



## OPEN ACCESS

## EDITED BY

Anastasio Dimou,  
Centre for Research and Technology  
Hellas (CERTH), Greece

## REVIEWED BY

Takeshi Kano,  
Tohoku University, Japan  
Ángel Soriano,  
Robotnik (Spain), Spain

## \*CORRESPONDENCE

Tetsushi Kamegawa,  
✉ kamegawa@okayama-u.ac.jp

## SPECIALTY SECTION

This article was submitted to  
Computational Intelligence in Robotics,  
a section of the journal Frontiers in  
Robotics and AI

RECEIVED 05 January 2023

ACCEPTED 10 March 2023

PUBLISHED 29 March 2023

## CITATION

Tamura H and Kamegawa T (2023),  
Parameter search of a CPG network  
using a genetic algorithm for a snake  
robot with tactile sensors moving on a  
soft floor.

*Front. Robot. AI* 10:1138019.

doi: 10.3389/frobt.2023.1138019

## COPYRIGHT

© 2023 Tamura and Kamegawa. This is  
an open-access article distributed under  
the terms of the [Creative Commons  
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,  
distribution or reproduction in other  
forums is permitted, provided the  
original author(s) and the copyright  
owner(s) are credited and that the  
original publication in this journal is  
cited, in accordance with accepted  
academic practice. No use, distribution  
or reproduction is permitted which does  
not comply with these terms.

# Parameter search of a CPG network using a genetic algorithm for a snake robot with tactile sensors moving on a soft floor

Hajime Tamura and Tetsushi Kamegawa\*

Graduate School of Interdisciplinary Science and Engineering in Health Systems, Okayama University, Okayama, Japan

When a snake robot explores a collapsed house as a rescue robot, it needs to move through various obstacles, some of which may be made of soft materials, such as mattresses. In this study, we call mattress-like environment as a soft floor, which deforms when some force is added to it. We focused on the central pattern generator (CPG) network as a control for the snake robot to propel itself on the soft floor and constructed a CPG network that feeds back contact information between the robot and the floor. A genetic algorithm was used to determine the parameters of the CPG network suitable for the soft floor. To verify the obtained parameters, comparative simulations were conducted using the parameters obtained for the soft and hard floor, and the parameters were confirmed to be appropriate for each environment. By observing the difference in snake robot's propulsion depending on the presence or absence of the tactile sensor feedback signal, we confirmed the effectiveness of the tactile sensor considered in the parameter search.

## KEYWORDS

snake robot, tactile sensor, CPG network, soft floor, genetic algorithm

## 1 Introduction

Snake robots are engineering mimics of biological snakes. Despite their simple shape, biological snakes are able to move flexibly and adapt to various environments, such as poorly secured ground or narrow spaces. These characteristics of snakes suggest that snake robots could move through environments that are difficult for conventional mobile robots to traverse, and snake robots are expected to be applied to various tasks [Hirose \(1993\)](#).

Rescue robots are one application of snake robots. In the case of snake robot moving in a complex and narrow environment, it is necessary to adapt its shape to the environment by using tactile sensors. Snake robots equipped with tactile sensors across their bodies have been developed [Liljebäck et al. \(2012\)](#); [Tadokoro \(2019\)](#); [Kano and Ishiguro \(2020\)](#); [Thandiackal et al. \(2021\)](#). These snake robots are expected to be used for rescue activities in collapsed houses at disaster sites damaged by earthquakes and tsunamis as fast as possible and as long time as possible with as little energy consumption as possible without robot failure, thereby reducing secondary disasters. When a snake robot searches a collapsed house, it must overcome various obstacles including made of soft materials, such as futons and mattresses. For example, [Micire \(2008\)](#) conducted a mission to explore a collapsed building

damaged by a hurricane with a robot and reported that a mattress in one bedroom became an obstacle to the robot's movement. However, few studies have been conducted on robots that propel themselves over such mattress-like soft obstacles. We call mattress-like environment as a soft floor, which deforms when some force is added to it. The final goal of this study is to apply a snake robot to special and complex environments such as disaster sites. On disaster sites, rather than designing optimal control strategies for a snake robot in one specific environment, it is necessary to seek control strategies for a robot that can run in a variety of situations, including environments that have not been covered in previous research. One of the characteristics of the environment that has not been covered in previous research [Liljebäck et al. \(2013\)](#) is a soft floor. While there has been some research on mathematical models of soft environments. For example, literature [Vogt et al. \(2017\)](#) describes a soft tube environment, in this study it was necessary to construct a new soft planar model that interacts with a moving robot in a dynamics simulator. We propose the modeling method for a soft floor. In addition, we show that the control parameters of a snake robot moving on a hard floor are not necessarily optimal when the robot moves on a soft floor.

In this study, we focused on the central pattern generator (CPG) network as a control strategy for a snake robot to move on a soft floor. The CPG network is a control method for robots that applies a neural circuit believed to be involved in the periodic motion of living organisms [Matsuoka \(1987\)](#); [Ijspeert \(2008\)](#); [Yu et al. \(2014\)](#). Previous studies of CPG networks for snake robots include a study that constructed a CPG network that generates lateral undulation propulsion for a 2D snake robot [Inoue et al. \(2004\)](#). It is believed that by adding information from the outside world as feedback signals to the CPG network, environmentally adaptive behavior could be generated. For example, a study has been conducted in which using friction force as feedback signals [Otaka et al. \(2018\)](#). A study that generates undulatory swimming for a snake robot by CPG network with feedback information of force sensors has also been conducted [Thandiackal et al. \(2021\)](#). In this study, we constructed a new CPG network for the snake robot with tactile sensors that can move in three dimensions, which has been developed by our research group [Tadokoro \(2019\)](#). The constructed CPG network enabled the snake robot to move with sidewinding locomotion. However, it is difficult to find the appropriate parameters of the CPG network analytically because there are many parameters and they are related each other complex. In a previous study, Inoue et al. used the genetic algorithm (GA) [Holland \(1992\)](#) to find the parameters in a simple CPG network for their 2D snake robot performing lateral undulation propulsion [Inoue et al. \(2007\)](#). In this study, we should find the CPG parameters for 3D snake robot including feedback from tactile sensor contact with environment. The feedback from tactile sensor makes the CPG network more complex to find appropriate parameters. We have constructed a model of the snake robot moving on a soft floor in the simulation environment and found the CPG parameter. To verify the obtained parameters, comparative simulations were conducted using the parameters obtained for the soft and hard floor, and the parameters were confirmed to be appropriate for each environment. By observing the difference in snake robot's propulsion depending on the presence or absence of the tactile sensor feedback signal, we confirmed the effectiveness of the tactile sensor considered in the parameter search.

## 2 A model of the snake robot and the environment

### 2.1 A model of the snake robot

A model of the snake robot was built on a simulator for the simulation. CoppeliaSim version 4.2.0 (Coppelia Robotics, Ltd., Switzerland) was used as the robot simulator, and the Open Dynamics Engine was used as the physics engine.

**Figure 1** shows the snake robot model built on the simulator. The snake robot has a total length of 1.35 m, a weight of 12.3 kg, a diameter of 0.09 m, a distance between joints of 0.09 m, 14 joints, and 15 tactile sensors. The diameter of the snake robot is the diameter of the outer circumference of the tactile sensor.

#### 2.1.1 Joint configuration of the snake robot

Snake robots are elongated robots with multiple active joints connected in series. In this study, the joint configuration is based on alternating connections of the pitch and yaw axes, which have been widely used in snake robots developed to date. The first joint is assumed to be a pitch-axis joint. The joints are numbered as "0, 1, 2, ..." starting from the first joint, and the joint numbered  $i$  is called joint  $i$ . The even-numbered joints are the pitch-axis joints, and the odd-numbered joints are the yaw-axis joints. The range of a joint is  $\pm 81.9$  deg, this is due to contact of the adjacent links. The maximum torque is set as 8.4 Nm and the maximum angular speed is set as 360.0 deg/s.  $\theta_i$  is the target joint angle, and control to the target angle is controlled by the PID controller in the joint model of the robot simulator CoppeliaSim. In this study, the gains are set to  $p = 1.0$ ,  $I = 0.0$ , and  $D = 0.0$ , making it a simple proportional control.

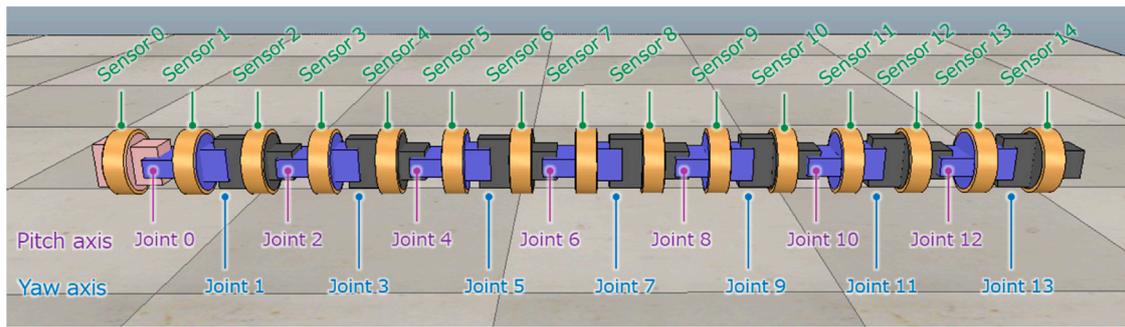
#### 2.1.2 Tactile sensor configuration of the snake robot

The tactile sensor model used in this study is a model of a center of pressure (CoP) sensor. A CoP sensor has a pressure-sensitive sheet on its circumference and can measure the position and magnitude of the applied force on the circumference [Tadokoro \(2019\)](#). Tactile sensors are attached to the center of each link to cover that link. When tactile sensors contact the external environment, the contact force can be obtained as a sensor value. The links are numbered as "0, 1, 2, ..." starting from the first link, and the link numbered  $i$  is called the link  $i$ . The tactile sensor attached to link  $i$  is called sensor  $i$ . The detail of the CoP sensor is described in the literature [Tadokoro \(2019\)](#). The simulation model in this study, it is modeled as having very high resolution which returns value of double precision floating point number.

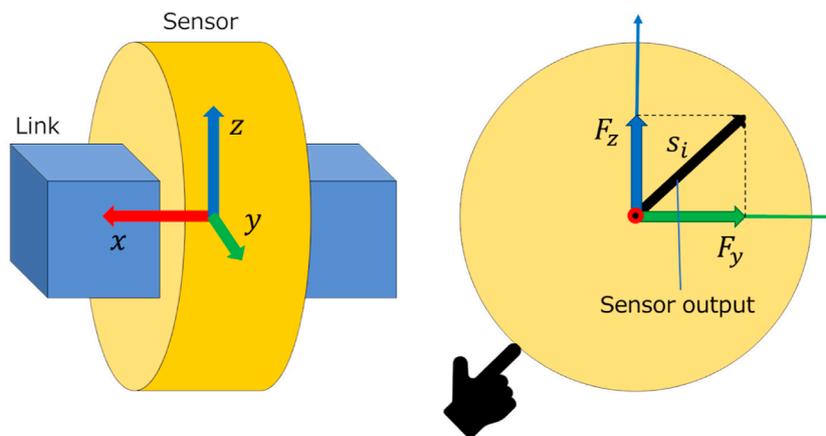
The tactile sensor model was simulated using the model of the force sensor in CoppeliaSim. The coordinate of the force sensor is set as **Figure 2** shows, and the components  $F_y$  and  $F_z$  of the force are obtained when contact occurs.  $s_i = \sqrt{F_y^2 + F_z^2}$  denotes the output  $s_i$  of sensor  $i$ . Note that the component of  $F_x$  is not used in this study.

### 2.2 Model of the soft floor

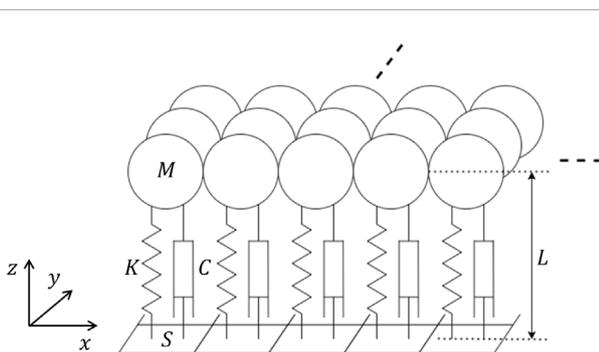
In this study, the small units that make up a soft floor are modeled as a spring-mass-damper system. A conceptual diagram of



**FIGURE 1**  
Simulation model of the snake robot.



**FIGURE 2**  
Model of tactile sensor.



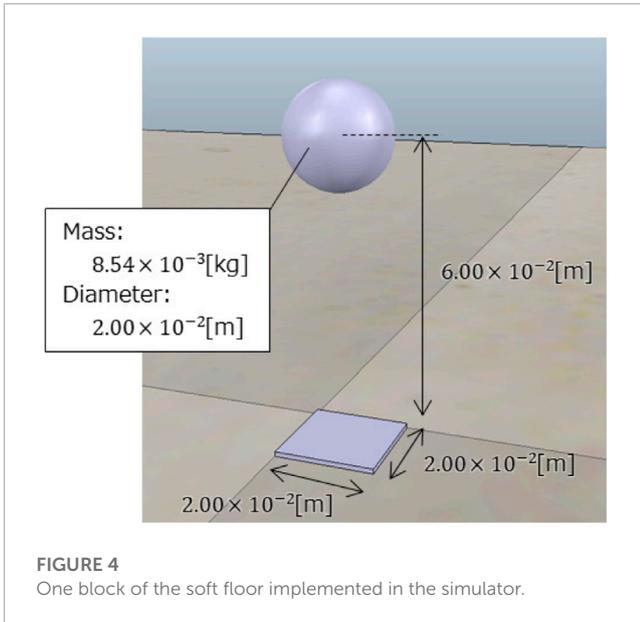
**FIGURE 3**  
Soft floor model with a spring-mass-damper system.

the model is shown in **Figure 3**. Each block consists of one spring-mass-damper system. In our study,  $L$  is the thickness of the soft floor,  $S$  is the area of one block viewed from above,  $M$  is the mass of one block,  $K$  is the spring constant of one block, and  $C$  is the damping coefficient of one block.

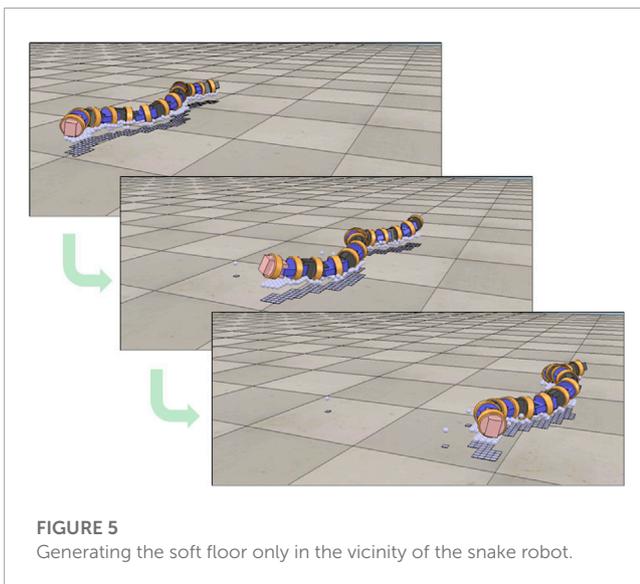
The spring constant  $K$  of this model can be expressed using Young's modulus  $E$  as follows:

$$K = \frac{S}{L}E. \tag{1}$$

The mass, spring, and damper system of one block satisfy the critical damping condition  $C = 2\sqrt{MK}$  so that it does not vibrate because this study focus on mattress-like soft obstacles as described in the introduction.  $L$  was set to a value sufficiently large for deformation volume that occurs when the snake robot is on a soft floor.  $S$  was set to a reasonable value considering the simulation accuracy and computation time. Although  $S$  should be as small as possible from the standpoint of simulation accuracy, too small  $S$  requires many blocks to support the snake robot, which increases the computation time. In this study,  $S$  was set to be as small as possible so that the simulation could be completed in realistic time.  $M$  was set to a value small enough relative to the mass of the snake robot. Specifically, we estimated the mass that a single block would have to support when the snake robot is placed on the soft floor in a straight line and set  $M$  to one-tenth of that mass. The reason for the sphere shape of the soft floor blocks is to eliminate the corners of the blocks and to avoid snagging between the snake robot and the soft floor.



**FIGURE 4**  
One block of the soft floor implemented in the simulator.



**FIGURE 5**  
Generating the soft floor only in the vicinity of the snake robot.

One block of the soft floor implemented in the simulator is shown in **Figure 4**. For implementation, the Young’s modulus of the soft floor was set to  $E = 8.75 \times 10^3 \text{ Pa}$ , referring to a mattress used in a paper on the mechanical properties of mattresses **Inoue and Yamada (2016)**. Other parameters were as follows:  $L = 6.00 \times 10^{-2} \text{ m}$ ,  $S = 4.00 \times 10^{-4} \text{ m}^2$ , and  $M = 8.54 \times 10^{-3} \text{ kg}$ . Note that the simulated part of the soft floor surface was spherical, with a  $2.00 \times 10^{-2} \text{ m}$  diameter. The interaction between adjacent spheres is not considered in this study, and the soft floor is modeled as being deformed only in the vertical direction by a vertical downward force. In addition, the spheres have a set mass and are subjected to gravity in the dynamics simulator. The natural length of the spring is adjusted so that the thickness is  $6.00 \times 10^{-2} \text{ m}$  when there is nothing on the sphere under gravity.

The soft floor was modeled by arranging several of the single blocks shown in **Figure 4** without gaps between them. **Figure 5**

shows the soft floor and the snake robot in the simulator. It can be seen that the blocks are clustered together to form the floor that supports the snake robot. The friction was set to 1.00 which is the default value of physics engine ODE to the snake robot and the soft floor surface, respectively. This allows the snake robot to be propelled almost without slipping on the soft floor surface as far as observing the behavior of the simulation.

### 3 CPG network

The CPG network constructed in this study is an extension of the CPG network that generates the sidewinding propulsion of a snake robot **Tamura et al. (2020)** and is shown in **Figure 6**.

Let  $CPG_i$  be the CPG model corresponding to joint  $i$  of the snake robot. There is only one CPG model that does not correspond to any joint, which is  $CPG_{top}$ . The CPG model corresponding to the pitch-axis joint is called  $CPG_{pitch}$ , and the CPG model corresponding to the yaw-axis joint is called  $CPG_{yaw}$ .

For the CPG model in this study, Matsuoka’s model **Matsuoka (1987)** (**Figure 7**) is used, and the relationship between the output  $y_i$  of  $CPG_i$  and the angle  $\theta_i$  of joint  $i$  is assumed to be  $\theta_i = y_i$ . The internal parameters  $\beta$ ,  $w_{fe}$ ,  $\tau$ ,  $\tau'$ , and  $u_0$ , which are constant parameters determine the behavior of this CPG model.

The values of the internal parameters used are also shown in **Figure 7**. These values were obtained in preliminary study **Tamura et al. (2020)** that heuristically determine the parameters for the snake robot to be propel by sidewinding on a hard floor. Although  $u_0$  is originally given externally as an input from the upper center, it is classified as an internal parameter and as a constant in this study because it is not assumed that the behavior of the snake robot changes by the command from upper central nervous system.

The CPG network is formed by connecting the CPG models to each other. Each CPG model affects up to two CPG models ahead by excitatory connections. The connection weights, which determine the strength of the connections between CPGs, are  $W_{t-p}$  from  $CPG_{top}$  to  $CPG_{pitch}$ ,  $W_{t-y}$  from  $CPG_{top}$  to  $CPG_{yaw}$ ,  $W_{p-y}$  from  $CPG_{pitch}$  to  $CPG_{yaw}$ ,  $W_{p-p}$  from  $CPG_{pitch}$  to  $CPG_{pitch}$ ,  $W_{y-p}$  from  $CPG_{yaw}$  to  $CPG_{pitch}$ , and  $W_{y-y}$  from  $CPG_{yaw}$  to  $CPG_{yaw}$ . Each CPG model is also connected to tactile sensors attached to the links before and after the joint to which it corresponds. The feedback weights, which determine the strength of the feedback signal, are  $F_{p-f}$  and  $F_{y-f}$  for the front side and  $F_{p-b}$  and  $F_{y-b}$  for the back side for  $CPG_{pitch}$  and  $CPG_{yaw}$ , respectively. Using these feedback weights, the feedback signals  $Feed_i^f$  and  $Feed_i^e$  of the CPG model are as follows:

$$Feed_i^f = Feed_i^e = F_{p-f}s_i + F_{p-b}s_{i+1} \tag{2}$$

where  $s_i$  is the output of sensor  $i$ . When  $Feed_i^f$  equals to  $Feed_i^e$ , a snake robot moves straight.

### 4 Parameter search using a GA

As usual, for this study the optimization problem of the CPG network which has a high number of parameters has been addressed using GA. The details are shown below.

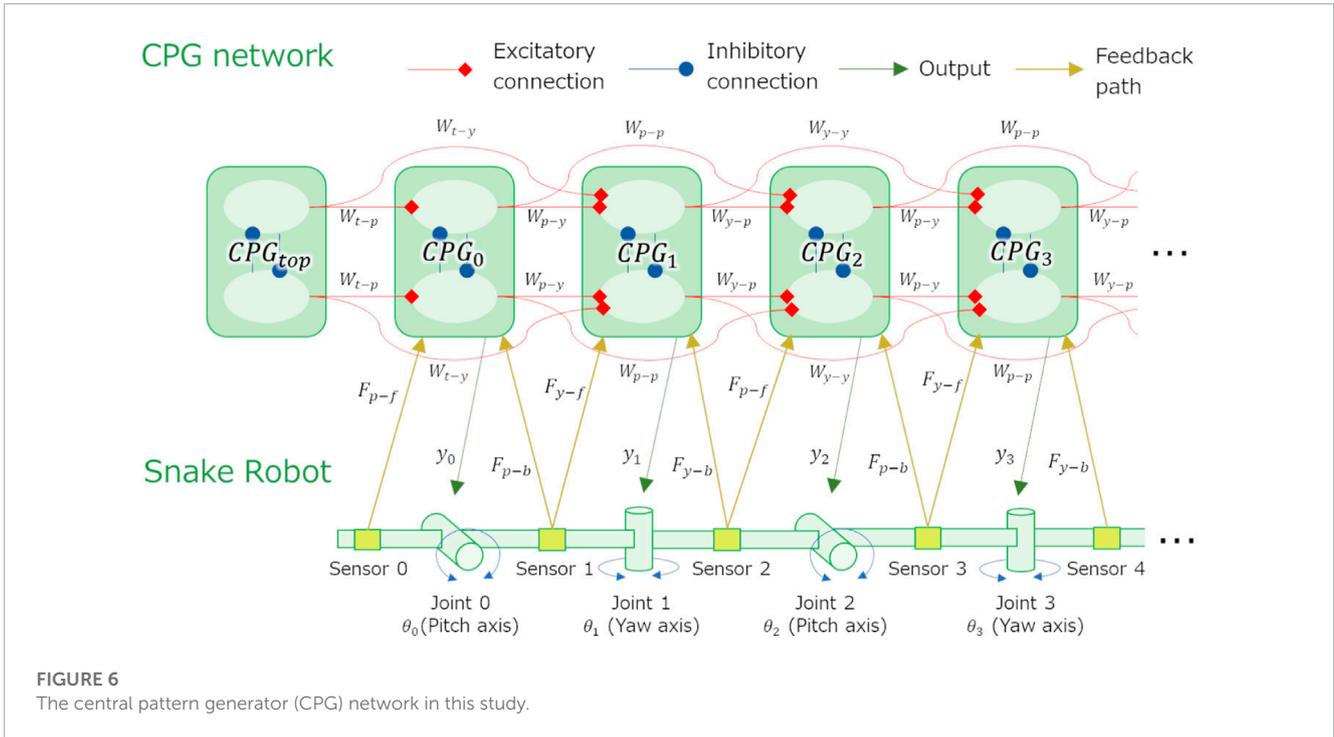


FIGURE 6 The central pattern generator (CPG) network in this study.

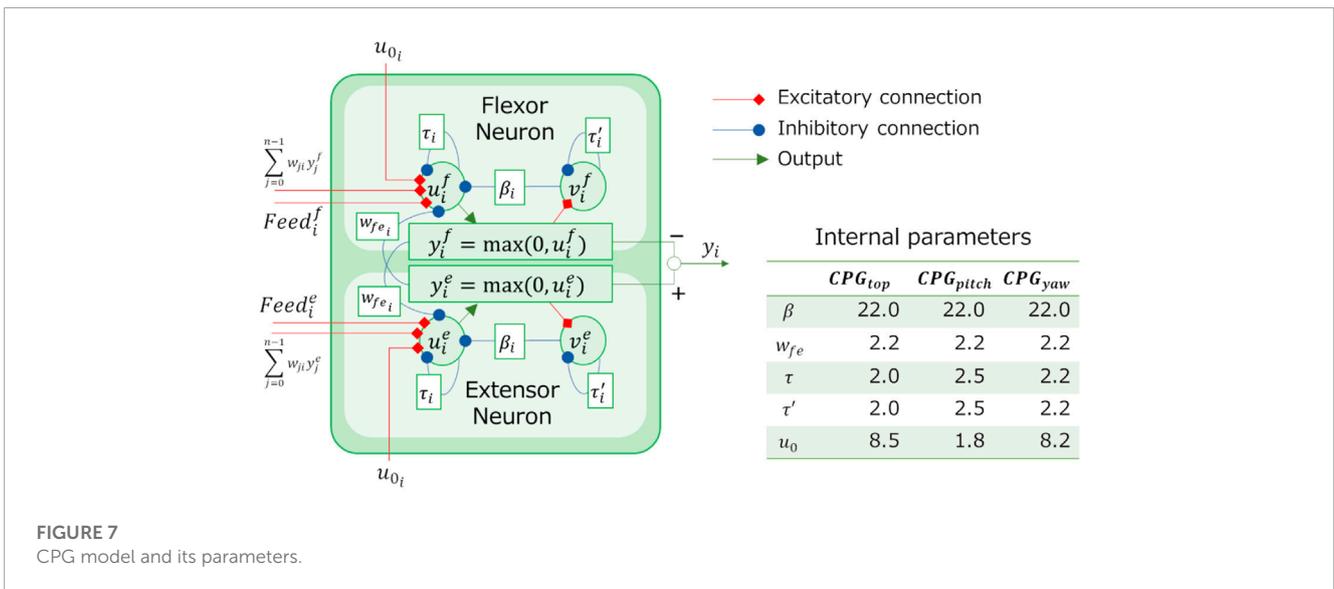


FIGURE 7 CPG model and its parameters.

### 4.1 Problem setting

The parameter search problem is a minimization problem for an objective function  $f(\mathbf{a})$  whose argument is a vector  $\mathbf{a} = (a_1, a_2, a_3, \dots, a_n)$  of parameters to be searched. In this study, the objective function is calculated using the results of moving a snake robot on a soft floor for a certain period of time in a simulator (Figure 8).

#### 4.1.1 Setting of search parameters

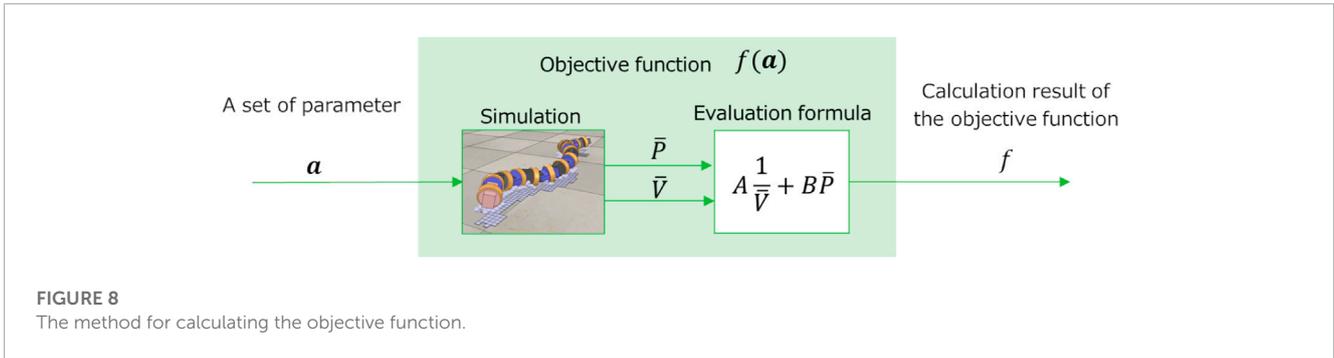
In this study, the search parameter  $\mathbf{a}$  is the sequence of parameters of the CPG network. Because it is difficult to search for all

parameters of the CPG network, we searched for only the connection weights and feedback weights of the CPG network parameter. In other words,  $\mathbf{a}$  is a sequence of the parameters illustrated in Figure 6, which is expressed as follows:

$$\mathbf{a} = (W_{t-p}, W_{t-y}, W_{p-y}, W_{p-p}, W_{y-p}, W_{y-y}, F_{p-f}, F_{p-b}, F_{y-f}, F_{y-b}). \tag{3}$$

#### 4.1.2 Setting of the objective function

For a mobile snake robot used in rescue operation, it is important to increase the speed of movement. Thus, a propulsive motion with high speed can be regarded as a good motion. In addition, because



**FIGURE 8**  
The method for calculating the objective function.

battery consumption should be minimized when the snake robot moves, a propulsive motion with low power consumption can be regarded as a good motion. Accordingly, we define the objective function  $f$  in this study as follows.

$$f = A \frac{1}{\bar{V}} + B\bar{P}, \tag{4}$$

$$\bar{V} = \frac{|\mathbf{d}|}{T}, \tag{5}$$

$$\bar{P} = \frac{W}{T}, \tag{6}$$

Where  $\bar{V}$  and  $\bar{P}$  are the respective average movement speed and average power consumption of the snake robot when it is moved on a soft floor for a certain time,  $A$  and  $B$  are the weights of each term,  $T$  is the duration of time the robot moved on the soft floor,  $\mathbf{d}$  is the displacement of the snake robot's center of gravity between time 0 and  $T$  (see Figure 9), and  $W$  is the total power consumption of the snake robot between time 0 and  $T$ . It is assumed that we are going to conduct an experiment using a real snake robot which joints are driven by DC motor. We are going to measure voltage and current value of the DC motor in the experiment. In this study, we also use the voltage and estimate the current of the motor in the simulation as following. The total power consumption of the snake robot is defined as the sum of the power consumed by the motors that drive the joints of the snake robot, and is calculated using the following equations.

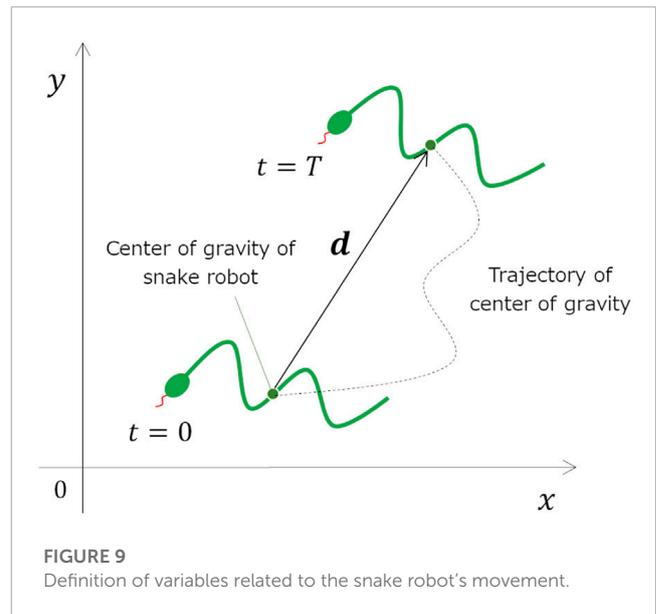
$$W = \sum_{i=0}^{N-1} \int_0^T |E_v I_i(t)| dt, \tag{7}$$

$$I_i = \lambda \tau_i, \tag{8}$$

Where  $N$  is the number of snake robot joints,  $E_v$  is the power supply voltage,  $I_i$  is the current flowing in the motor of joint  $i$ ,  $\tau_i$  is the torque of joint  $i$ , and  $\lambda$  is the proportionality constant between the current and torque of the motor.  $W$  is calculated using the torque that can be obtained in the simulator according to Eq. 8. The values of  $E_v$  and  $\lambda$  were set to  $E_v = 12.0V$  and  $\lambda = 0.6548A/(N \cdot m)$ , following the specifications of the motor used in the actual snake robot owned by our laboratory.

### 4.2 Setup of the GA

Because this study considers the problem of minimizing the objective function, reversing the magnitude of the objective function and the fitness of the GA is necessary. Thus, the



**FIGURE 9**  
Definition of variables related to the snake robot's movement.

fitness of the GA was set to  $fitness = -f$  using the objective function  $f$ .

This study used a genotype in which the search parameters defined in Eq. 3 are expressed as a bit string and are arranged in a sequence. Each parameter was set to 12 bits, with  $W_{t-p}$ ,  $W_{t-y}$ ,  $W_{p-y}$ ,  $W_{p-p}$ ,  $W_{y-p}$ , and  $W_{y-y}$  as values between 0.00 and 2.00 and  $F_{p-f}$ ,  $F_{p-b}$ ,  $F_{y-f}$ , and  $F_{y-b}$  as values between -0.20 and 0.20. The range of parameters are decided according to a preliminary study so that the robot can move by keeping periodical motion.

In this study, ranking selection was used for selection to speed up the convergence of GA solutions; uniform crossover, for crossover; and simple mutation, for mutation as the genetic manipulation of the GA. The details of the ranking selection used in this study are described below. The best individual is ranked as the 0th, followed by the first, second, and so on in order of fitness. Let  $p_k$  be the selection probability of the  $k$ th individual. A truncated rank  $k_t$  is defined so that the selection probability of individuals with a rank lower than  $k_t$  is zero. Subsequently, the selection probabilities of individuals from the 0th to the  $k$ th ranks are decreased at a constant rate.

## 5 Simulation

### 5.1 Simulation of parameter search using the GA

We conducted simulations to determine the parameters of the CPG network using the GA described in Chapter 4. The snake robot was operated for a certain period of time on a soft floor and a hard floor, successively, and the objective function  $f$  was calculated. In order to determine the parameters for hard and soft floors, the hard floor was implemented also in the simulation environment. The hard floor was realized without the vertical movement of the spring and damper described in Section 2.2. The search on the soft floor was referred to as the soft floor condition, the search on the hard floor was referred to as the hard floor condition, and the parameters were obtained using a GA for each condition.

The parameters included in the objective function equations (Eqs. 4–6) are as follows: the weight of the average speed term,  $A = 100$ ; the weight of the average power consumption term,  $B = 1$ ; and the robot’s operating time,  $T = 30$ s. The weight  $A$  and  $B$  was decided according to a preliminary study. In the preliminary study, the snake robot moved in typical sidewinding motion on the soft floor, and its average movement speed and average power consumption were recorded. Based on the recorded value, the weight of  $A$  and  $B$  were adjusted so that the terms were of similar magnitude. The weights  $A$  and  $B$  are adjusted according to a robot’s application. For example, if power consumption or actuator load is more important than travel speed, the value of  $B$  can be adjusted to a larger value. The flow and parameters of the GA are shown in Figure 10. The elitism was adopted in the selection of genetic operations.

TABLE 1 Parameters obtained using the GA.

Parameters	The soft floor condition	The hard floor condition
$W_{t-p}$	0.553	0.912
$W_{t-y}$	0.0215	0.0518
$W_{p-y}$	0.0694	0.0186
$W_{p-p}$	1.30	0.871
$W_{y-p}$	0.752	1.40
$W_{y-y}$	1.32	0.915
$F_{p-f}$	0.0184	0.00650
$F_{p-b}$	0.0608	0.0426
$F_{y-f}$	-0.182	0.0171
$F_{y-b}$	0.162	0.0185

Table 1 shows the results of the parameter search using the GA. It can be seen that different parameters were obtained when searching under the soft and hard floor conditions. The feedback weights tend to be smaller on a hard floor than on a soft floor. This is because the feedback signal from the tactile sensor tends to increase momentarily on hard floors due to collisions between the robot and the floor, and the weights are adjusted to reduce the effect of such collisions. Figure 11 shows the change in calculation result of the objective function for each number of generations during GA execution. In addition, Figure 12 and 13 show how the snake robot was propelled with the parameters obtained from the GA. In both conditions, the snake robot remains the starting point with the parameters before the search. However, with the parameters after the search, the snake robot propelled efficiently by sidewinding propulsion in each condition as shown in Figure 12 and 13.

### 5.2 Simulations to validate the obtained parameters

The parameters obtained for the soft and hard floor conditions were used to compare the propulsion performance on the soft and hard floors, respectively. To confirm the effect of the feedback signal, a simulation in which the snake robot was moved by cutting off the feedback signal was conducted. The objective function in Eq. 4 was used as a measure of propulsion performance. Simulations were conducted five times under each condition, and the averages were compared. Figure 14 shows the results of the simulation. Each graph compares the value of the objective function  $f$ , the robot’s average velocity  $\bar{V}$  and average of power consumption  $\bar{P}$  when the snake robot moves on the soft floor and hard floor with the obtained parameters. For each graph, smaller  $f$  is better, larger  $\bar{V}$  is better, and smaller  $\bar{P}$  is better. On the soft floor, the movement by the parameters under the soft floor condition reduced the objective function value. Similarly, on the hard floor, the movement by the parameters under the hard floor conditions reduced the objective function value. When the feedback signal was cut off, the objective function values were larger than those when the feedback signal was present on both the soft and hard floors, indicating that the snake

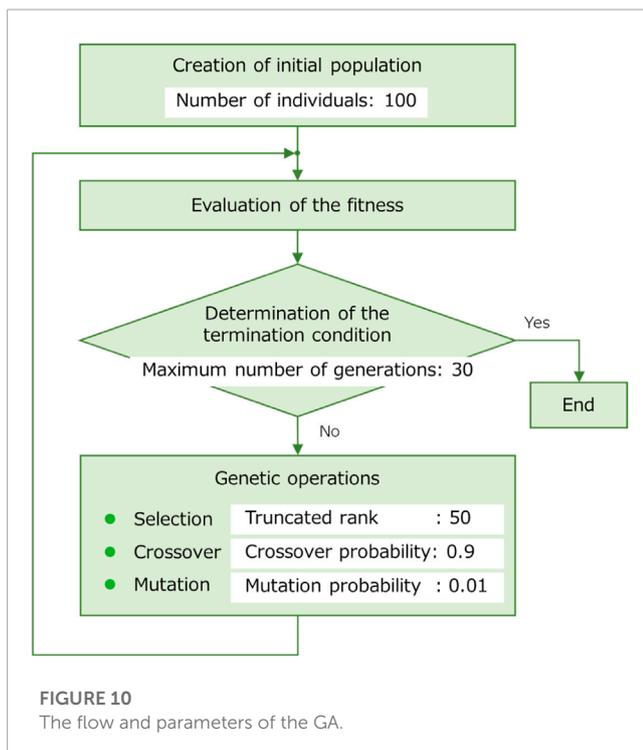
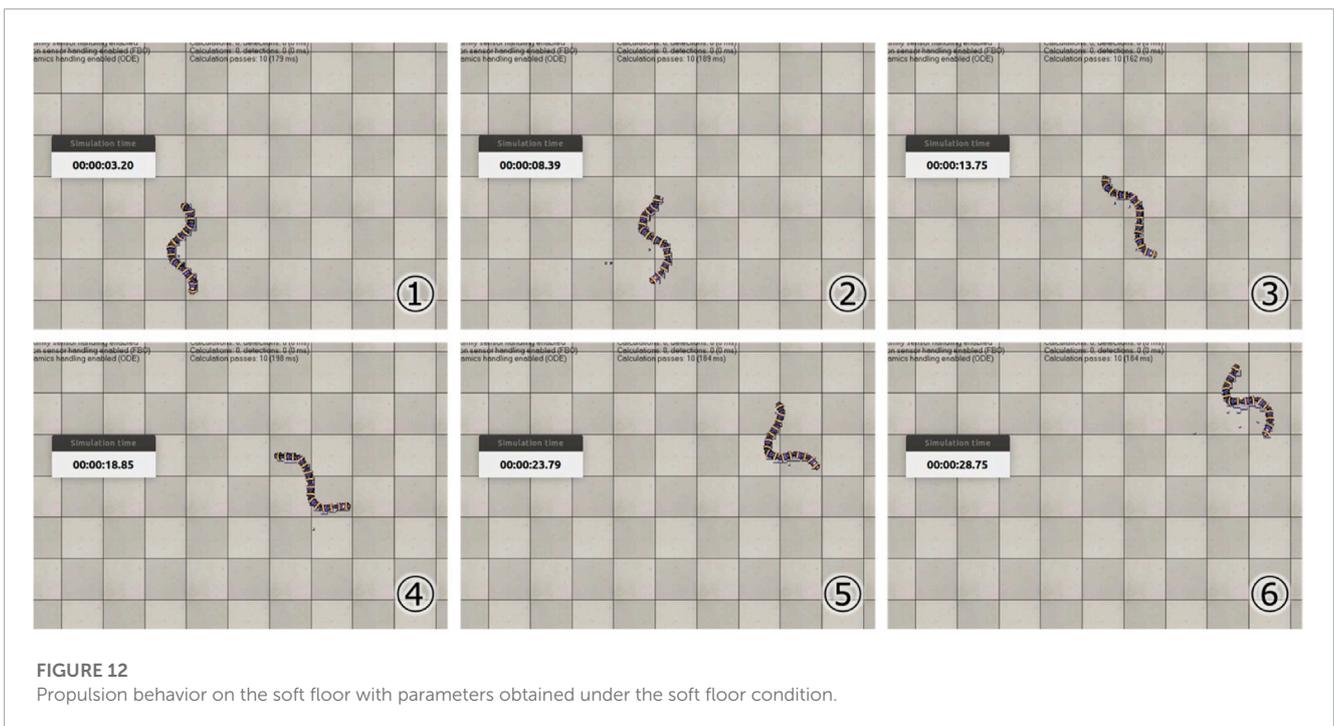
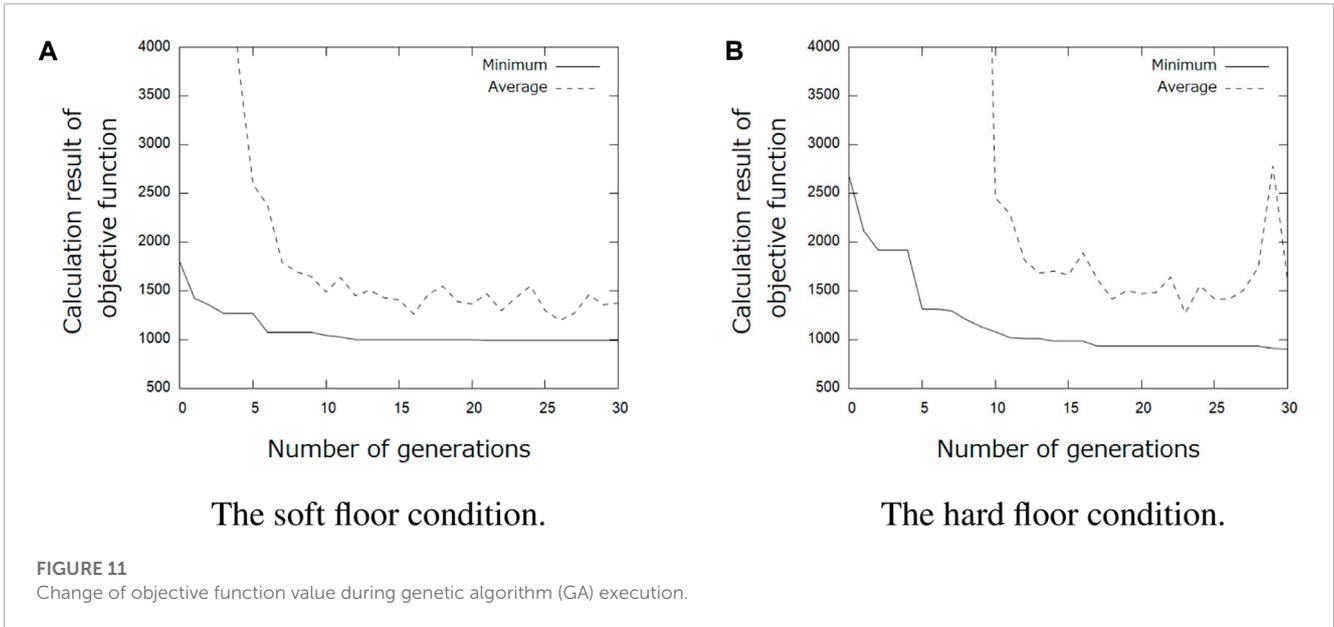


FIGURE 10 The flow and parameters of the GA.



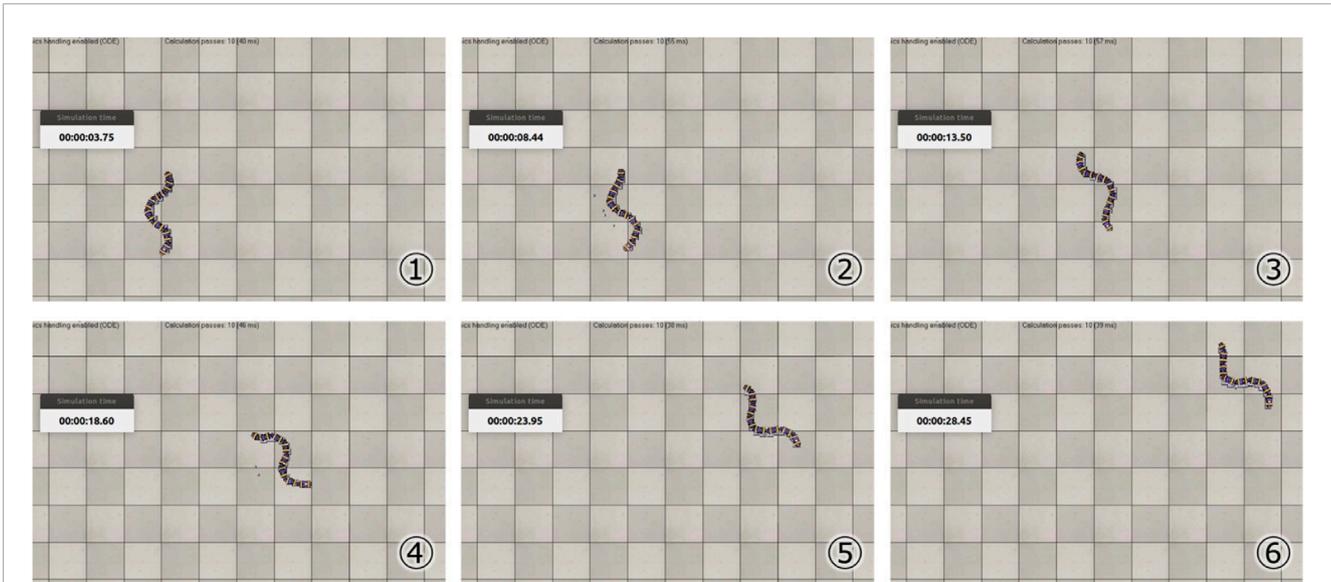
robot was unable to move appropriately because of the loss of tactile sensor feedback.

### 5.3 Discussion

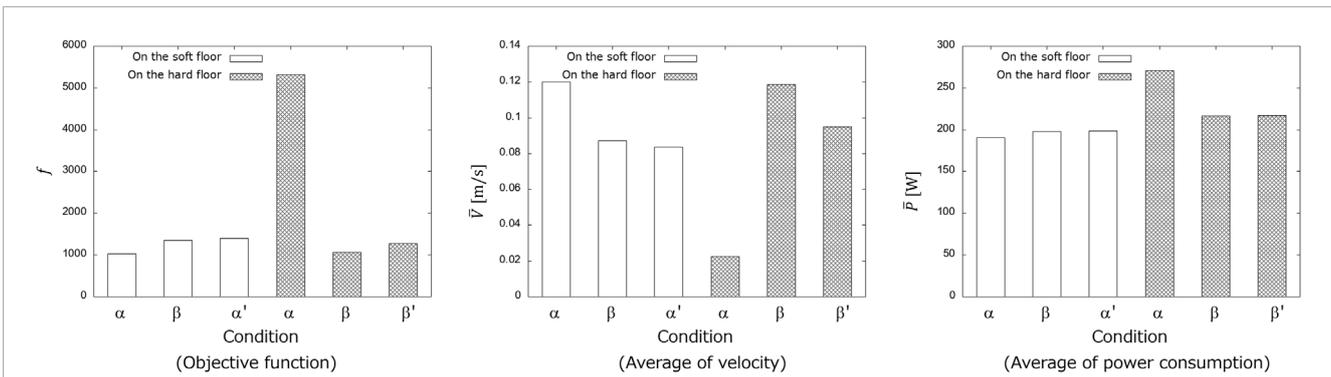
We assume that the snake robot began sidewinding propulsion with the obtained parameters because the parameters of the preliminary study were used for the internal parameters of the CPG

model. If the internal parameters of the CPG model can also be searched using the GA in future studies, a different propulsion behavior from the one observed in this study could be generated.

The results in **Figure 14** showed that the appropriate parameters for the CPG network vary depending on the environment and that the GA could search for suitable parameters for the soft floor. In addition, it was confirmed that propulsion efficiency changes depending on the presence or absence of feedback signals, indicating that the GA can search for parameters that consider the effect



**FIGURE 13** Propulsion behavior on the hard floor with parameters obtained under the hard floor condition.



**FIGURE 14** Results of propulsion experiment. Condition  $\alpha$ ,  $\beta$ ,  $\alpha'$  and  $\beta'$  are in parameters obtained under the soft floor condition, obtained under the hard floor condition, obtained under the soft floor condition when feedback was cut off, obtained under the hard floor condition when feedback was cut off, respectively.

of feedback signals. These results showed that the appropriate propulsion of a snake robot depends not only on the robot's internal parameters but also on the interaction between the robot and the environment.

## 6 Conclusion

In this study, we constructed a CPG network for a snake robot equipped with tactile sensors and searched for parameters to propel the snake robot on the soft floor. The models of the snake robot and the soft floor were simulated so that the objective function in the parameter search could be calculated based on the simulation results. To use the GA for parameter search, fitness, genotype, and genetic manipulation were considered. Parameter

search simulations were conducted using the constructed software in two different environments, a soft floor and a hard floor, and behaviors of the snake robot were compared using the obtained parameters. As a result of the simulation, the parameters of the CPG network suitable for the soft floor were obtained.

One of the future tasks is to investigate a method to adaptively operate the snake robot on floors with different softness and friction. Although this study explored the parameters of the CPG network considering the feedback signals from the sensors, it was for the specific given environment. CPG-based gait generation is expected to allow a snake robot to adapt to environments and move while updating internal parameters in real time, even during periodic movements. We believe that by developing an objective function that considers the different softness and different coefficient of friction of the floor, the snake robot will be able to adaptively move to

the floor. Another future task includes searching for the internal parameters of the CPG model, which were not searched in this study, to generate propulsive motions that are more suitable for soft floors. By the further work, we expect that a different propulsive motion from sidewinding propulsion, which is the motion obtained in this study, will emerge. In addition, the proposed method will be implemented to a real snake robot and should be verified by conducting experiments. Ultimately, a snake robot will be used as a rescue robot in real-world complex environments.

## Data availability statement

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Author contributions

HT and TK contributed to conception and design of the study. HT created the program and ran the simulation. HT visualized the simulation results. HT wrote the first draft of the manuscript. TK reviewed and edited the manuscript. TK raised funds and supervised the project. All authors contributed to manuscript revision, read, and approved the submitted version.

## References

- Hirose, S. (1993). *Biologically inspired robots: Snake-like locomotors and manipulators*. Oxford, United Kingdom: Oxford University Press.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge: MIT Press.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Netw.* 21, 642–653. doi:10.1016/j.neunet.2008.03.014
- Inoue, K., Ma, S., and Jin, C. (2004). “Neural oscillator network-based controller for meandering locomotion of snake-like robots,” in IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, New Orleans, LA, USA, 26 April 2004 - 01 May 2004, 5064–5069. doi:10.1109/ROBOT.2004.1302520
- Inoue, K., Sumi, T., and Ma, S. (2007). “CPG-based control of a simulated snake-like robot adaptable to changing ground friction,” in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October 2007 - 02 November 2007, 1957–1962. doi:10.1109/IROS.2007.4399552
- Inoue, Y., and Yamada, H. (2016). Investigation of pressure redistribution for bony protrusions in polyurethane foam mattresses by mechanical tests and hyperelastic modeling. *SOBIM* 23, 233–242. doi:10.3951/biomechanisms.23.233
- Kano, T., and Ishiguro, A. (2020). Decoding decentralized control mechanism underlying adaptive and versatile locomotion of snakes. *Integr. Comp. Biol.* 60, 232–247. doi:10.1093/icb/icaa014
- Liljebäck, P., Pettersen, K. Y., Stavadahl, Ø., and Gravidahl, J. T. (2012). Snake robot locomotion in environments with obstacles. *IEEE/ASME Trans. Mechatronics* 17, 1158–1169. doi:10.1109/TMECH.2011.2159863
- Liljebäck, P., Pettersen, K. Y., Stavadahl, Ø., and Gravidahl, J. T. (2013). *Snake robots - modeling mechatronics, and control*. Berlin, Germany: Springer.
- Matsuoka, K. (1987). Mechanisms of frequency and pattern control in the neural rhythm generators. *Biol. Cybern.* 56, 345–353. doi:10.1007/BF00319514
- Micire, M. J. (2008). Evolution and field performance of a rescue robot. *J. Field Robotics* 25, 17–30. doi:10.1002/rob.20218
- Otaka, K., Ibrahim, A. N. b., Inoue, K., and Fukuoka, Y. (2018). “Adaptive serpentine locomotion of a snake robot involuntarily activated and modified through sensory feedback affected by longitudinal frictional anisotropy,” in 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018, 1795–1801. doi:10.1109/ROBIO.2018.8664830
- Tadokoro, S. (2019). *Disaster Robotics: Results from the ImPACT tough Robotics challenge*. Berlin, Germany: Springer.
- Tamura, H., Kamegawa, T., and Gofuku, A. (2020). Generation of sidewinding by snake robot introducing cpg network (in Japanese). *SICE SI* 2020, 284–289.
- Thandiackal, R., Melo, K., Paez, L., Herault, J., Kano, T., Akiyama, K., et al. (2021). Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing. *Sci. Robotics* 6, eabf6354. doi:10.1126/scirobotics.abf6354
- Vogt, W., Steigenberger, J., and Maïßer, P. (2017). Quasistatic inflation processes within compliant tubes. *ZAMM - J. Appl. Math. Mech.* 97, 973–989. doi:10.1002/zamm.201500276
- Yu, J., Tan, M., Chen, J., and Zhang, J. (2014). A survey on CPG-inspired control models and system implementation. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 441–456. doi:10.1109/TNNLS.2013.2280596

## Funding

This work was supported by JSPS Research Grant 20K04380.

## Acknowledgments

We would like to thank Akio Gofuku for the useful discussions.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Appendix A

Details of the CPG model used in this study are described below. The CPG model used in this study is Matsuoka's CPG model (Figure 7). The output of the CPG model is the difference between the outputs of the extensor and flexor neurons. The dynamics of the CPG model can be expressed mathematically by Eq. 9.

$$\begin{cases} \tau_i \dot{u}_i^{\{f,e\}} + u_i^{\{f,e\}} = \sum_{j=1}^n w_{ji} y_j^{\{f,e\}} - w_{fe} y_i^{\{e,f\}} - \beta_i v_i^{\{f,e\}} + u_{0_i} + Feed_i^{\{f,e\}} \\ y_i^{\{f,e\}} = \max(0, u_i^{\{f,e\}}) \\ \tau_i^f \dot{v}_i^{\{f,e\}} + v_i^{\{f,e\}} = y_i^{\{f,e\}} \\ y_i = y_i^e - y_i^f \end{cases} \quad (9)$$

where the subscripts *e* and *f* denote flexor and extensor neurons,

respectively, and the subscripts *i* and *j* denote those of the *i*th or *j*th CPG.  $u_i^{\{f,e\}}$  is the state of membrane potential of a CPG neuron,  $v_i^{\{f,e\}}$  is the state of autoinhibition of a CPG neuron,  $y_i^{\{f,e\}}$  is the output of the CPG neuron,  $u_{0_i}$  is the driving input from the upper neuron of the CPG,  $w_{fe}$  is the mutual inhibition coefficient between extensor and flexor neurons,  $y_i$  is the output of the CPG,  $y_j^{\{f,e\}}$  is the input from the other CPG,  $n$  is the number of CPG model,  $w_{ji}$  is the connection weight from *j*th to *i*th CPG model  $Feed_i^{\{f,e\}}$  is the feedback signal from the sensor.