Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 54 (2006) 911–920
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Learning CPG-based biped locomotion with a policy gradient method

Takamitsu Matsubara[a,c,*], Jun Morimoto[b,c], Jun Nakanishi[b,c], Masa-aki Sato[c], Kenji Doya[a,c,d]

[a] *Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0101, Japan*
[b] *ICORP, JST, 2-2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan*
[c] *ATR, CNS, 2-2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan*
[d] *Neural Computation Unit, Initial Research Project, Okinawa Institute of Science and Technology, 12-22, Suzaki, Gushikawa, Okinawa 904-2234, Japan*

Received 15 October 2005; received in revised form 12 May 2006; accepted 24 May 2006
Available online 1 August 2006

Abstract

In this paper, we propose a learning framework for CPG-based biped locomotion with a policy gradient method. We demonstrate that appropriate sensory feedback to adjust the rhythm of the CPG (Central Pattern Generator) can be learned using the proposed method within a few hundred trials in simulations. We investigate linear stability of a periodic orbit of the acquired walking pattern considering its approximated return map. Furthermore, we apply the controllers acquired in numerical simulations to our physical 5-link biped robot in order to empirically evaluate the robustness of walking in the real environment. Experimental results demonstrate that the robot was able to successfully walk using the acquired controllers even in the cases of an environmental change by placing a seesaw-like metal sheet on the ground and a parametric change of the robot dynamics with an additional weight on a shank, which was not modeled in the numerical simulations.

Keywords: Reinforcement learning; Policy gradient; Biped locomotion; Central pattern generator

1. Introduction

Recently, there has been a growing interest in biologically inspired control approaches for biped locomotion using neural oscillators (e.g., [1]) as a central pattern generator (CPG). Notably, Taga [2] demonstrated the effectiveness of this approach for biped locomotion to achieve the desired walking behavior in unpredicted environments in numerical simulations. Following this pioneering work, several attempts have been made to explore neural oscillator based controllers for legged locomotion [3,4]. Neural oscillators have desirable properties such as entrainment through interaction with the environment. However, in order to achieve the desired behavior of the oscillators, much effort is required in manually tuning their parameters. Our goal in this study is to develop an efficient learning framework for CPG-based locomotion of biped robots.

As parameter optimization methods for CPG-based locomotion controllers, a genetic algorithm [5] and reinforcement learning [6] were applied to determine the open parameters of the CPG considering high dimensional state space. However, these methods often require a large number of iterations to obtain the solution, and typically suffer from high computational costs with an increase of dimensionality of the state space. These undesirable features make it infeasible to directly apply these methods to real robots in real-time implementation.

In this paper, we focus on learning appropriate sensory feedback to the CPG in order to achieve the desired walking behavior. The importance of sensory feedback to CPG in order to achieve adaptation to the environment is pointed out in [2]. We propose a learning framework for a CPG-based biped locomotion controller using a policy gradient reinforcement learning method for a 5-link biped robot (Fig. 1). The policy gradient method is a technique for maximizing an accumulated reward with respect to the parameters of a stochastic policy by trial-and-error in an unknown environment [7–11]. However, the policy gradient method also suffers from high computational costs with an increase of dimensionality of the state space when the use of function approximator with a large number of parameters is desirable to represent a nonlinear controller. Thus, in order to reduce the dimensionality of the

* Corresponding author at: Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0101, Japan. Fax: +81 774 95 1236.
E-mail address: takam-m@atr.jp (T. Matsubara).

0921-8890/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2006.05.012

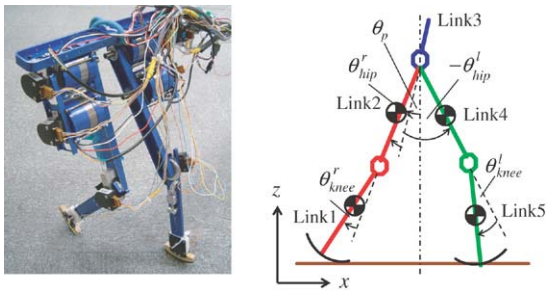


Fig. 1. 5-link biped robot (left) and its model (right). x–z plane is defined as "sagittal plane".

state space used for learning, we only use partial physical states of the robot in our proposed learning system, i.e., we do not use internal states of the CPG and the rest of the states of the robot for learning. As a result, we conceive of the proposed learning framework as a partially observable Markov decision process (POMDP). A general reinforcement learning approach, for example, Q-learning, finds a deterministic optimal policy that maximizes the values of all the states simultaneously assuming that the environment has the Markov property [12]. However, as discussed in [13], stochastic policies often show better performance than deterministic policies in POMDPs. Moreover, the effectiveness of the policy gradient method in POMDPs has been empirically demonstrated for a 4-legged robot [14] and a passive-dynamics based biped robot [15]. Thus, we choose to use a policy gradient method for our learning system among other possible reinforcement learning methods.

This paper is organized as follows. In Section 2, we introduce a central pattern generator which is used for generation of walking behavior in this study. In Section 3, we describe a policy gradient reinforcement learning method for a CPG-based biped locomotion controller. In Section 4, we present the proposed control architecture for our 5-link robot to achieve biped walking behavior. In Section 5, first, we demonstrate the effectiveness of the proposed learning framework by numerical simulations. Then, we investigate linear stability of a periodic orbit of the acquired walking pattern considering its approximated return map. In Section 6, we present experimental results suggesting that successful biped walking with our physical 5-link robot can be achieved using the controller learned in numerical simulations. Moreover, we analyze convergence properties to steady-state walking with variations in initial conditions based on a return map, and experimentally demonstrate the robustness of the acquired walking against environmental changes and parametric changes in the robot dynamics. In Section 7, we summarize this paper and discuss the approaches of policy search in motor learning and control. Finally, we address open problems and future work.

2. Central pattern generator

The CPG-based controller is composed of a neural oscillator model and a sensory feedback controller which maps the states of the robot to the input to the neural oscillator model. In Section 2.1, we present the neural oscillator model. Then, in Section 2.2, we introduce sensory feedback to the neural oscillator model.

2.1. Neural oscillator model

We use a neural oscillator model proposed by Matsuoka [1]. The oscillator dynamics of i-th unit are:

$$\tau_{CPG}\dot{z}_i(t) = -z_i(t) - \sum_{j=1}^{n} w_{ij}q_j(t) - \beta p_i(t) + z_0 + v_i(t), \quad (1)$$

$$\tau'_{CPG}\dot{p}_i(t) = -p_i(t) + q_i(t), \quad (2)$$

$$q_i(t) = \max(0, z_i(t)), \quad (3)$$

where n is the number of neurons, $z_i(t)$ and $p_i(t)$ are internal states of a CPG. τ_{CPG} and τ'_{CPG} are time constants for the internal states. w_{ij} is a inhibitory synaptic weight from the j-th neuron to the i-th neuron. z_0 is a bias. $v_i(t)$ is a feedback signal which will be defined in (4) below.

2.2. Sensory feedback

The feedback signal to the neural oscillator model $v_i(t)$ in Eq. (1) is given by

$$v_i(t) = v_i^{\max} g(a_i(t)), \quad (4)$$

where $g(a_i) = \frac{2}{\pi}\arctan\left(\frac{\pi}{2}a_i\right)$, and $v_i^{\max}$ is the maximum value of the feedback signal. The output of the feedback controller $\mathbf{a} = (a_1, \ldots, a_m)^{\mathrm{T}}$ is sampled from a stochastic policy:

$$\pi(\mathbf{x}, \mathbf{a}; \mathbf{W}^{\mu}, \mathbf{w}^{\sigma}) = \frac{1}{(\sqrt{2\pi})^m |\mathbf{D}(\mathbf{w}^{\sigma})|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{a} - \boldsymbol{\mu}(\mathbf{x}; \mathbf{W}^{\mu}))^{\mathrm{T}}\mathbf{D}^{-1} \times (\mathbf{w}^{\sigma})(\mathbf{a} - \boldsymbol{\mu}(\mathbf{x}; \mathbf{W}^{\mu}))\right), \quad (5)$$

where $\mathbf{x}$ are partial states of the robot. $\mathbf{W}^{\mu}$ is the $m \times k$ parameter matrix, $\mathbf{w}^{\sigma}$ is the m-dimensional parameter vector of the policy, where m is the number of outputs, and k is the number of parameters. $\boldsymbol{\mu}(\mathbf{x}; \mathbf{W}^{\mu})$ is the mean vector of the policy. The covariance matrix $\mathbf{D}$ is defined as $\mathbf{D}(\mathbf{w}^{\sigma}) = \mathbf{S}^{\mathrm{T}}(\mathbf{w}^{\sigma})\mathbf{S}(\mathbf{w}^{\sigma})$. We can equivalently represent $\mathbf{a}$ by

$$\mathbf{a}(t) = \boldsymbol{\mu}(\mathbf{x}(t); \mathbf{W}^{\mu}) + \mathbf{S}(\mathbf{w}^{\sigma})\mathbf{n}(t), \quad (6)$$

where $\mathbf{n}(t) \in \Re^m$ is a noise vector and $n_i(t)$ is sampled from the normal distribution with the mean of 0 and the variance of 1. Note that the matrix $\mathbf{S}(\mathbf{w}^{\sigma})$ must be chosen such that $\mathbf{D}(\mathbf{w}^{\sigma})$ is positive definite.

3. Learning sensory feedback to CPG with a policy gradient method

We describe the use of a policy gradient method in order to acquire a policy of the sensory feedback controller to the neural oscillator model. In Section 3.1, we first define the value function and temporal difference error (TD error) in continuous time and space [16], which is used in the policy gradient method. In Section 3.2, we describe the learning method to improve the policy of the sensory feedback controller.

3.1. Learning the value function

Consider a continuous-time system which represents both the robot and the CPG dynamics,

$$\frac{\mathrm{d}\mathbf{x}^{all}(t)}{\mathrm{d}t} = f(\mathbf{x}^{all}(t), \mathbf{a}(t)), \tag{7}$$

where $\mathbf{x}^{all} \in X \subset \Re^l$ consists of the state of the robot and the CPG, and $\mathbf{a} \in A \subset \Re^m$ is the output of the feedback controller, that is, input to the CPG. We denote the immediate reward as

$$r(t) = r(\mathbf{x}^{all}(t), \mathbf{a}(t)). \tag{8}$$

The value function of the state $\mathbf{x}^{all}(t)$ based on a policy $\pi(\mathbf{x}^{all}, \mathbf{a})$ is defined as

$$V^{\pi}(\mathbf{x}^{all}(t)) = E\left\{ \int_t^{\infty} \mathrm{e}^{-\frac{s-t}{\tau}} r(\mathbf{x}^{all}(s), \mathbf{a}(s))\mathrm{d}s \,\middle|\, \pi \right\}, \tag{9}$$

where τ is a time constant for discounting future rewards. The consistency condition for the value function is given by the time derivative of (9) as

$$\frac{\mathrm{d}V^{\pi}(\mathbf{x}^{all}(t))}{\mathrm{d}t} = \frac{1}{\tau} V^{\pi}(\mathbf{x}^{all}(t)) - r(t). \tag{10}$$

We denote a current estimate of the value function as $V(\mathbf{x}^{all}(t)) = V(\mathbf{x}^{all}(t); \mathbf{w}^c)$, where $\mathbf{w}^c$ is the parameter of the function approximator. If the current estimate of the value function V is perfect, it should satisfy the consistency condition (10). If this condition is not satisfied, the prediction should be adjusted to decrease the inconsistency

$$\delta(t) = r(t) - \frac{1}{\tau} V(t) + \dot{V}(t). \tag{11}$$

This is the continuous-time counterpart of the TD error [16]. Because we consider a learning framework in POMDPs, i.e., we observe only the partial state $\mathbf{x}$ of the state $\mathbf{x}^{all}$, the TD error does not usually converge to zero. However, Kimura et al. [8] suggested that the approximated value function can be useful to reduce the variance of the gradient estimation in (13) even if the consistency condition in (10) is not satisfied.

The update laws for the parameter vector of the value function $\mathbf{w}^c$ and the eligibility trace vector $\mathbf{e}^c$ for $\mathbf{w}^c$ are defined respectively as

$$\dot{\mathbf{w}}^c(t) = \alpha\delta(t)\mathbf{e}^c(t), \qquad \dot{\mathbf{e}}^c(t) = -\frac{1}{\kappa^c}\mathbf{e}^c(t) + \frac{\partial V_{\mathbf{w}^c}}{\partial \mathbf{w}^c}, \tag{12}$$

where α is the learning rate and κ^c is the time constant of the eligibility trace.

3.2. Learning a policy of the sensory feedback controller

In [8], Kimura et al. presented that by using TD error $\delta(t)$ and eligibility trace vector $\mathbf{e}^a(t)$, it is possible to obtain an estimate of the gradient of the expected actual return V_t with respect to the parameter vector $\mathbf{w}^a$ in the limit of $\kappa^a = \tau$ as

$$\frac{\partial}{\partial \mathbf{w}^a} E\{V_t \mid \pi_{\mathbf{w}^a}\} = E\{\delta(t)\mathbf{e}^a(t)\}, \tag{13}$$

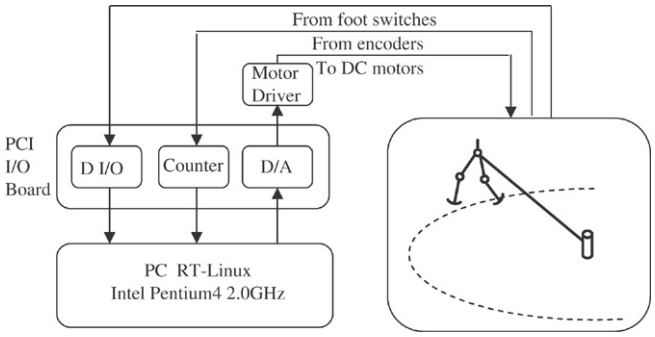


Fig. 2. Experimental setting.

where

$$V_t = \int_t^{\infty} \mathrm{e}^{-\frac{s-t}{\tau}} r(s)\mathrm{d}s, \tag{14}$$

$\mathbf{w}^a$ is the parameter vector of the policy $\pi_{\mathbf{w}^a} = \pi(\mathbf{x}, \mathbf{a}; \mathbf{w}^a)$, and $\mathbf{e}^a(t)$ is the eligibility trace vector for the parameter vector $\mathbf{w}^a$. The parameter vector $\mathbf{w}^a$ is represented as $\mathbf{w}^a = (\mathbf{w}_1^{\mu\mathrm{T}}, \ldots, \mathbf{w}_m^{\mu\mathrm{T}}, \mathbf{w}^{\sigma\mathrm{T}})^{\mathrm{T}}$, where $\mathbf{w}_j^{\mu}$ is the j-th column vector of the parameter matrix $\mathbf{W}^{\mu}$. The update laws for the parameter vector of the policy $\mathbf{w}^a$ and the eligibility trace vector $\mathbf{e}^a(t)$ can be derived respectively as

$$\dot{\mathbf{w}}^a(t) = \beta\delta(t)\mathbf{e}^a(t), \quad \dot{\mathbf{e}}^a(t) = -\frac{1}{\kappa^a}\mathbf{e}^a(t) + \frac{\partial \ln \pi_{\mathbf{w}^a}}{\partial \mathbf{w}^a} \tag{15}$$

where β is the learning rate and κ^a is the time constant of the eligibility trace. In the case of $\kappa^a = \tau \approx \infty$, the actual return used as a criteria for the policy improvement in this algorithm is similar to the average reward criteria widely used in other policy gradient methods [9–11]. (See Section 7.3 for more details.)

4. Control architecture for 5-link robot

In this paper, we use a planar 5-link biped robot (Fig. 1) developed in [17]. The experimental setting is depicted in Fig. 2. The height of the robot is 0.4 m and the total mass is about 2 kg. The length of each link of the leg is 0.2 m. The masses of the body, thigh and shank are 1.0 kg, 0.43 kg and 0.05 kg, respectively. The motion of the robot is constrained within the sagittal plane which is defined as shown in Fig. 1 (right) by a tether boom. The hip joints are directly actuated by direct drive motors, and the knee joints are driven by motors through a wire transmission mechanism with a reduction ratio of 2.0. These transmission mechanisms with low reduction ratio provide high back drivability at the joints. Foot contact with the ground is detected by foot switches. The robot is an underactuated system having rounded soles with no ankles. Thus, it is challenging to design a controller to achieve biped locomotion with this robot since no actuation can be applied between the stance leg and the ground unlike many of the existing biped robots which have flat feet with ankle joint actuation. In the following, we denote the left hip and knee angles by θ_{hip}^l and θ_{knee}^l, respectively. Similar definitions are also applied to the joint angles of the right leg.

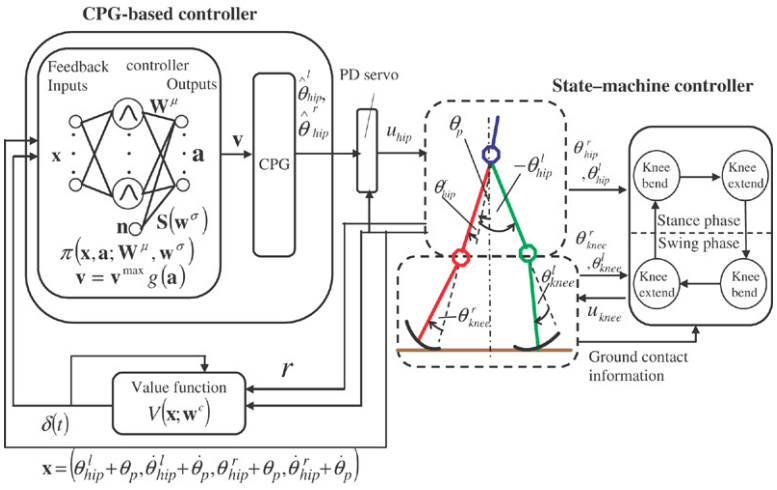


Fig. 3. Proposed control architecture for 5-link biped robot.

Fig. 3 illustrates our control architecture for the biped robot, which consists of the CPG-based controller for the hip joints and the state-machine controller for the knee joints. Section 4.1 presents a CPG-based controller which generates periodic walking patterns. Section 4.2 presents a state-machine controller which makes foot clearance at appropriate timing according to the state of the hip joint and the foot contact information with the ground.

4.1. CPG-based controller for the hip joints

In the proposed control architecture, the hip joints of the robot are driven by the CPG-based controller described in Section 2. The hip joint controller is composed of four neurons ($i = 1, \ldots, 4$) in Eqs. (1)–(3): $i = 1$: extensor neuron for left hip, $i = 2$: flexor neuron for left hip, $i = 3$: extensor neuron for right hip, $i = 4$: flexor neuron for right hip. For the sensory feedback (5), we consider the states of the hip joints $\mathbf{x} = (\theta^l_{hip} + \theta_p, \dot\theta^l_{hip} + \dot\theta_p, \theta^r_{hip} + \theta_p, \dot\theta^r_{hip} + \dot\theta_p)^{\mathrm{T}}$ as the input states. The target joint angle for the hip joint is determined by the oscillator output q_i:

$$\hat\theta^l_{hip} = -q_1 + q_2, \qquad \hat\theta^r_{hip} = -q_3 + q_4. \tag{16}$$

The torque output u at each hip joint is given by a PD controller:

$$u_{hip} = K^{hip}_p(\hat\theta_{hip} - \theta_{hip}) - K^{hip}_d\dot\theta_{hip}, \tag{17}$$

where K^{hip}_p is a position gain and K^{hip}_d is a velocity gain.

4.2. State-machine controller for the knee joints

We design a state-machine controller for the knee joints as depicted in Fig. 4. The state-machine controller changes the pre-designed target joint angles for the knee joints according to transition conditions defined by the hip joint angles and the foot contact information with the ground. The torque command to each knee joint is given by a PD controller:

$$u_{\mathrm{knee}} = K^{\mathrm{knee}}_p(\hat\theta_{\mathrm{knee}} - \theta_{\mathrm{knee}}) - K^{\mathrm{knee}}_d\dot\theta_{\mathrm{knee}}, \tag{18}$$

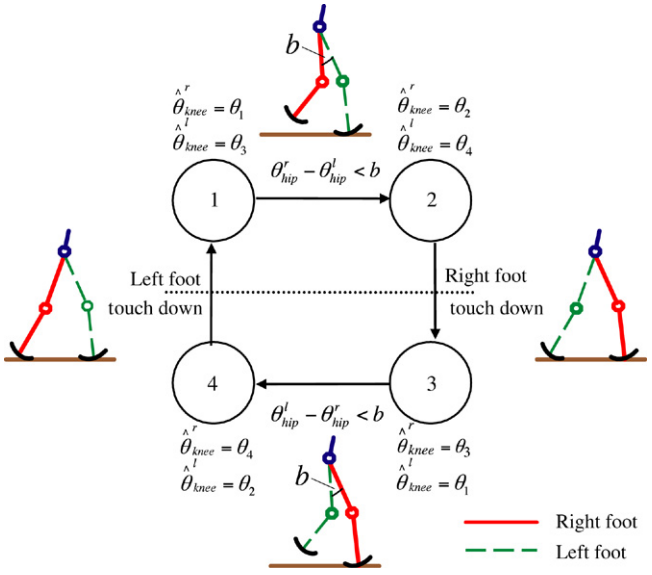


Fig. 4. State transition $1 \sim 4$ in the state-machine controller for knee joints.

where K^{knee}_p is a position gain and K^{knee}_d is a velocity gain. We define four target joint angles, $\theta_1, \ldots, \theta_4$, for the state-machine controller (Fig. 4). We use the hip joint angles and the foot contact information to define the transition conditions of the state-machine controller. The transition conditions defined by the hip joint angles are given by $\theta^l_{hip} - \theta^r_{hip} < b$ or $\theta^r_{hip} - \theta^l_{hip} < b$, where b is a threshold of the transition conditions.

5. Numerical simulations

5.1. Simulation setup

In Section 5.1.1, we present function approximators for the value function and policy of sensory feedback to CPG. In Section 5.1.2, we describe reward function designed to achieve biped walking though learning. In Section 5.1.3, we present the parameter settings in the controller used for the simulations.

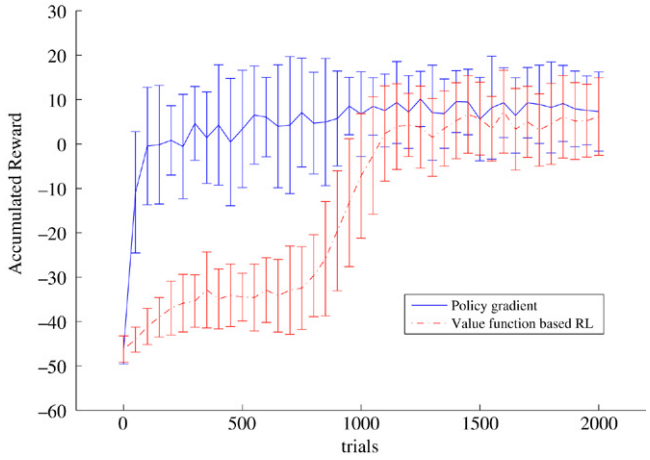


Fig. 5. Accumulated reward at each trial: averaged by 50 experiments and smoothed out by taking a 10-moving average. Error bar is the standard deviation. Solid line: policy gradient method. Dash–dot line: value-function-based RL.

5.1.1. Function approximator for the value function and the policy

We use a normalized Gaussian Network (NGnet) [16] to model the value function and the mean of the policy. This function approximator was used in the authors' previous studies of reinforcement learning in continuous time and space, and shown to be effective in the examples of a swing-up task of an inverted pendulum and a dynamic stand-up behavior of a real robot [16,18]. By this way, it is possible to achieve smooth control compared to the tile-coding approach often used in discrete reinforcement learning [12]. In addition, practical feasibility of this function approximator was demonstrated for real-time implementation of the control policy on a hardware robot to achieve the desired behavior [18]. The variance of the policy is modeled by a sigmoidal function [8]. The value function is approximated with the NGnet:

$$V(\mathbf{x};\mathbf{w}^c) = \mathbf{w}^{c\mathrm{T}}\mathbf{b}(\mathbf{x}) \tag{19}$$

where $\mathbf{b}(\mathbf{x}) = (b_1(\mathbf{x}), b_2(\mathbf{x}), \ldots, b_K(\mathbf{x}))^{\mathrm{T}}$,

$$b_k(\mathbf{x}) = \frac{\phi_k(\mathbf{x})}{\sum_{l=1}^{K}\phi_l(\mathbf{x})} \quad \text{and} \quad \phi_k(\mathbf{x}) = \mathrm{e}^{-\|\mathbf{s}_k^{\mathrm{T}}(\mathbf{x}-\mathbf{c}_k)\|^2}. \tag{20}$$

K is the number of the basis functions, and $\mathbf{w}^c$ is the parameter vector of value function. The vectors $\mathbf{c}_k$ and $\mathbf{s}_k$ define the center and the size of the k-th basis function, respectively. The mean $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{D}$ of the policy are represented with the NGnet and the sigmoidal function, respectively:

$$\boldsymbol{\mu}(\mathbf{x};\mathbf{W}^{\mu}) = \mathbf{W}^{\mu\mathrm{T}}\mathbf{b}(\mathbf{x}), \quad \mathbf{D}(\mathbf{w}^{\sigma}) = \mathbf{S}^{\mathrm{T}}(\mathbf{w}^{\sigma})\mathbf{S}(\mathbf{w}^{\sigma}), \tag{21}$$

where $\mathbf{S}(\mathbf{w}^{\sigma}) = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$,

$$\sigma_i = \frac{1}{1+\exp(-w_i^{\sigma})} \quad \text{and} \quad \mathbf{w}^{\sigma} = (w_1^{\sigma}, w_2^{\sigma}, w_3^{\sigma}, w_4^{\sigma})^{\mathrm{T}}. \tag{22}$$

We locate basis functions $\phi_k(\mathbf{x})$ on a grid with an even interval in each dimension of the input space ($-\frac{\pi}{3} \le \theta_{hip}^l + \theta_p \le \frac{\pi}{3}$, $-3.0\pi \le \dot{\theta}_{hip}^l + \dot{\theta}_p \le 3.0\pi$, $-\frac{\pi}{3} \le \theta_{hip}^r + \theta_p \le \frac{\pi}{3}$, $-3.0\pi \le \dot{\theta}_{hip}^r + \dot{\theta}_p \le 3.0\pi$). We used 9216 ($=12 \times 8 \times 12 \times 8$) basis functions to approximate the value function and the mean of the policy respectively.

5.1.2. Rewards

We used the following simple reward function:

$$r = k_v \max(0, v), \tag{23}$$

where the reward is designed to encourage forward progress of the robot by giving a reward proportional to the forward velocity of walking v. In this study, the parameter for the reward is chosen as $k_v = 0.05$. The robot also receives a punishment (negative reward) $r = -1$ for 0.5 s if it falls over.

5.1.3. Parameters for the controllers

Parameters of the neural oscillators used in (1)–(3) are $\tau_{CPG} = 0.041$, $\tau'_{CPG} = 0.36$, $\beta = 2.5$, $z_0 = 0.4$, $w_{12} = w_{21} = w_{34} = w_{43} = 2.0$, $w_{13} = w_{31} = w_{24} = w_{42} = 1.0$. Initial values of the internal states are given by $z_1(0) = 0.05$, $z_2(0) = 0.05$. We select the learning parameters as $\tau = 1.0$, $\alpha = 50$, $\beta^{\mu} = 20$, $\beta^{\sigma} = 10$, $\kappa^c = 0.1$, $\kappa^{\mu} = 1.0$, $\kappa^{\sigma} = 1.0$. PD gains for hip joints are set to $K_p^{hip} = 4.0$ N m/rad and $K_d^{hip} = 0.07$ N m s/rad, respectively. These CPG parameters were roughly tuned to achieve some desirable natural frequency and amplitude through numerical simulations. However, note that as seen in Fig. 6, the robot cannot walk only with the CPG, i.e., appropriate learned sensory feedback is necessary for successful walking. Moreover, we will demonstrate that choice of CPG parameters does not significantly affect the performance of learning in our proposed framework (see Section 5.2 below). Parameters of the state-machine are $\theta_1 = 32°$, $\theta_2 = 16°$, $\theta_3 = 15°$, $\theta_4 = 7.5°$ and $b = 8.6°$. PD gains for knee joints are chosen as $K_p^{\mathrm{knee}} = 8.0$ N m/rad and $K_d^{\mathrm{knee}} = 0.09$ N m s/rad, respectively.

5.2. Simulation results

In the following simulations, the initial posture of the robot is determined as $\theta_{hip}^l = 5.5°$, $\theta_{hip}^r = -5.5°$, $\theta_p = 0.0°$, $\theta_{\mathrm{knee}}^l = 20.5°$, $\theta_{\mathrm{knee}}^r = 0.0°$ (see the definition of each angle in Fig. 3) and the initial velocity of the robot is randomly sampled from an uniform distribution between 0.05 m/s and 0.20 m/s. In these simulations, we define that a learning episode is successful when the biped robot does not fall over for 10 successive trials. We applied the policy gradient method with these settings to the biped robot. Fig. 5 (solid line) shows an accumulated reward at each trial with the policy gradient method. An appropriate feedback controller of the CPG-based controller was acquired in 181 trials (averaged over 50 experiments). Fig. 6(a) shows the initial walking pattern before learning, where the robot falls over after a few steps. Fig. 6(b) shows an acquired walking pattern at the 1000-th trial with the learned sensory feedback of the CPG-based controller.

As a comparison, we also implemented a value-function-based reinforcement learning method proposed in [16]. The

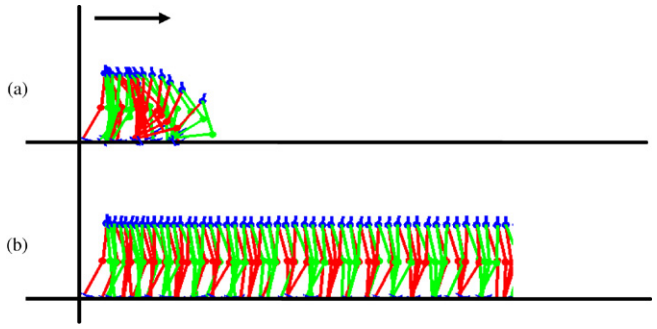


Fig. 6. Acquired biped walking pattern: (a) before learning, (b) after learning. The arrow indicates the direction of walking.

result is also presented in Fig. 5 (dash–dot line). Although the value-function-based RL could also acquire appropriate biped walking controllers, it required a larger number of trials compared with the policy gradient method (1064 trials was required with the value-function-based RL on average). Moreover, we observed that the learning was unstable with higher learning rate in updating of policy parameters with the value-function-based RL. The result is consistent to a consideration that value-function-based reinforcement learning methods are not suitable for POMDPs as pointed out in [13]. This point will be discussed in Section 7.3 in more detail.

We observed a large phase difference between the target and actual trajectories in the hip joints while the knee joint trajectories achieved good tracking of the target. This is due to the choice of low PD gains for the hip joints. Despite this large phase difference between the target and actual hip joint trajectories, the robot could achieve successful walking. This suggests that our method does not necessarily require very accurate tracking with a high gain servo which is typically used in model-based trajectory planning approaches [19–21].

In order to investigate sensitivity of learning against the changes in the CPG parameters, we varied the CPG parameters which characterize the frequency (τ_{CPG}, τ'_{CPG}) and amplitude (z_0) from the values chosen above by $\pm 25\%$, respectively. In all cases, we could acquire successful walking within 1000 trials. We did not observe significant differences in the resultant walking with the learned feedback controller even with these varied CPG parameters. This suggests that careful tuning of CPG parameters is not a prerequisite in our learning framework.

5.3. Linear stability of a periodic orbit in learned biped walking

In this section, we analyze linear stability of a periodic orbit in learned biped walking around a fixed point using a return map [22]. The return map is defined as a mapping of the states of the robot and CPG from the 4th step to the 6th step when the right hip is in swinging phase and the angle is 0.2 rad. The return map is an 18 dimensional mapping which consists of the states of the robot and CPG except for the walking distance of the robot. Initial velocity of the robot was randomly sampled from an uniform distribution between 0.05 m/s and 0.15 m/s, introducing perturbations in each dimension.

We analyzed the linearized return map which was approximated using 1500 sampled data, and confirmed all eigenvalues were inside of the unit circle. The result implies that the periodic biped walking is locally stable around the fixed point.

6. Hardware experiments

In this section, we implement the proposed control architecture on the physical biped robot depicted in Fig. 1. We use the same parameters for the CPG and state-machine and also the same PD gains as used in the numerical simulations presented in Section 5.1. In the state-machine controller, a low-pass filter, with the time constant of 0.03 s, is used to avoid discontinuous change in the target angles of the knee joint, which is practically undesirable. To initiate locomotion in the experiments, we first suspend the robot with the legs swinging in the air, and then place the robot on the ground manually. Thus, the initial condition of each run was not consistent. Occasionally, the robot could not start walking or fell over after a couple of steps when the timing was not appropriate.

6.1. Walking performance of the learned controller in the real environment

We implemented ten feedback controllers acquired in the numerical simulations, and then we confirmed that seven controllers out of ten successfully achieved biped locomotion with the physical robot. Fig. 7 shows the walking pattern without a learned feedback controller, and Fig. 8 shows snapshots of a walking pattern using one of the feedback controllers. Fig. 11 presents trajectories of a successful walking pattern at each joint in the right foot.

6.2. Convergence property from various initial conditions

The robot could achieve biped walking even though the initial conditions in these experiments were not consistent. In order to investigate the convergence property to steady-state walking with variations in initial conditions, we analyze the linear stability of a periodic orbit in learned biped walking

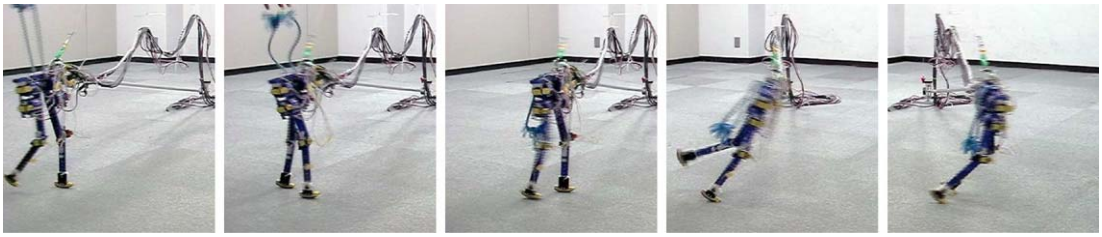


Fig. 7. Initial walking pattern without a feedback controller. The robot could not walk.

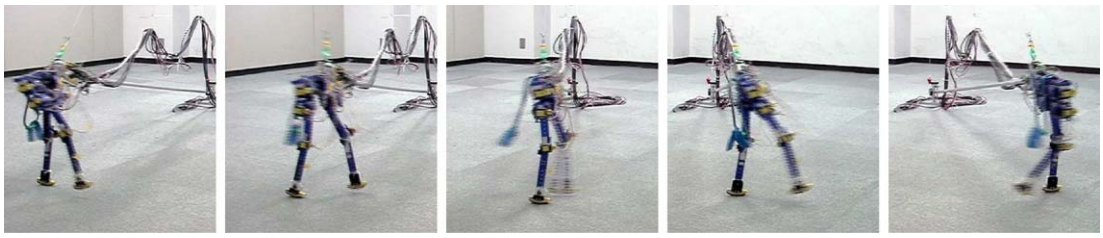


Fig. 8. Successful walking pattern with a learned feedback controller in numerical simulation with 1000 trials.

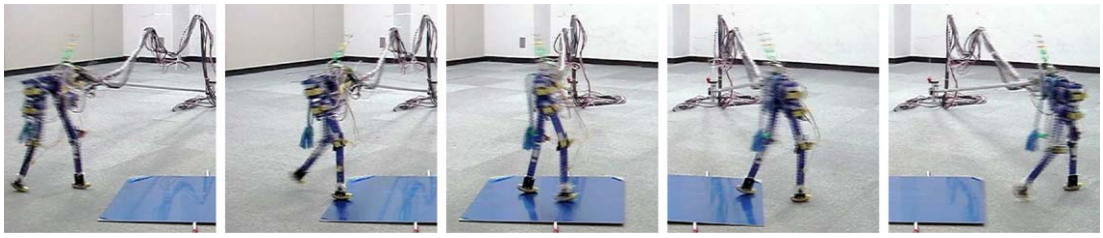


Fig. 9. Example of an environmental change. Walking pattern on a seesaw-like metal sheet.

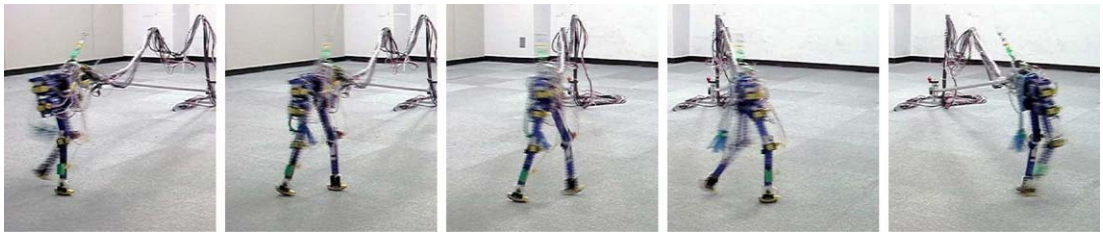


Fig. 10. Example of a parametric change of the robot dynamics. Walking pattern with an additional weight of 150 g on the right shank.

around a fixed point using its return map [22]. We consider a one dimensional return map with respect to the successive step length d defined as a distance between the right and left foot when the right leg touches down with the ground. In Fig. 12, we plot the return map obtained in the experiments. The absolute value of the slope of the return map is 0.82, which is less than 1. The result implies that walking with the physical robot converges to steady-state walking even if the initial conditions are not consistent.

6.3. Robustness of the learned controllers

We experimentally investigate the robustness of the learned controller against environmental changes and parametric changes in the robot dynamics. As an example of an environmental change, we placed a seesaw-like metal sheet with a slight change in the slope on the ground (Fig. 9). As an example of a parametric change in the robot dynamics, we added a weight (150 g) on the right shank (Fig. 10), which is about 38% increase of right leg mass. Figs. 9 and 10 suggest the robustness of the learned walking against environmental changes and parametric changes in the robot dynamics, respectively.

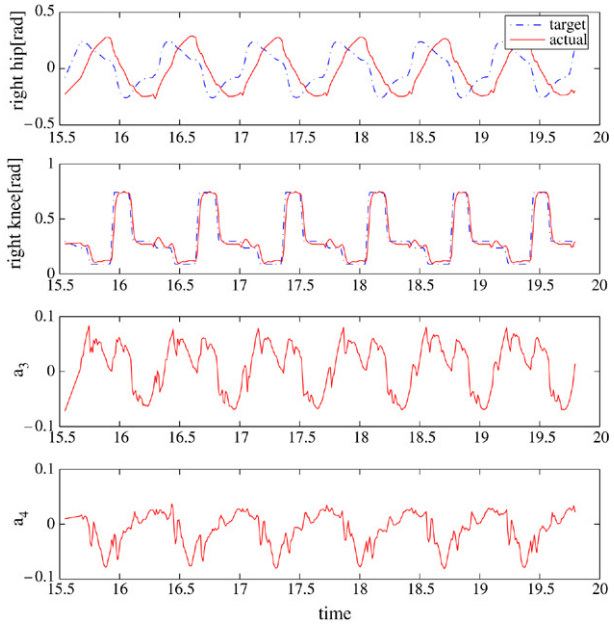


Fig. 11. Joint angles and sensory feedback signals of successful walking with the physical robot using a controller acquired in numerical simulations. The top and second plots are joint trajectories of the right hip and knee, respectively. The third and the bottom plots show sensory feedback signals corresponding to the extensor and flexor neurons for the right hip joint, respectively.

7. Discussion

7.1. Summary

In this paper, we presented a learning framework for a CPG-based biped walking controller with a policy gradient method. Numerical simulations demonstrated that an appropriate sensory feedback controller to the CPG could be acquired

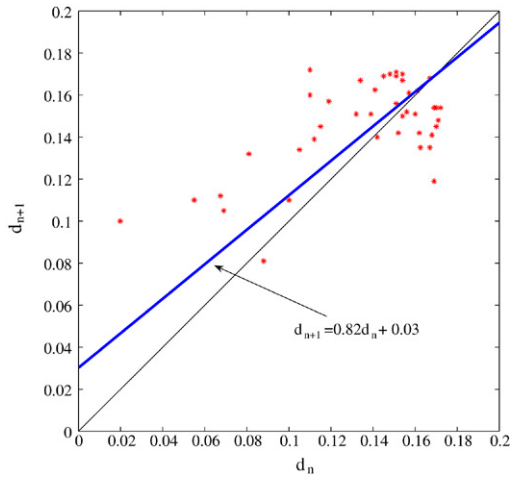


Fig. 12. The linearized return map of acquired walking with the physical robot. d is a step length when the right leg touches down with the ground. The thick line is the return map from d_n to d_{n+1}, and the thin line represents the identity map.

with the proposed learning architecture to achieve the desired walking behavior. We showed that the acquired walking was a locally stable periodic orbit based on a linearized return map around a fixed point. We also implemented the learned controller on the physical 5-link biped robot. We analyzed the convergence property of the learned walking to steady-state walking with variations in initial conditions based on a return map, and experimentally demonstrated the robustness of the learned controller against environmental changes and parametric changes in robot dynamics such as placing a seesaw-like metal sheet on the ground and adding a weight. As a immediate next step, we address improvement of the acquired controller by additional learning with the physical robot.

7.2. Issues in motor skill learning with reinforcement learning

In this study, our general interest is in the acquisition of motor skills or dynamic behavior of complex robotic systems such as humanoid robots. This paper has focused on the development of a learning framework for a simple biped robot to achieve the desired walking behavior. Among learning motor skill problems, in particular, learning biped locomotion is a challenging task which involves dynamic interaction with an environment and it is desirable that the controller be robust enough to deal with uncertainties of the environment and unexpected disturbances.

Model-based approaches for motion generation of biped robots have been successfully demonstrated to be effective [19–21]. However, they typically require precise modeling of the dynamics of the robot and the structure of the environment. Thus, we employed our proposed CPG-based control framework with a policy gradient method which does not require such precise models to achieve robust locomotion in an unstructured environment. Our empirical results demonstrated the effectiveness of the proposed approach. However, in general, there are difficulties in the application of the reinforcement learning framework to motor skill learning with robotic systems. First, in motor control, it is desirable to use smooth continuous actions, i.e., the output of the policy should be smooth and continuously computed from the current state which is typically measured by sensors in the real robotic systems. Previously, in many applications of reinforcement learning, discretization techniques have been widely used [12]. However, as pointed out in [16], coarse discretization may result in poor performance, and fine discretization would require a large number of states and iteration steps. Thus, in order to deal with continuous state and action, we find it useful to use function approximators. Moreover, the use of algorithms derived in continuous time is also suitable for such dynamical systems [16]. Second, when considering hardware implementation of the policy for robot control, calculation of motor commands needs to be done in real-time. Thus, computationally efficient representation of the policy should be considered. To our knowledge, there have been few successful applications of a reinforcement learning framework to motor skill learning with physical robotic systems [14,18] in which the dimensionality of the systems is still relatively small. In this research, we used a CPG-based controller to achieve robust biped walking with a rather high dimensional system. The use of a CPG-based controller also makes learning of such a complex motor skill much simpler by introducing a periodic rhythm. However, still other alternative approaches and algorithms can be considered. In the following section, we discuss several possible policy search approaches which might be applicable to the learning problem in this paper.

7.3. Comparison to alternative policy search approaches

In this paper, we adopted a policy gradient method proposed in [8] as a method for policy search in a CPG-based locomotion controller. This section discusses possible alternative policy search approaches, for example, genetic algorithms [5,23], value-function-based reinforcement learning methods [12,16], and other policy gradient algorithms such as GPOMDP [10] and IState-GPOMDP [24].

Genetic algorithms (GAs) are an optimization method inspired by evolution processes. This method is known to be effective for complex search problems (typically discrete problems) in a large space, and also was applied to policy search in biped locomotion [5] and locomotion of a snake-like robot [23]. However, the optimization process does not use the gradient information which is useful to determine how to improve the policy in a systematic manner. Also, there are a number of open design parameters, for example, the number of individuals and the probability of mutation, which need to be determined somewhat in a heuristic manner. Moreover, there is a problem of policy coding—it is not clear how to represent a policy in an appropriate way for the given problem.

Value function based reinforcement learning (RL) methods have been successfully applied to many policy search problems [16,18,25,26]. However, value function based RL assumes MDPs (Markov decision process) in which all the states are observable, and it is not suitable for POMDPs as pointed out in [13]. In fact, we performed additional numerical simulations to test a value function based RL for the locomotion task in the

same simulation settings which is conceived of as a POMDP (see the result in Fig. 5). However, the value function based RL[1] needed a larger number of trials to acquire an appropriate feedback controller compared with the policy gradient method [8] in this POMDP environment. There is still a possibility to consider full state observation including all the robot states and the internal states of the CPG to make the learning problem of biped locomotion be an MDP. However, due to the significant increase of dimensionality of the state space, it is computationally too expensive for real-time implementation in a hardware system in the current settings. These observations above indicate that policy search algorithms which are capable of handling the POMDP situation would be preferable.

Policy gradient methods are policy search techniques which are suitable for POMDPs [7,8,10]. In this paper, we chose to use the policy gradient algorithm proposed in [8] as a policy search method, which has been empirically shown to be effective as a learning method for physical legged robotic systems [14,15] in POMDPs. We would like to mention that this algorithm is essentially equivalent to another policy gradient algorithm, GPOMDP, developed by Baxter [10]. Although objective functions used in Kimura's algorithm (expected actual return) and Baxter's GPOMDP algorithm (average reward) are different, [10] shows that the gradients of the expected actual return is proportional to the gradient of the average reward. Both Kimura's formulation and Baxter's formulation obtain a gradient of the average reward with respect to the policy parameters if the probability distribution of all the state and action is known, i.e., the environment is completely known. However, in practice, we need to estimate the environment's dynamics from sampled data when there is no prior knowledge of the environment. In such a case, Kimura's algorithm which uses an approximated value function as *the reward baseline* (introduced in [7]) is empirically shown to be advantageous in reducing the variance of the estimation of the policy gradient [8].

Finally, we would like to mention the internal-state policy gradient algorithm for POMDP (IState-GPOMDP) which has internal states with memory as an extension of the GPOMDP algorithm [24]. Conceptually, this framework has a similarity to the structure of our learning system with the CPG in that it contains internal states. Thus, there might be a potential possibility to optimize the parameters of the learning system including the mapping from the oscillator output to the torque at hip joints, which was implemented by a pre-designed PD controller in (17). However, learning additional parameters would be computationally too expensive due to the complex representation of the entire policy, and therefore would not be suitable for real-time implementation in a hardware system in the current settings.

Although policy gradient methods are generally considered to be suitable for POMDPs, these methods find a local optimal solution only within the parameter space of the state-dependent policy designed in advance. In this study, we manually selected the partial states (only hip joint states) for the policy from all the states including the robot and CPG. Because of this simplification, the real-time implementation was achieved in the current settings. On the other hand, this simplification might reduce the performance of the resultant policy acquired though learning. One of the key factors for successful learning in this study was the choice of those partial states selected by our intuition, which are likely to be dominant states in our proposed CPG-based biped locomotion controller. If a different simplification is introduced for this learning task, for example if CPG's internal states are only used for the learning, acquired controllers may not be good enough to achieve biped walking.

7.4. Open problems and future work

With the recent progress in the theoretical studies of policy gradient methods and development of advanced algorithms, it is possible to improve the policy for a given reward using a policy gradient method towards a local optimal policy. One's hope is that the desired task or behavior can be achieved if the reward is chosen appropriately. In this paper, we have successively achieved learning CPG-based biped walking with a robot which is considered as a high dimensional system for learning using a policy gradient method by using simple reward in (23). However, it is not clear how to choose a reward which best describes the desired task. This remains still an open problem not only in the policy gradient method but also in the reinforcement learning framework in general. The direction of our future work is towards development of an efficient learning algorithm for integration of dynamic behaviors, e.g., combination of walking, balancing and reactive motions against unexpected disturbances, in highly complex systems such as full body humanoid robots addressing the problem above.

References

[1] K. Matsuoka, Sustained oscillations generated by mutually inhibiting neurons with adaptation, Biological Cybernetics 52 (1985) 367–376.
[2] G. Taga, Y. Yamaguchi, H. Shimizu, Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment, Biological Cybernetics 65 (1991) 147–159.
[3] Y. Fukuoka, H. Kimura, A. Cohen, Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts, The International Journal of Robotics Research 22 (3–4) (2003) 187–202.
[4] G. Endo, J. Morimoto, J. Nakanishi, G. Cheng, An empirical exploration of a neural oscillator for biped locomotion control, in: IEEE International Conference on Robotics and Automation, 2004, pp. 3036–3042.
[5] K. Hase, N. Yamazaki, Computer simulation of the ontogeny of biped walking, Anthropological Science 106 (4) (1998) 327–347.
[6] M. Sato, Y. Nakamura, S. Ishii, Reinforcement learning for biped locomotion, in: International Conference on Artificial Neural Networks, 2002, pp. 777–782.
[7] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (1992) 229–256.
[8] H. Kimura, S. Kobayashi, An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value function, Internal Conference on Machine Learning (1998) 278–286.
[9] R.S. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, Advances in Neural Information Processing Systems 12 (2000) 1057–1063.

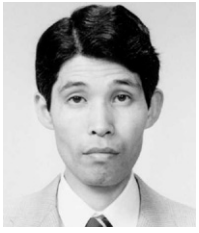
[1] We used the value function based RL in continuous time and state proposed in [16] and also used it for the real robot control in [18].

[10] J. Baxter, P.L. Bartlett, Infinite-horizon policy-gradient estimation, Journal of Artificial Intelligence Research 15 (2001) 319–350.
[11] V. Konda, J. Tsitsiklis, On actor-critic algorithms, Society for Industrial and Applied Mathematics 42 (4) (2003) 1143–1166.
[12] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.
[13] S. Singh, T. Jaakkola, M. Jordan, Learning without state-estimation in partially observable markovian decision processes, in: In Machine Learning: Proceedings of the Eleventh International Conference, 1994, pp. 284–292.
[14] H. Kimura, T. Yamashita, S. Kobayashi, Reinforcement learning of walking behavior for a four-legged robot, in: Proceedings of the IEEE Conference on Decision and Control, 2001, pp. 411–416.
[15] R. Tedrake, T.W. Zhang, H.S. Seung, Stochastic policy gradient reinforcement learning on a simple 3D biped, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 2004, pp. 2849–2854.
[16] K. Doya, Reinforcement learning in continuous time and space, Neural Computation 12 (2000) 219–245.
[17] J. Morimoto, G. Zeglin, C. Atkeson, Minimax differential dynamic programming: Application to a biped walking robot, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, pp. 1927–1932.
[18] J. Morimoto, K. Doya, Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning, Robotics and Autonomous Systems 36 (2001) 37–51.
[19] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, H. Inoue, Autobalancer: An online dynamic balance compensation scheme for humanoid robots, in: B.R. Donald, K. Lynch, D. Rus (Eds.), Algorithmic and Computational Robotics: New Directions, A K Peters, Ltd., 2001, pp. 329–340.
[20] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, A fast dynamically equilibrated walking trajectory generation method of humanoid robot, Autonomous Robots 12 (2002) 71–82.
[21] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, The development of Honda humanoid robot, in: IEEE International Conference on Robotics and Automation, 1998, pp. 1321–1326.
[22] S.H. Strogatz, Nonlinear Dynamics and Chaos, Westview press, 1994.
[23] C. Tuchiya, H. Kimura, S. Kobayashi, Policy learning by ga using importance sampling, in: The 8th Conference on Intelligent Autonomous Systems, 2004, pp. 281–290.
[24] D. Aberdeen, J. Baxter, Scalable internal-state policy-gradient methods for pomdps, in: ICML, 2002, pp. 3–10.
[25] G. Tesauro, Td-gammon, A self teaching backgammon program, achieves master legel play, Neural Computation 6 (1994) 215–219.
[26] M.J. Mataric, Reward functions for accelerated learning, in: Machine Learning: Proceedings of the Eleventh International Conference, 1994, pp. 181–189.

Takamitsu Matsubara received his B.E. in electrical and electronic systems engineering from Osaka Prefecture University, Osaka, Japan, in 2002, his M.E. in information science from Nara Institute of Science and Technology, Nara, Japan, in 2004. He is currently a Ph.D. student in the Department of Information Science at Nara Institute of Science and Technology, Nara, Japan. His research interests include machine learning and robotics.

Jun Morimoto received his B.E. in computer-controlled mechanical systems from Osaka University, Osaka, Japan, in 1996, his M.E. in information science from Nara Institute of Science and Technology, Nara, Japan in 1998, and his Ph.D. in information science from Nara Institute of Science and Technology, Nara, Japan, in 2001. He was Research Assistant at Kawato Dynamic Brain Project, ERATO, JST, from 1999 to 2001. He was a postdoctoral fellow at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, from 2001 to 2002. He is currently a researcher at ATR Computational Neuroscience Laboratories, Kyoto, Japan, and with Computational Brain Project, ICORP, Japan Science and Technology Agency. He is a member of Japanese Neural Network Society, and Robotics Society of Japan. His research interests include reinforcement learning and robotics.

Jun Nakanishi received the B.E. and M.E. degrees both in mechanical engineering from Nagoya University, Nagoya, Japan, in 1995 and 1997, respectively. He received the Ph.D. degree in engineering from Nagoya University in 2000. He also studied in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, USA, from 1995 to 1996. He was a Research Associate at the Department of Micro System Engineering, Nagoya University, from 2000 to 2001, and was a presidential postdoctoral fellow at the Computer Science Department, University of Southern California, Los Angeles, USA, from 2001 to 2002. He joined ATR Human Information Science Laboratories, Kyoto, Japan, in 2002. He is currently a researcher at ATR Computational Neuroscience Laboratories and with the Computational Brain Project, ICORP, Japan Science and Technology Agency. His research interests include motor learning and control in robotic systems. He received the IEEE ICRA 2002 Best Paper Award.



Masa-aki Sato is a department head of Computational Brain Imaging group at ATR Computational Neuroscience Laboratories. He received his Ph.D. in 1980 from Osaka University, studying high energy physics. His current research interests include Bayesian estimation, noninvasive brain imaging, neural networks and learning of dynamical systems.

Kenji Doya took his B.S. in 1984, M.S. in 1986, and Ph.D. in 1991 at the University of Tokyo. He became a research associate at the University of Tokyo in 1986, University of California San Diego in 1991, and Salk Institute in 1993. He joined ATR in 1994 and is currently the head of Computational Neurobiology Department, ATR Computational Neuroscience Laboratories. In 2004, he was appointed as a principal investigator of Initial Research Project, Okinawa Institute of Science and Technology. He is interested in understanding the functions of basal ganglia and neuromodulators based on the theory of reinforcement learning. Contact: doya@oist.jp, 12-22 Suzaki, Uruma, Okinawa 904-2234, Japan.