Proceedings of the 2nd International Symposium
on Adaptive Motion of Animals and Machines,
Kyoto, March.4-8, 2003

Reinforcement Learning for Biped Robot

Yutaka Nakamura[1], Masa-aki Sato[2,3] and Shin Ishii[1,3]

[1] Nara Institute of science and technology. 8916-5 Takayama-cho, Ikoma, Nara 630-0192. yutak-na@is.aist-nara.ac.jp

[2] ATR, Human Information Science Laboratories. 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288.

[3] CREST, JST.

Abstract

Animal rhythmic movements such as locomotion are considered to be controlled by neural circuits called central pattern generators (CPGs), which generate oscillatory signals. Motivated by such a biological mechanisms, rhythmic movements controlled by CPG has been studied. As an autonomous learning framework for the CPG controller, we propose an reinforcement learning method , which is called the CPG-actor-critic method. We apply this method to the reinforcement learning for the biped robot. The computer simulation shows that our method is able to train the CPG such that the biped robot walks stably. We also examine the characteristic of this CPG controller.

1. Introduction

Rhythmic movements are fundamental in animals' movements. For example, locomotion and swimming are crucial for animals to survive. These rhythmic movements are further characterized by their rapid adaptability to various environments, and the control mechanism has been extensively studied both in biological science and in engineering. Recent studies on the biped locomotion enabled humanoid robots to walk in real environments [1]. Despite these advancements, further studies are still needed because human locomotion is much more flexible and robust than that of present robots.

Neurobiological studies revealed that rhythmic motor patterns are controlled by neural oscillators referred to as central pattern generators (CPGs) [2]. It has been also suggested that sensory feedback plays an important role in stabilizing rhythmic movements by coordinating the physical system and the CPGs. Inspired by these findings, human-like biped walking was successfully simulated in [3] by using the CPG controller, whose weights were carefully tuned by hand. However, it is very difficult to determine the CPG parameter values for various robots and environments, since there is no design principle to determine the parameter values.

The main aim of this article is to study a reinforcement learning (RL) method for a CPG controller that generates stable rhythmic movements. RL methods have been successfully applied to various Markov decision problems (MDPs) with finite state and action spaces [4]. RL in continuous state and action spaces need efficient function approximators. Especially, RL for biped locomotion is very difficult, because the biped robot is a highly unstable dynamic system that has continuous state and action spaces with a high degree of freedom. Standard RL methods [4] such as temporal-difference (TD) learning, Q-learning and actor-critic methods are not suited for training the CPG controller, which is an instance of recurrent neural networks. In order to deal with the CPG controller, we propose a new RL method called the CPG-actor-critic method. We applied this method to the biped robot simulator used in [3]. A computer simulation showed that our RL method was able to train the CPG so that the biped robot stably walked in a saggital plane.

2. Central pattern generator

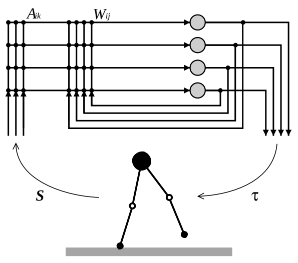


Figure 1: CPG drive system

The motion of a physical system such as a robot is expressed as

$$\dot{\mathbf{x}} = F(\mathbf{x}, \boldsymbol{\tau}), \tag{1}$$

where $\mathbf{x}$ and $\dot{\mathbf{x}}$ denote the physical state and its time derivative, respectively. $\boldsymbol{\tau}$ denotes the control signal (torque) from the controller. $F(\mathbf{x}, \boldsymbol{\tau})$ represents the vector field of the system dynamics.

In this article, we assume that a physical system is controlled by a CPG controller as depicted in Fig.1. The CPG controller is implemented as a neural oscillator network, and outputs control signals corresponding to neurons' state in the oscillator network. The CPG controller receives sensory feedback signals from the physical system.

A neural oscillator network is an instance of recurrent neural networks, and the dynamics of the i-th neuron is given by

$$\begin{aligned} c_i \dot{\nu}_i &= -\nu_i + I_i \\ y_i &= G_i(\nu_i), \end{aligned} \tag{2}$$

where ν_i, y_i, I_i and c_i denote the state, output,input, and time constant, respectively, of the i-th neuron. Output y_i is calculated from the state ν_i by the output function G_i. A sigmoidal function or a threshold function is used as the output function G_i.

Input I_i is given by

$$I_i = \sum_i W_{ij} y_j + I_i^{ext} + B_i, \tag{3}$$

where the first term is the feedback input from the other neurons, and the second term is the external input dependent on the sensory feedback signal from the physical system, and the third term is a bias. W_{ij} represents the weight of mutual connection from the j-th neuron to the i-th neuron. The external input I_i^{ext} is defined by

$$I_i^{ext} = \sum_k A_{ik} S_k, \tag{4}$$

namely, a sum of the sensory feedback signal $\mathbf{S}$ weighted by connection weight A_{ik}.

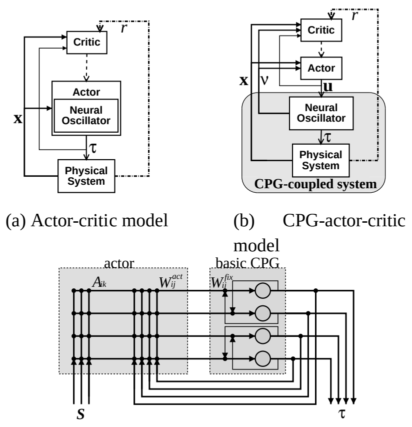
The control signal to the physical system is given by a weighted sum of the outputs of the CPG neurons:

$$\tau_n = \sum_i T_{ni} y_i, \tag{5}$$

where τ_n is the n-th control signal and T_{ni} represents the weight.

3. CPG-actor-critic model

An actor-critic model is a popular RL method. In that method, the actor is a controller that generates control signals to the physical system; it observes the physical system state and outputs control signals according to



(a) Actor-critic model (b) CPG-actor-critic model

(c) The actor and the basic CPG

Figure 2:

the system's state. The critic predicts the reward accumulation toward the future when the current actor is used as a controller. The parameters of the actor are updated so that the reward accumulation predicted by the critic becomes large.

When we try to apply the actor-critic model to the CPG controller, there occur several difficulties. In this method, the CPG controller is treated as an actor, as depicted in Fig.2(a). Since the output of the CPG controller depends on the state of the neural oscillator network, the reward accumulation also depends on the network state, that the critic does not observe. Therefore, the RL task becomes a partially observable problem, even if the physical system state is fully observable. A partially observable RL is much more difficult than a fully observable RL. In addition, the neural oscillator network in the CPG controller is a recurrent neural network. The usual gradient-based learning algorithm for the actor-critic model is not suited for training recurrent neural networks.

In order to overcome these difficulties, the CPG controller is divided into two modules, i.e., the basic CPG and the actor, as depicted in Fig.2(c). Corresponding to this modification, the input to the CPG neuron, I_i, is divided into two parts:

$$I_i = I_i^{fix} + u_i \tag{6}$$

$$I_i^{fix} = \sum_j W_{ij}^{fix} y_j + B_i \tag{7}$$

$$u_i = \sum_j W_{ij}^{act} y_j + \sum_k A_{ik} S_k, \tag{8}$$

where W_{ij}^{fix} is the connection weight in the basic CPG, and this weight value is fixed. The basic CPG neuron receives an indirect control signal $\mathbf{u}$ from the actor. The actor is a linear controller, which receives the output of the basic CPG, $\mathbf{y}$, and the sensory feedback signal $\mathbf{S}$, and outputs indirect control signal $\mathbf{u}$ to the basic CPG according to equation (8). The weight parameter W_{ij}^{act} and A_{ik} are adjustable, and are learned by the RL. The control torque to physical system is calculated by equation (5), and the weight parameter $\mathbf{T}$ is fixed. We call this RL method the CPG-actor-critic method.

The CPG-actor-critic method has two aspects. From a control viewpoint, the CPG controller consists of the basic CPG and the actor, as depicted in Fig.2(c). Namely, the CPG controller is a neural oscillator network whose connection weight is given by $W_{ij} = W_{ij}^{fix} + W_{ij}^{act}$. From the RL viewpoint, the actor outputs an indirect control signal $\mathbf{u}$ to a CPG-coupled system, which is composed of the basic CPG and the physical system, as depicted in Fig.2(b). In this view, the actor is a linear controller without any mutual feedback connection, and the parameter of the actor, W_{ij}^{act} and A_{ik}, can be determined by a gradient method describe below.

The critic observes the CPG-coupled system state $(\boldsymbol{\nu}, \mathbf{x})$, i.e., the basic CPG state $\boldsymbol{\nu}$ and the physical system state $\mathbf{x}$, and predicts the reward accumulation for the current state. If the physical system state is fully observable, RL task in this CPG-actor-critic method becomes a fully observable task.

4. Learning Algorithm

In this section, we explain an RL algorithm for the CPG-actor-critic method. To simplify the explanation, we assume that differential equations, (1) and (2), are discretized by an appropriate method.

The current state of the CPG-coupled system at time t consists of the physical system state $\mathbf{x}(t)$ and the basic CPG state $\boldsymbol{\nu}(t)$. The actor receives the basic CPG output $\mathbf{y}(t) = G(\boldsymbol{\nu}(t))$ and the sensory feedback signal $\mathbf{S}(t)$ which is a certain function of $\mathbf{x}$, and outputs an indirect control signal $\mathbf{u}$ calculated by equation (8). The CPG-coupled system receives the actor output $\mathbf{u}(t)$, and changes its state to $(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1))$ according to the dynamics of the basic CPG and the physical system , (1)-(7). After that, it is assumed that the critic receives an immediate reward $r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$.

4.1. Value function

The goal of an RL method is to obtain the actor that maximizes the expected reward accumulation toward the future, which is defined by

$$V(\boldsymbol{\nu}, \mathbf{x}) \equiv \sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)), \qquad (9)$$

where $\nu(0) = \nu$ and $\mathbf{x}(0) = \mathbf{x}$ are assumed. $\gamma \in (0, 1]$ is a discount factor. The action-value function $Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})$, which is often called the Q-function, is defined by

$$Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u}) = r(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u}) + \gamma V(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1)). \qquad (10)$$

The Q-function $Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})$ indicates the expected reward accumulation for the current state and action $(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})$ when the subsequent control signals are generated by the current actor. From equations (9) and (10), Q-function should satisfy the following consistency condition:

$$\begin{aligned} Q(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)) &= r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)) \\ &+ \gamma Q(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1), \mathbf{u}(t+1)), \end{aligned} \qquad (11)$$

which is called the Bellman equation. In our CPG-actor-critic method, the critic approximates the Q-function based on equation (11), like in the Sarsa algorithm [4].

As a function approximator for the critic, we employ a normalized Gaussian neural network (NGnet). An NGnet, which transforms an N-dimensional vector $\mathbf{X} = (\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})$ to a scalar q, is defined by

$$q = Q(\mathbf{X}) = \sum_{m=1}^{M} \left(\frac{\mathcal{N}_m(\mathbf{X})}{\sum_{j=1}^{M} \mathcal{N}_j(\mathbf{X})} \right) \mathbf{K}'_m \tilde{\mathbf{X}} \qquad (12)$$

$$\mathcal{N}_m(\mathbf{X}) \equiv (2\pi)^{-N/2} |\boldsymbol{\Sigma}_m|^{-1/2} \exp\left[-\frac{1}{2} (\mathbf{X} - \boldsymbol{\mu}_m)' \boldsymbol{\Sigma}_m^{-1} (\mathbf{X} - \boldsymbol{\mu}_m) \right],$$

where $(')$ denotes a vector transpose, and $\tilde{\mathbf{X}}' \equiv (\mathbf{X}', 1)$. M denotes the number of units, and $|\cdot|$ is a determinant. $\mathcal{N}_m(\mathbf{X})$ represents a N-dimensional Gaussian function whose mean and covariance matrix are $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$, respectively, and $\mathbf{K}_m$ represents a linear regression matrix.

An NGnet is formulated as a stochastic model, in which an pair of an input vector and an output vector, $(\mathbf{X}, q)$, is a stochastic event. For each event, a unit index $\{m \in 1, 2, \ldots, M\}$ is assumed to be selected as a hidden variable. $(\mathbf{X}, q, m)$ is called a complete event, and the stochastic model is defined by the probability distribution:

$$P(\mathbf{X}, q, m|\theta) = M^{-1} \mathcal{N}_m(\mathbf{X}) (2\pi)^{-1/2} \sigma_m^{-1} \exp\left[-\frac{1}{2\sigma_m^2} (q - \mathbf{K}_m \cdot \tilde{\mathbf{X}})^2 \right], \qquad (13)$$

where $\theta = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, \sigma_m, \mathbf{K}_m | m = 1, \ldots, M\}$ is the model parameter. σ_m is the standard deviation of output q. Under this distribution, the expected output $E[q|\mathbf{X}] \equiv \int qP(q|\mathbf{X})dq$ for a given $\mathbf{X}$ is identical to the output of NGnet (equation (12)). Namely, the probability distribution (13) defines the stochastic model of the NGnet. The model parameter of the stochastic model, θ, is estimated by the maximum likelihood estimation. An EM algorithm can be applied to stochastic models with hidden variables. An efficient on-line EM algorithm for the NGnet has been devised in [5].

In the CPG-actor-critic method, the Q-function predicted by the critic depends on the current actor, and the actor learns based on the critic's prediction. Then, the target function approximated by the critic changes along the learning. Since the on-line EM algorithm [5] has dynamic unit allocation mechanisms, it is suited for learning dynamic environments such as the case of the CPG-actor-critic model.

4.2. Episodic learning

The whole learning process with the CPG-actor-critic model is as follows.

Critic learning phase The current actor receives a CPG-coupled system state $(\boldsymbol{\nu}(t), \mathbf{x}(t))$ and outputs an indirect control signal $\mathbf{u}(t)$ according to equation (8). Then, the basic CPG and the physical system changes their states to $(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1))$, according to equations (1)-(7).

The critic receives an immediate reward $r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$, and is trained in an online fashion so to satisfy the consistency condition (11). The input to the critic is the CPG-coupled system state and the actor output, $(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$, and the target output is the right hand side of equation (11). The parameter of the NGnet, θ, is updated by using the on-line EM algorithm.

The learning system repeats the control and the critic learning process above, until t_{max} sec elapses or the physical state becomes a terminal state. During this time period, the actor is fixed, and the CPG-coupled system state and the action trajectory $\{\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t) | t = 0, 1, \ldots, t_{max}\}$ are saved.

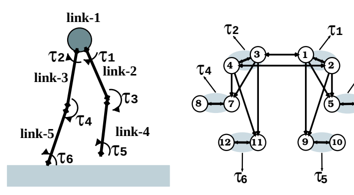
Actor learning phase In the second phase, the actor is trained using the saved trajectory. In order to increase the Q-function value, the connection weight of the actor are updated according to the gradient ascent method:

$$\Delta\psi \propto \frac{\partial Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \psi}, \tag{14}$$

where ψ represents connection weights of the actor, i.e., W_{ij}^{act} and A_{ik}. The derivative of the Q-function is calculated by using the current critic.

The above two phases constitute one episode, and the RL proceeds by repeating these episodes.

5. Biped Robot



(a) Biped Robot (b) CPG

Figure 3:

We apply the CPG-actor-critic method to the biped robot simulator [3], depicted in Fig.3(a). The biped robot is constructed by five links connected to each other. The motions of these links are restricted in a sagittal plane. Link-1 is a point mass representing the upper body. Each leg consists of thigh (link-2 and 3) and shank (link-4 and 5), and has three joints, i.e., hip, knee and ankle. The length of thigh or shank is 0.5m or 0.6m, respectively. The robot is controlled by the torque τ_1-τ_6, each of which is applied to a joint. When a shank is off the ground, the torque at the ankle joint is not generated.

A biped robot state is described by $\mathbf{x} = (x_1, h_1, \theta_2, \theta_3, \theta_4, \theta_5\ \dot{x}_1, \dot{h}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5)$, where x_1 and h_1 denote the horizontal and vertical coordinates of link-1, and θ_i represents the angle of link-i from the vertical axis.

5.1. CPG controller

The structure of the neural oscillator network, which composes the CPG controller for the biped robot, is also adopted from [3]. There are 24 neurons; the i-th and the $(i+12)$-th neurons are called a primary and a supplementary neurons, respectively. Each supplementary neuron is solely connected to its primary neuron by excitation-inhibition mutual connections. An output of the i-th primary neuron is given by $G_i(\nu_i) = \max(0, \nu_i)$, and an output of the $(i+12)$-th supplementary neuron is given by $y_{i+12} = \nu_{i+12}$, for

$i = 1, \ldots, 12$. The dynamics of the neural oscillator network is described as

$$\begin{aligned} c_i \dot{\nu}_i &= -\nu_i - \beta y_{i+12} + I_i, \quad y_i = \max(0, \nu_i) \\ c_{i+12} \dot{\nu}_{i+12} &= -\nu_{i+12} + y_i, \quad y_{i+12} = \nu_{i+12} \\ &\text{for } i = 1, \ldots, 12. \end{aligned}$$

The $(2k-1)$-th, $2k$-th, $(2k+11)$-th and $(2k+12)$-th neurons ($k = 1, 2, \ldots, 6$) compose a single neural oscillator, and hence there are 6 neural oscillators connected mutually as in Fig.3(b). If there is no sensory feedback signal, each neural oscillator spontaneously oscillates, and hence outputs periodic signals.

Torque τ_i, which is applied to the i-th joint (Fig.3) is calculated from the output of the CPG neurons:

$$\begin{aligned} \tau_i &= -T_i^F y_{2i-1} + T_i^E y_{2i} \quad (i = 1, \ldots, 4) \\ \tau_i &= (-T_i^F y_{2i-1} + T_i^E y_{2i}) \Xi_i \quad (i = 5, 6). \end{aligned} \tag{15}$$

where Ξ_i represents an indicator function of shank link-i ($i = 4, 5$), i.e., $\Xi_i = 1$ (or 0) when the link-i touches (or, is off) the ground. T_i^F and T_i^E represent weights of the flexor and extensor, respectively, and their values are adopted from [3]. $\tau_{1,2}$, $\tau_{3,4}$ and $\tau_{5,6}$ represent the torque applied to hip, knee and ankle, respectively. There is an additional feedback torque, i.e., when the knee joint angle becomes larger than π, an impulsive torque is applied to limit the angle within the range $[0, \pi]$.

A sensory feedback signal from the biped robot is $\mathbf{S} = \{\theta_2, \theta_3, \theta_4\Xi_4, \theta_5\Xi_5, \Xi_4, \Xi_5, \dot{\theta}_4\Xi_4, \dot{\theta}_5\Xi_5\}$; this is also adopted from [3].

6. Experiment

The aim of the simulation experiment is examine whether the CPG-actor-critic method is able to adjust the actor parameter such that the biped robot is able to walk stably. For simplicity of the learning, we assume here all mutual connection weights in the neural oscillator network are fixed, namely, $W_{ij}^{act} \equiv 0$. We also assume that connection patterns from sensory feedback signal to the CPG neurons have specific forms, as in [3]:

$$\begin{aligned} I_1^{ext} &= a_1 S_1 - a_2 S_2 + a_3 S_3 + a_4 S_6, \\ I_3^{ext} &= a_1 S_2 - a_2 S_1 + a_3 S_4 + a_4 S_5, \\ I_5^{ext} &= a_5 S_4, \quad I_7^{ext} = a_5 S_3, \\ I_9^{ext} &= -a_6 S_3 - a_7 S_4 - a_8 S_7, \\ I_{11}^{ext} &= -a_6 S_4 - a_7 S_3 - a_8 S_8, \\ I_{2i}^{ext} &= -I_{2i-1}^{ext} \quad \text{for } i = 1, \ldots, 6. \end{aligned} \tag{16}$$

In the following experiment, therefore, $\{a_i | i = 1, \ldots, 8\}$ are adjusted by the CPG-actor-critic method.

We assume that an immediate reward $r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$ is determined by the next robot state $\mathbf{x}(t+1)$:

$$\begin{aligned} &\tilde{r}(\mathbf{x}) = 0.5 r_{height}(\mathbf{x}) + 0.02 r_{speed}(\mathbf{x}) \\ &\left\{ \begin{array}{l} r_{height}(\mathbf{x}) = h_1 - 0.8 - \min(h_4, h_5) \\ r_{speed}(\mathbf{x}) = \max(-1, \min(\dot{x}_1, 1)) \end{array} \right. , \end{aligned}$$

where h_i ($i = 4, 5$) represents the height of link-i. $r_{height}(\mathbf{x})$ encourages the robot not to fall down, and $r_{speed}(\mathbf{x})$ encourages the robot to proceed to the forward direction.

The maximum period length in one episode was 5 sec. If the robot tumbled before 5 sec, the episode ended at that point.

At the beginning in each episode, the robot was initialized such that it did not move and two legs were slightly opened. In the progress of learning, if the robot did not fall down during one episode, the initial state of the CPG-coupled system in the subsequent episodes was taken from one of the states in the successful episode.

The RL proceeds by repeating such learning procedure.

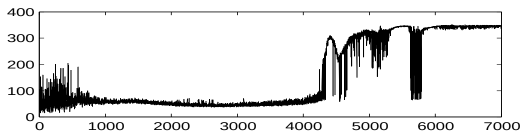


Figure 4: Learning curve

Fig.4 shows the learning curve. The horizontal axis denotes the number of learning episodes, and the vertical axis denotes the amount of rewards received in one episode. At the beginning of the learning, the robot easily fell down. After about 4000 learning episodes, the robot started to run in place, and after about 5800 learning episodes, the robot started to walk stably. Fig.5 shows a stroboscopic gate pattern of the biped robot after 7000 learning episodes. The sensory feedback weight values after 7000 learning episodes were $\mathbf{a}_{RL} = \{10.0, -3.1, 10.0, 6.5, 1.1, 2.5, 10.0, 4.9\}$, which are quite different from the hand-tuned value found in [3], $\mathbf{a}_{HT} = \{1.5, 1.0, 1.5, 1.5, 3.0, 1.5, 3.0, 1.5\}$.

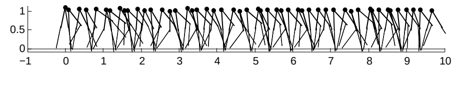


Figure 5: A gate pattern after learning

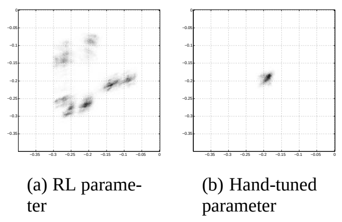

(a) RL parameter (b) Hand-tuned parameter

Figure 6: Return map

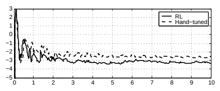


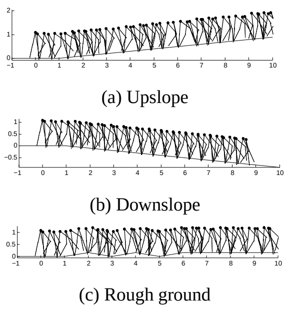
Figure 7: Maximum Lyapunov exponent

The comparison of RL and hand-tuned parameter Fig.6 shows the return map of θ_2 (the thigh angle) whose Poincare section is $y_3 = 3.0$. Figs.6(a) and 6(b) are the return maps when the robot is controlled by $\mathbf{a}_{RL}$ and $\mathbf{a}_{HT}$, respectively. Although the return map by using $\mathbf{a}_{RL}$ has a complex shape, the maximum Lyapunov exponent of the acquired dynamics is negative, as shown in Fig.7. This suggests that the robot trajectory generated by the trained controller $\mathbf{a}_{RL}$ seems to be an aperiodic attractor with external disturbances. When the robot's foot touches the ground or the knee joint angle becomes larger than π, impulsive forces are generated and these forces act an external noise.

Figs.8 and 9 show the gate patterns on various grounds. The biped robot controlled by the learned parameter $\mathbf{a}_{RL}$ is able to walk on inclined or rough ground more stably than by the hand-tuned parameter $\mathbf{a}_{HT}$.

7. Concluding remarks

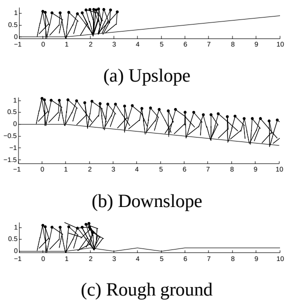
In this article, we proposed a new RL method called the CPG-actor-critic method and applied it to automatic acquisition of the biped locomotion. By using our method, a CPG controller that can walk in various environments more stably than the hand-tuned one was obtained. Although the result was successful, the learning process was still unstable. It is necessary to enhance the stability of our RL method in the near future. In the current simulation, the CPG internal weights were fixed and only sensory feedback connections were adjusted by RL. It also remains a future study to adjust the CPG internal weights by the CPG-actor-critic method.


(a) Upslope

(b) Downslope

(c) Rough ground

Figure 8: Three gate patterns controlled by $\mathbf{a}_{RL}$


(a) Upslope

(b) Downslope

(c) Rough ground

Figure 9: Three gate patterns controlled by $\mathbf{a}_{HT}$

Acknowledgments

This research was supported in part by the Telecommunicatios Advancement Organization of Japan.

References

[1] Hirai, K., et al. 1998. The development of Honda humanoid robot. *Proceedings of ICRA* 2:1321-1326

[2] Grillner, S., Wallen, P. and Brodin, L. 1991. Neuronal network generating locomotor behavior in lamprey. *Annu. Rev. Neurosci.* 14:169-199

[3] Taga, G., Yamaguchi, Y., and Shimizu, H. 1991. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biol. Cybern.* 65:147-159

[4] Sutton, R. S. and Barto, A. G. 1998. *Reinforcement learning.* MIT Press

[5] Sato, M. and Ishii, S. 2000. On-line EM algorithm for the normalized Gaussian network. *Neural Computation* 12:407-432

[6] Sato, M. and Ishii, S. 1999. Reinforcement learning based on on-line EM algorithm. *NIPS 11*, 1052-1058

[7] Morimoto J. and Doya K. 2001. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robot. Auton. Syst.*, 36:37-51.

[8] Ogihara, N. and Yamazaki, N. 2001. Generation of human bipedal locomotion by a bio-mimetic neuro-musculo-skeletal model. *Biol. Cybern.* 84:1-11.