Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Improving humanoid locomotive performance with learnt approximated dynamics via Gaussian processes for regression

Jun Morimoto, Christopher G. Atkeson, Gen Endo, and Gordon Cheng

Abstract— **We propose to improve the locomotive performance of humanoid robots by using approximated biped stepping and walking dynamics with reinforcement learning (RL). Although RL is a useful non-linear optimizer, it is usually difficult to apply RL to real robotic systems - due to the large number of iterations required to acquire suitable policies. In this study, we first approximated the dynamics by using data from a real robot, and then applied the estimated dynamics in RL in order to improve stepping and walking policies. Gaussian processes were used to approximate the dynamics. By using Gaussian processes, we could estimate a probability distribution of a target function with a given covariance function. Thus, RL can take the uncertainty of the approximated dynamics into account throughout the learning process. We show that we can improve stepping and walking policies by using a RL method with the approximated models both in simulated and real environments. Experimental validation on a real humanoid robot of the proposed learning approach is presented.**

I. Introduction

The dynamics of biped robots include contact and collision with the ground. Modeling the interaction with the ground can be very cumbersome. Reinforcement learning (RL), which does not require a precise environmental model, can be a useful technique to improve the walking performance of biped robots. However, one drawback of using RL is that RL usually requires a large number of iterations to improve policies. Thus, applications of RL have been limited to simulation studies or small-sized real robots [1]–[5].

In this study, we directly approximate stepping and walking dynamics without explicitly identifying rigid body parameters and without a ground contact model. To approximate these dynamics, we explore the use of a Gaussian process model for regression. A Gaussian process model allows us to estimate the probability distribution of a target function with a given covariance function. Since system identification of a deterministic biped model including ground contact is difficult, we utilize a stochastic representation. By using a stochastic model, RL can take uncertainties into account through the learning process. Gaussian processes show us the reliability of the approximated function according to the density of the sampled data. This is beneficial, as it is difficult to uniformly collect data from a real robot due to unknown dynamics. We apply our learning framework to a biped simulation model (see Fig. 1(Right)) of our humanoid robot CB (see Fig. 1(Left)) [6] and a small humanoid robot (see Fig. 2). Figures 3 and 4 provide schematic diagrams of our learning framework.

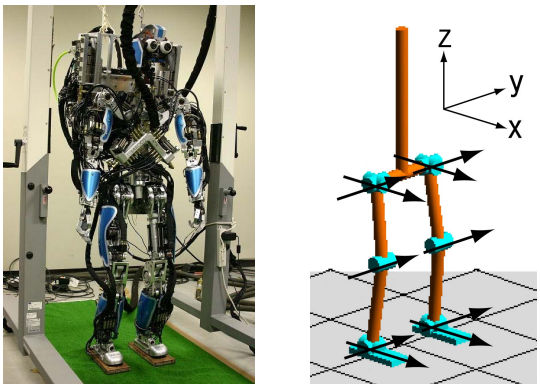


Fig. 1. (Left) Our human sized hydraulic humanoid robot CB developed by SARCOS. height: 1.59 m, total weight: 95 kg. (Right) Simplified 3D biped simulation model of our humanoid robot.

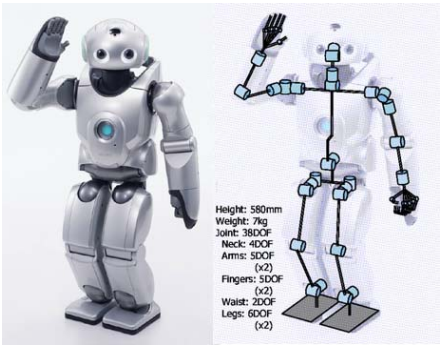


Fig. 2. Small humanoid robot used in the experiment

In our approach, we first construct a stepping and a walking controller based on our previous study [7]. The previous study proposed using the center of pressure to detect the phase of the inverted pendulum dynamics for stepping and walking (Fig. 5). We used simple periodic functions (sinusoids) as desired joint trajectories. We showed that synchronization of the desired trajectories at each joint with the inverted pendulum dynamics could generate stepping and walking movements. In this study, we modulate the amplitude of the sinusoids according to the current state of the inverted pendulum to improve locomotive performance.

In Section II, our off-line RL method, which uses approx-

J. Morimoto is with the Japan Science and Technology Agency, ICORP, Computational Brain Project, and with ATR Computational Neuroscience Laboratories. xmorimo@atr.jp

C. G. Atkeson is with the Robotics Institute, Carnegie Mellon University. cga@cs.cmu.edu

G. Endo is with the Dept. of Mechanical and Aerospace Engineering, Tokyo Institute of Technology. gendo@sms.titech.ac.jp

G. Cheng is with the Japan Science and Technology Agency, ICORP, Computational Brain Project, and with ATR Computational Neuroscience Laboratories. gordon@atr.jp

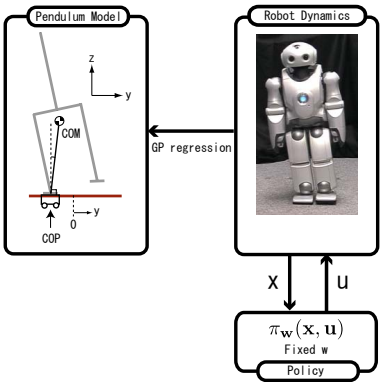


Fig. 3. Step 2 and 3 in algorithm 1. Apply the current policy with fixed parameter $\mathbf{w}$ to the actual robot dynamics and sample data. Then, generate a Gaussian process model which represents the stepping and walking dynamics.

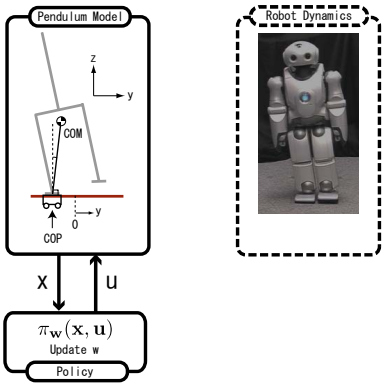


Fig. 4. Step 4 in algorithm 1. Update policy parameters $\mathbf{w}$ by applying a reinforcement learning method to the acquired inverted pendulum model represented by a Gaussian process.

imated stepping and walking dynamics is introduced. Our biped controller uses a coupled phase oscillator model to modulate the phase of sinusoidal patterns. The aim of using the coupled phase oscillator model is to synchronize periodic patterns generated by the controller with the dynamics of the robot. To use the coupled phase oscillator model, detection of the phase of the robot is needed. We introduce a method to detect the robot phase in Section III-A. We briefly explain phase coordination for biped walking in Section III-B. As in our previous study [7], we use simple sinusoidal patterns as nominal trajectories for each joint. We describe the design of the nominal trajectories for stepping movement in Section III-C, and walking movement in Section III-D. In Section IV, we explain how we applied a Gaussian process model to approximate stepping and walking dynamics. In section V, we describe implementation of a RL method for our off-line RL approach, that uses the dynamics approximated by a Gaussian process.

II. Learning framework

In this study, we consider approximate stepping and walking dynamics to improve task performance through model-based reinforcement learning [8]. A number of biped

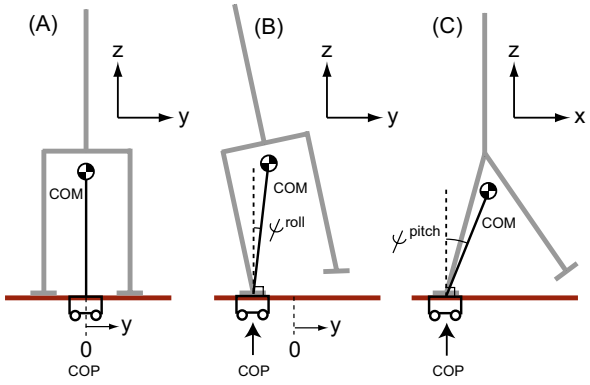


Fig. 5. Inverted pendulum model represented by the center of pressure (COP) and the center of mass (COM). ψ^{roll} denotes roll angle of the pendulum. ψ^{pitch} denotes pitch angle of the pendulum.

Algorithm 1
1. Initialize policy parameters.
2. Apply the current policy to the actual robot dynamics and sample data at the defined Poincaré section (see Fig. 3).
3. Generate a Gaussian process model which represents the stepping and walking dynamics in (1).
4. Update policy parameters by applying a reinforcement learning method to the acquired Gaussian process (see Fig. 4).
5. If the policy is not improved, terminate the iteration. Otherwise, go back to step 2.

walking studies have emphasized that humanoid robots have inverted pendulum dynamics (see Fig. 5), with the top of the pendulum at the center of mass and the base at the center of pressure. Control strategies to stabilize those dynamics have been proposed [9]–[12]. In this study, we propose using the state of the inverted pendulum dynamics as the input state for the learning system.

We assume that nominal stepping and walking controllers are provided, and our learning system improves the performance of these controllers. Since the nominal controller can generate periodic movements, we only consider the pendulum state at a Poincaré section.

For example, we consider the dynamics $\dot{\boldsymbol{\xi}} = \mathbf{g}(\boldsymbol{\xi})$ of a state vector $\boldsymbol{\xi} \in \mathbf{R}^n$. The Poincaré map is a mapping from an $n-1$ dimensional surface S defined in the state space to itself [13]. If $\boldsymbol{\xi}(k) \in S$ is the k-th intersection, then the Poincaré map $\mathbf{h}$ is defined by $\boldsymbol{\xi}(k+1) = \mathbf{h}(\boldsymbol{\xi}(k))$. In our study, we defined the section which satisfies the roll angle of the pendulum dynamics equals zero $\dot{\psi}^{roll} = 0$ (see Fig. 5).

The policy of the learning system is updated and outputs the next action only at this section. We also assume that we can represent the Poincaré map by a stochastic model. If $\mathbf{x}(k)$ is the k-th intersection, the model is defined by:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{n}(k), \tag{1}$$

where $\mathbf{x} = (\psi^{roll})$ for stepping and $\mathbf{x} = (\psi^{roll}, \psi^{pitch}, \dot{\psi}^{pitch})$ for walking (see Fig.5). $\mathbf{n}(k)$ is the noise input. $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$ represents the deterministic part of Poincaré map. We selected amplitudes of the sinusoids as the control input $\mathbf{u}(k)$ to the pendulum

dynamics (see section IV). Note that we flip the sign of the roll angle ψ^{roll} in the state vector $\mathbf{x}$ when the sign of COP in the lateral (y) direction (see Fig. 5(A)) changes so that we can use the same policy for the left stance phase and the right stance phase.

To improve task performance, we stochastically modulate the amplitude of the sinusoidal patterns according to the current policy $\pi_{\mathbf{w}}$:

$$\pi_{\mathbf{w}}(\mathbf{x}(k), \mathbf{u}(k)) = p(\mathbf{u}(k)|\mathbf{x}(k); \mathbf{w}), \tag{2}$$

where $\mathbf{w}$ is the parameter vector of the policy $\pi_{\mathbf{w}}$. In following sections, we explain how we approximate the stochastic dynamics (1), and how we acquire the control policy $\pi_{\mathbf{w}}$.

In our learning framework, we improve the approximated stochastic model and the policy iteratively (see Algorithm 1). We first sample data from a simulated, or a real robot model by using the current policy for a Gaussian process regression (see Fig. 3), then improve the policy by using the dynamics approximated by the Gaussian process (see Fig. 4).

III. Basic walking and stepping controllers

A. Phase detection of the robot dynamics

As in our previous study [7], we use the center of pressure y_{cop} and the velocity of the center of pressure $\dot{y}_{cop}$ to detect the phase of the robot dynamics:

$$\phi(\mathbf{y}_{cop}) = -\arctan\left(\frac{\dot{y}_{cop}}{y_{cop}}\right), \tag{3}$$

where $\mathbf{y}_{cop} = (y_{cop}, \dot{y}_{cop})$[1] (see Fig. 5).

B. Phase coordination

In this study, we use four oscillators with phases ϕ_c^i, where $i = 1, 2, 3, 4$. We introduce coupling between the oscillators and the phase of the robot dynamics $\phi(\mathbf{y}_{cop})$ in (3) to regulate the desired phase relationship between the oscillators:

$$\dot{\phi}_c^i = \omega_c + K_c \sin(\phi(\mathbf{y}_{cop}) - \phi_c^i + \alpha_i), \tag{4}$$

where α_i is the desired phase difference, K_c is a coupling constant, and ω_c is natural angular frequency of oscillators.

We use four different phase differences, $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{-\frac{1}{2}\pi, 0.0, \frac{1}{2}\pi, \pi\}$, to make symmetric patterns for a stepping movement with the left and right limbs (see section III-C.2), and also to make symmetric patterns for a forward movement with the left and right limbs (see section III-D).

C. Stepping controller for lateral movement

1) Side-to-side controller for lateral movement: First, we introduce a controller to generate side-to-side movement. We control the hip joints θ_{h_roll} and the ankle joints θ_{a_roll} (Fig. 6(A)) for this movement. Desired joint angles for each joint are:

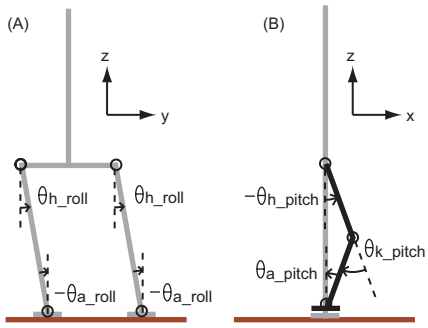


Fig. 6. Stepping controller: (A) Controller for side-to-side movement. (B) Controller for foot clearance.

$$\theta^d_{h\_roll}(\phi_c) = A_{h\_roll} \sin(\phi_c), \tag{5}$$
$$\theta^d_{a\_roll}(\phi_c) = -A_{a\_roll} \sin(\phi_c), \tag{6}$$

where A_{h_roll} and A_{a_roll} are the amplitudes of a sinusoidal function for side-to-side movements at the hip and the ankle joints, and we use an oscillator with the phase $\phi_c = \phi_c^1$.

2) Vertical foot movement to make clearance: To achieve foot clearance, we generate vertical movement of the feet (Fig. 6(B)) by using simple sinusoidal trajectories:

$$\begin{aligned}
\theta^d_{h_pitch}(\phi_c) &= (A_{pitch} + A_{step}) \sin(\phi_c) + \theta^{res}_{h_pitch}, \\
\theta^d_{k_pitch}(\phi_c) &= -2(A_{pitch} + A_{step}) \sin(\phi_c) + \theta^{res}_{k_pitch}, \\
\theta^d_{a_pitch}(\phi_c) &= -(A_{pitch} + A_{step}) \sin(\phi_c) + \theta^{res}_{a_pitch},
\end{aligned} \tag{7}$$

where A_{pitch} is the amplitude of a sinusoidal function to achieve foot clearance, $\theta^{res}_{h\_pitch}$, $\theta^{res}_{k\_pitch}$, $\theta^{res}_{a\_pitch}$ represent the rest posture of the hip, knee, and ankle joints respectively. We use the oscillator with phase $\phi_c = \phi_c^1$ for right limb movement and use the oscillator with phase $\phi_c = \phi_c^3$, which has phase difference $\phi_c^3 = \phi_c^1 + \pi$, for left limb movement. We modulate the amplitude of the sinusoidal patterns by changing A_{step} according to the current pendulum state for the stepping task.

D. Biped walking controller

To walk forward, we use an additional sinusoidal trajectory. Thus, the desired nominal trajectories for right hip and ankle pitch joints become:

$$\begin{aligned}
\theta^d_{h_pitch} &= A_{pitch} \sin(\phi_c^1) + A_{walk} \sin(\phi_c^2) + \theta^{res}_{h_pitch}, \\
\theta^d_{a_pitch} &= -A_{pitch} \sin(\phi_c^1) - A_{walk} \sin(\phi_c^2) + \theta^{res}_{a_pitch}.
\end{aligned} \tag{8}$$

We use the phase $\phi_c = \phi_c^2$, which has $\frac{1}{2}\pi$ phase difference with ϕ_c^1 for right limb. We use the phase $\phi_c = \phi_c^4$, which has π phase difference with ϕ_c^2. We use ϕ_c^3 and ϕ_c^4 for left limb instead of ϕ_c^1 and ϕ_c^2. We then modulate the amplitude of the sinusoidal patterns by changing A_{walk} according to the current pendulum state of the biped walking task.

[1] We use a simplified COP detection method introduced in [7].

IV. Gaussian process regression for stepping and walking dynamics

We use a Gaussian process [14] to approximate the dynamics in (1). Gaussian processes provide us a stochastic representation of an approximated function. With Gaussian processes for regression, the output values are sampled from a zero-mean Gaussian whose covariance matrix is a function of the input vectors:

$$(y_1, \ldots, y_N | \mathbf{z}_1, \ldots, \mathbf{z}_N) \sim \mathcal{N}(0, \mathbf{K}), \tag{9}$$

where $\mathbf{K}_{ij} = \kappa(\mathbf{z}_i, \mathbf{z}_j)$ is a covariance matrix of input vectors. Here, we used the following covariance matrix model:

$$\kappa(\mathbf{z}_i, \mathbf{z}_j) = v_0 \exp\left(-\sum_{d=1}^{N_d} \alpha_d \frac{1}{2} ||\mathbf{z}_i - \mathbf{z}_j||^2\right) + v_1 \delta_{ij}, \tag{10}$$

where δ_{ij} is Kronecker delta. v_0, v_1, and α_d are parameters for the covariance matrix. N_d denotes the number of input dimensions. These parameters can be optimized by using a type-II maximum likelihood method [14]. Bayesian prediction of an output y_{N+1} corresponding to a new input $\mathbf{z}_{N+1}$ is given as:

$$(y_{N+1} | \mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{z}_{N+1}, y_1, \ldots, y_N) \sim \mathcal{N}(\mu, \sigma^2), \tag{11}$$

where

$$\mu = \boldsymbol{\kappa}(\mathbf{z}_{N+1})^T \mathbf{K}^{-1} \mathbf{y}, \tag{12}$$

$$\sigma^2 = \boldsymbol{\kappa}(\mathbf{z}_{N+1}, \mathbf{z}_{N+1}) - \boldsymbol{\kappa}(\mathbf{z}_{N+1})^T \mathbf{K}^{-1} \boldsymbol{\kappa}(\mathbf{z}_{N+1}), \tag{13}$$

$\mathbf{y} = (y_1, \ldots, y_N)$, and $\boldsymbol{\kappa}(\mathbf{z}_{N+1}) = [\kappa(\mathbf{z}_1, \mathbf{z}_{N+1}), \ldots, \kappa(\mathbf{z}_N, \mathbf{z}_{N+1})]^T$.

The input vector $\mathbf{z}$ for the Gaussian process is composed of the current state and control input: $\mathbf{z} = (\mathbf{x}(k)^T, \mathbf{u}(k)^T)^T$. The output value y is a component of the state vector at the next intersection $\mathbf{x}(k+1)$, and the control input is the additional amplitude of the sinusoidal patterns $\mathbf{u}(k) = A_{step}(k)$ in (7) for stepping, and $\mathbf{u}(k) = A_{walk}(k)$ in (8) for biped walking. We can then estimate probabilistic model of the stepping and walking dynamics.

V. Policy improvement by using a reinforcement learning method

Here, we explain how we applied reinforcement learning to our biped stepping and walking tasks. We use a policy gradient method proposed by [15] to implement the RL framework.

The basic goal is to find a policy $\pi_{\mathbf{w}}(\mathbf{x}, \mathbf{u}) = P(\mathbf{u}|\mathbf{x}; \mathbf{w})$ that maximizes the expectation of the discounted accumulated reward:

$$E\{V(k)|\pi_{\mathbf{w}}\} = E\left\{\sum_{i=k}^{\infty} \gamma^{i-k} r(i) \middle| \pi_{\mathbf{w}}\right\}, \tag{14}$$

where r denotes reward, $V(k)$ is the actual return, $\mathbf{w}$ is the parameter vector of the policy $\pi_{\mathbf{w}}$, and γ, $0 \leq \gamma < 1$, is the discount factor.

In the policy gradient methods, we calculate the gradient direction of the expectation of the actual return with respect to parameters of a policy $\mathbf{w}$. [15] suggested that we can estimate the expectation of the gradient direction as:

$$\frac{\partial}{\partial \mathbf{w}} E\{V(0)|\pi_{\mathbf{w}}\} \approx E\left\{\sum_{k=0}^{\infty} (V(k) - \hat{V}(\mathbf{x})) \frac{\partial \ln \pi_{\mathbf{w}}}{\partial \mathbf{w}} \middle| \pi_{\mathbf{w}}\right\}, \tag{15}$$

where $\hat{V}(\mathbf{x})$ is an approximation of the value function for a policy $\pi_{\mathbf{w}}$: $V^{\pi_{\mathbf{w}}}(\mathbf{x}) = E\{V(k)|\mathbf{x}(k) = \mathbf{x},\ \pi_{\mathbf{w}}\}$.

A. Value function approximation

The value function is approximated using a normalized Gaussian network [16]:

$$\hat{V}(\mathbf{x}) = \sum_{i=1}^{N} v_i b_i(\mathbf{x}), \tag{16}$$

where v_i is a i-th parameter of the approximated value function, and N is the number of basis functions $b_i(\mathbf{x})$. An approximation error of the value function is represented by the temporal difference (TD) error [17]:

$$\delta(k) = r(k+1) + \gamma \hat{V}(\mathbf{x}(k+1)) - \hat{V}(\mathbf{x}(k)), \tag{17}$$

We update the parameters of the value function approximator using the TD(0) method [17]:

$$v_i(k+1) = v_i(k) + \alpha \delta(k) b_i(\mathbf{x}(k)), \tag{18}$$

where α is the learning rate.

B. Policy parameter update

We update the parameters of a policy $\mathbf{w}$ by using the estimated gradient direction in (15). [15] showed that we can estimate the gradient direction by using TD error:

$$E\left\{\sum_{k=0}^{\infty} (V(k) - \hat{V}(\mathbf{x}(k))) \frac{\partial \ln \pi_{\mathbf{w}}}{\partial \mathbf{w}} \middle| \pi_{\mathbf{w}}\right\} = E\left\{\sum_{k=0}^{\infty} \delta(k) \mathbf{e}(k) \middle| \pi_{\mathbf{w}}\right\}, \tag{19}$$

where $\mathbf{e}$ is the eligibility trace of the parameter $\mathbf{w}$. Then, we can update the parameter $\mathbf{w}$ as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \delta(k) \mathbf{e}(k), \tag{20}$$

where the eligibility trace is updated as:

$$\mathbf{e}(k+1) = \eta \mathbf{e}(k) + \left.\frac{\partial \ln \pi_{\mathbf{w}}(\mathbf{x}(k), \mathbf{u}(k))}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}(k)}, \tag{21}$$

where η is the decay factor for the eligibility trace. Equation (19) can be derived if the condition $\eta = \gamma$ is satisfied.

C. Biped stepping and walking policy

We construct the biped stepping and walking policies based on a normal distribution:

$$\pi_{\mathbf{w}}(\mathbf{x}, \mathbf{u}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}; \mathbf{w}^{\mu}), \boldsymbol{\Sigma}(\mathbf{x}; \mathbf{w}^{\sigma})) \tag{22}$$

where $\mathbf{u}$ is the output vector and Σ is the covariance matrix of the policy $\pi_{\mathbf{w}}$. In this study, we defined the covariance matrix as a diagonal matrix, where j-th diagonal element

is represented as σ_j. j-th element of the mean output $\boldsymbol{\mu}$ is modeled by the normalized Gaussian network:

$$\mu_j(\mathbf{x}) = \sum_{i=1}^{N} w_i^{\mu_j} b_i(\mathbf{x}). \tag{23}$$

Here, $w_i^{\mu_j}$ denotes the i-th parameter for j-th output of the policy $\pi_{\mathbf{w}}$, and N is the number of basis functions. We represent the diagonal element of the covariance matrix Σ using a sigmoid function [15]:

$$\sigma_j(\mathbf{x}) = \frac{\sigma_0}{1+\exp(-\sigma_j^w(\mathbf{x}))}, \text{ where } \sigma_j^w(\mathbf{x}) = \sum_{i=1}^{N} w_i^{\sigma_j} b_i(\mathbf{x}), \tag{24}$$

and σ_0 denotes the scaling parameter. $w_i^{\sigma_j}$ denotes the i-th parameter for the j-th diagonal element of the covariance matrix. We update the parameters by applying the update rules in (20) and (21).

VI. SIMULATION

We applied our proposed method to a simplified simulation model of our humanoid robot CB [6] (Fig. 1(Right)).

A. Improvement of biped stepping performance

We applied our proposed method to improve stepping in place. We used the amplitudes $A_{h\_roll} = 4.0^\circ$ and $A_{a\_roll} = 4.0^\circ$ for side-to-side movement, and $A_{pitch} = 6.0^\circ$ to attain foot clearance. We used the coupling constant $K_c = 10.0$. We set the natural frequency of the controller to $\omega_c = 3.5$ rad/sec. We defined the target of the stepping task to keep the state at $\psi^{roll} = 2.0^\circ$. We use a reward function:

$$r = -0.1(\psi^{roll} - 2.0)^2 \tag{25}$$

for this stepping task.

Figure 7 shows learning performance of the stepping task by using a simple implementation of the policy gradient method [15] and the proposed off-line learning approach. Thus, this demonstrate that a full dynamic model is not necessary in order to achieve sufficient results. Even for extensive simulation studies, we can save computation time[2] and still be able acquire better performance, so that we can easily generate different policies for different objective functions without running a full dynamic simulation.

Figure 8 shows policies acquired by using a simple implementation of the policy gradient method and the proposed off-line learning approach. Similar policies were acquired in which the pendulum angle ranged from $\psi^{roll} = 0.0^\circ$ to $\psi^{roll} = 4.0^\circ$.

An acquired stepping movement is shown in Fig. 9. This result suggests that the approximated dynamics can be used to improve performance of the stepping task.

[2]in this case, a stepping task.

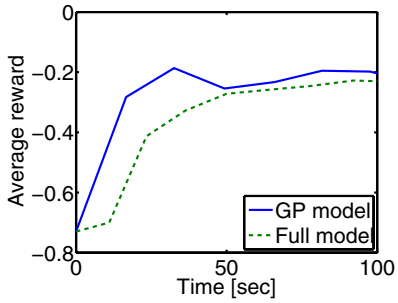


Fig. 7. Comparison of learning performance. Horizontal axis represents actual calculation time. The solid line represents the learning performance of the proposed learning method which used the acquired Gaussian process model. The dotted line represents the learning performance of the standard reinforcement learning method, which used the full dynamics simulation.

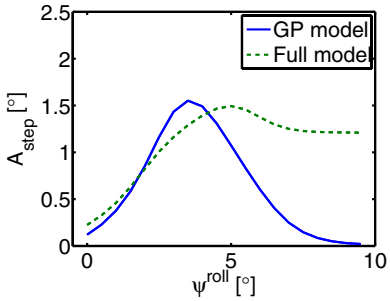


Fig. 8. Acquired stepping policy. The solid line represents an acquired policy using the proposed learning method. The dotted line represents an acquired policy using the standard reinforcement learning method. Similar policies were acquired in the range from $\psi^{roll} = 0.0^\circ$ to $\psi^{roll} = 4.0^\circ$.

B. Improvement of biped walking performance

We also applied our proposed method to improve walking performance. We used the amplitudes $A_{h\_roll} = 3.5^\circ$ and $A_{a\_roll} = 3.5^\circ$ for the side-to-side movement, and $A_{pitch} = 7.0^\circ$ to get foot clearance. We used the coupling constant $K_c = 10.0$. We set the natural frequency of the controller to $\omega_c = 3.5$ rad/sec. We only modulate the amplitude A_{walk} in (8) to generate forward movement for the biped walking task (we set $A_{step} = 0.0^\circ$ in (7)). The target of walking task is to increase the angular velocity of the pendulum $\dot{\psi}^{pitch}$ at the Poincaré section. We use a reward function:

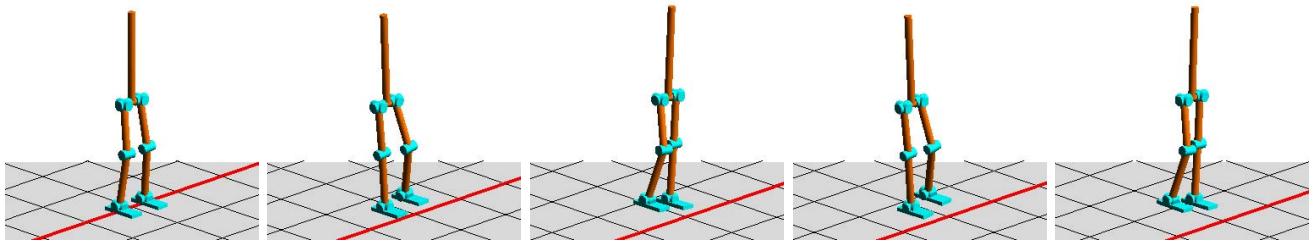
$$r = 0.1(\dot{\psi}^{pitch}) \tag{26}$$

for this biped walking task.

Figure 10 shows initial performance of the biped walking policy. Figure 11 shows walking performance after one iteration of the proposed method. This result suggests that approximated dynamics can be used to improve biped walking performance.

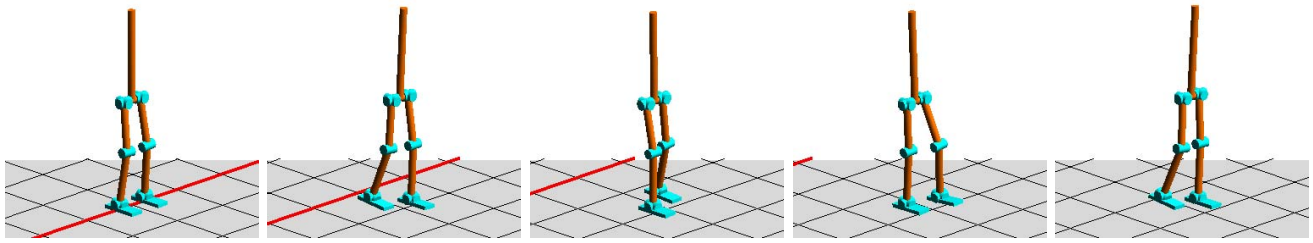
VII. EXPERIMENTAL RESULT

We applied our proposed method to a small size humanoid robot (see Fig. 2). We use the roll angle ψ^{roll} defined by the pendulum model (see Fig. 5) as the state, and modulate A_{a_roll} in (6) as the action of the learning system. We use a reward function $r = -0.1(\psi^{roll})^2$ for this stepping task.



(a) t=0.0 sec (b) t=6.0 sec (c) t=9.0 sec (d) t=12.0 sec (e) t=15.5 sec

Fig. 10. Initial walking pattern. The red line represents the starting position. Initially, the simulated robot explore around the starting position. Time proceeds from left to right.



(a) t=0.0 sec (b) t=5.0 sec (c) t=8.3 sec (d) t=11.3 sec (e) t=14.0 sec

Fig. 11. Improved walking pattern after one iteration of the proposed learning process. Walking speed is 0.14 m/sec. The red line represents the starting position. Time proceeds from left to right.

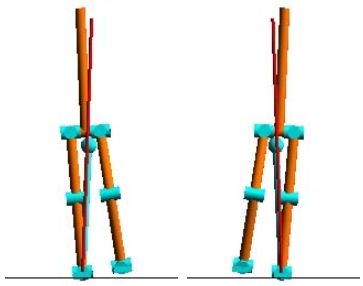


Fig. 9. Acquired stepping movement. The red thin line represents desired angle. After one iteration of the proposed learning process, the pendulum state represented by the light blue line behind the red line came close to the desired state at the Poincaré section. The light blue sphere represents the center of mass.

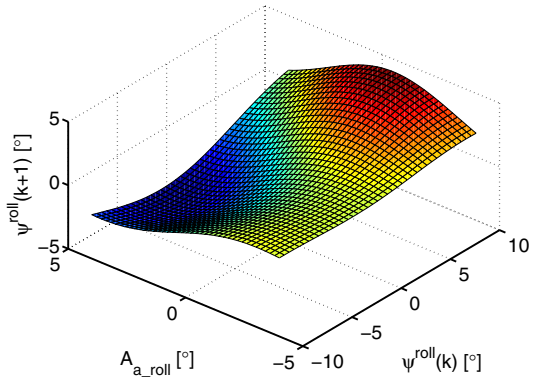


Fig. 12. Approximated stepping dynamics of the small size humanoid robot by a Gaussian process. The input vector is defined as $\mathbf{z} = (\psi^{roll}(k), A_{a\_roll}(k))$, and the output is define as $y = (\psi^{roll}(k+1))$ (see (12)).

Figure 12 shows approximated stepping dynamics of the small size humanoid robot by a Gaussian process. Here we define the input vector as $\mathbf{z} = (\psi^{roll}(k), A_{a\_roll}(k))$ and the output as $y = (\psi^{roll}(k+1))$ (see (12)). We apply the reinforcement learning algorithm to this acquired stepping dynamics to improve stepping performance.

Figure 13 shows the roll angle ψ^{roll} at the Poincaré section $\dot{\psi}^{roll} = 0$. This result suggests that a stepping policy was acquired by using our proposed method, and it can keep the roll state ψ^{roll} around the desired angle ($0.0°$). An average angle from 10 to 40 steps were $0.12°$.

Figure 14 shows the acquired stepping movement of the real robot after one iteration of the proposed learning process, applied to the real environment. We will soon apply our proposed method to the biped walking task in the real environment.

VIII. DISCUSSION

We proposed using approximated biped stepping and walking dynamics for Reinforcement Learning (RL) to improve task performance. In this study, we first approximated the stepping and walking dynamics by using collected data from a simulated model or a real robot, then use the approximated dynamics for RL to improve stepping and walking policies. We explored using a Gaussian process to approximate the dynamics. By using a Gaussian process, we can estimate a probability distribution of a target dynamics with a given covariance function. We showed that we could improve stepping and walking policies by using a RL method with approximated models both in simulated and real environ-

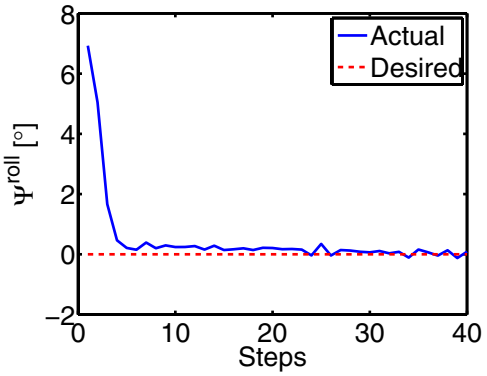


Fig. 13. The roll angle ψ^{roll} at the Poincaré section $\dot{\psi}^{roll} = 0$ (solid line). The dotted line represents the desired angle for this stepping task. We used a policy acquired by the proposed learning method.

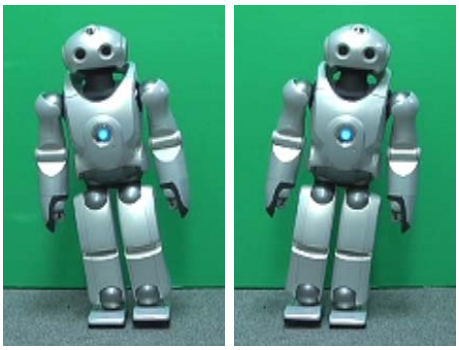


Fig. 14. Acquired stepping movement of the real robot after one iteration of the proposed learning process.

ments. We applied the proposed control approach to a small humanoid robot.

We took a similar approach to improve the model and controller, proposed by [18]–[20] – a method that improves policy parameters by using inaccurate models. Nevertheless, we proposed the use of a stochastic model to represent the stepping and walking tasks – since an acquisition of deterministic models that include ground contact being difficult.

In this study, we used a reinforcement learning method [15] to improve policy parameters. In future work, we will consider using a dynamic programming approach to efficiently update policy parameters using the Gaussian process model [21] since an analytical update using dynamic programming may reduce the number of iterations to achieve better task performance. We will also work on the application of the proposed method to our newly developed human-sized humanoid robot CB in Fig. 1(Left) [6].

ACKNOWLEDGMENT

We thank Sony Corp. for allowing us the opportunity to use the small humanoid robot. This material is based upon work supported in part by the DARPA Learning Locomotion Program and the National Science Foundation under grants CNS-0224419, DGE-0333420, ECS-0325383, and EEC-0540865.

REFERENCES

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

[2] 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*, 2004, p. (to appear).

[3] J. Morimoto, J. Nakanishi, G. Endo, G. Cheng, C. G. Atkeson, and G. Zeglin, "Poincaré-Map-Based Reinforcement Learning For Biped Walking," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2392–2397.

[4] T. Matsubara, J. Morimoto, J. Nakanishi, M. Sato, and K. Doya, "Learning Sensory Feedback to CPG with Policy Gradient for Biped Locomotion," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 4175–4180.

[5] G. Endo, J. Morimoto, J. Nakanishi, T. Matsubara, and G. Cheng, "Learning CPG Sensory Feedback with Policy Gradient for Biped Locomotion for a Full-body Humanoid," in *The Twentieth National Conference on Artificial Intelligence*, 2005, pp. 1267–1273.

[6] G. Cheng, S. Hyon, J. Morimoto, A. Ude, J. G. Hale, G. Colvin, W. Scroggin, and S. C. Jacobsen, "CB: A Humanoid Research Platform for Exploring NeuroScience," *Advanced Robotics*, vol. 21, no. 10, 2007.

[7] J. Morimoto, G. Endo, J. Nakanishi, S. Hyon, G. Cheng, C. G. Atkeson, and D. Bentivegna, "Modulation of Simple Sinusoidal Patterns by a Coupled Oscillator Model for Biped Walking," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006, pp. 1579–1584.

[8] L. Kuvayev and R. Sutton, "Model-based reinforcement learning with an approximate, learned model," in *Proceedings of the Ninth Yale Workshop on Adaptive and Learning Systems*, 1996, pp. 101–105.

[9] F. Miyazaki and S. Arimoto, "Implementation of a hierarchical control for biped locomotion," in *8th IFAC*, 1981, pp. 43–48.

[10] H. Miura and I. Shymoyama, "Dynamical walk of biped locomotion," *Int. J. of Robotics Research*, vol. 3, no. 2, pp. 60–74, 1984.

[11] T. Sugihara and Y. Nakamura, "Whole-body Cooperative COG Control through ZMP Mnipulation for Humanoid Robots," in *IEEE Int. Conf. on Robotics and Automation*, Washington DC, USA, 2002.

[12] S. Hyon and G. Cheng, "Passivity-based full-Body force control for humanoids and application to dynamic balancing and locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 4915–4922.

[13] S. H. Strogatz, *Nonlinear Dynamics and Chaos*. Addison-Wesley Publishing Company, 1994.

[14] C. K. I. Williams and C. E. Rasmussen, "Gaussian processses for regression," in *Advances in Neural Information Processing Systems*, vol. 8. The MIT Press, 1996, pp. 514–520.

[15] H. Kimura and S. Kobayashi, "An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value functions," in *Proceedings of the 15th Int. Conf. on Machine Learning*, 1998, pp. 284–292.

[16] K. Doya, "Reinforcement Learning in Continuous Time and Space," *Neural Computation*, vol. 12, no. 1, pp. 219–245, 2000.

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

[18] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proc. 14th International Conference on Machine Learning*. Morgan Kaufmann, 1997, pp. 12–20. [Online]. Available: citeseer.ist.psu.edu/atkeson97robot.html

[19] C. G. Atkeson, "Nonparametric model-based reinforcement learning," in *Advances in neural information processing systems 10*, M. K. M. I. Jordan and S. Solla, Eds. MIT Press, 1998, pp. 1008–1014.

[20] P. Abbeel, M. Quigley, and A. Y. Ng, "Using inaccurate models in reinforcement learning," in *Proceedings of the 23rd international conference on Machine learning*. ACM Press, 2006, pp. 1–8.

[21] C. E. Rasmussen and M. Kuss, "Gaussian processses in Reinforcement Learning," in *Advances in Neural Information Processing Systems*, vol. 16. The MIT Press, 2004, pp. 751–759.