Reinforcement Learning of Humanoid Rhythmic Walking Parameters based on Visual Information

Masaki Ogino, Yutaka Katoh, Masahiro Aono, Minoru Asada, and Koh Hosoda

Abstract

This paper presents a method for learning the parameters of rhythmic walking to generate purposive humanoid motions. The controller consists of the two layers: rhythmic walking is realized by the lower layer, which adjusts the speed of the phase on the desired trajectory depending on sensory information, and the upper layer learns (1) the feasible parameter sets that enable stable walking, (2) the causal relationship between the walking parameters to be given to the lower-layer controller and the change in the sensory information, and (3) the feasible rhythmic walking parameters by reinforcement learning so that a robot can reach to the goal based on visual information. The experimental results show that a real humanoid learns to reach the ball and to shoot it into the goal in the context of the *RoboCupSoccer* competition, and the further issues are discussed.

Keywords: humanoid, reinforcement learning, rhythmic walking

1 INTRODUCTION

Recent progresses in humanoid studies have made bipedal walking possible in real robots. The main approaches can be divided into two categories: the Zero Moment Point (ZMP) approach [14] and the inverted pendulum model [3]. They are characterized by using the physical parameters of a robot model explicitly to obtain the desired trajectory of each joint to be realized during walking.

However, another approach has been studied as a method which does not need to represent explicitly the precise structural parameters of a robot for walking control. This is also called a rhythmic walking based approach because the controller in this method adjusts its walking rhythm depending on sensory information so that the global entrainment of dynamics between the robot and the environment takes place.

Taga et al. [11] propose the Central Pattern Generator (CPG) model [2] for human walking based on the formulations of nonlinear dynamics. The network system changes its phase depending on the sensor information. In the simulation experiments, this model realizes stable walking under various kinds of disturbances [11, 7]. In the his CPG model, the output value of each neuron is used as the torque to be applied to a corresponding joint while almost of the currently existing humanoid robots are driven

by high gain proportional derivative (PD) controllers instead of torque control. Therefore, it is difficult to apply Taga's CPG model directly to real robots. However, even such a robot with high gain PD controllers can realize stable walking with a controller that uses sensor information properly. Pratt [9] realizes energy efficient walking in a real robot with a controller that consists of state machines. The state transition of the controller occurs when the swing leg touches the ground. Tsuchiya et al. [13] realize stable walking based on a method in which a trajectory controller determines the shape of the trajectory and a phase controller changes the speed of the desired angle on the trajectory, so that the sensor information adjusts the phase speed.

In rhythmic walking, the control parameters are found heuristically, not by planning as in the ZMP approach. This makes it difficult to construct an upper-layer controller to drive the movement of a robot because the walking parameters such as walking step are not found until the robot interacts with the environment. Taga [12] and Kimura et al. [4] construct the upper-layer controller, which gives the control parameters to the lower CPG controller depending on visual information so that the robot can avoid obstacles or climb over a step. In these methods, the designer supplies the adjusting parameters in advance. However, for making a robot more adaptive to dynamic situations, it is necessary to learn the relationship between the parameters of the lower rhythmic walking controller and the resultant change of the environment.

This paper introduces the layered controller, in which the lower-layer controller realizes rhythmic walking based on the controller proposed by Tsuchiya et al. [13] and the upper-layer controller learns the parameters of the lower-layer controller based on visual information. There are three points in learning the upper-layer controller: (1) in the first stage, it learns the feasible parameters of the lower-layer controller that enable a robot to walk, (2) to accelerate the learning process, the upper-layer controller learns the model of the world: the relationship between the control parameters given to the lower rhythmic walking controller and the change of the visual sensor information, and (3) the upper-layer controller learns which parameters should be given to reach a goal by reinforcement learning.

The rest of this paper is organized as follows: First, the lower-layer controller that enables a rhythmic walk is introduced. Next, we describe the upper-layer controller in which the parameters of the lower-layer controller are learned by reinforcement learning. Then, the experimental results applied to a RoboCupSoccer task [5], "approach a ball," are shown, and discussion is given.

2 A RHYTHMIC WALKING CONTROLLER

2.1 A biped robot model

Fig. 1 shows a biped robot model used in the experiment which has a one-link torso, two four-link arms, and two six-link legs. All joints rotate with a single degree-of-freedom (DoF). Each foot has four force-sensing-resistor (FSR) sensors to detect reaction force from the floor, and a CCD camera with a fish-eye lens is attached at the top of the torso.

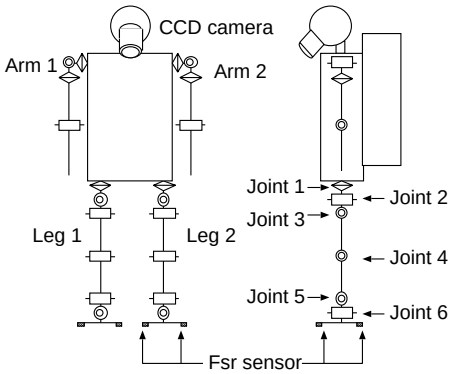


Figure 1: A model of biped locomotion robot

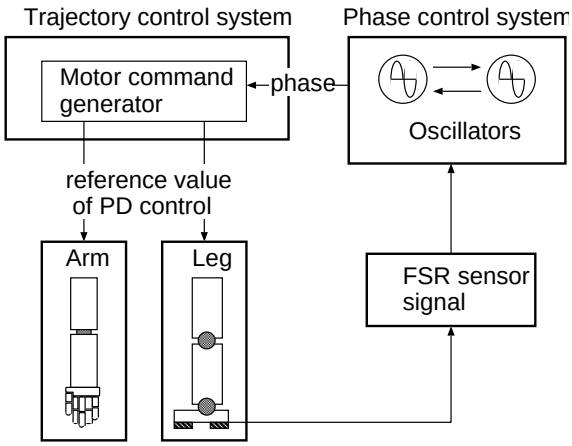


Figure 2: A walking control system

2.2 A rhythmic walking controller based on CPG principle

We build a lower-layer controller based on one proposed by Tsuchiya et al. [13]. The controller consists of two sub-controllers: *a trajectory controller* and *a phase controller* (see Fig. 2). The trajectory controller outputs the desired trajectory of each limb depending on the phase that is given by the phase controller. The phase controller consists of four oscillators, each of which is responsible for the movement of each limb (see Fig. 3). Each oscillator changes its speed depending on the touch sensor signal, and the effect is reflected on the oscillator in each limb. As a result, the desired trajectory of each joint is adjusted so that the global entrainment of dynamics between the robot and the environment takes place. In the following, the details of each controller are given.

2.2.1 Trajectory controller

The trajectory controller calculates the desired trajectory of each joint depending on the phase given by the corresponding oscillator in the phase controller. Four parameters characterize the trajectory of each joint as shown in Fig. 4. For joints 3, 4 and 5, which coincide with pitch axis, the desired trajectory is determined so that in the swing phase the foot trajectory draws a ellipse that has the radii h in the vertical direction and β in the horizontal direction, respectively. For joints 2 and 6, which coincide with roll axis, the desired trajectory is determined so that the leg tilts from $-W$ to W relative to the vertical axis. The amplitude of the oscillation, α, determines the desired trajectory of joint 1. The desired trajectories are summarized by the following functions:

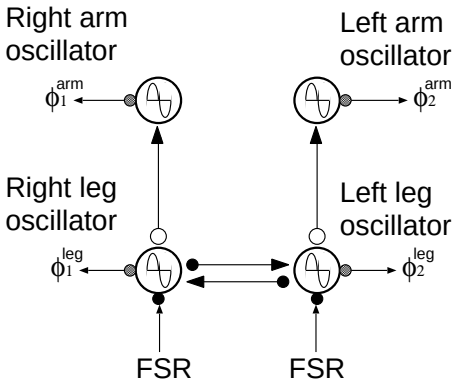


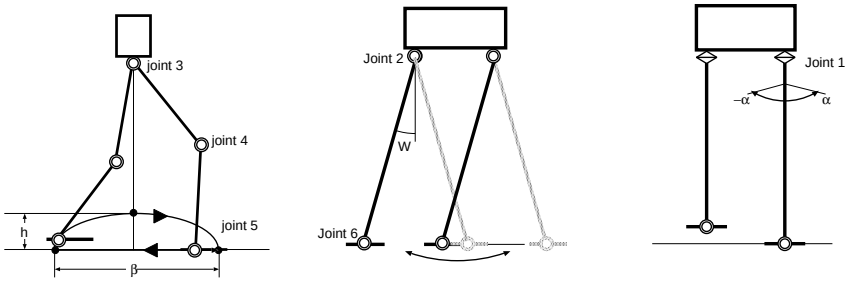
Figure 3: A phase control system

$$\theta_1 = \alpha \sin(\phi), \tag{1}$$

$$\theta_2 = W \sin(\phi), \tag{2}$$

$$\theta_i = f_i(\phi, h, \beta), \qquad (i = 3, 4, 5) \qquad \text{and} \tag{3}$$

$$\theta_6 = -W \sin(\phi). \tag{4}$$



(a) Joint angles around pitch axis

(b) Joint angles around roll axis

(c) Joint angle around yaw axis

Figure 4: Joint angles

The detail of f_i is explained in the Appendix. Among the four parameters described above, α, which determines the walking step length, and β, which determines the walking direction, are selected as rhythmic parameters of walking. Although these parameters characterize approximate direction and step length, they do not determine the resultant walking as precisely because of slippage between the support leg and the ground. These parameters are learned in the upper-layer learning module, which is described in **3**.

2.2.2 Phase controller

The phase controller sets the phase that determines the desired value of each joint. The phase controller consists of two oscillators, ϕ_R and ϕ_L, for the right and left leg, respectively. The dynamics of each oscillator is determined by the basic frequency, ω, the interaction term between two oscillators, and the feedback signal from the sensory information,

$$\dot{\phi}_L = \omega - K(\phi_L - \phi_R - \pi) + g_L \tag{5}$$

$$\dot{\phi}_R = \omega - K(\phi_R - \phi_L - \pi) + g_R. \tag{6}$$

The second term on the RHS in the above equations ensures that the oscillators have opposite phases. The third term, feedback signal from sensor information, is given as follows:

$$g_i = \begin{cases} K' Feed_i & (0 < \phi < \phi_C) \\ -\omega(1 - Feed_i) & (\phi_C \leq \phi < 2\pi) \end{cases} \quad i = \{R, L\}, \tag{7}$$

where K', ϕ_C and $Feed_i$ denote feedback gain, the phase when the swing leg contacts with the ground, and the feedback sensor signal, respectively. $Feed_i$ returns 1 if the FSR sensor value of the corresponding leg exceeds a certain threshold value, otherwise 0. The third term in (5) and (6) ensures that the mode switching between the swing phase and the support phase happens appropriately according to the ground

contact information from the FSR sensors. In this paper, the values of parameters are set as follows: $\phi_C = \pi$ [rad], $\omega = 5.23$ [rad/sec], $K = 15.7$ and $K' = 1$.

3 REINFORCEMENT LEARNING WITH RHYTHMIC WALKING PARAMETERS

3.1 The principle of reinforcement learning

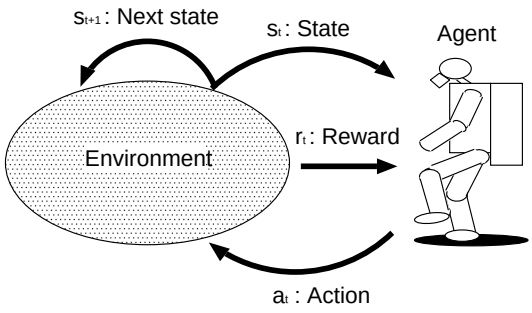


Figure 5: A basic model of agent-environment interaction

Reinforcement learning has been receiving increased attention as a method of robot learning with little or no *a priori* knowledge and a higher capability for reactive and adaptive behaviors. Fig. 5 shows a basic model of robot-environment interaction [10], in which a robot and environment are modelled by two synchronized finite state automatons interacting in a discrete time cyclical processes. The robot senses the current state $s_t \in \boldsymbol{S}$ of the environment and selects an action $a_t \in A$. Based on the state and action, the environment makes a transition to a new state $s_{t+1} \in \boldsymbol{S}$ and generates a reward r_{t+1} that is passed back to the robot. Through these interactions, the robot learns a purposive behavior to achieve a given goal. For the learning to converge correctly, the environment should satisfy the Markovian assumption that the state transition depends on only the current state and the action taken. A stochastic function $\boldsymbol{T}$ which maps a state-action pair to the next state ($\boldsymbol{T} : \boldsymbol{S} \times \boldsymbol{A} \rightarrow \boldsymbol{S}$) models the state transition. Using $\boldsymbol{T}$, the state transition probability $P_{s_t,s_{t+1}}(a_t)$ is given by

$$P_{s_t,s_{t+1}}(a_t) = Prob(\boldsymbol{T}(s_t, a_t) = s_{t+1}). \tag{8}$$

The reward function gives the immediate reward, r_t, in terms of the current state by $R(s_t)$, that is $r_t = R(s_t)$. Generally, $P_{s_t,s_{t+1}}(a_t)$ (hereafter $\mathcal{P}^a_{ss'}$) and $R(s_t)$ (hereafter $\mathcal{R}^a_{ss'}$) are unknown.

The aim of the reinforcement learner is to maximize the accumulated summation of the given rewards (called *return*) given by

$$return(t) = \sum_{n=0}^{\infty} \gamma^n r_{t+n}, \tag{9}$$

where $\gamma\,(0 \leq \gamma \leq 1)$ denotes a discounting factor to give the temporal weight to the reward.

If the state transition probability is known, the optimal policy that maximizes the expected *return* is given by finding the optimal value function $V^*(s)$ or the optimal action value function $Q^*(s,a)$ as follows. Their derivation can be found elsewhere [10].

$$\begin{aligned} V^*(s) &= \max_a E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^*(s')\right] \end{aligned} \tag{10}$$

$$\begin{aligned} Q^*(s,a) &= E\{r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a\} \\ &= \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q^*(s', a')\right] \end{aligned} \tag{11}$$

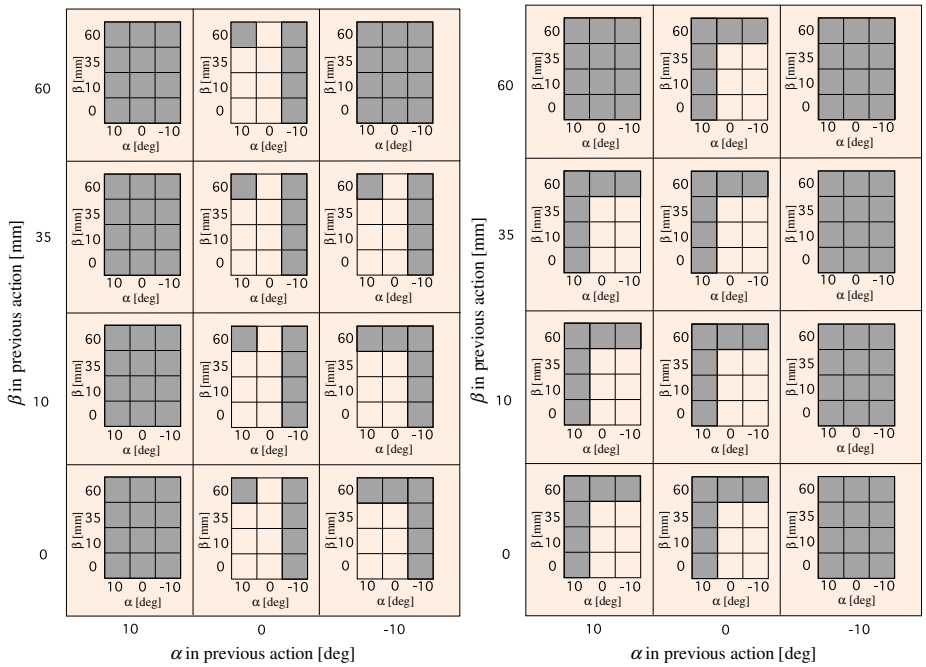
In this paper, the learning module examines the state transition when both feet are in contact with the ground so that the stable visual information can be obtained (some experiments show the possibility that human brains may calculate the length to the obstacle by visual information at the double stance phase [6]). The state space s consists of the visual information s_v and the robot posture s_p, and the action is setting the two parameters of rhythmic walking. Details are explained in the following subsections.

3.2 Construction of action space based on rhythmic parameters

The learning process has two stages. The first stage constructs the action space consisting of feasible combinations of two rhythmic walking parameters (α, β). To do that, we prepared the three-dimensional posture space s_p in terms of the forward length β (quantized into four lengths: 0, 10, 35 60 [mm]) and the turning angle α (quantized into three angles: -10, 0, 10 [deg]), which are the previous action command and the leg side (left or right). Therefore, we have 24 kinds of postures. First, we have constructed the action space of the feasible combinations of (α, β) and excluded the infeasible combinations which cause collisions with its own body. Then, various combinations of actions are examined for stable walking in the real robot. Fig. 6 shows the feasible actions (empty boxes) for each leg corresponding to the previous actions. Owing to physical differences between the two legs, the constructed action space was not symmetric, although theoretically it should be.

3.3 Reinforcement learning with visual information

Fig. 7 shows an overview of the whole system, which consists of two layers: adjusting walking based on visual information and generating walking based on neural oscillators. The state space consists of the visual information s_v and the robot posture s_p, and adjusted action a is learned by a dynamic programming (DP) method based on the rhythmic walking parameters (α, β). For the ball shooting task, s_v consists of ball substates and goal substates, which are quantized as shown in Fig. 8. We add two more substates, that is, "the ball is missing" and "the goal is missing" because they are necessary to recover from loosing sight of the ball or goal.



(a) left leg (b) right leg

Figure 6: Experimental result of action rule

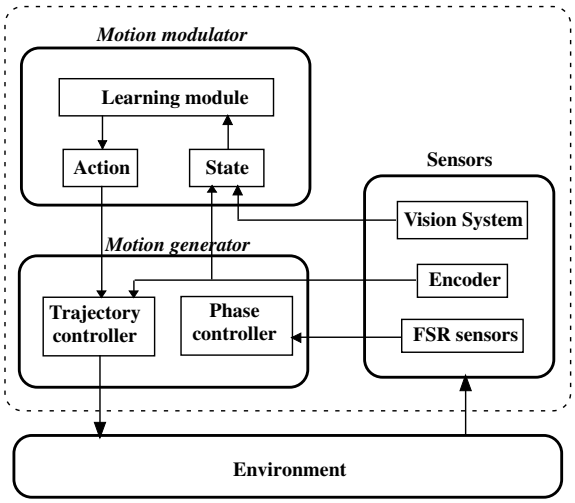
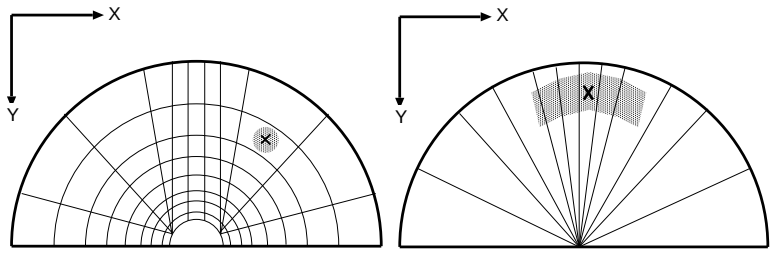


Figure 7: The biped walking system with visual perception



(a) The state space of ball (b) The state space of goal

Figure 8: The state space of ball and goal

The learning module consists of a planner that determines an action a based on the current state s, a state transition model that estimates the state transition probability $\mathcal{P}^a_{ss'}$ through the interactions, and a reward model (see Fig. 9). Based on DP, the action value function $Q(s,a)$ is updated and the learning stops when there are no more changes in the summation of action values.

$$Q(s,a) = \sum_{s'} \mathcal{P}^a_{ss'}[\mathcal{R}_s + \gamma \max_{a'} Q(s',a')], \tag{12}$$

where $\mathcal{R}_s$ denotes the expected reward at the state s.

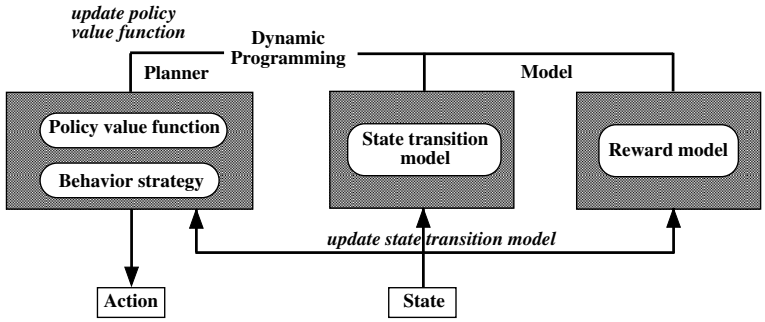


Figure 9: The learning module

4 EXPERIMENTS

4.1 A robot platform and environment set-up

We use a humanoid platform HOAP-1 by Fujitsu Automation Ltd. [8] attaching a CCD camera with a fish-eye lens at the head. Figs. 10 and 11 show a picture and a system configuration, respectively. The height and the weight are about 480 mm and 6 kg, and each leg has six degrees-of-freedom and each arm has four. Joint encoders have high resolution of 0.001 [deg/pulse] and reaction force sensors (FSRs) are attached to the soles. Color image processing to detect an orange ball and a blue goal is performed on the CPU (Pentium Ⅲ 800 MHz) under RT-Linux. Fig. 12 shows an on-board image.

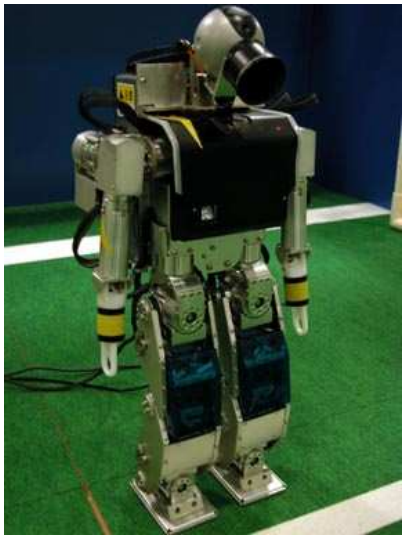


Figure 10: HOAP-1

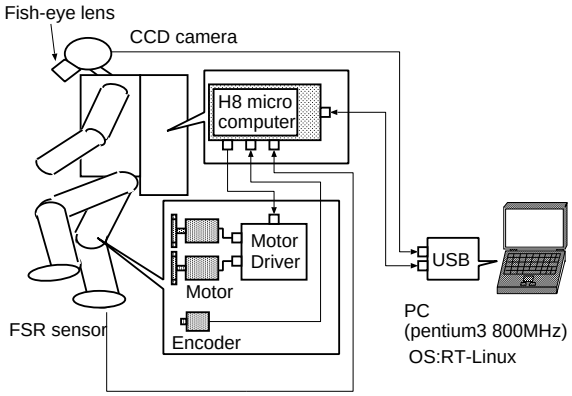


Figure 11: Overview of robot system

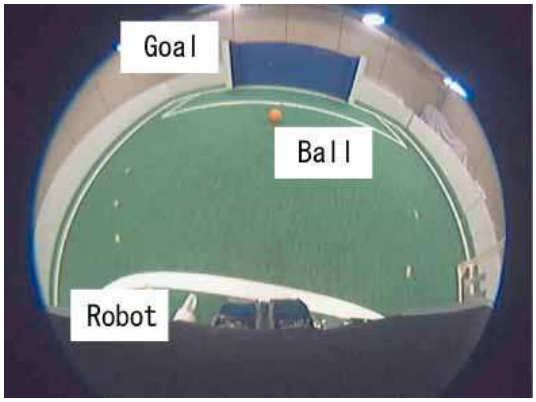


Figure 12: Robot's view (CCD camera image through fish-lens)

The experimental set-up is shown in Fig. 13 where the initial robot position is inside the circle whose center and radius are the ball position and 1000 mm, respectively, and the initial ball position is located less than 1500 mm from the goal whose width is 1800 mm and height is 900 mm. The task is to take a position just before the ball so that the robot can shoot the ball into the goal. Each episode ends when the robot succeeds in getting such positions or fails (touches the ball or the pre-specified time period expires).

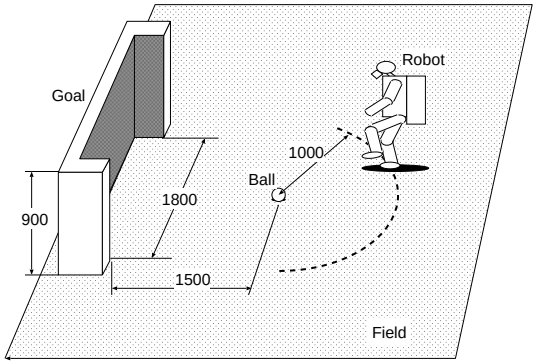


Figure 13: Experimental environment

4.2 Experimental results

One of the most serious issues in applying the reinforcement learning method to real robot tasks is how to accelerate the learning process. Instead of using Q-learning that is most typically used in many applications, we use a DP approach based on the state transition model $\mathcal{P}^a_{ss'}$, which is obtained separately from the learning behavior. Furthermore, we give the instructions to start up the learning: during the first 50 episodes (about half an hour), the human instructor avoids useless exploration by directly specifying the action command to the learner about 10 times per episode. After that, the learner experienced about 1500 episodes. Owing to the state transition model and initial instructions, learning converged in 15 hours, and the robot learned to get to the right position from any initial positions inside the half field.

Fig. 14 shows the learned behaviors from various initial positions. In Figs. 14 (a)-(e), the robot can capture the image including both the ball and the goal from the initial position while in Fig. 14 (f) the robot cannot see the ball or the goal from the initial position.

5 DISCUSSION

This study shows the possibilities for humanoid to correlate its walking parameters and on-board visual information through its experiences based on the so-called model-free approach which does not need very precise model parameters that are usually necessary for the model-based approach.

(a) Result 1 (b) Result 2 (c) Result 3

(d) Result 4 (e) Result 5 (f) Result 6

Figure 14: Experimental results

In our approach, motion commands are directly correlated with camera image without the complicated calibration process. It enables to evaluate the motion commands from the viewpoint of the achievement of the task.

This sort of approach has been already studied in wheel robots [1]. It is necessary to keep walking stabilization to apply reinforcement learning to humanoid robot. There are two points to realize stable walking in this study. The first point is keeping the walking trajectory continuous when walking parameters are changed. To do that, the planned trajectory before the change is modified so that the effects on walking can be as little as possible. The second is the action rules described in Fig. 6. These rules impose constraints on the selection of action parameters. For example, a robot cannot select left turn command with long step length just after right turn command in the previous step.

There is still much room for improvement in this study as a model-free approach. One of the problems is learning time. In our experiments, although 1500 episodes are examined and convergence is conducted with the state transition probability acquired through those episodes, learning results are not completely optimal. For example, the selected step length is not maximum limit at the place where a robot is far from the goal place. Learning shows good convergence when the experimental setting is simplified: approach to the ball on a straight line. When a robot is far from the goal, the maximum step length is selected. This may be because the number of the states and actions in this simplified experiment is much smaller than that in the experiment of approaching to a ball from the various

positions. Therefore, learning acceleration in the complicated environment is one of our future works.

6 CONCLUSION

Vision-based humanoid behavior was generated by reinforcement learning with rhythmic walking parameters. Since the humanoid generally has many DoFs, it is very hard to control all of them. Instead of using these DoFs in the action space, we adopted rhythmic walking parameters, which drastically reduces the search space and, therefore, real robot learning was possible in a reasonable amount of time. In this study, the designer specified the state space consisting of visual features and robot postures. State space construction by learning is an issue for future exploration.

Acknowledgments

We would like to thank Karl F. MacDorman for stimulating discussions and suggestions. This study was partially funded by the Advanced and Innovational Research Program in the Life Sciences of the Ministry of Education, Culture, Sports, Science, and Technology of the Japanese Government.

REFERENCES

[1] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda, Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning, Machine Learning, **23**, pp. 279-303, (1996).

[2] S. Grillner, Neurobiological bases of rhythmic motor acts in vertebrates, *Science*, **228**, pp. 143-149 (1985).

[3] S. Kajita and K. Tani, Adaptive gait control of a biped robot based on realtime sensing of the ground profile, in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 570-577 (1996).

[4] H. Kimura, Y. Fukuoka, and H. Nakamura, Biologically inspired adaptive dynamic walking of the quadruped on irregular terrain, in *Proc. of 9th International Symposium of Robotics Research*, pp. 271-278 (1999).

[5] H. Kitano and M. Asada, The RoboCup humanoid challenge as the millennium challenge for advanced robotics, *Advanced Robotics*, **13** (8), pp. 723-736 (2000).

[6] M. Laurent and J. A. Thomson, The role of visual information in control of a constrained locomotor task. *Journal of Motor Behavior*, **20**, pp. 17–37 (1988).

[7] S. Miyakoshi, G. Taga, Y. Kuniyoshi, A. Nagakubo, Three dimensional bipedal stepping motion using neural oscillators –towards humanoid motion in the real world–, in *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 84-89 (1998).

[8] Y. Murase, Y. Yasukawa, K. Sakai, Design of a compact humanoid robot as a platform, in *Proceedings of 19th Conference of Robotics Society of Japan*, pp. 789-790 (2001).

[9] J. Pratt, Exploiting Inherent robustness and natural dynamics in the control of bipedal walking robots, Doctoral thesis, MIT, June (2000).

[10] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Bradford Books, March (1998).

[11] G. Taga, Y. Yamaguchi, H. Shimizu, Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment, *Biological Cybernetics*, **65**, pp. 147–159 (1991).

[12] G. Taga, A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance, *Biological Cybernetics*, **78**, pp. 9–17 (1998).

[13] K. Tsuchiya, K. Tsujita, K. Manabu, S. Aoi, An emergent control of gait patterns of legged locomotion robots", in *Proc. of the Symposium on Intelligent Autonomous Vehicles*, pp. 271-276 (2001).

[14] J. Yamaguchi, N. Kinoshita, A. Takanishi, I. Kato, Development of a dynamic biped walking system for humanoid –development of a biped walking robot adapting to the human's living floor–, in *Proc. of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 232-239 (1996).

APPENDIX: Planning the reference trajectory around the pitch axis

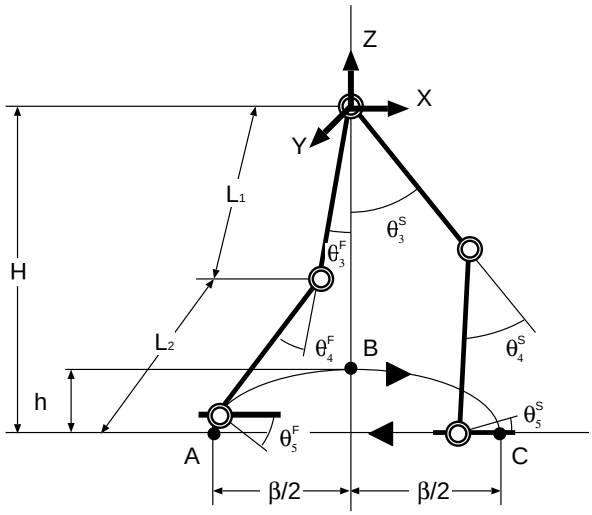


Figure 15: Joint angles and the reference trajectory of the foot

The position of the foot determines the reference trajectories of joints 3, 4 and 5. Let x and z be the position of the foot in the plane XZ which is perpendicular to the pitch axis. The reference trajectory of the foot is given by

$$x_F = \frac{\beta}{2}\cos(\phi^F), \tag{13}$$

$$z_F = -H + h\sin(\phi^F), \tag{14}$$

$$x_S = -\frac{\beta}{2}\cos(\phi^S), \tag{15}$$

$$z_S = -H, \tag{16}$$

$$\tag{17}$$

where (x_F, z_F) and (x_S, z_S) are the positions of the foot in the swing and support phase, respectively, H is the length from the ground to the joint 3, β is the step length, and h is the maximum height of the foot from the ground (Fig. 15). When the position of the foot is determined, the angle of each joint to be realized is calculated by the inverse kinematics as follows,

$$\theta_3 = \frac{\pi}{2} + atan2(z, x) - atan2(k, x^2 + z^2 + L_1^2 - L_2^2), \tag{18}$$

$$\theta_4 = atan2(k, x^2 + z^2 - L_1^2 - L_2^2), \tag{19}$$

$$\theta_5 = -(\theta_3 + \theta_4), \tag{20}$$

where k is given by the following equation,

$$k = \sqrt{(x^2 + z^2 + L_1^2 + L_2^2)^2 - 2\{(x^2 + z^2)^2 + L_1^4 + L_2^4\}}. \tag{21}$$

In this research, the value of each parameter is set as follows: $H = 185$ [mm], $h = 8$ [mm], $W = 13$ [deg], $L_1 = 100$ [mm] and $L_2 = 100$ [mm].