Adaptation of a group to various environments through local interactions between individuals based on estimated global information

Tomohiro Hayakawa

Abstract

Groups such as a social insect swarm are composed of many individuals which behave locally to achieve the purpose of the entire group. Those systems are regarded as autonomous distributed systems. By changing the manner of cooperation between individuals, the group can take several functions; thus, the group potentially has adaptability to various environments. However, it is not easy to construct an autonomous distributed system in which the group can adapt to various environments because of the following reasons. First, individuals can observe only local environment, so they cannot directly observe changes of the global environment. Second, individuals can determine only their own behavior, so they cannot directly control the behavior of all other individuals in the group. In this thesis, to construct autonomous distributed systems in which the group can adapt to various environments, an individual behavior that estimates the global information using local interactions and thereby changes its own behavior is focused on. This thesis is composed of five chapters as follows.

In Chapter 1, previous methods to construct individual behaviors in autonomous distributed systems are summarized. Moreover, problems existing in previous individual behaviors in terms of environmental adaptation are explained and then the motivation to focus on an individual behavior that estimates the global information using local interactions and thereby changes its own behavior is presented.

In Chapter 2, how change in the behavior of individual ants based on estimated global information influences to the environmental adaptation of the colony is analyzed. Biological ants change their behaviors based on their surrounding nestmate density that is correlated to the colony size and their own nutritional state that is correlated to the nutritional state of the colony through the repeated trophallaxis. However, changes in individual behavior based on the global information are different depending on the species, so how each change in individual behavior influences to the survivability of the colony is not fully understood. In this chapter, the state transition model is constructed for the lifecycle of red harvester ants that perform four types of tasks, that is, foraging, midden work, nest maintenance, and brood care. Through the simulations, the influences of each change in individual behavior to the survivability of the colony are investigated. The simulation reveals that when individual ants decrease their contact rates in response to increasing colony size, the transition to the foraging task that induces high death rate is suppressed; thus, they enable to maintain a large colony size. This adaptation is regarded as the adaptation to inner environmental changes. Furthermore, when individual ants increase their transition rate to the foraging task in response to decreasing nutritional state of the colony, they increase resilience to outer environmental changes.

In Chapter 3, a unified controller of individual single-legged reconfigurable modular robots that compose multi-legged robot to achieve static walking of the multi-legged robot is constructed. By extending the leg configuration of the multi-legged robot to those that are not

found in nature, the multi-legged robot can execute several types of tasks. The center of mass of the multi-legged robot is important information to achieve the static walking, but each module does not have the information a priori. This problem can be resolved using a strategy similar to trophallaxis of ants in Chapter 2. That is, each module estimates the center of mass of the multi-legged robot using locally obtained information based on the average-consensus control. Using the center of mass information, modules determine their foot toe movement so that the support polygon always encloses the center of mass of the multi-legged robot. Using the proposed system, a multi-legged robot with several inner environments, that is, various leg configurations including the leg malfunction, was mathematically guaranteed to achieve static walking and achievement of static walking for several types of multi-legged robots was demonstrated through dynamic simulations and robot experiments.

In Chapter 4, a unified controller of individual single-legged modules in the multi-legged robot so that the multi-legged robot repeatedly adds a new module to the robot and tries to traverse the outer uneven terrain environment, and finally adapts the morphology to the environment is constructed. Each module is required to detect whether the current morphology of the multi-legged robot can traverse the environment or is required to add a new module to avoid the danger of falling down. Even in the case of uneven terrains, when the center of mass of the multi-legged robot is enclosed by the support polygon, the multi-legged robot can avoid the dangerous situation and achieve static walking, where each module can estimate the information of center of mass of the multi-legged robot by the average-consensus control as mentioned in Chapter 3. Using the estimated center of mass information, each module monitors the risk of falling down in a distributed manner from geometrical relationship between the center of mass and the support polygon of the multi-legged robot. When one of the modules detects the dangerous situation, it transmits the dangerous signal to other modules so that the multi-legged robot returns to the starting area. Then, each module transmits information about a connecting place of a new module around its own body. In accordance with the requested connecting place, a new module is added. By repeating the procedure, the morphology of the multi-legged robot gradually adapts to the environment. Through dynamic simulation and robot experiment, it was demonstrated that a multi-legged robot successfully changed its morphology in response to and traversed the outer environment.

Finally, Chapter 5 concludes this thesis and summarizes the future works.

Acknowledgment

First and foremost, I would like to thank my supervisor, Prof. Fumitoshi Matsuno at Kyoto University, for providing me the opportunity to do the research on swarm robotics.

I thank Profs. Tetsuo Sawaragi and Kei Senda at Kyoto University. They gave me time to discuss on the dissertation and very important suggestions.

I thank associate professor Shigeto Dobata at the University of Tokyo. He provided me the knowledge about biological ants and many advices to progress the research in the field of biology.

I thank associate professor Shizuo Kaji at Kyushu University. He gave me many comments about gait generation system from mathematically viewpoint and many advices about the constitution of the paper.

I thank senior lecturer Roderich Groß at the University of Sheffield. He provided me the opportunity to study at his laboratory and many advices to progress the research in the field of reconfigurable modular robots.

I thank all of the members of the Mechatronics Lab. Especially, members who study legged robot provided me many advices for research and presentation through discussions in the group seminar.

I thank my girlfriend, Maki Sugimura, who always gave me mental and health support.

Finally, I thank all of my family. I thank my parents, Toru Hayakawa and Sachiko Hayakawa, who always understood me and supported me in every aspects.

Contents

Chapter 1

Introduction

In nature, there are many groups composed of many individuals which behave locally to cooperatively achieve the purpose of the entire group. For example, in the division of labor for ants, individual ants locally determine their engaging task for the survival of the colony. In the gap crossing for ants, individual ants locally determine their behavior so that the group can efficiently cross the the gap. In the flocking motion of birds, individual birds locally determine their movement so that the group can move together to a desired location. Those examples show cooperation between organisms; whereas, cooperations inside an organism are shown in the following. For example, in the locomotion of insects, a pattern generator that is embedded in individual leg locally determines movement of the leg to achieve stable locomotion of the entire insect. In the human's body, there are many individual cells that locally produce antibodies that eliminate antigens.

Those systems observed in nature are denoted as autonomous distributed systems. An autonomous distributed system involves many benefits, so the system has been applied to artificial engineering systems.

1.1 What are autonomous distributed systems?

In the following subsections, components, advantages, and disadvantages of autonomous distributed systems are summarized.

1.1.1 Components of the system

Autonomous distributed systems are composed of individuals which share a common purpose of the group, and there is no central integrator which has access to the global information of the entire group. Individuals locally interact with other individuals and the outer environment, then determine their own behavior to achieve the given purpose of the group.

1.1.2 Advantages of autonomous distributed systems

In the following, advantages of autonomous distributed systems are summarized. The first advantage is adaptability [1]. In contrast to a system composed of a single highly functional individual, a group has many degree of freedom owing to a number of individuals; thus, the group can take several functions by changing the manner of cooperation between individuals and can adapt to various environments. In the following, adaptations of the group to inner environmental changes are presented. In foraging task allocation for a group of individual robots, the proportion of individual robots engaged in the foraging task is adjusted in response to the the change of individual energy consumption rate [2, 3]; thus, the group can always perform the foraging at a high efficiency. In gait pattern generation for a multi-legged robot, their phases that determine their foot toe positions are adjusted in response to the change of the phase differences between adjacent legs [4–8]; thus, the multi-legged robot can always achieve stable walking. In flocking motion for a group of individual robots, the individual robots' position and orientation are adjusted in response to the change of relative position and orientation of their nearby individual robots [9]; thus, the group can always move together. In the following, adaptations of the group to outer environmental changes are presented. In foraging task allocation for a group of individual robots, the proportion of individual robots engaged in the foraging task is adjusted in response to the change of amount of food distributed in the outer environment [2, 3, 10]; thus, the group can always perform the foraging at a high efficiency. In flocking motion for a group of individual robots, the position of individuals is adjusted in response to the change of position of obstacles in the outer environment [9]; thus, the individuals can always avoid collisions with the obstacles. In uneven terrain traversal for a group of individual robots, the morphology of the entire robot that is composed of individuals connecting with each other is adjusted in response to the difficulty of the environment [11]; thus, the group can always traverse the given environment.

The second advantage is scalability [1]. Even if the number of individuals in the group is extended or shrunk, the program of individuals remains constant. Moreover, program simplicity, reliability of information exchange, communication cost and memory usage of individuals are constant because individuals utilize only local information whose amount does not depend on the number of individuals; thus, scalability is inherent in the system.

The third advantage is robustness [1]. Malfunction of an individual does not directly lead to breakdown of the entire system; thus, robustness is inherent in the system. Moreover, when a communication error between individuals occurs, influence of the error on the entire system is smaller than that in a centralized system [12]; thus, robustness against communication error is also inherent in the system.

1.1.3 Disadvantages of autonomous distributed systems

In the following, disadvantages of autonomous distributed systems are summarized. First, achievement of the group's purpose is more difficult than that for centralized systems because the controller of individuals are homogeneous and they determine their own behavior based only on local information. As an example, autonomous distributed systems often face the "deadlock" problem, in which a set of individuals' states converge to a set of equilibrium states which is different from the group's desired state. For example, when the Kuramoto model, which is a famous phase difference controller in coupled oscillators [13], is applied to a ring of nine identical oscillators, many stable phase difference sets exist [14]; thus, the set of phases does not always converge to the desired state. Note that a centralized system can always control the phase difference set to a desired state. To resolve a deadlock problem, prior deadlock avoidance or posterior deadlock resolution such as [15] is necessary.

Second, even if a control framework to achieve the group's purpose can be constructed, autonomous distributed systems take more time to achieve the group's purpose than a centralized system [16] as the following reason. In a centralized system, a central integrator obtains information of the entire group and processes it at once to achieve the group's purpose; whereas, in an autonomous distributed system, individuals in the group repeatedly obtain a limited amount of local information and change their own states toward the group's desired state.

1.2 How are individual controllers constructed?

As shown in Section 1.1.3, it is difficult to construct an individual controller that always achieves the group's purpose. In this section, methods to construct an individual controller are presented.

To construct an individual controller, there are two approaches: system analysis and system synthesis. System analysis corresponds to analysis of living organisms' intelligence and synchronization theory from a mathematical viewpoint. System synthesis corresponds to construction of an individual controller using knowledges obtained by system analysis. In the following subsections, examples of system synthesis using knowledges obtained by system analysis are presented.

1.2.1 Task allocation for a group of social insects and robots

Task allocation in social insects has been extensively studied. The study of task allocation in social insects is particularly interesting since regulation of their labor division is achieved by surprisingly simple strategies which also inspire the solutions for task allocation in a group of robots. In 1984, Wilson observed that individual ants (*Pheidole*) change their state from resting to working when their demand exceeds their own thresholds which are different for

individuals [17]. When working individuals with low response thresholds were removed manually, the associated demand increased until it eventually reaches the higher thresholds of the remaining resting individuals. Then, the increase of demand beyond those thresholds stimulates those remaining individuals to perform the task. The range of thresholds contributes to replacement of working individuals when they become fatigued; thus, the group is able to always obtain a certain amount of energy [18].

Bonabeau et al. proposed a state transition model of ants denoted as fixed response threshold model [19], which is a simple model of regulation of division of labor in insect societies. In the model, each ant i in resting state changes its state to working at a rate

$$P_i = \frac{s_i^n}{s_i^n + \theta_i^n}, \tag{1.1}$$

where P_i is the state transition rate of ant i, s_i is the environmental stimulus to ant i, θ_i is the threshold parameter of ant i, and n is a constant value. In the fixed response threshold model, θ_i is constant and identical to all ants.

In the following, examples of the system synthesis for state transition for individual robots are presented. The fixed response threshold model was applied to a controller for individual robots [10]. Through simulation, the number of working robots was adjusted in response to the amount of food at the group's nest. Later, as an extension of the fixed response threshold model, adaptive response threshold model, in which the response threshold θ_i is not identical and calculated dynamically by the individuals, was applied to a controller for individual robots [2, 3]. The adaptive response threshold model gives robots higher adaptation capabilities than the previous fixed response threshold model, making a group of robots able to adapt more efficiently to dynamical environments. Through simulation [2] and robot experiment [3], the number of working robots was adjusted to maintain a constant amount of food at the group's nest for several energy consumption rates of individual robots. Lee proposed another rule to update θ_i, under which it is mathematically guaranteed that the proportion of task allocation of workers converges to the proportion of the task demand [20].

1.2.2 Cooperative uneven terrains traversal for a group of social insects and robots

For some social insects, it has been observed that individuals assemble with one another to achieve tasks that cannot be achieved by themselves. For example, ants link their bodies together to build bridges for gap traversal [21–23].

In the following, examples of the system synthesis for uneven terrains traversal for a group of robots are presented. Inspired by the self-assembly in social insects, reconfigurable modular robot systems, which are composed of repeated robot elements with a mutual physical connection mechanism, have been invented [24–28]. Compared with a single robot with a constant morphology, the reconfigurable modular robots can change the morphology in response to the task; thus, reconfigurable modular robot system involves high adaptability. O'Grady et

al. [11] proposed a reconfiguring strategy for each module to traverse unknown slope and gap. In the environment, there are a certain number of modules and a goal that is located beyond unknown slope or gap environment. When one of the modules detects that the environment cannot be traversed by itself, all of the modules assembly into a large robot and then the large robot tries to traverse the environment again; thus, the modules can traverse the environment.

1.2.3 Gait pattern generation for a natural creature and a multi-legged robot

In 1911, Graham Brown observed that mechanisms located entirely within the spinal cord are responsible for generating the basic rhythm for stepping in each leg [29]. Such mechanisms are called the central pattern generator, abbreviated by CPG. Later, it was observed in cats and cockroaches that each CPG in the corresponding leg adjusts its movement patterns from signal of other CPGs and load carried by the corresponding leg [30]. For cockroaches, a leg's stance state induces adjacent leg's flight state, whereas a leg's flight state induces adjacent leg's stance state. Meanwhile, large load of the leg induces the corresponding leg's stance state. From these, a CPG's behavior is locally determined through inner interaction with other CPGs and the outer environment. Based on these observations in living organisms, a CPG has been modeled as an oscillator and a CPG network has been modeled as a set of coupled oscillators.

Synchronization in several types of coupled oscillator networks has been elucidated from a mathematical viewpoint. The Kuramoto model is a famous coupled oscillator system [31] in which each oscillator i is actuated as follows:

$$\frac{d\theta_i}{dt} = \omega_i - \frac{K}{n} \sum_{j \in N_i} \sin\left(\theta_i - \theta_j\right), \tag{1.2}$$

where the number of oscillator is n, phase of oscillator i is θ_i, phase velocity of oscillator i is ω_i, the set of indices of neighboring nodes in the network for oscillator i is N_i, and K is a constant value. Dörfler and Bullo summarized characteristics, convergence of several oscillator networks, and parameter conditions for emergence of synchronization about Kuramoto model [32].

In the following, examples of the system synthesis for CPG-based locomotion control of multi-legged robots are presented. Each leg in a multi-legged robot is modeled as an oscillator. First, CPG models for quadruped type are presented. Collins and Stewart [4] developed a van der Pol type CPG-based controller for a quadruped, where some parameters were homogeneous and the CPG network was determined a priori. By changing the parameter values, animal gait patterns such as the pronk, trot, and gallop, were reproduced. Owaki et al. [33] developed a homogeneous CPG-based controller for a quadruped, where the CPGs were not coupled explicitly but implicitly through environmental interaction. There was no communication between CPGs, but gait patterns observed in animals were reproduced. Second, CPG models for hexapod and myriapod type are presented. Odashima et al. [5] developed a homogeneous

CPG-based controller for a hexapod and achieved static walking for not only a translation movement but also a turning movement. Inagaki et al. [6] developed a homogeneous CPG-based controller for myriapods including a hexapod and adjusted the gait pattern explicitly in response to the target locomotion speed. Homogeneous CPG-based controller with explicit inter-CPGs coupling and implicit phase modulation based on the interaction with the ground was developed for a hexapod by Fujiki et al. [7], and for myriapods by Tsuchiya et al. [8]. They produced a gait transition in response to the target speed change.

1.2.4 Flocking motion for a group of animals in nature and robots

The flocking motion of a flock of birds, a herd of land animals, and a school of fish is a beautiful behavior in which a group of individuals forms a large group moving together toward a common target location. Flocking motion emerges at the group level in a distributed manner as a consequence of local interactions between individuals. In 1987, Reynolds constructed an individual behavior denoted as Boid model to emerge flocking motion in simulation [34]. The individual behavior is as follows:

- Collision avoidance: avoid collisions with nearby flockmates.
- Velocity matching: attempt to match velocity with nearby flockmates.
- Flock centering: attempt to stay close to nearby flockmates.

Flocking motion is regarded as the consensus problem in multi-agent systems, which has been discussed from a mathematical viewpoint. Among well-known consensus methods, average-consensus is a particularly famous consensus algorithm, which is as follows:

$$\frac{dx_i}{dt} = \sum_{\forall j \in N_i} (x_j - x_i)\,, \tag{1.3}$$

where x_i is the state of agent i, and the neighbor indices set of agent i is N_i. Under Eq.(1.3), the states of all agents converge to the average of their initial states [35].

In the following, an example of the system synthesis for flocking motion of a group is presented. Robots with limited sensing capabilities must keep a compact formation by measuring relative position and orientation of neighbors. In the first successful implementation of flocking motion on robots, the average-consensus Eq.(1.3) was applied to achieve heading direction consensus [9].

1.2.5 The human immune system and the application to a robot's controller

In the human's body, there are many cells interacting with each other to eliminate foreign substances invading the body, such as viruses, bacteria, and other parasites, which are collectively called antigens [36]. The protection system that eliminates antigens is referred as the immune system. The cells do not follow commands from the brain but instead cope with local

situations autonomously. In the following, a brief process for the cells to collectively produce antibodies is presented. B-cells are responsible for the recognition and elimination of antigens, where they are capable of recognizing only one type of antigen. When B-cells interact with an antigen, the B-cells are stimulated and then proliferate rapidly (called clonal selection [37]). Moreover, B-cells are stimulated or suppressed by interaction with other B-cells. Finally, increased B-cells eventually mature into plasma cells which produce antibodies corresponding to one type of antigen.

Farmer et al. [38] proposed a model of antibody concentration dynamics as follows:

$$\frac{dx_i}{dt} = c\left(\sum_{j=1}^{N} m_{ji}x_i x_j - k_1\sum_{j=1}^{N} m_{ij}x_i x_j + \sum_{j=1}^{n} m_{ji}x_i y_j\right) - k_2 x_i, \tag{1.4}$$

where the number antibody types is N, the concentration of antibody type i is x_i, the number of antigen types is n, the concentration of antigen type i is y_i, and c, m_{ij}, m_{ji}, k_2 are constant values. On the right side of Eq.(1.4), the first term is the stimulus from other B-cells, the second term is the suppression from other B-cells, the third term is the interaction with antigens, and the fourth term is natural death of B-cells.

In the following, examples of the system synthesis using the immune system model Eq.(1.4) are presented. Lee et al. applied the model to the problem of consensus achievement of multi robots under changing environment [39]. Each robot has four types of behaviors associated with antibody types and corresponding variables. The corresponding variables associated with antibodies are updated based on Eq.(1.4) through interaction with other robots and the environment associated with antigens. Each robot chooses one of the four behaviors with the highest concentration of antibodies. Finally, the robots' behaviors synchronized to the behavior suitable for the environment. Luh et al. applied the immune system model to a mobile robot's navigation in unknown environment [40]. Surrounding obstacles and the goal direction are regarded as antigens and steering direction choices are regarded as antibody types. The corresponding variables associated with antibodies are updated based on Eq.(1.4) and the robot determines its moving direction based on the state of the antibodies. Through simulation, it was validated that the robot successfully traversed a given maze.

1.3 Problems for some previous autonomous distributed systems

As mentioned in Section 1.1.2, a certain level of adaptability to environmental changes is embedded in some previous autonomous distributed systems. However, individual behaviors in those systems are constructed under some assumptions about the environment. Therefore, the groups cannot adapt to the unexpected environments beyond the scope of the assumptions. The examples are presented as follows:

First, in task allocation system for a group of robots, only foraging task was assumed to exist in the environment [2,3,10]. Therefore, the group of robots cannot adapt to the environments

where several types of tasks exist.

Second, in gait generation systems for a multi-legged robot, the leg configurations of the multi-legged robot were assumed to be only natural creature types [4–8, 33]. Therefore, the entire multi-legged robot cannot adapt to the change of leg configuration, such as the leg malfunction.

Third, in cooperative uneven terrain traversal for a group of robots, the number of individual robots in the group is constant regardless of the environment [11]. Therefore, the group of robots cannot adapt to difficult uneven terrains beyond the expectations.

1.4 Common solution for the problems in Section 1.3

In nature, individuals in a group are informed of the global information of the group through repeated local interactions. For example, ants are informed of colony-wide nutritional state via frequent trophallaxis [41]. Repeated trophallaxis should result in averaging of individual nutritional states, making each worker aware of the colony's nutritional state from its own nutritional state [42]. As shown above, individuals can estimate the global information and thereby change their own behavior. When each individual changes its own behavior based on estimated global information, the global state is reflected to the individual behavior, so the adaptability of the group to environments is considered to be enhanced. In this thesis, three problems explained in Section 1.3 are resolved using the common strategy such that individuals estimate the global information using local interactions and thereby change their own behaviors.

1.5 Organization of the thesis

The thesis is written as follows: In Chapter 2, to consider an individual behavior that achieves multi-task allocation, a multi-task allocation system of red harvester ants (*Pogonomyrmex barbatus*) in nature is focused on. How behavior change of individual ants based on the estimated global information enhances the inner and outer environmental adaptation of an ant colony is analyzed. Actually, the global state of the colony is time-dependent, so each ant cannot always estimate the global state of the colony accurately. Chapter 2 is discussed under the assumption that all individual ants can always estimate the global information accurately.

In Chapter 3, a unified controller of individual single-legged reconfigurable modular robots that compose multi-legged robot to achieve static walking of the multi-legged robot is constructed. The leg configuration of the multi-legged robot is not limited to the natural creature types; thus, the multi-legged robot can achieve several tasks. The individual controller is constructed using a strategy such that each module estimates the global information of the multi-legged robot using exchanged local information.

In Chapter 4, a unified controller of individual single-legged modules in the multi-legged

robot so that the multi-legged robot repeatedly adds a new module to the robot and tries to traverse the outer uneven terrain environment, and finally adapts the morphology to the environment is constructed. Outer environment refers to the slope and gap environments with several difficulties. The individual controller is constructed using a strategy such that each module estimates the global information of the multi-legged robot using exchanged local information.

Finally, Chapter 5 concludes the thesis. Future work is also summarized in this chapter.

Chapter 2

Adaptation of an ant colony to inner and outer environment

Biological swarms, such as flocks of birds, schools of fish, or colonies of bees, are one of the most striking real-world examples of autonomous distributed systems. These systems have no central control; each agent determines its behavior based only on locally available property. However, the whole system can often achieve "intelligent" global-scale behaviors, such as self-assembly [43], cooperative object transport [44, 45], bridge construction [21, 22], and task allocation [46]. However, the mechanisms connecting individual behaviors and their interactions to such swarm intelligence are not self-evident.

The individual-level behaviors underpinning biological swarms have been studied through quantitative observation of animal swarms and by using mathematical models. One of the practical applications of the results is in designing multi-robot control systems, which to date have replicated a variety of tasks performed by biological swarms, such as self-assembly [11,47], cooperative object transport [48, 49], bridge construction [11], and task allocation [3, 50–52].

Colonies of social insects are a typical example of autonomous distributed systems and have been extensively studied due to the ease of observing both individual-level behaviors and colony-level properties. It has been observed that social insect workers can acknowledge colony-level properties, such as colony size and nutritional status, most likely through local interactions with other workers; their understanding of these properties then regulates their behaviors via positive or negative feedback [53–59]. For example, an individual worker can estimate colony size based on her rate of contacts with nestmates. Gordon et al. [53] examined how ants of three species (*Lasius fuliginosus*, *Myrmica rubra*, and *Solenopsis invicta*) modified individual contact rates in nest sites of different size. *Lasius fuliginosus* adjusted the contact rate between nestmates in response to colony size by aggregating at low densities and avoiding each other at high densities. Interestingly, however, this pattern was not observed in *M. rubra* and *S. invicta*. Similarly, individual insects alter their behavior in response to colony nutritional state, another colony-level property that is crucial to colony survival. A previous study of honeybees (*Apis mellifera*) and black garden ants (*Lasius niger*) found that the

number of foragers increased as colony nutritional state deteriorated [54, 58]. However, the opposite pattern was reported in ants *Temnothorax rugatulus*, in which foragers decreased foraging activity in response to the threat of starvation [57]. These previous studies suggest that, although individuals' indirect knowledge of colony-level properties appears to be adaptive in these species of ants, how each worker alters her behavior in response to these properties is not self-evident and depends on species-specific life history strategies.

In this study, we focus on how the modes of feedback from colony-level properties to individual behaviors affect the colony's survival by modeling colonies of the red harvester ant (*Pogonomyrmex barbatus*), which has been well studied in the context of colony organization [46]. We describe several scenarios under which the ants show contrasting modes of feedback from two colony-level properties, colony size and nutritional state. The artificial behavioral rules are as follows: ants change their contact rate based on one of three types of feedback (negative, absent, positive) about the current colony size, and they change their rate of transition to a foraging task based on one of three types of feedback (negative, absent, positive) about the current nutritional state of the colony.

Below, we briefly review previous mathematical models describing task allocation in social insects. The life cycles of social insects such as wasps (*Metapolybia cingulata*) [60] and *A. mellifera* [61] have mainly been analyzed with multi-agent modeling, which describes the behavior of each agent. The models assumed a constant number of workers and focused on changes in task allocation. For example, Karsai and Runciman [60] investigated how colony efficiency changed as a function of task allocation rate, and Schmickl and Crailsheim [61] investigated how task allocation rate changed in response to changes in workers' internal parameter values.

Ordinary differential equation (ODE) models, which describe each agent's behavior from a macroscopic viewpoint, have also been used in the study of social insect colonies [62–65]. Again, the above-mentioned models assumed a constant number of workers and focused on changes in task allocation. They investigated the response of task allocation rate to various kinds of environmental disturbance, such as changing weather [65], food availability [64], successful task execution [62], and a stimulus for a task [63].

Previous studies have not addressed the dynamics of colony size, which would be highly susceptible to environmental disturbance. Therefore, in this study, we investigate colony demography and various types of disturbance resistance by using an ODE model to elucidate the behavioral rules that are most beneficial for colony survival. We explicitly incorporate colony demography, that is, the hatching to death process of workers, to consider the dynamics of colony size.

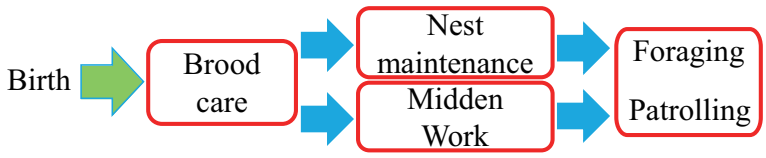


Fig. 2.1 Task transition rules for workers. A young worker tends to undertake brood care task and an older worker tends to undertake foraging or patrolling tasks.

2.1 ODE model of worker behaviors

We constructed the ODE model based on the life cycle of *P. barbatus*, which is the best-studied species in the context of self-organizing task allocation systems [46]. *Pogonomyrmex barbatus* is distributed in North America and feeds mainly on seeds. A mature colony consists of one queen and approximately 10,000 workers. The workers' tasks are divided into brood care, nest maintenance (removing debris from the nest), midden work (constructing a midden on the nest mound), foraging [46, 66], and patrolling. Information about the ants' ecology is largely based on [46].

Briefly, the tasks and task allocation system are as follows. Workers demonstrate age polyethism, whereby their tasks depend on their age [67]. The switch between tasks is unidirectional, which means that the youngest workers are initially engaged in brood care, shift to nest maintenance [66] or midden construction [68], and finally shift to foraging or patrolling [69] (Fig. 2.1).

Recruiting workers are defined as workers that are inside the nest, after they have executed their tasks and returned to the nest. When recruiting workers who undertake a task contact inactive workers, the inactive workers switch their task to that of the recruiting workers [70–75]. Workers' cuticular hydrocarbon profiles differ by task, allowing workers to identify the tasks undertaken by other workers [76].

Based on these observations, we constructed an ODE model of the *P. barbatus* life cycle. The main states and variables used in the ODE model are summarized in Table 2.1. Constant parameters are summarized in Table 2.2. States and variables are indicated by capital letters and constant parameters are indicated by lowercase letters. To minimize differences between the ODE model and the actual ant life cycle, we determined the parameter values based on observational data wherever possible.

2.1.1 Definitions of worker states

For simplicity, we assume that workers always undertake one of four tasks: foraging, midden work, nest maintenance, or brood care, with patrolling included in the foraging task. Additionally, we assume that workers display one of three activity levels: recruiting, engaged, or

Table 2.1 Description of states and variables. A midden unit indicates the total amount of midden constructed by one worker in her task executing period and a debris unit indicates the total amount of debris carried by one worker in her task executing period.

State and variable	Description [unit]
R,E,I	Activity level of workers: recruiting, engaged, and inactive, respectively
F,M,N,B	Tasks: foraging, midden work, nest maintenance, and brood care, respectively
$W_i(t)$	Number of workers in state i at time t [number]
$K_i(t)$	Number of newly hatched workers added to state i at time t [number/s]
$P_{i\to j}(t)$	Rate at which workers change their state from i to j at time t [/s]
$G_i(t)$	Death rate of workers in state i killed by enemy attack at time t [/s]
$H_i(t)$	Death rate of workers in state i killed by starvation at time t [/s]
$X(t)$	Contact rate between two random workers in the nest at time t [/(s·number)]
$Y(t)$	Coefficient of state transition towards foraging at time t [-]
$C(t)$	Degree of midden construction at time t [midden]
$D(t)$	Amount of debris inside the nest at time t [debris]
$A(t)$	Surplus nutritional energy of the colony at time t [kcal]

inactive. Recruiting workers recruit inactive workers in the nest to replace them after they have executed their tasks. Engaged workers are searching a task (searching food, searching materials for midden, searching debris, or searching brood) or successfully executing a task (carrying food, constructing the midden, removing debris, or performing brood care). Inactive workers rest inside the nest.

Worker states are summarized in Fig. 2.2. Activity levels of recruiting, engaged, and inactive are indicated by R, E, and I, respectively. The foraging task is indicated by F, midden work by M, nest maintenance by N, and brood care by B. Using activity level i and task j, each worker's state is described by $i(j)$. Workers in state E(F) and E(M) work outside the nest, whereas the others work inside the nest. The number of workers in each state increases as a result of hatching and inward state transitions, and decreases as a result of outward state transitions and deaths. Therefore, the dynamics of the number of workers in each state is written as

$$\frac{dW_i(t)}{dt} = K_i(t) + \sum_{j(\neq i)} P_{j\to i}(t)W_j(t) - \sum_{j(\neq i)} P_{i\to j}(t)W_i(t) - \left(G_i(t) + H_i(t) + q_i\right)W_i(t), \quad (2.1)$$

where we denote the number of workers in state i at time t by $W_i(t)$, the number of workers hatched in state i at time t by $K_i(t)$, the transition of workers from state i to state j at time t by $P_{i\to j}(t)$, the number of workers in state i killed by external enemy attack at time t by $G_i(t)$, the number killed by starvation at time t by $H_i(t)$, and the number dying of natural causes by q_i. Note that colony size is equivalent to $\sum_i W_i(t)$.

2.1.2 Hatch rate

In this section, we formulate the hatch rate of workers in state i (i.e., $K_i(t)$). Eggs laid by a queen are cared for by brood care workers, who also feed the larvae that hatch from the eggs.

Table 2.2 Description and value of constant parameters. [H] means a heuristic value.

Parameter	Description [unit]	Value [source]
s_{Rstop}	Rate at which recruiting workers spontaneously stop recruiting [/s]	0.1 [H]
s_{Estop}	Rate at which engaged workers spontaneously stop engaging in a task [/s]	1×10^{-4} [H]
s_{Esame}	Rate at which inactive workers are spontaneously engaged in the same task [/s]	1×10^{-4} [H]
s_{Ediff}	Rate at which inactive workers are spontaneously engaged in a different task [/s]	2×10^{-7} [H]
ρ_{fdet}	Rate at which foraging workers perceptually detect food [/s]	1.3×10^{-3} [77]
ρ_{cdet}	Coefficient for perception of midden detection [midden]	1 [H]
ρ_{ddet}	Coefficient for perception of debris detection [debris]	1 [H]
ϕ	Rate at which brood care workers are engaged in the task [-]	0.2 [H]
$k_{\max}$	Maximum hatch rate [number/s]	$\frac{1.0\times10^4}{\tau_{year}}$ [46]
l	Rate at which larvae become workers owing to a brood care worker's care [/s]	1×10^{-5} [H]
α	Maximum value of $X(t)$ [/(s·number)]	6×10^{-2} [H]
β	Coefficient for $Y(t)$ [kcal]	$u_{\mathrm{I(B)}} \times 1.4\tau_{day}$ [H]
g	Rate at which outside workers are killed by external enemies [/s]	5×10^{-7} [H]
q_i	Death rate of workers in state i dying of natural causes [/s]. Parameter values correspond to those of workers undertaking tasks F, M, N, and B, respectively	1×10^{-6} [78], 1×10^{-7} [H], 1×10^{-7} [H], 1×10^{-8} [H]
γ	Coefficient for decrease of enemy attack owing to midden construction [midden]	1 [H]
θ	Natural midden collapse rate [/s]	1×10^{-6} [H]
ψ	Natural debris inflow rate [debris/s]	1×10^{-5} [H]
λ	Amount of energy per unit of food [kcal]	0.3 [H]
μ	Coefficient for energy shortage [kcal]	$u_{\mathrm{I(B)}} \times 7000\tau_{day}$ [H]
$a_{\max}$	Maximum surplus nutritional energy of a worker [kcal]	$u_{\mathrm{I(B)}} \times 5\tau_{day}$ [H]
u_i	Metabolic energy consumed by a worker in state i per unit of time [kcal/s]. Parameter values correspond to $i \in${E(F),E(M)} and $\forall i \notin$ {E(F),E(M)}	9.4×10^{-6} [79, 80], 3.2×10^{-6} [79, 80]
τ_{day}	Workers' active time in one day [s]	$6.5 \times 60 \times 60$ [46]
τ_{year}	Workers' active time in one year [s]	$210\tau_{day}$ [46]

These larvae become workers engaged in brood care. The rate at which the queen lays eggs has a physiological limit, and larvae do not grow if their nutritional state is poor [81, 82]. We assume that the hatch rate decreases as the number of workers engaged in brood care, that is, $W_{\mathrm{E(B)}}$, decreases. In addition, when the colony's nutritional surplus is a negative value, the hatch rate of workers decreases. Therefore, let the hatch rate $K_i(t)$ be represented by

$$K_i(t) = \begin{cases} \min\left(k_{\max}\exp\left(\dfrac{\min(0, \frac{A(t)}{\sum_i W_i(t)})}{\mu}\right), l\ W_{\mathrm{E(B)}}(t)\right) & (i = \mathrm{E(B)}) \\ 0 & (\text{otherwise}), \end{cases} \tag{2.2}$$

where $k_{\max}$ is the maximum rate of egg-laying by the queen, $A(t)$ is the surplus nutritional energy of the colony at time t, the dynamics of which are explained in Eq.(2.9), μ is a coefficient for the egg-laying rate, and l is the rate at which larvae become workers owing to a brood care worker's care. Note that $A(t)/\sum_i W_i(t)$ is the average surplus nutritional energy of one worker and $\min(0, A(t)/\sum_i W_i(t))$ is a non-positive value that represents the shortage of nutritional energy of one worker.

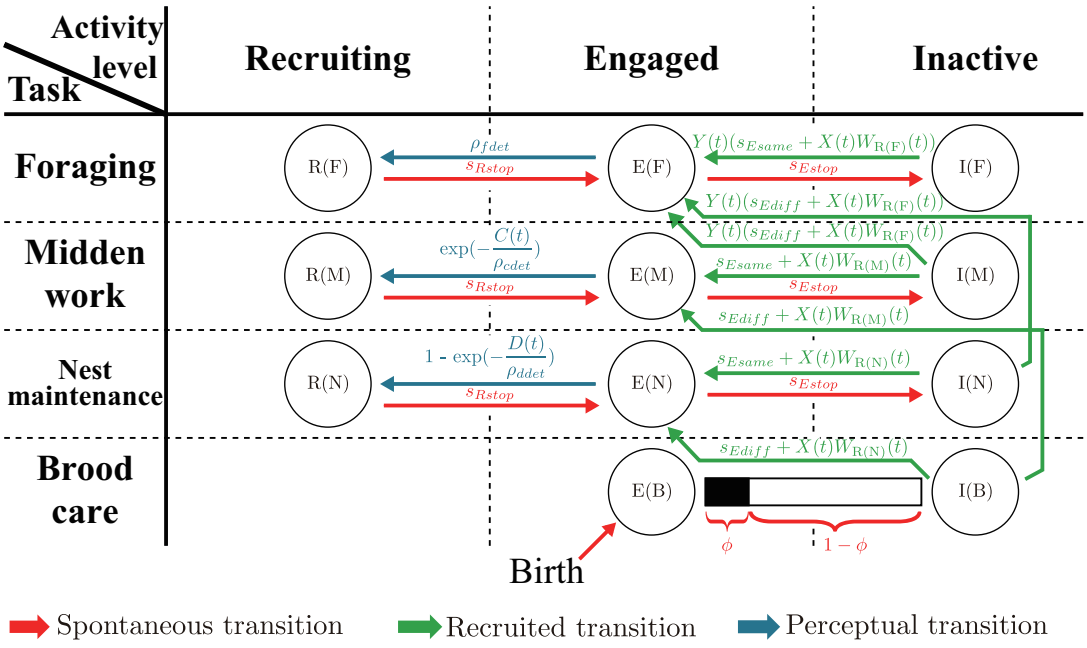


Fig. 2.2 State transitions. A worker always undertakes one of the 11 states defined by activity level (columns) and task (rows). She changes her state in the direction of the arrows, and arrow colors indicate the types of transitions. Each equation assigned to each arrow from state i to j corresponds to the state transition rate $P_{i \to j}$.

2.1.3 State transition

In this section, we formulate the state transition of workers from state i to state j (i.e., $P_{i \to j}(t)$). In this study, workers change their state as a result of either spontaneous transition, recruited transition, or perceptual transition. All state transition rates are shown in Fig. 2.2.

Spontaneous transition

First, we formulate spontaneous transition. In the state transition model of Pacala et al. [62], there are active workers and inactive workers, and active workers spontaneously change their state to inactive at a constant rate. Based on this model, we constructed our model as follows. Recruiting workers, which belong to state R(i)$|i \in$\{F, M, N\}, stop recruiting and spontaneously transition to the engaged state E(i) at rate s_{Rstop}. Engaged workers, which belong to E(i)$|i \in$\{F, M, N\}, stop searching a task and transition spontaneously to the inactive state I(i) at rate s_{Estop}. In addition, we assume that brood care workers always work at a constant rate, ϕ.

Recruited transition and feedback from colony-level properties to individual behavior

Second, we formulate recruited transition, which is including feedback from colony-level properties. Based on the state transition model of Pereira and Gordon [63], we model recruited transition as follows. Inactive workers, which belong to I(i), change their state to E(j) not only spontaneously but also by contacting recruiting workers in state R(j), where inactive workers are spontaneously engaged in the same task at rate s_{Esame}, and a different task at rate s_{Ediff}. Let $X(t)$ be the contact rate between two randomly selected workers in the nest at time t. Therefore, $X(t)W_{\mathrm{R}(j)}(t)$ denotes the contact level between one worker and any recruiting workers in state R(j).

In the following, we formulate $X(t)$. Social insects change their contact rate in response to current colony size in different ways. Interactions among *P. barbatus* have been well studied [71, 73,83], but to the best of our knowledge, it remains unclear whether *P. barbatus* workers change their contact rate based on current colony size. To investigate how different types of feedback from current colony size influence colony demography, we assume three types of behavioral rules for $X(t)$, as follows:

$$X(t) = \begin{cases} \dfrac{\alpha}{w_-} W_{in}(t)^{-1} & \text{(Negative feedback)} \\ \dfrac{\alpha}{w_0} & \text{(Without feedback)} \\ \dfrac{\alpha}{w_+} W_{in}(t) & \text{(Positive feedback),} \end{cases} \tag{2.3}$$

where $W_{in}(t) \equiv \sum_i W_i(t) - W_{\mathrm{E(F)}}(t) - W_{\mathrm{E(M)}}(t)$ is the number of workers inside the nest and α, w_-, w_0 and w_+ are constant values. A negative feedback means that $X(t)$ decreases as the colony size increases, and vice versa, and w_- is determined in such a way that $X(t) = \alpha$ when $W_{in}(t)$ is minimized (i.e., 1), hence $w_- = 1$. A positive feedback means that $X(t)$ increases as the colony size increases, and vice versa, and w_+ is determined in such a way that $X(t) = \alpha$ when $W_{in}(t)$ is maximized (i.e., the colony is mature and contains 10,000 workers), hence $w_+ = 10,000$. A situation without feedback means that $X(t)$ does not depend on the colony size, and w_0 is a free parameter within the range $1 \leq w_0 \leq 10,000$.

Let $Y(t)$ represent a coefficient for state transition toward foraging at time t. Social insects change their rate of transition to the foraging task in different ways based on the nutritional state of their colony. To investigate how different types of feedback from the current nutritional state influence colony demography, we assume three types of behavioral rules for $Y(t)$ as a function of the average nutritional state of the colony, that is, $\frac{A(t)}{\sum_i W_i(t)}$, as follows:

$$Y(t) = \begin{cases} y_- \exp(-\dfrac{A(t)}{\beta \sum_i W_i(t)}) & \text{(Negative feedback)} \\ 1 & \text{(Without feedback)} \\ y_+ \exp(\dfrac{A(t)}{\beta \sum_i W_i(t)}) & \text{(Positive feedback),} \end{cases} \tag{2.4}$$

where the constants $y_-(\equiv \frac{a_{\max}}{\beta}(1-\exp(-\frac{a_{\max}}{\beta}))^{-1})$ and $y_+(\equiv \frac{a_{\max}}{\beta}(\exp(\frac{a_{\max}}{\beta})-1)^{-1})$ are determined in such a way that the average of $Y(t)$ is equal to 1 within the positive range of the surplus nutritional energy of each worker. $a_{\max}$ and β are constant values, where $a_{\max}$ is the maximum surplus nutritional energy of each worker. A negative feedback means that $Y(t)$ decreases as the nutritional energy of the colony increases, and vice versa. A positive feedback means that $Y(t)$ increases as the nutritional energy of the colony increases, and vice versa. A situation without feedback means that $Y(t)$ does not depend on the nutritional energy of the colony.

Perceptual transition

Finally, we formulate perceptual transition. Let the rate at which foragers detect food be a constant, ρ_{fdet}. Let the degree of midden construction at time t be $C(t)$, the dynamics of which are explained in Eq.(2.7). Gordon [68] reported that after the midden was removed by a researcher, the mean number of ants engaged in midden work increased. Based on this observation, we assume that as $C(t)$ decreases, the rate at which midden workers successfully execute the task (constructing the midden) increases (Fig. 2.2), where ρ_{cdet} is a coefficient for perception of midden detection. Let the amount of debris inside the nest at time t be $D(t)$, the dynamics of which are explained in Eq.(2.8). As the amount of debris inside the nest increases, the rate at which nest maintenance workers detect the debris also increases. Thus, we assume that as $D(t)$ increases, the rate at which nest maintenance workers successfully execute the task (removing debris from the nest) increases (Fig. 2.2), where ρ_{ddet} is a coefficient for perception of debris detection.

2.1.4 Death rate

In this section, we formulate three types of death rate (i.e., $G_i(t)$, $H_i(t)$, and q_i). First, we formulate $G_i(t)$. In this study, we assume that enemies do not enter the nest, hence workers inside the nest are not killed by external enemies. Workers in state E(F) and E(M) work outside the nest, hence they are sometimes killed by an enemy. Workers in state E(F) search for food far from the nest, and the death rate of these workers is represented by a constant, g. Workers in state E(M) construct a midden on the nest mound. It has been suggested that the midden functions to repel external enemies [68]. Therefore, we assume that $G_{\mathrm{E(M)}}(t)$ decreases as $C(t)$ increases. Thus, we represent $G_i(t)$ by

$$G_i(t) = \begin{cases} g & (i = \mathrm{E(F)}) \\ g\,\exp(-\dfrac{C(t)}{\gamma}) & (i = \mathrm{E(M)}) \\ 0 & (\text{otherwise}), \end{cases} \tag{2.5}$$

where γ is a constant.

Second, we formulate $H_i(t)$. Workers start to die when their level of nutritional energy

declines beyond a threshold value. Thus, we assume that the death rate of workers killed by starvation, that is, $H_i(t)$, increases as the shortage of nutritional energy of each worker increases. Therefore, we represent $H_i(t)$ by

$$H_i(t) = 1 - \exp(\frac{\min(0, \frac{A(t)}{\sum_i W_i(t)})}{\mu}), \tag{2.6}$$

where μ is a constant.

Finally, we formulate q_i. Workers transition between tasks based on the unidirectional rules outlined in Fig. 2.1. Hence, we set the life-span of foragers to be the shortest among all workers, and that of brood care workers to be the longest (see Table 2.2).

2.1.5 Environmental dynamics

In this section, we formulate the environmental dynamics, that is, the degree of midden construction (i.e., $C(t)$), the amount of debris inside the nest (i.e., $D(t)$), and the surplus nutritional energy of the colony (i.e., $A(t)$). These environments have a significant influence on the workers' life cycle, that is, the hatch rate, state transition rate, and death rate. Meanwhile, the environment is changed by the workers' activity.

Midden construction

First, we formulate the dynamics of the degree of midden construction (i.e., $dC(t)/dt$). When left unattended, the midden spontaneously collapses. When midden workers construct the midden, the midden grows. Thus, $dC(t)/dt$ is represented by

$$\frac{dC(t)}{dt} = -\theta C(t) + P_{\mathrm{E(M)}\to\mathrm{R(M)}}(t) W_{\mathrm{E(M)}}(t), \tag{2.7}$$

where θ is the natural rate of midden collapse and $P_{\mathrm{E(M)}\to\mathrm{R(M)}}$ is the rate at which the task is successfully executed (the midden is constructed) by E(M)-state workers (Section 2.1.3). A midden unit indicates the total amount of midden constructed by one worker in her task executing period.

Debris inside the nest

Second, we formulate the dynamics of the amount of debris in the nest (i.e., $dD(t)/dt$). When left unattended, debris spontaneously accumulates in the nest. When nest maintenance workers detect debris in the nest, they carry it outside, decreasing the amount of debris in the nest. Thus, $dD(t)/dt$ is represented by

$$\frac{dD(t)}{dt} = \psi - P_{\mathrm{E(N)}\to\mathrm{R(N)}}(t) W_{\mathrm{E(N)}}(t), \tag{2.8}$$

where ψ is the natural rate of debris inflow and $P_{\mathrm{E(N)}\to\mathrm{R(N)}}$ is the rate at which the task is successfully executed (debris is removed from the nest) by E(N)-state workers (Section 2.1.3).

A debris unit indicates the total amount of debris carried by one worker in her task executing period.

Surplus nutritional energy of the colony

Finally, we formulate the dynamics of the surplus nutritional energy of the colony (i.e., $dA(t)/dt$), which is equal to the difference between the energy contained in food obtained by foragers and the metabolic energy consumed by workers. Thus, $dA(t)/dt$ is represented by

$$\frac{dA(t)}{dt} = \lambda\ \rho_{fdet} W_{\mathrm{E(F)}}(t) - \sum_i u_i W_i(t) \frac{24 \times 60 \times 60}{\tau_{day}}, \tag{2.9}$$

where λ denotes the energy contained in one unit of food and u_i denotes the metabolic energy consumed by a worker in state i. Because ants consume metabolic energy during the nighttime even though they are inactive, we compensate for nighttime consumption of metabolic energy with $\frac{24\times60\times60}{\tau_{day}}$, where τ_{day} is workers' active time in one day. Workers outside the nest consume more energy than workers inside the nest because the outside temperature is higher than the temperature inside the nest [79], so u_i is a value dependent on the workers' state (Table 2.2). Finally, each worker has a limit of surplus nutritional energy, therefore $A(t)$ does not increase when $A(t) \geq \sum_i W_i(t) a_{\max}$ holds.

2.2 Simulation settings

All parameter values were set as shown in Table 2.2. To allow for inactivity at night and during hibernation, we set one day, that is, τ_{day}, to 6.5 h, and one year, that is, τ_{year}, to 210 days, and we simulated $15\tau_{year}$, which is equivalent to the lifespan of the queen. In nature, at the very beginning of colony growth, the *P. barbatus* queen cares for the eggs by herself and produces the first batch of workers, which ranges in number from one to eight [84]. To exclude this behavior, we set the initial state of the colony in our ODE model as follows. It has been observed that the colony increases to 27 workers 3 weeks after the first batch of workers hatches [84], so we assumed that the colony in its initial state has a total of 25 workers (10 brood care workers and 5 inactive workers that are available for each of the other tasks), with a total of $25a_{\max}$ surplus nutritional energy. For environmental conditions, we assumed no midden and no debris in the nest. All calculations were conducted using the ODE23tb solver in MATLAB version R2015a. When colony size decreases to one, the colony is regarded as being extinct, ending the simulation.

To examine the effect of feedback from colony-level properties to individual behaviors on colony performance, we simulated 12 scenarios representing colonies with four types of $X(t)$ and three types of $Y(t)$. The four $X(t)$ types are denoted as follows:

- $\mathbf{S_X(-1)}$: $X(t)$ is set to a negative feedback, which means that $X(t)$ is high when the current colony size is small and low when the current colony size is large.

- **$S_X(0H)$**: $X(t)$ is set without feedback and $w_0 = 10$, which means that $X(t)$ is always a high value.
- **$S_X(0L)$**: $X(t)$ is set without feedback and $w_0 = 100$, which means that $X(t)$ is always a low value.
- **$S_X(+1)$**: $X(t)$ is set to a positive feedback, which means that $X(t)$ is high when the current colony size is large and low when the current colony size is small.

The three $Y(t)$ types are denoted as follows:

- **$S_Y(-1)$**: $Y(t)$ is set to a negative feedback, which means that $Y(t)$ is high when the nutritional energy of the colony is insufficient and low when the nutritional energy of the colony is sufficient.
- **$S_Y(0)$**: $Y(t)$ is set without feedback, which means that $Y(t)$ is always a constant value.
- **$S_Y(+1)$**: $Y(t)$ is set to a positive feedback, which means that $Y(t)$ is high when the nutritional energy of the colony is sufficient and low when the nutritional energy of the colony is insufficient.

2.2.1 Normal environment

First, we simulated the ant life cycle for these 12 scenarios and investigated the process of colony growth and patterns of task allocation over 15 years.

2.2.2 Environmental disturbance

Scenarios other than those involving $S_X(-1)$ did not have the large colony size observed in *P. barbatus* colonies in nature (Fig. 2.3(a)). Additionally, scenarios other than those involving $S_X(-1)$ had a task allocation rate that is not observed in nature (Fig. 2.3(b)). Therefore, in the environmental disturbance analyses, we focused on nutritional state feedback ($S_Y(-1)$ vs. $S_Y(0)$ vs. $S_Y(+1)$) under negative colony size feedback, that is, $S_X(-1)$-type behavior. We assumed three types of environmental disturbances: starvation, midden collapse, and debris inflow, and that one of these occurred during a predefined term (called the disturbance term), and that at the end of the disturbance term, the disturbance stopped. In detail, during the disturbance term (from τ_{dstart} to τ_{dend}) the parameter corresponding to the disturbance (starvation, ρ_{fdet}; midden collapse, θ; debris inflow, ψ) was changed from the value listed in Table 2.2 to a predefined value.

First, for $S_Y(-1)$, $S_Y(0)$ and $S_Y(+1)$, we investigated over what range of the three parameters (ρ_{fdet}, θ, ψ) the colony did not go extinct over 15 years as the degree of disturbance gradually increased from mild to severe. We assumed three types of disturbance terms: a long-term disturbance after maturity, $(\tau_{dstart}, \tau_{dend}) = (5\tau_{year}, 7\tau_{year})$; a short-term disturbance after maturity, $(\tau_{dstart}, \tau_{dend}) = (5\tau_{year}, 5\tau_{year} + 30\tau_{day})$; and a short-term disturbance

before maturity, $(\tau_{dstart}, \tau_{dend}) = (30\tau_{day}, 60\tau_{day})$.

Then, we investigated why each scenario displayed a different level of resilience against disturbances. Based on the survival range in each scenario for the three types of disturbances for the three disturbance terms, we selected parameters such that the colony survives over 15 years in only one of the three scenarios (red dashed lines in Fig. 2.4) and examined the differences between the three scenarios in colony size and task allocation rate during the disturbance term. All of the code for these models has been uploaded to GitHub (https://github.com/tomohaya/antODEsimulator).

2.3 Results and Discussion

2.3.1 Normal environment

For all scenarios except those with $\mathrm{S_X}(-1)$, colony size remained small (Fig. 2.3(a)); in scenarios with $\mathrm{S_X}(-1)$, colony sizes were almost 10,000 (i.e., the number of workers in a mature colony of *P. barbatus*). To eliminate the possibility of unintended effects caused by heuristic parameters, we conducted a sensitivity analysis of the heuristic parameters by changing the parameter values to between 50% and 150% of the original values and investigated colony size and proportion of engaged workers involved in each task after 15 years (see Appendix A). In most cases for scenarios with behavior decisions other than $\mathrm{S_X}(-1)$, colony size was very small, typically fewer than 5000 members. Therefore, we concluded that colony size feedback based on the $\mathrm{S_X}(-1)$ behavior decision is required if the colony is to grow to a size similar to that observed in *P. barbatus* colonies in nature. In addition, in scenarios with $\mathrm{S_X}(0\mathrm{H})$, almost all of the workers became foragers, which is unusual in natural colonies.

We conclude that scenarios without $\mathrm{S_X}(-1)$ did not produce large colonies for the following reason. In scenarios with $\mathrm{S_X}(0\mathrm{H})$, the contact rate $X(t)$ is always high. Thus, workers in state I(B) frequently change their task as a result of frequent contact with recruiting workers. As a result, the proportion of brood care workers, which have a low death rate, remains small and the proportion of foragers, which have a high death rate, remains large (Fig. 2.3(b)). By contrast, in scenarios with $\mathrm{S_X}(0\mathrm{L})$, the contact rate $X(t)$ is always low, and workers in state I(B) rarely change their tasks. The resulting small proportion of foragers (Fig. 2.3(b)) causes the nutritional state of the colony to decline, resulting in an increased death rate from starvation and a reduced hatch rate until the colony eventually becomes extinct. In scenarios with $\mathrm{S_X}(+1)$, the colony becomes extinct for similar reasons to those pertaining to scenarios with $\mathrm{S_X}(0\mathrm{L})$.

Therefore, a high contact rate is required for the survival of an incipient colony that does not have a sufficient number of foragers, but once the colony has grown to medium size and has a sufficient number of foragers, a low contact rate is necessary for the colony to grow large. Thus, negative feedback about colony size serves as a mechanism that permits flexibility in

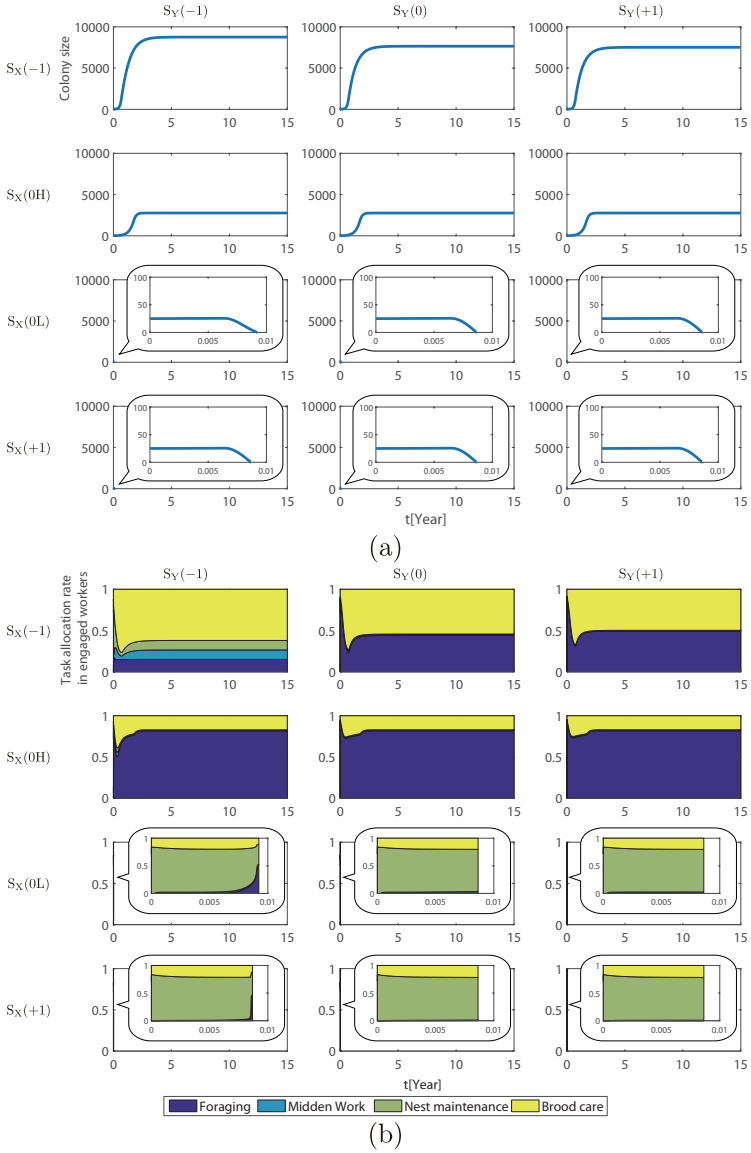


Fig. 2.3 (a) Colony sizes for the 12 scenarios in a normal environment over 15 years and (b) task allocation of engaged workers, that is, $W_{E(F)}$:$W_{E(M)}$:$W_{E(N)}$:$W_{E(B)}$. In the scenarios with $S_X(0L)$ and $S_X(+1)$, the colonies became extinct soon after the simulation started, and colony size and task allocation until extinction are shown in the inset figures.

relation to contact rates. To investigate the influence of initial task allocation, we fixed the number of initial workers at 25 and changed the number of inactive workers for each of the tasks other than blood care (foraging, midden work, and nest maintenance) to either 0, 1, 2, 3, or 4, across all 12 scenarios. In scenarios with $S_X(0L)$ or $S_X(+1)$, colonies were always more likely to become extinct than in scenarios with $S_X(-1)$ or $S_X(0H)$, regardless of initial task allocation.

2.3.2 Disturbance resistance

In the following analysis of disturbance resistance, using only scenarios with $S_X(-1)$, we compare the outcomes of scenarios with $S_Y(-1)$, $S_Y(0)$, and $S_Y(+1)$ in terms of the effects of nutritional state feedback. Against starvation disturbance, the scenario with $S_Y(-1)$ was the most robust in both long-term and short-term disturbances after maturity, whereas the scenario with $S_Y(+1)$ was the most robust in short-term disturbance before maturity (Fig. 2.4(a)). Against both midden collapse and debris inflow disturbances, the scenario with $S_Y(-1)$ was the most robust over all three periods (i.e., long-term disturbances after maturity, short-term disturbances after maturity, and short-term disturbances before maturity) (Fig. 2.4(b) and (c)).

2.3.3 Colony resistance to starvation

Only in the scenario with $S_Y(-1)$ did the colony respond to the threat of both long-term and short-term starvation after maturity by increasing the proportion of foragers (Fig. 2.5(a) and (b)). The results can be understood as follows. When foragers are unable to detect food, in the scenarios with $S_Y(0)$ and $S_Y(+1)$, they recruit fewer inactive workers to the foraging task than in the scenario with $S_Y(-1)$. Hence, the proportion of foragers in the scenarios with $S_Y(0)$ and $S_Y(+1)$ is smaller than in the scenario with $S_Y(-1)$ during disturbance (Fig. 2.5(a) and (b)). Consequently, in scenarios with $S_Y(0)$ and $S_Y(+1)$, surplus nutritional energy becomes negative, which results in reduced worker production. By contrast, the negative feedback about colony nutritional state in the scenario with $S_Y(-1)$ enables colonies to maintain a sufficient proportion of foragers because recruitment behavior is flexible in response to nutritional shortages.

When faced with short-term starvation before maturity, only in the scenario with $S_Y(+1)$ did the colony respond by maintaining a high proportion of foragers (Fig. 2.5(c)). We conclude that colonies survived the threat of early starvation only in the scenario with $S_Y(+1)$ for the following reason. Initially, all colonies had sufficient nutritional energy, and in the scenario with $S_Y(+1)$ the transition to foraging was therefore promoted, whereas in the scenario with $S_Y(-1)$ the transition was suppressed (Fig. 2.5(c)). Once the starvation disturbance commenced, colonies in the scenario with $S_Y(-1)$ transitioned toward foraging, but the proportion of foragers remained smaller than that in the scenario with $S_Y(+1)$, in which foragers were already numerous. Hence, in the scenario with $S_Y(+1)$ resistance to starvation is higher than in the scenario with $S_Y(-1)$.

In scenarios with $S_Y(0)$ and $S_Y(+1)$, colonies have much greater resistance to starvation before maturity than after maturity (see Fig. 2.4(a)) because in these scenarios the proportion of foragers is higher before maturity than after maturity (Fig. 2.5(b) and (c)).

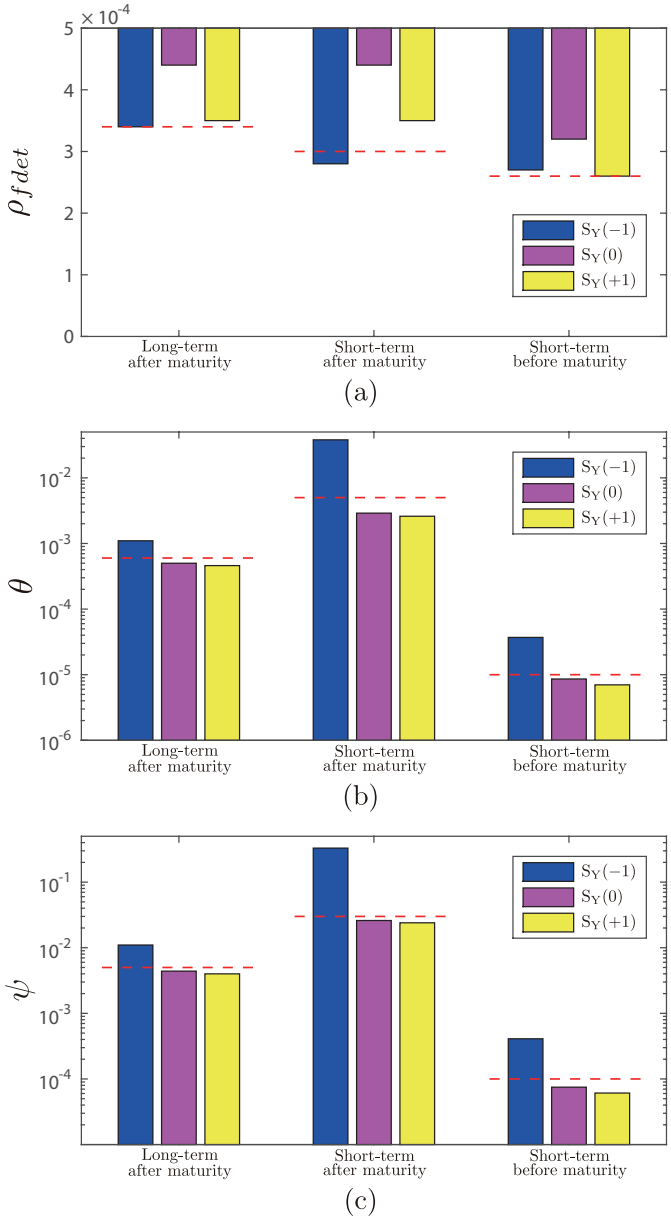


Fig. 2.4 Range of parameters (ρ_{fdet}, θ, ψ) for each scenario in which the colony did not go extinct within 15 years. (a) starvation (b) midden collapse, and (c) debris inflow. In Sections 2.3.3–2.3.5, we focused on the parameters bisected by the red dashed lines.

2.3.4 Colony resistance to midden collapse

Only the colony in the scenario with $S_Y(-1)$ survived midden collapse over all three terms; it did so by increasing the proportion of midden workers (Fig. 2.6).

The relationship between nutritional state feedback and resistance to midden collapse is as

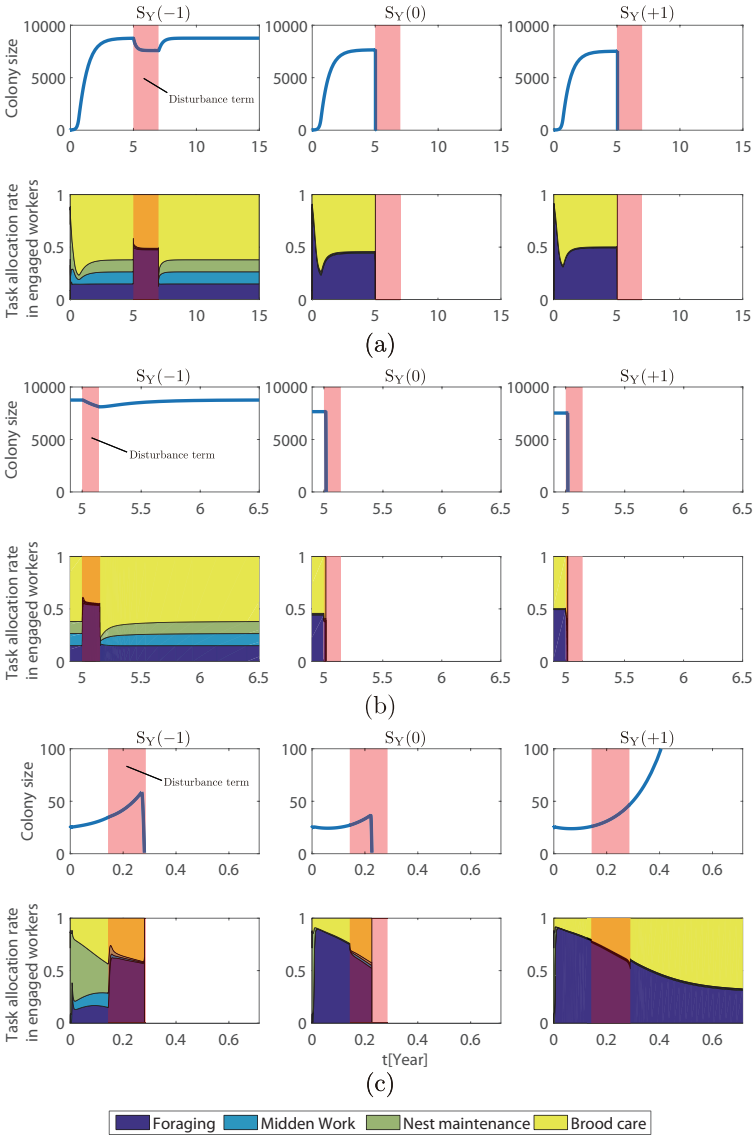


Fig. 2.5 Colony size and task allocation of engaged workers for scenarios with $S_Y(-1)$, $S_Y(0)$, and $S_Y(+1)$ facing starvation in the (a) long term after maturity, (b) short term after maturity, and (c) short term before maturity.

follows. Given that the nutritional energy of the colony is sufficient, the transition to the foraging task is suppressed only in scenarios with $S_Y(-1)$ based on negative nutritional state feedback. Hence, in the scenario with $S_Y(-1)$, a higher proportion of midden workers can be maintained than in scenarios with $S_Y(0)$ and $S_Y(+1)$ (see Fig. 2.3(b)). When the midden collapses, brood care workers are recruited to compensate for the midden work labor shortage. In the scenario with $S_Y(-1)$, a sufficient proportion of midden workers are maintained because of suppression of the transition to the foraging task (Fig. 2.6), but in scenarios with $S_Y(0)$ and

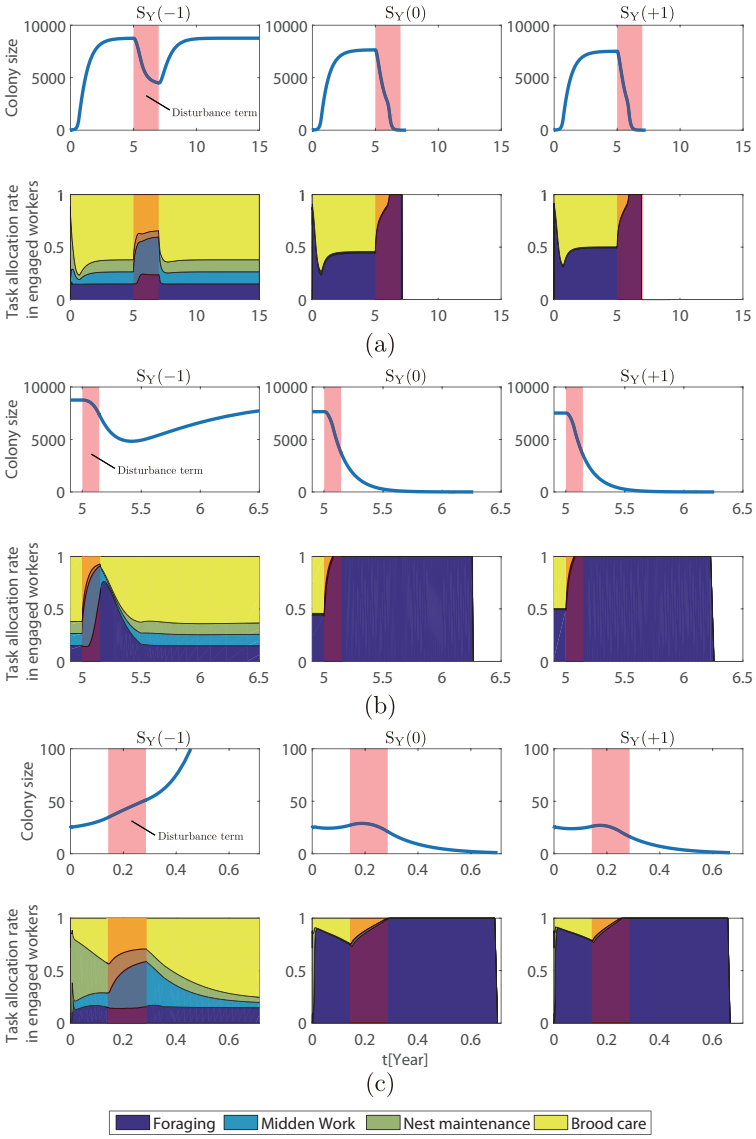


Fig. 2.6 Colony size and task allocation of engaged workers for scenarios with $S_Y(-1)$, $S_Y(0)$, and $S_Y(+1)$ facing midden collapse in the (a) long term after maturity, (b) short term after maturity, and (c) short term before maturity.

$S_Y(+1)$, rate at which brood care workers are recruited to midden work is more because of promotion of the transition to the foraging task, resulting in a shortage of brood care workers and a decline in the hatch rate.

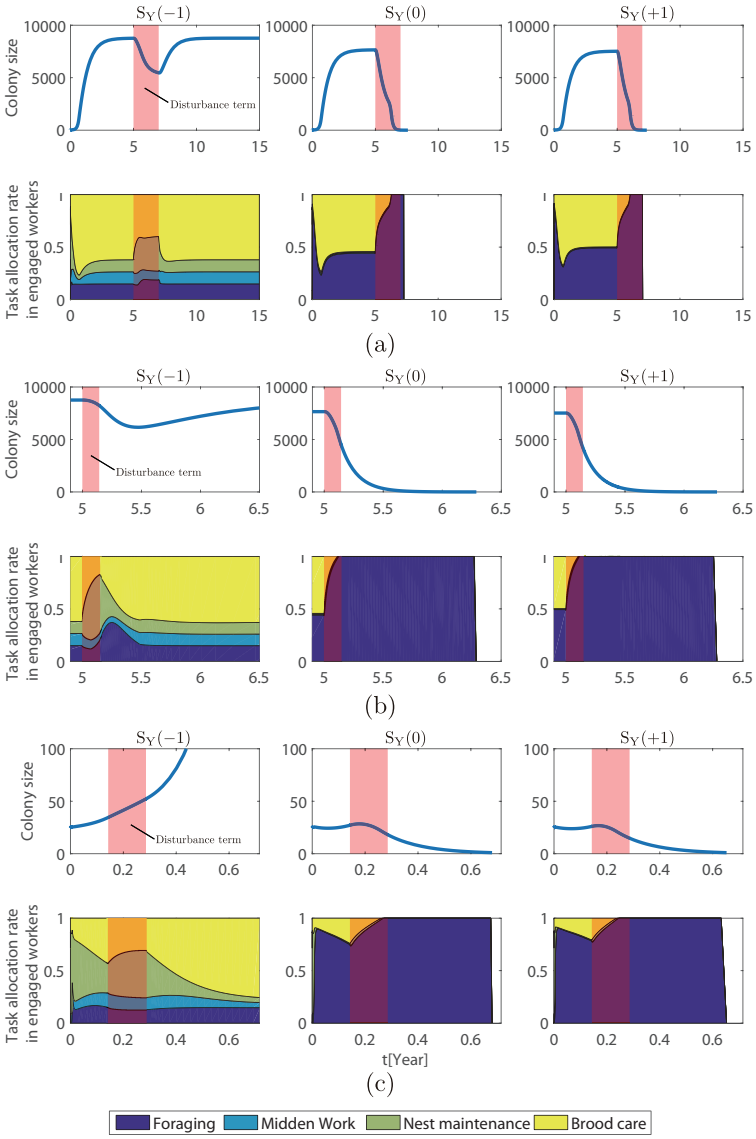


Fig. 2.7 Colony size and task allocation of engaged workers for scenarios with $S_Y(-1)$, $S_Y(0)$, and $S_Y(+1)$ facing debris inflow in the (a) long term after maturity, (b) short term after maturity, and (c) short term before maturity.

2.3.5 Colony resistance to debris inflow

Only the colony in the scenario with $S_Y(-1)$ survived the inflow of debris over all three terms; it did so by increasing the proportion of nest maintenance workers (Fig. 2.7). The relationship between nutritional state feedback and resistance to debris inflow is similar to that in the case of midden collapse.

Throughout our analysis of the dynamics of colony size under disturbance, the minimum size of mature colonies during disturbance term was always 4000 (Figs. 2.5, 2.6, 2.7(a) and (b)) and that for incipient colonies was 25 (Figs. 2.5, 2.6, 2.7(c)). These observations indicate that our stochastic ODE analysis, which allowed for decimals in colony size, did not underestimate extinction associated with extremely small integer colony size.

2.4 Conclusion

In this study, we examined how the modes of feedback from colony-level properties to individual behaviors influenced colony growth and resistance to various types of disturbances of simulated colonies of *P. barbatus*. First, negative feedback from increasing colony size to individual activity (a high level of contact when the colony is small and a low level of contact when the colony reaches medium size) promoted colony growth to the size observed in nature (10,000 workers). Second, negative feedback from a deteriorating nutritional state to foraging activity (numerous foragers were recruited in response to a starvation disturbance, whereas few were recruited in response to midden collapse or debris inflow disturbances) made a greater contribution to colony survival than either no feedback or positive feedback for most types of disturbance.

Because workers do not have a bird's-eye view of their colony, access to colony-level properties is achieved only through local interactions with nestmates. Regarding colony size, local density of nestmates should be positively correlated with colony size, and thus each worker can in principle reflect colony size in her behavior. Interestingly, *L. fuliginosus* is known to adjust the contact rate between nestmates based on colony size by tending to group together when colony size is small and to avoid each other when colony size is large, keeping contact levels between one worker and any nestmates constant irrespective of colony size [53]. This is consistent with a negative feedback rule regarding colony size, that is, scenarios with $\mathrm{S_X}(-1)$, which keeps the contact level between one worker and any nestmates $X(t)W_{in}(t)$ constant, independent of $W_{in}(t)$. Therefore, although this observation and our model are based on different species, our negative feedback rule regarding colony size may play a role in enabling colonies of both species to grow to a large size.

Regarding nutritional state, it has been suggested that social insects are informed of colony nutritional state via frequent trophallaxis [41]. Repeated trophallaxis should result in averaging of individual nutritional states, making each worker aware of the colony's nutritional state [42]. Note that these proximate mechanisms of nestmate interactions were not incorporated into our ODE model. Various social insects, such as the honeybee *A. mellifera* or the ant *L. niger*, increase the number of foragers as the nutritional state of their colony deteriorates [54, 58]. This is consistent with a negative feedback rule regarding the nutritional state of their colony, that is, scenarios with $\mathrm{S_Y}(-1)$, in which $Y(t)$ increases as averaged nutritional energy $\frac{A(t)}{\sum_i W_i(t)}$ decreases. Therefore, the negative feedback from improving states of colony-

level properties to decreasing levels of foraging engagement might be a general tendency that contributes to survival of colonies in the face of disturbances.

Finally, it should be noted that our model was designed to replicate the dynamic properties of *P. barbatus* colonies. Although we believe that the essential feature of adaptive flexibility observed in our simulations can be applied to other social insect species with age polyethism and similar task allocation systems, the modes of feedback from colony-level properties would be more complex and species-specific in real ant colonies [53,57]. Future research should incorporate species-specific characteristics in the model. The resulting understanding of adaptive flexibility and robustness in autonomous distributed systems could be applied to developing more flexible and robust artificial multi-robot control systems.

Chapter 3

Adaptation of a multi-legged robot to inner environment

Natural creatures, for example, humans, animals, insects, and centipedes, use walking as a means of moving. Because walking provides natural creatures with a greater ability to travel on rough terrain than moving using wheels, it is expected to be a useful movement means for a robot used on rough terrain, such as a rescue robot. To date, various types of legged robots, such as biped [85], quadruped [86], hexapod [87], and myriapod [88], have been developed.

Because the locomotion of natural creatures has high stability, adaptability, and robustness, understanding their walking strategy and applying it to a robot control system is beneficial. As a feature of the locomotion of natural creatures, they produce interlimb coordination through interactions with the inner neural system, their musculoskeletal system, and the environment [89]. Interlimb coordination is sometimes modeled as a system of coupled nonlinear oscillators. To date, several types of models of coupled nonlinear oscillators have been developed for bipeds [90, 91], quadrupeds [4, 33, 92, 93], hexapods [5–7, 94], and myriapods [6, 8, 95].

Regarding the inner neural interaction of insects, Bässler and Büschges observed that each leg has an independent pattern generator [96], and the pattern corresponds to the foot movement pattern. Moreover, physically adjacent pattern generators locally interact with each other [30]. Based on these biological observations, hexapod or myriapod models are generally designed as a cluster of autonomous legged modules that determine their own foot movement pattern based on local information from adjacent legged modules [5–8, 94, 95].

In the following, we present the details of existing distributed gait generation systems modeled as coupled nonlinear oscillators for $2 \times N$ type multi-legged robots, such as hexapods or myriapods. Odashima et al. [5] assumed the inner neural interaction of physically adjacent legs, and controlled the phase difference of adjacent oscillators explicitly to achieve static walking not only for translation movement but also turning movement. Inagaki et al. [6] also assumed the inner neural interaction of physically adjacent legs and controlled the gait pattern explicitly based on the desired walking speed. By contrast, Fujiki et al. [7], Ambe et al. [94], Tsuchiya et al. [8], and Yasui et al. [95] assumed not only weak inner neural interaction

but also interaction with the ground. When interaction with the ground is considered, the dynamics of nonlinear oscillators becomes complicated, but the interaction enables adaptive interlimb coordination according to the current ground scenario. In detail, Fujiki et al. [7] and Tsuchiya et al. [8] controlled each leg's phase based on timing when the foot toe in flight touches the ground, and then produced a gait transition that depended on the desired walking speed. Ambe et al. [94] also assumed similar feedback and analyzed the terminal gait pattern mathematically. Yasui et al. [95] controlled each leg's phase based on the foot toe's ground reaction force feedback and achieved gap-crossing behavior similar to a centipede in nature. As a limitation of these distributed gait generation systems, inner neural interaction is coordinated initially based on a $2 \times N$ type leg configuration; hence, these systems cannot directly be applied to a leg configuration other than the $2 \times N$ type.

In the following, we present existing gait generation strategies for an extraordinary leg configuration robot. Kim et al. developed a single-legged reconfigurable modular robot that can connect with others and the cluster composed of the modules can change its number of legs from zero to six [97], where Ha et al. designed each single module's locomotion using reinforcement learning [98]. Whitman et al. [99] also developed a reconfigurable legged robot that can change its configuration from hexapod to myriapod. Kim et al. [97] and Whitman et al. [99] predesigned a corresponding gait pattern for each leg configuration, and a central control unit controlled the movement of all the legs. In the following, we present gait recovery strategies of a hexapod after leg malfunction. Some researchers have developed strategies by which a hexapod generates adaptive locomotion after leg malfunction [100–105]. Koos et al. [100], Cully et al. [101], and Ren et al. [102] proposed learning strategies to obtain adaptive locomotion after leg malfunction. Johnson [103] proposed a centralized system that transits its gait pattern toward another predesigned gait pattern when a specific leg is broken. Schilling et al. [104] and Auf et al. [105] predesigned an interlimb network for the case of leg malfunction based on the hexapod configuration. When a leg was broken, the hexapod achieved stable locomotion based on the predesigned interlimb network. As a limitation of these gait generation strategies for an extraordinary leg configuration robot, they can only generate a gait pattern for the pre-assumed limited types of leg configuration; hence, these strategies cannot be applied directly to general leg configurations.

As shown in previous research, many distributed gait generation strategies have been designed for $2 \times N$ type multi-legged robots. However, little research has been conducted on the gait generation strategy for a robot with a general leg configuration, including one that differs from natural creatures, that is, the $2 \times N$ type. Because much of a robot's functionality results from its morphology (the shape of its body and limbs) [106], a multi-legged robot other than a $2 \times N$ type has the potential to perform several types of task that the $2 \times N$ type cannot perform. Additionally, if each leg is modularized like those in reconfigurable modular robots [26–28,107–109], then the current configuration of the robot can change in response to the encountered scenario. A multi-legged robot whose leg configuration is not limited to the

$2 \times N$ type has several benefits, such as the following:

The first benefit is traversing uneven terrain. To traverse a slope environment, the robot can traverse the slope safely by changing the leg configuration in the direction in which the support polygon enlarges toward the gradient of the slope. Meanwhile, to traverse a gap environment in which the ground exists discontinuously, the robot can traverse the environment safely by changing the leg configuration in the direction in which the leg position fits the existing ground. The second benefit is achieving the manipulation task. Generally, legged robots can use their legs for manipulation [99, 110, 111] and not only for locomotion. The robot can manipulate various shapes of object by changing the leg configuration flexibly in the direction in which the leg fits the shape of the manipulated object. The third benefit is more choices of physical connections. For the case in which one of the connectable places of the robot is broken, the robot has an opportunity to connect with other robots using another connectable place that is in a healthy condition.

To obtain the above benefits, a stable gait pattern generation strategy for various leg configurations is required. It is time-consuming to develop gait patterns individually for all assumed leg configurations, and a centralized system is difficult to control for a large number of modules because of the capacity limit of the central control unit. Therefore, to obtain the above benefits sufficiently, it is necessary to construct a unified distributed control framework to achieve stable locomotion for various leg configurations. Furthermore, a distributed system has no central control unit; hence, if one of the modules is broken, this does not lead to the breakdown of the entire system. Therefore, by constructing a distributed control framework that considers leg malfunction, the robot can adapt to the leg malfunction by changing its gait pattern appropriately.

In this study, we discuss a single-legged reconfigurable modular robot called "KARAKASA" (named after the umbrella ghost, which is a Japanese single-legged ghost). We design an autonomous distributed control system for each legged module by which a cluster (a robot composed of many single-legged module components) with various leg configurations is guaranteed to achieve static walking on planar ground. In Table 3.1, we summarize the features of our system relative to existing systems.

This study is structured as follows: We present the details of the problem setting in Section 3.1, propose the autonomous distributed system in Section 3.2, verify the effectiveness of the system using dynamic simulations in Section 3.3, demonstrate the effectiveness of the system using robot experiments in Section 3.4, and finally, present the conclusion in Section 3.5.

3.1 Problem setting

We developed a single-legged reconfigurable modular robot called "KARAKASA," as shown in Fig. 3.1. Any number of single-legged modules can be connected to form a cluster. We consider a cluster composed of n identical single-legged components and propose an autonomous

Table 3.1 Summary of the features of existing systems and the proposed system for gait generation for multi-legged robots. H denotes hexapod, M denotes myriapod, and V denotes various leg configurations. ◯ is superior to ×.

System	Applicable leg configurations	Recovery from leg malfunction	Distributed control
Odashima [5]	H	×	◯
Inagaki [6]	H&M	×	◯
Fujiki [7]	H	×	◯
Ambe [94]	H	×	◯
Tsuchiya [8]	M	×	◯
Yasui [95]	M	×	◯
Koos [100]	H	◯	×
Cully [101]	H	◯	×
Ren [102]	H	◯	×
Johnson [103]	H	◯	×
Schilling [104]	H	◯	◯
Auf [105]	H	◯	◯
Proposed	V	◯	◯

distributed control scheme for each module to achieve static walking of the cluster on planar ground. Static walking of the cluster means that, during walking, the center of mass (COM) of the cluster is always inside the support polygon, that is, the convex hull of all the foot toes of the legged modules in the stance phase. We denote the COM of the cluster by COM-C, and the COM of each module by COM-M. We summarize the main symbols and their descriptions used in this study in Table 3.2. We provide details of the problem setting in the following subsections.

3.1.1 Characteristics of each single-legged module

Each legged module, denoted by mod_i, is composed of a disc-shaped body of radius r, CPU (STM32F429), a three-cell Li-ion battery, where each cell is 3.7 V, and a three-DOF leg that can move in three-dimensional space, as shown in Fig. 3.1(a). The total weight is approximately 700 g. The three motors for a leg are Dynamixel XM430-W350-R and denoted by motor1, motor2, and motor3 from the body side. Each module has connectable places every 30 degrees on the side of the disc-shaped body, as shown in Fig. 3.1(b), and can physically and electrically connect with a maximum of six other modules. Each mod_i can communicate to exchange information with the adjacent modules using P2P communication via UART. Note that the physical and electrical connections between modules are achieved manually. Each mod_i has its own local coordinate system $\sum_i$, whose origin o_i is fixed at the center of the disc body. The x_i and y_i-axes are also fixed on the body, as shown in Fig. 3.1(b).

We assume that even when a leg of mod_i moves, COM-M of mod_i always remains at the center of the disc body o_i. Note that the weight of the leg and body of each module are 230g

Table 3.2 Description of the main symbols used in this study.

Symbol	Description
n	Number of modules in the cluster
r	Radius of each module's disc body
$\sum_i \equiv (o_i - x_i\ y_i\ z)$	Local coordinate system of mod_i, where o_i be the center of the disc of mod_i
$\theta_i[t] \pmod{2\pi}$	Phase of mod_i at time t
ω_i	Phase velocity of mod_i
h	Length from the ground to the rotational axis of motor2
d	Length from the ground to the highest foot toe position in the flight phase
$\sum_{wi} \equiv (o_i - x_{wi}\ y_{wi}\ z)$	Coordinate system whose x_{wi}-axis is the moving direction of mod_i
${}^i\psi_j \in [0\ 2\pi)$	Angle between the x_{wj}- and x_i-axes with respect to the x_i-axis
v_i	Moving speed of mod_i
$\beta_i \in [0\ 1]$	Stance rate of mod_i
$s_i \equiv \dfrac{2\pi v_i \beta_i}{\omega_i}$	Stride length of mod_i
N_i	Index set of the physically adjacent modules of mod_i
G	Interlimb coordination network, whose vertex is composed of all the modules
$R(i), L(i)$	Index that is the right and left side neighbor of mod_i at G, respectively
$v_{\max}^{\mathrm{st}}, v_{\max}^{\mathrm{fl}}$	Foot toe's maximum moving speed in the stance and flight phases, respectively
${}^i\boldsymbol{V} \equiv \left[{}^iV_x\ {}^iV_y\right]^{\mathrm{T}}$	Cluster's relative desired translational velocity with respect to $\sum_i$
Ω	Cluster's desired angular velocity around COM-C
${}^i\boldsymbol{T}_j$	Two-dimensional coordinate transformation matrix from $\sum_j$ to $\sum_i$
${}^i\boldsymbol{g}_{\mathrm{est}}[t]$	Relative COM-C position estimated by mod_i at time t with respect to $\sum_i$
${}^i\boldsymbol{g} \equiv \left[{}^ig_x\ {}^ig_y\right]^{\mathrm{T}}$	Relative COM-C position with respect to $\sum_i$
${}^i\varphi_j \in (-\pi\ \pi]$	Angle composed of o_i, COM-C, and o_j
σ_i	$\in \{Stance, Flight\}$. Phase state of mod_i
$\gamma(i)$	$R(i)$ if $\sigma_{R(i)} = Stance$, otherwise $R(R(i))$
P	Polygon whose edges are foot toe of $\mathrm{mod}_i(\forall i \in \{i \mid \sigma_i = Stance\})$ and $\mathrm{mod}_{\gamma(i)}$
$D_{i,j}$	Convex hull composed of four points, that is, anterior extreme position and posterior extreme position of mod_i's foot toe, and that of mod_j's foot toe
$l_i^c[t]$	Hop number between mod_i and its communication target modules at time t
${}^i\boldsymbol{p}_j \equiv \left[{}^ip_{jx}\ {}^ip_{jy}\right]^{\mathrm{T}}$	Relative position of o_j with respect to $\sum_i$
v_i^{ideal}	Ideal moving speed of mod_i
$r_i^v \equiv \dfrac{v_i}{v_i^{\mathrm{ideal}}}$	Ratio between the ideal and actual moving speed of mod_i
${}^i\phi_j \equiv \theta_i[t] - \theta_j[t] \pmod{2\pi}$	Phase difference between mod_i and mod_j
s_i^{RL}, s_i^R	Maximum stride length of mod_i such that COM-C $\notin D_{R(i),L(i)}, D_{R(i),i}$, respectively
$s_i^{\min} \equiv \min(s_i^{RL}, s_i^R, 2r)$	Minimum value among s_i^{RL}, s_i^R, and $2r$
$\omega_{\mathrm{opt}}, r_{\mathrm{opt}}^v$	Optimal common value of ω_i and r_i^v, respectively
$m_i^a[t]$, $m_i^b[t]$	Variables of mod_i for maximum consensus at time t

and 470g, respectively (Fig. 3.1(a)). The influence of this assumption on the walking of the cluster is investigated in Appendix B.

In the following, we present how the relative foot toe position of mod_i with respect to $\sum_i$ is determined. Each module has its own pattern generator, and the phase corresponds to its foot toe position. The time-dependent phase of mod_i, that is, $\theta_i[t] \pmod{2\pi}$, changes with time based on the phase velocity of mod_i, that is, ω_i, as shown in Fig. 3.2(a). The foot toe trajectory of mod_i is designed as shown in Fig. 3.2(b). In the following, we explain the

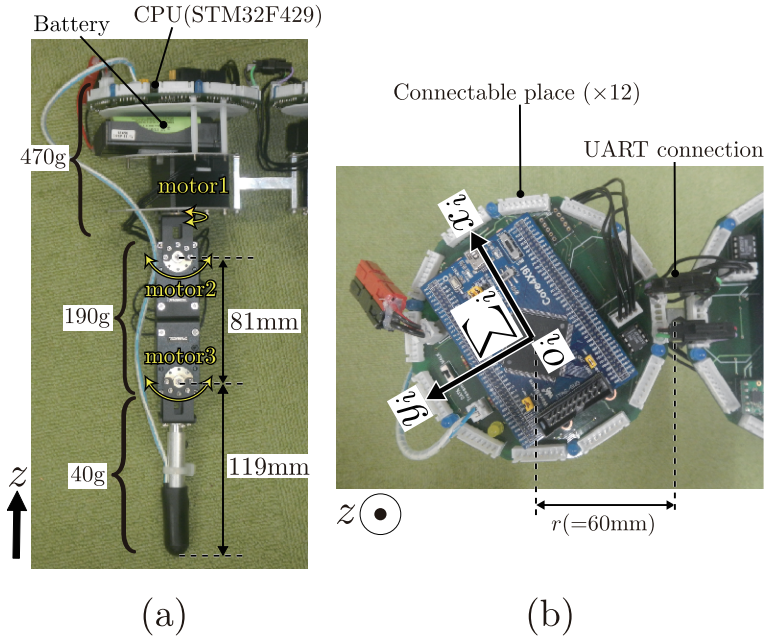


Fig. 3.1 Single-legged modular robot called "KARAKASA" (a) viewed from the side. The leg's degree-of-freedom (DOF) arrangement is yaw, pitch, and pitch from the body side. (b) "KARAKASA" viewed from the top.

variables used in Fig. 3.2(b). The length from the ground to the rotational axis of motor2 is denoted by h and the length from the ground to the highest foot toe position in the flight phase is denoted by d. The moving direction of mod_i is denoted by x_{wi}, and the angle between the x_{wj} and x_i-axes with respect to the x_i-axis is denoted by ${}^i\psi_j$. The moving speed of mod_i is denoted by v_i and the stance rate of mod_i is denoted by β_i. The stride length of mod_i is denoted by s_i, which is equivalent to the product of v_i and the total time of the stance phase, that is, $s_i = \dfrac{2\pi v_i \beta_i}{\omega_i}$. The foot toe's anterior extreme position is denoted by AEP and posterior extreme position is denoted by PEP. Given seven variables $h, d, {}^i\psi_i, v_i, \omega_i, \theta_i[t]$, and β_i, the foot toe position on the trajectory of mod_i with respect to $\sum_i$ is determined uniquely, where the three motor's joint angles are calculated using inverse kinematics to move the foot toe to the position. The lengths h and d are constant values common to all modules given as the initial state, and the remaining five variables, ${}^i\psi_i, \omega_i, v_i, \theta_i[t]$, and β_i, are not common to all legs; they are determined by each mod_i based on the cluster's leg configuration and desired movement.

3.1.2 Leg configuration of the cluster

n legged modules are connected in such a manner that the bodies form a planar shape composed of discs tangent to each other. Different modules' local coordinate systems are not

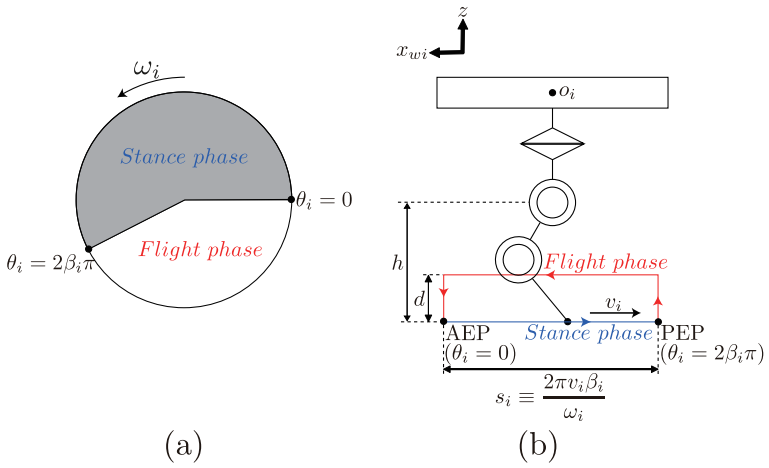


Fig. 3.2 Relationship between (a) the phase and (b) foot toe trajectory. The range of the stance phase is $0 \leq \theta_i(t) < 2\beta_i\pi$ and that of the flight phase is $2\beta_i\pi \leq \theta_i(t) < 2\pi$.

required to be aligned. Examples of clusters are shown in Fig. 3.3. As Fig. 3.3 shows, a cluster's leg configuration is not limited to the $2 \times N$ type, such as hexapod or myriapod, and can be composed of various two-dimensional leg configurations. As a limitation of the applicable leg configurations of our control system, we assume

$$\text{COM-C} \in \text{inside the convex hull of } \{o_1, o_2, \ldots, o_n\} \setminus \{o_i\} \quad (1 \leq \forall i \leq n), \tag{3.1}$$

which means that if any one module is in the flight phase and all the remaining modules are in the stance phase such that their foot toes are under each COM-M, then COM-C is inside the support polygon. Note that $n \geq 5$ is a necessary condition for the condition (3.1) to hold; thus, hexapod and myriapod are included in the applicable leg configurations, but biped and quadruped are excluded. In fact, a quadruped can achieve static walking using the crawl gait, but the crawl gait requires a body shift such that COM-C is always inside the support polygon [112]. In our control system, such a body shift is not considered. Thus, we exclude the quadruped type from the applicable leg configurations, but this simplification contributes to the construction of a distributed control system.

In the following, we assume leg malfunction. As a feature of the Dynamixel motor, the motor's angle is maintained at the same angle after signal lines are cut. Hence, when a module's signal lines are cut for all three motors, the foot toe is maintained at the same position and not used for the cluster's movement. We refer to this case as leg malfunction. In practice, the module's CPU always checks the health condition of the Dynamixel motor through communication. When a leg is broken, the module's CPU does not receive a signal from the motor within a certain time period; thus, the module can detect the leg malfunction.

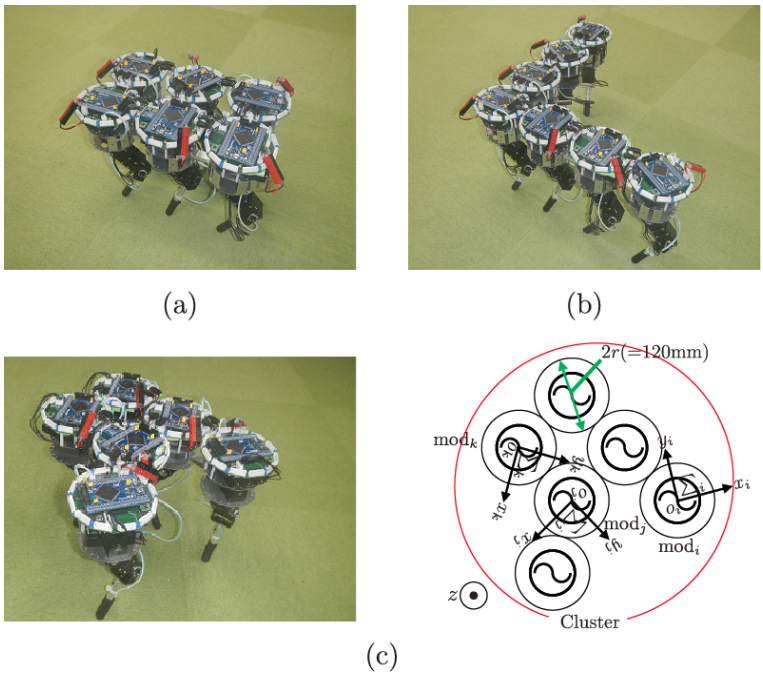


Fig. 3.3 Several types of cluster. (a) Six-legged hexapod cluster. (b) Seven-legged V-shaped cluster. (c) Six-legged asymmetric cluster and its model viewed from the top. Different modules' local coordinate systems are not required to be aligned.

As an extension of the condition (3.1) to cover the case of leg malfunction, we assume

$$\text{COM-C} \in \text{inside the convex hull of } \{o_1, o_2, \ldots, o_n\} \setminus \{o_i, o_{b1}, \ldots, o_{bj}\} \quad (1 \leq \forall i \leq n), \tag{3.2}$$

where $b1 \ldots, bj$ are the indices of the broken modules. Note that $n \geq j + 5$ is a necessary condition for the condition (3.2) to hold.

3.1.3 Distributed control

In our proposed system, all modules' controls are identical and there is no central control unit. Our control system adopts a hierarchical control system, as shown in Fig. 3.4. Each stage in Fig. 3.4 has some target variables of mod_i that are iteratively computed by mod_i. Each module calculates the target variables of all stages described by $[\cdots]$ based on information obtained via communication described by $\langle \cdots \rangle$ and variables in dependent stages, where the dependencies between stages are shown in Fig. 3.4 using arrows.

In the following, we explain the information described by $\langle \cdots \rangle$ in Fig. 3.4 that each module is allowed to obtain via UART communication. First, we describe two types of graphs whose vertices are composed of all modules in a cluster. Let a communication graph be a graph such that all mod_i are adjacent to physically adjacent modules, whose index set is denoted by N_i, as shown in Fig. 3.5(a). Information exchange through the communication graph is performed

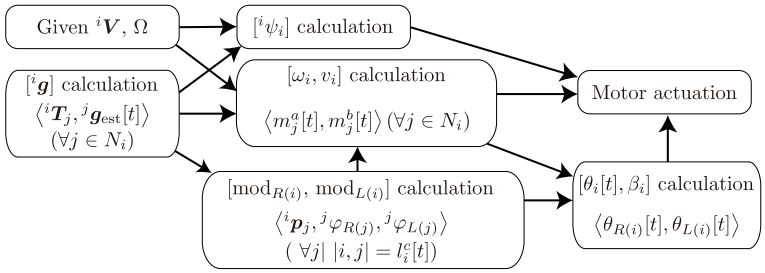


Fig. 3.4 Dependencies between each stage. Each stage at the end of the arrow calculates the target variables denoted by $[\cdots]$ using local information denoted by $\langle\cdots\rangle$, which is obtained via communication, and the target variables of a stage at the start of the arrow.

in the $[{}^i\boldsymbol{g}]$ calculation stage and $[\omega_i, v_i]$ calculation stage in Fig. 3.4, where the purpose of the stages is to agree on the states of all modules. Let interlimb coordination graph G be a graph provided in the $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage in Fig. 3.4 such that all mod_i are adjacent to two modules, that is, $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$, as shown in Fig. 3.5(b). Information exchange through graph G is performed in the $[\theta_i[t], \beta_i]$ calculation stage in Fig. 3.4, where the purpose of the stage is to adjust the phase difference of all modules appropriately. Again, to obtain several benefits of an autonomous distributed system, each module is not allowed to communicate with all the other modules to directly obtain global information, such as the entire leg configuration of the cluster. In our proposed system, each mod_i is allowed to obtain only local information from a limited number of modules (maximum of eight), that is, information about $\mathrm{mod}_j(\forall j \in N_i)$, $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$. When $R(i) \notin N_i$ or $L(i) \notin N_i$, mod_i is allowed to obtain information about the module via multi-hop communication. Additionally, to construct G in the $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage, each mod_i is allowed to communicate with $\mathrm{mod}_j(\forall j|\ |i,j| = l_i^c[t] \leq \max(|i, R(i)|, |i, L(i)|))$, where $|i,j|$ denotes the minimum hop number between mod_i and mod_j on the communication graph in Fig. 3.5(a), and variable $l_i^c[t]$ is explained in Section 3.2.2. We discuss the relationship between $|i,j|(j \in \{R(i), L(i)\})$ and the leg configuration of the cluster in Section 3.5.

In the following, we summarize all information used by each mod_i. The values that are independent of the cluster's leg configuration and desired movement, that is, h, d, r, $v_{\max}^{\mathrm{st}}$, and $v_{\max}^{\mathrm{fl}}$, are given as the initial state, where $v_{\max}^{\mathrm{st}}$ and $v_{\max}^{\mathrm{fl}}$ are the foot toe's maximum moving speed in the stance and flight phases determined by the motor limitation, respectively. Generally, $v_{\max}^{\mathrm{fl}} > v_{\max}^{\mathrm{st}}$ holds because the motor's maximum speed increases as the load torque decreases. The cluster's desired translational and angular velocity, that is, ${}^i\boldsymbol{V}$ and Ω, are also given as the initial state. Translational velocity ${}^i\boldsymbol{V}$ is given to all modules such that ${}^i\boldsymbol{V} = {}^i\boldsymbol{R}_j{}^j\boldsymbol{V}$ holds $\forall j$, where ${}^i\boldsymbol{R}_j$ is the rotation matrix from $\sum_j$ to $\sum_i$. In the robot experiment, to share the cluster's desired movement with all modules, we assign ${}^i\boldsymbol{V}$ and Ω to one mod_i and let the module broadcast the information after ${}^i\boldsymbol{R}_j \forall j \in N_i$ is obtained in the $[{}^i\boldsymbol{g}]$ calculation stage. By contrast, the values that depend on the cluster's leg configuration

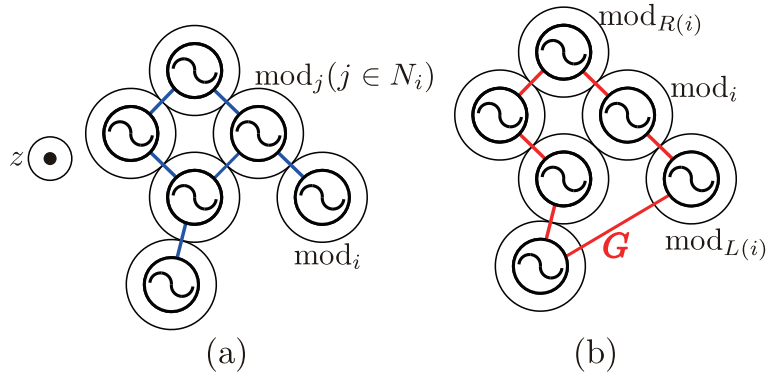


Fig. 3.5 Two types of graphs used in our system. The leg configuration is an example. (a) Communication graph. (b) Interlimb coordination graph G.

and desired movement, that is, $\left({}^i\boldsymbol{T}_j\,,\ {}^j\boldsymbol{g}_{\mathrm{est}}[t],\ m_j^a[t],\ m_j^b[t]\right)$ $(\forall j \in N_i)$, $\left({}^i\boldsymbol{p}_j\,,\ {}^j\varphi_{R(j)},\ {}^j\varphi_{L(j)}\right)$ $(\forall j|\ |i,j| = l_i^c[t])$, $\theta_{R(i)}[t]$, and $\theta_{L(i)}[t]$, are obtained via local communication at each stage in Fig. 3.4. These variables are explained in Section 3.2.

The information processing step of the cluster proceeds as follows: We assume that all the modules in the cluster share global clock t $(t = 1, 2, 3...)$. At every t, each module transmits its current state in addition to receiving the states of the communication target modules simultaneously, and then processes the information. All stages are always performed in parallel, and the convergence of each stage is guaranteed by assuming that the dependent stage has already converged. Note that each mod_i starts the [ω_i, v_i calculation] and [θ_i, β_i calculation] stages after both $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ have been determined. For the case of the leg malfunction of $\mathrm{mod}_j (j \in \{R(i), L(i)\})$, mod_j notifies both mod_i and another module that is adjacent to mod_j at G of its malfunction. Then, mod_i performs the [$\mathrm{mod}_{R(i)}$, $\mathrm{mod}_{L(i)}$] calculation stage again and hence G is reconstructed.

3.1.4 Control purpose

The control purpose of our system is to converge the cluster's gait pattern to satisfy the following four conditions based on the distributed control in Section 3.1.3:

(C1) COM-C is always inside the support polygon.
(C2) The foot toe trajectories of different modules do not interfere physically.
(C3) The foot toe moving speeds of all modules in the stance and flight phases do not exceed the limit speeds $v_{\max}^{\mathrm{st}}$ and $v_{\max}^{\mathrm{fl}}$, respectively.
(C4) The cluster achieves the desired movement, that is, ${}^i\boldsymbol{V}$ and Ω, as close as possible.

Among the seven variables for determining the foot toe position, the variables h and d are constant values given as the initial state, as mentioned. The remaining five variables ${}^i\psi_i, \omega_i, v_i, \theta_i[t]$, and β_i are determined to satisfy the four conditions (C1)–(C4). The approach

used to determine variable ${}^i\psi_i$ is presented in Section 3.2.3, variables ω_i and v_i are presented in Section 3.2.4, and variables $\theta_i[t]$ and β_i are presented in Section 3.2.5.

3.2 Proposed autonomous distributed system

In this section, we present the autonomous distributed system for each module to determine ${}^i\psi_i, \omega_i, v_i, \theta_i[t]$, and β_i that achieve the control purpose the conditions (C1)–(C4). As shown in Fig. 3.4, our control system adopts a hierarchical control system. We explain the details of all the stages in Fig. 3.4 in the following subsections.

3.2.1 COM-C calculation

In the following, we present the method by which all mod_i obtain the relative COM-C position with respect to $\sum_i$, that is, ${}^i\boldsymbol{g}$, in a distributed manner without any prior information on the global configuration of the cluster. To estimate the COM-C, each module uses a strategy similar to trophallaxis of ants that was explained in Capter 2 as follows: Let the coordinate transformation matrix from $\sum_j$ to $\sum_i$ be ${}^i\boldsymbol{T}_j$. Each mod_i has time-dependent variable ${}^i\boldsymbol{g}_{\mathrm{est}}[t]$ and acquires physically adjacent modules' information $\left({}^i\boldsymbol{T}_j, {}^j\boldsymbol{g}_{\mathrm{est}}[t]\right)$ $(\forall j \in N_i)$ via information exchange through a communication graph, and changes ${}^i\boldsymbol{g}_{\mathrm{est}}[t]$ based on the dynamics

$$ {}^i\boldsymbol{g}_{\mathrm{est}}[t+1] = {}^i\boldsymbol{g}_{\mathrm{est}}[t] - \alpha_1 \sum_{\forall j \in N_i} ({}^i\boldsymbol{g}_{\mathrm{est}}[t] - {}^i\boldsymbol{T}_j {}^j\boldsymbol{g}_{\mathrm{est}}[t]), \tag{3.3} $$

where ${}^i\boldsymbol{g}_{\mathrm{est}}[0]$ is $\mathbf{0}$, that is, the COM-M position of mod_i, and α_1 is a constant value such that $\alpha_1 < 1/6$ holds, where 6 is the maximum number of physically adjacent modules. Because Eq.(3.3) coincides with the well-known average-consensus control [35] and α_1 is given as above, time-dependent variable ${}^i\boldsymbol{g}_{\mathrm{est}}[t]$ follows

$$ \lim_{t\to\infty} {}^i\boldsymbol{g}_{\mathrm{est}}[t] = \frac{1}{n}\sum_{j=1}^{n} {}^i\boldsymbol{T}_j {}^j\boldsymbol{g}_{\mathrm{est}}[0] = {}^i\boldsymbol{g}, \tag{3.4} $$

which means that ${}^i\boldsymbol{g}_{\mathrm{est}}[t]$ converges to the position of the average of the COM-M position of all modules among the cluster, that is, ${}^i\boldsymbol{g}$; thus, all mod_i obtain ${}^i\boldsymbol{g}$ based on local information. If o_i overlaps with the COM-C position, then mod_i is excluded from the later strategy because the stride lengths of all modules converge to 0 because of mod_i, as described in Section 3.2.3. In that extra case, the control purpose can be achieved by the remaining modules.

3.2.2 $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ calculation

In the following, we present a method that constructs interlimb coordination graph G, which is time-invariant for one configuration. Note that G has to be recalculated when the cluster's configuration changes as a result of leg malfunction. G contributes to the derivation of two conditions, (3.7) and (3.8), that guarantee the condition (C1). The details are provided later.

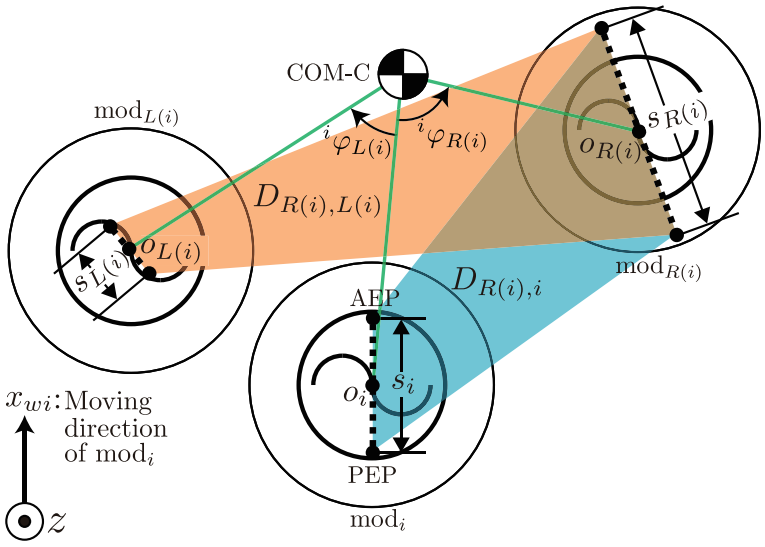


Fig. 3.6 Angles $^{i}\varphi_j (j \in \{R(i), L(i)\})$, and geometric relationship between $D_{R(i),i}$, $D_{R(i),L(i)}$, and COM-C. This figure shows a case in which the desired movement of the cluster is turning. Black dotted lines indicate the foot toe trajectory in the stance phase of each module.

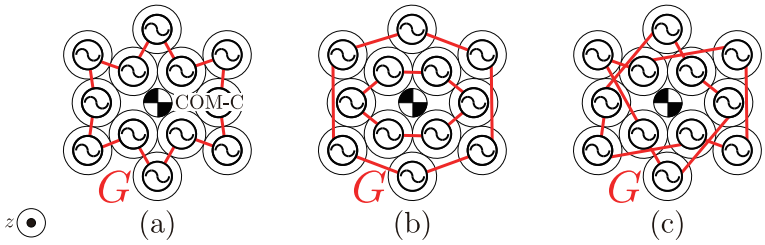


Fig. 3.7 Example of several G for one cluster.

Conditions of G and its application

First, we explain the two conditions of G. Let $^{i}\varphi_j \in (-\pi\ \pi]$ be an angle composed of o_i, COM-C, and o_j, as shown in Fig. 3.6. As a special case, we define $^{i}\varphi_i$ as π to simplify Algorithm 1 described in Section 3.2.2.

Let G be any one graph whose vertex is composed of all modules such that both of the following conditions hold:

$$G \text{ is one or more cycle graphs, that is, all the vertices have degree two} \tag{3.5}$$

$$^{i}\varphi_{R(i)} + |^{i}\varphi_{L(i)}| < \pi \wedge {}^{i}\varphi_{R(i)} \geq 0 \wedge {}^{i}\varphi_{L(i)} \leq 0. \tag{3.6}$$

Graph G is not unique, as shown in Fig. 3.7.

In the following, we present how G contributes to static walking of the cluster. Let $D_{i,j}$ be a convex hull composed of four points, that is, AEP and PEP of mod$_i$ and those of mod$_j$.

When all modules move their foot toes to satisfy both

$$\text{If } \mathrm{mod}_i \text{ is in the flight phase, then both } \mathrm{mod}_{R(i)} \text{ and } \mathrm{mod}_{L(i)} \text{ are in the stance phase} \tag{3.7}$$

$$\text{COM-C} \notin \left(D_{R(i),L(i)} \cup D_{R(i),i}\right), \tag{3.8}$$

COM-C is always inside the support polygon, that is, the condition (C1) holds (see Appendix C). An example of the condition (3.8) is shown in Fig. 3.6. We explain Appendix C briefly. Let $\sigma_i \in \{Stance, Flight\}$ be the phase state of $\mathrm{mod}_i (i \in G_V)$ and let $\gamma(i)$ be an index of a module given as

$$\gamma(i) = \begin{cases} R(i) & \left(\sigma_{R(i)} = Stance\right) \\ R(R(i)) & \left(\sigma_{R(i)} = Flight\right). \end{cases} \tag{3.9}$$

If the relationship between modules in the stance phase and the flight phase is determined based on the condition (3.7), $\sigma_{\gamma(i)} = Stance(\forall i \in \{i|\sigma_i = Stance\})$ holds. Under the condition (3.7), let P be a time-dependent polygon whose edges are all line segments connecting foot toes of $\mathrm{mod}_i(\forall i \in \{i|\sigma_i = Stance\})$ and those of $\mathrm{mod}_{\gamma(i)}$. From the condition (3.6), COM-C is always interior of P, which is composed of foot toes of all modules in the stance phase, under the assumption that foot toes of all mod_i are located under o_i. Moreover, when AEP and PEP of the all modules corresponding to $D_{R(i),L(i)}$ and $D_{R(i),i}$ are determined such that the condition (3.8) holds, COM-C is always interior of P even in the waking. Therefore, we assure that COM-C is always interior of P ($\subseteq$ support polygon) during the walking.

$\mathsf{mod}_{R(i)}$ and $\mathsf{mod}_{L(i)}$ searching algorithm

In this section, we present the algorithm by which all mod_i search both $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ to construct G in a distributed manner. We present the algorithm in Algorithm 1 and explain the details in the following.

Each mod_i has three time-dependent state variables $[R(i)\ L(i)\ l_i^c[t]]^{\mathrm{T}}$, where $l_i^c[t]$ is the hop number between mod_i and its communication target modules at time t, initialized to 1. Let the relative position of o_j with respect to $\sum_i$ be ${}^i\boldsymbol{p}_j$. Each mod_i can calculate ${}^i\varphi_j$ based on ${}^i\boldsymbol{p}_j$, which is obtained through communication. Algorithm 1 is designed to change $R(i)$ and $L(i)$ with time such that ${}^i\varphi_{R(i)}$ and $|{}^i\varphi_{L(i)}|$ monotonically decrease. Fig. 3.8 shows an example in which ${}^i\varphi_{R(i)}$ and $|{}^i\varphi_{L(i)}|$ monotonically decrease with time based on Algorithm 1. In Algorithm 1, each mod_i communicates with more distant modules by incrementing $l_i^c[t]$ with time using a breadth-first search algorithm until the following condition holds:

$$R(L(i)) = i\ \wedge L(R(i)) = i\ \wedge {}^i\varphi_{R(i)} + |{}^i\varphi_{L(i)}| < \pi \wedge {}^i\varphi_{R(i)} \geq 0 \wedge {}^i\varphi_{L(i)} \leq 0. \tag{3.10}$$

When Eq.(3.10) holds $\forall i$, the construction of G is complete. To avoid complexity, Algorithm 1 excludes the case in which different origins o_i and o_j are on the same half-line from COM-C.

In the following, we prove that G is always constructed by Algorithm 1. As mentioned, each mod_i changes $R(i)$ and $L(i)$ for monotonically decreasing ${}^i\varphi_{R(i)}$ and $|{}^i\varphi_{L(i)}|$. Therefore, it is

Algorithm 1 $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ searching algorithm.

Initialize:

$[R(i)\ L(i)\ l_i^c[0]]^{\mathrm{T}} = [i\ i\ 1]^{\mathrm{T}}$

while Eq.(3.10) does not hold **do**

Get ${}^i\boldsymbol{p}_j$, ${}^j\varphi_{R(j)}$ and ${}^j\varphi_{L(j)}$ $(\forall j | |i,j| = l_i^c[t])$.

Calculate ${}^i\varphi_j$ based on ${}^i\boldsymbol{p}_j$.

if ${}^i\varphi_j > 0 \wedge {}^i\varphi_j < {}^i\varphi_{R(i)} \wedge {}^i\varphi_j < |{}^j\varphi_{L(j)}|$ **then**

$R(i) \Leftarrow j$

end if

if ${}^i\varphi_j < 0 \wedge |{}^i\varphi_j| < |{}^i\varphi_{L(i)}| \wedge |{}^i\varphi_j| < {}^j\varphi_{R(j)}$ **then**

$L(i) \Leftarrow j$

end if

$l_i^c[t+1] = l_i^c[t] + 1$

end while

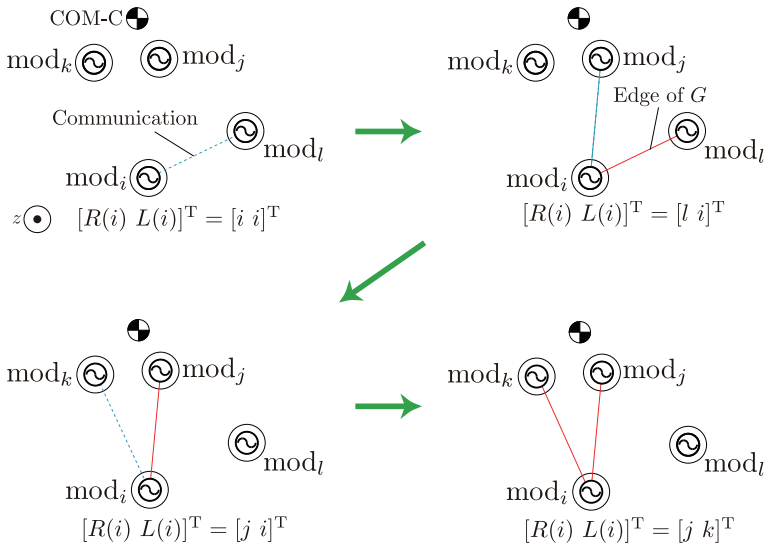


Fig. 3.8 Procedure of G construction. Red lines indicate the edge of G and blue dotted lines indicate that modules connected by the line are communicating. Angles ${}^i\varphi_{R(i)}$ and $|{}^i\varphi_{L(i)}|$ monotonically decrease with time.

clear that $W = \sum_{i=1}^{n}({}^i\varphi_{R(i)} + |{}^i\varphi_{L(i)}|)$ monotonically decreases with time. Because the total number of W is finite, the value of W converges to one value within a finite time. Moreover, from the condition (3.1) and Lemma 1 (see Appendix C.1), $G \neq \emptyset$. Thus, Algorithm 1 always obtains $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ such that Eq.(3.10) holds for mod_i $\forall i$ within a finite time. Therefore, graph G is always constructed. After G is constructed, all mod_i are allowed to communicate with $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$.

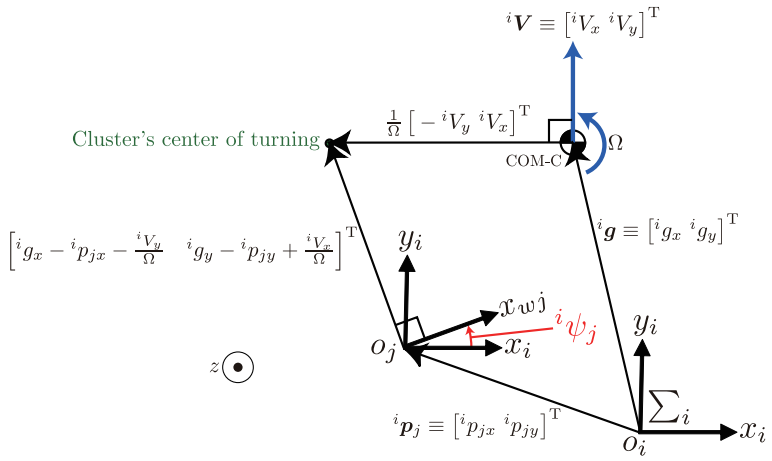


Fig. 3.9 Geometric relationship of $\sum_i$, ${}^i\boldsymbol{g}$, ${}^i\boldsymbol{p}_j$, ${}^i\boldsymbol{V}$, and Ω.

3.2.3 Control purpose formulation

In Section 3.1, control purposes the conditions (C1)–(C4) were given, but mathematically sufficient conditions for the conditions (C1)–(C4), that is, conditions of walking variables ${}^i\psi_i, v_i, w_i, \theta_i[t]$ and β_i, have not been formulated yet. In the following, we formulate mathematically sufficient conditions for the conditions (C1)–(C4) using the discussions in Sections 3.2.1 and 3.2.2. Because the conditions (C2) and (C3) are easy to formulate, we formulate the mathematically sufficient conditions for the conditions (C1) and (C4). Because the discussion of the condition (C4) can be used for that of the condition (C1), we formulate the condition (C4) first.

The condition (C4) formulation

In the following, we present the moving direction and moving speed of each mod_i such that the condition (C4) holds. First, we consider each module's moving direction to achieve the cluster's desired moving direction. We consider ${}^i\psi_j (\forall j \in \{i, R(i), L(i)\})$ such that mod_j's moving direction is vertical to the cluster's center of turning point, where ${}^i\psi_{R(i)}$ and ${}^i\psi_{L(i)}$ are used for the condition (C1) formulation in Section 3.2.3. Because the geometric relationship in Fig. 3.9 holds, angle ${}^i\psi_j$ can be represented as follows using ${}^i\boldsymbol{g}, {}^i\boldsymbol{p}_j, {}^i\boldsymbol{V}$, and Ω:

$$
{}^i\psi_j = \begin{cases} \mathrm{atan2}({}^iV_y, {}^iV_x) & (\Omega = 0) \\ \mathrm{atan2}\left((-{}^ig_x + {}^ip_{jx} + \dfrac{{}^iV_y}{\Omega})\mathrm{sgn}(\Omega), ({}^ig_y - {}^ip_{jy} + \dfrac{{}^iV_x}{\Omega})\mathrm{sgn}(\Omega)\right) & (\Omega \neq 0). \end{cases} \tag{3.11}
$$

The value $\mathrm{atan2}(y, x) \in [0\ 2\pi)$ is $\mathrm{atan}(y/x)$ with its quadrant using two parameters x and y. Note that all mod_i have obtained information ${}^i\boldsymbol{p}_j (\forall j \in \{R(i), L(i)\})$ in Section 3.2.2.

Then, we consider each module's ideal moving speed to achieve the cluster's desired moving

speed. The ideal moving speed of mod_i is described by v_i^{ideal} (not equivalent to v_i). We consider $v_j^{\mathrm{ideal}}(\forall j \in \{i, R(i), L(i)\})$, where $v_{R(i)}^{\mathrm{ideal}}$ and $v_{L(i)}^{\mathrm{ideal}}$ are also used for the condition (C1) formulation in Section 3.2.3. The ratio between v_j^{ideal} and $|{}^i\boldsymbol{V}|$ is equivalent to the ratio between the distance from the center of turning to o_j and distance from the center of turning to COM-C. From the geometric relationship in Fig. 3.9, v_j^{ideal} are calculated based on ${}^i\boldsymbol{g}, {}^i\boldsymbol{p}_j, {}^i\boldsymbol{V}$, and Ω as

$$v_j^{\mathrm{ideal}} = |\begin{bmatrix}\Omega\left({}^ig_x - {}^ip_{jx}\right) - {}^iV_y & \Omega\left({}^ig_y - {}^ip_{jy}\right) + {}^iV_x\end{bmatrix}^{\mathrm{T}}|. \tag{3.12}$$

When the ideal foot toe moving speeds of one or more modules are beyond the motor's speed limitation, let all modules decrease their moving speed at an agreed rate to allow the cluster to walk on the desired trajectory in which all module's speeds are within the motor's speed limitation. Additionally, from the condition (C4), the error between the ideal and actual speed should be as small as possible. The condition (C4) can be formulated as follows: Let $r_i^v \equiv \dfrac{v_i}{v_i^{\mathrm{ideal}}}$ be the ratio between the ideal and actual moving speed of mod_i. Given ${}^i\psi_i$ and v_i^{ideal} by Eq.(3.11) and Eq.(3.12), respectively, and when $r_i^v = r_j^v \forall j$ holds and $r_i^v (\leq 1)$ is maximized, then the condition (C4) holds.

The condition (3.7) formulation

In the following, we present the stance rate range of each mod_i such that the condition (3.7) holds. We denote the phase difference between mod_i and $\mathrm{mod}_j(\forall j \in \{R(i), L(i)\})$ by ${}^i\phi_j \equiv \theta_i[t] - \theta_j[t] \pmod{2\pi}$. If $\omega_i = \omega_j$ and

$$\left(\frac{{}^i\phi_j}{2\pi} \leq \beta_i \leq 1\right) \wedge \left(\frac{{}^j\phi_i}{2\pi} \leq \beta_j \leq 1\right), \tag{3.13}$$

then both mod_i and mod_j are not in the flight phase simultaneously, as shown in Fig. 3.10, where Fig. 3.10 shows the minimum stance rate case, that is, $\beta_i = \dfrac{{}^i\phi_j}{2\pi}$ and $\beta_j = \dfrac{{}^j\phi_i}{2\pi}$. Considering that G is a cycle graph, if $\omega_i = \omega_{R(i)} \forall i$ and

$$\left(\frac{{}^i\phi_{R(i)}}{2\pi} \leq \beta_i \leq 1\right) \wedge \left(\frac{{}^i\phi_{L(i)}}{2\pi} \leq \beta_i \leq 1\right) \tag{3.14}$$

holds for $\mathrm{mod}_i \forall i$, then the condition (3.7) holds.

The condition (3.8) formulation

In the following, we present the stride length range of each mod_i such that the condition (3.8) holds. Assuming that $\beta_i = \beta_{R(i)}$, $\omega_i = \omega_{R(i)}$, and $r_i^v = r_{R(i)}^v \forall i$, then $s_j(\forall j \in \{R(i), L(i)\})$ depend on s_i linearly as follows:

$$s_j = \frac{2\pi v_j \beta_j}{\omega_j} = \frac{2\pi v_j \beta_i}{\omega_i} = \frac{v_j^{\mathrm{ideal}}}{v_i^{\mathrm{ideal}}} s_i. \tag{3.15}$$

When Eq.(3.15) holds, let the maximum stride length of mod_i that satisfies the condition that COM-C $\notin D_{R(i),L(i)}$ be s_i^{RL} and that for $D_{R(i),i}$ be s_i^R as shown in Fig. 3.11. Using

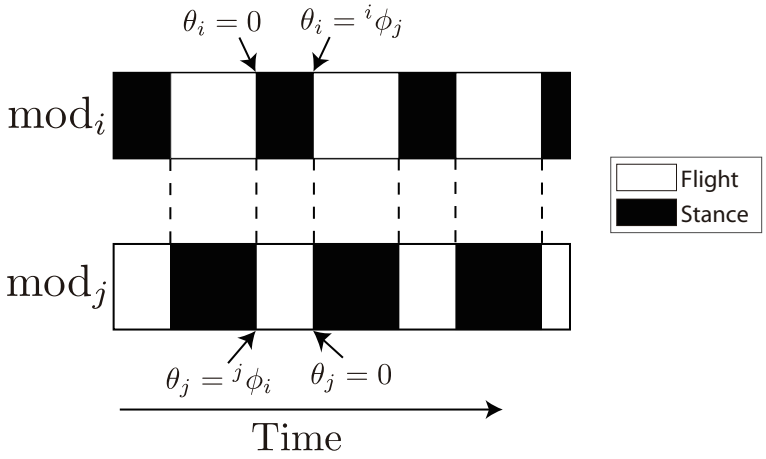


Fig. 3.10 Minimum stance rate of mod_i and mod_j such that mod_i and mod_j are not in the flight phase simultaneously. Minimum stance rate depends on their phase difference.

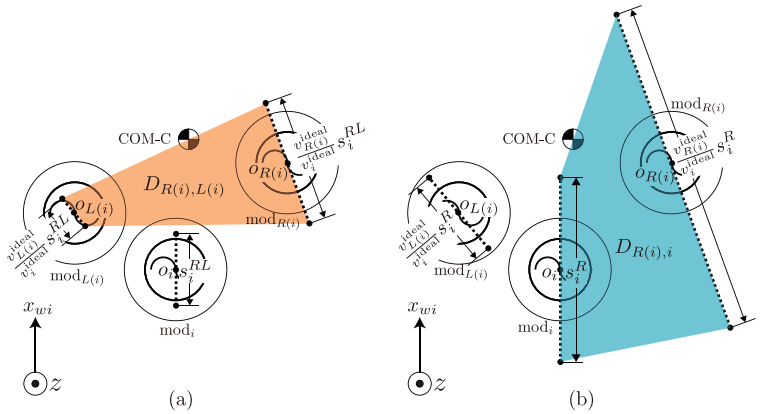


Fig. 3.11 Scenario in which the stride length of mod_i is (a) s_i^{RL} and (b) s_i^R.

Eq.(3.15) and the previously obtained $\left({}^i\psi_j, v_j^{\mathrm{ideal}}\right)$ $(\forall j \in \{R(i), L(i)\})$, s_i^{RL} and s_i^R are determined geometrically (see Appendix D). If $s_i \leq \min\left(s_i^{RL}, s_i^R\right)$ holds, then the condition (3.8) holds.

Control purpose formulation summary

In the following, we summarize the mathematically sufficient conditions for the conditions (C1)–(C4) and then present the desired values of ${}^i\psi_i, \omega_i, v_i, \theta_i[t]$, and β_i such that all the conditions (C1)–(C4) hold.

First, given ${}^i\psi_i$ and v_i^{ideal} by Eq.(3.11) and Eq.(3.12), respectively, the undetermined walking variables of mod_i are decreased to $r_i^v, w_i, \theta_i[t]$, and β_i. From the discussion in Section 3.2.3,

the sufficient condition for the condition (3.7) is as follows:

$$\beta_i \geq \max(\frac{{}^i\phi_{R(i)}}{2\pi}, \frac{{}^i\phi_{L(i)}}{2\pi}) \wedge \omega_i = \omega_{R(i)}. \tag{3.16}$$

From the discussion in Section 3.2.3, the sufficient condition for the condition (3.8) is as follows:

$$s_i \leq \min\left(s_i^{RL}, s_i^R\right) \wedge \omega_i = \omega_{R(i)} \wedge \beta_i = \beta_{R(i)} \wedge r_i^v = r_{R(i)}^v. \tag{3.17}$$

When s_i is not larger than its body diameter $2r$, the foot toe trajectories of different modules do not interfere physically with each other. Therefore, the sufficient condition for the condition (C2) is

$$s_i \leq 2r. \tag{3.18}$$

If mod_i's foot toe speed in the stance phase v_i does not exceed $v_{\max}^{\mathrm{st}}$ and mod_i's foot toe speed in the flight phase, that is, the value for which the flight trajectory length is divided by the total flight time, does not exceed $v_{\max}^{\mathrm{fl}}$, then the foot toe moving speed can be achieved by the motors. Therefore, the sufficient condition for the condition (C3) is

$$v_i \leq v_{\max}^{\mathrm{st}} \wedge \frac{s_i + 2d}{\frac{2\pi}{\omega_i}(1-\beta_i)} \leq v_{\max}^{\mathrm{fl}}. \tag{3.19}$$

Finally, from the discussion in Section 3.2.3, the sufficient condition for the condition (C4) is

$$r_i^v = r_j^v \forall j \wedge \text{maximize } r_i^v (\leq 1). \tag{3.20}$$

In the following, we determine the variables $r_i^v, w_i, \theta_i[t]$, and β_i to satisfy the above five conditions Eqs.(3.16)–(3.20). From Eqs.(3.17), (3.18), and (3.19), stance rate β_i should be as small as possible to maximize r_i^v. Therefore, β_i is given by the minimum value under the condition Eq.(3.16) as follows:

$$\beta_i = \max(\frac{{}^i\phi_{R(i)}}{2\pi}, \frac{{}^i\phi_{L(i)}}{2\pi}). \tag{3.21}$$

Only if ${}^i\phi_{R(i)} = \pi \forall i$, then $\max_i \beta_i$ has the minimum value 0.5. Because G is a cycle graph, if ${}^i\phi_{R(i)}$ is $\pi \forall i$, then ${}^i\phi_{L(i)}$ is also $\pi \forall i$. Each mod_i can control θ_i, but it is difficult to control the phase difference ${}^i\phi_{R(i)}$ to achieve an arbitrary value (in this case, π) because the control of each module is identical. We assume that if $\omega_i = \omega_{R(i)}$ holds $\forall i$, then ${}^i\phi_{R(i)} \forall i$ converges near to π in the range of the following:

$$\left({}^i\phi_{R(i)} = {}^{R(i)}\phi_{R(R(i))}\right) \wedge \left(\frac{\pi}{2} \leq {}^i\phi_{R(i)} \leq \frac{3\pi}{2}\right), \tag{3.22}$$

where the dynamics of θ_i is proposed to satisfy the assumption Eq.(3.22) in Section 3.2.5. To the best of our knowledge, there is no other dynamics of θ_i that can converge ${}^i\phi_{R(i)} \forall i$ nearer to π than Eq.(3.22).

Given β_i by Eq.(3.21) and if Eq.(3.22) holds, then $\beta_i = \beta_{R(i)} \leq 0.75$ holds $\forall i$. Additionally, $v_i \equiv r_i^v v_i^{\text{ideal}}$, $s_i \equiv \dfrac{2\pi r_i^v v_i^{\text{ideal}} \beta_i}{\omega_i}$, $s_i^{\min} \equiv \min(s_i^{RL}, s_i^R, 2r)$, and considering that r_i^v is a common value to all modules, then Eq.(3.16) is rewritten as follows:

$$\omega_i = \omega_{R(i)}, \tag{3.23}$$

Eqs.(3.17) and (3.18) are rewritten as follows:

$$r_i^v \leq \frac{2\omega_i}{3\pi \max_i \left(\dfrac{v_i^{\text{ideal}}}{s_i^{\min}} \right)} \wedge \omega_i = \omega_{R(i)} \wedge r_i^v = r_{R(i)}^v \wedge \text{ Eq.(3.22) holds}, \tag{3.24}$$

and Eq.(3.19) is rewritten as follows:

$$r_i^v \leq \frac{v_{\max}^{\text{st}}}{\max_i \left(v_i^{\text{ideal}} \right)} \wedge r_i^v \leq \frac{\pi v_{\max}^{\text{fl}} - 4d\omega_i}{3\pi \max_i (v_i^{\text{ideal}})} \wedge \ \omega_i = \omega_{R(i)} \wedge \text{ Eq.(3.22) holds}. \tag{3.25}$$

A set of ω_i and r_i^v that satisfies the above inequality conditions Eqs.(3.20), (3.24), and (3.25) is shown as the green area in Fig. 3.12. One of scenarios (a), (b), or (c) in Fig. 3.12 is determined depending on $\max_i \left(\dfrac{v_i^{\text{ideal}}}{s_i^{\min}} \right)$ and $\max_i \left(v_i^{\text{ideal}} \right)$ values and the scenarios differ in the maximum r_i^v value. One of the ω_i and r_i^v sets that maximizes r_i^v is a red point in Fig. 3.12, which is denoted by ω_{opt} and r_{opt}^v, and given as follows:

$$\omega_{\text{opt}} = \frac{3\pi}{2} r_{\text{opt}}^v \max_i \left(\frac{v_i^{\text{ideal}}}{s_i^{\min}} \right) \tag{3.26}$$

$$r_{\text{opt}}^v = \min \left(1, \frac{v_{\max}^{\text{st}}}{\max_i (v_i^{\text{ideal}})}, \frac{v_{\max}^{\text{fl}}}{3 \max_i (v_i^{\text{ideal}}) + 6d \max_i \left(\dfrac{v_i^{\text{ideal}}}{s_i^{\min}} \right)} \right). \tag{3.27}$$

When all mod_i determine ω_i as Eq.(3.26), r_i^v as Eq.(3.27), θ_i such that Eq.(3.22) holds, and β_i as Eq.(3.21), then Eqs.(3.16)–(3.20) are satisfied and therefore all the conditions (C1)–(C4) conditions are satisfied. In the next subsections, we present how each mod_i determines those four variables in a distributed manner.

3.2.4 ω_i and v_i calculation

In the following, we present a method for mod_i to determine ω_i as Eq.(3.26) and r_i^v as Eq.(3.27) based on local information. To determine ω_i as Eq.(3.26) and r_i^v as Eq.(3.27), the $\max_i \left(\dfrac{v_i^{\text{ideal}}}{s_i^{\min}} \right)$ value and $\max_i \left(v_i^{\text{ideal}} \right)$ value are required. To share those values with all modules, we use the well-known maximum consensus [113]. Let $m_i^k[t] (\forall k \in \{a, b\})$ be variables of mod_i at time t following

$$m_i^k[t+1] = \max(m_i^k[t], m_j^k[t] (\forall j \in N_i)). \tag{3.28}$$

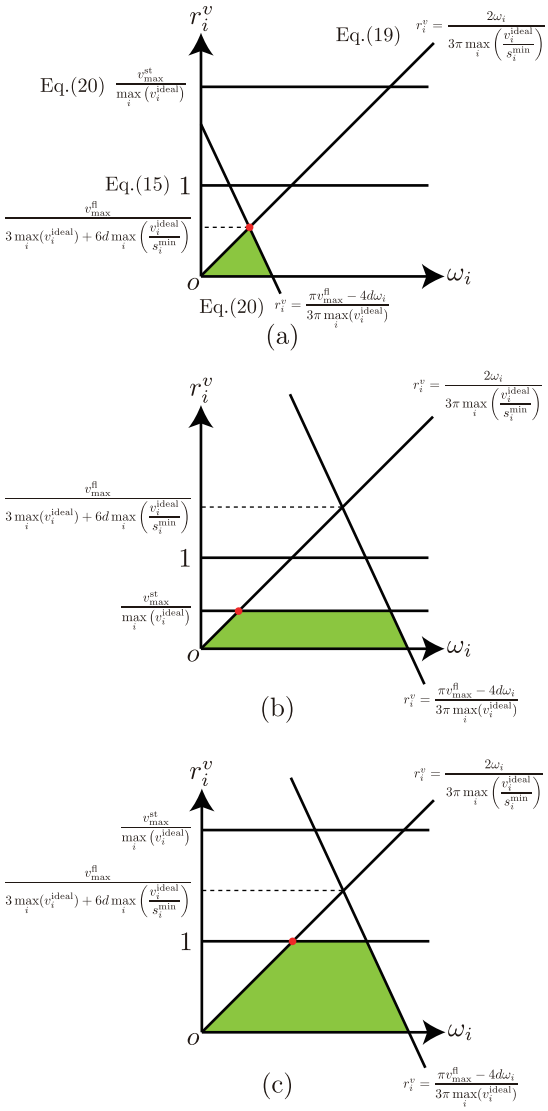


Fig. 3.12 Area satisfying all inequality conditions Eqs.(3.20), (3.24), and (3.25). The area is indicated in green. Scenarios (a), (b), and (c) have different $\max_i\left(\frac{v_i^{\text{ideal}}}{s_i^{\min}}\right)$ and $\max_i\left(v_i^{\text{ideal}}\right)$ values. One of the optimal points $\left(\omega_{\text{opt}}, r_{\text{opt}}^v\right)$ for each case that maximizes r_i^v is marked as a red point.

Given that $m_i^a[0] = \dfrac{v_i^{\text{ideal}}}{s_i^{\min}}$ and $m_i^b[0] = v_i^{\text{ideal}}$, then $m_i^a[t]$ converges to $\max_i\left(\dfrac{v_i^{\text{ideal}}}{s_i^{\min}}\right)$ and $m_i^b[t]$ converges to $\max_i\left(v_i^{\text{ideal}}\right)$ within a finite time [113]. Therefore, given ω_i and r_i^v as

$$\omega_i = \frac{3\pi}{2} r_i^v m_i^a[t] \tag{3.29}$$

$$r_i^v = \min\left(1, \frac{v_{\max}^{\text{st}}}{m_i^b[t]}, \frac{v_{\max}^{\text{fl}}}{3m_i^b[t] + 6dm_i^a[t]}\right), \tag{3.30}$$

ω_i and r_i^v converge to ω_{opt} and r_{opt}^v, respectively, within a finite time based on local information.

3.2.5 $\theta_i[t]$ and β_i calculation

In the following, we present the dynamics of each θ_i such that the phase difference converges in the range of Eq.(3.22). As methods of phase difference control, various types of dynamics for nonlinear coupling oscillators have been well studied, not only for central pattern generators for robot locomotion but also for decision-making in animal groups and clock synchronization [32]. We design a phase difference control system based on the well-known Kuramoto model [114].

We create a potential function that has a minimal value when ${}^i\phi_{R(i)} = \pi \forall i$, and use the well-known gradient descent method to control ${}^i\phi_{R(i)}$. Each mod_i acquires local information $\theta_j[t](\forall j \in \{R(i), L(i)\})$ by communicating with mod_j, and changes its phase $\theta_i[t]$ based on

$$\begin{aligned}\theta_i[t+1] =& \theta_i[t] + \omega_i dt - \alpha_2 \frac{\partial}{\partial \theta_i}\Big(\sum_i \cos {}^i\phi_{R(i)}\Big) \\ =& \theta_i[t] + \omega_i dt + \alpha_2 \left(\sin(\theta_i[t] - \theta_{R(i)}[t]) + \sin(\theta_i[t] - \theta_{L(i)}[t])\right), \end{aligned} \tag{3.31}$$

where dt is the real-time period per $\theta_i[t]$ update and α_2 is the step size of the gradient descent method. Let $\theta_i[0]$ be set to a random value. If $\omega_i \forall i$ are the same value, then ${}^i\phi_{R(i)} \forall i$ converge in the range of Eq.(3.22) based on Eq.(3.31) (see Appendix E), where the initial phase set $[\theta_1[0], \theta_2[0], \cdots, \theta_n[0]]^{\mathrm{T}}$ determines the terminal phase difference values.

Given ${}^i\phi_{R(i)}$ and ${}^i\phi_{L(i)}$, mod_i determines the β_i value by Eq.(3.21). As mentioned in Section 3.2.3, the terminal β_i value satisfies $\beta_i \leq 0.75$.

3.2.6 Motor actuation

After both $\text{mod}_{R(i)}$ and $\text{mod}_{L(i)}$ are determined, each mod_i actuates its motors and moves the foot toe to the position given by h, d, ${}^i\psi_i$ by Eq.(3.11), ω_i by Eq.(3.29), v_i by Eqs.(3.12) and (3.30), $\theta_i[t]$ by Eq.(3.31), and β_i by Eq.(3.21). Note that variables $\omega_i, v_i, {}^i\phi_{R(i)}, {}^i\phi_{L(i)}$, and β_i change with time before they converge.

Because we do not guarantee that $s_i \leq 2r$ before $\omega_i, v_i, {}^i\phi_{R(i)}, {}^i\phi_{L(i)}$, and β_i converge, the foot toe position may interfere with an adjacent module's foot toe. Thus, if mod_i's desired foot toe position is out of its body radius, then mod_i stops its foot toe movement and avoids physical interference with the adjacent module's foot toe.

3.3 Dynamic simulation

Using the proposed gait generation system, it is guaranteed mathematically that the cluster's gait pattern converges to static walking with a feasible foot toe speed for various leg configurations. In this section, we verify that several clusters achieve static walking for the

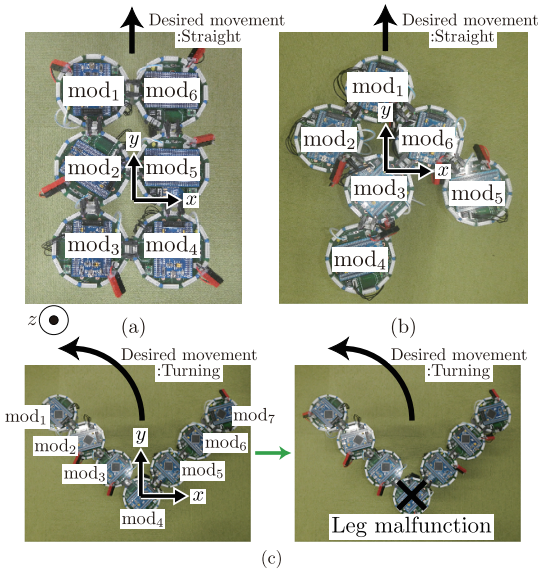


Fig. 3.13 Clusters' leg configurations used in the dynamic simulation. (a) Six-legged hexapod configuration. Its desired movement was straight movement. Vector $[V_x \;\; V_y]^{\mathrm{T}}$ is the cluster's velocity with respect to cluster coordinate $[x \;\; y]^{\mathrm{T}}$ in this figure. (b) Six-legged asymmetric configuration. Its desired movement was straight movement. (c) Seven-legged V-shaped configuration. During walking, mod_4's motor broke and the foot toe was fixed in the air. Its desired movement was turning movement.

desired movement using a dynamic simulation. For the simulation, we used the dynamic simulator lpzrobots (http://robot.informatik.uni-leipzig.de/software/). In the simulation, because we focused only on the dynamic behavior of the cluster, we assumed that the $[{}^i\boldsymbol{g}]$ calculation stage and $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage had already converged and simulated their walking behavior while the $[\omega_i, v_i]$ calculation stage and $[\theta_i[t], \beta_i]$ calculation stage were converging. In the robot experiment in Section 3.4, all stages including the $[{}^i\boldsymbol{g}]$ calculation stage and $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage were performed.

3.3.1 Simulation setting

We prepared three types of cluster: a six-legged hexapod cluster, as shown in Fig. 3.13(a), a six-legged asymmetric cluster, as shown in Fig. 3.13(b), and a seven-legged V-shaped cluster, as shown in Fig. 3.13(c). For the V-shaped cluster, after 10 seconds, we considered the leg malfunction of mod_4, that is, all three motors' angles of mod_4 were fixed and the foot toe was fixed in the air. The desired movement of the hexapod cluster and the asymmetric cluster was $[V_x \;\; V_y \;\; \Omega]^{\mathrm{T}} = [0[\mathrm{m/s}] \;\; 0.03[\mathrm{m/s}] \;\; 0[\mathrm{rad/s}]]^{\mathrm{T}}$ and simulated straight movement, where V_x and V_y were fixed to the cluster with respect to the x and y-axes, respectively, as shown in Fig. 3.13. Additionally, the desired movement of the V-shaped cluster was $[V_x \;\; V_y \;\; \Omega]^{\mathrm{T}} = [0[\mathrm{m/s}] \;\; 0.025[\mathrm{m/s}] \;\; 0.1[\mathrm{rad/s}]]^{\mathrm{T}}$ and simulated turning movement.

Table 3.3 Parameters used in the simulation.

Parameter	Value	Unit
h	180	mm
d	50	mm
α_2	0.1	-
dt	0.05	sec

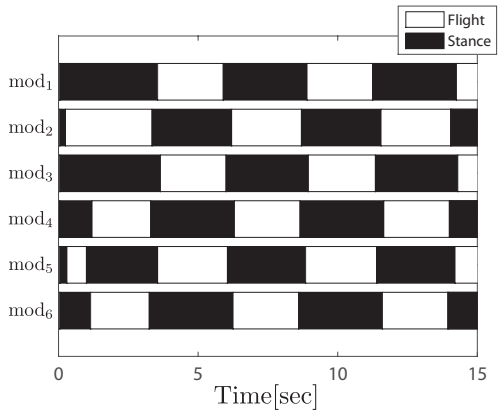


Fig. 3.14 Six-legged hexapod cluster's footprint diagram.

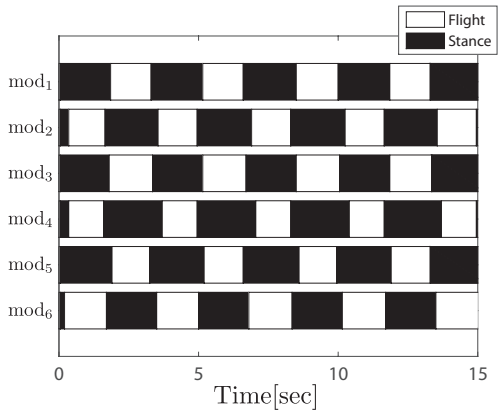


Fig. 3.15 Six-legged asymmetric cluster's footprint diagram.

The common parameters for the simulations are presented in Table 3.3, where we assumed that each leg's motors had sufficient power to achieve the desired speed of each module, that is, $r_{\text{opt}} = 1$.

3.3.2 Simulation results

Straight movement for the six-legged hexapod cluster

The footprint diagram for the six-legged hexapod cluster is shown in Fig. 3.14 (Movie: see Supplementary Materials 1). From Fig. 3.14, we can observe that the cluster's gait pattern converged within 2 seconds, or 40 iterations (because $dt = 0.05$), and achieved static walking. Additionally, because we can observe that ${}^{i}\phi_{R(i)} \forall i$ converged to π (see Appendix E for the mathematical proof), the gait pattern is similar to the tripod gait pattern, as seen in the insect gait pattern.

Straight movement for the six-legged asymmetric cluster

The footprint diagram for the six-legged asymmetric cluster is shown in Fig. 3.15 (Movie: see Supplementary Materials 2). From Fig. 3.15 and Appendix E, we can observe that ${}^{i}\phi_{R(i)} \forall i$ converged to π based on the gradient descent method Eq.(3.31) within 2 seconds, or 40 iterations. Additionally, we can observe that one walking cycle of the six-legged asymmetric cluster

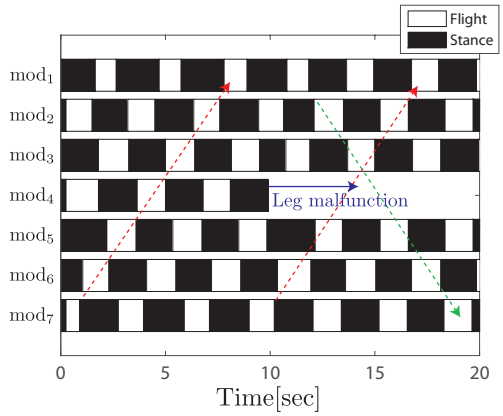


Fig. 3.16 Seven-legged V-shaped cluster's footprint diagram. The dotted arrows indicate that the flight phases propagated one after another.

was faster than that of the hexapod cluster. This is because $\max_i \left(\frac{v_i^{\text{ideal}}}{s_i^{\min}} \right)$ for the six-legged asymmetric cluster case $\left(\approx \frac{0.03}{0.075} \text{ for } i = 3 \right)$ was larger than that for the hexapod cluster case $\left(= \frac{0.03}{0.12} \forall i \right)$. Therefore, the phase velocity determined by Eq.(3.26) for the six-legged asymmetric cluster case was larger than that for the hexapod cluster case.

Turning movement for the seven-legged V-shaped cluster

The footprint diagram for the seven-legged V-shaped cluster is shown in Fig. 3.16 (Movie: see Supplementary Materials 3). From Fig. 3.16 and Appendix E, we can observe that ${}^i\phi_{R(i)} \forall i$ converged to $8\pi/7$ based on the gradient descent method Eq.(3.31) within 2 seconds, or 40 iterations. Thus, the flight phase propagated as indicated by the red dotted arrows in Fig. 3.16 as a feature of the converged gait pattern. Additionally, we can observe that one walking cycle of the seven-legged V-shaped cluster was faster than that of the hexapod cluster. This is because $\max_i \left(\frac{v_i^{\text{ideal}}}{s_i^{\min}} \right)$ for the seven-legged V-shaped cluster case $\left(\approx \frac{0.052}{0.12} \text{ for } i = 7 \right)$ was larger than that for the hexapod cluster case $\left(= \frac{0.03}{0.12} \forall i \right)$, where v_i^{ideal} in Eq.(3.12) $\forall i$ are not identical for the case in which the cluster's desired movement is turning. Therefore, the phase velocity determined by Eq.(3.26) for the seven-legged V-shaped cluster case was larger than that for the hexapod cluster case. As shown in Fig. 3.16, after mod_4's leg was broken, we can observe that the cluster adapted to the leg malfunction by changing the phase difference within 1 second, or 20 iterations. Then ${}^i\phi_{R(i)} \forall i$ converged to π, and the flight phases propagated as indicated by the red dotted arrow and the green dotted arrow in Fig. 3.16.

In the following, we explain the details of the adaptation. Immediately after mod_4's leg was broken, mod_3 changed $\text{mod}_{R(3)}$ from mod_4 to mod_5, and mod_5 changed $\text{mod}_{L(5)}$ from mod_4 to mod_3 based on Algorithm 1, hence a new type of G was constructed. Before the

malfunction, mod_3 and mod_5 were almost in phase, but after the malfunction, mod_3 and mod_5 were adjusted to anti-phase based on Eq.(3.31).

3.3.3 Walking of several other types of clusters

It is difficult to describe the walking of several other types of clusters in this study; thus, we summarized other clusters' walking in videos (see Supplementary Materials 4 and Supplementary Materials 5). In detail, using a dynamic simulation, we validated that seven types of clusters (5–11 modules) successfully achieved two types of straight and one type of turning walking (Movie: see Supplementary Materials 4), where the target speeds of straight and turning walking were the same values to be used in Section 3.3.1. Furthermore, we validated that an eight-legged cluster achieved static walking even after the malfunction of one leg, and even after the broken leg worked again (Movie: see Supplementary Materials 5), where different gait patterns were observed depending on the timing when the broken leg worked again. Finally, we validated that a 12-legged cluster achieved static walking even after the malfunction of six legs (Movie: see Supplementary Materials 5).

3.4 Robot experiment

In this section, we investigate the effectiveness of our proposed system using real moduled robots. In the dynamic simulation, because we focused on the dynamic behavior of the cluster, the $[{}^i\boldsymbol{g}]$ calculation and $[\text{mod}_{R(i)}, \text{mod}_{L(i)}]$ calculation stages were assumed to be complete. However, in the robot experiment, all stages in Fig. 3.4 were performed by all legged modules.

3.4.1 Experimental setting

The cluster's leg configuration and desired movement were the same as those of the dynamic simulation; that is, we investigated the straight movement of the six-legged hexapod cluster in Fig. 3.13(a), the six-legged asymmetric cluster in Fig. 3.13(b), and turning movement of the seven-legged V-shaped cluster in Fig. 3.13(c). For the V-shaped cluster, we cut the signal lines for the three motors of mod_4 during walking and mod_4's foot toe was fixed in the air. The cluster's actual movement V_x, V_y, and Ω were recorded by a motion capture system and plotted, and V_x, V_y, and Ω in the simulation were also plotted for comparison. Parameters h, d, and α_2 were the same as the dynamic simulation settings, as shown in Table 3.3, and $v^{\text{st}}_{\text{max}}, v^{\text{fl}}_{\text{max}}$ were given by high values that were sufficient to achieve the desired speed of each module. Hence, $r_{\text{opt}} = 1$.

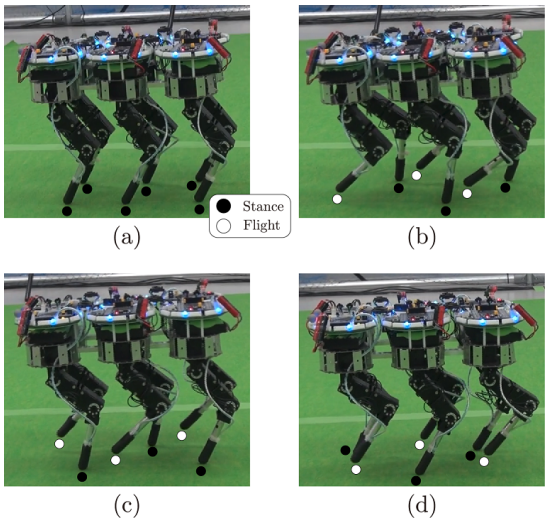


Fig. 3.17 Six-legged hexapod cluster's walking behavior. (a) t = 0[sec]. (b) t = 9[sec]. (c) t = 12[sec]. (d) t = 15[sec].

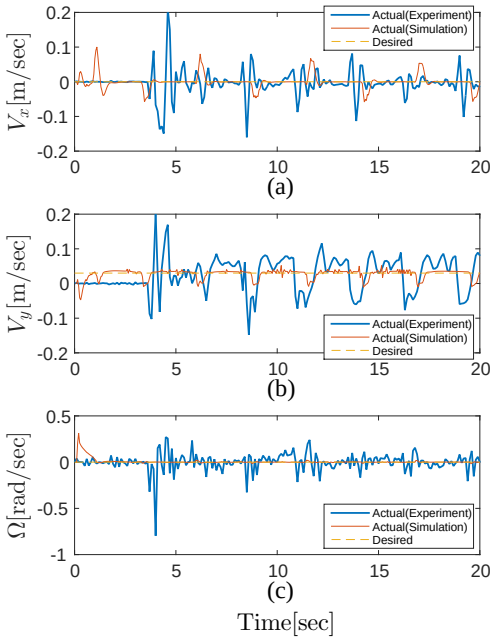


Fig. 3.18 Six-legged hexapod cluster's actual (experiment, simulation) and desired (a) speed toward the x-axis, (b) speed toward the y-axis, and (c) angular velocity. The speed in the dynamic simulation was plotted for the walking for which the footprint diagram Fig. 3.14 was obtained.

3.4.2 Experimental results

Straight movement for the six-legged hexapod cluster

We show the six-legged hexapod cluster's walking behavior in Fig. 3.17 (Movie: see Supplementary Materials 6). Figs. 3.17(b), (c), and (d) show that $(\mathrm{mod}_1, \mathrm{mod}_3, \mathrm{mod}_5)$ and $(\mathrm{mod}_2, \mathrm{mod}_4, \mathrm{mod}_6)$ were alternately in the stance phase. As Fig. 3.17, Supplementary Materials 6, and Appendix E show, the cluster's gait pattern converged to a tripod gait pattern such that ${}^i\phi_{R(i)} \forall i$ converged to π. Until approximately 4 seconds after control started, each module completed the $[{}^i\boldsymbol{g}]$ calculation stage and $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage. Then, each module started walking and the cluster's gait pattern converged approximately 5 seconds after all the modules started walking (analyzed by video). In a real robot experiment, the information transmitting timing of all modules is not synchronized and a communication error sometimes occurs; thus, after walking starts, the convergence duration of the gait pattern takes more time than that of a simulation.

From Fig. 3.18, we observe that the six-legged hexapod cluster's actual translational velocity

toward the x-axis and y-axis, and the angular velocity oscillated around the desired velocity. Furthermore, the cluster's moving speed toward the moving direction, that is, the y-axis, decreased per stance-flight transition time. This speed deterioration based on the stance-flight transition is observed in natural creatures' locomotion and explained from the physical viewpoint [115].

Straight movement for the six-legged asymmetric cluster

We show the six-legged asymmetric cluster's walking behavior in Fig. 3.19 (Movie: see Supplementary Materials 7). Figs. 3.19(b), (c), and (d) show that $(\mathrm{mod}_1, \mathrm{mod}_3, \mathrm{mod}_5)$ and $(\mathrm{mod}_2, \mathrm{mod}_4, \mathrm{mod}_6)$ were alternately in the stance phase. As Fig. 3.19, Supplementary Materials 7, and Appendix E show, the cluster's gait pattern converged to a gait pattern such that ${}^i\phi_{R(i)} \forall i$ converged to π. Until approximately 5 seconds after control started, each module completed the $[{}^i\boldsymbol{g}]$ calculation stage and $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage. Then, each module started walking, and the cluster's gait pattern converged approximately 5 seconds after all the modules started walking. Convergence duration of the gait pattern takes more time than that of a simulation for the same reason as the hexapod cluster case.

From Fig. 3.20, we observe that the six-legged asymmetric cluster's actual translational velocity toward the x-axis and y-axis, and the angular velocity oscillated around the desired velocity. Furthermore, the cluster's moving speed toward the y-axis decreased per stance-flight transition time for the same reason as that for the hexapod cluster case.

Turning movement of the seven-legged V-shaped cluster

We show the seven-legged V-shaped cluster's walking behavior in Fig. 3.21 (Movie: see Supplementary Materials 8). Until approximately 10 seconds after control started, each module completed the $[{}^i\boldsymbol{g}]$ calculation stage and $[\mathrm{mod}_{R(i)}, \mathrm{mod}_{L(i)}]$ calculation stage. Then, each module started walking and the cluster's gait pattern converged approximately 5 seconds after all the modules started walking. Convergence duration of the gait pattern takes more time than that of a simulation for the same reason as the hexapod cluster case.

In Fig. 3.21(b), $(\mathrm{mod}_1, \mathrm{mod}_3, \mathrm{mod}_5, \mathrm{mod}_6)$ are in the stance phase, and in Fig. 3.21(c), $(\mathrm{mod}_1, \mathrm{mod}_3, \mathrm{mod}_5, \mathrm{mod}_7)$ are in the stance phase, therefore COM-C is inside the support polygon at the two times. Regarding the other times, COM-C is always inside the support polygon (see Supplementary Materials 8). Thus, the cluster achieved static walking.

As Fig. 3.21, Supplementary Materials 8, and Appendix E show, the cluster's gait pattern converged to a static walking pattern such that ${}^i\phi_{R(i)} \forall i$ converged to $6\pi/7$. Note that the initial phase of each module $\theta_i[0]$ was set to a random value and the initial phase value set determines the terminal phase difference; thus, the terminal phase difference in the experiment was different from that in the simulation case, in which ${}^i\phi_{R(i)} = 8\pi/7$. After mod_4's signal lines were cut at $t = 28$[sec] in Fig. 3.21(d), the cluster's gait pattern shifted to another pattern. Figs. 3.21(e) and (f) show that $(\mathrm{mod}_2, \mathrm{mod}_5, \mathrm{mod}_7)$ and $(\mathrm{mod}_1, \mathrm{mod}_3, \mathrm{mod}_6)$ were

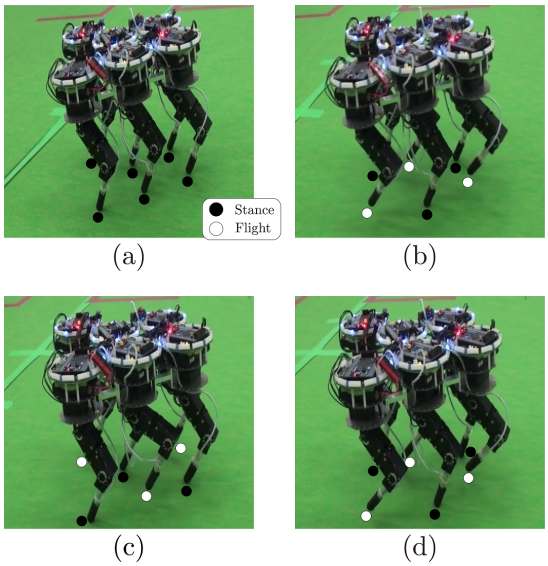


Fig. 3.19 Six-legged asymmetric cluster's walking behavior. (a) t = 0[sec]. (b) t = 10[sec]. (c) t = 12[sec]. (d) t = 13[sec].

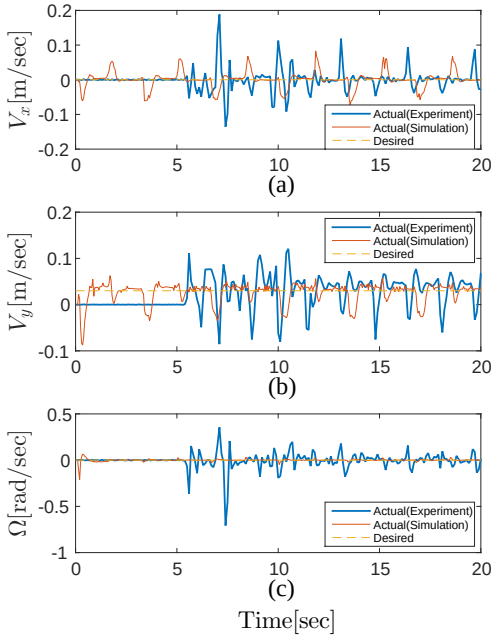


Fig. 3.20 Six-legged asymmetric cluster's actual (experiment, simulation) and desired (a) speed toward the x-axis, (b) speed toward the y-axis, and (c) angular velocity. The speed in the dynamic simulation was plotted for the walking for which the footprint diagram Fig. 3.15 was obtained.

alternately in the stance phase. Thus, after a leg was broken, the cluster achieved static walking again.

From Fig. 3.22, we observe that the cluster's actual translational velocity toward the x-axis and y-axis, and the angular velocity oscillated around the desired velocity, where the seven-legged V-shaped cluster's actual movement after mod_4's leg malfunction was not substantially different from that before mod_4's leg malfunction. Furthermore, the cluster's moving speed toward the y-axis decreased per stance-flight transition time for the same reason as that for the hexapod cluster case.

3.5 Discussion and conclusion

In this study, we developed an autonomous distributed system for single-legged modular robot "KARAKASA" to achieve the static walking of arbitrary clusters for arbitrary movement with a feasible speed. All modules can calculate the target variables of all the stages in Fig. 3.4 without falling into inter-module deadlock, and therefore it is guaranteed that arbitrary types of clusters achieve the desired movement. Using the proposed system, we can obtain several

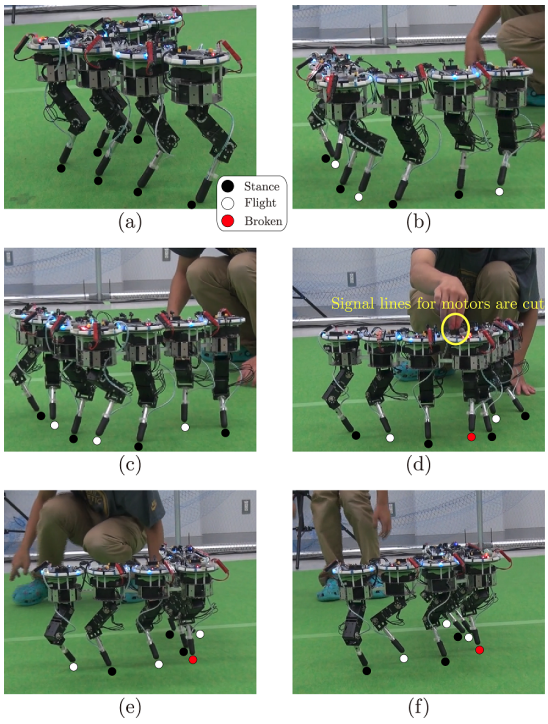


Fig. 3.21 Seven-legged V-shaped cluster's walking. (a) t = 0[sec]. (b) t = 19[sec]. (c) t = 22[sec]. (d) t = 28[sec]. Signal lines for the motors were cut. (e) t = 33[sec]. (f) t = 35[sec].

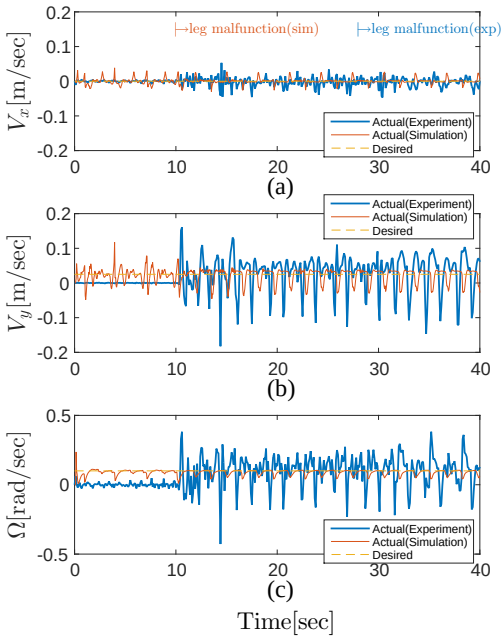


Fig. 3.22 Seven-legged V-shaped cluster's actual (experiment, simulation) and desired (a) speed toward the x-axis, (b) speed toward the y-axis, and (c) angular velocity. The speed in the dynamic simulation was plotted for the walking for which the footprint diagram Fig. 3.16 was obtained.

benefits of a multi-legged robot with various leg configurations.

The major difference between the proposed system and existing gait generation systems under a distributed control framework [5–8, 94, 95, 104, 105] is that existing systems provide the interlimb network as the initial state based on the hexapod or myriapod configuration. However, for the case of an arbitrary configuration, the interlimb network cannot be provided initially. In the proposed system, each module locally constructs interlimb network G, which contributes to the static walking of the cluster with an arbitrary configuration. Furthermore, most existing systems with a complicated neural network or interaction with the ground did not guarantee the convergence of the gait pattern mathematically because of the complexity of the neural network or difficulty in formulating the interaction with the ground. In the proposed system, we simplified the problem in such a manner that interaction with the ground was not considered, and therefore we were able to guarantee the convergence of the gait pattern from the mathematical viewpoint.

Because the system is an autonomous distributed system, it potentially has the benefits of scalability and fault tolerance. In the following subsections, we verify that those benefits are

embedded in the proposed system.

3.5.1 Scalability

When hop number $|i, R(i)|$ or $|i, L(i)|$ is large, multi-hop information delivery is required for information exchange, and the number of communication exchanges per time is large; therefore, the hop number should be small. In our proposed system, $|i, j|(\forall j \in \{R(i), L(i)\})$ depend on the leg configuration, that is, the number of modules and their positions. If the communication cost dependent on $|i, j|(\forall j \in \{R(i), L(i)\})$ increases directly as the number of modules increases, then the system has low scalability. By contrast, if the communication cost is constant regardless of the number of modules, then the system has high scalability. In this subsection, we discuss the relationship between the communication cost and number of modules.

We summarized the relationship between a system's applicable leg configurations and communication cost for both existing gait generation systems without interaction with the ground [5, 6] and our proposed system (Fig. 3.23). Fig. 3.23(a) shows the relationship for the existing gait generation systems, and Fig. 3.23(b) shows that for our proposed system. Regarding leg configurations in the yellow sets, because $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ are physical neighbors to $\mathrm{mod}_i \forall i$, mod_i uses only information about the neighbors; hence, the communication cost is small. From Fig. 3.23, we can observe that our proposed system has a larger yellow set than the existing systems. Regarding other leg configurations in the orange set, because $\mathrm{mod}_{R(i)}$ or $\mathrm{mod}_{L(i)}$ is not a physical neighbor of mod_i for any i, the communication cost is larger than that for the yellow set, but only our proposed system always achieves the control purpose. Thus, from the viewpoint of communication cost and the applicable leg configurations, the proposed system is superior to existing systems.

In the following, we summarize robot hardware for which the proposed system can be applied. According to the leg motors' DOF configuration, legged robots can be classified into two groups: a mammal type, where the leg exists under the body like animals [6], and a sprawling type, where the leg exists on the side of the body like insects [5, 97, 99, 101, 102]. In this study, we considered the mammal-type robot KARAKASA, whose COM-M and center position of the stride exist at the same position from the top viewpoint. Meanwhile, for a sprawling-type robot, the COM-M position and center position of the stride exist at different positions from the top viewpoint. The proposed system can also be applied to a sprawling-type robot by letting ${}^{i}\boldsymbol{g}_{\mathrm{est}}[0]$ be the COM-M position of each mod_i and o_i be the center position of the stride of each mod_i. However, our proposed system cannot be applied to a legged robot whose body is flexible, as in [88], because COM-C does not exist at a constant position.

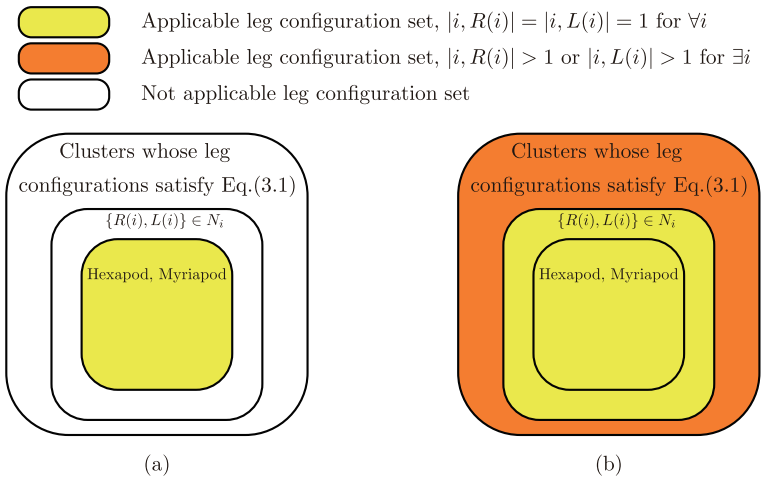


Fig. 3.23 Relationship between the system's applicable leg configurations and communication cost. (a) Existing gait generation systems. (b) Proposed gait generation system.

3.5.2 Fault tolerance

As mentioned in Section 3.1, if the condition (3.2) holds, then the cluster achieves static walking even after leg malfunction. In our proposed system, we only guarantee that the cluster's gait pattern converges to the static walking pattern; thus, stability (cluster does not fall down) during the transition from one gait pattern before leg malfunction to another pattern after leg malfunction is not guaranteed.

As an extension to fault tolerance, the case in which some legs are used for manipulation (i.e., grasping) during walking can be regarded as being similar to leg malfunction; therefore, the cluster achieves static walking after some legs are used for manipulation.

Furthermore, our system can potentially be applied not only to leg malfunction but also other types of malfunction, for example, CPU malfunction in which a broken module cannot communicate with others at all or actuate the motor. In the normal condition, each mod_i always communicates with mod_j ($\forall j \in N_i \cup \{R(i), L(i)\}$), so mod_j can detect mod_i's CPU malfunction if mod_i does not transmit its information within a certain time period. For the case of one mod_i's CPU malfunction, the new assumption that the "communication network of all the remaining modules except mod_i are connected" is required in addition to the assumption about the leg configuration, that is, the condition (3.2). Under the assumptions about the network and the condition (3.2), any one module can transmit its information to any other module and a new graph G can be constructed using Algorithm 1; therefore, the cluster's gait pattern after CPU malfunction is guaranteed to converge to the static walking pattern. In a centralized system, when the central control unit is broken, the entire system stops, but in our distributed system, CPU malfunction does not lead to complete system outage.

3.5.3 Difference between biological and proposed gait patterns

In the following, we explain the difference between gait pattern observed in natural creatures and that of the proposed system. As insects increase their walking speed, they decrease their stance rate and change their gait pattern from the wave gait to the tripod gait [30]; such a gait transition is energy efficient [116]. Whereas, regarding the gait pattern of the proposed system, the stance rate is constant (see Eq.(3.21)) regardless of the walking speed; hence, gait transition based on the walking speed is not observed in the proposed system. Additionally, the tripod gait pattern is observed (see Fig. 3.14), but the wave gait pattern is not observed in the six-legged hexapod cluster.

The gait pattern observed in a centipede is the retrograde wave gait (locomotory waves travel backward), and that in a millipede is the direct wave gait (locomotory waves travel forward), where simultaneous flight phases for the front-back adjacent legs have been observed [117]. Whereas, regarding the gait pattern of the proposed system, when mod_i is in the flight phase, both $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ are always in the stance phase. Therefore, simultaneous flight phases for the front-back adjacent legs are not observed in our proposed system.

The proposed system focuses only on guaranteeing that the arbitrary cluster's gait pattern converges to a static walking pattern; thus, as these examples show, some of the characteristics of animal-like locomotion are not observed.

3.5.4 Future extensions

In this study, because we considered walking only on planar ground, in future work, we will include the ground-interaction term in the dynamics of oscillators (see Eq.(3.31)) like [7, 8, 94, 95] and construct an autonomous distributed system that generates more adaptive locomotion against uneven terrain. Furthermore, in this study, we focused only on static walking and did not consider the stability margin or energy efficiency; thus, we will consider those important criteria for walking.

Supplementary materials

Please see the corresponding journal page at https://doi.org/10.1109/TRO.2020.2992983.

Supplementary Materials 1: Movie of six-legged hexapod cluster's walking in the simulation.

Supplementary Materials 2: Movie of six-legged asymmetric cluster's walking in the simulation.

Supplementary Materials 3: Movie of seven-legged V-shaped cluster's walking in the simulation.

Supplementary Materials 4: Movie of 5-11 legged clusters' several walkings in the simulation.

Supplementary Materials 5: Movie of leg malfunction and recovery for eight-legged cluster and malfunction of six legs for 12-legged cluster in the simulation.

Supplementary Materials 6: Movie of six-legged hexapod cluster's walking in the experiment.

Supplementary Materials 7: Movie of six-legged asymmetric cluster's walking in the experiment.

Supplementary Materials 8: Movie of seven-legged V-shaped cluster's walking in the experiment.

Chapter 4

Adaptation of a multi-legged robot to outer environment

For some social insects, it has been observed that individuals assemble with one another to achieve tasks that cannot be achieved by themselves. For example, ants link their bodies together to build bridges for gap traversal [21–23], to exert large forces for prey transport [22, 45], and to build rafts for water traversal [22, 118]. Surprisingly, ants do not have any leader to manage the global morphology, but achieve these tasks by assembling into an appropriate morphology through local information exchange.

Achieving a task by self-assembly is also beneficial for artificial reconfigurable modular robot systems, which are composed of repeated robot elements with a mutual physical connection mechanism [24–28]. It is difficult for a robot with a constant morphology to achieve several tasks, but the reconfigurable modular robots can reconfigure the morphology in response to the task. However, to achieve an appropriate reconfiguration in response to the given task is a challenge [119]. In the following, we overview the related works for achieving the appropriate reconfiguration in response to the given tasks.

Kamimura et al. [120] designed a library between tasks and configurations (denoted as task-configuration library) for M-TRAN modules to traverse slopes with several inclinations. When a cluster (an entire robot composed of modules) of M-TRANs with a given configuration climbs a slope and one of the modules' posture inclines more than a threshold, they reconfigure into another configuration that is determined a priori; thus, the cluster can keep climbing the slope. However, global information about the initial configuration is given to all the modules. Li et al. [121] designed a task-configuration library for Sambots to traverse walls of different height. When a module is blocked by a wall, the module denoted as seed module measures the wall's height and chooses the corresponding configuration from the library. Then, the seed module controls the reconfiguring procedures of other modules in the cluster to achieve the desired configuration by obtaining global information about the current configuration of the cluster. After the reconfiguration is complete, the cluster starts to traverse the wall. Jing et al. [122] distinguished specification of each configuration of SMORES-EPs

and designed a task-configuration library to achieve various tasks, such as object transport and climbing stairs. In the robot experiments, a centralized high-level mission planner uses feedback from a camera to choose a corresponding configuration from the library in response to the environment [123]. The mission planner estimates the location of the SMORES-EPs using the camera and controls where the modules should connect. Khodr et al. [124] created a planning framework for Roombots to reconfigure from a given configuration into a desired one. Using a reconfiguring procedure which is calculated a priori based on the framework, three Roombot modules cooperatively traversed a gap. However, global information such as the initial configuration, desired configuration and the environmental information are required to calculate the reconfiguring procedure.

All above strategies [120,121,123,124] utilize a task-configuration library, which means that their desired reconfiguration procedures are determined a priori. Utilizing a task-configuration library has disadvantages, such as the following. First, it takes considerable efforts to construct the library itself; for example, a specification of each configuration needs to be provided. Second, assuming all of the tasks in highly complex and nonlinear nature is not possible; thus, robots can respond to only a limited number of tasks within the library [125]. Finally, after choosing a desired configuration from the library, global information, such as an initial configuration of the cluster [120,124] or current configuration of the cluster in the all reconfiguration procedure [121,123], is required to reconfigure into the desired morphology.

Meanwhile, O'Grady et al. [11] proposed a reconfiguring strategy for each s-bot (a wheeled mobile robot) to traverse an unknown slope and gap environments without a task-configuration library. When one of the modules detects that the environment cannot be traversed by itself, they return to the starting area and assemble into a cluster called swarm-bot. After the assembly is complete, the cluster tries to traverse the environment again. As a limitation of the strategy, after assembling into a cluster, each module no longer monitors whether the environment can be traversed. Therefore, the cluster was sometimes in short of modules to traverse the environment and fell down [11]. Another disadvantage is that either all modules are recruited or none. In other words, the strategy is not able to tune the number of modules involved in the cluster according to the faced environment. We denote the strategy as "heuristic".

In this study, we consider two tasks: unknown slope and gap traversal and a single-legged reconfigurable modular robot system as explained in Chapter 3. There are a cluster composed of a certain number of single-legged modules and a certain number of unconnected free modules at an unbounded flat environment. The cluster is allowed to recruit the free modules as additional new components of the cluster by indicating the connecting place in the cluster. Moreover, there is a goal that is located beyond the unknown slope and gap environments to which the cluster walks. O'Grady et al. [11] studied detection of success or failure of environment traversal for the individual module of a wheeled mobile robot. However, it is not applicable to any cluster constructed by any number of the wheeled mobile robots and

Table 4.1 Summary of the relative position of this study among reconfiguring strategies to traverse uneven terrains.

Research	Robot hardware	Reconfiguring strategy		Environment
		2D	3D	
[11]	S-bot	Heuristic		Slope and Gap
[120]	M-TRAN Ⅱ		Library	Slope
[121]	Sambot		Library	Wall
[123]	SMORES-EP		Library	Stairs
[124]	Roombots		Library	Gap
This study	KARAKASA	Adaptive		Slope and Gap

any connection patterns. We propose a control system for each KARAKASA (a single-legged robot) module to detect success or failure of environment traversal for the cluster. Owing to the success or failure detection, the cluster obtains a choice between departing to the goal and returning to the starting area; thus, the cluster can reconfigure its morphology adaptively in response to the difficulty of the environment. A legged robot has characteristics such that the support polygon is composed of all grounded foot toe points; thus, the stability margin of the cluster on slope and gap environments can be calculated more easily than for wheel-type robot clusters such as swarm-bots. This facilitates the cluster's success or failure detection. We summarize the relative position of this study among self-assembly strategies in Table 4.1.

4.1 KARAKASA robot system

In this section, we describe the KARAKASA robot hardware, its notations, and our previous gait pattern generation system for the cluster. The hardware has been improved from that in Chapter 3.

4.1.1 Hardware and notations

Each module is composed of a disc-shaped body whose radius is r (=60mm) including a CPU and a battery, and a three-DOF leg which can move in three-dimensional space. The CPU processes the sensor information, communicates with other modules, and actuates the motors for the leg. Total weight of the module is approximately 560g without the connecting unit. The size and weight of each segment in the module are described in Fig. 4.1(a) and (b).

To measure an unknown environment, each module is equipped with three types of sensors (Fig. 4.1(a) and (b)): a touch sensor at the foot toe to measure the ground existence, an IMU to measure the posture of the cluster, and ambient IR sensors to measure the position of a light source as a goal.

Each module has connecting places every 30 degrees, and can physically connect with a

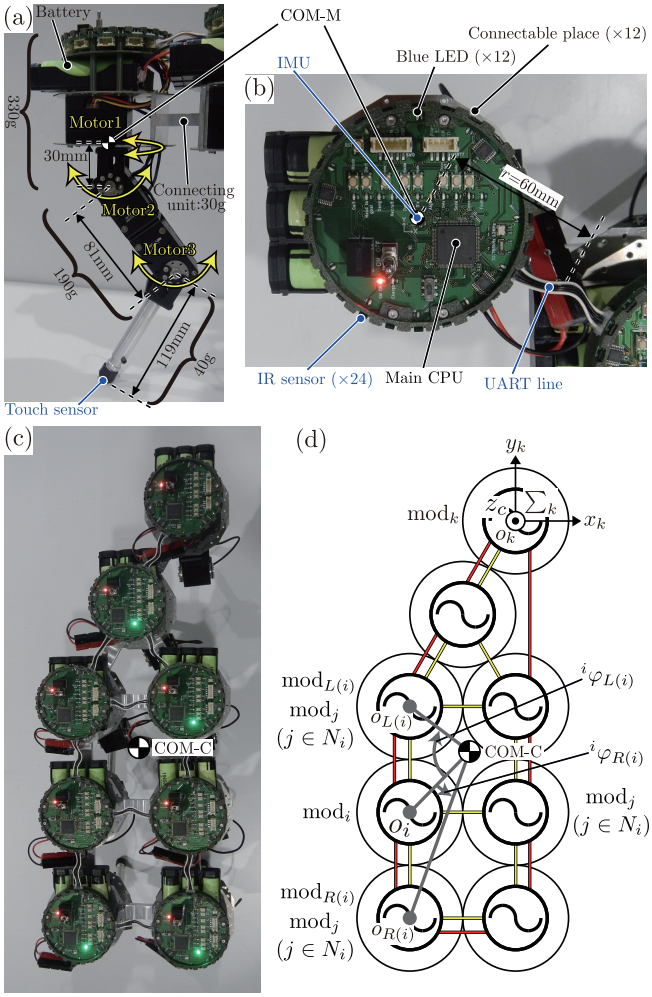


Fig. 4.1 Single-legged modular robot "KARAKASA". Components to obtain outer environmental information and other modules' information are represented by blue color. (a) Single module viewed from the side. (b) Single module viewed from the top. (c) An example cluster's configuration from the top. COM-C means the center of mass of the cluster. (d) Model of single-legged modular robots. A mark in each circle body represents a central pattern generator. The set of yellow lines is a graph such that all modules are adjacent to physically adjacent modules and the set of red lines is a graph G explained in Section 4.2.2.

maximum of six other modules using a connecting unit. Each module can communicate via UART with the physically adjacent modules. Autonomous inter-module assembling capacity has not been implemented in the current hardware, so inter-module assembling procedure (physically and electrically) is performed manually. Modules can be connected in such a way that the bodies form a planar shape composed of discs tangent to each other, for an example, see Fig. 4.1(c); thus, the cluster's leg configuration is not limited to those of natural creatures (e.g. hexapod or myriapod). Moreover, each module is equipped with LEDs that are located

above the connecting places. Modules indicate the connecting place of free modules by turning on the LED.

Each module, denoted by mod_i, has its own local coordinate system $\sum_i$, whose origin o_i is fixed at the the center of mass of the module (COM-M). The x_i, y_i, and z_c-axes are also fixed on the body, where z_c-axis is identical to all modules (Fig. 4.1(d)). In the control system, we assume that even when the leg moves, COM-M for mod_i remains at a constant position, that is, the center of the disc-shaped body o_i. The validity of this assumption is provided in Appendix B. Let ${}^i\varphi_j \in (-\pi\ \pi]$ be an angle composed of o_i, the center of mass of the cluster (COM-C), and o_j. Let N_i be an index set of modules that mod_i is physically adjacent to. Let $R(i)$ and $L(i)$ be indicies of modules that mod_i is adjacent to at graph G as ${}^i\varphi_{R(i)} \geq 0$ and ${}^i\varphi_{L(i)} \leq 0$, respectively, which is explained in Section 4.2.2.

4.2 Problem setting

In this study, we consider slope and gap environments. As the cluster achieves its static walking on such environments, the previous gait generation system should be modified. The modification points are as follows: First, the foot toe trajectory is modified to a new one in Section 4.3.2. Second, to retain a certain stability margin for the walking of the cluster, the definition of G is redefined in Section 4.2.2, where G_V is modified to a time-dependent set of vertices because a module whose foot toe in the stance phase falls into the gap is excluded from G. Description of the main symbols used in this study are provided in Table 4.2.

4.2.1 Environmental setting

We focus on two tasks: slope and gap traversal. There are a cluster composed of a certain number of modules and a certain number of free modules at an unbounded flat environment, denoted as starting area. The cluster's initial configuration satisfies the condition explained in Section 4.2.3. The cluster is allowed to recruit the free modules as additional new components of the cluster. In the proposed system, the cluster recruits one free module by indicating the connecting place in the cluster per recruitment. The number of recruitment is adjusted in response to the difficulty of the environment as explained in Section 4.3.1. Moreover, there is a goal beyond unknown slope or gap environment, and the purpose of the cluster is to reach the goal.

As for the slope, we consider a sequence of M gradually inclining slopes (Fig. 4.2(a)). The angle between adjacent slopes is λ, the angle from the cluster's moving direction to the side of the slope is κ, and the width of each slope is L. In the slope environment, the goal is located at the edge of the slope M. While the cluster moves to the goal, if one of the modules in the cluster reaches the goal side edge of the slope, we regard that the cluster reaches the goal. Let W be a width of gap as shown in Fig. 4.2(b). In the gap environment, the goal is located at

Table 4.2 Description of main symbols used in this study.

Symbol	Description
r	Radius of each module's disc body
$\sum_i$	$\equiv (o_i - x_i\ y_i\ z_c)$. Local coordinate system of mod_i
N_i	Index set of the physically adjacent modules of mod_i
${}^i\varphi_j$	Angle composed of o_i, COM-C and o_j
$\theta_i[t]$	Phase of mod_i at time t
ω_i	Phase velocity of mod_i
h	A constant length between o_i for mod_i and ground at a flat environment
d	A constant length between highest foot toe position and ground at a flat environment
${}^i\boldsymbol{V}$	Cluster's desired translational velocity with respect to $\sum_i$
β_i	Stance rate of mod_i
${}^i\boldsymbol{g}$	Relative COM-C position with respect to $\sum_i$
$D_{i,j}$	Convex hull composed of AEP and PEP of mod_i and those of mod_j
$G_V[t]$	The time-dependent index set of modules that can support the cluster
$G[t]$	Interlimb network whose vertex set is $G_V[t]$
$G^*[t]$	A set of all possible $G[t]$
$R(i), L(i)$	Index that is the right and left side neighbor of mod_i at $G[t]$, respectively
σ_i	$\in \{Stance, Flight\}$. Phase state of $\mathrm{mod}_i (i \in G_V[t])$
$\gamma(i)$	$R(i)$ if $\sigma_{R(i)} = Stance$, otherwise $R(R(i))$
P	Polygon whose edges are composed of foot toe of $\mathrm{mod}_i (\forall i \in \{i \mid \sigma_i = Stance\})$ and $\mathrm{mod}_{\gamma(i)}$
E	A constant index set of modules that do not belong to $G_V[0]$
c_i	$\in \{Depart, Return_S, Return_G\}$. Variable of mod_i to describe the cluster's moving state
q_i	$\in \{Include, Exclude\}$. The state of mod_i whether the index i is included in $G_V[t]$
${}^i\boldsymbol{p}_j$	Relative position of o_j with respect to $\sum_i$
${}^i\boldsymbol{f}_j$	Relative foot toe position of mod_j with respect to $\sum_i$
d_s	A constant ground search depth under the ground at a flat environment
$s_i^{\mathrm{front}}, s_i^{\mathrm{rear}}$	Maximum moving range of foot toe of mod_i in the front and the rear side, respectively
${}^i\boldsymbol{r}_{\mathrm{goal}}$	Relative goal position with respect to $\sum_i$
$\alpha_{i,j}$	Stability margin angle for a line segment connecting foot toe of mod_i and mod_j
α_i	Stability margin angle monitored by mod_i
α_{close}	A constant stability margin for P to determine $\mathrm{mod}_i (\forall i \in E)$ that is close to COM-C
α_{str}	A constant stability margin for P to determine the stride range of each module
α_{min}	A constant minimum angle for the stability margin for falling down
η	A constant angle for recruitment in response to warning from G sustainable monitoring

the ground that is different from the ground in the starting area. While the cluster moves to the goal, if all of the modules in the cluster reach the goal side ground of the gap, we regard that the cluster reaches the goal. The environmental parameters λ, κ, L, M, and W are not given to any modules.

4.2.2 Distributed control framework

Our control framework is similar to that explained in Section 3.1.3, that is, each module obtains only local information through two graphs as explained in Section 3.1.3. Because this study focuses on the walking on slope and gap environments, the definition of graph G should

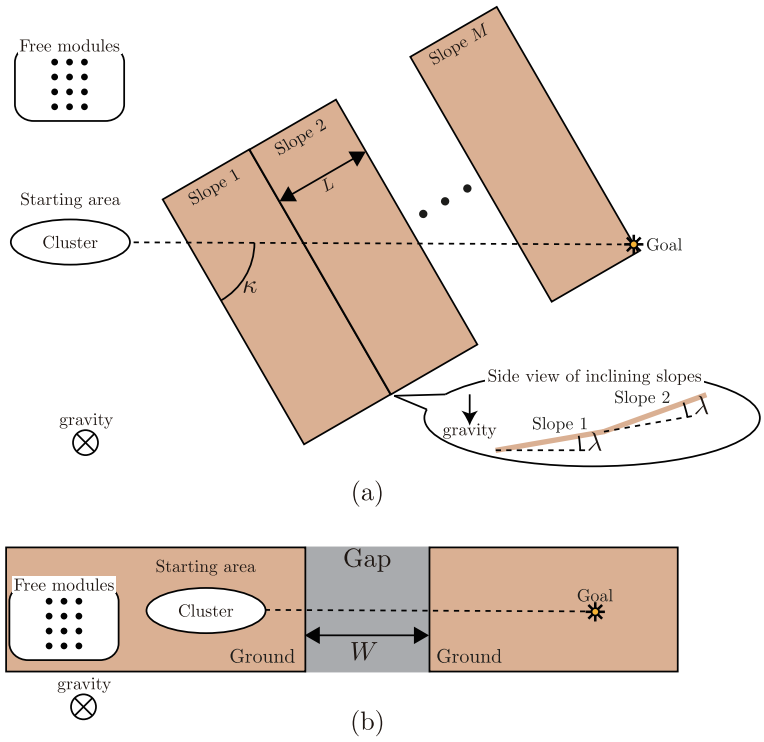


Fig. 4.2 Environments from top viewpoint. (a) Slope (b) Gap.

be modified.

Redefinition of graph G

As we consider walking on slope and gap environments, we redefine graph G in Chapter 3. First, the graph $G[t]$ and the index set of vertices $G_V[t]$ are time-dependent. This is because when a leg falls into the gap during the gap traversal, the leg cannot physically support the cluster. In the case, an index corresponding to the unsupportable leg is left from $G_V[t]$ (the detail is explained in Section 4.3.3); thus, $G[t]$ is time-dependent. Second, the stability margin for static walking of the cluster is newly considered to predict the success or failure of the slope traversal for the cluster; thus, the definition of $G_V[t]$ and a set of edges of $G[t]$ should be modified from that in Chapter 3.

Next, we present the new definition of $G_V[t]$ and a set of edges of $G[t]$. Two stability margins for the static walking of the cluster are introduced. First, when distance between COM-C and o_i for a mod_i is very small, the module's foot toe point is always very close to the COM-C. In that case, the distance between COM-C and the polygon P cannot be large (please see Section 3.2.2 for the definition of P). Therefore, the corresponding mod_i is excluded from $G_V[t]$ to retain a certain stability margin. In detail, let E be a constant index set of mod_i such that the distance between COM-C and o_i is smaller than $h \tan \alpha_{\mathrm{close}}$, where

h is a constant length between o_i for mod_i and ground at a flat environment and α_{close} is a constant margin to exclude modules that are close to COM-C. $G_V[0]$ is the set of indicies of all modules in the cluster except $\forall i \in E$. Second, let α_{str} be a constant stability margin to determine stride length of each module such that the cluster retains a certain stability margin during the walking. For a given $G_V[t]$, $G[t]$ is any cycle graph whose edge set satisfies the following. Each vertex $i \in G_V[t]$ is connected in an undirected way to vertices of two different modules (denoted by $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$) whose relative positions satisfy the condition (3.6) and the following condition (4.1):

$$\begin{array}{l}\text{A circle with the radius of } h\tan\alpha_{\mathrm{str}} \text{ around COM-C}\\ \text{does not cross with a line segment } \overline{o_{R(i)}o_{L(i)}} \text{ nor } \overline{o_{R(i)}o_i}.\end{array} \tag{4.1}$$

The graph $G[t]$ is not uniquely determined and the set of all possible $G[t]$ is denoted by $G^*[t]$. When the conditions (3.6) and (4.1) about the graph G and the condition (3.7) about the foot toe movement hold, COM-C is always interior of P with the stability margin $h\tan\alpha_{\mathrm{str}}$ under the assumption that foot toes of all $\mathrm{mod}_i(\forall i \in \{i|\sigma_i = \mathit{Stance}\})$ are located under o_i.

In Chapter 3, the stride range was determined without considering the stability margin to satisfy the condition (3.8). However, in this study, owing to the condition (4.1), each mod_i can determine the stride range considering the stability margin as explained later in Section 4.3.2; thus, the cluster can achieve static walking with the stability margin $h\tan\alpha_{\mathrm{str}}$.

Locally exchanged information

Let $c_i \in \{\mathit{Depart}, \mathit{Return}_S, \mathit{Return}_G\}$ be the cluster's moving state estimated by mod_i that is defined in Section 4.3.1, $q_i \in \{\mathit{Include}, \mathit{Exclude}\}$ be the state of mod_i whether the index i is included in $G_V[t]$, ${}^i\boldsymbol{p}_j$ be the position of o_j with respect to $\sum_i$, and ${}^i\boldsymbol{f}_j$ be the foot toe position of mod_j with respect to $\sum_i$. Each mod_i always locally obtains c_j from $\mathrm{mod}_j(\forall j \in N_i)$. $c_j(\forall j \in N_i)$ is used to determine whether mod_i moves the cluster to the goal or the starting area. Moreover, each mod_i always locally obtains $q_{R(i)}$, ${}^i\boldsymbol{p}_{R(R(i))}$, ${}^i\boldsymbol{f}_{R(i)}$, and $\sigma_{R(i)}$ from $\mathrm{mod}_{R(i)}$ and $q_{L(i)}$, ${}^i\boldsymbol{p}_{L(L(i))}$, ${}^i\boldsymbol{f}_{L(i)}$, and $\sigma_{L(i)}$ from $\mathrm{mod}_{L(i)}$. Those information are used to calculate the stride range or predict the success or failure of the environment traversal for the cluster.

4.2.3 Initial configuration of the cluster

Let G_{config} be a constant graph whose set of vertices is equivalent to $G_V[0]$. As for the edge set, each vertex $i \in G_V[0]$ is connected in an undirected way to two vertices as follows: One is satisfied $\underset{j\in G_V[0]\setminus\{i\}}{\arg\min}\ {}^i\varphi_j$, subject to ${}^i\varphi_j \geq 0$ and the other $\underset{k\in G_V[0]\setminus\{i\}}{\arg\max}\ {}^i\varphi_k$, subject to ${}^i\varphi_k \leq 0$. For an initial configuration of the cluster, G_{config} is defined uniquely. If $G_{\mathrm{config}} \in G[0]^*$ holds, all $\mathrm{mod}_i(i \in G_V[0])$ can find a set of $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ and construct $G[0]$; thus, the cluster can achieve static walking with a stability margin $h\tan\alpha_{\mathrm{str}}$. For example, a set of red lines written in Fig. 4.1(d) is equivalent to G_{config} such that $G_{\mathrm{config}} \in G[0]^*$ holds.

For the initial cluster to achieve static walking with a stability margin $h \tan \alpha_{\mathrm{str}}$, we assume that a certain number of modules have connected into a cluster whose configuration satisfies that $G_{\mathrm{config}} \in G[0]^*$.

4.2.4 Control purpose

The control purpose in this study is to construct a control system which satisfies the following based on the distributed control framework in Section 4.2.2:

- A cluster reaches the goal by function of adaptive reconfiguration to the difficulty of environments.
- A cluster never falls down.

4.3 Proposed control system

4.3.1 Outline of proposed algorithm

We propose autonomous distributed system for each module to reach the goal in the slope and gap environments as shown in Algorithm 2. First, each module performs COM-C calculation as explained in Section 3.2.1 and $G[0]$ construction as explained in Section 3.2.2, which are required to achieve static walking of the cluster. Then, the cluster starts walking to the goal. While the cluster walks to the goal, each module locally monitors whether the cluster can keep static walking using the monitoring for slope environment and that for gap environment based on the estimated COM-C information. A situation such that the cluster is not guaranteed to keep static walking is denoted by "dangerous situation". If mod_i in the cluster detects that the cluster falls into the dangerous situation by monitoring for slope (gap) environment, the module changes its c_i state from *Depart* to $Return_S$ ($Return_G$). The $Return_a (a \in \{S, G\})$ state is transmitted to the physically adjacent $\mathrm{mod}_j (\forall j \in N_i)$ and they change their c_j state from *Depart* to $Return_a$; thus, $c_i = Return_a \forall i$ holds within a finite time. Then, the cluster returns to the starting area to recruit one free module at the connecting position determined by each module to avoid dangerous situation. This procedure is repeated and the cluster enlarges by adding free modules until the cluster reaches the goal safely. In following subsections, we explain the detail of Algorithm 2.

4.3.2 Foot toe trajectory of each module

Considering the walking on the slope and gap environments, we modify the foot toe trajectory of each module from Fig. 3.2 to Fig. 4.3. The time-dependent phase of mod_i, that is, $\theta_i[t] \pmod{2\pi}$, changes with time based on the phase velocity of mod_i, that is, ω_i, as shown in Fig. 4.3(a). The length from the ground to the highest foot toe position in the flight phase

Algorithm 2 Proposed algorithm for each mod_i

```
while The cluster has not reached the goal do
  c_i = Depart
  COM-C calculation
  G[0] construction
  repeat
    depart to the goal monitoring dangerous situations
  until The cluster reached the goal or ∃j ∈ {i} ∪ N_i s.t. c_j = Return_S or ∃j ∈ {i} ∪ N_i
  s.t. c_j = Return_G
  if ∃j ∈ {i} ∪ N_i s.t. c_j = Return_S then
    c_i = Return_S
    Return to the starting area
    Recruit one free module to a connecting position to help the cluster traverse the slope
    environment
  else if ∃j ∈ {i} ∪ N_i s.t. c_j = Return_G then
    c_i = Return_G
    Return to the starting area
    Recruit one free module to a connecting position to help the cluster traverse the gap
    environment
  end if
end while
```

is denoted by d. The moving velocity of mod_i is denoted by ${}^i\boldsymbol{V}$, which is equivalent to the cluster's desired translational velocity with respect to $\sum_i$. The stance rate of mod_i is denoted by β_i. In the following, we explain the detail of the modifications.

First, we add a condition for the transition from the flight phase to the stance phase. In the flight phase, each module searches the ground from the ground search start point until the ground search end point (Fig. 4.3(c)) to measure the ground existence under the foot toe. The constant length between the maximum ground search depth and the ground at a flat environment is denoted by d_s. After detecting the ground or reaching the ground search end point, the module transits from the flight phase to the stance phase by resetting $\theta_i[t]$. When a module does not detect the ground until the foot toe reaches the ground search end point, the module performs a dangerous situation monitoring (see Section 4.3.3).

Second, we add a condition for the transition from the stance phase to the flight phase. When the cluster walks on slope and gap environments, the foot toe of $\mathrm{mod}_{R(i)}$ or $\mathrm{mod}_{L(i)}$ in the flight phase does not always touch the ground and transit from the flight phase to the stance phase at the phase 2π. In that case, if mod_i transits the phase state from the stance phase to the flight phase at $\theta_i[t] = 2\beta_i\pi$ as shown in Fig. 3.2, the condition (3.7) does not hold and then P cannot be defined. Therefore, each mod_i should keep the stance phase until both

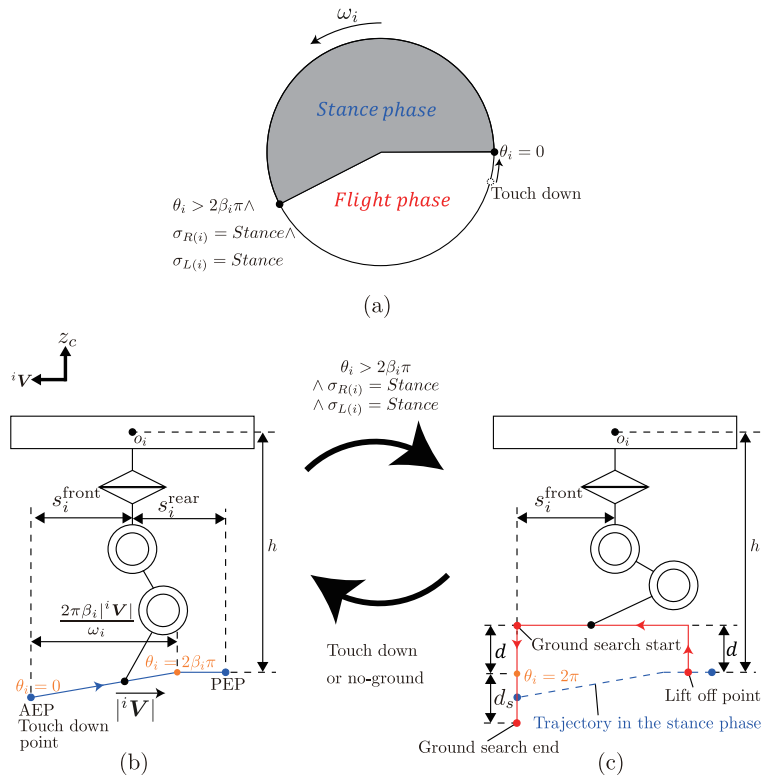


Fig. 4.3 Foot trajectory of mod_i in the proposed system with respect to $\sum_i$. (a) Phase. (b) Trajectory in the stance phase (blue). The foot toe position is controlled to reach a lift off position with distance h to o_i in the direction of z_c when $\theta_i[t] \geq 2\beta_i\pi$ holds. Therefore, to traverse slope environment with varying gradients, the cluster's body posture becomes parallel to the current gradient of the slope. (c) Trajectory in the flight phase (red). As a reference, the trajectory in the stance phase is represented by dotted blue color.

$\text{mod}_{R(i)}$ and $\text{mod}_{L(i)}$ become the stance phase.

Third, we newly introduce s_i^{front} (s_i^{rear}) as the length between o_i and AEP (PEP). In the previous study, AEP and PEP were determined without considering the stability margin to satisfy the condition (3.8). However, in this study, owing to the condition (4.1) about the graph G, each mod_i can determine the lengths s_i^{front} and s_i^{rear} considering the stability margin to satisfy the following condition (4.2) using information of ${}^i\boldsymbol{p}_{R(R(i))}$, ${}^i\boldsymbol{p}_{R(i)}$, ${}^i\boldsymbol{p}_{L(i)}$, and ${}^i\boldsymbol{p}_{L(L(i))}$:

$$\begin{aligned} &\text{A circle with the radius of } h\tan\alpha_{\text{str}} \text{ around COM-C} \\ &\cap \left(D_{R(R(i)),i} \cup D_{R(i),i} \cup D_{i,L(i)} \cup D_{i,L(L(i))}\right) = \emptyset. \end{aligned} \tag{4.2}$$

Please see Appendix D for the detailed stride range calculation. The graphical situation satisfying the condition (4.2) is shown in Fig. 4.4. Moreover, the foot toe in the stance phase is always moved at speed $|{}^i\boldsymbol{V}|$ in the direction of ${}^i\boldsymbol{V}$ within the range of AEP and PEP regardless of $\theta_i[t]$. If the foot toe moving speed and direction are not identical to all modules in the stance phase, the cluster cannot achieve the desired movement such that the COM-C

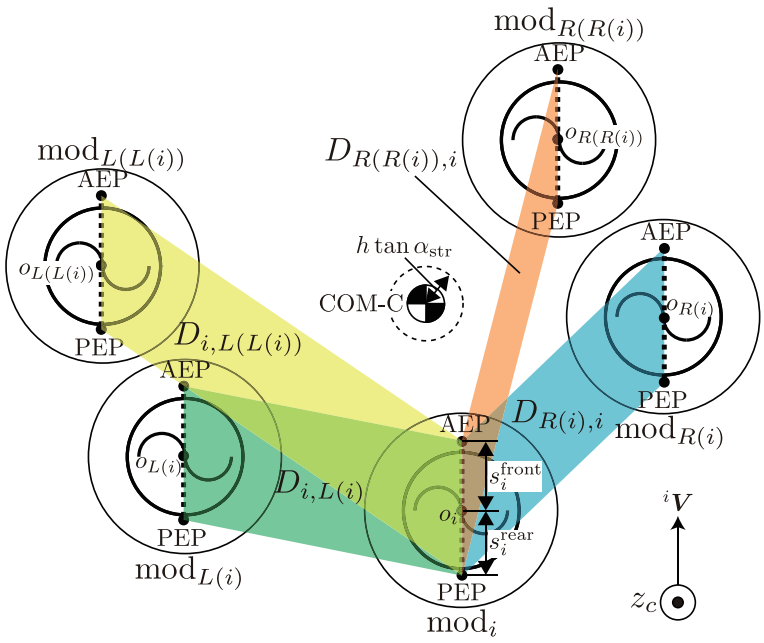


Fig. 4.4 Geometric relationship between $D_{R(R(i)),i}$, $D_{R(i),i}$, $D_{i,L(i)}$, $D_{i,L(L(i))}$, and COM-C to satisfy the condition (4.2)

is moved towards the goal. Owing to the strategy, the foot toe of mod_i in the stance phase moves at speed $|{}^i\boldsymbol{V}|$ even in the following situations: First, $\theta_i > 2\beta_i\pi$ holds but the condition for phase transition to the flight phase does not hold. When the phase transition rule to the flight phase is added to the previous foot toe trajectory as shown in Fig. 3.2, the foot toe stops at the lift off position $\theta_i = 2\beta_i\pi$, but the foot toe trajectory in Fig. 4.3 does not stop it. Second, when $G_V[t]$ dynamically changes in the gap environment, the variables ω_i and β_i are changed to different values in accordance with the change of $G[t]$. In the previous foot toe trajectory, the foot toe position in the stance phase suddenly changes to a different position in accordance with the change of ω_i and β_i, but the foot toe trajectory in Fig. 4.3 does not change it suddenly.

The foot trajectory of each mod_i includes nine variables h, d, d_s, s_i^{front}, s_i^{rear}, ω_i, $\theta_i[t]$, β_i, and ${}^i\boldsymbol{V}$. Lengths h, d, and d_s are constant values common to all modules. s_i^{front} and s_i^{rear} are determined as explained in this section. ω_i is determined to a common value such that θ_i reaches $2\beta_i\pi$ by the time the foot toe reaches PEP in a flat environment as explained in Section 3.2.4. $\theta_i[t]$ and β_i are determined as explained in Section 3.2.5. The velocity ${}^i\boldsymbol{V}$ is determined such in a way that COM-C moves towards the goal at a given speed $|{}^i\boldsymbol{V}|$ as

$$ {}^i\boldsymbol{V} = |{}^i\boldsymbol{V}| \frac{{}^i\boldsymbol{r}_{\mathrm{goal}} - {}^i\boldsymbol{g}}{|{}^i\boldsymbol{r}_{\mathrm{goal}} - {}^i\boldsymbol{g}|}, \tag{4.3} $$

where ${}^i\boldsymbol{r}_{\mathrm{goal}}$ is the goal position with respect to $\sum_i$, which is measured by the IR sensor. When all the variables are determined as above, the foot toe position is determined uniquely

through interaction with the ground and then the cluster achieves static walking with a certain stability margin in a flat environment.

4.3.3 Decentralized dangerous situation monitoring

While the cluster departs to the goal, each module monitors whether the cluster can keep static walking through two types of monitorings, stability margin monitoring and G sustainable monitoring. First, the stability margin monitoring is performed to warn the situation as following: When the cluster walks on a steep slope, the cluster's posture inclines too much. Then, a line that is parallel to the gravitational vector and passes through the COM-C goes out of the support polygon and finally the cluster falls down.

Second, we explain the purpose of G sustainable monitoring. When the cluster walks on a gap, some foot toes are on the gap during their stance phase and therefore cannot physically support the cluster. The index of the corresponding module is left spontaneously from G_V. G sustainable monitoring is performed to warn the situation as following: After an index of a module leaves from G_V, the rest modules cannot construct G and therefore cannot achieve periodic static walking of the cluster based on the gait generation system. In next subsections, we explain detail of the two types of monitorings.

Stability margin monitoring

For the stability margin monitoring, each mod_i obtains ${}^i\boldsymbol{f}_{R(i)}$, ${}^i\boldsymbol{f}_{L(i)}$, $\sigma_{R(i)}$, and $\sigma_{L(i)}$ via communication and measures the cluster's posture by IMU. Stability margin monitoring is composed of two types of monitorings, that is, continuous monitoring and discontinuous monitoring.

First, we explain the continuous monitoring. Let us define a half line l with its origin COM-C that is parallel to the gravitational vector. As shown in Fig. 4.5(a), let us consider a corn, whose vertex is COM-C and axis is l, and a line segment $l_{i,j}$ between two contact points of the foot toes of mod_i and mod_j on the environment. By increasing the vertex angle, the generating line l_g of the corn has a intersection with $l_{i,j}$. Let us define $\alpha_{i,j}$ as the angle of two lines l and l_g. This angle $\alpha_{i,j}$ can be regarded as stability margin for falling down. Each mod_i continuously calculates the stability margin angle α_i as

$$\alpha_i = \begin{cases} \alpha_{R(i),i} & \left(\sigma_i = \sigma_{R(i)} = \mathit{Stance}\right) \\ \alpha_{R(i),L(i)} & \left(\sigma_i = \mathit{Flight}\right). \end{cases} \tag{4.4}$$

From the definition of P as explained in Section 3.2.2, the stability margin angle for any edge of P is calculated by any one of all modules. Let $\alpha_{\min}$ be a constant minimum angle for the stability margin for falling down. When a mod_i detects that $\alpha_i < \alpha_{\min}$ (Fig. 4.5(b)), then the COM-C is just before going outside P; therefore mod_i changes c_i from *Depart* to Return_S.

Second, we explain the discontinuous monitoring. Stability margin monitoring so far does not cover the case of phase transition ($\mathit{Flight} \leftrightarrow \mathit{Stance}$), when P changes discontinuously.

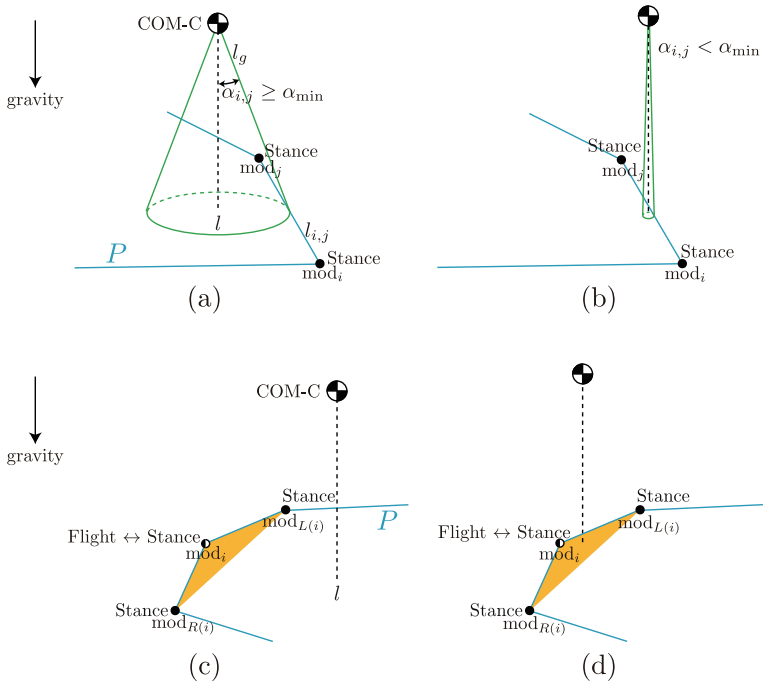


Fig. 4.5 Safety and dangerous situations for stability margin monitoring. (a) Safety situation in continuous monitoring such that stability margin angle $\alpha_{i,j} \geq \alpha_{\min}$. (b) Dangerous situation such that $\alpha_{i,j} < \alpha_{\min}$. (c) Safety situation in discontinuous monitoring such that the half line l is not within the orange region. (d) Dangerous situation such that the half line l is within the orange region.

When each mod_i transits its σ_i state, mod_i monitors whether the half line l is interior of a triangle composed of foot toe points of mod_i, $\text{mod}_{R(i)}$, and $\text{mod}_{L(i)}$. If the half line l is interior of the triangle (Fig. 4.5(d)), then the COM-C is just before going outside P; therefore mod_i changes c_i from *Depart* to $Return_S$.

For both continuous and discontinuous monitorings, when each mod_i changes c_i from *Depart* to $Return_S$, it memorizes l direction measured by IMU at that time to determine connecting place of a free module as explained in Section 4.3.5.

G sustainable monitoring

For the G sustainable monitoring, each mod_i obtains ${}^i\boldsymbol{p}_{R(R(i))}$, ${}^i\boldsymbol{p}_{L(L(i))}$, $q_{R(i)}$, and $q_{L(i)}$ via communication. When each $\text{mod}_i (i \in G_V[t])$ does not detect ground until the end of the flight phase, mod_i cannot physically support the cluster in the next stance phase as a vertex of the next support polygon. In the case, mod_i monitors whether both

$$ {}^{L(i)}\varphi_{R(i)} + |{}^{L(i)}\varphi_{L(L(i))}| < \pi \wedge {}^{R(i)}\varphi_{R(R(i))} + |{}^{R(i)}\varphi_{L(i)}| < \pi \tag{4.5} $$

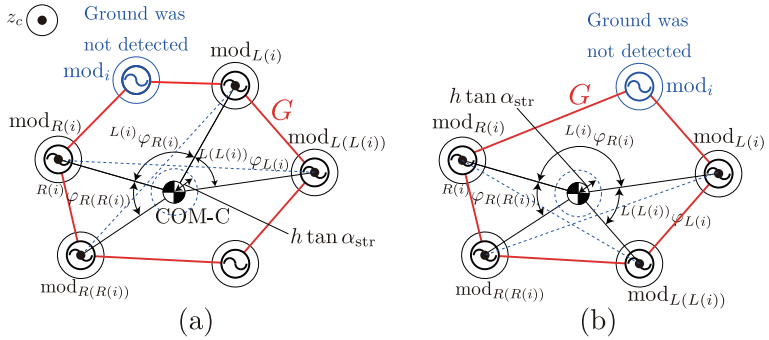


Fig. 4.6 Safety and dangerous situations for G sustainable monitoring. (a) Safety situation such that the conditions (4.5) and (4.6) hold. (b) Dangerous situation such that the condition (4.5) does not hold.

and

$$\begin{array}{l}\text{A circle with the radius of } h\tan\alpha_{\mathrm{str}} \text{ around COM-C}\\ \text{does not cross with either line segment } \overline{o_{R(R(i))}o_{L(i)}} \text{ or } \overline{o_{R(i)}o_{L(L(i))}}\end{array} \tag{4.6}$$

hold. Let $G_V[t+1]$ be $G_V[t] \setminus \{i\}$ and $\widehat{G}[t+1]$ be a graph that vertex i and edges $(i, R(i))$ and $(i, L(i))$ are excluded from $G[t]$ whereas a edge $(R(i), L(i))$ is newly added. For $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ in the graph $\widehat{G}[t+1]$, When the condition (4.5) (condition (4.6)) holds, the condition (3.6) (condition (4.1)) holds. Therefore, if both the conditions (4.5) and (4.6) hold as shown in Fig. 4.6(a), $\widehat{G}[t+1] \in G[t+1]^*$ holds and $G[t+1]$ is set as $\widehat{G}[t+1]$ by changing mod_i's state q_i from *Include* to *Exclude* and informing $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ of the change. Meanwhile, if either the condition (4.5) or the condition (4.6) does not hold as shown in Fig. 4.6(b), the cluster cannot achieve static walking and therefore mod_i changes c_i from *Depart* to $Return_G$.

When a $\mathrm{mod}_i (i \notin (G_V[t] \cup E))$ detects ground again during the ground search in the flight phase, it can physically support the cluster in the next stance phase as a vertex of the next support polygon; thus, mod_i searches $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$. If the corresponding mod_i finds $\mathrm{mod}_{R(i)}$ and $\mathrm{mod}_{L(i)}$ such that the three modules satisfy the conditions (3.6) and (4.1), then let $G_V[t+1]$ be $G_V[t] \cup \{i\}$ and q_i be changed from *Exclude* to *Include*.

4.3.4 Return to the starting area

When a mod_i whose c_i is *Depart* receives information that $c_j = Return_a (j \in N_i, a \in \{S, G\})$, mod_i transits its c_i from *Depart* to $Return_a$ (see Algorithm 2). The warning spreads to all modules soon; thus, contradiction of moving direction of each module does not happen and dangerous situation is warned from only one of monitoring. Each mod_i whose c_i is $Depart_a$ moves to reverse side of the goal for the same period between the start time and the time when $c_i = Return_a$ holds. Note that a footprint in departing to the goal and a footprint

in returning to the starting area do not match because each module starts the stance phase at AEP in Fig. 4.3(b) during the departure, whereas PEP during the return. During return, even if the cluster falls into dangerous situation, the modules neglect it and keep walking to the starting area.

4.3.5 Recruitment of a free module

When a cluster returns to the starting area, then the modules in the cluster recruit one free module to the component of the cluster. To traverse the faced environment, each module calculates the connecting place candidates independently according to the type of monitoring by which the dangerous situation of the cluster was detected.

Recruitment in response to warning from the stability margin monitoring

This recruitment based on stability margin monitoring is mainly considered to help the cluster traverse the slope environment. When the cluster detects the dangerous situation by the stability margin monitoring, the module component is in short to the direction in which the cluster's posture inclined. Therefore, a free module is recruited to a direction that the cluster inclined at the encountered dangerous situation to enlarge the support polygon in the direction. Fig. 4.7(a) shows the connecting place candidate of a free module. As mentioned in Section 4.3.3, each mod_i memorized l direction at the encountered dangerous situation. When the distance between o_i for a mod_i and a half line that l is projected on the $x_i y_i$ plane is less than $2r$, the corresponding mod_i indicates the connecting place candidate of a free module at one of the two points that the half line and the circle with center o_i and the radius $2r$ cross. Among the two points, the point whose distance from COM-C is longer is selected as the connecting place candidate of a free module. Connecting places of a real module exist discretely, so a connecting place in the module that is the nearest to the exact connecting place candidate is selected as an actual connecting place candidate under the condition that the place is empty and able to add a free module. Among the connecting place candidates of all modules, one place is selected randomly as a free module's connecting place.

Recruitment in response to warning from G sustainable monitoring

This recruitment based on G sustainable monitoring is mainly considered to help the cluster traverse the gap environment. When the cluster detects the dangerous situation by the G sustainable monitoring, the number of modules belonging to $G_V[t]$ is typically small. To increase instantaneous minimum number of modules belonging to $G_V[t]$ during gap traversal, the modules in the cluster should cross the gap at different timing. Therefore, the cluster should enlarge towards the goal or the anti-goal direction. Either direction is fine, so we limit connecting place candidates to the goal direction from COM-C to decrease the number of them. Additionally, if a free module is simply recruited to the goal direction, the cluster enlarges to

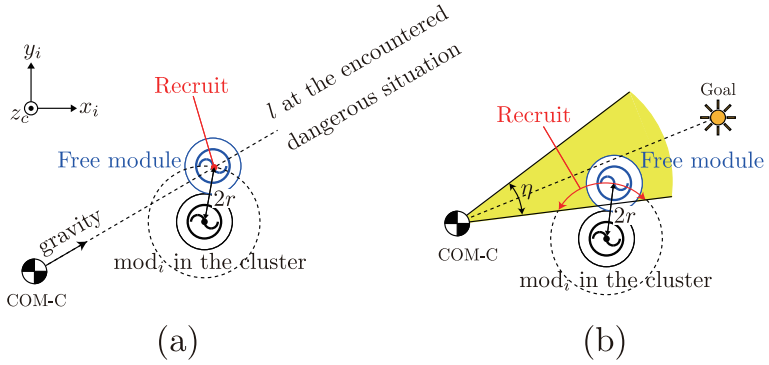


Fig. 4.7 The connecting place for a free module. (a) Recruitment in response to warning from stability margin monitoring.(b) Recruitment in response to warning from G sustainable monitoring. η is a constant viewing angle of the yellow region.

one straight direction and the stability margin for P is not large; therefore, a free module is recruited randomly within the yellow region defined by η in Fig. 4.7(b). When the distance between o_i for a mod$_i$ and an edge of the yellow region is less than $2r$, the corresponding mod$_i$ indicates the connecting place candidates of a free module at all places that are included in the intersection of the yellow region and the circle with center o_i and the radius $2r$ cross. The connecting places of a real module exist discretely, so all connecting places that are within the connecting place candidates are selected as actual connecting place under the condition that the place is empty and able to add a free module. Among the connecting place candidates of all modules, one place is selected randomly as a free module's connecting place.

4.3.6 Preparation for the next traversal

For the new configuration after a free module was added, it is not guaranteed that $G_{\text{config}} \in G[0]^*$ holds for the new cluster's configuration. If one or more mod$_i$ cannot find a set of mod$_{R(i)}$ and mod$_{L(i)}$ in a certain time period, mod$_i$ broadcasts "Recruit again" signal to all other modules and one more free module is recruited in the same way to the previous recruitment.

4.4 Dynamic simulation

We investigate the performance of dangerous situation monitoring and adaptive recruitment using dynamic simulation. For the simulation, we used a dynamic simulator lpzrobots (http://robot.informatik.uni-leipzig.de/software/). In the dynamic simulation, each module was allowed to connect with other modules at any place around the body (not limited to per 30 degree as shown in Fig. 4.1(b)) to investigate the validity of the proposed recruitment exactly. Parameter values used in the simulation are shown in Table 4.3.

Table 4.3 Parameter values.

Symbol	Parameter value
r	60[mm]
h	200[mm]
d	20[mm]
$\|^i\boldsymbol{V}\|$	10[mm/s]
d_s	20[mm]
α_{close}	0.18[rad]
α_{str}	0.12[rad]
α_{min}	0.02[rad]
η	$\frac{\pi}{5}$[rad]

4.4.1 Simulation setting

Environment traversal by proposed system

First, we validate that several types of initial clusters traverse several types of slope and gap environments based on the proposed system. We prepared total nine types of environments as follows: two types of slope environments $(\lambda, \kappa, L) = (3[\text{deg}], 90[\text{deg}], 0.5[\text{m}])$ and $(\lambda,\ \kappa,\ L) = (2[\text{deg}], 30[\text{deg}], 0.5[\text{m}])$. For both types, we prepared three types of difficulty of the slope environments $M = 2, 5, 10$. Meanwhile, we prepared gap environments with three types of difficulty $W = 0.02, 0.05, 0.1[\text{m}]$.

To verify that the proposed system is applicable to any initial configurations of the cluster, we consider five types of initial configurations, namely, hexapod, regular hexagon, hexapod with 90 degree rotation, h-shaped, and irregular-shaped. All of them are composed of six number of modules.

Moreover, the proposed system includes some randomness. For example, initial phase of each module $\theta_i[0]$ is determined randomly and a free module's connecting place is determined randomly from all candidates. To verify that the randomness does not give a significant influence to the proposed system, we used three different random seeds.

As a summary, for each of the total nine environments above, fifteen simulations were carried out, that is, three random seed for five types of initial configurations. For each of the environments, we summarized the average and the standard deviation (SD) for the number of added free modules in the final configuration of the cluster in the fifteen simulations.

Evaluation of the adaptive recruitment

To evaluate the performance of adaptive recruitment, we prepared "random connection system" denoted as *Rand-s*. *Rand-s* is similar to the proposed system except that connecting place of the free module is randomly determined from all physically available candidates without warning information from monitoring systems. For *Rand-s*, we performed same simulations of

the proposed system. If the cluster was not able to reach the goal within addition of twelve free modules, then the simulation was counted as failure.

4.4.2 Simulation result

Environment traversal by proposed system

Fig. 4.8 shows the graphical procedure such that three types of initial clusters traversed three types of environments based on the proposed system. We can observe that all the clusters detect dangerous situations and then recruit free modules in response to the dangerous situation. To validate the effectiveness of the dangerous situation monitoring, we observed that a six-legged hexapod cluster without dangerous situation monitoring fell down in two slope environments and a gap environment (Fig. 4.9).

Additionally, the number of free module recruited to the cluster correlates with difficulty of the environments (Fig. 4.10). Therefore, in the proposed system, the number of modules in the cluster can be adaptively tuned in response to the environmental difficulty.

Furthermore, we summarized relationship between three types of environments and corresponding morphologies of the clusters which reached the goal (Fig. 4.11). In two types of the slope environments, free modules were recruited to the gradient direction of each slope as explained in Section 4.3.5. We observed that one of the legs of the cluster did not touch ground on the slope environment and therefore the cluster recruits a free module in response to warning from G sustainable monitoring (Fig. 4.11(h)). In the gap environments, free modules were recruited to the direction towards the goal with a certain randomness as explained in Section 4.3.5.

Evaluation of the adaptive recruitment

We evaluate the performance of adaptive recruitment explained in Section 4.3.5 by comparing proposed system and *Rand-s*. From Fig. 4.10, we can observe that the average number of free modules in the final clusters in the proposed system is fewer than that of *Rand-s* for all the nine environments. Therefore, we conclude that adaptive recruitment outperforms random recruitment in terms of the slope and gap environment traversal.

We report rare cases such that a cluster with dangerous situation monitoring fell down in the gap environment in the following: First, during return to the starting area, one of the foot toes fell into the gap and then the cluster fell into the dangerous situation. During the return, the modules neglected the dangerous situation, so the cluster fell down. Second, one of the foot toes located at the edge of the ground slipped into the gap during the stance phase and was not able to support the cluster. This is because of the dynamic behavior, which was not assumed in the proposed system. We neglected those rare cases; thus, when the rare cases happen, initial phase of each module is set to another random value and then the cluster departs to the goal again.

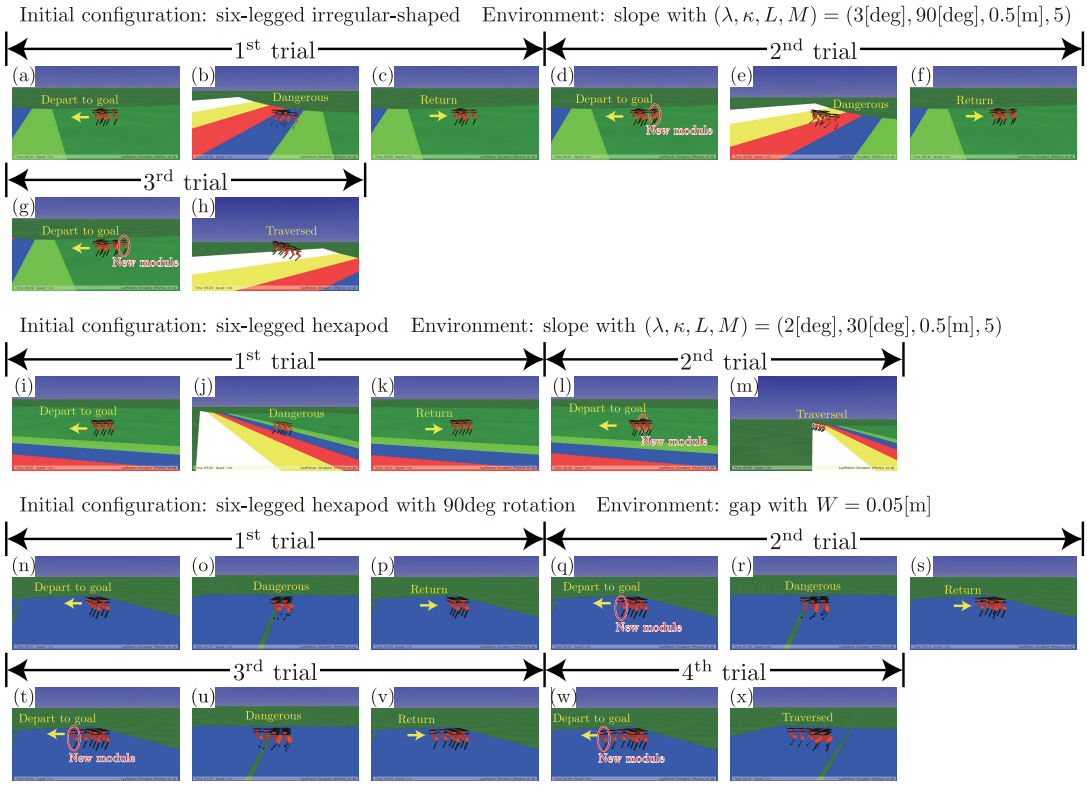


Fig. 4.8 Procedure such that initial clusters added free modules in response to the dangerous situation and then reached the goal. Top: Initial configuration is six-legged asymmetric and environment is a slope with $(\lambda, \kappa, L, M) = (3[\text{deg}], 90[\text{deg}], 0.5[\text{m}], 5)$. (a) Initial six-legged cluster departs to the goal. (b) Dangerous situation for six-legged cluster. (c) Six-legged cluster returned to the starting area to recruit a free module. (d)-(f) Repeat of (a)-(c) for seven-legged cluster. (g) Eight-legged cluster departs to the goal. (h) Eight-legged cluster reaches the goal. Middle: Initial configuration is six-legged hexapod and environment is a slope with $(\lambda, \kappa, L, M) = (2[\text{deg}], 30[\text{deg}], 0.5[\text{m}], 5)$. (i)-(k) Repeat of (a)-(c) for six-legged cluster. (l) Seven-legged cluster departs to the goal. (m) Seven-legged cluster reaches the goal. Below: Initial configuration is six-legged hexapod cluster with 90 degree rotation and environment is a gap with $W = 0.05[\text{m}]$. (n)-(p) Repeat of (a)-(c) for six-legged cluster. (q)-(s) Repeat of (a)-(c) for seven-legged cluster. (t)-(v) Repeat of (a)-(c) for eight-legged cluster. (w) Nine-legged cluster departs to the goal. (x) Nine-legged cluster reaches the goal.

4.5 Robot experiment

In this section, we verify that real robot modules traverse slope and gap environments through dangerous situation monitoring and adaptive recruitment.

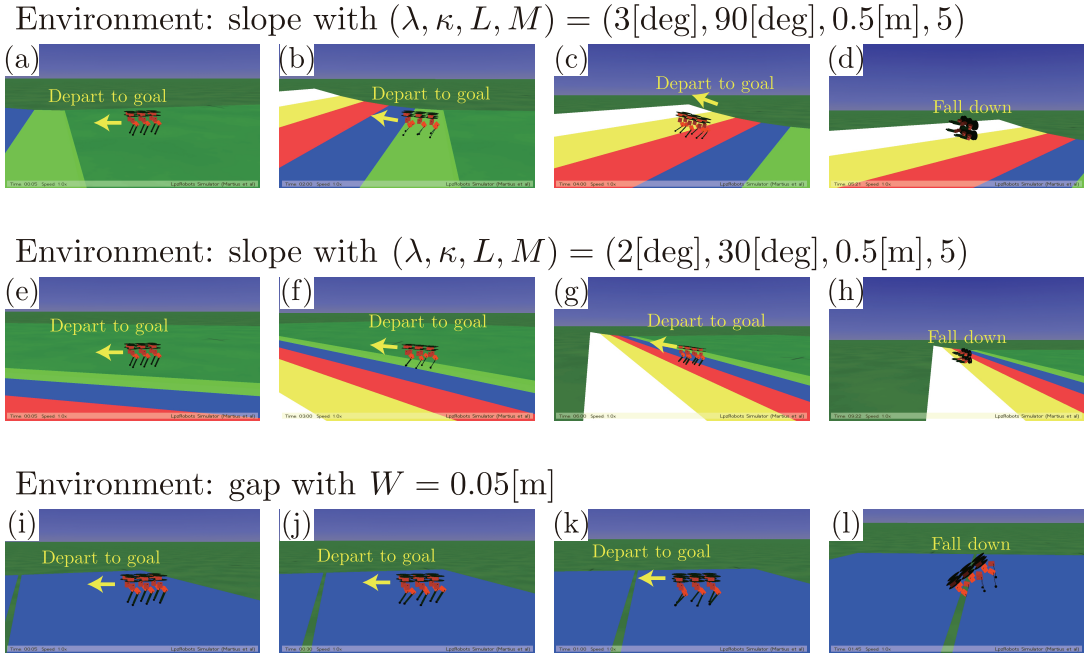


Fig. 4.9 Procedure such that six-legged hexapod cluster falls down in three environments without dangerous situation monitorings.

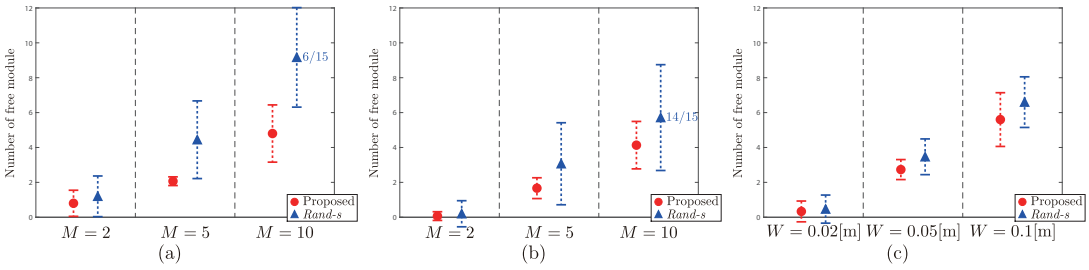


Fig. 4.10 Relationship between an environment and the number of added free modules in a final cluster that reached the goal for the proposed system (Proposed) and the random connection system (*Rand-s*). (a) Slope with $(\lambda, \kappa, L) = (3[\text{deg}], 90[\text{deg}], 0.5[\text{m}])$. (b) Slope with $(\lambda, \kappa, L) = (2[\text{deg}], 30[\text{deg}], 0.5[\text{m}])$. (c) Gap environment. Fraction aside mark is a success rate for fifteen times of simulations, where success means that the cluster reaches the goal within addition of twelve free modules.

4.5.1 Experimental setting

As for a slope environment, we prepared a slope with $(\lambda, \kappa, L, M) = (3[\text{deg}], 90[\text{deg}], 0.3[\text{m}], 4)$. To obtain enough friction between the foot toe and the ground, a rubber sheet was laid on the slope environment. As for a gap environment, we prepared a gap with $W = 0.05[\text{m}]$. For both environments, we prepared a light source as a goal position. Parameter values used in

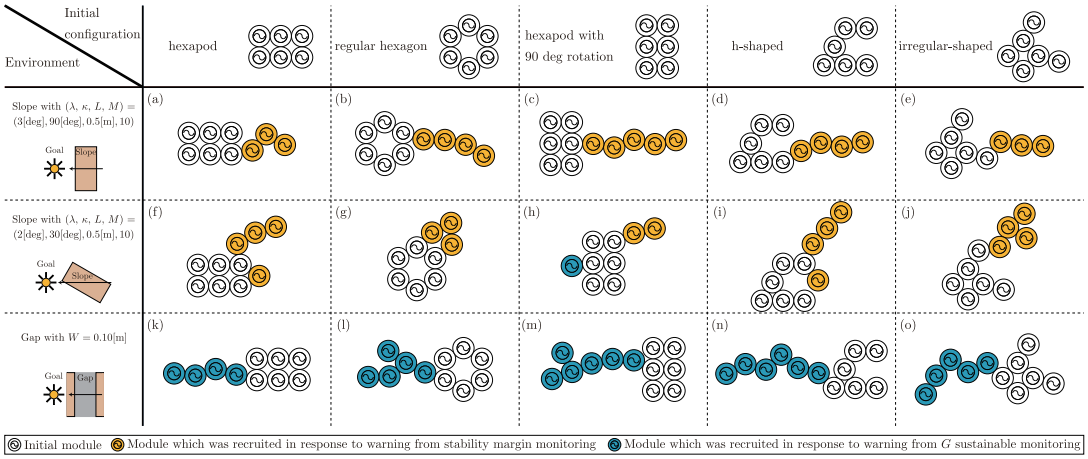


Fig. 4.11 Relationship between each environment and free module's connecting place in a final cluster that reached the goal for each initial configuration. For each cluster, one of the typical morphologies in the three times of simulations is shown.

experiments are same as that of simulations shown in Table 4.3.

An initial configuration of the cluster was set to the six-legged hexapod type. When the cluster recruits a free module, each module turns on its blue LEDs (see Fig. 4.1(b)) to indicate connecting place candidates of a free module. As mentioned in Section 4.1.1, physical reconfiguration was performed manually.

4.5.2 Experimental result

Fig. 4.12 shows the procedure for a cluster of a single-legged modular robot that traverses a slope environment with (λ, κ, L, M) = (3[deg], 90[deg], 0.3[m], 4) by reconfiguring from the six-legged hexapod to an eight-legged cluster. We can observe that the cluster detected dangerous situation before falling down (Fig. 4.12(b) and (f)), then they returned to the starting area (Fig. 4.12(c) and (g)) to recruit a free module. The free modules were recruited to the slope's gradient direction (Fig. 4.12(d) and (h)). Because of the IMU's measurement error, the cluster's posture measured by modules were not identical and therefore several LEDs indicated different connecting places. One of the LED places is randomly selected as a free module's connection position. Finally, eight-legged cluster traversed the slope environment (Fig. 4.12(l)).

Fig. 4.13 shows the procedure for a cluster of a single-legged modular robot that traverses a gap environment with $W = 0.05$[m] by reconfiguring from the six-legged hexapod to an eight-legged cluster. We can observe that the cluster detected the dangerous situation (Fig. 4.13(b) and (f)) just after one of the legs fell into the gap. Thus, the cluster did not fall down, and then it returned to the starting area (Fig. 4.13(c) and (g)) to recruit a free module. The free

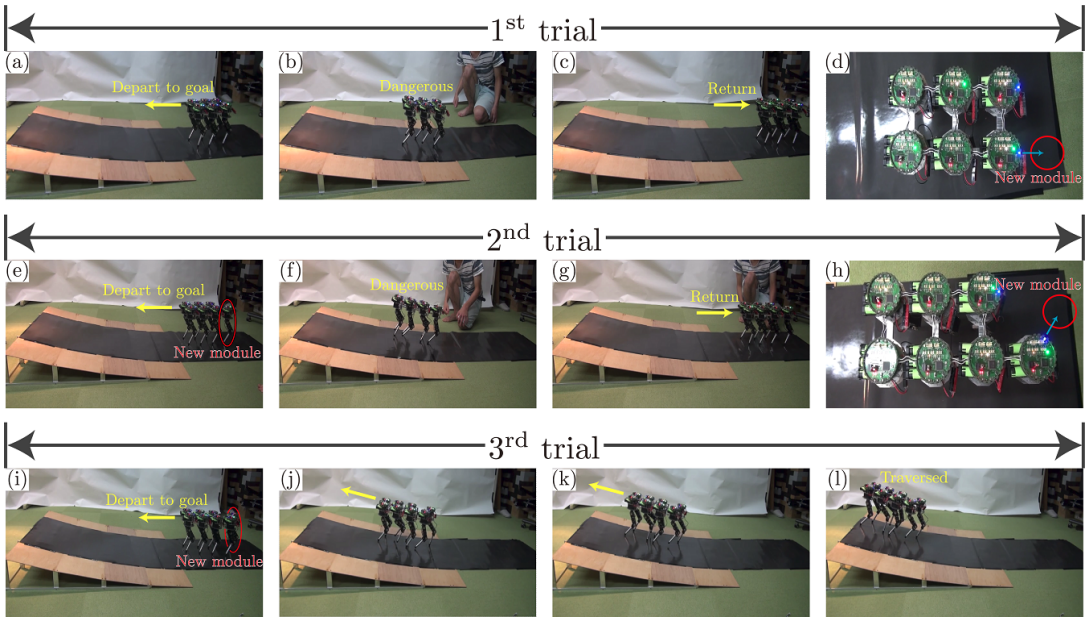


Fig. 4.12 Procedure for a cluster that traverses a slope environment with (λ, κ, L, M) = (3[deg], 90[deg], 0.3[m], 4) in the experiment. (a) Initial state for six-legged cluster. (b) Dangerous situation for six-legged cluster. (c) Six-legged cluster returns to the starting area and recruits a free module. (d) One of the two lighting LEDs is selected randomly as a free module's connection place. (e) Initial state for seven-legged cluster. A module enclosed by a red circle is the new recruited module. (f) Dangerous situation for seven-legged cluster. (g) Seven-legged cluster returned to the starting area and recruits a free module. (h) One of the two lighting LEDs is selected randomly as a free module's connection place. (i) Initial state for eight-legged cluster. A module enclosed by a red circle is the new recruited module. (j)(k) Eight-legged cluster departs to the goal climbing the slope. (l) One of the modules in eight-legged cluster reaches the edge of the slope.

modules were recruited to the goal direction (Fig. 4.13(d) and (h)). Finally, an eight-legged cluster was able to keep walking even after one of the legs fell into the gap (Fig. 4.13(j)) and traversed the gap environment (Fig. 4.13(l)).

4.6 Discussion and conclusion

In this study, we proposed an autonomous distributed system for a single-legged modular robot KARAKASA to traverse unknown slopes and gaps. We constructed the dangerous situation monitoring for the cluster and the adaptive recruitment such that the connecting place of a free module is determined in response to the encountered dangerous situation. Through dynamic simulations and robot experiments, we validated that the dangerous situation monitoring prevents the cluster in several morphologies from falling down. Furthermore, the connecting place of a free module determined by the proposed adaptive recruitment outper-

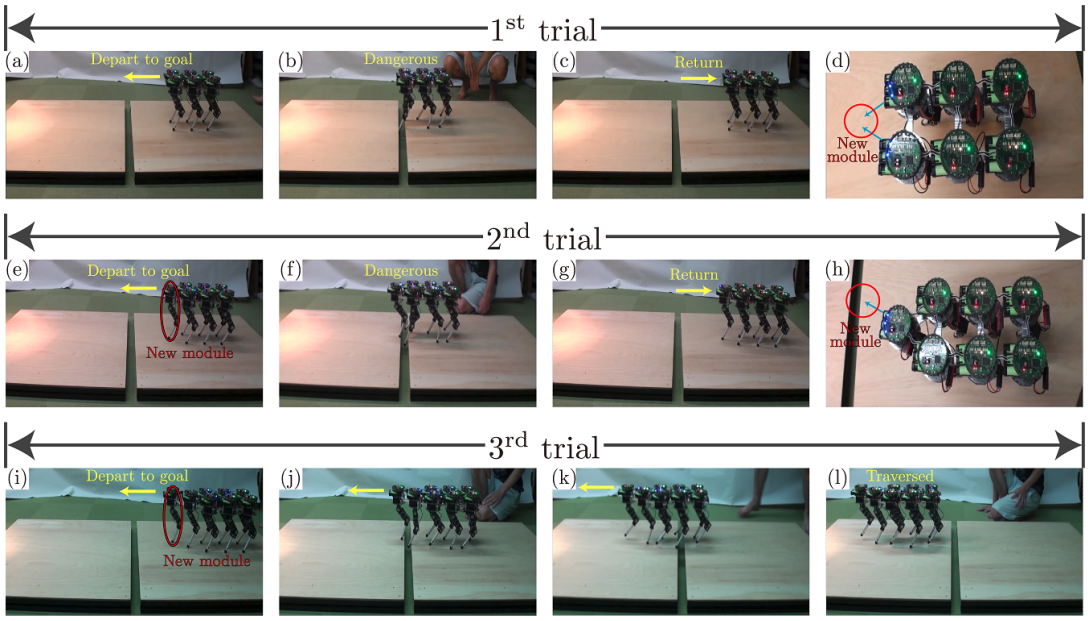


Fig. 4.13 Procedure for a cluster that traverses a gap environment with $W = 0.05$[m] in the experiment. (a) Initial state for six-legged cluster. (b) Dangerous situation for six-legged cluster. (c) Six-legged cluster returned to the starting area and recruits a free module. (d) One of the four lighting LEDs is selected randomly as a free module's connection place. (e) Initial state for seven-legged cluster. A module enclosed by a red circle is the new recruited module. (f) Dangerous situation for seven-legged cluster. (g) Seven-legged cluster returned to the starting area and recruits a free module. (h) One of the three lighting LEDs is selected randomly as a free module's connection place. (i) Initial state for eight-legged cluster. A module enclosed by a red circle is the new recruited module. (j)(k) Eight-legged cluster departs to the goal traversing the gap. (l) All modules in the eight-legged cluster reaches the goal side of the gap.

forms that by the random recruitment (*Rand-s*) in terms of difficult slope and gap traversal. In this study, one module is recruited to the cluster per each environment traversing trial, but the number of modules recruited to the cluster can be easily extended to multi modules by repeating the recruiting procedure for one free module.

Compared with existing reconfiguring strategies using task-configuration library, the proposed system does not require considerable efforts to determine the desired morphology a priori; instead the morphology is adaptively changed through interaction with the environment. Moreover, the existing reconfiguring strategies using task-configuration library such as [121, 123] are centralized control systems that use global information about the current configuration of the cluster; whereas the proposed system does not need any leader and all the modules use only local information.

Meanwhile, O'Grady et al. [11] constructed the dangerous situation monitoring for each module, whereas we constructed the dangerous situation monitoring for the cluster. Therefore

in our study, the cluster can avoid falling down when the number of modules in the cluster is in short to traverse the environment. Moreover, the number of free modules in the cluster is tuned in response to the difficulty of the given environment.

4.6.1 Comparison between the proposed and the existing systems for a multi-legged robot to traverse slope and gap environments

In existing systems to traverse a slope environment, the posture of the robot body is kept perpendicular to the gravity vector by adjusting all foot toe positions with respect to the robot body [126, 127]. They cannot be applied to a steep slope which the robot body cannot be kept perpendicular to the gravity vector because of the limit of the leg length.

Meanwhile, in existing systems to traverse a gap environment, when a foot toe in the flight phase does not touch the other side ground, the foot toe searches the ground by modifying the trajectory until the foot toe touches the ground [128, 129]. They cannot be applied to a wide gap which a foot toe does not reach the other side ground.

In the proposed system, the support polygon of the cluster can be enlarged by adding free modules. Therefore, the proposed system has the potential to traverse wider range of slope and gap environments than the existing systems.

4.6.2 Limitations of the proposed system

The stability margin for monitoring dangerous situation is calculated for a polygon P. P is not a corresponding support polygon ($\subseteq$ support polygon). The stability margin based on the polygon P is conservative compared to the exact stability margin based on the support polygon. Therefore, even when the exact stability margin for the support polygon of the cluster is large, the modules in the cluster may identify the safety situation as a dangerous situation. The misunderstanding of the dangerous situation leads to excessive recruitment of free modules. Additionally, regarding adaptive recruitment, there might not be a module in the cluster which can indicate the connecting place. For example, when modules in V-shaped cluster want to indicate the just upward direction of the "V" from COM-C that is located at the valley of "V", there is no module near to the upward direction and therefore no module can indicate the connecting place by turning on the LED.

4.6.3 Future extensions

As future extensions, we would like to resolve the limitations of the proposed system discussed in Section 4.6.2 through improvement of hardware perspective. For example, modifying the possible configurations is one of the solutions.

Chapter 5

Conclusion

Through this thesis, individual behaviors that estimate the global information using local interactions and thereby change their own behaviors were focused on, and how the changes in individual behavior based on global information enhance the environmental adaptability of the group was discussed from system analysis and system synthesis.

In Chapter 2, how change in the behavior of individual ants based on estimated global information enhances the inner and outer environmental adaptation of an ant colony was analyzed, where inner environment refers to the colony size and outer environment refers to the amount of food in the outside nest, midden collapse, and debris inflow. Each ant can estimate the colony size based on its surrounding nestmate density and the nutritional state of the colony based on its own nutritional state that is averaged by frequent trophallaxis. The simulation reveals that when individual ants decrease their contact rates in response to increasing colony size, the transition to the foraging task that induces high death rate is suppressed; thus, they enable to maintain a large colony size. Furthermore, when individual ants increase their transition rate to the foraging task in response to decreasing nutritional state of the colony, they increase resilience to outer environmental changes.

In Chapter 3, a unified controller of individual single-legged reconfigurable modular robots that compose multi-legged robot to achieve static walking of the multi-legged robot is constructed. The cluster can take various inner environment, that is, various leg configurations including the leg malfunction. Inspired by ants' frequent trophallaxis strategy in Chapter 2, each module estimates the global information of the cluster, that is, the center of mass of the cluster, using local information exchange based on the average-consensus control. Then, each module utilizes the information to construct interlimb network, and then it determines its own foot toe position based on local information of modules that are adjacent at the interlimb network so that the support polygon always encloses the center of mass of the multi-legged robot. Using the proposed system, clusters with various configurations including leg malfunction were mathematically guaranteed to achieve static walking and achievement of static walking for various clusters was demonstrated through dynamic simulations and robot experiments.

In Chapter 4, a unified controller of individual single-legged modules so that the cluster

repeatedly adds a new module to the cluster and tries to traverse the outer environment, and finally adapts the morphology to the environment was constructed. The outer environment refers to the slope and gap environments with several difficulties. Each module is required to detect whether the current morphology of the cluster can traverse the environment or is required to add a new module to avoid the danger of falling down. For the decentralized monitoring, the center of mass of the cluster information is important; thus, each module estimates the global information of the cluster by the average-consensus control as mentioned in Chapter 3. Using the estimated center of mass information, each module monitors the risk of falling down in a distributed manner from geometrical relationship between the center of mass and the support polygon of the cluster. Through dynamic simulations and robot experiments, it was demonstrated that several types of initial clusters adaptively changed the morphology in response to the type and difficulty of outer environment and successfully traversed the environment.

5.1 Future work

In the following, future works are summarized. Short-term future works are as follows: First one is to perform multi-agent simulation for individual ants' behavior that was investigated in Chapter 2 and verify the difference from ODE model. Second one is to construct a controller for single-legged modular robots to cooperatively achieve an object transportation task by using the leg as a manipulator. Third one is to invest a new single-legged robot hardware to achieve autonomous reconfiguration. Forth one is to construct a system combining task allocation in Chapter 2 and task execution in Chapter 4, where the optimization of the number of added free modules per recruitment in Chapter 4 is a problem.

Long-term future works are as follows: First one is to generalize a theory to construct an autonomous distributed system in which the group can adapt to several environments. Second one is to construct a controller for single-legged modular robots by which the modules can cooperatively achieve several types of tasks. Third one is to elucidate the swarm intelligence in living organisms and then apply the intelligence to artificial multi-robot systems.

Appendix A

Influence of parameter change

Table A.1 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(-1)$, $S_Y(-1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	16%	15%	9%	14%	9%	14%	67%	58%	8493	8977
p_{estop}	14%	16%	20%	7%	20%	7%	46%	71%	9328	8478
p_{iteng}	16%	14%	5%	17%	5%	17%	73%	53%	8417	9102
p_{difeng}	15%	15%	4%	17%	4%	17%	77%	51%	14026	7009
p_{larva}	0%	15%	0%	10%	0%	10%	0%	66%	0	9522
n_{food}	29%	9%	3%	18%	3%	18%	64%	54%	8037	9496
b_{larva}	15%	15%	12%	12%	11%	11%	62%	62%	8764	8764
q_x	14%	15%	15%	10%	15%	10%	56%	65%	9155	8594
q_y	15%	24%	12%	5%	11%	5%	62%	66%	8764	8212
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	15%	15%	12%	11%	11%	11%	62%	62%	8801	8730
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	15%	15%	12%	11%	12%	11%	62%	62%	8803	8728
$a_{E_{intra}}, a_{I_{intra}}$	15%	15%	12%	12%	11%	11%	62%	62%	8899	8634
p_{enemy}	15%	15%	12%	11%	12%	11%	61%	63%	8863	8667
p_{mcoll}	15%	15%	11%	12%	11%	11%	62%	62%	8772	8757
p_{din}	15%	15%	11%	12%	11%	11%	62%	62%	8769	8760
$p_{mdetect}$	15%	15%	11%	12%	11%	11%	62%	62%	8772	8757
$p_{ddetect}$	15%	15%	12%	12%	11%	11%	62%	62%	8764	8764
q_{enemy}	15%	15%	12%	12%	11%	11%	62%	62%	8764	8764
$n_{\max}$	41%	41%	1%	1%	1%	1%	57%	57%	7738	7738
n_{hunger}	15%	15%	12%	12%	11%	11%	62%	62%	8764	8764
original	15%		12%		11%		62%		8764	

Table A.2 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(0H)$, $S_Y(-1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	77%	0%	1%	0%	1%	0%	22%	0	3247
p_{estop}	81%	82%	1%	0%	1%	0%	17%	17%	2754	2779
p_{iteng}	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
p_{difeng}	80%	83%	1%	1%	1%	1%	19%	16%	2935	2630
p_{larva}	0%	80%	0%	1%	0%	1%	0%	19%	0	2962
n_{food}	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
b_{larva}	0%	82%	0%	1%	0%	1%	0%	17%	0	2766
q_x	73%	86%	1%	1%	1%	1%	25%	13%	3644	1421
q_y	80%	82%	1%	1%	1%	1%	19%	17%	2842	2764
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
$a_{E_{intra}}, a_{I_{intra}}$	82%	82%	1%	1%	1%	1%	17%	17%	2775	2757
p_{enemy}	84%	79%	1%	1%	0%	1%	15%	19%	3032	2575
p_{mcoll}	78%	84%	1%	1%	1%	0%	21%	15%	3148	2526
p_{din}	79%	84%	1%	1%	0%	1%	20%	15%	3025	2577
$p_{mdetect}$	78%	84%	1%	1%	1%	0%	21%	15%	3148	2526
$p_{ddetect}$	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
q_{enemy}	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
$n_{\max}$	82%	82%	1%	1%	1%	1%	17%	17%	2763	2763
n_{hunger}	82%	82%	1%	1%	1%	1%	17%	17%	2766	2766
original	82%		1%		1%		17%		2766	

Table A.3 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(0L)$, $S_Y(-1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	63%	0%	1%	0%	0%	0%	36%	0%	5040	0
p_{estop}	0%	56%	0%	0%	0%	0%	0%	43%	0	6016
p_{iteng}	57%	0%	0%	0%	0%	0%	42%	0%	5985	0
p_{difeng}	0%	64%	0%	1%	0%	1%	0%	35%	0	4775
p_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{food}	0%	57%	0%	1%	0%	0%	0%	42%	0	5984
b_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_x	0%	60%	0%	1%	0%	0%	0%	39%	0	5453
q_y	17%	0%	7%	0%	7%	0%	69%	0%	6683	0
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{E_{intra}}, a_{I_{intra}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{mcoll}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{din}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{mdetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{ddetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$n_{\max}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{hunger}	0%	57%	0%	1%	0%	0%	0%	42%	0	5984
original	0%		0%		0%		0%		0	

Table A.4 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(+1)$, $S_Y(-1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{estop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{iteng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{difeng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{food}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
b_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{E_{intra}}, a_{I_{intra}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{mcoll}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{din}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{mdetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{ddetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$n_{\max}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{hunger}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
original	0%		0%		0%		0%		0	

Table A.5 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(-1)$, $S_Y(0)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	43%	0%	1%	0%	1%	0%	55%	0	7713
p_{estop}	47%	41%	1%	1%	1%	1%	50%	58%	7609	7715
p_{iteng}	42%	45%	1%	1%	1%	1%	57%	53%	7694	7636
p_{difeng}	0%	55%	0%	1%	0%	1%	0%	43%	0	5630
p_{larva}	0%	40%	0%	1%	0%	1%	0%	58%	0	8514
n_{food}	0%	44%	0%	1%	0%	1%	0%	54%	0	7657
b_{larva}	0%	44%	0%	1%	0%	1%	0%	54%	0	7657
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	44%	44%	1%	1%	1%	1%	54%	54%	7657	7657
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	44%	44%	1%	1%	1%	1%	54%	54%	7657	7656
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	44%	44%	1%	1%	1%	1%	54%	54%	7657	7656
$a_{E_{intra}}, a_{I_{intra}}$	44%	44%	1%	1%	1%	1%	54%	54%	7774	7542
p_{enemy}	48%	41%	1%	1%	1%	1%	50%	57%	7860	7505
p_{mcoll}	44%	44%	1%	1%	1%	1%	54%	54%	7666	7648
p_{din}	44%	44%	1%	1%	1%	1%	54%	54%	7663	7650
$p_{mdetect}$	44%	44%	1%	1%	1%	1%	54%	54%	7666	7648
$p_{ddetect}$	44%	44%	1%	1%	1%	1%	54%	54%	7657	7657
q_{enemy}	44%	44%	1%	1%	1%	1%	54%	54%	7657	7657
$n_{\max}$	44%	44%	1%	1%	1%	1%	54%	54%	7657	7657
n_{hunger}	44%	44%	1%	1%	1%	1%	54%	54%	7657	7657
original	44%		1%		1%		54%		7657	

Table A.6 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(0H)$, $S_Y(0)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	81%	77%	1%	1%	2%	1%	16%	22%	2	3242
p_{estop}	81%	82%	1%	0%	1%	0%	17%	17%	2752	2774
p_{iteng}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
p_{difeng}	80%	83%	1%	1%	1%	1%	19%	16%	2931	2627
p_{larva}	0%	80%	0%	1%	0%	1%	0%	19%	0	2958
n_{food}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
b_{larva}	0%	82%	0%	1%	0%	1%	0%	17%	0	2762
q_x	74%	86%	1%	1%	1%	1%	25%	13%	3637	1443
q_y	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	82%	82%	1%	1%	1%	1%	17%	17%	2763	2762
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	82%	82%	1%	1%	1%	1%	17%	17%	2763	2762
$a_{E_{intra}}, a_{I_{intra}}$	82%	82%	1%	1%	1%	1%	17%	17%	2772	2753
p_{enemy}	84%	79%	1%	1%	0%	1%	15%	19%	3030	2571
p_{mcoll}	78%	84%	1%	1%	1%	0%	21%	14%	3144	2522
p_{din}	79%	84%	1%	1%	0%	1%	20%	15%	3022	2573
$p_{mdetect}$	78%	84%	1%	1%	1%	0%	21%	14%	3144	2522
$p_{ddetect}$	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
q_{enemy}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
$n_{\max}$	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
n_{hunger}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
original	82%		1%		1%		17%		2762	

Table A.7 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(0L)$, $S_Y(0)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{estop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{iteng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{difeng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{food}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
b_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{E_{intra}}, a_{I_{intra}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{mcoll}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{din}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{mdetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{ddetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$n_{\max}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{hunger}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
original	0%		0%		0%		0%		0	

Table A.8 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(+1)$, $S_Y(0)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{estop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{iteng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{difeng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{food}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
b_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{E_{intra}}, a_{I_{intra}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{mcoll}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{din}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{mdetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{ddetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$n_{\max}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{hunger}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
original	0%		0%		0%		0%		0	

Table A.9 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(-1)$, $S_Y(+1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{estop}	0%	49%	0%	0%	0%	0%	0%	51%	0	7538
p_{iteng}	49%	50%	0%	1%	0%	1%	50%	49%	7533	7517
p_{difeng}	32%	60%	0%	1%	0%	1%	67%	39%	13278	5523
p_{larva}	0%	46%	0%	0%	0%	0%	0%	53%	0	8372
n_{food}	0%	49%	0%	0%	0%	0%	0%	50%	0	7523
b_{larva}	0%	49%	0%	0%	0%	0%	0%	50%	0	7523
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	50%	49%	0%	1%	0%	1%	49%	50%	7499	7543
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	49%	49%	0%	0%	0%	0%	50%	50%	7524	7523
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	49%	49%	0%	0%	0%	0%	50%	50%	7524	7523
$a_{E_{intra}}, a_{I_{intra}}$	49%	49%	0%	0%	0%	0%	50%	50%	7639	7411
p_{enemy}	54%	46%	0%	1%	0%	1%	45%	53%	7768	7346
p_{mcoll}	49%	49%	0%	0%	0%	0%	50%	50%	7532	7514
p_{din}	49%	49%	0%	0%	0%	0%	50%	50%	7530	7516
$p_{mdetect}$	49%	49%	0%	0%	0%	0%	50%	50%	7532	7514
$p_{ddetect}$	49%	49%	0%	0%	0%	0%	50%	50%	7523	7523
q_{enemy}	49%	49%	0%	0%	0%	0%	50%	50%	7523	7523
$n_{\max}$	46%	46%	1%	1%	1%	1%	52%	52%	7605	7605
n_{hunger}	49%	49%	0%	0%	0%	0%	50%	50%	7523	7523
original	49%		0%		0%		50%		7523	

Table A.10 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(0H)$, $S_Y(+1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	85%	77%	1%	1%	1%	1%	14%	22%	3	3241
p_{estop}	81%	82%	1%	0%	1%	0%	17%	17%	2751	2774
p_{iteng}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
p_{difeng}	80%	83%	1%	1%	1%	1%	19%	16%	2931	2626
p_{larva}	0%	80%	0%	1%	0%	1%	0%	19%	0	2958
n_{food}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
b_{larva}	0%	82%	0%	1%	0%	1%	0%	17%	0	2762
q_x	74%	86%	1%	1%	1%	1%	25%	13%	3636	1445
q_y	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
$a_{E_{intra}}, a_{I_{intra}}$	82%	82%	1%	1%	1%	1%	17%	17%	2771	2753
p_{enemy}	84%	79%	1%	1%	0%	1%	15%	19%	3030	2570
p_{mcoll}	78%	84%	1%	1%	1%	0%	21%	14%	3144	2522
p_{din}	79%	84%	1%	1%	0%	1%	20%	15%	3022	2572
$p_{mdetect}$	78%	84%	1%	1%	1%	0%	21%	14%	3144	2522
$p_{ddetect}$	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
q_{enemy}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
$n_{\max}$	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
n_{hunger}	82%	82%	1%	1%	1%	1%	17%	17%	2762	2762
original	82%		1%		1%		17%		2762	

Table A.11 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(0L)$, $S_Y(+1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{estop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{iteng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{difeng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{food}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
b_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{E_{intra}}, a_{I_{intra}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{mcoll}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{din}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{mdetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{ddetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$n_{\max}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{hunger}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
original	0%		0%		0%		0%		0	

Table A.12 Relationship between colony size, the rate of engaged workers involved in each task after 15 years and parameter disturbance for the scenario ($S_X(+1)$, $S_Y(+1)$)

Parameter	Forager		Midden		Maintenance		Intra-nest tasks		Colony size	
	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%	−50%	+50%
p_{rstop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{estop}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{iteng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{difeng}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{food}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
b_{larva}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_x	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_y	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{midden}}, a_{E_{midden}}, a_{I_{midden}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{R_{nest}}, a_{E_{nest}}, a_{I_{nest}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$a_{E_{intra}}, a_{I_{intra}}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{mcoll}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
p_{din}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{mdetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$p_{ddetect}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
q_{enemy}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
$n_{\max}$	0%	0%	0%	0%	0%	0%	0%	0%	0	0
n_{hunger}	0%	0%	0%	0%	0%	0%	0%	0%	0	0
original	0%		0%		0%		0%		0	

Appendix B

Influence of leg weight on the cluster's walking

To compute COM-M in Section 3.1.1, we assumed that COM-M always remains at the center of the disc. In this section, we investigate the validity of the assumption. We calculated the position error between the actual COM-M with a leg weight and the center of the disc in the direction of x_{wi} during one walking cycle, and found that the error was always less than 15 mm (Fig. B.1), where the stride length s_i was set to the maximum value $2r$ (=120mm). Therefore, for an arbitrary cluster's arbitrary movement, the position error between the actual COM-C and estimated COM-C, that is, ${}^i\boldsymbol{g}$, was always less than 15 mm. To evaluate the error influence, we calculated the stability margin of three types of simulated walking in Fig. 3.13, that is, walking corresponding to Figs. 3.14–3.16, where the stability margin means the minimum distance between the estimated COM-C and edges of the current support polygon.

For all three cluster cases, we found that their minimum stability margins during walking were approximately 15 mm (Fig. B.2), which is equivalent to the maximum position error of COM-C; thus, the actual COM-C was inside the support polygon during walking for the three cases. However, the stability margin for an arbitrary cluster is not larger than the position error of COM-C. For example, the stability margin is very small for a cluster such that the distance between COM-C and one of the lines composed of any one of o_i and $o_j (j \in \{R(i), R(R(i)\})$ is very small. Whereas, the COM-C error can be decreased via sophisticated robot hardware design so that the weight ratio of the leg to the total is decreased.

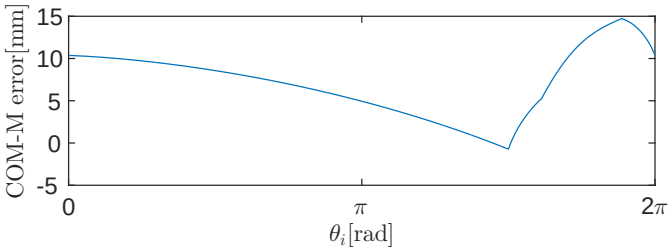


Fig. B.1 Position error between the actual and estimated COM-M in the direction of x_{wi} during one walking cycle. A positive value indicates that the actual COM-M is in the moving direction of the module. Stride length s_i was set to the maximum value $2r$ (=120mm). Stance rate β_i was set to 0.75.

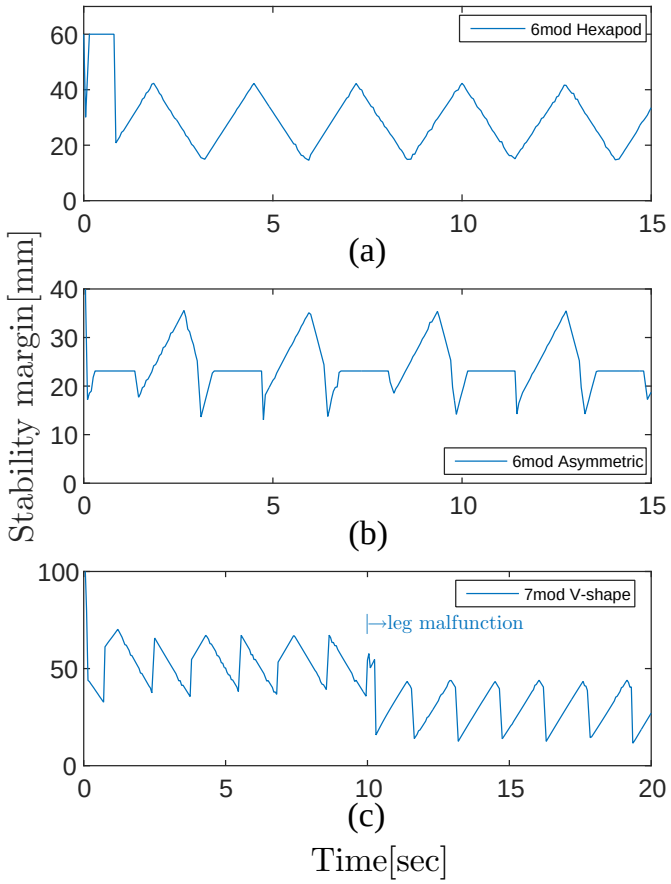


Fig. B.2 Stability margin of three types of walking in Fig. 3.13. The stability margins were calculated for the walkings for which the footprint diagrams Figs. 3.14–3.16 were obtained. (a) Six-legged hexapod cluster's straight walking. (b) Six-legged asymmetric cluster's straight walking. (c) Seven-legged V-shaped cluster's turning walking. One module was broken at $t = 10$ [sec].

Appendix C

Relationship between the conditions (3.7), (3.8), and (C1)

We explain the reason that the conditions (3.7) and (3.8) guarantee the condition (C1). We consider that at any one moment $t = t_0$ when all modules move their legs to satisfy both the conditions (3.7) and (3.8). Let S and F be the set of indices of the modules in the stance and flight phases, respectively. Let $U \subset S \cup F$ and $P_o(U)$ be the convex hull of o_i with $\forall i \in U$. From assumption Eq.(3.1), COM-C is inside $P_o(S \cup F)$. Then, from the condition (3.7) and Lemma 1 (see Appendix C.1), COM-C is inside $P_o\left((S \cup F) \setminus \{j\}\right)$ for any $j \in F$. We inductively see that COM-C is inside $P_o(S)$; that is, COM-C is inside the convex hull of the centers of the modules in the stance phase. From the condition (3.8), $D_{R(i),L(i)}$ and $D_{R(i),i}$ are defined so that the support polygon is always larger than $P_o(S) \setminus \left(D_{R(i),L(i)} \cup D_{R(i),i}\right)$ for any i. Therefore, COM-C is guaranteed to be inside the support polygon.

C.1 Proof of Lemma 1

Lemma 1. *Let $X \subset \mathbb{R}^2$ be a finite set of points in the plane. Denote the convex hull of X by $\mathrm{conv}(X) \subset \mathbb{R}^2$. Consider any points $x \in X$ and $g \in \mathrm{conv}(X)$. Then, the following are equivalent:*

1. *$g \in \mathrm{conv}(X \setminus \{x\})$.*
2. *There exist two points $y, z \in X \setminus \{x\}$ such that line segment yz intersects the half-line from g to x.*
3. *There exist two points $y, z \in X \setminus \{x\}$ such that*

$$\angle xgy + |\angle xgz| < \pi \text{ and } \angle xgy \geq 0 \text{ and } \angle xgz \leq 0.$$

Proof. The equivalence between 2 and 3 is trivial. Thus, we show that 1 and 2 are equivalent.

First, assume 1. The half-line from g to x should cross the boundary of $\mathrm{conv}(X \setminus \{x\})$.

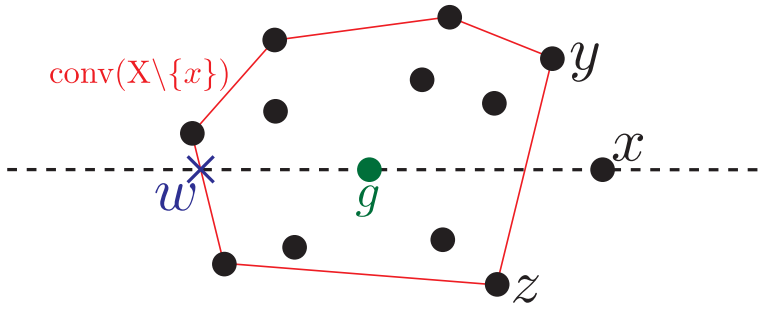


Fig. C.1 Typical example for Lemma 1.

Because the boundary of $\mathrm{conv}(X \setminus \{x\})$ is composed of line segments that connect points in $X \setminus \{x\}$, we have 2. Next, assume 2. The half-line from x to g intersects with the boundary of $\mathrm{conv}(X)$, which we denote by w. By convexity, point w is not on line segment xg. Therefore, point g is in triangle wyz, which is a subset of $\mathrm{conv}(X \setminus \{x\})$. Note that an example of a geometric relationship between x, y, z, w, and g is shown in Fig. C.1. □

Appendix D

s_i^{RL} calculation

In the following, we calculate s_i^{RL} and s_i^R. Because both calculation procedures are similar, we only present the calculation of s_i^{RL}. Let ${}^i\boldsymbol{\Lambda}_j (\forall j \in \{R(i), L(i)\})$ be a vector:

$$ {}^i\boldsymbol{\Lambda}_j = [\cos {}^i\lambda_j \ \ \sin {}^i\lambda_j]^{\mathrm{T}}, \tag{D.1} $$

where

$$ {}^i\lambda_j = \begin{cases} {}^i\psi_j & \left([\cos {}^i\psi_j \ \ \sin {}^i\psi_j]^{\mathrm{T}} \times ({}^i\boldsymbol{g} - {}^i\boldsymbol{p}_j) \geq 0\right) \\ {}^i\psi_j + \pi & \left([\cos {}^i\psi_j \ \ \sin {}^i\psi_j]^{\mathrm{T}} \times ({}^i\boldsymbol{g} - {}^i\boldsymbol{p}_j) < 0\right). \end{cases} \tag{D.2} $$

Examples of the relationship between ${}^i\lambda_j$, COM-C, and x_{wj} are shown in Fig. D.1.

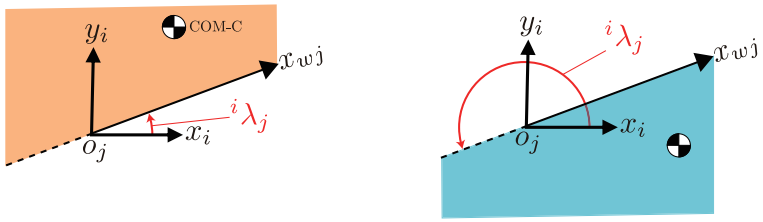
When the geometric scenario in Fig. 3.11(a) holds, three points located at ${}^i\boldsymbol{p}_{R(i)} + \left(\frac{v_{R(i)}^{\mathrm{ideal}}}{2v_i^{\mathrm{ideal}}} s_i^{RL}\right) {}^i\boldsymbol{\Lambda}_{R(i)}$, ${}^i\boldsymbol{g}$, and ${}^i\boldsymbol{p}_{L(i)} - \left(\frac{v_{L(i)}^{\mathrm{ideal}}}{2v_i^{\mathrm{ideal}}} s_i^{RL}\right) {}^i\boldsymbol{\Lambda}_{L(i)}$ or three points located at ${}^i\boldsymbol{p}_{R(i)} - \left(\frac{v_{R(i)}^{\mathrm{ideal}}}{2v_i^{\mathrm{ideal}}} s_i^{RL}\right) {}^i\boldsymbol{\Lambda}_{R(i)}$, ${}^i\boldsymbol{g}$, and ${}^i\boldsymbol{p}_{L(i)} + \left(\frac{v_{L(i)}^{\mathrm{ideal}}}{2v_i^{\mathrm{ideal}}} s_i^{RL}\right) {}^i\boldsymbol{\Lambda}_{L(i)}$ are on one line, in that order. For the two cases, the following holds:

$$ {}^i\boldsymbol{p}_{R(i)} + \frac{v_{R(i)}^{\mathrm{ideal}} s_i^{RL}}{2v_i^{\mathrm{ideal}}} {}^i\boldsymbol{\Lambda}_{R(i)} + k_i^{RL} \left({}^i\boldsymbol{p}_{L(i)} - \frac{v_{L(i)}^{\mathrm{ideal}} s_i^{RL}}{2v_i^{\mathrm{ideal}}} {}^i\boldsymbol{\Lambda}_{L(i)} - \left({}^i\boldsymbol{p}_{R(i)} + \frac{v_{R(i)}^{\mathrm{ideal}} s_i^{RL}}{2v_i^{\mathrm{ideal}}} {}^i\boldsymbol{\Lambda}_{R(i)} \right) \right) = {}^i\boldsymbol{g}, \tag{D.3} $$

where $0 \leq k_i^{RL} \leq 1$. Eq.(D.3) is equivalent to

$$ \begin{aligned} &\frac{v_{R(i)}^{\mathrm{ideal}} s_i^{RL}}{2v_i^{\mathrm{ideal}}} \begin{bmatrix} \cos {}^i\lambda_{R(i)} \\ \sin {}^i\lambda_{R(i)} \end{bmatrix} + \frac{k_i^{RL} s_i^{RL}}{2v_i^{\mathrm{ideal}}} \begin{bmatrix} -v_{L(i)}^{\mathrm{ideal}} \cos {}^i\lambda_{L(i)} - v_{R(i)}^{\mathrm{ideal}} \cos {}^i\lambda_{R(i)} \\ -v_{L(i)}^{\mathrm{ideal}} \sin {}^i\lambda_{L(i)} - v_{R(i)}^{\mathrm{ideal}} \sin {}^i\lambda_{R(i)} \end{bmatrix} \\ &+ k_i^{RL} \begin{bmatrix} {}^ip_{L(i)x} - {}^ip_{R(i)x} \\ {}^ip_{L(i)y} - {}^ip_{R(i)y} \end{bmatrix} = \begin{bmatrix} {}^ig_x - {}^ip_{R(i)x} \\ {}^ig_y - {}^ip_{R(i)y} \end{bmatrix}. \end{aligned} \tag{D.4} $$

By solving the simultaneous equations in Eq.(D.4) for two variables s_i^{RL} and k_i^{RL}, we can derive two solution sets, denoted by $\left\{s_{i,1}^{RL}, k_{i,1}^{RL}\right\}$ and $\left\{s_{i,2}^{RL}, k_{i,2}^{RL}\right\}$. They can explicitly be represented using $\left(v_j^{\mathrm{ideal}}, {}^i\lambda_j, {}^i\boldsymbol{p}_j\right) (\forall j \in \{R(i), L(i)\})$, v_i^{ideal}, and ${}^i\boldsymbol{g}$. Note that only one solution set is derived for the case in which the cluster's desired movement is straight. Using the two



(a) (b)

Fig. D.1 Relationship between $^i\lambda_j$ and the COM-C position. $^i\lambda_j$ depends on whether COM-C is in (a) the orange region or (b) blue region.

solution sets, s_i^{RL} is represented as

$$s_i^{RL} = \begin{cases} |s_{i,1}^{RL}| & \left(BE \neq AF \wedge \left(\left(Bs_{i,1}^{RL} + C \neq 0 \wedge 0 \leq \dfrac{D - As_{i,1}^{RL}}{Bs_{i,1}^{RL} + C} \leq 1\right) \vee \left(Fs_{i,1}^{RL} + G \neq 0 \wedge 0 \leq \dfrac{H - Es_{i,1}^{RL}}{Fs_{i,1}^{RL} + G} \leq 1\right)\right)\right) \\ |s_{i,2}^{RL}| & \left(BE \neq AF \wedge \left(\left(Bs_{i,2}^{RL} + C \neq 0 \wedge 0 \leq \dfrac{D - As_{i,2}^{RL}}{Bs_{i,2}^{RL} + C} \leq 1\right) \vee \left(Fs_{i,2}^{RL} + G \neq 0 \wedge 0 \leq \dfrac{H - Es_{i,2}^{RL}}{Fs_{i,2}^{RL} + G} \leq 1\right)\right)\right) \\ |s_{i,3}^{RL}| & \left(BE = AF \wedge CE + DF \neq AG + BH \wedge \left(\left(Bs_{i,3}^{RL} + C \neq 0 \wedge 0 \leq \dfrac{D - As_{i,3}^{RL}}{Bs_{i,3}^{RL} + C} \leq 1\right) \vee \left(Fs_{i,3}^{RL} + G \neq 0 \wedge 0 \leq \dfrac{H - Es_{i,3}^{RL}}{Fs_{i,3}^{RL} + G} \leq 1\right)\right)\right) \\ undefined & (otherwise), \end{cases} \tag{D.5}$$

where

$$s_{i,1}^{RL} = \frac{-(CE + DF - AG - BH) + \sqrt{(CE + DF - AG - BH)^2 - 4(BE - AF)(DG - CH)}}{2(BE - AF)},$$

$$s_{i,2}^{RL} = \frac{-(CE + DF - AG - BH) - \sqrt{(CE + DF - AG - BH)^2 - 4(BE - AF)(DG - CH)}}{2(BE - AF)},$$

$$s_{i,3}^{RL} = -\frac{DG - CH}{CE + DF - AG - BH},$$

$A = \frac{v_{R(i)}^{\text{ideal}}}{v_i^{\text{ideal}}} \cos {}^i\lambda_{R(i)}$, $B = \frac{-v_{L(i)}^{\text{ideal}} \cos {}^i\lambda_{L(i)} - v_{R(i)}^{\text{ideal}} \cos {}^i\lambda_{R(i)}}{2v_i^{\text{ideal}}}$, $C = {}^ip_{L(i)x} - {}^ip_{R(i)x}$, $D = {}^ig_x - {}^ip_{R(i)x}$,
$E = \frac{v_{R(i)}^{\text{ideal}}}{v_i^{\text{ideal}}} \sin {}^i\lambda_{R(i)}$, $F = \frac{-v_{L(i)}^{\text{ideal}} \sin {}^i\lambda_{L(i)} - v_{R(i)}^{\text{ideal}} \sin {}^i\lambda_{R(i)}}{2v_i^{\text{ideal}}}$, $G = {}^ip_{L(i)y} - {}^ip_{R(i)y}$, $H = {}^ig_y - {}^ip_{R(i)y}$.

Appendix E

Convergence of Eq.(3.31)

In the following, we calculate the terminal phase difference set of Eq.(3.31) and prove that the terminal phase difference is equivalent to Eq.(3.22).

Because its potential function U, that is, $U = \sum_i \cos {}^i\phi_{R(i)}$, only depends on the phase difference, if ω_i is a common value to all modules, then ω_i has no influence on the phase difference. Therefore, the terminal phase difference given by Eq.(3.31) is the same as the terminal phase difference of

$$\theta_i[t+1] = \theta_i[t] - \alpha_2 \frac{\partial}{\partial \theta_i}(\sum_i \cos {}^i\phi_{R(i)}), \tag{E.1}$$

which coincides with the gradient descent method. Therefore, the terminal phase difference is one of the stable (or neutral stable) stationary points of U. First, we calculate all the points in the stationary point set, that is, $\frac{\partial}{\partial \theta_i}(\sum_i \cos {}^i\phi_{R(i)}) = 0$.

From the condition (3.5), graph G is one or more cycle graphs. We focus on one of the cycles composed of $N(\leq n)$ modules. We provide a new index to the modules in the focused graph clockwise, such as mod_1, mod_2, $\cdots$, mod_N, where the index set is denoted by I. When $\frac{\partial}{\partial \theta_i}(\sum_i \cos {}^i\phi_{R(i)}) = 0$, the following holds $\forall i$:

$$\sin(\theta_i[t] - \theta_{R(i)}[t]) = \sin(\theta_{L(i)}[t] - \theta_i[t]) = \sin(\theta_{(i+1)}[t] - \theta_{R((i+1))}[t]), \tag{E.2}$$

which is equivalent to

$${}^i\phi_{R(i)}(\forall i \in I) = \begin{cases} {}^N\phi_{R(N)} \\ \pi - {}^N\phi_{R(N)}. \end{cases} \tag{E.3}$$

Additionally, because G is circular, the following holds:

$$\sum_i {}^i\phi_{R(i)} = 0 \ (\mathrm{mod}\ 2\pi). \tag{E.4}$$

Thus far, we have calculated the stationary point set. In the following, we calculate the stability of all the stationary points above. By substituting Eq.(E.4) into potential function U and deleting ${}^N\phi_{R(N)}$, we obtain

$$U = \cos\Phi_1 + \cos\Phi_2 + \cdots + \cos\Phi_{(N-1)} + \cos(-(\Phi_1 + \Phi_2 + \cdots \Phi_{(N-1)})), \tag{E.5}$$

where each Φ_i is an independent variable. Hereafter, we use Φ_i instead of ${}^i\phi_{R(i)}$ for simplicity. Hessian matrix A, that is, the second partial derivative of U with respect to independent variables $\Phi_i \forall i \in I$, follows

$$
\begin{aligned}
A &= \begin{bmatrix}
\frac{\partial^2 U}{\partial\ \Phi_1^2} & \cdots & \frac{\partial^2 U}{\partial\ \Phi_1 \partial\ \Phi_{(N-1)}} \\
\frac{\partial^2 U}{\partial\ \Phi_2 \partial\ \Phi_1} & \cdots & \frac{\partial^2 U}{\partial\ \Phi_2 \partial\ \Phi_{(N-1)}} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2 U}{\partial\ \Phi_{(N-1)} \partial\ \Phi_1} & \cdots & \frac{\partial^2 U}{\partial\ \Phi_{(N-1)}^2}
\end{bmatrix} \\
&= -\begin{bmatrix}
\cos\Phi_1 + \cos\Phi_N & \cos\Phi_N & \cdots & \cos\Phi_N \\
\cos\Phi_N & \cos\Phi_2 + \cos\Phi_N & \cdots & \cos\Phi_N \\
\vdots & \vdots & \ddots & \vdots \\
\cos\Phi_N & \cos\Phi_N & \cdots & \cos\Phi_{(N-1)} + \cos\Phi_N
\end{bmatrix}. \qquad \text{(E.6)}
\end{aligned}
$$

Generally, if all the eigenvalues λ of A for solution x^* are non-negative, then x^* is a stable or neutral stable solution, which is equivalent to one of the terminal values.

Then, we calculate λ. First, we calculate λ for the top line of Eq.(E.3), that is, $\Phi_i(\forall i \in I) = \Phi_N$. The characteristic polynomial of A can be written as

$$(N\cos\Phi_N + \lambda)(\cos\Phi_N + \lambda)^{N-2} = 0. \qquad \text{(E.7)}$$

When $\cos\Phi_i \leq 0 \forall i$, all eigenvalues are non-negative; that is, if $\pi/2 \leq \Phi_i \leq 3\pi/2 \forall i$ holds, then all eigenvalues are non-negative. Thus, phase differences converge to the solution.

Next, we calculate λ for the bottom line of Eq.(E.3), that is, $\Phi_i(\exists i \in I) = \pi - \Phi_N$. The characteristic polynomial of A can be written as

$$(\cos\Phi_N + \lambda)^{N-k-2}(-\cos\Phi_N + \lambda)^{k-1}\left(\lambda^2 + \cos\Phi_N(N-1)\lambda + \cos^2\Phi_N(2k-N)\right) = 0, \qquad \text{(E.8)}$$

where $k(1 \leq k \leq N-1)$ is the number of $i|\Phi_i = \pi - \Phi_N$.

When $k = 1$, Eq.(E.8) is rewritten as following:

$$(\cos\Phi_N + \lambda)^{N-3}(\lambda^2 + (N-1)\cos\Phi_N\lambda + (2-N)\cos^2\Phi_N) = 0. \qquad \text{(E.9)}$$

From Eq.(E.9) and $N \geq 3$, positive and negative eigenvalues are always confused.

When $2 \leq k \leq N-3$, From Eq.(E.8), positive and negative eigenvalues are always confused.

When $k = N-2$, Eq.(E.8) is rewritten as following:

$$(-\cos\Phi_N + \lambda)^{N-3}(\lambda^2 + (N-1)\cos\Phi_N\lambda + (N-4)\cos^2\Phi_N) = 0. \qquad \text{(E.10)}$$

From Eq.(E.10), positive and negative eigenvalues are always confused.

When $k = N-1$, Eq.(E.8) is rewritten as following:

$$(-\cos\Phi_N + \lambda)^{N-2}(\lambda + (N-2)\cos\Phi_N) = 0. \qquad \text{(E.11)}$$

From Eq.(E.11) and $N \geq 3$, positive and negative eigenvalues are always confused.

From these discussions, when $\Phi_i(\exists i \in I) = \pi - \Phi_N$, positive and negative eigenvalues are always confused. Thus, the solution is unstable.

For the case in which G is more than one cycle, then the above discussion can be applied to every other cycle. Therefore, it has been proved that the terminal phase difference is ${}^{i}\phi_{R(i)} = {}^{R(i)}\phi_{R(R(i))}$ and $\pi/2 \leq {}^{i}\phi_{R(i)} \leq 3\pi/2 \forall i$, which is equivalent to Eq.(3.22).

Bibliography

[1] L. Bayindir, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[2] E. Castello, T. Yamamoto, Y. Nakamura, and H. Ishiguro, "Task allocation for a robotic swarm based on an adaptive response threshold model," *International Conference on Control, Automation and Systems*, pp. 259–266, 2013.

[3] E. Castello, T. Yamamoto, F. Dalla, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach," *Swarm Intelligence*, vol. 10, pp. 1–31, 2016.

[4] J. J. Collins and I. N. Stewart, "Coupled nonlinear oscillators and the symmetries of animal gaits," *Journal of Nonlinear Science*, vol. 3, pp. 349–392, 1993.

[5] T. Odashima, Z. Luo, and S. Hosoe, "Hierarchical control structure of a multilegged robot for environmental adaptive locomotion," *Artificial Life and Robotics*, vol. 6, pp. 44–51, 2002.

[6] S. Inagaki, H. Yuasa, and T. Arai, "CPG model for autonomous decentralized multi-legged robot system - generation and transition of oscillation patterns and dynamics of oscillators," *Robotics and Autonomous Systems*, vol. 44, pp. 171–179, 2003.

[7] S. Fujiki, S. Aoi, T. Funato, N. Tomita, K. Senda, and K. Tsuchiya, "Hysteresis in the metachronal-tripod gait transition of insects: A modeling study," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 88, no. 1, 2013.

[8] K. Tsuchiya, S. Aoi, and K. Tsujita, "Locomotion control of a Multi-legged locomotion robot using oscillators," in *IEEE International Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, Tunisia, 2002.

[9] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, pp. 97–120, 2008.

[10] Y. Yang, C. Zhou, and Y. Tian, "Swarm robots task allocation based on response threshold model," in *2009 4th International Conference on Autonomous Robots and Agents*, 2009, pp. 171–176.

[11] R. O'Grady, R. Groß, A. L. Christensen, and M. Dorigo, "Self-assembly strategies in a group of autonomous mobile robots," *Autonomous Robots*, vol. 28, no. 4, pp. 439–455, 2010.

[12] V. P. Veetil, "Coordination in centralized and decentralized systems," *International Journal of Microsimulation*, vol. 10, no. 2, pp. 86–102, 2017.

[13] S. Ito, M. Sasaki, Y. Fujita, and H. Yuasa, "A circularly coupled oscillator system for relative phase regulation and its application to timing control of a multicylinder engine," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 10, pp. 3100–3112, 2010.

[14] J. Rogge and D. Aeyels, "Stability of phase locking in a ring of unidirectionally coupled oscillators," *Journal of Physics A: Mathematical and General*, vol. 37, no. 46, pp. 11 135–11 148, 2004.

[15] D. Sakai, H. Fukushima, and F. Matsuno, "Leader - follower navigation in obstacle environments while preserving connectivity without data transmission," *IEEE Transactions on control systems technology*, vol. 26, no. 4, pp. 1233–1248, 2018.

[16] C. H. Yu and R. Nagpal, "A self-adaptive framework for modular robots in a dynamic environment: Theory and applications," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 1015–1036, 2011.

[17] E. O. Wilson, "The relation between caste ratios and division of labor in the ant genus *Pheidole* (Hymenoptera: Formicidae)," *Behavioral Ecology and Sociobiology*, vol. 16, pp. 89–98, 1984.

[18] E. Hasegawa, Y. Ishii, K. Tada, K. Kobayashi, and J. Yoshimura, "Lazy workers are necessary for long-term sustainability in insect societies," *Scientific Reports*, vol. 6, no. 20846, pp. 1–9, 2016.

[19] E. Bonabeau, G. Theraulaz, and J. L. Deneubourg, "Fixed response thresholds and the regulation of division of labor in insect societies," *Bulletin of Mathematical Biology*, vol. 60, pp. 753–807, 1998.

[20] W. Lee and D. E. Kim, "Adaptive approach to regulate task distribution in swarm robotic systems," *Swarm and Evolutionary Computation*, vol. 44, pp. 1108–1118, 2019.

[21] A. Lioni, C. Sauwens, G. Theraulaz, and J.-L. Deneubourg, "Chain formation in *Œcophylla longinoda*," *Journal of Insect Behavior*, vol. 14, no. 5, pp. 679–696, 2001.

[22] C. Anderson, G. Theraulaz, and J. L. Deneubourg, "Self-assemblages in insect societies," *Insectes Sociaux*, vol. 49, no. 2, pp. 99–110, 2002.

[23] C. R. Reid, M. J. Lutz, S. Powell, A. B. Kao, I. D. Couzin, and S. Garnier, "Army ants dynamically adjust living bridges in response to a cost-benefit trade-off," in *Proceedings of the National Academy of Sciences*, vol. 112, no. 49, 2015, pp. 15 113–15 118.

[24] A. Kamimura, H. Kurokawa, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji, "Automatic locomotion pattern generation for modular robots," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 714–720.

[25] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-l. Deneubourg, and M. Dorigo, "The cooperation of swarm-bots," *IEEE Robotics and Automation Magazine*, vol. 12, no. 2, pp. 21–28, 2005.

[26] H. Wei, Y. Chen, J. Tan, and T. Wang, "Sambot: A self-assembly modular robot system," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 745–757, 2011.

[27] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A. J. Ijspeert, "Roombots : A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 1016–1033, 2014.

[28] G. Jing, T. Tosun, M. Yim, and H. Kress-Gazit, "An end-to-end system for accomplishing tasks with modular robots: Perspectives for the AI community," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4879–4883.

[29] T. Graham Brown, "The intrinsic factors in the act of progression in the mammal," *Proceedings of the Royal Society B, Containing Papers of a Biological Character*, vol. 84, no. 572, pp. 308–319, 1911.

[30] K. Pearson, "The control of walking," *Scientific American*, vol. 235, no. 6, pp. 72–86, 1976.

[31] Y. Kuramoto, "Cooperative dynamics of oscillator community," *Progress of Theoretical Physics Supplement*, vol. 79, pp. 223–240, 1984.

[32] F. Dörfler and F. Bullo, "Synchronization in complex networks of phase oscillators: A survey," *Automatica*, vol. 50, no. 6, pp. 1539–1564, 2014.

[33] D. Owaki, T. Kano, K. Nagasawa, A. Tero, and A. Ishiguro, "Simple robot suggests physical interlimb communication is essential for quadruped walking," *Journal of the Royal Society Interface*, vol. 10, no. 78, 2013.

[34] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[35] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[36] J. P. Sluijter, V. Verhage, J. C. Deddens, F. Van Den Akker, and P. A. Doevendans, "Microvesicles and exosomes for intracardiac communication," *Cardiovascular Research*, vol. 102, no. 2, pp. 302–311, 2014.

[37] F. M. Burnet, "A modification of jerne's theory of antibody production using the concept of clonal selection," *CA: A Cancer Journal for Clinicians*, vol. 26, no. 2, pp. 119–121, 1976.

[38] J. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1–3, pp. 187–204, 1986.

[39] D. W. Lee and K. B. Sim, "Artificial immune network-based cooperative control in collective autonomous mobile robots," in *Proceedings 6th IEEE International Workshop on Robot and Human Communication. RO-MAN'97 SENDAI*, 1997, pp. 58–63.

[40] G. C. Luh and W. W. Liu, "An immunological approach to mobile robot reactive navi-

gation," *Applied Soft Computing*, vol. 8, no. 1, pp. 30–45, 2008.

[41] C. R. Ribbands, "Division of labour in the honeybee community," *Proceedings of the Royal Society B*, vol. 140, no. 898, pp. 32–43, 1952.

[42] W. M. Wheeler, "A study of some ant larvæ, with a consideration of the origin and meaning of the social habit among insects," *Proceedings of the American Philosophical Society*, vol. 57, no. 4, pp. 293–343, 1918.

[43] G. M. Whitesides and B. Grzybowski, "Self-assembly at all scales," *Science*, vol. 295, no. 5564, pp. 2418–2421, 2002.

[44] N. R. Franks, A. B. Sendova-Franks, and C. Anderson, "Division of labour within teams of new world and old world army ants," *Animal Behaviour*, vol. 62, no. 4, pp. 635–642, 2001.

[45] C. Peeters and S. De Greef, "Predation on large millipedes and self-assembling chains in Leptogenys ants from Cambodia," *Insectes Sociaux*, vol. 62, no. 4, pp. 471–477, 2015.

[46] D. M. Gordon, *Ants at work*. The Free Press, A Division of Simon & Schuster Inc., 1999.

[47] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.

[48] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Teamwork in self-organized robot colonies," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 695–711, 2009.

[49] R. Fujisawa, S. Dobata, K. Sugawara, and F. Matsuno, "Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance," *Swarm Intelligence*, vol. 8, no. 3, pp. 227–246, 2014.

[50] M. J. B. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," *Robotics and Autonomous Systems*, vol. 30, pp. 65–84, 2000.

[51] T. H. Labella and M. Dorigo, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, no. 1, pp. 4–25, 2006.

[52] A. Brutschy, G. Pini, C. Pinciroli, M. Birattari, and M. Dorigo, "Self-organized task allocation to sequentially interdependent tasks in swarm robotics," *Autonomous Agents and Multi-Agent Systems*, vol. 28, no. 1, pp. 101–125, 2014.

[53] D. M. Gordon, R. Paul, and K. Thorpe, "What is the function of encounter patterns in ant colonies?" *Animal Behaviour*, vol. 42, no. 6, pp. 1083–1100, 1993.

[54] D. J. Schulz, M. J. Vermiglio, Z. Y. Huang, and G. E. Robinson, "Effects of colony food shortage on social interactions in honey bee colonies," *Insectes Sociaux*, vol. 49, pp. 50–55, 2002.

[55] A. Mailleux, J. Deneubourg, and C. Detrain, "How does colony growth influence communication in ants ?" *Insectes Sociaux*, vol. 50, pp. 24–31, 2003.

[56] M. L. Thomas and M. A. Elgar, "Colony size affects division of labour in the ponerine ant *Rhytidoponera metallica*." *Naturwissenschaften*, vol. 90, no. 2, pp. 88–92, 2003.

[57] O. Rueppell and R. W. Kirkman, "Extraordinary starvation resistance in Temnothorax rugatulus (*Hymenoptera, Formicidae*) colonies: demography and adaptive behavior," *Insectes Sociaux*, vol. 52, no. 3, pp. 282–290, 2008.

[58] A. C. Mailleux, C. Devigne, J. L. Deneubourg, and C. Detrain, "Impact of starvation on *Lasius niger*' exploration," *Ethology*, vol. 116, no. 3, pp. 248–256, 2010.

[59] K. Katz and D. Naug, "Energetic state regulates the exploration - exploitation trade-off in honeybees," *Behavioral Ecology*, vol. 26, no. 4, pp. 1045–1050, 2015.

[60] I. Karsai and A. Runciman, "The 'common stomach' as information source for the regulation of construction behaviour of the swarm," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 18, no. 1, pp. 13–24, 2012.

[61] T. Schmickl and K. Crailsheim, "TaskSelSim: A model of the self-organization of the division of labour in honeybees," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 14, no. 2, pp. 101–125, 2008.

[62] S. W. Pacala, D. M. Gordon, and H. C. Godfray, "Effects of social group size on information transfer and task allocation," *Evolutionary Ecology*, vol. 10, no. 2, pp. 127–165, 1996.

[63] H. Pereira and D. Gordon, "A trade-off in task allocation between sensitivity to the environment and response time," *Journal of Theoretical Biology*, vol. 208, no. 2, pp. 165–184, 2001.

[64] T. Schmickl and I. Karsai, "Sting, carry and stock: how corpse availability can regulate de-centralized task allocation in a ponerine ant colony," *PLOS ONE*, vol. 9, no. 12, pp. 1–33, 2014.

[65] T. Schmickl and I. Karsai, "How regulation based on a common stomach leads to economic optimization of honeybee foraging," *Journal of Theoretical Biology*, vol. 389, pp. 274–286, 2016.

[66] D. M. Gordon, "The dynamics of the daily round of the harvester ant colony (*Pogonomyrmex badius*)," *Animal Behaviour*, vol. 34, no. 5, pp. 1402–1419, 1986.

[67] G. E. Robinson, "Regulation of division of labor in insect societies," *Annual Review of Entomology*, vol. 37, pp. 637–665, 1992.

[68] D. M. Gordon, "The harvester ant (*Pogonomyrmex badius*) midden: refuse or boundary?" *Ecological Entomology*, vol. 9, no. 4, pp. 403–412, 1984.

[69] D. M. Gordon, "Dynamics of task switching in harvester ants," *Animal Behaviour*, vol. 38, no. 2, pp. 194–204, 1989.

[70] D. M. Gordon, "Group-level dynamics in harvester ants: young colonies and the role of patrolling," *Animal Behaviour*, vol. 35, no. 3, pp. 833–843, 1987.

[71] D. M. Gordon and N. J. Mehdiabadi, "Encounter rate and task allocation in harvester ants," *Behavioral Ecology and Sociobiology*, vol. 45, no. 5, pp. 370–377, 1999.

[72] M. J. Greene and D. M. Gordon, "Interaction rate informs harvester ant task decisions," *Behavioral Ecology*, vol. 18, no. 2, pp. 451–455, 2007.

[73] B. Prabhakar, K. N. Dektar, and D. M. Gordon, "The regulation of ant colony foraging activity without spatial information," *PLOS Computational Biology*, vol. 8, no. 8, 2012.

[74] N. Pinter-Wollman, A. Bala, A. Merrell, J. Queirolo, M. C. Stumpe, S. Holmes, and D. M. Gordon, "Harvester ants use interactions to regulate forager activation and availability," *Animal Behaviour*, vol. 86, no. 1, pp. 197–207, 2013.

[75] J. D. Davidson, R. P. Arauco-Aliaga, S. Crow, D. M. Gordon, and M. S. Goldman, "Effect of interactions between harvester ants on forager decisions," *Frontiers in Ecology and Evolution*, vol. 4, pp. 1–17, 2016.

[76] D. Wagner, M. J. F. Brown, P. Broun, W. Cuevas, L. E. Moses, D. L. Chao, and D. M. Gordon, "Task-related differences in the cuticular hydrocarbon composition of harvester ants, *Pogonomyrmex barbatus*," *Journal of Chemical Ecology*, vol. 24, no. 12, pp. 2021–2037, 1998.

[77] B. D. Beverly, H. McLendon, S. Nacu, S. Holmes, and D. M. Gordon, "How site fidelity leads to individual differences in the foraging activity of harvester ants," *Behavioral Ecology*, vol. 20, no. 3, pp. 633–638, 2009.

[78] D. M. Gordon and B. Hölldobler, "Worker longevity in harvester ants(*Pogonomyrmex*)," *Psyche*, vol. 94, pp. 341–346, 1987.

[79] J. R. B. Lighton and G. A. Bartholomew, "Standard energy metabolism of a desert harvester ant, *Pogonomyrmex rugosus*: Effects of temperature, body mass, group size, and humidity," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 85, no. 13, pp. 4765–4769, 1988.

[80] J. B. V. Weir, "New methods for calculating metabolic rate with special reference to protein metabolism," *The Journal of Physiology*, vol. 109, no. 1-2, pp. 1–9, 1949.

[81] R. Brodschneider and K. Crailsheim, "Nutrition and health in honey bees," *Apidologie*, vol. 41, no. 3, pp. 278–294, 2010.

[82] C. L. Kwapich and W. R. Tschinkel, "Demography, demand, death, and the seasonal allocation of labor in the Florida harvester ant (*Pogonomyrmex badius*)," *Behavioral Ecology and Sociobiology*, vol. 67, no. 12, pp. 2011–2027, 2013.

[83] N. Pinter-Wollman, R. Wollman, A. Guetz, S. Holmes, and D. M. Gordon, "The effect of individual variation on the structure and function of interaction networks in harvester ants," *Journal of the Royal Society Interface*, vol. 8, no. 64, pp. 1562–1573, 2011.

[84] A. Lara, *Ecology and evolution of offspring size in Pogonomyrmex colony founding*. University of Houston, Ph.D. thesis, 2013.

[85] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

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

[87] M. Buehler and D. E. Koditschek, "RHex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[88] D. Koh, J. Yang, and S. Kim, "Centipede robot for uneven terrain exploration: Design and experiment of the flexible biomimetic robot mechanism," in *3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2010*, Tokyo, Japan, 2010, pp. 877–881.

[89] S. Aoi, P. Manoonpong, Y. Ambe, F. Matsuno, and F. Wörgötter, "Adaptive control strategies for interlimb coordination in legged robots: A review," *Frontiers in Neurorobotics*, vol. 11, no. 39, pp. 1–21, 2017.

[90] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological Cybernetics*, vol. 65, no. 3, pp. 147–159, 1991.

[91] S. Fujiki, S. Aoi, T. Funato, N. Tomita, K. Senda, and K. Tsuchiya, "Adaptation mechanism of interlimb coordination in human split-belt treadmill walking through learning of foot contact timing: A robotics study," *Journal of the Royal Society Interface*, vol. 12, no. 110, 2015.

[92] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1421, 2007.

[93] Y. Fukuoka, Y. Habu, and T. Fukui, "Analysis of the gait generation principle by a simulated quadruped model with a CPG incorporating vestibular modulation." *Biological Cybernetics*, vol. 107, no. 6, pp. 695–710, 2013.

[94] Y. Ambe, S. Aoi, T. Nachstedt, P. Manoonpong, F. Wörgötter, and F. Matsuno, "Simple analytical model reveals the functional role of embodied sensorimotor interaction in hexapod gaits," *PLOS ONE*, vol. 13, no. 2, pp. 1–28, 2018.

[95] K. Yasui, K. Sakai, T. Kano, D. Owaki, and A. Ishiguro, "Decentralized control scheme for myriapod robot inspired by adaptive and resilient centipede locomotion," *PLOS ONE*, vol. 12, no. 2, pp. 1–12, 2017.

[96] U. Bässler and A. Büschges, "Pattern generation for stick insect walking movements - Multisensory control of a locomotor program," *Brain Research Reviews*, vol. 27, no. 1, pp. 65–88, 1998.

[97] J. Kim, A. Alspach, and K. Yamane, "Snapbot: A reconfigurable legged robot," in *IEEE International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, 2017, pp. 5861–5867.

[98] S. Ha, J. Kim, and K. Yamane, "Automated deep reinforcement learning environment for hardware of a modular legged robot," in *15th International Conference on Ubiquitous Robots (UR)*, Honolulu, HI, USA, 2018, pp. 348–354.

[99] J. Whitman, S. Su, S. Coros, A. Ansari, and H. Choset, "Generating gaits for simultaneous locomotion and manipulation," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 2723–2729.

[100] S. Koos, A. Cully, and J. B. Mouret, "Fast damage recovery in robotics with the T-resilience algorithm," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1700–1723, 2013.

[101] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[102] G. Ren, W. Chen, S. Dasgupta, C. Kolodziejski, F. Wörgötter, and P. Manoonpong, "Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation," *Information Sciences*, vol. 294, pp. 666–682, 2015.

[103] A. M. Johnson, G. C. Haynes, and D. E. Koditschek, "Disturbance detection, identification, and recovery by gait transition in legged robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 5347–5353.

[104] M. Schilling, H. Cruse, and P. Arena, "Hexapod walking: An expansion to Walknet dealing with leg amputations and force oscillations," *Biological Cybernetics*, vol. 96, no. 3, pp. 323–340, 2007.

[105] A. E. S. Auf, M. Litza, and E. Maehle, "Distributed fault-tolerant robot control architecture based on organic computing principles," in *International Conference on Biologically-Inspired Collaborative Computing*, 2008, pp. 115–124.

[106] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *Science*, vol. 318, no. 5853, pp. 1088–1093, 2007.

[107] B. T. Kirby, B. Aksak, J. D. Campbell, J. F. Hoburg, T. C. Mowry, P. Pillai, and S. C. Goldstein, "A modular robotic system using magnetic force effectors," in *IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 2787–2793.

[108] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed self-reconfiguration of M-TRAN III modular robotic system," *The International Journal of Robotics Research*, vol. 27, no. 3–4, pp. 373–386, 2008.

[109] J. W. Romanishin, K. Gilpin, S. Claici, and D. Rus, "3D M-Blocks : self-reconfiguring robots capable of locomotion via pivoting in three dimensions," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, pp. 1925–1932.

[110] T. Takubo, T. Arai, K. Inoue, H. Ochi, T. Konishi, T. Tsurutani, Y. Hayashibara, and E. Koyanagi, "Integrated limb mechanism robot ASTERISK," *Journal of Robotics and Mechatronics*, vol. 18, no. 2, pp. 203–214, 2006.

[111] M. Ohira, R. Chatterjee, T. Kamegawa, and F. Matsuno, "Development of three-legged modular robots and demonstration of collaborative task execution," in *Proceedings - IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007, pp.

3895–3900.

[112] V. G. Loc, I. M. Koo, D. T. Tran, S. Park, H. Moon, and H. R. Choi, "Body workspace of quadruped walking robot and its applicability in legged locomotion," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 67, no. 3-4, pp. 271–284, 2012.

[113] B. M. Nejad, S. A. Attia, and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of switching communication topologies," in *XXII International Symposium on Information, Communication and Automation Technologies (ICAT)*, Sarajevo, Bosnia and Herzegovina, 2009, pp. 198–204.

[114] P. Monzón and F. Paganini, "Global considerations on the Kuramoto model of sinusoidally coupled oscillators," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, 2005, pp. 3923–3928.

[115] J. E. A. Bertram and A. Gutmann, "Motions of the running horse and cheetah revisited : Fundamental mechanics of the transverse and rotary gallop," *Journal of the Royal Society Interface*, vol. 6, no. 35, pp. 1–11, 2008.

[116] J. Nishii, "An analytical estimation of the energy cost for legged locomotion," *Journal of Theoretical Biology*, vol. 238, pp. 636–645, 2006.

[117] S. Kuroda, I. Kunita, Y. Tanaka, A. Ishiguro, R. Kobayashi, and T. Nakagaki, "Common mechanics of mode switching in locomotion of limbless and legged animals," *Journal of the Royal Society Interface*, vol. 11, no. 95, 2014.

[118] N. J. Mlot, C. A. Tovey, and D. L. Hu, "Fire ants self-assemble into waterproof rafts to survive floods," *Proceedings of the National Academy of Sciences*, vol. 108, no. 19, pp. 7669–7673, 2011.

[119] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[120] A. Kamimura, H. Kurokawa, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, "Distributed adaptive locomotion by a modular robotic system, M-TRAN II," in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2370–2377.

[121] H. Li, T. Wang, H. Wei, and C. Meng, "Response strategy to environmental cues for modular robots with self-assembly from swarm to articulated robots," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 81, no. 3-4, pp. 359–376, 2016.

[122] G. Jing, T. Tosun, M. Yim, and H. Kress-Gazit, "Accomplishing high-level tasks with modular robots," *Autonomous Robots*, vol. 42, pp. 1337–1354, 2018.

[123] J. Daudelin, G. Jing, T. Tosun, M. Yim, H. Kress-Gazit, and M. Campbell, "An integrated system for perception-driven autonomy with modular robots," *Science Robotics*, vol. 3, no. 23, 2018.

[124] H. Khodr, M. Mutlu, S. Hauser, A. Bernardino, and A. Ijspeert, "An optimal planning framework to deploy self-reconfigurable modular robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4278–4285, 2019.

[125] J. Seo, J. Paik, and M. Yim, "Modular reconfigurable robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 63–88, 2019.

[126] A. Roennau, G. Heppner, M. Nowicki, J. M. Zoellner, and R. Dillmann, "Reactive posture behaviors for stable legged locomotion over steep inclines and large obstacles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4888–4894.

[127] Y. Tian, F. Gao, J. M. Liu, and X. B. Chen, "Step rolling planning of a six-legged robot with 1-DOF waist for slope climbing," *Science China Technological Sciences*, vol. 62, pp. 597–607, 2019.

[128] K. S. Espenschied, R. D. Quinn, R. D. Beer, and H. J. Chiel, "Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot," *Robotics and Autonomous Systems*, vol. 18, no. 1-2, pp. 59–64, 1996.

[129] U. Asif and J. Iqbal, "An approach to stable walking over uneven terrain using a reflex-based adaptive gait," *Journal of Control Science and Engineering*, pp. 1–12, 2011.

Publication List

Journal papers Related to The Dissertation

[1] T. Hayakawa, S. Dobata, and F. Matsuno, "Behavioral responses to colony-level properties affect disturbance resistance of red harvester ant colonies," *Journal of Theoretical Biology*, vol.492: 110186, 2020, https://doi.org/10.1016/j.jtbi.2020.110186.

[2] T. Hayakawa, T. Kamimura, S. Kaji, and F. Matsuno, "Autonomous distributed system for gait generation for single-legged modular robots connected in various configurations," *IEEE Transactions on Robotics*, https://doi.org/10.1109/TRO.2020.2992983(in press).

[3] T. Hayakawa and F. Matsuno, "Autonomous distributed system for single-legged modular robots to traverse environments by adaptive reconfiguration," *Autonomous Robots*, (submitted).

International conference papers Related to The Dissertation

[1] T. Hayakawa, Y. Ogawa, S. Dobata, K. Tsuji, and F. Matsuno, "Possibility of adaptive behavior of ants based on global information," *SWARM 2015: The First International Symposium on Swarm Behavior and Bio-Inspired Robotics*, Kyoto, Japan, October 28-30, 2015, pp.166-167.

[2] T. Hayakawa, Y. Ogawa, Y. Tsujimoto, S. Dobata, K. Tsuji, and F. Matsuno, "Relationship between robustness of ant colonies and behaviour change based on global information," *SWARM 2017: The 2nd International Symposium on Swarm Behavior and Bio-Inspired Robotics*, Kyoto, Japan, October 29-Novrmber 1, 2017, pp.373-374.

[3] T. Hayakawa, and F. Matsuno, "Self-organized shape-optimizing strategy for single-legged modular robots to traverse unknown gap environment," *SWARM 2019: The 3rd International Symposium on Swarm Behavior and Bio-Inspired Robotics*, Okinawa, Japan, November 20-22, 2019, pp.157-160.

Domestic conference papers Related to The Dissertation

[1] T. Hayakawa, S. Kaji, and F. Matsuno, "Autonomous distributed system for gait generation of single-leg modular robots connecting into various leg configuration," *SCI'18*, Kyoto, 2018 (in Japanese).

[2] T. Hayakawa, S. Kaji, and F. Matsuno, "Development of autonomous distributed system for gait generation of one-leg modular robot connecting into various leg configuration and its experimental verification," *SI2018*, Osaka, 2018 (in Japanese).

[3] T. Hayakawa, and F. Matsuno, "Autonomous distributed system for single-leg modular robot realizing slope-type based optimal leg configuration," *SCI'19*, Osaka, 2019 (in Japanese).

Journal papers Unrelated to The Dissertation

[1] D. Fujiwara, K. Yamamoto, R. Ariizumi, T. Hayakawa, and F. Matsuno, "Update Method of Cost Function to Learn Robust Policy Parameters," *Transactions of the Institute of Systems, Control and Information Engineers*, vol.33, no.6, pp.191-200, 2020(in Japanese).

[2] K. Yamamoto, R. Ariizumi, T. Hayakawa, and F. Matsuno, "Path Integral Policy Improvement with Population Adaptation," *IEEE Transactions on Cybernetics*, https://doi.org/10.1109/TCYB.2020.2983923(in press).

Copyright