Doctoral Dissertation

Studies on a Novel Design Principle for a Navigation Robot in Unknown Environments – Explicit-Implicit Coordinated Robot Design Principle –

Runze Xiao

December, 2023

Graduate School of Engineering,
Osaka University

OSAKA UNIVERSITY

Abstract

Graduate School of Engineering

Doctor of Engineering

Studies on a Novel Design Principle for a Navigation Robot in Unknown Environments – Explicit-Implicit Coordinated Robot Design Principle –

by Runze Xiao

This study aims to introduce a novel design principle for navigation robots. By leveraging the environment as an "assist" rather than viewing it as an "obstacle", these robots can navigate unknown environments using simpler systems while ensuring substantial adaptability. As artificial intelligence(AI) advances, robotics, serving as a vital link between AI and the physical world, has become increasingly important. Among its capabilities, navigation in unknown environments stands as a cornerstone technology, essential for the application of robots in real-world human settings. Traditional research in this field often views the environment as an "obstacle", requiring a complex suite of control and sensing systems to avoid or confront these challenges. This approach does indeed work well in specific settings, but as robots encounter more complex environments, it not only reduces their adaptability but also makes their control and sensing systems more complex and costly. To address this issue, our research steps away from the traditional approach and proposes a novel navigation robot design principle called the "Explicit-Implicit Coordinated Robot Design Principle (E-I Coordinated Robot Design Principle)," reframing the environment as an "assist" rather than an "obstacle". In this design principle, we view the beneficial influences of the environment on the robot as a form of control, referred to in our study as "implicit control", while the traditional controller's influence is termed "explicit control". This principle advocates for designing both the physical structure and control system of the robot in a way that maximizes the beneficial implicit control as much as possible while harmoniously coordinating them with explicit control. Adopting this approach, navigation robots designed in this manner can leverage the power of the environment to fulfill some of their navigational functions, so as to simplify the robot's system while ensuring considerable adaptability to the environment. Following this design principle, we developed several centipede-like navigation robots for different tasks: individual navigation in 2D unknown environments, swarm navigation in 2D unknown environments, and individual navigation in 3D unknown environments. Using these robots as examples, we demonstrated the specific implementation steps and advantages of our design principle through simulations or experiments, thereby validating its feasibility.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

This study aims to introduce a novel design principle for navigation robots. By leveraging the environment as an "assist" rather than viewing it as an "obstacle", these robots can navigate unknown environments using simpler systems while ensuring substantial adaptability.

In today's era, generative language models and other artificial intelligence(AI) technologies are rapidly transforming our world[1][2][3][4]. From voice assistants[5][6] to the creation of media content[7][8], AI's influence is ubiquitous. Despite this, the current capabilities of AI are primarily confined to processing digitized data such as text and images. However, when it comes to industrial or everyday problems that require deep interaction with the physical world, there remains a significant gap between AI and reality.

To bridge this gap, robotics emerges as a key conduit linking AI with the physical world. Robots are capable not only of translating digitized commands into physical actions in the real world but also of combining the strengths of "brain" (AI) and "body" (mechanical action) to tackle complex real-life challenges. They are pivotal in transforming AI's theoretical potential into practical applications. Consequently, robotics is increasingly becoming the central force driving future transformations.

In this transformation process, robotic navigation technology plays a crucial role. From deep-sea exploration[9] to space missions[10], and disaster response[11], effective navigation in unknown environments is key for the successful application of robotics in high-risk, high-value areas. However, while existing robotic navigation technologies have achieved notable successes in certain specific applications, the increasing complexity of unknown environments faced by robots often leads to corresponding increases in the expense and complexity of their control and sensing systems. Actually, in both theory and practice, the approach of relying on ever-increasing system complexity to cope with and counteract unknown environments faces significant challenges. The main obstacles include high costs due to expensive hardware and maintenance; reduced reliability because complex systems introduce more points of failure; and theoretical limitations in systems based on finite designs struggling to address the infinite variations of unknown environments fully. These issues suggest that the current mindset of engaging in an "arms race" with the environment may not be the most effective strategy.

This leads to my research: a novel approach to designing navigation robots. The core of this design principle is a shift in mindset, viewing the environment as an "assist" for robots, not just an "obstacle" to be countered or avoided. In other words, It capitalizes on the environment's own forces to aid in robot navigation. This design principle, which leverages the environment's power to perform certain robot functions, not only reduces dependence on expensive sensors and complex control systems, thus lowering costs, but also ensures considerable adaptability and robustness of robots in unknown environments. It offers a fresh perspective to break through the limitations of existing technology.

In this paper, I will first delve deeply into the philosophy and methodologies behind the proposed design principle of navigation robots. Subsequently, I will illustrate the specific implementation steps of this principle using examples of centipede-like navigation robots designed according to this methodology, which are applied to various types of tasks and environments. This will highlight the unique features and advantages of robots developed using this design principle, furthermore demonstrating the feasibility and significant importance of it.

1.1 Overview of Robot Navigation Problem in Unknown Environments

The problem of robot navigation in unknown environments refers to how a robot can effectively move to a target location in an environment where it has no prior knowledge or maps, as illustrated in **Fig. 1.1**. This issue holds a central position in the field of robot research and application. It is the cornerstone of achieving robot autonomy and is crucial for enhancing the practicality and intelligence of robots. Research on this problem is not only significant for enabling robot applications in various complex environments such as disaster rescue[11], autonomous driving[12], home service[13], and space exploration[10], but also indirectly promotes advancements in fields like artificial intelligence, machine learning, computer vision, and sensor technology. With the advancement of technology, the capability of robot navigation in unknown environments will directly influence the future direction of robot technology and the expansion of its application areas.

1.2 Current Research and Challenges in Robot Navigation in Unknown Environments

Robot navigation in unknown environments is one of the foundations and core elements for achieving autonomous movement in robots performing various tasks. Considerable progress has been made in this field through extensive research.

Firstly, methods based on SLAM (Simultaneous Localization and Mapping)[14][15][16][17] and path planning[18][19][20][21] have been widely studied and implemented, yielding significant results. SLAM is a computational process used by robots and automated systems to map an environment while simultaneously determining its location within that map. In the context of robot navigation in unknown settings, robots often first use or integrate information from sensors like LIDAR, cameras, and inertial navigation systems to map the unknown environment with SLAM technology. Subsequently, they complete the navigation task using path planning algorithms based on the map information derived from SLAM. Notable

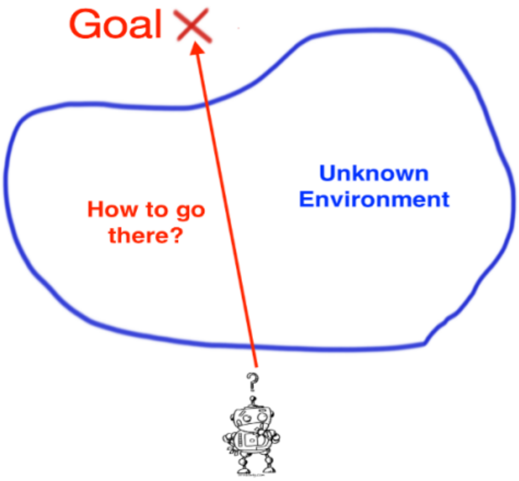


FIGURE 1.1: Robot navigation problem in unknown environments

methods in the SLAM domain include Gmapping[23], Cartographer[24], and Hector[25], which combine LIDAR and inertial navigation systems for environmental modeling[22]; RTAB-MAP[27] and VSLAM[28], which use depth cameras for 3D modeling; and methods utilizing 3D laser scanners[26] for 3D environmental modeling. For path planning, existing research often employs mature map-based algorithms like A*[18], Dijkstra[19], RRT*(Rapidly-exploring Random Tree)[20], and Genetic algorithms[21]. By integrating SLAM and path planning algorithms, robot navigation in unknown environments, especially in some indoor or simple outdoor settings, can be effectively accomplished.

Another common approach to robot navigation is based on machine learning[29]. Through learning from existing experience and data or continuous training in virtual environments, robots can not only gain the ability to recognize complex environments but can also acquire the optimal behavioral strategies for different settings. For instance, in methods based on supervised[30], unsupervised[31], and self-supervised learning[32][33][34][35], robots initially learn from existing labeled or unlabeled environmental and navigational data. They then acquire an understanding of the relationship between sensor inputs and the navigability of the environment, using this knowledge to establish corresponding navigation algorithms. And in methods based on reinforcement learning[36][37][38][39], robots can train an end-to-end navigation method from sensor input to action output in simulated environments through pre-training simulations. This learned model can then be transferred to real-world robots for navigation. These two methods often perform well within environments covered by existing knowledge and training, and their end-to-end control approach makes the controllers more streamlined compared to SLAM-based navigation methods.

Additionally, another common method for robot navigation is the behavior-based approach. The essence of this method lies in designing a combination of simple behaviors that directly respond to the environment, achieving flexible and adaptive navigation in unknown settings[40][41][42][43]. In this approach, designers often pre-set several basic navigational behaviors for the robot, such as approaching targets, avoiding obstacles, and loop escaping.

The robot then fuses the outputs of these different behaviors based on current sensor input information, using a behavior coordinator, to form a comprehensive set of navigational actions. As this method directly responds to the environment, it is particularly effective in dynamic and unpredictable settings. Moreover, its behavior-based control can rapidly respond to environmental changes, providing real-time navigation capabilities.

At the same time, inspired by the behavior of animal swarms in nature, many researchers have introduced the concept of "swarm" into the field of robot navigation in unknown environments, proposing the research direction of swarm robot navigation. By leveraging the coordination and information exchange among multiple robots, this approach allows even very simple individual robots to explore and navigate unknown environments by relying on the strength of the swarm. Moreover, It also significantly enhances the efficiency of exploring and navigating these environments. For example, robots can create markers using pheromones emitted by individual robots, allowing navigation without colliding with the environment through simple marker following[44][45][46]. Additionally, by forming robot flocks[47][48][49][50], simple individual robots can explore and navigate unknown environments without colliding with each other or the environment, using information sharing and interactions within the swarm, as seen in approaches like leader-follower swarm navigation[52][53] and shepherd navigation[54][55][56]. Conversely, in [57] and [58], which contrasts with [47], [48], [49] and [50], the robot swarms do not maintain a flocking formation. They complete swarm navigation tasks in a dispersed manner using methods such as shared map memories[57] or sequentially executing exclusive path planning[58]. These strategies also inherently prioritize the prevention of collisions, both with the environment and among the robots. Furthermore, by taking advantage of the swarm, individual robots can use mutual observation and information exchange to correct observation signals, thereby reducing errors caused by observation noise and signal intermittence, and improving the accuracy of observations[59][60][61].

All the aforementioned research achievements in robot navigation have their strengths and have made significant contributions within their respective fields. However, one shared feature across these studies is that they typically view the environment as an "obstacle" to be dealt with, requiring the consumption of software and hardware resources to either counter or avoid it, as illustrated in **Fig. 1.2(A)**. Following this approach, these studies generally encounter two types of problems:

1. **The requirement for complex control and sensing systems. And the more complex the environment, the more complex the required systems become.**
2. **Limited adaptability to unknown environments due to constraints of the available software and hardware.**

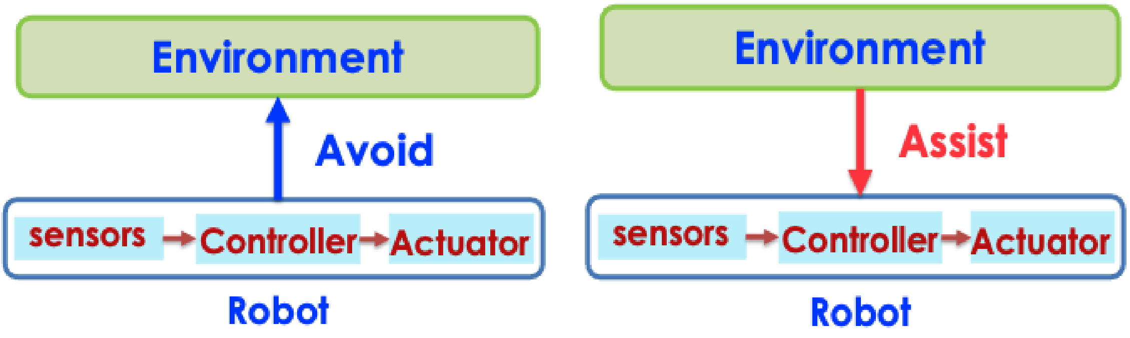
The following details the two identified issues:

- **Complexity of the system:** Due to the reliance of existing robot navigation methods on continually increasing system complexity to address and counteract increasingly complex unknown environments, the issue of system complexity becomes more pronounced as the complexity of the environment escalates. Specifically, in navigation methods that utilize SLAM and path planning[23][24][25][27][28], the system must first perform an all-around observation and modeling of the environment before engaging in path planning. This process in itself tends to be quite intricate and time-consuming. Moreover, this approach often necessitates high-cost sensors like lidars and cameras, along with complex SLAM and path planning control systems, resulting in an overall cumbersome and bloated system. In learning-based navigation methods, although end-to-end behavior control from input to output significantly simplifies the process compared to SLAM modeling, the learning phase, especially in supervised learning that requires manual labeling of data, demands substantial human labor and adds to system complexity[30]. Even in swarm robot navigation, which utilizes simple individual robots, the complexity of communication and coordination control systems escalates substantially as the number and scale of robots increase[47][48][49][50]. Finally, with the rise in system complexity, inevitable issues such as increased costs and reduced system reliability emerge naturally.

- **Limitations in adaptability:** Another major issue in existing research is the insufficient adaptability to unknown environments with theoretically limitless possibilities. In methods based on SLAM, most existing studies focus on relatively structured indoor environments and simpler outdoor settings, with limited adaptability to complex outdoor environments[23][24][25][27][28]. In machine learning-based methods, robots often heavily rely on training and knowledge, leading to poor adaptability in unknown environments beyond their cognition[29][30][32][33][34][35]. Additionally, methods that involve learning from simulation to reality face the challenge of a mismatch between simulated environments and the real world[36][37][38][39]. In behavior-based methods[40][41][42][43], adapting to truly complex environments and navigation tasks with a limited set of pre-designed behaviors is also challenging. In fact, attempting to counter or evade theoretically limitless complex environments with limited human design is inherently difficult.

In fact, these two issues are not specific, independent technical problems faced by specific robot navigation methods; instead, they are inherent problems arising from the mindset of treating the environment as an obstacle and relying on ever-increasing system complexity to confront or avoid it. Therefore, without changing the aforementioned robot design approach of engaging in an “arms race” with the environment, these issues may prove difficult to resolve completely.

At the same time, some animals in nature seem unaffected by these issues. Many of them, without highly complex neural and sensory systems, can still move freely in complex and unknown environments. For example, worms often lack advanced sensory organs, but they can navigate around obstacles by feeling their way through and conforming to the guidance of these barriers, sometimes even using the obstacles to aid in accelerating their movement.[62]. Single-celled organisms like amoebae, despite being just one cell, can still maneuver in complex environments by altering their shape to comply with environmental cues[63]. In the animal swarm, ants use interactions and mutual contact to form structures like “ant bridges”[66][67] or “ant islands”[68][69] to overcome obstacles. Observing these



(A) Traditional scheme

(B) Proposed scheme

FIGURE 1.2: Robot navigation scheme

species reveals that their relationship with the environment is not one of separation but of integration. Elements of the environment that are perceived as disturbances, such as contact between individuals or between individuals and the environment, are in fact extensively utilized[64][65].

Inspired by these animals in nature, a new strategy has emerged that abandons the traditional belief that "the environment is an obstacle that requires software and hardware resources to confront or avoid." This strategy involves viewing the environment as an "assist", using its assistance to supplement or replace part of the robot's tasks. Based on this mindset, the concept of implicit control has naturally evolved, as illustrated in **Fig. 1.2(B)**.

1.3 Implicit Control

Indeed, leveraging the environment's interaction to facilitate the movement of robots is not a new concept. Many researchers have employed this approach to varying extents, producing notable research outcomes. For instance, Pfeifer et al. introduced the concept of "embodiment," a strategy that involves modifying the dynamics of the robot to utilize interactions between the robot and its environment for adaptation[70]. Ishiguro et al. proposed a snake-like robot that actively uses interactions with obstacles, detected through tactile sensors, to aid navigation[71][72]. Greer et al. suggested an air-pressure-driven soft growing robot that elongates while using the reaction force from an obstacle to creating a bend in its soft body, thus bypassing the obstacle to reach its destination[73][74]. Subsequently, by abstracting the common elements from this type of research that utilizes environmental forces and considering environmental effects as a distinct part of the control model of the system, Osuka et al. proposed the theory of implicit control[75][76]. We will introduce implicit control from four aspects: its meaning, its functions, the designer's task, and the current research progress and bottlenecks.

1.3.1 Meaning of implicit control

Implicit control refers to an interaction between the environment and the system which is beneficial towards the control objective. In traditional control theory, control is generated by a human-designed controller, requiring the achievement of the control objective while combating disturbances from the environment. However, environmental influences on the system are not necessarily harmful disturbances that inhibit the control objective. In fact, some environmental interactions are beneficial for control. For instance, in this study, certain collisions of the robot with the environment, can actually guide the robot closer to the navigation target or assist in finding an effective path toward it, which can be advantageous for the control objective of "navigation towards the target". If we view the whole robot-environment system in the language of control theory, we can get a block diagram as shown in **Fig. 1.3**(A). As we can see in this figure, the control plant (robot) is affected by both controller control and interactions with the environment. Sometimes, the interactions with the environment are harmful, the interaction is a disturbance to the system at this time. And sometimes, the interactions are helpful, the interactions can become assist at this time, as shown in **Fig. 1.3**(B). In our research, we regard the helpful part of the robot-environment interactions as a form of control in a broad sense, and we call it "implicit control". At the same time, we call the controller commands-based control "explicit control" in this research. It's worth noting that in the broad definition of implicit control, the term interaction does not only refer to mechanical interactions with the environment. Instead, any beneficial influence exerted by the environment on the robotic system can be considered as implicit control. This includes beneficial sensor noise or other external factors caused by environmental interference.

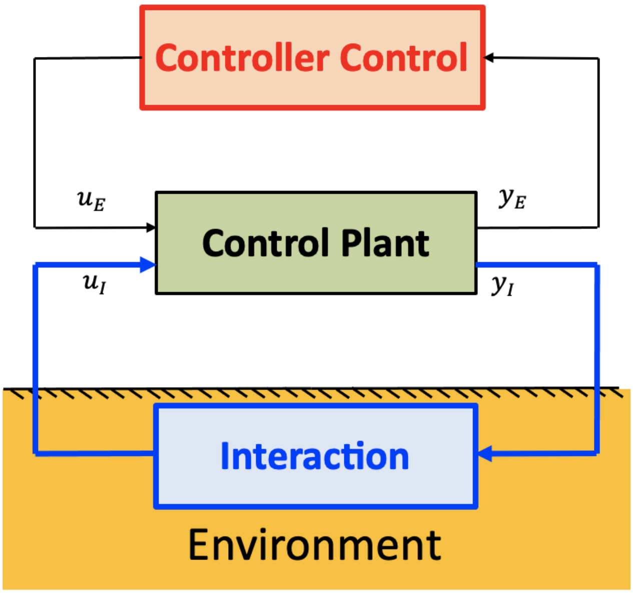
1.3.2 Function of implicit control

Simplifying robotic systems through environmental assistance

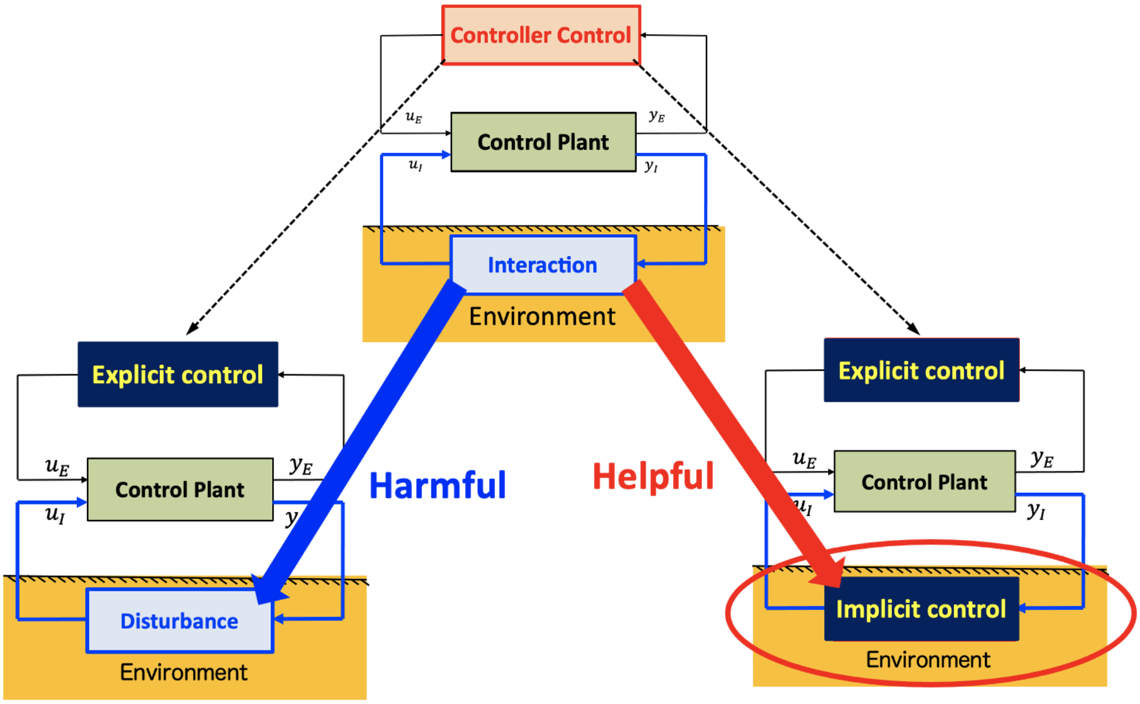
Implicit control can replace or share the work of explicit control, accomplishing complex tasks with a simpler system by leveraging environmental factors. For instance, in this research, we utilize interactions such as beneficial obstacle collisions with the navigation robot, beneficial inter-robot collisions in the robot swarm, or even the beneficial noise and intervals of target direction signals as implicit control. This implicit control can supplant the traditional functions achieved by explicit control, such as obstacle observation and modeling, path planning, and obstacle avoidance, ultimately enabling robotic navigation and exploration in unknown environments with minimal explicit control by utilizing implicit control. This greatly simplifies the complexity of the system.

Enhancing adaptability to unknown environments

A significant challenge in robotics is employing a "finite" system to deal with theoretically "infinite" possibilities in unknown environments. Conventional methods based on learning or pre-designed behavior are often limited to "finite" experiences and struggle to adapt to entirely unknown environments. Implicit control, by leveraging environmental factors, transforms the environment's "infinite" randomness into the system's "infinite" randomness,



(A) Block diagram of robot sysytem



(B) Defination of implicit control and explicit control

FIGURE 1.3: Meaning of implicit control

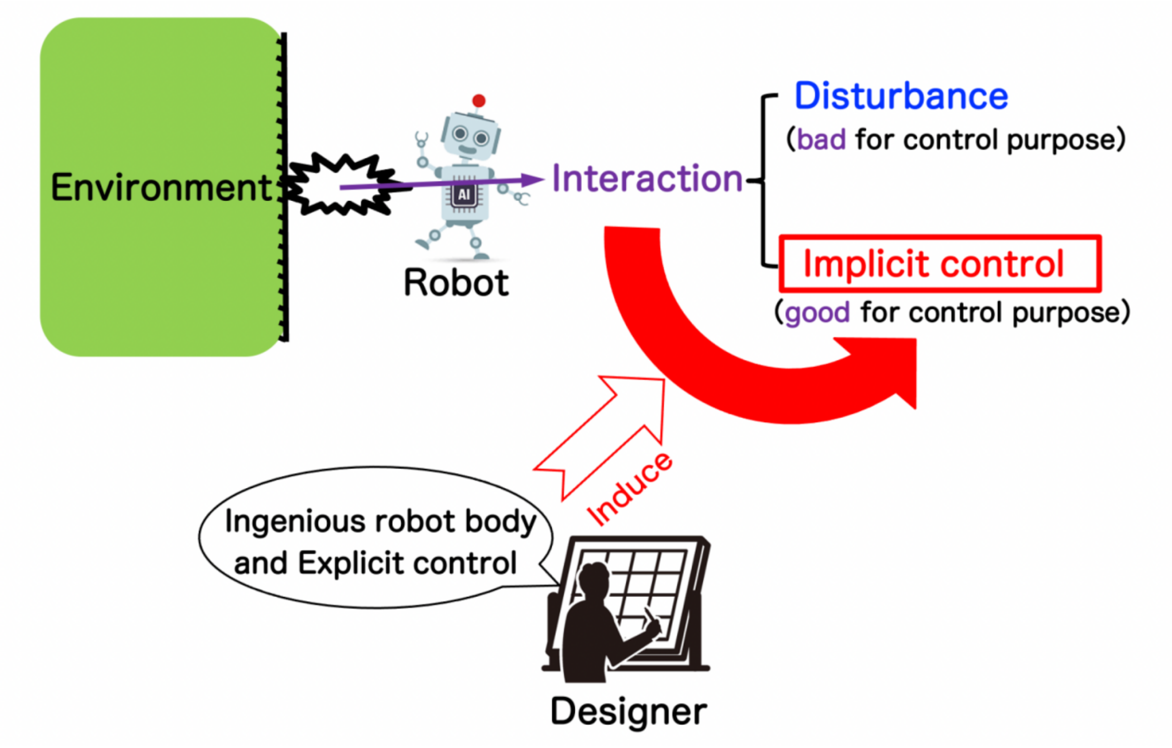
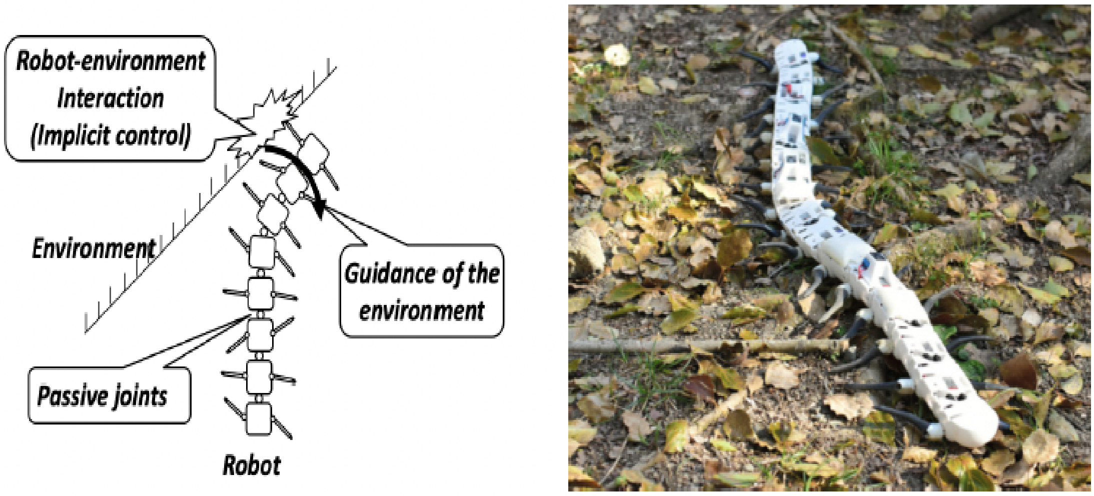

Environment
Robot
AI
Interaction
Disturbance
(bad for control purpose)
Implicit control
(good for control purpose)
Induce
Ingenious robot body and Explicit control
Designer

FIGURE 1.4: Designer's task

theoretically enabling unlimited adaptability. For example, in this study, the robot's action is influenced by both the "finite" explicit control and the "infinite" implicit control resulting from collisions with the environment or other robots. The combination of these two factors gives the robot's movement pattern a rich variety. While keeping the main goal of navigation, it provides a high level of behavior that changes with environmental variations, allowing the robot to adjust its actions smoothly and dynamically to fit different unknown environments.

1.3.3 Designer's task

Since implicit control stems from environmental influences and varies with environmental changes, it is inherently unpredictable and challenging to manipulate. However, designers can induce as much implicit control as possible to aid the system through ingenious robot body structure and explicit control design, as shown in **Fig. 1.4**. The design of the swarm centipede-like navigation robot, which is proposed in Chapter 3 of this study, will be a good example. In this research, the robots' slender and flexible body design, combined with the CWO control method, efficiently translated collisions with obstacles, other robots, and even signal noise and intervals into beneficial implicit control. This approach simplified the swarm robotic system while maintaining considerable adaptability and performance in navigation and exploration by utilizing this implicit control.


Robot-environment Interaction (Implicit control)
Environment
Guidance of the environment
Passive joints
Robot

(A) Interaction of i-CentiPot with the environment

(B) i-CentiPot in complex terrain

FIGURE 1.5: i-CentiPot

1.3.4 Current research progress and positioning of this study

Up to now, the concept of implicit control has been concretely defined at the conceptual level, with details available in [75][76]. Then, in order to explore how much of a role purely implicit control can play in robot locomotion, Osuka et al. proposed a centipede-like robot called i-CentiPot that does not carry any controller or sensors but is controlled entirely by implicit control[77][78]. As shown in **Fig. 1.5(A)**, i-CentiPot is made up of several trunks connected by passive joints without any controller. Each trunk is equipped with a pair of feet and is driven by a shaft linked to one DC motor. By using a passive and soft body, i-CentiPot can yield to the forces from the environment and automatically find an easy path to move through complex terrain, as shown in **Fig. 1.5(B)**. This research, by completely removing explicit control, fully validates the effectiveness of implicit control in robot movement.

However, so far, there has not been a systematic robot design principle as described in Section 1.3.3, that can ingeniously use robot body structure and explicit control design to induce as much implicit control as possible from the environment, while simultaneously achieving the navigation objective. Actually, in order to propose this systematic navigation robot design principle, relying solely on free and random implicit control is inadequate. Take the i-CentiPot example: it can move freely in rugged, complex environments, but its lack of explicit control hinders it from moving towards specific targets to fulfill particular navigation tasks. Thus, the central issue in establishing the aforementioned design principle lies in effectively combining implicit and explicit control to harness the strengths of both for specific navigation tasks.

Following this concept, this study introduces the "Explicit-Implicit Coordinated Robot Design Principle (E-I Coordinated Robot Design Principle)" for designing navigation robots in unknown environments. The subsequent sections will delve into the specifics of this design principle.

1.4 Explicit-Implicit Coordinated Robot Design Principle

The design philosophy of the "E-I Coordinated Robot Design Principle" is centered on ingeniously designing robot body structure and explicit control to induce as much implicit control from the environment as possible while coordinating the relationship between implicit and explicit controls to ultimately achieve navigation control. In this chapter, we will detail the specifics of the design methodology of this approach and, using several centipede-like navigation robots designed following this principle, demonstrate the practical implementation steps of the method.

1.4.1 Methodology

The core of this methodology lies in combining the strengths of both implicit and explicit control. Specifically, it involves developing a robot structure design methodology and control system design methodology that can coordinate control objectives (explicit control) with compliance with the environment (implicit control).

Robot structure design methodology

Essence: Gentle in surface, tough in spirit, and variable in rigidity.

The core essence of our robot structure design principle can be summarized in the above statement. Next, we will specifically delve into the meanings of "Gentle in surface, tough in spirit" and "Variable in rigidity" respectively:

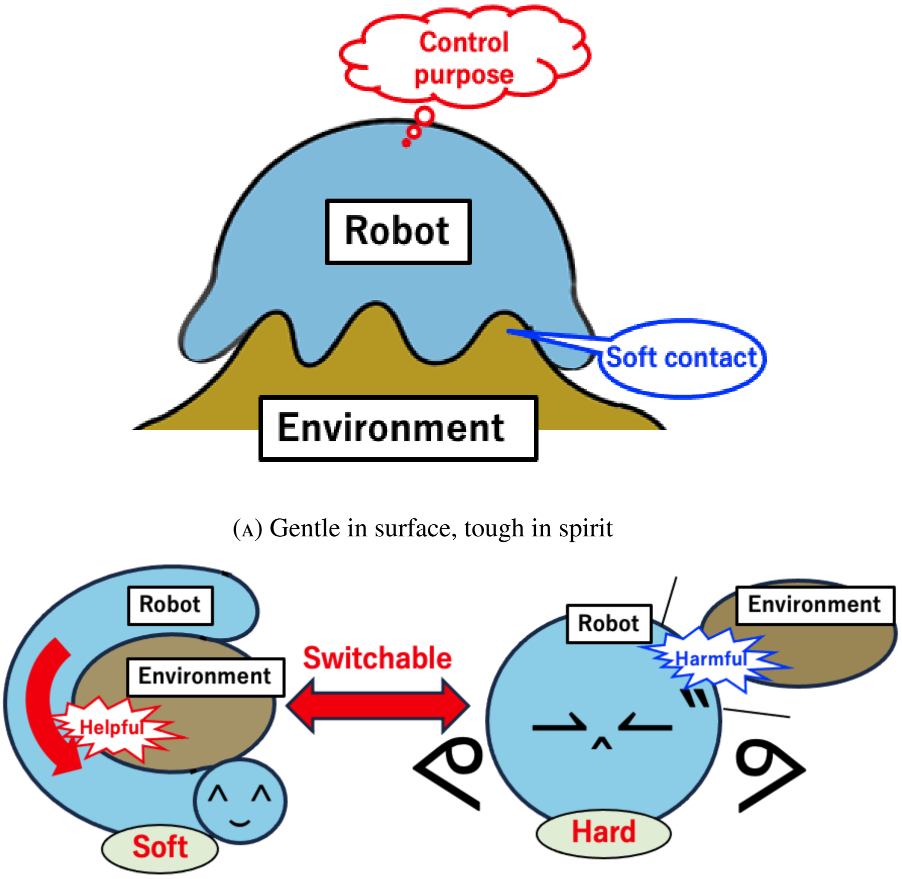
- **Gentle in surface, tough in spirit:** As shown in **Fig. 1.6(A)**, the robot we designed should, on one hand, be equipped with a firm "spirit" for achieving its control objectives, along with the necessary mobility capabilities to fulfill these objectives. On the other hand, the part of its body that interacts with the environment can be made soft, aiming to minimize confrontation with its surroundings. When necessary, it can adapt or yield to the guidance of the environment.

- **Variable in rigidity:** As shown in **Fig. 1.6(B)**, to balance both achieving control purposes and yielding to the environment, the robot's body can be designed with variable rigidity. This allows soften and yield to environmental interactions when they are favorable, and to harden and resist when the interactions are adverse. By doing so, the robot can strategically utilize environmental influences to its advantage, avoiding harm and seeking benefit.

Control system design methodology

Essence: Combining explicit and implicit control while dynamically adjusting their proportion.

In this control system design methodology, the control system is composed of two parts: the human-designed explicit control part and the environment-exerted implicit control part.



(A) Gentle in surface, tough in spirit

(B) Variable in rigidity

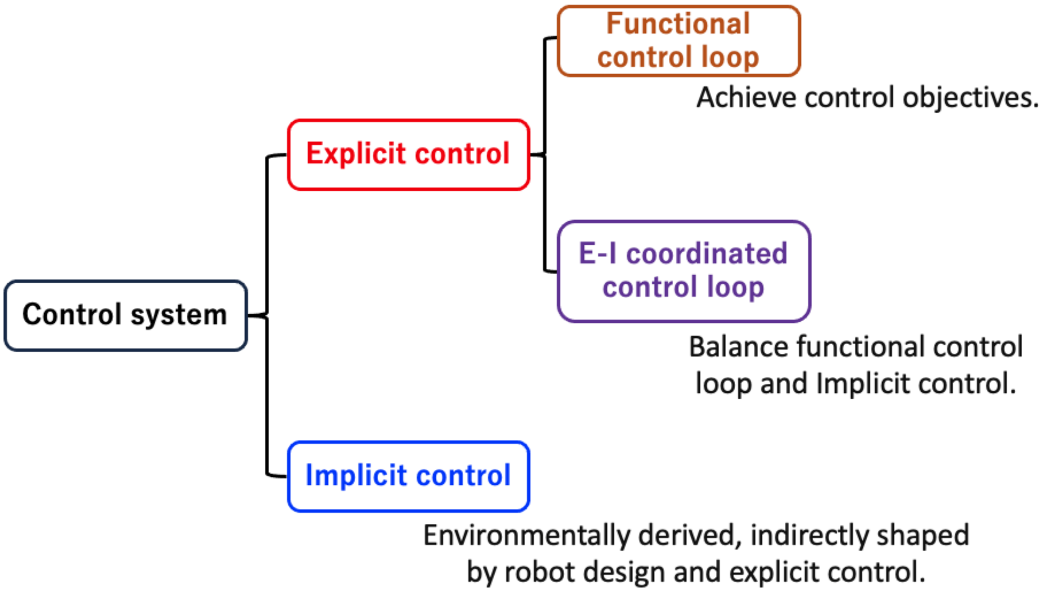
FIGURE 1.6: Robot structure design methodology

Then in the explicit control part, as illustrated in **Fig. 1.10 (B)**, besides the functional control loop tasked with achieving control objectives, there is an added independent Explicit-Implicit coordinated control loop (E-I coordinated control loop) designed to balance the proportion of functional control and yielding to the environment (implicit control).

Specifically, the functional control loop is primarily responsible for achieving control objectives. Its function is similar to traditional navigation control, such as directing the robot towards a specific target. However, due to the involvement of implicit control in this system, which undertakes many tasks, the functional control loop in this study is simpler than traditional navigation control. Usually, it does not require a complex sensor input and intricate path planning.

Regarding the E-I coordinated control loop, its main role is to coordinate the relative influence of the current functional control loop and the environmental impact (implicit control) on the system. By dynamically adjusting whether functional control or environmental influence predominates at every moment, the robot system can receive beneficial implicit control and counteract adverse disturbances while accomplishing its control objectives.

Finally, concerning implicit control, this is determined by the environment's interaction with the robot and is not directly designed by humans. However, favorable implicit control can be guided and induced by appropriate robot structural design and explicit control design as described above.


Control system
Explicit control
Functional control loop
Achieve control objectives.
E-I coordinated control loop
Balance functional control loop and Implicit control.
Implicit control
Environmentally derived, indirectly shaped by robot design and explicit control.

(A) Overview of control system

FIGURE 1.7: Control system design methodology

1.4.2 Design example

In Section 1.4.1, we outlined the design methodology of the “E-I Coordinated Robot Design Principle”. In this section, we use several centipede-like navigation robots, designed based on the aforementioned principles, as examples to demonstrate how our proposed robot design method is applied in practice. The detailed descriptions of these robots will be elaborated in Chapters 2, 3, and 4.

Robot structure design: centipede-like robot

Inspired by the centipede-like robot i-CentiPot introduced in 1.3.4, which is operated with pure implicit control, we designed the basic model of the centipede-like navigation robot as shown in as illustrated in **Fig. 1.8**. Subsequent designs of centipede robots for various unknown navigation environments and tasks are all based on improvements to this initial model.

As illustrated in **Fig. 1.8**, the design of the centipede-like robot’s body has the potential to meet the principles of being “gentle in surface, tough in spirit” and “variable in rigidity”. Firstly, the robot can achieve the “firm spirit” that navigates towards the goal by controlling the angle of certain trunk joints or the rotating speed of the legs, thereby achieving “tough in spirit”. At the same time, the robot can also become “gentle in surface” by designing some

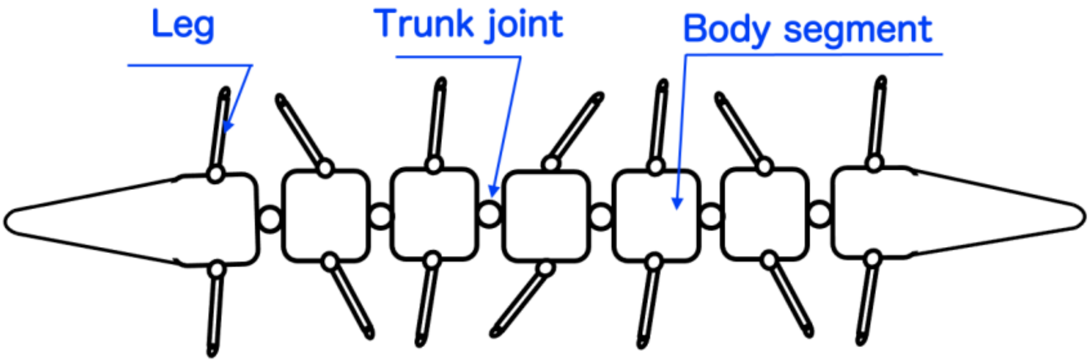
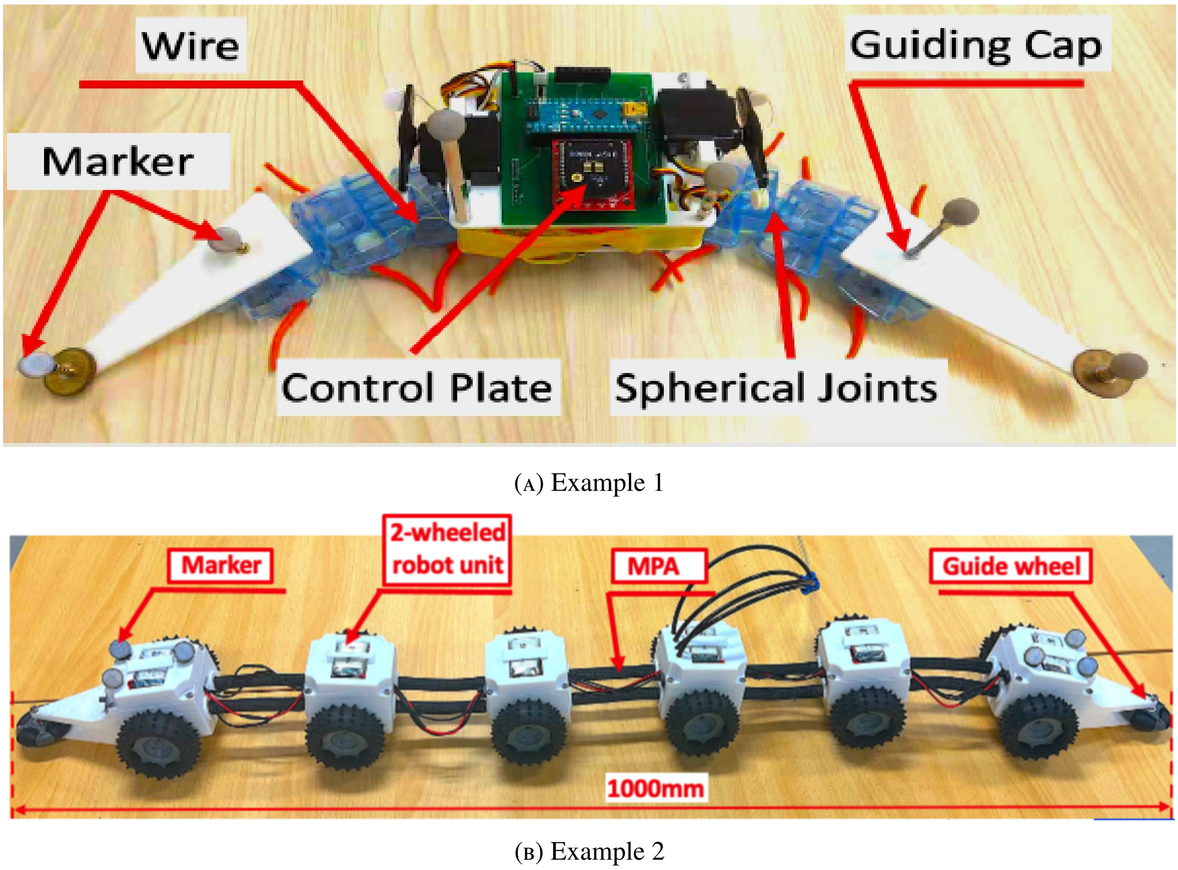


FIGURE 1.8: Basic model of the centipede-like navigation robot



(A) Example 1

(B) Example 2

FIGURE 1.9: Examples of the centipede-like navigation robot

trunk joints to be passively soft, thus allowing parts of the body to adapt to environmental influences. By integrating these two design aspects, the robot can navigate to its target adeptly while conforming to environmental influences, achieving “gentle in surface, tough in spirit”. Additionally, the trunk joints can be designed with variable stiffness, such as using materials with variable rigidity to connect different body segments, allowing for changes in the body’s softness, embodying “variable in rigidity”. **Fig. 1.9** show various centipede-like navigation robots proposed in this study for different navigation tasks and environments.

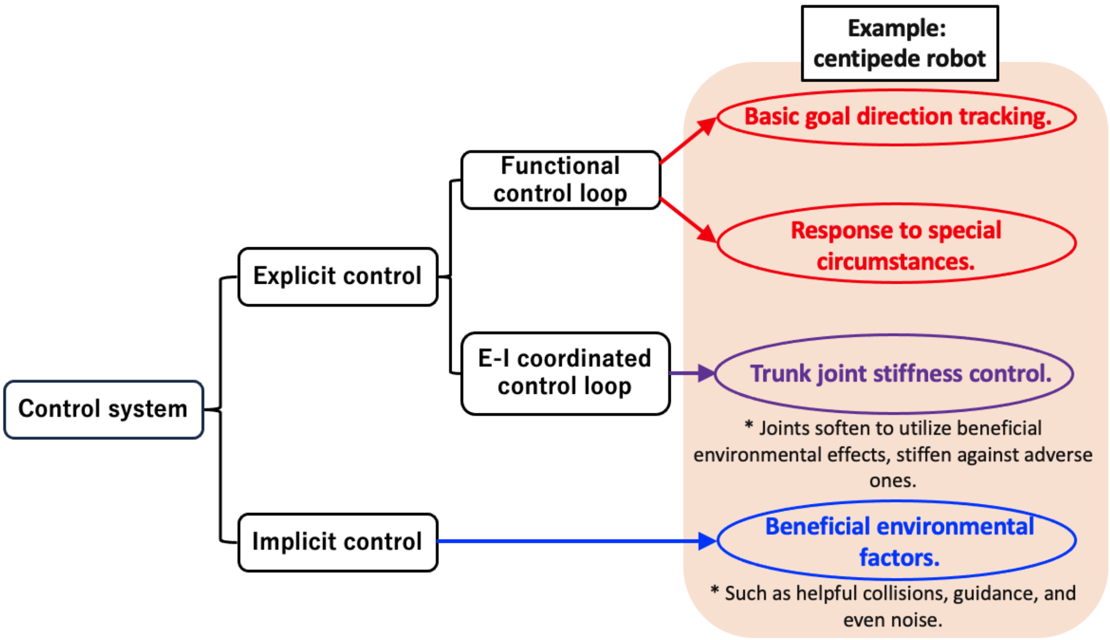

Example:
centipede robot
Control system
Explicit control
Implicit control
Functional control loop
E-I coordinated control loop
Basic goal direction tracking.
Response to special circumstances.
Trunk joint stiffness control.
* Joints soften to utilize beneficial environmental effects, stiffen against adverse ones.
Beneficial environmental factors.
* Such as helpful collisions, guidance, and even noise.

FIGURE 1.10: Control system design of centipede-like robot

Control system design of centipede-like robot

In the design example of the centipede-like robot, the composition of its control system is as shown in **Fig. 1.10**.

The functional control loop primarily includes basic goal direction tracking and responses to certain special circumstances, which is quite simple. Besides, the E-I coordinated control loop is composed of the stiffness control loop of the trunk joints. The general control principle of this stiffness control loop can be summarized as follows: when environmental influences are favorable, the trunk joints soften to accept this guidance; when the influences are unfavorable, the joints harden to resist. It's important to note that not all centipede-like navigation robots proposed in our study require dynamic stiffness control loops. In fact, for simpler scenarios, like navigation in a flat 2D environment, a fixed stiffness configuration of the centipede segments suffices to induce enough implicit control to achieve the navigation mission. Finally, the implicit control part of the centipede robot varies depending on each specific robot and its navigation environment. It is composed of one or more environmental factors, such as beneficial environmental collisions, guidance, and others even like observation signal noise. These factors are induced by the aforementioned robot structure and explicit control and are determined and generated by the environment itself.

Based on these characteristics of the centipede-like robots, this study has proposed several robots adapted to different types of unknown environments[82][83][84][85]. For detailed information on these robots, please refer to Chapters 2 to 4.

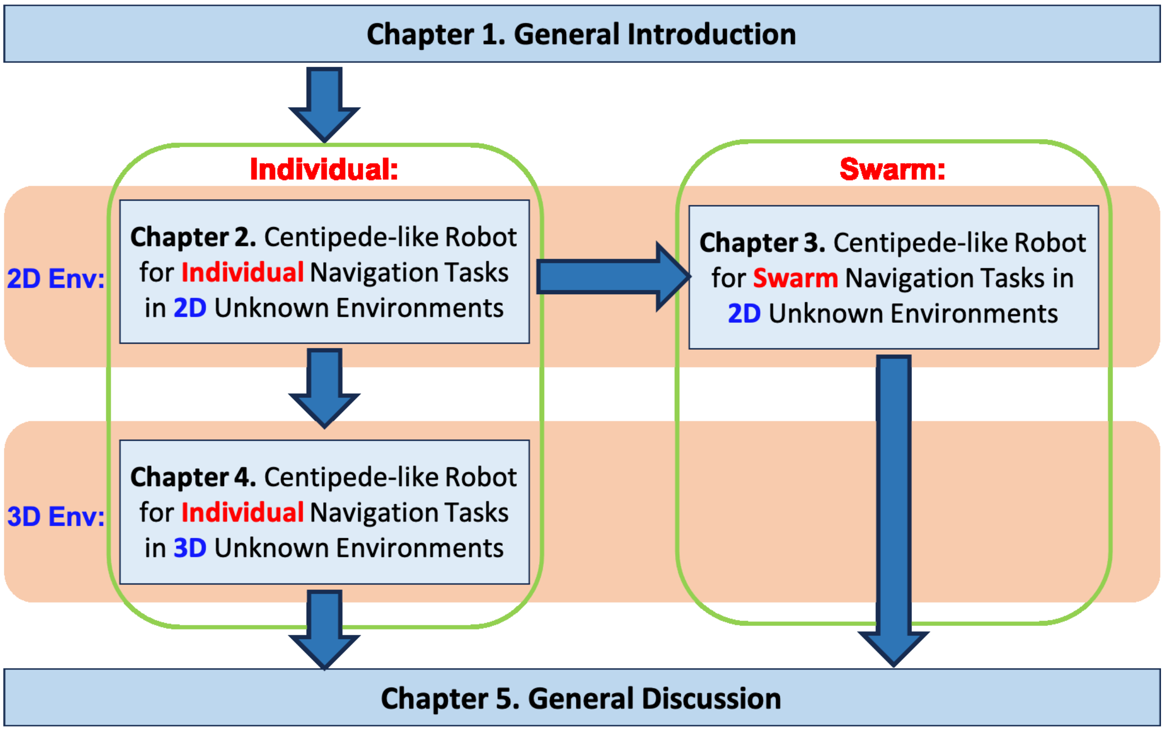

Chapter 1. General Introduction
Individual:
Swarm:
2D Env:
3D Env:
Chapter 2. Centipede-like Robot for Individual Navigation Tasks in 2D Unknown Environments
Chapter 3. Centipede-like Robot for Swarm Navigation Tasks in 2D Unknown Environments
Chapter 4. Centipede-like Robot for Individual Navigation Tasks in 3D Unknown Environments
Chapter 5. General Discussion

FIGURE 1.11: Composition of the thesis

1.5 Composition of the Thesis

The structure of this thesis is illustrated in **Fig. 1.11**. Following this general introduction chapter, the navigation problem is categorized into three types based on two dimensions: the type of unknown environment (2D vs 3D) and the number of navigation robots (individual vs swarm). Corresponding centipede-like navigation robots are proposed for each category. Specifically:

- Chapter 2 introduces individual centipede-like navigation robots designed for 2D unknown environments composed of flat ground and wall obstacles, along with their experimental validation[82].
- Chapter 3 shifts focus to swarm navigation problems, discussing the development and simulation validation of swarm centipede-like navigation robots in simulated 2D unknown environments[83].
- Chapter 4 tackles 3D unknown environments characterized by mountains, flat ground, and wall obstacles. It presents the development, simulation, and experimental validation of individual centipede robots for these 3D environments[84][85].
- Chapter 5 summarizes the content of the study, highlighting future directions and potential developments in this research area.

Chapter 2

Centipede-like Robot for Individual Navigation Tasks in 2D Unknown Environments[82]

2.1 Introduction

First, based on the "E-I Coordinated Robot Design Principle," we designed a centipede-like navigation robot for individual robot navigation tasks in 2D unknown environments, along with a corresponding navigation control method named the "Control without Overdoing (CWO)" scheme. In this scheme, the robot takes implicit control for most of the obstacle coping function and explicit control for only the goal tracking function and response function for some special terrains, as shown in **Fig. 2.1**. Consequently, by coordinating the use of implicit and explicit controls, the navigation robots proposed in this study do not require perceiving information about the surrounding environment. They only need the goal direction information as the sole observational input to navigate in 2D unknown environments with a very simple robot system. Based on this idea, we built a robot model and control model for the "Control without Overdoing" scheme and built a prototype of the proposed centipede robot. Then we conducted navigation experiments with this prototype in 3 environments with different levels of complexity and analyzed the characteristics and applicability scenarios of the proposed navigation scheme compared to the traditional ones.

It is worth mentioning that due to the relatively simple composition of 2D unknown environments, it is not necessary to deliberately change the body's stiffness. Simply relying on clever robot body structural design and a fixed distribution of body stiffness suffices to coordinate the relationship between implicit and explicit controls. Therefore, in 2D environments, we did not specifically design the E-I coordinated control loop, namely the stiffness control loop.

This research is structured as follows. In Section 2.2, we present the problem statement. Then, we present the robot model of the proposed robot in Section 2.3. In Section 2.4, we present the control model of the proposed scheme. Then we present the making of the robot prototype and the verification experiment of this research in Section 2.5. Finally, in Section 2.6 we present the conclusion.

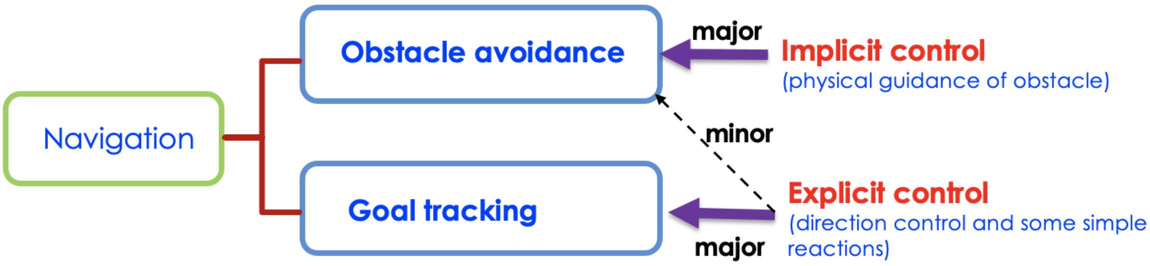


FIGURE 2.1: “Control without Overdoing” scheme

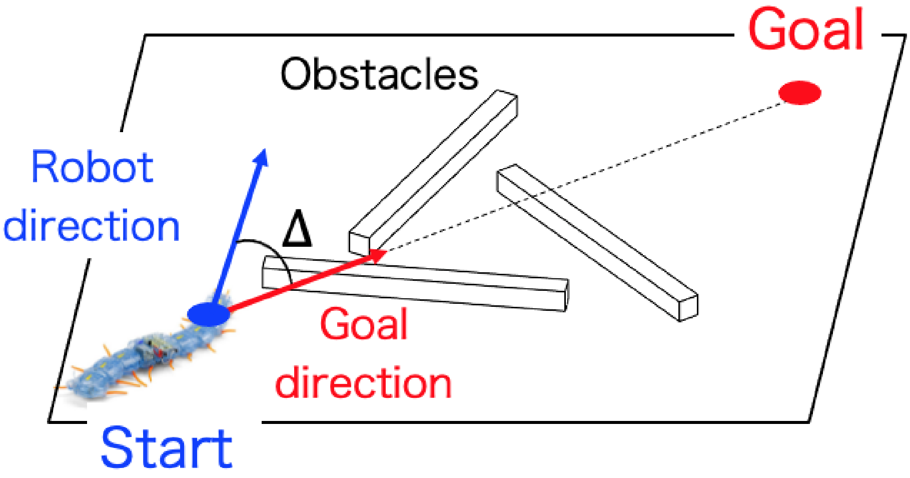


FIGURE 2.2: Problem Statement

2.2 Problem Statement

In this study, we place the centipede robot at the starting point in an unknown 2D obstacle environment, and the navigation goal is placed at a location distinct from the start point, as shown in **Fig. 2.2**. Between the starting point and the goal, a set of rectangular obstacles unknown to the robot is placed. Wherein, the obstacle height is higher than or equal to the height of the robot, and there is objectively at least one pathway with a width greater than the width of the robot between the starting point and the goal. Among them, the direction angle $\Delta(t) \in \mathbb{R}$ of the goal direction relative to the robot's forward direction is the only observation information of the centipede robot, which can theoretically be obtained by a variety of sensor schemes such as GPS, acoustic waves, and electromagnetic waves. In this study, we use an external motion capture system to obtain this angle, the details of which are described in detail in Chapter 4. The navigation can be considered successful if the centipede robot can successfully travel through the obstacle area to reach the goal with $\Delta(t)$ as the only input. It is important to note that the purpose of this study is not to propose a navigation scheme that goes beyond existing methods such as the SLAM-based scheme, but to demonstrate that “navigation can still be accomplished with minimum explicit control and sensing by utilizing implicit control”. Thus, we do not need to compare the performance of the traditional and

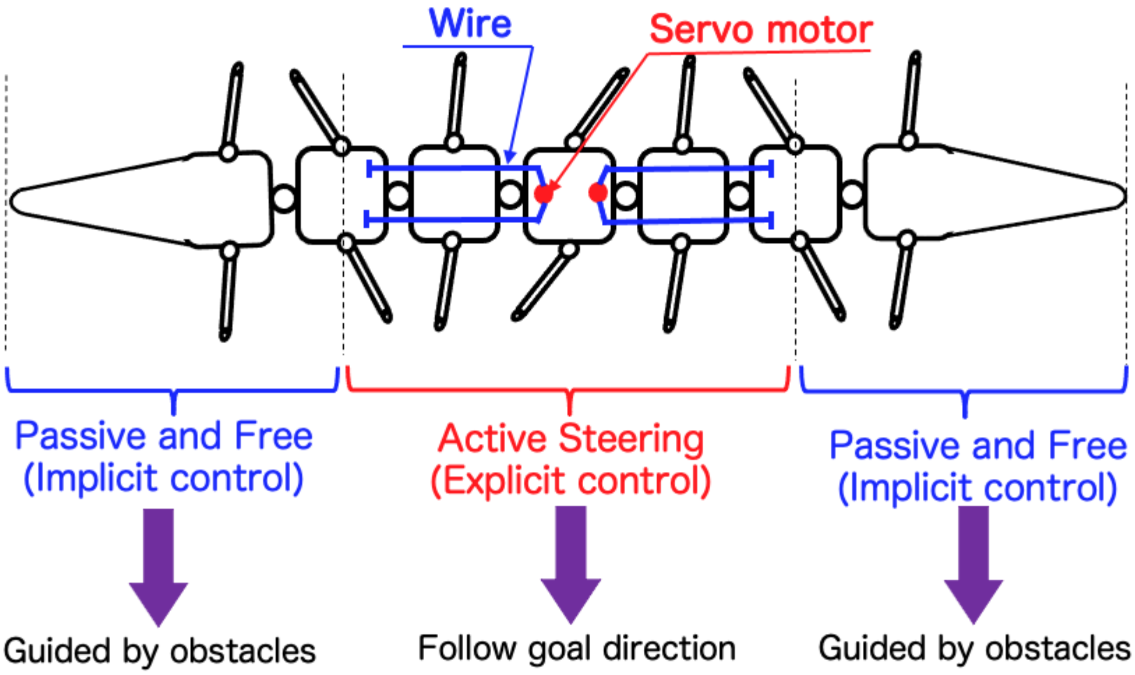


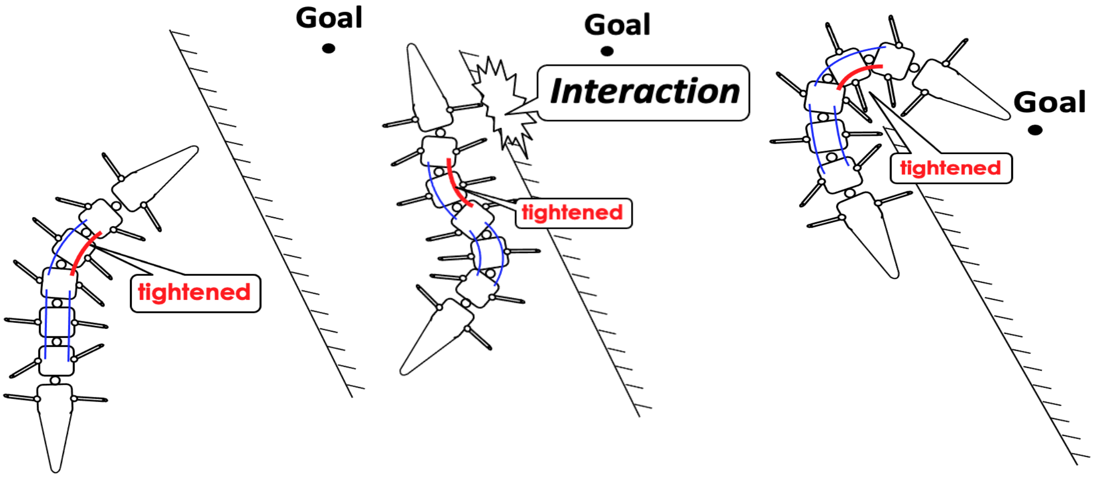
FIGURE 2.3: Robot model

proposed navigation schemes. As long as navigation using the proposed scheme in this study can be successful, the purpose of this study has been achieved.

2.3 Robot Model

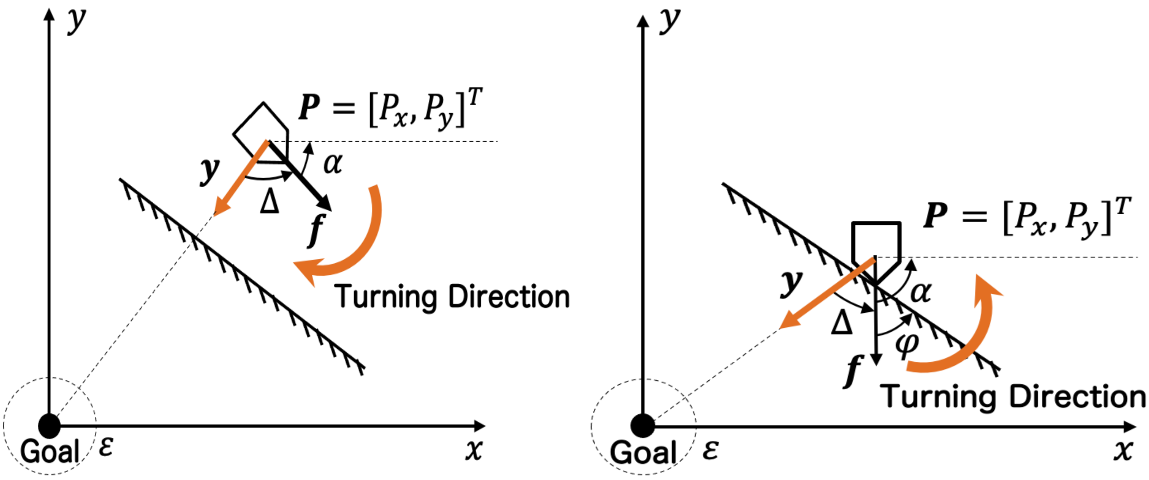
To realize the function division of explicit control and implicit control mentioned in Section 2.1, we designed the robot model as shown in **Fig. 2.3**. This model is inspired by the structure of “i-CentiPot” proposed by Prof. Osuka. Based on its structure, an active steering system driven by a servo motor and wire is added. The robot model can be divided into three parts: the active steering part in the middle and the passive guiding part at both ends of the robot. The middle steering part is controlled by two servo motors pulling the wires located on both sides of the robot. When the wire on one side is tightened, the robot’s torso bends to that side, and the robot turns in the corresponding direction. When the wires on both sides are relaxed, the robot returns to a free passive guidance state. On the other hand, at both ends of the robot, because the wires do not extend to these joints, these joints are always passive and free and can be guided by implicit control generated from the environment.

Unlike the wheeled robot model which can be viewed as a point rigid body, the centipede robot proposed in this study has more mechanical flexibility because it has a long and slender body consisting of many mobile units connected by passive joints. Therefore, when its head collides with an obstacle, it can naturally bend and move along it until it bypasses the obstacle. Specifically, when there is no obstacle, the robot is mainly affected by the active steering in the middle part and is controlled to move to the goal by explicit control, as shown in **Fig. 2.4(A)**. When an obstacle is encountered, as shown in **Fig. 2.4(B)**, the interaction



(A) Move to the goal (B) Yeild to obstacles (C) Bypass obstacles

FIGURE 2.4: Working principle



(A) Not encountering the obstacle (B) Encountering the obstacle

FIGURE 2.5: Control process demonstration

between the passive robot head and the obstacle produces a strong implicit control, which makes the explicit control of the robot succumb to the implicit control and make the robot move along the edge of the obstacle. Then, when the obstacle is bypassed, the explicit control occupies the dominant position again, as shown in **Fig. 2.4(C)**.

In this way, local navigation including obstacle avoidance and goal tracking is completed. With this robot model, the robot can complete the switching of implicit control and explicit control unconsciously without perception and judgment. This kind of unconscious switching has also become the basis of the whole “Control without Overdoing” navigation scheme.

2.4 Control Model

Based on the robot model mentioned in the previous section and a control model for an obstacle-free environment proposed in our previous research[80][81], for which a brief introduction is illustrated in Appendix A, we propose the control model of the "Control without Overdoing" navigation scheme for an unknown environment in this section. To facilitate mathematical modeling, in this section, we abstract the robot model as a directional point. The position and the direction of this point represent the position of the center of gravity and the forward direction of the robot respectively. The motion characteristics of the point are consistent with the actual centipede robot model.

2.4.1 Basic Control Model

"Control without Overdoing" can be interpreted as: controlling the robot to move towards the goal, but never over controlling. The basic control logic can be described as follows:

- Follow goal direction and move to it by simple P control.
- Yield to obstacles by the passive head until bypassing them.

Based on the control logic, we propose the following control model. As shown in **Fig. 2.5**, we set up a coordinate system with the goal as the origin $[0, 0]^T$. $t \in \mathbb{R}_+$ represents the time. The position of the robot is represented by the vector $\boldsymbol{P}(t)$, which can be defined as $\boldsymbol{P}(t) = [P_x(t), P_y(t)]^T$. Vector $\boldsymbol{f}(t) \in \mathbb{R}^2$ is a unit vector representing the direction of the robot and vector $\boldsymbol{y}(t) \in \mathbb{R}^2$ is a unit vector pointing to the origin. $\Delta(t)$ is the angle between $\boldsymbol{f}(t)$ and $\boldsymbol{y}(t)$, which is the only observation value of this system. Besides, $\alpha(t) \in \mathbb{R}$ is the angle between $\boldsymbol{f}(t)$ and horizontal direction, and $\varphi(t) \in \mathbb{R}$ is the angle between $\boldsymbol{f}(t)$ and the tangent line of the obstacle. This control model can be described as:

$$\begin{cases} \dot{\boldsymbol{P}} = \boldsymbol{A}(\Delta + \delta_1\varphi)v\boldsymbol{y}\cos(\delta_1\varphi), \\ \dot{\alpha} = -K_1\Delta - \delta_1 K_2\varphi. \end{cases} \tag{2.1}$$

In this equation, $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ represents the rotation matrix in the plane, which can be expressed as：

$$\boldsymbol{A}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.2}$$

And $v \in \mathbb{R}_+$ is the magnitude of robot velocity. "$-K_1\Delta$" is the explicit control part which is decided by the designer when designing the controller. And $K_1 \in \mathbb{R}_+$ is the gain coefficient of this controller. Correspondingly, "$-\delta_1 K_2\varphi$" is the implicit control part passively generated when the robot contacts with the environment. In other words, when the robot contacts an obstacle, it is naturally affected by the obstacle. And to describe this effect from a control perspective, we write it as "$-\delta_1 K_2\varphi$" to describe this physical effect. Here, δ_1 is the condition coefficient which is passively generated and can be defined by

$$\delta_1 = \begin{cases} 0 \text{ (if there is no obstacle)}, \\ 1 \text{ (if contacted with obstacle)}. \end{cases} \tag{2.3}$$

Meanwhile, $K_2 \in \mathbb{R}_+$ is the coefficient that is passively determined by the physical contact factors between the robot and the environment. In order to realize "Control without Overdoing", K_1 and K_2 should meet the condition:

$$K_1 << K_2. \tag{2.4}$$

As a result, when the robot doesn't encounter obstacles, as shown in **Fig. 2.5(A)**, δ_1 is 0, and the explicit control part dominates, robots are controlled to turn to the direction of the goal and move forward. But when the robot encounters obstacles, as shown in **Fig. 2.5(B)**, the δ_1 turns to 1 and the explicit control part yields to the implicit control part because $K_1 << K_2$. Until the resistance disappears, the explicit control takes the dominant position again. In this way, robots can be navigated to reach the goal gradually.

2.4.2 Response to Special Terrain

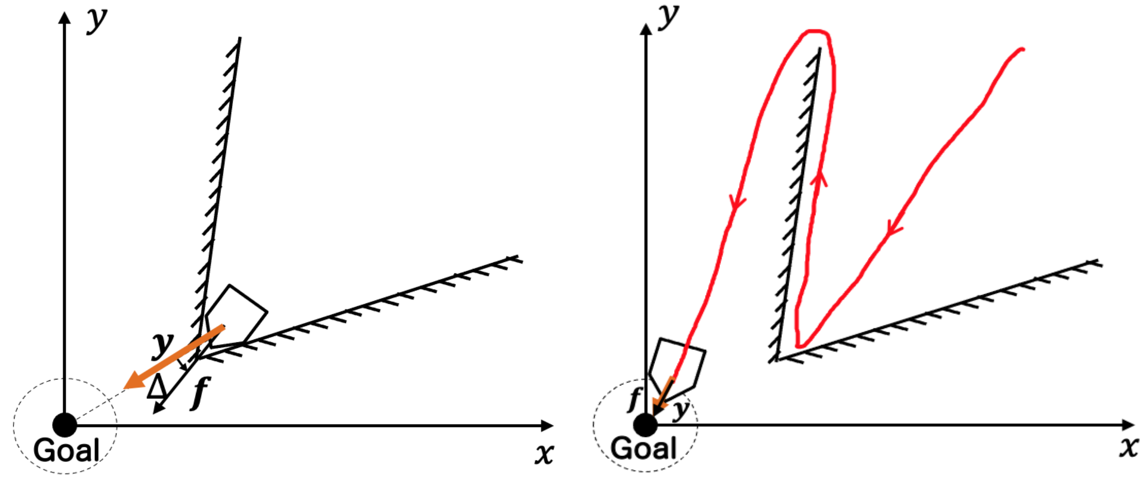
Although the navigation tasks can be achieved in most simple environments with the above basic control model, we find that this basic control model sometimes can not achieve desired navigation goals with the increase in the complexity of the environment. After simple computer simulation, we find that the majority of task failures can be divided into two categories: stuck in a corner and trapped in an infinite loop. In this section, we put forward "Reversing Behavior" and "Random Direction Behavior" respectively in order to solve these two problems.

Instead of adding additional sensor inputs, these two behaviors successfully solve the above problems by utilizing the goal direction relative to the robot's forward direction(Δ) as the only observation input.

"Stuck in Corner" Problem and "Reversing Behavior"

Sometimes the robot might be stuck when it encounters a sharp corner, as shown in **Fig. 2.6(A)**. At this time, the "Reversing Behavior" is activated. The contents of "Reversing Behavior" can be described as following: when the robot is judged to be stuck, the tail of the robot becomes the head of the robot, while the original head becomes the tail, and the robot moves in a reverse direction, as shown in **Fig. 2.6(B)**.

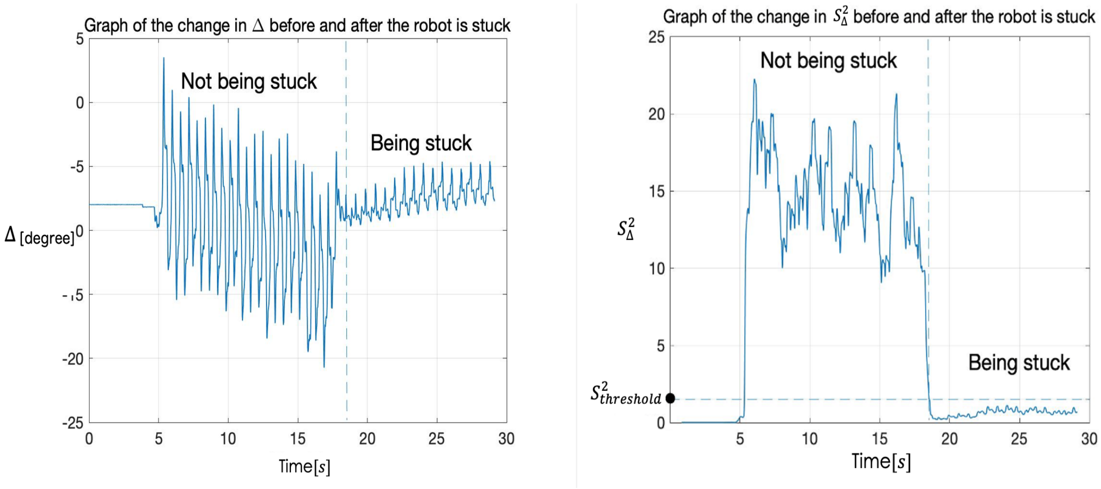
And in terms of the judgement of "stuck" situation, we propose a method that determines whether the robot is stuck by only detecting variance $S_{\Delta}^2(t) \in \mathbb{R}_+$ of Δ over a time interval $[t - T_{interval}, t]$, where $T_{interval} \in \mathbb{R}_+$ is a constant time period. We define the number of Δ derived in time period $[t - T_{interval}, t]$ as $n_{interval} \in \mathbb{R}_+$, then the average value $\bar{\Delta} \in \mathbb{R}$ of Δ over this time period can be defined as:



(A) Stuck situation

(B) Reversing Behavior

FIGURE 2.6: Stuck situation and "Reversing Behavior"



(A) Change of Δ when "stuck" happened

(B) Change of S_Δ^2 when "stuck" happened

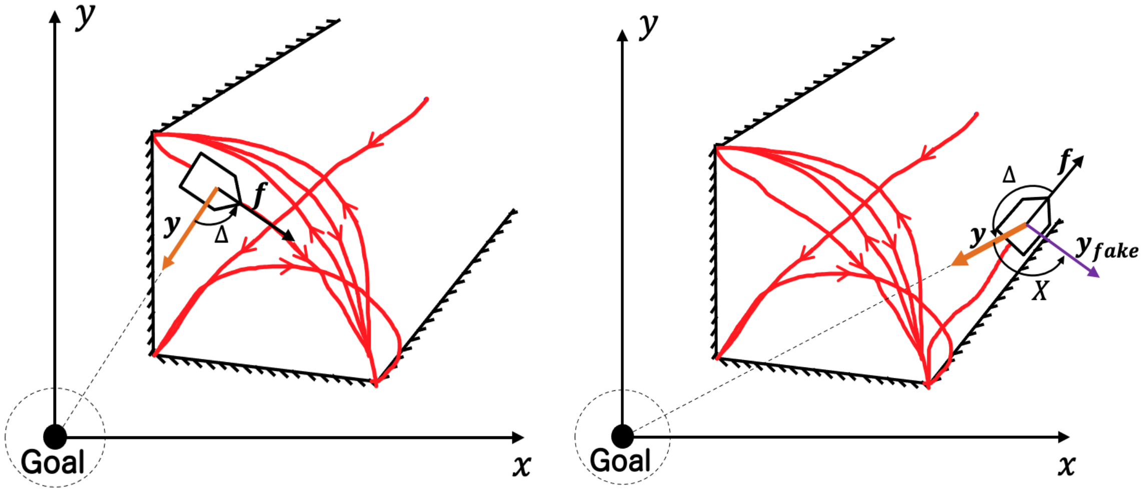
FIGURE 2.7: Detection of "stuck" situation

$$\bar{\Delta} = \frac{1}{n_{interval}} \sum_{i=1}^{n_{interval}} \Delta_i \tag{2.5}$$

The variance S_Δ^2 of Δ over this time period can be defined as:

$$S_\Delta^2 = \frac{1}{n_{interval} - 1} \sum_{i=1}^{n_{interval}} (\Delta_i - \bar{\Delta})^2 \tag{2.6}$$

According to the research of Kinugasa et al., the centipede robot which is made up of passive joints moves in an undulation pattern[78]. And in our previous experiments, we also found that the Δ of the centipede robot fluctuates during its motion. And when the robot



(A) Infinite loop (B) Random Direction Behavior

FIGURE 2.8: Infinite loop and "Random Direction Behavior"

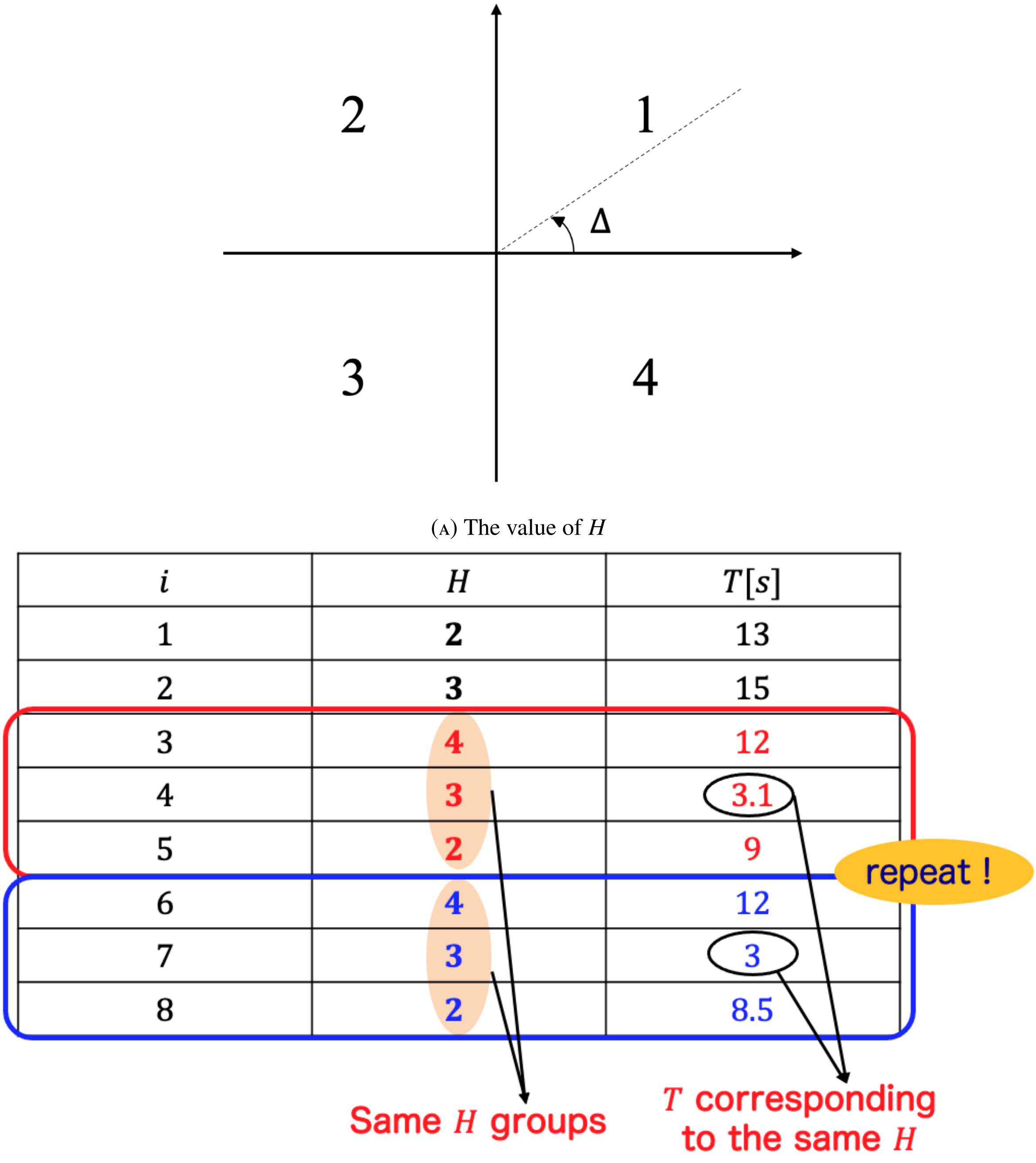
is "stuck" in a corner, the magnitude of this fluctuation decreases significantly compared to when the robot is not stuck. Therefore, we can determine whether the robot is stuck by detecting whether the variance S_{Δ}^{2} over the past time interval $[t - T_{interval}, t]$ is less than a certain critical value $S_{threshold}^{2} \in \mathbb{R}_{+}$ at each moment t. The determination condition can be expressed as follows:

- If $S_{\Delta}^{2} \leq S_{threshold}^{2}$, then the robot is judged to be stuck.
- If $S_{\Delta}^{2} > S_{threshold}^{2}$, then the robot is judgedd not to be stuck.

Here, in order to obtain the value of $S_{threshold}^{2}$, we conducted a preparatory experiment using the prototype made in this study. We put the robot equipped with only the " basic control model " without "Reversing Behavior" in a corner environment similar to the one shown in **Fig. 2.6(A)**, and recorded the change of Δ before and after the robot was stuck, as shown in **Fig. 2.7(A)**. Then we calculated the corresponding S_{Δ}^{2} for each instant using Eq. (2.5) and Eq. (2.6) according to the values of $T_{interval} = 0.83$s and $n_{interval} = 100$, and the results are shown in **Fig. 2.7(B)**. By analyzing the change of S_{Δ}^{2} before and after being stuck, we set $S_{threshold}^{2}$ to 1.5 and used it in the subsequent experiments.

"Infinite Loop" Problem and "Random Direction Behavior"

In the process of navigation, the robot may encounter a situation where it repeats the same path and movements in an area so that it cannot get out from there. We describe this situation as "trapped in an infinite loop", as shown in **Fig. 2.8(A)**. At this time, the "Random Direction Behavior" is activated. The contents of "Random Direction Behavior" is: when the robot is judged to be trapped into an "infinite loop", the robot moves randomly towards a virtual direction $\boldsymbol{y}_{fake} \in \mathbb{R}^{2}$ for time $t_{keep} = T_{phase} \in \mathbb{R}_{+}$. Here, $T_{phase} \in \mathbb{R}_{+}$ is the amount of time the robot has been trapped in "infinite loop". The calculation method of T_{phase} is explained in detail later.



(A) The value of H

i	H	$T[s]$
1	**2**	13
2	**3**	15
3	**4**	12
4	**3**	3.1
5	**2**	9
6	**4**	12
7	**3**	3
8	**2**	8.5

(B) Direction history table

FIGURE 2.9: The judgement of infinite loop

Then, in order to judge whether the robot is trapped in an "infinite loop", we first define a discrete quantity $H \in \mathbb{R}_+$ that represents the quadrant of Δ as shown in **Fig. 2.9(A)**. The value rules of H can be expressed as follows:

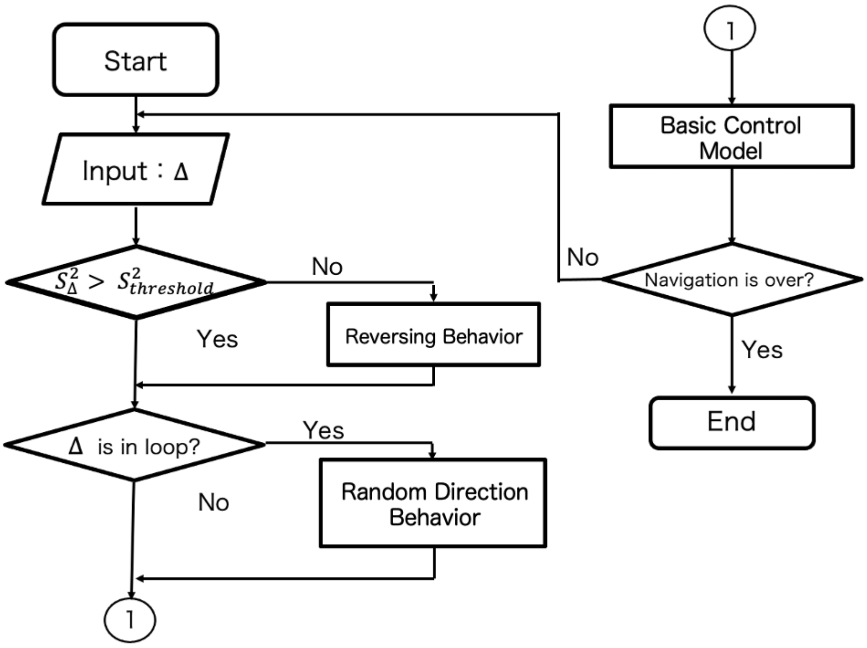


FIGURE 2.10: Control flow chart

$$H = \begin{cases} 1\ (0 < \Delta \le \frac{\pi}{2}), \\ 2\ (\frac{\pi}{2} < \Delta \le \pi), \\ 3\ (-\pi < \Delta \le -\frac{\pi}{2}), \\ 4\ (-\frac{\pi}{2} < \Delta \le 0). \end{cases} \tag{2.7}$$

Then the robot records this quadrant H and the time $T \in \mathbb{R}_+$ the robot spent in each quadrant into a table, as shown in **Fig. 2.9(B)**. The first column of the table is the time sequence number $i \in \mathbb{R}_+$ of H and T. The larger the time sequence number, the newer H and T are recorded. The second column of the table represents the series of H that the robot has experienced during the robot's navigation, and the third column indicates the time T that the robot has experienced in each quadrant H. Then, the robot judges whether the latest generated series of H and T satisfies:

- The same H group appears 2 times at the bottom of the table.
- In the H group that occurs 2 times, the values of T corresponding to the same H in 2 groups differ by no more than 1s.

If the above conditions are satisfied, then we can judge that the robot is caught in an "infinite loop". And we can get the time T_{phase}, the amount of time the robot has been trapped in the "infinite loop", by accumulating all the values of T in the two repeated groups.

Control Model with New Behaviors

In summary, the control model together with these two new behaviors can be described as follows:

- Follow goal direction and move to it by simple Proportional control.
- Yield to obstacles by the passive head until bypassing them.
- When stuck in a corner, move in a reverse direction.
- When trapped in an infinite loop, go in a random direction until getting out of the loop.

The control flow chart of it can be shown in the **Fig. 2.10**.

Then, we add these new behaviors to the original mathematical model. We define the time from robot being judged as trapped in an infinite loop as $t_{from\ last\ trap} \in \mathbb{R}_+$. The new model can be expressed as:

$$\begin{cases} \dot{\boldsymbol{P}} = \boldsymbol{A}(\Delta + \delta_1\varphi + \delta_{2,n}\pi + \delta_3 X)v\boldsymbol{y}\cos(\delta_1\varphi), \\ \dot{\alpha} = -K_1(\Delta + \delta_{2,n}\pi + \delta_3 X) - \delta_1 K_2\varphi. \end{cases} \tag{2.8}$$

In this equation, X is a random variable with uniform distribution, which can be defined by

$$X \sim U(-\pi, \pi) \tag{2.9}$$

δ_2 is the condition coefficient of “Reverse Behavior” which can be defined by

$$\begin{aligned} \delta_{2,0} &= 0, \\ \delta_{2,n} &= \begin{cases} \delta_{2,n-1} + 1 \text{ (if stuck)}, \\ \delta_{2,n-1} \text{ (if not stuck)}. \end{cases} \end{aligned} \tag{2.10}$$

Here, n represents the number of times the robot has been stuck from the start.

And δ_3 is the condition coefficient of “Random Direction Behavior” which can be defined by

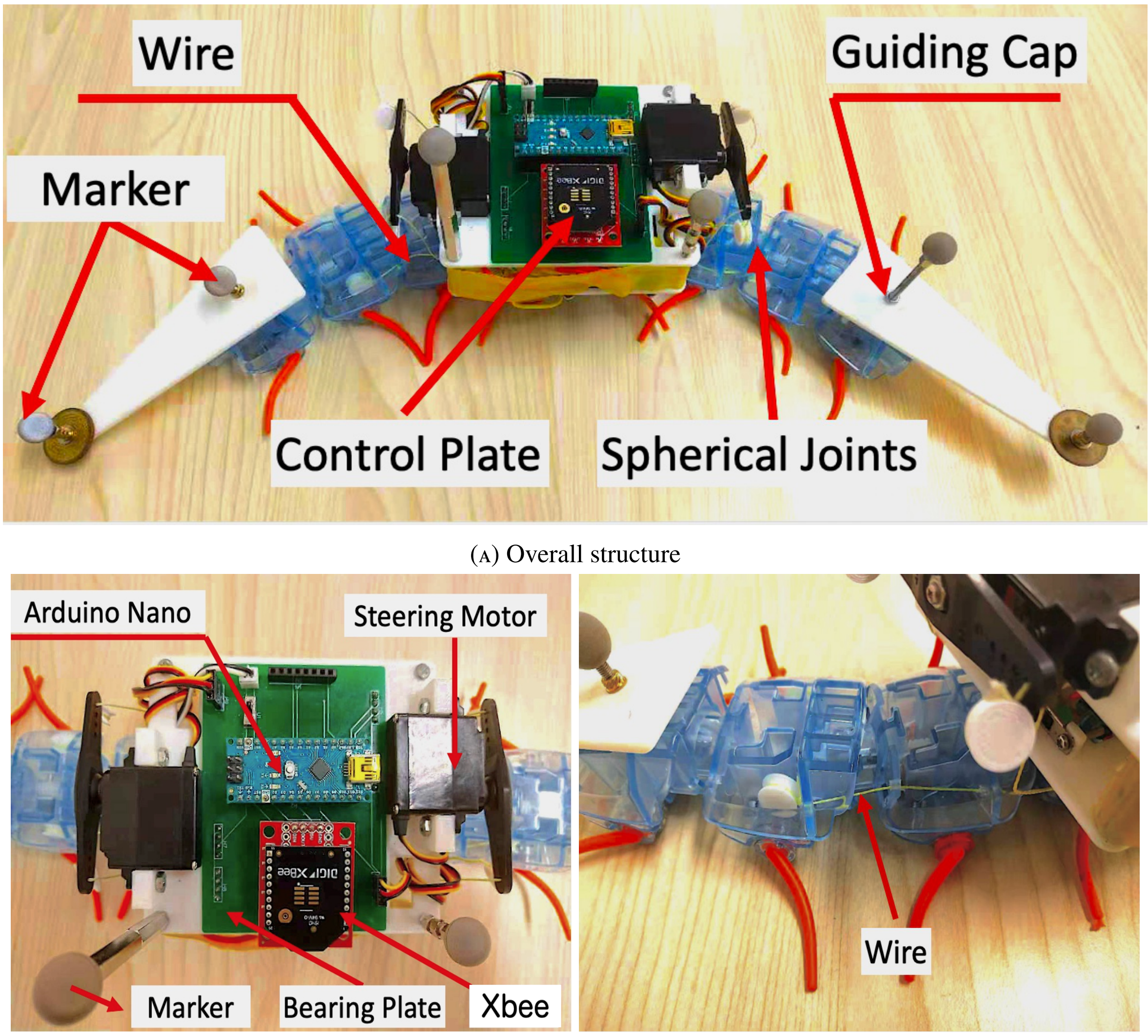
$$\delta_3 = \begin{cases} 1 \ (t_{from\ last\ trap} < t_{keep}), \\ 0 \ (t_{from\ last\ trap} \geq t_{keep}). \end{cases} \tag{2.11}$$

2.5 Experiment with Prototype

2.5.1 Prototype Making

Based on the robot model and control model, we designed and made a prototype robot.

As shown in **Fig. 2.11(A)**, we employ TAMIYA parts (Fun Craft Series No. 230, TAMIYA)to make most of the robot torso and use 3D printing parts to make the controller bearing plate and guiding cap. In the aspect of controller, as shown in **Fig. 2.11(B)**, the prototype utilizes Arduino (Arduino Nano, Arduino) as the core controller, which controls the rotation angle of two servo motors (32085S HS-85MG Metal Gear Servo, Hitec) and the rotation direction of a DC motor (Power Dash Motor, TAMIYA). Then, the two servo motors pull the wires (X-Core Fishing Line, Kurosawa) located on both sides of the robot to control the robot steering, as shown in **Fig. 2.11(C)**. As for the sensor system, the prototype only installs the



(A) Overall structure

(B) Control plate

(C) Wire

FIGURE 2.11: Prototype

Xbee (XBee3 ZigBee 3.0, Digi) wireless communication module to obtain the goal direction relative to the robot Δ from the motion capture system (V120: Trio, OptiTrack). The specifications of the prototype is shown in **Table 2.1**.

2.5.2 Experiment System

In this section, we designed an experiment system as shown in **Fig. 2.12**. As shown in the **Fig. 2.12**, in a plane experiment field of 1.8m in length and 1.3m in width, the goal was taken as the origin $[0,0]^T$ to establish a plane coordinate system. The starting point of the robot was set as $[1.4,1.25]^T$. In the space from the starting point to the goal, obstacles built by square steel with different complexity were placed. Constrained by the size of the robot prototype, the following conditions need to be met for the experiment environment setting:

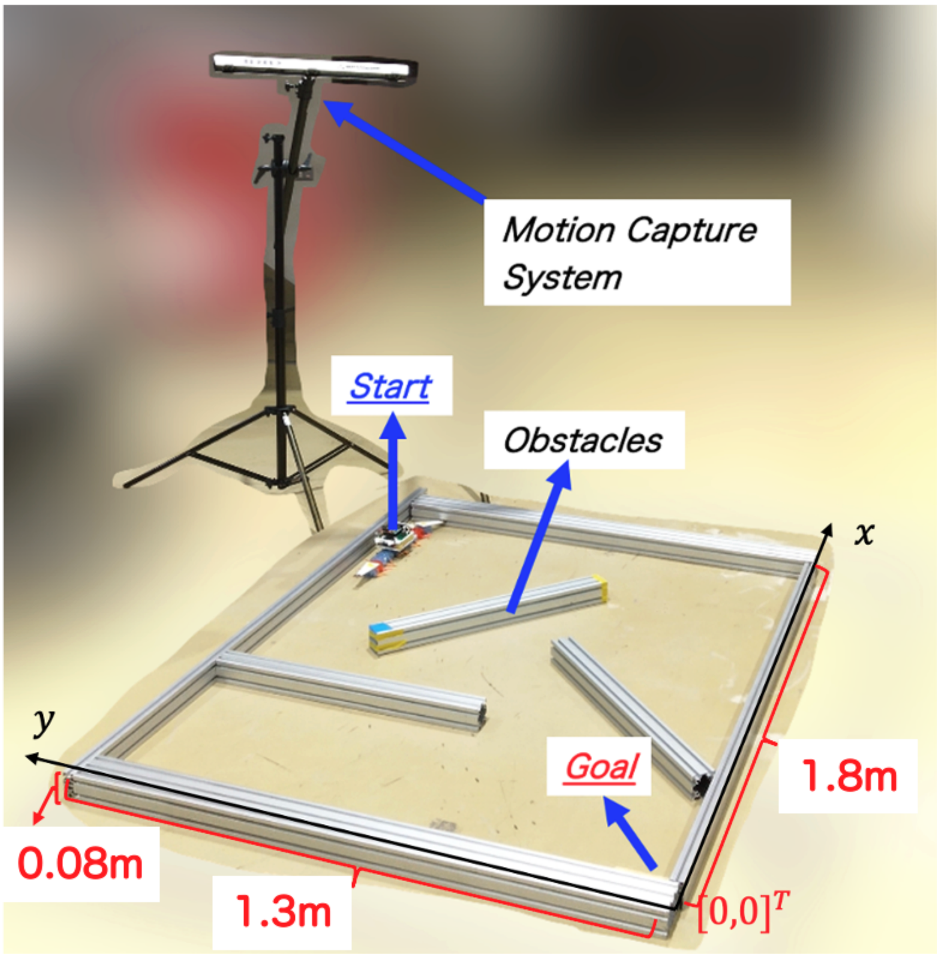


FIGURE 2.12: Experiment system

- There exists at least one pathway connecting the starting point to the end point with a width greater than 85mm, which is the width of the prototype.
- The environment is on a flat surface with obstacles, and the obstacle height is higher than or equal to the height of the robot, which is set as 80mm in this experiment.

As shown in the **Fig. 2.12**, robots obtained the goal direction information Δ from a motion capture system (V120: Trio, OptiTrack) placed on one side and moved at a constant speed of $v = 0.065$m/s which is not so fast that explicit control cannot respond, where v is the magnitude of robot velocity in Eq. (2.1) and Eq. (2.8). It should be emphasized that the motion capture system is not necessary for this navigation scheme, and any sensing system which can obtain the goal direction relative to the robot (Δ) is feasible. Since the design of a specific sensing scheme is not the focus of this study, the motion capture system was chosen as the way to obtain Δ for convenience in this experiment.

TABLE 2.1: Specifications of the prototype

Length	470 mm
Width	85 mm
Height	80 mm
Weight	440 g

2.5.3 Experiment Contents

In order to verify whether the proposed robot can navigate successfully with “Control without Overdoing” scheme in different environments, we conducted experiments in the following three different environments. First, we set up a simple “One Obstacle” environment to verify the effect of the implicit control in “basic control model”. Then, to verify the effect of “Reversing Behavior” when the robot is stuck in corners, we set up the “Maze with Corners” environment. And last, we set up “ Maze with a Trap” environment to verify the effect of “Random Direction Behavior” when the robot is trapped in an infinite loop. In each environment we conducted 10 experiments with the same initial conditions to record the navigation route map and the navigation time distribution. And in all trials, the robot started from the starting point with the initial direction toward the negative direction of the x-axis, and the navigation can be considered successful when the robot arrives within 0.3m of the goal.

2.5.4 Results of Experiment

Environment 1: One Obstacle

Firstly, as shown in [0sec] part of **Fig. 2.13**, “One Obstacle” environment consists of a simple obstacle built with a rectangular steel part of 0.7m in length in the center of the experimental field. **Figure 2.13** shows an example of the navigation process in this environment. In this trial, the robot used only the basic control model and started to apply the implicit control when it encountered the obstacle at 14s until the robot was out of contact with the obstacle at 28s. During the contact between the obstacle and the robot’s body, especially its head and legs, the orientation of the robot gradually changes under the influence of the implicit control, which eventually makes the robot successfully bypass the obstacle. Then, **Fig. 2.14(A)** shows the navigation route map for 10 trials of environment 1. As shown in the graph, the robot’s routes were generally consistent across the 10 trials, although there were minor differences. Finally, **Fig. 2.14(B)** shows the distribution of the navigation time for the 10 trials. Analyzing the graph, we found that the navigation time was distributed as a unimodal and the average navigation time of 10 trails was 32.14s with an extreme difference of 6.86s. Analyzing **Fig. 2.14(A)** and **Fig. 2.14(B)**, we noticed that in the “one obstacle” environment, the robot’s navigation performance was generally stable, with less variability and uncertainty.

Environment 2: “Maze” with Corners

Then, we increased the complexity of the environment and set up “Maze with corners” environment built with rectangular steel parts of 0.7m, 0.6m and 0.4m in length in the experimental field, as shown in [0sec] part of **Fig. 2.15**. Here, there are not only simple rectangular obstacles but also “corners” where the robot might be stuck. **Figure 2.15** shows an example of the navigation process in this environment. In this trial, while using the basic control model with implicit control, the robot also activated the “Reversing Behavior” when it encountered the “corner” at 29s. After that, we made the navigation route map for 10 trials in this environment as shown in **Fig. 2.16(A)**. As shown in the figure, the robot’s routes in Environment 2 were more diverse than those in Environment 1, and there were some distinctive navigation routes different from others. Finally, as shown in **Fig. 2.16(B)**, the distribution graph of the navigation time for the 10 trials was made. The navigation time in this graph was distributed as a multimodal and the average navigation time of the 10 trails was 79.39s with an extreme difference of 50.54s. In summary, in the “Maze with Corners” environment, the robot’s navigation has more variability and uncertainty compared to environment 1.

Environment 3: “Maze” with a Trap

Finally, with the increasement of the complexity of the environment, we conducted navigation experiments in a “Maze with a Trap” environment where there were both rectangular obstacles, “corners” and “infinite loop trap”, as shown in [0sec] part of **Fig. 2.17**. Here, obstacles are built with rectangular steel parts of 0.7m and 0.6m in length in the experimental field. **Figure 2.17** shows an example of the navigation process in this environment. In this trial, while using the basic control model with implicit control and “Reversing Behavior”, the robot also applied the “Random Direction Behavior”, which started at 44s and ended at 65s, when it was trapped in an “infinite loop” shown between 22s and 44s. Next, in the navigation route map for 10 trials shown in **Fig. 2.18(A)**, the robot’s routes in Environment 3 were more diverse than those in Environment 1 and 2, so that the navigation routes were different almost every time. At last, **Fig. 2.18(B)** shows the distribution of the navigation time for the 10 trials. Here, the navigation time was distributed as a multimodal and the average navigation time of 10 trails was 138.23s with an extreme difference of 80.09s. In general, the robot’s navigation has the greatest variability and uncertainty in the “Maze with a trap” environment compared to environments 1 and 2.

2.5.5 Discussion

By analyzing the route maps and time distribution maps, we find that the “ Control without Overdoing ” scheme has the following 3 characteristics, compared to traditional methods.

Firstly, compared to conventional navigation robots, the robot in this study is simple and inexpensive because it navigates with only minimal explicit control and perception by using the interaction between the soft centipede robot body and the environment. However, although the robot can guarantee a considerable navigation success rate by making a gradual detour to the goal as long as a path exists, the route chosen by this scheme is often not the optimal solution. In other words, the “ Control without Overdoing ” scheme may not be suitable for some tasks requiring high navigation efficiency. Correspondingly, it can be used in 2D unknown environment navigation scenarios with cost and computational power constraints, but with less stringent requirements for efficiency. Moreover, it can also be combined with traditional navigation methods, as a backup when sensor systems such as radar are damaged or as a supplement for environmental mechanics information.

Secondly, even under the same conditions, the route and time of each experiment are different, with considerable “uncertainty” and “irreducibility”. This “uncertainty” of the proposed scheme is sometimes an advantage and sometimes a disadvantage in practical navigation applications compared to traditional methods with more certain routes. For example, when a robot is required to perform an unknown environment exploration task, the “uncertainty” of this approach is beneficial and can bring a greater breadth of exploration. Whereas, in some object transportation tasks that emphasize trajectory repeatability, the route uncertainty of this method can be harmful. And how to make this “ uncertainty ” of this method play a unique role in suitable application scenarios might be an interesting research direction in the future.

Thirdly, the “uncertainty” of the navigation is strongly correlated with the complexity of the environment. By analyzing the experimental results, we found that when the environmental complexity is low, the navigation trajectory and time distribution are relatively concentrated, while as the environmental complexity increases, the navigation trajectory and time distribution become more and more dispersed, and the “uncertainty” and “irreducibility” increase. This is mainly because the implicit control fed by the environment can vary greatly depending on the small changes in the robot’s actions, and thus the more complex the environment, the greater the variation in navigation results accumulated by the implicit control.

Besides, we also consider some possible effects of some parameters of the robot on the navigation results. First, regarding the noise of the robot’s direction perception, we consider that its impact on the navigation results may vary depending on the type of noise. For example, the Gaussian noise might have less impact on the navigation results because it keeps oscillating randomly on both sides of 0. In other words, the mutual offset of the noise and the constant correction of the feedback control will weaken the effect of the noise. However, the bias noise, on the other hand, might cause a large deviation in the navigation results because it causes a fixed shift in the perceived direction. Second, regarding the effect of robot size on navigation, we expect that different sizes of robots are adapted to different navigation environments. When the robot size is smaller than the obstacle size or comparable, the robot may perform the “Control without Overdoing” navigation scheme well. And when the robot size is much larger than the obstacle size, the obstacle can be crossed by the robot. This part is not considered in this study about 2D environment and is planned to be studied in detail in the future. Finally, regarding the effect of robot speed on navigation, if the robot speed is too fast, active control and perception may produce a large delay, which might make the robot produce problems such as untimely turns or even loss of control. And when the speed is small, the navigation efficiency can be greatly reduced. Therefore, how to choose the appropriate travel speed according to the different environments and tasks will be an important research direction for us in the future.

2.6 Conclusions

In this study, as the first step to explore how to make full use of “ implicit control ” with minimum “ explicit control ” and sensing to accomplish the unknown environment navigation, we proposed a centipede robot and a corresponding navigation scheme called “Control without Overdoing” scheme to navigate an unknown 2D obstacle environment without perceiving information of the surrounding environment, but with the goal direction information as the only observation input. Then, we built a prototype robot and conducted navigation experiments in 3 environments with different levels of complexity. As a result, we obtained the navigation route map and navigation time distribution of each environment, and analyzed the characteristics and applicability scenarios of the proposed scheme compared to the traditional ones.

In future, optimizing the proposed navigation scheme by adding speed control and an onboard goal direction sensing system to the robot with corresponding noise processing algorithms will be a potential research topic. Or, extending the proposed scheme to practical 3D rugged mountain scenarios by changing the robot’s size, structure and algorithms is also a possible research theme.

Acknowledgements
This research was supported in part by grants-in-aid for JSPS KAKENHI (Grant-in-Aid for Scientific Research (S), Grant Number JP17H06150), Japan.

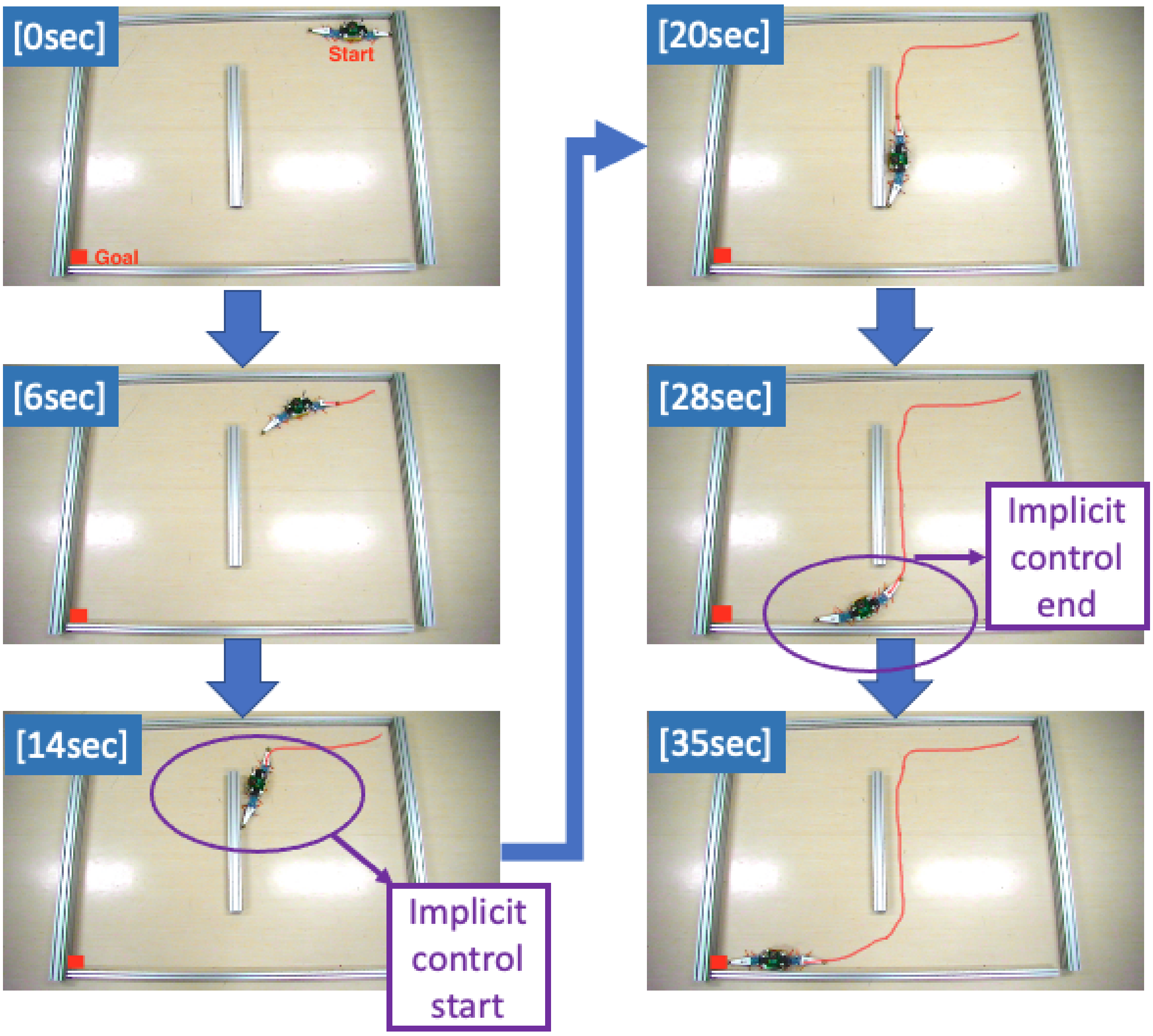
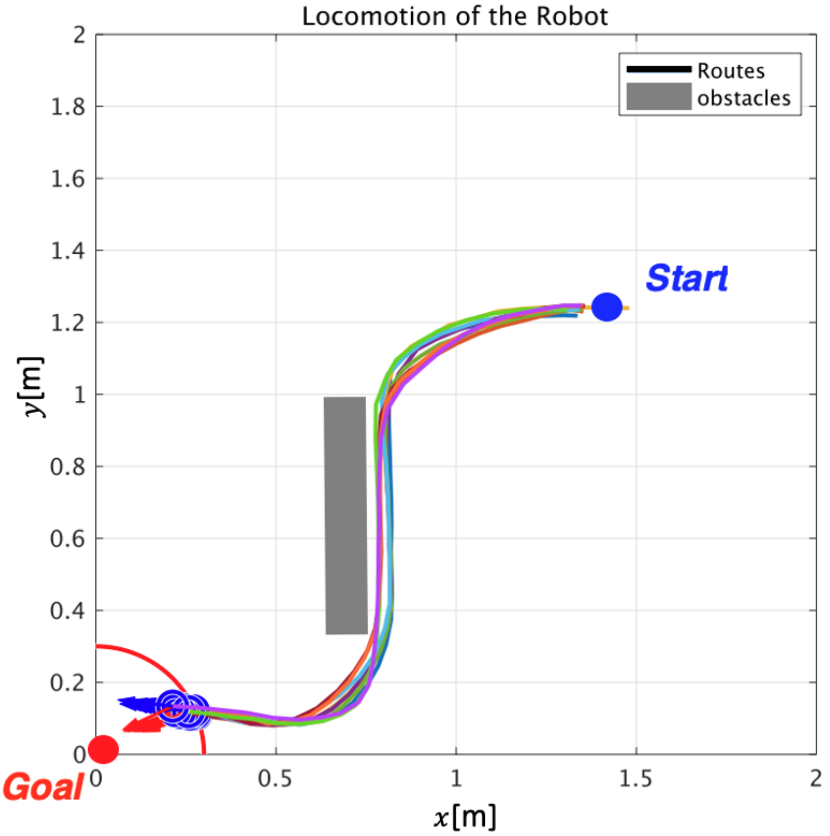
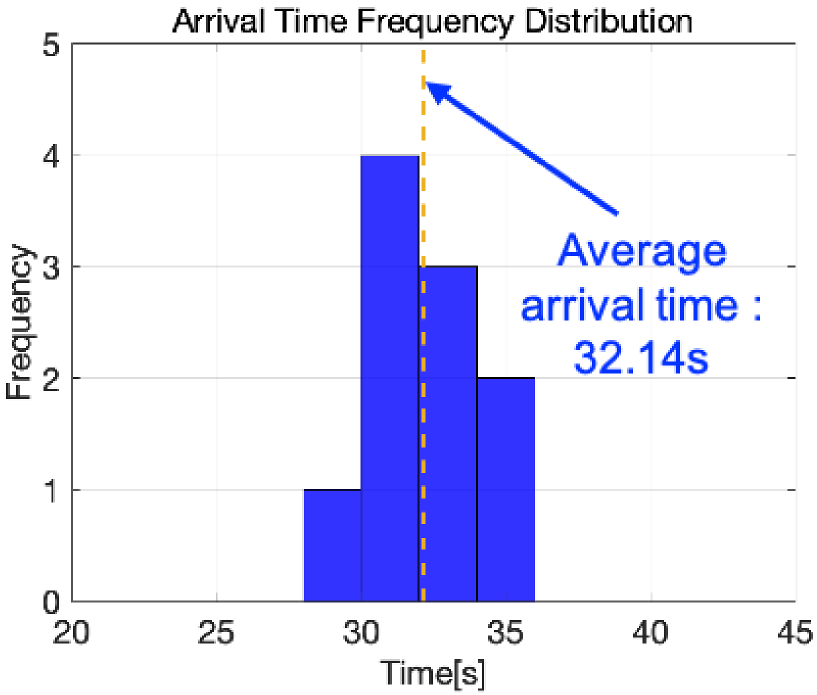


Figure 2.13: Environment 1: one obstacle



(A) Traveling route



(B) Time distribution

FIGURE 2.14: Route and time distribution for environment 1

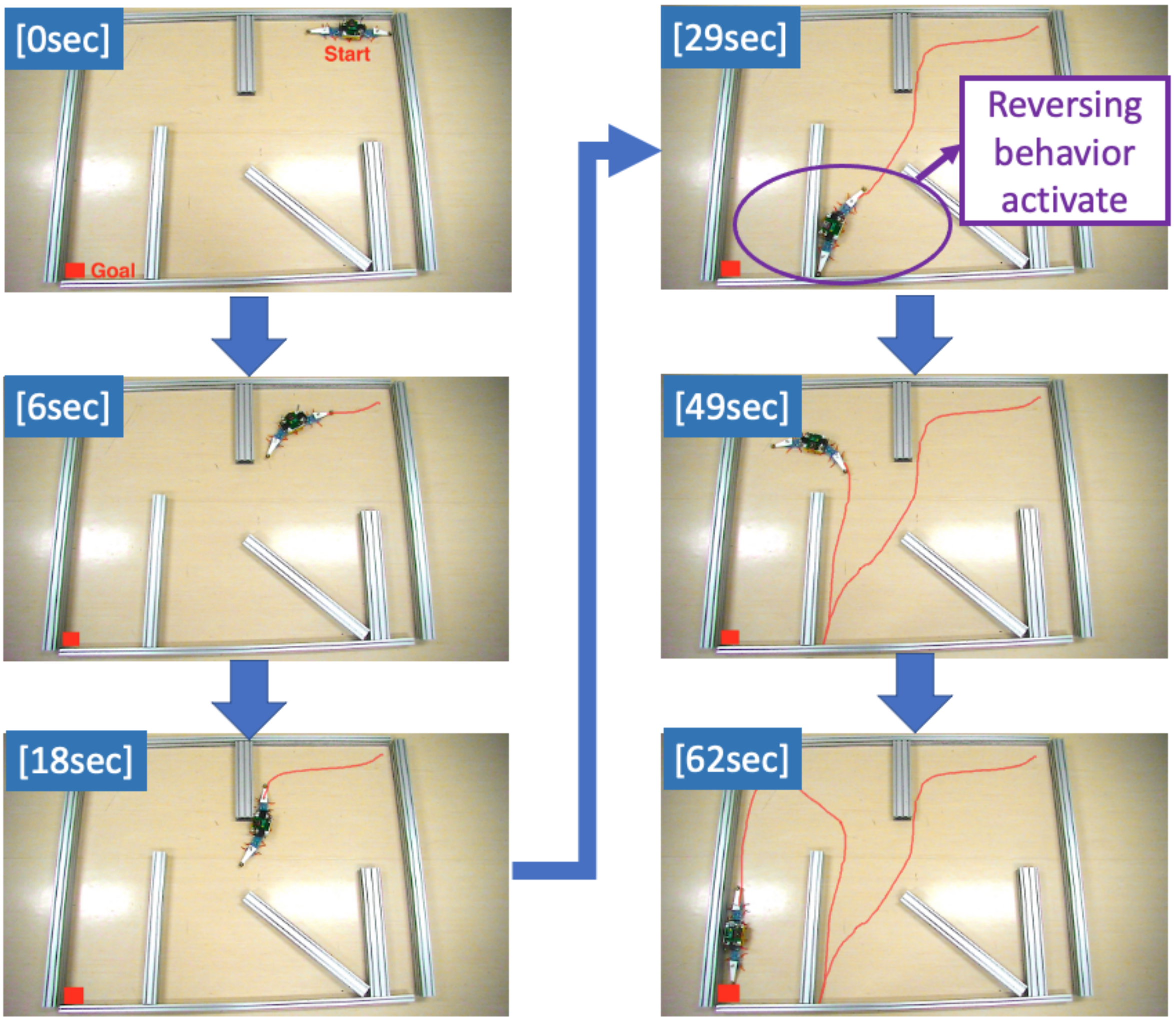
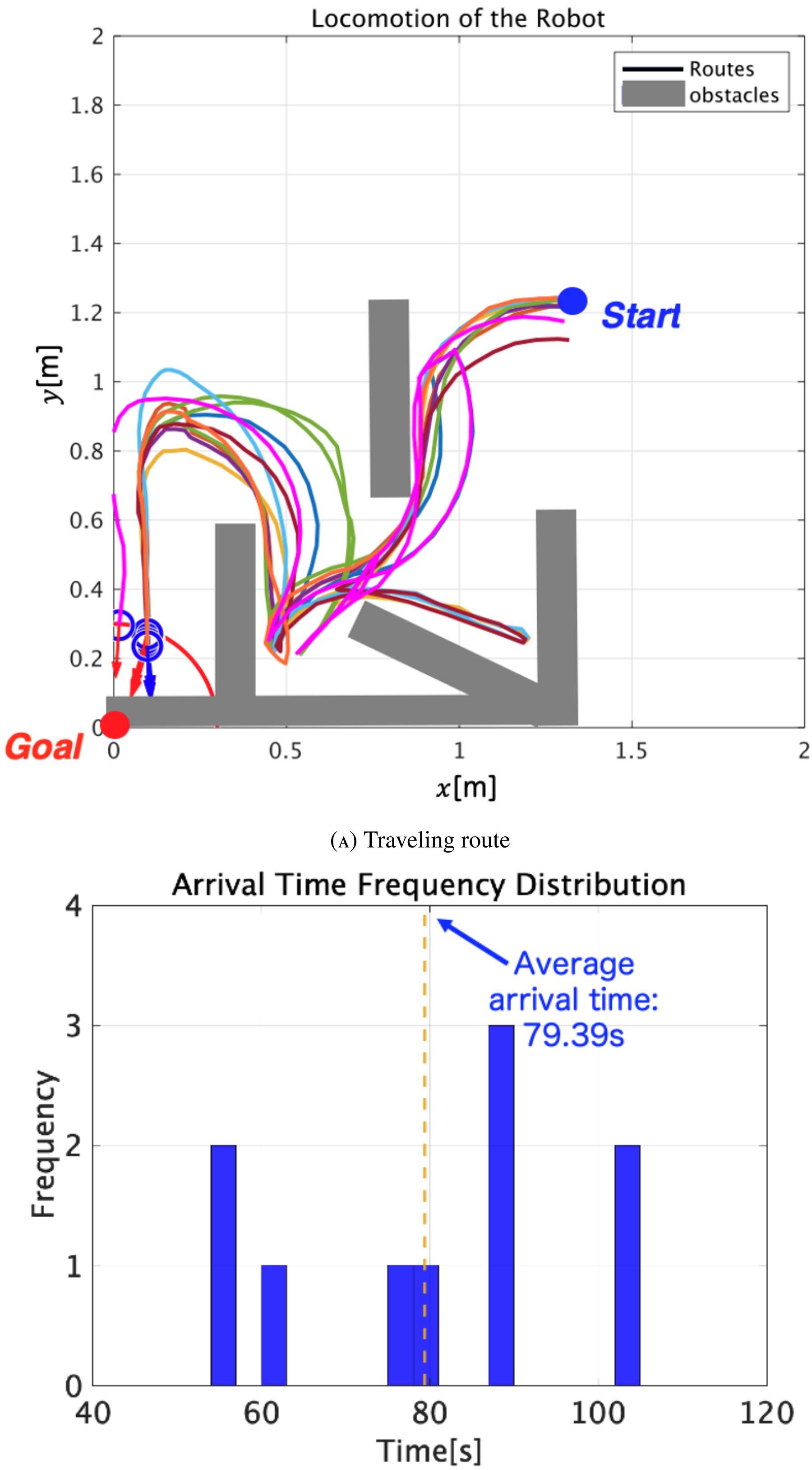


FIGURE 2.15: Environment 2: “maze” with corners



(A) Traveling route

(B) Time distribution

FIGURE 2.16: Route and time distribution for environment 2

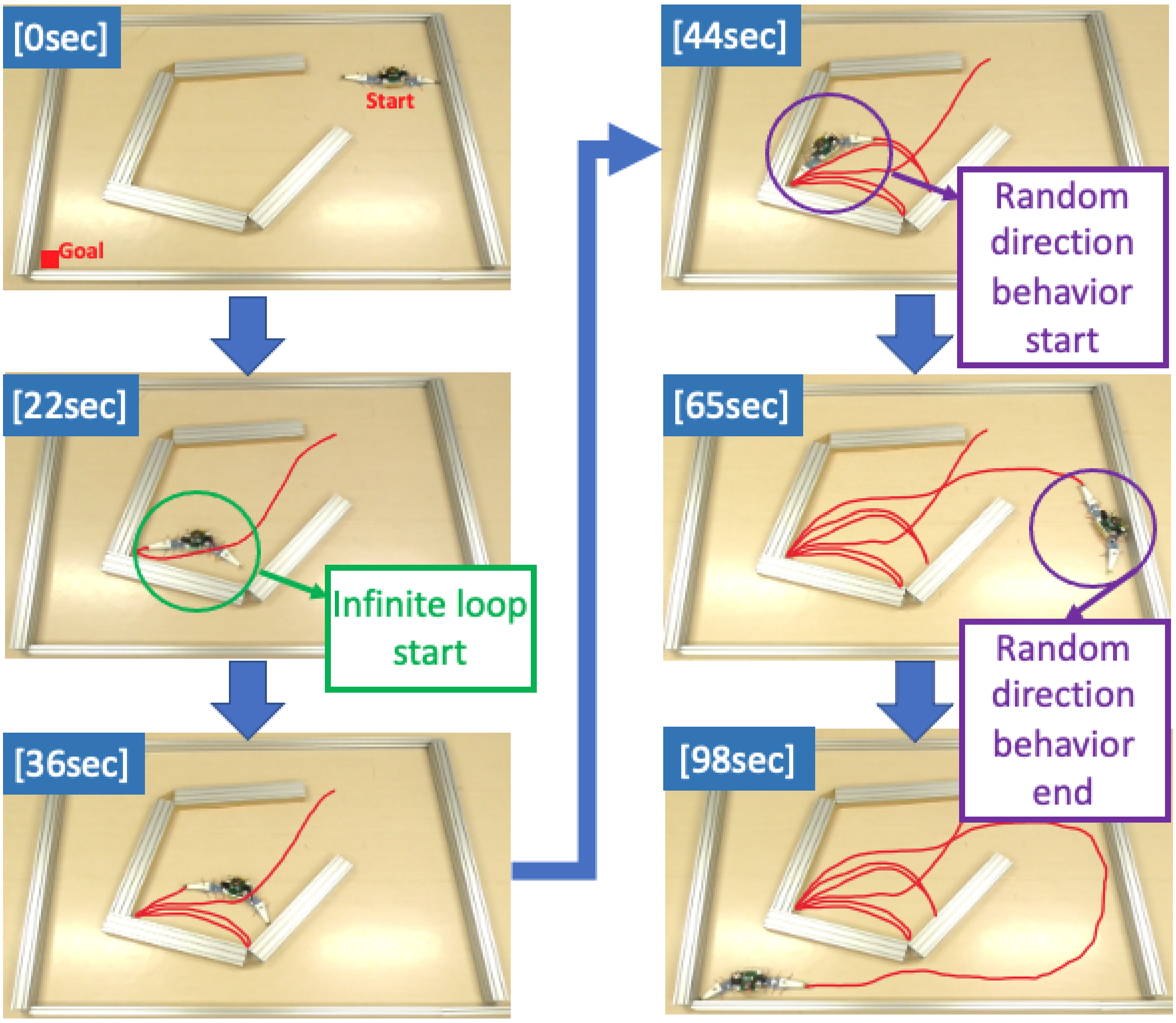
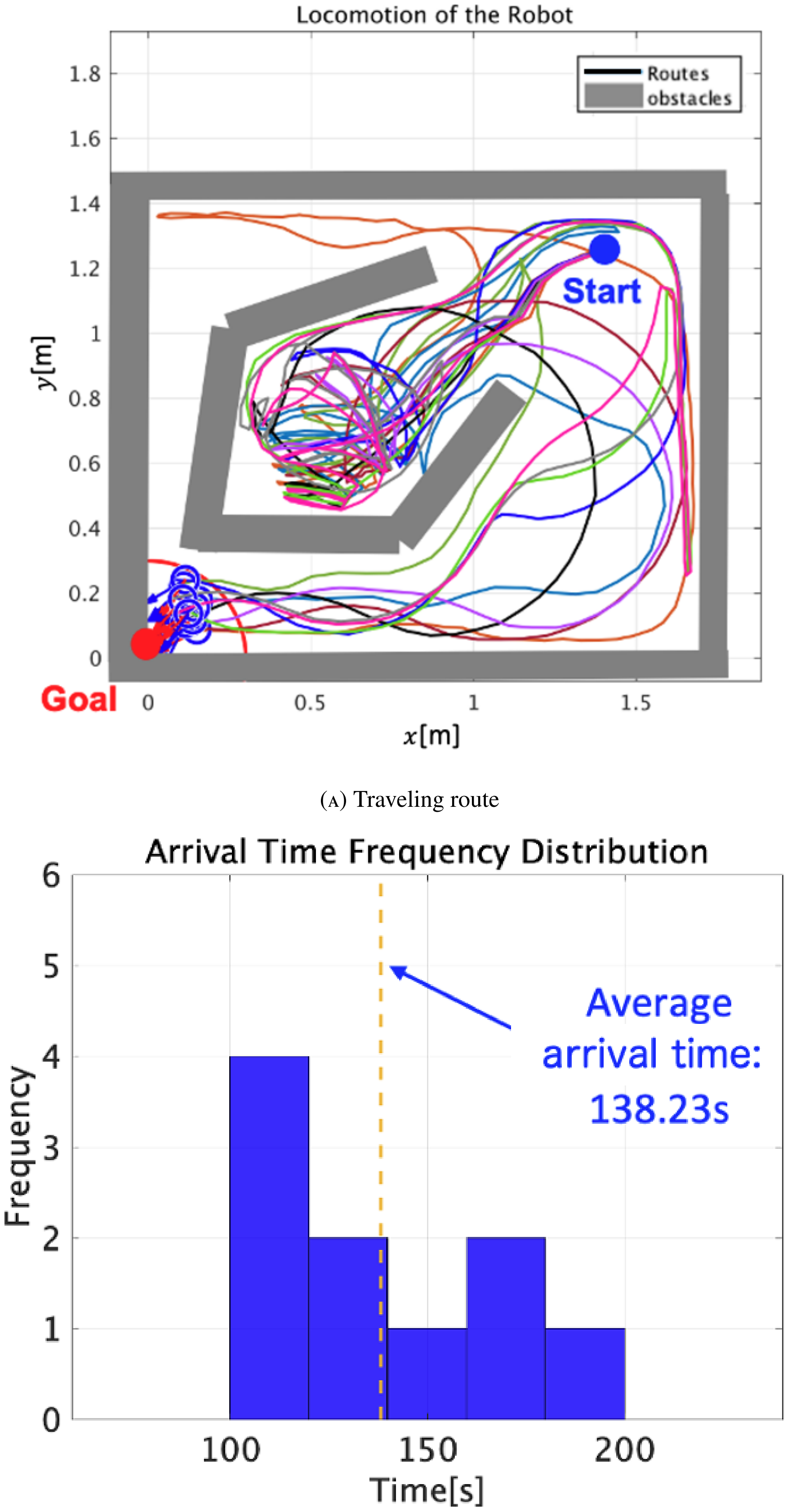


FIGURE 2.17: Environment 3: “maze” with a trap



(A) Traveling route

(B) Time distribution

FIGURE 2.18: Route and time distribution for environment 3

Chapter 3

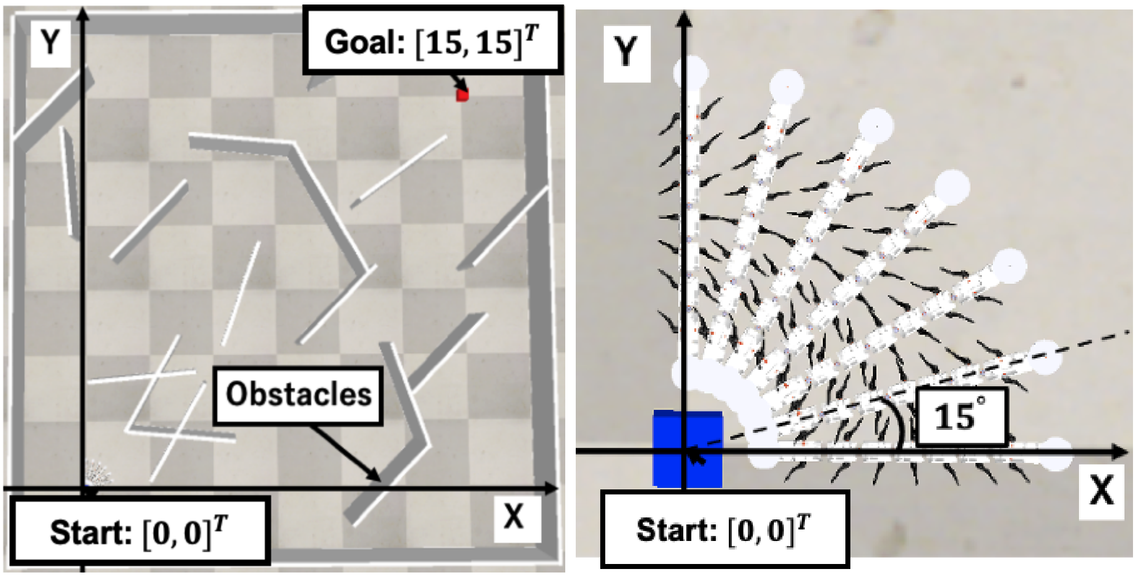
Centipede-like Robot for Swarm Navigation Tasks in 2D Unknown Environments[83]

3.1 Introduction

In Chapter 2, we proposed a centipede-like navigation robot that is systemically simple yet capable of adeptly performing individual robot navigation tasks in 2D unknown environments. However, it's important to note that the centipede-like navigation robot proposed in Chapter 2 is not without its challenges. It faces the following issues: A solitary robot employing CWO scheme possesses limited environmental exploration capabilities, resulting in circuitous and protracted navigation paths. To remedy this problem, it may be advantageous to transform the individual centipede-like robots in Chapter 2 into a swarm and furthermore expand the content of implicit control by capitalizing on more environmental factors which are generally considered "unfavorable" as implicit control in addition to robot-environment collisions. Therefore, in this chapter, our focus shifts to the navigation problem of swarm robots in 2D unknown environments. Expanding on the robot introduced in Chapter 2, we further embrace the principle of utilizing environmental factors and propose a swarm navigation robot system. This system is even simpler in terms of robot design, yet it surpasses the Chapter 2 model in environmental exploration capabilities and navigation efficiency.

Specifically, by utilizing unfavorable environmental factors such as robot-robot collisions, signal noise, and signal intervals as implicit control, the robot in this research further reduces the proportion of explicit control within the robotic system in comparison to the robot described in Chapter 2. This simplification not only renders the individual robotic system more simple and brainless but also enables it to obtain randomness of movement to help expand the exploration range in unknown environments and increase the probability of finding shorter paths. For specific details on how the individual navigation control of the robot in this study differs from the robot in Chapter 2, please refer to the introduction of Eq. (3.1) in Section 3.3.

Consequently, this study aims to propose a centipede-like swarm robot system for a 2D unknown environment called "i-CentiPot-swarm"(ICT-swarm), which harnesses "unfavorable" environmental effects to accomplish unknown environment exploration and navigation



(A) Environment (B) Initial state

FIGURE 3.1: Problem statement

without mutual observation, mutual communication, and obstacle sensing. The distinguishing feature of this system is its ability to utilize "unfavorable" environmental factors as implicit control, simplifying the swarm robotic system while maintaining considerable adaptability and performance in navigation and exploration. We will first validate the positive impacts of these "unfavorable" environmental effects in a 2D unknown environment via simulations in the robotics simulator CoppeliaSim, which has been preliminarily introduced at the 28th International Symposium on Artificial Life and Robotics. Subsequently, we analyze and summarize the environmental characteristics associated with generating these favorable effects and present initial validation of its efficacy in a random unknown environment by comparing experiment results in four different simulation environments through simulations.

The organization of this research is articulated as follows: Section 3.2 presents an introduction to the problem statement. The individual robot model of our proposed swarm system is detailed in Section 3.3. Section 3.4 further delves into the overall model of the proposed robotic swarm. The preliminary validation of the proposed method's effectiveness, via a simulation experiment in an unknown environment, is outlined in Section 3.5. Subsequent simulations in four additional unknown environments are then presented and contrasted in Section 3.6. Finally, Section 3.7 concludes the research.

Notation: $\mathbb{R}$ represents the real number field and $\mathbb{R}_+$ represents the positive real number field.

3.2 Problem Statement

In this study, we establish a coordinate system in several 2-dimensional maze environments within the CoppeliaSim simulator [79], using the starting point as the origin $[0,0]^T$ and the goal point as $[15, 15]^T$. Exemplified by **Fig. 4.1(A)**, we randomly position several rectangular wall obstacles between the starting point and the goal point. Subsequently, as depicted in **Fig. 4.1(B)**, we place seven centipede robots at 15-degree intervals from the positive x-axis direction to the positive y-axis direction with a radius of 0.65m from the origin, forming a 90-degree sector formation of robots as the robot swarm's initial position. During each navigation simulation, all seven robots initiate navigation simultaneously, and after a certain time $T = 780$s, we record the navigation success rate, minimum journey, and exploration breadth of the seven robots as benchmarks for evaluating navigation and exploration effects. As for the reason for selecting 780s, we will provide a detailed explanation in Section 3.5.2.

This study compares the navigation and exploration effectiveness of the ICT-swarm with and without robot-robot collisions, sensor signal noise, and signal intervals in one environment initially to ascertain whether these environmental factors are advantageous for navigation and exploration in details. Subsequently, by conducting and comparing simulations in four different environments, we will analyze and summarize the environmental characteristics associated with generating these favorable effects and present initial validation of its efficacy in a random type of unknown environment. It is important to note that this study is based on the analysis of simulation results, thus the signal noise and intervals involved are actually artificially designed and attributed to the system. This simulates the characteristics of noise or intervals generated by commonly used sensors such as sonar and GPS in mobile robots in the real world, which are determined by the environment in practical use. And also notably, the effect of utilizing "unfavorable" environmental effects as implicit control in this research is not measured by how much improvement it brings in terms of navigation success rate, shortest navigation distance, and exploration range compared to traditional robot navigation methods. Instead, it simplifies the navigation methods by utilizing implicit control. This implicit control enables robots to navigate and explore unknown environments "brainlessly" without complex observation and calculation. Therefore, as long as this method can "brainlessly" complete navigation and exploration with relatively acceptable efficiency, it can be considered successful, and the research can be proved to be meaningful.

3.3 Individual Robot Model

3.3.1 Robot Structure

The individual robot in this study is modified and built on the basis of i-CentiPot. As depicted in **Fig. 4.2**, the robot exhibits a centipede-like configuration, comprising seven interconnected body segments. Each segment is connected via a three-degree-of-freedom passive joint possessing an elasticity coefficient of $K = 100$ with a straight body as the initial state. Every body segment features a pair of legs maintaining a constant angular velocity of 2π/s. The robot is composed of two parts, the active part in the middle section and the purely passive part at the ends. The second, third, fourth and fifth joints have a horizontal rotatable motor for steering. The first and sixth joints are always passive, with guide wheels at the head and tail, so as to yield to the guidance of the obstacles or other robot individuals to achieve implicit control.

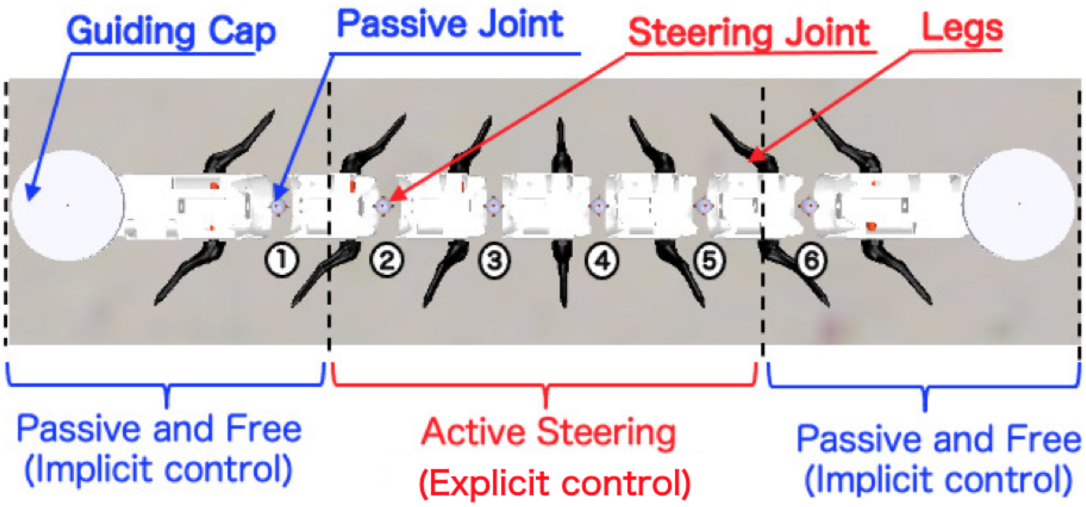
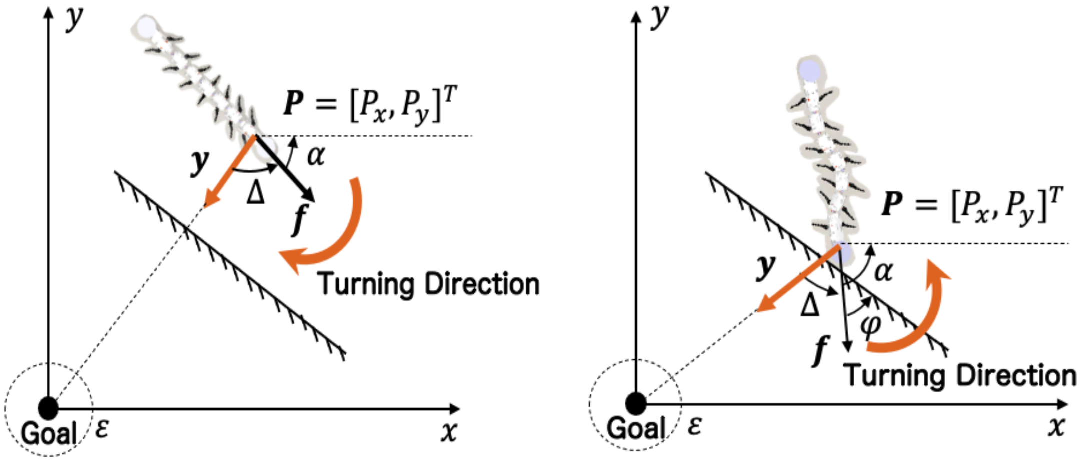


FIGURE 3.2: Robot structure in CoppeliaSim simulator



(A) Move to the goal

(B) Obstacle situation

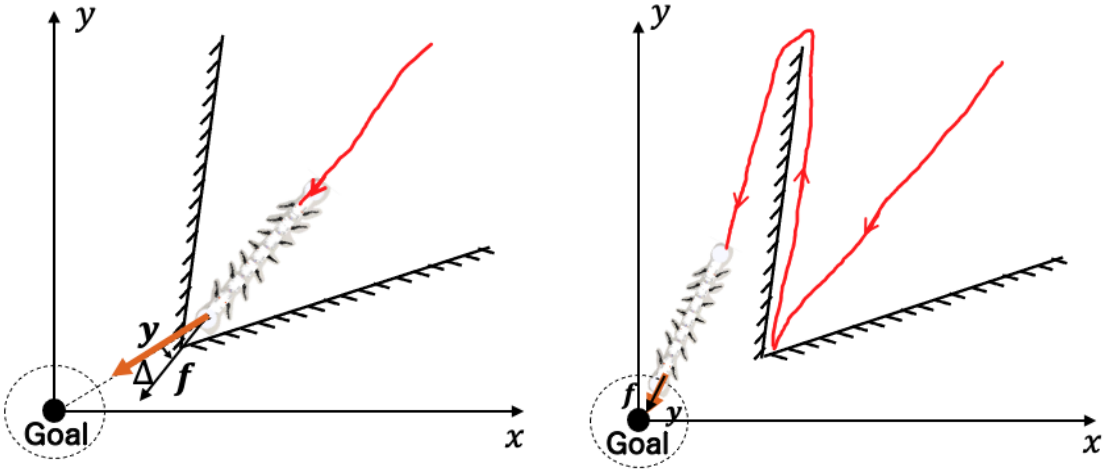
FIGURE 3.3: Behavioral logic 1

3.3.2 Sensor System

In this study, the robot employs the angle $\Delta \in \mathbb{R}$ between the target direction and the robot's forward direction as the sole input signal, which is directly provided to each individual robot through simulation software. This signal can be processed to emulate signal noise, intervals, and other disturbances frequently encountered in real-world sensing methodologies based on sound source orientation, GPS technology, and so on.

3.3.3 Control Model

Compared to the control model utilized in the study concerning the CWO navigation method[82], the control model implemented in this work is further simplified, resulting in explicit control



(A) Stuck in corner

(B) Reversing behavior

FIGURE 3.4: Behavioral logic 2

derived from the controller with two fundamental behavioral logics as follows:

- Directing the robot to turn towards the goal direction and proceed towards the goal, as illustrated in **Fig. 4.3 (A)**.
- When the robot becomes stuck, as depicted in **Fig. 4.4(A)**, the robot's head functions as the tail, and the tail operates as the head, reversing the forward direction and persisting in its movement, as demonstrated in **Fig. 4.4(B)**.

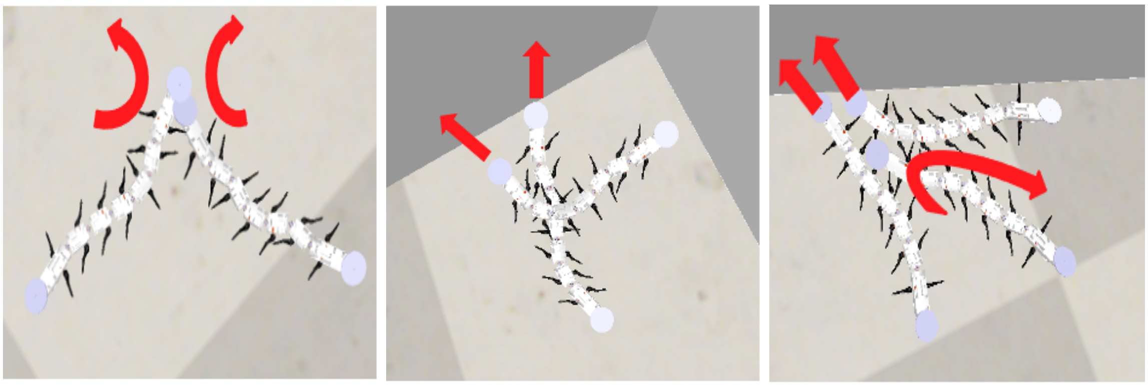
It is noteworthy that when encountering an obstacle, as displayed in **Fig. 4.3 (B)**, the robot's body yields to the obstacle's influence, generating implicit control even as it endeavors to steer towards the goal. Consequently, under the combined effect of implicit control and explicit control, the robot exhibits the behavior of moving along the obstacle's edge until it circumvents the obstacle and resumes progress towards the goal.

Regarding the determination of whether the robot is stuck, we analyze the variance of Δ to ascertain the centipede robot's head swing variance, thereby determining if the robot is stuck. This is because the robot head's swing amplitude is larger during normal movement and diminishes sharply when the robot is stuck[82]. Thus, we only need to calculate whether the variance $S_{\Delta}^{2} \in \mathbb{R}+$ of Δ over a time interval $T_i = 0.1$s is greater than a specific critical value $S_t^2 \in \mathbb{R}_+$, which is 300 deg^2 in degree measurement in this study, to determine if the robot is stuck or not, without incorporating additional observation information beyond Δ.

The mathematical model of the above control logic is shown below.

$$\begin{cases} \dot{P} = A(\Delta + \delta_1\varphi + \delta_{2,n}\pi)vy\cos(\delta_1\varphi), \\ \dot{\alpha} = -K_1(\Delta + \delta_{2,n}\pi) - \delta_1 K_2\varphi. \end{cases} \tag{3.1}$$

For a detailed description of these parameters please refer to our research[82]. It's worth noting here that the control model employed by the robot in this research differs slightly



(A) Collision form 1

(B) Collision form 2

(C) Collision form 3

FIGURE 3.5: Collision form

from the model in Equation (8) of reference [82]. Broadly speaking, the control model in this study has been further simplified compared to that in reference [82]. In reference [82], when the robot falls into an infinite loop within the "trap" terrain mentioned in Section 2.4.2 in this paper, it utilizes a random walk logic to escape the loop. Specifically, in the equation, the random walk logic is represented by the $\delta_3 X$ term in Equation (8) of reference [82]. In this research, our robot has removed the random walk logic. Instead, it replaces the randomness originally generated by the random walk logic with the passively introduced randomness from "unfavorable" environmental factors such as robot-robot collisions, signal noise and signal intervals. Thus, Equation (1) lacks the $\delta_3 X$ term compared to Equation (8) in reference [82]. Through this improvement, we have further expanded the implicit control, simplified the explicit control, and made the robotic system more concise.

3.4 Swarm Robot model

The proposed ICT-swarm system, as detailed in Section 3.4, comprises of 7 individual robots. This swarm operates without information exchange, mutual observation, or obstacle observation. Instead, the swarm robot navigates and explores an unknown environment, relying on implicit control generated by robot-robot and robot-environment collisions, as well as the randomness induced by noise and intervals of sensor signals Δ.

Firstly, concerning collisions, the utility and advantages of robot-environment collisions have been thoroughly described in the study of individual robot CWO navigation[82]; thus, this research emphasizes mutual collisions among robots in an ICT-swarm. In fact, when swarm robots collide with each other, the combination of explicit control and implicit control, as mentioned in section 3.3, yields a novel effect distinct from individual CWO navigation. For instance, as depicted in **Fig. 4.5**, robots may alter their original orientation after a collision between each other, cross over one another, or even trigger reversing behavior.

Subsequently, with respect to signal noise and intervals of Δ, the presence of signal noise and intervals passively introduces a certain degree of randomness in each robot's behavior, leading to a more dispersed and randomized navigation path distribution for the robot swarm.

These environmental influences can, on one hand, assist robots in extricating themselves from challenging situations, such as infinite loops, and on the other hand, enhance the path randomness, augmenting the exploration scope of the robot swarm and thereby increasing the likelihood of discovering shorter navigation paths. The specific effects and advantages of these environmental factors are elaborated upon in Section 3.5 and Section 3.6.

3.5 Simulation Experiment and Analysis in One Environment

3.5.1 Simulation Environment and Parameters

In this section, the proposed swarm robot navigation method and the positive impact of "unfavorable" environmental effects are verified through simulation experiments using the bullet 2.78 dynamics engine in CoppeliaSim in one unknown environment in details firstly. The simulation environment, number of robots, and initial state are set up according to the problem statement in Section 3.2, as shown in **Fig. 4.1(A)** and **Fig. 4.1(B)**.

Three sets of controlled experiments are conducted with the presence or absence of robot-robot collisions, observation signal noise, and observation signal intervals. In set 1, one controlled trial between "With no collision" group and "With only collision" group is conducted and the navigation success rates, minimum navigation journey, and exploration breadth are compared. In set 2 and 3, the controlled trial between "With collision & signal noise"group and "With only collision" group and controlled trial between "With collision & signal interval"group and "With only collision" group are conducted. Especially for "With collision & signal noise" group and "With collision & signal interval"group, 100 trials are performed to calculate the average and frequency distribution of navigation success rate, minimum navigation journey, and exploration breadth due to the introduction of random elements such as noise and intervals, and these results are compared to demonstrate the positive effects of signal noise and signal interval.

3.5.2 Evaluation Method

In this study, we evaluate the effectiveness of navigation in terms of navigation success rate, minimum journey, and exploration breadth.

Navigation success rate.

The robots that are able to reach the goal within 780 seconds are considered successful navigators. If the number of successful navigators among the seven robots in an experiment is $N_{suc} \in \mathbb{R}+$, then the navigation success rate $Rsuc \in \mathbb{R}_{+}$ can be defined as follows.

$$R_{suc} = \frac{N_{suc}}{7} \tag{3.2}$$

It's important to emphasize that the selection of 780 seconds is based on it being the longest achievable time for reaching the goal without being trapped in an infinite loop among groups that have not introduced random variables—specifically the "With no collision" group and the "With only collision" group. This time, rounded to the nearest ten from 778 seconds, serves as a critical benchmark for distinguishing whether a robot will fall into an infinite loop. Specifically, in these relevant groups, if the navigation time exceeds 780 seconds, it indicates an infinite loop, whereas less than 780 seconds signifies successful navigation. Groups with random variables, such as the "With collision & signal interval" group and the "With collision & signal noise" group, were excluded from this analysis as they could not provide a clear statistical basis for determining the longest achievable time due to the introduction of random variables.

Minimum journey.

The minimum journey $J_{min} \in \mathbb{R}_+$ is the distance traveled by the robot with the shortest path to the goal among the seven robots in a single trial.

Exploration breadth.

We divide the entire map into 25 square regions with $4m$ sides as shown in **Fig. 4.6 (B)**, each with a variable $a_i \in \mathbb{R}_+$ that is 1 if at least one robot passes through it, and 0 otherwise. Then we define the exploration breadth evaluation coefficient $S_a \in \mathbb{R}_+$ using the following method.

$$S_a = \frac{1}{25}\sum_{i=1}^{25} a_i \tag{3.3}$$

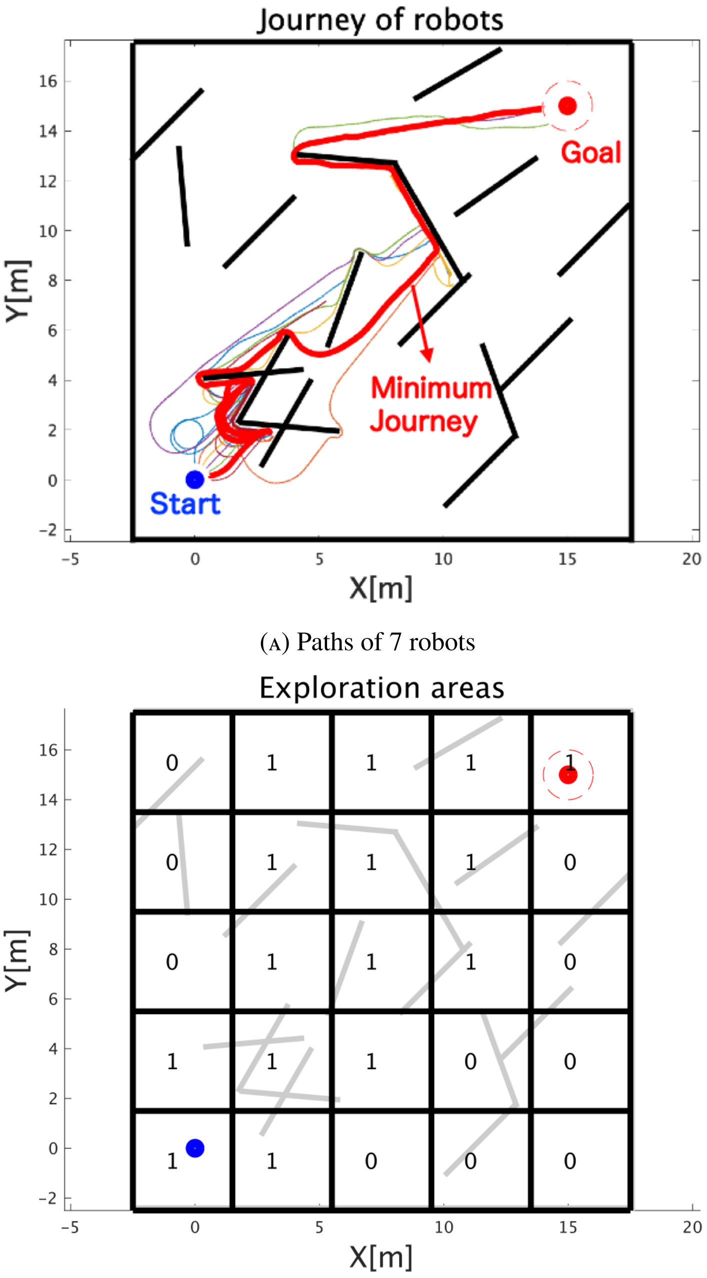
It is worth mentioning that we calculate the average success rate $\bar{R}_{suc} \in \mathbb{R}_+$, average minimum journey $\bar{J}_{min} \in \mathbb{R}_+$, and average exploration breadth $\bar{S}_a \in \mathbb{R}_+$ of 100 trials as evaluation criteria in "With collision & signal noise" group and "With collision & signal interval"group due to the introduction of random elements such as noise and intervals. Specifically, we also calculate the average value of a_i for each region in the 100 trials to reflect the distribution of the explored region more intuitively.

3.5.3 Experimental content and results

Experiments to verify the significance of robot-robot collisions.

First, we carried out a simulation experiment without robot-robot collisions, referred to as the "With no collision" group. To prevent collisions between the robots, each robot in the swarm was navigated individually without others, and the results of the seven trials were aggregated to calculate the overall navigation success rate, shortest path, and exploration breadth.

After the experiments, we obtained a navigation success rate of $R_{suc} = 42.86\%$, a minimum journey of $J_{min} = 55.73$m, and an exploration breadth of $S_a = 60.00\%$. **Fig. 4.6 (A)**



(A) Paths of 7 robots

0	1	1	1	1
0	1	1	1	0
0	1	1	1	0
1	1	1	0	0
1	1	0	0	0

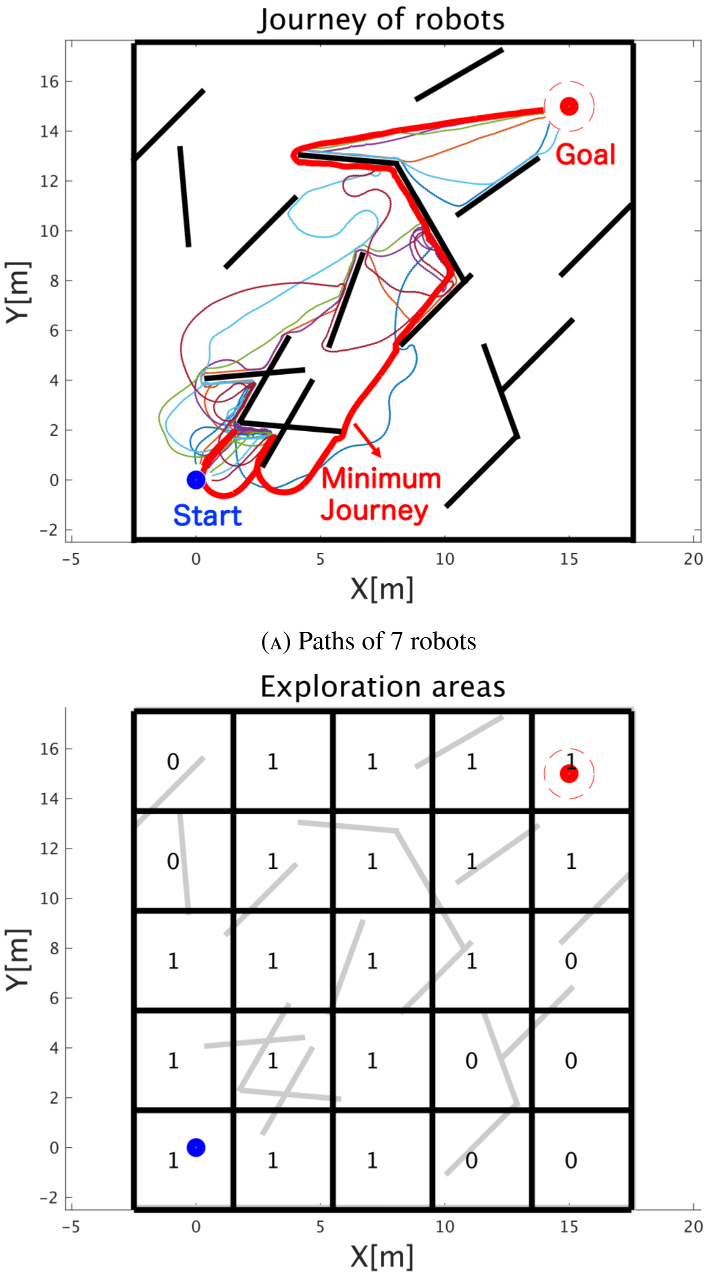
(B) Exploration map

FIGURE 3.6: Situation without robot collision

displays the navigation paths of the seven robots under this condition, with the thickest red line representing the shortest path among the seven robots, and **Fig. 4.6 (B)** illustrating the distribution of the explored areas using the values of a_i in each small square area.

Subsequently, we conducted simulation experiments in the condition with robot-robot collisions, referred to as the “With only collision” group. In this group, we had seven robots navigate simultaneously using the control model presented in Section 3.3.

Following the experiments, we obtained a navigation success rate of $R_{suc} = 85.71\%$, a minimum journey of $J_{min} = 47.98$m, and an exploration breadth of $S_a = 72.00\%$. **Fig. 4.7**



(A) Paths of 7 robots

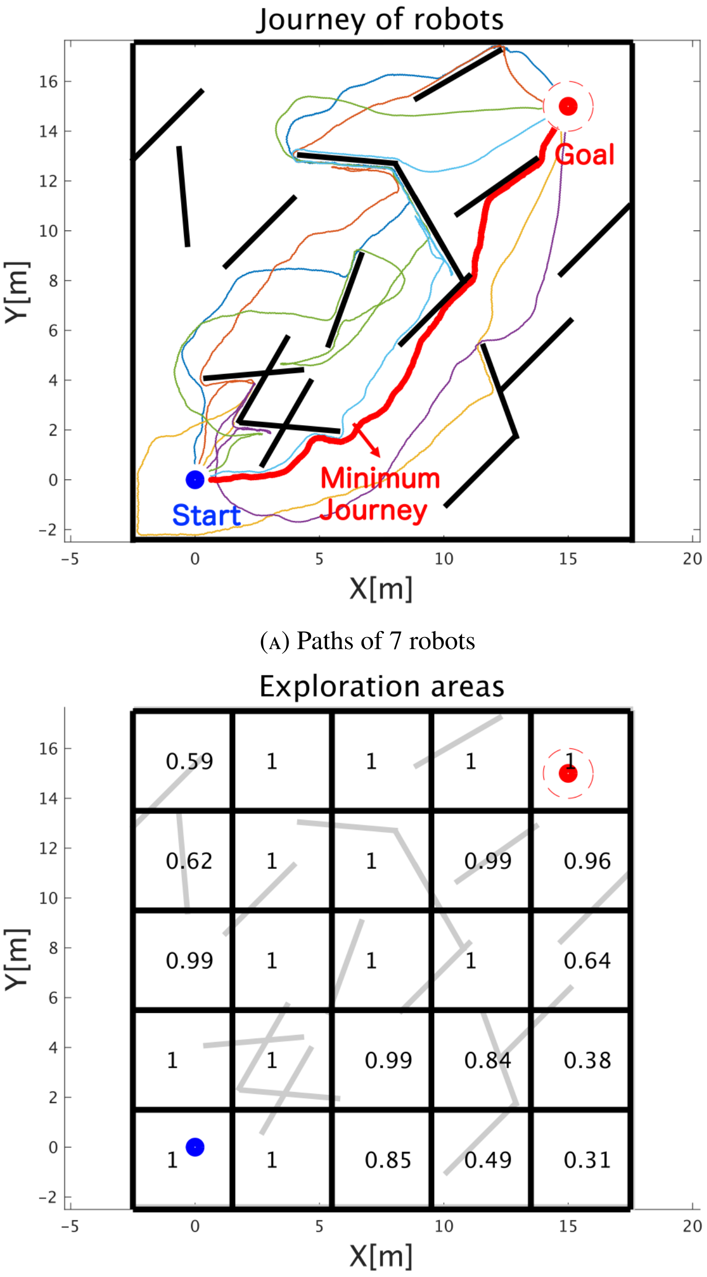
0	1	1	1	1
0	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	1	0	0

(B) Exploration map

Figure 3.7: Situation with robot collision

(A) depicts the navigation paths of the seven robots under this condition, with the thickest red line representing the shortest path among the seven robots, and **Fig. 4.7 (B)** illustrating the distribution of the explored areas with the values of a_i.

Upon comparing the results of the two groups of experiments, we discovered that the "With only collision" group exhibited a significant improvement in navigation success (85.71% vs. 42.86%), minimum journey (47.98m vs. 55.73m), and exploration breadth (72.00% vs. 60.00%) compared to the "With no collision" group.



(A) Paths of 7 robots

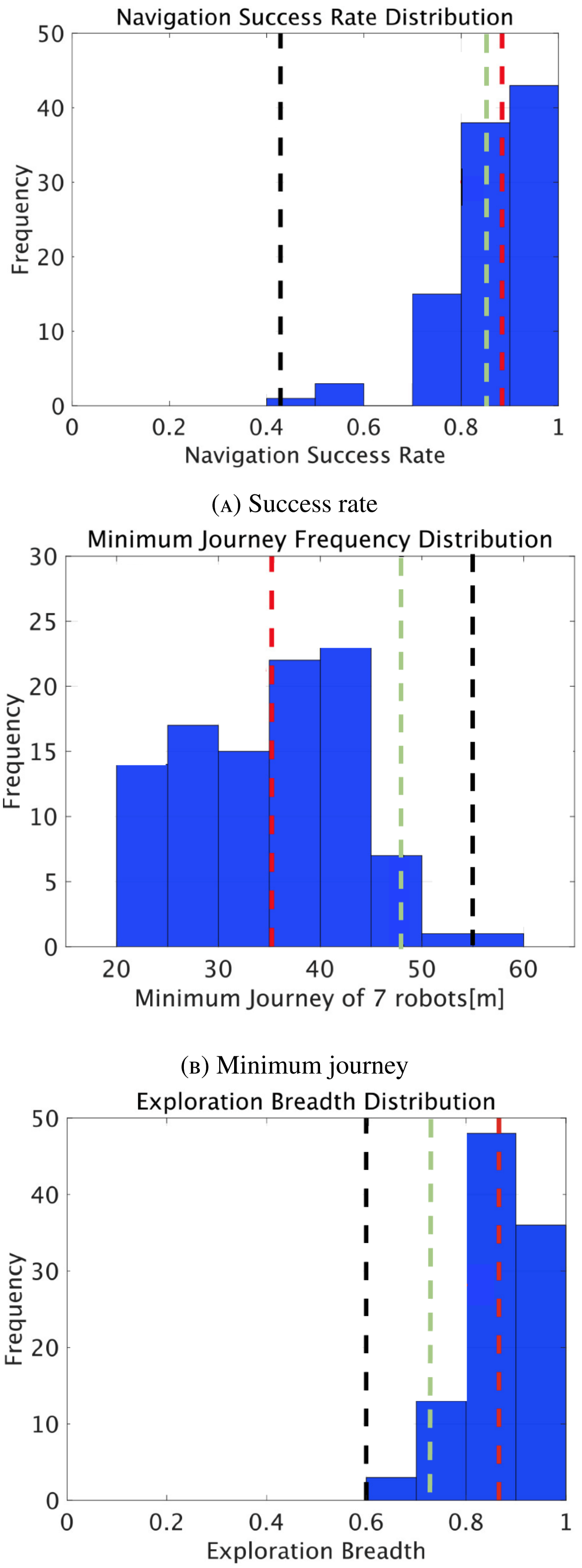
0.59	1	1	1	1
0.62	1	1	0.99	0.96
0.99	1	1	1	0.64
1	1	0.99	0.84	0.38
1	1	0.85	0.49	0.31

(B) Exploration map

FIGURE 3.8: Situation with signal noise

Experiments to verify the significance of signal noise.

In this section, we conducted the simulation experiment of "With collision & signal noise" group. Here, we simulate signal noise by adding noise to the goal direction signal Δ in the control model. Since in reality, microphones, radars, and other sensors generally have the characteristic of being noisier the further they are from the observed object and less noisy the closer they are, for this experiment, we simulate a linear noise model defined as follows.



(A) Success rate

(B) Minimum journey

(C) Exploration breadth

FIGURE 3.9: Distribution of each evaluation coefficient in “With collision & signal noise” group (The red dashed line represents the average value of the “With collision and noise” group, the green dashed line represents the average value of the “With only collision” group, and the black dashed line represents the average value of the “With no collision” group.)

$$Noise \sim N(0, \delta^2) \tag{3.4}$$

where δ^2 starts at 90 and slowly decreases to 0 linearly as it approaches the goal.

After 100 trails with the noise model, we obtained an average navigation success rate of $\bar{R}_{suc} = 85.71\%$, an average minimum journey of $\bar{J}_{min} = 35.24$m, and an average breadth of exploration of $\bar{S}_a = 86.64\%$. **Fig. 4.8 (A)** shows the navigation paths of the seven robots for one of the 100 trials. **Fig. 4.8 (B)** shows the pattern of average a_i in each exploration region in 100 trials. **Figure 3.9 (A), (B), (C)** indicate the frequency distribution of the navigation success rate R_{suc}, and minimum journey J_{min} and exploration breadth S_a in 100 experiments, respectively. In each figure, the red dashed line represents the average value of the "With collision and noise" group, the green dashed line represents the average value of the "With only collision" group, and the black dashed line represents the average value of the "With no collision" group.

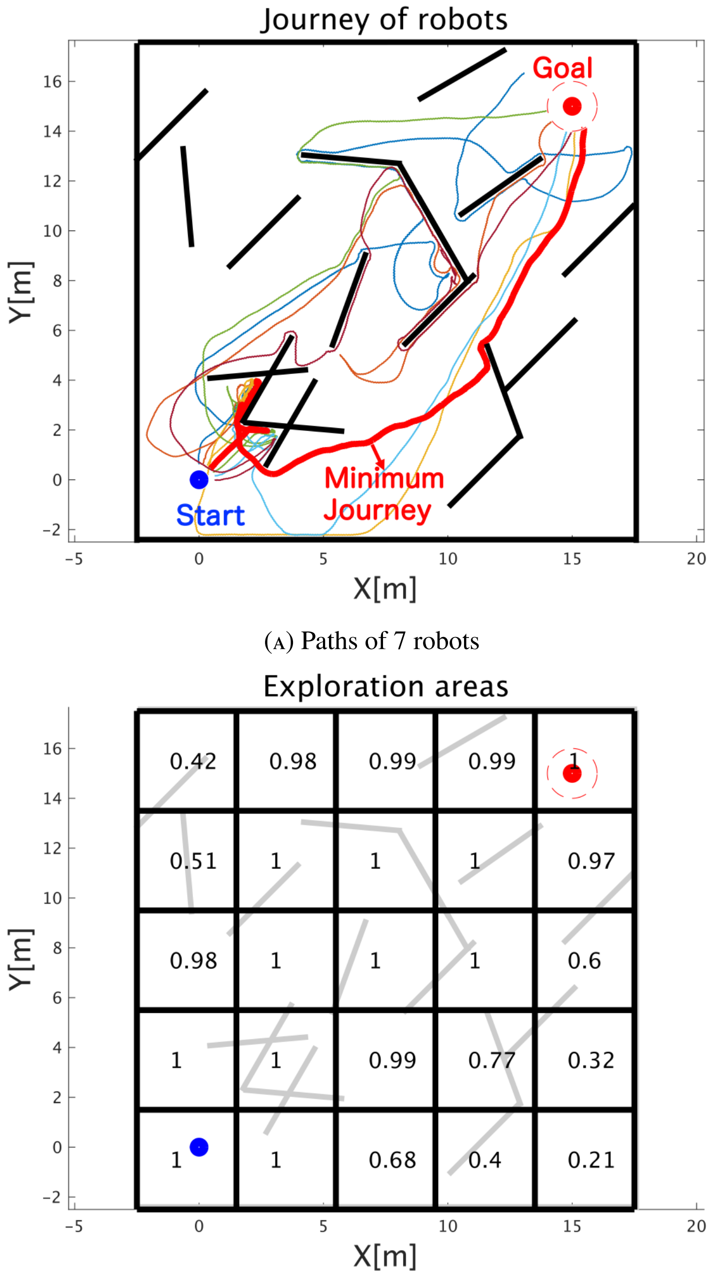
After comparing the results of this set with "With no collision" group, we found that the "With collision & signal noise" group showed an improvement in navigation success (88.43% vs. 42.86%), minimum journey (35.24m vs. 55.73m), and exploration breadth (86.64% vs. 60.00%). And compared to the "With only collision" group, this set also improved in navigation success (88.43% vs. 85.71%), minimum journey (35.24m vs. 47.98m), and exploration breadth (86.64% vs. 72.00%).

Experiments to verify the significance of signal interval.

In this section, we conducted the simulation experiment of "With collision & signal interval" group. In realistic robotic navigation, we sometimes encounter situations where external signals are lost, such as a loss of GPS signal. In this group, we simulated the interval of the goal direction signal Δ by making it disappear intermittently. Specifically, there was a probability of 20% that the robot would lose the signal every second when the signal was available, and the duration of the lost signal $T \in \mathbb{R}_+$ was defined in uniform distribution as follows.

$$T \sim U[1, 5] \tag{3.5}$$

After 100 trails with the interval model, we obtained an average navigation success rate of $\bar{R}_{suc} = 85.43\%$, an average minimum journey of $\bar{J}_{min} = 41.51$m, and an average breadth of exploration of $\bar{S}_a = 83.24\%$. **Fig. 4.9 (A)** shows the navigation paths of the seven robots for one of the 100 trials. **Fig. 4.9 (B)** shows the pattern of average a_i in each exploration region in 100 trials. **Figure 3.11 (A), (B), (C)** indicate the frequency distribution of the navigation success rate R_{suc}, and minimum journey J_{min} and exploration breadth S_a in 100 experiments, respectively. In every diagram, the red dashed line signifies the mean value of the group "With collision and noise", the green dashed line indicates the mean value of the group "With only collision", and the black dashed line denotes the mean value of the group "With no collision".

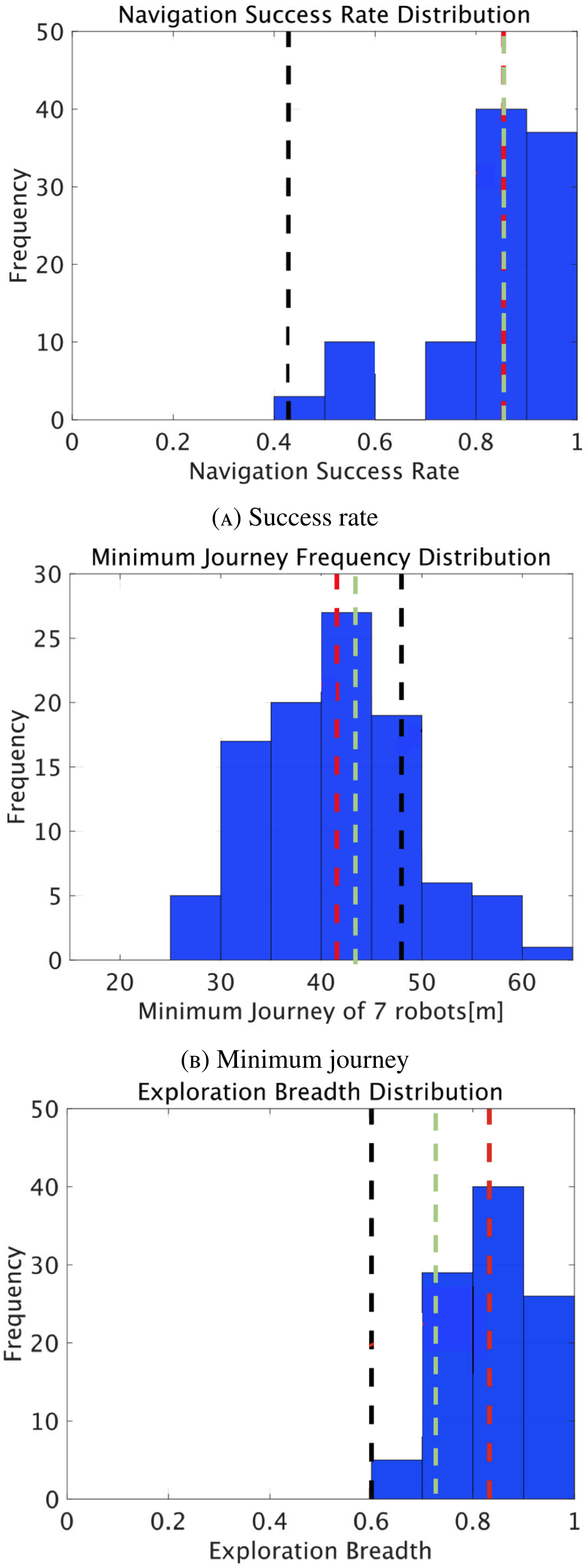


(A) Paths of 7 robots

(B) Exploration map

FIGURE 3.10: Situation with signal interval

After comparing the results of this group with "With no collision" group, we found that "With collision & signal interval" group showed an improvement in navigation success (85.43% vs. 42.86%), minimum journey (41.45m vs. 55.73m), and exploration breadth (83.24% vs. 60.00%). And compared to the "With only collision" group, this group also improved in average minimum journey (41.45m vs 47.98m), and the average breadth of exploration (83.24% vs 72.00%), although the success rate of navigation in this set was slightly lower but almost the same (85.43% vs 85.71%).



(A) Success rate

(B) Minimum journey

(C) Exploration breadth

FIGURE 3.11: Distribution of each evaluation coefficient in "With collision & signal interval" group (The red dashed line represents the average value of the "With collision and interval" group, the green dashed line represents the average value of the "With only collision" group, and the black dashed line represents the average value of the "With no collision" group.)

Table 3.1: Overall comparison

	With no collision	With only collision	With collision & signal noise	With collision & signal interval
Navigation success rate	42.86%	85.71%	88.43%	85.43%
Minimum journey	55.73m	47.98m	35.24m	41.45m
Exploration breadth	60%	72%	86.64%	83.24%

3.5.4 Discussion

We have summarised the results for navigation success rate, minimum journey, and exploration breadth for "With no collision" group, "With only collision" group, "With collision & signal noise" group, and "With collision & signal interval" group in Table 3.1.

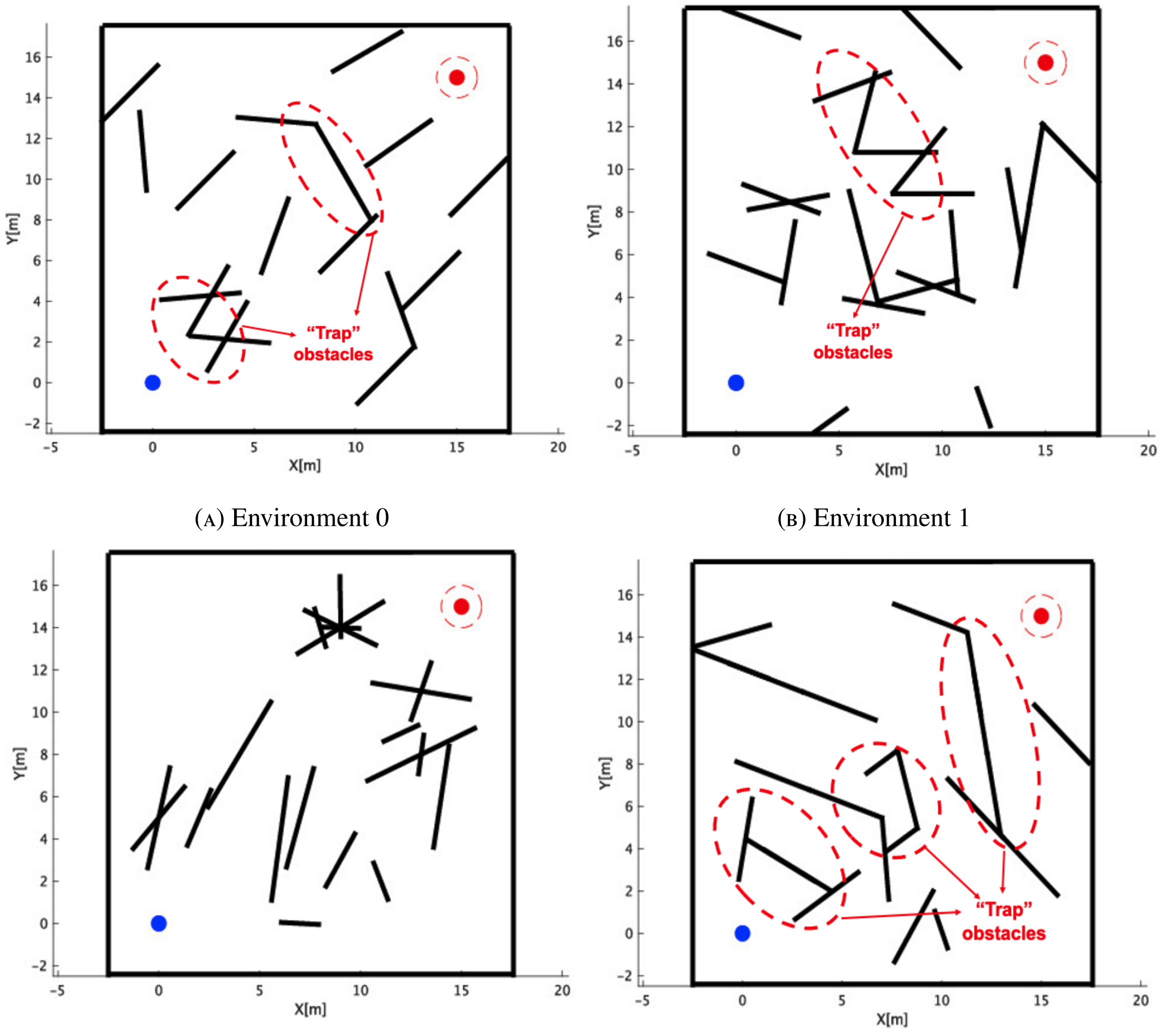
Upon analyzing the results, we discovered that collisions between robots, signal noise, and signal interval all improve or maintain the robot's navigation success rate, shorten the minimum journey, and increase the exploration range. Through these simulation experiments, we initially demonstrated the positive significance of collision between robots, signal noise, and signal interval.

3.6 Multi-Environment Comparative Simulation Experiment

In simulations in Section 3.5, we preliminarily verified that in our proposed centipede robot swarm system, collisions between robots, observation signal noise, and observation signal intervals are advantageous for robot navigation and exploration tasks in some unknown environments. In this section, we further explore whether the aforementioned "unfavorable" environmental effects are still advantageous in more diverse environments and what characteristics of the environment make these "unfavorable" environmental effects advantageous. Finally, based on the results of these explorations, we preliminarily demonstrate the efficacy of our proposed method in any type of unknown environment by comparing simulation results in four different environments.

3.6.1 Simulation Environment and Content

Following the same navigation rules and robot initial conditions as in Section 3.5, we prepared three different unknown environments for the robots, as shown in the **Fig. 3.12**, namely Environment 1, Environment 2, and Environment 3. While, the simulation environment in Section 3.5 is referred to as Environment 0. Then, in each environment, we still conducted simulations in four groups according to "With no collision" group, "With only collision" group, "With collision & signal noise" group, and "With collision & signal interval" group, using the same parameters and methods as in Section 3.5, and calculated the average navigation success rate, average minimum journey, and average exploration breadth for each group in each environment.



(A) Environment 0

(B) Environment 1

(C) Environment 2

(D) Environment 3

Figure 3.12: Simulation environments (The obstacles inside the red dashed circle are categorized as "Trap" obstacles which can trap robots to some extent by allowing them to loop around similar paths near the obstacle.)

3.6.2 Simulation Results

As shown in **Fig. 3.13**, the average navigation success rates for each experimental group in Environment 0, Environment 1, Environment 2, and Environment 3 are compared. In Environment 0, the average navigation success rates were 42.86%, 85.71%, 88.43%, and 85.43% for "With no collision" group, "With only collision" group, "With collision & signal noise" group, and "With collision & signal interval" group, respectively; in Environment 1, the corresponding rates were 57.14%, 71.43%, 75.57%, and 77.29%; in Environment 2, the rates were 100%, 100%, 98.14%, and 97.00%; and in Environment 3, the rates were 0, 14.29%, 34.86%, and 30.71%. We found that in Environments 0, 1, and 3, the navigation success rates of the groups with collisions were significantly higher than those with no collisions, while the navigation success rates of the groups with noise and intervals were also improved compared to the groups with collisions only. In Environment 2, all groups achieved near

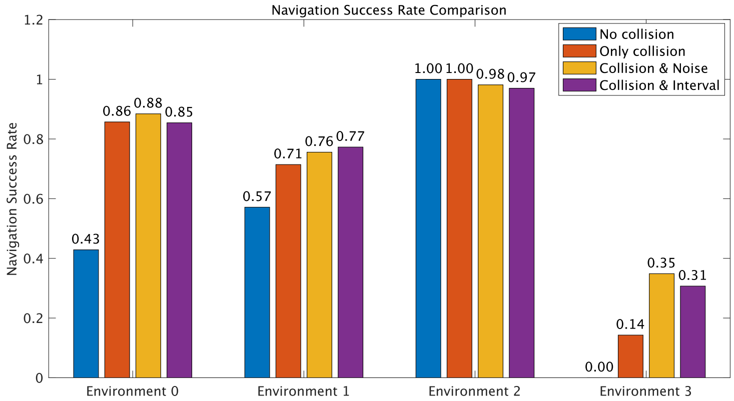


Figure 3.13: Comparison of success rate in 4 environments

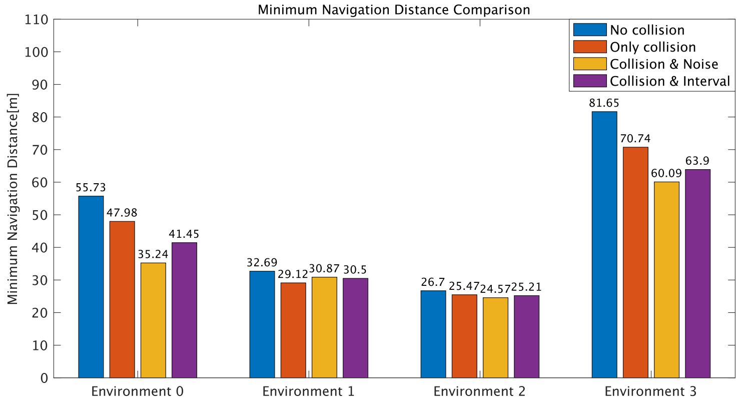


Figure 3.14: Comparison of minimum journey in 4 environments

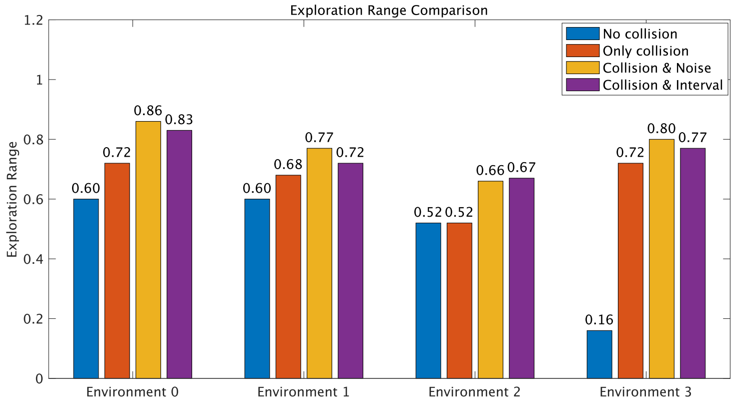


Figure 3.15: Comparison of exploration breadth in 4 environments

100% navigation success rates, with the success rates of the groups with noise and intervals slightly lower (98.14% and 97.00%), but the difference was small and could be approximated to 100%. Finally, comparing the data of the groups with intervals and groups with noise, the results were similar with no significant advantage or disadvantage.

Next, as shown in **Fig. 3.14**, the average minimum navigation journey for each experimental group in Environment 0, Environment 1, Environment 2, and Environment 3 are compared. In Environment 0, the average minimum navigation journeys were 55.73m, 47.98m, 35.24m, and 41.45m for the “With no collision” group, “With only collision” group, “With collision & signal noise” group, and “With collision & signal interval” group, respectively; in Environment 1, the corresponding journeys were 32.69m, 29.12m, 30.87m, and 30.50m; in Environment 2, the journeys were 26.70m, 25.47m, 24.57m, and 25.21m; and in Environment 3, the journeys were 81.65m, 70.74m, 60.09m , and 63.90m. We found that in all environments, the average minimum navigation journeys of the groups with collisions were shorter than those with no collisions, with the average minimum journey decreasing more significantly in Environments 0, 1, and 3 and less so in Environment 2. Moreover, in Environments 0 and 3, the minimum navigation journeys of the groups with noise and intervals decreased significantly compared to the groups with collisions only. In Environments 1 and 2, the minimum journeys of the groups with noise and intervals showed no significant change or slight increase compared to the groups with collisions only, but the difference was small and could be ignored. Finally, comparing the data of the groups with intervals and groups with noise, the results were similar with no significant advantage or disadvantage.

Finally, as shown in **Fig. 3.15**, the average exploration breadth for each experimental group in Environment 0, Environment 1, Environment 2, and Environment 3 is compared. In Environment 0, the average exploration breadth was 60%, 72%, 86%, and 83% for the “With no collision” group, “With only collision” group, “With collision & signal noise” group, and “With collision & signal interval” group, respectively; in Environment 1, the corresponding breadth was 60%, 68%, 77%, and 72%; in Environment 2, the breadth was 52%, 52%, 66%, and 67%; and in Environment 3, the breadth was 16%, 72%, 80%, and 77%. We found that in all environments, the average exploration breadth of the groups with collisions was larger than those with no collisions, and the average exploration breadth of the groups with noise and intervals was also larger than those with collisions only. Finally, comparing the data of the groups with intervals and groups with noise, the results were similar with no significant advantage or disadvantage.

3.6.3 Discussion

First, in terms of the navigation performance of ICT-swarm robots, by comparing the results of the simulations, we found that the use of “unfavorable” environmental effects generally improved navigation success rates and shortened the minimum journey in some environments, such as Environments 0, 1, and 3. In environments like Environment 2, the use of “unfavorable” environmental effects did not significantly improve navigation performance but also did not significantly impair navigation efficiency and success rates. By analyzing the characteristics of Environments 0, 1, 3, and Environment 2, we found that in more complex environments, especially those with “trap” type obstacles, as shown by the obstacle within the red circle in **Fig. 3.12**, that can trap robots to some extent by allowing them to loop around similar paths near the obstacle (such as Environments 0, 1, and 3), the use of "unfavorable" environmental effects helps robots escape more quickly, thus improving navigation efficiency and success rates. In simpler environments without "trap" type complex obstacles (such as Environment 2), the use of "unfavorable" environmental effects may not significantly improve navigation performance. However, overall, they do not cause unbearable damage to navigation performance, either.

Second, in terms of the environmental exploration tasks of swarm robots, we found that the use of "unfavorable" environmental effects significantly increased the exploration range of robot swarms, regardless of the environment.

In summary, when facing completely unknown obstacle environments, since we cannot know in advance whether there are many "trap" type complex obstacles in the environment, the use of "unfavorable" environmental effects can greatly enhance the ability of robot swarms to cope with such possibilities in navigation tasks. Even if the unknown environment is simple without "trap" type complex obstacles, the use of "unfavorable" environmental effects does not cause significant damage to navigation performance. Meanwhile, the use of "unfavorable" environmental effects can also expand the exploration range of robot swarms in exploration tasks. Therefore, for a random unknown environment, the method we propose has practical value.

3.7 Conclusions

This study proposes the ICT-swarm system that simplifies the control and sensing system of robots by using "unfavorable" environmental effects for unknown environment exploration and path navigation without mutual communication and obstacle sensing. We have also demonstrated the positive effects of robot-robot collisions, signal noise, and signal interval on robot swarm navigation and exploration through simulation experiments. Then, by analyzing the results obtained from simulations conducted in four distinct environments, we analyzed and summarized the environmental characteristics associated with generating these favorable effects, providing preliminary validation of the system's efficacy in a random unknown environment.

It is noteworthy that this study demonstrates the potential of ICT-swarm in addressing two issues: navigation and exploration in unknown environments. In the future, we can design specialized ICT-swarm systems centered on either of these issues, enhancing their performance in the specific tasks. Moreover, based on the findings of this research, we can not only effectively utilize "unfavorable" environmental elements, but also propose a design method of a swarm robotics control system that actively introduces signal noise or intervals in the future. And the effectiveness of these proposed methods can be validated through the creation of robotic prototypes in real world.

Acknowledgements

This research was supported in part by grants-in-aid for JSPS KAKENHI Grant Number JP22K14277, Grant-in-Aid for JSPS Fellows Grant Number 23KJ1445 and JST Moonshot

Research and Development Program JPMJMS2032 (Innovation in Construction of Infrastructure with Cooperative AI and Multi-Robots Adapting to Various Environments).

Chapter 4

Centipede-like Robot for Individual Navigation Tasks in 3D Unknown Environments[84][85]

4.1 Introduction

In Chapters 2 and 3, the success of centipede-like navigation robots in 2D environments demonstrates that softening the robots' structure to follow the cues of their surroundings and attain implicit control is a potent strategy for robot navigation. Yet, in these researches, the robot essentially interpreted all environmental interactions as beneficial implicit control. While this can facilitate the robot in brainlessly circumventing obstacles in relatively simpler 2D settings, it encounters challenges in intricate 3D unknown terrains. For instance, when the objective is at the top of a mountain and the robot must ascend to reach it, the interaction prompting the robot to slide downhill becomes a harmful interference, as showcased in **Fig. 4.1**. Indiscriminately heeding such guidance will hinder the robot from ever attaining its destination. Thus, a pressing research concern becomes distinguishing between environmental interactions, embracing the favorable while resisting the adverse, as illustrated in **Fig. 4.2**.

Following this line of thought, Xiao et al. validated that centipede robots can decide whether to resist or follow environmental cues by modulating their body stiffness through slope-climbing experiments. Specifically, reducing the body's stiffness makes the robot more compliant to environmental guidance, while enhancing rigidity empowers the robot to resist environmental effects. Building upon these findings, in this chapter, we shift our focus to the navigation problem of individual robots in 3D unknown environments and propose a centipede-like navigation robot that is simple yet capable of adapting to various rugged 3D terrains through real-time adjustments of the robot's body stiffness. This approach enables optimal adaptation to favorable environmental influences and effective contending with unfavorable ones, enhancing the robot's ability to navigate complex 3D terrains.

The organization of this research is presented as follows:

In Section 4.2, we introduce a centipede robot model capable of completing the aforementioned navigation tasks in a simulated environment. This model's effectiveness is preliminarily validated through simulation experiments. Then, in Section 4.3, we further optimize the robot's structure and control, introducing a robot prototype. This prototype undergoes

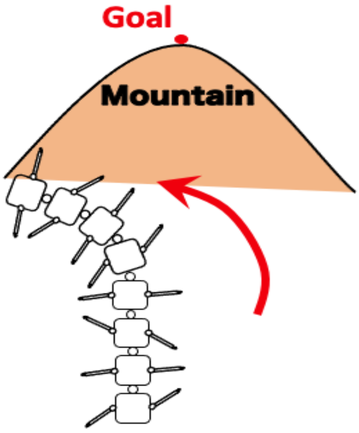


FIGURE 4.1: Problem of CWO navigation method in 3D environment

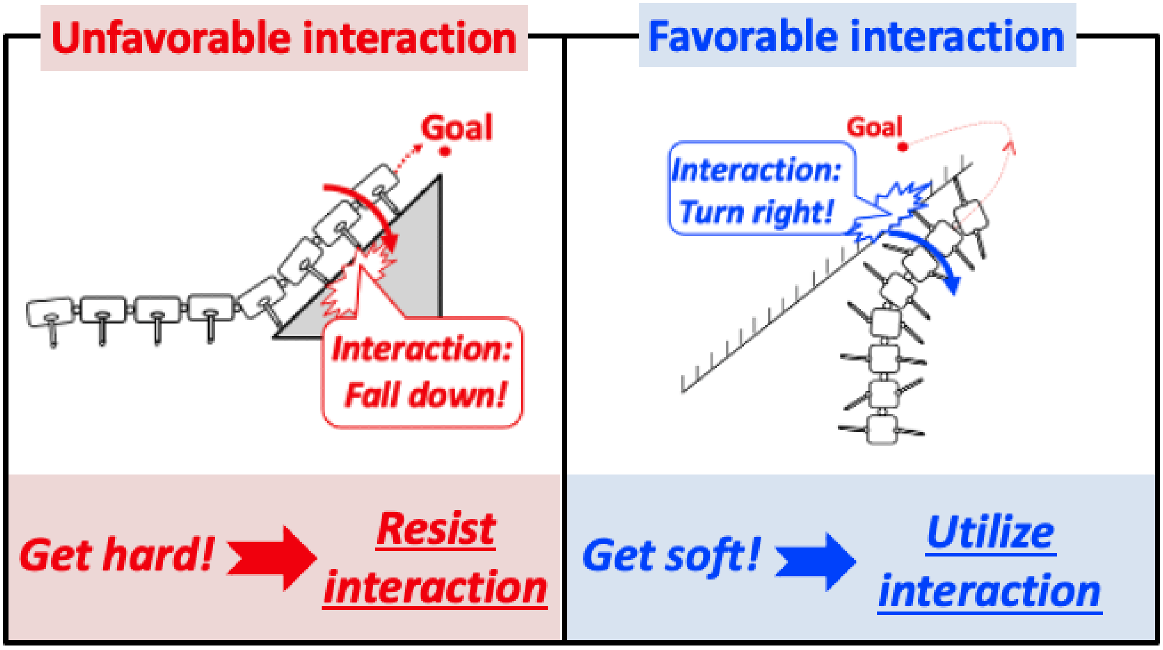


FIGURE 4.2: Idea of this research

comparative experiments in a constructed real 3D navigational environment, emphasizing the significance of the stiffness control loop in the proposed control model. Finally, Section 4.4 synthesizes the content from both the simulation and experimental parts to draw conclusions for this study.

Notation: $\mathbb{R}$ represents the real number field and $\mathbb{R}_+$ represents the positive real number field.

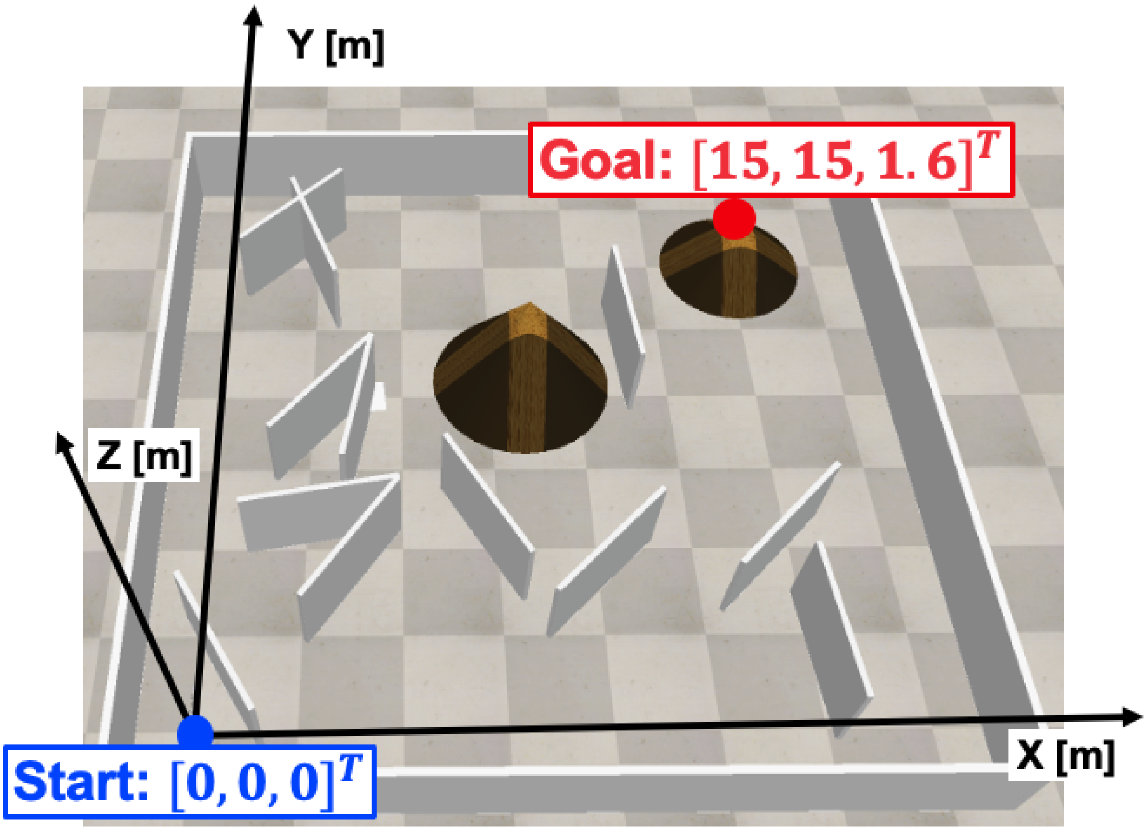


FIGURE 4.3: Problem statement for simulation

4.2 Proposal of Robot Model for Simulation

In this section, we first present our centipede-like navigational robot model for 3D unknown environments within a simulation setting. Here, we sequentially introduce the specific navigation problem statement in the simulated environment, the robot's structural model, the control model, and the content of the simulation. In this way, the advantages and feasibility of the proposed navigation robot are demonstrated preliminarily.

4.2.1 Problem Statement

In this study, we established a coordinate system with the starting point as the origin $[0, 0, 0]^T$ and the goal point as $[15, 15, 1.6]^T$ in a 3D maze environment as shown in **Fig. 4.3** in the Coppeliasim simulator. Between the starting point and the goal point, several rectangular wall obstacles and conical mountains are randomly distributed throughout the area. Notably, the goal point is located at the top of one of these conical mountains. We then placed the robot, proposed in this study, at the starting point and allowed it to navigate towards the goal point following the navigation control method introduced in our research. During this, we recorded its navigational route, internal sensor observations, variations in robot stiffness, and other related data. If the robot can successfully navigate within a confined timeframe, then the proposed navigation method can be preliminarily proven effective. It's important to stress

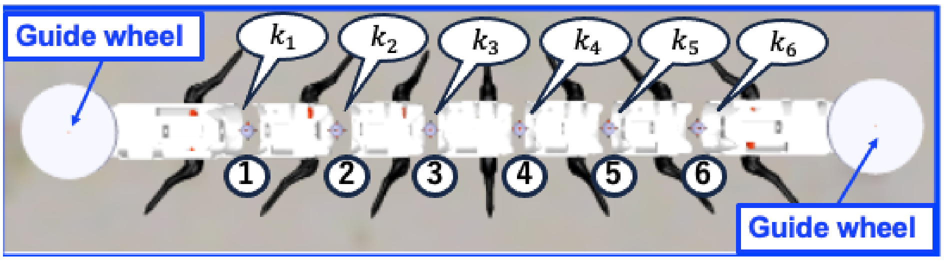


FIGURE 4.4: Robot structure in CoppeliaSim simulator

that the main objective of this study isn't to suggest a method superior in efficiency to the existing ones, but rather to introduce a navigational robot and corresponding control method that can navigate in unknown 3D environments without needing to observe obstacle information, undertake path planning, and relying only on basic control and assistance from the environment. Therefore, as long as the robot can navigate under these specified conditions, this research can be deemed successful.

4.2.2 Robot Structural Model

The robot structure simulated in this study is modified and built on the basis of "i-CentiPot". As illustrated in the **Fig. 4.4**, the robot is a centipede-inspired biomimetic robot, consisting of seven segments each equipped with two driving legs. The segments at both ends are fitted with guide wheels that are easily directed by the environment in a horizontal direction. During the robot's motion, all its legs rotate at a constant angular speed of $2\pi/s$, which can move both forwards and backwards, allowing for bi-directional movement of the robot. Meanwhile, each pair of segments is interconnected through a three-degree-of-freedom joint. We applied Proportional (P) control on the rotational freedom about the vertical axis of every joint with a proportional gain of $k_i \in \mathbb{R}_+$. The target posture is to keep the torso straight during straighte movement and bend it to one side during turning. The rotation in the other two directions is not actively controlled. The index $i \in \{1, 2, 3, 4, 5, 6\}$ represents the joint number, and the value of k_i can be freely switched between 0.01 and 1000, enabling the adjustment of the robot's body stiffness in the horizontal direction. In this study, the k_i for all joints is uniformly set to $k_j \in \mathbb{R}_+$. When k_j is 0.01, the explicit control of each joint is weak, making the body more flexible and easily guided by the environmental forces, with implicit control dominating the robot's actions. Conversely, when k_j is 1000, the explicit control of each joint is strong, making the body more rigid, capable of resisting unfavorable environmental guidance, and thus reducing the influence of implicit control.

4.2.3 Robot Control Model

This study concerns a robot with three input signals. Firstly, the goal direction in relation to the robot on a horizontal plane is defined by the angle $\Delta \in \mathbb{R}$ between the robot's head direction and the goal direction on the XY plane, as illustrated in the **Fig. 4.5**. Secondly, the pitch angle of the robot's head in relation to the horizontal ground is defined as $\theta \in \mathbb{R}$, representing the angle between the robot's head direction and the horizontal plane in a global coordinate system, as depicted in **Fig. 4.6**. Lastly, the average torque $T \in \mathbb{R}_+$ of the robot represents the mean torque of all driving legs at any given moment, measured in $N.m$. Among these, only the goal direction Δ is obtained from external sensors, whereas θ and T are internally sensed.

The control model's principle can be described as follows: When not encountering any obstacles, the robot maintains soft and moves towards the goal direction, as shown in **Fig. 4.7(A)**. When encountering an obstacle like a climbable slope, the robot stiffens to counteract environmental effects like "sliding down the slope", thereby relying on explicit control to traverse the slope, as seen in **Fig. 4.7(B)**. If the obstacle is an insurmountable slope or even a wall, the robot softens, yielding to the obstacle's guidance, and navigates around the obstacle using implicit control, as illustrated in **Fig. 4.7(C)**. In hazardous situations, such as getting stuck or facing a steep slope, the robot can invert, making its tail its head and vice versa, and move in the opposite direction using the same control rule to escape danger, as shown in **Fig. 4.7(D)**.

To realize this aim, the robot's control model consists of two separate part as following.

- Explicit control part: The behavior control loop for navigation and the stiffness control loop for stiffness control of the robot body.

- Implicit control part: When faced with obstacles, it responds based on the robot's varying stiffness.

Then, we will elaborate in detail on the contents of the two control loops in explicit control:

Behavior Control Loop

This control loop governs the robot's forward movement towards the target and, when necessary, retreats in reverse to escape dangerous situations.

Its design essentially aligns with the plane-based CWO navigation model:

$$\begin{cases} \dot{P} = A(\Delta + \delta_1 \varphi + \delta_{2,n}\pi) v y \cos(\delta_1 \varphi), \\ \dot{\alpha} = -K_1(\Delta + \delta_{2,n}\pi) - \delta_1 K_2 \varphi. \end{cases} \tag{4.1}$$

For mathematical representations and symbol definitions, please refer to reference [81][82]. The term $-K_1(\Delta + \delta_{2,n}\pi)$ represents the explicit control component, while $-\delta_1 K_2 \varphi$ represents the implicit control component. The values of K_1 and K_2 are directly related to the value of k_j in the joint stiffness control loop. In this study, the decision to reverse due to danger is determined by whether T exceeds a threshold value $T_s = 0.045 N.m$ or θ surpasses a threshold value$\theta_s = 70 degrees.$ Its value was determined by considering the navigation performance in this simulation environment.

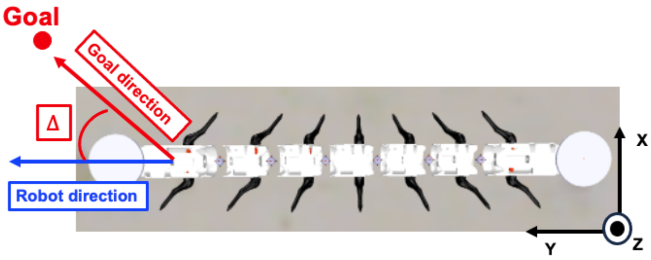


FIGURE 4.5: Goal direction Δ on a horizontal plane

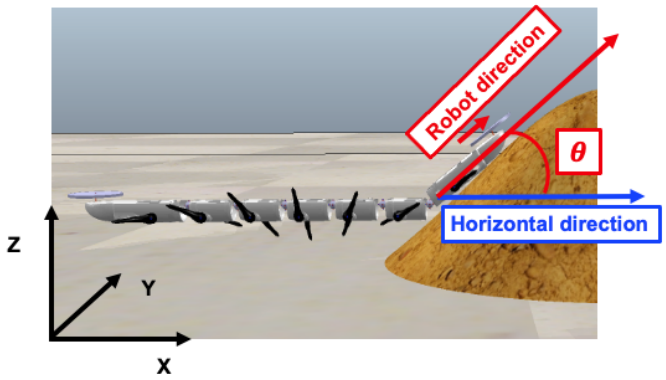


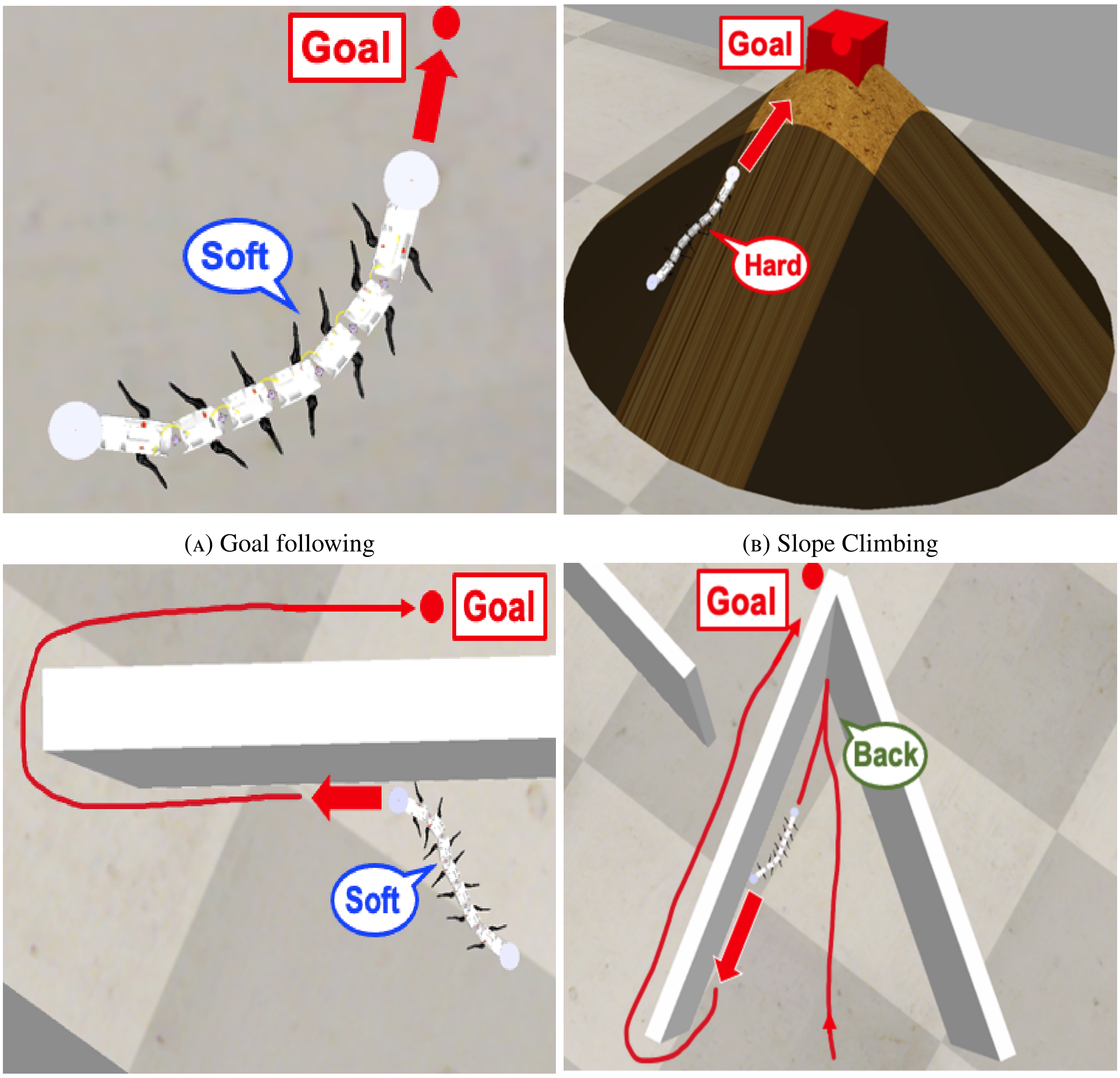
FIGURE 4.6: Pitch angle θ of robot head

Stiffness Control Loop

This governs the stiffness k_j of the robot's body joints. On encountering a traversable slope, the robot becomes rigid; otherwise, it softens. Specifically, as the robot approaches a sloped obstacle, its head gets lifted due to terrain influences. The angle θ can be measured using an IMU sensor mounted on the robot's head. At this time, the stiffness control of the robot's joints can be represented as:

$$K_j = \begin{cases} 0.01 & , \quad if \;\; \theta < \theta_1 \;\; or \;\; \theta > \theta_2 \\ 1000 & , \quad if \;\; \theta_1 \leq \theta \leq \theta_2 \end{cases} \tag{4.2}$$

For conditions where θ is less than upper limit $\theta_1 \in \mathbb{R}$ or greater than lower limit $\theta_2 \in \mathbb{R}$, the robot is perceived to be in a state without obstacles or in front of insurmountable obstacles. These insurmountable obstacles can be wall obstacles (θ less than θ_1) or steep slopes (θ greater than θ_2). Under these conditions, the robot softens, causing K_1 in 4.3 to be much smaller than K_2, allowing the robot to succumb to the environment's implicit control guidance and navigate around obstacles. Conversely, when $\theta_1 \leq \theta \leq \theta_2$, the robot is determined to be facing a traversable obstacle. It stiffens, causing K_1 in 4.3 to be much larger than K_2, allowing it to overcome the environment's guidance and cross the obstacle. Through simulation, θ_1 is set at 13 degrees, and θ_2 is 48 degrees in this research.



(A) Goal following

(B) Slope Climbing

(C) Yielding

(D) Go backing

FIGURE 4.7: Control principle

4.2.4 Simulation

Simulation Environment

In the CoppeliaSim simulation environment, we designed an experimental setting as illustrated in **Fig. 4.8**. The basic details of this setup can be found in Section 4.2.1. Within this setting, the typical obstacle zones the robot encounters are labeled as zones A, B, C, D, E, and F in the illustrations. Specifically:

- Zones A and E represent insurmountable wall obstacles.
- Zones B and C are dangerous “Corner” areas that could trap the robot.
- Zone D represents an insurmountable slope obstacle with a slope gradient of 53 degrees.

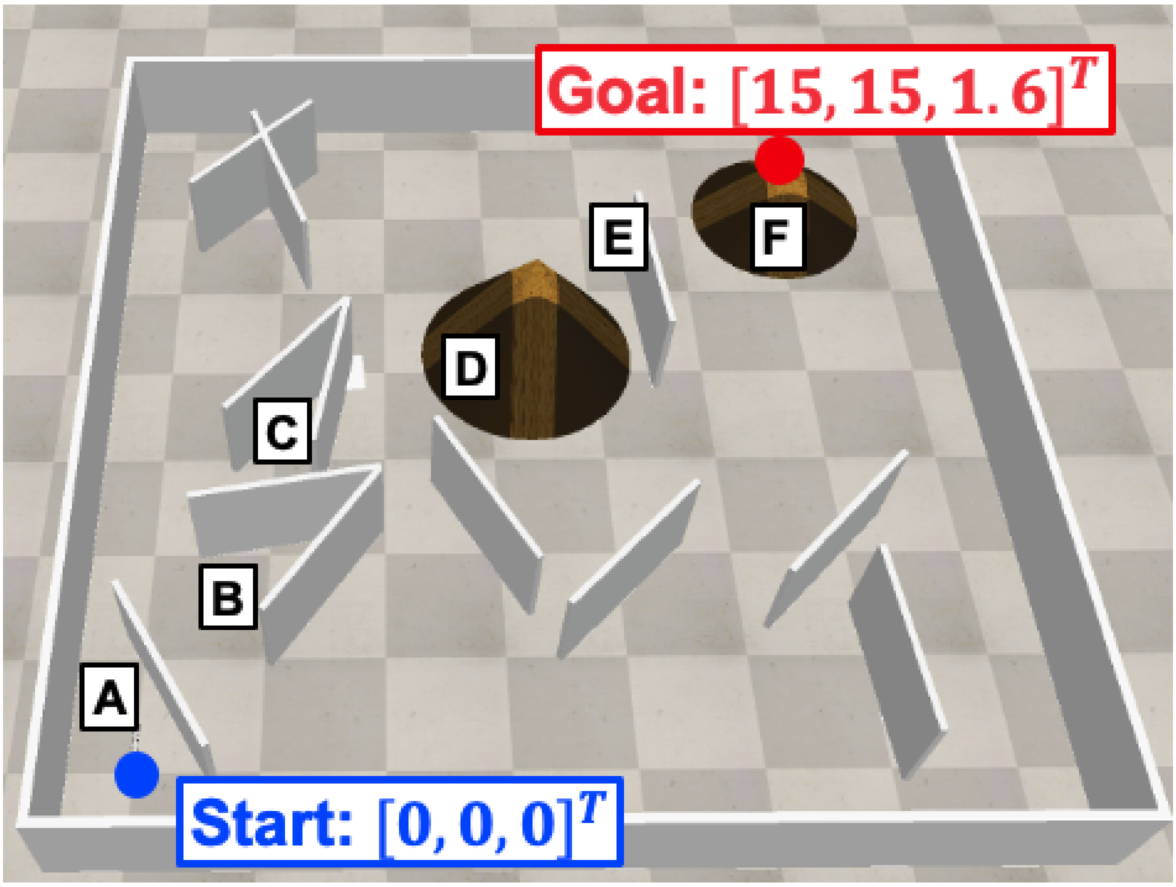


FIGURE 4.8: Simulation environment

- Zone F is a climbable slope obstacle with a gradient of 40 degrees.

Simulation Contents

In our research, we positioned the centipede-like navigation robot at the starting point, facing the y-axis direction, and directed it to navigate towards the goal point using the control method proposed in this study. Throughout the navigation process, we documented the robot's navigational path, the stiffness K_j of at every moment, the pitch angle θ of the robot's head, the average torque T exerted by the legs, and the rotation direction $S \in \mathbb{R}$ of the legs.

For the rotation direction S, a value of 1 indicates the legs are rotating forward, meaning the robot is moving in a forward direction. Conversely, a value of 0 denotes the legs rotating backward, implying the robot is moving in the reverse direction.

Simulation Results

After simulation, the robot successfully reached the goal point located at the peak of Area F. **Fig. 4.9** shows the robot's final navigation route from a top-down view. **Fig. 4.10**, **Fig. 4.11**, **Fig. 4.12**, **Fig. 4.13** display the variations of K_j, θ, T, and S over time, respectively. The sections separated by dashed lines, with a gray background, and marked with letters in these

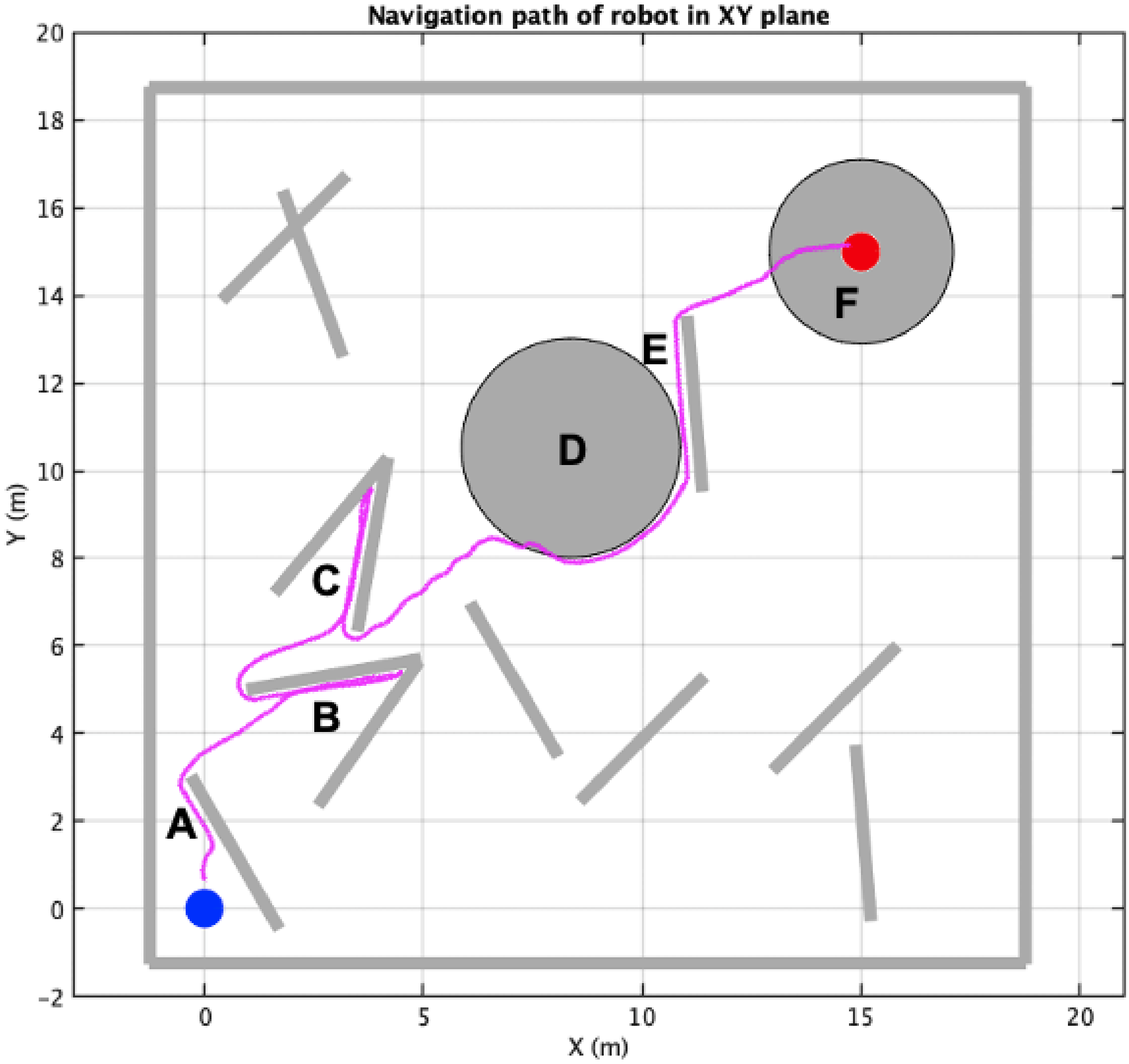


FIGURE 4.9: Navigation path of robot in XY plane

figures represent the six regions from A to F. By analyzing the data, we observed the following:

As shown in **Fig. 4.9**, the robot moves towards the goal point firstly when there are no obstacles encountered. As the robot encounters wall barriers in Areas A and E, θ is less than 13 degrees as illustrated in **Fig. 4.11**. Consequently, the robot's k_j remains at 0.01 as depicted in **Fig. 4.10**, ensuring the robot's body remains soft. This allows the robot to yield to the wall, as seen in **Fig. 4.9**, and navigate around it by following its edge.

When the robot passes through the "corner" regions of Areas B and C, it changes its direction of movement due to θ being greater than 70 degrees (climbing the wall) as shown in **Fig. 4.11** and T being greater than 0.045N.m as shown in **Fig. 4.12**, affecting the value of S as shown in **Fig. 4.13**.

As the robot navigates through Area D, exceeds 48 degrees according to **Fig. 4.11**. Thus, the robot determines that the obstacle in Area D is insurmountable. As a result, its 'k_j remains

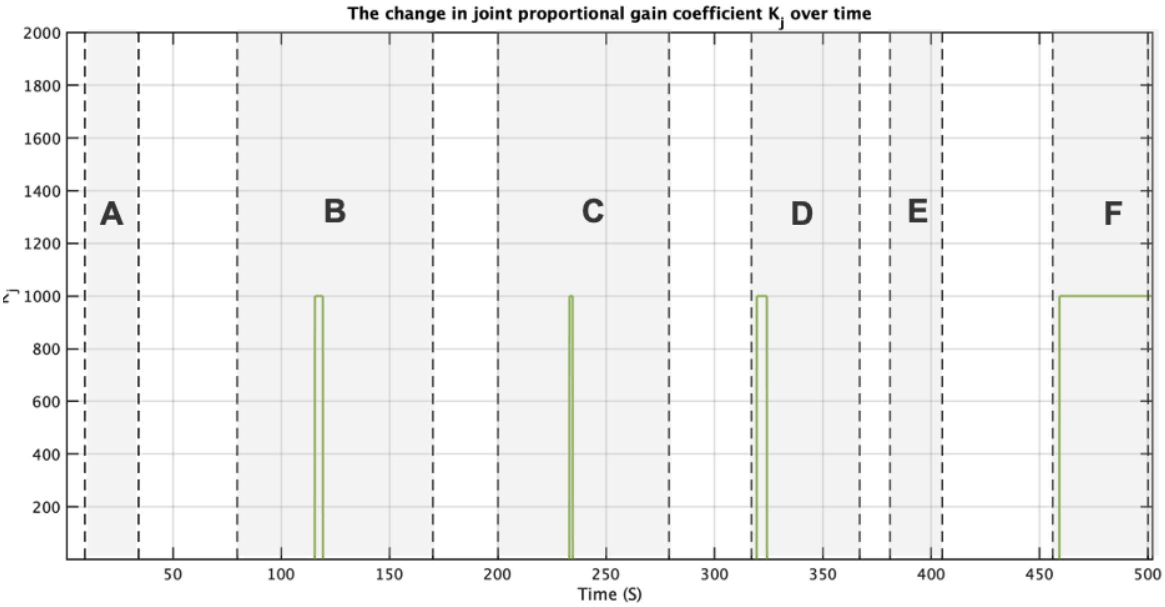


FIGURE 4.10: The change in joint proportional gain coefficient K_j over time

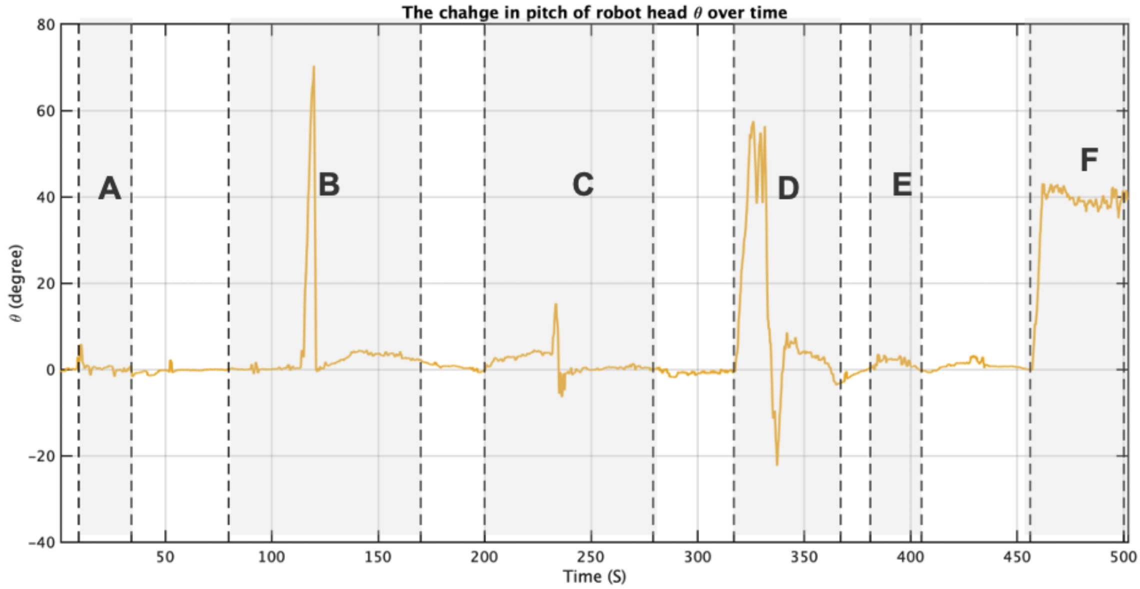


FIGURE 4.11: The chahge in pitch of robot head θ over time

at 0.01 as shown in **Fig. 4.10** to keep its body soft. This leads the robot to yield to the slope, as illustrated in **Fig. 4.9**, and navigate around it by following its edge.

When the robot traverses Area F, **Fig. 4.11** is greater than 13 degrees but less than 48 degrees as indicated in **Fig. 4.11**. Therefore, the robot deems the obstacle in Area F as surmountable. The robot's k_j then increases to 1000 as portrayed in **Fig. 4.10**, hardening its body. This enables the robot to climb the peak as depicted in **Fig. 4.9** and reach the goal point at the summit.

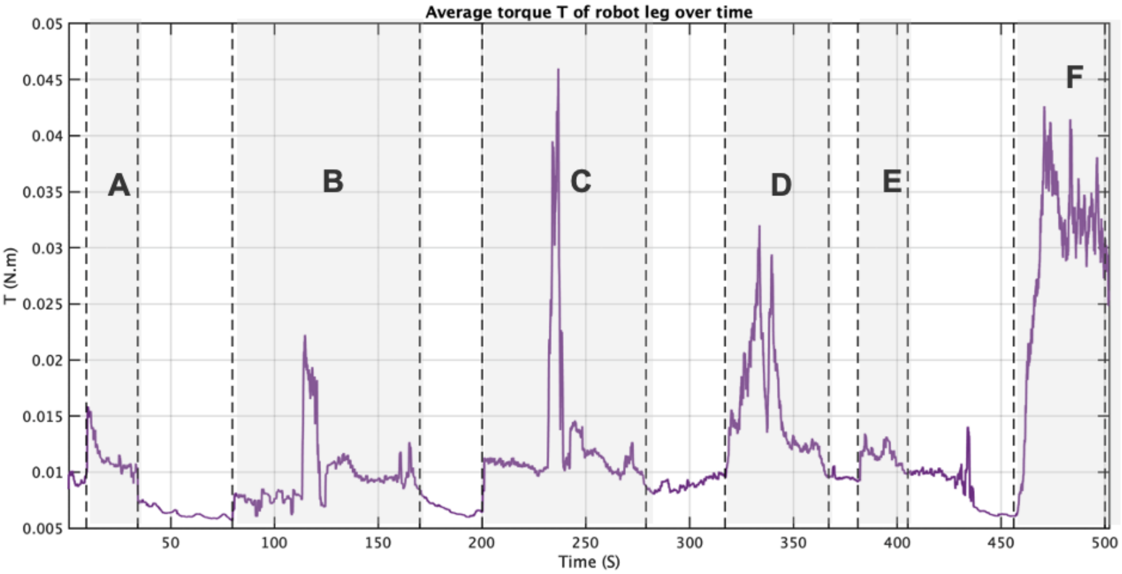


FIGURE 4.12: Average torque T of robot leg over time

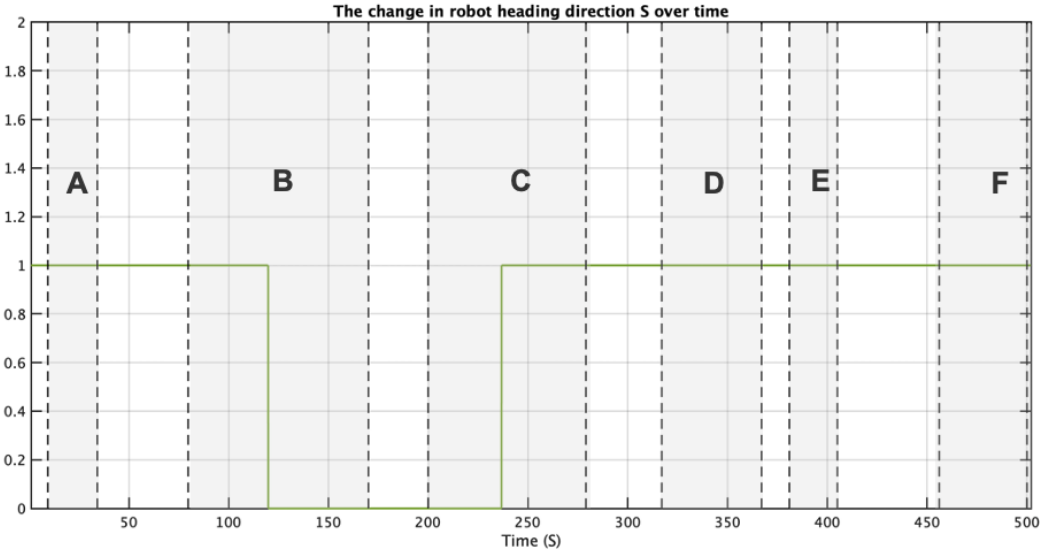


FIGURE 4.13: The change in robot heading direction S over time

Discussion

Through this simulation, it was proven that the method we proposed can successfully navigate to the destination in an unknown 3D environment composed of the aforementioned walls and conical slopes. This initial proof demonstrates the feasibility of our proposed method.

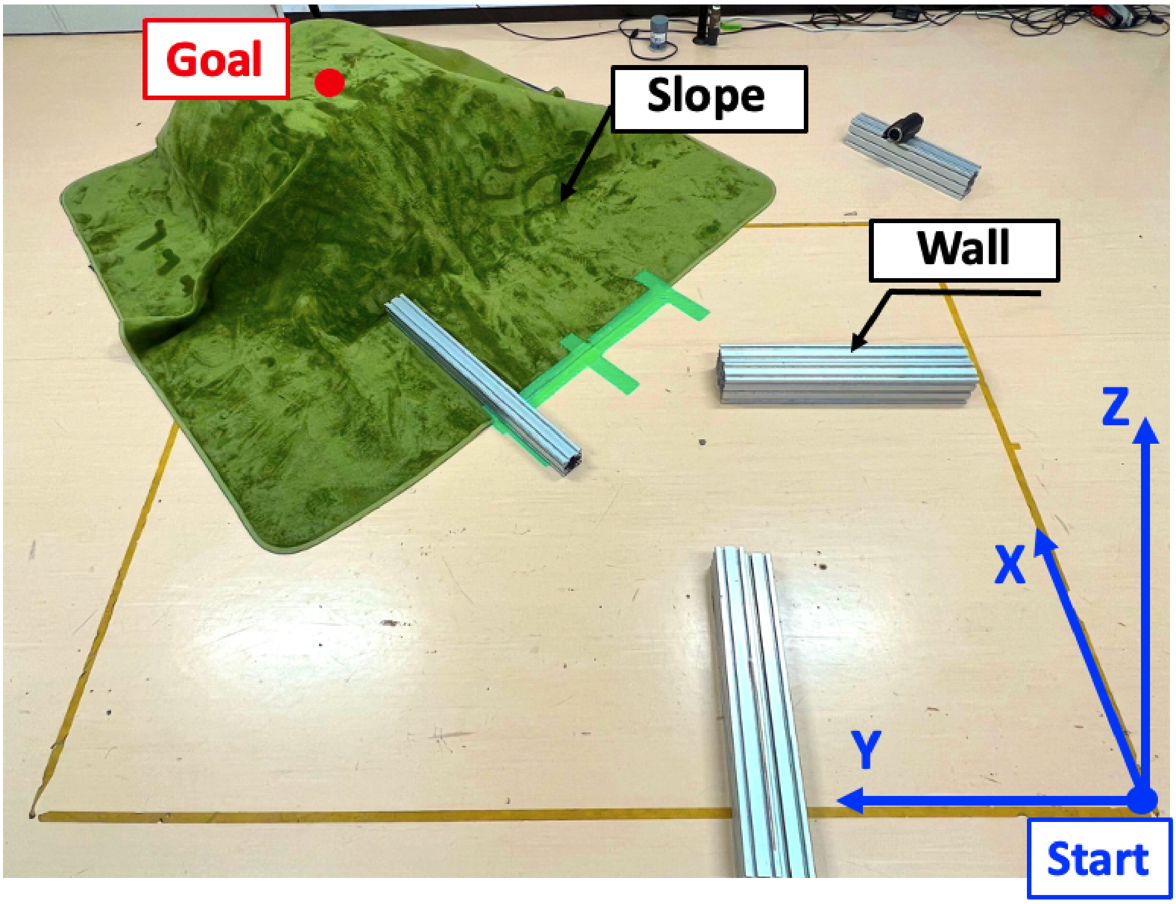


FIGURE 4.14: Problem statement for prototype experiment

4.3 Proposal of Robot Prototype for Experiment

In this section, building upon the previously mentioned robot simulation model, we constructed a prototype of the robot and conducted comparative experiments. These experiments validated the effectiveness of our proposed navigational robot in real-world environments and, through comparison, highlighted the importance and significance of our proposed stiffness control loop. Here, we will sequentially introduce the statement of the experimental navigation problem, the robot's structure, the control model, and the specific experimental content. It is worth mentioning that in the process of creating the prototype, we made certain adjustments and improvements to the robot's structure and control model compared to the simulation model. For example, in the prototype, we used wheels instead of legs for robot locomotion, and temporarily removed the design for reversing behavior in the control model, among other changes. For more detailed information and reasoning, please refer to Sections 4.3.2 and 4.3.3.

4.3.1 Problem Statement

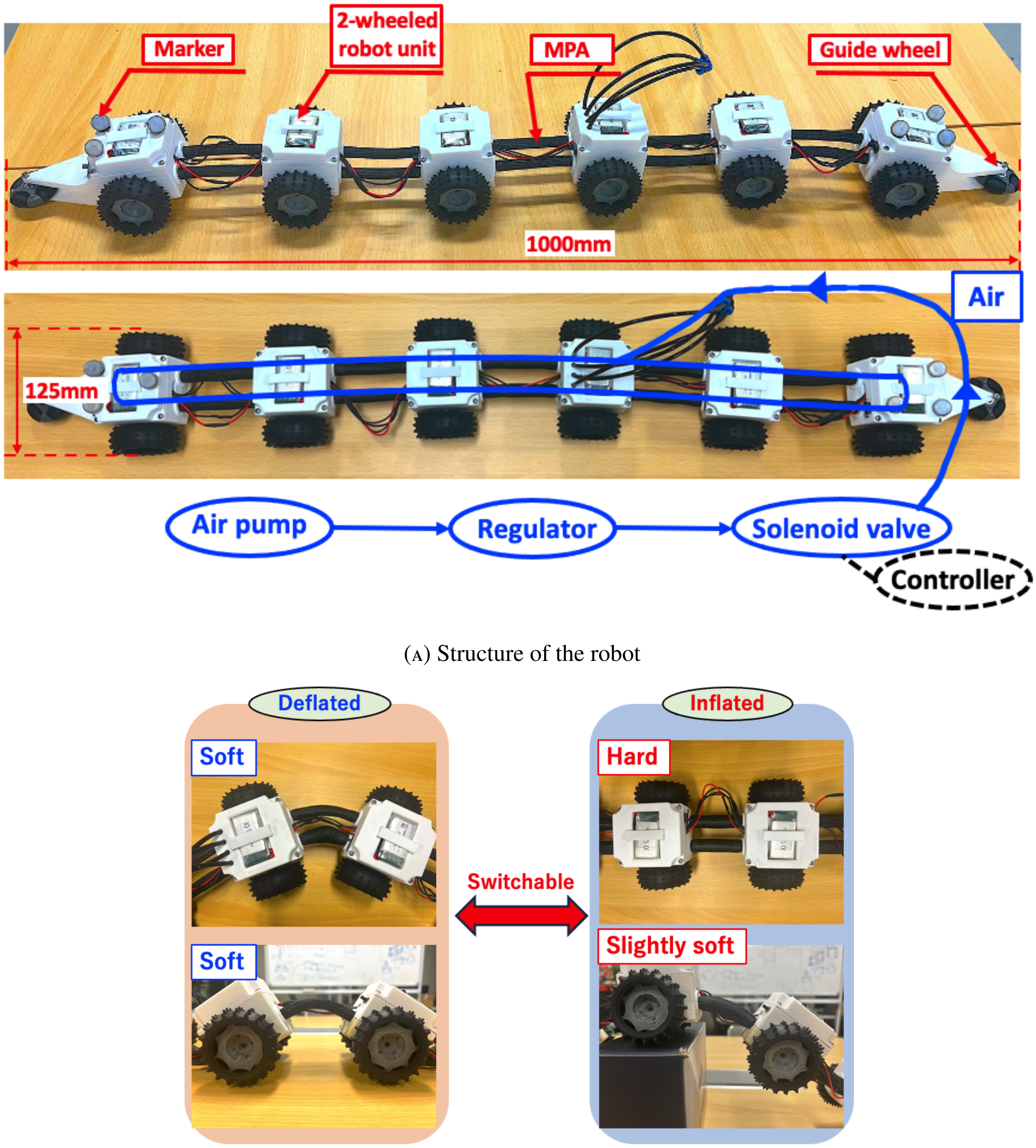
In this study, we aim to design robots capable of navigating through an unknown 3D environment composed of obstacles such as "walls" and "mountain terrains," as illustrated in

Fig. 4.14. Specifically, we establish a Cartesian coordinate system with the starting point at $[0, 0, 0]^T$ and the endpoint at $[1.72, 1.66, 0.45]^T$. Between the start and end points, there are "wall" obstacles made of rectangular metal tubes and "mountain terrain" obstacles resembling mountain terrain covered with carpet, with the endpoint located at the top of the mountain. The proposed robots are then placed at the starting point to conduct comparative navigation experiments using navigation control methods with and without stiffness control. If the robot using the proposed method successfully completes the navigation within a limited time, while the control group robot fails to do so, the effectiveness of the proposed navigation method is preliminarily proven.

4.3.2 Robot Structure Design

To achieve the research objectives, we have developed a robot named ICT-MPA-D, as shown in **Fig. 5.2(A)**. The ICT-MPA-D robot is 1000mm long and 125mm wide, consisting of a series of serially connected dual-wheel mobile robot units. Each unit comprises two wheels driven by 360-degree continuous rotation servo motors, capable of independently adjusting their speeds. When the robot needs to turn, the wheels of the first three dual-wheel robot units rotate in opposite directions, providing turning torque, while the wheels of the last three units rotate in the same direction, supplying the necessary propulsion for the turn, ultimately allowing the robot to move forward while turning. Additionally, the head and tail units of the robot are equipped with lateral guide wheels featuring multiple longitudinal rollers, enabling the robot to smoothly receive environmental guidance in both horizontal and vertical directions.

Regarding the connection between different robot units, we used two rows of parallel McKibben-type pneumatic actuator (MPA), which are artificial muscles that use compressed air to mimic the contraction and relaxation of real muscles, to sequentially connect these units. In this study, we discovered another characteristic of MPA: their stiffness increases with inflation and decreases with deflation, as shown in **Fig. 5.2(B)**. We exploited this feature by adjusting the air pressure in the MPAs to control their stiffness, allowing the robot to soften and accept guidance when interacting with favorable environments and to stiffen and resist unfavorable environmental interactions. This way, the robot can selectively utilize beneficial environmental interactions to aid in navigating unknown environments. In the specific scenarios addressed in this study, when the robot encounters insurmountable obstacles, such as walls, bypassing them is the optimal strategy. In these instances, the wall's impact and guidance are advantageous for the robot, leading it to become more flexible and compliant, thereby facilitating its movement around the obstacle. On the other hand, when the robot encounters a climbable slope, climbing over it is the preferred solution. Forces that might cause the robot to slide down are seen as adverse. To counteract these, the robot adjusts to become stiffer, enabling it to overcome these challenges and successfully navigate the slope. Moreover, to ensure that when the MPAs are inflated, the robot has enough horizontal stiffness to resist the detrimental effects that might cause sliding, while maintaining vertical flexibility to adapt to terrain changes, we adopted a design connecting robot units with two rows of parallel MPAs, as shown in **Fig. 5.2(B)**. This approach achieved anisotropic stiffness

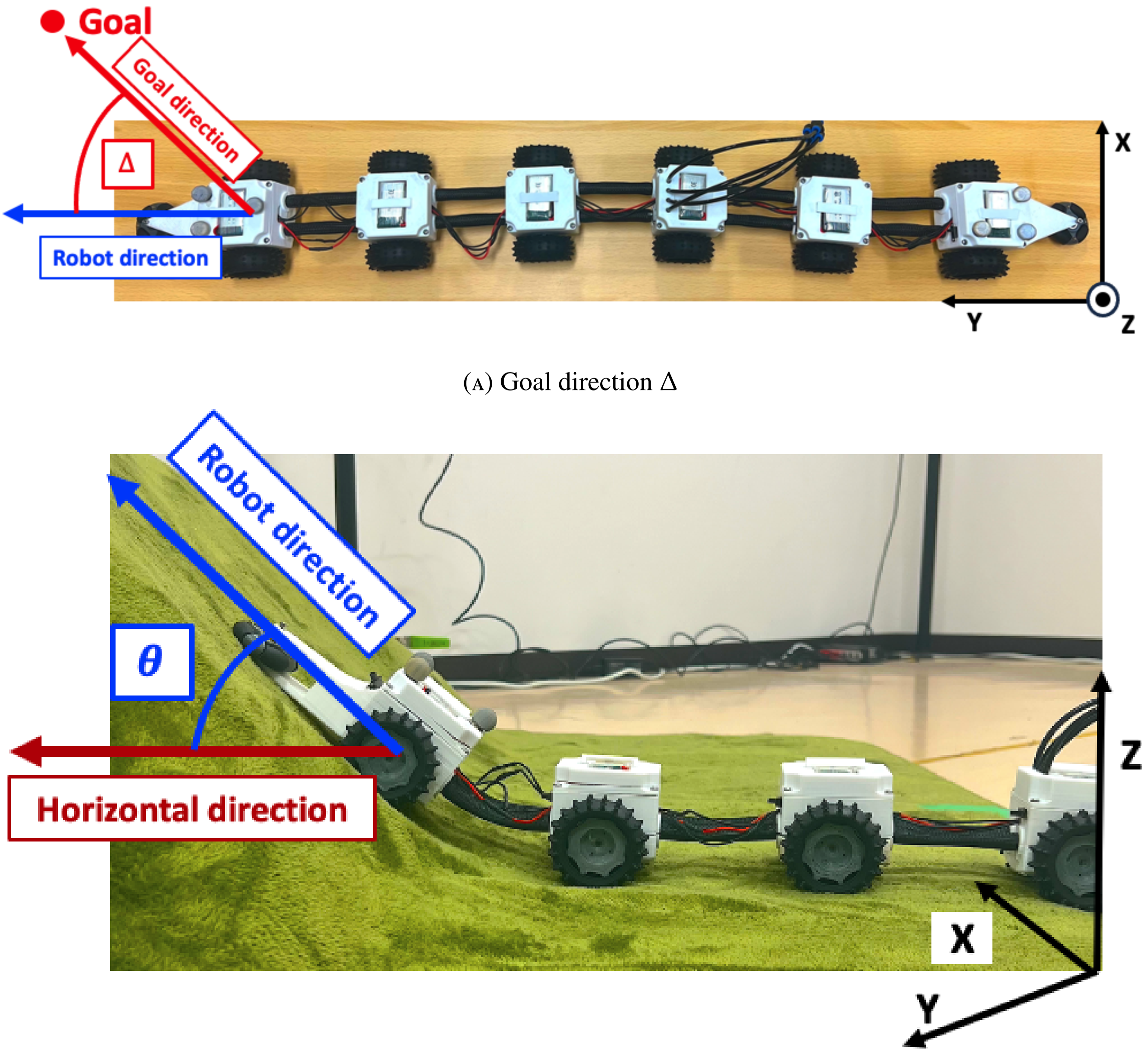


(A) Structure of the robot

(B) Stiffness switchability of MPA

FIGURE 4.15: Robot Structure Design

adjustment of the robot's trunk. Notably, in this study, all MPAs connecting the robot units are driven by the same air tube and connected to an external air pump, enabling synchronized inflation and deflation of all MPAs. The robot hardens corresponding to a constant inflation of 450kpa in all MPAs, and softens in a deflated state. And the process of inflation and deflation is directly controlled by solenoid valves connected to the controller.



(A) Goal direction Δ

(B) Pitch angle θ

FIGURE 4.16: Robot sensing input

4.3.3 Control and Sensing System Design

Hardware Design

Before delving into the hardware components of the control and sensing system, let's first introduce the only two input observation signals for the ICT-MPA-D robot.

The first signal is the angle between the robot's heading direction and the target direction within the horizontal plane, denoted as $\Delta \in \mathbb{R}$, as shown in **Fig. 4.16(A)**. The second signal is the robot head's pitch angle relative to the horizontal ground, represented by $\theta \in \mathbb{R}$, as illustrated in **Fig. 4.16(B)**. Here, the target direction is the only external sensory information, while the head posture can be measured by the robot's internal sensor, IMU. For details on how these two input signals are processed, please refer to the contents about the control model in the next subsection.

To obtain Δ and θ information and achieve navigation control, we designed the control

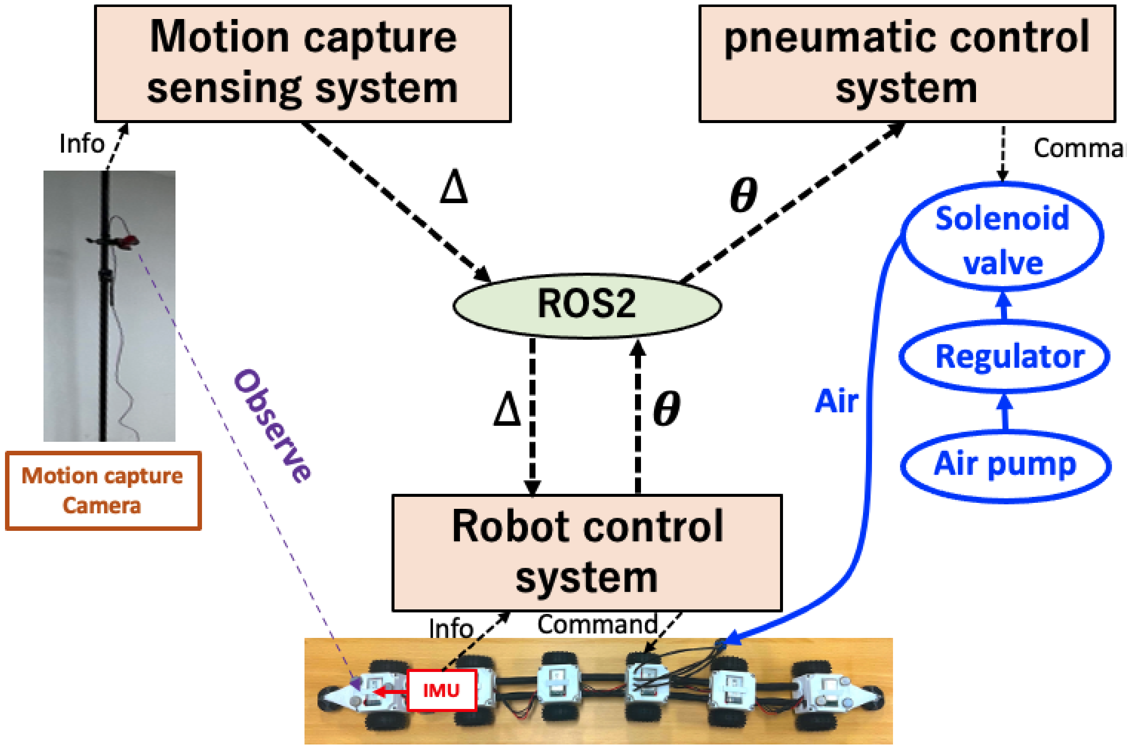
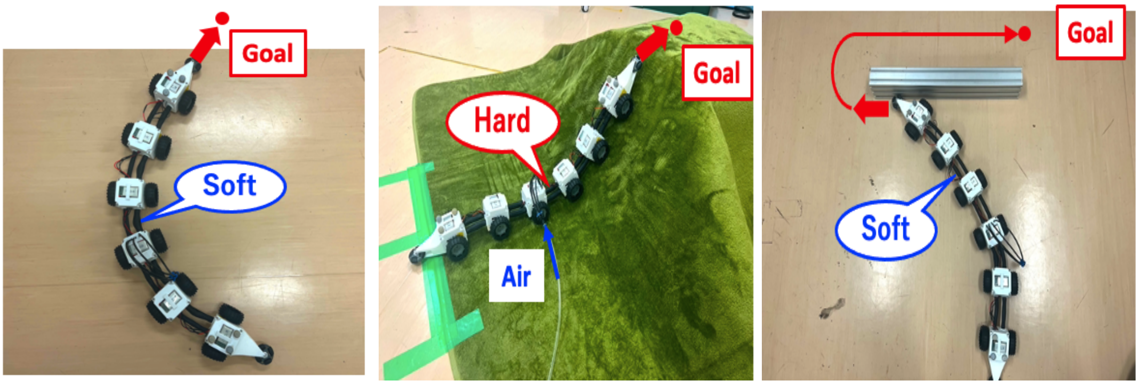


FIGURE 4.17: Control and sensing system

and sensing system as shown in **Fig. 4.17**. The ICT-MPA-D robot's control and sensing system broadly consists of three parts: the robot control system, the pneumatic control system, and the motion capture sensing system. These components are interconnected and communicate via ROS2, forming a complete control and sensing system. Specifically, the motion capture system captures the direction angle of the target relative to the robot (Δ) and sends this information to the robot control system via ROS2. The mobile robot control system then receives the Δ information through a Stamp-pico ESP32 controller over Wi-Fi, navigating towards the target based on the control model proposed in the next subsection. At the same time, the robot's head unit, equipped with an IMU posture sensor, sends the measured θ information to the pneumatic control system via ROS2. Finally, the pneumatic control system uses an ESP32 Dev controller to receive θ information and controls the opening and closing of solenoid valves based on this information, ultimately achieving control of inflation and deflation of MPAs.

Control Model Design

The control model in this study is based on a guiding principle described as follows: When the robot does not encounter any obstacles, it maintains a soft body and moves towards the target direction, as shown in **Fig. 4.18(A)**. Upon encountering an obstacle, if it is a climbable slope, the robot becomes rigid to counteract the environmental effect of "sliding down the slope." This allows it to climb the mountain slope with firm explicit control, as depicted in **Fig. 4.18(B)**. However, if the obstacle is an insurmountable slope or even a wall, the robot



(A) Goal following (B) Mountain climbing (C) Yielding

FIGURE 4.18: Control model of the prototype

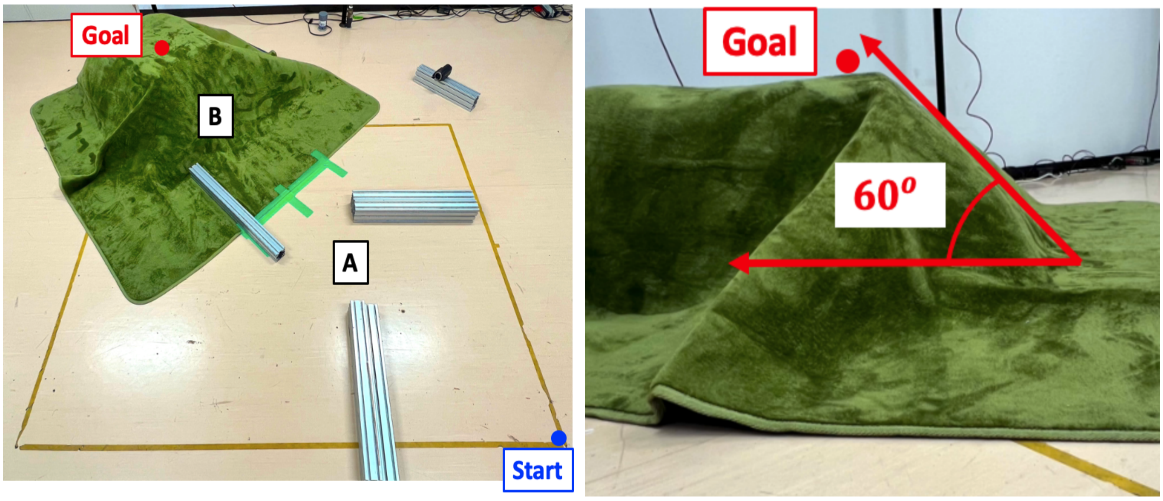
softens to yield to the guidance of the obstacle, thereby circumnavigating it along the edges using implicit control, as illustrated in **Fig. 4.18(C)**. To achieve this guiding principle, the robot's explicit control component of the control model is designed with two independent control loops: Functional control loop and stiffness control loop.

- **Function control loop:** The work of function control loop is guiding the robot towards the target. The design of this loop is fundamentally similar to the CWO navigation model in our previous research, with its mathematical expression provided below:

$$\begin{cases} \dot{P} = A(\Delta + \delta_1\varphi)vy\cos(\delta_1\varphi), \\ \dot{\alpha} = -K_1(\Delta) - \delta_1 K_2\varphi. \end{cases} \tag{4.3}$$

  , for details on the meanings of each symbol, please refer to reference [81][82]. The $-K_1(\Delta)$ term represents the explicit control part, while the $-\delta_1 K_2\varphi$ term represents the implicit control part. The values of K_1 and K_2 are directly related to the stiffness control loop. A notable difference from the method in [??] is that, given the focus of this study on adapting to different 3D terrains through stiffness modulation, the handling of complex 2D wall terrains is not a primary concern. Therefore, the reversing behavior and the input motor torque T required for it, mainly dealing with such situations, have been temporarily removed in this study.

- **Stiffness Control Loop:** The stiffness control loop primarily adjusts the robot's body stiffness by controlling the inflation and deflation of the MPAs. The guiding principle can be summarized as: harden when encountering a climbable slope and soften otherwise. Specifically, when the robot approaches a slope, its head is raised due to the terrain's influence, and the angle θ with the horizontal plane can be measured by the IMU sensor mounted on the robot's head. If we define a variable $K_i \in \mathbb{R}$ for the MPA inflation and deflation coefficient to represent the robot's stiffness state, where $K_i = 1$ represents inflation to 4.5kpa, making the robot rigid, and $K_i = 0$ represents a 0kpa state, making the robot soft, then the control of the robot's stiffness can be represented



(A) Overall setting

(B) Mountain slope

FIGURE 4.19: Experiment environment

as follows:

$$K_i = \begin{cases} 0 \quad , & if \ \ \theta < \theta_1 \ \ or \ \ \theta > \theta_2 \\ 1 \quad , & if \ \ \theta_1 \leq \theta \leq \theta_2 \end{cases} \tag{4.4}$$

Specifically, when θ is less than θ_1 or greater than θ_2, the robot is judged to be in an unobstructed or insurmountable obstacle state. Insurmountable obstacles can be further classified into wall obstacles (θ less than θ_1) and steep slopes (θ greater than θ_2). In these scenarios, the robot softens its body, reducing K_1 relative to K_2 in the formula, thereby yielding to the implicit control guidance given by the environment and navigating around the obstacle along its edges. When θ is greater than or equal to θ_1 and less than or equal to θ_2, the robot is judged to be in a climbable obstacle state, stiffening its body, thereby significantly increasing K_1 relative to K_2, allowing the robot to overcome the unfavorable environmental interactions and climb over the obstacle. In this study, θ_1 is empirically set to 10 degrees and θ_2 to 75 degrees.

Following this control model design for explicit control, beneficial implicit controls for the control objectives are also accordingly generated. This is specifically demonstrated as: when the robot encounters obstacles, depending on its current stiffness state, the robot will be guided by the environment to varying degrees.

4.3.4 Navigation experiment

To verify whether the navigation robot proposed in this study can achieve its set navigation objectives in an unknown 3D environment, we designed the following navigation experiment:

Experimental environment

We designed an experimental environment as shown in **Fig. 4.19(A)**, with its basic information available in Section 4.3.1. In this environment, the obstacle area the robot needs to traverse is divided into two zones, A and B, as illustrated. In zone A, the obstacles are insurmountable wall-type barriers, whereas zone B features climbable mountain obstacles with an angle of 60 degrees, as shown in **Fig. 4.19(B)**.

Experimental Content

This navigation experiment is divided into three groups: *I*, *II*, and *III*, for comparative analysis. Group *I* serves as a control group, where the robot's K_i is constantly set to 0, meaning the robot remains deflated and therefore soft-bodied. Group *II* is also a control group, but here the robot's K_i is constantly set to 1, keeping the robot inflated and thus rigid-bodied at all times. Group *III* is the experimental group, where the robot adjusts its stiffness according to the control model proposed in this study.

In each experimental group, we record the robot's movements with cameras from an oblique angle above. Additionally, the robot's movement trajectories are recorded using a motion capture system. In Group *III*'s experiment, we also track the variation of θ and K_i over time, characterizing the process of stiffness control.

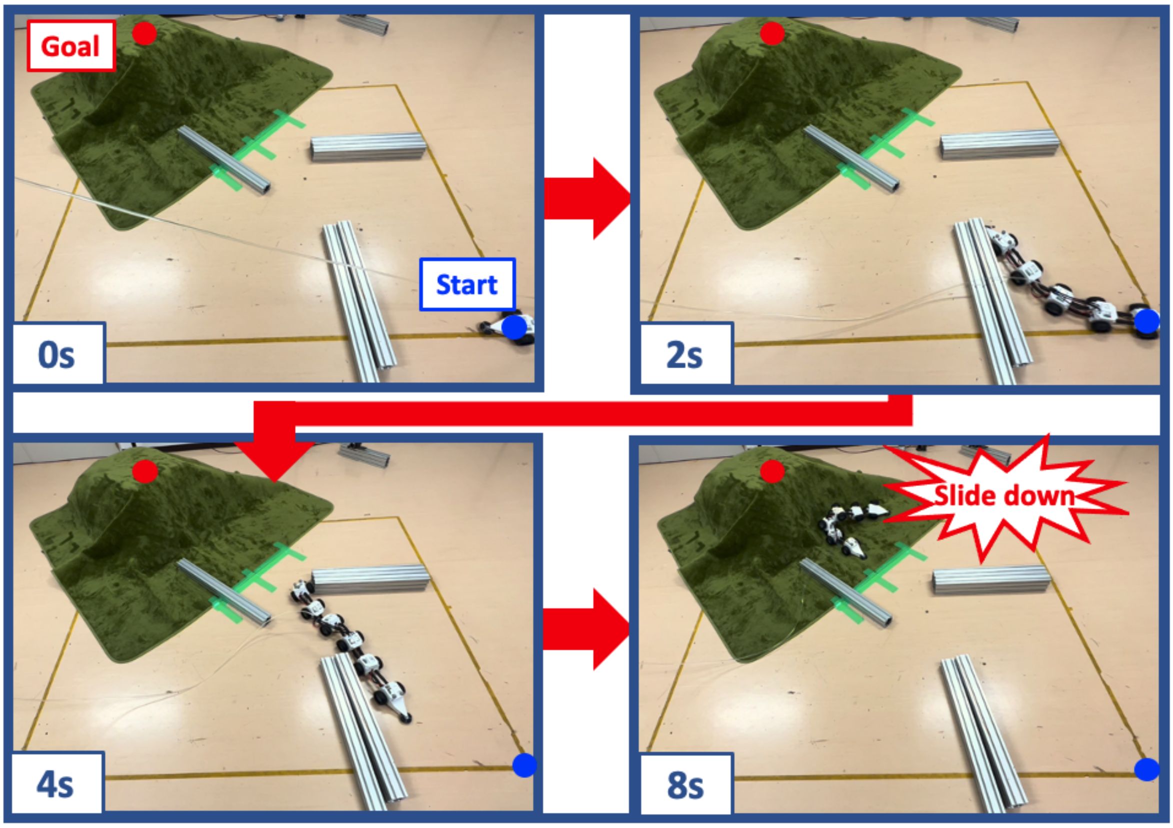
Experimental Results

The results of the three groups of navigation experiments are as follows:

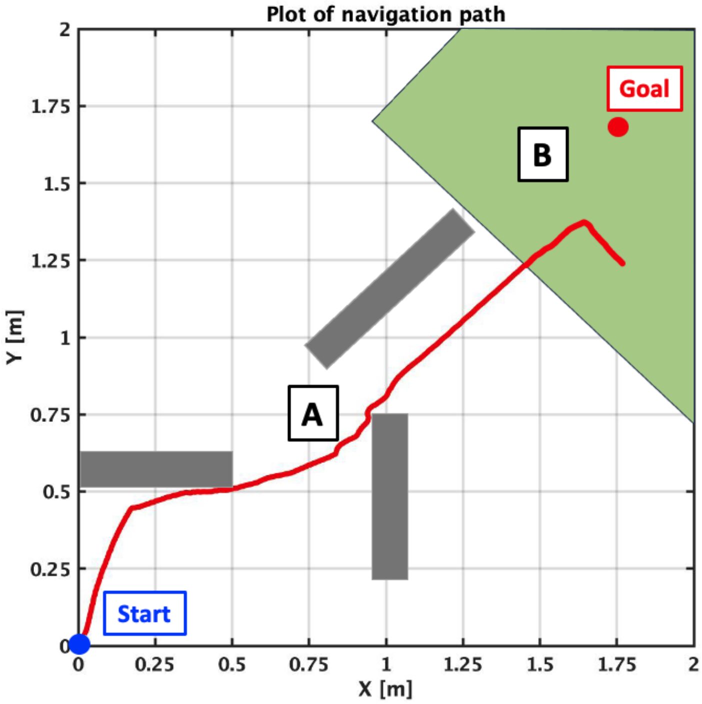
Group *I*: The navigation process of Group *I*'s robot, as shown in **Fig. 4.20(A)**, reveals that the robot in its soft state successfully navigates through the wall obstacle in Area A. Even when briefly stuck at a corner of the wall, the robot's soft body allows it to smoothly circumnavigate the obstacle. However, upon reaching the slope obstacle in Area B, the robot slides down the mountain slope and fails to reach the navigation target, resulting in a failed navigation attempt. The navigation path is depicted in **Fig. 4.20(B)**.

Group *II*: The navigation process for Group *II*'s robot, as illustrated in **Fig. 4.21(A)**, shows that the robot in its rigid state encounters a violent collision with the first wall obstacle in Area A, dislodging the barrier. When it reaches the second wall, the robot gets stuck at the corner, hindering further progress and ultimately failing to reach the navigation target. The navigation path is shown **Fig. 4.21(B)**.

Group *III*: The navigation process of Group *III*'s robot, as depicted in **Fig. 4.22(A)**, indicates that the robot enters Area A in a soft state and successfully navigates through it. Despite being momentarily trapped by a wall corner, the robot's flexibility allows it to smoothly bypass the wall obstacle. As the robot enters Area B at T0=6.69s, its head is raised, triggering MPA inflation and thus hardening the robot's body. With its stiffened body, the robot overcomes the environmental forces that would have caused it to slide down the mountain slope, successfully climbs the mountain, and reaches the navigation target at the top. The navigation is successful. The navigation path is illustrated in **Fig. 4.22(B)**. The

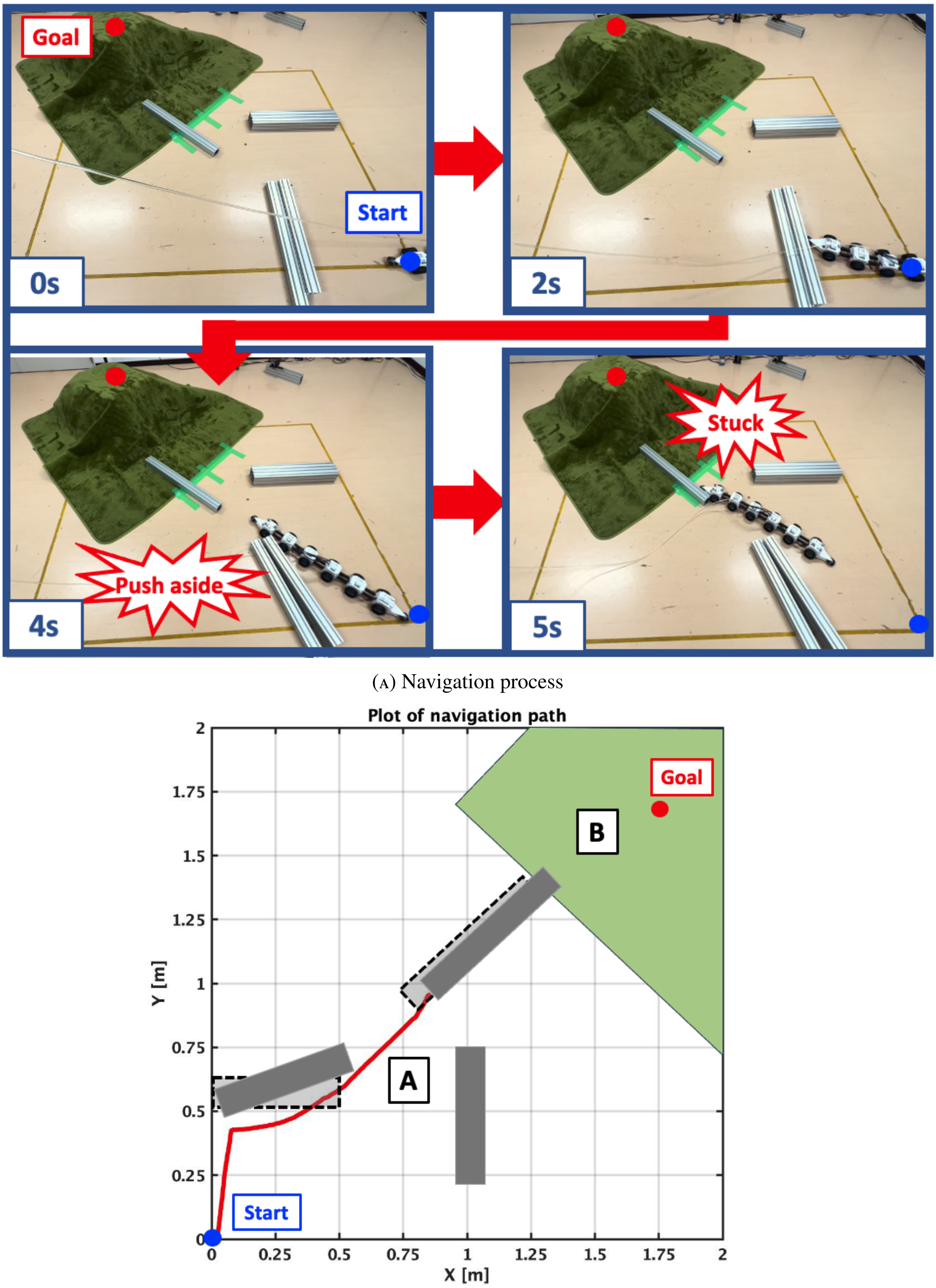


(A) Navigation process



(B) navigation path

FIGURE 4.20: Experiment result of Group *I*



(A) Navigation process

(B) navigation path

Figure 4.21: Experiment result of Group *II*

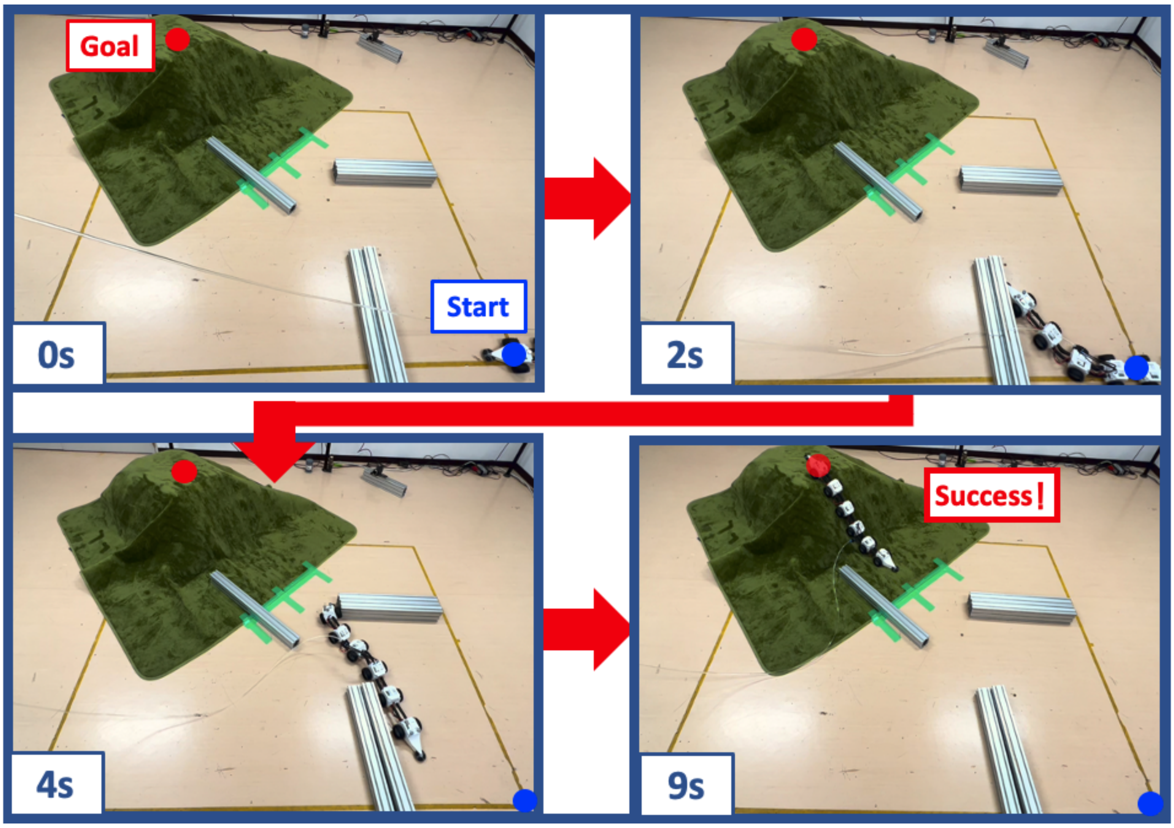
variation of θ and K_i over time, as shown in **Fig. 4.23**, indicates that the stiffness control loop functions as anticipated.

Discussion

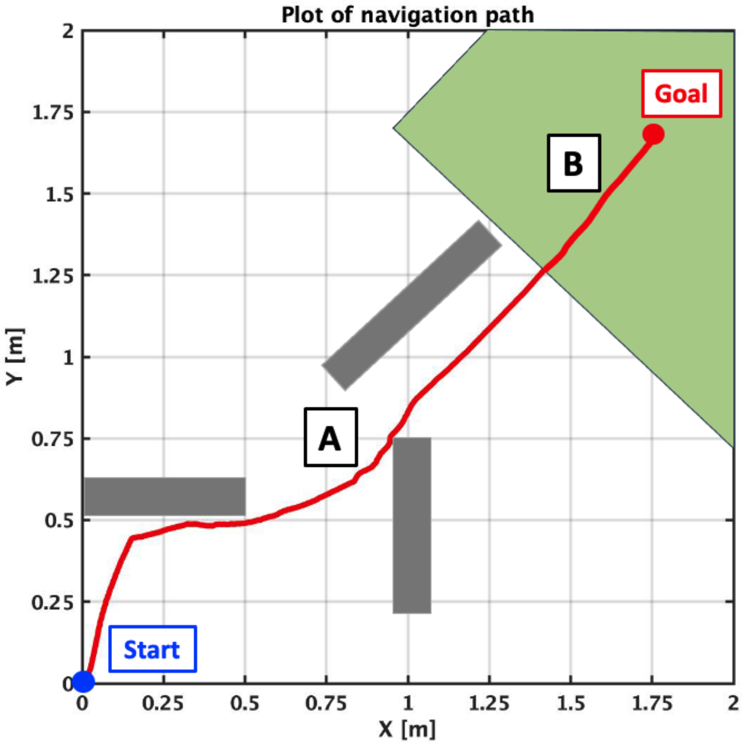
This comparative experiment preliminarily demonstrates the feasibility of the proposed navigation robot in navigating through an unknown 3D environment composed of walls and mountian terrains. Although the experimental results indicate that the navigation paths derived from this method are not the most optimal, the ability to successfully navigate in an unknown 3D environment without observing obstacle information or conducting path planning is in itself a significant achievement. Therefore, this study can be considered valuable and meaningful.

4.4 Conclusion

This study introduces a centipede robot simulation model and a wheel-based centipede navigation robot prototype that dynamically adjust its body stiffness to conform as much as possible to beneficial environmental influences and counteract detrimental ones respectively. This enables the robot to navigate in unknown 3D environments using only internal sensor information, target direction information, and a simple control system. The feasibility of the proposed robot and navigation method has been preliminarily demonstrated through simulation and comparative experiments. In the future, we plan to further explore the navigation performance of robots that can continuously control stiffness and those with independently controllable stiffness at different body parts.

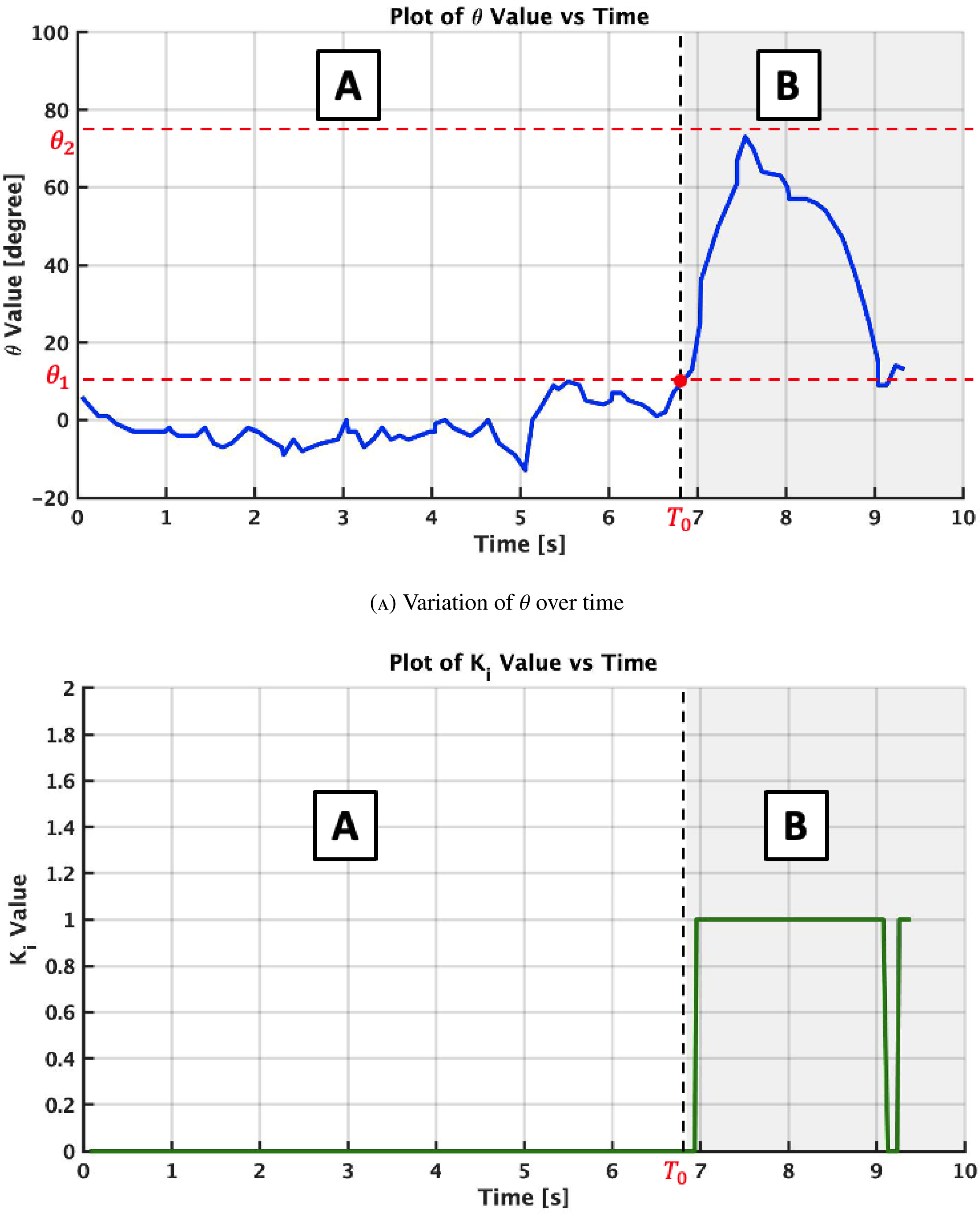


(A) Navigation process



(B) navigation path

FIGURE 4.22: Experiment result of Group *III* (A)



(A) Variation of θ over time

(B) Variation of K_i over time

FIGURE 4.23: Experiment result of Group *III* (B)

Chapter 5

General discussion

5.1 Summary of this thesis

So far, this study has introduced a novel navigation robot design principle, the “E-I Coordinated Robot Design Principle”, which treats the environment as an “assist” rather than an “obstacle”. Utilizing the guidance and assistance of the environment, this design principle enables navigation robots to navigate unknown environments using simpler systems while ensuring substantial adaptability. We have detailed the methodology and key points of this approach, and then, guided by this principle, designed several centipede-like navigation robots for individual and swarm navigation tasks in 2D unknown environments, as well as individual navigation tasks in 3D unknown environments. Through simulation and experimental validations, the advantages and feasibility of this design principle have been demonstrated using these robots as examples.

5.2 Future works

In the future, the first step will be to develop centipede-like navigation robots for swarm navigation tasks in 3D unknown environments, further expanding the range of environments and navigation tasks that our proposed centipede robots can handle, as shown in **Fig. 5.1**. At the same time, following the “E-I Coordinated Robot Design Principle,” we can also design a variety of new navigation robots with unique features, distinct from centipede robots, such as the sheet-type navigation robots and slime navigation robots as shown in **Fig. 5.2**. Lastly, as we accumulate more design experience with various types of navigation robots, we can, in turn, enrich and optimize the “E-I Coordinated Robot Design Principle” itself, allowing this design methodology to continuously evolve and improve.

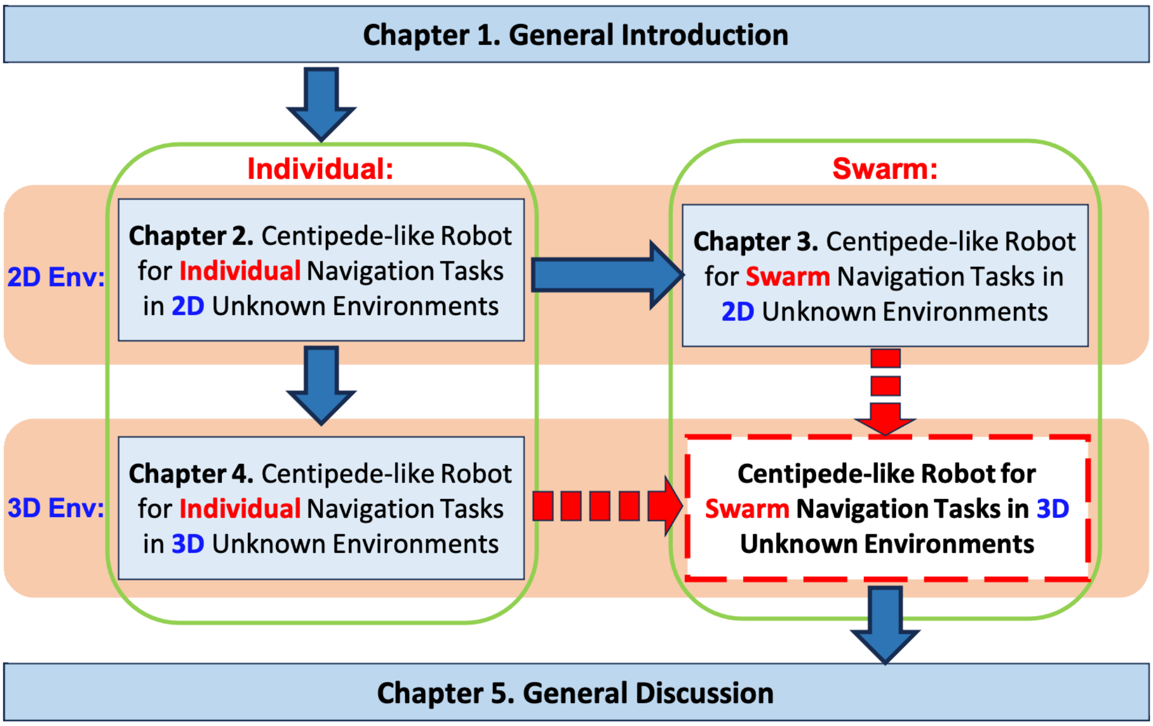
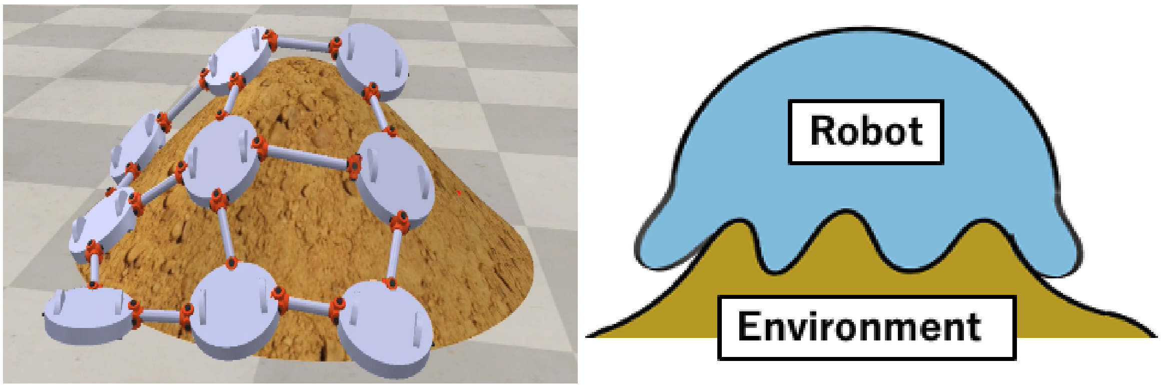

Chapter 1. General Introduction
Individual:
Swarm:
2D Env:
3D Env:
Chapter 2. Centipede-like Robot for Individual Navigation Tasks in 2D Unknown Environments
Chapter 3. Centipede-like Robot for Swarm Navigation Tasks in 2D Unknown Environments
Chapter 4. Centipede-like Robot for Individual Navigation Tasks in 3D Unknown Environments
Centipede-like Robot for Swarm Navigation Tasks in 3D Unknown Environments
Chapter 5. General Discussion

FIGURE 5.1: Next step of this research



(A) Sheet nvaigation robot

(B) Slime navigation robot

FIGURE 5.2: Other possible design examples using E-I coordinated robot design principle

Acknowledgements

I am deeply thankful to Prof. Koichi Osuka from Osaka University for his invaluable help in crafting this thesis. I also want to express my heartfelt gratitude to my co-supervisors, Prof. Masato Ishikawa and Prof. Mitsuru Higashimori, also from Osaka University, for their insightful guidance and steadfast support during this journey.

My sincere appreciation is extended to my co-researcher, Yusuke Tsunoda, whose collaboration and assistance have been crucial in my research. I'm also greatly indebted to Special Lecturer Keisuke Naniwa, Prof. Yasuhiro Sugimoto, Assistant Professor Yuichiro Sueoka, and Lecturer Hisashi Ishihara for their ongoing help and motivation in my academic pursuits.

I'd like to acknowledge my juniors and peers at the Osuka-Sugimoto Laboratory at Osaka University. Your dedication, encouragement, and backing, in spite of my shortcomings, have fueled my inspiration and energy in my everyday work.

Special thanks to Takahiro Goto and Kazuki Ito for their invaluable help and technical advice since I started my studies, which has been especially significant for me as an international student. I've gained so much from your support and mentorship.

Lastly, and with the utmost importance, I would like to express my heartfelt gratitude to my family. To my parents, Xiao Mingjun and Zhang Yuan, I owe a debt of gratitude that words can scarcely convey. Their unyielding support and sacrifice for my studies abroad have been the guiding force in my academic pursuit. In the same vein, my profound thanks go to my girlfriend, Ma Yihe. Her constant care, immeasurable patience, and unwavering support have been a beacon of hope and comfort in my journey.

Bibliography

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. “Attention is all you need”, *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. “Language models are unsupervised multitask learners”, *OpenAI Blog*, vol. 1, no. 8, pp. 9, 2019.

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. “Language models are few-shot learners”, *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. “BERT: Pre-training of deep bidirectional transformers for language understanding”, *Proceedings of NAACL-HLT*, vol. 1, pp. 2, 2019.

[5] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. “Deep speech: Scaling up end-to-end speech recognition”, *arXiv preprint arXiv:1412.5567*, 2014.

[6] Van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. “WaveNet: A generative model for raw audio”, *arXiv preprint arXiv:1609.03499*, 2016.

[7] Gatys, L. A., Ecker, A. S., & Bethge, M. “A neural algorithm of artistic style”, *arXiv preprint arXiv:1508.06576*, 2015.

[8] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. “Show and Tell: A Neural Image Caption Generator”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[9] Kunz, C., et al. “Deep sea underwater robotic exploration in the ice-covered Arctic Ocean with AUVs”, in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.

[10] Crisp, J. A., et al. “Mars Exploration Rover Mission”, Journal of Geophysical Research: Planets, vol. 108, no. E12, 2003.

[11] Kruijff, G.-J. M., et al. “Designing, developing, and deploying systems to support human–robot teams in disaster response”, Advanced Robotics, vol. 28, no. 23, pp. 1547-1570, 2014.

[12] Campbell, M., et al. “Autonomous driving in urban environments: approaches, lessons and challenges”, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 368, no. 1928, pp. 4649-4672, 2010.

[13] Forlizzi, J., & DiSalvo, C. “Service robots in the domestic environment: a study of the Roomba vacuum in the home”, in Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, 2006.

[14] Durrant-Whyte, H., & Bailey, T. “Simultaneous localization and mapping: part I”, *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[15] Bailey, T., & Durrant-Whyte, H. “Simultaneous localization and mapping (SLAM): Part II”, *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[16] Chong, T. J., et al. “Sensor technologies and simultaneous localization and mapping (SLAM)”, *Procedia Computer Science*, vol. 76, pp. 174–179, 2015.

[17] Smith, R., Self, M., & Cheeseman, P. “Estimating uncertain spatial relationships in robotics”, in *Autonomous Robot Vehicles*, Springer, New York, NY, pp. 167–193, 1990.

[18] Kurzer, K. “Path planning in unstructured environments: A real-time hybrid A* implementation for fast and deterministic path generation for the KTH Research Concept Vehicle”, 2016.

[19] Wang, H., Yu, Y., & Yuan, Q. “Application of Dijkstra algorithm in robot path-planning”, in *2011 Second International Conference on Mechanic Automation and Control Engineering*, pp. 1067–1069, 2011.

[20] Seif, R., & Oskoei, M. A. “Mobile robot path planning by RRT* in dynamic environments”, *International Journal of Intelligent Systems and Applications*, vol. 7, no. 5, pp. 24, 2015.

[21] Manikas, T. W., Ashenayi, K., & Wainwright, R. L. “Genetic algorithms for autonomous robot navigation”, *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 6, pp. 26–31, 2007.

[22] Blok, P. M., van Boheemen, K., van Evert, F. K., IJsselmuiden, J., & Kim, G.-H. “Robot navigation in orchards with localization based on Particle filter and Kalman filter”, *Computers and Electronics in Agriculture*, vol. 157, pp. 261–269, 2019.

[23] Balasuriya, B. L. E. A., Chathuranga, B. A. H., Jayasundara, B. H. M. D., Napagoda, N. R. A. C., Kumarawadu, S. P., Chandima, D. P., & Jayasekara, A. G. B. P. “Outdoor robot navigation using Gmapping based SLAM algorithm”, in *2016 Moratuwa Engineering Research Conference (MERCon)*, pp. 403–408, 2016.

[24] Song, K.-T., Chiu, Y.-H., Kang, L.-R., Song, S.-H., Yang, C.-A., Lu, P.-C., & Ou, S.-Q. “Navigation control design of a mobile robot by integrating obstacle avoidance and LiDAR SLAM”, in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1833–1838, 2018.

[25] Kohlbrecher, S., et al. “Hector open source modules for autonomous mapping and navigation with rescue robots”, in RoboCup 2013: Robot World Cup XVII, vol. 17, Springer Berlin Heidelberg, 2014.

[26] Siemiątkowska, B., Szklarski, J., & Gnatowski, M. “Mobile robot navigation with the use of semantic map constructed from 3D laser range scans”, *Control and Cybernetics*, vol. 40, no. 2, pp. 437–453, 2011.

[27] Zhang, G., Zhang, Z., Xia, Z., Dai, M., Peng, M., & Cen, J. “Implementation and research on indoor mobile robot mapping and navigation based on RTAB-Map”, in *2022 28th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pp. 1–6, 2022.

[28] Chang, W.-C., & Ling, H.-C. “Visual environment mapping for mobile robot navigation”, in *10th France-Japan Congress, 8th Europe-Asia Congress on Mecatronics, MECATRONICS 2014*, pp. 244–249, 2015.

[29] Batta, M. “Machine learning algorithms-a review”, International Journal of Science and Research (IJSR), vol. 9, no. 1, pp. 381-386, 2020.

[30] Souza, J.R., Pessin, G., Osório, F.S., Wolf, D.F. (2011). Vision-Based Autonomous Navigation Using Supervised Learning Techniques. In: Iliadis, L., Jayne, C. (eds) *Engineering Applications of Neural Networks. EANN AIAI 2011 2011. IFIP Advances in Information and Communication Technology*, vol 363. Springer, Berlin, Heidelberg.

[31] Hirose, N., Sadeghian, A., Goebel, P., & Savarese, S. “To Go or Not To Go? A Near Unsupervised Learning Approach For Robot Navigation”, 2017.

[32] Baleia, J., Santana, P. & Barata, J. On Exploiting Haptic Cues for Self-Supervised Learning of Depth-Based Robot Navigation Affordances. *J Intell Robot Syst* 80, 455–474, 2015.

[33] Sofman, B., Lin, E., Bagnell, J., Cole, J., Vandapel, N., & Stentz, A. “Improving robot navigation through self‐supervised online learning”, *Journal of Field Robotics*, vol. 23, pp. 1059–1075, 2006.

[34] Kahn, G., Abbeel, P., & Levine, S. “BADGR: An Autonomous Self-Supervised Learning-Based Navigation System”, *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[35] Kouris, A., & Bouganis, C.-S. “Learning to Fly by MySelf: A Self-Supervised CNN-Based Approach for Autonomous Navigation”, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[36] Mirowski, P. W., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., & Hadsell, R. “Learning to Navigate in Complex Environments”, *arXiv preprint arXiv:1611.03673*, 2016.

[37] Xie, L., Wang, S., Markham, A., & Trigoni, N. "Towards Monocular Vision based Obstacle Avoidance through Deep Reinforcement Learning", 2017.

[38] Tai, L., Paolo, G., & Liu, M. "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation", in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 31–36, 2017.

[39] Surmann, H., Jestel, C., Marchel, R., Musberg, F., Elhadj, H., & Ardani, M. "Deep Reinforcement Learning for Real Autonomous Mobile Robot Navigation in Indoor Environments", *arXiv preprint arXiv:2005.13857*, 2020.

[40] Wang, D., Zhang, Y., & Si, W. "Behavior-based hierarchical fuzzy control for mobile robot navigation in dynamic environment", in *2011 Chinese Control and Decision Conference (CCDC)*, pp. 2419–2424, 2011.

[41] Taliansky, A., & Shimkin, N. "Behavior-based navigation for an indoor mobile robot", in *21st IEEE Convention of the Electrical and Electronic Engineers in Israel. Proceedings*, pp. 281–284, 2000.

[42] Arkin, R. C. "Behavior-Based Robot Navigation for Extended Domains", *Adaptive Behavior*, vol. 1, pp. 201–225, 1992.

[43] Lumelsky, V., & Stepanov, A. "Dynamic path planning for a mobile automaton with limited information on the environment", *IEEE Transactions on Automatic Control*, vol. 31, no. 11, pp. 1058–1063, 1986.

[44] Cohen, W. W., et al. "Adaptive mapping and navigation by teams of simple robots", *Robotics and Autonomous Systems*, vol. 18, no. 4, pp. 411–434, 1996.

[45] Payton, D., et al. "Pheromone robotics", *Autonomous Robots*, vol. 11, pp. 319-324, 2001.

[46] Wurr, A., & Anderson, J. "Multi-agent trail making for stigmergic navigation", *Advances in Artificial Intelligence*, pp. 422–428, 2004.

[47] Hayes, A. T., & Dormiani-Tabatabaei, P. "Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots", in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3900-3905, 2002.

[48] Cezayirli, A., & Kerestecioğlu, F. "Navigation of non-communicating autonomous mobile robots with guaranteed connectivity", *Robotica*, vol. 31, no. 5, pp. 767-776, 2013.

[49] Merheb, A.-R., Gazi, V., & Sezer-Uzol, N. "Implementation studies of robot swarm navigation using potential functions and panel methods", *IEEE/ASME Transactions On Mechatronics*, vol. 21, no. 5, pp. 2556-2567, 2016.

[50] Silva Junior, L., & Nedjah, N. "Efficient Strategy for Collective Navigation Control in Swarm Robotics", *Procedia Computer Science*, vol. 80, pp. 814-823, 2016.

[51] Quesada, W. O., et al. “Leader-follower formation for UAV robot swarm based on fuzzy logic theory”, in Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, Proceedings, Part II, vol. 17, Springer International Publishing, 2018.

[52] Quesada, W. O., et al. “Leader-follower formation for UAV robot swarm based on fuzzy logic theory”, in Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, Proceedings, Part II, vol. 17, Springer International Publishing, 2018.

[53] Qiao, Z., et al. “Dynamic self-organizing leader-follower control in a swarm mobile robots system under limited communication”, IEEE Access, vol. 8, pp. 53850-53856, 2020.

[54] Tsunoda, Y., et al. “Sheepdog-type robot navigation: Experimental verification based on a linear model”, in 2020 IEEE/SICE International Symposium on System Integration (SII), 2020.

[55] Vaughan, R., et al. “Robot sheepdog project achieves automatic flock control”, in Proceedings of the Fifth International Conference on the Simulation of Adaptive Behaviour, vol. 489, 1998.

[56] Sueoka, Y., Ishitani, M., & Osuka, K. “Analysis of sheepdog-type robot navigation for goal-lost-situation”, Robotics, vol. 7, no. 2, 21, 2018.

[57] Hussein, S. K., & Al-Mutairi, M. A. “A Novel Prototype Model for Swarm Mobile Robot Navigation Based Fuzzy Logic Controller”, *International Journal of Computer Science And Information Technology (IJCSIT)*, vol. 11, no. 2, 2019.

[58] Agrawal, A., Sudheer, A. P., & Ashok, S. “Ant colony based path planning for swarm robots”, in *Proceedings of the 2015 Conference on Advances In Robotics*, pp. 1-5, 2015.

[59] Lee, Y.-W. “Automatic mutual localization of swarm robot using a particle filter”, *Journal of Information and Communication Convergence Engineering*, vol. 10, no. 4, pp. 390-395, 2012.

[60] Wang, X., Sun, S., Li, T., & Liu, Y. “Fault tolerant multi-robot cooperative localization based on covariance union”, *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7799-7806, 2021.

[61] Fox, D., Burgard, W., Kruppa, H., & Thrun, S. “A probabilistic approach to collaborative multi-robot localization”, *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[62] Majmudar, T., Keaveny, E. E., Zhang, J., & Shelley, M. J. “Experiments and theory of undulatory locomotion in a simple structured medium”, *Journal of The Royal Society Interface*, vol. 9, no. 73, pp. 1809–1823, 2012. DOI: 10.1098/rsif.2011.0856.

[63] Campbell, E. J., & Bagchi, P. “A computational model of amoeboid cell motility in the presence of obstacles”, Soft Matter, vol. 14, no. 28, pp. 5741–5763, 2018.

[64] Szabó-Patay, J. “A kapus-mangay”, *Természettudományi Közlöny*, Budapest, pp. 215-219, 1928.

[65] Lindauer, M. L. “Schwarmbienen auf Wohnungssuche”, *Z. vergl. Physiol.*, vol. 37, pp. 263-324, 1955.

[66] Anderson, C., Theraulaz, G., & Deneubourg, J.-L. “Self-assemblages in insect societies”, *Insectes sociaux*, vol. 49, no. 2, pp. 99-110, 2002.

[67] Franks, N. R., & Hölldobler, B. “Sexual competition during colony reproduction in army ants”, *Biol. J. Linn. Soc.*, vol. 30, pp. 229-243, 1987.

[68] Jaffe, K. “Surfing ants”, *Fla. Entomol.*, vol. 76, pp. 182-183, 1993.

[69] Savage, T. S. “On the habits of the ’drivers’ or visiting ants of West Africa”, *Trans. R. Entomol. Soc. Lond.*, vol. 5, pp. 1-15, 1847.

[70] Pfeifer, Rolf, and Josh Bongard. How the body shapes the way we think: a new view of intelligence. MIT press, 2006.

[71] Kano, T., Yoshizawa, R., & Ishiguro, A. “Tegotae-based decentralised control scheme for autonomous gait transition of snake-like robots”, *Bioinspiration & Biomimetics*, vol. 12, no. 4, 046009, 2017.

[72] Sato, T., Kano, T., & Ishiguro, A. “A decentralized control scheme for an effective coordination of phasic and tonic control in a snake-like robot”, *Bioinspiration & Biomimetics*, vol. 7, no. 1, 016005, 2011.

[73] Greer, J. D., Blumenschein, L. H., Okamura, A. M., et al. “Obstacle-aided navigation of a soft growing robot”, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4165-4172, 2018.

[74] Greer, J. D., Blumenschein, L. H., Alterovitz, R., et al. “Robust navigation of a soft growing robot by exploiting contact with the environment”, *The International Journal of Robotics Research*, vol. 39, no. 14, pp. 1724-1738, 2020.

[75] K.Osuka, A.Ishiguro, X.Zheng, Y.Sugimoto and D.Owai:Implicit Control Law Embeded in Control System Solves Problem of Adaptive Function!?, J.of Robotic Society of Japan, Vol.28, No.4, 491/502,2010(in Japanese).

[76] Osuka, K., Ishiguro, A., Zheng, X., Sugimoto, Y., & Owai, D. “Dual structure of Mobiligence—Implicit Control and Explicit Control—”, *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (2010).*

[77] Osuka, K., Kinugasa, T., Hayashi, R., et al. “Centipede Type Robot i-CentiPot: From Machine to Creatures”, *Journal of Robotics and Mechatronics*, vol. 31, no. 5, pp. 723-726, 2019.

[78] Kinugasa, T., Miyamoto, N., Osuka, K., et al. “Myriapod Robot i-CentiPot via passive dynamics—emergence of various locomotions for foot movement”, *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 7-9, 2017.

[79] Rohmer, E., Singh, S. P. N., & Freese, M. “V-REP: A versatile and scalable robot simulation framework”, in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.

[80] Xiao, R., Tsunoda, Y., & Osuka, K. “A New Robot Navigation Scheme Combining Implicit and Explicit control to Travel Through the Unknown Environment to Reach the Destination”, *21st SI Division Conference of the Society of Instrument and Control Engineers (SI2020)*, 2020.

[81] Xiao, R., Tsunoda, Y., & Osuka, K. “Stability Analysis of Homing Control of a Mobile Robot.”, *SICE DAS Symposium*, 2021 (in Japanese).

[82] Xiao, R., Tsunoda, Y., & Osuka, K. “Proposal and Experimental Verification of an Implicit Control Based Navigation Scheme in Unknown Environment for a Centipede Type Robot”, *Journal of Robotics and Mechatronics*.

[83] Xiao, R., Tsunoda, Y., & Osuka, K. “’Unfavorable’ environmental effects can also be beneficial: a simulation analysis of centipede-like swarm robots based on implicit control for navigation and exploration in unknown environments”, Artificial Life Robotics, vol. 28, pp. 690–702, 2023.

[84] Xiao, R., Tsunoda, Y., & Osuka, K. “Proposal and Simulation Analysis of a Centipede Type Robot Dynamically Adjusting Body Stiffness to Harness Favorable Environmental Effects for Navigation in Unknown 3D Environments”, 2023 2nd International Conference on Automation, Robotics and Computer Engineering (ICARCE 2023).

[85] Xiao, R., Tsunoda, Y., & Osuka, K. “Proposal and Experimental Validation of a Wheeled Centipede-inspired Navigation Robot in 3D Unknown Environments: Harnessing Environmental Interactions by Modulating Body Rigidity”, 29th International Symposium on Artificial Life and Robotics (AROB 29th).

Appendix A

.1 Homing Control of a Mobile Robot in an Obstacle-free Environment

The focus of this section is the derivation of the mathematical model for the “Control without Overdoing ” scheme in an obstacle-free environment. The navigation control model with obstacle conditions presented in Section 2.4 is modified based on the model in this chapter.

.1.1 Problem Setting

As shown in the **Fig. 3**, in a plane coordinate system, we are trying to control the mobile robot back to the origin position $[0,0]^{\mathrm{T}}$, which is called “Homing Control” in this section. Actually, the goal of the robot can be set to any point in the plane coordinate system in the homing control. But in this section, the goal is set to the origin for ease of analysis. At time $t(t \geq 0)$, the position vector, propulsive velocity and steering angular velocity of the mobile robot is defined as $\boldsymbol{P}(t) = [P_x(t), P_y(t)]^{\mathrm{T}} \in \mathbb{R}^2$, $\boldsymbol{f}(t) \in \mathbb{R}^2$ and $\tau(t) \in \mathbb{R}$ respectively.

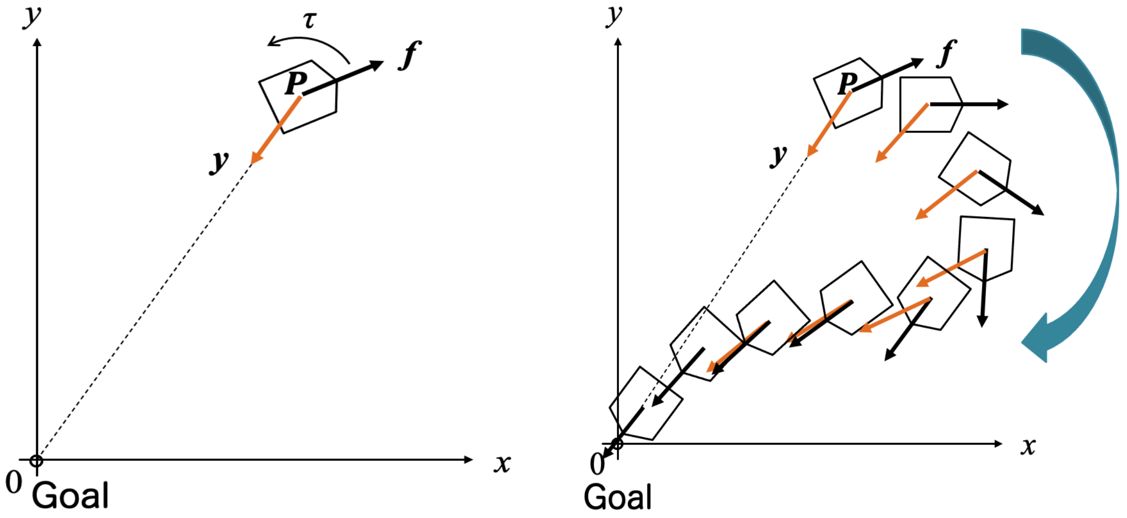
In the process of navigation, there are 4 key rules as follows:

- The robot moves based on the two-wheeled vehicle model. $\boldsymbol{f}$ is the propulsive velocity in the straight ahead direction and τ is the angular velocity in the turning direction.
- The robot knows the direction of the propulsive velocity $\boldsymbol{f}$ and the direction of the origin $\boldsymbol{y}(t) \in \mathbb{R}^2$. The $\boldsymbol{y}$ is a vector of $||\boldsymbol{y}|| = ||\boldsymbol{f}||$ pointing from the robot to the origin.
- The robot does not know its position in the area outside the neighborhood of the origin. In this case, $||\boldsymbol{y}|| = ||\boldsymbol{f}|| = 1$. While in the neighborhood of the origin, the robot knows its position, and the magnitudes of $\boldsymbol{f}$ and $\boldsymbol{y}$ gradually decrease along the direction closer to the origin.
- $\tau(t) \in \mathbb{R}$ brings the direction of the robot closer to $\boldsymbol{y}$.

.1.2 Modeling of Homing Control

As shown in **Fig. 4(a)**, $r(t) \in \mathbb{R}_+$ represents the radial distance of the robot from the origin and $\theta(t) \in \mathbb{R}$ represents the angle between the radial direction and the positive direction of the x-axis. The position of the robot $\boldsymbol{P}$ can be expressed as follows:

$$\boldsymbol{P} = \begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}. \tag{1}$$



(A) Initial state

(B) Control purpose

FIGURE 3: Problem setting

And the time evolution equations for the position and orientation of the robot are expressed as follows:

$$\begin{cases} \dot{\boldsymbol{P}} = \boldsymbol{f}, \\ \dot{\alpha} = \tau, \end{cases} \tag{2}$$

Here, $\alpha(t) \in \mathbb{R}$ is the angle between $\boldsymbol{f}$ and the positive direction of the x-axis. And the angle between $\boldsymbol{f}$ and $\boldsymbol{y}$ is defined as $\Delta(t) \in \mathbb{R}$, which has the following relationship with θ and α:

$$\Delta = \pi + \alpha - \theta, \left(\left|\Delta(0)\right| \leq \pi\right). \tag{3}$$

Then, the present navigation problem can be summarized as follows:

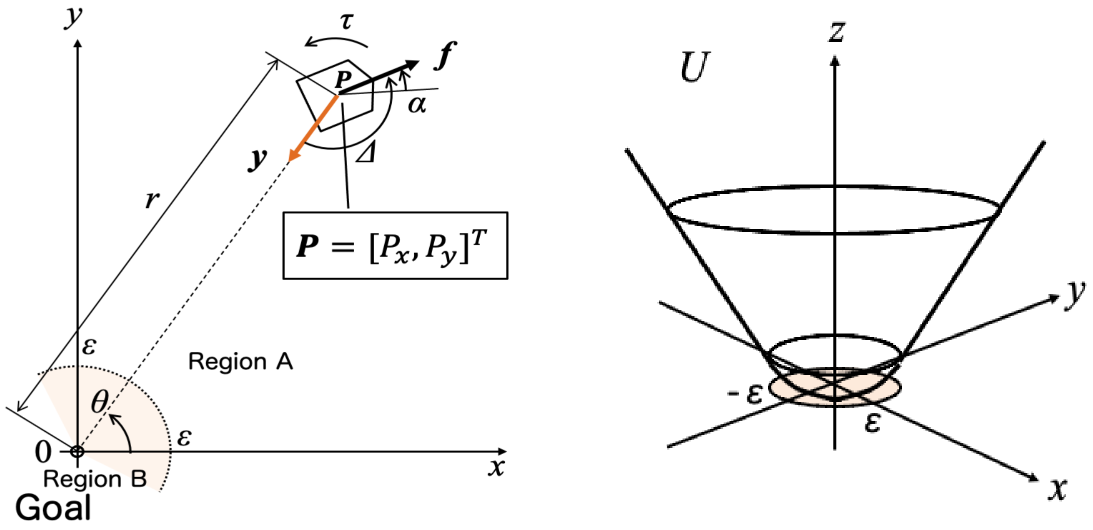
- For a mobile robot placed at any position in any orientation on the x-y plane, find the $\boldsymbol{f}$ and τ that induce the robot toward the target by observing Δ.

In order to achieve homing control of the robot, $\boldsymbol{f}$ and τ are designed correspondingly in two regions: the neighborhood of the origin and the rest of the region, respectively. First, we divide the navigation region into the following two regions.

$$\text{Region A} = \left\{[x, y]^{\mathrm{T}} | x^2 + y^2 \geq \varepsilon\right\}, \tag{4}$$

$$\text{Region B} = \left\{[x, y]^{\mathrm{T}} | x^2 + y^2 < \varepsilon\right\}, \tag{5}$$

where $\varepsilon \in \mathbb{R}$ is the radius of region A.



(A) Region devision

(B) Potential field

FIGURE 4: Mathematical model

Next, for each region, we design $\boldsymbol{f}$ firstly. And in order to design $\boldsymbol{f}$, we can first define $\boldsymbol{y}$ and get $\boldsymbol{f}$ by rotating it by Δ. Therefore, we first construct the potential field U to define $\boldsymbol{y}$.

$$U = \begin{cases} \sqrt{P_x^2 + P_y^2} - \dfrac{1}{2}\varepsilon \text{ (Region A)}, \\ \dfrac{1}{2\epsilon}(P_x^2 + P_y^2) \quad \text{ (Region B)}. \end{cases} \tag{6}$$

The schematic diagram of the potential field is shown in **Fig. 4(b)**. Then, the gradient of this potential field can be expressed as:

$$\frac{\partial U}{\partial \boldsymbol{P}} = \begin{bmatrix} \frac{\partial U}{\partial P_x} \\ \frac{\partial U}{\partial P_y} \end{bmatrix} = \begin{cases} \begin{bmatrix} \frac{P_x}{\sqrt{P_x^2+P_y^2}} \\ \frac{P_y}{\sqrt{P_x^2+P_y^2}} \end{bmatrix} = \dfrac{\boldsymbol{P}}{||\boldsymbol{P}||} \text{ (Region A)}, \\ \begin{bmatrix} \frac{1}{\varepsilon}P_x \\ \frac{1}{\varepsilon}P_y \end{bmatrix} = \dfrac{1}{\varepsilon}\boldsymbol{P} \qquad \text{ (Region B)}, \end{cases} \tag{7}$$

As shown in **Fig. 4(b)**, in region A, the gradient pointing to the origin direction is constantly equal to 1, while in region B, the gradient pointing to the origin direction is linearly varying. Then, we assign the result of this gradient calculation to $\boldsymbol{y}$:

$$\boldsymbol{y} = -\frac{\partial U}{\partial \boldsymbol{P}} = \begin{cases} -\dfrac{\boldsymbol{P}}{||\boldsymbol{P}||} \text{ (Region A)}, \\ -\dfrac{1}{\varepsilon}\boldsymbol{P} \text{ (Region B)}. \end{cases} \tag{8}$$

In this way, $\boldsymbol{y}$ is a unit vector pointing to the origin in region A, while in region B it is a vector whose direction points to the origin and decreases in size as the position radius r becomes smaller.

Based on the above conditions, We define $\boldsymbol{f}$ as follows：

$$\boldsymbol{f} = \boldsymbol{A}(\Delta)\boldsymbol{y}. \tag{9}$$

Here, $\boldsymbol{A}$ represents the rotation matrix, which can be defined as follows:

$$\boldsymbol{A}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{10}$$

Next, we design τ. We design τ as the following proportional control over Δ.

$$\tau = -K\Delta, \tag{11}$$

where $K \in \mathbb{R}_+$ is the proportionality coefficient.

Based on the above conditions, the controller is designed as follows.

$$\begin{cases} \dot{\boldsymbol{P}} = \boldsymbol{f} = \boldsymbol{A}(\Delta)\boldsymbol{y}, \\ \dot{\alpha} = \tau = -K\Delta. \end{cases} \tag{12}$$

Thus, $\boldsymbol{f}$ characterizes the result of $\boldsymbol{y}$ rotated by Δ, while τ is actuated by the proportional control with gain K applied to the system according to Δ.

According to Eq. (8), Eq. (10) and Eq. (12), we get the feedback control system in region A as:

$$\begin{cases} \dot{\boldsymbol{P}} = -\begin{bmatrix} \cos\Delta & -\sin\Delta \\ \sin\Delta & \cos\Delta \end{bmatrix}\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \\ \dot{\alpha} = -K\Delta. \end{cases} \tag{13}$$

And in region B, the feedback control system is similarly obtained as:

$$\begin{cases} \dot{\boldsymbol{P}} = -\frac{r}{\varepsilon}\begin{bmatrix} \cos\Delta & -\sin\Delta \\ \sin\Delta & \cos\Delta \end{bmatrix}\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \\ \dot{\alpha} = -K\Delta. \end{cases} \tag{14}$$

.1.3 Stability Analysis of the Control System

To simplify the proof, we transform the Cartesian coordinate system into a polar coordinate system. Differentiating the equation (**1**), we can derive the following result:

$$\dot{\boldsymbol{P}} = \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \end{bmatrix} = \begin{bmatrix} \dot{r}\cos\theta - r\dot{\theta}\sin\theta \\ \dot{r}\sin\theta + r\dot{\theta}\cos\theta \end{bmatrix}. \tag{15}$$

Combining Eq. (15) and Eq. (13), the feedback control system in polar coordinates of region A is collated as follows：

$$\begin{cases} \dot{r} = -\cos\Delta, \\ \dot{\Delta} = -K\Delta + \dfrac{1}{r}\sin\Delta, \\ \dot{\theta} = -\dfrac{1}{r}\sin\Delta. \end{cases} \tag{16}$$

Similarly, the feedback control system in polar coordinates of region B is as follows:

$$\begin{cases} \dot{r} = -\dfrac{r}{\varepsilon}\cos\Delta, \\ \dot{\Delta} = -K\Delta + \dfrac{1}{\varepsilon}\sin\Delta, \\ \dot{\theta} = \dfrac{1}{\varepsilon}\sin\Delta. \end{cases} \tag{17}$$

Observing Eq. (16) and Eq. (17), we find that the differentiation of r and Δ does not contain θ whether in region A or region B. In other words, the variation of r and Δ is independent of θ.

By analyzing the stability of this control system through the Lyapunov method, we can sequentially prove the following conclusions.

Firstly, a robot placed at an arbitrary position in region A with an arbitrary orientation always reaches the boundary between regions A and B in a finite time. Secondly, a robot reaching the boundary between regions A and B then converges toward the origin without going outside of region B. Finally, a robot placed at an arbitrary position in region B in an arbitrary direction may go out to region A once, but then it returns to Step 2.

And draw the general conclusion that：

For the navigation problem of returning a mobile robot based on a two-wheeled vehicle model to the origin in a planar coordinate system, when the time evolution equations for the robot's position and orientation is defined as:

$$\begin{cases} \dot{\boldsymbol{P}} = \boldsymbol{f} = \boldsymbol{A}(\Delta)\boldsymbol{y}, \\ \dot{\alpha} = \tau = -K\Delta. \end{cases} \tag{18}$$

for any $\varepsilon > 0$ and $c > 0$, if:

$$K > \frac{1}{\varepsilon} + c, \ \left|\Delta(0)\right| \leq \pi \tag{19}$$

, and:

$$\boldsymbol{y} = \begin{cases} -\dfrac{\boldsymbol{P}}{\|\boldsymbol{P}\|} & \text{(Region A)}, \\ -\dfrac{1}{\varepsilon}\boldsymbol{P} & \text{(Region B)}. \end{cases} \tag{20}$$

we can derive the conclusion as following:

$$\lim_{t\to\infty} \left\|\boldsymbol{P}(t)\right\| = 0, \ \lim_{t\to\infty} \Delta(t) = 0 \tag{21}$$

In other words, no matter where and in what direction it is placed, the mobile robot is able to reach the goal. For details of the proof process, please refer to our previous research [81].

List of Publications and Conference Presentation

Journal

[1] **R. Xiao**, Y. Tsunoda, & K. Osuka: “Unfavorable” environmental effects can also be beneficial: a simulation analysis of centipede-like swarm robots based on implicit control for navigation and exploration in unknown environments. Artificial Life Robotics, Vol.28, pp. 690-702, (2023).

[2] **R. Xiao**, Y. Tsunoda, & K. Osuka: Proposal and Experimental Verification of an Implicit Control Based Navigation Scheme in Unknown Environment for a Centipede Type Robot, Journal of Robotics and Mechatronics, Vol.34, No.4, pp. 829-843, (2022).

International Conference, Symposium

[1] **R. Xiao**, Y. Tsunoda, & K. Osuka: Proposal and Experimental Validation of a Wheeled Centipede-inspired Navigation Robot in 3D Unknown Environments: Harnessing Environmental Interactions by Modulating Body Rigidity, 29th International Symposium on Artificial Life and Robotics (2024). Peer-reviewed

[2] **R. Xiao**, Y. Tsunoda, & K. Osuka: Proposal and Simulation Analysis of a Centipede Type Robot Dynamically Adjusting Body Stiffness to Harness Favorable Environmental Effects for Navigation in Unknown 3D Environments, 2023 2nd International Conference on Automation, Robotics and Computer Engineering (ICARCE 2023). Peer-reviewed

[3] **R. Xiao**, Y. Tsunoda, & K. Osuka: The Exploitation of “Unfavorable” Environmental Effects in a Centipede-type Swarm Robot System for Unknown Environment Navigation and Exploration, 28th International Symposium on Artificial Life and Robotics (2023). Peer-reviewed

[4] **R. Xiao**, Y. Tsunoda, & K. Osuka: The Development of a Centipede-type Swarm Robot System for Unknown Environment Exploration and Navigation with Exploitation of “Unfavorable” Environmental Effects, The 16th International Symposium on Distributed Autonomous Robotic Systems (2022). Poster presentation

Domestic Conference, Symposium

[1] **R. Xiao**, K. Ito, T. Goto, Y. Tsunoda, & K. Osuka: The Development of a Multi-Jointed Mobile Robot with a Torso Featuring Flexible and Rigid Switchability for Navigation in Unknown 3D Environments, in Proceedings of SCI'23, (2023).

[2] **R. Xiao**, Y. Tsunoda, & K. Osuka: Proposal of "Weak Constraint Swarm" Approach and its Application in Robot Navigation on Rugged Mountainous Terrain, ROBOMECH2022 in Sapporo (2022).

[3] **R. Xiao**, Y. Tsunoda, & K. Osuka: Stability Analysis of Homing Control of a Mobile Robot, *SICE DAS Symposium*, (2021) (in Japanese).

[4] **R. Xiao**, Y. Tsunoda, & K. Osuka: A New Robot Navigation Scheme Combining Implicit and Explicit control to Travel Through the Unknown Environment to Reach the Destination, *21st SI Division Conference of the Society of Instrument and Control Engineers (SI2020)*, (2020).

Others

Domestic Conference, Symposium

[1] 大須賀 公一, 角田 祐輔, & **肖 潤澤**: 対河道閉塞用ポンプシステム i-CentiPot-Hose の提案, 第 22 回計測自動制御学会システムインテグレーション部門講演会 (SI2021), (2021).