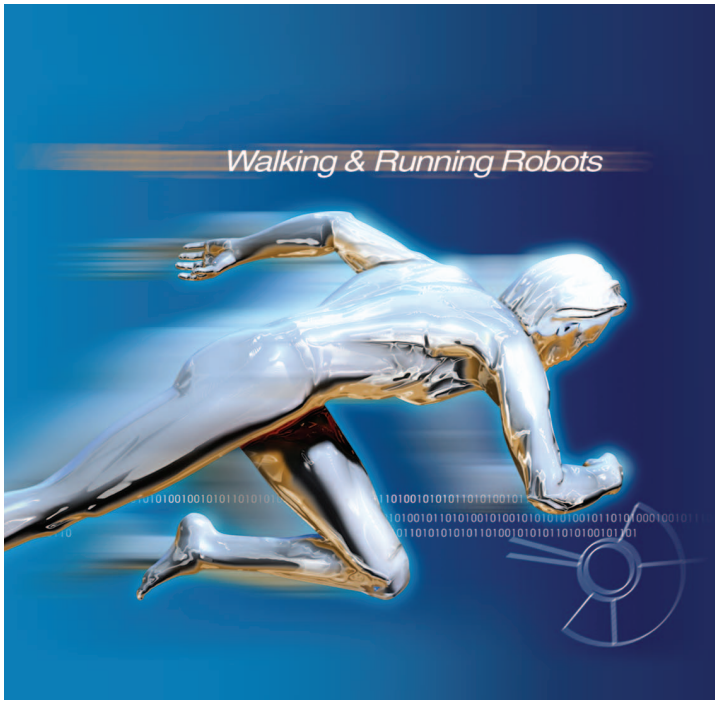

Learning Biped Locomotion

Application of Poincaré-Map-Based Reinforcement Learning

We propose a model-based reinforcement learning (RL) 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 controls the via-points 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 controlled via-points 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 were acquired.

Sophisticated biped walking controllers have been proposed in robotics [9], [17], [30], [10]. However, human-like agility, robustness, and energy efficiency have not been achieved. One possible approach is for the biped controller to learn to walk as humans do. However, applying learning to biped walking is difficult because biped robots usually have many degrees of freedom and a high-dimensional state space. Our solution to this problem is to modulate an observed walking pattern that gives good adjustable parameters for the biped controller. We select via-points from an observed walking trajectory and use them as control actions.

We use RL methods [24] to optimize biped walking controllers. Because biped walking dynamics include contact and collision between the robot and the ground, which make precise modeling of the dynamics difficult, RL has an advantage that we do not require an exact model of the environment in advance. We are using model-based RL, where we learn an approximated model of a Poincaré map and then choose control actions based on a computed value function.

Several researchers have applied RL to biped locomotion [18], [2]. Few studies deal with a physical robot because RL methods often require large numbers of trials. The policy gradient method [25] is one of the RL methods successfully applied to learn biped walking on actual robots [1], [27]. However, [1] requires hours to learn a walking controller, and [27] requires a mechanically stable robot. On the other hand, [4] reported that a model-based approach to RL is able to accomplish tasks much faster than without using knowledge of the environment.

In this study, we use observed trajectories, such as those of humans or other robots controlled by our learning algorithm or other algorithms, as nominal trajectories. We show that the proposed method can be applied to an actual robot [see Figure 1(a)]. First we use a simulated five-link biped robot, depicted in Figure 1, to evaluate our proposed method. Physical parameters of the five-link simulated robot in Table 1 are selected to model the actual biped robot fixed to a boom that constrains the robot to the sagittal plane. Our biped has a short torso and round feet without ankle joints. Due to the round feet and lack of ankles, controlling biped walking trajectories with the popular zero moment point (ZMP) approach [9] is difficult or impossible.

The article is organized as follows. In the following section, we introduce our RL method for biped walking. Then we show simulation results, approximately analyze the local stability of the learned controller, an implementation of the proposed method on the actual robot, and demonstrate that the robot acquires a successful walking pattern within 100 trials.

BY JUN MORIMOTO AND CHRISTOPHER G. ATKESON

1070-9932/07/$25.00©2007 IEEE

Poincaré-Map-Based RL for Biped Locomotion

We improve biped walking controllers based on an approximated Poincaré map using a model-based RL framework [4], [24]. The Poincaré map represents the locus of intersection of the biped trajectory with a hyperplane in the full state space (see Appendix I). 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 $\phi = \pi/2\pi$ and $\phi = 3\pi/2$ (see Figure 2), where we define phase $\phi = 0$ as the left foot touchdown.

The input state is defined as $\mathbf{x} = (\psi, \dot{\psi})$, where ψ denotes the angle of the inverted pendulum dynamics represented by the dynamics of the center of gravity (CoG) and the center of pressure (CoP) (see Figure 3). We use a human walking pattern in [6] as the nominal trajectory (see Figure 4). The action of the robot $\mathbf{u} = \theta^{\text{act}}$ modulates the via-points of the nominal trajectories:

$$\theta_{vp} = \bar{\theta}_{vp} + \theta^{\text{act}}, \tag{1}$$

where $\bar{\theta}_{vp}$ denotes the nominal value of the selected via-points. In this study, we manually selected a via-point of the knee joint to change foot placement. If we assume that foot exchange occurs instantaneously, the mass of the swing leg is negligible, the biped robot has point feet, the distance between the CoG and the CoP is fixed, and the angle ψ is small, the approximate biped robot dynamics are those of an inverted pendulum model:

$$\ddot{\psi} = \frac{g}{l}\psi, \tag{2}$$

where l represents the fixed distance between the CoG and the CoP and g denotes the acceleration due to gravity. Because we can modulate the angle ψ by changing the foot placement, we can expect that we can control the state $\mathbf{x} = (\psi, \dot{\psi})$ at the Poincaré section by modulating the selected via-point. Although we modulate the selected via-point according to the pendulum state, behavior of the biped model does not need to be precisely approximated by the pendulum model since our RL method do not explicitly use the model (2).

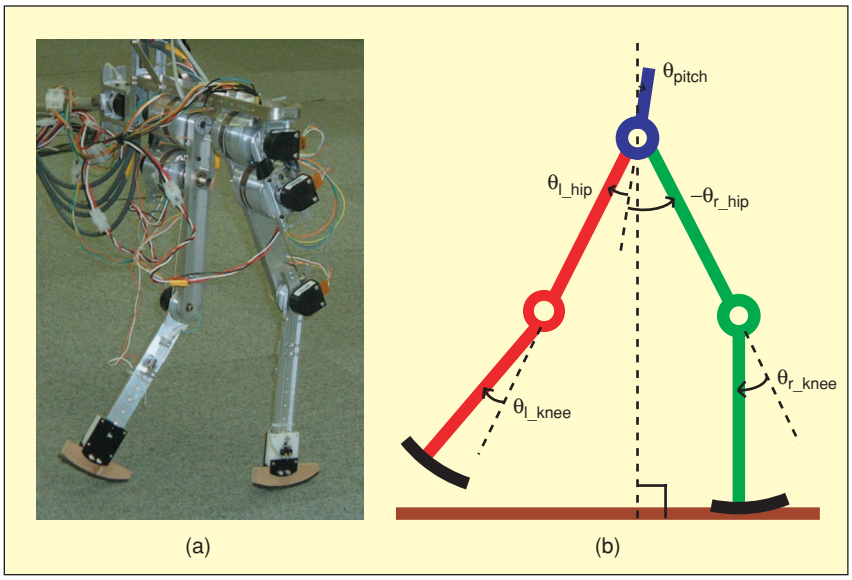

Figure 1. *(a) Five link biped robot. (b) Simulated biped robot model.*

Table 1. 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 · m^2])	1.0	6.9	1.4

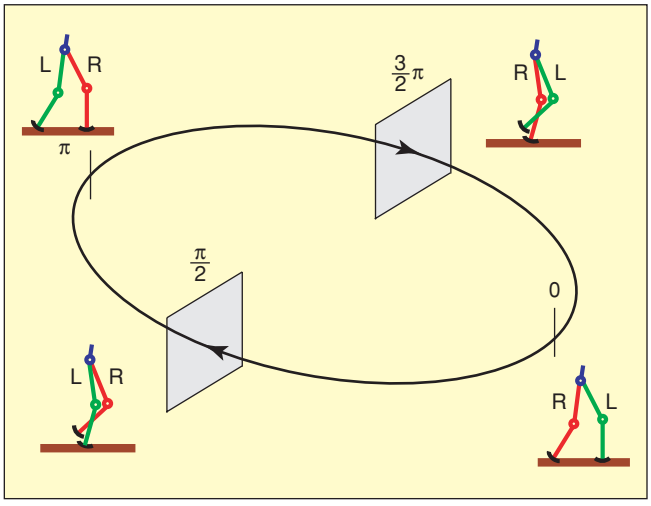

Figure 2. *Biped walking cycle: we update parameters and select actions at Poincaré sections at phase $\phi = \pi/2$ and $\phi = 3\pi/2$. L: left leg, R: right leg.*

Representation of Biped Walking Trajectories and the Low-Level Controller

We interpolated trajectories between the via-points by using the minimum jerk criteria [7], [29] (see Figure 4 and Appendix II). To follow the generated target trajectories, the torque output at each joint is given by a servo controller:

$$\boldsymbol{\tau} = \mathbf{K}_p(\boldsymbol{\theta}^d(\phi) - \boldsymbol{\theta}) + \mathbf{K}_d(\dot{\boldsymbol{\theta}}^d(\phi) - \dot{\boldsymbol{\theta}}), \tag{3}$$

where $\boldsymbol{\tau}$ is the joint torque, $\boldsymbol{\theta}^d(\phi) \in \mathbf{R}^4$ is the target joint angle vector, $\mathbf{K}_p$ denotes the position gain matrix, and $\mathbf{K}_d$ denotes the velocity gain matrix.

We reset the phase [28], [19] to $\phi = 0$ at left foot touchdown and to $\phi = \pi$ at right foot touchdown.

Learning the Poincaré Map

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 $\phi = 3\pi/2$ based on the system's location in a Poincaré section at phase $\phi = \pi/2$ (see Figure 2).

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

$$\hat{\mathbf{x}}_{\frac{3\pi}{2}} = \hat{\mathbf{f}}(\mathbf{x}_{\frac{\pi}{2}}, \mathbf{u}_{\frac{\pi}{2}}; \mathbf{m}), \tag{4}$$

where the input state is defined as $\mathbf{x} = (\psi, \dot{\psi})$, and the action of the robot is defined as $\mathbf{u} = \theta^{\text{act}}$. There is a second model mapping the state at $3\pi/2$ to the state at $\pi/2$. We use receptive field weighted regression (RFWR) [22] as the function approximator (see Appendix III).

Learning the Value Function

In a stochastic RL framework, the basic goal is to find a stochastic policy, a biped controller in this study, $\pi_{\mathbf{w}}(\mathbf{x}, \mathbf{u}) = \mathrm{p}(\mathbf{u}|\mathbf{x}; \mathbf{w})$ that maximizes the expectation of the discounted accumulated reward:

$$E\{V(k)|\pi_{\mathbf{w}}\} = E\left\{\sum_{i=k+1}^{\infty} \gamma^{i-(k+1)} r(i) \middle| \pi_{\mathbf{w}}\right\}, \tag{5}$$

where $r(i)$ denotes reward, $V(k)$ is the actual return that represents an accumulated reward associated with one sample path, $\mathbf{w}$ is the parameter vector of the policy $\pi_{\mathbf{w}}$, and $\gamma (0 \leq \gamma < 1)$ is the discount factor. The value function for the policy $\pi_{\mathbf{w}}$ is defined as:

$$V^{\pi_{\mathbf{w}}}(\mathbf{x}) = E\{V(k)|\mathbf{x}(k) = \mathbf{x},\ \pi_{\mathbf{w}}\}. \tag{6}$$

In this framework, we evaluate the value function only at $\phi(k_n) = \pi/2$ and $\phi(k_n) = 3\pi/2$, where k_n denotes the time step when the nth intersection with the Poincaré section is occurred. Thus, we consider our learning framework has model-based RL for a semi-Markov decision process (SMDP) [26] (see Appendix IV). We use RFWR with a parameter vector $\mathbf{v}$ as the function approximator to represent the value function:

$$\hat{V}^{\pi_{\mathbf{w}}}(k_n) = \hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x}(k_n); \mathbf{v}). \tag{7}$$

By considering the deviation from (5), we can define the temporal difference error [24], [26]:

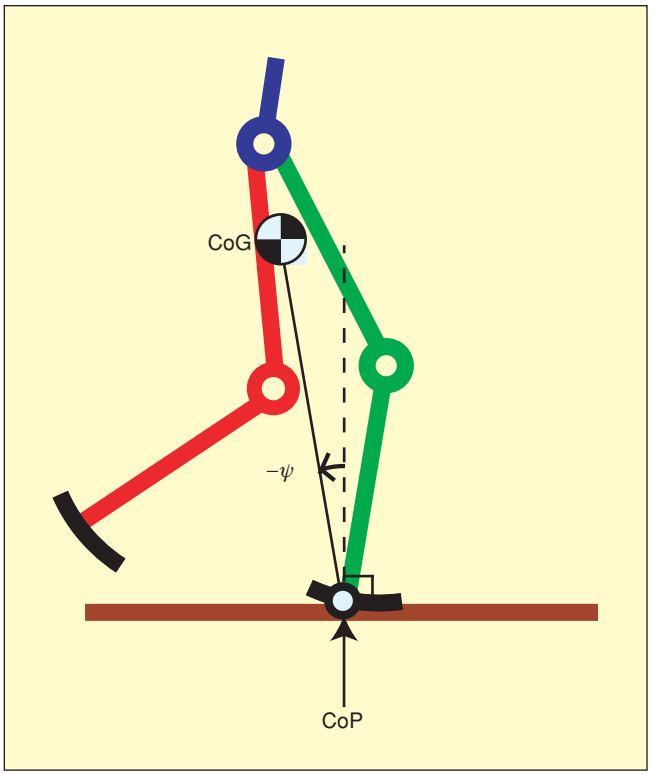

Figure 3. *The biped robot dynamics are approximated as those of an inverted pendulum and correspond to the dynamics of the center of gravity (CoG) relative to the center of pressure (CoP).*

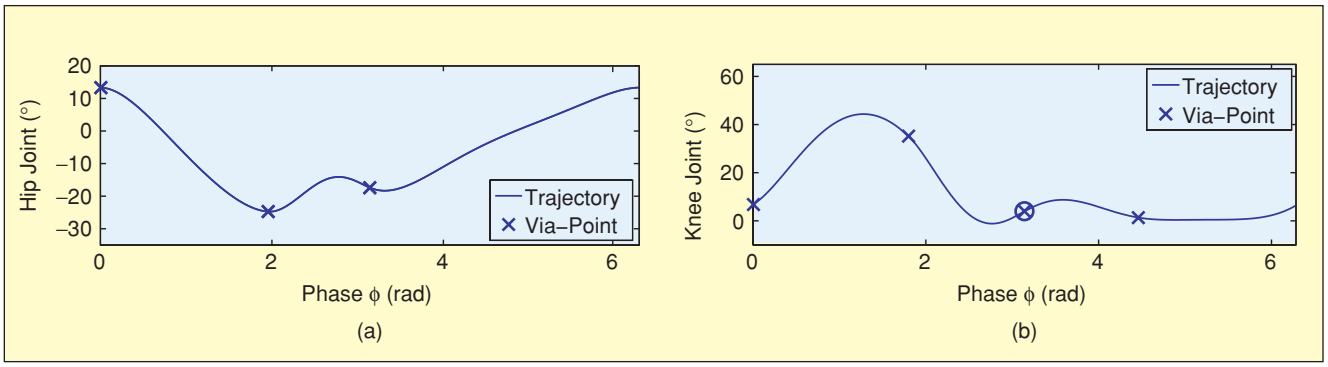

Figure 4. *Nominal joint-angle trajectories of the right leg observed from a human walking pattern and detected via-points represented by crosses (×). Nominal joint-angle trajectories for the left leg have π radians phase difference. A manually selected via-point represented by a circle (○) is modulated by the control output θ^{act} [see (1)] to change the foot placement. Note that the amplitude of the human walking pattern is multiplied by 0.7.*

$$\delta = \sum_{j=k_n+1}^{k_{n+1}} \gamma^{j-(k_n+1)} r(j) + \gamma^{k_{n+1}-k_n} \hat{V}^{\pi_{\mathbf{w}}}(k_{n+1}) - \hat{V}^{\pi_{\mathbf{w}}}(k_n). \quad (8)$$

To update the value function, a desired output for RFWR is:

$$\hat{V}_d^{\pi_{\mathbf{w}}}(\mathbf{x}(k_n)) = \hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x}(k_n); \mathbf{v}) + \alpha\delta, \quad (9)$$

where α is a learning rate.

Learning a Policy for Biped Locomotion

We use a stochastic policy to generate exploratory action. The policy is represented by a Gaussian distribution:

$$\pi_{\mathbf{w}}(\mathbf{x}, \mathbf{u}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}; \mathbf{w}), \Sigma), \quad (10)$$

where $\boldsymbol{\mu}(\mathbf{x}; \mathbf{w}) \in U \subset \mathbf{R}^m$ denotes the mean of the model, which is represented by RFWR. Σ denotes the covariance matrix $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_m^2)$. In this study, the number of outputs is $m = 1$.

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

1) Derive the gradient of the learned Poincaré map model: $(\partial \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})/\partial \mathbf{u})|_{\mathbf{x}=\mathbf{x}(k_n), \mathbf{u}=\mathbf{u}(k_n)}$.
2) Derive the gradient of the approximated value function: $(\partial \hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x})/\partial \mathbf{x})|_{\mathbf{x}=\mathbf{x}(k_{n+1})}$.
3) To update the policy parameter $\mathbf{w}$, compute a desired output for RFWR as:

$$\boldsymbol{\mu}_d(\mathbf{x}(k_n)) = \boldsymbol{\mu}(\mathbf{x}(k_n); \mathbf{w}) + \beta \frac{\partial \hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}, \quad (11)$$

where β is a learning rate.

Since the goal of the learning system is to increase the discounted accumulated reward, (5), by updating the parameter of the policy $\mathbf{w}$, we use the gradient direction of the approximated value function $\hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x})$. But because the direction $(\partial \hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x})/\partial \mathbf{x})$ is represented in the state space, we need to project the derived direction onto the action space by using the learned Poincaré map $\hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})$ as $(\partial \hat{V}^{\pi_{\mathbf{w}}}(\mathbf{x})/\partial \mathbf{x}) \times (\partial \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})/\partial \mathbf{u})$.

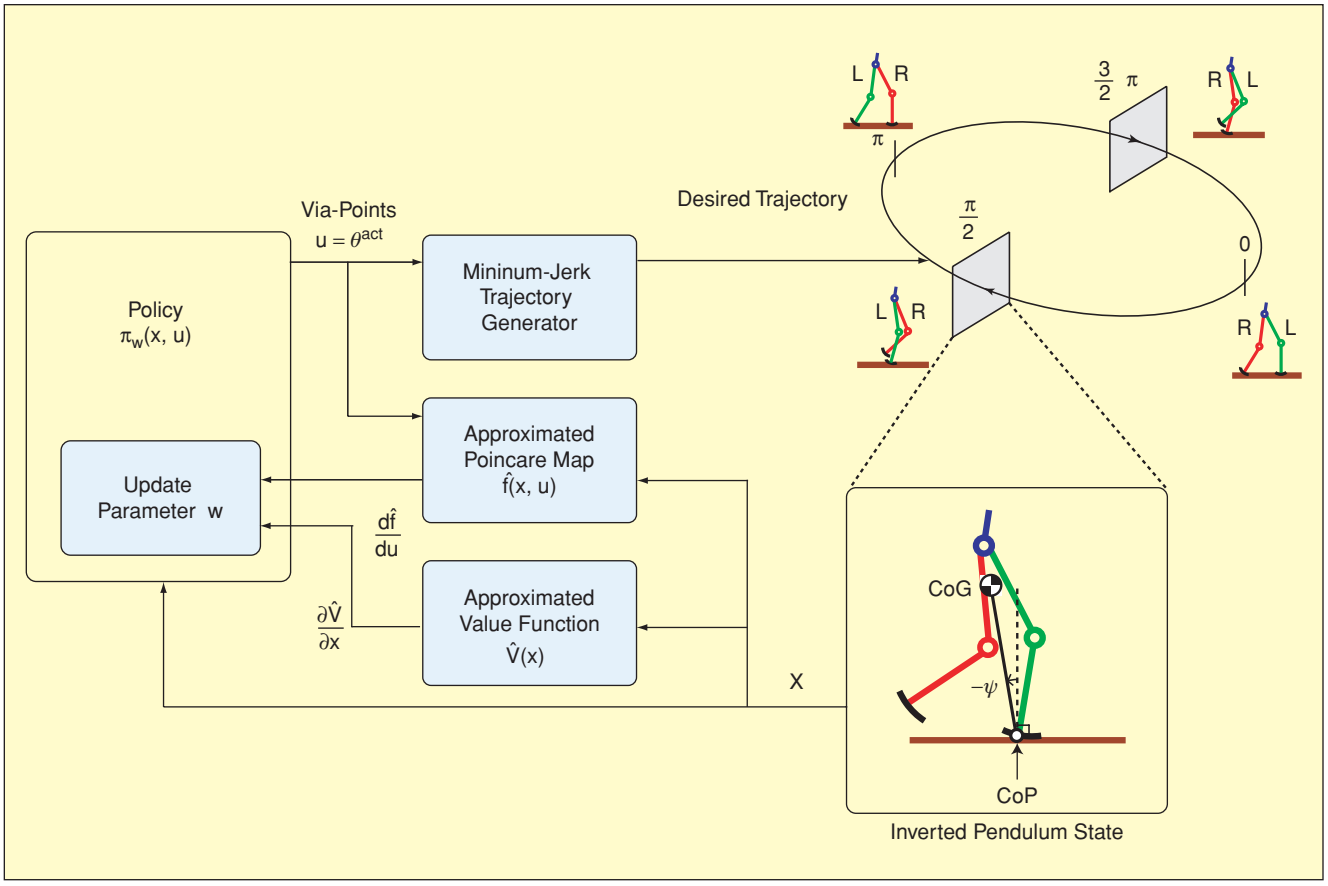

Figure 5. *Schematic diagram of Poincaré-map-based RL. The inverted pendulum state $(\psi, \dot{\psi})$ is detected at the section $\phi = \pi/2$ and $\phi = 3\pi/2$. The Poincaré map is approximated by the detected state and the state detected at the previous section. The gradient of the approximated Poincaré map $d\hat{\mathbf{f}}/d\mathbf{u}$ is estimated. The gradient of the approximated value function $\partial \hat{V}/\partial \mathbf{x}$ is estimated. The parameter vector $\mathbf{w}$ of the policy is updated by using the derived gradients. The via-points are modulated according to the current policy. The minimum-jerk trajectory is generated using the modulated via-points and used as the desired trajectory for the biped robot.*

We can consider the output $\mathbf{u}(k_n)$ as an *option* in the SMDP [26] initiated in state $\mathbf{x}(k_n)$ at time k_n when $\phi(k_n) = \pi/2$ (or $\phi(k_n) = 3\pi/2$), and it terminates at time k_{n+1} when $\phi(k_{n+1}) = 3\pi/2$ (or $\phi(k_{n+1}) = \pi/2$).

Figure 5 shows schematic diagram of the Poincaré-map-based RL method.

The learning performance can depend on the observed trajectories. Since we only evaluate current performance and update parameters at the Poincaré section, the amount of calculation can be reduced, but the observed trajectories are required to be good enough to generate at least two steps initially so that our algorithm can evaluate and update the parameters. We scaled the human walking pattern in order to satisfy this requirement (see Figure 4).

Local Stability Analysis of a Learned Biped Controller

We approximately analyze the local stability of the acquired policy around a fixed point $\mathbf{x}^*$ in terms of the Poincaré map, the mapping from a Poincaré section at phase $\phi = \pi/2$ to phase $\phi = 3\pi/2$ (see Figure 2).

We estimate the Jacobian matrix of the Poincaré map at the Poincaré sections,and check if the maximum eigenvalue is inside of the unit circle [3], [8]. Because we use differentiable functions as function approximators, we can estimate the Jacobian matrix J as follows:

$$J = \left.\frac{d\mathbf{f}}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}}. \tag{12}$$

Simulation

We applied the proposed method to the five-link simulated robot [see Figure 1(b)]. We used a manually generated initial step to get the pattern started. We set the walking period to $T = 1.0$ s. A trial is terminated after 20 steps or after the robot falls. Figure 6(a) shows the walking pattern before learning.

We numerically integrate the dynamics of the biped model with a time step of $\Delta t = 0.001$s. Each element of the diagonal position gain matrix $\mathbf{K}_p$ in (3) is set to 4.0 except for the knee joint of the stance leg, which is set to 9.0, and each element of the diagonal velocity gain matrix $\mathbf{K}_d$ is set to 0.1.

The control cost is defined as $r(k) = -(1/4)\sum_{j=1}^{4} 0.2\tau_j^2(k)\Delta t$. On each transition from phase $\phi = \pi/2$ (or $\phi = 3\pi/2$) to phase $\phi = 3\pi/2$ (or $\phi = \pi/2$), the robot gets a positive reward (0.1). If the height of the body goes below 0.38 m, the robot is given a negative reward (-1) and the trial is terminated.

We varied the element of the covariance matrix Σ in (10) according to the trial as $\sigma = 0.2\,((50 - N_{\text{trial}})/50)+0.002$ for $N_{\text{trial}} \leq 50$ and $\sigma = 0.002$ for $N_{\text{trial}} > 50$, where N_{trial} denotes the number of trials. The learning rates are set as $\alpha = 0.2$ (9) and $\beta = 0.1$ (11).

A successful trial occurred when the robot continuously walked for more than 20 steps. We use the accumulated reward of each transition from phase $\phi = \pi/2$ (or $\phi = 3\pi/2$) to phase $\phi = 3\pi/2$ (or $\phi = \pi/2$), averaged over the number of the transitions, to show progress of the learning process. Figure 7 shows the accumulated reward at each trial averaged over ten simulation runs. Walking controllers were acquired within 100 trials.

We improve biped walking controllers based on an approximated Poincaré map using a model-based RL framework.

The shape of the approximated values function is shown in Figure 8(a). The approximate value function has two local minima. One is located at a position corresponding to a negative angle ψ and negative angular velocity $\dot{\psi}$ because this state leads the robot to fall backward. The other is located at a position corresponding to a positive value of ψ and a positive angular velocity $\dot{\psi}$ because this state leads the robot to fall forward. The maximum value of the value function is located at

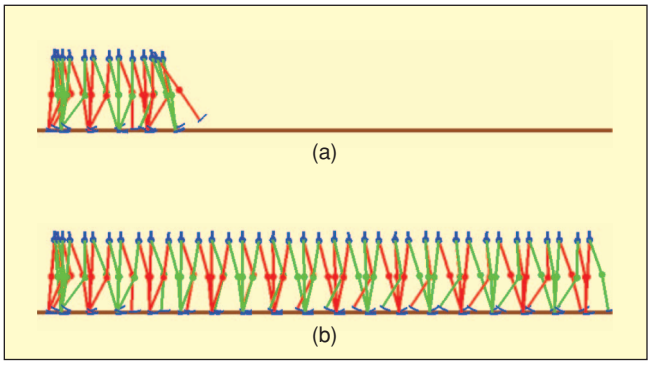

Figure 6. *Acquired biped walking pattern. (a) Before learning. (b) After learning.*

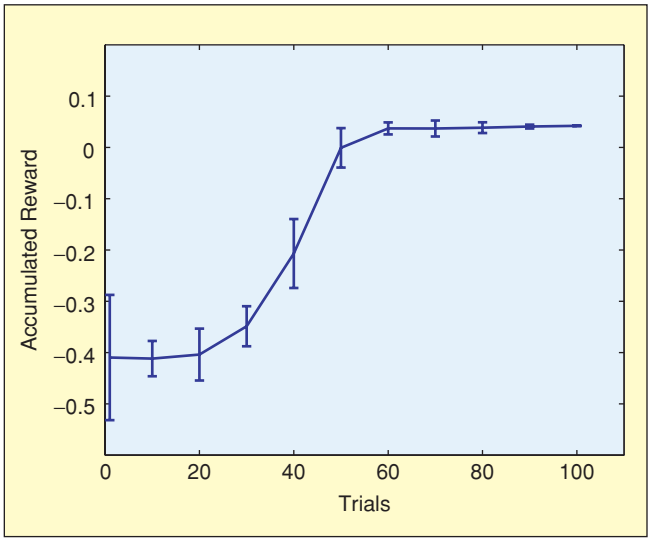

Figure 7. *Accumulated reward at each trial: We averaged ten simulation runs and present standard deviations as well.*

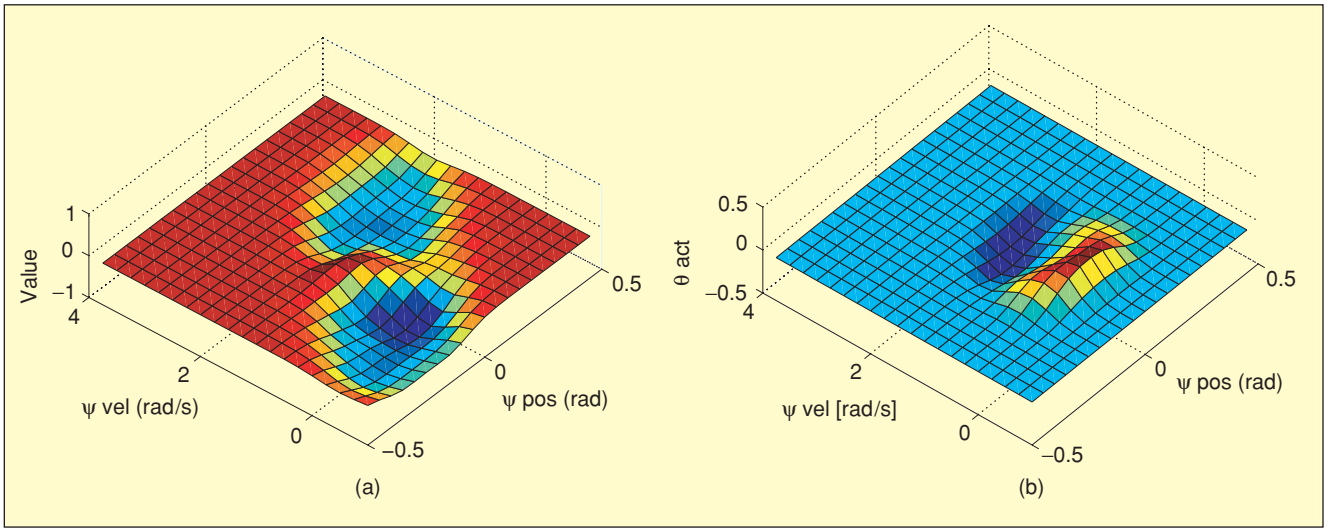

Figure 8. *(a) Shape of acquired value function. (b) Shape of acquired policy.*

a position corresponding to a negative value of ψ and a positive angular velocity $\dot{\psi}$, which leads to successful walking.

The shape of the acquired policy is shown in Figure 8(b). The policy increases θ^{act} [see (1)] to bend the knee if the angular velocity $\dot{\psi}$ is too small for a successful walk. By bending the knee, the angle ψ increases on touchdown. As the angle ψ increases, the acceleration of the angle $\ddot{\psi}$ increases [see (2)]. As a result, successful walking is achieved by using the learned policy.

The learned walking pattern is shown in Figure 6(b). Figure 9(a) shows a phase diagram of a successful walking pattern in the state space $\mathbf{x} = (\psi, \dot{\psi})$ after learning. A gradual increase of the walking speed can be observed. Figure 9(b) shows the loci of the walking trajectory in the Poincaré section. The walking trajectory after learning converges to to a fixed point in the Poincaré section after a few steps.

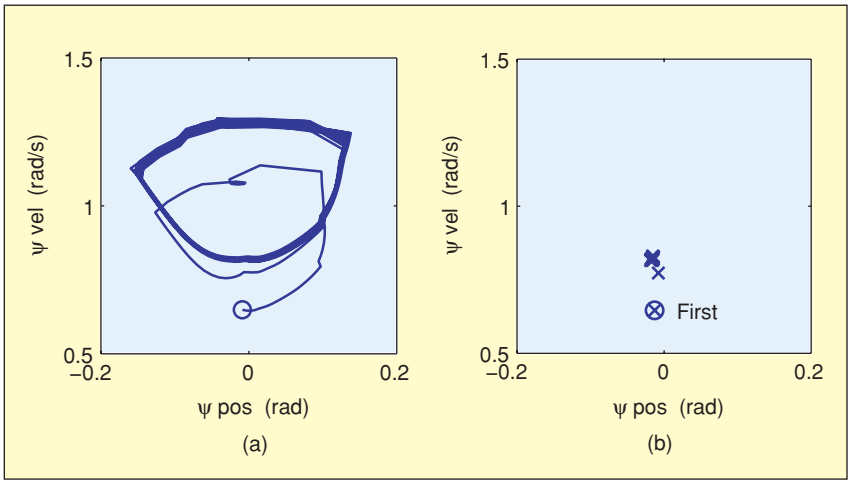

Figure 9. *(a) Phase diagram in the state space $(\psi, \dot{\psi})$. ○ represents the state at the first return to the section. (b) Loci of the walking trajectory at the Poincaré section. ○ represents the first locus of the walking trajectory that passes through the section.*

So far, we have used the round foot model depicted in Figure 10(a). Because many biped walking robots, including humanoid robots, have flat feet, we also applied the proposed method to a simulated robot that has flat feet [see Figure 10(b)]. Figure 11(a) shows the walking pattern before learning. We used a different initial step for the flat footed model. The other simulation settings were the same as the round footed model. Figure 11(b) shows the walking pattern generated by an acquired policy. The robot model could learn biped walking controllers in different environments (i.e., with two different feet shapes) without explicit knowledge of the foot shape. This result demonstrates an advantage of using RL in the design of controllers for biped walking.

We verified that the maximum eigenvalue of the Jacobian matrix J in (12) is always inside of the unit circle after 100 trials in ten simulation runs. Equivalently, we showed that the simulated biped robot could acquire locally stable controllers by using our proposed learning framework. The fixed point $\mathbf{x}^*$ is estimated by averaging the state $\mathbf{x}$ in the Poincaré section for ten steps starting at the tenth step.

Implementation on the Physical Robot

We applied the proposed model-based RL scheme to a biped robot [see Figure 1(a)]. We used a walking pattern generated by a previously designed state machine controller [21] run on the same robot as the nominal walking pattern (see Figure 12). We created via-points in this nominal walking pattern and manually selected via-points that correspond to foot placement.

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

We changed the covariance matrix Σ in (10) depending on the trial number: $\sigma = 0.1\,((50 - N_{\text{trial}})/50) + 0.01$ for $N_{\text{trial}} \leq 50$ and $\sigma = 0.01$ for $N_{\text{trial}} > 50$. We set the walking period to $T = 0.84$ s. A trial was terminated after 30 steps or after the robot falls. We use the predesigned state machine for the initial six steps.

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

Figure 13 shows a biped walking pattern before learning. The robot fell over with the nominal walking pattern. Figure 14 shows a biped walking pattern after learning. After 100 trials, the robot acquired a policy that generated a biped walking pattern. We applied the acquired controller to a different ground surface. Even on a metal surface, the robot successfully walked using the learned biped walking policy (see Figure 15).

Figure 16 shows joint angle trajectories of the actual robot. The robot generated a stable periodic pattern after 100 trials. During each step, the robot straightened its leg, which is uncommon in the popular ZMP approach due to the necessity of avoiding singularities.

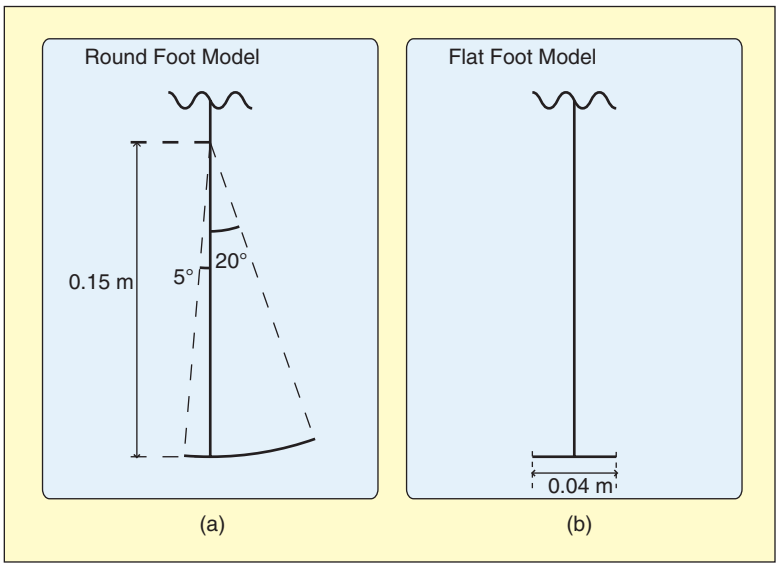

Figure 10. *Feet shape. (a) Round foot model. (b) Flat foot model.*

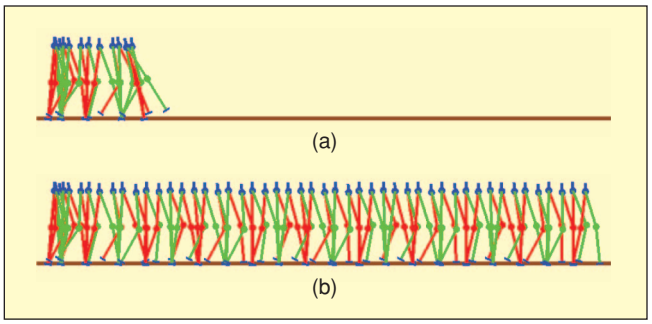

Figure 11. *Acquired biped walking pattern with flat feet. (a) Before learning. (b) After learning.*

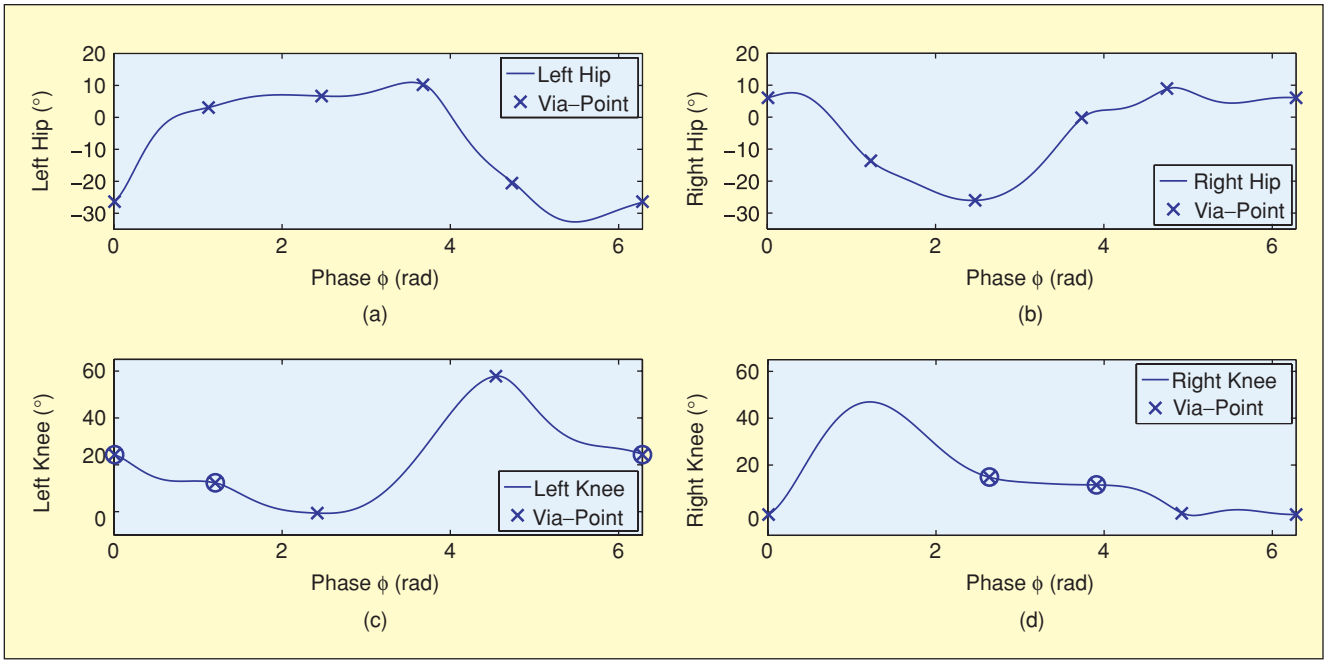

Figure 12. *Nominal joint angle trajectories and detected via-points represented by crosses (×). Manually selected via-points represented by circles (○) are modulated by control output θ^{act}. Each selected via-point is equally modulated by the control output θ^{act}.*

Because biped walking dynamics include contact and collision between the robot and the ground, which make precise modeling of the dynamics difficult, RL has an advantage that we do not require an exact model of the environment in advance.

Figure 17 shows the accumulated reward at each trial using the real robot. The robot learned a walking controller within 100 trials.

Conclusions

In this study, we proposed Poincaré-map-based RL and applied the proposed method to biped locomotion. The simulated robot acquired a biped walking controller using the 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.

Automatic selection of the via-points to be used as control actions is part of our future work. We will develop a method to evaluate the contribution of each via-point to predict the state at next Poincaré section. Automatic relevance determination (ARD) [20] is a candidate approach to select relevant via-points to predict the next state.

In our previous work, we proposed a trajectory optimization method for biped locomotion [15], [16] based on differential dynamic programming [5], [11]. We are now considering combining this trajectory optimization method with the proposed RL method.

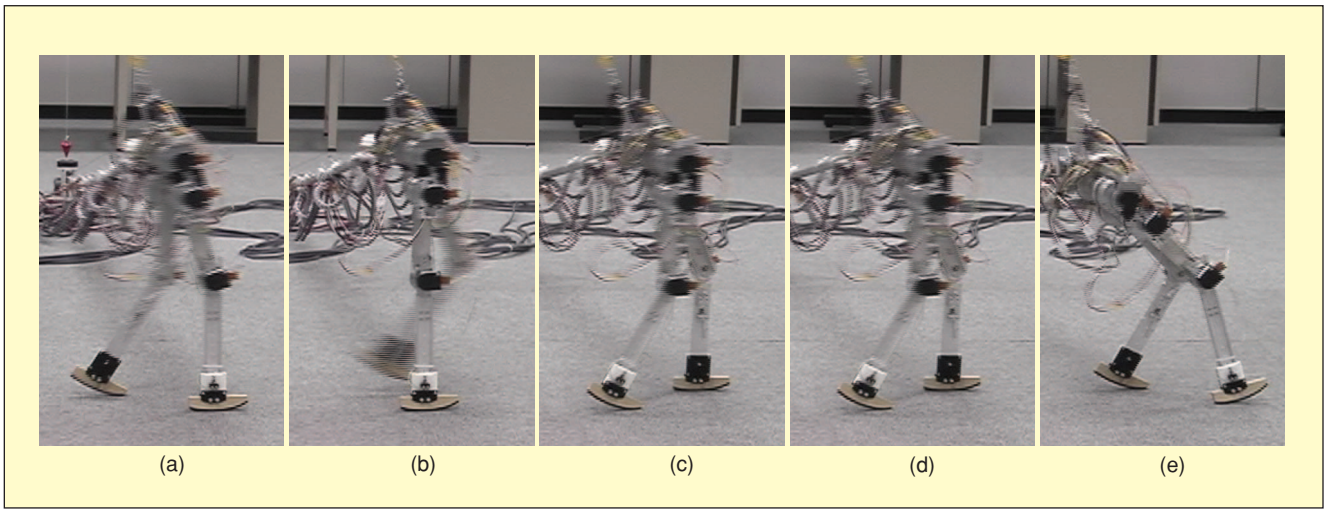

Figure 13. *Biped walking pattern before learning.*

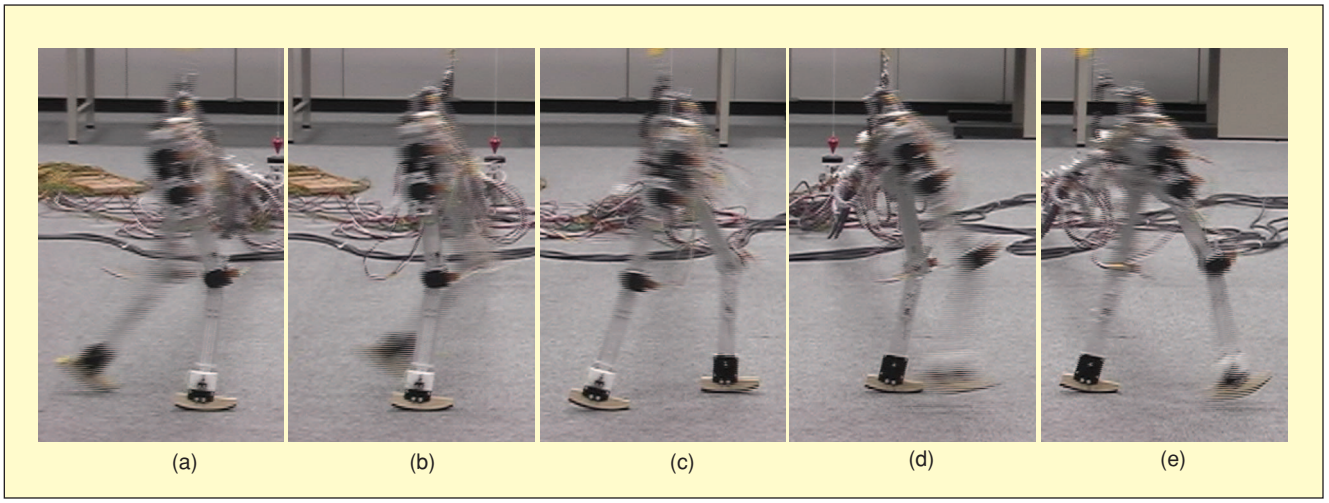

Figure 14. *Biped walking pattern after learning.*

Appendix I

Poincaré-Maps

Here we consider the dynamics $\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z})$ of a state vector $\mathbf{z} \in \mathbf{R}^n$. The Poincaré map is a mapping from an $n-1$ dimensional surface S defined in the state space to itself [23]. If $\mathbf{z}(k_n) \in S$ is the n-th intersection, then the Poincaré map $\mathbf{f}$ is defined by

$$\mathbf{z}(k_{n+1}) = \mathbf{f}(\mathbf{z}(k_n)). \tag{13}$$

Suppose that $\mathbf{z}^*$ is a fixed point of $\mathbf{g}$. We can evaluate the local stability of the periodic orbit near this fixed point. In our study, we assumed that we can represent the biped walking dynamics by using simple pendulum dynamics with the dynamics of the phase ϕ of the controller. The state of these dynamics were $\mathbf{z} = (\psi, \dot{\psi}, \phi)$ and the section S was defined at $\phi = (\pi/2)$ and $\phi = (3\pi/2)$. Therefore, the state of the learning system was $\mathbf{x} = (\psi, \dot{\psi})$.

Appendix II

Minimum-Jerk Trajectories and Via-Point Detection

We used minimum-jerk trajectories to represent our desired joint trajectories. A minimum-jerk trajectory is a fifth-order spline function that minimizes the criterion:

$$\int_{t_s}^{t_f} \left(\frac{d^3\theta(t)}{dt^3} \right)^2 dt, \tag{14}$$

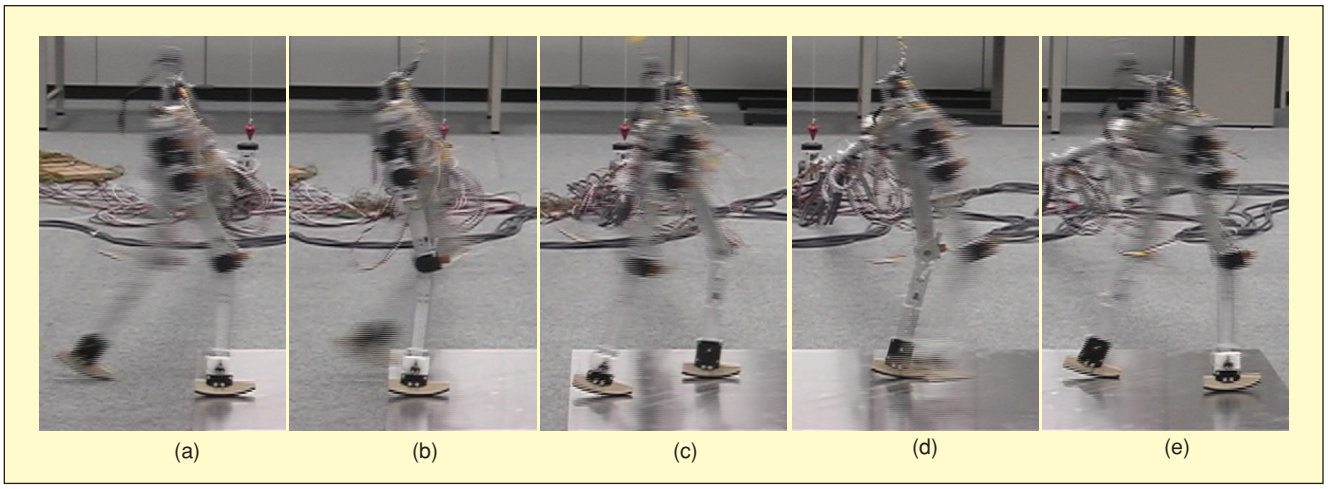

Figure 15. *Biped walking on a metal surface.*

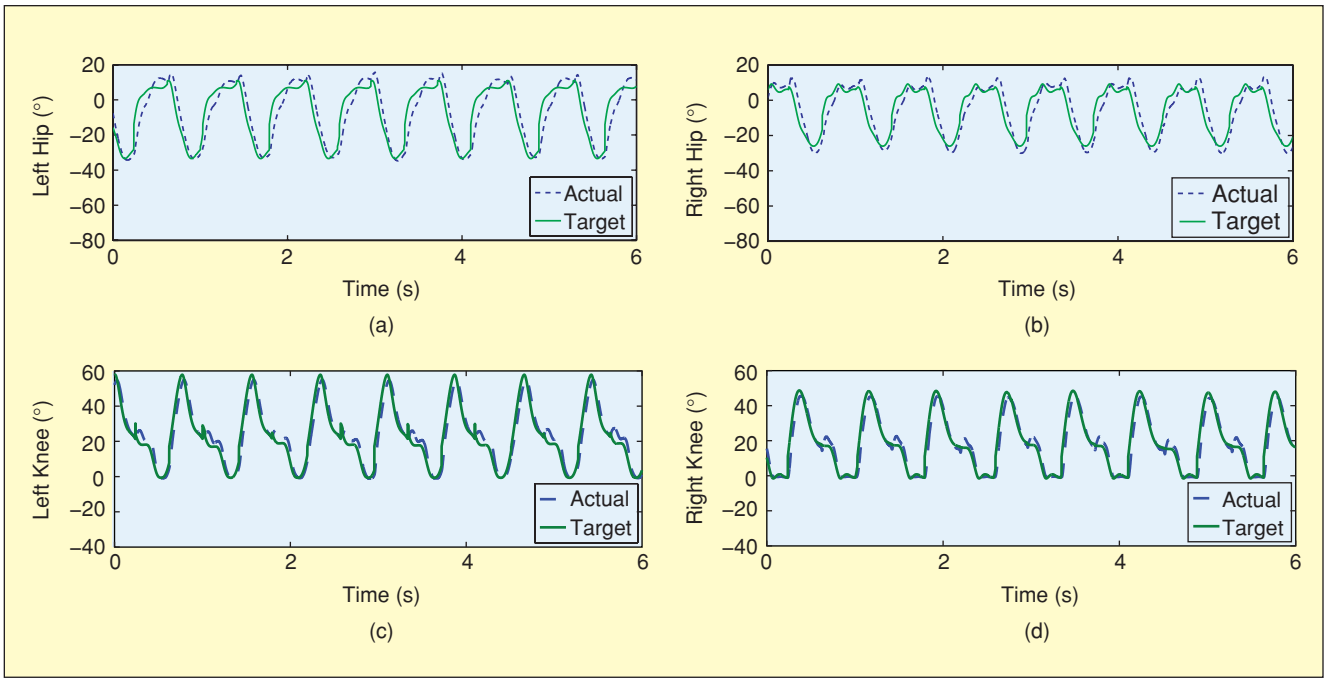

Figure 16. *Joint angle trajectories after learning on the real robot.*

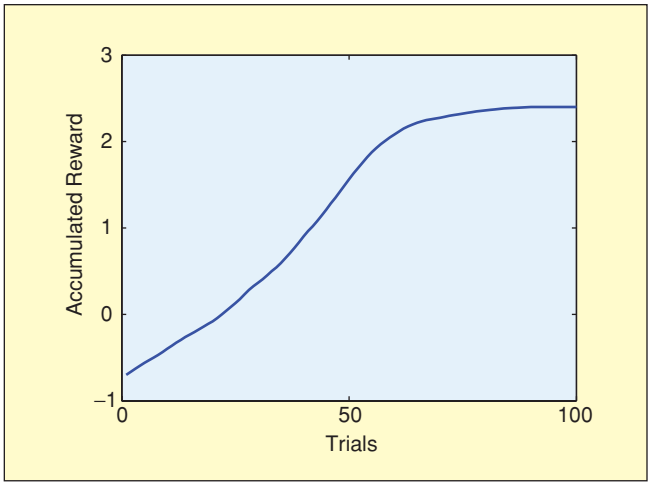

Figure 17. *Accumulated reward at each trial using the physical robot. We filtered the data with a moving average of 20 trials.*

where jerk was defined as third order derivative of a trajectory $d^3\theta(t)/dt^3$, t_s is start time and t_f is end time of the trajectory. This minimum-jerk criterion was originally defined in Cartesian space and used to fit human arm movements [7]. The minimum-jerk criterion defined in joint space was used to generate arm movement of a serial-link robot for the *kendama* and tennis serve tasks [14], [12]. Application of minimum-jerk trajectories with via-point to RL was studied in [13].

The minimum-jerk trajectory has the form:

$$x(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5, \quad (15)$$
$$\dot{x}(t) = c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4, \quad (16)$$
$$\ddot{x}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3, \quad (17)$$

where parameters c_i $(i = 0, \ldots, 5)$ are derived from the six boundary conditions at the initial and terminal times.

In this study, via-points detected from an observed trajectory represent initial and terminal boundary conditions.

The via-points are detected by the algorithm proposed in [29]:

1) Detect two via-points from the initial and terminal point of observed trajectory.
2) Generate a minimum-jerk trajectory using the detected via-points.
3) Check the error between the minimum-jerk trajectory and the observed trajectory at each point. If the error is smaller than a given threshold, terminate the algorithm.
4) Detect the time that has the maximum error and use the state $x(t), \dot{x}(t), \ddot{x}(t)$ at the detected time t as the newly detected via-point. Go to step 2.

Appendix III

Function Approximator

We used Receptive Field Weighted Regression (RFWR) [22] as the function approximator for the policy, the value function, and the estimated Poincaré map. We approximate a target function $g(\mathbf{x})$ with

$$\hat{g}(\mathbf{x}) = \frac{\sum_{i=1}^{N_b} a_i(\mathbf{x}) h_i(\mathbf{x})}{\sum_{i=1}^{N_b} a_i(\mathbf{x})}, \quad (18)$$
$$h_i(\mathbf{x}) = \mathbf{w}_i^T \tilde{\mathbf{x}}_i, \quad (19)$$
$$a_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_i)^T \mathbf{D}_i (\mathbf{x} - \mathbf{c}_i)\right), \quad (20)$$

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

$$w_i \leftarrow w_i + a_i(x) P_i \tilde{x}_i (g(x) - h_i(x)), \quad (21)$$

where

$$\mathbf{P}_i \leftarrow \frac{1}{\lambda}\left(\mathbf{P}_i - \frac{\mathbf{P}_i \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T \mathbf{P}_i}{\frac{\lambda}{a_i} + \tilde{\mathbf{x}}_i^T \mathbf{P}_i \tilde{\mathbf{x}}_i}\right), \quad (22)$$

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.,

$$\max_i a_i(\mathbf{x}) < a_{\min}, \quad (23)$$

where $a_{\min} = \exp(-\frac{1}{2})$. We initially align basis functions $a_i(\mathbf{x})$ at even intervals in each dimension of the input space $\mathbf{x} = (\psi, \dot{\psi})$ (see Figure 3). The ranges of the input space's states are -0.5 rad $\le \psi \le 0.5$ rad and -1.0 rad/sec $\le \dot{\psi} \le 4.0$ rad/s. Initial number of basis functions is 225(=15 × 15) for approximating the policy and the value function. We put one basis function at the origin for approximating the Poincaré map. We set the distance metric $\mathbf{D}_i$ to $\mathbf{D}_i = \mathrm{diag}(400, 25)$ for the policy and the value function, and $\mathbf{D}_i = \mathrm{diag}(625, 6.25, 225)$ for the Poincaré map. The centers of the basis functions $\mathbf{c}_i$ and the distance metrics of the basis functions $\mathbf{D}_i$ are fixed during learning.

Appendix IV

RL for Semi-Markov Decision Processes

Since the output of the policy is changed only at the Poincaré section, our method can be considered as a learning scheme for a policy to output a proper *option* for a Semi-Markov Decision Process (SMDP). The term *option* is introduced in [26] to represent temporally extended courses of action in SMDPs. An option consist of three components $\mathcal{I}$, η, and ρ, where $\mathcal{I}$ denotes an input set, η denotes a policy, and ρ defines terminal condition. In our study, ρ is defined at a Poincaré section, η is a modulated desired trajectory generated by minimum-jerk criteria with a PD servo controller (3). The input set $\mathcal{I}$ is also defined at a Poincaré section.

Acknowledgments

The authors would like to thank Garth Zeglin and Gen Endo for the handware development. We also would like to thank Gordon Cheng, Jun Nakanishi, Mitsuo Kawato, Kenji Doya, the reviewers, and the editor for their helpful comments. This material is based upon work supported in part by the US National Science Foundation under NSF Grant ECS-0325383.

Keywords

Biped walking, reinforcement learning, Poincaré map.

References

[1] H. Benbrahim and J. Franklin, "Biped dynamic walking using reinforcement learning," *Robot. Autonom. Syst.*, vol. 22, no. 3-4, pp. 283–302, 1997.

[2] C. Chew and G.A. Pratt, "Dynamic bipedal walking assisted by learning," *Robotica*, vol. 20, no. 5, pp. 477–491, 2002.

[3] R.Q. Van der Linde, "Passive bipedal walking with phasic muscle contraction," *Biol. Cybern.*, vol. 81, no. 3, pp. 227–237, 1999.

[4] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

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

[6] Y. Ehara and S. Yamamoto, *Introduction to Body-Dynamics—Analysis of Gait and Gait Initiation*. Bunkyo-ku, Tokyo: Ishiyaku Publishers, Inc., 2002 (in Japanese).

[7] T. Flash and N. Hogan, "The coordination of arm movements: An exper-imentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, 1985.

[8] M. Garcia, A. Chatterjee, A. Ruina, and M.J. Coleman, "The simplest walking model: stability, complexityu, and scaling," *ASME J. Biomech. Eng.*, vol. 120, no. 2, pp. 281–288, 1998.

[9] K. Hirai, M. Hirose, and T. Takenaka, "The Development of Honda Humanoid Robot," in *Proc. 1998 IEEE Int. Conf. Robotics Automation*, 1998, pp. 160–165.

[10] S. Hyon and T. Emura, "Symmetric Walking Control: Invariance and Global Stability," in *IEEE Int. Conf. Robotics Automation*, Barcelona, Spain, 2005, pp. 1456–1462.

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

[12] H. Miyamoto and M. Kawato, "A tennis serve and upswing learning robot based on dynamics optimization theory," *Neural Netw.*, vol. 11, no. 7-8, pp. 1331–1344, 1998.

[13] H. Miyamoto, J. Morimoto, K. Doya, and M. Kawato, "Reinforcement learning with via-point representation," *Neural Netw.*, vol. 17, no. 3, pp. 299–305, 2004.

[14] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato, "A Kendama Learning Robot Based on Bi-directional Theory," *Neural Netw.*, vol. 9, no. 8, pp. 1281–1302, 1996.

[15] J. Morimoto and C.G. Atkeson, "Robust low-torque biped walking using differential dynamic programming with a minimax criterion," in *Proc. 5th Int. Conf. Climbing Walking Robots*, 2002, pp. 453–459.

[16] J. Morimoto and C.G. Atkeson,"Minimax differential dynamic programming: An application to robust biped walking," in *Advances in Neural Information Processing Systems,* vol. 15, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003*,* pp. 1563–1570.

[17] K. Nagasaka, M. Inaba, and H. Inoue, "Stabilization of dynamic walk on a humanoid using torso position compliance control," in *Proc. 17th Annu. Conf. Robotics Society Japan*, 1999, pp. 1193–1194.

[18] Y. Nakamura, M. Sato, and S. Ishii, "Reinforcement learning for biped robot," in *Proc. 2nd Int. Symp. Adaptive Motion Animals Machines*, 2003, pp. ThP–II–5.

[19] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robot. Autonom. Syst.*, vol. 47, no. 2-3, pp. 79–91, 2004.

[20] R.M. Neal, "Bayesian learning for neural networks," in *Lecture Notes in Statistics*, vol. 118. New York: Springer-Verlag, 1996.

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

[22] S. Schaal and C.G. Atkeson, "Constructive incremental learning from only local information," *Neural Comput.*, vol. 10, no. 8, pp. 2047–2084, 1998.

[23] S.H. Strogatz, *Nonlinear Dynamics and Chaos*. Reading, MA: Addison-Wesley, 1994.

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

[25] 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*, vol. 12. Cambridge, MA: MIT Press, 2000, pp. 1057–1063.

[26] R.S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A Framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, no. 1-2, pp. 181–211, 1999.

[27] R. Tedrake, T.W. Zhang, and H.S. Seung, "Stochastic policy gradient reinforcement learning on a simple 3-D biped," in *Proc. 2004 IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2004, pp. 2849-2854.

[28] K. Tsuchiya, S. Aoi, and K. Tsujita, "Locomotion control of a biped locomotion robot using nonlinear oscillators," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Las Vegas, NV, 2003, pp. 1745–1750.

[29] Y. Wada and M. Kawato, "A theory for cursive handwriting based on the minimization principle," *Biol. Cybern.*, vol. 73, pp. 3–15, 1995.

[30] E.R. Westervelt, G. Buche, and J.W. Grizzle, "Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds," *Int. J. Robot. Res.*, vol. 23, no. 6, pp. 559-582, 2004.

Jun Morimoto is a researcher at ATR Computational Neuroscience Laboratories and with Computational Brain Project, ICORP, JST. He received his Ph.D. in information science from Nara Institute of Science and Technology, Nara, Japan, in 2001. He was Research Assistant at Kawato Dynamic Brain Project, ERATO, JST from 1999–2001. He was a postdoctoral fellow at the Robotics Institute, Carnegie Mellon University from 2001–2002. He joined ATR in 2002, and ICORP, JST in 2004.

Chistopher G. Atkeson is a professor in the Robotics Institute and Human-Computer Interaction Institute at CMU. He received the M.S. degree in applied mathematics (computer science) from Harvard University and the Ph.D. degree in brain and cognitive science from MIT. He joined the MIT faculty in 1986, moved to the Georgia Institute of technology College of Computing in 1994, and then joined CMU in 2000. He has received an NSF Presidential Young Investigator Award, a Sloan Research Fellowship, and a Teaching Award from the MIT Graduate Student Council.

Address for Correspondence: J. Morimoto, Japan Science and Technology Agency, ICORP, Computational Brain Project, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan, E-mail: xmorimo@atr.jp.