Reinforcement Learning for Biped Locomotion

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

[1] ATR, Human Information Science Laboratories, and CREST, JST,
masa-aki@atr.co.jp
[2] Nara Institute of Science and Tecnology, yutak-na@is.aist-nara.ac.jp

Abstract. This paper studies the reinforcement learning (RL) method for central pattern generators (CPG) that generates stable rhythmic movements such as biped locomotion. RL for biped locomotion is very difficult, since the biped robot is highly unstable and the system has continuous state and action spaces with a high degree of freedom. In order to deal with RL for CPG, we propose a new RL method called the CPG-actor-critic method. We applied this method to the RL for the biped robot. The computer simulation showed that our RL method was able to train the CPG such that the biped robot walk stably.

1 Introduction

Biological systems exhibit various kinds of rhythmic movement such as locomotion and swimming. The control mechanism of these movements has been extensively studied both in biological science and engineering. Recent studies of biped locomotion enabled biped 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 current robots. Neurobiological studies revealed that rhythmic motor patterns are controlled by neural oscillators referred to as central pattern generators (CPG) [2]. It has been also suggested that sensory feedback plays an important role in stabilizing rhythmic movements by coordinating the physical system and the CPG. 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 parameter values for various robots and environments, since there is no design principle to determine the CPG parameter values.

The main aim of this paper is to study the reinforcement learning (RL) method for a CPG controller that generates stable rhythmic movements. RL methods have been successfully applied to various Markov decision problems (MDP) with finite state and action spaces [4]. RL for biped locomotion is very difficult, because the biped robot is highly unstable and the system 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 train-

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

ing the CPG, which is a special case of recurrent neural networks. In order to deal with RL for CPG, we propose a new RL method called the CPG-actor-critic method. We applied this method to the biped robot used in [3]. The computer simulation showed that our RL method was able to train the CPG so that the biped robot walk stably.

2 CPG

In this paper, we study reinforcement learning (RL) for robot rhythmic movement using a central pattern generator (CPG), depicted in Fig. 1(a). The equation of motion for a physical system such as for a robot is formally written as

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

where $\mathbf{x}$ and $\dot{\mathbf{x}}$ denotes the physical state and its time derivative, respectively. The control signal (torque) from the CPG is denoted by $\boldsymbol{\tau}$. $F(\mathbf{x}, \boldsymbol{\tau})$ represents the vector field of the system dynamics. The equation of motion for a CPG, which is a special case of recurrent neural networks, is given by

$$\begin{aligned} c_i \dot{\nu}_i &= -\nu_i + I_i, \quad y_i = G_i(\nu_i), \quad (2) \\ I_i &= \sum_j W_{ij} y_j + I_i^{bias} + I_i^{ext}, \\ I_i^{ext} &= \sum_k A_{ik} S_k, \end{aligned}$$

where ν_i, y_i and c_i represent the i-th neuron state variable, its output and its time constant, respectively. I_i, I_i^{bias} and I_i^{ext} represent the total input, the bias input and the external input to the i-th neuron, respectively. The external input I_i^{ext} is a weighted sum of the sensory feedback signal S_k with the connection weight A_{ik}. The output function $G_i(\cdot)$ is assumed to be a sigmoidal function or some threshold function. The connection weight from the j-th neuron to the i-th neuron is denoted by W_{ij}. The control signal to the physical system is given by a weighted sum of the CPG neuron output:

$$\tau_\alpha = \sum_i T_{\alpha i} y_i, \tag{3}$$

where τ_α represents the α-th control signal and $T_{\alpha i}$ represents the weight factor.

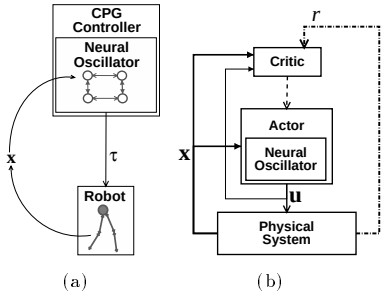

Fig. 1. Actor-critic method.

3 CPG-actor-critic

When we try to apply the actor-critic method to the CPG controller system, there are several difficulties. In this method, (Fig. 1(b)), the CPG controller becomes the actor. Since the CPG output depends on its current internal state, the future reward also depends on the internal state of the CPG. Therefore, the reinforcement learning task in this method becomes a partially observable Markov decision problem (POMDP), which is much more difficult than a MDP, even if the physical system state is fully observable. Another source of difficulty comes from the fact that the CPG is a recurrent

neural network. The standard actor-critic algorithm is not suited for training recurrent neural networks.

In order to overcome these difficulties, we propose a new RL method that is called the CPG-actor-critic method (Fig. 2(b)). In this method, the CPG is divided into two modules, i.e., the basic CPG and the actor. Correspondingly, input to the CPG neuron, I_i, is divided into two parts:

$$I_i = I_i^{fix} + u_i, \tag{4}$$

$$I_i^{fix} = \sum_j W_{ij}^{fix} y_j + I_i^{bias},$$

$$u_i = \sum_j W_{ij}^{act} y_j + \sum_k A_{ik} S_k. \tag{5}$$

The basic CPG is defined by the fixed connection weight W_{ij}^{fix} and receives the input, $\mathbf{u}$, from the actor. The actor in this method is a linear controller and receives the basic CPG neuron output $\mathbf{y}$ and the physical system feedback signal $\mathbf{S}$. The actor calculates its output to the basic CPG, $\mathbf{u}$, by (5). This CPG-actor-critic method has dual aspects. From the control viewpoint, the CPG controller consists of the basic CPG and the actor (i.e., $W_{ij} = W_{ij}^{fix} + W_{ij}^{act}$) as shown in Fig. 2(a). From the RL viewpoint, the actor sends the virtual control signal, $\mathbf{u}$, to the virtual system, which consists of the basic CPG and the physical system as shown in Fig. 2(b). Since the actor in this method is a linear controller, training of the actor can be done by using the gradient ascent method described below.

The critic receives the virtual system state, namely, it receives both the physical state and the basic CPG state, and predicts the future reward for the current state. When the physical system state is fully observable, the RL problem in this CPG-actor-critic method becomes a MDP. Consequently, we can avoid the difficulties mentioned earlier.

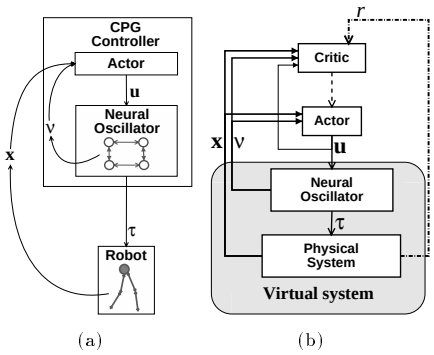

Fig. 2. CPG-actor-critic method.

4 Learning Algorithm

The RL algorithm for the CPG-actor-critic method is explained in this section. For explanatory simplicity, we assume the discrete time notation, that is, it is assumed that the differential equations, (1) and (2), are discretized by an appropriate method.

The current virtual system state at time t is represented by the physical state $\mathbf{x}(t)$ and the basic CPG state $\boldsymbol{\nu}(t)$. The actor receives the basic CPG neuron output $\mathbf{y}(t) = G(\boldsymbol{\nu}(t))$ and the sensory feedback signal $\mathbf{S}(t)$, which is some function of $\mathbf{x}(t)$, from the physical system. The actor calculates the output $\mathbf{u}(t)$ by (5). After receiving the actor output, the virtual system consisting of the basic CPG and the physical system changes its state to $(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1))$ according to the basic CPG and the physical system equations (1)-(4). Then, the critic receives an immediate reward $r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$.

The goal of the RL is to find the optimal actor that maximizes the expected future return defined by

$$V(\boldsymbol{\nu}, \mathbf{x}) \equiv \sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)), \tag{6}$$

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

$$Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u}) = r(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u}) + \gamma V(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1)), \tag{7}$$

where $\boldsymbol{\nu}(t) = \boldsymbol{\nu}$, $\mathbf{x}(t) = \mathbf{x}$ and $\mathbf{u}(t) = \mathbf{u}$ are assumed. The Q-function, $Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})$, indicates the expected future return for the current state and action $(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})$, when the current actor is used for the subsequent states. From (6) and (7), the Q-function should satisfy the following consistency condition:

$$Q(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)) = r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)) + \gamma Q(\boldsymbol{\nu}(t+1), \mathbf{x}(t+1), \mathbf{u}(t+1)). \tag{8}$$

The critic in the CPG-actor-critic method approximates the Q-function based on (8) as in the Sarsa algorithm [4]. As a function approximator for the critic, we employ the normalized Gaussian neural network (NGnet). The efficient on-line EM algorithm for the NGnet has been derived in [5]. The unit deletion and creation mechanism have also been incorporated in this on-line EM algorithm. This algorithm is suited for learning dynamic environment [6] such as the CPG-actor-critic method where the target Q-function for the critic depends on the current actor that is also modified according to the critic prediction.

The learning process for the CPG-actor-critic method is as follows. In the first phase, the state of the basic CPG and the physical system, $(\boldsymbol{\nu}(t), \mathbf{x}(t))$, is updated according to (1)-(5) by using the fixed actor for a given period of time. In this period, the critic receives the immediate reward $r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$ and is trained in on-line fashion such that the consistency condition (8) is satisfied. The input to the critic is the system state and the action $(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$. The teacher output for the critic is given by the right hand side of (8). Then, the model parameters of the NGnet are adjusted by using the on-line EM algorithm. The system state and the action trajectory $\{(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t)) | t = 0, 1, \ldots, t_{max}\}$ are saved.

In the second phase, the actor is trained using the saved trajectory. In order to increase the Q-function value (i.e., the expected future return), the weight parameters of the actor are updated by the gradient ascent method

$$\Delta\psi \propto \frac{\partial Q(\boldsymbol{\nu}, \mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \psi}, \tag{9}$$

where ψ represents the weight parameters of the actor, W_{ij}^{act} or A_{ik}. The above procedure defines one episode. The reinforcement learning proceeds by repeating these episodes.

5 Biped Robot

In the following, we apply the CPG-actor-critic method to the biped robot studied in [3]. The robot consists of five links in the sagittal plane as shown in Fig. 3(a). Each leg consists of two links and link-1 represents the remainder of

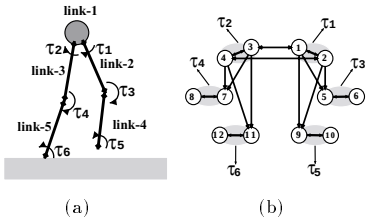

Fig. 3. (a)Biped robot. (b)CPG.

the body, which is given as a point mass. The robot state is represented by $\mathbf{x} = (x_1, \dot{x}_1, h_1, \dot{h}_1, \theta_2, \dot{\theta}_2, \ldots, \theta_5, \dot{\theta}_5)$, where x_1 and h_1 represent the horizontal and vertical coordinates of link-1, respectively. $\theta_i (i = 2, \ldots, 5)$ represents the angle of link-i from the vertical axis.

The structure of the CPG that controls the biped robot is also adopted from [3] and shown in Fig. 3(b). There are six oscillators interacting with each other. An oscillator consists of two mutually inhibiting parent neurons, ν_{2i-1} and ν_{2i} $(i = 1, \ldots, 6)$, whose output functions are the threshold function defined by $y_i = G_i(\nu_i) = \max(0, \nu_i), (i = 1, \ldots, 12)$. Each parent neuron, $\nu_i (i = 1, \ldots, 12)$, has a daughter neuron ν_{i+12} whose output function is the identity function. The daughter neuron is solely connected to its parent neuron with excite-inhibit mutual connections.

The torque $\tau_i (i = 1, \ldots, 6)$, which is applied to the i-th joint (Fig. 3), is calculated from the CPG neuron output: $\tau_i = -T_i^I y_{2i-1} + T_i^E y_{2i}$ for $i = 1, \ldots, 4$, and $\tau_i = (-T_i^I y_{2i-1} + T_i^E y_{2i}) \Xi_{i-1}$ for $i = 5, 6$. Ξ_i is an indicator function of the link-i $(i = 4, 5)$, i.e., $\Xi_i = 1$ (0) when the foot link-i touches (is off) the ground. The values of T_i^I and T_i^E are given in [3]. In the following experiment, all connection weights between CPG neurons are fixed to the value given in [3], namely, $W_{ij}^{act} \equiv 0$ is assumed in (5). We also assume a specific form of the sensory feedback to the CPG as in [3]:

$$
\begin{aligned}
I_1^{ext} &= a_1 S_1 - a_2 S_2 + a_3 S_3 + a_4 S_6, \\
I_3^{ext} &= a_1 S_2 - a_2 S_1 + a_3 S_4 + a_4 S_5, \\
I_5^{ext} &= a_5 S_4, \\
I_7^{ext} &= a_5 S_3, \\
I_9^{ext} &= -a_6 S_3 - a_7 S_4 - a_8 S_7, \\
I_{11}^{ext} &= -a_6 S_4 - a_7 S_3 - a_8 S_8, \\
I_{2i}^{ext} &= -I_{2i-1}^{ext} \quad \text{for } i = 1, \ldots, 6,
\end{aligned}
\tag{10}
$$

where $\mathbf{S} = \{\theta_2, \theta_3, \theta_4 \Xi_4, \theta_5 \Xi_5, \Xi_4, \Xi_5, \dot{\theta}_4 \Xi_4, \dot{\theta}_5 \Xi_5\}$.

6 Experiment

The main task of the reinforcement learning for the biped robot is to adjust the sensory feedback connections $\{a_i | i = 1, \ldots, 8\}$ in (10) such that the robot is able to walk stably. In order to encourage the robot to walk, an immediate reward $r(\boldsymbol{\nu}(t), \mathbf{x}(t), \mathbf{u}(t))$ is given by $\tilde{r}(\mathbf{x}(t+1))$,

$$\tilde{r}(\mathbf{x}) = 0.5 r_{height}(\mathbf{x}) + 0.02 r_{speed}(\mathbf{x}).$$

The reward $r_{height}(\mathbf{x}) = h_1 - 0.8 - \min(h_4, h_5)$, encourages the hip position to stay high, where h_i $(i = 4, 5)$ denotes the foot height of the link-i. The reward $r_{speed}(\mathbf{x}) = \max(-1, \min(\dot{x}_1, 1))$ encourages the robot to move toward the right direction. The maximum period length of one episode was 5 sec. If the robot tumbled before 5 sec, the episode was terminated at that point. The initial value of the sensory feedback connections were set to small random values.

The learning curves for the total return, the total walking length and

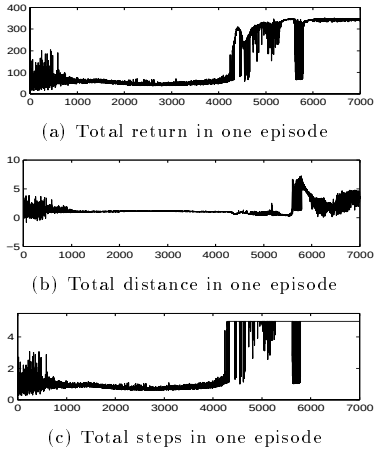

Fig. 4. Learning process.

the period length in one episode are shown in Fig. 4. At the beginning of the learning, the robot soon fell down (Fig. 5(a)). After about 4000 episodes, the robot started to run in place (Fig. 5(b)). After about 5800 episodes, the robot started to walk (Fig. 5(c)). The final sensory feedback weight values after learning were $\mathbf{a} = \{1.00, -0.31, 1.00, 0.65, 0.11, 0.25, 1.00, 0.49\}$ which are quite different from the hand-tuned values used in [3], $\mathbf{a} = \{0.15, 0.10, 0.15, 0.15, 0.30, 0.15, 0.30, 0.15\}$.

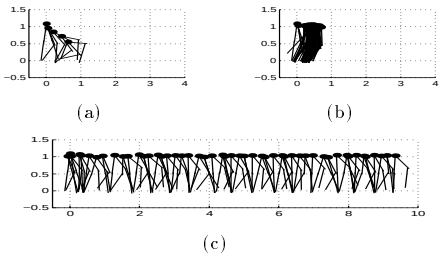

Fig. 5. Robot gait pattern. (a)Before learning. (b)After 5500 episodes. (c)After 7000 episodes.

7 Discussion

In this paper, we proposed a new RL method called the CPG-actor-critic method and applied it to biped locomotion. Although the result was successful, the learning process was rather unstable. It is necessary to increase the stability of our RL method in the future. In the current simulation, the CPG internal weights were fixed and only sensory feedback connections were adjusted by RL. It remains for future study to adjust the CPG internal weights by the CPG-actor-critic method.

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. Online 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.