UNSYS digital

International Journal of Robotics and Mechatronics

Adjustment Method of Phase Lag Using Neural Oscillator Network for a Snake-like Robot

Takayuki Matsuo[†], Takahiro Kakigi[†], Takashi Sonoda[§], and Kazuo Ishii[§]

[†]Kitakyushu National College of Technology, Japan

[§]Kyushu Institute of Technology, Japan

***Abstract*—**This paper presents a design method for a phase lag adjustment system using neural oscillators for a snake-like robot. Robots are expected to be tools for operations and observations in extreme environments that humans cannot directly access without difficulty. The adaptability of robots is a very promising feature. Recently, development of biomimetic control systems has been inspired by the properties of animals, such as the brain and nervous systems, motor systems, and so on. In this paper, we present an adaptive control system based on the CPG consisting of neural oscillators, which generate rhythmical signals to control the periodical motion of animals. We propose a motion control system that enables a snake-like robot to adjust the phase lag of target joint trajectories according to changing environments. Simulations and an experiment with a real robot confirmed the effectiveness.

***Keywords*—**Snake-like robot, central pattern generator, neural oscillator, biomimetic robot, adaptive control system.

I. Introduction

As the performance of processors increases, various new applications of mobile robots have been proposed and evaluated for use in extreme environments, such as observations and operations at the bottom of oceans, in space, or in nuclear plants. One of the important issues to be solved is the robot's adaptability to such environments. In nature, it is known that animals adapt to environmental changes. For example, snake-like animals are able to move across flat land and slopes, and under water. Their motion mechanisms have been studied by Hirose [1] and Azuma [2]. A snake's body is covered by a special scale that slips in the tangential direction and has high friction in the normal direction. Snakes are able to convert the reaction force in the normal direction from the ground to an impellent force in the tangential direction by wiggling its body with a certain phase lag between joint trajectories. Similarly, eels and sea snakes generate an impellent force by using hydrodynamic force underwater.

Recently, snake-like robots that imitate the properties of snake-like animals have been developed and are expected to be able to travel through various environments [3][4]. Snake-like robots generate an impellent force by passive wheels, which realize the properties of a real snake's scales. Real snake-like animals change their behaviors depending on their working environments because their motions are affected by different forces such as frictional force, hydrodynamic force, and so on.

To realize a robust and adaptive robot system, an adaptive control system that focuses on the nervous systems of animals is introduced in this study. In the nervous system of animals, it has been proved that motion patterns such as walking, flapping, etc., are controlled by rhythm generator mechanisms called central pattern generators (CPGs) [5]. The CPG consists of many neural oscillators, and the effects of each neural oscillator on another generate the rhythm patterns. Additionally, a CPG and a neural oscillator have an entrainment feature that synchronizes the external force conveyed from the sensory system of animals, so that animals are able to adapt to environmental changes. CPGs and neural oscillators are attractive and have already been applied to adaptive control systems in several previous studies. In this paper, we propose an adaptive control system, which we applied to a snake-like robot by using the entrainment feature of a neural oscillator. We analyzed the ability of the snake-like robot to adapt to changing environments and dynamics.

In previous works, methods that realize adaptation mechanisms and algorithms for changing environments by using the entrainment feature of a neural oscillator have been proposed. Williamson [6] developed a robot arm control system using neural oscillators that carried out several motions, such as throwing and drumming. Matsuoka et al. [7] proposed a giant swing robot using neural oscillators for a small swing and a giant swing with different dynamics. In the approach of both studies, the connections between neural oscillators were not coupled directly, but instead had dynamical coupling based on the entrainment feature of a neural oscillator. Inoue et al. proposed an adaptive control system that was able to adapt to changing friction by using Matsuoka's model [8]. Watanabe et al. realized an adaptive control system using phase oscillators [9]. However, the previous works of Matsuoka and Watanabe presumed a direct connection between oscillators, unlike the use of indirect connections between oscillators as proposed by Williamson and Matsuoka. In this paper, a CPG and neural oscillators are applied to the adaptive control system. As a result, we propose an adaptive control system that is coupled indirectly between neural oscillators by using the dynamical interactions applied to a snake-like mobile robot. We introduce an amphibious snake-like robot AMMR (Amphibious Multi-link Mobile Robot) and describe the construction method

Corresponding author: Takayuki Matsuo (e-mail: matsuo@kct.ac.jp).
This paper was submitted on February 28, 2014; revised on April 09, 2014; and accepted on May 14, 2014.

of the kinetic model in section II. In section III, we propose the phase lag adjustment system using neural oscillators and analyze the capability of the proposed system through simulations. We conduct an experiment and present the results in section IV, followed by a discussion in section V. Finally, we give our conclusions and state the direction of our future work in section VI.

II. Dynamic Model of a Snake-Like Robot

AMMR, which was developed in our previous work [10], is introduced in this section; and a kinetic model of the robot for the motions on ground is constructed by using the center of gravity (COG) Jacobian. Among the motions on ground, the motion property of the robot on a slope is analyzed.

A. AMMR

The snake-like robot AMMR is shown in **Figure 1** and its specifications are listed in **TABLE I**. AMMR, which has passive wheels and fins, can generate an impellent force by wiggling its body like a snake. In this paper, we particularly discuss the motion properties on the ground and its adaptive capacity. AMMR has seven joints. Each joint has 2 DOF in the yaw and pitch directions. A joint consists of an active joint controlled by a motor in the yaw direction and a passive joint that enables movement in the pitch direction. The passive joint consists of a free joint and two springs, as shown in **Figure 2**.

Figure 1 Overview of AMMR. The robot can be operated on ground and under water. In this paper, we considered the motion on ground

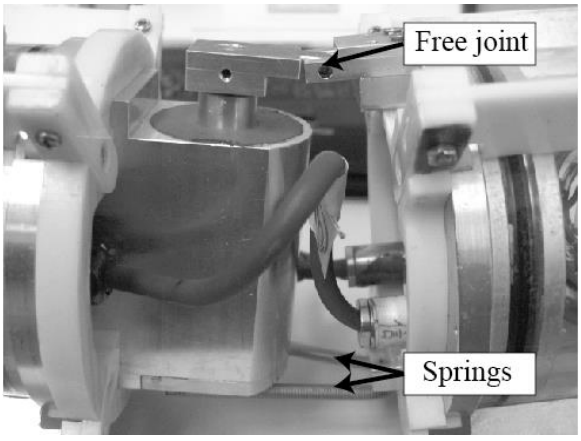


Figure 2 Passive joint mechanism in the pitch direction. The mechanism consists of a free joint and two springs

TABLE I Specifications of AMMR

Length	2.2 m (0.275 m/link)
Weight	12.8 kg (1.6 kg/link)
Number of joint	7
Number of link	8
MPU	PIC32MX795F512H x8
Communication	CAN

A. Kinematics of the Snake-like Robot

We previously proposed a construction method of a kinetic model employing COG Jacobian for the snake-like robot [11]. The method derives matrices that make a correlation between the COG of each link and the kinetic pairs of links, and calculates the inverse dynamics by using the principle of virtual work. In this method, the equation of motion is derived efficiently because COG Jacobian matrices can be derived by using velocities or accelerations. Moreover, the method is able to represent equations of inverse dynamics as equations of matrices.

The orientation vectors of the COG of the robot and each link i are denoted by $\boldsymbol{q}_c = [x_c \quad y_c]$ and $\boldsymbol{q}_c = [x_i \quad y_i \quad \theta_i^a]$, where x_c and y_c are the coordinates of COG of the robot, x_i and y_i are the coordinates of the COG of link i (i=1,…,8) and θ_i^a is the absolute angle of link i. In addition, the orientation vector regarding all links $\boldsymbol{q}$ is given by

$$\boldsymbol{q} = \boldsymbol{g}(\boldsymbol{\theta}_m), \tag{1}$$

where $\boldsymbol{g}(\boldsymbol{\theta}_m)$ is the function of the vector of kinematic pairs $\boldsymbol{\theta}_m$ expressed by

$$\boldsymbol{\theta}_m = [x_c \quad y_c \quad \theta_5^a \quad \theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7] \tag{2}$$

The acceleration of the COG of the links is written in the following form:

$$\boldsymbol{q} = \boldsymbol{G}(\boldsymbol{\theta}_m)\ddot{\boldsymbol{\theta}}_m + \dot{\boldsymbol{G}}(\boldsymbol{\theta}_m)\dot{\boldsymbol{\theta}}_m. \tag{3}$$

where $\boldsymbol{G}(\boldsymbol{\theta}_m)$ is the COG Jacobian of links which is written the flowing form:

$$\boldsymbol{G}(\boldsymbol{\theta}_m) = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}_m} \tag{4}$$

The external force vector $\boldsymbol{F}$ around the COG of a links is expressed as following equation:

$$\boldsymbol{F} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b} + \boldsymbol{f} \tag{5}$$

where $\boldsymbol{M}$ is the mass matrix of links, $\boldsymbol{b}$ is the term regarding the gravitational force, etc., and $\boldsymbol{f}$ is the drag force, such as frictional force.

Next, the torque vector $\boldsymbol{\tau}_m$ of the kinematic pairs regarding link *i* is defined in Equation 6.

$$\boldsymbol{\tau}_m = [0 \quad 0 \quad 0 \quad \tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4 \quad \tau_5 \quad \tau_6 \quad \tau_7] \qquad (6)$$

$\boldsymbol{\tau}_m$ is obtained by the transposed matrix of $\boldsymbol{G}$ and $\boldsymbol{F}$. Thus, $\boldsymbol{\tau}_m$ is calculated by

$$\boldsymbol{\tau}_m = \boldsymbol{G}^T \boldsymbol{F}. \qquad (7)$$

Here, $\boldsymbol{\tau}_m$ is derived using Equation 3, 5 and 7:

$$\boldsymbol{\tau}_m = \boldsymbol{G}^T \boldsymbol{M} \boldsymbol{G} \ddot{\boldsymbol{\theta}}_m + \boldsymbol{G}^T \boldsymbol{M} \dot{\boldsymbol{G}} \dot{\boldsymbol{\theta}}_m + \boldsymbol{G}^T \boldsymbol{b} + \boldsymbol{G}^T \boldsymbol{f}. \qquad (8)$$

The drag force $\boldsymbol{f}$ on the flat area and the slope is defined as $\boldsymbol{f}^{flat}$ and $\boldsymbol{f}^{slope}$. On the flat area, the drag force $\boldsymbol{f}_i^{flat}$ is denoted by

$$\boldsymbol{f}_i^{flat} = -\frac{\dot{\boldsymbol{q}}_i}{|\dot{\boldsymbol{q}}_i|} \boldsymbol{\mu}' \boldsymbol{M}_i g \qquad (9)$$

In Equation 9, $\mu' = [\mu_t' \quad \mu_n' \quad 0]$ is dynamic friction coefficient, and g is gravity acceleration. The friction coefficients are $\mu_t' = 0.01$ and $\mu_n' = 0.01$ throughout this paper. On the slope area, the drag force $\boldsymbol{f}_i^{slope}$ at slope angle ψ ° is shown in Equation 10. Also, $\boldsymbol{f}_i^{flat}$ and $\boldsymbol{f}_i^{slope}$ are combined as $\boldsymbol{f}^{flat}$ and $\boldsymbol{f}^{slope}$.

$$\boldsymbol{f}_i^{slope} = -\frac{\dot{\boldsymbol{q}}_i}{|\dot{\boldsymbol{q}}_i|} \boldsymbol{\mu}' \boldsymbol{M}_i g \cos\psi - \boldsymbol{M}_i g \sin\psi \begin{bmatrix} \sin\theta_i^a \\ \cos\theta_i^a \\ 0 \end{bmatrix} \qquad (10)$$

B. Simulations of Climbing changing phase lag

Here, we consider how to set the changing environment. In another study of an adaptive control system for a snake-like robot, changing slope angles or frictional force were adopted as the changing environment, and the climbing ability of a snake-like robot was discussed [12]. The authors in that study analyzed the adaptation ability by adjusting the amplitude of the joint angle of the robot. We adopted a changing slope angle as the changing environment and adjusted the phase lag between target joint trajectories of the robot.

We assumed that each target joint trajectory $\hat{\theta}_k$ of a robot can be approximated by the sinusoidal function as shown in Equation 11, and the robot can be controlled by three parameters viz. amplitude A°, frequency f Hz of each joint trajectory, and phase lag φ between the trajectories of adjacent joints. The snake-like robot is able to move on the ground by driving each joint with a certain phase lag.

$$\hat{\theta}_k = \begin{cases} A\sin 2\pi f t & (k=1) \\ A\sin\left(s\pi f t + \dfrac{\pi}{180}\varphi\right) & (k \geq 2) \end{cases} \qquad (11)$$

Figure 3 shows the relationships between the travel velocity v and phase lag φ when slope angle ψ was changed from 0° to 3.5° with a step of 0.5°. In simulations, amplitude A° and frequency f were fixed $A = 30$° and $f = 0.3Hz$ fixed. Phase lag φ was changed from 40 to 100° with a step of 5°. We evaluated the travel velocity as the climbing ability. The data of the travel velocity was not plotted in the case that the robot was not able to climb the slope.

Intriguing relationships between φ and v when changing the slope angle ψ appeared in the simulation results. When was 40°, v was the fastest in $\varphi = 0$°. However, when the slope angle becomes steeper, the travel velocity became slow and the robot was not able to climb as much. When other phase lags were used, the same tendency appeared. The climbing ability becomes worse when the slope angle was increased for the same phase differences. However, the climbing ability was improved by extending the phase lag when the slope angle becomes steeper. For example, when phase lag φ was 40°, the robot was able to climb the sloped area from 0.5 to 1.0°, but the robot was not able to climb the sloped area over 1.5°. However, setting the phase lag from 45 to 100° enabled the robot to go up the slope.

These results suggest that the climbing ability of AMMR can be improved by increasing phase lag φ between the target joint trajectories of the robot. In this paper, we propose an adjustment system that changes phase lag φ between the target joint trajectories according to the variation of slope angle ψ.

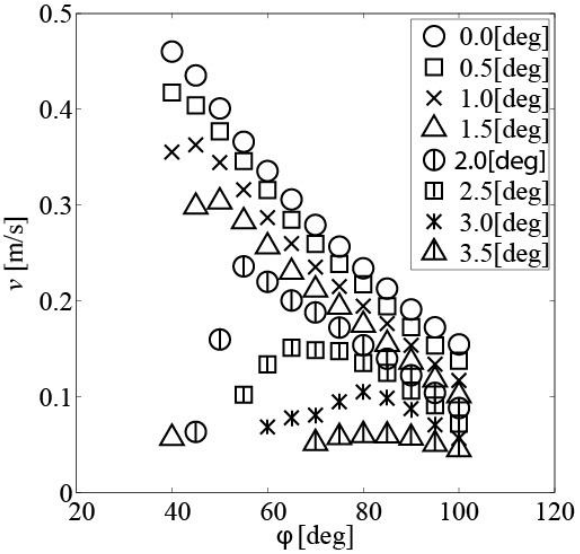


Figure 3 The relationships between velocities *v* and phase lag φ on the sloped area. The robot is not able to climb the slope at the same phase lag when the slope angle becomes steeper. However, the robot is able to climb the slope when phase lag is increased

III. Phase Lag Adjustment System Using Neural Oscillators

The oscillators used in this study are introduced in this section. We first present the governing equations of the oscillators and how they are coupled with each other for constructing the phase lag adjustment system. Then, we propose the phase lag adjustment system using neural oscillators for a snake-like robot, and the capability of phase lag adjustment by the kinetic model constructed in section II is analyzed in simulations.

A. Proposed Method

The neural oscillator employed in this paper was the Matsuoka model [13][14]. It consists of two neurons that represent the extensor neuron and the flexor neuron in mutual inhibition as shown in **Figure 4**. The black and white circles of the neurons correspond to inhibitory and excitatory connections. The mutual inhibition is weighted by w_{ef} and w_{fe}. The neural oscillator has four state variables u^e, u^f, v^e and v^e, governed by the following differential equations.

$$\tau_u \dot{u}^e = -u^e - \beta v^e + w_{ef} y^f + u_0 - h g^e \tag{12}$$

$$\tau_v \dot{v}^e = -v^e + y^e \tag{13}$$

$$y^e = \max(u^e, 0) \tag{14}$$

$$g^e = \max(g, 0) \tag{15}$$

$$\tau_u \dot{u}^f = -u^f - \beta v^f + w_{fe} y^e + u_0 - h g^f \tag{16}$$

$$\tau_v \dot{v}^f = -v^f + y^f \tag{17}$$

$$y^f = \max(u^f, 0) \tag{18}$$

$$g^f = \max(-g, 0) \tag{19}$$

$$y^{out} = y^e - y^f \tag{20}$$

Here, superscript *e* and *f* denote the extensor neuron and the flexor neuron, respectively. Variables u^e and u^f represent the firing rate of the neurons, and variables v^e and v^f are internal states representing the self-inhibition whose strength is determined by the parameter β. The outputs of neuron y^e and y^f are taken as the positive parts of the firing rate by using the piecewise linear function. Tonic input u_0 determines the amplitude of oscillator output y^{out}. Time constants τ_u and τ_v determine the frequency and shape of the output. The feedback signal from the sensory system is applied to the oscillator trough variable *g*, weighted by gain *h*. The feedback signal inhibits the neuron by applying the positive part g^e of variable *g* to the extensor neuron, and the negative part g^f of variable *g* to the flexor neuron. Taga et al. [15][16] applied the outputs of the neural oscillators to the torque control of bipedal locomotion in unpredictable environment. In this study, the output of the oscillator is applied to the target joint angles of the snake-like robot, and the adjustment system of phase lag for control of the snake-like robot is proposed.

Figure 5 illustrates the proposed system for adjusting the phase lag. The proposed system is expressed by Equations 21-29, and target joint trajectories $\hat{\theta}_k$ (k=1,…,7) are generated by a sinusoidal function or the outputs of neural oscillators NO_n (*n*=1,2,…,6) weighted by gain *H* as shown in Equation 30. The adjustment system consists of the sinusoidal function and seven neural oscillators. The joint trajectory θ_1 is controlled by a PD controller using the target joint trajectory $\hat{\theta}_1$ generated by the sinusoidal function, as shown in Equation 31. In addition, θ_1 is used by neural oscillator NO_1 as a feedback signal. The target joint trajectory of the second joint $\hat{\theta}_2$ is generated by an output signal of NO_1, and θ_2 is also controlled by the PD controller using the target joint trajectory $\hat{\theta}_2$. The target trajectories from $\hat{\theta}_3$ to $\hat{\theta}_7$ are generated in the same method used for $\hat{\theta}_2$.

$$\tau_u \dot{u}_n^e = -u_n^e - \beta v_n^e + w_{ef} y_n^e + u_0 - h g_{k-1}^e \tag{21}$$

$$\tau_v \dot{v}_n^e = -v_n^e + y_n^e \tag{22}$$

$$y_n^e = \max(u_n^e, 0) \tag{23}$$

$$\theta_{k-1}^e = \max(\theta_{k-1}, 0) \tag{24}$$

$$\tau_u \dot{u}_n^f = -u_n^f - \beta v_n^f + w_{fe} y_n^f + u_0 - h \theta_{k-1}^f \tag{25}$$

$$\tau_v \dot{v}_n^f = -v_n^f + y_n^f \tag{26}$$

$$y_n^f = \max(u_n^f, 0) \tag{27}$$

$$\theta_{k-1}^f = \max(-\theta_{k-1}, 0) \tag{28}$$

$$y_n^{out} = y_n^e - y_n^f \tag{29}$$

$$\hat{\theta}_k = \begin{cases} A \sin 2\pi f t & (k = 1) \\ H y_n^{out} & (k \geq 2) \end{cases} \tag{30}$$

$$\tau_k = K_p(\hat{\theta}_k - \theta_k) + K_d \frac{d(\hat{\theta}_k - \theta_k)}{dt} \tag{31}$$

A feature of our proposed system is that the system has no direct connections in the sense that the network has no direct nerve connections, such as the inhibitory (or excitatory) connections that a neural oscillator has between neurons.

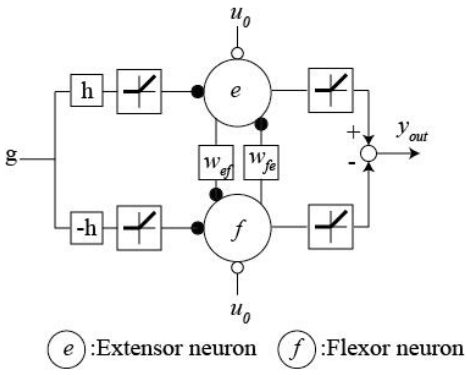


Figure 4 The oscillator architecture.

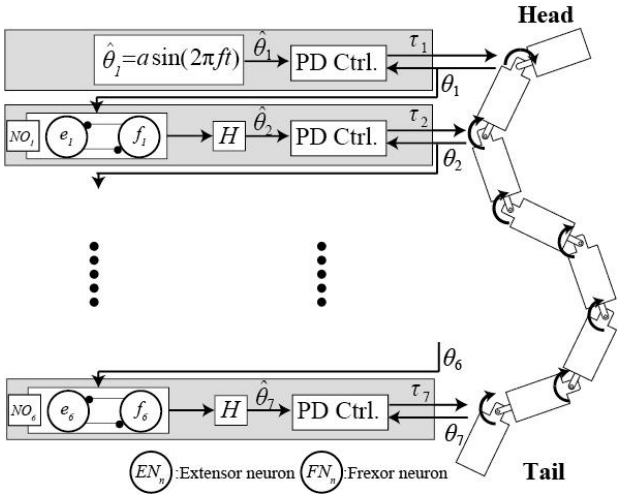


Figure 5 Architecture of proposed system. The network between neural oscillators has indirectly connections

B. *Adjustment Mechanism*

In general, the control abilities of the actuators deteriorate in an environment where the robot is influenced by high drag force (e.g. a high friction area, a sloped area), and the deterioration of the control abilities appears as the phase delay and the amplitude damping of the joint trajectories in comparison with the target joint trajectories, as shown in **Figure 6 (a)**. In our proposed system, we focused on the phase delay of joint trajectory θ_j to target joint trajectory $\hat{\theta}_j$. Joint trajectory θ_j which has the phase delay to target joint trajectory $\hat{\theta}_j$, is conveyed into a neural oscillator as a feedback signal, and the oscillator outputs a synchronized signal as the target joint trajectory $\hat{\theta}_{j+1}$, as shown in **Figure 6 (b)**. As the result, the phase lags between target joint trajectories $\hat{\theta}_j$ and $\hat{\theta}_{j+1}$ increases. The proposed system is able to create the phase lag by using the deterioration of control abilities that occurs due to the changing environment.

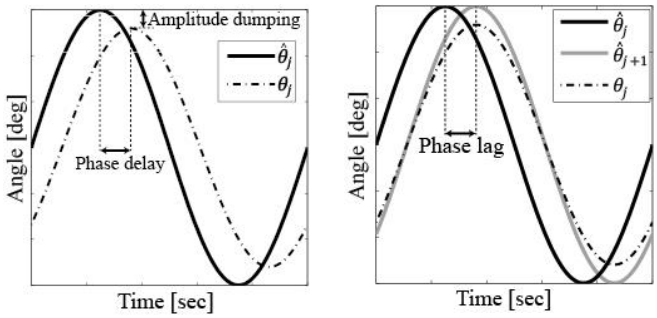


Figure 6 The adjustment mechanism of our proposed system. (a) The relationship between target joint trajectory $\hat{\theta}_j$ and joint trajectory θ_j when the robot is influenced by a large drag. Amplitude dumping and phase delay appear. (b) The relationship between target joint trajectories $\hat{\theta}_j$, $\hat{\theta}_{j+1}$ and joint angle θ_j.

C. *Simulations of Adjustment capability*

To test our proposed system and allow easier adjustments of the environments, a simulated model of the robot was created by using the kinetic model constructed in Section II. We carried out an analysis of our proposed system with regard to the ability of phase lag adjustment.

Figure 7 is the snapshots which shows a simulation result from operation of the robot on flat and slope area (slope angle $\psi = 3.5^{o}$) for 20.0 s. The time step between frames is 2.0 s. The parameters of the sinusoidal function and the neural oscillators are set as shown in **TABLE II** (parameter set 1). The robot was able to climb the slope.

We compared the robot behavior when implementing the proposed system with the robot behavior when not implementing the system. For creating the target joint trajectories that are not generated by the proposed system, the time-series data of the target joint trajectories for the robot going on a flat area for 20 s is saved and the data are loaded for the simulations of operation on the flat and the sloped areas. The snapshots in **Figure 8** shows the simulation results from operation of the robot on the flat and the sloped areas (slope angle $\psi = 3.5^{o}$) for 20.0 s without the proposed system. The time step between frames is the same as that (2.0 s) in **Figure 7**. In the case that the proposed system is not implemented in the robot control system, the robot was not able to climb the slope and slipped backward. One plausible explanation for the difference is the adjustment property of the proposed system: the proposed system enables a change of the phase lag between target joint trajectories, as shown in **Figure 9** and **TABLE III**. **Figure 9** shows the variation of the waves between the flat area and the sloped area, and **TABLE III** shows the values of phase lag φ_{j_j+1} between target joint trajectories $\hat{\theta}_j$ and $\hat{\theta}_{j+1}$. In this paper, we obtained the phase lag φ_{j_j+1} between target joint trajectories as

$$\varphi_{i\_j+1} = 360\, f\,(t_{j-1}^{p} - t_{j}^{p}) \tag{32}$$

where t_j^p is the time when the amplitude of the target joint

trajectory is a maximum value, as shown in **Figure 10**. The changing phase lag appeared between operations on the flat and the sloped areas. In addition, parameter set 2 shown in **TABLE IV** was tested; the resulting motion pattern is shown in **Figure 11**, and the changing phase lags are shown in **Figure 12** and **TABLE V** when using parameter set 2. Then, the robot motion not implementing the proposed system in the robot control system was analyzed. As shown in **Figure 13**, the robot was able to climb the slope, unlike the result for parameter set 1, because the phase lags when using parameter set 2 were larger than the phase lags when using parameter set 1. However, when the variation of the total torque of the motors was analyzed, the following result appeared.

The variation of the total torque generated by all motors that drive each joint of the robot was analyzed. **Figure 14 (a)** and **(b)** illustrate variations of the total torque generated by the motors. The dotted line shows the total torque when not implementing the proposed system and the solid line shows the total torque when implementing the proposed system. When using parameter set 1, shown in **Figure 14 (a)**, the amplitude of the total torque for the robot not implementing the proposed system diverges. However, almost no changes in the amplitude appear for the robot implementing the proposed system. When using parameter set 2, shown in **Figure 14 (b),** almost no changes appear in the amplitude of the total torque for the robot implementing the proposed system, and the amplitude is the same as that when using parameter set 1.

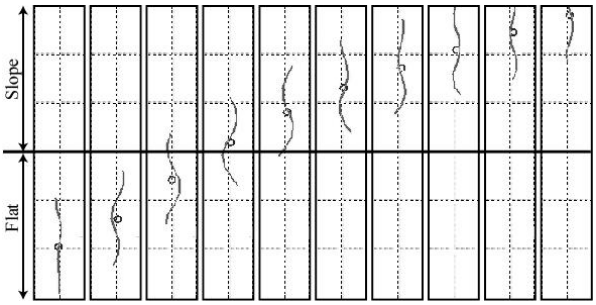


Figure 7 Behavior of robot on flat and slope ($\psi = 3.5^{\circ}$) areas for the robot implementing the proposed system with parameter set 1 (Simulation)

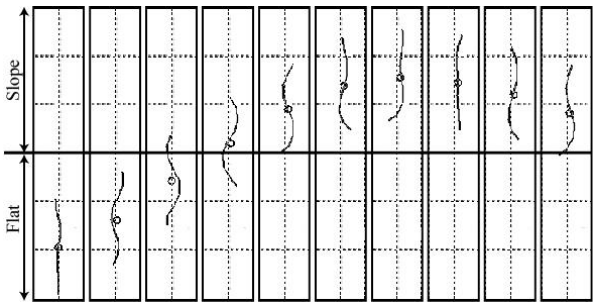


Figure 8 Behavior of robot on flat and slope ($\psi = 3.5^{\circ}$) areas for the robot not implementing the proposed system.

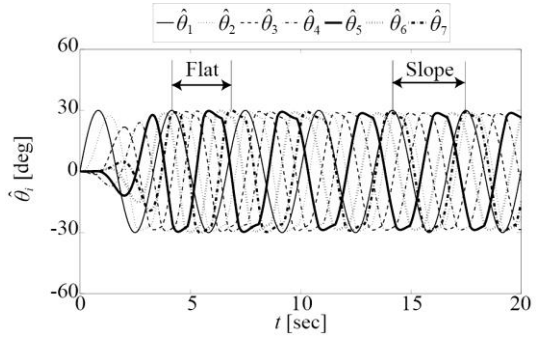


Figure 9 Phase lags between target joint angle on flat and sloped areas and implementing the proposed system with parameter set 1 (Simulation).

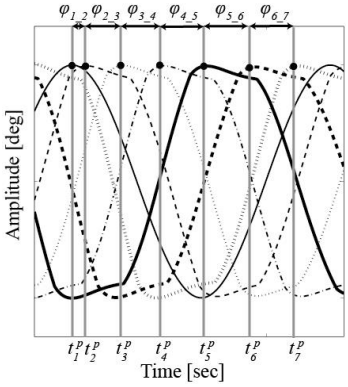


Figure 10 The calculation method of phase lag. The black circles are the points where the amplitude is the maximum value.

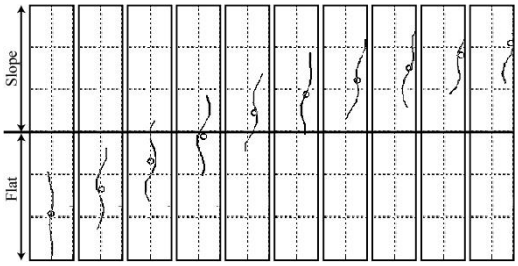


Figure 11 Behavior of robot on flat and slope ($\psi = 3.5^{\circ}$) areas for the robot implementing the proposed system with parameter set 2 (Simulation).

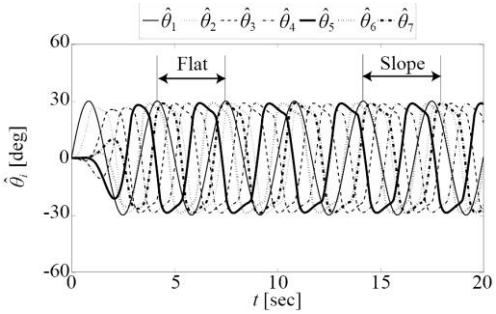


Figure 12 Phase lags between target joint angle on flat and sloped areas for the robot implementing the proposed system with parameter set 2 (Simulation).

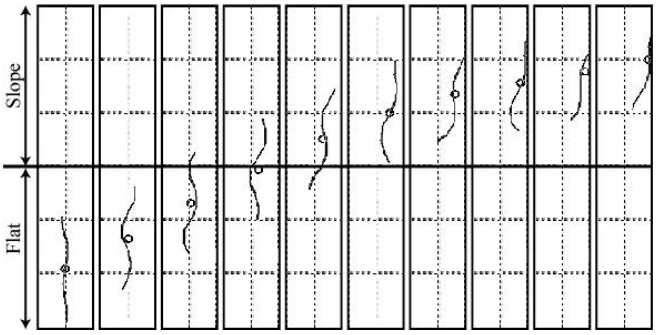
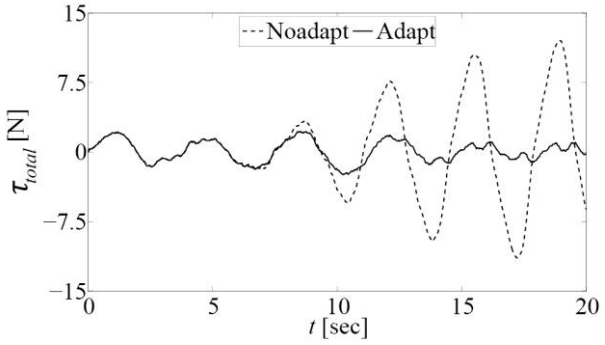
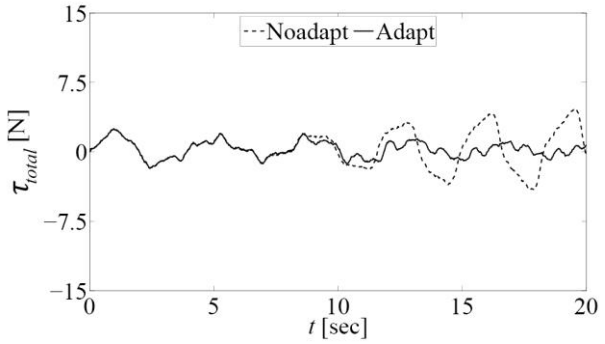


Figure 13 Behavior of robot on flat and slope ($\psi = 3.5^{o}$) areas for the robot not implementing the proposed system.



(a)



(b)

Figure 14 Variation of total torque of all motors. (a) Variation of total torque with parameter set 1. (b) The variation of total torque with parameter set 2. The solid line shows the variation of total torque for the robot implementing the proposed phase lag adjustment system, and the dotted line shows the variation of total torque for the robot not implementing the proposed system.

TABLE II Setting parameters of neural oscillators and the sinusoidal function (parameter set 1)

τ_u	0.3	τ_v	0.36	w_{fe}	1.5	w_{ef}	1.5	u_0	126
β	1	h	-1	H	0.41	A	30	f	0.3

TABLE III Values of each phase lag φ_{j_j+1} º on flat and sloped areas when using parameter set 1

	φ_{1_2}	φ_{2_3}	φ_{3_4}	φ_{4_5}	φ_{5_6}	φ_{6_7}
Flat area $\psi = 0^{o}$	18.4	42.1	46.4	47.5	46.4	46.4
Sloped area $\psi = 3.5^{o}$	22.7	62.7	69.1	67.0	69.1	70.2

TABLE IV Setting parameters of neural oscillators and the sinusoidal function (parameter set 2)

τ_u	0.3	τ_u	0.6	w_{fe}	1.5	w_{ef}	1.5	u_0	100
β	1	h	-1	H	0.41	A	30	f	0.3

TABLE V Values of each phase lag φ_{j_j+1} º on flat and sloped areas when using parameter set 2

	φ_{1_2}	φ_{2_3}	φ_{3_4}	φ_{4_5}	φ_{5_6}	φ_{6_7}
Flat area $\psi = 0^{o}$	32.4	57.2	58.3	60.5	61.6	59.4
Sloped area $\psi = 3.5^{o}$	32.4	71.3	77.8	75.6	81.0	58.3

IV. Experimental Result Using a Real Robot

Finally, an experiment using a real robot (AMMR) was carried out. We confirmed that our proposed system is able to generate the property of phase lag adjustment, as explained in Section III. The setup of the hardware and experimental environment are explained next along with the results.

A. Set up of Hardware and Experimental Environment

AMMR consists of two kinds of modules: a module for the head (head module) and seven modules for the body (motor modules). AMMR implements a distributed control system. A motor module has a motor and a circuit board to control the joint angle. The circuit has a microprocessor unit (MPU, PIC32MX795F512H), a potentiometer to measure the joint angle, and a CAN transceiver (MAX3051ESA) for communication. A block diagram of the circuit is shown in **Figure 15**. The MPU calculates the target trajectory by using the neuron potential of a neural oscillator, manages the angle data from the sensory system, which is able to use a potentiometer for the feedback signal, controls the motor by PD control, and communicates with the circuits of other modules by using the CAN. A motor module sends the angle data of the joint under its own control and receives the angle data from another module. The head module is the interface device between the robot and the host PC, and it transfers the target behavior to the motor modules.

In the experiment, we constructed the experimental environment shown in **Figure 16.** The experimental environment has two areas: a flat area and a sloped area. Each area consists of a wood board and aluminum frames, and the slope angle can be adjusted.

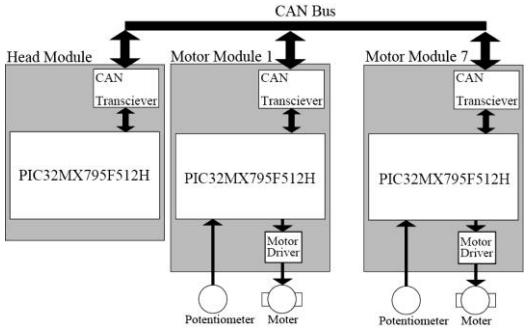


Figure 15 System architecture of the distributed control system employed for robot control.

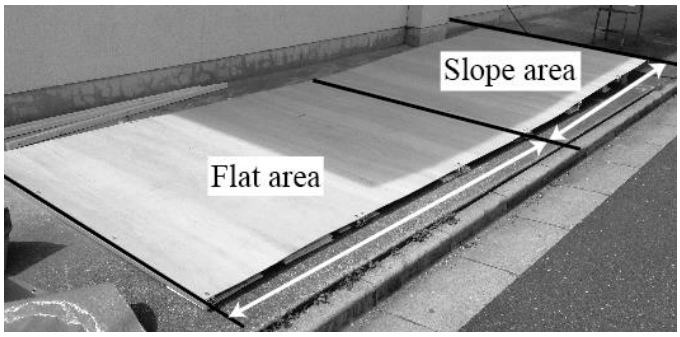


Figure 16 Experimental environment.

B. Experimental result

An experiment was carried out with the real snake robot AMMR using parameter set 1 shown in **TABLE II**. As the experimental result, the robot was not able to climb a slope at 3.5°. Therefore, the slope angle was changed to 3.0°, and the robot was able to climb the slope by changing the phase lag. Target joint trajectories were generated by the proposed system on the flat and the sloped areas, as shown in **Figure 17**. **TABLE VI** lists the values of the phase lags between the target joint trajectories. The change of each phase lag between the target joint angles appears as shown in the simulation results.

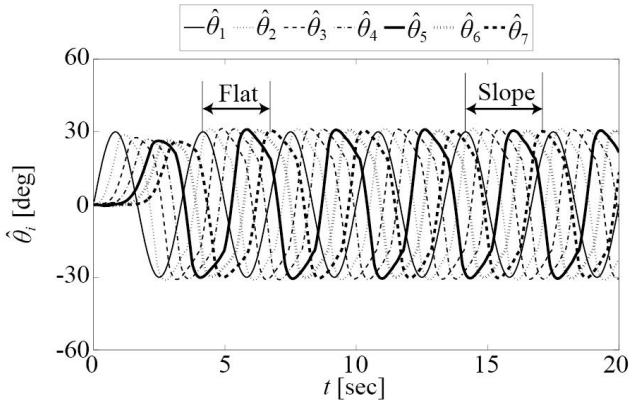


Figure 17 Phase lags between target joint angles on flat and sloped areas when using the proposed system with parameter set 1 (real robot). The slope angle was set at 3.0°.

TABLE VI Values of each phase lag on flat and sloped areas

	φ_{1_2}	φ_{2_3}	φ_{3_4}	φ_{4_5}	φ_{5_6}	φ_{6_7}
Flat area $\psi = 0°$	28.1	49.7	51.8	51.8	45.4	51.8
Sloped area $\psi = 3.5°$	30.2	51.8	56.2	56.2	60.5	58.3

V. DISCUSSIONS

The goal of this article was to propose a phase lag adjustment system using the Matsuoka model. The Matsuoka model has more advantages than other oscillators. First, the architecture of the oscillator using the Matsuoka model is simpler than that of other oscillators. Second, even if the feedback signal is very small, the output of the oscillator is able to be generated [17]. As has been mentioned in section I, previous works applied the oscillator using the Matsuoka model for the control of a snake-like robot. However, it is difficult to change the phase lag by a direct connection, because a phase lock occurs between the oscillators by the entrainment feature. Therefore, even if the adjustment system has a direct connection, certain strategies are needed. The method using the direct connection was shown to be difficult to analyze and tune due to the complex nonlinear interactions between oscillators, as shown in the actuated system in [6]. In this paper, we simplified the system by using an indirect connection between the oscillators; and the phase lags between target joint trajectories were able to be generated depending on the phase delay of the joint trajectory. No previous works have used the Matsuoka model specifically for phase lag control of a snake-like robot.

VI. CONCLUSIONS

In this paper, an adjustment method of the phase lag for controlling a snake-like robot was proposed. The adjustment system consists of a neural oscillator network have an indirect connection between oscillators. The effectiveness of the proposed system was analyzed by using a kinetic model of a real robot (AMMR). The model was constructed by employing COG Jacobian. We tested two parameter sets for simulations. Both parameter sets were able to generate and change the wave patterns, which were different on the flat surface and the inclined surface in the simulations. Results of the simulations suggested that our proposed adjustment system was able to adjust the phase lag between target joint trajectories according to the variation of the slope angle. Thus, the total torque of the motors did not change much on the flat and the inclined surface when the proposed system was implemented. In addition, we used a real robot for evaluating variations of the phase lag. As the result, the function of the adjustment of the phase lag was confirmed.

In future work, we will develop a robot with more links than AMMR. AMMR has eight links. However, the number of links is important to analyze the effectiveness of the changing phase lag because the number of links influences the shape of the robot. Therefore, a new robot with twenty links and current sensors for measuring torque is under development.

REFERENCES

[1] Shigeo Hirose, "Bionic machine engineering", Kougyo chosakai, 1987 (in Japanese)

[2] A. Azuma, The subject book of Animal's Motion", Asakura shoten, 1997 (in japanese)

[3] Makoto Mori, Hiroya Yamada and Shigeo Hirose, "Design and Development of Active Cord Mechanism "ACM-R3" and its 3-dimensional Locomotion Control", *Journal of Robotics Society of Japan Vol.23 No.7*, pp. 886-897,2005. CrossRef

[4] Taro Ohashi, Hiroya Yamada and Shigeo Hirose, "Study of Active Cord Mechanism ACM-R7 Capable of Loop-Gait Locomotion", *Journal of Robotics Society of Japan Vol.28 No.7*, pp. 880-888, 2010. CrossRef

[5] S. Grillner, "Neural Networks for Vertebrate Locomotion", Scientific American Japanese Edition, March 1996, pp. 76-83

[6] M. M. Williamson, "Robot Arm Control Exploting Natural Dynamics", Ph. D thesis, Masachusetts Institute of Technology, 1999

[7] K. Matsuoka, N. Ohyama, A. Watanabe and M. Ooshima, "Control of a giant swing robot using a neural oscillator", Advance in Neural Computation (Proc. ICNC 2005, Part II), pp. 274-282, 2005

[8] K. Inoue, T. Sumi and S. Ma, "CPG-based Control of a Simulated Snake-like Robot Adaptable to Changing Ground Friction", Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1957-1962, 2007 CrossRef

[9] W. Watanabe, T. Sato and A. Ishiguro "Autonomous Decentralized Control of Serpentine Robot and Environment", ROBOMEC 2009, 4pages, 2009

[10] T. Matsuo, T. Yokoyama, D. Ueno and K. Ishii, "Development of an Amphibious Multi Link Mobile Robot and Motion Control Using CPG", Proc. of RSJ2008, 4pages.

[11] T. Matsuo, T. Sonoda, T. Yokoyama, D. Ueno and K. Ishii, "Construction of a Kinetic Model Employing COG Jacobian for an Amphibious Snake-like Robot, Proc. of World Automation Congress (WAC2010), 6pages (CD-ROM).

[12] G. Endo, K. Togawa and S. Hirose, "Study on Self-Contained and Terrain Adaptive Active Cord Mechanism", *Journal of Robotics Society of Japan Vol.18 No.3*, pp. 419-425, 2000. CrossRef

[13] K. Matsuoka, "Sustained oscillators generated by mutually inhibiting neurons with adaptation", Biological Cybernetics 52, pp. 367-376, 1985. CrossRef

[14] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators", Biological Cybernetics 56, pp. 345-353, 1987. CrossRef

[15] G. Taga, Y. Yamaguchi and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment", Biological Cybernetics Vol.65, pp. 147-159, 1991. CrossRef

[16] G. Taga, "A model of the neuro-musculo-skeletal system for human locomotion II. Real-time adaptability under various constraints", Biological Cybernetics Vol.73, pp.113-121, 1998. CrossRef

[17] Daniel P. Ferris, Tiffany L. Viant and Ryan J. Campbell, "Artificial Neural Oscillators as Controllers for Locomotion Simulations and Robotic Exoskeletons", World Congress of Biomechanics, 2002.