematics of the spine.



(a) Amphibot1 [34] (b) Amphibot2 [35]

(c) Salamandra1 [36] (d) Salamandra2 [32]

Figure 2.6 EPFL Continuum Robot: Amphibot 1 and 2 are snake robots, salamandra 1 and 2 are salamander robots with four legs.

In terms of quadruped robots spine, some research explored the influence of the spine on energy efficiency [12, 39], moving speed [40–42], self-stability [43, 44] and agile turning behaviour [20]. However, the literature in this section will focus on the mechanical structure of quadruped robot spines. It can be divided into three types: rigid spine, passive spine, and active spine. A rigid spine does not have any movable joints between

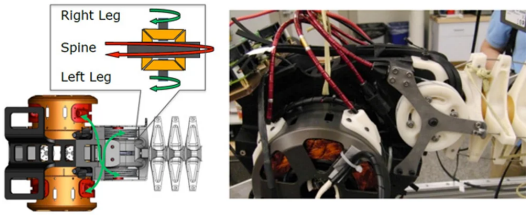
the shoulder and pelvic girdle of the body. It serves as a load-bearing and leg connection platform and very limited contributes to the movement. Those kinds of connections have a simple structure, low cost, and are easy to control, which is the choice of most quadruped robots, such as the previous section introduced quadrupedal platforms.

Passive Spine

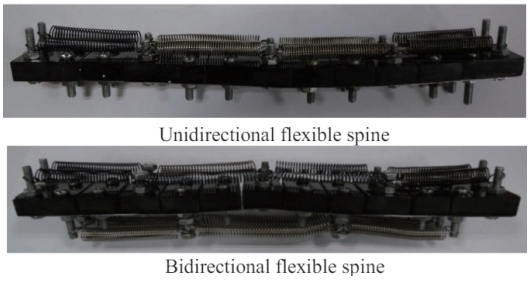
A passive spine usually uses elastic elements at the waist to connect the robot's body segments. It has several degrees of freedom but is not actively controllable. The elastic properties of a passive spine can reduce the negative effect of vibration when the robot performs high-speed movements and improve the ability to resist the impact of external forces. In 2013, MIT developed a quadruped robot with a passive spine called MIT Cheetah [12]. It is a flexible spine that can be actuated in the sagittal plane when the rear legs are in phase. Their research indicates that a passive spine can improve energy efficiency by storing and gaining energy from contracting and releasing the polyurethane rubber ring. However, those properties will cause additional disturbance to the robot as well as increase the difficulty of controlling and modelling the robot.

Another passive quadruped robot with a flexible spine, called Fanari, has been developed to investigate the impact of spine flexibility on stability, adaptability and velocity [45]. The robot spine structure is quite simple. It has three designs with springs: a bidirectional flexible spine (BFS) capable of bending upward and downward on pitch direction, a unidirectional flexible spine (UFS) that only bends upward, and a rigid spine (RS) that does not bend. Each spine consists of several Plexiglas segments mounted on a flexible metal sheet. The gap between Plexiglas segments allows the spine to bend on the sagittal plane. Springs can be mounted on the above and below the structure to form different spinal designs. The study demonstrated that BFS have the best performance, making BFS particularly effective in varied slope conditions.

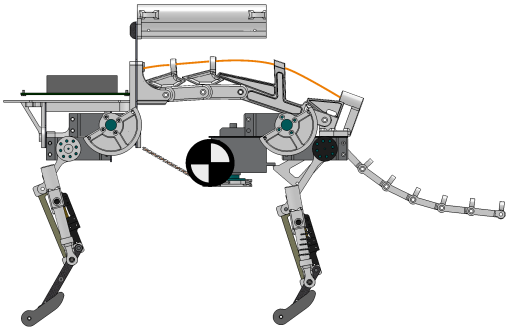
Also, Renny, a pneumatically-driven quadruped robot with a spine, was developed by UZH to emulate the running behaviour of a cheetah and investigate the effect of spine



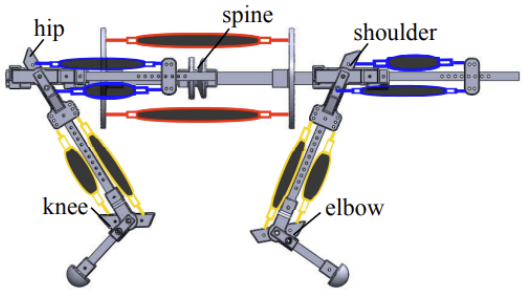
(a) MIT cheetah 1, passive spine [12]



(b) Fanari spines: passive spine [45]



(c) EPFL lynx spine, active spine [46]



(d) Renny spine, active spine [41]

Figure 2.7 Robotic spines. All spines in the figure can only move in the sagittal plane.

morphology on locomotion [41]. The study introduces three spine configurations: rigid, passive with tunable stiffness, and actuated spine. Experimental results indicate that a passive spine with higher dorsal stiffness compared to ventral stiffness achieves faster speeds than a rigid spine, and its ability to store and transfer energy efficiently because the compliant spine acts like a nonlinear spring, storing energy during the flexion phase and releasing it during extension. This energy transfer mechanism enhances locomotive efficiency and speed. Moreover, synchronizing spinal movements with leg actions in the actuated spine configuration significantly enhances speed by increasing limb swing and stride length.

Similarly, EPFL developed Lynx-robot, a quadruped robot with different spine and leg designs [46]. The spine include two configurations:(1)a directly actuated, single-joint spine and an actively supported; (2)passive compliant, multi-joint spine. Both configurations bend in the sagittal plane. These designs are analyzed for their impact on bounding gait, which is compared against an earlier robot model named Bobcat

that featured a two-segment leg design and a single-joint spine. The paper studied the effect of different spines and legs in terms of speed and stability. The study shows that while both spine designs enable the robot to reach moderate, self-stable speeds, the multi-joint spine provides better interaction with the environment, contributing positively to stable, open-loop locomotion patterns.

Additionally, they also developed another kind of quadruped robot with spines to explore the effect of spinal joint compliance and actuation on the bounding gait performance of a simulated quadruped robot [43]. The spine has one joint and bends on the sagittal plane only. The paper concludes that both passive and actuated spinal joints can significantly influence the dynamic performance of quadruped robots. The low-stiffness spine can improve bounding speed and tends to be more energy-efficient and self-stable than the actuated spine. On the other hand, the actuated spine enables better control over performance across varying conditions and generally leads to improved bounding speed and adaptability. The research indicated that the stiffness of the spinal joint is crucial for optimizing locomotion in robotic systems.

Except for the research on physical passive spines, the study demonstrates that while passive springs can utilize natural resonance for enhanced movement efficiency, the stability often remains a challenge by theoretical modelling and simulation [47]. rigid, passive spring, active spring, and virtual spring spine design are integrated into quadruped robot locomotion to research their synergetic effect. Through theoretical modeling and simulation, the paper demonstrates that while passive springs can utilize natural resonance for enhanced movement efficiency, the stability often remains a challenge. Conversely, active and virtual springs provide more precise control over spine dynamics, potentially improving the robot's interaction with its environment by dynamically adjusting spine stiffness in response to real-time feedback. These insights not only underscore the importance of spine flexibility in robotic design but also bridge a gap in the understanding of biological locomotion, offering a foundation for future innovations in robotic mobility.

Active Spine

Similar to passive spines, an active spine provides additional degrees of freedom but allows robots to actively change the motion of the spinal joints during the movement. This feature can increase the workspace of the whole robot to perform rich motor skills [20], and a large motion workspace can increase the stride length to improve the travel speed [40]. However, it will increase the weight of the robot's body, energy consumption and the difficulty to control due to the additional actuators and degrees of freedom. On the other hand, a flexible actuated spine is critical for quadruped mammal locomotion, which reduces energy consumption, keeps the entire body stable, and generates agile manoeuvres on different terrains. In terms of quadruped robots, an active spine allows for a richer range of behaviours, providing the potential for animal-like performance. Furthermore, the active spines of quadruped robots can be classified into two types based on the stiffness of their connecting parts: rigid active, soft active and wire-driven spine.

The rigid active spine commonly provides a solid connection that can improve turning efficiency and movement speed [39, 48]. It is commonly achieved by mounting actuators at the torso of the robot, but the weight of the entire robot and energy consumption are increased significantly. Indian Institute of Science developed a quadruped robot called Stoch 2 to research the spinal joint performance on the bounding gait [49]. Its active rigid spine features 2 degrees of freedom in the sagittal and transverse planes. Each joint in the spine is actuated independently with a range of motion from -15 to $+15$ degrees. They also utilised deep reinforcement learning to learn spinal movement from scratch to enhance dynamic and efficient movement by improving stride length. Studies also show that a robot's stability during movement would be reduced with increasing rigidity of spinal connection [43, 45]. Those problems pose a challenge to designing a spine with low stiffness and minimising weight gain.

A soft connected spine can solve the above problems to a certain extent [50–52, 47]. However, the additional perturbation caused by its elasticity also complicated accurate

modelling, making it difficult to combine the spine with the legs to generate agile gaits. In addition, most low-stiffness spines are designed with 1 or 2 rotational DOFs. Designing a structure with 3 DOFs and allowing the spine to move in all three rotational directions simultaneously is another challenge.

Wire-driven spine is an active spine positioned between a rigid spine and a soft spine [53]. It can provide better self-stability than a rigid active spine and causes lower disturbances than a soft spine. Another kind of wire-driven spine is a tensegrity spine that is a "tensile-integrity" structure and has unique physical properties that make it ideal for interaction with uncertain environments [54, 55]. This structure can perform simultaneous movement of multi-DOF, which lays the foundation for generating agile locomotion. However, those tensegrity spines are only used with immovable legs.

Another research introduces quadruped models with varying spinal joints, using minimalistic control strategies focussing solely on spinal movements [56], identifying that a model with two spinal joints significantly improves speed and stability by increasing stride length and effectively adjusting the centre of mass. Although the paper only conducted experiments in simulation, the results provide valuable insights into the significance of spine joint numbers in enhancing quadruped locomotion and offer a foundation for future studies in bio-inspired robotics.

The robots mentioned above and their spine information are shown in Table 2.1. It can be seen that most of the robot spines mentioned above have only one joint and one DOF, and most of the DOFs are rotated in the pitch direction. Only the Laika robot has the tensegrity spine with 3 DOF in the roll, pitch, and yaw directions.

2.3 Central Pattern Generators

In addition to the spine structure design, the generation of leg-spine coordination gaits is another critical problem. The performance of the bionic spine is highly related to the generation method. Central pattern generators (CPGs) are neural networks responsible

Table 2.1 Specification of mentioned robotic spines.

Year	Name	Joints Number	Actuated	DOF	Direction
2013	MIT Cheetah I	1	Passive	1	Pitch
2011	Fanari	1	Passive	1	Pitch
2013	Renny	1	Active	1	Pitch
2015	Bobcat	1	Active	1	Pitch
2015	Lynx	1/3	Active	1	Pitch
2015	Plearobot	5	Active	1	Yaw
2017	Inu	1	Active	1	Pitch
2018	Laika	1	Active	3	Roll/Pitch/Yaw
2019	Malinois	1	Active	1	Pitch
2020	Serval	3	Active	2	Pitch/Roll
2022	SQuRo	2	Active	1	Yaw
2022	Rat Robot	1	Active	1	Pitch
2023	SCIR robot	1	Active	1	Linear stretch
2023	Yat-sen Lion	1	Active	2	Pitch/Yaw
2023	Twist	1	Active	1	Roll

for producing rhythmic behaviours and are commonly found in both vertebrate and invertebrate animals. It has several properties desirable for locomotion: generating smooth trajectories, are robust to perturbations, and simple to implement.

However, the number of papers in terms of the leg-spine gaits generation method is limited. The survey revealed three main methods used to control the spinal motion of quadruped robots: (1) modelling the robot and defining desired spine movement in Cartesian space, then using robot kinematics to fit it [7,8]; (2) using a parameterised wave function to generate spinal motion but the cooperation of leg and spine is a problem[6]; (3) combining a bio-inspired method, CPG, with inverse kinematics can achieve precise control of the spine and coordinate it with legs, but this method is only available to control one of the orientations of the spine [spine controller sprawling]. In general, research on the spine of quadrupedal robots can be broadly divided into two areas: the impact of the spine on quadruped robots and controlling leg-spine coordination to generate locomotion. However, none of these studies had a universal control method that could integrate the spine and legs to generate various gaits.

2.3.1 Concept of CPG

Central pattern generator is a distributed neural network that exists both in vertebrate and invertebrate animals, capable of generating rhythmic behaviours, such as breath, chew and digest [57].

Additionally, the gait of most animals is certain rhythmic locomotion that is continuously repeated and coordinated between the limbs. However, it has evidence that CPG does not need sensory feedback to produce those rhythmic behaviours. For example, applying simple electrical or chemical stimulation to the spinal cord of the lamprey separated from the body can generate rhythmic patterns similar to intact locomotion, called fictive locomotion [58, 59]. On the other hand, it proved that CPG could output high-dimensional rhythmic behaviours while only receive a few low-dimensional input signals. Other similar experiments on different animals have proved the fictive locomotion is not a unique phenomenon, such as salamander [60], frog embryos [61] and other animals [62].

Although sensory feedback is not necessary for CPG to generate rhythmic behaviours, several experiments show it has a critical influence on shaping rhythmic patterns. For instance, manually moving the tail of lamprey can induce the locomotor activity of CPG; also, the frequency of locomotor will change with that of manually moving the tail [63]. Similar experiments on a decerebrated cat also confirmed that a looks like walking gait is induced by driving a treadmill, and changing the speed of the treadmill can induce a gait transition to trot [64]. Further research in many vertebrate animals have indicated that stimulate a specific region in the brain stem called the Mesencephalic locomotor Region (MLR) can also lead animals to generate fictive locomotion [62]. In some experiments, high-level stimulation on MLR will lead animals to generate high-speed gait, while different level stimulation on both sides of the MLR leads to turning gait [62, 65–68].

Furthermore, CPG is a distributed network consisting of multiple coupled oscillatory centres observed in the spinal cord of lamprey and salamanders [59, 60]. Each part of

the spinal cords has at least one unit of CPG to produce rhythmic patterns, and each unit at least outputs one degree of freedom. These units can be further divided into extensor and flexor muscles with independent oscillatory units separately [69].

To sum up, CPG is a complex distributed neural network in animals bodies, which can generate essential rhythmic behaviours. At the same time, CPG is also a basic module in the locomotion system of vertebrate animals. Other higher-level centres, such as the motor cortex, cerebellum, and basal ganglia, are responsible for adjusting those essential rhythmic behaviours according to environmental information. It makes animals have more robust gait control methods to deal with a changing environment and perturbation. The characteristics of the CPG model are summarised as follows:

(i) CPG can produce complex and high-dimensional rhythmic outputs by receiving a few low-dimensional inputs;

(ii) Sensory feedback is not necessary for the CPG model, but it is important for shaping rhythmic patterns;

(iii) CPG is a distributed neural network as a part of the locomotor system in vertebrate animals;

2.3.2 CPG models

CPG can be treated as a robot locomotion control method when transferred to mathematical models according to its properties. Due to the CPG lack of sound design methodology and a solid theoretical foundation, various CPG models emerge endlessly designed for different robots [57]. According to its structures and neuron type, there are four types of CPG models in robot control: neuron, oscillator, network, and hybrid.

Neuron models have precise biological meanings, but they are not suitable for robot control due to many control parameters and complex dynamic properties. Meanwhile, those models are mainly used to explain biology problems or test hypotheses on

biological CPGs. In contrast, oscillator models focus more on coupling relationships of CPG and how its control parameters affect the synchronisation and the phase lags. Network models are used to compensate for some disadvantages of classic CPG models and hybrid network models combining different types of CPG models into one model that aims to integrate the advantages to overcome the disadvantages of different models.

Neuron models

The most popular biological neuron model is the Hodgkin-Huxley model (i.e., H-H model) [70]. The H-H model is a set of nonlinear ordinary differential equations, which has four parameters: membrane voltage, activation and inactivation variables of the sodium and activation variable of potassium. A series of experiments determined those parameters; it also quantitatively described how sodium and potassium affect an alternation of membrane and action potential. However, it is a very sophisticated model and expensive to compute in real-time for robot control involving many neurons. Other CPG models based on the H-H model, such as FitzHugh-Nagumo, the Morris-Lecar, and the Hindmarsh-Rose models, also has detailed biological meanings includes many input parameters and complex neuron properties [71–73]. Those models provide new mathematics insights into the generation of action potentials and neuronal mechanisms.

In addition to detailed biological neuron model mentioned above, the leaky-integrator models and integrate-and-fire models are simplified connectionist neurons models [74–77]. The characteristics of these models are to describe behaviours of neurons and focus on producing rhythmic behaviours through network properties and how to couple each neuron via interneuron connections. The integrate-and-fire model will retain the voltage lifting when it receives a range of signals until it fires again, which does not fully conform to the characteristic of observed CPG behaviours [71]. In general, the biological neuron models should be further modified to apply it to robot locomotion control even though its characteristics are consistent with the CPG in biology.

Oscillator models

Another widely used CPG model in robot control is the oscillator model, a dynamical system consisting of coupled nonlinear oscillators. It is also the popular model in robot locomotion control, which is used in many types of robots, such as swimming robots [78, 35, 79, 36, 80], hexapod robots [81–84], quadruped robots [85–91] and biped robots [92–96]. One type of oscillator model can be applied to different robots. Therefore, the review of oscillator models will be based on their types.

(1) **Matsuoka Model**: Matsuoka model is a popular oscillator model named by its developer. He developed this model by analysing the properties of the mutual inhibition network and the generation mechanism of rhythmic pattern and proved the importance of adapting neurons for generating oscillation [97, 98]. The mutual inhibition network in this model consists of several neurons represented by a continuous variable model with a kind of adaption effect. On the other hand, Matsuoka model does not consider receiving sensory feedback as an input signal, so it cannot shape rhythmic behaviours according to environments. Other researchers improved Matsuoka model to deal with perturbation and rough terrain by receiving sensory feedback information [85–87, 92, 93]. Kimura et al. proposed Matsuoka models containing sensory feedback signals [86, 87], generating dynamic gaits for a quadruped robot to walk on 2-dimensional and 3-dimensional rough terrain separately. The neuron oscillator in the improved Matsuoka model is composed of an extensor neuron and a flexor neuron. A nonlinear differential equation represents these two neurons mutual inhibiting each other and each neuron.

(2) **Cyclic inhibitory Model**: A new CPG model is developed that enables a snake-like robot to produce 3-dimensional locomotion by controlling yaw and pitch joints [99]. Each oscillator in the cyclic inhibitory model has three neurons: yaw, pitch, and modulator. Those neurons are sustained-type and connect them through unilateral excitation to construct the CPG neuron network. They tested

the model on both simulation and a physical snake robot and got similar results, which is the snake robot can move forward on a smooth floor through sidewinding locomotion generated by the cyclic inhibitory model. However, the structure and environment of the simulated robot are different from those in reality. Besides, they also proved the necessary conditions for the cyclic inhibitory model to output stable rhythmic patterns continuously. Significantly, both the cyclic inhibitory model and Matsuoka model are sustained-type models, but they have different mechanisms to generate rhythmic patterns. The Matsuoka model mainly depends on a modulation function of the neuron to adjust periodic outputs. In contrast, the cyclic inhibitory model employs a potent cyclic inhibition among the neurons for generating rhythmic patterns. The cyclic inhibitory model can be an alternative approach to single-oscillator-dual-output cases because the oscillator can generate two different rhythmic outputs signals simultaneously;

(3) **Wilson-Cowan Model**: The Wilson-Cowan model is a classical mathematical model that was published in the early 1970s [100]. It is a model of the population dynamics of oscillators coupled with two types of interacting neurons: excitatory and inhibitory neurons. Some researchers are constantly improving it since then [101, 102]. A simplified model called Amari-Hopfield neuron is proposed for applying the Wilson-Cowan model to a robot [102]. This model is also composed of interconnected excitatory neurons and inhibitory neurons. It can receive two external inputs as sensory feedback and generate various rhythmic patterns through fine-tuning the parameters and has the limit-cycle behaviour. Besides, the Wilson-Cowan Model is also be used to study the properties of synchronisation and desynchronisation in a locally coupled network [103].

(4) **Kuramoto Model**: Synchronisation is one of the most common phenomena in physical and biological, and the fundamental in nonlinear systems. An efficient way to study the problem of synchronisation in a complex network is to model each neuron as a phase oscillator. The Kuramoto model is such a phase oscillator

model for analysing a set of coupled oscillators [104]. The Kuramoto model consists of several weakly coupled phase oscillators. The oscillators in this model are almost the same, and they interact through the phase difference between them. Each oscillator operates independently according to its given frequency, but the coupling mechanism will try to synchronise their respective frequencies. Due to its characteristic synchronisation, the Kuramoto model is widely used to study neuron oscillations and increase its neurobiological plausibility [105]. Additionally, it can be applied as a control method for robots gait generating. For example, a salamander robot can generate swimming and walking gaits through a CPG model composed of a phase oscillator [31]. Also, swimming gaits can be smoothly converted into a walking gait by changing the speed parameters, which is consistent with the phenomenon found in the CPG of vertebrate animals. Other similar researches are shown in these papers [106, 107].

(5) **Hopf Oscillator**: Hopf oscillator is another popular nonlinear oscillator that are ordinary differential equations [108]. It is commonly used to construct CPG models and applied to robot control for various locomotion generations, such as swimming, flying and walking. The CPG network composed of Hopf oscillators has some good properties to control robots. Firstly, different gaits can be generated through the same CPG network by modifying its phase, amplitude and frequency. Other attractions of the Hopf oscillator are the intrinsic synchronisation property and its limit cycle behaviour, which allow it to deal with external perturbations. Furthermore, the Hopf oscillator model has a good scalable. Different topological structures and coupling matrices can represent the coupling relationship between different joints. While the sensory feedback also can be added to the model to generate dynamic gaits [109]. An adaptive frequency Hopf oscillator is proposed for adapting its frequency without any input signals [110]. It can allow people to construct a CPG that enables robots to dynamically adapt their frequencies to generate desired rhythmic patterns in case of external perturbation. The adaptive frequency Hopf oscillator composed a successfully applied controller to

a compliant quadruped robot [111]. In this research, the adaptive controller can adjust body parameters according to the robot's resonant frequency. It means that the adaptive controller will adjust parameters to the actual gait of the quadruped robot operating, regardless of the gait given on the level of the CPG. The experiments demonstrated the controller can produce dynamic bounding and jumping gaits and proved that a few important gait parameters are affected by the geometry and movement of the robot.

(6) **Other Limit-Cycle Oscillators**: The nonlinear oscillator models mentioned so far all have a limit cycle behaviour that is good for coping with perturbation. The van der Pol oscillator and the Rayleigh oscillator are two types of oscillators to generate self-sustained oscillation. Both are represented by two similar second-order differential equations separately. The main difference between these two oscillators is the control of frequency. For the van der Pol oscillator, the change of the frequency parameter will only change its frequency, while for the Rayleigh oscillator, it will also change its amplitude. These two oscillators are used in electronic applications and robot control fields [112, 113].

In summary, the purpose of oscillator models is not to explain rhythmogenesis or test biological hypotheses but to research how the coupling relationship between oscillators and natural frequency difference affect the synchronisation and the phase lag among a population of oscillatory centres. Another proven fact in dynamical systems theory is that the dynamics of oscillatory centres' populations depend on the topological structure and type of couplings rather than the local mechanisms of rhythmic patterns generation. The fact also is the motivation of using oscillators to construct CPG models.

Network Models

Except for the chain connection, CPG models can be constructed as an artificial neural network. It allows to implementation of more coupling relationships between oscillators

and overcomes some of the disadvantages of classical CPG models. For instance, a two-level CPG model is developed to solve some shortcomings of the classical half-centre CPG models. The model consists of a half-centre rhythm generator(RG) and a pattern formation (PF) network [114, 115], which has mutual inhibitory interactions between neurons at each level. The neurons in the network are two types: interneurons and motoneurons, which are represented by the Hodgkin-Huxley type. In this model, RG can control the PF network activity and can also be operated independently; also, the separate networks demonstrated some properties of real-CPG that other single-level CPG models are difficult to show. However, this model focuses on re-implementing the CPG features found in cats, so that may need some modification in robot applications. The modified two-level CPG model with the phase resetting mechanism achieved a dynamic gait transition on a biped robot, which transfer quadruped gait to bipedal gait [116].

In order to control multiple-joints humanoid robots, a CPG model constructed of recurrent neural network(RNN) is proposed [117]. The humanoid robot, called HOAP-1, can generate walk gaits without sensory feedback through the RNN CPG model, and it also enables the robot to produce a pattern for up and downstairs using sensory feedback. Some researchers developed another CPG model, which uses an analysed RNN as the oscillator [118]. The novel CPG model keeps limit cycle behaviour and receives sensory feedback, and can online modify the control parameters during the generation of gait for controlling a quadruped robot. The results showed that the quadruped robot with this model has the ability to move in different terrains through dynamic trotting gaits. Although this model has achieved good results, its control of the robot is more complex than adjusting many parameters. Besides, combining the CPG model with RNN as a controller to apply to a hexapod robot [119]. Unlike the previous model, the RNN in this model is distributed on each leg and used for sensory predictions and instantaneous state estimations rather than as an oscillator in CPG. The experimental results showed the controller enables the hexapod robot to imitate complex insects' gaits and have a certain degree of damage recovery.

Spiking neural network is a model that imitates the human brain, which can also be used in the CPG model. According to the demonstration from [120], an adaptive CPG model based on spiking neural networks is proposed to generate desired rhythmic patterns. The CPG model only got good results on the data, but it has not been tested on a real robot.

Hybrid models

The CPG models mentioned so far have their own advantages and disadvantages. It is naturally thinking of combining multiple models into a hybrid system to overcome their shortcomings and make full use of advantages. Some researches focus on finding an appropriate approach to achieve the goal. For example, a combination of the van der Pol-Rayleigh oscillator system is developed to control a bipedal robot [121]. The simulated experiments demonstrated that the hybrid system got better performance than the system that uses only a van der Pol or Rayleigh oscillator. For controlling a complex humanoid robot with a flexible spine as humans do, the critical problem is making the robot keep balance during walking or running. Thus, a hybrid approach is proposed to solve the problem, which integrated the biologically-inspired approach and the engineering-oriented approach, namely CPG and zero moment point(ZMP) [122]. The hybrid control system can maintain stable walking in real-time with an adaption of flexible spine motions. The CPG and ZMP are responsible for generating rhythmic movement and adjusting the robot's postures to keep balance respectively. Nevertheless, this research only tested in simulation and cannot accept sensor feedback. Another similar research is to enhance the ability to against disturbances and uncertainties through an adaptive supper-twisting sliding mode (ASTSM) [123]. In order to enable a robot to walk on uneven terrains, sensory feedback is necessary. A combination of CPG and hormone inspired hybrid control is proposed for controlling a chain type modular robot [124]. The CPG receives hormone messages as sensory feedback to generate a specific motor primitive on modular robots regardless of structures. The

experiments are carried out in the simulation, and a hybrid control system makes the robot successfully walk through the obstacles and uneven ground.

2.4 Robot Reinforcement Learning

Reinforcement learning (RL) has been applied in various areas as a popular research field since it has made impressive achievements [125, 126, 8]. As an end-to-end learning method, reinforcement learning allows robots to find an optimal behaviour through constant trial-and-error experiments autonomously. It brings hope to develop intelligent robots, also provides new solutions for some control problems. On the other hand, the application of reinforcement learning in robots will also lead to problems that will not be encountered when applied in other fields: the curse of dimensionality, real-world samples, modelling uncertainty, and goal specification. In this chapter, those challenges are explained in detail and showed the corresponding solutions. In particular, the solutions of action and state-space dimensionality will be widely investigated.

2.4.1 Challenges in Robot Reinforcement Learning

The actions of robots in the real world are continuous, which is difficult to handle in reinforcement learning, even using discretisation or function approximation approaches. The generated data and computation are increasing exponentially for any legged robots as the number of joints grows. For instance, a quadruped robot with twelve joints and each leg has three joints, which has a total of twelve degrees of freedom. The number of optional actions is the twelfth power of the actions that each joint can perform. Moreover, adding one more joint will increase the number of actions by one more order of magnitude, called the curse of dimensionality [87]. It was unquestionably that training a robot without any limitation on the huge action space is challenging to discover an optimal policy. For robot control, it is not necessary to explore in entire action space. The reason is taking inspiration from human beings; the range of motion

of human joints is limited, which is different from robot joints. Most legged robot joints can rotate 270 degrees or even 360 degrees. With the growth, the range of motion of human joints will be further reduced, but human beings will be more proficient in gait control.

On the other hand, it will encounter problems that only exist in the real world when applying reinforcement learning in robot control. Due to a physical robot has complex systems to execute tasks and requires the control algorithm to run in real-time, the algorithm must have the ability to cope with delays in sensing and execution while reinforcement learning algorithms usually do not consider it. Besides, the safety of robots during the training process also needs to be considered to avoid human injury and hardware damage. In addition, the robot needs to be reset by human assistance at the beginning of each training episode. For example, an assistant needs to put the pole back in the initial position when training a cart pole balancing task. However, some robot training tasks may be started from an arbitrary initial state or even from an arbitrary environment. The above problems, called the curse of real-world samples, make it impossible to train robots in the real world.

Training robots usually take thousands of times, which has expensive cost in time and money in the real world due to wear and tear maintenance and repair, and manual assistance. An alternative is to use one or more physical robots to collect data in the real world and then training it in a computer by the collected data. Although the method can save time, it still cannot avoid high maintenance costs and robots purchase costs. All those reasons make training a robot in the real world almost unrealistic; hence, a robust model of robots and environments is necessary for training it. However, it is challenging to build a completely accurate model because most robots have a complicated physical system; even a tiny model error can be accumulated to a considerable error in a long time. It may lead to a tremendous difference in the performance of a robot between simulation and reality, which is also called the reality gap.

The reinforcement learning algorithms aim to maximise the accumulated long-term reward through performing desired behaviours. In contrast with board games and video games, some robot control tasks do not clearly define the reward function. If it is just simply defining a binary reward, for example, receiving a reward when finished a task, otherwise not receiving a reward, may cause robots to learn nothing or not expected behaviour. Therefore, the reward function should contain intermediate rewards to guide robot behaviours in the learning process, called reward shaping [127]. Unfortunately, designing a more complicated reward function still has problems. A poorly designed reward function can also cause robots to learn nothing. Similarly, the same reward function may lead to various policies that got the same rewards but learned behaviours are different. Although these learned behaviours may be better than the desired behaviours, they will cause dangerous situations for robots in most cases. For example, training a legged robot to learn to walk gait by a reward function that gives a positive reward if the robot moves forward or give a negative reward if the robot falls. A possible situation is that a robot only learned to stand still if it got a big negative reward when it fell. Another situation is that the robot may learn to use several legs to drag it forward, while the other legs do not move to maintain balance. Those uncertainties are not what we expected.

2.4.2 The Principles for Solving the Issues

Reinforcement learning provides a remarkable framework for robot control. Meanwhile, it also brings the unique challenges that reinforcement learning encountered in the fields of robotics. In order to solve the issues, three principles are proposed: effective representations, prior knowledge or information and approximate models [57].

As the previous section mentioned, one of the critical problems is the representation of action space. Reducing the dimensionality of the action space can significantly improve the role of reinforcement learning in robotics. On the other hand, most reinforcement learning algorithms include state and action-value functions, which estimate the value

of the action in the current state and the value of the robot in that state respectively. A good representation of value function also is the crucial component that successfully applied reinforcement learning to robots. Besides, a policy is a mapping from the current state of the robot to the possibility of the next selected action. Some reinforcement learning algorithms select actions without referencing the value function, but the value function is required to learn a parameterised policy. In this case, an approximation function representing a policy plays a vital role in policy-search methods. The above three representations that greatly influence robot reinforcement learning are classified to effective representation principles.

It is well known that human beings have specific prior knowledge before learning new skills in most cases, which speeds up the learning process. Similarly, prior knowledge may also be used to guide the robot learning process so that the robot can learn faster and better. Robot reinforcement learning can use three methods to apply prior knowledge: demonstrations, a predefined task structure, and directing exploration with prior knowledge. Animals and human beings usually learn some sports through observation and imitation, such as play football and basketball, and then practice to improve their level. According to the thinking, robots can also combine imitation learning with reinforcement learning to achieve the same goal [128], called apprenticeship learning, or using demonstrations to initialise reinforcement learning. When it comes to a redefined task structure, the prior knowledge may decompose a task into several basic parts or a series of tasks from easy to difficult. It can greatly reduce the difficulty of robot learning a complex task. The balance of exploration and exploitation always plays a key role in reinforcement learning; the research and discussion on it have never stopped. In robot reinforcement learning, sophisticated prior knowledge can guide robots to focus on the action or state spaces with a high probability of finding good behaviours or reaching good states. It prevents robots from exploring some meaningless spaces, thus saving training time and computing resources.

The first two principles discussed the improvement methods for robot reinforcement learning from the perspective of model-free. While model-based reinforcement learning

is efficient and popular in robot fields recently, which has explicit modelling of the transition dynamics. It is the third principle called approximate models. The purpose of the approach is to reduce the interaction with the real environment in the process of training. It is similar to how people rehearse in their minds to do something; hence, the learning process on a simulated system is called "mental rehearsal". Meanwhile, model-based methods can improve the sampling efficiency but may require more memory space. Mental rehearsal has a long history in robotics and has become a popular method in model-based reinforcement learning in recent years.

In summary, robot reinforcement learning has many research fields that are worthy of study. The literature review aims to explore one of the most promising directions that can be combined with CPG. In my opinion, the action space of reinforcement learning is the key factor that affects whether reinforcement learning can make robots intelligent enough. A good representation of action space can improve the learning speed and performance and enhance the policy-search and value-function-based algorithms. Therefore, the next section will investigate the representation of state-action space, value function and policy in robot reinforcement learning.

2.4.3 The Representation of Action Space

In the robot reinforcement learning field, the action space can be represented by low-level motor commands such as the angle of each joint. Also, it can be composed of high-level action commands, such as a lifting legs behaviour combined by the fixed angle of each joint. In terms of state space, it usually includes each joint angle and sensor data, such as posture, position and touch data. The robot will select actions in the action space to explore the optimal policy; in other words, to explore which action can get the maximum expected rewards in a state. It makes the representation of action space become one of the most important problems in robot reinforcement learning.

A common and straightforward representation of action space is hand-crafted discretisation. For some simple tasks, the dimensionality of action space is reduced by splitting into several regions. Robots will select different regions to explore according to a scheme in different situations. Some studies have used the method to enable the robot to achieve a good performance and to learn quickly, such as a ball on a beam [129], one degree of freedom ball in a cup [130], two degrees of freedom crawling motions [131] and gait generation for a quadruped robot [132]. Although it can reduce the difficulty of robot exploration in action space, the main issue is to find the correct number of regions to allow robots to learn a good policy. Hence, the performance of the approach depends on human experience, especially in some complex tasks. In other studies, the action space is manually discretised to a few actions rather than other regions. A study used reinforcement learning to help the controller achieve a stable gait transition and generate stable gaits in hexapod robots [133]. The actions in this paper are manually discretised into five footholds per leg. This approach of discrete action space may be more efficient for learning; it also has significant limitations. Fewer actions in action space mean that the robot cannot explore more agile motor skills and deal with a changing environment. Therefore, this kind of action space representation is suitable for robots working in simple environments or performing simple tasks.

In addition to the above discrete method, another smart way to deal with the action space in robotics is called meta-actions. It divides the action space into several basic actions, which robot can perform more intelligent actions to complete a task through executing these meta-actions continuously. For instance, a quadruped robot has an action space composed of meta-action that "move forward 1 meter", "turning 10 degrees", "stopping" and others. While a low-level controller can execute the specific operation of meta-action, such as rotating joints, stopping and correcting errors. Another advantage of the approach is that it can be applied to a wide range of reinforcement learning algorithms, such as Q-learning [134], policy-search policy [135] and others [130, 136–139]. Furthermore, if the meta-actions are not fixed defined in advance but are learning during the training process, it is called the hierarchical reinforcement learning approach.

2.4.4 The Representation of Value Function

There is usually a huge action and state space for robot control problems that cannot use tabular methods to find the optimal policy or optimal value function in a finite time and data. It causes the function approximation methods to be popular and successful in robot reinforcement learning. On the other hand, a robot cannot experience all situations due to a huge state space, so it should have the ability to deal with the situation they have never seen before but is very similar to the previous. Thus, it can be seen that the representation of value function is another critical factor for the successful application of reinforcement learning in the field of robotics. In order to construct a better value-function approximation, good features are necessary. For on-policy reinforcement learning, linear function approximation can yield good results with sufficient features [57]. On the other hand, it can be known that linear function approximation in the off-policy cases is not stable through literature survey [140–142]. However, as good features are rarely available in most robotics applications, various neural networks have been applied in robot reinforcement learning. For example, multi-layer perceptron was used to enable a mobile robot to learn some vision-based behaviours, such as servoing and wandering [143], as well as learning a walking gait for a quadruped robot [8]; fuzzy neural networks [144] and explanation-based neural networks [145] are used to learn basic navigation for mobile robots in an unknown and changing environment; also, CMAC neural networks are used in a biped robot to generate dynamic locomotion [146]. In general, neural networks are global function approximators that a local error may affect all neurons. It also may cause a neural network to overestimate the value of frequently occurring states in the value function, leading to fast divergence [141, 142]. To solve the above problem, an alternative approach is first proposed in 1992, which is to use statically clustering states to accelerate the process of a mobile robot learning to push a box [147]. In addition to using this method to enable a robot to learn behaviours, it can also be used to learn a navigation and obstacle avoidance task for mobile robots [148]. The decision

tree is another method to generalise actions to unseen states to speed up the learning process, applied to learn a penalty kick on a humanoid robot [108]. In contrast, those approaches are lack scalability to high-dimensional state and action space [149].

The Gaussian process regression (GPR) is another widely used method in reinforcement learning [150]. It is a non-parametric function approximator but potentially has higher computational complexity. An aerial blimp combines reinforcement learning with GPR to deal with high-dimensional state-action space to estimate the values of all state-action pairs [151]. This control method allows an aerial blimp to adapt to changing conditions through online learning without a complex hand-tuned model.

2.4.5 The Representation of Policies

Policy-search is an important method in reinforcement learning and can use an appropriate function approximation to represent it. However, the construction of function approximation needs to take into account task-relevant knowledge. Due to the policy choose an action to depend on the current state and action; thus it can be seen as a closed-loop controller. In another case, the policy can be regarded as an open-loop controller when the choice of action only depends on the time.

The first representation method is to use the joint trajectory points as the policy. It is usually an open-loop controller that the actions are a series of via-points generating from a function of time. In this case, the action is typically the desired joint angle, velocities and accelerations. In general, via-points are represented by splines, which are a particular piecewise function defined by polynomials. For example, a robotic arm similar to a human arm can learn to play with a Japanese toy called Kendama (very similar to a ball in a cup) by optimising the position of via-points depending on time [152]. Other similar research uses a cubic spline as a policy function approximation to represent a 13-point trajectory for a flapping system [153] or arm movement trajectory in the external perturbation recovery task [154]. A recent study proposed a policy representation consists of reference trajectories and a feedback component that provides

an action range to adjust the leg poses based on states [30]. The user can enable quadruped robots to learn a gait from scratch or follow the open-loop reference gait given by human guidance.

Furthermore, another method, called motor primitives [155], combines reinforcement learning with a basic movement parametrisation, such as biologically-inspired methods. CPG is one of the most popular methods to be a policy representation for reinforcement learning of discrete movements [156]. It is a dynamical system that generates basic movement related to phase, amplitude and frequency. Also, CPG can be an open-loop controller depending on time and a closed-loop controller to receive sensory feedback. Classic research combining reinforcement learning with CPG is used to generate a dynamic walking gait for a biped robot [146]. Similarly, reinforcement learning is used to adjust the representation of a method based on the half-elliptical locus that generates a rhythmic gait for an AIBO quadruped robot [157]. Besides, other dynamical systems also can be combined with reinforcement learning for different applications, such as learning a T-ball batting task [158], flapping a light switch [159], pouring water [160].

Furthermore, neural networks and radial basis functions (also called Gaussian kernels) can also be a representation of a policy. A multi-layered perceptron has been used to represent three policies for locomotion of a quadruped robot: high-speed, command-conditioned and recovery from fall [8]. Other closed-loop neural networks also can be used to represent a policy for learning rhythmic gaits of a biped robot [161, 162]. Radial basis function usually can obtain a good learning performance with fixed centres and widths while it may be a challenge in robot reinforcement learning. Nevertheless, it also has been applied to learn an open-loop gait [163, 164], a closed-loop cart-pole swing-up task [165] and a block-stacking task [166].

2.5 Summary

Currently, most quadruped robots do not have a movable spine on their torso due to high development costs and the complexities of design and control. While a passive spine can improve energy efficiency and impact resistance, it introduces additional perturbations that complicate control and modelling. On the other hand, an active spine provides additional degrees of freedom and an expanded joint workspace, but it also increases the overall weight and cost of robots.

An active spine of a quadruped robot has the potential to contribute to movement speed, turning efficiency, agile motion, energy efficiency, and self-stability. However, developing a spine that simultaneously boosts all these performance metrics is challenging. Therefore, the design of a spine should follow the below design features to maximise the performance of a quadruped robot:

(1) Multiple degrees of freedom: Multiple degrees of freedom allow the spine to function in a variety of gaits, e.g., bending of the spine in the sagittal plane can significantly improve the running gait of a quadrupedal robot. For example, bending the spine in the sagittal plane can significantly improve the running gait of a quadrupedal robot.

(2) Large workspace: The largest possible workspace for the spine joints while having multiple degrees of freedom allows the robot's leg space to be greatly increased to produce longer stride lengths to increase the robot's travelling speed.

(3) Simultaneous Actuation of Multiple Degrees of Freedom: The quadruped spine has a very important ability to actuate multiple degrees of freedom in the spine simultaneously to produce a more robust gait and agile movements.

(4) Low Stiffness: Various studies have shown that a low-stiffness spine performs better in adapting to different terrains and in terms of its own stability. Also, a

low-stiffness spine can be more energy efficient than a stiffly attached spine by using elastic elements to store energy.

(5) Lightweight: Adding an active spine usually requires the addition of extra motors, which will increase the weight of the robot and consume more energy. It is important to design a lightweight spine to minimise the extra weight of the spine.

The other hand, CPG is a promising method for robot locomotion control through extensive literature investigation. Firstly, CPG can be applied to a wide range of robots, especially robots with many joints and need to coordinate with each other. The reason is that CPG is well suited for distribution implementation due to each oscillator corresponds to a joint and independently output the angle. At the same time, the calculation of each oscillator needs the information of the connected oscillator, which gives an intrinsic synchronisation property to CPG. Hence, it is especially suitable for modular robots, such as snake robot [35, 36, 99], salamander robot [31] and fish robot [79, 80]. Secondly, CPG can generate various locomotion through adjust parameters in the same topology structure. Almost all of the CPG models mentioned in the second section can produce more than one gait for a robot. At the same time, the number of control parameters used in CPG models is obviously less than those of traditional control methods. For generating robust locomotion that can cope with different terrains, environmental information and sensory feedback are necessary. CPG provides a sound basis to integrate sensor data, adding feedback signals as coupling terms in the differential equations. Furthermore, the limit cycle behaviour exists in CPG models, which enhances the stability of generated gait. In this case, the control system is able to gradually recover unstable locomotion to rhythmic gaits from the external perturbation. It may provide a new way to generate dynamic gait through differential equations. Finally, it is laid a new foundation for learning methods. Supervised learning can be used to optimise the CPG if the desired rhythmic patterns are known or unsupervised learning if the desired rhythmic patterns are unknown. On

the other hand, CPG can be a basic controller in a robot while other learning methods as a higher-level controller to modify the rhythmic patterns into dynamic patterns.

However, CPG requires extensive experiments and tedious manual tuning, while reinforcement learning as an end-to-end method can automate this process. Some recent research has successfully applied reinforcement learning to real legged robots. For instance, the proximal policy optimisation(PPO) algorithm has been applied to a quadruped robot called Minitaur [30]. It provides a controllable learning process to users. The user can enable robots to learn a gait from scratch or follow the open-loop reference gait given by human guidance. It should be noted that the action space of this paper is continuous but limited to a range by human experience. Limiting the action space by human experience can reduce the training time but also causes robots not to learn agile motor skills in a short training time. Hence, the learned gait can only allow robots to walk on flat ground and not change the speed and directions. In another study, Trust Region Policy Optimization(TRPO) is trained for a quadruped robot to generate various speed locomotion and recovery from a fall [8]. The action space in this study is continuous and does not have any conditions to limit its range. Thus, they designed an extreme complex reward function and spent a long time training the quadruped robot. It can be seen from these latest researches that researchers did not focus on solving the curse of dimensionality in the legged robot applications; hence, they must spend more time on designing the reward function or training robots.

Reinforcement learning is popular in robotics but faced several challenges: the curse of dimensionality, the curse of real-world samples, the reality gap, and designing the reward function. Three principles for solving those challenges are effective representations, prior knowledge or information, and approximate models. Among them, the curse of dimensionality is the study's focus issue, which has the most significant impact on robot learning gaits, and effective representations are one of the solutions. Therefore, effective representations are investigated through the literature survey. The representation of action space, value function and policy are critical research directions, and both can be

combined with CPG. From the biological point of view, CPG is promising to instruct reinforcement learning to explore the action space efficiently.

Chapter 3

Design, Implementation and Control of TQbot

3.1 Introduction

In chapter 2, the previous design of quadruped robot spines was introduced. TQbot is a quadruped robot with a novel active spine, specifically built to validate the usage of the spine in different gaits and adapt to harsh environments. Thus, the spine design and its drive method are the main novelty of the robot. Four moveable legs are also designed to attach to the body and explore the leg-spine coordination gaits.

In this chapter, the modelling, implementation, and control of TQbot will be described in detail. This chapter is structured as follows: Section 3.2 explains the design concepts. The mechanical design of the spines, legs, and feet will be presented in Section 3.3. Physical implementation, including hardware and electronics, is described in Section 3.4. Finally, the control methods of the spine and the whole robot are described in Section 3.5.

3.2 Design Concept

Developing a quadrupedal robot from scratch is a complex project. In terms of TQbot, a novel wire-driven spine is more complicated than traditional quadruped robots. To rapidly the development of robotic prototypes and validate various design concepts, 3D printing and laser-cut acrylic plates are primarily used for fabrication. Before designing a robot, it is crucial to define the design concept based on the chosen materials. This ensures that all components are manufacturable, easy to assemble, and can be conveniently modified or upgraded as needed.

3.2.1 Size

The main methods used to build the robot were 3D printing and laser cutting of acrylic sheets. Anycubic and Ultimaker printers were employed to fabricate the designed parts for TQbot. The maximum printing dimensions of these 3D printers

are $256 \times 256 \times 256mm$. The laser cutting equipment can handle acrylic sheets up to $1000mm$ in size and supports thicknesses of $3mm$ and $5mm$. To ensure sufficient strength, all acrylic sheets used in the construction were $5mm$ thick. However, due to the workspace limitations of the 3D printers and the size constraints of the acrylic sheets, the dimensions of the robot's largest components could not exceed these maximum sizes. Additionally, due to the slow speed of 3D printing, the printed parts should be designed at an optimised size to increase the success rate and minimise printing time.

3.2.2 Tolerance

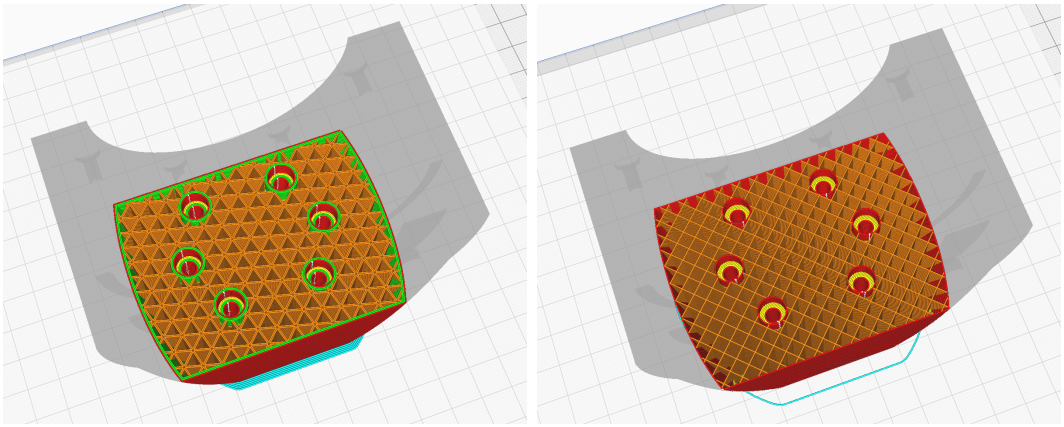
Tolerance is always a critical problem that should be considered during mechanism design. Specifically for TQbot, it was designed in multiple components that fit with others due to the printing size limitations of the 3D printer. Considering that compared to computer numerical control (CNC) milling machines, 3D printers have larger errors between the printed parts and the design size and the errors vary with different 3D printers, generally between $\pm 0.3mm$. The tolerance of TQbot parts is designed to $\pm 0.25mm$. Additionally, some printed parts need to be assembled with motors, so the error design of the screw holes should be slightly larger than the size of the screws to facilitate the installation of all screws with a tolerance $+0.1mm$.

3.2.3 Modular

During the design process, it is inevitable that parts already designed and produced will have to be modified and reworked. Since the time for 3D printing cannot be reduced by increasing the number of printers, the design should be as modular as possible for all components to allow for rapid updates and iterations of the hardware.

3.2.4 Strength

Additionally, due to the printing dimension limitations of 3D printers, many components need to be designed in segments and assembled post-production. To ensure adequate strength for the connecting and load-bearing parts, a fill density of 25% was selected, with a triangular double-line fill pattern, as illustrated in Figure 3.1. For non-load-bearing parts, a 20% fill density with a grid pattern and single-line fill was used, as shown in Figure 3.1b. These settings provide sufficient strength for the printed parts to support the robot to generate varies actions while optimising the robot mass.



(a) Strong print (b) Normal print

Figure 3.1 Different Fill rates and patterns setting. The figure shows a cross-sectional view of the hip joint connection components after slicing in CURA software.

3.2.5 Building

The 3D printers used in this project are employed fused deposition modelling (FDM). It is an additive manufacturing process that manufactures physical parts based on three-dimensional CAD data by adding materials layer by layer. Therefore, the design of the robotic part should follow the FDM characteristics so that it can be easily created by 3D printers. In addition, all parts are assembled by screws. As mentioned before, since there are many parts that need to be assembled with screws, it is necessary to consider the design of easy assembly and disassembly, so that all mounting screws

can be easily assembled with an electric screwdriver and without too many screws for assembly while ensuring strength.

In summary, the design of TQbot quadrupedal robot focuses on optimising component fabrication within the constraints of 3D printing and laser cutting technologies. All parts were designed to be compact and modular, facilitating easy assembly and rapid updated iteration. Tolerances were reasonably determined, with specific allowances made for screw holes to ensure proper fit despite the larger errors associated with 3D printing. To balance strength and weight, different fill densities and patterns were employed for load-bearing and non-load-bearing parts. The overall design was tailored to the capabilities of FDM to ensure efficient building and robust assembly.

3.3 Mechanical Design

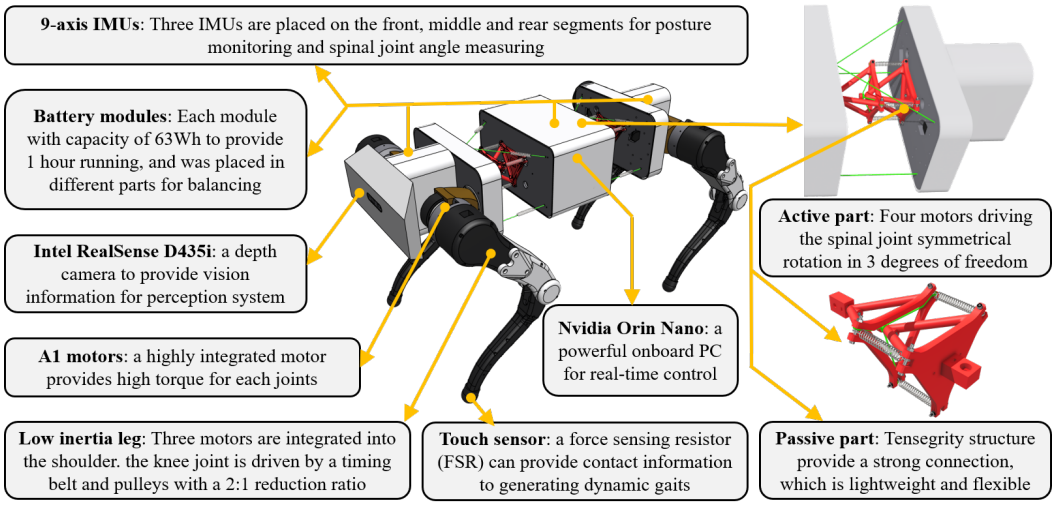


Figure 3.2 TQbot model in Shapr3D modelling software. All modules and the second spine configuration are described.

To explore the performance of a bionic spine in quadruped robot gaits, building a quadruped robot with moveable legs is necessary. TQbot is a low-cost, medium-sized quadruped robot platform, as shown in Figure 3.2. Its standing size is approximately $0.94m$ in length, $0.34m$ in width, and $0.5m$ in height. The mechanical design is divided

into three main components: the spine, body, and legs. The robot is equipped with a total of twelve degrees of freedom in its legs and three in its spinal joints. It weights approximately $30kg$ and is constructed from 3D printed components and acrylic sheets to reduce cost.

The spine is biomimetically designed, comprising active and passive parts, which mimic the musculature and vertebral segments of quadruped animals, respectively. This chapter presents two configurations of the active spine and two structures of the passive spine, exploring their effects on gait dynamics. The body is segmented into three modules, designated for housing the battery, control system, and sensory system. The leg structure resembles that of most current quadruped robots, with each leg featuring three degrees of freedom: one for extension and flexion, and two for rotation, corresponding to the adduction/abduction, hip, and knee joints. The robot's feet are also equipped with low-cost tactile sensors although they are not used in the thesis.

3.3.1 Bionic Spine

In mammals, the spine serves as a connection, and muscles around the spine generate torques or forces that control the spine to generate various motions. The bionic structures of quadruped robots have attracted an increasing number of researchers with the development of robotics. An actuatable spine for a quadruped robot provides a larger workspace to perform a wider range of motor skills. It can potentially improve the speed, stability, and energy efficiency of a quadruped robot during movement. However, most current quadrupedal robots with a spine lack bionic performance in their spine design, that is, they mainly focus on degrees of freedom and control of active joints while ignoring passive structures. Also, since the flexible spine can cause an extra perturbation that is difficult to control, it is usually rigidly connected or has a passive part only. The spine in this thesis includes passive and active parts, which are used to connect and drive the body segments separately. In this section, two different bionic spine configurations are presented.

Spine Configuration I

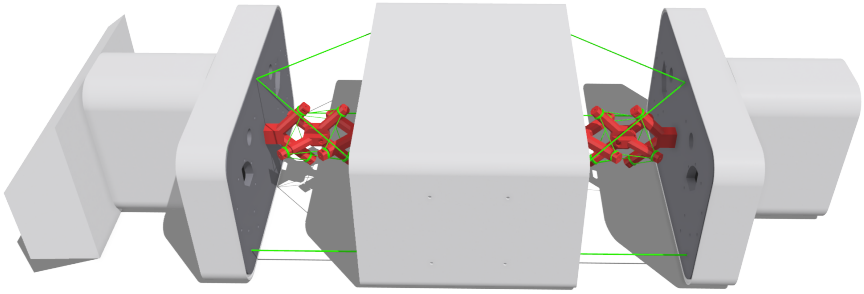
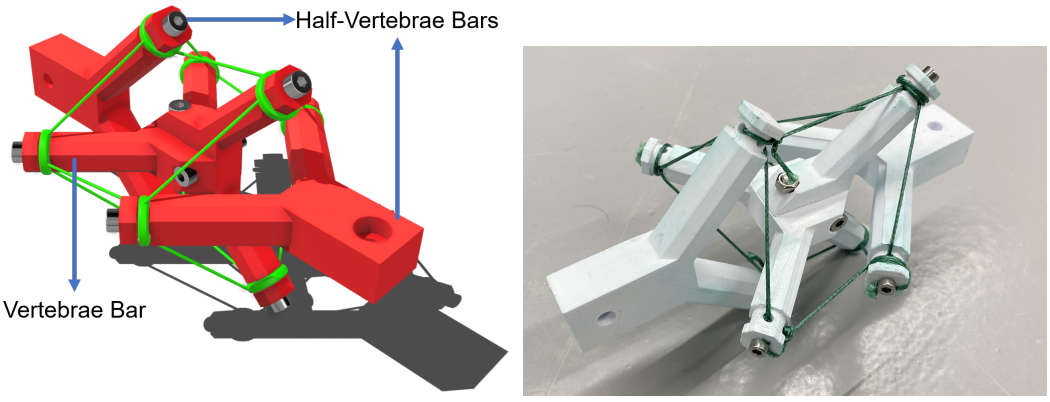


Figure 3.3 Bionic spine configuration 1. The green lines represent fishing lines, and the red parts are a 3d printed passive spine. Eight fishing lines in between body segments, representing the active spine, which connects front, middle and rear body segments.

The first bionic spine design, depicted in Figure 3.3, is constructed from fishing lines and 3D-printed components. This design features both passive and active parts. The passive part, made of 3D-printed parts and fishing lines, connects the front and rear body segments to the middle body segments. The active part, consisting solely of eight fishing lines, controls the spine to generate movement.

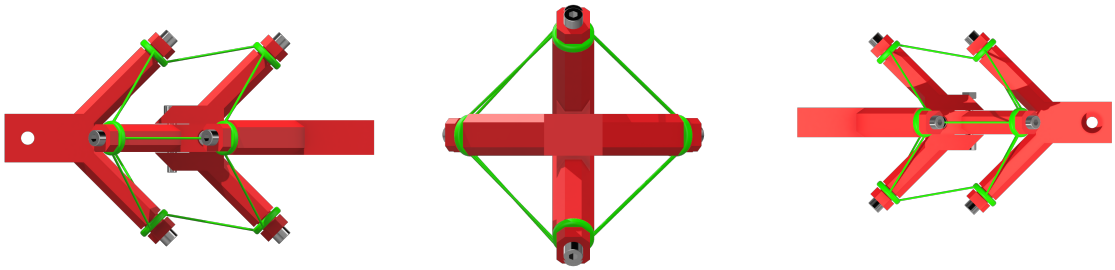
Passive Part



(a) Passive spine model (b) physical passive spine

Figure 3.4 Passive Spine 1. The left is the 3D model in software, and the green lines are fishing line. The right is the passive physical spine.

Figure 3.4 shows the passive part model in simulation and its physical entity. Figure 3.5 shows four different views of the first version of the passive tensegrity spine. As shown in Figure 3.4, the first version comprises one tensegrity vertebra and two half-vertebrae as the rigid elements to imitate its biological counterparts. The vertebra part is divided into two parts and is connected with two M3 screws for easy 3D printing. The ends of the two half-vertebrae are the mounting parts, which are designed with M4 screw holes for assembling them into the robot body.



(a) Side view (b) Front view (c) Top view

Figure 3.5 Three views of passive spine 1. Green lines are fishing lines, and each bar has screws at the end for adjusting the fishing line tension.

A similar tensegrity spine is reported in [55], but without fully tensioned. It results in a loss of inherent compliance and causes uncertained translational displacement during movement. The reason it is not fully tensioned is due to its drive method, which relies on changing the length of the rope in the tensioning structure to move the spine. While the spine of TQbot as a passive structure needs to connect different body parts. Under tension spine will lead it difficult to control and result in an error. Therefore, the first version of the passive spine in this project was assembled manually using fishing lines, resulting in continuous pretensioned passively for structural integrity and to constrain the translational displacement. The passive spine has a total length of 16 cm and can create a 10 cm gap between the two body parts to allow spinal movements.

However, manually assembling can lead to large errors that make it impossible to fully tension the spine, so a tensioner was designed on the head of each bar to adjust the

tension, as shown in Figure 3.6. According to this tensioner, the passive spine can be pre-tensioned well and the tensioning can be adjusted.

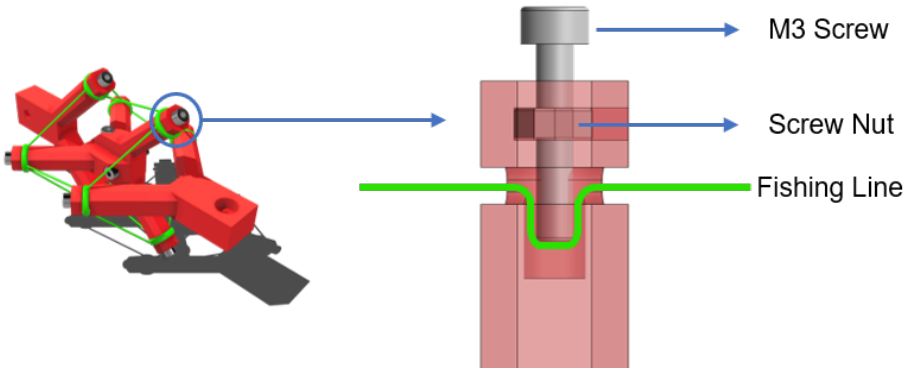


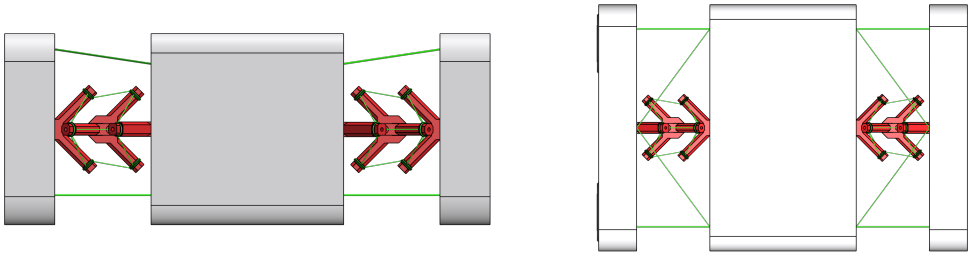
Figure 3.6 Tensioner for spine 1. Green lines represent fishing lines.

Active Part

To drive the tensegrity spine, a method is proposed in [55], which mounts motors on the tensegrity spine to adjust the lengths of the connecting cables, and its rotation in the roll direction is based on a motor installed on the tensegrity structure. This method can only afford small-size robots due to the limitation of the spine size and motors. To take advantage of the properties of the tensegrity structure to generate spine movements and activate three degrees of rotational freedom in roll, pitch, and yaw orientation in a bionic way, the completely restrained parallel manipulator (CRPM) configuration is adopted [167]. Four cables are the active part of the spine, which are used to directly connect and activate the relative movement of the adjacent body segments. It is a bionic method in which the active part imitates the muscles around the vertebrae to drive the spinal joints to generate various movements.

The first version of the active spine was designed to be symmetrical front to back as shown in Figure 3.7. This would cause it to produce symmetrical motions in the pitch and yaw angle directions and the same motions in the roll angle. The entire CRPM configuration of the bionic spine consists of eight $2mm$ diameter braided polyethylene lines with a load capacity of $250kg$. Adjacent body segments are connected by four

wires and a tensegrity part. Four motors are placed in the middle body segment, and each controls one wire that is connected to the front and rear body at the same time. Such configuration allows fewer actuators to actuate three DOFs of front and rear joints at the same time by adjusting the length of active cables but also results in a symmetrical motion of the two spines in the time and spatial domain.



(a) Side view (b) Top view

Figure 3.7 Views of the entire spine 1, green lines are fishing lines. The gap between each interval body segment is 10 cm.

Spine Configuration II

Similar to the first configuration of the spine, the second configuration of the spine consists of active and passive parts. The second passive spine is a tetrahedron tensegrity structure, while The structure of the active part of the front and rear spine was modified to be identical. It is worth noting that the second spine configuration adds springs to passive structures and replaces the fishing line with stainless steel wire for better strength, as shown in Figure 3.8.

Passive Part

The second structure is an upgraded version of version 1, as shown in Figure 3.9. The spine has two regular tetrahedron structures and these two parts are connected by springs and stainless steel wires. It has smaller limitations in roll angle than the first structure. Each of them is connected to different body parts with M4 screw holes.

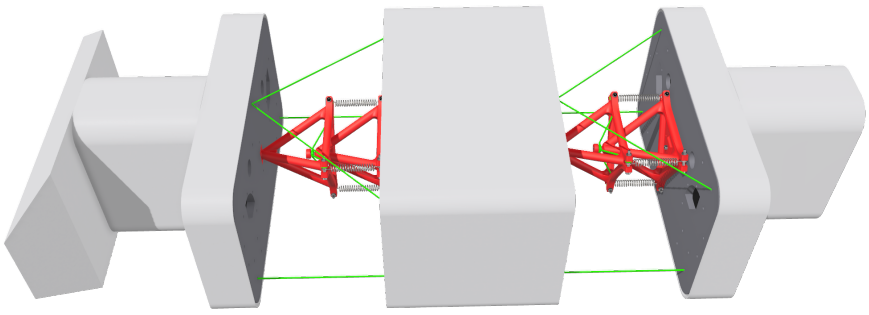
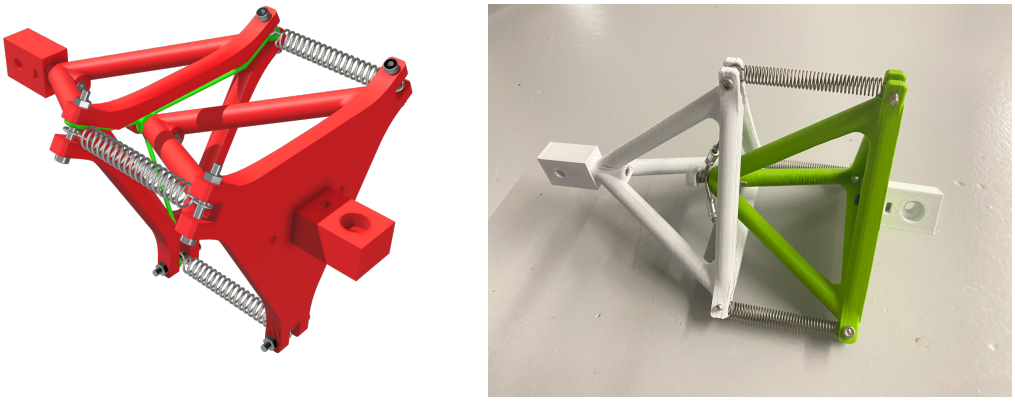


Figure 3.8 Bionic spine configuration 2. The green lines represent fishing lines, and the red parts are a 3d printed passive spine. Eight fishing lines in between body segments, representing the active spine, which connects front, middle and rear body segments

The connection between these two parts is elastic. Also, the structure of the spine constrains the translational displacement. The passive spine has a total length of 21cm and can create a 15cm gap between the two body parts to allow for spinal movements, which provides a larger workspace than the first version of the spine.

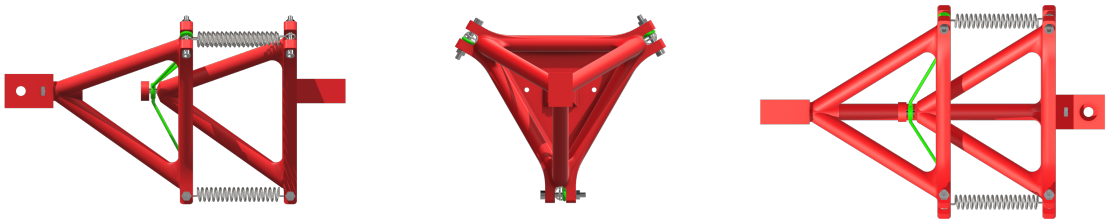


(a) Passive spine model 2 (b) Physical passive spine 2

Figure 3.9 Passive Spine 2. The left is the 3D model in modeling software and the right is the passive physical spine. It is connected by fishing lines (green lines) and springs.

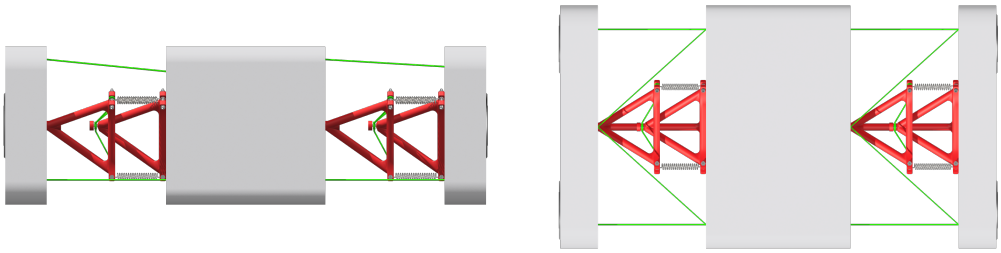
Active Part

The second spine configuration differs from the first active spine mainly in the connection between the middle body section and the back body section. The rear active spine is



(a) Side view (b) Front view (c) Top view

Figure 3.10 Three views of passive spine 2. Green lines are fishing lines, and each bar has screws at the end for adjusting the fishing line tension



(a) Side view (b) Top view

Figure 3.11 Views of the entire spine 2. The green lines are fishing lines. The gap between each interval body segment is 15 cm.

the translation of the front connection structure. This results in symmetrical movement of the front and rear body blocks in all three rotational directions.

The tensegrity structure as a robot spine has the following advantages compared with existing rigid actuated spines:

(1) Its independent three degrees of rotational freedom can be simultaneously actuated. The overall workspace of the robot is thus extensively increased to realise more advanced and agile manoeuvres.

(2) Tensegrity structures have higher structural efficiency compared with rigid structures, which allows a greater payload ratio of the robot.

(3) The inherent compliance of the tensegrity spine improves the robustness of the robot for situations such as external impacts and hazards.

(4) The flexibility of the tensegrity spine can potentially improve the energy efficiency of the robot and deal with external perturbation.

Differences Between Two Configurations

These two spine configurations are based on tensegrity structures but with distinct mechanical designs. It makes them have different functional characteristics. Structure differences are summarised in Table 3.1.

Table 3.1 Structure differences between two spine configurations

Parts	**Properties**	**Configuration 1**	**Configuration 2**
Passive	Stiffness	High	Low
	Connection	Fishing lines	Fishing lines, Springs
Active	Rear Structure	Symmetrical with front	Same as the front
	Rear Roll motion	Same as the front	Symmetrical with front

In terms of the first spine configuration, its passive parts are connected by high-stiffness fishing lines only, which causes it to maintain high structural integrity while also limiting spine movements in roll, pitch and yaw directions. The second passive structure not

only has fishing lines to limit the translation of elements for structural integrity but also replaces some fishing lines with springs to reduce movement restriction in all directions.

Quadruped mammals lifting diagonally opposite legs simultaneously during trot gait while they lift the same side legs together in pace gait. This difference in limb coordination resulted in different spine movements, which inspired the two designs of active parts with varying rear body motion in the roll direction. The first active part allows the robot to generate a natural pace gait as the front and rear bodies move in the same roll direction, whereas the second active part can generate a natural trot gait by making the front and rear bodies move in opposite directions.

Spine Assembly Challenges

Compared to rigid robots, wire-driven robots are cumbersome to assemble. Especially in tensegrity structures with continuous tension and compressive components. Most robots with tensegrity structure were under-tensioned, while others equipped driven parts on tensegrity parts, such as actuators, to change the tension state of it. Both are easy to assemble, but the under-tensioned tensegrity structure is unsuitable for use as a connecting part. In TQbot, the passive part is a tensegrity structure and is used for connecting body segments only. It requires a tensegrity structure to be in fully tensioned to provide a stable connection and withstand the load. However, assembling a fully tensioned passive spine has many challenges.

In both passive spines, each rigid part is theoretically exactly the same distance from each other. It leads to the length of the lines connecting them should be the same. However, it is difficult to do with manual assembly. Therefore, each passive spine length is about $2 - 3mm$ from the theoretical length. The active spine lines are also manually assembled, thus the active lines also have the same problem of inconsistent lengths, with an error of about $1 - 3mm$.

Moreover, manually assembling a passive spine is time-consuming. It usually takes 2-4 hours to complete one passive spine, as most of the time reassembly is required if the passive spine is not fully tensioned or the assembly error is larger than $3mm$.

Finally, the passive spine of the first design will become loose after 2-3 experiments and need to be reassembled, which is not suitable as a passive structure for real use. Therefore, the thesis applies the second spine configuration to the simulation in Chapter 5.

3.3.2 Body

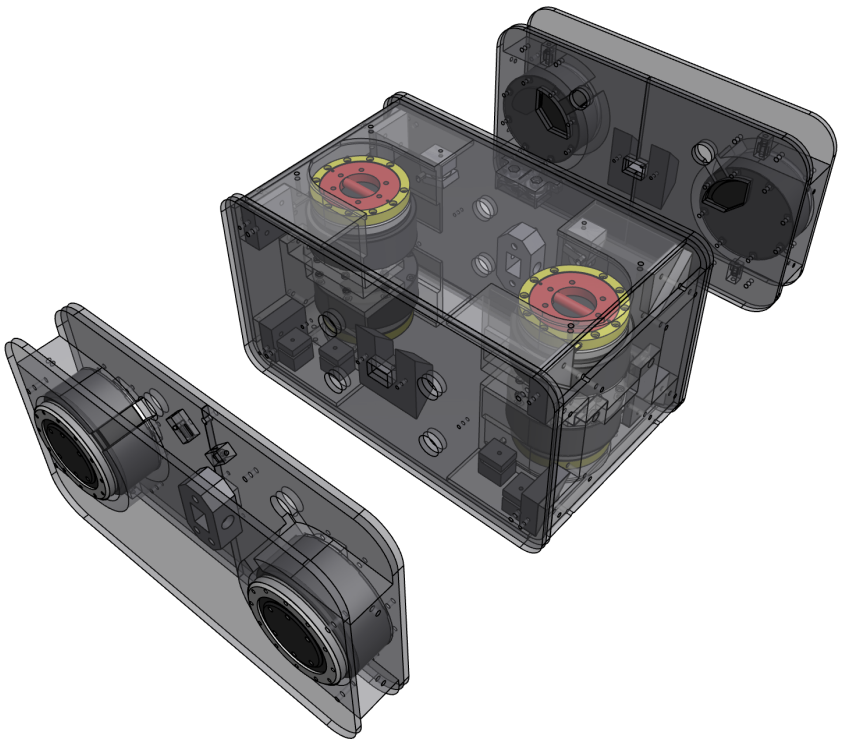


Figure 3.12 The front, middle and rear body segment without the spine. The two motors are placed in each of the front and rear body segments, and four motors are placed in the middle body segment. The shell is set to transport to show the inner structure.

Typically, the body of a quadrupedal robot mounts the motors, batteries, and other electronics, such as control circuits. Unlike traditional quadrupedal robots, TQbot

features a wire-driven bionic spine, which is more complex to design and build compared to conventional quadrupedal robot bodies. The body consists of three parts: front, middle, and rear, made using 3D printing and acrylic sheets, as shown in Figure 3.12. These parts are connected by the active and passive spine mentioned in the previous section. An onboard computer, batteries, power management system, and a depth camera are integrated into different parts of the body.

The front segment serves as the head of TQbot and houses two actuators for the shoulder joints. An Intel RealSense D435i stereo camera with IMU is mounted on the front of the head to provide real-time monitoring of the ground and the area ahead. In the middle segment, four actuators for spinal control, an IMU for posture measurement, and a powerful Nvidia Jetson Orin Nano for locomotion planning are installed. The rear segment mounted two additional actuators that control the shoulder joints of the rear legs. Each body segment has its own battery module to power it independently.

The internal structure described in this section is the second spine configuration, while the first spine configuration can be achieved simply by exchanging some parts.

Front and Rear Body

The front and rear body segments have identical dimensions, measuring $330 \times 52 \times 190$ mm in length, width, and height, as shown in Figure 3.13. To ensure sufficient strength, a $5mm$ thick acrylic sheet is placed on both the front and rear segments, sandwiching the two 3D-printed components in between, and secured with screws. Due to the limitations of the 3D printer size, the 3D-printed part in the middle section is divided into two separate left and right pieces. Each piece includes the motor mounting slots, power and control cable outlet. In addition, rope tensioners are designed to adjust the four spinal active fishing lines to reduce errors caused by manual installation. To make the structure as modular as possible and have sufficient strength, passive spine connectors were printed separately and connected with M3 screws.

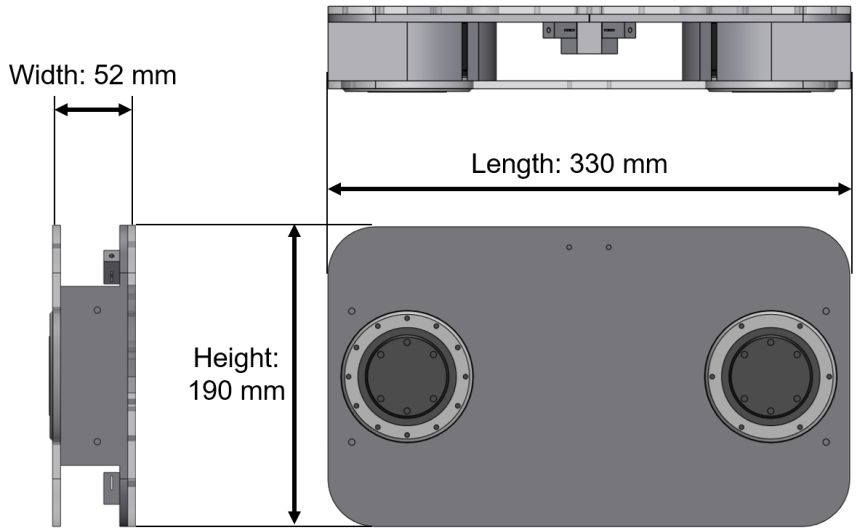


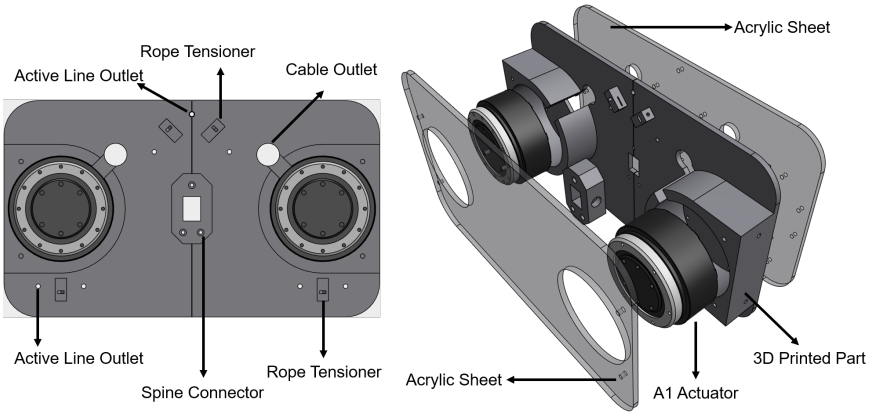
Figure 3.13 Front body segment dimensions diagram. The front and rear body segments are the same dimensions.

Middle Body

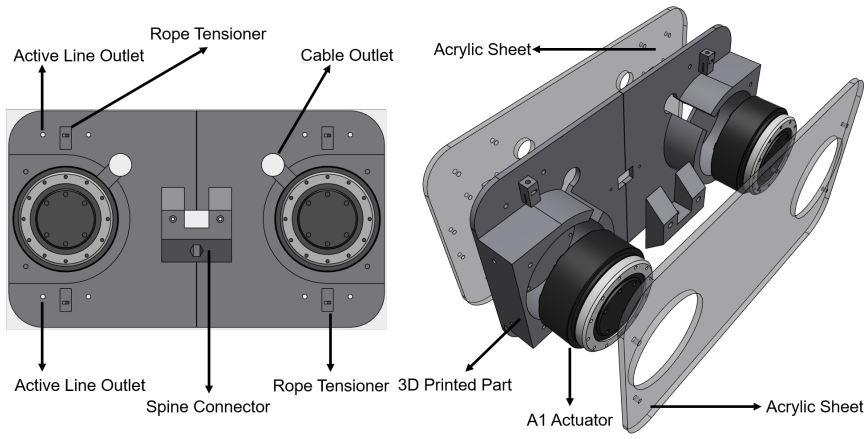
The overall design of the middle section of the body is similar to that of the front and rear segments, both of which utilize 3D-printed internal structures that are attached and encased in cut acrylic panels, as shown in Figure 3.15. Inside, there are four actuators that control the active fishing lines of the spine to generate motions, also space for the control circuits and battery. To facilitate easy installation and debugging, the mounting components for each motor are 3D-printed, and a hollow design is used in the direction of the motor's rotating surface for easier access.

Internal details The middle body segment has four parts, each of them contains an actuator, fishling line guide bearing and connector for connecting acrylic sheet. Figure 3.16 demonstrated the details of the left bottom part, the other three parts have differences in guide bearing only.

Fishing line path In terms of a wire-driven spine, the most important design is the fishing line routing, which should follow the force balance and reduce friction to achieve the best performance. Figure 3.17 illustrates the outlets of the fish line are



(a) Front body segment exploded view. The segment is composed of motors, 3D printed parts(grey parts) and acrylic sheets (transport grey parts).



(b) Rear body exploded segment view. The segment is composed of motors, 3D printed parts(grey parts) and acrylic sheets (transport grey parts).

Figure 3.14 Front and Rear body

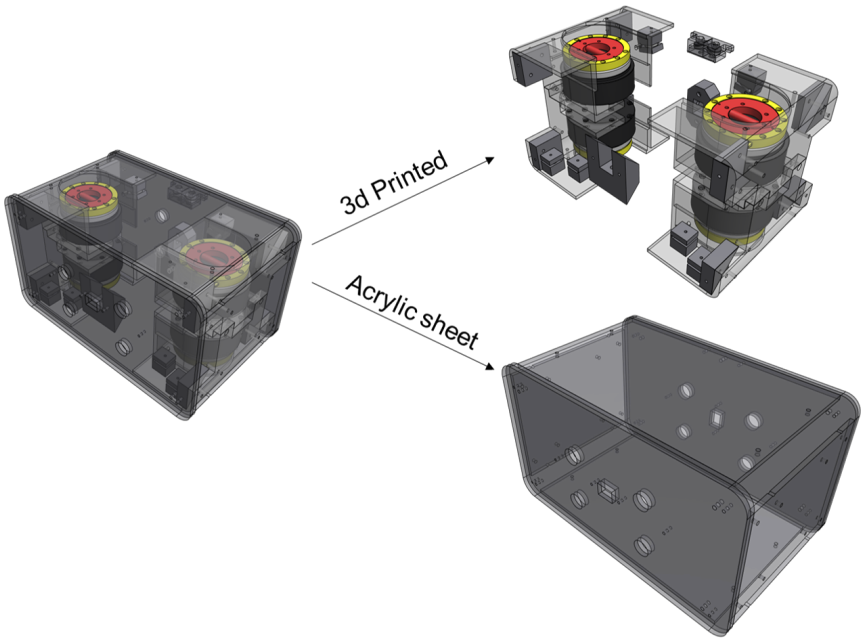


Figure 3.15 Middle body segment composition diagram. The outer shell is made of 8 acrylic sheets and other parts are 3D printed.

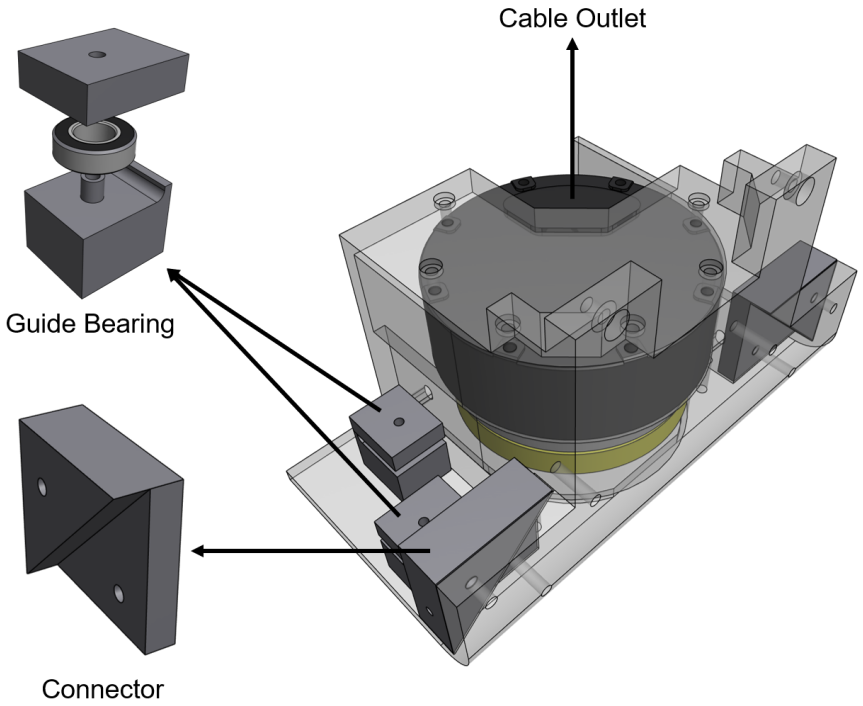


Figure 3.16 Lower left part of the middle body segment. The transparent grey part as a whole is 3D printed, other non-transparaent grey parts are used for mounting the pulleys and connecting acrylic sheets. The left and right parts are completed mirrored.

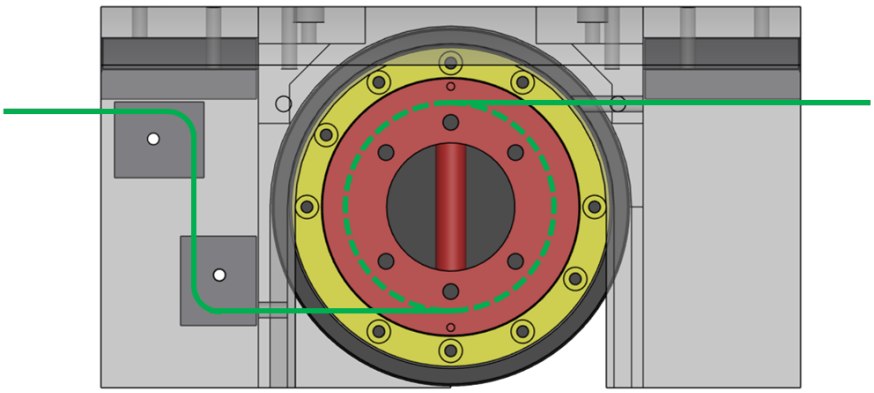


Figure 3.17 Inner fishing line path diagram. The green line represents the fishing line path. Green dash line is wound on the winch(red part). The yellow part is the external wall to avoid fishing line coming off.

symmetric and positioned at each end of the winch to balance the force. In addition, all fishing line holes are at the same level to avoid the friction of the holes on the lines. The fishing line on the left side of the figure is connected to the front body segment. To guide the fishing line to the front outlet and minimise wear and tear, two identical guide bearings modules were incorporated into the design. The outlet for the rear body segment is in a straight line with the outlet on the acrylic plate, so bearing is not required for guidance.

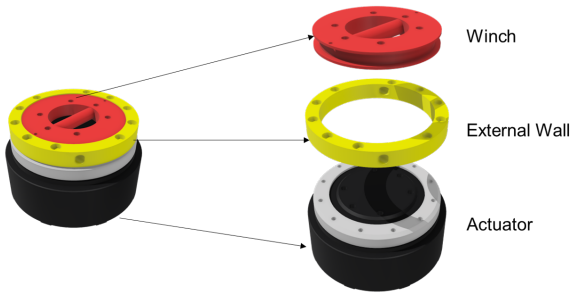


Figure 3.18 Active fishing line control motor. A fishing line is threaded through the bar on the winch and wound around it. The external wall has two outer that lead fishing line to the front and rear body segments respectively.

Winch To prevent the active fishing lines from wearing out, knotting, or slipping off during actuator rotation, a winch and ring as an external wall are also designed and mounted on the actuator, as shown in Figure 3.18. The central column of the winch is

specifically designed to secure the fish line, preventing loss of control due to slippage during operation.

Assemble The 3D-printed parts are individually printed, mounted on the acrylic plates, and secured with screws for optimal strength. It is installed as shown in Figure 3.19. The connector is printed with a fill rate of 100%.

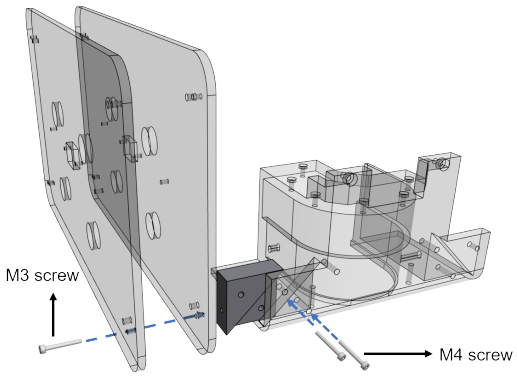


Figure 3.19 Acrylic sheets connection diagram. Two acrylic sheets connected to middle body segment through a connection part with one M3 screw and two M4 screws.

3.3.3 Legs

Leg design plays an important role in a legged robot in terms of its stability, flexibility and speed. In order to make quadruped robots adapt to varied terrains, not only control methods are needed to deal with external perturbation, but passive controls such as springs are also required to buffer external force interference caused by changing terrains. However, a compliant leg may also cause additional vibration that improves the model and control difficulty. A large range of leg motion and entire body workspace allows robots to adjust the appropriate posture for stability and flexible movements in various terrains. In addition to the workspace, the mass of a leg is another important factor in robot movement. The excessive mass of the knee joint increases the inertia of the leg, which is not conducive to the high-speed movement of the leg. According to the above aspects, TQbot leg design aims to maximise the leg range of motion for

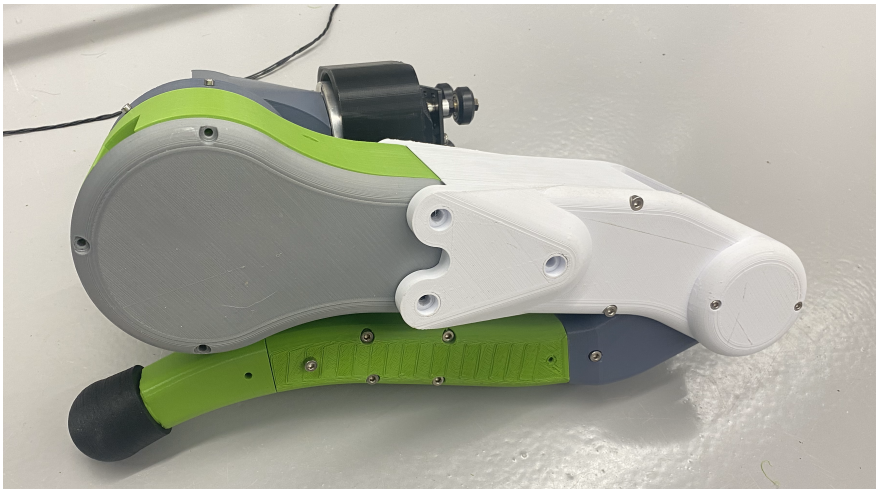


Figure 3.20 A physical leg made by 3D printing.

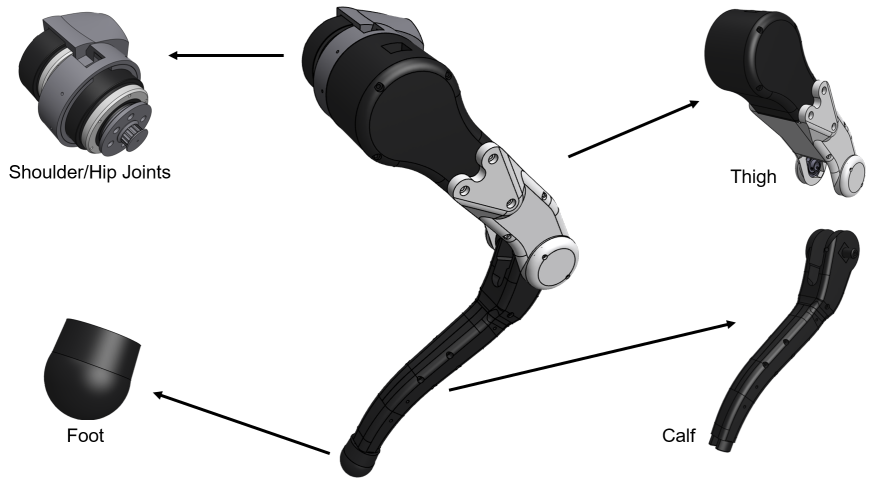


Figure 3.21 The whole leg consists of four parts: should-hip joint connector, thigh, calf and foot. All 3D printed and made of PLA materials except the feet are TPE.

rich motor skills and low inertial of the leg for easy control and high-speed locomotion. Hence, all TQbot motors are placed in the hip joint and the timing belt drive is used to reduce the moment of inertia of the leg. The physical leg made of 3D printed shown in Figure 3.20.

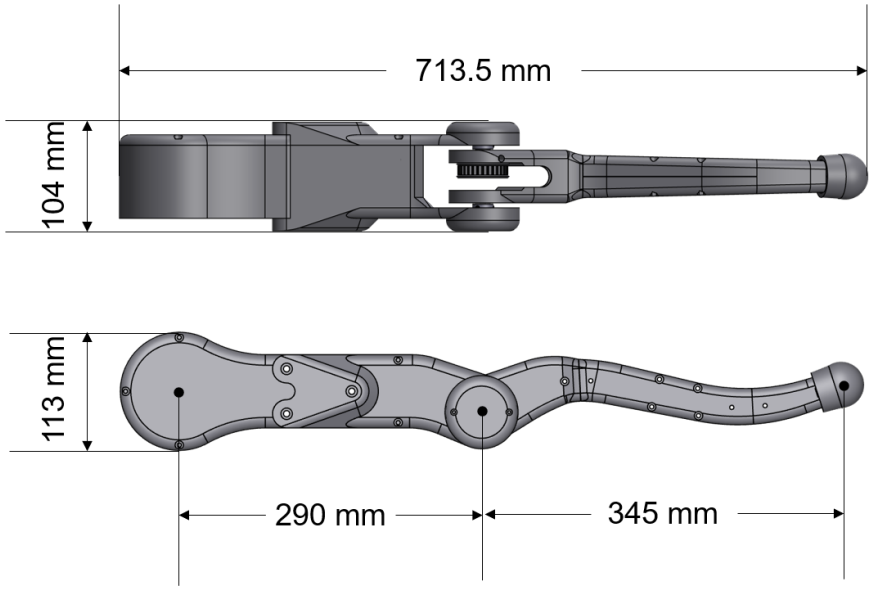


Figure 3.22 Left leg dimension diagram. The lower diagram includes joint distances. Right and left legs are the same size.

Figure 3.21 shows the design of the left leg which has 3 degrees of freedom and operates serially. The entire leg of the robot is divided into four parts: the shoulder-hip joint, thigh, calf, and foot. TQbot's legs are designed similarly to a dog's full knee configuration. Three actuators are positioned on the same horizontal plane to control the shoulder (abduction/adduction), hip, and knee joints of the robot, respectively. The knee joint is actuated by a timing belt inside the thigh, while the shoulder and hip joints are directly driven by actuators. The shoulder joints have a range of motion of 180 degrees, allowing the robot to potentially use its legs to withstand lateral impact forces. The thigh links of TQbot are designed with an offset between the hip and shoulder joints, enabling full rotation of the hip joints. The knee joints provide an additional 2:1 reduction ratio between the hip and knee joints, increasing the torque at the knee. This design reduces the transmission speed but ensures sufficient torque

at the knee joint. To adjust the tension of the timing belt, a simple and detachable tensioner is also designed to be mounted on the inner thigh. The calf design allows for a 240-degree range of motion, enabling the robot to adjust its knee joint angles to perform different gaits.

For ease of installation, pulleys with flanges are used at the knee joint. Since the robot's legs are made of 3D-printed polylactic acid (PLA) material, the two knee pulleys are embedded at the end of the thighs and secured with two M3 screws. An 80 cm long M6 screw passes through the centres of the two pulleys, ensuring that the knee joints have sufficient strength to support the entire body and making it difficult for the pulleys to detach.

Shoulder and Hip Joints

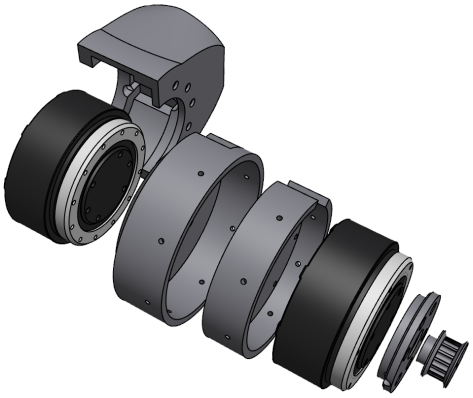


Figure 3.23 Shoulder-Hip joint exploded view. The joint has four 3D printed parts and one pulley.

Figure 3.23 shows the exploded view of the shoulder and hip joints. It should be noted that the motor corresponding to the shoulder joint is not depicted in the diagram as it is mounted on the body. To maximise the range of motion, the shoulder joint is designed to rotate 360 degrees. However, the overall range of leg motion at the roll orientation is limited by the body segment, resulting in an effective extension and contraction range of approximately ±90 degrees for the shoulder joint. Additionally, the hip joint, which connects to the leg, is designed with outlets to route control and

power cables to the knee joint motor. For ease of installation, the components attached to the hip joint motor and those on the knee joint motor are designed separately and assembled using screws. The pulley on the knee motor has a specification of 15 teeth, 5M, and is mounted on the motor using a 3D-printed circle pad. To prevent falling and sliding, the pulley is processed and can be assembled on the circle pad with two M3 screws.

Thigh

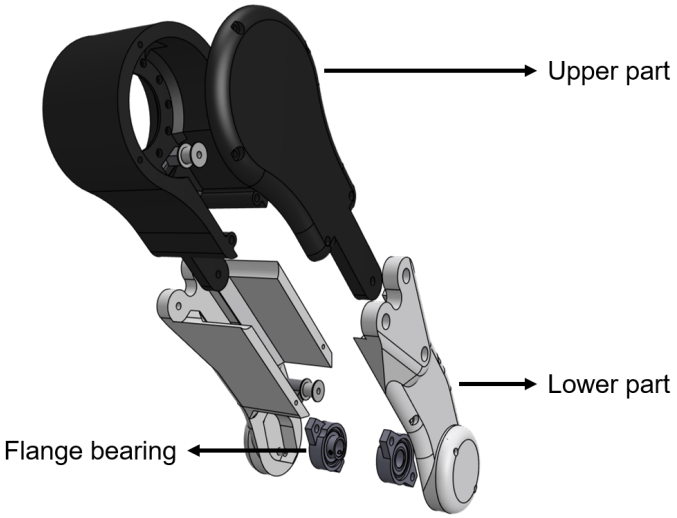


Figure 3.24 Thigh exploded view. The thigh is divided into upper(black) and lower(white) parts, each consisting of a shell and a cover.

To provide sufficient space for the timing belt and other cables, the interior of the thigh part is hollow and has upper and lower parts. The two parts are divided into a cover plate and a housing. The cover plate is attached to the housing using M6 and M4 screws. This design facilitates the installation and adjustment of the internal components within the legs, while the 10mm thick outer wall ensures that the legs are strong enough to support most of the robot's leg movements. There are two tensioners—one on the upper part and one on the lower part of the leg—to adjust the tension of the timing belt, as shown in Figure 3.25. The upper tensioner is designed to be adjusted with screws, while the lower tensioner can be adjusted by substituting

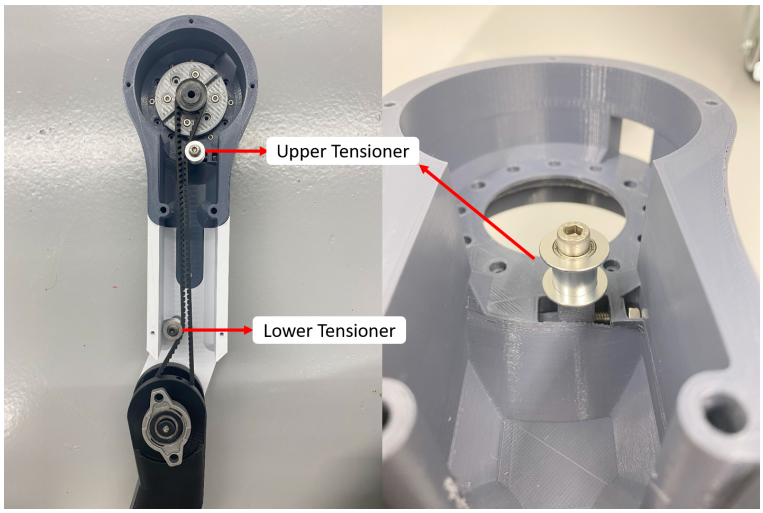


Figure 3.25 Tensioners for adjusting timing belt in the thigh.

different tensioning parts. For ease of installation, a flange bearing is installed at the lower part of the thigh, with an $80mm$ long M6 screw at the rotation centre of the bearing to ensure it has sufficient strength to support the entire body, though this is not shown in the figure.

Calf

The calf is also a hollow design in the middle, which reduces mass while maintaining sufficient strength. The 30-tooth pulley is also processed and installed at the knee joint of the calf. Unlike the design of the thigh part, the calf has three parts and is installed with M4 screws. The upper and lower parts are a whole, and the middle is divided into left and right parts. The shell thickness is $5mm$.

3.3.4 Feet

For future research and development, the robot's feet are designed to be detachable from the legs, as shown in Figure 3.27. The feet are made of flexible 3D-printed material, TPE, which provides elasticity to cushion the impact when the foot contacts the ground. The design of the robot's foot combines spherical and conical structures.

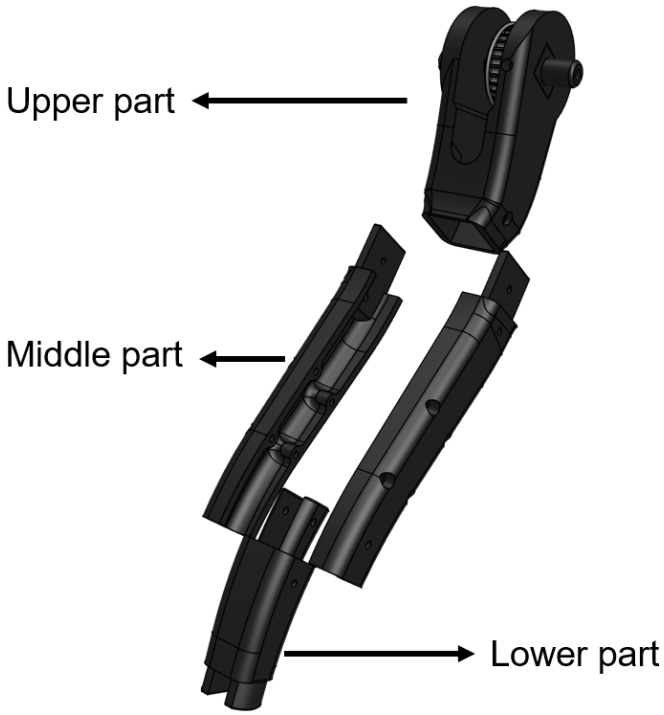
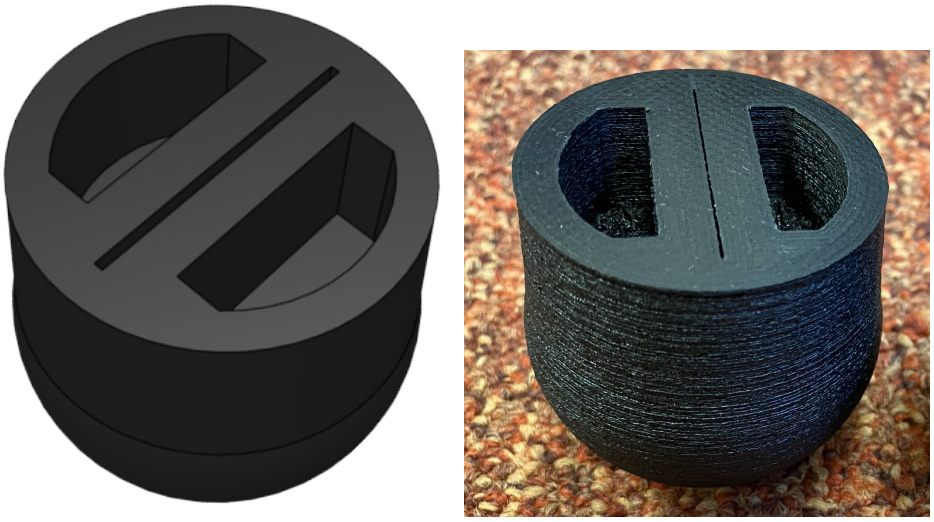


Figure 3.26 Calf exploded view. It has three parts and connected by M4 screws. The middle part has left and right parts while both the upper and lower parts each are a single part.



(a) Foot model (b) 3D printed foot

Figure 3.27 Foot model and 3D printed. The $1mm$ gap in the centre for the pressure sensor, and mounting holes on both sides.

The spherical part of the foot has a diameter of 44mm, and its wide contact angle with the ground enables a variety of leg movements. Additionally, the 3D-printed foot features a rough surface to increase friction with the ground. A force-sensitive resistor (FSR) is integrated into each foot to gather contact information with the ground and assess the posture of the legs. The purchased FSR can withstand pressures ranging from 0 to 30 kg and changes its resistance based on the applied pressure. This allows it to not only detect contact between the foot and the ground, but also measure the amount of pressure exerted, which is reflected in its output voltage.

3.4 Electronics

In terms of robot development, highly integrated electronic products can greatly improve development efficiency, and an electronic system with powerful computing power is essential for locomotion control performance. Following the above two points, TQbot electronic system consists 16 highly integrated actuators, an inertial measurement unit (IMU), a high-performance onboard personal computer (PC) and 3 battery modules. Figure 3.28 shows the complete communication and connection of the electronic system.

3.4.1 Onboard Personal Computer

For robot locomotion control and sensor data processing, a high-performance onboard PC, Jetson Orin Nano, was used. It is a popular computing module in robotics development designed by NVIDIA. The purchased version is 8GB RAM model using LPDDR5 memory. It is based on the ARM64 platform and comes with the Ubuntu 20.04 operating system. It offers high performance while consuming only 7 to 15 watts of power, making it suitable for battery-powered devices. On the other hand, the Jetson Orin Nano is equipped with 3 GPIO interfaces, 2 RS232 interfaces, 2 USB 2.0 interfaces, 1 HDMI interface, and a gigabit Ethernet port. This makes it easy to connect and interact with various external devices and sensors for data exchange. In

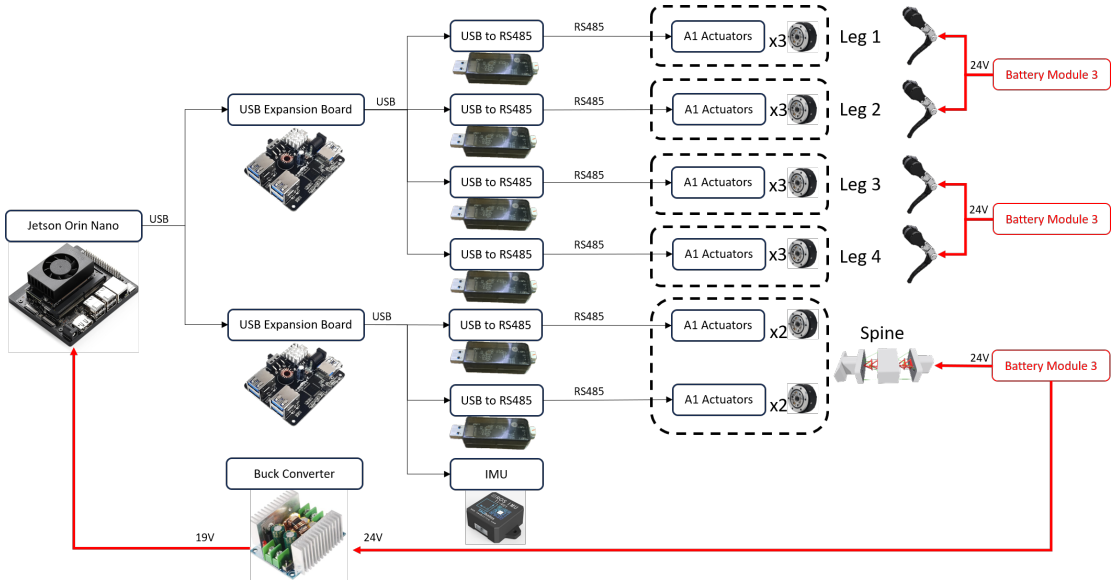


Figure 3.28 Diagram of electronics architecture. Red lines represent power supply and black lines represent communication.

this work, the Jetson Orin Nano is used to control the real-time computing and data processing of 16 actuators and an IMU at a frequency of $300HZ$.

3.4.2 Actuators

The actuator is the key component of a quadruped robot platform, which can directly affect locomotion. The leg and spine of TQbot have three degrees of freedom and are driven by a high torque density actuator A1 from Unitree [168]. It is a commercial-grade actuator dedicated to robot joints and is used in the Unitree A1 quadruped robot. A1 actuator is a permanent magnet synchronous motor(PMSM) consists of a stator connected to the motor housing and a rotatable rotor with permanent magnets. The reduction ratio $9.1:1$ allows the A1 actuator to reach a maximum torque of $33.5N \cdot M$, which is enough to support the various locomotion of TQbot. Its maximum angular velocity is $21rad/s$, which theoretically allows TQbot to move forward at a speed of $3m/s$ and rotate its spine. For safety reasons, the maximum speed of the spine will be limited to $1rad/s$ (approximately $57.3degrees/s$). The A1 actuator control frequency supports up to $1000Hz$, but $300hz$ is used for control in this robot

Control

The control method of A1 actuator is based on Field-Oriented Control (FOC). It can achieve precise control by changing the direction and magnitude of the stator's magnetic field, which is to control the currents and their directions in the three windings of the stator. However, the underlying control algorithm has been encapsulated internally within APIs provided by Unitree, which can greatly accelerate the development progress of the entire robot.

The specific control method is called hybrid control:

$$\tau = \tau_f + k_p \cdot (q_{des} - q) + k_d \cdot (dq_{des} - dq) \tag{3.1}$$

where the control parameters are

- τ_f: feedforward torque of the motor rotor, measured in $N \cdot M$
- q_{des}: expected angular position of the motor rotor, in rad
- qd_{des}: expected rotor angular velocity of the motor rotor. in rad/s
- k_p: stiffness coefficient
- k_d: damping coefficient

and the state parameters are

- τ: expected output torque of the motor rotor, in $N \cdot M$
- q: current rotor angle position of the motor rotor, in rad
- dq: current rotor angle velocity of the motor rotor, in rad/s

In real control, a PD controller is used to generate the torque output τ according to the deviation of the motor at the output position.

3.4.3 IMU

The IMU used for TQbot is HFI-A9, as shown in Figure 3.29. It is a high-precision 9-axis Inertial Measurement Unit (IMU) that integrates a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer. It features an advanced fusion algorithm, providing accurate attitude, acceleration, and angular velocity data in both dynamic and static conditions. With a control frequency of up to 200Hz or higher, it is ideal for applications requiring high-precision attitude and motion control, such as drones and robots. Additionally, the HFI-A9 supports multiple communication interfaces, making it easy to integrate into various systems. Its detailed specification is shown in Table 3.2.

Table 3.2 IMU: HFI-A9 specification

Parameters	**Values**
Power Supply	4.7~5.5V
Current Consumption	5.0mA
Communication Method	USB Serial
Output Data	(Gyroscope + Accelerometer + Magnetometer), Quaternion
Gyroscope Range	±2000°/s
Accelerometer Range	±8g
Magnetometer Range	±1.3Gauss
Angle Accuracy	Static 0.1° Dynamic 0.5°
Transmission Rate	300Hz
Baud Rate	921600
ROS Topics	/handsfree/imu /handsfree/mag

Figure 3.29 9-axis IMU: HFI-A9

3.4.4 Battery and Power System

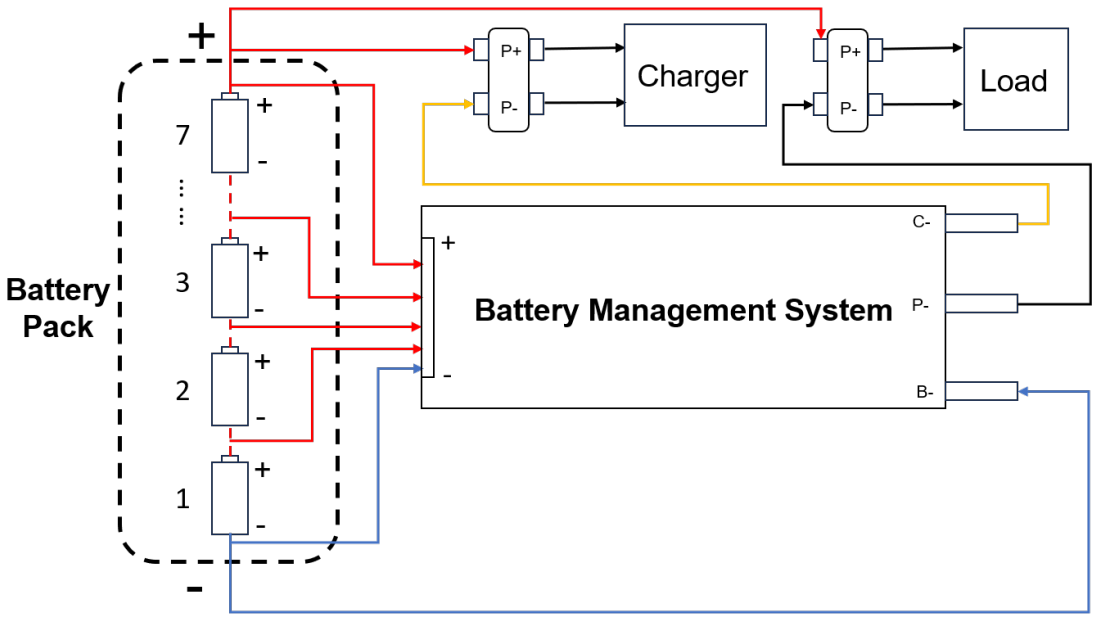


Figure 3.30 Battery module connection diagram. The red lines are positive wires and the blue line is a negative wire. The yellow line is a charger negative wire.

TQbot is powered by 3 battery modules that are placed in the front, middle, and rear of the body to maintain balance. Each battery module includes 7 series and 1 parallel Lithium Ion battery with a battery management system (BMS). The battery module is shown in Figure 3.30. A purchased BMS can afford a maximum of 80A current. Each battery pack is made by welding seven ternary lithium batteries together in series using a welding machine. The specifications of each battery are shown in Table 3.3.

3.5 Control Architecture

The whole body control of the robot is an important part. This section describes the overall control method of the robot. In order to coordinate the robot's spine and leg movements, a model-free control method is used for the whole body of the robot in

Figure 3.31 Battery module. The top shows the internal battery pack, and the bottom picture is a complete battery module

Table 3.3 Samsung 25R 18650 battery specification

Attribute	**Specification**
Brand	Samsung SDI
Model	25R, INR18650-25R (25RM)
Size	18650
Chemistry	INR(Lithium, Nickel, Round cell)
Nominal Capacity	2500mAh
Nominal Voltage	3.6V
Discharge	20A Max Continuous
Positive	Flat
Protected	NO
Rechargeable	Yes
Dimensions	18.22mm x 64.80mm
Weight	43.71g

this paper. CPG is used to output trajectories for all joints, and PD controllers are used to make each joint track these trajectories for control.

3.5.1 Single Leg Control

In this work, CPG is used as a basic gait generator to generate execution angles for each joint and thus generate whole-body motion. Therefore, forward kinematics and inverse kinematics are mainly used for the leg motion control of this robot. Other legs' kinematics and inverse kinematics models can be obtained by adjusting DH parameters as they have symmetry structures and configuration.

Kinematics

In general, the construction process of forward kinematics is as follows:

- Establish Coordinate Systems: Define a local coordinate system for each link and joint of the robot. Typically, the origin and axis directions are chosen according to the DH convention.
- Determine DH Parameters: Identify the four DH parameters based on the relative positions and orientations of the links and joints.
- Construct Transformation Matrices: Use the DH parameters to construct homogeneous transformation matrices that convert from one joint coordinate system to the next.
- Multiply Transformation Matrices: Multiply all transformation matrices to obtain the overall transformation matrix from the base coordinate system to the end effector coordinate system, thus determining the robot pose.

Coordinate Systems

To establish the single leg coordination, the following convention rules should be followed:

- Assign the Z_i axis pointing along the i_{th} joint axis. For a rotational joint, this is the axis around which the rotation occurs. For a prismatic joint, this is the direction of the linear motion.

- Assign the X_i axis pointing along the common perpendicular of both the Z_i and Z_{i-1} axes., or, if the two axes intersect, assign X_i to be normal to the plane containing the Z_i and Z_{i-1} axes.

- Assign the Y_i axis to complete a right-hand coordinate system according to X_i axis and Z_i axis.

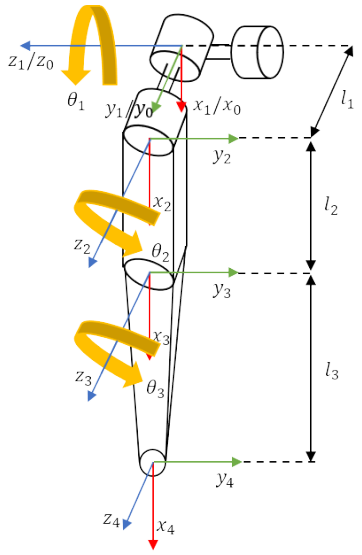


Figure 3.32 Illustration of a left front leg showing its kinematic configuration

DH Parameters

In conventional robot control, Denavit-Hartenberg (DH) method is a standardised method for describing the geometric relationship between the various links of a robot, describing and analyzing the relative motion and position relationship between the various links of a robot. The DH notation is widely used to establish reference coordinate frames in real robot applications. It has four variables, two for describing the link

itself and two for describing the connection to a neighbouring link. In a defined D-H coordinate frame, the detailed descriptions of these four parameters are as follows:

- a_i:, link length, the distance from Z_i to Z_{i+1} measured along X_i
- α_i:, link twist, the angle from Z_i to Z_{i+1} measured about X_i
- d_i:, joint distance, the distance from X_{i-1} to X_i measured along Z_i
- θ_i: joint angle, the angle from X_{i-1} to X_i measured about Z_i

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	$-90°$	l_1	θ_2
3	l_2	0	0	θ_3
4	l_3	0	0	0

Table 3.4 DH parameters of front left leg in TQbot

Transformation Matrices

According to the DH parameters, the transformation of Frame $i-1$ to i can be obtained:

$$ {}^{i}_{i-1}T = R_X(\alpha_{i-1})D_X(a_{i-1})R_Z(\theta_i)D_Z(d_i) \tag{3.2} $$

where

- $R_X(\alpha_{i-1})$: A rotation of α_{i-1} degrees about the translated x_i axis

$$ \boldsymbol{R}_x(a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_i & -s\theta_i & 0 \\ 0 & s\theta_i & c\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3} $$

- $D_X(a_{i-1})$: A translation about the x_i axis by a distance a_i

$$\boldsymbol{D}_X(a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

- $R_Z(\theta_i)$: A rotation of θ_i degrees about the translated z_i axis

$$\boldsymbol{R}_Z(\theta_i) = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & -s\theta_i & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

- $D_Z(d_i)$: A translation about the z_i axis by a distance d_i

$$\boldsymbol{D}_Z(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

Thus, a Denavit-Hartenberg transformation can be obtained

$${}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

As a result, the conversion formula between the coordinate systems of the individual joints of the left front leg can be easily obtained:

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8} $$

$$ {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9} $$

$$ {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & l_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10} $$

$$ {}^{3}_{4}T = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11} $$

$$ {}^{0}_{4}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \tag{3.12} $$

Thus,

$$
{}^{0}_{4}T = \begin{bmatrix} c\theta_1 \cdot c\theta_{23} & -c\theta_1 \cdot s\theta_{23} & -s\theta_1 & l_3 \cdot c\theta_1 \cdot c\theta_{23} - l_1 \cdot s\theta_1 + l_2 \cdot c\theta_1 \cdot c\theta_2 \\ s\theta_1 \cdot c\theta_{23} & -s\theta_1 \cdot s\theta_{23} & c\theta_1 & l_3 \cdot s\theta_1 \cdot c\theta_{23} + l_1 \cdot c\theta_1 + l_2 \cdot s\theta_1 \cdot c\theta_2 \\ -s\theta_{23} & -c\theta_{23} & 0 & -l_3 \cdot s\theta_{23} - l_2 \cdot s\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}
$$

Inverse Kinematics

Based on the transformation matrix obtained from forward kinematics, the inverse kinematics of the single-leg can be solved using algebraic methods, that is, the angles of each joint can be calculated based on the foot end position. First of all, a trick of multiplying the inverse matrix of the transformation on both sides of the equation at the same time is applied to solve for the separated variables. By transposing 3.8, Equation 3.12 can be written as

$$
({}^{0}_{1}T)^{-1}{}^{0}_{4}T = {}^{1}_{4}T \tag{3.14}
$$

which is

$$
\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{1}_{4}T \tag{3.15}
$$

Equating the $(2, 4)$ elements from both sides of equation 3.15:

$$
-s_1 p_x + c_1 p_x = l_1 \tag{3.16}
$$

To solve this equation, trigonometric substitutions can be used:

$$\begin{cases} p_x = \rho \cos \phi \\ p_y = \rho \sin \phi \end{cases} \tag{3.17}$$

where

$$\begin{cases} \rho = \sqrt{p_x^2 + p_y^2} \\ \phi = Atan2(p_y^2, p_x^2) \end{cases} \tag{3.18}$$

Substituting 3.17 into 3.16 can get:

$$c_1 s_\phi - s_1 c_\phi = \frac{l_1}{\rho} \tag{3.19}$$

According to the difference-of-angles formula, 3.19 can be written as:

$$\sin(\phi - \theta_1) = \frac{l_1}{\rho} \tag{3.20}$$

Hence,

$$\cos(\phi - \theta_1) = \pm\sqrt{1 - \frac{l_1^2}{\rho^2}} \tag{3.21}$$

and so

$$\phi - \theta_1 = Atan2(\frac{l_1}{\rho}, \pm\sqrt{1 - \frac{l_1^2}{\rho^2}}) \tag{3.22}$$

Since TQbot's legs are configured with backward knee joints, the plus-or-minus sign here is always plus. Then, the final solution for θ_1 can be obtained by the following equation:

$$\theta_1 = Atan2(p_y, p_x) - Atan2(l_1, \sqrt{p_x^2 + p_y^2 - l_1^2}) \tag{3.23}$$

The solution process of θ_3 is similar to that of θ_1, equating both the $(1, 4)$ elements and the $(3, 4)$ elements from the two sides of Equations 3.15, then obtained:

$$\begin{cases} c_1 p_x + s_1 p_y &= l_3 \cdot c_{23} + l_2 \cdot c_2 \\ -p_z &= l_3 \cdot s_{23} + l_2 \cdot s_2 \end{cases} \tag{3.24}$$

Then, squaring equation 3.24 and 3.16 and add the resulting equations:

$$p_x^2 + p_y^2 + p_z^2 = l_3^2 + l_2^2 + l_2 \cdot l_3 \cdot (c_2 \cdot c_{23} + s_2 \cdot s_{23}) + l_1^2 \tag{3.25}$$

Using angle sum and difference identities, 3.25 can be written as:

$$c_3 = \frac{p_x^2 + p_y^2 + p_z^2 - l_3^2 - l_2^2 - l_1^2}{2 \cdot l_2 \cdot l_3} \tag{3.26}$$

To get the solution for θ_3, the same kind of trigonometric substitution can be used:

$$\theta_3 = Atan2(s_3, c_3) \tag{3.27}$$

where

$$s_3 = \pm\sqrt{1 - c_3^2} \tag{3.28}$$

To solve the θ_2, considering rewrite Equation 3.12 to a new equation with all known functions and θ_2 on the left side:

$$({}^{0}_{3}T)^{-1}{}^{0}_{4}T = {}^{3}_{4}T \tag{3.29}$$

where

$${}^{0}_{3}T^{-1} = \begin{bmatrix} c\theta_1 \cdot c\theta_{23} & s\theta_1 \cdot c\theta_{23} & -s\theta_{23} & -l_2 \cdot c\theta_3 \\ -c\theta_1 \cdot s\theta_{23} & -s\theta_1 \cdot s\theta_{23} & -c\theta_{23} & l_2 \cdot s\theta_3 \\ -s\theta_1 & c\theta_1 & 0 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.30}$$

Let the elements (1,4) and (2,4) of Equation 3.29 equal:

$$\begin{cases} c_1c_{23} \cdot p_x + s_1c_{23} \cdot p_y - s_{23} \cdot p_z - l_2c_3 = l_3 \\ -c_1s_{23} \cdot p_x - s_1s_{23} \cdot p_y - c_{23} \cdot p_z + l_2s_3 = 0 \end{cases} \tag{3.31}$$

And further obtain:

$$s_{23} = \frac{l_2s_3 \cdot a - (l_3 + l_2c_3) \cdot p_z}{a^2 \cdot p_z^2} \tag{3.32}$$

$$c_{23} = \frac{(l_3 + l_cc_3)a + l_2s_3 \cdot p_z}{a^2 \cdot p_z^2} \tag{3.33}$$

where

$$a = c_1p_x + s_1p_y \tag{3.34}$$

which is

$$\theta_{23} = Atan2(s_{23}, c_{23}) \tag{3.35}$$

Finally, the θ_2 can be obtained by:

$$\theta_2 = Atan2(s_{23}, c_{23}) - \theta_3 \tag{3.36}$$

3.5.2 Spine Control Method

Controlling a spine for quadruped robots includes motion control and leg-spine gait coordination. This work focuses on the former where it can be used for an entire quadruped robot as a subsystem in subsequent development. Since the rigid elements are not in direct contact and are suspended in space completely by wires, the tensegrity spine needs to preserve a pretensioned state to maintain the stability of the structure as the foundation for its active control. Thus, a pretension ratio-based method is proposed to control the spine by adjusting the active wire length.

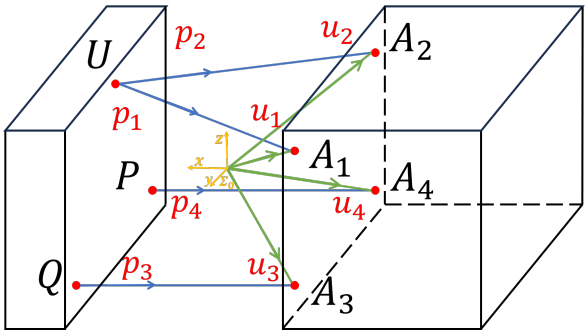


Figure 3.33 The geometric model of the front half of the proposed bionic tensegrity spine. The left and right cuboids represent the front and middle body segments, respectively. The origin Σ_0 is at the geometric centre between the two body segments.

Kinematics and Inverse Kinematics

The control algorithm of the spine is developed according to parallel wire-driven robot control [169]. Since the four actuators in the middle body part control the front and

rear body simultaneously and symmetrically, only one kinematic model is required to control both parts of the entire spine. Figure 3.33 shows the kinematic structure of the front part of the spine. Each torque w_i applied on the joint exerted by the i^{th} wire which induces its rotation around the origin is given by:

$$w_i = \left[u_i \times p_i \right] \quad (i \in \mathbb{Z}, 1 \leq i \leq 4) \tag{3.37}$$

where p_i is a unit vector that denotes the wire direction, and the vector u_i denotes the position of the wire attachment points on the middle body. The operator $\times$ represents a cross-product. Following the naming of the points illustrated in Figure 3.33, the directional unit vector is calculated from the following equation:

$$p_i = l_i \left\| l_i \right\|_2^{-1} \tag{3.38}$$

with

$$l_i = U - A_i, \quad (i = 1, 2) \tag{3.39}$$

$$l_3 = Q - A_3 \tag{3.40}$$

$$l_4 = P - A_4 \tag{3.41}$$

where l_i denotes the i^{th} wire, A_i is the outlet attachment point of wire in the middle body and "$\|\quad\|$" denotes Euclidean norm. Assuming the middle body is fixed, the coordinates of points U, P and Q can be obtained by Inverse Kinematic based on the expected posture:

$$\begin{bmatrix} U \\ P \\ Q \end{bmatrix} = \boldsymbol{R}_{YXZ} \begin{bmatrix} U_0 \\ P_0 \\ Q_0 \end{bmatrix} \tag{3.42}$$

with

$$\boldsymbol{R}_{YXZ} = \begin{bmatrix} c_1c_3 + s_1s_2s_3 & c_3s_1s_2 - c_1s_3 & c_2s_1 \\ c_2s_3 & c_2c_3 & -s_2 \\ c_1s_2s_3 - c_3s_1 & c_1c_3s_2 + s_1s_3 & c_1c_2 \end{bmatrix}$$

where U_0, P_0 and Q_0 are coordinates at the neutral posture. $\boldsymbol{R}_{YXZ}$ is the rotation matrix with Euler angles as inputs. c and s are the abbreviations for *cos* and *sin* respectively where the subscripts denote the corresponding rotation angles in a sequential order.

To establish a stable posture of the spine, its force equilibrium is given by:

$$WF_t + F_d = 0 \tag{3.43}$$

where the matrix $W = [w_1w_2...w_m]$ is the wire structure matrix, F_t is the tension forces in the wires that hold the spine in position, and F_d is the external torque applied to the spine, i.e., the desired output torque of the spine at a given posture. It should be noted that the robot spine differs from other common parallel robots that it does not have translational DOF. Thus F_t can be simplified to $[\tau_1\tau_2\tau_3\tau_4]^T$.

For a wire-driven robot with n DOF, in general, at least $m = n + 1$ wires are required to fully control the motion due to the fact that wires can only pull and not push on an object. Here, an important concept "Vector Closure" is introduced to calculate the wire tension force [169].

Vector Closure. *In an n-dimensional space, a set of vector V is a Vector Closure if and only if V has at least n+1 vectors*$(v_1v_2...v_{n+1})$ *satisfying the following two conditions.*

(1) *Each set of n vectors in n+1 vectors is linearly independent.*

(2) *A vector* $\beta = (\beta_1\beta_2...\beta_{n+1})^T$ *exists, which satisfies*

$$V\beta = \sum_{i=1}^{n+1} v_i\beta_i = 0, \quad (\beta_i > 0 \text{ or } \beta_i < 0) \tag{3.44}$$

The wire structure matrix W is the Cartesian representation of vector V and the internal force vector corresponds to vector β. The conditions of Vector Closure ensure that the tension on each wire is positive and can generate a resultant force vector of any magnitude and direction. As full-body dynamics of the robot is not involved in this paper, the spine control can thus be simplified in that F_d is set to zero and it only calculates the forces for pretension in the wire. Thus, the relationship between wire structure and wire tension forces can be written as:

$$[\tau_1\tau_2...\tau_n]^T = -N^{-1}w_{n+1}\tau_{n+1} \tag{3.45}$$

where the matrix N denotes $[w_1w_2...w_n]$. The matrix N can be guaranteed as non-singularity if the spine motion satisfies each w_i vector is linearly independent. Then, the internal force is calculated as follows:

$$F_t = \begin{bmatrix} -N^{-1}w_{n+1}\tau_{n+1} \\ \tau_{n+1} \end{bmatrix} \tag{3.46}$$

The τ_{n+1} is set to positive values resulting in all other elements becoming positive due to the Vector Closure condition being satisfied. As a result, the spine can be tensed in any posture within the movement workspace.

Accurate modelling of the tensegrity spine is challenging. An approach based on the derived form of Hooke's Law is employed to convert the tension forces to the length of each wire that needs to be stretched according to a pretension ratio. First, getting the

pretension ratios, ρ_i for the active wire:

$$\rho_i = \frac{\|\tau_i\|}{\|\tau_{base}\|}\rho_{base}, \quad (i = 1, 2, 3, 4) \tag{3.47}$$

where τ_{base} is the tension force on the wire for a given pretension ratio ρ_{base}. In this way, the general pretension of the spine can be adjusted by varying ρ_{base}. The calculation for the length of the wire to be changed can thus be given by:

$$\Delta l_i = l_i \rho_i \tag{3.48}$$

3.6 Summary

This chapter provides a detailed overview of the design, implementation, and control of TQbot, featuring two novel active spines. TQbot is a unique quadruped robot with a wire-driven spine composed of a tensegrity structure. It is used to explore how a flexible, actuated spine can improve speed, stability, and energy efficiency in legged robots. In contrast to existing quadruped robots with rigid connections, TQbot employs a tensegrity spine with a CRPM configuration, combining both active and passive elements to emulate the biological structures of mammals.

The design process for TQbot required careful consideration of the size and strength of materials, primarily relying on 3D printing and laser-cut acrylic sheets. The modular approach allowed for rapid updates during development, ensuring flexibility in iterations while maintaining the structural integrity of the robot.

Mechanically, TQbot's spine and body are made up of two main configurations. The first tensegrity spine features passive parts constructed from 3D-printed components and active elements driven by tensioning fishing lines, providing rotational freedom in three directions: roll, pitch, and yaw. The second configuration is a more advanced

version, incorporating springs and stainless steel wires for greater strength and range of motion. The robot's legs are designed with three degrees of freedom per leg, maximising its range of motion and reducing inertia for high-speed movement.

The control architecture for TQbot is equally significant, relying on a model-free control approach using CPG for gait generation and the coordination of leg-spine motions. The kinematic and inverse kinematics models of the spine and legs are established, ensuring that both passive and active components work in unison for efficient and adaptive locomotion.

In general, this chapter presents a comprehensive exploration of the design challenges and solutions encountered while developing TQbot, highlighting its novel spine design and integration with legged locomotion. This foundation supports the broader goal of improving the performance of quadrupedal robots in real-world applications, especially in unstructured or challenging terrains.

Chapter 4

Central Pattern Generator Design and Implementation

4.1 Introduction

In the case of robots with a rigidly connected spine, its motion can be effectively generated by solving inverse kinematics. However, applying this method to a flexible spine presents challenges, making it difficult to precisely manipulate the spine and naturally coordinate the motions of the spine and the legs. In contrast, using Central Pattern Generator (CPG), as a model-free method in robotics that is inspired by a neural circuit commonly found in vertebrate and invertebrate animals, has a natural advantage in coordinating and synchronising spine and leg movements. While many CPG models have been developed to generate quadruped robot locomotion, only a few studies have explored the coordination between the legs and spine.

In this chapter, a completed process for designing a CPG model will be presented in detail since there is not yet a well-established design method. The fundamentals of CPG models are presented and described in Section 4.2, which includes the type of oscillator and its mathematical formula. Section 4.3 introduced an internal feedback mechanism to independently adjust the frequency of the stance and swing phase in a gait. In Section 4.4, the type and topology of couplings are described and a reference oscillator is used to simplify the parameter adjustment. Various gaits and the methodology for adjusting the phase lag parameters are given in Section 4.5. In addition, Section 4.6 proposed a novel CPG model with learning sensory feedback to generate dynamic trot gait on rough terrains. Section 4.7 summaries the work described in this chapter and considers how it will be used in subsequent chapters.

4.2 Definition of CPG model with Phase Oscillators

Different mathematical models of CPGs were discussed in Chapter 2, where they were used to generate gaits for quadruped, salamander, and other robots. However, some CPG models, such as those in [87, 170], are complex, making the integration of new sensory feedback challenging. To address this, the phase oscillator model was chosen

as the basis for studying sensory feedback, as its parameters correspond clearly to gait parameters, and integrating sensory feedback is relatively straightforward. The definition and fundamentals of this model, along with the significance of its various parameters, will be presented next.

4.2.1 Phase

Phase oscillator is also called Kuramotor oscillator, which is a first-order system of differential equations used to study systems of phase oscillators. It was proposed by Kuramotor in 1975, and has the following form:

$$\dot{\phi} = v_i + \sum_{j}^{N} \omega_{ij} f(\phi_j - \phi_i) \tag{4.1}$$

To explain the notation a little, each oscillator ϕ_i has a preferred frequency ω_i, but is biased away from this by its interaction with other oscillators. This interaction f is periodic in the difference of the phases of the oscillators concerned, and is zero if the phases are identical. This leads one to believe that f is probably a polynomial or power series in terms of $sin(\phi_j - \phi_i)$ which it can be, but for the sake of simplicity, it is typically taken to be only the lowest order of such term. The phase oscillator can correspond in gait either to the output of each joint or to the output of a foot end. Inverse kinematics is used to solve for the angle of each joint.

4.2.2 Amplitude and Offset

In addition, amplitude and offset are control parameters to generate various gaits. It is generally expressed in CPG using a first or second-order differential equation, for

example:

$$\dot{r_i} = a_r(R - r_i) \tag{4.2}$$

$$\dot{x_i} = a_x(X - x_i) \tag{4.3}$$

or

$$\ddot{r_i} = a_r(\frac{a_r}{4}(R - r_i) - \dot{r_i}) \tag{4.4}$$

$$\ddot{x_i} = a_x(\frac{a_x}{4}(X - x_i) - \dot{x_i}) \tag{4.5}$$

R and X are control parameters determined by human assistance, which are the desired amplitude and offset during moving. r_i and x_i are state variables that represent the amplitude and offset of $i - th$ oscillator. a_r and a_x are constant positive gains that represent the convergence speed of amplitude and offset. The first order differential equation characterizes the velocity $\dot{r_i}$ as a linear function of the distance difference $R - r_i$, whereas the second equation provides a more intricate dynamic model, where the acceleration $\ddot{r_i}$ is modulated not only by the distance difference $R - r_i$ but also by a feedback mechanism involving the current velocity $\dot{r_i}$. This indicates a more complex interdependence between velocity and acceleration in the system.

4.2.3 Set-Point

Finally, the complete CPG contains an output function that obtains all control parameters to generate a smooth trajectory. Typically, the function is a periodic function, such as a trigonometric function:

$$\theta = x_i + r_i cos(\phi) \tag{4.6}$$

θ_i is the set-point generated by the oscillator that can be regarded as an angle, torque or angular of a joint, it varies with time, and the unit can be determined according to the actual application. The above is a classical phase oscillator-based CPG model that generates smooth rhythmic gaits for quadrupedal robots. However, integrating sensory feedback into the CPG model is necessary to generate dynamic gaits. Using phase oscillators with output shaping as CPGs is one of the solutions. It having an explicit phase variable eases the process of phase coupling, which is needed for extension to higher dimensions (e.g. synchronised control of multiple legs / joints). Second, the output shaping allows CPG to generate the desired output waveforms. However, the drawback of using such CPGs is the limitation of output feedback integration. If one desires to regulate the output of the CPG through feedback, then everything needs to go through the phase, ϕ, and possibly amplitude, r, dynamics to see a reflected and indirect effect on the output. This is not always straightforward to implement.

4.3 Modified Phase Oscillator

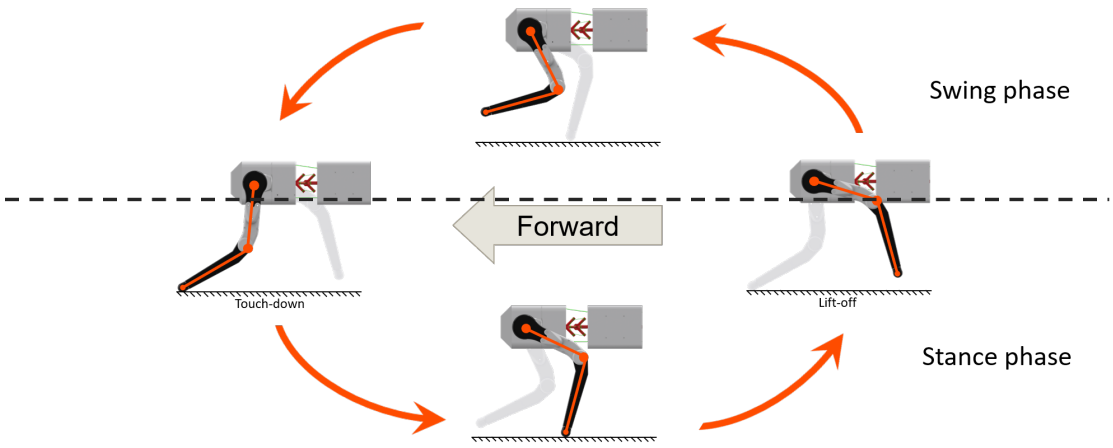


Figure 4.1 Stance and swing phase definition for quadruped robots

An important concept in quadrupedal robot gait is the stance phase and the swing phase, which are defined as shown in Figure 4.1. The amplitude and frequency of some

quadrupedal gaits vary in these two phases. Independent control of amplitude and frequency allows the potential generation of more dynamic and stable gaits. Although the CPG model mentioned in the previous subsection can produce rhythmic gaits, it is not able to adjust the stance and swing phases of the gait individually. Some studies have used piecewise linear functions to control the support and swing phases individually, but the mechanism relies on hard-coding to the extent that it can only control the support and swing phases for individual gaits or only one fixed waveform can be generated [171, 172]. For flexible control of the stance and swing phase, an internal feedback mechanism is proposed based on the basic CPG model in this thesis. The mechanism enables the CPG to independently control the frequency and amplitude of the stance and swing durations while also modifying the definition of stance and swing phases to generate more gaits.

The modified oscillators that composed the CPG model are based on [32], which is a set of differential equations. Due to the special mechanical design, the robot has two symmetric spine joints, which can only simultaneously be in a symmetrical orientation. Thus an oscillator is used to drive one orientation of the spine joint, a total of three oscillators controlling the spine of the robot, which are responsible for controlling the rotation in transverse, sagittal, and coronal planes separately. The equations of the modified phase oscillators used for all joints are given in equations 4.7-4.13:

$$\dot{\phi_i} = 2\pi * f(v_i) + \sum_j \omega_{ij} sin(\phi_j - \phi_i - \varphi_{ij}) \tag{4.7}$$

$$\dot{r_i^{st}} = a_r(R_i^{st} - r_i^{st}) \tag{4.8}$$

$$\dot{r_i^{sw}} = a_r(R_i^{sw} - r_i^{sw}) \tag{4.9}$$

$$\dot{x_i} = a_x(X_i - x_i) \tag{4.10}$$

$$\theta_i = x_i + f(r_i)cos(\phi_i) \tag{4.11}$$

with

$$f(r_i) = \lambda_i(r_i^{st} + r_i^{sw}) + r_i^{sw} \tag{4.12}$$

$$f(v_i) = \lambda_i(v_i^{st} + v_i^{sw}) + v_i^{sw} \tag{4.13}$$

where ϕ_i and v_i are the phase and intrinsic frequency of the *i-th* oscillator separately. The parameters w_{ij} are the coupling weights (set to 4 for all gaits in this project), which are used to represent the oscillator connection. Its value is the coupling strength and equals zero if oscillators i and j do not have a connection. φ_{ij} represents the phase difference between oscillator i and j. θ_i is the set-point generated by the oscillator that can be regarded as an angle, torque or angular of a joint. It varies with time, and the unit can be determined according to the actual application. In this study, θ_i represents a joint angle, and a PD controller is used to make an actuator follow it. X_i and x_i are the target and current offset of *i-th* oscillator. a_r and a_x are constant positive gains, set to 20 for all gaits, representing the convergence speed of amplitude and offset.

Equations 4.8 and 4.9 are used separately to achieve independent control of the amplitude in the stance and swing phase for the *i-th* oscillator. R_i^{st} and R_i^{sw} are control parameters used to set the desired amplitudes in the stance and swing, r^{st} and r^{sw} are state parameters that represent the current amplitudes. The intrinsic frequency in the stance and swing phases, v^{st} and v^{sw}, will be input directly into the CPG model.

4.3.1 Internal Feedback Mechanism

The modified phase oscillator described above uses two equations, 4.12 and 4.13, to represent the amplitude or intrinsic frequency in the stance and swing phases. Then, the corresponding amplitude or intrinsic frequency values are selected at different phases of the gait to generate gaits. However, another important question that follows is the definition of stance and swing phase in gaits. Since the stance phase and the

swing phase correspond to different phases of the output waveform in different gaits, the expectation here is to be able to adjust it with a parameter.

For the automatic selection of amplitudes and intrinsic frequencies corresponding to the phases in the gaits, an internal feedback mechanism is proposed in the CPG model, as illustrated in Figure 4.2. The idea of an internal feedback mechanism comes from the fact that the hip joint is always swinging back and forth in any gait, and that the phasing relationships of the shoulder and knee joints usually remain fixed with respect to the hip. Therefore, using information (ϕ_k) from the hip joint should be able to generate a signal (λ_k) to determine the stand and swing phase in a gait. Moreover, an ideal situation should be that different stance and swing phases can be defined by a control parameter (μ_k) for generating different gaits.

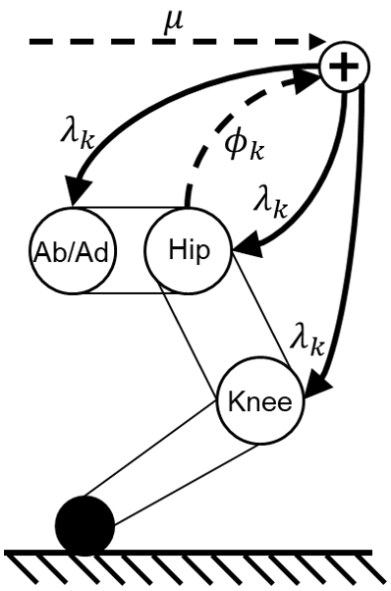


Figure 4.2 Internal feedback mechanism in a leg: μ_k is an user input and ϕ is obtained from a hip joint. Output λ_k value to shoulder(Ab/Ad), hip and knee joint to independent control of swing and stance amplitude and frequency.

The equation 4.12-4.13 utilises the mechanism to dictate the amplitude and frequency in each leg's stance and swing phases. λ_i is an internal feedback signal obtained from the *k-th* hip joint oscillator ($k \in (8, 9, 10, 11)$). It is used to determine the stance and swing phases of the oscillators of the shoulder and knee joints of the same leg.

The λ_k is calculated using equations 4.14-4.15:

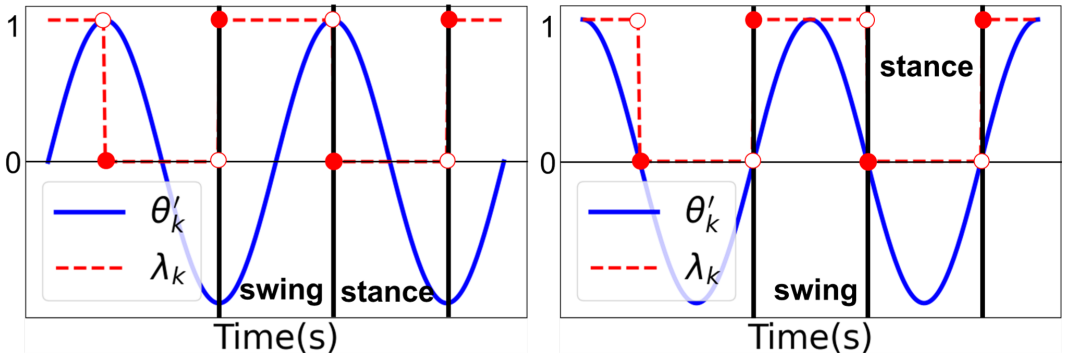
$$\lambda_k = \begin{cases} \frac{|\dot{\theta'_k}| - \dot{\theta'_k}}{2|\dot{\theta'_k}|}, & \dot{\theta'_k} \neq 0 \\ 1, & \dot{\theta'_k} = 0, \ddot{\theta'_k} \leq 0 \\ 0, & \dot{\theta'_k} = 0, \ddot{\theta'_k} > 0 \end{cases} \tag{4.14}$$

with

$$\theta'_k = cos(\phi_k + \mu_k) \tag{4.15}$$

Equation (4.14) is a linear piecewise function that only has two values: 0 and 1, which represent the swing and stance phases respectively. According to the definition of hip joint angle, the stance phase is a falling edge from the highest point to the lowest point during movement, while the swing phase is the opposite. Hence, determining whether the leg is in the swing or stance phase can be achieved by calculating the first and second-order differential of (4.11). Moreover, the definition of stance and swing phase should be changeable, considering that those are different in some behaviours, such as walking on the spot. To achieve this, equation (4.15) is introduced, which takes the *k-th* hip joint phase, ϕ_k, and a phase offset, μ_k, as input. The θ'_k has the same phase as the *k-th* oscillator and the parameter μ_k is used to change the definition of stance and swing phase in gaits. For example, Figure 4.3 shows the value of λ_k that varies with the *k-th* hip joint phase and how the value of μ_k' affects the definition of the stance and swing phase.

According to the internal feedback mechanism, the CPG can generate specific trajectories for joints by adjusting amplitudes (r_i^{sw}, r_i^{st}) and internal feedback signal (μ_k). Figure 4.4 demonstrates how the internal feedback mechanism is used to independently control the amplitudes in different phases to obtain the desired trajectories of the knee and hip joints. It should be noted that there is a phase difference between the knee trajectory in Figure 4.4a and the hip joint trajectory in Figure 4.3a, but the internal



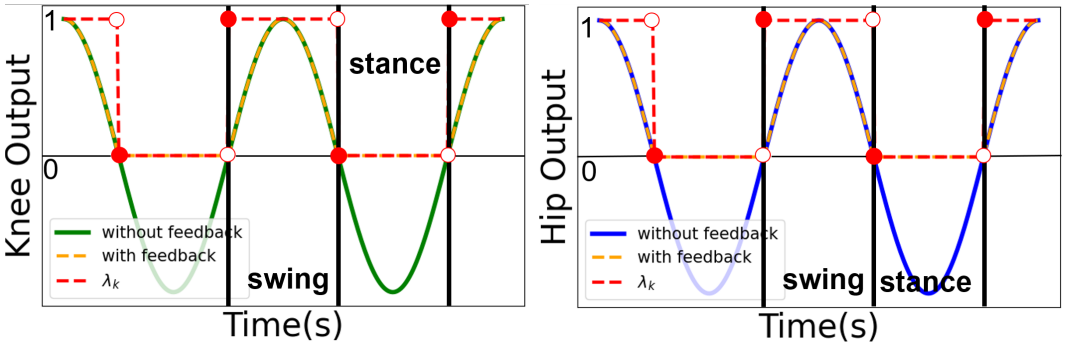
(a) phase offset $\mu_k = 0$ (b) phase offset $\mu_k = \frac{\pi}{2}$

Figure 4.3 Internal feedback signal: The dashed red line is the signal (λ_k), and the blue line is the trajectory of θ'_k. In (a), The definition of swing phase is defined as a rising edge of the blue line, while the stance phase is the falling edge. In (b), the swing phase corresponds to θ'_k value greater than 0, and the stance phase is the opposite. The red solid point indicates that the λ_k is defined, while the hollow point is not defined.

feedback signal is in the same position. The knee joint trajectory is usually seen in trot gait and walk gait. It does not move during the stance phase for stability. Figure 4.4b demonstrates a hip joint trajectory that is generated to trot on the spot. The internal feedback signal changes the definition of stance and swing phases as it differs from that in other moving gaits. However, most CPGs have a fixed or uncontrollable definitions of stance and swing phases, which means that the trajectory cannot be generated by them.

4.4 Topologies of CPG

Oscillator models are always used to study population dynamics and how inter-oscillator couplings and differences of intrinsic frequencies affect the synchronisation and the phase lags within a group of oscillators in nonlinear systems. There is a fact that the dynamics of a group of oscillators depend mainly on the type and topology of couplings rather than on the local mechanisms of rhythm generation, something that is well established in dynamical systems theory [173]. In robotics, designing the topologies of a CPG is still an empirical problem. As robots have different structures and DOF, it is



(a) knee joint trajectory, phase offset $\mu_k = 0$, $r_i^{sw} = 1$, $r_i^{st} = 0$

(b) hip joint trajectory, phase offset $\mu_k = \frac{\pi}{2}$, $r_i^{sw} = 1$, $r_i^{st} = 0$

Figure 4.4 Joints trajectories with/without internal feedback mechanism: The dashed red line is the signal (λ_k), and the solid line is the trajectory without feedback and dashed orange lines are trajectories with feedback. In (a), the green solid line is the knee joint trajectory without feedback. The blue solid line in (b) is the hip joint trajectory without feedback.

usually to design a suitable topology for a particular robot to generate its desired pattern in a stable and fast coupling. Fish and salamander-shaped robots use chain CPGs [32, 36, 31, 79], while quadrupedal robots usually use mesh structures [109, 174, 172]. Among them, the methods for mapping the outputs of the CPG network to the joint action space vary. For example, some studies use fewer oscillators to construct a CPG network, use the output of a single oscillator to correspond to the trajectory of the foot end, and then use inverse kinematics to derive the outputs of the other joints of a single leg based on the foot end position for the purpose of generating gaits. The joint dimensions of quadrupedal robots are more complex, typically 12 degrees of freedom, compared to snake or fish robots. This has the advantage of not constructing an overly complex CPG network, which means that the uncertainty in the coupling of all the oscillators in the whole CPG is reduced, and the complexity of parameter setting is also greatly reduced. However, due to its relatively low dimensionality, it makes it difficult to produce flexible and dynamic gaits. Another more intuitive approach is to construct an equal number of oscillators as joints or degrees of freedom so that each oscillator corresponds to the output of one joint or one degree of freedom. This definitely increases the complexity of the overall CPG, slows down the coupling and

makes tuning the parameters very difficult. However, it has the advantage of having the potential to generate highly dynamic gaits and many types of gaits.

4.4.1 Oscillators Number

One of the purposes of using the CPG model is to synchronise the movements of the spine and leg for TQbot naturally. In order to enable it to produce a variety of movements and gaits, a network structure was selected. In this topology, each oscillator corresponds to one degree of freedom and its output is the angle of the joint. It should be emphasized here that TQbot has 15 DOF in the whole body, 12 DOF in the legs and 3 DOF in the spine. There is one more oscillator that does not correspond to any DOF, and its role will be explained later. Therefore, the CPG model is designed with 16 oscillators, its correspondence with each joint is shown in Figure.4.5.

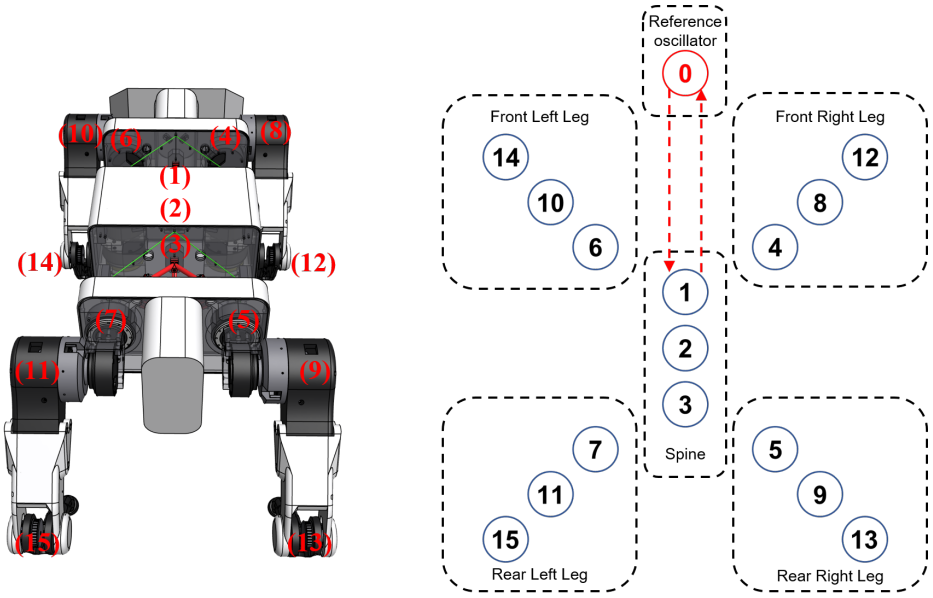


Figure 4.5 CPG and TQbot joint correspondence diagram. The left shows TQbot model, with indices corresponding to those in the CPG model on the right.

4.4.2 Reference Oscillator

For CPG models with a limited number of oscillators and simple connection relationships, the phase matrix utilises relative phase to determine the phase difference between oscillators. However, as the number of oscillators grows and the topology structure becomes increasingly complex, identifying the relative phase is challenging. For example, a fully connected CPG network comprising four oscillators necessitates at least six relative phases to derive a complete phase matrix. In our 15-oscillator CPG network, a minimum of 27 relative phases between oscillators must be determined due to the specified connectivity.

Although it is possible to calculate the entire phase matrix from the relative phase of any one of the oscillators to the others, this approach leads to modifying the phase difference with all other oscillators when the phase of that oscillator needs to be altered. This undoubtedly complicates the process of adjusting the phase difference matrix. For instance, if the oscillator that controls the roll motion of the spine joint is employed as a reference oscillator, adjusting its phase in a specific gait would require changing the phases of all other oscillators. Consequently, using the relative phase to obtain the entire phase difference matrix can be troublesome for complex CPG models.

To resolve this issue, a reference oscillator is introduced to acquire the phase difference from others and calculate the relative phase between oscillators, thereby obtaining the entire phase matrix. The reference oscillator in the CPG is labelled as zero and is connected solely to the second oscillator, marked as one, that controls the roll orientation of the spinal joint, as shown in Figure 4.5. The reference oscillator does not output externally and maintains an unchanged phase.

Overall, the role of the reference oscillator is similar to that of data normalisation in machine learning. It makes the phase lag parameters of each oscillator represent the periodic distance of its trajectory from the cosine function trajectory, which simplifies the design of phase parameters.

4.4.3 Oscillator Connection

Synchronisation has always been a vital aspect of CPGs, mainly determined by the type and topology of the couplings [173]. Once the number of oscillators is determined, it is essential to establish a CPG structure that facilitates stable and rapid coupling between oscillators. In this study, five different topologies were designed and tested: (1) full connection, (2) adjacent connection, (3) adjacent connection with leg-spine coupling, (4) divergent connection and (5) divergent connection with leg-spine coupling. Each topology was evaluated for its coupling effectiveness using five gaits: spot trot, walk, trot, pace, and bounding gait. Order parameter is used to quantify the degree of coupling in each structure.

Order Parameter

In a nonlinear system, the order parameter is used to visualise the dynamics of the phases, which can be imagined as a swarm of points running around the unit circle in the complex plane [175]. Using the same notation with the modified CPG model, the order parameter can be defined as:

$$he^{i\Psi} = \frac{1}{N}\sum_{j=1}^{N} e^{i(\phi_j - \varphi_{0j})} \tag{4.16}$$

where h is the modulus of $e^{i\Psi}$ and ranges from 0 to 1 and measures the degree to which the oscillators are synchronised. Ψ is the average phase of $(\phi_j - \varphi_{0j})$. In the modified CPG model, there is a phase difference between each oscillator. Therefore, in order to meet the definition of the order parameter, the phase of each oscillator ϕ_j is subtracted from its phase difference φ_{0j} with the reference oscillator when calculating the average phase Ψ. In this way, when h=0, it means that the phase of the oscillator is completely uncoupled, and when h=1, it means that the phase of all oscillators is coupled to the given phase difference.

In each topological mechanism, all the oscillators were bi-directionally connected and the reference oscillator was only connected to the first spine oscillator. During the test, the CPG first outputs a spot trot gait, then switches to a walk gait at the third second, and then switches to a gait every 5 seconds, with the sequence of gait transitions being spot trot-walk-trot-pace-bound. While gait transitions are quite simple in CPG and can be realised by replacing the phase-lag matrices of the different gaits directly. The parameters of the five gaits will be given in the next subsection.

Full Connection

Fully connected structures are more common in CPG models with few oscillators, and the interaction between each oscillator can make its whole network coupled faster and more stable. In this CPG model, full connection enables the leg oscillators and the spinal oscillators to interact with each other, as shown in Figure 4.6. However, as the number of oscillators increases, their coupling will become complicated. So the fully connected CPG model will also be explored here and used as a baseline for later experiments.

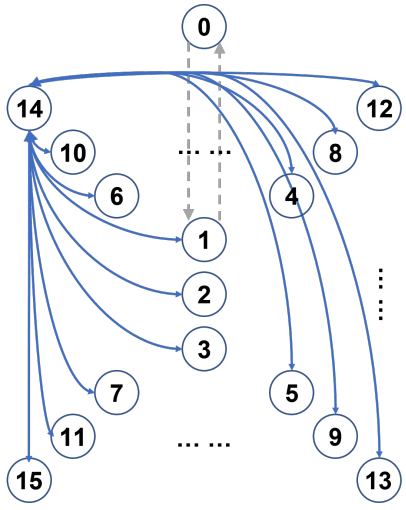


Figure 4.6 Full connection topology. The picture shows the connection of the 14th oscillator. It connects with other oscillators except the reference oscillator, and each connection line is bidirectional. Other oscillators are connected to each other which as same as the 14th oscillator except the reference oscillator.

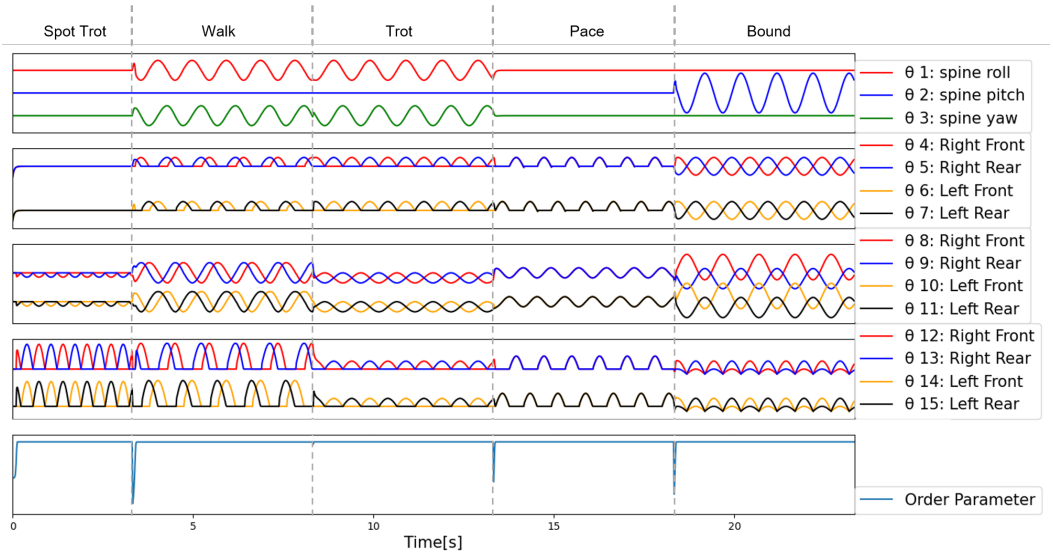


Figure 4.7 Full connection joints trajectories with order parameter. Gait transitions are achieved by replacing the parameter configurations of different gaits. Changed at 3, 8, 13 and 18 seconds. Initial positions of all joints are 0. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints, respectively.

Figure 4.7 shows the results of the fully connected CPG network generating gaits. The smooth gait transition between each gait can be seen in the figure. Moreover, based on the order parameter h it can be seen that the full connection CPG coupling is indeed fast and stable. Among them, the gait transition between walking and trotting gait was the most smooth and natural, with the h-value remaining at 1 with almost no fluctuation. However, although all oscillators were coupled at a fast speed during the generation of the walking gait, their intrinsic frequencies did not correspond to the desired values. From the figure, it appears that the swing and stance phases of the walking gait have the same frequency, whereas the desired should be $3:1$.

Adjacent Connection I

Some CPG models with many oscillators do not employ a fully connected approach but instead utilise an adjacent connection. For instance, in the CPG model of a quadrupedal salamander robot, each oscillator is connected only to its adjacent oscillator [31]. For

the TQbot CPG model, two adjacent connected structures are applied. The first connection topology is shown in Figure 4.8, which has adjacent connections between the leg oscillators, while the three spine oscillators are still fully connected to the other oscillators. This structure reduces the interaction between the leg oscillators compared to a full connection structure.

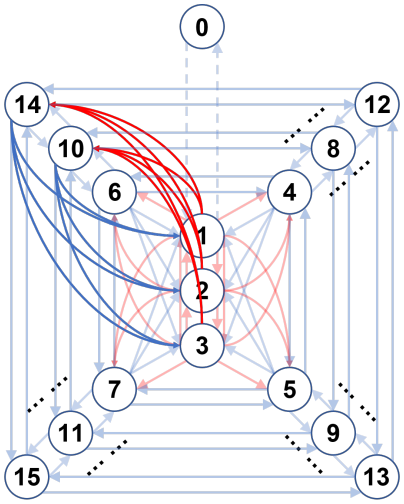


Figure 4.8 Adjacent connection I topology. The connecting lines of adjacent oscillators are lighter in colour and the connection between non-adjacent oscillators are indicated by solid lines. The connection between oscillators 8, 9, 11, 12, 13, 15 and the spine oscillators, 1, 2, 3, are omitted, which are consistent with the connection between oscillators 10, 14 and spine oscillators. Each line is a unidirectional connection. The red lines indicate spinal oscillators connected to other joints, and the blue lines are leg joint oscillators' connections. The arrow shows the connection direction between each oscillator.

As can be seen in Figure 4.9, although the speed at which to complete the coupling for this connection structure is slightly lower than that of the CPG for the fully connected structure, all the gaits can still complete the coupling in an appropriate time. Similarly, the CPG of this structure, although it completes coupling the phase difference of the walking gait, still does not have an internal desired velocity in line with the desired value, as can be seen in Figure 4.9.

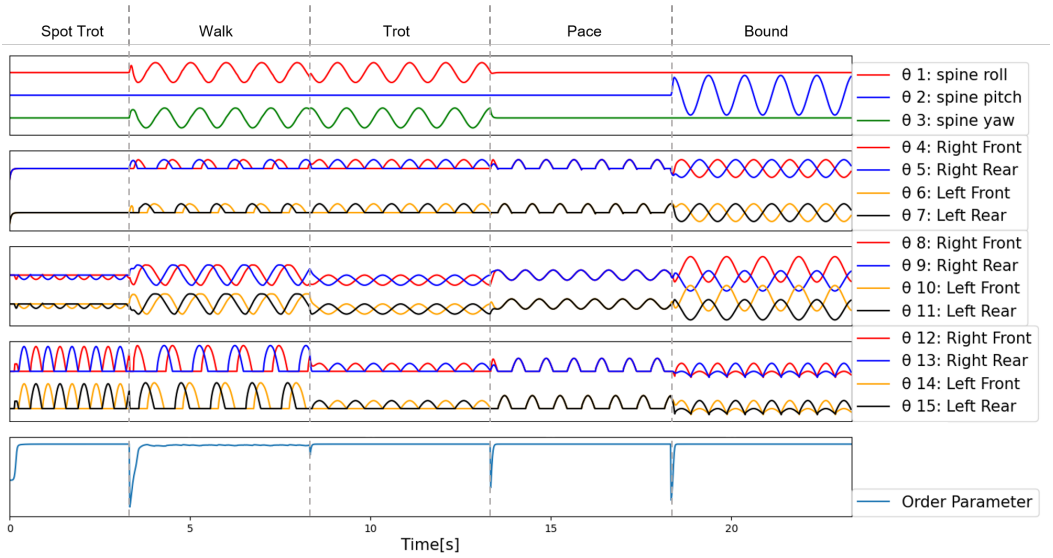


Figure 4.9 Adjacent connection I joints trajectories with order parameter. Gait transitions is achieved by replacing the parameter configurations of different gaits. Changed at 3, 8, 13 and 18 seconds. Initial positions of all joints are 0. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints, respectively.

Adjacent Connection II

The second adjacent connection is shown in Figure 4.10. Compared to the first adjacent connection topology, the second retains only the full connections between the three spinal oscillators and the shoulder joint oscillators, while the other oscillators are connected only to their adjacent counterparts. This topology minimises the influence of the spine oscillator on the hip and knee oscillators while preserving interaction with the shoulder joint, thereby enabling the coupling of the entire CPG model.

After removing the spine connections to the knee and hip oscillators from the first neighbouring connections the coupling speed decreases significantly but all gaits except the walking gait can be coupled to the desired phase difference. For the walking gait, the coupling is stabilised with the oscillators coupled to the desired phase difference by about 95%. However, the upside is that the walking gait is much closer to the desired velocity in the support and swing phases than the previous structure.

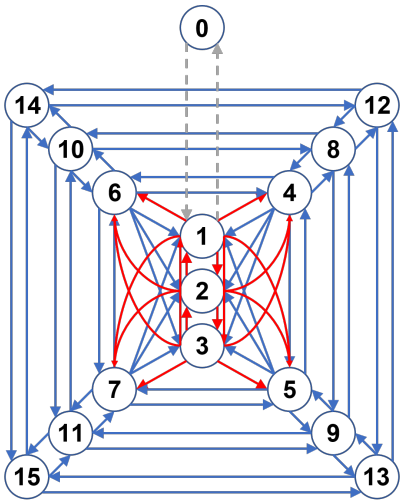


Figure 4.10 Adjacent connection II topology. The connecting lines of adjacent oscillators are lighter in colour and the connection between non-adjacent oscillators are indicated by solid lines. The connection between oscillators 8, 9, 11, 12, 13, 15 and the spine oscillators, 1, 2, 3, are omitted, which are consistent with the connection between oscillators 10, 14 and spine oscillators. Each line is a unidirectional connection. The red lines indicate spinal oscillators connected to other joints, and the blue lines are leg joint oscillators' connections. The arrow shows the connection direction between each oscillator.

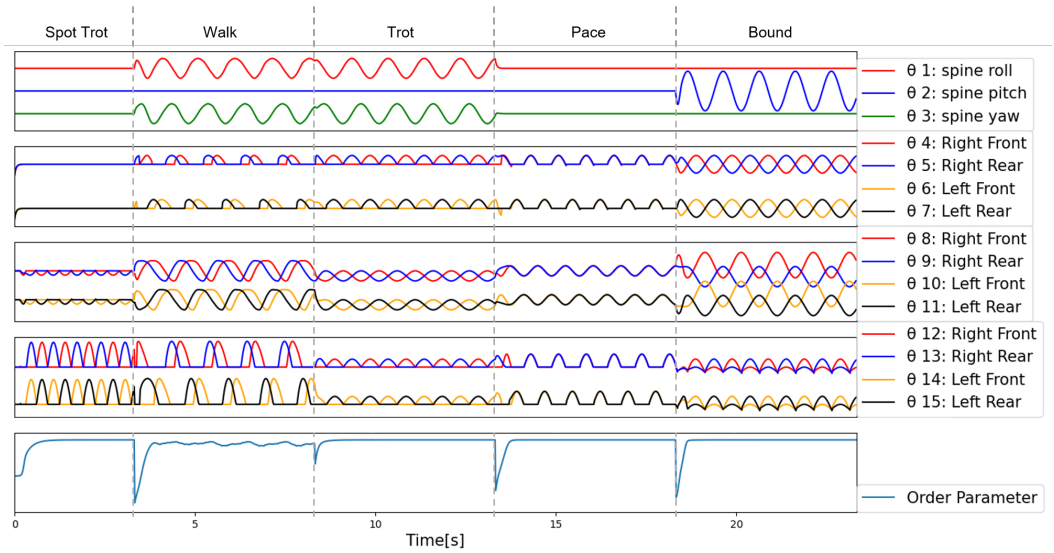


Figure 4.11 Adjacent connection II joints trajectories with order parameter. Gait transitions is achieved by replacing the parameter configurations of different gaits. Changed at 3, 8, 13 and 18 seconds. Initial positions of all joints are 0. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints, respectively.

Divergent Connection I

In terms of the CPG model based on divergent connection topology, the spine oscillator is viewed as the centre of the structure and the connection is thus diverged to the oscillators of the four legs. As shown in Figure 4.12, the first divergent connection does not have a connection between the oscillators of each leg. The three oscillators of each leg are chain-connected and are only connected to the spine oscillator through the shoulder joint, while the spine oscillators are connected to each other. In this topology, the influence between the leg oscillators is further attenuated, but it still retains the direct influence of the spine oscillators on it.

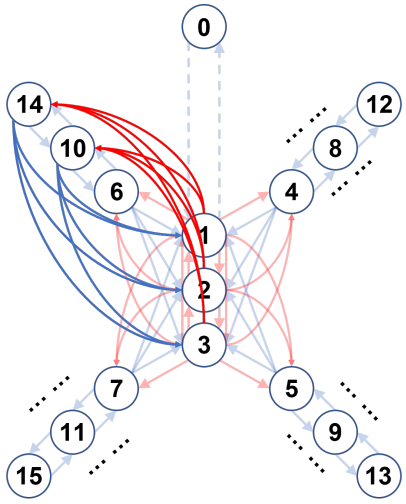


Figure 4.12 Divergent connection I topology. Each line is a unidirectional connection. The red lines indicate spinal oscillators connected to other joints, and the blue lines are leg joint oscillators' connections. The arrow shows the connection direction between each oscillator.

The results for divergent connection I are similar to those for adjacent connection II. The coupling speeds are slightly slower than the CPG with the full connection topology, but the intrinsic frequency of the walking gait is closer to the desired value.

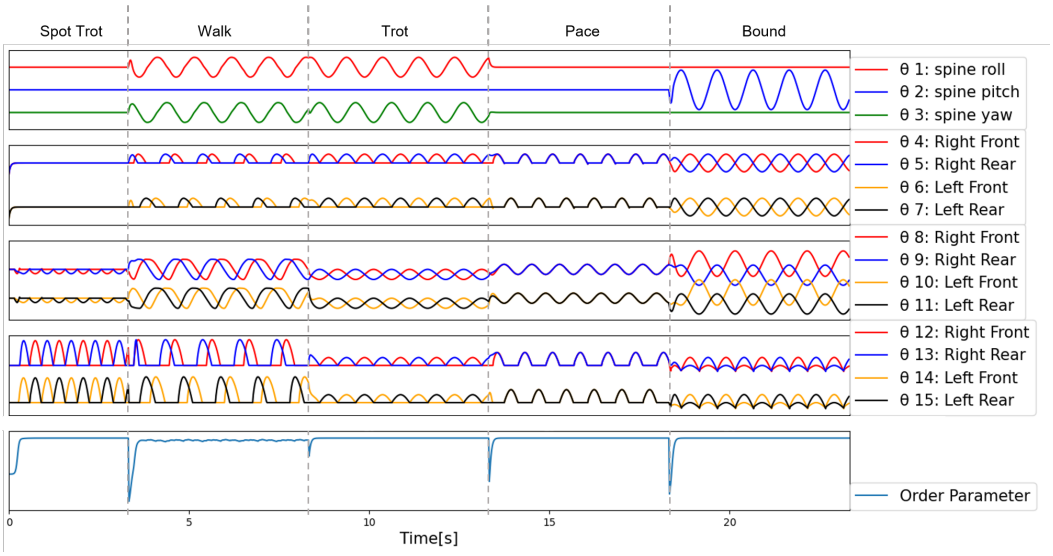


Figure 4.13 Divergent connection joints trajectories with order parameter. Gait transitions is achieved by replacing the parameter configurations of different gaits. Changed at 3, 8, 13 and 18 seconds. Initial positions of all joints are 0. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints, respectively.

Divergent Connection II

The second divergence structure also differs from the first in that it cancels the connection between the spine oscillator and each leg oscillator, making it a full divergence connection. In this structure, the coupling of the oscillators starts from the reference oscillator to the spine oscillator, and then to the oscillator of each leg for a stepwise coupling.

Divergent connection II is the topology with the least amount of connection structure between oscillators, the results of which are shown in Figure 4.15. In this topology, the time required for all gaits to couple to a fixed phase difference is the longest. In addition, the phase of the walking gait cannot be coupled to the desired phase difference, although its intrinsic frequency is very close to the desired value.

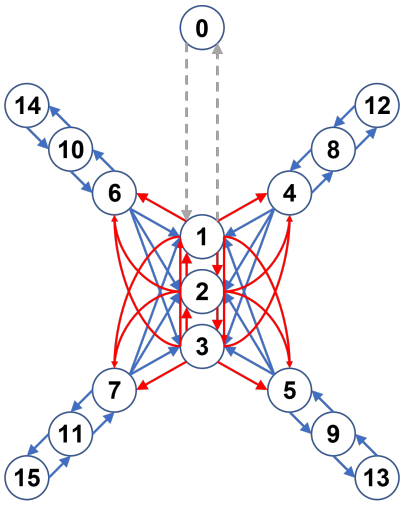


Figure 4.14 Divergent connection II topology. Each line is a unidirectional connection. The red lines indicate spinal oscillators connected to other joints, and the blue lines are leg joint oscillators' connections. The arrow shows the connection direction between each oscillator.

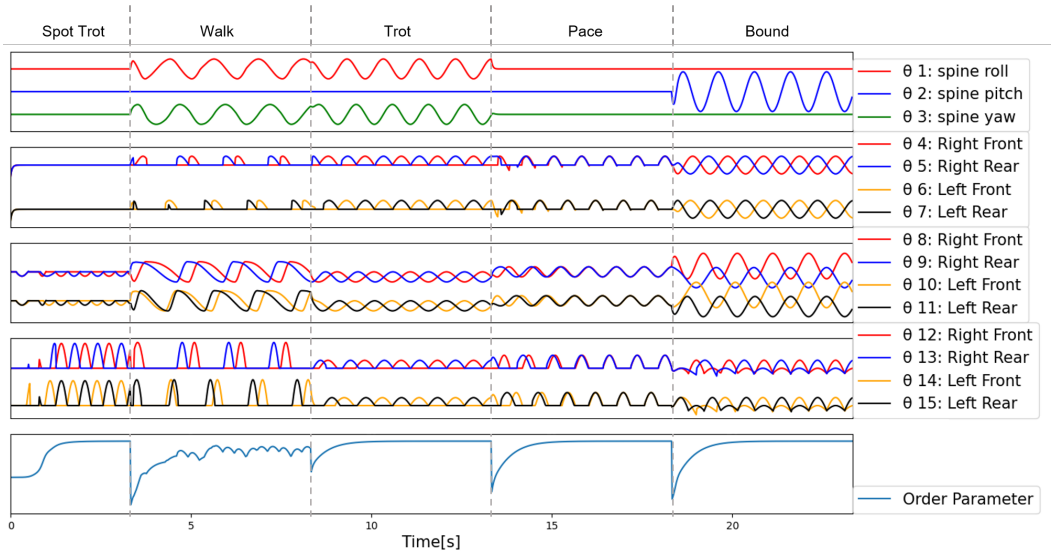


Figure 4.15 Divergent connection joints trajectories with order parameter. Gait transitions is achieved by replacing the parameter configurations of different gaits. Changed at 3, 8, 13 and 18 seconds. Initial positions of all joints are 0. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints, respectively.

The Difference of Five Connection Topologies

The number of connections between the oscillators of these five topologies decreases gradually, the first fully connected topology coupling is the fastest but there is a risk that the intrinsic frequency will not reach the desired value. The main difference between neighbouring and divergent connections is whether the hip and knee oscillators of neighbouring legs are connected to each other. In both connection structures, coupling results with and without spine oscillators fully connected to the leg oscillators are also explored. Adjacent connection type II reduces the direct influence of spine oscillators on all leg oscillators compared to type I, while divergent links further weaken the interaction between legs.

From the results of the gaits generated by these five topologies, it is clear that although the higher number of connections between the oscillators allows them to couple quickly to the desired phase difference, as the gait becomes more complex, its oscillators may appear to be inconsistent with the given intrinsic frequency. A walking gait, for example, possesses different velocities in the stance and swing phases, but with a fully connected topology, its intrinsic frequency is exactly the same in both phases. According to the experimental results, as the number of connections between oscillators decreases, their coupled and intrinsic frequencies will tend to be similar. However, this comes at the cost of slower coupling speed and lower coupling degree. Therefore, when designing a complex CPG network structure, it is necessary to make a trade-off between the coupling speed and the coupling degree of the intrinsic frequency and design a topology corresponding to the number of connections. In this work, the divergent connection II topology will be used for the CPG model in all subsequent experiments.

4.5 Output Trajectories

Currently, CPG is not the main gait generation method for quadruped robots because its ability to generate dynamic gaits is limited. One of the reasons for this is that the

mechanism utilising feedback to shape locomotion is unknown in biology. The conventional CPG model with phase oscillators can only generate rhythmic but unnatural gaits for quadruped robots [32]. Other modified CPG models usually use hard-code programs to control the amplitude and frequency in the stance and swing phases, but it is difficult to generate different patterns [172]. However, the proposed internal feedback mechanism provides a way to shape more natural and various gaits that are difficult for other models. In this section, several gaits are analysed for tuning the parameters of the CPG to generate different frequencies and amplitude in the stance and swing phases. Especially trot-in-place gait, which other CPG models generate poorly. The results of the diversity and effectiveness of gait patterns shaped by the internal feedback mechanism are verified by conducting experiments in a simulation called Isaac Sim[1].

4.5.1 Default Angle Definition

After setting up the reference oscillators, the phases of the individual joints of the robot can be easily determined. In order to further determine the specific values of their amplitude and bias, the direction of rotation of the joints needs to be defined. For TQbot, the leg joints with amplitudes in CPG are set as shown in Figure 4.16. The figure shows the definitions of the rotation angles for the shoulder joint, the model shutter, and the knee joint in each direction of rotation. These definitions can help first determine the value of the amplitude in CPG for each joint.

Shoulder Joint

Figure 4.16a presents a simplified rear view of the robot joints, illustrating the definition of the offset and amplitude angle for the left shoulder joint. In the TQbot design, the shoulder joint, which is aligned with the hip joint, controls the leg's rotation in the transverse plane. The 0 degree angle is defined as the line connecting the shoulder and

[1]All experimental results can be found at https://youtu.be/NJrMatc6PwU

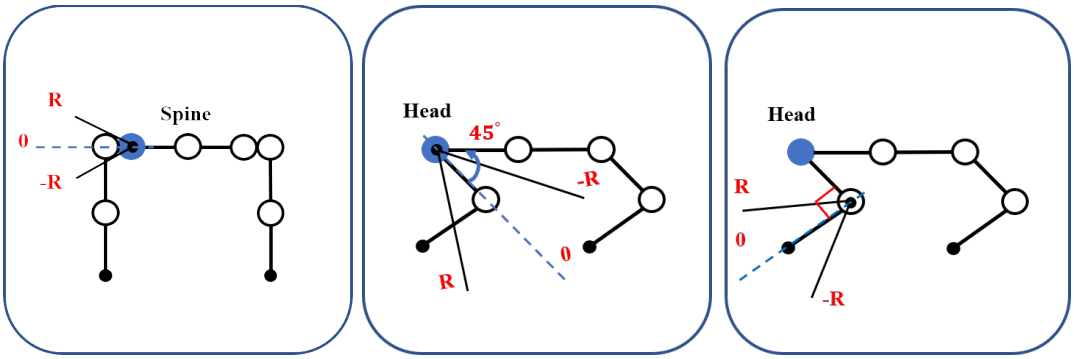
hip joints and 45 degrees from the horizontal plane. Its clockwise rotation designated as positive and counterclockwise as negative.

Hip Joint

Similarly, as shown in Figure 4.16b, the angle of the hip joint is defined with the 0-degree position pointing towards the knee joint, following the same convention for the rotation direction.

Knee Joint

For the knee joint, shown in Figure 4.16c, the angle of 0 degrees is defined along the line from the knee to the foot, with clockwise and counterclockwise rotations labelled as positive and negative, respectively. For the right leg, on the other hand, the definition of the joints angle is perfectly symmetrical along the sagittal plane with the definition of the joint angle for the left leg, which will make the subsequent determination of the parameters easier and more intuitive.



(a) Shoulder joint(rear view) (b) Hip joint(left view) (c) Knee joint(left view)

Figure 4.16 Definition of leg joint amplitude direction. The pictures are simplified diagrams of the joint positions. Solid circles represent defined joints, while hollow circles represent other joints.

Spine Joint

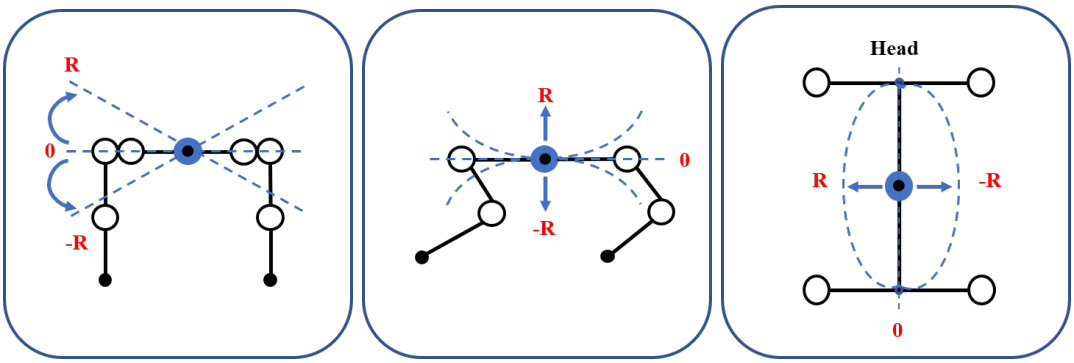
The TQbot spine consists of two joints, each with three degrees of rotational freedom. Due to the adoption of the CPRM configuration, the front and rear joints of the spine move symmetrically. In the most common CPG model, it is usual to assign one oscillator to each joint or DOF. Consequently, the TQbot CPG is designed to employ three oscillators, corresponding to the three rotational DOFs: roll, pitch, and yaw. These oscillators, labelled 1-3, generate the joint angle trajectories for the roll, pitch, and yaw orientations, respectively. The definitions of these rotational directions are illustrated in Figure 4.17.

When the front, middle, and rear segments of the body are aligned on the same level and connected in a straight line, the spinal joints are defined as being at 0 degrees for roll, pitch, and yaw. For the roll angle, clockwise rotation of the front body segment is defined as the positive direction, while counterclockwise rotation is negative; the rear body segment rotates in the opposite direction due to the body's configuration. The pitch angle is positive when both the front and rear body segments are lifted upwards and negative when they are lowered. For the yaw angle, a leftward curvature of the spine is defined as the positive direction, while a rightward curvature is considered negative.

Once all the joint angles have been defined, it is easy to later determine the parameters for each gait to generate the correct gait for the robot. All joint amplitude values for different gaits are shown in Table 4.1

4.5.2 Basic Gaits

Following the gaits often observed in quadruped mammal walking videos, the phase difference between each joint can be determined. In addition, the phases of the shoulder, hip, and knee joints in the same leg are fixed. Therefore, the phase lag matrix of a gait in CPG can be obtained by first determining the phase differences of the three



(a) Roll direction(rear view) (b) Pitch direction(left view) (c) Yaw direction(top view)

Figure 4.17 Definition of spine joint amplitude direction. The pictures are simplified diagrams of the joint positions. Solid circles represent defined joints, while hollow circles represent other joints. Note that, the roll direction is defined for the first spine joint(near the front body segment).

joints in a leg and then determining the phase differences between each leg. Here, four basic gaits are developed for TQbot: walk, trot, pace and bounding gaits, which the phase difference of shoulder, hip and knee joints in the same leg are shown in Figure 4.18. For all gaits, a leg must be lifted off the ground to allow effective forward motion during the swing phase, and the foot must touch the ground to form a stable support during the stance phase.

To achieve the swing phase, the knee joint is contracted and the shoulder joint is lifted upward. Then, as the hip joint swings back to its original position from the highest position, the shoulder and knee joints are raised to their highest points. At the start of the stance phase, the shoulder and knee joints return to their original positions, and the hip joints reach the lowest position. The knee and shoulder joints generally remain stationary during the stance phase to ensure the stability of the robot while in motion unless the swing of the hip joint is too large, in which case the knee and shoulder joints need to be extended until they hit the ground and then bent back into their original position. In the walk, trot and pace experiment, the swing of the hip joint was kept at small angles, figure 4.18a demonstrates the single leg joints phase lag for the three gaits.

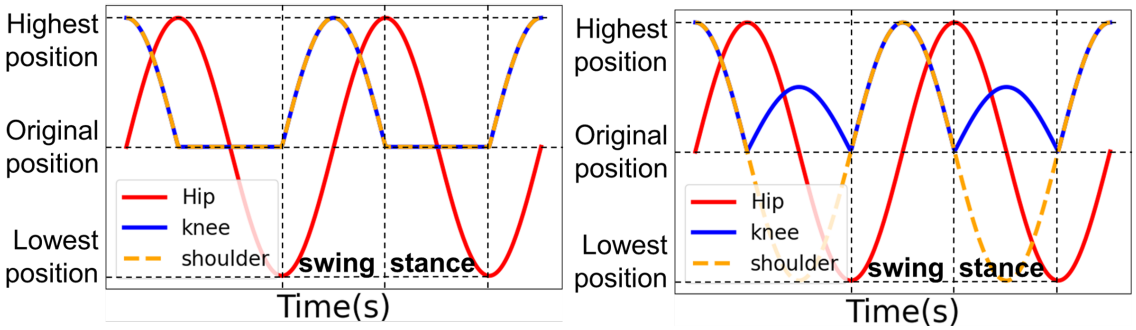
Table 4.1 Joint amplitude values for different gaits(in radians). The amplitude values of the same type of joints are identical

Gait	**Phase**	**Oscillator Amplitudes**					
		Roll	**Pitch**	**Yaw**	**Shoulder**	**Hip**	**Knee**
Walk	Swing	0.044	0	0.044	0.025	0.35	0.52
	Stance	0.044	0	0.044	0	0.35	0
Trot	Swing	0.087	0	0.087	0.025	0.175	0.16
	Stance	0.087	0	0.087	0	0.175	0
Pace	Swing	0	0	0	0.025	0.175	0.52
	Stance	0	0	0	0	0.175	0
Bound	Swing	0	0.175	0	0.025	0.52	0.262
	Stance	0	0.175	0	0.025	0.52	-0.1
Spot Trot	Swing	0	0	0	0	0.15	0.5
	Stance	0	0	0	0	0	0

Figure 4.18b shows the single leg joints phase in the gallop gait, which the shoulder and knee joint phase lag is different with the previous three gaits. The shoulder joint did not remain in position in the stance phase but continued to swing to obtain a greater stride length to increase speed. The phase of the knee joint is also different from other gaits in the gallop gait due to the hip joint swinging at big angles. In order to maintain stability during the run, the knee joint in the stance phase will, the same as in the swing phase, contract slightly at first, reaching its highest position at the original position of the hip swing and then gradually extending back to its original position.

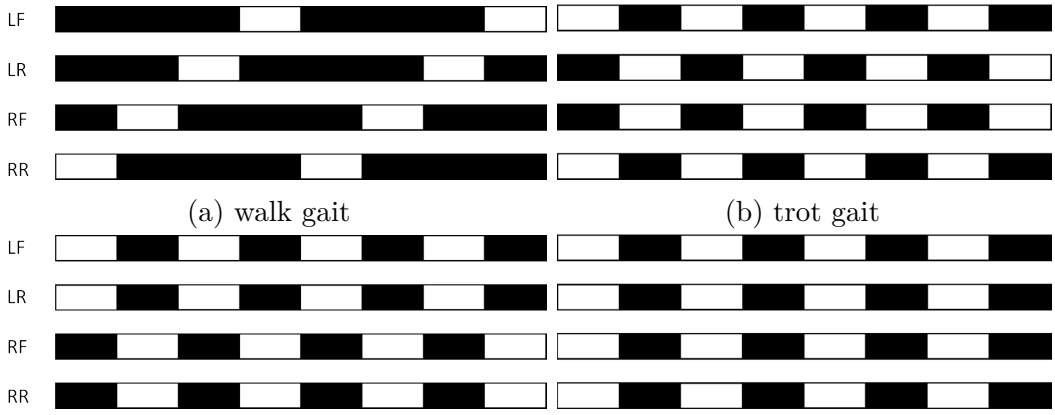
For quadruped robots, different gaits produced by CPG are needed to adapt to the surrounding environment. The four basic gaits used in TQbot are obtained by changing the phase lag between the oscillators. The phases of four legs for walking, trotting, pace, and bounding gaits are shown in Figure 4.19 and their values are shown in Table 4.2.

Additionally, the frequency of all oscillators in each gait is consistent. The frequency parameters are shown in the Table 4.3.



(a) Joints outputs in walk, trot, pace gaits (b) Joints outputs in gallop gaits

Figure 4.18 Single leg joints trajectories shaped by the internal feedback mechanism: phase and amplitude relationships in gaits.



(a) walk gait (b) trot gait

(c) pace gait (d) bound gait

Figure 4.19 Different gaits phase lag. The white block represents the swing phase and the black block represents the stance phase.

Table 4.2 Basic gaits phase lag. The phase lag between the reference oscillator and other oscillators is always 0.

Gait	**Oscillator**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Walk	0	0	$\frac{3\pi}{2}$	π	$\frac{3\pi}{2}$	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	0	π	$\frac{3\pi}{2}$	0	$\frac{\pi}{2}$
Trot	0	0	$\frac{3\pi}{2}$	π	0	0	π	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	π	0	0	π
Pace	0	0	0	$\frac{3\pi}{2}$	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	π	π	0	0	$\frac{3\pi}{2}$	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$
Bound	0	$\frac{\pi}{2}$	0	0	π	0	π	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	0	π	0	π
Spot Trot	0	0	0	0	0	0	0	π	0	0	π	0	π	π	0

Table 4.3 Basic gaits frequency. The reference oscillator frequency is always 1 for all gaits.

Gait	Stance Frequency	Swing Frequency
Walk	0.4	1.6
Trot	1.0	1.0
Pace	1.3	1.3
Bound	1.0	1.0

Finally, the CPG update step size is set to match the simulation physics engine update frequency of 60 Hz, resulting in a step size of 1/60.

Walk

As shown in Figure 4.19a, the four legs in walk gait swing alternately in the sequence of left front (LF), left rear (LR), right front (RF), right rear (RR). When one leg is in the swing phase, the other three legs are in the stance phase. This results in a hip joint speed ratio of 1:3 between the stance and swing phases for each leg. The gait exhibits a stable three-point support pattern, with at least three legs in contact with the ground at any given time, ensuring the robot's stability.

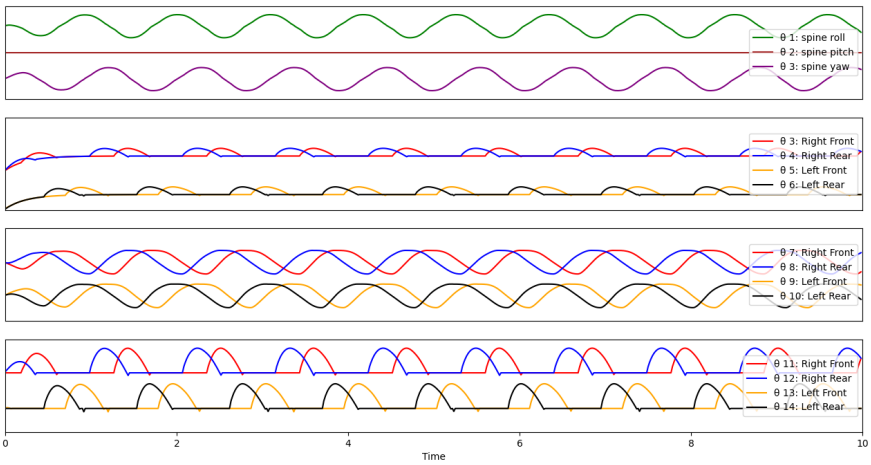


Figure 4.20 Joint trajectories of walk gait. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints.

Trot

In the trot gait, the LF leg and the RR leg move in sync, while the RF leg and the LR leg also move in sync, as shown in Figure 4.19b. This gait is characterised by diagonal pairs of legs moving together, forming an alternating two-point support pattern. This gait is faster than walk gait while maintaining a degree of stability.

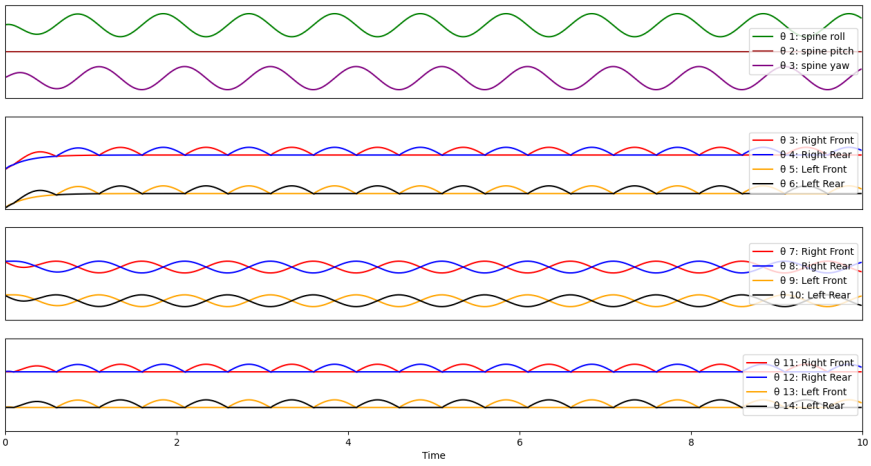


Figure 4.21 Joint trajectories of trot gait. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints.

Pace

Figure 4.19c shows the pace gait phase, the LF leg and LR leg move in sync, as do the RF leg and RR. This means that the legs on the same side swing or support simultaneously. The gait is characterized by a two-point support pattern, typically used in scenarios requiring higher speed but offering lower stability.

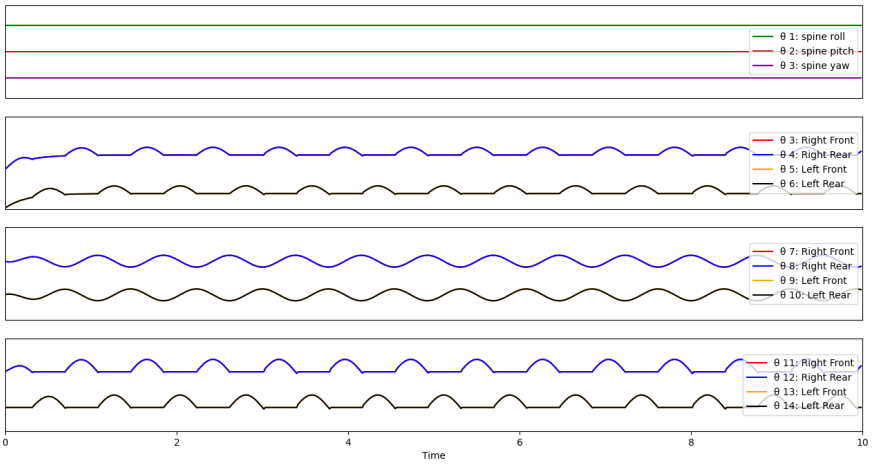


Figure 4.22 Joint trajectories of pace gait. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints.

Bound

In the bound gait, the front legs (LF and RF) swing together, and the rear legs (LR and RR) also swing together 4.19d. In this gait, the robot's front and rear legs move in unison, creating a pattern where pairs of legs either swing or support at the same time. The bound gait is used for high-speed movement, but during this gait, the robot may have moments where no legs are in contact with the ground, leading to lower stability.

Trot in Place Gait

Except for the four basic gaits mentioned, the gait of the trot in place is another gait commonly used for quadruped robots. Its phase difference between legs is the same as trot gait, but the definition of swing and stance phases is different. The swing phase is when the hip joint swings backwards from the original position to the lowest position and back to the original position, while the whole leg does not move in the stance phase. It can be easily generated by adjusting the phase offset ($\mu_k = \frac{\pi}{2}$) in the internal feedback mechanism.

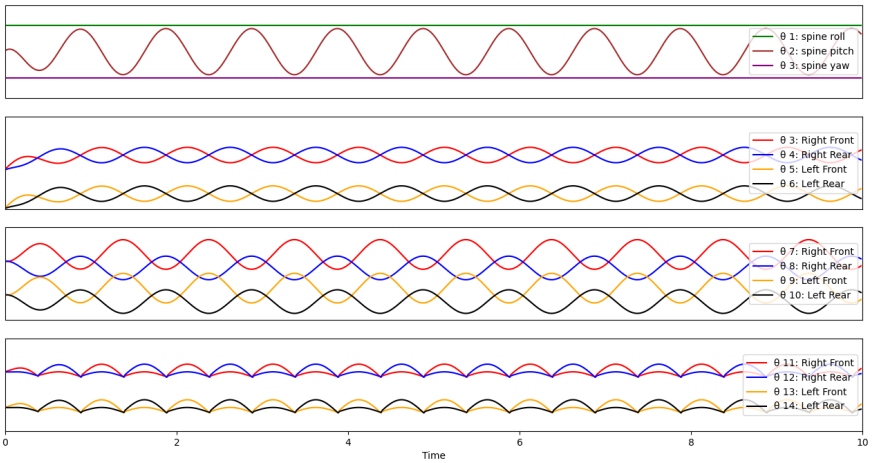


Figure 4.23 Joint trajectories of bound gait. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints.

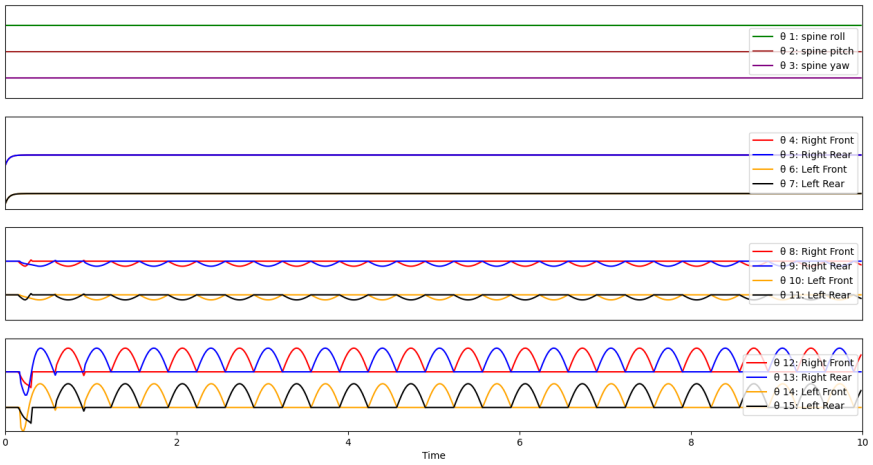


Figure 4.24 Joint trajectories of spot trot gait. 1 to 3 are spinal joints corresponding to the movement in the transverse, sagittal and coronal planes. 4-7, 8-11, and 12-15 are shoulder, hip and knee joints.

The Usage of the Spine

Typically, spine flexion at the coronal plane increases the length of the stride to improve speed, while flexion in the transverse plane helps with turning. In the above gaits, except pace and trot in place gait does not use spine movement, other gaits utilise the spine to varying extents. Walk and trot gaits incorporate the spinal movements in the coronal and transverse planes. As the robot's front leg swings forward, the curvature of the spine in the coronal plane is toward the side of the front swinging forward leg. The movement of the spine in the transverse plane serves as a functional equivalent to the shoulder joint, lifting the leg during the swing phase. In the gallop gait, the spine bends downwards in the sagittal plane as the two front legs swing forward and upwards in the stance phase of the front legs, increasing the stride and speed of the robot.

4.5.3 Multi-direction Movement in the Trot Gait

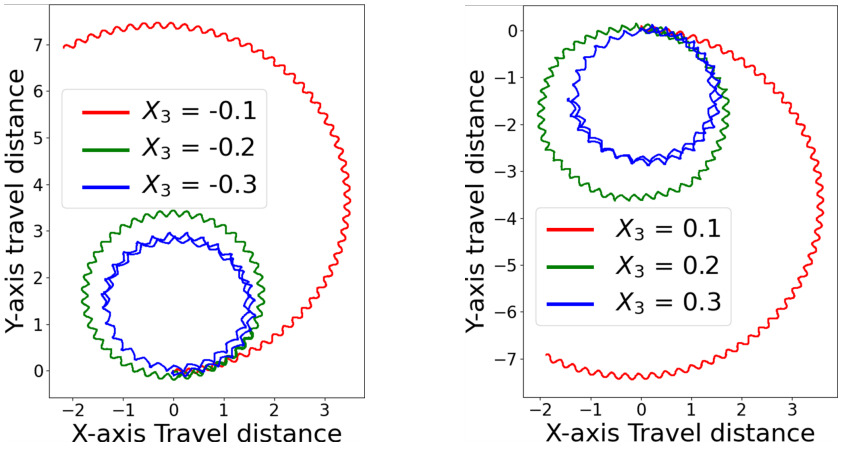
Move Backwards

In addition to implementing the basic gait for forward movement, a trot gait to enable the robot to move backwards has been developed. Notably, opposite to the forward gait, the hip joint in the backward gait enters the stance phase while swinging forward and the swing phase while swinging backwards. Thanks to the reference oscillator, the new phase difference matrix can be obtained by swapping the phase of the hip joints of the left and right legs. Other parameters and phase differences are the same as the forward trot gait parameters.

Sideways Movement

Additionally, sideways movements are implemented based on the trot gait, which allows the robot to move laterally to the left or right. In this gait, the shoulder joints of the left and right legs are in phase, while that of the front and rear legs are in phase for

half a cycle. In other words, the shoulder joints of the left front and right front legs are simultaneously extended to their highest position while the shoulder joints of the left rear and right rear legs are simultaneously contracted to their lowest position. It is important to note that the hip joint does not move during this gait; also, the stance and swing phase of shoulder and knee joints are different when moving left and right. For translating to the left, the shoulder joint of the left front leg is abducted in the swing phase, and that of the right rear leg is adducted simultaneously. Meanwhile, the knee joints of the left front leg and right rear leg rotate to the highest position, while those of the right front and left back legs rotate to the lowest position. The shoulder joint is in the opposite phase when moving to the right, which can be achieved by swapping the phase difference between the right and left shoulder joints.



(a) Turning left (b) Turning right

Figure 4.25 Turning by bending the spinal joint in the coronal plane: The robot started to walk at (0,0) coordinates and moved for 50 seconds. X_3 is the offset of the spinal joint in the coronal plane, which is negative to turn left and positive to turn right.

Turn in Circles

Regarding turning that encircles the gaits, quadrupedal robots use the spine to turn simply by giving it a bias in the coronal plane without modifying the trajectory of the leg joints. In this gait, the spine is not moved in the transverse and sagittal planes. The results in the simulation showed that the robot could generate efficient and stable

turning gaits when the curvature of the spine in the coronal plane is within plus or minus 0.3 radians. Figure Turn Gait shows the travel trajectories using the turning gait with a bending spine in the simulation. Fig.4.25 shows the travel trajectories using the turning gait with a bending spine in the simulation.

4.6 CPG with Learning Sensory Feedback

The previous section introduced the modified oscillator with an internal feedback mechanism and its capability to generate various gaits through fine-tuning control parameters. However, the CPG model with an internal feedback mechanism can produce natural gaits on flat ground, it faces challenges in generating dynamic gaits on rough terrains and dealing with external disturbances. Since it is not clear how sensory feedback affects gait generation in biology, making it difficult to create an accurate mathematical model to integrate sensory feedback mechanisms to generate dynamic gaits. To address this, a novel approach was proposed that integrates sensory feedback into the CPG model using reinforcement learning, enabling the robot to learn and adapt its gait to rough terrains.

4.6.1 Integrating Sensory Feedback in the CPG

In robotics, sensory feedback influencing gait generation can be categorised into proprioceptive and exteroceptive feedback. Proprioception provides the robot with information about the internal state, such as joint angles, positions, velocities, and accelerations, which is crucial to maintaining stable locomotion. Exteroception involves sensing the external environment through vision, touch, and other environmental variables, helping the robot navigate dynamic surroundings. While both types of feedback are essential, this study focuses on the integration of proprioceptive feedback into the CPG model, as it is critical for maintaining stable gait patterns during locomotion.

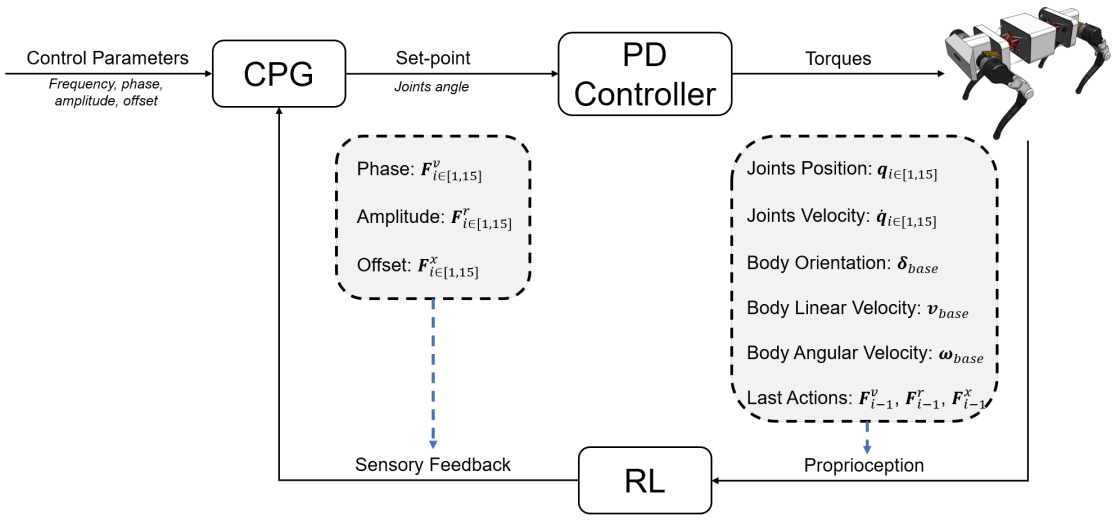


Figure 4.26 CPG with learning proprioceptive feedback. The proprioception obtained from TQbot sensors and then RL policy output feedback terms to CPG to generate dynamic gaits.

Figure 4.26 illustrates the proposed approach that integrates proprioceptive information into the CPG model using reinforcement learning, shaping the gaits generated. This bioinspired framework uses proprioception to modify the phase ϕ_i, amplitude r_i, and offset x_i of the CPG outputs in Equation 3.6, generating dynamic and adaptive gaits suitable for rough terrain. The CPG model and its parameters used here are based on the one presented in Section 4.3.

Importantly, in biological systems, CPGs are often intrinsically coupled, meaning they can generate rhythmic movement patterns without external input. Therefore, in this model, sensory feedback is not used to modify the intrinsic parameters in gaits but to influence the CPG outputs to handle unforeseen situations. Overly strong coupling of sensory feedback with the CPG could lead to unpredictable results, potentially destabilizing the robot's gait. Instead, our approach carefully balances the integration of sensory feedback to enhance gait generation without compromising the stability of the robot.

According to Equation 4.11, the CPG mathematical model that integrates sensory feedback can be written as:

$$\theta_i = (x_i + F_i^x) + f(r_i + F_i^r)cos(\phi_i + F_i^v) \tag{4.17}$$

Where the F_i^x, F_i^r, F_i^v represents the offset, amplitude and Phase feedback respectively. Compared to another research to modulate the intrinsic control parameters, amplitude and phase, of a oscillator [172, 176], the proposed method modularises sensory feedback that CPG can decouple from it and still generate a basic gait for robots. On the other hand, the method has higher robustness than other methods. Due to the unpredictability of CPG coupling with multiple oscillators and complex structures, adding sensory feedback parameters directly to the first-order or second-order differential formulas of amplitude, offset, and phase may lead to coupling disorder, especially when the oscillator needs to change the output value of a certain parameter in a short period of time. For example, when the robot's feet are blocked while swinging, the whole body needs to quickly restore balance before it loses balance and falls. It needs to quickly change the joint position in a short period of time to balance the body posture.

4.6.2 Proximal Policy Optimisation

With the development of reinforcement learning in recent years, many algorithms have been applied to the field of robotics. Among them, Proximal Policy Optimisation (PPO), which has been used many times, is a policy gradient algorithm that is widely used in robot learning [177]. Policy gradient algorithms are very sensitive to step size, and choosing a suitable step size can make training more efficient. The new objective function proposed by the PPO algorithm can enable multiple training steps to achieve small batch updates, solving the problem of difficult step size determination in the policy gradient algorithm. The main objective function is as follow:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{4.18}$$

where the $r_t(\theta)$ is the probability ratio, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. $\hat{A}_t$ is an estimator of the advantage function in the timestep t, which is used to evaluate the performance of an action compared to the average action. ϵ is a hyperparameter to define the allowable range of policy updates. The PPO algorithm as shown below:

Algorithm 1 PPO, Actor-Critic Style

for iteration $= 1, 2, \ldots$ **do**
  for actor $= 1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T time steps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L w.r.t. θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

In this study, PPO was adopted as the algorithm for learning sensory feedback and its parameters and neural network architecture are shown in Table 3.

Table 4.4 PPO Hyperparameters and Neural Network Size

Parameter	**Value**
Batch size	98304 (4096x24)
Mini-batch size	16384 (4096x4)
Number of epochs	5
Clip range	0.2
Entropy coefficient	0.01
Discount factor	0.99
GAE discount factor	0.95
Desired KL-divergence kl^*	0.01
Learning rate α	adaptive
Number of hidden layers (all networks)	3
Number of hidden units per layer	[512, 256, 128]
Activation	elu

4.6.3 Action Space

In the field of reinforcement learning in robotics, the action space is generally a continuous space with the size of the workspace of each joint. In the proposed

CPG model, the actions represent sensory feedback for each oscillator and those are continuous as discrete feedback leads to output discontinuity. For each oscillator, it has amplitude, offset, and phase sensory feedback: F^r, F^x, F^v. The CPG has 15 joint oscillators, thus the entire action space includes 45 actions in total.

4.6.4 State Space

The state space includes two parts: proprioception and the last actions chosen by the policy networks for each oscillator. The proprioception including joint positions q_i, velocities $\dot{q}_i$, orientation of the robot body in roll, pitch, and yaw $\delta_{base} \in \{\delta_{roll}, \delta_{pitch}, \delta_{yaw}\}$, linear velocities $v_{base} \in \{v_x, v_y, v_z\}$ and angular velocities $\omega_{base} \in \{\omega_x, \omega_y, \omega_z\}$. The last actions are F^x_{i-1}, F^r_{i-1}, F^v_{i-1}.

4.6.5 Reward Function

The design of the reward function is to integrate sensory feedback to CPG and generate dynamic and robust gaits based on proprioception. For this purpose, the reward function is designed from the following items:

- Forward Velocity: Reward the robot forward speed, $R(v_x) = v_x$.
- Angular Velocity: Penalise the lateral movement of the robot, $R(\omega_z) = exp(-1 * \omega_z^2)/0.25$.
- Roll and Pitch Velocity: Penalise rapid middle body movement in the roll and pitch directions, $R(\omega_x, \omega_y) = \omega_x^2 + \omega_y^2$
- Z direction Velocity: Penalise middle body movement velocity in z-direction, $R(\omega_{x,y}) = -0.005 * \omega_z^2$
- Joint Acceleration: Penalise joints acceleration, $R(\ddot{q}_i) = -2.5^{-7}(\sum \ddot{q}_i^2)$
- Leg lifting: Reward leg lift, $R(LiftTime) = LiftTime$

- Action Difference: Penalise change action, $R(action_t, action_{t-1}) = -0.005 * (\sum(action_{t-1} - action_t)^2)$
- Collision: Penalise body collision, $R(Collision) = -1$

The total reward is obtained by summing all the above rewards, which the formula is:

$$\begin{aligned} R(total) =& R(v_x) + R(\omega_z) + R(\omega_x, \omega_y) + R(\omega_{x,y}) + R(\ddot{q}_i) + \\ & R(LiftTime) + R(action_t, action_{t-1}) + R(Collision) \end{aligned} \tag{4.19}$$

4.7 Summary

The TQbot project explores an innovative approach to gait generation using Central Pattern Generators integrated with internal sensory feedback and reinforcement learning. Unlike traditional model-based methods that rely on accurate kinematic and dynamic modeling, this project adopts a model-free strategy using CPGs to generate natural gaits. The CPG model initially proposed has limitations in handling dynamic terrains and external perturbations, which prompts the integration of sensory feedback to improve adaptability and stability.

The CPG model incorporates proprioceptive feedback -specifically joint velocity and acceleration - to shape rhythmic gaits through reinforcement learning, allowing the robot to learn how to adjust its movements dynamically. This approach is grounded in a biological analogy in which sensory inputs do not tightly couple with basic CPG outputs but instead modulate them to adapt to unexpected changes in the environment.

Simulation environments such as Isaac Sim and Isaac Gym provide the necessary platform for developing and testing these learning-based CPG models, leveraging high-performance simulation tools to refine robot gait control policies. Through reinforcement learning, the gait dynamics of the robot are continuously optimised, with careful consideration of action and state spaces, and a reward function designed to encourage efficient and stable locomotion.

Chapter 5

Experiments on Bio-inspired Spine

5.1 Introduction

The previous chapters have presented the completed TQbot mechanical design and its control and gait generation methods. The bio-inspired tensegrity spines proposed in the thesis are expected to improve gaits. Therefore, the leg-spine coordinated gaits generated by the CPG and the different spine movements in terms of stability and speed in each gait will be evaluated using a quantitative method. The role of the spine in basic gaits will be explored in simulation, where spine motion will be autonomously generated using reinforcement learning and compared to the CPG-based results. The results of CPG with the learning proprioceptive feedback model were also demonstrated.

The spine plays a critical role in quadrupedal animals, and the novel tensegrity spine proposed in the thesis is also expected to improve gaits. However, a number of preliminary validation experiments require a high precision and computational efficiency simulation, which is introduced in Section 5.2. In Section 5.3, two metrics, stability value and travel distance, to quantify gait performance are described, also all simulated results are demonstrated. Section 5.4 presents the results of the control method for two configurations of the tensegrity spine in real-world scenarios.

5.2 Simulation Environment

In this work, Isaac Sim and Isaac Gym are used for TQbot simulation [6]. These are high-performance simulation platforms developed by Nvidia, both of which use PhysX as their physics engine. Isaac Sim provides high-fidelity physical and visual simulations, making it suitable for verifying robot algorithms and developing perception tasks, while Isaac Gym focuses on reinforcement learning, accelerating robot strategy learning and training by implementing large-scale parallel simulations on a single GPU. Both platforms support importing URDF and MJCF files with automatic convex decomposition of imported 3D meshes, which is convenient for importing customised

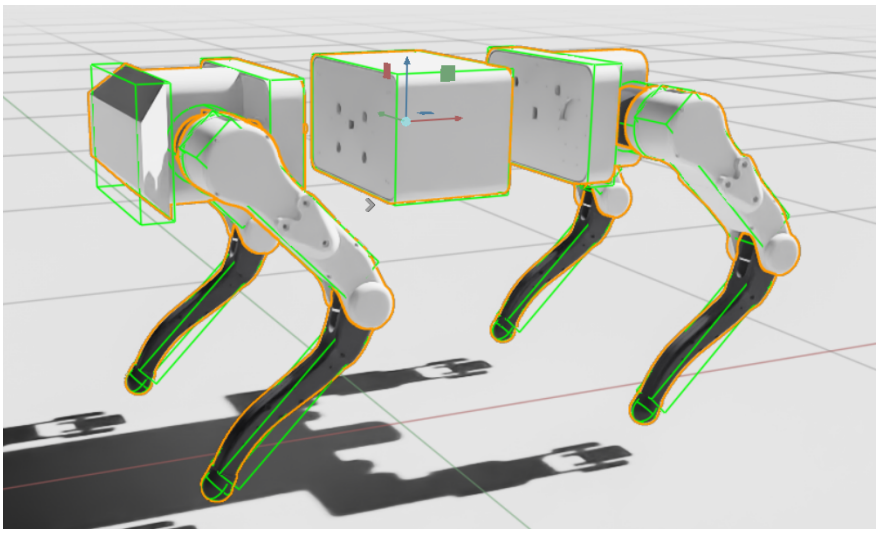


Figure 5.1 TQbot collision model in Isaac Sim. TQbot presents the default posture, and the green border is the collision model.

robot platforms for physical simulation. These two platforms combine high-precision simulation with efficient computational power, offering robust support for robotics research and applications.

Understanding how to use the spine to enhance speed and stability is crucial for quadruped robots using different gaits. However, due to the limitations of the mechanical structure, no research has yet explored the effect of the spine on gait when it is driven simultaneously with multiple degrees of freedom. Therefore, reinforcement learning is applied here to allow the robot to learn spine movements that improve stability and speed across the four basic gaits.

5.2.1 Simulated Model and Reality Gap

The TQbot model is represented by a unified robotics description format (URDF) file, which stands for Unified Robot Description Format. It is an XML format used to represent a robot model with links and joint motions. The visual model is exported directly from 3D modeling software in STL format. The collision model consists of simple geometries, such as cubes and spheres of different sizes, and is defined within

the URDF file. Additionally, the kinematic model is consistent with the description provided in Chapter 3. The mass of each simulated part, including the batteries, control board, and other components, is consistent with the actual mass.

The experimental results in Section 5.4.2 indicate that the high stiffness of the passive part in the first spine configuration significantly restricts movement in all directions, causing it from accurately achieving the desired angles. Therefore, all simulation results were performed with the second spine configuration.

Figure 5.2 shows the URDF model structure, the leg joint kinematics are created by adding "revolute" type joint and its characteristics, such as maximum torque and velocity, are the same as the corresponding actuator. Each spine joint have three "revolute" type joints to achieve three DOFs rotation and each of them connected by a link without distance. In simulation, the motion of the front and rear spine joints is hard-coded to move simultaneously, and their movement in all directions is consistent with the second spine configuration. However, the simulation used in the thesis cannot simulate the physical properties of ropes or lines, thus there are inevitably a certain degree of reality gaps. To the best of the author's knowledge, there is no simulation that can simulate the characteristics of ropes while supporting the robot simulation and reinforcement learning. Nevertheless, the reality gap may be further reduced in the simulation after obtaining some real data in future experiments.

5.2.2 Sensors

A wide variety of sensors are provided in both Isaac Sim and Isaac Gym, including radar, cameras. The main sensory feedback used in this study is proprioception, that is, joint position, joint velocity, and robot attitude. All of this information is available directly from the simulation and can be easily accessed physically through highly integrated motors and IMUs.

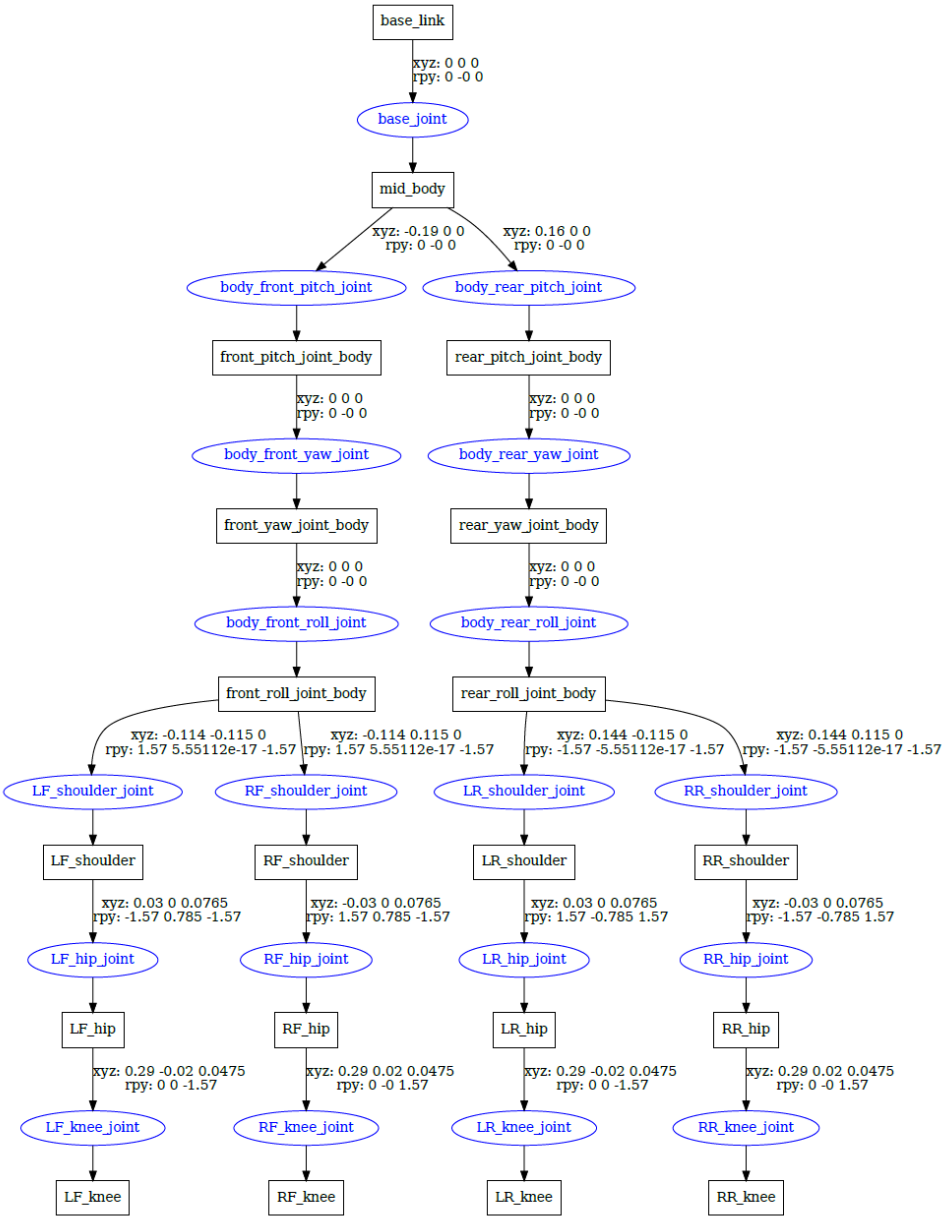


Figure 5.2 TQbot URDF model structure. LF: Left Front, RF: right front, LR: left rear, RR: right rear.

5.3 Simulation Results

For quadruped robots using different gaits, what kinds of spinal movement can increase speed and maintain stable is a critical problem. However, due to the limitations of the mechanical structure, no research has yet explored the effect of the spine on gait in a spine that is driven simultaneously with multiple degrees of freedom. To investigate this problem, a total of three experiments from simple to complex were conducted in the simulation:

(1) Spine phase difference in trot gait: exploring the impact of various phase differences of the spine in the roll and yaw directions on the trot gait;

(2) Using PPO to learn spinal movement and to investigate the role and performance of spinal movements in four basic gaits: walk, trot, pace, and bound;

(3) Based on the CPG with learning proprioceptive feedback, to explore the functionality of the spinal movement in dynamic trot gait on uneven terrain.

The robot in the first two experiments moves on flat ground, whereas the third moves on uneven terrain. In all experiments, robot movement is restricted to the forward direction only and each episode takes 20 seconds in simulation, that is 1000 iterations. In addition, The spine configurations used in the simulations are all of the second type due to the bionic roll motion. The simulation setup and quantification methods are described in detail in this section.

5.3.1 Stability Value and Travel Distance

In nature, the spine usually plays a role in speed and stability. Similarly, it is hypothesised that the TQbot spine can also play a role in these two aspects. Therefore, a systematic quantitative method is proposed here to evaluate the stability and performance of the robot gaits. The stability value is calculated as follows:

$$S = \frac{1}{N}\sum_{t=1}^{N}(\Delta\alpha_t + \Delta\beta_t) + max(|\alpha|) + max(|\beta|) \tag{5.1}$$

with

$$\Delta\alpha_t = \alpha_t - \alpha_{t-1} \tag{5.2}$$

$$\Delta\beta_t = \beta_t - \beta_{t-1} \tag{5.3}$$

where α_t and β_t are the angles of roll and pitch DOF of the middle body segment at time t respectively. $\Delta\alpha_t$ and $\Delta\beta_t$ are the angle differences between the last and the current timestep: $t-1$ and t. The last two items are the absolute values of the maximum movement angles in the roll and pitch directions. It can be observed that the stability of TQbot is measured by the movement of its middle body segment in the roll and pitch angles. During locomotion, the middle body segment is expected to remain as stable as possible, while the front and rear body segments adjust their posture. Therefore, the average movement angles of the middle body segment, as well as the maximum movement angles in both the roll and pitch directions, are used to quantify gait stability. A higher stability value indicates a greater degree of body movement, while a lower value means a more stable gait.

In addition, the robot is also expected to move as efficiently as possible in a stable gait. Therefore, the travel distance is also used as a metric to evaluate the gait, the formula is as follows:

$$d = \sum_{t=1}^{N}\sqrt{\Delta x_t^2 + \Delta y_t^2} \tag{5.4}$$

with

$$\Delta x_t = x_t - x_{t-1} \tag{5.5}$$

$$\Delta y_t = y_t - y_{t-1} \tag{5.6}$$

where x_t and y_t are the TQbot position coordinates of the robot in the global coordinate system. The above formula calculated the travel distance of the TQbot in an experiment.

5.3.2 Spinal Movement in Trot Gait

In this experiment, the traversal method was used to explore the effect of spinal motion with various roll-yaw phase differences on trot gait. The movement of the spine in the pitch direction was not used and was always set to the default position. However, the leg joints configuration is the same as that in Chapter 4, and in this experiment the legs only generate rhythmic motion. Therefore, the movement of spinal joints should also be set within a reasonable range, otherwise it may cause the robot to fall.

Amplitudes in Trot Gait

While the phase of spinal motion in the trot gait can be roughly estimated by observing the animal motion videos in Chapter 4, its amplitude remains less clearly defined. Therefore, when investigating spinal movements in the roll and yaw directions, it is necessary to first determine the appropriate amplitude. Here, an initial experiment was carried out to explore the appropriate amplitude values with the robot tested in simulations using amplitudes of $0, 2, 3, 4, 6, 8$. Amplitudes greater than 8 were not tested, as excessive spinal movement would cause the robot to lose stability and fail to generate an effective gait.

In this experiment, the roll-yaw phase difference was varied from 0 to $\frac{3\pi}{2}$ in $\frac{\pi}{2}$ increments across different amplitude values to collect data. Amplitude = 0, indicating no spinal

movement, was used as a baseline to evaluate the performance of the other amplitude values. Performance was assessed in terms of travel distance and stability value. As

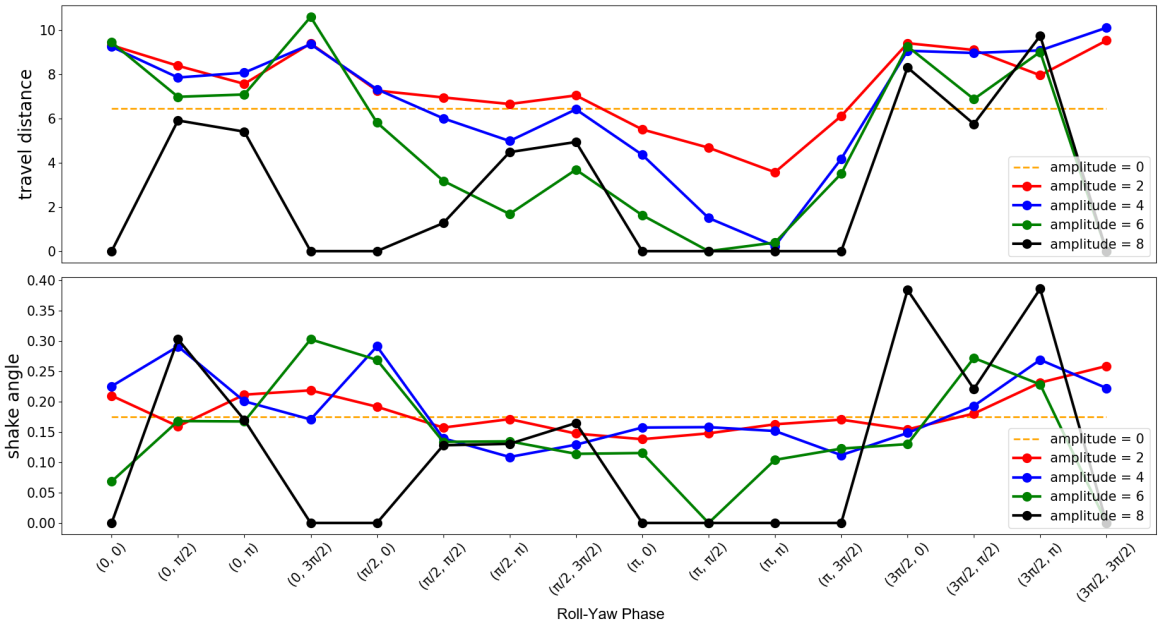


Figure 5.3 Different amplitudes performance, travel distance(top) and stability value(bottom), in roll-yaw phases. The x-axis represent the roll-yaw phase from $(0, 0)$ to $(\frac{3\pi}{2}, \frac{3\pi}{2})$.

shown in Figure 5.3, when the roll phase of the spinal movement is $\frac{\pi}{2}$ and π, the TQbot travel distance is lower than that of the trot gait without spinal movement. The reason is, during these phases in the roll direction, the spinal motion causes the robot's shoulder to dip when the leg is lifted. As the amplitude increases, the legs fail to lift to a sufficient height during walking, and at amplitude = 8, the legs fail to lift at all. Leading the robot generating an invalid gait that does not allow it to move forward. At the same time, the spinal movement in the yaw direction becomes irrelevant due to legs not lifted during swing phase. Also phase changes in the yaw direction cannot improve gait performance. Consequently, since the robot did not move significantly, the middle body segment experiences less shaking, resulting in lower stability values compared to the gait without spinal action, indicating that the body is more stable and shakes less.

When amplitude reaches 8, the effectiveness of the gait is compromised. The data shows that, at amplitude = 8, the performance of the trot gait is almost entirely worse than the trot gait without spinal movements. Especially when roll phase=π and amplitude=8, a trot gait makes the robot move backwards. For amplitudes of 2, 4, and 6, the gait performs as expected at roll phase = 0 and achieves optimal results at yaw phase = $\frac{3\pi}{2}$, with walking speed increasing as amplitude increases. Notably, when the roll phase = $\frac{3\pi}{2}$ the gait outperforms the gait without spinal movement in terms of travel distance, but only at amplitudes of 2 and 4.

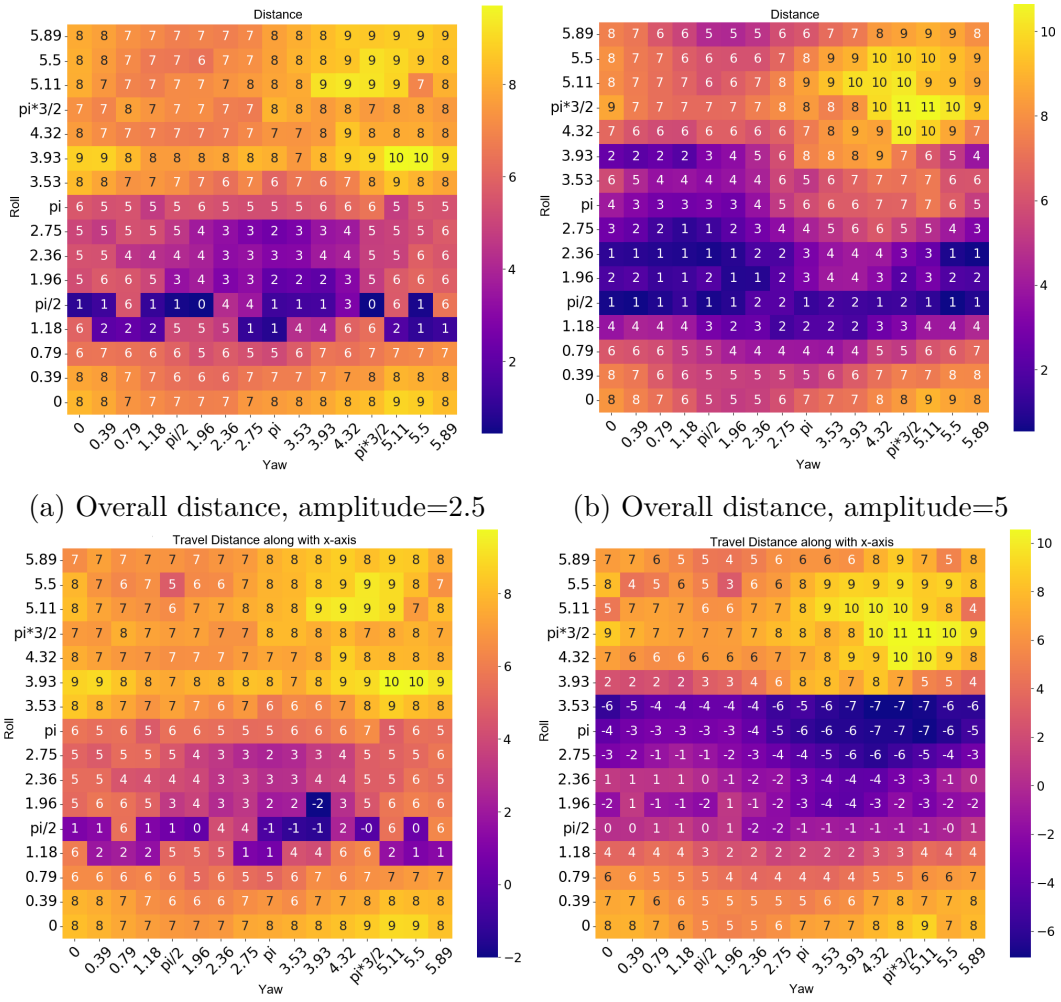
However, in all phase combinations that generate valid gaits, the stability value indicates that the gait with the spinal movement is less stable than the gait without the spinal movement. It may be because the connections of the spine are rigid in the simulation, and as mentioned in Chapter 2, a high-stiffiness spine connection can lead to unstable gaits.

Based on the above experimental results, it can be preliminarily obtained that under the CPG parameters set in Section 4.5.2, the effective range of amplitude of spinal movement should be set between 0-6. An amplitude greater than 6 would cause the gait to become unstable to the point of falling.

Roll-Yaw Phases in Trot Gait

Based on the results obtained in the previous experiment, in order to investigate the effect of different spine phases in the roll and yaw directions during the trot gait, two amplitude values, 2.5 and 5 degrees, were selected for the experiment. In the experiment, the amplitudes of the spine joints in roll and yaw are set to the same. The experimental results will be evaluated in terms of both travel distance and stability. Figure 5.4 shows the distance results. Since the travel distance measures the total distance travelled by the robot without accounting for directionality, the distance travelled along the x-axis is used as a secondary reference.

DISTANCE



(a) Overall distance, amplitude=2.5

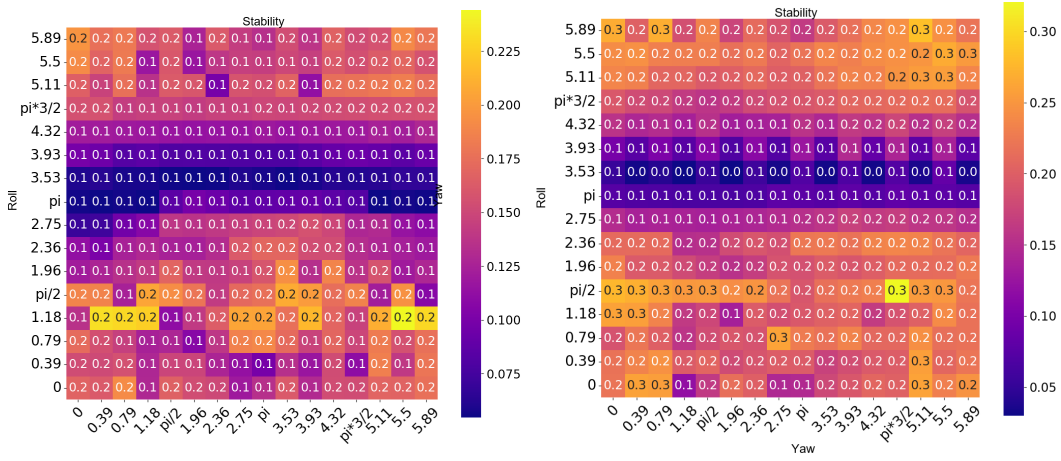
(b) Overall distance, amplitude=5

(c) Travel distance along with x-axis, amplitude=2.5

(d) Travel distance along with x-axis, amplitude=5

Figure 5.4 Two amplitudes results: the robot travel distances (in meters) generated by different roll-yaw phases of spinal movement. The long travel distance correspond to brighter color.

As can be seen from Figure 5.4, the maximum travel distance can be approximately 11 with amplitude=5, while the maximum travel distance is 9 with amplitude=2.5. In terms of the yaw phase, it can be seen that amplitude = 2.5 causes the robot to move further than amplitude = 5 when the yaw phase is close to $\frac{\pi}{2}$, while the robot with amplitude=5 moves further when the yaw phase is close to $\frac{3\pi}{2}$. This is due to the fact that spinal movement with a $\frac{3\pi}{2}$ phase difference in the yaw direction increases the step length, while its opposite, the $\frac{\pi}{2}$ phase difference, shortens the step length. It is worth noting that the manually fine-tuned phase of $(0, \frac{3\pi}{2})$ in Section 4.5.2 did not make the robot move the longest distance. Regardless of the value of amplitude, the overall distance near the phase difference $(\frac{3\pi}{2}, \frac{3\pi}{2})$ is the farthest and the gait is valid. In Figure 5.4c and 5.4d, when the roll phase lag between $\frac{\pi}{2}$ and π, the overall distance is shorter than other phases, which cannot produce a efficient forward gait. As can be seen in Figure 5.4d, the gait generated by the phases between $\frac{\pi}{2}$ and π causes the robot move to backward. For yaw phase, the robot can generally move farther when it is greater than pi and vice versa.



(a) Roll vs Yaw phase, amplitude = 2.5 (b) Roll vs Yaw phase, amplitude = 5

Figure 5.5 Stability values for trot gait with two amplitudes 2.5 (left) and 5 (right) varying with changing roll-yaw phases. The brighter color (higher stability value) represents the roll-yaw phase generate unstable trot gait.

STABILITY

From the perspective of stability, phase and amplitude values that result in a high speed lead to lower overall stability of the robot. This finding aligns with the analysis from the Roll-Yaw amplitude experiments, where the shaking of the middle body segment is minimised when the roll phase is near π. As previously discussed, setting π phase in roll direction prevents the robot from adequately lifting its legs during the swing phase, which is further cooroborated by the reduced travel distance in the results. The manually fine-tuned phase lag $(0, \frac{3\pi}{2})$ exhibit slight differences in stability compared to the phase, $(\frac{3\pi}{2}, \frac{3\pi}{2})$ that achieve the longest travel distance. Specifically, their stability values are around 0.2 when amplitude is 2.5; and around 0.3 when amplitude is 5. Gaits with a roll-yaw phase of $(\frac{3\pi}{2}, \frac{3\pi}{2})$ show minimal differences in stability across these two amplitudes, both approximately around 0.2.

Figure 5.6 illustrates the middle body segment's movement angles in roll and pitch direction when amplitude = 5 for two different roll-yaw phase configurations: $(\frac{3\pi}{2}, \frac{3\pi}{2})$ and manually tuned phase parameters $(0, \frac{3\pi}{2})$. From the figure, it is evident that the optimal phase derived through exhaustive search results in a similar range of body oscillation in the roll direction, but significantly reduced oscillation in the pitch direction, remaining almost entirely within the range of $[0, 0.06]$ in radians.

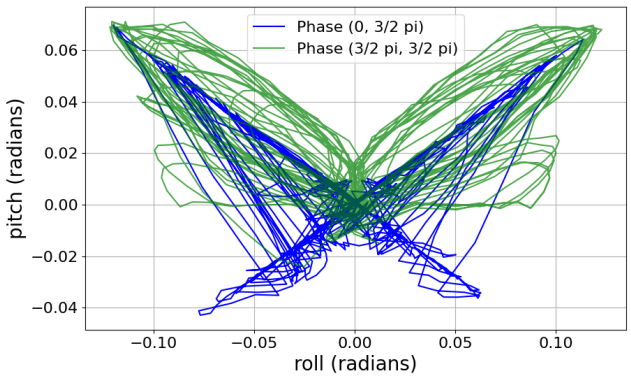


Figure 5.6 Roll vs pitch angle over (in radians) 20 seconds. Blue is manual roll-yaw phase $(0, \frac{3\pi}{2})$ and green is best performance phase $(\frac{3\pi}{2}, \frac{3\pi}{2})$

This experiment explored the impact of spinal motion on the trot gait of a quadruped robot, with particular focus on roll and yaw orientation. Initially, the effects of spinal joint amplitudes (ranging from 0 to 8) on gait performance were investigated, leading

to the selection of two appropriate amplitude values: 2.5 and 5. These values were then used to explore the influence of spinal movement in the roll and yaw orientation on the trot gait. The results indicated that amplitudes greater than 6 significantly increased the probability of the robot falling, while the optimal spinal roll and yaw phases for performance were both $\frac{3\pi}{2}$. The experiment demonstrates that different phase and amplitude combinations can significantly affect gait distance and stability, highlighting the critical role of spinal motion in improving gait performance. The baseline for each gait is the mean and standard deviation of the results obtained by running 10 times with default parameters and no spine movement.

5.3.3 Learning Spinal Movement in Basic Gaits

From the previous preliminary experiment, it is known that the spine of this structure has a significant positive gain in the trot gait of the quadruped robot. It is also expected that the bio-inspired spine can improve the speed and stability of different gaits. Therefore, here the PPO algorithm, introduced in Section 4.6, is used to enable TQbot to learn spinal movement to improve the stability and speed of the four basic gaits and control the speed by adjusting the spine according to commands.

In this experiment, the phase and amplitude values of all gaits are consistent with those set in Section 4.5.2. Different offset values are set for the basic gaits to form an effective spineless gait, as shown in Table 5.1. the command speed ranges of the four gaits are set as follows. The lower bound of the speed is set to the speed of the gait without spinal movement, while the upper bound is uniformly set to 1.5 m/s. In addition, each robot was commanded to move forward, with a range consistent with its gait range and command speed interval is 0.01. Each speed command was run ten times and the results were collected.

- Walk gait: $[0.35, 1.5], m/s$
- Trot gait: $[0.5, 1.5], m/s$

- Pace gait: $[0.35, 1.5], m/s$
- Bound gait: $[0.5, 1.5], m/s$

Table 5.1 Joint offsets. LF: left front, RF: right front, LR: left rear, RR: right rear

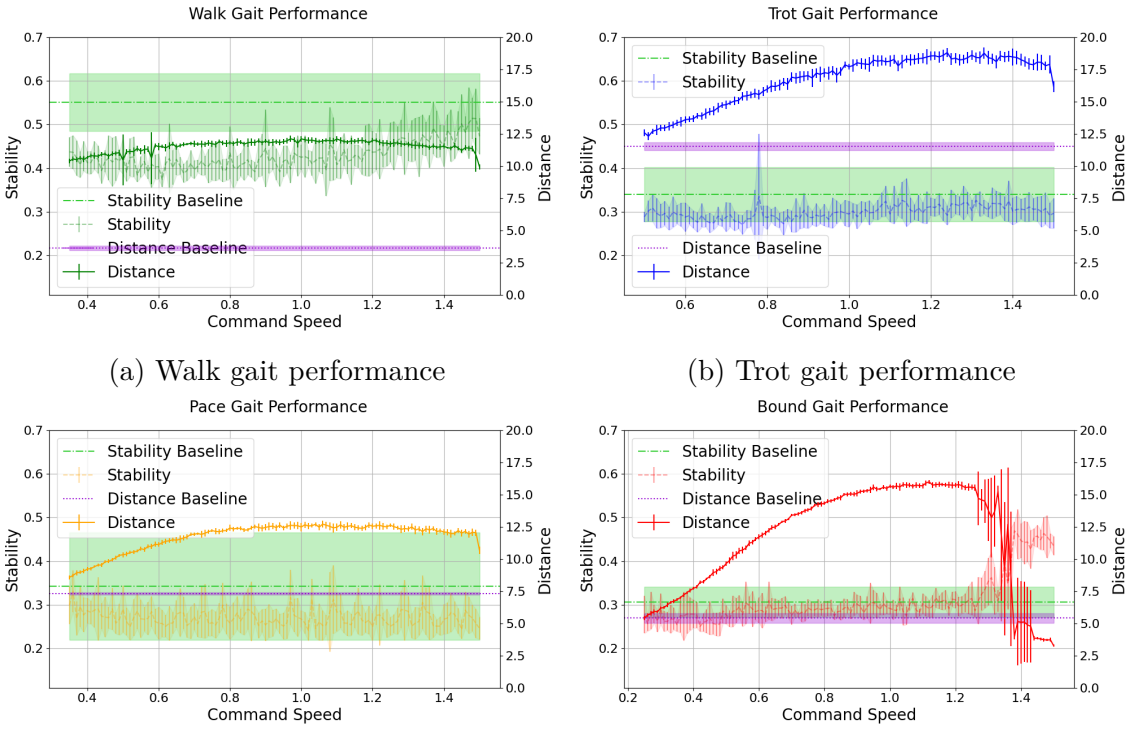
Gait	**Joint**	**LF**	**RF**	**LR**	**RR**
Walk	Hip	0.175	0.175	-0.26	-0.26
	Knee	-0.349	-0.349	-0.175	-0.175
Trot	Hip	-0.087	-0.087	-0.087	-0.087
	Knee	0	0	0	0
Pace	Hip	-0.087	-0.087	-0.087	-0.087
	Knee	0	0	0	0
Bound	Hip	0.087	0.087	-0.087	-0.087
	Knee	0	0	-0.1745	-0.1745

Action, State and Reward Function Setting

In this experiment, the state space is the same as that introduced in Section 4.6. The Foward Velocity term in the reward function is changed to $R(cv_x) = \frac{exp(-1*(cv_x - v_x)^2)}{0.25}$, which means that the robot follows the command velocity cv_x. The action space is continuous, including three rotational orientations in roll, pitch, and yaw: $\{A_{roll}, A_{pitch}, A_{yaw}\}$.

Learning Performance

From the results in Figure 5.7, the spine has a positive effect on improving the performance of trot and bound gaits but has a limited impact on walk and pace gaits. In terms of stability, the walk gait becomes unstable when combined with spinal movement, exhibiting the greatest body oscillation among the four gaits. When the command speed exceeds 1.2, the bound gait not only becomes unstable but also prone to losing balance and falling. For the trot gait, when the command speed exceeds 1.2, there is no significant change in speed or stability, but once the speed reaches 1.5, performance suddenly deteriorates. The pace gait shows no further speed increase beyond a command speed of 1.0, and the speed of the walk gait stops increasing at



(a) Walk gait performance

(b) Trot gait performance

(c) Pace performance

(d) Bound gait performance

Figure 5.7 Basic Gaits performance in terms of travel distance and stability: the performance corresponding to each command speed is calculated by running the results ten times. The scales for stability and distance are on the left and right sides respectively. The violet dashed line is the distance baseline and the light green dashed line is the stability value baseline.

around 0.7. The reason for the sudden performance drop in all gaits after reaching a certain command speed is that the leg movements of these gaits are rhythmic. As the robot approaches maximum speed of a gait, it requires adjustments in leg movements, rather than relying solely on spine movements, to continue improving speed. Moreover, an interesting point is that before the gaits' ultimate speed, the increase in robot travel distance has little effect on stability. It suggests that dynamic spinal movement could potentially increase the robot's forward speed while maintaining stable.

Spinal Movement Analysis

Each spine movement figure shows the mean and standard deviation of the spinal move in the roll, pitch and yaw directions of the four gaits of the best performance. These two values can show the trend of spine amplitude changing with speed.

In terms of travel distance, the spinal motion is most helpful for trot gait and bound gait, while for walk gait and pace gait, the spinal movement is to adjust the posture so that the rhythmic leg action can move efficiently.

WALK GAIT Figure 5.9a presents the average angle and standard deviation of the spinal motion in the roll, pitch, and yaw directions during the walk gait. From the figure, it is evident that the learned spinal movement exhibits minimal offset in the roll and yaw directions, approximately 0.025 radians and 0.04 radians (1.4 degrees and 2.3 degrees), respectively. The standard deviation in all directions is approximately 0.025. Since the walk gait without spinal involvement generates little effective movement, the spine plays a crucial role in adjusting posture in this case. Figure 5.9b shows the spine joint trajectory in the roll, pitch, and yaw directions when the command speed is 0.8, which is the learned optimal walking gait. The range of spinal movement is most significant in the pitch direction, with an amplitude of about 0.1 radians (about 5 degrees). The yaw direction is second, and the roll direction has the smallest amplitude. Moreover, these three directions are periodic and coupled with the periodic movement of the legs.

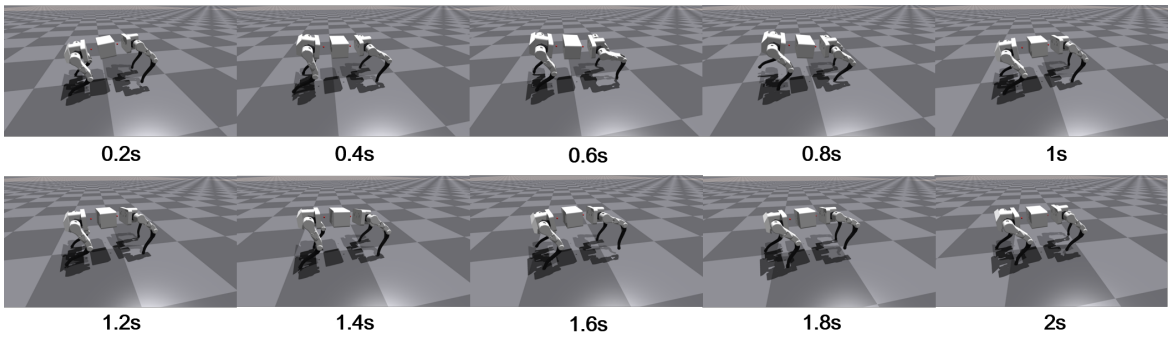
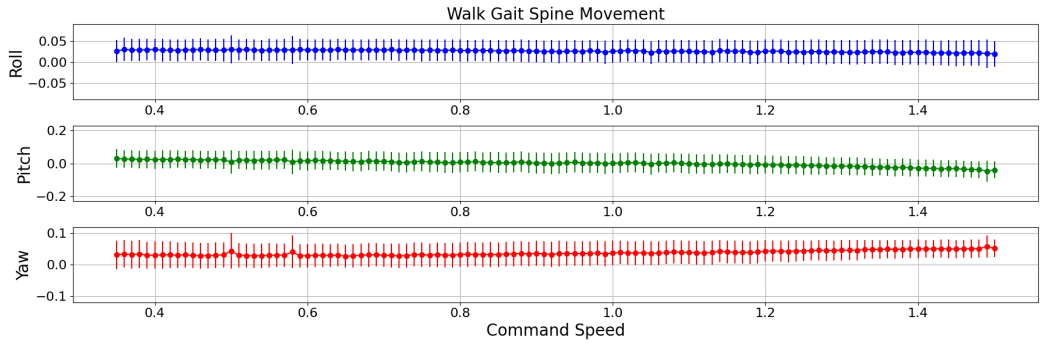
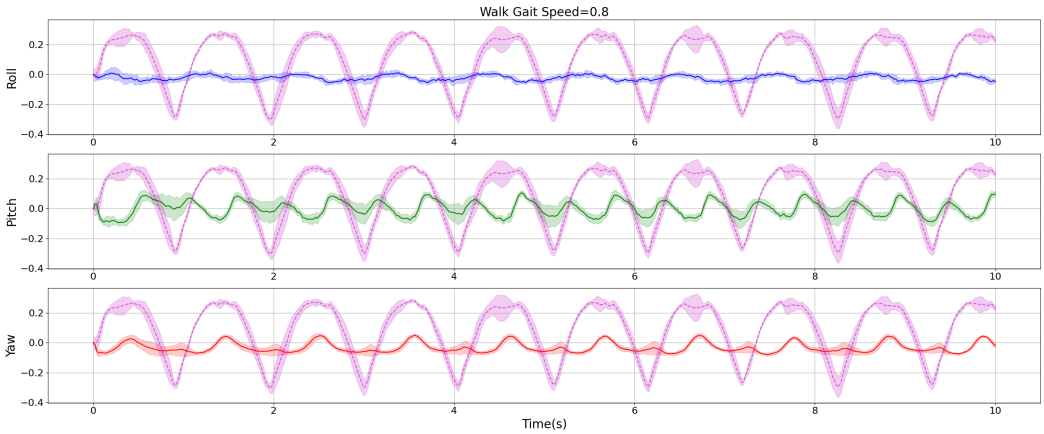


Figure 5.8 Screenshots of learned spinal movement for walk gait between 0.2 and 2 seconds



(a) Average angle with standard deviation of spinal movement versus commanded speed (in m/s)



(b) Spinal joint and left front hip joint trajectories (in radians), commanded speed=0.8 m/s

Figure 5.9 Learned spinal movement in walk gait: spine joint trajectory in roll, pitch and yaw direction are represented by blue, green and red lines separately. The purple lines are left front hip joint trajectory as a reference. Each trajectories are obtained by calculating the average results of ten runs, the shaded area and bar are the standard deviation

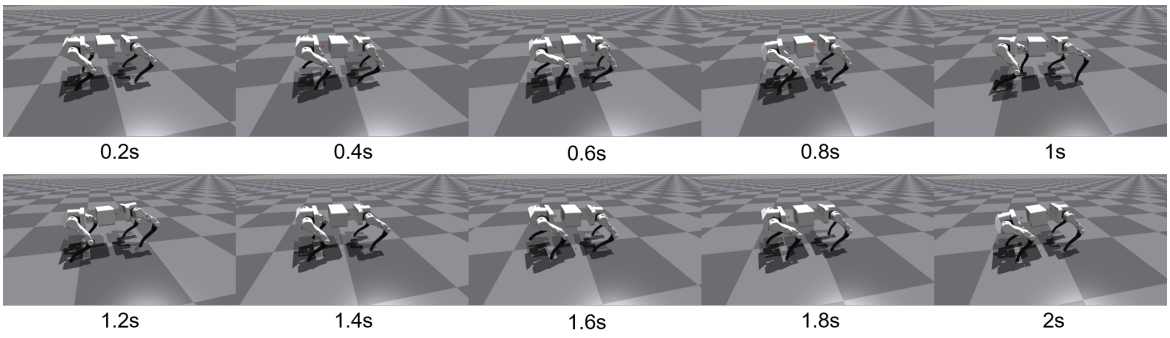
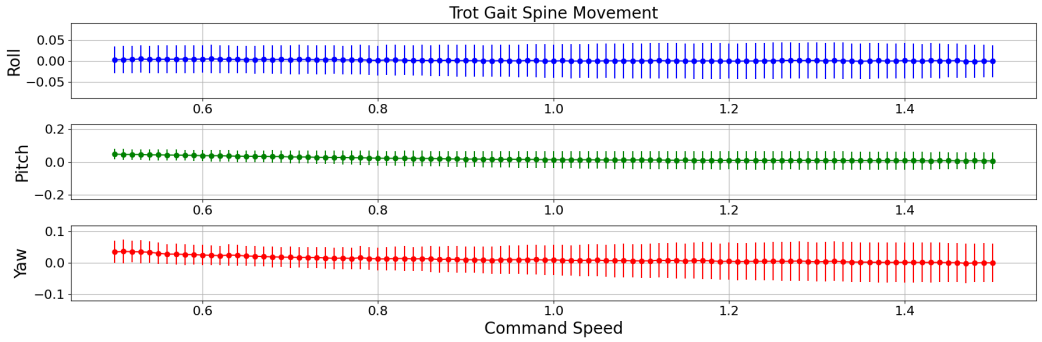


Figure 5.10 Screenshots of learned spinal movement for trot gait between 0.2 and 2 seconds

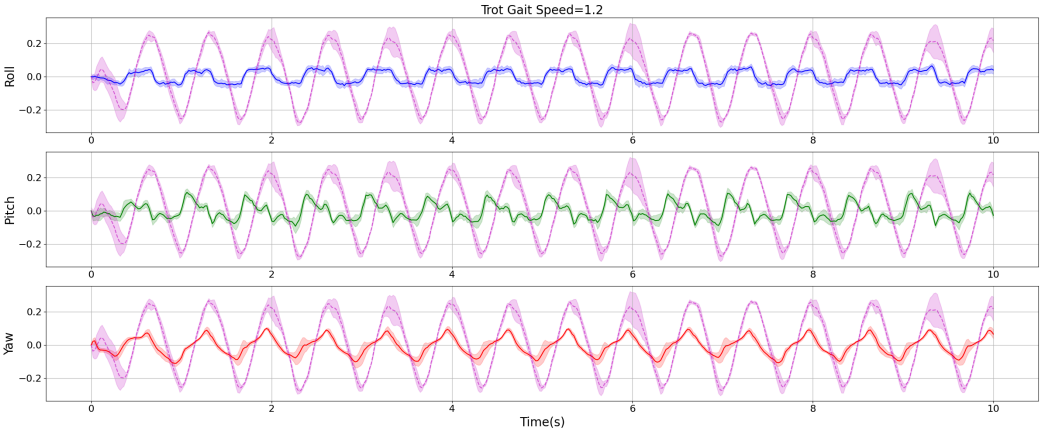
TROT GAIT For the trot gait, it can be seen from Figure 5.11a that the spinal movement almost does not have offset across all rotational directions, and as the command speed increases, the amplitudes in the roll and yaw directions also increase. It confirms that, for the trot gait, movements in the roll and yaw directions play a significant role. Compared with previous experiments, reinforcement learning enables the robot to consider movement in the pitch direction. Notably, after incorporating movement in the pitch direction, the robot does not become more unstable as forward speed increases. Also shown in Figure 5.11b, the stability value remains approximately 0.3 when the command speed is below 1.2. This suggests that spinal movement in the pitch direction may contribute to maintaining stability.

Figure 5.11 shows the movement of the spine in the trot gait when the command velocity is 1.2. It can be clearly seen that the amplitude in the roll direction is significantly smaller than the amplitude in the yaw and pitch directions.

PACE GAIT As for pace gait, except for the roll direction, which has almost no offset, the pitch and yaw directions have offsets, shown in Figure 5.13a. Before the command speed is 1.0, the amplitude change of the spine movement in the three directions is relatively small compared to the offset change. And there is almost no significant change in the roll direction, while the main changes are in the offset in the pitch and yaw directions. This shows that spinal movement in these two directions may have an impact on pace gait. Figure 5.13b demonstrates that the offsets in the pitch and yaw



(a) Average angle with standard deviation of spinal movement versus commanded speed (in m/s)



(b) Spinal joint and left front hip joint trajectories (in radians), commanded speed=1.2 m/s

Figure 5.11 Learned spinal movement in trot gait: spine joint trajectory in roll, pitch and yaw direction are represented by blue, green and red lines separately. The purple lines are left front hip joint trajectory as a reference. Each trajectories are obtained by calculating the average results of ten runs, the shaded area and bar are the standard deviation

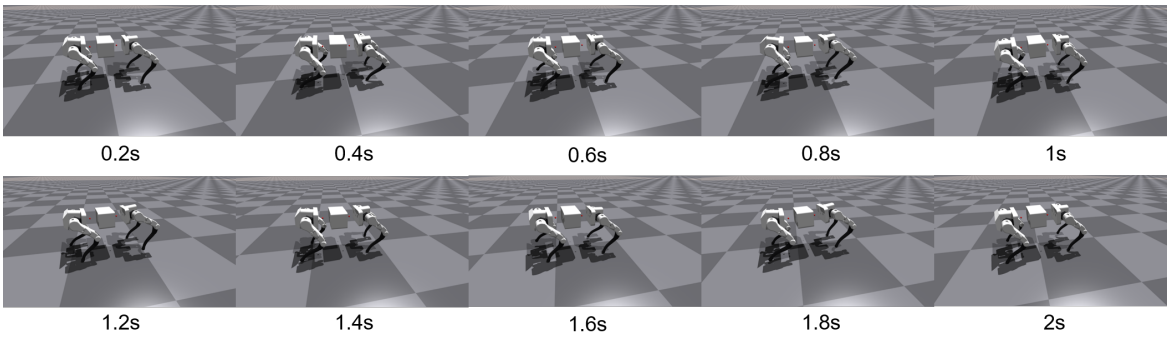


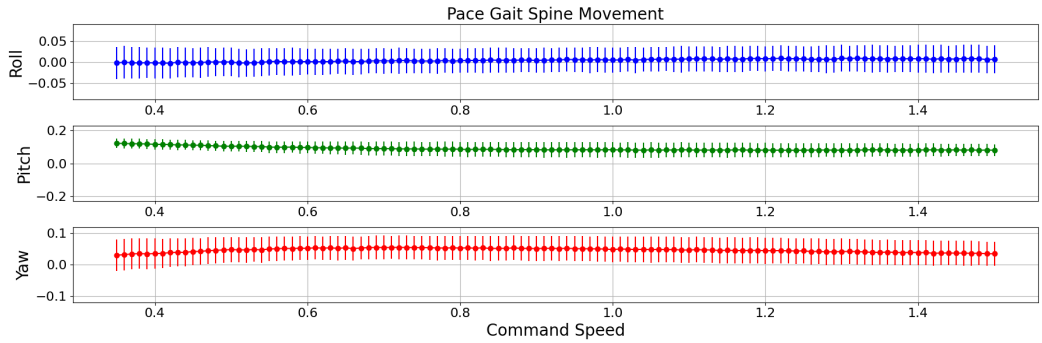
Figure 5.12 Screenshots of learned spinal movement for pace gait between 0.2 and 2 seconds

directions of pace gait at the optimal command speed of 1.0 have increased by about 0.1 and 0.05 radians respectively. And the amplitudes in these two directions are both about 0.05 radians. The amplitude in the roll direction is about 0.025 radians.

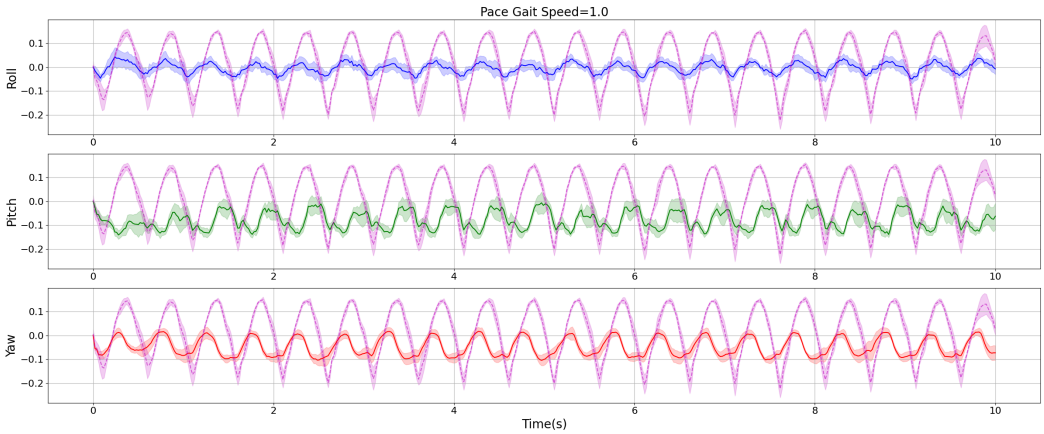
BOUND GAIT Compared to the other three gaits, the spine has the greatest impact on the bound gait. Without spinal movement, bound gait struggles to generate effective motion with only rhythmic leg movements for the robot. As shown in Figure 5.15a, spinal movement in the yaw direction is almost negligible, while movement in the pitch direction is the most obvious. In the command speed range of 1 to 1.2, where performance is optimal, there is minimal offset in the pitch direction, and the amplitude is larger than in the other two directions. The spine's movement in the roll direction likely plays a role in adjusting posture during this gait.

Figure 5.15b shows the spinal movement of TQbot using bound gait when the command velocity is 1.2. It is evident that there is minimal movement in both the roll and yaw directions, with amplitudes and offsets in these directions being extremely small, approximately 0.02 radians (1 degree). The primary movement of the spine occurs in the pitch direction, with an offset of around -0.05 radians (3 degrees), resembling a bending motion. The amplitude in the pitch direction is also the largest among the four gaits, measuring approximately 0.15 radians (8.5 degrees).

According to the experiments described above and their analysis, the findings reveal that spinal action significantly enhances trot and bound gaits, but has limited impact



(a) Average angle with standard deviation of spinal movement versus commanded speed (in m/s)



(b) Spinal joint and left front hip joint trajectories (in radians), commanded speed=1.0 m/s

Figure 5.13 Learned spinal movement in pace gait: spine joint trajectory in roll, pitch and yaw direction are represented by blue, green and red lines separately. The purple lines are left front hip joint trajectory as a reference. Each trajectories are obtained by calculating the average results of ten runs, the shaded area and bar are the standard deviation

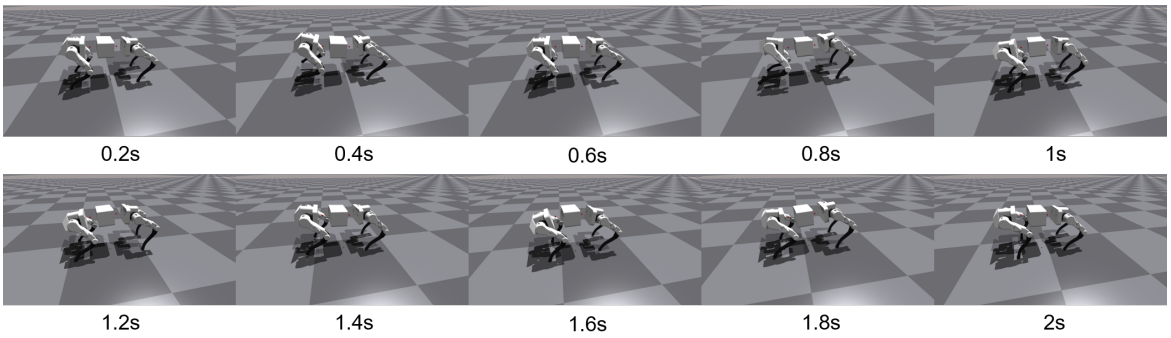
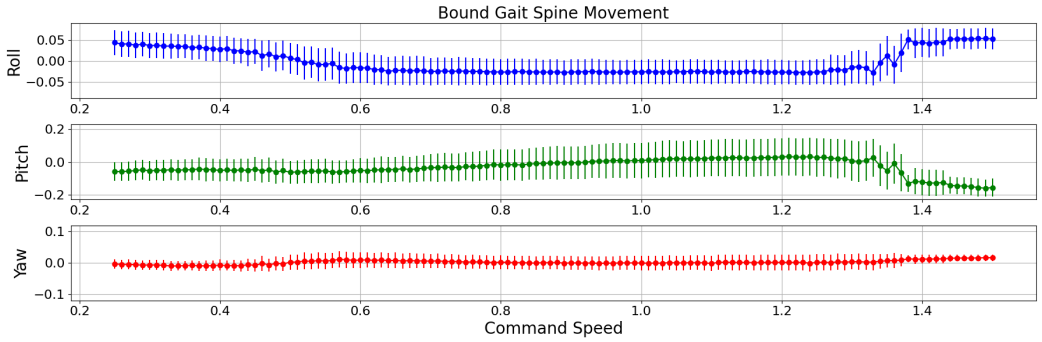
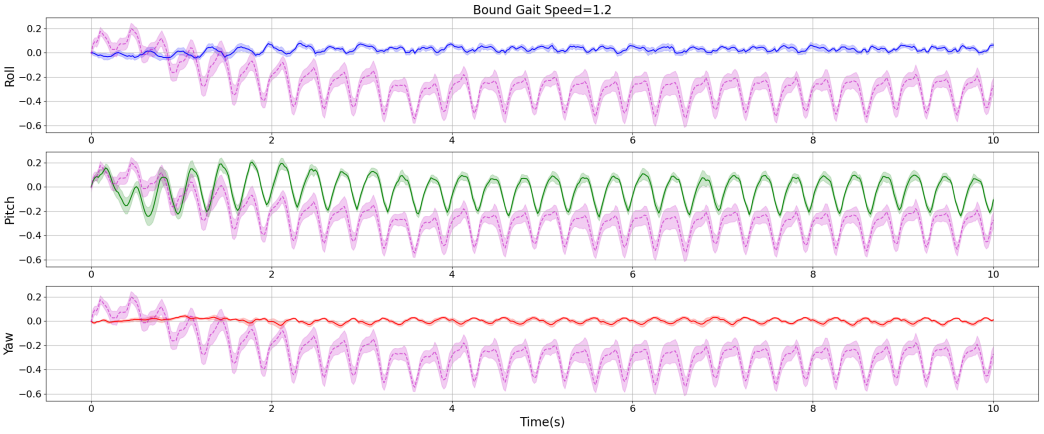


Figure 5.14 Screenshots of learned spinal movement for bound gait between 0.2 and 2 seconds

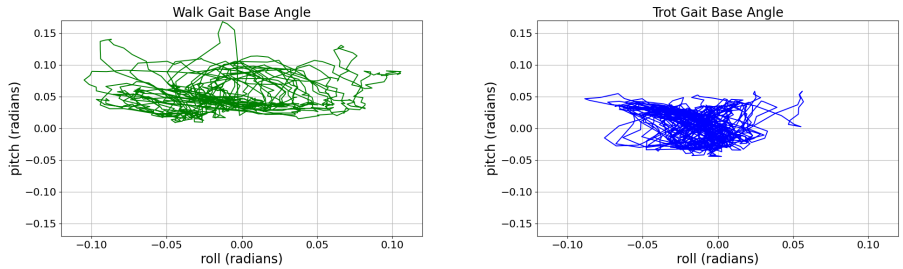


(a) Average angle with standard deviation of spinal movement versus commanded speed (in m/s)



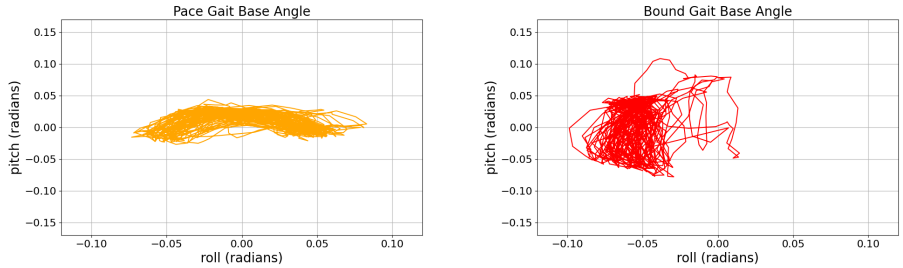
(b) Spinal joint and left front hip joint trajectories (in radians), commanded speed=1.2 m/s

Figure 5.15 Learned spinal movement in bound gait: spine joint trajectory in roll, pitch and yaw direction are represented by blue, green and red lines separately. The purple lines are left front hip joint trajectory as a reference. Each trajectories are obtained by calculating the average results of ten runs, the shaded area and bar are the standard deviation



(a) Walk, 0.8 m/s

(b) Trot, 1.2 m/s



(c) Pace, 1.0 m/s

(d) Bound, 1.2 m/s

Figure 5.16 Middle body segment angles in roll and pitch directions for basic gaits: all trajectories obtained by averaging the results of ten runs. Each run is performed for 1000 iterations.

on walking and pace gaits. The spinal movement trajectories in the roll and pitch directions for these gaits at optimal speeds are shown in Figure 5.16. It can be seen that the spinal movement range of the trot gait is close to that of the previous experiment (Figure 5.6), and its range of movement in the roll and pitch directions is approximately ± 0.05. However, the learned spinal movement in the trot gait allows TQbot move a greater distance than in the previous experiment. The characteristics of the spine in each gait are summarised as follows:

Walk gait

- Stability: introducing spinal movements makes the walk gait highly unstable, causing significant body shaking.
- Spinal role: the spine primarily adjusts the robot's posture to enable effective movement, as the walk gait without spinal action struggles to produce forward motion.

- Spinal movement: the offsets in roll and yaw directions are minimal (0.025 radians and 0.04 radians, respectively), with a standard deviation of about 0.025 in all three directions.

Trot gait

- Speed and stability: beyond a command speed of 1.2 m/s, there is no significant improvement in speed or stability; performance drops sharply at 1.5 m/s.
- Spinal role: movements in the roll and yaw directions are crucial, with amplitudes increasing as speed increases.
- Pitch direction: incorporating spinal movement in pitch direction via reinforcement learning enhances stability without compromising speed, suggesting a stabilising role for pitch direction spinal movements.

Pace gait

- Speed limitation: speed does not increase after reaching a commanded speed of 1.0 m/s. However, the spine also has a limited contribution to pace gait.
- Spinal role: the spine can adjust posture, with increased offsets in pitch (0.1 radians) and yaw (0.05 radians) directions. The amplitude changes are minor, especially in the pitch direction.

Bound gait

- Stability issues: becomes unstable and prone to falling when commanded speed exceeds 1.2 m/s.
- Spinal role: the spine is essential for movement, especially in the pitch direction, which shows the most significant activity. The roll direction movement can replace the legs to assist in adjusting the posture.

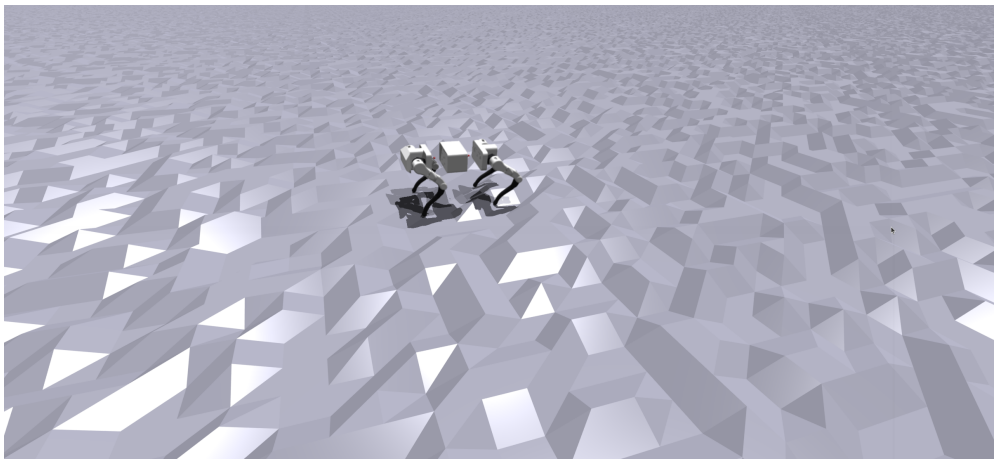


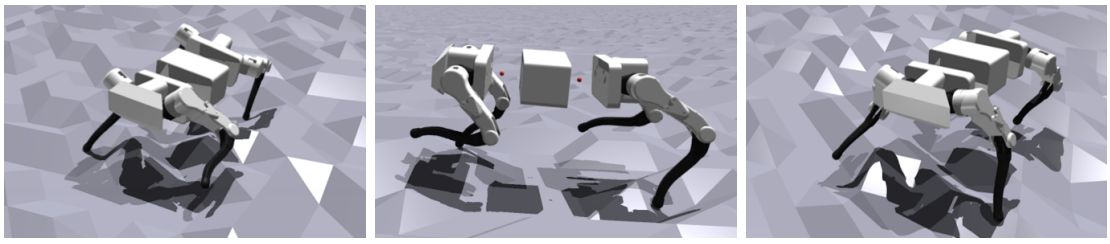
Figure 5.17 Uneven Terrain: Random rough terrain with variations of 0.1m

- Spinal movement: minimal movement in the yaw direction; larger amplitude in the pitch direction between command speeds of 1.0 to 1.2 m/s.

However, regardless of the gait, it can be seen that the movement of the spine has a certain improvement, thus it is reasonable to hypothesise that the spine should play a more important role on uneven surfaces.

5.3.4 Learning Sensory Feedback for Trot Gait

From the previous two experiments, it can be preliminarily concluded that the bio-inspired tensegrity spine provides benefit to some extent to the basic gaits, with the impact being particularly significant on the trot and bound gaits. However, the modified CPG can generate rhythmic locomotion for TQbot but was unable to enable it to move on uneven terrain. In this section, the novel CPG model with proprioception feedback proposed in Section 4.6 will be used. It allows the robot to generate dynamic trot gaits with spinal movements.



(a) All Feedback (b) Amplitude Feedback (c) Offset Feedback

Figure 5.18 TQbot screenshots for learned different proprioception feedback: TQbot walking on uneven terrain

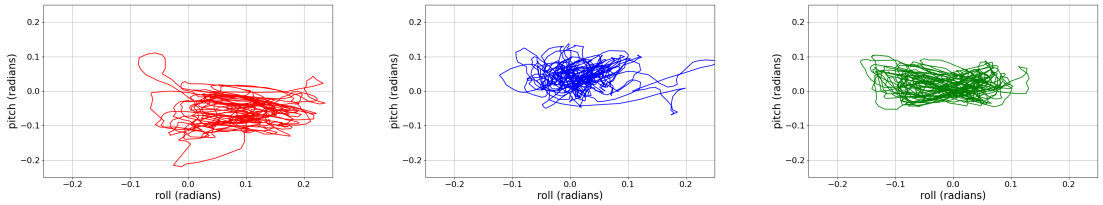
Uneven Terrain

In this experiment, 4096 simulated TQbot copies were placed in a 288*288m field, which consisted of 24 12*12m small fields. Each small field was composed of 48*48 blocks of different heights. The height difference of these small blocks was randomly generated, ranging from plus or minus 5 cm, and the terrain they formed was shown in Figure 5.17. At the beginning of training, a total of 4096 TQbots were evenly placed in the field with a certain spacing, but no collision volume between them. Figure 5.18 are screenshots of the results of learning different sensory feedback.

Feedback Performance

In this experiment, reinforcement learning was used to train and explore the effects of three different feedback on gait: (1) CPG only integrates the amplitude feedback term; (2) CPG only integrates the offset term; (3) CPG integrates the amplitude, offset and phase terms. When using CPG with default parameters and no sensory feedback, the generated trot gait cannot make the robot walk forward. The main reason is that TQbot with default parameters cannot lift its legs to cross uneven ground. In this case, phase feedback alone cannot change posture or lift legs, so it does not learn phase feedback alone.

In the three cases of integrating different sensory feedback, TQbot can generate dynamic and effective trot gait. Its performance in moving speed is better than all the results



(a) All feedback (b) Amplitude feedback (c) Offset feedback

Figure 5.19 Middle body segment movement angle in roll and pitch directions: All trajectories obtained by averaging the results of ten runs. Each run is performed for 1000 iterations.

of the first two experiments, and it can quickly adjust its posture to maintain balance during walking. The environment configuration of this experiment is the same as that of the first two experiments, and the time of travel distance is 20s. When learning offset and amplitude feedback, the number of training iterations are 1500, and the actual training time is about 30-40 minutes. When training all feedback, the learning rate is reduced to 0.00005 and the number of iterations is 3000, and the actual training time is about 1.5 hours. As shown in Figure 5.20, the gait performance of integrating different sensory feedback is measured by the stability value and walking distance respectively, and the data of each case is obtained from the results of ten runs. It can be clearly seen that whether in terms of stability or walking distance, it performs best when only learning offset feedback. It also can be seen in Figure 5.19c, the range of spinal motion is minimal after learning offset feedback. Although it can form an effective gait when only learning amplitude, the walking distance is the shortest. When learning all sensory feedback, although it can reach a walking distance close to that of learning only offset, the degree of body shaking is very high. Although the stability value is higher in the all feedback case, indicating that the body is shaking more, the distance it walks on uneven terrain is also very long, which means that it has adjusted its posture many times in the process to avoid falling and regain its speed.

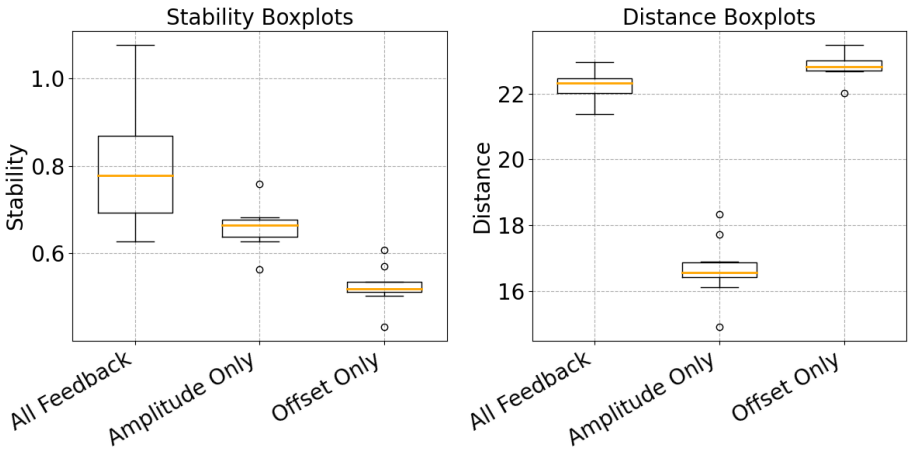


Figure 5.20 The performance of learned different sensory feedback: All results are based on ten runs

Offest Feedback

Figure 5.21 shows the motion trajectories of the robot's joints on uneven terrain after learning only offset feedback. From the trajectory of the hip joints in Figure 5.21, it can be seen that the phase difference between the legs has not changed, but the hip joints of the two front legs have an offset that makes them form a more stable posture. In addition, the shoulder joints on the left and right sides also have an offset that makes them stretch to both sides to obtain a more stable posture, that is, the coverage area of the robot's feet is larger. At the same time, there are some offsets in the movement of the spine in the roll and yaw directions to adjust the posture.

However, it can be seen from the waveforms of each joint that the feedback to the offset not only modifies the offset of each joint, but also modifies its amplitude to a certain extent. This is because the offset feedback can also be seen as another output superimposed on the output waveform of the CPG. In theory, reinforcement learning can change the phase and amplitude of the CPG output waveform by outputting a series of continuous actions based on proprioception, but the mechanism is too complicated for practical implementation on TQbot at this stage.

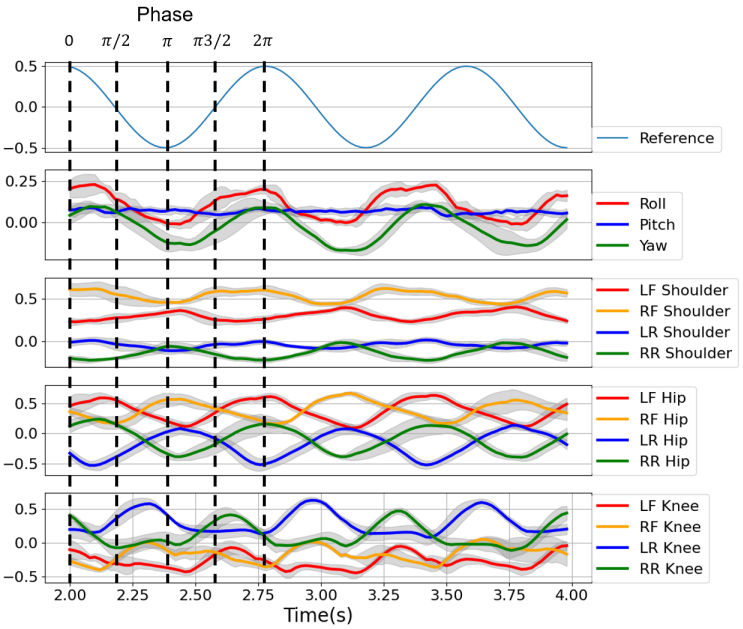


Figure 5.21 Learned joint trajectories for offset feedback and a reference oscillator trajectory between 2 and 4 seconds. Left y-axis is the joint radians, upper x-axis is the phase and lower x-axis is the time in seconds. Each line is the average results of ten runs. The grey areas are standard deviations

Amplitude Feedback

Figure 5.22 shows the output angle trajectory of each joint output by CPG after learning only amplitude feedback. It can be clearly seen that the amplitude of the spinal motion in the yaw direction increases to nearly 0.2 radians (about 10 degrees). The amplitude in the roll and pitch directions is smaller. The phase difference between the other joints has not changed and the gait can be adjusted according to the terrain to stabilise walking. From the results, the robot can generate a dynamic gait by adjusting the amplitude alone when walking on uneven terrain, but the speed improvement is limited. Moreover, unlike adding offset feedback to adjust the robot's posture to make it walk in a more stable posture, the gait generated by the CPG waveform with only amplitude feedback added needs to constantly adjust the amplitude to keep it balanced. The large area of grey in Figure 5.22 shows that the amplitude feedback value is not concentrated, which just illustrates this point. In particular, the standard deviation of the spinal motion in the pitch direction is larger, which means that the spinal motion

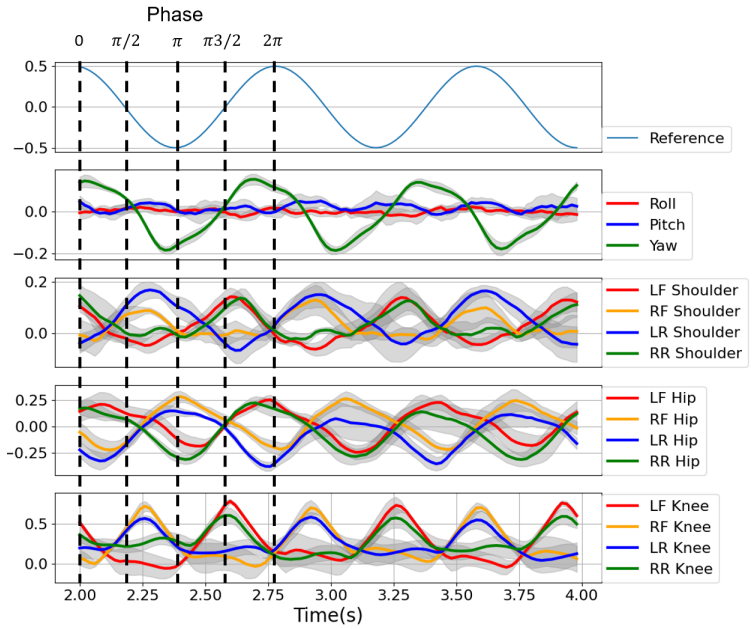


Figure 5.22 Learned joint trajectories for amplitude feedback and a reference oscillator trajectory between 2 and 4 seconds. Left y-axis is the joint radians, upper x-axis is the phase and lower x-axis is the time in seconds. Each line is the average results of ten runs. The grey areas are standard deviations

in the pitch direction may play a greater role in adjusting balance than the spinal motion in other directions. This also confirms the speculation of learning spinal motion in the previous experiment.

Amplitude, Offest and Phase Feedback

When learning the three feedbacks, phase, amplitude, and offset, the action space will be three times larger than before and the state space will be 30 parameters larger than before. Training these three feedbacks simultaneously is more complex and difficult than training any of them individually. Figure 5.23 shows the actual output trajectory of the CPG model with all sensory feedback. As can be seen Figure 5.23, compared to the hip and knee joints, the main function of the shoulder joint is to change posture. The posture learned is similar to the shoulder joint posture learned when only offset is learned, and both front legs stretch outward to maintain a stable posture. When phase feedback is integrated, it is possible to change the phase between the different

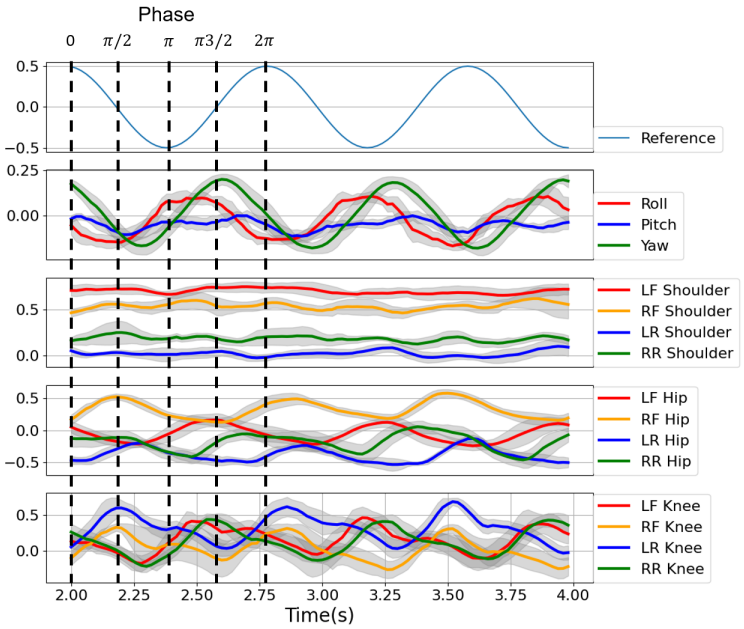


Figure 5.23 Learned joint trajectories for all proprioception feedback and a reference oscillator trajectory between 2 and 4 seconds. Left y-axis is the joint radians, upper x-axis is the phase and lower x-axis is the time in seconds. Each line is the average results of ten runs. The grey areas are standard deviations

oscillators. An interesting observation is that the phase in the roll direction becomes different from the default value of $\frac{3\pi}{2}$, and it can be seen that its actual value is between 5.11 and 5.5. This result is consistent with the optimal phase obtained in the first experiment. However, the difference is that the amplitude of the spine in the yaw direction reaches about 0.2 radians, which is much larger than the amplitudes in section 5.3.2. At the same time, it can be seen that the phase between the legs remains trot gait.

5.4 Physical Results

This section demonstrates the physical experimental results conducted in two spine configurations. A total of six experiments were conducted for one spine to validate the effectiveness of the control method and the performance of the spine. A PD controller

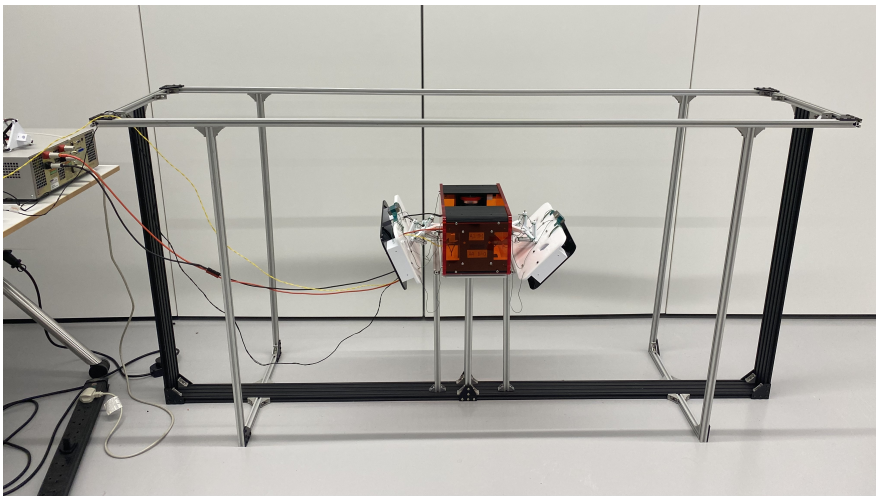


Figure 5.24 Gantry and bio-inspired spine: Left is the bench power supply and control PC, the first configuration spine is placed on a panel

is used to control the wire length to allow the spine to rotate in roll, pitch and yaw directions within ±20 degrees.

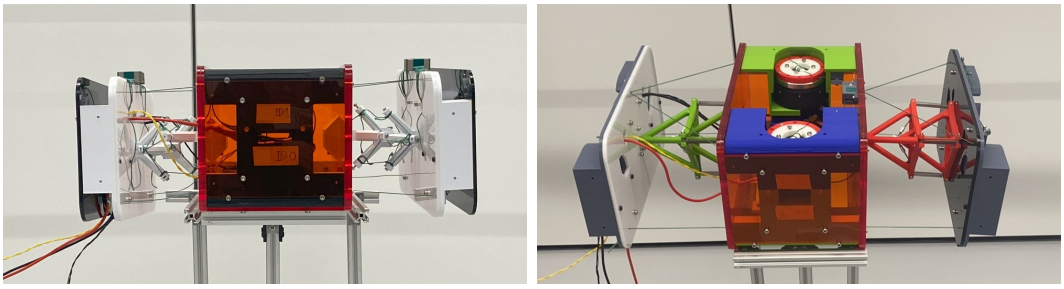
5.4.1 Experiment Setup

A gantry with a size of $2m \times 0.5m \times 1m$ (length × width × height) was built for testing the tensegrity spine, as shown in Figure 5.24. The middle body segment is placed on a platform, with front and rear parts hanging off the two ends. Two IMUs are mounted in the top centre of the front and rear segments respectively, in the same direction. A jetson orin nano is placed in the middle body to collect data from two IMUs and control the spine. All actuators and the onboard PC are powered by a bench power supply that is placed on the left side of the gantry but not shown in the picture. The phase of spine control and IMU data collection is 300hz, and the speed of spine movement was set to a low range for safety. The IMU coordinates are used to measure the angles of the front and back body parts are consistent and the same as those modelled by the control method as coordinate Σ_0 shown in Figure 3.33.

The bionic spine is in a state of relaxation at start-up hence a pretension process is needed. Before the start of each experiment, the spine was manually tensioned. During the control spine, the pretension ratio was set to 0.01 for all experiments.

5.4.2 Experiments

Each spine was automatically tensioned before the start of the experiments, as shown in Figure 5.25.

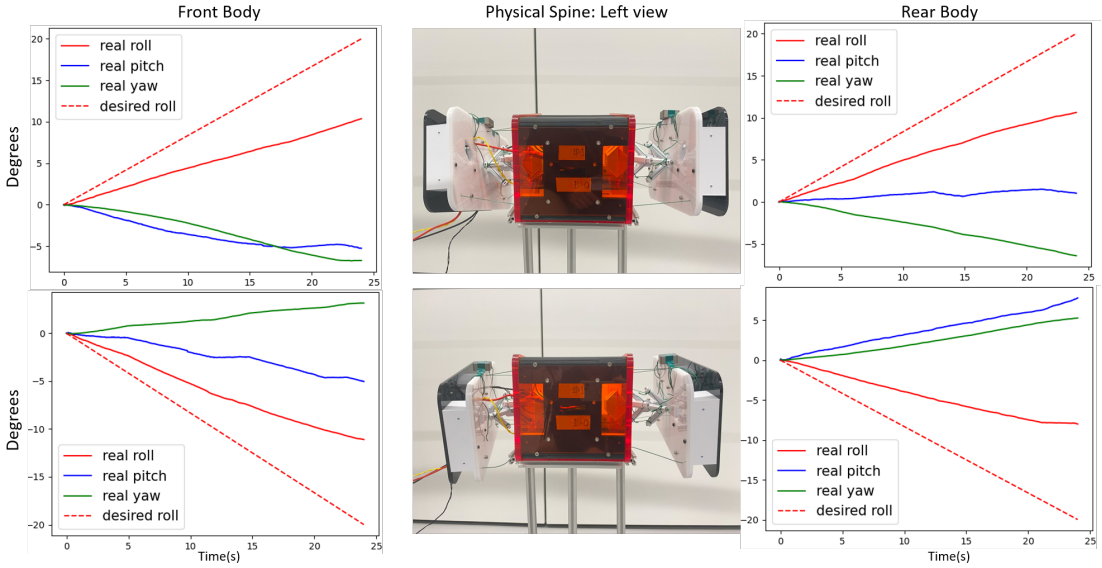


(a) Spine Configuration 1 (b) Spine Configuration 2

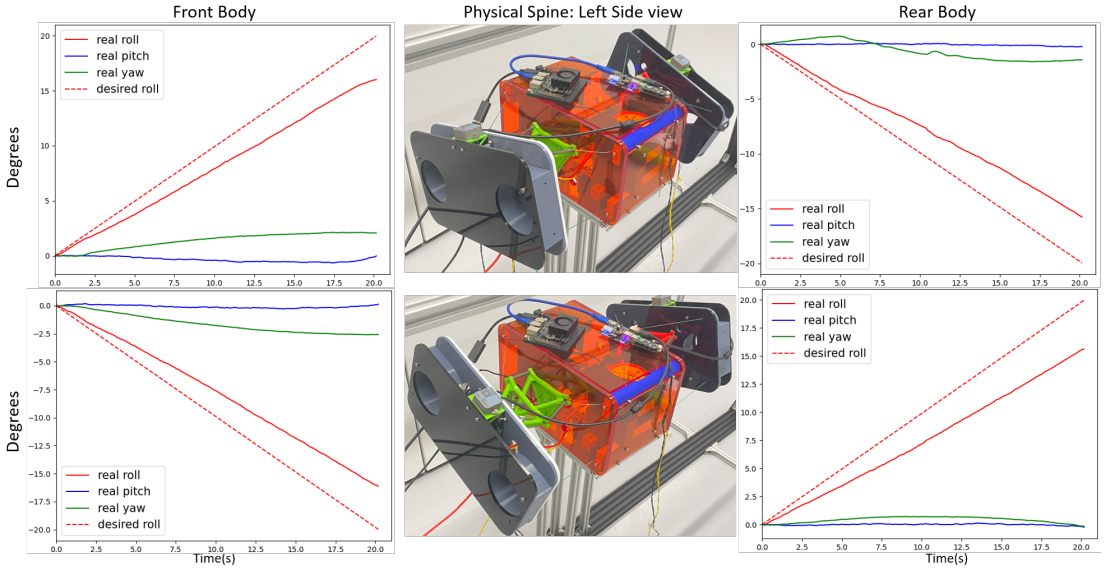
Figure 5.25 Left side view of the two configurations bionic spines, which are tensioned and placed on a panel in the centre of the gantry.

Roll Motion

Spine Configuration 1 The roll motion experiments are rotating the spine to ±20 degrees in the transverse plane and the results are shown in Figure 5.26. The spinal movement in these experiments lifts the left shoulder at about half the desired angle due to the restrictions of the passive spine. It also leads to the front and back body parts being slightly deflected to the right and lifted upwards simultaneously, which means the spine bends and rotates in both the pitch and yaw directions. Although these appear to be inaccurate results, the motion closely resembles the natural twisting movement of the waist seen in quadrupedal animals as they walk and demonstrates the structure's bionic nature.



(a) Tensegrity Spine 1



(b) Tensegrity Spine 2

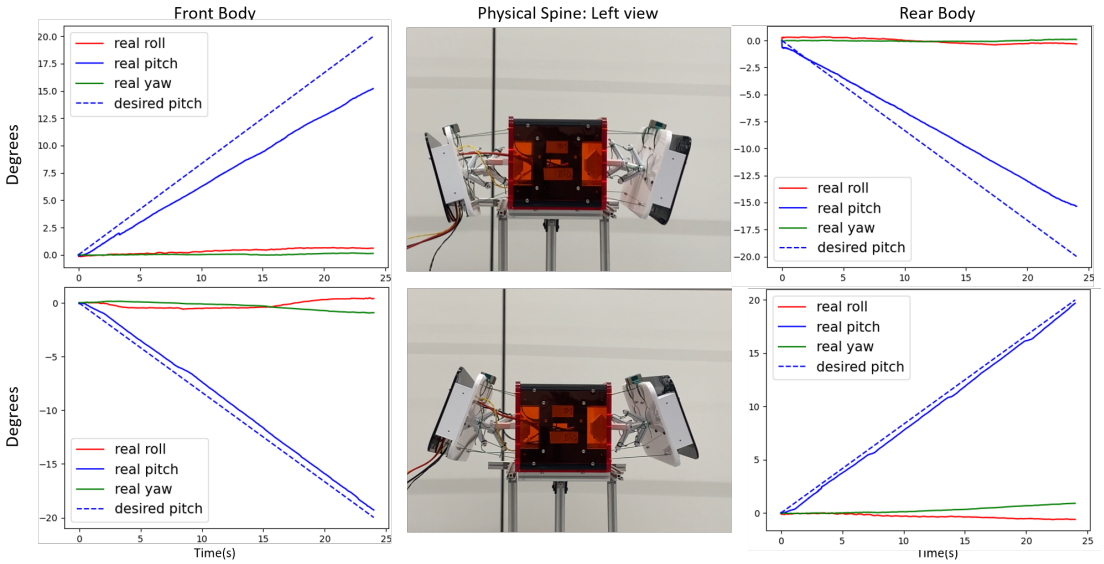
Figure 5.26 The experimental results in which the spine was rotated $+/-20$ degrees in the transverse plane in the first and second rows respectively; the left and right columns correspond to the results of the left and right body parts. The middle column shows snapshots of the end of the spinal rotation; the front of the body part on the left.

Spine Configuration 2 The motion generated by the second configuration of bio-inspired in the roll direction differs from that of the first configuration, which the rear body segment rotating in the opposite direction to the front body segment. As evident in Figure 5.26a, the second spine has more accurate spinal movements in the roll direction. This improvement is due to passive spine components, including springs with an elasticity higher than that of the fishing line used in the first configuration. As a result, the passive spine imposes fewer restrictions on movement in the pitch and yaw directions during motion.

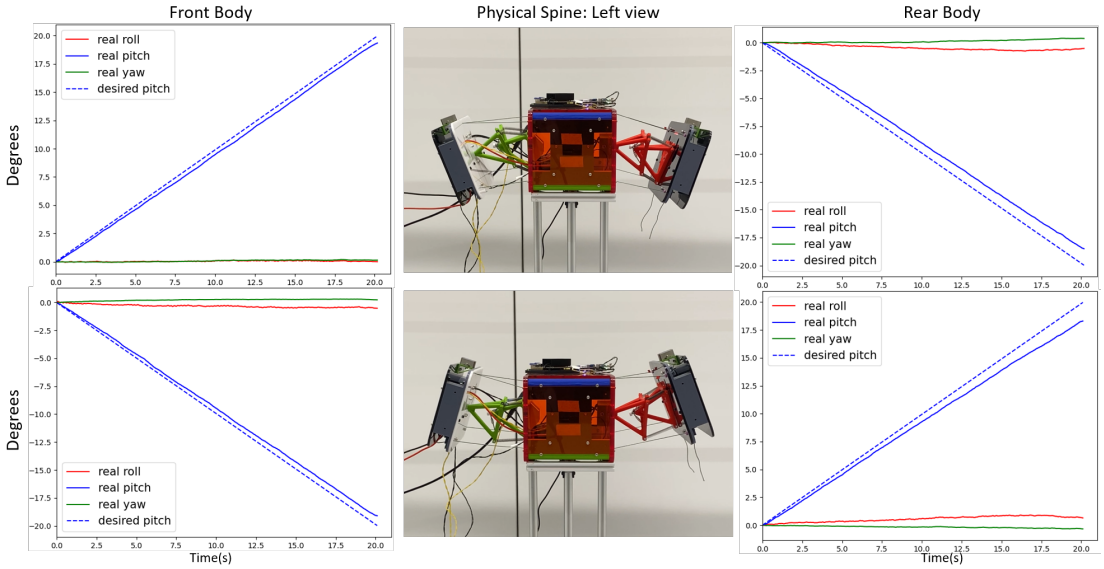
Pitch Motion

Spine Configuration 1 According to the results from experiments in the pitch direction, the passive spine imposes much less restriction on the movement of the spine in the pitch direction than in the roll direction. Figure 5.27 illustrates the spine bending motion in the sagittal plane. The upward bending motion of the spine is as expected but has an error of about 5 degrees in the downward bending motion. The main reason for this behaviour is that gravity gain was not added to the control process. It leads to the spine not being fully pretensioned when the front and rear segments are rotating in the pitch direction. In more detail, there is not enough tension on the two top wires (p_1, p_2) that causes the bottom wires (p_3, p_4) to be a little loose. Thus, the bottom wires are not tightened to the desired length.

Spine Configuration 2 Similar to the results in the roll direction, the spine of the second configuration also has higher accuracy in the pitch direction than the first configuration. However, the motion of the rear spinal joints had higher errors caused by manual assembly, which results in the front and rear lengths of the active fishing line were inconsistent. In addition, the passive spine has almost no restrictions in the roll and yaw directions during pitch movements, as there is minimal error in these two directions.



(a) Tensegrity Spine 1



(b) Tensegrity Spine 2

Figure 5.27 The experimental results in which the spine was rotated $+/-20$ degrees in the sagittal plane in the first and second rows respectively; the left and right columns correspond to the results of the left and right body parts. The middle column shows snapshots of the end of the spinal rotation; the front of the body part on the left.

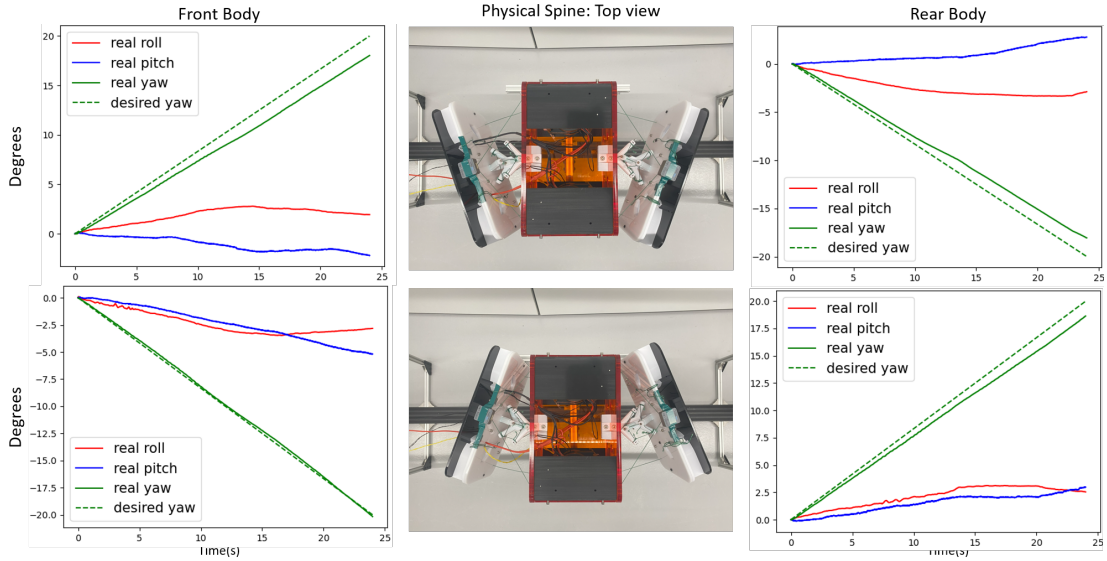
Yaw Motion

Spine Configuration 1 The lateral flexion of the spine in the yaw direction is commonly employed to improve turning efficiency for robots. As depicted in Figure 5.28a, the bending error of the spine in the coronal plane does not exceed 3 degrees. Nonetheless, a slight error in the roll direction persists which is attributed to the limitation of the passive spine. In addition, since there is no gravity gain in the control, the spine will lift slightly upward during the movement, resulting in the spine being slightly deflected in the pitch direction.

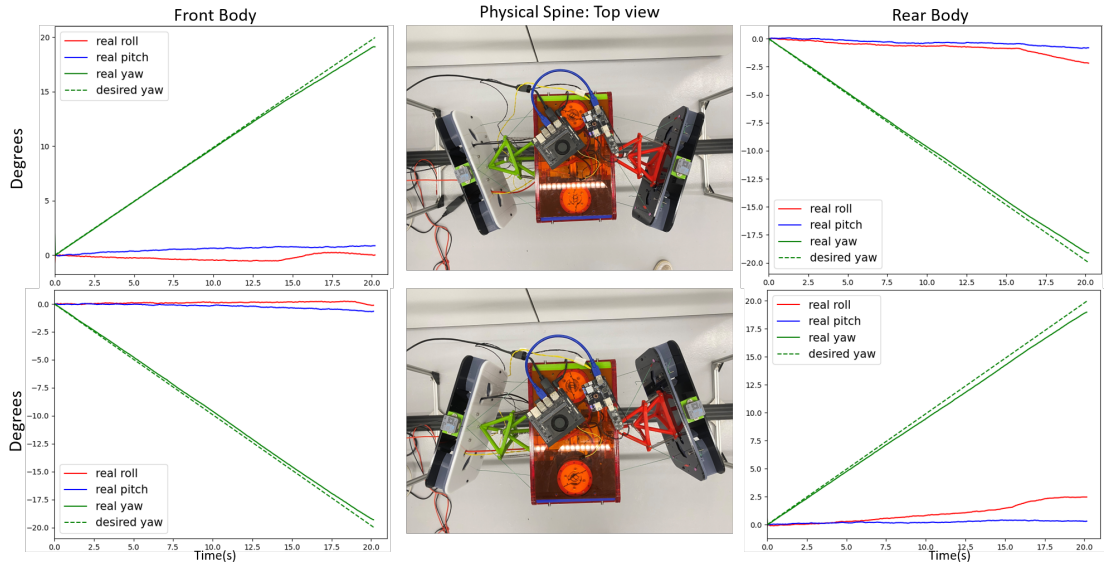
Spine Configuration 2 The second configuration of the spine has a much higher accuracy in the yaw direction. As shown in Figure 5.28b, the error in the yaw direction is less than 1 degree, and the motion limitations of the passive spine in the roll direction and the pitch direction after adding the spring are smaller than those of the passive spine without the spring.

5.4.3 Restriction on Passive Spines

Based on the physical experimental results, it can be concluded that the first spine configuration, which lacks springs, imposes restrictions on spinal movement when it is fully tensioned, especially in the roll direction where the limitation is most pronounced. In contrast, the springs in the second configuration allow less restriction during spinal movement, as the springs can stretch more than the fishing line. It is important to note that most robots use tensegrity structures without being fully tensioned. Some of them directly use the cables in the tensegrity structure to control robot movement, which compromises the bioinspired characteristics to some extent.



(a) Tensegrity Spine 1



(b) Tensegrity Spine 2

Figure 5.28 The experimental results in which the spine was rotated $+/-20$ degrees in the coronal plane in the first and second rows respectively; the left and right columns correspond to the results of the left and right body parts. The middle column shows snapshots of the end of the spinal rotation; the front of the body part on the left.

5.5 Summary

This chapter presents both the simulation and physical experiments conducted on the bio-inspired spine. The simulation experiments primarily explored the role and impact of the spine on four basic gaits: walk, trot, pace and bound, with a focus on analysing the spinal movement and influence during the trot gait. In the first experiment, roll-yaw phase parameters that were better than manually tuned parameters were identified, leading to an improvement in travel distance. In the second experiment, dynamic spinal movements were generated using reinforcement learning, enabling the robot to adjust the speed of different gaits based on commands. Although spinal movements improved all four gaits to different levels, it contributes most to the trot and bound gaits. Additionally, in this experiment, the spine not only contributed to increasing gait speed but also played a key role in maintaining stable. Notably, for the trot gait, the command speed with optimal performance revealed that the learned spinal phase was similar to the optimal phase found in the first experiment.

However, in the first two experiments, the leg movements were rhythmic and could not enable the robot to move on unstructured terrain. Therefore, in the third experiment, a CPG integrated with proprioceptive feedback was used to generate a dynamic trot gait. In this experiment, the robot learned to adjust feedback parameters based on proprioception, resulting in the generation of an efficient and stable gait. The role of the spine in dynamic gaits became even more evident, as it not only enhanced speed but also adjusted posture through pitch movements. In summary, the simulation experiments not only validated the effectiveness of the spine structure but also provided insights into designing optimal spinal movements.

The physical experiments validated the effectiveness of the control method mentioned in Chapter 3 and analysed the bio-inspired characteristics of the two spines. The experimental results showed that the second spine configuration had higher control accuracy due to the reduced restriction from its passive structure, providing valuable guidance for future tensegrity structure designs.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

With the development of robotics in recent years, quadruped robots show great mobility and balance on flat ground. However, its performance does not match that of quadruped robots when walking on extreme terrains, such as sand and ice. To enable quadruped robots to cope with such extreme terrains, most researchers have focused on control and gait generation methods, as well as sensors. When compare it with those fields, only a few devote their efforts to spinal structure, although it plays a crucial role in improving the mobility of quadruped robots. Therefore, this work aims to develop a bionic spine to further enhance the mobility of legged robots. To achieve this, a literature review in Chapter 2 given a comprehensive investigation by exploring exist spine mechanism structures and coordinated leg-spine gait generation methods. Chapter 3 demonstrates the mechanism structure of two bionic spines and the whole quadruped robot platform. Furthermore, a gait generation and control method applicable to the novel quadruped robot is designed and demonstrated in Chapter 4. Chapter 5 evaluates the gait generation method in simulation and validates the novel spines through experiments. This Chapter will summarise the highlights of all the previous chapters. In addition, the three core questions posed in Chapter 1 are answered in detail according to the entire research.

6.1.1 Summary of Chapters

In the literature review presented in Chapter 2, the thesis first investigates quadruped robots with spinal structures developed to date. According to their spine, these robots are divided into actively controlled spines and passively controlled spines. The focus then shifts to exploring different types of active spines, which are further divided into high-stiffness and low-stiffness spines. The review examines these spines from the perspectives of structure, driven methods, and their advantages and disadvantages. The review concludes that high-stiffness, actively driven spines tend to decrease energy

efficiency and self-stability, while low-stiffness, passive spines improve both aspects. Additionally, it is suggested that a spine with two joints may be most beneficial for improving the speed of quadruped robots. Another important finding is that the most quadruped robots can only actuate one degree of freedom in their spines at a time during spine movement, which limits the role of the spine in quadruped robots.

The investigation of the second question requires consideration of the spinal structure designed in this study. The TQbot designed in this work has a total of 15 degrees of freedom, and its bio-inspired spine exhibits characteristics such as low stiffness, elasticity, and cable-driven mechanisms, which are challenging to model. As a result, a model-free method, Central Pattern Generator (CPG), was chosen as the gait generation method. CPG is a biologically inspired neural network, offers a natural advantage in coordinating the movements of different joints in the robot, without the need for complex modeling. Consequently, the literature review also explores various CPG models. In exploring the second question, an assumption is that the spine should provide benefits for quadruped robots on uneven terrain. Therefore, generating dynamic gaits using a CPG is also a necessary task. However, the investigation reveals that current CPG model struggle to generate dynamic gaits due to the lack of sensory feedback. The challenge in mathematical modeling sensory feedback because of the mechanism by which sensory feedback affects dynamic gait shaping is not yet fully understood from a biological perspective. As a result, reinforcement learning is considered as a potential solution, as it enable an agent learning from reward functions. Consequently, a brief investigation into the application of reinforcement learning in robotic learning was also conducted from this study.

Chapter 3 provides a detailed explanation of the hardware design and control methods of the TQbot. To ensure an efficient design process, the concepts of the robot design were established. This is followed by a description of the mechanical structure, with particular emphasis on the bio-inspired spine, which is divided into active and passive components. In this work, two spine configurations were designed and developed. Both have differences between their active and passive parts. The primary difference

in the active part lies in the rotation of the two spinal joints in the roll direction. The first configuration spine enables the front and rear body segments to rotate in the same roll direction, while the second rotates in the opposite direction in the roll direction. The passive structure employs two different tensegrity structures, with the second incorporating springs. In addition, the chapter details the mechanical design of the robot's body, legs, and feet. To facilitate rapid and cost-effective prototyping, electronic hardware, battery units, and battery management systems (BMS) were procured, and battery modules were assembled. Additionally, a kinematic model for TQbot was established, and the control method for the spine was explained to support the subsequent hardware testing and experiments.

Chapter 4 introduces a phase oscillator based CPG that generates various gaits for TQbot, which can be integrated with sensory feedback. The chapter presents the definition of phase oscillator and proposes an internal sensory feedback mechanism to independently control the stance and swing phases of the gait. Following this, the topology of the CPG model is designed based on the degrees of freedom of TQbot, with the optimal connections determined using the order parameter. The concept of a reference oscillator is also introduced to simplify subsequent phase parameter settings. Additionally, various gaits are designed for TQbot, including four fundamental gaits as well as backward and sideways gaits. Finally, a proprioceptive feedback integrated CPG model based on reinforcement learning is proposed, and the applied reinforcement learning algorithm, PPO, is introduced.

In order to answer the third question, three simulated experiments and a method for quantifying gait performance are designed and described in Chapter 5.

In the first experiment, spinal movement in the roll and yaw directions is evaluated in trot gait by traversing 256 phase combinations. The experiment results show that the roll-yaw phase, $(\frac{3\pi}{2}, \frac{3\pi}{2})$ performed better than the gait that was manually tuned in Chapter 4. However, in a rhythmic trot gait, the spinal movement increases the robot's speed while decreasing its stability. The second experiment uses reinforcement

learning to generate spinal movement to cooperate with the locomotion of the legs. Spinal movement was found to increase the speed of trot and bound gaits, but its main effect on pace and walk gait is to adjust posture. The third experiment not only validated that the proposed CPG model can utlise proprioceptive feedback to generate a dynamic trot gait, but also its results are better than those of the trot gait in the first two experiments. However, it was also found that in the learned dynamic trot gait, the roll-yaw phase of the spinal movement also is around $(\frac{3\pi}{2}, \frac{3\pi}{2})$.

6.1.2 Bio-inspired Spine

Question 1: What kind of spine mechanical structure can maintain lightness and compliance while simultaneously providing multiple DOFs?

In this work, two tensegrity-based spines were designed and developed for a quadruped robot. The spines are composed of rigid bars and cables to minimise weight and maintain compliance while providing multiple DOFs. Both integrate active parts for controlled movement and passive elements to absorb shocks and enhance compliance. With three DOFs in roll, pitch and yaw directions, the spine allows for generating agile and dynamic movements, enabling robots to navigate uneven terrains and maintain stability effectively. Most importantly, the tensegrity structure allows the spine to move in all three DOFs simultaneously. Using lightweight PLA and fishing lines further reduces the overall mass, while the modular design enhances adaptability and scalability to different robot configurations. The combination of lightweight materials, tensegrity, and active-passive parts enables the spine to meet compliance, adaptability, and multi-DOF motion requirements6. However, some engineering issues of the tensegrity structure were identified during the experiments and assembly process, and the second spine improves these issues.

For the first spine configuration, when the passive spine is fully tensioned, the spinal movement angle in a direction will be different from the expected angle due to the restriction of the tensioned fishing lines in other directions. In particular, the movement

error in the roll direction is the largest. In addition, the tensioned passive spine becomes loose after each movement since the elasticity of the fishing line is small. It makes the passive spine lose its compliant state, which might play an important role in robot stability. Resulting in the need to replace the passive spine or reassemble it to return to fully tensioned state after every movement. To solve this problem, the second passive spine with springs was designed. The experiments in Chapter 5 show that the second spine configuration is less restricted and can produce more precise movements. Moreover, because of its lower stiffness and certain elasticity, it is speculated that its stability performance in reality may be better than that suggested in the simulation.

6.1.3 Gait Generation Method

Question 2: Which method can parameterise the spine movement and coordinate it with the movement of the legs and generate various gaits by a few parameters? Currently, most quadruped robot spines are developed for different functions that require special structural designs. It also lead to each spine needing a specific method to control it. On the other hand, the spine joints add more DOFs to the robot resulting in complex control of it. Therefore, the CPG-based control framework were proposed for TQbot. This approach leverages the following features.

The CPG uses phase oscillators to generate each DOF trajectory, which are controlled by a set of variables, such as frequency, amplitude and phase lag. It parameterises the spine and leg movements to efficiently control complex multi-joint systems. Secondly, the CPG network is composed of coupled oscillators that can synchronise spine and leg movements. Adjusting phase lag between these oscillators can automatically generate coordinated leg-spine gaits. In addition, proprioceptive feedback from the robot's sensors allows for real-time adaptation to external disturbances or uneven terrain, enhancing stability and responsiveness. By utilising the CPG-based control method, the spine movement can be effectively parameterised and coordinated with leg movements, enabling the robot to adapt to different terrains and tasks with minimal adjustments.

6.1.4 The Role of the Spine

Question 3: What phase difference, amplitude and frequency of spinal movements can improve the velocity and stability of gaits on various terrains?

In the literature review presented in Chapter 2, several spinal structures of quadruped robots were investigated. However, most of these spines possess one or two degrees of rotational freedom, while some spines with three degrees of freedom are unable to rotate simultaneously in multiple directions. It reduces the potential benefits that the spine could bring to improving gait. As a result, current exploration of leg-spine coordinated gaits focuses primarily on the impact of a single degree of freedom movement on gait performance. In this work, the three experiments in Chapter 5 progressively analyse the role of spinal movement, beginning with rhythmic spinal and leg movements, followed by dynamic spinal movements with rhythmic leg movements, and finally, fully dynamic gaits. The results demonstrate that spinal movement in this structure not only has the potential to enhance speed but also helps maintain body posture stability. A significant and probably generic conclusion drawn from these experiments is that spinal movement in the pitch direction appears to play a significant role in regulating the stability of leg-spine coordinated trot gaits. Overall, the thesis designs and develops a novel bionic spine for quadruped robots and proposes a CPG-based method to generate leg-spine coordinated gaits; also, the simulation work explored the role of this spine in various gaits in a quadruped robot. The results show that the bionic spine can improve the ability of robot movement on uneven ground and that spinal movement patterns that have beneficial effects on different gaits have been discovered.

6.2 Future Work

This thesis presents a novel bio-inspired spine and validates its effectiveness through simulation. While the prototype design and initial spine testing have been completed. In the future, it is necessary to complete the testing of the complete TQbot and conduct

experiments to explore the potential of this bionic spine. Based on the literature review and the findings presented, future research directions for this thesis can be broadly categorised as follows.

6.2.1 Passive Spine Assembly Device

The assembly of wire-type robots has always been the biggest challenge during the building process. It is cumbersome and time-consuming, but may also cause errors and instability of the structure.

To achieve easy assembly and reduce the error generated by it, improving the design of the structure is necessary. In addition, an external device can be designed to assist in assembly. For example, a device with a linear rail and two rotatable ends which both ends can rotate and translate for assembling different sizes of parts. The parts to be assembled can be fixed at both ends. The linear rail has scales and fixtures to adjust the distance between the two ends. It ensures that each line has the same length after assembly. The rotation function at both ends is used to change the direction of the object and assemble other lines.

6.2.2 Bio-inspired Spine with Adjustable Stiffness

For bio-inspired robotic spines, the stiffness of the spine plays a crucial role in influencing gait performance. A high stiffness spine for quadruped robots can increase weight and energy consumption and can even lead to unstable movement on flat terrain. However, on uneven terrain, the compliance of a low stiffness spine can enhance gait stability and may deal with external perturbation. Additionally, an elastic spine could potentially conserve energy while a non-elastic spine cannot. Therefore, a spine that can adjust its stiffness and has elastic properties seems to be the optimal structure for quadruped robots. Fortunately, these features can be achieved in the bionic spine presented in this work.

The second spine structure in the thesis includes springs in its passive components. Due to the characteristics of tensegrity structures, the passive spine has the potential to store elastic potential energy in multiple directions and release it to save energy. Additionally, the stiffness of the wire-driven bionic spine can be easily controlled by adding elastic elements. For example, replacing fishing lines with other materials that offer elasticity or add springs to active fishing lines.

6.2.3 Learning Control Parameters of CPG

It can be noted that all control parameters of CPG are manually fine-tuned in this thesis. It requires prior knowledge and expert experience, which cannot provide effective parameters for unknown situations and may challenge finding the optimal parameters. For example, the roll-yaw phase parameters found in Section 5.3.2 are better than the manually adjusted parameters in terms of forward velocity and body stability.

From a biological perspective, higher brain centres, such as the motor cortex and cerebellum, send signals to the CPG to generate locomotion. In robotics, this process can be approximated using learning methods to adjust the CPG's control parameters, allowing the robot to adapt to various conditions and extreme terrains, such as ice surfaces and sand.

6.2.4 Bio-inspired Spine for Biped Robots

Compared to quadruped robots, bipedal robots face greater challenges in maintaining balance and gait stability because they have less ground support. It makes them more susceptible to external perturbation, such as uneven terrain or external forces. Introducing a bionic spine like that presented in this study can enhance the balance and stability of bipedal robots. A low stiffness bionic spine provides additional degrees of freedom and compliance, allowing the robot to adjust postures as it moves on different terrains and environments. The elastic parts can absorb the forces generated during

movement. At the same time, it can mimic the natural movement patterns of humans, making the robot's gait smoother and more natural. Overall, a soft bionic spine is crucial for bipedal robots. It not only enhances the robot's stability and balance but also makes its movements more natural and adaptable, enabling it to better cope with various challenges in the real world.

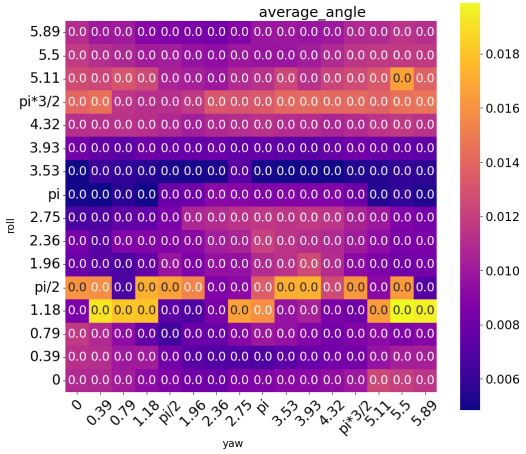
Appendix

In this appendix, supplementary data figures of experimental results in Chapter 5 are given.

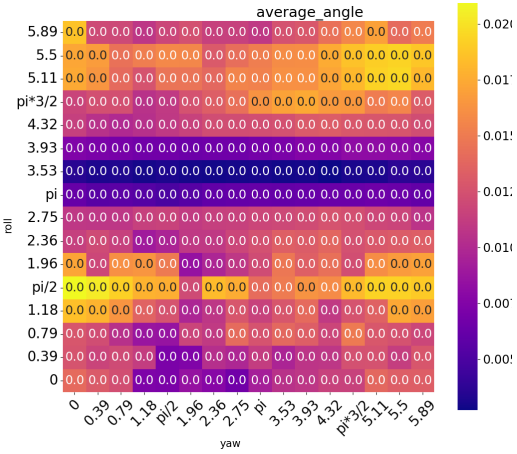
Figure 6.1 shown the data that are required to calculate the stability in Section 5.3.2, which are average angles, maximum angles in roll and pitch directions.

Figure 6.2 shows the performance of all gaits in section 5.3.3 after learning spinal movements.

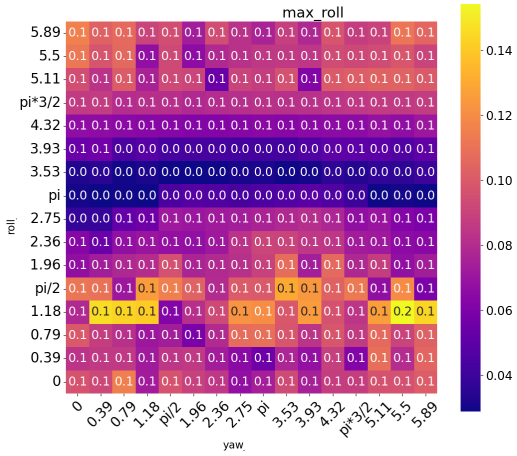
Figures 6.3, 6.4 and 6.5 demonstrate the complete waveform of the learning offset, amplitude, and all proprioception feedback in Section 5.3.4.



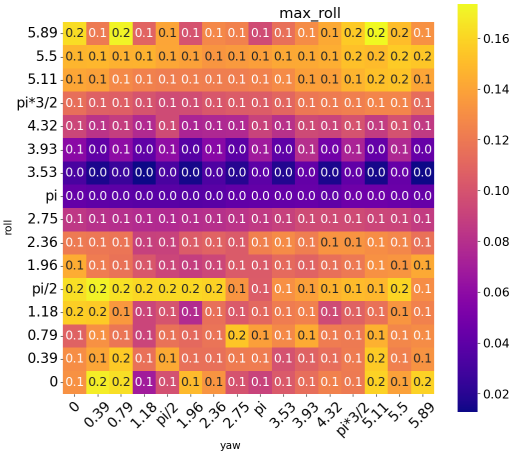
(a) Average angle, amplitude=2.5



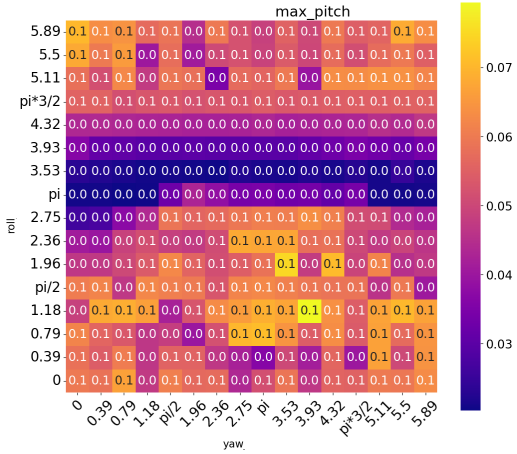
(b) Average angle, amplitude=5



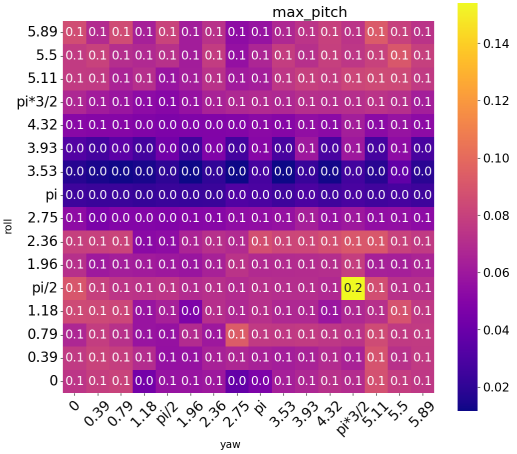
(c) Maximum angle in roll direction, amplitude=2.5



(d) Maximum angle in roll direction, amplitude=5



(e) Maximum angle in roll direction, amplitude=2.5



(f) Maximum angle in roll direction, amplitude=5

Figure 6.1 Average angles, maximum angles in roll and pitch directions: all angles in radians. The amplitude of the left column pictures is 2.5 and that of the right column pictures is 5.

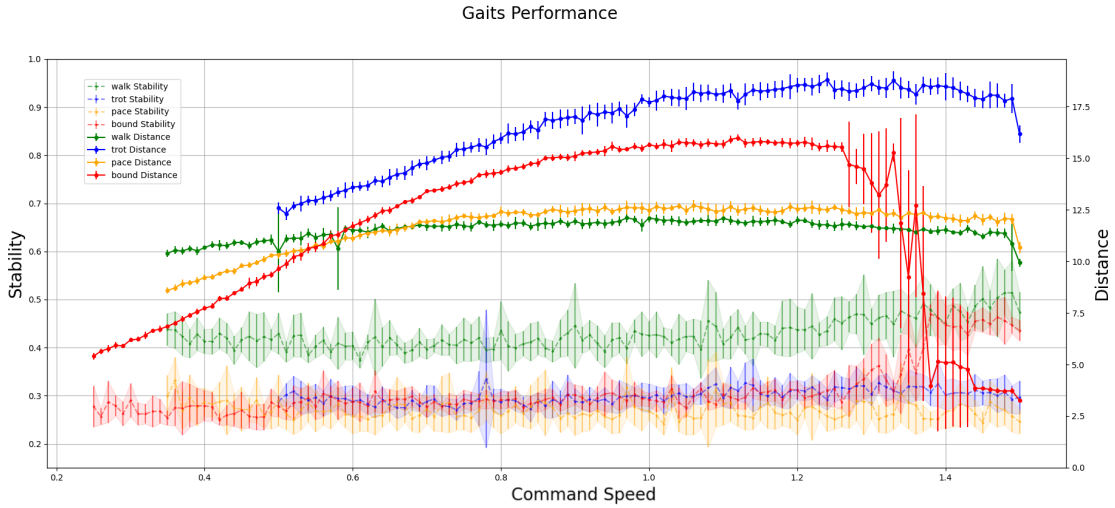


Figure 6.2 All gaits performance with learning spinal movement.

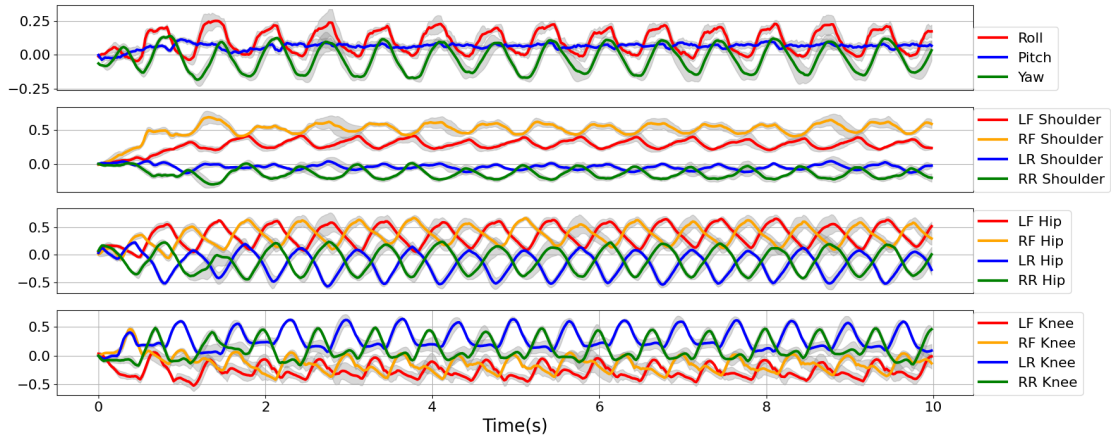


Figure 6.3 Learned joint trajectories for offset feedback and a reference oscillator trajectory within 10 seconds. Left y-axis is the joint radians, upper x-axis is the phase and lower x-axis is the time in seconds. Each line is the average results of ten runs. The grey areas are standard deviations

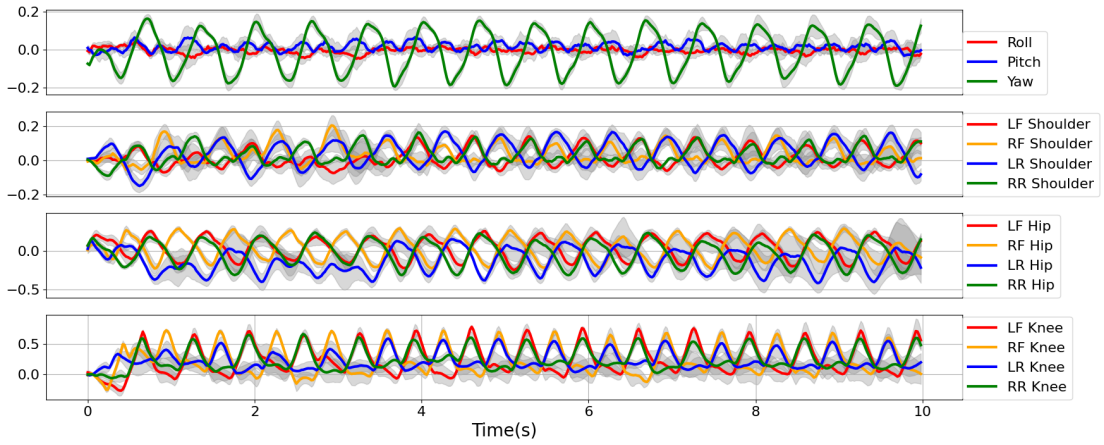


Figure 6.4 Learned joint trajectories for amplitdude feedback and a reference oscillator trajectory within 10 seconds. Left y-axis is the joint radians, upper x-axis is the phase and lower x-axis is the time in seconds. Each line is the average results of ten runs. The grey areas are standard deviations

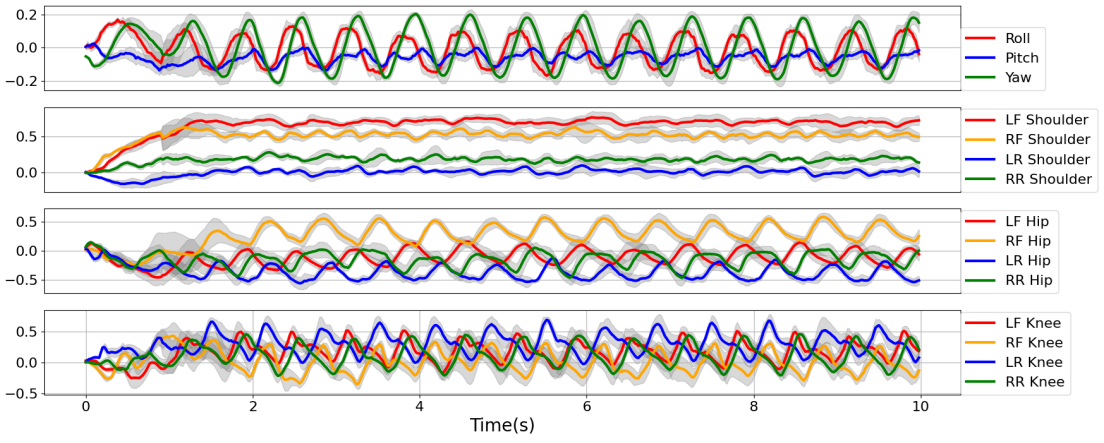


Figure 6.5 Learned joint trajectories for all proprioception feedback and a reference oscillator trajectory within 10 seconds. Left y-axis is the joint radians, upper x-axis is the phase and lower x-axis is the time in seconds. Each line is the average results of ten runs. The grey areas are standard deviations

Abbreviations

ASTSM Adaptive supper-twisting sliding mode

BFS Bidirectional flexible spine

BLDC Brushless Direct Current

BMS Battery management system

CNC computer numerical control

CPGs Central Pattern Generators

CPG Central Pattern Generator

CRPM Completely restrained parallel manipulator

DARPA Defense Advanced Research Projects Agency

DH Denavit–Hartenberg

DOFs Degrees of Freedoms

DOF Degrees of Freedom

EPFL Swiss Federal Institute of Technology Lausanne

ETH Swiss Federal Institute of Technology in Zurich

FDM Fused deposition modelling

FOC Field-Oriented Control

FSR Force-sensitive resistor

GPR Gaussian process regression

H-H Hodgkin-Huxley

IIT Italian Institute of Technology

IMU Inertial measurement unit

LF Left front

LR Left rear

MIT Massachusetts Institute of Technology

MLR Mesencephalic locomotor Region

PC Personal computer

PF Pattern formation

PLA Polylactic acid

PMSM Permanent magnet synchronous motor

PPO Proximal policy optimisation

QDD Quasi-direct-drive

RF Right front

RG Rhythm generator

RL Reinforcement learning

RNN Recurrent neural network

RR Right rear

RS Rigid spine

SLIP Spring-loaded inverted pendulum

TQbot Tensegrity quadruped robot

TRPO Trust Region Policy Optimization

UCB University of California, Berkeley

UFS Unidirectional flexible spine

URDF Unified robotics description format

ZMP Zero moment point

References

[1] Y. Lian, T. Wang, J. Ingham, M. A. Post, and A. Tyrrell, "Cpg-based locomotion control of a quadruped robot with an active spine," in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2023, pp. 177–189.

[2] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 822–10 825, 2008.

[3] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *2013 IEEE international conference on Robotics and automation*. IEEE, 2013, pp. 3287–3292.

[4] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1096–1103.

[5] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[6] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[7] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," *arXiv preprint arXiv:1812.11103*, 2018.

[8] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[9] A. Bergmark, "Stability of the lumbar spine: a study in mechanical engineering," *Acta Orthopaedica Scandinavica*, vol. 60, no. sup230, pp. 1–54, 1989.

[10] UnitreeRobotics, "Unitree," Available from: https://www.unitree.com Accessed March 25th, 2024.

[11] GhostRobotics, "Version60 robot," Available from: https://www.ghostrobotics.io/vision-60 Accessed July 25th, 2022.

[12] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3307–3312.

[13] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the mit cheetah 2: Control design and experiments," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.

[14] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[15] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[16] C. Semini, "Hyq-design and development of a hydraulically actuated quadruped robot," *Doctor of Philosophy (Ph. D.), University of Genoa, Italy*, 2010.

[17] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[18] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max," *IEEE/Asme Transactions on Mechatronics*, vol. 22, no. 2, pp. 635–646, 2016.

[19] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.

[20] P. Eckert, A. E. Schmerbauch, T. Horvat, K. Söhnel, M. S. Fischer, H. Witte, and A. J. Ijspeert, "Towards rich motion skills with the lightweight quadruped robot serval-a design, control and experimental study," in *International Conference on Simulation of Adaptive Behavior*. Springer, 2018, pp. 41–55.

[21] N. Kau, A. Schultz, N. Ferrante, and P. Slade, "Stanford doggo: An open-source, quasi-direct-drive quadruped," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6309–6315.

[22] C. Semini, J. Goldsmith, D. Manfredi, F. Calignano, E. P. Ambrosio, J. Pakkanen, and D. G. Caldwell, "Additive manufacturing for agile legged robots with hydraulic actuation," in *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, 2015, pp. 123–129.

[23] BostonDynamics, "Spot," Available from: https://www.bostondynamics.com/products/spot Accessed April 17th, 2023.

[24] C. Semini, V. Barasuol, M. Focchi, C. Boelens, M. Emara, S. Casella, O. Villarreal, R. Orsolino, G. Fink, S. Fahmi *et al.*, "Brief introduction to the quadruped robot hyqreal," *Istituto di Robotica e Macchine Intelligenti (I-RIM)*, 2019.

[25] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[26] A. Ananthanarayanan, M. Azadi, and S. Kim, "Towards a bio-inspired leg design for high-speed running," *Bioinspiration & biomimetics*, vol. 7, no. 4, p. 046005, 2012.

[27] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[28] M. Bjelonic, R. Grandia, M. Geilinger, O. Harley, V. S. Medeiros, V. Pajovic, E. Jelavic, S. Coros, and M. Hutter, "Offline motion libraries and online mpc for advanced mobility skills," *The International Journal of Robotics Research*, p. 02783649221102473, 2022.

[29] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[30] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[31] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[32] A. Crespi, K. Karakasiliotis, A. Guignard, and A. J. Ijspeert, "Salamandra robotica ii: an amphibious robot to study salamander-like swimming and walking gaits," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 308–320, 2013.

[33] K. Karakasiliotis, R. Thandiackal, K. Melo, T. Horvat, N. K. Mahabadi, S. Tsitkov, J.-M. Cabelguen, and A. J. Ijspeert, "From cineradiography to biorobots: an approach for designing robots to emulate and study animal locomotion," *Journal of The Royal Society Interface*, vol. 13, no. 119, p. 20151089, 2016.

[34] A. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, "Amphibot i: an amphibious snake-like robot," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163–175, 2005.

[35] A. Crespi and A. J. Ijspeert, "Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator," in *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, no. CONF, 2006, pp. 19–27.

[36] A. J. Ijspeert and A. Crespi, "Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 262–268.

[37] S. S. R. CH, R. Godiyal, T. Zodage, and T. Rane, "2dxopod-a modular robot for mimicking locomotion in vertebrates," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–16, 2021.

[38] T. Horvat, K. Melo, and A. J. Ijspeert, "Spine controller for a sprawling posture robot," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1195–1202, 2017.

[39] J. D. Caporale, Z. Feng, S. Rozen-Levy, A. M. Carter, and D. E. Koditschek, "Twisting spine or rigid torso: Exploring quadrupedal morphology via trajectory optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1177–1184.

[40] Y. Huang, Z. Bing, F. Walter, A. Rohregger, Z. Zhang, K. Huang, F. O. Morin, and A. Knoll, "Enhanced quadruped locomotion of a rat robot based on the lateral flexion of a soft actuated spine," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2622–2627.

[41] Q. Zhao, B. Ellenberger, H. Sumioka, T. Sandy, and R. Pfeifer, "The effect of spine actuation and stiffness on a pneumatically-driven quadruped robot for cheetah-like locomotion," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2013, pp. 1807–1812.

[42] BostonDynamics, "Wildcat robot," Available from: https://www.youtube.com/watch?v=wE3fmFTtP9g Accessed July 4th, 2024.

[43] S. Pouya, M. Khodabakhsh, A. Spröwitz, and A. Ijspeert, "Spinal joint compliance and actuation in a simulated bounding quadruped robot," *Autonomous Robots*, vol. 41, pp. 437–452, 2017.

[44] Z. Bing, A. Rohregger, F. Walter, Y. Huang, P. Lucas, F. O. Morin, K. Huang, and A. Knoll, "Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot," *Science Robotics*, vol. 8, no. 85, p. eadg7165, 2023.

[45] M. H. H. Kani, M. Derafshian, H. J. Bidgoly, and M. N. Ahmadabadi, "Effect of flexible spine on stability of a passive quadruped robot: Experimental results," in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 2793–2798.

[46] P. Eckert, A. Spröwitz, H. Witte, and A. J. Ijspeert, "Comparing the effect of different spine and leg designs for a small bounding quadruped robot," in *2015*

IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 3128–3133.

[47] L. Li, S. Ma, I. Tokuda, F. Asano, M. Nokata, Y. Tian, and L. Du, "Synergetic effect between limbs and spine dynamics in quadruped walking robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6818–6823.

[48] W. Li, Z. Zhou, and H. Cheng, "Dynamic locomotion of a quadruped robot with active spine via model predictive control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1185–1191.

[49] S. Bhattacharya, A. Singla, D. Dholakiya, S. Bhatnagar, B. Amrutur, A. Ghosal, S. Kolathaya *et al.*, "Learning active spine behaviors for dynamic and efficient locomotion in quadruped robots," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–6.

[50] K. Ye, K. Chung, and K. Karydis, "A novel lockable spring-loaded prismatic spine to support agile quadrupedal locomotion," *arXiv preprint arXiv:2308.00923*, 2023.

[51] J. Duperret and D. E. Koditschek, "Empirical validation of a spined sagittal-plane quadrupedal model," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1058–1064.

[52] K. Ye and K. Karydis, "Modeling and trajectory optimization for standing long jumping of a quadruped with a preloaded elastic prismatic spine," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 902–908.

[53] K. Weinmeister, P. Eckert, H. Witte, and A.-J. Ijspeert, "Cheetah-cub-s: Steering of a quadruped robot using trunk motion," in *2015 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE, 2015, pp. 1–6.

[54] A. P. Sabelhaus, H. Ji, P. Hylton, Y. Madaan, C. Yang, A. M. Agogino, J. Friesen, and V. SunSpiral, "Mechanism design and simulation of the ultra spine: a tensegrity robot," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 57120. American Society of Mechanical Engineers, 2015, p. V05AT08A059.

[55] A. P. Sabelhaus, L. J. van Vuuren, A. Joshi, E. Zhu, H. J. Garnier, K. A. Sover, J. Navarro, A. K. Agogino, and A. M. Agogino, "Design, simulation, and testing of a flexible actuated spine for quadruped robots," *arXiv preprint arXiv:1804.06527*, 2018.

[56] Q. Zhao, H. Sumioka, K. Nakajima, X. Yu, and R. Pfeifer, "Spine as an engine: effect of spine morphology on spine-driven quadruped locomotion," *Advanced Robotics*, vol. 28, no. 6, pp. 367–378, 2014.

[57] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[58] A. H. Cohen and P. Wallén, "The neuronal correlate of locomotion in fish," *Experimental brain research*, vol. 41, no. 1, pp. 11–18, 1980.

[59] S. Grillner, "Neural control of vertebrate locomotion-central mechanisms and reflex interaction with special reference to the cat," in *Feedback and motor control in invertebrates and vertebrates*. Springer, 1985, pp. 35–56.

[60] I. Delvolvé, P. Branchereau, R. Dubuc, and J.-M. Cabelguen, "Fictive rhythmic motor patterns induced by nmda in an in vitro brain stem–spinal cord preparation from an adult urodele," *Journal of Neurophysiology*, vol. 82, no. 2, pp. 1074–1077, 1999.

[61] S. Soffe and A. Roberts, "Tonic and phasic synaptic input to spinal cord motoneurons during fictive locomotion in frog embryos," *Journal of Neurophysiology*, vol. 48, no. 6, pp. 1279–1288, 1982.

[62] P. S. Stein, D. G. Stuart, S. Grillner, and A. I. Selverston, *Neurons, networks, and motor behavior*. MIT press, 1999.

[63] A. D. McClellan and W. Jang, "Mechanosensory inputs to the central pattern generators for locomotion in the lamprey spinal cord: resetting, entrainment, and computer modeling," *Journal of Neurophysiology*, vol. 70, no. 6, pp. 2442–2454, 1993.

[64] S. Rossignol, "Locomotion and its recovery after spinal injury," *Current opinion in neurobiology*, vol. 10, no. 6, pp. 708–716, 2000.

[65] S. Shik, "Orlovskii. control of walking by means of electrical stimulation of the mid-brain," *Biophysics*, vol. 11, pp. 756–765, 1966.

[66] J.-M. Cabelguen, C. Bourcier-Lucas, and R. Dubuc, "Bimodal locomotion elicited by electrical stimulation of the midbrain in the salamander notophthalmus viridescens," *Journal of Neuroscience*, vol. 23, no. 6, pp. 2434–2439, 2003.

[67] M. G. Sirota, G. V. Di Prisco, and R. Dubuc, "Stimulation of the mesencephalic locomotor region elicits controlled swimming in semi-intact lampreys," *European Journal of Neuroscience*, vol. 12, no. 11, pp. 4081–4092, 2000.

[68] T. Deliagina, P. Zelenin, P. Fagerstedt, S. Grillner, and G. Orlovsky, "Activity of reticulospinal neurons during locomotion in the freely behaving lamprey," *Journal of neurophysiology*, vol. 83, no. 2, pp. 853–863, 2000.

[69] J. Cheng, R. B. Stein, K. Jovanovic, K. Yoshida, D. J. Bennett, and Y. Han, "Identification, localization, and modulation of neural networks for walking in the mudpuppy (necturus maculatus) spinal cord," *Journal of Neuroscience*, vol. 18, no. 11, pp. 4295–4304, 1998.

[70] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[71] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on cpg-inspired control models and system implementation," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 3, pp. 441–456, 2013.

[72] J. Hellgren, S. Grillner, and A. Lansner, "Computer simulation of the segmental neural network generating locomotion in lamprey by using populations of network interneurons," *Biological cybernetics*, vol. 68, no. 1, pp. 1–13, 1992.

[73] H. Traven, L. Brodin, A. Lansner, O. Ekeberg, P. Wallén, and S. Grillner, "Computer simulations of nmda and non-nmda receptor-mediated synaptic drive: sensory and supraspinal modulation of neurons and small networks," *Journal of neurophysiology*, vol. 70, no. 2, pp. 695–709, 1993.

[74] J. T. Buchanan, "Neural network simulations of coupled locomotor oscillators in the lamprey spinal cord," *Biological cybernetics*, vol. 66, no. 4, pp. 367–374, 1992.

[75] Ö. Ekeberg, "A combined neuronal and mechanical model of fish swimming," *Biological cybernetics*, vol. 69, no. 5, pp. 363–374, 1993.

[76] T. L. Williams, "Phase coupling by synaptic spread in chains of coupled neuronal oscillators," *Science*, vol. 258, no. 5082, pp. 662–665, 1992.

[77] ——, "Phase coupling in simulated chains of coupled oscillators representing the lamprey spinal cord," *Neural Computation*, vol. 4, no. 4, pp. 546–558, 1992.

[78] D. Lachat, A. Crespi, and A. J. Ijspeert, "Boxybot: a swimming and crawling fish robot controlled by a central pattern generator," in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.* IEEE, 2006, pp. 643–648.

[79] A. Crespi, D. Lachat, A. Pasquier, and A. J. Ijspeert, "Controlling swimming and crawling in a fish robot using a central pattern generator," *Autonomous Robots*, vol. 25, no. 1, pp. 3–13, 2008.

[80] F. Xie, Y. Zhong, R. Du, and Z. Li, "Central pattern generator (cpg) control of a biomimetic robot fish for multimodal swimming," *Journal of Bionic Engineering*, vol. 16, no. 2, pp. 222–234, 2019.

[81] S. Inagaki, H. Yuasa, and T. Arai, "Cpg model for autonomous decentralized multi-legged robot system—generation and transition of oscillation patterns and dynamics of oscillators," *Robotics and Autonomous Systems*, vol. 44, no. 3-4, pp. 171–179, 2003.

[82] P. Arena, L. Fortuna, M. Frasca, and G. Sicurella, "An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 4, pp. 1823–1837, 2004.

[83] S. Inagaki, H. Yuasa, T. Suzuki, and T. Arai, "Wave cpg model for autonomous decentralized multi-legged robot: Gait generation and walking speed control," *Robotics and Autonomous systems*, vol. 54, no. 2, pp. 118–126, 2006.

[84] H. Yu, H. Gao, and Z. Deng, "Enhancing adaptability with local reactive behaviors for hexapod walking robot via sensory feedback integrated central pattern generator," *Robotics and Autonomous Systems*, vol. 124, p. 103401, 2020.

[85] H. Kimura, S. Akiyama, and K. Sakurama, "Realization of dynamic walking and running of the quadruped using neural oscillator," *Autonomous robots*, vol. 7, no. 3, pp. 247–258, 1999.

[86] H. Kimura, Y. Fukuoka, and K. Konaga, "Adaptive dynamic walking of a quadruped robot using a neural system model," *Advanced Robotics*, vol. 15, no. 8, pp. 859–878, 2001.

[87] Y. Fukuoka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, 2003.

[88] Y. Fukuoka, H. Kimura, Y. Hada, and K. Takase, "Adaptive dynamic walking of a quadruped robot'tekken'on irregular terrain using a neural system model," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 2037–2042.

[89] H. Kimura and Y. Fukuoka, "Biologically inspired adaptive dynamic walking in outdoor environment using a self-contained quadruped robot:'tekken2'," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1. IEEE, 2004, pp. 986–991.

[90] H. Kimura, Y. Fukuoka, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 475–490, 2007.

[91] J. Zhang, F. Gao, X. Han, X. Chen, and X. Han, "Trot gait design and cpg method for a quadruped robot," *Journal of Bionic Engineering*, vol. 11, no. 1, pp. 18–25, 2014.

[92] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological cybernetics*, vol. 65, no. 3, pp. 147–159, 1991.

[93] G. Taga, "A model of the neuro-musculo-skeletal system for human locomotion," *Biological cybernetics*, vol. 73, no. 2, pp. 97–111, 1995.

[94] S. Aoi and K. Tsuchiya, "Locomotion control of a biped robot using nonlinear oscillators," *Autonomous robots*, vol. 19, no. 3, pp. 219–232, 2005.

[95] ——, "Stability analysis of a simple walking model driven by an oscillator with a phase reset using sensory feedback," *IEEE Transactions on robotics*, vol. 22, no. 2, pp. 391–397, 2006.

[96] J.-J. Kim, J.-W. Lee, and J.-J. Lee, "Central pattern generator parameter search for a biped walking robot using nonparametric estimation based particle swarm optimization," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 447–457, 2009.

[97] K. Matsuoka, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biological cybernetics*, vol. 52, no. 6, pp. 367–376, 1985.

[98] ——, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological cybernetics*, vol. 56, no. 5, pp. 345–353, 1987.

[99] Z. Lu, S. Ma, B. Li, and Y. Wang, "3d locomotion of a snake-like robot controlled by cyclic inhibitory cpg model," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3897–3902.

[100] H. R. Wilson and J. D. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophysical journal*, vol. 12, no. 1, pp. 1–24, 1972.

[101] S. Campbell and D. Wang, "Synchronization and desynchronization in a network of locally coupled wilson-cowan oscillators," *IEEE transactions on neural networks*, vol. 7, no. 3, pp. 541–554, 1996.

[102] K. Nakada, T. Asai, and Y. Amemiya, "Biologically-inspired locomotion controller for a quadruped walking robot: Analog ic implementation of," *Journal of Robotics and Mechatronics*, vol. 16, no. 4, p. 397, 2004.

[103] ——, "Design of an artificial central pattern generator with feedback controller," *Intelligent Automation & Soft Computing*, vol. 10, no. 2, pp. 185–192, 2004.

[104] J. A. Acebrón, L. L. Bonilla, C. J. P. Vicente, F. Ritort, and R. Spigler, "The kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of modern physics*, vol. 77, no. 1, p. 137, 2005.

[105] M. Breakspear, S. Heitmann, and A. Daffertshofer, "Generative models of cortical oscillations: neurobiological implications of the kuramoto model," *Frontiers in human neuroscience*, vol. 4, p. 190, 2010.

[106] J. Yu, R. Ding, Q. Yang, M. Tan, W. Wang, and J. Zhang, "On a bio-inspired amphibious robot capable of multimodal motion," *IEEE/ASME Transactions On Mechatronics*, vol. 17, no. 5, pp. 847–856, 2011.

[107] J. Conradt and P. Varshavskaya, "Distributed central pattern generator control for a serpentine robot," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. Citeseer, 2003, pp. 338–341.

[108] P. G. Drazin and P. D. Drazin, *Nonlinear systems*. Cambridge University Press, 1992, no. 10.

[109] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 819–824.

[110] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D: Nonlinear Phenomena*, vol. 216, no. 2, pp. 269–281, 2006.

[111] J. Buchli and A. J. Ijspeert, "Self-organized adaptive legged locomotion in a compliant quadruped robot," *Autonomous Robots*, vol. 25, no. 4, pp. 331–347, 2008.

[112] A. C. de Pina Filho and M. S. Dutra, "Application of hybrid van der pol-rayleigh oscillators for modeling of a bipedal robot," *Mechanics of Solids in Brazil*, vol. 1, pp. 209–221, 2009.

[113] C. Liu, Q. Chen, and J. Zhang, "Coupled van der pol oscillators utilised as central pattern generators for quadruped locomotion," in *2009 Chinese Control and Decision Conference*. IEEE, 2009, pp. 3677–3682.

[114] I. A. Rybak, N. A. Shevtsova, M. Lafreniere-Roula, and D. A. McCrea, "Modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion," *The Journal of physiology*, vol. 577, no. 2, pp. 617–639, 2006.

[115] I. A. Rybak, K. Stecina, N. A. Shevtsova, and D. A. McCrea, "Modelling spinal circuitry involved in locomotor pattern generation: insights from the effects of afferent stimulation," *The Journal of physiology*, vol. 577, no. 2, pp. 641–658, 2006.

[116] S. Aoi, Y. Egi, R. Sugimoto, T. Yamashita, S. Fujiki, and K. Tsuchiya, "Functional roles of phase resetting in the gait transition of a biped robot from quadrupedal to bipedal locomotion," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1244–1259, 2012.

[117] J. Shan and F. Nagashima, "Neural locomotion controller design and implementation for humanoid robot hoap-1," in *20th annual conference of the robotics society of Japan*, 2002.

[118] D. T. Tran, I. M. Koo, Y. H. Lee, H. Moon, S. Park, J. C. Koo, and H. R. Choi, "Central pattern generator based reflexive control of quadruped walking robots using a recurrent neural network," *Robotics and autonomous systems*, vol. 62, no. 10, pp. 1497–1516, 2014.

[119] S. Dasgupta, D. Goldschmidt, F. Wörgötter, and P. Manoonpong, "Distributed recurrent neural forward models with synaptic adaptation and cpg-based control for complex behaviors of walking robots," *Frontiers in neurorobotics*, vol. 9, p. 10, 2015.

[120] F. Ponulak, D. Belter, and A. Kasinski, "Adaptive central pattern generator based on spiking neural networks," in *Proceedings of EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices*, 2006, pp. 121–122.

[121] A. C. de Pina Filho, M. S. Dutra, and L. S. Raptopoulos, "Modeling of a bipedal robot using mutually coupled rayleigh oscillators," *Biological cybernetics*, vol. 92, no. 1, pp. 1–7, 2005.

[122] J. Or, "A hybrid cpg–zmp control system for stable walking of a simulated flexible spine humanoid robot," *Neural Networks*, vol. 23, no. 3, pp. 452–460, 2010.

[123] M. Mokhtari, M. Taghizadeh, and M. Mazare, "Hybrid adaptive robust control based on cpg and zmp for a lower limb exoskeleton," *Robotica*, vol. 39, no. 2, pp. 181–199, 2021.

[124] R. Moreno and J. Gomez, "Central pattern generators and hormone inspired messages: A hybrid control strategy to implement motor primitives on chain type modular reconfigurable robots," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1014–1019.

[125] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[126] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[127] A. D. Laud, *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[128] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[129] H. Benbrahim, J. Doleac, J. Franklin, and O. Selfridge, "Real-time learning: A ball on a beam," in *International Joint Conference on Neural Networks*, vol. 1, 1992, pp. 98–103.

[130] B. Nemec, M. Zorko, and L. Žlajpah, "Learning of a ball-in-a-cup playing robot," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*. IEEE, 2010, pp. 297–301.

[131] M. Tokic, W. Ertel, and J. Fessler, "The crawler, a class room demonstrator for reinforcement learning," in *Twenty-Second International FLAIRS Conference*, 2009.

[132] H. Kimura, T. Yamashita, and S. Kobayashi, "Reinforcement learning of walking behavior for a four-legged robot," *IEEJ Transactions on Electronics, Information and Systems*, vol. 122, no. 3, pp. 330–337, 2002.

[133] M. S. Erden and K. Leblebicioğlu, "Free gait generation with reinforcement learning for a six-legged robot," *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 199–212, 2008.

[134] Z. Kalmár, C. Szepesvári, and A. Lorincz, "Modular reinforcement learning: An application to a real robot task," in *European Workshop on Learning Robots*. Springer, 1997, pp. 29–45.

[135] P. Fidelman and P. Stone, "Learning ball acquisition on a physical robot," in *2004 International Symposium on Robotics and Automation (ISRA)*, 2004, p. 6.

[136] M. Huber and R. A. Grupen, "A feedback control structure for on-line learning tasks," *Robotics and autonomous systems*, vol. 22, no. 3-4, pp. 303–315, 1997.

[137] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive behavior acquisition for a real robot by vision-based reinforcement learning," *Machine learning*, vol. 23, no. 2, pp. 279–303, 1996.

[138] V. Soni and S. Singh, "Reinforcement learning of hierarchical skills on the sony aibo robot," in *Proceedings of the Fifth International Conference on Development and Learning*, 2006.

[139] R. Platt, R. A. Grupen, and A. H. Fagg, "Improving grasp skills using schema structured learning," in *5th International Conference on Development and Learning*, 2006.

[140] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[141] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE transactions on automatic control*, vol. 42, no. 5, pp. 674–690, 1997.

[142] G. J. Gordon, *Approximate solutions to Markov decision processes*. Carnegie Mellon University, 1999.

[143] C. Gaskett, L. Fletcher, and A. Zelinsky, "Reinforcement learning for a vision based mobile robot," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 1. IEEE, 2000, pp. 403–409.

[144] Y. Duan, B. Cui, and H. Yang, "Robot navigation based on fuzzy rl algorithm," in *International Symposium on Neural Networks*. Springer, 2008, pp. 391–399.

[145] S. Thrun, "An approach to learning mobile robot navigation," *Robotics and Autonomous systems*, vol. 15, no. 4, pp. 301–319, 1995.

[146] H. Benbrahim and J. A. Franklin, "Biped dynamic walking using reinforcement learning," *Robotics and Autonomous Systems*, vol. 22, no. 3-4, pp. 283–302, 1997.

[147] S. Mahadevan and J. Connell, "Automatic programming of behavior-based robots using reinforcement learning," *Artificial intelligence*, vol. 55, no. 2-3, pp. 311–365, 1992.

[148] C. F. Touzet, "Neural reinforcement learning for behaviour synthesis," *Robotics and Autonomous Systems*, vol. 22, no. 3-4, pp. 251–281, 1997.

[149] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[150] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[151] A. Rottmann, C. Plagemann, P. Hilgers, and W. Burgard, "Autonomous blimp control using model-free reinforcement learning in a continuous state and action space," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1895–1900.

[152] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato, "A kendama learning robot based on bi-directional theory," *Neural networks*, vol. 9, no. 8, pp. 1281–1302, 1996.

[153] J. W. Roberts, L. Moret, J. Zhang, and R. Tedrake, "Motor learning at intermediate reynolds number: experiments with policy gradient on the flapping flight of a rigid wing," in *From motor learning to interaction learning in robots*. Springer, 2010, pp. 293–309.

[154] S. Kuindersma, R. Grupen, and A. Barto, "Learning dynamic arm motions for postural recovery," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, pp. 7–12.

[155] J. Kober and J. Peters, "Policy search for motor primitives in robotics," in *Learning Motor Skills*. Springer, 2014, pp. 83–117.

[156] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," *Advances in neural information processing systems*, vol. 15, 2002.

[157] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 3. IEEE, 2004, pp. 2619–2624.

[158] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7-9, pp. 1180–1190, 2008.

[159] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[160] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter, "Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 910–922, 2011.

[161] T. Geng, B. Porr, and F. Wörgötter, "Fast biped walking with a reflexive controller and real-time policy searching," in *Advances in Neural Information Processing Systems*. Citeseer, 2006, pp. 427–434.

[162] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng, "Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 213–228, 2008.

[163] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *Advanced Robotics*, vol. 21, no. 13, pp. 1521–1544, 2007.

[164] H.-I. Lin and C.-C. Lai, "Learning collision-free reaching skill from primitives," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2383–2388.

[165] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 465–472.

[166] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," in *Robotics: Science and Systems VII*, vol. 7, 2011, pp. 57–64.

[167] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Franitza, "Design, analysis and realization of tendon-based parallel manipulators," *Mechanism and Machine Theory*, vol. 40, no. 4, pp. 429–445, 2005.

[168] UnitreeRobotics, "A1 motor," Available from: https://www.unitree.com/products/a1_motor Accessed July 25th, 2022.

[169] S. Kawamura, H. Kino, and C. Won, "High-speed manipulation by using parallel wire-driven robots," *Robotica*, vol. 18, no. 1, pp. 13–21, 2000.

[170] L. Li, C. Wang, and G. Xie, "A general cpg network and its implementation on the microcontroller," *Neurocomputing*, vol. 167, pp. 299–305, 2015.

[171] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, 2013.

[172] S. Gay, J. Santos-Victor, and A. Ijspeert, "Learning robot gait stability using neural networks as sensory feedback function for central pattern generators," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. Ieee, 2013, pp. 194–201.

[173] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.

[174] C. Liu, Y. Chen, J. Zhang, and Q. Chen, "Cpg driven locomotion control of quadruped robot," in *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 2368–2373.

[175] S. H. Strogatz, "From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators," *Physica D: Nonlinear Phenomena*, vol. 143, no. 1-4, pp. 1–20, 2000.

[176] G. Bellegarda and A. Ijspeert, "Cpg-rl: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 547–12 554, 2022.

[177] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.