Poincaré-Map-Based Reinforcement Learning For Biped Walking

Jun Morimoto[1,2], Jun Nakanishi[1,2], Gen Endo[2,3], and Gordon Cheng[1,2]
[1]*Computational Brain Project, ICORP, JST*
[2]*ATR Computational Neuroscience Laboratories*
[3]*Sony Intelligence Dynamics Laboratories, Inc.*
2-2-2 Hikaridai Soraku-gun Seika-cho, Kyoto, 619-0288, JAPAN
xmorimo@atr.jp

Christohper G. Atkeson and Garth Zeglin
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, 15213, USA
cga@cs.cmu.edu

***Abstract*— We propose a model-based reinforcement learning algorithm for biped walking in which the robot learns to appropriately modulate an observed walking pattern. Via-points are detected from the observed walking trajectories using the minimum jerk criterion. The learning algorithm modulates the via-points as control actions to improve walking trajectories. This decision is based on a learned model of the Poincaré map of the periodic walking pattern. The model maps from a state in the single support phase and the control actions to a state in the next single support phase. We applied this approach to both a simulated robot model and an actual biped robot. We show that successful walking policies are acquired.**

***Index Terms*— Biped Walking; Reinforcement Learning; Poincaré map**

I. Introduction

We propose a learning algorithm to acquire an appropriate biped walking controllers by modulating an observed walking pattern. We are using model-based reinforcement learning, where we learn a model of a Poincaré map and then choose control actions based on a computed value function. We detect via-points from an observed walking trajectory and use the via-points as control actions.

Several researchers have applied reinforcement learning to biped locomotion [12], [2]. Few studies deal with a physical robots because reinforcement learning methods often require large numbers of trials. The policy gradient method [17] is one of the reinforcement learning methods successfully applied to learn biped walking on actual robots [1], [19]. However, [1] requires hours to learn a walking controller, and [19] requires a mechanically stable robot.

On the other hand, [3] reported that a model-based approach to reinforcement learning is able to accomplish given tasks much faster than without using knowledge of the environment. In our previous work [11], we showed that a model-based approach using an approximated Poincaré map could be applied to learn biped walking in small numbers of trials. However, we used an empirically designed nominal trajectory for the proposed method, and acquired a successful walking pattern only in a simulated environments. In this study, we use observed trajectories, such as those of humans or other robots controlled by this or other algorithms, as nominal trajectories. We show that the proposed method can be applied to an actual robot (Fig. 1).

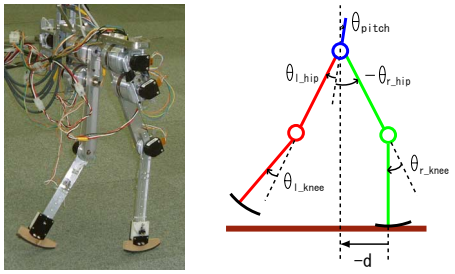

Fig. 1. Five link biped robot. Input state $\mathbf{x} = (d, \dot{d})$

First, we use a simulated 5 link biped robot (Fig. 1) to evaluate our proposed method. Physical parameters of the 5 link simulated robot in *TABLE I* are selected to model the actual biped robot fixed to a boom that keeps the robot in the sagittal plane (Fig. 1). Our biped has a short torso and round feet without ankle joints. For these bipeds, controlling biped walking trajectories with the popular ZMP approach [7], [22] is difficult or impossible, and thus an alternative method for controller design must be used.

In section II, we introduce our reinforcement learning method for biped walking. In section III, we show simulation results. In section IV, we present an implementation of the proposed method on the real robot, and then demonstrate that the robot acquires a successful walking pattern within 100 trials.

II. Poincaré-map-based reinforcement learning for biped locomotion

We improve biped walking controllers based on an approximated Poincaré map using a model-based reinforce-

TABLE I

PHYSICAL PARAMETERS OF THE FIVE LINK ROBOT MODEL

	trunk	thigh	shin
mass $[kg]$	2.0	0.64	0.15
length $[m]$	0.01	0.2	0.2
inertia $(\times 10^{-4}\ [kg \cdot m^2])$	1.0	6.9	1.4

ment learning framework [3], [16]. The Poincaré map represents the locus of intersection of the biped trajectory with a hyperplane subspace of the full trajectory state space. In our case, we are interested in the system state at two symmetric phase angles of the walking gait. Modulating via-points affects the locus of intersection and our learned model reflects this effect. Given a learned mapping, we proceed to learn a corresponding value function for states at phases $\phi = \frac{1}{2}\pi$ and $\phi = \frac{3}{2}\pi$ (Fig. 2), where we define phase $\phi = 0$ as the left foot touchdown.

The input state is defined as $\mathbf{x} = (d, \dot{d})$, where d denotes the horizontal distance between the stance foot position and the body position (Fig. 1). We use the hip position as the body position because the center of mass is nearly coincident with the hips (Fig. 1). We use a human walking pattern in [5] as the nominal trajectory (Fig. 3). The action of the robot $\mathbf{u} = \boldsymbol{\theta}^{act}(\mathbf{x}) = (\theta^{act}_{hip}, \theta^{act}_{knee})$ modulates the via-points of the nominal trajectory at each joint:

$$\theta^i_{hip\_vp} = \bar{\theta}^i_{hip\_vp} + \theta^{act}_{hip}(\mathbf{x}) \quad (i = 1, \cdots, n^v_{hip}), \tag{1}$$

$$\theta^i_{knee\_vp} = \bar{\theta}^i_{knee\_vp} + \theta^{act}_{knee}(\mathbf{x}) \quad (i = 1, \cdots, n^v_{knee}), \tag{2}$$

where $n^v_{hip} = 1$ and $n^v_{knee} = 2$ denote the number of selected via-points, and $\bar{\theta}^i_{hip\_vp}$ and $\bar{\theta}^i_{knee\_vp}$ denote the nominal value of the selected via-points. Each selected via-point on a same joint is equally modulated by the control output θ^{act}.

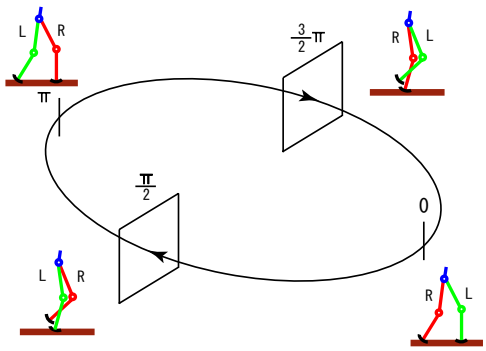

Fig. 2. Biped walking cycle: we update parameters and select actions at Poincaré sections at phase $\phi = \frac{\pi}{2}$ and $\phi = \frac{3\pi}{2}$. L:left leg, R:right leg

A. Function approximator

We use Receptive Field Weighted Regression(RFWR) [15] as the function approximator for the policy, the value function and the estimated Poincaré

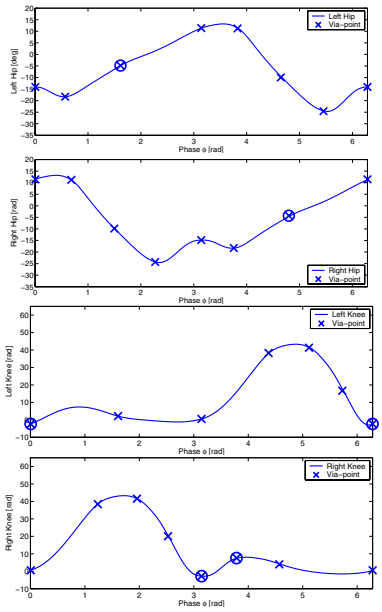

Fig. 3. Nominal joint-angle trajectories observed from a human walking pattern and detected via-points represented by cross ($\times$). Manually selected via-points represented by circle ($\circ$) are modulated by control output θ_{act}. Note that amplitude of the human walking pattern is multiplied by 0.7 to match the small size robot (Fig 1).

map. We approximate a target function $g(\mathbf{x})$ with

$$\hat{g}(\mathbf{x}) = \frac{\sum_{k=1}^{N_b} a_k(\mathbf{x}) h_k(\mathbf{x})}{\sum_{k=1}^{N_b} a_k(\mathbf{x})}, \tag{3}$$

$$h_k(\mathbf{x}) = \mathbf{w}_k^T \tilde{\mathbf{x}}_k, \tag{4}$$

$$a_k(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_k)^T \mathbf{D}_k (\mathbf{x} - \mathbf{c}_k)\right), \tag{5}$$

where $\mathbf{c}_k$ is the center of the k-th basis function, $\mathbf{D}_k$ is the distance metric of the k-th basis function, N_b is the number of basis functions, and $\tilde{\mathbf{x}}_k = ((\mathbf{x} - \mathbf{c}_k)^T, 1)^T$ is the augmented state. The update rule for the parameter $\mathbf{w}$ is given by:

$$\Delta \mathbf{w}_k = a_k \mathbf{P}_k \tilde{\mathbf{x}}_k (g(\mathbf{x}) - h_k(\mathbf{x})), \tag{6}$$

where

$$\mathbf{P}_k \leftarrow \frac{1}{\lambda}\left(\mathbf{P}_k - \frac{\mathbf{P}_k \tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T \mathbf{P}_k}{\frac{\lambda}{a_k} + \tilde{\mathbf{x}}_k^T \mathbf{P}_k \tilde{\mathbf{x}}_k}\right), \tag{7}$$

and $\lambda = 0.999$ is the forgetting factor.

In this study, we allocate a new basis function if the activation of all existing units is smaller than a threshold a_{min}, i.e.,

$$\max_k a_k(\mathbf{x}) < a_{\min}, \tag{8}$$

where $a_{\min} = \exp(-\frac{1}{2})$. We initially align basis functions $a_k(\mathbf{x})$ at even intervals in each dimension of input space $\mathbf{x} = (d, \dot{d})$ (Fig. 1) $[-0.2(m) \leq d \leq 0.2(m)$ and $-1.0(m/s) \leq \dot{d} \leq 1.0(m/s)]$. Initial numbers of basis

functions are 400(=20×20) for approximating the policy and the value function. We put 1 basis function at the origin for approximating the Poincaré map. We set the distance metric $\mathbf{D}_k$ to $\mathbf{D}_k = diag\{2500, 90\}$ for the policy and the value function, and $\mathbf{D}_k = diag\{2500, 225, 1600, 1600\}$ for the Poincaré map. The centers of the basis functions $\mathbf{c}_k$ and the distance metrics of the basis functions $\mathbf{D}_k$ are fixed during learning.

B. Learning the Poincaré map of biped walking

We learn a model that predicts the state of the biped a half cycle ahead, based on the current state and the modulated via-points. We are predicting the location of the system in a Poincaré section at phase $\phi = \frac{3\pi}{2}$ based on the system's location in a Poincaré section at phase $\phi = \frac{\pi}{2}$(Fig. 2). We use a different model to predict the location at phase $\phi = \frac{\pi}{2}$ based on the location at phase $\phi = \frac{3\pi}{2}$ due to the real robot possessing asymmetries caused by a supporting boom.

Because the state of the robot drastically changes at foot touchdown ($\phi = 0, \pi$), we select the phases $\phi = \frac{\pi}{2}$ and $\phi = \frac{3\pi}{2}$ as Poincaré sections. We approximate this Poincaré map using a function approximator with a parameter vector $\mathbf{w}^m$,

$$\hat{\mathbf{x}}_{\frac{3\pi}{2}} = \hat{\mathbf{f}}_1(\mathbf{x}_{\frac{\pi}{2}}, \mathbf{u}_{\frac{\pi}{2}}; \mathbf{w}_1^m), \quad (9)$$

$$\hat{\mathbf{x}}_{\frac{\pi}{2}} = \hat{\mathbf{f}}_2(\mathbf{x}_{\frac{3\pi}{2}}, \mathbf{u}_{\frac{3\pi}{2}}; \mathbf{w}_2^m), \quad (10)$$

where the input state is defined as $\mathbf{x} = (d, \dot{d})$, and the action of the robot is defined as $\mathbf{u} = \boldsymbol{\theta}^{act}(\mathbf{x})$.

C. Representation of biped walking trajectories and the low-level controller

We interpolated trajectories between the via-points by using the minimum jerk criteria [6], [21]. To follow the generated target trajectories, the torque output at each joint is given by a PD servo controller:

$$\tau_j = k(\theta_j^d(\phi) - \theta_j) - b\dot{\theta}_j, \quad (11)$$

where $\theta_j^d(\phi)$ is the target joint angle for j-th joint ($j = 1 \cdots 4$), position gain k is set to $k = 4.0$ except for the knee joint of the stance leg (we use $k = 9.0$ for the knee joint of the stance leg), and the velocity gain b is set to $b = 0.1$.

We reset the phase [20], [13] to $\phi = \phi_{reset}$ at left foot touchdown and to $\phi = \pi + \phi_{reset}$ at right foot touchdown, where $\phi_{reset} = 0.7$ rad is empirically determined.

D. Rewards

The robot gets a reward r according to the control cost r_{cost} and walking velocity r_{vel}:

$$r(t) = r_{cost}(t) + r_{vel}(t), \quad (12)$$

where $r_{cost}(t) = 0.1 \sum_j \tau_j^2(t)\Delta t$, $r_{vel}(t) = v(t)\Delta t$, $v(t)$ m/sec denotes walking speed, and $\Delta t = 0.001$ sec. The robot gets punishment (negative reward) if it falls down. If the height of the body goes below 0.38m, the robot is given a negative reward (-1) and the trial is terminated.

E. Learning the value function

In a reinforcement learning framework, the learner tries to create a controller which maximizes expected total return. We define the value function for the policy μ:

$$V^{\mu}(\mathbf{x}(t)) = E[r(t+1) + \gamma r(t+2) + \gamma^2 r(t+3) + ...], \quad (13)$$

where $r(t)$ is the reward at time t, and γ ($0 \leq \gamma \leq 1$) is the discount factor. In this framework, we evaluate the value function only at $\phi(t) = \frac{\pi}{2}$ and $\phi(t) = \frac{3}{2}\pi$. Thus, we consider our learning framework as model-based reinforcement learning for a semi-Markov decision process (SMDP) [18]. We use a function approximator with a parameter vector $\mathbf{w}^v$ to represent the value function:

$$\hat{V}(t) = \hat{V}(\mathbf{x}(t); \mathbf{w}^v). \quad (14)$$

By considering the deviation from equation (13), we can define the temporal difference error (TD-error) [16], [18]:

$$\delta(t) = \sum_{k=t+1}^{t_T} \gamma^{k-t-1} r(k) + \gamma^{t_T - t}\hat{V}(t_T) - \hat{V}(t), \quad (15)$$

where t_T is the time when $\phi(t_T) = \frac{1}{2}\pi$ or $\phi(t_T) = \frac{3}{2}\pi$. The update rule for the value function can be derived as

$$\hat{V}(\mathbf{x}(t)) \leftarrow \hat{V}(\mathbf{x}(t)) + \beta\delta(t), \quad (16)$$

where $\beta = 0.2$ is a learning rate. The parameter vector $\mathbf{w}^v$ is updated by equation (6).

F. Learning a policy for biped locomotion

We use a stochastic policy to generate exploratory action. The policy is represented by a probabilistic model:

$$\mu(\mathbf{u}(t)|\mathbf{x}(t)) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{u}(t) - \mathbf{A}(\mathbf{x}(t); \mathbf{w}^a))^2}{2\sigma^2}\right), \quad (17)$$

where $\mathbf{A}(\mathbf{x}(t); \mathbf{w}^a)$ denotes the mean of the model, which is represented by a function approximator, where $\mathbf{w}^a$ is a parameter vector. We changed the variance σ according to the trial as $\sigma = 0.1\left(\frac{150 - N_{trial}}{150}\right) + 0.01$ for $N_{trial} \leq 150$ and $\sigma = 0.01$ for $N_{trial} > 150$, where N_{trial} denotes the number of trials. The output of the policy is

$$\mathbf{u}(t) = \mathbf{A}(\mathbf{x}(t); \mathbf{w}^a) + \sigma\mathbf{n}(t), \quad (18)$$

where $\mathbf{n}(t) \sim N(0, 1)$. $N(0, 1)$ indicate a normal distribution which has mean of 0 and variance of 1.

We derive the update rule for a policy by using the value function and the estimated Poincaré map.

1) Predict the next state $\hat{\mathbf{x}}(t_T)$ from the current state $\mathbf{x}(t)$ and the nominal action $\mathbf{u} = \mathbf{A}(\mathbf{x}(t); \mathbf{w}^a)$ using the Poincaré map model $\hat{\mathbf{x}}(t_T) = \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t); \mathbf{w}^m)$.
2) Derive the gradient of the value function $\frac{\partial V}{\partial \mathbf{x}}$ at the predicted state $\hat{\mathbf{x}}(t_T)$.

3) Derive the gradient of the dynamics model $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ at the current state $\mathbf{x}(t)$ and the nominal action $\mathbf{u} = \mathbf{A}(\mathbf{x}(t); \mathbf{w}^a)$.
4) Update the policy μ:

$$\mathbf{A}(\mathbf{x}; \mathbf{w}^a) \leftarrow \mathbf{A}(\mathbf{x}; \mathbf{w}^a) + \alpha \frac{\partial V(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}, \quad (19)$$

where $\alpha = 0.2$ is the learning rate. The parameter vector $\mathbf{w}^a$ is updated by equation (6). We can consider the output $\mathbf{u}(t)$ is an *option* in the SMDP [18] initiated in state $\mathbf{x}(t)$ at time t when $\phi(t) = \frac{1}{2}\pi$ (or $\phi = \frac{3}{2}\pi$), and it terminates at time t_T when $\phi = \frac{3}{2}\pi$ (or $\phi = \frac{1}{2}\pi$).

III. Simulation Results

We applied the proposed method to the 5 link simulated robot (Fig. 1). We used a manually generated initial step to get the pattern started. We set the walking period to $T = 0.9\ sec$ ($\omega = 7.0\ rad/sec$). A trial is terminated after 15 seconds or after the robot falls down. Figure 4 (Top) shows the walking pattern before learning.

We defined a successful trial when the robot continuously walks for more than 15 seconds, which approximately corresponds to 30 steps. Figure 5 shows the accumulated reward, averaged over 10 simulation runs, at each trial. Stable walking controllers were acquired within 200 trials (Fig. 5) for every 10 simulation runs. However, the performance of the acquired controller has a large variance at each simulation run. This is probably because scheduling of the stochasticity of the policy σ in (17) was not appropriate. We need to include σ as a parameter of the policy in our future work [17].

The shape of the value function is shown in Figure 6. The minimum value of the value function is located at negative body position d and negative body velocity $\dot{d}$ because this state leads the robot to fall backward. The maximum value of the value function is located at negative body position d and positive body velocity $\dot{d}$ that leads to a successful walk. The number of allocated basis functions are 400 for approximating the value function, 400 for approximating the policy, 443 for the Poincaré map $\hat{\mathbf{f}}_1$ in equation (9), and 393 for the Poincaré map $\hat{\mathbf{f}}_2$ in equation (10).

The acquired walking pattern is shown in Figure 4 (Bottom). Figure 7 (Left) shows a phase diagram of a successful walking pattern in the state space $\mathbf{x} = (d, \dot{d})$ after learning. A gradual increase of walking speed can be observed. Figure 7 (Right) shows loci of the walking trajectory at the Poincaré section. The walking trajectory after learning passes through the section at almost same place after a few steps.

IV. Real Robot Implementation

We applied the proposed model-based reinforcement learning scheme to a real biped (Fig. 1). We use a walking pattern generated by a pre-designed state machine controller [14] as the nominal walking pattern (Fig 8). We detect via-points from this nominal walking pattern

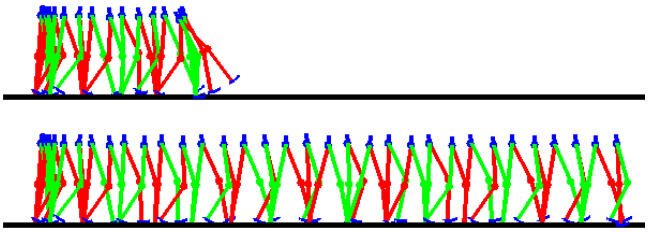

Fig. 4. Acquired biped walking pattern: (Top) Before learning, (Bottom) After learning

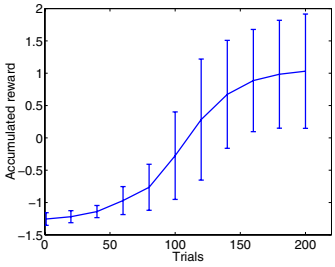

Fig. 5. Accumulated reward at each trial: Results of 10 experiments. We filtered the data with moving average of 20 trials.

and manually select via-points that correspond to foot placement (Fig 8). In this experiment, the control output $\mathbf{u} = \boldsymbol{\theta}^{act}(\mathbf{x}) = (\theta^{act}_{knee})$ modulate the selected via-points:

$$\theta^i_{knee\_vp} = \bar{\theta}^i_{knee\_vp} + \theta^{act}_{knee}(\mathbf{x}) \quad (i = 1, \cdots, n^v_{knee}). \quad (20)$$

On each transition from phase $\phi = \frac{1}{2}\pi$ (or $\phi = \frac{3}{2}\pi$) to phase $\phi = \frac{3}{2}\pi$ (or $\phi = \frac{1}{2}\pi$), the robot gets a reward of 0.1, if the height of the body remains above 0.38m during the past half cycle. The robot gets punishment (negative reward -1) if it falls down.

We changed the variance σ in equation (17) according to the trial of $\sigma = 0.1\left(\frac{50 - N_{trial}}{50}\right) + 0.01$ for $N_{trial} \leq 50$ and $\sigma = 0.01$ for $N_{trial} > 50$, where N_{trial} denotes the number of trials. We set the walking period to $T = 0.84\ sec$ ($\omega = 7.5\ rad/sec$). A trial is terminated after 30 steps or after the robot fell down. We use the pre-designed state machine for the initial 6 steps. We set the distance metric $\mathbf{D}_k$ in equation (5) to $\mathbf{D}_k = diag\{2500, 90\}$ for the policy and the value function, and $\mathbf{D}_k = diag\{2500, 90, 1600\}$ for the Poincaré map.

We also used a phase resetting method for the real robot experiment. We reset the phase to $\phi = \phi_{reset}$ at left foot touchdown and to $\phi = \pi + \phi_{reset}$ at right foot touchdown, where $\phi_{reset} = 0.3$ rad.

Figure 9 shows a biped walking pattern before learning. The robot fell over with the nominal walking pattern. After 100 trials in the real environment, the robot acquired a policy that generated a stable biped walking pattern. We applied the acquired controller to a different ground surface. Even on a metal surface, the robot successfully

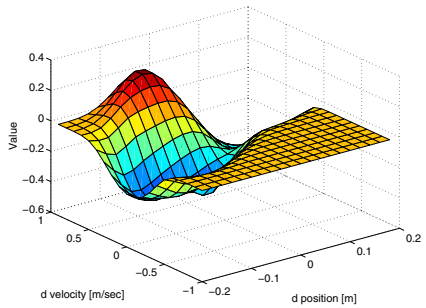

Fig. 6. Shape of acquired value function

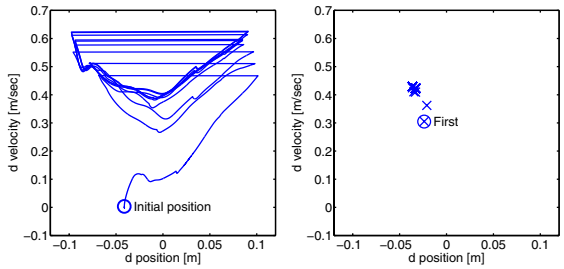

Fig. 7. (Left) Phase diagram in the state space $(d, \dot{d})$. (Right) Loci of the walking trajectory at the Poincaré section. $\otimes$ represents the first locus of the walking trajectory which passes through the section.

walked using the learned biped walking policy (Fig. 10).

Figure 11 shows the accumulated reward at each trial using the real robot. The robot learned a stable walking controller within 100 trials.

An acquired value function after 100 trials is shown in Figure 12. The minimum value of the value function is located around zero body position $d = 0.0$ and negative body velocity $\dot{d}$, and the maximum value of the value function is located around zero body position $d = 0.0$ and positive body velocity $\dot{d}$. The difference between shape of the value function acquired in the simulated environment (Fig. 6) and the real environment (Fig. 12) is possibly caused by the effect of the boom. The number of allocated basis functions are 407 for approximating the value function, 401 for approximating the policy, 59 for the Poincaré map $\hat{\mathbf{f}}_1$ in equation (9), and 59 for the Poincaré map $\hat{\mathbf{f}}_2$ in equation (10).

V. Discussion

In this study, we proposed Poincaré-map-based reinforcement learning and applied the proposed method to biped locomotion. The simulated robot acquired the biped walking controller using observed human walking pattern as the nominal trajectory. We also applied the proposed approach to a physical biped robot and acquired a policy, which successfully generated a walking pattern. We are currently trying to use a human walking trajectory as the nominal trajectory on the real biped. Automatic selection of the via-points to be used as control actions is part of our

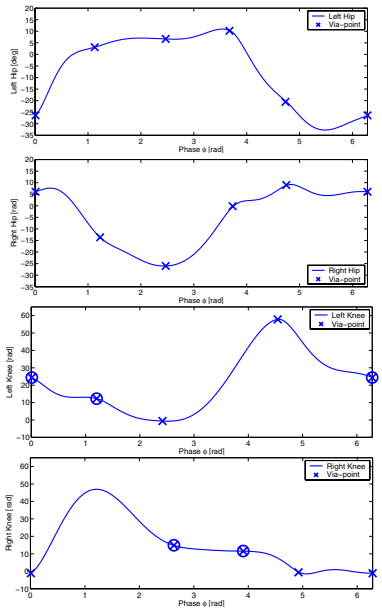

Fig. 8. Nominal joint-angle trajectories and detected via-points represented by cross ($\times$). Manually selected via-points represented by circle ($\circ$) are modulated by control output θ^{act}.

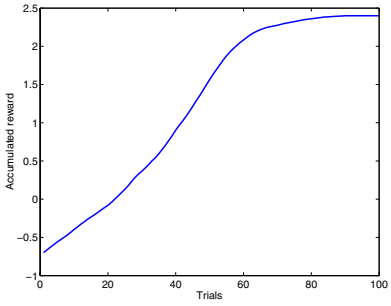

Fig. 11. Accumulated reward at each trial using real robot. We filtered the data with moving average of 20 trials.

future work. In our previous work, we have proposed a trajectory optimization method for biped locomotion [9], [10] based on differential dynamic programming [4], [8]. We are now considering combining this trajectory optimization method with the proposed reinforcement learning method.

References

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

[2] C. Chew and G. A. Pratt. Dynamic bipedal walking assisted by learning. *Robotica*, 20:477–491, 2002.

[3] K. Doya. Reinforcement Learning in Continuous Time and Space. *Neural Computation*, 12(1):219–245, 2000.

[4] P. Dyer and S. R. McReynolds. *The Computation and Theory of Optimal Control*. Academic Press, New York, NY, 1970.

[5] Y. Ehara and S. Yamamoto. *Introduction to Body-Dynamics – Analysis of Gait and Gait Initiation*. Ishiyaku Publishers, 2002 (in Japanese).

[6] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5:1688–1703, 1985.

Fig. 9. Biped walking pattern before learning

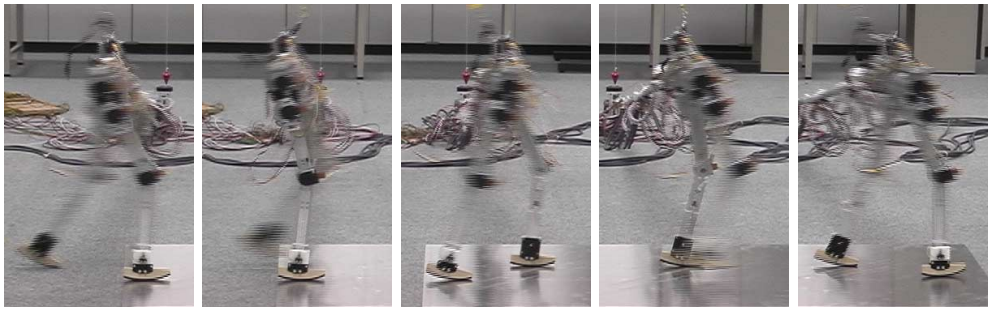

Fig. 10. Biped walking pattern on metal surface

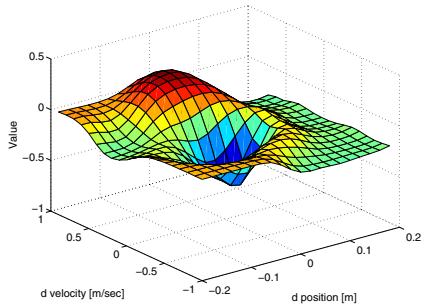

Fig. 12. Shape of acquired value function in real environment

[7] K. Hirai, M. Hirose, and T. Takenaka. The Development of Honda Humanoid Robot. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 160–165, 1998.

[8] D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier, New York, NY, 1970.

[9] J. Morimoto and C. G. Atkeson. Robust low-torque biped walking using differential dynamic programming with a minimax criterion. In Philippe Bidaud and Faiz Ben Amar, editors, *Proceedings of the 5th International Conference on Climbing and Walking Robots*, pages 453–459. Professional Engineering Publishing, Bury St Edmunds and London, UK, 2002.

[10] J. Morimoto and C. G. Atkeson. Minimax differential dynamic programming: An application to robust biped walking. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1563–1570. MIT Press, Cambridge, MA, 2003.

[11] J. Morimoto, G. Cheng, C. G. Atkeson, and G. Zeglin. A Simple Reinforcememtn Learning Algorithm For Biped Walking. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 3030–3035, 2004.

[12] Y. Nakamura, M. Sato, and S. Ishii. Reinforcement learning for biped robot. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, pages ThP–II–5, 2003.

[13] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47:79–91, 2004.

[14] M. H. Raibert. *Legged Robots That Balance*. The MIT Press, Cambridge, MA, 1986.

[15] S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084, 1998.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

[17] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063, Cambridge, MA, 2000. MIT Press.

[18] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112:181–211, 1999.

[19] R. Tedrake, T. W. Zhang, and H. S. Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligen Robots and Systems*, page (to appear), 2004.

[20] K. Tsuchiya, S. Aoi, and K. Tsujita. Locomotion control of a biped locomotion robot using nonlinear oscillators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1745–1750, Las Vegas, NV, USA, 2003.

[21] Y. Wada and M. Kawato. A theory for cursive handwriting based on the minimization principle. *Biological Cybernetics*, 73:3–15, 1995.

[22] J. Yamaguchi, A. Takanishi, and I. Kato. Development of a biped walking robot compensating for three-axis moment by trunk motion. *Journal of the Robotics Society of Japan*, 11(4):581–586, 1993.

ACKNOWLEDGMENT

We would like to thank Mitsuo Kawato, at ATR Computational Neuroscience Laboratories, Japan, and Seiichi Miyakoshi of the Digital Human Research Center, AIST, Japan for helpful discussions. Atkeson is partially supported by NSF awards ECS-0325383 and CNS-0224419.