Low-dimensional Feature Extraction for Humanoid Locomotion using Kernel Dimension Reduction

Jun Morimoto, Sang-Ho Hyon, Christopher G. Atkeson, and Gordon Cheng

***Abstract*— We propose using the kernel dimension reduction (KDR) to extract a low-dimensional feature space for humanoid locomotion tasks. Although humanoids have many degrees of freedom, task relevant feature spaces can be much smaller than the number of dimension of the original state space. We consider an application of the proposed approach to improve the locomotive performance of humanoid robots using an extracted low-dimensional state space. To improve the locomotive performance, we use a reinforcement learning (RL) framework. While 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 use the extracted low-dimensional feature space for RL so that the learning system can improve task performance quickly. The kernel dimension reduction method allows us to extract the feature space even if the task relevant mapping is non-linear. This is an essential property to improve humanoid locomotive performance since stepping or walking dynamics involves highly nonlinear dynamics. We show that we can improve stepping and walking policies by using a RL method on an extracted feature space by using KDR.**

I. Introduction

Reinforcement learning (RL), which does not require a precise environmental model, can be a useful technique to improve task performance of real robots. However, one drawback of utilizing RL is that it usually requires a large number of iterations to improve policies in a high-dimensional space. Thus, applications of RL have been limited to robots with small numbers of degrees of freedom [1]–[4].

In our approach, we propose using kernel dimension reduction (KDR) [5], [6] to extract a low-dimensional feature space so that the learning system can improve task performance even when a target robot model has many degrees of freedom including the complexity of humanoid robots.

In this study, we focus on improving the locomotive performance of humanoid robots as an application of our learning framework. The dynamics of biped robots characteristically includes contact and collision with the ground. Modeling the interaction with the ground can be very cumbersome. Using RL methods can be a suitable approach to improve biped walking. We use the extracted low-dimensional feature space for RL.

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

S. Hyon is with the Japan Science and Technology Agency, ICORP, Computational Brain Project, and with ATR Computational Neuroscience Laboratories. sangho@atr.jp

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

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

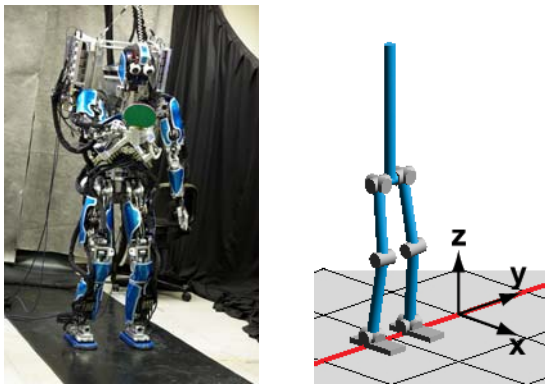


Fig. 1. (Left) Our human sized hydraulic humanoid robot CB developed by SARCOS. height: 1.59 m, total weight: 95 kg. (Right) Simplified 3D biped simulation model of our humanoid robot.

We extract the low-dimensional feature space from stepping and walking dynamics without explicitly identifying rigid body parameters and without the use of a ground contact model. KDR allows us to extract the feature space even if the task relevant mapping is non-linear. This is essential property to improve humanoid locomotive performance since stepping or walking dynamics involve highly nonlinear dynamics. We show that we can improve stepping and walking policies by using a RL method applied to the extracted feature space.

We apply our learning framework to a biped simulation model (see Fig. 1(Right)) of our humanoid robot CB (see Fig. 1(Left)) [7].

In our approach, we first construct a stepping and a walking controller based on our previous study [8]. The previous study proposed using the center of pressure to detect the phase of the robot dynamics for both stepping and walking (Fig. 2). We used simple periodic functions (sinusoids) as desired joint trajectories (see Appendix). We showed that synchronization of the desired trajectories at each joint with the detected phase from the center of pressure could generate stepping and walking movements. In this study, we modulate the amplitude of the sinusoids according to the current state of the robot to improve locomotive performance.

In Section II, our learning framework is introduced. In Section III-A, we explain how we applied the kernel dimension reduction (KDR) method to the tasks of stepping and walking. In Section III-B, we describe our implementation of a RL method on a low-dimensional feature space extracted

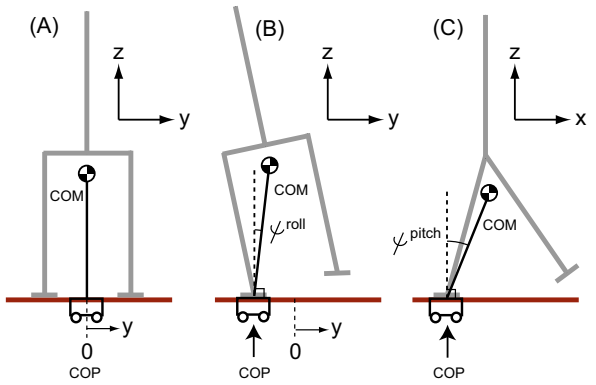


Fig. 2. Inverted pendulum model represented by the center of pressure (COP) and the center of mass (COM). ψ^{roll} denotes roll angle of the pendulum. ψ^{pitch} denotes pitch angle of the pendulum.

using KDR. In Section IV, we show our simulation results. Our biped controller as proposed in [8] is introduced in appendix.

II. Learning framework

In this study, we consider extracting a low-dimensional feature space to improve task performance by using KDR. We assume that nominal stepping and walking controllers are provided (see Appendix), and our learning system improves the performance of these controllers. Since the nominal controller can generate periodic movements, we only consider the robot 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 [9]. 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 defined by the COM and COP equaling zero $\dot{\psi}^{roll} = 0$ (see Fig. 2).

Since we consider an application of feature extraction to an RL framework, we are interested in figuring out the appropriate feature vector which can predict state variables $\mathbf{x}^r \in \mathbf{R}^r$ used to represent a reward function $r(\mathbf{x}^r)$. We then formulate the problem to find the state vector $\mathbf{x} \in \mathbf{R}^n$ as:

$$p(\mathbf{x}^r(k+1)|\mathbf{x}^f(k), \mathbf{u}(k)) \simeq p(\mathbf{x}^r(k+1)|\mathbf{x}(k), \mathbf{u}(k)), \quad (1)$$

where $\mathbf{x}^f \in \mathbf{R}^m$ denotes the state vector in the original state space. We consider a projection of the state to the low-dimensional feature vector $\mathbf{x} = B\mathbf{x}^f$ such that $n < m$, where B is a projection matrix. $\mathbf{u} \in \mathbf{R}^l$ is the output of a policy. Then, in our learning framework, we consider three sets of state variables: 1) the state variables used to represent reward function in the original state space $\mathbf{x}^r$, 2) the original state variables $\mathbf{x}^f$, and 3) the state variables in the feature space after projection $\mathbf{x}$.

The relationship in (1) implies that the low-dimensional feature space $\mathbf{x}(k)$ tries to keep the Markov property for the state $\mathbf{x}^r$. We use KDR to derive the projection matrix B.

The policy of the learning system is updated and outputs the next action only at this Poincaré section.

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

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

where $\mathbf{w}$ is the parameter vector of the policy $\pi_{\mathbf{w}}$. In the following sections, we explain how we extract the low-dimensional feature space and how we acquire the control policy $\pi_{\mathbf{w}}$.

III. Feature extraction and policy improvement

A. Kernel dimension reduction

We use kernel dimension reduction (KDR) [5], [6] to extract a low-dimensional feature space. Here we consider a regression problem in which we try to explain a variable $Y \in \mathcal{Y}$ by using a variable $X \in \mathcal{X}$. The target of KDR is to find a low-dimensional subspace $\mathcal{Z}$ of the original input space $\mathcal{X}$ such that we can keep information of the variable Y even in the subspace $\mathcal{Z}$.

More concretely, KDR tries to find a projection matrix B, which specifies the subspace of the original high-dimensional state space, to make the conditional distribution $p(Y|Z)$ close to $p(Y|X)$:

$$p(Y|X) \simeq p(Y|Z), \quad (3)$$

where $Z = B^T X$. In our study, Y corresponds to $\mathbf{x}^r(k+1)$, X includes $\mathbf{x}^f$ and $\mathbf{u}$, and Z includes $\mathbf{x}$ and $\mathbf{u}$ in (1).

This method does not impose assumptions on either the distributions of X or the conditional distribution $P(Y|X)$.

The idea of KDR is to map random variables X and Y to reproducing kernel Hilbert spaces (RKHS) [10], [11] and evaluate conditional independence using cross-covariance operators.

Let $\mathcal{H}_{\mathcal{X}}$ be an RKHS of functions on $\mathcal{X}$ induced by the kernel function $k_{\mathcal{X}}(\cdot, X)$ for $X \in \mathcal{X}$. We also define the space $\mathcal{H}_{\mathcal{Y}}$ and the kernel function $k_{\mathcal{Y}}(\cdot, Y)$. Then, we define the cross-covariance between a pair of functions $f \in \mathcal{H}_{\mathcal{X}}$ and $g \in \mathcal{H}_{\mathcal{Y}}$ as follows:

$$\begin{aligned} &< g, \Sigma_{YX} f >_{\mathcal{H}_{\mathcal{Y}}} \\ &\quad = E_{XY}\left[(f(X) - E_X[f(X)])(g(Y) - E_Y[g(Y)])\right] \quad (4) \end{aligned}$$

for all functions f and g, where Σ_{YX} is a cross-covariance operator. Similarly, we define covariance operators $\boldsymbol{\Sigma}_{XX}$ and $\boldsymbol{\Sigma}_{YY}$. Now, we can use these operators to define a conditional cross-covariance operators as:

$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY} \quad (5)$$

This definition assumes that Σ_{XX} is invertible.

To derive the projection matrix B, we try to minimize $\mathrm{Tr}\left[\hat{\Sigma}_{YY|Z}\right]$, where $\hat{\Sigma}_{YY|Z}$ is the empirical conditional covariance operator which corresponds to (5).

Let $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ denote N samples from the joint distribution $p(X, Y)$, and let $\mathbf{K}_Y \in \mathbf{R}^{N \times N}$ and $\mathbf{K}_Z \in \mathbf{R}^{N \times N}$ denote the Gram matrices computed over $\{\mathbf{y}_i\}$ and $\{z_i =$

$B^T\mathbf{x}_i\}$. Then, this minimization problem can be formulated in terms of $\mathbf{K}_Y$ and $\mathbf{K}_Z$, so that the optimal projection matrix B^* is derived as:

$$B^* = \arg\min_B \mathrm{Tr}\left[\mathbf{K}_Y^c(\mathbf{K}_Z^c + N\epsilon\mathbf{I}_N)^{-1}\right] \quad (6)$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix, and ϵ is a regularization coefficient [5], [6]. The matrix $\mathbf{K}^c$ denotes the centered kernel matrices

$$\mathbf{K}^c = \left(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\right)\mathbf{K}\left(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\right), \quad (7)$$

where $\mathbf{1}_N = (1, \ldots, 1)$ is the vector with all elements equal to 1. We used a Gaussian kernel:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2s^2}\right), \quad (8)$$

where K_{ij} is the ij element of the Gram matrix, and s denotes the parameter of the kernel.

B. Policy improvement by using a reinforcement learning method

We use a policy gradient method proposed by [12] to improve the the stepping and walking policies.

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:

$$E_{\pi_{\mathbf{w}}}\{V(k)\} = E_{\pi_{\mathbf{w}}}\left\{\sum_{i=k}^{\infty}\gamma^{i-k}r(i)\right\}, \quad (9)$$

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 this study, $\mathbf{x}$ represents an extracted feature vector and $\mathbf{u}$ denotes the output of a policy. We modulate amplitudes of sinusoidal patterns as the output of our stepping and walking policies (see Section IV). Both the approximated value functions and the policies are represented by a normalized Gaussian network [13].

IV. SIMULATION

We applied our proposed method to a simplified simulation model of our humanoid robot CB [7] (Fig. 1(Right)).

A. Improvement of biped stepping performance

We applied our proposed method to improve stepping in place.

We modulate the amplitude A_{step} in (16) to improve the stepping performance. We defined the target of the stepping task to keep the desired state at ψ^{roll}. We use a reward function:

$$r = -0.1(\psi_d^{roll} - \psi^{roll})^2 \quad (10)$$

for this stepping task, where $\psi_d^{roll} = 2.0°$. The learning system also receives a negative reward $r = -1$ if the biped model falls over.

Since the reward function is defined as a function of the variable ψ^{roll}, we try to find a low-dimensional feature space that can predict ψ^{roll} at the next step by using KDR. We use 200 samples to find the projection matrix B. The parameter of the kernel in (8) is $s = 0.75$.

The simplified humanoid model has 10 joints, and the base link has 6 degrees of freedom. We did not consider translational degrees of freedom of the base link since our humanoid robot CB does not have a sensor to detect the 3D position of the base link. Therefore, we need to consider 13 degrees of freedom. As a consequence, we need to consider a 26 dimensional state space as the original state space that includes the time derivative of each degree of freedom. We applied KDR for the original 26 dimensional state space to find the proper projection to a 1-dimensional feature space. The number of dimensions of the feature space is predetermined.

Figure 3 shows learning performance of the stepping task on the extracted feature space. This results showed that the extracted feature can be used to learn the stepping task.

Here we compared with the learning performance using randomly extracted feature space. We randomly selected the elements of the projection matrix B such that $B^TB = 1$ for this comparison. Figure 3 shows that the learning system could not acquire a good stepping policy and had large variance in stepping performance when we used a randomly selected feature space.

Figure 4 shows the relationship between the extracted 1-dimensional feature and the roll angle ψ^{roll}. This result showed that KDR extracted a 1-dimensional feature, which has high correlation to the roll angle (correlation coefficient was 0.93). This is interesting because the learning system automatically finds a feature space which corresponds to the roll angle of the center of mass while a number of biped walking studies have emphasized that humanoid robots have inverted pendulum dynamics (see Fig. 2), with the top of the pendulum at the center of mass and the base at the center of pressure [14]–[17].

Note that the reward function is a function of the roll angle. However, it is not obvious whether the proper variable to predict the roll angle at the next time step $(k + 1)$ is the roll angle at current time (k) or not.

Figure 5 shows a comparison between control performance of an acquired policy and that of the initial policy. The roll angle is kept around desired state by using the acquired policy.

An acquired stepping movement is shown in Fig. 6.

B. Improvement of biped walking performance

We also applied our proposed method to improve walking performance.

We modulate the amplitude A_{walk} in (17) to generate forward movement for the biped walking task. The target of the walking task is to increase the angular velocity of the pendulum $\dot{\psi}^{pitch}$ (see Fig 2(C)) at the Poincaré section. We use the reward function:

$$r = 0.1(\dot{\psi}^{pitch}) \quad (11)$$

for this biped walking task. The learning system also receives a negative reward $r = -1$ if the biped model falls over.

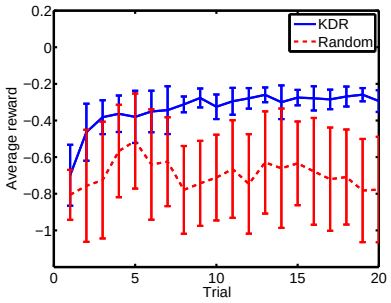


Fig. 3. Learning performance of the stepping controller in the simulated environment. The blue solid line represents the learning performance using a feature space extracted by KDR. The red dashed line represents the learning performance using a randomly selected feature space. The learning system could not acquire a good stepping policy and had large variance in stepping performance when we used a randomly selected feature space. Means and standard deviations of the learning performances were derived from 10 simulation runs.

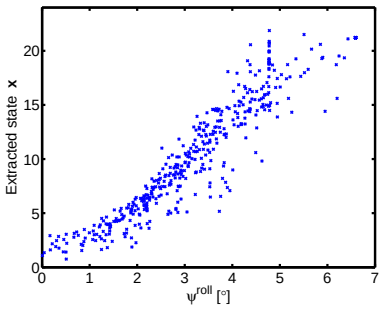


Fig. 4. Relationship between the COM roll angle ψ^{roll} and the extracted feature $\mathbf{x}$. The extracted feature is highly correlated to the COM roll angle ψ^{roll}. The correlation coefficient was 0.93. The extracted feature captures physical property of the robot model.

Since the reward function is defined as a function of the variable $\dot{\psi}^{pitch}$, we try to find the low-dimensional feature space that can predict $\dot{\psi}^{pitch}$ at next step by using KDR. In this case, we also use 200 samples to find the projection matrix B. The parameter of the kernel in (8) is $s = 0.75$.

We applied KDR for the original 26 dimensional state space to find the proper projection to a 3-dimensional feature space. The number of dimensions of the feature space is predetermined.

Figure 7 shows learning performance of the walking task on the feature extracted by using KDR. This results showed that the extracted feature can be used to learn the walking task.

We again compared with the learning performance using randomly extracted feature space. Since the amount of the reward could be mostly explained only by the output of policies A_{walk}, the learning system could acquire walking policies even using a randomly selected feature space. However, learning performance kept increasing only when we used the feature space extracted by KDR possibly because the extracted feature space has a larger emphasis on the reward.

Figure 8 shows a comparison between control performance of an acquired policy and that of the initial policy for the

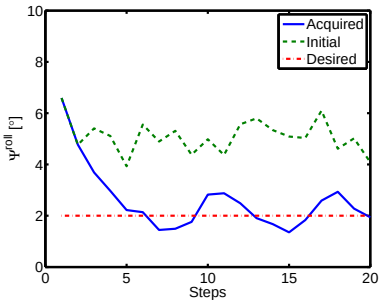


Fig. 5. Comparison between control performance of an acquired policy and that of the initial policy. The roll angle ψ^{roll} at the Poincaré section $\dot{\psi}^{roll} = 0$ (solid line). The dotted line represents the desired angle for this stepping task.

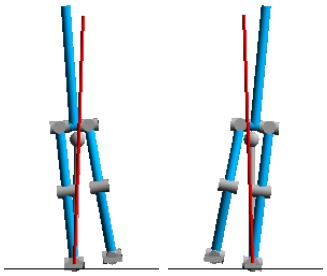


Fig. 6. Acquired stepping movement. The red thin line represents desired angle. the pendulum state represented by the light gray line behind the red line came close to the desired state at the Poincaré section. The light gray sphere represents the center of mass.

walking task. The pitch angular velocity $\dot{\psi}^{pitch}$ had a larger value using the acquired policy compared to using the initial policy.

Figure 9 shows the acquired walking performance.

V. DISCUSSION

We proposed using KDR to extract a low-dimensional feature space for a humanoid locomotion task. In this study, we used the extracted low-dimensional state space for RL so that the learning system could improve task performance. We showed that we could improve stepping and walking policies by using a RL method on the extracted feature space by using KDR. In our future work, we will consider application of the proposed method to our new humanoid robot CB [7].

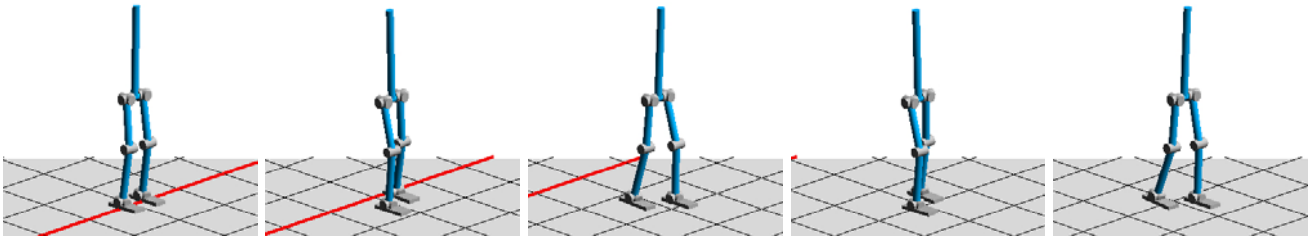
So far, we empirically determined the sufficient dimensionality of the feature space. Automatic selection of the number of dimensions for the humanoid locomotion tasks is also included as part of our future work.

ACKNOWLEDGMENT

This material is based upon work supported in part by the National Science Foundation under grants CNS-0224419, DGE-0333420, ECS-0325383, and EEC-0540865.

APPENDIX

Our biped controller uses a coupled phase oscillator model to modulate the phase of the sinusoidal patterns. The aim



(a) t=0.0 s (b) t=3.2 s (c) t=6.5 s (d) t=9.7 s (e) t=13.0 s

Fig. 9. Acquired walking pattern. Average walking speed is 0.16 m/s for initial 20 seconds. The red line represents the starting position. We showed one snap in three walking steps.

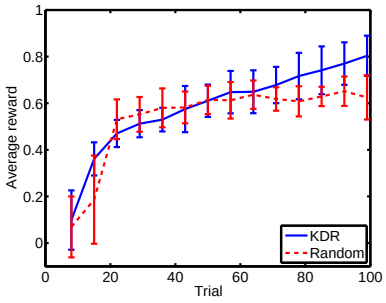


Fig. 7. Learning performance for the walking task in the simulated environment. The blue solid line represents the learning performance using a feature space extracted by KDR. The red dashed line represents the learning performance using a randomly selected feature space. Since the amount of the reward could be mostly explained only by the output of policies A_{walk}, the learning system could acquire walking policies even using a randomly selected feature space. However, learning performance kept increasing only when we used the feature space extracted by KDR possibly because the extracted feature space has a larger amount of information on the reward. Means and standard deviations of the learning performances were derived from 10 simulation runs.

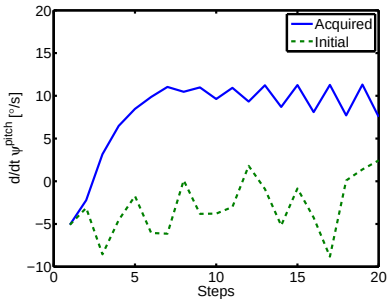


Fig. 8. Comparison between control performance of an acquired policy and that of the initial policy. The pitch angular velocity $\dot{\psi}^{pitch}$ at the Poincaré section $\psi^{roll} = 0$ (solid line).

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 A. We briefly explain phase coordination for biped walking in Section B. As in our previous study [8], we use simple sinusoidal patterns as nominal trajectories for each joint. We describe the design of the nominal trajectories for stepping movement in Section C, and walking movement in Section D. Parameters used in the controllers are summarized in Table I.

A. Phase detection of the robot dynamics

As shown by our previous study [8], we can 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:

$$\phi(\mathbf{y}_{cop}) = -\arctan\left(\frac{\dot{y}_{cop}}{y_{cop}}\right), \tag{12}$$

where $\mathbf{y}_{cop} = (y_{cop}, \dot{y}_{cop})$ (see Fig. 2). We use a simplified COP detection method introduced in [8].

B. Phase coordination

In this study, we use four oscillators with phases ϕ_c^i, where $i = 1, 2, 3, 4$. We introduce couplings between the oscillators and the phase of the robot dynamics $\phi(\mathbf{y}_{cop})$ in (12) to regulate the desired phase relationship between the oscillators:

$$\dot{\phi}_c^i = \omega_c + K_c \sin(\phi(\mathbf{y}_{cop}) - \phi_c^i + \alpha^i), \tag{13}$$

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, $\{\alpha^1, \alpha^2, \alpha^3, \alpha^4\} = \{-\frac{1}{2}\pi, 0.0, \frac{1}{2}\pi, \pi\}$, to make symmetric patterns for a stepping movement with the left and right limbs (see Section C.2), and also to make symmetric patterns for a forward movement with the left and right limbs (see Section 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. 10(A)) for this movement. Desired joint angles for each joint are:

$$\theta^d_{h\_roll}(\phi_c) = A_{h\_roll} \sin(\phi_c), \tag{14}$$

$$\theta^d_{a\_roll}(\phi_c) = -A_{a\_roll} \sin(\phi_c), \tag{15}$$

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 $\phi_c = \phi_c^1$.

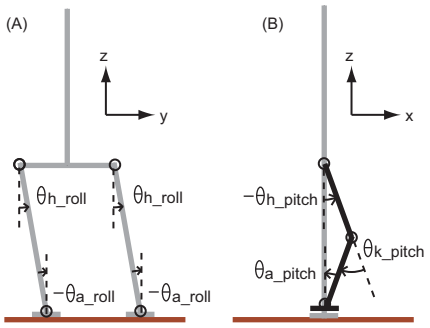


Fig. 10. Stepping controller: (A) Controller for side-to-side movement. (B) Controller for foot clearance.

2) Vertical foot movement to make clearance: To achieve foot clearance, we generate vertical movement of the feet (Fig. 10(B)) by using simple sinusoidal trajectories:

$$\begin{aligned}\theta^d_{h\_pitch}(\phi_c) &= (A_{pitch} + A_{step})\sin(\phi_c) + \theta^{res}_{h\_pitch},\\ \theta^d_{k\_pitch}(\phi_c) &= -2(A_{pitch} + A_{step})\sin(\phi_c) + \theta^{res}_{k\_pitch},\\ \theta^d_{a\_pitch}(\phi_c) &= -(A_{pitch} + A_{step})\sin(\phi_c) + \theta^{res}_{a\_pitch}, \quad (16)\end{aligned}$$

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 $\phi_c = \phi^1_c$ for right limb movement and use the oscillator with phase $\phi_c = \phi^3_c$, which has phase difference of $\phi^3_c = \phi^1_c + \pi$, 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 (see Sections IV-A).

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:

$$\begin{aligned}\theta^d_{h\_pitch} &= A_{pitch}\sin(\phi^1_c) + A_{walk}\sin(\phi^2_c) + \theta^{res}_{h\_pitch},\\ \theta^d_{a\_pitch} &= -A_{pitch}\sin(\phi^1_c) - A_{walk}\sin(\phi^2_c) + \theta^{res}_{a\_pitch}, \quad (17)\end{aligned}$$

where the phase $\phi_c = \phi^2_c$ has $\frac{1}{2}\pi$ phase difference of ϕ^1_c. We use ϕ^3_c and ϕ^4_c for left limb instead of ϕ^1_c and ϕ^2_c, where the phase $\phi_c = \phi^4_c$ has π phase difference of ϕ^2_c. We then modulate the amplitude of the sinusoidal patterns by changing A_{walk} according to the current pendulum state of the biped walking task (see Section IV-B).

TABLE I

PARAMETERS OF THE BIPED STEPPING AND WALKING MODEL FOR EACH EXPERIMENT

	Stepping	Walking
ω_c	3.5	3.5
K_c	10.0	10.0
A_{h_roll}	4.0	3.5
A_{a_roll}	4.0	3.5
A_{pitch}	6.0	7.0

REFERENCES

[1] J. Morimoto and K. Doya, "Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning," *Robotics and Autonomous Systems*, vol. 36, pp. 37–51, 2001.

[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 Intelligent Robots and Systems*, 2004, pp. 2849–2854.

[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 CPG-based Biped Locomotion with a Policy Gradient Method," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 911–920, 2006.

[5] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *Journal of Machine Learning Research*, vol. 5, pp. 73–99, 2004.

[6] K. Fukumizu, , F. R. Bach, and M. I. Jordan, "Kernel dimension reduction in regression," *Technical Report of the Department of Statistics, University of California, Berkeley*, 2006.

[7] 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.

[8] 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.

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

[10] N. Aronszajn, "Theory of rproducing kernels," *Trans. Amer. Math. Soc.*, vol. 69, no. 3, pp. 337–404, 1950.

[11] B. Scholkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: The MIT Press, 2002.

[12] 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.

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

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

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

[16] 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.

[17] 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.