Learning CPG Sensory Feedback with Policy Gradient for Biped Locomotion for a Full-body Humanoid

Gen Endo[*†], **Jun Morimoto**[†‡], **Takamitsu Matsubara**[†§], **Jun Nakanishi**[†‡] and **Gordon Cheng**[†‡]

[*]Sony Intelligence Dynamics Laboratories, Inc., 3-14-13 Higashigotanda, Shinagawa-ku, Tokyo, 141-0022, Japan

[†]ATR Computational Neuroscience Laboratories, [‡]Computational Brain Project, ICORP, Japan Science and Technology Agency
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan

[§]Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma-shi, Nara, 630-0192, Japan
gendo@idl.sony.co.jp, {xmorimo, takam-m, jun, gordon}@atr.jp

Abstract

This paper describes a learning framework for a central pattern generator based biped locomotion controller using a policy gradient method. Our goals in this study are to achieve biped walking with a 3D hardware humanoid, and to develop an efficient learning algorithm with CPG by reducing the dimensionality of the state space used for learning. We demonstrate that an appropriate feedback controller can be acquired within a thousand trials by numerical simulations and the obtained controller in numerical simulation achieves stable walking with a physical robot in the real world. Numerical simulations and hardware experiments evaluated walking velocity and stability. Furthermore, we present the possibility of an additional online learning using a hardware robot to improve the controller within 200 iterations.

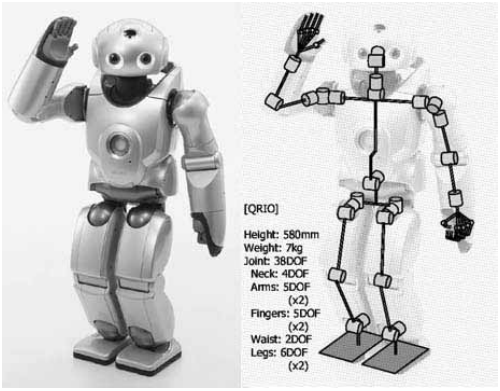

Figure 1: Entertainment Robot QRIO (SDR-4X II)

Introduction

Humanoid research and development has made remarkable progress during the past 10 years. Most of presented humanoids utilize a pre-planned nominal trajectory designed in a known environment. Despite our best effort, it seems that we cannot consider every possible situation in advance when designing a controller. Thus learning capability to acquire or improve a walking pattern is essential for broad range of humanoid application in an unknown environment.

Our goals in this paper are to acquire a successful walking pattern through learning and to achieve walking with a hardware 3D full-body humanoid robot (Fig. 1). While many attempts have been made to investigate learning algorithms for simulated biped walking, there are only a few successful implementation on real hardware, for example (Benbrahim & Franklin 1997; Tedrake, Zhang, & Seung 2004). To the best of our knowledge, Tedrake et al. (Tedrake, Zhang, & Seung 2004) is the only example of an implementation of learning algorithm on a 3D hardware robot. They implemented a learning algorithm on a simple physical 3D biped robot possessing basic properties of passive dynamic walk, and successfully obtained appropriate feedback controller for ankle roll joints via online learning. With the help of their specific mechanical design to embed an intrinsic walking pattern with the passive dynamics, the state space for learning was drastically reduced from 18 to 2 in spite of the complexity of the 3D biped model, which usually suffers from dimensionality explosion. From a learning point of view, the dimensionality reduction is an important issue in practice. However, developing a specific humanoid hardware with uni-functionality, for example walking, may lose an important feature of humanoid robot such as versatility and capability of achieving various tasks.

Therefore, instead of gait implementation by mechanical design, we introduce the idea of using a Central Pattern Generator (CPG), which has been hypothesized to exist in the central nervous system of animals (Cohen 2003). It is demonstrated that the CPG can generate a robust biped walking pattern with appropriate feedback signals by using entrainment property even in an unpredictable environment in numerical simulations (Taga 1995). However, designing appropriate feedback pathways of neural oscillators often requires much effort to manually tune the parameters of the oscillator. Thus, a genetic algorithm (Hase & Yamazaki 1998) and reinforcement learning (Mori *et al.* 2004) were applied to optimize the open parameters of the CPG for biped locomotion. However, these methods often require a large number of iteration to obtain a solution due to the large dimensionality of the state space used for optimization. In this paper, our primary goals are to achieve biped walking with

learning for a 3D full-body humanoid hardware, which is not designed for a specific walking purpose, and to develop an efficient learning algorithm which can be implemented on a hardware robot for an additional online learning to improve a controller. In a physical robot, we can not accurately observe all states of the system due to limited number of equipped sensors and measurement noise in practice. Thus, we find it natural to postulate the learning problem as a partially observable Markov decision problem (POMDP). In the proposed learning system, we use a policy gradient method which can be applied to POMDP (Kimura & Kobayashi 1998). In POMDP, it is generally known that a large amount of iteration would be required for learning compared with learning in MDP because lack of information yields large variance of the estimated gradient of expected reward with respect to the policy parameters (Sutton *et al.* 2000; Konda & Tsitsiklis 2003). However, in the proposed framework, when the CPG and the mechanical system of the robot converge to a periodic trajectory due to the entrainment property, the internal states of the CPG and the states of the robot will be synchronized. Thus, by using the state space only composed of the observable reduced number of states, efficient learning can be possible to achieve steady periodic biped locomotion even in the POMDP.

CPG Control Architecture

In our initial work, we explored CPG-based control with the policy gradient method for a planar biped model (Matsubara *et al.* 2005), suggesting that the policy gradient method is a promising way to acquire biped locomotion within a reasonable numbers of iteration. In this section, we extend our previous work to a 3D full-body humanoid robot, QRIO (Ishida, Kuroki, & Yamaguchi 2003; Ishida & Kuroki 2004).

Neural Oscillator Model

We use the neural oscillator model proposed by Matsuoka (Matsuoka 1985), which is widely used as a CPG in robotic applications (Kimura, Fukuoka, & Cohen 2003; Williamson 1998):

$$\tau_{CPG}\dot{z}_j = -z_j - \sum_{j=1}^{6} w_{ij}q_j - \gamma z'_j + c + a_j, \quad (1)$$

$$\tau'_{CPG}\dot{z'}_j = -z_j + q_j, \quad (2)$$

$$q_j = \max(0, z_j), \quad (3)$$

where i is the index of the neurons, τ_{CPG}, τ'_{CPG} are time constants for the internal states z_j and z'_j. c is a bias and γ is an adaptation constant. w_{ij} is a inhibitory synaptic weight from the j-th neuron to the i-th neuron. q_j is an output of each neural unit and a_j is a feedback signal which will be defined in the following section.

CPG Arrangement

In many of the previous applications of neural oscillator based locomotion studies, an oscillator is allocated at each joint and its output is used as a joint torque command to the robot (Taga 1995). However, it is difficult to obtain appropriate feedback pathways for all the oscillators to achieve the desired behavior with the increase of the number of degrees of freedom of the robot because neural oscillators are intrinsically nonlinear. Moreover, precise torque control of each joints is also difficult to be realized for a hardware robot in practice. In this paper, to simplify the problem, we propose a new oscillator arrangement with respect to the position of the tip of the leg in the Cartesian coordinate system, which is reasonably considered as the task coordinates for walking. We allocate only 6 neural units exploiting symmetry of the walking pattern between the legs. We decompose overall walking motion into stepping motion in place produced in the frontal plane and propulsive motion generated in the sagittal plane.

Fig. 2 illustrates the proposed neural arrangement for the stepping motion in place in the frontal plane. We employ two neurons to form a coupled oscillator connected by a mutual inhibition ($w_{12} = w_{21} = 2.0$) and allocate it to control the position of both legs p^l_z, p^r_z along the Z (vertical) direction in a symmetrical manner with $\pi\ rad$ phase difference:

$$p^l_z = Z_0 - A_z\,(q_1 - q_2), \quad (4)$$

$$p^r_z = Z_0 + A_z\,(q_1 - q_2), \quad (5)$$

where Z_0 is a position offset and A_z is the amplitude scaling factor.

For a propulsive motion in the sagittal plane, we introduce a quad-element neural oscillator to produce coordinated leg movements with stepping motion based on the following intuition: as illustrated in Fig. 3, when the robot is walking forward, the leg trajectory with respect to the body coordinates in the sagittal plane can be roughly approximated by the shape of an ellipsoid. Suppose the output trajectories of the oscillators can be approximated as $p^l_x = A_x\cos(\omega t + \alpha_x)$ and $p^l_z = A_z\cos(\omega t + \alpha_z)$, respectively. Then, to form the ellipsoidal trajectory on the X-Z plane, p^l_x and p^l_z need to satisfy the relationship $p^l_x = A_x\cos\phi$ and $p^l_z = A_z\sin\phi$, where ϕ is the angle defined in Fig. 3. Thus, the desired phase difference between vertical and horizontal oscillation

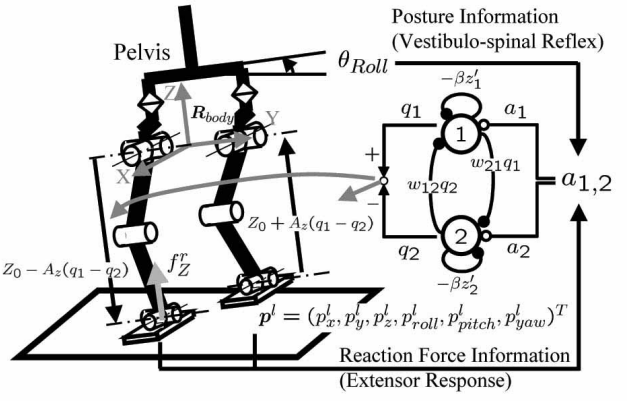

Figure 2: Neural oscillator allocation and biologically inspired feedback pathways for a stepping motion in place

should be $\alpha_x - \alpha_z = \pi/2$. To embed this phase difference as an intrinsic property, we install a quad-element neural oscillator with uni-directional circular inhibitions ($w_{34} = w_{43} = w_{56} = w_{65} = 2.0$, $w_{35} = w_{63} = w_{46} = w_{54} = 0.5$). It generates inherent phase difference of $\pi/2$ between two coupled oscillators, $(q_3 - q_4)$ and $(q_5 - q_6)$ (Matsuoka 1985). Therefore, if $(q_3 - q_4)$ is entrained to the vertical leg movements, then an appropriate horizontal oscillation with desired phase difference is achieved by $(q_5 - q_6)$.

Similar to the Z direction, the neural output $(q_5 - q_6)$ is allocated to control the position of both legs p_x^l, p_x^r along the X (forward) direction in the sagittal plane:

$$p_x^l = X_0 - A_x\,(q_5 - q_6), \quad (6)$$
$$p_x^r = X_0 + A_x\,(q_5 - q_6), \quad (7)$$

where X_0 is an offset and A_x is the amplitude scaling factor.

This framework provides us with a basic walking pattern which can be modulated by feedback signals a_j.

Sensory Feedback

With an assumption of symmetric leg movements, feedback signals to the oscillator eqn.(1) can be expressed as

$$a_{2m} = -a_{2m-1}, \;\; (m = 1, 2, 3). \quad (8)$$

As a first step of this study, we focus on acquiring a feedback controller (a_5) for the propulsive leg movement in the X direction. Thus, we explicitly design sensory feedback pathways for stepping motion in place (a_1) motivated by biological observations (Fig.2). We introduce *Extensor Response* and *Vestibulo-spinal Reflex* (Kimura, Fukuoka, & Cohen 2003). Both sensory feedback pathways generate recovery motion by adjusting leg length according to the vertical reaction force or the inclination of the body.

$$a_1 = h_{ER}\,(f_z^r - f_z^l)\,/\,mg \;+\; h_{VSR}\,\theta_{roll}, \quad (9)$$

where $(f_z^r - f_z^l)$ are right/left vertical reaction force differences normalized by total body weight mg. h_{ER}, h_{VSR} are scaling factors. Stepping motion robustness against perturbation was experimentally discussed in (Endo *et al.* 2005) and we chose $h_{ER} = 0.4, h_{VSR} = 1.8$.

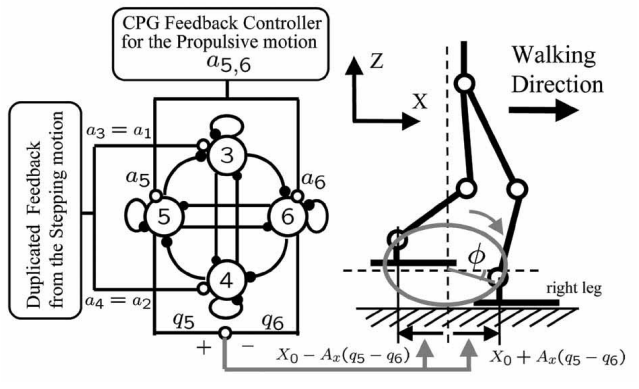

Figure 3: A quad-element neural oscillator for a propulsive motion in the sagittal plane

The same feedback signal is fed back to a quad-element neural oscillator for the propulsive motion, $a_3 = a_1$, to induce a cooperative leg movement with $\pi/2$ phase difference between the Z and X direction (Fig.3).

For the propulsive leg movements in the X direction, the feedback signal a_j are represented by a policy gradient method.

$$a_j(t) = a_j^{\max} g(v_j(t)), \quad (10)$$

where $g(x) = \frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)$, v_j is sampled from a stochastic policy defined by a probability distribution $\pi_{\mathbf{w}}(\mathbf{x}, v_j) = P(v_j|\mathbf{x}; \mathbf{w}^\mu, \mathbf{w}^\sigma)$:

$$\pi_{\mathbf{w}}(\mathbf{x}, v_j) = \frac{1}{\sqrt{2\pi}\sigma_j(\mathbf{w}^\sigma)}\exp\left(\frac{-(v_j - \mu_j(\mathbf{x}; \mathbf{w}^\mu))^2}{2\sigma_j^2(\mathbf{w}^\sigma)}\right), \quad (11)$$

where $\mathbf{x}$ denotes the state of the robot, and $\mathbf{w}^\mu$,$\mathbf{w}^\sigma$ are parameter vectors of the policy. We can equivalently represent v_j by

$$v_j(t) = \mu_j(\mathbf{x}(t); \mathbf{w}^\mu) + \sigma_j(\mathbf{w}^\sigma) n_j(t), \quad (12)$$

where, $n_j(t) \sim N(0, 1)$. $N(0, 1)$ is a normal distribution which has a mean μ of 0 and a variance σ^2 of 1. In the next section, we discuss the learning algorithm to acquire a feedback controller $a_{j=5}$ for the propulsive motion.

Learning the sensory feedback controller

Previous studies have demonstrated promising results to apply a policy gradient method for POMDP on biped walking tasks(Tedrake, Zhang, & Seung 2004; Matsubara *et al.* 2005). We use it for acquisition of a policy of the sensory feedback controller to the neural oscillator model.

To describe the representative motion of the robot, we consider the states of the pelvis of the robot where gyro sensors are located, which also roughly approximates the location of the center of mass (COM) of the system. We chose the pelvis angular velocity $\mathbf{x} = (\dot{\theta}_{roll}, \dot{\theta}_{pitch})^T$ as the input states to the learning algorithm. Therefore, we can conceive that the rest of the state of the pelvis such as inclinations with respect to the world coordinates as well as the internal variables in the neural oscillator model in equations (1)(2) are hidden variables for the learning algorithm. We show that we can acquire a policy of the sensory feedback $a_j(\mathbf{x})$ in (10) by using a policy gradient method even with considerable numbers of hidden variables.

In the following section, we introduce the definition of the value function in continuous time and space to derive the temporal difference error (TD error)(Doya 2000). Then we discuss the learning method of a policy for the sensory feedback controller.

Learning the value function

Consider the continuous-time and continuous-states system

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{u}(t)), \quad (13)$$

where $\mathbf{x} \in X \subset \Re^n$ is the state and $\mathbf{u} \in U \subset \Re^m$ is the control input. We denote the immediate reward for the state and action as

$$r(t) = r(\mathbf{x}(t), \mathbf{u}(t)). \quad (14)$$

The value function of state $\mathbf{x}(t)$ based on a policy $\pi(\mathbf{u}(t) \mid \mathbf{x}(t))$ is defined as

$$V^{\pi}(\mathbf{x}(t)) = E\left\{ \int_t^{\infty} e^{-\frac{s-t}{\tau}} r(\mathbf{x}(s), \mathbf{u}(s)) ds \Big| \pi \right\}, \quad (15)$$

where τ is a time constant for discounting future rewards. A consistency condition for the value function is given by differentiating the definition (15) by t as

$$\frac{dV^{\pi}(\mathbf{x}(t))}{dt} = \frac{1}{\tau} V^{\pi}(\mathbf{x}(t)) - r(t). \quad (16)$$

We denote the current estimate of the value function as $V(\mathbf{x}(t)) = V(\mathbf{x}(t); \mathbf{w}^c)$, where $\mathbf{w}^c$ is the parameter of the function approximator. If the current estimate $\hat{V}$ of the value function is perfect, it should satisfy the consistency condition (16). If this condition is not satisfied, the prediction should be adjusted to decrease the inconsistency,

$$\delta(t) = r(t) - \frac{1}{\tau} V(t) + \dot{V}(t). \quad (17)$$

This is the continuous-time counterpart of TD error (Doya 2000). The update laws for the parameter of the policy w_i^c and the eligibility trace e_c^i of w_i^c are defined respectively as

$$\dot{e}_i^c(t) = -\frac{1}{\kappa^c} e_i^c(t) + \frac{\partial V_{\mathbf{w}^c}}{\partial w_i^c}, \quad (18)$$

$$\dot{w}_i^c(t) = \alpha \delta(t) e_i^c(t), \quad (19)$$

where α is the learning rate and κ^c is the time constant of the eligibility trace. In this study, we select the learning parameters as $\tau = 1.0$, $\alpha = 78$, $\kappa^c = 0.5$.

Learning a policy of the sensory feedback controller

We can estimate the gradient of the expected total reward $V(t)$ with respect to the policy parameter w^a:

$$\frac{\partial}{\partial w_i^a} E\{ V(t) \mid \pi_{\mathbf{w}^a} \} = E\{\delta(t) e_i^a(t)\}, \quad (20)$$

where w_i^a is the parameter of policy $\pi_{\mathbf{w}}$ and $e_i^a(t)$ is the eligibility trace of the parameter w_i^a. The update law for the parameter of the policy w_i^a and the eligibility trace $e_i^a(t)$ are derived respectively as

$$\dot{e}_i^a(t) = -\frac{1}{\kappa^a} e_i^a(t) + \frac{\partial \ln \pi_{\mathbf{w}^a}}{\partial w_i^a}, \quad (21)$$

$$\dot{w}_i^a(t) = \beta \delta(t) e_i^a(t), \quad (22)$$

where β is the learning rate and κ^a is the time constant of the eligibility trace. The definition (20) implies that by using $\delta(t)$ and $e_i^a(t)$, we can calculate the unbiased estimator of the gradient of the value function with respect to the parameter w_i^a. We set the learning parameters as $\beta^{\mu} = \beta^{\sigma} = 195$, $\kappa^{\mu} = 1.0$, $\kappa^{\sigma} = 0.1$.

Numerical Simulation Setup

Function Approximator for the Value Function and the Policy

We use a normalized Gaussian network (NGnet) (Doya 2000) to model the value function and the mean of the policy. The variance of the policy is modelled by a sigmoidal function (Kimura & Kobayashi 1998; Peters, Vijayakumar, & Schaal 2003). The value function is represented by the NGnet:

$$V(\mathbf{x}; \mathbf{w}^c) = \sum_{k=1}^{K} w_k^c b_k(\mathbf{x}), \quad (23)$$

where

$$b_k(\mathbf{x}) = \frac{\phi_k(\mathbf{x})}{\sum_{l=1}^{K} \phi_l(\mathbf{x})}, \phi_k(\mathbf{x}) = e^{-\|\mathbf{s}_k^T(\mathbf{x}-\mathbf{c}_k)\|}, \quad (24)$$

k is the number of the basis functions. The vectors $\mathbf{c}_k$ and $\mathbf{s}_k$ characterize the center and the size of the k-th basis function, respectively. The mean μ and the variance σ of the policy are represented by the NGnet and the sigmoidal function:

$$\mu_j = \sum_{i=1}^{K} w_{ij}^{\mu} b_i(\mathbf{x}), \quad (25)$$

and

$$\sigma_j = \frac{1}{1 + \exp(-w_j^{\sigma})}, \quad (26)$$

respectively. We locate basis functions $\phi_k(\mathbf{x})$ at even intervals in each dimension of the input space ($-2.0 \leq \dot{\theta}_{roll}, \dot{\theta}_{pitch} \leq 2.0$). We used 225($= 15 \times 15$) basis functions for approximating the value function and the policy respectively.

Rewards

We used a reward function:

$$r(\mathbf{x}) = k_H (h_1 - h') + k_S v_x, \quad (27)$$

where h_1 is the pelvis height of the robot, h' is a threshold parameter for h_1 and v_x is forward velocity with respect to the ground. The reward function is designed to keep the height of the pelvis by the first term, and at the same time to achieve forward progress by the second term. In this study, the parameters are chosen as $k_S = 0.0 - 10.0$, $k_H = 10.0$, $h' = 0.272$. The robot receives a punishment (negative reward) $r = -1$ if it falls over.

Experiments

Simulation Results and Hardware Verifications

The learning sequence is as follows: at the beginning of each trial, we utilize a hand-designed feedback controller to initiate walking gait for several steps. Then we switch the feedback controller for the learning algorithm at random in order to generate various initial state input. Each trial is terminated in the case where the immediate reward is less than -1 or the robot walks for 20 sec. The learning process is considered as success when the robot does not fall over in 20 successive trials.

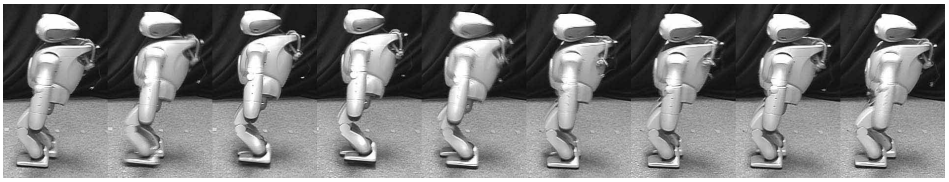

Figure 6: Snapshots of straight steady walking with acquired feedback controller ($A_x = 0.015\,m$, $A_z = 0.005\,m$, $v_x = 0.077\,m/s$. Photos were captured every 0.1 $sec.$)

k_s	Num. of Exp.	Num. of Achievements Sim.(trials)		Hard.(trials)	
0.0	4	1	(2385)	0	(-)
1.0	3	3	(528)	3	(1600)
2.0	3	3	(195)	1	(800)
3.0	4	4	(464)	2	(1500)
4.0	3	2	(192)	2	(350)
5.0	5	2	(1011)	1	(600)
10.0	5	5	(95)	5	(460)
Sum(Ave.)	27	20	(696)	14	(885)

Table 1: Achievements of acquisition

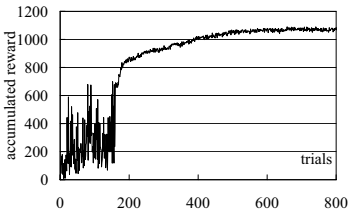

Figure 4: Typical learning curve

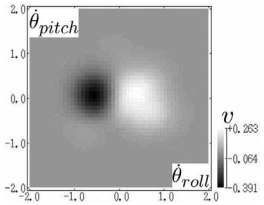

Figure 5: Learned policy

At the beginning of a learning process, the robot immediately fell over within a few steps as expected. The policy gradually increased the output amplitude of feedback controller to improve walking motion as the learning proceeded. We did 27 experiments with various velocity reward, k_s, and walking motion was successfully acquired in 20 experiments (Table. 1). Typically, it took 20 hours to run one simulation for 1000 trials and the policy was acquired after 696 trials on average. Fig. 4 and Fig. 5 show a typical example of accumulated reward at each trial and an acquired policy, respectively. In Fig.5, while $\dot{\theta}_{roll}$ dominates the policy output, $\dot{\theta}_{pitch}$ does not have much influence. The reason would be that $\dot{\theta}_{roll}$ is always generating by stepping motion in the Z direction regardless of the propulsive motion, thus the policy tries to utilize $\dot{\theta}_{roll}$ to generate cooperative leg movements. On the other hand, $\dot{\theta}_{pitch}$ is suppressed by the reward function not to decrease the pelvis height by a pitching oscillation which usually cause falling.

We implemented the acquired 20 feedback controllers in Table. 1 on the hardware and confirmed that 14 of them successfully achieved steady walking on the carpet floor with slight undulation. Fig. 6 shows snapshots of acquired walking pattern. Also, we carried out walking experiments on a slope and the acquired policy achieved steady walking in the range of +3 to -4 deg inclination, suggesting enough walking stability.

We also verified improvements of the policy by implementing the policies with different trials with the same learning process. With the policy on the early stage of a learning process, the robot exhibited back and forth stepping then immediately fell over. With the policy on the intermediate stage, the robot performed unsteady forward walking and occasional stepping on the spot. With policy after substantial trials, the robot finally achieved steady walking.

On average, additional 189 trials in the numerical simulation were required for the policy to achieve walking in the physical environment. This result suggests the learning process successfully improves robustness against perturbation by using entrainment property.

Velocity Control

To control walking velocity, we investigated the relationship between the reward function and the acquired velocity. We set the parameters in eqn.(1)(2) as $\tau_{CPG} = 0.105$, $\tau'_{CPG} = 0.132$, $c = 2.08$, $\gamma = 2.5$ to generate an inherent oscillation where amplitude and period are 1.0 and 0.8, respectively. Since we set A_x=0.015m, expected walking velocity with intrinsic oscillation is 0.075m/s.

We measured average walking velocity both in numerical simulations and hardware experiments with various k_s in the 0.0 to 5.0 range (Fig. 7). The resultant walking velocity in the simulation increased as we increased k_s and hardware experiments demonstrated similar tendency.

This result shows the reward function works appropriately to obtain a desirable feedback policy, which is difficult for a hand-designed controller to achieve. Also, it would be possible to acquire different feedback controllers with some other criteria such as energy efficiency or walking direction by using the same scheme.

Stability Analysis

To quantify the stability of an acquired walking controller, we consider the periodic walking motion as discrete dynamics and analyze the local stability around a fix point using a return map. We perturbed target trajectory on $(q_5 - q_6)$ to change step length at random timing during steady walking, and captured the states of the robot when left leg touched down. We measured two steps, right after the perturbation ($\mathbf{x}_n$) and the next step ($\mathbf{x}_{n+1}$). If acquired walking motion is

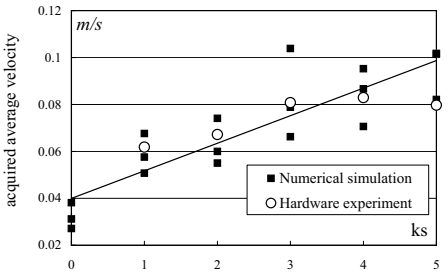

Figure 7: The relationship between acquired velocity and velocity reward

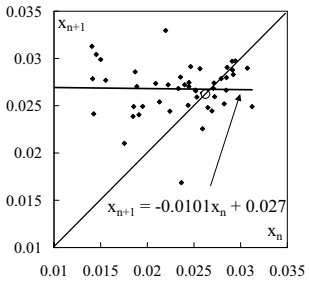

Figure 8: Return map of step length

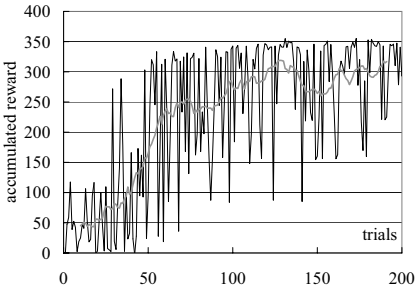

Figure 9: An example of additional online learning using the physical robot.

locally stable, absolute eigenvalue of the return map should be less than 1.

Fig.8 shows the return map with 50 data points and a white dot indicates a fix point derived by averaging 100 steps without perturbation. The estimated eigenvalue is -0.0101 calculated by a least squares fit. The results suggests that even if step length was reduced to half of the nominal step length by perturbation, for example pushed forward, the feedback controller quickly converges to the steady walking pattern within one step.

Additional online learning

As shown in Table. 1, 6 policies obtained in numerical simulations could not achieve walking with the physical robot in the real environment. Even in this case, we could make online additional learning using a real hardware because a calculation cost of the numerical simulation is mainly due to the dynamics calculation, not to the learning algorithm itself. In this section, we attempt to improve the obtained policies in numerical simulations which could not originally produce steady walking in the hardware experiments.

For the reward function, walking velocity was calculated by the relative velocity of the stance leg with respect to the pelvis and the body height was measured by the the joint angles of the stance leg and the absolute body inclination derived from integration of gyration sensor. We introduced digital filters to cut off the measurement noise. Note that we did not introduce any external sensor for this experiment.

Despite delayed and inaccurate reward information, the online learning algorithm successfully improved the initial policy and performed steady walking within 200 trials (which took us 2.5 hours to do). Fig. 9 shows an example of online additional learning where $k_s = 4.0$, $k_h = 10.0$ and $h' = 0.275$. (Note that the value of the accumulated reward differs from the simulated result in Fig. 4 due to different time duration $16sec$ for one trial.) The gray line indicates running average of accumulated reward for 20 trials.

Discussion

In this section, we would like to discuss our motivation of the proposed framework.

Most of the existing humanoids utilize a pre-planned nominal trajectory and requires precise modeling and precise joint actuation with high joint control gains to track the nominal trajectory in order to accomplish successful locomotion. However, it is not possible to design the nominal trajectory for every possible situation in advance. Thus, these model-based approaches may not be desirable in an unpredictable or dynamically changing environment.

Learning can be one of the alternative approaches, as it has the potential capability of adapting to environmental changes and modeling error. Moreover, learning may provide us with an efficient way of producing various walking patterns by simply introducing different higher level criteria such as walking velocity and energy efficiency without changing the learning framework itself.

However, humanoid robots have many degrees of freedom, we cannot directly apply conventional reinforcement learning methods to the robots. To cope with this large-scale problem, Tedrake et al. exploited inherent passive dynamics. The question is whether we can easily extend their approach to a general humanoid robot. They used the desired state on the return map taken from the gait of the robot walking down on a slope without actuation in order to define the reward function for reinforcement learning. Their learning algorithm owes much to the intrinsic passive dynamical characteristics of the robot that can walk down a slope without actuation. However, their approach cannot be directly applied to general humanoid robots that are not mechanically designed with specific dynamical characteristics for only walking.

Therefore, in this study, instead of making use of passive dynamics of the system, we proposed to use CPGs to generate basic periodic motions and reduce the number of dimensions of the state space for the learning system exploiting entrainment of neural oscillators for biped locomotion for a full-body humanoid robot. As a result, we only considered 2-dimensional state space for learning. This is a drastic reduction of the state space from the total 36 dimensions including the robot and CPG dynamics if we would treat this as an MDP assuming that all the states are measurable.

Since we only considered low-dimensional state space, we had to treat our learning problem as a POMDP. Although it is computationally infeasible to calculate the optimal value function for the POMDP by using dynamic programming in high-dimensional state space (Kaelbling, Littman, & Cassandra 1998), we can still attempt to find a locally optimal policy by using policy gradient methods (Kimura &

Kobayashi 1998; Jaakkola, Singh, & Jordan 1995; Sutton *et al.* 2000). We succeeded to acquire the biped walking controller by using the policy gradient method with the CPG controller within a feasible numbers of trials. One of important factors to have the successful result was selection of the input states. Inappropriate selection of the input states may cause large variance of the gradient estimation and may lead to the large number of trials. Though we manually selected $\dot{\theta}_{roll}$ and $\dot{\theta}_{pitch}$ as the input states for our learning system, development of a method to automatically select input states forms part of our future work.

Conclusion

In this paper, we proposed an efficient learning framework for CPG-based biped locomotion controller using the policy gradient method. We decomposed a walking motion into a stepping motion in place, and propulsive motion and feedback pathways for the propulsive motion were acquired using the policy gradient method. Despite considerable number of hidden variables, the proposed framework successfully obtained a walking pattern within 1000 trials on average in the simulator. Acquired feedback controllers were implemented on a 3D hardware robot and demonstrated robust walking in the physical environment. We discuss velocity control and stability as well as a possibility of online additional learning with the hardware robot.

We plan to investigate energy efficiency and robustness against perturbation using the same scheme as well as the acquisition of the policy for stepping motion. Also, we plan to implement the proposed framework on a different platform to demonstrate sufficient capability to handle configuration changes. We will introduce additional neurons to generate more natural walking.

Acknowledgements

We would like to thank Seiichi Miyakoshi of the Digital Human Research Center, AIST, Japan and Kenji Doya of Initial Research Project, Okinawa Institute of Science and Technology, for productive and invaluable discussions. We would like to thank all the persons concerned in QRIO's development for supporting this research.

References

Benbrahim, H., and Franklin, J. A. 1997. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems* 22:283–302.

Cohen, A. H. 2003. Control principle for locomotion – looking toward biology. In *2nd International Symposium on Adaptive Motion of Animals and Machines*. (CD-ROM, TuP-K-1).

Doya, K. 2000. Reinforcement learning in continuous time and space. *Neural Computation* 12:219–245.

Endo, G.; Nakanishi, J.; Morimoto, J.; and Cheng, G. 2005. Experimental studies of a neural oscillator for biped locomotion with QRIO. In *IEEE International Conference on Robotics and Automation*, 598–604.

Hase, K., and Yamazaki, N. 1998. Computer simulation of the ontogeny of biped walking. *Anthropological Science* 106(4):327–347.

Ishida, T., and Kuroki, Y. 2004. Development of sensor system of a small biped entertainment robot. In *IEEE International Conference on Robotics and Automation*, 648–653.

Ishida, T.; Kuroki, Y.; and Yamaguchi, J. 2003. Mechanical system of a small biped entertainment robot. In *IEEE International Conference on Intelligent Robots and Systems*, 1129–1134.

Jaakkola, T.; Singh, S. P.; and Jordan, M. I. 1995. Reinforcement learning algorithm for partially observable Markov decision problems. 7:345–352.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Kimura, H., and Kobayashi, S. 1998. An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value function. In *International Conference on Machine Learning*, 278–286.

Kimura, H.; Fukuoka, Y.; and Cohen, A. H. 2003. Biologically inspired adaptive dynamic walking of a quadruped robot. In *8th International Conference on the Simulation of Adaptive Behavior*, 201–210.

Konda, V., and Tsitsiklis, J. 2003. On actor-critic algorithms. *Society for Industrial and Applied Mathematics* 42(4):1143–1166.

Matsubara, T.; Morimoto, J.; Nakanishi, J.; and Doya, K. 2005. Learning feedback pathways in CPG with policy gradient for biped locomotion. In *IEEE International Conference on Robotics and Automation*, 4175–4180.

Matsuoka, K. 1985. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics* 52:345–353.

Mori, T.; Nakamura, Y.; Sato, M.; and Ishii, S. 2004. Reinforcement learning for a cpg-driven biped robot. In *Nineteenth National Conference on Artificial Intelligence*, 623–630.

Peters, J.; Vijayakumar, S.; and Schaal, S. 2003. Reinforcement learning for humanoid robotics. In *International Conference on Humanoid Robots*. (CD-ROM).

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with imperfect value function. *Advances in Neural Information Processing Systems* 12:1057–1063.

Taga, G. 1995. A model of the neuro-musculo-skeletal system for human locomotion I. emergence of basic gait. *Biological Cybernetics* 73:97–111.

Tedrake, R.; Zhang, T. W.; and Seung, H. S. 2004. Stochastic policy gradient reinforcement learning on a simple 3D biped. In *International Conference on Intelligent Robots and Systems*, 2849–2854.

Williamson, M. 1998. Neural control of rhythmic arm movements. *Neural Networks* 11(7–8):1379–1394.