Doctor of Philosophy

Multi-scopic Neuro-Cognitive Adaptation for Legged Locomotion Robots

2021, March

Department of Mechanical Systems Engineering
Graduate School of Systems Design
Tokyo Metropolitan University

Azhar Aulia Saputra

Doctor of Philosophy

Multi-scopic Neuro-Cognitive Adaptation for Legged Locomotion Robots

2021, March

Department of Mechanical Systems Engineering
Graduate School of Systems Design
Tokyo Metropolitan University

18962605

Azhar Aulia Saputra

Professor Naoyuki Kubota

Abstract

Recently, various types of robots have been gradually replacing human workers in the tasks of exploring unknown, dangerous and dynamic environments. Especially, the mobility in dynamic environments is strongly required of robots. The human dynamic locomotion is realized by the simultaneous integration based on adaptability and optimality. The adaptability depends on the real-time perception as a bottom-up learning-based approach from the microscopic point of view, while the optimality depends on cognition as a top-down knowledge-based approach from the macroscopic point of view. Various cognitive architectures have been proposed in the cognitive science until now, but it is difficult for robotic researchers to apply such cognitive models to deal with adaptability and optimality simultaneously in real dynamic environments, because most of them only explain the information flow of human cognitive behaviors in the conceptual level, and the methodology to implement functions like human cognitive behaviors is not explained in detail. Therefore, we need a cognitive model that can cover and integrate the mobility from microscopic control to macro-scopic planning and from short-term adaptation to long-term optimization.

This thesis proposes a neuro-cognitive model for multi-legged locomotion to realize the seamless integration from multi-modal sensing, ecological perception and cognition through the coordination of interoceptive and exteroceptive sensory information. A cognitive model can be discussed from three different scopes: micro-, meso-, and macro-scopic, corresponding to sensing, perception, and cognition and short-, medium-, and long-term of adaptation related with neuro science and ecological psychology. Basically, the multi-legged locomotion requires the intelligent functions of 1) attention module, 2) adaptive locomotion control module, 3) Object recognition module, 4) environmental map building module, and 5) optimal motion planning module, and the proposed neuro-cognitive model integrates the above intelligent functions from the multi-scopic point of view.

At the microscopic level, we build an attention mechanism for exteroceptive sensory information according to the current interoceptive sensory information, and adaptive locomotion control is done by (the lower-level of) sensorimotor coordination based on interoceptive and exteroceptive sensory information as a short-term adaptation. Furthermore, online locomotion generator is also processed in this scope. I proposed the concept of sensory-motor coordination model based on perceiving-acting cycle in the microscopic level, that is a lower level control system interacting directly with the environment. The sensory-motor coordination model is composed of 1) attention mechanism module that controls the topological structure of 3D point cloud information by using Dynamic Density Growing Neural Gas (DD-GNG). DD-GNG can control the density of topological structures in specific ar-ea based on the attention, 2) object affordance detection module for the direct perception to generally identify the environmental condition based on the physical embodiment, 3) neural based locomotion module that generates dynamic gait patterns by integrating with sensorimotor coordination. The proposed model has been optimized through a learning process in the computer simulation beforehand. The locomotion ability of real multi-legged robot and the leg malfunction tests are demonstrated in the experiments by several different terrains. The experimental results show that a smooth gait-pattern transition could be generated during sudden leg malfunction.

At the macroscopic level, we build environmental knowledge with higher level of behavior planning from (the collection or memory of) large scale of sensory information. The robot can conduct optimal motion planning using the built environmental knowledge. I proposed the method of building environmental knowledge by us-ing the topological structure-based map reconstruction. Next, we proposed an optimal path planning method on the built map. Experimental results show that the proposed method can extract environmental features for multi-legged robots and build envi-ronmental knowledge for optimal path planning.

At the mesoscopic level, the proposed neuro-cognitive model integrates these two approaches of microscopic and macroscopic levels. Especially, the proposed neuro-cognitive model builds a cognitive map using the bottom-up facial environmental information and top-down map information, and generates intention towards the final goal from the macroscopic level. I proposed a neuro-cognitive model integrates the sensory-motor coordination model with environmental knowledge as a cognitive map using the bottom-up facial environmental information and top-down map information. The cognitive map is used to recognize a current facing situation by the topological structure information from lower level, composed as 3D vector position of nodes, edges, and 3D surface vectors of nodes. The robot generates intention towards the final goal from the macroscopic level. Furthermore, the learning process to integrate the relationship between macroscopic level of behavior commands and locomotion performance is developed as well. The proposed model has been tested for omnidirectional movement in biped and quadruped robot. Furthermore, the proposed neuro-cognitive model has also been implemented for robot climbing behavior, performing a horizontal-vertical-horizontal movement.

The effectiveness of the proposed model has been evaluated through a series of experiments. First, I conduct experiments on locomotion learning and control based on sensory-motor coordination in the microscopic level. Next, I conduct experiments on environmental knowledge building and global path planning. Finally, I conduct experiments on the real-time re-planning and behavior coordination in the rough terrain and dynamic environments in the mesoscopic level. Through the above experiments, I showed the adaptability and optimality in the multi-legged locomotion by the proposed multi-scale neuro-cognitive model. Future research directions not only implemented for robotics application but also can be implemented for the interdisciplinary study on cognitive science and ecological psychology.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, various types of robots have been gradually replacing human works in the tasks of exploring unknown, dangerous and dynamic environments. A robot should have a structure suitable for a purpose. In general, mobile robots can be classified into legged robots and wheeled robots. From a broader perspective, legged robots are more versatile than wheeled robots simply because there are less than half of the worlds terrain that can be accessed on wheels. Especially, the mobility, the ability to move, in dynamic environments is strongly required of robots, but the mobility in dynamic environments is one of the most important challenges in the study on robot locomotion behaviors nowadays.

The main issue in robot locomotion behaviors is to deal with adaptability and optimality to dynamic environments simultaneously. Many researchers strive for dynamics in robot locomotion behaviors. A human can move efficiently by integrating many aspects of the embodied cognition based on central nervous structures and morphological features. Time-series of external and internal sensory information enables an animal to make a decision to move in dynamic environments. Likewise, a legged robot also needs a dynamic locomotion generator to cope with diverse environmental conditions. In this study, a model is defined to represent the complex relationship between inputs and outputs. A cognitive model is composed of cognitive architecture and knowledge. The cognitive architecture is discussed from the functional points of view such as perception, decision making, prediction and learning to deal with intelligent abilities of humans and other animals. The need for such a cognitive model on the locomotion connects the ideas of biologists and physiologists with those of roboticists.

The dynamic human locomotion is realized by the simultaneous integration based on adaptability and optimality. The adaptability depends on the real-time perception as a bottom-up learning-based approach from the microscopic point of view, while the optimality depends on cognition as a top-down knowledge-based approach from the macroscopic point of view. In dynamic locomotion, the generated movement has to have an objective. According to Pfeifer, cognition, embodiment structure, and the locomotion generator should be integrated when developing a reliable neurobiological locomotion model [218]. By incorporating cognition into the model, it becomes possible to advance from sensory-response systems to a more sophisticated interpretative response system that can navigate dynamic environments. Therefore, I propose a neuro-cognitive model for multi-legged locomotion that integrates strongly with both external and internal S.I. By considering interdisciplinary studies, I develop a cognitive model useful from lower-level until higher-level controller, from short-term to long-term adaptation. It realizes the seamless integration from multi-modal sensing, ecological perception and cognition through the coordination of interoceptive and exteroceptive sensory information. A cognitive model can be discussed from three different scopes: micro-, meso-, and macro-scopic corresponding to sensing, perception, cognition, in short-, medium-, and long-term adaptation related to neuroscience and ecological psychology. Basically, the multi-legged locomotion requires the intelligent functions of 1) attention module, 2) an adaptive locomotion control module, 3) an Object recognition module, 4) an environmental map building module, and 5) an optimal motion planning module, and the proposed neuro-cognitive model integrates the above intelligent functions from the multi-scopic point of view. The flow diagram of the problem statement to the methodology can be seen in Fig. 1.1.

Robot behavior in ecological scale Behavior generation implies understanding of perception, action, cognition. They must identify properties of the environment that define what is perceived, acted, and known. It indicates the individual behavior and environment is inseparable. In ecological psychology, this integration is not represented by the physical science. The integration behavior and environment has different representation in different individuals. The behavior is respected with what the sense organs of individuals perceive to the environment that can not detect the physical information directly. However, the perceptual systems still can recognize a certain range and events, tends to perceive the changes, process, events, and sequence of event rather than the level of time. It implies scaled-based perception of individuals. Human can recognize the moving trajectory of animal in large scale area and detect the change of movement of animal in small scale [87].

In behavior-related descriptions is defined at ecological scale, begin with substance, surfaces, places, objects, and events [87]. The substance influences to the characteristic properties of the surfaces, key aspect where the leg should step on. Object and Place is the extension of surface, while events is related with the temporal time. In macroscopic scale, the behavior

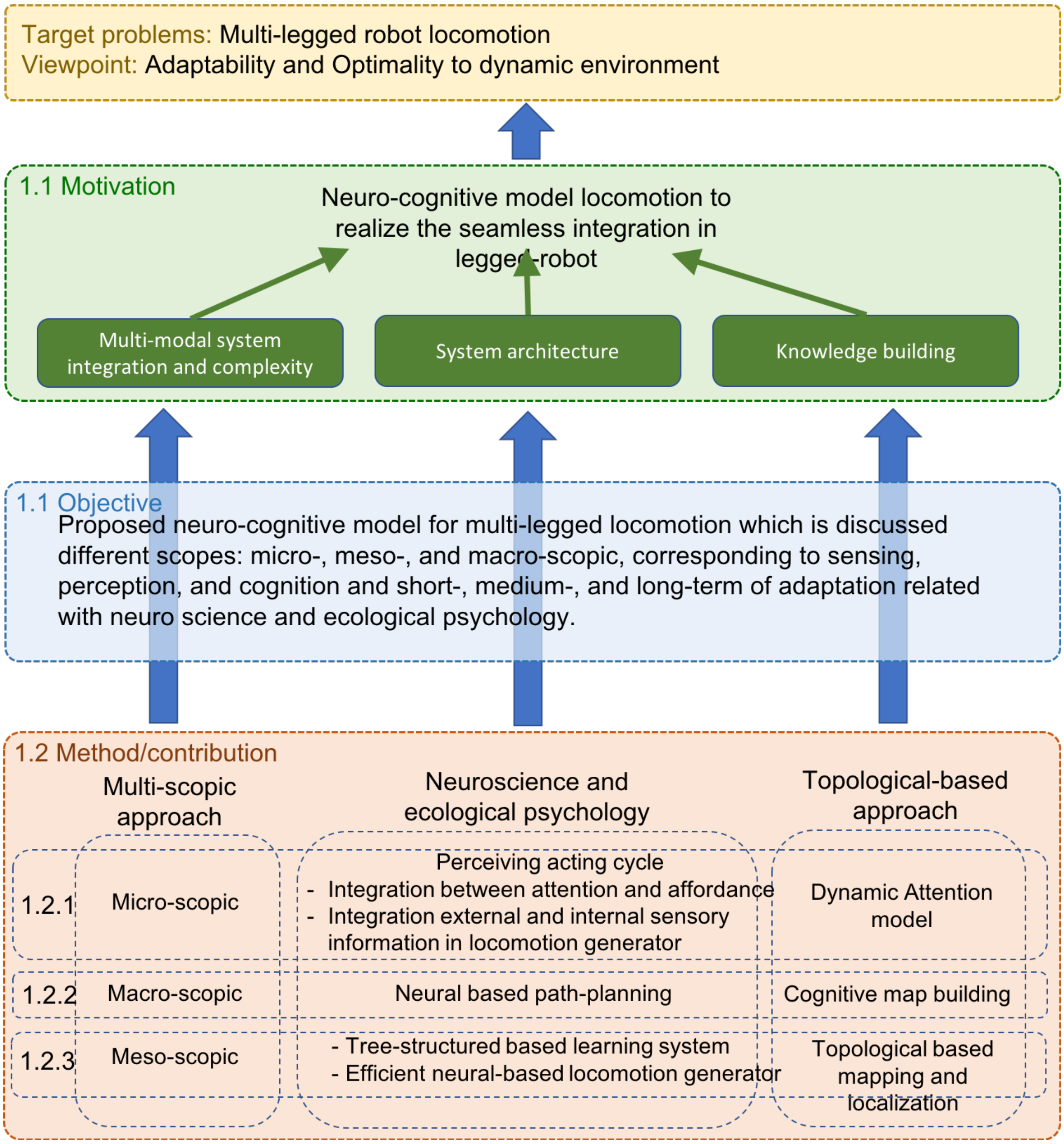

Target problems: Multi-legged robot locomotion
Viewpoint: Adaptability and Optimality to dynamic environment
1.1 Motivation
Neuro-cognitive model locomotion to realize the seamless integration in legged-robot
Multi-modal system integration and complexity
System architecture
Knowledge building
1.1 Objective
Proposed neuro-cognitive model for multi-legged locomotion which is discussed different scopes: micro-, meso-, and macro-scopic, corresponding to sensing, perception, and cognition and short-, medium-, and long-term of adaptation related with neuro science and ecological psychology.
1.2 Method/contribution
Multi-scopic approach
Neuroscience and ecological psychology
Topological-based approach
1.2.1
Micro-scopic
Perceiving acting cycle
- Integration between attention and affordance
- Integration external and internal sensory information in locomotion generator
Dynamic Attention model
1.2.2
Macro-scopic
Neural based path-planning
Cognitive map building
1.2.3
Meso-scopic
- Tree-structured based learning system
- Efficient neural-based locomotion generator
Topological based mapping and localization

Figure 1.1: *The structure of the dissertation*

represented as the movement planning decided by some intention and situation. In microscopic, it tend to focus on how the to manage the movement of muscle or actuator. Here, the systems in mesoscopic scale has important role integrate between more microscopic and macroscopic scale [229]. In this case, the behavior provision, such as speed and moving direction can be defined at mesoscopic scale. This mesoscopic approach is also used for conceptualizing the functionality of organism [26].

This integration between embodiment and environmental defined in ecological scale might be an alternative strategy to decrease the complexity in developing integration between robot behavior and environment properties. However this is still challenging to realize in robotics behavior.

Cognitive Science and Multi-scopic Processing From the viewpoint of cognitive science, the human dynamic locomotion is realized by the simultaneous integration based on adaptability and optimality. The adaptability depends on the real-time perception as a bottom-up learning-based approach from the microscopic point of view, while the optimality depends on cognition as a top-down knowledge-based approach from the macroscopic point of view. I behavior generation, optimality implies how to evolve to form efficient behavior by considering the aspects of costs and benefits [213]. For example, lower-level processing can strongly integrate sensory input and embodiment structure from the microscopic point of view deals. Higher-level processing can conduct motion planning, behavior generation, and knowledge building from the macroscopic point of view. In addition, the cognitive processing module can integrate these two processing at the mesoscopic level. Artificial Intelligence and Macroscopic Processing Various types of methods on artificial Intelligence have been applied to robotics from the macroscopic point of view. Especially, path planning and map building. In robot locomotion, map building and path planning has strong relationship in higher-level processing which has general characteristic. Map building can be represented by a topological map that represents the possibility of movement and its cost. It is become important input for path planning to generate appropriate action. In multiscopic level, path planning generate general action such as speed and direction of movement.

Neuro-science and Microscopic Processing Sensory-motor coordination such as spinal tracts, central pattern generators (CPG), muscle synergies, and afferent feedback is important to discuss adaptability in biologically inspired robotic. Here the sensorimotor coordination implies a coupling mechanism between sensory process and action generation in microscopic point of view. In legged locomotion, research of sensorimotor coordination draws together numerous underlying principles, but it is still unclear how these principles are integrated into animal motor control. It is, therefore, essential to analyze their natural integration. Animals have a complex neuronal structure for generating dynamic locomotion. The basic locomotion pattern is generated by central pattern generator (CPG) in the spinal cord. From there, the pattern stimulates other neural pools towards muscle synergies [34, 205, 239]. This spinal reflex process contributes to the overall movement pattern, muscle activation, and the modulation of CPG. Biologically-inspired models for generating locomotion include those with central pattern generator [121, 304, 315, 244], and may include muscle models as the target actuator [80]. However, it is difficult for the above models to generate dynamic locomotion faithful to that of humans or other animals. In order to realize the dynamic locomotion, it is important to observe how a human or animal realize dynamic locomotion behaviors from the multiscopic point of view.

Cognitive Psychology and Multiscopic Processing The cognitive psychology is the field of psychology that analyze the process of human thinking, such as perception, attention, action, learning, and planning from the lower-level of adaptive processing to higher-level of intelligent processing. Each process has its unique characteristics. Lower cognitive processes arise naturally with lower levels of consciousness, whereas higher cognitive processes actively exercise mental tasks with intention. Such kind of process implies multiscopic processing that integrates lower- and higher -level processing. Various cognitive architectures have been proposed in the cognitive science until now, but it is difficult for robotic researchers to apply such cognitive models to deal with adaptability and optimality simultaneously in real dynamic environments because most of them only explain the information flow of human cognitive behaviors in the conceptual level, and the methodology to implement functions like human cognitive behaviors is not described in detail. Some researchers developed integration methods of the perceptual system and behavioral system. Some of them used a vision sensor combined with a control system to detect an obstacle for generating appropriate robot behavior [115]. However, most of the locomotion and perceptional systems were separately developed. They built a perception model for obstacle avoidance and used its output (movement plan) as the input of the locomotion model. They implemented their approach in a legged robot [24, 265]. Barron et al. [18] developed perception-based locomotion implemented in a hexapod robot. They used visual information from the perception model to provide feedback in obstacle avoidance and target tracking behaviors. The perceptual model may be used for motion planning that generates the location of footsteps. Thus, trajectory-based locomotion generates stepping movement in certain leg [145, 327]. It is also used for affordance-based perception [65]. In bio-inspired locomotion, currently, the motion controller does not reach the short-term adaptation. It only controls the movement plan based on cognitive processes [264, 266, 92]. Xiong et al. [314] proposed a short-term adaptation model in a legged robot. They considered only internal sensory feedback. However, human or animal movement does not require exact planning, it may be generated online as a consequence of elementary behaviors of steering and obstacle avoidance [64]. Therefore, realizing multiscopic processing implies seamless integration is essential to achieve adaptability and optimality in dynamic environmental condition.

1.1 Problems statement

1.1.1 The important of multi-legged robot in real world application

Robots have different structures for different purposes [77]. Arm-like robots feature in industrial contexts for performing hand-like functions. Multi-legged robots with a wheeled base are often used in social and entertainment contexts. Likewise, robots with legs have an advantage on rough terrain, making them suitable for disaster contexts and dynamic environment [20]. While, wheeled robot has more efficient and has higher speed on flat terrain (see Table 1.1). However, from a broader perspective, legged robots are more versatile than wheeled robots simply because less than half of the world's terrain can be accessed on wheels. Furthermore, Quadruped robot has larger coverage area and higher prospect if its movement and energy efficiency can be improved [35].

Table 1.1: *Comparison between wheeled and legged robot*

Criterion	Wheeled	Legged	Wheeled-legged
Speed maneuvering	High	Low	Medium
Obstacle crossing capability	Low	High	Medium
Step climbing capability	Low	High	High
Walking on soft terrain	Low	Medium	Medium
Walking capability on uneven terrain	Low	High	High
Energy efficiency	High	Low	Medium
Mechanical complexity	Low	High	High
Control complexity	Low	High	High
Adaptation capability	Low	High	High
Adaptation capability	Medium	Low	Low
Coverage Area	Smaller	Larger	Larger

Current legged robot has disadvantage on the movement efficiency and its adaptation capabilities. Higher (task) level control operates by assembling goal pursuit dynamics from efficient behavioral primitives remain as outstanding challenges [111]. Robots' ability to move in dynamic environments is strongly required, making it a challenge today for robot locomotion developers. From the table 1 information, some researchers try to improve efficient movement and adaptation capabilities by developing dynamic locomotion behavior.

1.1.2 The issue of legged robot locomotion in the adaptability and optimality to dynamic environment

The human dynamic locomotion is realized by the simultaneous integration based on adaptability and optimality, implies seamless integration from sensing, perception, cognition, and behavior generation. Researchers often impose constraints in order to simplify the integration that may limit how much dynamic integration can be achieved. Various cognitive architectures have been proposed in the cognitive science until now, but it is difficult for robotic researchers to apply such cognitive models to deal with adaptability and optimality simultaneously in real dynamic environments because most of them only explain the information flow of human cognitive behaviors in the conceptual level, and the methodology to implement functions like human cognitive behaviors is not described in detail.

Some researchers developed the integration of the perceptual system and motion behavior. Some of them used vision sensor combined with a control system to detect an obstacle for generating appropriate robot behavior [115]. However, most of the locomotion and perceptional systems were separately developed. They built a perception model for obstacle avoidance and used its output (movement plan) as the input of the locomotion model. They implemented their approach in a legged robot [24, 265]. Barron et al. [18] developed perception-based locomotion implemented in a hexapod robot. They used visual information from the perception model to provide feedback in obstacle avoidance and target tracking behaviors.

The perceptual model may be used for motion planning that generates the location of footsteps. Thus, trajectory-based locomotion generates stepping movement in certain leg [145, 327]. It is also used for affordance-based perception [65]. In bio-inspired locomotion, currently, the motion controller does not reach the short term adaptation. It only controls the movement plan based on cognitive processes [264, 266, 92]. Xiong et al. [314] proposed a short term adaptation model in a legged robot. They consider only internal sensory feedback.

However, human or animal movement does not require exact planning, it may be generated online as a consequence of elementary behaviors of steering and obstacle avoidance [64]. In the current models, the perceptual information is used to control higher-level motion planning, such as path planning or walking plan generation. Therefore, I aim to realize the seamless integration to achieve adaptability and optimality in dynamic environmental condition.

1.1.3 Legged robot locomotion from human or animal principle

Legged locomotion research draws together numerous underlying principles, but it is still unclear how these principles are integrated into animal motor control. It is, therefore, essential to analyze their natural integration. Animals have a complex neuron structure for generating locomotion. The basic locomotion pattern is generated by central pattern generator (CPG) in the spinal cord. From there, the pattern stimulates other neural pools towards muscle synergies [34, 205, 239]. This spinal reflex process contributes to the overall movement pattern, muscle activation, and the modulation of CPG. Biology-inspired models for generating locomotion include those with central pattern generator [121, 304, 315, 244], and may include muscle models as the target actuator [80]. However, current models still have not achieved dynamic locomotion faithful to that of humans or other animals.

1.1.4 The needs of strong integration between internal and external sensory information implies strong integration between cognitive information and locomotion generator

In the human viewpoint, the locomotion system involves sensorimotor coordination, which smoothly combines internal and external sensory information (S.I.) to produce agile locomotion behaviors. The exteroceptive process plays a role in directly controlling the joints to avoid the obstacle. Integrating the sensory information can be difficult when cognition is implemented separately from the locomotion generator. In current models of legged robot locomotion, most researchers consider only interoceptive information when building the locomotion generator. Some integrate locomotion and perceptional systems separately. In these cases, external S.I. is not directly used in low-level motion planning, and internal S.I. is not strongly affected in higher-level motion planning.

In current models of legged robot locomotion, most researchers consider only interoceptive information when building the locomotion generator. Some integrate locomotion and perceptional systems separately. In these cases, external S.I. is not directly used in low-level motion planning, and internal S.I. is not strongly affected in higher-level motion planning. Existing systems that integrate perception and locomotion mostly deal with foothold planning, which limits adaptation to the much longer timescale of whole footsteps [101, 212, 308, 145, 50, 91, 112, 81]. While [301, 171] dealt with real-time obstacle avoidance. Their system is unable to avoid obstacles that suddenly block a swinging limb. Researchers often impose constraints to simplify the integration of cognition and behavior generator, but this strategy may limit how much dynamic integration can be achieved.

Therefore, I proposed a neuro-cognitive model for multi-legged locomotion to realize the seamless integration from multi-modal sensing, ecological perception and cognition through the coordination of internal and external sensory information.

1.2 Motivation and Objectives

The main motivation of this thesis is to realize the seamless integration from multi-modal sensing, ecological perception, and cognition through the coordination of interoceptive and exteroceptive sensory information in multi-legged robot locomotion. In order to deal with such kind of model, I need to consider the complexity of multi-modal system integration, the architecture of the system, and the knowledge building. There are many cognitive models developed by robotics researchers; however, currently, there are no cognitive models dealing with adaptability and optimality to dynamic environments and dealt with such system integration. In this proposed thesis, I integrated interdisciplinary study from biological, physiological, and robotics viewpoints. I use a multi-scopic approach to solve the complexity of multi-modal system integration, a neuro-science and ecological psychology approach to deal with proposed system architecture, and a Topological-based approach to deal with knowledge building and external sensory processing.

The proposed cognitive model is discussed from three different scopes: micro-, meso-, and macro-scopic, corresponding to sensing, perception, and cognition and short-, medium-, and long-term of adaptation related with neuro science and ecological psychology.

At the microscopic level, I proposed the sensorimotor coordination model concept based on the perceiving-acting cycle, which is a lower-level control system interacting directly with the environment. I develop of attention model using topological structure by considering the topological based approach. Next, I build the integration between attention and affordance in moving behavior and integrating exteroceptive sensory information to lower level control of locomotion generator in short-term adaptation, by considering neuroscience and ecological psychology approach. Furthermore, I develop an efficient neural-based locomotion model with sensorimotor coordination.

At the macroscopic level, I develop a higher-level controller and long-term adaptation. I develop a cognitive map of the robot based on a topological-based approach, integrated with embodiment of the robot. Furthermore, I develop dynamic path planning model based on neuro-activity.

At the mesoscopic level, I develop the middle-layer system to integrate lower level and higher-level controller. I design localization and mapping models using topological-based information. Next, by considering the neuro-musculoskeletal model, I build behavior coordination for the omnidirectional movement controller. Furthermore, to realize an efficient learning model for neural-based locomotion models, I build a learning model using a tree structure for neuron interconnection. Finally, to shows the effectiveness of the proposed system Integration, I show its implementation for advanced application.

1.3 Knowledge Contribution

Based on the empirical observation series experimental evaluation, the proposed thesis's main contribution is developing a neuro-cognitive model for legged locomotion behavior. It covers and integrates from the low-level controller to the high-level controller, from short-term adaptation and long-term adaptation. Furthermore, there are detailed technical contributions divided into three scope system, MiSc, MeSc, and MaSc. Those are explained as follow:

1.3.1 Contribution in Microscopic level

Based on the experimental results, the contribution of the MiSc level can be treated in three categories: 1) Development of attention model using topological structure, 2) Integration between attention and affordance in moving behavior, 3) Integration of exteroceptive sensory information to lower level control of locomotion generator in short-term adaptation, and 4) Efficient neural-based locomotion with sensorimotor coordination.

Development of attention model using topological structure Attention can be represented by any parameter, depending on the data processed. In our case, I use only time-of-flight sensors for the external information; these generate 3D point-cloud data. I therefore chose the topological map model for optimal data representation. However, in the existing topological map building process offers no way to control node density in only a localized area. Our contribution addresses that through a dynamic density topological map-building process based on a growing neural gas. We put a strength parameter into every node to indicate each node's importance in the topological structure. This parameter will be defined based on the segmentation process in the previous time. The experiment in Section 3.1 shows that the improved dynamic attention process can clarify obstacles in front of the robot by increasing the number of nodes.

Other topological generator systems such as Self-Organizing Map (SOM), Growing Cell Structure (GCS), Neural Gas (N.G.), cannot increase node density in localized areas. Therefore, they need to increase the node density over the entire map in order to clarify even localized objects. [139, 75, 74]. Compared with other multi-density topological maps, such as multi-layer growing neural gas (ML-GNG) [285], our improved system may decrease the processing time by as much as 70 percent (ML-GNG = 3.1567×10^{-4} s, DD-GNG = 1.0255×10^{-4}). The localized attention-focusing process has also been proven to decrease the computational cost.

Building the integration between attention and affordance When affordance and attention are strongly integrated, affordance can be perceived with high accuracy. From the experimental results, the attention controller provides the topological structure information to the affordance detector. Then the affordance detector will interpret the environmental information by calculating the normal surface vectors. When the normal vectors indicate a suspected obstacle, the attention controller will increase the node density, hence the suspected area's information density. We can reduce computational overhead by reducing the raw data points (from the 3D point-cloud) to a triangulated mesh, which then serves as a topological map structure. The attention controller also automatically increases the node density when it is needed to clarify suspected objects. This mechanism is efficient for cognitive processing since only important information is processed. In contrast with existing methods [132, 81], our affordance detection can be 10 times faster, in comparison with the 1.36 ms required by [81].

integration of exteroceptive sensory information to lower-level control of locomotion generator in short-term adaptation From the experiment conducted, we can prove that our improved system integrates exteroceptive sensory information and low-level control of the locomotion generator. The system can, therefore respond to environmental changes in every time cycle. Existing systems that integrate perception and locomotion mostly deal with foothold planning, which limits adaptation to the much longer timescale of whole footsteps [101, 212, 308, 145, 50, 91, 112, 81]. While [301, 171] dealt with real-time obstacle avoidance. Their system is unable to cope with obstacles that suddenly block an already-swinging limb. Our proposed system has the advantage of controlling the locomotion in every time-cycle, which has been proved in several experiments avoiding sudden unknown obstacles.

Efficient neural-based locomotion with sensorimotor coordination In this part, we build a robust CPG model with an efficient structure that can generate various gait patterns.

It's integrated with sensory feedback that substantially impacts the CPG-modulation output. I also considering dynamic sensorimotor integration to compensate leg injuries at multiple speeds.

1.3.2 Contribution in Macroscopic system

The contribution in this system is in the higher-level controller and long term adaptation. We develop a topological-based cognitive map of the robot, integrated with the embodiment of the robot. Therefore, the cognitive map of each individual may be different.

Furthermore, we develop neuro-activity-based path planning that integrated with robot embodiment. We create a new model of dynamic path planning based on neuronal activity. A 4-legged robot is applied to prove the effectiveness of the proposed path planning model. The proposed model is expected to be online generated with obstacle avoidance and applied with unpredictable travel costs. These advantages do not exist in the current state-of-the-art research. Nevertheless, an integrated system is required for supporting the proposed model. The proposed model's main contribution is to use the human brain's natural mechanism to generate online path planning in 3-D rough terrain with unpredictable travel costs. The proposed model emphasizes the inner state process of the neuron and the development process of the neurons in the brain.

1.3.3 Contribution in Mesoscopic system

The contribution in this system is mainly on integration between higher level and lower level controller, which is classified into 1) Localization model using the topological map as input, 2) Omni directional behavior controller, 3) Building learning strategy using tree structure for neuron interconnection 4) System Integration and its implementation for advance application.

Localization model using topological map We proposed a real-time and continuous map building algorithm using the topological structure as an input. The topological structure is composed of 3D vector positions of nodes, edges, and 3D surface vectors of nodes generated from growing neural gas. By using this model, the number of data representations of the map can be decreased.

Behavior coordination for omni-directional movement This section contributes to developing the learning strategy to develop an omnidirectional controller to convert the higher-level information (speed, direction) to the lower-level controller (limit cycle signal).

Most of the limit cycle development only considers the speed in unidirectional walking [107, 61, 97]. Endo *et al* developed adjusted walking velocity for neural oscillator based locomotion. However, the range of adjusted velocity is small [61]. In 2008, Manoonpong *et al* developed neural-based locomotion to generate omnidirectional movement in any type of legged robot. The proposed model can easily be adapted to control other walking machines without changing the internal network structure and its parameters. This model also can produce at least 11 different walking patterns and a self-protective reflex by using five input neurons [175]. Therefore, in this thesis, we proposed a dynamic structure model in the neural oscillator-based locomotion and learning model to generate unscaled omnidirectional movement in biped robots. Proposed omnidirectional walking cover combined 180 degrees in horizontal walking direction and turning walking level.

Building learning model using tree structure for neuron interconnection Overall, based on the experiments' validation, the contributions can be listed as follows: 1) using a tree structure based optimization strategy that can simplify the representation of the sensory-motor interconnection structure model. 2) using proposed a novel Bacterial Programming for optimizing the complex structure of the neural-based locomotion model. 3) Other novelties are regarding the initial population generation and the gene transfer operation in B.P.

Implementation in the challenging terrain The vertical ladder is the most challenging in these current issues in legged robot. However, Most researchers did not consider the movement transition between horizontal and vertical movements. The locomotion model was performed separately. Realizing the transitional movement to upper stair from the vertical ladder without handrail was the new problem [7]. Thus, a unique behavior is required to pass this challenge. In order to overcome that problem, we proposed a novel behavior that able to move from horizontal to vertical movement and transition to horizontal movement in the upper stair. We develop posture and swinging behavior that autonomously generated depending on the environmental condition.

1.4 Thesis Structure

This thesis is organized based on the multi-scopic viewpoint shown in Fig. 1.2. Chapter 1 explains the social background leading to this thesis research objectives. The main contribution of the research is also discussed and highlighted in this chapter.

Chapter 2 introduces behavior coordination for multi-legged robots to understand better the challenges reflected in this thesis. Next, I explain the study on locomotion control from conventional control theory to biologically-inspired control methods in neuroscience and cognitive science. Furthermore, I review well-established find-ings regarding the cognitive process from interoceptive and exteroceptive sensory information to motion control in human behaviors. In addition, the background concept of multi-scopic adaptation is discussed in this chapter.

Chapter 3 proposed the sensorimotor coordination model based on the perceiving-acting cycle at the microscopic level, a lower-level control system interacting directly with the environment. The sensory-motor coordination model comprises 1) attention mechanism module that controls the topological structure of 3D point cloud information by using Dynamic Density Growing Neural Gas (DD-GNG). DD-GNG can control the density of topological structures in specific area based on the attention, 2) object affordance detection module for the direct perception to generally identify the environmental condition based on the physical embodiment, 3) neural based locomotion module that generates dynamic gait patterns by integrating with sensorimotor coordination. The proposed model has been optimized through a learning process in the computer simulation beforehand. The locomotion ability of real multi-legged robot and the leg malfunction tests are demonstrated in the experiments by several different terrains. The experimental results show that a smooth gait-pattern transition could be generated during sudden leg malfunction.

Chapter 4 proposed the method of building environmental knowledge by using the topological structure-based map reconstruction. Next, we proposed an optimal path planning method on the constructed map. Experimental results show that the proposed method can extract environmental features for multi-legged robots and build environmental knowledge for optimal path planning.

Chapter 5 proposed a neuro-cognitive model that integrates the sensory-motor coordination model with environmental knowledge as a cognitive map using the bot-tom-up facial environmental information and top-down map information. The cognitive map is used to recognize a current facing situation by the topological structure information from a lower level, composed as a 3D vector position of nodes, edges, and 3D surface vectors of nodes. The robot generates intention towards the final goal from the macroscopic level. Further-

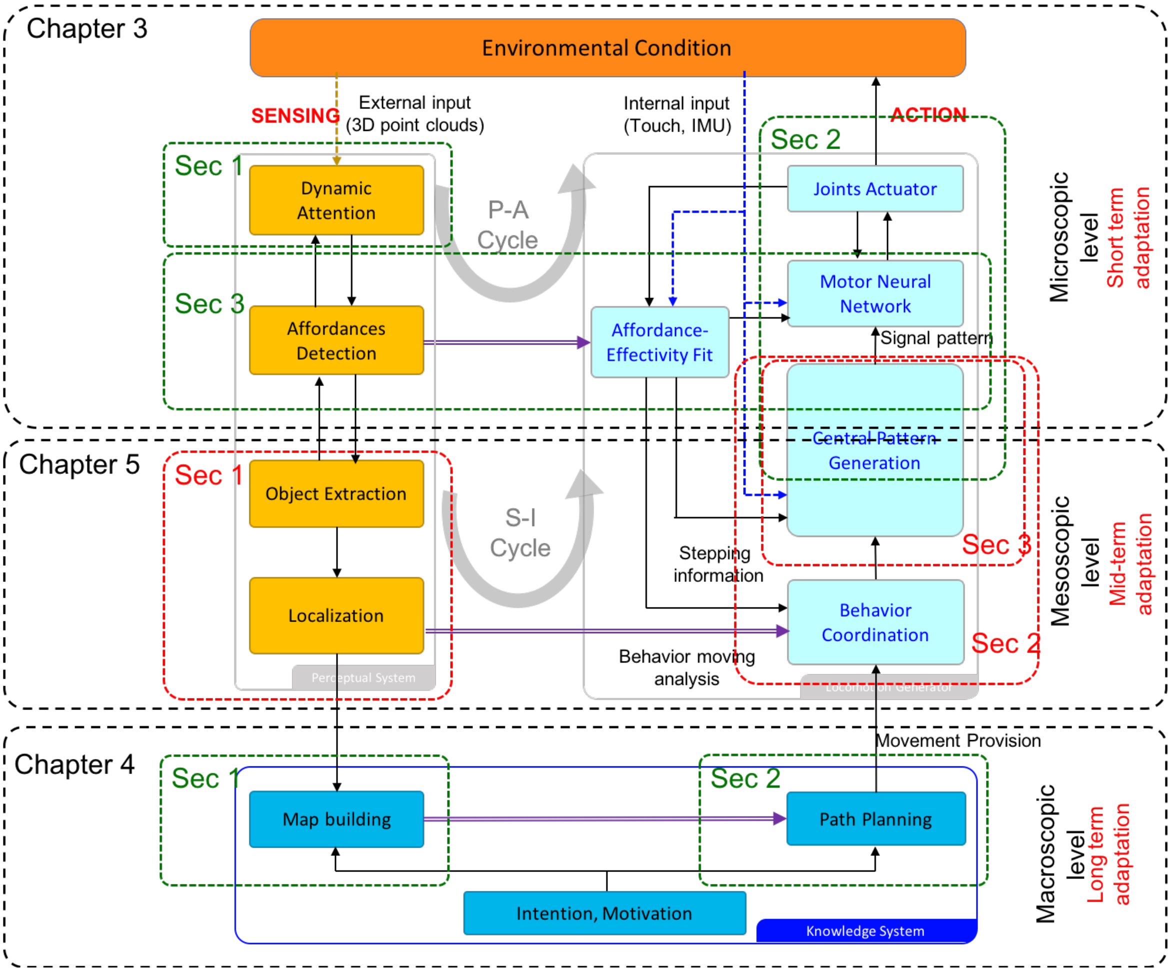


Figure 1.2: *The structure of the dissertation*

more, the learning process to integrate the relationship between the macroscopic level of behavior commands and locomotion performance is developed as well. The proposed model has been tested for omnidirectional movement in biped and quadruped robot. Furthermore, the proposed neuro-cognitive model has also been implemented for robot climbing behavior, performing a horizontal-vertical-horizontal movement.

Concluding remarks are summarized in Chapter 6. The experimental results on the multi-legged robot locomotion with the proposed neuro-cognitive model are discussed from the multi-scopic point of view. Finally, I discuss several future research directions in the inter-disciplinary study on cognitive science and ecological psychology.

Chapter 2

Locomotion Model in Legged Robot, Advantage, and General Issue

Robots technology has been implemented in many fields of our life. They have become necessary to ease human tasks in many contexts such as industrial, military, entertainment, and disaster settings. Robots have different structures for different purposes. Arm-like robots feature in industrial contexts for performing hand-like functions. Humanoid robots with a wheeled base are often used in social and entertainment contexts. Likewise, robots with legs have an advantage on rough terrain, making them suitable for military and disaster contexts. Some researchers use tank model robots for disaster problems and navigation in dangerous areas [322], while other researchers build a robot partner to support elderly people [324]. Furthermore, some researchers use humanoid robots for dangerous areas and rescuing humans [300]. Honda produced the biped robot “ASIMO” that can serve people in their social life. DARPA developed a legged robot for military service. Nevertheless, the legged robot is a suitable robot in many fields: it can be applied for social life [241], rescue [322], [300], military purposes, or entertainment (Soccer Robot, Dancing Robot) [257], [252]. From a broader perspective, legged robots are more versatile than wheeled robots simply because less than half of the worlds terrain can be accessed on wheels. Therefore, it is important to improve the development of legged robots.

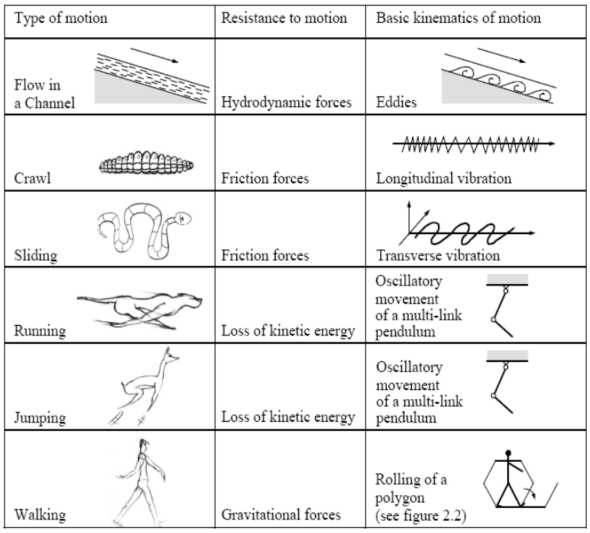
There are currently many varieties of legged robot exhibiting inspired designs and performance. Boston Dynamics has built many quadruped robots that have excellent capability on rough terrain. Quadruped robots developed at Waseda University have also demonstrated performance on rough terrain and in ladder-climbing. Their movement, however, seems slow compared with existing quadruped robots. Most legged robot researchers implement biological structures of quadruped animals to benefit from the animals performance. MIT, for example, has built a Cheetah-like robot that moves at high speed [118]. BigDog [223], Spotmini [4], HyQ [267], and Laikago [274] are inspired by dogs. They show flexibility of omni-directional movement on natural terrain. BigDog walks with a dynamically balanced trot gait. It balances using an estimate lateral velocity and acceleration, determined from the sensed behavior the legs during stance combined with the inertial sensors. BigDogs control system coordinates the kinematics and ground reaction forces of the robot while responding to basic postural commands. The control distributes load amongst the legs to optimize their load carrying ability. The vertical loading across the limbs is kept as equal as possible while individual legs are encouraged to generate ground reactions directed toward the hips, thus lowering required joint torques and actuator efforts [223, 313]. Ijspeert et al. took their inspiration from salamanders [49]. Animal-inspired robots, however, draw their mobility capabilities from the animals that they are designed after. In contrast to dogs and salamanders, cats are able to climb as well as walk, run and leap over rough terrain.

Robot technology has been implemented in many fields of our life, such as entertainment, security, rescue, rehabilitation, social life, and the military. Most researchers build robots for particular purposes. Some researchers use tank model robots for disaster problems and navigation in dangerous areas [322], while other researchers build a robot partner to support elderly people [324]. Furthermore, some researchers use humanoid robots for dangerous areas and rescuing humans [300]. Honda produced the humanoid robot “ASIMO” that can serve people in their social life. DARPA developed a humanoid robot for military service. Nevertheless, the humanoid biped robot is a suitable robot in many fields: it can be applied for social life [241], rescue [322], [300], military purposes, or entertainment (Soccer Robot, Dancing Robot) [257], [252]. Therefore, it is important to improve the development of legged robots capabilities.

2.1 Current Issue in Locomotion Behavior

Robotic locomotion behavior is one of the important functions of the robotic device that helps the platform in traversing rough terrain; moving and interacting in human environment. Although wheeled locomotion is very common, foolproof, energy efficient and easily controllable; there are other motion alternativesare also available such as legged motion including Bipedal, Quadrapedal, Hexapedal, and soon. Track belt, walking, running, hopping, swimming, slithering, brachiating, etc are the other behavior forms used in locomotion behavior.

Some researcher used nature principle as the basic of locomotion which can be seen in Fig. 2.1. In this part, we separate the locomotion model into 3 parts, which are, trajectory



Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel	Hydrodynamic forces	Eddies
Crawl	Friction forces	Longitudinal vibration
Sliding	Friction forces	Transverse vibration
Running	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum
Jumping	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum
Walking	Gravitational forces	Rolling of a polygon (see figure 2.2)

Figure 2.1: *Nature based locomotion principles [2]*

based locomotion, limit cycle based locomotion, and passive locomotion.

The locomotion generator, as its name suggests, generates the movement behavior appropriate to particular conditions. There are many models for legged-robot locomotion. Most researchers implement trajectory-based locomotion for its simplicity; this has been done in bipedal [174, 330, 195, 260, 133, 253, 308][141][178, 329, 184], and hexapodal robots [219, 335]. Trajectory-based models control the motion planning in Cartesian coordinates using polynomial equations or Bzier curves [174]. Other researchers use center-of-gravitybased trajectory generation for quadrupedal robots [308, 178]. These center-of-gravity trajectory models have been successfully implemented for complex terrain. However, this approach has proven lacking on dynamic locomotion behavior. The trajectory-based approach needs to plan scenario motion planning in advance, and requires extensive parameter-tuning.

2.1.1 Integration of Perception and Locomotion Behavior

In the integration locomotion behavior and perception process, researchers have used different sensors and different strategies. LittleDog [140] used stereo-vision to build the terrain model for the space in front of the robot. Then, it performs footstep planning for the next stepping movement [141]. Other researchers have done similar work in perception strategy [52] [79]. Havoutis et al. used an RGBD camera to perceive environmental conditions. Their robot then generates a motion pattern and undertakes foothold planning [101]. Their subsequent work continues on to advanced implementation, such as stair-climbing [308]. The MIT Cheetah robot performs impressively while running and jumping to avoid an obstacle [212]. This robot uses LRF sensors to detect upcoming obstacles, and identifies them using an iterative end-point fitting (IEPF) algorithm. Once an obstacle is perceived, the robot prepares the jump by controlling speed.

Manchester et al., used more complex external sensors such as vision, laser, and radar sensors. Their robot builds a terrain map model and then generates a sequence of footstep locations and associated joint trajectories. The perception is only effective on slow timescales. The footstep planning is updated in every footstep [174]. The high-rate timescale is used only for internal sensory response. Many researchers also conducted footstep planning, updated at every footstep, in both bipedal [50] [173] [145] and hexapodal robots [24]. Taking a different approach, Hoffmann et al. use a closed-loop strategy for perception and action. They developed interaction between the robots embodiment and its environmental context. The robot adjusts its gait or speed when environmental changes are detected [109]. In this work, the robot reconstructs its map before generating motion plans that address only high-level motion (speed, step length, step height). Next, the stability model controls the low-level motion. The external sensory information is hence not directly used in low-level motion planning. In our proposal, the cognitive model plays a role in the lower-level locomotion model. Using external sensory information and a laser sensor costs less to detect object shapes, than using a vision sensor.

2.1.2 Bio-inspired model as a alternative model for achieving dynamics

A major problem in legged robot locomotion is the dynamism to the environmental condition. It is difficult for the robot to keep the respond in different kind of condition. Therefore a dynamic locomotion system is required for the humanoid robot to be able to move on different ground surfaces and different slope terrains. A human-like locomotion model is desired for solving the above problems. As naturally evolved, the human locomotion process is the best model generating efficient movement and energy which has been optimized genetically. Because of that, several researchers tried to imitate this natural process which is called bio-inspired model. Besides, many researchers applied control based locomotion to achieving instant desired performance. In control based locomotion, joint angle values are commanded according to the trajectory scenario in Cartesian level. Therefore this locomotion model tends to control the desired joint angle instead of decreasing the required energy in its movement generation, where, in reality, joint positions are not commanded in natural locomotion.

In our previous research, we dealt with conventional trajectory generation that used zero moment point and the inverted pendulum approach [257], [252], [325]. Yoshida et al. implemented the double inverted pendulum for the balancing system in their humanoid robot [325]. Kim et al. also used a conventional approach for development of the HUBO robot: they applied a mathematical method to develop the trajectory generator in the robot [135].

Nevertheless, the conventional approach does not represent well human behavior during locomotion. It also needs high mathematical complexity to realize the dynamic walking pattern. Therefore, we propose locomotion, based on a biological approach, that represents the human behavior system, and that can be stable and flexible, depending on the environmental condition.

On the other hand, some researchers have tried other ways to develop dynamic locomotion patterns that can synchronize automatically with sensory feedback. They consider natural processes to develop locomotion models from human and animal gaits. Quadrupedal animals can generate gait patterns (walk, pace, amble, trot, gallop) automatically, depending on the animals intentions and environmental conditions. The animals body structure also regulates the gait pattern, which means every kind of animal has different gait efficiencies. Nakada et al. propose a neuromorphic locomotion model with a CMOS controller for inter-limb coordination in quadrupedal robots [194], while other researchers propose central pattern generation (CPG) for quadrupedal robot locomotion [119, 14, 186, 330, 277, 163]. Ijspeert et al. proposed CPGbased control of their salamander robot [119], which can transition dynamically from walking to swimming. Other researchers have developed integration between CPG and ground reaction feedback to synchronize the gait with terrain conditions [186]. Zhang et al., for example, designed a CPG-based controller for trotting [330]. CPG gait generators can be implemented using a spiking neural network [63] or a recurrent neural network [292]. Sun et al. used a decoupled neural CPG circuit for adaptive locomotion [277]. In our previous model, we combined the CPG with a Bzier curve model for efficiency. We implemented our ideas in a small quadrupedal robot, but it showed limitations on handling variant gait [244, 258]. The quadrupedal robot proposed in the present thesis will be implemented as an efficient neural-based locomotion model using a single-rhythm generator-based CPG model, and will include a reflex system for synchronizing with locomotion events.

2.2 Central Pattern Generation based Locomotion Generator

Researchers are pursuing the design of human-like and animal-like robots because of its advantages. Some researchers have developed robots focusing on communication and social skills. Nonetheless, it is also essential to develop locomotion as support to robots movements. Conventional method [130, 253] with bottom-up model using planning-based development is popular to realize robot locomotion. Researchers also considered interdisciplinary viewpoints, such as biological approach, in their works of robot locomotion. Bio-inspired models with central pattern generation (CPG) locomotion have been proposed previously [121, 244] for legged robot implementation. Some researchers implemented muscle model as target actuator for mimicking human or animal locomotion [80]. However, existing models have yet to achieve dynamic locomotion as human or animal does. In this thesis, we proposed the locomotion model for the quadruped robot by taking the benefit of the biological model in quadruped animals.

The basic central pattern generation has been proposed previously by several researchers [181, 182, 235]. This model is generated by mutual inhibition between certain neurons. Each neuron also acquires adaptation signal input. The rhythmic signal activity generated by the neural oscillator consists of two tonically excited neurons with the self-inhibition effect linked reciprocally via an inhibitory connection. Matsuoka discusses various aspects of frequency and pattern control: we can also generate different rhythm patterns by modifying parameters [181, 182]. Rowat et al. proposed a coupled neural oscillator with self-rhythmic generation ability. In this model, each neuron generates the signal to its pair. He divided the behavior pattern into six different types of neural oscillator [235].

The CPG can be used as alternative way to represent well human behavior during locomotion. The controller system in the brain has been proposed by Roy [236]: this is the basis of how the brain controls locomotion. Locomotion models based on a biological approach have been proposed by several researchers [119, 122, 279, 189, 197]. Before we applied locomotion in a biped humanoid robot based on the neuro-biological approach, we studied the locomotion system as adjusted by animal morphologies [59]. Four-legged animal locomotion has been proposed by Ijspeert et al. They control animal locomotion by using a neural oscillator and also design the transition mechanism from swimming to walking [119], [122], [123]. Locomotion based on the central pattern generation (CPG) approach in four-legged animal robots has been applied by several researchers [59]. Furthermore, CPG can also be applied for malfunction compensation in six-legged robots [226].

2.2.1 Implementation on robotics locomotion

2.2.1.1 Biped Robot

A CPG has been implemented for biped robot locomotion, as proposed by several researchers [279, 189, 197, 126]. Taga et al. used coupled neurons for generating the oscillated signal to drive the joint. They dealt with a sensory feedback system to adapt to the environmental conditions and created a mathematical model in order to acquire the feedback calculation. Their proposed method is applied in computer simulation [279]. Another neuro-model of locomotion is presented by Matos et al., who proposed a CPG approach based on phase oscillators for bipedal locomotion [180]. However, the ability to recover the disturbance is required. Ishiguro et al. also proposed the concept of a neural oscillator to realize two-legged robot locomotion. This model is applied to control a three-dimensional biped robot that is intrinsically unstable. They applied a feedback sensor to form dynamic locomotion; however, the robot has a limited degree of freedom and is applied at simulation level only [126]. In 2014, Nassour et al. proposed the locomotion model in the humanoid biped robot: they extended the mathematical model of CPG and designed a multi-layered neuron connection in order to control various models of walking [197], [196]. Nassours research has good stability; however, the aim of our research is to improve the stability level of walking. In 2010, Park et al. designed a locomotion using an evolutionary optimized CPG. They also proposed sensory feedback to support the walking model; however, they did not consider a learning system for stability [211]. Other researchers have considered center of mass (COM) for their locomotion, based on central pattern generation [113], [210]. They have not considered however the stability of the learning system, or control to get various walking patterns.

2.2.1.2 Quadruped Robot

Locomotive capacity is essential for movement in a dynamic environment; animals exemplify this, having to move efficiently to conduct their activities. Quadruped animals, specifically, produce dynamic locomotion patterns ranging from walking to galloping. These patterns are generated by CPG structures using sensory feedback and spinal reflexes [215]. Primitive CPG structures generate dynamic patterns even when sensory feedback and signal commanding are absent [205]. In the last two decades, integrating neuroscience and robotic locomotion has been considered as an alternative approach to dynamic locomotion. Many researchers have proposed CPG concepts in the service of robot locomotion [326, 248, 122]. However, the question of the best way of controlling quadruped locomotion has yet to be resolved.

Quadruped locomotion based on CPG is a part of various strategies for controlling speed and direction. Some researchers have used mapping selection of the CPG network to generate various gait patterns [242, 160, 230, 78, 330]. Our previous research used a generic locomotion model that enabled the generation of various gaits using simple parameter changes made by selecting designed coupling matrices. Sensory feedback affects the CPGs change-of-the-state pace [230, 245]. Furthermore, we used an optimization model to identify the appropriate CPG structure for the desired gait pattern [254]. Other researchers have also used CPG for pattern generation [161]. In [161], CPG generated a signal to stimulate a certain leg to perform a stepping movement; the stepping pattern had been mapped beforehand. Kimura et al. developed various CPG that were integrated with sensory feedback to generate torque [136] and gait-pattern modulation [78]. However, CPG has been reconstructed as development changes the individual body (e.g., infants to mature, frog) [283]. Generating different gait patterns without abruptly reconstructing the network is still challenging: creating an efficient structure that can generate dynamic patterns and developing integration of sensory feedback to establish swing timing have still not been achieved [122, 216].

In some cases, the spinal cord systems of humans and animals regulate locomotion when a leg is injured, changing the role of sensory feedback within the locomotion network [149, 233]. In the field of robotics, Ren et al. have considered the leg-injury compensation in the hexapod robot using multiple chaotic CPG and a master-client strategy [227]. Some researchers tend to use conventional approaches to solving leg malfunction problems in the legged robots [153]. Still, conventional models need to consider many aspects, leading to a high computational cost, and it has proven difficult to find the correct strategy for compensating a broken leg using CPG in quadrupedal locomotion. Considering the sensorimotor coordination mechanism is important for developing dynamic locomotion in cases of malfunction.

This thesis responds to the current challenges facing developing CPG for quadruped locomotion, presenting an efficient, robust CPG model that is dynamically integrated with sensory feedback to generate various gaits while also considering leg malfunction compensation without involving too many parameters. The model uses two sensorimotor-coordinationbased mechanisms [233, 149]: 1) because sensory feedback is critical for adjusting CPG modulation, proprioceptive signals from the leg force and swinging phases are integrated with rhythm-generator neurons (RG) in the CPG; 2) because leg malfunction affects the integration of sensory feedback and RG, a neuron representing a certain injured leg sends a signal and influences the effect of the sensory signal on RG.

2.2.2 CPG-based omni-directional movement controller

Omni-directional movement controller is current issue of bio-inspired locomotion model. Omni directional locomotion model provides dynamic movement and ability to modify its motion quickly to support the robot to move in dynamic environment [6]. Most of the limit cycle development only consider speed in unidirectional walking [107, 61, 97]. Endo *et al* developed adjusted walking velocity for neural oscillator based locomotion. However, range of adjusted velocity is small [61]. In 2008, Manoonpong *et al* developed neural based locomotion in order to generate omnidirectional movement in any types of legged robot. The proposed model can easily be adapted to control other kinds of walking machine without changing the internal network structure and its parameters. This model also can produce at least 11 different walking patterns and a self-protective reflex by using five input neurons [175]. Therefore in this thesis, we proposed dynamic structure model in neural oscillator based locomotion and learning model in order to generate unscaled omnidirectional movement in biped robot.

In quadruped robot Matos *et al*, presented a methodology to modulate the CPGs parameters in different speed and orientation and reducing the control dimensionality [179].

2.2.3 Integration to Muscle based actuator

Locomotion principles have been proposed in terms of limbed element. However, it is unclear how this principle works to regulate motor control in human or animal. Therefore, analyzing natural integration is important to find the role of motor control. Human or animal involves a complex architecture of the neurons to produce their locomotion. Basically, locomotion pattern is produced by CPG in spinal cord interconnected by other motor neural pools to stimulate and synchronize with muscle synergies [34, 205, 239]. This spinal reflex has a big contribution in movement pattern control, muscle stimulation, and CPG modulation. Quadruped animal can demonstrate that CPG produces variations in gait patterns such as walking, trotting, running, and galloping [215]. Spinal reflexes will incorporate mechanical sensory information by alpha motoneurons bypassing central inputs into muscle activation.

Researchers have developed muscle-based model for both biped and quadruped robots. They focused on the development of mechanical design based on the musculoskeletal model. Some researchers developed musculoskeletal based actuator for their quadruped robot. They also succeeded developing its stable locomotion [293, 317]. Other researchers focused on locomotion control using musculoskeletal model. Toeda et al, have designed the locomotion model of rats based on the CPG and the muscle model [289]. However, the model is still far to be applied to robotics implementation. Besides, the development of both forelimb and hindlimb muscle reflex model is still few. In this thesis, our contribution is development of a muscle-based locomotion model inspired by cat musculoskeletal model. We are exploring further the benefit of locomotion process in biological system for its application in dynamic locomotion of legged robotics. It shows not only the locomotion performance in robotics but also the mechanism of locomotion process in a cat.

2.3 Embodiment, Cognition, locomotion for Dynamic Behavior

When we talk about a legged locomotion system, one of the most important words coming to our mind is “dynamic”. Nowadays, a lot of researchers attempted to approach that word in robot locomotion. An important question is what the indicators of a dynamic locomotion should be. How far is a robot’s behavior able to imitate a human or animal’s behavior?

Human locomotion system is not limited only to motion generation. It integrates several aspects such as stability, perception, sensing, and memorization. From the higher-level control through the actuator parameter generator to the movement behavior generation, there is no exact parameter for commanding the movement. Researchers try to realize some constraints in order to simplify the integration of cognition and behavior generation. This strategy may limit their dynamic integration. When the constraints are decreased the dynamic level may be increased.

Nevertheless, an imitation model can be described from biological, mechanical, and cognitive viewpoints. Some researchers consider the mechanical structure of animal [247] or human [259]. There are biologically inspired models, too [106]. Furthermore, there are models considered from the cognitive viewpoint [312, 47]. This viewpoint depends on its implementation.

Human or animal movement will always consider sensory-motor coordination implying the integration of sensory information and action realized by the motor neurons. The sensory information will adaptively affect the motor action. In this implementation, sensory information is composed of visual, vestibular, and somatosensory components. The action is represented by the motor’s action generated by the brain and spinal cord. Vestibular, visual, and somatosensory components are combined smoothly to produce a sense of orientation and movement. This information is processed in the cerebellum, basal ganglia, and supplementary motor areas. Somatosensory information has the fastest processing time for rapid response, followed by the inputs from the visual and vestibular systems. When sensory information from one system is inaccurate because of an injury, the central nervous system (CNS) will combine information from the other two systems. This adaptation process is called sensorimotor coordination [137].

Based on the information above, we may divide the sensory system into internal sensory information (somatosensory and vestibular) and external sensory information (vision). Both internal and external sensory information should be coordinated in order to perform a real bio-inspired system. In fact, the visual system (external sensory information) influences the musculoskeletal system in the body in coordination with the other sensory systems [72]. The separation between cognitive and locomotion model may limit their dynamic integration.

In order to consider a dynamic locomotion model, the generated movement has to have an objective. According to Pfeifer, cognition, embodiment structure, and locomotion generator should be integrated for developing a reliable neurobiological locomotion model [218]. Thus, in order to develop a dynamic locomotion model, the cognitive model should also be considered. It has to use not only internal sensory information but external sensory information as well.

In order to realize the proposed model, it is important to look how human or animal behavior works. We analyze the human cognitive-action model from the viewpoint of ecological psychology. We propose neuro-cognitive locomotion that shows a strong integration of cognition and locomotion. The proposed locomotion model covers the attention of external sensory information, perception, and locomotion generation. We emphasize the attention mechanism as the primary processing mechanism in our proposed locomotion model. On the higher level, we developed behavior adaptation in three different terms, short term, medium term, and long term as explained in Section 2.6.

2.3.1 Perception and Motion Integration

Some researchers developed the integration of the perceptual system and motion behavior. Some of them used vision sensor combined with a control system to detect an obstacle for generating appropriate robot behavior [115]. However, most of the locomotion and perceptional systems were separately developed. They built a perception model for obstacle avoidance and used its output (movement plan) as the input of the locomotion model. They implemented their approach in a legged robot [24, 265]. Barron et al. [18] developed perception-based locomotion implemented in a hexapod robot. They used visual information from the perception model in central pattern generator (CPG) to provide feedback in obstacle avoidance and target tracking behaviors.

The perceptual model may be used for motion planning that generates the location of footsteps. Thus, trajectory-based locomotion generates stepping movement in certain leg [145, 327]. It is also used for affordance based perception [65]. In bio-inspired locomotion, currently, the motion controller does not reach the short term adaptation. It only controls the movement plan based on cognitive processes [264, 266, 92]. Xiong et al. [314] proposed a short term adaptation model in a legged robot. They consider only internal sensory feedback. In our previous research, we reconstructed the environmental condition and control motion planning by generating the walking plan in four-legged robots [44, 261].

Human movement does not require exact planning, it may be generated online as a consequence of elementary behaviors of steering and obstacle avoidance [64]. In the current models, the perceptual information is used to control higher-level motion planning, such as path planning or walking plan generation. In this thesis, we propose a cognitive model integrated with a neural-based motion generator that is able to perform in all term adaptation. The perceptual information is not only processed in higher-level planning, but also in short term adaptation. It can respond to some sudden upcoming obstacles during the swinging process.

2.3.2 Gaze Attention for Bio-inspired Robotics

When we are walking through a free and safe area, our attention may not totally focus on the way we are moving. Once we meet an obstacle or disturbance, our attention is increasing or focusing on the obstacle. It is required for us to evaluate the obstacle for deciding the next appropriate action intensely. Furthermore, when we are walking in unsafe environmental condition or on uneven terrain, our attention will always strongly focus on the way. We ensure carefully every stepping action generated. Therefore, this mechanism implies the local perception may change the attention dynamically.

Gaze attention system using an image sensor was proposed by several researchers [128, 127]. It is inspired by neurobiological model and the behavior of the early primate visual system. The review of image-based visual attention was provided in [73]. Ude et al. implemented the attentional model in their humanoid robot [296]. The gaze concept in robotics was implemented for mapping in robot soccer [138]. It was also implemented for selecting an appropriate object in robot manipulation problem [305]. Nevertheless, image information requires deep processing for building object's shape reconstruction in 3D. On the other hand, point cloud data generated by depth sensor is more efficient to reconstruct in 3D [145, 29, 172]. There are many robots that effectively detect and recognize obstacles using depth sensors [212, 99, 37]. The WAREC robot, for example, has a rotating laser range-finder array for scanning the surrounding environment [99]. Since depth sensors are limited in frequency rate, size, weight, and range, we propose a light-weight array of time-of-flight sensors which alleviates these limitations. In our proposed model, 3D information is essential to control the swinging of the leg. Therefore, we use 3D point cloud data generated by the depth sensor as the input data.

The processing of the raw data of 3D point cloud requires a preliminary step to reduce the computational cost. Topological map model is effective to represent the raw data with smaller size [287, 237, 291]. The current topological map generation has to be improved with the attentional system in order to clarify the detailed surface of some objects. In this thesis, we propose a dynamic density topological map building based on growing neural gas. The intensity of nodes and edges can be dynamic depending on the attention area. Therefore, the attention area can be better detailed and clarified due to be represented by a higher number of nodes.

2.4 Integration from external sensory information to movement commanding

There are a lot of locomotion principles in legged stuff proposed by researchers. But it is still unclear how these principles are integrated with human or animal motor control. It is therefore essential to analyze their natural integration. Human and animal have a complex neuron structure for generating their locomotion. The locomotion pattern is basically generated by central pattern generation (CPG) in the spinal cord that is connected by other neural pools to integrate with muscle synergies [34, 205, 239]. This spinal reflex contributes to control of the movement pattern, muscle activation, and the modulation of CPG. Quadruped animal shows that the function of CPG generates differences in gait patterns from walking, trotting, pacing, and galloping [215]. Spinal reflexes are able to integrate the mechanical sensory information to the muscle activation via alpha motoneurons bypassing central inputs.

Nevertheless, there are other reflexes effects to the motion generation, such as placing reflex. It maintains body posture and supports its stability. The placing reflex manages the information of tactile stimuli from the soles of the feet [66]. For example, when our leg is stumbled by the stone during its swinging phase, the leg is lifted in reaction to avoid the stone. Then it rapidly swings forward and stretches for supporting the body going to fall. It has a similar reflex mechanism when we perceive a sudden obstacle and try to avoid it before touched. However, this mechanism has a more complex integration that involves our cognitive system. Therefore, in order to achieve dynamic locomotion, the role of cognitive information should be considered.

2.4.1 Involving Information from Supraspinal Level to Motoneuron

In order to analyze the involvement of cognitive process in supraspinal level, some researchers analyzed corticospinal tract in the animal. When there is no obstacle given, the activity of corticospinal is low. When the obstacle is given during walking, the activity of the corticospinal is high during swing phase [56]. The animal required to visually regulate the limb trajectory for avoiding the obstacle or positioning on the safe location [214]. Furthermore, by analyzing the motor cortex in the limb, there is a significant increase of neuron activity during high attention movement, such as in narrow way, rough terrain, avoiding obstacle, especially during the swing phase that required high preciseness foot placement by controlling the limb trajectory [232, 11, 12, 13, 21, 22, 23, 306]. Those studies and observations show the effect of visual information to the locomotion from motor cortical discharge. Other examination has also been conducted by analyzing the activity of red nucleus neuron in cat animals during performing walking. There is an increasing activity during stepping the obstacle [151]. It shows that the red nucleus also affects the modifications of the pattern of muscle activity. The red nucleus signal through rubrospinal tract primarily has functioned in forelimb or upper limb, and has lower function from corticospinal tract [228, 154]. However, the transformation process of visual information from occipitoparietal area into corticospinal tract does not have clear regulation [298].

2.4.2 Mechanism of Motoneuron Commanding by Cognitive Information During Obstacle Avoidance

In neuro-physiological studies, when facing the obstacle, there are several strategies generated depending on the kinesthetic of the environmental condition. Chu et al, analyzed the behavior of cat avoiding obstacle during walking [46]. They analyzed that cats tend to change the muscle activity without altering the locomotion pattern. This shows the effectivity of neuro-motor process. Human also has similar behavior when facing the obstacle. He tends to maintain the phase duration or locomotion pattern, and prefers to alter the muscle activity [64]. This strategy is supported with CPG model proposed by Rybak et al that shows the mechanism of commanding the motoneuronal pools for controlling direct muscle and bypassing the spinal neural network of rhythm generator [187, 239, 238]. In contrast, when forced to step over the obstacle, human and cat tend to control the trajectory of swing as well as its duration. From the inspection in cat locomotion, the stimulation from both red nucleus and motor cortial effect to flexor muscle in the swing phase [228, 206].

2.5 Visualization technology and environmental map construction method for locomotion guidance

In order to perform goal oriented movement, it is necessary to estimate the self-position from the distance sensor mounted on the remote-controlled robot and construct a map. Such a method of simultaneously performing map construction and self-position estimation is called

Simultaneous Localization and Mapping (SLAM), and is one of the most basic techniques for search and decision making in an unknown environment. Even in remote monitoring using intelligent technology, map information is indispensable when the remote operator operates the robot safely and when the robot autonomously moves in the environment, and the accuracy is high. It is desirable to construct map information that is expensive and easy for both operators and robots to recognize. Here, we will describe the conventional research of the basic SLAM method and also explain the visualization method.

2.5.1 Simultaneous Localization and Mapping

Basically, the SLAM problem occurs when the robot cannot access the environment map or identify its own position. In other words, if all the accurate measurement data $z_1 : t$ and the control amount $u_1 : t$ up to the time t are given, the robot can easily construct the map and the SLAM problem does not occur. Therefore, in SLAM, it is considered to be an important key to appropriately estimate the boundary between the map data constructed by the robot and the data in an unknown environment. Here, we first explain the formulation of the SLAM problem from a probabilistic point of view [284]. In general, SLAM problems are divided into the following two methods. One is an online SLAM that estimates only the posture xt at the current time t.

The method for solving such SLAM problems is also roughly divided into two. One is a map construction method based on the above-mentioned stochastic expression whose technology was established by Thrun et al. Various probabilistic methods have been proposed, such as EKF-SLAM [147, 116] using the Extended Kalman Filter (EKF) and Graph SLAM [129, 70, 134, 200, 27, 71] using the graph representation. There is. In EKF-SLAM, it is one of the solutions to the online SLAM problem, and the feature map (landmark) represented by the coordinate points is used as the map representation. In EKF-SLAM, a provisional current position is estimated from the control amount at the current time, and based on the position information, maximum likelihood estimation is used to construct the feature amount measured at the current time and the features so far. This is a method of estimating the current position by minimizing the error with the feature map The problem with EKF-SLAM is that as the feature map is constructed, the amount of calculation for comparison between the measured data and the feature map at the current time becomes enormous, and SLAM in real time becomes difficult. There are things like putting it away. Graph SLAM is one of the solutions of the complete SLAM method, and it is a method to perform SLAM offline using all the data acquired up to the current time. Therefore, in Graph SLAM, all data can be accessed, and the result of overall map construction is corrected each time data is added, making it possible to construct a highly accurate map. However, SLAM in real time is difficult, and there are problems such as the need for a huge data space because all data must be retained. The most successful example of the most successful two-dimensional map construction at present is the Rao-Blackwellized Particle Filters (RBPF) SLAM proposed by et al. Due to its robustness and high accuracy in various studies. It shows its effectiveness [284, 129, 71, 38, 93]. However, in order to obtain highly accurate self-position estimation results using this method, highly accurate odometry information is required. However, many remote-controlled robots use a special mechanism, and it is often difficult to acquire highly accurate odometry data from sensors such as encoders. In such cases, these probabilities It is difficult to use a conventional method.

2.6 Proposed Multi-scopic Model for Locomotion

To dealt with the dynamism in robot locomotion, integration of several aspect should be considered, external and internal sensory information, embodiment structure, cognitive model, and locomotion generator. Separation of the model might limit the generated dynamic behavior. Therefore, this proposed research is done by the multilateral interdisciplinary collaboration based on the integration of mechanical model, neuro-musculoskeletal approach to modelling the flow of data information, ecological psychology approach to build some systems, and the multi-scale systems approach by computer scientists to classifying complex system.

Developing multi integrated system increases the complexity exponentially. Dealing with scaling segregation is one way to realize a model for a complex system. We, therefore, classify the system into three scope, microscopic level, mesoscopic level, and macroscopic level, dealing with short term, mid term, and long term adaptation, respectively.

The whole-system model, shown in Fig. 2.2, represents neuro-cognitive locomotion system which considers not only internal sensory information but external sensory information as well. The diagrammed system integrates the cognitive model with behavior generation in all three of short-, medium-, and long-term adaptation.

Microscopic level implies short-term adaptation involves the response to environmental changes by controlling low-level signals. Leg swings are controlled directly, using both internal sensory information and also external perceptual information. Mesoscopic level implies medium-term adaptation involves responding to environmental changes at each footstep by changing the neural structure of the locomotion generator. The neural structure controls the motion pattern depending on the walking provision (sagittal speed, coronal speed, and direction of motion) from a higher-level model (path planning). Medium-term adaptation entails an Intention–Situation (I–S) cycle: the intention behind the behavior is decided depending on the situation. Furthermore, map reconstruction and localization based on topological structure will be developed to support the input of cognitive map in Macroscopic level. Macroscopic level long-term adaptation involves the model adapting by adjusting the intention (movement planning) in response to environmental conditions. Building cognitive map will be provided to inform possible coverage area of the robot. Then, it will be input of the motion planning model.

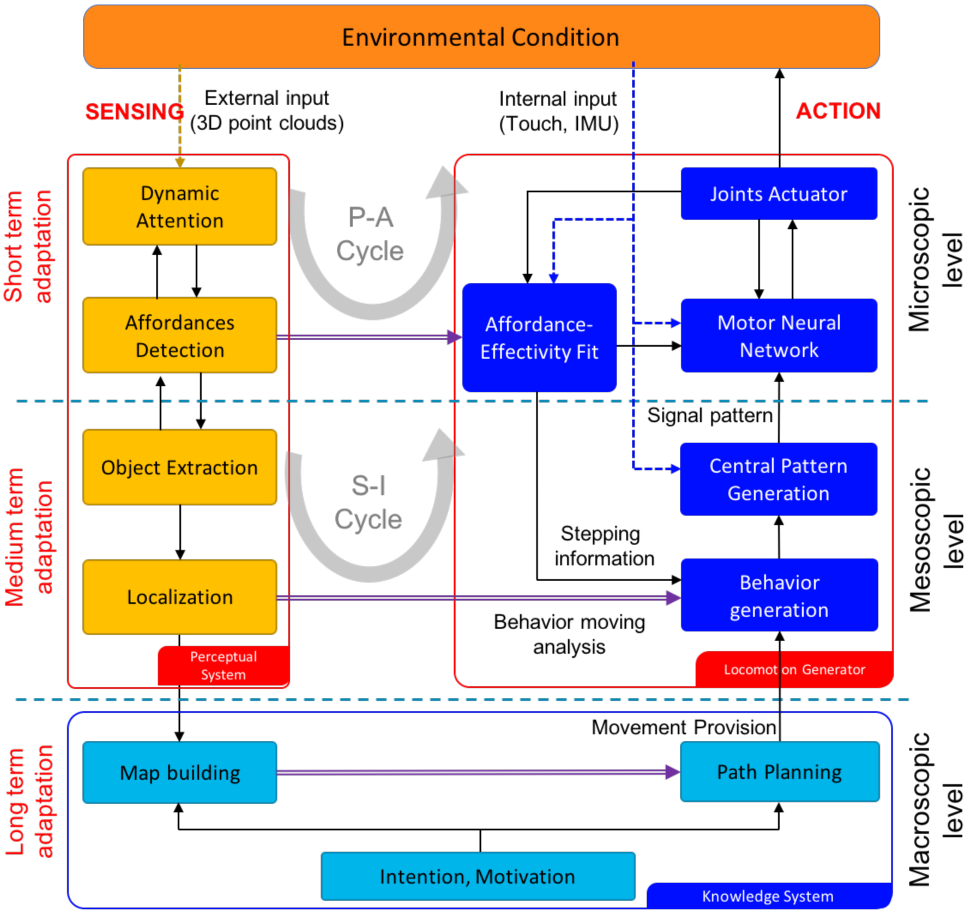


Figure 2.2: *Diagram of multi-scopic neuro-cognitive locomotion*

Chapter 3

Microscopic Adaptation in Locomotion Behavior

In this chapter, I focus on microscopic level implies short-term adaptation system as our novel contribution. The system diagram in microscopic level can be seen in Fig. 3.1 Short-term adaptation requires a direct response to detected changes in every time cycle. To achieve this, I use point-cloud data from the external sensors. First, to reduce data representation overheads, I use the dynamic attention model (DD-GNG) explained in Section 3.1.1 to generate a topological map model in a neural gas network with a dynamic node density. The network's node density represents the degree of attention to corresponding regions. The Dynamic Attention model outputs a $3 \times A$ matrix to represent the network's nodes, and a $A \times A$ matrix to represent its edges, where A is the number of nodes.

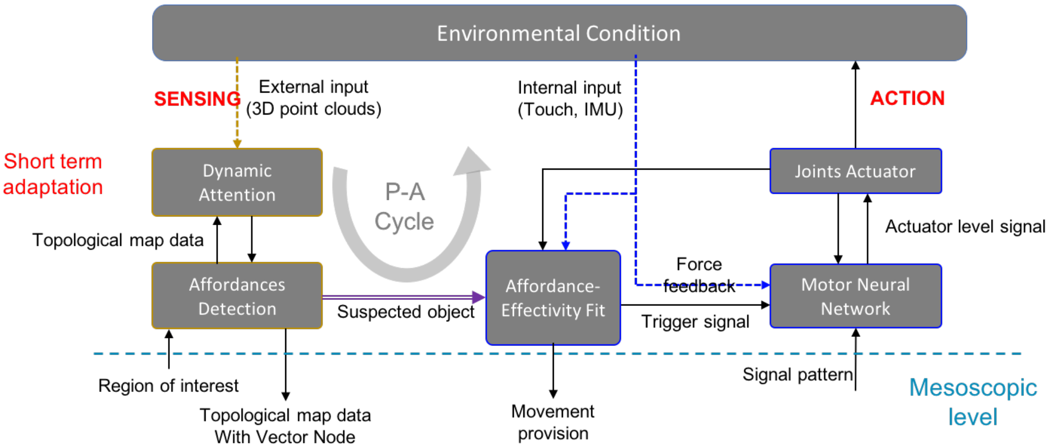


Figure 3.1: *Diagram of microscopic level of neuro-cognitive model*

After that, the dynamic attention matrices are interpreted by an Affordance Detection (see Section 3.3.2) system. Detected objects are represented by a group of triangular planes whose size represents the degree of stepping precision. The Affordance Detection system then outputs the size of each area, and an associated normal vector representing how safe that area is. If a nonhomogeneous normal vector for any area, the Dynamic Attention system increases the density of nodes there. For example, when the robot faces a suspected obstacle, the topological structure in the suspected area becomes denser so that more information will be collected for analysis. The Affordance result is processed also to detect objects for medium-term adaptation. Affordance results will be considered by the path planning system in the higher-level controller. Depending on the intention, motivation, and goal position given by a human operator, the path planner sets an overall speed and direction.

In order to generate appropriate action and integrate the Affordance Detector with the locomotion generator, I build an *Affordance Effectivity Fit* (AEF) process. The AEF determines whether an object affects the robot's immediate needs: when the object is close enough, The AEF will interrupt and control the motor neural network (lower level), the degree of interruption, and the swing direction. Swing generation, from a coupled neural oscillator, controls the angular velocities of joints in each leg. The stepping precision will be affected by the intention of the controller for positioning the feet in the target area. In some conditions, the AEF process will change the walking provision to encourage medium-term adaptation. Details will be explained in Section 3.3.

3.1 Dynamic Attention Model of External S.I.

The development of robotic sensory systems is increasing significantly in recent years. Robots are able to extract surrounding information from image data and/or point cloud data. However, for building the environment map and also shaping objects, 3D point cloud data is more efficient to be processed. Once facing the huge data, the process of the extraction may have a problem in computational cost. Therefore, topological map structure is one of good alternative for representing the raw data, in order to decrease the cost in the main process.

There are several topological structure generation model proposed by several researchers [237, 142, 291, 199].I developed topological map building based on Adaptive resonance theory (ART) [45, 44]. Toda el al, has built a real-time 3D topological map building using growing neural gas (GNG). However, it is difficult to shape the detail object [287] and the density of structure and number of nodes must be defined in advance. Therefore, I proposed a method for dynamic density topological structure building based on growing neural gas (DD-GNG) for specifying the detailed shape of an object in the attention area. The density of nodes in the topological structure will increase when the proposed segmentation model detects the speculated obstacle, while lesser nodes are generated (low density) in the safe area, such as a wall, flat terrain, and memorized objects.

I implemented the proposed DD-GNG on a 4-legged robot to detect a ladder in real-time for validation. In disaster area or hazardous environments, vertical ladders are often broken, fragile, and out of shape. Therefore, in order to perform ladder climbing, the robot has to always pay attention to the grasping area of the ladder. It requires a real-time ladder detection that able to continually track and estimate the position of rungs of the ladder.

Currently there are several researcher conducting ladder and stair detection, or object detection using deep learning models [124], [185, 58]. [124] applied a Convolution Neural network (CNN) for recognizing and localizing a staircase. The detection may failed if the detected staircase is different from the training samples. VOxNet [185] is developed based on CNN architecture integrating with a volumetric Occupancy Grid representation for object detection from RGBD and LiDaR point clouds. These methods suffer from high computational cost.

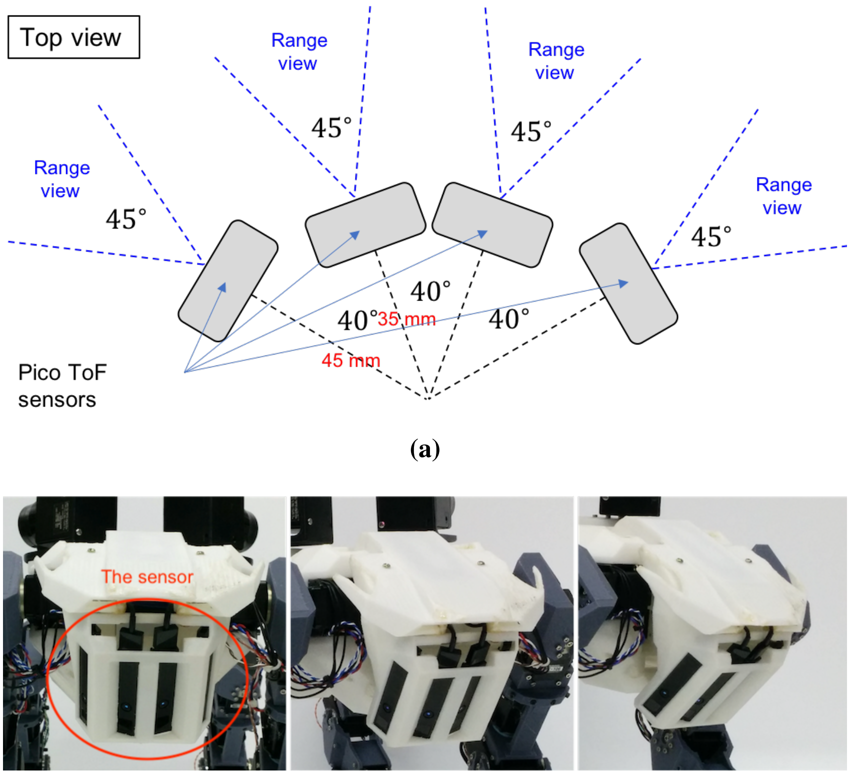
Some researcher implemented ladder detection for supporting their behavior model. Vaillant and Yoneda implemented vertical ladder detection for climbing behavior on a biped robot [297, 323]. The DRC-Hubo humanoid robot successfully performed the ladder climbing in real world environment [169]. However, the authors did not mention the process of ladder detection and I are uncertain if the robot continuously detects the ladder during movement or just one time detection before the move. Zhang et. al. [332] presented a planning strategy for the humanoid robot Hubo-II+ to generate multi-limbed locomotion sequences automatically for a given specification of a ladder. The proposed work is validated in simulation only and real-world implementation remained a concern.

The development of ladder observation has been proposed by several researchers [43, 96, 60]. Chen et. al. [43] developed a system to fuse vision and laser point cloud data for tracking ladder in 3D. The proposed work was validated in a humanoid robot ATLAS in the DARPA Robotics Challenge. However, if the tracking error is large, the visual tracker cannot be recovered and thus lead to tracking failure. In [96], the authors apply RANSAC algorithm with voxel grid filter to determine the ladder and its rung from 3D point clouds and represent them in lines. The method is suffer of high computational cost. Anna [60] implemented a supervised learning algorithm that trains a Markov random field (MRF) to segment 3D point clouds for detecting support surfaces of climbable structures such ladder. However, this system requires a pre-training process.. [287, 208, 10] proposed a GNG for generated topological map. The proposed method works well for time series 3D point cloud data by providing a rough segmentation. However, the work is unable to generate details topological structure of objects such as a ladder without providing safe and unsafe grasping.

The contribution of this section are i) a real time ladder detection method is developed for detecting and tracking rungs of a ladder from 3D point cloud data, meanwhile continuously tracking and updating the rungs condition and position with low computational cost; ii) the density of structure generated by the proposed method is continuously changing for adapting the speculated objects.

3.1.1 Dynamic Density Growing Neural Gas

Attention can often represent an animal's movement intentions. When intending to go straight ahead, the animal focuses in front, ready to react to suspected obstacles. On detecting an obstacle, attention is redirected to analyze the suspected obstacle in detail. If the obstacle is deemed harmless, the attention is moved away [42].



(a)

(b)

Figure 3.2: *Design of the 3D point-cloud sensor*

Our proposed model realizes the above attention mechanism for robot movement, implementing attention in the dynamic density of a growing neural gas. The gas density manifests in the number and distribution of nodes and edges throughout the space. The attention process entails controlling the density of data representation to focus on specific attention areas, being circles of a particular radius, centered on a suspected node. I use four Pico Flexx time-of-flight sensors, developed by PMD technologies [1]. Each sensor captures depth information only, covering a 45° × 62° field of view. The four sensors are arranged as depicted in Fig. 3.2 for a composite view covering 165° × 62°. The depth information that they gather is processed into a topological structure model using unsupervised learning. There are lots of unsupervised models such as Self-Organizing Map (SOM), Growing Cell Structure (GCS), Neural Gas (NG), etc. [139, 75, 74]. These models are not ideal for real-time processing, but the basic Growing Neural Gas (GNG) technique can be implemented for real-time topological map generation as proposed by Toda et al. [285].

Algorithm 1 Dynamic Attention Model

```
Init: generate two nodes at random position
    $c_{i,j} = 0\ (\forall_i, \forall_j), A = 1,2, r = 2, t = 0$
loop
    $\mathbb{P} \leftarrow$ raw data of proposed sensors
    $\lambda \leftarrow 0$
    for $i \leftarrow 1$ to $\max_\lambda$ do
        if $t \mod 2 = 0 \bigwedge$ obstacle detected then
            $V_\lambda \leftarrow$ a random of $\mathbb{P}$ in suspected obstacle
        else
            $V_\lambda \leftarrow$ a random of $\mathbb{P}$
        $\lambda \leftarrow \lambda + 1$
    DD-GNG main process ($V$) // (main layer)
    if $r > \gamma_r$ then
        Triangulation process ($\mathbf{c}, \mathbf{h}$)
        $\mathbf{L} \leftarrow$ Segmentation Process ($\mathbf{c}, \mathbf{h}$)
        $\mathbf{L}_{suspected} \leftarrow$ Suspected obstacle detection ($\mathbf{L}$)
        $\mathbf{L}^{(size)}_{suspected} \leftarrow$ Get suspected obstacle area ($\mathbf{L}$)
```

The topology-represented dynamic attentional process is grounded in the Dynamic Density Growing Neural Gas (DD-GNG) Algorithm (algorithm 1). In our system, the raw data ($\mathbb{P}$) around the suspected obstacle are processed separately.

The initialization process begins by generating two initial, random nodes. After that, in the main loop, V is generated as raw point-cloud input data. Then, the main GNG process is performed as presented in Algorithm 2, where h_i and w represent the n dimensional node value and weight value, respectively; A is composed as a set of h_i; C_{ij} and g_{ij} represent edge values and the age between the i-th and j-th nodes. The GNG algorithm chooses the nearest unit (winner) node as s_1 and the second-nearest node as s_2 from the set of nodes A. If the connection among them does not yet exist, then a connection $c_{s_1,s_2} = 1$ will be created. Every node has a degree of strength represented by parameter δ generated in Alg. 3 when the node is created. The node with the minimum utility value is removed from the topological structure if $E_u/U_l > k \cdot \delta_u$, where E_u is the local error variable of node u, U_l is the utility value of node l, δ_u is the degree of strength of node u and k is the parameter for representing degree of deletion. When the number of nodes (r) is more than the threshold (γ_r), then I perform triangulation and segmentation.

Algorithm 2 DD-GNG main process

```
for $t \leftarrow 1$ to $\lambda$ do
    Generate random input data $v_t$
    $s_1 = \arg\min_{i \in A} ||w \cdot (v_t - h_i)||$
    $s_2 = \arg\min_{i \in A \setminus s_1} ||w \cdot (v_t - h_i)||$
    if $C_{s_1,s_2} = 0$ then
        $C_{s_1,s_2} = 1$
    $g_{s_1,s_2} = 0$
    $E_{s_1} \leftarrow E_{s_1} + ||w \cdot (v_t - h_i)||$
    $U_{s_1} \leftarrow U_{s_1} + ||w \cdot (v_t - h_2)|| - ||w \cdot (v_t - h_i)||$
    $h_{s_1} \leftarrow h_{s_1} + \eta_1 \cdot (v_t - h_i)$
    if $C_{s_1,j} = 0$ then
        $h_j \leftarrow h_j + \eta_2 \cdot (v_t - h_j)$
        $g_{s_1,j} \leftarrow g_{s_1,j} + 1$
        if $g_{s_1,j} < g_{max}$ then
            $C_{s_1,j} = 0$
    if $(t \equiv \kappa) = 1$ then
        $u = \arg\max_{i \in A} E_i$
        $l = \arg\min_{i \in A} U_i$
        if $E_u/U_l < k \cdot \delta_u$ then
            $A \leftarrow A - l$
    $E_i \leftarrow E_i - \beta E_i (\forall i)$
    $U_i \leftarrow U_i - \chi U_i (\forall i)$
if $A <$ Max. Num. of Nodes then
    $r \leftarrow r + 1$
    $u = \arg\max_{i \in A} E_i$, $f = \arg\max_{C_{u,i}=0} E_i$
    $h_r = 0.5(h_u + h_f)$
    $E_u \leftarrow E_u - \alpha E_u$, $E_f \leftarrow E_f - \alpha E_f$
    $E_r = 0.5(E_u + E_f)$
    $\delta_r$ = generate the strength of node $(h_r)$
    $A \leftarrow A + r$
```

Algorithm 3 Generate the strength of node h_i

```
$\delta_i \leftarrow 1$
for $j \leftarrow 1$ to $Ob^{(num)}$ do
    if $Ob_j$ is suspected obstacle then
        $A_{min} = Ob_i^{(position)} - Ob_i^{(size)}/2$
        $A_{max} = Ob_i^{(position)} + Ob_i^{(size)}/2$
        if $h_i > A_{min} \bigwedge h_i < A_{max}$ then
            $\delta_i \leftarrow \delta_i + 1$
return $\delta_i$
```

3.1.1.1 Object Segmentation Process

In our model, information of object is represented using a collection of triangular facets and normal vectors, approximating the physical surface. Each facet has a degree of stepping stability calculated from the degree of safeness. The first step in developing this representation is to triangulate the surface, as illustrated in Fig. 3.3. After triangulation, the each triangle's normal vector is calculated by Eq. (3.1) as illustrated in Fig. 3.4. $\mathbf{n}_0$, $\mathbf{n}_1$, and $\mathbf{n}_2$ are the three DD-GNG nodes ($\mathbf{h}$) at the triangle's vertices.

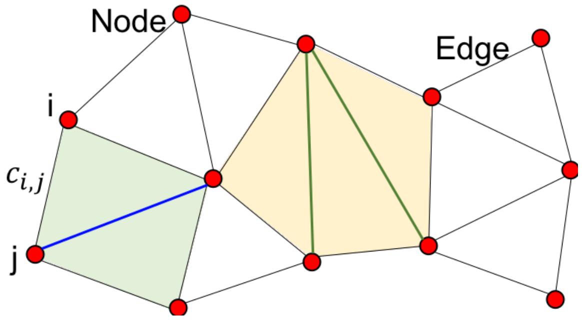


Figure 3.3: *Triangulation. Black lines are the edges generated by the proposed GNG, blue line is the supported edge of the light blue rectangular area, and the green lines represents supported edges of the pink pentagonal area.*

$$\mathbf{N_i} = \frac{(\mathbf{n}_0 - \mathbf{n}_1) \times (\mathbf{n}_0 - \mathbf{n}_2)}{||(\mathbf{n}_0 - \mathbf{n}_1) \times (\mathbf{n}_0 - \mathbf{n}_2||} \tag{3.1}$$

After calculating the normal vector for every triangular facet, I calculate the weight connection between one facet and the surrounding facets; each facet thus has up to three connections. The weight is represented by the angle between two normal vectors, calculated in Eq. (3.2).

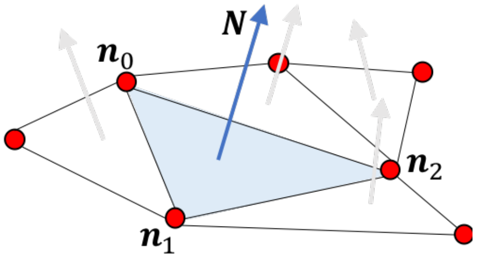


Figure 3.4: *Calculation of normal vector*

$$w_{ij} = \cos^{-1}\left(\frac{\overrightarrow{\mathbf{N}}_i \cdot \overrightarrow{\mathbf{N}}_j}{||\overrightarrow{\mathbf{N}}_i|| \cdot ||\overrightarrow{\mathbf{N}}_j||}\right) \tag{3.2}$$

After building the weight connections, I segment the facet collection into broader surfaces corresponding to walls, safe terrain, unknown objects. Segmentation takes place as follows:

1. Set the labeling ID parameter as 0 ($c_i = 0, \forall i \in B$), where B is the number of facets. Set the number of labels as 0 ($n_l = 0$). Set the number of facets in a set of labels as null ($L_a = 0$) and the set of vectors in the label as empty ($\overrightarrow{\mathbf{T}} = \{\}$). Set incremental value as 0 ($i = 0$).

2. If the label ID is 0 ($c_i = 0$), then a new label is added ($n = n+1, c_i \leftarrow n$), L_n++, $\overrightarrow{\mathbf{T}}_{n,L_n} \leftarrow \overrightarrow{\mathbf{N}}_i$.

3. Analyze the weight value w_{ij} between facet i and neighbouring facet j. If w_{ij} is lower than the threshold (ψ), then the ith and jth facets are deemed to be joined to each other, $c_j = c_i, L_{c_i} = L_{c_i} + 1$, $\overrightarrow{\mathbf{T}}_{c_i,++} \leftarrow \overrightarrow{\mathbf{N}}_j$.

4. If all facets are selected ($i = B$), then stop the segmentation process. Otherwise, select the next surface ($i = i+1$) and go to step 2.

In the segmentation process, I collect the number of labels (n), the number of facets in each label (L), and the normal vectors of the facets in every label ($\overrightarrow{\mathbf{T}}$) calculated by Eq. (3.3).

$$\overrightarrow{Ob^{(v)}}_i = \frac{1}{L_i}\sum_{j=0}^{L_i} \overrightarrow{\mathbf{T}}_{i,j} \tag{3.3}$$

The segmentation process generates both the object position ($Ob^{(\text{position})}$) and also the object size ($Ob^{(\text{size})}$). After that, I classify objects into four kinds: object, wall, safe ground, and obstacle. The ‘wall’ label is selected when the average angle between facets and the vertical unit vector $v = [0,0,1]$ is close to 90 degrees. ‘Safe ground’ is chosen when the facets’ gradient is within a given range of safe slopes. Otherwise, the object is classified as an obstacle that needs further analysis.

Before the classification, the orientation of every surrounding surface needs to be calculated by Eq. (3.4), from which to calculate safeness by comparison with the vertical unit vector.

$$\gamma_i = \cos^{-1}\left(\frac{\overrightarrow{Ob^{(v)}}_i \cdot \overrightarrow{\mathbf{N}}_{(ver)}}{||\overrightarrow{Ob^{(v)}}_i|| \cdot ||\overrightarrow{\mathbf{N}}_{(ver)}||}\right), \mathbf{N}_{(ver)} = \{0,1,0\} \tag{3.4}$$

Whenever the density of perceived affordance d, calculated by Eq. (3.5), is lower than the threshold (υ), then it will show the integration between affordance and attention. If $d > \upsilon$, then the perception will change the robot’s behavior.

$$d = \text{Number of nodes in object/Object size} \tag{3.5}$$

3.1.2 Experimental of Dynamic Attention Model

The proposed model is applied in the four legged robot used in our previous research [248]. The robot is equipped various sensors for example, Quad Time of Flight (ToF) sensor as shown in Fig. 3.2 for detecting rungs position; set of force and touch sensors in the end effectors for grasping mechanism, and an inertial measurement unit (IMU) for pose analysis. Quad Pico sensors is used for sensing the environment and generating 3D point cloud. The robot also implements the locomotion system proposed in the previous research [247]. The parameters setting of the DD-GNG is tabulated in Table 4.1.

Table 3.1: *DD-GNG parameters*

λ	κ	η_1	η_2	g_{max}
300	10	0.08	0.008	100
α	β	χ	μ	k
0.5	0.005	0.005	0.5	1000

ig. 3.5a shows the raw sensory data generated by the sensor. Then, the segmentation model is processed as shown in Fig. 3.5b. The obstacle detection is representing with a red color box (Fig. 3.5c). Nodes that fall inside the red color box has a higher strength value than others. The strength value of each node is used for the next process. After several time steps, the number of nodes in the red box area was increased.

As the result depicted in Figs. 3.10a and 3.10b, comparing with the previous model of GNG, the proposed DD-GNG is able to generate the dynamic density of the topological

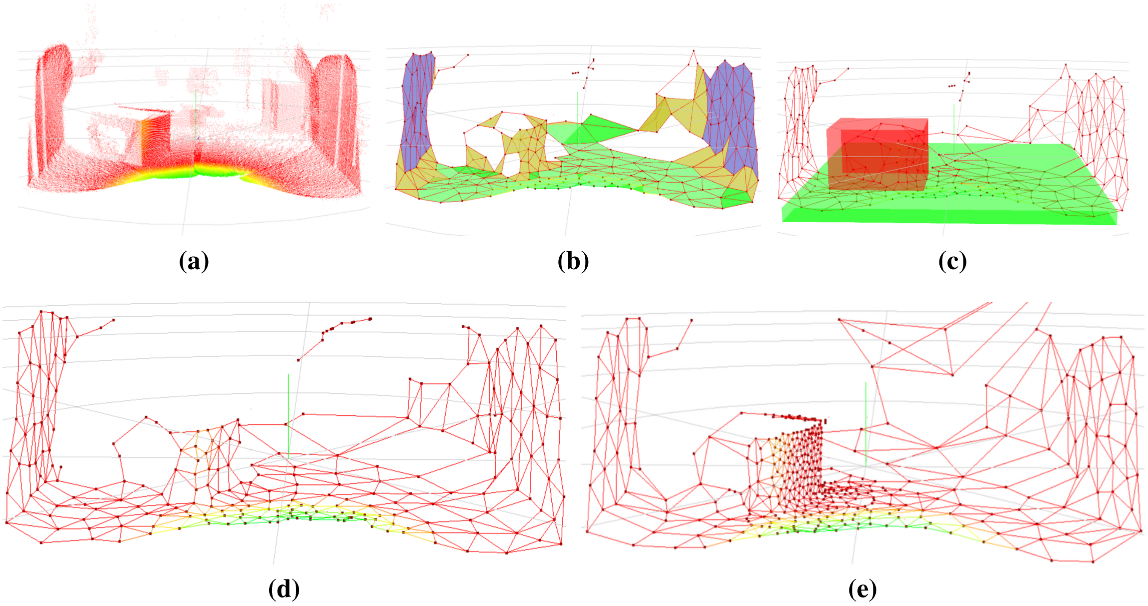


Figure 3.5: *Results of topological structure using DD-GNG with an obstacle (a) Raw point clouds data (238530 3D points) (b) Result after segmentation process (c) Object detection differentiate between terrains and obstacles (d) Generated structure without dynamic density (e) Generated topological structure with dynamic density.*

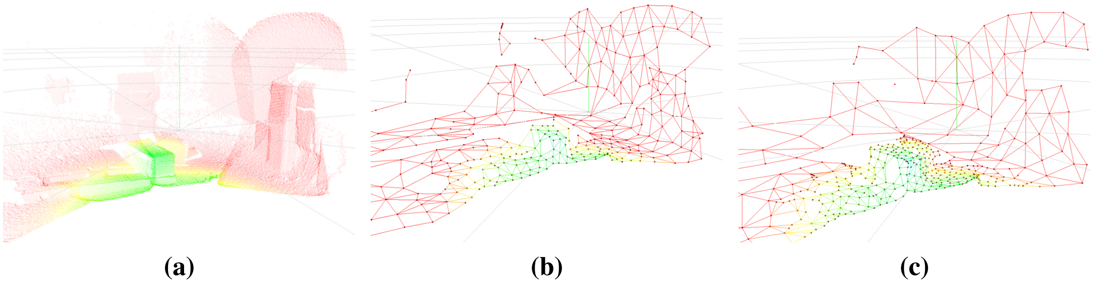


Figure 3.6: *The result of second sample data (a) Raw point clouds data (b) generated structure without dynamic density (c) generated topological structure with dynamic density.*

structure. In Fig. 3.5e, I show that the number of nodes is denser in the speculated area, while the other GNG shown in Fig. 3.5d has lower number of nodes.

Results showed that the DD-GNG is more effective for determining the speculated object. Thus, the higher level processor can process the topological information directly without going through multi-layer topological map building. Thus, the computational cost is lower.

3.1.2.1 Discussion on Dynamic Attention model

In this section, I proposed a dynamic density topological structure generator based on growing neural gas, termed as DD-GNG. In DG-GNG, the density of nodes in speculated areas or objects is increased automatically. The segmentation model can differentiate the safe terrain, wall, and speculated obstacles. Once speculated obstacles are detected, the proposed model automatically increase nodes density around that area and the information is used for calculating the strength of nodes. Nodes around the speculated object area have a higher value than the safe area (flat terrain, wall). Nodes with higher strength retain longer iterations and new nodes will be generated around these nodes.

3.1.2.2 Comparison of attention model using topological structure

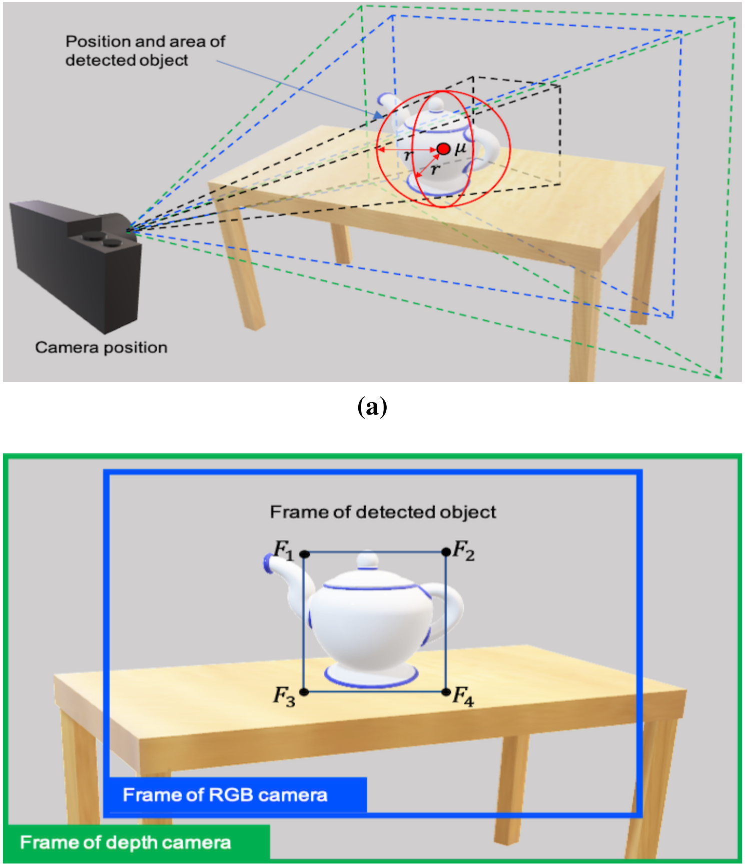
Attention can be represented by any parameter, depending on the data processed. In our case, I use only time-of-flight sensors for the external information; these generate 3D point-cloud data. I therefore chose the topological map model for optimal data representation. However, in the existing topological map building process offers no way to control node density in only a localized area. Our contribution addresses that through a dynamic density topological map-building process based on a growing neural gas. I put a strength parameter into every node to indicate the importance of each node in the topological structure. This parameter will be defined based on the segmentation process in the previous time. The experiment in Section 3.1.2 shows that the improved dynamic attention process can clarify obstacles in front of the robot by increasing the number of nodes.

Other topological generator systems such as Self-Organizing Map (SOM), Growing Cell Structure (GCS), Neural Gas (NG), cannot increase node density in localized areas. Therefore, they need to increase the node density over the entire map in order to clarify even localized objects. [139, 75, 74]. Compared with other multi-density topological maps, such as multi-layer growing neural gas (ML-GNG) [285], our improved system may decrease the processing time by as much as 70 percent (ML-GNG = $3.1567 \times 10^{-4}\ s$, DD-GNG = 1.0255×10^{-4}). The localized attention-focusing process has also been proven to decrease the computational cost.

3.1.3 Implementation for Grasping Affordance Detection

The proposed algorithm aims to detect the affordance of grasping. The affordances serve seven dimension grasping position. It uses a combination of RGB camera and depth camera as the sensory information. RGB camera is used for detecting the desired object and Depth camera for detecting the affordance of grasping. However, the object detection has been done by Redmon et al. [224]. I use their YOLO v3 Framework [225] for object detection.

The detected frame position and size served by object detection part will be matched to the depth camera frame and position. After that, I calculate the average position ($\precneqq$) of the depth points in the selected frame ($\mathbb{X}$) and calculate the radius of the detected area by



(a)

(b)

Figure 3.7: *The illustration for generating the object location and area (a) Perspective view of object detection (b) Frame view shows the four frame point of object detection and comparison of depth and RGB camera's frame.*

calculating as $r = 2 * \sigma$, where σ is the standard deviation of $\mathbb{X}$. The illustration can be seen in Fig. 3.7. The spherical area information will be used for the dynamic density topological map building. The main algorithm of our proposed grasping affordance detection can be seen in Algorithm 10. In the first state, object detection is conducted. After that, it generates the topological map using dynamic density growing neural gas. Then, the grasping affordance detecting will be processed.

Algorithm 4 Dynamic Topological Structure

```
Initialization
loop
    $\bar{}, r \leftarrow$object detection framework ()
    Generate dynamic density topological structure ()
    //grasping affordance detection
    for $i \leftarrow 1$ to $\max_{\lambda}$ do
        Generate new possible grasping position
        Evaluating new possible grasping position
        Update and evaluate current grasping position
        Calculate the rank of grasping position
```

3.1.3.1 Dynamic Density Topological Map Building

The topological map building aims to decrease the number of data information by representing all raw data input with several points and edge information. I used GNG based model with dynamic density. It can be more clarifying small and detail object than other common GNG. The previous development of our GNG has been implemented for real-time topological map building [287]. In this model, I add a parameter in every node for representing the strength of the node. The strength parameter of kth node (δ_k) will be calculated in Eq. (3.6),

$$\delta_k = \frac{e^{-(r+\|h_k-\mu\|)}}{1+e^{-(r+\|h_k-\mu\|)}} \tag{3.6}$$

where h_k, r, and μ are the kth 3D node's position, the radius of sphere, and 3D position of sphere.

Data collection ($\mathbb{V}$)in the proposed DD-GNG is also affected by the spherical area generated by object detection. The data will be randomly collected with different search space area. When t is an odd value, the data will be generated from all raw data ($\mathbb{P}$). When t is an even value, the search space is only inside the sphere(r, μ) area. The data collection of DD-GNG can be seen in the Algorithm 5.

Algorithm 5 Dynamic Topological Structure

```
Init: generate two nodes at random position
    $c_{i,j} = 0\ (\forall_i, \forall_j), A = 1,2,\ r = 2, t = 0$
loop
    $\mathbb{P} \leftarrow$ raw data from depth sensors
    $\mathbb{Q} \leftarrow$ raw data inside $Sphere(r, \mu)$
    $\lambda \leftarrow 0$
    for $i \leftarrow 1$ to $\max_\lambda$ do
        if $t \mod 2 = 0 \bigwedge$ object detected then
            $V_\lambda \leftarrow$ a random of $\mathbb{Q}$
        else
            $V_\lambda \leftarrow$ a random of $\mathbb{P}$
        $\lambda \leftarrow \lambda + 1$
    DD-GNG main process ($V$)
    Calculate discount rate ()
    $t \leftarrow t + 1$
```

Furthermore, the strength parameter will effect to utility rate calculation. The utility value of vital nodes will be slower decreased than the utility value in weak nodes. The calculation can be seen in Algorithm 6, where E_i and U_i are the GNG error value and GNG utility value of ith nodes. The node with the utility value lower than the defined threshold is removed from the topological structure, $E_u/U_l > k$. The utility value will effect to the node deletion and node addition. A weak node will easy to be deleted and strength node will easily produce a neighbor node.

Algorithm 6 Calculate discount rate

```
for $i \leftarrow 1$ to node's number do
    $E_i \leftarrow E_i \times \upsilon_E$
    $U_i \leftarrow U_I \times \upsilon_E / \delta_i^2$
```

3.1.3.2 Detection of Grasping Position

The proposed detection of grasping detection uses the topological structure generated by DD-GNG as the input data rather than the raw data. It will only consider the vital nodes with the strength value bigger than the defined threshold. The proposed detection will detect multiple possible grasping position by considering the gripper embodiment. The gripper embodiment composed as the tightness of the gripper, maximum and minimum gripping can be freely defined.

Generating Possible Grasping Position In order to find the perspective grasping position, two strong nodes (h_{n1}, h_{n2}) will be randomly generated. After that, the nodes will be evaluated by considering the edge connectivity and the density ratio of inlier and outlier nodes surrounding the pivot from n_1th and n_2th nodes. I calculate the evaluation value as computed as Eqs. (3.7) and (3.8), where c_i and c_o are the number of strong nodes in the inlier and outliers area respectively. In Eq. (3.8), when the evaluation value higher than the threshold λ_Θ, then the generated possible gripping position will be added. Otherwise, the detected will be deleted. The illustration of accepted or rejected generation is shown in Figs. 3.8. The node detection and the length of the pivot should not longer the the tightness of gripper (l_{max}), see Fig. 3.8a.

$$\Theta = \frac{(c_i)/(1+c_o)}{||\mathbf{n}^{(2)} - \mathbf{n}^{(1)}||} \tag{3.7}$$

$$\text{Eval. result} = \begin{cases} \text{Accepted}, N_g ++ & \text{if } \Theta \geq \lambda_\Theta \\ \text{Rejected} & \text{otherwise} \end{cases} \tag{3.8}$$

Updating Grasping Position This process aims to correct the axle position of grasping $\bar{\mathbf{n}}$ in the center position of node's members. The illustration of this process and the parameters used can be seen in Fig. 3.9. I will consider and calculate only the nodes surrounding the axle of the grasping position. First, I will calculate the similarity of the nodes using Eq. (3.9),

$$s_{i,k} = \frac{||\mathbf{v}_i - \mathbf{v}_k|| \cdot |\theta_i - \theta_k|}{||\mathbf{d}_i - \mathbf{d}_k||} \tag{3.9}$$

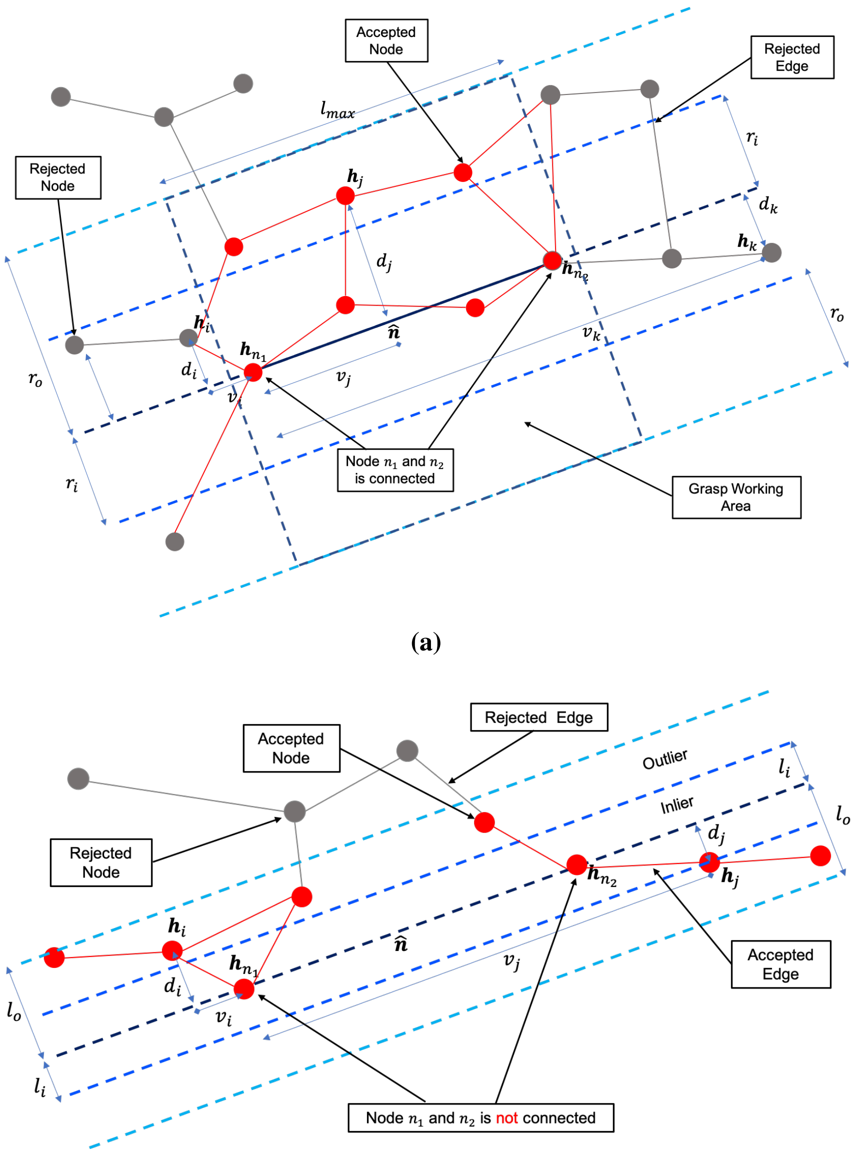
where v_i and d_i is the parallel and perpendicular distance between ith node and the axle. (see Figs. 3.9a and 3.8a). Parameter $s_{i,k}$ represents the degree of similarity between ith and kth nodes. If there are same nodes, then I will only consider the nearest nodes.

The alteration movement of starting axle node and end axle node are calculated in Eqs. 3.10 and 3.11, where c_i is the number of node in inlier area.

$$\Delta e_1 = \frac{1}{c_i} \sum_{j=\Re\rightarrow N_i}^{j=0} \frac{||\mathbf{v}_j||}{||\mathbf{v}^{(2)}||} (\mathbf{r}_j - \mathbf{v}_j) \tag{3.10}$$

$$\Delta e_2 = \frac{1}{c_i} \sum_{j=\Re\rightarrow N_i}^{j=0} \frac{||(\mathbf{v}^{(2)} - \mathbf{v}^{(1)}) - \mathbf{v}_j||}{||\mathbf{v}^{(2)}||} (\mathbf{r}_j - \mathbf{v}_j) \tag{3.11}$$

N_i is the number of nodes involve in ith detected rung. After calculating the error position of



(a)

(b)

Figure 3.8: *Illustration of topological structure (a) Generated nodes n_1 and n_2 is considered as a candidate of possible grasping position. The figure show the parameter involved for evaluation and position correction. The parameter l_{max} represents the tightness of the gripper. (b) Illustration of evaluation when generated nodes n_1 and n_2 is not connected. It is not considered candidate.*

detected grasping position, the start and end points are updated using Eqs. 3.12 and 3.11.

$$\mathfrak{R} \to h_1 = (\mathfrak{R} \to \mathbf{R}_{h_1} - \mathbf{v}^{(1)}) - \varepsilon \Delta e_1 \tag{3.12}$$

$$\mathfrak{R} \to h_2 = (\mathfrak{R} \to \mathbf{R}_{h_1} - \mathbf{v}^{(2)}) - \varepsilon \Delta e_2 \tag{3.13}$$

h_1 and h_2 are the start and end point of the detected rung.

Calculating grasping direction In order to get simple assuming, I consider the average vector of the associated node from the axle of grasping detection. The direction of the grasping vector of jth grasping position is calculated in Eq. 3.14. Parameter i indicates the id of the associated neuron. N is the number of neurons associated with a certain grasping position. The calculation will be affected by the degree of similarity to other nodes ($s_{i,k}$).

$$\theta_j = \sum_{i=0}^{i<N} \sum_{k=0}^{k<N} (s_{i,k}) \cdot \frac{\mathbf{v}_i}{N} \tag{3.14}$$

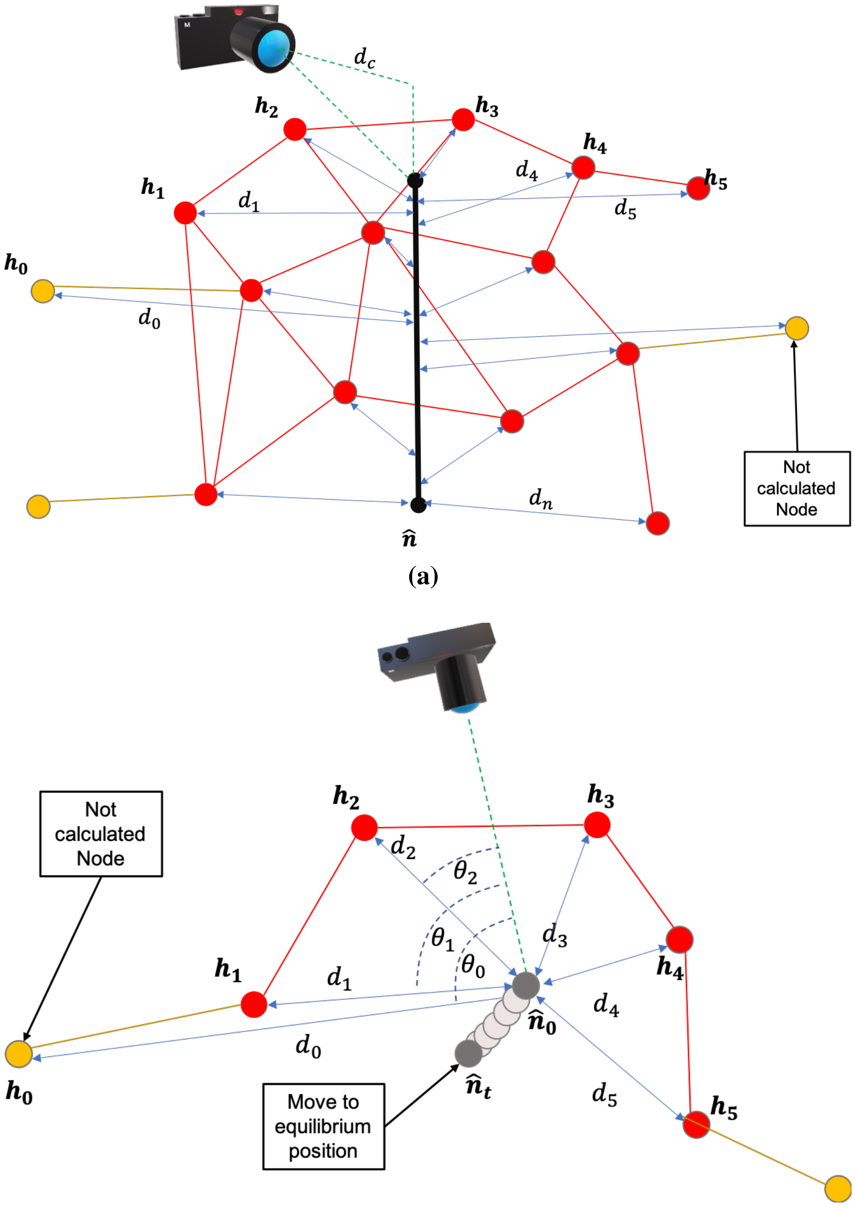
3.1.3.3 Experimental result of grasping ladder detection

These experiments show the effectiveness of our proposed dynamic density topological map building combined with visual information representing the shape of the object. After that, I will show the grasping affordance detection for static object and moving object. I used standard Web camera as the visual sensor and used four time-of-flight sensors the depth sensors. Each of the sensor coverage range of $45^0 \times 62^0$. Therefore, they can cover the view of $165^0 \times 62^0$. I perform the experiments in computer NUC i5 with 8 GB of RAM without any GPU installed.

3.1.3.4 Result of Dynamic Density Topological Map

Before testing the grasping affordance detection, I will test the proposed topological map building combined with object detection framework. The parameter setting of the DD-GNG is adopted from the basic idea in [287]. There are 300 points from all raw data generated randomly is processed in the proposed GNG.

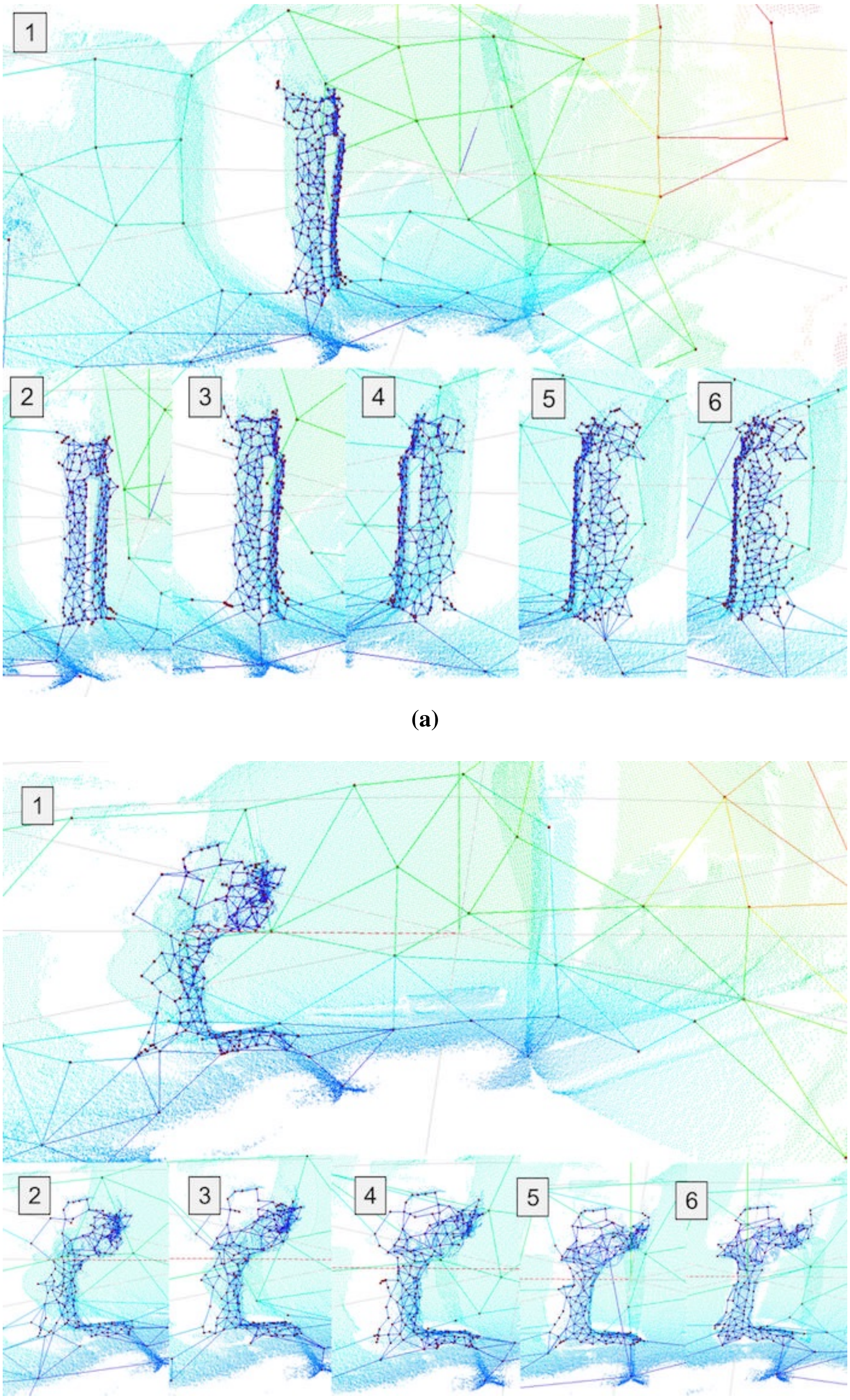
There were 2 moving objects tested, a bottle and a radio remote control. The result is shown in Fig. 3.10. The real-time performance is achieved. By increasing the density of structure in the object area, the object's shape can be clarified. It required 0.003 seconds for executing the topological map building. Therefore, it can be used for real-time detection. However, the computational cost can increase and decrease depending on the number of node and parameter λ in the topological map.



(a)

(b)

Figure 3.9: *The illustration for generating the object location and area (a) Perspective view of object detection (b) Frame view shows the four frame point of object detection and comparison of depth and RGB camera's frame.*



(a)

(b)

Figure 3.10: *The snapshot of the proposed dynamic density topological map building in moving object. Small dots represent the raw data of the depth sensor and the black dots with edges represent the generated topological map. (a) a bottle. (b) a radio control*

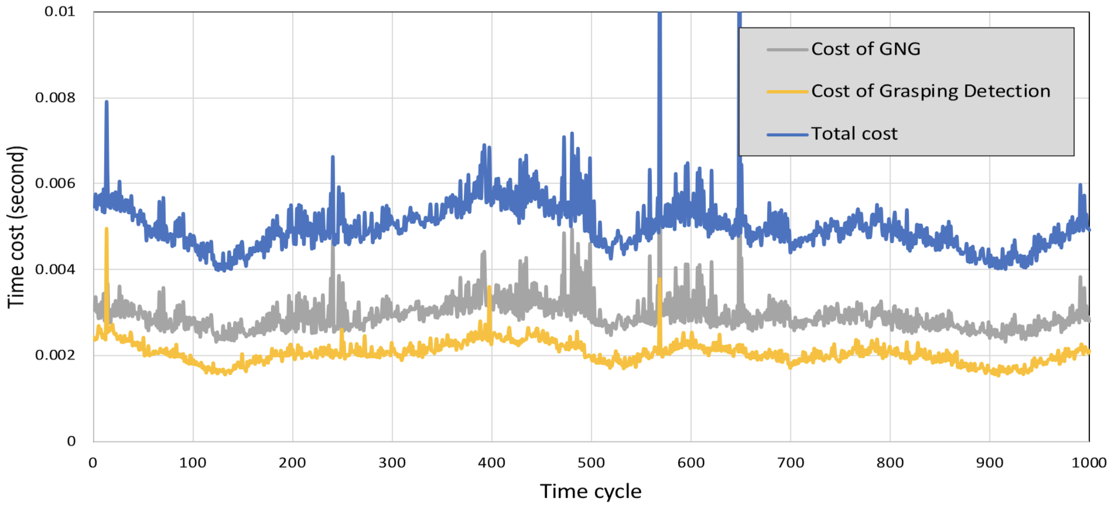


Figure 3.11: *The graph of time consuming during the proposed model performance*

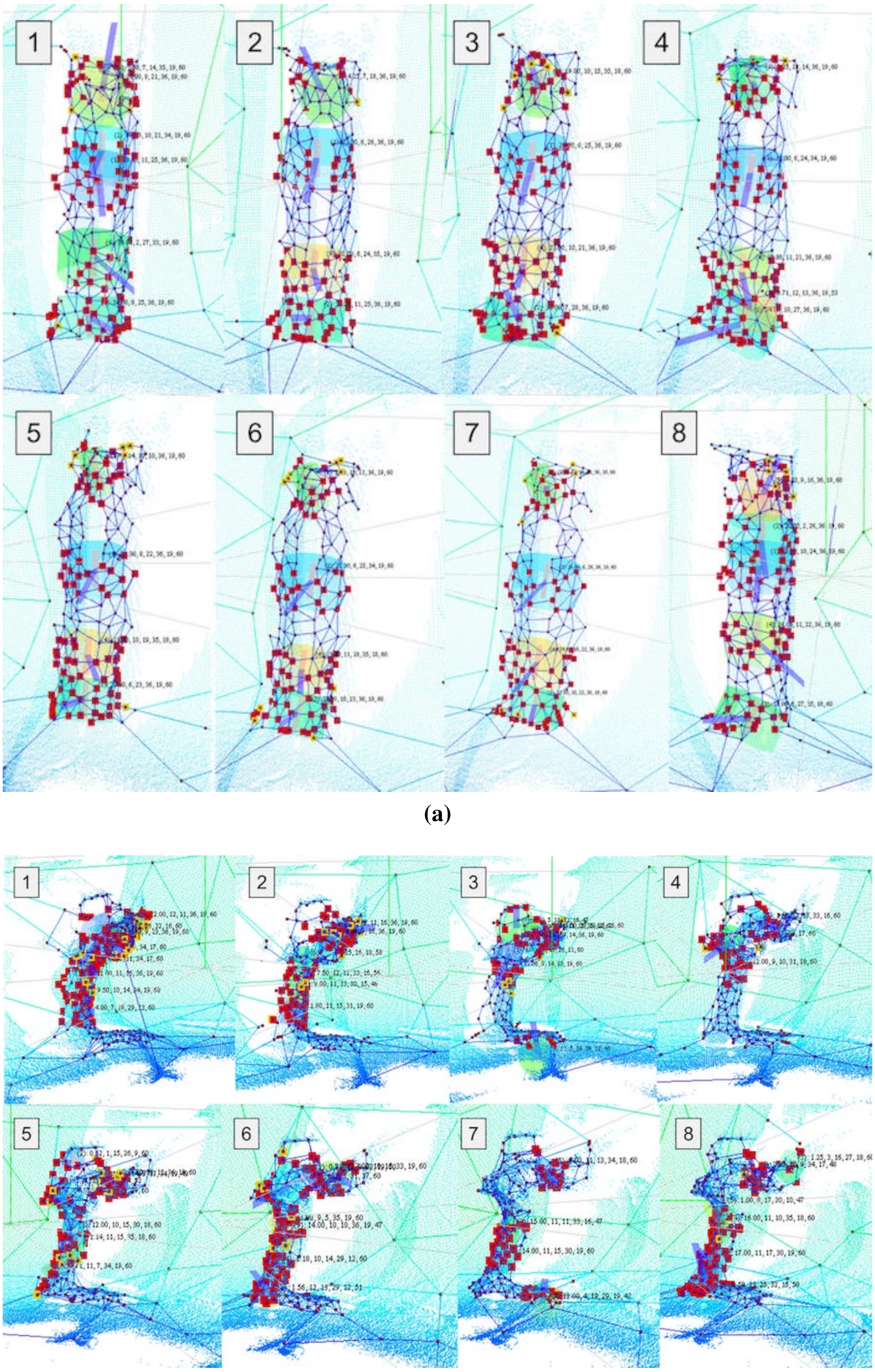
3.1.3.5 Grasping Affordance Detection

In order to prove the effectiveness of the proposed model, I tested the detection model in static objects and moving objects. In these experiments, there were three objects used as the sample, which are screw-driver, radio remote control, and a bottle. The experiments for the static object are shown in Fig. 3.13. There are several possible grasping positions in the radio remote control and a bottle. However, there is only one possible grasping position in the screwdriver. The size of the object influences the number of detected possible grasping position. The defined gripper size also affected. If the tightness of the gripper is small, then the possible grasping position will be generated more. The grasping affordance detection model requires around 0.002 seconds for executing. However, this can increase and decrease depending on the number of detected possible grasping position.

In order to prove the efficiency of computation, I tested the proposed model for detecting a moving object. Fig. 3.12 shows a snapshot of successful performances.

3.1.3.6 Discussion Implementation of Dynamic Attention on Grasping Affordance Detection

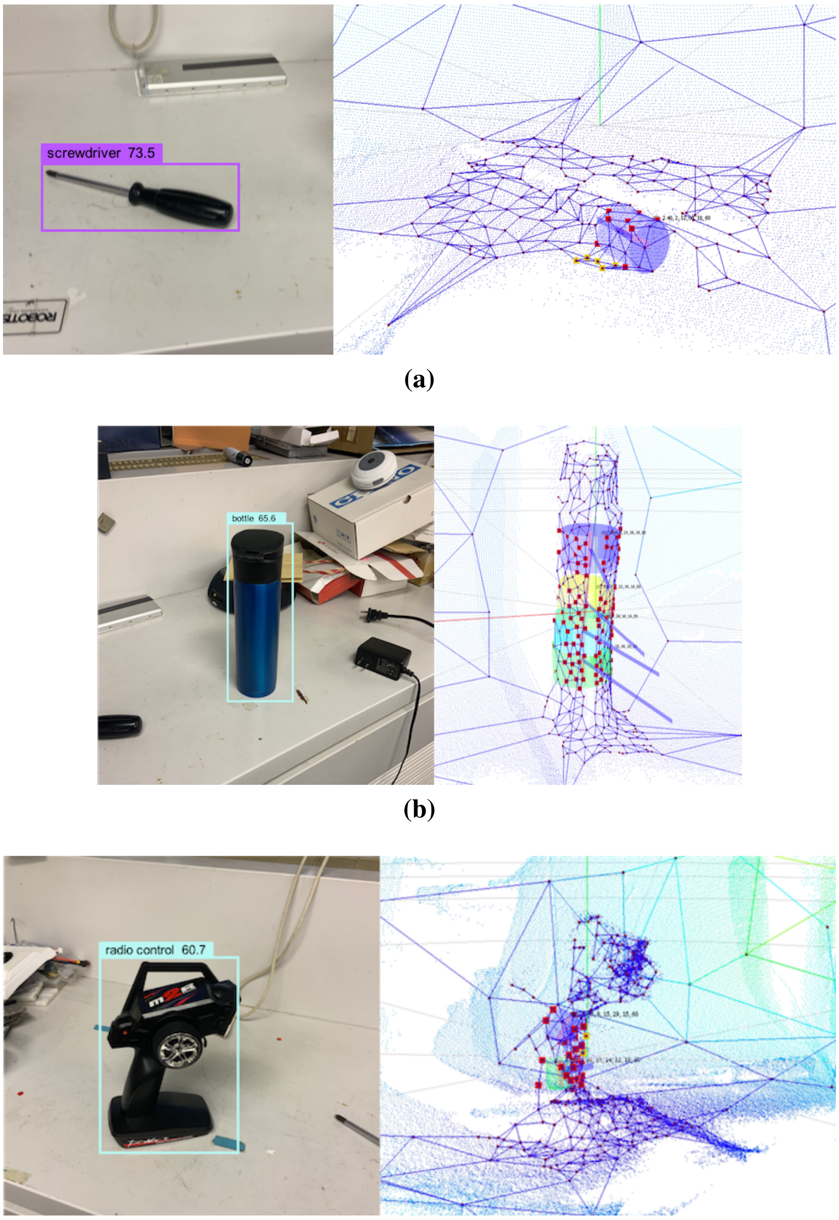
I have proposed the real-time object affordance detection for robot grasping. The sensory input is composed as the combination of visual information and depth information. However, it works separately, visual information for detecting the desired object using a common object detection framework. Our proposed work mostly occupy the depth of sensory information. I propose the dynamic density topological map building for decreasing the number of depth data representation. I add strength parameter in every node and give higher value where the position is approaching the center of the selected frame detected by object detection. The



(a)

(b)

Figure 3.12: *The snapshot of the proposed grasping affordance detection for moving object. (a) a bottle. (b) a radio control*



(a)

(b)

(c)

Figure 3.13: *The left figures are the result from object detection framework. The right side figures are the snapshot of the proposed grasping affordances detection for static object. There are 7 information provided every detected grasping position. 3D position of tube, dimension of tube represents the tightness of the grasping, and 3D Blue vector represents the grasping direction. (a) a screw driver (b) a bottle (c) a radio control.*

novelty of the proposed model has been shown in the experiment Section IV-A. The proposed topological map generator can clarify the desired object by increasing the density of node specific on the object area. This model can solve the obscurity of a small object by using topological map building with the low computational cost. DD-GNG required around 0.003 seconds for processing the data. Therefore, this model can be used for other implementation instead of object grasping detection for achieving lower computational cost.

In this section, the topological map generated by proposed DD-GNG is used for grasping detection. The affordance detection will serve a set of seven-dimension gripping information (3D location, 3D Rotation, and gripping size). The experiment in section IV-B shows the effectiveness of the proposed model detecting the grasping affordances of the desired object. There are several possible grasping positions, each grasping position is represented by a tube with an arrow. The tube has 4 information which is the 3D position, and the diameter represents the tightness of the grasping. The arrow represents the posture direction of the grasping. It also succeeded in performing detection for moving object. Comparing with current grasping detection, the proposed model has a lower computational cost. Therefore it can detect real-time moving objects.

3.2 Neural Primitive model with Sensorimotor Coordination

This research responds to the current challenges facing developing CPG for quadruped locomotion, presenting an efficient, robust CPG model that is dynamically integrated with sensory feedback to generate various gaits while also considering leg malfunction compensation without involving too many parameters. The model uses two sensorimotor-coordinationbased mechanisms [233, 149]: 1) because sensory feedback is critical for adjusting CPG modulation, proprioceptive signals from the leg force and swinging phases are integrated with rhythm-generator neurons (RG) in the CPG; 2) because leg malfunction affects the integration of sensory feedback and RG, a neuron representing a certain injured leg sends a signal and influences the effect of the sensory signal on RG. The proposed models novel contributions follow:

1. Developing a robust CPG model with an efficient structure that can generate various gait patterns.

2. Integrating sensory feedback that substantially impacts the CPG-modulation output.

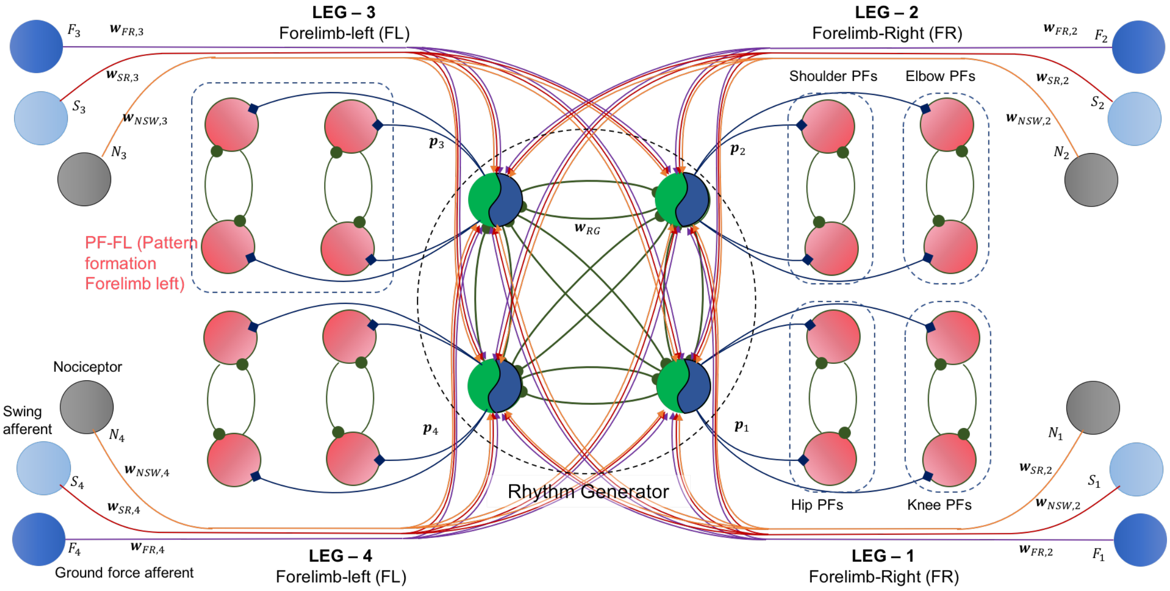


Figure 3.14: *Design of the single rhythm CPG model with two-layered CPG. Every leg has three sensory-feedback-to-all RG, a force sensor, a pain receptor, and a swing sensor*

3. Considering dynamic sensorimotor integration to compensate leg injuries at various speeds.

Furthermore, to demonstrate the proposed models effectiveness, I have used a cat-like robot that performs in various contexts, including in different terrains, at different speeds, and with leg injuries.

This section is organized as follows: the proposed CPG model with sensorimotor coordination is presented in Section 3.2.1; the strategy for finding the optimum structure is delineated in Section 3.2.2; Section 3.2.3 presents the experimental results; finally, in Section 3.2.4, I conclude the research and discuss the prospects of the proposed model.

3.2.1 CPG with Sensorimotor Coordination

The CPG model dynamically generates locomotion patterns. In cat studies, two-layered CPG has been observed, incorporating rhythm generators (RG) and pattern formation (PF) [239]. I have designed a single-model CPG where one RG represents one legs movement pattern, and one PF neuron represents the activation of one muscle. Since each leg uses four muscles (flexor and extensor muscles of the hip and knee joints), there is one RG neuron and four PF neurons for each CPG structure. Our model uses two CPG pairs; every pair represents the forelimb and hindlimb for another pair. Fig. 3.14 shows the complete design of the proposed CPG model.

Rhythm-generator neurons generate oscillation signals responsible for controlling the patterns of a certain leg. All of the RG are interconnected and control the timing of the legs swing action. I used the Matsuoka neural-oscillator model to generate a dynamic signal. The inner state of the RG neuron is as follows:

$$\tau\frac{\mathrm{d}}{\mathrm{d}t}x_i = \left(\upsilon_i - x_i - \sum_{j=1}^{n}\mathbf{w}_{RG,ij}y_j + \alpha_i - bv_i\right)(\tau_f * S_{STIM}) \tag{3.15}$$

The received-inhibition effect of its self-adaptation (v_i) and the signal from other RG neurons (y_j) influenced by synaptic weight (w_{ij}). y_i is calculated as $y_i = \max(x_i, 0)$ and v_i is calculated as follows:

$$T\frac{\mathrm{d}}{\mathrm{d}t}v_i = (y_i - v_i)(\tau_f S_{STIM}) \tag{3.16}$$

The inner-state and self-adaptation effects are respectively influenced by time constants τ and T. The RG neuron is also influenced by sensory feedback (α_i) from ground-force afferent, swing-related afference, and nociceptors.

$$\alpha_i = \alpha_{i,0} + \sum_{j=1}^{n}(F_i\mathbf{w}_{FR,ij}) + \sum_{j=1}^{n}\left(G_{STIM}S_i(\mathbf{w}_{SR,ij}N_j\mathbf{w}_{NS,ij})\right) \tag{3.17}$$

where, $\alpha_{i,0}$ is the basic stimulation of the ith neuron, $\mathbf{w}_{FR,ij}$ and $\mathbf{w}_{SR,ij}$ are the synaptic weights of the force afferent (F_i) and the swing-phase afferent (S_i) of the ith leg to the jth RG neuron. Swing-afferent feedback is influenced by the nociceptor (N_i); its weight is represented by $\mathbf{w}_{NS}$. The nociceptor (N_i) is a pain receptor that detects the condition of leg damage and sends damage stimuli to RG neurons, affecting sensory stimulation from the ground-force afferent to the RG neurons and from the swing-related afferent to the RG neurons. Parameter G_{STIM} is the gain parameter controlling the relationship between speed stimulation S_{STIM} and the sensory network. The gain parameter is assumed to differently affect the forelimbs and hindlimbs. Based on preliminary tests, the G_{STIM} for hindlimbs is calculated as $G_{STIM} = \mu_{HL}/(1 + \exp(-\lambda_{HL}S_{STIM} + \eta_{HL}))$ and the G_{STIM} for hindlimbs is calculated as $G_{STIM} = \exp(\log(\mu_{HL})(\lambda_{FL}S_{STIM} - \eta_{FL})^2)$

Given the RG neurons control the movement-pattern phase for each leg, they transmit the firing signal (p_i) to all PF neurons in a certain leg to activate swing behavior. This is calculated as follows:

$$p_i = \begin{cases} 1 & \text{if } (y_i + h_i^{ref}) \geq \Theta \\ 0 & \text{0therwise} \end{cases} \tag{3.18}$$

where Θ is the firing threshold. The value of h_i^{ref} is calculated as follows:

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } p_i(t-1) = 1 \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & \text{otherwise} \end{cases} \tag{3.19}$$

When the RG signal is fired, R is subtracted from the $h_i^{ref}(t)$ value of neuron i. $R > 0$, ^{ref} is a discount rate of h_i^{ref}, and $0\gamma^{ref}1$. The firing value of the ith RG neuron $p_i(t)$ is generated for all PF neurons in the ith leg.

Upon receiving the firing signal from RG neurons (p_i), PF neurons perform the swing behavior by activating muscle stimulation (see Appendix B). The PF neurons control the timing of each muscle contraction. The activation signal generated by PF neurons is calculated as follows: (To control the speed, speed stimulation is adjusted from parameter S_{STIM}, which influence Eqs. (5.13) and (3.17))

$$\theta_{i,k}(t) = e^{\left(\log(0.5)\times\left(\frac{|PF_i-\mu|^3}{(\mu\times w)}\right)\right)} \tag{3.20}$$

where, PF_i neurons are calculated as $PF_i(t) = PF_i(t-1) + p_i$, d_{Opt} represents the starting control, calculated by $\mu = (30 - \phi_i^{(LEG)})/30$, w represents the time of the activation signal, calculated as $w = \psi_i^{(LEG}/50$. $\phi_i^{(LEG}$, and $\psi_i^{(LEG}$ is the parameter for controlling swing activation and the timing of the ith PF neuron in certain LEG; F for forelimb and H for hindlimb. The values of ϕ and ψ were optimized in our previous research using a multi-objective evolutionary algorithm.

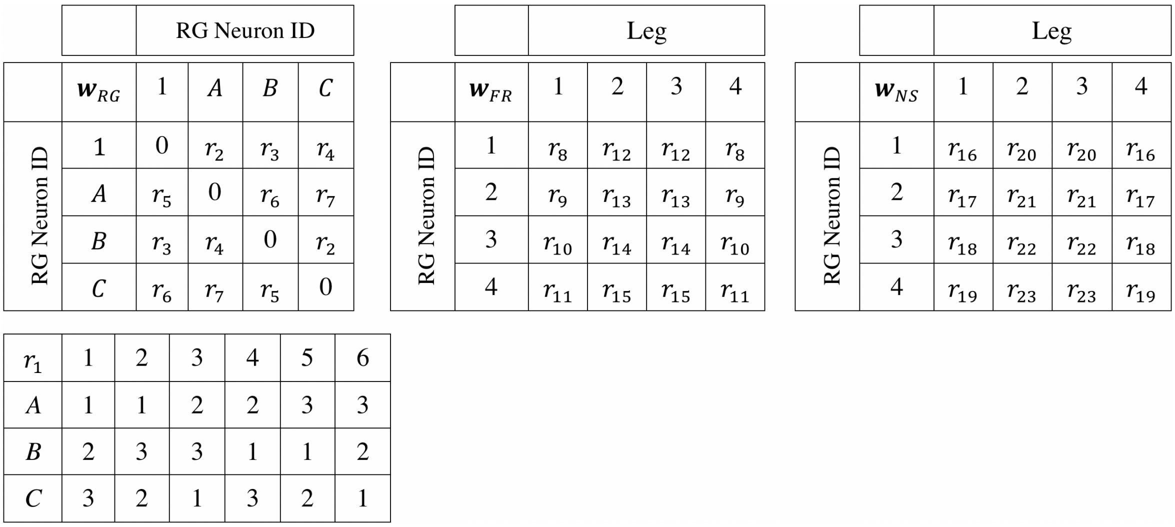
3.2.2 Optimization Parameters

To find the best CPG structure, I optimized the weights representing the interconnection of RG neurons ($\mathbf{w}_{RG}$) and those representing interconnection with sensory feedback ($\mathbf{w}_{FR}$, $\mathbf{w}_{SR}$, $\mathbf{w}_{NS}$). There were two optimization steps: 1) building a dynamic gait pattern; 2) building a malfunction compensator.

3.2.2.1 Dynamic Gait Pattern Optimization

This optimization aims to identify the best interconnection of RG neurons, and integration with sensory feedback, for generating dynamic gait patterns without changing the interconnection structure. I optimized a matrix parameter, representing the interconnection of RG neurons ($\mathbf{w}_{FR}$), the connection between force sensory feedback and RG neurons $\mathbf{w}_{SR}$, and the swing-phase-to-RG-neuron feedback ($\mathbf{w}_{NS}$). Those parameters are represented by 23

Table 3.2: *Representation of optimized parameter*



		RG Neuron ID			
	w_{RG}	1	A	B	C
RG Neuron ID	1	0	r_2	r_3	r_4
	A	r_5	0	r_6	r_7
	B	r_3	r_4	0	r_2
	C	r_6	r_7	r_5	0

		Leg			
	w_{FR}	1	2	3	4
RG Neuron ID	1	r_8	r_{12}	r_{12}	r_8
	2	r_9	r_{13}	r_{13}	r_9
	3	r_{10}	r_{14}	r_{14}	r_{10}
	4	r_{11}	r_{15}	r_{15}	r_{11}

		Leg			
	w_{NS}	1	2	3	4
RG Neuron ID	1	r_{16}	r_{20}	r_{20}	r_{16}
	2	r_{17}	r_{21}	r_{21}	r_{17}
	3	r_{18}	r_{22}	r_{22}	r_{18}
	4	r_{19}	r_{23}	r_{23}	r_{19}

r_1	1	2	3	4	5	6
A	1	1	2	2	3	3
B	2	3	3	1	1	2
C	3	2	1	3	2	1

parameters $(r_1, r_2, r_3, \cdots, r_2 3)$ optimized using SOEA; Table 1 shows a detailed representation of the parameter 3.2. The value of A, B, and C identify the RG neurons calculated from parameter r_1.

I minimized errors of speed oscillation, desired speed, and speed stimulation. For each evaluation process, I performed the robot simulation six times at different speed stimulations compared to parameters τ_f in Eq. (5.13); each performance required 1000 time cycle processes. I recorded the distance achieved (d_i) by each performance. The evaluation function is as follows:

$$f = \sum_{i=1}^{5} \left(\frac{1}{1 + \exp(d_i - d_{i+1})} \right) + \frac{1}{T} \sum_{t=1}^{T} \sqrt{(\alpha(t))^2} \tag{3.21}$$

where, T is the number of time steps, defined as 1000, and $\alpha(t)$ is the torso acceleration for time step t.

3.2.2.2 Malfunction Compensation Optimization

After optimizing the dynamic gait pattern, I optimized the connection between nociceptor and RG neurons ($\mathbf{w}_{NS}$). For this optimization, I found the best motion pattern was the leg malfunction condition. I used the optimized $\mathbf{w}_{RG}$, $\mathbf{w}_{SR}$, and $\mathbf{w}_{FR}$. I assume that the nociceptor may stimulate each RG neurons with a different effect. There were 16 parameters: $(\mathbf{w}_{NS,1-1}, \mathbf{w}_{NS,1-2}, \cdots, \mathbf{w}_{NS,4-4},)$.

To evaluate each set of parameters, I performed the simulation four times with different leg injuries. Under normal conditions, the value of N_i was zero ($N = \{0,0,0,0\}$). When the ith leg was injured, the value of N_i was one ($N_i = 1$). There were four steps, where $N_{STEP} = 1$.

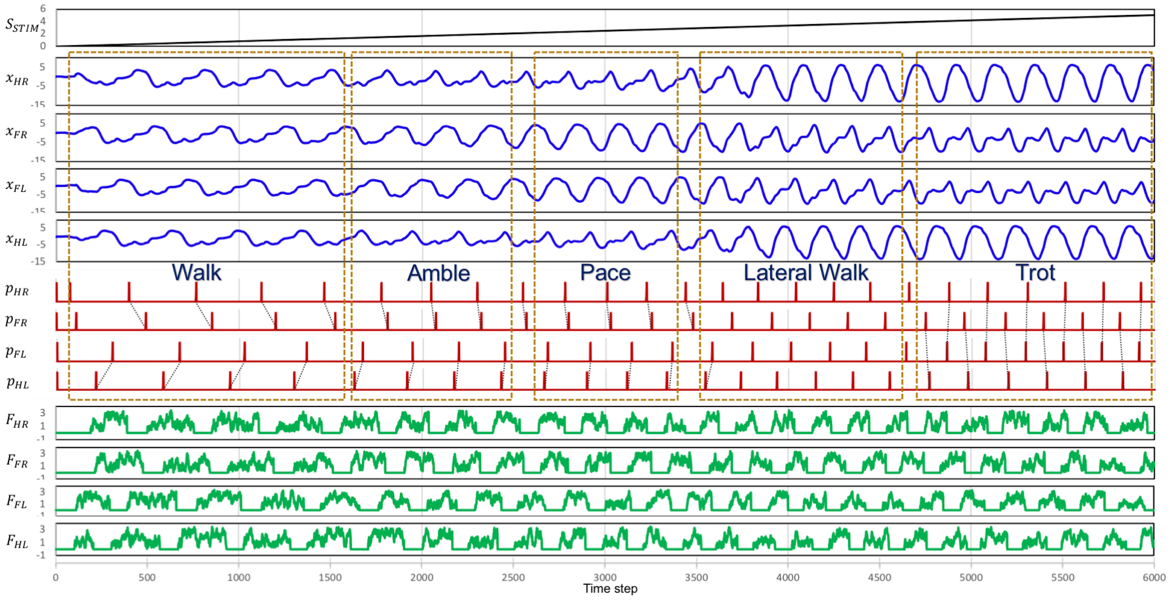


Figure 3.15: *The CPG model can generate a dynamic gait pattern through differing speed stimulation S_{STIM}. This increases the frequency of the CPG outputs $\mathbf{x}$ of five different known gait patterns from slow speed to high speed. Parameter F_{HR}, F_{FR}, F_{HL}, F_{FL} shows the ground reaction force for every limb.*

For every step, I evaluated the performance of the robot as follows:

$$f_{STEP} = \frac{1}{1+\exp(-d^{SAG}_{STEP})} + \sqrt{(d^{FRO}_{STEP})^2} \tag{3.22}$$

where f_{STEP} is the fitness function of a particular step-malfunction performance and d^{SAG}_{STEP} and d^{FRO}_{STEP} are the distance of the robot's movement in the sagittal and frontal directions. This optimization forces the robot to move straight ahead with minimal frontal direction movement.

3.2.3 Experimental Result

This experiment shows the results of the optimization processes of dynamic gait pattern and malfunction compensator in a computer simulation using an Open Dynamics Engine (ODE). Following this, I demonstrated the dynamic movement of the robot in the contexts of simulation and real robot implementation. For the robot simulation, I used a cat-like robot with an implemented muscle-based actuator model. The robots size was adjusted according to a cats size. I have defined the parameters not optimized in Table 3.3

Table 3.3: *The value of defined pattern formation parameters from preliminary test*

	val.		val.		val.		val.
$\phi_1^{(F)}$	0.034	$\phi_2^{(F)}$	12	$\phi_3^{(F)}$	12	$\phi_4^{(F)}$	0.034
$\phi_1^{(H)}$	13	$\phi_2^{(H)}$	0.012	$\phi_3^{(H)}$	0.012	$\phi_4^{(H)}$	13
$\psi_1^{(F)}$	20	$\psi_2^{(F)}$	10	$\psi_3^{(F)}$	10	$\psi_4^{(F)}$	20
$\psi_1^{(H)}$	0.034	$\psi_2^{(H)}$	0.013	$\psi_3^{(H)}$	0.012	$\psi_4^{(H)}$	0.034
μ_{HL}	1.676	μ_{FL}	0.501	T	1 .0	λ_{FL}	2.04
h	1.5	τ_f	1.2	R	12.5	α_{i0}	3.45

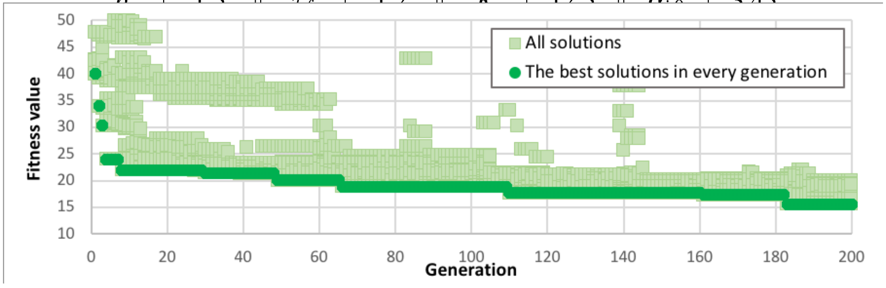


Figure 3.16: *The graph shows the evolution of the fitness value of gait pattern*

3.2.3.1 Optimization of Dynamic Gait Pattern

This optimization process used SSGA, which features 128 individuals in every population, and I optimized until the 200 generation. This optimization performance was validated to optimize neural interconnection [256]. Fig. 3.16 shows the results, indicating the fitness value decreasing gradually. The signal response to the gait pattern was evaluated using different speed stimulations. I performed the robot simulation in 6000 time steps (30 seconds). Fig.3.17, is a snapshot of the robots performance; more robot performance documentation is supplied in the supplementary video. I set S_{STIM} from zero and gradually increased it with the value of the time step (S_{STIM} = time step/1200); the result are presented in Fig. 3.15. The CPG model can generate dynamic gait patterns from five known gait patterns. These range from slow to high speeds: walk; amble; pace; symmetrical walk; trot. However, the symmetrical-walk pattern between pace and trot gaits does not exist in animal locomotion. However, in the proposed model, the symmetrical walk gait makes the transition from pace to trot.

Furthermore, to demonstrate the sensory aspect of gait-pattern generation, I tested the robot in rough terrain and evaluated the RG signal. I set the S_{STIM} at 0.5 so that the robot would generate the walking gait normally; Fig. 3.18 shows the results. During movement on rough terrain, swing time changed depending on contact time. Force feedback, or the end of swing time, influenced the RG as well as the gait pattern.

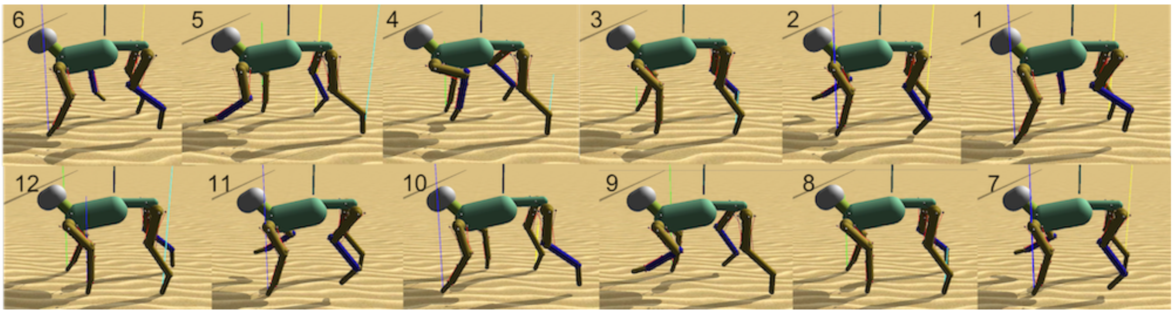


Figure 3.17: *The sample result in Open Dynamic Engines*

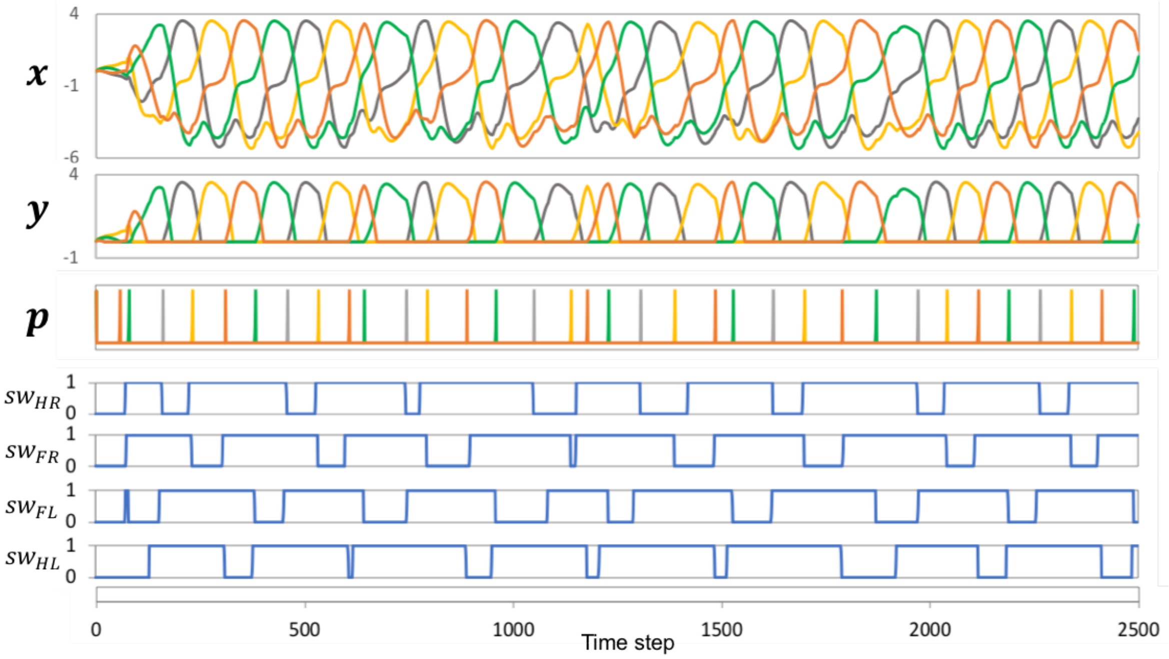


Figure 3.18: *The effect of sensory feedback in the RG during movement on rough terrain. sw_{HL} is the swing phase condition of the left hindlimb.* **x** *is the output signals of CPG calculated by Eq. 5.13. By analyzing the signal pattern* **p**, *I can see the pattern is changing to respond to the different force feedback or swing phase.*

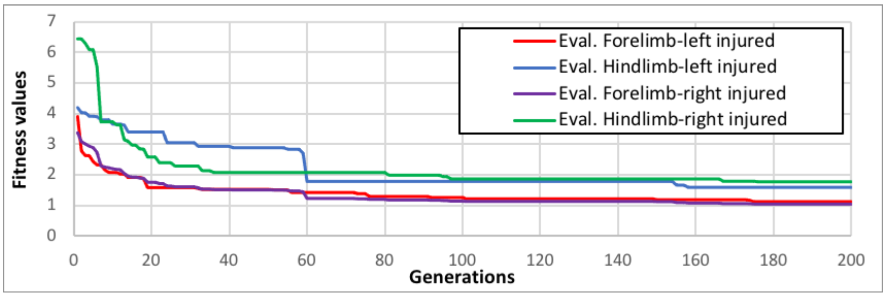


Figure 3.19: *The graph shows the evolution of the fitness value for malfunction compensation*

3.2.3.2 Optimization of Malfunction Compensation

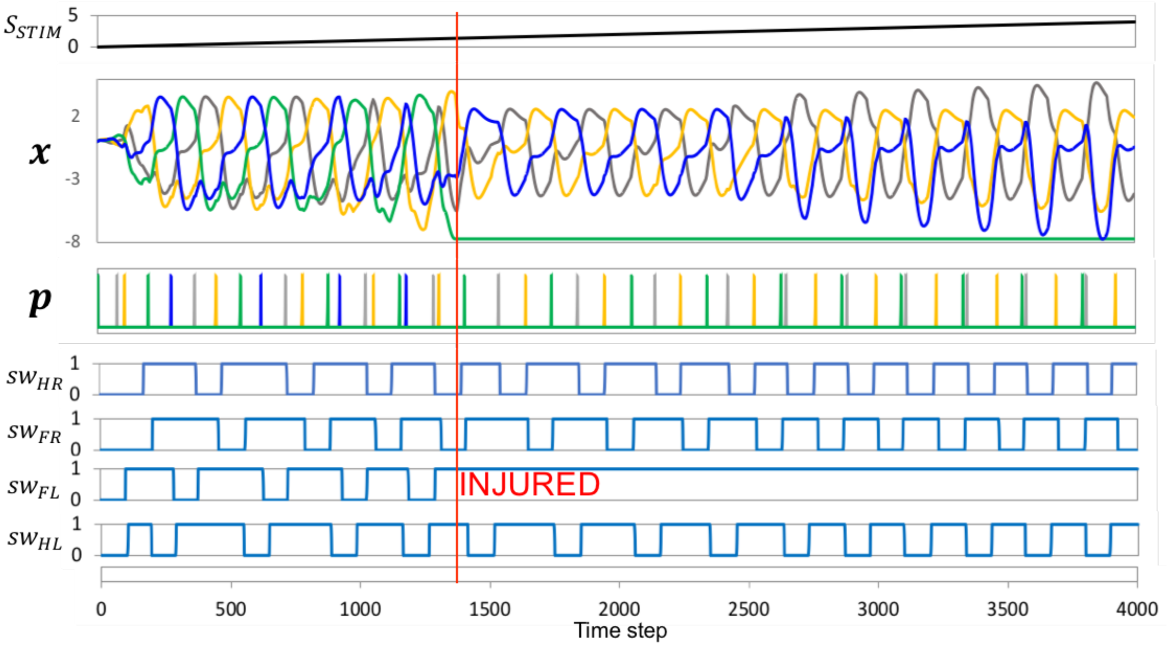
This optimization used the best parameters from the previous optimization. There were 64 individuals in each population. I optimized up to 200 generations; that is, 12800 evaluations. A similar evaluation was performed for different leg injury conditions. The fitness functions results are shown in Fig. 3.19 shows the results for this evaluation, indicating that four fitness values were decreasing. The best fitness value was achieved for the hindlimb condition.

To evaluate the optimized parameters effectiveness, I performed the simulation in normal conditions and then disabled one of the forelimbs or hindlimbs through injury. I also gradually changed the speed stimulation (S_{STIM}) to evaluate the gait-pattern response during leg injury. Fig. 3.20 shows the results, indicating that the proposed CPG model can respond to the leg injury signal through a smooth transition to both forelimb and hindlimb leg injury. There are different gait patterns under leg injury conditions. The model changed the gait pattern by controlling the signal input using sensory feedback. During leg injury, the model can still generate dynamic gait patterns by stimulating a different speed.

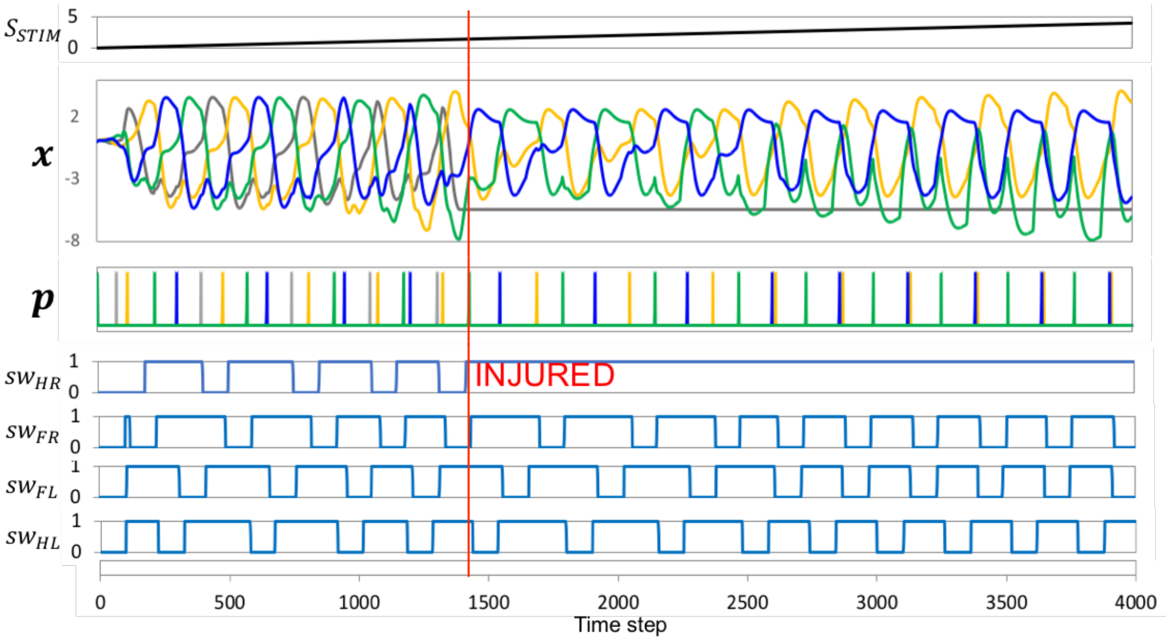
3.2.3.3 Real Robot Implementation

To demonstrate the effectiveness of the proposed model, I implemented the proposed model in the cat-like robot AQuRo-v3. The robots gait pattern has been controlled by the proposed CPG model. I used similar parameters for the simulation performance. First, I operated the robot with different speed stimulations on different terrains. The robot was able to move on both terrains with a dynamic gait pattern. There is a supplementary video of the robots performance, while Fig. 3.21 shows a sample snapshot. The robot was able to generate four gaits: walk, amble, pace, and trot.

Furthermore, I demonstrated the effectiveness of malfunction compensation by conducting tests in two conditions: injured forelimb and injured hindlimb. In these tests, the robot first moves normally. Then, after a few seconds, one of the legs was set to injured.



(a)



(b)

Figure 3.20: *The generated gait patterns in malfunction conditions and the speed stimulation responses. The signal pattern* ($\mathbf{p}$) *is changing to respond to the absence of CPG signals. The time phase decreases after a leg is injured. a) malfunction of the right forelimb at time step 1380. During injury, the model tends to generate a pattern with the same phase difference at a lower speed. At high speeds, left and right hindlimbs feature the same phase. b) malfunction of the left hindlimb at time step 1400. In this condition, the left and right forelimbs feature the same phase at a higher speed.*

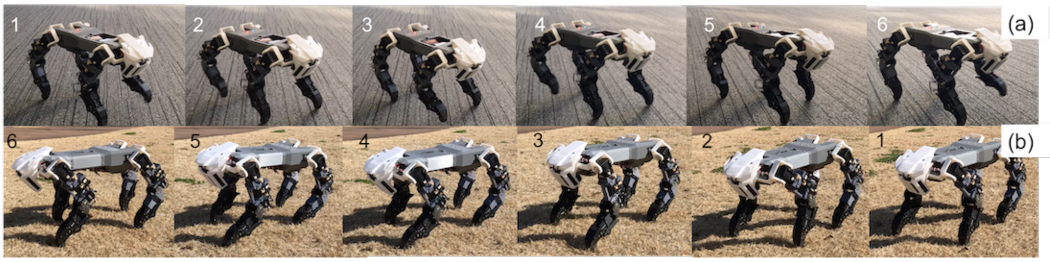


Figure 3.21: *Snapshots of the robot performance show the dynamic gait pattern: a) the robots performance on flat terrain; b) the robots performance on natural terrain.*

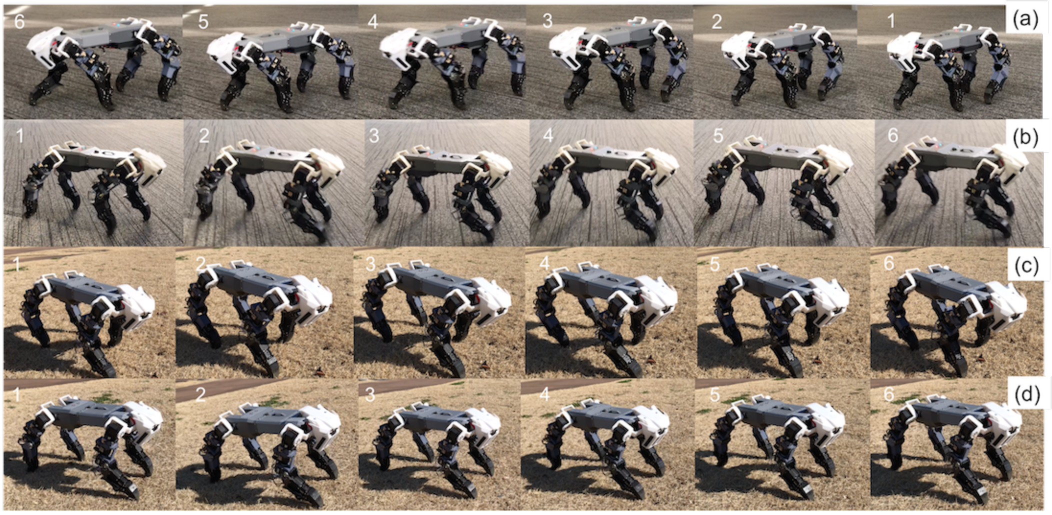


Figure 3.22: *Snapshots of the robots performance shows the dynamic gait pattern during leg injury: a) the robots performance on the flat terrain with a forelimb injury; b) the robots performance on flat terrain with a hindlimb injury; c) the robots performance on natural terrain with a forelimb injury; d) the robots performance on natural terrain with a hindlimb injury.*

The robot could generate a gait transition without falling in both conditions. I operated the robot on both flat and natural terrain. Fig. 3.22 is a snapshot of the robots performance; more detail is shown in the supplementary video provided or otherwise located at `https://youtu.be/C_XEsahqjpA`.

3.2.4 Conclusion and Discussion of Proposed Neural Primitive Model

This section has presented an efficient dynamic-gait-pattern-generationbased CPG mechanism with sensorimotor coordination. The model developed features substantial integration of CPG and sensory feedback, indicating that sensory feedback has a role in gait-pattern generation. The CPG structure uses a two-layer CPG model comprising a single RG and PF pools. Each leg is represented by a single CPG. Four RGs represent the four limbs. There are three sensory neurons in every leg: force sensor, swing phase, and pain receptor neurons. Section 3.2.1 explained the assumption of the integration of sensory neurons and RG neurons. However, it required optimization to achieve the best integration.

I used SOEA to optimize the network, a process separated into two steps: 1) gait pattern optimization; 2) malfunction compensation. There were two steps to optimizing integration and structure under normal conditions before manipulating the malfunction problem, enabling the model to generate a dynamic gait pattern through stimulation from only a single parameter (S_{STIM}). There are five patterns generated by robot performance: walk, amble, pace, symmetrical walk, and trot gait. Force feedback also has a big influence on generating gait pattern; this was demonstrated by the robots performance on rough terrain (see Fig. 3.18). The RG signal changed dynamically with changes to sensory input, and the proposed model generated gait transition between normal and injured conditions. It was also shown to generate dynamic gait during leg injuries.

In the real implementation, a cat-like quadruped robot demonstrated the effectiveness of the proposed model on steep terrain. Actuator control levels differed between the real robot and the simulation. However, these were controlled by the same gait pattern and did not impact the effectiveness of the proposed CPG model.

In future work, the proposed model should be applied to more challenging behavior, such as transitioning from ground movement to water movement. For this next step, understanding the role of sensory feedback for cat or dog water movements would be critical.

3.3 Cognitive Process to Movement-related Tract

In order to realize the integration between cognitive information and action in real-time, it requires the ingrated system from 1) a visual-attention control mechanism for controlling attention to external inputs, 2) object recognition representing the primary motor cortex, 3) a motor control model that determines motor commands, 4) a central pattern generation model representing pattern generator in the spinal cord, and 5) action generation in actuator level (joint angle level). This flows is active short term adaptation in microscopic level.

In this model I represent the model by integrating with biological process in human or animals point of view. I propose a process from cognition to the movement-related tract, based on a top-down model in mammals. Cognition involves the perception of external sensory information. It also includes a way to control the attention to external information for movement behavior. External sensory information plays a big role in controlling locomotion behavior. Visual information is received and processed by the retina and optic nerve and transferred to the thalamus and superior colliculus. The superior colliculus contributes to coordinating eye movement, responding to strong gaze direction commands from the cortex.

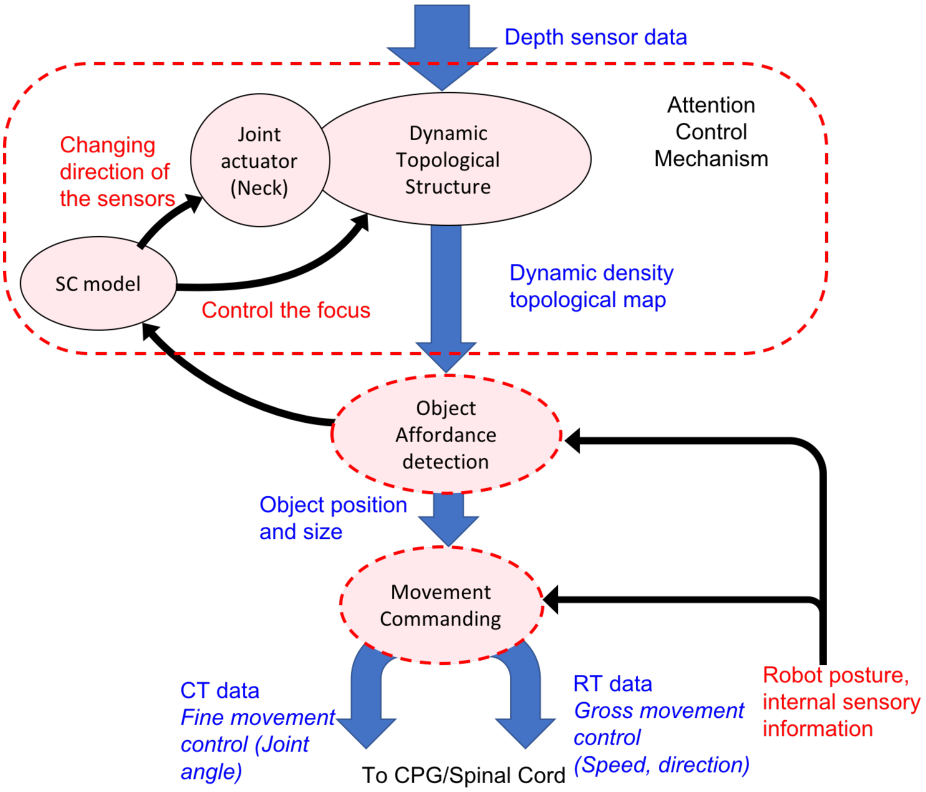


Figure 3.23: *Design of the attention control mechanism and process of external sensory information pathway until becoming movement-related information*

The thalamus passes visual information to the cortex. Most studies show that there are two processing streams in the cortex, the ventral system (the ‘what pathway) for recognizing and identifying the object, and the dorsal system (the ‘where pathway) for identifying the objects location and motion. The dorsal system is also involved in visually guided locomotion [90]. Movement-related information from the motor cortex is carried to the spinal cord by the corticospinal tract.

In order to represent this abstraction mechanism, I draw on neurophysiology. First, I build a dynamic attention model explained in Section 3.3.1 to represent the gaze. To recognize objects in front of the robot, I build an object affordance detector as explained in Section 3.3.2. This detector provides the object size, location, movement, providing for possible action at lower levels. Furthermore, I develop an interaction between the superior colliculus and the motor cortex. Our model connects the dynamic attention model with the affordance detection model for controlling the gaze or attention area.

3.3.1 Dynamic Attention to Movement-related Affordance detection

During movement, gaze attention is dynamic and strongly integrated with the surrounding environment. To model that, I imitate the superior colliculus mechanism. The superior colliculus receives data from ganglion cells in the retina. I use time-of-flight sensors to provide external sensory information, as explained in Section A.2. The aim of our model is to combine the attention and action which is effectively implemented for sudden obstacle response. Visual information requires a large amount of processing and knowledge building to estimate the shape and size of the object. On the other hand, the 3D point cloud information can provide the vector position directly which makes easier to recognize the shape of some object. Therefore point cloud information can be used directly without much processing, e.g. without object recognition. Topological map structure comprising neurons connected with their neighbors can also reduce the memory usage. The topological map represents the attention, with neuron density in this map representing the attention level. Figure 3.23 shows how the SC model controls the coordination between neck joint and the focus or density of topological structure entering the Object Affordance Detection system. I consider the neck joint in roll direction (θ_{neck}) by calculating $\dot{\theta}_{\text{neck}} = \tanh((\theta_{\text{off}} - \theta_{\text{neck}}) + (M_{off} - M_y)^3) \cdot \exp(\ln(0.5))\theta_{\text{neck}}^2$, where θ_{off}, M_{off}, M_y are the default value of neck joint angle, the center of frame in y-direction, and the attention position of the frame in y-direction, respectively. The SC model will select the area where an obstacle blocks the movement. It controls the attention by adjusting the number of neurons in associated areas and coordinating the neck joint to avoid the attention in the edge area of the gaze. The illustration of the attention model is shown in Fig. 3.24. In flat terrain, the normal vector ($\mathbf{N}$) is homogeneous. The model has low attention where the nodes are sparse. When I put a sudden red box, then few nodes are shifted. There are non-homogeneous normal vectors (red color). In this state, the object still cannot be detected precisely. Then, the attention increases the number of nodes around the non-homogeneous node. After the nodes are increased, then the object is clarified. The information is enough for further processes.

Our topological structure is built on a dynamic density growing neural gas model (DD-GNG) (see Section 3.1). Local node density is set by controlling the probability of finding raw data in that neighborhood. I hence calculate the strength of each node as influenced by the attended area. Node strength is calculated using Eq. (3.23) for the discount rate, while Eq. (3.24) controls the probability of finding information by local random search. The parameter $\mathbf{h}$ is all nodes's position and h_k represents the kth 3D node's position; r, and μ are the radius of the sphere, and the 3D position of the sphere received from the Affordance detection.

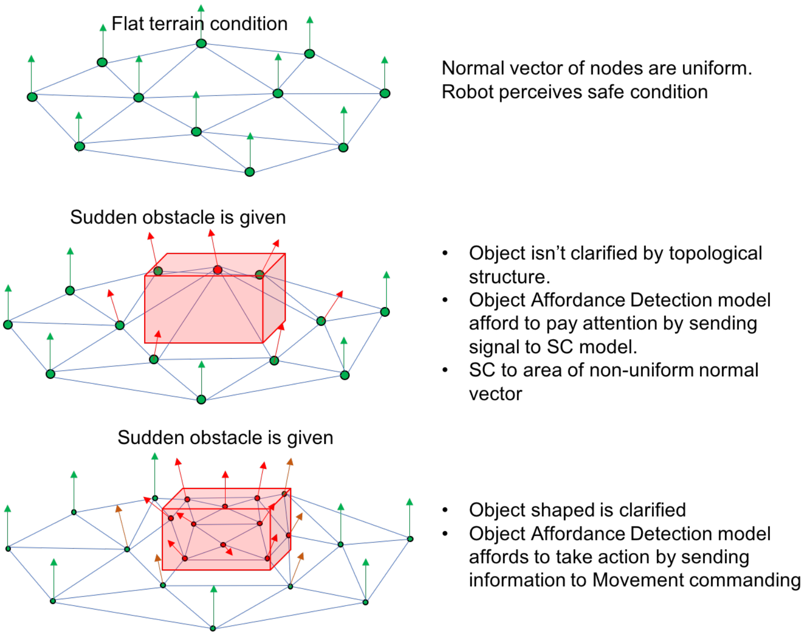

Flat terrain condition
Normal vector of nodes are uniform.
Robot perceives safe condition
Sudden obstacle is given
• Object isn't clarified by topological structure.
• Object Affordance Detection model afford to pay attention by sending signal to SC model.
• SC to area of non-uniform normal vector
Sudden obstacle is given
• Object shaped is clarified
• Object Affordance Detection model affords to take action by sending information to Movement commanding

Figure 3.24: *Attention system using a topological map model*

$$\delta_k = \frac{e^{-(r+\|h_k-\mu\|)}}{1+e^{-(r+\|h_k-\mu\|)}} \tag{3.23}$$

$$S(P) = 0.2 + \frac{e^{-(r+\|P-\mu\|)}}{1+e^{-(r+\|P-\mu\|)}}) \tag{3.24}$$

The attended area will be selected and calculated by the affordance detection model that represents the cortexs main role. The main algorithm is shown in Alg. (7).

Algorithm 7 Dynamic Topological Structure

```
Init: generate two nodes at random position
    $0\ c_{i,j} = 0\ (\forall_i, \forall_j)$, $A = 1, 2$, $r = 2, t = 0$, $r = -1, \mu = -1$, $N^{(non-H)} = 0$
while $\mathbb{P} \neq$ NULL do
    $\mathbb{P} \leftarrow$ raw data from depth sensors
    if $N_i$ then
        $\mathbb{Q} \leftarrow$ raw data inside $Sphere(r, \mu)$
    $\lambda \leftarrow 0$
    while $\lambda < \max_\lambda$ do
        $V_\lambda \leftarrow$ a random of $\mathbb{Q}$
        if $(S(V_\lambda) = 1$ calculated in Eq. (3.24) then
            $\lambda \leftarrow \lambda + 1$
    h = DD-GNG main process $(V)$ explained in [250]
    for $i \leftarrow 1$ to node's number do
        $E_i \leftarrow E_i \times \upsilon_E$
        $U_i \leftarrow U_I \times \upsilon_E / \delta_i^2$
    $t \leftarrow t + 1$
    $(r, \mu, N^{(non-H)}) \leftarrow$ Object Affordance Detection in Section 3.3.2
```

3.3.2 Object Affordance Detection

The concept of affordance was proposed by Gibson in ecological psychology [85]. Affordance is what the environment offers to individuals. Affordance does not depend on the ability of the individual to recognize or use it [87]. Affordance is also defined by Turvey as the dispositional properties of the environment [294], complemented by the effectivity or dispositional properties of the actor. Affordance is hence not inherent to the environment: it depends also on particular actions and the actor's capabilities. Differences between in individuals' bodies may lead to a different perceptions of affordance. Perception also depends on the intention: while affordances exist in the environment regardless of whether they are perceived, perception of an affordance depends on the affordance being there to create it.

Animal locomotion is controlled by perceiving affordances [86]. In other words, prospective action is generated depending on the affordance information that the locomotion generator receives [295]. In free space, animal stepping movements are governed according to the body's inertial condition. The adaptation process compares the estimated next stepping point, accounting for current inertial conditions, with the affordances of the surface.

In this model, affordance detection will provide possible actions. There are two feedback flows involving object affordance detection in the cortex. The first one commands the superior colliculus to control the appropriate attention. The second one commands the spinal cord and corticospinal tract. I calculate the affordance from the vectors of available edges in every node (N) calculated in Eq. (3.25)

$$\mathbf{N} = \frac{1}{N_i^{(edge)}} \sum_{k=0}^{k=N_i} \left(\frac{(\mathbf{h}^i - \mathbf{n}_k^i) \times (\mathbf{h}^i - \mathbf{n}_{k+1}^i)}{||(\mathbf{h}^i - \mathbf{n}_k^i) \times (\mathbf{h}^i - \mathbf{n}_{k+1}^i||} \right) \tag{3.25}$$

where N_i and $\mathbf{h}^i$ is the number of edges and vector node position in the ith node. $\mathbf{n}_k^i$ is the vector position of the node connected with the kth edge in the ith neuron.

3.3.2.1 Representing Cortex and SC Integration

The accuracy of the affordance depends on the degree of awareness and attention. Depending on the accuracy of the perception, affordances can be wrongly perceived [105]. Awareness and attention are certainly closely related but perhaps not inseparable. When I are walking, our attention may not be focused on the path, however, the attention typically shifts toward the path when danger is noticed [84]. It seems that some unconscious knowledge tells the conscious self to "look out". In fact, the perceptual–cognitive system is continuously processing, though our awareness of it is minimal. This system is extremely efficient, shifting attention automatically when needed. In biological process point of view, it shows the integration of visual motor cortex and superior colliculus level.

In our approach, the degree of attention is generated automatically by Dynamic Attention with Density Representation (see Section 3.1.1). Outside the focus of attention, the accuracy of perceiving the affordance is lower, owing to the low information density. When a risky obstacle is detected, the attention focuses onto it, and perceptual accuracy increases because the information density increases (Fig. 3.24). The affordance-perception process receives external sensory information from the attention process in the form of a topological structure.

When the density of node (α) in the non-homogeneous vector ($N^{(\text{non-H})}$) calculated in Eq. (3.1) is lower than the threshold calculated in Eq. (3.26), the affordance model prompts the SC to increase the node density in a certain area explained in Section 3.3.1. The model sends the center of attention (C_a), the radius size (R_a) of non-homogeneous vectors calculated as follows: $C_a = 1/N^{(\text{non-H})} \sum_{i=1}^{i=N^{(\text{non-H})}} h_i(\text{for } \theta(N_i) > \Theta), R_a = \sum_{i=1}^{i=N^{(\text{non-H})}} \arg max(h_i - C_a)$; where $\theta(N_i)$ is the angle between vector N_i and the vertical vector. $N^{(\text{non-H})}$ represents the number of non-homogeneous normal vectors of nodes The attention area in the proposed model is limited to the pathway area of the robot movement.

$$\alpha = R_a/N^{(\text{non-H})} \tag{3.26}$$

3.3.2.2 Representing Cortex and Movement-related Processing Integration

The pathway process from cortex to movement-related motor cortex shows the how the affordances are represented to the motor cortex. The visual cortex model sends information about the detected object or obstacle (size and position), where its size and position are represented by parameter C_a, R_a in Section 3.3.2.1. In our model, the non-homogeneous vector in the attention area will be detected as the object or obstacle during movement. After that, the affordance perception will serve the possible action and safe area by detecting normal vector to the vertical direction in the attention area.

3.3.3 Affordance Effectivity - fit

For actual usefulness, affordance needs to be integrated with effectivity. Effectivity provides a way to consider whether the individual robot or animal can actually make use of the affordance. Affordance and effectivity are hence important, and form the ecological boundary of the perceiving–acting cycle that allows actions to succeed in attaining their goals [268, 269]. In our locomotion system, the effectivity of behavior complements the affordance information. In short-term adaptation control, two effectivities are used: the prediction

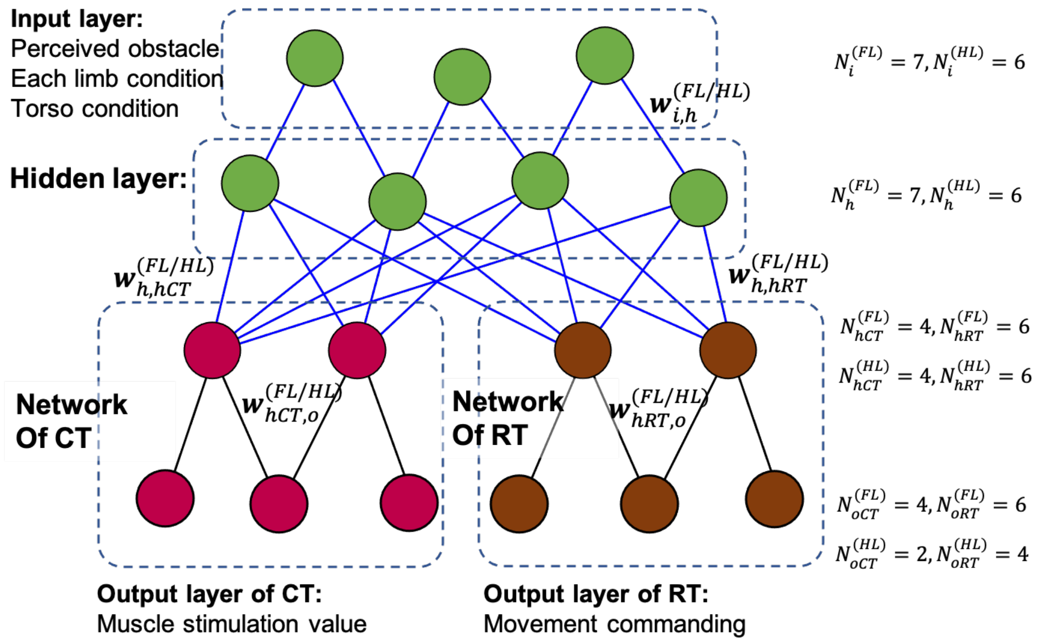


Figure 3.25: *Design of the movement command model integrates perception information into the locomotion generator model. The CT (corticospinal tract) network connects directly to the muscle reflex system. The RT (reticulospinal tract) network drives signal to the central pattern generator.*

of the next stepping point, and the inertial leg movement during the swing. The affordance perceiver will send the possible swing action to the behavior interrupt. This information will then be considered for generating the appropriate action, whether as short-term behavior or in medium-term adaptation (see Fig. 2.2). In this model I represent the model by integrating with biological process in human or animals.

Locomotion involves two command pathways: one for gross movement mainly via the reticulospinal tract, and another for fine movement mainly via the corticospinal tract [16]. By taking advantage of this biological process, I build an efficient mechanism between external sensory information (cognition) and action (muscle model). I separate the control pathway for gross movement (speed, direction) and fine movement (positioning, joint control). It will be effective when the system detects a sudden obstacle . When the system recognizes a safe or flat terrain, the model communicates movement speed and direction to the CPG model through the reticulospinal pathway. In other circumstances, when the system detects obstacles to avoid and changes the swing pattern, associated signals are sent through the corticospinal pathway. The model is represented by an artificial neural network that deals with both object detection and movement commands. The network structure is illustrated in Fig. 3.25. The main network is represented by green neurons and separated into the CT

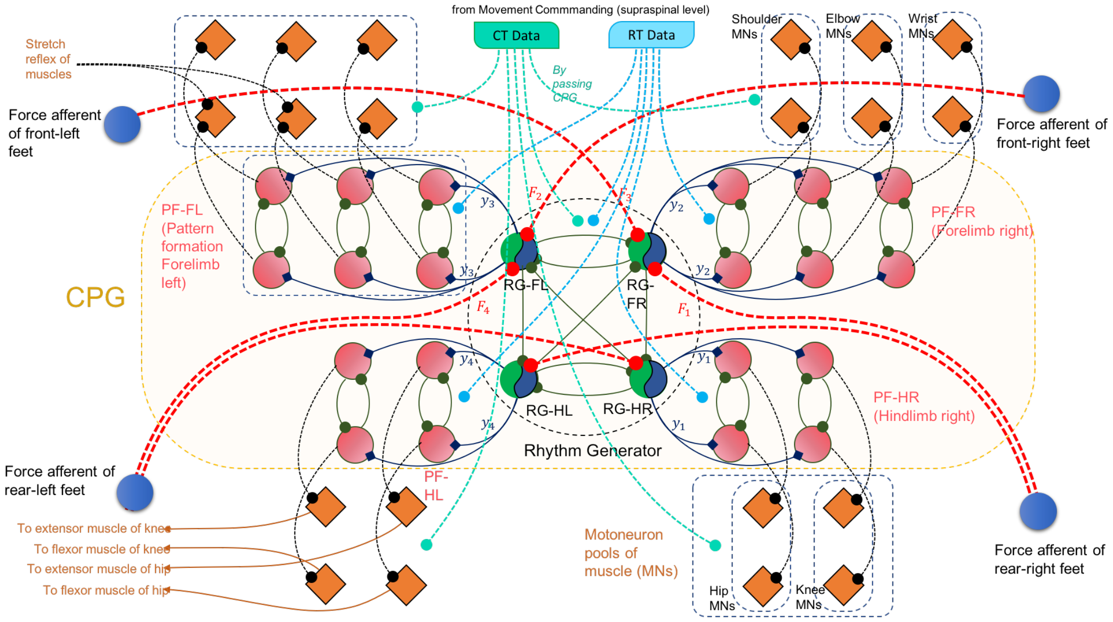

from Movement Commmanding (supraspinal level)
CT Data
RT Data
By passing CPG
Stretch reflex of muscles
Force afferent of front-left feet
Shoulder MNs
Elbow MNs
Wrist MNs
Force afferent of front-right feet
PF-FL (Pattern formation Forelimb left)
CPG
y_3
y_3
F_2
F_3
y_2
y_2
PF-FR (Forelimb right)
RG-FL
RG-FR
F_4
F_1
y_4
y_4
y_1
y_1
RG-HL
RG-HR
PF-HR (Hindlimb right)
Rhythm Generator
Force afferent of rear-left feet
PF-HL
To extensor muscle of knee
To flexor muscle of knee
To extensor muscle of hip
To flexor muscle of hip
Motoneuron pools of muscle (MNs)
Hip MNs
Knee MNs
Force afferent of rear-right feet

Figure 3.26: *Design of two-layered CPG with distributed rhythm generator and pattern formation*

network representing the Corticospinal Tract and the RT network representing the Reticulospinal Tract. The input layer takes the perceived obstacle information (location, size), the state parameters of each limb (phase, joint angles, limb vector movement), and the torsos state parameters (tilt and speed). $\mathbf{w}^{(FL/HL)}$ represents the synaptic weights and $N^{(FL/HL)}$ represents the number of neurons in each layer.

When an obstacle is perceived during a limb's swing phase, the CT network is activated to overwrite the current muscle stimulation, bypassing the CPG network. If the obstacle is perceived before the limb's swinging phase is activated, the RT network will be activated to send swing commands through the CPG level. The output of CT network contains 2 parameters $(l_{CT,1}, l_{CT,2})$ and 4 parameters $(l_{CT,3}, l_{CT,4}, l_{CT,5}, l_{CT,6})$ transferred to the motoneuron of hindlimb and forelimb, respectively, explained in Section 3.3.4.2. Then, the output of RT network represented by $c_{RT,LEG,i}$ parameters is transferred to PF neurons, explained in Section 3.3.4.1. There are 4 parameters influence to hindlimb leg and 6 parameters influence to forelimb leg.

3.3.4 Integrated CPG Model

In this part, the CPG model is integrated with Affordance Efectivity fit module. the integrated model of CPG can be seen in Fig. 3.26. The RG neurons model can be seen in Section 3.2.1.

3.3.4.1 Pattern Formation Model

PF neurons receive signal from Affordance-Effectivity fit module in microscopic level to generate an action responding sudden obstacle. The inner state of the PF neuron in *i*th joint (PF_i) is calculated as follow:

$$PF_{i,k}(t) = e^{\left(\log(0.5)\times\left(\frac{|pf_i-\varphi|}{(\mu\times w\times c_{RT,LEG,i})}^{3}\right)\right)} \tag{3.27}$$

$$pf_i(t) = pf_i(t-1) + p_i \tag{3.28}$$

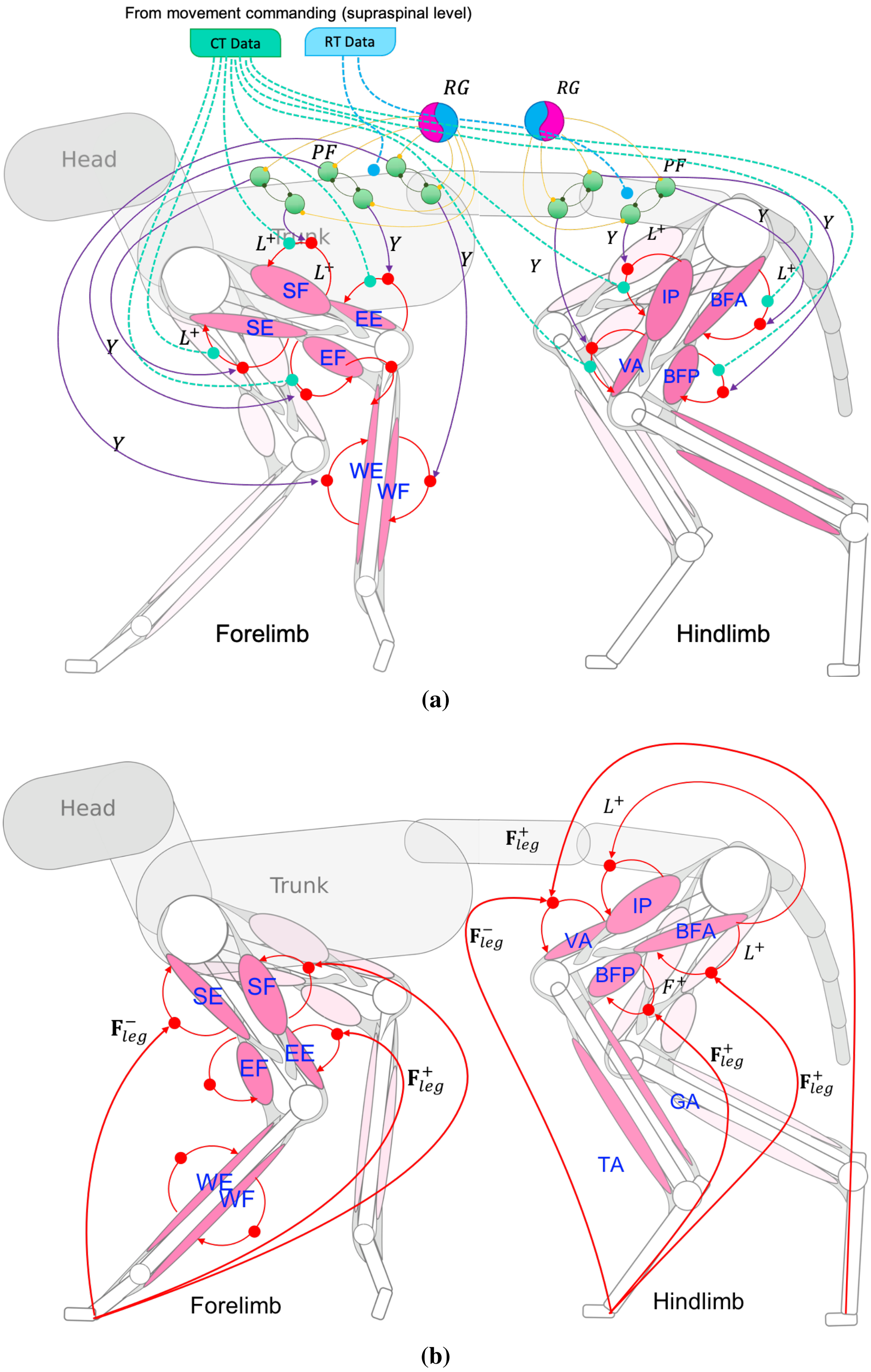
where $c_{RT,LEG,i}$ represent the effect of RT network to the *i*-th PF neuron of certain LEG. φ represents the starting control calculated by $\varphi = (30-\phi_i^{(LEG)})/30$, and w represents the time of activation signal calculated as $w = \psi_i^{(LEG)}/50$. $\phi_i^{(LEG)}$ and $\psi_i^{(LEG)}$ are the parameters for controlling swing activation and timing of the *i*th PF neuron in certain LEG. The value of ϕ and ψ will be optimized using a multi-objective evolutionary algorithm. p_i is the spike signal calculated in the Supplementary material - 1. The value PF_i in then transmitted to the muscle model to implement the actual swing.

3.3.4.2 Musculoskeletal Model

All mammals have a similar skeleton which includes both forelimb and hindlimb structures. In general, mamallian locomotion has both stance and swing phases. Several decades ago, some researchers proposed that rhythmic alternating contractions of muscles bring about cycling between the stance phase (extensor muscles) and the swing phase (flexor muscles). The rhythm can be generated without any sensory information from receptors in the skin, joints, and muscle [217, 33]. Research on the stanceswing transition has yielded a substantial body of evidence that this transition is controlled by sensory signals from leg proprioceptors and has identified some of the receptors providing these signals [217].

The swing-to-stance transition unifies the two phases of stepping, rendering them inseparable [188, 148]. Based on this evidence, the swing-to-stance phase transition is activated automatically in response to stimuli from ground forces and hip positions. Thus, the swing-to-stance transition is the best phase in which to control the locomotion pattern. I control the swing stimulation using signal pattern modulation from a neural oscillator proposed by Matsuoka. One CPG neuron represents one joint. The signal will stimulate the muscle to perform the swing phase. The muscle model is separated into two submodels for hindlimb and forelimb control depicted in Fig. B.1.

From the Figure B.1a, I can see the effect of CT directly to the muscle activation during



(a)

(b)

Figure 3.27: *The muscle reflex structure during its phase. a) Muscle reflex structure in swing phase. b) Muscle reflex structure in stance phase. F^+_{leg} and F^-_{leg} represent the positive and negative feedback of the force afferent*

the swing phase. It effects the value of the length offset in certain muscle stimulation. Equations (3.29-3.32) shows muscle stimulation for hindlimb during swing phase. The effect of CT network to hindlimb is in $l_{CT,1}$ and $l_{CT,2}$ parameters.

$$S_{BFA}(t) = S_{0,BFA} - G^{y}_{BFA} Y_{IP} + G^{l,S}_{BFA}(l_{CE,BFA} - l_{off,BFA} - l_{CT,1}) \tag{3.29}$$

$$S_{IP}(t) = S_{0,IP} + G^{y}_{IP} Y_{IP} + G^{l,S}_{IP}(l_{CE,IP} - l_{off,IP} - l_{CT,2}) \tag{3.30}$$

$$S_{VA}(t) = S_{0,VA} - G^{Y}_{VA} Y_{IP} + C_{VA} \tag{3.31}$$

$$S_{BFP}(t) = S_{0,BFP} + G^{Y}_{BFP} Y_{BFP} - C_{BFP} + G^{F}_{BFP} F_{leg} \tag{3.32}$$

Furthermore, Equations (3.33-3.38) shows muscle stimulation for forelimb during swing phase. The effect of CT network to forelimb is in $l_{CT,3}$ to $l_{CT,6}$ parameters.

$$S_{SF}(t) = S_{(0,SF} + G^{Y}_{SF} Y_{SF} + G^{l}_{SF}(l_{CE,SF} - l_{off,SF} - l_{CT,3}) \tag{3.33}$$

$$S_{SE}(t) = S_{0,SE} - G^{Y}_{SE} Y_{SE} + G^{l}_{SE}(l_{CE,SF} - l_{off,SF} - l_{CT,4}) \tag{3.34}$$

$$S_{EE}(t) = S_{0,SE} - G^{Y}_{EE} Y_{EE} + G^{l}_{EE} \cdot Y_{EE} \cdot (l_{EE} - (-l_{off,EF} - l_{CT,5}) \tag{3.35}$$

$$S_{EF}(t) = S_{0,EF} - G^{Y}_{EF}(1 - Y_{EF}) + G^{l}_{EF} \cdot Y_{EF} \cdot (l_{EF} - l_{off,EF} - l_{CT,6}) \tag{3.36}$$

$$S_{WE}(t) = S_{0,WE} + G^{Y}_{WE} Y_{WE} \tag{3.37}$$

$$S_{WF}(t) = S_{0,WF} + G^{Y}_{WF}(1 - Y_{WF}) \tag{3.38}$$

In the robot implementation I convert the muscle activation to joint angle by calculating the resultant of those muscle torque to drive the joint motor. The calculation of the torque can be seen in Appendix B.

3.3.5 Optimization Process for the Movement Command Model

Once the lower spinal level has been optimized, I can optimize the movement command model for controlling the appropriate signal for either the corticospinal pathway or reticulospinal pathway. I optimize the network for hindlimb and forelimb separately. Each network separates the optimized parameters into 3 groups: 1) A: weights parameter associated with green neurons only, B: weights parameters associated with CT neurons, and C: weights parameters associated with RT neurons. In the case when the fitness related to RT has good result and the CT has bad result then the SSGA optimization evaluates the parameter of group B only. In the case when the fitness related to CT has good result and RT has bad result then the SSGA optimization evaluates the parameter of group C only. The number of

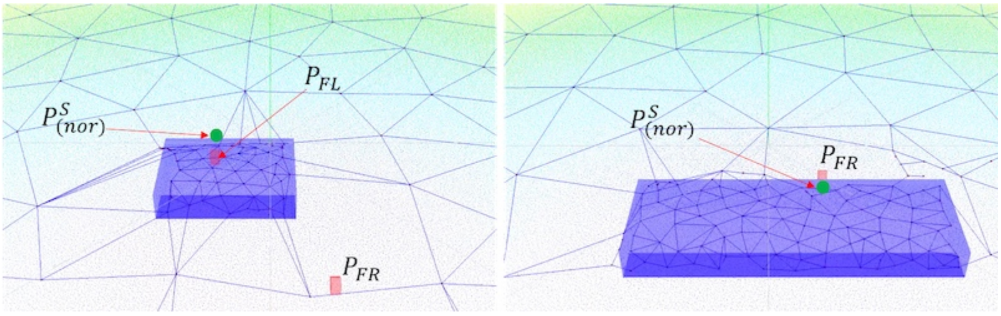


Figure 3.28: *Sample of the result of affordance detection. P_{FL} is the position of forelimb left foot and P_{RF} is the position of forelimb right foot.*

parameters optimized can be seen in Table 3.4.

Table 3.4: *Total Number of Optimized Parameters*

Group		A	B		C		Total
Weight paramters		$\mathbf{w}_{i,h}$	$\mathbf{w}_{h,hCT}$	$\mathbf{w}_{hCT,o}$	$\mathbf{w}_{h,hRT}$	$\mathbf{w}_{hRT,o}$	
Leg	Forelimb	49	28	16	42	36	171
	Hindlimb	36	24	8	36	24	128

Before that, I show the result of the attention control process and the object recognition as the input of the optimized model. A sample affordance detection with movement commanding result taken from the experiment in Section 3.3.6 can be seen in Fig. 3.28.

3.3.5.1 Optimization for the CT Network

The CT network stimulates the muscular stretch reflex. I optimize the network to control the swing acceleration when facing a sudden obstacle. I evaluate the congruence of step length commands to the CT network and the resulting robot step length. In one evaluation, the robot performed 1000 time steps on flat terrain. I supply the network of CT with 9 different command values, $I = \{-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8\}$ in the middle of the swing, when the feet are at or near their highest phase. I evaluate using the following measure:

$$f = \sum_{i=1}^{9} \left(\frac{1}{1 + \exp(s_i - s_{i+1})} \right), s_5 = S_{(nor)} \tag{3.39}$$

where $S_{(nor)}$ is the average step length in normal swing; s_i is the step length of the ith command. I assume the value of $s_5 = 0$ when $I_5 = 0$. I optimized using a single-objective evolutionary algorithm.

3.3.5.2 Optimization of the RT Network

The RT network will drive the CPG network, RG neurons and/or PF neurons. I optimize the network to avoid obstacles with a certain height and a certain step length. I evaluate the congruence of step length and step height commands to the RT network with robot performance. In one evaluation, the robot performed 1000 time steps on flat terrain. I supply the network of RT with 9 different command values, $I = \{-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8\}$ when the feet start swinging. The evaluation function is:

$$f = \sum_{i=1}^{9} \left(\frac{1}{1 + \exp(s_i - s_{i+1})} \right) + \sum_{i=1}^{9} (-hS_i), s_5 = S_{(nor)} \tag{3.40}$$

where hS_i is the height of step of ith command.

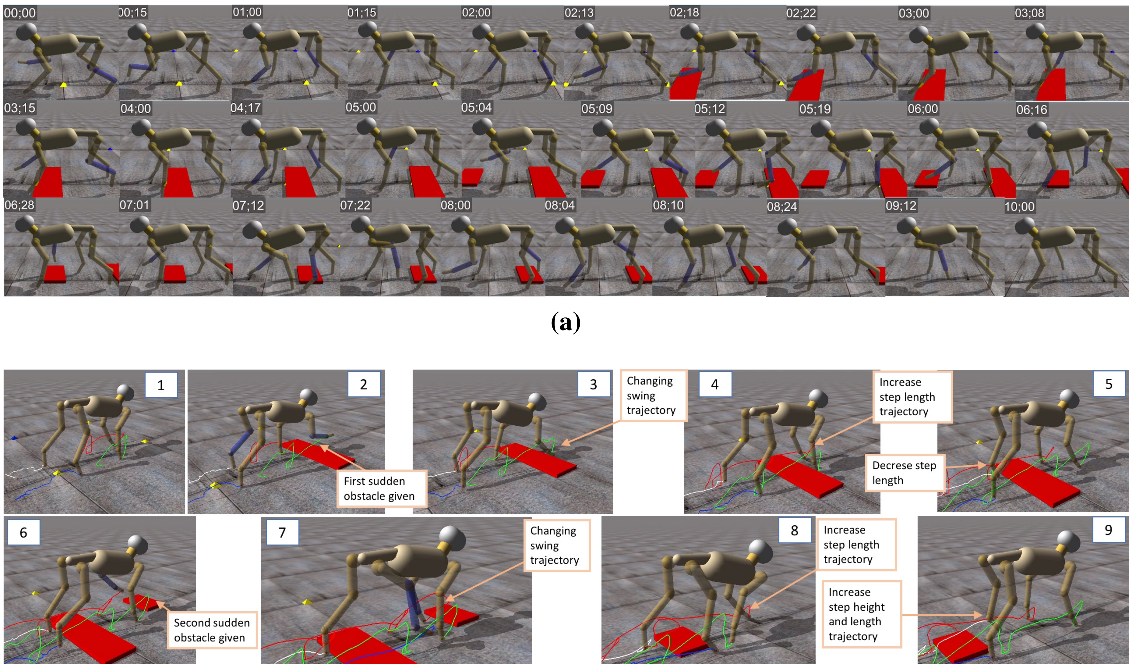
3.3.6 Robot Performance when Suddenly Encountering an Obstacle

In order to show the effectiveness of the proposed model, I run the quadruped robot both in simulation and in real performance.

3.3.6.1 On flat terrain

In testing scenario, I twice put sudden obstacles, of different sizes, in the robots way while it was walking. The obstacles were cuboids of size (length × width × height) 700 mm × 230 mm × 40 mm, and 300 mm × 200 mm × 40 mm. In simulation I put the first obstacle at time step equals to 555 and the second obstacle at time step equals to 1000. The robot performance snapshot can be seen in Fig. 3.29. Figure 3.29a illustrates the whole robot performance avoiding the obstacle from side view and Fig. 3.29b shows a snapshot of the robots behavior when avoiding the obstacle under several conditions. To analyze the effect of muscle force, I show the recorded robot performance in Fig. 3.32.

In Fig. 3.29b: 1) The robot moved in normal condition from the starting condition. 2) The sudden obstacle is given at time step 541, when the left forelimb is swinging. 3) The robot perceived the sudden obstacle at time step 555. It required 15 time samplings (0.070 second) for the cognitive model to perceive the model (see Fig. 3.30a, perceiving processing) and generate the overwriting muscle stimulation signal from the CT network. It stimulated the shoulder extensor (SF) muscle to advance the limb (see Fig. 3.31 in 4-th signal, for stimulation). 4) Since the existing obstacle did not affect the next foothold position, the right forelimb stepped over the obstacle without any overwriting command. 5) There was no possibility for the right hindlimb to step over the obstacle due to the limitation of the possible


00;00
00;15
01;00
01;15
02;00
02;13
02;18
02;22
03;00
03;08
03;15
04;00
04;17
05;00
05;04
05;09
05;12
05;19
06;00
06;16
06;28
07;01
07;12
07;22
08;00
08;04
08;10
08;24
09;12
10;00
1
2
First sudden obstacle given
3
Changing swing trajectory
4
Increase step length trajectory
Decrese step length
5
6
Second sudden obstacle given
7
Changing swing trajectory
8
Increase step length trajectory
Increase step height and length trajectory
9

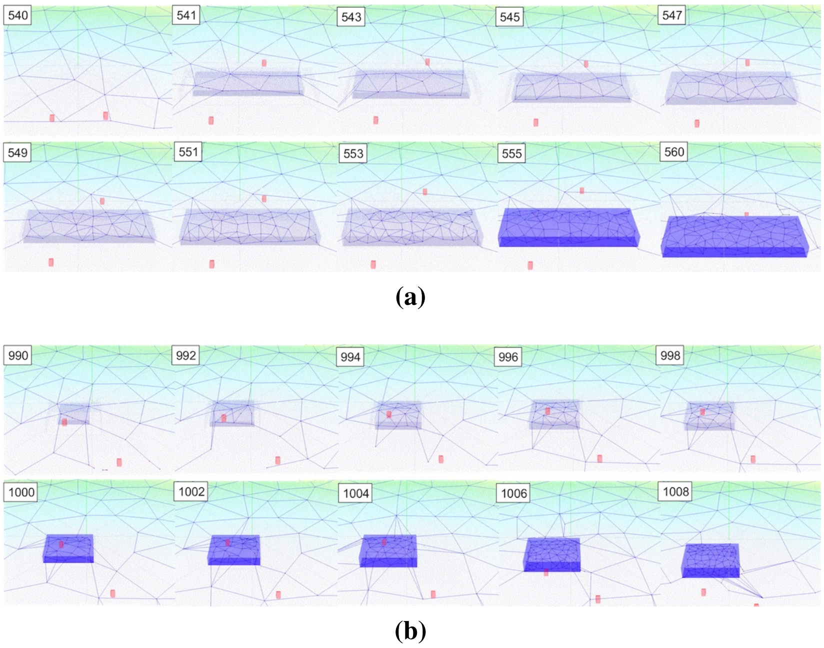
(a)

(b)

Figure 3.29: *The robot's performance in simulation. The detailed robot performance can be seen in the supplementary video. a) The time series snapshot from side view. b) The snapshot of the robot performance in certain condition from perspective view. 1] Robot performs in normal walking 2] First sudden obstacle is given during left forelimb swing 3] The left-forelimb changes the swinging pattern by reflex placing (see green line trajectory) 4] Right forelimb steps over the obstacle. 5] Right hindlimb decreased the step length because stepping over would have been too far 6] Second sudden obstacle is given during the right forelimb swing phase 7] Placing reflex to avoid the obstacle (see red line trajectory) 8] Right forelimb steps over the obstacle by increasing step length 9] Right hindlimb increases step height and step length to avoid the second obstacle*

step length; the robot decreased the step length. 6) At time step 990, a second sudden obstacle was given when the right forelimb was in the swing phase. 7) At time step 1000, the robot perceived the obstacle (see Fig. 3.30b, perceiving processing) in 12 time steps (0.05 second). It stimulated the shoulder flexor (SF) muscle to draw back the limb (see Fig. 3.31 in 2nd signal, for stimulation). 8) The right forelimb step length was increased by command from the RT network. 9) The right hindlimb received a command signal from the CT to change the swing pattern to avoid the obstacle by increasing step height and step length (see white line trajectory in Fig. 3.29b-9).

Furthermore, in order to evaluate the robustness of the proposed model, the robot performed 50 times, where in each trial, I put one sudden obstacle (300 mm × 200 mm × 40 mm) in a random position in front of the robot. The robot successfully avoided the obstacle 41 times, i.e. in 82% of the trials.



(a)

(b)

Figure 3.30: *The result of Affordance detection 1) first obstacle affordance detection 2) second obstacle affordance detection*

3.3.6.2 On sloped terrain

The robot performed on sloped terrain with 5 obstacles which are listed below:

1. 700 mm × 230 mm × 40 mm
2. 300 mm × 200 mm × 40 mm
3. 500 mm × 140 mm × 45 mm
4. 200 mm × 100 mm × 50 mm
5. 600 mm × 200 mm × 45 mm.

I placed the first and second sudden obstacles during uphill movement. After that, I put the third, fourth, and fifth during downhill movement. A sample snapshot of the robots performance can be seen in Fig. 3.33 and a detailed video can be seen in the supplementary video.

Furthermore, to confirm robustness on terrain at 20° inclination, the robot performed 25 times uphill and 25 times downhill terrain. I conducted similar trials to evaluate robustness on flat terrain. The robot successfully avoided the obstacle 17 times uphill (68% success) and 22 times (88% success) downhill. The difference is due to the fact that, when walking uphill, the robot needs a higher swing to avoid the obstacle than when walking on the flat or downhill.

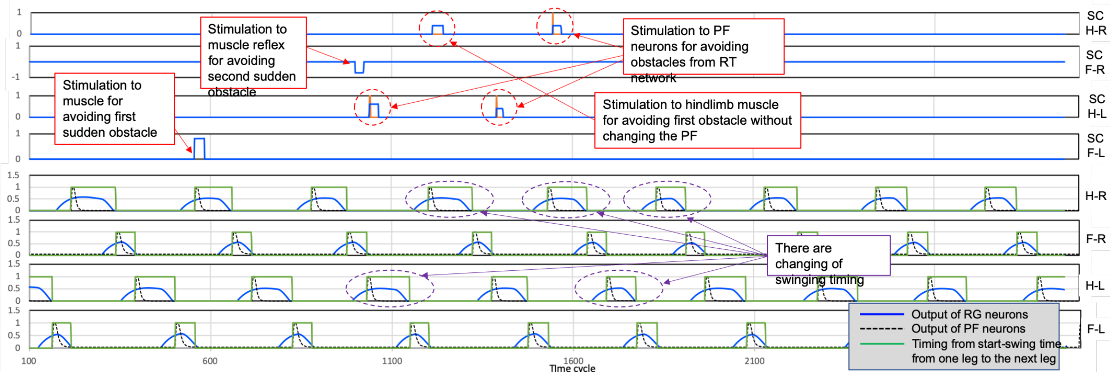
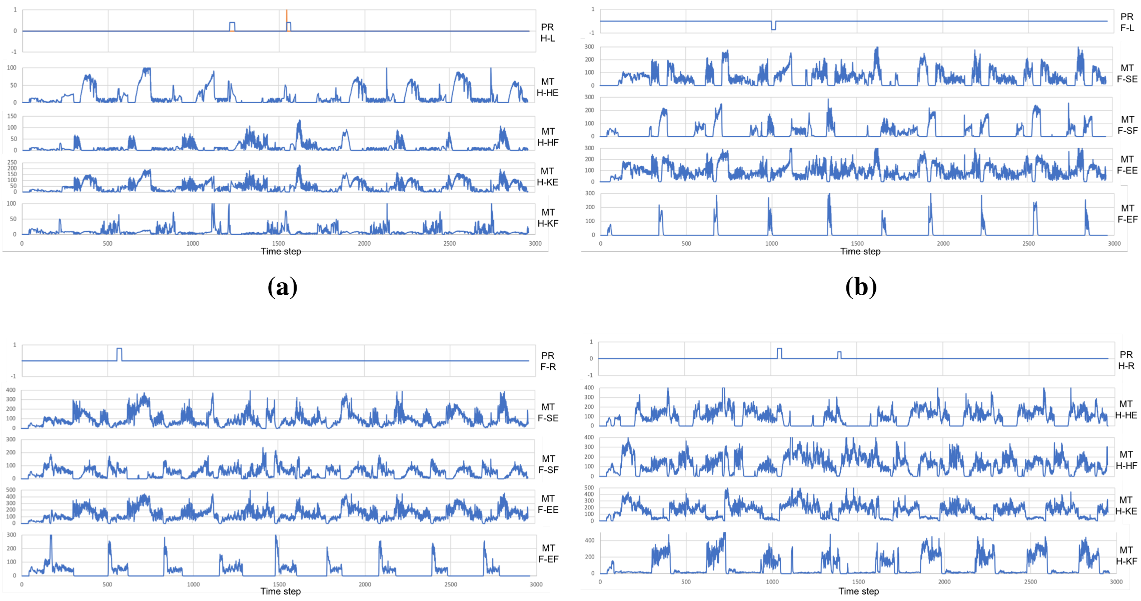


Figure 3.31: *Analysis of signal commanding and its effect on the CPG and movement timing*



(a)

(b)

(c)

(d)

Figure 3.32: *The recorded torque when avoiding the sudden obstacle. a) right-hindlimb. b) right-forelimb. c) left-forelimb. d) left-hindlimb.*

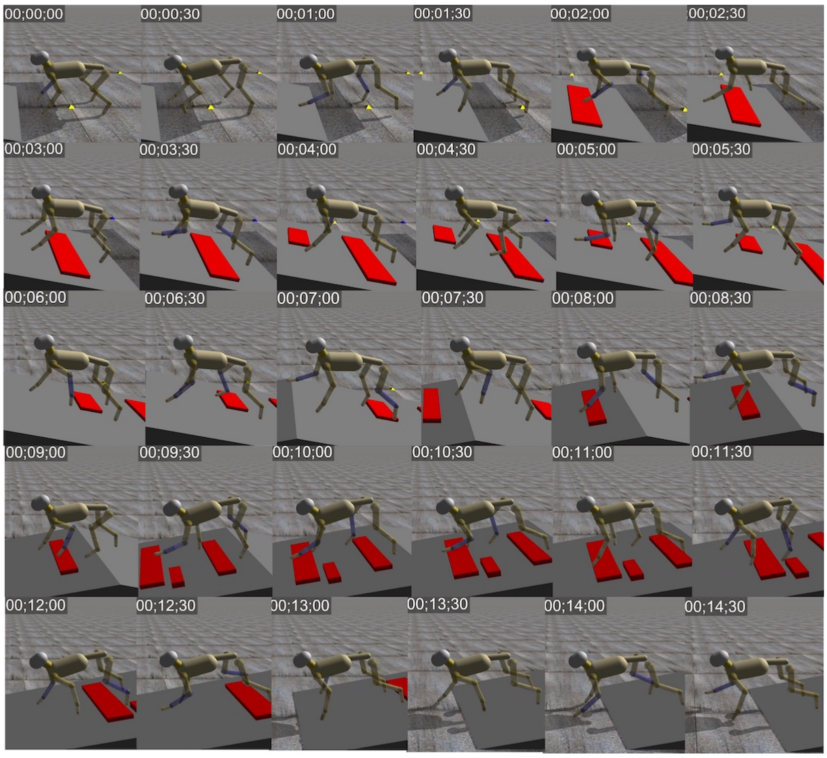


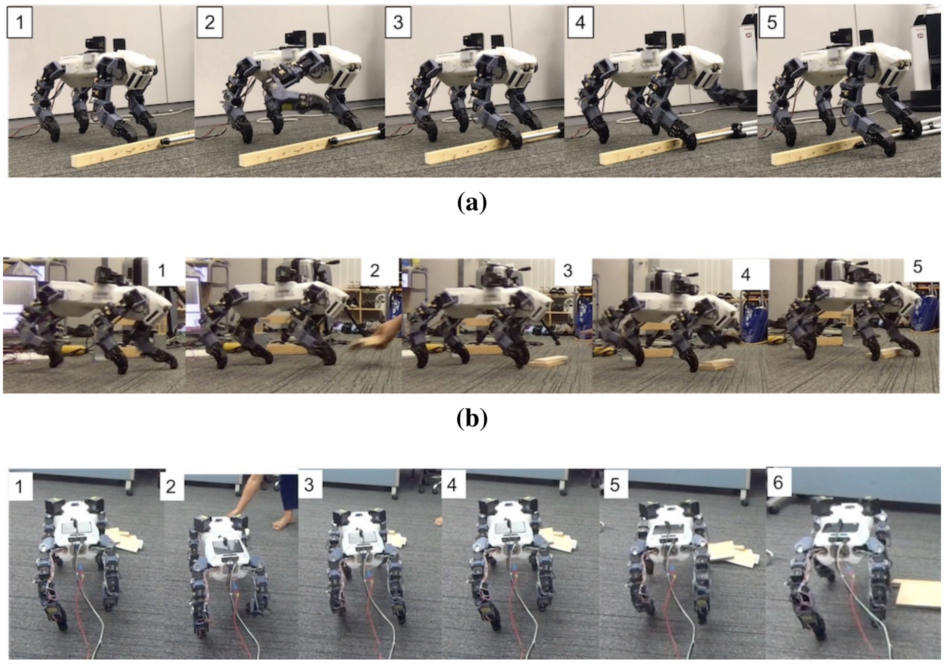
Figure 3.33: *The time series snapshot of the simulated robot performance in slope terrain with sudden obstacles given. The detailed robot performance can be seen in the supplementary video.*

3.3.6.3 Real Robot Implementation

After successfully implementing the simulation, I tested the proposed model in a medium-size cat-like robot (see Appendix A). Joints have an angle-based actuator. Therefore, I converted the value of muscle force to joint angle. In the first performance, I placed an obstacle before the robot started. In the second performance, I put the sudden obstacle exactly in front of the robot three times. In the third performance, I put a bigger obstacle a further distance in front of the robot. The result can be seen in Fig. 3.34. The robot successfully avoided the obstacle in all three performances. The video demonstration can be seen in the supplementary material.

3.3.7 Discussion on Integration Between Cognitive and Locomotion in Short-term adaptation

I proposed a model for neuro-cognitive locomotion in a neuro-musculoskeletal model. Our model integrates the external sensory information process with the locomotion generator. In this section, I focus on modelling the placing reflex to generate behavior for avoiding sudden obstacles in a muscle-based model. Our results suggest that it is worth viewing the



(a)

(b)

(c)

Figure 3.34: *Snapshot of real robot performance a) with obstacle in front of the robot b) with sudden obstacle exactly in front of the robot c) with bigger obstacle a little bit far in front of the robot.*

cognitive and muscle stimulation models as a single, unified system, rather than separately

I began this section by analyzing human and animal locomotion from a physiological view point that showed the role of cognition in locomotion [214, 166, 150]. Our viewpoint is also supported by neuro-activity experiments that demonstrate the contribution of supraspinal activity to motor neuron activation. Specifically, analysis of the corticospinal tract has shown that there is a low activity of CT in uniform area or on flat paths but, when obstacles stand in the way, CT activity is higher especially in the swing phase [56]. Thus, limb control to avoid obstacles requires cognitive processing of external sensory information [214]. From these foundations, I sought to mimic nature as I developed a locomotion model that included supraspinal commands to the locomotion generator. I implemented a descending process from external sensory process to limb movement generator. I applied the proposed model to a quadruped robot.

I implemented an attention model mimicking that of humans. The robot's external input uses point cloud data instead of visual information, however, this choice of representation does not influence the attention mechanisms implementation. Attention was represented using nodal density in a topological map structure, as had been successfully implemented in our previous research [246]. In this section, I built a dynamic attention model inspired by visual processing processes in nature. Topological structure has a low density for walking across flat, clear surfaces, and the density increases when there are obstacles. This mechanism is computationally efficient, with one process costing 0.00314 second. As the number of obstacles increased, the processing cost also increased. Our mechanism is, in this way, similar to the human visual process: when there are a lot of obstacles, cortex activity also increases [307].

The topological information is then processed to detect object affordances. In our proposed system, I do not need to recognize objects in detail: our model successfully avoided obstacles without detailed obstacle recognition. In the experiments, I assume that the objects should be avoided. However, the way to avoid depends on the object affordance detection. The robot avoids the obstacle by shortening the swing, changing the swing direction, lifting the swing, and changing the walking direction. Our results show that the locations and sizes of suddenly appearing obstacles can be recognized in 0.07 and 0.06 second for the first and second object, respectively. Our results also show an interesting integration of attention control, object affordance recognition, and movement commanding. In the case of unclear recognition, the object recognition stimulated increased attention to the suspected area. Once the object is recognized, the object affordance detector provides for an appropriate motor response.

The command process successfully commanded the forelimb to avoid sudden obstacles by stimulation through the CT network. While avoiding the obstacle, changes to the RG and PF signals modify the timing of the swing phase. In the robustness evaluation, the robot had a success rate of 82 percent on flat terrain, 68 percent on uphill terrain, and 88 percent on downhill terrain. The robot could easily avoid the obstacle when going downhill because it does not need much higher swinging movement. Vice versa, on the uphill terrain, the robot needs to swing its legs higher. A large experimental trial would be required to investigate the applicability of the proposed model. However, the present experimental results show the effectiveness and the importance of integrating cognition with the locomotion generator. For future work, the stability of the system should be improved in order to increase the success rates.

I have proved and showed that our model combines cognitive information and musculoskeletal model by integrating exteroceptive sensory information and low-level control muscle control in Section VII. The system can therefore respond to environmental changes in every time cycle. Existing systems that integrate perception and locomotion mostly deal with foothold planning, which limits adaptation to the much longer timescale of whole footsteps [101, 212, 308, 145, 50, 91, 112, 81]. While [301, 171] dealt with real-time obstacle avoidance, their system is unable to cope with obstacles that suddenly block an already-swinging limb. Furthermore, bio-inspired based locomotion model's short-term adaptation controllers consider only internal sensory information such as from posture, touching, and inertial information. Thus, the robot cannot respond to an obstacle before touching it. Our proposed system has the advantage in controlling the locomotion in every time cycle, which has been proved in several experiments avoiding unknown sudden obstacles.

More generally, our neuro-cognitive locomotion model shows a way to integrate neuro-musculoskeletal models with cognitive models for achieving dynamic locomotion. Such integration may have a sizeable impact not only in robotics but in also in medical applications such as in the development of active prosthetic limbs. Active prosthetics have been shown to increase patient confidence during ambulation [263]. By monitoring eye movement signals from the posterior parietal reach region, [193, 9] and using visual priority map[8], the neuro-cognitive locomotion model can be integrated into knee prosthetics to actively avoid sudden obstacles or suddenly changing swing patterns.

Chapter 4

Macroscopic Neuro-Cognitive Adaptation

In this chapter, we focusing on the system development in macroscopic level implies long-term adaptation. The system diagram can be seen in Fig. 4.1. We proposed the concept of MaSc which involves higher level controller. We present cognitive map building by using the topological structure based map reconstruction generated from Mesoscopic level. Cognitive map requires an integration with robot embodiment. Different embodiment of the robot may have difference cognitive map in coverage motion viewpoint. Cognitive map will be transferred to the path planning model. Then motion planning will be generated based intention, embodiment, and condition, by using neural based path planner. The model can find the best pathway and facilitate the safe movement of the robot. When the robot found an unpredictable collision, the path planner dynamically changed the pathway.

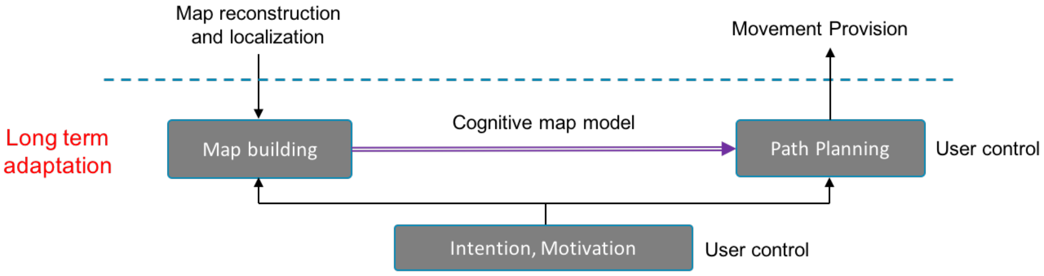


Figure 4.1: *Diagram of microscopic level of neuro-cognitive model*

4.1 Cognitive Map Buiding

We proposed a real time and continuous cognitive map building algorithm using topological structure as an input. This cognitive map model represents the coverage area that can be reach by some robot embodiment. It depends on the abiity and the embodiment of the robot. The model can be seen in Fig. 4.2. The robot posisiton and topological map information is achieved from localization and mapping model in the mesoscopic level. After that, the model analyze the successful rate of the robot travel from one node to another node in the topological map. After that, the

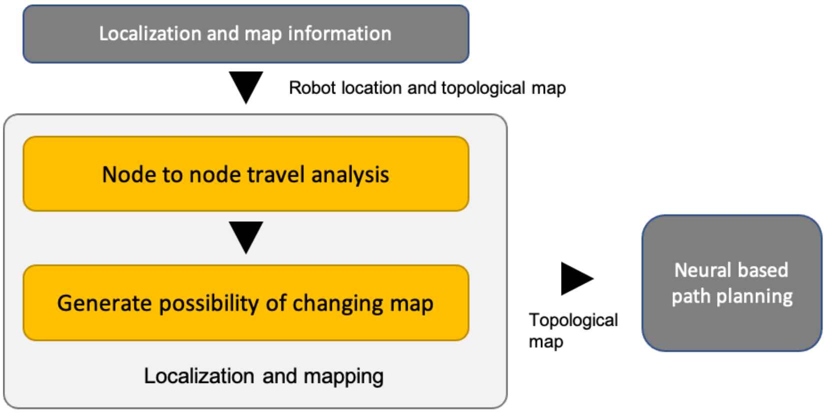

Localization and map information
Robot location and topological map
Node to node travel analysis
Generate possibility of changing map
Localization and mapping
Topological map
Neural based path planning

Figure 4.2: *Diagram of cognitive map model*

4.1.1 Node to node travel analysis

The module analyzes the energy required and the successful rate of of the robot movement from one node to another node.

4.1.2 Generate possiblity of changing map

Q : The nodes of maps (3D position (n), 3D surface vector (v), Label (l)) N_Q: Number of node of maps A : The nodes of GNG (3D position (n), 3D surface vector (v), Label (l)) N_A: Number of nodes of GNG s : Global position θ: Euler angle from earth

4.2 Neural-based Path Planning

During a disaster, the terrain, route, area, and also building are becoming unstructured, diverse, and challenging. It makes the rescuing process difficult and challenging for human or robot who is in charge in disaster area. The route in disaster area becomes unstructured, making it difficult for the rescuer whether human or robot to find the best and possible route in the rough and unstructured terrain. Since it is very dangerous for human to rescue in unstable environment, robot is the best choice for exploring and investigating the disaster area. Most researchers proposed wheeled mobile robot for rescuing in disaster areas [192, 231, 270], but legged robot is more effective in rough terrain [5]. Therefore, we applied this proposed algorithm in a four-legged robot which is equipped with several supporting sensors. Four-legged robots are more efficient than biped robots in the stabilization cases and are able to achieve the maximum movement speed compared to robots with more legs. Furthermore, 3-D path planning model is required in order to support and facilitate those robots while moving on rough terrain. Therefore, in this research, we propose an online 3-D path planning optimization based on neural activity.

Three dimensional path planning studies have used images for path planning and also applied multiclass support vector machines for obstacle avoidance [191]. By using this idea, the robot can generate the safe pathway. In [262], 3-D based path planning was proposed, and D* algorithm was modified in order to estimate the distance in sloping terrain. Beside that, Dogru et al proposed genetic algorithm for optimizing pathway with minimum energy required [54, 55]. However, [262, 54, 55] did not consider the unpredictable obstacle. Beside that, Kroumov et al solved the obstacle problem but there is still problem with concave 3-D obstacles [144]. 3-D path planning was also proposed for UAV movement algorithm [207, 240, 302]. These algorithms were applied in computer simulation. In the integration system, UAV or drone is used for generating the map. Some researchers used 3-D map reconstruction in order to acquire the 3-D map model and to find the best pathway based on the result of reconstruction map. The robot was equipped with laser range finder (LRF) or Kinect sensor in order to support the reconstruction algorithm. [31] also used LRF to generate the 2-D maps. This idea deals with static environment. Henry et al used Kinect camera in order to model 3-D indoor environment [102]. In the previous research, multi-resolution map was also used for decreasing the computational cost existing in high resolution map of path planning [286]. After that, a real-time feature extraction and segmentation method for a 3-D map [288] was proposed in order to increase the efficiency of the topological map. This map model can decrease the memory usage for reconstructing the map. Therefore, [288] is suitable to be combined for the proposed 3-D path planning model in the next stage.

In further cases of the environment model generated by LRF and Kinect sensor, the unpredictable surface such as friction, softness of the terrain, and unpredictable collision sometimes become the problem. Those sensors are used as initial data in path planning, therefore those ones only deal with static path planning. In the real cases, the robot deals with dynamic environment, and it should autonomously work through dynamic environment. When the algorithm deals only with static environment, the robot will get problem when it finds an unpredictable collision. Therefore, the dynamic path planning is important in order to deal with dynamic environment. Dynamic path planning will regenerate the path planning when the algorithm finds unpredictable condition or changing environmental condition. In order to support the dynamic path planning, either additional sensors in mobile robot or measurement tools for human are required to measure and detect unpredictable collisions (friction, obstacle, and softness of terrain) which cannot be considered in initial map generated by LRF or Kinect sensor.

Most mobile robots (wheeled and legged) are supplemented with the capability for finding the pathway. Some people used traditional algorithm such as: A*, D*, and Dijkstra algorithm in order to find the best pathway [159], [68], [201], [53], [158]. Ferguson et al proposed a modified D* algorithm in order to reduce the path cost in non-uniform environment. An interpolation equation was proposed in order to cut the inefficient pathway [68]. However, in [68]'s algorithm more computational cost was required. In common cases, D* path planning algorithm is faster than A*, but [103] modified the adaptive A*, therefore it could be faster than D* in some cases. These algorithms [103, 68] required predefined travel cost, therefore these algorithms seem difficult to cover unpredictable obstacles.

However, the traditional algorithm of path planning requires the determination of the path planning rule and it uses a recursive algorithm. Therefore, these algorithms require high computational cost. Bio-inspired algorithm provides a natural process that is able to dynamically generate the best pathway [234, 222]. Some researchers have proved the effectiveness of neural based model for path planning problems [88], [318], [321], [222]. In [88], Hopfield method was applied in a neural network to generate the pathway with obstacle avoidance. This proposed model was applied in a 2-dimensional simulation. Neural based approach can also be applied for complete coverage path planning with obstacle avoidance [318]. Quoy et al proposed the control model in mobile robot by using dynamical neural networks based on the neural field formalism that was applied in the mobile robot path planner [222]. Most of the researchers applied pulsed neural network in order to find the efficient pathway [220], [221], [334, 316]. They focused on the inner activity of the neuron. Qu et al proposed pulsed neural network to generate real time collision free path planning [220]. Zhang et al applied a simple pulse coupled neural network to decrease the computational cost, but the defined travel cost is required in this case [334].

Furthermore, modified pulsed neural networks were proposed by several researchers in order to increase the performance of the path planning algorithm [156, 95, 221]. In [95], quick optimization process of path planning was proposed by using PCNN model. However, there is no efficient model to reduce the number of evolved neurons, there are many neurons required for representing the best path way. [331] presents a coupled neural network, called output-threshold coupled neural network (OTCNN). However, this algorithm was very hard to be implemented in the real cases and it was improved in [221] by proposing a new modified model of Pulse-coupled neural networks (M-PCNNs). This model was improved by Liu et al by solving the K Shortest Paths (KSPs) problem [164]. In addition, the neural network based path planner can also be applied in non-holonomic mobile robot [321]. Furthermore, shunting based neural model first proposed by Hodgkin and Huxley [108] and refined by Grossberg in the neural mechanism [94] was also used to solve path planning model [320, 319]. However, the current neural based models [334, 320, 319, 164, 321, 156] have not considered rough terrain with undefined travel cost or weighting cost.

In this proposed research, we create a new model of dynamic path planning based on neuronal activity. A 4-legged robot is applied to prove the effectiveness of the proposed path planning model. The proposed model is expected to be online generated with obstacle avoidance and applied with undefined travel cost. These advantages do not exist in the current state-of-the-art research. Nevertheless, an integrated system is required for supporting the proposed model.

The main contribution of the proposed research is to use the natural mechanism of the human brain for generating the online path planning in 3-D rough terrain with undefined travel cost. Our proposed reserach not only emphasizes the inner state process of the neuron, but also the development process of the neurons in the brain. Based on [39], the development of the brain involves the synaptic pruning process. The weak synapses will be deleted based on their synaptic efficacy and replaced by efficient synapses [40]. These processes are applied in this proposed path planning optimization process. Other contribution is to generate dynamic 3-D path planning with undefined synaptic weight. This proposed model let the neuron generate the signal based on the terrain condition in order to create the synaptic weight connection.

4.2.1 The Proposed Model

The proposed model will be applied as the path planner in the robot movement on rough terrain. The full design of the system is depicted in Fig. 4.3. In the first step we generate the structure of the map by using Kinect sensor and send it to the proposed system by using radio communication. In the second and third steps, the proposed path planning are performed. After that the path planner sends the best pathway to the robot by using Bluetooth communication. In the fifth step, the robot moves and updates the current position and also updates

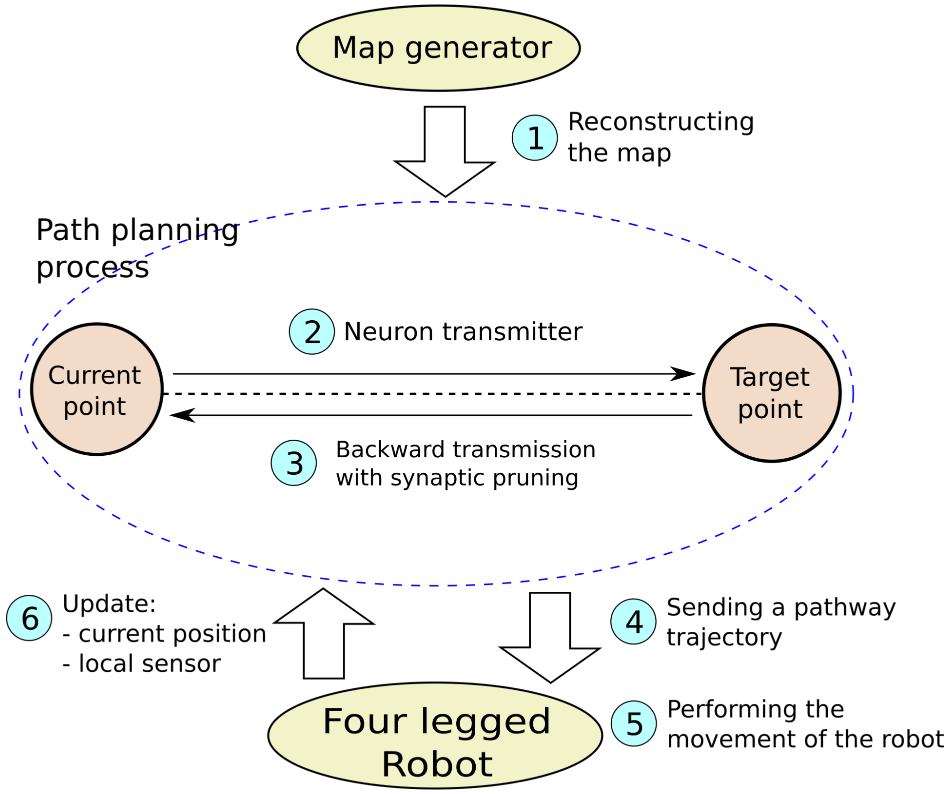


Figure 4.3: *Design of the proposed model*

some map information. The details will be explained in Section 4.2.3.

In this section we explain the novel neuro-activity based path planner. There are two algorithm processes in this proposed model, the forward transmission to construct the neuron connections for finding the possible way and the backward neuron transmission with synaptic pruning model for finding the best pathway from the current position to the target position and for reducing inefficient neurons. Dynamic path planning is also considered in this proposed model in order to respond to unpredictable obstacles. In the process mechanism, first the starting point neuron generates the signal to the neighbor neurons and calculates the synaptic weight value. After the signal triggers the target neuron, then the transmission process is stopped. After that, the backward signal process with optimizing the number of synapses is performed.

4.2.2 Neural Forward Transmission Based Model

The aim of forward transmission is to construct and build the neuron connections in order to find the possible way. In neural transmission studies, a signal is transmitted from one neuron to another neuron through electrical and chemical process (neurotransmission) occurring in the synapse. There are 2 types of synapse, electrical synapse (dendrite - dendrite) and chemical synapse (axon - dendrite). Chemical neurotransmitter requires releasing neurotransmitter into the synaptic cleft before a synaptic potential can be produced as input to the other cell. Membrane potentials are generated by chemical and electrical synapses function as a neuron's input.

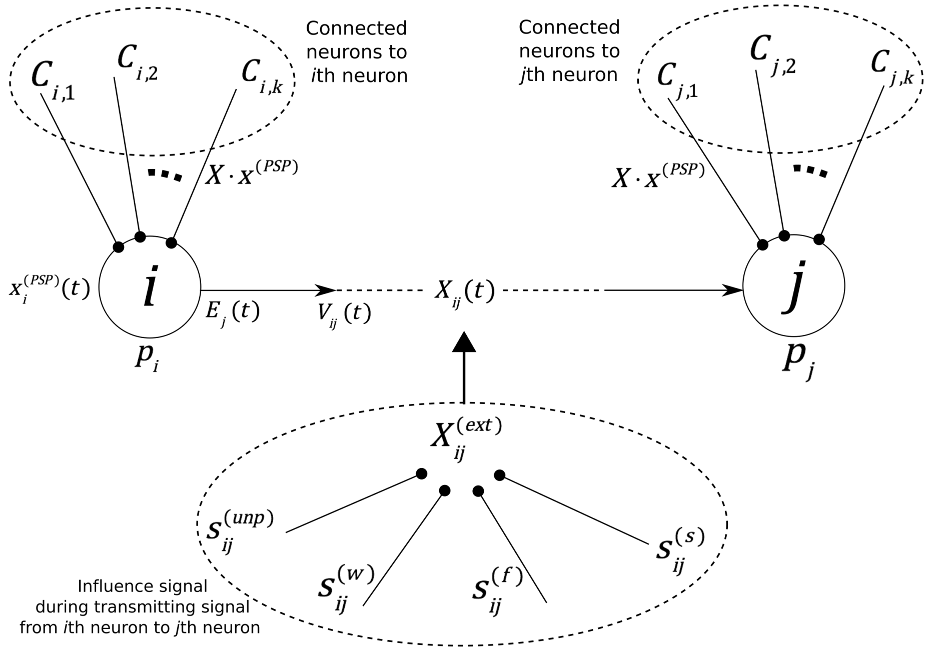


Figure 4.4: *Neuro-transmission model*

The neuron typically has thousands input synapses and produces an action potential when the post-synaptic potential passes the threshold of membrane potential. Once there, it will trigger its own neurotransmitter release, which will cause a synaptic potential in the new post-synaptic neuron. The action potential will travel along the axon until it reaches the output (chemical) synapse at its axon terminal.

In the spiking model, there are timing models proposed for post-synaptic potential model [108, 170]. We use an exponential equation in order to model the decreasing energy. We modified and applied the neuron transmission activity in order to generate the best path planning in 3 dimensional rough terrain with unpredictable obstacle.

In this algorithm we use a forward transmission model depicted in Fig. 4.4. The strength of signal diversity from the ith neuron to the jth neuron is represented by X_{ij}. We assume, that the synaptic weights between each neuron are not defined. The strength of signal diversity is influenced by the potential energy in source neuron, the external influence between 2 neurons, and the internal disturbance. The strength of signal diversity decides whether the neuron to be connected or not as calculated in Eq. (4.1), where notation A represents the inhibitory effect calculated in Eq. (4.2). The potential energy in the ith neuron is represented by $x_i^{(PSP)}(t)$ and the external influence between the ith and the jth neuron is represented by $X_{ij}^{(ext)}(t)$. V_{ij} is the potential force from the ith neuron to the jth neuron.

In the transmission process, after certain neuron is activated (firing), the neuron is preparing the impulse signal, therefore the signal can reach the target neuron. If the neuron is already activated, then the preparation state to certain neuron is stopped. After the value of signal reaches the threshold of the target signal, then the firing process is performed. The neuron's capability for firing the signal is computed in Eq. (4.3), where $\mathbf{C}_i$ is a matrix representing the possible connection between the ith neuron and other neurons. In order to find the possible connection, we used a simple surrounding area algorithm which is computed in Eq. (4.10) and explained in Algorithm 9.

$$\dot{X}_{ij} = X_{ij}(t-1) + V_{ij}(t) - A \tag{4.1}$$

$$A = \sum_{j=1, j\neq i}^{N} X_{ij}(t-1) \cdot x_i^{(PSP)}(t-1) \tag{4.2}$$

In the initial condition, the strength of signal diversity ($\mathbf{X}$) and the potential energy ($\mathbf{x}^{(PSP)}$) are zero in each neuron and the time signal impulse ($\mathbf{t}^{(imp)}$) is 10000. When the path planning generation starts, the starting neuron is firing by the given parameter $t^{(imp)}$ representing the starting neuron as zero. It causes the parameter $\mathbf{x}^{(PSP)}(t)$ triggered, as calculated in Eq. (4.3), where $P_i^{(PSP)}$ is the maximum potential value in each neuron. After the neuron is fired, then the neuron generates the release energy (E_i) depending on the derivative of potential energy degradation at time signal impulse $(dx_i^{(PSP)}/dt_i^{(imp)})$.

During the signal transferring process from the ith neuron to the jth neuron, signal strength X_{ij} is supported by potential force V_{ij} which is influenced by $h_{ij}^{(ext)}$, where V_{ij} is calculated in Eq. (4.6). If the signal strength X_{ij} is greater or equal than the distance between 2 neurons $\|\mathbf{r}_i - \mathbf{r}_j\|$ then the jth neuron is firing and $t_j^{(imp)} = 0$. If the signal strength is still lower than required, then $t_j^{(imp)}$ is increased and the potential energy is decreased. In order to acquire the travel time from the ith to the jth neuron, variable T_{ij} is calculated in Eq. (4.4).

$$t_i^{(imp)} = \begin{cases} 0 & \text{if } X_{ij} \geq \|\mathbf{r}_i - \mathbf{r}_j\|, j \in \mathbf{C}_i \\ t_i^{(imp)} + 1 & \text{otherwise} \end{cases} \tag{4.3}$$

$$T_{ij} = \left\{ \begin{array}{ll} t_i^{(imp)} & \text{if } X_{ij} \leq \|\mathbf{r}_i - \mathbf{r}_j\|, j \in \mathbf{C}_i \end{array} \right. \tag{4.4}$$

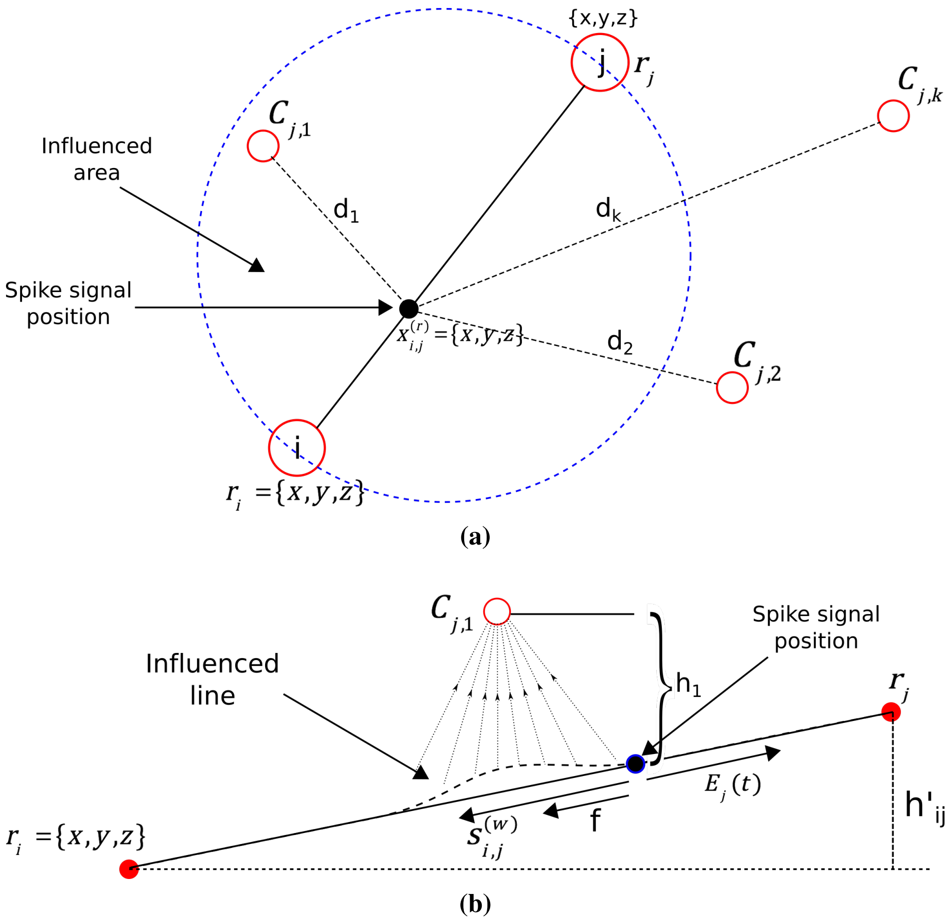


Figure 4.5: *Environmental influence represented by surrounding neurons, where each neuron has 3-D coordinate information (a) from top view (b) side view*

$$x_i^{(PSP)}(t) = P_i^{(PSP)} \cdot e^{\left(\frac{-\mathbf{t}_i^{(imp)}}{\tau_i} \right)} \tag{4.5}$$

$$\frac{dV_{ij}(t)}{dt} = V_{ij}(t-1) + (E_i - X_{ij}^{(ext)}(t)) \tag{4.6}$$

Figure 4.5 shows how the environmental condition influences the spike signal travel. Figure 4.5a illustrates the neuron connection from top view. Only the neuron inside the influenced area can influence the spike signal travel. In Fig. 4.5b, the height difference between 2 neurons causes the weight force ($s^{(w)}$) and the friction force ($s^{(f)}$) to also influence the travel of spike signal.

The external influence signal between the ith neuron and the jth neuron at certain time sampling t is represented by $X_{ij}^{ext}(t)$ and computed in Eq. (4.7). There are 4 influence parameters calculated in Eq. (4.7), where $s_{ij}^{(unp)}$ represents the unpredictable obstacle, $s_{ij}^{(w)}$ represents the influence of weight force of the robot which is illustrated in Fig. 4.5, $s_{ij}^{(f)}$ is the constant value representing the predicted friction value, and $s_{ij}^{(s)}$ represents the influence of the other neurons against the signal between the ith neuron and the jth neuron. In Eq. (4.8), $s_{ij}^{(w)}$ is the external disturbance representing the self influence of the robot's weight between the height position of the ith neuron and the jth neuron depicted in Fig. 4.5b, where m is the robot's mass and g is the gravity acceleration. In Eq. (4.9), the effect of the height of the neuron around the signal line is calculated, where $d_{C_{j,k}} = (\mathbf{r}_j - \mathbf{x}_{i,j}^{(r)}) \cdot (\mathbf{r}_{C_{j,k}} - \mathbf{x}_{i,j}^{(r)})$, $h_{C_{j,k}} = (\mathbf{r}_{C_{j,k}} - \mathbf{x}_{i,j}^{(r)})[0\ 0\ 1]$, $\mathbf{x}_{i,j}^{(r)}$ is the location vector of the spike signal during neuron transmitting from the ith neuron to the jth neuron, and d_{max} is a constant value representing the maximum influence of the neuron signal.

$$X_{ij}^{ext}(t) = s_{ij}^{(unp)} + s_{ij}^{(w)} + s_{ij}^{(f)} + s_{ij}^{(s)} \tag{4.7}$$

$$s_{ij}^{(w)} = m \cdot g \cdot [0\ 0\ 1] \frac{\mathbf{r}_i - \mathbf{r}_j}{\|\mathbf{r}_i - \mathbf{r}_j\|} \tag{4.8}$$

$$s_{ij}^{(s)} = \sum_{k=1}^{C_{j,0}} d_{C_{j,k}} \cdot h_{C_{j,k}} \cdot \exp\left(- \left(\frac{\|\mathbf{r}_{C_{j,k}} - \mathbf{x}_{i,j}^{(r)}\|}{d_{max}} \right) \right) \tag{4.9}$$

h_{ij}^{ext} represents the external influence between the ith neuron and the jth neuron. $\mathbf{C}_i$ contains the neuron IDs which are possibly connected to the ith neuron. $\mathbf{C}_i$ is defined based on Eq. (4.10), where $\mathbf{C}_{i,0}$ is the number of possible connections to the ith neuron. The spreading of possible connections is influenced by γ^{inf} representing the strength of scope.

$$\mathbf{C}_{i,k} = j, \quad \mathbf{C}_{i,0} = k \qquad \text{if } \|\mathbf{r}_i - \mathbf{r}_j\| \leq \gamma^{inf}, \quad j \in \mathbf{C}_i, \quad k{+}{+} \tag{4.10}$$

The main process of the proposed algorithm is explained in Algorithm 10, which starts with initializing all components and finding the possible interconnection in each neuron explained in Algorithm 9.

4.2.2.1 Synaptic Pruning with Backward Transmission Model

In human brain development, during childhood, more than half of the synapses in the human brain are removed until puberty phase, which process is called synaptic pruning. In 1979, Huttenlocher showed the evidence for synaptic degradation in some areas of the human brain [117]. This process improves the performance of an associative memory network with limited synaptic resource. The strategy of the synaptic pruning is based on the weakness of

Algorithm 8 Pathway generation represented by Fig. 4.4

```
Initialization; $i \leftarrow 1$ to $N_{max}^{(neuron)}$
Possible interconnection generation in $i$th neuron explained in Alg. 9 $i \leftarrow 1$ to $N_{max}^{(neuron)}$
$k \leftarrow 1$ to $C_{i,0}$
$j \leftarrow C_{i,k}$ $F_{i,j}(t) > 0$
calculate $X_{i,j}^{ext}(t)$ by Eq. (4.7)
calculate $X_{i,j}(t)$ by Eq. (4.1)
calculate $T_{i,j}(t)$ by Eq. (4.4) $X_{i,j} \geq \|\mathbf{r}_i - \mathbf{r}_j\|$
$i$th neuron spikes
$t_j^{(imp)} = 0$
```

Algorithm 9 Possible connection searching algorithm

```
input: $i$ as neuron ID
$C_{i,0} \leftarrow 0$
$k \leftarrow 0$ $j \leftarrow 1$ to $N_{max}^{(neuron)}$ $i \neq j$
$d \leftarrow \|\mathbf{r}_i - \mathbf{r}_j\|$ $d \leq \gamma_{inf}$
$C_{i,(k+1)} \leftarrow j$
$k++$
$C_{i,0} \leftarrow k$
```

the synapses [39].

In this grand system, we use communication system where the size of data causes the accuracy and the speed of transmission. In the proposed path planning, the number of synapses is based on the signal transmission process. If we have a big number of neurons representing the map, then the number of synapses will increase. Therefore, we will optimize the number of synapses representing the path planning trajectory by using the synaptic pruning mechanism.

In the mathematical model shown in Eqs. (4.11) and (4.12), we use neural regulation-driven synaptic modification (NRSM) first proposed by Chechik et al [40].

$$\mathbf{W}^{'}(t+1) = \mathbf{W}(t) - (\mathbf{W}(t))^{\alpha}\mu(t) \tag{4.11}$$

$$\mathbf{W}(t+1) = \mathbf{W}^{'}(t+1)\frac{f_i^0}{f_i^t} \tag{4.12}$$

In this proposed model, notation $\mathbf{W}_i$ represents the age of the ith neuron which also represents the age of the weight connections to the ith neuron. In Eq. (4.11), $\mu(t)$ is the sigmoid function sgn$(\eta - \varepsilon)$ at time step t, where η is the threshold value of the neuron, and synaptic reduction, ε is the degree of pruning calculated in Eq. (4.13), and power α defines the degradation dimension parameter chosen in the range $[0,1]$. In Eq. (4.13), $\alpha^{(l)}$ and $\alpha^{(d)}$ are constant variables representing the degree of neuron distance effect and the deviation angle effect, respectively and $h_i^{PSP}(t)$ is the presynaptic spike output in the ith neuron. In this SP model, we set f_i^0/f_i^t that represents the input field of the ith neuron at time step t in Eq. (4.12) as one.

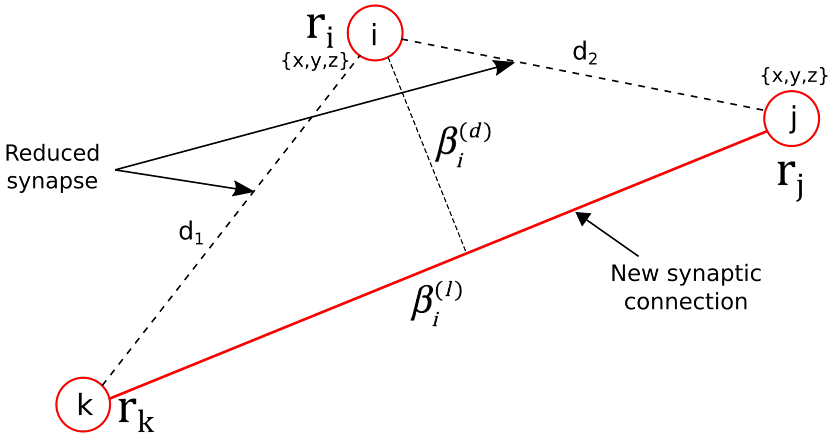


Figure 4.6: *Illustration of the components that influence the synaptic pruning*

$$\varepsilon = h_i^{PSP}(t)\cdot(\alpha^{(l)}\beta_i^{(l)} + \alpha^{(d)}\beta_i^{(d)}) \tag{4.13}$$

$$\beta_i^{(l)} = \|\mathbf{r}_k - \mathbf{r}_j\| \tag{4.14}$$

$$\beta_i^{(d)} = \|(\mathbf{r}_k - \mathbf{r}_i) - ((\mathbf{r}_k - \mathbf{r}_i)\cdot\mathbf{n})\mathbf{n}\| \tag{4.15}$$

In this stage, we calculate the influence component in synaptic pruning process which is depicted in Fig. 4.6. We consider the length of the point $\beta_i^{(l)}$ calculated in Eq. (4.14) and the deviation angle $\beta_i^{(d)}$ calculated in Eq. (4.15) in the synaptic pruning model for the proposed path planning where its illustration is shown in Fig. 4.6. In the future we will consider the effect of the neighbor neuron in synaptic pruning. In Eq. (4.15), $\mathbf{n}$ is the unit vector of $(\mathbf{r}_k - \mathbf{r}_j)$.

The synaptic pruning model is performed during the backward signal processing. When the neuron is firing, then the SP decision is performed. When the age of the ith neuron $\mathbf{W}_i$ is getting zero then the connections to that neuron are reduced, and a new connection is created to the previous firing neuron that is connected to the ith neuron.

In backward signal processing, we use a simple spiking neural network depicted in Fig. 4.7, whose mathematical model can be seen in Eqs. (4.17), (4.18), (4.19), (4.20). After the forward signal processing finishes, then the neuron at target point fires the impulse sig-

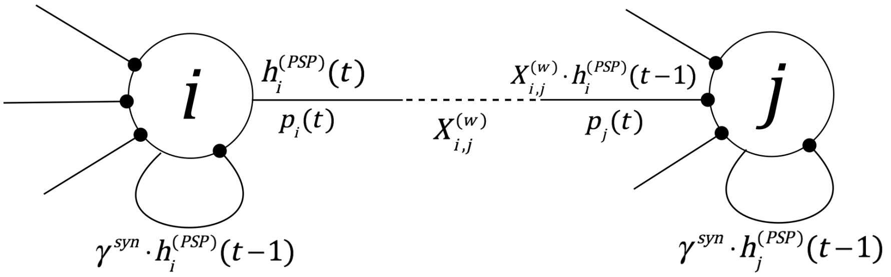

i
$h_i^{(PSP)}(t)$
$p_i(t)$
$X_{i,j}^{(w)}$
$X_{i,j}^{(w)} \cdot h_i^{(PSP)}(t-1)$
$p_j(t)$
j
$\gamma^{syn} \cdot h_i^{(PSP)}(t-1)$
$\gamma^{syn} \cdot h_j^{(PSP)}(t-1)$

Figure 4.7: *Simple spiking neuron model for backward transmission*

nal. The signal will be generated to the neuron at starting point. The backward signal will be calculated in Eqs. (4.17) and it is influenced by the synaptic weights calculated in the forward transmission. h_i is calculated in Eq. (4.17), where $h_i^{syn}(t)$ includes the weighted pulse outputs from the other neurons which is calculated by Eq. (4.18) and the reduced value of the internal state in the previous time step which is calculated by Eq. (4.19).

In Eq. (4.18), $X_{ji}^{(w)}$ is a weight from the jth neuron to the ith neuron calculated from the travel time of neuron presented by variable T_{ij} calculated in Eq. (4.4), $h_i^{PSP}(t)$ is the Post-Synaptic Potential (PSP) approximately transmitted from the ith neuron at the discrete time t, and N is the number of neurons. The presynaptic spike output is transmitted to the connected neuron through the weight connection. $h_i^{PSP}(t)$ is calculated in Eq. (4.20), where $t_i^{(f)}$ is the last firing of the ith neuron, and τ_i is a parameter which influences how long the firing of the neuron has an effect to the connecting neurons. When the internal state (membrane potential) h_i in the ith neuron is higher than the threshold Θ, then the ith neuron fires the pulse signal calculated as follows:

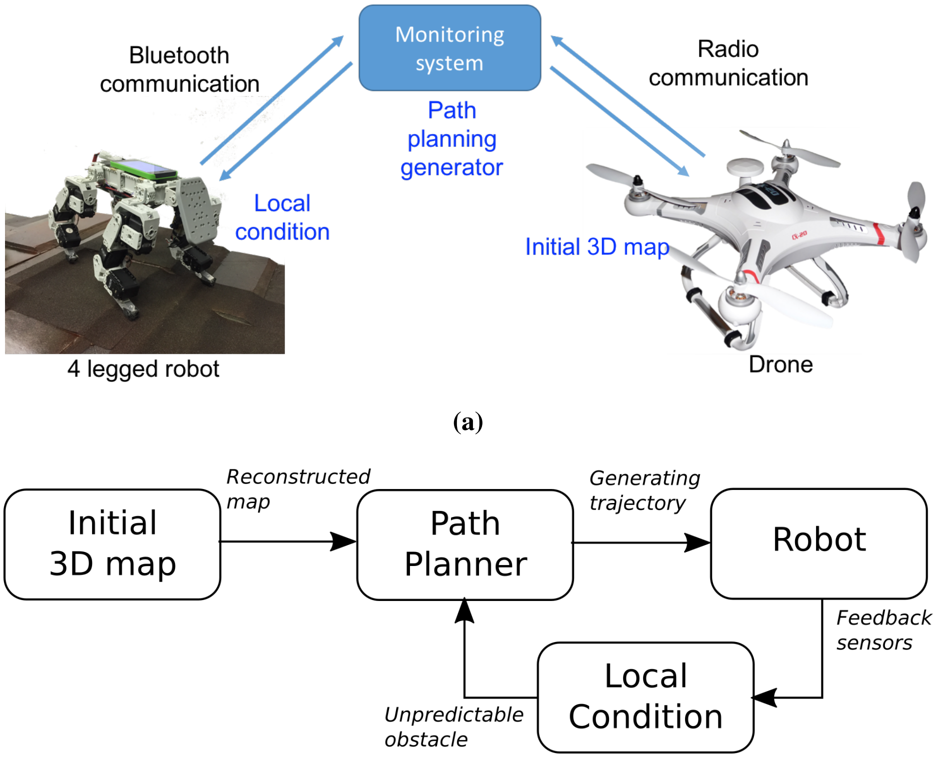
$$p_i = \begin{cases} 1 & \text{if } h_i \geq \Theta \\ 0 & \text{0therwise} \end{cases} \tag{4.16}$$

$h_i^{ref}(t)$ is used for representing the refractoriness of the neuron. This means that after the neuron fires, its internal state value is decreased using the refractoriness component, in order to avoid the continuous firing of the neuron within a short time. In Eq. (4.19) $R > 0$, γ^{ref} is a discount rate of h_i^{ref}, and $0 \leq \gamma^{ref} \leq 1$.

$$h_i(t) = \tanh(h_i^{syn}(t) + h_i^{ref}(t)) \tag{4.17}$$

$$h_i^{syn}(t) = \gamma^{syn} \cdot h_j^{(PSP)}(t) + \sum_{j=1, j \neq i}^{N} \mathbf{X}_{ij}^{(w)}(t+1) h_j^{(PSP)}(t) \tag{4.18}$$

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } p_i(t-1) = 1 \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & \text{otherwise} \end{cases} \tag{4.19}$$



(a)

(b)

Figure 4.8: *Grand design of the integrated system (a) connection model (b) system model*

$$\mathbf{h}_i^{(PSP)} = e^{\left(-\frac{t-t_i^{(f)}}{\tau_i}\right)} \tag{4.20}$$

4.2.3 Application design

In this part we explain the integration system in order to support the proposed model. There are three supporting systems required for this proposed model, which are drone, server monitor, and robot. The drone is for generating the map and sending it to the server monitor via radio communication. It captures the area by using Kinect camera. The server monitor processes the proposed path planning based on the local condition that is sent from the robot and the primary map data sent from the drone. In order to find the unpredictable obstacle, the robot is equipped with ultra sonic (US) sensor and touch sensor, and periodically sent those sensor data to the server monitor via Bluetooth communication. The systems integration is depicted in Fig. 4.8.

4.2.3.1 Robot Design

In this research, we applied the proposed path planner model for walking trajectory generation of a 4-legged robot with 12 degrees of freedom depicted in Fig. 4.10 and its joint structure is depicted in Fig. 4.9, where there are 3 degrees of freedom (hip-x, hip-y, and knee joint) in each leg. The robot's size is approximately $30cm \times 18cm \times 20cm$, and its weight is 2 kg. Its mechanical frame is built using Bioloid Comprehensive Kit. The locomotion system and its stability have been developed in the previous research [259] combined with trajectory based locomotion [177, 253].

In order to support the proposed path planner and detect unpredictable collision, the robot is equipped with additional sensors, which are 4 touch sensors and 2 ultra sonic sensors.

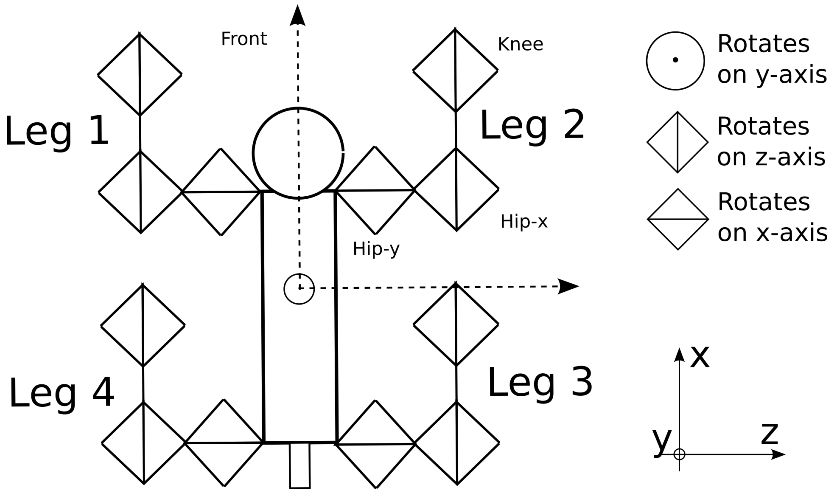


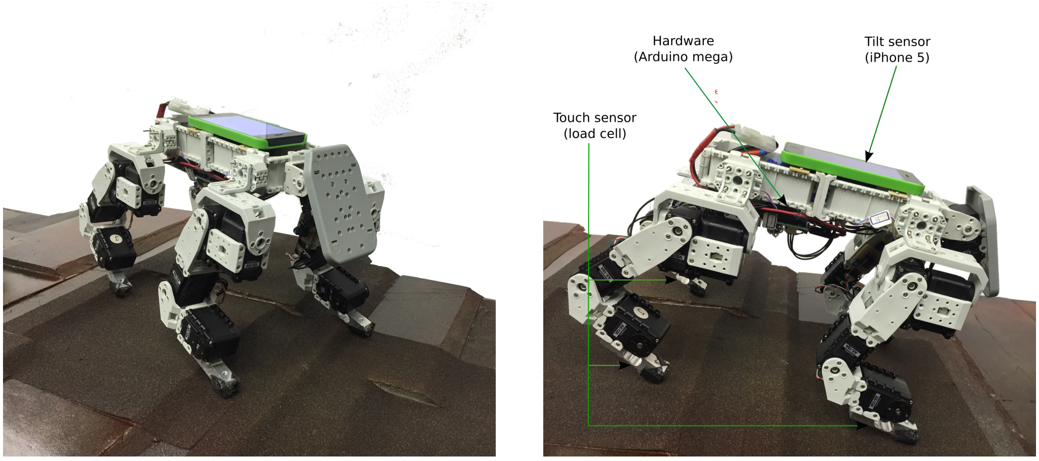
Figure 4.9: *Joint structure of the robot*

4.2.3.2 Integrated System

By using the proposed neuro-cognitive model, the possibility of the implementation of the dynamic path planning in disaster area is very high. The connection model of the supporting system is illustrated in Fig. 5.23a, where each system is integrated through wireless connection. In the grand application, we use Bluetooth connection for communicating between the robot and the monitoring system, and radio communication for communicating between the drone and the monitoring system.

4.2.4 Experimental Results

In the experimental result, we illustrate the effectiveness of the proposed neuro-activity based path planning model in both grid map model and topological map model. We show the



(a) **(b)**

Figure 4.10: *Design of the real robot a) from the front; b) from the left side*

signal transmission model and the synaptic pruning that is used in this proposed model. We create artificial terrain as the experimental object and show the effectiveness of the model responding to the environmental changing. In order to show the integration system, we applied the proposed model in a 4-legged robot movement trajectories in computer simulation ODE. Moreover, we implemented the proposed model in a real 4-legged robot, in order to show the feasibility in the real case.

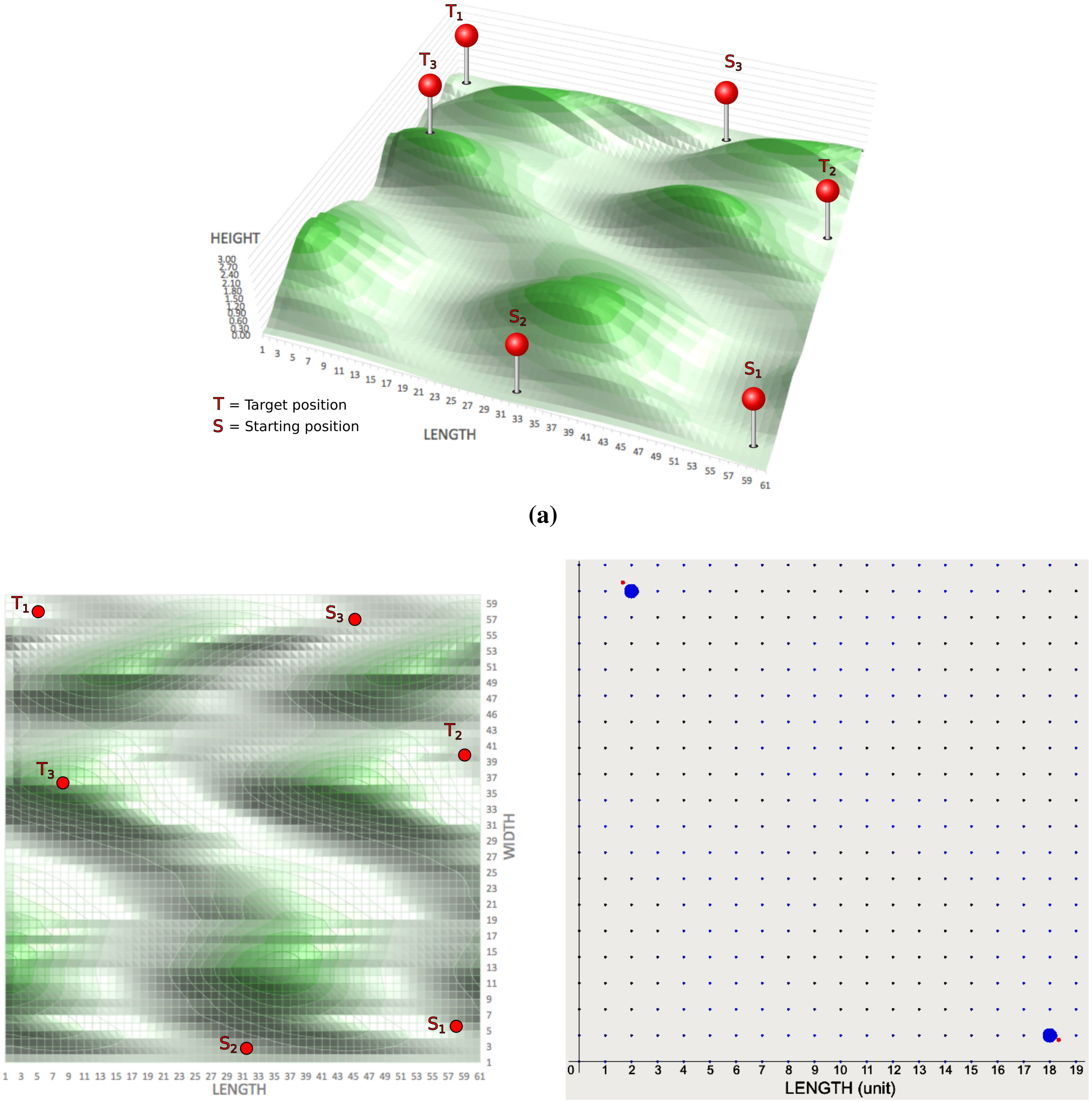
Table 4.1: *Parameter Values*

Parameter	d_{max}	$P_i^{(PSP)}$	γ^{syn}	γ^{ref}	R	Θ	$\alpha^{(l)}$	$\alpha^{(d)}$
Value	1.2	30	0.05	0.2	10	0.8	0.3	0.7

4.2.4.1 Neuron Transmission and Synaptic Pruning Experiment

In this experiment we create the illustration of the 3-dimensional uneven terrain. The artificial map can be seen in Fig. 4.11. In order to implement the proposed path planning, we converted the map data into grid map model and topological map. Furthermore, we compare the trajectory resulted by the proposed model. The parameters of FT and SP-BT model used in this experiment are shown in Table 4.1.

In this path planning we design the backward transmission with short term impulse firing signal. We put the same starting point and target point in grid map experiment and topological map experiment, both of them having 3 scenarios in this experiment with different starting point and target point which can be seen in Fig. 4.11. We choose the most difficult combinations of starting point and target point. In the first experiment, the starting point was



(a)

(b)

(c)

Figure 4.11: *Experimental terrain (a) perspective view (b) top view (c) Grid map model of experimental terrain*

$\{0.5, 11\}$ and the target point was $\{11.0, 1.0\}$, in the second experiment, the starting point was $\{0.2, 7\}$ and the target point was $\{6.5, 11.5\}$, and in the third experiment, the starting point was $\{11.2, 9\}$ and the target point was $\{7.0, 1.0\}$.

Grid map model In the grid map model, we adjust the resolution of the grid map model. Figure 4.11c displays a grid map model whose resolution is 20 x 20 unit. The different color in each point represents the height of the point. When the map is looked from the perspective view in Fig. 4.11a, the difference between the heights of each point can be seen. In this grid map model based path planning, there are 400 neurons representing the map model, since one neuron represents one point.

In these experiments, we show the difference between path planning model without synaptic pruning model and with synaptic pruning model as shown in Fig. 4.12.

The experiments in Fig. 4.12 confirm the effectiveness of the forward neuron transmission model in generating the possible pathway with undefined travel cost. The backward transmission also successfully formed the pathway based on the connection generated by forward neuron transmission.

From the result shown in Fig. 4.12, the path plannings using neuron transmission model with synaptic pruning model represent the path with smaller number of neurons. There are 25 neurons, 27 neurons, and 17 neurons representing the pathway without SP-BT model and 13 neurons, 14 neurons, and 10 neurons representing the pathway with SP-BT model in the first, second, and third experiment, respectively. SP-BT model proposed in this path planning model can effectively deactivate the weak neurons and also reduce the synapse which is connected to that neurons. In the case of huge map, this SP-BT model can reduce the size of the data representing the pathway therefore the communication weight can be reduced between the monitoring system and the robot.

Topological map model In the topological map based model, we reconstruct the map shown in Fig. 4.13 by using GNG proposed in [288]. By using this topological based map, the number of neurons required is smaller than in the grid map model. In the topological based map there are 165 neurons. In this experiment we set the maximum potential value $P^{(PSP)}$ as 45, 50, 60 in the first, second, and third experiment.

In these experiments shown in Fig. 4.14, we present the performance of the proposed path planning model. The SP-BT model was also applied in topological map model which is depicted in Fig. 4.14b. The SP-BT model reduced the number of evolved neurons in the pathway from 16 neurons to 8 neurons. The spike signal of neurons in this experiment can be seen in Fig. 4.15.

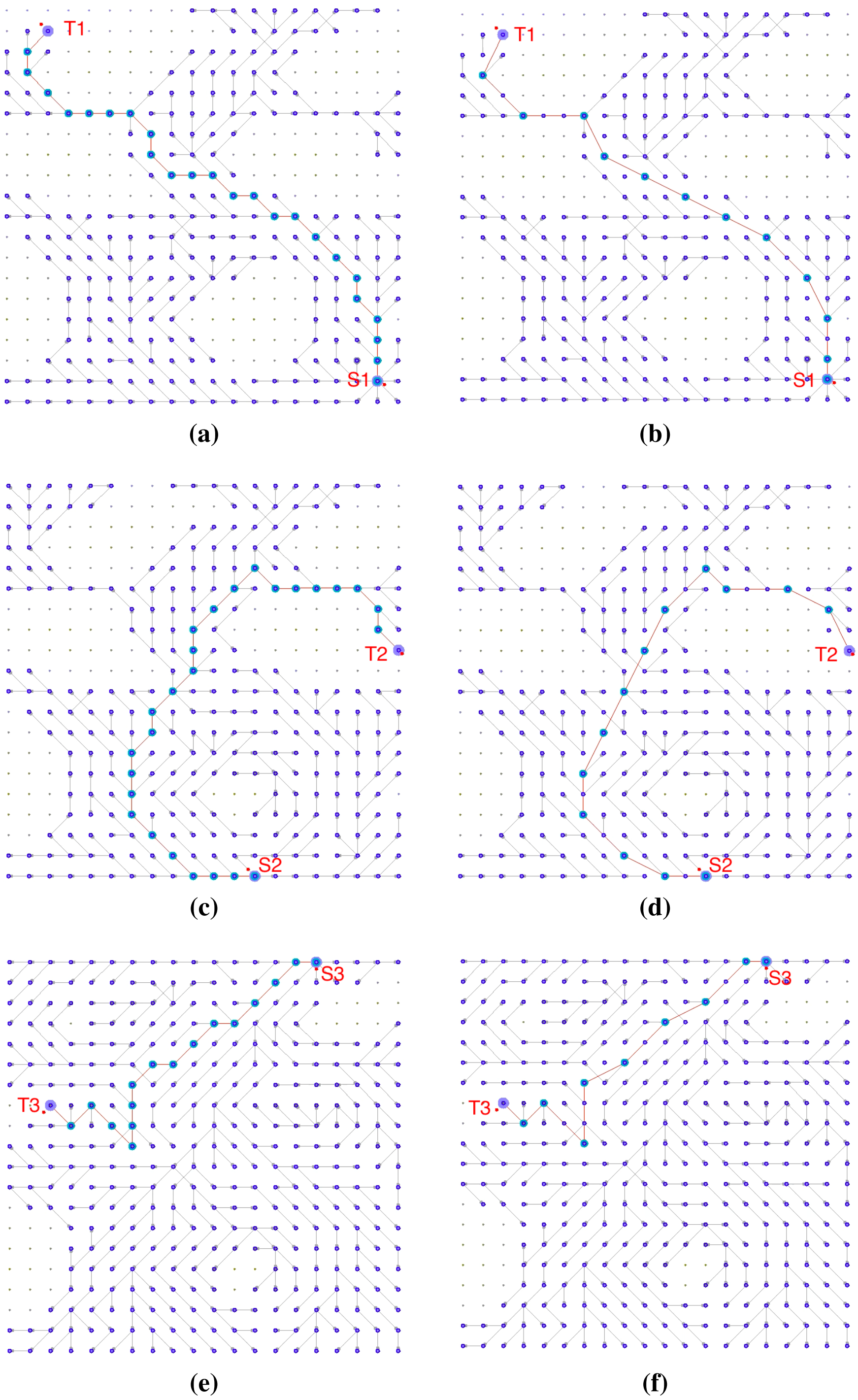
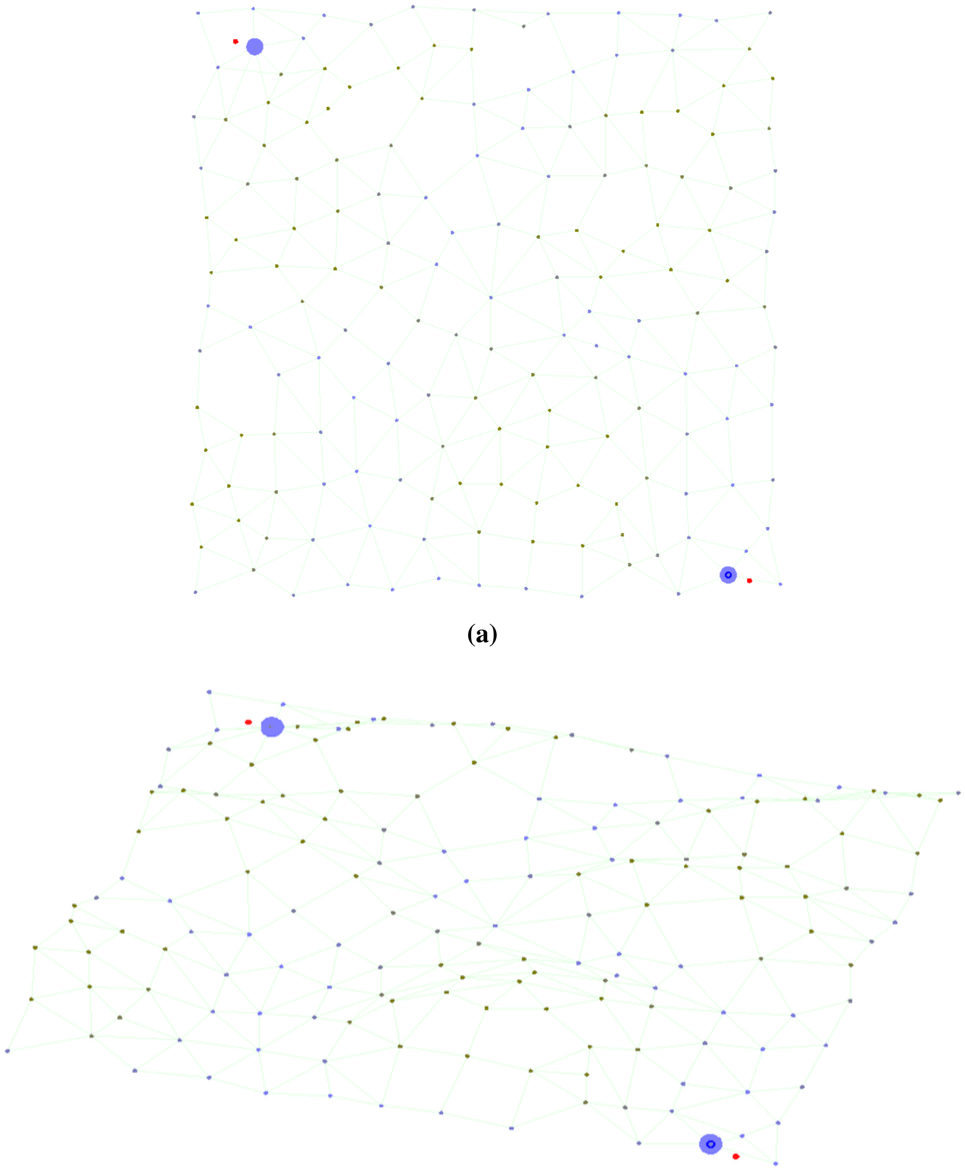


Figure 4.12: *Result of neuron transmission model in grid map model from top view (a) first experiment ($S1 = \{0.5, 11\}$ and $T1 = \{11.0, 1.0\}$) without SP-BT model (b) with SP model (c) second experiment ($S2 = \{0.2, 7\}$ and $T2 = \{6.5, 11.5\}$) without SP-BT model (d) with SP-BT model (e) third experiment ($S3 = \{11.2, 9\}$ and $T3 = \{7.0, 1.0\}$) without SP-BT model (f) with SP-BT model*



(a)

(b)

Figure 4.13: *Topological map model of experimental terrain (Fig. 4.11a) (a) top view (b) perspective view*

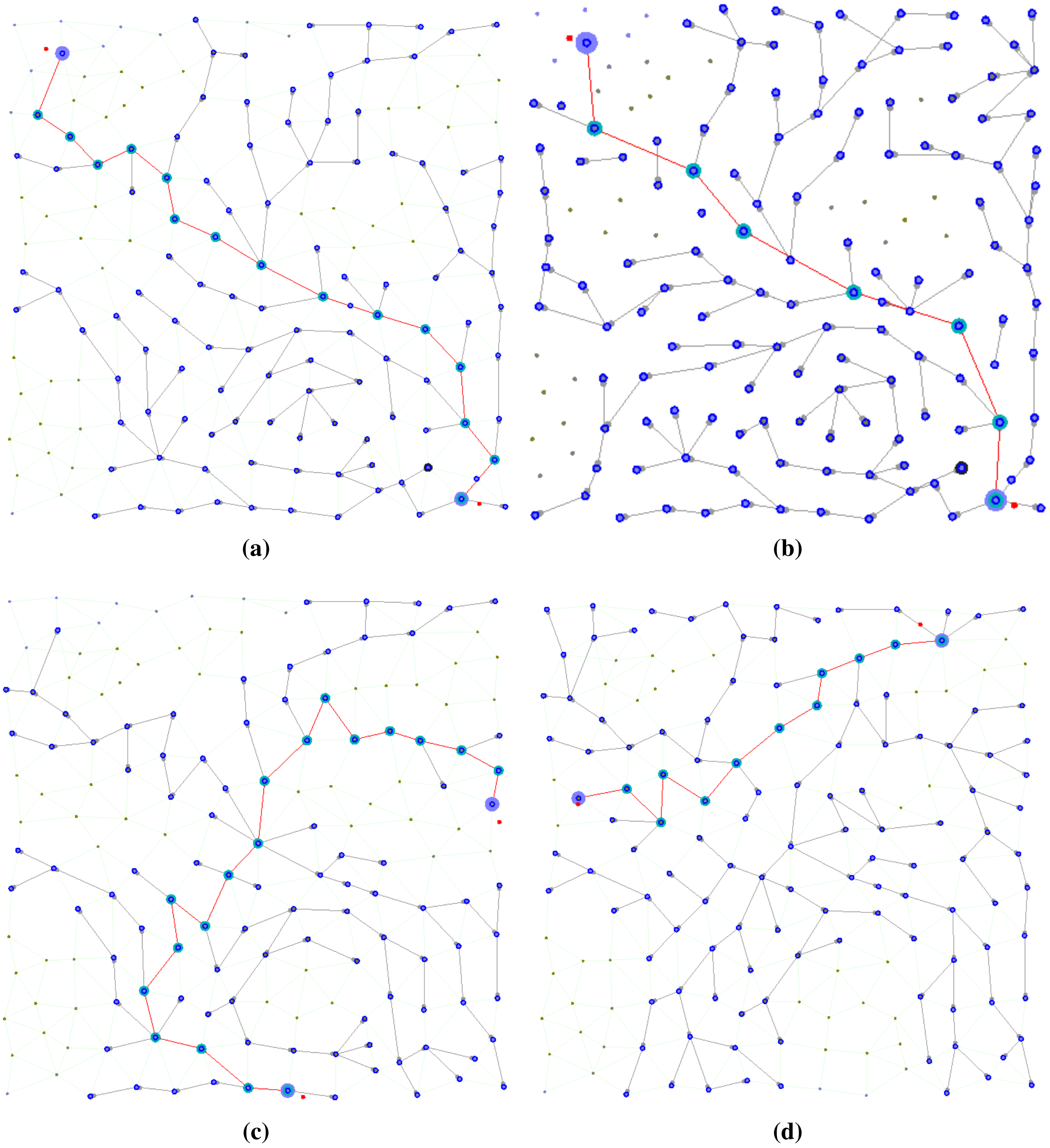


Figure 4.14: *Result of neuron transmission model in topological map model from top view (a) first experiment (b) first experiment with SP-BT model (c) second experiment (d) third experiment*

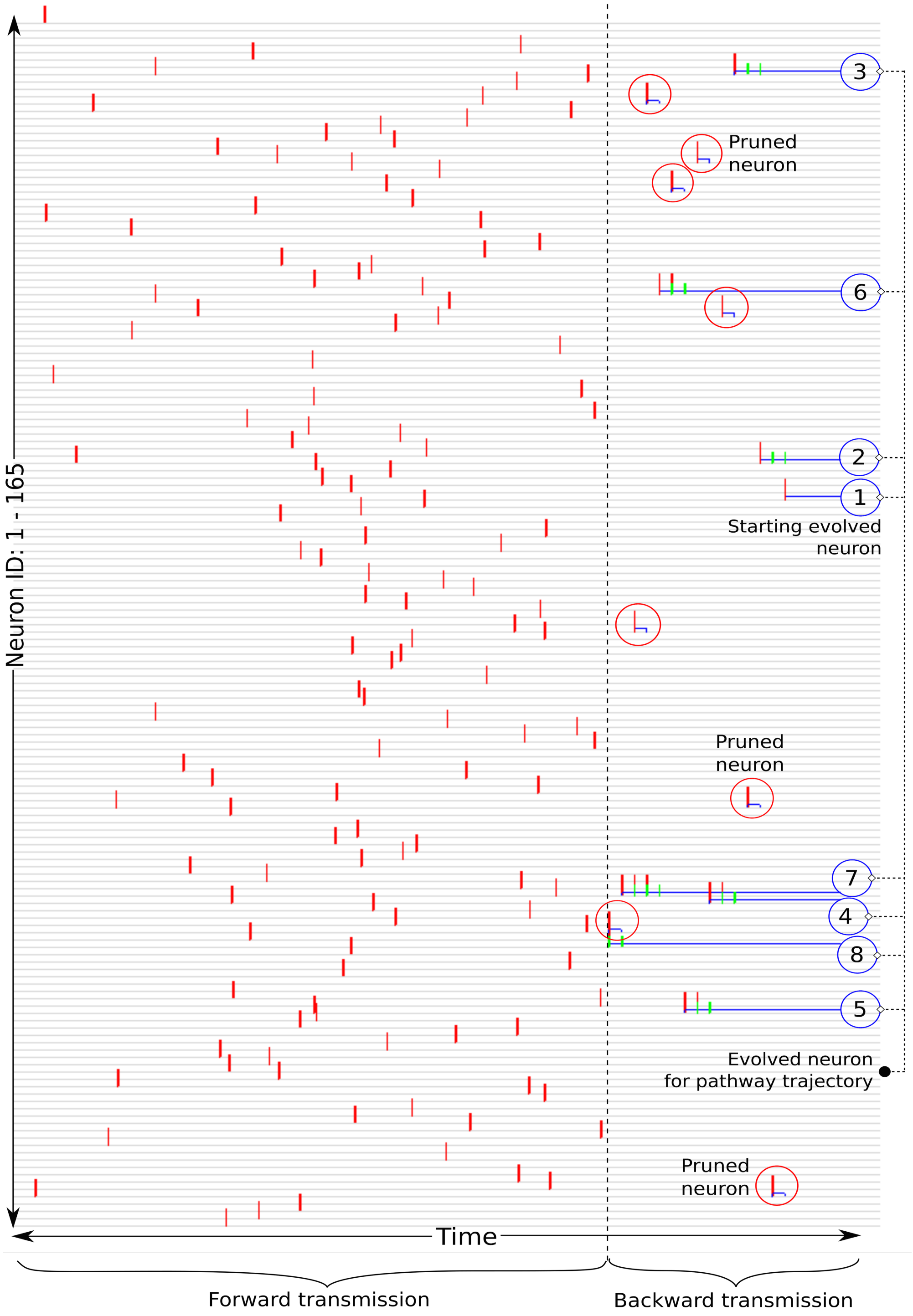


Figure 4.15: *Firing signal of neurons in FT and SP-BT processes, where the red circle represented pruned neuron and blue circle represented the neurons which constructed the pathway trajectory*

4.2.4.2 Performed with unpredictable collisions

We put 3 unpredictable collisions in different positions, where the first, second, and third obstacle are located in $p_1 = (4.0; 9.0)$, $p_2 = (2.0; 10.0)$, and $p_3 = (7.0; 5.0)$, respectively. The result depicted in Fig. 4.12 shows, that the path planning model can dynamically change the pathway when it finds unpredictable collision. In Fig. 4.16a, the proposed model generates the initial pathway based on the initial map reconstruction. When the robot moves to a certain position, the robot finds the obstacle and the path planning model changes the pathway depicted in Figs. 4.16b, 4.16c, and 4.16d.

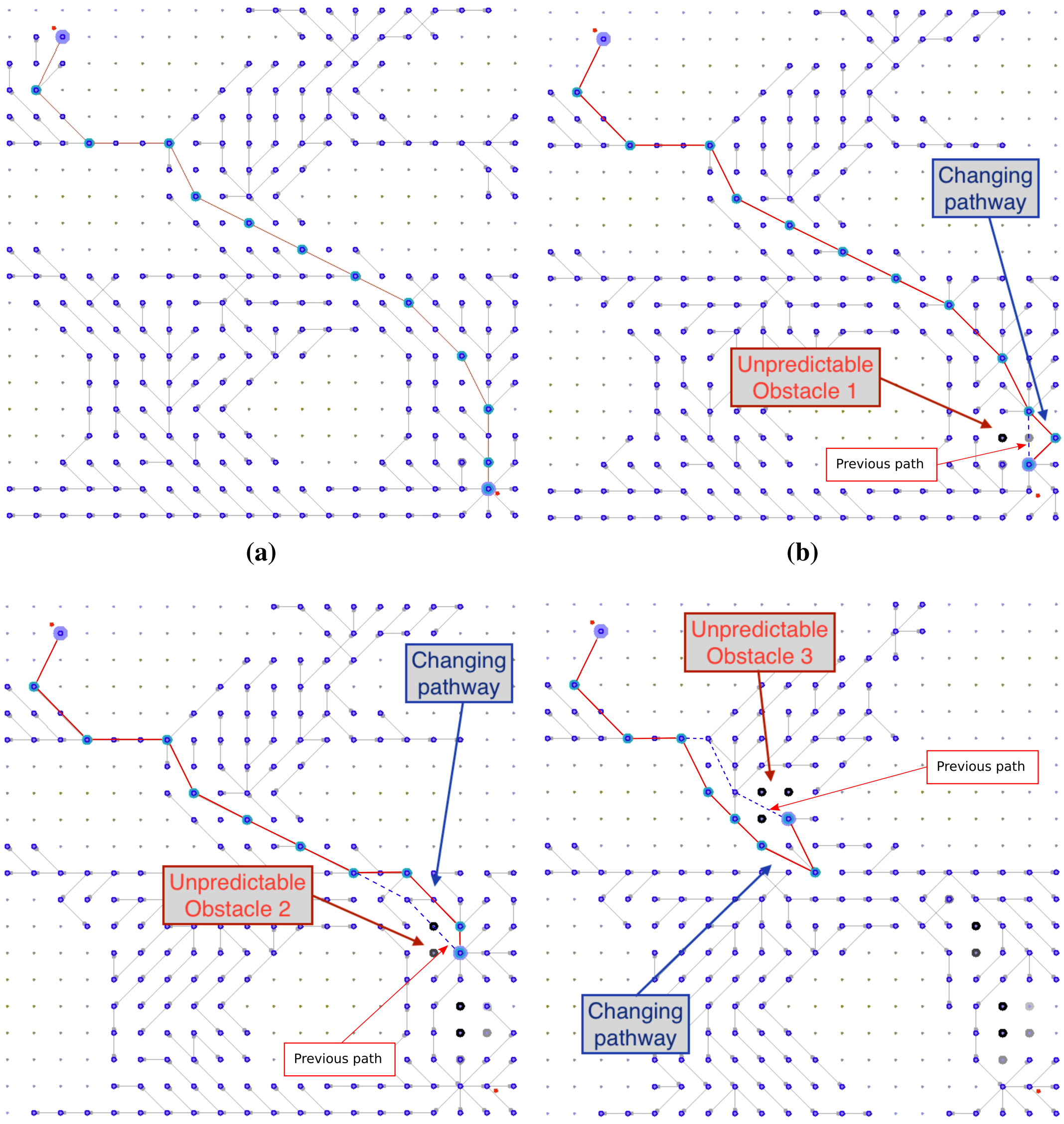
This experiment proves the flexibility of the proposed path planning model when finding unpredictable collisions.

4.2.5 System Integration

In this experiment we prove the effectiveness and the feasibility of the proposed model to be applied in the real cases. In order to be applied into the robot, it requires system integration for integrate all subsystem of motion capabilities. The system integration model can be seen in Fig. 4.8. There are 3 subsystem to be integrated, 1) motion planning model, 2) locomotion generator, and 3) stability model. From the topological map model, proposed neural based motion planning model generate path planning and send the desired walking provision to locomotion generator. After that locomotion generator generates signal in joint angle level. In order to recover and stabilize from disturabance, stability model sends feedback information to locomotion generator. These experiments are divided into 2 experiments, a simulation case and a real case.

4.2.6 Simulation case

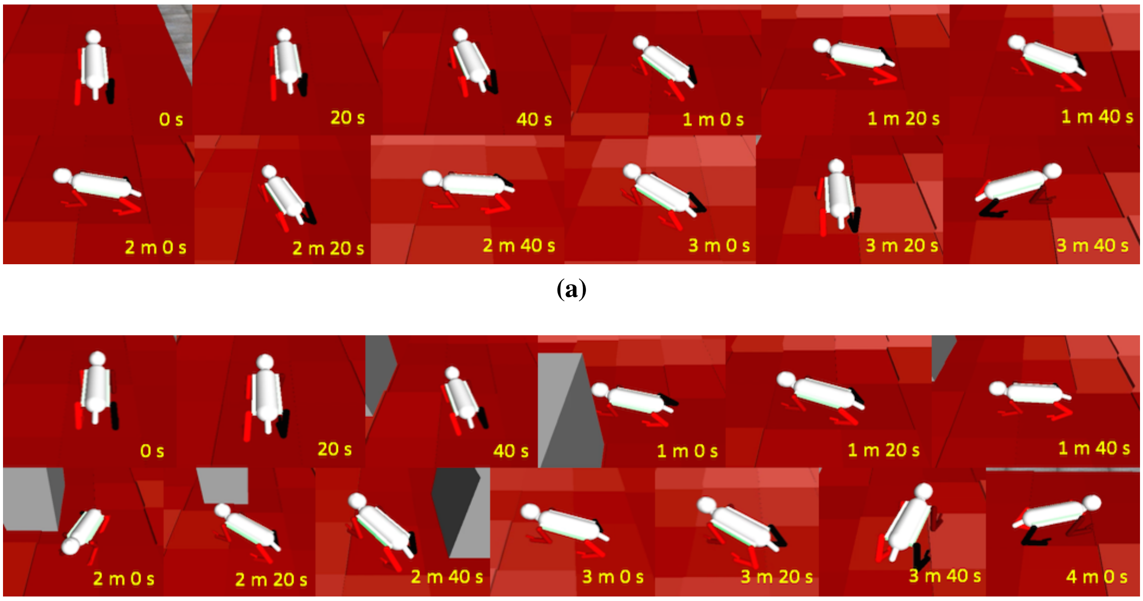
Open Dynamics Engine is used as the 3-D simulation software. We implemented the grid map model in order to adjust with simulation software ODE. The map model that is used for this experiment can be seen in Fig. 4.11c. In this simulation case, we perform the robot to move from starting point to target point based on the pathways generated in the Grid Map Model experiment. The result of the robot movement in each experiment can be seen in Fig. 4.18a and 4.18b. The result of the first experiment in the case without unpredictable obstacle shown from perspective view can be seen in Fig. 4.17. In the first experiment in the case with unpredictable obstacle, when the robot's sensor approaches and finds the obstacle, then the path planner dynamically changes the pathway shown in Fig. 4.18c. Figure 4.18c



(a) **(b)**

(c) **(d)**

Figure 4.16: *Result of neuron transmission model in grid map model (a) initial pathway (b) changing pathway when the robot found the first obstacle (c) second obstacle (d) third obstacle. The black point represents the obstacle position and the gray point represents the influence of the existence of the obstacle*



(a)

(b)

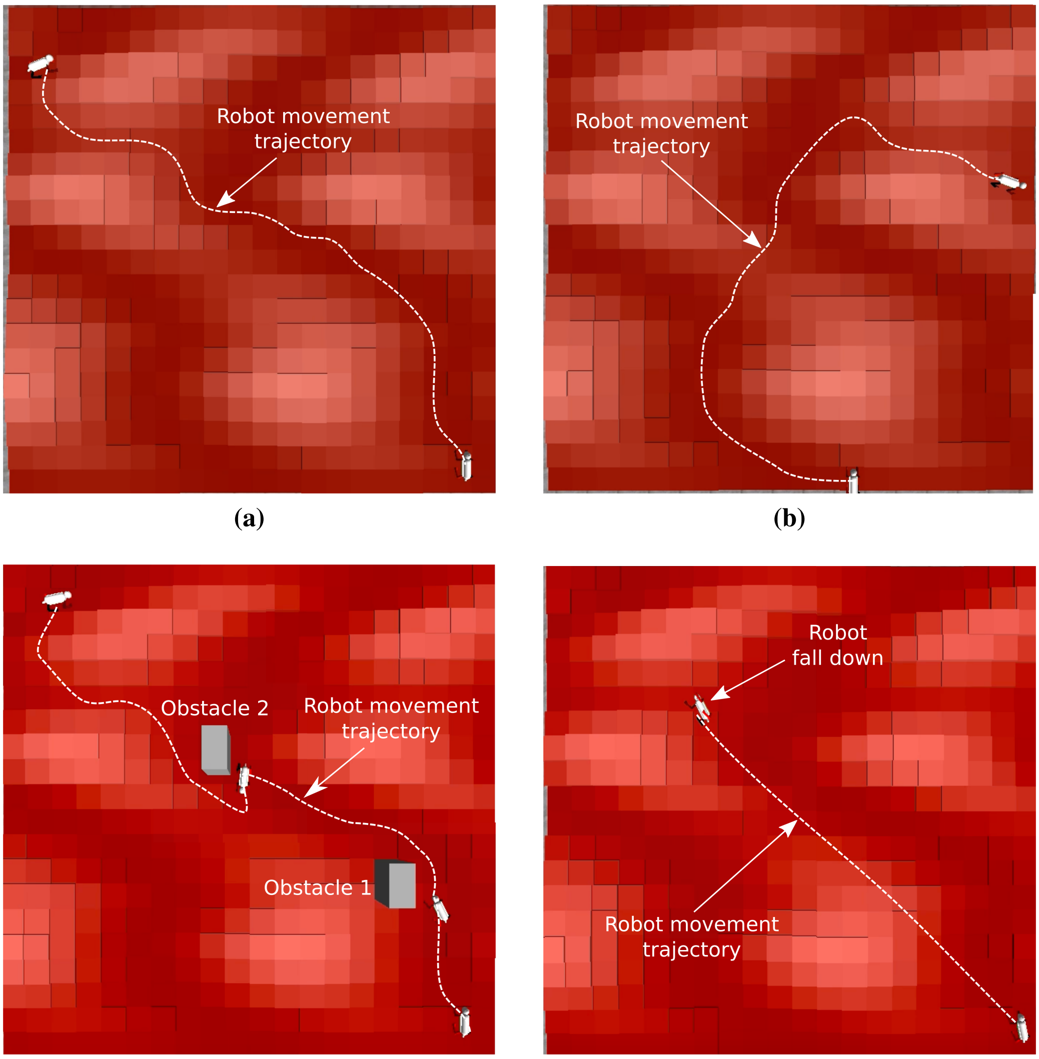
Figure 4.17: *Robot is performing the movement on rough terrain (first experiment) from perspective view (a) without unpredictable obstacle (b) with unpredictable obstacle*

shows the effectiveness of the proposed model with unpredictable obstacles, where the pathway changes from the pathway shown in Fig. 4.18a.

When the robot moves directly from the starting point to the target point without following the pathway generated by the proposed path planning model, the robot will be falling down as shown in Fig. 4.18d. By following the pathway generated by the proposed path planning model, the robot successfully moved from certain starting point to target point. Figure 4.18 shows the comparison between robot movement with the proposed model and without the proposed model. It also proves the effectiveness of the path planning model that can generate a safe path and improve the performance of the robot movements on uneven terrain. The robot was falling down because it was moving on a high slope terrain that is not covered by the robot's stability system. The proposed model chose the fastest way but considering the safe pathway and the performance of the robot itself.

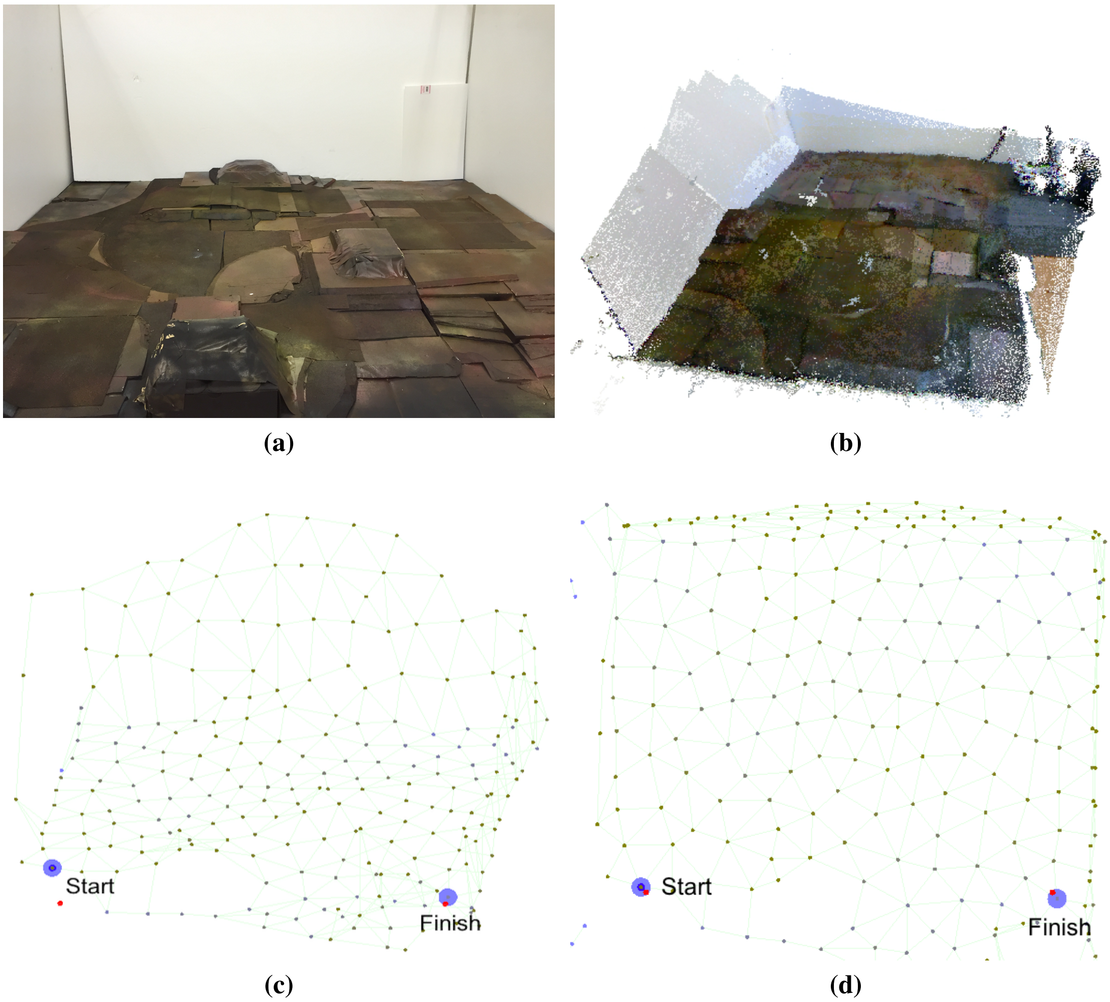
4.2.6.1 Real case

In the real case, we built the miniature of the map depicted in Fig. 5.54 and implemented the 4-legged robot explained in Section 7.3. We set low speed in the small 4-legged robot with maximum speed 5 cm/s. The size of map is 200 cm times 180 cm and it was reconstructed into topological map depicted in Fig. 4.19c and 4.19d by using 3-D reconstruction algorithm proposed in [288]. The dynamic environment is represented by the unpredictable obstacle.


Robot movement trajectory
Robot movement trajectory
Obstacle 2
Robot movement trajectory
Obstacle 1
Robot fall down
Robot movement trajectory

(a) (b) (c) (d)

Figure 4.18: *Result of the robot movement based on the generated pathways (a) pathway generated in the first experiment (b) second experiment (c) first experiment with unpredictable obstacle (gray color as the obstacle and the black one is a shadow of the obstacle) (d) without performing the generated pathway (fallen down)*


(a)
(b)

(c)

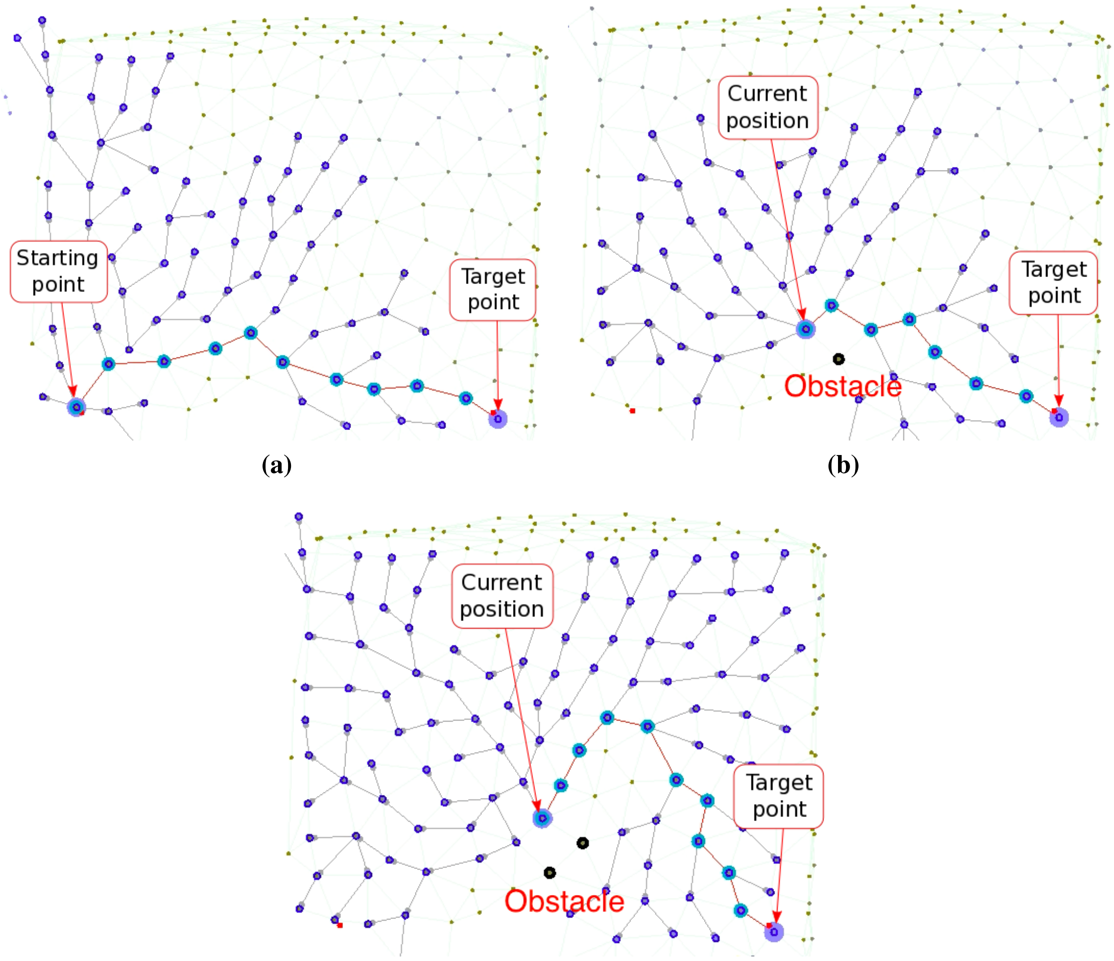
(d)

Figure 4.19: *Artificial map represents real cases (a) real terrain (b) feature extraction by using Kinect camera (c) reconstruction into topological map (d) topological map from top view*

In this experiments, we conducted 2 scenarios which are path planning without and with unpredictable obstacle.

In the first scenario, we chose the starting point at coordinate (15, 15) and target point at coordinate (180, 15) in centimeter. The flexibility with any unpredictable obstacle was also performed in this experiment. We used the same starting and target point, and put the obstacle during performing the movement.

The initial pathway (without obstacle) generated by the proposed model can be seen in Fig. 4.20a, when the robot detects the obstacle by using ultrasonic sensor, then the robot sends the information that there is an obstacle at certain position to the path planning system through Bluetooth communication. After that, the system changes to the other possible pathway. The changing pathway can be seen in Fig. 4.20b and 4.20c. By following the pathway generated by the proposed path planning model, the 4-legged robot successfully moved from

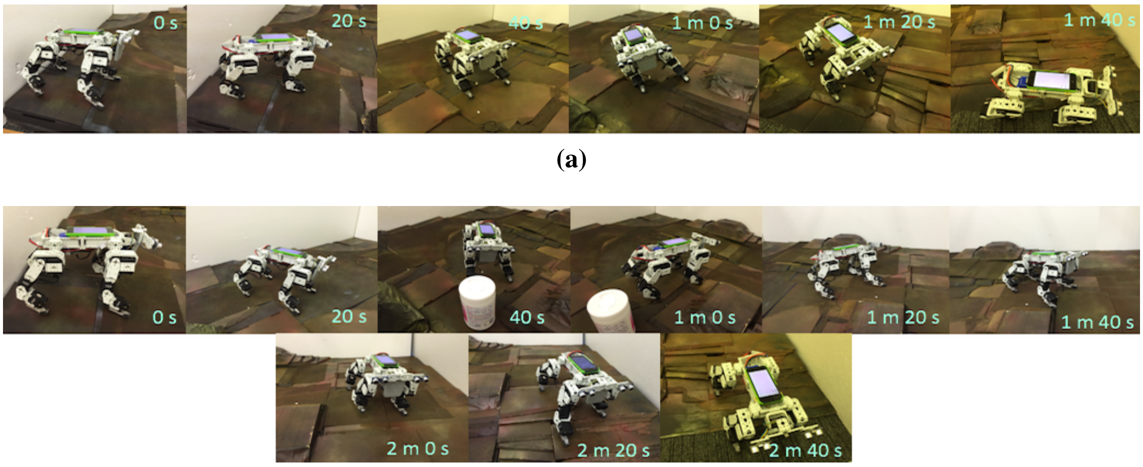


(a)

(b)

(c)

Figure 4.20: *Result of path planning model in topological map model (real terrain) (a) initial pathway (b) changing pathway when the robot found first obstacle (c) final pathway*



(a)

(b)

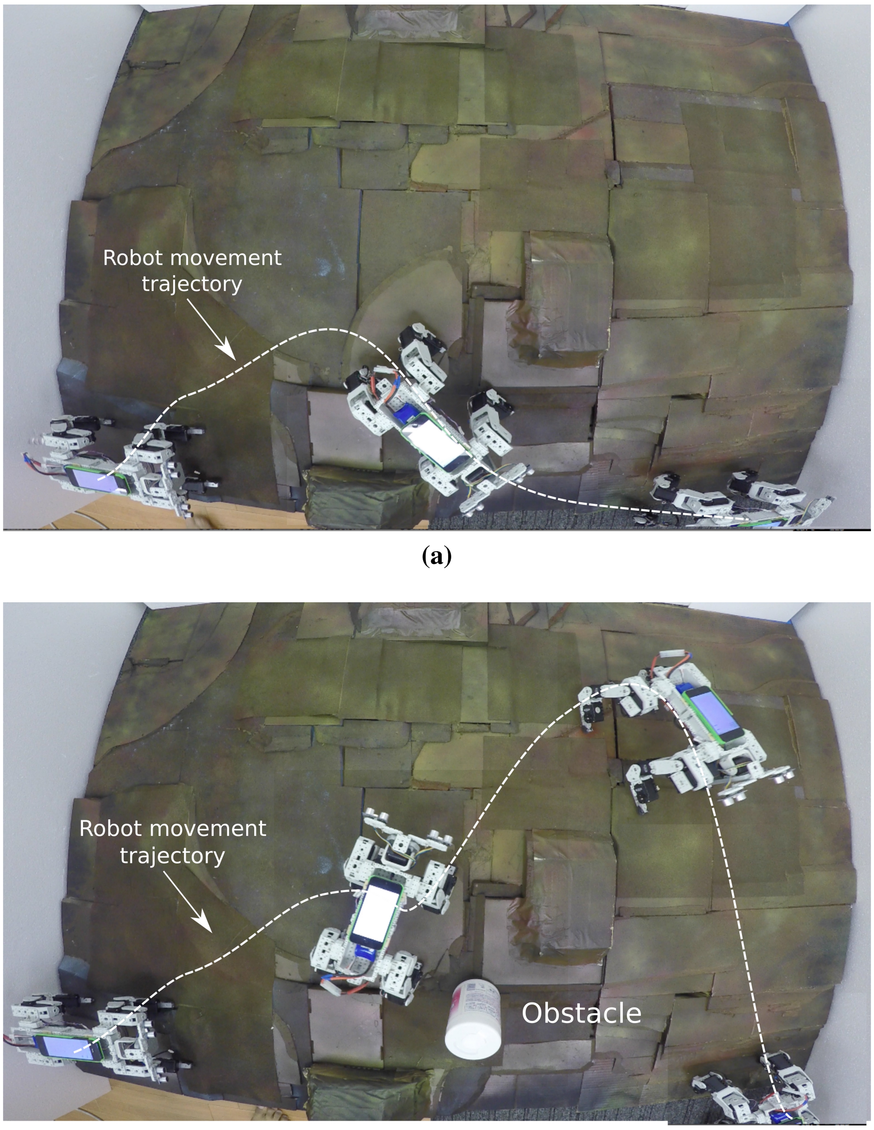
Figure 4.21: *4-legged robot is performing the movement on real rough terrain (a) without unpredictable obstacle (b) with unpredictable obstacle*

certain starting point to target point and also successfully avoided the obstacle. This path planning model also improves the performance of the robot movement on uneven terrain as depicted in Figs. 4.21 and 4.22. Figures 4.21 and 4.22 show the robot movement trajectory on uneven terrain from perspective view and top view in both without unpredictable obstacle and with unpredictable obstacle.

In the parameter effect discussion, we analyze the adjusted parameter with the result of the pathway. Spreading parameter (γ^{inf}) representing the strength of scope in forward transmission model will affect the number of connections. If the parameter is too small, only few pathway possibilities can be generated by forward transmission. If the parameter is too big, inappropriate connection can also be generated. Therefore, this parameter should be adjusted according to the resolution of the topological map. Constant variables (α^{l}) and (α^{d}) representing the degree of neuron distance effect and the deviation angle effect in synaptic pruning model are the parameters that affect the reducing of the represented neuron in the pathway. The value of parameters will influence the number of reduced neurons and the pathway changing. Therefore, the optimal value should be chosen based on the topological map resolution.

These implementations are still not enough for representing advance cases such as disaster area or mountains area. However, these implementations and the conducted experiments prove the capability of the proposed path planning model to be applied in further advance implementation.

In the performance aspect analysis based on the experimental result, the proposed model has advantages which are listed as follows:

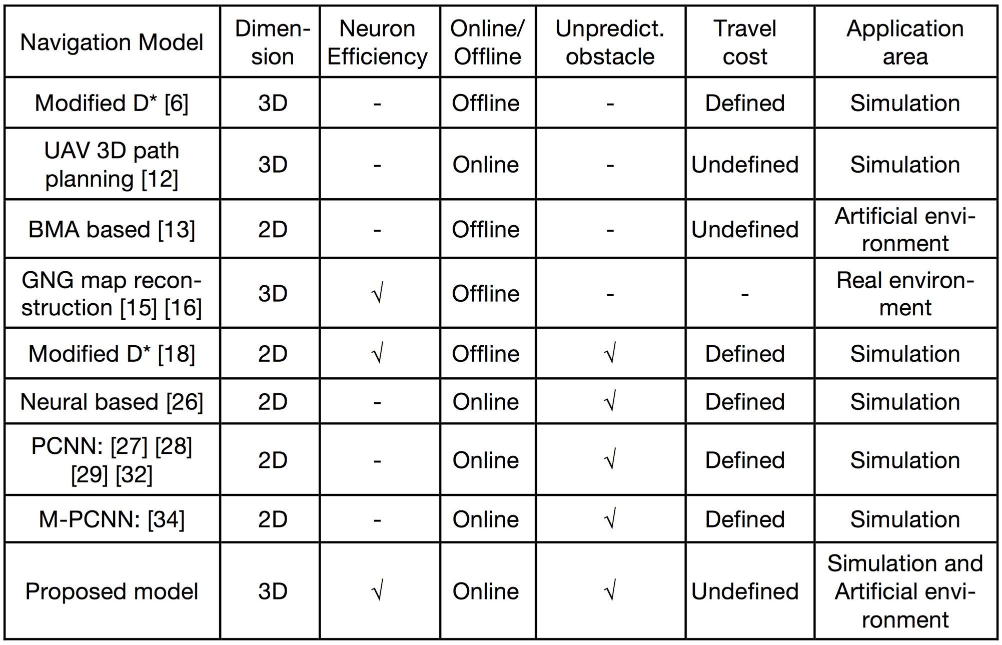


(a)

(b)

Figure 4.22: *Result of the 4-legged robot movement based on the generated pathways on real rough terrain (a) without unpredictable obstacle (b) with unpredictable obstacle*

Table 4.2: *Comparison of Path Planning Models and their Performances Aspects*



Navigation Model	Dimen-sion	Neuron Efficiency	Online/ Offline	Unpredict. obstacle	Travel cost	Application area
Modified D* [6]	3D	-	Offline	-	Defined	Simulation
UAV 3D path planning [12]	3D	-	Online	-	Undefined	Simulation
BMA based [13]	2D	-	Offline	-	Undefined	Artificial environment
GNG map reconstruction [15] [16]	3D	√	Offline	-	-	Real environment
Modified D* [18]	2D	√	Offline	√	Defined	Simulation
Neural based [26]	2D	-	Online	√	Defined	Simulation
PCNN: [27] [28] [29] [32]	2D	-	Online	√	Defined	Simulation
M-PCNN: [34]	2D	-	Online	√	Defined	Simulation
Proposed model	3D	√	Online	√	Undefined	Simulation and Artificial environment

- Implementing online path planning with unpredictable obstacle.
- Considering 3-D path planning with undefined travel cost.
- Considering the neuron efficiency for reducing the memory usage in communication.

In this performance analysis, the computational cost is difficult to be compared, because the application and the constraint are different such as in 2-D or 3-D, the existence of travel cost, and its application area. Nevertheless, there are 6 aspects tabulated in Table 4.2 that shows the superiority of the proposed path planning model.

4.2.7 Discussion on Neural-based Path Planning

This research proposed a natural mechanism of the human brain in order to generate a dynamic path planning in rough terrain with undefined travel cost. There are 2 algorithm processes in this proposed model, forward transmission and synaptic pruning with backward transmission. First, FT generates the impulse signal to neighbor neurons, defines the travel cost, and creates the neuron connection. Second, after target neuron is connected and gets the impulse, then the synaptic pruning and backward transmission are performed. These algorithms are implemented in grid map model and topological map model. One point is represented by a neuron in this proposed model.

Based on the experiments in both grid map model and topological map model, FT model successfully constructed the neuron connections for finding the possible way. SP with BT also successfully found the possible pathway from current position to target position and reduced inefficient neuron in both map models, however the considered condition in SP model is required to be increased. These possible pathways are also proved by performing the robot in both computer simulation and real robot. The robot successfully moved from starting position and target position when following the pathway generated by the proposed path planning. This proposed path planning model supports the performance of the robot movement. When the robot moved directly from the starting point to the target point without following the pathway generated by the proposed path planning model, the robot will fall down.

The flexibility of this dynamic path planning was proved by the experimental result in computer simulation and real robot. When the robot follows the pathway and finds the obstacles, then the path planning model is changed to another possible pathway which is appropriate with robot ability.

In the future application, the proposed path planning will be applied for advance cases such as in disaster area, hills or mountains area. In order to improve the flexibility of the path planning, the supporting sensor for this proposed model will be improved in the robot application.

Chapter 5

Mesoscopic to multi-scopic Adaptation

In this chapter, I provides the concept model in mesoscopic level, which manages and processes data from microscopic to macroscopic and vice versa. The model can be seen in Fig. 5.1. The important of this system is to integrate smoothly and strongly between lower level planning (MiSc) and higher level planning (MaSc).

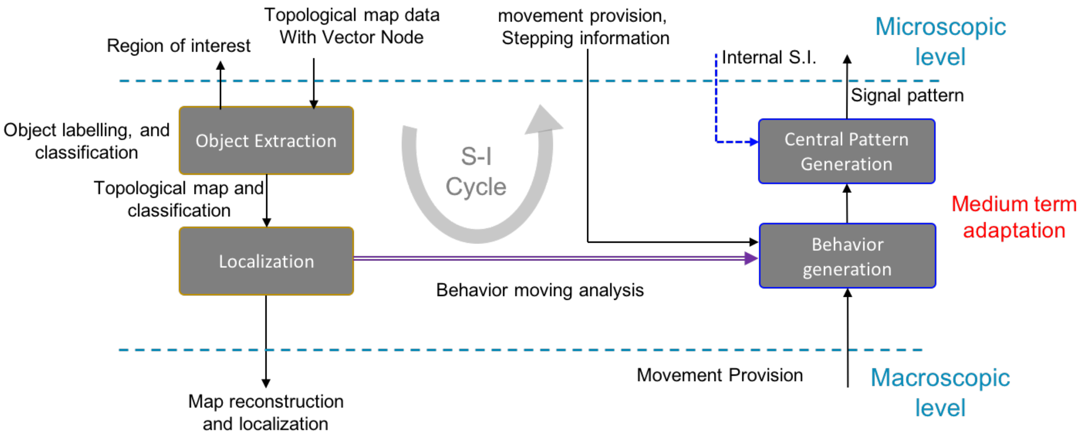

Region of interest
Topological map data With Vector Node
movement provision, Stepping information
Internal S.I.
Microscopic level
Signal pattern
Object labelling, and classification
Object Extraction
S-I Cycle
Central Pattern Generation
Topological map and classification
Medium term adaptation
Localization
Behavior generation
Behavior moving analysis
Movement Provision
Macroscopic level
Map reconstruction and localization

Figure 5.1: *Diagram of mesoscopic level of neuro-cognitive model*

I build the localization model by using topological map generated by DD-GNG in MiSc. I present continuous and real-time the cognitive map building by using the topological structure information from lower level, composed as 3D vector position of nodes, edges, and 3D surface vectors of nodes. The model also classifies obstacle, wall, flat terrain or rough terrain, and certain object such as rungs of the ladder. Thus, this information will be transferred to the MaSc level. Besides that, the motion planning generated from the MaSc level will be processed for the neuro-locomotion in the MiSc level by using behavior generation and its localization. The learning process to integrate the relationship be-tween behavior command and locomotion performance is developed as well. The proposed model has been tested for omnidirectional movement in biped and quadruped robot. It has also been implemented for robot climbing behavior, performing a horizontal-vertical-horizontal movement.

5.1 Localization model from Topological Structure

To support the cognitive map model, mesoscopic level should provides some localization information such as SLAM. The localization should be continuously generated. The localization algorithm there are integration of many sensor such as laser range finder, rotary encoder, IMU sensor, GPS and also camera [15, 48]. Nowdays, SLAM using 3D point cloud information provided by LiDAR or depth sensor is the one of the favorite model [51]. It also can be use for underwater localization [209]. There are many method for localization and map building using 3D point cloud. The Iterative Closest Point (ICP) algorithm is an efficient model for registering the point cloud from different perspective [25]. The model has been successfully combined with a heuristic for closed loop detection and a global relaxation method for 6D SLAM [202]. Ohno et al, has a similar model to build real-time 3D map reconstruction and trajectory estimation [203]. However, the current state of 3D localization and map building are required high computational cost and quite sensitive with noise of 3D point cloud data, especially for continuous localization.

To dealt with the memory consumption, OctoMap presents probabilistic occupancy estimation for generating volumetric 3D environmental model [114]. This model is difficult to build for high resolution map [328]. The size of map memory should be defined in prior. Vespa et al improve the occupancy mapping and the accuracy of the map by integrating with TSDF mapping [299]. However the volumetric strategy based map representation has useless voxels in flat area of non rough area. Moreover, the dynamism of spreadness of data representation is limited. It cause the increasing of computational cost for large area.

Therefore in this section, I proposed a real time and continuous map building algorithm using topological structure as an input. Bloesch et al use triangular meshes as both compact and dense geometry representation. They propose a view-based formulation for which I predict the in-plane vertex coordinates directly from images and then employ the remaining vertex depth components as free variables, to simplify and incrase the computational speed [28].

The problem in the topological input is in the representation of small object of small texture. Our model is supported by proposed attention control mechanism powered by DD-GNG. It generated dynamic density topological node that can control the number of node representation depending on the texture of the detected area.

In other hand, building a cognitive map requires an integration with robot embodiment.

Different embodiment of the robot may have difference cognitive map in coverage motion viewpoint. Current SLAM model has limited on such kind of information. It provides a map reconstruction and localization without considering the capability of the robot.

The topological structure is composed as 3D vector position of nodes, edges, and 3D surface vectors of nodes generated from growing neural gas. I use only 3D point cloud data generated from time of flight sensors.

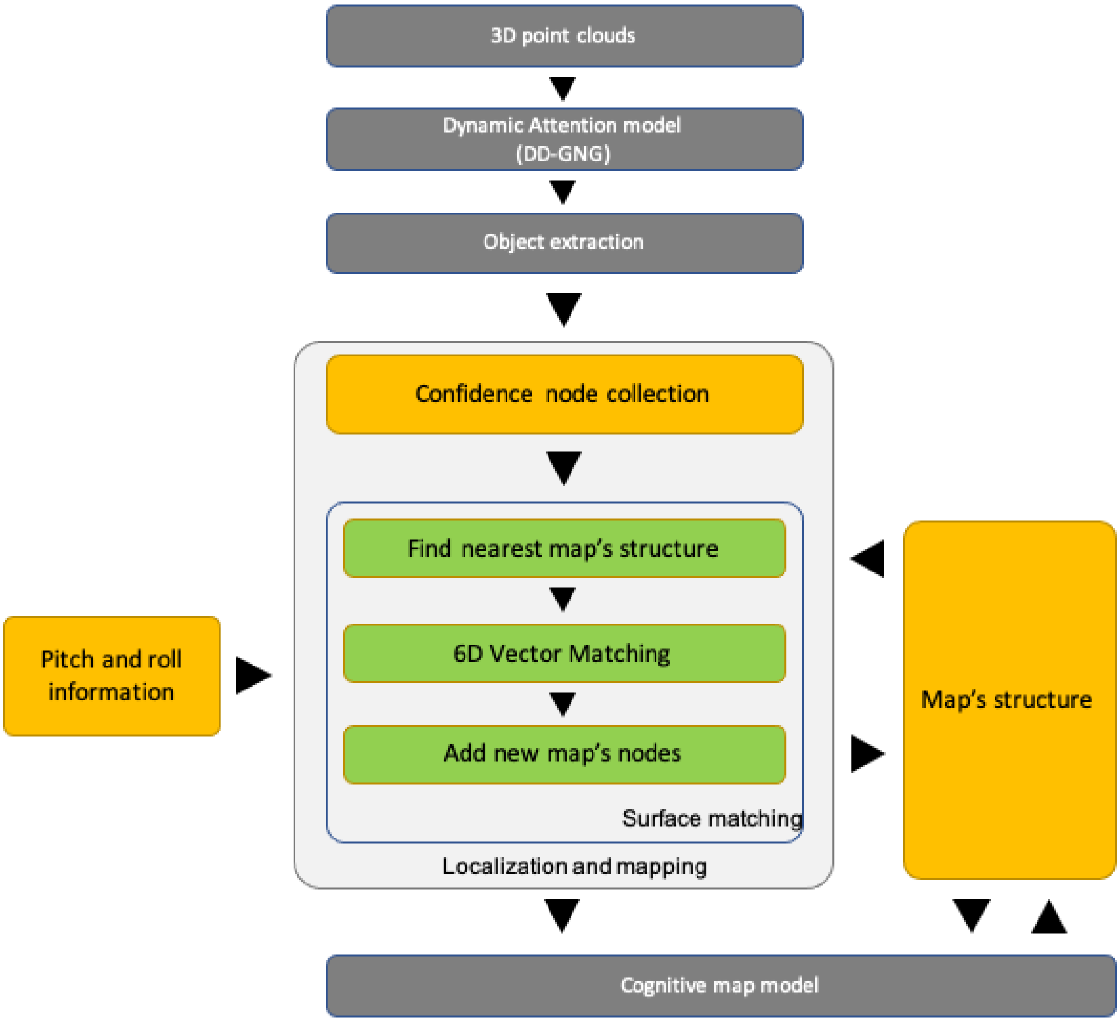


Figure 5.2: *Diagram of Localization and Mapping using topological information*

The localization model can be seen in Fig. 5.2. The topological structure with label information is the input of the system. There are two step for building the localization and mapping, which is Confidence Node collection and Surface matching. Assumed, $\mathbb{Q}$ as the nodes of map composed as (3D posisition ($\mathbb{Q}_n$), 3D surface vector ($\mathbb{Q}_v$), label ($\mathbb{Q}_l$), and magnitude of surface vector ($\mathbb{Q}_m$)), $N_{\mathbb{Q}}$ is the number of map's nodes, $\mathbb{A}$ as the nodes of GNG composed as (3D posisition ($\mathbb{A}_n$), 3D surface vector ($\mathbb{A}_v$), label ($\mathbb{A}_l$), and magnitude of surface vector ($\mathbb{A}_m$)), $N_{\mathbb{A}}$ is the number of GNG's nodes, $\mathbf{s}$ is the global position of the robot, and ` is the euler angle from the earth.

5.1.1 Confidence Node Selection

In this part, I select the appropriate node's which has stable normal vector calculated in Dynamic Attention Model in Section 3.1. In the initial condition, the selected node will be added directly to the map's structure. The step of node selection can be seen as follows:

1. $N_d \leftarrow 0$, N_d is the number of confidence nodes.
2. $d =$ the vector of assigned node id, $i = 0$.
3. $d_{N_d} \leftarrow i, N_d \leftarrow N_d + 1$ if ith node has stable normal vector
4. $i++$, return to step 2 if $i \leq N_{\mathbb{A}}$

5.1.2 Surface Matching

After selecting some appropriate nodes, the module will match the normal vector and position with the map's structure. There are three steps, 1) Find nearest map's structure, 2) 6D vector matching, and 3) Add new map's nodes.

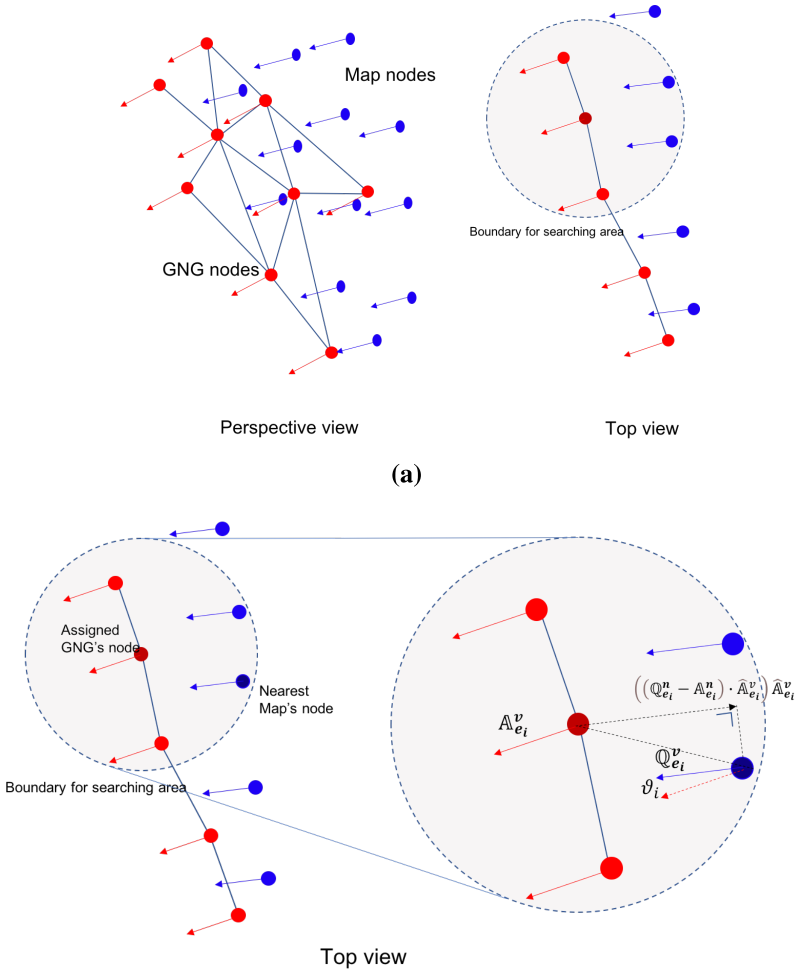
5.1.2.1 Find nearest map's structure

The module will search the nearest map's node and similar normal vector with the assigned node. If the module can not find in the defined range, then the node will be not processed for the next step. Only node that has assigned to map's node will be selected to be processed in the next step. The Illustration of the assigned node can be seen in Fig. 5.3b. For each selected GNG nodes, I calculate the different of the position and vector of surface with surrounding map's node in Eqs. (5.1) and (5.2).

$$l_{d_i,j} = \|\mathbb{Q}_j^n - \mathbb{A}_{d_i}^n\| \tag{5.1}$$

$$v_{d_i,j} = \hat{\mathbb{A}}_{d_i}^v + \hat{\mathbb{A}}_{d_i}^v \tag{5.2}$$

where, $l_{d_i,j}$ and $v_{d_i,j}$ are the distance and degree of vector similarity between d_ith GNG node and jth Map's node, respectively. The module will choose the nearest distance of node in limited area with high similarity value of surface vector.


Map nodes
GNG nodes
Perspective view
Boundary for searching area
Top view
Assigned GNG's node
Nearest Map's node
Boundary for searching area
$\mathbb{A}_{e_i}^{v}$
$\left(\left(\mathbb{Q}_{e_i}^{n} - \mathbb{A}_{e_i}^{n}\right) \cdot \widehat{\mathbb{A}}_{e_i}^{v}\right) \widehat{\mathbb{A}}_{e_i}^{v}$
$\mathbb{Q}_{e_i}^{v}$
ϑ_i
Top view

(a)

(b)

Figure 5.3: *(a) Robot design from front side (b) Robot design from side.*

5.1.2.2 6D Vector Matching

In this part, I fit the assigned nodes to the map's node. To fit the position of the nodes, I calculate the different of the 3D position in Eq. 5.3, where $\dot{s}(t)$ the difference of position

$$\dot{s}(t) = \frac{1}{N_d} \sum_{i=0}^{N_d} ((\mathbb{Q}^n_{e_i} - \mathbb{A}^n_{d_i})\dot{\hat{}}\mathbb{A}^v_{d_i})\hat{\mathbb{A}}^v_{d_i} \tag{5.3}$$

$$\mathbf{s}(t+1) = \mathbf{s}(t) + \dot{\mathbf{s}}(t) \tag{5.4}$$

$$\dot{\grave{}}(t) = \frac{1}{N_d} \sum_{i=0}^{N_d} \Theta \tag{5.5}$$

$$\grave{}(t+1) = \grave{}(t) + \dot{\grave{}}(t) \tag{5.6}$$

5.1.2.3 Add new map's node

In this part we merge the associated GNG node and maps nodes. The associated pair of gng node and map node is compared. Next, the associated gng node which doesnt have pair with maps node, will be added as new maps nodes.

5.1.3 Performance's test

.

In order to show the effectiveness of the proposed localization and mapping model, I conducted several test in simulation and real experiments. I use the topological structure generated by dynamic attention model explained in Section 3.1 as the input of the model.

5.1.3.1 Simulation

In computer simulation, we created artificial room and corridors with some obstacles. We created buggy-type robot which is attached by artificial depth camera in simulation Open Dynamics Engines.

5.1.3.2 Real implementation

In the real implementation, I tested the proposed model for localization and mapping in indoor and outdoor. In the indoor implementation, we use

5.2 Behavior coordination

The aim of this system is to generate the behaviors for representing omni-directional movement. In this section, walking behavior and optimization strategy are explained. In order to build omni-directional movement, at least there are 3 parameters, sagittal movement, coronal movement, and turning movement [276, 273]. Therefore, in the behavior building processes, behavior of 3 walking parameters are optimized in different value of each parameters.

5.2.1 Behavior Building

Since the neuro based locomotion are implemented, behavior building implies to build the neuron interconnection. Different structure will generate different walking behavior. In this part, I will explain the strategy for optimizing the interconnection structure depending on the desired walking behavior. In this behavior forming, the optimization processes are separated by 3 parts, forming sagittal direction behaviors walking in different speed, forming coronal direction behaviors in different speed, and turning behaviors in different degree of turning level.

5.2.1.1 Sagittal direction behaviors optimization

Optimization objective is improved from single objective to multi-objective optimization and also the encoding process. I expect to build the walking behavior with stable pelvis speed with desired length of step and height of step. This model will be optimized using the proposed learning model explained in Section 5.3.

In the fitness model, I minimize the error of pelvis speed, error of step length, and error of step height. Error of pelvis speed $E_v^{(a)}$ is calculated as $E_v^{(a)} = \sum_{t=1}^{T}(d_x^{(v)} - v_x(t))^2$, where t and T are the time unit and maximum time, $d_x^{(v)}$ is the desired speed, and $v(t)$ is the current speed in sagittal direction. Error of step length $(E_l^{(a)})$ is calculated as $E_l^{(a)} = \sum_{s=1}^{N_s}(d^{(l)} - l(s))^2$, where, parameter s and N_s are the current step and maximum step resulted until maximum time (T), $d^{(l)}$ is the desired length of step, and $l(t)$ is the length of step in sth step. Error of step height $(E_h^{(a)})$ is calculated as $E_h^{(a)} = \sum_{s=1}^{N_s}(d^{(h)} - h(t))^2$, where, parameter $d^{(h)}$ is the desired height of step and $h(s)$ is the height of step in sth step. Considered parameters are grouped into 2 fitness functions are calculated as minimization shown in Eqs. (5.7) and (5.8).

$$g_{1-28} = \arg\min_{w_{i,j}}(E_v^{(a)} + E_l^{(a)}) \tag{5.7}$$

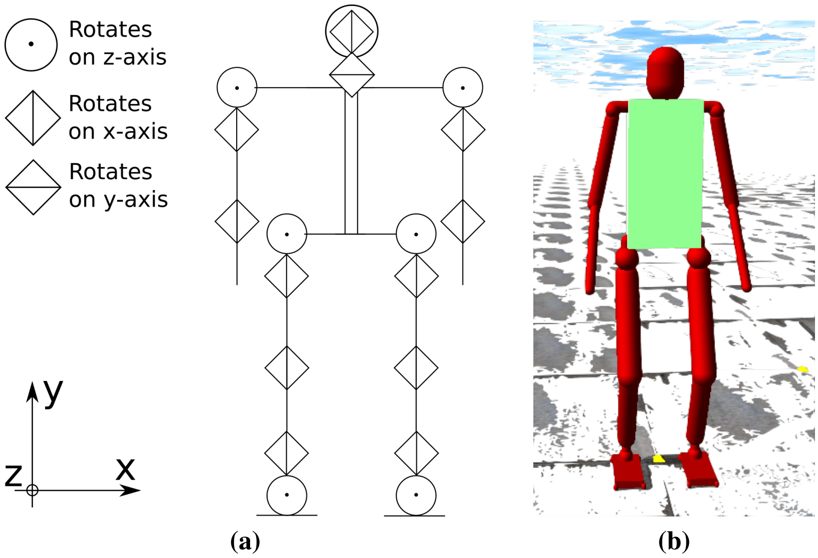


Figure 5.4: *(a) Robot design from front side (b) Robot design from side.*

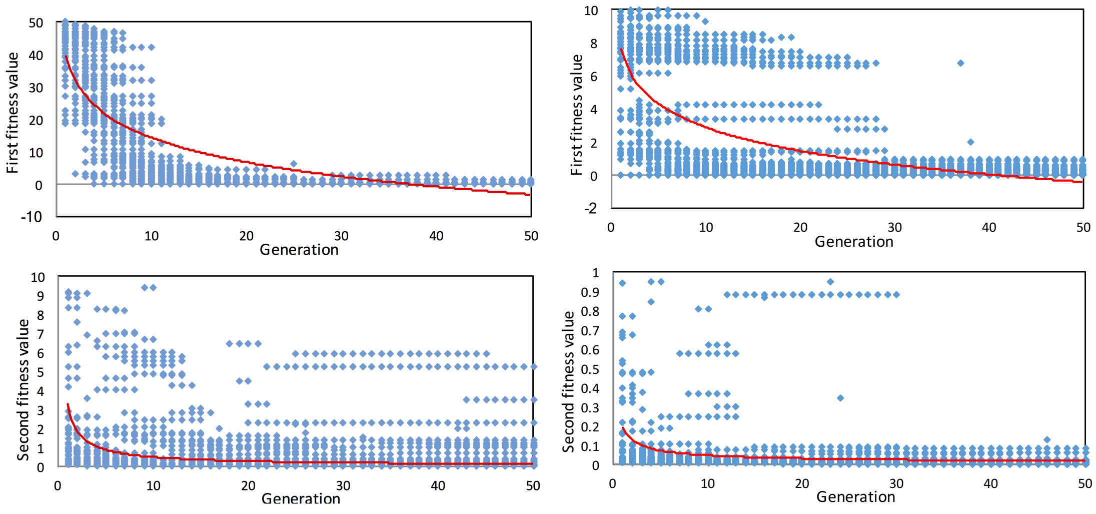
$$g_{1-28} = \underset{w_{i,j}}{\arg\min} E_h^{(a)} \tag{5.8}$$

This optimization will be conducted $P+1$ times with different walking speed, where $d_0^{(v)}$ is minimum desired speed v_min and $d_{P+1}^{(v)}$ is maximum desired speed v_{max} where the speed iteration is $(v_{max} - v_{min})/P$.

5.2.1.2 Coronal direction behaviors optimization

In this interconnection optimization, I expect to build the walking behavior with desired step length in coronal direction. In the evaluation model, I minimize the error of pelvis speed in coronal direction, error in length of step in coronal direction, and error of center of mass. The error of pelvis speed in coronal direction $E_{v_y}^{(b)}$ is calculated as $E_{v_y}^{(b)} = \sum_{t=1}^{T}(d_y^{(v)} - v_y(t))^2$, where, t and T are the time unit and maximum time, $d_y^{(v)}$ is the desired speed, and $v_y(t)$ is the current speed in coronal direction. The error in length of step in coronal direction $E_{l_y}^{(b)}$ is calculated as $E_{l_y}^{(b)} = \sum_{s=1}^{N_s}(d^{(l)_y} - l_y(s))^2$, where, $d^{(l)_y}$ is the desired length of step, and $l_y(s)$ is the length of step in sth step in coronal direction. and error of center of mass $E_{CoM}^{(b)}$ is calculated as $E_{CoM}^{(b)} = \sum_{t=1}^{T}(y_{CoM}(t) - y_{pelvis}(t))^2$ where, parameter $y_{CoM}(t)$ is the center of mass and y_{pelvis} is the pelvis position in coronal direction. Considered parameters are grouped into 2 fitness functions are calculated as minimization shown in Eqs. (5.9) and (5.10).

$$g_{29-51} = \underset{w_{i,j}}{\arg\min}(E_{v_y}^{(b)} + E_{l_y}^{(b)}) \tag{5.9}$$



(a) **(b)**

Figure 5.5: *Sample of fitness evolution of population in every generation (a) with 5 m/s of desire walking speed (b) with 7 m/s of desire walking speed*

.

$$g_{29-51} = \arg\min_{w_{i,j}} E_{CoM}^{(b)} \tag{5.10}$$

5.2.1.3 Turning direction behaviors optimization

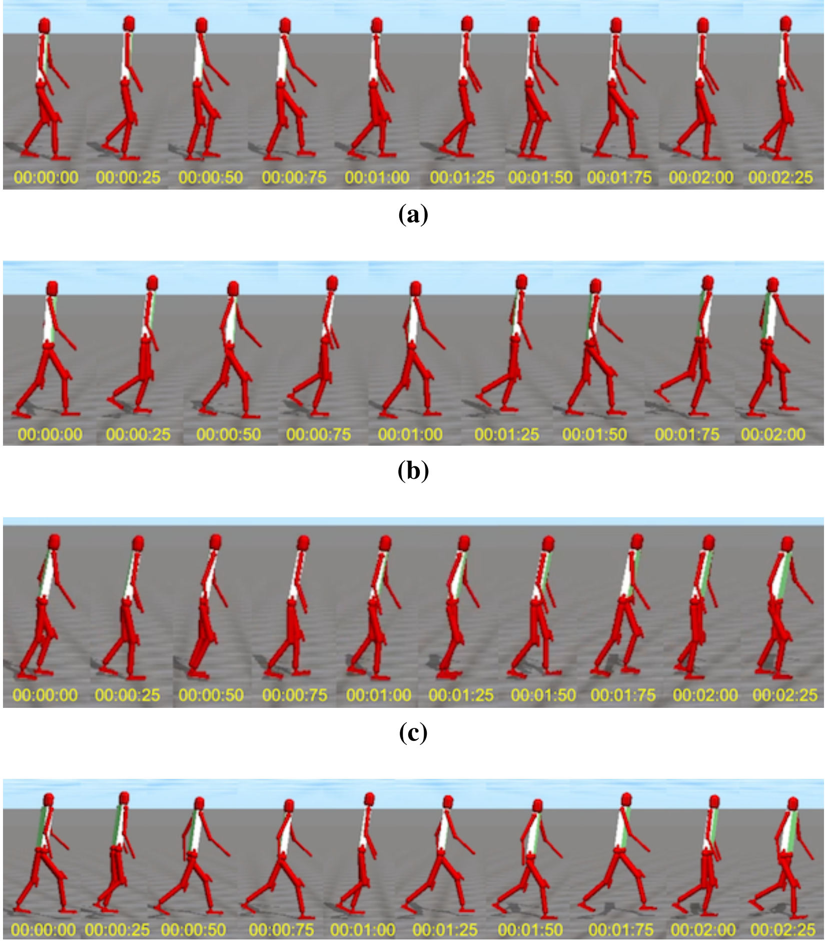
Neuron representing the turning behavior will be optimized in each speed level. In the evaluation process, I minimize the error of the walking direction $E_t^{(c)}$ which is calculated in Eqs. (5.11), (5.12). In Eq. (5.11), d_α is the desired turning angle and $\alpha = \theta_(l) - \theta_(r)$, where, $\theta_(l)$ and $\theta_(r)$ are the angle value of hip-z joint in left and right leg, respectively. Second evaluation, I consider to minimize turning speed which is calculated in Eq. (5.12).

$$E_t^{(c)} = \begin{cases} \alpha - d_\alpha & \text{if } s\%2 = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

$$E_a^{(c)} = \sum_{t=1}^{T} (\alpha(t) - \alpha(t-1))^2 \tag{5.12}$$

5.2.2 Behavior Learning Strategy

The aim of this learning model is for learning the walking behavior of biped robot in unscaled omni-directional movement and avoiding the complexity of neural interconnection. Omni-directional locomotion based on neural oscillator in this proposed section is separated by 2 interconnections depicted in Fig. 5.7, where main interconnection structure represents



(a)

(b)

(c)

(d)

Figure 5.6: *Generated sagittal movement behavior in different speed (a) normal walking behavior in low speed (b) fast walking behavior in medium speed (c) jogging behavior in medium speed (d) running behavior in high speed movement*

.

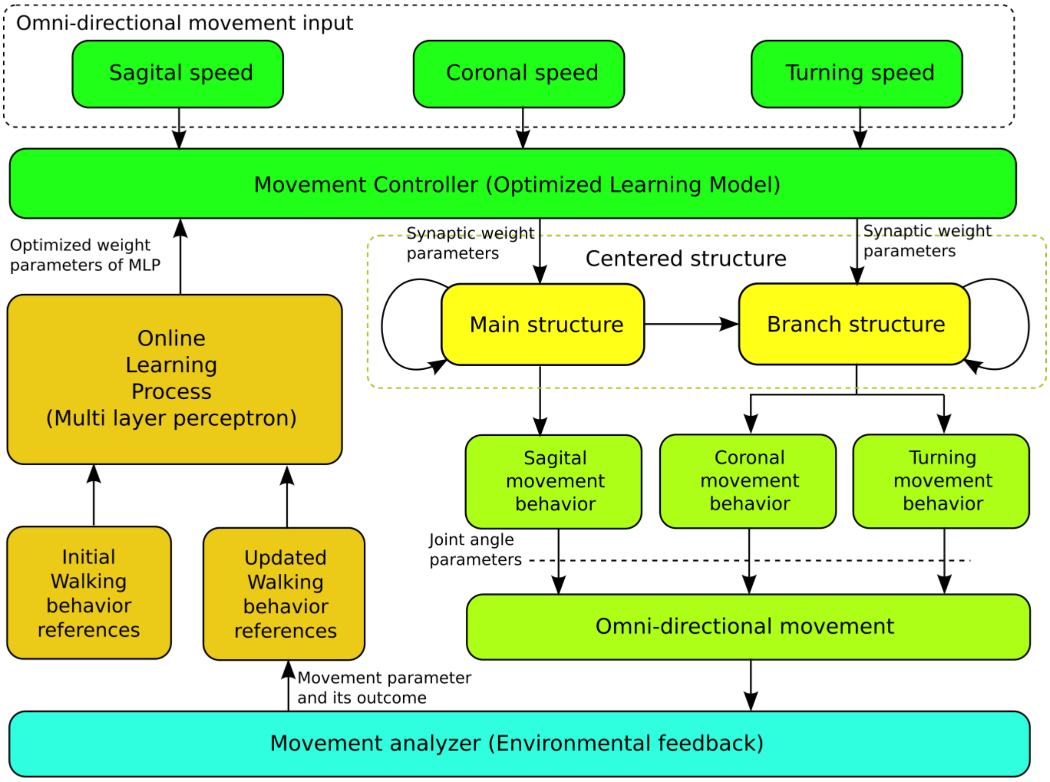

Omni-directional movement input
Sagital speed
Coronal speed
Turning speed
Movement Controller (Optimized Learning Model)
Optimized weight parameters of MLP
Synaptic weight parameters
Centered structure
Synaptic weight parameters
Main structure
Branch structure
Online Learning Process (Multi layer perceptron)
Sagital movement behavior
Coronal movement behavior
Turning movement behavior
Joint angle parameters
Initial Walking behavior references
Updated Walking behavior references
Omni-directional movement
Movement parameter and its outcome
Movement analyzer (Environmental feedback)

Figure 5.7: *Online model of centered interconnection structure of omni-directional neuro based locomotion model*

.

sagital walking behavior which has important role for generating the walking phase and pattern, and branch interconnection structure represents turning and coronal behavior.

In centered learning model for omni-directional movement, I used modified multi layer perceptron (MLP) which given some initial relationship walking pattern references generated from behavior learning system in Section 5.2.4. The mathematical equations of the proposed MLP has been explain in [243]. In order to optimize every walking pattern reference, I used learning model explained in Section 5.3. Online terminology representing online reference data will be updated from the walking parameter generated by optimized learning model and walking analysis parameter. Reference data in this learning model, compose 11 input walking provision and 72 output parameter representing walking parameter. Since the reference data will be updated every walking generation, the error resulted between desire walking provision and the result of walking which its parameters generated learning model will be reduced. In this model, unscaled omni-directional control and combination of walking behaviors are learned. Therefore after learned, walking provision in omni-directional movement without scalling parameter can be as the input parameter in locomotion generator. The synaptic weights in neural structure are generated based on the input of walking provision.

5.2.3 Experimental Result and Discussion

In order to prove the defectiveness of the proposed model, I implemented the proposed model into computer simulation and using Open Dynamic Engine as the physics engine. I designed biped robot with 4 DoF in each legs which can be seen in Fig. 5.4. In first experiment, I optimized the interconnection structure respected to sagittal direction as the center structure in order to get different speed of walking behavior. In second experiment, the branch structures (Coronal direction and turning behavior) and their interconnection with center structure will be optimized.

5.2.3.1 Sagital Movement Optimization

In this optimization, I generated the walking pattern in different walking speed between 0m/s - 5m/s with iteration 0.2 m/s unit speed. I will generate the walking pattern with minimum energy required in every desire walking speed. I run the evolution process until 50 generations and took the best individual in every desired walking speed as the walking pattern references. The sample of evolutionary process can be seen in Fig. 5.5. They also show that the population is approaching the minimum error value in both fitness evaluation. Therefore, the optimization will generate the desired movement with minimum energy required.

From this experiment, I took the best pareto front solution in every desired sagital speed. Therefore, one desired speed may generate more than one solution with different structure. Since I had 26 desired sagital speed, I acquired 26 unique interconnection patterns. The sample of sagital behavior can be seen in Fig. 5.6.

5.2.3.2 Coronal Movement Behavior

In coronal movement behavior, I generated 9 desired coronal movement speed in every interconnection patterns, from 0.8 m/s to -0.8 m/s with interval 0.2 m/s. I chose the best solution in pareto front from with minimum desire speed error in every optimization. Therefore, this coronal optimization I generated 234 interconnection structures. The sample of turning behavior can be seen in Fig. 5.8.

5.2.3.3 Turning Movement Behavior

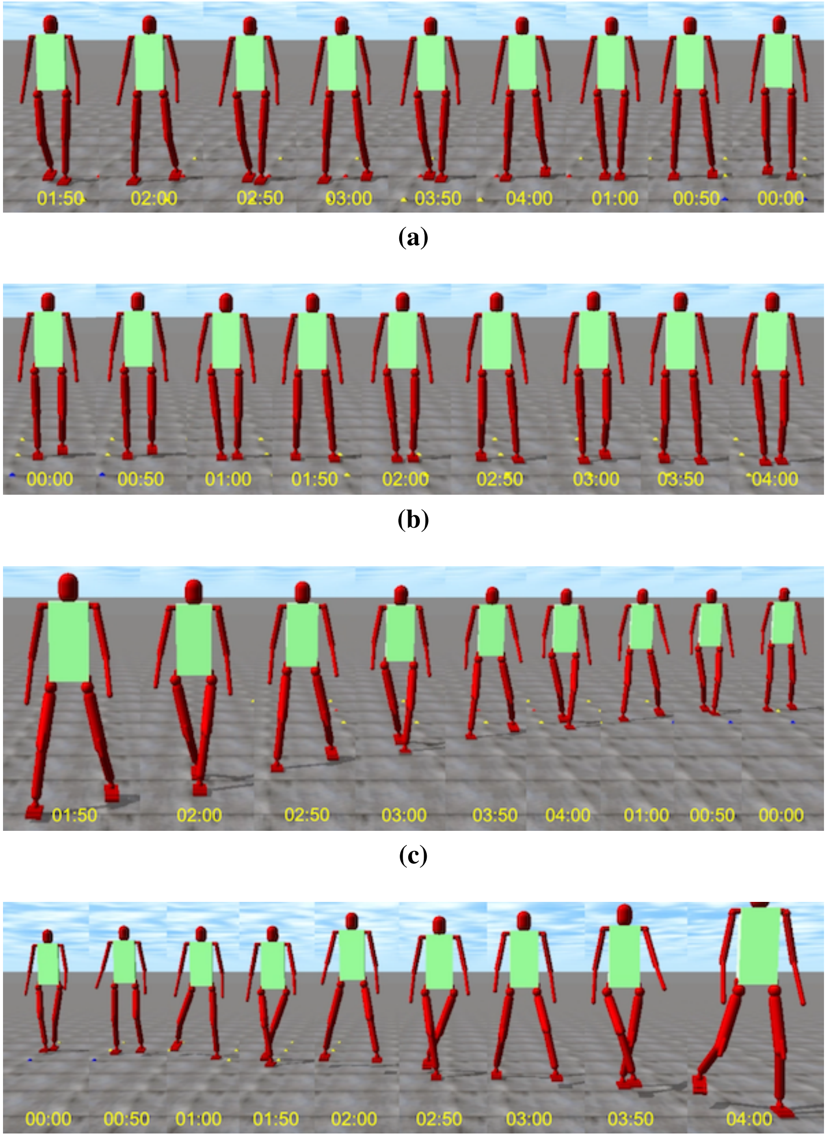
In turning movement behavior, I generated 9 desired turning movement speed in every interconnection patterns, from $-90^0/s$ until $90^0/s$ with turning speed interval $22.5^0/s$. I also chose the best solution in pareto from with minimum desire speed error in every optimization. Therefore, optimization processes also generated 234 interconnection structures. The sample of turning behavior can be seen in Fig. 5.9.

Since, every optimization behavior in branch structures had 234 interconnection structures, I got 468 behavior references in different movement provision.

5.2.4 Learning of Omni-directional Walking Behavior

After having generated behavior references in every movement behavior, I conducted the unscaled omni-directional motion behavior by using proposed MLP. In this learning process, I would like to achieve the relationship between walking input (degree of turning, sagittal speed, coronal speed, phase different of 8 joint angle signals) and the interconnection structure of neural oscillator. There were 11 neurons in input layer, 20 neurons in every hidden layer, and 72 output layer.

In order to decrease the complexity, I divided the MLP system into 36 system. Therefore, every MLP had 2 neurons in output layer, where the structure can be seen in Fig. 5.10. In order to generalize the system, I used k-fold cross validation, where the k value is 5. I trained the training data (walking references data) until 2.10^6 iterations and the relationship error can be decreased as shown in Fig.5.12.



(a)

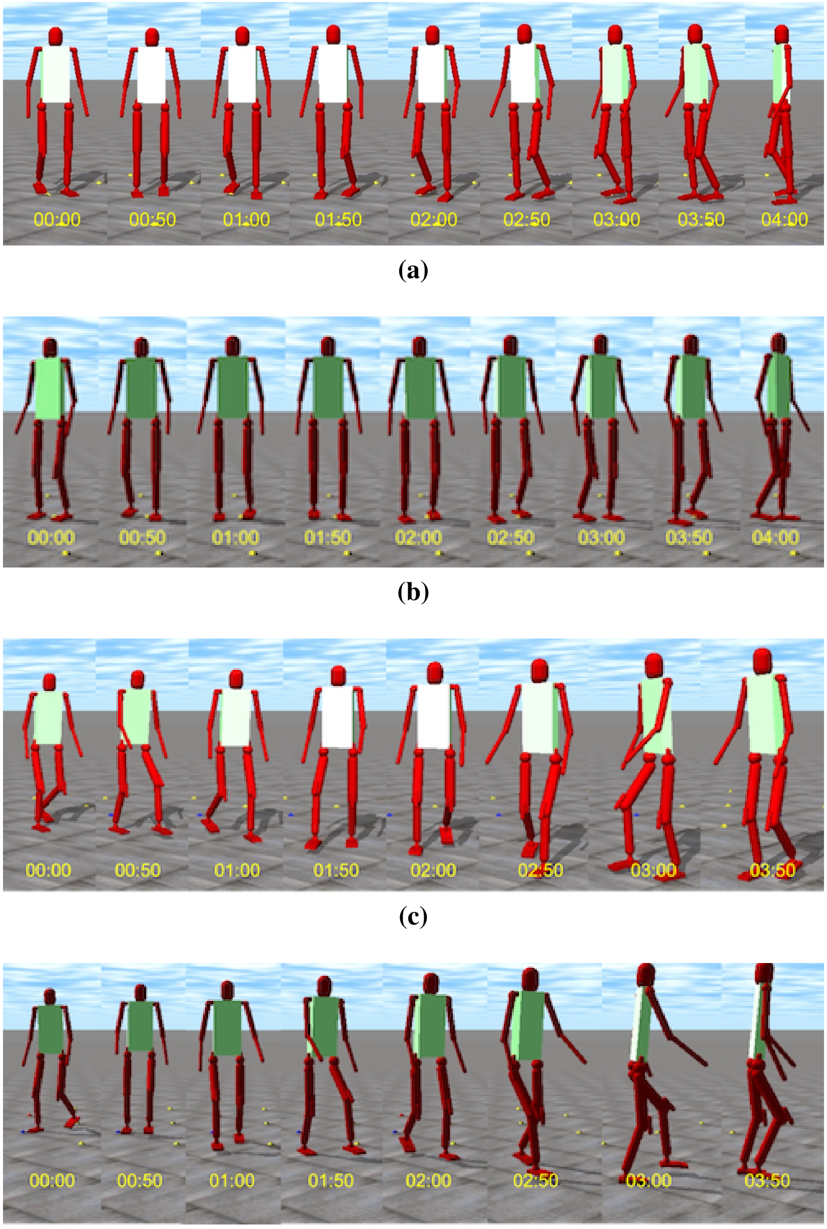
(b)

(c)

(d)

Figure 5.8: *Generated coronal movement behavior in different speed (a) slow coronal walking behavior in right direction (b) left direction (c) fast coronal walking in right direction (d) left direction*

.



(a)

(b)

(c)

(d)

Figure 5.9: *Generated turning movement in different behavior (a) turning right on a place (b) turning left on a place (c) walking with turning right (d) walking with turning left*

.

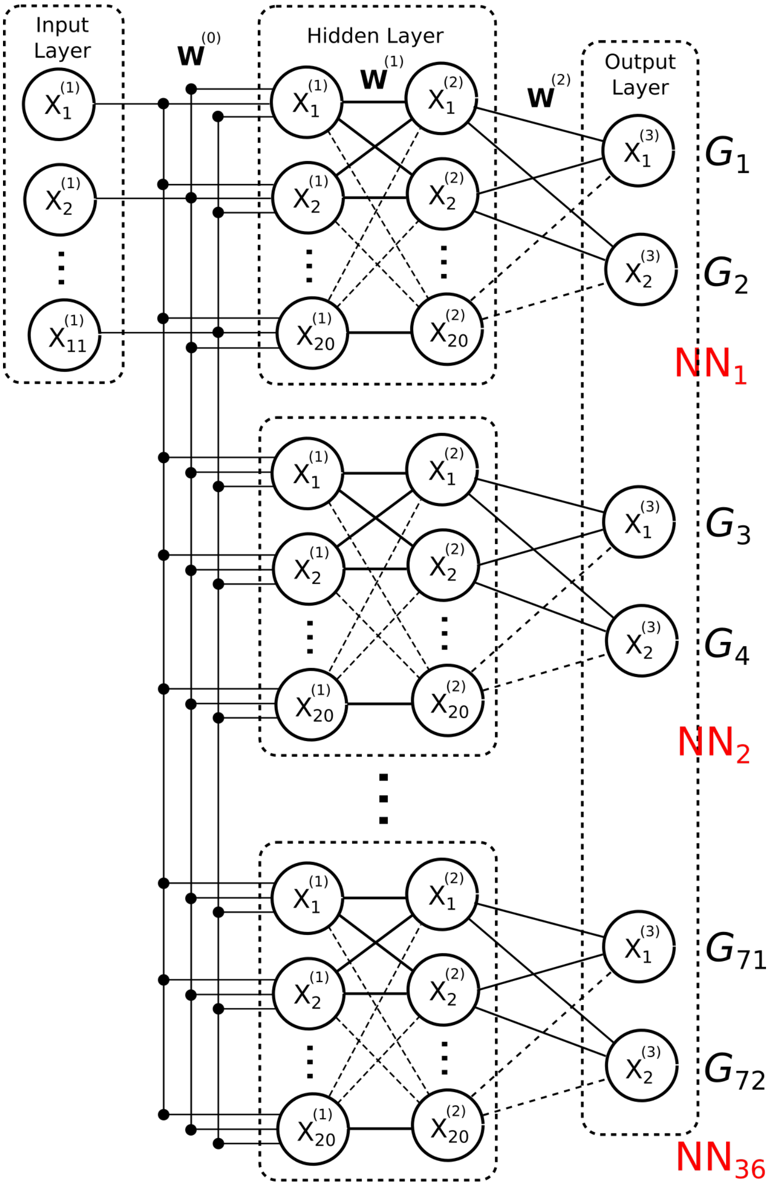
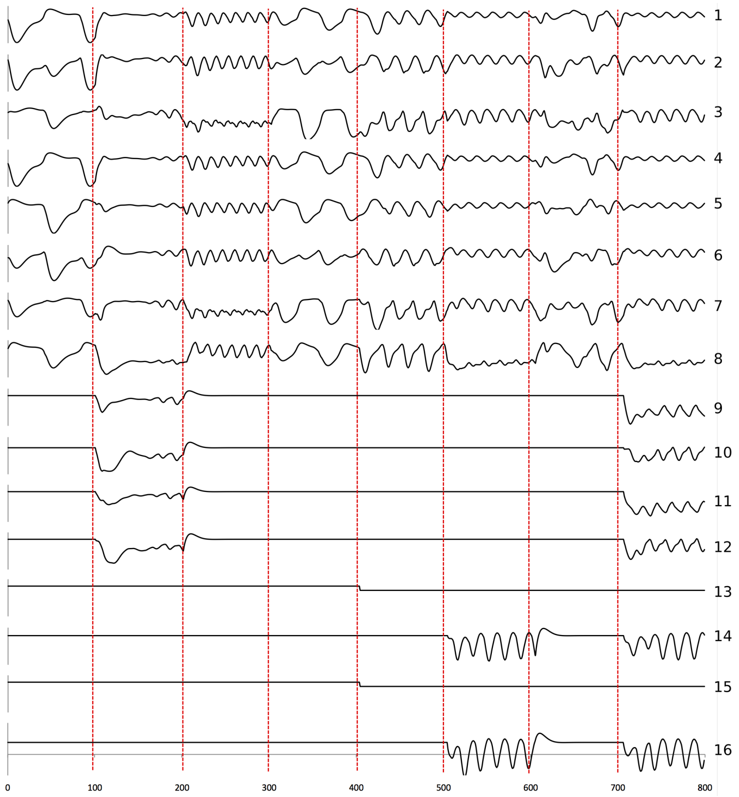
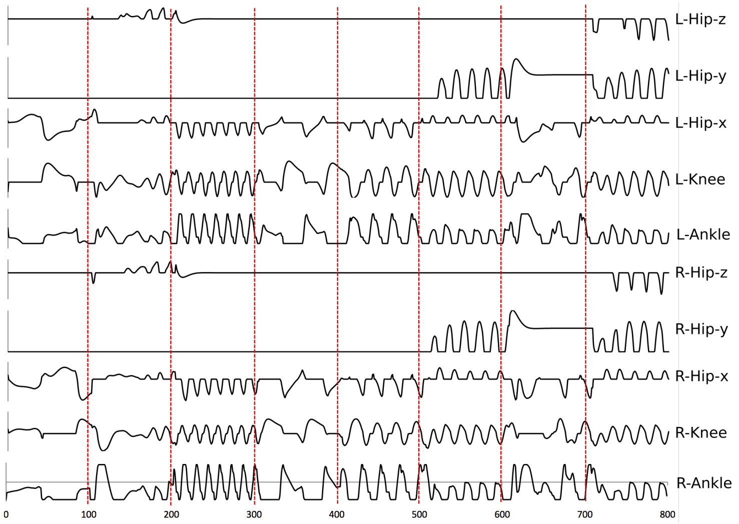


Figure 5.10: *Proposed MLP structure*

.



(a)



(b)

Figure 5.11: *Generated dynamic signal movement in different behavior (a) signal output of the motor neurons (b) the signal output converted in joint angle level*

.

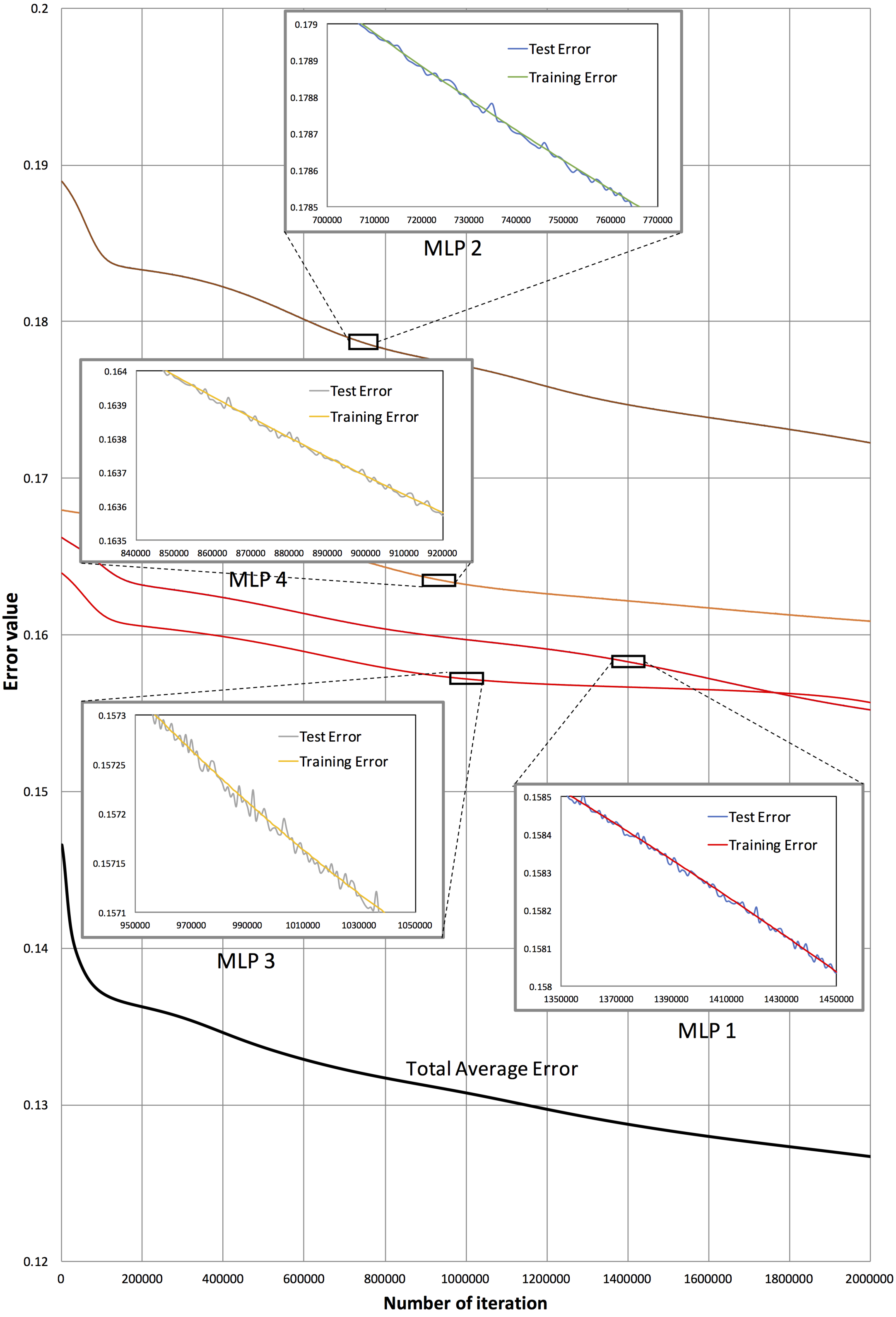


Figure 5.12: *Average error evolution with 4 samples of MLP's cross validation errors*

.

Using this proposed learning model, dynamic signal in unscaled omni-directional walking controller can be achieved. Fig.5.11a show the neuron signal outputs with movement provision changing in every 100 time sampling and Fig.5.11b shows the dynamic signal in joint angle level.

5.2.5 Conclusion and Discussion

This section proposed the learning strategy for solving the complexity of neural structure in neural oscillator based locomotion for generating omni-directional movement behavior. Since multi-objective evolutionary algorithm is used as the behavior optimization, movement provision and resulted energy are calculated as the fitness calculation, desired movement behavior can be acquired with minimum energy required. In this optimization, the behavior solutions was not deserved well, however, by increasing the number of generation, good diversity can be acquired. In this proposed method, walking behavior references generation took a long time, one reference behavior requires one optimization process. This is one of the problems which should be solved in next development.

Walking behavior references is used as the training data of the learning model. Separated MLP strategy is successfully approaching the relationship between input and output of walking behavior. Training iteration and learning process take a long time, and also require higher performance of learning model, such deep learning model. Input behavior references are also required to be increased in order to specify variant behavior in one movement provision, since in this current state, redundant behavior may be generated in the walking reference optimization. However, by using this proposed learning model with cross validation, omni-directional movement behavior can be generated. The model can generate sagittal movement from 0 m/s - 25 m/s, coronal movement from -5 m/s - 5 m/s, and turning speed $(-\pi/2)^0/s$ - $(\pi/2)^0/s$.

In the future, variant of walking input references should be increased by considering internal and external sensory information in order to classify redundant behavior. In order to improve the performance of learning model, high performance deep learning with online application may be implemented.

5.3 Tree-structured Learning Model and Optimization of CPG

As naturally evolved, the human or animal locomotion process is the best model generating efficient movement and energy which has been optimized genetically. Because of that, several researchers tried to imitate this natural process which is called bio-inspired model. Besides, many researchers applied control based locomotion to achieving instant desired performance. This section proposes the neural based locomotion using CPG model that can be applied in many kinds of environmental condition inspired by the evolutionary process of bacteria as the optimization model. Optimization is required for acquiring the best stability in bio-inspired based locomotion.

Biological approach in locomotion is the alternative model of realizing the dynamical system in order to move in an unpredictable area or any environmental conditions such as rocky surface, soft, slippery ground and so on. The robot can be applied as a multi-purposed robot in this way. There are still many constraints in this approach such as environmental constraint, neuron constraint, joint constraint etc. Some researchers applied their locomotion system in certain environmental condition, in certain slope terrain, and in certain slippery level. Bio-inspired approach was also applied for identifying the appropriate number of legs of a multi-legged robot in certain environmental condition [272]. Muscle-based skeletal model is the part of bio-inspired locomotion as well. Topchyan et al. applied it for controlling the biped robot locomotion [290]. Currently, muscle-based locomotion is applied only for animation or simulation purposes [80, 152, 290, 89, 303]. Therefore, I implement sensory-motor coordination which can be applied for robotic application using sensors. Geijtenbeek et al. built muscle-based locomotion control for 3D simulated biped robot where muscle forces are activated by signal activation [80], while Wang et al. built it for 2D simulation [303]. Lee et al. also realized muscle-based locomotion for humanoid models actuated by more than one hundred Hill-type muscles [152]. However, the muscle based and angluar based actuator has own advantages.

Locomotion models based on a biological approach were proposed by several researchers [120, 119, 122, 279, 189, 197, 247]. In bio-inspired approach, most researchers applied two neurons representing one joint that generated mutual inhibition between certain neurons called central pattern generator (CPG). This approach is inspired by the spinal cord system in the vertebrates. Matos et al. implemented CPG in their bipedal robot locomotion [180]. Nassour et al. proposed a neuro-locomotion model based on multi-layered neuron structure for bipedal robot locomotion. Their proposed locomotion model achieved satisfactory walking pattern with good stability [197].

Bio-inspired adaptive locomotion implies the activity of the sensory-motor coordination. Sensory information has to be utilized as the feedback input. However, this sensory-motor coordination requires a complex interconnection structure. The major issue in this model is the optimization. Some researchers used genetic algorithm to evolve the configuration of the coupled neural oscillator in order to optimize the locomotion gaits [176]. However, that does not represent the sensory-motor coordination. The dynamic structure of sensory-motor neurons is hard to be optimized. Some researchers defined the neuron structure based on the case study, in a certain environmental condition. A little difference in the implementation of its parameter causes a big influence in the resulted locomotion pattern. Therefore, I represent the neural interconnection by tree structure for simplifying the complexity of the structure.

In the optimization model, some researchers used genetic algorithm to optimize the performance of locomotion [19, 310]. Baydin et al. applied genetic algorithm to form the walking pattern applied in a 5-joint biped robot [19]. In 2010, Park et al. designed a locomotion using an evolutionary optimized CPG. They also proposed sensory feedback to support the walking model [211]. In order to define the dynamic problem with unpredictable model, some researchers applied Genetic Programming (GP) [143], the extension model of the genetic algorithm [110]. GP can also be applied for the bio-inspired locomotion [69, 204]. Some researchers used GP in order to synchronize several joints in the locomotion generator [69].

In bio-inspired locomotion, I started to develop neural oscillator based 4-legged animal locomotion. I designed a fix neural structure and applied multi-objective evolutionary algorithm for the synaptic weight optimization [259]. Furthermore, I modified the neural structure for biped robot locomotion [245]. These systems still have high constraint in their experiments. I had to conduct preliminary tests in order to design the appropriate neural structure. Therefore, neural structure should be evolved in order to adapt to unpredictable environmental conditions.

Evolving neural networks were applied to increase the performance of the locomotion by either decreasing the computational cost or increasing the walking stability. Some researchers conducted the evolving neural network in locomotion. Inden et al. developed the evolving neural network to control a multi-legged robot with a ball-like morphology to move on rough terrain [125]. Evolving spiking neural network was also applied to increase the performance of the locomotion of a 4-legged robot [280].

In unpredictable environmental condition, I conducted the interconnection structure optimization in neural oscillator based locomotion which was applied in a humanoid robot [255]. In that research, the interconnection structure can be optimized, but the number of neurons was fixed and could not be optimized. Therefore, a dynamical number of neurons is required to be implemented for increasing the dynamic level of locomotion.

Based on that, I aim to develop an evolving neural oscillator in humanoid robot locomotion that can optimize not only the interconnection structure of the neurons but can also reconstruct the number of joints and the number of neurons in each joint depending on the given environmental condition. The idea is, that either the number of joints, or the number of neurons, or the interconnection structure are dynamically changed depending on the condition acquired from the sensor equipped in the robot. For simplifying the optimization strategy, I represent the solution of the neural interconnection structure by a tree structure. While some researchers applied GP as the optimization model in robot locomotion, I apply Bacterial Programming (BP) [30] for optimizing the neural structure of neural based locomotion model. This optimization approach is inspired by the evolutionary process of bacteria. Bacterial programming is implemented for optimizing the system in several environmental conditions.

5.3.1 Sensory-motor Interconnection Model

In this model, I implement central pattern generator as the locomotion generator. I integrate the motor neurons as output signal and sensory neurons as the feedback signal. I use internal sensory information as the feedback signal which are tilt sensor (pitch and roll) and ground touch sensor in both legs. In this locomotion model, the number of joints which are activated in the robot's legs depends on the environmental condition. This system deactivates unused joints therefore it can reduce the energy consumption and computational cost. In the system when a joint is deactivated, the neuron that represents it will not be calculated in the proposed system. It can reduce the computational cost and energy consumption in the CPU or Microcontroller. In addition, I may consider the number of joints required in a certain environmental condition. It can also reduce the inefficient motor usage and energy consumption.

I design the locomotion including 3 aspects which are gait generator system, optimization system, and adaptive intelligent control system. The role of the gait generator is to generate the walking pattern signal in joint angle level based on neural oscillator. The role of the optimization system is to optimize the best neuron structure of gait generator in certain environmental condition. This system creates several best neuron structures in different environmental conditions. The adaptive intelligent control has to control the neural structure against the change of environmental conditions.

The gait generator is composed of the dynamic neural structure which is inspired by the spinal cord system in vertebrates that produce dynamic signal for locomotion. This signal model is generated by mutual inhibition between certain neurons in a neural structure. Each neuron also acquires adaptation signal input. The rhythmic signal activity generated by the neural oscillator consists of multi tonically excited neurons with the self-inhibition effect linked reciprocally via an inhibitory connection. In Matsuoka's model, the signal is generated by coupled neuron and it was implemented by several researchers for locomotion where a coupled neuron represents one joint angle [183, 181].

The neural model of joints in this proposed research is illustrated in Fig. 5.13 for biped robot and Fig. 5.14 for quadruped robot. Their mathematical model is shown in Eqs. (5.13), (5.14), (5.15). The motor neurons interconnection is represented by parameter w_{ij} and the sensory motor neurons interconnection is represented by m_{il}.

$$\tau \dot{u}_i = \left(u_0 - u_i - \sum_{j=1}^{n} w_{ij} y_j J_j^k + \sum_{l=1}^{n} m_{il} s_l - b v_i \right) \tau_f \tag{5.13}$$

$$\tau' \dot{v}_i = (y_i - v_i) \tau_f \tag{5.14}$$

$$y_i = \max(u_i, 0) \tag{5.15}$$

$$\Theta_i^{(l,r)} = \sum_{j=1}^{N_i^{neuron}} J_j^{l,r} y_j \tag{5.16}$$

where u_i, y_i, and v_i are the inner state, output value, and a variable that represents the adaptation value or the self-inhibition effect of the ith neuron, respectively; J_j^k is the operator representing +1, -1, or 0; s_l is the output of the lth sensory neuron; b is the rate of the adaptation value. The external input for coupled neurons, which has a constant rate, is denoted by u_0. The time constant of the inner state and the adaptation effect in the neuron are represented by τ and τ', respectively. In Eq. (5.13), w_{ij} represents the strength of the inhibitory effect between the motor neurons that is optimized offline; m_{il} represents the strength of the sensory signal effect to the motor neurons; $\sum_{j=1}^{n} w_{ij} y_j$ represents the total of the signal input from the motor neurons; $\sum_{l=1}^{n} m_{il} s_l$ represents the total of the feedback signal from the sensory neurons. In Eqs. (5.13) and (5.14), τ_f is used for controlling the frequency of oscillation. In Eq. (5.16), J_j^i is the neuron effect of the jth neuron in the ith joint represented by joint node in the tree structure of bacterial programming (see Section 5.3.2) and N_i^{neuron} is the number of neurons in the ith joint.

In our proposed system, one joint angle is represented by uncertain number of neurons depending on the requirement of the given environmental condition. Sometimes, a joint angle needs only one neuron for generating the rhythmic signal and sometimes it needs three neurons for generating the complex signal pattern.

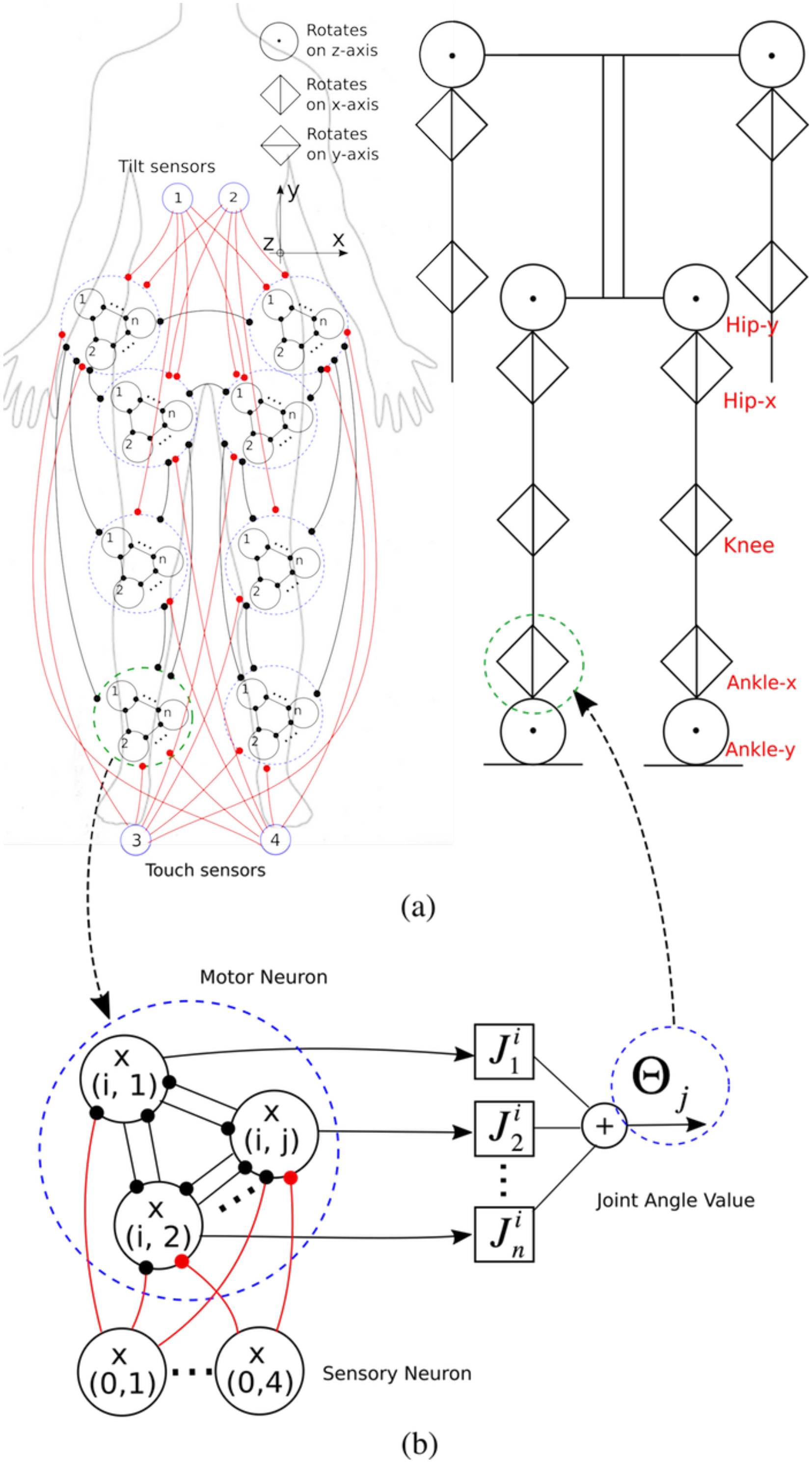


Figure 5.13: *Representation of the evolving neural model. The black line represents the motor–motor interconnection and the red line represents the sensory–motor interconnection. (a) Neural structure and mechanical structure of the robot. (b) Neuron structure in joint angle level.*

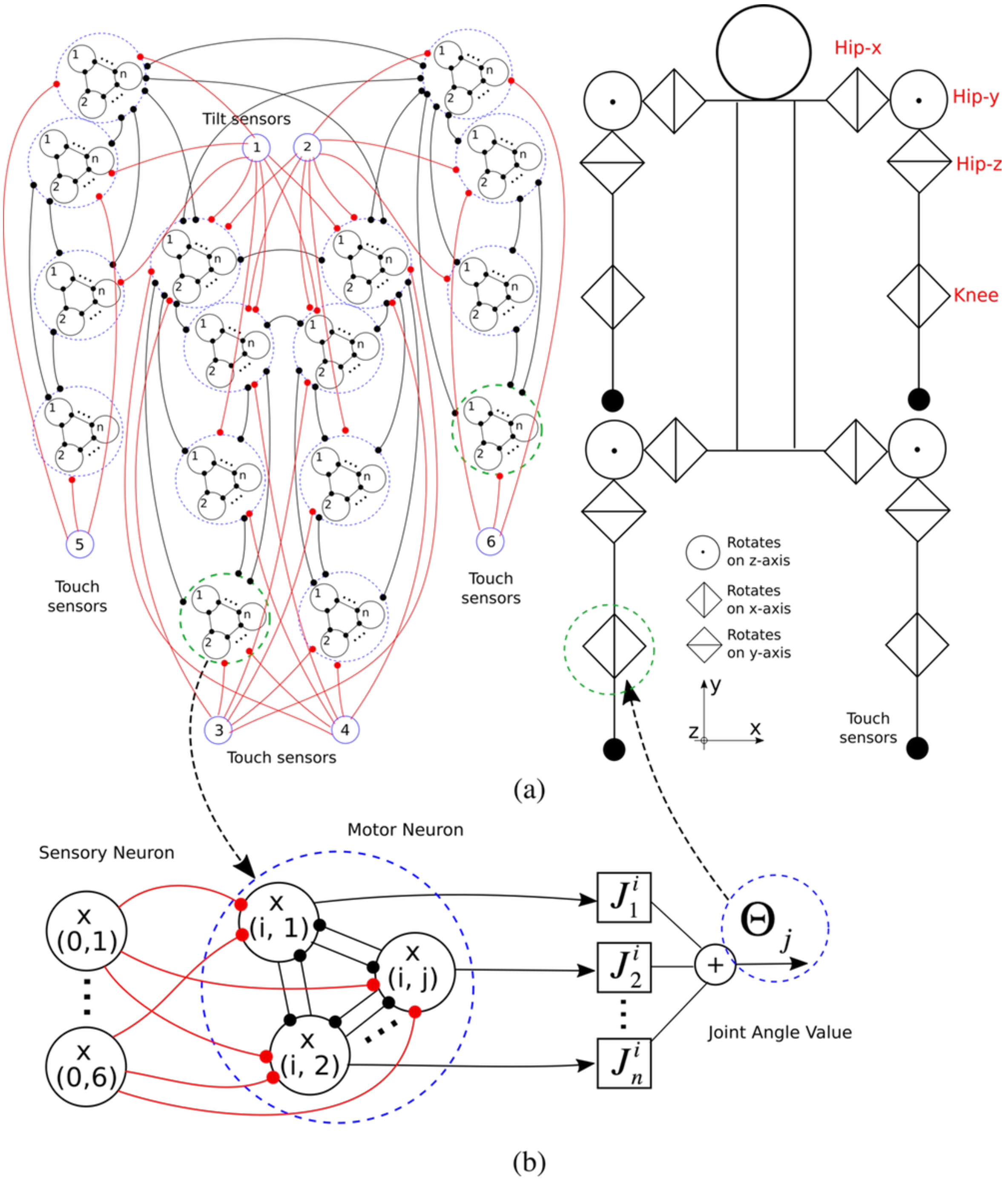


Figure 5.14: *Representation of the evolving neural model. The black line represents the motor–motor interconnection and the red line represents the sensory–motor interconnection. (a) Neural structure and mechanical structure of the robot. (b) Neuron structure in joint angle level.*

In the walking pattern, knee (θ_2^d) and hip joints as the main joints (θ_0^d & θ_1^d) have a role to form the walking pattern. Ankle joints (θ_3^d & θ_4^d) adapt the condition of the main joint and support the stabilization and landing system in locomotion. Sometimes, ankle joints are required for forming the walking pattern in certain condition. The mathematical equation of the angle value in every joint can be seen in Eqs. (5.17), (5.18), (5.19), (5.20), and (5.21), where θ_k^d represents signal to the robot's joint at k joint ID on d side, (l) and (r) represents left and right side; Θ_k represents the signal output from the neuron in joint angle level.

$$\theta_0 = \begin{cases} \theta_1^{(l)} - \theta_1^{(r)} & \text{if } N^{joint} < 3 \\ \Theta_3 & \text{otherwise} \end{cases} \tag{5.17}$$

$$\theta_1^{(l,r)} = \Theta_0 \tag{5.18}$$

$$\theta_2^{(l,r)} = \Theta_1 \tag{5.19}$$

$$\theta_3 = \begin{cases} \theta_2^{(l,r)} - \theta_1^{(l,r)} & \text{if } N^{joint} < 4 \\ \Theta_4 & \text{otherwise} \end{cases} \tag{5.20}$$

$$\theta_4^{(l,r)} = -\theta_0^{(l,r)} \tag{5.21}$$

When the environmental condition requires the robot needs only two joints in each leg (N^{joint} = 2), the system prefers to choose the hip joint in x-axis position and knee joint to be activated. When the environmental condition requires the robot needs three joints in each leg (N^{joint} = 3), the system prefers to choose the hip joint x-axis, y-axis position, and knee joint to be activated. When the environmental condition requires the robot needs more than three joints (N^{joint} = 4), the system activates ankle joint as the additional joint. This condition represents the high complexity of the environment when the robot needs to activate the ankle joint. The neural representation of the robot structure and the neural model in joint angle level are depicted in Fig. 5.13.

This proposed system will be applied in a biped robot which has 2 legs and 5 joints in each leg. However, only the important joints are generated by motor neurons. The joint structure of the robot is depicted in Fig. 5.15. In this system I ignore the function of actuator in upper body.

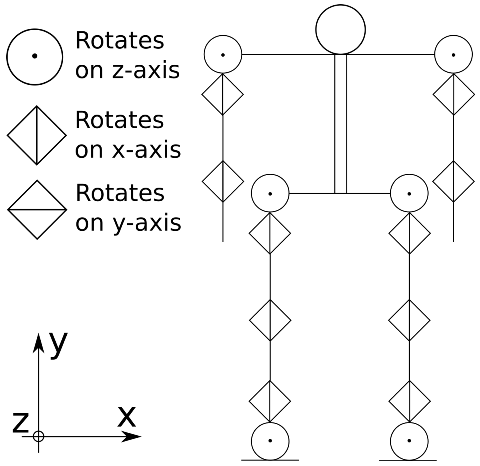


Figure 5.15: *The mechanical structure of the robot*

5.3.2 Optimization Strategy

Evolutionary computation was implemented in several case studies. Several researchers implemented it for locomotion [282, 155, 204, 271]. Most of them are inspired by a genetic based evolutionary process. Tanev et al. applied interactively learned consensus sequence (ILCS) GP for efficient evolution of simulated snake-like robot (Snakebot) situated in a challenging environment. Their proposed research can improve the efficiency of GP compared to canonical GP [282]. In his other work, Tanev proposed an approach for incorporating learning probabilistic context-sensitive grammar in genetic programming applied also for Snakebot [281]. Kuyucu et al. proposed a genetic transposition inspired genetic programming applied for the coevolution of locomotion gaits and sensing of Snakebot [146]. Chee et al. proposed a hybridized genetic programming and self-adaptive differential evolution algorithm to simultaneously co-evolve both the morphology and controller of a snake-like modular robot [41]. Wolff et al. proposed linear genetic programming to generate gait control programs for simulated bipedal robots [311]. In some research, genetic programming was applied for optimizing the weights in complex motor pattern generation (MPG). Lewis et al. applied it in a six-legged robot. First, MPG was evolved for one leg. Then, a network was evolved to coordinate all legs. Staged evolution could improve the algorithm's performance [155]. Genetic programming can also be applied to evolve neural oscillators producing coordinated movements of human-like bipedal locomotion [204][190]. Silva et al. used GP as an automatic search method for motion primitives of a biped robot that optimizes a given criterion. Feedback pathways were directly included into the evolutionary process through sensory inputs in order to increase the adaptability of the achieved solutions. Their proposed research was applied in Darwin OP robot [271].

A slightly different evolutionary computation technique which is based on the combination of Bacterial Evolutionary Algorithm [198] and Genetic Programming [143] is Bacterial Programming [30]. This algorithm is based on the bacterial operations, however, it uses tree structures similar to the ones in the GP. Bacterial Programming was successfully applied for designing B-spline Neural Networks [30] and Hierarchical Fuzzy Systems [17]. It provided superior results compared to GP.

The novelty of our proposed model is inspired by the evolutionary process of bacteria applied for humanoid biped robot locomotion. Bacteria can transfer genes to other bacteria. The bacterial mutation performs local optimization whilst the gene transfer allows the bacteria to directly transfer information to the other individuals in the population. Based on these bacterial operations but using tree structures I optimize the neural structure of locomotion in certain ground condition and certain slope terrain. Another novelty of our approach is in the BP compared to [30] and [17]. A novel initial population generation technique and an improved gene transfer operation are proposed in this section.

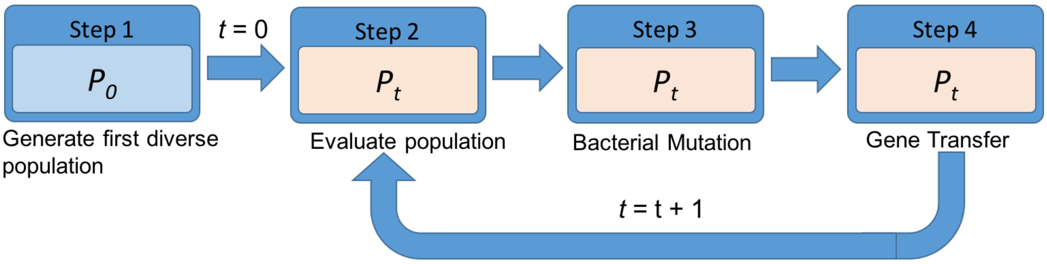


Figure 5.16: *Process of bacterial programming*

In BP, the population is composed of several individuals that are represented by a tree model. The parameter structure is explained in Section 5.3.3. The evolutionary process of BP is depicted in Fig. 5.16. First, an initial population is generated (Section 5.3.3.1). After that, the evolutionary process is started. Bacterial mutation operation (Section 5.3.3.2) and gene transfer operation (Section 5.3.3.3) are applied until the desired criterion is achieved. The evolutionary process is conducted until the desired number of generations (N_{gen}) is achieved. Beside the encoding process, another key point of all evolutionary computation techniques is the evaluation of individuals, which is important for the evolutionary operations. This is discussed in Section 5.3.3.4. The process of BP can be seen in Algorithm 10.

5.3.3 Encoding Model

One bacterium's parameter packet depicted in Fig. 5.17 is composed of several parameters representing the neural oscillator based locomotion.

Algorithm 10 Process of Bacterial Programming

```
//Generating an initial population:
for $i \leftarrow 1$ to $N_{ind}$ do
    Generating tree structure of the $i$th individual
    Evaluating the $i$th individual
//Evolutionary loop:
for $i \leftarrow 1$ to $N_{gen}$ do
    Bacterial Mutation operation
    Gene Transfer operation
    Evaluating the population
```

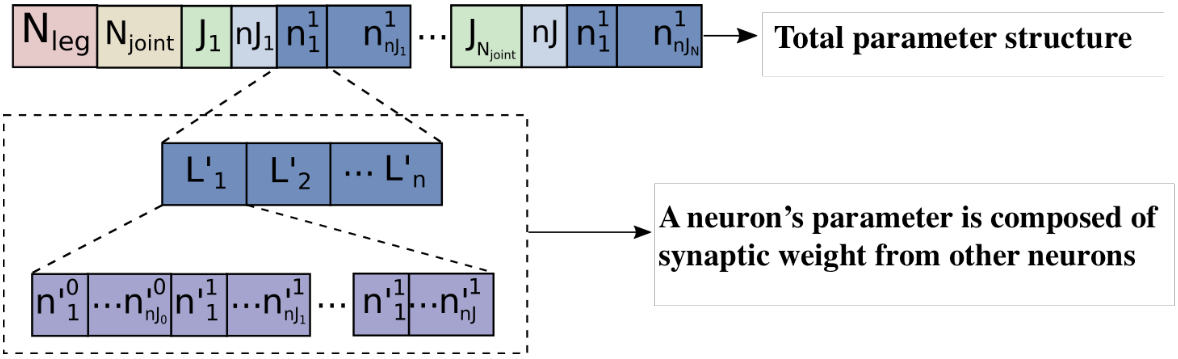


Figure 5.17: *Packet of parameters in bacterial structure*

N^{leg} is a random integer between N^{leg}_{min} and N^{leg}_{max} which represents the number of legs. N^{joint} is a random integer between N^{joint}_{min} and N^{joint}_{max} which represents the number of joints in each leg. N^{neuron} is a random integer between N^{neuron}_{min} and N^{neuron}_{max} which represents the number of neurons in each joint. $L^{'}_{k}$ represents a packet of neurons. The number of neurons in the kth joint is notated by nJ_k, and n^k_l is a random value between n^{min} and n^{max} representing the set of synaptic weights from other neurons to the lth neuron in the kth leg.

In order to simplify the complexity of the neural oscillator model for humanoid biped locomotion, one bacterium is represented by a tree structure, composed of joint and neuron nodes as depicted in Fig. 5.18a.

A joint node can be composed of operators "+", "-", or "0" which represent addition operation, reduction operation, and no operation, respectively. These operators are randomly generated and will effect to the motor neuron signal calculated in Eq. (5.13). A neuron node is composed of the synaptic weight values from other neurons.

This tree structure, as well as the characteristics of the nodes, evolves from generation to generation. The tree model structure is converted to neural oscillator structure which refers to a symmetric structure depicted in Fig. 5.18 in order to reduce the size of the search space. The neural oscillator structure of a leg is mirrored to another leg. Therefore, in a biped robot, the right leg mirrors to the left leg of the robot.

For example, from the evolutionary process, I get a tree structure (i.e. a bacterium) like in Fig. 5.18a, where the red circle (k,l) represents the synaptic weight value from the lth neuron of the kth leg to the neuron (m,n) with the black circle with the nth neuron of the mth leg. This information will be encoded to the synaptic weight parameter w_{ij}, where $i = (k-1) \cdot N_k^{neuron} + l$ and $j = (m-1) \cdot N_m^{neuron} + n$, where N_k^{neuron} and N_m^{neuron} represent the number of neurons in the kth leg and mth leg, respectively. The black circle represents the inner state of neuron calculated in Eqs. (5.13), (5.14), (5.15). The blue square represents the signal effect in joint angle level. The red and blue connections in Fig. 5.18 represent motor neurons interconnection and sensory-motor interconnection, respectively.

The tree structure in Fig. 5.18a has 2 joints which are hip-x joint and knee joint and it is converted to neural structure that can be seen in Fig. 5.18b.

5.3.3.1 Initial Population Generation

I propose a novel initial population generation for BP. I randomly generate N_{ind} individuals (i.e. bacteria) one by one. In order to create a diverse population, I check the similarities to the already generated individuals. If the newly generated bacterium has similarities to any already generated bacteria, then the bacterium is regenerated until it has no similarity to any bacteria.

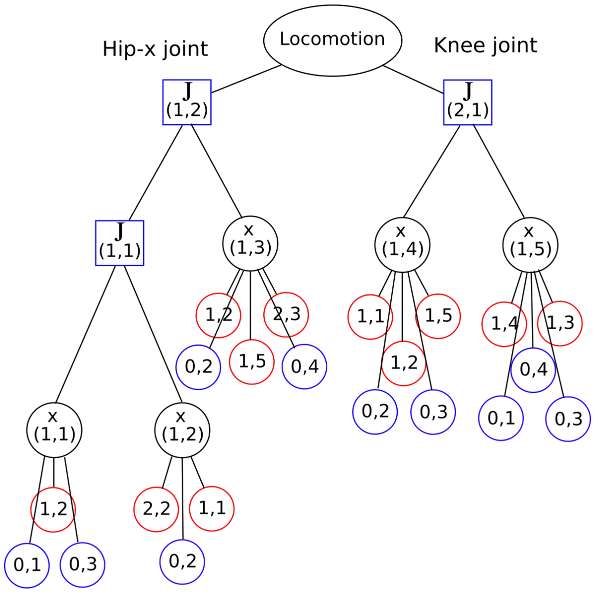
The steps of checking the structure similarities between two bacteria are as follows:

1. Checking the number of joint nodes at every joint branch
2. Checking the number of neuron nodes at every neuron branch
3. Checking the number of neuron leaves at every neuron node.
4. Checking the ID of neuron leaves at every neuron node.

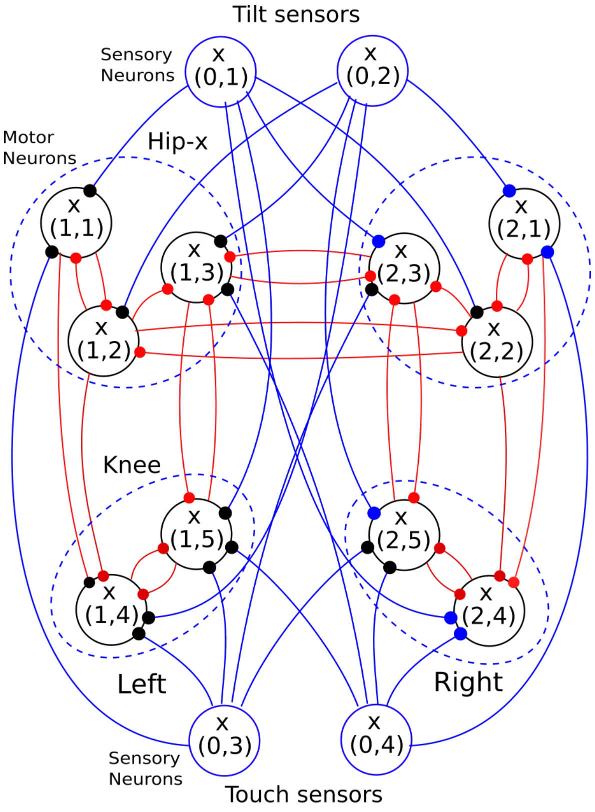
The similarity check is performed in a hierarchical way. If there is a dissimilarity in a step, I do not continue to the next step. If there is still similarity, next step checking is required.

5.3.3.2 Bacterial Mutation

The bacterial mutation is applied to each bacterium one by one. First, N_{clones} copies (clones) of the bacterium are generated. Then, a certain node of the bacterium is randomly selected and the subtree defined by this node is randomly changed in each clone (mutation). In our proposed method, because coding is given by an expression tree for neural oscillator



(a)



(b)

Figure 5.18: *a) The tree structure represents the neuro-based locomotion; b) Symmetric model of the neuron structure as the result of the tree structure conversion.*

based biped robot locomotion, there are two types of mutation nodes: joint node mutation and neuron node mutation. Next, all the clones and the original bacterium are evaluated by a performance criterion (see Section 5.3.3.4). The best individual among the clones and the original bacterium transfers the mutated part into the other individuals. This cycle is repeated for the remaining nodes until all of the subtrees of the bacterium have been mutated and evaluated. At the end, the best bacterium is kept and the remaining N_{clones} are discharged. The algorithm process of bacterial mutation can be seen in Algorithm 11. By the help of this operation, the bacterium will be at least as good as before, but in most of the cases it will be better. Bacterial mutation is more efficient than classical mutation in GA because of the nature of cloning. Every clone brings a new chance to find a better solution anywhere in the search space, thus a wider space can be explored. The bacterial mutation operation is illustrated in Fig. 5.19. Figure 5.19 shows two substeps of the bacterial mutation process, where three clones are applied. In the first substep a neuron node is selected randomly. The subtree defined by this neuron node is mutated in the three clones and left unchanged in the original bacterium. Then, all four individuals (the original bacterium and the three clones) are evaluated, and the best one is selected. Clone 1 is the winner, thus it transfers its subtree to the other three individuals (the original bacterium and two clones). Then, another node is selected from the original bacterium. This is a joint node, so a joint node mutation is performed on the clones. After the mutation, the best individual (among the three clones and the unmutated bacterium) is selected. Since there is no more unselected node in the original structure, this will be the final step, so the winner individual is kept as the new bacterium.

5.3.3.3 Gene Transfer

In the gene transfer operation, there is the exchange of genetic information between two bacteria. In this process, the population of bacteria is ordered according to the performance criterion (see Section 5.3.3.4). The population is divided into two halves: the superior half and the inferior half. Then, a superior bacterium and an inferior bacterium are randomly selected. The superior bacterium transfers one of its subtree to the inferior bacterium which overwrites one of its subtree by the transferred subtree. The above process (ordering the population, selecting the superior and inferior bacteria, and transferring the subtree) is repeated N_{inf} times. The gene transfer operation is illustrated in Fig. 5.20 and in Algorithm 12.

There is a novelty in the gene transfer operation in this section compared to the original BP. When selecting the subtree from the superior bacterium to be transferred, and the subtree in the inferior bacterium to be overwritten, I check the qualities of the subtrees. Based on the result of walking evaluation calculated in Eq. (5.24), a good subtree is transferred to

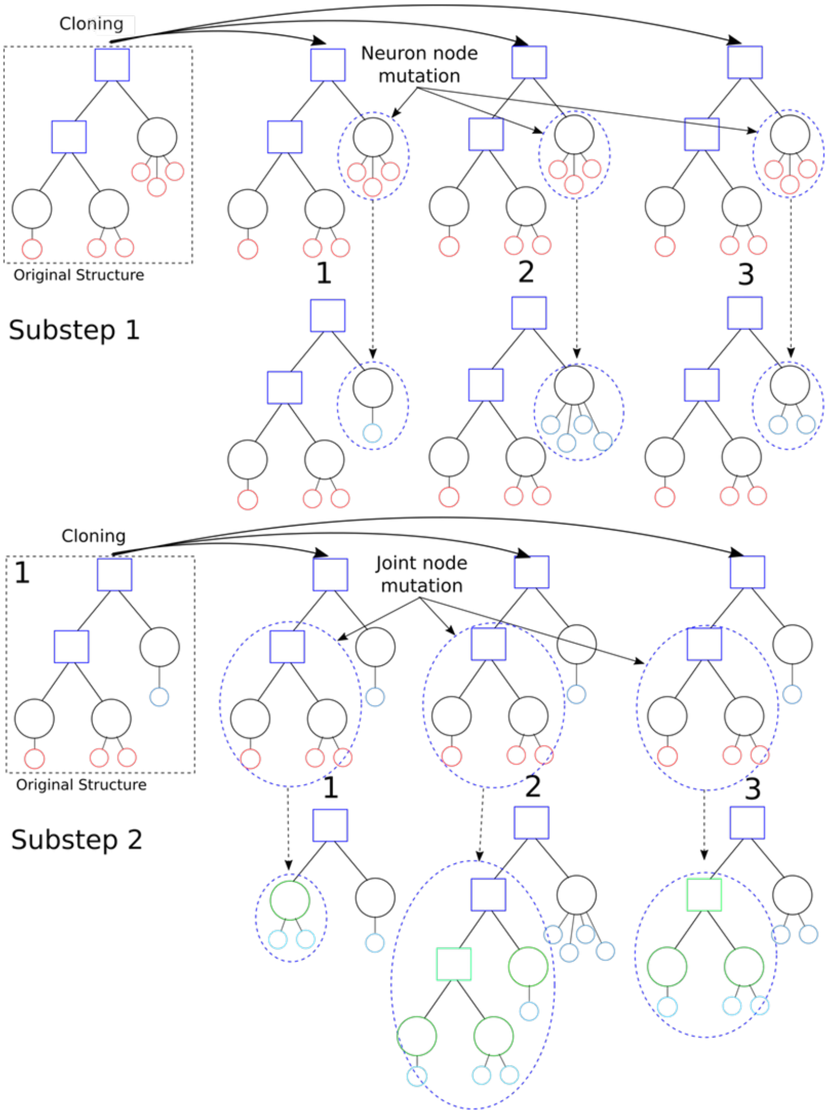


Figure 5.19: *Illustration of bacterial mutation*

Algorithm 11 Bacterial Mutation Operation

```
for $i \leftarrow 1$ to $N_{ind}$ do
    Cloning the $i$th individual
    repeat
        Defining a node randomly
        if node is a Joint node then
            //Joint node mutation in the clones
            for $j \leftarrow 1$ to $N_{clones}$ do
                Processing Joint node mutation
                Evaluating the mutated clone
        else
            //Neuron node mutation in the clones
            for $j \leftarrow 1$ to $N_{clones}$ do
                Processing Neuron node mutation
                Evaluating the mutated clone
        Selecting the best individual
        The best individual transfers the mutated subtree
        to the other individuals
    until all subtrees are mutated
    Keeping the best individual as the new $i$th individual
    Deleting the clones
```

overwrite a not so good subtree.

Algorithm 12 Gene Transfer Operation

```
for $i \leftarrow 1$ to $N_{inf}$ do
    Ordering the population
    //Selecting a random superior bacterium
    $Superior = RandomValue(1, N_{ind}/2)$
    //Selecting a random inferior bacterium
    $Inferior = RandomValue(N_{ind}/2+1, N_{ind})$
    Defining the subtree to be transferred
    from the Superior bacterium
    Defining the subtree to be overwritten
    in the Inferior bacterium
    Transferring and overwriting
    Evaluating the new Inferior bacterium
```

5.3.3.4 Evaluation

Evaluation calculation is represented by a walking analyzer. Walking speed and walking stability are required in the walking analyzer. The evaluation is formulated as a minimization problem. Two evaluation criteria are calculated in the evaluation process which are the

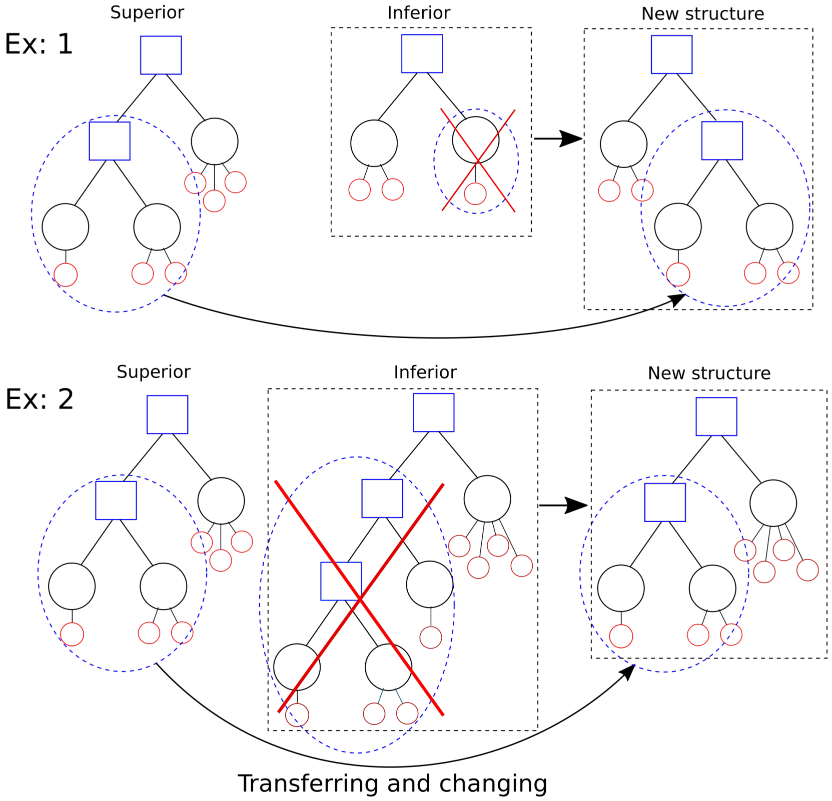


Figure 5.20: *Illustration of gene transfer*

desired walking length and the body tilt oscillation in pitch and roll direction $\bar{\dot{\beta}}$. The desired walking length, $\bar{\upsilon}$ is represented by the remaining distance to the target.

The value of tilt body oscillation represents the stability of movement. If the robot locomotion has low oscillation, it implies good stabilization. The remaining distance represents the speed of the robot walking. If the robot has a high value in the remaining distance, the robot has a low speed in walking. Our other goal is to realize a locomotion pattern with maximum possible speed. Therefore, our objectives are to acquire locomotion that has a good stabilization and a high walking speed.

In Eq. (5.22), $\dot{\beta}_{pitch}$ and $\dot{\beta}_{roll}$ are tilt oscillation in pitch and roll direction that has absolute value. In Eq. (5.23), $\ell(t,w_{ij})$ is the resultant value of $\mathrm{x}(t)$ and $\mathrm{y}(t)$ in each time sampling. Parameter Υ represents the value of desired walking speed. The $\ell(t,w_{ij})$, $\mathrm{x}(t)$, $\mathrm{y}(t)$ notations were defined as real numbers. Parameter w_{ij} is synaptic weight explained in Section 5.3.1. Parameters α_1 and α_2 represent the weight factor of fitness. The goal of this optimization problem is to minimize the fitness described in Eq. (5.24).

$$\bar{\dot{\beta}} = \frac{1}{T}\sum_{t=0}^{T}\left(\beta_{pitch}(t) + \beta_{roll}(t)\right) \tag{5.22}$$

$$\bar{\upsilon}_{(w_{ij})} = \Upsilon - \frac{1}{T}\sum_{t=0}^{T}\frac{\delta}{\delta t}\ell\Big(t, w_{ij}, \mathrm{x}(t), \mathrm{y}(t)\Big) \tag{5.23}$$

$$f = \bar{\dot{\beta}}\alpha_1 + \bar{\upsilon}\alpha_2 \tag{5.24}$$

This evaluation is conducted in computer simulation using Open Dynamics Engine (ODE) [3]. The evaluation time requires 10 seconds or 1000 time samplings since one time sampling requires 0.01 second. The timing process in the evaluation is the same as the real timing that is applied in the real robot.

5.3.4 Experimental Results in Biped Robot

In the experimental result, I used computer simulation and a small biped robot for implementing the proposed algorithm. There are two experiments which are required to conduct, the locomotion optimization and its application in the small biped robot.

5.3.4.1 Locomotion Optimization

The experiment is conducted on flat terrain and on rough terrain by using the proposed neural evolving algorithm, whose parameters can be seen in Table 5.1. I evaluate the walking performance based on two aspects which are the desired speed and the stability. The $1st$ experiment was conducted on a flat terrain. The $2nd$ experiment was conducted on a rough terrain. The $3th$ and $4th$ experiments were performed on a slope terrain with 10° and 16°, respectively.

Table 5.1: *Bacterial Programming Parameters for BIped Robot*

Parameter	Value
$N_{ind}, N_{gen}, N_{clones}, N_{inf}$	100, 50, 10, 30
$n^{min}; n^{max}$	0; 4
$N^{joint}_{min}; N^{joint}_{max}$	1; 3
$N^{neuron}_{min}; N^{neuron}_{max}$	1; 4
$\alpha_1; \alpha_2$	0.75; 0.25

In the optimization process, I set the friction to a certain value in Open Dynamics Engine. The neural oscillator's parameters τ and τ' are 0; τ_f was set as 12.0; 1.2; 1.0 respectively, and I also set b as 2.5 and u_0 as 1. In this experiment, the time cycle is 0.01 second.

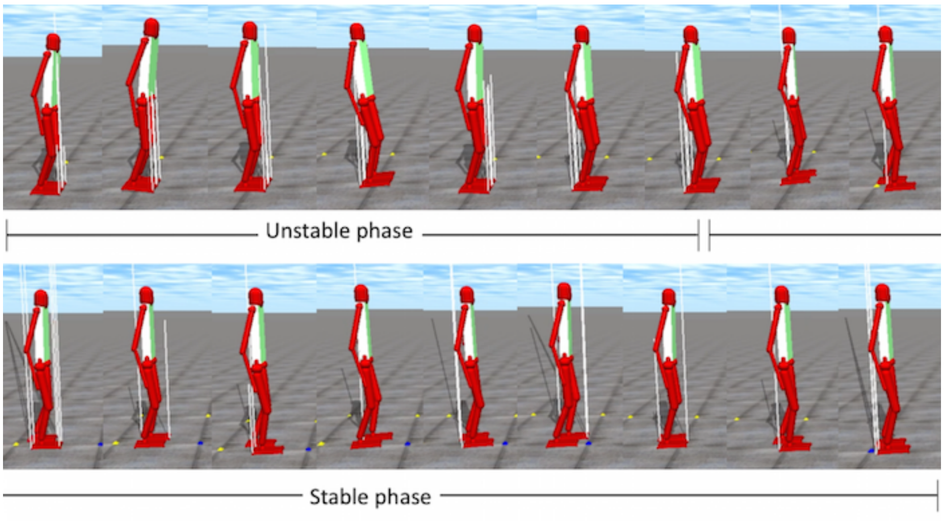


Figure 5.21: *Simulation of the locomotion on a flat terrain (sample experiment 1)*

Experiment on a flat terrain (Exp. 1) In this experiment, I set the terrain without any object or obstacle. The desired walking speed Υ was set as 1.0 m/sec. The sample walking simulation can be seen in Fig. 5.21. The optimized tree structure and its neuron interconnection structure can be seen in Fig. 5.24a. In this experiment, there are 2 motor neurons required in hip-x joint, 1 motor neuron in knee joint and 1 motor neuron in ankle joint.

Experiment on a rough terrain (Exp. 2) On rough terrain, I put some obstacles with various sizes on the terrain. The sample walking simulation can be seen in Fig. 5.22. The desired walking speed Υ was set as 1.0 m/sec. There are 2 motor neurons required in hip-x joint and 1 motor neuron in knee joint in this experiment. The optimized tree structure and its neuron interconnection structure can be seen in Fig. 5.24b. The experiment on rough terrain shows the effectiveness of sensory-motor coordination.

Experiment on slope terrains (Exp. 3 and 4) Furthermore, I train the robot on different slope terrains with 10° and 16°, respectively. Their simulations can be seen in Fig. 5.23. The experiments on slope terrains also show the effectiveness of sensory-motor coordination. In these experiments, I ran the optimization process until the 50^{th} generation. On slope terrain experiments, I can see the activity of sensory-motor coordination which is depicted in Fig. 5.25. The ground touch sensor input gave some effects to the motor neurons, therefore in Fig. 5.25, the signal is changing adaptively. There are only 6 motor neurons required for performing on the trained terrains, 1 motor neuron representing hip-x joint and 2 motor neurons representing knee joint.

The optimized tree structure and its neuron interconnection structure can be seen in

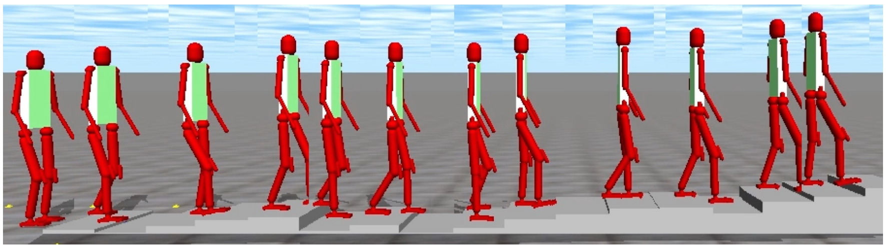
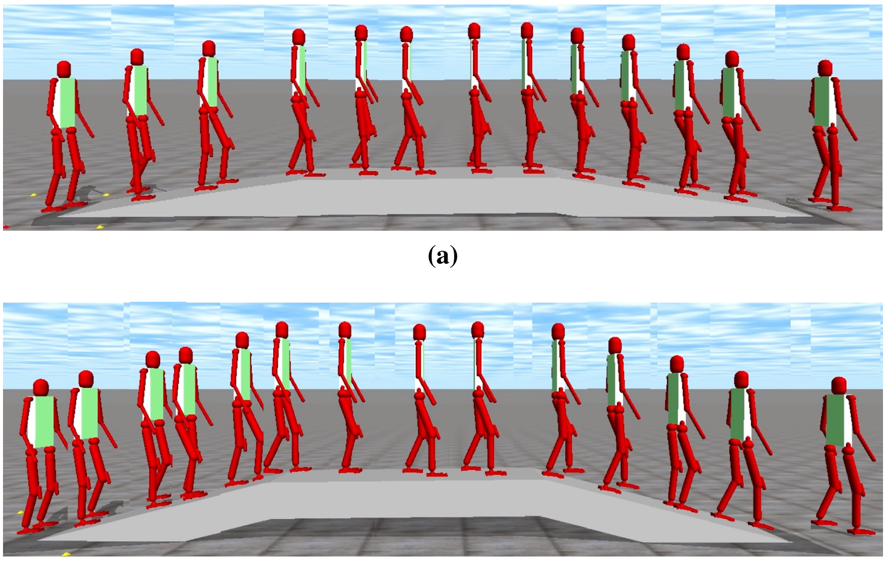


Figure 5.22: *Simulation of the locomotion on an uneven terrain (sample experiment 2)*



(a)

(b)

Figure 5.23: *Simulation of the locomotion on slope terrains (a) low slope terrain (sample experiment 3) (b) high slope terrain (sample experiment 4)*

Figs. 5.24c and 5.24d. The desired walking speed Υ was set as $1.0\ m/sec$. In experiment 3, there are 1 motor neuron required in hip-x joint and 2 motor neurons in knee joint. In experiment 4, there are 1 motor neuron required in hip-x joint and 2 motor neurons in knee joint. Since they have similar environmental conditions, experiments 3 and 4 have similar neuron interconnection structure which are depicted in Figs. 5.24c and 5.24d.

Comparing with existing models, the locomotion proposed in [197] enables walking on a slope surface with 11° tilt angle with horizontal direction and the method in [162] enables walking on a slope surface with 10° tilt angle with horizontal direction, our proposed locomotion can be implemented on a slope surface with 16° tilt angle with horizontal direction.

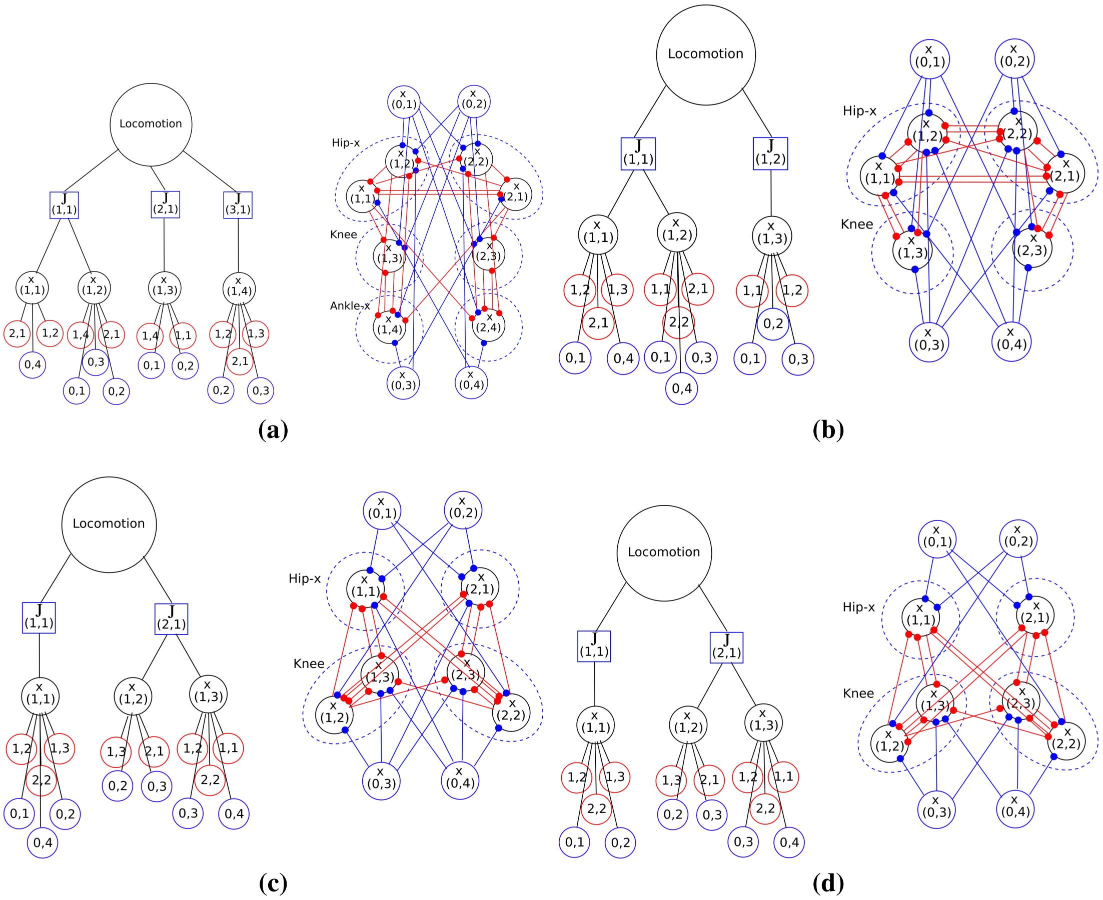


Figure 5.24: *Optimized tree structure and neuron interconnection in every sample experiment (a) Experiment 1 (b) Experiment 2 (c) Experiment 3 (d) Experiment 4*

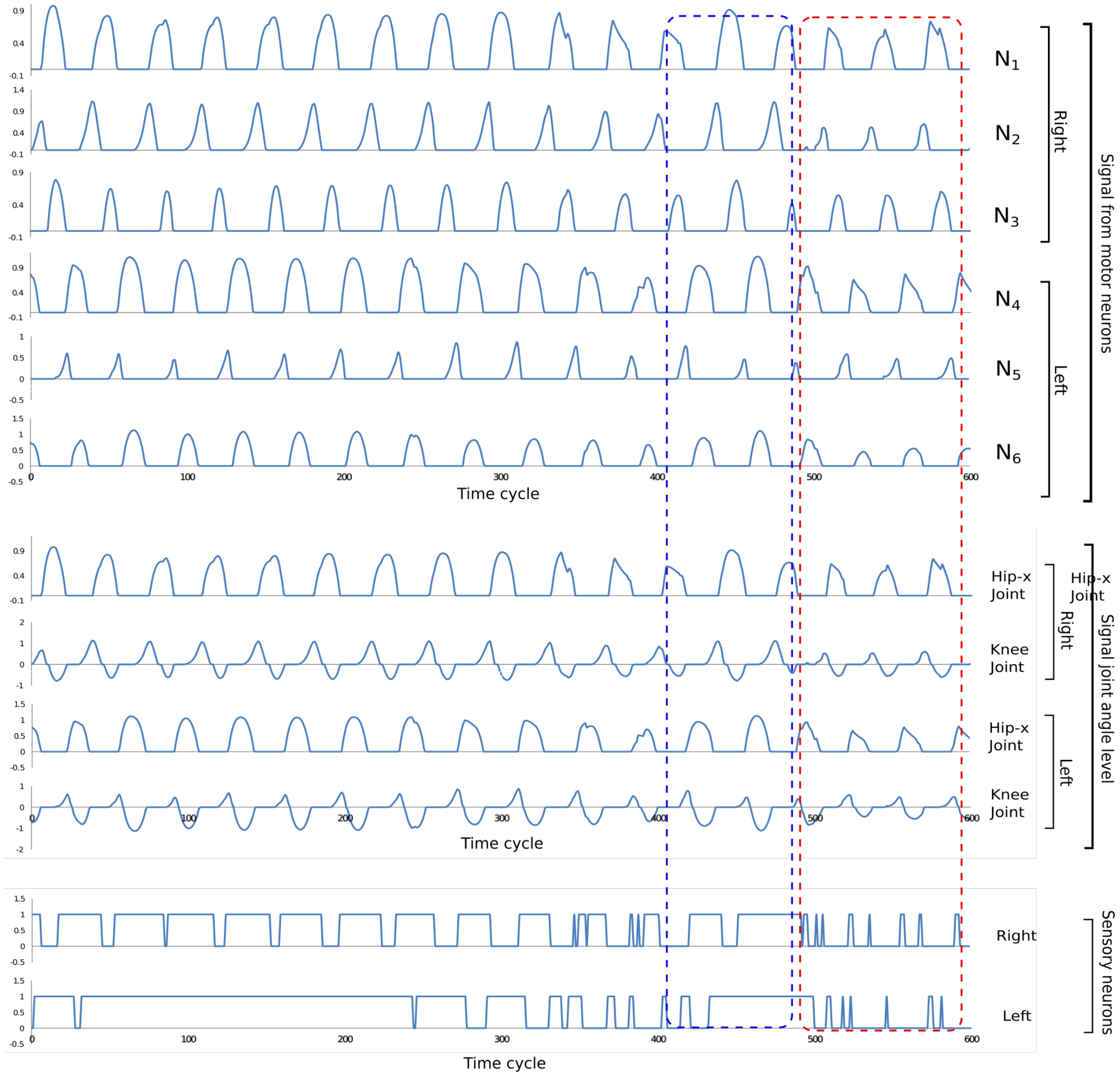


Figure 5.25: *The signal generated in the experiment on rough terrain (experiment 4). The effect of sensory input can be seen to the motor neuron signal. Blue and red blocks show the difference of signal patterns because of the different sensory inputs.*

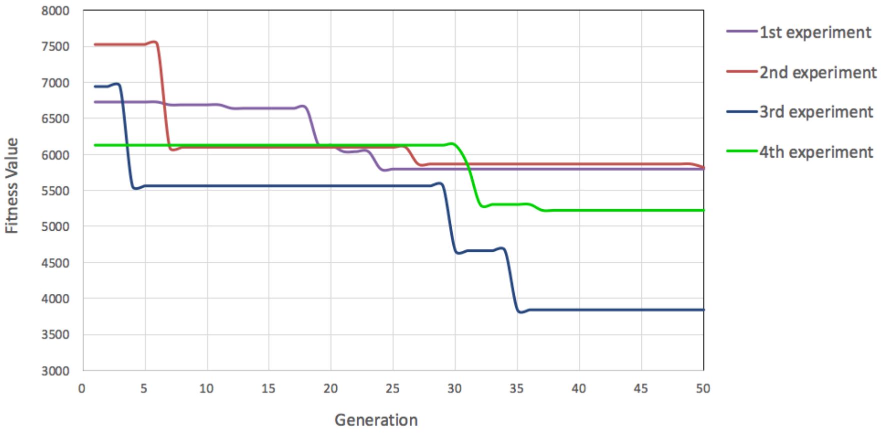


Figure 5.26: *Fitness evolution, 1st experiment for flat terrain, 2nd experiment for uneven terrain, 3rd experiment for 1st slope terrain, and 4th experiment for 2nd slope terrain*

5.3.4.2 Performance Evaluation

Optimization performance The optimization model successfully optimized the structure of neural locomotion in all terrains. The fitness evolutions can be seen in Fig. 5.26. The proposed algorithm succeeded to form the neuron structure for locomotion on the flat terrain. The walking speed performance is depicted in Fig. 5.30. The robot walking speed was able to approach the desired walking speed. In Fig. 5.27, the evolution of joints and neuron numbers in each joint are shown. Those numbers are adaptively changed in some generations. In sample experiment 1, the generation started using 2 joints in each leg. After several generations, the 2 joints in each leg were not stable enough for locomotion. Then, the number of joints was increased and became 3 joints in each leg. Likewise in the evolution of neuron numbers in each joint, the number of neurons is dynamically changed until the most effective number is achieved. In the final structure, there are 2 neurons in the first joint (hip-x joint), 1 neuron in the second joint (knee joint), and 1 neuron in the third joint (ankle-x joint). Therefore, in this experiment, the locomotion model required 8 neurons for 2 legs, where their signal in neuron level and joint angle level can be seen in Fig. 5.28. This model could not reduce the number of neurons in hip-x like the number of neurons in the other joints. Nevertheless, from the result I can understand which are the important joints in the biped robot locomotion. Like in human, these joints are required to produce rhythmic signal for walking; at least 2 neurons are required for representing extensor and flexor muscle. The hip-x joint is important for swinging the leg and control the walking phase. Furthermore, human is still able to walk even though ankle joints or knee joints are disabled.

In our previous research [245], in order to form the stable locomotion pattern, two neurons were required in each joint, but in this proposed research, the automatic evolution is

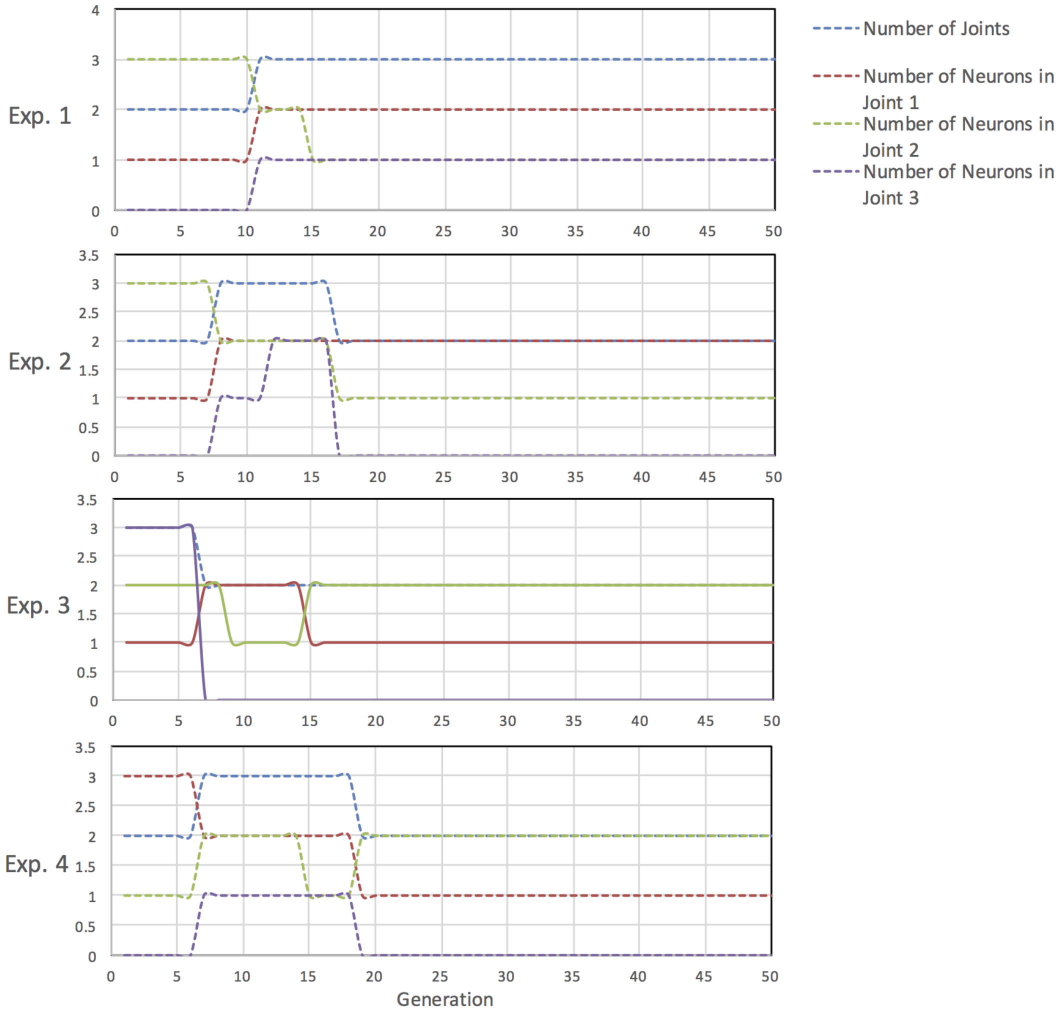
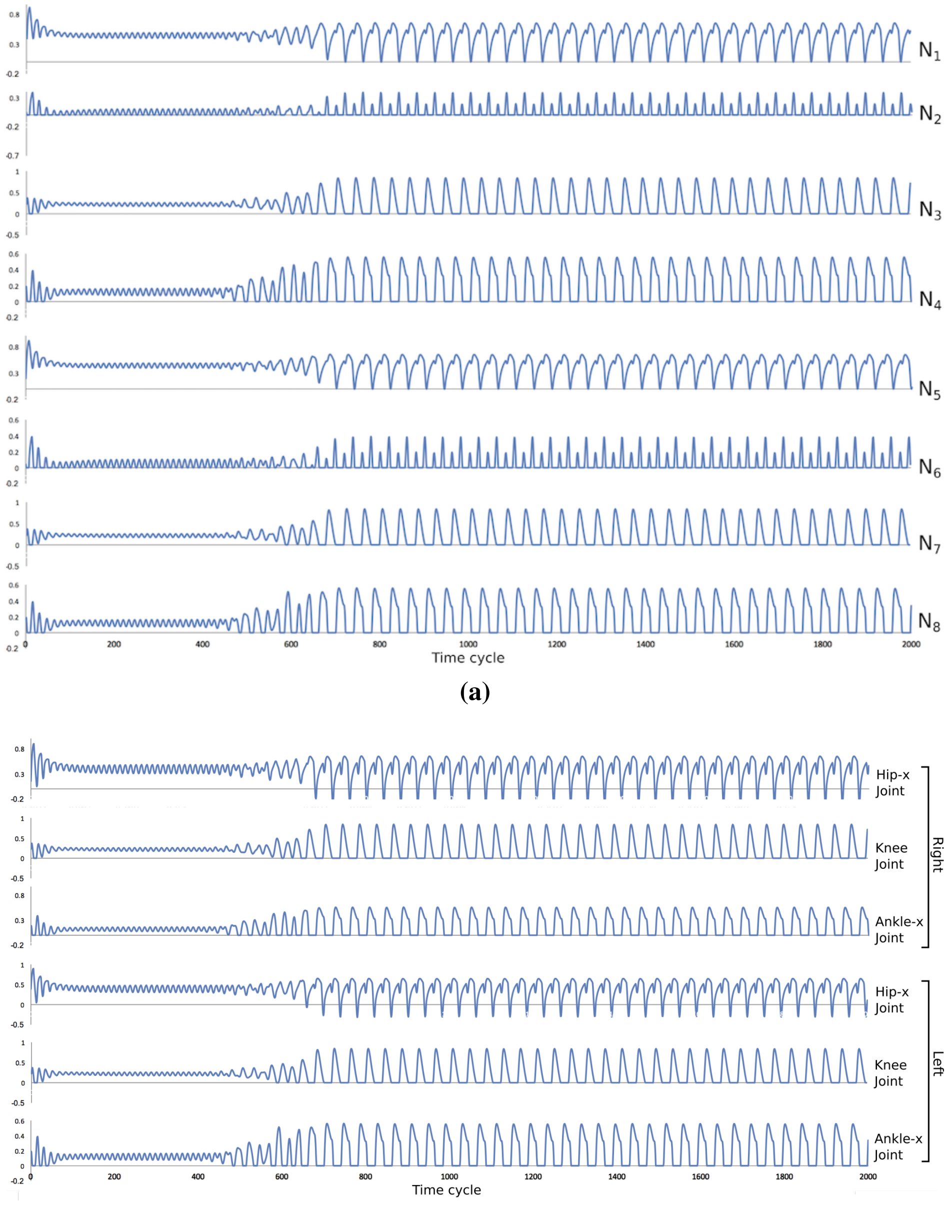


Figure 5.27: *The number of joints and the number of neurons in each joint in every generation*



(a)

(b)

Figure 5.28: *Signal output (a) in neuron level (b) in joint angle level*

applied and the number of neurons in each joint can be decreased to one neuron in each joint. Based on the walking analysis, the proposed model has similar performance as the previous model. Therefore, this proposed algorithm can reduce the computational cost.

Comparing with other similar works, while the models in [176, 279, 189, 197, 19] use 2 neurons in every joint, the proposed model may use only one neuron in every joint. It implies that the proposed model has a lower computational cost.

Tree-structure evaluation Figure 5.24 shows the comparison between tree-structure based and common based neural locomotion. From the visual evaluation it can be seen, that the proposed tree-structure is simpler than the common structure. It can easily cluster and show that one neuron is affected by several sensory and motor neurons. In the synaptic weight optimization cost, since the proposed tree-structure uses a mirror structure, only $(N/2)^2 - (N/2)$ synaptic weights are required to be optimized. The common structure, however, required $N^2 - N$ synaptic weights to be optimized, where N is the number of neurons in total. It implies that by using the proposed tree-structure, the complexity can be reduced.

Stability performance In order to prove the effectiveness of the proposed model, I evaluate the walking performance. I analyze the interaction between sensory-motor neuron and the effect of the sensory neuron to the signal generation. I recorded the signal generated by neuron and joint angle level.

In order to determine stable and unstable phase, I evaluate using a degree of oscillation amplitude of robots body tilt angle and Cobweb diagram. If the graph is not approaching the center point, the robot is unstable. If the graph is approaching and encircling the center point, the robot is stable. I recorded the body tilt sensor information which can be seen in Fig. 5.29 and analyzed using Poincare map and Cobweb map illustrated in Figs. 5.31a and 5.31b. These figures recorded the point value of tilt angle of the robot body, where the angular velocity is zero and the previous angular velocity is greater than zero. In Fig. 5.31a, the initial condition of tilt is 0 rad; it has unstable phase in the early second after that it gradually changed to a stable tilt angle of 0.02 rad.

5.3.4.3 Implementation in Real Robot

In this implementation, I use the optimized neural structure and run it in the applied robot. I use a small biped robot with 55 cm of height and 5 kg of weight, where the mechanical structure of the robot is the same as that of the simulated robot. However, the size was not the same. The detailed robot specification can be found in our previous researches [257]

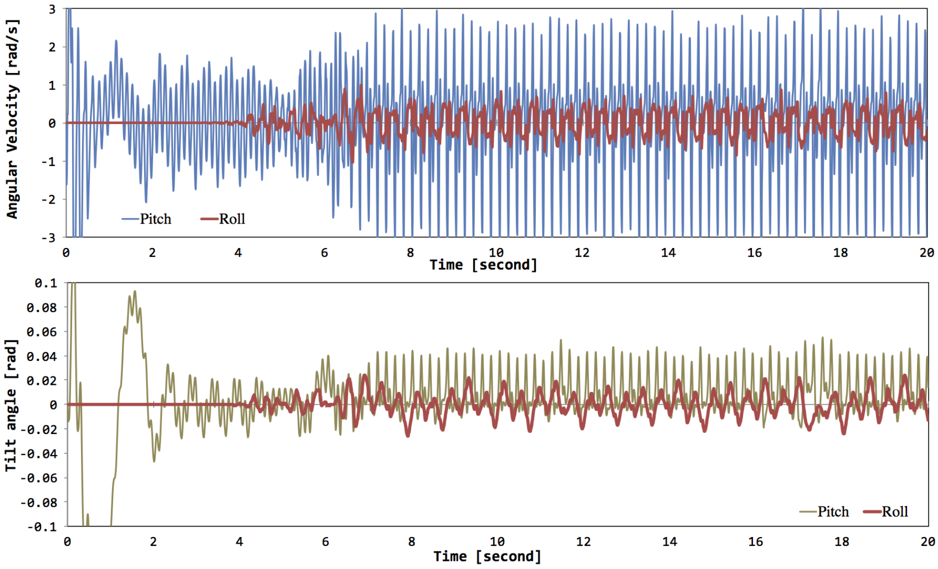


Figure 5.29: *The signal oscillation of angular velocity and tilt angle in pitch and roll direction measured from robot's body. The robot is not stable during the first 6 seconds and getting stable after that.*

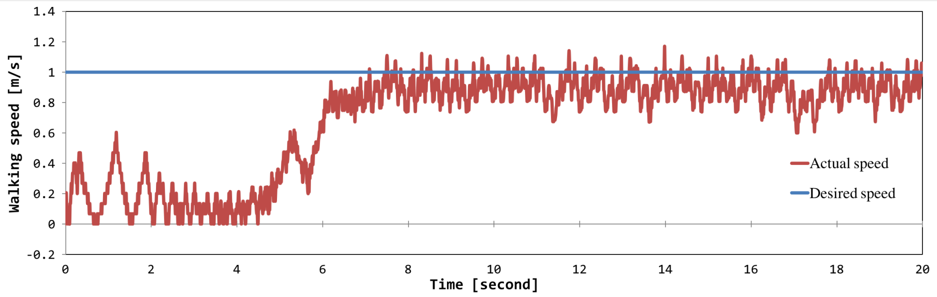
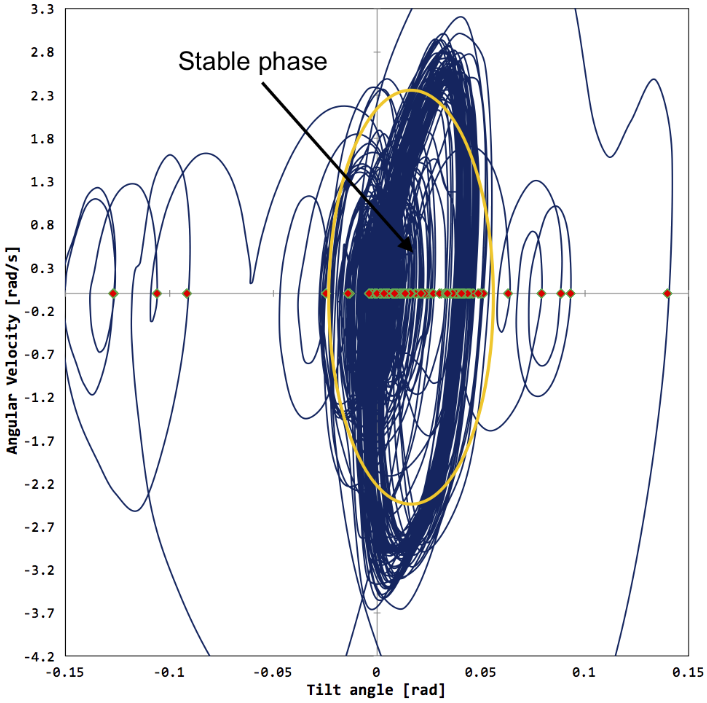
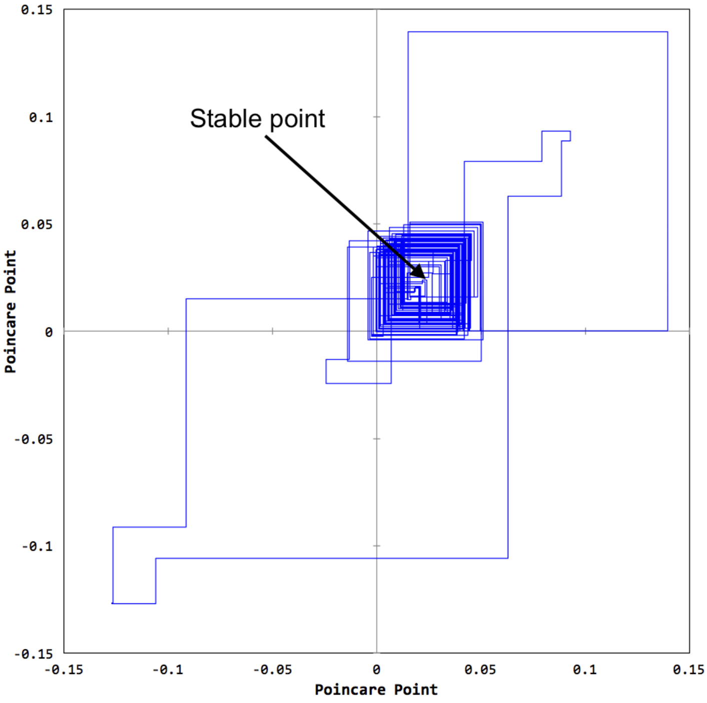


Figure 5.30: *The sample of recorded walking speed. The robot speed is approaching the desired speed after 6 second*

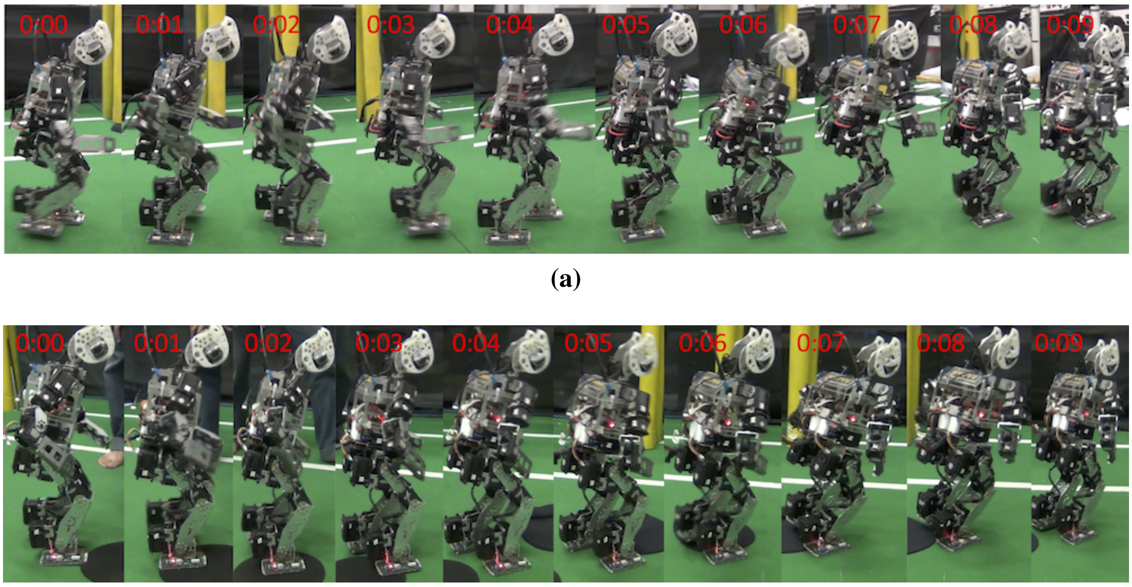


(a)



(b)

Figure 5.31: *Stability analysis diagram (a) Phase diagram of robot tilt angle and stability analysis, based on Poincare map (b) Cobweb diagram representation of Fig. 5.31a*



(a)

(b)

Figure 5.32: *Experimental result in a real biped robot. The robot was tested (a) on flat terrain (b) on small uneven obstacle.*

and [251]. The robot has been equipped with a tilt sensor and ground touch sensors (4 touch sensors in every sole of feet). Since the robot representation is smaller than in the simulation, its desired walking speed (Υ) was set only as 30 cm/sec. In order to prove the stability and the coordination of sensory-motor neuron, I used an uneven obstacle with 7 mm of height.

The proposed model was successfully applied, and the robot was able to walk on both flat terrain and small uneven obstacle. The sample figure of the implementation is depicted in Fig. 5.32.

5.3.5 Experimental Results in Quadruped Robot

I validated the proposed method though several computer simulation and real robot implementation. Two experiments were conducted to validate the locomotion optimization and its application in the middle size quadruped robot.

5.3.5.1 Locomotion Optimization

The experiment was conducted on a rough terrain with different slope degrees by using the proposed neural evolving algorithm, whose parameter values have been tabulated in Table 5.2. We then evaluate the walking performance based on the two aspects which are the speed and the stability.

Table 5.2: *Bacterial Programming Parameters for Quadruped Robot*

Parameter	Value
$N_{ind}, N_{gen}, N_{clones}, N_{inf}$	100, 50, 10, 30
$p^{min}; p^{max}$	0; 4
$N^{joint}_{min}; N^{joint}_{max}$	1; 3
$N^{neuron}_{min}; N^{neuron}_{max}$	1; 4
$\alpha_1; \alpha_2$	0.75; 0.25

For simulation, we first set the friction to a certain value in Open Dynamics Engine [3]. The sample walking simulation can be seen in Fig. 5.33. The proposed algorithm succeeded to form the neuron structure for locomotion on the uneven terrain.

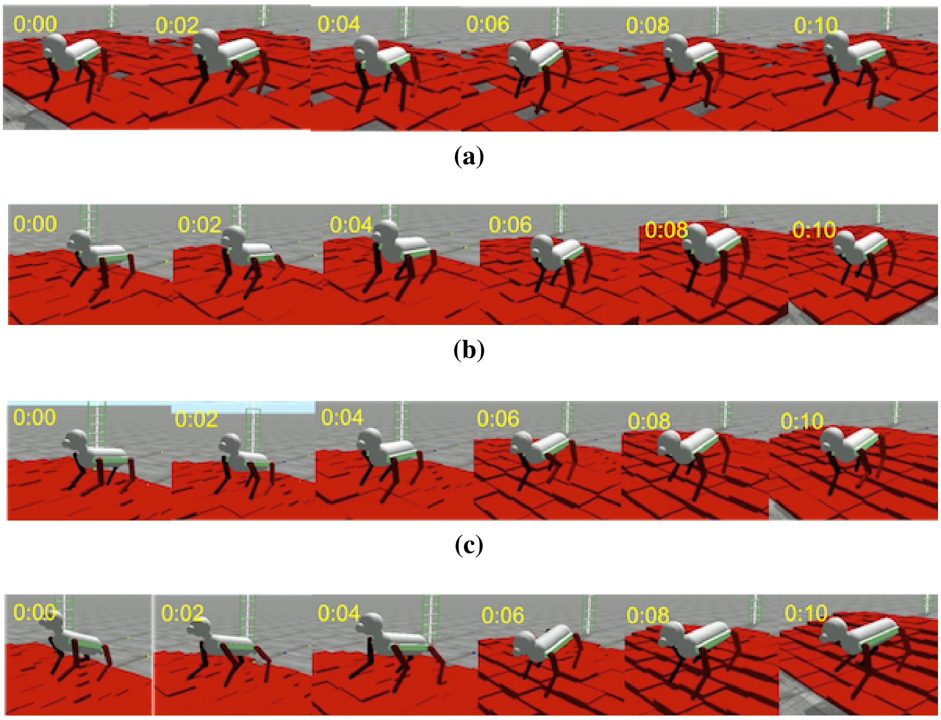
The evolution of joints and neuron numbers in each joint are shown in Fig. 5.34. Those numbers are adaptively changed in some generations. In flat terrain experiment, after several generations, 3 joints in each leg were not stable enough for locomotion. Then, the number of joints was decreased and became 2 joints in each leg. Thus, in the final structure, there are 2 neurons in the first joint (hip-x joint) and 2 neuron in the second joint (knee joint). In this experiment, the locomotion model required 16 neurons for 4 legs, where their signal in joint angle level can be seen in Fig. 5.35.

The results showed the important joints in the robot locomotion. Like animal, these joints are required to produce rhythmic signal for walking; at least 2 neurons are required for representing extensor and flexor muscle. The hip-x joint is important for swinging the leg and controlling the walking phase.

In the rough terrain experiment, we can see the activity of sensory-motor coordination as shown in Fig. 5.37. The ground touch sensor input gave some effects to the motor neurons, therefore the signal is changing adaptively as shown in Fig. 5.37. The figure shows the signal between left part and right part of the robot legs.

The optimized tree structure and its neuron interconnection structure can be seen in Fig. 5.36. In the rough terrain, 2 joints in every leg was not stable enough. Therefore, after a few generations, the number of joints became 3 joints in every leg. In this experiments, there are 16 motor neurons required for performing on the trained terrains, 1 motor neuron representing hip-x joint and hip-y joint, and 2 motor neurons representing knee joint.

In order to evaluate the stability level, we recorded the body tilt sensor information which can be seen in Fig. 5.38 and analyzed it using Poincare map and Cobweb map which are illustrated in Fig. 5.39.



(a)

(b)

(c)

(d)

Figure 5.33: *Simulation of proposed locomotion on uneven terrain in different degree of slope (a) slope* 0° *(b) slope* 5° *(c) slope* 14° *(d) slope* 20°

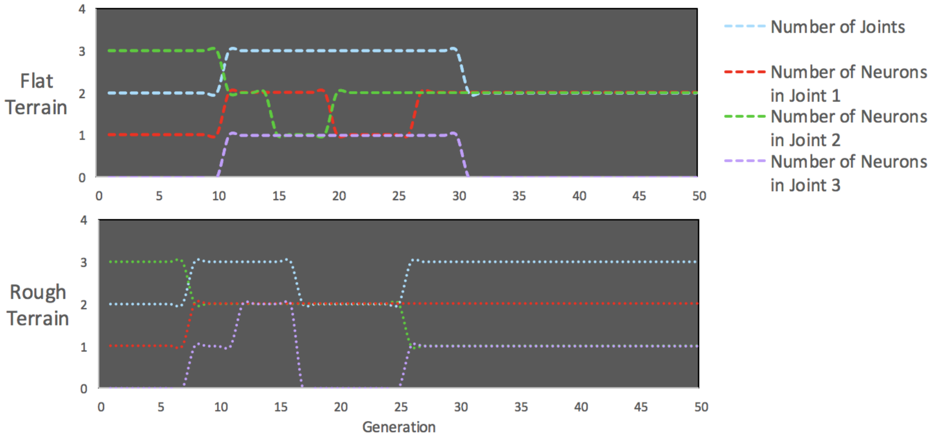


Figure 5.34: *The number of joints and the number of neurons in each joint in every generation*

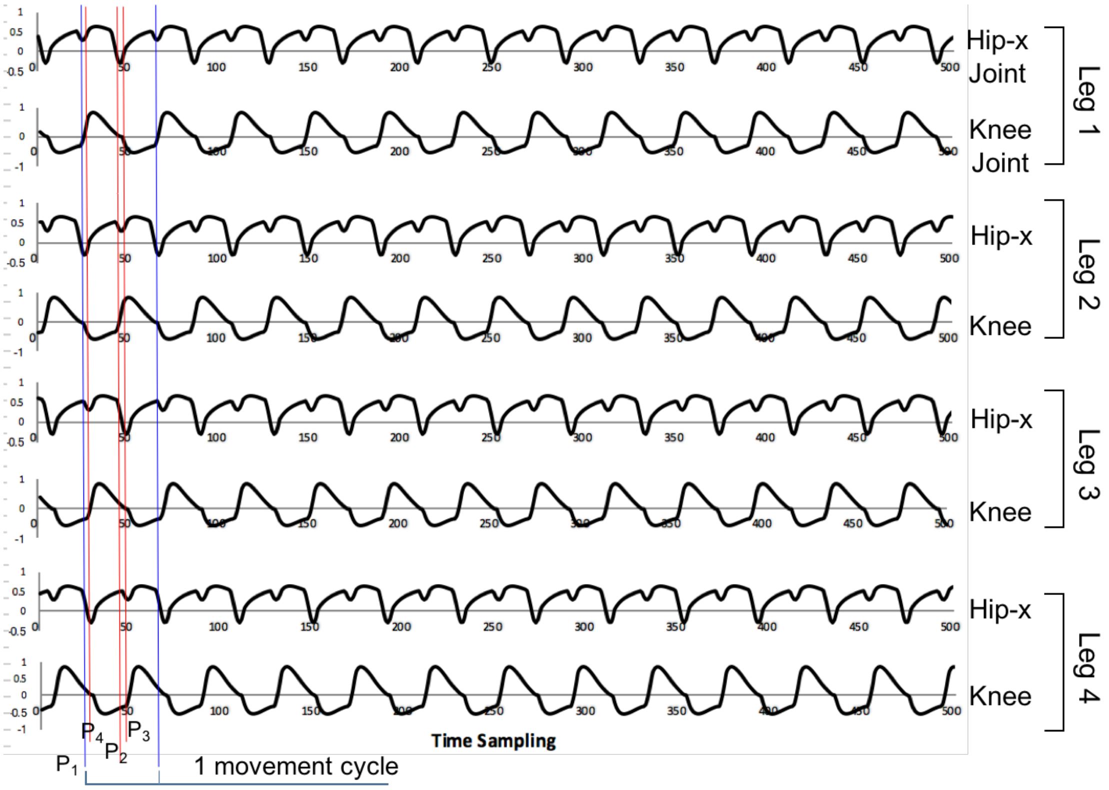


Figure 5.35: *Sample of signal output in joint angle level ($\Theta_i^{(l,r)} = \sum_{j=1}^{N_i^{neuron}} J_j^{l,r} y_j$) on the flat terrain. The signal movement pattern tends to the walk-like movement. Signal in first (P_1) and third (P_3) leg has the same phase, and the second (P_2) and the fourth (P_4) has almost the same phase.*

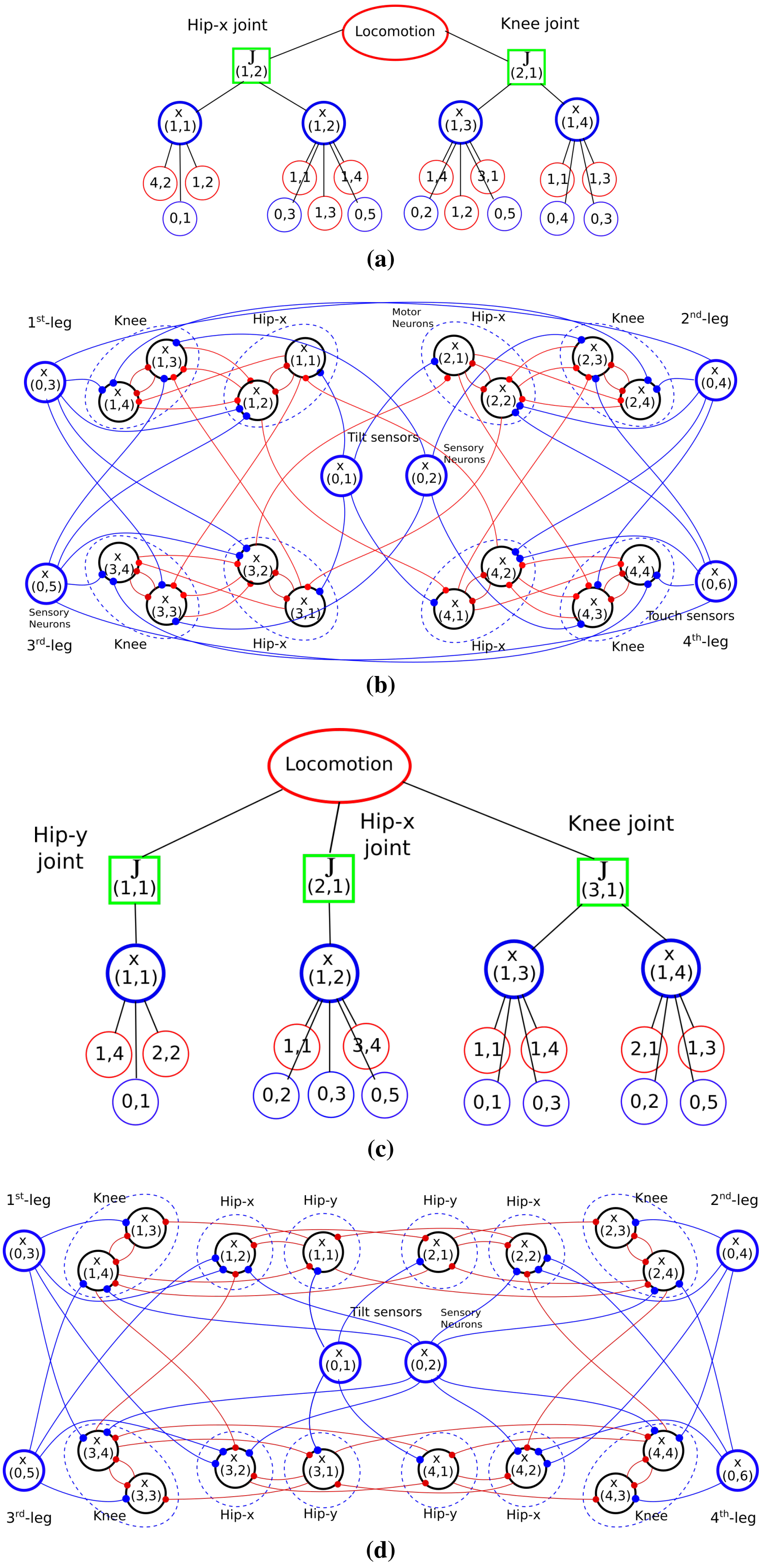


Figure 5.36: *(a) Optimized tree structure for flat terrain (b) Optimized neuron interconnection (c) Optimized tree structure for rough terrain (d) Optimized neuron interconnection*

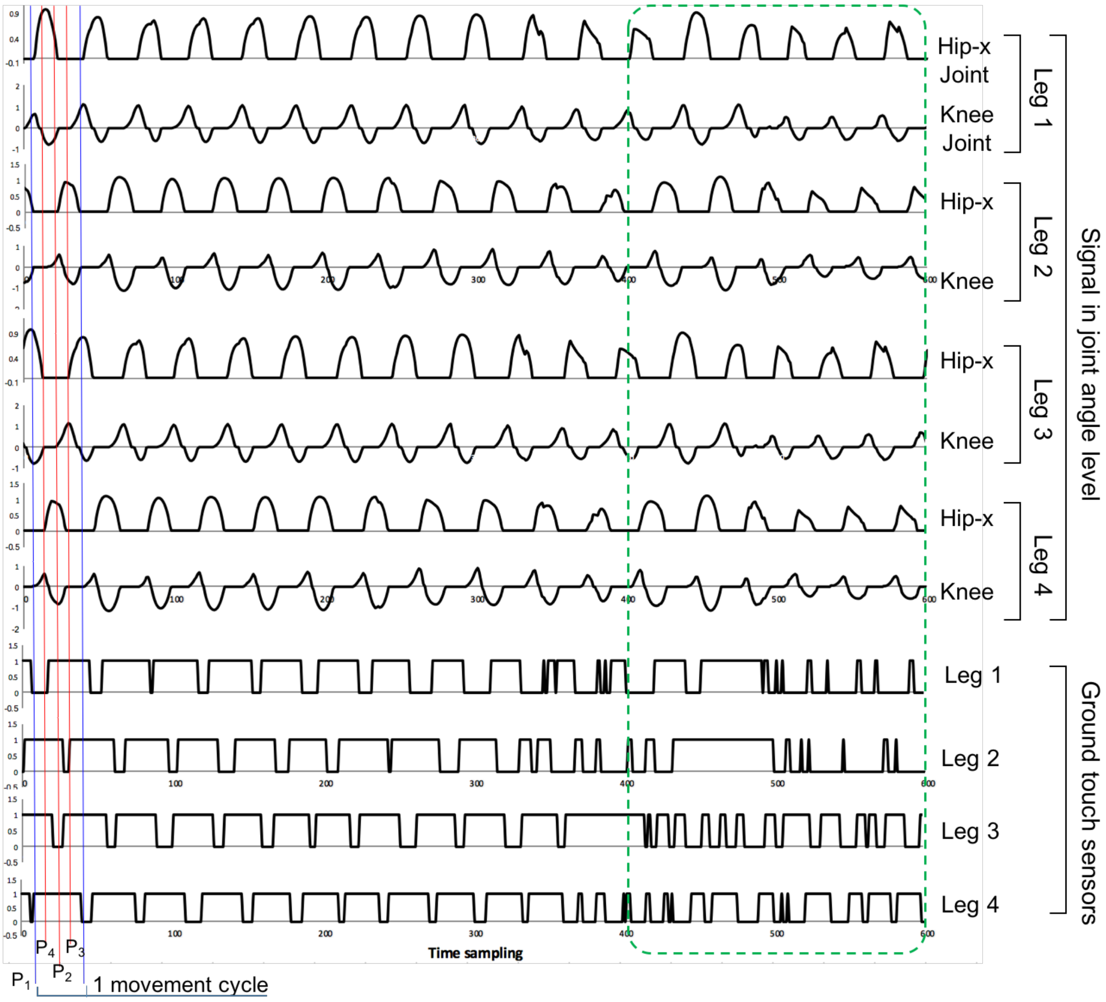
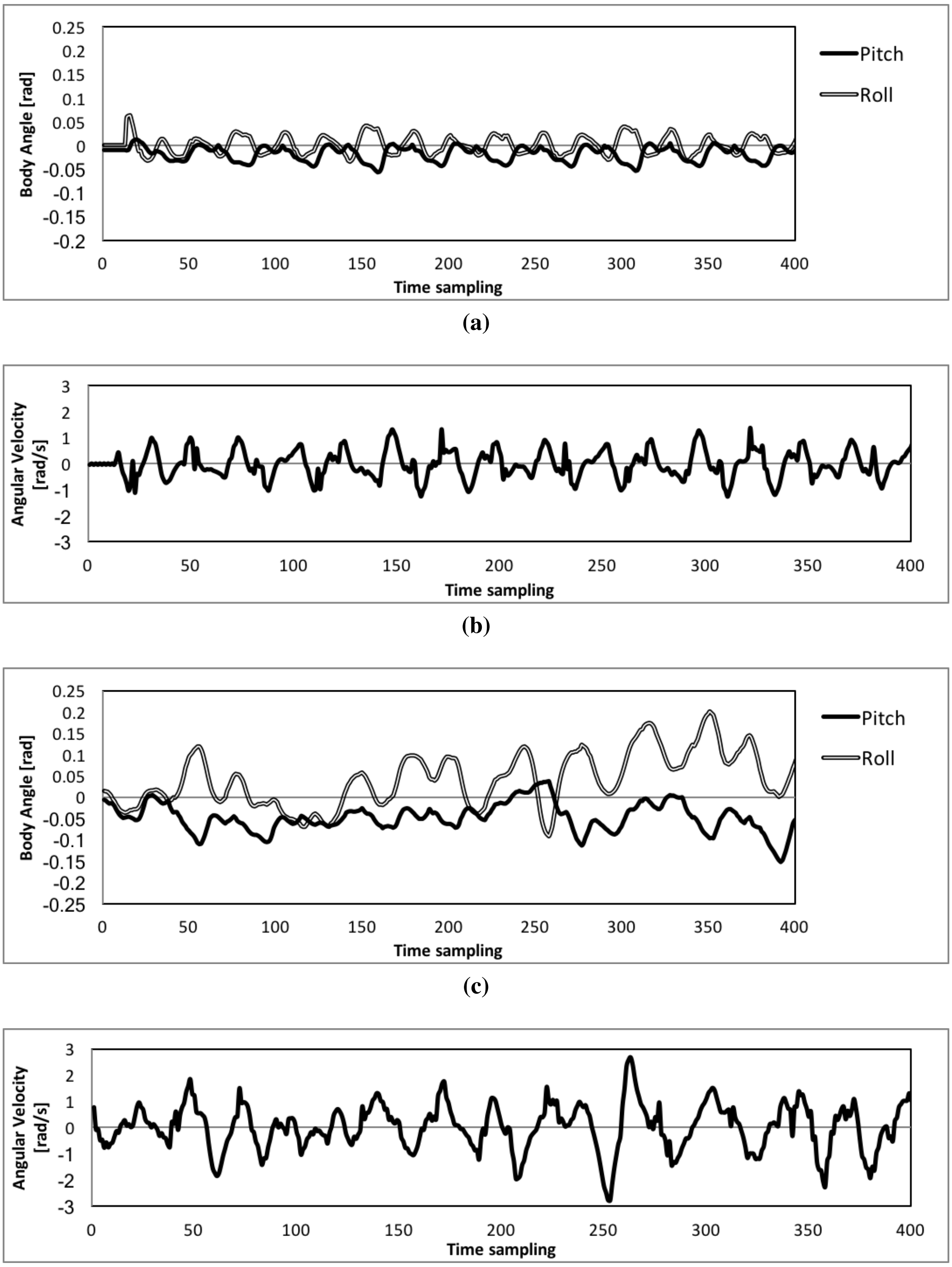


Figure 5.37: *The signal generated in the experiment on rough terrain. The effect of sensory input can be seen to the motor neuron signal. Green blocks show the difference of signal patterns because of the different sensory inputs. The signal phase in every leg ($P_1, \cdots, P_4$) has 0.25 phase difference that makes the movement slower than on the flat terrain.*



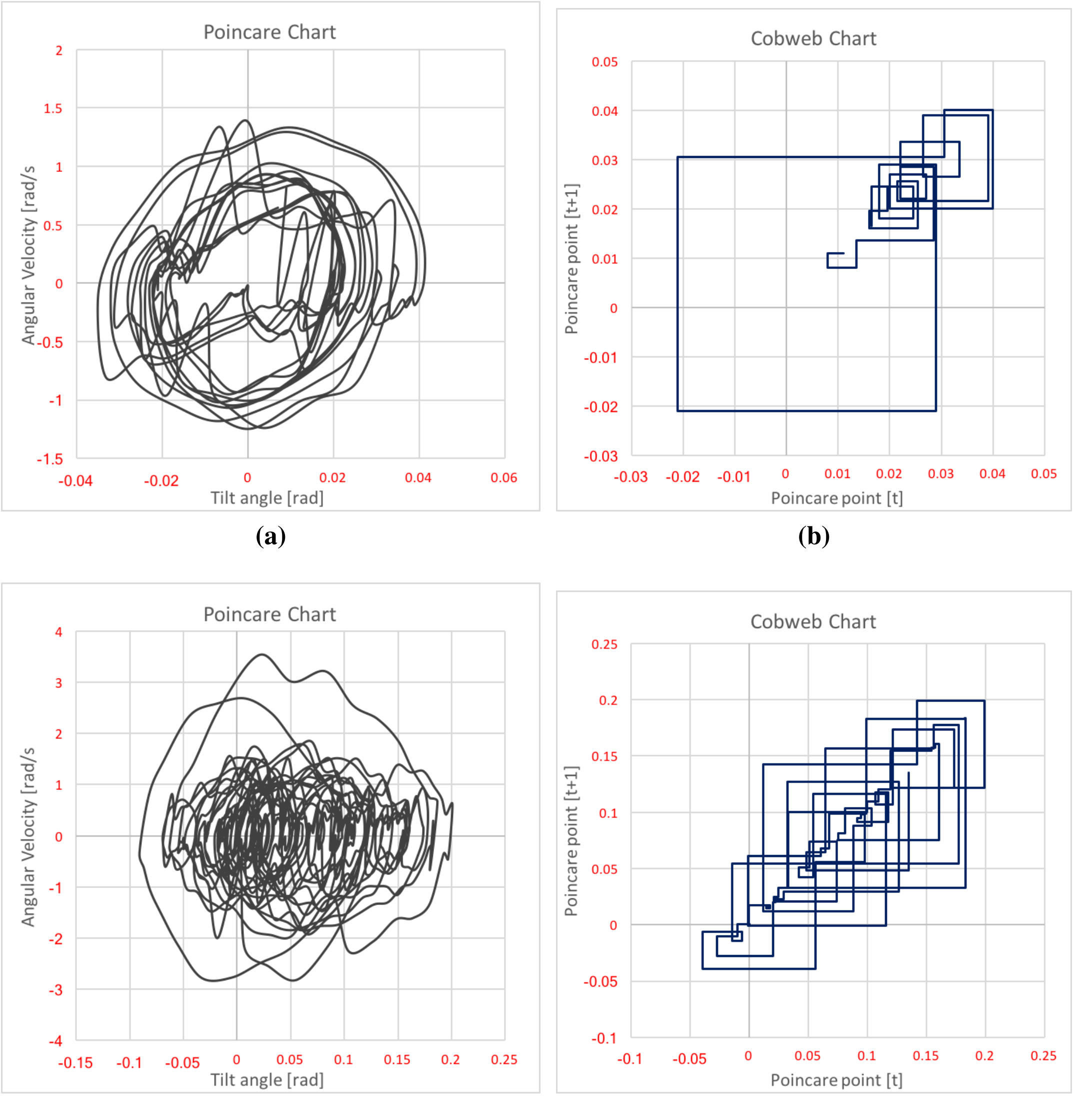
(a)

(b)

(c)

(d)

Figure 5.38: *The signal oscillation of angular velocity and tilt angle in pitch and roll direction measured from robot's body. (a) Body tilt signal on flat terrain. (b) Angular velocity signal on flat terrain. (c) Body tilt signal on rough terrain (outdoor grass). (d) Angular velocity signal on rough terrain. The signal on flat terrain is more stable than on rough terrain. Nevertheless, the signal oscillation is still acceptable for stability.*



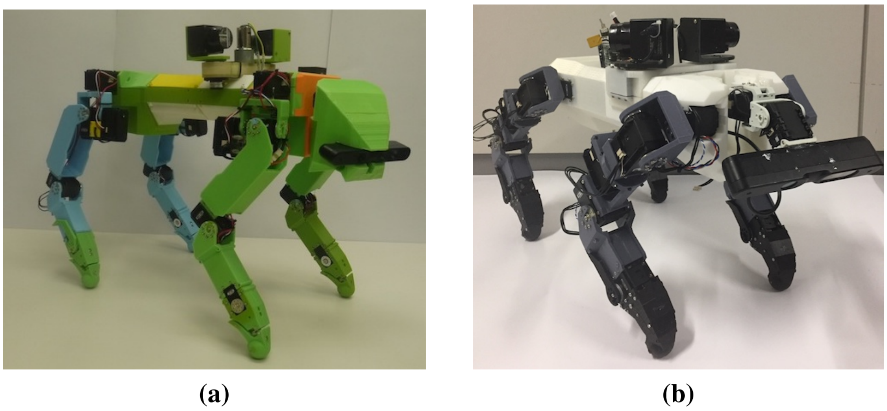
(a)

(b)

(c)

(d)

Figure 5.39: *Stability analysis diagram (a) Phase diagram of robot tilt angle and stability analysis on the flat terrain, based on Poincare map (b) Cobweb diagram representation of Fig. 5.39a (c) Poincare map during robot's performance on the rough terrain (d) Cobweb diagram representation of Fig. 5.39c*



(a)

(b)

Figure 5.40: *Proposed middle size quadruped robot (a) first prototype (b) second prototype.*

5.3.5.2 Implementation in Real Robot

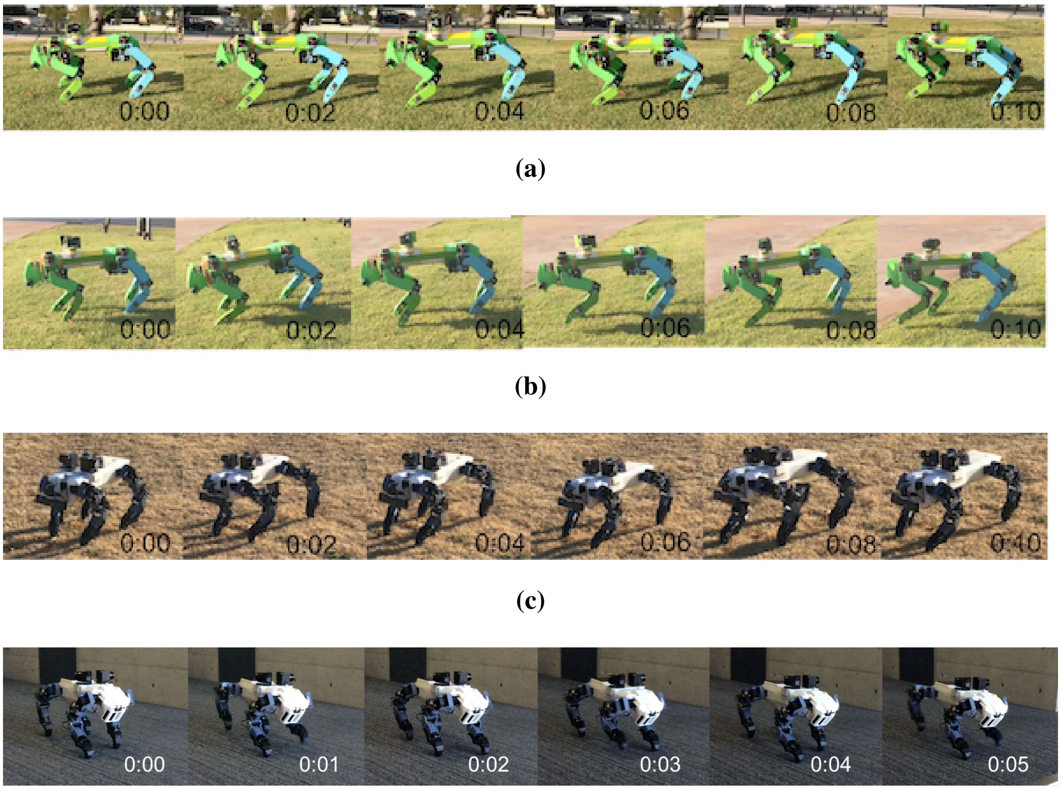
The proposed method was further validated by real robot implementation. We utilized the optimized neural structure and uploaded it to the robot. We built a quadruped robot with 55 cm of height and 6 kg of weight, where the mechanical structure of the robot is the same as the robot in simulation experiment. In this experiment we have built 2 middle size quadruped robots, where the first robot is shown in Fig. 5.40a and the second one is shown in Fig. 5.40b. The proposed model is successfully applied to the real robots, and both robots are able to walk on both flat terrain and small uneven obstacles. The sample figure of the implementation is depicted in Fig. 5.41. The detail of robot design can be seen in Appendix B

5.3.6 Discussion of Proposed Learning and Optimization model

This section proposed the module for evolving the sensory-motor integration in a neural based legged robot locomotion. Since the sensory-motor neuron interconnection is getting more complex, a new tree structure based model of neuron interconnection structure was proposed. Bacterial programming is used as the optimization technique. Based on the optimization, the proposed tree structure can simplify the process and successfully generate stable walking in biped robot and quadruped robot as well.

In the evolutionary process of neurons, the number of motor neurons in every joint was successfully optimized. Based on the result, one neuron is enough for generating signal in some joints in the experiments. This evolving model can reduce the number of neurons involved depending on the requirement of the robot's performance.

In order to represent the sensory-motor coordination, this proposed model can also gen-



(a)

(b)

(c)

(d)

Figure 5.41: *Experimental result in a real robot. (a) First prototype of robot on the grass. (b) First prototype of robot on the grass with slope. (c) Second prototype of robot on the rough grass (d) Second prototype of robot on the flat terrain.*

erate the sensory-motor neuron interconnection. There are 4 sensory neurons representing 4 pieces of internal sensory information which are tilt sensor in pitch and roll, ground sensor at left and right. Based on the experimental result, the signal generated by motor neurons is adaptively changed depending on the sensory condition.

Based on the result, the proposed bacterial programming based tree structure model can be applied for simplifying the representation of sensory-motor interconnection in any multi-legged robots, it is not limited to biped and quadruped robots.

Overall, based on the validation by the experiments, the contributions of the section are listed below:

- I used a tree structure based optimization strategy which can simplify the representation of the sensory-motor interconnection structure model.
- I proposed a novel Bacterial Programming for optimizing the complex structure of the neural based locomotion model.
- Other novelties are regarding the initial population generation and the gene transfer operation in BP.

5.4 Implementation in for challenging action (Ladder Climbing)

In the disaster area, a robot is a helpful important part to investigate inside a dangerous place such as vulnerable building, radioactive leak, or fire condition. Besides the robot's intelligence, it is also important for the robot to have motion capabilities. There is a lot of divers environmental condition will be faced by the robot in the disaster area, such as unstable terrain, rough terrain, kind of ladders, etc. Besides the terrain condition, ladders have a different challenge for some researchers. There are many types of the ladder, where each ladder has unique movement behavior. The robot involves a lot of constraints for performing the movement through the ladder. The robot has to keep the body in a stable moment, while another part of body grasping or contacting the ladder. This combinational motion has been summarized as an optimization problem by several researchers [100, 32].

The development of movement on the ladder has been conducted by several researcher [297, 323, 131, 333, 76, 278]. The vertical ladder is the most challenging in these current issues. Vaillant and Yoneda developed the climbing behavior on the vertical ladder in their biped robot [297, 323]. Zhang et al. also presented a planning strategy for the Hubo-II+ robot that automatically generates multi-limbed locomotion sequences that satisfy contact, collision and torque limit constraints for a ladder-climbing motion [333]. The climbing movement model has also been developed in the six-leg robot that proposed by Fujii et al. [76]. However, they did not consider the movement transition between horizontal and vertical movements. The locomotion model was performed separately.

Therefore, in this section, I will realize the capabilities of the quadruped robot that able to move from downstair to upstair through the vertical ladder. There are three states exist in this performance, horizontal movement in the lower stair, vertical movement on the ladder, and horizontal movement on the upper stair. Movement transition is an important part that should be required in order to realize the proposed movement capabilities.

Some of the transitional movement from horizontal, vertical ladder problem has been solved by some researcher [67, 131, 169, 167, 168, 98]. They built motion behavior of humanoid robots when climbing vertical ladders and transitioning between ladders and catwalks. They realized bidirectional transitioning from ladders to catwalks [131]. In the DARPA Challenge experiment, the DRC-Hubo humanoid robot successfully performed mounting, climbing, and dismounting an industrial ladder [169]. It succeeded transitioning from ground to upper stair (from horizontal to vertical). However, they did not solve the transition movement from vertical movement to horizontal movement in the upper stair. Their transitioning movement to the upper stair become their challenging problems (vertical to horizontal). Furthermore, it required handrail support for transitioning to the upper stair. However, not all ladder has handrail support. Realizing the transitional movement to upper stair from the vertical ladder without handrail was the new problem [7]. Thus, a unique behavior is required to pass this challenge.

In order to overcome that problem, I proposed a novel behavior that able to move from horizontal to vertical movement and transition to horizontal movement in the upper stair. I develop posture and swinging behavior that autonomously generated depending on the environmental condition. The posture behavior is developed mainly for performing a transitional movement from horizontal to vertical or vice versa. The swinging behavior is for generating foothold planning or grasping behavior. The posture and swinging behavior will be enlarged into several behaviors that are separately activated depending on the environmental condition.

5.4.1 Concept strategy

The illustration of the transition movement strategy on the vertical ladder without handrailing can be seen in Fig. 5.42. It shows the movement process from stand up position in lower

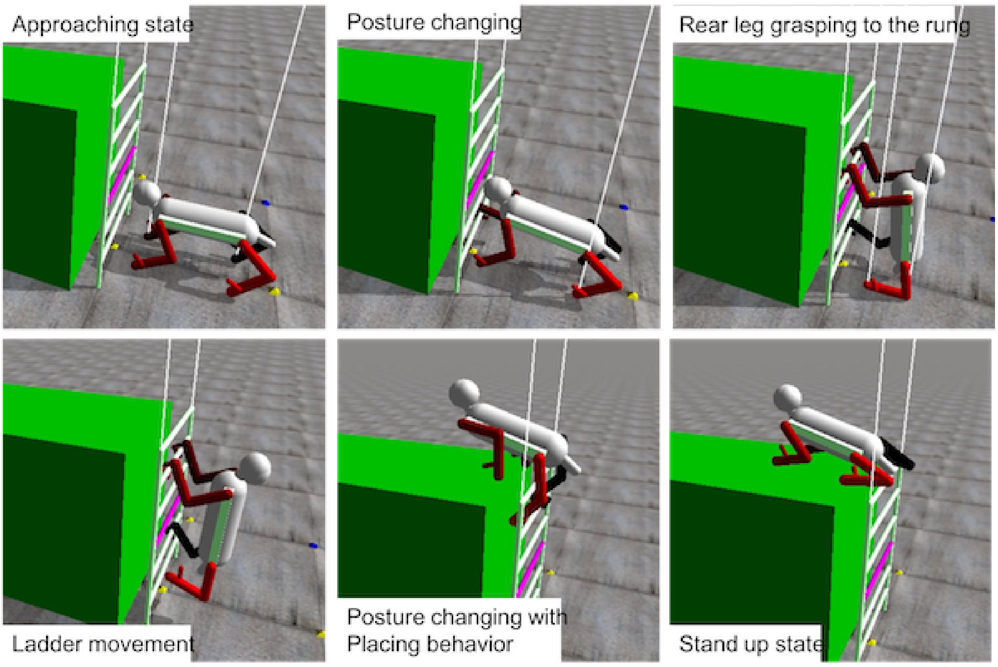


Figure 5.42: *The process of behavior moving through vertical ladder without handrail support*

stair until standing up position in the upper stair. In the first step, the robot walks approaching the rung of the ladder. Second, pose changing and grasping movement behavior on the rung of the ladder is performed until rear legs approaching the rung of the ladder. Third, the rear leg grasps the rung. Fourth, the robot performs the vertical movement with full grasping. Fifth, the body pose changing to horizontal movement and placing the robot's body in the safe area. Finally, the robot put all the legs on the initial position and ready to stand up in upper stair.

The semi-autonomous controller will be applied in the proposed robot. However, in order to realize those performances, the autonomous behavior generation should be installed. The user will only be giving the command at a higher level, such as movement goal position. After that, the robot will automatically generate the movement behavior depending on the given information from sensors.

5.4.2 Autonomous Behavior Generation

Environmental condition surrounding the ladder required several behaviors in movement, such as capabilities, crawling, posturing, climbing, etc. Those behaviors should be generated depending on the environmental condition being faced at a certain time. In order to move automatically, the autonomous model that can generate appropriate behavior is required.

The behavior generation is composed of artificial recurrent neural network (RNN) that use internal and external sensory information as the input. Internal sensory information implies the information from the internal body of the robot, such as pose angle, touching information, ground force information, etc. External sensory information implies the environmental condition such as ground condition, ladder or rung position, etc.

There are 9 neurons as input neurons (m). 4 neurons represent touching conditions (c_{tc}), if $(c_{tc} = 0)$ implies there is no touching, $(c_{tc} = 1)$ implies there is touching without grasping, $(c_{tc} = 2)$ implies there is touching with grasping condition. 2 neurons are in boolean value represent range of posture condition (c_{cp}) and stable pose condition (c_{sp}). The last 3 neurons also in boolean value represent a distance condition of ladder (c_{ld}), safe area condition (c_{sa}), and obstacle condition (c_{ob}).

In the output layer, one output neuron (O_1) will generate the activation value that choose the appropriate behavior. There are 4 behaviors defined in proposed model, approaching behavior, body placing behavior, stepping behavior, and grasping behavior.

$$\text{Activated behavior} = \begin{cases} \text{Approaching} & \text{if } 0 \leq O_1 < 1 \\ \text{Body Placing} & \text{if } 1 \leq O_1 < 2 \\ \text{Stepping} & \text{if } 2 \leq O_1 < 3 \\ \text{Grasping} & \text{if } 3 \leq O_1 < 4 \\ \text{None} & \text{otherwise} \end{cases} \tag{5.25}$$

The detail mathematical model and the structure of the proposed RNN have been explained in the our previous research [245].

In general, the process of the proposed model can be seen in Alg. 13. The weight parameter will be optimized in the computer simulation using an evolutionary algorithm.

Algorithm 13 General algorithm of the proposed model

```
loop
    Performing gaze analysis
    Generate the appropriate behavior
    loop
        Performing selected behavior
        if movement finished then
            exit
```

5.4.2.1 Posturing Behavior

In order to adjust and place the robot's body in the correct position for the next movement or stabilizing the robot's posture, posturing behavior is required. There are two sub-behaviors inside the posturing behavior, 1) Approaching behavior 2) body Placing behavior. Approaching behavior controls the body for approaching the next target stepping or grasping point in stable condition. Body Placing behavior is performed when all legs are not possible to reach or swing to the goal point. The difference between Approaching behavior and body Placing behavior can be seen in Fig. 5.44. When a certain leg is in its maximum extension, the behavior will stop. The other behavior may be required to be performed.

I define the length between the center of body's coordinate and base of leg as $O_i^{x,y,z}$ and the coordinate of leg's end of effector with leg's base as X_{leg}^i, Y_{leg}^i, Z_{leg}^i, where i is the identity of the leg. There are 6 DoF data that will be informed as the target movement, pitch (b^{α}), roll (b^{β}), yaw (b^{γ}), and X (b^x), Y (b^y), Z (b^z) will be controlled depending on the behavior command and the current body condition.

The kinematic illustration of pose control can be seen in Figs. 5.43 and 5.44. $\mathbf{O}_{LEG}$ is the leg position composed parameter X_{leg}^i, Y_{leg}^i, and Z_{leg}^i, where i is the identity of the leg. The rotational pose angle speed and translation movement speed are represented by $\dot{\theta}$ and $\mathbf{s}$ calculated in Eqs. 5.26 and 5.27.

$$\dot{\theta} = \|(\mathbf{F}_f - (\mathbf{F}_f \cdot \mathbf{n}_f)\mathbf{n}_f) + (\mathbf{F}_r - (\mathbf{F}_r \cdot \mathbf{n}_r)\mathbf{n}_r)\| \tag{5.26}$$

$$\dot{\mathbf{s}} = (\mathbf{F}_f \cdot \mathbf{n}_f)\mathbf{n}_f + (\mathbf{F}_r \cdot \mathbf{n}_r)\mathbf{n}_r \tag{5.27}$$

It is effected by front vector direction (F_f) and rear vector distance (F_R). F_F and F_R composed by goal distance vector (f_G), leg extension effect $(\mathbf{f}_E)$, and stepping movement effect $(\mathbf{f}_S)$. $\mathbf{f}_G$ calculated as $\mathbf{f}_G = \mathbf{G}_{GRAPS} - \mathbf{O}_{LEG}$. $(\mathbf{f}_E)$ and $(\mathbf{f}_s)$ calculate in Eqs. (5.28) and (5.29)

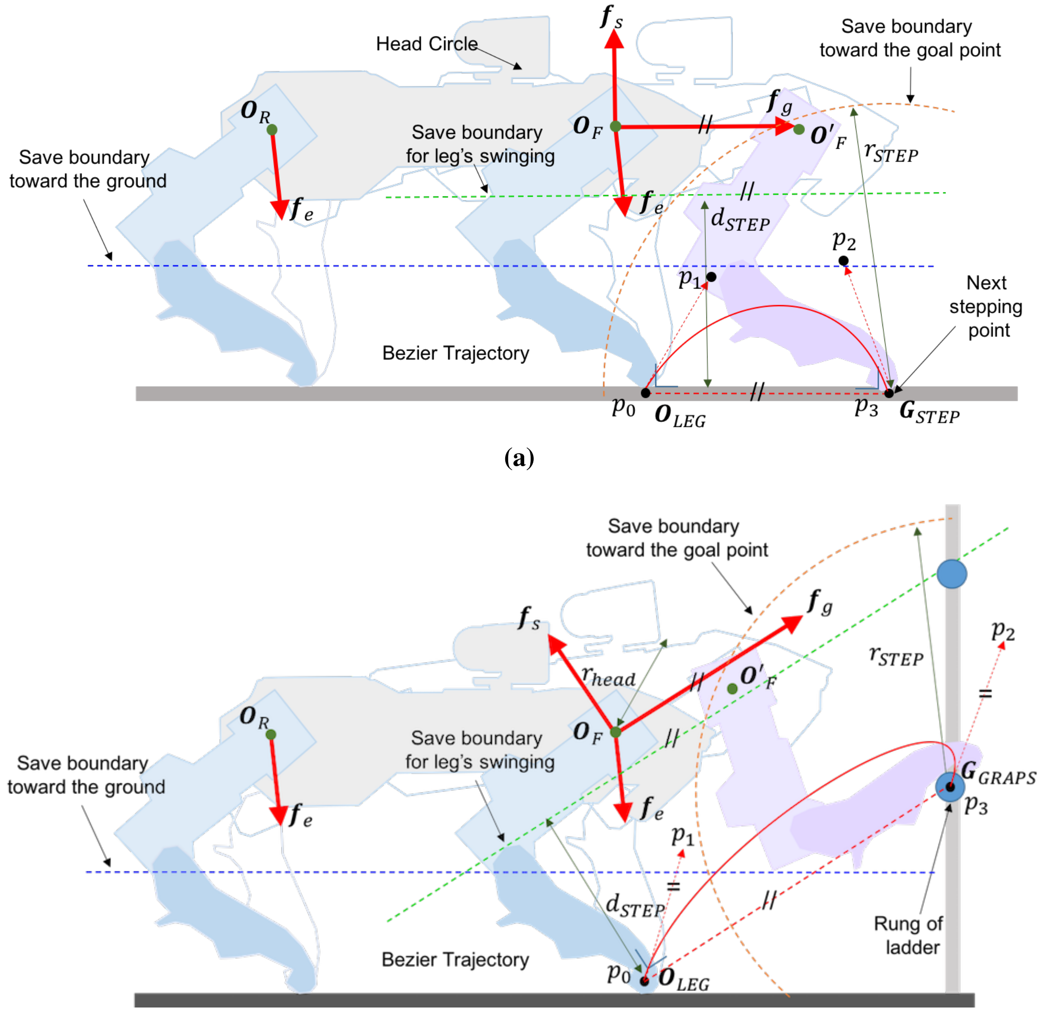
$$\mathbf{f}_e = \frac{1}{1+e^{-(|\mathbf{O}_f - \mathbf{O}_{LEG}|)/5}}(\mathbf{O}_f - \mathbf{O}_{LEG}) \tag{5.28}$$

$$\mathbf{f}_s = \begin{cases} \frac{d-|\mathbf{s}_f|}{|\mathbf{s}_f|}\mathbf{s}_f & \text{if } |\mathbf{s}_f| \leq d \\ 0 & \text{0therwise} \end{cases} \tag{5.29}$$

Where, $\mathbf{s}_f = (\mathbf{O}_f - \mathbf{O}_{LEG}) - ((\mathbf{O}_f - \mathbf{O}_{LEG}) \cdot \mathbf{n}_s)\mathbf{n}_s)$. $\mathbf{n}_s$ is unit vector of $(\mathbf{O}_f - \mathbf{O}_{LEG})$.

In order to consider the robustness of the posture, the center of mass (COM) effect is also considered. I define $\mathbf{O}_{COG}$ as the COM of robot and $\mathbf{O}_{CENTER}$ as the center of supporting area. The effect of COM vector (f_{COM}) is calculated in Eq. (5.30)

$$\mathbf{f}_{COM} = \frac{|\mathbf{O}_{COG} - \mathbf{O}_{CENTER}|^2}{100}(\mathbf{O}_{COG} - \mathbf{O}_{CENTER}) \tag{5.30}$$



(a)

(b)

Figure 5.43: *Posturing behavior from horizontal position to vertical position a) Approaching behavior to the certain point in safe area b) Approaching behavior to the grasping position in the rung of ladder*

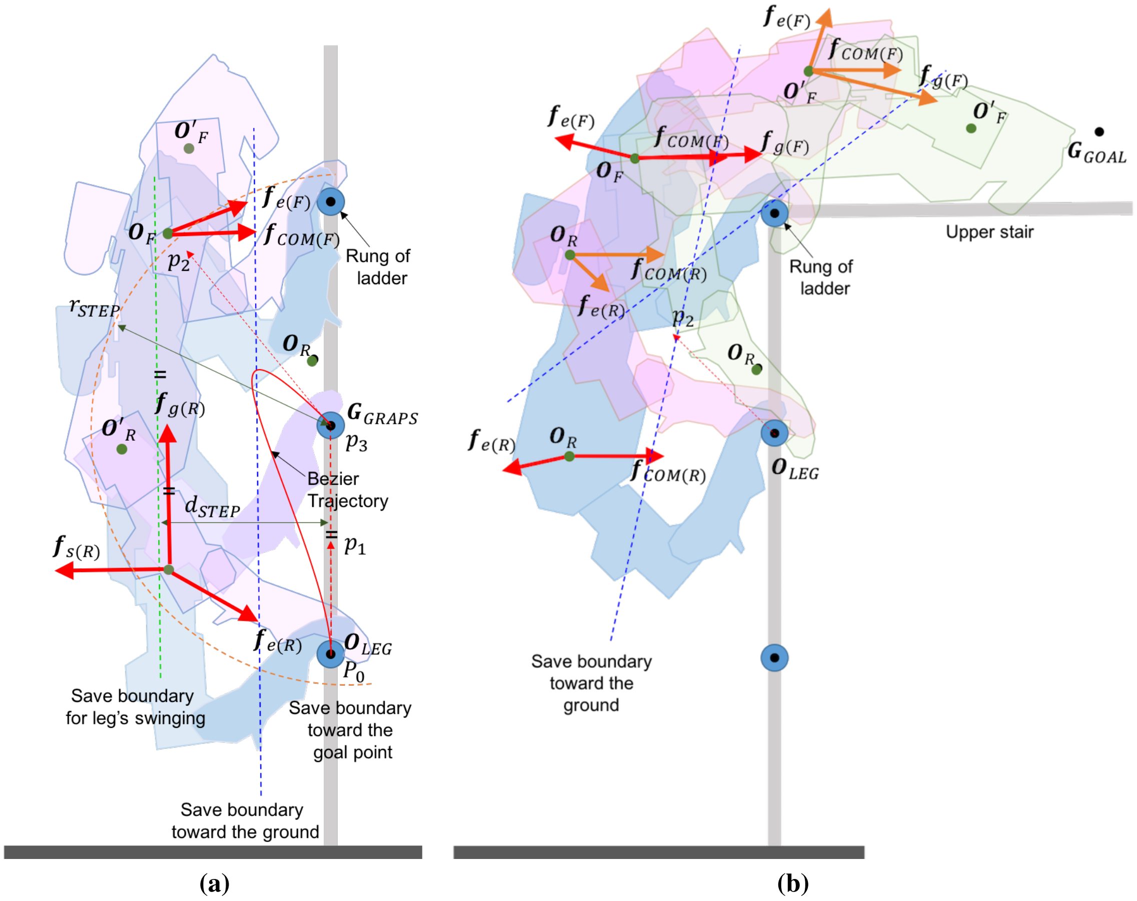


Figure 5.44: *Posturing behavior in vertical position a) Approaching behavior to the grasping position in the rung of ladder b) Body placing behavior to the certain position on the safe area*

The current position and posture parameter can be seen in Eq. 5.31 and Eq. 5.32, respectively.

$$\mathbf{s}(t+1) = \mathbf{s}(t) + \dot{\mathbf{s}} \tag{5.31}$$

$$`(t+1) = `(t) + \dot{`} \tag{5.32}$$

In every time cycle (t), the current position ($\mathbf{s}$) and current posture ($`$) are added by $\dot{\mathbf{s}}$ and $\dot{`}$, respectively. When condition $||\mathbf{O}^{i}_{LEG}|| \geq L^{i}_{max}$ is occur certain ith leg, the posturing control will be interrupted. L_{max} is the maximum extension of the leg.

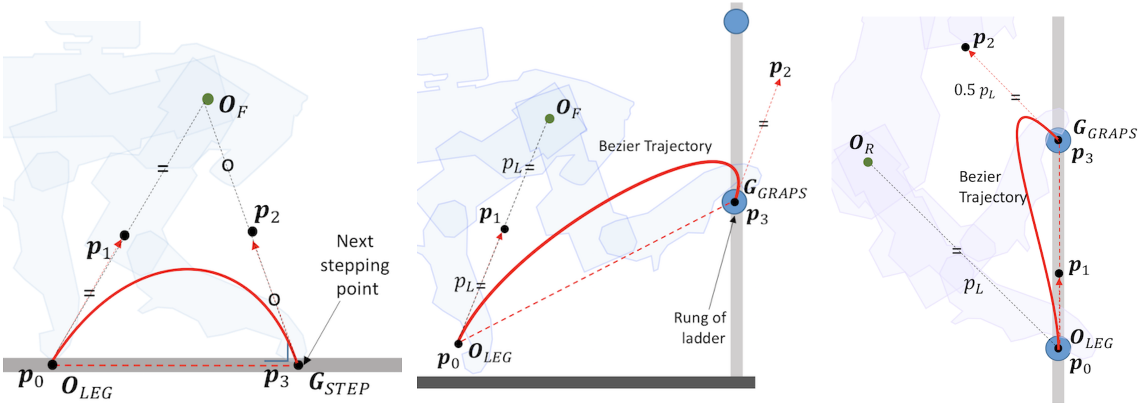
5.4.2.2 Swinging Behavior

Swinging behavior is separated into two behavior depending on the touching goal position, which is stepping behavior and grasping behavior. When the robot faces the safe area and being asked the leg to move to the goal point, or when a certain leg is in its maximum extension during posturing behavior, stepping behavior is performed. When the robot faces the ladder and being asked the leg to move to the rung, then grasping behavior will be performed. The model will generate the swinging movement of the certain leg in Cartesian level. The swinging movement illustration can be seen in Fig. 5.45. The swinging trajectory is generated from a 2-dimensional quadratic Bézier curve.

$$\mathbf{B}(t) = \sum_{i=0}^{n} \binom{n}{i} t^{i}(1-t)^{n-1}\mathbf{P}_{i} \tag{5.33}$$

The mathematical model of Bézier curve model is presented in Eq. (5.33), where the Bézier curve point in time t is denoted by $\mathbf{B}(t)$; t is from 0 to 1; n represents the number of degrees in Bézier curve and $\mathbf{P}_i$ is the ith selection point. The swinging direction in starting point is composed by $\mathbf{p}_0$ and $\mathbf{p}_1$ and finishing direction to goal point is composed by $\mathbf{p}_2$ and $\mathbf{p}_3$.

There are 3 behaviors in swinging movement shown 1) stepping behavior from ungrasping condition 2) grasping behavior from ungrasping behavior 3) grasping behavior from grasping condition. The difference of these behavior is in the end of effector position and pattern of trajectory or the calculation of selection point $\mathbf{p}_i$. If The starting position in ungrasping condition, $\mathbf{p}_i$ is calculated as $\mathbf{p}_0 = \mathbf{O}_{LEG}$ and $\mathbf{p}_1 = 0.5(\mathbf{O}_{FR} - \mathbf{O}_{LEG}) + \mathbf{O}_{LEG}$. If the starting is in grasping condition, $\mathbf{p}_0 = \mathbf{O}_{gLEG}$ and $\mathbf{p}_1 = 0.3(\mathbf{G}_{GRASP} - \mathbf{O}_{gLEG}) + \mathbf{O}_{gLEG}$. There are also different direction to the goal point between grasping behavior and stepping behavior. If the grasping mode, $\mathbf{p}_2 = (\mathbf{O}_{FR} - \mathbf{O}_{LEG}) + \mathbf{G}_{GRASP}$ and $\mathbf{p}_3 = \mathbf{G}_{GRASP}$. If in the stepping mode, $\mathbf{p}_2 = (\mathbf{O}_{FR} - \mathbf{G}_{STEP}) + \mathbf{G}_{STEP}$ and $\mathbf{p}_3 = \mathbf{G}_{STEP}$. If the next step is front leg, then $\mathbf{O}_F$ is used as the reference point. If the next step is rear leg, then $\mathbf{O}_R$ is used as the



(a) **(b)** **(c)**

Figure 5.45: *Kind of behavior in swinging movement a) stepping behavior from ungrasping condition b) grasping behavior from ungrasping behavior c) grasping behavior from grasping condition.*

reference point.

5.4.2.3 Recovering from the swinging failure

During the swinging movement, some failures maybe happened. Sometimes the touching position is incorrect, or the swinging is stuck by some obstacle. In the proposed model, I utilize the two sensors (force and switch sensor) attached at the end of effector for failure analysis. When the switch sensor is not detected at the end of grasping behavior or the force sensor is detected without any detection from the switch sensor, then the model considers that there is failure swinging. Then the model will re-analyze the environmental condition and re-performed the swinging movement with the corrected position.

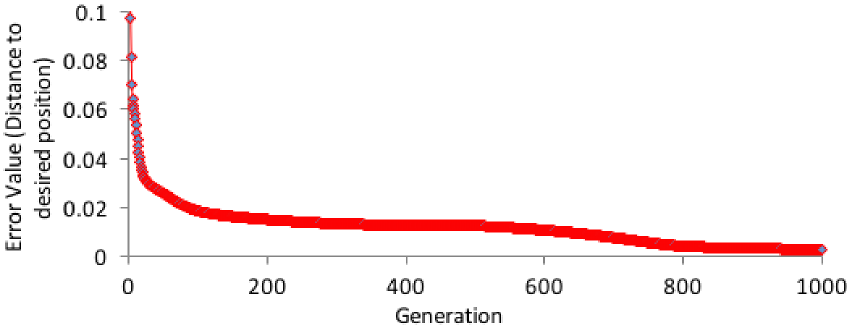


Figure 5.46: *The error evolution in every generation*

5.4.3 Real-Time Ladder Affordance Detection

The proposed DD-GNG is applied to continuously search, detect and track rungs of the ladder by generating topological structure information according to input data. In addition, the generated topological structure is utilized to determine the possible grasping point on the detected rung. The main algorithm is detailed in Algorithm 14, where $\mathfrak{R}$ is the collection of parameters for the detected rungs. $\mathfrak{R} \rightarrow N > 0$ is the number of detected rungs. For each iteration, detected rungs are evaluated and its position is updated depending on the movement of nodes.

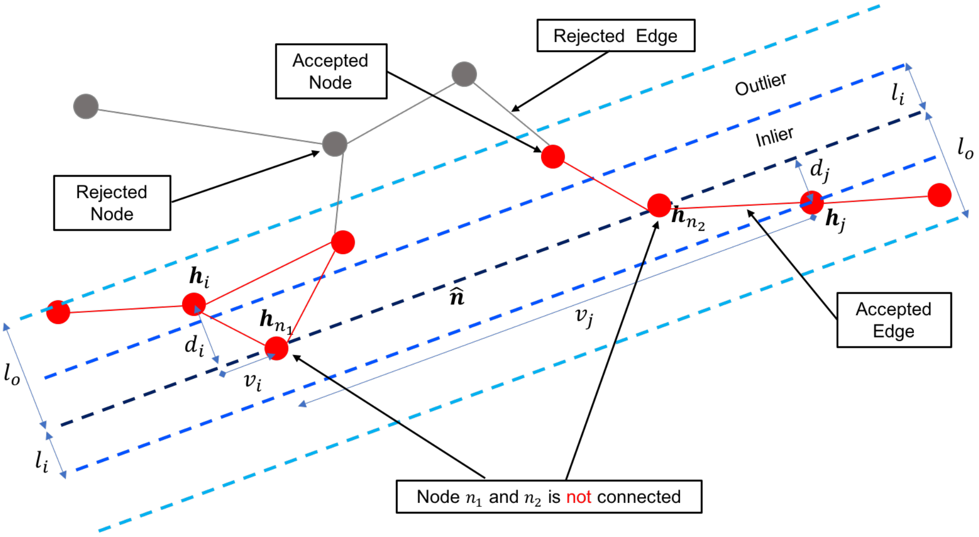


Figure 5.47: *Illustration of evaluation when generated nodes n_1 and n_2 is not connected*

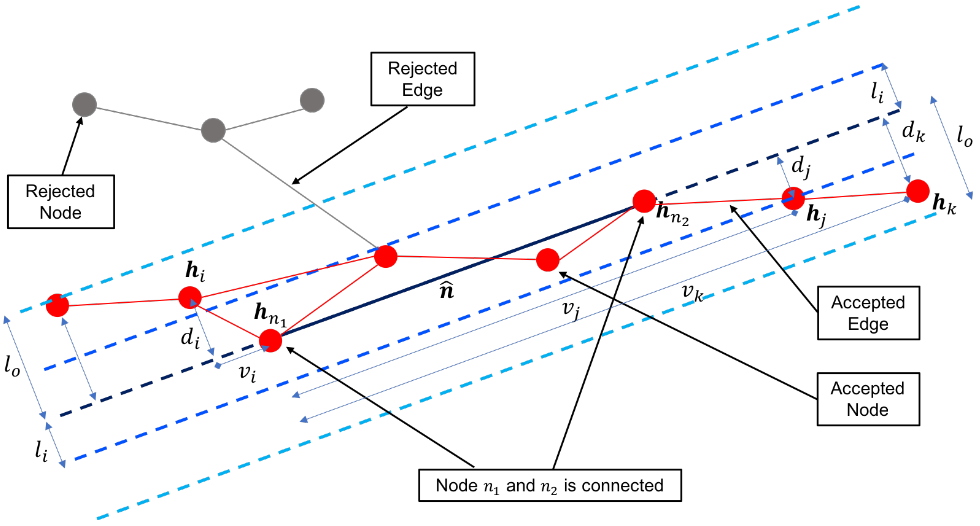


Figure 5.48: *Generated nodes n_1 and n_2 is considered as a rung. The figure show the parameter involved for evaluation and position correction*

5.4.3.1 Determine Perspective Rung

In order to find perspective rungs, two nodes (h_{n_1}, h_{n_2}) in the speculated area are selected randomly. After that, I evaluate the nodes by considering the nodes connectivity and density in the surrounding line through node n_1 and n_2. Fig. 5.48 shows the example condition of accepted connectivity. Then, Fig. 5.47 shows the failed ladder detection, thus the next step will not processed when face this condition The evaluation steps are shown in Algorithm 15. The evaluation value (Θ) is calculated using Eq. 5.34. c_i and c_o represents the number of nodes in the inlier area and outliers area respectively. $\mathbf{n}^{(1)}$ and $\mathbf{n}^{(2)}$ is the starting nodes and end nodes of the rung. If the evaluation value lower than the given threshold λ_Θ, then the detected rung is rejected.

Algorithm 14 Main algorithm of ladder affordance detection

```
data: $\mathfrak{R}$ as the data struct of rung, $\mathfrak{N}$ as the data struct of node DD-GNG
if there is suspected object then
    if $\mathfrak{R} \to N > 0$ then
        rung_position_correction($\mathfrak{R}$)
        calculating_age_of_rung($\mathfrak{R}$)
    searching prospective rung
    $n_1$ = get_random_suspected_node
    $n_2$ = get_random_suspected_node, $n_2 \neq n_1$
    Evaluating_rung($\mathfrak{N}, \mathfrak{R}, n_1, n_2$), (Alg. 15)
```

$$\mathfrak{R} \to \Theta = \frac{(c_i)/(1+c_o)}{\|\mathbf{n}^{(2)} - \mathbf{n}^{(1)}\|} \tag{5.34}$$

$$\text{Eval. result} = \begin{cases} \text{Pass, } \mathfrak{R} \to N++ & \text{if } \Theta \geq \lambda_\Theta \\ \text{Reject} & \text{otherwise} \end{cases} \tag{5.35}$$

5.4.3.2 Rung Position Correction and Update

In order to correct the position of the rung, the proposed method tracks the moving nodes in the inlier and outlier area. Based on the definition in Fig. 5.48, the error movement of starting node and end node are calculated using Eqs. 5.36 and 5.37, where c_i is the number of node in inlier area. $\mathfrak{R} \to N_i$ is the number of nodes involve in ith detected rung.

After calculating the error position of detected rungs, the start and end points of the rungs are updated using Eqs. 5.38 and 5.37. $\mathfrak{R} \to h_1$ and $\mathfrak{R} \to h_2$ are the start and end point of the detected rung.

$$\Delta e_1 = \frac{1}{c_i} \sum_{j=\mathfrak{R}\to N_i}^{j=0} \frac{\|\mathbf{v}_j\|}{\|\mathbf{v}^{(2)}\|} (\mathbf{r}_j - \mathbf{v}_j) \tag{5.36}$$

Algorithm 15 Evaluating_Rung($\mathfrak{N}, \mathfrak{R}, n_1, n_2$)

```
$\hat{n} = \frac{\mathfrak{N}\to\mathbf{h}_{n_1} - \mathfrak{N}\to\mathbf{h}_{n_2}}{\|\mathfrak{N}\to\mathbf{h}_{n_1} - \mathfrak{N}\to\mathbf{h}_{n_2}\|}$
$i = \mathfrak{R} \to N$, *get rung id*
$\mathfrak{R} \to N_i = 1; \mathfrak{R} \to L_{i,0} = n_1; \mathfrak{R} \to L^{id}_{i,n_1} = n_1$
$c \leftarrow 0$, *define initial connection condition*
Get_Connection_Condition($\mathfrak{N}, \mathfrak{R}, n_1, n_2, \hat{n}$)
$c_o \leftarrow 0, c_i \leftarrow 0$
**if** $c = 1$ **then**
    $\mathbf{v}^{(1)} = 0$
    $\mathbf{v}^{(2)} = \mathfrak{N} \to \mathbf{h}_{n_1} - \mathfrak{N} \to \mathbf{h}_{n_2}$
    **for** $j \leftarrow 1$ to $\mathfrak{R} \to N_i$ **do**
        **if** $\mathfrak{R} \to L^{id}_{i,\mathfrak{R}\to L_{i,j}} = 1$ **then**
            $c_i \leftarrow c_o + 1$
        **else if** $\mathfrak{R} \to L^{id}_{i,\mathfrak{R}\to L_{i,j}} = 2$ **then**
            $c_o \leftarrow c_i + 1$
        **if** $c_i > \lambda_1$ **then**
            **for** $j \leftarrow 1$ to $\mathfrak{R} \to N_i$ **do**
                **if** $\mathfrak{R} \to L^{id}_{i,\mathfrak{R}\to L_{i,j}} = 1$ **then**
                    $\mathbf{r} = \mathfrak{N} \to \mathbf{h}_{n_1} - \mathfrak{N} \to \mathbf{h}_{\mathfrak{R}\to L_{i,j}}$
                    $\mathbf{v} = (r \cdot \hat{n})\hat{n}$
                    **if** $\|\mathbf{v} - \mathbf{v}^{(2)}\| \geq \|\mathbf{v} - \mathbf{v}^{(1)}\|$ **then**
                        **if** $\|\mathbf{v} - \mathbf{v}^{(2)}\| \geq \|\mathbf{v}^{(1)} - \mathbf{v}^{(2)}\|$ **then**
                            $\mathbf{v}^{(1)} = \mathbf{v}$
                    **else if** $\|\mathbf{v} - \mathbf{v}^{(2)}\| < \|\mathbf{v} - \mathbf{v}^{(1)}\|$ **then**
                        **if** $\|\mathbf{v} - \mathbf{v}^{(1)}\| \geq \|\mathbf{v}^{(1)} - \mathbf{v}^{(2)}\|$ **then**
                            $\mathbf{v}^{(2)} = \mathbf{v}$
    *calculating evaluation value*
    $\Theta \leftarrow$ Eq. 5.34
    **if** Passed **then**
        $\mathfrak{R} \to N{+}{+}$
        $\mathfrak{R} \to h_1 = \mathfrak{N} \to \mathbf{h}_{n_1} - \mathbf{v}^{(1)}$
        $\mathfrak{R} \to h_2 = \mathfrak{N} \to \mathbf{h}_{n_1} - \mathbf{v}^{(2)}$
```

$$\Delta e_2 = \frac{1}{c_i} \sum_{j=\mathfrak{R}\rightarrow N_i}^{j=0} \frac{\|(\mathbf{v}^{(2)} - \mathbf{v}^{(1)}) - \mathbf{v}_j\|}{\|\mathbf{v}^{(2)}\|} (\mathbf{r}_j - \mathbf{v}_j) \tag{5.37}$$

$$\mathfrak{R} \rightarrow h_1 = (\mathfrak{R} \rightarrow \mathbf{R}_{h_1} - \mathbf{v}^{(1)}) - \varepsilon \Delta e_1 \tag{5.38}$$

$$\mathfrak{R} \rightarrow h_2 = (\mathfrak{R} \rightarrow \mathbf{R}_{h_1} - \mathbf{v}^{(2)}) - \varepsilon \Delta e_2 \tag{5.39}$$

5.4.3.3 Calculating Age of Rung

Once the speculated rung is fulfilled the evaluation stage, it does not imply that the rung is detected. It takes several iteration for measuring the confidence level the speculated rung by calculating the their age. To do that, the initial age value of the rung is given ($\mathfrak{R} \rightarrow g \leftarrow \lambda_{g1}$).The evaluation value ($\mathfrak{R} \rightarrow \Theta$) of each speculated rung is vary for each iteration. If $\mathfrak{R} \rightarrow \Theta < \lambda_{\Theta}$, the age of rung is then decreased. If the age value is lower than 0, then the rung is removed. If $\mathfrak{R} \rightarrow \Theta > \lambda_{\Theta}$, the age value is increased. When the age is more than a trusted threshold (λ_{g2}), it means that the rung is detected.

5.4.4 Experimental Result

I tested the proposed model in both simulation and real quadruped robot explained in Appendix A. The robot is facilitated by Quad Time of Flight (ToF) sensor for detecting the rung position, force, and touch sensors at every end of effector, inertial measurement unit (IMU) for pose analysis, and grasping mechanism also at the end of the effector.

5.4.4.1 Ladder detection test

Ladder Detection I performed the proposed ladder detection using a hand-made ladder, where the space between two rungs in the vertical ladder is 180 mm, the length of each rung is 455 mm. There are a total of four rungs in the ladder. The height between the ground and the highest point of the ladder is 710 mm. The ladder detection process is recorded during the experiment and the result of the detection is shown in Fig.5.49. The result shows that the conventional GNG is unable to detect the ladder. The DD-GNG with dynamic density features able to generate a topological structure that representing the shape of objects in real-time by increasing the number of nodes on speculated objects. In addition, DD-GNG generates a safe grasping point and unsafe grasping point as the affordance input for robot behavior control.

Real Robot Grasping Performance In order to show the efficiency of the proposed model, I show the performance of ladder detection while the robot approaching the ladder.

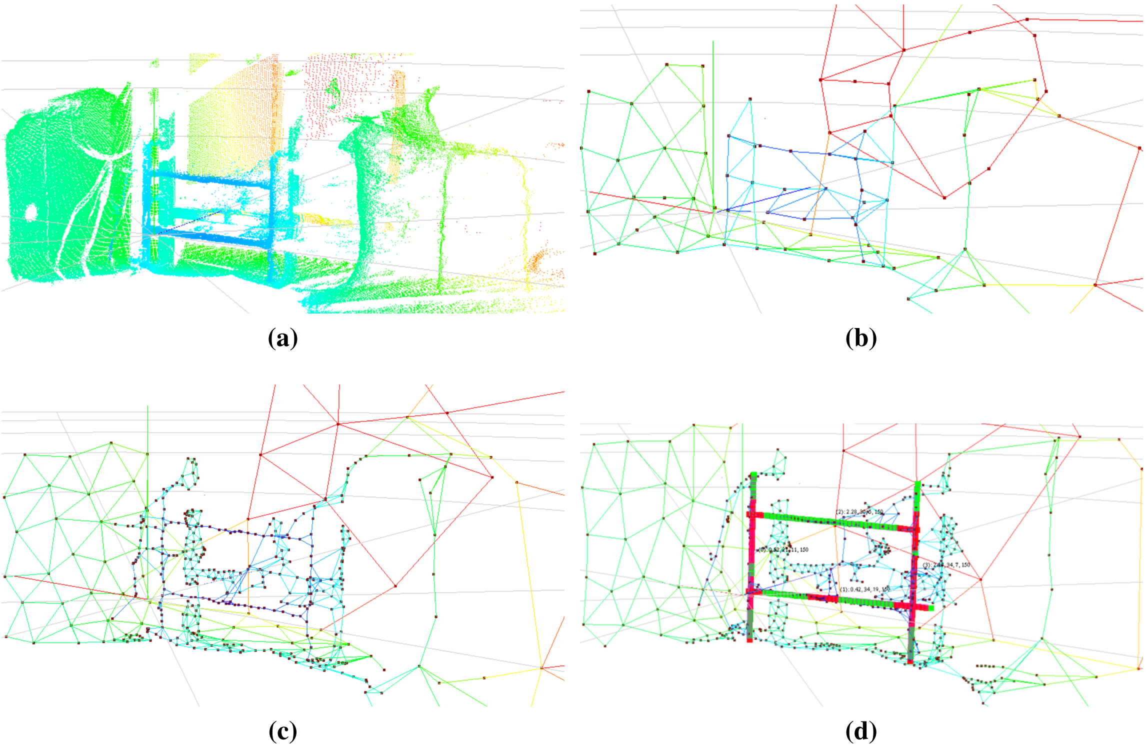


Figure 5.49: *The result of ladder detection. (a) Raw point clouds data (b) Generated structure without dynamic density (c) Topological structure generated by DD-GNG (d) Result of ladder and rungs detection with grasping location information, green area represents the safe grasping zone and red area represent unsafe zone for grasping.*

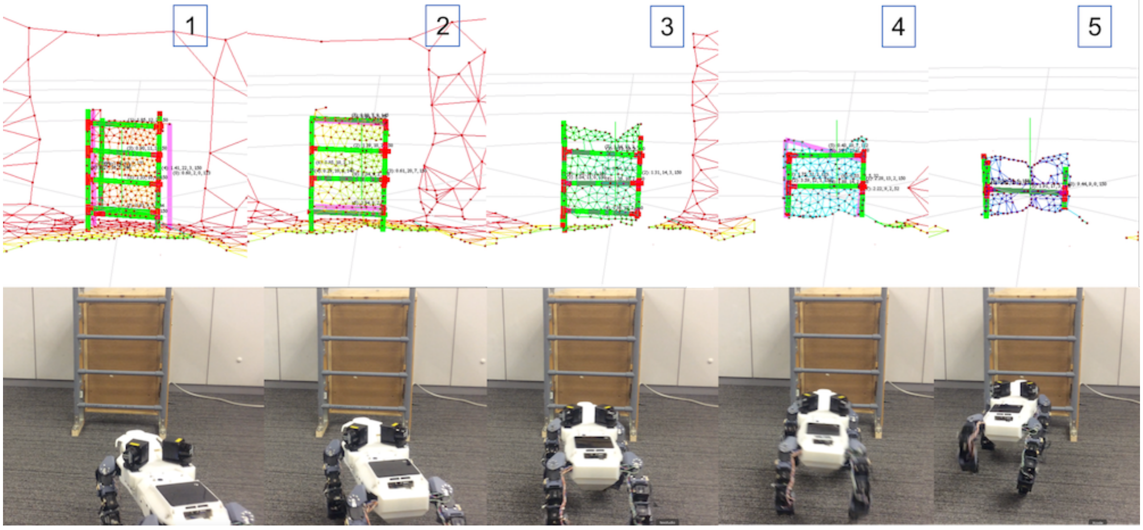


Figure 5.50: *The performance of proposed model while approaching the ladder*

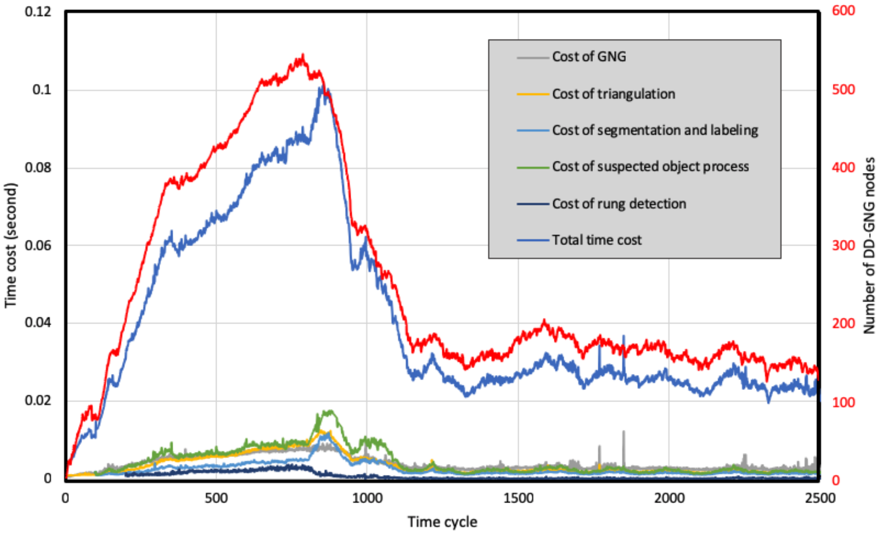


Figure 5.51: *Processing time of proposed model while approaching the ladder*

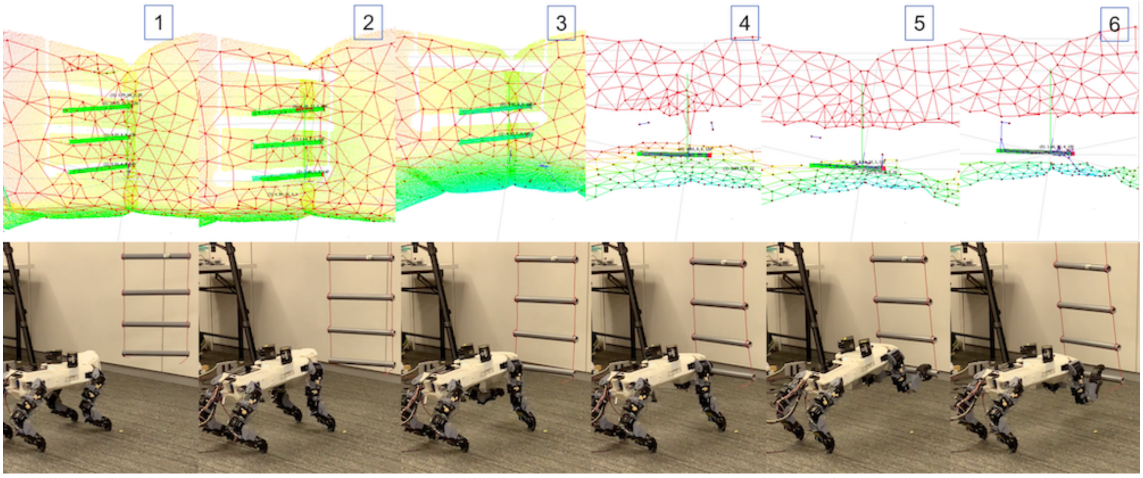


Figure 5.52: *The robot performance while tracking the moving ladder and grasping the rung of ladder*

The performance results are shown in Fig. 5.50. Further experimental results can refer to the supplementary video. There are three colors provided by ladder affordances, Pink, Red, and Green represent speculated rung of the ladder, unsafe grasping position, and safe grasping position, respectively.

The computational cost during ladder detection experiment is recorded as shown in Fig. 5.51. The average processing time for each iteration is about 30 ms. Thus, it is fast enough for tracking the position of the ladder during robot movement. In addition, I conducted an experiment in which a robot has to detect a moving rope ladder and perform grasping behaviors for further validation of the proposed method. The snapshot of the experimental results is shown in Fig. 5.52 and details of the experiment is available in the supplementary video.

5.4.5 Climbing behavior experiment

Before performing the proposed movement behavior in the real environmental condition, the optimization process is required for optimizing the relationship between the environmental condition and the generated behavior. The process was conducted in a computer simulation open dynamic engine. The robot and environmental condition were set similar to the real one. In order to respond to the internal feedback, the stability control developed in Section 5.3 is installed. It uses sensorimotor coordination based control [249].

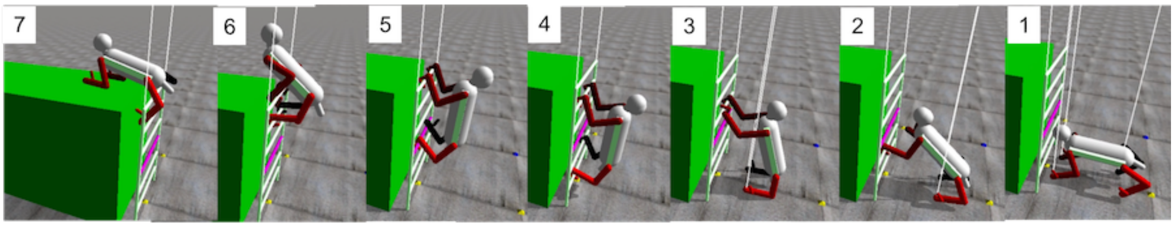


Figure 5.53: *Robot performance in computer simulation Open Dynamics Engine*

In the simulation, the robot was retrained continuously from standing on the ground until the robot stands on the upper stair using the evolutionary algorithm. The rest of the distance between the current position and goal position is used as the error evaluation. The decreasing of error can be seen in Fig. 5.46

After 1000 generation, the optimization process finished. The robot could move from through vertical ladder without a handrail. The proposed behavior generation could generate appropriate behavior depending on the environmental condition facing. The robot performance in simulation can be seen in Fig. 5.53. The detail of the robot in real performance and in the simulation are shown in attached video supplementary.

5.4.5.1 Real Implementation

After optimizing the process in simulation, I performed the proposed model with optimized parameter in the real implementation using described quadruped robot. The environmental information was set as follows:

- Interval between two rungs in the vertical ladder is 180 mm.
- The length of each rung is 455 mm
- number of the rung in the ladder is four rungs.
- The height between the ground and upper stair is 710 mm.

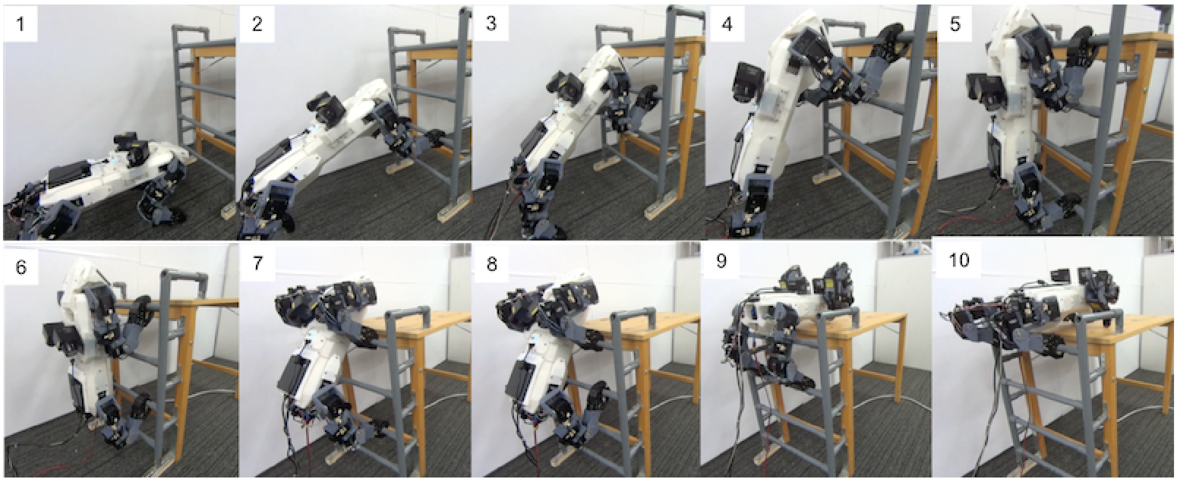


Figure 5.54: *The performance of the real robot moving from ground to upper position through the ladder without handrail supporter. The robot successfully performed the proposed behavior especially Body Placing behavior (see num. 9 - 10)*

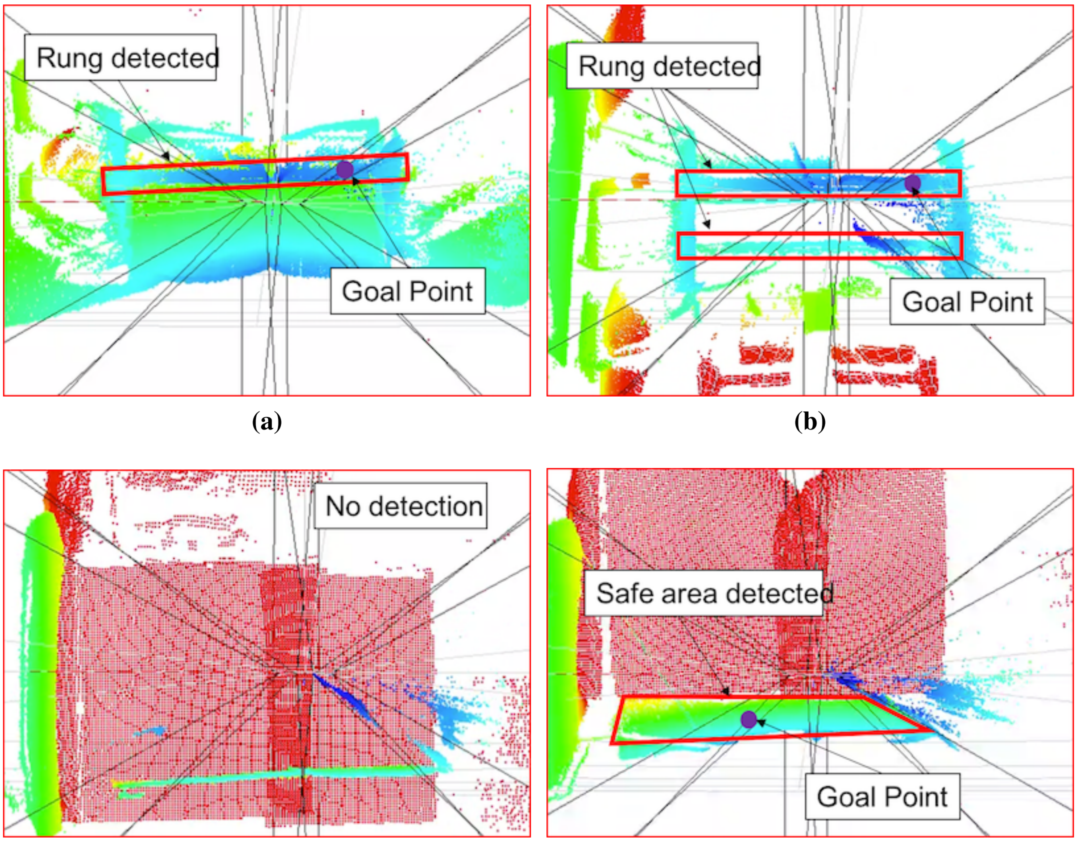
The robot was set on the ground position in front of the ladder with the default posture. The goal position is set as the robot standing on the upper stair. The snapshots of the robot performance can be seen in Fig. 5.54.

5.4.5.2 Gaze Analysis (External Information)

In the gaze analysis, the external information from the Quad Time of Flight sensor is processed to the topological maps data. Next, the topological maps data are processed for environmental detection. The sample of result from the robot's performance can be seen in Fig. 5.55.

5.4.5.3 Error grasping recovery

When the robot performs grasping the rung of the ladder, the grasping failures happened during robot performance. The snapshots of sample failure can be seen in Fig. 5.56. When the front left leg was swinging for grasping the rung, the force sensor detected touching something without any touching detections in switch sensor (see Fig. 5.56 no. 5). The model analyzed that there was a failure in the performance. After that, the robot re-performed the swinging movement to the corrected position (see Fig. 5.56 no. 6-9). This proved the effectiveness of the recovery process in the proposed model.



(a) (b)

(c) (d)

Figure 5.55: *Sample of external information from point cloud data generated by Quad Pico Flexx sensor a) The rung of ladder was detected and the goal point was decided parallel with body posture direction in z-axis b) There are two rungs detected and the most appropriate goal point was decided c) There is no information detected d) Safe area is detected and goal point decided in the center of safe area*

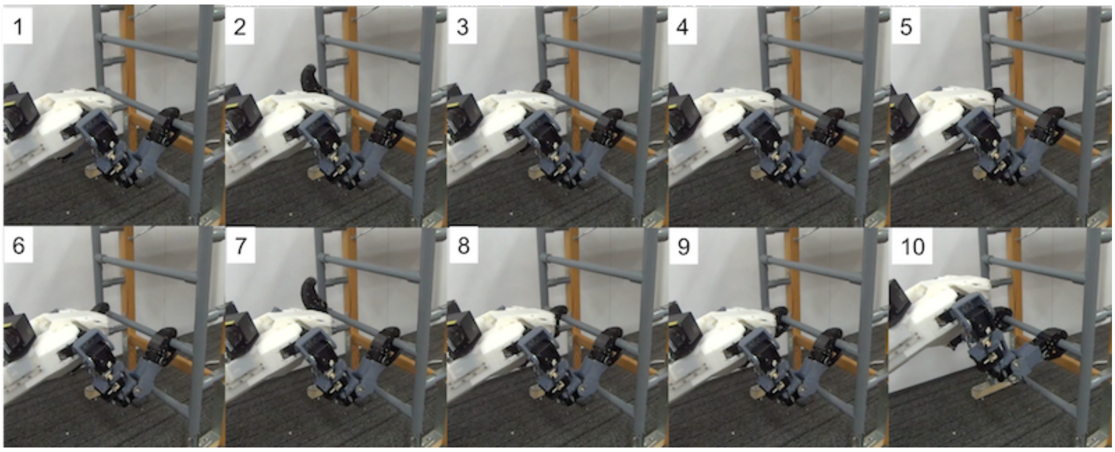


Figure 5.56: *Grasping failure during robot performance and its auto recovery behavior*

5.4.6 Conclusion on Climbing Behavior

I have proposed novel capabilities in quadruped robot that can move from the ground to upper stair through a vertical ladder without handrail supporter. The autonomous behavior generation is proposed for performing appropriate action when facing certain environmental condition. There are four behaviors that separately activated depending on the environmental condition and desired goal position, which is posturing behavior and swinging behavior. Posturing behavior is expanded into approaching behavior and body placing behavior. Then, swinging behavior is expanded into stepping behavior and grasping behavior. In order to optimize the behavior generation model, I successfully used the evolutionary algorithm in the computer simulation after 1000 generations. The optimized parameter was used for real performance. The robot succeeded to move from ground position to upper position. One of the difficulties in vertical ladder movement without a handrail is the transition from vertical climbing to the upper stair. The body placing behavior is the most important strategy in the proposed model to solve the difficulties. The performance also proved the effectiveness of the proposed recovery process in swinging behavior. However, in order to perform in the real implementation, I needed to attempt five performance before achieving success performance. The failure of the performance is caused by the limitation of the motor's torque. However, the failure does not reduce the superiority of the proposed model.

In order to perform in a new vertical ladder environment, it requires a learning process in the simulation before applied in the real implementation. In the future plan, I will consider the general learning model for general vertical ladder implementation. Then, additional behavior will be developed for covering many environmental conditions. Furthermore, a friendly user interface will also be developed to support the command system.

Chapter 6

Concluding Remarks

6.1 Conclusion

This thesis aims to realize adaptability and optimality in the multi-legged locomotion robot. The human dynamic locomotion is realized by the simultaneous integration based on adaptability and optimality. The adaptability depends on the real-time perception as a bottom-up learning-based approach from the microscopic point of view, while the optimality depends on cognition as a top-down knowledge-based approach from the macroscopic point of view. Various cognitive architectures have been proposed in the cognitive science until now. Still, it is difficult for robotic researchers to apply such cognitive models to deal with adaptability and optimality simultaneously in real dynamic environments. Most of them only explain the information flow of human cognitive behaviors at the conceptual level, and the methodology to implement functions like human cognitive behaviors is not described in detail. Therefore, I proposed the cognitive model that can cover and integrate the mobility from microscopic control to macroscopic planning and from short-term adaptation to long-term optimization.

This thesis proposes a neuro-cognitive model for multi-legged locomotion to realize the seamless integration from multi-modal sensing, ecological perception, behavior generation, and cognition through the coordination of interoceptive and exteroceptive sensory information. The proposed cognitive model is discussed from three different scopes: micro-, meso-, and macro-scopic, corresponding to sensing, perception, and cognition, and short-, medium- and long-term of adaptation related with neuro science and ecological psychology. I built the intelligent functions for the multi-legged locomotion, which includes: 1) attention module, 2) adaptive locomotion control module, 3) Object recognition module, 4) environmental map building module, and 5) optimal motion planning module. The proposed neuro-cognitive model integrates the above intelligent functions from the multi-scopic point of view.

6.1.1 Microscopic level

The proposed model at the microscopic level has been explained in Chapter 3. I proposed an attention mechanism for exteroceptive sensory information according to the current interoceptive sensory information, and adaptive locomotion control is done by (the lower-level of) sensorimotor coordination based on interoceptive and exteroceptive sensory information as a short-term adaptation. Furthermore, online locomotion generation is also processed at this level. I proposed the concept of sensorimotor coordination model based on the perceiving-acting cycle at the microscopic level, which is a lower-level control system interacting directly with the environment. The system at microscopic level is composed by three modules 1) attention mechanism module (see Section 3.1), 2) neural-based locomotion (see Section 3.2), and 3) object affordance-effectivity fit (see Section 3.3).

1. The attention mechanism module controls the topological structure of 3D point cloud information by using Dynamic Density Growing Neural Gas (DD-GNG). It is explained in Section 3.1. DD-GNG can control the density of topological structures in specific area based on the attention. In DG-GNG, the density of nodes in speculated areas or objects is increased automatically. The segmentation model can differentiate the safe terrain, wall, and speculated obstacles. Once speculated obstacles are detected, the proposed module automatically increase density of node around that area and the information is used for calculating the strength of nodes. I also proved the effectiveness of the proposed dynamic attention module for object grasping detection in Section 3.1.3.

2. Neural based locomotion module generates dynamic gait patterns by integrating with sensorimotor coordination. The module has presented an efficient dynamic-gait-pattern-generatorbased CPG mechanism with sensorimotor coordination. The module developed features substantial integration of CPG and sensory feedback, indicating that sensory feedback has a role in the gait-pattern generator. It has been explained in Section 3.2. The proposed model has been optimized through a learning process in the computer simulation beforehand. The locomotion ability of the real multi-legged robot and the leg malfunction tests are demonstrated in the experiments by several different terrains. The experimental results show that a smooth gait-pattern transition could be generated during sudden leg malfunction.

3. Object affordance-effectivity fit is for the direct perception to generally identify the environmental condition based on the physical embodiment. It integrates exteroceptive sensory information and locomotion generator module in short-term adaptation. In this module, I represent the model from the biological process in human or animals point of view. I propose a process from cognition to the movement-related tract, based on a top-down model in mammals. Cognition involves the perception of external sensory information. It also includes a way to control the attention to external information for movement behavior. I have proved and showed that our model integrates exteroceptive sensory information and low-level control muscle control in Section 3.3. The system can therefore respond to environmental changes in every time cycle. I also show the effectiveness of the proposed module with other locomotion model and its prospect to interdisciplinary studies.

6.1.2 Macroscopic level

At the macroscopic level, I focus on the development of optimality model explained in Chapter 4.1 design higher-level processing can conduct motion planning, behavior generation, and knowledge building. We build two modules, 1) environmental knowledge by using the topological structure-based map reconstruction 2) optimal path planning method on the constructed map.

1. I build environmental knowledge with higher level of behavior planning from (the collection or memory of) large scale of sensory information. The robot can conduct optimal motion planning using the built environmental knowledge. I proposed the method of building environmental knowledge by using the topological structure-based map reconstruction. Experimental results show that the proposed method can extract environmental features for multi-legged robots and build environmental knowledge for optimal path planning.

2. Next, I proposed an optimal path planning method on the built map based on neuronal activity. It has been explained in Section 4.2. There are two algorithm processes in this proposed module, forward transmission (FT) and synaptic pruning with the backward transmission. First, FT generates the impulse signal to neighbor neurons, defines the travel cost, and creates the neuron connection. Second, after target neuron is connected and gets the impulse, then the synaptic pruning and backward transmission are performed. These algorithms are implemented in grid map model and topological map model. One point is represented by a neuron in this proposed model. Based on the ex-

periments in both grid map model and topological map model, FT model successfully constructed the neuron connections for finding the possible way. SP with BT also successfully found the possible pathway from the current position to target position and reduced inefficient neuron in both map models, however the considered condition in SP model is required to be increased. These possible pathways are also proved by performing the robot in both computer simulation and real robot. The robot successfully moved from starting position and target position. The flexibility of this dynamic path planning was proved by the experimental result in computer simulation and real robot. When the robot follows the pathway and finds the obstacles, then the path planning model is changed to another possible pathway which is appropriate with robot ability.

6.1.3 Mesoscopic level

At the mesoscopic level, the proposed neuro-cognitive model integrates these two approaches of microscopic and macroscopic levels. The proposed neuro-cognitive model builds a map reconstruction using the bottom-up facial environmental information and top-down map information, and generates intention towards the final goal from the macroscopic level. I proposed a neuro-cognitive model integrates the sensory-motor coordination model with environmental knowledge as a cognitive map using the bottom-up facial environmental information and top-down map information. In this chapter, I develop the localization and mapping to recognize a current facing situation by the topological structure information from lower level, composed as 3D vector position of nodes, edges, and 3D surface vectors of nodes. The robot generates intention towards the final goal from the macroscopic level. Furthermore, to integrate the relationship between macroscopic level of behavior commands and locomotion performance, I develop the bahavior coordination module composed as behavior learning strategy for omnidirectional movement. I also build the tree-structured learning model to deal with the complexity of the neural structure of CPG. The proposed model has been tested for omnidirectional movement in biped and quadruped robot. Furthermore, the proposed neuro-cognitive model has also been implemented for robot climbing behavior, performing a horizontal-vertical-horizontal movement. The detail explanation every module is shown as follow:

1. This section shows localization and mapping module integrate the external information from microscopic level to the cognitive map building. The topological structure with label information is the input of the module. There are two steps for building the localization and mapping, which is Confidence Node collection and Surface matching. The nodes of map is composed as (3D position, 3D surface vector, label, and magnitude

of surface vector). The nodes of GNG is composed as (3D posisition, 3D surface vector, label, and magnitude of surface vector). To show the effectiveness of the proposed module, I tested the module into computer simulation. The module can reconstruct the map and localize the robot position simultaneously.

2. This section proposed the learning strategy for generating omni-directional movement behavior. The module process the information from macroscopic level and generate neural structure for the locomotion generator at microscopic level. To achieve the robot optimatily, movement provision and resulted energy are calculated as the fitness calculation. In this optimization, the behavior solutions was not deserved well, however, by increasing the number of generation, good diversity can be acquired. From the experiment shown in Section [], by using this proposed learning model with cross validation, omni-directional movement behavior can be generated. The model can generate sagittal movement from 0 m/s - 25 m/s, coronal movement from -5 m/s - 5 m/s, and turning speed $(-\pi/2)^0/s$ - $(\pi/2)^0/s$. In this proposed method, walking behavior references generation took a long time, one reference behavior requires one optimization process. This is one of the problems which should be solved in next development. Walking behavior references is used as the training data of the learning model. Separated MLP strategy is successfully approaching the relationship between input and output of walking behavior. Training iteration and learning process take a long time, and also require higher performance of learning model, such deep learning model that will be the future works.

3. This section proposed the module for evolving the sensory-motor integration in a neural based legged robot locomotion. The learning module integrates the behavior coordination at mesoscopic level and neural model in microscopic level. A new tree structure-based model of neuron interconnection structure was proposed to simplify the representation of neural structure. Bacterial programming is used as the optimization technique. Based on the optimization, the proposed tree structure can simplify the process and successfully generate stable walking in biped robot and quadruped robot as well. From the experimental result in Section 5.3.4, the proposed module can also generate the sensory-motor neuron interconnection. The signal generated by motor neurons is adaptively changed depending on the sensory condition. The tree structure model can be applied for simplifying the representation of sensory-motor interconnection in any multi-legged robots, it is not limited to biped and quadruped robots.

6.2 Future Prospect

In the future, I will improve the neuro-cognitive model especially in the proposed cognitive model. Furthermore, the implementation of the proposed model will be extended for more wider area with higher durability and applicability.

References

[1] Development Kit Brief Camboard Pico flexx. https://pmdtec.com/picofamily/wp-content/uploads/2018/03/PMD_DevKit_Brief_CB_pico_flexx_CE_V0218-1.pdf.

[2] Locomotion concept. https://www.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous_mobile_robots/spring-2017/Locomotion%20Concepts.pdf. Accessed: 2017-12-22.

[3] Open {D}ynamics {E}ngine. \url{http://ode.org}.

[4] E. Ackerman. Boston Dynamics' SpotMini Is All Electric, Agile, and Has a Capable Face-Arm. *IEEE SPECTRUM*, 2016.

[5] M. Ahmed, M. Khan, M. Billah, and S. Farhana. *Walking hexapod robot in disaster recovery: developing algorithm for terrain negotiation and navigation*. INTECH Open Access Publisher, 2010.

[6] J. J. Alcaraz-Jimenez, D. Herrero-Perez, and H. Martınez-Barberá. Motion planning for omnidirectional dynamic gait in humanoid soccer robots. *Journal of Physical Agents*, 5(1):25–34, 2011.

[7] H. Amano et al. Development of vertically moving robot with gripping handrails for fire fighting. In *Proc. of Intl. Conf. on Intelligent Robots and Systems.*, volume 2, pages 661–667, 2001.

[8] N. Anantrasirichai, K. A. J. Daniels, J. F. Burn, I. D. Gilchrist, and D. R. Bull. Fixation Prediction and Visual Priority Maps for Biped Locomotion. *IEEE transactions on cybernetics*, 48(8):2294–2306, 2017.

[9] R. A. Andersen, J. W. Burdick, S. Musallam, B. Pesaran, and J. G. Cham. Cognitive neural prosthetics. *Trends in cognitive sciences*, 8(11):486–493, 2004.

[10] A. Angelopoulou and Others. Fast 2D/3D object representation with growing neural gas. *Neural Computing and Applications*, 29(10):903–919, may 2018.

[11] D. M. Armstrong and T. Drew. Discharges of pyramidal tract and other motor cortical neurones during locomotion in the cat. *The Journal of Physiology*, 346(1):471–495, 1984.

[12] D. M. Armstrong and T. Drew. Locomotor-related neuronal discharges in cat motor cortex compared with peripheral receptive fields and evoked movements. *The Journal of physiology*, 346(1):497–517, 1984.

[13] D. M. Armstrong and T. Drew. Topographical localization in the motor cortex of the cat for somatic afferent responses and evoked movements. *The Journal of physiology*, 350(1):33–54, 1984.

[14] F. Asadi, M. Khorram, and S. A. A. Moosavian. CPG-based gait transition of a quadruped robot. In *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pages 210–215. IEEE, 2015.

[15] J. Aulinas, Y. R. Petillot, J. Salvi, and X. Lladó. The slam problem: a survey. *CCIA*, 184(1):363–371, 2008.

[16] S. N. Baker. The primate reticulospinal tract, hand function and functional recovery. *The Journal of physiology*, 589(23):5603–5612, 2011.

[17] K. Balázs, J. Botzheim, and L. T. Kóczy. Hierarchical Fuzzy System Modeling by Genetic and Bacterial Programming Approaches. In *Proc. of the IEEE World Congress on Computational Intelligence, WCCI 2010*, pages 1866–1871, Barcelona, Spain, jul 2010.

[18] J. H. Barron-Zambrano, C. Torres-Huitzil, and B. Girau. Perception-driven adaptive {CPG}-based locomotion for hexapod robots. *Neurocomputing*, 170:63–78, 2015.

[19] A. G. Baydin. Evolution of Central Pattern Generators for the control of a five-link bipedal walking mechanism. *Paladyn journal of behavioral robotics*, 3(1):45–53, 2012.

[20] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter. Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326, 2018.

[21] I. N. Beloozerova and M. G. Sirota. The role of the motor cortex in the control of accuracy of locomotor movements in the cat. *The Journal of Physiology*, 461(1):1–25, 1993.

[22] I. N. Beloozerova, M. G. Sirota, H. A. Swadlow, G. N. Orlovsky, L. B. Popova, and T. G. Deliagina. Activity of different classes of neurons of the motor cortex during postural corrections. *Journal of Neuroscience*, 23(21):7844–7853, 2003.

[23] I. N. Beloozerova, M. G. Sirota, H. A. Swadlow, G. N. Orlovsky, L. B. Popova, and T. G. Deliagina. Activity of different classes of neurons of the motor cortex during postural corrections. *Journal of Neuroscience*, 23(21):7844–7853, 2003.

[24] D. Belter, P. Labecki, and P. Skrzypczynski. On-Board Perception and Motion Planning for Legged Locomotion over Rough Terrain. In *ECMR*, pages 195–200, 2011.

[25] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[26] M. Bizzarri, A. Giuliani, A. Pensotti, E. Ratti, and M. Bertolaso. Co-emergence and collapse: The mesoscopic approach for conceptualizing and investigating the functional integration of organisms. *Frontiers in Physiology*, 10:924, 2019.

[27] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. Gonzalez. Toward a unified bayesian approach to hybrid metric–topological slam. *IEEE Transactions on Robotics*, 24(2):259–270, 2008.

[28] M. Bloesch, T. Laidlow, R. Clark, S. Leutenegger, and A. J. Davison. Learning meshes for dense visual slam. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5855–5864, 2019.

[29] R. Bogdan Rusu, A. Sundaresan, B. Morisset, K. Hauser, M. Agrawal, J.-C. Latombe, and M. Beetz. Leaving Flatland: Efficient real-time three-dimensional perception and motion planning. *Journal of Field Robotics*, 26(10):841–862, 2009.

[30] J. Botzheim, C. Cabrita, L. T. Koczy, and A. E. Ruano. Genetic and Bacterial Programming for {B}-Spline Neural Networks Design. *Journal of Advanced Computational Intelligence Informatics*, 11(2):220–231, 2007.

[31] J. Botzheim, Y. Toda, and N. Kubota. Bacterial memetic algorithm for offline path planning of mobile robots. *Memetic Comp.*, 4(1):73–86, 2012.

[32] K. Bouyarmane et al. Exploring humanoid robots locomotion capabilities in virtual disaster response scenarios. In *Proc. of IEEE-RAS Intl. Conf. on Humanoid Robots*, pages 337–342, 2012.

[33] T. G. Brown. The intrinsic factors in the act of progression in the mammal. *Proceedings of the Royal Society of London. Series B, containing papers of a biological character*, 84(572):308–319, 1911.

[34] T. G. Brown. On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *The Journal of physiology*, 48(1):18–46, 1914.

[35] L. Bruzzone and G. Quaglia. Locomotion systems for ground mobile robots in unstructured environments. *Mechanical sciences*, 3(2):49–62, 2012.

[36] F. Caliebe, J. Häußler, P. Hoffmann, M. Illert, J. Schirrmacher, and E. Wiedemann. Cat distal forelimb joints and locomotion: an X-ray study. *European Journal of Neuroscience*, 3(1):18–31, 1991.

[37] M. Camurri, S. Bazeille, D. G. Caldwell, and C. Semini. Real-time depth and inertial fusion for local SLAM on dynamic legged robots. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 259–264. IEEE, 2015.

[38] H. J. Chang, C. G. Lee, Y.-H. Lu, and Y. C. Hu. P-slam: Simultaneous localization and mapping with environmental-structure prediction. *IEEE Transactions on Robotics*, 23(2):281–293, 2007.

[39] G. Chechik, I. Meilijson, and E. Ruppin. Synaptic pruning in development: a computational account. *Neural Comp.*, 10(7):1759–1777, 1998.

[40] G. Chechik, I. Meilijson, and E. Ruppin. Neuronal regulation: A mechanism for synaptic pruning during brain maturation. *Neural Comp.*, 11(8):2061–2080, 1999.

[41] W. S. Chee and J. Teo. Using a co-evolutionary approach to automatically generate vertical undulation and lateral rolling motions for snake-like modular robot. In *Proc. of the 2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 236–241, 2014.

[42] H.-C. Chen et al. Stepping over obstacles: dividing attention impairs performance of old more than young adults. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 51(3):M116–M122, 1996.

[43] X. Chen and Others. 3D model based ladder tracking using vision and laser point cloud data. In *Proc. of Intl. Conf. on Robotics and Biomimetics*, pages 1365–1370, dec 2015.

[44] W. H. Chin and Others. A neuro-based network for on-line topological map building and dynamic path planning. In *Intl. Joint Conf. on Neural Networks*, pages 2805–2810, 2017.

[45] W. H. Chin and Others. An Incremental Episodic Memory Framework for Topological Map Building. In *Intl. Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, pages 322–327, 2018.

[46] K. M. I. Chu, S. H. Seto, I. N. Beloozerova, and V. Marlinski. Strategies for obstacle avoidance during walking in the cat. *Journal of neurophysiology*, 118(2):817–831, 2017.

[47] A. Clark and R. Grush. Towards a cognitive robotics. *Adaptive Behavior*, 7(1):5–16, 1999.

[48] D. M. Cole and P. M. Newman. Using laser range data for 3d slam in outdoor environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1556–1563. IEEE, 2006.

[49] A. Crespi, K. Karakasiliotis, A. Guignard, and A. J. Ijspeert. Salamandra robotica II: an amphibious robot to study salamander-like swimming and walking gaits. *IEEE Transactions on Robotics*, 29(2):308–320, 2013.

[50] R. Deits and R. Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *2014 IEEE-RAS international conference on humanoid robots*, pages 279–286. IEEE, 2014.

[51] J.-E. Deschaud. Imls-slam: scan-to-model matching based on 3d data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2480–2485. IEEE, 2018.

[52] J. Diebel, K. Reutersward, S. Thrun, J. Davis, and R. Gupta. Simultaneous localization and mapping with active stereo vision. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 4, pages 3436–3443. IEEE, 2004.

[53] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[54] S. Dogru and L. Marques. Energy efficient coverage path planning for autonomous mobile robots on 3D terrain. In *Proc. of IEEE Int. Conf. on Auton. Robot Syst. and Competitions*, pages 118–123, April 2015.

[55] S. Dogru and L. Marques. Towards fully autonomous energy efficient coverage path planning for autonomous mobile robots on 3D terrain. In *Proc. of European Conf. on Mobile Robots*, pages 1–6, Sept 2015.

[56] T. Drew, W. Jiang, B. Kably, and S. Lavoie. Role of the motor cortex in the control of visually triggered gait modifications. *Canadian journal of physiology and pharmacology*, 74(4):426–442, 1996.

[57] T. Drew and S. Rossignol. A kinematic and electromyographic study of cutaneous reflexes evoked from the forelimb of unrestrained walking cats. *Journal of neurophysiology*, 57(4):1160–1184, 1987.

[58] R. Dube and Others. SegMatch: Segment based place recognition in 3D point clouds. In *Proc. of Intl. Conf. on Robotics and Automation*, pages 5266–5272, may 2017.

[59] J. Duysens and H. Van de Crommert. Neural control of locomotion; the central pattern generator from cats to humans. *Gait Posture*, 7(2):131–141, Mar 1998.

[60] A. Eilering and Others. Identifying support surfaces of climbable structures from 3D point clouds. In *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, pages 6226–6231, may 2014.

[61] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng. An empirical exploration of a neural oscillator for biped locomotion control. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 3036–3042. IEEE.

[62] A. W. M. ENGLISH. An electromyographic analysis of forelimb muscles during overground stepping in the cat. *Journal of Experimental Biology*, 76(1):105–122, 1978.

[63] A. Espinal, H. Rostro-Gonzalez, M. Carpio, E. I. Guerra-Hernandez, M. Ornelas-Rodriguez, H. J. Puga-Soberanes, M. A. Sotelo-Figueroa, and P. Melin. Quadrupedal robot locomotion: a biologically inspired approach and its hardware implementation. *Computational intelligence and neuroscience*, 2016, 2016.

[64] B. R. Fajen and W. H. Warren. Behavioral dynamics of steering, obstable avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):343, 2003.

[65] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, and Others. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, 32(2):229–254, 2015.

[66] J. J. Feher. *Quantitative human physiology: an introduction*. Academic press, 2017.

[67] S. Feng et al. Optimization based full body control for the atlas robot. In *Proc. of Intl. Conf. on Humanoid Robots*, pages 120–127. IEEE, 2014.

[68] D. Ferguson and A. Stentz. Field D*: An interpolation-based path planner and replanner. In *Robot. Res.*, pages 239–253. 2007.

[69] C. Ferreira, P. Silva, J. Andre, C. P. Santos, and L. Costa. Genetic programming applied to biped locomotion control with sensory information. In *Proc. of the 2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 01, pages 53–62, sep 2014.

[70] J. Folkesson and H. I. Christensen. Closing the loop with graphical slam. *IEEE Transactions on Robotics*, 23(4):731–741, 2007.

[71] J. Folkesson, P. Jensfelt, and H. I. Christensen. The m-space feature representation for slam. *IEEE Transactions on Robotics*, 23(5):1024–1035, 2007.

[72] K. T. Forbush, M. Shaw, M. A. Graeber, L. Hovick, V. J. Meyer, D. J. Moser, J. Bayless, D. Watson, and D. W. Black. Neuropsychological characteristics and personality traits in pathological gambling. *CNS spectrums*, 13(4):306–315, 2008.

[73] S. Frintrop, E. Rome, and H. I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception (TAP)*, 7(1):6, 2010.

[74] B. Fritzke. Unsupervised clustering with growing cell structures. In *in Proc. of Intl. Joint Conf.on Neural Networks*, volume 2, pages 531–536, 1991.

[75] B. Fritzke. A growing neural gas network learns topologies. In *Advances in neural information processing systems*, pages 625–632, 1995.

[76] S. Fujii et al. Ladder climbing control for limb mechanism robot asterisk. In *Proc. of Intl. Conf. on Robotics and Automation*, pages 3052–3057, 2008.

[77] T. Fukuda, Y. Hasegawa, K. Sekiyama, and T. Aoyama. *Multi-Locomotion Robotic Systems: New Concepts of Bio-inspired Robotics*, volume 81. Springer, 2012.

[78] Y. Fukuoka and H. Kimura. Dynamic locomotion of a biomorphic quadruped tekkenrobot using various gaits: walk, trot, free-gait and bound. *Applied Bionics and Biomec.*, 6(1):63–71, 2009.

[79] L.-F. Gao, Y.-X. Gai, and S. Fu. Simultaneous localization and mapping for autonomous mobile robots using binocular stereo vision system. In *2007 International Conference on Mechatronics and Automation*, pages 326–330. IEEE, 2007.

[80] T. Geijtenbeek, M. van de Panne, and A. F. van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):206, 2013.

[81] M. Geisert et al. Contact planning for the anymal quadruped robot using an acyclic reachability-based planner. In *Annual Conf. Towards Autonomous Robotic Systems*, pages 275–287. Springer, 2019.

[82] H. Geyer and H. Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering*, 18(3):263–273, 2010.

[83] H. Geyer, A. Seyfarth, and R. Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1529):2173–2183, 2003.

[84] E. Gibson and N. Rader. Attention. In *Attention and cognitive development*, pages 1–21. Springer, 1979.

[85] J. J. Gibson. The theory of affordances. *Hilldale, USA*, 1:2, 1977.

[86] J. J. Gibson. The theory of proprioception and its relation to volition: An attempt at clarification. *Reasons for realism: Selected essays of James J. Gibson*, pages 385–388, 1982.

[87] J. J. Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[88] R. Glasius, A. Komoda, and S. C. Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Netw.*, 8(1):125–133, 1995.

[89] K. Glette and M. Hovin. Evolution of artificial muscle-based robotic locomotion in PhysX. In *in Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 1114–1119, oct 2010.

[90] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992.

[91] R. J. Griffin et al. Footstep planning for autonomous walking over rough terrain. *arXiv preprint arXiv:1907.08673*, 2019.

[92] E. Grinke, C. Tetzlaff, F. Wörgötter, and P. Manoonpong. Synaptic plasticity in a recurrent neural network for versatile and adaptive behaviors of a walking robot. *Frontiers in neurorobotics*, 9:11, 2015.

[93] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007.

[94] S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Netw.*, 1(1):17 – 61, 1988.

[95] X. Gu, L. Zhang, and D. Yu. Delay PCNN and its application for optimization. In *Proc. of Int. Symp. on Neural Netw.*, pages 413–418, 2004.

[96] P. Gyawali and J. McGough. Simulation of detecting and climbing a ladder for a humanoid robot. In *Proc. of Intl. Conf. on Electro-Information Technology*, pages 1–6, may 2013.

[97] T. Haarnoja. Dynamic modeling and velocity control for limit cycle walking, 2010.

[98] K. Hashimoto et al. Warec-1a four-limbed robot having high locomotion ability with versatility in locomotion styles. In *Proc. of Intl. Symp. on Safety, Security and Rescue Robotics*, pages 172–178, 2017.

[99] K. Hashimoto, T. Matsuzawa, X. Sun, T. Fujiwara, X. Wang, Y. Konishi, N. Sato, T. Endo, F. Matsuno, N. Kubota, and Others. WAREC-1–a four-limbed robot with advanced locomotion and manipulation capabilities. In *Disaster Robotics*, pages 327–397. Springer, 2019.

[100] K. Hauser et al. Non-gaited humanoid locomotion planning. In *Proc. of Intl. Conf. on Humanoid Robots*, pages 7–12, 2005.

[101] I. Havoutis, J. Ortiz, S. Bazeille, V. Barasuol, C. Semini, and D. G. Caldwell. Onboard perception-based trotting and crawling with the hydraulic quadruped robot (HyQ). In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6052–6057. IEEE, 2013.

[102] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *The Int. Journal of Robot. Res.*, 31(5):647–663, 2012.

[103] C. Hernández and J. A. Baier. Making A* run faster than D*-lite for path-planning in partially known terrain. 2014.

[104] J. R. Hinchliffe and D. R. Johnson. *The development of the vertebrate limb: an approach through experiment, genetics, and evolution*. Oxford University Press, USA, 1980.

[105] A. Hinton. *Understanding context: Environment, language, and information architecture*. " O'Reilly Media, Inc.", 2014.

[106] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and Others. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[107] D. G. Hobbelen and M. Wisse. Controlling the walking speed in limit cycle walking. *The International Journal of Robotics Research*, 27(9):989–1005, 2008.

[108] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500, 1952.

[109] M. Hoffmann, N. Schmidt, K. Nakajima, F. Iida, R. Pfeifer, and A. Ishiguro. Perception, motor learning, and speed adaptation exploiting body dynamics: case studies in a quadruped robot. In *International Symposium on Adaptive Motion of Animals and Machines (AMAM 2011)*, number 5, pages 39–40. Tohoku University, 2011.

[110] J. H. Holland. *Adaption in Natural and Artificial Systems*. The MIT Press, 1992.

[111] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM review*, 48(2):207–304, 2006.

[112] Y.-D. Hong and B. Lee. Real-time feasible footstep planning for bipedal robots in three-dimensional environments using particle swarm optimization. *IEEE/ASME Tran. on Mechatronics*, 2019.

[113] Y.-D. Hong, C.-S. Park, and J.-H. Kim. Stable bipedal walking with a vertical center-of-mass motion by an evolutionary optimized central pattern generator. *IEEE Transactions on Industrial Electronics*, 61(5):2346–2355, May 2014.

[114] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[115] K. Hosoda and M. Asada. Adaptive visual servoing for various kinds of robot systems. In *Experimental Robotics V*, pages 546–558. Springer, 1998.

[116] S. Huang and G. Dissanayake. Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on robotics*, 23(5):1036–1049, 2007.

[117] P. R. Huttenlocher. Synaptic density in human frontal cortexdevelopmental changes and effects of aging. *Brain Res.*, 163(2):195–205, 1979.

[118] D. J. Hyun, S. Seok, J. Lee, and S. Kim. High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *The International Journal of Robotics Research*, 33(11):1417–1445, 2014.

[119] A. Ijspeert and J.-M. Cabelguen. Gait Transition from Swimming to Walking: Investigation of Salamander Locomotion Control Using Nonlinear Oscillators. In H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witte, editors, *Adaptive Motion of Animals and Machines*, pages 177–188. Springer Tokyo, 2006.

[120] A. J. Ijspeert. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 85(5):331–348, 2001.

[121] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653, 2008.

[122] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review, 2008.

[123] A. J. Ijspeert, A. Crespi, and J.-M. Cabelguen. Simulation and robotics studies of salamander locomotion - Applying neurobiological principles to the control of locomotion in robots. *Neuroinformatics*, 3(3):171–195, 2005.

[124] M. Ilyas and Others. Staircase Recognition and Localization using Convolution Neural Network (CNN) for Cleaning Robot Application. *Preprints*, 2018.

[125] B. Inden, Y. Jin, R. Haschke, and H. Ritter. Exploiting inherent regularity in control of multilegged robot locomotion by evolving neural fields. In *Proc. of the 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 401–408, oct 2011.

[126] A. Ishiguro, A. Fujii, and P. E. Hotz. Neuromodulated control of bipedal locomotion using a polymorphic CPG circuit. *Adaptive Behavior*, 11(1):7–17, 2003.

[127] L. Itti, N. Dhavale, and F. Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation VI*, volume 5200, pages 64–79. International Society for Optics and Photonics, 2003.

[128] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.

[129] M. Kaess, A. Ranganathan, and F. Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[130] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1620–1626. IEEE, 2003.

[131] M. Kanazawa et al. Robust vertical ladder climbing and transitioning between ladder and catwalk for humanoid robots. In *Proc. of Intl. Conf. on Intelligent Robots and Systems*, pages 2202–2209, 2015.

[132] P. Karkowski and M. Bennewitz. Prediction maps for real-time 3d footstep planning in dynamic environments. In *2019 Intl. Conf. on Robotics and Autom. (ICRA)*, pages 2517–2523. IEEE, 2019.

[133] R. Khusainov, I. Shimchik, I. Afanasyev, and E. Magid. 3D modelling of biped robot locomotion with walking primitives approach in Simulink environment. In *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, pages 287–304. Springer, 2016.

[134] C. Kim, R. Sakthivel, and W. K. Chung. Unscented fastslam: a robust and efficient solution to the slam problem. *IEEE Transactions on robotics*, 24(4):808–820, 2008.

[135] J.-Y. Kim, I.-W. Park, and J.-H. Oh. Experimental realization of dynamic walking of the biped humanoid robot {KHR-2} using zero moment point feedback and inertial measurement. *Advanced Robotics*, 20(6):707–736, 2006.

[136] H. Kimura, S. Akiyama, and K. Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous robots*, 7(3):247–258, 1999.

[137] C. Kisner and L. A. Colby. Therapeutic exercise. *Foundations and techniques*, 4, 2002.

[138] S. Kohlbrecher, A. Stumpf, and O. Von Stryk. Grid-based occupancy mapping and automatic gaze control for soccer playing humanoid robots. In *Proc. 6th Workshop on Humanoid Soccer Robots at the*, 2011.

[139] T. Kohonen and S.-O. Maps. Springer series in information sciences. *Self-organizing maps*, 30, 1995.

[140] J. Z. Kolter, Y. Kim, and A. Y. Ng. Stereo vision and terrain modeling for quadruped robots. In *2009 IEEE International Conference on Robotics and Automation*, pages 1557–1564. IEEE, 2009.

[141] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng. A control architecture for quadruped locomotion over rough terrain. In *2008 IEEE International Conference on Robotics and Automation*, pages 811–818. IEEE, 2008.

[142] H. S. Koppula and Others. Semantic labeling of 3d point clouds for indoor scenes. In *Advances in neural information processing systems*, pages 244–252, 2011.

[143] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.

[144] V. Kroumov and J. Yu. 3D path planning for mobile robots using annealing neural network. In *Proc. of Int. Conf. on Networking, Sensing and Control*, pages 130–135, March 2009.

[145] 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*, 40(3):429–455, 2016.

[146] T. Kuyucu, I. Tanev, and K. Shimohara. Genetic transposition inspired incremental genetic programming for efficient coevolution of locomotion and sensing of simulated snake-like robot. In *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 439–446, 2011.

[147] N. Kwak, I.-K. Kim, H.-C. Lee, and B.-H. Lee. Adaptive prior boosting technique for the efficient sample size in fastslam. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 630–635. IEEE, 2007.

[148] T. Lam and K. G. Pearson. Proprioceptive modulation of hip flexor activity during the swing phase of locomotion in decerebrate cats. *Journal of Neurophysiology*, 86(3):1321–1332, 2001.

[149] T. Lam and K. G. Pearson. The role of proprioceptive feedback in the regulation and adaptation of locomotor activity. In *Sensorimotor Control of Movement and Posture*, pages 343–355. Springer, 2002.

[150] M. Laurent and J. A. Thomson. The role of visual information in control of a constrained locomotor task. *Journal of motor behavior*, 20(1):17–37, 1988.

[151] S. Lavoie and T. Drew. Discharge characteristics of neurons in the red nucleus during voluntary gait modifications: a comparison with the motor cortex. *Journal of neurophysiology*, 88(4):1791–1814, 2002.

[152] Y. Lee, M. S. Park, T. Kwon, and J. Lee. Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics (TOG)*, 33(6):218, 2014.

[153] Y.-J. Lee and S. Hirose. Three-legged walking for fault-tolerant locomotion of demining quadruped robots. *Advanced Robotics*, 16(5):415–426, 2002.

[154] C. T. Leonard. The neuroscience of human movement. Mosby-Year Book. *Inc., St. Louis*, 1998.

[155] M. A. Lewis, A. H. Fagg, and A. Solidum. Genetic programming approach to the construction of a neural network for control of a walking robot. In *Proc. of the 1992 IEEE International Conference on Robotics and Automation*, pages 2618—-2623 vol.3, may 1992.

[156] X. Li, Y. Ma, and X. Feng. Self-adaptive autowave pulse-coupled neural network for shortest-path problem. *Neurocomp.*, 115:63–71, 2013.

[157] R. L. Lieber, B. M. Fazeli, and M. J. Botte. Architecture of selected wrist flexor and extensor muscles. *The Journal of hand surgery*, 15(2):244–250, 1990.

[158] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun. Anytime dynamic A*: An anytime, replanning algorithm. 2005.

[159] M. Likhachev, G. J. Gordon, and S. Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Adv. in Neural Inform. Process. Syst.*, 2003.

[160] C. Liu et al. Cpg driven locomotion control of quadruped robot. In *Intl. Conf. on Systems, Man and Cyb.*, pages 2368–2373, 2009.

[161] C. Liu et al. Adaptive walking control of quadruped robots based on central pattern generator (cpg) and reflex. *Journal of Control Theory and Applications*, 11(3):386–392, 2013.

[162] C. Liu, D. Wang, and Q. Chen. Central pattern generator inspired control for adaptive walking of biped robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(5):1206–1215, 2013.

[163] C. Liu, L. Xia, C. Zhang, and Q. Chen. Multi-layered CPG for adaptive walking of quadruped robots. *Journal of Bionic Engineering*, 15(2):341–355, 2018.

[164] G. Liu, Z. Qiu, H. Qu, and L. Ji. Computing k shortest paths using modified pulse-coupled neural network. *Neurocomp.*, 149:1162–1176, 2015.

[165] B. P. Livingston and T. R. Nichols. Effects of reinnervation of the triceps brachii on joint kinematics and electromyographic patterns of the feline forelimb during level and upslope walking. *Cells Tissues Organs*, 199(5-6):405–422, 2014.

[166] D. Logan, T. Kiemel, N. Dominici, G. Cappellini, Y. Ivanenko, F. Lacquaniti, and J. J. Jeka. The many roles of vision during walking. *Experimental brain research*, 206(3):337–350, 2010.

[167] Z. Lu et al. Transition motion from ladder climbing to brachiation for multi-locomotion robot. In *Proc of Intl. Conf. on Mechatronics and Automation*, pages 1916–1921, 2009.

[168] Z. Lu et al. Transition motion from ladder climbing to brachiation with optimal load-allocation control. *Advanced Robotics*, 26(8-9):1075–1098, 2012.

[169] J. Luo et al. Robust ladder-climbing with a humanoid robot with application to the darpa robotics challenge. In *Proc. of Intl. Conf. on Robotics and Automation*, pages 2792–2798, 2014.

[170] W. Maass. Computing with spiking neurons. In *Pulsed neural networks*, pages 55–85. MIT Press, 1999.

[171] O. A. V. Magana et al. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *IEEE Robotics and Autom. Letters*, 4(2):2140–2147, 2019.

[172] D. Maier, A. Hornung, and M. Bennewitz. Real-time navigation in 3D environments based on depth camera data. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 692–697. IEEE, 2012.

[173] D. Maier, C. Lutz, and M. Bennewitz. Integrated perception, mapping, and footstep planning for humanoid navigation among 3d obstacles. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2658–2664. IEEE, 2013.

[174] I. R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[175] P. Manoonpong, F. Pasemann, and F. Wörgötter. Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines. *Robotics and Autonomous Systems*, 56(3):265–288, 2008.

[176] D. Marbach and A. J. Ijspeert. Online optimization of modular robot locomotion. In *Proc. of the 2005 IEEE International Conference on Mechatronics and Automation*, volume 1, pages 248—-253 Vol. 1, jul 2005.

[177] L. Martins-Filho and R. Prajoux. Locomotion control of a four-legged robot embedding real-time reasoning in the force distribution. *Robotics and Auton. Syst.*, 32(4):219 – 235, 2000.

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

[179] V. Matos and C. P. Santos. Omnidirectional locomotion in a quadruped robot: A cpg-based approach. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3392–3397. IEEE, 2010.

[180] V. Matos and C. P. Santos. Towards goal-directed biped locomotion: Combining {CPG}s and motion primitives. *Robotics and Autonomous Systems*, 62(12):1669–1690, 2014.

[181] K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics*, 52:367–376, 1985.

[182] K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56:345–353, 1987.

[183] K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56:345–353, 1987.

[184] T. Matsuzawa, A. Koizumi, K. Hashimoto, X. Sun, S. Hamamoto, T. Teramachi, S. Kimura, N. Sakai, and A. Takanishi. Crawling gait for four-limbed robot and simulation on uneven terrain. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1270–1275. IEEE, 2016.

[185] D. Maturana and S. Scherer. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *Proc. of Intl. Conf. on Intelligent Robots and Systems*, pages 922–928, 2015.

[186] C. Maufroy, H. Kimura, and K. Takase. Towards a general neural controller for quadrupedal locomotion. *Neural Networks*, 21(4):667–681, 2008.

[187] D. A. McCrea and I. A. Rybak. Organization of mammalian locomotor rhythm and pattern generation. *Brain research reviews*, 57(1):134–146, 2008.

[188] D. A. McVea, J. M. Donelan, A. Tachibana, and K. G. Pearson. A role for hip position in initiating the swing-to-stance transition in walking cats. *Journal of neurophysiology*, 94(5):3497–3508, 2005.

[189] S. Miyakoshi, G. Taga, Y. Kuniyoshi, and A. Nagakubo. Three dimensional bipedal stepping motion using neural oscillators-towards humanoid motion in the real world. In *Proc. of the 1998 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, volume 1, pages 84—-89 vol.1, oct 1998.

[190] K. Miyashita, S. Ok, and K. Hase. Evolutionary generation of human-like bipedal locomotion. *Mechatronics*, 13(8-9):791–807, 2003.

[191] N. Morales, J. Toledo, and L. Acosta. Path planning using a multiclass support vector machine. *Applied Soft Comp.*, 43:498–509, 2016.

[192] Y. Mori, K. Takayama, T. Adachi, S. Omote, and T. Nakamura. Feasibility study on an excavation-type demining robot. *Auton. Robots*, 18(3):263–274, 2005.

[193] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen. Cognitive control signals for neural prosthetics. *Science*, 305(5681):258–262, 2004.

[194] K. Nakada, T. Asai, and Y. Amemiya. An analog CMOS central pattern generator for interlimb coordination in quadruped locomotion. *IEEE Transactions on Neural Networks*, 14(5):1356–1365, 2003.

[195] G. C. Nandi, V. B. Semwal, M. Raj, and A. Jindal. Modeling bipedal locomotion trajectories using hybrid automata. In *2016 IEEE region 10 conference (TENCON)*, pages 1013–1018. IEEE, 2016.

[196] J. Nassour, P. Henaff, F. Benouezdou, and G. Cheng. Experience-based learning mechanism for neural controller adaptation: Application to walking biped robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 2616–2621, Oct 2009.

[197] J. Nassour, P. Henaff, F. B. Ouezdou, and G. Cheng. Multi-layered multi-pattern {CPG} for adaptive locomotion of humanoid robots. *Biological Cybernetics*, pages 291–303, 2014.

[198] N. E. Nawa and T. Furuhashi. Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Transactions on Fuzzy Systems*, 7(5):608–616, 1999.

[199] A. Nguyen and B. Le. Contextual labeling 3D point clouds with conditional random fields. In *Proc. of Asian Conf. on Intelligent Information and Database Systems*, pages 581–590, 2014.

[200] K. Ni and F. Dellaert. Multi-level submap based slam using nested dissection. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2558–2565. IEEE, 2010.

[201] N. J. Nilsson. *Principles of artificial intelligence*. Springer-Verlag Berlin Heidelberg, 2014.

[202] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6d slam3d mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722, 2007.

[203] K. Ohno, T. Nomura, and S. Tadokoro. Real-time robot trajectory estimation and 3d map construction using 3d camera. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5279–5285. IEEE, 2006.

[204] S. Ok, K. Miyashita, and K. Hase. Evolving bipedal locomotion with genetic programming - a preliminary report. In *Proc. of the 2001 Congress on Evolutionary Computation*, volume 2, pages 1025—-1032 vol. 2, 2001.

[205] G. N. Orlovski\u\i, T. G. Deliagina, and S. Grillner. *Neuronal control of locomotion: from mollusc to man*. Oxford University Press, 1999.

[206] G. N. Orlovsky. The effect of different descending systems on flexor and extensor activity during locomotion. *Brain research*, 40(2):359–372, 1972.

[207] D. Ortiz-Arroyo. A hybrid 3D path planning method for UAVs. In *Proc. of Workshop on Res., Edu. and Develop. of Unmanned Aerial Systems*, pages 123–132, Nov 2015.

[208] S. Orts-Escolano and Others. 3D Surface Reconstruction of Noisy Point Clouds Using Growing Neural Gas: 3D Object/Scene Reconstruction. *Neural Processing Letters*, 43(2):401–423, apr 2016.

[209] A. Palomer, P. Ridao, and D. Ribas. Inspection of an underwater structure using point-cloud slam with an auv and a laser scanner. *Journal of Field Robotics*, 36(8):1333–1344, 2019.

[210] C.-S. Park, Y.-D. Hong, and J.-H. Kim. Evolutionary-optimized central pattern generator for stable modifiable bipedal walking. *IEEE/ASME Transactions on Mechatronics*, 19(4):1374–1383, Aug 2014.

[211] C.-S. Park, Y.-D. Hong, and J.-H. Kim\. Full-body joint trajectory generation using an evolutionary central pattern generator for stable bipedal walking. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 160–165, oct 2010.

[212] H. W. Park, P. M. Wensing, and S. Kim. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In *2015 Robotics: Science and Systems Conference, RSS 2015*. MIT Press Journals, 2015.

[213] G. A. Parker and J. M. Smith. Optimality theory in evolutionary biology. *Nature*, 348(6296):27–33, 1990.

[214] A. E. Patla. Understanding the roles of vision in the control of human locomotion. *Gait & posture*, 5(1):54–69, 1997.

[215] K. Pearson, Ö. Ekeberg, and A. Büschges. Assessing sensory function in locomotor systems using neuro-mechanical simulations. *Trends in neurosciences*, 29(11):625–631, 2006.

[216] K. G. Pearson. Generating the walking gait: role of sensory feedback. In *Prog. in brain rsrch.*, volume 143, pages 123–129. Elsevier, 2004.

[217] K. G. Pearson. Role of sensory feedback in the control of stance duration in walking cats. *Brain research reviews*, 57(1):222–227, 2008.

[218] R. Pfeifer and J. Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[219] F. Qian and D. Goldman. Anticipatory control using substrate manipulation enables trajectory control of legged locomotion on heterogeneous granular media. In *Micro-and Nanotechnology Sensors, Systems, and Applications VII*, volume 9467, page 94671U. International Society for Optics and Photonics, 2015.

[220] H. Qu, S. X. Yang, A. R. Willms, and Z. Yi. Real-time robot path planning based on a modified pulse-coupled neural network model. *IEEE Trans. on Neural Netw.*, 20(11):1724–1739, 2009.

[221] H. Qu, Z. Yi, and S. X. Yang. Efficient shortest-path-tree computation in network routing based on pulse-coupled neural networks. *IEEE Trans. on Cybern.*, 43(3):995–1010, 2013.

[222] M. Quoy, S. Moga, and P. Gaussier. Dynamical neural networks for planning and low-level robot control. *IEEE Trans. on Syst., Man and Cybern., Part A: Syst. and Humans*, 33(4):523–532, 2003.

[223] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2):10822–10825, 2008.

[224] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[225] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

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

[227] G. Ren et al. Multiple chaotic central pattern generators for locomotion generation and leg damage compensation in a hexapod robot. In *IEEE Intl. Conf. on Intell. Robots and Systems*, pages 2756–2761, 2012.

[228] M.-J. Rho, S. Lavoie, and T. Drew. Effects of red nucleus microstimulation on the locomotor pattern and timing in the intact cat: a comparison with the motor cortex. *Journal of neurophysiology*, 81(5):2297–2315, 1999.

[229] M. J. Richardson, K. Shockley, B. R. Fajen, M. A. Riley, and M. T. Turvey. Ecological psychology: Six principles for an embodied–embedded approach to behavior. In *Handbook of cognitive science*, pages 159–187. Elsevier, 2008.

[230] L. Righetti and A. J. Ijspeert. Pattern generators with sensory feedback for the control of quadruped locomotion. In *Intl. Conf. on Robotics and Auton.*, pages 819–824, 2008.

[231] J. Rizo, J. Coronado, A. Forero, C. Otalora, and M. Devy. URSULA: robotic demining system. In *Proc. of the 11th Int. Conf. on Adv. Robot.*, pages 538–543, 2003.

[232] S. Rossignol, R. Dubuc, and J.-P. Gossard. Dynamic sensorimotor interactions in locomotion. *Physiological reviews*, 86(1):89–154, 2006.

[233] S. Rossignol and A. Frigon. Recovery of locomotion after spinal cord injury: some facts and mechanisms. *Annual review of neuroscience*, 34:413–440, 2011.

[234] U. Roth, M. Walker, A. Hilmann, and H. Klar. Dynamic path planning with spiking neural networks. In *Biological and Artificial Computation: From Neuroscience to Technology*, pages 1355–1363. 1997.

[235] P. F. Rowat, Allen, and I. Selverston. Oscillatory mechanisms in pairs of neurons connected with fast inhibitory synapses. *J. Comp. Neurosci*, pages 103–127, 1997.

[236] A. Roy. Connectionism, controllers, and a brain theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(6):1434–1441, Nov 2008.

[237] R. B. Rusu and Others. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[238] I. A. Rybak, N. A. Shevtsova, M. Lafreniere-Roula, and D. A. McCrea. Modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *The Journal of physiology*, 577(2):617–639, 2006.

[239] I. A. Rybak, K. Stecina, N. A. Shevtsova, and D. A. McCrea. Modelling spinal circuitry involved in locomotor pattern generation: insights from the effects of afferent stimulation. *The Journal of physiology*, 577(2):641–658, 2006.

[240] L. R. Sahawneh, M. E. Argyle, and R. W. Beard. 3D path planning for small UAS operating in low-altitude airspace. In *Proc. of Int. Conf. on Unmanned Aircraft Systems*, pages 413–419, June 2016.

[241] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: system overview and integration. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, volume 3, pages 2478–2483 vol.3, 2002.

[242] C. P. Santos and V. Matos. Gait transition and modulation in a quadruped robot: A brainstem-like modulation approach. *Robotics and Autonomous Systems*, 59(9):620–634, 2011.

[243] A. A. Saputra, J. Botzheim, and N. Kubota. Walking speed control in human behavior inspired gait generation system for biped robot. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 4895–4902. IEEE, 2016.

[244] A. A. Saputra, J. Botzheim, I. A. Sulistijono, and N. Kubota. Biologically inspired control system for 3-D locomotion of a humanoid biped robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):898–911, 2015.

[245] A. A. Saputra, J. Botzheim, I. A. Sulistijono, and N. Kubota. Biologically Inspired Control System for 3-{D} Locomotion of a Humanoid Biped Robot. *IEEE Transaction on Systems, Man, and Cybernetics: Systems*, 46(7):898–911, 2016.

[246] A. A. Saputra, W. H. Chin, Y. Toda, N. Takesue, and N. Kubota. Dynamic density topological structure generation for real-time ladder affordance detection. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3439–3444, Nov 2019.

[247] A. A. Saputra et al. Bézier curve model for efficient bio-inspired locomotion of low cost four legged robot. In *Proc. of Intl. Conf. on Intelligent Robots and Systems*, pages 4443–4448, 2016.

[248] A. A. Saputra et al. Evolving a sensory-motor interconnection for dynamic quadruped robot locomotion behavior. In *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 7089–7095, 2018.

[249] A. A. Saputra et al. Evolving a sensorymotor interconnection structure for adaptive biped robot locomotion. *IEEE Transactions on Cognitive and Developmental Systems*, 11(2):244–256, June 2019.

[250] A. A. Saputra, C. W. Hong, and N. Kubota. Real-time grasp affordance detection of unknown object for robot-human interaction. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3093–3098, Oct 2019.

[251] A. A. Saputra, A. S. Khalilullah, and N. Kubota. Development of Humanoid Robot Locomotion Based on Biological Approach in EEPIS Robot Soccer (EROS). In *Robot Soccer World Cup*, pages 303–315. Springer, 2015.

[252] A. A. Saputra, A. S. Khalilullah, and I. A. Sulistijono. Acceleration and Deceleration Optimization Using Inverted Pendulum Model on Humanoid Robot {EROS}-2. In *Proc. of the Indonesian Symposium on Robot Soccer Competition*, pages 17–22, Yogyakarta, Indonesia, jul 2014.

[253] A. A. Saputra, A. S. Khalilullah, I. A. Sulistijono, and N. Kubota. Adaptive motion pattern generation on balancing of humanoid robot movement. In *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*, pages 1479–1484. IEEE, 2015.

[254] A. A. Saputra and N. Kubota. Synthesis of neural oscillator based dynamic rhythmic generation in quadruped robot locomotion. In *Proc. of Intl. Elec. Symp. on Knowledge Creation and Intell. Comp. (IES-KCIC)*, pages 184–191, 2018.

[255] A. A. Saputra, I. A. Sulistijono, J. Botzheim, and N. Kubota. Inter-connection Structure Optimization for Neural Oscillator Based Biped Robot Locomotion. In *Proc. of the IEEE Symposium on Robotic Intelligence in Informationally Structured Space*, pages 288–294, Cape Town, South-Africa, dec 2015.

[256] A. A. Saputra, I. A. Sulistijono, J. Botzheim, and N. Kubota. Interconnection structure optimization for neural oscillator based biped robot locomotion. In *IEEE Symposium Series on Computational Intelligence*, pages 288–294, 2015.

[257] A. A. Saputra, I. A. Sulistijono, A. S. Khalilullah, T. Takeda, and N. Kubota. Combining Pose Control and Angular Velocity Control for Motion Balance of Humanoid Robot Soccer {EROS}. In *Proc. of the IEEE Symposium on Robotic Intelligence in Informatically Structured Space*, pages 16–21, Orlando, USA, dec 2014.

[258] A. A. Saputra, T. Takeda, J. Botzheim, and N. Kubota. Multi-objective evolutionary algorithm for neural oscillator based robot locomotion. In *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, pages 2655–2660. IEEE, 2015.

[259] A. A. Saputra, T. Takeda, J. Botzheim, and N. Kubota. Multi-objective evolutionary algorithm for neural oscillator based robot locomotion. In *Proc. of the 41st Annual Conference of the IEEE Industrial Electronics Society*, pages 2655–2660, Yokohama, Japan, November 2015.

[260] A. A. Saputra, T. Takeda, and N. Kubota. Efficiency Energy on Humanoid Robot Walking using Evolutionary Algorithm. In *Proc. of the IEEE Congress on Evolutionary Computation*, pages 573–578, Sendai, Japan, 2015.

[261] A. A. Saputra, Y. Toda, J. Botzheim, and N. Kubota. Neuro-activity-based dynamic path planner for 3-d rough terrain. *IEEE Transactions on Cognitive and Developmental Systems*, 10(2):138–150, 2018.

[262] C. Saranya, M. Unnikrishnan, S. A. Ali, D. Sheela, and D. V. Lalithambika. Terrain based d algorithm for path planning. *IFAC-PapersOnLine*, 49(1):178 – 182, 2016.

[263] A. B. Sawers and B. J. Hafner. Outcomes associated with the use of microprocessor-controlled prosthetic knees among individuals with unilateral transfemoral limb loss: a systematic review. *JPO: Journal of Prosthetics and Orthotics*, 25(4S):P4—-P40, 2013.

[264] M. Schilling, T. Hoinville, J. Schmitz, and H. Cruse. Walknet, a bio-inspired controller for hexapod walking. *Biological cybernetics*, 107(4):397–419, 2013.

[265] A. Schmidt and A. Kasiński. The visual SLAM system for a hexapod robot. In *International Conference on Computer Vision and Graphics*, pages 260–267. Springer, 2010.

[266] A. Schneider, J. Paskarbeit, M. Schaeffersmann, and J. Schmitz. Hector, a new hexapod robot platform with increased mobility-control approach, design and communication. In *Advances in Autonomous Mini Robots*, pages 249–264. Springer, 2012.

[267] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell. Design of HyQ–a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(6):831–849, 2011.

[268] R. Shaw. Processes, acts, and experiences: Three stances on the problem of intentionality. *Ecological Psychology*, 13(4):275–314, 2001.

[269] R. Shaw. The agent-environment interface: {S}imon's indirect or {G}ibson's direct coupling? *Ecological Psychology*, 15(1):37–106, 2003.

[270] L. Silberbauer. A new 6 wheeled robot for humanitarian demining. In *EURON/IARP Int. Workshop on Robot. for Risky Interventions and Surveillance of the Environment*, 2008.

[271] P. Silva, C. P. Santos, V. Matos, and L. Costa. Automatic generation of biped locomotion controllers using genetic programming. *Robotics and Autonomous Systems*, 62(10):1531–1548, 2014.

[272] B. G. R. Smith, C. M. Saaj, and E. Allouis. Evolving legged robots using biologically inspired optimization strategies. In *Proc. of the 2010 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1335–1340, dec 2010.

[273] S. Song. *Development of an omni-directional gait generator and a stabilization feedback controller for humanoid robots*. PhD thesis, Virginia Polytechnic Institute and State University, 2010.

[274] I. Spectrum. Laikago, feb 2019.

[275] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert. Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot. *The International Journal of Robotics Research*, 32(8):932–950, 2013.

[276] J. Strom, G. Slavov, and E. Chown. Omnidirectional walking using zmp and preview control for the nao humanoid robot. In *Robot Soccer World Cup*, pages 378–389. Springer, 2009.

[277] T. Sun, D. Shao, Z. Dai, and P. Manoonpong. Adaptive neural control for self-organized locomotion and obstacle negotiation of quadruped robots. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1081–1086. IEEE, 2018.

[278] X. Sun et al. Planning and control of stable ladder climbing motion for the four-limbed robot warec-1. In *Proc. of Intl. Conf. on Intelligent Robots and Systems*, pages 6547–6554, 2017.

[279] G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159, 1991.

[280] N. Takase, J. Botzheim, and N. Kubota. Evolving spiking neural network for robot locomotion generation. In *Proc. of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 558–565, may 2015.

[281] I. Tanev. Genetic programming incorporating biased mutation for evolution and adaptation of Snakebot. *Genetic Programming and Evolvable Machines*, 8(1):39–59, mar 2007.

[282] I. T. Tanev and K. Shimohara. Interactive learning of consensus sequences in genetic programming for evolution of snake-like robot. In *Proc. of the 2007 IEEE Congress on Evolutionary Computation*, pages 3662–3670, sep 2007.

[283] H. Ten Donkelaar. Evolution of vertebrate motor systems. *Brain evolution and cognition*, pages 77–112, 2001.

[284] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[285] Y. Toda, Z. Ju, H. Yu, N. Takesue, K. Wada, and N. Kubota. Real-time 3d point cloud segmentation using growing neural gas with utility. In *Human System Interactions (HSI), 2016 9th International Conference on*, pages 418–422. IEEE, 2016.

[286] Y. Toda and N. Kubota. Path planning using multi-resolution map for a mobile robot. In *Proc. of SICE Annual Conf.*, pages 1276–1281, 2011.

[287] Y. Toda and Others. Real-time 3D point cloud segmentation using Growing Neural Gas with Utility. In *Proc. of Intl. Conf. on Human System Interactions*, pages 418–422, jul 2016.

[288] Y. Toda, N. Takesue, K. Wada, N. Kubota, and H. Yu. Real-time 3D point cloud segmentation using Growing Neural Gas with utility. In *Proc. of Int. Conf. on Human Syst. Interactions*, pages 418–422, 2016.

[289] M. Toeda, S. Aoi, S. Fujiki, T. Funato, K. Tsuchiya, and D. Yanagihara. Gait generation and its energy efficiency based on rat neuromusculoskeletal model. *Frontiers in Neuroscience*, 13:1337, 2019.

[290] A. Topchyan and T. Topchyan. Muscle-based skeletal bipedal locomotion using neural evolution. In *Computer Science and Information Technologies (CSIT), 2013*, pages 1–6, sep 2013.

[291] D. Tóvári and N. Pfeifer. Segmentation based robust interpolation-a new approach to laser data filtering. *Intl. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/19):79–84, 2005.

[292] D. T. Tran, I. M. Koo, Y. H. Lee, H. Moon, S. Park, J. C. Koo, and H. R. Choi. Central pattern generator based reflexive control of quadruped walking robots using a recurrent neural network. *Robotics and autonomous systems*, 62(10):1497–1516, 2014.

[293] K. Tsujita, T. Kobayashi, and T. Masuda. Feasibility study on stability of gait patterns with changeable body stiffness using pneumatic actuators in a quadruped robot. *Advanced robotics*, 23(5):503–520, 2009.

[294] M. T. Turvey. Affordances and prospective control: An outline of the ontology. *Ecological psychology*, 4(3):173–187, 1992.

[295] M. T. Turvey and R. E. Shaw. Toward an ecological physics and a physical psychology. *The science of the mind: 2001 and beyond*, pages 144–169, 1995.

[296] A. Ude, V. Wyart, L.-H. Lin, and G. Cheng. Distributed visual attention on a humanoid robot. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 381–386. IEEE, 2005.

[297] J. Vaillant et al. Multi-contact vertical ladder climbing with an hrp-2 humanoid. *Autonomous Robots*, 40(3):561–580, 2016.

[298] D. C. Van Essen, C. H. Anderson, and D. J. Felleman. Information processing in the primate visual system: an integrated systems perspective. *Science*, 255(5043):419–423, 1992.

[299] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. Kelly, and S. Leutenegger. Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151, 2018.

[300] A. Wagoner, A. Jagadish, E. T. Matson, L. EunSeop, Y. Nah, K. K. Tae, D. H. Lee, and J.-E. Joeng. Humanoid robots rescuing humans and extinguishing fires for cooperative fire security system using HARMS. In *Proc. of the 6th Intl. Conf. on Automation, Robotics and Applications (ICARA)*, pages 411–415, Feb 2015.

[301] D. Wahrmann et al. Vision-based 3d modeling of unknown dynamic environments for real-time humanoid navigation. *Intl. Journal of Humanoid Robotics*, 16(01):1950002, 2019.

[302] H. Wang, W. Lyu, P. Yao, X. Liang, and C. Liu. Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system. *Chinese Journal of Aeronautics*, 28(1):229–239, 2015.

[303] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics*, 31(4), 2012.

[304] Y. Wang, X. Xue, and B. Chen. Matsuoka's CPG with desired rhythmic signals for adaptive walking of humanoid robots. *IEEE transactions on cybernetics*, 50(2):613–626, 2018.

[305] K. Welke, D. Schiebener, T. Asfour, and R. Dillmann. Gaze selection during manipulation tasks. In *2013 IEEE International Conference on Robotics and Automation*, pages 652–659. IEEE, 2013.

[306] W. Widajewicz, B. Kably, and T. Drew. Motor cortical activity during voluntary gait modifications in the cat. II. Cells related to the hindlimbs. *Journal of Neurophysiology*, 72(5):2070–2089, 1994.

[307] S. R. Williams and L. N. Fletcher. A dendritic substrate for the cholinergic control of neocortical output neurons. *Neuron*, 101(3):486–499, 2019.

[308] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini. Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5148–5154. IEEE, 2015.

[309] H. Witte, R. Hackert, K. E. Lilje, N. Schilling, D. Voges, G. Klauer, W. Ilg, J. Albiez, A. Seyfarth, D. Germann, and Others. Transfer of biological principles into the construction of quadruped walking machines. In *Proceedings of the Second International Workshop on Robot Motion and Control. RoMoCo'01 (IEEE Cat. No. 01EX535)*, pages 245–249. IEEE, 2001.

[310] K. Wolff, J. Pettersson, A. Heralic, and M. Wahde. Structural Evolution of Central Pattern Generators for Bipedal Walking in 3D Simulation. In *Proc. of the 2006 International Conference on Systems, Man and Cybernetics*, volume 1, pages 227–234, oct 2006.

[311] K. Wolff and M. Wahde. Evolution of Biped Locomotion Using Linear Genetic Programming. In H. Zhang, editor, *Climbing and Walking Robots Towards New Applications*, chapter 16, pages 335–356. intechweb.org, 2007.

[312] J. Woo, J. Botzheim, and N. Kubota. System Integration for Cognitive Model of a Robot Partner. *Intelligent Automation & Soft Computing*, pages 1–14, 2017.

[313] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert. Autonomous navigation for bigdog. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4736–4741. IEEE, 2010.

[314] X. Xiong, F. Wörgötter, and P. Manoonpong. Neuromechanical control for hexapedal robot walking on challenging surfaces and surface classification. *Robotics and Autonomous Systems*, 62(12):1777–1789, 2014.

[315] X. Xiong, F. Wörgötter, and P. Manoonpong. Adaptive and energy efficient walking in a hexapod robot under neuromechanical control and sensorimotor learning. *IEEE transactions on cybernetics*, 46(11):2521–2534, 2015.

[316] W. Xueli, G. Yapei, and Z. Jianhua. A novel algorithm for shortest path problem based on pulse coupled neural network. In *Proc. of The 27th Chinese Control and Decision Conf.*, pages 2468–2473, May 2015.

[317] Y. Yamada, S. Nishikawa, K. Shida, R. Niiyama, and Y. Kuniyoshi. Neural-body coupling for emergent locomotion: A musculoskeletal quadruped robot with spinobulbar model. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1499–1506. IEEE, 2011.

[318] S. X. Yang and C. Luo. A neural network approach to complete coverage path planning. *IEEE Trans. on Syst., Man, and Cybern., B, Cybern.*, 34(1):718–724, 2004.

[319] S. X. Yang and M. Meng. An efficient neural network approach to dynamic robot motion planning. *Neural Netw.*, 13(2):143 – 148, 2000.

[320] S. X. Yang and M. Meng. Neural network approaches to dynamic collision-free trajectory generation. *IEEE Trans. on Syst., Man, Cybern., B, Cybern.*, 31(3):302–318, 2001.

[321] S. X. Yang and M. Q. Meng. Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach. *IEEE Trans. on Neural Netw.*, 14(6):1541–1552, 2003.

[322] Z. Yang, K. Ito, K. Saijo, K. Hirotsune, A. Gofuku, and F. Matsuno. A combined navigation strategy by a steering wheel and a mouse for a tank rescue robot. In *Proc. of the IEEE Intl. Conf. on Robotics and Biomimetics*, pages 239–244, Aug 2004.

[323] H. Yoneda et al. Vertical ladder climbing motion with posture control for multi-locomotion robot. In *Proc of Intl. Conf. on Intelligent Robots and Systems*, pages 3579–3584, 2008.

[324] A. Yorita and N. Kubota. Cognitive development in partner robots for information support to elderly people. *IEEE Transactions on Autonomous Mental Development*, 3(1):64–73, 2011.

[325] Y. Yoshida, K. Takeuchi, Y. Miyamoto, D. Sato, and D. Nenchev. Postural Balance Strategies in Response to Disturbances in the Frontal Plane and Their Implementation With a Humanoid Robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(6):692–704, jun 2014.

[326] J. Yu et al. A survey on cpg-inspired control models and system implementation. *IEEE transactions on neural networks and learning systems*, 25(3):441–456, 2013.

[327] Z. Yu, X. Chen, Q. Huang, W. Zhang, L. Meng, W. Zhang, and J. Gao. Gait planning of omnidirectional walk on inclined ground for biped robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):888–897, 2015.

[328] M. Zeng, F. Zhao, J. Zheng, and X. Liu. Octree-based fusion for realtime 3d reconstruction. *Graphical Models*, 75(3):126–136, 2013.

[329] G. Zhang, X. Rong, C. Hui, Y. Li, and B. Li. Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control. *Advanced Robotics*, 30(4):284–297, 2016.

[330] J. Zhang, F. Gao, X. Han, X. Chen, and X. Han. Trot gait design and CPG method for a quadruped robot. *Journal of Bionic Engineering*, 11(1):18–25, 2014.

[331] J. Zhang, D. Wang, M. Shi, and J. Y. Wang. Output-threshold coupled neural network for solving the shortest path problems. *Science in China Series F: Information Sciences*, 47(1):20–33, 2004.

[332] Y. Zhang et al. Motion planning of ladder climbing for humanoid robots. In *Proc. of Conf. on Technologies for Practical Robot Applications*, pages 1–6, 2013.

[333] Y. Zhang et al. Motion planning of ladder climbing for humanoid robots. In *Proc. of Intl. Conf. on Technologies for Practical Robot Applications*, pages 1–6, 2013.

[334] Y. Zhang, L. Wu, G. Wei, and S. Wang. A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network. *Digital Signal Proces.*, 21(4):517–521, 2011.

[335] Y. Zhu, B. Jin, Y. Wu, T. Guo, and X. Zhao. Trajectory correction and locomotion analysis of a hexapod walking robot with semi-round rigid feet. *Sensors*, 16(9):1392, 2016.

Acknowledgement

I would like to express deepest gratitude to my supervisor Prof. Naoyuki Kubota for his full support, expert guidance and giving me the opportunity to work in his lab. I have gained valuable experiences during my doctoral course study.

My sincere thanks go to the sponsors of my scholarship,Tokyo Metropolitan Government, for their support. Their generosity provided me with the freedom to focus wholeheartedly on my research, a freedom that was instrumental in seeing this research through.

During my stay at the Kubota laboratory, I had the chance to collaborate with other researchers around the world. I would like to thanks all the lab members for their assistance and guidance.

Last but not least, I would like to thanks to my wife Hanayuki and my family for their unconditional support.

Appendices

Appendix A

Design of Quadruped Robot

Our robot is equipped with external and internal sensors. We use point-cloud data information generated by a laser depth sensor as external sensory information. Since depth sensors are limited in frequency rate, size, weight, and range, we propose a light-weight array of time-of-flight sensors which alleviates these limitations. To provide internal sensory information, we use an inertial measuring unit (IMU), four force sensors, and four grip-touch sensors.

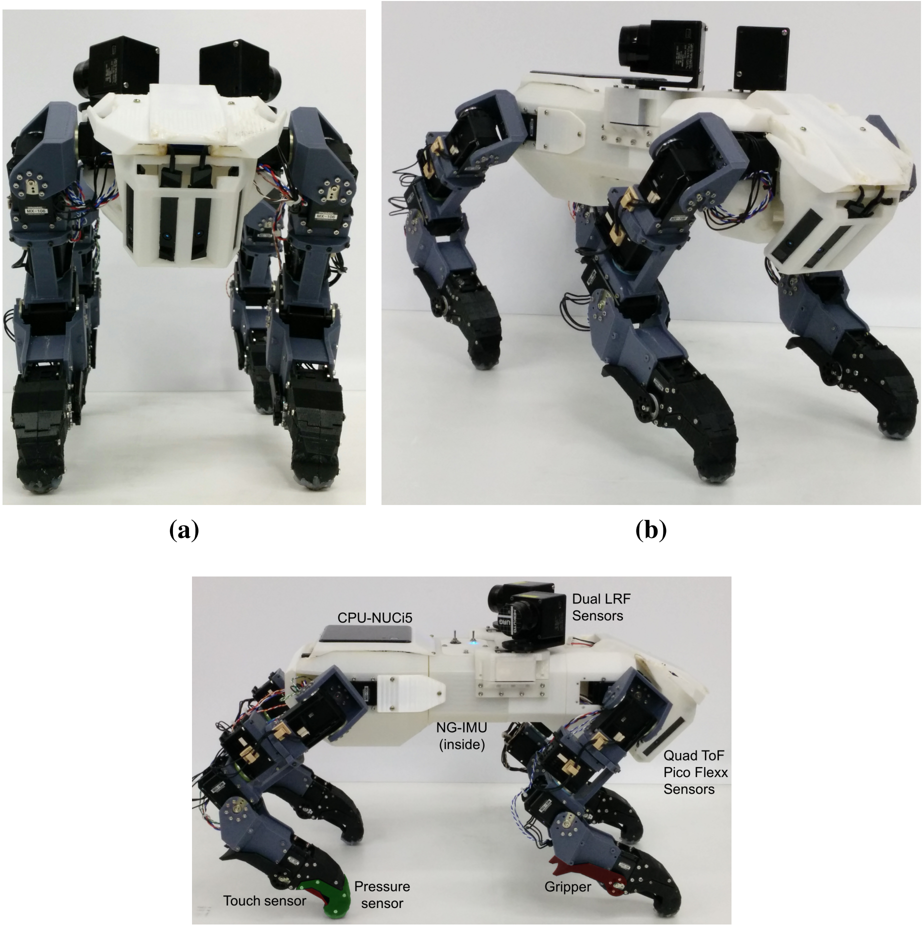
As stated in the Council on CompetitivenessNippon (COCN) report, robots suitable for use in disaster situations must be able to move over all of rough, sloped and natural terrain (grass, uneven soil), through narrow spaces, and be able to climb stairs and vertical ladders. When we seek inspiration from the animal kingdom, the cat family (Felidae) stands out as able do all of these things. Cats can handle many complex environmental conditions. They can swim, are agile, and can climb trees. The cat offers a most appropriate archetype to imitate in agile quadrupedal robots. The feline robot that we developed is shown in Figure A.1.

A.1 Mechanical design

Our proposed quadrupedal robot is similar in size to a mature domestic cat: 25 cm (width) 60 cm (length) 30 cm (height).

A.1.1 Leg's design

The robots foreleg imitates the cats forelimb structure minus the wrist joint. It has only two joints, the shoulder and elbow. There are three actuators associated with the ball joint structure of the shoulder, and one actuator associated with the hinge joint structure of the elbow. To design the robots hindleg, we considered the cats rhythmic motion, in which the proximal and distal leg segments maintain their relative angular orientation during most of the cycle, the deviation of angular joints differing only at the onset of toe-off. In the hindlegs, therefore, we simplified by eliminating the knee



(a)

(b)

(c)

Figure A.1: *The quadruped robot*

.

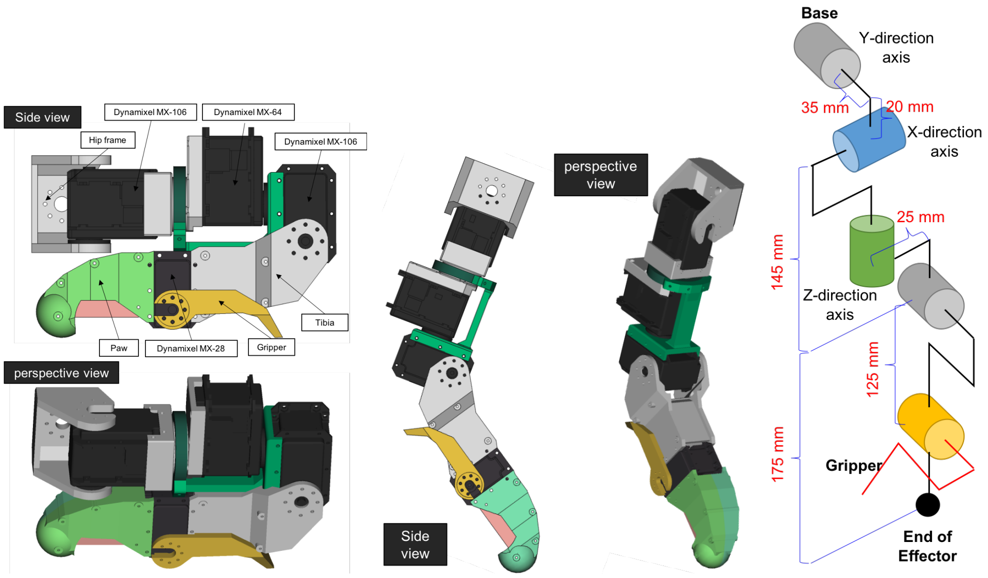


Figure A.2: *The design of Leg a) flexed, b) extended. c) Leg actualization*

.

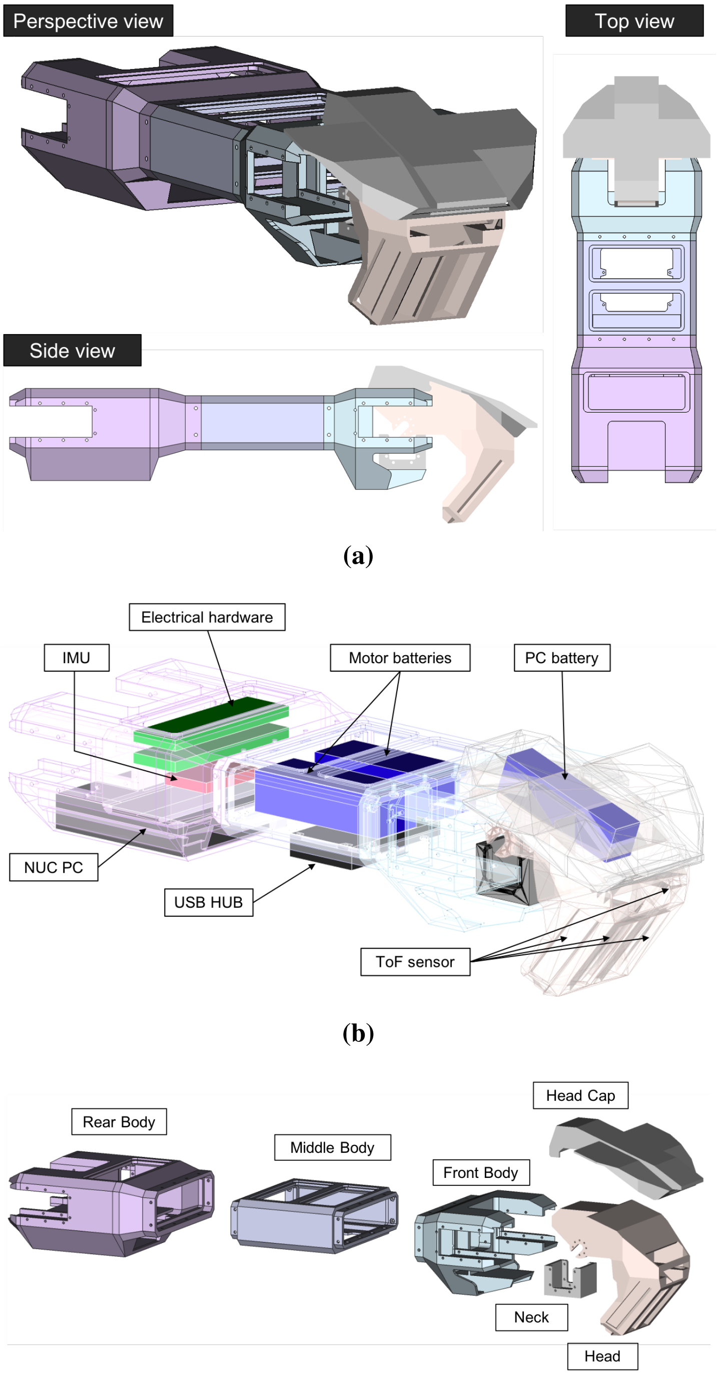
joint so the ankle and hip joints could be directly integrated. The leg can be seen in Figure A.2. There are five degrees of freedom in each leg, one of which is used as the gripper joint. The tibia is 175 mm long, and the femur is 145 mm long. The robots Denavit-Hartenberg parameters are summarized in Table A.1.

Table A.1: *DH Table of the joint leg structure.*

Joint	Hip-y	Hip-x	Hip-z	Knee
α_i	$\pi/2$	$-\pi/2$	$-\pi/2$	0
a_i	20 mm	0	25 mm	172 mm
d_i	35 mm	0	145 mm	0
$\theta_{0,i}$	$\pi/2$	$-\pi/2$	$-\pi/2$	$-\pi/2$
θ_i	$\theta_{0,1}+\theta_1$	$\theta_{0,1}+\theta_2$	$\theta_{0,1}+\theta_3$	$\theta_{0,1}+\theta_4$

A.1.2 Body design

The robots overall body shape can be seen in Figure A.3. The shell was 3D-printed in poly-(lactic acid) (PLA). The robot body comprises three parts: rear, middle, and front. The rear legs are attached to the rear part, which also holds the NUC PC, IMU sensor, and electrical hardware. The middle part holds two batteries for the motors and a USB Hub. The front part provides an attachment point for the neck and head. The head holds a battery for the PC.



(a)

(b)

(c)

Figure A.3: *The robot body (torso and head). (a) Orthographic projections. (b) Interior hardware placements. (c) Exploded parts.*

A.1.3 End of Effector

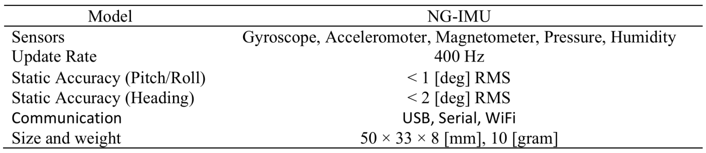
We designed the end effector to support agile movements such as walking on rough terrain and climbing vertical ladders. The end effectors must also measure the ground reaction force, and must satisfy size constraints. When climbing, cats use claws to grasp rocky walls, trees, poles, etc, however, than is appropriate for our proposed robot. Futhermore, cats grasp by using two limbs in concert. Cats also find it difficult to climb vertical ladders.

In contrast, humans and monkeys have hands to hold and hang from ladder rungs. However, the hand mechanism for such hanging behavior needs a huge torque, which would require a correspondingly bigger servomotor. We simplified using a hook-shaped end effector that requires no actuator. The design can be seen in Figure A.4. The end is rounded to simplify footing, eliminating the need for an actuator. Furthermore, in the sensory design, we put a force-sensitive resistor (FSR) between the upper and lower parts of the paw. A switch inside the hook cavity serves as a sensor to detect whether the paw is hooked over a rung. Behind the paw is a moveable claw for grasping and for supporting the hindleg to stand on a rung. The claw, moved by a low-torque servomotor, helps to avoid slippage.

A.2 Sensory Systems

We provided the cat robot with several sensors representing exteroceptors and interoceptors. To represent the exteroceptors, we built a quad-composite time-of-flight sensor for detecting the surroundings in front the robot, and a dual-laser range-finder for observing more widely. To represent interoceptors, we installed force sensors (force-sensitive resistors) and touch sensors (microswitches) in each leg, and an inertial measurement unit (IMU) inside the body. We use IMU module NG-IMU, as specified in Table A.2.

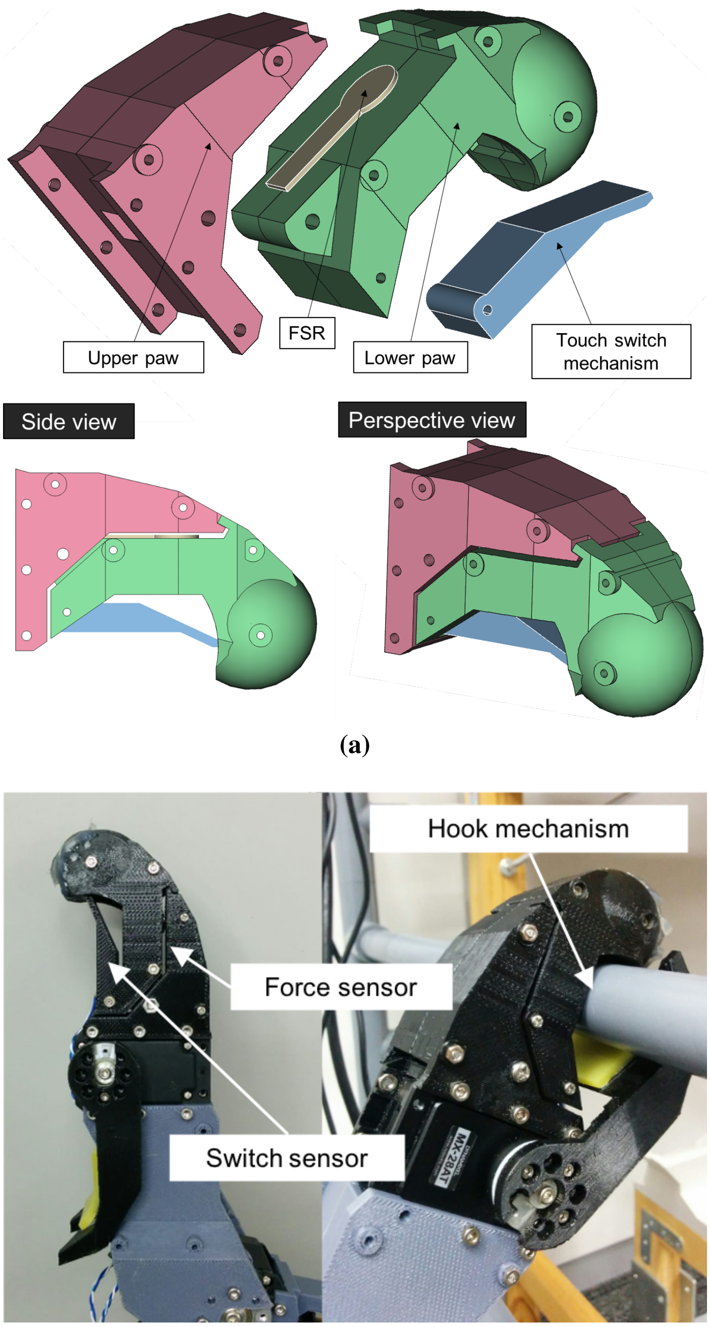
Table A.2: *NG-IMU sensor specifications*



Model	NG-IMU
Sensors	Gyroscope, Acceleromoter, Magnetometer, Pressure, Humidity
Update Rate	400 Hz
Static Accuracy (Pitch/Roll)	< 1 [deg] RMS
Static Accuracy (Heading)	< 2 [deg] RMS
Communication	USB, Serial, WiFi
Size and weight	50 × 33 × 8 [mm], 10 [gram]

A.2.1 Quad-Composite Time of Flight sensor

This sensor will be installed in the head of the robot. It combines four CamBoard pico flexx ToF sensors made by pmd, as specified in Table A.3. The composite sensor structure is depicted in Figure A.5. This design, combined with the robots head shape, provides a wide field of view in order



(a)

(b)

Figure A.4: *The end effector. (a) CAD drawings (b) The 3D-printed effector.*

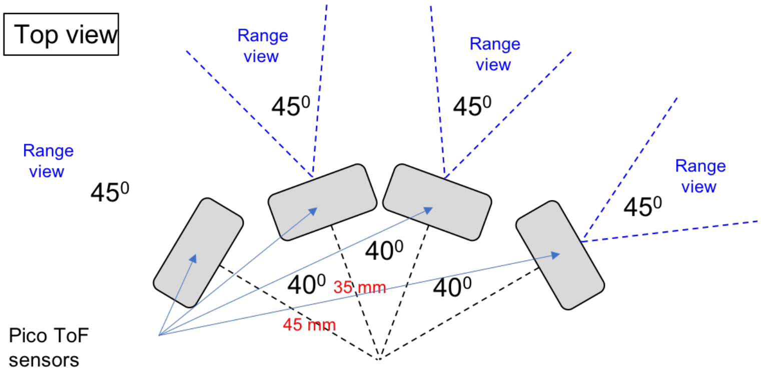


Figure A.5: *The quad-composite time-of-flight sensor arrangement*

to minimize the number of actuators needed in the robots neck. The neck hence contains only one actuator, rotating in the sagittal plane. The CAD design drawings and photographs can be seen in Figure A.6.

Table A.3: *Time-of-flight sensor specifications*

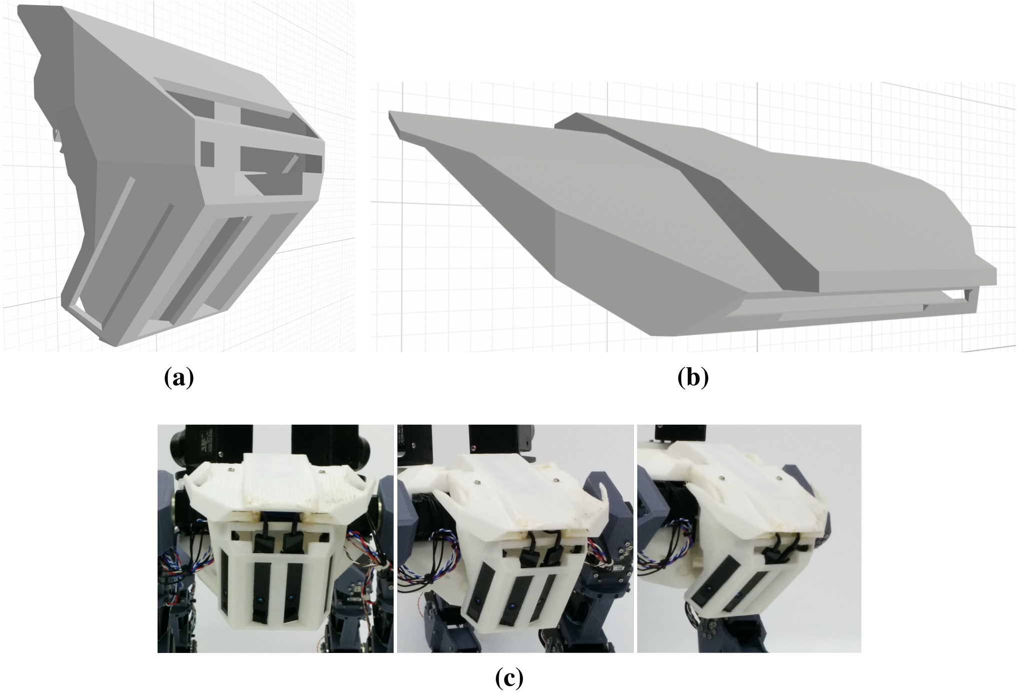
Model	CamBoard pico flexx
Dimension, weight	68 × 17 × 7.35 mm, 8 g
Measurement range	0.1–4 m
Framerate	Up to 45 fps (3D frames)
Resolution	224 × 171 (38k) pixels
Viewing angle (H x V)	62° × 45°

A.2.2 Dual-Laser Range Finder sensor (DLRF)

The DLRF is composed of two LRF sensors, with each LRF is attached to a Dynamixel MX-28 servomotor. This mechanism allows the sensors to measure distances. The design can be seen in Figure A.7. Table A.4 shows the specifications of the LRF sensor used. In Figure A.7c we can see the moving mechanism of the sensors. The sensors will move symmetrically, where if the left sensor moving clockwise then the opposite sensor will move counterclockwise. Each sensor will move 240 [degree] of range. After reaching the limit degree, then the sensor will move the opposite direction.

Table A.4: *LRF sensor specifications*

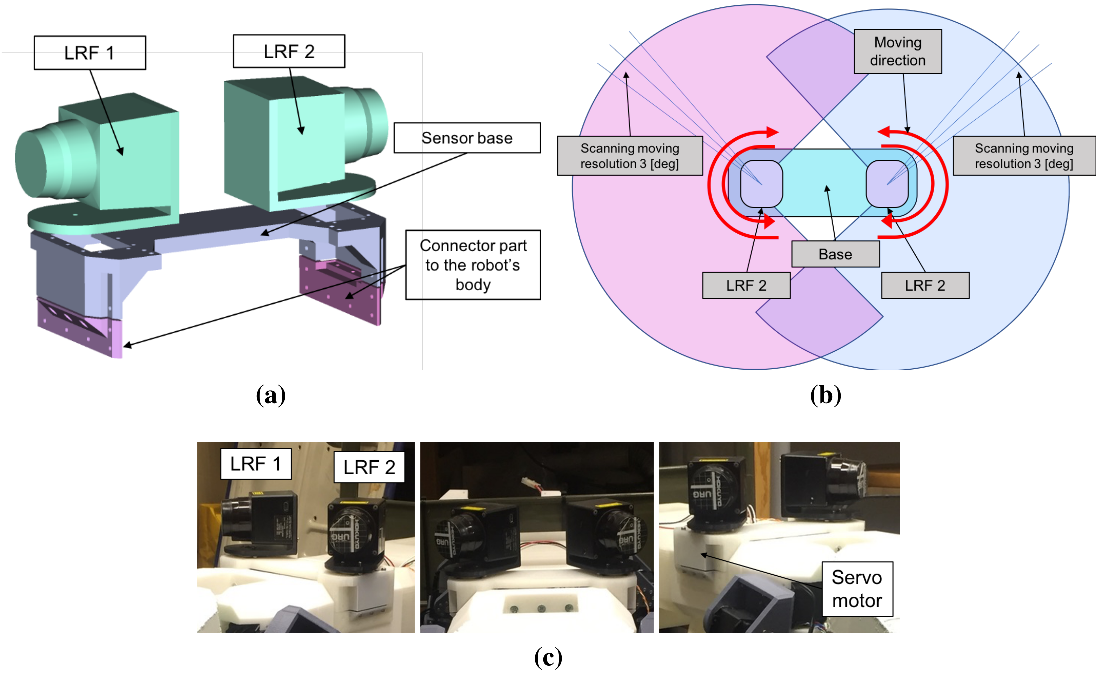
Model	URG–04LX–UG01
Weight	160 g
Measurement range	20–5600 mm
Scanning time	100 ms
Scanning accuracy	60–1000 mm: ±30 mm; 1000–4095 mm : ±3 mm
Measurement range	240°


(a)

(b)

(c)

Figure A.6: *The quad-composite time-of-flight sensors. (a) Head housing; the sensors are mounted inside the downward-facing slits. (b) Head cap. (c) The head with installed sensors*


LRF 1
LRF 2
Sensor base
Connector part to the robot's body
Moving direction
Scanning moving resolution 3 [deg]
Scanning moving resolution 3 [deg]
Base
LRF 2
LRF 2
LRF 1
LRF 2
Servo motor

(a)

(b)

(c)

Figure A.7: *Design of DLRF sensor. (a) CAD design. (b) The structure of the sensor (c) The appearance design of the Dual LRF sensor*

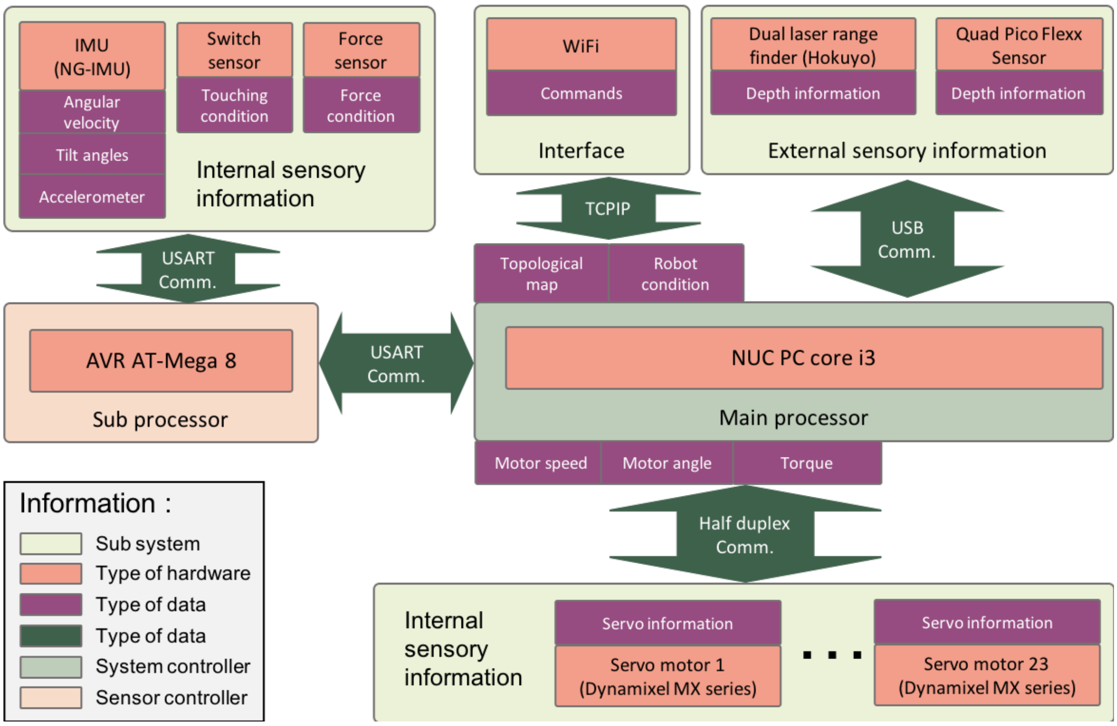


Figure A.8: *Structure of the electrical system*

A.3 Electrical hardware

The robot has been equipped with a hardware configuration to handle both the internal and the external sensory information. The hardware structure can be seen in Figure 8. We use an ATmega 8 microcontroller as the sub-controller for pre-processing the internal sensory inputs from the force sensor, touch sensor, and IMU. A NUC PC core i3 serves as the main controller for processing several advanced systems such as perception, motion control, communication, and interfacing.

The sub-controller processes analog signals from the force sensor, touch sensor, and IMU sensor through its analog to digital converter input. The data stream is then transferred to the main controller via a USB connection. All external sensory information is also conveyed via USB connections. The main controller generates digital motor control signals for all of the servomotors. There are 12 servo motors Dynamixel MX-106, 4 servo motors Dynamixel MX-64, and 7 servo motor Dynamixel MX-28.

The electrical system is powered by two 4cell lithiumpolymer batteries holding 2700 mAh and one 4cell lithiumpolymer battery holding 2200 mAh. The batteries are expected to be power the robot for around 15 minutes.

Appendix B

Musculoskeletal Model of Forelimb and Hindlimb of Felidae Family of Quadruped Animal

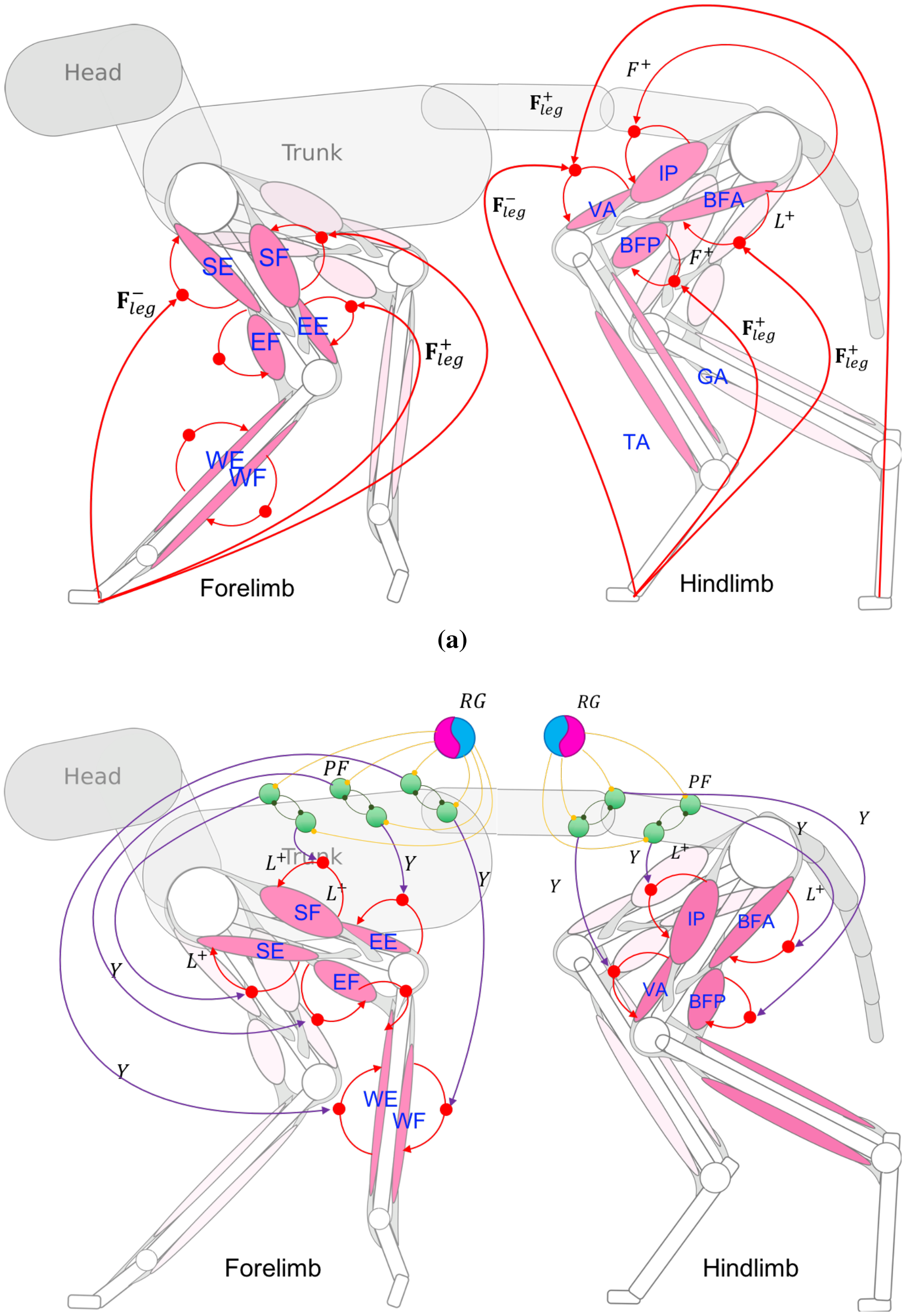
I design the reflex model of hindlimb and forelimb based on mammal structure. Human beings have only hindlimb to support their body for locomotion. However, most legged mammals use both hindlimb and forelimb to support their locomotion [104]. Some researchers have proposed the stance and swing phase in muscle integrated locomotion. Extensor muscle is involved in stance phase and flexor muscle is involved in swing phase.

The investigation process of stance-swing phase transition have shown that transition is controlled by sensory signals from leg proprioceptors. Some of the receptors providing these signals have been identified [217].

Stepping are process involving swing and stance transition inseparably [188, 148]. Swing to stance phase transition should be activated automatically based on stimuli from ground force and hip position. Thus, swing to stance transition is the best timing for controlling movement pattern. I control the swing stimulation by signal pattern modulation generated by neural oscillator proposed by Matsuoka. One CPG neuron represents one joint. This signal will stimulate muscle reflex to perform swing phase. The proposed method consists of hindlimb and forelimb control model as shown in Figure B.1.

B.1 Hindlimb reflex model

I are referring to the rhythmic motion study of the cat, where proximal and distal leg segments retain their relative angular orientation for most parts of the cyclic motion. Compliance mechanism is implemented between ankle and knee joint [275, 309]. I implement some important muscles and chose



(a)

(b)

Figure B.1: *The muscle reflex structure during its phase. a) Muscle reflex structure in stance phase. b) Muscle reflex structure in swing phase.*

two muscle in one joint: 1) iliopsoas (IP, hip flexor) 2) Bicep femoris anterior (BFA, hip extensor) 3) biceps femoris posterior (BFP, knee flexor, and hip extensor) 4) Vastii (VA, knee extensor). I build the force reflex from the same leg (F_{leg}) and from the other 3 legs (F^A_{leg}, F^B_{leg}, F^C_{leg}), where A, B, C, are the contralateral hindlimb, ipsilateral forelimb, and contralateral forelimb, respectively.

B.1.1 Stance phase of hindlimb reflex model

During the stance phase, the muscle stimulation mostly effected by the ground reaction force (F^A_{leg}) [82, 83]. The ground force signal stimulates positive feedback to extensor muscle in hip joint (BFA). BFA stimulation (S_{BFA}) gets force feedback from local leg (F_{leg}) with calculated in Eq (B.1). Where $l_{(CE,BFA)}$ is BFA fiber length and $l_{(off,BFA)}$ is its desired length offset. The BFA's stretch reflex prevents large extension torque in hip joint. To reduce inhibition effect to BFA, negative force feedback of BFA is given to hip flexor (IP). For representing the spring mechanism during the stance phase, VA is stimulated with its own muscle activity using F+. Then, to initiate the transition to swing phase, VA and BFP receive positive and negative feedback respectively from contralateral ground force feedback. The overall muscle stimulation for hindlimb during stance phase can be seen in the following equations:

$$S_{BFA}(t) = S_{0,BFA} + G^F_{BFA} F_{leg} \cdot G^l_{BFA}(l_{CE,BFA} - l_{off,BFA}) \tag{B.1}$$

$$S_{IP}(t) = S_{0,IP} + G^f_{IP} F_{leg} \cdot G^l_{IP}(l_{CE,IP} - l_{off,BFA}) \tag{B.2}$$

$$S_{BFP}(t) = S_{0,BFP} + C^{st}_{BFP} - G^F_{BFP} F^A_{leg} F^B_{leg} \tag{B.3}$$

$$S_{VA}(t) = S_{0,VA} + G^F_{VA} F^A_{leg} F^B_{leg} \tag{B.4}$$

B.1.2 Swing phase of hindlimb reflex model

The swing transition is initiated in the late stance phase, however, the swing movement is controlled and triggered by a signal from CPG. The ith CPG signal Y_i from neuron will trigger hip flexor muscle (IP) and knee flexor muscle (BFP) for several millisecond. To inhibit early extending process, IP receives stretch length reflex. Opposite muscle (BFA) also receives its stretch reflex to force the leg touching the ground. In the end of swing phase, extensor muscle of knee (VA) is stimulated after ending stimulation of CPG. At this time, BFP tends to be silent. The overall muscle stimulation for hindlimb during swing phase can be seen in the following equations:

$$S_{BFA}(t) = S_{0,BFA} - G^y_{BFA} Y_{IP} + G^{l,S}_{BFA}(l_{CE,BFA} - l_{off,BFA}) \tag{B.5}$$

$$S_{IP}(t) = S_{0,IP} + G^y_{IP} Y_{IP} + G^{l,S}_{IP}(l_{CE,IP} - l_{off,IP}) \tag{B.6}$$

$$S_{VA}(t) = S_{0,VA} - G^Y_{VA} Y_{IP} + C_{VA} \tag{B.7}$$

$$S_{BFP}(t) = S_{0,BFP} + G^{Y}_{BFP}Y_{BFP} - C_{BFP} + G^{F}_{BFP}F_{leg} \tag{B.8}$$

B.2 Forelimb reflex model

The forelimb has an important role as the stability controllerinstead of the locomotion. Forelimb tends to monitor stability and positioning rather than locomotion. Therefore it has greater action than hindlimb. Within forelimb, there are some neurological and physiological studies and observational work about muscle activity. Some research investigated the synchronization of muscle patterns in cat's forelimb motion [62, 36, 57].

Forelimb's movement is produced by many combined muscles, monoarticular and biarticular muscle[157]. In order to minimize the number of muscle without eliminating its characteristics, I cluster muscles in every joint into two groups: flexor and extensor muscle. In shoulder joint, I consider anterior deltoid, pectoralis major, and coracobrachialis muscle as flexor muscle, then latissimus dorsi, teres major and minor and posterior deltoid muscles as extensor muscle. There are two muscles in the shoulder and two muscles in the elbow joint (triceps (Tm) and brachialis (Br)). Then there are two muscles in the wrist (flexor carpi ulnaris and Palmaris Lungus for flexor, extensor Carpi Ulnaris and extensor Digitorum for extensor muscle). I only consider the monoarticular muscle in flexor or extensor action.

B.2.1 Stance phase of forelimb reflex model

In stance phase, extensor muscle of elbow, extensor muscle of wrist, and flexor muscle in the shoulder have stance-related activity predominantly. Shoulder flexor muscle activity starts 20-30 ms before forelimb lift-off, and continues until the end of the flexion epoch or slightly into E1. In elbow muscle, activity patterns of all extensors are similar in terms of onset and duration. All begin activity during E1, usually, after the onset of elbow extension, continue activity across limb placement, and cease activity just prior to limb lift-off. Activity in all flexors of elbow is also similar. In wrist joint, flexor muscle is activated in a similar phase with extensor muscle of elbow. After that, it will stop just prior to lift-off. The overall muscle stimulation for forelimb during stance phase can be seen in the following equations:

$$S_{SF}(t) = S_{0,SF} + G^{F}_{SF}F_{leg}G^{l}_{SF}(l_{CE,SF} - l_{off,SF}) + G^{F,A}_{SF}F^{A}_{leg} \tag{B.9}$$

$$S_{SE}(t) = S_{0,SE} - G^{F}_{SE}F^{A}_{leg}G^{l}_{SE}(l_{CE,SF} - l_{off,SF}) \tag{B.10}$$

$$S_{EE}(t) = S_{0,SE} + G^{F}_{SF}F^{A}_{leg} + C_{EE} \tag{B.11}$$

$$S_{EF}(t) = S_{0,EF} \tag{B.12}$$

$$S_{WE}(t) = S_{0,WE} + G^{F}_{WF}F^{A}_{leg} \tag{B.13}$$

$$S_{WF}(t) = S_{0,WF} + C_{WF} \tag{B.14}$$

B.2.2 Swing phase of forelimb reflex model

In swing phase, extensor muscle in shoulder joint is monophasically active during extension epoch. It begins activity during E1, and stop just before lift-off. Flexor muscle in elbow joint generally begins just prior to limb lift-off (F onset) and ceases at or near the onset of E1. Br, which is the only single joint flexor muscle of elbow, is active just prior to foot lift and remains active until approximately midswing. Cleidobrachialis (ClB) and biceps brachii (Bit) normally commence their periods of locomotor activity at about the same time as Br. The activity in ClB, however, always continue throughout the period of swing. Muscle related to wrist joint generally has low activity during swing phase [165]. In elbow muscle, activity patterns of all extensors are similar in terms of onset and duration. All begin activity during E1, usually, after onset of elbow extension, continue activity across limb placement, and cease activity just prior to limb lift-off. Activity in all flexors of elbow is also similar. In swing phase, flexor muscle in elbow joint generally begins just prior to limb lift-off (F onset) and ceases at or near the onset of E1. Br, which is the only single joint flexor muscle of elbow, is active just prior to foot lift and remains active until approximately midswing. Cleidobrachialis (ClB) and biceps brachii (Bit) normally commence their periods of locomotor activity at about the same time as Br. The activity in ClB, however, always continue throughout the period of swing. The overall muscle stimulation for forelimb during swing phase can be seen in the following equations:

$$S_{SF}(t) = S_{(0,SF} + G^{Y}_{SF} Y_{SF} + G^{l}_{SF}(l_{CE,SF} - l_{off,SF}) \tag{B.15}$$

$$S_{SE}(t) = S_{0,SE} - G^{Y}_{SE} Y_{SE} + G^{l}_{SE}(l_{CE,SF} - l_{off,SF}) \tag{B.16}$$

$$S_{EE}(t) = S_{0,SE} - G^{Y}_{EE} Y_{EE} + G^{l}_{EE} \cdot Y_{EE} \cdot (l_{EE} - (-l_{off,EF}) \tag{B.17}$$

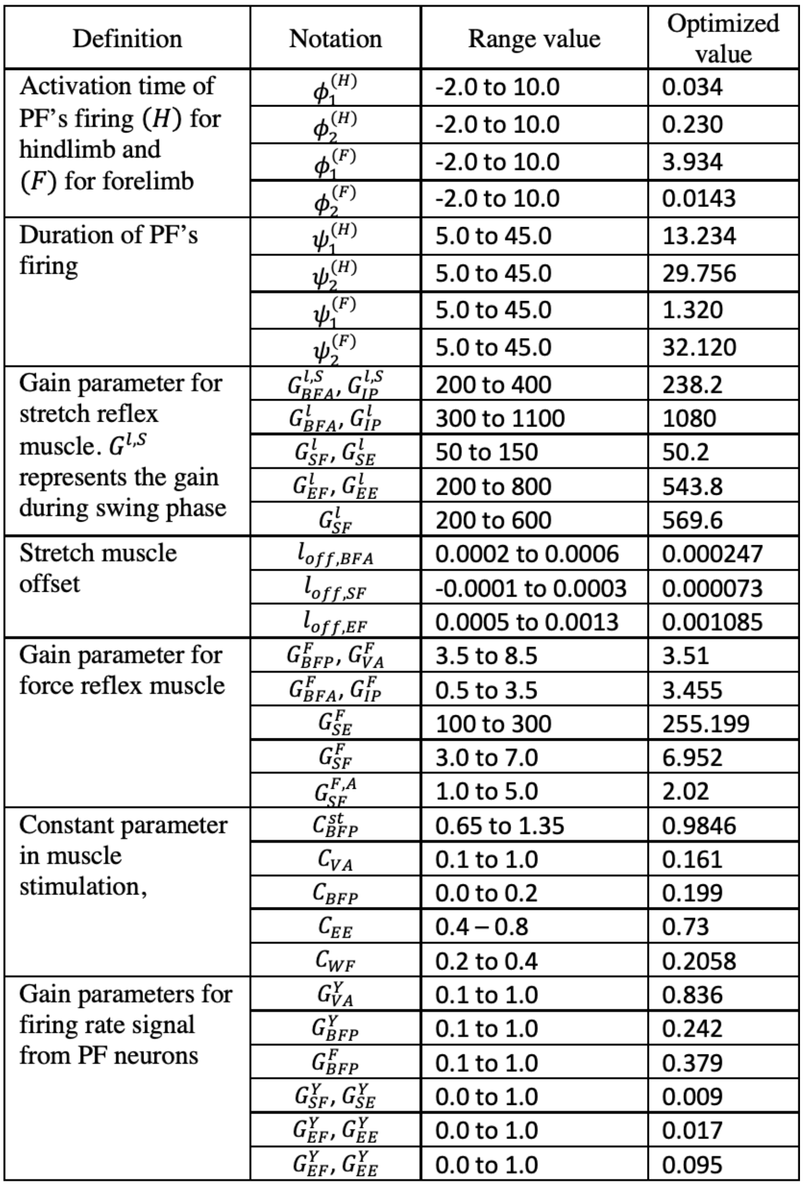
$$S_{EF}(t) = S_{0,EF} - G^{Y}_{EF}(1 - Y_{EF}) + G^{l}_{EF} \cdot Y_{EF} \cdot (l_{EF} - l_{off,EF}) \tag{B.18}$$

$$S_{WE}(t) = S_{0,WE} + G^{Y}_{WE} Y_{WE} \tag{B.19}$$

$$S_{WF}(t) = S_{0,WF} + G^{Y}_{WF}(1 - Y_{WF}) \tag{B.20}$$

Detail explanation of the parameters can be seen in Table. B.1.

Table B.1: *Range value of the optimized parameter*



Definition	Notation	Range value	Optimized value
Activation time of PF's firing (H) for hindlimb and (F) for forelimb	$\phi_1^{(H)}$	-2.0 to 10.0	0.034
	$\phi_2^{(H)}$	-2.0 to 10.0	0.230
	$\phi_1^{(F)}$	-2.0 to 10.0	3.934
	$\phi_2^{(F)}$	-2.0 to 10.0	0.0143
Duration of PF's firing	$\psi_1^{(H)}$	5.0 to 45.0	13.234
	$\psi_2^{(H)}$	5.0 to 45.0	29.756
	$\psi_1^{(F)}$	5.0 to 45.0	1.320
	$\psi_2^{(F)}$	5.0 to 45.0	32.120
Gain parameter for stretch reflex muscle. $G^{l,S}$ represents the gain during swing phase	$G_{BFA}^{l,S}, G_{IP}^{l,S}$	200 to 400	238.2
	G_{BFA}^{l}, G_{IP}^{l}	300 to 1100	1080
	G_{SF}^{l}, G_{SE}^{l}	50 to 150	50.2
	G_{EF}^{l}, G_{EE}^{l}	200 to 800	543.8
	G_{SF}^{l}	200 to 600	569.6
Stretch muscle offset	$l_{off,BFA}$	0.0002 to 0.0006	0.000247
	$l_{off,SF}$	-0.0001 to 0.0003	0.000073
	$l_{off,EF}$	0.0005 to 0.0013	0.001085
Gain parameter for force reflex muscle	G_{BFP}^{F}, G_{VA}^{F}	3.5 to 8.5	3.51
	G_{BFA}^{F}, G_{IP}^{F}	0.5 to 3.5	3.455
	G_{SE}^{F}	100 to 300	255.199
	G_{SF}^{F}	3.0 to 7.0	6.952
	$G_{SF}^{F,A}$	1.0 to 5.0	2.02
Constant parameter in muscle stimulation,	C_{BFP}^{st}	0.65 to 1.35	0.9846
	C_{VA}	0.1 to 1.0	0.161
	C_{BFP}	0.0 to 0.2	0.199
	C_{EE}	0.4 – 0.8	0.73
	C_{WF}	0.2 to 0.4	0.2058
Gain parameters for firing rate signal from PF neurons	G_{VA}^{Y}	0.1 to 1.0	0.836
	G_{BFP}^{Y}	0.1 to 1.0	0.242
	G_{BFP}^{F}	0.1 to 1.0	0.379
	G_{SF}^{Y}, G_{SE}^{Y}	0.0 to 1.0	0.009
	G_{EF}^{Y}, G_{EE}^{Y}	0.0 to 1.0	0.017
	G_{EF}^{Y}, G_{EE}^{Y}	0.0 to 1.0	0.095